SIEMENS

SINUMERIK

SINUMERIK ONE
PLC

Function Manual

Preface

Introduction 1

Fundamental safety instructions 2

Overview 3

Commissioning the PLC program 4

Handling of blocks and libraries 5

Inserting the PLC basic program into the project 6

Specifying the execution structure and OBs of the PLC program 7

NC process alarms 8

NC variables 9

Reference on the PLC basic program 10

Appendix A

Valid for

Control system
SINUMERIK ONE

CNC software Version 6.14

10/2020
A5E48053808B AD

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ **DANGER**
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ **WARNING**
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ **CAUTION**
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ **WARNING**
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E48053808B AD
Ⓟ 10/2020 Subject to change

Copyright © Siemens AG 2019 - 2020.
All rights reserved

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

Training

At the following address (http://www.siemens.com/sitrain), you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

Target group

This publication is intended for:

- Project engineers
- Technologists (from machine manufacturers)
- System startup engineers (Systems/Machines)
- Programmers

Benefits

The function manual describes the functions so that the target group knows them and can select them. It provides the target group with the information required to implement the functions.

Standard version

This documentation only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Further, for the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation or maintenance.

Note regarding the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the principle of privacy by design. That means that

this product does not process / store any personal data, only technical functional data (e.g. time stamps). If a user links this data with other data (e.g. a shift schedule) or stores personal data on the same storage medium (e.g. hard drive) and thus establishes a link to a person or persons, then the user is responsible for ensuring compliance with the relevant data protection regulations.

Technical Support

Country-specific telephone numbers for technical support are provided on the Internet at the following address (https://support.industry.siemens.com/cs/ww/en/sc/2090) in the "Contact" area.

If you have any technical questions, use the online form in the "Support Request" area.

Programming and migration

Recommendations for programming and the PLC migration sequence are subsequently described.

Guidelines and sequence for PLC migration

Guidelines on migrating existing PLC programs.

- Migrating the standard parts of PLC user programs (see SINUMERIK migration guidelines).
- New programming of the safety part of PLC user programs (see the SINUMERIK Safety Function description).

Guidelines for programming new PLC user programs

When programming new PLC user programs, we recommend complying with the TIA Portal programming guidelines (see below). As a consequence, you ensure that programs can be simply modularized - and are therefore more transparent and with a higher performance.

Use symbolic programming for accessing data. In the block descriptions, the absolute addresses are specified in brackets for reasons of compatibility and ease of migration.

Recommendations for programming and adapting new PLC user programs are listed in the following table:

Recommendation	Description
Using libraries	See also guidelines for standardization in companies
Wherever possible, use UDTs (no anonymous structures)	Improved reusability, higher runtime performance for known data types
Activate the IEC check (according to IEC 61131) in the block properties	Non-compatible operands must be explicitly converted.
Use IEC timer & IEC counter, wherever possible, do not use bit memories	
Eliminate pointer arithmetic (memory access operations) and ideally implement in SCL	It is necessary to have a good understanding about the previous programming logic used
Remove STL, and preferably implement in SCL	It certainly makes sense to still use LAD / FBD for logical interconnections.
Use symbols in conformance with the TIA Portal style guide for your own functions/blocks	
Programming in conformance with the TIA Portal style guide for user program parts that do not access the interface, can be innovated independently	If at all possible, encapsulate blocks that to do not come into contact with the VDI interface (preferred approach is to use call parameters without accessing global variables within the block)

Further information

- Information about programming in conformance with the style guide is available at Programming guidelines (https://support.industry.siemens.com/cs/de/en/view/81318674).
- Information about handling libraries is available at Library handling (https://support.industry.siemens.com/cs/de/en/view/109747503).

Table of contents

Preface **3**

1 Introduction **15**

1.1 About this documentation 15

2 Fundamental safety instructions **17**

2.1 General safety instructions 17

2.2 Warranty and liability for application examples 17

2.3 Security information 17

3 Overview **19**

3.1 Introduction 19

4 Commissioning the PLC program **21**

4.1 Overview of the PLC program 21

4.2 Installation of the basic program 21

4.3 Application of the basic program 22

4.4 Version codes 23

4.5 Machine program 24

4.6 Data backup 25

4.7 PLC series startup, PLC archive 25

4.8 I/O modules (CP modules) 26

5 Handling of blocks and libraries **27**

5.1 Use and handling of groups 27

5.2 Copying blocks into a project 29

5.3 Conflicts when copying blocks 32

5.4 Restrictions that apply to the user blocks in libraries 33

5.5 Block address range for PLC user program 33

6 Inserting the PLC basic program into the project **37**

6.1 Opening the PLC basic program system library 37

6.2 Inserting PLC data types 38

6.3 Inserting PLC basic program blocks 40

6.4 Blocks with user-specific adaptations 43

7 Specifying the execution structure and OBs of the PLC program **45**

7.1 Overview of the PLC program 45

7.2 Execution structure of the PLC program ... 45
7.3 Relevant organization blocks ... 49
7.4 Inserting program code for calling the basic program ... 49
7.5 Programming examples (SCL) ... 51
7.5.1 Example for calling LBP_ConfigBP [FC1] ... 51
7.5.2 Examples for calling LBP_MainBP [FC2] ... 52
7.5.3 Example for calling LBP_NCKProcessIRT [FC3] ... 53
7.6 Retentivity behavior of the PLC basic program ... 55
8 NC process alarms ... 57
8.1 General information on hardware interrupt OBs ... 57
8.2 Transfer and acknowledgment time of NCK signals ... 57
8.3 Identification of an NCK event in the PLC program ... 58
8.4 Defining NCK hardware interrupt event ... 59
8.5 Use of the associated system constant ... 61
9 NC variables ... 63
9.1 Overview of tag configuring ... 63
9.2 Corresponding data type declarations for NCK and PLC ... 63
9.3 NC variable configuration in TIA Portal ... 66
9.3.1 Relevant objects and resources ... 66
9.3.2 Types of NC variables in the TIA Portal ... 68
9.3.3 Working with NC variable tables ... 68
9.3.3.1 Creating and managing a user-defined variable table ... 68
9.3.3.2 Managing entries in variable tables ... 70
9.3.3.3 Adapt the view of the variable table ... 71
9.3.4 Working with the list of available NC variables ... 71
9.3.4.1 Searching and filtering NC variables ... 71
9.3.4.2 Use of existing NC variables (NC-VAR selector) ... 74
9.3.4.3 Taking over NC variables into the tag table of the configured NC variables ... 75
9.3.5 Configuring selected NC variable instances ... 76
9.3.6 Integrating into the PLC program ... 78
9.3.6.1 Reading/writing the NC variables in the PLC program ... 78
9.3.6.2 Creating a data block from an NC variable table ... 80
9.3.6.3 Copying configured NC variables to an existing DB/FB ... 82
9.3.6.4 Corresponding data type declarations for NCK and PLC ... 83
9.3.7 Managing NC variables in libraries ... 85
10 Reference on the PLC basic program ... 87
10.1 Overview of the PLC blocks ... 87
10.1.1 Data blocks (DB) and associated PLC user data types ... 87
10.1.2 Function blocks (FB) ... 89
10.1.3 Function blocks (FC) ... 91
10.2 Operator panels ... 94
10.2.1 Introduction ... 94
10.3 Data types ... 95

10.3.1 LBP_typeHandwheel ... 95
10.3.2 LBP_typeMcp* ... 95
10.3.3 LBP_typeModularMcpKeypadRecord ... 96

10.4 Data blocks (DBs) ... 96
10.4.1 Overview of the interface signals ... 97
10.4.2 Overview of further interface signals (ONE) ... 97
10.4.3 LBP_AlarmMsgs [DB2] - PLC alarms / messages ... 98
10.4.3.1 Overview ... 98
10.4.3.2 LBP_AlarmMsgs [DB2] ... 112
10.4.3.3 STRUCT "Chan" alarms Channel without message ... 113
10.4.3.4 STRUCT "Axis" alarms Axis without message ... 114
10.4.3.5 STRUCT "ChanA" alarms Channel with message ... 114
10.4.3.6 STRUCT "AxisA" alarms Axis with message ... 116
10.4.3.7 STRUCT "UserA" alarms User area with message ... 117
10.4.4 LBP_ParamAlarmMsgs [DB5] - message type in LBP_AlarmMsgs [DB2] ... 117
10.4.4.1 LBP_ParamAlarmMsgs [DB5] ... 117
10.4.5 LBP_ConfigData [DB7] - Configuration data ... 121
10.4.5.1 LBP_ConfigData [DB7] ... 121
10.4.5.2 STRUCT Pointer to data block, input or output ... 132
10.4.5.3 STRUCT "GenerateAlarmMsgs" alarm message ... 132
10.4.6 LBP_NC [DB10] - Signals from/to the NC, PLC and operating software ... 133
10.4.6.1 Overview ... 133
10.4.6.2 LBP_NC [DB10] ... 145
10.4.7 LBP_ModeGroup [DB11] - Mode-group-specific signals ... 184
10.4.7.1 Overview ... 184
10.4.7.2 LBP_ModeGroup [DB11] ... 187
10.4.8 LBP_HMI [DB19] - Signals from/to the operator panel (OP) ... 209
10.4.8.1 Overview ... 209
10.4.8.2 LBP_HMI[DB19] ... 215
10.4.9 LBP_Chan1 [DB21], ... - Channel-specific signals ... 241
10.4.9.1 Overview ... 241
10.4.9.2 LBP_Chan1 [DB21], ... - Control signals to channel (1) ... 258
10.4.9.3 LBP_Chan1 [DB21], ... - Control signals to geometry axes ... 282
10.4.9.4 LBP_Chan1 [DB21], ... - HMI signals to channel / OEM signals from/to channel ... 282
10.4.9.5 LBP_Chan1 [DB21], ... - Control signals from geometry axes ... 319
10.4.9.6 LBP_Chan1 [DB21], ... - Change signals for auxiliary function transfer from the channel ... 319
10.4.9.7 LBP_Chan1 [DB21], ... - Transferred M and S functions ... 322
10.4.9.8 LBP_Chan1 [DB21], ... - Transferred T/D/DL functions ... 323
10.4.9.9 LBP_Chan1 [DB21], ... - Transferred H/F functions ... 326
10.4.9.10 LBP_Chan1 [DB21], ... - Decoded M signals ... 327
10.4.9.11 LBP_Chan1 [DB21], ... - Active G functions ... 328
10.4.9.12 LBP_Chan1 [DB21], ... Protection areas from channel ... 330
10.4.9.13 LBP_Chan1 [DB21], ... - Synchronized actions, signals from/to channel ... 332
10.4.9.14 LBP_Chan1 [DB21], ... - Control signals from/to channel ... 333
10.4.9.15 LBP_Chan1 [DB21], ... - Signals to orientation axes ... 340
10.4.9.16 LBP_Chan1 [DB21], ... - Signals from orientation axes ... 341
10.4.9.17 LBP_Chan1 [DB21], ... - Tool management functions from channel ... 341
10.4.9.18 LBP_Chan1 [DB21], ... - Control signals from/to channel (2) ... 343
10.4.9.19 LBP_Chan1 [DB21], ... - control signals from/to HMI ... 349
10.4.9.20 STRUCT "A_Geo", control signals for geometry axis ... 350
10.4.9.21 STRUCT "E_Geo", control signals of the geometry axis ... 359
10.4.9.22 STRUCT "A_Ori" Control signals for orientation axes ... 366

10.4.9.23 STRUCT "E_Ori" control signals of the orientation axis 375
10.4.10 LBP_Axis 1 [DB31], ... - Axis/spindle signals 381
10.4.10.1 Overview 381
10.4.10.2 LBP_Axis1 [DB31], ... - Signals to the axis/spindle 388
10.4.10.3 LBP_Axis1 [DB31], ... - Signals from the axis/spindle 449
10.4.10.4 STRUCT "SCC" PLC to drive 511
10.4.10.5 STRUCT "SIC" drive to PLC 512
10.4.11 LBP_TMLoad [DB71] - Interface for loading/unloading the magazine: 515
10.4.11.1 Overview 515
10.4.11.2 LBP_TMLoad [DB71] 516
10.4.11.3 STRUCT - "TIF" for loading/unloading the magazine: 517
10.4.12 LBP_TMSpindle [DB72] - Spindle as change position 522
10.4.12.1 Overview 522
10.4.12.2 LBP_TMSpindle [DB72] 524
10.4.12.3 STRUCT "TIF" for spindle as change position 525
10.4.13 LBP_TMTurret [DB73] - Turret 532
10.4.13.1 Overview 532
10.4.13.2 LBP_TMTurret [DB73] 534
10.4.13.3 STRUCT "TIF" for turret 535
10.4.14 LBP_MFuncDecListConfig [DB75] - Decoding list for M decoding 541
10.4.14.1 M decoding acc. to list 541
10.4.14.2 LBP_MFuncDecListConfig [DB75] 545
10.4.14.3 STRUCT "MSigGrp" group parameter 545
10.4.15 LBP_MFuncDecListSignals [DB76] - Signal list for M decoding 546
10.4.15.1 LBP_MFuncDecListSignals [DB76] 546
10.4.15.2 STRUCT "MSigGrp*" signal list of a group 546
10.4.16 LBP_PowerrideConfig [DB78] - preconfigured behavior for Powerride 547
10.4.16.1 Overview 547
10.4.16.2 LBP_PowerrideConfig [DB78] 549
10.4.16.3 UDT "LBP_typePowerrideInput" input data 550
10.4.16.4 UDT "LBP_typePowerrideOutput" output data 551
10.4.16.5 UDT "LBP_typePowerrideConfigPos" configuration data for a position 552
10.4.16.6 UDT "LBP_typePowerrideConfig" configuration data 553
10.4.17 LBP_CtrlEnergy [DB1000] - Energy-saving profiles 554
10.4.17.1 Overview 554
10.4.17.2 LBP_CtrlEnergy [DB1000] 555
10.4.17.3 STRUCT "Profiles" for energy-saving profiles 556
10.4.18 LBP_SentronPac [DB1001] - SENTRON PAC 559
10.4.18.1 Overview 559
10.4.18.2 LBP_SentronPac [DB1001] 562
10.4.18.3 STRUCT "CtrlE_AuxDevice" for SENTRON PAC auxiliary units 567
10.4.19 LBP_SpindleTempSensor [DB1002] - Spindle temperature sensors 570
10.4.19.1 Overview 570
10.4.19.2 LBP_SpindleTempSensor [DB1002] 572
10.4.19.3 STRUCT "SP" for spindle 572
10.4.19.4 STRUCT "TS" for temperature sensors 573
10.4.20 LBP_TMLoadMT [DB1071] - Loading/unloading magazine: Multitool 575
10.4.20.1 Overview 575
10.4.20.2 LBP_TMLoadMT [DB1071], interface for loading/unloading the magazine: Multitool 575
10.4.20.3 STRUCT Load/unload MTIF for magazine Multitool 576
10.4.21 LBP_TMSpindleMT [DB1072] - Spindle: Multitool 577
10.4.21.1 Overview 577

10.4.21.2 LBP_TMSpindleMT [DB1072], interface for the spindle: Multitool ... 578
10.4.21.3 STRUCT Load/unload MTIF for spindle: Multitool ... 578
10.4.22 LBP_TMTurretMT [DB1073] - Turret: Multitool ... 583
10.4.22.1 Overview ... 583
10.4.22.2 LBP_TMTurretMT [DB1073], interface for the turret: Multitool ... 584
10.4.22.3 STRUCT MTIF for turret: Multitool ... 584

10.5 Input and output signals ... 589
10.5.1 General ... 589
10.5.2 Input signals ... 589
10.5.2.1 M version, signals from the MCP: Input image ... 589
10.5.2.2 T version, signals from the MCP: Input image ... 590
10.5.2.3 Slimline version, signals from the MCP: Input image ... 590
10.5.2.4 Signals from the handheld unit: Input image ... 591
10.5.2.5 Signals from the handheld unit HT 8: Input image ... 591
10.5.2.6 Signals from the handheld unit HT 10: Input image ... 592
10.5.3 Output signals ... 593
10.5.3.1 M version, signals to the MCP: Output image ... 593
10.5.3.2 T version, signals to the MCP: Output image ... 593
10.5.3.3 Slimline version, signals to the MCP: Output image ... 594
10.5.3.4 Signals to the handheld unit: Output image ... 595
10.5.3.5 Signals to handheld terminal HT 8: Output image ... 596
10.5.3.6 Signals to handheld terminal HT 10: Output image ... 596

10.6 Function blocks (FB) ... 597
10.6.1 LBP_ReadVar [FB2]: Read NC variable ... 597
10.6.1.1 LBP_ReadVar [FB2]: Read NC variable ... 597
10.6.1.2 Generating a data block for accessing NC variables ... 604
10.6.2 LBP_WriteVar [FB3]: Write NC variable ... 605
10.6.3 LBP_ReadGUD [FB5]: Read GUD variable ... 611
10.6.4 LBP_ReqPIService [FB7]: Request PI service ... 618
10.6.4.1 List of available Pl services ... 621
10.6.4.2 PI service: ASUP ... 622
10.6.4.3 PI service: CANCEL ... 623
10.6.4.4 PI service: CONFIG ... 624
10.6.4.5 PI service: DIGIOF ... 624
10.6.4.6 PI service: DIGION ... 625
10.6.4.7 PI service: FINDBL ... 625
10.6.4.8 PI service: LOGIN ... 626
10.6.4.9 PI service: LOGOUT ... 626
10.6.4.10 PI service: NCRES ... 627
10.6.4.11 PI service: RETRAC ... 627
10.6.4.12 PI service: SELECT ... 628
10.6.4.13 PI service: SETUDT ... 629
10.6.4.14 PI service: SETUFR ... 629
10.6.4.15 PI service: CRCEDN ... 630
10.6.4.16 PI service: CREACE ... 630
10.6.4.17 PI service: CREATO ... 631
10.6.4.18 PI service: DELECE ... 631
10.6.4.19 PI service: DELETO ... 632
10.6.4.20 PI service: MMCSEM ... 632
10.6.4.21 PI service: TMCRTO ... 633
10.6.4.22 PI service: TMFDPL ... 634

10.6.4.23 PI service: TMFPBP 635
10.6.4.24 PI service: TMGETT 637
10.6.4.25 PI service: TMMVTL 637
10.6.4.26 PI service: TMPOSM 639
10.6.4.27 PI service: TMPCIT 640
10.6.4.28 PI service: TMRASS 640
10.6.4.29 PI service: TRESMO 641
10.6.4.30 PI service: TSEARC 641
10.6.4.31 PI service: TMCRMT 644
10.6.4.32 PI service: TMDLMT 645
10.6.4.33 PI service: POSMT 646
10.6.4.34 PI service: FDPLMT 647
10.6.5 LBP_Powerride [FB21]: rotary/push control dial 648
10.6.5.1 Signal transfer between Powerride and PLC 653
10.6.5.2 LED color palette 653
10.6.5.3 Examples 654
10.6.5.4 Status and error IDs 657
10.6.6 LBP_OpUnitComm [FB25000]: Operator component communication 657
10.6.7 LBP_OpUnitWriteRecord [FB25002]: Writing a data record to MCP 659

10.7 Functions (FC) 662
10.7.1 LBP_ConfigBP [FC1]: Basic program, startup section 662
10.7.1.1 STRUCT "LBP_typeTMMagazines" magazine data 662
10.7.1.2 STRUCT "Magazines" magazine data 663
10.7.1.3 PLC machine data 669
10.7.2 LBP_MainBP [FC2]: Basic program, cyclic section 671
10.7.3 LBP_NCKProcessIRT [FC3]: Basic program, interrupt-driven section 672
10.7.4 LBP_TMTransferMT [FC6]: Transfer block for tool management and multitool 674
10.7.5 LBP_TMTransferTurret [FC7]: Acknowledge tool change with turret 676
10.7.6 LBP_TMTransfer [FC8]: Acknowledge tool change 678
10.7.7 LBP_ReqASUP [FC9]: Start asynchronous subprogram (ASUP) 687
10.7.8 LBP_GenerateAlarmMsgs [FC10]: error and operating messages 690
10.7.9 LBP_CallBackAuxFunc [FC12]: Call interface for users for auxiliary functions 696
10.7.10 LBP_HTCtrlHT2 [FC13]: Display control for HT 2 698
10.7.11 LBP_CtrlAxisSpindle [FC18]: Spindle control 702
10.7.12 LBP_MCPCtrlMilling [FC19]: Transfer of the MCP signals on the interface 712
10.7.13 LBP_TransferSelData [FC21]: Data exchange NC/PLC 720
10.7.13.1 Function 1, 2: Signals synchronized actions to / from Channel 721
10.7.13.2 Function 3, 4: Fast data exchange PLC-NC 722
10.7.13.3 Function 5: Update control signals to channel 727
10.7.13.4 Function 6: Update control signals to axes 728
10.7.13.5 Function 7: Update control signals to axes 728
10.7.14 LBP_MCPCtrlMillingSmall [FC24]: Transfer of the MCP signals on the interface 729
10.7.15 LBP_MCPCtrlTurning [FC25]: Transferring the MCP/OP signals to the interface 737
10.7.16 LBP_HTCtrlHT8 [FC26]: Transferring the HT 8 / HT 10 signals to the interface 744
10.7.17 LBP_Handwheels [FC25002]: transfer handwheel signals 751

10.8 General information on configuration and commissioning 751
10.8.1 Configuration machine control panel, handheld unit, direct keys 751
10.8.1.1 Coupling via PROFINET 752
10.8.1.2 Coupling via PROFIBUS 753
10.8.1.3 Connection via Industrial Ethernet 753
10.8.2 Identification of operator components 768

10.8.3 Switchover of machine control panel, handheld unit 770
10.9 Appendix 772
10.9.1 References 772
A Appendix 773
A.1 List of abbreviations 773
A.2 Documentation overview 778
Index 779

Introduction 1

1.1 About this documentation

This documentation is part of the documentation category of SINUMERIK Function Manuals.

SINUMERIK Function Manuals

The SINUMERIK Function Manuals describe the NC functions of a SINUMERIK control.

The target group are configuring engineers, technologists, commissioning engineers and programmers.

Each Function Manual covers a specific topic and contains all the function descriptions for this topic.

The following table shows which Function Manuals are available for your SINUMERIK control and which topics a Function Manual covers:

Function Manual	Topic
Basic functions	Basic functions of a CNC control
Axes and spindles	Axis and spindle functions, axis couplings
Transformations	Transformation functions
Monitoring and compensating	Axis monitoring and compensation functions, collision avoidance
Tools	Functions for selection, correction and monitoring of tools
Synchronized actions	Functionality of synchronized actions
PLC	Structure and functions of the PLC
Tool management	Function, commissioning and programming of tool management
Technologies (for SINUMERIK ONE and SINUMERIK 840D sl only)	Extended technological functionalities

Overview of contents

You will find an overview of the main chapters of the descriptions of functions that are contained in a Function Manual on the title page.

Validity

The title page also contains all data about the validity of a document, i.e. for which SINUMERIK control and for which software version this edition of the Function Manual is valid.

Fundamental safety instructions 2

2.1 General safety instructions

⚠ WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Malfunctions of the machine as a result of incorrect or changed parameter settings

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization against unauthorized access.
- Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

2.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

2.3 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to

an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit
https://www.siemens.com/industrialsecurity (https://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
https://www.siemens.com/industrialsecurity (https://new.siemens.com/global/en/products/services/cert.html#Subscriptions).

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ WARNING

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- On completion of commissioning, check all security-related settings.

Overview 3

3.1 Introduction

TIA Portal

In the TIA Portal, you create devices, create programs, and load them into your control. In order to be able to use SINUMERIK devices and software modules, you must install the SINUMERIK ONE STEP 7 Toolbox V16 in addition to the TIA Portal.

SINUMERIK ONE STEP 7 Toolbox V16

Extended functionality after installation:

- Supplement to the hardware catalog with the following SINUMERIK ONE modules:
  - NCU 1750
  - NCU 1760
  - PPU 1740
- The NX15.3 or NX10.3 modules that match the NCU 1750 / NCU 1760 sl can be found in the hardware catalog in the folder "Controllers > SINUMERIK 840D sl > NX".
  The firmware version of the NX module must match the firmware version of SINAMICS_Integrated and must be compatible with the firmware version of the NCU. The version compatibility can be found in the following table.
- SINUMERIK basic PLC program
- Creation of SINUMERIK PLC commissioning archives in DSF format
- Use of NC variables
- Import of SINUMERIK user alarm texts (alarm number range 500,000 ... 999,999)
- Export all alarms (e.g. system diagnostics) and text lists for SINUMERIK Operate
- Support of SINUMERIK Safety Integrated (F-PLC) for SINUMERIK ONE

Commissioning the PLC program

4

4.1 Overview of the PLC program

The PLC program of a SINUMERIK NCU is modular and consists of the following parts:

- Organization blocks (OB)
  The SINUMERIK PLC basic program must be opened and initialized in the respective organization blocks (OB) of the PLC program. You also call your PLC user program in these organization blocks.
- PLC basic program
  The PLC basic program organizes the exchange of signals and data between the PLC user program and the NCK.
- PLC user program
  The PLC user program is the user-specific part of the PLC program by which the PLC basic program has been augmented or extended.
- Safety program of the F-PLC (optional)
  If you want to use the safety functions of the F-PLC, add a safety program to the PLC program (see the Safety Integrated Commissioning Manual).

Note

The system library of the TIA Portal contains the SINUMERIK PLC basic program and data types as templates. You can edit the individual blocks of the PLC basic program directly in the TIA Portal using the STEP 7 editors.

4.2 Installation of the basic program

The PLC program is installed in the TIA Portal by performing the following steps:

- Create the PLC basic program and insert it into the project
- Edit the blocks
- Load the PLC program into the PLC

The SINUMERIK PLC basic program and data types in the form of templates are contained in the system library of the TIA Portal. You can copy the blocks that you need by dragging them into the program blocks.

You can edit the individual blocks of the PLC basic program directly in the TIA Portal using the STEP 7 editors.

Note

Installation / Update

The PLC of a SINUMERIK NCU is always installed in the TIA Portal.

After you start the TIA Portal and create a project, the functionalities of the SINUMERIK NCU are available in the TIA Portal.

Content

The source programs of the organization blocks, including standard parameterization, interface symbols and DB templates for the handheld unit and M decoding functions are included in the basic program.

4.3 Application of the basic program

A new basic program must be created in the TIA Portal project for each plant or machine.

The system library of the TIA Portal contains the SINUMERIK PLC basic program and data types in the form of templates.

Remark

The structures of a project and the procedure for creating projects and user programs are described in the TIA Portal documentation.

Procedure

The basic program blocks are copied in the TIA Portal via the system library. The copy templates are located in the "Libraries" task card under "Global Libraries".

The components of the PLC basic program are contained in the corresponding subfolders and structured as shown below:

Folder	Purpose	Content
Copy templates > SINUMERIK ONE PLC Basic Program > SINUMERIK ONE PLC BP	Use this "Copy templates" folder when you insert a new SINUMERIK ONE.	Contains all of the blocks that are needed for the maximum configuration (31 axes, 10 channels) of a SINUMERIK ONE. The individual blocks are sorted into the subfolders "DB", "FB", "FC" and "Internal".
Copy templates > SINUMERIK ONE PLC Basic Program (upgrade) > SINUMERIK ONE PLC BP	Use this copy template folder when you upgrade or migrate from an older version of the PLC basic program.	Contains all of the know-how-protected blocks that are needed for the maximum configuration (31 axes, 10 channels) of a SINUMERIK ONE. Blocks that are supplemented with user-specific adaptations are not included. The individual blocks are sorted into the subfolders "DB", "FB", "FC" and "Internal".
Copy templates > SINUMERIK ONE PLC Basic Program > SINUMERIK ONE PLC BP data types	The content of this copy template folder is used by the PLC basic program and must be copied into the PLC data types folder.	Contains all the user-defined data types (UDT) that are used by the PLC basic program.

You can copy the block copy templates and PLC data type copy templates individually or collectively within their folder structure by dragging and inserting them into your project.

4.4 Version codes

Basic program

The version of the basic program is displayed on the Version screen of the user interface along with the control system type.

The control system type is encoded as follows:

Leftjustified decade of DB17.DBD0 (byte 0)	Control system type
03	SINUMERIK NCU/PPU 17x0

User program version codes

Users can also display their own PLC version codes on the HMI version screen. For this purpose, a data of type STRING containing a maximum of **54** characters must be defined in any data block. However, the version is not interpreted. It is just taken over as a string. The parameterization on this string is done via a pointer on FB1. For this, the data block must be defined symbolically.

The version code can be formatted in the string as follows:

- xx.yy
- xx.yy.zz

- ww.xx.yy.zz
- vv.ww.xx.yy.zzz
- x.y
- x.y.z
- w.x.y.z
- v.w.x.y.z

In addition to the version identification, a date can be entered in the string, which with appropriate formatting, is displayed in the HMI in the version screen. However, the date is not interpreted. It is just taken over as a string. A combination with version identifications is possible. The following formats are currently supported:

- 00/00/0000
- 0000/00/00
- 00/00/00

Examples:

- "Test project version 01.02.03 01/01/2015"
- "1.2 2015/01/01 test project"
- "01/01/15 version 01.02 test project"

4.5 Machine program

The machine manufacturer creates the machine program using the library routines supplied with the basic program. The machine program contains the logic operations and sequences on the machine. The interface signals to the MCU are also controlled in this program. More complex communication functions with the NCU, e.g. read/write NCU data, tool management acknowledgments, etc. are activated and executed via the FC and FB blocks of the basic program.

The machine program can be created in various STEP 7 languages, e.g. STL, LAD, FBD, S7GRAPH, SCL. The complete machine program must be generated and compiled in the correct sequence.

This means that blocks that are called by other blocks must generally be compiled **before these** blocks.

If blocks that are called by other blocks are subsequently modified in the interface (VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR) as the program is developed, the call block and all blocks associated with it must be compiled again. This general procedure applies analogously to instance data blocks for FBs. If this sequence of operations is not observed, time stamp conflicts occur when the data is decompiled into STEP 7. Decompilation of the blocks can therefore not be ensured and, with the function "Status of block", unnecessary conflicts can also occur. It is also advisable to generate blocks that were created in Ladder Diagram or in single statements (incremental mode) in ASCII-STL with the STEP 7 editor.

4.6 Data backup

Storing of the project is required to completely restore the software with all global settings.

Note

Sensible decompilation is not possible without the associated project for the machine. Therefore store the STEP 7 project of the machine that is located in the PLC CPU. This is a great help for service and saves unnecessary time-consuming restoration of the original project.

If the STEP 7 project exists and has been created according to the instructions stated above, symbols can be processed in the PLC-CPU on this machine.

The source programs of all organization blocks and all instance data blocks should always be available.

4.7 PLC series startup, PLC archive

SINUMERIK archive

Once the blocks have been loaded into the PLC CPU, a SINUMERIK archive can be generated in the TIA Portal.

A SINUMERIK archive has the following uses:

- Direct image of the data of a commissioned PLC in a file for data backup on the machine.
- To simplify the series commissioning
- To commission the PLC with the SINUMERIK archive directly on the NCU without using a PG/PC, TIA Portal or STEP 7
- To transfer the data to the NCU without establishing an online connection to the actual hardware

To ensure data consistency, this backup must be created immediately after block loading when the PLC is in the Stop state.

The following data are contained in a SINUMERIK archive:

- Hardware data of the PLC
- Program blocks of the PLC Version for actual hardware
- Hardware data of the CP
- PLC alarms and text lists (PLCAlarms.xml, PLCTextLists.xml)

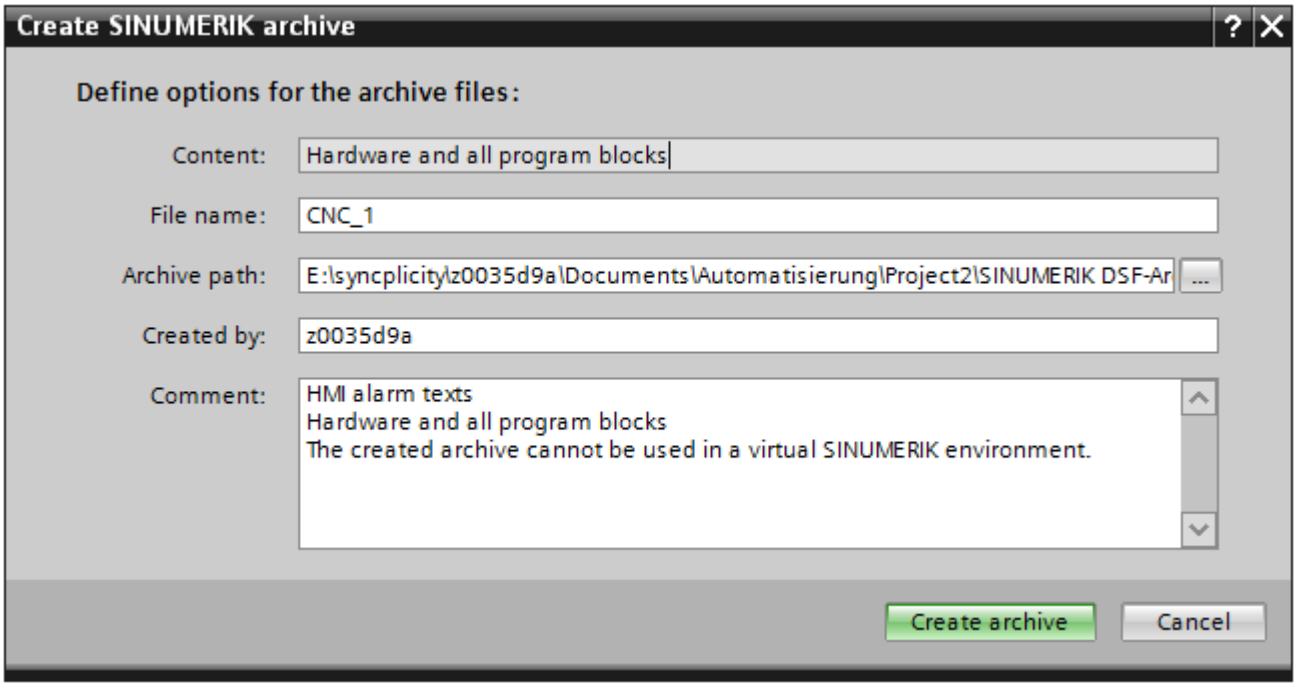
SINUMERIK PLC commissioning archives (*.dsf) created with the TIA Portal only contain PLC data.

Note

Alternatively, a SINUMERIK archive can be generated with the commissioning tool.

Creating a SINUMERIK archive in the TIA Portal

1. In the project tree, right-click the device name, e.g. "NCU_1".
2. Select "Create SINUMERIK archive > Hardware and all program blocks" from the shortcut menu.
   The "Create SINUMERIK archive" dialog box opens.
3. Make the necessary settings, e.g. file name and archive path.



4. Click "Create archive".
   The SINUMERIK archive is created and stored in the specified path.

4.8 I/O modules (CP modules)

Additional packages for STEP 7 are generally required for more complex I/O modules (CP modules). Support blocks (FC/FB) are provided in these additional packets. The blocks contain specific functions for operating the relevant module. These functions can be parameterized and called in the user program.

Identical numbers

If handling and basic program blocks have identical numbers, the block numbers of the basic program must remain unchanged. The block numbers of the handling blocks must be renamed to free numbers via STEP 7.

Handling of blocks and libraries

5

5.1 Use and handling of groups

In the Program blocks folder in the project tree, you can optionally create groups to organize your program blocks.

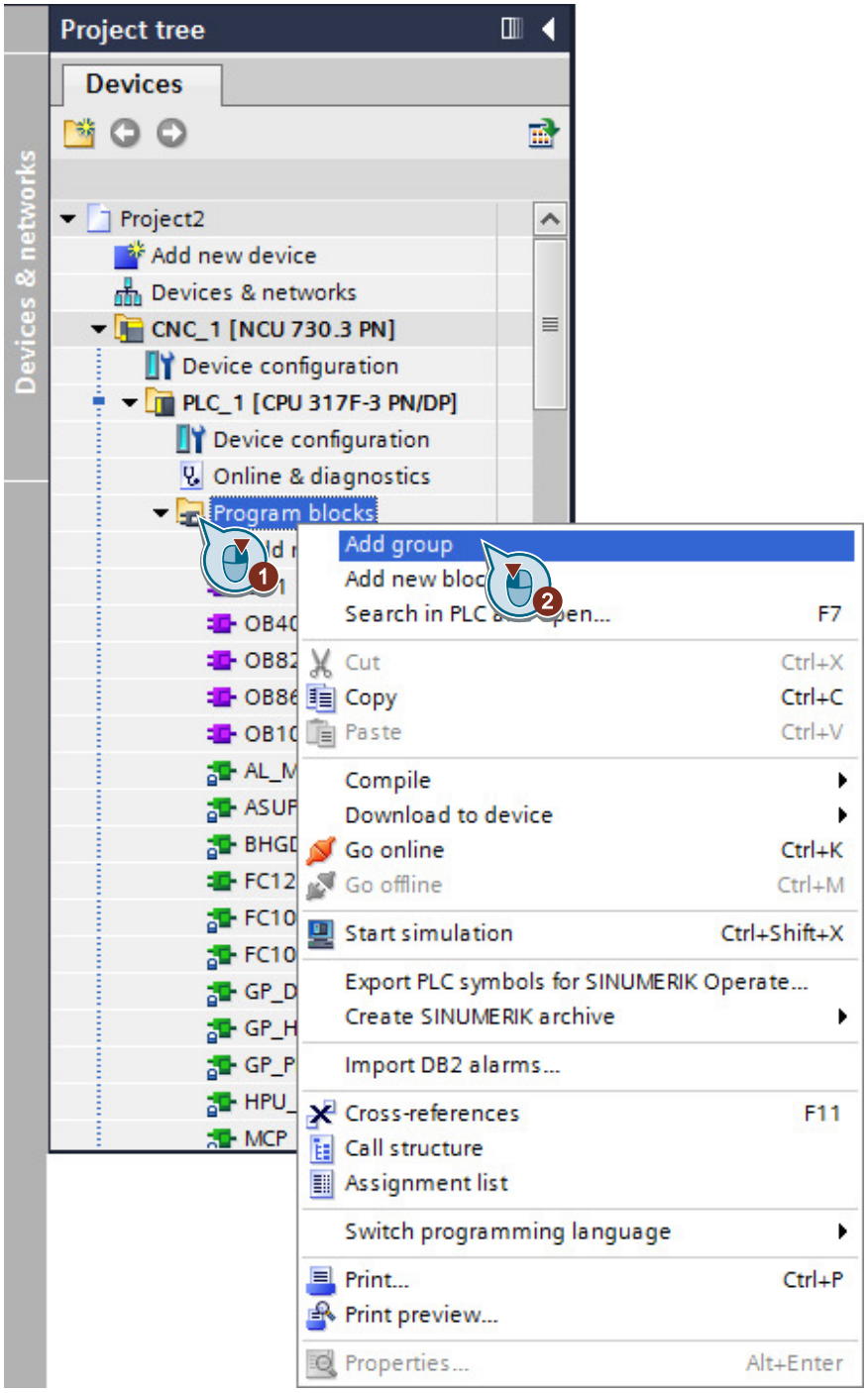


Figure 5-1 Add new group

For example, this function is useful in the following situations:

- Creating a dedicated group for the blocks of the basic program and of the user program. This allows you to individually display the editable blocks of the user program.
- Organizing specific related program blocks (e.g. axis DBs).
- Organizing specific program blocks in a group to save them quickly and easily in a reload archive.

To maximize the overview window, select the "Program blocks" folder and click the "Maximizes/ minimizes the Overview" icon on the project tree toolbar.

5.2 Copying blocks into a project

Here the procedure how to copy blocks from one project to another is described.

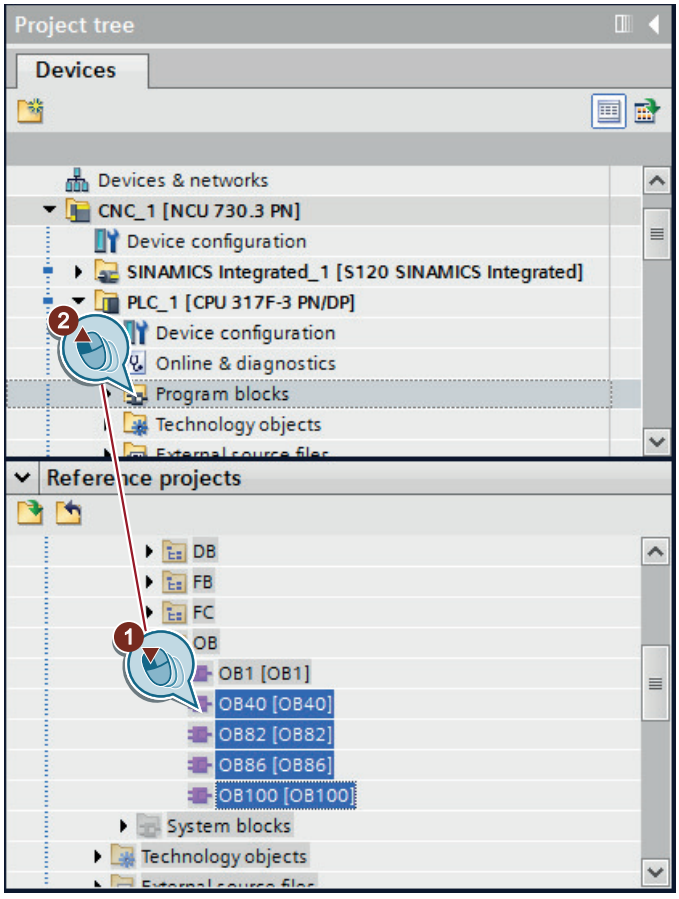
Alternatively, the library functionality in the TIA Portal can be used. Existing program modules can be provided, for example, in the form of copy templates or typified objects.

Procedure

Proceed as follows to import blocks from an existing project:

1. From the "View" menu, choose the command "Reference projects".
   The "Reference projects" palette is displayed under the project tree.
2. Click the "Open reference project" icon in the "Reference projects" palette, select the project, and confirm with "Open".
   The project is opened (write-protected) and the associated project structure is displayed in the "Reference projects" palette.

3. Navigate in the reference project to the "Program blocks" folder and select the blocks that you want to copy.
4. Drag-and-drop the blocks from the reference project to the "Program blocks" folder of your current project.



Note

Copy additional objects separately (e.g. variables or PLC data types)

When you copy the program blocks, objects that belong together, such as variables, PLC data types or technology objects, are not automatically included in the copy. This applies both to copying from reference projects and to copying to another PLC.

Error messages are issued during the compilation, e.g. if the variables used in the copied program block are not defined in the variable table.

Therefore, additionally copy the objects used by the copied blocks.

Note

Conflicts due to names, addresses or numbers that are used multiple times

When you paste objects from a reference project whose names have already been used in the project, these are automatically renamed under certain circumstances:

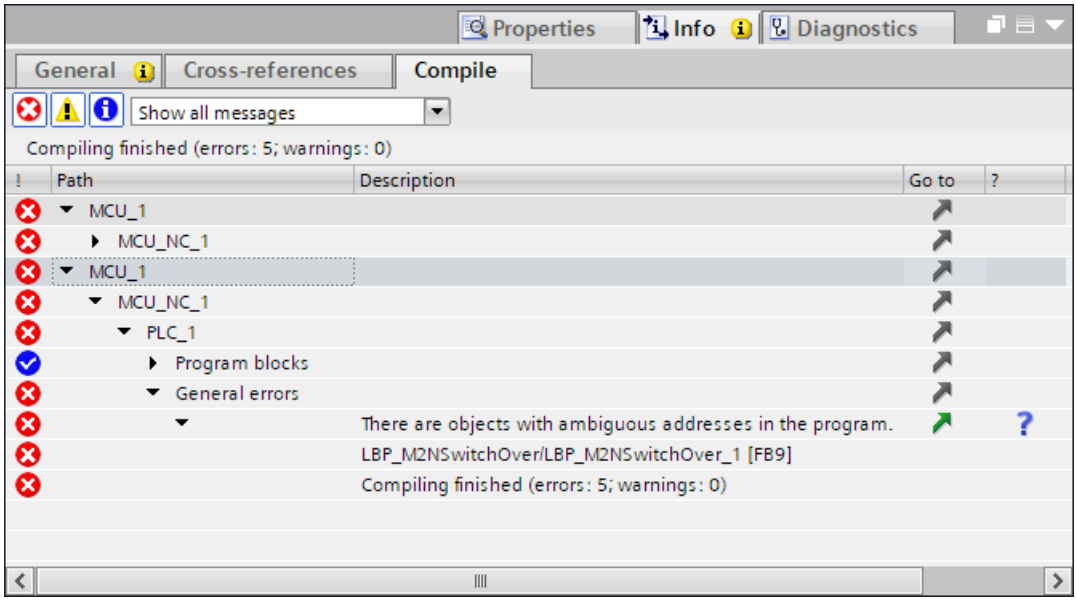
- If you copy objects with the same name into the same folder or the same group, the "Conflicts when copying" dialog (Page 32) is displayed.
- If you copy objects with previously assigned names to a different folder (or group), the newly inserted objects are renamed without asking.

A check for duplicate block numbers or addresses is, however, only performed during compilation.

5.3 Conflicts when copying blocks

Two different conflict types can occur during copying and pasting of program blocks:

- If several blocks with the same block numbers exist in the "Program blocks" folder, a message is not displayed immediately. As duplicate block numbers are not flagged until the during the compilation process,
  we recommend you compile your project after making a copy in order to detect any conflicts at an early stage.

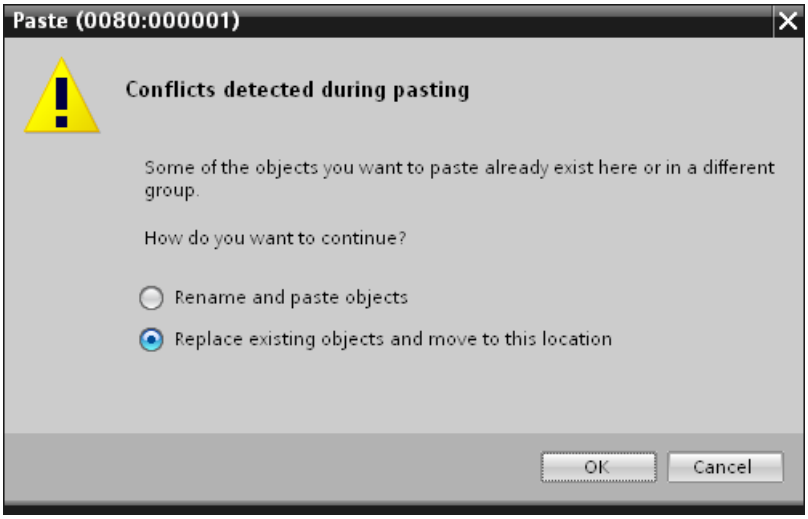


  If duplicate block numbers exist, proceed as follows:

  – If this involves one of the blocks of the basic program that you do not require (e.g. unused axis DB, channel DB, FB or FC), you can delete it.

  – If a user block has the same block number as a required basic program block, you must change the number of the user block.

- The presence of duplicate symbolic block names is already checked during the copying procedure. A message is issued if conflicts result because of existing symbolic names.



The dialog offers you the following ways of rectifying the conflict:

- You can cancel the copy process, search for the duplicate blocks in the project tree and delete them before you initiate the copy process again. In this case, click in the "Cancel" dialog box.
- You can copy the template to the project and overwrite duplicate blocks without knowing which blocks have caused the conflict. In this case, select in the dialog box "Replace existing objects and move to this location" and confirm with "OK".

The other option mentioned in the dialog, "Rename and add objects", does not resolve the conflict in this case.

5.4 Restrictions that apply to the user blocks in libraries

The blocks of the basic program library are not compatible with a project library. When a user block calls up blocks from the basic program library, the user block cannot be stored in a project library.

5.5 Block address range for PLC user program

From the following block number assignment overview, you can find out whether a specific block number (or block address) is assigned or reserved by the PLC basic program, or whether it can be used for the PLC user program.

Table 5-1 Assignment overview of the DB numbers

DB number	Availability	Occupied by
1	Reserved	Siemens
2 ... 3	Occupied	Siemens (basic PLC program)
4	Reserved	Siemens
5	Occupied	Siemens (basic PLC program)

DB number	Availability	Occupied by
6	Reserved	Siemens
7 ... 8	Occupied	Siemens (basic PLC program)
9	Reserved	Siemens
10 ... 11	Occupied	Siemens (basic PLC program)
12 ... 15	Reserved	Siemens
16	Occupied	Siemens (basic PLC program)
17 ... 18	Reserved	Siemens (basic PLC program)
19	Occupied	Siemens (basic PLC program)
20	Reserved	Siemens
21 ... 30	Occupied[1]	Siemens (PLC basic program: interface for NC channels)
31 ... 61	Occupied[1]	Siemens (PLC basic program: interface for axes/spindles)
62 ... 70	Freely available (user program)	-
71...73	Occupied[1]	Siemens (PLC basic program: tool management)
74	Reserved	Siemens
75 ... 76	Occupied	Siemens (M group decoding)
77 ... 80	Reserved	Siemens
81 ... 999	Occupied or reserved	Siemens (ShopMill, ManualTurn)
1000 ... 1002	Occupied	Siemens (basic PLC program)
1003 ... 1019	Reserved	Siemens
1020	Occupied	Siemens (basic PLC program)
1021 ... 1070	Reserved	Siemens
1071 ... 1073	Occupied	Siemens (basic PLC program)
1074 ... 1099	Reserved	Siemens
1100 ... 24999	Freely available (user program)	-
25000 ... 29999	Reserved	Siemens
30000 ... 59999	Reserved[2]	Standard number range for generated F-system blocks
60000 ... 60999	Occupied or reserved	CPU DBs

[1]Not recommended for use, but possible with limitations. Data blocks of channels, axes/spindles and tool management functions that have not been activated can be used by the user, but this can result in conflicts when upgrading or migrating the PLC basic program.

[2]If you use SINUMERIK Safety Integrated (F-PLC), this number range is used for automatically generated F-blocks by default. You can adapt the number range used by the F-system, but you must select a sufficiently large range of numbers.

Table 5-2 Assignment overview of the FB numbers

FB number	Availability	Occupied by
0 ... 1	Reserved	Siemens
2 ... 3	Occupied	Siemens (basic PLC program)
4	Reserved	Siemens

FB number	Availability	Occupied by
5	Occupied	Siemens (basic PLC program)
6	Reserved	Siemens
7	Occupied	Siemens (basic PLC program)
8	Reserved	Siemens
9	Occupied	Siemens (basic PLC program)
10 ... 29	Reserved	Siemens
30...999	Freely available (user program)	-
1000 ... 1023	Reserved	Siemens
1024 ... 24999	Freely available (user program)	-
25000 ... 32766	Reserved	Siemens (basic PLC program)
32767 ... 65535	Reserved[2]	Standard number range for generated F-system blocks

[2]If you use SINUMERIK Safety Integrated (F-PLC), this number range is used for automatically generated F-blocks by default. You can adapt the number range used by the F-system, but you must select a sufficiently large range of numbers.

Table 5-3 Assignment overview of the FC numbers

FC number	Availability	Occupied by
0	Reserved	Siemens
1 ... 3	Occupied	Siemens (basic PLC program)
4 ... 5	Reserved	Siemens
6 ... 10	Occupied	Siemens (basic PLC program)
11	Reserved	Siemens
12 ... 13	Occupied	Siemens (basic PLC program)
14 ... 16	Reserved	Siemens
17 ... 19	Occupied	Siemens (basic PLC program)
20	Reserved	Siemens
21	Occupied	Siemens (basic PLC program)
22 ... 23	Reserved	Siemens
24 ... 26	Occupied	Siemens (basic PLC program)
27 ... 29	Reserved	Siemens
30 ... 999	Freely available (user program)	-
1000 ... 1023	Reserved	Siemens
1024 ... 24999	Freely available (user program)	-
25000 ... 25001	Reserved	Siemens (basic PLC program)
25002	Occupied	Siemens (basic PLC program)
25003 ... 32766	Reserved	Siemens (basic PLC program)
32767 ... 65535	Reserved[2]	Standard number range for generated F-system blocks

[2]If you use SINUMERIK Safety Integrated (F-PLC), this number range is used for automatically generated F-blocks by default. You can adapt the number range used by the F-system, but you must select a sufficiently large range of numbers.

Inserting the PLC basic program into the project 6

6.1 Opening the PLC basic program system library

Procedure

Proceed as follows to open and display the system library of the SINUMERIK PLC basic program:

1. Switch to the task card "Libraries" and open the library "SINUMERIK ONE PLC Basic Program V6.x".

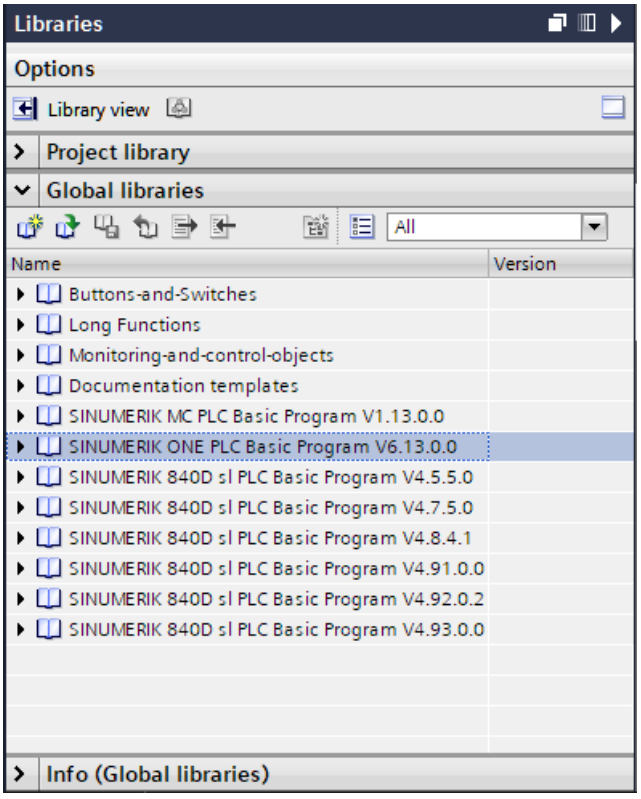


Figure 6-1 Opening the PLC basic program library

Result

The library opens. In the subfolders, you will find the components of the PLC basic program structured in the following way:

Lower-level folders	Purpose	Content
Copy templates > SINUMERIK ONE PLC Basic Program > SINUMERIK ONE PLC BP	Use this "Copy templates" folder when you insert a new SINUMERIK ONE.	Contains all of the blocks that are needed for the maximum configuration (31 axes, 10 channels) of a SINUMERIK ONE. The individual blocks are sorted into the subfolders "DB", "FB", "FC" and "Internal".
Copy templates > SINUMERIK ONE PLC Basic Program (upgrade) > SINUMERIK ONE PLC BP	Use this copy template folder when you upgrade or migrate from an older version of the PLC basic program.	Contains all of the know-how-protected blocks that are needed for the maximum configuration (31 axes, 10 channels) of a SINUMERIK ONE. Blocks that are supplemented with user-specific adaptations are not included. The individual blocks are sorted into the subfolders "DB", "FB", "FC" and "Internal".
Copy templates > SINUMERIK ONE PLC Basic Program > SINUMERIK ONE PLC BP data types	The content of this copy template folder is used by the PLC basic program and must be copied into the PLC data types folder.	Contains all the user-defined data types (UDT) that are used by the PLC basic program.

You can copy these block copy templates (Page 40) and PLC data types copy templates (Page 38) directly or together with the folder structure into your project.

6.2 Inserting PLC data types

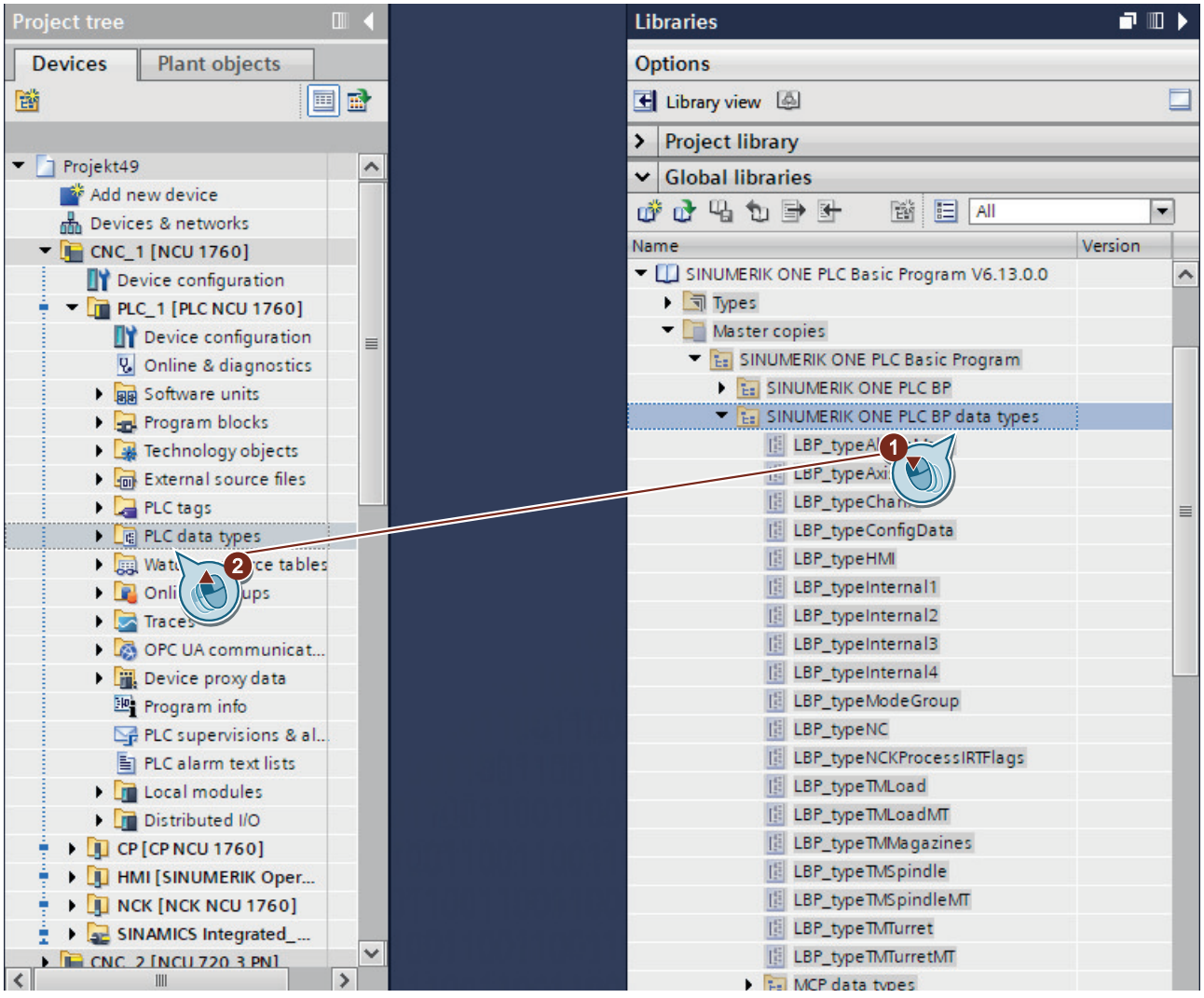
The PLC interface data blocks of the basic program make it possible to access various interfaces and data from the PLC program (axes/spindles, channels, mode groups, alarms/messages, tool management, PLC/HMI, MCP/HT).

PLC data types (UDT), which define the content of the data blocks, are available in the system library of the SINUMERIK ONE PLC basic program. Copy these PLC data types into your project.

Procedure

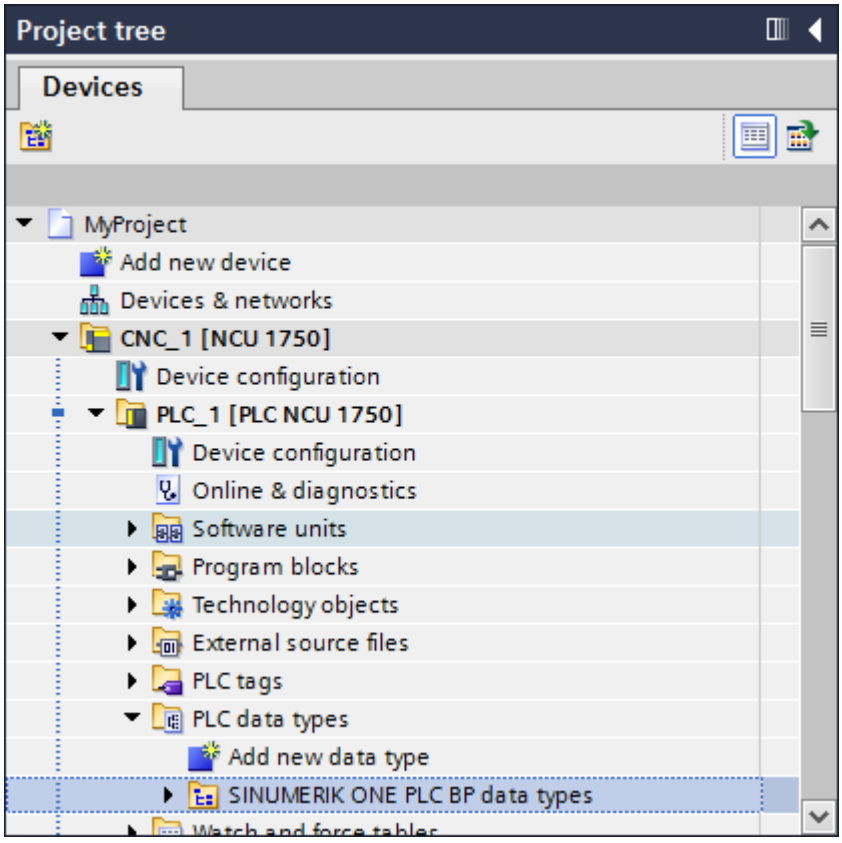
Proceed as follows to copy the PLC data types into your project:

1. Switch to the task card "Libraries" and open the system library "SINUMERIK ONE PLC Basic Program V6.x"
2. Drag and drop the folder "Master copies > SINUMERIK ONE PLC Basic Program > SINUMERIK ONE PLC BP data types" into the folder PLC data types of your project (e.g. under "CNC_1 > PLC_1 > PLC data types").



Result

The data types were copied into your project and assigned to the corresponding blocks of the PLC basic program.



Note

Deleting unused data types

Check the "PLC data types" folder, if applicable, for data types that are not used by the blocks. By inserting the SINUMERIK ONE PLC data types, for example, no (obsolete) data types of the SINUMERIK 840D sl are overwritten, since they have different names.

6.3 Inserting PLC basic program blocks

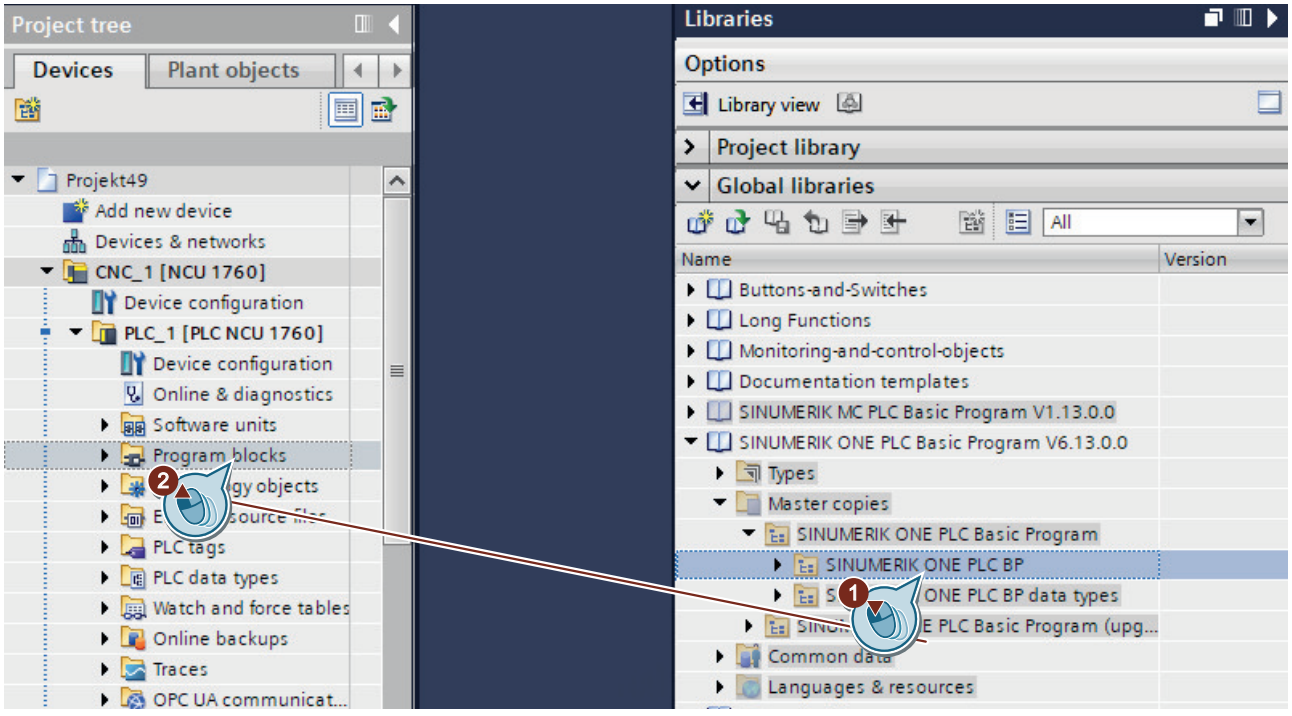
Requirement

- An NCU/PPU 17x0 is added.
- The data types of the PLC basic program blocks are inserted.

Procedure

To copy program blocks of the basic program from a copy template to the program blocks folder of the project tree, proceed as follows:

1. Switch to the task card "Libraries" and open the system library "SINUMERIK ONE PLC Basic Program V6.x"
2. You have several options for copying the PLC basic program blocks to your project:
   – Drag and drop the corresponding folder "SINUMERIK ONE PLC BP" into the folder Program blocks, e.g. under "CNC_1 > PLC_1 > Program blocks".



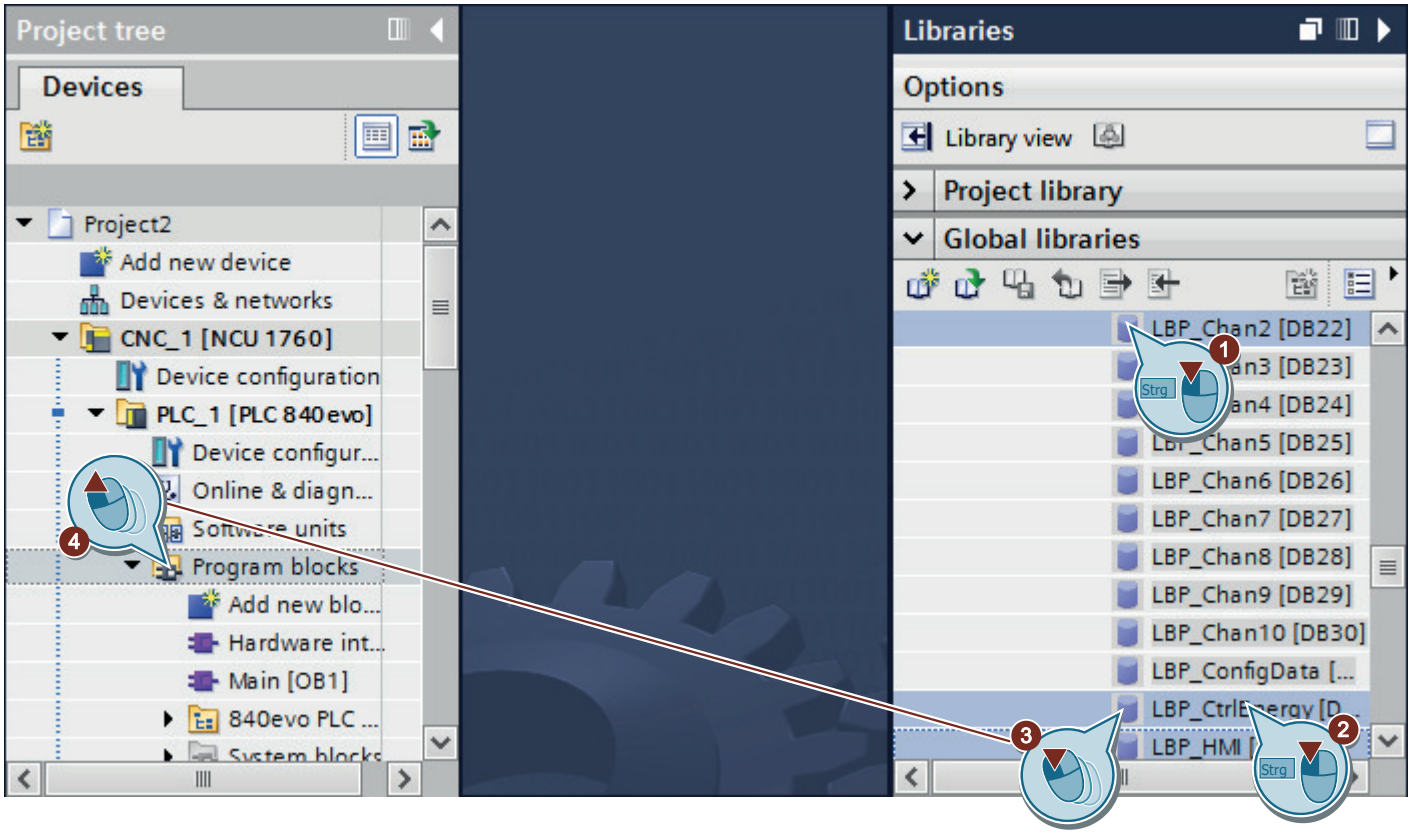
Note

Copying this folder changes the user-defined folder structure

Copying the "SINUMERIK ONE PLC BP" folder automatically creates new groups (subfolders) in your "Program Blocks" folder and moves the existing blocks there.

If you have already organized the PLC basic program into separate groups and would like to keep this structure, copy the individual block copy templates from the lower-level folders instead.

– Open the "Copy templates" folder and, if applicable, other lower-level folders and manually define the blocks which you would like to copy by selecting them and then dragging and dropping them into the "Program blocks" folder.
To select successive objects, click on the first object, press and hold the Shift key and then click on the last object.
To select non-successive objects, press and hold the Ctrl key and click on the objects one by one.

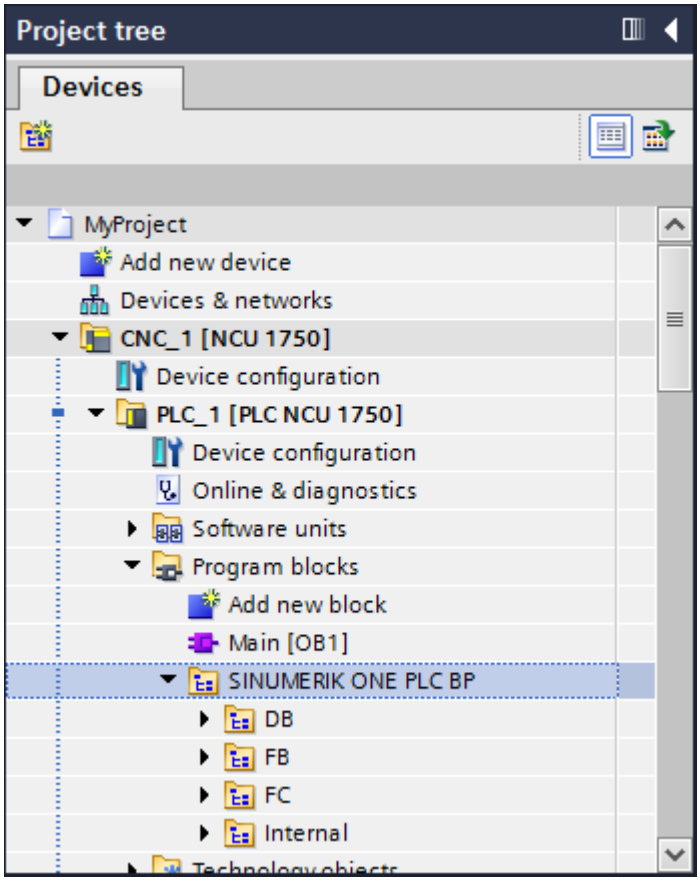


The blocks are copied into your project. A check is made for duplicate block names and, if applicable, the "Conflicts when copying" dialog is displayed.

3. In the project navigation, right-click "Program blocks" and select the "Software (compile all blocks)" command in the "Compile" shortcut menu.
The PLC program is compiled and a check is made for duplicate block addresses.

Result

The blocks of the basic SINUMERIK PLC program have been copied to your project. Any folder structure of the copy template was imported.



Now you can check the copied blocks and where necessary manually delete any PLC basic program blocks that you do not need (e.g. unused axis DB, channel DB, FB or FC).
If you have unused blocks in your project, some tasks take unnecessarily long, e.g. the loading or the creation and import of SINUMERIK archives.

6.4 Blocks with user-specific adaptations

In the following, you will find a list of all of the blocks of the SINUMERIK basic program that can contain user-specific adaptations and therefore cannot be automatically upgraded. These blocks are not included in the copy template "UpgradeProject".

Diagnostics [FB29] are no longer a direct component of the PLC basic program but are now available as an example application.

Table 6-1 PLC basic program blocks with user-specific adaptations

Block	Name	Description
DB5	LBP_ParamAlarmMsgs	Interface between user program and LBP_GenAlarmMsgs [FC10], contains the parameter assignment of the alarms for the bit fields in the LBP_AlarmMsgs [DB2] as error or operating message.
DB75	LBP_MFuncDecList-Config	Decoding list for M decoding according to list.
FC12	LBP_CallBackAuxFunc	Call interface for users for auxiliary functions, the block is called on an event-driven basis in the basic program when new auxiliary functions are available.

Specifying the execution structure and OBs of the PLC program

7

7.1 Overview of the PLC program

The PLC program of a SINUMERIK NCU is modular and consists of the following parts:

- Organization blocks (OB)
  The SINUMERIK PLC basic program must be opened and initialized in the respective organization blocks (OB) of the PLC program. You also call your PLC user program in these organization blocks.
- PLC basic program
  The PLC basic program organizes the exchange of signals and data between the PLC user program and the NCK.
- PLC user program
  The PLC user program is the user-specific part of the PLC program by which the PLC basic program has been augmented or extended.
- Safety program of the F-PLC (optional)
  If you want to use the safety functions of the F-PLC, add a safety program to the PLC program (see the Safety Integrated Commissioning Manual).

Note

The system library of the TIA Portal contains the SINUMERIK PLC basic program and data types as templates. You can edit the individual blocks of the PLC basic program directly in the TIA Portal using the STEP 7 editors.

7.2 Execution structure of the PLC program

Overview

The SINUMERIK ONE PLC basic program comprises various function blocks (FB), functions (FC) and data blocks (DB). These blocks are called by various parts of the PLC program and organize the exchange of signals and data between the PLC user program and the NCK, HMI, and machine control panel components. From a chronological viewpoint, the main part of the PLC basic program runs before the PLC user program, but must partly be called via the same organization blocks.

The following figure and its description illustrate the structure of the PLC program:

Note

Information on the PLC basic program blocks required in the project

The following description does not list individually all the PLC basic program blocks that are required for an executable project. Refer to the overview tables of the PLC basic program reference for this kind of information or for the dependencies between the individual PLC basic program blocks.

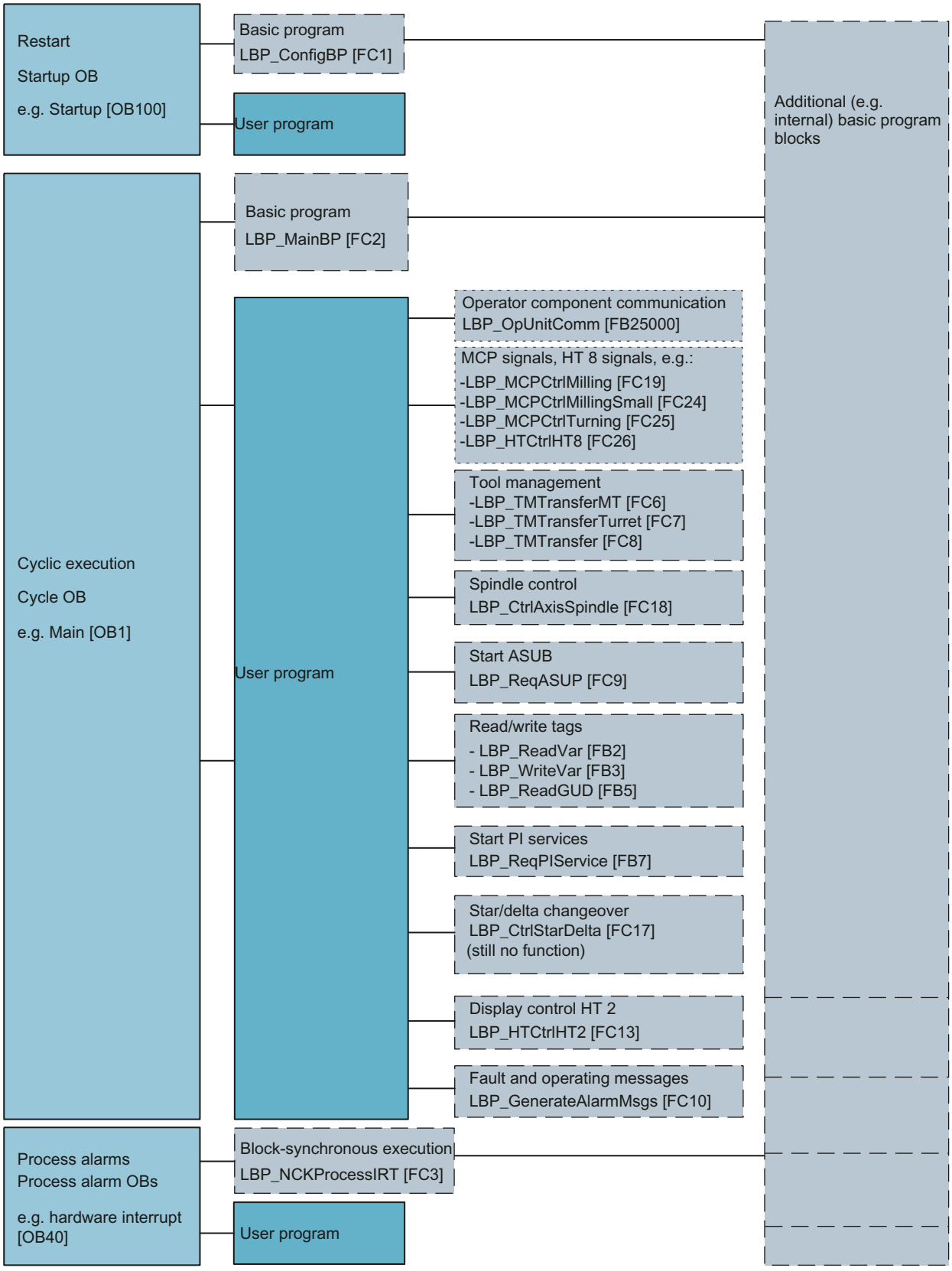

Restart
Startup OB
e.g. Startup [OB100]
Basic program
LBP_ConfigBP [FC1]
User program
Cyclic execution
Cycle OB
e.g. Main [OB1]
Basic program
LBP_MainBP [FC2]
User program
Operator component communication
LBP_OpUnitComm [FB25000]
MCP signals, HT 8 signals, e.g.:
-LBP_MCPCtrlMilling [FC19]
-LBP_MCPCtrlMillingSmall [FC24]
-LBP_MCPCtrlTurning [FC25]
-LBP_HTCtrlHT8 [FC26]
Tool management
-LBP_TMTransferMT [FC6]
-LBP_TMTransferTurret [FC7]
-LBP_TMTransfer [FC8]
Spindle control
LBP_CtrlAxisSpindle [FC18]
Start ASUB
LBP_ReqASUP [FC9]
Read/write tags
- LBP_ReadVar [FB2]
- LBP_WriteVar [FB3]
- LBP_ReadGUD [FB5]
Start PI services
LBP_ReqPIService [FB7]
Star/delta changeover
LBP_CtrlStarDelta [FC17]
(still no function)
Display control HT 2
LBP_HTCtrlHT2 [FC13]
Fault and operating messages
LBP_GenerateAlarmMsgs [FC10]
Process alarms
Process alarm OBs
e.g. hardware interrupt [OB40]
Block-synchronous execution
LBP_NCKProcessIRT [FC3]
User program
Additional (e.g. internal) basic program blocks

Figure 7-1 Structure of the PLC program

The PLC always starts up in RESTART mode, i.e. the PLC operating system runs the block **Startup [OB100]** after initialization:

- In Startup [OB100], call **LBP_ConfigBP [FC1]**, which configures and initializes the PLC basic program.
  - LBP_ConfigBP [FC1] calls further PLC basic program blocks internally.
- In addition, you need to insert the respective calls for your **PLC user program** in Startup [OB100]

Once Startup [OB100] has run, the PLC commences cyclic mode at the beginning of **Main [OB1]**:

- In cyclic mode, the PLC basic program is called with **LBP_MainBP [FC2]** in Main [OB1]. The NCK-PLC interface is now processed in full. A cyclic monitoring function is activated between PLC and NCK once control system run-up and the first OB cycle have been completed. LBP_MainBP [FC2] calls further PLC basic program blocks internally.
- In addition, you need to insert the respective calls for your **PLC user program** in Main [OB1]
  - To facilitate communication with the various operator components, for the real SINUMERIK ONE, first call block "LBP_OpUnitComm [FB25000]". This means that actual operator component data is available for subsequent program parts.
  - You configure the transfer of machine control panel signals (for example, MCP or HT8) in this part of your user program and call the required basic program block, for example **LBP_MCPCtrlMilling [FC19]**.
  - From the PLC user program, you can call various PLC basic program blocks yourself at any point, e.g. to start ASUB or PI services or for tool management.
- Configure the **LBP_GenerateAlarmMsgs [FC10]** so that the alarms that occur are entered in the alarm buffer of the PLC.

If a hardware interrupt is triggered during runtime due to a configured hardware event, cyclic program processing is interrupted and the corresponding hardware interrupt OB is triggered.

- If a command that requires a synchronous reaction in the PLC program is processed during runtime in an NC part program, this reaction is executed in the part of the basic program that is synchronous with the block (LBP_NCKProcessIRT [FC3]). You configure the signal exchange and the hardware interrupt required for it in the NCK properties and you call the **LBP_NCKProcessIRT [FC3]** in the corresponding hardware interrupt OB, for example, in **Hardware Interrupt [OB40]**. LBP_NCKProcessIRT [FC3] calls further PLC basic program blocks internally.
- In addition, you insert the respective calls for your **PLC user program** in this and/or in different hardware interrupt OBs and define the associated events in the properties of the corresponding device or module.

A PLC failure produces the "2000 Sign-of-life monitoring PLC" alarm in SINUMERIK Operate. No return is made to the interruption point (for example, in the event of a power failure). Some data of the PLC basic program are stored in a retentive memory area.

7.3 Relevant organization blocks

Below, you will find a list of all of the organization blocks from which the basic SINUMERIK PLC program must be called.

The organization blocks are not available as a copy template in the system library because you can also insert calls of your user program in the OBs.

Instead, add these organization blocks yourself and insert call and initialization of the basic program at the beginning of each OB.

Table 7-1 Organization blocks of the basic SINUMERIK PLC program

Name (default setting)	**Address (default setting)**	**Description**	**PLC basic program block to be called**
Startup	OB100	Startup OBs are processed once when the mode of the CPU switches from STOP to RUN. After execution of the startup OB, execution of the cycle OB starts.	LBP_ConfigBP [FC1]
Main	OB1	Cycle OBs are higher-level logic blocks in the program that are cyclically processed and in which you can program instructions or call additional blocks.	LBP_MainBP [FC2]
Hardware interrupt	OB40	Hardware interrupt OBs interrupt the cyclic program processing due to a hardware event. You define the event in the hardware properties.	LBP_NCKProcessIRT [FC3]

7.4 Inserting program code for calling the basic program

Because the organization blocks of the PLC program must contain both calls of the SINUMERIK PLC basic program and calls of your PLC user program, these organization blocks are not available as copy templates in the system library.

Instead, add these organizational blocks yourself and then insert the calls of the SINUMERIK PLC basic program.

For example, you can type these calls directly in the editor, copy them from a program example (SCL) and paste them into the editor, or create them with drag and drop as described below.

Requirement

- The respective organization block is added.

  Note

  Selecting the programming language for Main [OB1]

  A newly added NCU already contains Main [OB1] in the LAD programming language. You can change the programming language to FBD in the block properties or delete the existing block and add it back again as an STL or SCL block.

- The basic SINUMERIK PLC program blocks that are to be called from the respective organization block are added in the "Program blocks" folder of the project tree.

Procedure

To insert the program code for calling the SINUMERIK PLC basic program with drag and drop, proceed as follows:

1. In the project tree, double-click the relevant organization block, e.g. "CNC_1 > PLC_1 > Program blocks > OB > Startup [OB100]".
   The organization block is opened.
2. Drag the basic program block to be called from the program blocks folder and drop it in Network 1 of the relevant organization block.
   – The startup OB, e.g. Startup [OB100], must include the call of LBP_ConfigBP [FC1] in network 1.

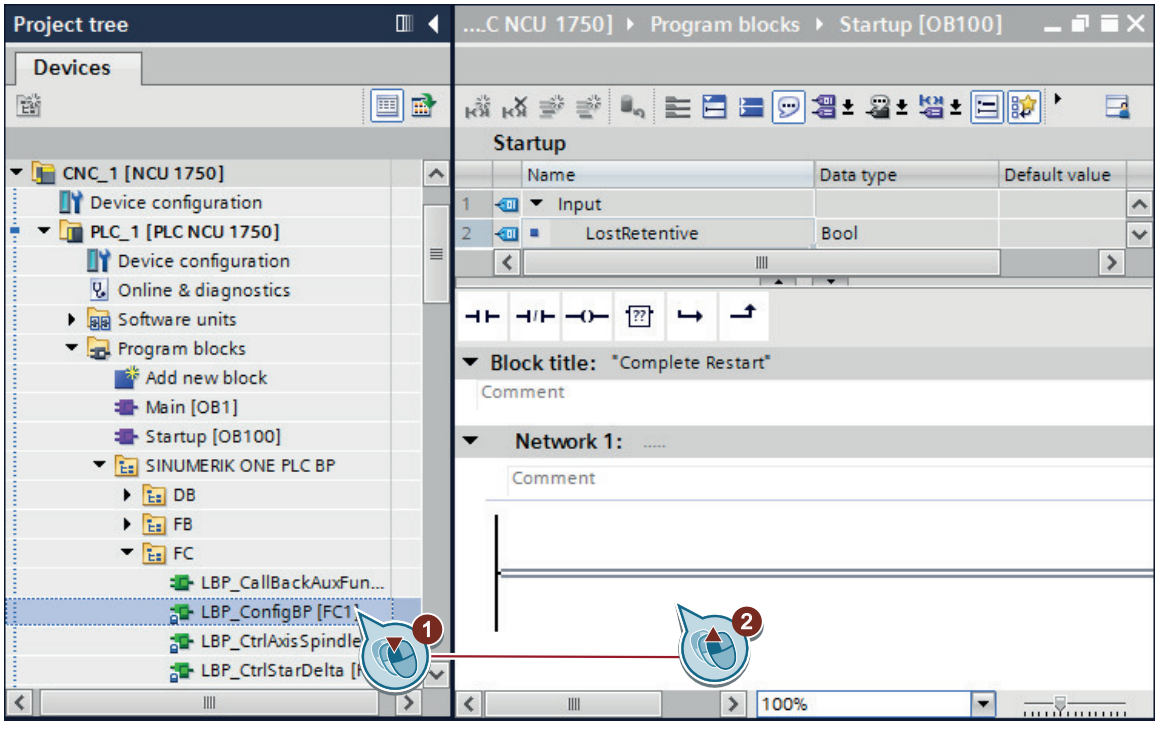


Figure 7-2 Inserting basic PLC program call (example of LBP_ConfigBP [FC1] in Startup [OB100] in LAD programming language)

– The cycle OB, e.g. Main [OB1], must include the call of LBP_MainBP [FC2] in network 1.

– The hardware interrupt OB, e.g. Hardware interrupt [OB40], must include the call of LBP_NCKProcessIRT [FC3] in network 1.

Result

The calls of the SINUMERIK PLC basic program have been added. For the hardware interrupt OB or the call of LBP_NCKProcessIRT [FC3], the event-driven signal exchange NCK to PLC must be configured in addition.

7.5 Programming examples (SCL)

7.5.1 Example for calling LBP_ConfigBP [FC1]

In the startup [OB100], LBP_ConfigBP [FC1] must be called in which the basic PLC program is initialized and configured.

You can copy or input the call in the editor or create it with drag and drop.

SCL

```
"LBP_ConfigBP"(MCPNum := 1,
MCP1In := P#I0.0,
MCP1Out := P#Q0.0,
MCP1BusAdr := 6,
MCP1Timeout := S5T#700MS,
MCP2In := NULL,
MCP2Out := NULL,
MCP2BusAdr := 0,
MCP2Timeout := S5T#700MS,
MCP1Stop := FALSE,
MCP2Stop := FALSE,
MCP1NotSend := FALSE,
MCP2NotSend := FALSE,
MCPBusType := B#16#6,
HTIf := 0,
HTIn := NULL,
HTOut := NULL,
HTAdr := 2,
HTStop := FALSE,
HTNotSend := FALSE,
NCCyclTimeout := S5T#200MS,
NCRunupTimeout := S5T#50S,
ListMDecGrp := 0,
MMCToIF := TRUE,
HWheelMMC := TRUE,
MCP_IF_TCS := FALSE,
ExtendChanAxMsg := FALSE,
HMIAxisSel := FALSE,
MsgUser := 10,
IRAuxfuT := FALSE,
IRAuxfuH := FALSE,
IRAuxfuE := FALSE,
OpKeyNum := 0,
Op1KeyIn := NULL,
Op1KeyOut := NULL,
```

SCL

```
Op1KeyBusAdr := 0,
Op2KeyIn := NULL,
Op2KeyOut := NULL,
Op2KeyBusAdr := 0,
Op1KeyStop := FALSE,
Op2KeyStop := FALSE,
Op1KeyNotSend := FALSE,
Op2KeyNotSend := FALSE,
OpUnitCommId :=1,
OpUnitInterfaceId :=262,
MCP1ConnectionId :=1001,
MCP2ConnectionId :=1002,
HTConnectionId :=1003,
Op1KeyConnectionId :=1004,
Op2KeyConnectionId :=1005,
IdentConnectionId :=1006,
MCP1LocalUdpPort :=16001,
MCP2LocalUdpPort :=16002,
HTLocalUdpPort :=16003,
Op1KeyLocalUdpPort :=16004,
Op2KeyLocalUdpPort :=16005,
IdentLocalUdpPort :=16006,
MaxModeGroup => "LBP_ConfigData".MaxModeGroup,
MaxChan => "LBP_ConfigData".MaxChan,
MaxAxis => "LBP_ConfigData".MaxAxis,
ActiveChan => "LBP_ConfigData".ActiveChan,
ActiveAxis => "LBP_ConfigData".ActiveAxis,
MaxNumUserDataInt => "LBP_ConfigData".MaxNumUserDataInt,
MaxNumUserDataHex => "LBP_ConfigData".MaxNumUserDataHex,
MaxNumUserDataReal => "LBP_ConfigData".MaxNumUserDataReal,
UserDataIntArray => NULL,
UserDataHexArray => NULL,
UserDataRealArray => NULL,
TMMagazines => NULL);
//INSERT USER PROGRAM HERE
```

7.5.2 Examples for calling LBP_MainBP [FC2]

Call LBP_MainBP [FC2] in Main [OB1], thus starting cyclic processing of the SINUMERIK PLC basic program.

You can copy or enter the call in the editor or create it with drag and drop.

SCL

```
// Call of basic program as first FC
"LBP_MainBP"();
// Add user program here
```

7.5.3 Example for calling LBP_NCKProcessIRT [FC3]

If a command that requires a synchronous reaction in the PLC program is processed during runtime in an NC part program, this reaction is executed in the part of the basic program that is synchronous with the block (LBP_NCKProcessIRT [FC3]). You configure the signal exchange and the hardware interrupt required for it in the NCK properties. You call LBP_NCKProcessIRT [FC3] in the corresponding hardware interrupt OB, e.g. in Hardware Interrupt [OB40].

You can copy this call to the editor or enter it or create it with drag and drop.

Example 1 without transfer of the output parameters

SCL

```
//Call without output parameters
"LBP_NCKProcessIRT"(Flags=>NULL);
```

Example 2 with transfer of the output parameters

In this example, "LBP_NCKProcessIRT" is called directly in the hardware interrupt OB with the transfer of the output flags. The output flags are subsequently evaluated.

Variables to be declared in the "Temp" section of the hardware interrupt OBs

Temp	
GP	"LBP_typeNCKProcessIRTFlags"
NckEvent	Bool
Axis1InPos	Bool
Axis2InPos	Bool
Axis3InPos	Bool
AuxFunctionInChan1	Bool
AuxFunctionInChan2	Bool
TMCommand	Bool

Program code to be inserted into the hardware interrupt OB

SCL

```
// GP: "LBP_typeNCKProcessIRTFlags";
// NckEvent : Bool;
// Axis1InPos: Bool;
// Axis2InPos: Bool;
// Axis3InPos: Bool;
// AuxFunctionInChan1: Bool;
```

SCL

```
// AuxFunctionInChan2: Bool;
// TMCommand: Bool;
//Call with output parameters
"LBP_NCKProcessIRT"(Flags=>#GP);
//Use of the output parameters
IF #GP.IRFromNck THEN
#NckEvent := TRUE;
END_IF;
//Use of the flags for axis 1 to 3
IF #GP.InPosition[1] THEN
#Axis1InPos := TRUE;
END_IF;
IF #GP.InPosition[2] THEN
#Axis2InPos := TRUE;
END_IF;
IF #GP.InPosition[3] THEN
#Axis3InPos := TRUE;
END_IF;
//Use of the flags for channel 1 and 2
IF #GP.AuxFunction[1] THEN
#AuxFunctionInChan1 := TRUE;
END_IF;
IF #GP.AuxFunction[2] THEN
#AuxFunctionInChan2 := TRUE;
END_IF;
//Is a tool change pending?
IF #GP.TM THEN
#TMCommand := TRUE;
END_IF;
```

Example 3 - as nested call without transfer of the output parameters

In this example "LBP_NCKProcessIRT [FC3]" is called nested over an own FC (e.g. hardware interrupt [OB40] > MyProcessAlarmFC [FC30] > LBP_NCKProcessIRT [FC3]) without transfer of the output flags.

Hardware interrupt OB, e.g. Hardware interrupt [OB40]:

SCL

```
//Call own FCs
"MyProcessAlarmFC"();
```

Own FC, e.g. MyProcessAlarmFC [FC30]:

SCL

```
//Call without output parameters
```

SCL

```
"LBP_NCKProcessIRT"(Flags=>NULL);
```

7.6 Retentivity behavior of the PLC basic program

The following data of the basic PLC program data is stored in a retentive memory area to avoid data loss in the event of a power failure.

Name	Block address or number
LBP_AlarmMsgs	DB2
LBP_ParamAlarmMsgs	DB5
LBP_ConfigData	DB7
LBP_InternalDB8	DB8
LBP_NC	DB10
LBP_ModeGroup	DB11
LBP_HMI	DB19
LBP_Chan1 ... LBP_Chan4	DB21 ... DB24
LBP_Axis1 ... LBP_Axis8	DB31 ... DB38
LBP_TMLoad	DB71
LBP_TMSpindle	DB72
LBP_TMTurret	DB73
LBP_MFuncDecListConfig	DB75
LBP_CtrlEnergy	DB1000
LBP_SentronPac	DB1001
LBP_SpindleTempSensor	DB1002
LBP_TMLoadMT	DB1071
LBP_TMSpindleMT	DB1072
LBP_TMTurretMT	DB1073

NC process alarms

8

8.1 General information on hardware interrupt OBs

If a hardware interrupt is triggered by a module during runtime, the hardware interrupt OB defined in the associated event is triggered or entered in the queue depending on its priority.

- This allows you to react specifically in the PLC user program to events that are triggered by various modules.
  You can use the same or different hardware interrupt OBs for different hardware/module events. When you assign the same hardware interrupt OB to different events, you must distinguish between the triggering events in the program code of the hardware interrupt OB.
- The hardware interrupt mechanism is also used by the event-driven signal exchange (NCK to PLC) to be able to respond to NCK events in the PLC program.
  You must activate the event-driven signal exchange in the NCK properties. You should assign an exclusive hardware interrupt OB during this. The hardware interrupt OB must contain the call of LBP_NCKProcessIRT [FC3] and the queries for the occurrence of the NCK events.

See also

Transfer and acknowledgment time of NCK signals (Page 57)

8.2 Transfer and acknowledgment time of NCK signals

Various events in the NCK are transferred to the PLC. These are accessible at different times in the PLC user program, depending on their type, and are acknowledged at different times to the NCK again. You can reconfigure this behavior for auxiliary functions as described in the table.

When you activate hardware interrupts in the properties of the NCK and assign a hardware interrupt OB, this assigned hardware interrupt OB is called for all NCK events. When multiple events occur at the same time, a separate instance of the same hardware interrupt OB is executed for each event.

All NCK signals that are accessible in the PLC user program are listed in the table below.

Signal type	Handover to user	Acknowledgment to NCK	Remark
Standard auxiliary function	Cycle OB	After cycle OB	
Fast auxiliary function (user-defined auxiliary functions with fast acknowledgment)	Cycle OB	Immediately in the hardware interrupt OB (see note)	Fast auxiliary functions (user-defined auxiliary functions with fast acknowledgment) are auxiliary functions that are already acknowledged in the hardware interrupt OB in the NC before they are pending for processing in the cycle (cycle OB). This means the acknowledgment does not acknowledge the execution of the PLC user function in this case. Especially when fast auxiliary functions are output in consecutive part program blocks there is no guarantee that the reaction in the PLC user program is synchronous with the block. Acknowledgment is delayed in such cases until after the cycle OB.
Auxiliary function to be executed event-driven	Hardware interrupt OB	Immediately in the hardware interrupt OB	An auxiliary function can be defined as an auxiliary function that is executed event-driven by setting the corresponding parameter of the LBP_ConfigBP [FC1]: • T function: IRAuxfuT • H function: IRAuxfuH • DL function: IRAuxfuE
G commands	Hardware interrupt OB	Immediately in the hardware interrupt OB	
Auxiliary function axis signals	Hardware interrupt OB	Immediately in the hardware interrupt OB	M, S, F assignment for MCU link axes
Signal from positioning axis	Next call of LBP_CtrlAxisSpindle [FC18]	Immediately in the hardware interrupt OB	
ASUP interrupt executed	Next call of LBP_ReqASUP [FC9]	Immediately in the hardware interrupt OB	Special type. No function but acknowledgment that ASUP interrupt was executed.
Compile cycles/data records	Hardware interrupt OB	Depending on programming in the compile cycle	Special type, is not evaluated.

8.3 Identification of an NCK event in the PLC program

A hardware interrupt OB should be assigned to the event-driven signal exchange from NCK to PLC that exclusively contains the call of LBP_NCKProcessIRT [FC3].

You will only need to distinguish between the triggering events in the program code of the hardware interrupt OBs if you did not follow this recommendation and have assigned the same hardware interrupt OBs to different events.

Whether or not an event was triggered by the NCK can be queried as follows:

- LADDR in the start information of the hardware interrupt OB
  LADDR contains the hardware identifier of the module that has triggered the hardware interrupt. The HW identifiers that occur in your project can be viewed in the system constants.
- PointAddr in the start information of the hardware interrupt OB

Slice access	#PointAddr.B3			#PointAddr.B2			#PointAddr.B1			#PointAddr.B0		
Bit number	31	...	24	23	...	16	15	...	8	7	...	0
Meaning	currently not used (set to 0)			currently not used (set to 1)			Channel currently not used (set to 0)			Event that triggered the hardware interrupt. 10 or B#16#0A: Event-driven signal exchange NCK to PLC		

LBP_NCKProcessIRT [FC3] should only be called after such a successful query to avoid unnecessary calling.

Alternatively, you can call the LBP_NCKProcessIRT [FC3] without any distinction at the start of the hardware interrupt OB and then specifically query the pending individual NCK signals in the output parameters of LBP_NCKProcessIRT [FC3].

8.4 Defining NCK hardware interrupt event

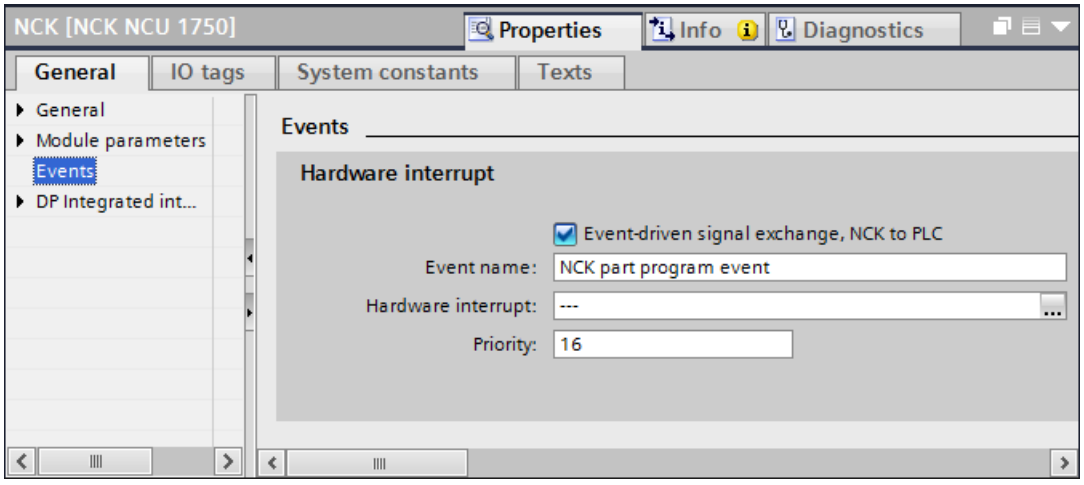
Requirement

- An NCU/PPU 17x0 is added.

Procedure

Proceed as follows to configure process alarms in the properties of the NCK:

1. Click on the NCK in the network view or device view.
2. In the "Properties" inspector window, select the "Events" entry.
   The hardware interrupt settings are displayed. The event-controlled signal exchange from the NCK to the PLC is activated by default.



3. Make the following settings:

Setting	Purpose
Event name	Name for identification of the event, for example, in the list of the start events of assigned hardware interrupt OBs and the system constants. A language-neutral event name is set as default. You can change the event name as necessary and select a language-neutral name, for example.
Hardware interrupt	Select an existing hardware interrupt OB from the list and confirm with the check mark button or click on "Add" to add a new block and assign it to the event. In special applications (e.g. modular machine) it may be useful to assign the hardware interrupt during runtime in the user program with an attach command instead.
Priority	Define the priority of this event in relation to other hardware interrupt OBs or the priority in the queue.

Result

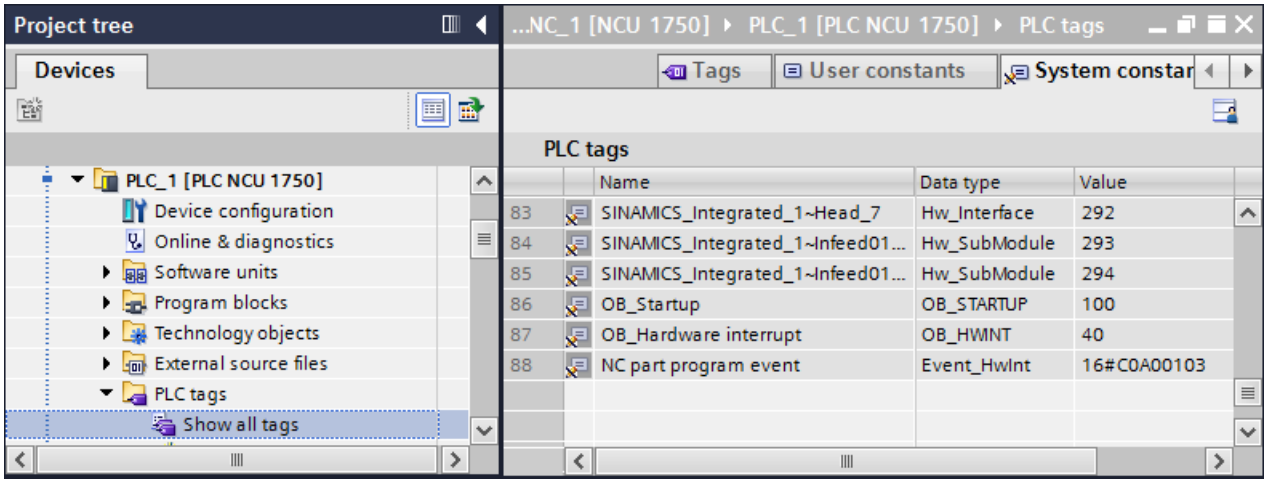
The properties of the NCK process alarm events have been specified and may have been assigned to a process alarm OB. The event is also listed in the properties of the selected hardware interrupt OB under "Start events". A corresponding system constant was added (Page 61).

For the NCK signals to be processed in the PLC basic program, you must call LBP_NCKProcessIRT [FC3] in the hardware interrupt OB (e.g. hardware interrupt [OB40]) next.

8.5 Use of the associated system constant

When you define the event for NCK hardware interrupts, a system constant is automatically added whose name corresponds to the associated event name. You can use the system constant to query in the PLC user program whether a hardware interrupt was triggered and by which module.

You can find the system constant in the "PLC tags" editor in the "System constants" tab.



NC variables

9

9.1 Overview of tag configuring

The following variable types are available to you in the TIA Portal when programming the SINUMERIK PLC:

- PLC variables
  The declaration and use of PLC variables corresponds to the default procedures in SIMATIC STEP 7.
- NC variables
  In your PLC program, you can use LBP_ReadVar [FB2] and LBP_WriteVar [FB3] to access NC variables that you added to an NC variable DB.

When working with variables in the TIA Portal, observe the different data type declaration for PLC variables and NC/GUD variables.

9.2 Corresponding data type declarations for NCK and PLC

The data types define the properties of the data, e.g. the display of the content and the permitted memory areas.

NC and GUD variables use different data type declarations than STEP 7 variables (PLC program). In the NC variable list, the data type is displayed in NCK declaration. When you insert an NC variable in your variable table and use it in the PLC program, you must use the PLC declaration.

Table 9-1 Assigning the data type declarations of NCK and STEP 7 (PLC)

Data type		Format	Value range	Brief description
NCK declaration	**PLC declaration**			
DOUBLE	REAL	Floating-point numbers according to IEEE754	-3.402823e+38 to -1.175495e-38 ±0.0 +1.175495e-38 to +3.402823e+38	Operands of the data type REAL have a length of 32 bits and are used for displaying floating point numbers.
FLOAT		Floating-point numbers		

Data type		Format	Value range	Brief description
NCK declaration	**PLC declaration**			
LONG	DINT	Integers with sign (decimal system)	-2147483648 to +2147483647	An operand of the data type DINT (Double INT) has a length of 32 bits and comprises two components: a sign and a numerical value in the two's complement.
		Binary numbers (only positive)	2#0 bis 2#0111111111111111111111111111111	
INTEGER		Octal numbers (only positive)	8#0 to 8#17777777777	
		Hexadecimal numbers	16#0 to 16#7FFFFFFF	
UINT_32	DWORD	Integers (decimal system)	-2147483648 to 4294967295	An operand of the data type DWORD is a bit string made up of 32 bits.
		Binary numbers	2#0 bis 2#11111111111111111111111111111111	
		Octal numbers	8#0 to 8#37777777777	
		Hexadecimal numbers	DW#16#00000000 to DW#16#FFFFFFFF, 16#00000000 to 16#FFFFFFFF	
		Decimal sequence	B#(0, 0, 0, 0) to B#(255, 255, 255, 255)	
INT_16	INT	Integers with sign (decimal system)	-32768 to 32767	An operand of the data type INT has a length of 16 bits and comprises two components: a sign and a numerical value in the two's complement.
		Binary numbers (only positive)	2#0 to 2#0111111111111111	
		Octal numbers	8#0 to 8#77777	
		Hexadecimal numbers (only positive)	16#0 to 16#7FFF	

Data type		Format	Value range	Brief description
NCK declaration	**PLC declaration**			
UINT_16	WORD	Integers (decimal system)	-32768 to 65535	An operand of the data type WORD is a bit string comprising 16 bits.
		Binary numbers	2#0 to 2#1111111111111111	
		Octal numbers	8#0 to 8#177777	
		Hexadecimal numbers	W#16#0 to W#16#FFFF, 16#0 to 16#FFFF	
		BCD	C#0 to C#999	
		Decimal sequence	B#(0, 0) to B#(255, 255)	
CHAR	CHAR	ASCII characters	ASCII character set	An operand of the data type CHAR has a length of 8 bits and occupies one BYTE in the memory.
	BYTE	Integers (decimal system)	-128 to +127 or 0 to +255	An operand of the data type BYTE is a bit string comprising 8 bits.
		Binary numbers	2#0 to 2#11111111	
		Octal numbers	8#0 to 8#377	
		Hexadecimal numbers	B#16#0 to B#16#FF, 16#0 to 16#FF	
STRING	STRING	ASCII character set including special characters	0 to 254 characters	An operand of the data type STRING stores several characters in a character string, which can consist of a maximum of 254 characters.
BOOL	BOOL	Boolean	FALSE or TRUE BOOL#0 or BOOL#1 BOOL#FALSE or BOOL#TRUE	An operand of the data type BOOL represents a bit value and contains one of the following values: • TRUE • FALSE
		Integers without sign (decimal system)	0 or 1	
		Binary numbers	2#0 or 2#1	
		Octal numbers	8#0 or 8#1	
		Hexadecimal numbers	16#0 or 16#1	
DATE-TIME	DT (DATE_AND_TIME)	Date and time (year-month-day-hour:minute:second:millisecond)	Min.: DT#1990-01-01-00:00:00.000 Max.: DT#2089-12-31-23:59:59.999	The data type DT (DATE_AND_TIME) stores date and time information in the BCD format.

Example

For example, to be able to read an NC variable of the type DOUBLE without adapting the format, an ANY pointer with REAL2 type must be specified in the destination area "RDx" (e.g.: P#M100.0 REAL2). If the basic program recognizes REAL2 as the target type when reading a variable of the DOUBLE type, the data is transferred to the PLC data area as a 64-bit floating-point number.

9.3 NC variable configuration in TIA Portal

9.3.1 Relevant objects and resources

To be able to access NC variables in your PLC user program, you must configure objects that are required for this in your project. Use the following resources in the TIA Portal for this:

- The system library with copy templates for the SINUMERIK PLC basic program 6.x contains the program blocks and PLC data types (UDT) that are required for access. These must be inserted in your project.
- Using the NC variable list, you can select NC variables, put them together in a user-defined way, configure them, and add them to an NC variable DB. Depending on the configuration step of an NC variable, it can differ conceptually.
- In the programming editor, create your user program with the desired NC variable access.

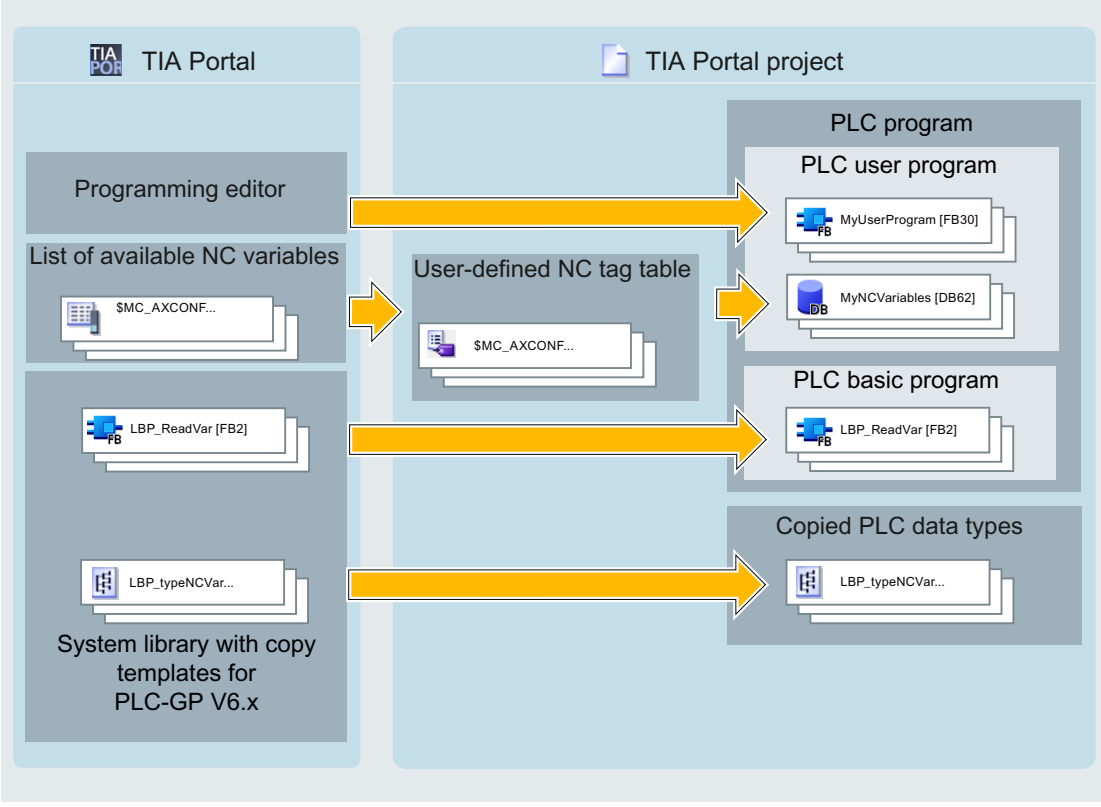
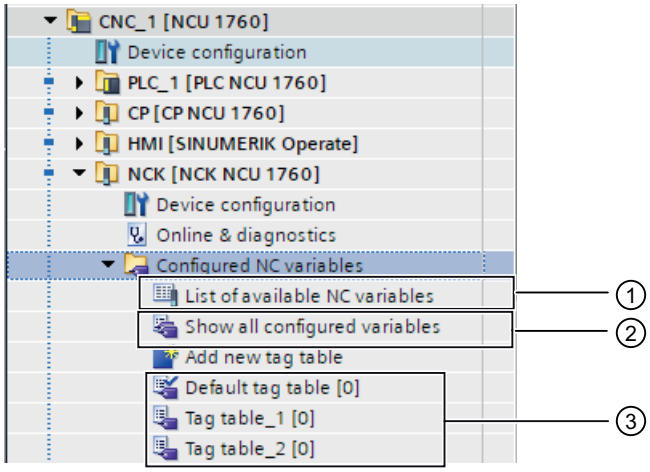


Figure 9-1 Configuring the objects required for NC variable access

For each NCU created in the project, there is the folder "Configured NC variables" under the NCK item in the project tree. This folder contains the various NC tag tables.

The following figure shows the tables contained in the "Configured NC variables" folder:



① Table "List of available NC variables"
② Table "All configured variables"
③ User-defined tag tables

Types of NC variables in the TIA Portal

Depending on the current configuration step and their properties, NC variables in the TIA Portal have different manifestations with different designations and properties at the respective time:

	Generic (available) NC variable	Configured NC variable	NC variable in DB/FB
Location	In the list of available NC variables	Stored in the user-defined tag table and in the list of available NC variables	In the DB or FB
Designation	Designations of the variable in the context of the NC, i.e. names of the associated NC variables and, if applicable, system variables.	User-defined name	User-defined name
Data type	Specified in NC declaration	Specified in PLC declaration	Linked with the PLC data types (UDT) of the PLC basic program
Configuration	No specific information entered.	Configuration of specific information available, e.g. about desired NCK object, channel or axis.	Specific token available for the purpose of accessing from the PLC program.
	-	-	Element for the variable value

9.3.2 Types of NC variables in the TIA Portal

Depending on the current configuration step and their properties, NC variables in the TIA Portal have different manifestations with different designations and properties at the respective time:

	Generic (available) NC variable	Configured NC variable	NC variable in DB/FB
Location	In the list of available NC variables	Saved in a user-defined variable table	Available in the DB or FB
Designation	Designations of the variable in the context of the NC, i.e. names of the associated NC variables and, if applicable, system variables.	User-defined name	User-defined name
Data type	Specified in NC declaration	Specified in PLC declaration	Linked with the PLC data types (UDT) of the PLC basic program
Configuration	No specific information entered.	Configuration of specific information available, e.g. about desired NCK object, channel or axis.	Specific token available for the purpose of accessing from the PLC program.
	-	-	Element for the variable value

9.3.3 Working with NC variable tables

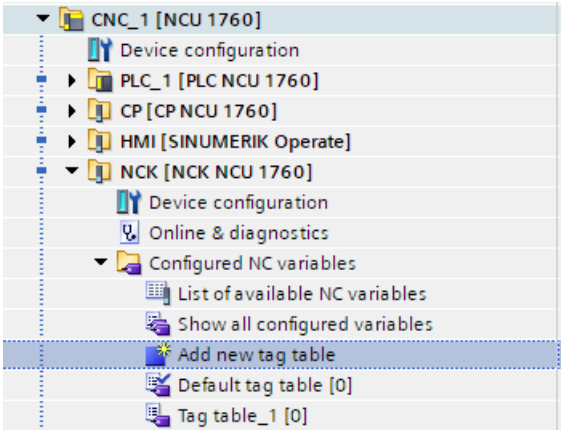
9.3.3.1 Creating and managing a user-defined variable table

Creating a tag table

To create a new NC tag table, proceed as follows:

1. In the project tree, open folder "Configured NC variables", e.g. under "CNC_1 > NCK > Configured NC variables".
2. Double-click the command "Add new tag table".

3. Enter a unique name for the new tag table.



4. Click "OK" to confirm your entries.

Opening the tag table

To open the NC tag table, proceed as follows:

1. In the project tree, open folder "Configured NC variables", e.g. under "CNC_1 > NCK > Configured NC variables".
2. Double-click on the NC tag table in the folder.

Copying a tag table

To copy NC tag tables, proceed as follows:

1. Select the NC tag table that you want to copy.
2. Select the "Copy" command in the shortcut menu.
3. Position the cursor at the position where you want to insert the NC tag table.
   You can insert the NC tag table in the same or in another NCU, in each case under the folder "Configured NC variables".
4. Select the "Paste" command in the shortcut menu.
   - If you insert the NC tag table into the same folder, the copy will be inserted with the name extension "_<consecutive number>".
   - If you insert the NC tag table into another folder that already contains a table with the same name, the table will be inserted with the name extension "_<consecutive number>".

Alternatively, you can also copy the NC tag table via drag & drop while pressing and holding the <Ctrl> key.

9.3.3.2 Managing entries in variable tables

Copying entries

Note

Compatibility check

If you copy variables into a variable table of another system, the compatibility is checked during inserting and, in the event of incompatibility, the copying is canceled.

To copy a variable, proceed as follows:

1. Select the variable table that you would like to copy.
   You can also select several variables by clicking on them while pressing and holding the <Ctrl> key or by clicking on the first and last variable while pressing and holding the <Shift> key.
2. Select the "Copy" command in the shortcut menu.
3. Position the cursor at the position where you want to insert the variable.
4. Select the "Insert" command in the shortcut menu.

Or

1. Configure the variable.
2. Press and hold the left mouse button.
3. At the same time, press the <Ctrl> key.
4. Drag the variable to the target.

Delete entries

To delete items, proceed as follows:

1. Select the line with the item that is to be deleted. You can also select several lines by clicking on them while pressing and holding the <Ctrl> key or by clicking on the first and last line while pressing and holding the <Shift> key.
2. Select "Delete" in the shortcut menu.

9.3.3.3 Adapt the view of the variable table

Sorting rows

To sort lines in a table, proceed as follows:

1. Select the column that you would like to sort by.
2. Click in the column header.
   The column is sorted in ascending order.
   An UP arrow shows the sorting order.
   - To change the sorting order, click on the arrow.
     The column is sorted in descending order. A DOWN arrow shows the sorting order.
   - To restore the original arrangement, click on the column header a third time.

Showing or hiding columns

To show or hide table columns, proceed as follows:

1. Click in a column header.
2. Select the command "Show/Hide" in the shortcut menu.
   The selection of the available columns is displayed.
   - To show a column, activate the check box for the column.
   - To hide a column, deactivate the check box for the column.
   - To show or hide several columns, click on "More" and check or uncheck the checkboxes of the corresponding columns in the "Show/Hide" dialog.

9.3.4 Working with the list of available NC variables

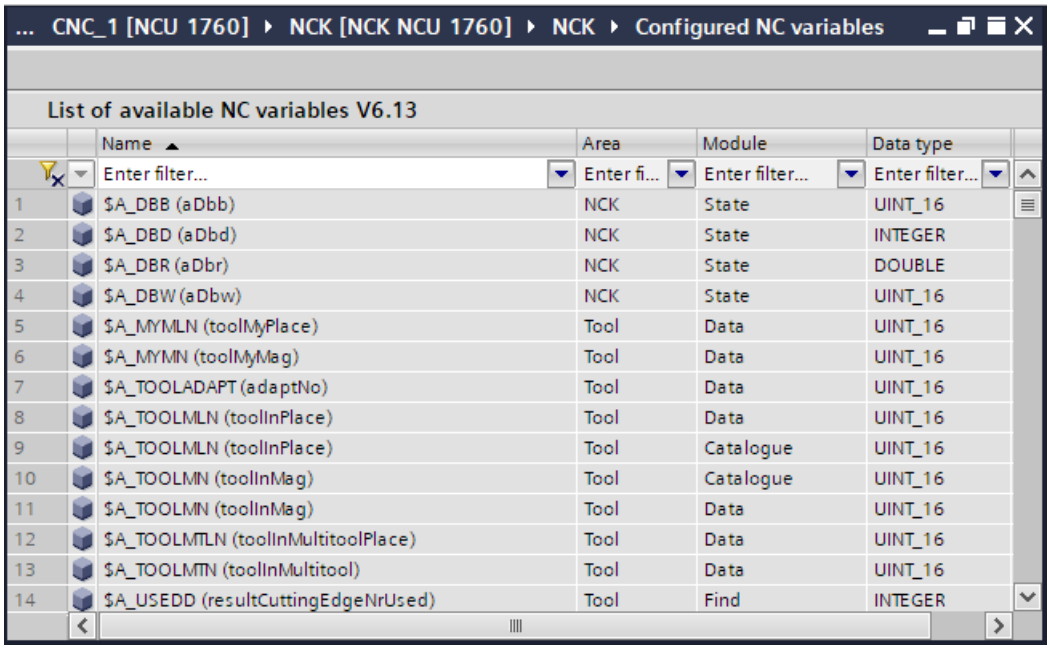
9.3.4.1 Searching and filtering NC variables

Under the NCK in the project tree, you will find the "List of available NC variables", which contains all of the NC variables that you can use with the configured NCU firmware.

All of the available variables are displayed in the NC variable list as per the default setting. You can filter and browse this complete list to then copy the desired variables to your tag table.

Filter

You can filter the NC variable list by various criteria to find the NC variables that are needed for your PLC program. To do this, click "Enter Filter" in the respective column.



The following filter options are available depending on the column type:

Filter criterion	Procedure
Name	Enter a user-defined filter text and confirm with the Enter key. Since this field may contain several names for a variable (system variable name or NC variable name), these are also taken into account during the search. Note that NC variables cannot always be unambiguously assigned via the NC variable name. Show the columns "Area" and "Module" as needed.
Area	Select the desired area in the dropdown list of the filter cell.
Module	Enter a user-defined filter text and confirm with the Enter key.
Description	Enter a user-defined filter text which appears in the descriptions to be filtered, then confirm with the Enter key. Using *, you can filter by several, non-consecutive words.
Data type	In the drop-down list of the filter cell, select a data type (NCK declaration) in order to filter by it.
Access rights	In the drop-down list of the filter cell, select an access rights type in order to filter by it: read only, read/write, or * (all).

You can also define several simultaneously effective filter criteria in the various columns.

Select the text, press the Del key, and confirm with the Enter key to delete a filter criterion.

Showing or hiding columns

To show or hide table columns, proceed as follows:

1. Click in a column header.
2. Select the command "Show/Hide" in the shortcut menu.
   The selection of the available columns is displayed.
   - To show a column, activate the check box for the column.
   - To hide a column, deactivate the check box for the column.
   - To show or hide several columns, click on "More" and select or clear the check boxes of the corresponding columns in the "Show/Hide" dialog.

Sorting rows

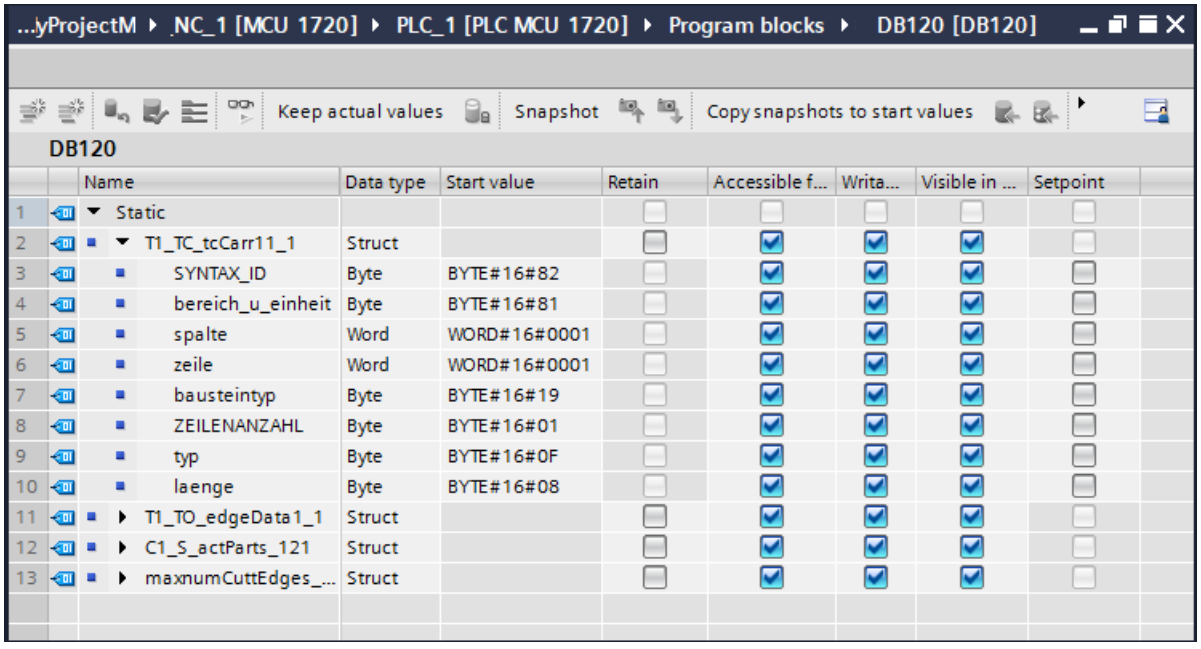
To sort lines in a table, proceed as follows:

1. Select the column that you would like to sort by.
2. Click in the column header.
   The column is sorted in ascending order.
   An UP arrow shows the sorting order.
3. To change the sorting order, click on the arrow.
   The column is sorted in descending order. A DOWN arrow shows the sorting order.
4. To restore the original arrangement, click on the column header a third time.

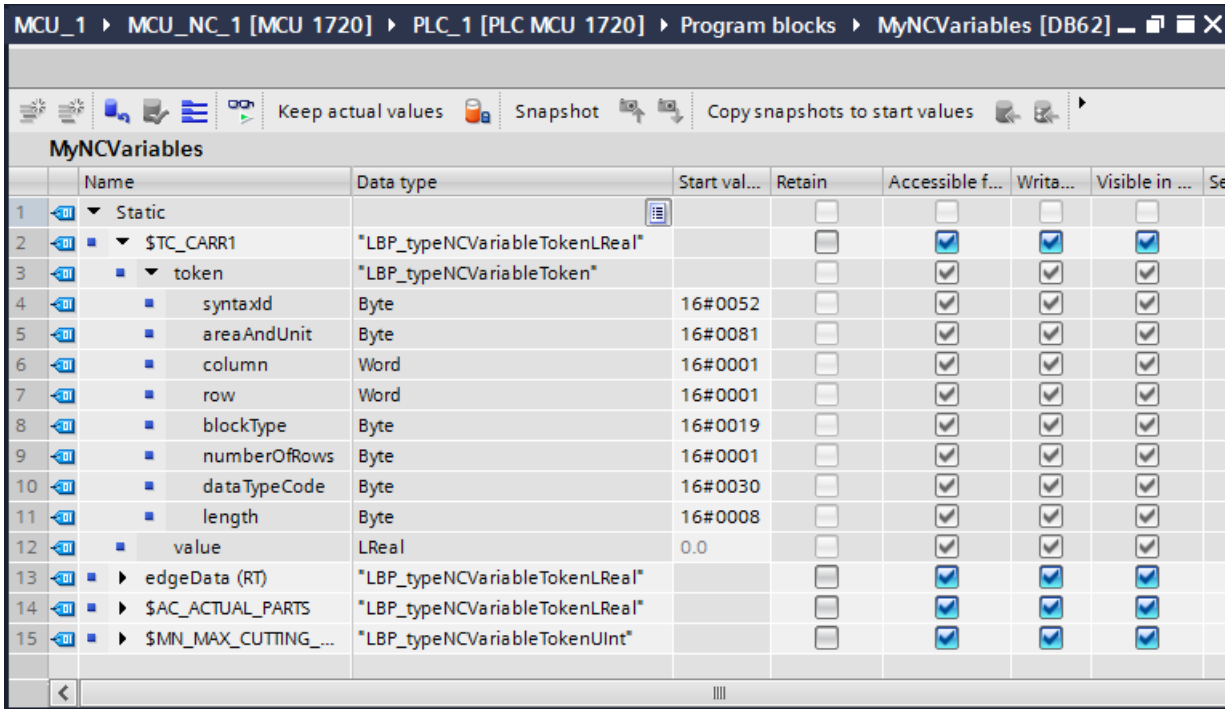
9.3.4.2 Use of existing NC variables (NC-VAR selector)

The NC variables configured in the TIA Portal differ from NC variables that were generated using the NC-VAR selector.

- The data structure of variables that were generated using the NC-VAR selector is defined using "STRUCT ... END_STRUCT".



- NC variables configured in the TIA Portal use the PLC data types (UDT) of the SINUMERIK PLC basic program.



Existing NC variables that were generated using the NC-VAR selector can continue to be used and combined with any variables configured in the TIA Portal.

9.3.4.3 Taking over NC variables into the tag table of the configured NC variables

Under the NCK in the project tree, you will find the "List of available NC variables", which contains all of the NC variables that you can use with the configured NCU firmware.

The NC variables that you want to access in the PLC user program can be managed in your project in user-defined NC tag tables.

Requirement

- A user-defined NC tag table is available (or the standard tag table is to be used).

Procedure

1. Double-click "List of all available NC variables", e.g. under "CNC_1 > NCK > Configured NC variables".
   The list opens.

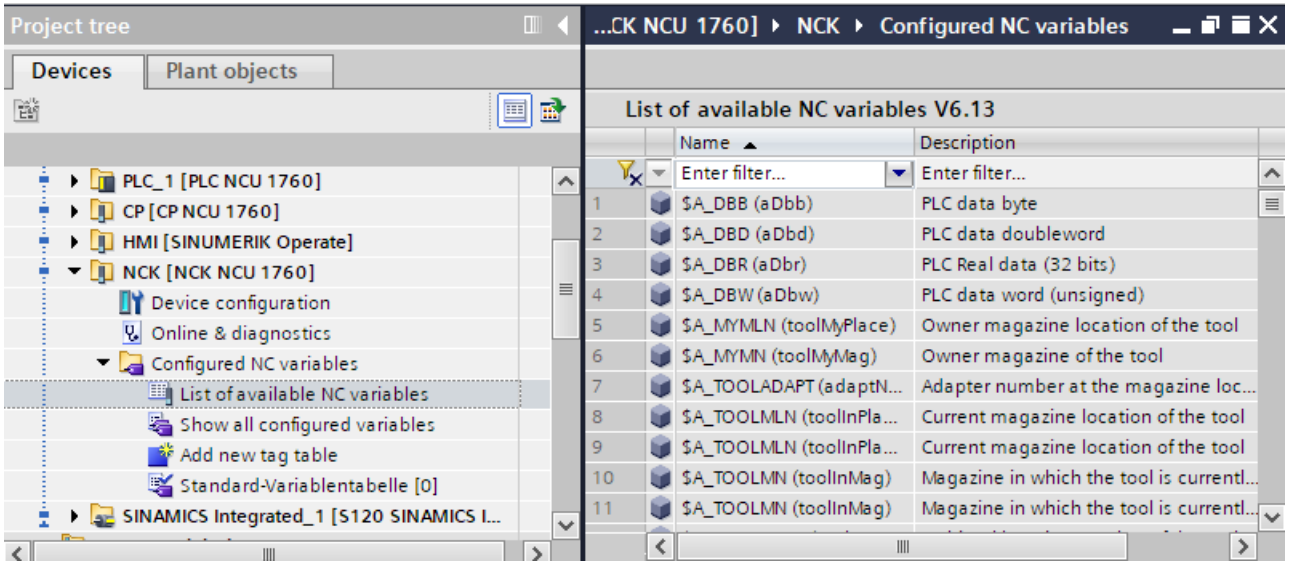
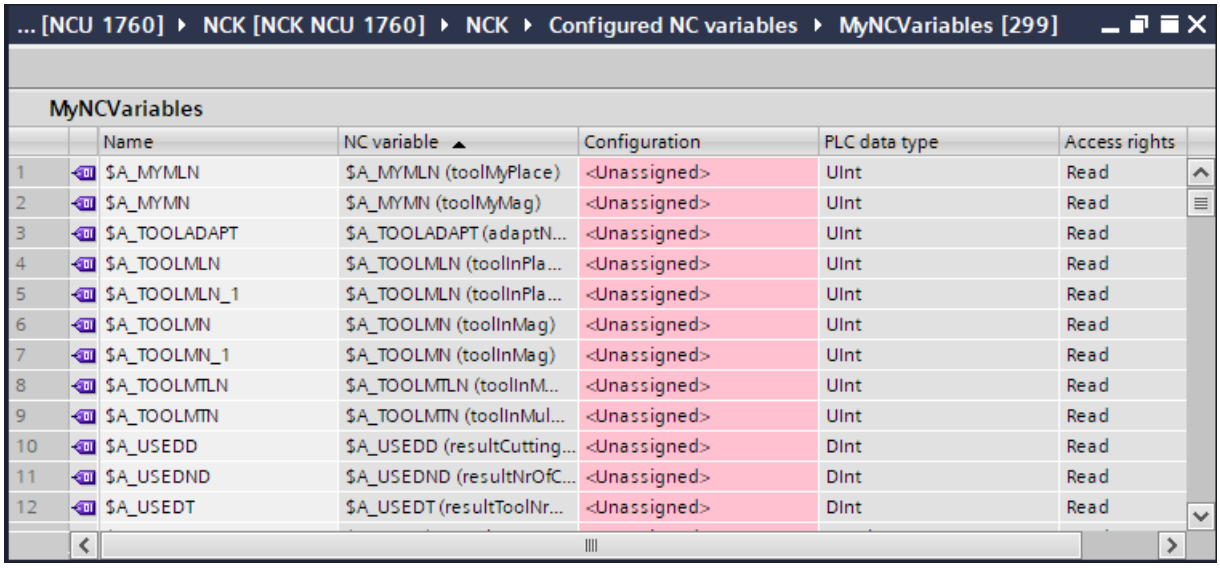


Figure 9-2 List of available NC variables

2. Select the required NC variables.
   - You can filter and browse the variables in the NC variable list to find the necessary variables.
   - You can select several NC variables using the Shift or Ctrl key to copy them simultaneously.
3. Copy the selected NC variables into your tag table using Copy and Paste or the drag-and-drop feature.

Result

The selected NC variables have been compiled in a user-defined tag table and can then be configured.



	Name	NC variable	Configuration	PLC data type	Access rights
1	$A_MYMLN	$A_MYMLN (toolMyPlace)	<Unassigned>	UInt	Read
2	$A_MYMN	$A_MYMN (toolMyMag)	<Unassigned>	UInt	Read
3	$A_TOOLADAPT	$A_TOOLADAPT (adaptN...	<Unassigned>	UInt	Read
4	$A_TOOLMLN	$A_TOOLMLN (toolInPla...	<Unassigned>	UInt	Read
5	$A_TOOLMLN_1	$A_TOOLMLN (toolInPla...	<Unassigned>	UInt	Read
6	$A_TOOLMN	$A_TOOLMN (toolInMag)	<Unassigned>	UInt	Read
7	$A_TOOLMN_1	$A_TOOLMN (toolInMag)	<Unassigned>	UInt	Read
8	$A_TOOLMTLN	$A_TOOLMTLN (toolInM...	<Unassigned>	UInt	Read
9	$A_TOOLMTN	$A_TOOLMTN (toolInMul...	<Unassigned>	UInt	Read
10	$A_USEDD	$A_USEDD (resultCutting...	<Unassigned>	DInt	Read
11	$A_USEDND	$A_USEDND (resultNrOfC...	<Unassigned>	DInt	Read
12	$A_USEDT	$A_USEDT (resultToolNr...	<Unassigned>	DInt	Read

9.3.5 Configuring selected NC variable instances

After you have imported variables from the list of all available NC variables into user-defined tag tables, make the required and, if applicable, optional settings for the selected NC variables. Missing or incorrect variable settings are highlighted in color in the table.

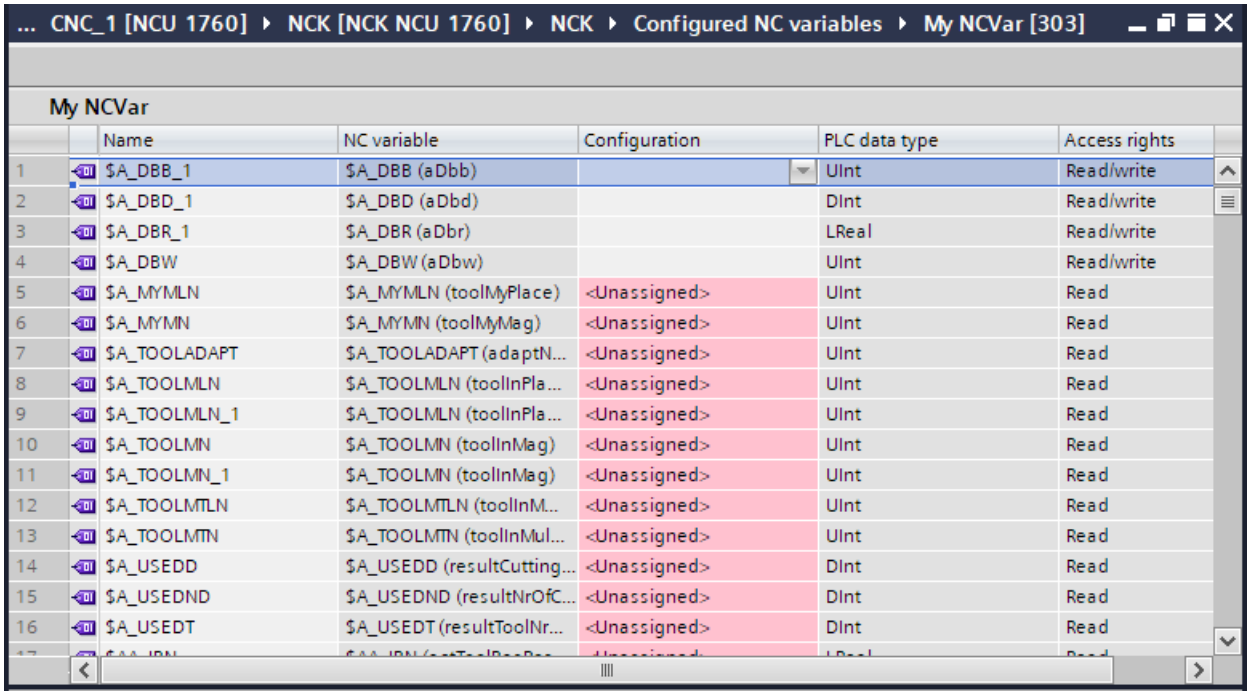
Requirement

- The variable instances to be configured are compiled in user-defined NC tag tables.

Procedure

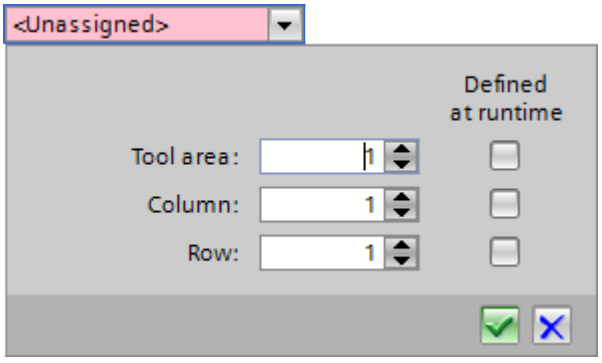
To configure NC tag tables, proceed as follows:

1. In the "Configured NC variables" folder in the project tree, double-click the command "Show all configured variables", e.g. under "CNC_1 > NCK > Configured NC variables > Show all configured variables".



2. Select the variables in succession and perform the required configuration and other adaptations as needed:

Property	Description
Name	Freely changeable symbolic name (alias name) of the variable via which you can address the variable in your PLC program. The default setting depends on the type of variable, either the MD number, the system variable name, or OPI name. If you paste the same NC variable several times, a number is appended to the name by default.
NC variable	Variable in the NCK on which this PLC tag instance is based. Depending on the type of variable, either the system variable name or the NC variable name is displayed.

Property	Description
Configuration (mandatory, if applicable)	Opens a dialog with required settings for the selected NC variable or the corresponding NCK object.  Depending on the type of variable, you must specify to which channel, axis, tool area or tool number the variable should refer to. You can enter these during the configuration as constant values or check the check box to define the values during runtime with the formal parameters for variable addressing in LPB_ReadVar [FB2] (Page 597) or LBP_WriteVar [FB3] (Page 605). Alternatively, you can edit the cell contents directly in the cell. You can find further information on the individual variables in the associated Help.
PLC data type	Shows the data type of the variable in PLC declaration.
Access rights	Displays whether a variable can only be read or also written. Note that the access rights are not checked on the PLC side, but only on the NCK side. In the PLC program, an unauthorized write request would be transferred to the interface, even if the associated NC variable cannot be written.
Comment	Tooltip for the variable. Freely editable, e.g. for documenting the use of the variable.

Result

The NC variables have been configured. A new DB can be generated from the underlying tag tables or the NC variables can be added to existing DB/FBs.

9.3.6 Integrating into the PLC program

9.3.6.1 Reading/writing the NC variables in the PLC program

To use the NC variable interface of the PLC basic program in your user program, insert a call-up of the corresponding PLC basic program block in your user program:

- LBP_ReadVar [FB2]
- LBP_WriteVar [FB3]

In the formal parameters of the block call, specify one or more of the previously configured access tokens, the data targets of the read/write access, and additional parameters as needed.

The required components of your user program can be located in the same blocks or different blocks, as seen in the two examples in the figures. You can find detailed information in the descriptions for the corresponding PLC basic program blocks LBP_ReadVar [FB2] (Page 597) and LBP_WriteVar [FB3] (Page 605).

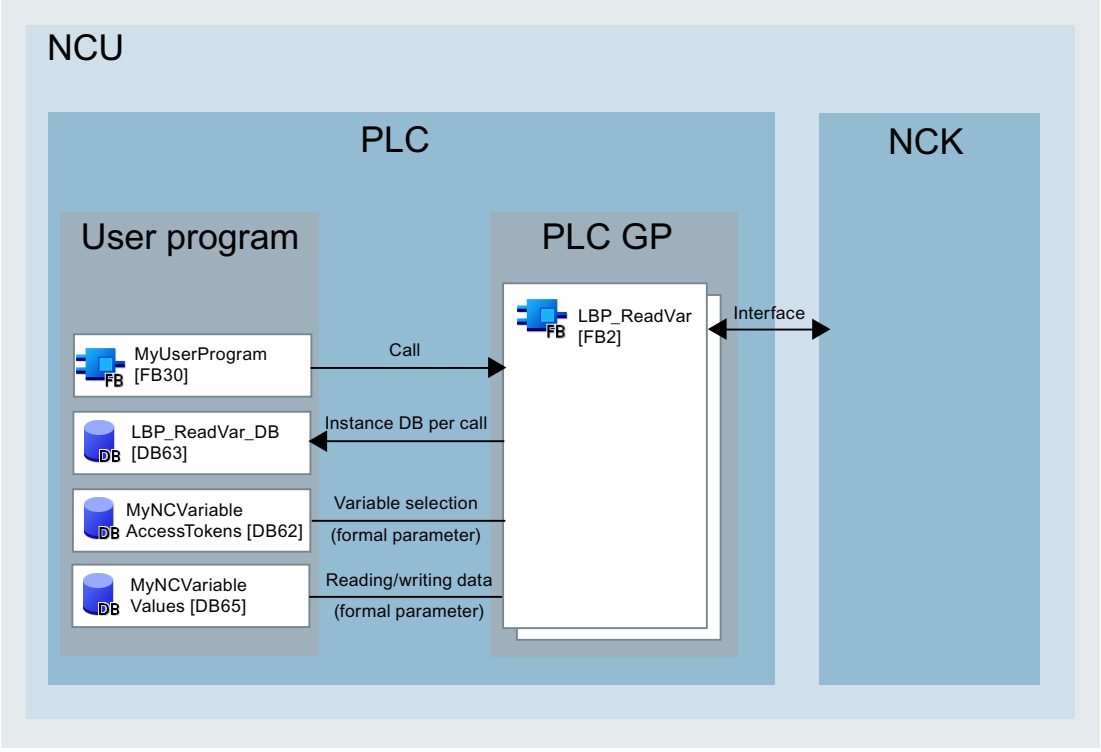


Figure 9-3 NC variable access via PLC basic program - Example with several DBs

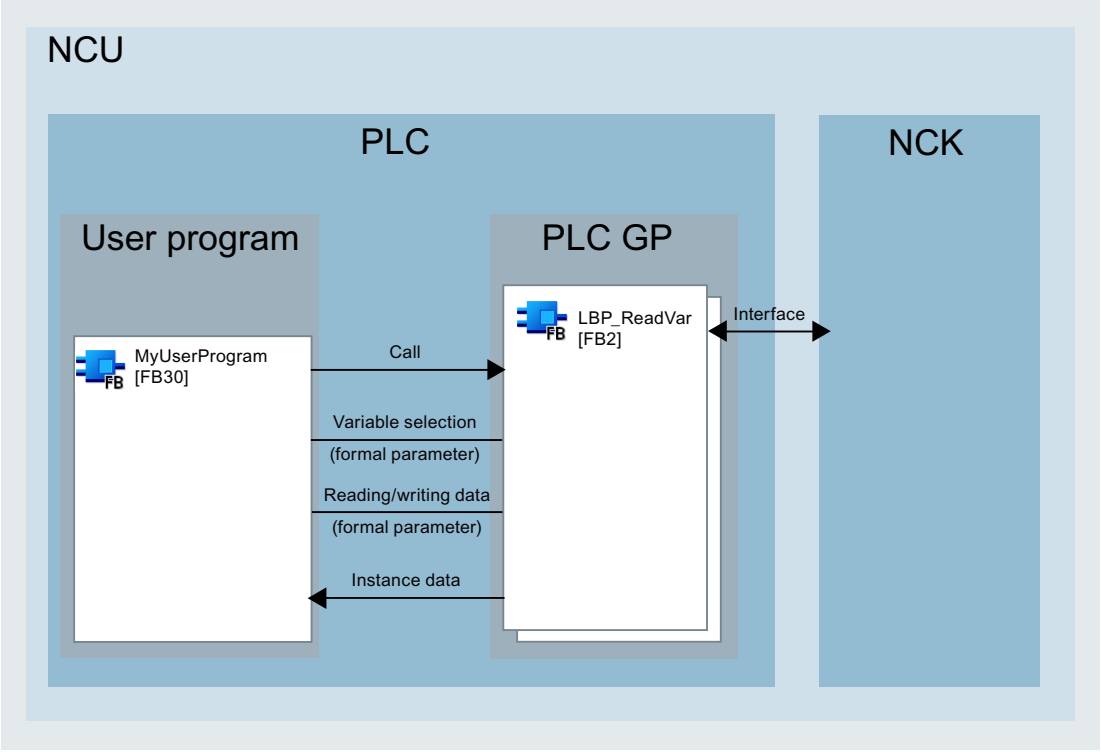


Figure 9-4 NC variable access via PLC basic program - Example with individual FB

9.3.6.2 Creating a data block from an NC variable table

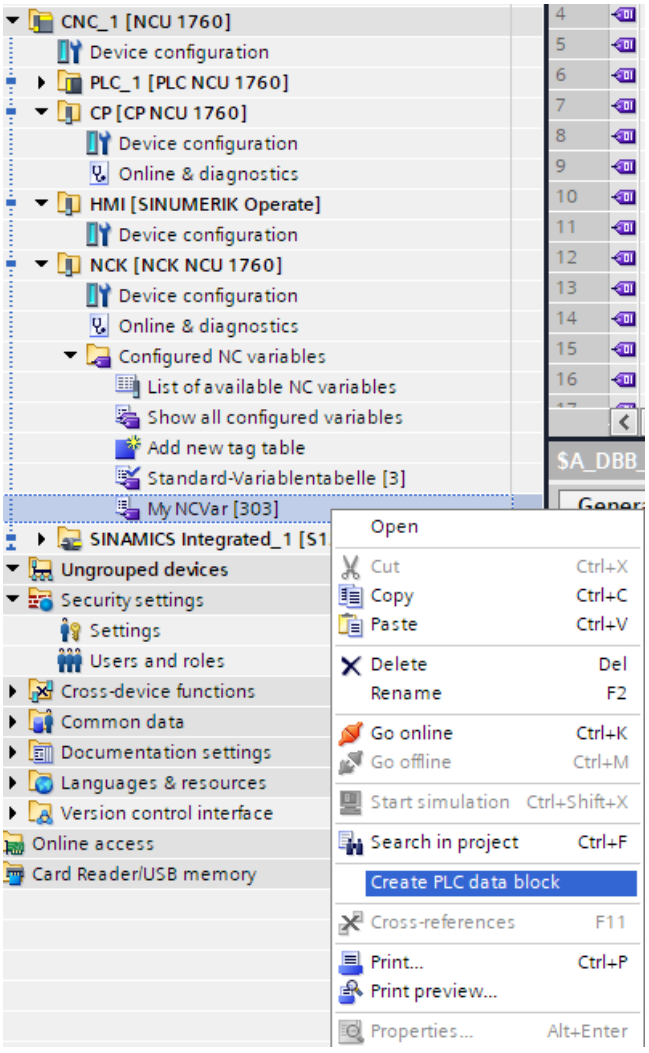
Requirement

- A user-defined NC tag table is available.
- All of the NC variables of the user-defined NC tag table are completely and correctly configured.
- The data types of the NC variables are added.

Procedure

To generate a data block from an NC tag table, proceed as follows:

1. In the project navigation, open the "NC variables" folder, e.g. under "CNC_1 > NCK > NC variables".
2. Right-click on the tag table and select the command "Create PLC data block".



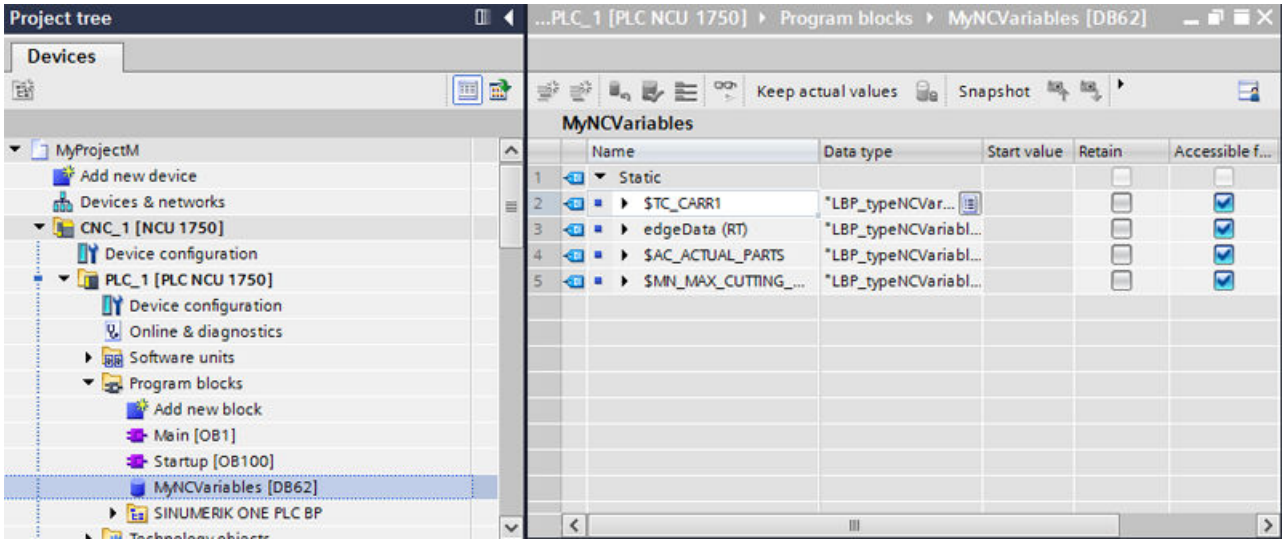
Note

The name of the PLC data block must only ever be assigned once. If a PLC data block with the name already exists, it will be overwritten.

You must confirm overwriting.

Result

The data block has been generated and added in the program blocks folder. The name of the underlying tag table is used as a block name.



With the generated DB, you can now address the compiled variables in your PLC program (Page 78) and read or write the variables via the blocks LBP_WriteVar [FB3] (Page 605) and LBP_ReadVar [FB2] (Page 597).

9.3.6.3 Copying configured NC variables to an existing DB/FB

Requirement

- The NC variables to be inserted are completely and correctly configured.
- The data types of the NC variables are added.
- A DB/FB with write access is available as the destination for the NC variables.
- The target system is compatible with the NC variables to be inserted.

Note

No initial values (token constants) for textual block interface

If you have set the textual representation of the block interface that can be activated for SCL, no constants of the tokens can be generated for NC variables inserted into FBs.

Alternatively, you can use the tabular representation or access NC variables that were generated with the NC-VAR selector.

Procedure

To insert configured NC variables into an existing DB or FB, proceed as follows:

1. In the project navigation, open the corresponding user-defined tag table or click on "Display all configured variables", e.g. under "CNC_1 > NCK > NC variables".
2. Select all of the variables that you want to copy.
3. Insert the variables into the DB/FB, e.g. as follows:
   - Right-click on one of the selected NC variables and choose "Copy", then right-click on the DB/FB in the project tree and click "Insert".
   - Drag the selected variables and drop them on the DB/FB in the project tree.

Result

The NC variables have been inserted into the DB/FB.

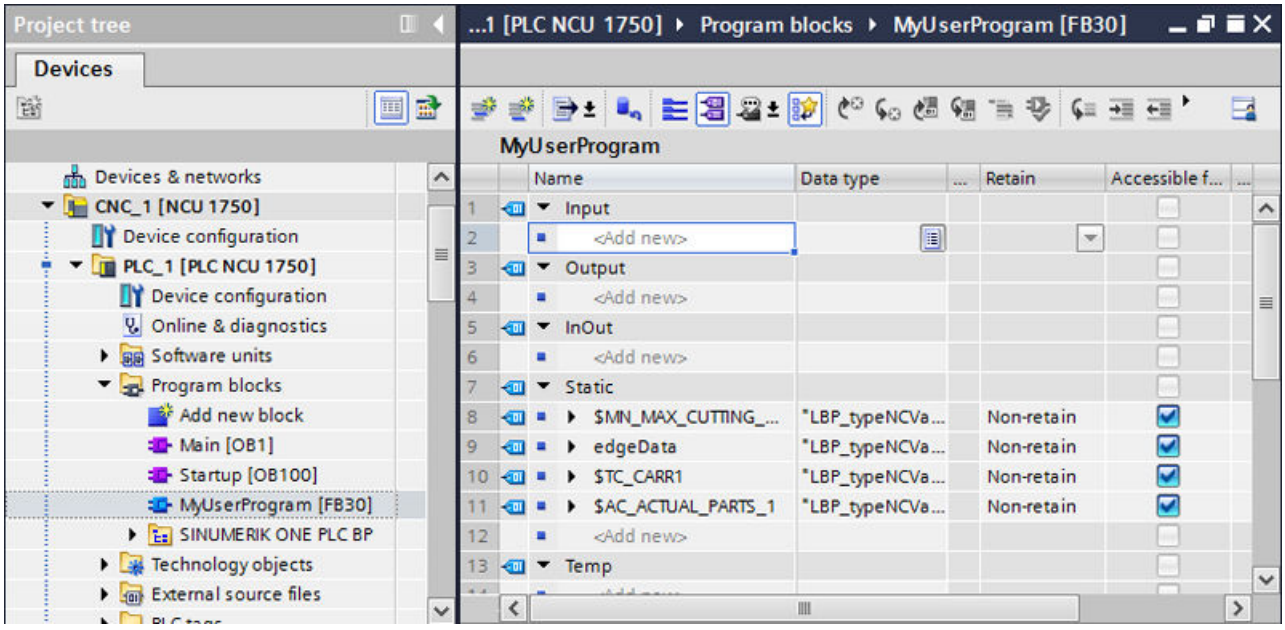


Figure 9-5 Example block interface from MyUserProgram [FB30] with NC variables

With this DB/FB, you can now address the compiled variables in your PLC program (Page 78) and read or write the variables via the blocks LBP_WriteVar [FB3] (Page 605) and LBP_ReadVar [FB2] (Page 597).

9.3.6.4 Corresponding data type declarations for NCK and PLC

The data types define the properties of the data, e.g. the display of the content and the permitted memory areas.

NC variables use different data type declarations than STEP 7 variables (PLC program). In the NC variable list, the data type is displayed in NCK declaration. When you insert an NC variable in your tag table and use it in the PLC program, you must use the PLC declaration.

Table 9-2 Assigning the data type declarations of NCK and STEP 7 (PLC)

Data type		**Format**	**Value range**	**Brief description**
NCK declaration	**PLC declaration**			
BOOL	BOOL	Unsigned integers (decimal number system)	0 or 1 or FALSE or TRUE	An operand of data type BOOL represents a bit value.
		Binary numbers	2#0 or 2#1	
		Octal numbers	8#0 or 8#1	
		Hexadecimal numbers	16#0 or 16#1	
BYTE	USINT	Unsigned integers (decimal number system)	0 to 255	An operand of data type USINT has a length of 8 bits and contains unsigned numerical values.
		Binary numbers	2#0 to 2#01111_1111	
		Octal numbers	8#0 to 8#377	
		Hexadecimal numbers	16#0 to 16#FF	
DATE-TIME	DT (DATE_AND_TIME)	Date and time (year-month-day-hour:minute:second:millisecond)	Min.: DT#1990-01-01-00:00:00.000 Max.: DT#2089-12-31-23:59:59.999	The data type DT (DATE_AND_TIME) stores date and time information in the BCD format.
DOUBLE	LREAL	Floating-point numbers according to IEEE754	-1.7976931348623157e+308 to -2.2250738585072014e-308 ±0.0 +2.2250738585072014e-308 to +1.7976931348623157e+308	Operands of data type LREAL have a length of 64 bits and are used for displaying floating point numbers.
INTEGER	DINT	Signed integers (decimal number system)	-2_147_483_648 to +2_147_483_647	An operand of data type DINT (Double INT) has a length of 32 bits and comprises two components: a sign and a numerical value in the two's complement.
		Binary numbers (only positive)	2#0 to 2#0111_1111_1111_1111_1111_1111_1111_1111	
		Octal numbers (only positive)	8#0 to 8#177_7777_7777	
		Hexadecimal numbers	16#0 to 16#7FFF_FFFF	

Data type		Format	Value range	Brief description
NCK declaration	PLC declaration			
STRING	STRING	ASCII character set including special characters	0 to 254 characters	An operand of data type STRING stores several characters in a character string, which can consist of a maximum of 254 characters.
UDWORD	UDINT	Unsigned integers (decimal number system)	0 to 4_294_967_295	An operand of data type UDINT has a length of 32 bits and contains unsigned numerical values.
		Binary numbers	2#0 to 2#1111_1111_1111_1111_1111_1111_1111_1111	
		Octal numbers	8#0 to 8#377_7777_7777	
		Hexadecimal numbers	16#0 to 16#FFFF_FFFF	
UINT_16	UINT	Unsigned integers (decimal number system)	0 to 65_535	An operand of data type UINT (Unsigned INT) has a length of 16 bits and contains number values without sign. An operand of the data type UINT occupies two bytes in the memory.
		Binary numbers	2#0 bis 2#1111_1111_1111_1111	
		Octal numbers	8#0 to 8#17_7777	
		Hexadecimal numbers	16#0 to 16#FFFF	

See also: LBP_ReadVar [FB2] (Page 597), LBP_WriteVar [FB3] (Page 605), STEP 7 help on the topic Overview of the valid data types.

9.3.7 Managing NC variables in libraries

NC variables are compatible with global libraries. You can save the following objects with NC variables as copy templates in libraries:

- DB/FB with NC variables
- User-defined NC variable table

Thus, you can reuse your user-defined configured NC variable instances in other projects. The compatibility with the new controller in the target device is checked when the copy template is inserted.

Reference on the PLC basic program 10

10.1 Overview of the PLC blocks

10.1.1 Data blocks (DB) and associated PLC user data types

In the following, you will find a list of all of the data blocks of the SINUMERIK PLC basic program and, if applicable, the assigned PLC user data types (UDT).

Data blocks are required for the parameter settings made in the NC machine data, e.g. the channel and axis data blocks. You can use the copy templates in the system library to copy these blocks and UDT.

Note

Limited functionality of PLC basic program blocks

Some PLC basic program blocks do not yet have their full functionality. This is noted in the table below.

However, all blocks are contained in the system library of the basic program as objects so that they can be included in your project or called in the user program. After the publication of an updated version of the basic program, you can upgrade these blocks to establish the functionality.

Note

Block numbers available in the user program

The block assignment is described in the assignment overview in the Help for the SINUMERIK ONE STEP 7 Toolbox V16. Look there to find out which block numbers you can use in the user program.

Note

Symbolic programming

Use symbolic programming for accessing the data. In the block descriptions, the absolute addresses are specified in brackets for reasons of compatibility and ease of migration.

Table 10-1 Data blocks (DB) and associated PLC data types (UDT) of the PLC basic program

DB	Name	Meaning	UDT
DB2	LBP_AlarmMsgs	Interface between user program and LBP_GenAlarmMsgs [FC10], contains bit arrays for blocking and stop signals as well as an error and system status messages.	LBP_typeAlarmMsgs
DB3	LBP_InternalDB3	Only for internal use in the basic program.	LBP_typeAlarmMsgs

DB	Name	Meaning	UDT
DB5	LBP_ParamAlarmMsgs	Interface between user program and LBP_GenAlarmMsgs [FC10], contains the parameter assignment of the alarms for the bit arrays in the LBP_AlarmMsgs [DB2] as an error or system status message.	
DB7	LBP_ConfigData	Global DB, contains parameters for the configuration and initialization of the basic program.	LBP_typeConfigData
DB8	LBP_InternalDB8	DB only for internal use in the basic program.	
DB10	LBP_NC	Interface between user program and NC, contains signals from/to the NC, PLC and operating software.	LBP_typeNC
DB11	LBP_ModeGroup	Interface between user program and mode groups, contains signals from/to the NC, PLC and operating software.	LBP_typeModeGroup
DB16	LBP_PIServices	Interface for LBP_ReqPIService [FB7], contains definitions for all available PI services.	
DB19	LBP_HMI	Interface between user program and operator panel, contains signals from/to the operating software.	LBP_typeHMI
DB21...30	LBP_Chan1...LBP_Chan10	Interface between user program and channel, contains signals from/to channel x. **Information about the current functionality:** H-function values are not yet available divided into REAL and LREAL data types. Symbolic names of the bits have not yet been finally defined.	LBP_typeChanX
DB31...61	LBP_Axis1...LBP_Axis31	Interface between user program and axis/spindle, contains signals from/to axis/spindle x.	LBP_typeAxisX
DB71	LBP_TMLoad	Interface between user program and tool management, contains signals from/to Load/Unload magazine.	LBP_typeTMLoad
DB72	LBP_TMSpindle	Interface between user program and tool management, contains signals for spindle as change position.	LBP_typeTMSpindle
DB73	LBP_TMTurret	Interface between user program and tool management, contains signals for turret.	LBP_typeTMTurret
DB75	LBP_MFuncDecListConfig	Decoding list for M decoding according to list	
DB76	LBP_MFuncDecListSignals	Signal list for M decoding according to list	
DB78	LBP_PowerrideConfig	Data records with preconfigured behavior for the Powerride	
DB1000	LBP_CtrlEnergy	Interface between user program and operating software, contains energy saving profiles for Ctrl-Energy. **No support available yet for the current SINUMERIK PLC Basic Program for SINUMERIK ONE (V06.xx.xx.xx).**	
DB1001	LBP_SentronPac	Interface between user program and operating software, contains signals for SENTRON PAC energy monitoring devices. **No support available yet for the current SINUMERIK PLC Basic Program for SINUMERIK ONE (V06.xx.xx.xx).**	
DB1002	LBP_SpindleTempSensor	Interface between user program and operating software, contains signals for spindle temperature sensors. **No support available yet for the current SINUMERIK PLC Basic Program for SINUMERIK ONE (V06.xx.xx.xx).**	
DB1020	LBP_InternalDB1020	Only for internal use in the basic program.	

DB	Name	Meaning	UDT
DB1071	LBP_TMLoadMT	Interface between user program and tool management, contains signals from/to Load/Unload magazine (multi-tool).	LBP_typeTMLoadMT
DB1072	LBP_TMSpindleMT	Interface between user program and tool management, contains signals for spindle as change position (multitool).	LBP_typeTMSpin-dleMT
DB1073	LBP_TMTurretMT	Interface between user program and tool management, contains signals for turret (multitool).	LBP_typeTMTurretMT

10.1.2 Function blocks (FB)

In the following, you will find a list of all of the function blocks of the SINUMERIK PLC basic program.

You can copy these blocks to your project using copy templates from the system library. If you copy a basic program block into your project, which itself calls further basic program blocks (column "Blocks used"), you must also copy these other blocks into your project.

Note

Limited functionality of PLC basic program blocks

Some PLC basic program blocks do not yet have their full functionality. Where relevant, this is noted in the table below:

However, all blocks are contained in the system library of the basic program as objects so that they can be included in your project or called in the user program. After publication of the updated basic program version you can upgrade these blocks in order to establish the functionality.

Note

Block numbers available in the user program

The assignment overview provided in the Help of the SINUMERIK ONE STEP 7 Toolbox V16 shows which block numbers or block addresses are reserved by Siemens and the PLC basic program, and which are available for use in the user program.

Table 10-2 Function blocks (FB) of the SINUMERIK PLC basic program

FB	Name	Description	Blocks used
FB2	LBP_ReadVar	The block is used for reading NC variables.	LBP_InternalProc1 [FC25000] LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB3 [DB8]
FB3	LBP_WriteVar	The block is used for writing NC variables.	LBP_InternalProc1 [FC25000] LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB3 [DB8]
FB5	LBP_ReadGUD	The block is used for reading global user data (GUD) from the NC and for determining the GUD variable address.	LBP_InternalProc1 [FC25000] LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB3 [DB8]
FB7	LBP_ReqPIService	The block is used for starting PI services that are defined in the LBP_PIServices [DB16].	LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB3 [DB8] LBP_PIServices [DB16]
FB9	LBP_M2NSwitchOver	The block is used for switching over operator components (MCP/OP), which are connected with one or several control modules NCU; uses the signals from LBP_HMI [DB19]. **No support available yet for the current SINUMERIK PLC Basic Program for SINUMERIK ONE (V06.xx.xx.xx).**	

FB	Name	Description	Blocks used
FB21	LBP_Powerride	The block integrates a rotary/push control dial (SINUMERIK Powerride) into the SINUMERIK environment.	LBP_ConfigBP [FC1]
FB25000	LBP_OpUnitComm	Block is used for communication with operator components via Industrial Ethernet (IE), e.g. machine control panels (MCP), handheld units (HT), and direct key modules (OpKey). **This function is not available in the virtual SINUMERIK.**	LBP_InternalProc2 [FB25001] LBP_InternalProc5 [FC25002] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB3 [DB8] LBP_ConfigData [DB7]
FB25002	LBP_OpUnitWriteRecord	Writes a data record to a modular MCP. **This function is not available in the virtual SINUMERIK.**	

10.1.3 Function blocks (FC)

In the following, you will find a list of all of the functions of the SINUMERIK PLC basic program.

You can copy these blocks to your project using copy templates from the system library. If you copy a basic program block into your project, which itself calls further basic program blocks (column "Blocks used"), you must also copy these other blocks into your project.

Note

Limited functionality of PLC basic program blocks

Some PLC basic program blocks do not yet have their full functionality. Where relevant, this is noted in the table below:

However, all blocks are contained in the system library of the basic program as objects so that they can be included in your project or called in the user program. After publication of the updated basic program version you can upgrade these blocks in order to establish the functionality.

Note

Block numbers available in the user program

The assignment overview provided in the Help of the SINUMERIK ONE STEP 7 Toolbox V16 shows which block numbers or block addresses are reserved by Siemens and the PLC basic program, and which are available for use in the user program.

Table 10-3 Functions (FC) of the SINUMERIK PLC basic program

FC	Name	Description	Blocks used
FC1	LBP_ConfigBP	Block configures and initializes the basic program, starts synchronization between the PLC and NCK, must be called in the startup OB.	LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB3 [DB3] LBP_ConfigData [DB7] LBP_InternalDB8 [DB8]
FC2	LBP_MainBP	Block processes the cyclic part of the basic program, must be called in the cycle OB.	LBP_InternalProc3 [FC25001] LBP_InternalProc4 [FC25003] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_ConfigData [DB7] LBP_InternalDB8 [DB8]
FC3	LBP_NCKProcessIRT	Block processes the block-synchronous part of the basic program, must be called in the hardware interrupt OB.	LBP_InternalProc3 [FC25001] LBP_InternalProc4 [FC25003] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB8 [DB8]
FC6	LBP_TMTransferMT	The block transfers position and status information of the tools to the tool management interface in connection with Multitool.	LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB8 [DB8]
FC7	LBP_TMTransferTurret	The block acknowledges an implemented tool change of a turret to the tool management interface.	
FC8	LBP_TMTransfer	The block transfers position and status information of the tools to the tool management interface.	

FC	Name	Description	Blocks used
FC9	LBP_ReqASUP	The block starts asynchronous subprograms, the prerequisite is that it must be selected and parameterized by an NC program or by the LBP_ReqPI-Service [FB7] (PI service ASUB).	LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB8 [DB8]
FC10	LBP_GenerateAlarmMsgs	The block evaluates the signals entered in LBP_AlarmMsgs [DB2], generates incoming and outgoing error and system status messages of the operating software and acknowledges error messages. Optionally, the disable and stop signals can be influenced.	LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_ConfigData [DB7]
FC12	LBP_CallBackAuxFunc	Call interface for users for auxiliary functions, the block is called on an event-driven basis in the basic program when new auxiliary functions are available.	
FC13	LBP_HTCtrlHT2	Block takes over display control of the handheld unit (HT2).	LBP_ConfigData [DB7]
FC17	LBP_CtrlStarDelta	The block is used for star-delta changeover for digital main spindle drives. **No support available yet for the current SINUMERIK PLC Basic Program for SINUMERIK ONE (V06.xx.xx.xx).**	
FC18	LBP_CtrlAxisSpindle	The block controls axes and spindles from the user program.	LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004]
FC19	LBP_MCPCtrlMilling	The block is used for transferring data from the machine control panel (MCP milling version) to the NC/PLC interface.	LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_ConfigData [DB7] LBP_InternalDB8 [DB8] LBP_NC [DB10] LBP_ModeGroup [DB11]
FC21	LBP_TransferSelData	The block is used for high-speed data exchange between PLC and NCK.	LBP_InternalProc3 [FC25001] LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB8 [DB8]

FC	Name	Description	Blocks used
FC24	LBP_MCPCtrlMillingSmall	The block is used for transferring data from the machine control panel (MCP milling version compact) to the NC/PLC interface.	LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_InternalDB3 [DB3] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB8 [DB8] LBP_ConfigData [DB7]
FC25	LBP_MCPCtrlTurning	The block is used for transferring data from the machine control panel (MCP turning version) to the NC/PLC interface.	
FC26	LBP_HTCtrlHT8	The block is used for transferring data of the HT 8 to the NC/PLC interface.	LBP_InternalProc6 [FC25004] LBP_InternalFB1020 [FB1020] LBP_InternalDB1020 [DB1020] LBP_AlarmMsgs [DB2] LBP_ParamAlarmMsgs [DB5] LBP_InternalDB8 [DB8] LBP_ConfigData [DB7]
FC25002	LBP_Handwheels	Block used to transfer handwheel signals	LBP_OpUnitComm [FB25000]

10.2 Operator panels

10.2.1 Introduction

General

The data interface comprises the following parts:

- Data interface
- Function interface

The exchange of signals and data is organized by the PLC basic program and performed between the following components:

- PLC user program
- NC
- Operating software
- Machine control panel

The blocks and data of the NC/PLC interface signals are described in this manual.

Inverse signals

Inverse signals are designated with "*".

Example:

Signals from the machine control panel, IB n + 2, Bit4: *Spindle stop:

1: Spindle stop is not requested
0: Spindle stop is requested

Access rights

The following access rights are specified:

[r] Designated area "read only" permitted
[r/w] Designated area "read and write" permitted

10.3 Data types

10.3.1 LBP_typeHandwheel

The data type "LBP_typeHandwheel" facilitates the handwheel signals state check and supports symbolic programming.

Variables in "LBP_typeHandwheel"

handWheel1, handWheel2

Description	Image of the inputs of the handwheel **handWheel1** p+0 and p+1 **handWheel2** p+2 and p+3
Data type	INT

10.3.2 LBP_typeMcp*

The basic program library contains a range of "LBP_typeMcp*" data types to simplify use of machine control panels and, in particular, to support symbolic programming.

The data types contain the input and output images of the machine control panels. The description of the input and output images can be found in the Equipment Manuals for the operator components/MCPs on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109736211) and in Chapter Input and output signals (Page 589).

Data types for PROFINET MCP

The table lists the available data types depending on the machine control panel.

Machine control panel	MCP data type
MCP310CPN	LBP_typeMcp310CPnInput, LBP_typeMcp310CPnOutput
MCP310PN	LBP_typeMcp310PnInput, LBP_typeMcp310PnOutput
MCP483CPN	LBP_typeMcp483CPnInput, LBP_typeMcp483CPnOutput
MCP483PN	LBP_typeMcp483PnInput, LBP_typeMcp483PnOutput

Data types for PROFIBUS MCP

The table lists the available data types depending on the machine control panel.

Machine control panel	MCP data type
MCP310	LBP_typeMcp310StandardInput, LBP_typeMcp310StandardOutput, LBP_typeMcp310AdditionalInput, LBP_typeMcp310AdditionalOutput
MCP483	LBP_typeMcp483StandardInput, LBP_typeMcp483StandardOutput, LBP_typeMcp483AdditionalInput, LBP_typeMcp483AdditionalOutput

Data types for Industrial Ethernet (IE)

The table lists the available data types depending on the machine control panel.

Machine control panel	MCP data type
slimline version	LBP_typeMcp310StandardInput, LBP_typeMcp310StandardOutput
M version	LBP_typeMcp483MillingInput, LBP_typeMcp483MillingOutput
T version	LBP_typeMcp483TurningInput, LBP_typeMcp483TurningOutput

See also

Configuration machine control panel, handheld unit, direct keys (Page 751)

10.3.3 LBP_typeModularMcpKeypadRecord

Data type "LBP_typeModularMcpKeypadRecord" supports writing a data record to an MCP with LBP_OpUnitWriteRecord [FB25002]: Writing a data record to MCP (Page 659). The hardware and data assignment of the MCP are described in the Operating Components Manual Modular MCPs.

10.4 Data blocks (DBs)

In this chapter, each data block is described in a separate section.

To structure the descriptions more clearly, anonymous structures are described under the name "STRUCT ..." outside the data block description but in the same section as the data block.

10.4.1 Overview of the interface signals

The following table provides an overview of the interface signals:

Signals from/to the machine control panel	
	M version, signals from the MCP: Input image (Page 589) M version, signals to the MCP: Output image (Page 593)
	T version, signals from the MCP: Input image (Page 590) T version, signals to the MCP: Output image (Page 593)
	Slimline version, signals from the MCP: Input image (Page 590) Slimline version, signals to the MCP: Output image (Page 594)
Signals from/to the handheld unit HT 2	
	Signals from the handheld unit: Input image (Page 591) Signals to the handheld unit: Output image (Page 595)
Signals from/to the handheld unit HT 8	
	Signals from the handheld unit HT 8: Input image (Page 591) Signals to handheld terminal HT 8: Output image (Page 596)
Signals from/to the handheld unit HT 10	
	Signals from the handheld unit HT 10: Input image (Page 592) Signals to handheld terminal HT 10: Output image (Page 596)
PLC alarms/messages	
	LBP_AlarmMsgs [DB2] - PLC alarms / messages (Page 98)
Signals from/to the NC, PLC and operating software	
	LBP_NC [DB10] - Signals from/to the NC, PLC and operating software (Page 133)
Mode-group-specific signals	
	LBP_ModeGroup [DB11] (Page 187)
Signals from/to the operator panel (OP)	
	LBP_HMI[DB19] (Page 215)
Channel-specific signals	
	LBP_Chan1 [DB21], ... - Channel-specific signals (Page 241)
Axis/spindle signals	
	LBP_Axis 1 [DB31], ... - Axis/spindle signals (Page 381)

10.4.2 Overview of further interface signals (ONE)

The following table provides an overview of the interface signals:

Tool management	
	LBP_TMLoad [DB71] - Interface for loading/unloading the magazine: (Page 515)
	LBP_TMSpindle [DB72] - Spindle as change position (Page 522)
	LBP_TMTurret [DB73] - Turret (Page 532)
Signals for Ctrl-Energy	
	LBP_CtrlEnergy [DB1000] - Energy-saving profiles (Page 554)
SENTRON PAC	

	LBP_SentronPac [DB1001] - SENTRON PAC (Page 559)
Spindle temperature sensor	
	LBP_SpindleTempSensor [DB1002] - Spindle temperature sensors (Page 570)
Interface to the tool management, extended area	
	LBP_TMLoadMT [DB1071] - Loading/unloading magazine: Multitool (Page 575)
	LBP_TMSpindleMT [DB1072] - Spindle: Multitool (Page 577)
	LBP_TMTurretMT [DB1073] - Turret: Multitool (Page 583)

10.4.3 LBP_AlarmMsgs [DB2] - PLC alarms / messages

Message type

- **FM**: An error message with the associated alarm number as the error number is triggered by the signal.
- **OM**: A system status message with the associated alarm number as the message number is triggered by the signal.

References:
A detailed description of error and system status messages can be found in Chapter "LBP_GenerateAlarmMsgs [FC10]: error and operating messages (Page 690)":

See also

LBP_AlarmMsgs [DB2] (Page 112)

10.4.3.1 Overview

LBP_AlarmMsgs [DB2], channel range 1

Table 10-4 Channel range 1

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Signals WITHOUT the display of a fault/operating message (DBB0 - 309)							
	Channel 1							
0	Feedrate disable							
1	Feedrate disable							
2	Read-in disable							
3	Read-in disable							
4	Start disable							

LBP_Alarm Msgs [DB2]	**Signals for PLC events (PLC → HMI)**							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
5	Start disable							
6	Feedrate stop, geo axis 1, byte 1							
7	Feedrate stop, geo axis 1, byte 2							
8	Feedrate stop, geo axis 2, byte 1							
9	Feedrate stop, geo axis 2, byte 2							
10	Feedrate stop, geo axis 3, byte 1							
11	Feedrate stop, geo axis 3, byte 2							
12 - 119	**Channel 2 - channel 10,** see above "Channel 1"							
	Axis/spindle 1							
120	Feedrate stop / spindle stop, byte 1							
121	Feedrate stop / spindle stop, byte 2							
122 - 181	**Axis / spindle 2 - 31,** see above "Axis /spindle 1"							
	Additional values for user range 0							
182	Additional value for event number 700000							
184	Additional value for event number 700001							
…	…							
308	Additional value for event number 700063							
	Signals WITH the display of a fault/operating message (from DBB 310)							
	Channel 1							
	Feedrate disable (event no.: 510000-510015)							
310 (FM)	510007	510006	510005	510004	510003	510002	510001	510000
311 (OM)	510015	510014	510013	510012	510011	510010	510009	510008
312 (FM)	Feedrate and read-in disable: Byte 1 (event no.: 510100-510107)							
313 (FM)	Feedrate and read-in disable: Byte 2 (event no.: 510108-510115)							
314 (OM)	Feedrate and read-in disable: Byte 3 (event no.: 510116-510123)							
315 (OM)	Feedrate and read-in disable: Byte 4 (event no.: 510124-510131)							
316 (FM)	Read-in disable Byte 1 (event no.: 510200-510207)							
317 (FM)	Read-in disable Byte 2 (event no.: 510208-510215)							
318 (OM)	Read-in disable Byte 3 (event no.: 510216-510223)							
319 (OM)	Read-in disable Byte 4 (event no.: 510224-510231)							
320 (FM)	NC start disable: Byte 1 (event no.: 510300-510307)							
321 (OM)	NC start disable: Byte 2 (event no.: 510308-510315)							
322 (FM)	Feedrate stop, geo axis 1: Byte 1 (event no.: 511100-511107)							
323 (OM)	Feedrate stop, geo axis 1: Byte 2 (event no.: 511108-511115)							
324 (FM)	Feedrate stop, geo axis 2: Byte 1 (event no.: 511200-511207)							
325 (OM)	Feedrate stop, geo axis 2: Byte 2 (event no.: 511208-511215)							

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
326 (FM)	Feedrate stop, geo axis 3: Byte 1 (event no.: 511300-511307)							
327 (OM)	Feedrate stop, geo axis 3: Byte 2 (event no.: 511308-511315)							

LBP_AlarmMsgs [DB2], channel range 2

Table 10-5 Channel range 2

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Channel 2							
	Feedrate disable (event no.: 510000-520015)							
328 (FM)	520007	520006	520005	520004	520003	520002	520001	520000
329 (OM)	520015	520014	520013	520012	520011	520010	520009	520008
330 (FM)	Feedrate and read-in disable, byte 1 (event no.: 520100-520107)							
331 (FM)	Feedrate and read-in disable, byte 2 (event no.: 520108-520115)							
332 (OM)	Feedrate and read-in disable, byte 3 (event no.: 520116-520123)							
333 (OM)	Feedrate and read-in disable, byte 4 (event no.: 520124-520131)							
334 (FM)	Read-in disable, byte 1 (event no.: 520200-520207)							
335 (FM)	Read-in disable, byte 2 (event no.: 520208-520215)							
336 (OM)	Read-in disable, byte 3 (event no.: 520216-520223)							
337 (OM)	Read-in disable, byte 4 (event no.: 520224-520231)							
338 (FM)	NC start disable, byte 1 (event no.: 520300-520307)							
339 (OM)	NC start disable, byte 2 (event no.: 520308-520315)							
340 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 521100-521107)							
341 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 521108-521115)							
342 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 521200-521207)							
343 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 521208-521215)							
344 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 521300-521307)							
345 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 521308-521315)							

LBP_AlarmMsgs [DB2], channel range 3

Table 10-6 Channel range 3

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 3							
	Feedrate disable (event no.: 530000-530015)							
346 (FM)	530007	530006	530005	530004	530003	530002	530001	530000
347 (OM)	530015	530014	530013	530012	530011	530010	530009	530008
348 (FM)	Feedrate and read-in disable, byte 1 (event no.: 530100-530107)							
349 (FM)	Feedrate and read-in disable, byte 2 (event no.: 530108-530115)							
350 (OM)	Feedrate and read-in disable, byte 2 (event no.: 530108-530115)							
351 (OM)	Feedrate and read-in disable, byte 4 (event no.: 530124-530131)							
352 (FM)	Read-in disable, byte 1 (event no.: 530200-530207)							
353 (FM)	Read-in disable, byte 2 (event no.: 530208-530215)							
354 (OM)	Read-in disable, byte 3 (event no.: 530216-530223)							
355 (OM)	Read-in disable, byte 4 (event no.: 530224-530231)							
356 (FM)	NC start disable, byte 1 (event no.: 530300-530307)							
357 (OM)	NC start disable, byte 2 (event no.: 530308-530315)							
358 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 531100-531107)							
359 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 531108-531115)							
360 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 531200-531207)							
361 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 531208-531215)							
362 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 531300-531307)							
363 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 531308-531315)							

LBP_AlarmMsgs [DB2], channel range 4

Table 10-7 Channel range 4

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 4							
	Feedrate disable (event no.: 540000-540015)							
364 (FM)	540007	540006	540005	540004	540003	540002	540001	540000
365 (OM)	540015	540014	540013	540012	540011	540010	540009	540008
366 (FM)	Feedrate and read-in disable, byte 1 (event no.: 540100-540107)							
367 (FM)	Feedrate and read-in disable, byte 2 (event no.: 540108-540115)							

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
368 (OM)	Feedrate and read-in disable, byte 3 (event no.: 540116-540123)							
369 (OM)	Feedrate and read-in disable, byte 4 (event no.: 540124-540131)							
370 (FM)	Read-in disable, byte 1 (event no.: 540200-540207)							
371 (FM)	Read-in disable, byte 2 (event no.: 540208-540215)							
372 (OM)	Read-in disable, byte 3 (event no.: 540216-540223)							
373 (OM)	Read-in disable, byte 4 (event no.: 540224-540231)							
374 (FM)	NC start disable, byte 1 (event no.: 540300-540307)							
375 (OM)	NC start disable, byte 2 (event no.: 540308-540315)							
376 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 541100-541107)							
377 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 541108-541115)							
378 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 541200-541207)							
379 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 541208-541215)							
380 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 541300-541307)							
381 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 541308-541315)							

LBP_AlarmMsgs [DB2], channel range 5

Table 10-8 Channel range 5

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Channel 5							
	Feedrate disable (event no.: 550000-550015)							
382 (FM)	550007	550006	550005	550004	550003	550002	550001	550000
383 (OM)	550015	550014	550013	550012	550011	550010	550009	550008
384 (FM)	Feedrate and read-in disable, byte 1 (event no.: 550100-550107)							
385 (FM)	Feedrate and read-in disable, byte 2 (event no.: 550108-550115)							
386 (OM)	Feedrate and read-in disable, byte 3 (event no.: 550116-550123)							
387 (OM)	Feedrate and read-in disable, byte 4 (event no.: 550124-550131)							
388 (FM)	Read-in disable, byte 1 (event no.: 550200-550207)							
389 (FM)	Read-in disable, byte 2 (event no.: 550208-550215)							
390 (OM)	Read-in disable, byte 3 (event no.: 550216-550223)							
391 (OM)	Read-in disable, byte 4 (event no.: 550224-550231)							
392 (FM)	NC start disable, byte 1 (event no.: 550300-550307)							
393 (OM)	NC start disable, byte 2 (event no.: 550308-550315)							
394 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 551100-551107)							
395 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 551108-551115)							

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
396 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 551200-551207)							
397 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 551208-551215)							
398 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 551300-551307)							
399 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 551308-551315)							

LBP_AlarmMsgs [DB2], channel range 6

Table 10-9 Channel range 6

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Channel 6							
	Feedrate disable (event no.: 560000-560015)							
400 (FM)	560007	560006	560005	560004	560003	560002	560001	560000
401 (OM)	560015	560014	560013	560012	560011	560010	560009	560008
402 (FM)	Feedrate and read-in disable, byte 1 (event no.: 560100-560107)							
403 (FM)	Feedrate and read-in disable, byte 2 (event no.: 560108-560115)							
404 (OM)	Feedrate and read-in disable, byte 3 (event no.: 560116-560123)							
405 (OM)	Feedrate and read-in disable, byte 4 (event no.: 560124-560131)							
406 (FM)	Read-in disable, byte 1 (event no.: 560200-560207)							
407 (FM)	Read-in disable, byte 2 (event no.: 560208-560215)							
408 (OM)	Read-in disable, byte 3 (event no.: 560216-560223)							
409 (OM)	Read-in disable, byte 4 (event no.: 560224-560231)							
410 (FM)	NC start disable, byte 1 (event no.: 560300-560307)							
411 (OM)	NC start disable, byte 2 (event no.: 560308-560315)							
412 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 561100-561107)							
413 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 561108-561115)							
414 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 561200-561207)							
415 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 561208-561215)							
416 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 561300-561307)							
417 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 561308-561315)							

LBP_AlarmMsgs [DB2], channel range 7

Table 10-10 Channel range 7

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 7							
	Feedrate disable (event no.: 570000-570015)							
418 (FM)	570007	570006	570005	570004	570003	570002	570001	570000
419 (OM)	570015	570014	570013	570012	570011	570010	570009	570008
420 (FM)	Feedrate and read-in disable, byte 1 (event no.: 570100-570107)							
421 (FM)	Feedrate and read-in disable, byte 2 (event no.: 570108-570115)							
422 (OM)	Feedrate and read-in disable, byte 3 (event no.: 570116-570123)							
423 (OM)	Feedrate and read-in disable, byte 4 (event no.: 570124-570131)							
424 (FM)	Read-in disable, byte 1 (event no.: 570200-570207)							
425 (FM)	Read-in disable, byte 2 (event no.: 570208-570215)							
426 (OM)	Read-in disable, byte 3 (event no.: 570216-570223)							
427 (OM)	Read-in disable, byte 4 (event no.: 570224-570231)							
428 (FM)	NC start disable, byte 1 (event no.: 570300-570307)							
429 (OM)	NC start disable, byte 2 (event no.: 570308-570315)							
430 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 571100-571107)							
431 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 571108-571115)							
432 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 571200-571207)							
433 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 571208-571215)							
434 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 571300-571307)							
435 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 571308-571315)							

LBP_AlarmMsgs [DB2], channel range 8

Table 10-11 Channel range 8

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Channel 8							
	Feedrate disable (event no.: 580000-580015)							
436 (FM)	580007	580006	580005	580004	580003	580002	580001	580000
437 (OM)	580015	580014	580013	580012	580011	580010	580009	580008
438 (FM)	Feedrate and read-in disable, byte 1 (event no.: 580100-580107)							
439 (FM)	Feedrate and read-in disable, byte 2 (event no.: 580108-580115)							

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
440 (OM)	Feedrate and read-in disable, byte 3 (event no.: 580116-580123)							
441 (OM)	Feedrate and read-in disable, byte 4 (event no.: 580124-580131)							
442 (FM)	Read-in disable, byte 1 (event no.: 580200-580207)							
443 (FM)	Read-in disable, byte 2 (event no.: 580208-580215)							
444 (OM)	Read-in disable, byte 3 (event no.: 580216-580223)							
445 (OM)	Read-in disable, byte 4 (event no.: 580224-580231)							
446 (FM)	NC start disable, byte 1 (event no.: 580300-580307)							
447 (OM)	NC start disable, byte 2 (event no.: 580308-580315)							
448 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 581100-581107)							
449 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 581108-581115)							
450 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 581200-581207)							
451 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 581208-581215)							
452 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 581300-581307)							
453 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 581308-581315)							

LBP_AlarmMsgs [DB2], channel range 9

Table 10-12 Channel range 9

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Channel 9							
	Feedrate disable (event no.: 590000-590015)							
454 (FM)	590007	590006	590005	590004	590003	590002	590001	590000
455 (OM)	590015	590014	590013	590012	590011	590010	590009	590008
456 (FM)	Feedrate and read-in disable, byte 1 (event no.: 590100-590107)							
457 (FM)	Feedrate and read-in disable, byte 2 (event no.: 590108-590115)							
458 (OM)	Feedrate and read-in disable, byte 3 (event no.: 590116-590123)							
459 (OM)	Feedrate and read-in disable, byte 4 (event no.: 590124-590131)							
460 (FM)	Read-in disable, byte 1 (event no.: 590200-590207)							
461 (FM)	Read-in disable, byte 2 (event no.: 590208-590215)							
462 (OM)	Read-in disable, byte 3 (event no.: 590216-590223)							
463 (OM)	Read-in disable, byte 4 (event no.: 590224-590231)							
464 (FM)	NC start disable, byte 1 (event no.: 590300-590307)							
465 (OM)	NC start disable, byte 2 (event no.: 590308-590315)							
466 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 591100-591107)							
467 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 591108-591115)							

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
468 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 591200-591207)							
469 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 591208-591215)							
470 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 591300-591307)							
471 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 591308-591315)							

LBP_AlarmMsgs [DB2], channel range 10

Table 10-13 Channel range 10

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Channel 10							
	Feedrate disable (event no.: 500000-500015)							
472 (FM)	500007	500006	500005	500004	500003	500002	500001	500000
473 (OM)	500015	500014	500013	500012	500011	500010	500009	500008
474 (FM)	Feedrate and read-in disable, byte 1 (event no.: 500100-500107)							
475 (FM)	Feedrate and read-in disable, byte 2 (event no.: 500108-500115)							
476 (OM)	Feedrate and read-in disable, byte 3 (event no.: 500116-500123)							
477 (OM)	Feedrate and read-in disable, byte 4 (event no.: 500124-500131)							
478 (FM)	Read-in disable, byte 1 (event no.: 500200-500207)							
479 (FM)	Read-in disable, byte 2 (event no.: 500208-500215)							
480 (OM)	Read-in disable, byte 3 (event no.: 500216-500223)							
481 (OM)	Read-in disable, byte 4 (event no.: 500224-500231)							
482 (FM)	NC start disable, byte 1 (event no.: 500300-500307)							
483 (OM)	NC start disable, byte 2 (event no.: 500308-500315)							
484 (FM)	Feedrate stop, geo axis 1, byte 1 (event no.: 501100-501107)							
485 (OM)	Feedrate stop, geo axis 1, byte 2 (event no.: 501108-501115)							
486 (FM)	Feedrate stop, geo axis 2, byte 1 (event no.: 501200-501207)							
487 (OM)	Feedrate stop, geo axis 2, byte 2 (event no.: 501208-501215)							
488 (FM)	Feedrate stop, geo axis 3, byte 1 (event no.: 501300-501307)							
489 (OM)	Feedrate stop, geo axis 3, byte 2 (event no.: 501308-501315)							

LBP_AlarmMsgs [DB2], axis ranges

Table 10-14 Axis ranges

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Axis/spindle							
	Feedrate stop / spindle stop for axis/spindle 1 (event no.: 600100-600115)							
490 (FM)	600107	600106	600105	600104	600103	600102	600101	600100
491 (OM)	600115	600114	600113	600112	600111	600110	600109	600108
492 (FM)	Feedrate stop/spindle stop for axis/spindle 2 (event no.: 600200-600207)							
493 (OM)	Feedrate stop/spindle stop for axis/spindle 2 (event no.: 600208-600215)							
494 (FM)	Feedrate stop/spindle stop for axis/spindle 3 (event no.: 600300-600307)							
495 (OM)	Feedrate stop/spindle stop for axis/spindle 3 (event no.: 600308-600315)							
496 (FM)	Feedrate stop/spindle stop for axis/spindle 4 (event no.: 600400-600407)							
497 (OM)	Feedrate stop/spindle stop for axis/spindle 4 (event no.: 600408-600415)							
498 (FM)	Feedrate stop/spindle stop for axis/spindle 5 (event no.: 600500-600507)							
499 (OM)	Feedrate stop/spindle stop for axis/spindle 5 (event no.: 600508-600515)							
500 (FM)	Feedrate stop/spindle stop for axis/spindle 6 (event no.: 600600-600607)							
501 (OM)	Feedrate stop/spindle stop for axis/spindle 6 (event no.: 600608-600615)							
502 (FM)	Feedrate stop/spindle stop for axis/spindle 7 (event no.: 600700-600707)							
503 (OM)	Feedrate stop/spindle stop for axis/spindle 7 (event no.: 600708-600715)							
504 (FM)	Feedrate stop/spindle stop for axis/spindle 8 (event no.: 600800-600807)							
505 (OM)	Feedrate stop/spindle stop for axis/spindle 8 (event no.: 600808-600815)							
506 (FM)	Feedrate stop/spindle stop for axis/spindle 9 (event no.: 600900-600907)							
507 (OM)	Feedrate stop/spindle stop for axis/spindle 9 (event no.: 600908-600915)							
508 (FM)	Feedrate stop/spindle stop for axis/spindle 10 (event no.: 601000-601007)							
509 (OM)	Feedrate stop/spindle stop for axis/spindle 10 (event no.: 601008-601015)							
510 (FM)	Feedrate stop/spindle stop for axis/spindle 11 (event no.: 601100-601107)							
511 (OM)	Feedrate stop/spindle stop for axis/spindle 11 (event no.: 601108-601115)							
512 (FM)	Feedrate stop/spindle stop for axis/spindle 12 (event no.: 601200-601207)							
513 (OM)	Feedrate stop/spindle stop for axis/spindle 12 (event no.: 601208-601215)							
514 (FM)	Feedrate stop/spindle stop for axis/spindle 13 (event no.: 601300-601307)							
515 (OM)	Feedrate stop/spindle stop for axis/spindle 13 (event no.: 601308-601315)							
516 (FM)	Feedrate stop/spindle stop for axis/spindle 14 (event no.: 601400-601407)							
517 (OM)	Feedrate stop/spindle stop for axis/spindle 14 (event no.: 601408-601415)							
518 (FM)	Feedrate stop/spindle stop for axis/spindle 15 (event no.: 601500-601507)							
519 (OM)	Feedrate stop/spindle stop for axis/spindle 15 (event no.: 601508-601515)							
520 (FM)	Feedrate stop/spindle stop for axis/spindle 16 (event no.: 601600-601607)							
521 (OM)	Feedrate stop/spindle stop for axis/spindle 16 (event no.: 601608-601615)							
522 (FM)	Feedrate stop/spindle stop for axis/spindle 17 (event no.: 601700-601707)							
523 (OM)	Feedrate stop/spindle stop for axis/spindle 17 (event no.: 601708-601715)							

LBP_Alarm Msgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
524 (FM)	Feedrate stop/spindle stop for axis/spindle 18 (event no.: 601800-601807)							
525 (OM)	Feedrate stop/spindle stop for axis/spindle 18 (event no.: 601808-601815)							
526 (FM)	Feedrate stop/spindle stop for axis/spindle 19 (event no.: 601900-601907)							
527 (OM)	Feedrate stop/spindle stop for axis/spindle 19 (event no.: 601908-601915)							
528 (FM)	Feedrate stop/spindle stop for axis/spindle 20 (event no.: 602000-602007)							
529 (OM)	Feedrate stop/spindle stop for axis/spindle 20 (event no.: 602008-602015)							
530 (FM)	Feedrate stop/spindle stop for axis/spindle 21 (event no.: 602100-602107)							
531 (OM)	Feedrate stop/spindle stop for axis/spindle 21 (event no.: 602108-602115)							
532 (FM)	Feedrate stop/spindle stop for axis/spindle 22 (event no.: 602200-602207)							
533 (OM)	Feedrate stop/spindle stop for axis/spindle 22 (event no.: 602208-602215)							
534 (FM)	Feedrate stop/spindle stop for axis/spindle 23 (event no.: 602300-602307)							
535 (OM)	Feedrate stop/spindle stop for axis/spindle 23 (event no.: 602308-602315)							
536 (FM)	Feedrate stop/spindle stop for axis/spindle 24 (event no.: 602400-602407)							
537 (OM)	Feedrate stop/spindle stop for axis/spindle 24 (event no.: 602408-602415)							
538 (FM)	Feedrate stop/spindle stop for axis/spindle 25 (event no.: 602500-602507)							
539 (OM)	Feedrate stop/spindle stop for axis/spindle 25 (event no.: 602508-602515)							
540 (FM)	Feedrate stop/spindle stop for axis/spindle 26 (event no.: 602600-602607)							
541 (OM)	Feedrate stop/spindle stop for axis/spindle 26 (event no.: 602608-602615)							
542 (FM)	Feedrate stop/spindle stop for axis/spindle 27 (event no.: 602700-602707)							
543 (OM)	Feedrate stop/spindle stop for axis/spindle 27 (event no.: 602708-602715)							
544 (FM)	Feedrate stop/spindle stop for axis/spindle 28 (event no.: 602800-602807)							
545 (OM)	Feedrate stop/spindle stop for axis/spindle 28 (event no.: 602808-602815)							
546 (FM)	Feedrate stop/spindle stop for axis/spindle 29 (event no.: 602900-602907)							
547 (OM)	Feedrate stop/spindle stop for axis/spindle 29 (event no.: 602908-602915)							
548 (FM)	Feedrate stop/spindle stop for axis/spindle 30 (event no.: 603000-603007)							
549 (OM)	Feedrate stop/spindle stop for axis/spindle 30 (event no.: 603008-603015)							
550 (FM)	Feedrate stop/spindle stop for axis/spindle 31 (event no.: 603100-603107)							
551 (OM)	Feedrate stop/spindle stop for axis/spindle 31 (event no.: 603108-603115)							

LBP_AlarmMsgs [DB2], user ranges

Table 10-15 User ranges

LBP_AlarmMsgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	User ranges							
	User range 0 (event no.: 700000-700015)							

LBP_AlarmMsgs [DB2]	**Signals for PLC events (PLC → HMI)**							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
554 (FM)	700007	700006	700005	700004	700003	700002	700001	700000
555 (FM)	700015	700014	700013	700012	700011	700010	700009	700008
556 (FM)	User range 0: Byte 3 (event no.: 700016-700023)							
557 (FM)	User range 0: Byte 4 (event no.: 700024-700031)							
558 (OM)	User range 0: Byte 5 (event no.: 700032-700039)							
559 (OM)	User range 0: Byte 6 (event no.: 700040-700047)							
560 (OM)	User range 0: Byte 7 (event no.: 700048-700055)							
561 (OM)	User range 0: Byte 8 (event no.: 700056-700063)							
562 - 565 (FM)	User range 1: Byte 1 - 4 (event no.: 700100-700131)							
566 - 569 (OM)	User range 1: Byte 5 - 8 (event no.: 700132-700163)							
570 - 573 (FM)	User range 2: Byte 1 - 4 (event no.: 700200-700231)							
574 - 577 (OM)	User range 2: Byte 5 - 8 (event no.: 700232-700263)							
578 - 581 (FM)	User range 3: Byte 1 - 4 (event no.: 700300-700331)							
582 - 585 (OM)	User range 3: Byte 5 - 8 (event no.: 700332-700363)							
586 - 589 (FM)	User range 4: Byte 1 - 4 (event no.: 700400-700431)							
590 - 593 (OM)	User range 4: Byte 5 - 8 (event no.: 700432-700463)							
594 - 597 (FM)	User range 5: Byte 1 - 4 (event no.: 700500-700531)							
598 - 601 (OM)	User range 5: Byte 5 - 8 (event no.: 700532-700563)							
602 - 605 (FM)	User range 6: Byte 1 - 4 (event no.: 700600-700631)							
606 - 609 (OM)	User range 6: Byte 5 - 8 (event no.: 700632-700663)							
610 - 613 (FM)	User range 7: Byte 1 - 4 (event no.: 700700-700731)							
614 - 617 (OM)	User range 7: Byte 5 - 8 (event no.: 700732-700763)							
618 - 621 (FM)	User range 8: Byte 1 - 4 (event no.: 700800-700831)							
622 - 625 (OM)	User range 8: Byte 5 - 8 (event no.: 700832-700863)							
626 - 629 (FM)	User range 9: Byte 1 - 4 (event no.: 700900-700931)							
630 - 633 (OM)	User range 9: Byte 5 - 8 (event no.: 700932-700963)							
634 - 637 (FM)	User range 10: Byte 1 - 4 (event no.: 701000-701031)							
638 - 641 (OM)	User range 10: Byte 5 - 8 (event no.: 701032-701063)							
642 - 645 (FM)	User range 11: Byte 1 - 4 (event no.: 701100-701131)							
646 - 649 (OM)	User range 11: Byte 5 - 8 (event no.: 701132-701163)							
650 - 653 (FM)	User range 12: Byte 1 - 4 (event no.: 701200-701231)							
654 - 657 (OM)	User range 12: Byte 5 - 8 (event no.: 701232-701263)							
658 - 661 (FM)	User range 13: Byte 1 - 4 (event no.: 701300-701331)							
662 - 665 (OM)	User range 13: Byte 5 - 8 (event no.: 701332-701363)							
666 - 669 (FM)	User range 14: Byte 1 - 4 (event no.: 701400-701431)							
670 - 673 (OM)	User range 14: Byte 5 - 8 (event no.: 701432-701463)							
674 - 677 (FM)	User range 15: Byte 1 - 4 (event no.: 701500-701531)							
678 - 681 (OM)	User range 15: Byte 5 - 8 (event no.: 701532-701563)							
682 - 685 (FM)	User range 16: Byte 1 - 4 (event no.: 701600-701631)							
686 - 689 (OM)	User range 16: Byte 5 - 8 (event no.: 701632-701663)							

LBP_AlarmMsgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
690 - 693 (FM)	User range 17: Byte 1 - 4 (event no.: 701700-701731)							
694 - 697 (OM)	User range 17: Byte 5 - 8 (event no.: 701732-701763)							
698 - 701 (FM)	User range 18: Byte 1 - 4 (event no.: 701800-701831)							
702 - 705 (OM)	User range 18: Byte 5 - 8 (event no.: 701832-701863)							
706 - 709 (FM)	User range 19: Byte 1 - 4 (event no.: 701900-701931)							
710 - 713 (OM)	User range 19: Byte 5 - 8 (event no.: 701932-701963)							
714 - 717 (FM)	User range 20: Byte 1 - 4 (event no.: 702000-702031)							
718 - 721 (OM)	User range 20: Byte 5 - 8 (event no.: 702032-702063)							
722 - 725 (FM)	User range 21: Byte 1 - 4 (event no.: 702100-702131)							
726 - 729 (OM)	User range 21: Byte 5 - 8 (event no.: 702132-702163)							
730 – 733 (FM)	User range 22: Byte 1 - 4 (event no.: 702200-702231)							
734 - 737 (OM)	User range 22: Byte 5 - 8 (event no.: 702232-702263)							
738 - 741 (FM)	User range 23: Byte 1 - 4 (event no.: 702300-702331)							
742 - 745 (OM)	User range 23: Byte 5 - 8 (event no.: 702332-702363)							
746 - 749 (FM)	User range 24: Byte 1 - 4 (event no.: 702400-702431)							
750 - 753 (OM)	User range 24: Byte 5 - 8 (event no.: 702432-702463)							
754 - 757 (FM)	User range 25: Byte 1 - 4 (event no.: 702500-702531)							
758 - 761 (OM)	User range 25: Byte 5 - 8 (event no.: 702532-702563)							
762 - 765 (FM)	User range 26: Byte 1 - 4 (event no.: 702600-702631)							
766 - 769 (OM)	User range 26: Byte 5 - 8 (event no.: 702632-702663)							
770 - 773 (FM)	User range 27: Byte 1 - 4 (event no.: 702700-702731)							
774 - 777 (OM)	User range 27: Byte 5 - 8 (event no.: 702732-702763)							
778 - 781 (FM)	User range 28: Byte 1 - 4 (event no.: 702800-702831)							
782 - 785 (OM)	User range 28: Byte 5 - 8 (event no.: 702832-702863)							
786 - 789 (FM)	User range 29: Byte 1 - 4 (event no.: 702900-702931)							
790 - 793 (OM)	User range 29: Byte 5 - 8 (event no.: 702932-702963)							
794 - 797 (FM)	User range 30: Byte 1 - 4 (event no.: 703000-703031)							
798 - 801 (OM)	User range 30: Byte 5 - 8 (event no.: 703032-703063)							
802 - 805 (FM)	User range 31: Byte 1 - 4 (event no.: 703100-703131)							
806 - 809 (OM)	User range 31: Byte 5 - 8 (event no.: 703132-703163)							
810 - 813 (FM)	User area 32: Byte 1 - 4 (event no.: 703200 - 703231)							
814 - 817 (BM)	User area 32: Byte 5 - 8 (event no.: 703232 - 703263)							
818 - 821 (FM)	User area 33: Byte 1 - 4 (event no.: 703300 - 703331)							
822 - 825 (OM)	User area 33: Byte 5 - 8 (event no.: 703332 - 703363)							
826 - 829 (FM)	User area 34: Byte 1 - 4 (event no.: 703400 - 703431)							
830 - 833 (OM)	User area 34: Byte 5 - 8 (event no.: 703432 - 703463)							
834 - 837 (FM)	User area 35: Byte 1 - 4 (event no.: 703500 - 703531)							
838 - 841 (OM)	User area 35: Byte 5 - 8 (event no.: 703532 - 703563)							
842 - 845 (FM)	User area 36: Byte 1 - 4 (event no.: 703600 - 703631)							
846 - 789 (OM)	User area 36: Byte 5 - 8 (event no.: 703632 - 703663)							

LBP_AlarmMsgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
850 - 853 (FM)	User area 37: Byte 1 - 4 (event no.: 703700 - 703731)							
854 - 857 (OM)	User area 37: Byte 5 - 8 (event no.: 703732 - 703763)							
858 - 861 (FM)	User area 38: Byte 5 - 8 (event no.: 703800 -703831)							
862 - 865 (OM)	User area 38: Byte 1 - 4 (event no.: 703832 -703863)							
866 - 869 (FM)	User area 39: Byte 5 - 8 (event no.: 703900 -703931)							
870 - 873 (OM)	User area 39: Byte 1 - 4 (event no.: 703932 -703963)							
874 - 877 (FM)	User area 40: Byte 1 - 4 (event no.: 704000-704031)							
878 - 881 (OM)	User area 40: Byte 5 - 8 (event no.: 704032-704063)							
882 - 885 (FM)	User area 41: Byte 1 - 4 (event no.: 704100-704131)							
886 - 889 (OM)	User area 41: Byte 5 - 8 (event no.: 704132-704163)							
890 - 893 (FM)	User area 42: Byte 1 - 4 (event no.: 704200-704231)							
894 - 897 (OM)	User area 42: Byte 5 - 8 (event no.: 704232-704263)							
898 - 901 (FM)	User area 43: Byte 1 - 4 (event no.: 704300-704331)							
902 - 905 (OM)	User area 43: Byte 5 - 8 (event no.: 704332-704363)							
906 - 909 (FM)	User area 44: Byte 1 - 4 (event no.: 704400-704431)							
910 - 913 (OM)	User area 44: Byte 5 - 8 (event no.: 704432-704463)							
914 - 917 (FM)	User area 45: Byte 1 - 4 (event no.: 704500-704531)							
918 - 921 (OM)	User area 45: Byte 5 - 8 (event no.: 704532-704563)							
922 - 925 (FM)	User area 46: Byte 1 - 4 (event no.: 704600-704631)							
926 - 929 (OM)	User area 46: Byte 5 - 8 (event no.: 704632-704663)							
930 - 933 (FM)	User area 47: Byte 1 - 4 (event no.: 704700-704731)							
934 - 937 (OM)	User area 47: Byte 5 - 8 (event no.: 704732-704763)							
938 - 941 (FM)	User area 48: Byte 1 - 4 (event no.: 704800-704831)							
942 - 945 (OM)	User area 48: Byte 5 - 8 (event no.: 704832-704863)							
946 - 989 (FM)	User area 49: Byte 1 - 4 (event no.: 704900-704931)							
950 - 953 (OM)	User area 49: Byte 5 - 8 (event no.: 704932-704963)							
954 - 957 (FM)	User area 50: Byte 1 - 4 (event no.: 705000-705031)							
958 - 961 (OM)	User area 50: Byte 5 - 8 (event no.: 705032-705063)							
962 -965 (FM)	User area 51: Byte 1 - 4 (event no.: 705100-705131)							
966 - 969 (OM)	User area 51: Byte 5 - 8 (event no.: 705132-705163)							
970 - 973 (FM)	User area 52: Byte 1 - 4 (event no.: 705200-705231)							
974 - 977 (OM)	User area 52: Byte 5 - 8 (event no.: 705232-705263)							
978 - 981 (FM)	User area 53: Byte 1 - 4 (event no.: 705300-705331)							
982 - 985 (OM)	User area 53: Byte 5 - 8 (event no.: 705332-705363)							
986 - 989 (FM)	User area 54: Byte 1 - 4 (event no.: 705400-705431)							
990 - 993 (OM)	User area 54: Byte 5 - 8 (event no.: 705432-705463)							
994 - 997 (FM)	User area 55: Byte 1 - 4 (event no.: 705500-705531)							
998 - 1001 (OM)	User area 55: Byte 5 - 8 (event no.: 705532-705563)							
1002 - 1005 (FM)	User area 56: Byte 1 - 4 (event no.: 705600-705631)							

LBP_AlarmMsgs [DB2]	Signals for PLC events (PLC → HMI)							
Byte (message type)	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
1006 - 1009 (OM)	User area 56: Byte 5 - 8 (event no.: 705632-705663)							
1010 -1013 (FM)	User area 57: Byte 1 - 4 (event no.: 705700-705731)							
1014 - 1017 (OM)	User area 57: Byte 5 - 8 (event no.: 705732-705763)							
1018 - 1021 (FM)	User area 58: Byte 1 - 4 (event no.: 705800-705831)							
1022 - 1025 (OM)	User area 58: Byte 5 - 8 (event no.: 705832-705863)							
1026 - 1029 (FM)	User area 59: Byte 1 - 4 (event no.: 705900-705931)							
1030 - 1033 (OM)	User area 59: Byte 5 - 8 (event no.: 705932-705963)							
1034 - 1037 (FM)	User area 60: Byte 1 - 4 (event no.: 706000-706031)							
1038 - 941 (OM)	User area 60: Byte 5 - 8 (event no.: 706032-706063)							
1042 - 1045 (FM)	User area 61: Byte 1 - 4 (event no.: 706100-706131)							
1046 - 1089 (OM)	User area 61: Byte 5 - 8 (event no.: 706132-706163)							
1050 - 1053 (FM)	User area 62: Byte 1 - 4 (event no.: 706200-706231)							
1054 - 1057 (OM)	User area 62: Byte 5 - 8 (event no.: 706232-706263)							
1058 - 1061 (FM)	User area 63: Byte 1 - 4 (event no.: 706300-706331)							
1062 -1065 (BM)	User area 63: Byte 5 - 8 (event no.: 706332-706363)							

10.4.3.2 LBP_AlarmMsgs [DB2]

Signals for PLC alarms that are sent to an HMI are stored in "LBP_AlarmMsgs [DB2]".

Variables in "LBP_AlarmMsgs [DB2]"

Chan (DB2 from DBB0)

Description	Alarms of a channel without a message
Data type	ARRAY[1..10] of STRUCT STRUCT "Chan" alarms Channel without message (Page 113)

Axis (DB2 from DBB120)

Description	Alarms of an axis without a message
Data type	ARRAY[1..31] of STRUCT STRUCT "Axis" alarms Axis without message (Page 114)

AddValUserA0 (DB2 ab DBB182)

Description	Additional values for user area 0 Index 0 ... 63 corresponds to alarm number 700000 ... 700063
Data type	ARRAY[0..63] of INT

ChanA (DB2 from DBB310)

Description	Alarms of a channel with a message Channel 1 corresponds to alarm number from 5**1**0000 Channel 2 corresponds to alarm number from 5**2**0000 ... Channel 9 corresponds to alarm number from 5**9**0000 Channel 10 corresponds to alarm number from 5**0**0000
Data type	ARRAY[1..10] of STRUCT STRUCT "ChanA" alarms Channel with message (Page 114)

AxisA (DB2 from DBB490)

Description	Alarms of an axis/spindle with message Axis/spindle 1 corresponds to alarm number from 600100 Axis/spindle 2 corresponds to alarm number from 600200 ... Axis/spindle 31 corresponds to alarm number from 603100
Data type	ARRAY[1..31] of STRUCT STRUCT "AxisA" alarms Axis with message (Page 116)

UserA (DB2 from DBB554)

Description	User area for alarms with message User area 0 corresponds to alarm number from 700000 User area 1 corresponds to alarm number from 700100 ... User area 63 corresponds to alarm number from 706300
Data type	ARRAY[0..63] of STRUCT STRUCT "UserA" alarms User area with message (Page 117)

10.4.3.3 STRUCT "Chan" alarms Channel without message

Variable structure "Chan" describes the signals that trigger a follow-up response in the channel. The signals do not trigger any error or system status messages.

Variables in "Chan"

FDD (DB2 DBW0, 12, ..., 108)

Description	Feed disable in channel
Data type	ARRAY[0..15] of BOOL

RID (DB2 DBW2, 14, ..., 110)

Description	Read-in disable in the channel
Data type	ARRAY[0..15] of BOOL

NCSD (DB2 DBW4, 16, ..., 112)

Description	Start disable in the channel
Data type	ARRAY[0..15] of BOOL

FdStop_1 (DB2 DBW6, 18, ..., 114)

Description	Feed stop geometry axis 1 in the channel
Data type	ARRAY[0..15] of BOOL

FdStop_2 (DB2 DBW8, 20, ..., 116)

Description	Feed stop geometry axis 2 in the channel
Data type	ARRAY[0..15] of BOOL

FdStop_3 (DB2 DBW10, 22, ..., 118)

Description	Feed stop geometry axis 3 in the channel
Data type	ARRAY[0..15] of BOOL

See also

LBP_AlarmMsgs [DB2] (Page 112)

10.4.3.4 STRUCT "Axis" alarms Axis without message

Variable structure "Axis" describes the signals that trigger a follow-up response in the channel. The signals do not trigger any error or system status messages.

Variables in "Axis"

FdStop (DB2 DBW120, 122, ..., 180)

Description	Feedrate / spindle stop
Data type	ARRAY[0..15] of BOOL

See also

LBP_AlarmMsgs [DB2] (Page 112)

10.4.3.5 STRUCT "ChanA" alarms Channel with message

Variable structure "ChanA" describes the alarm messages of a channel. Error messages or system status messages are displayed for the alarms.

The alarm numbers are shown as examples for Channel 1. (channel 1 from 5**1**0000, channel 2 from 5**2**0000, ..., channel 9 from 5**9**0000, channel 10 from 5**0**0000)

The variable structure is used in an array starting with index 1, where the array index corresponds to the channel number.

A message and optional follow-up response is assigned to each of the bits in the variable arrays described below. The symbol name is the abbreviation of the response performed, e.g. "FDD_RID..." for feed and read-in disable.

The displayed alarm number is also suggested in the symbol name. E.g. "LBP_AlarmMsgs.ChanA[4].FDD_RID_5C01xx[2]" (DB2.DBX366.2) triggers message number 540102 and results in the follow-up response feed and read-in disable in channel 4.

Variables in "ChanA"

FDD_5C00xx (DB2 DBB310, 328, ..., 472)

Description	Feed disable in channel Index 0 ... 7 corresponds to alarm number 510000 ... 510007, default message type FM (Page 98) Index 8 ... 15 corresponds to alarm number 510008 ... 510015, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..15] of BOOL

FDD_RID_5C01xx (DB2 DBD312, 330, ..., 474)

Description	Feed and read-in disable in the channel Index 0 ... 15 corresponds to alarm number 510100 ... 510115, default message type FM (Page 98) Index 16 ... 31 corresponds to alarm number 510116 ... 510131, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..31] of BOOL

RID_5C02xx (DB2 DBD316, 334, ..., 478)

Description	Read-in disable in the channel Index 0 ... 15 corresponds to alarm number 510200 ... 510215, default message type FM (Page 98) Index 16 ... 31 corresponds to alarm number 510216 ... 510231, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..31] of BOOL

NCSD_5C03xx (DB2 DBW320, 338, ..., 482)

Description	Start disable in the channel Index 0 ... 7 corresponds to alarm number 510300 ... 510307, default message type FM (Page 98) Index 8 ... 15 corresponds to alarm number 510308 ... 510315, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..15] of BOOL

FdStop_1_5C11xx (DB2 DBW322, 340, ..., 484)

Description	Feed stop geometry axis 1 in the channel Index 0 ... 7 corresponds to alarm number 511100 ... 511107, default message type FM (Page 98) Index 8 ... 15 corresponds to alarm number 511108 ... 511115, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..15] of BOOL

FdStop_2_5C12xx (DB2 DBW324, 342, ..., 486)

Description	Feed stop geometry axis 2 in the channel Index 0 ... 7 corresponds to alarm number 511200 ... 511207, default message type FM (Page 98) Index 8 ... 15 corresponds to alarm number 511208 ... 511215, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..15] of BOOL

FdStop_3_5C13xx (DB2 DBW326, 344, ..., 488)

Description	Feed stop geometry axis 3 in the channel Index 0 ... 7 corresponds to alarm number 511300 ... 511307, default message type FM (Page 98) Index 8 ... 15 corresponds to alarm number 511308 ... 511315, default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..15] of BOOL

See also

LBP_AlarmMsgs [DB2] (Page 112)

10.4.3.6 STRUCT "AxisA" alarms Axis with message

Variable structure "AxisA" describes the alarm messages of an axis/spindle. Error messages or system status messages are displayed for the alarms.

Variables in "AxisA"

AAFdStop60AAxx (DB2 DBW490, 492, ..., 550)

Description	Feed and spindle stop for axis/spindle Index 0 ... 7: Default message type FM (Page 98) Index 8 ... 15: Default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..15] of BOOL

10.4.3.7 STRUCT "UserA" alarms User area with message

Variable structure "UserA" describes the user area of the alarm messages. Error messages or system status messages are displayed for the alarms.

Variables in "UserA"

A70UUxx (DB2 ab DBB554, 562, ..., 1058)

Description	User area for alarms Index 0 ... 31: Default message type FM (Page 98) Index 32 ... 63: Default message type BM (Page 98) Parametrization of message type in LBP_ParamAlarmMsgs [DB5] (Page 117)
Data type	ARRAY[0..63] of BOOL

10.4.4 LBP_ParamAlarmMsgs [DB5] - message type in LBP_AlarmMsgs [DB2]

10.4.4.1 LBP_ParamAlarmMsgs [DB5]

In "LBP_ParamAlarmMsgs [DB5]", you parameterize the message type for alarm groups in "LBP_AlarmMsgs [DB2]". Depending on the parameter settings, the messages for TRUE are output as an error message (EM) or for FALSE as a system status message (SSM).

Variables in "LBP_ParamAlarmMsgs [DB5]" - Channels

Variable	Data type	Variables in "LBP_AlarmMsgs [DB2]"	Description
Channel 1			
Ch1_DB2_310 (DB5.DBX0.0)	BOOL	"LBP_AlarmMsgs.ChanA[1].FDD_5C00xx[i]", Index 0 ... 7 (DB2.DBB310)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM
Ch1_DB2_311 (DB5.DBX0.1)	BOOL	"LBP_AlarmMsgs.ChanA[1].FDD_5C00xx[i]", Index 8 ... 15 (DB2.DBB311)	See above
Ch1_DB2_312 (DB5.DBX0.2)	BOOL	"LBP_AlarmMsgs.ChanA[1].FDD_RID_5C01xx[i]", Index 0 ... 7 (DB2.DBB312)	See above
Ch1_DB2_313 (DB5.DBX0.3)	BOOL	"LBP_AlarmMsgs.ChanA[1].FDD_RID_5C01xx[i]", Index 8 ... 15 (DB2.DBB313)	See above
Ch1_DB2_314 (DB5.DBX0.4)	BOOL	"LBP_AlarmMsgs.ChanA[1].FDD_RID_5C01xx[i]", Index 16 ... 23 (DB2.DBB314)	See above
Ch1_DB2_315 (DB5.DBX0.5)	BOOL	"LBP_AlarmMsgs.ChanA[1].FDD_RID_5C01xx[i]", Index 24 ... 31 (DB2.DBB315)	See above
Ch1_DB2_316 (DB5.DBX0.6)	BOOL	"LBP_AlarmMsgs.ChanA[1].RID_5C02xx[i]", Index 0 ... 7 (DB2.DBB316)	See above
Ch1_DB2_317 (DB5.DBX0.7)	BOOL	"LBP_AlarmMsgs.ChanA[1].RID_5C02xx[i]", Index 8 ... 15 (DB2.DBB317)	See above
Ch1_DB2_318 (DB5.DBX1.0)	BOOL	"LBP_AlarmMsgs.ChanA[1].RID_5C02xx[i]", Index 16 ... 23 (DB2.DBB318)	See above
Ch1_DB2_319 (DB5.DBX1.1)	BOOL	"LBP_AlarmMsgs.ChanA[1].RID_5C02xx[i]", Index 24 ... 31 (DB2.DBB319)	See above
Ch1_DB2_320 (DB5.DBX1.2)	BOOL	"LBP_AlarmMsgs.ChanA[1].NCSD_5C03xx[i]", Index 0 ... 7 (DB2.DBB320)	See above
Ch1_DB2_321 (DB5.DBX1.3)	BOOL	"LBP_AlarmMsgs.ChanA[1].NCSD_5C03xx[i]", Index 8 ... 15 (DB2.DBB321)	See above
Ch1_DB2_322 (DB5.DBX1.4)	BOOL	"LBP_AlarmMsgs.ChanA[1].FdStop_1_5C11xx[i]", Index 0 ... 7 (DB2.DBB322)	See above
Ch1_DB2_323 (DB5.DBX1.5)	BOOL	"LBP_AlarmMsgs.ChanA[1].FdStop_1_5C11xx[i]", Index 8 ... 15 (DB2.DBB323)	See above
Ch1_DB2_324 (DB5.DBX1.6)	BOOL	"LBP_AlarmMsgs.ChanA[1].FdStop_2_5C12xx[i]", Index 0 ... 7 (DB2.DBB324)	See above
Ch1_DB2_325 (DB5.DBX1.7)	BOOL	"LBP_AlarmMsgs.ChanA[1].FdStop_2_5C12xx[i]", Index 8 ... 15 (DB2.DBB325)	See above
Ch1_DB2_326 (DB5.DBX2.0)	BOOL	"LBP_AlarmMsgs.ChanA[1].FdStop_3_5C13xx[i]", Index 0 ... 7 (DB2.DBB326)	See above
Ch1_DB2_327 (DB5.DBX2.1)	BOOL	"LBP_AlarmMsgs.ChanA[1].FdStop_3_5C13xx[i]", Index 8 ... 15 (DB2.DBB327)	See above
Channel 2			

Variable	Data type	Variables in "LBP_AlarmMsgs [DB2]"	Description
Ch2_DB2_328 (DB5.DBX2.2)	BOOL	"LBP_AlarmMsgs.ChanA[2].FDD_5C00xx[i]", Index 0 ... 7 (DB2.DBB328)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM
...	...		...
...			
Channel 10			
...	...		...
Ch10_DB2_489 (DB5.DBX22.3)	BOOL	"LBP_AlarmMsgs.ChanA[10].FdStop_3_5C13xx[i]", Index 8 ... 15 (DB2.DBB489)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM

Example for setting the message type for an alarm in channel 4

Searched alarms: "LBP_AlarmMsgs.ChanA[4].FDD_RID_5C01xx[i]", index 16 to 23

The byte offset for the alarms in "LBP_AlarmMsgs" is 368 (DB2.DBB368).

You can set the message type in the variable with the following symbol name: "LBP_ParamAlarmMsgs.Ch4_DB2_368" (DB5.DBX7.2)

Variables in"LBP_ParamAlarmMsgs [DB5]" - Axes and spindles

Variable	Data type	Variables in "LBP_AlarmMsgs [DB2]"	Description
Axis/spindle 1			
Ax1_DB2_490 (DB5.DBX22.4)	BOOL	"LBP_AlarmMsgs.AxisA[1].AAFdStop60AAxx[i]", Index 0 ... 7 (DB2.DBB490)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM
Ax1_DB2_491 (DB5.DBX22.5)	BOOL	"LBP_AlarmMsgs.AxisA[1].AAFdStop60AAxx[i]", Index 8 ... 15 (DB2.DBB491)	See above
Axis/Spindle 2			
Ax2_DB2_492 (DB5.DBX22.6)	BOOL	"LBP_AlarmMsgs.AxisA[2].AAFdStop60AAxx[i]", Index 0 ... 7 (DB2.DBB492)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM
...	...	...	...
Axis/Spindle 31			

Variable	Data type	Variables in "LBP_AlarmMsgs [DB2]"	Description
...	...	...	...
Ax31_DB2_551 (DB5.DBX30.1)	BOOL	"LBP_AlarmMsgs.AxisA[31].AAFdStop60AAxx[i]", Index 8 ... 15 (DB2.DBB551)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM

Example for setting the message type for an alarm in axis/spindle 4

Searched alarms: "LBP_AlarmMsgs.AxisA[4].AAFdStop60AAxx[i]", index 8 to 15

The byte offset for the alarms in "LBP_AlarmMsgs" is 497 (DB2.DBB497).

You can set the message type in the variable with the following symbol name: LBP_ParamAlarmMsgs.Ax4_DB2_497" (DB5.DBX23.3)

Variables in "LBP_ParamAlarmMsgs [DB5]" - User data

Variable	Data type	Variables in "LBP_AlarmMsgs [DB2]"	Description
User 0			
User0_DB2_554 (DB5.DBX30.4)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 0 ... 7 (DB2.DBB554)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM
User0_DB2_555 (DB5.DBX30.5)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 8 ... 15 (DB2.DBB555)	See above
User0_DB2_556 (DB5.DBX30.6)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 16 ... 23 (DB2.DBB556)	See above
User0_DB2_557 (DB5.DBX30.7)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 24 ... 31 (DB2.DBB557)	See above
User0_DB2_558 (DB5.DBX31.0)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 32 ... 39 (DB2.DBB558)	See above
User0_DB2_559 (DB5.DBX31.1)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 40 ... 47 (DB2.DBB559)	See above
User0_DB2_560 (DB5.DBX31.2)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 48 ... 55 (DB2.DBB560)	See above
User0_DB2_561 (DB5.DBX31.3)	BOOL	"LBP_AlarmMsgs.UserA[0].A70UUxx[i]", Index 56 ... 63 (DB2.DBB561)	See above
User 1			
User1_DB2_562 (DB5.DBX31.4)	BOOL	"LBP_AlarmMsgs.UserA[1].A70UUxx[i]", Index 0 ... 7 (DB2.DBB562)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM
...	...	...	...
User 63			

Variable	Data type	Variables in "LBP_AlarmMsgs [DB2]"	Description
...	...	...	...
User63_DB2_1065 (DB5.DBX167.7)	BOOL	"LBP_AlarmMsgs.UserA[63].A70UUxx[i]", Index 56 ... 63 (DB2.DBB1065)	Setting the message type in "LBP_AlarmMsgs [DB2]" TRUE: EM FALSE: SSM

Example for setting the message type for an alarm from user area 4

Searched alarms: "LBP_AlarmMsgs.UserA[4].A70UUxx[i]", index 24 to 31

The byte offset for the alarms in "LBP_AlarmMsgs" is 589 (DB2.DBB589).

You can set the message type in the variable with the following symbol name: "LBP_ParamAlarmMsgs.User4_DB2_589" (DB5.DBX34.7)

See also

LBP_GenerateAlarmMsgs [FC10]: error and operating messages (Page 690)

LBP_AlarmMsgs [DB2] - PLC alarms / messages (Page 98)

10.4.5 LBP_ConfigData [DB7] - Configuration data

10.4.5.1 LBP_ConfigData [DB7]

All input and output parameters of "LBP_ConfigBP [FC1]" are mirrored in global data block "LBP_ConfigData [DB7]". It also contains parameters that are only evaluated in cyclic operation (e.g. "IdentMcp"...).

The parameters for the configuration can be set in the Startup OB via ""LBP_ConfigBP [FC1]"". "LBP_ConfigBP [FC1]" transfers all its input and output parameters to "LBP_ConfigData [DB7]". Data can be changed in cyclic program execution by means of the variables in "LBP_ConfigData [DB7]".

Note

The input and output parameters of "LBP_ConfigBP [FC1]" are assigned predefined actual parameters and correspond to the values and symbols that are also available in "LBP_ConfigData [DB7]".

"LBP_ConfigData [DB7]" is an (unoptimized) global DB. DB7 as an instance DB had the symbol "LBP_InstConfig" in the 840D sl. Unused variables have the prefix reserved in "LBP_ConfigData [DB7]". Some symbols have been renamed.

For reasons of compatibility with the 840D sl, no offsets have changed.

Variables in "LBP_ConfigBP [DB7]"

MCPNum (DB7.DBW0)

Description	Number of MCPs (input parameter)
Data type	INT
Value range	0, 1, 2
Value 0	No MCP available, default

MCP1In, MCP2In

(DB7 from DBB2, from DBB32)

Description	Start address of a data block or input for the input signals of the machine control panel (input parameter)
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

MCP1Out, MCP2Out

(DB7 from DBB8, from DBB38)

Description	Start address of a data block or output for the output signals of the machine control panel (output parameter)
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

MCP1BusAdr, MCP2BusAdr

(DB7 DBW26, 56)

Description	DIP setting on the Ethernet MCP (input parameter)
Data type	INT
Value range	1, 2, 3 ... 254

MCP1Stop, **MCP2Stop**

(DB7 DBX62.1, 62.2)

Description	Start/stop transfer of machine control panel signals (input parameter)
Data type	BOOL
Value FALSE	Start transfer
Value TRUE	Stop transfer

MCP1NotSend, **MCP2NotSend**

(DB7 DBX62.3, 62.4)

Description	Activates send and receive operation with the MCP (input parameters)
Data type	BOOL
Value FALSE	Send and receive operation activated
Value TRUE	Receive machine control panel signals only

MCPBusType (DB7.DBB63)

Description	Bus type (input parameter) b#16#yx: • Bus type MCP1: lower nibble (x) • Bus type MCP2: upper nibble (y) Mixed operation is possible (see Chapter General information on configuration and commissioning (Page 751))
Data type	BYTE
Value range	3, 4, 5, 6
Value 3	PROFIBUS
Value 4	PROFIBUS on the MPI/DP port
Value 5	Ethernet
Value 6	PROFINET

HTIf (DB7.DBW64)

Description	Handheld unit interface (input parameter)
Data type	INT
Value range	0, 5
Value 0	No HT
Value 5	HT on Ethernet

HTIn (DB7 from DBB66)

Description	Start address of a data block or input for the received data of the PLC from the handheld unit (input parameter)
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

HTOut (DB7 from DBB72)

Description	Start address of a data block or output for the transmitted data of the PLC to the handheld unit (input parameter)
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

HTAdr (DB7.DBW96)

Description	DIP setting on the HT (input parameter)
Data type	INT
Value range	1, 2, 3 ... 254

HTStop (DB7.DBX108.1)

Description	Start/stop transfer of handheld unit signals (input parameter)
Data type	BOOL
Value FALSE	Start transfer
Value TRUE	Stop transfer

HTNotSend (DB7.DBX108.2)

Description	Activates send and receive operation with the HT (input parameter)
Data type	BOOL
Value FALSE	Send and receive operation activated
Value TRUE	Receive handheld unit signals only

NCCyclTimeout (DB7.DBW110)

Description	Time for NCK cyclic sign-of-life monitoring (input parameter)
Data type	S5time
Value range	Recommendation: 200 ms

NCRunupTimeout (DB7.DBW112)

Description	Time for NCK run-up monitoring (input parameter)
Data type	S5time
Value range	Recommendation: 50 ms

ListMDecGrp (DB7.DBW114)

Description	Activation of the expanded M group decoding (input parameter)
Data type	INT
Value range	0, 1, 2 ... 16
Value 0	Not active
Value 1 ... 16	Number of M groups

MMCToIF (DB7.DBX116.1)

Description	Activation of the transfer of HMI signals to the interface (operating modes, program control, etc.) (input parameter)
Data type	BOOL
Value TRUE	Active

HWheelMMC (DB7.DBX116.2)

Description	Definition of the handwheel selection (input parameter)
Data type	BOOL
Value FALSE	Handwheel selection via HMI
Value TRUE	Handwheel selection via user program

MCP_IF_TCS (DB7.DBX116.4)

Description	Definition of the evaluation R11 key (input parameter)
Data type	BOOL
Value FALSE	No evaluation
Value TRUE	R11 key acts as manual traversing in tool orientation (see Section "Manual traversing in tool orientation" in Chapter LBP_MCPCtrlMilling [FC19]: Transfer of the MCP signals on the interface (Page 712))

ExtendChanAxMsg (DB7.DBX116.5)

Description	Activates all areas for error and system status messages of the "LBP_GenerateAlarmMsgs [FC10]" (input parameter, not available on every SINUMERIK-PLC)
Data type	BOOL

MsgUser (DB7.DBW118)

Description	Number of user areas (input parameter) See also "LBP_AlarmMsgs [DB2]". (not available on every SINUMERIK-PLC)
Data type	INT
Value range	0, 1, 2 ... 64

IRAuxfuT (DB7.DBX120.1)

Description	Evaluation of the T function in "Hardware interrupt [OB40]" (input parameter)
Data type	BOOL

IRAuxfuH (DB7.DBX120.2)

Description	Evaluation of the H function in "Hardware interrupt" (input parameter)
Data type	BOOL

IRAuxfuE (DB7.DBX120.2)

Description	Evaluation of the DL function in "Hardware interrupt" (input parameter)
Data type	BOOL

UserVersion (DB7 from DBB122)

Description	Start address of the string variable that is displayed in the version display of the user interface (input parameter) The string variable has the following notation (max. 41 characters): <**Name**> <versionersion **xx.xx.xx**> <date **yy/mm/dd**> Example: Test version 07.06.02 13/06/04
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

OpKeyNum (DB7.DBW128)

Description	Number of active direct key modules (input parameter)
Data type	INT
Value range	0, 1, 2
Value 0	No Ethernet direct keys available

Op1KeyIn, Op2KeyIn

(DB7 from DBB130, from DBB144)

Description	Start address of a data block or input for the input signals from direct keys (input parameter)
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

Op1KeyOut, Op2KeyOut

(DB7 from DBB136, from DBB150)

Description	Start address of a data block or output for the output signals from the direct keys (input parameter)
Data type	STRUCT STRUCT Pointer to data block, input or output (Page 132)

Op1KeyBusAdr, Op2KeyBusAdr

(DB7 from DBB130, from DBB144)

Description	TCU index or DIP setting on the MCP (input parameter)
Data type	INT
Value range	1, 2, 3 ... 254

Op1KeyStop, Op2KeyStop

(DB7 DBX158.0, 158.1)

Description	Start/stop transfer of the direct key signals (input parameter)
Data type	BOOL
Value FALSE	Start transfer
Value TRUE	Stop transfer

Op1KeyNotSend, Op2KeyNotSend

(DB7 DBX158.2, 158.3)

Description	Activate send and receive mode of the direct key signals (input parameter)
Data type	BOOL
Value FALSE	Send and receive operation activated
Value TRUE	Receive direct key signals only

IdentMcpBusAdr (DB7.DBW160)

Description	TCU index or the DIP setting on the MCP, HT or direct key module (input parameter)
Data type	INT
Value range	1, 2, 3 ... 254

IdentMcpProfilNo (DB7.DBB162)

Description	Profile of a device (input parameter)
Data type	BYTE
Value range	0, 1
Value 0	Complete device
Value 1	Only direct keys

IdentMcpBusType (DB7.DBB163)

Description	Type of Ethernet component (input parameter) Only IE devices
Data type	BYTE
Value range	b#16#5

IdentMcpStrobe (DB7.DBX164.0)

Description	Activate query (input parameter)
Data type	BOOL
Value 1	Activate query

MaxModeGroup (DB7.DBW166)

Description	Number of active mode groups (output parameter)
Data type	INT
Value range	1, 2, 3 ... 10

MaxChan (DB7.DBW168)

Description	Highest active channel number (output parameter)
Data type	INT
Value range	1, 2, 3 ... 10

MaxAxis (DB7.DBW170)

Description	Highest active axis number (output parameter)
Data type	INT
Value range	1, 2, 3 ... 31

ActiveChan (DB7 ab DBB172)

Description	List of the active (TRUE) and inactive (FALSE) channels (output parameter)
Data type	ARRAY[1..10] of BOOL

ActiveAxis (DB7.DBD174)

Description	List of the active (TRUE) and inactive (FALSE) axes (output parameter)
Data type	ARRAY[1..31] of BOOL

MaxNumUserDataInt (DB7.DBW178)

Description	Number of INTEGER machine data (output parameter)
Data type	INT

MaxNumUserDataHex (DB7.DBW180)

Description	Number of HEX machine data (output parameter)
Data type	INT

MaxNumUserDataReal (DB7.DBW182)

Description	Number of REAL machine data (output parameter)
Data type	INT

IdentMcpType (DB7.DBB184)

Description	MCP type (output parameter)
Data type	BYTE
Value range	0, 16#80, 16#81, ...

IdentMcpLengthIn (DB7.DBB185)

Description	Length of the input data (MCP → PLC) (output parameter)
Data type	BYTE

IdentMcpLengthOut (DB7.DBB186)

Description	Length of the output data (PLC → MCP) (output parameter)
Data type	BYTE

LBPVersionInfo (DB7 from DBB200)

Description	Version of the basic program (output parameter)
Data type	STRING[44]

UserVersionInfo (DB7 from DBB246)

Description	Version of the user program (output parameter)
Data type	STRING[44]

GenerateAlarmMsgs (DB7 from DBB292)

Description	Output data of "LBP_GenerateAlarmMsgs [FC10]" (output parameter, not available on every SINUMERIK-PLC)
Data type	STRUCT STRUCT "GenerateAlarmMsgs" alarm message (Page 132)

OpUnitCommId (DB7.DBW300)

Description	Identifier of the communication unit (default: 1) (input parameter)
Data type	UINT
Value range	1, 2, 3, ...

OpUnitInterfaceId (DB7.DBW302)

Description	Hardware identifier of the network interface (default: 262) (input parameter) Predefined constants for the CP interface: • X120 Local~CP~Ethernet_interface_120 • X130: Local~CP~Ethernet_interface_130
Data type	HW_ANY

MCP1ConnectionId (DB7.DBW304)

Description	Connection identifier for MCP 1 on (default: 1001) (input parameter)
Data type	CONN_OUC
Value range	1, 2, 3 ... 4095

MCP2ConnectionId (DB7.DBW306)

Description	Connection identifier for MCP 2 on (default: 1002) (input parameter)
Data type	CONN_OUC
Value range	1, 2, 3 ... 4095

HTConnectionId (DB7.DBW308)

Description	Connection identifier for the HT (default: 1003) (input parameter)
Data type	CONN_OUC
Value range	1, 2, 3 ... 4095

Op1KeyConnectionId (DB7.DBW310)

Description	Connection identifier for OP1Key (default: 1004) (input parameter)
Data type	CONN_OUC
Value range	1, 2, 3 ... 4095

Op2KeyConnectionId (DB7.DBW312)

Description	Connection identifier for OP2Key (default: 1005) (input parameter)
Data type	CONN_OUC
Value range	1, 2, 3 ... 4095

IdentConnectionId (DB7.DBW314)

Description	Connection identifier for Ident (default: 1006) (input parameter)
Data type	CONN_OUC
Value range	1, 2, 3 ... 4095

MCP1LocalUdpPort (DB7.DBW316)

Description	Local UDP port for the connection to MCP 1 (default: 16001) (input parameter)
Data type	UINT
Value range	1, 2, 3 ... 49151

MCP2LocalUdpPort (DB7.DBW318)

Description	Local UDP port for the connection to MCP 2 (default: 16002) (input parameter)
Data type	UINT
Value range	1, 2, 3 ... 49151

HTLocalUdpPort (DB7.DBW320)

Description	Local UDP port for the connection to HT (default: 16003) (input parameter)
Data type	UINT
Value range	1, 2, 3 ... 49151

Op1KeyLocalUdpPort (DB7.DBW322)

Description	Local UDP port for the connection to Op1Key (default: 16004) (input parameter)
Data type	UINT
Value range	1, 2, 3 ... 49151

Op2KeyLocalUdpPort (DB7.DBW324)

Description	Local UDP port for the connection to Op2Key (default: 16005) (input parameter)
Data type	UINT
Value range	1, 2, 3 ... 49151

IdentLocalUdpPort (DB7.DBW326)

Description	Local UDP port for the connection to Ident (default: 16006) (input parameter)
Data type	UINT
Value range	1, 2, 3 ... 49151

MCP1Error (DB7.DBX328.0)

Description	Error status of the connection to MCP 1 (output parameter)
Data type	BOOL

MCP1Status (DB7.DBW330)

Description	Information on the error status of the connection to MCP 1 (output parameter)
Data type	WORD

MCP2Error (DB7.DBX332.0)

Description	Error status of the connection to MCP 2 (output parameter)
Data type	BOOL

MCP2Status (DB7.DBW334)

Description	Information on the error status of the connection to MCP 2 (output parameter)
Data type	WORD

HTError (DB7.DBX336.0)

Description	Error status of the connection to an HT (output parameter)
Data type	BOOL

HTStatus (DB7.DBW338)

Description	Information on the error status of the connection to an HT (output parameter)
Data type	WORD

Op1KeyError (DB7.DBX340.0)

Description	Error status of the connection to Op1Key (output parameter)
Data type	BOOL

Op1KeyStatus (DB7.DBW342)

Description	Information on the error status of the connection to Op1Key (output parameter)
Data type	WORD

Op2KeyError (DB7.DBX344.0)

Description	Error status of the connection to Op2Key (output parameter)
Data type	BOOL

Op2KeyStatus (DB7.DBW346)

Description	Information on the error status of the connection to Op2Key (output parameter)
Data type	WORD

IdentError (DB7.DBX348.0)

Description	Error status of the connection for Ident (output parameter)
Data type	BOOL

IdentStatus (DB7.DBW350)

Description	Information on the error status of the connection for Ident (output parameter)
Data type	WORD

MCP1Ready (DB7.DBX352.0)

Description	Status of the connection to MCP1 (output parameter)
Data type	BOOL
Value TRUE	Connection has been established

MCP2Ready (DB7.DBX352.1)

Description	Status of the connection to MCP2 (output parameter)
Data type	BOOL
Value TRUE	Connection has been established

HTReady (DB7.DBX352.2)

Description	Status of the connection to HT8 (output parameter)
Data type	BOOL
Value TRUE	Connection has been established

Op1KeyReady (DB7.DBX352.3)

Description	Status of the connection to direct keys 1 (output parameter)
Data type	BOOL
Value TRUE	Connection has been established

Op2KeyReady (DB7.DBX352.4)

Description	Status of the connection to direct keys 2 (output parameter)
Data type	BOOL
Value TRUE	Connection has been established

See also

STRUCT "Magazines" magazine data (Page 663)

Configuration machine control panel, handheld unit, direct keys (Page 751)

10.4.5.2 STRUCT Pointer to data block, input or output

The variable structure describes a point to the address of a data block, input or output.

Variables in MCP1In, MCP1Out, MCP2In, MCP2Out, HTIn, HTOut, UserVersion, Op1KeyIn, Op1KeyOut, Op2KeyIn, Op2KeyOut

DBNo

Description	Returns the number of the DB
Data type	WORD
Range of values	Depends on the CPU (data block number) or the I/Os (input or output)
Value 0	For P#M, P#I, or P#Q, parameterize value 0.

Addr

Description	Specify the point to the address in the memory area. Only global memory areas are permissible.
Data type	DWORD
Range of values	P#DBXx.0 or P#Ix.0 for inputs P#Qx.0 for outputs:
Value 0	For P#M, P#I, or P#Q, parameterize value 0.

10.4.5.3 STRUCT "GenerateAlarmMsgs" alarm message

The variable structure "AlarmMsgs" describes the alarm information.

Variables in "GenerateAlarmMsgs"

NumActAlarmMsgs (DB7.DBW292)

Description	Returns the number of active alarms/messages
Data type	UINT
Range of values	0 ... 65535

Error (DB7.DBX294.0)

Description	Returns an error bit Error = TRUE means that an error occurred during the processing. The possible cause of the error is shown in the "StatusId" and "Status" parameters.
Data type	BOOL

StatusId (DB7.DBW296)

Description	Returns a status identifier
Data type	UINT
Range of values	0 ... 65535

Status (DB7.DBW298)

Description	Returns status information
Data type	WORD
Range of values	16#0 ... 16#FFFF

10.4.6 LBP_NC [DB10] - Signals from/to the NC, PLC and operating software

10.4.6.1 Overview

LBP_NC [DB10], onboard inputs and outputs of the NC

Table 10-16 Onboard inputs and outputs of the NC

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Disable of the digital NC inputs							
	Input without hardware				Input onboard			
	8	7	6	5	4	3	2	1
DBB1	Setting of the digital NC inputs from the PLC							
	Input without hardware				Input onboard			
	8	7	6	5	4	3	2	1
DBB2 - DBB3	Not assigned							
DBB4	Disable of the digital NC outputs							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB5	Overwrite mask of the digital NC outputs							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1
DBB6	Setting value of the digital NC outputs from the PLC							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1
DBB7	Input mask of the digital NC outputs							
	Output without hardware				Output onboard			
	8	7	6	5	4	3	2	1
DBB8 - DBB29	Machine axis numbers table for "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]", "LBP_MCPCtrlTurning [FC25]", "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK-PLC) (1st MCP)							
DBW30	Upper limit of the machine axis numbers for "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]" (1st MCP) With 0, the maximum number of machine axis numbers applies							
DBB32 - DBB53	Machine axis numbers table for "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]", "LBP_MCPCtrlTurning [FC25]", "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK-PLC) (2nd MCP)							
DBW54	Upper limit of the machine axis numbers for "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]" (2nd MCP) With 0, the maximum number of machine axis numbers applies							

LBP_NC [DB10], general signals to the NC

Table 10-17 General signals to the NC

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB56	Key-operated switch position					Acknowledge Emergency Stop	Emergency Stop	
	Bit 3	Bit 2	Bit 1	Bit 0				
DBB57					Reserved			INC inputs in the mode group range active
DBB58	Collision avoidance: Deactivate protection area group							
	For operating mode: JOG				For operating mode: AUTOMATIC			
	Workpieces	Workholder	Tools	Machine	Workpieces	Workholder	Tools	Machine
DBB59								

LBP_NC [DB10], onboard inputs and outputs from the NC / operating software

Table 10-18 Onboard inputs and outputs from the NC/operating software

LBP_NC [DB10]	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60	Actual value of the digital NC inputs				Actual value of the digital onboard inputs of the NC			
	Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1
DBB61 - DBB63								
DBB64	Setpoint for the digital outputs of the NC without hardware				Setpoint for the digital onboard outputs of the NC			
	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
DBB65 - DBB67								
DBB68	Handwheel 1 is operated							
DBB69	Handwheel 2 is operated							
DBB70	Handwheel 3 is operated							
DBB71	Change counter, inch/metric system of units							
DBB72	Status of the displayed actual value screen (1st MCP)							
HT 8 → operating software	Traversing keys shown						MCS / WCS	Display valid
DBB73	Status of the displayed actual value screen (2nd MCP)							
HT 8 → operating software	Traversing keys shown						MCS / WCS	Display valid

LBP_NC [DB10]	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB74 - DBB79	Machine axis numbers of the displayed axes (1st MCP) MCPT1AxisFromHMI							
HT 8 → operating software								
DBB80 - DBB85	Machine axis numbers of the displayed axes (2nd MCP) MCP2AxisFromHMI							
HT 8 → operating software								
DBW86	Reserved							
DBB88	Reserved							

LBP_NC [DB10], selection and status signals from the operating software

Table 10-19 Selection and status signals from the operating software

LBP_NC [DB10]	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB90								
SINUMERIK Integrate → PLC								
DBB91								
PLC → SINUMERIK Integrate								
DBB92		Suppress fault message in case of failure				Slave OK		
GP → PLC		PN bus	DP1 bus	MPI/DP bus		PN bus	DP1 bus	MPI/DP bus
DBB93	Deactivate collision avoidance							
Operating software → PLC	JOG mode				AUTO mode			
	Workpieces	Workholder	Tools	Machine	Workpieces	Workholder	Tools	Machine
DBB94				Alarm Direct keys 2 Communication error	Alarm, direct keys 1 Communication error	Warning HT communication disrupted	Warning MCP 2 communication disrupted	Warning MCP 1 communication disrupted
DBB95								

LBP_NC [DB10]	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB96	Set language ID of the operating software							
Operating software → PLC								
DBB97					Channel number for handwheel 1			
Operating software → PLC					D	C	B	A
DBB98					Channel number for handwheel 2			
Operating software → PLC					D	C	B	A
DBB99					Channel number for handwheel 3			
Operating software → PLC					D	C	B	A
DBB100				Axis number for handwheel 1				
Operating software → PLC	Machine axis	Handwheel 1 selected	Define handwheel 1 as contour handwheel	E	D	C	B	A
DBB101				Axis number for handwheel 2				
Operating software → PLC	Machine axis	Handwheel 2 selected	Define handwheel 2 as contour handwheel	E	D	C	B	A
DBB102				Axis number for handwheel 3				
Operating software → PLC	Machine axis	Handwheel 3 selected	Define handwheel 3 as contour handwheel	E	D	C	B	A
DBB103 Operating software → PLC	Operating software battery alarm	Operating software temperature limit	AT box ready	Operating software fan monitoring	Operating software monitor hard disk			Remote diagnostics active

LBP_NC [DB10], general signals from the NC

Table 10-20 General signals to the PLC

LBP_NC [DB10]	**Signals to the PLC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB104 GP → PLC	NC CPU: Ready[1)]	1st OB1 cycle		Op2Key ready	Op1Key ready	HT ready	MCP 2 ready	MCP 1 ready
DBB105 GP → PLC								TOOLMAN: Command Cancel
DBB106 NC → PLC							Emergency Stop: Active	Collision avoidance: Protection area monitoring active
DBB107 NC → PLC	Inch system of units	NCU link: Active					Probe actuated	
							Pushbutton 2	Pushbutton 1
DBB108 NC → PLC	NC ready	Drive ready	Drives in cyclic operation		Operating software ready	Operator panel at MPI: "ready"	Operator panel 2: "ready"	
DBB109 NC → PLC	NC battery alarm	Air temperature alarm	Heat sink temperature NCU alarm	PC operating system error				NC alarm is active
DBB110	Software cam minus							
NC → PLC	7	8	5	4	3	2	1	0
DBB111	Software cam minus							
NC → PLC	15	14	13	12	11	10	9	8
DBB112	Software cam minus							
NC → PLC	23	22	21	20	19	18	17	16
DBB113	Software cam minus							
NC → PLC	31	30	29	28	27	26	25	24
DBB114	Software cam plus							
NC → PLC	7	6	5	4	3	2	1	0
DBB115	Software cam plus							
NC → PLC	15	14	13	12	11	10	9	8
DBB116	Software cam plus							
NC → PLC	23	22	21	20	19	18	17	16
DBB117	Software cam plus							
NC → PLC	31	30	29	28	27	26	25	24

LBP_NC [DB10]	**Signals to the PLC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB118-DBB121 SINUMERIK Integrate → PLC	SINUMERIK Integrate data							

1) "LBP_NC.E_NCKready" (DB10.DBX104.7) (NC-CPU: ready) The signal should be included in the machine safety circuit.

LBP_NC [DB10], external digital NC inputs

Table 10-21 External digital NC inputs

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB122	Disable of the external digital NC inputs							
	16	15	14	13	12	11	10	9
DBB123	Values from the PLC for the external digital NC inputs							
	16	15	14	13	12	11	10	9
DBB124	Disable of the external digital NC inputs							
	24	23	22	21	20	19	18	17
DBB125	Values from the PLC for the external digital NC inputs							
	24	23	22	21	20	19	18	17
DBB126	Disable of the external digital NC inputs							
	32	31	30	29	28	27	26	25
DBB127	Values from the PLC for the external digital NC inputs							
	32	31	30	29	28	27	26	25
DBB128	Disable of the external digital NC inputs							
	40	39	38	37	36	35	34	33
DBB129	Values from the PLC for the external digital NC inputs							
	40	39	38	37	36	35	34	33

LBP_NC [DB10], external digital NC outputs

Table 10-22 External digital NC outputs

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB130	Disable of the external digital NC outputs							
	16	15	14	13	12	11	10	9
DBB131	Overwrite mask for the external digital NC outputs							
	16	15	14	13	12	11	10	9

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB132	Value from the PLC for the external digital NC outputs							
	16	15	14	13	12	11	10	9
DBB133	Input mask for the external digital NC outputs							
	16	15	14	13	12	11	10	9
DBB134	Disable of the external digital NC outputs							
	24	23	22	21	20	19	18	17
DBB135	Overwrite mask for the external digital NC outputs							
	24	23	22	21	20	19	18	17
DBB136	Value from the PLC for the external digital NC outputs							
	24	23	22	21	20	19	18	17
DBB137	Input mask for the external digital NC outputs							
	24	23	22	21	20	19	18	17
DBB138	Disable of the external digital NC outputs							
	32	31	30	29	28	27	26	25
DBB139	Overwrite mask for the external digital NC outputs							
	32	31	30	29	28	27	26	25
DBB140	Value from the PLC for the external digital NC outputs							
	32	31	30	29	28	27	26	25
DBB141	Input mask for the external digital NC outputs							
	32	31	30	29	28	27	26	25
DBB142	Disable of the external digital NC outputs							
	40	39	38	37	36	35	34	33
DBB143	Overwrite mask for the external digital NC outputs							
	40	39	38	37	36	35	34	33
DBB144	Value from the PLC for the external digital NC outputs							
	40	39	38	37	36	35	34	33
DBB145	Input mask for the external digital NC outputs							
	40	39	38	37	36	35	34	33

LBP_NC [DB10], external analog NC inputs

Table 10-23 External analog NC inputs

LBP_NC [DB10]	**Signals to the NC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB146	Analog NC inputs disabled							
	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
DBB147	Input analog value input for the NC from the PLC							
	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1

LBP_NC [DB10]	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW148	Setpoint from the PLC for analog input 1 of the NC							
DBW150	Setpoint from the PLC for analog input 2 of the NC							
DBW152	Setpoint from the PLC for analog input 3 of the NC							
DBW154	Setpoint from the PLC for analog input 4 of the NC							
DBW156	Setpoint from the PLC for analog input 5 of the NC							
DBW158	Setpoint from the PLC for analog input 6 of the NC							
DBW160	Setpoint from the PLC for analog input 7 of the NC							
DBW162	Setpoint from the PLC for analog input 8 of the NC							
DBW164								

LBP_NC [DB10], external analog NC outputs

Table 10-24 External analog NC outputs

LBP_NC [DB10]	Signals to the NC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB166	Overwrite mask for the analog NC outputs							
	8	7	6	5	4	3	2	1
DBB167	Input mask for the analog NC outputs							
	8	7	6	5	4	3	2	1
DBB168	Analog NC outputs disabled							
	8	7	6	5	4	3	2	1
DBB169	Reserved							
DBW170	Setpoint from the PLC for analog output 1 of the NC							
DBW172	Setpoint from the PLC for analog output 2 of the NC							
DBW174	Setpoint from the PLC for analog output 3 of the NC							
DBW176	Setpoint from the PLC for analog output 4 of the NC							
DBW178	Setpoint from the PLC for analog output 5 of the NC							
DBW180	Setpoint from the PLC for analog output 6 of the NC							
DBW182	Setpoint from the PLC for analog output 7 of the NC							
DBW184	Setpoint from the PLC for analog output 8 of the NC							

LBP_NC [DB10], external digital NC inputs and outputs

Table 10-25 External digital NC inputs and outputs

LBP_NC [DB10]	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB186	Actual value of the external digital NC inputs							
	16	15	14	13	12	11	10	9
DBB187	Actual value of the external digital NC inputs							
	24	23	22	21	20	19	18	17
DBB188	Actual value of the external digital NC inputs							
	32	31	30	29	28	27	26	25
DBB189	Actual value of the external digital NC inputs							
	40	39	38	37	36	35	34	33
DBB190	NC setpoint for the external digital NC outputs							
	16	15	14	13	12	11	10	9
DBB191	NC setpoint for the external digital NC outputs							
	24	23	22	21	20	19	18	17
DBB192	NC setpoint for the external digital NC outputs							
	32	31	30	29	28	27	26	25
DBB193	NC setpoint for the external digital NC outputs							
	40	39	38	37	36	35	34	33

LBP_NC [DB10], analog NC inputs and outputs

Table 10-26 Analog NC inputs and outputs

DB10	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW194	Actual value of analog input 1 of the NC							
DBW196	Actual value of analog input 2 of the NC							
DBW198	Actual value of analog input 3 of the NC							
DBW200	Actual value of analog input 4 of the NC							
DBW202	Actual value of analog input 5 of the NC							
DBW204	Actual value of analog input 6 of the NC							
DBW206	Actual value of analog input 7 of the NC							
DBW208	Actual value of analog input 8 of the NC							
DBW210	Setpoint of analog output 1 of the NC							
DBW212	Setpoint of analog output 2 of the NC							
DBW214	Setpoint of analog output 3 of the NC							
DBW216	Setpoint of analog output 4 of the NC							
DBW218	Setpoint of analog output 5 of the NC							
DBW220	Setpoint of analog output 6 of the NC							

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW222	Setpoint of analog output 7 of the NC							
DBW224	Setpoint of analog output 8 of the NC							

LBP_NC [DB10], collision avoidance: Protection area active

Table 10-27 Collision avoidance: Protection zone active

DB10	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB226	Collision avoidance: Protection zone active (bit)							
	7	6	5	4	3	2	1	0
DBB227	Collision avoidance: Protection zone active (bit)							
	15	14	13	12	11	10	9	8
DBB228	Collision avoidance: Protection zone active (bit)							
	23	22	21	20	19	18	17	16
DBB229	Collision avoidance: Protection zone active (bit)							
	31	30	29	28	27	26	25	24
DBB230	Collision avoidance: Protection zone active (bit)							
	39	38	37	36	35	34	33	32
DBB231	Collision avoidance: Protection zone active (bit)							
	47	46	45	44	43	42	41	40
DBB232	Collision avoidance: Protection zone active (bit)							
	55	54	53	52	51	50	49	48
DBB233	Collision avoidance: Protection zone active (bit)							
	63	62	61	60	59	58	57	56

LBP_NC [DB10], collision avoidance: Activate protection area

Table 10-28 Collision avoidance: Activate protection zone

LBP_NC [DB10]	Signals from the PLC (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB234	Collision avoidance: Activate protection zone (bit)							
	7	6	5	4	3	2	1	0
DBB235	Collision avoidance: Activate protection zone (bit)							
	15	14	13	12	11	10	9	8
DBB236	Collision avoidance: Activate protection zone (bit)							
	23	22	21	20	19	18	17	16
DBB237	Collision avoidance: Activate protection zone (bit)							
	31	30	29	28	27	26	25	24

LBP_NC [DB10]	**Signals from the PLC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB238	Collision avoidance: Activate protection zone (bit)							
	39	38	37	36	35	34	33	32
DBB239	Collision avoidance: Activate protection zone (bit)							
	47	46	45	44	43	42	41	40
DBB240	Collision avoidance: Activate protection zone (bit)							
	55	54	53	52	51	50	49	48
DBB241	Collision avoidance: Activate protection zone (bit)							
	63	62	61	60	59	58	57	56

LBP_NC [DB10], extension, handwheel signals from the NC

Table 10-29 Extension, handwheel signals from the NC

LBP_NC [DB10]	**Signals from the NC (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB242	Handwheel 4 is operated							
DBB243	Handwheel 5 is operated							
DBB244	Handwheel 6 is operated							
DBB245			Ethernet handwheel is stationary					
			Handwheel 6	Handwheel 5	Handwheel 4	Handwheel 3	Handwheel 2	Handwheel 1
DBB246			Handwheel active					
			Handwheel 6	Handwheel 5	Handwheel 4	Handwheel 3	Handwheel 2	Handwheel 1

LBP_NC [DB10], interface robot status

Table 10-30 Signals from the robot

LBP_NC [DB10]	**Signals from the PLC (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB248	Robot status byte 0							
DBB249	Robot status byte 1							
DBB250	Robot status byte 2							
DBB251	Robot status byte 3							
DBB252	Robot status byte 4							
DBB253	Robot status byte 5							
DBB254	Robot status byte 6							
DBB255	Robot status byte 7							

LBP_NC [DB10], interface robot status

Table 10-31 Signals to the robot

LBP_NC [DB10]	Signals from the NC (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB256	Robot control byte 0							
DBB257	Robot control byte 1							
DBB258	Robot control byte 2							
DBB259	Robot control byte 3							
DBB260	Robot control byte 4							
DBB261	Robot control byte 5							
DBB262	Robot control byte 6							
DBB263	Robot control byte 7							

10.4.6.2 LBP_NC [DB10]

Variables in "LBP_NC [DB10]"

In "LBP_NC [DB10]", the signals are stored between PLC, NC, and HMI.

Inputs and outputs of the NC

A_Disabl_Inp1, A_Disabl_Inp2, A_Disabl_Inp3, A_Disabl_Inp4, A_Disabl_Inp5, A_Disabl_Inp6, A_Disabl_Inp7, A_Disabl_Inp8

(DB10 DBX0.0, 0.1, ..., 0.7)

A_Disabl_Inp9, A_Disabl_Inp10, A_Disabl_Inp11, A_Disabl_Inp12, A_Disabl_Inp13, A_Disabl_Inp14, A_Disabl_Inp15, A_Disabl_Inp16

(DB10 DBX122.0, 122.1, ..., 122.7)

A_Disabl_Inp17, A_Disabl_Inp18, A_Disabl_Inp19, A_Disabl_Inp20, A_Disabl_Inp21, A_Disabl_Inp22, A_Disabl_Inp23, A_Disabl_Inp24

(DB10 DBX124.0, 124.1, ..., 124.7)

A_Disabl_Inp25, A_Disabl_Inp26, A_Disabl_Inp27, A_Disabl_Inp28, A_Disabl_Inp29, A_Disabl_Inp30, A_Disabl_Inp31, A_Disabl_Inp32

(DB10 DBX126.0, 126.1, ..., 126.7)

A_Disabl_Inp33, A_Disabl_Inp34, A_Disabl_Inp35, A_Disabl_Inp36, A_Disabl_Inp37, A_Disabl_Inp38, A_Disabl_Inp39, A_Disabl_Inp40

(DB10 DBX128.0, 128.1, ..., 128.7)

Description	Disabling of the digital input 1 ... 40 Input 1 ... 4: Onboard input Input 5 ... 8: Input without hardware Input 9 ... 40: External digital NC input **Note** When read, a locked input supplies a value of 0.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The digital NC input is enabled.
Value TRUE	The digital NC input is locked.
Corresponds to	• "LBP_NC.A_Disabl_Inp..." (DB10 DBB0, 122, 124, 126, 128) Digital NC inputs: Lock • "LBP_NC.A_Set_Inp..." (DB10 DBB1, 123, 125, 127, 129) Digital NC inputs: Set • "LBP_NC.E_ActVal_In..." (DB10 DBB60, 186, 187, 188, 189) Digital NC inputs: Actual value • MD10350 $MN_FASTIO_DIG_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Digital NC inputs/outputs" > "NC inputs"

A_Set_Inp1, A_Set_Inp2, A_Set_Inp3, A_Set_Inp4, A_Set_Inp5, A_Set_Inp6, A_Set_Inp7, A_Set_Inp8

(DB10 DBX1.0, 1.1, ..., 1.7)

A_Set_Inp9, A_Set_Inp10, A_Set_Inp11, A_Set_Inp12, A_Set_Inp13, A_Set_Inp14, A_Set_Inp15, A_Set_Inp16

(DB10 DBX123.0, 123.1, ..., 123.7)

A_Set_Inp17, A_Set_Inp18, A_Set_Inp19, A_Set_Inp20, A_Set_Inp21, A_Set_Inp22, A_Set_Inp23, A_Set_Inp24

(DB10 DBX125.0, 125.1, ..., 125.7)

A_Set_Inp25, A_Set_Inp26, A_Set_Inp27, A_Set_Inp28, A_Set_Inp29, A_Set_Inp30, Set_Inp31, A_Set_Inp32

(DB10 DBX127.0, 127.1, ..., 127.7)

A_Set_Inp33, A_Set_Inp34, A_Set_Inp35, A_Set_Inp36, A_Set_Inp37, A_Set_Inp38, A_Set_Inp39, A_Set_Inp40

(DB10 DBX129.0, 129.1, ..., 129.7)

Description	Setting of the digital input 1 ... 40 Input 1 ... 4: Onboard input Input 5 ... 8: Input without hardware Input 9 ... 40: External digital NC input If the PLC user program sets the NC input to a value of TRUE, the signal state at the onboard input of the NCU - as well as the lock of the NC input - have no effect.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Do not influence NC input value.
Value TRUE	Set NC input value to a defined value of 1.
Corresponds to	• "LBP_NC.A_Disabl_Inp..." (DB10 DBB0, 122, 124, 126, 128) Digital NC inputs: Lock • "LBP_NC.A_Set_Inp..." (DB10 DBB1, 123, 125, 127, 129) Digital NC inputs: Set • "LBP_NC.E_ActVal_In..." (DB10 DBB60, 186, 187, 188, 189) Digital NC inputs: Actual value • MD10350 $MN_FASTIO_DIG_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Digital NC inputs/outputs" > "NC inputs"

A_Disabl_Out1, A_Disabl_Out2, A_Disabl_Out3, A_Disabl_Out4, A_Disabl_Out5, A_Disabl_Out6, A_Disabl_Out7, A_Disabl_Out8

(DB10 DBX4.0, 4.1, ..., 4.7)

A_Disabl_Out9, A_Disabl_Out10, A_Disabl_Out11, A_Disabl_Out12, A_Disabl_Out13, A_Disabl_Out14, A_Disabl_Out15, A_Disabl_Out16

(DB10 DBX130.0, 130.1, ..., 130.7)

A_Disabl_Out17, A_Disabl_Out18, A_Disabl_Out19, A_Disabl_Out20, A_Disabl_Out21, A_Disabl_Out22, A_Disabl_Out23, A_Disabl_Out24

(DB10 DBX134.0, 134.1, ..., 134.7)

A_Disabl_Out25, A_Disabl_Out26, A_Disabl_Out27, A_Disabl_Out28, A_Disabl_Out29, A_Disabl_Out30, A_Disabl_Out31, A_Disabl_Out32

(DB10 DBX138.0, 138.1, ..., 138.7)

A_Disabl_Out33, A_Disabl_Out34, A_Disabl_Out35, A_Disabl_Out36, A_Disabl_Out37, A_Disabl_Out38, A_Disabl_Out39, A_Disabl_Out40

(DB10 DBX142.0, 142.1, ..., 142.7)

Description	Disabling of the digital NC output 1 ... 40 Output 1 ... 4: Onboard output Output 5 ... 8: Output without hardware Output 9 ... 40: External digital NC output If the digital NC output is locked, 0 V (defined) is output at the hardware output. If the digital NC output is not locked, the value specified in the NC program or by the PLC user program is output at the hardware output.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The digital NC output is enabled.
Value TRUE	The digital NC output is locked.
Corresponds to	• "LBP_NC.A_OvMask_Out..." (DB10 DBB5, 131, 135, 139, 143) Digital NC outputs: Overwrite screen form • "LBP_NC.A_Set_Out..." (DB10 DBB6, 132, 136, 140, 144) Digital NC outputs: Setting value from the PLC • "LBP_NC.A_InMask_Out..." (DB10 DBB7, 133, 137, 141, 145) Digital NC outputs: Setting screen form • MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Digital NC inputs/outputs" > "NC outputs"

A_OvMask_Out1, A_OvMask_Out2, A_OvMask_Out3, A_OvMask_Out4, A_OvMask_Out5, A_OvMask_Out6, A_OvMask_Out7, A_OvMask_Out8

(DB10 DBX5.0, 5.1, ..., 5.7)

A_OvMask_Out9, A_OvMask_Out10, A_OvMask_Out11, A_OvMask_Out12, A_OvMask_Out13, A_OvMask_Out14, A_OvMask_Out15, A_OvMask_Out16

(DB10 DBX131.0, 131.1, ..., 131.7)

A_OvMask_Out17, A_OvMask_Out18, A_OvMask_Out19, A_OvMask_Out20, A_OvMask_Out21, A_OvMask_Out22, A_OvMask_Out23, A_OvMask_Out24

(DB10 DBX135.0, 135.1, ..., 135.7)

A_OvMask_Out25, A_OvMask_Out26, A_OvMask_Out27, A_OvMask_Out28, A_OvMask_Out29, A_OvMask_Out30, A_OvMask_Out31, A_OvMask_Out32

(DB10 DBX139.0, 139.1, ..., 139.7)

A_OvMask_Out33, A_OvMask_Out34, A_OvMask_Out35, A_OvMask_Out36, A_OvMask_Out37, A_OvMask_Out38, A_OvMask_Out39, A_OvMask_Out40

(DB10 DBX143.0, 143.1, ..., 143.7)

Description	Overwrite screen form of the digital NC output 1 ... 40 Output 1 ... 4: Onboard output Output 5 ... 8: Output without hardware Output 9 ... 40: External digital NC output For a positive edge change 0 → 1, the setting value specified from the PLC user program is used for the corresponding output, instead of the value written using the system variable $A_OUT. The value written via system variables $A_OUT is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept. **Note** The variable "A_Set_Out1", ... (DB10.DBX6.0, ...) (digital NC outputs: setting value) is jointly used by: • "A_OvMask_Out1", ... (DB10.DBX5.0, ...) (overwrite) for edge change 0 → 1 • "A_InMask_Out1", ... (DB10.DBX7.0, ...) (target) for signal state TRUE Activating the two interfaces simultaneously should be avoided.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	The associated "setting value" is activated.
Edge change 1 → 0	No effect.
Corresponds to	• "LBP_NC.A_Disabl_Out..." (DB10 DBB4, 130, 134, 138, 142) Digital NC outputs: Lock • "LBP_NC.A_OvMask_Out..." (DB10 DBB5, 131, 135, 139, 143) Digital NC outputs: Overwrite screen form • "LBP_NC.A_Set_Out..." (DB10 DBB6, 132, 136, 140, 144) Digital NC outputs: Setting value from the PLC • MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O"

A_Set_Out1, A_Set_Out2, A_Set_Out3, A_Set_Out4, A_Set_Out5, A_Set_Out6, A_Set_Out7, A_Set_Out8

(DB10 DBX6.0, 6.1, ..., 6.7)

A_Set_Out9, A_Set_Out10, A_Set_Out11, A_Set_Out12, A_Set_Out13, A_Set_Out14, A_Set_Out15, A_Set_Out16

(DB10 DBX132.0, 132.1, ..., 132.7)

A_Set_Out17, A_Set_Out18, A_Set_Out19, A_Set_Out20, A_Set_Out21, A_Set_Out22, A_Set_Out23, A_Set_Out24

(DB10 DBX136.0, 136.1, ..., 136.7)

A_Set_Out25, A_Set_Out26, A_Set_Out27, A_Set_Out28, A_Set_Out29, A_Set_Out30, A_Set_Out31, A_Set_Out32

(DB10 DBX140.0, 140.1, ..., 140.7)

A_Set_Out33, A_Set_Out34, A_Set_Out35, A_Set_Out36, A_Set_Out37, A_Set_Out38, A_Set_Out39, A_Set_Out40

(DB10 DBX144.0, 144.1, ..., 144.7)

Description	Setting value of the digital NC output 1 ... 40 of PLC Output 1 ... 4: Onboard output Output 5 ... 8: Output without hardware Output 9 ... 40: External digital NC output Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated from the overwrite screen form or the setting screen form. **Note** The variable "A_Set_Out1", ... (DB10.DBX6.0, ...) (digital NC outputs: setting value) is jointly used by: • "A_OvMask_Out1", ... (DB10.DBX5.0, ...) (overwrite) for edge change 0 → 1 • "A_InMask_Out1", ... (DB10.DBX7.0, ...) (target) for signal state TRUE Activating the two interfaces simultaneously should be avoided.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Value of the setting value is 0.
Value TRUE	Value of the setting value is 1.
Corresponds to	• "LBP_NC.A_Disabl_Out..." (DB10 DBB4, 130, 134, 138, 142) Digital NC outputs: Lock • "LBP_NC.A_OvMask_Out..." (DB10 DBB5, 131, 135, 139, 143) Digital NC outputs: Overwrite screen form • "LBP_NC.A_Set_Out..." (DB10 DBB6, 132, 136, 140, 144) Digital NC outputs: Setting value from the PLC • MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Digital NC inputs/outputs" > "NC outputs"

A_InMask_Out1, A_InMask_Out2, A_InMask_Out3, A_InMask_Out4, A_InMask_Out5, A_InMask_Out6, A_InMask_Out7, A_InMask_Out8

(DB10 DBX7.0, 7.1, ..., 7.7)

A_InMask_Out9, A_InMask_Out10, A_InMask_Out11, A_InMask_Out12, A_InMask_Out13, A_InMask_Out14, A_InMask_Out15, A_InMask_Out16

(DB10 DBX133.0, 133.1, ..., 133.7)

A_InMask_Out17, A_InMask_Out18, A_InMask_Out19, A_InMask_Out20, A_InMask_Out21, A_InMask_Out22, A_InMask_Out23, A_InMask_Out24

(DB10 DBX137.0, 137.1, ..., 137.7)

A_InMask_Out25, A_InMask_Out26, A_InMask_Out27, A_InMask_Out28, A_InMask_Out29, A_InMask_Out30, A_InMask_Out31, A_InMask_Out32

(DB10 DBX141.0, 141.1, ..., 141.7)

A_InMask_Out33, A_InMask_Out34, A_InMask_Out35, A_InMask_Out36, A_InMask_Out37, A_InMask_Out38, A_InMask_Out39, A_InMask_Out40

(DB10 DBX145.0, 145.1, ..., 145.7)

Description	Setting screen form of the digital NC output 1 ... 40 Output 1 ... 4: Onboard output Output 5 ... 8: Output without hardware Output 9 ... 40: External digital NC output If the bit is set, the setting value specified by the PLC user program is used for the corresponding output instead of the NC output value. The current NC output value is retained. If the bit is reset, the last NC output value becomes active again for the corresponding output. **Note** The variable "A_Set_Out1", ... (DB10.DBX6.0, ...) (digital NC outputs: setting value) is jointly used by: • "A_OvMask_Out1", ... (DB10.DBX5.0, ...) (overwrite) for edge change 0 → 1 • "A_InMask_Out1", ... (DB10.DBX7.0, ...) (target) for signal state TRUE Activating the two interfaces simultaneously should be avoided.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The target is **not** active.
Value TRUE	The target is active.
Corresponds to	• "LBP_NC.A_Disabl_Out..." (DB10 DBB4, 130, 134, 138, 142) Digital NC outputs: Lock • "LBP_NC.A_OvMask_Out..." (DB10 DBB5, 131, 135, 139, 143) Digital NC outputs: Overwrite screen form • "LBP_NC.A_Set_Out..." (DB10 DBB6, 132, 136, 140, 144) Digital NC outputs: Setting value from the PLC • MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Digital NC inputs/outputs" > "NC outputs"

MCP1AxisTbl (DB10 from DBB8)

Description	Table with numbers of the machine axes (1st MCP) Use in the following functions: • "LBP_MCPCtrlMilling [FC19]" • "LBP_MCPCtrlMillingSmall [FC24]" • "LBP_MCPCtrlTurning [FC25]" • "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK PLC)
Signal flow	PLC → NC
Data type	ARRAY[1..22] of BYTE

MCP1MaxAxis (DB10.DBW30)

Description	Upper limit of the machine axis numbers (1st MCP) Use in the following functions: • "LBP_MCPCtrlMilling [FC19]" • "LBP_MCPCtrlMillingSmall [FC24]"
Signal flow	PLC → NC
Data type	WORD
Value 0	Max. number of machine axis numbers
Value not equal to 0	Upper limit of the machine axis numbers

MCP2AxisTbl (DB10 from DBB32)

Description	Table with numbers of the machine axes (2nd MCP) Use in the following functions: • "LBP_MCPCtrlMilling [FC19]" • "LBP_MCPCtrlMillingSmall [FC24]" • "LBP_MCPCtrlTurning [FC25]" • "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK PLC)
Signal flow	PLC → NC
Data type	ARRAY[1..22] of BYTE

MCP2MaxAxis (DB10.DBW54)

Description	Upper limit of the machine axis numbers (2nd MCP) Use in the following functions: • "LBP_MCPCtrlMilling [FC19]" • "LBP_MCPCtrlMillingSmall [FC24]"
Signal flow	PLC → NC
Data type	WORD
Value 0	Max. number of machine axis numbers
Value not equal to 0	Upper limit of the machine axis numbers

A_EMERGENCY (DB10.DBX56.1)

Description	Emergency stop All machine axes are decelerated in the relevant axis-specific parameterized time: MD36610 $MA_AX_EMERGENCY_STOP_TIME
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Emergency Stop is **not** requested.
Value TRUE	Emergency Stop is requested.

Corresponds to	• "LBP_NC.A_EMERGENCY_Ackn" (DB10.DBX56.2) Acknowledge emergency stop • "LBP_NC.E_EMERGENCY" (DB10.DBX106.1) Emergency stop active • MD36610 $MA_AX_EMERGENCY_STOP_TIME Time of braking ramp in case of errors
Additional references	Function Manual Basic Functions; Section "Emergency stop"

A_EMERGENCY_Ackn (DB10.DBX56.2)

Description	Acknowledge Emergency Stop To acknowledge the "Emergency Stop" state of the NC, the following interface signals must remain set until variable "LBP_NC.E_EMERGENCY" (DB10.DBX56.1) (emergency stop active) has been reset: • "LBP_NC.A_EMERGENCY_Ackn" (DB10.DBX56.2) = TRUE (Acknowledge Emergency Stop) • "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) = TRUE (mode group reset) for all mode groups of the NC
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Acknowledgement of the "Emergency Stop" state is not requested.
Value TRUE	Acknowledgement of the "Emergency Stop" state is requested.
Corresponds to	• "LBP_NC.A_EMERGENCY" (DB10.DBX56.1) Emergency stop • "LBP_NC.E_EMERGENCY" (DB10.DBX106.1) Emergency stop active • "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) Mode group reset
Additional references	Function Manual Basic Functions; Section "Emergency stop"

A_Keyswitch0, A_Keyswitch1, A_Keyswitch2, A_Keyswitch3

(DB10 DBX56.4, 56.5, 56.6, 56.7)

Description	Key-operated switch position 0 ... 3 Depending on the key-operated switch position, access to certain elements in the NC can be enabled or disabled: • Key-operated switch position 0 represents the lowest access rights • Key-operated switch position 3 represents the highest access rights The interface signals of key-operated switch positions 1 ... 3 can either be directly assigned from the key-operated switch on the machine control panel or from the PLC user program. It is only permissible to set one bit. If several bits are set simultaneously, the control internally activates switch position 3.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value of A_Keyswitch0 = TRUE	Key-operated switch position 0

Value of A_Keyswitch1 = TRUE	Key-operated switch position 1
Value of A_Keyswitch2 = TRUE	Key-operated switch position 2
Value of A_Keyswitch3 = TRUE	Key-operated switch position 3
Corresponds to	Machine data for access levels: MD11612, MD51044 - MD51064, MD51070 - MD51073, MD51199 - MD51211, MD51215 - MD51225, MD51235 Disabling using a password

A_IncInModeGroup (DB10.DBX57.0)

Description	INC inputs in the mode group area active The INC selections are transferred to the mode group interface. If TRUE, activation is performed once by "LBP_MCPCtrlMillingSmall [FC24]" after start-up.
Signal flow	PLC → NC
Data type	BOOL
Value TRUE	Transferring the INC selections to the mode group interface

A_CA_DisableAutoMatch, A_CA_DisableAutoTools, A_CA_DisableAutoWorkh, A_CA_DisableAutoWorkp, A_CA_DisableJogMach, A_CA_DisableJogTools, A_CA_DisableJogWorkh, A_CA_DisableJogWorkp

(DB10 DBX58.0, 58.1, ..., 58.7)

<table>
<tr><td>Description</td><td colspan="3">Collision avoidance: Deactivate protection area group</td></tr>
<tr><td></td><td>Variable</td><td>Operating mode</td><td>Protection area type [1]</td></tr>
<tr><td></td><td>A_CA_DisableAutoMatch (DB10.DBX58.0)</td><td rowspan="4">Automatic</td><td>Machine (MACHINE)</td></tr>
<tr><td></td><td>A_CA_DisableAutoTools (DB10.DBX58.1)</td><td>Tools (TOOL)</td></tr>
<tr><td></td><td>A_CA_DisableAutoWork (DB10.DBX58.2)</td><td>Workholder (FIXTURE)</td></tr>
<tr><td></td><td>A_CA_DisableAutoWork (DB10.DBX58.3)</td><td>Workpieces (WORKPIECE)</td></tr>
<tr><td></td><td>A_CA_DisableJogMach (DB10.DBX58.4)</td><td rowspan="4">Jog</td><td>Machine (MACHINE)</td></tr>
<tr><td></td><td>A_CA_DisableJogTools (DB10.DBX58.5)</td><td>Tools (TOOL)</td></tr>
<tr><td></td><td>A_CA_DisableJogWorkh (DB10.DBX58.6)</td><td>Workholder (FIXTURE)</td></tr>
<tr><td></td><td>A_CA_DisableJogWorkp (DB10.DBX58.7)</td><td>Workpieces (WORKPIECE)</td></tr>
<tr><td></td><td colspan="3">[1] Type of a protection area ($NP_PROT_TYPE)
Note
A protection area group is deactivated via the SINUMERIK Operate user interface in the "AUTOMATIC", "JOG" or "MDI" > "ETC key (">")" > "Settings" > "Collision avoidance" > "Switch collision avoidance on and off" operating area by setting the HMI/PLC interface signal "LBP_NC.E_CA_Disable..." (DB10.DBX93.0, ...) of the selected protection area group.
Depending on the value of the "LBP_ConfigBP [FC1]" parameter <code>MMCToIf</code>, the interface signal is transferred from the PLC basic program to the NC/PLC interface signal "LBP_NC.A_CA_Disable..." (DB10.DBX58.0, ...):
• TRUE: Transmission
• FALSE: No transmission
The default value of the parameter is TRUE.
Note
If manual traversing has been enabled in AUTOMATIC mode (MD10735 with bit 0 == 1: allow jogging in automatic mode) the settings for JOG mode also apply in AUTOMATIC mode during manual traversing.</td></tr>
<tr><td>Signal flow</td><td colspan="3">PLC → NC</td></tr>
<tr><td>Data type</td><td colspan="3">BOOL</td></tr>
<tr><td>Update</td><td colspan="3">Cyclic</td></tr>
<tr><td>Value FALSE</td><td colspan="3">The deactivation of all protection areas of the protection area type in the selected operating mode is not requested.</td></tr>
<tr><td>Value TRUE</td><td colspan="3">The deactivation of all protection areas of the protection area type in the selected operating mode is requested.</td></tr>
</table>

Corresponds to	• "LBP_NC.E_CA_Disable..." (DB10.DBX93.0, ...) Collision avoidance: Deactivate protection area group HMI → PLC • $NP_PROT_TYPE Type of the protection area • MD10735 $MN_JOG_MODE_MASK Settings for JOG mode
Additional references	Operating Manual; Section "Collision avoidance" Function Manual Monitoring and Compensating; Section "Geometric machine modeling" > "Commissioning" > "System variables: Protection areas" > "$NP_PROT_TYPE" Function Manual Monitoring and Compensating; Section "Collision avoidance"

E_ActVal_In1, E_ActVal_In2, E_ActVal_In3, E_ActVal_In4

(DB10 DBX60.0, 60.1, ..., 60.3)

E_ActVal_In9, E_ActVal_In10, E_ActVal_In11, E_ActVal_In12, E_ActVal_In13, E_ActVal_In14, E_ActVal_In15, E_ActVal_In16

(DB10 DBX186.0, 186.1, ..., 186.7)

E_ActVal_In17, E_ActVal_In18, E_ActVal_In19, E_ActVal_In20, E_ActVal_In21, E_ActVal_In22, E_ActVal_In23, E_ActVal_In24

(DB10 DBX187.0, 187.1, ..., 187.7)

E_ActVal_In25, E_ActVal_In26, E_ActVal_In27, E_ActVal_In28, E_ActVal_In29, E_ActVal_In30, E_ActVal_In31, E_ActVal_In32

(DB10 DBX188.0, 188.1, ..., 188.7)

E_ActVal_In33, E_ActVal_In34, E_ActVal_In35, E_ActVal_In36, E_ActVal_In37, E_ActVal_In38, E_ActVal_In39, E_ActVal_In40

(DB10 DBX189.0, DB10 DBX189.1, ..., DB10 DBX189.7)

Description	Digital NC inputs 1 ... 40: Actual value Input 1 ... 4: Onboard input Input 9 ... 40: External digital NC input The actual NC output value can be read in the PLC user program via the actual value. **Note** The value in the "Actual value" interface can, as a result of the various subsequent influencing possibilities "Lock" and "Set", have a different value than that available at the NC output.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The actual value is 0.
Value TRUE	The actual value is 1.
Corresponds to	• "LBP_NC.A_Disabl_Inp..." (DB10 DBB0, 122, 124, 126, 128) Digital NC inputs: Lock • "LBP_NC.A_Set_Inp..." (DB10 DBB1, 123, 125, 127, 129) Digital NC inputs: Set • "LBP_NC.E_ActVal_In..." (DB10 DBB60, 186, 187, 188, 189) Digital NC inputs: Actual value • MD10350 $MN_FASTIO_DIG_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O"

E_Setpoint_Out1, E_Setpoint_Out2, E_Setpoint_Out3, E_Setpoint_Out4, E_Setpoint_Out5, E_Setpoint_Out6, E_Setpoint_Out7, E_Setpoint_Out8

(DB10 DBX64.0, 64.1, ..., 64.7)

E_Setpoint_Out9, E_Setpoint_Out10, E_Setpoint_Out11, E_Setpoint_Out12, E_Setpoint_Out13, E_Setpoint_Out14, E_Setpoint_Out15, E_Setpoint_Out16

(DB10 DBX190.0, 190.1, ..., 190.7)

E_Setpoint_Out17, E_Setpoint_Out18, E_Setpoint_Out19, E_Setpoint_Out20, E_Setpoint_Out21, E_Setpoint_Out22, E_Setpoint_Out23, E_Setpoint_Out24

(DB10 DBX191.0, 191.1, ..., 191.7)

E_Setpoint_Out25, E_Setpoint_Out26, E_Setpoint_Out27, E_Setpoint_Out28, E_Setpoint_Out29, E_Setpoint_Out30, E_Setpoint_Out31, E_Setpoint_Out32

(DB10 DBX192.0, 192.1, ..., 192.7)

E_Setpoint_Out33, E_Setpoint_Out34, E_Setpoint_Out35, E_Setpoint_Out36, E_Setpoint_Out37, E_Setpoint_Out38, E_Setpoint_Out39, E_Setpoint_Out40

(DB10 DBX193.0, 193.1, ..., 193.7)

Description	Digital NC outputs 1 ... 40: Setpoint Output 1 ... 4: Onboard output Output 5 ... 8: Output without hardware Output 9 ... 40: External digital NC output The actual NC output value can be read in the PLC user program via the setpoint. **Note** The value in the "Setpoint" interface can, as a result of the various subsequent influencing possibilities "Target" and "Lock", have a different value than that available at the NC output.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The setpoint is 0.
Value TRUE	The setpoint is 1.
Corresponds to	• "LBP_NC.A_Disabl_Out..." (DB10 DBB4, 130, 134, 138, 142) Digital NC outputs: Lock • "LBP_NC.A_OvMask_Out..." (DB10 DBB5, 131, 135, 139, 143) Digital NC outputs: Overwrite screen form • "LBP_NC.A_Set_Out..." (DB10 DBB6, 132, 136, 140, 144) Digital NC outputs: Setting value from the PLC • MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O"

E_HW1_moved, E_HW2_moved, E_HW3_moved, E_HW4_moved, E_HW5_moved, E_HW6_moved

(DB10 DBB68, 69, 70, 242, 243, 244)

Description	Handwheel is operated (handwheel 1 ... 6)
Signal flow	NC → PLC
Data type	BYTE

E_CounterInchMetr (DB10.DBB71)

Description	Change counter, inch/metric system of units
Signal flow	NC → PLC
Data type	BYTE

E_MCP1AxisValid (DB10.DBX72.0)

Description	Status of the displayed actual value screen (1st MCP) - display valid
Signal flow	NC → PLC (HT 8 → operating software)
Data type	BOOL

E_MCP1WCS (DB10.DBX72.1)

Description	Status of the displayed actual value screen (1st MCP) - MCS / WCS active
Signal flow	NC → PLC (HT 8 → operating software)
Data type	BOOL

E_MCP1AxKeyDisp (DB10.DBX72.7)

Description	Status of the displayed actual value screen (1st MCP) - traversing keys displayed
Signal flow	NC → PLC (HT 8 → operating software)
Data type	BOOL

E_MCP2AxisValid (DB10.DBX73.0)

Description	Status of the displayed actual value screen (2nd MCP) - display valid
Signal flow	NC → PLC (HT 8 → operating software)
Data type	BOOL

E_MCP2WCS (DB10.DBX73.1)

Description	Status of the displayed actual value screen (2nd MCP) - MCS / WCS active
Signal flow	NC → PLC (HT 8 → operating software)
Data type	BOOL

E_MCP2AxKeyDispAktiv (DB10.DBX73.7)

Description	Status of the displayed actual value screen (2nd MCP) - traversing keys displayed
Signal flow	NC → PLC (HT 8 → operating software)
Data type	BOOL

MCP1AxisFromHMI (DB10 from DBB74)

Description	Machine axis numbers of the displayed axes (1st MCP)
Signal flow	NC → PLC (HT 8 → operating software)
Data type	ARRAY[1..6] of BYTE

MCP2AxisFromHMI (DB10 from DBB80)

Description	Machine axis numbers of the displayed axes (2nd MCP)
Signal flow	NC → PLC (HT 8 → operating software)
Data type	ARRAY[1..6] of BYTE

epsToPlc (DB10.DBB90)

Signal flow	NC → PLC (SINUMERIK Integrate → PLC)
Data type	BYTE

epsFromPlc (DB10.DBB91)

Signal flow	NC → PLC (PLC → SINUMERIK Integrate)
Data type	BYTE

MpiDpOk (DB10.DBX92.0)

Description	Slave OK - MPI/DP bus
Signal flow	NC → PLC (GP → PLC)
Data type	BOOL

DpOk (DB10.DBX92.1)

Description	Slave OK - DP1 bus
Signal flow	NC → PLC (GP → PLC)
Data type	BOOL

PnOk (DB10.DBX92.2)

Description	Slave OK - PN bus
Signal flow	NC → PLC (GP → PLC)
Data type	BOOL

Disabl_AlarmMpiDp (DB10.DBX92.4)

Description	Suppress error message in case of failure - MPI/DP bus
Signal flow	NC → PLC (GP → PLC)
Data type	BOOL

Disabl_AlarmDp (DB10.DBX92.5)

Description	Suppress error message in case of failure - DP1 bus
Signal flow	NC → PLC (GP → PLC)
Data type	BOOL

Disabl_AlarmPn (DB10.DBX92.6)

Description	Suppress error message in case of failure - PN bus
Signal flow	NC → PLC (GP → PLC)
Data type	BOOL

E_CA_DisableAutoMach, E_CA_DisableAutoTools, E_CA_DisableAutoWorkh, E_CA_DisableAutoWorkp, E_CA_DisableJogMach, E_CA_DisableJogTools, E_CA_DisableJogWorkh, E_CA_DisableJogWorkp

(DB10 DBX93.0, 93.1, ..., 93.7)

Description	Collision avoidance: Deactivate protection area group		
	Variable	**Operating mode**	**Protection area type** [1]
	E_CA_DisableAutoMatch (DB10.DBX93.0)	Automatic	Machine (MACHINE)
	E_CA_DisableAutoTools (DB10.DBX93.1)		Tools (TOOL)
	E_CA_DisableAutoWorkh (DB10.DBX93.2)		Workholder (FIXTURE)
	E_CA_DisableAutoWorkp (DB10.DBX93.3)		Workpieces (WORKPIECE)
	E_CA_DisableJogMach (DB10.DBX93.4)	Jog	Machine (MACHINE)
	E_CA_DisableJogTools (DB10.DBX93.5)		Tools (TOOL)
	E_CA_DisableJogWorkh (DB10.DBX93.6)		Workholder (FIXTURE)
	E_CA_DisableJogWorkp (DB10.DBX93.7)		Workpieces (WORKPIECE)
	[1] Type of a protection area ($NP_PROT_TYPE) **Note** A protection area group is deactivated via the SINUMERIK Operate user interface in the "AUTOMATIC", "JOG" or "MDI" > "ETC key (">")" > "Settings" > "Collision avoidance" > "Switch collision avoidance on and off" operating area by setting the HMI/PLC interface signal "LBP_NC.E_CA_Disable..." (DB10.DBX93.0, ...) of the selected protection area group. Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program to the NC/PLC interface signal "LBP_NC.A_CA_Disable..." (DB10.DBX58.0, ...): • TRUE: Transmission • FALSE: No transmission The default value of the parameter is TRUE. **Note** If manual traversing has been enabled in AUTOMATIC mode (MD10735 with bit 0 == 1: allow jogging in automatic mode) the settings for JOG mode also apply in AUTOMATIC mode during manual traversing.		
Signal flow	NC → PLC (operating software → PLC)		
Data type	BOOL		
Update	Cyclic		

Value FALSE	The deactivation of all protection areas of the protection area type in the selected operating mode is **not** requested.
Value TRUE	The deactivation of all protection areas of the protection area type in the selected operating mode is requested.
Corresponds to	• "LBP_NC.A_CA_Disable..." (DB10.DBX58.0, ...) Collision avoidance: Deactivate protection area group • $NP_PROT_TYPE Type of protection area • MD10735 $MN_JOG_MODE_MASK Settings for JOG mode
Additional references	Operating Manual; Section "Collision avoidance" Function Manual Monitoring and Compensating; Section "Geometric machine modeling" > "Commissioning" > "System variables: Protection areas" > "$NP_PROT_TYPE" Function Manual Monitoring and Compensating; Section "Collision avoidance"

E_MCP1LostPackageWarn, E_MCP2LostPackageWarn

(DB10 DBX94.0, 94.1)

Description	Warning of connection termination after failure of the cyclic communication for MCP1/MCP2
Signal flow	BP → PLC
Data type	BOOL
Update	Cyclic

E_HTLostPackageWarn (DB10 DBX94.2)

Description	Warning of connection termination after failure of the cyclic communication for HT
Signal flow	BP → PLC
Data type	BOOL
Update	Cyclic

E_Op1KeyLostPackageWarn, E_Op2KeyLostPackageWarn

(DB10 DBX94.3, 94.4)

Description	Warning of connection termination after failure of the cyclic communication for Op1Key/ Op2Key
Signal flow	BP → PLC
Data type	BOOL
Update	Cyclic

E_LanguageID (DB10.DBB96)

Description	Set language ID of the operating software
Signal flow	NC → PLC (operating software → PLC)
Data type	BYTE

E_Chan_HW1_A, E_Chan_HW1_B, E_Chan_HW1_C, E_Chan_HW1_D

(DB10 DBX97.0, 97.1, 97.2, 97.3)

E_Chan_HW2_A, E_Chan_HW2_B, E_Chan_HW2_C, E_Chan_HW2_D

(DB10 DBX98.0, 98.1, 98.2, 98.3)

E_Chan_HW3_A, E_Chan_HW3_B, E_Chan_HW3_C, E_Chan_HW3_D

(DB10 DBX99.0, 99.1, 99.2, 99.3)

<table>
<tr><td>Description</td><td>Channel number for the handwheels
Handwheel 1: "E_Chan_HW1_..."
Handwheel 2: "E_Chan_HW2_..."
Handwheel 3: "E_Chan_HW3_..."
The operator can assign an axis to the handwheel (1, 2, 3) directly at the operator panel. If this axis is a geometry axis (interface signal "Machine axis handwheel <n>" = 0), then the PLC basic program provides the corresponding channel number at the HMI interface as binary-coded value.
Example for handwheel 1
<table>
<tr><th>E_Chan_HW1_D (DB10.DBX97.3)</th><th>E_Chan_HW1_C (DB10.DBX97.2)</th><th>E_Chan_HW1_B (DB10.DBX97.1)</th><th>E_Chan_HW1_A (DB10.DBX97.0)</th><th>Channel number</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
</table>
Note
For machine axes (interface signal "Machine axis handwheel <n>" = 1), the interface signal "Channel number geometry axis handwheel <n>" has no significance.</td></tr>
<tr><td>Signal flow</td><td>NC → PLC (operating software → PLC)</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
</table>

Corresponds to	• "LBP_NC.E_Chan_HW1_..." (DB10 DBX97.0, 97.1, 97.2, 97.3) Channel number geometry axis, handwheel 1 • "LBP_NC.E_Chan_HW2_..." (DB10 DBX98.0, 98.1, 98.2, 98.3) Channel number geometry axis, handwheel 2 • "LBP_NC.E_Chan_HW3_..." (DB10 DBX99.0, 99.1, 99.2, 99.3) Channel number geometry axis, handwheel 3 • "LBP_NC.E_Ax_..._HW1" (DB10 DBX100.0, 100.1, 100.2, 100.3, 100.4) Axis number handwheel 1 • "LBP_NC.E_Ax_..._HW2" (DB10 DBX101.0, 101.1, 101.2, 101.3, 101.4) Axis number handwheel 2 • "LBP_NC.E_Ax_..._HW3" (DB10 DBX102.0, 102.1, 102.2, 102.3, 102.4) Axis number handwheel 3 • "LBP_NC.E_HW_sel1" (DB10.DBX100.6) Handwheel 1 selected • "LBP_NC.E_HW_sel2" (DB10.DBX101.6) Handwheel 2 selected • "LBP_NC.E_HW_sel3" (DB10.DBX102.6) Handwheel 3 selected • "LBP_NC.E_Mach_Ax1" (DB10.DBX100.7) Machine axis handwheel 1 • "LBP_NC.E_Mach_Ax2" (DB10.DBX101.7) Machine axis handwheel 2 • "LBP_NC.E_Mach_Ax3" (DB10.DBX102.7) Machine axis handwheel 3 • "LBP_Chan*.A_Geo[1].HW..." (DB21, ... DBX12.0, 12.1, 12.2) Geometry axis 1: Activate handwheel • "LBP_Chan*.A_Geo[2].HW..." (DB21, ... DBX16.0, 16.1, 16.2) Geometry axis 2: Activate handwheel • "LBP_Chan*.A_Geo[3].HW..." (DB21, ... DBX20.0, 20.1, 20.2) Geometry axis 3: Activate handwheel • "LBP_Axis*.A_HW..." (DB31, ... DBX4.0, 4.1, 4.2) Activate handwheel
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_Ax_A_HW1, E_Ax_B_HW1, E_Ax_C_HW1, E_Ax_D_HW1, E_Ax_E_HW1

(DB10 DBX100.0, 100.1, 100.2, 100.3, 100.4)

E_Ax_A_HW2, E_Ax_B_HW2, E_Ax_C_HW2, E_Ax_D_HW2, E_Ax_E_HW2

(DB10 DBX101.0, 101.1, 101.2, 101.3, 101.4)

E_Ax_A_HW3, E_Ax_B_HW3, E_Ax_C_HW3, E_Ax_D_HW3, E_Ax_E_HW3

(DB10 DBX102.0, 102.1, 102.2, 102.3, 102.4)

<table>
<tr><td>Description</td><td>Axis number for the handwheels
Handwheel 1: "E_Ax_..._HW1"
Handwheel 2: "E_Ax_..._HW2"
Handwheel 3: "E_Ax_..._HW3"
The operator can assign an axis to the handwheel (1, 2, 3) directly at the operator panel. To do this, it specifies the requested axis (e.g. X). The PLC basic program provides the axis number, associated with the axis, at the HMI interface as a binary-coded value.
Example for handwheel 1
<table>
<tr><th>E_Ax_E_HW1 (DB10.DBX100.4)</th><th>E_Ax_D_HW1 (DB10.DBX100.3)</th><th>E_Ax_C_HW1 (DB10.DBX100.2)</th><th>E_Ax_B_HW1 (DB10.DBX100.1)</th><th>E_Ax_A_HW1 (DB10.DBX100.0)</th><th>Axis number</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
</table>
The following applies when assigning the axis identifier to the axis number:
• Machine axis (interface signal "Machine axis" = 1):
→ The assignment is realized using machine data MD10000.
• Geometry axis (interface signal "Machine axis" = 0):
→ The assignment is realized using machine data MD20060.</td></tr>
<tr><td>Signal flow</td><td>NC → PLC (operating software → PLC)</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
</table>

Corresponds to	• "LBP_NC.E_Ax_..._HW1" (DB10 DBX100.0, 100.1, 100.2, 100.3, 100.4) Axis number handwheel 1 • "LBP_NC.E_Ax_..._HW2" (DB10 DBX101.0, 101.1, 101.2, 101.3, 101.4) Axis number handwheel 2 • "LBP_NC.E_Ax_..._HW3" (DB10 DBX102.0, 102.1, 102.2, 102.3, 102.4) Axis number handwheel 3 • "LBP_NC.E_Chan_HW1_..." (DB10 DBX97.0, 97.1, 97.2, 97.3) Channel number geometry axis, handwheel 1 • "LBP_NC.E_Chan_HW2_..." (DB10 DBX98.0, 98.1, 98.2, 98.3) Channel number geometry axis, handwheel 2 • "LBP_NC.E_Chan_HW3_..." (DB10 DBX99.0, 99.1, 99.2, 99.3) Channel number geometry axis, handwheel 3 • "LBP_NC.E_HW_sel1" (DB10.DBX100.6) Handwheel 1 selected • "LBP_NC.E_HW_sel2" (DB10.DBX101.6) Handwheel 2 selected • "LBP_NC.E_HW_sel3" (DB10.DBX102.6) Handwheel 3 selected • "LBP_NC.E_Mach_Ax1" (DB10.DBX100.7) Machine axis handwheel 1 • "LBP_NC.E_Mach_Ax2" (DB10.DBX101.7) Machine axis handwheel 2 • "LBP_NC.E_Mach_Ax3" (DB10.DBX102.7) Machine axis handwheel 3 • "LBP_Chan*.A_Geo[1].HW..." (DB21, ... DBX12.0, 12.1, 12.2) Geometry axis 1: Activate handwheel • "LBP_Chan*.A_Geo[2].HW..." (DB21, ... DBX16.0, 16.1, 16.2) Geometry axis 2: Activate handwheel • "LBP_Chan*.A_Geo[3].HW..." (DB21, ... DBX20.0, 20.1, 20.2) Geometry axis 3: Activate handwheel • "LBP_Axis*.A_HW..." (DB31, ... DBX4.0, 4.1, 4.2) Activate handwheel • MD10000 $MN_AXCONF_MACHAX_NAME_TAB [<n>] Machine axis name • MD20060 $MC_AXCONF_GEOAX_NAME_TAB [<n>] Geometry axis name in the channel
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_CHW_sel1, E_CHW_sel2, E_CHW_sel3

(DB10 DBX100.5, 101.5, 102.5)

Description	Define handwheel as contour handwheel Handwheel 1: "E_CHW_sel1" (DB10.DBX100.5) Handwheel 2: "E_CHW_sel2" (DB10.DBX101.5) Handwheel 3: "E_CHW_sel3" (DB10.DBX102.5) For the handwheel defined via the user interface also to be active as contour handwheel, signal "Define handwheel <n> as contour handwheel" must be linked to signal "Activate handwheel <n> as contour handwheel". **Note** Depending on the settings of parameter "HWheelMMC" in "LBP_ConfigBP [FC1]" of the PLC basic program, these signals are either supplied by the basic program or must be supplied by the PLC user program.
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	The handwheel is not defined as contour handwheel.
Value TRUE	The handwheel is defined as contour handwheel via the user interface.
Corresponds to	• "LBP_Chan*.A_ContHW1" (DB21, ... DBX30.0) Activate contour handwheel 1 • "LBP_Chan*.A_ContHW2" (DB21, ... DBX30.1) Activate contour handwheel 2 • "LBP_Chan*.A_ContHW3" (DB21, ... DBX30.2) Activate contour handwheel 3 • "LBP_ConfigBP [FC1]" parameter "HWheelMMC"
Additional references	Function Manual for PLC Function Manual Axes and Spindles; Section "Manual traversing"

E_HW_sel1, E_HW_sel2, E_HW_sel3

(DB10 DBX100.6, 101.6, 102.6)

Description	Handwheel selected Handwheel 1: "E_HW_sel1" (DB10.DBX100.6) Handwheel 2: "E_HW_sel2" (DB10.DBX101.6) Handwheel 3: "E_HW_sel3" (DB10.DBX102.6) The PLC basic program provides this interface signal at the HMI interface if, at the operator panel, the operator selects or deselects handwheel 1 for the specified axis, i.e. enables or disables for activation. The information is transferred from the PLC basic program to the PLC (precondition: "LBP_ConfigBP [FC1]" parameter "HWheelMMC" == "TRUE") and the interface signal "Activate handwheel" is appropriately set for the specified axis. Depending on the setting in the HMI interface signal "machine axis", either the interface for the geometry axis or for the machine axis is used.
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	Disable handwheel.
Value TRUE	Enable handwheel for activation.

Corresponds to	• "LBP_NC.E_Chan_HW1_..." (DB10 DBX97.0, 97.1, 97.2, 97.3) Channel number geometry axis, handwheel 1 • "LBP_NC.E_Chan_HW2_"... (DB10 DBX98.0, 98.1, 98.2, 98.3) Channel number geometry axis, handwheel 2 • "LBP_NC.E_Chan_HW3_..." (DB10 DBX99.0, 99.1, 99.2, 99.3) Channel number geometry axis, handwheel 3 • "LBP_NC.E_Ax_..._HW1" (DB10 DBX100.0, 100.1, 100.2, 100.3, 100.4) Axis number handwheel 1 • "LBP_NC.E_Ax_..._HW2" (DB10 DBX101.0, 101.1, 101.2, 101.3, 101.4) Axis number handwheel 2 • "LBP_NC.E_Ax_..._HW3" (DB10 DBX102.0, 102.1, 102.2, 102.3, 102.4) Axis number handwheel 3 • "LBP_NC.E_Mach_Ax1" (DB10.DBX100.7) Machine axis handwheel 1 • "LBP_NC.E_Mach_Ax2" (DB10.DBX101.7) Machine axis handwheel 2 • "LBP_NC.E_Mach_Ax3" (DB10.DBX102.7) Machine axis handwheel 3 • "LBP_Chan*.A_Geo[1].HW..." (DB21, ... DBX12.0, 12.1, 12.2) Geometry axis 1: Activate handwheel • "LBP_Chan*.A_Geo[2].HW..." (DB21, ... DBX16.0, 16.1, 16.2) Geometry axis 2: Activate handwheel • "LBP_Chan*.A_Geo[3].HW..." (DB21, ... DBX20.0, 20.1, 20.2) Geometry axis 3: Activate handwheel • "LBP_Axis*.A_HW..." (DB31, ... DBX4.0, 4.1, 4.2) Activate handwheel • "LBP_ConfigBP [FC1]" parameter "HWheelMMC"
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_Mach_Ax1, E_Mach_Ax2, E_Mach_Ax3

(DB10 DBX100.7, 101.7, 102.7)

Description	Machine axis handwheel Handwheel 1: "E_Mach_Ax1" (DB10.DBX100.7) Handwheel 2: "E_Mach_Ax2" (DB10.DBX101.7) Handwheel 3: "E_Mach_Ax3" (DB10.DBX102.7) The PLC basic program provides this interface signal at the HMI interface if the operator has assigned an axis to the handwheel (1, 2, 3) directly at the operator panel.
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	Axis is a geometry axis.
Value TRUE	Axis is a machine axis.

Corresponds to	• "LBP_NC.E_Chan_HW1_..." (DB10 DBX97.0, 97.1, 97.2, 97.3) Channel number geometry axis, handwheel 1 • "LBP_NC.E_Chan_HW2_..." (DB10 DBX98.0, 98.1, 98.2, 98.3) Channel number geometry axis, handwheel 2 • "LBP_NC.E_Chan_HW3_..." (DB10 DBX99.0, 99.1, 99.2, 99.3) Channel number geometry axis, handwheel 3 • "LBP_NC.E_Ax_..._HW1" (DB10 DBX100.0, 100.1, 100.2, 100.3, 100.4) Axis number handwheel 1 • "LBP_NC.E_Ax_..._HW2" (DB10 DBX101.0, 101.1, 101.2, 101.3, 101.4) Axis number handwheel 2 • "LBP_NC.E_Ax_..._HW3" (DB10 DBX102.0, 102.1, 102.2, 102.3, 102.4) Axis number handwheel 3 • "LBP_NC.E_HW_sel1" (DB10.DBX100.6) Handwheel 1 selected • "LBP_NC.E_HW_sel2" (DB10.DBX101.6) Handwheel 2 selected • "LBP_NC.E_HW_sel3" (DB10.DBX102.6) Handwheel 3 selected
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_RemDiag (DB10.DBX103.0)

Description	Remote diagnosis active
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	Remote diagnosis is not active.
Value TRUE	Remote diagnosis (optional) is active, i.e. the control is operated via an external PC.

E_ATready (DB10.DBX103.5)

Description	AT box ready
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	The AT box is not ready. An expansion module conforming to the AT specification has either no functionality or restricted functionality.
Value TRUE	The AT box for expansion modules is ready.

E_MMCTemp (DB10.DBX103.6)

Description	HMI temperature limit
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	The temperature is **outside** the permissible tolerance range from 5 to 55 °C. The temperature monitoring has responded and the PCU has been disabled.
Value TRUE	The temperature is **within** the permissible tolerance range from 5 to 55 °C.

E_MMCBatt (DB10.DBX103.7)

Description	HMI battery alarm
Signal flow	NC → PLC (operating software → PLC)
Data type	BOOL
Update	Cyclic
Value FALSE	No HMI battery alarm is present.
Value TRUE	The battery monitor has responded. Power failure can cause the loss of recently changed data and a correct device configuration. An appropriate alarm is issued. The buffer battery should be checked. An insufficient battery voltage also affects the current time on the user interface.
Additional references	Operator Components Manual (PCU)

E_MCP1ready, E_MCP2ready

(DB10 DBX104.0, 104.1)

Description	MCP1/MCP2 ready
Signal flow	GP → PLC
Data type	BOOL
Update	Cyclic

E_HTready (DB10.DBX104.2)

Description	HT ready
Signal flow	GP → PLC
Data type	BOOL
Update	Cyclic

E_OP1KeyReady, E_OP2KeyReady

(DB10 DBX104.3, 104.4)

Description	Op1Key/Op2Key ready
Signal flow	GP → PLC
Data type	BOOL
Update	Cyclic

E_FirstOB1 (DB10.DBX104.6)

Description	First Main [OB1] cycle
Signal flow	GP → PLC
Data type	BOOL
Update	Cyclic

E_NCKready (DB10.DBX104.7)

Description	NC CPU ready The signal should be included in the machine safety circuit.
Signal flow	GP → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	NC CPU is not ready for operation. The following measures are started by the PLC basic program: • The status signals of the NC to the PLC (user interface) are deleted • The change signals of auxiliary functions are deleted • Cyclic processing of the user interface PLC to NC is terminated.
Value TRUE	The NC CPU is ready and registers itself cyclically with the PLC. After a correct initial start and the first complete "Main [OB1]" cycle, the PLC and NC continuously exchange sign-of-life signals.
Additional references	• Diagnostics Manual • Function Manual for PLC

E_CancelToolCmd (DB10.DBX105.0)

Description	Cancel TM command
Signal flow	GP → PLC
Data type	BOOL
Update	Cyclic

E_CollCheckOff (DB10.DBX106.0)

Description	Collision avoidance - protection area monitoring active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

E_EMERGENCY (DB10.DBX106.1)

Description	Emergency stop active To acknowledge the "Emergency Stop" state of the NC, the following interface signals must remain set until interface signal "LBP_NC.E_EMERGENCY" (DB10.DBX106.1) has been reset: • "LBP_NC.A_EMERGENCY_Ackn" (DB10.DBX56.2) = 1 (acknowledge emergency stop) • "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) = 1 (HT reset) for all HTs of the NC
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Emergency Stop is not active.
Value TRUE	Emergency Stop is active.
Corresponds to	• "LBP_NC.A_EMERGENCY" (DB10.DBX56.1) Emergency stop • "LBP_NC.A_EMERGENCY_Ackn" (DB10.DBX56.2) Acknowledge emergency stop • "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) Mode group reset
Additional references	Function Manual Basic Functions; Section "Emergency stop"

E_InspProbe1, E_InspProbe2

(DB10 DBX107.0, 107.1)

Description	Probe actuated Probe 1: "E_InspProbe1" Probe 2: "E_InspProbe2"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Probe not deflected.
Value TRUE	Probe deflected.
Additional references	Function Manual Technologies; Section "Measuring"

E_NCU_LinkActive (DB10.DBX107.6)

Description	NCU link: Active The signal is irrelevant for a system with an NCU.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	NCU link communication is not active.
Value TRUE	NCU link communication is active.
Additional references	NCU Equipment Manual

E_SystemInchDim (DB10.DBX107.7)

Description	Inch system of units
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

E_MMC2Ready (DB10.DBX108.1)

Description	Operator panel 2 "ready"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

E_MMC2MPIready (DB10.DBX108.2)

Description	Operator panel at MPI "ready"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

E_MMC2BTSSready (DB10.DBX108.3)

Description	Operating software ready
Signal flow	NC → PLC
Data type	BOOL

Update	Cyclic
Value FALSE	SINUMERIK Operate is not ready.
Value TRUE	SINUMERIK Operate is ready and registers itself cyclically with the NC.
Additional references	Diagnostics Manual

E_DrivesInCylOp (DB10.DBX108.5)

Description	Drives in cyclic operation
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	For at least one machine axis of the NC, the corresponding drive is not in cyclic operation, i.e. it is not cyclically exchanging PROFIdrive telegrams with the NC.
Value TRUE	For all machine axes of the NC, their drives are in cyclic operation, i.e. they cyclically exchange PROFIdrive telegrams with the NC.

E_611Dready (DB10.DBX108.6)

Description	Drives ready
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	For at least one machine axis of the NC, the corresponding drive is not ready: "LBP_Axis*.E_MotOK" (DB31, ... DBX93.5) == 0 (DRIVE ready)
Value TRUE	For all machine axes of the NC, the corresponding drives are ready: "LBP_Axis*.E_MotOK" (DB31, ... DBX93.5) == 1 (DRIVE ready)
Corresponds to	• "LBP_Axis*.E_MotOK" (DB31, ... DBX93.5) Activate handwheel

E_NCready (DB10.DBX108.7)

Description	NC Ready
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	The control is not ready. The signal is reset when the relay contact "NC ready" is opened. Possible causes: • The undervoltage or overvoltage monitoring function has responded • Individual components are not ready (NC CPU ready) • The NC CPU watchdog has responded If still possible, the following internal control responses occur: • NC: The controller enable signals are withdrawn ⇒ the drives stop • PLC basic program: – The status signals from the NC to the PLC (user interface) are deleted – The change signals of the auxiliary functions are deleted – Cyclic processing of the user interface PLC to NC is terminated Remedy: At least a warm restart must be initiated to exit the fault state.
Value TRUE	The control is ready. The interface signal is an image of the relay contact "NC Ready". The signal is set when: • The relay contact "NC Ready" is closed • All the voltages in the control have been built up • The control is in the condition "cyclic mode"
Corresponds to	• "LBP_NC.E_NCKready" (DB10.DBX104.7) NC-CPU ready
Additional references	• Diagnostics Manual • Function Manual for PLC

E_NCKalarm (DB10.DBX109.0)

Description	NC alarm pending The interface signal is a combination of interface signals of all of the existing channels: "LBP_Chan*.E_NCKalarmChan" (DB21, ... DBX36.6) (channel-specific NC alarm active)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No NC alarm is active.
Value TRUE	At least one NC alarm is active.
Corresponds to	• "LBP_Chan*.E_NCKalarmChan" (DB21, ... DBX36.6) Channel-specific NC alarm active • "LBP_Chan*.E_NCKalarmStop" (DB21 ... DBX36.7) NC alarm with machining stop pending
Additional references	Diagnostics Manual

A_PC_OSfault (DB10.DBX109.4)

Description	PC operating system error
Signal flow	NC → PLC

Data type	BOOL
Update	Cyclic

E_HTempAlarm (DB10.DBX109.5)

Description	NCU heat sink temperature alarm
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The heat sink temperature limit of the NCU has not been exceeded.
Value TRUE	The heat sink temperature limit of the NCU has been exceeded. Continuous NCU operation can no longer be guaranteed.

E_ATempAlarm (DB10.DBX109.6)

Description	Air temperature alarm
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The monitoring of the ambient temperature and the fan monitoring has not responded.
Value TRUE	The monitoring of the ambient temperature and/or the fan monitoring has responded. Possible causes: • Ambient temperature too high • The speed monitoring of the 24 VDC fan used to cool the module has responded. Possible remedy: Replace the fan and/or ensure that additional ventilation/cooling is provided.
Corresponds to	Relay contacts of the infeed/regenerative feedback unit: Terminal 5.1, 5.2 or 5.1, 5.3)
Additional references	Diagnostics Manual

E_BattAlarm (DB10.DBX109.7)

Description	NC battery alarm The NC battery should only be replaced while the NC is switched on to avoid data loss as there is no memory backup.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The battery voltage is higher than the lower limit value (normal situation).
Value TRUE	The NC battery voltage monitoring function has responded. Possible causes: • The battery voltage is within the pre-warning limit range (approx. 2.7 to 2.9 V). • The battery voltage is below the pre-warning limit range (≤ 2.6 V). • When the controller powered-up, it was identified that the battery voltage was below the pre-warning limit range (≤ 2.6 V).
Corresponds to	Relay contacts of the infeed/regenerative feedback unit: Terminal 5.1, 5.2 or 5.1, 5.3)
Additional references	Diagnostics Manual NCU Equipment Manual

E_SWCamMinus (DB10.DBD110)

Description	Software cams minus cam signals 1 to 32 The switching edge of the minus cam signal 1 to 32 is generated depending on the traversing direction of the (rotary) axis, and is transferred in the IPO clock cycle to the PLC interface.
Signal flow	NC → PLC
Data type	ARRAY[0..31] of BOOL
Value FALSE	**Linear axis** The minus cam signal switches from 1 to 0 when the axis traverses the minus cam in the positive axis direction. **Modulo rotary axis** The minus cam signal changes level in response to every positive edge of the plus cam signal.
Value TRUE	**Linear axis** The minus cam signal switches from 0 to 1 when the axis passes over the minus cam in the negative direction. **Modulo rotary axis** The minus cam signal changes level in response to every positive edge of the plus cam signal.
Corresponds to	• "LBP_NC.E_SWCamPlus" (DB10.DBD114) Software cams: Plus output cam signal 1 to 32 • "LBP_Axis*.A_SWCam" (DB31, ... DBX2.0) Software cams: Activation
Additional references	Function Manual Axes and Spindles; Section "Software cams, cam sequencers"

E_SWCamPlus (DB10.DBD114)

Description	Software cams plus cam signals 1 to 32 The switching edge of the plus cam signal 1 to 32 is generated depending on the traversing direction of the (round) axis, and is transferred in the IPO clock cycle to the PLC interface. **Note** The described response of the plus cam for modulo rotary axes applies under the condition: plus cam - minus cam < 180 degrees If this condition is not fulfilled or if the minus cam is set to a greater value than the plus cam, then the behavior of the plus cam signal is inverted. The response of the minus cam signal remains unchanged.
Signal flow	NC → PLC
Data type	ARRAY[0..31] of BOOL
Value FALSE	**Linear axis** The plus cam signal switches from 1 to 0 when the axis traverses the plus cam in the negative direction. **Modulo rotary axis** When passing over the plus cam in the positive axis direction, the plus cam signal switches back from 1 to 0.
Value TRUE	**Linear axis** The plus cam signal switches from 0 to 1 when the axis traverses the plus cam in the positive direction. **Modulo rotary axis** When passing over the minus cam in the positive axis direction, the plus cam signal switches from 0 to 1.

Corresponds to	• "LBP_NC.E_SWCamMinus" (DB10.DBD110) Software cams: Minus output cam signal 1 to 32 • "LBP_Axis*.A_SWCam" (DB31, ... DBX2.0) Software cams: Activation
Additional references	Function Manual Axes and Spindles; Section "Software cams, cam sequencers"

A_Disabl_anaIn1, A_Disabl_anaIn2, A_Disabl_anaIn3, A_Disabl_anaIn4, A_Disabl_anaIn5, A_Disabl_anaIn6, A_Disabl_anaIn7, A_Disabl_anaIn8

(DB10 DBX146.0, 146.1, ..., 146.7)

Description	Disabling of the analog input 1 ... 8 **Note** When read, a locked input supplies a value of 0.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The analog NC input is enabled.
Value TRUE	The analog NC input is locked.
Corresponds to	• "LBP_NC.A_InMask_anaIn..." (DB10.DBB147) Analog NC inputs 1 ... 8: Default • "LBP_NC.A_Setval_anaIn..." (DB10 from DBW148) Analog NC inputs 1 ... 8: Setting value • "LBP_NC.E_ActVal_anaIn..." (DB10 from DBW194) Analog NC inputs 1 ... 8: Actual value • MD10300 $MN_FASTIO_ANA_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O access via PLC" > "Analog NC inputs/outputs" > "NC inputs"

A_InMask_anaIn1, A_InMask_anaIn2, A_InMask_anaIn3, A_InMask_anaIn4, A_InMask_anaIn5, A_InMask_anaIn6, A_InMask_anaIn7, A_InMask_anaIn8

(DB10 DBX147.0, 147.1, ..., 147.7)

Description	Setting of a setting value for analog input 1 ... 8 If the target of the setting value is activated, then the setting value is active as NC input value. If the setting value target is de-activated, then as NC input value, the analog value available at the NC input is active or for an active lock, the value of 0.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The target of the setting value is deactivated.
Value TRUE	The target of the setting value is activated.

Corresponds to	• "LBP_NC.A_Disabl_anaIn..." (DB10.DBB146) Analog NC inputs 1 ... 8: Lock • "LBP_NC.A_InMask_anaIn..." (DB10.DBB147) Analog NC inputs 1 ... 8: Default • "LBP_NC.A_Setval_anaIn..." (DB10 from DBW148) Analog NC inputs 1 ... 8: Setting value • "LBP_NC.E_ActVal_anaIn..." (DB10 from DBW194) Analog NC inputs 1 ... 8: Actual value • MD10300 $MN_FASTIO_ANA_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O access via PLC" > "Analog NC inputs/outputs" > "NC inputs"

A_Setval_anaIn1, A_Setval_anaIn2, A_Setval_anaIn3, A_Setval_anaIn4, A_Setval_anaIn5, A_Setval_anaIn6, A_Setval_anaIn7, A_Setval_anaIn8

(DB10 DBW148, 150, ..., 162)

Description	Setpoint for the analog input 1 ... 8 of the NC The setting value is transferred as NC input value when input "Target" "LBP_NC.A_InMask_anaIn..." (DB10.DBB147) is switched on. The setting value must be specified as a fixed-point number (16 bit value including sign) in 2's complement.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_NC.A_Disabl_anaIn..." (DB10.DBB146) Analog NC inputs 1 ... 8: Lock • "LBP_NC.A_InMask_anaIn..." (DB10.DBB147) Analog NC inputs 1 ... 8: Default • "LBP_NC.A_Setval_anaIn..." (DB10 from DBW148) Analog NC inputs 1 ... 8: Setting value • "LBP_NC.E_ActVal_anaIn..." (DB10 from DBW194) Analog NC inputs 1 ... 8: Actual value • MD10300 $MN_FASTIO_ANA_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O access via PLC" > "Analog NC inputs/outputs" > "NC inputs"

A_OvMask_anaOut1, A_OvMask_anaOut2, A_OvMask_anaOut3, A_OvMask_anaOut4, A_OvMask_anaOut5, A_OvMask_anaOut6, A_OvMask_anaOut7, A_OvMask_anaOut8

(DB10 DBX166.0, 166.1, ..., 166.7)

Description	Overwrite screen form of the analog NC output 1 ... 8 For a positive edge change 0 → 1, the setting value specified from the PLC user program is used for the corresponding output, instead of the value written using the system variable $A_OUTA. The value written via system variables $A_OUTA is then lost. For a negative edge change 1 → 0, for the corresponding output, the actual value at the hardware output is kept. **Note** The variable "A_Setval_anaOut1", ... (DB10.DBW170.0, ...) (analog NC outputs 1 ... 8: setting value) is jointly used by: • "A_OvMask_anaOut1", ... (DB10.DBX166.0, ...) (overwrite) for edge change 0 → 1 • "A_InMask_anaOut1", ... (DB10.DBX167.0, ...) (target) for signal state TRUE Activating the two interfaces simultaneously should be avoided.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	The associated "setting value" is activated.
Edge change 1 → 0	No effect.
Corresponds to	• "LBP_NC.A_InMask_anaOut..." (DB10.DBB167) Analog NC outputs 1 ... 8: Default • "LBP_NC.A_Disabl_anaOut..." (DB10.DBB168) Analog NC outputs 1 ... 8: Lock • "LBP_NC.A_Setval_anaOut..." (DB10 from DBW170) Analog NC output 1 ... 8: Setting value • "LBP_NC.E_SetVal_anaOut..." (DB10 from DBW210) Analog NC outputs 1 ... 8: Setpoint • MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Analog NC inputs/outputs" > "NC outputs"

A_InMask_anaOut1, A_InMask_anaOut2, A_InMask_anaOut3, A_InMask_anaOut4, A_InMask_anaOut5, A_InMask_anaOut6, A_InMask_anaOut7, A_InMask_anaOut8

(DB10 DBX167.0, 167.1, ..., 167.7)

Description	Setting screen form of the analog NC output 1 ... 8 If the bit is set, the setting value specified by the PLC user program is used for the corresponding output instead of the NC output value. The current NC output value is retained. If the bit is reset, the last NC output value becomes active again for the corresponding output. **Note** The variable "A_Setval_anaOut1", ... (DB10.DBW170.0, ...) (analog NC outputs 1 ... 8: setting value) is jointly used by: • "A_OvMask_Out1", ... (DB10.DBX5.0, ...) (overwrite) for edge change 0 → 1 • "A_InMask_Out1", ... (DB10.DBX7.0, ...) (target) for signal state TRUE Activating the two interfaces simultaneously should be avoided.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The target is not active.
Value TRUE	The target is active.
Corresponds to	• "LBP_NC.A_OvMask_anaOut..." (DB10.DBB166) Analog NC outputs 1 ... 8: Overwrite • "LBP_NC.A_Disabl_anaOut..." (DB10.DBB168) Analog NC outputs 1 ... 8: Lock • "LBP_NC.A_Setval_anaOut..." (DB10 from DBW170) Analog NC output 1 ... 8: Setting value • "LBP_NC.E_SetVal_anaOut..." (DB10 from DBW210) Analog NC outputs 1 ... 8: Setpoint • MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Analog NC inputs/outputs" > "NC outputs"

A_Disabl_anaOut1, A_Disabl_anaOut2, A_Disabl_anaOut3, A_Disabl_anaOut4, A_Disabl_anaOut5, A_Disabl_anaOut6, A_Disabl_anaOut7, A_Disabl_anaOut8

(DB10 DBX168.0, 168.1, ..., 168.7)

Description	Disabling of the analog NC output 1 ... 8 If a bit is set, then for the corresponding output, instead of the NC output value, the setting value specified by the PLC user program is used. The current NC output value is retained. If a bit is reset, then for the corresponding output, the last NC output value becomes active again.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The target is not active.
Value TRUE	The target is active.

Corresponds to	• "LBP_NC.A_OvMask_anaOut..." (DB10.DBB166) Analog NC outputs 1 ... 8: Overwrite • "LBP_NC.A_InMask_anaOut..." (DB10.DBB167) Analog NC outputs 1 ... 8: Default • "LBP_NC.A_Setval_anaOut..." (DB10 from DBW170) Analog NC output 1 ... 8: Setting value • "LBP_NC.E_SetVal_anaOut..." (DB10 from DBW210) Analog NC outputs 1 ... 8: Setpoint • MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Analog NC inputs/outputs" > "NC outputs"

A_Setval_anaOut1, A_Setval_anaOut2, A_Setval_anaOut3, A_Setval_anaOut4, A_Setval_anaOut5, A_Setval_anaOut6, A_Setval_anaOut7, A_Setval_anaOut8

(DB10 DBW170, 172, ..., 184)

Description	Setpoint for the analog output 1 ... 8 of the NC Using the setting value, the PLC user program can specify a defined output value. In order that the setting value becomes active, it must be activated via the interface for "Override" or "Target". **Note** The variable "A_Setval_anaOut1", ... (DB10.DBW170.0, ...) (analog NC outputs 1 ... 8: setting value) is jointly used by: • "A_OvMask_anaOut1", ... (DB10.DBX166.0, ...) (overwrite) for edge change 0 → 1 • "A_InMask_anaOut1", ... (DB10.DBX167.0, ...) (target) for signal state TRUE Activating the two interfaces simultaneously should be avoided.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_NC.A_OvMask_anaOut..." (DB10.DBB166) Analog NC outputs 1 ... 8: Overwrite • "LBP_NC.A_InMask_anaOut..." (DB10.DBB167) Analog NC outputs 1 ... 8: Default • "LBP_NC.A_Disabl_anaOut..." (DB10.DBB168) Analog NC outputs 1 ... 8: Lock • "LBP_NC.E_SetVal_anaOut..." (DB10 from DBW210) Analog NC outputs 1 ... 8: Setpoint • MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Analog NC inputs/outputs" > "NC outputs"

E_ActVal_anaIn1, E_ActVal_anaIn2, E_ActVal_anaIn3, E_ActVal_anaIn4, E_ActVal_anaIn5, E_ActVal_anaIn6, E_ActVal_anaIn7, E_ActVal_anaIn8

(DB10 DBW194, 196, ..., 208)

Description	Actual value for the analog input 1 ... 8 of the NC The actual value must be specified as a fixed-point number (16 bit value including sign) in a 2's complement.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_NC.A_Disabl_anaIn..." (DB10.DBB146) Analog NC inputs 1 ... 8: Lock • "LBP_NC.A_InMask_anaIn..." (DB10.DBB147) Analog NC inputs 1 ... 8: Default • "LBP_NC.A_Setval_anaIn..." (DB10 from DBW148) Analog NC inputs 1 ... 8: Setting value • MD10300 $MN_FASTIO_ANA_NUM_INPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O access via PLC" > "Analog NC inputs/outputs" > "NC inputs"

E_SetVal_anaOut1, E_SetVal_anaOut2, E_SetVal_anaOut3, E_SetVal_anaOut4, E_SetVal_anaOut5, E_SetVal_anaOut6, E_SetVal_anaOut7, E_SetVal_anaOut8

(DB10 DBW210, 212, ..., 224)

Description	Setpoint for the analog output 1 ... 8 of the NC The actual NC output value can be read in the PLC user program via the setpoint. The setpoint is represented as a fixed point number (16 bit value including sign) in a 2's complement. **Note** The value in the "Setpoint" interface can have a different value than that available at the NC output as a result of the various subsequent influencing possibilities.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_NC.A_OvMask_anaOut..." (DB10.DBB166) Analog NC outputs 1 ... 8: Overwrite • "LBP_NC.A_InMask_anaOut..." (DB10.DBB167) Analog NC outputs 1 ... 8: Default • "LBP_NC.A_Disabl_anaOut..." (DB10.DBB168) Analog NC outputs 1 ... 8: Lock • "LBP_NC.A_Setval_anaOut..." (DB10 from DBW170) Analog NC output 1 ... 8: Setting value • MD10310 $MN_FASTIO_ANA_NUM_OUTPUTS
Additional references	Function Manual Basic Functions; Section "Digital and analog NC I/O" > "Indirect I/O accesses via PLC" > "Analog NC inputs/outputs" > "NC outputs"

E_CollCheck (DB10 ab DBB226)

<table>
<tr><td>Description</td><td colspan="9">Collision avoidance: Protection area active
The assignment protection area ↔ interface signal (<Index>.<Bit>) is achieved by parameterizing the system variables $NP_BIT_NO:
$NP_BIT_NO[<protection area>] = <bit number></td></tr>
<tr><td></td><td rowspan="2">Index</td><td colspan="8">Bit</td></tr>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>1</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td></td><td>2</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td></td><td>3</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td></td><td>4</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td>32</td></tr>
<tr><td></td><td>5</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td></tr>
<tr><td></td><td>6</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td>48</td></tr>
<tr><td></td><td>7</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td></tr>
<tr><td>Signal flow</td><td colspan="9">NC → PLC</td></tr>
<tr><td>Data type</td><td colspan="9">ARRAY[0..7] of BYTE</td></tr>
<tr><td>Update</td><td colspan="9">Cyclic</td></tr>
<tr><td>Value FALSE</td><td colspan="9">The protection area connected to the interface signal is not active.
If several protection areas are connected to the interface signal: At least one protection area is not active.</td></tr>
<tr><td>Value TRUE</td><td colspan="9">The protection area connected to the interface signal is active.
If several protection areas are connected to the interface signal: All protection areas are active.</td></tr>
<tr><td>Corresponds to</td><td colspan="9">• "LBP_NC.A_CollCheck" (DB10 from DBB234 up)
Collision avoidance: Activate protection area
• System variable $NP_BIT_NO number of the interface bit for switchover activated</td></tr>
<tr><td>Additional references</td><td colspan="9">Function Manual Monitoring and Compensating; Section "Geometric machine modeling" > "Commissioning" > "System variables: Protection areas" > "$NP_BIT_NO"
Function Manual Monitoring and Compensating; Section "Collision avoidance"</td></tr>
</table>

A_CollCheck (DB10 ab DBB234)

Description	Collision avoidance: Activate protection area The assignment protection area ↔ interface signal (<Index>.<Bit>) is achieved by parameterizing the system variables $NP_BIT_NO: $NP_BIT_NO[<protection area>] = <bit number>							
	Index	**Bit**						
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0	7	6	5	4	3	2	1	0
	1	15	14	13	12	11	10	9	8
	2	23	22	21	20	19	18	17	16
	3	31	30	29	28	27	26	25	24
	4	39	38	37	36	35	34	33	32
	5	47	46	45	44	43	42	41	40
	6	55	54	53	52	51	50	49	48
	7	63	62	61	60	59	58	57	56
Signal flow	PLC → NC							
Data type	ARRAY[0..7] of BYTE							
Update	Cyclic							
Value FALSE	Request to deactivate the protection areas that are connected to the interface signal.							
Value TRUE	Request to activate the protection areas that are connected to the interface signal.							
Corresponds to	• "LBP_NC.E_CollCheck" (DB10 ab DBB226) Collision avoidance: Activate protection area • System variable $NP_BIT_NO number of the interface bit for switchover activated							
Additional references	Function Manual Monitoring and Compensating; Section "Geometric machine modeling" > "Commissioning" > "System variables: Protection areas" > "$NP_BIT_NO" Function Manual Monitoring and Compensating; Section "Collision avoidance"							

E_EHW1_stands, E_EHW2_stands, E_EHW3_stands, E_EHW4_stands, E_EHW5_stands, E_EHW6_stands

(DB10 DBX245.0, 245.1, ..., 245.5)

Description	Ethernet handwheel is stationary Handwheel 1: "E_EHW1_stands" (DB10.DBX245.0) Handwheel 2: "E_EHW2_stands" (DB10.DBX245.1) Handwheel 3: "E_EHW3_stands" (DB10.DBX245.2) Handwheel 4: "E_EHW4_stands" (DB10.DBX245.3) Handwheel 5: "E_EHW5_stands" (DB10.DBX245.4) Handwheel 6: "E_EHW6_stands" (DB10.DBX245.5) A stationary state is detected by the Ethernet modules to which the handwheel is connected. If a handwheel does not transfer any handwheel pulses for a defined period of time, the module detects this to be a stationary state and transfers it to the NC/PLC interface.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	Ethernet handwheel is operated
Value TRUE	Ethernet handwheel is stationary
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

DB10 from DBB246

Description	Handwheel active One signal is available for each handwheel:	
	Bit 0	Handwheel 1
	Bit 1	Handwheel 2
	Bit 2	Handwheel 3
	Bit 3	Handwheel 4
	Bit 4	Handwheel 5
	Bit 5	Handwheel 6
Signal flow	HMI → PLC	
Data type	BOOL	
Update	Cyclic	
Value FALSE	The button for the handwheel in the HMI is not depressed for the HT10.	
Value TRUE	The button for the handwheel in the HMI is depressed for the HT10.	
Additional references	Function Manual Axes and Spindles	

A_RobotStatus (DB10 ab DBB248)

Description	Signals from the robot
Signal flow	PLC → NC
Data type	ARRAY[0..7] of BYTE

E_RobotStatus (DB10 from DBB256)

Description	Signals to the robot
Signal flow	NC → PLC
Data type	ARRAY[0..7] of BYTE

10.4.7 LBP_ModeGroup [DB11] - Mode-group-specific signals

10.4.7.1 Overview

LBP_ModeGroup [DB11], mode signals 1 to the NC

The start address of a mode group interface can be calculated as follows:

Start address = 20 * (n - 1), with n = mode group number = 1, 2, 3, ...

Table 10-32 Mode signals mode group 1 to the NC

LBP_ModeGroup [DB11]	**Signals to mode group 1 (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Mode group reset	Mode group stop axes plus spindles	Mode group Stop	Mode change disable		Mode: Requirement		
						JOG	MDI	AUTO
DBB1	Single block					Machine function: Requirement		
	Type A	Type B				REF	REPOS	TEACH IN
DBB2	Machine function: Requirement **Note**: The interface must be enabled explicitly: "LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE							
		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB3								

LBP_ModeGroup [DB11], mode signals 1 from the NC

Table 10-33 Mode signals mode group 1 from the NC

LBP_ModeGroup [DB11]	**Signals from mode group 1 (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB4 Operating software → PLC						Mode strobe		
						JOG	MDI	AUTO
DBB5 Operating software → PLC						Machine function strobe		
						REF	REPOS	TEACH IN
DBB6	All channels in the reset state		NC internal JOG active	Mode group reset performed	Mode group ready	Active operating mode		
						JOG	MDI	AUTO
DBB7						Active machine function		
						REF	REPOS	TEACH IN
DBB8	Machine functions: Selection **Note**: The interface must be enabled explicitly: "LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE							
		Continuous travel active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB9								

LBP_ModeGroup [DB11], mode signals 1 from the operating software

Table 10-34 Mode signals mode group 1 from the operating software

LBP_ModeGroup [DB11]	**Signals to mode group 1 (operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB10	Selected from the HMI							
Operating software → PLC		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB11								

LBP_ModeGroup [DB11], mode signals 2 to the NC

Table 10-35 Mode signals mode group 2 to the NC

LBP_ModeGroup [DB11]	**Signals to mode group 2 (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB20	Mode group reset	Mode group Stop axes plus spindles	Mode group Stop	Mode change disable		Mode: Requirement		
						JOG	MDI	AUTO
DBB21	Single block					Machine function: Requirement		
	Type A	Type B				REF	REPOS	TEACH IN
DBB22	Machine function: Requirement **Note**: The interface must be enabled explicitly: "LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE							
		Continuous travel active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB23								

LBP_ModeGroup [DB11], mode signals 2 from the NC

Table 10-36 Mode signals mode group 2 from the NC

LBP_ModeGroup [DB11]	**Signals from mode group 2 (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB24						Mode strobe		
Operating software → PLC						JOG	MDI	AUTO

LBP_ModeGroup [DB11]	**Signals from mode group 2 (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB25						Machine function strobe		
Operating software → PLC						REF	REPOS	TEACH IN
DBB26	All channels in state: Reset		JOG is active internally in the NC	Mode group reset performed	Mode group ready	Active operating mode		
						JOG	MDI	AUTO
DBB27						Active machine function		
						REF	REPOS	TEACH IN
DBB28	Active machine function **Note**: The interface must be enabled explicitly: "LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE							
		Continuous travel active	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB29								
DBB30	Selected from the HMI							
Operating software → PLC		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1

10.4.7.2 LBP_ModeGroup [DB11]

The data block "LBP_ModeGroup [DB11]" contains mode signals to and from the NC.

The addresses of the mode group interfaces are assigned as follows:

Mode group	Address of the mode signals
Mode group 1	"LBP_ModeGroup.<symbol name>" (DB11.DBB0 ... DBB19)
Mode group 2	"LBP_ModeGroup.MG[2].<symbol name>" (DB11.DBB20 ... DBB39)
Mode group 3	"LBP_ModeGroup.MG[3].<symbol name>" (DB11.DBB40 ... DBB59)
...	...
Mode group 10	"LBP_ModeGroup.MG[10].<symbol name>" (DB11.DBB180 ... DBB199)

Variables in "LBP_ModeGroup [DB11]"

A_AUTO

(DB11 DBX0.0, 20.0, ..., 180.0)

Description	Request for AUTOMATIC mode to mode group Irrelevant for "LBP_ModeGroup.A_MCDisable" (DB11 DBX0.4, 20.4, ...,180.4) (operating mode changeover inhibit) == 1
Signal flow	PLC → NC
Data type	BOOL

Update	Cyclic
Value FALSE	There is no active request to switch over into AUTOMATIC mode.
Value TRUE	There is an active request to switch over into the AUTOMATIC mode.
Corresponds to	• "LBP_ModeGroup.A_MDA" (DB11 DBX0.1, 20.1, ...,180.1) MDI mode • "LBP_ModeGroup.A_JOG" (DB11 DBX0.2, 20.2, ...,180.2) JOG mode • "LBP_ModeGroup.A_MCDisable" (DB11 DBX0.4, 20.4, ...,180.4) Mode change disable • "LBP_ModeGroup.E_AUTO" (DB11 DBX6.6, 26.0, ...,186.0) AUTOMATIC mode active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_MDA

(DB11 DBX0.1, 20.1, ..., 180.1)

Description	Request for MDI mode to mode group Irrelevant for "LBP_ModeGroup.A_MCDisable" (DB11 DBX0.4, 20.4, ...,180.4) (operating mode changeover inhibit) == 1
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request to switch over into the MDI mode.
Value TRUE	There is an active request to switch over into the MDI mode.
Corresponds to	• "LBP_ModeGroup.A_AUTO" (DB11 DBX0.0, 20.0, ...,180.0) AUTOMATIC mode • "LBP_ModeGroup.A_JOG" (DB11 DBX0.2, 20.2, ...,180.2) JOG mode • "LBP_ModeGroup.A_MCDisable" (DB11 DBX0.4, 20.4, ...,180.4) Mode change disable • "LBP_ModeGroup.E_MDA" (DB11 DBX6.1, 26.1, ...,186.1) MDI mode active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_JOG

(DB11 DBX0.2, 20.2, ..., 180.2)

Description	Request for JOG mode to mode group Irrelevant for "LBP_ModeGroup.A_MCDisable" (DB11 DBX0.4, 20.4, ...,180.4) (operating mode changeover inhibit) == 1
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request to switch over into JOG mode.
Value TRUE	There is an active request to switch over into the JOG mode.

Corresponds to	• "LBP_ModeGroup.A_AUTO" (DB11 DBX0.0, 20.0, ...,180.0) AUTOMATIC mode • "LBP_ModeGroup.A_MDA" (DB11 DBX0.1, 20.1, ...,180.1) MDI mode • "LBP_ModeGroup.A_MCDisable" (DB11 DBX0.4, 20.4, ...,180.4) Mode change disable • "LBP_ModeGroup.E_JOG" (DB11 DBX6.2, 26.2, ...,186.2) JOG mode active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_MCDisable

(DB11 DBX0.4, 20.4, ..., 180.4)

Description	Mode change inhibit
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The mode of the mode group can be changed.
Value TRUE	The active mode of the mode group cannot be changed. **Note** The machine functions that can be selected within a mode group can be changed.
Corresponds to	• "LBP_ModeGroup.A_AUTO" (DB11 DBX0.0, 20.0, ...,180.0) AUTOMATIC mode • "LBP_ModeGroup.A_MDA" (DB11 DBX0.1, 20.1, ...,180.1) MDI mode • "LBP_ModeGroup.A_JOG" (DB11 DBX0.2, 20.2, ...,180.2) JOG mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_MGStop

(DB11 DBX0.5, 20.5, ..., 180.5)

Description	Mode group stop
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request for "Mode group stop".

Value TRUE	There is an active request for "Mode group stop". Responses: • An "NC Stop" is initiated for all the channels of the mode group. • Channel state change as a result of the mode group stop: – "Active" → "Interrupted" – "Reset" → "Reset" • Program state change as a result of the mode group stop – "Running" → "Stopped" • All traversing axes of the mode group are decelerated without contour violation to zero speed along their acceleration curves. Programs that have been stopped can be restarted with "NC Start". The spindles of the mode group are not influenced.
Corresponds to	• "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) NC stop at the block limit • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_MGStopASp

(DB11 DBX0.6, 20.6, ..., 180.6)

Description	Mode group stop axes plus spindles
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request for "Mode group stop axes plus spindles".
Value TRUE	There is an active request for "Mode group stop axes plus spindles". Responses: • An "NC Stop" is initiated for all the channels of the mode group. • Channel state change as a result of the mode group stop: – "Active" → "Interrupted" – "Reset" → "Reset" • Program state change as a result of the mode group stop – "Running" → "Stopped" • All traversing axes and spindles of the mode group are decelerated without contour violation to zero speed along their acceleration curves. Programs that have been stopped can be restarted with "NC Start". The spindles of the mode group are not influenced.
Corresponds to	• "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) NC stop at the block limit • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_MGReset

(DB11 DBX0.7, 20.7, ..., 180.7)

Description	Mode group reset **Special case** If an alarm resets the interface signal "LBP_ModeGroup.E_MGOK" (DB11 DBX6.3, 26.3, ...,186.3) (mode group ready), all channels of the mode group are no longer in the "reset" channel state. In order that the mode of the mode group can be switched over, initially, a "Mode group reset" must be initiated using "LBP_ModeGroup.A_MGRESET" (DB11 DBX0.7, 20.7, ...,180.7). **Note** Before a new action can be requested in a channel of the mode group from the PLC user program after a mode group reset (e.g. NC start or axis interchange), the system must first wait for either the mode group state "All channels of the mode group in the reset state" ("LBP_ModeGroup.E_ChanReset" (DB11 DBX6.7, 26.7, ...,186.7) == 1) or the channel state "Reset" ("LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) == 1) of the channel involved.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	A request was initiated for a "Mode group reset". Responses: • In all channels of the mode group, a "channel reset" is initiated in the control system corresponding to "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) = 1. • All the channels are then in the channel state "reset" "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) == 1 • All active NC programs are then in the "canceled" program state "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) == 1 • All traversing axes and spindles are decelerated without contour violation to zero speed according to their acceleration curves. • The channel-specific reset positions become active. • All alarms of the mode group are cleared - with the exception of power on alarms.
Edge change 1 → 0	Reset of the last request for a "Mode group reset". Starting at this time, a new "Mode group reset" can be requested.
Corresponds to	• "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) Channel reset • "LBP_ModeGroup.E_ChanReset" (DB11 DBX6.7, 26.7, ...,186.7) All channels in the reset state
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_TEACHIN

(DB11 DBX1.0, 21.0, ..., 181.0)

Description	TEACH IN machine function **Note** Only active in JOG mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic

Value FALSE	There is no active request to activate the TEACH IN machine function.
Value TRUE	There is an active request to activate the TEACH IN machine function.
Corresponds to	• "LBP_ModeGroup.E_JOG" (DB11 DBX6.2, 26.2, ...,186.2) JOG mode active • "LBP_ModeGroup.E_TEACHIN" (DB11 DBX7.0, 27.0, ...,187.0) Machine function TEACH IN active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_REPOS

(DB11 DBX1.1, 21.1, ..., 181.1)

Description	REPOS machine function **Note** Only active in JOG mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request to activate the REPOS machine function.
Value TRUE	There is an active request to activate the REPOS machine function.
Corresponds to	• "LBP_ModeGroup.E_JOG" (DB11 DBX6.2, 26.2, ...,186.2) JOG mode active • "LBP_ModeGroup.E_REPOS" (DB11 DBX7.1, 27.1, ...,187.1) Machine function REPOS active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_REF

(DB11 DBX1.2, 21.2, ..., 181.2)

Description	REF machine function **Note** Only active in JOG mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request to activate the REF machine function.
Value TRUE	There is an active request to activate the REF machine function.
Corresponds to	• "LBP_ModeGroup.E_JOG" (DB11 DBX6.2, 26.2, ...,186.2) JOG mode active • "LBP_ModeGroup.E_REF" (DB11 DBX7.2, 27.2, ...,187.2) Machine function REF active
Additional references	Function Manual Axes and Spindles; Section "Referencing"

A_SingleBlock_B

(DB11 DBX1.6, 21.6, ..., 181.6)

Description	Single block, type B Cross mode group response when activating "Single block, type B": • All channels are stopped • All channels receive a start command • Channel KS stops at the end of the block • The KA channels contain the instruction STOPATEND, comparable with "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) (NC stop at block limit) All channels are stopped at a block limit (at some point in time). **Note** • "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 1 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 0 ⇒ single block type B • "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 0 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 1 ⇒ single block type A • "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 1 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 1 **OR** "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 0 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 0 ⇒ No mode group specific single block selected
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request to activate the "Single block, type B".
Value TRUE	There is an active request to activate the "Single block, type B".
Corresponds to	• "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) Single block type A
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_SingleBlock_A

(DB11 DBX1.7, 21.7, ..., 181.7)

Description	Single block, type B Cross mode group response when activating "Single block, type A": • All channels are stopped • All channels receive a start command • Channel KS stops at the end of the block • The KA channels contain the instruction STOPATEND, comparable with "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) (NC stop at block limit) All channels are stopped at a block limit (at some point in time). **Note** • "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 1 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 0 ⇒ single block type B • "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 0 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 1 ⇒ single block type A • "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 1 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 1 **OR** "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) == 0 AND "LBP_ModeGroup.A_SingleBlock_A" (DB11 DBX1.7, 21.7, ...,181.7) == 0 ⇒ No mode group specific single block selected
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active request to activate the "Single block, type A".
Value TRUE	There is an active request to activate the "Single block, type A".
Corresponds to	• "LBP_ModeGroup.A_SingleBlock_B" (DB11 DBX1.6, 21.6, ...,181.6) Single block, type B
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_INC1

(DB11 DBX2.0, 22.0, ..., 182.0)

Description	Machine function: Request **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC1 is not requested.
Value TRUE	Machine function INC1 is active.

Corresponds to	• "LBP_ModeGroup.A_INC10" (DB11 DBX2.1, 22.1, ..., 182.1) • "LBP_ModeGroup.A_INC100" (DB11 DBX2.2, 22.2, ..., 182.2) • "LBP_ModeGroup.A_INC1000" (DB11 DBX2.3, 22.3, ..., 182.3) • "LBP_ModeGroup.A_INC10000" (DB11 DBX2.4, 22.4, ..., 182.4) • "LBP_ModeGroup.A_INCVar" (DB11 DBX2.5, 22.5, ..., 182.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_INC10

(DB11 DBX2.1, 22.1, ..., 182.1)

Description	Machine function: Request **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC10 is not requested.
Value TRUE	Machine function INC10 is active.
Corresponds to	• "LBP_ModeGroup.A_INC1" (DB11 DBX2.0, 22.0, ..., 182.0) • "LBP_ModeGroup.A_INC100" (DB11 DBX2.2, 22.2, ..., 182.2) • "LBP_ModeGroup.A_INC1000" (DB11 DBX2.3, 22.3, ..., 182.3) • "LBP_ModeGroup.A_INC10000" (DB11 DBX2.4, 22.4, ..., 182.4) • "LBP_ModeGroup.A_INCVar" (DB11 DBX2.5, 22.5, ..., 182.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_INC100

(DB11 DBX2.2, 22.2, ..., 182.2)

Description	Machine function: Request **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC100 is not requested.
Value TRUE	Machine function INC100 is active.

Corresponds to	• "LBP_ModeGroup.A_INC1" (DB11 DBX2.0, 22.0, ..., 182.0) • "LBP_ModeGroup.A_INC10" (DB11 DBX2.1, 22.1, ..., 182.1) • "LBP_ModeGroup.A_INC1000" (DB11 DBX2.3, 22.3, ..., 182.3) • "LBP_ModeGroup.A_INC10000" (DB11 DBX2.4, 22.4, ..., 182.4) • "LBP_ModeGroup.A_INCVar" (DB11 DBX2.5, 22.5, ..., 182.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_INC1000

(DB11 DBX2.3, 22.3, ..., 182.3)

Description	Machine function: Request **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC1000 is not requested.
Value TRUE	Machine function INC1000 is active.
Corresponds to	• "LBP_ModeGroup.A_INC1" (DB11 DBX2.0, 22.0, ..., 182.0) • "LBP_ModeGroup.A_INC10" (DB11 DBX2.1, 22.1, ..., 182.1) • "LBP_ModeGroup.A_INC100" (DB11 DBX2.2, 22.2, ..., 182.2) • "LBP_ModeGroup.A_INC10000" (DB11 DBX2.4, 22.4, ..., 182.4) • "LBP_ModeGroup.A_INCVar" (DB11 DBX2.5, 22.5, ..., 182.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_INC10000

(DB11 DBX2.4, 22.4, ..., 182.4)

Description	Machine function: Request **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC10000 is not requested.
Value TRUE	Machine function INC10000 is active.

Corresponds to	• "LBP_ModeGroup.A_INC1" (DB11 DBX2.0, 22.0, ..., 182.0) • "LBP_ModeGroup.A_INC10" (DB11 DBX2.1, 22.1, ..., 182.1) • "LBP_ModeGroup.A_INC100" (DB11 DBX2.2, 22.2, ..., 182.2) • "LBP_ModeGroup.A_INC1000" (DB11 DBX2.3, 22.3, ..., 182.3) • "LBP_ModeGroup.A_INCVar" (DB11 DBX2.5, 22.5, ..., 182.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_INCVar

(DB11 DBX2.5, 22.5, ..., 182.5)

Description	Machine function: Request **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INCVar is not requested.
Value TRUE	Machine function INCVar is active.
Corresponds to	• "LBP_ModeGroup.A_INC1" (DB11 DBX2.0, 22.0, ..., 182.0) • "LBP_ModeGroup.A_INC10" (DB11 DBX2.1, 22.1, ..., 182.1) • "LBP_ModeGroup.A_INC100" (DB11 DBX2.2, 22.2, ..., 182.2) • "LBP_ModeGroup.A_INC1000" (DB11 DBX2.3, 22.3, ..., 182.3) • "LBP_ModeGroup.A_INC10000" (DB11 DBX2.4, 22.4, ..., 182.4)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MMC_AUTO

(DB11 DBX4.0, 24.0, ..., 184.0)

Description	AUTOMATIC mode selected
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the AUTOMATIC mode.
Value TRUE	There is an active HMI request to select the AUTOMATIC mode.
Corresponds to	• "LBP_ModeGroup.A_AUTO" (DB11.DBX0.0) AUTOMATIC mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MMC_MDA

(DB11 DBX4.1, 24.1, ..., 184.1)

Description	MDI mode selected
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the MDI mode.
Value TRUE	There is an active HMI request to select the MDI mode.
Corresponds to	• "LBP_ModeGroup.A_MDA" (DB11.DBX0.1) MDI mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MMC_JOG

(DB11 DBX4.2, 24.2, ..., 184.2)

Description	JOG mode selected
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the JOG mode.
Value TRUE	There is an active HMI request to select the JOG mode.
Corresponds to	• "LBP_ModeGroup.A_JOG" (DB11.DBX0.2) JOG mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MMC_TEACHIN

(DB11 DBX5.0, 25.0, ..., 185.0)

Description	TEACH IN machine function selected
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the TEACH IN machine function.
Value TRUE	There is an active HMI request to select the TEACH IN machine function.
Corresponds to	• "LBP_ModeGroup.E_TEACHIN" (DB11.DBX7.0) Active machine function TEACH IN
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MMC_REPOS

(DB11 DBX5.1, 25.1, ..., 185.1)

Description	REPOS machine function selected
Signal flow	HMI → PLC
Data type	BOOL

Update	Cyclic
Value FALSE	There is no active HMI request to select the REPOS machine function.
Value TRUE	There is an active HMI request to select the REPOS machine function.
Corresponds to	• "LBP_ModeGroup.E_REPOS" (DB11.DBX7.1) Active REPOS machine function
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MMC_REF

(DB11 DBX5.2, 25.2, ..., 185.2)

Description	REF machine function selected
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the REF machine function.
Value TRUE	There is an active HMI request to select the REF machine function.
Corresponds to	• "LBP_ModeGroup.E_REF" (DB11.DBX7.2) Active machine function REF
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_AUTO

(DB11 DBX6.0, 26.0, ..., 186.0)

Description	AUTOMATIC mode active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The AUTOMATIC mode is not active.
Value TRUE	The AUTOMATIC mode is active.
Corresponds to	• "LBP_ModeGroup.A_AUTO" (DB11.DBX0.0) AUTOMATIC mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MDA

(DB11 DBX6.1, 26.1, ..., 186.1)

Description	Active MDI mode
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The MDI mode is not active.
Value TRUE	The MDI mode is active.

Corresponds to	• "LBP_ModeGroup.A_MDA" (DB11.DBX0.1) MDI mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_JOG

(DB11 DBX6.2, 26.2, ..., 186.2)

Description	Active JOG mode
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The JOG mode is not active.
Value TRUE	The JOG mode is active.
Corresponds to	• "LBP_ModeGroup.A_JOG" (DB11.DBX0.2) JOG mode
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MGOK

(DB11 DBX6.3, 26.3, ..., 186.3)

Description	Mode group ready Response when the signal changes from 0 → 1: • Traversing axes and spindles are decelerated to standstill with setpoint 0 or maximum braking current. • The interface signals from the PLC to the NC are brought into an inactive state (reset position). **Note** If an alarm occurs that clears "Mode group ready", before switching over the mode, a mode group reset "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) must first be initiated.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The mode group is not ready.
Value TRUE	The mode group is ready.
Corresponds to	• "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) Mode group reset
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_MGreseted

(DB11 DBX6.4, 26.4, ..., 186.4)

Description	Mode group reset performed In "JOG in AUTOMATIC", the NC/PLC interface displays whether the mode group is in "Mode group RESET".
Signal flow	NC → PLC

Data type	BOOL
Update	Cyclic
Value FALSE	The mode group has not been reset.
Value TRUE	The mode group has been reset and is in the reset state.
Corresponds to	• "LBP_ModeGroup.A_MGReset" (DB11.DBX0.7) Mode group reset
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_NCKintJOG

(DB11 DBX6.5, 26.5, ..., 186.5)

Description	NC internal JOG active In "JOG in AUTOMATIC", the NC/PLC interface displays whether the NC has automatically switched to "Internal JOG".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	NC internal JOG mode is not active
Value TRUE	NC internal JOG is active
Corresponds to	• MD10735 $MN_JOG_MODE_MASK
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ChanReset

(DB11 DBX6.7, 26.7, ..., 186.7)

Description	All channels in the "Reset" state
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	At least one channel of the mode group is not in the "Reset" state.
Value TRUE	All mode group channels are in the "Reset" state.
Corresponds to	• "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) Channel state, "Reset"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_TEACHIN

(DB11 DBX7.0, 27.0, ..., 187.0)

Description	TEACH IN machine function active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function TEACH IN is not active.
Value TRUE	The TEACH IN machine function is active.

Corresponds to	• "LBP_ModeGroup.A_TEACHIN" (DB11.DBX1.0) TEACH IN machine function • "LBP_ModeGroup.E_MMC_TEACHIN" (DB11.DBX5.0) TEACH IN machine function selected
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_REPOS

(DB11 DBX7.1, 27.1, ..., 187.1)

Description	REPOS machine function active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function REPOS is not active.
Value TRUE	Machine function REPOS is active.
Corresponds to	• "LBP_ModeGroup.A_REPOS" (DB11.DBX1.1) REPOS machine function • "LBP_ModeGroup.E_MMC_REPOS" (DB11.DBX5.1) REPOS machine function selected
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_REF

(DB11 DBX7.2, 27.2, ..., 187.2)

Description	REF machine function active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function REF is not active.
Value TRUE	Machine function REF is active.
Corresponds to	• "LBP_ModeGroup.A_REF" (DB11.DBX1.2) REF machine function • "LBP_ModeGroup.E_MMC_REF" (DB11.DBX5.2) REF machine function selected
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_INC1

(DB11 DBX8.0, 28.0, ..., 188.0)

Description	Machine function: Selection **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	NC → PLC

Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC1 is not selected.
Value TRUE	Machine function INC1 is selected.
Corresponds to	• "LBP_ModeGroup.E_INC10" (DB11 DBX8.1, 28.1, ..., 188.1) • "LBP_ModeGroup.E_INC100" (DB11 DBX8.2, 28.2, ..., 188.2) • "LBP_ModeGroup.E_INC1000" (DB11 DBX8.3, 28.3, ..., 188.3) • "LBP_ModeGroup.E_INC10000" (DB11 DBX8.4, 28.4, ..., 188.4) • "LBP_ModeGroup.E_INCVar" (DB11 DBX8.5, 28.5, ..., 188.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_INC10

(DB11 DBX8.1, 28.1, ..., 188.1)

Description	Machine function: Selection **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC10 is not selected.
Value TRUE	Machine function INC10 is selected.
Corresponds to	• "LBP_ModeGroup.E_INC1" (DB11 DBX8.0, 28.0, ..., 188.0) • "LBP_ModeGroup.E_INC100" (DB11 DBX8.2, 28.2, ..., 188.2) • "LBP_ModeGroup.E_INC1000" (DB11 DBX8.3, 28.3, ..., 188.3) • "LBP_ModeGroup.E_INC10000" (DB11 DBX8.4, 28.4, ..., 188.4) • "LBP_ModeGroup.E_INCVar" (DB11 DBX8.5, 28.5, ..., 188.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_INC100

(DB11 DBX8.2, 28.2, ..., 188.2)

Description	Machine function: Selection **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC100 is not selected.
Value TRUE	Machine function INC100 is selected.

Corresponds to	• "LBP_ModeGroup.E_INC1" (DB11 DBX8.0, 28.0, ..., 188.0) • "LBP_ModeGroup.E_INC10" (DB11 DBX8.1, 28.1, ..., 188.1) • "LBP_ModeGroup.E_INC1000" (DB11 DBX8.3, 28.3, ..., 188.3) • "LBP_ModeGroup.E_INC10000" (DB11 DBX8.4, 28.4, ..., 188.4) • "LBP_ModeGroup.E_INCVar" (DB11 DBX8.5, 28.5, ..., 188.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_INC1000

(DB11 DBX8.3, 28.3, ..., 188.3)

Description	Machine function: Selection **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC1000 is not selected.
Value TRUE	Machine function INC1000 is selected.
Corresponds to	• "LBP_ModeGroup.E_INC1" (DB11 DBX8.0, 28.0, ..., 188.0) • "LBP_ModeGroup.E_INC10" (DB11 DBX8.1, 28.1, ..., 188.1) • "LBP_ModeGroup.E_INC100" (DB11 DBX8.2, 28.2, ..., 188.2) • "LBP_ModeGroup.E_INC10000" (DB11 DBX8.4, 28.4, ..., 188.4) • "LBP_ModeGroup.E_INCVar" (DB11 DBX8.5, 28.5, ..., 188.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_INC10000

(DB11 DBX8.4, 28.4, ..., 188.4)

Description	Machine function: Selection **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INC10000 is not selected.
Value TRUE	Machine function INC10000 is selected.

Corresponds to	• "LBP_ModeGroup.E_INC1" (DB11 DBX8.0, 28.0, ..., 188.0) • "LBP_ModeGroup.E_INC10" (DB11 DBX8.1, 28.1, ..., 188.1) • "LBP_ModeGroup.E_INC100" (DB11 DBX8.2, 28.2, ..., 188.2) • "LBP_ModeGroup.E_INC1000" (DB11 DBX8.3, 28.3, ..., 188.3) • "LBP_ModeGroup.E_INCVar" (DB11 DBX8.5, 28.5, ..., 188.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_INCVar

(DB11 DBX8.5, 28.5, ..., 188.5)

Description	Machine function: Selection **Note** The interface must be enabled explicitly. ("LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) := TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Machine function INCVar is not selected.
Value TRUE	Machine function INCVar is selected.
Corresponds to	• "LBP_ModeGroup.E_INC1" (DB11 DBX8.0, 28.0, ..., 188.0) • "LBP_ModeGroup.E_INC10" (DB11 DBX8.1, 28.1, ..., 188.1) • "LBP_ModeGroup.E_INC100" (DB11 DBX8.2, 28.2, ..., 188.2) • "LBP_ModeGroup.E_INC1000" (DB11 DBX8.3, 28.3, ..., 188.3) • "LBP_ModeGroup.E_INC10000" (DB11 DBX8.4, 28.4, ..., 188.4)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_INC1

(DB11 DBX10.0, 30.0, ..., 190.0)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_INC1.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the INC1 machine function.
Value TRUE	There is an active HMI request to select the INC1 machine function.

Corresponds to	• "LBP_ModeGroup.E_HMI_INC10" (DB11 DBX10.1, 30.1, ..., 190.1) • "LBP_ModeGroup.E_HMI_INC100" (DB11 DBX10.2, 30.2, ..., 190.2) • "LBP_ModeGroup.E_HMI_INC1000" (DB11 DBX10.3, 30.3, ..., 190.3) • "LBP_ModeGroup.E_HMI_INC10000" (DB11 DBX10.4, 30.4, ..., 190.4) • "LBP_ModeGroup.E_HMI_INCVar" (DB11 DBX10.5, 30.5, ..., 190.5) • "LBP_ModeGroup.E_HMI_ContManTravel" (DB11 DBX10.6, 30.6, ..., 190.6) • "LBP_ModeGroup.A_INC1" (DB11 DBX2.0, 22.0, ..., 182.0)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_INC10

(DB11 DBX10.1, 30.1, ..., 190.1)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_INC10.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the INC10 machine function.
Value TRUE	There is an active HMI request to select the INC10 machine function.
Corresponds to	• "LBP_ModeGroup.E_HMI_INC1" (DB11 DBX10.0, 30.0, ..., 190.0) • "LBP_ModeGroup.E_HMI_INC100" (DB11 DBX10.2, 30.2, ..., 190.2) • "LBP_ModeGroup.E_HMI_INC1000" (DB11 DBX10.3, 30.3, ..., 190.3) • "LBP_ModeGroup.E_HMI_INC10000" (DB11 DBX10.4, 30.4, ..., 190.4) • "LBP_ModeGroup.E_HMI_INCVar" (DB11 DBX10.5, 30.5, ..., 190.5) • "LBP_ModeGroup.E_HMI_ContManTravel" (DB11 DBX10.6, 30.6, ..., 190.6) • "LBP_ModeGroup.A_INC10" (DB11 DBX2.1, 22.1, ..., 182.1)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_INC100

(DB11 DBX10.2, 30.2, ..., 190.2)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_INC100.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the INC100 machine function.
Value TRUE	There is an active HMI request to select the INC100 machine function.

Corresponds to	• "LBP_ModeGroup.E_HMI_INC1" (DB11 DBX10.0, 30.0, ..., 190.0) • "LBP_ModeGroup.E_HMI_INC10" (DB11 DBX10.1, 30.1, ..., 190.1) • "LBP_ModeGroup.E_HMI_INC1000" (DB11 DBX10.3, 30.3, ..., 190.3) • "LBP_ModeGroup.E_HMI_INC10000" (DB11 DBX10.4, 30.4, ..., 190.4) • "LBP_ModeGroup.E_HMI_INCVar" (DB11 DBX10.5, 30.5, ..., 190.5) • "LBP_ModeGroup.E_HMI_ContManTravel" (DB11 DBX10.6, 30.6, ..., 190.6) • "LBP_ModeGroup.A_INC100" (DB11 DBX2.2, 22.2, ..., 182.2)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_INC1000

(DB11 DBX10.3, 30.3, ..., 190.3)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_INC1000.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the INC1000 machine function.
Value TRUE	There is an active HMI request to select the INC1000 machine function.
Corresponds to	• "LBP_ModeGroup.E_HMI_INC1" (DB11 DBX10.0, 30.0, ..., 190.0) • "LBP_ModeGroup.E_HMI_INC10" (DB11 DBX10.1, 30.1, ..., 190.1) • "LBP_ModeGroup.E_HMI_INC100" (DB11 DBX10.2, 30.2, ..., 190.2) • "LBP_ModeGroup.E_HMI_INC10000" (DB11 DBX10.4, 30.4, ..., 190.4) • "LBP_ModeGroup.E_HMI_INCVar" (DB11 DBX10.5, 30.5, ..., 190.5) • "LBP_ModeGroup.E_HMI_ContManTravel" (DB11 DBX10.6, 30.6, ..., 190.6) • "LBP_ModeGroup.A_INC1000" (DB11 DBX2.3, 22.3, ..., 182.3)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_INC10000

(DB11 DBX10.4, 30.4, ..., 190.4)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_INC10000.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the INC10000 machine function.
Value TRUE	There is an active HMI request to select the INC10000 machine function.

Corresponds to	• "LBP_ModeGroup.E_HMI_INC1" (DB11 DBX10.0, 30.0, ..., 190.0) • "LBP_ModeGroup.E_HMI_INC10" (DB11 DBX10.1, 30.1, ..., 190.1) • "LBP_ModeGroup.E_HMI_INC100" (DB11 DBX10.2, 30.2, ..., 190.2) • "LBP_ModeGroup.E_HMI_INC1000" (DB11 DBX10.3, 30.3, ..., 190.3) • "LBP_ModeGroup.E_HMI_INCVar" (DB11 DBX10.5, 30.5, ..., 190.5) • "LBP_ModeGroup.E_HMI_ContManTravel" (DB11 DBX10.6, 30.6, ..., 190.6) • "LBP_ModeGroup.A_INC10000" (DB11 DBX2.4, 22.4, ..., 182.4)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_INCVar

(DB11 DBX10.5, 30.5, ..., 190.5)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_INCVar.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the INCVar machine function.
Value TRUE	There is an active HMI request to select the INCVar machine function.
Corresponds to	• "LBP_ModeGroup.E_HMI_INC1" (DB11 DBX10.0, 30.0, ..., 190.0) • "LBP_ModeGroup.E_HMI_INC10" (DB11 DBX10.1, 30.1, ..., 190.1) • "LBP_ModeGroup.E_HMI_INC100" (DB11 DBX10.2, 30.2, ..., 190.2) • "LBP_ModeGroup.E_HMI_INC1000" (DB11 DBX10.3, 30.3, ..., 190.3) • "LBP_ModeGroup.E_HMI_INC10000" (DB11 DBX10.4, 30.4, ..., 190.4) • "LBP_ModeGroup.E_HMI_ContManTravel" (DB11 DBX10.6, 30.6, ..., 190.6) • "LBP_ModeGroup.A_INCVar" (DB11 DBX2.5, 22.5, ..., 182.5)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_HMI_ContManTravel

(DB11 DBX10.6, 30.6, ..., 190.6)

Description	Machine function: Request from the HMI **Note** This signal is only active if "LBP_ConfigData.MMCToIf" (DB7 DBX116.1) is = TRUE. The signal is then logically combined with A_ContManTravel.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no active HMI request to select the ContManTravel machine function.
Value TRUE	There is an active HMI request to select the ContManTravel machine function.

Corresponds to	• "LBP_ModeGroup.E_HMI_INC1" (DB11 DBX10.0, 30.0, ..., 190.0) • "LBP_ModeGroup.E_HMI_INC10" (DB11 DBX10.1, 30.1, ..., 190.1) • "LBP_ModeGroup.E_HMI_INC100" (DB11 DBX10.2, 30.2, ..., 190.2) • "LBP_ModeGroup.E_HMI_INC1000" (DB11 DBX10.3, 30.3, ..., 190.3) • "LBP_ModeGroup.E_HMI_INC10000" (DB11 DBX10.4, 30.4, ..., 190.4) • "LBP_ModeGroup.E_HMI_INCVar" (DB11 DBX10.5, 30.5, ..., 190.5) • "LBP_ModeGroup.A_ContManTravel" (DB11 DBX2.6, 22.6, ..., 182.6)
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

10.4.8 LBP_HMI [DB19] - Signals from/to the operator panel (OP)

10.4.8.1 Overview

LBP_HMI [DB19], signals to the operator panel (OP)

Table 10-37 Signals to the operator panel (OP)

LBP_HMI [DB19]	**Signals from the operator panel (OP) (PLC → operating software)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	**Interface of 1st Control (DBB0 - DBB49)**							
	Actual value in the WCS (TRUE) / MCS (FALSE)	Save to action log	HMI Advanced: Shutdown	Delete recall alarms	Delete cancel alarms	Key lock	Darken screen	Brighten screen
DBB1							Rights for the external viewer	External viewer
DBB2								
DBB4								
DBB6	Analog spindle 1: Utilization as a percentage							
DBB7	Analog spindle 2: Utilization as a percentage							
DBB8	Channel number of the machine control panel to the control							
DBB9	Reserved selection					Autotool measure	OEM2	OEM1
DBB10	Hardkeys							
DBB11	Reserved							
DBB12								
DBB13	Part program			Reserved				Disable teach-in transfer
	Selection	Load	Unload					

LBP_HMI [DB19]	**Signals from the operator panel (OP) (PLC → operating software)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB14	Active (FALSE) / passive (TRUE) file system	V24 active file system: Index of the file to be transferred from the standard list. V24 passive file system: Number of the control file for user file names.						
DBB15	V24 active file system: Index which specifies the axis, channel or TO No. V24 passive file system: Index of the file to be transferred from the user list.							
DBB16	Always TRUE	Program selection from the PLC: Index of the program list						
DBB17	Program selection from the PLC: Program index in the program list							
DBB18								
DBB19	Reserved (message counter)							

LBP_HMI [DB19], signals from the operator panel (OP)

Table 10-38 Signals from the operator panel (OP)

LBP_HMI [DB19]	**Signals from the operator panel (OP) (Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB20	Switch over MCS/WCS	Simulation active	Language 2 switched over	Recall alarm deleted	Cancel alarm deleted	Cancel key actuated	Screen is dark	
DBB21	Current number of the active operating area							
DBB22	Current channel number							
DBB23						Control 1		
						Screen change active	Data transfer active	Keyboard has been used
DBW24								
DBB25								
DBB26	**Program selection from the PLC**: Status signals							
	Selection	Load	Unload		Active	Error	Job completed	Reserved
DBB27	**Program selection from the PLC**: Error code							
DBW28	Screen number for "Supplement user interface"							
DBB30	**Screen selection from the PLC:** Control signals							
PLC → OP							Screen deselection	Screen selection
DBB31	**Screen selection from the PLC:** Status signals							
Operating software → PLC	Inactive			Error, screen selection not possible	Screen is deselected	Screen active	Screen is selected	Screen selection accepted

LBP_HMI [DB19]	**Signals from the operator panel (OP) (Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB32 PLC → operating software	Busy function	Strobe function	Function selection No. from the PLC					
DBB33 PLC → operating software	Parameter 1 for function selection No. (function selection from DBB32)							
DBB34 PLC → operating software	Parameter 2 for function selection No. (function selection from DBB32)							
DBB35 PLC → operating software	Parameter 3 for function selection No. (function selection from DBB32)							
DBB36 Operating software → PLC	Error code for function selection No. (function selection from DBB32)							
DBB37 Operating software → PLC	Parameter 1 for function selection No. (function selection from DBB48)							
DBB38 Operating software → PLC	Parameter 2 for function selection No. (function selection from DBB48)							
DBB39 Operating software → PLC	Parameter 3 for function selection No. (function selection from DBB48)							
DBB40 - DBB47	Reserved							
DBB48 Operating software → PLC	PLC busy Function	Operating software strobe Function	Function selection No. from the operating software					
DBB49 PLC → operating software	Error code for function selection No. (function selection from DBB48)							
DBB50 - DBB99	**Interface of 2nd Control** (assignment the same as DBB0 - DBB49)							

LBP_HMI [DB19]	Signals from the operator panel (OP) (Operating software → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB100	**Switchover interface to the operating software** **Call waiting interface (operating software announces itself to the NC)** ONL_REQUEST (online request of operating softwareI) Operating software writes its client identification as online request (bits 8-15: bus type, bits 0-7: bus address)							
DBB102	ONL_CONFIRM (acknowledgment from PLC after online request) PLC writes as acknowledgment the operating software client identification (bus type, bus address; such as DBB100)							
DBB104	PAR_CLIENT_IDENT Operating software writes its client identification (bus type, bus address; such as DBB100).							
DBB106	PAR_MMC_TYP Type of operating software acc. to NETNAMES.INI: Main/secondary operator panel / server /...							
DBB107	PAR_MCP_ADR Operating software writes address of the MCP to be activated; 255, if no MCP activation							
DBB108	PAR_STATUS PLC writes the online enable for the operating software							
DBB109	PAR_Z_INFO PLC writes additional information on the status							
DBB110	M_TO_N_ALIVE Sign-of-life from the PLC to the operating software using the M to N block							
DBB112	Res. Bus type MCP							
DBB113	ParOpKeyAdr Direct key index call waiting interface							
DBB114	ParTcuIndex TCU index call waiting interface							
DBB115	ParHt2Index Ht2 index logon interface							
DBB116	Direct key address of 1st Online interface							
DBB117	Direct key address of 2nd Online interface							
DBB118	TCU index of 1st Online interface							
DBB119	TCU index of 2nd Online interface							
DBB120	**Online interface OP 1 (user)** MMC1_CLIENT_IDENT PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT, if operating software goes online.							
DBB122	MMC1_TYP PLC writes PAR_MMC_TYP to MMCx_TYP, if operating software goes online.							
DBB123	MMC1_MCP_ADR PLC writes PAR_MCP_ADR to MMCx_MCP_ADR, if operating software goes online.							
DBB124	MMC1_STATUS Connection state, operating software and PLC write alternating their requests/acknowledgments.							
DBB125	MMC1_Z_INFO Additional information, connection state (pos./neg. acknowledgment, error messages, etc.)							

LBP_HMI [DB19]	**Signals from the operator panel (OP) (Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB126	Reserved	TCU1_ SHIFT_ LOCK	MMC1_ CHANGE_ DENIED	MMC1_ ACTIVE_ CHANGED	MMC1_ ACTIVE_ PERM	MMC1_ ACTIVE_ REQ	MMC1_ MCP_ SHIFT-LOCK	MMC1_ SHIFT_ LOCK
DBB127	Reserved bus type MCP							
DBB128 - DBB129	Reserved Transline (Transline DB number) (SINUMERIK ONE)							
DBB130	**Online interface OP 2 (user)** MMC2_CLIENT_IDENT PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT, if operating software goes online.							
DBB132	MMC2_TYP PLC writes PAR_MMC_TYP to MMCx_TYP, if operating software goes online.							
DBB133	MMC2_MCP_ADR PLC writes PAR_MCP_ADR to MMCx_MCP_ADR, if operating software goes online.							
DBB134	MMC2_STATUS Connection state, operating software and PLC write alternating their requests/acknowledgments.							
DBB135	MMC2_Z_INFO Additional information, connection state (pos./neg. acknowledgment, error messages, etc.)							
DBB136	Reserved	TCU2_ SHIFT_ LOCK	MMC2_ CHANGE_ DENIED	MMC2_ ACTIVE_ CHANGED	MMC2_ ACTIVE_ PERM	MMC2_ ACTIVE_ REQ	MMC2_ MCP_ SHIFT_ LOCK	MMC2_ SHIFT_ LOCK
DBB137	Reserved bus type MCP							
DBB138 - DBB139	Reserved Transline (Transline DB number) (SINUMERIK ONE)							
DBB140 - DBB197	Assignment of the PLC for transfer parameters These data areas are reserved for the "Tool Ident Connection" option. (SINUMERIK ONE)							
DBB198 - DBB249	Assignment of the PLC for return values These data areas are reserved for the "Tool Ident Connection" option. (SINUMERIK ONE)							
DBB250 - DBB255	Function call of the PLC interface These data areas are reserved for the "Tool Ident Connection" option. (SINUMERIK ONE)							
DBB256 - DBB267	Commands for Paramtm.exe (SINUMERIK ONE)							
DBB268	Traffic light status (SINUMERIK ONE)							

LBP_HMI [DB19]	**Signals from the operator panel (OP) (Operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD270 - DBD394	Counter [1...32] (SINUMERIK ONE)							
DBB398	Handwheel number for simulation override							
DBW400	Simulation override							
DBW402	Simulation state							

LBP_HMI [DB19], sidescreen - interface for the MCP function

Table 10-39 Sidescreen - interface for the MCP function

LBP_HMI [DB19]	**Signals to/from the operator panel Signal from the operating software → PLC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB404	Customer keys							
	8	7	6	5	4	3	2	1
DBB405	Customer keys							
	16	15	14	13	12	11	10	9
DBB406	Customer keys							
	24	23	22	21	20	19	18	17
DBB407	Customer keys							
	32	31	30	29	28	27	26	25
DBB408	Customer keys							
	40	39	38	37	36	35	34	33
DBB409	Customer keys							
	48	47	46	45	44	43	42	41
DBB410	Customer keys							
	56	55	54	53	52	51	50	49
DBB411	Customer keys							
	64	63	62	61	60	59	58	57

LBP_HMI [DB19], sidescreen - interface for the MCP function

Table 10-40 Sidescreen - Interface for the MCP function signals to/from the operator panel

LBP_HMI [DB19]	**Signals to/from the operator panel [r/w] Signal from PLC → operating software**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB412	Customer LED							
	8	7	6	5	4	3	2	1
DBB413	Customer LED							
	16	15	14	13	12	11	10	9

LBP_HMI [DB19]	**Signals to/from the operator panel [r/w]** **Signal from PLC → operating software**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB414	Customer LED							
	24	23	22	21	20	19	18	17
DBB415	Customer LED							
	32	31	30	29	28	27	26	25
DBB416	Customer LED							
	40	39	38	37	36	35	34	33
DBB417	Customer LED							
	48	47	46	45	44	43	42	41
DBB418	Customer LED							
	56	55	54	53	52	51	50	49
DBB419	Customer LED							
	64	63	62	61	60	59	58	57
DBB420	Customer key deactivated							
	8	7	6	5	4	3	2	1
DBB421	Customer key deactivated							
	16	15	14	13	12	11	10	9
DBB422	Customer key deactivated							
	24	23	22	21	20	19	18	17
DBB423	Customer key deactivated							
	32	31	30	29	28	27	26	25
DBB424	Customer key deactivated							
	40	39	38	37	36	35	34	33
DBB425	Customer key deactivated							
	48	47	46	45	44	43	42	41
DBB426	Customer key deactivated							
	56	55	54	53	52	51	50	49
DBB427	Customer key deactivated							
	64	63	62	61	60	59	58	57

10.4.8.2 LBP_HMI[DB19]

Variables in "LBP_HMI [DB19]"

Data block "LBP_HMI [DB19]" contains signals to and from the operator panel (OP).

STRUCT "MMC2" (DBB50 ... DBB99) contains the interface of the second control.

Interface 1st and 2nd Control

A_SCLight, MMC2.A_SCLight

(DB19 DBX0.0, 50.0)

Description	Brighten screen
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	Screen darkening is active.
Value TRUE	The screen content is displayed. Screen darkening is ignored.
Corresponds to	• "LBP_HMI.A_SCDark" (DB19 DBX0.1, 50.1) Darken screen

A_SCDark, MMC2.A_SCDark

(DB19 DBX0.1, 50.1)

Description	Darken screen
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	The screen is brightened. In this state, the control automatically brightens/darkens the screen: • The screen is darkened if no keystroke is made at the keyboard for a time that is defined using the following machine data: MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL (time to darken the screen) • The screen is brightened the next time a key on the operator panel front is pressed.
Value TRUE	The screen is darkened. The automatic screen brightening/darkening is therefore inactive. The screen does not automatically become bright when a key on the keyboard is pressed. **Note** If the interface signal is set, and the screen is darkened, then the keyboard of the operator panel front still remains active. It is therefore recommended that the keyboard of the operator panel front is also locked: "LBP_HMI.A_KeyDisable" (DB19 DBX0.2, 50.2) = TRUE (key disable)
Corresponds to	• "LBP_HMI.A_SCLight" (DB19 DBX0.0, 50.0) Brighten screen • "LBP_HMI.A_KeyDisable" (DB19 DBX0.2, 50.2) Key disable • MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL Time to darken the screen

A_KeyDisable, MMC2.A_KeyDisable

(DB19 DBX0.2, 50.2)

Description	Key disable **Note** If the screen is darkened with ("LBP_HMI.A_SCDark" (DB19 DBX0.1, 50.1) = 1), the operator panel front keyboard still remains active. It is therefore recommended that the keyboard of the operator panel front is also locked: "LBP_HMI.A_KeyDisable" (DB19 DBX0.2, 50.2) = TRUE (key disable)
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	The keyboard is enabled for the operator.
Value TRUE	The keyboard is locked for the operator.
Corresponds to	• "LBP_HMI.A_SCLight" (DB19 DBX0.0, 50.0) Brighten screen • "LBP_HMI.A_SCDark" (DB19 DBX0.1, 50.1) Darken screen

A_ClearCanAl, MMC2.A_ClearCanAl

(DB19 DBX0.3, 50.3)

Description	Delete cancel alarms
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	The delete error key is not pressed on the operator panel.
Value TRUE	The delete error key on the operator panel is pressed ⇒ Request to acknowledge all cancel alarms of the NC and operator panel.
Corresponds to	• "LBP_HMI.E_CanCleared" (DB19 DBX20.3, 70.3) Cancel alarms cleared

A_ClearRecAl, MMC2.A_ClearRecAl

(DB19 DBX0.4, 50.4)

Description	Delete recall alarms
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	The delete error key is not pressed on the operator panel.
Value TRUE	The delete error key is pressed on the operator panel. Request to acknowledge all recall alarms of the NC and operator panel.
Corresponds to	• "LBP_HMI.E_RecCleared" (DB19 DBX20.4, 70.4) Recall alarms cleared

A_ActWCS, MMC2.A_ActWCS

(DB19 DBX0.7, 50.7)

Description	Actual values in the WCS
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	If the "machine" operating area is selected on the user interface, the axis-specific positions and distance-to-go are displayed in the previous active coordinate system.
Value TRUE	If the "machine" operating area is selected on the user interface, the axis-specific positions and distance-to-go are displayed in the workpiece coordinate system (WCS).
Corresponds to	• "LBP_HMI.E_ActWCS" (DB19 DBX20.7, 70.7) Switch MCS / WCS

A_PLCExtViewerReject

(DB19 DBX1.0)

Description	External viewer
Signal flow	PLC → OP
Data type	BOOL

A_PLCExtViewerMode

(DB19 DBX1.1)

Description	Rights for the external viewer
Signal flow	PLC → OP
Data type	BOOL

A_ASpi1percent, MMC2.A_ASpi1percent

(DB19 DBB6, 56)

Description	Analog spindle 1, utilization in percent
Signal flow	PLC → OP
Data type	UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Update	Cyclic

A_ASpi2percent, MMC2.A_ASpi2percent

(DB19 DBB7, 57)

Description	Analog spindle 2, utilization in percent
Signal flow	PLC → OP
Data type	UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Update	Cyclic

A_MCPChan, MMC2.A_MCPChan

(DB19 DBB8, 58)

Description	Channel number of the machine control panel (MCP) at the control
Signal flow	PLC → OP

Data type	UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Update	Cyclic
Value range	1 ... maximum number of NC channels

A_OEM1, MMC2.A_OEM1

(DB19 DBX9.0, 59.0)

Description	OEM1
Signal flow	PLC → OP

A_OEM2, MMC2.A_OEM2

(DB19 DBX9.1, 59.1)

Description	OEM2
Signal flow	PLC → OP

A_AToolMeas, MMC2.A_AToolMeas

(DB19 DBX9.2, 59.2)

Description	Autotool measure
Signal flow	PLC → OP

A_Hardkey, MMC2.A_Hardkey

(DB19 DBB10, 60)

Description	PLC hardkeys
Signal flow	PLC → OP
Data type	UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Update	Cyclic
Value 0	No selection
Value ≠ 0	1 ... 255: Selects the corresponding programming area

A_TeachDisable, MMC2.A_TeachDisable

(DB19 DBX13.0, 63.0)

Description	Disable teach-in transfer
Signal flow	PLC → OP
Data type	BOOL

A_PPdeload, MMC2.A_PPdeload

(DB19 DBX13.5, 63.5)

Description	NC program: Unload
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	Unload not active
Value TRUE	Unload active

A_PPload, MMC2.A_PPload

(DB19 DBX13.6, 63.6)

Description	NC program: Load
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	Load not active
Value TRUE	Load active

A_PPsel, MMC2.A_PPsel

(DB19 DBX13.7, 63.7)

Description	NC program: Selection
Signal flow	PLC → OP
Data type	BOOL
Update	Cyclic
Value FALSE	Selection not active
Value TRUE	Selection active

A_V24Index1, MMC2.A_V24Index1

(DB19 DBB14, 64)

Description	**PLC index (bits 0 to 6)** The V24 interface is controlled with bits. They specify the PLC index for the standard control file, which specifies the axis, channel or TO number. The file system is selected via bit 7: • DB19 DBX14.7, 64.7 == 1 ⇒ passive file system: PLC index for the user control file • DB19 DBX14.7, 64.7 == 0 ⇒ active file system: PLC index, which specifies the axis, channel or TO number **Selection of file system (bit 7)** Value FALSE: Active file system Value TRUE: Passive file system
Signal flow	PLC → OP
Data type	**Bits 0 to 6** UINT (unsigned, 0 ... 127_D or 0 ... $7F_H$) **Bit 7** BOOL
Update	Cyclic

A_V24Index2, MMC2.A_V24Index2

(DB19 DBB15, 65)

Description	PLC line offset The byte to control the V24 interface defines the line of the standard or user control file in which the control file to be transferred is specified. The control file is selected using "LBP_HMI.A_V24Index1" bit 7 (DB19 DBX14.7, 64.7): • == 1 ⇒ passive file system: PLC line offset in a user control file • == 0 ⇒ active file system: PLC line offset in a standard control file
Signal flow	PLC → OP
Data type	UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Update	Cyclic
Corresponds to	• "LBP_HMI.A_V24Index1" bit 7 (DB19 DBX14.7, 64.7) Selection of file system

A_V24NoFile1, MMC2.A_V24NoFile1

(DB19 DBB16, 66)

<table>
<tr><td>Description</td><td colspan="3">Program selection from the PLC: Index of the program list
Preselected programs/workpieces in the PLC program lists can be selected for machining by the NC via the PLC/HMI interface.
To specify the program list, its number (index) is output in binary code via this control byte. Bit 7 in the byte is always 1.
"LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) = <number of the program list></td></tr>
<tr><td></td><td>Number
(binary-coded)</td><td>Number
(decimal)</td><td>Program list</td></tr>
<tr><td></td><td>1000 0001</td><td>129</td><td>User program list (/user/sinumerik/hmi/plc/programlist/plc_proglist_user.ppl)</td></tr>
<tr><td></td><td>1000 0011</td><td>131</td><td>Manufacturer program list (/oem/sinumerik/hmi/plc/programlist/plc_proglist_manufacturer.ppl)</td></tr>
<tr><td>Signal flow</td><td colspan="3">PLC → OP</td></tr>
<tr><td>Data type</td><td colspan="3">BYTE</td></tr>
<tr><td>Update</td><td colspan="3">Cyclic</td></tr>
</table>

Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

A_V24NoFile2, MMC2.A_V24NoFile2

(DB19 DBB17, 67)

Description	Program selection from the PLC: Program index in the program list Preselected programs/workpieces in the PLC program lists can be selected for machining by the NC via the PLC/HMI interface. To specify the program within the program list selected via "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66), the program number is output in binary code via this control byte.	
	Area	**Program number**
	user	1 ... 100
	oem	201 ... 255
Signal flow	PLC → OP	
Data type	BYTE	
Update	Cyclic	
Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area	
Additional references	Function Manual for PLC	

E_SCDark, MMC2.E_SCDark

(DB19 DBX20.1, 70.1)

Description	Screen is dark
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The screen is not darkened.
Value TRUE	The screen is darkened.
Corresponds to	• "LBP_HMI.A_SCDark" (DB19 DBX0.1, 50.1) Darken the screen • MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL Time to darken the screen

E_Cancel, MMC2.E_Cancel

(DB19 DBX20.2, 70.2)

Description	Cancel key actuated
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	**Note** The interface signal must be reset in the PLC user program.
Value TRUE	Cancel key is pressed

E_CanCleared, MMC2.E_CanCleared

(DB19 DBX20.3, 70.3)

Description	Cancel alarm deleted
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	**Note** The interface signal must be reset in the PLC user program.
Value TRUE	Feedback signal: Cancel alarms were deleted
Corresponds to	• "LBP_HMI.A_ClearCanAl" (DB19 DBX0.3, 50.3) Clear cancel alarms

E_RecCleared, MMC2.E_RecCleared

(DB19 DBX20.4, 70.4)

Description	Recall alarms deleted
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	**Note** The interface signal must be reset in the PLC user program.

Value TRUE	Feedback signal: Recall alarms were cleared
Corresponds to	• "LBP_HMI.A_ClearRecAl" (DB19 DBX0.4, 50.4) Clear recall alarms

E_SimActiv, MMC2.E_SimActiv

(DB19 DBX20.6, 70.6)

Description	Simulation active
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The "Simulation" function is not active.
Value TRUE	An NC program has been selected at the user interface and the "Simulation" function activated for this program.
Additional references	• Operating Manuals; Section "Simulating machining"

E_ActWCS, MMC2.E_ActWCS

(DB19 DBX20.7, 70.7)

Description	Switch over MCS/WCS
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No effect
Value TRUE	Request to switch over the display of the axis-specific positions and residual distances to go on the user interface of the workpiece coordinate system (WCS) into the machine coordinate system (MCS) or from MCS to WCS. The signal is active for one PLC cycle.
Corresponds to	• "LBP_HMI.A_ActWCS" (DB19 DBX0.7, 50.7) Actual value in WCS

E_ActivWA, MMC2.E_ActivWA

(DB19 DBB21, 71)

Description	Number of the active operating area when the HMI monitor is not active.
Signal flow	OP → PLC
Data type	BYTE
Update	Cyclic
Value	Current operating area (201 ... 206)
Corresponds to	• MD9032 $MM_HMI_MONITOR = "String" where "String" = "DB<DB number>.DBB<byte address>"

E_Chan, MMC2.E_Chan

(DB19 DBB22, 72)

Description	Current channel number
Signal flow	OP → PLC

Data type	UINT (unsigned, 0 ... 255_D or 0 ... FF_H)
Update	Cyclic
Value	Valid channel numbers: 1 ... maximum number of NC channels

E_KeybHandl

(DB19 DBX23.0)

Description	Control 1: Keyboard has been used
Signal flow	OP → PLC
Data type	BOOL

E_DataTransf

(DB19 DBX23.1)

Description	Control 1: Data transfer active
Signal flow	OP → PLC
Data type	BOOL

E_DisplChange

(DB19 DBX23.2)

Description	Control 1: Screen change active
Signal flow	OP → PLC
Data type	BOOL

E_ScreenNo, MMC2.E_ScreenNo

(DB19 DBW24, 74)

Description	Display the current screen number If the HMI monitor is active, the current screen number is no longer displayed in this variable, but instead in the user-specific configured area of the HMI monitor.
Signal flow	HMI → PLC
Data type	INT
Update	Cyclic
Value > 0	Screen number
Additional references	• Function Manual for PLC

E_PPOK, MMC2.E_PPOK

(DB19 DBX26.1, 76.1)

Description	Program selection from the PLC: Job completed
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Job not completed
Value TRUE	Job completed

Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

E_PPErr, MMC2.E_PPErr

(DB19 DBX26.2, 76.2)

Description	Program selection from the PLC: Error
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Job completed correctly
Value TRUE	Job completed with errors
Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

E_PPAct, MMC2.E_PPAct

(DB19 DBX26.3, 76.3)

Description	Program selection from the PLC: Active
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No job active
Value TRUE	Job active
Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

E_PPdeload, MMC2.E_PPdeload

(DB19 DBX26.5, 76.5)

Description	Program selection from the PLC: Unload
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Unload inactive
Value TRUE	Unload active

Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

E_PPload, MMC2.E_PPload

(DB19 DBX26.6, 76.6)

Description	Program selection from the PLC: Load
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Load inactive
Value TRUE	Load active
Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

E_PPsel, MMC2.E_PPsel

(DB19 DBX26.7, 76.7)

Description	Program selection from the PLC: Selection
Signal flow	OP → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Selection inactive
Value TRUE	Selection active
Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • "LBP_HMI.E_PPError" (DB19 DBB27, 77) Program selection from the PLC: Error code • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

E_PPError, MMC2.E_PPError

(DB19 DBB27, 77)

Description	Program selection from the PLC: Error code Output byte for the error values of the data transfer.	
	Value	**Meaning**
	0	No error
	1	Invalid program list number "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66)
	3	User-spec. Program list plc_proglist_main.ppl not found (only for "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) ≠ 129, 131)
	4	Invalid program number "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67)
	5	The job list in the selected workpiece could not be opened.
	6	Error in job list (Job list Interpreter returns error)
	7	Job list Interpreter returns empty job list
Signal flow	OP → PLC	
Data type	BOOL	
Update	Cyclic	
Value FALSE	Selection inactive	

Value TRUE	Selection active
Corresponds to	• "LBP_HMI.A_PPsel" (DB19 DBX13.7, 63.7) NC program: Selection • "LBP_HMI.A_V24NoFile1" (DB19 DBB16, 66) Program selection from the PLC: Index of the program list • "LBP_HMI.A_V24NoFile2" (DB19 DBB17, 67) Program selection from the PLC: Program index in the program list • "LBP_HMI.E_PPOK..." (DB19 DBB26, 76) Program selection from the PLC: Status signals • MD9106 $MM_SERVE_EXTCALL_PROGRAMS Editing EXTCALL calls • MD51041 $MN_ENABLE_PROGLIST_USER Activation of PLC program list USER area • MD51043 $MN_ENABLE_PROGLIST_MANUFACT Activation of PLC program list MANUFACTURER area
Additional references	Function Manual for PLC

A_PictNo, MMC2.A_PictNo

(DB19 DBW28, 78)

Description	Screen number for "Supplement user interface"
Signal flow	OP → PLC
Data type	BOOL

A_PictOn, MMC2.A_PictOn

(DB19 DBX30.0, 80.0)

Description	Screen selection from the PLC: Control signal for screen selection
Signal flow	PLC → OP
Data type	BOOL

A_PictOff, MMC2.A_PictOff

(DB19 DBX30.1, 80.1)

Description	Screen selection from the PLC: Control signal for screen deselection
Signal flow	PLC → OP
Data type	BOOL

E_PictOK, MMC2.E_PictOK

(DB19 DBX31.0, 81.0)

Description	Screen selection from the PLC: Status signal for screen selection accepted
Signal flow	OP → PLC
Data type	BOOL

E_PictOn, MMC2.E_PictOn

(DB19 DBX31.1, 81.1)

Description	Screen selection from the PLC: Status signal of screen is selected
Signal flow	OP → PLC
Data type	BOOL

E_PictAct, MMC2.E_PictAct

(DB19 DBX31.2, 81.2)

Description	Screen selection from the PLC: Status signal of screen active
Signal flow	OP → PLC
Data type	BOOL

E_PictOff, MMC2.E_PictOff

(DB19 DBX31.3, 81.3)

Description	Screen selection from the PLC: Status signal of screen is deselected
Signal flow	OP → PLC
Data type	BOOL

E_PictErr, MMC2.E_PictErr

(DB19 DBX31.4, 81.4)

Description	Screen selection from the PLC: Status signal error Screen selection is not possible
Signal flow	OP → PLC
Data type	BOOL

E_NotAct, MMC2.E_NotAct

(DB19 DBX31.7, 81.7)

Description	Screen selection from the PLC: Status signal inactive
Signal flow	OP → PLC
Data type	BYTE

A_FuncNo, MMC2.A_FuncNo

(DB19 DBB32, 82)

Description	**Function selection from PLC**
	Function number (bits 0 to 5) The function number is entered in bits 0 to 5, which after setting the requesting strobe in bit 6, is executed. Function number 0: No function Function number 1: Channel selection
	Request strobe (Bit 6) Value FALSE: The function was executed by the NC or no function execution is requested Value TRUE: Execution of the set function by the NC is requested
	Busy function (bit 7) Value FALSE: The processing of the requested function has been completed. Value TRUE: The requested function is currently being processed by the NC.
Signal flow	PLC → OP
Data type	BYTE
Update	Cyclic
Corresponds to	• "LBP_HMI.A_ParX" (DB19 DBB33 ... 35, 83 ... 85) Parameter 1 ... 3 • "LBP_HMI.E_FuncErr" (DB19 DBB36, 86) Error code
Additional references	Function Manual for PLC

A_Par1, A_Par2, A_Par3, MMC2.A_Par1, MMC2.A_Par2, MMC2.A_Par3

(DB19 DBB33, 34, 35, 83, 84, 85)

Description	Parameters 1 to 3 for function selection No. Parameters can be specified for the function requested from "LBP_HMI.A_FuncNo" (DB19 DBB32, 82) in the function selection: • Parameter 1: "LBP_HMI.A_Par1" (DB19 DBB33, 83) • Parameter 2: "LBP_HMI.A_Par2" (DB19 DBB34, 84) • Parameter 3: "LBP_HMI.A_Par3" (DB19 DBB35, 85) The number and value range of the parameters depends on the associated function.
Signal flow	PLC → OP
Data type	BYTE
Update	Cyclic
Corresponds to	• "LBP_HMI.A_FuncNo" (DB19 DBB32, 82) Function number, request strobe, status • "LBP_HMI.E_FuncErr" (DB19 DBB36, 86) Error code
Additional references	Function Manual for PLC

E_FuncErr, MMC2.E_FuncErr

(DB19 DBB36, 86)

Description	Error code for function selection No. For the function requested from "LBP_HMI.A_FuncNo" (DB19 DBB32, 82) in the function selection, an error code can be displayed. The following error codes are output by the NC after completion of the function execution: • 0: No error • 1: Function number invalid "LBP_HMI.A_FuncNo" bit 0 ... 5 (DB19 DBX32.0 ... 32.5, 82.0 ... 82.5) • 2: Parameter invalid "LBP_HMI.A_ParX" (DB19 DBB33 ... 35, 83 ... 85) • 3: Error when writing the HMI-internal variable. • 10: Channel does not exist "LBP_HMI.A_Par1" (DB19 DBB33, 83)
Signal flow	OP → PLC
Data type	BYTE
Update	Cyclic
Corresponds to	• "LBP_HMI.A_FuncNo" (DB19 DBB32, 82) Function number, request strobe, status • "LBP_HMI.A_ParX" (DB19 DBB33 ... 35, 83 ... Parameters 1 - 385) Parameters 1 - 3
Additional references	Function Manual for PLC

Switchover interface to the operating software

Call waiting interface (operating software announces itself to the NC)

ONL_REQUEST

(DB19.DBW100)

Description	Online request from the operating software The operating software writes its client identification as an online request (bits 8 ... 15: bus type, bits 0 ... 7: bus address)
Data type	WORD

ONL_CONFIRM

(DB19.DBW102)

Description	Acknowledgement from PLC for online request PLC writes the operating software client identification as acknowledgement (bits 8 ... 15: bus type, bits 0 ... 7: bus address)
Data type	WORD

PAR_CLIENT_IDENT

(DB19.DBW104)

Description	Operating software writes its client identification (bits 8 ... 15: bus type, bits 0 ... 7: bus address)
Data type	WORD

PAR_MMC_TYP

(DB19.DBB106)

Description	Type of operating software acc. to NETNAMES.INI: Main/secondary operator panel / server /...
Data type	BYTE

PAR_MSTT_ADR

(DB19.DBB107)

Description	Operating software writes address of the MCP to be activated; 255, if no MCP activation
Data type	BYTE

PAR_STATUS

(DB19.DBB108)

Description	PLC writes the online enable for the operating software
Data type	BYTE

PAR_Z_INFO

(DB19.DBB109)

Description	PLC writes supplementary information on the status
Data type	BYTE

MtoNAlive

(DB19.DBW110)

Description	Sign-of-life from the PLC to the operating software using the M to N block
Data type	WORD

ParOpKeyAdr

(DB19.DBB113)

Description	Direct key index call waiting interface
Data type	BYTE

ParTcuIndex

(DB19.DBB114)

Description	TCU index call waiting interface
Data type	BYTE

Tcu1KeyAdr

(DB19.DBB116)

Description	Direct key address of 1st Online interface
Data type	BYTE

Tcu2KeyAdr

(DB19.DBB117)

Description	Direct key address of 2nd Online interface
Data type	BYTE

Tcu1Index

(DB19.DBB118)

Description	TCU index of 1st Online interface
Data type	BYTE

Tcu2Index

(DB19.DBB119)

Description	TCU index of 2nd Online interface
Data type	BYTE

Online interface OP 1 (user)

MMC1_CLIENT_IDENT

(DB19.DBW120)

Description	PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT when operating software goes online
Data type	WORD

MMC1_TYP

(DB19.DBB122)

Description	PLC writes PAR_MMC_TYP to MMCx_TYP when operating software goes online
Data type	BYTE

MMC1_MSTT_ADR

(DB19.DBB123)

Description	PLC writes PAR_MCP_ADR to MMCx_MCP_ADR when operating software goes online
Data type	BYTE

MMC1_STATUS

(DB19.DBB124)

Description	Connection state, operating software and PLC alternately write their requests/acknowledgements
Data type	BYTE

MMC1_Z_INFO

(DB19.DBB125)

Description	Additional information, connection state (pos./neg. acknowledgement, error messages, etc.)
Data type	BYTE

MMC1_SHIFT_LOCK

(DB19.DBX126.0)

Description	MMC1_SHIFT_LOCK
Data type	BOOL

MMC1_MSTT_SHIFT_LOCK

(DB19.DBX126.1)

Description	MMC1_MSTT_SHIFT_LOCK
Data type	BOOL

MMC1_ACTIVE_REQ

(DB19.DBX126.2)

Description	MMC1_ACTIVE_REQ
Data type	BOOL

MMC1_ACTIVE_PERM

(DB19.DBX126.3)

Description	MMC1_ACTIVE_PERM
Data type	BOOL

MMC1_ACTIVE_CHANGED

(DB19.DBX126.4)

Description	MMC1_ACTIVE_CHANGED
Data type	BOOL

MMC1_CHANGED_DENIED

(DB19.DBX126.5)

Description	MMC1_CHANGED_DENIED
Data type	BOOL

MMC1_TCU_SHIFT_LOCK

(DB19.DBX126.6)

Description	MMC1_TCU_SHIFT_LOCK
Data type	BOOL

MMC1_TRANS_DB

(DB19.DBW128)

Description	Reserved Transline (Transline DB number) (SINUMERIK ONE)
Data type	BOOL

Online interface OP 2 (user)

MMC2_CLIENT_IDENT

(DB19.DBW130)

Description	PLC writes PAR_CLIENT_IDENT to MMCx_CLIENT_IDENT when operating software goes online
Data type	WORD

MMC2_TYP

(DB19.DBB132)

Description	PLC writes PAR_MMC_TYP to MMCx_TYP when operating software goes online
Data type	BYTE

MMC2_MSTT_ADR

(DB19.DBB133)

Description	PLC writes PAR_MCP_ADR to MMCx_MCP_ADR when operating software goes online
Data type	BYTE

MMC2_STATUS

(DB19.DBB134)

Description	Connection state, operating software and PLC alternately write their requests/acknowledgements
Data type	BYTE

MMC2_Z_INFO

(DB19.DBB135)

Description	Additional information, connection state (pos./neg. acknowledgement, error messages, etc.)
Data type	BYTE

MMC2_SHIFT_LOCK

(DB19.DBX136.0)

Description	MMC2_SHIFT_LOCK
Data type	BOOL

MMC2_MSTT_SHIFT_LOCK

(DB19.DBX136.1)

Description	MMC2_MSTT_SHIFT_LOCK
Data type	BOOL

MMC2_ACTIVE_REQ

(DB19.DBX136.2)

Description	MMC2_ACTIVE_REQ
Data type	BOOL

MMC2_ACTIVE_PERM

(DB19.DBX136.3)

Description	MMC2_ACTIVE_PERM
Data type	BOOL

MMC2_ACTIVE_CHANGED

(DB19.DBX136.4)

Description	MMC2_ACTIVE_CHANGED
Data type	BOOL

MMC2_CHANGED_DENIED

(DB19.DBX136.5)

Description	MMC2_CHANGED_DENIED
Data type	BOOL

MMC2_TCU_SHIFT_LOCK

(DB19.DBX136.6)

Description	MMC2_TCU_SHIFT_LOCK
Data type	BOOL

MMC2_TRANS_DB

(DB19.DBW138)

Description	Reserved Transline (Transline DB number) (SINUMERIK ONE)
Data type	BOOL

Assignment of the PLC for transfer parameters (SINUMERIK ONE)

These data areas are reserved for the "Tool Ident Connection" option.

A_Subtype (DB19.DBW140)

A_Duplo (DB19.DBW142)

A_Ident (DB19 ab DBB144)

A_Tnumber (DB19.DBW178)

A_MagazinePlace (DB19.DBW180)

A_MagazinePlaceType (DB19.DBW182)

A_ToolSize_Upper (DB19.DBB184)

A_ToolSize_Down (DB19.DBB185)

A_ToolSize_Left (DB19.DBB186)

A_ToolSize_Right (DB19.DBB187)

A_Channel (DB19.DBB188)

A_LoadingPlace (DB19.DBB189)

A_Magazine (DB19.DBW190)

A_Unit (DB19.DBB192)

A_NCU_Index (DB19.DBB193)

A_FileIndex (DB19.DBW194)

A_ActiveEnable (DB19.DBX196.0)

A_Override (DB19.DBX196.1)

A_LoadTool (DB19.DBX196.2)

A_DeleteTool (DB19.DBX196.3)

Assignment of the PLC for return values (SINUMERIK ONE)

These data areas are reserved for the "Tool Ident Connection" option.

E_Subtype (DB19.DBW198)

E_Duplo (DB19.DBW200)

E_Ident (DB19 ab DBB202)

E_Tnumber (DB19.DBW236)

E_ToolState (DB19.DBW238)

E_MagazinePlaceType (DB19.DBW240)

E_ToolSize_Upper (DB19.DBB242)

E_ToolSize_Down (DB19.DBB243)

E_ToolSize_Left (DB19.DBB244)

E_ToolSize_Right

E_Magazine (DB19.DBW246)

E_MagazinePlace (DB19.DBW248)

Function call of the PLC interface (SINUMERIK ONE)

These data areas are reserved for the "Tool Ident Connection" option.

A_ReqMMC1 (DB19.DBX250.0)

A_ReqMMC2 (DB19.DBX250.1)

A_K_Code (DB19.DBB251)

E_Done (DB19.DBX252.0)

E_Error (DB19.DBX252.1)

E_Active (DB19.DBX252.2)

E_Ident_Ready (DB19.DBX252.7)

E_K_CodeError (DB19.DBB253)

E_Error_Code (DB19.DBW254)

Commands for Paramtm.exe (SINUMERIK ONE)

Commands for Paramtm.exe

E_Req (DB19.DBX256.0)

A_Done (DB19.DBX258.0)

A_Error (DB19.DBX258.1)

A_Active (DB19.DBX258.2)

E_TDC_Read (DB19.DBX259.0)

E_TDC_Write (DB19.DBX259.1)

E_TDC_Cancel (DB19.DBX259.2)

E_TDC_Read_before_Write (DB19.DBX259.3)

A_Error_Code (DB19.DBW260)

E_PMagazine (DB19.DBW262)

E_PMagazinePlace (DB19.DBW264)

E_PTnumber (DB19.DBW266)

Various signals

A_TPMStatus

(DB19.DBW268)

Description	Traffic light status (SINUMERIK ONE)
Data type	INT

A_TPMCounter

(DB19 from DBB270)

Description	Counters [1 ... 32] (SINUMERIK ONE)
Data type	ARRAY [1..32] OF DINT

A_HWeelSimOvr

(DB19.DBB398)

Description	Handwheel number for simulation override
Data type	BYTE

A_SimOvr

(DB19.DBW400)

Description	Simulation override
Data type	INT

A_SimState

(DB19.DBW402)

Description	Simulation state
Data type	INT

10.4.9 LBP_Chan1 [DB21], ... - Channel-specific signals

10.4.9.1 Overview

LBP_Chan1 [DB21], ... - Control signals to channel (1)

LBP_Chan1 [DB21], ...	**Signals to channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0		Activate dry run feedrate	Activate M01	Activate single block	Activate DRF			
DBB1	Activate program test	PLC action completed	CLC correction	CLC stop	Time monitoring active (tool management)	Disable all synchronized actions	Enable protection areas	Activate referencing
DBB2	Activate skip block							
	/7	/6	/5	/4	/3	/2	/1	/0
DBB3	Nibbling and punching							
			Manual stroke initiation	Stroke inoperative	Delayed stroke	Stroke suppression	Manual stroke initiation	Stroke enable
DBB4	Path feedrate override							
	H	G	F	E	D	C	B	A
DBB5	Path rapid traverse override							
	H	G	F	E	D	C	B	A
DBB6	Path feedrate override active	Path rapid traverse override active		Program level abort	Delete UP number of passes	Delete distance-to-go	Read-in disable	Feed disable
DBB7	Reset	Activate configured stop	Deactivate start disable	NC stop axes plus spindle	NC stop	NC stop at block limit	NC start	NC start disable

LBP_Chan1 [DB21], ...	Signals to channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB8	Activate machine-related protection area							
	8	7	6	5	4	3	2	1
DBB9	Activate machine-related protection area							
							10	9
DBB10	Activate channel-specific protection area							
	8	7	6	5	4	3	2	1
DBB11	Activate channel-specific protection area							
							10	9

Note

- Feedrate override active: Even if the feedrate override is not active (= 100%), the 0% position is still effective.
- Feedrate override: Either 31 positions (Gray code) with 31 MD for % evaluation, or 0% to 200% corresponding to the dual value in the byte (201 to 255 = max. 200%)
- Rapid traverse override: Either 31 positions (Gray code) with 31 MD for % evaluation, or 0% to 100% corresponding to the dual value in the byte (101 to 255 = max. 100%)
- Single block: Select the version using "Write variable"
- Delete distance-to-go: is only active for path axes and not for positioning axes

LBP_Chan1 [DB21], ... - Control signals to geometry axes

LBP_Chan1 [DB21], ...	Signals to channel (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Geometry axis 1							
DBB12	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB13		Requested machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB14	OEM signals							
DBB15								Invert handwheel direction of rotation
	Geometry axis 2							

LBP_Chan1 [DB21], ...	**Signals to channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB16	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB17		Requested machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB18	OEM signals							
DBB19								Invert handwheel direction of rotation
	Geometry axis 3							
DBB20	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB21		Requested machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB22	OEM signals							
DBB23								Invert handwheel direction of rotation

Note

The NC only evaluates the machine function signals if signal "LBP_NC.A_IncInModeGroup" (DB10 DBX57.0) "INC inputs in mode group area active" is not set.

LBP_Chan1 [DB21], ... - HMI signals to channel / OEM signals from/to channel

LBP_Chan1 [DB21], ...	**Signals from channel/PLC/operating software (operating software → PLC, PLC → NC, NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB24 Operating software → PLC		Dry run feedrate selected	M01 selected	M01 associated with NC selected	DRF selected		Activate configured stop	
DBB25 Operating software → PLC	Program test selected			REPOS mode change	Feedrate override selected for rapid traverse	REPOS mode C	B	A
DBB26 Operating software → PLC	Skip block requested, level /x							
	/7	/6	/5	/4	/3	/2	/1	/0
DBB27 Operating software → PLC							Skip block requested, level /x	
							/9	/8
DBB28 PLC → NC	OEM signals: Request							
DBB29 PLC → NC	Tool do not lock	Deactivate wear monitoring	Deactivate workpiece counter	Activate PTP traversal	Activate fixed feedrate 4	3	2	1
DBB30 PLC → NC	No tool change commands (SINUMERIK ONE)	Jog circles	Activate NC-associated M0/1	Contour handwheel simulation, negative direction	Contour handwheel simulation on	Activate contour handwheel C	B	A
DBB31 PLC → NC	Activate skip block		Invert contour handwheel direction of rotation	REPOS mode change		REPOS mode C	B	A
	/9	/8						
DBB32 NC → PLC		Last action block active	M00/M01 active	Approach block active	Action block active	RESU: Retrace support active	RESU: Retrace mode active	Execution from external active
DBB33 NC → PLC	Program test active	Transformation active	M02/M30 active	Block search active	Handwheel override active	Revolutional feedrate active	Orientable tool carrier active	Referencing active
DBB34 NC → PLC	OEM signals: Feedback signal							
DBB35 NC → PLC	Channel state			Program status				
	Reset	Interrupted	Active	Aborted	Interrupted	Stopped	Waiting	Running

LBP_Chan1 [DB21], ...	**Signals from channel/PLC/operating software (operating software → PLC, PLC → NC, NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB36 NC → PLC	NC alarm with machining stop is active	Channel-specific NC alarm is active	Channel is ready	Interrupt handling active	All axes stationary	All axes that have to be referenced are referenced		
DBB37 NC → PLC	"Stop at the end of block for single block (SBL)" is suppressed	Read-in disable is ignored	CLC stopped Upper limit	CLC stopped Lower limit	CLC active	Contour handwheel active		
						C	B	A
DBB38 NC → PLC	Nibbling and punching							
							Manual stroke initiation: Acknowledgement	Stroke initiation active
DBB39 NC → PLC	Stop at end of block due to configured stop	Configured stop is activated	Contour handwheel direction of rotation inverted active	Stop at end of block due to single block			NC alarm with program stop	Protection zone monitoring not guaranteed

LBP_Chan1 [DB21], ... - Control signals from geometry axes

LBP_Chan1 [DB21], ...	**Signals from channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Geometry axis 1							
DBB40	Traverse command		Traverse requests			Handwheel active		
	Plus	Minus	Plus	Minus		C	B	A
DBB41		Active machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB42	OEM signals							
DBB43								Handwheel direction of rotation inverted
DBB44 Operating software → PLC								
	Geometry axis 2							

LBP_Chan1 [DB21], ...	**Signals from channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB46	Traverse command		Traverse requests			Handwheel active		
	Plus	Minus	Plus	Minus		C	B	A
DBB47		Active machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB48	OEM signals							
DBB49								Handwheel direction of rotation inverted
DBB50								
Operating software → PLC								
	Geometry axis 3							
DBB52	Traverse command		Traverse requests			Handwheel active		
						C	B	A
DBB53		Active machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB54	OEM signals							
DBB55								Handwheel direction of rotation inverted
DBB56								
Operating software → PLC								

LBP_Chan1 [DB21], ... - Change signals for auxiliary function transfer from the channel

LBP_Chan1 [DB21], ...	**Signals from channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB58				change				
				M fct. 5	M fct. 4	M fct. 3	M fct. 2	M fct. 1
DBB59				Not decoded				
				M fct. 5	M fct. 4	M fct. 3	M fct. 2	M fct. 1

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60		Quick				change		
		S fct. 3	S fct. 2	S fct. 1		S fct. 3	S fct. 2	S fct. 1
DBB61		Quick				change		
		T fct. 3	T fct. 2	T fct. 1		T fct. 3	T fct. 2	T fct. 1
DBB62		Quick				change		
		D fct. 3	D fct. 2	D fct. 1		D fct. 3	D fct. 2	D fct. 1
DBB63				DL fct. Quick				DL fct.change
DBB64		Quick				change		
		H fct. 3	H fct. 2	H fct. 1		H fct. 3	H fct. 2	H fct. 1
DBB65			change					
			F fct. 6	F fct. 5	F fct. 4	F fct. 3	F fct. 2	F fct. 1
DBB66				Quick				
				M fct. 5	M fct. 4	M fct. 3	M fct. 2	M fct. 1
DBB67			Quick					
			F fct. 6	F fct. 5	F fct. 4	F fct. 3	F fct. 2	F fct. 1

Note

- For 10-decade T numbers, only the signal DBB61, DBX0 "T fct.1 change" is available.
- For 5-decade D numbers, only the signal DBB62, DBX0 "D fct.1 change" is available.

LBP_Chan1 [DB21], ... - Transferred M and S functions

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW68	Extended address, M function 1 (16-bit INT)							
DBD70	M function 1 (DINT)							
DBW74	Extended address, M function 2 (16-bit INT)							
DBD76	M function 2 (DINT)							
DBW80	Extended address, M function 3 (16-bit INT)							
DBD82	M function 3 (DINT)							
DBW86	Extended address, M function 4 (16-bit INT)							
DBD88	M function 4 (DINT)							
DBW92	Extended address, M function 5 (16-bit INT)							
DBD94	M function 5 (DINT)							
DBW98	Extended address, S function 1 (16-bit INT)							
DBD100	S function 1 (REAL format)							
DBW104	Extended address, S function 2 (16-bit INT)							
DBD106	S function 2 (REAL format)							

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW110	Extended address, S function 3 (16-bit INT)							
DBD112	S function 3 (REAL format)							

Note

M functions are programmed in the part program in the INTEGER format (8 decades plus leading sign).

LBP_Chan1 [DB21], ... - Transferred T/D/DL functions

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW116	Extended address, T function 1 (16-bit INT)							
DBW118 **DBD118**	T function 1 (dual) / For 8-decade T No., DBD118 is used as T function 1 (32-bit INT) (see note below)							
DBW120	Extended address, T function 2 (16-bit INT)							
DBW122	T function 2 (INT)							
DBW124	Extended address, T function 3 (16-bit INT)							
DBW126	T function 3 (INT)							
DBB128	Extended address D function 1 (8-bit INT)							
DBB129	D function 1 (binary)							
DBW130 **DBB130**	For 5-decade D No., DBW130 is used as D function 1 (16-bit INT) Extended address D function 2 (8-bit INT)							
DBB131	D function 2 (8-bit INT)							
DBB132	Extended address D function 3 (8-bit INT)							
DBB133	D function 3 (8-bit INT)							
DBW134	Extended address DL function (16-bit INT)							
DBD136	DL function (REAL)							

Note

- Programmed T functions are not output to the PLC when tool management is active.
- 8 decade T numbers are only available at DBD118 "T function 1".
- Programmed D functions with names (e.g. D=CUTTING EDGE_1) cannot be output to the PLC in ASCII format.
- 5-decade D numbers are only available as DBW130 "D function 1".
- The REAL format corresponds to the floating-point representation in STEP 7 (24-bit mantissa and 8-bit exponent). This floating point format supplies a maximum of 7 valid places.

LBP_Chan1 [DB21], ... - Transferred H/F functions

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBW140	Extended address, H function 1 (16-bit INT)							
DBD142	H function 1 (REAL or DINT), DINT also in DBD396							
DBW146	Extended address, H function 2 (16-bit INT)							
DBD148	H function 2 (REAL or DINT), DINT also in DBD400							
DBW152	Extended address, H function 3 (16-bit INT)							
DBD154	H function 3 (REAL or DINT), DINT also in DBD404							
DBW158	Extended address F function 1 (16-bit INT)							
DBD160	F function 1 (REAL format)							
DBW164	Extended address F function 2 (16-bit INT)							
DBD166	F function 2 (REAL format)							
DBW170	Extended address F function 3 (16-bit INT)							
DBD172	F function 3 (REAL format)							
DBW176	Extended address F function 4 (16-bit INT)							
DBD178	F function 4 (REAL format)							
DBW182	Extended address F function 5 (16-bit INT)							
DBD184	F function 5 (REAL format)							
DBW188	Extended address F function 6 (16-bit INT)							
DBD190	F function 6 (REAL format)							

Note

- F functions are programmed in the part program in the REAL data format.
- The extended address of the F function contains an identifier with the following meaning:
  - 0: Path feedrate
  - 1 – 31: Machine axis number for feedrate for positioning axes
- The data type of the H function depends on machine data: MD22110 $MC_AUXFU_H_TYPE_INT

LBP_Chan1 [DB21], ... - Decoded M signals

LBP_Chan1 [DB21], ...	Signals from channel (M0 - M99) (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Dynamic M functions							
DBB194								
	M07	M06	M05 #	M04 #	M03 #	M02	M01	M00
DBB195								
	M15	M14	M13	M12	M11	M10	M09	M08

LBP_Chan1 [DB21], ...	Signals from channel (M0 - M99) (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB196								
	M23	M22	M21	M20	M19	M18	M17	M16
DBB197								
	M31	M30	M29	M28	M27	M26	M25	M24
DBB198								
	M39	M38	M37	M36	M35	M34	M33	M32
DBB199								
	M47	M46	M45	M44	M43	M42	M41	M40
DBB200								
	M55	M54	M53	M52	M51	M50	M49	M48
DBB201								
	M63	M62	M61	M60	M59	M58	M57	M56
DBB202								
	M71	M70 #	M69	M68	M67	M66	M65	M64
DBB203								
	M79	M78	M77	M76	M75	M74	M73	M72
DBB204								
	M87	M86	M85	M84	M83	M82	M81	M80
DBB205								
	M95	M94	M93	M92	M91	M90	M89	M88
DBB206								
					M99	M98	M97	M96
DBB207								

Note

- #: The M function is not displayed here, if a spindle is parameterized in the channel. In this case, the M function is displayed as an extended M function under DB21,... DBB68 ff. and axially under DB31,... DBB86 ff.
- **Dynamic** M functions (M00 to M99) are decoded by the PLC basic program.
  Static M functions must be generated in the PLC user program from dynamic M functions.

LBP_Chan1 [DB21], ... - Active G functions

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB208	Number of the active G function of G function group 1 (8-bit INT)							
DBB209	Number of the active G function of G function group 2 (8-bit INT)							

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB210	Number of the active G function of G function group 3 (8-bit INT)							
DBB211	Number of the active G function of G function group 4 (8-bit INT)							
DBB212	Number of the active G function of G function group 5 (8-bit INT)							
DBB213	Number of the active G function of G function group 6 (8-bit INT)							
DBB214	Number of the active G function of G function group 7 (8-bit INT)							
DBB215	Number of the active G function of G function group 8 (8-bit INT)							
...	...							
DBB270	Number of the active G function of G function group n-1 (8-bit INT)							
DBB271	Number of the active G function of G function group n (8-bit INT)							

Note

- The active G functions of the group, for each programming of a G function or a mnemonic identifier (e.g. SPLINE), are updated.
- G functions within a G group are output as binary value, starting with 1. A G function with the value 0 means that for this G group, no G function is active.

LBP_Chan1 [DB21], ... Protection areas from channel

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB272	Machine-related protection area preactivated							
	8	7	6	5	4	3	2	1
DBB273	Machine-related protection area preactivated							
							10	9
DBB274	Channel-specific protection area preactivated							
	8	7	6	5	4	3	2	1
DBB275	Channel-specific protection area preactivated							
							10	9
DBB276	Machine-related protection area violated							
	8	7	6	5	4	3	2	1
DBB277	Machine-related protection area violated							
							10	9
DBB278	Channel-specific protection area violated							
	8	7	6	5	4	3	2	1
DBB279	Channel-specific protection area violated							
							10	9

LBP_Chan1 [DB21], ... - Synchronized actions, signals from/to channel

Note

The request signals should be set in the PLC user program. After data transfer, they are reset by the PLC basic program.

LBP_Chan1 [DB21], ...	**Signals to the channel (PLC ← → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB280 PLC → NC							Request Lock synchron-ized actions	Reserved
DBB281 NC → PLC							Acknowl-edgement Synchron-ized actions disabled	
DBW282 - DBW298	Reserved							
DBB300	Lock synchronized action							
PLC → NC	8	7	6	5	4	3	2	1
DBB301	Lock synchronized action							
PLC → NC	16	15	14	13	12	11	10	9
DBB302	Lock synchronized action							
PLC → NC	24	23	22	21	20	19	18	17
DBB303	Lock synchronized action							
PLC → NC	32	31	30	29	28	27	26	25
DBB304	Lock synchronized action							
PLC → NC	40	39	38	37	36	35	34	33
DBB305	Lock synchronized action							
PLC → NC	48	47	46	45	44	43	42	41
DBB306	Lock synchronized action							
PLC → NC	56	55	54	53	52	51	50	49
DBB307	Lock synchronized action							
PLC → NC	64	63	62	61	60	59	58	57
DBB308	Synchronized action can be locked							
NC → PLC	8	7	6	5	4	3	2	1
DBB309	Synchronized action can be locked							
NC → PLC	16	15	14	13	12	11	10	9
DBB310	Synchronized action can be locked							
NC → PLC	24	23	22	21	20	19	18	17
DBB311	Synchronized action can be locked							
NC → PLC	32	31	30	29	28	27	26	25

LBP_Chan1 [DB21], ...	**Signals to the channel (PLC ← → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB312	Synchronized action can be locked							
NC → PLC	40	39	38	37	36	35	34	33
DBB313	Synchronized action can be locked							
NC → PLC	48	47	46	45	44	43	42	41
DBB314	Synchronized action can be locked							
NC → PLC	56	55	54	53	52	51	50	49
DBB315	Synchronized action can be locked							
NC → PLC	64	63	62	61	60	59	58	57

LBP_Chan1 [DB21], ... - Control signals from/to channel

Note

The request signals should be set in the PLC user program. After data transfer, they are reset by the PLC basic program.

LBP_Chan1 [DB21], ...	**Signals from/to channel (PLC ← → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB316	Active G functions							
								G00 geo.
DBB317	Tool missing	PTP travel active	Drive test traverse request				Target number of workpieces reached	External language mode active
DBB318	Overstore active	Dry run feedrate active	M01 associated with NC active	Delayed stop	TOFF motion active	TOFF active	Block search via program test, SERUPRO, is active	ASUB stopped
DBB319	No tool change commands active (SINUMERIK ONE)	Feed stop Delay Suppress	REPOS delay	Feed stop Delay	Active REPOS mode			Acknowledgement of the REPOS mode change
					C	B	A	

LBP_Chan1 [DB21], ... - Signals to orientation axes

LBP_Chan1 [DB21], ...	**Signals to channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Orientation axis 1							
DBB320	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB321		Requested machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB322	OEM signals							
DBB323								Handwheel direction of rotation inverted
	Orientation axis 2							
DBB324	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB325		Requested machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB326	OEM signals							
DBB327								Handwheel direction of rotation inverted
	Orientation axis 3							
DBB328	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB329		Requested machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB330	OEM signals							
DBB331								Handwheel direction of rotation inverted

LBP_Chan1 [DB21], ... - Signals from orientation axes

LBP_Chan1 [DB21], ...	**Signals from channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Orientation axis 1							
DBB332	Traverse command		Traverse request			Handwheel active		
	Plus	Minus	Plus	Minus		C	B	A
DBB333		Active machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB334	OEM signals							
DBB335								Handwheel direction of rotation inversion active
	Orientation axis 2							
DBB336	Traverse command		Traverse request			Handwheel active		
	Plus	Minus	Plus	Minus		C	B	A
DBB337		Active machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB338	OEM signals							
DBB339								Handwheel direction of rotation inversion active
	Orientation axis 3							
DBB340	Traverse command		Traverse request			Handwheel active		
	Plus	Minus	Plus	Minus		C	B	A
DBB341		Active machine function						
		Continuous manual travel	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB342	OEM signals							
DBB343								Handwheel direction of rotation inversion active

LBP_Chan1 [DB21], ... - Tool management functions from channel

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	Change signals, tool management functions							
DBB344					Last replacement tool of the tool group	Transition to new replacement tool	Tool limit value reached	Tool prewarning limit reached
DBB345 - DBB347								
	Transferred tool management functions							
DBD348	T number for tool prewarning limit (DINT)							
DBD352	T number for tool limit value (DINT)							
DBD356	T number of new replacement tool (DINT)							
DBD360	T number of last replacement tool (DINT)							

LBP_Chan1 [DB21], ... - Control signals from/to channel (2)

LBP_Chan1 [DB21], ...	Signals from channel (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB364	CH_CYCLES_SIG_IN (1)							
	8	7	6	5	4	3	2	1
DBB365	CH_CYCLES_SIG_IN (2)							
	16	15	14	13	12	11	10	9
DBB366	CH_CYCLES_SIG_OUT (1)							
	8	7	6	5	4	3	2	1
DBB367	CH_CYCLES_SIG_OUT (2)							
	16	15	14	13	12	11	10	9
DBB368	CH_OEM_TECHNO_SIG_IN (1)							
	8	7	6	5	4	3	2	1
DBB369	CH_OEM_TECHNO_SIG_IN (2)							
	16	15	14	13	12	11	10	9
DBB370	CH_OEM_TECHNO_SIG_IN (3)							
	24	23	22	21	20	19	18	17
DBB371	CH_OEM_TECHNO_SIG_IN (4)							
	32	31	30	29	28	27	26	25
DBB372	CH_OEM_TECHNO_SIG_OUT (1)							
	8	7	6	5	4	3	2	1
DBB373	CH_OEM_TECHNO_SIG_OUT (2)							
	16	15	14	13	12	11	10	9
DBB374	CH_OEM_TECHNO_SIG_OUT (3)							
	24	23	22	21	20	19	18	17

LBP_Chan1 [DB21], ...	**Signals from channel (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB375	CH_OEM_TECHNO_SIG_OUT (4)							
	32	31	30	29	28	27	26	25
DBB376	Display of the triggering event in case of event-driven program call ProgEventDisplay							
DBB377		Jog circle active	Retraction data available	JOG retract active			Stop condition	Collision avoidance: Stop
DBB378							Silent ASUB is active	ASUB is active
DBB379								
DBB380	Reserved ASUB							
DBB381	Reserved ASUB							
DBB382	Reserved ASUB							
DBB383	Reserved ASUB							

LBP_Chan1 [DB21], ...	**Signals to channel (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB384								Enable GOTOS
DBB385	Grinding: Input signals 1 ... 8 ($AC_IN_KEY_G[1 ... 8])							
	8	7	6	5	4	3	2	1
DBB386	Grinding: Disable input signals 1 ... 8							
	8	7	6	5	4	3	2	1
DBB387	Grinding: Status of the grinding functions 1 ... 8 ($AC_IN_KEY_G_RUN_IN[1...8])							
	8	7	6	5	4	3	2	1

LBP_Chan1 [DB21], ...	**Signals from/to channel (NC ↔ PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB388 NC → PLC	Index number of the active transformation in the channel							
DBB389	Reserved							

LBP_Chan1 [DB21], ...	**Signals from/to channel (NC ↔ PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB390 PLC → NC	Grinding: Enable status of input signals 1 ... 8 ($AC_IN_KEY_G_ISENABLE[1...8])							
	8	7	6	5	4	3	2	1
DBB391 NC → PLC	Grinding: Status of the grinding functions 1 ... 8 ($AC_IN_KEY_G_RUN_OUT[1...8])							
	8	7	6	5	4	3	2	1
DBB392 PLC → NC	Selection of the coordinate system for Cartesian manual traversing and handwheel override in the automatic mode in the tool direction (DRF)							
DBB393	Reserved							
DBB394	Reserved							
DBB395	Reserved							
DBD396 NC → PLC	H function 1 (DInt), also in DBD142							
DBD400 NC → PLC	H function 2 (DInt), also in DBD146							
DBD404 NC → PLC	H function 3 (DInt), also in DBD150							
DBB408 HMI → PLC	Axis selection type: 0: Machine axis, 1: Geo axis							
DBB409 HMI → PLC	Axis number: Machine / geo axis							
DBB410 HMI → PLC								Axis selection key
DBB411	Reserved							
DBB412 PLC → HMI	Active axis selection (feedback message to the HMI) 0: Machine axis, 1: Geo axis							
DBB413 PLC → HMI	Active axis number (feedback message to the HMI): Machine / geo axis							

10.4.9.2 LBP_Chan1 [DB21], ... - Control signals to channel (1)

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

A_DRF (DB21, ... DBX0.3)

Description	Activate (DRF) handwheel offset With handwheel offset (DRF), an axis-specific offset can be applied in AUTOMATIC and MDI modes using a handwheel. **Note** Handwheel offset (DRF) is selected on the SINUMERIK Operate user interface in the operating area "Automatic" > "Program Control" by setting HMI/PLC interface signal "LBP_Chan*.E_MMC_DRF" (DB21, ... DBX24.3). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_DRF" (DB21, ... DBX0.3): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of handwheel offset (DRF) is not requested.
Value TRUE	Activation of handwheel offset (DRF) is requested.
Corresponds to	• "LBP_Chan*.E_MMC_DRF" (DB21, ... DBX24.3) Handwheel offset (DRF) selected
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_SBL (DB21, ... DBX0.4)

Description	Activate single block In AUTOMATIC and MDI modes, the operator must enable processing of each individual part program block of the part program selected in the channel by reactivating NC start.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to activate the "single block" function.
Value TRUE	A request has been made to activate the "single block" function.
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program mode, reset response" > "Program control" > "Single block"

A_M01 (DB21, ... DBX0.5)

Description	Activate M01 Program control "Conditional Stop" M01 is selected at the SINUMERIK Operate user interface: Operating area "Automatic" > "Program Control" by setting HMI/PLC interface signal "LBP_Chan*.E_MMC_M01" (DB21, ... DBX24.5). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is then transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_M01" (DB21, ... DBX0.5): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of program control "Conditional Stop" M01 is not requested.
Value TRUE	Activation of program control "Conditional Stop" M01 is requested.
Corresponds to	• "LBP_Chan*.E_MMC_M01" (DB21, ... DBX24.5) M01 selected • "LBP_Chan*.E_M01" (DB21, ... DBX32.5) M0/M01 active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_DRY (DB21, ... DBX0.6)

Description	Activate dry run feedrate (DRY) If the channel is in the "Reset" state, at the next NC start, traversing is performed with the currently active feedrate when G function `G01`, `G02`, `G03` are active: • Edge change 0 → 1: Dry run feedrate • Edge change 1 → 0: Programmed feedrate **Note** If the programmed feedrate is requested within a `G33` block, the programmed feedrate is not activated until the end of the block is reached. NC stop is not executed in a `G33` block. **Note** Dry run feedrate (DRY) is selected on the SINUMERIK Operate user interface in the operating area "Automatic" > "Program Control" by setting HMI/PLC interface signal "LBP_Chan*.E_MMC_DRY" (DB21, ... DBX24.6). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_DRY" (DB21, ... DBX0.6): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	Activation of dry run feedrate (DRY) is requested.

Edge change 1 → 0	Activation of dry run feedrate (DRY) is not requested.
Corresponds to	• "LBP_Chan*.E_MMC_DRY" (DB21, ... DBX24.6) Dry run feedrate (DRY) selected • SD42100 $SC_DRY_RUN_FEED Dry run feedrate
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Program control" > "Dry run feedrate" Function Manual Axes and Spindles, Section "Feedrates" > "Feedrate control" > "Dry run feedrate"

A_Ref (DB21, ... DBX1.0)

Description	Activate referencing Acknowledgement signal: "LBP_Chan*.E_Ref" (DB21, ... DBX33.0) (referencing active) **Note** Using the axis-specific machine data, the sequence in which the machine axes are referenced for channel-specific referencing can be defined: MD34110 $MA_REFP_CYCLE_NR For channel-specific referencing, when all axes entered in machine data MD34110 $MA_REFP_CYCLE_NR have reached their reference point, then interface signal "LBP_Chan*.E_AxesStop" (DB21, ... DBX36.3) (all axes stationary) is set.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	The "referencing" function is requested.
Edge change 1 → 0	No effect
Corresponds to	• "LBP_Chan*.E_Ref" (DB21, ... DBX33.0) Activate referencing • "LBP_Chan*.E_AxesStop" (DB21, ... DBX36.3) All axes stationary • MD34110 $MA_REFP_CYCLE_NR Axis sequence for channel-specific referencing
Additional references	Function Manual Axes and Spindles; Section "Referencing"

A_ProtZone (DB21, ... DBX1.1)

Description	Enable protection areas
Signal flow	PLC → NC
Data type	BOOL

A_SynOff (DB21, ... DBX1.2)

Description	Disable all synchronized actions
Signal flow	PLC → NC
Data type	BOOL

A_TM_TimOut (DB21, ... DBX1.3)

Description	Time monitoring active The tool life is always acquired as standard, if the geometry axes are not traversed with rapid traverse (G0). With the interface signal, acquiring the tool life is deactivated and then activated again. **Requirement** Release of the functionality with MD20310 TOOL_MANAGEMENT_MASK, bit 17 = TRUE
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Do not acquire tool life.
Value TRUE	Acquire tool life.
Corresponds to	• $A_MONIFACT Factor for reading tool monitoring • $TC_MOP1 Prewarning limit for tool life • MD18080 $MN_MM_TOOL_MANAGEMENT_MASK Memory reserved for the tool management • MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER Tool holder number • MD20310 TOOL_MANAGEMENT_MASK, bit 17 Activation of the tool management functions • MD20320 $MC_TOOL_TIME_MONITOR_MASK Time monitoring for tool in the tool holder
Additional references	Function Manual Tool Management; • Section "Function description" > "Tool monitoring (workpiece count, tool life, wear)" > "Tool life monitoring" • Section "Function description" > "Tool monitoring without active tool management" > "Tool life monitoring"

A_CLC_Stop (DB21, ... DBX1.4)

Description	Clearance control (CLC): Stop
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activating clearance control (CLC) is not requested.
Value TRUE	Activating clearance control (CLC) is requested (analogous to the program instruction CLC_GAIN=0.0).
Corresponds to	• CLC_GAIN=0.0; deactivate control loop gain • "LBP_Chan*.E_CLC" (DB21, ... DBX37.3) Clearance control (CLC) active
Additional references	Function Manual Monitoring and Compensating, Section: "Clearance control"

A_CLC_OR (DB21, ... DBX1.5)

Description	Clearance control (CLC): Override For signal state 1, the channel-specific feedrate override acts on the maximum velocity of the clearance control motion (MD62516): • Override settings < 100% The velocity limit set in MD62516 for clearance control is appropriately reduced. • Override settings > 100% The limit value from MD62516 is active. Signal state 0 is the maximum velocity of the clearance control motion, independent of the override setting.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The channel-specific feedrate override "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) will not be active for clearance control (CLC).
Value TRUE	The channel-specific feedrate override "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) will be active for clearance control (CLC).
Corresponds to	• "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) Feedrate override • "LBP_Chan*.A_FD_ORA" (DB21, ... DBX6.7) Feedrate override active • "LBP_Chan*.E_CLC" (DB21, ... DBX37.3) Clearance control (CLC) active • MD62516 $MC_CLC_SENSOR_VELO_LIMIT Velocity of clearance control motion
Additional references	Function Manual Monitoring and Compensating, Section: "Clearance control"

A_PLC_ActCompl (DB21, ... DBX1.6)

Description	PLC action completed **Note** At the end of the block search, concluding action blocks are executed: "LBP_Chan*.E_ActBlock" (DB21, ... DBX32.3) (action block active) == TRUE AND "LBP_Chan*.E_LastActBlock" (DB21, ... DBX32.6) (last action block active) == TRUE Alarm "10208 Channel <*Channel Number*> Issue NC start to continue program" notifies that a new NC start is required to continue the NC program from the target block. If other actions are to be executed by the PLC user program prior to the NC start (e.g. tool change), by parameterizing the search mode, the output of the alarm can be delayed until the following signal is set again: MD11450 $MN_SEARCH_RUN_MODE = 1
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The PLC action has not been completed yet.
Value TRUE	The PLC action has been completed.

Corresponds to	• "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start • "LBP_Chan*.E_ActBlock" (DB21, ... DBX32.3) Action block active • "LBP_Chan*.E_LastActBlock" (DB21, ... DBX32.6) Last action block active • "LBP_Chan*.E_BlockSearch" (DB21, ... DBX33.4) Block search active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"; • Section, "Block search, types 1, 2, and 4" • Section, "Block search type 5 (SERUPRO)"

A_ProgTest (DB21, ... DBX1.7)

Description	Activate program test (PRT) When program test (PRT) is active, the machine axes do not move when a block or NC program is executed. However, the axis movements are indicated on the user interface by changing setpoint position values. **Note** While program test (PRT) is active, all traversing motions of the axes, but not the spindles, take place at "Axis disable". **Note** PRT (program test) is selected at the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_ProgTest" (DB21, ... DBX25.7). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_ProgTest" (DB21, ... DBX1.7): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Notice** Because of the axis disable, the assignment of a tool magazine is not changed while the program is being tested. The user / machine manufacturer must utilize a suitable PLC user program to ensure that the NC-internal tool management and the actual assignment of the tool magazine remain consistent.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of program test (PRT) is not requested.
Value TRUE	Activation of program test (PRT) is requested.
Corresponds to	• "LBP_Chan*.E_MMC_ProgTest" (DB21, ... DBX25.7) Program test (PRT) requested • "LBP_Chan*.E_ProgTest" (DB21, ... DBX33.7) Program test (PRT) active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"; Section "Program test"

A_SKP0, **A_SKP1**, **A_SKP2, A_SKP3, A_SKP4, A_SKP5, A_SKP6, A_SKP7**

(DB21, ... DBX2.0 ... 2.7)

Description	Activate "skip block" (SKP) Bit 0 ... Bit 7 are assigned to the skip levels 0 ... 7. **Note** We recommend that the interface signal is already set before the NC program starts. **Note** For multiple consecutive skip blocks, the blocks are only skipped if the interface signal was already present before the first skip block of the block sequence is decoded. **Note** "Skip block" (SKP) is selected at the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_SKP*" (DB21, ... DBX26.0 ... 26.7) Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_SKP*" (DB21, ... DBX2.0 ... 2.7): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of "Skip Block" (SKP) is not requested.
Value TRUE	Activation of "Skip Block" (SKP) is requested.
Corresponds to	• "LBP_Chan*.E_MMC_SKP*" (DB21, ... DBX26.0 ... 26.7) "Skip Block" (SKP) selected
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_StrokEnab (DB21, ... DBX3.0)

Description	Stroke enable This signal releases the punching strokes via the PLC. It is not permissible that the NC initiates a punch stroke if the signal is not set. The NC waits until the enable signal is available before continuing the part program.
Signal flow	PLC → NC
Data type	BOOL
Value FALSE	Execution of punch strokes is locked.
Value TRUE	Execution of punch strokes is enabled.
Additional references	Function Manual Technologies

A_ManStrokEnab (DB21, ... DBX3.1)

Description	Punch interface 1: Manual stroke initiation This signal enables a single stroke to be initiated at the 1st punch interface in manual mode. In principle, this is possible in every operating mode, assuming that the axes do not move. The signal is ignored if the axes are moved during the manual stroke initiation. During the stroke, the axes are inhibited, i.e. they can only be moved again after the "Stroke initiation active" signal has been withdrawn.
Signal flow	PLC → NC
Data type	BOOL
Value FALSE	There is no request to execute a manual punch stroke at the 1st punch interface.
Value TRUE	Execute a manual punch stroke at the 1st punch interface.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable • "LBP_Chan*.A_ManRelStroke2" (DB21, ... DBX3.5) Punch interface 2: Manual stroke initiation • "LBP_Chan*.E_StrokeEA" (DB21, ... DBX38.0) Stroke initiation active • "LBP_Chan*.E_AcknManStrokEnab" (DB21, ... DBX38.1) Manual stroke initiation: Acknowledgement
Additional references	Function Manual Technologies Function Manual Technologies

A_StrokSup (DB21, ... DBX3.2)

Description	Stroke suppression The interface signal suppresses the fast "Initiate stroke" signal. This allows the part program to be processed without initiating a punching operation (dry run). With active path segmentation, the axes traverse in the "Stop and go" mode.
Signal flow	PLC → NC
Data type	BOOL
Value FALSE	There is no request to activate stroke suppression.
Value TRUE	Activate stroke suppression.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable
Additional references	Function Manual Technologies

A_StrokDelayed (DB21, ... DBX3.3)

Description	Delayed stroke A "delayed stroke" can be activated with this signal. This functionally corresponds to programming `PDELAYON`.
Signal flow	PLC → NC
Data type	BOOL
Value FALSE	There is no request to activate the option "delayed stroke".
Value TRUE	Activate "delayed stroke" option.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable
Additional references	Function Manual Technologies

A_StrokStop (DB21, ... DBX3.4)

Description	Stroke inoperative The NC responds to this interface signal by immediately stopping motion. An alarm is output if motion or another action is to be interrupted due to this signal. In physical terms, the interface signal is identical to the fast signal "Stroke active" for the NC, i.e. the system is wired in such a way that the two signals are routed to the same NC input through an AND logic operation.
Signal flow	PLC → NC
Data type	BOOL
Value FALSE	Stroke operative.
Value TRUE	Stroke inoperative.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable
Additional references	Function Manual Technologies

A_ManRelStroke2 (DB21, ... DBX3.5)

Description	Punch interface 2: Manual stroke initiation This signal enables a single stroke to be initiated at the 2nd punch interface in manual mode. In principle, this is possible in every operating mode, assuming that the axes do not move. The signal is ignored if the axes are moved during the manual stroke initiation. During the stroke, the axes are inhibited, i.e. they can only be moved again after the "Stroke initiation active" signal has been withdrawn.
Signal flow	PLC → NC
Data type	BOOL
Value FALSE	There is no request to initiate a manual punch stroke at the 2nd punch interface.
Value TRUE	Perform manual punch stroke at the 2nd punch interface.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable • "LBP_Chan*.A_ManStrokEnab" (DB21, ... DBX3.1) Punch interface 1: Manual stroke initiation • "LBP_Chan*.E_StrokeEA" (DB21, ... DBX38.0) Stroke initiation active • "LBP_Chan*.E_AcknManStrokEnab" (DB21, ... DBX38.1) Manual stroke initiation: Acknowledgement
Additional references	Function Manual Technologies

A_FD_OR (DB21, ... DBB4)

Description

Path feedrate override

Binary/Gray coding

The override factors can be specified in the binary or Gray-coded format. The control must be informed of the used format via the following machine data:

MD12020 $MN_OVR_FEED_IS_GRAY_CODE = <Coding>

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12030 $MN_OVR_FACTOR_FEEDRATE[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.00
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.05
21	11111	1.10
22	11101	1.15

	23	11100	1.20
	24	10100	1.20
	25	10101	1.20
	26	10111	1.20
	27	10110	1.20
	28	10010	1.20
	29	10011	1.20
	30	10001	1.20
	31	10000	1.20
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN		
Signal flow	PLC → NC		
Data type	BYTE		
Update	Cyclic		
Corresponds to	• "LBP_Chan*.A_FD_ORA" (DB21, ... DBX6.7) Feedrate override active		
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"		

1) Default values

A_RT_OR (DB21, ... DBB5)

Description

Path rapid traverse override

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12050 $MN_OVR_FACTOR_RAPID_TRA[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.00
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.00
21	11111	1.00
22	11101	1.00
23	11100	1.00
24	10100	1.00
25	10101	1.00

	26	10111	1.00
	27	10110	1.00
	28	10010	1.00
	29	10011	1.00
	30	10001	1.00
	31	10000	1.00
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN **Note** Path rapid traverse override is not active with the following functions when cutting threads: G33, G331, G332, G63		
Signal flow	PLC → NC		
Data type	BYTE		
Update	Cyclic		
Corresponds to	• "LBP_Chan*.A_RT_ORA" (DB21, ... DBX6.6) Path rapid traverse override active		
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"		

1) Default values

A_FDdisable (DB21, ... DBX6.0)

Description	Feed disable • The interface signal is active for all of the geometry, synchronized and positioning axes traversing in the channel. • If the interface signal is set while traversing an axis, the axis is braked down to standstill along its braking characteristic. For geometry axes, the axes are braked so that they follow the contour. • After the interface signal has been reset, traversing motion that was stopped is continued. • The position control is retained and the following error is eliminated. • If the interface signal is set, and a request is issued to traverse an axis, the axis is not traversed. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the axis is traversed. • The interface signal is active in all modes. **Thread cutting** During various thread cutting operations, the interface signal is not active:	
	Thread cutting	**Effectiveness**
	G33, G34, G35	Not effective
	G331, G332	Effective
	G63	Effective
Signal flow	PLC → NC	
Data type	BOOL	
Update	Cyclic	

Value FALSE	Feed disable is not active in the channel.
Value TRUE	Feed disable is active in the channel.
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feed disable and feed/spindle stop"

A_RIdisable (DB21, ... DBX6.1)

Description	Read-in disable If the read-in disable is set, the data transfer for the next block is enabled in the interpolator. **Note** Only active in the AUTOMATIC and MDI modes. **Application** If the auxiliary functions must be completed (e.g. for the tool change) before the next NC block can be processed, the automatic block change must be prevented through the read-in disable.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to activate the "read-in disable" function.
Value TRUE	A request has been made to activate the "read-in disable" function.
Corresponds to	• "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state running
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_DeleteDTG (DB21, ... DBX6.2)

Description	Delete distance-to-go (channel-specific) **Signal irrelevant for** • Positioning axes • Dwell time **Application example** Termination of the traversing motion on account of an external signal (e.g. probe)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic

Edge change 0 → 1	Delete distance-to-go (channel-specific) is requested. **Geometry axes** For a rising edge of the signal, the geometry axes currently traversing in the block are brought to a standstill along a braking ramp and then the distance-to-go is deleted. Any remaining following error is still corrected. The next traversing block is then loaded. **Path axes** "Delete distance-to-go" for path axes acts only in the AUTOMATIC mode. **Note** "Delete distance-to-go" causes the next traversing block to be prepared with the new positions. After a "Delete distance-to-go", the geometry axes thus follow a different contour to the one originally programmed in the NC program. Traversal to absolute positions (G90) in the block after "Delete distance-to-go" causes at least exact approach to the block end position. For incremental traversing (G91), the position specified in the NC program is not approached in the following block.
Edge change 1 → 0	No effect
Corresponds to	• "LBP_Axis*.A_DelDTGSpReset" (DB31, ... DBX2.2) Delete distance-to-go (axis-specific)

A_SP_Clear (DB21, ... DBX6.3)

Description	Delete UP number of passes
Signal flow	PLC → NC
Data type	BOOL

A_ProgAbort (DB21, ... DBX6.4)

Description	Program level abort After a program level abort, the calling program of the next higher program level is processed further from the call location. The main program level cannot be aborted with the program level abort, but only with a channel reset.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Signal state 0 → 1	Request to abort program processing of the actual program level (subprogram level, ASUB level, data save routine).
Signal state 1 → 0	Reset of the last request.
Corresponds to	• "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) reset
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_F_Limit (DB21, ... DBX6.5)

Description	Tool path feedrate limitation
Signal flow	PLC → NC
Data type	BOOL

A_RT_ORA (DB21, ... DBX6.6)

Description	Path rapid traverse override active If the path rapid traverse override is not active, a value of 1.0 ≙ 100% is used as the factor in the control irrespective of the current switch position. The first switch position is an exception. Depending on the selected coding, the following factor is active for the first switch position: • Binary coding: 0 • Gray coding: MD12050 $MN_OVR_FACTOR_RAPID_TRA[0]
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Path rapid traverse override is not active.
Value TRUE	Path rapid traverse override "LBP_Chan*.A_RT_OR" (DB21, ... DBB5) is active.
Corresponds to	• "LBP_Chan*.A_RT_OR" (DB21, ... DBB5) Path rapid traverse override • MD12020 $MN_OVR_FEED_IS_GRAY_CODE Path feedrate override switch Gray-coded • MD12030 $MN_OVR_FACTOR_FEEDRATE Evaluation of the path feedrate override switch
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"

A_FD_ORA (DB21, ... DBX6.7)

Description	Path feedrate override active If the path feedrate override is not active, a value of 1.0 ≙ 100% is used as override factor in the control irrespective of the current switch position. The first switch position is an exception. Depending on the selected coding, for the first switch position the following applies: • Binary coding: Override factor = 0 • Gray coding: Override factor = MD12030 $MN_OVR_FACTOR_FEEDRATE[0]
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The path feedrate override is not active.
Value TRUE	The path feedrate override "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) is active.
Corresponds to	• "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) Path feedrate override • MD12020 $MN_OVR_FEED_IS_GRAY_CODE Path feedrate override switch Gray-coded • MD12030 $MN_OVR_FACTOR_FEEDRATE Evaluation of the path feedrate override switch
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"

A_NCStartDisabl (DB21, ... DBX7.0)

Description	NC start disable If the NC start disable is set, start "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) of the NC program selected in the channel is ignored. **Note** The NC start disable is not active if the part program is started by the part program command `START` in another channel of the mode group. **Application** e.g. to suppress new program execution because there is no lubricant.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of the function "NC start disable" is not requested.
Value TRUE	Activation of the function "NC start disable" is requested.
Corresponds to	• "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_NCStart (DB21, ... DBX7.1)

Description	NC start AUTOMATIC mode: The selected NC program is started or continued, or the auxiliary functions that were saved during the program interruption are output. If data is transferred from the PLC to the NC during program state "Program interrupted," this data is immediately cleared at NC start. MDI mode: The blocks in the MDI block memory are executed.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Signal state 0 → 1	Request to start the program selected in the channel.
Signal state 1 → 0	Reset of the last request.
Corresponds to	• "LBP_Chan*.A_NCStartDisabl" (DB21, ... DBX7.0) NC start disable • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) reset
Additional references	Function Manual Basic Functions; • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Channel status" • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Program status"

A_NCStopBlock (DB21, ... DBX7.2)

Description	NC stop at block limit The actual block is executed to the end of the block. Program execution is then stopped: • Channel state: "Interrupted" • Program state: "Stopped"
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	NC stop at the end of the current block is not requested.
Value TRUE	NC stop at the end of the current block is requested.
Corresponds to	• "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.A_NCStopASp" (DB21, ... DBX7.4) NC stop axes plus spindles • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state stopped • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state interrupted
Additional references	Function Manual Basic Functions; • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Channel status" • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Program status"

A_NCStop (DB21, ... DBX7.3)

Description	NC stop AUTOMATIC or MDI mode Execution of the active part program in the channel is stopped. The traversing axes (not spindles) are braked down to standstill with the parameterized acceleration rates. • Channel state: Interrupted • Program state: Stopped JOG mode In the JOG mode, incompletely traversed incremental paths (INC...) are executed at the next NC start "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1). **Note** If data is transferred to the NC after "NC stop" (e.g. tool offsets), the data is taken into account with the next NC start.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	NC stop is not requested.
Value TRUE	NC stop is requested.

Corresponds to	• "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start • "LBP_Chan*.A_NCStopASp" (DB21, ... DBX7.4) NC stop axes plus spindles • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state stopped • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state interrupted
Additional references	Function Manual Basic Functions; • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Channel status" • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Program status"

A_NCStopASp (DB21, ... DBX7.4)

Description	NC stop axes plus spindles AUTOMATIC or MDI mode Execution of the active part program in the channel is stopped. The traversing axes and spindles are braked down to standstill with the parameterized acceleration rates. • Channel state: Interrupted • Program state: Stopped JOG mode In the JOG mode, incompletely traversed incremental paths (INC...) are executed at the next NC start "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1). **Note** If data is transferred to the NC after "NC stop" (e.g. tool offsets), the data is taken into account with the next NC start.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	NC stop is not requested.
Value TRUE	NC stop is requested.
Corresponds to	• "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state stopped • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state interrupted
Additional references	Function Manual Basic Functions; • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Channel status" • Section "Mode group, channel, program mode, reset response" > "Program mode" > "Program status"

A_SuppStartLock (DB21, ... DBX7.5)

Description	Suppress start disable
Signal flow	PLC → NC
Data type	BOOL

A_CfgStopFunc (DB21, ... DBX7.6)

Description	Activate configured stop **Note** "Activate configured stop" (CST) can also be selected via the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_CfgStopFunc" (DB21, ... DBX24.1). Depending on the value of the "LBP_ConfigBP [FC1]" parameter MMCToIf, the interface signal is transferred from the PLC basic program to the NC/PLC interface signal "LBP_Chan*.A_CfgStopFunc" (DB21, ... DBX7.6): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of the configured stop is not requested.
Value TRUE	Activation of the configured stop is requested.
Corresponds to	• "LBP_Chan*.E_MMC_CfgStopFunc" (DB21, ... DBX24.1) Activate configured stop

A_Reset (DB21, ... DBX7.7)

Description	Reset Responses: • All traversing axes and spindles are decelerated without contour violation to zero speed according to their acceleration curves. • All channel-specific reset positions become active. • All alarms of the channel, other than Power On alarms, are deleted. • Channel state: "Reset" ("LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) == TRUE) • Program state of an active NC program: "Canceled" ("LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) == TRUE) **Note** After a channel reset from the PLC user program, before a new action can be requested in the channel, e.g. NC start or axis interchange, the system must wait for the channel state "reset" to be indicated at the NC/PLC interface: "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) == TRUE
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Channel reset is not requested.
Value TRUE	Channel reset is requested.

Corresponds to	• "LBP_ModeGroup.A_MGReset" (DB11 DBX0.7, 20.7, ...,180.7) Mode group reset • "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_MRP (DB21, ... ab DBX8.0)

Description	Activate machine-related protection area 1 - 10
Signal flow	PLC → NC
Data type	ARRAY[1..10] of BOOL
Update	Cyclic
Value FALSE	Request to deactivate the active machine-related protection area. The protection area is immediately inactive. **Note** Deactivating the protection area is only active if it was pre-activated via an NC program and was activated via the interface signal.
Value TRUE	Request to activate the machine-related protection area. The protection area is immediately active. **Note** Activating the protection area is only effective if it was already pre-activated via an NC program.
Corresponds to	• "LBP_Chan*.A_ProtZone" (DB21, ... DBX1.1) Enable protection areas • "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) Activate machine-related protection area 1 - 10 • "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) Activate channel-specific protection area 1 - 10 • "LBP_Chan*.E_MRPactivated" (DB21, ... DBX272.0 ... 273.1) Machine-related protection area 1 - 10 pre-activated • "LBP_Chan*.E_CSPactivated" (DB21, ... DBX274.0 ... 275.1) Channel-specific protection area 1 - 10 pre-activated • "LBP_Chan*.E_MRPViolated" (DB21, ... DBX276.0 ... 277.1) Machine-related protection area 1 - 10 violated • "LBP_Chan*.E_CSPViolated" (DB21, ... DBX278.0 ... 279.1) Channel-specific protection area 1 - 10 violated
Additional references	Function Manual Basic Functions; Section "Monitoring and compensating" > "Protection areas"

A_CSP (DB21, ... from DBX10.0)

Description	Activate channel-specific protection area 1 - 10
Signal flow	PLC → NC

Data type	ARRAY[1..10] of BOOL
Update	Cyclic
Value FALSE	Request to deactivate the active channel-specific protection area. The protection area is immediately inactive. **Note** Deactivating the protection area is only active if it was pre-activated via an NC program and was activated via the interface signal.
Value TRUE	Request to activate the channel-specific protection area. The protection area is immediately active. **Note** Activating the protection area is only effective if it was already pre-activated via an NC program.
Corresponds to	• "LBP_Chan*.A_ProtZone" (DB21, ... DBX1.1) Enable protection areas • "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) Activate machine-related protection area 1 - 10 • "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) Activate channel-specific protection area 1 - 10 • "LBP_Chan*.E_MRPactivated" (DB21, ... DBX272.0 ... 273.1) Machine-related protection area 1 - 10 pre-activated • "LBP_Chan*.E_CSPactivated" (DB21, ... DBX274.0 ... 275.1) Channel-specific protection area 1 - 10 pre-activated • "LBP_Chan*.E_MRPViolated" (DB21, ... DBX276.0 ... 277.1) Machine-related protection area 1 - 10 violated • "LBP_Chan*.E_CSPViolated" (DB21, ... DBX278.0 ... 279.1) Channel-specific protection area 1 - 10 violated
Additional references	Function Manual Basic Functions; Section "Monitoring and compensating" > "Protection areas"

10.4.9.3 LBP_Chan1 [DB21], ... - Control signals to geometry axes

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

A_Geo (DB21, ... from DBX12.0)

Description	Control signals to the geometry axes
Data type	ARRAY[1..3] of STRUCT STRUCT "A_Geo", control signals for geometry axis (Page 350)

10.4.9.4 LBP_Chan1 [DB21], ... - HMI signals to channel / OEM signals from/to channel

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_MMC_CfgStopFunc (DB21, ... DBX24.1)

Description	Activate configured stop (CST) via the operating software **Note** "Activate configured stop" (CST) is selected via the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_CfgStopFunc" (DB21, ... DBX24.1). Depending on the value of the "LBP_ConfigBP [FC1]" parameter MMCToIf, the interface signal is transferred from the PLC basic program to the NC/PLC interface signal "LBP_Chan*.A_CfgStopFunc" (DB21, ... DBX7.6): • "TRUE": Transmission" • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of the configured stop is not requested.
Value TRUE	Activation of the configured stop is requested.
Corresponds to	• "LBP_Chan*.A_CfgStopFunc" (DB21, ... DBX7.6) Activate configured stop

E_MMC_DRF (DB21, ... DBX24.3)

Description	Handwheel offset (DRF) selected As soon as handwheel offset (DRF) is active, the handwheel offset can be changed in AUTOMATIC or MDI mode using the handwheel assigned to the axis. **Note** Handwheel offset (DRF) is selected on the SINUMERIK Operate user interface in the operating area "Automatic" > "Program Control" by setting HMI/PLC interface signal "LBP_Chan*.E_MMC_DRF" (DB21, ... DBX24.3). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_DRF" (DB21, ... DBX0.3): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Handwheel offset (DRF) is not selected.
Value TRUE	Handwheel offset (DRF) is selected.
Corresponds to	• "LBP_Chan*.A_DRF" (DB21, ... DBX0.3) Activate handwheel offset (DRF)
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_MMC_M01AssocNC (DB21, ... DBX24.4)

Description	Associated help function selected "Associated help function" (M-1) is selected at the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_M01AssocNC" (DB21, ... DBX24.4). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_NCKrelatedM01" (DB21, ... DBX30.5): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	HMI → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	Associated help function is not selected.
Value TRUE	Associated help function is selected.
Corresponds to	• "LBP_Chan*.A_NCKrelatedM01" (DB21, ... DBX30.5) Activate associated help function • "LBP_Chan*.E_AssM01" (DB21, ... DBX318.5) Associated help function active • MD22254 $MC_AUXFU_ASSOC_M0_VALUE Additional M function for program stop • MD22256 $MC_AUXFU_ASSOC_M1_VALUE Additional M function for conditional stop
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC" > "Associated auxiliary functions"

E_MMC_M01 (DB21, ... DBX24.5)

Description	M01 selected Program control "Conditional Stop" M01 is selected at the SINUMERIK Operate user interface: Operating area "Automatic" > "Program Control" by setting HMI/PLC interface signal "LBP_Chan*.E_MMC_M01" (DB21, ... DBX24.5). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is then transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_M01" (DB21, ... DBX0.5): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program control "Conditional stop"M01 is not selected.
Value TRUE	Program control "Conditional stop"M01 is selected.

Corresponds to	• "LBP_Chan*.A_M01" (DB21, ... DBX0.5) Activate M01 • "LBP_Chan*.E_M01" (DB21, ... DBX32.5) M00 / M01 active
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program mode, reset response" > "Program control" > "Function selection via user interface or PLC user program" • Operating Manual, HMI Advanced "Machine operating area"

E_MMC_DRY (DB21, ... DBX24.6)

Description	Dry run feedrate (DRY) selected **Note** Dry run feedrate (DRY) is selected on the SINUMERIK Operate user interface in the operating area "Automatic" > "Program Control" by setting HMI/PLC interface signal "LBP_Chan*.E_MMC_DRY" (DB21, ... DBX24.6). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_DRY" (DB21, ... DBX0.6): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Note** The value for the dry run feedrate is set using: SD42100 $SC_DRY_RUN_FEED
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Dry run feedrate (DRY) is not selected.
Value TRUE	Dry run feedrate (DRY) is selected.
Corresponds to	• "LBP_Chan*.A_DRY" (DB21, ... DBX0.6) Activate dry run feedrate (DRY) • SD42100 $SC_DRY_RUN_FEED Dry run feedrate • SD42101 $SC_DRY_RUN_FEED_MODE Mode for dry run velocity
Additional references	• Function Manual Basic Functions – Section "Mode group, channel, program mode, reset response" > "Program test" > "Program processing with dry run feedrate" – Section "Feedrates" > "Feedrate control" > "Dry run feedrate" • Operating Manual – Section "Machining a workpiece" > "Controlling the program sequence" > "Program controls"

E_MMC_REPOSPM_0, E_MMC_REPOSPM_1, E_MMC_REPOSPM_2

(DB21, ... DBX25.0, 25.1, 25.2)

Description	REPOS mode
Signal flow	HMI → PLC
Data type	BOOL

E_MMC_FD_OR4RT_OR (DB21, ... DBX25.3)

Description	Feedrate override selected for rapid traverse If there is no dedicated rapid traverse override switch on the machine control panel, you can toggle between feedrate override and rapid traverse override. Rapid traverse override can be selected by selecting feedrate override for rapid traverse at the user interface. The interface signal is set: "LBP_Chan*.E_MMC_FD_OR4RT_OR" (DB21, ... DBX25.3) = TRUE The following interface signals are set by the PLC basic program: • Rapid traverse override active = feedrate override for rapid traverse selected "LBP_Chan*.A_RT_ORA" (DB21, ... DBX6.6) = "LBP_Chan*.E_MMC_FD_OR4RT_OR" • Rapid traverse override = feedrate override "LBP_Chan*.A_RT_OR" (DB21, ... DBB5) = "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) Override factors> 100% are limited to 100%. **Note** The PLC user program can also initiate the switchover between rapid traverse and feedrate override. To do this, the following interface signals must be set: • Rapid traverse override active "LBP_Chan*.A_RT_ORA" (DB21, ... DBX6.6) = TRUE • Rapid traverse override = feedrate override "LBP_Chan*.A_RT_OR" (DB21, ... DBB5) = "LBP_Chan*.A_FD_OR" (DB21, ... DBB4)
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Dry run feedrate (DRY) is not selected.
Value TRUE	Dry run feedrate (DRY) is selected.
Corresponds to	• "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) Feedrate override • "LBP_Chan*.A_RT_OR" (DB21, ... DBB5) Rapid traverse override • "LBP_Chan*.A_RT_ORA" (DB21, ... DBX6.6) Rapid traverse override active
Additional references	• Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"

E_MMC_REPOSMode (DB21, ... DBX25.4)

Description	REPOS mode change
Signal flow	HMI → PLC
Data type	BOOL

E_MMC_ProgTest (DB21, ... DBX25.7)

Description	Program test selected
Signal flow	HMI → PLC
Data type	BOOL

E_MMC_SKP0, **E_MMC_SKP1**, **E_MMC_SKP2, E_MMC_SKP3, E_MMC_SKP4, E_MMC_SKP5, E_MMC_SKP6, E_MMC_SKP7, E_MMC_SKP8, E_MMC_SKP9**

(DB21, ... DBX26.0 ... 27.1)

Description	Skip block requested, level /x	
	E_MMC_SKP0 (DB21, ... DBX26.0)	/0
	E_MMC_SKP1 (DB21, ... DBX26.1)	/1
	...	...
	E_MMC_SKP9 (DB21, ... DBX27.1)	/9
Signal flow	HMI → PLC	
Data type	BOOL	

A_OEM (DB21, ... DBB28)

Description	OEM signals: Request
Signal flow	PLC → NC
Data type	BYTE

A_FixedFD1, A_FixedFD2, A_FixedFD3, A_FixedFD4

(DB21, ... DBX29.0, 29.1, 29.2, 29.3)

Description

Activate fixed feedrate 1 to 4, path/geometry axes

With the interface signals, in the AUTOMATIC mode, instead of the programmed feedrate or the configured JOG velocities, the fixed feedrate - parameterized using machine data - is activated.

A_FixedFD4 Bit 3	A_FixedFD3 Bit 2	A_FixedFD2 Bit 1	A_FixedFD1 Bit 0	Meaning
0	0	0	0	Fixed feedrate not selected
0	0	0	1	Fixed feedrate 1
0	0	1	0	Fixed feedrate 2
0	1	0	0	Fixed feedrate 3
1	0	0	0	Fixed feedrate 4

The fixed feedrates are parameterized using the following machine data:

- Linear axes: MD12202 $MN_PERMANENT_FEED
- Rotary axes: MD12204 $MN_PERMANENT_ROT_AX_FEED

Notes

- The fixed feedrate is not used for:
  - Spindles
  - Positioning axes
  - Tapping
- The fixed feedrate is always interpreted as linear feedrate. Switchover to linear feedrate is performed internally in the control system even when revolutional feedrate is active.

Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Corresponds to	• MD12200 $MN_RUN_OVERRIDE_0 Traversing behavior when override 0 • MD12202 $MN_PERMANENT_FEED Fixed feedrates for linear axes • MD12204 $MN_PERMANENT_ROT_AX_FEED Fixed feedrates for rotary axes
Additional references	• Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate control" > "Fixed feedrate values"

A_PTP_Travel (DB21, ... DBX29.4)

Description	Activate PTP travel Using this signal, in JOG mode, you can switch between Cartesian path movement (CP) and the Cartesian PTP travel. **Note** The variable is only relevant in JOG mode when transformation is active.
Signal flow	PLC → NC
Data type	BOOL
Edge change 0 → 1	Activate PTP (Point-To-Point) travel.
Edge change 1 → 0	Activate CP (Continuous-Path) travel.
Corresponds to	• "LBP_Chan*.E_PTP_Travel" (DB21, ... DBX317.6) PTP travel active
Additional references	• Function Manual Technologies

A_DisablWP_Counter (DB21, ... DBX29.5)

Description	TOOLMAN: Deactivate workpiece counter The workpiece counter monitoring allows workpieces to be counted, depending on the process, the workpiece material or other factors. Workpiece counter monitoring can be deactivated with the interface signal.
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	Deactivate workpiece counter monitoring is not requested.
Value TRUE	Deactivate workpiece counter monitoring is requested.

Corresponds to	• SETPIECE Function • TMPCIT PI service • $TC_MOP4 Remaining quantity • $TC_TP9 Type of tool monitoring
Additional references	Function Manual Tool Management; • Section "Function description" > "Tool monitoring (workpiece count, tool life, wear)" > "Workpiece counter monitoring" • Section "Function description" > "Tool monitoring without active tool management" > "Workpiece counter monitoring"

A_DisablWearMon (DB21, ... DBX29.6)

Description	TOOLMAN: Deactivate wear monitoring
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	Deactivate wear monitoring is not requested.
Value TRUE	Deactivate wear monitoring is requested.
Corresponds to	• $TC_MOP15 Wear setpoint or additive offset value • $TC_MOP5 Wear pre-warning limit or additive offset pre-warning limit • $TC_MOP6 Wear actual value or additive offset actual value • $TC_TP9 Type of tool monitoring • MD18080 $MN_MM_TOOL_MANAGEMENT_MASK Memory reserved for the tool management • MD20310 TOOL_MANAGEMENT_MASK, bit 17 Activation of the tool management functions
Additional references	Function Manual Tool Management; • Section "Function description" > "Tool monitoring (workpiece count, tool life, wear)" > "Wear monitoring" • Section "Function description" > "Tool monitoring without active tool management" > "Wear monitoring"

A_DontDisablTool (DB21, ... DBX29.7)

Description	TOOLMAN: Tool lock not active A tool enters the "locked" state if the actual value of the active monitoring function (workpiece count, tool life or wear) has reached a value of zero. If the tool is still being used, it remains in use (for machining) until the next tool change. After that the tool can no longer be used. Using the interface signal, it can be set that the NC does not take the "locked" state into account when searching for a tool that can be used. The interface signal has no effect if tool selection is realized using Init blocks (reset and start mode mask).
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	Tool lock is active
Value TRUE	Tool lock is not active
Corresponds to	• MD22562 $MC_TOOL_CHANGE_ERROR_MODE response to tool change errors
Additional references	Function Manual, Tool Management; Chapter "Function description" > "Tool monitoring (workpiece count, tool life, wear)" > "Wear monitoring" > "Signals to the PLC and from the PLC"

A_ContHW1, **A_ContHW2**, **A_ContHW3**

(DB21, ... DBX30.0, 30.1, 30.2)

Description

Activate contour handwheel

The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.

Bit coded: maximum 3 handwheels

A_ContHW3 Bit 2	A_ContHW2 Bit 1	A_ContHW1 Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
1	0	0	3

Binary coded: maximum 6 handwheels

A_ContHW3 Bit 2	A_ContHW2 Bit 1	A_ContHW1 Bit 0	Number of the assigned handwheel
0	0	0	No handwheel assignment
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6

Note

At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".

Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_Chan*.E_CHW*" (DB21, ... DBX37.0, 37.1, 37.2) contour handwheel active • MD11324 $MN_HANDWH_VDI_REPRESENTATION Representation of the handwheel number at the VDI interface
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_ContHWsim (DB21, ... DBX30.3)

Description	Simulation contour handwheel: switch on During simulation, the feedrate is no longer defined by the contour handwheel, but traversing occurs with the programmed feedrate on the contour. When deactivated, the actual traversing motion is braked along the braking ramp. **Note** Simulation is only active in AUTOMATIC mode and can only be activated when the contour handwheel is activated.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Deactivate contour handwheel simulation.
Value TRUE	Activate contour handwheel simulation.
Corresponds to	• "LBP_Chan*.A_ContHWneg" (DB21, ... DBX30.4) Simulation contour handwheel: Negative direction
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_ContHWneg (DB21, ... DBX30.4)

Description	Simulation contour handwheel: Negative direction When switching over the traversing direction, the actual traversing motion is braked along the braking ramp and then traversing is performed in the opposite direction.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	During simulation of the contour handwheel, travel is along the contour in the programmed direction.
Value TRUE	During simulation of the contour handwheel, travel is along the contour in the opposite direction to the programmed direction.
Corresponds to	• "LBP_Chan*.A_ContHWsim" (DB21, ... DBX30.3) Simulation contour handwheel: switch on
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_NCKrelatedM01 (DB21, ... DBX30.5)

Description	Activate associated auxiliary function "Associated help function" (M-1) is selected at the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_M01AssocNC" (DB21, ... DBX24.4). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_NCKrelatedM01" (DB21, ... DBX30.5): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	Activation of "Associated help function" is not requested.
Value TRUE	Activation of "Associated help function" is requested.
Corresponds to	• "LBP_Chan*.E_MMC_M01AssocNC" (DB21, ... DBX24.4) Associated help function selected • "LBP_Chan*.E_AssM01" (DB21, ... DBX318.5) Associated help function active • MD22254 $MC_AUXFU_ASSOC_M0_VALUE Additional M function for program stop • MD22256 $MC_AUXFU_ASSOC_M1_VALUE Additional M function for conditional stop
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC" > "Associated auxiliary functions"

A_JOG_Circles (DB21, ... DBX30.6)

Description	JOG circular travel As soon as the function is active (see "LBP_Chan*.E_JOG_Circles" (DB21, ... DBX377.6)), the machine operator can simultaneously traverse the two geometry axes of the active plane along an arc using the traversing keys or handwheel. **Application** This function is used for machine tools that are exclusively operated manually. **Note** The following preconditions applying for the "Circular travel in JOG" function: • The "Circular travel in JOG" function can be activated only in the JOG mode. The function cannot be activated if the machine functions JOG-REPOS and JOG-REF are active. • The axes that participate in traversing must be referenced. • The active plane must not be inclined.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The "Circular travel in JOG" function is not requested.

Value TRUE	The "Circular travel in JOG" function is requested.
Corresponds to	• "LBP_Chan*.E_JOG_Circles" (DB21, ... DBX377.6) JOG circular travel active
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_NoToolChangeCmd (DB21, ... DBX30.7)

Description	No tool change commands (Signal available for SINUMERIK ONE)
Signal flow	PLC → NC
Data type	BOOL

A_REPOSPM_0, A_REPOSPM_1, A_REPOSPM_2

(DB21, ... DBX31.0, 31.1, 31.2)

Description	REPOS mode Selecting REPOS mode:			
	A_REPOSPM_2 Bit 2	**A_REPOSPM_1 Bit 1**	**A_REPOSPM_0 Bit 0**	**REPOS mode**
	0	0	0	No REPOS mode active
	0	0	1	RMB: Repositioning to beginning of block
	0	1	0	RMI: Repositioning to interrupt point
	0	1	1	RME: Repositioning to end of block
	1	0	0	RMN: Repositioning to the nearest path point
Signal flow	PLC → NC			
Data type	BOOL			
Update	Cyclic			

Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgement • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

A_REPOSMode (DB21, ... DBX31.4)

Description	REPOS activation
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Signal state 0 → 1	Request to activate the "REPOS" function for the actual main run block. The following become active: • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay
Signal state 1 → 0	Reset of the last request.

Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgement • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	Function Manual Basic Functions; • Section "Continuous-path mode, exact stop, LookAhead" > "Smoothing and repositioning (REPOS)" • Section "Mode group, channel, program mode, reset response" > "Repositioning (REPOS)"

A_InvCHWDirOfRot (DB21, ... DBX31.5)

Description	Contour handwheel: Invert handwheel direction of rotation **Note** The interface signal must only be changed when the axis is stationary.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to invert the direction of rotation of the contour handwheel.
Value TRUE	A request has been made to invert the direction of rotation of the contour handwheel.
Corresponds to	• "LBP_Chan*.E_InvCHWDirOfRot" (DB21, ... DBX39.5) Contour handwheel: Handwheel direction of rotation inversion active
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_SKP8, **A_SKP9**

(DB21, ... DBX31.6, 31.7)

Description	Activate skip block
Signal flow	PLC → NC
Data type	BOOL

E_ExecExtern (DB21, ... DBX32.0)

Description	Execution from external source active
Signal flow	NC → PLC
Data type	BOOL

(DB21, ... DBX32.1)

Description	RESU: Retrace mode active The interface signal is set if (DB21, ... DBX0.1) == TRUE The interface signal is reset (DB21, ... DBX0.2) == TRUE
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Retrace mode is not active.
Value TRUE	Retrace mode is active.
Corresponds to	• (DB21, ... DBX0.1) RESU: Backward / forward • (DB21, ... DBX0.2) RESU: Start retrace support • (DB21, ... DBX32.1) RESU: Retrace mode active • (DB21, ... DBX32.2) RESU: Retrace support active
Additional references	Function Manual Technologies, Section "Continue machining (retrace support)

(DB21, ... DBX32.2)

Description	RESU: Retrace support active The interface signal is set when (DB21, ... DBX0.2) == TRUE is detected. The interface signal is reset if the last action block has been completed.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Retrace support is not active.
Value TRUE	Retrace support is active.
Corresponds to	• (DB21, ... DBX0.1) RESU: Backward / forward • (DB21, ... DBX0.2) RESU: Start retrace support • (DB21, ... DBX32.1) RESU: Retrace mode active • (DB21, ... DBX32.2) RESU: Retrace support active
Additional references	Function Manual Technologies, Section "Continue machining (retrace support)

E_ActBlock (DB21, ... DBX32.3)

Description	Action block active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No action block is active.
Value TRUE	An action block is active / is being executed.
Corresponds to	• "LBP_Chan*.E_LastActBlock" (DB21, ... DBX32.6) Last action block active • "LBP_Chan*.E_BlockSearch" (DB21, ... DBX33.4) Block search active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 1, 2, and 4"

E_BegBlock (DB21, ... DBX32.4)

Description	Approach block active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No approach block is active.
Value TRUE	The approach block to continue the NC program for block search, type 2 "Block search with calculation at the contour", is active.
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_M01 (DB21, ... DBX32.5)

Description	M00 / M01 active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Auxiliary function M00 or M01 is not active.
Value TRUE	Auxiliary function M00 or M01 is active. The block in which the auxiliary function is programmed, is executed and the help functions output. Program state "stopped".
Corresponds to	• "LBP_Chan*.A_M01" (DB21, ... DBX0.5) Activate M01 • "LBP_Chan*.E_MMC_M01" (DB21, ... DBX24.5) M01 selected
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_LastActBlock (DB21, ... DBX32.6)

Description	Last action block active If the last action block is active, this means that all the action blocks on the NC side have been processed and the actions on the PLC side (ASUB, FC) or the operator such as overstore, mode change according to JOG/REPOS are possible. This allows the PLC to perform another tool change, for example, before the start of the motion. Action blocks contain the actions collected during "Block search with computation" such as: • Help function outputs H, M00, M01, M.. • Tool programming T, D, DL • Spindle programming S value, M3/M4/M5/M19, SPOS • Feed programming, F
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The last action block is not active.
Value TRUE	The last action block is active / is being executed.
Corresponds to	• "LBP_Chan*.E_BlockSearch" (DB21, ... DBX33.4) Block search active
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_Ref (DB21, ... DBX33.0)

Description	Referencing active Feedback signal for the request to activate referencing: 1. "LBP_Chan*.A_Ref" (DB21, ... DBX1.0): 0 → 1 (request) 2. "LBP_Chan*.E_Ref" (DB21, ... DBX33.0): 0 → 1 (feedback signal)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	Referencing is active.
Edge change 1 → 0	No effect
Corresponds to	• "LBP_Chan*.A_Ref" (DB21, ... DBX1.0) Activate referencing
Additional references	Function Manual Axes and Spindles; Section "Referencing"

E_OrieToolholder (DB21, ... DBX33.1)

Description	Orientable tool carrier active
Signal flow	NC → PLC
Data type	BOOL

E_RevFD (DB21, ... DBX33.2)

Description	Revolutional feedrate active The interface signal indicates that, in AUTOMATIC mode, path or synchronized axes move at revolutional feedrate.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Revolutional feedrate (G95) is not active.
Value TRUE	Revolutional feedrate (G95) is active.
Additional references	• Function Manual Axes and Spindles; Section "Feedrates" > "Path feedrate F" > "Type of feed G93, G94, G95"

E_HWOverlay (DB21, ... DBX33.3)

Description	Handwheel override active The handwheel pulses of the 1st geometry axis function as a velocity override on the programmed path velocity. In the following cases, the override is not effective: • The path axes have reached the programmed target position. • The distance-to-go has been deleted. • Reset was initiated.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The function "Handwheel override in automatic mode" is not active for the programmed path axes.
Value TRUE	The "Handwheel override in automatic mode" function is active for the programmed path axes.
Corresponds to	• "LBP_Chan*.A_DeleteDTG" (DB21, ... DBX6.2) Delete distance-to-go
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_BlockSearch (DB21, ... DBX33.4)

Description	Block search active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The "block search" function is not active or the search target has been found.
Value TRUE	The "block search" function is active. It was selected at the user interface and started with interface signal "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) (NC-Start).
Corresponds to	• "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) NC start
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 1, 2, and 4" or "Block search type 5 (SERUPRO)"

E_M30 (DB21, ... DBX33.5)

Description	M02 / M30 active • In the last part program block of an NC program, it is not permissible to program the following functions: – Auxiliary functions, which initiate a read-in stop – Programmed spindle speed (S value) to be effective beyond the end of program (M02 / M30) • After a program reset has been completed (M02 / M30), the signal is set to value TRUE. While processing a subsequent PROG_EVENT program, the signal is set to FALSE, and after completion of the PROG_EVENT program, it is again set to TRUE. **Note** The signal is not suitable to automatically initiate follow-on functions such as tool counting, bar feed etc. In order to identify the completion of the previous processing (NC program, ASUB, PROG_EVENT, etc.) the end of program command (M02 / M30) must be written to its own dedicated part program block. Command (M02 / M30) or the decoded signal of the M function can be used as the trigger.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	End of program M02 / M30 is not active The interface signal is reset or remains reset: • after the start and while executing an NC program, asynchronous subprogram (ASUB) or PROG_EVENT program. **Note:** The last signal state is maintained after the start of a block search with calculation in "Program test" mode (SERUPRO). • After a program abort as the result of an alarm. • During and after executing a channel reset without executing a PROG_EVENT program. • During and after control system run-up without executing a PROG_EVENT program.
Value TRUE	End of program M02 / M30 is active. The interface signal is set after: • A program reset has been completed (M02, M30 or M17) **Note:** If traversing motion is programmed in the block of M02, M30 or M17, then the signal is only set after all of the axes have reached their target positions. • Executed the PROG_EVENT program (PROG_EVENT.SPF) after: – End of program reset (M02 / M30) – Channel reset – Warm restart (power on) – Output of the last action block after a block search
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Program mode" > "Event driven program call (PROG_EVENT)"

E_Transform (DB21, ... DBX33.6)

Description	Transformation active One of the commands `TRANSMIT`, `TRACYL`, `TRAANG`, or `TRAORI` has been programmed in the channel in the active NC program. The block in question has been processed and the corresponding transformation is active.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No transformation is active.
Value TRUE	A transformation is active
Corresponds to	• "LBP_Chan*.E_TransformNo" (DB21, ... DBB388) Index number of the active transformation in the channel
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" • Function Manual Technologies

E_ProgTest (DB21, ... DBX33.7)

Description	Program test (PRT) active When program test (PRT) is active, the machine axes do not move when a block or NC program is executed. However, the axis movements are indicated on the user interface by changing setpoint position values. **Note** While program test (PRT) is active, all traversing motions of the axes, but not the spindles, take place at "Axis disable". **Note** Program test (PRT) is selected at the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_ProgTest" (DB21, ... DBX25.7). Depending on the value of the "LBP_ConfigBP [FC1]" parameter `MMCToIf`, the interface signal is transferred from the PLC basic program into the NC/PLC interface signal "LBP_Chan*.A_ProgTest" (DB21, ... DBX1.7): • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE". **Notice** Because of the axis disable, the assignment of a tool magazine is not changed while the program is being tested. The user / machine manufacturer must utilize a suitable PLC user program to ensure that the NC-internal tool management and the actual assignment of the tool magazine remain consistent.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program test (PRT) is not active.
Value TRUE	Program test (PRT) is active.

Corresponds to	• "LBP_Chan*.A_ProgTest" (DB21, ... DBX1.7) Activate program test (PRT) • "LBP_Chan*.E_MMC_ProgTest" (DB21, ... DBX25.7) Program test (PRT) requested
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_OEMChan (DB21, ... DBB34)

Description	OEM signals: Feedback signal
Signal flow	NC → PLC
Data type	BYTE

E_ProgRunning (DB21, ... DBX35.0)

Description	Program state "running" The signal is set after the NC program starts with "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) = TRUE. If execution of the NC program is stopped with "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) = TRUE, the signal remains set. The signal is not reset if NC program execution is stopped as a result of the following events: • A feed disable or spindle disable is active • "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) (read-in disable) • Feedrate override: 0 % • A spindle and axis monitoring function responds • Position setpoints are entered in the NC program for axes in the "follow-up mode," for axes without "controller enable," or for "parking axes" Program states: • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program state ≠ "running"
Value TRUE	Program state == "running"

Corresponds to	• "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) = TRUE Read-in disable • "LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) = TRUE NC start • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ProgWait (DB21, ... DBX35.1)

Description	Program state "wait" The program state "wait" is entered if `WAIT_M` or `WAIT_E` causes the NC program to wait for synchronization with an NC program from another channel. Program states: • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program state ≠ "wait"
Value TRUE	Program state == "wait"

Corresponds to	• "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ProgStop (DB21, ... DBX35.2)

Description	Program state "stopped" Events, which lead to the NC program being stopped: • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.A_NCStopASp" (DB21, ... DBX7.4) NC stop axes plus spindles • "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) NC stop at block limit • Programmed command `M00` or `M01` • Single block mode Program states: • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program state ≠ "stopped"
Value TRUE	Program state == "stopped"

Corresponds to	• "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) NC stop at block limit • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.A_NCStopASp" (DB21, ... DBX7.4) NC stop axes plus spindles • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ProgInterrupt (DB21, ... DBX35.3)

Description	Program state "interrupted" The signal indicates that the interrupted NC program can be continued with NC start ("LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) = TRUE). The "interrupted" state is entered, for example, if the program changes from AUTOMATIC or MDI mode to JOG mode in the "stopped" state ("LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) == TRUE). After a subsequent operating mode change from JOG to AUTOMATIC or MDI, the NC program can continue from the interruption point with NC start ("LBP_Chan*.A_NCStart" (DB21, ... DBX7.1) = TRUE). Program states: • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program state ≠ "interrupted"
Value TRUE	Program state == "interrupted"

Corresponds to	• "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ProgAborted (DB21, ... DBX35.4)

Description	Program state "aborted" The signal is set in the following cases: • The actual NC program was selected in the channel but has not been started. • The actual NC program was executed in the channel and aborted with "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) = TRUE (channel reset). Program states: • "LBP_Chan*.E_ProgAborted" (DB21, ... DBX35.4) Program state "canceled" • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Program state ≠ "aborted"
Value TRUE	Program state == "aborted"

Corresponds to	• "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) reset • "LBP_Chan*.E_ProgInterrupt" (DB21, ... DBX35.3) Program state "interrupted" • "LBP_Chan*.E_ProgStop" (DB21, ... DBX35.2) Program state "stopped" • "LBP_Chan*.E_ProgWait" (DB21, ... DBX35.1) Program state "wait" • "LBP_Chan*.E_ProgRunning" (DB21, ... DBX35.0) Program state "running"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ChanActive (DB21, ... DBX35.5)

Description	Channel state "active" The signal is set in the following cases: • An NC program is being executed in the Automatic or MDI mode. • At least one axis is being traversed in the JOG mode. Channel states: • "LBP_Chan*.E_ChanActive" (DB21, ... DBX35.5) Channel status "active" • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state "interrupted" • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Channel state ≠ "active"
Value TRUE	Channel state == "active"
Corresponds to	• "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state "interrupted" • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ChanInterrupt (DB21, ... DBX35.6)

Description	Channel state "interrupted" An interruption can be initiated during execution of an NC program in AUTOMATIC or MDI mode or in JOG mode while an axis is traversing as a result of the following events: • "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.A_NCStopASp" (DB21, ... DBX7.4) NC stop axes plus spindles • "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) NC stop at block limit • Programmed stop `M00` or `M01` • Single block mode Channel states: • "LBP_Chan*.E_ChanActive" (DB21, ... DBX35.5) Channel status "active" • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state "interrupted" • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Channel state ≠ "interrupted"
Value TRUE	Channel state == "interrupted"
Corresponds to	• "LBP_Chan*.A_NCStop" (DB21, ... DBX7.3) NC stop • "LBP_Chan*.A_NCStopASp" (DB21, ... DBX7.4) NC stop axes plus spindles • "LBP_Chan*.A_NCStopBlock" (DB21, ... DBX7.2) NC stop at block limit • "LBP_Chan*.E_ChanActive" (DB21, ... DBX35.5) Channel status "active" • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ChanReset (DB21, ... DBX35.7)

Description	Channel state "reset" The signal is set in the following cases: • End of program reset (M02 / M30) • Channel reset • Warm restart (power on) Channel states: • "LBP_Chan*.E_ChanActive" (DB21, ... DBX35.5) Channel status "active" • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state "interrupted" • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Channel state ≠ "reset"
Value TRUE	Channel state == "reset"
Corresponds to	• "LBP_Chan*.E_ChanActive" (DB21, ... DBX35.5) Channel status "active" • "LBP_Chan*.E_ChanInterrupt" (DB21, ... DBX35.6) Channel state "interrupted"
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_StartReq (DB21, ... DBX36.0)

Description	Start request
Signal flow	NC → PLC
Data type	BOOL

E_StopReq (DB21, ... DBX36.1)

Description	Stop request
Signal flow	NC → PLC
Data type	BOOL

E_AxesRef (DB21, ... DBX36.2)

Description	All axes that have to be referenced are referenced The spindles of the channel have no effect on this interface signal.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	One or more axes of the channel that have to be referenced are not referenced.

Value TRUE	All axes that must be referenced (linear axes and rotary axes) of the channel are referenced. The machine data: MD20700 $MC_REFP_NC_START_LOCK (NC start disable without reference point) is zero. If two position measuring systems are connected to an axis, that would prevent an NC start, then the active one must be referenced so that the axis is considered to have been referenced. An NC start command for the part program processing is only accepted when this signal is present. Axes must be referenced if: MD34110 $MA_REFP_CYCLE_NR _ = -1 and the axis is not in the parked position (position measuring system inactive and controller enable canceled).
Corresponds to	• "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) Referenced/synchronized 1 • "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) Referenced/synchronized 2
Additional references	Function Manual Axes and Spindles; Section "Referencing"

E_AxesStop (DB21, ... DBX36.3)

Description	All axes stationary Criterion for "axis stationary": Interpolator end
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Not all channel axes are stationary - or other traversing motion is presently active.
Value TRUE	All channel axes are stationary - and no other traversing motion is active.
Additional references	Function Manual Basic Functions; Section "Continuous-path mode, exact stop, look ahead" > "Exact-stop mode"

E_IR_Activ (DB21, ... DBX36.4)

Description	Interrupt handling active **Note** The interface signal is not set, if an interrupt is handled in a program mode (AUTOMATIC or MDI).
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Interrupt handling is not active. All channels are operating in the requested mode.
Value TRUE	Interrupt handling is active. One or more channels in the mode group are not in the desired operating mode as the result of an active interrupt routine.
Corresponds to	• MD11600 $MN_BAG_MASK Definition of the mode group response
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_ChanRO (DB21, ... DBX36.5)

Description	Channel is ready The channel is ready to execute an NC program and to traverse axes and spindle that are assigned to the channel.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The channel is not ready.
Value TRUE	The channel is ready.
Corresponds to	• MD11600 $MN_BAG_MASK Definition of the mode group response
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_NCKalarmChan (DB21, ... DBX36.6)

Description	Channel-specific NC alarm active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No NC alarm is active in the channel.
Value TRUE	At least one NC alarm is active in the channel.
Corresponds to	• "LBP_Chan*.E_NCKalarmStop" (DB21 ... DBX36.7) NC alarm with machining stop active • "LBP_NC.E_NCKalarm" (DB10 ... DBX109.0) NC alarm is active
Additional references	Diagnostics guide

E_NCKalarmStop (DB21, ... DBX36.7)

Description	NC alarm with machining stop is active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No NC alarm with machining stop is active in the channel.
Value TRUE	At least one NC alarm with machining stop is active in the channel.
Corresponds to	• "LBP_Chan*.E_NCKalarmChan" (DB21, ... DBX36.6) Channel-specific NC alarm is active • "LBP_NC.E_NCKalarm" (DB10 ... DBX109.0) NC alarm is active
Additional references	Diagnostics guide

E_CHW1, **E_CHW2**, **E_CHW3**

(DB21, ... DBX37.0, 37.1, 37.2)

<table>
<tr><td>Description</td><td>Contour handwheel active

The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.
Bit coded: maximum 3 handwheels
<table>
<tr><th>E_CHW3
Bit 2</th><th>E_CHW2
Bit 1</th><th>E_CHW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
</table>
Bit coded: maximum 6 handwheels
<table>
<tr><th>E_CHW3
Bit 2</th><th>E_CHW2
Bit 1</th><th>E_CHW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>NC → PLC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Chan*.A_ContHW*" (DB21, ... DBX30.0 ... 30.2)
Activate contour handwheel
• MD11324 $MN_HANDWH_VDI_REPRESENTATION
Representation of the handwheel number at the VDI interface</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Manual traversing"</td></tr>
</table>

E_CLC (DB21, ... DBX37.3)

Description	Clearance control (CLC) active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Clearance control (CLC) is not active.
Value TRUE	Clearance control (CLC) is active.

Corresponds to	• "LBP_Chan*.A_CLC_Stop" (DB21, ... DBX1.4) Clearance control (CLC): Stop • "LBP_Chan*.A_CLC_OR" (DB21, ... DBX1.5) Clearance control (CLC): Override • "LBP_Chan*.E_CLCStopLL" (DB21, ... DBX37.4) Clearance control (CLC): stop at the lower motion limit • "LBP_Chan*.E_CLCStopUL" (DB21, ... DBX37.5) Clearance control (CLC): stop at the upper motion limit • "LBP_Chan*.E_CLCStop*" (DB21, ... DBX37.4 ... 37.5) Clearance control (CLC): motion has stopped
Additional references	Function Manual Monitoring and Compensating, Section: "Clearance control"

E_CLCStopLL (DB21, ... DBX37.4)

Description	Clearance control (CLC): stop at the lower motion limit The lower motion limit of the clearance control is set in machine data: MD62505 $MC_CLC_SENSOR_LOWER_LIMIT The parameterized limit value can be adapted on a block-specific basis by programming `CLC_LIM(...)` in the part program. **Note** If, in addition to "LBP_Chan*.E_CLCStopLL" (DB21, ... DBX37.4), "LBP_Chan*.E_CLCStopUL" (DB21, ... DBX37.5) are also simultaneously set, the signal is active as follows: "Clearance control (CLC): motion has stopped".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The traversing motion of the clearance-controlled axis, generated by the clearance control, has not reached the lower motion limit.
Value TRUE	The traversing motion of the clearance-controlled axis, generated by the clearance control, has reached the lower motion limit and was stopped.
Corresponds to	• `CLC_LIM(...)`; the control range is limited • "LBP_Chan*.E_CLCStopUL" (DB21, ... DBX37.5) Clearance control (CLC): stop at the upper motion limit • "LBP_Chan*.E_CLCStop*" (DB21, ... DBX37.4 ... 37.5) Clearance control (CLC): motion has stopped • MD62505 $MC_CLC_SENSOR_LOWER_LIMIT Lower clearance control motion limit
Additional references	Function Manual Monitoring and Compensating, Section: "Clearance control"

E_CLCStopUL (DB21, ... DBX37.5)

Description	Clearance control (CLC): stop at the upper motion limit The upper motion limit of the clearance control is set in machine data: MD62506 $MC_CLC_SENSOR_UPPER_LIMIT The parameterized limit value can be adapted on a block-specific basis by programming `CLC_LIM(...)` in the part program. **Note** If, in addition to "LBP_Chan*.E_CLCStopLL" (DB21, ... DBX37.4), "LBP_Chan*.E_CLCStopUL" (DB21, ... DBX37.5) are also simultaneously set, the signal is active as follows: "Clearance control (CLC): motion has stopped".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The traversing motion of the clearance-controlled axis, generated by the clearance control, has not reached the upper motion limit.
Value TRUE	The traversing motion of the clearance-controlled axis, generated by the clearance control, has reached the upper motion limit and was stopped.
Corresponds to	• `CLC_LIM(...)`; the control range is limited • "LBP_Chan*.E_CLCStopLL" (DB21, ... DBX37.4) Clearance control (CLC): stop at the lower motion limit • "LBP_Chan*.E_CLCStop*" (DB21, ... DBX37.4 ... 37.5) Clearance control (CLC): motion has stopped • MD62506 $MC_CLC_SENSOR_UPPER_LIMIT Upper clearance control motion limit
Additional references	Function Manual Monitoring and Compensating, Section: "Clearance control"

E_RIEnabIgnored (DB21, ... DBX37.6)

Description	Read-in disable is ignored The read-in disable "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) is ignored, if the following applies: "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) == TRUE AND actual block has the status "read-in disable inactive" The read-in disable "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) is not ignored, if the following applies: "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) == FALSE OR ("LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) == TRUE AND actual block has the status "read-in disable active") Blocks for which read-in disable is ignored have the status "read-in disable inactive". The following machine data is used to specify that the read-in disable "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) is to be ignored: • MD11602 $MN_ASUP_START_MASK, bit 2 • MD20116 $MC_IGNORE_INHIBIT_ASUP • MD20107 $MC_ PROG_EVENT_IGN_INHIBIT
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Read-in disable "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) is not ignored.
Value TRUE	Read-in disable "LBP_Chan*.A_RIdisable" (DB21, ... DBX6.1) is ignored.
Corresponds to	• "LBP_Chan*.E_StopBlkEndSBLsuppr" (DB21, ... DBX37.7) Stop at block end is ignored during single block (SBL) • MD11602 $MN_ASUP_START_MASK, Bit 2 Start also permitted if read-in disable is active • MD20116 $MC_IGNORE_INHIBIT_ASUP Execute interrupt program despite read-in disable • MD20107 $MC_ PROG_EVENT_IGN_INHIBIT Prog events ignore read-in disable
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_StopBlkEndSBLsuppr (DB21, ... DBX37.7)

Description	Stop at block end is ignored during single block (SBL) Stop at block end is ignored during single block (SBL), if the following applies: "LBP_Chan*.A_SBL" (DB21, ... DBX0.4) == TRUE (single block) AND actual block has the status "single block inactive" Stop at block end is not ignored during single block (SBL), if the following applies: "LBP_Chan*.A_SBL" (DB21, ... DBX0.4) == FALSE (single block) OR ("LBP_Chan*.A_SBL" (DB21, ... DBX0.4) == TRUE (single block) AND actual block has the status "single block active") The following machine data and commands are used to specify that the stop at block end during single block "LBP_Chan*.A_SBL" (DB21, ... DBX0.4) is to be ignored: • MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (prevent single-block stop) • MD20117 $MC_IGNORE_SINGLEBLOCK_ASUB (Execute interrupt program completely despite single block) • MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK (prog events ignore single block) • `SBLOF` (suppress single block), `SBLON` (neutral single block suppression) Blocks for which stop at block end during single block is ignored are designated as "single block inactive".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Stop at block end is not ignored during single block (SBL)
Value TRUE	Stop at block end is ignored during single block (SBL).
Corresponds to	• "LBP_Chan*.A_SBL" (DB21, ... DBX0.4) Single block • "LBP_Chan*.E_RIEnabIgnored" (DB21, ... DBX37.6) Read-in disable is ignored
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_StrokeEA (DB21, ... DBX38.0)

Description	Stroke initiation active
Signal flow	NC → PLC
Data type	BOOL
Value FALSE	Stroke initiation is not active.
Value TRUE	Stroke initiation is active.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable
Additional references	Function Manual Technologies

E_AcknManStrokEnab (DB21, ... DBX38.1)

Description	Manual stroke initiation: Acknowledgement
Signal flow	NC → PLC
Data type	BOOL
Value FALSE	Manual stroke was not initiated.
Value TRUE	Manual stroke has been initiated.
Corresponds to	• "LBP_Chan*.A_StrokEnab" (DB21, ... DBX3.0) Stroke enable • "LBP_Chan*.A_ManStrokEnab" (DB21, ... DBX3.1) Manual stroke initiation • "LBP_Chan*.A_ManRelStroke2" (DB21, ... DBX3.5) Manual stroke initiation 2 • "LBP_Chan*.E_StrokeEA" (DB21, ... DBX38.0) Stroke initiation active
Additional references	Function Manual Technologies

E_ProtZoneNOK (DB21, ... DBX39.0)

Description	Protection zone monitoring not guaranteed
Signal flow	NC → PLC
Data type	BOOL

(DB21, ... DBX39.1)

Description	NC alarm with program stop
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No NC alarm with program stop is active in the channel.
Value TRUE	The interface signal signals when progress of the program is blocked by an active alarm. This applies to all alarms that are generated with a stop reaction: • Stop on the path with resetting of the ready signal • Axes are braked • Immediate stop on the path • Interpreter is stopped, content of IPO buffer is still being processed • Stop at the end of the block In addition, alarms are considered that have the following alarm reaction: • NC start disable in this channel • NC start disable in this channel, ASUB starts, where relevant, are permitted For alarms with this reaction, the interface signal is set only when the alarm has not yet been acknowledged and an NC start has been triggered. In this case, the interface signal is set only on NC start. The interface signal is set when the start of an ASUB is denied in this channel because of an NC start disable. The interface signal is reset as soon as no alarm reactions that have activated the signal are active. This depends on the reset conditions of the alarms.

Corresponds to	• "LBP_NC.E_NCKalarm" (DB10 ... DBX109.0) NC alarm is active • "LBP_Chan*.E_NCKalarmChan" (DB21, ... DBX36.6) Channel-specific NC alarm active • "LBP_Chan*.E_NCKalarmStop" (DB21 ... DBX36.7) NC alarm with machining stop active
Additional references	Diagnostics Manual

E_SblStopAtBlockEnd (DB21, ... DBX39.4)

Description	Stop at end of block due to single block
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Stop at end of block due to single block is not active.
Value TRUE	Stop at end of block due to single block is active.

E_InvCHWDirOfRot (DB21, ... DBX39.5)

Description	Contour handwheel: Handwheel direction of rotation inversion active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Inversion of the direction of rotation of the contour handwheel is not active.
Value TRUE	Inversion of the direction of rotation of the contour handwheel is active.
Corresponds to	• "LBP_Chan*.A_InvCHWDirOfRot" (DB21, ... DBX31.5) Contour handwheel: Invert handwheel direction of rotation
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_CfgStopFuncActive (DB21, ... DBX39.6)

Description	Configured stop is activated
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The "Configured stop" function was not activated internally.
Value TRUE	The "Configured stop" function was activated internally.
Corresponds to	• "LBP_Chan*.A_CfgStopFunc" (DB21, ... DBX7.6) Activate configured stop

E_CfgStopAtBlockEnd (DB21, ... DBX39.7)

Description	Stop at end of block due to configured stop
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No stop at end of block due to configured stop

Value TRUE	Stop at end of block due to configured stop
Corresponds to	• "LBP_Chan*.A_CfgStopFunc" (DB21, ... DBX7.6) Activate configured stop • "LBP_Chan*.E_CfgStopFuncActive" (DB21, ... DBX39.6) Configured stop is activated

10.4.9.5 LBP_Chan1 [DB21], ... - Control signals from geometry axes

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_Geo (DB21, ... from DBX40.0)

Description	Control signals from geometry axes
Data type	ARRAY[1..3] of STRUCT STRUCT "E_Geo", control signals of the geometry axis (Page 359)

10.4.9.6 LBP_Chan1 [DB21], ... - Change signals for auxiliary function transfer from the channel

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

M1Change (DB21, ... DBX58.0)

M2Change (DB21, ... DBX58.1)

M3Change (DB21, ... DBX58.2)

M4Change (DB21, ... DBX58.3)

M5Change (DB21, ... DBX58.4)

S1Change (DB21, ... DBX60.0)

S2Change (DB21, ... DBX60.1)

S3Change (DB21, ... DBX60.2)

T1Change (DB21, ... DBX61.0)

T2Change (DB21, ... DBX61.1)

T3Change (DB21, ... DBX61.2)

D1Change (DB21, ... DBX62.0)

D2Change (DB21, ... DBX62.1)

D3Change (DB21, ... DBX62.2)

EChange (DB21, ... DBX63.0)

H1Change (DB21, ... DBX64.0)

H2Change (DB21, ... DBX64.1)

H3Change (DB21, ... DBX64.2)

F1Change (DB21, ... DBX65.0)

F2Change (DB21, ... DBX65.1)

F3Change (DB21, ... DBX65.2)

F4Change (DB21, ... DBX65.3)

F5Change (DB21, ... DBX65.4)

F6Change (DB21, ... DBX65.5)

Description	M, S, T, D, H, F fct. Change
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	The change signals are reset by the PLC basic program at the start of the next "Main [OB1]" cycle. The value of the respective data is not valid.
Value TRUE	M, S, T, D, H or F information has been output to the interface with a new value together with the associated change signal at the beginning of a "Main [OB1]" cycle. In this case, the change signal indicates that the appropriate value is valid.
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

M1NDec (DB21, ... DBX59.0)

M2NDec (DB21, ... DBX59.1)

M3NDec (DB21, ... DBX59.2)

M4NDec (DB21, ... DBX59.3)

M5NDec (DB21, ... DBX59.4)

Description	M fct. 1-5 not decoded
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	M function less than 99 (for extended address = 0) or for extended address > 0 included in the decoding list.
Value TRUE	M function is greater than 99 (for extended address = 0) or for extended address > 0, not included in the decoding list. This signal is available - together with the associated M change signal - for one "Main [OB1]" cycle. Cause: • Incorrect M function programmed • M function not configured in the decoding list of the PLC Remedy e.g.: • PLC sets read-in inhibit • Output of a PLC alarm

S1Quick (DB21, ... DBX60.4)

S2Quick (DB21, ... DBX60.5)

S3Quick (DB21, ... DBX60.6)

T1Quick (DB21, ... DBX61.4)

T2Quick (DB21, ... DBX61.5)

T3Quick (DB21, ... DBX61.6)

D1Quick (DB21, ... DBX62.4)

D2Quick (DB21, ... DBX62.5)

D3Quick (DB21, ... DBX62.6)

EQuick (DB21, ... DBX63.4)

H1Quick (DB21, ... DBX64.4)

H2Quick (DB21, ... DBX64.5)

H3Quick (DB21, ... DBX64.6)

M1Quick (DB21, ... DBX66.0)

M2Quick (DB21, ... DBX66.1)

M3Quick (DB21, ... DBX66.2)

M4Quick (DB21, ... DBX66.3)

M5Quick (DB21, ... DBX66.4)

F1Quick (DB21, ... DBX67.0)

F2Quick (DB21, ... DBX67.1)

F3Quick (DB21, ... DBX67.2)

F4Quick (DB21, ... DBX67.3)

F5Quick (DB21, ... DBX67.4)

F6Quick (DB21, ... DBX67.5)

Description	M, S, T, D, H, F fct. Additional quick information (fast acknowledgment)
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	The change signals are reset by the PLC basic program at the start of the next "Main [OB1]" cycle. The value of the respective data is not valid.
Value TRUE	M, S, T, D, H or F information has been output to the interface with a new value together with the associated change signal at the beginning of a "Main [OB1]" cycle. In this case, the additional info "Quick" indicates the quick help function.

Note

- For 10-decade T numbers, only the signal DBB61, DBX0 "T fct.1 change" is available.
- For 5-decade D numbers, only the signal DBB62, DBX0 "D fct.1 change" is available.

10.4.9.7 LBP_Chan1 [DB21], ... - Transferred M and S functions

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

ExtM1 (DB21, ... DBW68.0)

M1 (DB21, ... DBD70.0)

ExtM2 (DB21, ... DBW74.0)

M2 (DB21, ... DBD76.0)

ExtM3 (DB21, ... DBW80.0)

M3 (DB21, ... DBD82.0)

ExtM4 (DB21, ... DBW86.0)

M4 (DB21, ... DBD88.0)

ExtM5 (DB21, ... DBW92.0)

M5 (DB21, ... DBD94.0)

Description	M function 1 - 5 and extended address M function 1 - 5 If an M function change signal ("LBP_Chan*.M*Change" (DB21, ... DBB58)) is present, then the up to five M functions - that can be programmed in an NC block - together with the number of the M function and the extended address are available here. The M function values remain until they are overwritten by new M functions. The M function values are deleted by the following events: • The PLC starts (system run-up). • A new M function is entered.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Corresponds to	• "LBP_Chan*.M*Change" (DB21, ... DBX58.0 ... 58.4) Change signals: M function • MD10715 $MN_M_NO_FCT_CYCLE • MD10716 $MN_M_NO_FCT_CYCLE_NAME • MD10718 $MN_M_NO_FCT_CYCLE_PAR • $C_M... System variable
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

ExtS1 (DB21, ... DBW98.0)

S1 (DB21, ... DBD100.0)

ExtS2 (DB21, ... DBW104.0)

S2 (DB21, ... DBD106.0)

ExtS3 (DB21, ... DBW110.0)

S3 (DB21, ... DBD112.0)

Description	S function 1 - 3 and extended address S function 1 - 3 If an S function change signal ("LBP_Chan*.S*Change" (DB21, ... DBB60)) is present, then the up to three S functions - that can be programmed in an NC block - together with the number of the S function and the extended address are available here. The S function values remain until they are overwritten by new S functions. The S function values are deleted by the following events: • The PLC starts (system run-up). • A new S function is entered.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Corresponds to	• "LBP_Chan*.S*Change" (DB21, ... DBX60.0, 60.1, 60.2) Change signals: S function
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

Note

M functions are programmed in the part program in the INTEGER format (8 decades plus leading sign).

10.4.9.8 LBP_Chan1 [DB21], ... - Transferred T/D/DL functions

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

ExtT1 (DB21, ... DBW116)

ExtT2 (DB21, ... DBW120)

ExtT3 (DB21, ... DBW124)

Description	Extended address T functions T function 1: "LBP_Chan*.ExtT1" (DB21, ... DBW116) T function 2: "LBP_Chan*.ExtT2" (DB21, ... DBW120) T function 3: "LBP_Chan*.ExtT3" (DB21, ... DBW124)
Signal flow	NC → PLC
Data type	INT

T1, T2, T3

(DB21, ... DBW118, 122, 126)

Description	T functions T function 1: "LBP_Chan*.T1" (DB21, ... DBW118) T function 2: "LBP_Chan*.T2" (DB21, ... DBW122) T function 3: "LBP_Chan*.T3" (DB21, ... DBW126) If a T function change signal is present, then number of the T functions, programmed in the NC block, is made available here. The T function values remain until they are overwritten by new T functions. The T function values are deleted by the following events: • The PLC starts (system run-up). • A new T function is entered.
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Corresponds to	• "LBP_Chan*.T*Change" (DB21, ... DBX61.0, 61.1, 61.2) T function change signal • MD10717 $MN_T_NO_FCT_CYCLE_NAME Name of the tool change cycle for T function replacement • MD10719 $MN_T_NO_FCT_CYCLE_MODE Parameterization of the T function replacement • MD22220 $MC_AUXFU_T_SYNC_TYPE Output time of the T functions • MD22550 $MC_TOOL_CHANGE_MODE New tool compensation for M function • $C_T... System variable
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

ExtD1 (DB21, ... DBB128)

ExtD2 (DB21, ... DBB130)

ExtD3 (DB21, ... DBB132)

Description	Extended address D functions D function 1: "LBP_Chan*.ExtD1" (DB21, ... DBB128) D function 2: "LBP_Chan*.ExtD2" (DB21, ... DBB130) D function 3: "LBP_Chan*.ExtD3" (DB21, ... DBB132)
Signal flow	NC → PLC
Data type	BYTE

D1 (DB21, ... DBB129)

D2 (DB21, ... DBB131)

D3 (DB21, ... DBB133)

Description	D functions D function 1: "LBP_Chan*.D1" (DB21, ... DBB129) D function 2: "LBP_Chan*.D2" (DB21, ... DBB131) D function 3: "LBP_Chan*.D3" (DB21, ... DBB133) If a D function change signal is present, then number of the D functions, programmed in the NC block, is made available here. The D function values remain until they are overwritten by new D functions. The D function values are deleted by the following events: • The PLC starts (system run-up). • A new D function is entered.
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Corresponds to	• "LBP_Chan*.D*Change" (DB21, ... DBX62.0, 62.1, 62.2) D function change signal • MD22250 $MC_AUXFU_D_SYNC_TYPE Output time D functions • MD22252 $MC_AUXFU_DL_SYNC_TYPE Output time DL functions
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

ExtE (DB21, ... DBW134)

Description	Extended address DL functions
Signal flow	NC → PLC
Data type	INT

I (DB21, ... DBD136)

Description	DL function
Signal flow	NC → PLC
Data type	REAL

Note

- Programmed T functions are not output to the PLC when tool management is active.
- 8 decade T numbers are only available at DBD118 "T function 1".
- Programmed D functions with names (e.g. D=CUTTING EDGE_1) cannot be output to the PLC in ASCII format.
- 5-decade D numbers are only available as DBW130 "D function 1".
- The REAL format corresponds to the floating-point representation in STEP 7 (24-bit mantissa and 8-bit exponent). This floating point format supplies a maximum of 7 valid places.

10.4.9.9 LBP_Chan1 [DB21], ... - Transferred H/F functions

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

ExtH1 (DB21, ... DBW140)

H1 (DB21, ... DBD142)

ExtH2 (DB21, ... DBW146)

H2 (DB21, ... DBD148)

ExtH3 (DB21, ... DBW152)

H3 (DB21, ... DBD154)

Description	H function 1 - 3 and extended address H function 1-3 If an H function change signal is present, then the up to three H functions - that can be programmed in an NC block - together with the number of the H function and the extended address are available here. The H function values remain until they are overwritten by new H functions. The H function values are deleted by the following events: • The PLC starts (system run-up). • A new H function is entered.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Corresponds to	• "LBP_Chan*.H*Change" (DB21, ... DBX64.0 ... 64.2) Change signals: H function • DB21, ... DBD396 - 404 H function 1–3 (DINT) • MD22110 $MC_AUXFU_H_TYPE_INT Type of H-auxiliary function is an integer • MD22230 $MC_AUXFU_H_SYNC_TYPE Output time of the H functions
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

ExtF1 (DB21, ... DBW158)

F1 (DB21, ... DBD160)

ExtF2 (DB21, ... DBW164)

F2 (DB21, ... DBD166)

ExtF3 (DB21, ... DBW170)

F3 (DB21, ... DBD172)

ExtF4 (DB21, ... DBW176)

F4 (DB21, ... DBD178)

ExtF5 (DB21, ... DBW182)

F5 (DB21, ... DBD184)

ExtF6 (DB21, ... DBW188)

F6 (DB21, ... DBD190)

Description	F function 1 - 6 and extended address F function 1 - 6 If an F function change signal is present, then the up to six F functions - that can be programmed in an NC block - together with the number of the F function and the extended address are available here. The extended address of the F function is generated from the feedrate type (path feed or axis-specific feed) and the axis names: • Path feedrate: 0 • Axis-specific feedrate: Machine axis number of positioning axis 1, 2, 3, ... maximum number of machine axes The F function values remain until they are overwritten by new F functions. The F function values are deleted by the following events: • The PLC starts (system run-up). • A new F function is entered.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Corresponds to	• "LBP_Chan*.F*Change" (DB21, ... DBX65.0 ... 65.5) Change signals: F function • MD22240 $MC_AUXFU_F_SYNC_TYPE Output time of F functions
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

Note

- F functions are programmed in the part program in the REAL data format.
- The extended address of the F function contains an identifier with the following meaning:
  - 0: Path feedrate
  - 1 - 31: Machine axis number for feedrate for positioning axes
- The data type of the H function depends on machine data: MD22110 $MC_AUXFU_H_TYPE_INT

10.4.9.10 LBP_Chan1 [DB21], ... - Decoded M signals

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

MDyn (DB21, ... from DBX194.0)

Description	Dynamic M functions M0 to M99
Signal flow	NC → PLC
Data type	ARRAY[0..99] of BOOL
Update	Job-controlled
Value FALSE	The associated M function has not been decoded or the interface signal has been reset from the PLC basic program as acknowledgment. For a general auxiliary function output, the interface signal is reset by the PLC basic program after "Main [OB1]" has been completely executed. For a fast auxiliary function output, the interface signal is reset by the PLC basic program in the same "Hardware interrupt [OB40]" cycle.
Value TRUE	The associated M function was has been decoded.
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

Note

- #: The M function is not displayed here, if a spindle is parameterized in the channel. In this case, the M function is displayed as an extended M function under DB21,... DBB68 ff. and axially under DB31,... DBB86 ff.
- **Dynamic** M functions (M00 to M99) are decoded by the PLC basic program.
  Static M functions must be generated in the PLC user program from dynamic M functions.

10.4.9.11 LBP_Chan1 [DB21], ... - Active G functions

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

GAct (DB21, ... ab DBB208)

<table>
<tr><td>Description</td><td>Active G command of group 1 to 60

The G command displayed in the BCD format or its mnemonic identifier is active in the specified G group.
<table>
<tr><th>Variable</th><th>Meaning</th></tr>
<tr><td>GAct.GAct[1]
DBB208</td><td>G group 1: Internal number of the active G command</td></tr>
<tr><td>GAct.GAct[2]
DBB209</td><td>G group 2: Internal number of the active G command</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>GAct.GAct[64]
DBB271</td><td>G group 64: Internal number of the active G command</td></tr>
</table>
Application example
G group 14 ≙ DBB208 + 14 - 1 = DBB221,
Active G commands: DBB221 == 1 ≙ internal number for G command G90
<table>
<tr><th>Bit</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><th>Value</th><th>128</th><th>64</th><th>32</th><th>16</th><th>8</th><th>4</th><th>2</th><th>1</th></tr>
<tr><th>DBB221</th><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
Special case
For a value of 0, no G command or mnemonic identifier of the G group is active.
Basic setting after POWER ON
After Power On, the value zero, i.e. active G group undefined, is transferred for all G groups.
Part program end or abort
After part program end or abort, the last active G command is retained.
NC START
After NC START the values of the eight G groups specified in the following machine data are overwritten in accordance with the basic setting set in the machine data as well as the values programmed in the part program:
MD22510 $MC_GCODE_GROUPS_TO_PLC
Note
In contrast to auxiliary functions, G commands are not output to the PLC subject to acknowledgment, i.e. processing of the part program is continued immediately after the G command output.</td></tr>
<tr><td>Signal flow</td><td>NC → PLC</td></tr>
<tr><td>Data type</td><td>ARRAY[1..64] of BYTE</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Additional references</td><td>• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"
• A complete list of the G groups, G commands and their internal numbers can be found in: Programming Manual Fundamentals; Section "Tables" > "G commands"</td></tr>
</table>

Note

- The active G functions of the group, for each programming of a G function or a mnemonic identifier (e.g. SPLINE), are updated.
- G functions within a G group are output as binary value, starting with 1. A G function with the value 0 means that for this G group, no G function is active.

10.4.9.12 LBP_Chan1 [DB21], ... Protection areas from channel

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_MRPactivated (DB21, ... from DBX272.0)

Description	Machine-related protection area 1 to 10 preactivated **Note** Only a preactivated machine-related protection area can be activated or deactivated using the corresponding interface signal: "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) (activate machine-related protection area 1 to 10)
Signal flow	NC → PLC
Data type	ARRAY[1..10] of BOOL
Update	Cyclic
Value FALSE	The machine-related protection area 1 (...10) is not preactivated.
Value TRUE	The machine-related protection area 1 (...10) is preactivated. Preactivation is realized in the NC program using command `NPROT`.
Corresponds to	• "LBP_Chan*.A_ProtZone" (DB21, ... DBX1.1) Enable protection areas • "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) Activate machine-related protection area 1 to 10 • "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) activate channel-specific protection area 1 to 10 • "LBP_Chan*.E_CSPactivated" (DB21, ... DBX274.0 ... 275.1) Channel-specific protection area 1 to 10 pre-activated • "LBP_Chan*.E_MRPViolated" (DB21, ... DBX276.0 ... 277.1) Machine-related protection area 1 to 10 violated • "LBP_Chan*.E_CSPViolated" (DB21, ... DBX278.0 ... 279.1) Channel-specific protection area 1 to 10 violated
Additional references	Function Manual Basic Functions; Section "Monitoring and compensating" > "Protection areas"

E_CSPactivated (DB21, ... ab DBX274.0)

Description	Channel-related protection areas 1 to 10 preactivated **Note** Only a preactivated channel-specific protection area can be activated or deactivated using the corresponding interface signal: "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) (activate channel-specific protection area 1 to 10)
Signal flow	NC → PLC
Data type	ARRAY[1..10] of BOOL
Update	Cyclic
Value FALSE	The channel-specific protection area 1 (...10) is not preactivated.
Value TRUE	The channel-specific protection area 1 (...10) is preactivated. Preactivation is realized in the NC program using command NPROT.
Corresponds to	• "LBP_Chan*.A_ProtZone" (DB21, ... DBX1.1) Enable protection areas • "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) Activate machine-related protection area 1 to 10 • "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) activate channel-specific protection area 1 to 10 • "LBP_Chan*.E_MRPactivated" (DB21, ... DBX272.0 ... 273.1) Machine-related protection area 1 to 10 pre-activated • "LBP_Chan*.E_MRPViolated" (DB21, ... DBX276.0 ... 277.1) Machine-related protection area 1 to 10 violated • "LBP_Chan*.E_CSPViolated" (DB21, ... DBX278.0 ... 279.1) Channel-specific protection area 1 to 10 violated
Additional references	Function Manual Basic Functions; Section "Monitoring and compensating" > "Protection areas"

E_MRPViolated (DB21, ... from DBX276.0)

Description	Machine-related protection area 1 to 10 violated
Signal flow	NC → PLC
Data type	ARRAY[1..10] of BOOL
Update	Cyclic
Value FALSE	The activated, machine-related protection area 1 to 10 was not violated in the current block or in the current JOG traversing motion. The preactivated, machine-related protection area 1 to 10 would not be violated in the current block if it would be active.
Value TRUE	The activated, machine-related protection area 1 to 10 was violated in the current block or in the current JOG traversing motion. The preactivated, machine-related protection area 1 to 10 would be violated in the current block if it would be active.

Corresponds to	• "LBP_Chan*.A_ProtZone" (DB21, ... DBX1.1) Enable protection areas • "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) Activate machine-related protection area 1 to 10 • "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) activate channel-specific protection area 1 to 10 • "LBP_Chan*.E_MRPactivated" (DB21, ... DBX272.0 ... 273.1) Machine-related protection area 1 to 10 pre-activated • "LBP_Chan*.E_CSPactivated" (DB21, ... DBX274.0 ... 275.1) Channel-specific protection area 1 to 10 pre-activated • "LBP_Chan*.E_CSPViolated" (DB21, ... DBX278.0 ... 279.1) Channel-specific protection area 1 to 10 violated
Additional references	Function Manual Basic Functions; Section "Monitoring and compensating" > "Protection areas"

E_CSPViolated (DB21, ... ab DBX278.0)

Description	Channel-specific protection area 1 to 10 violated
Signal flow	NC → PLC
Data type	ARRAY[1..10] of BOOL
Update	Cyclic
Value FALSE	The activated, channel-specific protection area 1 to 10 was not violated in the current block or in the current JOG traversing motion. The preactivated, channel-specific protection area 1 to 10 would not be violated in the current block if it would be active.
Value TRUE	The activated, channel-specific protection area 1 to 10 wasviolated in the current block or in the current JOG traversing motion. The preactivated, channel-specific protection area 1 to 10 would be violated in the current block if it would be active.
Corresponds to	• "LBP_Chan*.A_ProtZone" (DB21, ... DBX1.1) Enable protection areas • "LBP_Chan*.A_MRP" (DB21, ... DBX8.0 ... 9.1) Activate machine-related protection area 1 to 10 • "LBP_Chan*.A_CSP" (DB21, ... DBX10.0 ... 11.1) activate channel-specific protection area 1 to 10 • "LBP_Chan*.E_MRPactivated" (DB21, ... DBX272.0 ... 273.1) Machine-related protection area 1 to 10 pre-activated • "LBP_Chan*.E_CSPactivated" (DB21, ... DBX274.0 ... 275.1) Channel-specific protection area 1 to 10 pre-activated • "LBP_Chan*.E_MRPViolated" (DB21, ... DBX276.0 ... 277.1) Machine-related protection area 1 to 10 violated
Additional references	Function Manual Basic Functions; Section "Monitoring and compensating" > "Protection areas"

10.4.9.13 LBP_Chan1 [DB21], ... - Synchronized actions, signals from/to channel

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

A_Str_Synon (DB21, ... DBX280.1)

Description	Request disabling of synchronized actions
Signal flow	PLC → NC
Data type	BOOL

A_Str_Synfrom (DB21, ... DBX281.1)

Description	Acknowledgement of synchronized actions disabled
Signal flow	NC → PLC
Data type	BOOL

A_D1 (DB21, ... DBW282)

A_D2 (DB21, ... DBW284)

A_D3 (DB21, ... DBW286)

A_D4 (DB21, ... DBW288)

A_D5 (DB21, ... DBW290)

A_D6 (DB21, ... DBW292)

A_D7 (DB21, ... DBW294)

A_D8 (DB21, ... DBW296)

A_D9 (DB21, ... DBW298)

Description	Assignment D1 to D9
Data type	INT

A_SynDisabl (DB21, ... ab DBX300.0)

Description	Deactivate synchronized action
Signal flow	PLC → NC
Data type	ARRAY[1..64] of BOOL

E_SynDisabled (DB21, ... ab DBX308.0)

Description	Synchronized action deactivated
Signal flow	NC → PLC
Data type	ARRAY[1..64] of BOOL

10.4.9.14 LBP_Chan1 [DB21], ... - Control signals from/to channel

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_G00 (DB21, ... DBX316.0)

Description	Active G functions
Signal flow	PLC ↔ NC
Data type	BOOL

E_ExtLang (DB21, ... DBX317.0)

Description	External language mode active
Data type	BOOL

E_WS_Setp (DB21, ... DBX317.1)

Description	Target number of workpieces reached
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The number of machined workpieces (actual workpiece total) is not equal to the number of workpieces to be machined (target [required] number of workpieces): $AC_ACTUAL_PARTS <> $AC_REQUIRED_PARTS
Value TRUE	The number of machined workpieces (actual workpiece total) is equal to the number of workpieces to be machined (target [required] number of workpieces): $AC_ACTUAL_PARTS == $AC_REQUIRED_PARTS
Corresponds to	• MD27880 $MC_PART_COUNTER Activation of workpiece counters
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Program runtime / workpiece counter" > "Workpiece counter"

E_DriveTest (DB21, ... DBX317.5)

Description	Drive test traverse request
Data type	BOOL

E_PTP_Travel (DB21, ... DBX317.6)

Description	PTP traversing active **Note** The variable is only relevant in JOG mode when transformation is active.
Signal flow	NC → PLC
Data type	BOOL
Edge change 0 → 1	PTP (Point-To-Point) traversing is active.
Edge change 1 → 0	CP (Continuous Path) traversing is active.
Corresponds to	• "LBP_Chan*.A_PTP_Travel" (DB21, ... DBX29.4) Activate PTP travel
Additional references	Function Manual Technologies

E_ToolMissing (DB21, ... DBX317.7)

Description	TOOLMAN: Tool missing A tool is either missing or cannot be used.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	Irrelevant
Value TRUE	The programmed tool is missing.

E_ASUP_Stop (DB21, ... DBX318.0)

Description	ASUB is stopped The interface signal is only processed in the following states: • AUTOMATIC or MDI mode • Channel state "interrupted" • Program state "stopped"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The ASUB is not in the "stopped" state. **Note** For the following events, the signal is always set to 0: • NC start • Channel reset
Value TRUE	The ASUB is in the "stopped" state. **Note** The ASUB was stopped by the control system before it was completely executed.
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program mode, reset response" > "Asynchronous subprogram (ASUBs)" > "Function" > "ASUB with REPOSA"

E_SearchAct (DB21, ... DBX318.1)

Description	Block search via program test is active (SERUPRO) The block search (SERUPRO) can only be activated in AUTOMATIC mode in program state "Aborted".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Block search via program test is not active When the target block is changed to in the main run (internal channel state "Program test" is deselected; stop condition: "Search target found" is displayed), the interface signal is reset.

Value TRUE	Block search via program test is active During processing of the blocks as part of the block search (internal channel state: "Program test"), the interface signal is set up until the target block is changed to in the main run (program state: "stopped").
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_TOFF (DB21, ... DBX318.2)

Description	Online tool length compensation (TOFF) active With the online length compensation, the effective tool length can be changed in real time so that the length changes are also considered for changes in orientation of the tool. Application areas: • Orientation transformations (TRAORI) • Orientable tool carrier (TCARR)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Online tool length compensation (TOFF) is not active.
Value TRUE	Online tool length compensation (TOFF) is active.
Corresponds to	• "LBP_Chan*.E_TOFFmovem" (DB21, ... DBX318.3) Online tool length compensation (TOFF): Compensation motion active
Additional references	Function Manual Transformations, Section: "Multi-axis transformations"

E_TOFFmovem (DB21, ... DBX318.3)

Description	Online tool length compensation (TOFF): Compensation motion active If, for active online tool length compensation ("LBP_Chan*.E_TOFF" (DB21, ... DBX318.2) == TRUE), compensation motion is active, the interface signal is set to TRUE: "LBP_Chan*.E_TOFFmovem" (DB21, ... DBX318.3) == TRUE
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Compensation motion is not active.
Value TRUE	Compensation motion active.
Corresponds to	• "LBP_Chan*.E_TOFF" (DB21, ... DBX318.2) Online tool length compensation (TOFF) active
Additional references	Function Manual Transformations, Section: "Multi-axis transformations"

E_StopDelayed (DB21, ... DBX318.4)

Description	Delayed stop
Data type	BOOL

E_AssM01 (DB21, ... DBX318.5)

Description	Associated help function active "Associated help function" (M-1) is selected at the SINUMERIK Operate user interface in the operating area "Automatic" > "Program control" by setting the HMI/PLC interface signal "LBP_Chan*.E_MMC_M01AssocNC" (DB21, ... DBX24.4). Depending on the value of the FC1 parameter `MMCToIf`, the interface signal is transmitted from the PLC basic program to the NC/PLC interface signal DB21, ... DBX30.5: • "TRUE": Transmission • "FALSE": No transmission The default value of the parameter is "TRUE".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Associated help function is not active.
Value TRUE	"Associated help function" is active.
Corresponds to	• "LBP_Chan*.E_MMC_M01AssocNC" (DB21, ... DBX24.4) Associated help function selected • "LBP_Chan*.A_NCKrelatedM01" (DB21, ... DBX30.5) Activate associated help function • MD22254 $MC_AUXFU_ASSOC_M0_VALUE Additional M function for program stop • MD22256 $MC_AUXFU_ASSOC_M1_VALUE Additional M function for conditional stop
Additional references	Function Manual Basic Functions; Section "Auxiliary function outputs to PLC"

E_DRY (DB21, ... DBX318.6)

Description	Dry run feedrate active
Data type	BOOL

E_Overstore (DB21, ... DBX318.7)

Description	Overstore active
Data type	BOOL

E_REPOS_EdgeAckn (DB21, ... DBX319.0)

Description	REPOS mode change acknowledgment
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The acknowledgment of the REPOS mode change is not available. The SERUPRO-ASUB stops automatically before REPOS and DB21, ... DBX31.4 (REPOS mode change) does not affect the SERUPRO approach.

Value TRUE	The acknowledgment of the REPOS mode change is available. The interface signal DB21, ... DBX31.4 (REPOS mode change) identified by the NC is acknowledged if the requested REPOS mode DB21, ... DBX31.0-2 and delay signal DB31, ... DBX10.0 were accepted in the NC. The signal states refer to the actual main run block.
Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgement • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_REPOS_PMode0, **E_REPOS_PMode1**, **E_REPOS_PMode2**

(DB21, ... DBX319.1, 319.2, 319.3)

Description	Active REPOS mode Active REPOS mode:			
	E_REPOS_PMode2 Bit 3	**E_REPOS_PMode1 Bit 2**	**E_REPOS_PMode0 Bit 1**	**Active REPOS approach mode**
	0	0	0	No REPOS approach mode active
	0	0	1	Reposition to start of block `RMBBL`
	0	1	0	Reposition to interrupt point `RMIBL`
	0	1	1	Reposition to end of block `RMEBL`
	1	0	0	Reposition to nearest path point `RMNBL`

Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgement • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

E_DelayFTS (DB21, ... DBX319.4)

Description	Feed stop delay
Data type	BOOL

E_REPOS_DEFERRA (DB21, ... DBX319.5)

Description	REPOS delay All axes currently controlled by this channel have either no REPOS offset or their REPOS offsets are not traversed through.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	A REPOS delay is not active.
Value TRUE	A REPOS delay is active.

Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgement • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"

E_Stop_NoDelayR (DB21, ... DBX319.6)

Description	Suppress feed stop delay
Data type	BOOL

E_NoToolChangeCmd (DB21, ... DBX319.7)

Description	No tool change commands active (Signal available for SINUMERIK ONE)
Data type	BOOL

10.4.9.15 LBP_Chan1 [DB21], ... - Signals to orientation axes

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

A_Ori (DB21, ... ab DBB320)

Description	Control signals to the orientation axes
Data type	ARRAY[1..3] of STRUCT STRUCT "A_Ori" Control signals for orientation axes (Page 366)

10.4.9.16 LBP_Chan1 [DB21], ... - Signals from orientation axes

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_Ori (DB21, ... from DBB332)

Description	Control signals from the orientation axes
Data type	ARRAY[1..3] of STRUCT STRUCT "E_Ori" control signals of the orientation axis (Page 375)

10.4.9.17 LBP_Chan1 [DB21], ... - Tool management functions from channel

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_TPrewarnLimit (DB21, ... DBX344.0)

Description	TOOLMAN: Tool prewarning limit reached The T number of the tool, whose prewarning value has been reached, is located in: "LBP_Chan*.E_TNoPrewarnLimit" (DB21, ... DBD348) The change signal is active for one "Main [OB1]" cycle.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	Irrelevant
Value TRUE	The tool prewarning limit has been reached
Corresponds to	• "LBP_Chan*.E_TLimit" (DB21, ... DBX344.1) Tool management change signal: T number for tool limit value • "LBP_Chan*.E_TNewRepl" (DB21, ... DBX344.2) Tool management change signal: T number of the new replacement tool • "LBP_Chan*.E_TLastRepl" (DB21, ... DBX344.3) Tool change signal: T number of the last replacement tool • "LBP_Chan*.E_TNoPrewarnLimit" (DB21, ... DBD348) T number for tool prewarning limit
Additional references	Function Manual, Tool Management

E_TLimit (DB21, ... DBX344.1)

Description	TOOLMAN: Tool limit value reached The T number of the tool, whose limit value has been reached, is located in: "LBP_Chan*.E_TNo-Limit" (DB21, ... DBD352) The change signal is active for one "Main [OB1]" cycle.
Signal flow	NC → PLC

Data type	BOOL
Update	Job-controlled
Value FALSE	Irrelevant
Value TRUE	The tool limit value has been reached.
Corresponds to	• "LBP_Chan*.E_TPrewarnLimit" (DB21, ... DBX344.0) Tool management change signal: T number for tool prewarning limit • "LBP_Chan*.E_TNewRepl" (DB21, ... DBX344.2) Tool management change signal: T number of the new replacement tool • "LBP_Chan*.E_TLastRepl" (DB21, ... DBX344.3) Tool change signal: T number of the last replacement tool • "LBP_Chan*.E_TNoLimit" (DB21, ... DBD352) T number for tool limit value
Additional references	Function Manual, Tool Management

E_TNewRepl (DB21, ... DBX344.2)

Description	TOOLMAN: Transition to new replacement tool The T number for the new replacement tool is located in: "LBP_Chan*.E_TNoNewRepl" (DB21, ... DBD356) The change signal is active for one "Main [OB1]" cycle.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	Irrelevant
Value TRUE	A transition is made to a new replacement tool.
Corresponds to	• "LBP_Chan*.E_TPrewarnLimit" (DB21, ... DBX344.0) Tool management change signal: T number for tool prewarning limit • "LBP_Chan*.E_TLimit" (DB21, ... DBX344.1) Tool management change signal: T number for tool limit value • "LBP_Chan*.E_TLastRepl" (DB21, ... DBX344.3) Tool change signal: T number of the last replacement tool • "LBP_Chan*.E_TNoNewRepl" (DB21, ... DBD356) T number of the new replacement tool
Additional references	Function Manual, Tool Management

E_TLastRepl (DB21, ... DBX344.3)

Description	TOOLMAN: Last replacement tool of the tool group The T number of the last replacement tool of the tool group is located in: "LBP_Chan*.E_TNo-LastRepl" (DB21, ... DBD360) The change signal is active for one "Main [OB1]" cycle. **Note** The monitoring for the last replacement tool must be activated in MD20310 $MC_TOOL_MAN-AGEMENT_MASK, bit 18.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled

Value FALSE	Irrelevant
Value TRUE	The last replacement tool of the tool group has been reached
Corresponds to	• "LBP_Chan*.E_TPrewarnLimit" (DB21, ... DBX344.0) Tool management change signal: T number for tool prewarning limit • "LBP_Chan*.E_TLimit" (DB21, ... DBX344.1) Tool management change signal: T number for tool limit value • "LBP_Chan*.E_TNewRepl" (DB21, ... DBX344.2) Tool management change signal: T number of the new replacement tool • "LBP_Chan*.E_TNoLastRepl" (DB21, ... DBD360) T number of the last replacement tool
Additional references	Function Manual, Tool Management

E_TNoPrewarnLimit (DB21, ... DBD348)

Description	Transferred tool management functions: T number of the tool, whose prewarning value has been reached
Signal flow	NC → PLC
Data type	DINT

E_TNoLimit (DB21, ... DBD352)

Description	Transferred tool management functions: T number of the tool, whose limit value has been reached
Signal flow	NC → PLC
Data type	DINT

E_TNoNewRepl (DB21, ... DBD356)

Description	Transferred tool management functions: T number of the new replacement tool
Signal flow	NC → PLC
Data type	DINT

E_TNoLastRepl (DB21, ... DBD360)

Description	Transferred tool management functions: T number of the last replacement tool
Signal flow	NC → PLC
Data type	DINT

10.4.9.18 LBP_Chan1 [DB21], ... - Control signals from/to channel (2)

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_Cycle_Sig (DB21, ... DBW364)

Description	Cycle input signals
Data type	ARRAY[0..15] of BOOL

A_Cycle_Sig (DB21, ... DBW366)

Description	Cycle output signals
Data type	ARRAY[0..15] of BOOL

E_OEM_TechnoSig (DB21, ... DBD368)

Description	OEM Techno input signals
Data type	ARRAY[0..31] of BOOL

A_OEM_TechnoSig (DB21, ... DBD372)

Description	OEM Techno output signals
Data type	ARRAY[0..31] of BOOL

E_ProgEvent_Start (DB21, ... DBX376.0)

E_ProgEvent_M30 (DB21, ... DBX376.1)

E_ProgEvent_Reset (DB21, ... DBX376.2)

E_ProgEvent_PowerOn (DB21, ... DBX376.3)

E_ProgEvent_SearchRun (DB21, ... DBX376.4)

Description	PROG_EVENT trigger event The event that initiated the PROG_EVENT is displayed bit-coded:		
	Variable (Bit)	**Value**	**Event**
	E_ProgEvent_Start (Bit 0)	TRUE	NC start from channel state "reset"
	E_ProgEvent_M30 (Bit 1)	TRUE	End of program reset (e.g. M30)
	E_ProgEvent_Reset (Bit 2)	TRUE	Channel reset
	E_ProgEvent_PowerOn (Bit 3)	TRUE	Warm restart (power on)
	E_ProgEvent_SearchRun (Bit 4)	TRUE	1. NC start after search
	(Bits 5 ... 7)	---	Reserved, currently always FALSE
	Note As a minimum, the interface signal must remain for a complete PLC cycle.		
Signal flow	NC → PLC		
Data type	BOOL		
Update	Cyclic		
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response"		

E_CollCheckStop (DB21, ... DBX377.0)

Description	Collision avoidance: Stop
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The collision avoidance has not triggered a stop of the traversing motions in the channel.
Value TRUE	The collision avoidance has triggered a stop of the traversing motion in the channel.
Additional references	Function Manual Monitoring and Compensating; Section "Collision avoidance"

E_StopCondition (DB21, ... DBX377.1)

Description	Stop condition
Signal flow	NC → PLC
Data type	BOOL

E_JogRetract (DB21, ... DBX377.4)

Description	JOG retract active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	JOG retract has not been selected.
Value TRUE	JOG retract has been selected and is active.
Corresponds to	"LBP_Chan*.E_RetractData" (DB21, ... DBX377.5) JOG retract retraction data available
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_RetractData (DB21, ... DBX377.5)

Description	JOG retract retraction data available If the signal state is active, then the JOG retract can be selected (user interface or PI service "RETRAC").
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Retraction data is not available for traversing in the tool direction. JOG retract cannot be selected
Value TRUE	Retraction data is available for traversing in the tool direction.
Corresponds to	• "LBP_Chan*.E_JogRetract" (DB21, ... DBX377.4) JOG retract active
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_JOG_Circles (DB21, ... DBX377.6)

Description	JOG circular travel active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	The "Circular travel in JOG" function is not active.
Value TRUE	The "Circular travel in JOG" function is active.
Corresponds to	• "LBP_Chan*.A_JOG_Circles" (DB21, ... DBX30.6) JOG circular travel
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

E_AnyAsup (DB21, ... DBX378.0)

Description	ASUB is active **Note** Also outside block FC9, feedback about a running ASUB is realized using the interface signal.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No ASUB is active.
Value TRUE	An ASUB is active.
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Asynchronous subprograms (ASUBs)"

E_SilentAsup (DB21, ... DBX378.1)

Description	"Silent" ASUB is active A "silent" ASUB is an ASUB where blocks are not displayed on the screen as this has been suppressed with `DISPLOF`.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No "silent" ASUB is active.
Value TRUE	A "silent" ASUB is active.
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Asynchronous subprograms (ASUBs)"

A_ProgJump (DB21, ... DBX384.0)

Description	Enable GOTOS
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to enable GOTOS (return jump to the program start) in the channel. NC program execution is continued with the next command following GOTOS.
Value TRUE	A request has been made to enable `GOTOS` (return jump to the program start) in the channel.

Corresponds to	• MD27850 $MC_PROG_NET_TIMER_MODE Impact of the program runtime net counter • MD27860 $MC_PROCESSTIMER_MODE Activation of the program runtime measurement • MD27880 $MC_PART_COUNTER Activation of the workpiece counters
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Program mode" > "Program jumps" > "Return to start of program"

A_InKeyG1 (DB21, ... DBX385.0)

A_InKeyG2 (DB21, ... DBX385.1)

A_InKeyG3 (DB21, ... DBX385.2)

A_InKeyG4 (DB21, ... DBX385.3)

A_InKeyG5 (DB21, ... DBX385.4)

A_InKeyG6 (DB21, ... DBX385.5)

A_InKeyG7 (DB21, ... DBX385.6)

A_InKeyG8 (DB21, ... DBX385.7)

Description	Grinding: Input signals 1 ... 8
Signal flow	PLC → NC
Data type	BOOL

A_InKeyGEn1 (DB21, ... DBX386.0)

A_InKeyGEn2 (DB21, ... DBX386.1)

A_InKeyGEn3 (DB21, ... DBX386.2)

A_InKeyGEn4 (DB21, ... DBX386.3)

A_InKeyGEn5 (DB21, ... DBX386.4)

A_InKeyGEn6 (DB21, ... DBX386.5)

A_InKeyGEn7 (DB21, ... DBX386.6)

A_InKeyGEn8 (DB21, ... DBX386.7)

Description	Grinding: Disable input signals 1 ... 8
Signal flow	PLC → NC
Data type	BOOL

A_InKeyGRunIn1 (DB21, ... DBX387.0)

A_InKeyGRunIn2 (DB21, ... DBX387.1)

A_InKeyGRunIn3 (DB21, ... DBX387.2)

A_InKeyGRunIn4 (DB21, ... DBX387.3)

A_InKeyGRunIn5 (DB21, ... DBX387.4)

A_InKeyGRunIn6 (DB21, ... DBX387.5)

A_InKeyGRunIn7 (DB21, ... DBX387.6)

A_InKeyGRunIn8 (DB21, ... DBX387.7)

Description	Grinding: Status of the grinding functions 1 ... 8
Signal flow	PLC → NC
Data type	BOOL

E_TransformNo (DB21, ... DBB388)

Description	Index number of the active transformation in the channel If the transformation is active, the interface signal shows the transformer index of the active transformation. **Example** Program code: N24100 $MC_TRAFO_TYP_1=24 N24200 $MC_TRAFO_TYP_2=513 N24300 $MC_TRAFO_TYPE_3=24 Display in the interface signal: TRAORI → E_TransformNo=1 TRAORI(2) → E_TransformNo=3
Signal flow	NC → PLC
Data type	BYTE
Update	Cyclic
Corresponds to	• "LBP_Chan*.E_Transform" (DB21, ... DBX33.6) Transformation active

E_InKeyGIsEn1 (DB21, ... DBX390.0)

E_InKeyGIsEn2 (DB21, ... DBX390.1)

E_InKeyGIsEn3 (DB21, ... DBX390.2)

E_InKeyGIsEn4 (DB21, ... DBX390.3)

E_InKeyGIsEn5 (DB21, ... DBX390.4)

E_InKeyGIsEn6 (DB21, ... DBX390.5)

E_InKeyGIsEn7 (DB21, ... DBX390.6)

E_InKeyGIsEn8 (DB21, ... DBX390.7)

Description	Grinding: Enable status of input signals 1 ... 8
Signal flow	PLC → NC
Data type	BOOL

E_InKeyGRunOut1 (DB21, ... DBX391.0)

E_InKeyGRunOut2 (DB21, ... DBX391.1)

E_InKeyGRunOut3 (DB21, ... DBX391.2)

E_InKeyGRunOut4 (DB21, ... DBX391.3)

E_InKeyGRunOut5 (DB21, ... DBX391.4)

E_InKeyGRunOut6 (DB21, ... DBX391.5)

E_InKeyGRunOut7 (DB21, ... DBX391.6)

E_InKeyGRunOut8 (DB21, ... DBX391.7)

Description	Grinding: Status of the grinding functions 1 ... 8
Signal flow	NC → PLC
Data type	BOOL

E_CART_JOG_MODE (DB21, ... DBB392)

Description	Selection: Coordinate system for Cartesian manual traversing and handwheel override in automatic mode in the tool direction (DRF) Selection of the coordinate system for: • Cartesian manual traversing of the geometry axes in the tool direction (TCS) • Handwheel override of the geometry axes in the AUTOMATIC mode in the tool direction (DRF)	
Signal flow	PLC → NC	
Data type	BYTE	
Update	Cyclic	
Value	**Value**	**Description**
	0	The function is not selected
	1	Manual traversing of the geometry axes in the basic coordinate system (BCS)
	2	Manual traversing of the geometry axes in the workpiece coordinate system (WCS)
	3	Manual traversing or handwheel override (DRF) of the geometry axes in the workpiece coordinate system (WCS)
Additional references	• Function Manual Basic Functions, Section: – "Axes, coordinate systems, frames" > "Coordinate systems" > "Additive offsets" > "DRF offset" – Function Manual for PLC • Function Manual Transformations, Section: – "Kinematics transformation"	

10.4.9.19 LBP_Chan1 [DB21], ... - control signals from/to HMI

The data blocks "LBP_Chan1 [DB21], ..." are the interface between the user program and channel.

Variables in "LBP_Chan1 [DB21], ..."

E_AxisSelType (DB21, ... DBB408)

Description	Type of the axis selected at the HMI
Signal flow	HMI → PLC
Data type	BYTE

Update	On request from the HMI
Value	0: Machine axis 1: Geo axis

E_AxisSelAxis (DB21, ... DBB409)

Description	Number of the axis selected at the HMI
Signal flow	HMI → PLC
Data type	BYTE
Update	On request from the HMI

E_AxisSelRequest (DB21, ... DBX410.0)

Description	Axis selection key at the HMI is actuated
Signal flow	HMI → PLC
Data type	BOOL
Update	On request from the HMI

A_AxisSelTypeActive (DB21, ... DBB412)

Description	Feedback message of the axis selection to the HMI: active axis selection
Signal flow	PLC → HMI
Data type	BYTE
Update	After being requested from the HMI
Value	0: Machine axis 1: Geo axis

A_AxisSetAxisActive (DB21, ... DBB413)

Description	Feedback message to the HMI: active axis number
Signal flow	PLC → HMI
Data type	BYTE
Update	After being requested from the HMI

10.4.9.20 STRUCT "A_Geo", control signals for geometry axis

The variable structure "A_Geo" describes the control signals to the geometry axis.

Variables in "A_Geo"

HW1 (DB21, ... DBX12.0, 16.0, 20.0)

HW2 (DB21, ... DBX12.1, 16.1, 20.1)

HW3 (DB21, ... DBX12.2, 16.2, 20.2)

<table>
<tr><td>Description</td><td>Geometry axes: Activate handwheel

Geometry axis 1: "LBP_Chan*.A_Geo.A_Geo[1].HW*" (DB21, ... DBX12.0, 12.1, 12.2)
Geometry axis 2: "LBP_Chan*.A_Geo.A_Geo[2].HW*" (DB21, ... DBX16.0, 16.1, 16.2)
Geometry axis 3: "LBP_Chan*.A_Geo.A_Geo[3].HW*" (DB21, ... DBX20.0, 20.1, 20.2)

The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.
Bit coded: maximum 3 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
</table>
Binary coded: maximum 6 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Chan*.E_Geo.E_Geo[1].HW*" (DB21, ... DBX40.0, 40.1, 40.2)
Geometry axis 1: Handwheel active
• "LBP_Chan*.E_Geo.E_Geo[2].HW*" (DB21, ... DBX46.0, 46.1, 46.2)
Geometry axis 2: Handwheel active
• "LBP_Chan*.E_Geo.E_Geo[3].HW*" (DB21, ... DBX52.0, 52.1, 52.2)
Geometry axis 3: Handwheel active
• MD11324 $MN_HANDWH_VDI_REPRESENTATION
Representation of the handwheel number at the VDI interface</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Manual traversing"</td></tr>
</table>

FDStop (DB21, ... DBX12.3, 16.3, 20.3)

Description	Feed stop, geometry axis 1, 2, 3 • If the interface signal is set while traversing a geometry axis, then the geometry axis is braked down to standstill along its braking characteristic. • After the interface signal has been reset, traversing motion that was stopped is continued. • The position control is retained and the following error is eliminated. • If the interface signal is set, and a request is issued to traverse an axis, the axis is not traversed. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the axis is traversed. • The interface signal is only effective in JOG mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Feed disable for geometry axis is not active.
Value TRUE	Feed disable for geometry axis is active.

Disabl (DB21, ... DBX12.4, 16.4, 20.4)

Description	Geometry axes: Traversing key disable Geometry axis 1: "LBP_Chan*.A_Geo.A_Geo[1].Disabl" (DB21, ... DBX12.4) Geometry axis 2: "LBP_Chan*.A_Geo.A_Geo[2].Disabl" (DB21, ... DBX16.4) Geometry axis 3: "LBP_Chan*.A_Geo.A_Geo[3].Disabl" (DB21, ... DBX20.4) **Note** When the traversing key disable is activated while traversing, then traversing is canceled.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The traversing key lock for the geometry axis is not active.
Value TRUE	The traversing key lock for the geometry axis is active.
Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].Minus" (DB21, ... DBX12.6) Geometry axis 1: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[1].Plus" (DB21, ... DBX12.7) Geometry axis 1: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[2].Minus" (DB21, ... DBX16.6) Geometry axis 2: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[2].Plus" (DB21, ... DBX16.7) Geometry axis 2: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[3].Minus" (DB21, ... DBX20.6) Geometry axis 3: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[3].Plus" (DB21, ... DBX20.7) Geometry axis 3: Traversing key "Plus"
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

RapidTrOR (DB21, ... DBX12.5, 16.5, 20.5)

Description	Geometry axes: Rapid traverse override Geometry axis 1: "LBP_Chan*.A_Geo.A_Geo[1].RapidTrOR" (DB21, ... DBX12.5) Geometry axis 2: "LBP_Chan*.A_Geo.A_Geo[2].RapidTrOR" (DB21, ... DBX16.5) Geometry axis 3: "LBP_Chan*.A_Geo.A_Geo[3].RapidTrOR" (DB21, ... DBX20.5) The signal is only effective for continuous or incremental manual traversing in the JOG mode. The signal is irrelevant: • for reference point approach (JOG mode) • in the AUTOMATIC and MDI modes The rapid traverse velocity can be influenced using the rapid traverse override switch.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	When manually traversing the geometry axis using the traversing keys, the entered JOG velocity becomes active (SD41110 or MD32020).
Value TRUE	When manually traversing the geometry axis using the traversing keys, when the interface signal is set, rapid traverse velocity becomes effective (MD32010).
Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].Minus" (DB21, ... DBX12.6) Geometry axis 1: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[1].Plus" (DB21, ... DBX12.7) Geometry axis 1: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[2].Minus" (DB21, ... DBX16.6) Geometry axis 2: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[2].Plus" (DB21, ... DBX16.7) Geometry axis 2: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[3].Minus" (DB21, ... DBX20.6) Geometry axis 3: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[3].Plus" (DB21, ... DBX20.7) Geometry axis 3: Traversing key "Plus" • MD32010 $MA_JOG_VELO_RAPID Conventional rapid traverse • MD32020 $MA_JOG_VELO Conventional axis velocity • SD41110 $SN_JOG_SET_VELO Axis velocity for JOG
Additional references	• Function Manual Axes and Spindles; Section "Feedrates" • Function Manual Axes and Spindles; Section "Manual traversing"

Minus (DB21, ... DBX12.6, 16.6, 20.6)

Plus (DB21, ... DBX12.7, 16.7, 20.7)

Description	Geometry axes: Traversing keys "Minus" / "Plus" Geometry axis 1: "LBP_Chan*.A_Geo.A_Geo[1].Minus" (DB21, ... DBX12.6) Geometry axis 1: "LBP_Chan*.A_Geo.A_Geo[1].Plus" (DB21, ... DBX12.7) Geometry axis 2: "LBP_Chan*.A_Geo.A_Geo[2].Minus" (DB21, ... DBX16.6) Geometry axis 2: "LBP_Chan*.A_Geo.A_Geo[2].Plus" (DB21, ... DBX16.7) Geometry axis 3: "LBP_Chan*.A_Geo.A_Geo[3].Minus" (DB21, ... DBX20.6) Geometry axis 3: "LBP_Chan*.A_Geo.A_Geo[3].Plus" (DB21, ... DBX20.7) There is a request signal for every traversing key and/or axis direction: • "Minus" (Bit 6) Traversing key "Minus" for traversing in the negative axis direction • "Plus" (Bit 7) Traversing key "Plus" for traversing in the positive axis direction Depending on the active machine function, as well as the settings for jog and continuous operation (SD41050 and MD11300), different responses are initiated when the signal changes: 1. Continuous manual travel in the jog mode The geometry axis traverses in the direction concerned to the active limit switch as long as the interface signal is set to TRUE. 2. Continuous manual travel in continuous operation At the first edge change 0 →1, the geometry axis starts to traverse in the relevant direction. This traversing movement still continues when the edge changes from 1 → 0. Any new signal edge change 0 → 1 (same traversing direction!) stops the traversing movement. 3. Incremental manual travel in the jog mode With signal state TRUE, the geometry axis starts to traverse at the set increment. If the signal changes to the FALSE state before the increment is traversed, then traversing is interrupted. When the signal state changes to TRUE again, traversing is continued. The geometry axis can be stopped and started several times as described above until it has traversed the complete increment. 4. Incremental manual travel in continuous operation At the first edge change 0 → 1, the geometry axis starts to traverse at the set increment. If another edge change 0 → 1 is performed with the same traverse signal before the geometry axis has traversed the increment, the traversing movement will be canceled. The increment is no longer traversed to the end. **Note** • The interface signal is irrelevant in operating modes: – AUTOMATIC

	– MDI • If both traversing signals ("plus" and "minus") are set at the same time there is no traversing or the current traversing is aborted. • In contrast to machine axes, for geometry axes, only one geometry axis can be traversed at any one time using the traversing keys. • With the traversing key lock of the geometry axes, it is possible to lock traversing using traversing keys. • In the following cases, the geometry axis cannot be traversed in the JOG operating mode: – The geometry axis is already being traversed via the axis-specific interface as machine axis. – If another geometry axis is already being traversed with the traversing keys.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to traverse the geometry axis using the traversing key in the positive/negative axis direction.
Value TRUE	A request has been made to traverse the geometry axis using the traversing key in the positive/ negative axis direction.
Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].Disabl" (DB21, ... DBX12.4) Geometry axis 1: Traversing key disable • "LBP_Chan*.A_Geo.A_Geo[2].Disabl" (DB21, ... DBX16.4) Geometry axis 2: Traversing key disable • "LBP_Chan*.A_Geo.A_Geo[3].Disabl" (DB21, ... DBX20.4) Geometry axis 3: Traversing key disable • "LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) Traversing keys "Plus" / "Minus" • MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD INC and REF in jog mode • SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD Jog/continuous operation for JOG, continuous
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

INC1 (DB21, ... DBX13.0, 17.0, 21.0)

INC10 (DB21, ... DBX13.1, 17.1, 21.1)

INC100 (DB21, ... DBX13.2, 17.2, 21.2)

INC1000 (DB21, ... DBX13.3, 17.3, 21.3)

INC10000 (DB21, ... DBX13.4, 17.4, 21.4)

INCVar (DB21, ... DBX13.5, 17.5, 21.5)

ContManTravel (DB21, ... DBX13.6, 17.6, 21.6)

<table>
<tr><td>Description</td><td>Geometry axes: Machine function request

Geometry axis 1:
"LBP_Chan*.A_Geo.A_Geo[1].INC*", "LBP_Chan*.A_Geo.A_Geo[1].ContManTravel"
(DB21, ... DBX13.0 ... 13.6)
Geometry axis 2:
"LBP_Chan*.A_Geo.A_Geo[2].INC*", "LBP_Chan*.A_Geo.A_Geo[2].ContManTravel"
(DB21, ... DBX17.0 ... 17.6)
Geometry axis 3:
"LBP_Chan*.A_Geo.A_Geo[3].INC*", "LBP_Chan*.A_Geo.A_Geo[3].ContManTravel"
(DB21, ... DBX21.0 ... 21.6)

There is a request signal for every machine function to manually traverse the geometry axis in the JOG mode:
<table>
<tr><td>"INC1" (Bit 0)</td><td>INC1</td></tr>
<tr><td>"INC10" (Bit 1)</td><td>INC10</td></tr>
<tr><td>"INC100" (Bit 2)</td><td>INC100</td></tr>
<tr><td>"INC1000" (Bit 3)</td><td>INC1000</td></tr>
<tr><td>"INC10000" (Bit 4)</td><td>INC10000</td></tr>
<tr><td>"INCVar" (Bit 5)</td><td>INCvar</td></tr>
<tr><td>"ContManTravel" (Bit 6)</td><td>Continuous manual travel</td></tr>
</table>
Incremental manual travel
In addition to five fixed increment sizes (default setting in MD11330: INC1, INC10, INC100, INC1000 and INC10000), a variable increment size (INCvar) that can be set via the setting data SD41010 is also available. The distance evaluation of one increment for fixed and variable increment sizes is performed via the axis-specific machine data MD31090.
When pressing the "Plus" or "Minus" traversing key, or by rotating the electronic handwheel, the geometry axis starts to traverse in the appropriate direction, corresponding to the number of increments of the active machine function.
Continuous manual travel
For continuous manual travel, the plus and minus traversing keys are selected to move the geometry axis continuously and the appropriate direction.

Notes
• If several requests are set simultaneously, no machine function becomes active.
• If a geometry axis is currently being traversed via a machine function, the movement is aborted through deselection or change of the machine function.</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Value FALSE</td><td>The machine function to manually traverse the geometry axis has not been requested.</td></tr>
<tr><td>Value TRUE</td><td>The machine function to manually traverse the geometry axis has been requested.</td></tr>
</table>

Corresponds to	• "LBP_Chan*.E_Geo.E_Geo[1].INC*" (DB21, ... DBX41.0 ... 41.5) Geometry axis 1: Active machine function • "LBP_Chan*.E_Geo.E_Geo[2].INC*" (DB21, ... DBX47.0 ... 47.5) Geometry axis 2: Active machine function • "LBP_Chan*.E_Geo.E_Geo[3].INC*" (DB21, ... DBX53.0 ... 53.5) Geometry axis 3: Active machine function • MD11320 $MN_HANDWH_IMP_PER_LATCH Handwheel pulses per detent position • MD11330 $MN_JOG_INCR_SIZE_TAB Increment size for INC/handwheel • MD31090 $MA_JOG_INCR_WEIGHT Evaluation of an increment for INC/handwheel • SD41010 $SN_JOG_VAR_INCR_SIZE Size of the variable increment for JOG
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

OEM (DB21, ... DBB14, 18, 22)

Description	OEM signals
Signal flow	PLC → NC
Data type	BYTE

InvHWDirOfRot (DB21, ... DBX15.0, 19.0, 23.0)

Description	Geometry axes: Invert handwheel direction of rotation Geometry axis 1: "LBP_Chan*.A_Geo.A_Geo[1].InvHWDirOfRot" (DB21, ... DBX15.0) Geometry axis 2: "LBP_Chan*.A_Geo.A_Geo[2].InvHWDirOfRot" (DB21, ... DBX19.0) Geometry axis 3: "LBP_Chan*.A_Geo.A_Geo[3].InvHWDirOfRot" (DB21, ... DBX23.0) **Note** It is only permissible to change the interface signal when the geometry axis is at a standstill. **Application examples** • The direction of movement of the handwheel does not match the expected direction of the axis. • A handwheel is assigned to several axes with different orientations.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to invert the direction of rotation of the handwheel assigned to the geometry axis.
Value TRUE	A request has been made to invert the direction of rotation of the handwheel assigned to the geometry axis.

Corresponds to	• "LBP_Chan*.E_Geo.E_Geo[1].InvHWDirOfRot" (DB21, ... DBX43.0) Geometry axis 1: Handwheel direction of rotation inversion active • "LBP_Chan*.E_Geo.E_Geo[2].InvHWDirOfRot" (DB21, ... DBX49.0) Geometry axis 2: Handwheel direction of rotation inversion active • "LBP_Chan*.E_Geo.E_Geo[3].InvHWDirOfRot" (DB21, ... DBX55.0) Geometry axis 3: Handwheel direction of rotation inversion active
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

10.4.9.21 STRUCT "E_Geo", control signals of the geometry axis

The variable structure "E_Geo" describes the control signals of the geometry axis.

Variables in "E_Geo"

HW1 (DB21, ... DBX40.0, 46.0, 52.0)

HW2 (DB21, ... DBX40.1, 46.1, 52.1)

HW3 (DB21, ... DBX40.2, 46.2, 52.2)

<table>
<tr><td>Description</td><td>Geometry axes: Handwheel active

Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].HW*" (DB21, ... DBX40.0, 40.1, 40.2) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].HW*" (DB21, ... DBX46.0, 46.1, 46.2) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].HW*" (DB21, ... DBX52.0, 52.1, 52.2)

The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.

Bit coded: maximum 3 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="4">Bit coded: maximum 6 handwheels</td></tr>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>NC → PLC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Chan*.A_Geo.A_Geo[1].HW*" (DB21, ... DBX12.0, 12.1, 12.2)
Geometry axis 1: Activate handwheel
• "LBP_Chan*.A_Geo.A_Geo[2].HW*" (DB21, ... DBX16.0, 16.1, 16.2)
Geometry axis 2: Activate handwheel
• "LBP_Chan*.A_Geo.A_Geo[3].HW*" (DB21, ... DBX20.0, 20.1, 20.2)
Geometry axis 3: Activate handwheel
• MD11324 $MN_HANDWH_VDI_REPRESENTATION
Representation of the handwheel number at the VDI interface</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Manual traversing"</td></tr>
</table>

TReqMinus (DB21, ... DBX40.4, 46.4, 52.4)

TReqPlus (DB21, ... DBX40.5, 46.5, 52.5)

Description	Geometry axes: Traverse request "minus" / "plus" Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].TReqMinus" (DB21, ... DBX40.4) Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].TReqPlus" (DB21, ... DBX40.5) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].TReqMinus" (DB21, ... DBX46.4) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].TReqPlus" (DB21, ... DBX46.5) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].TReqMinus" (DB21, ... DBX52.4) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].TReqPlus" (DB21, ... DBX52.5) Depending on the mode selected, the traverse request is initiated in different ways: • JOG mode "Plus" or "Minus" traversing key • REF mode Traversing key that initiates traversing motion in the direction of the reference point. • AUTOMATIC or MDI mode A program block with a traversing operation is executed for a geometry axis. There is a signal for each axis direction: • "TReqMinus" (Bit4) Traverse request "Minus" (for traversing in the negative axis direction) • "TReqPlus" (Bit5) Traverse request "Plus" (for traversing in the positive axis direction)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no traverse request available for the geometry axis.
Value TRUE	There is a traverse request available for the geometry axis.

Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].Minus" (DB21, ... DBX12.6) Geometry axis 1: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[1].Plus" (DB21, ... DBX12.7) Geometry axis 1: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[2].Minus" (DB21, ... DBX16.6) Geometry axis 2: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[2].Plus" (DB21, ... DBX16.7) Geometry axis 2: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[3].Minus" (DB21, ... DBX20.6) Geometry axis 3: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[3].Plus" (DB21, ... DBX20.7) Geometry axis 3: Traversing key "Plus" • "LBP_Chan*.E_Geo.E_Geo[1].TCMinus" (DB21, ... DBX40.6) Geometry axis 1: Traverse command "Minus" • "LBP_Chan*.E_Geo.E_Geo[1].TCPlus" (DB21, ... DBX40.7) Geometry axis 1: Traverse command "Plus" • "LBP_Chan*.E_Geo.E_Geo[2].TCMinus" (DB21, ... DBX46.6) Geometry axis 2: Traverse command "Minus" • "LBP_Chan*.E_Geo.E_Geo[2].TCPlus" (DB21, ... DBX46.7) Geometry axis 2: Traverse command "Plus" • "LBP_Chan*.E_Geo.E_Geo[3].TCMinus" (DB21, ... DBX52.6) Geometry axis 3: Traverse command "Minus" "LBP_Chan*.E_Geo.E_Geo[3].TCPlus" (DB21, ... DBX52.7) Geometry axis 3: Traverse command "Plus"
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

TCMinus (DB21, ... DBX40.6, 46.6, 52.6)

TCPlus (DB21, ... DBX40.7, 46.7, 52.7)

Description	Geometry axes: Traverse command "Minus" / "Plus" Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].TCMinus" (DB21, ... DBX40.6) Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].TCPlus" (DB21, ... DBX40.7) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].TCMinus" (DB21, ... DBX46.6) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].TCPlus" (DB21, ... DBX46.7) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].TCMinus" (DB21, ... DBX52.6) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].TCPlus" (DB21, ... DBX52.7) Depending on the setting of MD17900, bit 0, the traverse command is already output if a "Traverse request" is active (bit 0 = FALSE) or only if the axis is actually traversing (bit 0 = TRUE). There is a signal for each axis direction: • **"**TCMinus" (Bit6) Traverse command "Minus" (for traversing in the negative axis direction) • **"**TCPlus" (Bit7) Traverse command "Plus" (for traversing in the positive axis direction) **Application example** Releasing the axis clamp when the traverse command is identified. **Note** For axes where clamping is not released until a traverse command is detected, the continuous-path mode (G64) is not possible.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is not a traverse request for the geometry axis or the geometry axis is not traversing.
Value TRUE	There is a traverse request for the geometry axis or the geometry axis is traversing.

Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].Minus" (DB21, ... DBX12.6) Geometry axis 1: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[1].Plus" (DB21, ... DBX12.7) Geometry axis 1: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[2].Minus" (DB21, ... DBX16.6) Geometry axis 2: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[2].Plus" (DB21, ... DBX16.7) Geometry axis 2: Traversing key "Plus" • "LBP_Chan*.A_Geo.A_Geo[3].Minus" (DB21, ... DBX20.6) Geometry axis 3: Traversing key "Minus" • "LBP_Chan*.A_Geo.A_Geo[3].Plus" (DB21, ... DBX20.7) Geometry axis 3: Traversing key "Plus" • "LBP_Chan*.E_Geo.E_Geo[1].TReqMinus" (DB21, ... DBX40.4) Geometry axis 1: Traverse request "Minus" • "LBP_Chan*.E_Geo.E_Geo[1].TReqPlus" (DB21, ... DBX40.5) Geometry axis 1: Traverse request "Plus" • "LBP_Chan*.E_Geo.E_Geo[2].TReqMinus" (DB21, ... DBX46.4) Geometry axis 2: Traverse request "Minus" • "LBP_Chan*.E_Geo.E_Geo[2].TReqPlus" (DB21, ... DBX46.5) Geometry axis 2: Traverse request "Plus" • "LBP_Chan*.E_Geo.E_Geo[3].TReqMinus" (DB21, ... DBX52.4) Geometry axis 3: Traverse request "Minus" • "LBP_Chan*.E_Geo.E_Geo[3].TReqPlus" (DB21, ... DBX52.5) Geometry axis 3: Traverse request "Plus" • MD17900 $MN_VDI_FUNCTION_MASK Setting for VDI signals
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

INC1 (DB21, ... DBX41.0, 47.0, 53.0)

INC10 (DB21, ... DBX41.1, 47.1, 53.1)

INC100 (DB21, ... DBX41.2, 47.2, 53.2)

INC1000 (DB21, ... DBX41.3, 47.3, 53.3)

INC10000 (DB21, ... DBX41.4, 47.4, 53.4)

INCVar (DB21, ... DBX41.5, 47.5, 53.5)

Description	Geometry axes: Active machine function Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].INC*" (DB21, ... DBX41.0 ... 41.5) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].INC*" (DB21, ... DBX47.0 ... 47.5) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].INC*" (DB21, ... DBX53.0 ... 53.5) There is a signal for every machine function to manually traverse the geometry axis in the JOG mode:	
	"INC1" (Bit 0)	INC1
	"INC10" (Bit 1)	INC10
	"INC100" (Bit 2)	INC100
	"INC1000" (Bit 3)	INC1000
	"INC10000" (Bit 4)	INC10000
	"INCVar" (Bit 5)	INCvar
	Note Depending on the machine function, the response when actuating the traversing key or the handwheel differs.	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Cyclic	
Value FALSE	The machine function to manually traverse the geometry axis is not active.	
Value TRUE	The machine function to manually traverse the geometry axis is active.	
Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].INC*", "LBP_Chan*.A_Geo.A_Geo[1].ContManTravel" (DB21, ... DBX13.0 ... 13.6) Geometry axis 1: Machine function request • "LBP_Chan*.A_Geo.A_Geo[2].INC*", "LBP_Chan*.A_Geo.A_Geo[2].ContManTravel" (DB21, ... DBX17.0 ... 17.6) Geometry axis 2: Machine function request • "LBP_Chan*.A_Geo.A_Geo[3].INC*", "LBP_Chan*.A_Geo.A_Geo[3].ContManTravel" (DB21, ... DBX21.0 ... 21.6) Geometry axis 3: Machine function request	
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"	

OEM (DB21, ... DBB42, 48, 54)

Description	OEM signals
Signal flow	NC → PLC
Data type	BYTE

InvHWDirOfRot (DB21, ... DBX43.0, 49.0, 55.0)

Description	Geometry axes: Handwheel direction of rotation inversion active Geometry axis 1: "LBP_Chan*.E_Geo.E_Geo[1].InvHWDirOfRot" (DB21, ... DBX43.0) Geometry axis 2: "LBP_Chan*.E_Geo.E_Geo[2].InvHWDirOfRot" (DB21, ... DBX49.0) Geometry axis 3: "LBP_Chan*.E_Geo.E_Geo[3].InvHWDirOfRot" (DB21, ... DBX55.0)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Inversion of the direction of rotation of the handwheel assigned to the geometry axis is not active.
Value TRUE	Inversion of the direction of rotation of the handwheel assigned to the geometry axis is active.
Corresponds to	• "LBP_Chan*.A_Geo.A_Geo[1].InvHWDirOfRot" (DB21, ... DBX15.0) Geometry axis 1: Invert handwheel direction of rotation • "LBP_Chan*.A_Geo.A_Geo[2].InvHWDirOfRot" (DB21, ... DBX19.0) Geometry axis 2: Invert handwheel direction of rotation • "LBP_Chan*.A_Geo.A_Geo[3].InvHWDirOfRot" (DB21, ... DBX23.0) Geometry axis 3: Invert handwheel direction of rotation
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

10.4.9.22 STRUCT "A_Ori" Control signals for orientation axes

The variable structure "A_Ori" describes the control signals to the orientation axes.

Variables in "A_Ori"

HW1 (DB21, ... DBX320.0, 324.0, 328.0)

HW2 (DB21, ... DBX320.1, 324.1, 328.1)

HW3 (DB21, ... DBX320.2, 324.2, 328.2)

<table>
<tr><td>Description</td><td>Orientation axes: Activate handwheel

Orientation axis 1: "LBP_Chan*.A_Ori[1].HW*" (DB21, ... DBX320.0, 320.1, 320.2)
Orientation axis 2: "LBP_Chan*.A_Ori[2].HW*" (DB21, ... DBX324.0, 324.1, 324.2)
Orientation axis 3: "LBP_Chan*.A_Ori[3].HW*" (DB21, ... DBX328.0, 328.1, 328.2)

The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.

Bit coded: maximum 3 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
</table>
Bit coded: maximum 6 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Chan*.E_Ori[1].HW*" (DB21, ... DBX332.0, 332.1, 332.2)
Orientation axis 1: Handwheel active
• "LBP_Chan*.E_Ori[2].HW*" (DB21, ... DBX336.0, 336.1, 336.2)
Orientation axis 2: Handwheel active
• "LBP_Chan*.E_Ori[3].HW*" (DB21, ... DBX340.0, 340.1, 340.2)
Orientation axis 3: Handwheel active
• MD11324 $MN_HANDWH_VDI_REPRESENTATION
Representation of the handwheel number at the VDI interface</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Manual traversing"</td></tr>
</table>

FDStop (DB21, ... DBX320.3, 324.3, 328.3)

Description	Feed stop
Signal flow	PLC → NC
Data type	BOOL

Disabl (DB21, ... DBX320.4, 324.4, 328.4)

Description	Orientation axes: Traversing key disable Orientation axis 1: "LBP_Chan*.A_Ori[1].Disabl" (DB21, ... DBX320.4) Orientation axis 2: "LBP_Chan*.A_Ori[2].Disabl" (DB21, ... DBX324.4) Orientation axis 3: "LBP_Chan*.A_Ori[3].Disabl" (DB21, ... DBX328.4) **Note** When the traversing key disable is activated while traversing, then traversing is canceled.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The traversing key lock for the orientation axis is not active.
Value TRUE	The traversing key lock for the orientation axis is active.
Corresponds to	• "LBP_Chan*.A_Ori[1].Minus" (DB21, ... DBX320.6) Orientation axis 1: Traversing keys "Minus" • "LBP_Chan*.A_Ori[1].Plus" (DB21, ... DBX320.7) Orientation axis 1: Traversing keys "Plus" • "LBP_Chan*.A_Ori[2].Minus" (DB21, ... DBX324.6) Orientation axis 2: Traversing keys "Minus" • "LBP_Chan*.A_Ori[2].Plus" (DB21, ... DBX324.7) Orientation axis 2: Traversing keys "Plus" • "LBP_Chan*.A_Ori[3].Minus" (DB21, ... DBX328.6) Orientation axis 3: Traversing keys "Minus" • "LBP_Chan*.A_Ori[3].Plus" (DB21, ... DBX328.7) Orientation axis 3: Traversing keys "Plus"
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

RapidTrOR (DB21, ... DBX320.5, 324.5, 328.5)

Description	Orientation axes: Rapid traverse override Orientation axis 1: "LBP_Chan*.A_Ori[1].RapidTrOR" (DB21, ... DBX320.5) Orientation axis 2: "LBP_Chan*.A_Ori[2].RapidTrOR" (DB21, ... DBX324.5) Orientation axis 3: "LBP_Chan*.A_Ori[3].RapidTrOR" (DB21, ... DBX328.5) The signal is only effective for continuous or incremental manual traversing in the JOG mode. The rapid traverse velocity can be influenced using the rapid traverse override switch.
Signal flow	PLC → NC
Data type	BOOL

Update	Cyclic
Value FALSE	When manually traversing the orientation axis using the traversing keys, the entered JOG velocity becomes active (SD41110 or MD32020).
Value TRUE	When manually traversing the orientation axis using the traversing keys, when the interface signal is set, rapid traverse velocity becomes effective (MD32010).
Corresponds to	• "LBP_Chan*.A_Ori[1].Minus" (DB21, ... DBX320.6) Orientation axis 1: Traversing keys "Minus" • "LBP_Chan*.A_Ori[1].Plus" (DB21, ... DBX320.7) Orientation axis 1: Traversing keys "Plus" • "LBP_Chan*.A_Ori[2].Minus" (DB21, ... DBX324.6) Orientation axis 2: Traversing keys "Minus" • "LBP_Chan*.A_Ori[2].Plus" (DB21, ... DBX324.7) Orientation axis 2: Traversing keys "Plus" • "LBP_Chan*.A_Ori[3].Minus" (DB21, ... DBX328.6) Orientation axis 3: Traversing keys "Minus" • "LBP_Chan*.A_Ori[3].Plus" (DB21, ... DBX328.7) Orientation axis 3: Traversing keys "Plus" • MD32010 $MA_JOG_VELO_RAPID Conventional rapid traverse • MD32020 $MA_JOG_VELO Conventional axis velocity • SD41110 $SN_JOG_SET_VELO Axis velocity for JOG
Additional references	• Function Manual Axes and Spindles; Section "Feedrates" • Function Manual Axes and Spindles; Section "Manual traversing"

Minus (DB21, ... DBX320.6, 324.6, 328.6)

Plus (DB21, ... DBX320.7, 324.7, 328.7)

Description	Orientation axes: Traversing keys "Minus" / "Plus" Orientation axis 1: "LBP_Chan*.A_Ori[1].Minus" (DB21, ... DBX320.6) Orientation axis 1: "LBP_Chan*.A_Ori[1].Plus" (DB21, ... DBX320.7) Orientation axis 2: "LBP_Chan*.A_Ori[2].Minus" (DB21, ... DBX324.6) Orientation axis 2: "LBP_Chan*.A_Ori[2].Plus" (DB21, ... DBX324.7) Orientation axis 3: "LBP_Chan*.A_Ori[3].Minus" (DB21, ... DBX328.6) Orientation axis 3: "LBP_Chan*.A_Ori[3].Plus" (DB21, ... DBX328.7) There is a request signal for every traversing key and/or axis direction: • Minus (Bit 6) Traversing key "Minus" (for traversing in the negative axis direction) • Plus (Bit 7) Traversing key "Plus" (for traversing in the positive axis direction) Depending on the active machine function, as well as the settings for jog and continuous operation (SD41050 and MD11300), different responses are initiated when the signal changes: 1. Continuous manual travel in the jog mode The orientation axis traverses in the direction concerned to the active limit switch as long as the interface signal is set to TRUE. 2. Continuous manual travel in continuous operation At the first edge change 0 → 1, the orientation axis starts to traverse in the appropriate direction. This traversing movement still continues when the edge changes from 1 → 0. Any new signal edge change 0 → 1 (same traversing direction!) stops the traversing movement. 3. Incremental manual travel in the jog mode With signal state TRUE, the orientation axis starts to traverse the set increment. If the signal changes to the FALSE state before the increment is traversed, then traversing is interrupted. When the signal state changes to TRUE again, traversing is continued. The orientation axis can be stopped and started several times as described above until it has traversed the complete increment. 4. Incremental manual travel in continuous operation At the first edge change 0 → 1, the orientation axis starts to traverse the set increment. If another edge change 0 → 1 is performed with the same traverse signal before the orientation axis has traversed the increment, the traversing movement will be canceled. The increment is no longer traversed to the end. **Note** • The interface signal is irrelevant in operating modes: – AUTOMATIC

	– MDI • If both traversing signals ("plus" and "minus") are set at the same time there is no traversing or the current traversing is aborted. • In contrast to machine axes, for orientation axes, only one orientation axis can be traversed at any one time using the traversing keys. • Via "LBP_Chan*.A_Ori[i].Disabl", Index 1 ... 3 (DB21, ... DBX320.4, 324.4, 328.4) (Orientation axis 1, 2, 3: Traversing key lock) traversing using traversing keys can be locked. • In the following cases, the orientation axis cannot be traversed in the JOG operating mode: – The orientation axis is already being traversed via the axis-specific interface as machine axis. – If another orientation axis is already being traversed with the traversing keys.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to traverse the orientation axis using the traversing key in the positive/negative axis direction.
Value TRUE	A request has been made to traverse the orientation axis using the traversing key in the positive/negative axis direction.
Corresponds to	• "LBP_Chan*.A_Ori[1].Disabl" (DB21, ... DBX320.4) Orientation axis 1: Traversing key disable • "LBP_Chan*.A_Ori[2].Disabl" (DB21, ... DBX324.4) Orientation axis 2: Traversing key disable • "LBP_Chan*.A_Ori[3].Disabl" (DB21, ... DBX328.4) Orientation axis 3: Traversing key disable • "LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) Traversing keys "Minus" / "Plus" • MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD INC and REF in jog mode • SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD Jog/continuous operation for JOG, continuous
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

INC1 (DB21, ... DBX321.0, 325.0, 329.0)

INC10 (DB21, ... DBX321.1, 325.1, 329.1)

INC100 (DB21, ... DBX321.2, 325.2, 329.2)

INC1000 (DB21, ... DBX321.3, 325.3, 329.3)

INC10000 (DB21, ... DBX321.4, 325.4, 329.4)

INCVar (DB21, ... DBX321.5, 325.5, 329.5)

<table>
<tr><td>Description</td><td>Orientation axes: Machine function request

There is a request signal for every machine function to manually traverse the orientation axis in the JOG mode:
<table>
<tr><td>"INC1" (Bit 0)</td><td>INC1</td></tr>
<tr><td>"INC10" (Bit 1)</td><td>INC10</td></tr>
<tr><td>"INC100" (Bit 2)</td><td>INC100</td></tr>
<tr><td>"INC1000" (Bit 3)</td><td>INC1000</td></tr>
<tr><td>"INC10000" (Bit 4)</td><td>INC10000</td></tr>
<tr><td>"INCVar" (Bit 5)</td><td>INCvar</td></tr>
</table>
Incremental manual travel
In addition to five fixed increment sizes (default setting in MD11330: INC1, INC10, INC100, INC1000 and INC10000), a variable increment size (INCvar) that can be set via the setting data SD41010 is also available. The distance evaluation of one increment for fixed and variable increment sizes is performed via the axis-specific machine data MD31090.
When pressing the "Plus" or "Minus" traversing key, or by rotating the electronic handwheel, the orientation axis starts to traverse in the appropriate direction, corresponding to the number of increments of the active machine function.
Continuous manual travel
For continuous manual travel, the plus and minus traversing keys are selected to move the orientation axis continuously and the appropriate direction.

Notes
• If several requests are set simultaneously, no machine function becomes active.
• If an orientation axis is currently being traversed via a machine function, the movement is aborted through deselection or change of the machine function.</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Value FALSE</td><td>A request has not been made to manually traverse the orientation axis.</td></tr>
<tr><td>Value TRUE</td><td>A request has been made to manually traverse the orientation axis.</td></tr>
</table>

Corresponds to	• "LBP_Chan*.E_Ori[1].INC*" (DB21, ... DBX333.0 ... 333.5) Orientation axis 1: Active machine function • "LBP_Chan*.E_Ori[2].INC*" (DB21, ... DBX337.0 ... 337.5) Orientation axis 2: Active machine function • "LBP_Chan*.E_Ori[3].INC*" (DB21, ... DBX341.0 ... 341.5) Orientation axis 3: Active machine function • MD11320 $MN_HANDWH_IMP_PER_LATCH Handwheel pulses per detent position • MD11330 $MN_JOG_INCR_SIZE_TAB Increment size for INC/handwheel • MD31090 $MA_JOG_INCR_WEIGHT Evaluation of an increment for INC/handwheel • SD41010 $SN_JOG_VAR_INCR_SIZE Size of the variable increment for JOG
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

OEM (DB21, ... DBB322, 326, 330)

Description	OEM signals
Signal flow	PLC → NC
Data type	BYTE

InvHWDirOfRot (DB21, ... DBX323.0, 327.0, 331.0)

Description	Orientation axes: Invert handwheel direction of rotation Orientation axis 1: "LBP_Chan*.A_Ori[1].InvHWDirOfRot" (DB21, ... DBX323.0) Orientation axis 2: "LBP_Chan*.A_Ori[2].InvHWDirOfRot" (DB21, ... DBX327.0) Orientation axis 3: "LBP_Chan*.A_Ori[3].InvHWDirOfRot" (DB21, ... DBX331.0) **Note** It is only permissible to change the interface signal when the orientation axis is at a standstill. **Application examples** • The handwheel direction of rotation should match the axis direction of motion. • A handwheel is assigned to several axes with different orientations.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to invert the direction of rotation of the assigned handwheel.
Value TRUE	A request has been made to invert the direction of rotation of the assigned handwheel.
Corresponds to	• "LBP_Chan*.E_Ori[1].InvHWDirOfRot" (DB21, ... DBX335.0) Orientation axis 1: Handwheel direction of rotation inversion active • "LBP_Chan*.E_Ori[2].InvHWDirOfRot" (DB21, ... DBX339.0) Orientation axis 2: Handwheel direction of rotation inversion active • "LBP_Chan*.E_Ori[3].InvHWDirOfRot" (DB21, ... DBX343.0) Orientation axis 3: Handwheel direction of rotation inversion active
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

10.4.9.23 STRUCT "E_Ori" control signals of the orientation axis

The variable structure "E_Ori" describes the control signals of the orientation axis.

Variables in "E_Ori"

HW1 (DB21, ... DBX332.0, 336.0, 340.0)

HW2 (DB21, ... DBX332.1, 336.1, 340.1)

HW3 (DB21, ... DBX332.2, 336.2, 340.2)

<table>
<tr><td>Description</td><td>Orientation axes: Handwheel active

Orientation axis 1: "LBP_Chan*.E_Ori[1].HW*" (DB21, ... DBX332.0, 332.1, 332.2)
Orientation axis 2: "LBP_Chan*.E_Ori[2].HW*" (DB21, ... DBX336.0, 336.1, 336.2)
Orientation axis 3: "LBP_Chan*.E_Ori[3].HW*" (DB21, ... DBX340.0, 340.1, 340.2)

The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.

Bit coded: maximum 3 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
</table>
Bit coded: maximum 6 handwheels
<table>
<tr><th>HW3
Bit 2</th><th>HW2
Bit 1</th><th>HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, an axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>NC → PLC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
</table>

Corresponds to	• "LBP_Chan*.A_Ori[1].HW*" (DB21, ... DBX320.0, 320.1, 320.2) Orientation axis 1: Activate handwheel • "LBP_Chan*.A_Ori[2].HW*" (DB21, ... DBX324.0, 324.1, 324.2) Orientation axis 2: Activate handwheel • "LBP_Chan*.A_Ori[3].HW*" (DB21, ... DBX328.0, 328.1, 328.2) Orientation axis 3: Activate handwheel • MD11324 $MN_HANDWH_VDI_REPRESENTATION Representation of the handwheel number at the VDI interface
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

TReqMinus (DB21, ... DBX332.4, 336.4, 340.4)

TReqPlus (DB21, ... DBX332.5, 336.5, 340.5)

Description	Orientation axes: Traverse request "minus" / "plus" Orientation axis 1: "LBP_Chan*.E_Ori[1].TReqMinus" (DB21, ... DBX332.4) Orientation axis 1: "LBP_Chan*.E_Ori[1].TReqPlus" (DB21, ... DBX332.5) Orientation axis 2: "LBP_Chan*.E_Ori[2].TReqMinus" (DB21, ... DBX336.4) Orientation axis 2: "LBP_Chan*.E_Ori[2].TReqPlus" (DB21, ... DBX336.5) Orientation axis 3: "LBP_Chan*.E_Ori[3].TReqMinus" (DB21, ... DBX340.4) Orientation axis 3: "LBP_Chan*.E_Ori[3].TReqPlus" (DB21, ... DBX340.5) The traverse request is initiated in different ways depending on the operating mode: • JOG mode "Plus" or "Minus" traversing key • REF mode Traversing key that initiates traversing motion in the direction of the reference point. • AUTOMATIC or MDI mode A program block with a traversing operation is executed for an orientation axis. There is a signal for each axis direction: • TReqMinus (Bit 4) Traverse request "Minus" (for traversing in the negative axis direction) • TReqPlus (Bit 5) Traverse request "Plus" (for traversing in the positive axis direction)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no traverse request available for the orientation axis.
Value TRUE	There is a traverse request available for the orientation axis.

Corresponds to	• "LBP_Chan*.A_Ori[1].Minus" (DB21, ... DBX320.6) Orientation axis 1: Traversing keys "Minus" • "LBP_Chan*.A_Ori[1].Plus" (DB21, ... DBX320.7) Orientation axis 1: Traversing keys "Plus" • "LBP_Chan*.A_Ori[2].Minus" (DB21, ... DBX324.6) Orientation axis 2: Traversing keys "Minus" • "LBP_Chan*.A_Ori[2].Plus" (DB21, ... DBX324.7) Orientation axis 2: Traversing keys "Plus" • "LBP_Chan*.A_Ori[3].Minus" (DB21, ... DBX328.6) Orientation axis 3: Traversing keys "Minus" • "LBP_Chan*.A_Ori[3].Plus" (DB21, ... DBX328.7) Orientation axis 3: Traversing keys "Plus" • "LBP_Chan*.E_Ori[1].TCMinus" (DB21, ... DBX332.6) Orientation axis 1: Traversing keys "Minus" • "LBP_Chan*.E_Ori[1].TCPlus" (DB21, ... DBX332.7) Orientation axis 1: Traversing keys "Plus" • "LBP_Chan*.E_Ori[2].TCMinus" (DB21, ... DBX336.6) Orientation axis 2: Traversing keys "Minus" • "LBP_Chan*.E_Ori[2].TCPlus" (DB21, ... DBX336.7) Orientation axis 2: Traversing keys "Plus" • "LBP_Chan*.E_Ori[3].TCMinus" (DB21, ... DBX340.6) Orientation axis 3: Traversing keys "Minus" • "LBP_Chan*.E_Ori[3].TCPlus" (DB21, ... DBX340.7) Orientation axis 3: Traversing keys "Plus"
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

TCMinus (DB21, ... DBX332.6, 336.6, 340.6)

TCPlus (DB21, ... DBX332.7, 336.7, 340.7)

Description	Orientation axes: Traverse command "Minus" / "Plus" Orientation axis 1: "LBP_Chan*.E_Ori[1].TCMinus" (DB21, ... DBX332.6) Orientation axis 1: "LBP_Chan*.E_Ori[1].TCPlus" (DB21, ... DBX332.7) Orientation axis 2: "LBP_Chan*.E_Ori[2].TCMinus" (DB21, ... DBX336.6) Orientation axis 2: "LBP_Chan*.E_Ori[2].TCPlus" (DB21, ... DBX336.7) Orientation axis 3: "LBP_Chan*.E_Ori[3].TCMinus" (DB21, ... DBX340.6) Orientation axis 3: "LBP_Chan*.E_Ori[3].TCPlus" (DB21, ... DBX340.7) Depending on the setting of MD17900, bit 0, the traverse command is already output if a "Traverse request" is active (bit 0 = FALSE) or only if the axis is actually traversing (bit 0 = TRUE). There is a signal for each axis direction: • TCMinus (Bit 6) Traverse command "Minus" (for traversing in the negative axis direction) • TCPlus (Bit 7) Traverse command "Plus" (for traversing in the positive axis direction) **Application example** Releasing the axis clamp when the traverse command is identified. **Note** For axes where clamping is not released until a traverse command is detected, the continuous-path mode (G64) is not possible.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no traverse request for the orientation axis or the orientation axis is not traversing.
Value TRUE	There is a traverse request for the orientation axis or the orientation axis is traversing.

Corresponds to	• "LBP_Chan*.A_Ori[1].Minus" (DB21, ... DBX320.6) Orientation axis 1: Traversing keys "Minus" • "LBP_Chan*.A_Ori[1].Plus" (DB21, ... DBX320.7) Orientation axis 1: Traversing keys "Plus" • "LBP_Chan*.A_Ori[2].Minus" (DB21, ... DBX324.6) Orientation axis 2: Traversing keys "Minus" • "LBP_Chan*.A_Ori[2].Plus" (DB21, ... DBX324.7) Orientation axis 2: Traversing keys "Plus" • "LBP_Chan*.A_Ori[3].Minus" (DB21, ... DBX328.6) Orientation axis 3: Traversing keys "Minus" • "LBP_Chan*.A_Ori[3].Plus" (DB21, ... DBX328.7) Orientation axis 3: Traversing keys "Plus" • "LBP_Chan*.E_Ori[1].TReqMinus" (DB21, ... DBX332.4) Orientation axis 1: Traverse request "Minus" • "LBP_Chan*.E_Ori[1].TReqPlus" (DB21, ... DBX332.5) Orientation axis 1: Traverse request "Plus" • "LBP_Chan*.E_Ori[2].TReqMinus" (DB21, ... DBX336.4) Orientation axis 2: Traverse request "Minus" • "LBP_Chan*.E_Ori[2].TReqPlus" (DB21, ... DBX336.5) Orientation axis 2: Traverse request "Plus" • "LBP_Chan*.E_Ori[3].TReqMinus" (DB21, ... DBX340.4) Orientation axis 3: Traverse request "Minus" • "LBP_Chan*.E_Ori[3].TReqPlus" (DB21, ... DBX340.5) Orientation axis 3: Traverse request "Plus" • MD17900 $MN_VDI_FUNCTION_MASK Setting for VDI signals
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

INC1 (DB21, ... DBX333.0, 337.0, 341.0)

INC10 (DB21, ... DBX333.1, 337.1, 341.1)

INC100 (DB21, ... DBX333.2, 337.2, 341.2)

INC1000 (DB21, ... DBX333.3, 337.3, 341.3)

INC10000 (DB21, ... DBX333.4, 337.4, 341.4)

INCVar (DB21, ... DBX333.5, 337.5, 341.5)

Description	Orientation axes: Active machine function There is a request signal for every machine function to manually traverse the orientation axis in the JOG mode:	
	"INC1" (Bit 0)	INC1
	"INC10" (Bit 1)	INC10
	"INC100" (Bit 2)	INC100
	"INC1000" (Bit 3)	INC1000
	"INC10000" (Bit 4)	INC10000
	"INCVar" (Bit 5)	INCvar
	Note Depending on the machine function, the response when actuating the traversing key or the handwheel differs.	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Cyclic	
Value FALSE	The machine function to manually traverse the orientation axis is not active.	
Value TRUE	The machine function to manually traverse the orientation axis is active.	
Corresponds to	• "LBP_Chan*.A_Ori[1].INC*" (DB21, ... DBX321.0 ... 321.5) Orientation axis 1: Machine function request • "LBP_Chan*.A_Ori[2].INC*" (DB21, ... DBX325.0 ... 325.5) Orientation axis 2: Machine function request • "LBP_Chan*.A_Ori[3].INC*" (DB21, ... DBX329.0 ... 329.5) Orientation axis 3: Machine function request	
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"	

OEM (DB21, ... DBB334, 338, 342)

Description	OEM signals
Signal flow	NC → PLC
Data type	BYTE

InvHWDirOfRot (DB21, ... DBX335.0, 339.0, 343.0)

Description	Handwheel direction of rotation inversion active
Signal flow	NC → PLC
Data type	BOOL

10.4.10 LBP_Axis 1 [DB31], ... - Axis/spindle signals

10.4.10.1 Overview

LBP_Axis1 [DB31], ... - Signals to the axis/spindle

Table 10-41 "LBP_Axis1 [DB31], ...", signals to the axis/spindle

LBP_Axis1 [DB31], ..."	**Signals to the axis/spindle (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Feedrate override							
Axis/spindle	H	G	F	E	D	C	B	A
DBB1 Axis/spindle	Override active	Position measuring system 1 / 2		Follow-up mode	Axis/spindle disable	Sensor for fixed stop	Acknowledge fixed stop reached	Drive test travel enable
		2	1					
DBB2 Axis/spindle	Reference point value 1-4				Clamping in progress	Delete distance-to-go / spindle reset	Controller enable	Software cams: Activation
	4	3	2	1				
DBB3 Axis/spindle	Program test Axis/ Spindle Enable	Velocity / spindle speed limitation	Activate fixed feedrate				Travel to fixed stop enabled	Accept offset external WO
			4	3	2	1		
DBB4 Axis/spindle	Traversing keys		Rapid traverse override	Traversing key disable	Feed stop/ spindle stop	Activate handwheel		
	Plus	Minus				C	B	A
DBB5 Axis/spindle	Machine function request							
		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1
DBB6	OEM signals							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB7	OEM signals							Handwheel direction of rotation inverted
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	
DBB8	Request PLC axis/ spindle			Channel assignment changed	Axis/spindle exchange request			
					D	C	B	A
DBB9					Parameter set changeover disabled	Selection: Position controller parameter set		
						C	B	A
DBB10								REPOS delay
DBB11								SI: Start brake test

LBP_Axis1 [DB31], ..."	**Signals to the axis/spindle (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB12 Axis	Reference point approach delay			Modulo rotary axes: Activate traversing range limits	2nd software limit switch /Z1-A3/		Hardware limit switch /Z1-A3/	
					Plus	Minus	Plus	Minus
DBB13 Axis					JOG to position	JOG fixed point approach		
						2	1	0
DBB14 Axis							Activate program test	Suppress program test
DBB15 Axis								
DBB16 Spindle	S value Delete	No speed monitoring during gear change	Resynchronize spindle		Gear is changed over	Actual gear stage		
			Measuring system 2	Measuring system 1		Bit 2	Bit 1	Bit 0
DBB17 Spindle		Invert M3/M4	Resynchronize spindle during positioning					Feedrate correction spindle valid
			Measuring system 2	Measuring system 1				
DBB18 Spindle	Oscillation direction of rotation		Oscillation enable	Oscillation controlled by the PLC				
	Links	Right						
DBB19 Spindle	Speed override, spindle-specific							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB20 Drive			Open motor holding brake				Ramp-function generator disable [1)]	
DBB21 Drive	Pulse enable	Integrator inhibit, speed controller	Motor being selected	Motor/drive data set: Request interface				
				E	D	C	B	A
DBB22								
DBB23								
DBB24	Master/ slave: Activate coupling		Setpoint exchange: Request drive control	Master/ slave: Activate torque equalization controller	MCS coupling: Activate collision protection	MCS coupling: Deactivate or do not permit	Control axis	Stepper motor: Rotation monitoring
DBB25								Dynam. backlash compensation activation

LBP_Axis1 [DB31], ..."	**Signals to the axis/spindle (PLC → NC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB26 Grinding				Following axis superimposition: Enable	Compensatory controller on			
DBB27 Grinding	Stop				Resume			
	HIAxMove	Corr	DEPBCS	DEPMCS	HIAxMove	Corr	DEPBCS	DEPMCS
DBB28 Oscillation grinding	Request PLC-controlled axis	PLC-controlled axis: Stop along braking ramp	PLC-controlled axis: Stop at the next reversal point	Change reversal point	Set reversal point	PLC-controlled axis: Continue	PLC-controlled axis: Reset	Oscillation reversal from external
DBB29 Couplings			Disable automatic synchronization	Start gantry synchronization				
DBB30 Technology				Spindle motion via the PLC interface				
				Spindle-start positioning	Select gear stage	Spindle start, counter-clockwise rotation	Spindle start, clockwise rotation	Spindle stop
DBB31 Technology	Delete synchronism correction	Track synchronism	Disable synchronization	Re-synchronization				
DBB32								
DBB33								
DBB34							Setpoint limitation	Setpoint limitation
DBB35								
DBB36 Technology								
DBB37								
DBB38								
DBB39								
DBB40 - DBB55								
DBB56 PLC → operating software						Spindle internal clamping	Spindle speed display	Separate feed drive coupled as C axis
DBB57								
DBB58								
DBB59								

LBP_Axis1 [DB31], ..."	Signals to the axis/spindle (PLC → NC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB188								
DBB189							Motor overtemperature: disable current reduction	DYNEGMA: Disable power reduction

[1] Only when cyclic interface between NC and drive is operated in the "611U compatibility mode".

Note

DBX8.4 is automatically reset after the assignment is executed

LBP_Axis1 [DB31], ... - Signals from the axis/spindle

Table 10-42 "LBP_Axis1 [DB31], ...", signals from the axis/spindle

LBP_Axis1 [DB31], ...	Signals from the axis/spindle (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB60 Axis/spindle	Position reached with exact stop		Referenced/synchronized Position measuring system		Encoder limit frequency exceeded, measuring system		NCU link: Axis active	Spindle / rotary axis
	fine	coarse	2	1	2	1		
DBB61 Axis/spindle	Current controller active	Speed controller active	Position controller active	Axis/spindle stationary ($n < n_{min}$)	Follow-up mode active	Axis ready	Axis-specific alarm	Drive test traverse request
DBB62	Axis container: Rotation active	Travel to fixed stop: Force limiting active	Travel to fixed stop: Fixed stop reached	Travel to fixed stop: Activate function	Measurement active	Revolutional feedrate active	Handwheel override active	Software cams active
DBB63	Stop				Axis/spindle inhibit active	Axis stop active	PLC-controlled axis	Reset executed
	HIAxMove active	Corr. active	DEPBCS active	DEPMCS active				
DBB64	Traverse command		Traverse request			Handwheel active		
Axis/spindle	Plus	Minus	Plus	Minus		Bit 2	Bit 1	Bit 0
DBB65 Axis/spindle		Active machine function						
		Continuous traversing	INCvar	INC10000	INC1000	INC100	INC10	INC1

LBP_Axis1 [DB31], ...	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB66 Axis/spindle								MCS coupling: Collision protection active
DBB67								Handwheel direction of rotation inversion active
DBB68	Status axis/spindle exchange							
	PLC axis/ spindle	Neutral axis/spindle	Axis interchange possible	New type requested from PLC	NC axis / spindle channel assignment			
					Bit 3	Bit 2	Bit 1	Bit 0
DBB69	NCU number in the NCU link group					Controller parameter set servo		
	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0
DBB70			DRV Safety Integrated with SIC/ SCC active			REPOS delay acknowledgement	REPOS offset valid	REPOS offset
DBB71	PLC axis permanently assigned		Position restored					Brake test active
			Encoder 2	Encoder 1				
DBB72 Operating software → PLC								REPOS delay active
DBB73 Operating software → PLC								
DBB74 Axis				Modulo rotary axes: Traversing range limits active				
DBB75 Axis	JOG position reached	JOG travel to position active	JOG approach fixed point reached			JOG approach fixed point active		
			Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0
DBB76 Axis	Rounding axis in position	Indexing axis in position	Positioning axis	Path axis				Lubrication pulse

LBP_Axis1 [DB31], ...	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB77 Axis								Collision avoidance: Velocity reduction
DBD78 Axis	F value (REAL) for positioning axis							
DBB82 Spindle					Change gear stage	Set gear stage		
						Bit 2	Bit 1	Bit 0
DBB83 Spindle	Actual direction of rotation clockwise	Speed monitoring	Spindle in setpoint range	Support range limits violated	Geometry monitoring	Setpoint speed		Speed limit exceeded
						Increased	limited	
DBB84 Spindle	Active spindle mode				Tapping without compensating chuck active	CLGON active	GWPS active	Const. cutting speed active
	Control mode	Oscillation mode	Positioning mode	Synchronous mode				
DBB85 Spindle			Spindle actually reached position					Tool with dynamic response limitation
DBW86 Spindle	M function (INT) for spindle (M3, M4, M5, M19, M70 or defined via MD)							
DBD88 Spindle	S function (REAL) for spindle							
DBB92 Drive	Drive operation enabled		Motor holding brake opened	Drive-autonomous motion active[1)]			Ramp-function generator disable active	
DBB93 Drive	Pulses enabled	Speed controller integrator disabled	Drive ready	Motor/drive data set: Display interface				
				Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB94 Drive	Variable signaling function [2)]	$n_{act} = n_{set}$	$\|n_{act}\| < n_x$	$\|n_{act}\| < n_{min}$	$M_d < M_{dx}$	Run-up completed	Temperature pre-alarm	
							Heat sink	Motor
DBB95 Drive	Alarm of alarm class C is active				ESR: Regenerative operation speed lower than minimum (p2161)	ESR: Reaction triggered or generator operation active (r0887.12)	ESR: DC link undervoltage (p1248)	UDC connection < Alarm threshold
DBB96	Master/slave: Coupling active		Setpoint exchange: Drive control active	Master/slave:			Control axis active	Stepper motor: Rotation monitoring error
				Compensatory controller active	Coarse speed difference	Fine speed difference		

LBP_Axis1 [DB31], ...	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB97					MCS coupling:			
					Offset change	Activate mirroring	Coupling active	Following axis
DBB98 Synchronous spindle	ESR response initiated	Acceleration warning threshold reached	Velocity warning threshold reached	Overlaid motion		Actual value coupling	Synchronous operation	
							coarse	fine
DBB99 Synchronous spindle		Max. acceleration reached	Max. velocity reached	Synchronization running	Axis accelerating	Synchronism correction implemented	Following spindle active	Leading spindle active
DBB100 Oscillation grinding	Oscillation active	Oscillation active	Sparking-out active	Fault during oscillation motion	Oscillation cannot be started	External oscillation reversal active		
DBB101 Gantry	Gantry axis	Gantry guide axis	Gantry grouping is synchronous	Gantry synchronization ready to start	Gantry warning limit exceeded	Gantry shutdown exceeded		
DBB102		Position measuring system activated			Clamping tolerance exceeded			Dynam. backlash compensation active
		2	1					
DBB103			Synchronism 2					Synchronism correction is taken into account
			coarse	fine				
DBB104	Active infeed axis							
Grinding	8	7	6	5	4	3	2	1
DBB105	Active infeed axis							
Grinding	16	15	14	13	12	11	10	9
DBB106	Active infeed axis							
Grinding	24	23	22	21	20	19	18	17
DBB107	Active infeed axis							
Grinding		31	30	29	28	27	26	25
DBB108-127								
DBB128							Program test	
Operating software → PLC							activate	suppress
DBB129								

LBP_Axis1 [DB31], ...	**Signals from the axis/spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB130	Motor/drive data set: Request and display interface valid			Motor/drive data set: Formatting interface				
				Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB131								
DBB132	Sensor configuration							
Weiss spindle		Sensor S6 available	Sensor S5 available (angular position, motor shaft)	Sensor S4 available (piston end position)			Sensor S1 available (clamped state)	Sensors available
DBB133	Sensor configuration							
Weiss spindle						Status value is generated, speed limitation p5043 is active		
DBW134 Weiss spindle	Clamping state (sensor S1) status value							
DBW136 Weiss spindle	Clamping state (sensor S1) analog value							
DBB138 Weiss spindle			Sensor S5 angular position, motor shaft	Sensor S4, piston end position				
DBB139 Weiss spindle	Status digital sensors							
DBB190								Encoder servicing required
DBB191	OR operation: Condition fulfilled	AND operation: Condition fulfilled			Pole position identification with encoder successfully carried out	Motor overtemperature		DYNEGMA: Torque limit applied
						Motor temperature model warning active	Current reduction active	

1) With SINAMICS valid as of NC 62.07 when using a 611U telegram type

2) With SINAMICS valid as of SW2.6

10.4.10.2 LBP_Axis1 [DB31], ... - Signals to the axis/spindle

Data blocks "LBP_Axis1 [DB31], ..." contain the axis and spindle signals.

This section describes the first part of the data block containing the signals to the axis and spindle.

Variables in "LBP_Axis1 [DB31], ..." - signals to the axis/spindle

A_FD_OR (DB31, ... DBB0)

Description

Feedrate override, axis-specific

Binary/Gray coding

The override factors can be specified in the binary or Gray-coded format. The control must be informed of the used format via the following machine data:

MD12000 $MN_OVR_AX_IS_GRAY_CODE = <Coding>

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:
MD12010 $MN_OVR_FACTOR_AX_SPEED[<switch position> - 1] = <correction factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.00
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.05
21	11111	1.10
22	11101	1.15

	23	11100	1.20
	24	10100	1.20
	25	10101	1.20
	26	10111	1.20
	27	10110	1.20
	28	10010	1.20
	29	10011	1.20
	30	10001	1.20
	31	10000	1.20
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN **Notes** The axis-specific feedrate override is **not effective** with the following functions when cutting threads: G33, G331, G332, G63		
Signal flow	PLC → NC		
Data type	BYTE		
Update	Cyclic		
Corresponds to	• "LBP_Chan*.A_FD_OR" (DB21, ... DBB4) Path feedrate override • "LBP_Axis*.A_SpOR" (DB31, ... DBB19) Speed override, spindle-specific • "LBP_Axis*.A_ORactive" (DB31, ... DBX1.7) Override active		
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"		

1) Default values

A_DriveEnable (DB31, ... DBX1.0)

Description	Drive test travel enable The signal is the feedback signal for the request: "LBP_Axis*.E_TravRequ" (DB31, ... DBX61.0) == 1 (drive test traverse request) Only the PLC can enable an axis so that it can be traversed.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The travel enable for the drive test is not set, i.e. axis motion for the drive test is inhibited.
Value TRUE	The travel enable for the drive test is set, i.e. axis motion for the drive test is enabled.
Corresponds to	• "LBP_Axis*.E_TravRequ" (DB31, ... DBX61.0) Drive test traverse request

A_AckFixedStop (DB31, ... DBX1.1)

Description	Acknowledge fixed stop reached Irrelevant for MD37060 $MA_FIXED_STOP_ACKN_MASK == 0 or 2
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Meaning when the fixed stop is reached: "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) (fixed stop reached) == 1 • The axis presses against the fixed stop with the clamping torque • The fixed stop monitoring window is activate • A block change is not executed, and the channel message "Wait: Aux. fct. acknowledgment missing" is displayed • The function is canceled, alarm "20094 axis %1 Function aborted" is output Meaning after deselection of the function `FXS=0` via the part program: • The torque limiting and the monitoring of the fixed stop monitoring window are canceled.
Value TRUE	Meaning when the fixed stop is reached: "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) (fixed stop reached) == 1 • The axis presses against the fixed stop with the clamping torque • The fixed stop monitoring window is activated. • A block change is executed.
Corresponds to	• "LBP_Axis*.A_SensorFixedStop" (DB31, ... DBX1.2) Fixed stop sensor • "LBP_Axis*.A_EnabTravFixedStop" (DB31, ... DBX3.1) Enable travel to fixed stop • "LBP_Axis*.E_ActTravFStop" (DB31, ... DBX62.4) Activate travel to fixed stop • "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) Fixed stop reached • MD37060 $MA_FIXED_STOP_ACKN_MASK Monitoring PLC acknowledgements for travel to fixed stop
Additional references	Function Manual Axes and Spindles, Section "Travel to fixed stop"

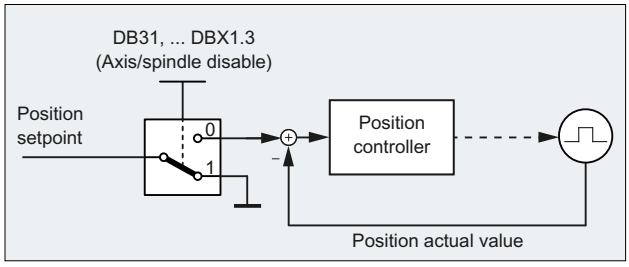
A_SensorFixedStop (DB31, ... DBX1.2)

Description	Sensor for fixed stop The signal is only active if: MD37040 $MA_FIXED_STOP_BY_SENSOR == 1
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Fixed stop has not been reached.
Value TRUE	Fixed stop has been reached.

Corresponds to	• "LBP_Axis*.A_AckFixedStop" (DB31, ... DBX1.1) Acknowledge fixed stop reached • "LBP_Axis*.A_EnabTravFixedStop" (DB31, ... DBX3.1) Enable travel to fixed stop • "LBP_Axis*.E_ActTravFStop" (DB31, ... DBX62.4) Activate travel to fixed stop • "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) Fixed stop reached • MD37040 $MA_FIXED_STOP_BY_SENSOR Fixed stop detection via sensor
Additional references	Function Manual Axes and Spindles, Section "Travel to fixed stop"

A_ASpDisable (DB31, ... DBX1.3)

Description	Axis/spindle disable
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is not request for an "axis/spindle disable". **Note** The reset of the signal does not take effect until the axis/spindle is stationary. • **Axis** Setpoints are output to the position controller again immediately after the signal has been reset. If a traverse request is active, it is executed immediately. • **Spindle** After the reset of the signal, the spindle disable which is still effective internally must be canceled by an axis-specific reset, channel reset or end of program (M30 / M2). Only then are setpoints output to the speed controller again.

Value TRUE	Axis/spindle disable is requested. **Axis** Effects when the axis is stationary: • No setpoints are output to the position controller, i.e. the axis traversing movement is disabled. The axis is in closed-loop position control and any remaining following error is compensated. • If the axis is traversed via an NC program or manually, the set position and set velocity are displayed as the actual position and actual velocity on the user interface. With channel reset or end of program (M30/M2), the display of the actual position is set to the actual value of the machine axis. • The traverse commands are output to the NC/PLC interface. Effects when the axis is moving: • The axis is stopped in accordance with the currently active braking characteristic and an alarm is displayed. Subsequently, the travel movements of the axis are disabled. Position control remains active and holds the axis in the current position. **Spindle** Effects in open-loop control mode when the spindle is stopped: • No setpoints are output to the speed controller, i.e. the spindle traversing movement is disabled. • If the spindle is traversed via an NC program or manually, the speed setpoint is displayed on the user interface as the speed actual value. Effects in control mode when the spindle is rotating: • Speed setpoint zero is immediately output to the speed controller. • The spindle is stopped in accordance with the currently active braking characteristic and an alarm is displayed. Subsequently, the travel movements of the spindle are disabled. Position control remains active and holds the axis in the current position. Effects in positioning mode: • See Stationary axis / Moving axis Structure graphic Axis/spindle disable **Structure graphic**  Axis/spindle disable **Note** • If one of the following interface signals is set for a traversing axis/spindle for which the "Axis/spindle disable" signal is active, the axis/spindle is **not** stopped: – "LBP_Chan*.A_Geo[*].FDStop" (DB21, ... DBX12.3, 16.3 oder 20.3) Feed stop for geometry axis 1, 2 and 3 – "LBP_Chan*.A_Ori[*].FDStop" (DB21, ... DBX320.3, 324.3 oder 328.3) Feed stop for orientation axis 1, 2 and 3

– "LBP_Axis*.A_FDSpStop" (DB31, ... DBX4.3)
Feed stop / spindle stop

- The axis/spindle can still be switched to the "hold" or "follow up" state with "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) (follow-up mode).
- An axis cannot be referenced either channel-specifically via "LBP_Chan*.A_Ref" (DB21, ... DBX1.0) from the NC program (G74) or axis-specifically (REF machine function).
- If the signal "Axis/spindle disable" is briefly set for a traversing axis, the axis is stopped without an alarm. With the next traverse request, the axis is traversed to the new position.
Example:

```
N10 G0 X0 Y0
N20 G1 F1000 X100
N30 Y100
N40 X200
```

Regarding N20: At position 20 mm, "Axis/spindle lock" is set briefly for axis X ⇒ axis X is stopped
Regarding N40: Axis X traverses from the last position (approx. 20 mm + braking distance) to position 200 mm.

- Effects of "Axis/spindle disable" for spindle or axis couplings:

"LBP_Axis*.E_MasterSp" [1] (DB31, ... DBX99.0)	**"LBP_Axis*.E_SlaveSp"** [2] (DB31, ... DBX99.1)	**Coupl.** [3]	**Effect**
0	0	Off	Setpoints are output
0	1	Off	No setpoint output for following spindle/axis
1	0	Off	No setpoint output for leading spindle/axis
1	1	Off	No setpoint output for leading and following spindle/axis
0	0	On	Setpoints are output
0	1	On	Axis/spindle disable has no effect on the FS / FA
1	0	On	Axis/spindle disable also has an effect on the FS / FA
1	1	On	No setpoint output

1) Following spindle/axis active
2) Following spindle/axis active
3) Status of the coupling between leading and following spindle/axis

Corresponds to

- "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.1)
Activate program test
- "LBP_Axis*.E_AxSpDisable" (DB31, ... DBX63.3)
Axis/spindle lock active
- "LBP_Chan*.A_ProgTest" (DB21, ... DBX1.7)
Activate program test
- "LBP_Chan*.E_ProgTest" (DB21, ... DBX33.7)
Program test active

Additional references

Response for synchronous operation:
Function Manual Axes and Spindles; synchronous spindle

A_Follow_upMode (DB31, ... DBX1.4)

Description	Follow-up mode If the controller enable is canceled in the control because of faults, the "hold" state should be activated for the axis before the NC start after the active alarms have been successfully reset, and the accompanying controller enable has been set in the in the control: "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) = FALSE (follow-up mode). Otherwise, for an NC start and active follow-up mode, the traversing distance of the previous NC block would not be executed due to the internal delete distance-to-go. **Notice** During the transition from the "follow-up" state to the "hold" state or when the controller enable is set in position control, delete distance-to-go is activated in the control. As a consequence, a traversing block in which only this axis is traversed, is terminated directly.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Follow-up mode is not active: • The position setpoint is not tracked: Position setpoint = programmed set position • Feedback signal: "LBP_Axis*.E_Follow_upMode" (DB31, ... DBX61.3) = FALSE (follow-up not active) • Standstill and clamping monitoring are active. **Note** If the controller enable is reset ("LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) == FALSE), the axis is in the "hold" state. The set position of the axis is not corrected to the actual position. If the axis is moved in this state, e.g. manually, the difference between the set position and the actual position increases constantly (following error). The following error is suddenly corrected to zero when the "Controller enable" is set (speed setpoint jump).
Value TRUE	Follow-up mode is active: • The position setpoint is continuously tracked: position setpoint = actual position value • Feedback signal: "LBP_Axis*.E_Follow_upMode" (DB31, ... DBX61.3) = TRUE (follow-up active) • Standstill and clamping monitoring are not active. • If an NC program is active when the closed-loop control system is switched on again, a control-internal repositioning operation is performed (REPOSA: Linear approach with all axes) to the last programmed position. **Note** Follow-up mode is only useful with a simultaneous reset of the controller enable: "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) = FALSE
Corresponds to	• "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • "LBP_Axis*.A_Clamp" (DB31, ... DBX2.3) Clamping in progress • "LBP_Axis*.E_Follow_upMode" (DB31, ... DBX61.3) Follow-up active

A_PosMeas1, A_PosMeas2

(DB31, ... DBX1.5, 1.6)

<table>
<tr><td rowspan="7">Description</td><td colspan="3">Position measuring system 1 (PMS1) / position measuring system 2 (PMS2)</td></tr>
<tr><td>Bit 6
(PMS 2)</td><td>Bit 5
(PMS 1)</td><td>Effect</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>Position measuring system 1 is active:
• Position control of the machine axis via position measuring system 1.
• Monitoring functions (measuring system, standstill, clamping monitoring, contour deviation, etc.) of the machine axis via position measuring system 1.
• If position measuring system 2 exists (MD30200 $MA_NUM_ENCS == 2), its actual position value is acquired, but not monitored by any of these functions.</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>Position measuring system 2 is active:
• Position control of the machine axis via position measuring system 2.
• Monitoring functions (measuring system, standstill, clamping monitoring, contour deviation, etc.) of the machine axis via position measuring system 2.
• If position measuring system 1 exists (MD30200 $MA_NUM_ENCS == 2), its actual position value is acquired, but not monitored by any of these functions.</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>• Position control of the machine axis via position measuring system 1.
• If position measuring system 2 is available (MD30200 $MA_NUM_ENCS == 2), its actual position value is also acquired.</td></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Position measuring systems 1 and 2 are inactive ("parking" of the machine axis):
• There is no actual value acquisition.
• The monitoring of the position measuring system has been deactivated
• The following interface signals are reset:
– "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) / "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) == 0 (referenced/synchronized encoder 1/2)
– "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) (position controller active)
– "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) (speed controller active)
– "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) (current controller active)</td></tr>
<tr><td colspan="3">Notes
• If the interface signal of the active position measuring system is reset for a traversing axis, the axis is stopped with a ramp stop without the controller enable being canceled internally in the control.</td></tr>
</table>

	• If a speed-controlled spindle does not have a position measuring system, the "Controller enable" interface signal must be set: "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) == 1 (controller enable) • After deactivation of the "parking" state, incremental position measuring systems have to be referenced to achieve the "referenced" encoder status. • If the "parking" state is active, the following interface signal is ignored at NC start for this axis: "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) / "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) (referenced/synchronized 1/ 2). **Application examples** 1. Switching over from position measuring system 1 to positioning measuring system 2 (and vice versa): If the axis was referenced in both position measuring systems and the limit frequency of the used measured value encoder has not been exceeded in the meantime, i.e. "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) and "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) == 1 (referenced/synchronized 1/2), a new reference point approach is not required after the switchover. On switchover, the current difference between position measuring system 1 and 2 is traversed immediately. A tolerance band in which the deviation between the two actual values is permitted to be at the time of switchover can be specified in the following machine data: MD36500 $MA_ENC_CHANGE_TOL (maximum tolerance for actual position value switchover) If the actual value difference is greater than the tolerance, switchover is not performed and alarm 25100 "Measuring system switchover not possible" is displayed. 2. Machine axis is parked: The position measuring system monitoring is switched off when the measured value encoder is removed. 3. Switch off position measuring system: When the position measuring system 1 or 2 is switched off, the associated interface signal is reset: "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) / "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) (referenced/synchronized 1/2) 4. Reference point approach: Reference point approach of the axis is executed with the selected position measuring system. Every position measuring system must be referenced separately.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The position measuring system is inactive.
Value TRUE	The position measuring system is active.

Corresponds to	• "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) Referenced/synchronized 1 • "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) Referenced/synchronized 2 • "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) Speed controller active • MD36500 $MA_ENC_CHANGE_TOL (max. tolerance on actual position value switchover) • MD30200 $MA_NUM_ENCS (number of encoders)
Additional references	Function Manual, Basic Functions; Velocities, Setpoint / Actual Value Systems, Closed-Loop Control (G2)

A_ORactive (DB31, ... DBX1.7)

Description	Override active The feedrate override factor "LBP_Axis*.A_FD_OR" (DB31, ... DBB0) specified by the NC/PLC interface is activated with the interface signal. If the feedrate/spindle override is not active, independent of the actual switch position, a value of 1.0 ≙ 100 % is used as override factor in the control. The first switch position is an exception. Depending on the selected coding, for the first switch position the following factor applies: • Binary coding: Override factor = 0 • Gray coding: Override factor = MD12030 $MN_OVR_FACTOR_FEEDRATE[0]
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Feedrate/spindle override is not active.
Value TRUE	Feedrate/spindle override is active.
Corresponds to	• MD12000 $MN_OVR_AX_IS_GRAY_CODE (axis-specific feedrate override switch Gray-coded) • MD12030 $MN_OVR_FACTOR_FEEDRATE (evaluation of the path feedrate override switch) • MD12100 $MN_OVR_FACTOR_LIMIT_BIN (limit for binary coded override switch)
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"

A_SWCam (DB31, ... DBX2.0)

Description	Software cams: Activation
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to activate the output of the minus and plus cam signals of an axis to the PLC interface.
Value TRUE	A request has been made to activate the output of the minus and plus cam signals of an axis to the PLC interface.

Corresponds to	• "LBP_NC.E_SWCamMinus" (DB10 DBX110.0 ... 113.7) Software cams: Minus output cam signal 1 to 32 • "LBP_NC.E_SWCamMinus" (DB10 DBX114.0 ... 117.7) Software cams: Plus output cam signal 1 to 32 • "LBP_Axis*.E_SWCam" (DB31, ... DBX62.0) Software cams active
Additional references	Function Manual Axes and Spindles; Section "Software cams, cam sequencers"

A_ContrEnable (DB31, ... DBX2.1)

Description	Controller enable **Mechanical clamping of an axis** If the axis is positioned at the clamping position, the clamping mechanism is closed. The controller enable is then reset. Otherwise, the position controller would constantly work against the clamping, if the axis was moved mechanically from its specified position during the clamping operation. When the clamping is removed, the controller enable is set first and then the mechanical clamping is released. **Note** • Traverse request for an axis/spindle without controller enable: – The axis/spindle is not traversed – The traverse command is output to the interface – As long as the traverse request is present, the axis/spindle is traversed immediately when the controller enable is set. • When the controller enable is reset for a traversing geometry axis, this always results in a contour violation.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic

Value FALSE	Controller is not enabled. The behavior when the "controller enable" is removed depends on whether the axis/spindle is stationary or traversing at this time: • Axis/spindle stationary: – The position control loop of the axis is opened. – For "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) == 1 (follow-up mode) ⇒ position setpoint = actual position value – The controller enable on the drive is reset – The following interface signals are reset: "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = 0 (position controller active) "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) = 0 (speed controller active) "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) = 0 (current controller active) • Axis/spindle traverses – The axis is stopped with rapid stop. – Alarm 21612 "Controller enable VDI signal reset during motion" – The position control loop of the axis/spindle is opened. – Independent of the interface signal "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) (follow-up mode), the position setpoint is corrected at the end of the braking operation (position setpoint = actual position value) and the feedback signal "LBP_Axis*.E_Follow_upMode" (DB31, ... DBX61.3) = 1 (follow-up mode) is set. – The following interface signals are reset: "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) **Note** The current set position is retained in the control when "Controller enable" is reset and "Follow-up mode" ("LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) == 0) is not set. If the axis is moved in this state, e.g. manually, the difference between the set position and the actual position increases constantly (following error). The following error is suddenly corrected to zero when the "Controller enable" is set (speed setpoint jump).

Value TRUE	Controller is enabled. The position control loop is closed and the axis/spindle is in closed-loop control. Feedback: "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = 1 (position controller active) If the axis/spindle was referenced before resetting the interface signal, the axis/spindle does not have to be re-referenced after the interface signal is set again. Supplementary condition: The limit frequency of the active measuring system must not be exceeded in the meantime. **Note** If the axis/spindle was moved from its position during the time in which the controller enable was not set, the behavior when the controller enable is set depends on the interface signal "follow-up mode": • "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) == 1 (follow-up mode) Position control is performed at the current position • "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) == 0 (follow-up mode) Position control is performed at the last position before the controller enable is reset
Corresponds to	• "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) Follow-up mode • "LBP_Axis*.E_Follow_upMode" (DB31, ... DBX61.3) Follow-up active • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) Position controller active • "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) Speed controller active • "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) Current controller active • MD36620 $MA_SERVO_DISABLE_DELAY_TIME Cutout delay of the controller enable • MD36610 $MA_AX_EMERGENCY_STOP_TIME Braking ramp time when errors occur

A_DelDTGSpReset (DB31, ... DBX2.2)

Description	Delete distance-to-go/spindle reset **Axis: Delete distance-to-go** • AUTOMATIC and MDI modes The interface signal is only active if the axis is traversed as positioning axis. A positioning axis is decelerated to standstill along its brake characteristic. The distance-to-go of the axis is deleted. • JOG mode The axis is braked down to standstill alone its deceleration characteristic and then a possible delete distance to go is a deleted. **Note** After the delete distance-to-go, in an NC program, the following block is prepared again with the actual axis position. As a consequence, a different contour is traversed along than programmed in the NC program. Traversing to absolute position `G90` in the block after "Delete distance-to-go" at least allows the programmed position is approached. When traversing using incremental dimension `G91`, then the programmed position would not be approached. **Spindle: Reset** • Control mode: – The spindle is stopped – The NC program is again executed – The spindle continues to rotate with the next M and S value programmed in the NC program. • Oscillation mode: – Oscillation is interrupted – The axes continue to traverse – The NC program continues with the actual gear stage – The spindle continues to rotate with the next M and S value programmed in the NC program. The speed obtained from these values (actual S value and the last active gear stage) could be too high. In this case, the interface signal is set: "LBP_Axis*.E_SetSpeedLimit" (DB31, ... DBX83.1) • Positioning mode: The spindle is stopped • Axis mode: The spindle is stopped **Note** The response for a spindle reset is independent of the setting in machine data: MD35040 $MA_SPIND_ACTIVE_AFTER_RESET **Dwell time (`G4`): Delete distance-to-go** During a block with dwell time (`G4`), the interface signal has no influence on the time that elapses.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic

Edge change 0 → 1	**Axis**: "Delete-distance-to-go" is requested. **Spindle**: Reset is requested
Edge change 1 → 0	No effect.
Corresponds to	• "LBP_Chan*.A_DeleteDTG" (DB21, ... DBX6.2) Delete distance-to-go • "LBP_Axis*.E_SetSpeedLimit" (DB31, ... DBX83.1) Programmed speed too high • MD35040 $MA_SPIND_ACTIVE_AFTER_RESET (individual spindle reset)
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

A_Clamp (DB31, ... DBX2.3)

Description	Clamping in progress
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Clamping completed. Standstill monitoring is active.
Value TRUE	Clamping in progress. The clamping monitoring is active.
Corresponds to	• MD36050 $MA_CLAMP_POS_TOL Clamping tolerance

A_RefVal1, A_RefVal2, A_RefVal3, A_RefVal4

(DB31, ... DBX2.4, 2.5, 2.6, 2.7)

<table>
<tr><td>Description</td><td>Reference point values 1 to 4
When the reference cam is reached, the NC is informed via the interface which coded reference cam has been approached or which reference point value must be set as the actual value for the axis:
Actual value = MD34100 $MA_REFP_SET_POS [<reference point value x>]
<table>
<tr><th>A_RefVal4
Bit 7</th><th>A_RefVal3
Bit 6</th><th>A_RefVal2
Bit 5</th><th>A_RefVal1
Bit 4</th><th>Meaning</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Reference point value 1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Reference point value 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Reference point value 2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Reference point value 3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>Reference point value 4</td></tr>
</table>
Notes
• The interface signal must remain set until the reference point is reached or until a new coded reference cam is approached.
• If the machine axis has reached the reference point, and no reference point value has been selected, reference point value 1 is used.
• For length measuring systems with distance-coded reference marks, the interface signal is irrelevant.
Application example
On a machine tool with large traversing paths, up to four coded reference cams can be distributed over the distance traveled by the axis; up to four different reference points approached. This allows the traversing time when referencing to be shortened.</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• MD34100 $MA_REFP_SET_POS
Reference point value</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Referencing"</td></tr>
</table>

A_ExtZO (DB31, ... DBX3.0)

Description	Accept external work offset With the interface signal, the value in system variable $AA_ETRANS is transferred into the frame of the external work offset as external work offset ($P_EXTFRAME, $P_EXTFR): "LBP_Axis*.A_ExtZO" (DB31, ... DBX3.0): 0 → 1 ⇒ $P_EXTFRAME[<axis>] = $P_EXTFR[<axis>] = $AA_ETRANS[<axis>]
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	Accepting the values for the external work offset is requested.

Edge change 1 → 0	Accepting the values for the external work offset is not requested.
Corresponds to	• $AA_ETRANS[<axis>] External work offset • $P_EXTFRAME[<axis>] Active system frame, external frame • $P_EXTFR[<axis>] Data management frame for external frame • MD28082 $MC_MM_SYSTEM_FRAME_MASK,bit1 System frame
Additional references	Function Manual Basic Functions; Section "Axes, coordinate systems, frames"

A_EnabTravFixedStop (DB31, ... DBX3.1)

Description	Enable travel to fixed stop Irrelevant for MD37060 $MA_FIXED_STOP_ACKN_MASK == 0 or 2
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	**Negative feedback** to the request to enable travel to the fixed stop with "LBP_Axis*.E_ActTravFStop" (DB31, ... DBX62.4) = 1. Meaning **after activating** the function "travel to fixed stop" in the NC program (`FXS`): • Travel to fixed stop is inhibited. • The axis is stationary at the start position with reduced torque. • The channel message "Wait: Aux fct acknowledgment missing" is displayed. Meaning **before reaching** the fixed stop "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) == 0 (fixed stop reached): • Travel to fixed stop interrupted. • Alarm "20094: Axis%1 function was canceled" is displayed. Meaning **after reaching** the fixed stop "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) == 1 (fixed stop reached): • The torque limiting and monitoring of the fixed stop monitoring window are canceled.
Value TRUE	**Positive feedback** to the request to enable travel to the fixed stop with "LBP_Axis*.E_ActTravFStop" (DB31, ... DBX62.4) = 1.
Corresponds to	• "LBP_Axis*.A_AckFixedStop" (DB31, ... DBX1.1) Acknowledge fixed stop reached • "LBP_Axis*.A_SensorFixedStop" (DB31, ... DBX1.2) Fixed stop sensor • "LBP_Axis*.A_EnabTravFixedStop" (DB31, ... DBX3.1) Enable travel to fixed stop • "LBP_Axis*.E_ActTravFStop" (DB31, ... DBX62.4) Activate travel to fixed stop • "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) Fixed stop reached • MD37060 $MA_FIXED_STOP_ACKN_MASK Monitoring PLC acknowledgements for travel to fixed stop
Additional references	Function Manual Axes and Spindles, Section "Travel to fixed stop"

A_FixedFD1, A_FixedFD2, A_FixedFD3, A_FixedFD4

(DB31, ... DBX3.2, 3.3, 3.4, 3.5)

<table>
<tr><td>Description</td><td>Activate fixed feedrate 1 to 4, machine axes
With the interface signals, in the JOG mode, the fixed feedrate - parameterized using machine data - is activated.
<table>
<tr><th>A_FixedFD4
Bit 5</th><th>A_FixedFD3
Bit 4</th><th>A_FixedFD2
Bit 3</th><th>A_FixedFD1
Bit 2</th><th>Meaning</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>Fixed feedrate not selected</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>Fixed feedrate 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>Fixed feedrate 2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>Fixed feedrate 3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>Fixed feedrate 4</td></tr>
</table>
Instead of the parameterized JOG velocity/ JOG rapid traverse velocity (MD... ...JOG_VELO_...), the axis is traversed with the selected fixed feedrate. The traversing direction is specified via the interface signal:
"LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) (traversing keys Minus / Plus)

The fixed feedrates are parameterized using the following machine data:
• Linear axes: MD12202 $MN_PERMANENT_FEED
• Rotary axes: MD12204 $MN_PERMANENT_ROT_AX_FEED

Notes
• The fixed feedrate is not used for:
– Spindles
– Positioning axes
– Tapping
• The fixed feedrate is always interpreted as linear feedrate. Switchover to linear feedrate is performed internally in the control system even when revolutional feedrate is active.</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• MD12200 $MN_RUN_OVERRIDE_0 (traversing behavior with override 0)
• MD12202 $MN_PERMANENT_FEED (fixed feedrates for linear axes)
• MD12204 $MN_PERMANENT_ROT_AX_FEED (fixed feedrates for rotary axes)
• MD21150 $MC_JOG_VELO_RAPID_ORI (conventional rapid traverse for orientation axes)
• MD21155 $MC_JOG_VELO_ORI (conventional velocity for orientation axes)
• MD21160 $MC_JOG_VELO_RAPID_GEO (conventional rapid traverse for geometry axes)
• MD21165 $MC_JOG_VELO_GEO (conventional velocity for geometry axes)
• MD32010 $MA_JOG_VELO_RAPID (conventional rapid traverse)
• MD32020 $MA_JOG_VELO (conventional axis velocity)</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate control" > "Fixed feedrate values"</td></tr>
</table>

A_VeloSpeedLimit (DB31, ... DBX3.6)

Description	Velocity/spindle speed limitation The control limits the velocity/spindle speed to the parameterized limit value: MD35160 $MA_SPIND_EXTERN_VELO_LIMIT
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The velocity/spindle speed limiting is not active.
Value TRUE	The velocity/spindle speed limiting is active.
Corresponds to	• MD35100 $MA_SPIND_VELO_LIMIT Maximum spindle speed • MD35160 $MA_SPIND_EXTERN_VELO_LIMIT Spindle speed limiting from the PLC • SD43220 $SA_SPIND_MAX_VELO_G26 Programmable upper spindle speed limitation for G26 • SD43230 $SA_SPIND_MAX_VELO_LIMS Spindle speed limitation for G96/G961/G97
Additional references	Function Manual Basic Functions; Section "Axes, coordinate systems, frames"

A_PrgtestAxRel (DB31, ... DBX3.7)

Description	Program test Axis/Spindle Enable
Signal flow	PLC → NC
Data type	BOOL

A_HW1, A_HW2, A_HW3

(DB31, ... DBX4.0, 4.1, 4.2)

<table>
<tr><td>Description</td><td>Activate handwheel
The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.
Bit coded: maximum 3 handwheels:
<table>
<tr><th>A_HW3
Bit 2</th><th>A_HW2
Bit 1</th><th>A_HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
<tr><td colspan="4">Binary coded: maximum 6 handwheels:</td></tr>
<tr><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
Note
At any one time, the machine axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Axis*.E_HW*" (DB31, ... DBX64.0, 64.1, 64.2)
Handwheel active
• MD11324 $MN_HANDWH_VDI_REPRESENTATION
Representation of the handwheel number at the VDI interface</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Manual traversing"</td></tr>
</table>

A_FDSpStop (DB31, ... DBX4.3)

<table>
<tr><td>Description</td><td>Feed/spindle stop, axis-specific
Axis
• If the interface signal is set while traversing the axis, the axis is braked down to standstill along its braking characteristic. If the axis is in an interpolatory relationship with other axes, these are also braked.
• After the interface signal has been reset, traversing motion that was stopped is continued.
• The position control is retained and the following error is eliminated.
• If the interface signal is set, and a request is issued to traverse, then the axis is not traversed. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the axis is traversed.
• The interface signal is active in all modes.
<table>
<tr><th>Thread cutting</th><th>Effectiveness</th></tr>
<tr><td>G33, G34, G35</td><td>Effective (contour deviations occur)</td></tr>
<tr><td>G331, G332</td><td>Effective</td></tr>
<tr><td>G63</td><td>Effective</td></tr>
</table>
Spindle
• If the interface signal is set while traversing the spindle, the spindle is braked down to standstill along its braking characteristic. If the spindle is in an interpolatory relationship with other spindles, these are also braked.
• After the interface signal has been reset, traversing motion that was stopped is continued.
• During positioning, the position control is retained and the following error is eliminated.
• If the interface signal is set, and a request is issued to move during positioning operation, the spindle is not moved. However, the traverse request is kept. When the interface signal is reset, the traverse request is immediately executed, i.e. the spindle is rotated.
• The interface signal is active in all modes.
• The interface signal is not active during tapping (G331, G332).
<table>
<tr><th>Thread cutting</th><th>Effectiveness</th></tr>
<tr><td>G33, G34, G35</td><td>Effective (contour deviations can occur depending on the dynamic performance that has been parameterized)</td></tr>
<tr><td>G331, G332</td><td>Not effective</td></tr>
<tr><td>G63</td><td>Effective</td></tr>
</table>
Note
The interface signal is not active when "Axis/spindle disable" ("LBP_Axis*.A_ASpDisable" (DB31, ... DBX1.3) == 1) is active.</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Value FALSE</td><td>Feed/spindle stop is not active.</td></tr>
<tr><td>Value TRUE</td><td>Feed/spindle stop is active.</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Axis*.A_ASpDisable" (DB31, ... DBX1.3)
Axis/spindle disable</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feed disable and feed/spindle stop"</td></tr>
</table>

A_Disable (DB31, ... DBX4.4)

Description	Traversing key disable **Note** If the traversing key disable is activated while traversing, the axis is stopped.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The "Plus" and "Minus" traversing keys are enabled.
Value TRUE	The "Plus" and "Minus" traversing keys are locked.
Corresponds to	• "LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) Traversing keys Minus / Plus
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_RapidTrOR (DB31, ... DBX4.5)

Description	Rapid traverse override The signal is only effective for continuous or incremental manual traversing in the JOG mode. The signal is irrelevant: • for reference point approach (JOG mode) • in the AUTOMATIC and MDI modes The rapid traverse velocity can be influenced using the rapid traverse override switch.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	During manual traversing of the machine axis using the traversing keys, the entered JOG velocity is active (SD41110 or MD32020).
Value TRUE	During manual traversing of the machine axis using the traversing keys, rapid traverse velocity becomes active (MD32010) when the interface signal is set.
Corresponds to	• "LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) Traversing keys Minus / Plus • SD41110 $SN_JOG_SET_VELO Axis velocity for JOG • MD32010 $MA_JOG_VELO_RAPID Conventional rapid traverse • MD32020 $MA_JOG_VELO Conventional axis velocity
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_Minus, A_Plus

(DB31, ... DBX4.6, 4.7)

Description	Traversing keys "Minus" / "Plus" There is a request signal for every traversing key and/or axis direction:	
	A_Minus (Bit 6)	Traversing key "Minus" (for traversing in the negative axis direction)
	A_Plus (Bit 7)	Traversing key "Plus" (for traversing in the positive axis direction)
	Depending on the active machine function, as well as the settings for jog and continuous operation (SD41050 and MD11300), different responses are initiated when the signal changes: 1. Continuous manual travel in the jog mode The machine axis traverses in the direction concerned to the active limit switch as long as the interface signal is set to 1. 2. Continuous manual travel in continuous operation On the first edge change 0 →1 the machine axis starts to traverse in the appropriate direction. This traversing movement still continues when the edge changes from 1 → 0. Any new signal edge change 0 → 1 (same traversing direction!) stops the traversing movement. 3. Incremental manual travel in the jog mode With signal state 1, the machine axis starts to traverse at the set increment. If the signal changes to the 0 state before the increment is traversed, then traversing is interrupted. When the signal state changes to 1 again, traversing is continued. The axis can be stopped and started several times as described above until it has traversed the complete increment. 4. Incremental manual travel in continuous operation At the first edge change 0 → 1, the machine axis starts to traverse at the set increment. If another edge change 0 → 1 is performed with the same traverse signal before the geometry axis has traversed the increment, the traversing movement will be canceled. The increment is no longer traversed to the end.	
	Note • The interface signal is irrelevant in operating modes: – AUTOMATIC – MDI • If both traversing signals ("plus" and "minus") are set at the same time there is no traversing or the current traversing is aborted. • Traversing by means of the traversing keys can be locked via "LBP_Axis*.A_Disable" (DB31, ... DBX4.4) (traversing key disable).	
Signal flow	PLC → NC	
Data type	BOOL	
Update	Cyclic	
Value FALSE	A request has not been made to traverse the machine axis using the traversing key in the positive/negative axis direction.	
Value TRUE	A request has been made to traverse the machine axis using the traversing key in the positive/negative axis direction.	

Corresponds to	• "LBP_Axis*.A_Disable" (DB31, ... DBX4.4) Traversing key disable • "LBP_Axis*.E_TCMinus", "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.6, 64.7) Traverse command "Minus" / "Plus" • MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD INC and REF in jog mode • SD41050 $SN_JOG_CONT_MODE_LEVELTRIGGRD Jog/continuous operation for JOG, continuous
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_INC1, A_INC10, A_INC100, A_INC1000, A_INC10000, A_INCVar, A_ContManTravel

(DB31, ... DBX5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6)

Description	Machine function request There is a request signal for every machine function to manually traverse the machine axis in the JOG mode. **Incremental manual travel** ("LBP_Axis*.A_INCVar" (DB31, ... DBX5.5)) In addition to five fixed increment sizes (default setting in MD11330: INC1, INC10, INC100, INC1000 and INC10000), a variable increment size (INCvar) that can be set via the setting data SD41010 is also available. The distance evaluation of one increment for fixed and variable increment sizes is performed via the axis-specific machine data MD31090. When the "Plus" or "Minus" traversing key is pressed or the electronic handwheel is turned, the axis starts to traverse in the appropriate direction by the number of increments corresponding to the active machine function. **Continuous manual travel** ("LBP_Axis*.A_ContManTravel" (DB31, ... DBX5.6)) For continuous manual travel, the "Plus" and "Minus" traversing keys are selected to move the axis continuously and the appropriate direction. **Notes** • If several requests are set simultaneously, no machine function becomes active. • If an axis is currently being traversed via a machine function, the traversing movement is aborted when the machine function is deselected or switched over.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The machine function to manually traverse the machine axis has not been requested.
Value TRUE	The machine function to manually traverse the machine axis has been requested.

Corresponds to	• "LBP_Axis*.E_INC1", "LBP_Axis*.E_INC10", "LBP_Axis*.E_INC100", "LBP_Axis*.E_INC1000", "LBP_Axis*.E_INC10000", "LBP_Axis*.E_INCVar", "LBP_Axis*.E_ContManTravel" (DB31, ... DBX65.0 ... 65.6) Active machine function • MD11320 $MN_HANDWH_IMP_PER_LATCH Handwheel pulses per detent position • MD11330 $MN_JOG_INCR_SIZE_TAB Increment size for INC/handwheel • MD31090 $MA_JOG_INCR_WEIGHT Evaluation of an increment for INC/handwheel SD41010 $SN_JOG_VAR_INCR_SIZE Size of the variable increment for JOG
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_OEMAxis (DB31, ... DBB6)

Description	OEM signals
Signal flow	PLC → NC
Data type	BYTE

A_InvHWDirOfRot (DB31, ... DBX7.0)

Description	Invert handwheel direction of rotation **Note** It is only permissible to change the interface signal when the machine axis is at a standstill. **Application examples** • The direction of movement of the handwheel does not match the expected direction of the axis. • A handwheel is assigned to several axes with different orientations.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A request has not been made to invert the direction of rotation of the handwheel assigned to the machine axis.
Value TRUE	A request has been made to invert the direction of rotation of the handwheel assigned to the machine axis.
Corresponds to	• "LBP_Axis*.E_InvHWDirOfRot" (DB31, ... DBX67.0) Handwheel direction of rotation inversion active
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_NCASpChanA, A_NCASpChanB, A_NCASpChanC, A_NCASpChanD, A_NCASpStrobe, A_PLCASp

(DB31, ... DBX8.0, 8.1, 8.2, 8.3, 8.4, 8.7)

<table>
<tr><td>Description</td><td colspan="6">Axis/spindle exchange request
From the PLC user program, via DBB8 an axis/spindle exchange can be requested:
• From an NC channel to the PLC
• From the PLC to an NC channel
• From an NC channel to another NC channel</td></tr>
<tr><td></td><td>Bit</td><td colspan="5">Meaning</td></tr>
<tr><td></td><td>0 ... 3</td><td colspan="5">Name of the channel to which the axis/spindle is to be assigned (binary code).
Example: The axis is to be assigned to channel 2.</td></tr>
<tr><td></td><td></td><td>A_NCASpC-hanD
Bit 3</td><td>A_NCASp-ChanC
Bit 2</td><td>A_NCASp-ChanB
Bit 1</td><td>A_NCASpCha-nA
Bit 0</td><td>Channel num-ber</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td></td><td>4</td><td colspan="5">Request axis/spindle interchange (A_NCASpStrobe)
Note
The bit is automatically reset after the assignment is executed</td></tr>
<tr><td></td><td></td><td>0 → 1</td><td colspan="4">Request from PLC</td></tr>
<tr><td></td><td></td><td>1 → 0</td><td colspan="4">Acknowledged by the NC</td></tr>
<tr><td></td><td>5</td><td colspan="5">-</td></tr>
<tr><td></td><td>6</td><td colspan="5">-</td></tr>
<tr><td></td><td>7</td><td colspan="5">Request axis type "PLC axis" (A_PLCASp)</td></tr>
<tr><td>Signal flow</td><td colspan="6">PLC → NC</td></tr>
<tr><td>Data type</td><td colspan="6">BOOL</td></tr>
<tr><td>Update</td><td colspan="6">Cyclic</td></tr>
<tr><td>Edge change 0 → 1</td><td colspan="6">Request from PLC, switchover axis</td></tr>
<tr><td>Edge change 1 → 0</td><td colspan="6">Acknowledged by the NC</td></tr>
<tr><td>Corresponds to</td><td colspan="6">• "LBP_Axis*.E_NCASpChan*", "LBP_Axis*.E_PLCType", "LBP_Axis*.E_ChPoss", "LBP_Axis*.E_NeutrASp", "LBP_Axis*.E_PLCASp"
(DB31, ... DBB68)
Axis/spindle interchange
• MD20070 $MC_AXCONF_MACHAX_USED
Machine axis number valid in channel
• MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN
Initial setting of channel for axis interchange</td></tr>
<tr><td>Additional references</td><td colspan="6">Function Manual Basic Functions, Section: "Cross-channel axis interchange"</td></tr>
</table>

A_Para_A, A_Para_B, A_Para_C

(DB31, ... DBX9.0, 9.1, 9.2)

Description	Selection: Position controller parameter set			
	A_Para_C Bit 2	**A_Para_B Bit 1**	**A_Para_A Bit 0**	**Parameter set**
	0	0	0	1
	0	0	1	2
	0	1	0	3
	0	1	1	4
	1	0	0	5
	1	0	1	6
	1	1	0	6
	1	1	1	6
	Note The selection of a position controller parameter set is ignored in the following case: MD35590 $MA_PARAMSET_CHANGE_ENABLE == 0			
Signal flow	PLC → NC			
Data type	BOOL			
Update	Job-controlled			
Corresponds to	• "LBP_Axis*.E_ParS_*" (DB31, ... DBX69.0, 69.1, 69.2) Feedback: Active position controller parameter set			

A_LockParSetDef (DB31, ... DBX9.3)

Description	Parameter set definitions disabled by NC
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	The parameter set switchover is enabled.
Value TRUE	The parameter set switchover is disabled.
Corresponds to	• "LBP_Axis*.A_Par_*" (DB31, ... DBX9.0, 9.1, 9.2) Select position controller parameter set

A_REPOSDelay (DB31, ... DBX10.0)

Description	REPOS delay **Note** If the machine axes are involved on a path ("LBP_Axis*.E_PathAxis" (DB31, ... DBX76.4) == 1 (path axis)), the interface signal is not active.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	A REPOS delay is not requested. After a block search, a REPOS offset is applied for this axis using the approach block
Value TRUE	A REPOS delay is requested. After a block search, a REPOS offset is applied for this axis. However it is not applied using the approach block, but rather using the next traversing block in which the axis is programmed.

Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgement • "LBP_Axis*.E_PathAxis" (DB31, ... DBX76.4) Path axis • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

A_StartBrakeTest (DB31, ... DBX11.0)

Description	SI: Start brake test
Signal flow	PLC → NC
Data type	BOOL

A_HWLimitMinus, A_HWLimitPlus

(DB31, ... DBX12.0, 12.1)

Description	Hardware limit switch minus/Hardware limit switch plus Alarm 21614 "Hardware limit switch -" or "Hardware limit switch +" is displayed, and the axis is stopped.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The hardware limit switch minus or plus was not activated.
Value TRUE	The hardware limit switch minus or plus was activated.
Corresponds to	• MD36600 $MA_BRAKE_MODE_CHOICE Deceleration behavior when the hardware limit switch responds

A_SWLimit2Minus, A_SWLimit2Plus

(DB31, ... DBX12.2, 12.3)

Description	Second software limit switch minus / software limit switch plus
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The second software limit switch minus or plus was not activated.
Value TRUE	The second software limit switch minus or plus was activated.
Corresponds to	• MD36100 $MA_POS_LIMIT_MINUS 1st software limit switch minus • MD36120 $MA_POS_LIMIT_MINUS2 2nd software limit switch minus • MD36110 $MA_POS_LIMIT_PLUS 1st software limit switch plus • MD36130 $MA_POS_LIMIT_PLUS2 2nd software limit switch plus

A_ModuloLimitEn (DB31, ... DBX12.4)

Description	Modulo rotary axis: activate traversing range limits **Note** The signal is irrelevant if linear axes/rotary axes do not have modulo functionality. **Application example** Mounted rotary axis with monitoring
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Deactivate traversing range limits for modulo rotary axes.
Value TRUE	Activate traversing range limits (software limit switch, working area limits) for modulo rotary axes.
Corresponds to	• "LBP_Axis*.E_ModLimEnAct" (DB31, ... DBX74.4) Modulo rotary axis: traversing range limits active
Additional references	Function Manual Axes and Spindles; Section "Rotary axes"

A_DelayRef (DB31, ... DBX12.7)

Description	Reference point approach delay **Note** Using a suitable reference cam, which extends from the reference point to the end of the traversing range, it is recommended to ensure that before the reference point approach, the machine axis can be located in the range behind the reference cam.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The machine axis is positioned in front of the reference cam.
Value TRUE	The machine axis is positioned at the reference cam.

Corresponds to	• "LBP_Axis*.A_RefVal*" (DB31, ... DBX2.4, 2.5, 2.6, 2.7) Reference point value 1 to 4
Additional references	Function Manual Axes and Spindles; Section "Referencing"

A_JogFixPPos0, A_JogFixPPos1, A_JogFixPPos2

(DB31, ... DBX13.0, 13.1, 13.2)

Description	JOG fixed point approach After selecting the "Approach fixed point in JOG" function, the PLC outputs the number of the fixed point to be approached binary coded to the NC in "LBP_Axis*.A_JogFixPPos*" (DB31, ... DBX13.0, 13.1, 13.2):			
	Bit 2	**Bit 1**	**Bit 0**	**Number of fixed point to be approached**
	0	0	0	-
	0	0	1	1
	0	1	0	2
	0	1	1	3
	1	0	0	4
	As soon as the function is active (see "LBP_Axis*.E_JogFixPPos*Act" (DB31, ... DBX75.0, 75.1, 75.2), the selected machine axis can be traversed to the corresponding fixed point using either the traversing keys or the handwheel. The fixed points are defined using machine data MD30600.			
Signal flow	PLC → NC			
Data type	BOOL			
Update	Cyclic			
Corresponds to	• "LBP_Axis*.E_JogFixPPos*Act" (DB31, ... DBX75.0, 75.1, 75.2) JOG fixed point approach • "LBP_Axis*.E_JogFixPPos*" (DB31, ... DBX75.3, 75.4, 75.5) JOG fixed point approach reached • MD30600 $MA_FIX_POINT_POS[<n>] Fixed value positions of the axis			
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"			

A_JogToPos (DB31, ... DBX13.3)

Description	JOG travel to position As soon as the function is active (see "LBP_Axis*.E_JogToPos" (DB31, ... DBX75.6)), then, using the traversing keys or the handwheel, the selected machine axis can be traversed to the position specified with setting data SD43320. **Application** The function is used for machine tools that are operated purely manually, e.g. to approach drill-holes in precise increments. **Note** The following preconditions applying for the "Position travel in JOG" function: • The function can only be activated in the JOG mode. The function cannot be activated if the machine function JOG-REPOS or JOG-REF is active and in JOG in AUTOMATIC. • The axis to be traversed must be referenced. • A kinematic transformation may not be active. • The axis to be traversed may not be a following axis of an active coupling.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The function "Position travel in JOG" is not requested.
Value TRUE	The function "Position travel in JOG" is requested.
Corresponds to	• "LBP_Axis*.E_JogToPos" (DB31, ... DBX75.6) JOG travel to position active • "LBP_Axis*.E_JogPos" (DB31, ... DBX75.7) JOG position reached • SD43320 $SA_JOG_POSITION JOG position
Additional references	Function Manual Axes and Spindles; Section "Manual traversing"

A_ProgtestSuppress (DB31, ... DBX14.0)

Description	Suppress program test If function "Program test" (PRT) is active in the channel to which the axis currently belongs, the axis is traversed with internal axis disable. Setpoints are generated but they are not output to the machine axes. Actual value = Setpoint. If the program test is suppressed for the axis, setpoints are output to the machine axis despite the function "program test" being active in the channel to which the axis currently belongs. **Automatic transfer of the interface signals** The HMI request signals "LBP_Axis*.E_MMC_ProgtestSuppress" (DB31, ... DBX128.0) / "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.1) are only transferred from the PLC basic program to the PLC request signals "LBP_Axis*.A_ProgtestSuppress" (DB31, ... DBX14.0) / "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.1) if the "LBP_ConfigBP [FC1]" parameter **MMCToIF** is set to **TRUE**. If the parameters are not set, the PLC request signals must be set by the PLC user program.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic

Value FALSE	Suppression of the program test (setpoint output locked) is not requested.
Value TRUE	Suppression of the program test (setpoint output locked) is requested.
Corresponds to	• "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.1) Activate program test • "LBP_Axis*.E_MMC_ProgtestSuppress" (DB31, ... DBX128.0) Suppress program test • "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.1) Activate program test
Additional references	Function Manual Basic Functions; Section "Cross-channel program coordination" > "Channel-by-channel running-in"

A_ProgtestActivate (DB31, ... DBX14.1)

Description	Activate program test If the function "program test" is active, the axis is traversed with internal axis disable. Setpoints are generated but they are not output to the machine axes. Actual value = Setpoint. **Automatic transfer of the interface signals** The HMI request signals "LBP_Axis*.E_MMC_ProgtestSuppress" (DB31, ... DBX128.0) / "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.1) are only transferred from the PLC basic program to the PLC request signals "LBP_Axis*.A_ProgtestSuppress" (DB31, ... DBX14.0) / "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.1) if the "LBP_ConfigBP [FC1]" parameter **MMCToIF** is set to **TRUE**. If the parameters are not set, the PLC request signals must be set by the PLC user program.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Suppression of the program test (setpoint output locked) is not requested.
Value TRUE	Suppression of the program test (setpoint output locked) is requested.
Corresponds to	• "LBP_Axis*.A_ProgtestSuppress" (DB31, ... DBX14.0) Suppress program test • "LBP_Axis*.E_MMC_ProgtestSuppress" (DB31, ... DBX128.0) Suppress program test • "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.1) Activate program test
Additional references	Function Manual Basic Functions; Section "Cross-channel program coordination" > "Channel-by-channel running-in"

A_ActGearA, A_ActGearB, A_ActGearC

(DB31, ... DBX16.0, 16.1, 16.2)

<table>
<tr><td>Description</td><td>Actual gear stage
If the new gear stage is selected at the machine, the PLC user program sets the following interface signals as feedback signal to the NC:
• "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) (actual gear stage)
• "LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3) (gear stage has been changed)
This signals the NC that the correct stage has been successfully selected. The associated parameter set becomes active in the NC.
<table>
<tr><th>A_Act-GearC Bit 2</th><th>A_Act-GearB Bit 1</th><th>A_Act-GearA Bit 0</th><th>Parameter set No.</th><th>Meaning: Parameter set for</th></tr>
<tr><td>---</td><td>---</td><td>---</td><td>0</td><td>Axis mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td rowspan="2">1</td><td>Gear stage 1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Gear stage 1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td><td>Gear stage 2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td><td>Gear stage 3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td><td>Gear stage 4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="3">5</td><td>Gear stage 5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Gear stage 5</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Gear stage 5</td></tr>
</table>
Note
If the PLC user program signals back to the NC with a different actual gear stage than issued by the NC as the set gear stage requested by the PLC, the gear change is still considered to have been successfully completed and the actual gear stage that is signaled back is activated in the NC.</td></tr>
<tr><td>Signal flow</td><td>PLC → NC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3)
Gear stage has been changed
• "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5)
Oscillation speed
• "LBP_Axis*.E_SetpGear*" (DB31, ... DBX82.0, 82.1, 82.2)
Set gear stage
• "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3)
Change gear stage</td></tr>
<tr><td>Additional references</td><td>Function Manual Axes and Spindles; Section "Spindles"</td></tr>
</table>

A_GearChangeOv (DB31, ... DBX16.3)

Description	Gear is changed over If the new gear stage is selected at the machine, the PLC user program sets the following interface signals as feedback signal to the NC: • "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) (actual gear stage) • "LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3) = 0 → 1 (gear stage has been changed) Feedback signal from the NC to the PLC: "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3) = 0 (change gear stage) Feedback signal from the PLC to the NC: LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3) = 0 (gear stage has been switched over) This completes the gear stage change in the control system. **Note** • The interface signal is irrelevant for all spindle modes with the exception of the oscillation mode. • If the PLC user program signals back to the NC with a different actual gear stage than issued by the NC as the set gear stage requested by the PLC, the gear change is still considered to have been successfully completed and the actual gear stage that is signaled back is activated in the NC.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	No effect
Value TRUE	Gear is changed over.
Corresponds to	• "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) Actual gear stage • "LBP_Axis*.E_SetpGear*" (DB31, ... DBX82.0, 82.1, 82.2) Set gear stage • "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3) Change gear stage • "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5) Oscillation speed
Additional references	Function Manual Axes and Spindles; Section "Spindles"

A_Sp1Syn, A_Sp2Syn

(DB31, ... DBX16.4, 16.5)

Description	Resynchronize spindle, measuring system 1 (A_Sp1Syn) / measuring system 2 (A_Sp2Syn) **Note** The interface signal is irrelevant for all spindle modes with the exception of the control mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	There is an active request to synchronize measuring system 1 or measuring system 2 of the spindle.
Edge change 1 → 0	No effect

Corresponds to	• "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) Referenced/synchronized 1 • "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) Referenced/synchronized 2
Additional references	Function Manual Axes and Spindles; Section "Spindles"

A_NoMonitorGear (DB31, ... DBX16.6)

Description	No speed monitoring during gear change
Signal flow	PLC → NC
Data type	BOOL

A_DeleteS (DB31, ... DBX16.7)

Description	Delete S value **Control mode:** • The spindle is braked via its parameterized braking ramp down to standstill. **Oscillation mode, axis mode, positioning mode:** • Deleting the actual S value has no effect on the actual function. However, if the control mode is selected again, a new S value must be programmed.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	There is an active request to delete the S value.
Edge change 1 → 0	No effect
Additional references	Function Manual Axes and Spindles; Section "Spindles"

A_FD_ORSp (DB31, ... DBX17.0)

Description	Feedrate correction spindle valid
Signal flow	PLC → NC
Data type	BOOL

A_Sp1SynP, A_Sp2SynP

(DB31, ... DBX17.4, 17.5)

Description	Resynchronizing the spindle before positioning, measuring system 1 / measuring system 2 **Note** The interface signal is irrelevant for all spindle modes with the exception of the positioning mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	There is an active request to resynchronize measuring system 1 or measuring system 2 before positioning the spindle.
Edge change 1 → 0	No effect

Corresponds to	• "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) Referenced/synchronized 1 • "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) Referenced/synchronized 2
Additional references	Function Manual Axes and Spindles; Section "Spindles"

A_M3M4Inv (DB31, ... DBX17.6)

Description	Invert M3/M4 The direction of rotation of the spindle motor changes for the following functions: • M3 • M4 • M5 • SPOS/M19/SPOSA from the motion; not active for SPOS/M19/SPOSA from standstill.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	There is an active request to invert the direction of rotation programmed with M3 / M4.
Edge change 1 → 0	No effect
Additional references	Function Manual Axes and Spindles; Section "Spindles"

A_OscilPLC (DB31, ... DBX18.4)

Description	Oscillation via the PLC For the "Oscillation of the machine axis controlled by the PLC user program" function the following system data must be set or parameterized: • "LBP_Axis*.A_OscilPLC" (DB31, ... DBX18.4) = 1 (oscillation controlled by the PLC) • "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5) = 1 (oscillation speed) • MD35400 $MA_SPIND_OSCILL_DES_VELO = <oscillation speed> • MD35410 $MA_SPIND_OSCILL_ACCEL = <oscillation acceleration> • "LBP_Axis*.A_SetRotDirectRight" (DB31, ... DBX18.6) = ... → 0 → 1 → 0 ... (set direction of rotation clockwise) • "LBP_Axis*.A_SetRotDirectLeft" (DB31, ... DBX18.7) = ... → 1 → 0 → 1 ... (set direction of rotation counterclockwise) The direction of rotation when starting and the traversing duration for each direction of rotation must be specified by the PLC user program.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	The "Oscillation of the machine axis controlled by the PLC user program" function is requested.

Edge change 1 → 0	The "Oscillation of the machine axis controlled by the PLC user program" function is not requested.
Corresponds to	• "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5) Oscillation speed • "LBP_Axis*.A_SetRotDirectRight" (DB31, ... DBX18.6) Setpoint direction of rotation, clockwise • "LBP_Axis*.A_SetRotDirectLeft" (DB31, ... DBX18.7) Setpoint direction of rotation, counterclockwise • MD35400 $MA_SPIND_OSCILL_DES_VELO Oscillation speed • MD35410 $MA_SPIND_OSCILL_ACCEL Acceleration when oscillating

A_OscilSpeed (DB31, ... DBX18.5)

Description	Oscillation enable **Note** The interface signal is irrelevant for all spindle modes with the exception of the oscillation mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Traversing the spindle in oscillation mode (reciprocating movement) has not been enabled by the NC.
Value TRUE	Traversing the spindle in oscillation mode (reciprocating movement) has been enabled by the NC.

Corresponds to	• "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) Actual gear stage • "LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3) Gear stage has been changed • "LBP_Axis*.A_OscilPLC" (DB31, ... DBX18.4) Oscillation controlled by the PLC • "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5) Oscillation speed • "LBP_Axis*.A_SetRotDirectRight" (DB31, ... DBX18.6) Setpoint direction of rotation, clockwise • "LBP_Axis*.A_SetRotDirectLeft" (DB31, ... DBX18.7) Setpoint direction of rotation, counterclockwise • "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) Spindle stopped • "LBP_Axis*.E_SetpGear*" (DB31, ... DBX82.0, 82.1, 82.2) Set gear stage • "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3) Change gear stage • "LBP_Axis*.E_SetRange" (DB31, ... DBX83.5) Spindle in setpoint range • "LBP_Axis*.E_OscillMode" (DB31, ... DBX84.6) DB31, ... DBX84.6 Active spindle mode: Oscillation mode
Additional references	Function Manual Axes and Spindles; Section "Spindles" > "Configurable gear adaptations" > "Gear stage change with oscillation mode"

A_SetRotDirectRight, A_SetRotDirectLeft

(DB31, ... DBX18.6, 18.7)

Description	Oscillating: Direction of rotation clockwise (A_SetRotDirectRight) / direction of rotation counterclockwise (A_SetRotDirectLeft) **Note** The interface signal is irrelevant for all spindle modes with the exception of the oscillation mode.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Clockwise or counterclockwise oscillation direction of rotation is not requested.
Value TRUE	Clockwise or counterclockwise oscillation direction of rotation is requested.

Corresponds to	• "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) Actual gear stage • "LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3) Gear stage has been changed • "LBP_Axis*.A_OscilPLC" (DB31, ... DBX18.4) Oscillation controlled by the PLC • "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5) Oscillation speed • "LBP_Axis*.A_SetRotDirectRight" (DB31, ... DBX18.6) Setpoint direction of rotation, clockwise • "LBP_Axis*.A_SetRotDirectLeft" (DB31, ... DBX18.7) Setpoint direction of rotation, counterclockwise • "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) Spindle stopped • "LBP_Axis*.E_SetpGear*" (DB31, ... DBX82.0, 82.1, 82.2) Set gear stage • "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3) Change gear stage • "LBP_Axis*.E_SetRange" (DB31, ... DBX83.5) DB31, ... DBX83.5 Spindle in setpoint range • "LBP_Axis*.E_OscillMode" (DB31, ... DBX84.6) DB31, ... DBX84.6 Active spindle mode: Oscillation mode
Additional references	Function Manual Axes and Spindles; Section "Spindles" > "Configurable gear adaptations" > "Gear stage change with oscillation mode"

A_SpOR (DB31, ... DBB19)

Description

Speed override, spindle-specific

Binary/Gray coding

The override factors can be specified in the binary or Gray-coded format. The control must be informed of the used format via the following machine data:

MD12060 $MN_OVR_SPIN_IS_GRAY_CODE = <Coding>

Binary coding

With binary coding, the value in the interface corresponds to the override factor.

Binary code	Decimal	Override factor
0000 0000	0	0.00
0000 0001	1	0.01
0000 0010	2	0.02
0000 0011	3	0.03
000 0100	4	0.04
...	...	...
0110 0100	100	1.00
...	...	...
1100 1000	200	2.00

Gray coding

The Gray-coded values of the interface are assigned the override factors via the following machine data:

MD12070 $MN_OVR_FACTOR_SPIND_SPEED[<Switch position> - 1] = <Override factor>

Switch position	Gray code	Override factor [1)]
1	00001	0.50
2	00011	0.55
3	00010	0.60
4	00110	0.65
5	00111	0.70
6	00101	0.75
7	00100	0.80
8	01100	0.85
9	01101	0.90
10	01111	0.95
11	01110	1.00
12	01010	1.05
13	01011	1.10
14	01001	1.15
15	01000	1.20
16	11000	1.20
17	11001	1.20
18	11011	1.20
19	11010	1.20
20	11110	1.20
21	11111	1.20
22	11101	1.20

	23	11100	1.20
	24	10100	1.20
	25	10101	1.20
	26	10111	1.20
	27	10110	1.20
	28	10010	1.20
	29	10011	1.20
	30	10001	1.20
	31	10000	1.20
	1) Default values		
	Limitations • The active override factor is limited internally in the control to 2.00 or 200%. • The effective override factor is limited to 100% in the "oscillation mode" spindle mode. • The maximum possible override factor can be limited to a value less than 200% with the following machine data: MD12100 $MN_OVR_FACTOR_LIMIT_BIN **Notes** • The spindle-specific speed override is not effective with the following function when cutting threads: G63 • The spindle-specific speed override acts on the programmed values before other limits, e.g. G26, LIMS..., take effect.		
Signal flow	PLC → NC		
Data type	BYTE		
Update	Cyclic		
Value FALSE	Clockwise or counterclockwise oscillation direction of rotation is not requested.		
Value TRUE	Clockwise or counterclockwise oscillation direction of rotation is requested.		
Corresponds to	• "LBP_Axis*.A_FD_OR" (DB31, ... DBB0) Feedrate override, axis-specific • "LBP_Axis*.A_ORactive" (DB31, ... DBX1.7) Feedrate/speed override active • MD12080 $MN_OVR_REFERENCE_IS_PROG_FEED Override reference velocity		
Additional references	Function Manual Axes and Spindles; Section "Feedrates" > "Feedrate control" > "Feedrate override from the machine control panel"		

A_RUEncQuStop (DB31, ... DBX20.1)

Description	Ramp-function generator disable Only when cyclic interface between NC and drive is operated in the "611U compatibility mode". Using drive parameters p0922 and p2038, a PROFIdrive telegram compatible with the "SIMODRIVE 611 universal" interface mode must be set in the drive.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	No rapid stop with speed setpoint 0 is requested for the drive.

Value TRUE	A rapid stop with speed setpoint 0 is requested for the drive. The drive is stopped without a ramp function (regenerative braking). Feedback for the triggering of the rapid stop in the drive is via: "LBP_Axis*.E_RUEncDisabl" (DB31, ... DBX92.1) == TRUE (ramp-function generator disable active)
Corresponds to	• "LBP_Axis*.E_RUEncDisabl" (DB31, ... DBX92.1) Ramp-function generator disable active • Drive parameter p0922 IF1 PROFIdrive PZD telegram selection • Drive parameter p2038 IF1 PROFIdrive STW/ZSW interface mode
Additional references	Drive: SINAMICS S120/S150 List Manual

A_SpeedSetpSmooth (DB31, ... DBX20.3)

Description	A_SpeedSetpSmooth
Signal flow	PLC → NC
Data type	BOOL

A_HoldBrakeToOpen (DB31, ... DBX20.5)

Description	Open motor holding brake
Signal flow	PLC → NC
Data type	BOOL

A_ParA, A_ParB, A_ParC, A_MotA, A_MotB

(DB31, ... DBX21.0, 21.1, 21.2, 21.3, 21.4)

Description	Motor/drive data set: Selection The switchover to a new motor (MDS) and/or to drive data sets (DDS) is requested via the interface. **Formatting** Formatting the request interface, i.e. which bits are used to address the motor data sets (MDS) – and which are used to address the drive data sets (DDS) is set via the formatting interface (DB31, ... DBX130.0 ... 130.4). **Main spindle drive** For main spindle drives, the following classification applies: • MDS[0] → star operation • MDS[1] → delta operation **Switchover time** In principle it is possible to switch over drive parameter sets at any time. However, as torque jumps can occur when switching over speed controller parameters and motor speed normalization, parameters should only be switched over in stationary states, in particular, while the axes are stopped. As soon as the request to switch over to another motor data set is identified, the pulse enable is reset. **Application examples** • Main spindle drive For example, it is possible to switch between operating mode 1 (star operation) and operating mode 2 (delta operation) for a main spindle drive (MSD) by switching over the motor data set. • Drive parameter set It may be necessary to switch over the drive parameter set, for: – Gear stage change – Changing the measuring circuit
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_Axis*.E_Par*", "LBP_Axis*.E_MotA", "LBP_Axis*.E_MotB" (DB31, ... DBX93.0 ... 93.4) Motor/drive data set: Display • (DB31, ... DBX130.0 ... 130.4) Motor / drive data set: Formatting • "LBP_Axis*.A_MotOK" (DB31, ... DBX21.5) Motor has been selected
Additional references	• Function Manual Basic Functions; Section "Various NC/PLC interface signals and functions" > "Functions" > "Switchover motor/drive data sets"

A_MotOK (DB31, ... DBX21.5)

Description	Motor being selected The signal must be reset before a request to switch over to a new motor data set (MDS) and/or drive data set (DDS) ("LBP_Axis*.A_Par*", "LBP_Axis*.A_MotA", "LBP_Axis*.A_MotB" (DB31, ... DBX21.0 ... 21.4)).
Signal flow	PLC → NC

Data type	BOOL
Update	Cyclic
Value FALSE	The necessary electrical (e.g. contactor switchover for star-delta switchover) and/or mechanical switchovers have been completed: • It is not permissible that the axis traverses • The drive does not enable the pulses
Value TRUE	The necessary electrical and/or mechanical switchover operations (e.g. contactor switchover for star-delta changeover) have been completed: • It is permissible that the axis traverses again • The drive enables the pulses
Corresponds to	• "LBP_Axis*.A_Par*", "LBP_Axis*.A_MotA", "LBP_Axis*.A_MotB" (DB31, ... DBX21.0 ... 21.4) Motor/drive data set: Request interface • "LBP_Axis*.E_Par*", "LBP_Axis*.E_MotA", "LBP_Axis*.E_MotB" (DB31, ... DBX93.0 ... 93.4) Motor/drive data set: Display interface • (DB31, ... DBX130.0 ... 130.4) Motor/drive data set: Formatting interface
Additional references	• Function Manual Basic Functions; Section "Various NC/PLC interface signals and functions" > "Functions" > "Switchover motor/drive data sets"

A_IntegratDisable (DB31, ... DBX21.6)

Description	Integrator disable, speed controller If the speed controller integrator disable is activated, compensatory action may be taken, e.g. if the integrator was already holding a load while stationary. The feedback signal regarding the speed controller disable is realized from the drive using: "LBP_Axis*.E_IntegratDisable" (DB31, ... DBX93.6) (integrator disable, speed controller)
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The integrator (I component) of the speed controller is enabled (PI behavior).
Value TRUE	The integrator (I component) of the speed controller is disabled or is to be disabled (P instead of PI behavior). **Note:** If the speed controller integrator disable is activated, compensatory action may be taken in certain applications (e.g. if the integrator was already holding a load while stationary). The drive acknowledges the integrator disable to the PLC using the interface signal: "LBP_Axis*.E_IntegratDisable" (DB31, ... DBX93.6) (speed controller integrator disabled).
Corresponds to	• "LBP_Axis*.E_IntegratDisable" (DB31, ... DBX93.6) Integrator disable, speed controller

A_PulseEnable (DB31, ... DBX21.7)

Description	Pulse enable The pulses are only enabled in the drive when the drive signals that it is ready: "LBP_Axis*.E_MotOK" (DB31, ... DBX93.5) == TRUE (feedback: drive ready) If the pulse enable is removed during motion (e.g. emergency stop), the axis/spindle is no longer braked under control. The axis coasts to standstill.
Signal flow	PLC → NC

Data type	BOOL
Update	Cyclic
Value FALSE	The pulses are disabled for the drive.
Value TRUE	The pulses are enabled for the drive.
Corresponds to	• "LBP_Axis*.E_MotOK" (DB31, ... DBX93.5) Feedback signal: Drive ready • "LBP_Axis*.E_PulseEnable" (DB31, ... DBX93.7) Feedback: Pulses are enabled

A_RotationMonit (DB31, ... DBX24.0)

Description	Stepper motor: Rotation monitoring
Signal flow	PLC → NC
Data type	BOOL

A_ControlAx (DB31, ... DBX24.1)

Description	Control axis
Signal flow	PLC → NC
Data type	BOOL

A_TorqComp (DB31, ... DBX24.4)

Description	Master-slave: activate torque compensatory controller The torque compensatory controller can be activated for specific axes via this variable. Preconditions for activation: • Input of the supplementary speed setpoint is enabled: MD37254 $MA_MS_TORQUE_CTRL_MODE[<slave axis>] = TRUE • The "fine" differential speed is reached: "LBP_Axis*.E_MSFine" (DB31, ... DBX96.2) == TRUE
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	Activation of the torque compensatory controller is requested.
Edge change 1 → 0	Deactivation of the torque compensatory controller is requested.
Corresponds to	• "LBP_Axis*.E_MSFine" (DB31, ... DBX96.2) Master-slave: Fine speed difference • "LBP_Axis*.E_MSCompContr" (DB31, ... DBX96.4) Master-slave: Compensatory controller active • MD37254 $MA_MS_TORQUE_CTRL_MODE Interconnection torque compensatory controller
Additional references	• Function Manual Technologies, Section: "Speed/torque coupling, master-slave"

A_ChangSetpOutAss0, A_ChangSetpOutAss1

(DB31, ... DBX24.5, 24.6)

Description	Setpoint exchange: Request drive control **Notes** • If the request to accept the drive control is active simultaneously in multiple axes, switch-over does not occur. The drive control then remains with the axis in which it is currently active. • If no transfer requests are active during control system run-up, drive control is assigned by the control to the first of the axes found in the machine data (MD30110 $MA_CTRLOUT_MODULE_NR[<Achse>]) with the same logical drive number.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Accepting the drive control is not requested.
Value TRUE	Accepting the drive control is requested.
Corresponds to	• "LBP_Axis*.E_BitValCtrlOutChang0" (DB31, ... DBX96.5) Setpoint exchange: Drive control active • "LBP_Axis*.E_BitValCtrlOutChang1" (DB31, ... DBX96.6) Setpoint exchange: Drive control active
Additional references	• Function Manual Axes and Spindles; Section "Setpoint exchange"

A_MS (DB31, ... DBX24.7)

Description	Master-slave: Activate coupling A master-slave coupling can be activated/deactivated for specific axes via this interface signal. The following conditions must be satisfied for activating/deactivating a master-slave coupling: • MD37262 = 0: The master-slave coupling is not continuously active. • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) == TRUE: Master and slave axis must be in position control. • "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) == 1: Master and slave axis must be stationary. • "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) == TRUE: The channels of the master and slave axes must be in the "Reset" state. If a condition is not fulfilled, the coupling will not be activated or deactivated. No alarm appears and the state of the coupling remains the same. If, at a later point, all conditions are fulfilled, the coupling will be activated or deactivated depending on the state of the signal.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	Activation of the master-slave coupling is requested.
Edge change 1 → 0	Deactivation of the master-slave coupling is requested.

Corresponds to	• "LBP_Chan*.E_ChanReset" (DB21, ... DBX35.7) Channel state "reset" • "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) Axis/spindle stationary (n < nmin) • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) Position controller active • "LBP_Axis*.E_MS" (DB31, ... DBX96.7) Master-Slave: Coupling active • MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE Permanent master-slave coupling
Additional references	• Function Manual Technologies, Section: "Speed/torque coupling, master-slave"

A_DynBComp (DB31, ... DBX25.0)

Description	Dynam. backlash compensation activation
Signal flow	PLC → NC
Data type	BOOL

A_CompContr (DB31, ... DBX26.3)

Description	Grinding: Compensatory controller on
Signal flow	PLC → NC
Data type	BOOL

A_FollowAxOv (DB31, ... DBX26.4)

Description	Grinding: Enable following axis overlay This signal is required for the flying synchronization of master and slave axes. As long as the "Enable following axis overlay" signal stays set to TRUE, the following axis selected with `EGONSYN` will traverse to synchronization in the EG coupling group. If the EG axis grouping includes modulo axes, their position values are reduced in the modulo. thereby ensuring that they approach the next possible synchronization. If, for the following axis "Enable following axis overlay" has not been set, the axis does not traverse to synchronization. Instead, the program is stopped at the `EGONSYN` block and the self-clearing alarm 16771 is signaled.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	The following axis cannot be overlaid and traversed.
Value TRUE	Additional traversing motion can be overlaid on the following axis.
Additional references	• Function Manual Axes and Spindles; Section "Axis couplings"

A_DEPMCS (DB31, ... DBX27.0)

Description	Continue grinding: DEPMCS
Signal flow	PLC → NC
Data type	BOOL

A_DEPBCS (DB31, ... DBX27.1)

Description	Continue grinding: DEPBCS
Signal flow	PLC → NC
Data type	BOOL

A_Corr (DB31, ... DBX27.2)

Description	Continue grinding: Corr
Signal flow	PLC → NC
Data type	BOOL

A_HIAxMov (DB31, ... DBX27.3)

Description	Continue grinding: HIAxMove
Signal flow	PLC → NC
Data type	BOOL

A_StopDEPMCS (DB31, ... DBX27.4)

Description	Stop grinding: DEPMCS
Signal flow	PLC → NC
Data type	BOOL

A_StopDEPBCS (DB31, ... DBX27.5)

Description	Stop grinding: DEPBCS
Signal flow	PLC → NC
Data type	BOOL

A_StopCorr (DB31, ... DBX27.6)

Description	Stop grinding: Corr
Signal flow	PLC → NC
Data type	BOOL

A_StopHIAxMove (DB31, ... DBX27.7)

Description	Stop grinding: HIAxMove
Signal flow	PLC → NC
Data type	BOOL

A_OscillAxExtRev (DB31, ... DBX28.0)

Description	Oscillation grinding: Oscillation reversal from external
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	Brake oscillation motion and move oscillating axis in the opposite direction with the oscillating axis.

Edge change 1 → 0	Continue oscillation without interruption.
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

A_AxReset (DB31, ... DBX28.1)

Description	Oscillation grinding: Reset of a PLC-controlled axis NC responses: • The axis is transitioned into the "Single axis in Reset" status: $AA_SNGLAX_STAT == TRUE • Stopped sequences are canceled. • Internal axis states are reset. • Axis specific, machine data active at reset become active. • Execution is confirmed: – "LBP_Axis*.E_AxReset" (DB31, ... DBX63.0) (reset executed) == TRUE – "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) (axis stop active) == FALSE • Result: The axis is in the reset state. **Note** In conjunction with a channel reset, no axial machine data is active for axes controlled from the PLC.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	PLC-controlled axis: Reset is requested.
Edge change 1 → 0	No effect.
Corresponds to	• "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) PLC-controlled axis • "LBP_Axis*.E_PLCCtrlAx" (DB31, ... DBX63.1) Reset executed • "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) Axis stop active • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

A_AxResume (DB31, ... DBX28.2)

Description	Oscillation grinding: Resuming a PLC-controlled axis Using the signal, the traversing motion of an axis - controlled from the PLC - which is in the "Single axis is interrupted" status ($AA_SNGLAX_STAT = 3), is resumed. NC responses: • Checks whether, for the axis, an axis-specific alarm with clearance criterion "CANCELCLEAR" or "NCSTARTCLEAR" is present? If so, this is cleared. • Checks whether axis motion can be resumed? If so, the axis makes the transition into the "Single axis is active" state: $AA_SNGLAX_STAT == 4 • Execution is confirmed: – "LBP_Axis*.A_ExactCoarse" (DB31, ... DBX60.6) (position reached with exact stop coarse) == FALSE – "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) (position reached with exact stop fine) == FALSE – "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) (axis stop active) == FALSE – "LBP_Axis*.E_TCMinus" (DB31, ... DBX64.6) (traverse command minus) == TRUE – "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.7) (traverse command plus) == TRUE • Result: Traversing motion of the axis is resumed. **Note** The signal is ignored in the following cases: • The axis is not controlled by the PLC. • The axis is not in the "Single axis is interrupted" status. • It is not permissible that axis traversing motion resumes due to an active alarm.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	PLC-controlled axis: Resume uninterrupted traversing motion is requested.
Edge change 1 → 0	No effect.

Corresponds to	• "LBP_Axis*.A_Stop" (DB31, ... DBX28.6) PLC-controlled axis: Stop along braking ramp • "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) PLC-controlled axis • "LBP_Axis*.E_PLCCtrlAx" (DB31, ... DBX63.1) Reset executed • "LBP_Axis*.A_ExactCoarse" (DB31, ... DBX60.6) Position reached with exact stop coarse • "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) Position reached with exact stop fine • "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) Axis stop active • "LBP_Axis*.E_TCMinus" (DB31, ... DBX64.6) Traverse command "Minus" • "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.7) Traverse command "Plus" • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

A_SetRP (DB31, ... DBX28.3)

Description	Oscillation grinding: Set reversal point
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Reversal point 1
Value TRUE	Reversal point 2
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

A_ChangeRP (DB31, ... DBX28.4)

Description	Oscillation grinding: Change reversal point The braking position of the axis can be accepted as the new reversal position by means of the PLC signal "LBP_Axis*.A_ChangeRP" (DB31, ... DBX28.4) (change reversal position). The signal "LBP_Axis*.A_SetRP" (DB31, ... DBX28.3) (set reversal point) is ignored, but rather the change affects the last initiated "Oscillation reversal from external" command. **Note** For the axis, it is not permissible that changing the reversal points via traversing keys or handwheel is active. In this case, display alarm 20081 "Braking position cannot be used as reversal point - handwheel active "is signaled.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic

Value FALSE	The reversal point position cannot be changed by manually traversing the axis. In conjunction with "LBP_Axis*.A_OscillAxExtRev" (DB31, ... DBX28.0) (oscillation reversal from external): No change of the reversal point position.
Value TRUE	The reversal point position can be changed by manually traversing the axis. In conjunction with "LBP_Axis*.A_OscillAxExtRev" (DB31, ... DBX28.0) (oscillation reversal from external): The position at which the axis is braked after "oscillation reversal from external". This will be the new reversal point.
Corresponds to	• "LBP_Axis*.A_OscillAxExtRev" (DB31, ... DBX28.0) Oscillation reversal from external • "LBP_Axis*.A_SetRP" (DB31, ... DBX28.3) Set reversal point
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

A_StopRP (DB31, ... DBX28.5)

Description	Oscillation grinding: Stopping a PLC-controlled axis at the next reversal point
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no request to interrupt the oscillation motion of the PLC-controlled axis at the next reversal point.
Value TRUE	Interrupt oscillation motion of the PLC-controlled axis at the next reversal point.
Corresponds to	• "LBP_Axis*.A_Stop" (DB31, ... DBX28.6) PLC-controlled axis: Stop along braking ramp • "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) PLC-controlled axis
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

A_Stop (DB31, ... DBX28.6)

Description	Oscillation grinding: Stopping a PLC-controlled axis along a braking ramp Using the signal, the traversing motion of an axis - controlled from the PLC - can be stopped. NC responses: • The axis is braked down to standstill along its deceleration characteristic. • The axis is transitioned into the "Single axis is interrupted" status: $AA_SNGLAX_STAT == 3 • Execution is confirmed: – "LBP_Axis*.A_ExactCoarse" (DB31, ... DBX60.6) (position reached with exact stop coarse) == TRUE – "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) (position reached with exact stop fine) == TRUE – "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) (axis stop active) == FALSE – "LBP_Axis*.E_TCMinus" (DB31, ... DBX64.6) (traverse command "Minus") == FALSE – "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.7) (traverse command "Plus") == FALSE • Result: Axis is stopped. **Note** Following axis movements can only be suppressed when the leading axis stops. Retraction motion triggered by the "Extended stop and retract" function cannot be stopped.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	PLC-controlled axis: Braking the axis along a braking ramp is not requested.
Value TRUE	PLC-controlled axis: Braking the axis along a braking ramp is requested.
Corresponds to	• "LBP_Axis*.A_StopRP" (DB31, ... DBX28.5) PLC-controlled axis: Stop at the next reversal point • "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) PLC-controlled axis • "LBP_Axis*.A_ExactCoarse" (DB31, ... DBX60.6) Position reached with exact stop coarse • "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) Position reached with exact stop fine • "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) Axis stop active • "LBP_Axis*.E_TCMinus" (DB31, ... DBX64.6) Traverse command "Minus" • "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.7) Traverse command "Plus" • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

A_PLCAxis (DB31, ... DBX28.7)

Description	Oscillation grinding: Requesting a PLC-controlled axis NC responses: • Checks whether the axis is a main run axis or a neutral axis. • Checks whether an additional axis may be controlled from the PLC. • Transfer is confirmed: – "LBP_Axis*.E_PLCCtrlAx" (DB31, ... DBX63.1) (PLC-controlled axis) == TRUE – $AA_SNGLAX_STAT == TRUE • Result: PLC controls axis.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	There is no request that axis control is transferred to the PLC.
Value TRUE	Axis control should be relinquished to the PLC.
Corresponds to	• "LBP_Axis*.E_PLCCtrlAx" (DB31, ... DBX63.1) Reset executed • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

A_StartGantSynRun (DB31, ... DBX29.4)

Description	Couplings: Start gantry synchronization
Signal flow	PLC → NC
Data type	BOOL

A_NoAutoSync (DB31, ... DBX29.5)

Description	Couplings: Disable automatic synchronization
Signal flow	PLC → NC
Data type	BOOL

A_StopSpindle (DB31, ... DBX30.0)

Description	Technology - Spindle motion via the PLC interface: Spindle stop
Signal flow	PLC → NC
Data type	BOOL

A_StartCW (DB31, ... DBX30.1)

Description	Technology - Spindle motion via the PLC interface: Spindle start, clockwise rotation
Signal flow	PLC → NC
Data type	BOOL

A_StartCCW (DB31, ... DBX30.2)

Description	Technology - Spindle motion via the PLC interface: Spindle start, counter-clockwise rotation
Signal flow	PLC → NC
Data type	BOOL

A_AutoGearStep (DB31, ... DBX30.3)

Description	Technology - Spindle motion via the PLC interface: Select gear stage
Signal flow	PLC → NC
Data type	BOOL

A_PosSpindle (DB31, ... DBX30.4)

Description	Technology - Spindle motion via the PLC interface: Spindle-start positioning
Signal flow	PLC → NC
Data type	BOOL

A_Resynchronize (DB31, ... DBX31.4)

Description	Technology: Re-synchronization
Signal flow	PLC → NC
Data type	BOOL

A_DisSync (DB31, ... DBX31.5)

Description	Technology: Disable synchronization When the main run advances to a block with the part program statement `COUPON(<FS>,<LS>,<offset>)`, then interface signal "LBP_Axis*.A_DisSync" (DB31, ... DBX31.5) (block synchronization) is evaluated for the following spindle: • For "LBP_Axis*.A_DisSync" (DB31, ... DBX31.5) == TRUE, only continuous velocity synchronous operation is established. The following spindle does not execute any additional movement. The coupling then responds analogously to a programmed `COUPON(<FS>,<LS>)`. • For "LBP_Axis*.A_DisSync" (DB31, ... DBX31.5) == FALSE, the axis only moves through the position offset.
Signal flow	PLC → NC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: The PLC does not block synchronization motion for the following spindle specified by offset programming.
Value TRUE	Synchronous spindle coupling: The PLC blocks synchronization motion for the following spindle specified by offset programming.
Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode • "LBP_Axis*.E_SyncronCoarse" (DB31, ... DBX98.1) Coarse synchronous operation • "LBP_Axis*.E_SyncronFine" (DB31, ... DBX98.0) Fine synchronous operation
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

A_Tracksync (DB31, ... DBX31.6)

Description	Technology: Track synchronism
Signal flow	PLC → NC
Data type	BOOL

A_DelSyncRunCorr (DB31, ... DBX31.7)

Description	Technology: Reset synchronism correction
Signal flow	PLC → NC
Data type	BOOL

A_SI_SetpLimBit0 (DB31, ... DBX34.0)

A_SI_SetpLimBit1 (DB31, ... DBX34.1)

Description	Setpoint limitation, see MD36933
Signal flow	PLC → NC
Data type	BOOL

A_FeedDriveCAxEng (DB31, ... DBX56.0)

Description	Separate feed drive coupled as C axis
Signal flow	PLC → Operation software
Data type	BOOL

A_SpSpeedDisp (DB31, ... DBX56.1)

Description	Spindle speed display
Signal flow	PLC → Operation software
Data type	BOOL

A_SpInsideClamp (DB31, ... DBX56.2)

Description	Spindle inside clamping
Signal flow	PLC → Operation software
Data type	BOOL

CW7.A_DisablePowerReduction (DB31, ... DBX189.0)

Description	DYNEGMA: Disable power reduction
Signal flow	PLC → NC
Data type	BOOL

CW7.A_DisableCurrReduction (DB31, ... DBX189.1)

Description	Motor overtemperature: Current reduction
Signal flow	PLC → NC
Data type	BOOL

10.4.10.3 LBP_Axis1 [DB31], ... - Signals from the axis/spindle

Data blocks "LBP_Axis1 [DB31], ..." contain the axis and spindle signals.

This section describes the second part of the data block containing the axis and spindle signals.

Variables in "LBP_Axis1 [DB31], ..." - signals from axis/spindle

E_Sp_NA (DB31, ... DBX60.0)

Description	Rotary axis / spindle If a machine axis is operated alternately as a spindle or rotary axis, the signal can be used to identify which mode is currently active: • Turning machine: Spindle ↔ C axis • Milling machine: Spindle ↔ rotary axis for rigid tapping
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The machine axis is operated as rotary axis. **Valid interface signals** To the axis: • "LBP_Axis*.A_HWLimitMinus" ... "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBB12 ... DBB15) From the axis: • "LBP_Axis*.E_ModLimEnAct" ... "LBP_Axis*.E_FPosAxis" (DB31, ... DBB74 ... DBB81) **Invalid interface signals** To the spindle: • "LBP_Axis*.A_ActGearA" ... "LBP_Axis*.A_SpOR" (DB31, ... DBB16 ... DBB19) From the spindle: • "LBP_Axis*.E_SetpGearA" ... "LBP_Axis*.E_SFunct" (DB31, ... DBB82 ... DBB91)

Value TRUE	The machine axis is operated as a spindle in the following modes: • Control mode • Oscillation mode • Positioning mode • Tapping with compensating chuck • Synchronous mode **Valid interface signals** To the spindle: • "LBP_Axis*.A_ActGearA" ... "LBP_Axis*.A_SpOR" (DB31, ... DBB16 ... DBB19) From the spindle: • "LBP_Axis*.E_SetpGearA" ... "LBP_Axis*.E_SFunct" (DB31, ... DBB82 ... DBB91) **Invalid interface signals** To the axis: • "LBP_Axis*.A_HWLimitMinus" ... "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBB12 ... DBB15) From the axis: • "LBP_Axis*.E_ModLimEnAct" ... "LBP_Axis*.E_FPosAxis" (DB31, ... DBB74 ... DBB81)
Corresponds to	• "LBP_Axis*.E_SetpGearA" ... "LBP_Axis*.E_SFunct" (DB31, ... DBB82 ... DBB91)
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_NCU_Link (DB31, ... DBX60.1)

Description	NCU link: Axis active The signal is irrelevant for a system with an NCU.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is not active as NCU link axis.
Value TRUE	The axis is active as NCU link axis.
Additional references	• NCU Equipment Manual

E_EncodeFreq1 (DB31, ... DBX60.2)

Description	Encoder limit frequency exceeded, measuring system 1 The encoder frequency limit to be monitored is set using the following machine data: MD36300 $MA_ENC_FREQ_LIMIT The monitoring of the encoder frequency limit always refers to the active measuring system selected in the NC/PLC interface: "LBP_Axis*.A_PosMeas*" (DB31, ... DBX1.5, 1.6) (position measuring system 1 / 2)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	The encoder frequency limit of measuring system 1 has not been exceeded, or after it was exceeded, the encoder frequency limit for encoder resynchronization was fallen below again: MD36302 $MA_ENC_FREQ_LIMIT_LOW
Value TRUE	The encoder frequency limit of measuring system 1 has been exceeded ⇒ • "LBP_Axis*.E_RefSyn1" (DB31, ... DBX60.4) = FALSE Referenced / synchronized, position measuring system 1 • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = FALSE Position controller active • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = TRUE (Speed controller active) only for spindle • Axis is stopped (fast stop)
Corresponds to	• "LBP_Axis*.A_PosMeas1" (DB31, ... DBX1.5) Position measuring system 1 • "LBP_Axis*.A_PosMeas2" (DB31, ... DBX1.6) Position measuring system 2 • MD36300 $MA_ENC_FREQ_LIMIT Encoder frequency limit • MD36302 $MA_ENC_FREQ_LIMIT_LOW Encoder frequency limit for encoder resynchronization
Additional references	• Function Manual Monitoring and Compensating, Section: "Axis monitoring functions" > "Measuring system monitoring" > "Encoder limit frequency monitoring"

E_EncodeFreq2 (DB31, ... DBX60.3)

Description	Encoder frequency limit exceeded, measuring system 2 The encoder frequency limit to be monitored is set using the following machine data: MD36300 $MA_ENC_FREQ_LIMIT The monitoring of the encoder frequency limit always refers to the active measuring system selected in the NC/PLC interface: "LBP_Axis*.A_PosMeas*" (DB31, ... DBX1.5, 1.6) (position measuring system 1 / 2)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The encoder frequency limit of measuring system 2 has not been exceeded, or after it was exceeded, the encoder frequency limit for encoder resynchronization was fallen below again: MD36302 $MA_ENC_FREQ_LIMIT_LOW
Value TRUE	The encoder frequency limit of measuring system 2 has been exceeded ⇒ • "LBP_Axis*.E_RefSyn2" (DB31, ... DBX60.5) = FALSE Referenced / synchronized, position measuring system 2 • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = FALSE Position controller active • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = TRUE (Speed controller active) only for spindle • Axis is stopped (fast stop)

Corresponds to	• "LBP_Axis*.A_PosMeas1" (DB31, ... DBX1.5) Position measuring system 1 • "LBP_Axis*.A_PosMeas2" (DB31, ... DBX1.6) Position measuring system 2 • MD36300 $MA_ENC_FREQ_LIMIT Encoder frequency limit • MD36302 $MA_ENC_FREQ_LIMIT_LOW Encoder frequency limit for encoder resynchronization
Additional references	• Function Manual Monitoring and Compensating, Section: "Axis monitoring functions" > "Measuring system monitoring" > "Encoder limit frequency monitoring"

E_RefSyn1, E_RefSyn2

(DB31, ... DBX60.4, 60.5)

Description	referenced / synchronized 1 and 2 **Axis** After successful referencing or synchronizing, the interface signal is set. **Spindle** The interface signal is set at the latest after one spindle revolution (360 degrees), if the zero mark was passed or the contactless proximity switch has responded.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Position measuring system 1 / 2 of the machine axis is not referenced / synchronized.
Value TRUE	Position measuring system 1 / 2 of the machine axis is referenced / synchronized.
Corresponds to	• "LBP_Axis*.A_PosMeas2" (DB31, ... DBX1.6) Position measuring system 2 • "LBP_Axis*.A_RefVal*" (DB31, ... DBX2.4, 2.5, 2.6, 2.7) Reference point value 1 to 4 • "LBP_Axis*.A_DelayRef" (DB31, ... DBX12.7) Reference point approach deceleration • "LBP_Axis*.E_POS_RESTORED1" (DB31, ... DBX71.4) Position restored encoder 1 • "LBP_Axis*.E_POS_RESTORED2" (DB31, ... DBX71.5) Position restored encoder 2 • MD34102 $MA_REFP_SYNC_ENCS Measuring system calibration
Additional references	• Function Manual Axes and Spindles; Section "Referencing"

E_ExactCoarse (DB31, ... DBX60.6)

Description	Position reached with exact stop coarse
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	At least one of the following states applies: • The axis is traversing • Traversing motion is active for the axis • The actual axis position lies outside the parameterized exact stop limit MD36000 $MA_STOP_LIMIT_COARSE • The spindle is in closed-loop speed control (SPCOF/SPOSA) • The axis is in the "Follow-up mode" state • The axis is in the "Park" state • The axis was switched over from closed-loop position to closed-loop speed control
Value TRUE	At least one of the following states applies: • The axis is not traversing, and the actual axis position is within the parameterized exact stop limit MD36000 $MA_STOP_LIMIT_COARSE • The control system is in the "Reset" state • The axis was last programmed as positioning axis or positioning spindle • Path motion was terminated with an NC stop • The spindle is in closed-loop position control (SPCON/SPOS) and is stationary • The axis was switched over from closed-loop speed to closed-loop position control
Corresponds to	• MD36000 $MA_STOP_LIMIT_COARSE Exact stop coarse

E_ExactFine (DB31, ... DBX60.7)

Description	Position reached with exact stop fine
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	At least one of the following states applies: • The axis is traversing • Traversing motion is active for the axis • The actual axis position lies outside the parameterized exact stop limit MD36010 $MA_STOP_LIMIT_FINE • The spindle is in closed-loop speed control (SPCOF/SPOSA) • The axis is in the "Follow-up mode" state • The axis is in the "Park" state • The axis was switched over from closed-loop position to closed-loop speed control

Value TRUE	At least one of the following states applies: • The axis is not traversing, and the actual axis position is within the parameterized exact stop limit MD36010 $MA_STOP_LIMIT_FINE. • The control system is in the "Reset" state • The axis was last programmed as positioning axis or positioning spindle • Path motion was terminated with an NC stop • The spindle is in closed-loop position control (SPCON/SPOS) and is stationary • The axis was switched over from closed-loop speed to closed-loop position control
Corresponds to	• MD36010 $MA_STOP_LIMIT_FINE Exact stop fine

E_TravRequ (DB31, ... DBX61.0)

Description	Drive test traverse request
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The travel enable for the drive test is not requested.
Value TRUE	The travel enable for the drive test is requested because the travel conditions for the axis are satisfied: • The mechanical brake of the axis involved has been released • Axis disable is not active: "LBP_Axis*.A_ASpDisable" (DB31, ... DBX1.3) == FALSE The feedback signal is issued with: "LBP_Axis*.A_DriveEnable" (DB31, ... DBX1.0) == TRUE
Corresponds to	• "LBP_Axis*.A_DriveEnable" (DB31, ... DBX1.0) Drive test travel enable

E_AxialAlarm (DB31, ... DBX61.1)

Description	Axis-specific alarm Responses: • The axis is braked down to standstill along its deceleration characteristic. • Status of the axis is transitioned into "Single axis alarm is active": $AA_SNGLAX_STAT == 5
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No alarm
Value TRUE	An axis-specific alarm is output.
Corresponds to	• "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) Request, PLC-controlled axis • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

E_AxisReady (DB31, ... DBX61.2)

Description	NCU link: Axis ready The signal is processed on the NCU to which the drive of the axis connected. The axis is not ready if on this NCU an alarm with system response "NC ..." or "Mode group ..." or "Channel not ready" is displayed that involves this specific axis.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is not ready.
Value TRUE	The axis is ready.

E_Follow_upMode (DB31, ... DBX61.3)

Description	Follow up active **Note** The signal is not set in the "Hold" state. **Notice** Delete distance-to-go is executed in the control for the following state transitions: • From the state "Follow-up" into the state "Hold" ("LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) = 1 → 0 (follow-up mode)) • From the state "Follow-up" into the state "Position control" ("LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) = TRUE (controller enable))
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The follow-up mode for the axis/spindle is not active. Standstill and clamping monitoring are active.
Value TRUE	Follow-up mode for the axis/spindle is active. Possible causes: • The controller enable for the drive has been withdrawn: – "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) == FALSE (controller enable) – in the control for faults; see under "Additional reference" • Follow-up mode was selected: – "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) == TRUE (follow-up mode) – in the control, e.g. when withdrawing the controller enable for a traversing axis During follow-up mode: position setpoint = actual position value **Note** Standstill and clamping monitoring are not active.
Corresponds to	• "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) Follow-up mode
Additional references	• Diagnostics Manual

E_Stat (DB31, ... DBX61.4)

Description	Axis/spindle stationary ($n < n_{min}$) (status) The signal is always 0 if a traverse command is active for the axis/spindle ("LBP_Axis*.E_TCMinus", "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.6, 64.7)) even if the actual velocity of the axis or actual speed of the spindle is less than or equal to the parameterized limit value.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis/spindle is not stationary. This means that the actual velocity of the axis or the actual speed of the spindle is greater than the parameterized limit value: MD36060 $MA_STANDSTILL_VELO_TOL If a traverse command is present, e.g. for a spindle, the signal is always = 0 - even if the actual speed lies below that specified in MD36060. If the interface signal: "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) (axis/spindle stationary) is signaled and there is no closed-loop position control active for the spindle, at the user interface, an actual speed of zero is displayed and with the system variable $AA_S[n] zero is read.
Value TRUE	The axis/spindle is stationary. This means that the actual velocity of the axis or the actual speed of the spindle is less than or equal to the parameterized limit value: MD36060 $MA_STANDSTILL_VELO_TOL
Corresponds to	• MD36060 $MA_STANDSTILL_VELO_TOL Maximum velocity/speed for signal "Axis/spindle stationary"

E_PositContr (DB31, ... DBX61.5)

Description	Position controller active **Spindle without position control** For a spindle without position control, the signal is always FALSE. **Spindle with position control** For spindles with the option of position control, after activating the position control, e.g. using `SPCON` or `M70`, then the signal is handled the same as for a position-controlled axis. **Vertical axis** For a vertical axis, the holding brake should be activated as soon as the position control is no longer active ("LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) == FALSE). **Simulation axis** The output of the signal can also be parameterized for a simulation axis: MD30350 $MA_SIMU_AX_VDI_OUTPUT = 1
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The position controller is not active, i.e. the position control loop of the axis/spindle is not closed. The signal is reset if the controller enable ("LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) == FALSE) is reset by the PLC user program, or as a result of an internal fault.
Value TRUE	The position controller is active, i.e. the position control loop of the axis/spindle is closed.

Corresponds to	• "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • "LBP_Axis*.A_Follow_upMode" (DB31, ... DBX1.4) Follow-up mode • "LBP_Axis*.A_PosMeas1" (DB31, ... DBX1.5) Position measuring system 1 • "LBP_Axis*.A_PosMeas2" (DB31, ... DBX1.6) Position measuring system 2
Additional references	• Diagnostics Manual

E_SpeedContr (DB31, ... DBX61.6)

Description	Speed controller active **Spindle without position control** If the spindle is not under position control, the signal can be used as a feedback signal for signal "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) (controller enable). **Simulation axis** The output of the signal can also be parameterized for a simulation axis: MD30350 $MA_SIMU_AX_VDI_OUTPUT = 1
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The speed controller is not active, i.e. the speed control loop of the axis is not closed. The speed controller output is cleared.
Value TRUE	The speed controller is active, i.e. the speed control loop of the axis/spindle is closed.
Corresponds to	• "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) Position controller active

E_CurrentContr (DB31, ... DBX61.7)

Description	Current controller active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The current controller is not active, i.e. the current control loop of the axis is not closed. The current controller output, including the injection variables on the control voltage, is cleared.
Value TRUE	The current controller is active, i.e. the current control loop of the axis/spindle is closed.
Corresponds to	• "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) Position controller active • "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) Speed controller active

E_SWCam (DB31, ... DBX62.0)

Description	Software cams active
Signal flow	NC → PLC

Data type	BOOL
Update	Cyclic
Value FALSE	The output of the minus and plus cam signals of an axis to the PLC interface is not active.
Value TRUE	The output of the minus and plus cam signals of an axis to the PLC interface is active.
Corresponds to	• "LBP_NC.E_SWCamMinus" (DB10 DBX110.0 ... 113.7) Software cams: Minus output cam signal 1 to 32 • "LBP_NC.E_SWCamMinus" (DB10 DBX114.0 ... 117.7) Software cams: Plus output cam signal 1 to 32 • "LBP_Axis*.A_SWCam" (DB31, ... DBX2.0) Software cams: Activation
Additional references	• Function Manual Axes and Spindles; Section "Software cams, cam sequencers"

E_HWOverlay (DB31, ... DBX62.1)

Description	Handwheel override active For the positioning axis, the handwheel pulses either act as path definition (for `FDA[<axis>]=0`) - or as velocity override (for `FDA[<axis>]>0`) on the programmed axis feed. In the following cases, the override is not effective: • The positioning axis has reached the programmed target position. • The distance-to-go has been deleted. • Reset was initiated. **Note** The interface signal will also be set if "Handwheel override in automatic mode" is active with a concurrent positioning axis with **"**LBP_CtrlAxisSpindle [FC18]**"**.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The "Handwheel override in automatic mode" function is not active for the programmed positioning axis.
Value TRUE	The "Handwheel override in AUTOMATIC mode" function is active for the programmed positioning axis (`FDA[<axis>]`).
Corresponds to	• "LBP_Axis*.A_DelDTGSpReset" (DB31, ... DBX2.2) Delete distance-to-go
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

E_RevFD (DB31, ... DBX62.2)

Description	Revolutional feedrate active The interface signal indicates that in JOG mode or AUTOMATIC mode, the axes traverse as a positioning axis with revolutional feedrate.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Revolutional feedrate (`G95`) is not active.
Value TRUE	Revolutional feedrate (`G95`) is active.

Corresponds to	• SD41100 $SN_JOG_REV_IS_ACTIVE JOG: Revolutional/linear feedrate • SD41120 $SN_JOG_REV_SET_VELO Revolutional feedrate of axes in JOG • SD42600 $SC_JOG_FEED_PER_REV_SOURCE Control of the revolutional feedrate in JOG • SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE Revolutional feedrate for positioning axes/spindles • MD32040 $MA_JOG_REV_VELO_RAPID Revolutional feedrate for JOG with rapid traverse override • MD32050 $MA_JOG_REV_VELO Revolutional feedrate for JOG mode
Additional references	• Function Manual Axes and Spindles; Section "Feedrates" > "Path feedrate F" > "Type of feed G93, G94, G95"

E_MeasAct (DB31, ... DBX62.3)

Description	Measurement active The interface signal shows the instantaneous measurement status of the axis and can be evaluated for all measuring functions.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	A measuring function is not active.
Value TRUE	A measuring function is active.
Additional references	• Function Manual Technologies; Section "Measuring"

E_ActTravFStop (DB31, ... DBX62.4)

Description	Activate travel to fixed stop The feedback signal is realized via "LBP_Axis*.A_EnabTravFixedStop" (DB31, ... DBX3.1) (enable travel to fixed stop)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No request
Value TRUE	Request to enable travel to fixed stop
Corresponds to	• "LBP_Axis*.A_EnabTravFixedStop" (DB31, ... DBX3.1) Enable travel to fixed stop
Additional references	• Function Manual Axes and Spindles, Section "Travel to fixed stop"

E_FixedStop (DB31, ... DBX62.5)

Description	Fixed stop reached **Application example** So that a programmable clamping torque can be specified, for analog drives, the signal is used to switch the actuator drive from closed-speed into closed-loop current or torque controlled operation.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Fixed stop has not been reached.
Value TRUE	Fixed stop has been reached.
Corresponds to	• "LBP_Axis*.A_AckFixedStop" (DB31, ... DBX1.1) Acknowledge fixed stop reached • "LBP_Axis*.A_SensorFixedStop" (DB31, ... DBX1.2) Fixed stop sensor • "LBP_Axis*.A_EnabTravFixedStop" (DB31, ... DBX3.1) Enable travel to fixed stop • "LBP_Axis*.E_ActTravFStop" (DB31, ... DBX62.4) Activate travel to fixed stop • "LBP_Axis*.E_FixedStop" (DB31, ... DBX62.5) Fixed stop reached
Additional references	• Function Manual Axes and Spindles, Section "Travel to fixed stop"

E_ForceFixedStop (DB31, ... DBX62.6)

Description	Travel to fixed stop: Force limiting active
Signal flow	NC → PLC
Data type	BOOL

E_AxContainerRot (DB31, ... DBX62.7)

Description	Axis container: Rotation active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	An axis container rotation is not active for the axis.
Value TRUE	An axis container rotation is active for the axis.

E_AxReset (DB31, ... DBX63.0)

Description	Reset executed Reset status: • The machine data of the axis is reloaded. • The axis status is at "Single axis in reset": $AA_SNGLAX_STAT == TRUE • "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) (axis stop active) == FALSE
Signal flow	NC → PLC

Data type	BOOL
Update	Cyclic
Value FALSE	The PLC-controlled axis is not in the reset state.
Value TRUE	The PLC-controlled axis is in the reset state.
Corresponds to	• "LBP_Axis*.A_AxReset" (DB31, ... DBX28.1) PLC-controlled axis: Reset • "LBP_Axis*.E_AxStop" (DB31, ... DBX63.2) Axis stop active • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

E_PLCCtrlAx (DB31, ... DBX63.1)

Description	PLC-controlled axis
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is controlled by the NC.
Value TRUE	Axis control is transferred to the PLC.
Corresponds to	• "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) Request, PLC-controlled axis • System variable: $AA_SNGLAX_STAT Status of the single axis
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

E_AxStop (DB31, ... DBX63.2)

Description	Axis stop active After stopping, the axis is in the "Single axis is interrupted" status: $AA_SNGLAX_STAT == 3
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The PLC-controlled axis is stopped.
Value TRUE	The PLC-controlled axis is stopped as a result of "LBP_Axis*.A_Stop" (DB31, ... DBX28.6) = TRUE.

Corresponds to	• "LBP_Axis*.A_Stop" (DB31, ... DBX28.6) PLC-controlled axis: Stop along braking ramp • "LBP_Axis*.A_PLCAxis" (DB31, ... DBX28.7) PLC-controlled axis • "LBP_Axis*.A_ExactCoarse" (DB31, ... DBX60.6) Position reached with exact stop coarse • "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) Position reached with exact stop fine • System variable: $AA_SNGLAX_STAT Status of the single axis • OPI variable: aaSnglAxStat
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

E_AxSpDisable (DB31, ... DBX63.3)

Description	Axis/spindle inhibit active
Signal flow	NC → PLC
Data type	BOOL

E_StopDEPMCS (DB31, ... DBX63.4)

Description	Stop: DEPMCS active
Signal flow	NC → PLC
Data type	BOOL

E_StopDEPBCS (DB31, ... DBX63.5)

Description	Stop: DEPBCS active
Signal flow	NC → PLC
Data type	BOOL

E_StopCorr (DB31, ... DBX63.6)

Description	Stop: Corr. active
Signal flow	NC → PLC
Data type	BOOL

E_StopHIAxMove (DB31, ... DBX63.7)

Description	Stop: HIAxMove active
Signal flow	NC → PLC
Data type	BOOL

E_HW1, E_HW2, E_HW3

(DB31, ... DBX64.0, 64.1, 64.2)

<table>
<tr><td>Description</td><td>Handwheel active
The interface can be interpreted either bit or binary-coded. The definition is realized using machine data MD11324.
Bit coded: maximum 3 handwheels
<table>
<tr><th>E_HW3
Bit 2</th><th>E_HW2
Bit 1</th><th>E_HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>3</td></tr>
</table>
Bit coded: maximum 6 handwheels
<table>
<tr><th>E_HW3
Bit 2</th><th>E_HW2
Bit 1</th><th>E_HW1
Bit 0</th><th>Number of the assigned handwheel</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>No handwheel assignment</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
</table>
If the assignment is active, the machine axis can be traversed with the handwheel in JOG mode or a DRF offset can be generated in AUTOMATIC or MDI mode.
Note
At any one time, the machine axis can only be assigned to one handwheel. If, for bit coding, several interface signals are set simultaneously, the following priority applies: "Handwheel 1" before "Handwheel 2" before "Handwheel 3".</td></tr>
<tr><td>Signal flow</td><td>NC → PLC</td></tr>
<tr><td>Data type</td><td>BOOL</td></tr>
<tr><td>Update</td><td>Cyclic</td></tr>
<tr><td>Corresponds to</td><td>• "LBP_Axis*.A_HW*" (DB31, ... DBX4.0, 4.1, 4.2)
Activate handwheel
• MD11324 $MN_HANDWH_VDI_REPRESENTATION
Representation of the handwheel number at the VDI interface</td></tr>
<tr><td>Additional references</td><td>• Function Manual Axes and Spindles; Section "Manual traversing"</td></tr>
</table>

E_TReqMinus, E_TReqPlus

(DB31, ... DBX64.4, 64.5)

Description	Traverse request "Plus" / "Minus" Depending on the mode selected, the traverse request is initiated in different ways: • JOG mode "Plus" or "Minus" traversing key • REF mode Traversing key that initiates traversing motion in the direction of the reference point. • AUTOMATIC or MDI mode A program block with a traversing operation is executed for a geometry axis.	
	E_TReqMinus (bit 4)	Traverse command "Minus", traverse request in the negative axis direction
	E_TReqPlus (bit 5)	Traverse command "Plus" traverse request in the positive axis direction
Signal flow	NC → PLC	
Data type	BOOL	
Update	Cyclic	
Corresponds to	• "LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) Traversing keys "Plus" / "Minus" • "LBP_Axis*.E_TCMinus", "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.6, 64.7) Traverse command "Plus" / "Minus"	
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"	

E_TCMinus, E_TCPlus

(DB31, ... DBX64.6, 64.7)

Description	Traverse command "Plus" / "Minus" The traverse command is output depending on MD17900 $MN_VDI_FUNCTION_MASK, bit 0.	
	E_TCMinus (bit 6)	Traverse command "Minus" for traversing in the negative axis direction
	E_TCPlus (bit 7)	Traverse command "Plus" for traversing in the positive axis direction
	Application example Releasing the axis clamp when the traverse command is identified. **Note** For axes where clamping is not released until a traverse command is detected, the continuous-path mode (G64) is not possible.	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Cyclic	
Corresponds to	• "LBP_Axis*.A_Minus", "LBP_Axis*.A_Plus" (DB31, ... DBX4.6, 4.7) Traversing keys "Plus" / "Minus" • "LBP_Axis*.E_TReqMinus", "LBP_Axis*.E_TReqPlus" (DB31, ... DBX64.4, 64.5) Traverse request "Plus" / "Minus" • MD17900 $MN_VDI_FUNCTION_MASK Setting for VDI signals	
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"	

E_INC1, E_INC10, E_INC100, E_INC1000, E_INC10000, E_INCVar, E_ContManTravelAct

(DB31, ... DBX65.0, 65.1, 65.2, 65.3, 65.4, 65.5, 65.6)

Description	Active machine function There is a signal for every machine function to manually traverse the machine axis in the JOG mode:	
	E_INC1 (bit 0)	INC1
	E_INC10 (bit 1)	INC10
	E_INC100 (bit 2)	INC100
	E_INC1000 (bit 3)	INC1000
	E_INC10000 (bit 4)	INC10000
	E_INCVar (bit 5)	INCvar
	E_ContManTravelAct (bit 6)	Continuous manual travel
	Note Depending on the machine function, the response when actuating the traversing key or the handwheel differs.	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Cyclic	
Value FALSE	The machine function to manually traverse the machine axis is not active .	
Value TRUE	The machine function to manually traverse the machine axis is active.	
Corresponds to	• "LBP_Axis*.A_INC1", "LBP_Axis*.A_INC10", "LBP_Axis*.A_INC100", "LBP_Axis*.A_INC1000", "LBP_Axis*.A_INC10000", "LBP_Axis*.A_INCVar", "LBP_Axis*.A_ContManTravel" (DB31, ... DBX5.0 ... 5.6) Machine function request	
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"	

E_OEMAxis (DB31, ... DBX66.0)

Description	MCS coupling: Collision protection active **Note**: The interface signal must be activated with: MD63543 $MD_CC_PROTECT_OPTIONS, bit 7 = TRUE
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	Collision protection is not active.
Value TRUE	Collision protection is active.

Corresponds to	• (DB31, ... DBX24.2) MCS coupling: deactivate or do not permit • (DB31, ... DBX24.3) MCS coupling: Activate collision protection • "LBP_Axis*.E_OEMAxis" (DB31, ... DBX66.0) MCS coupling: Collision protection active • (DB31, ... DBX97.0) MCS coupling: Following axis • (DB31, ... DBX97.1) MCS coupling: Coupling active • (DB31, ... DBX97.2) MCS coupling: Mirroring active • (DB31, ... DBX97.3) MCS coupling: Offset change
Additional references	• Function Manual Technologies, Section: "MCS coupling"

E_InvHWDirOfRot (DB31, ... DBX67.0)

Description	Handwheel direction of rotation inversion active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The inversion of the direction of rotation of the handwheel assigned to the machine axis is not active.
Value TRUE	The inversion of the direction of rotation of the handwheel assigned to the machine axis is active.
Corresponds to	• "LBP_Axis*.A_InvHWDirOfRot" (DB31, ... DBX7.0) Invert handwheel direction of rotation
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

E_NCASpChanA, E_NCASpChanB, E_NCASpChanC, E_NCASpChanD, E_PLCType, E_ChPoss, E_NeutrASp, E_PLCASp

(DB31, ... DBX68.0, 68.1, 68.2, 68.3, 68.4, 68.5, 68.6, 68.7)

<table>
<tr><td>Description</td><td colspan="6">Status axis/spindle exchange
The current status of an axis with regard to axis/spindle replacement can be read by the PLC user program via this variable.</td></tr>
<tr><td></td><td>Bit</td><td colspan="5">Meaning</td></tr>
<tr><td></td><td>0 ... 3</td><td colspan="5">Name of the channel to which the axis/spindle is assigned (binary code).
Example: The axis is assigned to channel 2.</td></tr>
<tr><td></td><td></td><td>E_NCASpChanD
Bit 3</td><td>E_NCASpC-hanD
Bit 2</td><td>E_NCASp-ChanB
Bit 1</td><td>E_NCASp-ChanB
Bit 0</td><td>Channel num-ber</td></tr>
<tr><td></td><td></td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td></td><td>4</td><td colspan="5">New axis type is requested from the PLC.</td></tr>
<tr><td></td><td>5</td><td colspan="5">Axis interchange is possible.</td></tr>
<tr><td></td><td>6</td><td colspan="5">Axis is "neutral axis/spindle".</td></tr>
<tr><td></td><td>7</td><td colspan="5">Axis is "PLC axis/spindle".</td></tr>
<tr><td>Signal flow</td><td colspan="6">NC → PLC</td></tr>
<tr><td>Data type</td><td colspan="6">BOOL</td></tr>
<tr><td>Update</td><td colspan="6">Cyclic</td></tr>
<tr><td>Corresponds to</td><td colspan="6">• (DB31, ... DBB8)
Request axis/spindle replacement
• MD20070 $MC_AXCONF_MACHAX_USED
Machine axis number valid in channel
• MD30550 $MA_AXCONF_ASSIGN_MASTER_CHAN
Initial setting of channel for axis interchange</td></tr>
<tr><td>Additional references</td><td colspan="6">• Function Manual Basic Functions, Section: "Cross-channel axis interchange"</td></tr>
</table>

E_ParS_A, E_ParS_B, E_ParS_C

(DB31, ... DBX69.0, 69.1, 69.2)

<table>
<tr><td>Description</td><td colspan="4">Active position controller parameter set</td></tr>
<tr><td></td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td><td>Parameter set</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td></td><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td></td><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td></td><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td></td><td>1</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td></td><td>1</td><td>0</td><td>1</td><td>6</td></tr>
<tr><td></td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td></td><td>1</td><td>1</td><td>1</td><td>6</td></tr>
<tr><td></td><td colspan="4">Note
The interface is irrelevant when switchover is deactivated:
MD35590 $MA_PARAMSET_CHANGE_ENABLE == FALSE
In this case, the 1st parameter set is always active.</td></tr>
<tr><td>Signal flow</td><td colspan="4">NC → PLC</td></tr>
<tr><td>Data type</td><td colspan="4">BOOL</td></tr>
</table>

Update	Job-controlled
Corresponds to	• "LBP_Axis*.A_Par_*" (DB31, ... DBX9.0, 9.1, 9.2) Select: Position controller parameter set

E_NCUNumLink3, E_NCUNumLink4, E_NCUNumLink5, E_NCUNumLink6, E_NCUNumLink7

(DB31, ... DBX69.3, 69.4, 69.5, 69.6, 69.7)

Description	NCU number in the NCU link group
Signal flow	NC → PLC
Data type	BOOL

E_ReposSh (DB31, ... DBX70.0)

Description	REPOS offset
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No REPOS offset needs to be applied for the axis.
Value TRUE	A REPOS offset must be applied for the axis.
Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgment • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Axis*.E_PathAxis" (DB31, ... DBX76.4) Path axis • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

E_ReposShValid (DB31, ... DBX70.1)

Description	REPOS offset valid
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The REPOS offset has been calculated as invalid.
Value TRUE	The REPOS offset has been calculated as valid.
Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposDelayQuit" (DB31, ... DBX70.2) REPOS delay acknowledgment • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Axis*.E_PathAxis" (DB31, ... DBX76.4) Path axis • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

E_ReposDelayQuit (DB31, ... DBX70.2)

Description	REPOS delay acknowledgment
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The REPOS delay is still not acknowledged - or there is no REPOS offset available.

Value TRUE	The REPOS delay is acknowledged. **Note** A REPOS offset was active for the axis and "REPOS delay" was active: "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) == TRUE (REPOS delay) The axis was programmed within a traversing block, and the REPOS offset was applied. The interface signal behaves just like: "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) (REPOS approach mode acknowledgment)
Corresponds to	• "LBP_Chan*.E_MMC_REPOSMode" (DB21, ... DBX25.4) REPOS mode activation • "LBP_Chan*.A_REPOSPM_*" (DB21, ... DBX31.0, 31.1, 31.2) REPOS mode • "LBP_Chan*.A_REPOSMode" (DB21, ... DBX31.4) REPOS mode change • "LBP_Chan*.E_REPOS_EdgeAckn" (DB21, ... DBX319.0) REPOS mode change acknowledgement • "LBP_Chan*.E_REPOS_PMode*" (DB21, ... DBX319.1, 319.2, 319.3) Active REPOS mode • "LBP_Chan*.E_REPOS_DEFERRA" (DB21, ... DBX319.5) REPOS delay acknowledgement • "LBP_Axis*.A_REPOSDelay" (DB31, ... DBX10.0) REPOS delay • "LBP_Axis*.E_ReposSh" (DB31, ... DBX70.0) REPOS offset • "LBP_Axis*.E_ReposShValid" (DB31, ... DBX70.1) REPOS offset valid • "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay • "LBP_Axis*.E_PathAxis" (DB31, ... DBX76.4) Path axis • MD11470 $MN_REPOS_MODE_MASK Repositioning properties
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

E_SI_Drv_SICSCC (DB31, ... DBX70.5)

Description	DRV Safety Integrated with SIC/SCC active
Signal flow	NC → PLC
Data type	BOOL

E_BrakeTest (DB31, ... DBX71.0)

Description	Brake test active
Signal flow	NC → PLC
Data type	BOOL

E_POS_RESTORED1, E_POS_RESTORED2

(DB31, ... DBX71.4, 71.5)

Description	Position restored, measuring system 1 / 2 MD34210 $MA_ENC_REFP_STATE == 3 After control system completed system run-up, for distance-coded, incremental measuring systems the last axis position buffered before switch off is restored. Referencing does not take place automatically. The position measuring system is in the "position restored" state.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The position of measuring system 1 / 2 of the machine axis is not restored.
Value TRUE	The position of measuring system 1 / 2 of the machine axis is restored.
Corresponds to	• MD34102 $MA_REFP_SYNC_ENCS Measuring system calibration • MD34210 $MA_ENC_REFP_STATE Adjustment status absolute encoder
Additional references	• Function Manual Axes and Spindles; Section "Referencing"

E_PLCAxDedic (DB31, ... DBX71.7)

Description	PLC axis permanently assigned
Signal flow	NC → PLC
Data type	BOOL

E_MMC_REPOSDelay (DB31, ... DBX72.0)

Description	REPOS delay active **Note** If the machine axis is involved on a path ("LBP_Axis*.E_PathAxis" (DB31, ... DBX76.4) == TRUE (path axis)), then the interface signal is not active.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	A REPOS delay is not active.
Value TRUE	The REPOS delay is active. After a block search, a REPOS offset is applied for this axis. However it is not applied using the approach block, but rather using the next traversing block in which the axis is programmed.
Corresponds to	• "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

E_ModLimEnAct (DB31, ... DBX74.4)

Description	Modulo rotary axis: traversing range limits active **Note** The signal is irrelevant if linear axes/rotary axes do not have modulo functionality. **Application example** Mounted rotary axis with monitoring
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Traversing range limitation for modulo rotary axes not active.
Value TRUE	Traversing range limits (software limit switch, working area limits) active for modulo rotary axes.
Corresponds to	• "LBP_Axis*.A_ModuloLimitEn" (DB31, ... DBX12.4) Modulo rotary axis: activate traversing range limits
Additional references	• Function Manual Axes and Spindles; Section "Rotary axes"

E_JogFixPPos0Act, **E_JogFixPPos1Act**, **E_JogFixPPos2Act**

(DB31, ... DBX75.0, 75.1, 75.2)

Description	JOG approach fixed point active As soon as the "Approach fixed point in JOG" function is active, the number of the fixed point to be approached is signaled back to the PLC binary-coded with "LBP_Axis*.E_JogFixPPos*Act" (DB31, ... DBX75.0, 75.1, 75.2):			
	E_JogFixP-Pos2Act Bit 2	**E_JogFixP-Pos1Act Bit 1**	**E_JogFixP-Pos0Act Bit 0**	**Number of fixed point to be approached**
	0	0	0	-
	0	0	1	1
	0	1	0	2
	0	1	1	3
	1	0	0	4
	The selected machine axis can be traversed to the corresponding fixed point with the traversing keys or the handwheel. The fixed points are defined using machine data MD30600.			
Signal flow	NC → PLC			
Data type	BOOL			
Update	Cyclic			
Corresponds to	• "LBP_Axis*.A_JogFixPPos*" (DB31, ... DBX13.0, 13.1, 13.2) JOG fixed point approach • "LBP_Axis*.E_JogFixPPos*" (DB31, ... DBX75.3, 75.4, 75.5) JOG fixed point approach reached • MD30600 $MA_FIX_POINT_POS[<n>] Fixed value positions of the axis			
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"			

E_JogFixPPos0, E_JogFixPPos1, E_JogFixPPos2

(DB31, ... DBX75.3, 75.4, 75.4)

Description	JOG approach fixed point reached Once the axis for "Approach fixed point in JOG" has reached the fixed point position with "Exact stop fine", using "LBP_Axis*.E_JogFixPPos*" (DB31, ... DBX75.3, 75.4, 75.5), the number of the approached fixed point is signaled back to the PLC in binary-coded form:

E_JogFixPPos2 Bit 2	**E_JogFixPPos1 Bit 1**	**E_JogFixPPos0 Bit 0**	**Number of the approached fixed point**
0	0	0	-
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

	This feedback signal is also issued if the axis reaches the fixed point position in the machine coordinate system via other methods e.g. NC program, "LBP_CtrlAxisSpindle [FC18]", or synchronized action on the setpoint side and comes to a standstill on the actual value side within the "Exact stop fine" tolerance window.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_Axis*.A_JogFixPPos*" (DB31, ... DBX13.0, 13.1, 13.2) JOG fixed point approach • "LBP_Axis*.E_JogFixPPos*Act" (DB31, ... DBX75.0, 75.1, 75.2) JOG fixed point approach • MD30600 $MA_FIX_POINT_POS[<n>] Fixed value positions of the axis • MD36010 $MA_STOP_LIMIT_FINE Exact stop fine
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

E_JogToPos (DB31, ... DBX75.6)

Description	JOG travel to position active The selected machine axis can be traversed to the position specified with setting data SD43320 using the traversing keys or the handwheel.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The function "Position travel in JOG" is not active.
Value TRUE	The function "Position travel in JOG" is active.
Corresponds to	• "LBP_Axis*.A_JogToPos" (DB31, ... DBX13.3) JOG travel to position • "LBP_Axis*.E_JogPos" (DB31, ... DBX75.7) JOG position reached • SD43320 $SA_JOG_POSITION JOG position
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

E_JogPos (DB31, ... DBX75.7)

Description	JOG position reached Approaching the position is started with the traverse keys or the handwheel. The axis traverses until it comes to an automatic standstill at the position to be approached.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	For "Position travel in JOG", the axis has still not reached the position specified with SD43320.
Value TRUE	For "Position travel in JOG", the axis has reached the position specified with SD43320 with "Exact stop fine".
Corresponds to	• "LBP_Axis*.A_JogToPos" (DB31, ... DBX13.3) JOG travel to position • "LBP_Axis*.E_JogToPos" (DB31, ... DBX75.6) JOG travel to position active • SD43320 $SA_JOG_POSITION JOG position
Additional references	• Function Manual Axes and Spindles; Section "Manual traversing"

E_ScratchPulse (DB31, ... DBX76.0)

Description	Lubrication pulse As soon as the axis/spindle has moved through the traversing distance set in machine data, then the interface signal is inverted. The position measurement is always restarted after control system run-up.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	The traversing path parameterized in machine data (MD33050 $MA_LUBRICATION_DIST) was traversed.
Edge change 1 → 0	The traversing path parameterized in machine data (MD33050 $MA_LUBRICATION_DIST) was traversed.
Corresponds to	• MD33050 $MA_LUBRICATION_DIST Lubrication pulse distance

E_PathAxis (DB31, ... DBX76.4)

Description	Path axis **Note** In conjunction with block search type 5 (SERUPRO) in the state: "Target block found", the interface signal refers to the property of the axis in the target block.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is not a path axis.
Value TRUE	The axis is not a path axis. This means that it is traversed with other axes along a path (path axis).

Corresponds to	• "LBP_Axis*.E_MMC_REPOSDelay" (DB31, ... DBX72.0) REPOS delay
Additional references	• Function Manual Basic Functions; Section "Mode group, channel, program operation, reset response" > "Block search type 5 (SERUPRO)" > "Repositioning (REPOS)"

E_PosAxis (DB31, ... DBX76.5)

Description	Positioning axis
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is not a positioning axis.
Value TRUE	The axis is a positioning axis.
Corresponds to	• "LBP_Axis*.E_FPosAxis" (DB31, ... DBD78) Feedrate, positioning axis
Additional references	• Function Manual Axes and Spindles; Section "Positioning axes"

E_IndexAxisPos (DB31, ... DBX76.6)

Description	Indexing axis in position Signal is irrelevant for axes that have not been defined as indexing axes. **Application example** Tool magazine: Activation of the gripper to remove the tool from the magazine is initiated as soon as the indexing axis is in position. The PLC user program must ensure this happens. **Special cases or errors** The axis positions entered in the indexing position table for the individual divisions can be changed using work offsets (including DRF). The interface signal "Indexing axis in position" is set if the actual position of the indexing axis takes on the position value entered into the indexing table plus the offset. If a DRF offset is applied to an indexing axis in AUTOMATIC mode, the "Indexing axis in position" interface signal remains active even though the axis is no longer at an indexing position.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	In the following cases, the signal is FALSE: • The axis is not defined as an indexing axis. • A traverse command is active, and the indexing axis is traversed. • The indexing axis is located at a position which is not an indexing position. • In the AUTOMATIC mode, the indexing axis was not positioned using `CAC`, `CACP`, `CACN`, `CDC` or `CIC` but instead is traversed to any position, for example with `AC` or `DC`. • The "Controller enable" of the indexing axis is withdrawn.
Value TRUE	In the following cases, the signal is TRUE: • The indexing axis has reached the indexing position with "Exact stop fine". • The indexing axis is located at the indexing position, which was approached in the AUTOMATIC mode with `CAC`, `CACP`, `CACN`, `CDC` or `CIC`.

Corresponds to	• "LBP_Axis*.E_TCMinus", "LBP_Axis*.E_TCPlus" (DB31, ... DBX64.6, 64.7) Traverse command "Plus" / "Minus" • "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • MD30500 $MA_INDEX_AX_ASSIGN_POS_TAB Axis is an indexing axis
Additional references	• Function Manual Axes and Spindles; Section "Indexing axes"

E_CurvePos (DB31, ... DBX76.7)

Description	Rounding axis in position
Signal flow	NC → PLC
Data type	BOOL

E_CollCheckRedSpeed (DB31, ... DBX77.0)

Description	Collision avoidance: Velocity reduction
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The traversing velocity of the axis is not reduced by the collision avoidance.
Value TRUE	The traversing velocity of the axis is reduced by the collision avoidance.
Additional references	• Function Manual Monitoring and Compensating; Section "Collision avoidance"

E_FPosAxis (DB31, ... DBD78)

Description	Feedrate, positioning axis The feedrate value of the axis can be read via the interface if this axis is traversed as positioning axis. The output time is specified using machine data: MD22240 $MC_AUXFU_F_SYNC_TYPE **Notes** • As default setting, the output is suppressed, because if the feedrate is output (auxiliary function output) in the continuous-path mode, then the velocity can drop. • The last feedrate value is kept until it is overwritten by a new value.
Signal flow	NC → PLC
Data type	REAL
Update	Cyclic
Corresponds to	• "LBP_Axis*.E_PosAxis" (DB31, ... DBX76.5) Positioning axis • MD22240 $MC_AUXFU_F_SYNC_TYPE Output time of F functions • MD32060 $MA_POS_AX_VELO Initial setting for positioning axis velocity
Additional references	• Function Manual Axes and Spindles; Section "Indexing axes"

E_SetpGearA, **E_SetpGearB, E_SetpGearC**

(DB31, ... DBX82.0, 82.1, 82.2)

Description	Set gear stage Gear stage requested from the NC, which must be selected at the machine.			
	E_SetpGearC Bit 2	**E_SetpGearB Bit 1**	**E_SetpGearA Bit 0**	**Meaning**
	---	---	---	Axis mode
	0	0	0	Gear stage 1
	0	0	1	Gear stage 1
	0	1	0	Gear stage 2
	0	1	1	Gear stage 3
	1	0	0	Gear stage 4
	1	0	1	Gear stage 5
	1	1	0	Gear stage 5
	1	1	1	Gear stage 5
Signal flow	NC → PLC			
Data type	BOOL			
Update	Cyclic			
Corresponds to	• "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) Actual gear stage • "LBP_Axis*.A_GearChangeOv" (DB31, ... DBX16.3) Gear stage has been changed • "LBP_Axis*.A_OscilSpeed" (DB31, ... DBX18.5) Oscillation speed • "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3) Change gear stage			
Additional references	• Function Manual Axes and Spindles; Section "Spindles"			

E_GearChange (DB31, ... DBX82.3)

Description	Change gear stage **Note** The gear is only changed over to the new set gear stage if the following applies: Set gear stage <> actual gear stage
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Edge change 0 → 1	There is an active requested to changeover the gear to the set gear stage.
Edge change 1 → 0	No effect.
Corresponds to	• "LBP_Axis*.A_ActGear*" (DB31, ... DBX16.0, 16.1, 16.2) Actual gear stage • "LBP_Axis*.E_SetpGear*" (DB31, ... DBX82.0, 82.1, 82.2) Set gear stage
Additional references	• Function Manual Axes and Spindles; Section "Spindles" > "Configurable gear adaptations" > "Gear stages for spindles and gear stage change"

E_SpeedLimit (DB31, ... DBX83.0)

Description	Speed limit exceeded The speed limit is exceeded if the following applies Actual speed > (MD35100 $MA_SPIND_VELO_ LIMIT + MD35150 $MA_SPIND_DES_VELO_TOL)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The speed limit is not exceeded.
Value TRUE	The speed limit is exceeded.
Corresponds to	• MD35150 $MA_SPIND_DES_VELO_TOL Spindle speed tolerance • MD35100 $MA_SPIND_VELO_LIMIT Maximum spindle speed
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_SetSpeedLimit (DB31, ... DBX83.1)

Description	Setpoint speed limited The NC has automatically limited the setpoint speed because it exceeds the effective maximum limit value: • MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT • MD35100 $MA_SPIND_VELO_LIMIT • "LBP_Axis*.A_VeloSpeedLimit" (DB31, ... DBX3.6) • G26 • LIMS • VELOLIM **Notes** • A possible response in the PLC user program would be to inhibit/lock the path feedrate: "LBP_Chan*.A_FDdisable" (DB21, ... DBX6.0) = TRUE (feedrate disable) • The interface signal "LBP_Axis*.E_SetRange" (DB31, ... DBX83.5) (spindle in the setpoint range) is updated. **Safety Integrated** Depending on the active safety speed level SG1 ... SGn, the following machine data should be taken into account: • MD36933 $MA_SAFE_DES_VELO_LIMIT Example: All standard limit values are greater than 1500 rpm. • SG1 is active • MD36933 $MA_SAFE_DES_VELO_LIMIT[<SG1>] = 90% Programming: M3 S1500 The speed setpoint is limited to 1000 rpm (MD36932 * MD36933) ⇒ "LBP_Axis*.E_SetSpeedLimit" (DB31, ... DBX83.1) = TRUE
Signal flow	NC → PLC
Data type	BOOL

Update	Cyclic
Value FALSE	The setpoint speed is not limited.
Value TRUE	The setpoint speed is limited.
Corresponds to	• "LBP_Chan*.A_FDdisable" (DB21, ... DBX6.0) Feed disable • "LBP_Axis*.A_FDSpStop" (DB31, ... DBX4.3) Feed/spindle stop • "LBP_Axis*.E_SetSpeedIncreased" (DB31, ... DBX83.2) Speed setpoint increased • "LBP_Axis*.E_SetRange" (DB31, ... DBX83.5) Spindle in setpoint range • "LBP_Axis*.A_VeloSpeedLimit" (DB31, ... DBX3.6) Spindle speed limitation to MD35160 $MA_SPIND_EXTERN_VELO_LIMIT • MD35100 $MA_SPIND_VELO_LIMIT Maximum spindle speed • MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT Maximum speed of gear stage • MD36933 $MA_SAFE_DES_VELO_LIMIT SG setpoint speed limit • G26 Upper spindle speed limit • LIMS Speed limit for the master spindle when G96/G961/G97 is active • VELOLIM Programmed spindle speed limit in open-loop speed controlled mode
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_SetSpeedIncreased (DB31, ... DBX83.2)

Description	Setpoint speed increased The NC has automatically increased the setpoint speed, as it had fallen below the effective minimum limit value: • MD35120 $MA_GEAR_STEP_MIN_VELO • MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT • G25 **Notes** • A possible response in the PLC user program would be to inhibit/lock the path feedrate: "LBP_Chan*.A_FDdisable" (DB21, ... DBX6.0) = TRUE (feedrate disable) • The interface signal "LBP_Axis*.E_SetRange" (DB31, ... DBX83.5) (spindle in the setpoint range) is updated.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The setpoint speed was not increased.
Value TRUE	The setpoint speed was increased.

Corresponds to	• "LBP_Chan*.A_FDdisable" (DB21, ... DBX6.0) Feed disable • "LBP_Axis*.A_FDSpStop" (DB31, ... DBX4.3) Feed/spindle stop • "LBP_Axis*.E_SetSpeedLimit" (DB31, ... DBX83.1) Speed setpoint limited • "LBP_Axis*.E_SetRange" (DB31, ... DBX83.5) Spindle in setpoint range • MD35120 $MA_GEAR_STEP_MIN_VELO Minimum speed for automatic gear stage selection M40 • MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT Minimum speed of the gear stage • G25 Lower spindle speed limitation
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_GeoMonit (DB31, ... DBX83.3)

Description	Geometry monitoring The geometry monitoring is a subfunction of the grinding-specific tool monitoring. The actual wheel radius and the actual wheel width are monitored.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	No error in the grinding wheel geometry
Value TRUE	Error in the grinding wheel geometry The actual wheel radius falls below the value specified in parameter $TC_TPG3 or the actual wheel width ($TC_TPG5) falls below the value defined in parameter $TC_TPG4. **Note** There is no controller-internal error response. Required responses must be programmed by the PLC user.
Additional references	• Function Manual Tool Offset; Section "Grinding-specific tool offset and tool monitoring"

E_SupportAreaViol (DB31, ... DBX83.4)

Description	Support range limits violated
Signal flow	NC → PLC
Data type	BOOL

E_SetRange (DB31, ... DBX83.5)

Description	Spindle in setpoint range The spindle speed is in the setpoint range if the following applies: | Actual speed - setpoint speed | < MD35150 $MA_SPIND_DES_VELO_TOL **Note** It is normal if the spindle speed is not in the setpoint range while accelerating or braking.
Signal flow	NC → PLC
Data type	BOOL

Update	Cyclic
Value FALSE	The spindle speed is not in the setpoint range.
Value TRUE	The spindle speed is in the setpoint range.
Corresponds to	• "LBP_Chan*.A_FDdisable" (DB21, ... DBX6.0) Feed disable • "LBP_Axis*.A_FDSpStop" (DB31, ... DBX4.3) Feed/spindle stop • "LBP_Axis*.E_SetSpeedLimit" (DB31, ... DBX83.1) Speed setpoint limited • "LBP_Axis*.E_SetSpeedIncreased" (DB31, ... DBX83.2) Speed setpoint increased
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_SpeedMonit (DB31, ... DBX83.6)

Description	Speed monitoring The speed monitoring is a subfunction of the grinding-specific tool monitoring. The speed monitor checks the grinding wheel peripheral speed in ms^{-1} (parameter $TC_TPG7) as well as the maximum spindle speed in rpm (parameter $TC_TPG6). The speed setpoint is monitored against the speed limitation cyclically with allowance for the spindle override.
Signal flow	NC → PLC
Data type	BOOL
Value FALSE	No error in grinding wheel speed
Value TRUE	Error in grinding wheel speed The speed limit value is reached, the speed is limited to the limit value. **Note** There is no controller-internal error response. Required responses must be programmed by the PLC user.
Additional references	• Function Manual Tool Offset; Section "Grinding-specific tool offset and tool monitoring"

E_ActRotRight (DB31, ... DBX83.7)

Description	Actual direction of rotation clockwise The actual direction of rotation is derived from the spindle position measuring encoder. **Note** The interface signal is irrelevant for: • "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) == TRUE (axis/spindle stationary) • Spindles without position encoder
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Actual direction of rotation: counterclockwise
Value TRUE	Actual direction of rotation: clockwise
Corresponds to	• "LBP_Axis*.E_Stat" (DB31, ... DBX61.4) axis/spindle stationary
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_ConstCuttSpeed (DB31, ... DBX84.0)

Description	Constant cutting speed active
Signal flow	NC → PLC
Data type	BOOL

E_SUG (DB31, ... DBX84.1)

Description	Grinding wheel peripheral speed active If the function is active, then all S value inputs from the PLC are interpreted as the grinding wheel peripheral speed.
Signal flow	NC → PLC
Data type	BOOL
Value FALSE	The function "Constant grinding wheel peripheral speed (GWPS)" is not active.
Value TRUE	The function "constant grinding wheel peripheral speed (GWPS)" is active.
Additional references	• Function Manual Tool Offset; Section "Grinding-specific tool offset and tool monitoring"

E_CLG (DB31, ... DBX84.2)

Description	CLGON active
Signal flow	NC → PLC
Data type	BOOL

E_Tapping (DB31, ... DBX84.3)

Description	Tapping without compensating chuck active If "Rigid tapping without compensating chuck" (G331/G332) is active, then internally the spindle is switched over to closed-loop position controlled axis mode. **Notice** Setting the following interface signals while tapping without compensating chuck (rigid tapping) will destroy the thread: • "LBP_ModeGroup.A_MGReset" (DB31, ... DBX0.7) (mode group reset) = TRUE • "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) (channel reset) = TRUE • "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) (controller enable) = FALSE • "LBP_Axis*.A_FDSpStop" (DB31, ... DBX4.3) (feed stop) = TRUE
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Rigid tapping without compensating chuck is not active
Value TRUE	Rigid tapping without compensating chuck is active

Corresponds to	• "LBP_ModeGroup.A_MGReset" (DB31, ... DBX0.7) Mode group reset • "LBP_Chan*.A_Reset" (DB21, ... DBX7.7) Channel reset • "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • "LBP_Axis*.A_FDSpStop" (DB31, ... DBX4.3) Feed stop
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_SyncMode (DB31, ... DBX84.4)

Description	Active spindle mode: Synchronous mode In synchronous mode, the following spindle tracks the motion of the leading spindle with respect to the ratio and the monitoring functions are active for coarse and fine synchronism. When the coupling is switched off (synchronous mode deselected), the following spindle is switched to "control mode".
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The spindle is not operated as a following spindle in "synchronous mode".
Value TRUE	The spindle is in the "synchronous mode" spindle mode. **Note** The signal is only set for the machine axis which is active as following spindle: "LBP_Axis*.E_SlaveSp" (DB31, ... DBX99.1) (following spindle active) == TRUE
Corresponds to	• "LBP_Axis*.E_SyncronFine" (DB31, ... DBX98.0) Fine synchronous operation • "LBP_Axis*.E_SyncronCoarse" (DB31, ... DBX98.1) Coarse synchronous operation • "LBP_Axis*.E_SlaveSp" (DB31, ... DBX99.1) Following spindle active
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_PosMode (DB31, ... DBX84.5)

Description	Active spindle mode: Positioning mode
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Positioning mode (SPOS or SPOSA) is not active.
Value TRUE	Positioning mode (SPOS or SPOSA) is active.
Corresponds to	• "LBP_Axis*.E_OscillMode" (DB31, ... DBX84.6) Spindle mode, oscillation mode • "LBP_Axis*.E_ContrMode" (DB31, ... DBX84.7) Spindle mode, control mode
Additional references	• Function Manual Axes and Spindles; Section "Spindles" > "Configurable gear adaptations" > "Gear stage change to fixed position"

E_OscillMode (DB31, ... DBX84.6)

Description	Active spindle mode: Oscillation mode **Note** When the gear stage is changed over, the spindle automatically goes into the oscillation mode.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Oscillation mode is not active.
Value TRUE	Oscillation mode is active.
Corresponds to	• "LBP_Axis*.E_GearChange" (DB31, ... DBX82.3) Change gear stage • "LBP_Axis*.E_PosMode" (DB31, ... DBX84.5) Spindle mode, positioning mode • "LBP_Axis*.E_ContrMode" (DB31, ... DBX84.7) Spindle mode, control mode
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_ContrMode (DB31, ... DBX84.7)

Description	Active spindle mode: Control mode The spindle is in control mode for the following functions: • Spindle direction of rotation specified by M3/M4 or spindle stop M5 • M41…M45, or automatic gear stage change M40
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Control mode is not active.
Value TRUE	Control mode is active.
Corresponds to	• "LBP_Axis*.E_PosMode" (DB31, ... DBX84.5) Spindle mode, positioning mode • "LBP_Axis*.E_OscillMode" (DB31, ... DBX84.6) Spindle mode, oscillation mode
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_ToolDynLimit (DB31, ... DBX85.0)

Description	Tool with dynamic response limitation
Signal flow	NC → PLC
Data type	BOOL

E_SpInPosition (DB31, ... DBX85.5)

Description	Spindle in position Precondition for the output of the interface signal: • "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) == TRUE (exact stop fine) **UND** • Programmed set position is reached **Note** • The interface signal is only processed in the positioning mode ("LBP_Axis*.E_PosMode" (DB31, ... DBX84.5) == TRUE), e.g.: – NC program: `SPOS`, `SPOSA` and `M19` – Synchronized action: `SPOS` and `M19` – PLC user program: Positioning with "LBP_CtrlAxisSpindle [FC18]" or "LBP_Axis*.A_Pos-Spindle" (DB31, ... DBX30.4) • The interface signal remains set if the spindle is already at the programmed set position.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The spindle is not in position.
Value TRUE	The spindle is in position.
Corresponds to	• "LBP_Axis*.A_PosSpindle" (DB31, ... DBX30.4) Spindle positioning start • "LBP_Axis*.A_ExactFine" (DB31, ... DBX60.7) Exact stop fine • "LBP_Axis*.E_PosMode" (DB31, ... DBX84.5) Spindle mode, positioning mode • SPOS • SPOSA • M19
Additional references	Function Manual Axes and Spindles; Section "Spindles" > • "Modes" > "Positioning mode" > "Signal "Spindle in position" for tool change" • "Programming" > "Special spindle motion via PLC interface"

E_MFunct (DB31, ... DBW86)

Description	M function for spindle The M function programmed for the spindle in the NC program is output.	
	Value	**M function**
	3	M3 (clockwise spindle rotation)
	4	M4 (counterclockwise spindle rotation)
	5	M5 (spindle stop)
	19	M19 (spindle positioning to the position entered in SD43240)
	70	M70 (switchover to axis mode)
Signal flow	NC → PLC	
Data type	INT	

Update	Job-controlled
Corresponds to	• "LBP_Chan*.M*Change" (DB21, ... DBX58.0 ... 58.4) M function: Change • "LBP_Chan*.M*NDec" (DB21, ... DBX59.0 ... 59.4) M function: Not decoded • "LBP_Chan*.ExtM1", "LBP_Chan*.ExtM2", ..., "LBP_Chan*.ExtM5", "LBP_Chan*.M1", "LBP_Chan*.M2", ..., "LBP_Chan*.M5" (DB21, ... DBB68 ... 97) M functions for spindles

E_SFunct (DB31, ... DBD88)

Description	S function for spindle The spindle-specific S function programmed in the NC program is output. The following S functions are output: • S... as spindle speed in rpm (programmed value) • S... as constant cutting speed in m/min or ft/min The following S functions are not output: • S... as the programmed spindle speed limiting G25 • S... as the programmed spindle speed limiting G26 • S... as spindle speed in rpm if a spindle was not defined in the control system • S... as the dwell time in spindle revolutions
Signal flow	NC → PLC
Data type	REAL
Update	Job-controlled
Corresponds to	• "LBP_Chan*.S*Change" (DB21, ... DBX60.0 ... 60.2) S function: Change • "LBP_Chan*.S*Quick" (DB21, ... DBX60.4 ... 60.6) S function: Quick • "LBP_Chan*.ExtS1", "LBP_Chan*.ExtS2", "LBP_Chan*.ExtS3", "LBP_Chan*.S1", "LBP_Chan*.S2", "LBP_Chan*.S3" (DB21, ... DBB98 ... 115) S functions for spindles, channel-specific

E_RUEncDisabl (DB31, ... DBX92.1)

Description	Ramp-function generator disable
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Ramp function generator fast stop is not active for the axis drive.
Value TRUE	Ramp function generator fast stop is active for the axis drive.
Corresponds to	• "LBP_Axis*.A_RUEncQuStop" (DB31, ... DBX20.1) Ramp-function generator fast stop

E_SpeedSetpSmooth (DB31, ... DBX92.3)

Description	Speed setpoint smoothing active
Signal flow	NC → PLC
Data type	BOOL

E_RLIact (DB31, ... DBX92.4)

Description	Drive-autonomous motion active **With SINAMICS, valid for NC 62.07 and higher if a 611U telegram type is used.** Die Variable = TRUE wenn MELDW.11 == TRUE (controller enable) and ZSW1.2 == FALSE (operation enabled) **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 References SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1) **Application examples** Internal drive functions: • Rotor or pole position identification • Function generator
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Drive-autonomous motion is not active.
Value TRUE	Drive-autonomous motion is active. The axis traverses due to setpoints generated by drive-internal functions. The drive still responds to control signals of the NC, e.g. controller enable. Setpoints from the NC are ignored.
Additional references	• SINAMICS S120 Function Manual; • Pole position identification: Section "Servo control" > "Pole position identification" • Function generator: Section "Servo control" > "Optimization of the current and speed controller"

E_HoldBrakeOpened (DB31, ... DBX92.5)

Description	Motor holding brake opened
Signal flow	NC → PLC
Data type	BOOL

E_DriveRunEn (DB31, ... DBX92.7)

Description	Drive operation enabled
Signal flow	NC → PLC
Data type	BOOL

E_ParA, E_ParB, E_ParC, E_MotA, E_MotB

(DB31, ... DBX93.0, 93.1, 93.2, 93.3, 93.4)

Description	Motor/drive data set: Display Display interface for the currently effective motor (MDS) / drive data set (DDS). **Formatting** Formatting the display interface, i.e. which bits are used to address the motor data sets (MDS) – and which are used to address the drive data sets (DDS) is set via the formatting interface (DB31, ... DBX130.0 ... 130.4).
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_Axis*.A_Par*", "LBP_Axis*.A_MotA", "LBP_Axis*.A_MotB" (DB31, ... DBX21.0 ... 21.4) Motor/drive data set: Selection • "LBP_Axis*.A_MotOK" (DB31, ... DBX21.5) Motor has been selected • (DB31, ... DBX130.0 ... 130.4) Motor / drive data set: Formatting
Additional references	• Function Manual Basic Functions; Section "Various NC/PLC interface signals and functions" > "Functions" > "Switchover motor/drive data sets"

E_MotOK (DB31, ... DBX93.5)

Description	Drive ready "LBP_Axis*.E_MotOK" (DB31, ... DBX93.5) = drive: MELDW.12 **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Reference SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) In the following, the status words ZSW1 or ZSW2 only refer to PROFIDrive telegrams compatible with SIMODRIVE 611U (SIMODRIVE 611U interface mode, p2038 = TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The drive is not ready. If the signal is reset while operational, the drive is stopped (pulse inhibit or rapid stop). During system run-up, the pulses are still inhibited. In addition, the following interface signals are reset: • "LBP_NC.E_611Dready" (DB31, ... DBX108.6) = FALSE (drive ready) • "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) = FALSE (current controller active) • "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) = FALSE (speed controller active)
Value TRUE	The drive is ready.

Corresponds to	• "LBP_NC.E_611Dready" (DB31, ... DBX108.6) Drive ready • "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) Current controller active • "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) Speed controller active
Additional references	• Function Manual Basic Functions; Section "Various NC/PLC interface signals and functions" > "Functions" > "Switchover motor/drive data sets" • SINAMICS S120/S150 List Manual

E_IntegratDisable (DB31, ... DBX93.6)

Description	Integrator disable, speed controller
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The integrator of the speed controller in the drive is not inhibited. The speed controller functions as a PI controller.
Value TRUE	The integrator of the speed controller in the drive is inhibited. The speed controller functions as a P controller.
Corresponds to	• "LBP_Axis*.A_IntegratDisable" (DB31, ... DBX21.6) Integrator disable, speed controller

E_PulseEnable (DB31, ... DBX93.7)

Description	Pulses enabled "LBP_Axis*.E_PulseEnable" (DB31, ... DBX93.7) = MELDW.13 **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Reference SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) In the following, the status words ZSW1 or ZSW2 only refer to PROFIDrive telegrams compatible with SIMODRIVE 611U (SIMODRIVE 611U interface mode, p2038 = TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The drive pulses are not enabled ⇒ • "LBP_Axis*.E_CurrentContr" (DB31, ... DBX61.7) = FALSE (current controller active) • "LBP_Axis*.E_SpeedContr" (DB31, ... DBX61.6) = FALSE (speed controller active) • "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) = FALSE (position controller active)
Value TRUE	The drive pulses are enabled.
Corresponds to	• "LBP_Axis*.A_PulseEnable" (DB31, ... DBX21.7) Pulse enable

E_MTempWarn (DB31, ... DBX94.0)

Description	Motor temperature prewarning The actual motor temperature is displayed on the user interface at: Operating area "Diagnostics" > "Service display: Axis/spindle"
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The motor temperature is below the warning threshold (p0604).
Value TRUE	The motor temperature has exceeded the warning threshold (p0604) configured in the drive. **Note** If the motor temperature remains too high for longer than the parameterized time (p0606), a fault is output, the drive is stopped and the pulse enable removed. If the motor temperature falls below the warning threshold (p0604) again before the time has expired (p0606), the interface signal is reset.
Corresponds to	• "LBP_Axis*.E_HTempWarn" (DB31, ... DBX94.1) Heat sink temperature prewarning

E_HTempWarn (DB31, ... DBX94.1)

Description	Heat sink temperature prewarning The interface signals "LBP_Axis*.E_MTempWarn" (DB31, ... DBX94.0) and "LBP_Axis*.E_HTempWarn" (DB31, ... DBX94.1) are derived from the following signals of the cyclic drive telegram: • Case 1: Temperature warning in the message word – "LBP_Axis*.E_MTempWarn" (DB31, ... DBX94.0) ≙ MELDW, Bit 6 (no motor overtemperature warning) – "LBP_Axis*.E_HTempWarn" (DB31, ... DBX94.1) ≙ MELDW, Bit 7 (no thermal overload in power unit warning) • Case 2: Alarm of alarm class B (interface mode "SIMODRIVE 611u", p2038=TRUE) "LBP_Axis*.E_MTempWarn" (DB31, ... DBX94.0) == TRUE and "LBP_Axis*.E_HTempWarn" (DB31, ... DBX94.1) == TRUE, if the following applies: Cycl. drive telegram, ZSW1: Bit 11 == FALSE and 12 == TRUE (alarm class B) The interface signals are derived from the warning of warning class B if there is no specific information from the message word. An alarm is displayed: Alarm number = 200.000 + alarm value (r2124) **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 Reference SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) In the following, the status words ZSW1 or ZSW2 only refer to PROFIDrive telegrams compatible with SIMODRIVE 611U (SIMODRIVE 611U interface mode, p2038 = TRUE)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic

Value FALSE	The heat sink temperature is below the warning threshold.
Value TRUE	The heat sink temperature of the power semiconductors has exceeded the parameterized warning threshold (p0294). **Note** The parameterized reaction (p0290) is performed in the drive. If the temperature violation remains, a fault is output after approx. 20 s, the drive is stopped and the pulse enable removed.
Additional references	• S120 Commissioning Manual, Section "Commissioning" > "Temperature sensors for SINAMICS components" • S120 Function Manual, Section "Monitoring and protective functions" • S120 List Manual – MELDW, bit 6 ≙ BO: r2135.14 → function diagram: 2548, 8016 – MELDW, bit 7 ≙ BO: r2135.15 → function diagram: 2548, 2452, 2456, 8016 • SINUMERIK Diagnostics Manual

E_RUComplete (DB31, ... DBX94.2)

Description	Run-up completed
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The power up operation is still active after the speed setpoint has been changed.
Value TRUE	After a new speed setpoint, the speed actual value has reached the speed tolerance band defined using drive parameter p2164, and has not left it for the time specified in p2166. Any subsequent speed fluctuations, also outside the tolerance band, e.g. due to load changes, will not affect the interface signal.
Corresponds to	• "LBP_Axis*.E_NactNset" (DB31, ... DBX94.6) "$n_{act} = n_{set}$" • "LBP_Axis*.E_MdMdx" (DB31, ... DBX94.3) "$\|M_D\| = M_{dx}$"
Additional references	• SINUMERIK Commissioning Manual IBN CNC: NCK, PLC, Drive • SIMATIC S120 List Manual

E_MdMdx (DB31, ... DBX94.3)

Description	$\|M_d\| < M_{dx}$ "LBP_Axis*.E_MdMdx" (DB31, ... DBX94.3) = MELDW.1 **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 References SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) In the following, the status words ZSW1 or ZSW2 only refer to PROFIdrive telegrams compatible with SIMODRIVE 611U (SIMODRIVE 611U interface mode, p2038 = 1)
Signal flow	NC → PLC
Data type	BOOL

Update	Cyclic
Value FALSE	The torque setpoint $\|M_d\|$ is higher than the threshold torque M_{dx}. A motor overload can be determined via the interface signal. An appropriate response can then be initiated in the PLC user program.
Value TRUE	The current torque utilization is below the torque utilization threshold (torque threshold 2, p2194). The power up operation has been completed, the drive is in the steady state and the torque setpoint $\|M_d\|$ drive does not exceed the threshold torque M_{dx}. The torque threshold characteristic is speed-dependent. During the power up operation, "LBP_Axis*.E_MdMdx" (DB31, ... DBX94.3) ($\|M_d\| < M_{dx}$) == TRUE. The interface signal is not updated until run-up is complete ("LBP_Axis*.E_RUComplete" (DB31, ... DBX94.2) == TRUE) and the signal interlocking time for the threshold torque has expired.
Additional references	• SIMATIC S120 List Manual

E_NactNmin (DB31, ... DBX94.4)

Description	$\|n_{act}\| < n_{min}$ "LBP_Axis*.E_NactNmin" (DB31, ... DBX94.4) = MELDW.2 **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 References SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The actual speed value is greater than the minimum threshold speed n_{min}.
Value TRUE	The actual speed value n_{act} is less than n_{min} (speed threshold value 3, p2161).

E_NactNx (DB31, ... DBX94.5)

Description	$\|n_{act}\| < n_x$ "LBP_Axis*.E_NactNx" (DB31, ... DBX94.5) = MELDW.3 **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 References SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Signal flow	NC → PLC
Data type	BOOL

Update	Cyclic
Value FALSE	The actual speed value n_{act} is greater than the threshold speed n_x.
Value TRUE	The actual speed value n_{act} is less than n_x (speed threshold value 2, p2155).

E_NactNset (DB31, ... DBX94.6)

Description	$n_{act} = n_{set}$
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The actual speed value is outside the tolerance band around the speed setpoint (speed threshold value 4, p2163).
Value TRUE	The actual speed value is at least the parameterized time (switch-on delay n_act = n_set, p2167) within the tolerance band around the speed setpoint (speed threshold value 4, p2163).

E_Message (DB31, ... DBX94.7)

Description	Variable signaling function **For SINAMICS, valid for SW2.6 and higher** Using the "Variable signaling" function, BICO interconnections and parameters which have the attribute traceable can be monitored in the drive. "LBP_Axis*.E_Message" (DB31, ... DBX94.7) = MELDW.5 **Note** • SINAMICS S120: Message word (MELDW) The message word (MELDW) is only contained in PROFIdrive telegrams compatible with SIMODRIVE 611U, for example, telegrams 102, 103, 105, 106, 110, 111, 116, 118, 125, 126, 136, 138, 139 References SINAMICS List Manual, Function Diagrams 2419 and 2420 • SINAMICS S120: Status word 1 / 2 (ZSW1/2) The status words ZSW1 or ZSW2 only refer to SIMODRIVE 611U-compatible PROFIdrive telegrams in the following (SIMODRIVE 611U interface mode, p2038 = 1)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The parameterized drive variable has fallen below the specified threshold value, including hysteresis.
Value TRUE	The parameterized drive variable has exceeded the specified threshold value, including hysteresis.
Additional references	• SINAMICS S120 Function Manual, Section "Servo control" > "Variable signaling function"

E_UDClessAlThreshold (DB31, ... DBX95.0)

Description	UDC connection < Alarm threshold
Signal flow	NC → PLC
Data type	BOOL

E_I2TLimit (DB31, ... DBX95.7)

Description	Alarm of alarm class C is active In the drive, an alarm is the response to a detected potential or expected fault condition that does not cause the drive to switch off and does not have to be acknowledged.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The drive signals that no alarm of alarm class C is active.
Value TRUE	The drive signals that a warning of warning class C is active.
Additional references	• SINAMICS S120 List Manual, Section "Faults and alarms"

E_SpindleMonit (DB31, ... DBX96.0)

Description	Stepper motor: Rotation monitoring error
Signal flow	NC → PLC
Data type	BOOL

E_AxisCtrl (DB31, ... DBX96.1)

Description	Control axis active
Signal flow	NC → PLC
Data type	BOOL

E_MSFine (DB31, ... DBX96.2)

Description	Master-slave: Fine speed difference
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The speed difference between the master and slave axis lies outside the tolerance specified with MD37272.
Value TRUE	The speed difference between the master and slave axis lies inside the tolerance specified with MD37272.
Corresponds to	• "LBP_Axis*.A_TorqComp" (DB31, ... DBX24.4) Master-Slave: activate torque compensatory controller • "LBP_Axis*.A_MS" (DB31, ... DBX24.7) Master-Slave: Activate coupling • "LBP_Axis*.E_MSCoarse" (DB31, ... DBX96.3) Master-Slave: Coarse speed difference • "LBP_Axis*.E_MS" (DB31, ... DBX96.7) Master-Slave: Coupling active • MD37272 $MA_MS_VELO_TOL_FINE Master-Slave: Velocity tolerance fine
Additional references	• Function Manual Technologies, Section: "Speed/torque coupling, master-slave"

E_MSCoarse (DB31, ... DBX96.3)

Description	Master-slave: Coarse speed difference
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The speed difference between the master and slave axis lies outside the tolerance specified with MD37270.
Value TRUE	The speed difference between the master and slave axis lies inside the tolerance specified with MD37270.
Corresponds to	• "LBP_Axis*.A_MS" (DB31, ... DBX24.7) Master-Slave: Activate coupling • "LBP_Axis*.E_MSFine" (DB31, ... DBX96.2) Master-Slave: Fine speed difference • "LBP_Axis*.E_MS" (DB31, ... DBX96.7) Master-Slave: Coupling active • MD37270 $MA_MS_VELO_TOL_COARSE Master-Slave: Velocity tolerance coarse
Additional references	• Function Manual Technologies, Section: "Speed/torque coupling, master-slave"

E_MSCompContr (DB31, ... DBX96.4)

Description	Master-slave: Compensatory controller active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Torque compensatory control is not active.
Value TRUE	Torque compensatory control is active.
Corresponds to	• "LBP_Axis*.A_TorqComp" (DB31, ... DBX24.4) Master-Slave: activate torque compensatory controller • MD37254 $MA_MS_TORQUE_CTRL_MODE Interconnection torque compensatory controller
Additional references	• Function Manual Technologies, Section: "Speed/torque coupling, master-slave"

E_BitValCtrlOutChang0 (DB31, ... DBX96.5)

Description	Setpoint exchange: Drive control active The controller enable signal ("LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1)) can only be enabled if the axis has control over the drive: "LBP_Axis*.E_BitValCtrlOutChang0" (DB31, ... DBX96.5) == TRUE **Note** All axes included in setpoint exchange that currently do not have drive control, are switched to follow-up mode by the control. I.e. they are not in position control. As a consequence, we recommend that suspended axes are assigned a brake control.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis has not taken over control of the drive.

Value TRUE	The axis has taken over control of the drive.
Corresponds to	• "LBP_Axis*.A_ContrEnable" (DB31, ... DBX2.1) Controller enable • "LBP_Axis*.A_ChangSetpOutAss0" (DB31, ... DBX24.5) Setpoint exchange: Accept drive control
Additional references	• Function Manual Axes and Spindles; Section "Setpoint exchange"

E_BitValCtrlOutChang1 (DB31, ... DBX96.6)

Description	UDC connection < Alarm threshold
Signal flow	NC → PLC
Data type	BOOL

E_MS (DB31, ... DBX96.7)

Description	Master/slave: Active **Note** For a master/slave coupling, for the brake control logic of the slave axes, it is no longer permissible that the interface signal "LBP_Axis*.E_PositContr" (DB31, ... DBX61.5) (position controller active) is evaluated, as it is no longer set for an active master/slave coupling. Instead, this interface signal must be used.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The coupling is not active.
Value TRUE	The coupling is active.
Additional references	• Function Manual Technologies, Section: "Speed/torque coupling, master-slave"

E_SyncronFine (DB31, ... DBX98.0)

Description	Synchronism fine **Note** The signal is relevant only for the following spindle in synchronous mode. **Application example** Clamping of the workpiece in the following spindle at the transfer from the leading spindle: The clamping of the workpiece is only initiated by the PLC user program when the spindles are running synchronous enough.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is not within the "Fine synchronous operation" tolerance band.
Value TRUE	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is within the "Fine synchronous operation" tolerance band.

Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode • MD37210 $MA_COUPLE_POS_TOL_FINE Threshold value for "fine synchronism" • MD37230 $MA_COUPLE_VELO_TOL_FINE "Fine" speed tolerance
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_SyncronCoarse (DB31, ... DBX98.1)

Description	Synchronism coarse **Note** The signal is relevant only for the following spindle in synchronous mode. **Application example** Clamping of the workpiece in the following spindle at the transfer from the leading spindle: The clamping of the workpiece is only initiated by the PLC user program when the spindles are running synchronous enough.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is not within the "Coarse synchronous operation" tolerance band.
Value TRUE	Synchronous spindle coupling: The position deviation or velocity difference between the following spindle and its leading spindle is within the "Coarse synchronous operation" tolerance band.
Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode • MD37200 $MA_COUPLE_POS_TOL_COARSE Threshold value for "coarse synchronism" • MD37220 $MA_COUPLE_VELO_TOL_COARSE "Coarse" speed tolerance
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_ActValCoupling (DB31, ... DBX98.2)

Description	Actual value coupling **Note** The signal is relevant only for the active following spindle in synchronous mode. **Special cases or errors** If faults occur on the following spindle that cause a withdrawal of the "Controller enable" for the following spindle, under certain circumstances, the control may exchange the coupling relationship of the following spindle and leading spindle and switch to actual value coupling.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: Setpoint coupling active
Value TRUE	Synchronous spindle coupling: Actual value coupling active

Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode • MD21310 $MC_COUPLING_MODE_1 Coupling type in synchr. spindle oper
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_OverlayMotion (DB31, ... DBX98.4)

Description	Overlaid motion **Note** The signal is relevant only for the following spindle in synchronous mode. **Application examples** Example for superimposed motion of the following spindle: • Activating the synchronous mode with a defined angular offset between the following spindle and leading spindle • Activating the synchronous mode for rotating leading spindle • Changing the ratio while the synchronous mode is active • Entering a new defined angular offset when the synchronous mode is active • Traversing the following spindle in the JOG mode with the traversing keys "Plus" / "Minus" or the handwheel while synchronous mode is active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: The following spindle does not traverse through any additional motion component or this has been completed.
Value TRUE	Synchronous spindle coupling: The following spindle executes an additional motion component that is superimposed on the motion from the coupling with the leading spindle.
Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_SWL (DB31, ... DBX98.5)

Description	Velocity alarm threshold reached The signal is only relevant when the "Electronic gearbox (EG)" coupling function is active. It is set if the velocity of the following axis, reaches or exceeds the velocity alarm threshold, defined with MD37550 and MD32000 (= percentage of the maximum axis velocity).
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Electronic gearbox (EG): The velocity of the following axis is below the parameterized velocity alarm threshold.
Value TRUE	Electronic gearbox (EG): The velocity of the following axis has reached or exceeded the parameterized velocity alarm threshold.

Corresponds to	• MD37550 $MA_EG_VEL_WARNING Threshold value velocity alarm threshold • MD32000 $MA_MAX_AX_VELO Maximum axis velocity
Additional references	• Function Manual Axes and Spindles; Section "Axis couplings"

E_AWL (DB31, ... DBX98.6)

Description	Acceleration alarm threshold reached The signal is only relevant when the "Electronic gearbox (EG)" coupling function is active. It is set if the acceleration of the following axis, reaches or exceeds the acceleration alarm threshold, defined with MD37550 and MD32300 (= percentage of the maximum axis acceleration).
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Electronic gearbox (EG): The acceleration of the following axis is below the parameterized acceleration alarm threshold.
Value TRUE	Electronic gearbox (EG): The acceleration of the following axis has reached or exceeded the parameterized acceleration alarm threshold.
Corresponds to	• MD37550 $MA_EG_VEL_WARNING Threshold value velocity alarm threshold • MD32300 $MA_MAX_AX_ACCEL Maximum axis acceleration
Additional references	• Function Manual Axes and Spindles; Section "Axis couplings"

E_EmRetr (DB31, ... DBX98.7)

Description	ESR response initiated
Signal flow	NC → PLC
Data type	BOOL

E_MasterSp (DB31, ... DBX99.0)

Description	Leading spindle active **Note** The signal is relevant only in synchronous mode. **Special cases or errors** If faults occur on the following spindle that cause a withdrawal of the "Controller enable" for the following spindle, under certain circumstances, the control may exchange the coupling relationship of the following spindle and leading spindle and switch to actual value coupling. In this case, the previous leading spindle becomes the new active following spindle (IS "Following spindle active ").
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: The machine axis is presently not active as leading spindle.
Value TRUE	Synchronous spindle coupling: The machine axis is presently active as leading spindle.

Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode • "LBP_Axis*.E_SlaveSp" (DB31, ... DBX99.1) Following spindle active
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_SlaveSp (DB31, ... DBX99.1)

Description	Following spindle active Bei "LBP_Axis*.E_SlaveSp" (DB31, ... DBX99.1) == TRUE the following spindle follows the motion of the leading spindle corresponding to the ratio. **Note** The signal is relevant only in synchronous mode. **Special cases or errors** If faults occur on the following spindle that cause a withdrawal of the "Controller enable" for the following spindle, under certain circumstances, the control may exchange the coupling relationship of the following spindle and leading spindle and switch to actual value coupling.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Synchronous spindle coupling: The machine axis is currently not being operated as a following spindle.
Value TRUE	Synchronous spindle coupling: The machine axis is currently being operated as a following spindle.
Corresponds to	• "LBP_Axis*.E_PSyncMode" (DB31, ... DBX84.4) Active spindle mode: Synchronous mode • "LBP_Axis*.E_MasterSp" (DB31, ... DBX99.0) Leading spindle active
Additional references	• Function Manual Axes and Spindles; Section "Synchronous spindle"

E_SyncRunCorrDOut (DB31, ... DBX99.2)

Description	Synchronism correction implemented
Signal flow	NC → PLC
Data type	BOOL

E_AxisAccel (DB31, ... DBX99.3)

Description	Axis accelerating The signal is only relevant when the "Electronic gearbox (EG)" coupling function is active. It is set if the acceleration of the following axis, reaches or exceeds the acceleration alarm value, defined with MD37560 and MD32300.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Electronic gearbox (EG): The acceleration of the following axis lies below the parameterized acceleration level, at which the acceleration of the axis is displayed.
Value TRUE	Electronic gearbox (EG): The acceleration of the following axis has reached or exceeded the parameterized acceleration level, at which the acceleration of the axis is displayed.

Corresponds to	• MD37560 $MA_EG_ACC_TOL Threshold value for "Axis accelerates" • MD32300 $MA_MAX_AX_ACCEL Maximum axis acceleration
Additional references	• Function Manual Axes and Spindles; Section "Axis couplings"

E_Sync (DB31, ... DBX99.4)

Description	Synchronization running
Signal flow	NC → PLC
Data type	BOOL

E_MSR (DB31, ... DBX99.5)

Description	Maximum velocity reached
Signal flow	NC → PLC
Data type	BOOL

E_MAR (DB31, ... DBX99.6)

Description	Maximum acceleration reached
Signal flow	NC → PLC
Data type	BOOL

E_OscillAxExtRev (DB31, ... DBX100.2)

Description	External oscillation reversal active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The braking phase after "oscillation reversal from external" is not active.
Value TRUE	The braking phase after "oscillation reversal from external" is active.
Corresponds to	• "LBP_Axis*.A_OscillAxExtRev" (DB31, ... DBX28.0) Oscillation reversal from external
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_StopOscill (DB31, ... DBX100.3)

Description	Oscillation cannot be started This signal is set if there is a programming error, for example. **Note** This state can also occur if the axis has already been traversed.
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Oscillation motion can be started.
Value TRUE	Oscillating axis cannot be started.
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_ErrorOscill (DB31, ... DBX100.4)

Description	Fault during oscillation motion
Signal flow	NC → PLC
Data type	BOOL
Value FALSE	Oscillation is error free.
Value TRUE	Oscillation motion was canceled as an error occurred while oscillating.
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_SparkOut (DB31, ... DBX100.5)

Description	Sparking-out active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is presently not executing any sparking-out strokes.
Value TRUE	The axis executes sparking-out strokes.
Corresponds to	• "LBP_Axis*.E_Oscill" (DB31, ... DBX100.7) Oscillation active
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_OscillMotion (DB31, ... DBX100.6)

Description	Oscillation active The signal is irrelevant: "LBP_Axis*.E_Oscill" (DB31, ... DBX100.7) == FALSE
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is not currently oscillating.
Value TRUE	The axis performs an oscillation movement between two reversal points.
Corresponds to	• "LBP_Axis*.E_Oscill" (DB31, ... DBX100.7) Oscillation active
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_Oscill (DB31, ... DBX100.7)

Description	Oscillation active
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The axis is a positioning axis.
Value TRUE	The axis is presently being traversed as an oscillating axis.

Corresponds to	• "LBP_Axis*.E_SparkOut" (DB31, ... DBX100.5) Sparking-out active • "LBP_Axis*.E_OscillMotion" (DB31, ... DBX100.6) Oscillation movement active
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_GantCOLimit (DB31, ... DBX101.2)

Description	Gantry shutdown exceeded
Signal flow	NC → PLC
Data type	BOOL

E_GantWarn (DB31, ... DBX101.3)

Description	Gantry warning limit exceeded
Signal flow	NC → PLC
Data type	BOOL

E_SynRunStart (DB31, ... DBX101.4)

Description	Gantry synchronization ready to start
Signal flow	NC → PLC
Data type	BOOL

E_GantGroupSyn (DB31, ... DBX101.5)

Description	Gantry grouping is synchronous
Signal flow	NC → PLC
Data type	BOOL

E_GantLeadAxis (DB31, ... DBX101.6)

Description	Gantry guide axis
Signal flow	NC → PLC
Data type	BOOL

E_GantAxis (DB31, ... DBX101.7)

Description	Gantry axis
Signal flow	NC → PLC
Data type	BOOL

E_DynBComp (DB31, ... DBX102.0)

Description	Dynamic backlash compensation active
Signal flow	NC → PLC
Data type	BOOL

E_ClampTolerance (DB31, ... DBX102.3)

Description	Clamping tolerance exceeded
Signal flow	NC → PLC
Data type	BOOL

E_PosMeas1, E_PosMeas2

(DB31, ... DBX102.5, 102.6)

Description	Position measuring system 1 / 2 on
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Position measuring system 1 is in "Park" state. Monitoring and updating the position measuring system is deactivated.
Value TRUE	Position measuring system 1 is in the "active" or "passive" state. Monitoring and updating the position measuring system is activated.
Corresponds to	• "LBP_Axis*.A_PosMeas1" (DB31, ... DBX1.5) Position measuring system 1 • "LBP_Axis*.A_PosMeas2" (DB31, ... DBX1.6) Position measuring system 2
Additional references	Function Manual Monitoring and Compensating: • Section "Axis monitoring functions" > "Parking a machine axis" • Section "Axis monitoring functions" > "Parking the passive position measuring system"

E_SyncRunCorrIncl (DB31, ... DBX103.0)

Description	Synchronism correction is taken into account
Signal flow	NC → PLC
Data type	BOOL

E_Syncron2Fine (DB31, ... DBX103.4)

Description	Synchronism 2 fine
Signal flow	NC → PLC
Data type	BOOL

E_Syncron2Coarse (DB31, ... DBX103.5)

Description	Synchronism 2 coarse
Signal flow	NC → PLC
Data type	BOOL

E_AIAxis (DB31, ... from DBB104)

Description	Active infeed axes Axis 1: "LBP_Axis*.E_AIAxis[1]" (DB31, ... DBX104.0) Axis 2: "LBP_Axis*.E_AIAxis[2]" (DB31, ... DBX104.1) ... Axis 9: "LBP_Axis*.E_AIAxis[9]" (DB31, ... DBX105.0) ... Axis 31: "LBP_Axis*.E_AIAxis[31]" (DB31, ... DBX107.6)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	For the actual axis (oscillating axis), the corresponding axis is not an infeed axis.
Value TRUE	For the actual axis (oscillating axis), the corresponding axis is an infeed axis.
Corresponds to	• "LBP_Axis*.E_Oscill" (DB31, ... DBX100.7) Oscillation active
Additional references	• Function Manual Axes and Spindles; Section "Oscillation"

E_MMC_ProgtestSuppress (DB31, ... DBX128.0)

Description	Suppress program test If function "Program test" (PRT) is active in the channel to which the axis currently belongs, the axis is traversed with internal axis disable. Setpoints are generated but they are not output to the machine axes. Actual value = Setpoint. If the program test is suppressed for the axis, setpoints are output to the machine axis despite the function "program test" being active in the channel to which the axis currently belongs. **Automatic transfer of the interface signals** The HMI request signals "LBP_Axis*.E_MMC_ProgtestSuppress" and "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.0, 128.1) are only transferred from the PLC basic program to the PLC request signals "LBP_Axis*.A_ProgtestSuppress", "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.0, 14.1) if "LBP_ConfigBP [FC1]"parameter MMCToIF is set to TRUE. If the parameters are not set, the PLC request signals must be set by the PLC user program.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Suppression of the program test (setpoint output locked) is not requested by the HMI.
Value TRUE	Suppression of the program test (setpoint output locked) is requested by the HMI.
Corresponds to	• "LBP_Axis*.A_ProgtestSuppress" (DB31, ... DBX14.0) Suppress program test • "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.1) Activate program test • "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.1) Activate program test
Additional references	• Function Manual Basic Functions; Section "Cross-channel program coordination and channel-by-channel running-in" > "Channel-by-channel running-in"

E_MMC_ProgtestActivate (DB31, ... DBX128.1)

Description	Activate program test If the function "program test" is active, the axis is traversed with internal axis disable. Setpoints are generated but they are not output to the machine axes. Actual value = Setpoint. **Automatic transfer of the interface signals** The HMI request signals "LBP_Axis*.E_MMC_ProgtestSuppress" and "LBP_Axis*.E_MMC_ProgtestActivate" (DB31, ... DBX128.0, 128.1) are only transferred from the PLC basic program to the PLC request signals "LBP_Axis*.A_ProgtestSuppress", "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.0, 14.1) if "LBP_ConfigBP [FC1]"parameter MMCToIF is set to TRUE. If the parameters are not set, the PLC request signals must be set by the PLC user program.
Signal flow	HMI → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Activation of the function "program test" (setpoint output locked) is not requested by the HMI.
Value TRUE	Activation of the function "program test" (setpoint output locked) is requested by the HMI.
Corresponds to	• "LBP_Axis*.A_ProgtestSuppress" (DB31, ... DBX14.0) Suppress program test • "LBP_Axis*.A_ProgtestActivate" (DB31, ... DBX14.1) Activate program test • "LBP_Axis*.E_MMC_ProgtestSuppress" (DB31, ... DBX128.0) Suppress program test
Additional references	• Function Manual Basic Functions; Section "Cross-channel program coordination and channel-by-channel running-in" > "Channel-by-channel running-in"

E_SensorAv (DB31, ... DBX132.0)

Description	Sensors available **Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The sensor required for spindles with SMI 24 is not available.
Value TRUE	The sensor required for spindles with SMI 24 is available.
Corresponds to	• "LBP_Axis*.E_S1aMeasARod" (DB31, ... DBX132.1) Sensor S1 (clamped state) is available • "LBP_Axis*.E_S4dPistonStopP" (DB31, ... DBX132.4) Sensor S4 is available • "LBP_Axis*.E_S5dAngleShaft" (DB31, ... DBX132.5) Sensor S5 is available
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_S1aMeasARod (DB31, ... DBX132.1)

Description	Sensor S1 available (clamped state) **Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Sensor S1 is not available.
Value TRUE	Sensor S1 is available.
Corresponds to	• "LBP_Axis*.E_ClampingState" (DB31, ... DBW134) Status of the clamping system • "LBP_Axis*.E_AnalogSensor" (DB31, ... DBW136) Analog value: Clamped state
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_S4dPistonStopP (DB31, ... DBX132.4)

Description	Sensor S4 available (piston end position) **Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Sensor S4 is not available.
Value TRUE	Sensor S4 is available.
Corresponds to	• "LBP_Axis*.E_S4PistonStopP" (DB31, ... DBX138.4) sensor S4: Piston end position
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_S5dAngleShaft (DB31, ... DBX132.5)

Description	Sensor S5 available (angular position of the motor shaft) **Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	Sensor S5 is not available.
Value TRUE	Sensor S5 is available.
Corresponds to	• "LBP_Axis*.E_S5AngleShaft" (DB31, ... DBX138.5) Sensor S5: Angular position of the motor shaft
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_S6Temp (DB31, ... DBX132.6)

Description	Sensor S6 available
Signal flow	NC → PLC
Data type	BOOL

E_ClampingState (DB31, ... DBW134)

<table>
<tr><td>Description</td><td colspan="2">State of the clamping system (sensor S1)
Note
Only relevant for spindles with SMI 24 (Weiss spindle)

Depending on the position of the clamping device, sensor S1 supplies an analog voltage value. To simplify the evaluation of the clamped state, the analog voltage of sensor module SMI 24 is converted into a state value.
The state values correspond to certain voltage ranges. The voltage ranges can be set via: Drive parameters p5041[0...5].</td></tr>
<tr><td></td><td>State value</td><td>Clamped state</td></tr>
<tr><td></td><td>0</td><td>Sensor S1 not available or state values inactive</td></tr>
<tr><td></td><td>1</td><td>State initialization running</td></tr>
<tr><td></td><td>2</td><td>Released with signal (error state)</td></tr>
<tr><td></td><td>3</td><td>Released</td></tr>
<tr><td></td><td>4</td><td>Clamping with tool</td></tr>
<tr><td></td><td>5</td><td>Releasing with tool</td></tr>
<tr><td></td><td>6</td><td>Releasing without tool</td></tr>
<tr><td></td><td>7</td><td>Clamped with tool AND S4 == FALSE</td></tr>
<tr><td></td><td>8</td><td>Clamped with tool AND S4 == TRUE</td></tr>
<tr><td></td><td>9</td><td>Clamping without tool</td></tr>
<tr><td></td><td>10</td><td>Clamped without tool</td></tr>
<tr><td>Signal flow</td><td colspan="2">NC → PLC</td></tr>
<tr><td>Data type</td><td colspan="2">BOOL</td></tr>
<tr><td>Update</td><td colspan="2">Cyclic</td></tr>
<tr><td>Corresponds to</td><td colspan="2">• "LBP_Axis*.E_AnalogSensor" (DB31, ... DBW136)
Analog value: Clamped state</td></tr>
<tr><td>Additional references</td><td colspan="2">• Function Manual Axes and Spindles; Section "Spindles"</td></tr>
</table>

E_AnalogSensor (DB31, ... DBW136)

Description	Analog measured value of the clamping system **Note** Only relevant for spindles with SMI 24 (Weiss spindle) Sensor S1 supplies an analog voltage value: 0 ... 10 V. The analog value of the clamped state is mapped to: 0 ... 10000 increments, resolution 1 mV **Note** SIMATIC S7 input module: 0 ... 27648 increments, resolution 0.36 mV Adaptation factor if you change to a spindle with SMI 24: 2.7648
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Corresponds to	• "LBP_Axis*.E_ClampingState" (DB31, ... DBW134) Clamping state • Drive parameters: p5041[0...5], p5043[0...6]
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_S4PistonStopP (DB31, ... DBX138.4)

Description	Sensor S4, piston end position **Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The piston is not in position
Value TRUE	The piston is in position, i.e. the piston is free to move
Corresponds to	• "LBP_Axis*.E_S4dPistonStopP" (DB31, ... DBX132.4) Sensor S4 available
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

E_S5AngleShaft (DB31, ... DBX138.5)

Description	Sensor S5, angular position of the motor shaft **Note** Only relevant for spindles with SMI 24 (Weiss spindle)
Signal flow	NC → PLC
Data type	BOOL
Update	Cyclic
Value FALSE	The motor shaft is not aligned
Value TRUE	The motor shaft is in position (requirement: The spindle is stationary)
Corresponds to	• "LBP_Axis*.E_S5dAngleShaft" (DB31, ... DBX132.5) Sensor S5 available
Additional references	• Function Manual Axes and Spindles; Section "Spindles"

SCC (DB31, ... DBD140)

Description	Signals from PLC to drive
Data type	STRUCT STRUCT "SCC" PLC to drive (Page 511)

SIC (DB31, ... DBB164 ... 174)

Description	Signals from the drive to the PLC
Data type	STRUCT STRUCT "SIC" drive to PLC (Page 512)

E_EncMaintenanceRequired (DB31, ... DBX190.0)

Description	Encoder servicing required
Signal flow	NC → PLC
Data type	BOOL

E_TorqueLimited (DB31, ... DBX191.0)

Description	DYNEGMA: Torque limit applied
Signal flow	NC → PLC
Data type	BOOL

E_CurrReductionActive (DB31, ... DBX191.1)

Description	Motor overtemperature: Current reduction active
Signal flow	NC → PLC
Data type	BOOL

E_MotorTempModelAlarm (DB31, ... DBX191.2)

Description	Motor overtemperature: Motor temperature model warning active
Signal flow	NC → PLC
Data type	BOOL

E_PolePosIdentPassed (DB31, ... DBX191.3)

Description	Pole position identification with encoder successfully carried out
Signal flow	NC → PLC
Data type	BOOL

E_AndConditionFulfilled (DB31, ... DBX191.6)

Description	AND operation: Condition fulfilled
Signal flow	NC → PLC
Data type	BOOL

E_OrConditionFulfilled (DB31, ... DBX191.7)

Description	OR operation: Condition fulfilled
Signal flow	NC → PLC
Data type	BOOL

10.4.10.4 STRUCT "SCC" PLC to drive

The variables structure "SCC" describes signals from the PLC to the drive.

Variables in "SCC"

CW1.TeststopStart (DB31, ... DBX140.0)

Description	Test stops for extended functions
Data type	BOOL

CW1.CloseBrake (DB31, ...DBX140.5)

Description	Closing an internal brake
Data type	BOOL

CW3.SelectionSBT (DB31, ... DBX143.0)

Description	Selection: safe brake test
Data type	BOOL

CW3.BT_Start (DB31, ... DBX143.1)

Description	Start: Brake test
Data type	BOOL

CW3.Brake_1_2 (DB31, ... DBX143.2)

Description	Brake test with brake 1 or 2
Data type	BOOL

CW3.DirOfROT (DB31, ... DBX143.3)

Description	Direction of rotation
Data type	BOOL

CW3.Testseq_1_2 (DB31, ... DBX143.4)

Description	Test sequence 1 or 2
Data type	BOOL

CW3.BrakeClosed (DB31, ... DBX143.5)

Description	Feedback signal external brake
Data type	BOOL

10.4.10.5 STRUCT "SIC" drive to PLC

The variables structure "SIC" describes the signals from the drive to the PLC.

Variables in "SIC"

SW1.SLA_Sel (DB31, ... DBX164.0)

Description	Safely limited acceleration selected
Data type	BOOL

SW1.SLS_LimBit0_Sel (DB31, ... DBX164.1)

Description	Safely limited speed: Limit Bit0 selected
Data type	BOOL

SW1.SLS_LimBit1_Sel (DB31, ... DBX164.2)

Description	Safely limited speed: Limit Bit1 selected
Data type	BOOL

SW1.ESR_Req (DB31, ... DBX164.6)

Description	Extended stop and retraction requested
Data type	BOOL

SW1.SF_Message_Act (DB31, ... DBX164.7)

Description	Safety error active
Data type	BOOL

SW1.STO_Act (DB31, ... DBX165.0)

Description	Safe torque off: Active
Data type	BOOL

SW1.SS1_Act (DB31, ... DBX165.1)

Description	Safe Stop1: Active
Data type	BOOL

SW1.SS2_Act (DB31, ... DBX165.2)

Description	Safe Stop2: Active
Data type	BOOL

SW1.SOS_Act (DB31, ... DBX165.3)

Description	Safe operating stop: Active
Data type	BOOL

SW1.SLS_Act (DB31, ... DBX165.4)

Description	Safely limited speed: Active
Data type	BOOL

SW1.SOS_Sel (DB31, ... DBX165.5)

Description	Safe operating stop: selected
Data type	BOOL

SW1.SLS_Sel (DB31, ... DBX165.6)

Description	Safely limited speed: selected
Data type	BOOL

SW1.InternalEvent (DB31, ... DBX165.7)

Description	Internal event
Data type	BOOL

SW2.SDI_P_Sel (DB31, ... DBX166.0)

Description	Safe direction positive: selected
Data type	BOOL

SW2.SDI_N_Sel (DB31, ... DBX166.1)

Description	Safe direction negative: selected
Data type	BOOL

SW2.TestStopAct (DB31, ... DBX166.4)

Description	Test stop active
Data type	BOOL

SW2.TestStopReq (DB31, ... DBX166.5)

Description	Test stop requested
Data type	BOOL

SW2.SLP_LimBit0_Sel (DB31, ... DBX167.4)

Description	Safely limited position: Limit Bit0 selected
Data type	BOOL

SW2.SLP_Sel (DB31, ... DBX167.7)

Description	Safely-Limited Position: selected
Data type	BOOL

V_Limit_B (DB31, ... DBD168)

Description	Set speed limit effective
Data type	DINT

SW3.SS2E_ACTIVE (DB31, ... DBX172.3)

Description	SS2E active
Data type	BOOL

SW3.SS2ESR (DB31, ... DBX172.4)

Description	SS2ESR active
Data type	BOOL

SW3.AcceptSLPAct (DB31, ... DBX172.6)

Description	Acceptance test safely limited position active
Data type	BOOL

SW3.AcceptTestSel (DB31, ... DBX172.7)

Description	Acceptance test selected
Data type	BOOL

SW3.SBT_Sel (DB31, ... DBX173.0)

Description	Emergency brake tests: selected
Data type	BOOL

SW3.SetpSettingDrive (DB31, ... DBX173.1)

Description	Setpoint input during SBT in the drive
Data type	BOOL

SW3.Brake_1_2 (DB31, ... DBX173.2)

Description	Test with brake 1 or 2
Data type	BOOL

SW3.BT_Act (DB31, ... DBX173.3)

Description	Brake test: Active
Data type	BOOL

SW3.BT_OK (DB31, ... DBX173.4)

Description	Brake test: OK
Data type	BOOL

SW3.BT_Finished (DB31, ... DBX173.5)

Description	Brake test: completed
Data type	BOOL

SW3.BrakeReq (DB31, ... DBX173.6)

Description	Close external brake requested
Data type	BOOL

SW3.SignLoadTorque (DB31, ... DBX173.7)

Description	Sign for load torque is negative
Data type	BOOL

10.4.11 LBP_TMLoad [DB71] - Interface for loading/unloading the magazine:

10.4.11.1 Overview

LBP_TMLoad [DB71], interface for loading/unloading the magazine

Table 10-43 Interface for loading/unloading magazine

LBP_TMLoad [DB71]	**Loading/unloading locations (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (IF) active							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB1								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBB2	Autom. acknowledgment							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB3								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBBn + 0	Reserved		Command:					
			Position a multitool	NC progr. positions magazine	Positioning to the loading location	Relocating	Unloading	Loading
DBBn + 1	Data in extended Area (LBP_TMLoadMT [DB1071] (Page 575))	Reserved						Acknowledgement status = 3
DBBn + 2	Assigned channel (8-bit INT)							
DBBn + 3	Tool management no. (8-bit INT)							
DBBn + 4	Reserved (free parameter 1 (DWord))							
DBBn + 8	Reserved (free parameter 2 (DWord))							
DBDn + 12	Reserved (free parameter 3 (DWord))							
DBWn + 16	Identifier for loading/unloading location (INT)							
DBWn + 18	Location no. of load/unload point (INT)							
DBWn + 20	Magazine no. (Source) for unloading/relocating/positioning (INT)							

LBP_TMLoad [DB71]	Loading/unloading locations (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 22	Location no. (Source) for unloading/relocating/positioning (INT)							
DBWn + 24	Magazine no. (Target) for loading/relocating/positioning (INT)							
DBWn + 26	Location no. (Target) for loading/relocating/positioning (INT)							
DBWn + 28	Reserved							Loading/ unloading without moving magazine
DBWn + 29	Reserved							

Starting addresses of loading/unloading locations:

Loading/unloading location 1:	After DBB4
Loading/unloading location 2:	After DBB34
Loading/unloading location 3:	After DBB64
Loading/unloading location 4:	After DBB94
...	...
Loading/unloading location 16:	After DBB454

Loading interface 1 is responsible for loading/unloading in (all) spindles/toolholders and for relocating tools and for positioning at any locations (e.g. buffer location).

Loading and unloading manual tools is always realized via loading interface 1.

10.4.11.2 LBP_TMLoad [DB71]

Variables in "LBP_TMLoad [DB71]"

The interface signals for loading/unloading the magazine are stored in "LBP_TMLoad [DB71]".

IFNo (DB71 from DBX0.0)

Description	Interface (IF) active If an interface is active, a valid data set exists for this interface. The index corresponds to the interface number (IF), e.g.: Index 1 corresponds to IF 1
Signal flow	NC → PLC
Data type	ARRAY[1..16] of BOOL
Update	Job-controlled

Value FALSE	The interface is **not** active.
Value TRUE	The interface is active.

Quit (DB71 from DBX2.0)

Description	"auto" acknowledgement of interfaces 1 ... 16
Signal flow	PLC → NC
Data type	ARRAY[1..16] of BOOL
Update	Job-controlled
Value: FALSE	Reset by the basic program when TRUE is detected.
Value: TRUE	When bit is set, the active command is acknowledged with Status:=1. Unless it is a relocation operation from a real magazine to a buffer. Then it is acknowledged with Status:=6. This reserves the source location.

TIF (DB71 from DBB4)

Description	Interface signals for loading/unloading the magazine Loading/unloading location 1: Index 1 Loading/unloading location 2: Index 2 ... Loading/unloading location 16: Index 16 Loading/unloading location 1 is responsible for loading/unloading in (all) spindles/tool holders and for relocating tools and for positioning at any locations (e.g. buffer location). Loading and unloading manual tools is always realized via loading/unloading location 1.
Signal flow	NC → PLC
Data type	ARRAY[1..16] of STRUCT STRUCT - "TIF" for loading/unloading the magazine: (Page 517)
Update	Job-controlled

10.4.11.3 STRUCT - "TIF" for loading/unloading the magazine:

The variable structure "TIF" describes interface signals for loading/unloading the magazine.

Variables in "TIF"

Loading (DB71 DBX4.0, 34.0, ..., 454.0)

Description	Command: Loading The bit is not updated by the basic program until a new job exists for this interface. It is current if the corresponding interface bit is TRUE in "LBP_TMLoad.IfNo[i]" (DB71 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled

Value TRUE	Loading operation for a tool is initiated. The magazine location into which the tool is to be loaded is defined in "TIF.TLoc". The loading location in question is the location number of the loading location. It is also located in "TIF.LoadNo"
Corresponds to	"TIF.LoadIdent" "TIF.LoadNo" "TIF.TLoc"

Unloading (DB71 DBX4.1, 34.1, ..., 454.1)

Description	Command: Unloading The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMLoad.IfNo[i]" (DB71 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	Unloading operation is initiated for a tool; the magazine location from which it is to be unloaded is in "TIF.SMag" and "TIF.SLoc". The number of the unloading location is in "TIF.LoadNo".
Edge evaluation	Signal(s) updated: Conditional
Corresponds to	"TIF.LoadIdent" "TIF.LoadNo" "TIF.SMag" "TIF.SLoc"

Reloading (DB71 DBX4.2, 34.2, ..., 454.2)

Description	Command: Relocating The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMLoad.IfNo[i]" (DB71 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	Relocate operation for a tool is initiated. From magazine/ location ("TIF.SMag", "TIF.SLoc"=source) to magazine/location ("TIF.TMag", "TIF.TLoc"=destination)

Positioning (DB71 DBX4.3, 34.3, ..., 454.3)

Description	Command: Positioning to the loading location The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMLoad.IfNo[i]" (DB71 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL

Update	Job-controlled
Value TRUE	A magazine location is to be positioned at the loading location (magazine no. 9999). The magazine location that is to be moved to the loading location is in "TIF.SMag" and "TIF.SLoc". The loading location is in "TIF.LoadNo".

PosByProgram (DB71 DBX4.4, 34.4, ..., 454.4)

Description	Command: Task comes from the NC program The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMLoad.IfNo[i]" (DB71 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The signal is set if the task is received from the part program or via a cycle. 1. Positioning task with the NC language command POSM 2. A relocation task or tool transport uses the NC language command MVTOOL 3. Positioning a multitool using the NC language command POSMT

PositioningMT (DB71 DBX4.5, 34.5, ..., 454.5)

Description	Command: Position a multitool The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMLoad.IfNo[i]" (DB71 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	Positioning a multitool has been initiated. This can be realized using an NC language command, PI service or the HMI user interface.

PerfStatusError (DB71 DBX5.0, 35.0, ..., 455.0)

Description	"auto" acknowledgement negative
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	The bit is only evaluated in conjunction with "auto" ackn. If it is not set, the auto acknowledge is positive.
Value TRUE	The bit is only evaluated in conjunction with "auto" acknowledge. If it is set, the auto acknowledge is negative, i.e. with Status:=3.

DataInDB1071 (DB71 DBX5.7, 35.7, ..., 455.7)

Description	Data in the extended range See also LBP_TMLoadMT [DB1071] - Loading/unloading magazine: Multitool (Page 575)
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The signal is set if a multitool is loaded, unloaded, or positioned. The multitool data are then available in "LBP_TMLoadMT [DB1071]".

Channel (DB71 DBB6, 36, ..., 456)

Description	Assigned channel
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Value	Number of the channel for which the active interface is valid.

TMNo (DB71 DBB7, 37, ..., 457)

Description	Tool management number (with TO unit)
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Value	Associated tool management number; corresponds to the number of the TO unit within a TO area.

LoadIdent (DB71 DBW20, 50, ..., 470)

Description	Identifier for loading/unloading location (fixed value 9999)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	If the multitool is loaded to or unloaded from a buffer, the magazine number of the buffer magazine is entered here.

LoadNo (DB71 DBW22, 52, ..., 472)

Description	Location no. of the loading/unloading location
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	The location no. of the loading/unloading location is displayed. If the multitool is loaded to or unloaded from a buffer, the location number of the buffer magazine is entered here.

SMag (DB71 DBW24, 54, ..., 474)

Description	Magazine no. (Source) for unloading/relocating/positioning
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No magazine is unloaded, relocated, or positioned.
Value > 0	Unloading: Magazine from which the tool is to be unloaded Relocating: Magazine from which the tool is taken Positioning: Magazine that is to be positioned
Corresponds to	"TIF.SLoc"

SLoc (DB71 DBW26, 56, ..., 476)

Description	Location no. (Source) for unloading/relocating/positioning
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No tool is unloaded, relocated, positioned from any location.
Value > 0	Unloading: Location from which the tool is to be unloaded Relocating: Location from which the tool is taken Positioning: Location to be positioned
Corresponds to	"TIF.SMag"

TMag (DB71 DBW28, 58, ..., 478)

Description	Magazine no. (Target) for loading/relocating/positioning
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No magazine number specified.
Value > 0	Loading: Magazine into which the tool is to be loaded Relocating: Magazine into which the tool is loaded Positioning: Magazine to be positioned to - tool remains in the original magazine Only meaningful for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (NC language command POSM).
Corresponds to	"TIF.TLoc"

TLoc (DB71 DBW30, 60, ..., 480)

Description	Location no. (Target) for loading/relocating/positioning
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No location number specified for loading, relocating, positioning.

Value > 0	Loading: Location into which the tool is to be loaded Relocating: Location into which the tool comes Positioning: Location at which the tool must be positioned. Tool remains at original location Only meaningful for interface 1. If values other than 0 are entered here, the data define the magazine or location for positioning (NC language command POSM).
Corresponds to	"TIF.TMag"

WithoutMagMoving (DB71 DBX32.0, 62.0, ..., 482.0)

Description	Loading/unloading without moving magazine
Signal flow	HMI → PLC
Data type	BOOL
Update	Job-controlled
Value FALSE	HMI / Jobshop delete this signal when requested by the operator.
Value TRUE	If the bit is active, there must be no traversing motion of the magazine, only mechanical unlocking/locking of the location. The loading/unloading command must be acknowledged after the action. For a positioning and relocating request, this signal is not valid for a traversing movement.

10.4.12 LBP_TMSpindle [DB72] - Spindle as change position

10.4.12.1 Overview

LBP_TMSpindle [DB72], interface for the spindle as change position

Table 10-44 Interface for spindle as change position

LBP_TMSpindle [DB72]	**Signals from the spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (IF) active							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB1								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBB2	Standard end acknowledgement							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB3								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9

LBP_TMSpindle [DB72]	**Signals from the spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 0	Command code:							
	Spindle tool remains in the spindle	Replace manual tool	Insert manual tool	Old T in buffer	T0	Prepare change	Change tool (initiated by: M06)	Obligatory change
DBBn + 1	Data in extended Area (LBP_TMSpindleMT [DB1072] (Page 577))	Reserved						Acknowledgement status = 3
DBBn + 2	Assigned channel (8-bit INT)							
DBBn + 3	Tool management No. (8-bit INT)							
DBDn + 4	User parameter 0 (DWord)							
DBDn + 8	User parameter 1 (DWord)							
DBDn + 12	User parameter 2 (DWord)							
DBWn + 16	Buffer identifier (INT)							
DBWn + 18	Relative location (target) in buffer magazine (INT)							
DBWn + 20	Magazine no. (Source) for new tool (INT)							
DBWn + 22	Location no. (Source) for new tool (INT)							
DBWn + 24	Magazine no. (Target) for old tool (INT)							
DBWn + 26	Location no. (Target) for old tool (INT)							
DBWn + 28	Tool new: Location type (INT)							
DBWn + 30	Tool new: Size left (INT)							
DBWn + 32	Tool new: Size right (INT)							
DBWn + 34	Tool new: Size top (INT)							
DBWn + 36	Tool new: Size bottom (INT)							
DBBn + 38	Tool status for new tool							
	Manual tool	1:1 exchange	Reserved	Master tool	Tool to be loaded	Tool to be unloaded	Disabled, but ignore	Tool in the buffer magazine
DBBn + 39	Tool status for new tool							
	Tool was in use	Tool fixed-location-coded	Tool being changed	Prewarning limit reached	Measure tool	Tool disabled	Enable tool	Active tool
DBWn + 40	Tool new: Internal T No. (INT)							
DBWn + 42	Buffer location of the old tool							
DBWn + 44	Original magazine of the new tool							

Start addresses of the buffer

Spindle 1:	After DBB4
Spindle 2:	After DBB52
Spindle 3:	After DBB100
...	...
Spindle 16:	After DBB724

10.4.12.2 LBP_TMSpindle [DB72]

In "LBP_TMSpindle [DB72]", the interface signals for a spindle are stored as the change position.

Variables in "LBP_TMSpindle [DB72]"

IFNo (DB72.DBW0)

Description	Interface (IF) active If an interface is active, a valid data set exists for this interface. The index corresponds to the interface number (IF), e.g.: Index 1 corresponds to IF 1
Signal flow	NC → PLC
Data type	ARRAY[1..16] of BOOL
Update	Job-controlled
Value FALSE	The interface is **not** active.
Value TRUE	The interface is active.

Quit (DB72.DBW2)

Description	"auto" acknowledgement of interfaces 1 - 16
Signal flow	PLC → NC
Data type	ARRAY[1..16] of BOOL
Update	Job-controlled
Value: FALSE	The basic program is reset when TRUE is detected.
Value: TRUE	When bit is set (edge), the active command is acknowledged with Status:=1. Unless it is a relocation operation from a real magazine to a buffer. Then it is acknowledged with Status:=6. This reserves the source location.

TIF (DB72 from DBB4)

Description	Interface signals for tool change on spindles: Spindle 1: Index 1 Spindle 2: Index 2 ... Spindle 16: Index 16
Signal flow	NC → PLC

Data type	ARRAY[1..16] of STRUCT STRUCT "TIF" for spindle as change position (Page 525)
Update	Job-controlled

10.4.12.3 STRUCT "TIF" for spindle as change position

The variable structure "TIF" describes interface signals for the tool change if the spindle is in the change position.

Variables in "TIF"

Comp (DB72 DBX4.0, 52.0, ..., 724.0)

Description	Command: Obligatory change The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The new tool is fixed-location-coded.
Corresponds to	"TIF.Perform" "TIF.Prepare"

Perform (DB72 DBX4.1, 52.1, ..., 724.1)

Description	Command: Perform change (M06 programmed) The bit in "TIF.Prepare" (prepare change) is not reset by the system with a change command. The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary. If "TIF.Perform" and "TIF.Prepare" are present at the same time, it means that T and M06 were programmed in one block.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The command "M06" for tool change was programmed and the tool change can be performed.
Corresponds to	"TIF.Perform" "TIF.Prepare"

Prepare (DB72 DBX4.2, 52.2, ..., 724.2)

Description	Command: Prepare a tool change The bit in "TIF.Prepare" (prepare change) is not reset by the system with a change command. The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary. If "TIF.Perform" and "TIF.Prepare" are present at the same time, it means that T and M06 were programmed in one block.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	Prepare a new tool for changing. If necessary, move the magazine location for the old tool to the spindle.

T0 (DB72 DBX4.3, 52.3, ..., 724.3)

Description	Command: T0 (run spindle without tool) The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	T0 (run spindle without tool) has been programmed.

OldTinDBW42 (DB72 DBX4.4, 52.4, ..., 724.4)

Description	Command: Old tool in buffer The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The number of the buffer of the tool to be changed is written in "TIF.OldTBufferNo".

ManTIn (DB72 DBX4.5, 52.5, ..., 724.5)

Description	Command: Load manual tool The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL

Edge evaluation	Signal(s) updated: Conditional
Value TRUE	Manual tool is loaded. The manual tool to be loaded is displayed on the HMI.

ManTOut (DB72 DBX4.6, 52.6, ..., 724.6)

Description	Command: Unload manual tool The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Edge evaluation	Signal(s) updated: Conditional
Value TRUE	The tool is changed by manual operation.

SpiTRSpi (DB72 DBX4.7, 52.7, ..., 724.7)

Description	Command: Tool remains in spindle The bit is not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMSpindle.IfNo[i]" (DB72 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Edge evaluation	Signal(s) updated: Conditional
Value TRUE	The bit is set at change from spindle to spindle. The change is initiated, e.g. by reset and start mode or block search.

PerfStatusError (DB72 DBX5.0, 53.0, ..., 725.0)

Description	"auto" acknowledgement negative
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value FALSE	The bit is only evaluated in conjunction with "auto" ackn. If it is not set, the auto acknowledge is positive.
Value TRUE	The bit is only evaluated in conjunction with "auto" acknowledge. If it is set, the auto acknowledge is negative, i.e. with Status:=3.

DataInDB1072 (DB72 DBX5.7, 53.7, ..., 725.7)

Description	Data in the extended range See also
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The signal is set if a multitool is loaded. The multitool data are then available in "LBP_TMSpindleMT [DB1072]".

Channel (DB72 DBB6, 54, ..., 726)

Description	Assigned channel
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Value	Number of the channel for which the active interface is valid.

TMNo (DB72 DBB7, 55, ..., 727)

Description	Tool management number
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Value	Associated tool management number; corresponds to the TO unit

VdiTcp0 (DB72 DBD8, 56, ..., 728)

VdiTcp1 (DB72 DBD12, 60, ..., 732)

VdiTcp2 (DB72 DBD16, 64, ..., 736)

Description	User-definable parameters • Parameter 0 (VdiTcp0) • Parameter 1 (VdiTcp1) • Parameter 2 (VdiTcp2)
Signal flow	NC → PLC
Data type	DINT
Update	Job-controlled
Value	A value can be passed to the PLC via the part program by programming: • $P_VDITCP[0]; for free parameter 0 • $P_VDITCP[1]; for free parameter 1 • $P_VDITCP[2]; for free parameter 2 Values are transferred with a T call.

BufferIdent (DB72 DBW20, 68, ..., 740)

Description	Buffer magazine No.; target position for new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value >0	Magazine number of the buffer magazine; target magazine for new tool

BufferNo (DB72 DBW22, 70, ..., 742)

Description	Location in buffer magazine (spindle)
Signal flow	NC → PLC
Data type	INT

Update	Job-controlled
Value	Location no. of buffer magazine to which the new tool must be loaded. This is normally the spindle. The location no. defined for this particular buffer during commissioning is output.

SMag (DB72 DBW24, 72, ..., 744)

Description	Magazine no. (source) for new tool to be loaded
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No magazine no. is output.
Value > 0	Magazine no. of the magazine from which the new spindle tool comes.
Corresponds to	"TIF.SLoc"

SLoc (DB72 DBW26, 74, ..., 746)

Description	Location no. (source) for new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No location no. is output.
Value > 0	Location no. of the magazine from which the new spindle tool comes.
Corresponds to	"TIF.SMag"

TMag (DB72 DBW28, 76, ..., 748)

Description	Magazine no. (target) for old tool to be removed
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No magazine number specified.
Value > 0	Number of the magazine in which the removed tool will be placed.
Corresponds to	"TIF.TLoc"

TLoc (DB72 DBW30, 78, ..., 750)

Description	Location no. (target) for old tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No location number specified.
Value > 0	Magazine location for tool that is unloaded at change.
Corresponds to	"TIF.TMag"

LocType (DB72 DBW32, 80, ..., 752)

Description	Location type of the new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	The location type of the new spindle tool is entered here.
Corresponds to	"TIF.SLeft" "TIF.SRight" "TIF.STop" "TIF.Bottom"

SLeft (DB72 DBW34, 82, ..., 754)

Description	Tool new: Size left
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **left** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SRight" "TIF.STop" "TIF.Bottom"

SRight (DB72 DBW36, 84, ..., 756)

Description	Tool new: Size right
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **right** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SLeft" "TIF.STop" "TIF.Bottom"

STop (DB72 DBW38, 86, ..., 758)

Description	Tool new: Size top
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **top** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SLeft" "TIF.SRight" "TIF.Bottom"

SBottom (DB72 DBW40, 88, ..., 760)

Description	Tool new: Size bottom
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **bottom** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SLeft" "TIF.SRight" "TIF.Top"

TInBuffer, IgnoreDisabled, Unload, Load, MasterTool, Exchange_1to1, HandTool, Active, Enabled, Measured, Prewarn, Change, FixedLoc, Used

(DB72 from DBX42.0, 90.0, ..., 762.0)

Description	Tool status of the new tool	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Job-controlled	
Status bits	TInBuffer (DBX42.0, 90.0, ..., 762.0)	Tool in the buffer magazine
	IgnoreDisabled	Disabled, but ignore
	Unload	Tool to be unloaded
	Load	Tool to be loaded
	MasterTool	Master tool
	Exchange_1to1	1:1 exchange
	HandTool	Manual tool
	Active	Active tool
	Enabled	Enable tool
	Disabled	Tool disabled
	Measured	Measure tool
	Prewarning	Prewarning limit reached
	Change	Tool is being changed
	FixedLoc	Tool is fixed-location-coded
	Used (DBX43.7, 91.7, ..., 763.7)	Tool was in use

TNo (DB72 DBW44, 92, ..., 764)

Description	Tool new: Internal T number of the NC
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Display of internal T number of NC for the new spindle tool.

OldTBufferNo (DB72 DBW46, 94, ..., 766)

Description	Buffer location of old tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	If "TIF.IFNo[5]=TRUE", the buffer location of the old tool must be entered here. This can be any buffer (also a gripper).

MyMag (DB72 DBW48, 96, ..., 768)

Description	Owner magazine of the new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Owner magazine of the new tool Corresponds to NC variables $A_MYMN[T no.] If the new tool is located in the magazine, then this value is identical with "TIF.BufferIdent". If the new tool is in the buffer magazine (e.g. gripper), the magazine no. of the magazine from which the tool originally came is entered here.

MyPlace (DB72 DBW50, 98, ..., 770)

Description	Original location of new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Owner location of the new tool Corresponds to NC variables $A_MYMLN[T no.]

10.4.13 LBP_TMTurret [DB73] - Turret

10.4.13.1 Overview

LBP_TMTurret [DB73], interface for the turret

Table 10-45 Interface for the turret

LBP_TMTurret [DB73]	**Signals for the turret (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0	Interface (IF) active							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1

LBP_TMTurret [DB73]	**Signals for the turret (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB1								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBB2	Standard end acknowledgement							
	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1
DBB3								
	SS16	SS15	SS14	SS13	SS12	SS11	SS10	SS9
DBBn	Reserved	Command code:						
		Unload manual tool	Reserved		T0	Reserved	Perform change	Obligatory change
DBBn + 1	Data in extended Area (LBP_TMTurretMT [DB1073] (Page 583))	Reserved						Acknowledgement status = 3
DBBn + 2	Assigned channel (8-bit INT)							
DBBn + 3	Tool management No. (8-bit INT)							
DBDn + 4	User parameter 1 (DWord)							
DBDn + 8	User parameter 2 (DWord)							
DBDn + 12	User parameter 3 (DWord)							
DBWn + 16	Buffer identifier (INT) (corresponds to "Target position for new tool")							
DBWn + 18	Relative location (target) in buffer magazine (INT)							
DBWn + 20	Magazine No. of the turret (INT)							
DBWn + 22	Location No. of new tool (INT)							
DBWn + 24	Magazine No. of old tool							
DBWn + 26	Location No. of old tool (INT)							
DBWn + 28	Tool new: Location type (INT)							
DBWn + 30	Tool new: Size left (INT)							
DBWn + 32	Tool new: Size right (INT)							
DBWn + 34	Tool new: Size top (INT)							
DBWn + 36	Tool new: Size bottom (INT)							
DBBn + 38	Tool status for new tool							
	Manual tool	1:1 exchange	Reserved	Master tool	Tool to be loaded	Tool to be unloaded	Disabled, but ignore	Tool in the buffer magazine
DBBn + 39	Tool status for new tool							
	Tool was in use	Tool fixed-location-coded	Tool being changed	Prewarning limit reached	Measure tool	Tool disabled	Enable tool	Active tool

LBP_TMTurret [DB73]	Signals for the turret (NC → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 40	Tool new: Internal T No. (INT)							
DBWn + 42	Original location of new tool in this circular magazine							

Start addresses of the turret:

Turret, 1:	After DBB4
Turret, 2:	After DBB48
Turret 3:	After DBB92
...	...
Turret 16:	After DBB664

10.4.13.2 LBP_TMTurret [DB73]

Variables in "LBP_TMTurret [DB73]"

The interface signals for the turret are stored in "LBP_TMTurret [DB73]".

IFNo (DB73.DBW0)

Description	Interface (IF) active If an interface is active, a valid data set exists for this interface. The index corresponds to the interface number (IF), e.g.: Index 1 corresponds to IF 1
Signal flow	NC → PLC
Data type	ARRAY[1..16] of BOOL
Update	Job-controlled
Value FALSE	The interface is **not** active.
Value TRUE	The interface is active.

Quit (DB73.DBW2)

Description	"auto" acknowledgement of interfaces 1 ... 16
Signal flow	PLC → NC
Data type	ARRAY[1..16] of BOOL
Update	Job-controlled
Value: FALSE	Reset by the basic program when TRUE is detected.
Value: TRUE	When bit is set (edge), the active command is acknowledged with Status:=1. Unless it is a relocation operation from a real magazine to a buffer. Then it is acknowledged with Status:=6. This reserves the source location.

TIF (DB73 from DBB4)

Description	Interface signals for tool change on spindles: Resolver 1: Index 1 Resolver 2: Index 2 ... Resolver 16: Index 16
Signal flow	NC → PLC
Data type	ARRAY[1..16] of STRUCT STRUCT "TIF" for turret (Page 535)
Update	Job-controlled

10.4.13.3 STRUCT "TIF" for turret

The variable structure "TIF" describes interface signals for tool change on turrets.

Variables in "TIF"

Comp (DB73 DBX4.0, 48.0, ..., 664.0)

Description	Command: Obligatory change The bits are not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMTurret.IfNo[i]" (DB73 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The new tool is fixed-location-coded.
Corresponds to	Position of participating tools

Perform (DB73 DBX4.1, 48.1, ..., 664.1)

Description	Command: Perform change The bits are not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMTurret.IfNo[i]" (DB73 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	Execute tool change
Corresponds to	"TIF.Perform" "TIF.Prepare"

T0 (DB73 DBX4.3, 48.3, ..., 664.3)

Description	Command: T0 (remove or replace tool) The bits are not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMTurret.IfNo[i]" (DB73 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	T0 (run spindle without tool) has been programmed.

ManTOut (DB73 DBX4.6, 48.6, ..., 664.6)

Description	Command: Unload manual tool The bits are not updated by the basic program until a new job exists for this interface. They are current only if the corresponding interface bit is TRUE in "LBP_TMTurret.IfNo[i]" (DB73 from DBX0.0). The user can reset the bit if necessary.
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The tool is changed by manual operation.

PerfStatusError (DB73 DBX5.0, 49.0, ..., 665.0)

Description	"auto" acknowledgement negative.
Signal flow	PLC → NC
Data type	BOOL
Update	Job-controlled
Value TRUE	The bit is only evaluated in conjunction with "auto" acknowledge. If it is set, the auto acknowledge is negative, i.e. with Status:=3.

DataInDB1073 (DB73 DBX5.7, 49.7, ..., 665.7)

Description	Data in the extended range See also
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value TRUE	The signal is set if a multitool is loaded. The multitool data are then available in "LBP_TMTurretMT [DB1073]".

Channel (DB73 DBB6, 50, ..., 666)

Description	Assigned channel
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Value	Number of the channel for which the active interface is valid.

TMNo (DB73 DBB7, 51, ..., 667)

Description	Tool management number
Signal flow	NC → PLC
Data type	BYTE
Update	Job-controlled
Value	Associated tool management number; corresponds to the number of the TO unit within a TO area.

VdiTcp0 (DB73 DBD8, 52, ..., 668)

VdiTcp1 (DB73 DBD12, 56, ..., 672)

VdiTcp2 (DB73 DBD16, 60, ..., 676)

Description	User-definable parameters • Parameter 0 (VdiTcp0) • Parameter 1 (VdiTcp1) • Parameter 2 (VdiTcp2)
Signal flow	NC → PLC
Data type	DINT
Update	Job-controlled
Value	A value can be passed to the PLC via the part program by programming: • $P_VDITCP[0] = value for user parameter 0 • $P_VDITCP[1] = value for user parameter 1 • $P_VDITCP[2] = value for user parameter 2 Parameters 0 ... 2 are transferred with the T command.

BufferIdent (DB73 DBW20, 64, ..., 680)

Description	Buffer magazine no. (fixed value 9998); target position for new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Magazine no. 9998 (buffer magazine); target magazine for new tool.

BufferNo (DB73 DBW22, 66, ..., 682)

Description	Location in buffer magazine (spindle)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location no. of buffer magazine to which the new tool must be loaded. This is normally the spindle. The location no. defined for this particular buffer during commissioning is output.

SMag (DB73 DBW24, 68, ..., 684)

Description	Magazine no. (source) for new tool to be loaded
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No magazine no. is output.
Value > 0	Magazine no. of the magazine from which the new spindle tool comes.
Corresponds to	"TIF.SLoc"

SLoc (DB73 DBW26, 70, ..., 686)

Description	Location no. (source) for new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No location no. is output.
Value > 0	Location no. of the magazine from which the new spindle tool comes.
Corresponds to	"TIF.SMag"

TMag (DB73 DBW28, 72, ..., 688)

Description	Magazine no. (target) for old tool to be removed
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No magazine number specified.
Value > 0	Number of the magazine in which the removed tool will be placed.
Corresponds to	"TIF.TLoc"

TLoc (DB73 DBW30, 74, ..., 690)

Description	Location no. (target) for old tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value 0	No location number specified.
Value > 0	Magazine location for tool that is unloaded at change.
Corresponds to	"TIF.TMag"

LocType (DB73 DBW32, 76, ..., 692)

Description	Location type of the new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled

Value	The location type of the new spindle tool is entered here.
Corresponds to	"TIF.SLeft" "TIF.SRight" "TIF.STop" "TIF.Bottom"

SLeft (DB73 DBW34, 78, ..., 694)

Description	Tool new: Size left
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **left** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SRight" "TIF.STop" "TIF.Bottom"

SRight (DB73 DBW36, 80, ..., 696)

Description	Tool new: Size right
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **right** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SLeft" "TIF.STop" "TIF.Bottom"

STop (DB73 DBW38, 82, ..., 698)

Description	Tool new: Size top
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **top** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SLeft" "TIF.SRight" "TIF.Bottom"

SBottom (DB73 DBW40, 84, ..., 700)

Description	Tool new: Size bottom
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled

Value	Specification of the tool size **bottom** in half locations for the new spindle tool.
Corresponds to	"TIF.LocType" "TIF.SLeft" "TIF.SRight" "TIF.Top"

TInBuffer, IgnoreDisabled, Unload, Load, MasterTool, Exchange_1to1, HandTool, Active, Enabled, Measured, Prewarn, Change, FixedLoc, Used

(DB73 from DBX42.0, 86.0, ..., 702.0)

Description	Tool status of the new tool	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Job-controlled	
Status bits	TInBuffer (DBX42.0, 86.0, ..., 702.0)	Tool in the buffer magazine
	IgnoreDisabled	Disabled, but ignore
	Unload	Tool to be unloaded
	Load	Tool to be loaded
	MasterTool	Master tool
	Exchange_1to1	1:1 exchange
	HandTool	Manual tool
	Active	Active tool
	Enabled	Enable tool
	Disabled	Tool disabled
	Measured	Measure tool
	Prewarning	Prewarning limit reached
	Change	Tool is being changed
	FixedLoc	Tool is fixed-location-coded
	Used (DBX43.7, 87.7, ..., 703.7)	Tool was in use

TNo (DB73.DBW44, 88, ..., 704))

Description	Tool new: Internal T number of the NC
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Display of internal T number of NC for the new spindle tool.

MyPlace (DB73 DBW46, 90, ..., 706)

Description	Original location of new tool
Signal flow	NC → PLC
Data type	INT

Update	Job-controlled
Value	Owner location of the new tool Corresponds to NC variables $A_MYMLN[T no.]

10.4.14 LBP_MFuncDecListConfig [DB75] - Decoding list for M decoding

10.4.14.1 M decoding acc. to list

Function description

Up to 256 M functions with extended address can be decoded from the basic program using the "M decoding according to list" function. The function is activated using the "LBP_ConfigBP [FC1]" parameter "ListMDecGrp" (number of M groups for decoding). The assignment of the M function with extended address and a signal in the signal list is defined in the decoding list. The signals are also grouped for this purpose.

Decoding list (LBP_MFuncDecListConfig [DB75])

The data block "LBP_MFuncDecListConfig [DB75]" is supplied with the basic program. The upgrade library does not contain this data block to prevent unintentional overwriting a "LBP_MFuncDecListConfig [DB75]" adapted by you during an upgrade. Always import this data block from the complete library when using the M decoding functionality according to the list for the first time.

An M function is decoded if it is in the decoding list. When decoding the M function, the corresponding signal is set in the signal list as a function of the specific group. When setting a signal in the signal list, the interface signal "Read in inhibit" is set by the basic program in the associated channel of the NC. The interface signal is reset again for the channel as soon as the user resets all of the signals output from this channel in the signal list; i.e. after they have been acknowledged.

Signal list (LBP_MFuncDecListSignals [DB76])

When activating the function in data block "LBP_MFuncDecListSignals [DB76]", the basic program creates the signal list. From then, for each M signal decoded according to the list, a signal is set in the signal list in the corresponding group. At the same time, the "Readin inhibit" interface signal is set in the channel in which the M function has been output. The interface signal is reset again for the channel as soon as the user resets all of the signals output from this channel in the signal list; i.e. after they have been acknowledged.

Highspeed auxiliary functions

When an M function contained in the decoding list is output as "fast help function", no read-in inhibit is set for the corresponding channel of the NC.

The figure below shows the structure of the **M decoding according to list**:

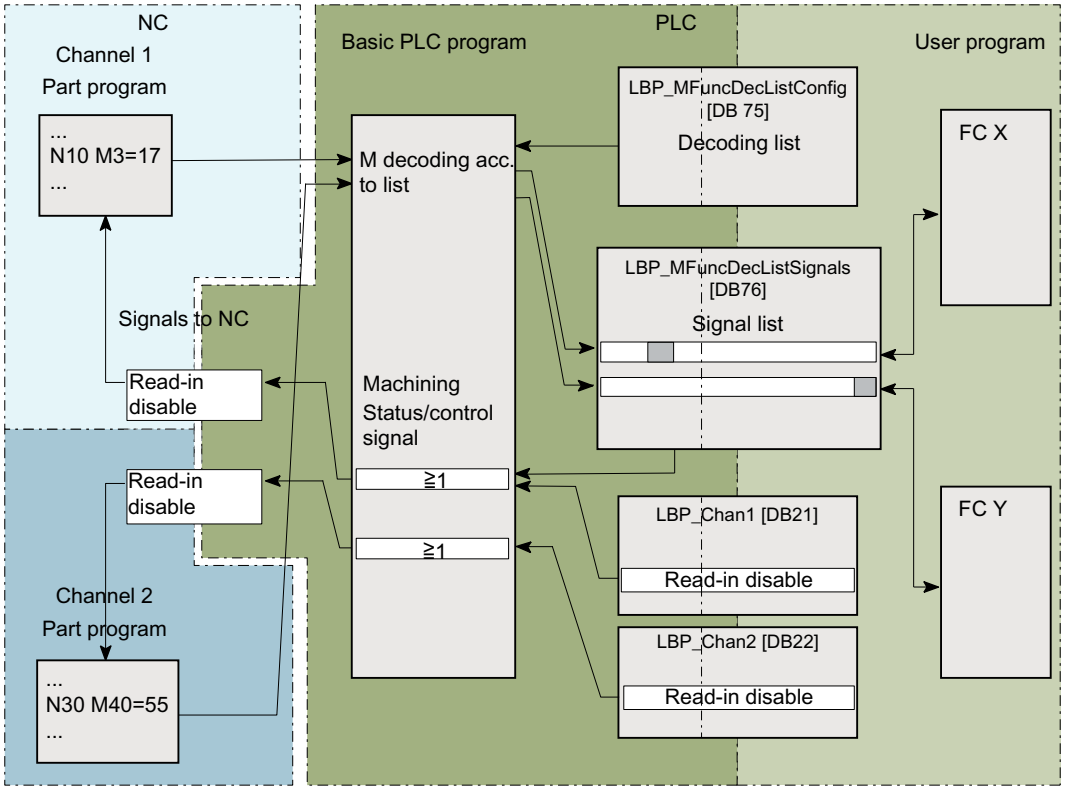


Figure 10-1 M decoding acc. to list

Activation

The M decoding is activated via the "LBP_ConfigBP [FC1]" parameter ListMDecGrp**.**

The number of M groups to the evaluated and/or decoded is specified using the appropriate parameter. The function is activated at a parameter value = 1 to 16.

- Basic program, startup OB (e.g. "Startup [OB100]"), "LBP_ConfigBP [FC1]" parameter ListMDecGrp = <number of M groups>.

Properties and structure of the decoding list (LBP_MFuncDecListConfig [DB75])

Properties of the decoding list:

- There is only one channel-independent decoding list.
- The decoding list can include a maximum of 16 groups.
- A group has a maximum of 16 signals

- There must be an entry in the decoding list for every group of M functions to be decoded.
- The assignment between the M function with extended address and the signal to be set in the signal list is specified in the decoding list using the first and last M function of the associated group.
  - First M function: Parameter: "MFirstAdr" ≙ signal or bit 0
  - Last M function: Parameter: "MLastAdr" ≙ dependent on the difference to the first M function maximum signal or bit 15

Structure of decoding list:

An entry in the decoding lists consists of 3 parameters, each of which is assigned to a group.

Group	Extended M address	First M address of the group	Last M address of the group
1	MSigGrp[1].MExtAdr	MSigGrp[1].MFirstAdr	MSigGrp[1].MLastAdr
2	MSigGrp[2].MExtAdr	MSigGrp[2].MFirstAdr	MSigGrp[2].MLastAdr
...	...	...	...
16	MSigGrp[16].MExtAdr	MSigGrp[16].MFirstAdr	MSigGrp[16].MLastAdr

Type and value range of the signals:

Signal	Type	Value range	Meaning
MExtAdr	INT	0 ... 99	Extended M address
MFirstAdr	DINT	0 to 99.999.999	First M address in group
MLastAdr	DINT	0 to 99.999.999	Last M address in group

Properties of the signal list (LBP_MFuncDecListSignals [DB76])

The signal list has the following properties:

- There is only one signal list independent of the channel.
- The signal list can include a maximum of 16 signals for each M group.

Example

3 groups of M functions are to be decoded:

- Group 1: M2 = 1 to M2 = 5
- Group 2: M3 = 12 to M3 = 23
- Group 3: M40 = 55

Structure of the decoding and signal list

Group	Decoding list (LBP_MFuncDecListConfig [DB75])			Signal list (LBP_MFuncDecListSignals [DB76])
	Extended M address	First M address of the group	Last M address of the group	
1	2	1	5	"LBP_MFuncDecListSignals.MSigGrp1.MSig[i]", Index 0 ... 4 (DB76 DBX0.0 ... 0.4)
2	3	12	23	"LBP_MFuncDecListSignals.MSigGrp2.MSig[i], Index 0 ... 11 (DB76 DBX2.0 ... 3.3)
3	40	55	55	"LBP_MFuncDecListSignals.MSigGrp3.MSig[0]" (DB76.DBX4.0)

Program code (SCL):

```
"LBP_MFuncDecListConfig".MSigGrp[1].MExtAdr // Extended M address of the
 := 2;                                      1st group
"LBP_MFuncDecListConfig".MSigGrp[1].MFirstA // First M address of the group
dr := 1;
"LBP_MFuncDecListConfig".MSigGrp[1].MLastAd // Last M address of the group
r := 5;


"LBP_MFuncDecListConfig".MSigGrp[2].MExtAdr // Extended M address of the
 := 3;                                      2nd group
"LBP_MFuncDecListConfig".MSigGrp[2].MFirstA // First M address of the group
dr := 12;
"LBP_MFuncDecListConfig".MSigGrp[2].MLastAd // Last M address of the group
r := 23;


"LBP_MFuncDecListConfig".MSigGrp[3].MExtAdr // Extended M address of the
 := 40;                                     3rd group
"LBP_MFuncDecListConfig".MSigGrp[3].MFirstA // First M address of the group
dr := 55;
"LBP_MFuncDecListConfig".MSigGrp[3].MLastAd // Last M address of the group
r := 55;
```

Call example of the "LBP_ConfigBP [FC1]" in the "Startup [OB100]" (SCL)

To activate the function, insert the parameter for the number of M groups to be decoded "ListMDecGrp".

```
"LBP_ConfigBP"(
   …
   ListMDecGrp := 3;    //M decoding of three groups
   …);
```

Description

After the entry for the call of "LBP_ConfigBP [FC1]" has been inserted in the startup OB, a restart must be carried out.

An NC program is then started, for instance in the 1st channel. An extended M function is included in this (M3=17). When decoding the M function (M3 ≙ group 2), the associated signal "LBP_MFuncDecListSignals.MSigGrp2.MSig[5] (DB76.DBX2.5) is set in the signal list and the interface signal "Read-inhibit" in the 1st channel. The execution of the NC program is stopped. Further, the "Extended address M function" and the "M function number" are displayed in the channel DB of the 1st channel.

The "Read-in inhibit" signal in the 1st channel is reset once the user has reset and thus acknowledged all of the signals output from this channel in the signal list.

10.4.14.2 LBP_MFuncDecListConfig [DB75]

Variables in "LBP_MFuncDecListConfig [DB75]"

The decoding list is saved in "LBP_MFuncDecListConfig [DB75]".

MSigGrp (DB75 ab DBB0)

Description	Decoding list
Data type	ARRAY[1..16] of STRUCT STRUCT "MSigGrp" group parameter (Page 545)

See also

M decoding acc. to list (Page 541)

10.4.14.3 STRUCT "MSigGrp" group parameter

The variable structure "MSigGrp" describes the parameters of a group with the address data.

Variables in "MSigGrp"

MExtAdr (DB75 DBW0, 10, ..., 150)

Description	Extended M address
Data type	INT
Range of values	0 ... 99
Corresponds to	• "LBP_MFuncDecListSignals" (DB76)

MFirstAdr (DB75 DBD2, 12, ..., 152)

Description	First M address in group
Data type	DINT

Range of values	0 ... 99999999
Corresponds to	• "LBP_MFuncDecListSignals" (DB76)

MLastAdr (DB75 DBD6, 16, ..., 156)

Description	Last M address in group
Data type	DINT
Range of values	0 ... 99999999
Corresponds to	• "LBP_MFuncDecListSignals" (DB76)

10.4.15 LBP_MFuncDecListSignals [DB76] - Signal list for M decoding

10.4.15.1 LBP_MFuncDecListSignals [DB76]

Variables in "LBP_MFuncDecListSignals [DB76]"

The signal list is saved in "LBP_MFuncDecListSignals [DB76]".

MSigGrp1, MSigGrp2, MSigGrp3, MSigGrp4, MSigGrp5, MSigGrp6, MSigGrp7, MSigGrp8

(DB76 DBW0, 2, 4, 6, 8, 10, 12, 14)

MSigGrp9, MSigGrp10, MSigGrp11, MSigGrp12, MSigGrp13, MSigGrp14, MSigGrp15, MSigGrp16

(DB76 DBW16, 18, 20, 22, 24, 26, 28, 30)

Description	Signal list
Data type	STRUCT STRUCT "MSigGrp*" signal list of a group (Page 546)

See also

M decoding acc. to list (Page 541)

10.4.15.2 STRUCT "MSigGrp*" signal list of a group

The variable structure "MSigGrp*" contains the signal list of a group.

Variables in "MSigGrp*"

MSig (DB76 DBW0, 2, ..., 30)

Description	Signal list of a group
Data type	ARRAY[0..15] of BOOL
Corresponds to	• "LBP_MFuncDecListConfig" (DB75)

10.4.16 LBP_PowerrideConfig [DB78] - preconfigured behavior for Powerride

10.4.16.1 Overview

LBP_PowerrideConfig [DB78] - Signal overview

Table 10-46 Signal overview

LBP_PowerrideConfig [DB78]	Signals		
DBB0	**Feed configuration** - UDT "LBP_typePowerrideConfig" (Page 553)		
		Switch position that represents the override factor 100 %	
DBB1		Switch position that represents the maximum override factor	
DBD2		Time delay for exceeding the 100% override factor	
DBB6		**Switch position 1** - UDT "LBP_typePowerrideConfigPos (Page 552)	
			Override factor
DBB7			Color of the LEDs of the LED strip
DBB8			Vibration pattern
DBB10		**Switch position 2** - UDT "LBP_typePowerrideConfigPos (Page 552)	
			...
...			...
DBB146		**Switch position 36** - UDT "LBP_typePowerrideConfigPos (Page 552)	
			Override factor
DBB147			Color of the LEDs of the LED strip
DBB148			Vibration pattern

LBP_PowerrideConfig [DB78]	Signals		
DBB150	**Spindle configuration -** UDT "LBP_typePowerrideConfig" (Page 553)		
		Switch position that represents the override factor 100 %	
DBB151		Switch position that represents the maximum override factor	
DBD152		Time delay for exceeding the 100% override factor	
DBB156		**Switch position 1** - UDT "LBP_typePowerrideConfigPos (Page 552)	
			Override factor
DBB157			Color of the LEDs of the LED strip
DBB158			Vibration pattern
DBB160		**Switch position 2** - UDT "LBP_typePowerrideConfigPos (Page 552)	
			...
...			...
DBB296		**Switch position 36** - UDT "LBP_typePowerrideConfigPos (Page 552)	
			Override factor
DBB297			Color of the LEDs of the LED strip
DBB298			Vibration pattern

10.4.16.2 LBP_PowerrideConfig [DB78]

Variables in "LBP_PowerrideConfig"

The data block "LBP_PowerrideConfig" contains preconfigured behavior for the Powerride. Transfer the desired configuration to function block "LBP_Powerride" at parameter "configuration".

Use a separate data block for user-specific configurations. The configurations in this data block are overwritten by updates.

Preconfigured behavior for the Powerride

configFeed

Description	Feed configuration
Data type	LBP_typePowerrideConfig (Page 553)

configSpindle

Description	Spindle configuration
Data type	LBP_typePowerrideConfig (Page 553)

See also

LBP_Powerride [FB21]: rotary/push control dial (Page 648)

10.4.16.3 UDT "LBP_typePowerrideInput" input data

Variables in "LBP_typePowerrideInput"

pushButton

Description	State of the Powerride With TRUE the switch is pressed
Data type	BOOL

powerrideCounter

Description	Position of the rotary wheel [0..255] Modulo, 256 notches at 36 notches per revolution
Data type	USINT

10.4.16.4 UDT "LBP_typePowerrideOutput" output data

Variables in "LBP_typePowerrideOutput"

ledInfo

Description	Control of the individual LEDs of the LED strip from Powerride With TRUE the respective LED lights up. **Bit assignment**							
	Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
	0	-	-	-	-	-	-	-	LED 24
	1	LED 23	LED 22	LED 21	LED 20	LED 19	LED 18	LED 17	LED 16
	2	LED 15	LED 14	LED 13	LED 12	LED 11	LED 10	LED 9	LED 8
	3	LED 7	LED 6	LED 5	LED 4	LED 3	LED 2	LED 1	LED 0
Data type	DWORD								

scaleRed, scaleGreen, scaleBlue

Description	Color of the LED strip "scaleRed", "scaleGreen", "scaleBlue" Color combinations (Page 653) are possible.
Data type	BOOL

buttonRed, buttonGreen, buttonBlue

Description	Color of the switch LED "buttonRed", "buttonGreen", "buttonBlue" Color combinations (Page 653) are possible.
Data type	BOOL

vibrationPattern

Description	Vibration pattern [0..7] (at 0 no vibration)
Data type	BYTE

10.4.16.5 UDT "LBP_typePowerrideConfigPos" configuration data for a position

Variables in "LBP_typePowerrideConfigPos"

value

Description	Override factor [0 ... 200] The value represents the override factor for a specific switch position. 100 corresponds to 100%, for example
Data type	USINT

ledColor

Description	Color of the LEDs from the LED strip of the Powerride (color combinations (Page 653) are possible) **Bit assignment**		
	Bit 2	**Bit 1**	**Bit 0**
	Blue	Green	Red
Data type	USINT		

vibrationIndex

Description	Vibration pattern [0 ... 7] (at 0 no vibration)
Data type	USINT

10.4.16.6 UDT "LBP_typePowerrideConfig" configuration data

Variables in "LBP_typePowerrideConfig"

pos100

Description	Switch position that represents the override factor 100% [1 ... 36]
Data type	USINT

posMax

Description	Switch position that represents the maximum override factor [1 ... 36]
Data type	USINT

delay100

Description	Time delay for exceeding the 100% override factor When the switch position for "pos100" is reached, the time delay is started. Only after the time has expired can you increase the switch position further.
Data type	TIME

switchPos

Description	Definition of the switch positions Each data record in the array represents a switch position. Define the corresponding properties for the respective switch position.
Data type	ARRAY[1..36] of LBP_typePowerrideConfigPos (Page 552)

Note

Activating changes to the configuration

The configuration is read in again with a positive edge on parameter "enable". To do this, set the parameter to FALSE and then back to TRUE.

10.4.17 LBP_CtrlEnergy [DB1000] - Energy-saving profiles

10.4.17.1 Overview

LBP_CtrlEnergy [DB1000], energy-saving profiles

Table 10-47 Energy-saving profiles

LBP_CtrlEnergy [DB1000]	**Ctrl-Energy (operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 0	Control signals							
							Set time to prewarning limit	Immediately activate energy-saving profile
DBBn + 1	Control signals							
								Directly activate energy-saving profile
DBBn + 2	Signals to check/test the energy-saving profile							
							PLC user signal	Master computer signal
DBBn + 3	Reserved							
DBBn + 4	Status signal							
							Activation time T1 has expired	Energy-saving profile active
DBBn + 5	Reserved							
DBWn + 6	Actual value: Actual value T1							
DBWn + 8	Actual value: Actual value T2							
DBBn + 10	Effectiveness, profile							
							Disable energy-saving profile	Energy-saving profile configured
	State conditions							
DBBn + 11								
						Screen change active	Data transfer active	Keyboard has been used

LBP_CtrlEnergy [DB1000]	**Ctrl-Energy (operating software → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 12								
								MP used for operation
DBBn + 13	NC channel in reset							
	8	7	6	5	4	3	2	1
DBBn + 14	NC channel in reset							
							10	9
DBBn + 15								
							PLC user signal	Master computer signal
DBWn + 16	Activation time T1							
DBWn + 18	Pre-warning time T2							

Additional profile instances

Energy profile 2: DB1000 DBB20...DBB39
Energy profile 3: DB1000 DBB40...DBB59
Energy profile 4: DB1000 DBB60...DBB79
Energy profile 5: DB1000 DBB80...DBB99
Energy profile 6: DB1000 DBB100...DBB119
Energy profile 7: DB1000 DBB120...DBB139
Energy profile 8: DB1000 DBB140...DBB159

10.4.17.2 LBP_CtrlEnergy [DB1000]

The data block "LBP_CtrlEnergy [DB1000]" contains the energy-saving profiles.

The profile instances are assigned as follows:

Energy profile 1:	"LBP_CtrlEnergy.Profile[1]" (DB1000 from DBB0)
Energy profile 2:	"LBP_CtrlEnergy.Profile[2]" (DB1000 from DBB20)
Energy profile 3:	"LBP_CtrlEnergy.Profile[3]" (DB1000 from DBB40)
Energy profile 4:	"LBP_CtrlEnergy.Profile[4]" (DB1000 from DBB50)
Energy profile 5:	"LBP_CtrlEnergy.Profile[5]" (DB1000 from DBB80)

Energy profile 6:	"LBP_CtrlEnergy.Profile[6]" (DB1000 from DBB100)
Energy profile 7:	"LBP_CtrlEnergy.Profile[7]" (DB1000 from DBB120)
Energy profile 8:	"LBP_CtrlEnergy.Profile[8]" (DB1000 from DBB140)

Variables in "LBP_CtrlEnergy [DB1000]"

Archive (DB1000 from DBB0)

Description	Energy-saving profiles 1 ... 8
Signal flow	Operating software → PLC
Data type	ARRAY[1..8] of STRUCT STRUCT "Profiles" for energy-saving profiles (Page 556)

10.4.17.3 STRUCT "Profiles" for energy-saving profiles

The variable structure "Profile" describes the energy-saving profiles.

Variables in "Profile"

ESaveOn (DB1000 DBX0.0, 20.0, ..., 140.0)

Description	Control signal: Immediately activate energy-saving profile
Signal flow	Operating software → PLC
Data type	BOOL

ReloadT2 (DB1000 DBX0.1, 20.1, ..., 140.1)

Description	Control signal: Set time to prewarning limit
Signal flow	Operating software → PLC
Data type	BOOL

HmiESaveOn (DB1000 DBX1.0, 21.0, ..., 141.0)

Description	Control signal: Directly activate energy-saving profile
Signal flow	Operating software → PLC
Data type	BOOL

MasterCompSig (DB1000 DBX2.0, 22.0, ..., 142.0)

Description	Signal to check/test the energy-saving profile Master computer signal
Signal flow	Operating software → PLC
Data type	BOOL

PlcUserSig (DB1000 DBX2.1, 22.1, ..., 142.1)

Description	Signal to check/test the energy-saving profile PLC user signal
Signal flow	Operating software → PLC
Data type	BOOL

ESaveIsOn (DB1000 DBX4.0, 24.0, ..., 144.0)

Description	Status signal: Energy-saving profile active
Signal flow	Operating software → PLC
Data type	BOOL

T1Expired (DB1000 DBX4.1, 24.1, ..., 144.1)

Description	Status signal: Activation time T1 has expired
Signal flow	Operating software → PLC
Data type	BOOL

ActValT1 (DB1000 DBW6, 26, ..., 146)

Description	Actual value: Actual value T1
Signal flow	Operating software → PLC
Data type	INT

ActValT2 (DB1000 DBW8, 28, ..., 148)

Description	Actual value: Actual value T2
Signal flow	Operating software → PLC
Data type	INT

ESaveValid (DB1000 DBX10.0, 30.0, ..., 150.0)

Description	Effectiveness, profile Energy-saving profile configured
Signal flow	Operating software → PLC
Data type	BOOL

ESaveOff (DB1000 DBX10.1, 30.1, ..., 150.1)

Description	Effectiveness, profile Disable energy-saving profile
Signal flow	Operating software → PLC
Data type	BOOL

CondOpPanel (DB1000 DBX11.0, 31.0, ..., 151.0)

Description	State condition: Keyboard has been used
Signal flow	Operating software → PLC
Data type	BOOL

CondDataTransfer (DB1000 DBX11.1, 31.1, ..., 151.1)

Description	State condition: Data transfer active
Signal flow	Operating software → PLC
Data type	BOOL

CondWindowChange (DB1000 DBX11.2, 31.2, ..., 151.2)

Description	State condition: Screen change active
Signal flow	Operating software → PLC
Data type	BOOL

CondMCP (DB1000 DBX12.0, 32.0, ..., 152.0)

Description	State condition: MCP operating
Signal flow	Operating software → PLC
Data type	BOOL

CondNcChan1, CondNcChan2, CondNcChan3, CondNcChan4, CondNcChan5

(DB1000 from DBX13.0, 33.0, ..., 153.0)

CondNcChan6, CondNcChan7, CondNcChan8, CondNcChan9, CondNcChan10

(DB1000 from DBX13.5, 33.5, ..., 153.5)

Description	State condition: NC channel in reset • NC channel 1: "LBP_CtrlEnergy.CondNcChan1" (DB1000 DBX13.0, 33.0, ..., 153.0) • NC channel 2: "LBP_CtrlEnergy.CondNcChan2" (DB1000 DBX13.1, 33.1, ..., 153.1) • ... • NC channel 10: "LBP_CtrlEnergy.CondNcChan10" (DB1000 DBX14.1, 34.1, ..., 154.1)
Signal flow	Operating software → PLC
Data type	BOOL

CondMasterCompSig (DB1000 DBX15.0, 35.0, ..., 155.0)

Description	State condition: Master computer signal
Signal flow	Operating software → PLC
Data type	BOOL

CondPlcUserSig (DB1000 DBX15.1, 35.1, ..., 155.1)

Description	State condition: PLC user signal
Signal flow	Operating software → PLC
Data type	BOOL

PresetT1 (DB1000 DBW16, 36, ..., 156)

Description	Activation time T1
Signal flow	Operating software → PLC
Data type	INT

PresetT2 (DB1000 DBW18, 38, ..., 158)

Description	Pre-warning time T2
Signal flow	Operating software → PLC
Data type	INT

10.4.18 LBP_SentronPac [DB1001] - SENTRON PAC

10.4.18.1 Overview

LBP_SentronPac [DB1001], SENTRON PAC

Table 10-48 Signals for SENTRON PAC

LBP_SentronPac [DB1001]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB0 PLC → operating software			Regenerative energy: Integration using FW	Regenerative energy: Integration using FW	Regenerative energy is read by SENTRON PAC	Energy fed in is read by SENTRON PAC	SENTRON PAC represents the machine	Display manual value
DBB1 GP								GP should perform measurement
DBB2								
Operating software → PLC								Measurement in progress
DBB3 PLC → operating software								Power display on
DBD4	Manual value (REAL) to operating software							
PLC → operating software								
DBD8	Total active power (REAL) to operating software							
PLC → operating software								

LBP_SentronPac [DB1001]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBD12	Measured drawn active energy in kWh (REAL) to the operating software							
PLC → operating software								
DBD16	Measured supplied active energy in kWh (REAL) to the operating software							
PLC → operating software								
DBD20	Total active power in watts (REAL) from SENTRON							
GP								
DBD24	Drawn active energy at tariff 1 (F) in Wh (REAL) from SENTRON							
GP								
DBD28	Supplied active energy at tariff 1 (F) in Wh (REAL) from SENTRON							
GP								
DBD32	Drawn active energy in kWh per day (REAL) to the operating software							
DBD36	Supplied active energy in kWh day (REAL) to the operating software							
DBD40	Drawn active energy in kWh previous day (REAL) to the operating software							
DBD44	Supplied active energy in kWh previous day (REAL) to the operating software							
DBD48	Drawn active energy in kWh month (REAL) to the operating software							
DBD52	Supplied active energy in kWh month (REAL) to the operating software							
DBD56	Drawn active energy in kWh previous month (REAL) to the operating software							
DBD60	Supplied active energy in kWh previous month (REAL) to the operating software							
DBD64	Drawn active energy in kWh year (REAL) to the operating software							
DBD68	Supplied active energy in kWh year (REAL) to the operating software							
DBD72	Drawn active energy in kWh previous year (REAL) to the operating software							
DBD76	Supplied active energy in kWh previous year (REAL) to the operating software							
DBB80 ... DBB95	Reserved							

LBP_SentronPac [DB1001]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBB96	ProductionAct							
PLC→GP/ operating software								
DBB97						Values invalid in		
GP						DBD28	DBD24	DBD20
DBB98	Values invalid in							
GP	DBD384	DBD344	DBD304	DBD264	DBD224	DBD184	DBD144	DBD104
DBB99							Values invalid in	
GP							DBD464	DBD424

LBP_SentronPac [DB1001], SENTRON PAC auxiliary devices

Table 10-49 Signals for SENTRON PAC

LBP_Sentron-Pac [DB1001]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn+100	Control/command bits							
PLC → GP operating software	Reset data structure			Input mode: (energy or power)	Different values when measuring energy	Energy measurement running	Read energy actual value update	Process device
DBBn+101	Reserved							
DBBn+102	Reserved							
DBBn+104	Active power or active energy of the auxiliary unit [kW] or [kWh]							
PLC → GP operating software								
DBDn+108	Drawn active energy of the auxiliary unit [kWh]							
PLC → operating software								
DBDn+112	Active energy supplied by the auxiliary unit [kWh]							
GP → operating software								
DBDn+116	Active energy drawn by the auxiliary unit at the measurement start [kWh]							
GP → operating software								

LBP_Sentron-Pac [DB1001]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBDn+120	Active energy supplied by the auxiliary unit at the measurement start [kWh]							
GP → operating software								
DBDn+124	Active energy drawn by the auxiliary unit at the measurement end [kWh]							
GP → operating software								
DBDn+128	Active energy supplied by the auxiliary unit at the measurement end [kWh]							
GP → operating software								
DBDn+132	Reserved							
DBDn+136	Reserved							

Auxiliary unit instances:

Auxiliary unit 1	(n=0):	DB1001 DBB100...DBB139
Auxiliary unit 2	(n=40):	DB1001 DBB140...DBB179
Auxiliary unit 3	(n=80):	DB1001 DBB180...DBB219
Auxiliary unit 4	(n=120):	DB1001 DBB220...DBB259
Auxiliary unit 5	(n=160):	DB1001 DBB260...DBB299
Auxiliary unit 6	(n=200):	DB1001 DBB300...DBB339
Auxiliary unit 7	(n=240):	DB1001 DBB340...DBB379
Auxiliary unit 8	(n=280):	DB1001 DBB380...DBB419
Auxiliary unit 9	(n=320):	DB1001 DBB420...DBB459
Auxiliary unit 10	(n=360):	DB1001 DBB460...DBB499

10.4.18.2 LBP_SentronPac [DB1001]

The data block "LBP_SentronPac [DB1001]" contains signals for SENTRON PAC.

Variables in "LBP_SentronPac [DB1001]"

ManualVal (DB1001.DBX0.0)

Description	Display manual value
Signal flow	PLC → operating software
Data type	BOOL

SentronPacMach (DB1001.DBX0.1)

Description	SENTRON PAC represents the machine
Signal flow	PLC → operating software
Data type	BOOL

FeedInEnergyReadSP (DB1001.DBX0.2)

Description	Energy fed in is read by SENTRON PAC
Signal flow	PLC → operating software
Data type	BOOL

FeedBackEnergyReadSP (DB1001.DBX0.3)

Description	Regenerative energy is read by SENTRON PAC
Signal flow	PLC → operating software
Data type	BOOL

FeedInEnergyReadAuxil (DB1001.DBX0.4)

Description	Regenerative energy: Integration using FW
Signal flow	PLC → operating software
Data type	BOOL

FeedBackEnergyReadAuxil (DB1001.DBX0.5)

Description	Regenerative energy: Integration using FW
Signal flow	PLC → operating software
Data type	BOOL

MeasAuto (DB1001.DBX1.0)

Description	GP should perform measurement
Signal flow	GP
Data type	BOOL

MeasAct (DB1001.DBX2.0)

Description	Measurement in progress
Signal flow	Operating software → PLC
Data type	BOOL

EStateOn (DB1001.DBX3.0)

Description	Power display on
Signal flow	PLC → operating software
Data type	BOOL

ManualValue (DB1001.DBD4)

Description	Manual value to operating software
Signal flow	PLC → operating software
Data type	REAL

ActivePowerItem (DB1001.DBD8)

Description	Total active power to operating software
Signal flow	PLC → operating software
Data type	REAL

ActiveEnergyItem (DB1001.DBD12)

Description	Measured imported active energy in [kWh] to operating software
Signal flow	PLC → operating software
Data type	REAL

ReactiveEnergyItem (DB1001.DBD16)

Description	Measured exported active energy in [kWh] to operating software
Signal flow	PLC → operating software
Data type	REAL

ActivePowerSentron (DB1001.DBD20)

Description	Total active power in [W] of SENTRON
Signal flow	GP
Data type	REAL

ActiveEnergySentron (DB1001.DBD24)

Description	Drawn active energy at tariff 1 (F) in [Wh] from SENTRON
Signal flow	GP
Data type	REAL

ReactiveEnergySentron (DB1001.DBD28)

Description	Exported active energy at tariff 1 (F) in [Wh] from SENTRON
Signal flow	GP
Data type	REAL

ActiveEnergyDay (DB1001.DBD32)

Description	Imported active energy in [kWh] per day to the operating software
Data type	REAL

ReactiveEnergyDay (DB1001.DBD36)

Description	Exported active energy in [kWh] per day to operating software
Data type	REAL

ActiveEnergyPrevDay (DB1001.DBD40)

Description	Imported active energy in [kWh] previous day to operating software
Data type	REAL

ReactiveEnergyPrevDay (DB1001.DBD44)

Description	Exported active energy in [kWh] of previous day to operating software
Data type	REAL

ActiveEnergyMonth (DB1001.DBD48)

Description	Imported active energy in [kWh] per month to the operating software
Data type	REAL

ReactiveEnergyMonth (DB1001.DBD52)

Description	Exported active energy in [kWh] per month to operating software
Data type	REAL

ActiveEnergyPrevMonth (DB1001.DBD56)

Description	Imported active energy in kWh previous month to operating software
Data type	REAL

ReactiveEnergyPrevMonth (DB1001.DBD60)

Description	Exported active energy in [kWh] of previous month to operating software
Data type	REAL

ActiveEnergyYear (DB1001.DBD64)

Description	Imported active energy in [kWh] per year to the operating software
Data type	REAL

ReactiveEnergyYear (DB1001.DBD68)

Description	Exported active energy in [kWh] per year to operating software
Data type	REAL

ActiveEnergyPrevYear (DB1001.DBD72)

Description	Imported active energy in [kWh] of previous year to operating software
Data type	REAL

ReactiveEnergyPrevYear (DB1001.DBD76)

Description	Exported active energy in [kWh] of previous year to operating software
Data type	REAL

ProductionAct (DB1001.DBB96)

Description	ProductionAct
Signal flow	PLC → GP / operating software
Data type	BYTE

ErrDBD20_NAN (DB1001.DBX97.0)

Description	Values invalid in DBD20
Signal flow	GP
Data type	BOOL

ErrDBD24_NAN (DB1001.DBX97.1)

Description	Values invalid in DBD24
Signal flow	GP
Data type	BOOL

ErrDBD28_NAN (DB1001.DBX97.2)

Description	Values invalid in DBD28
Signal flow	GP
Data type	BOOL

ErrDBD104_NAN (DB1001.DBX98.0)

Description	Values invalid in DBD104
Signal flow	GP
Data type	BOOL

ErrDBD144_NAN (DB1001.DBX98.1)

Description	Values invalid in DBD144
Signal flow	GP
Data type	BOOL

ErrDBD184_NAN (DB1001.DBX98.2)

Description	Values invalid in DBD184
Signal flow	GP
Data type	BOOL

ErrDBD224_NAN (DB1001.DBX98.3)

Description	Values invalid in DBD224
Signal flow	GP
Data type	BOOL

ErrDBD264_NAN (DB1001.DBX98.4)

Description	Values invalid in DBD264
Signal flow	GP
Data type	BOOL

ErrDBD304_NAN (DB1001.DBX98.5)

Description	Values invalid in DBD304
Signal flow	GP
Data type	BOOL

ErrDBD344_NAN (DB1001.DBX98.6)

Description	Values invalid in DBD344
Signal flow	GP
Data type	BOOL

ErrDBD384_NAN (DB1001.DBX98.7)

Description	Values invalid in DBD384
Signal flow	GP
Data type	BOOL

ErrDBD424_NAN (DB1001.DBX99.0)

Description	Values invalid in DBD424
Signal flow	GP
Data type	BOOL

ErrDBD464_NAN (DB1001.DBX99.1)

Description	Values invalid in DBD464
Signal flow	GP
Data type	BOOL

CtrlE_AuxDevice (DB1001 from DBB100)

Description	Signals of SENTRON PAC auxiliary units
Data type	ARRAY[1..10] of STRUCT STRUCT "CtrlE_AuxDevice" for SENTRON PAC auxiliary units (Page 567)

10.4.18.3 STRUCT "CtrlE_AuxDevice" for SENTRON PAC auxiliary units

The variable structure "CtrlE_AuxDevice" describes the signals of the SENTRON PAC auxiliary units.

The auxiliary unit instances are assigned as follows:

Auxiliary unit 1:	"LBP_SentronPac.CtrlE_AuxDevice[1]" (DB1001 from DBB100)
Auxiliary unit 2:	"LBP_SentronPac.CtrlE_AuxDevice[2]" (DB1001 from DBB140)

Auxiliary unit 3:	"LBP_SentronPac.CtrlE_AuxDevice[3]" (DB1001 from DBB180)
Auxiliary unit 4:	"LBP_SentronPac.CtrlE_AuxDevice[4]" (DB1001 from DBB220)
Auxiliary unit 5:	"LBP_SentronPac.CtrlE_AuxDevice[5]" (DB1001 from DBB260)
Auxiliary unit 6:	"LBP_SentronPac.CtrlE_AuxDevice[6]" (DB1001 from DBB300)
Auxiliary unit 7:	"LBP_SentronPac.CtrlE_AuxDevice[7]" (DB1001 from DBB340)
Auxiliary unit 8:	"LBP_SentronPac.CtrlE_AuxDevice[8]" (DB1001 from DBB380)
Auxiliary unit 9:	"LBP_SentronPac.CtrlE_AuxDevice[9]" (DB1001 from DBB420)
Auxiliary unit 10:	"LBP_SentronPac.CtrlE_AuxDevice[10]" (DB1001 from DBB460)

Variables in "CtrlE_AuxDevice"

ProcessAux (DB1001 DBX100.0, 140.0, ..., 460.0)

Description	Control bit / command bit: Process device
Signal flow	PLC → GP operating software
Data type	BOOL

ReadEnergy (DB1001 DBX100.1, 140.1, ..., 460.1)

Description	Control bit / command bit: Reading energy actual value updating
Signal flow	PLC → GP operating software
Data type	BOOL

MeasAct (DB1001 DBX100.2, 140.2, ..., 460.2)

Description	Control bit / command bit: Energy measurement running
Signal flow	PLC → GP operating software
Data type	BOOL

MeasMode (DB1001 DBX100.3, 140.3, ..., 460.3)

Description	Control bit / command bit: Different values when measuring energy
Signal flow	PLC → GP operating software
Data type	BOOL

InputMode (DB1001 DBX100.4, 140.4, ..., 460.4)

Description	Control bit / command bit: Input mode (energy or power)
Signal flow	PLC → GP operating software
Data type	BOOL

ResetAux (DB1001 DBX100.7, 140.7, ..., 460.7)

Description	Control bit / command bit: Reset data structure
Signal flow	PLC → GP operating software
Data type	BOOL

ActivePowerAux (DB1001 DBD104, 144, ..., 464)

Description	Active power or active energy of the auxiliary unit in [kW] or [kWh]
Signal flow	PLC → GP operating software
Data type	REAL

ActiveEnAux (DB1001 DBD108, 148, ..., 468)

Description	Active energy drawn by the auxiliary unit in [kWh]
Signal flow	PLC → operating software
Data type	REAL

ReActiveEnAux (DB1001 DBD112, 152, ..., 472)

Description	Active energy supplied by the auxiliary unit in [kWh]
Signal flow	GP → operating software
Data type	REAL

ActiveEnAuxMeasON (DB1001 DBD116, 156, ..., 476)

Description	Active energy drawn by the auxiliary unit at the start of measurement in [kWh]
Signal flow	GP → operating software
Data type	REAL

ReActiveEnAuxMeasON (DB1001 DBD120, 160, ..., 480)

Description	Active energy supplied by the auxiliary unit at the start of measurement in [kWh]
Signal flow	GP → operating software
Data type	REAL

ActiveEnAuxMeasOFF (DB1001 DBD124, 164, ..., 484)

Description	Active energy drawn by the auxiliary unit at the end of measurement in [kWh]
Signal flow	GP → operating software
Data type	REAL

ReActiveEnAuxMeasOFF (DB1001 DBD128, 168, ..., 488)

Description	Active energy supplied by the auxiliary unit at the end of measurement in [kWh]
Signal flow	GP → operating software
Data type	REAL

10.4.19 LBP_SpindleTempSensor [DB1002] - Spindle temperature sensors

10.4.19.1 Overview

LBP_SpindleTempSensor [DB1002], spindle temperature sensors

Table 10-50 Signals for **spindle temperature sensors**

LBP_SpindleTemp-Sensor [DB1002]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 0	Sensor n Sensor installation location							
DBWn + 2	Reserved							
DBDn + 4	Sensor n Temperature sensor actual value [°C]							
DBDn + 8	Sensor n Temperature sensor warning threshold value [°C]							
DBWn + 12	Sensor n Number of alarm limit value violations							
DBBn + 14	Sensor n Last alarm limit value violation: Year total active power (REAL) to the operating software							
DBBn + 15	Sensor n Last alarm limit value violation: Month							
DBBn + 16	Sensor n Last alarm limit value violation: Day							
DBBn + 17	Sensor n Last alarm limit value violation: Hour							
DBBn + 18	Sensor n Last alarm limit value violation: Minute							

LBP_SpindleTemp-Sensor [DB1002]	**SENTRON PAC**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBBn + 19	Sensor n Last alarm limit value violation: Seconds							
DBBn + 20	Sensor n Duration of the warning limit value violations							
DBBn + 24	Sensor n Temperature sensor fault threshold value [°C]							
DBBn + 28	Sensor n Number of fault limit value violations							
DBBn + 30	Sensor n Last alarm limit value violation: Year							
DBBn + 31	Sensor n Last alarm limit value violation: Month							
DBBn + 32	Sensor n Last alarm limit value violation: Day							
DBBn + 33	Sensor n Last alarm limit value violation: Hour							
DBBn + 34	Sensor n Last alarm limit value violation: Minute							
DBBn + 35	Sensor n Last alarm limit value violation: Seconds							
DBDn + 36	Sensor n Duration of the fault limit value violation							

Spindle/temperature sensor instances

Spindle_1, Temperature_sensor_1	(n=0):	DB1002 DBB00...DBB39
Spindle_1, Temperature_sensor_2	(n=40):	DB1002 DBB40...DBB79
Spindle_1, Temperature_sensor_3	(n=80):	DB1002 DBB80...DBB119

Spindle_1, Temperature_sensor_4	(n=120):	DB1002 DBB120...DBB159
Spindle_1, Temperature_sensor_5	(n=160):	DB1002 DBB160...DBB199
Spindle_1, Temperature_sensor_6	(n=200):	DB1002 DBB200...DBB239
Spindle_2, Temperature_sensor_1	(n=240):	DB1002 DBB240...DBB279
Spindle_2, Temperature_sensor_2	(n=280):	DB1002 DBB280...DBB319
Spindle_2, Temperature_sensor_3	(n=320):	DB1002 DBB320...DBB359
Spindle_2, Temperature_sensor_4	(n=360):	DB1002 DBB360...DBB399
Spindle_2, Temperature_sensor_5	(n=400):	DB1002 DBB400...DBB439
Spindle_2, Temperature_sensor_6	(n=440):	DB1002 DBB440...DBB479

10.4.19.2 LBP_SpindleTempSensor [DB1002]

The data block "LBP_SpindleTempSensor [DB1002]" contains signals for spindle temperature sensors.

The spindle/temperature sensor instances are assigned as follows:

Spindle_1, Temperature_sensor_1	"LBP_SpindleTempSensor.SP[1].TS[1]" (DB1002 from DBB0)
Spindle_1, Temperature_sensor_2	"LBP_SpindleTempSensor.SP[1].TS[2]" (DB1002 from DBB40)
Spindle_1, Temperature_sensor_3	"LBP_SpindleTempSensor.SP[1].TS[3]" (DB1002 from DBB80)
Spindle_1, Temperature_sensor_4	"LBP_SpindleTempSensor.SP[1].TS[4]" (DB1002 from DBB120)
Spindle_1, Temperature_sensor_5	"LBP_SpindleTempSensor.SP[1].TS[5]" (DB1002 from DBB160)
Spindle_1, Temperature_sensor_6	"LBP_SpindleTempSensor.SP[1].TS[6]" (DB1002 from DBB200)
Spindle_2, Temperature_sensor_1	"LBP_SpindleTempSensor.SP[2].TS[1]" (DB1002 from DBB240)
Spindle_2, Temperature_sensor_2	"LBP_SpindleTempSensor.SP[2].TS[2]" (DB1002 from DBB280)
Spindle_2, Temperature_sensor_3	"LBP_SpindleTempSensor.SP[2].TS[3]" (DB1002 from DBB320)
Spindle_2, Temperature_sensor_4	"LBP_SpindleTempSensor.SP[2].TS[4]" (DB1002 from DBB360)
Spindle_2, Temperature_sensor_5	"LBP_SpindleTempSensor.SP[2].TS[5]" (DB1002 from DBB400)
Spindle_2, Temperature_sensor_6	"LBP_SpindleTempSensor.SP[2].TS[6]" (DB1002 from DBB440)

Variables in "LBP_SpindleTempSensor [DB1002]"

SP (DB1002 from DBB0)

Description	Spindle 1 ... 2 The data sets of the spindles contain the variables of the temperature sensors.
Data type	ARRAY[1..2] of STRUCT STRUCT "SP" for spindle (Page 572)

10.4.19.3 STRUCT "SP" for spindle

The variable structure "SP" contains structures with temperature sensors of a spindle.

Variables in "SP"

TS (DB1002 from DBB0, from DBB240)

Description	Temperature sensors of a spindle
Data type	ARRAY[1..6] of STRUCT STRUCT "TS" for temperature sensors (Page 573)

10.4.19.4 STRUCT "TS" for temperature sensors

The variable structure "TS" describes temperature sensors of a spindle.

Variables in "TS"

MountPlace (DB1002 DBW0, 40, ..., 200; 240, 280, ..., 440)

Description	Sensor installation location
Data type	INT

ActVal (DB1002 DBD4, 44, ..., 204; 244, 284, ..., 444)

Description	Temperature sensor actual value [°C]
Data type	REAL

WarnThreshold (DB1002 DBD8, 48, ..., 208; 248, 288, ..., 448)

Description	Temperature sensor warning threshold value [°C]
Data type	REAL

WarnNumLimitViol (DB1002 DBW12, 52, ..., 212; 252, 292, ..., 452)

Description	Number of alarm limit value violations
Data type	INT

WarnTsViolYY (DB1002 DBB14, 54, ..., 214; 254, 294, ..., 454)

Description	Last alarm limit value violation: Year total active power (REAL) to the operating software
Data type	BYTE

WarnTsViolMM (DB1002 DBB15, 55, ..., 215; 255, 295, ..., 455)

Description	Last alarm limit value violation: Month
Data type	BYTE

WarnTsViolDD (DB1002 DBB16, 56, ..., 216; 256, 296, ..., 456)

Description	Last alarm limit value violation: Day
Data type	BYTE

WarnTsViolH (DB1002 DBB17, 57, ..., 217; 257, 297, ..., 457)

Description	Last alarm limit value violation: Hour
Data type	BYTE

WarnTsViolM (DB1002 DBB18, 58, ..., 218; 258, 298, ..., 458)

Description	Last alarm limit value violation: Minute
Data type	BYTE

WarnTsViolS (DB1002 DBB19, 59, ..., 219; 259, 299, ..., 459)

Description	Last alarm limit value violation: Seconds
Data type	BYTE

WarnDurLimitViol (DB1002 DBD20, 60, ..., 220; 260, 300, ..., 460)

Description	Duration of the warning limit value violations
Data type	DINT

FaultThreshold (DB1002 DBD24, 64, ..., 224; 264, 304, ..., 464)

Description	Temperature sensor fault threshold value [°C]
Data type	REAL

FaultNumLimitViol (DB1002 DBW28, 68, ..., 228; 268, 308, ..., 468)

Description	Number of fault limit value violations
Data type	INT

FaultTsViolYY (DB1002 DBB30, 70, ..., 230; 270, 310, ..., 470)

Description	Last alarm limit value violation: Year
Data type	BYTE

FaultTsViolMM (DB1002 DBB31, 71, ..., 231; 271, 311, ..., 471)

Description	Last alarm limit value violation: Month
Data type	BYTE

FaultTsViolDD (DB1002 DBB32, 72, ..., 232; 272, 312, ..., 472)

Description	Last alarm limit value violation: Day
Data type	BYTE

FaultTsViolH (DB1002 DBB33, 73, ..., 233; 273, 313, ..., 473)

Description	Last alarm limit value violation: Hour
Data type	BYTE

FaultTsViolM (DB1002 DBB34, 74, ..., 234; 274, 314, ..., 474)

Description	Last alarm limit value violation: Minute
Data type	BYTE

FaultTsViolS (DB1002 DBB35, 75, ..., 235; 275, 315, ..., 475)

Description	Last alarm limit value violation: Seconds
Data type	BYTE

FaultDurLimitViol (DB1002 DBD36, 76, ..., 236; 276, 316, ..., 476)

Description	Duration of the fault limit value violation
Data type	DINT

10.4.20 LBP_TMLoadMT [DB1071] - Loading/unloading magazine: Multitool

10.4.20.1 Overview

LBP_TMLoadMT [DB1071], interface for loading/unloading the magazine: Multitool

Table 10-51 Interface for loading/unloading magazine: Multitool

LBP_TMLoadMT [DB1071]	**Loading/unloading locations (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 0	Type of distance coding of the multitool (corresponds to $TC_MTP_KD)							
DBWn + 2	Number of multitool locations							
DBWn + 4	Multitool location distance							
DBWn + 8	Multitool number							
DBWn + 10	Multitool location number							
DBWn + 12	Tool holder							
DBWn + 14	Reserved							
DBWn + 16	Reserved							
DBWn + 18	Reserved							

10.4.20.2 LBP_TMLoadMT [DB1071], interface for loading/unloading the magazine: Multitool

Variables in "LBP_TMLoadMT [DB1071]"

MTIF (DB1071 from DBB0)

Description	Interface signals for loading/unloading the magazine Loading/unloading location 1 is responsible for loading/unloading in (all) spindles/tool holders and for relocating tools and for positioning at any locations (e.g. buffer location). Loading and unloading manual tools is always realized via loading/unloading location 1.
Signal flow	NC → PLC

Data type	ARRAY[1..16] of STRUCT STRUCT Load/unload MTIF for magazine Multitool (Page 576)
Update	Job-controlled

10.4.20.3 STRUCT Load/unload MTIF for magazine Multitool

The variable structure "MTIF" describes interface signals for loading/unloading multitools in the magazine.

Variables in "MTIF"

DistanceCoding (DB1071 DBW0, 20, ..., 300)

Description	Distance coding
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Type of distance coding of the multitool (corresponds to $TC_MTP_KD) 1 = location number 2 = distance 3 = angle

Places (DB1071 DBW2, 22, ..., 302)

Description	Number of multitool locations
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Number of locations of the multitool

Distance (DB1071 DBD4, 24, ..., 304)

Description	Multitool location distance
Signal flow	NC → PLC
Data type	REAL
Update	Job-controlled
Value	Distance of the MT location to be positioned from the reference location (real value), corresponding to the distance coding

MTNo (DB1071 DBW8, 28, ..., 308)

Description	Multitool number
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Internal T number of the multitool.

PlaceNo (DB1071 DBW10, 30, ..., 310)

Description	Multitool location number
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value	Location number within the multitool (which the system positions to)

Tool holder (DB1071 DBW12, 32, ..., 312)

Description	Tool holder
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled
Value	Spindle or tool holder number

10.4.21 LBP_TMSpindleMT [DB1072] - Spindle: Multitool

10.4.21.1 Overview

LBP_TMSpindleMT [DB1072], interface for the spindle: Multitool

Table 10-52 Interface for spindle: Multitool

LBP_TMSpindleMT [DB1072]	**Spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 0	Type of distance coding of the multitool (corresponds to $TC_MTP_KD)							
DBWn + 2	Number of multitool locations							
DBWn + 4	Multitool location distance							
DBWn + 8	Multitool number (new tool)							
DBWn + 10	Multitool location number (new tool)							
DBWn + 12	Multitool number (old tool)							
DBWn + 14	Multitool location number (old tool)							
DBWn + 16	Tool new: Location type							
DBWn + 18	Tool new: Size left							
DBWn + 20	Tool new: Size right							
DBWn + 22	Tool new: Size top							
DBWn + 24	Tool new: Size bottom							
DBWn + 26	Tool status for new tool (corresponds to $TC_TP8[T_Nr])							
DBWn + 28	Tool new: Internal T number of the NC							
DBWn + 30	Tool holder							

LBP_TMSpindleMT [DB1072]	**Spindle (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 32	Original magazine of new tool (corresponds to NC variables $A_MYMN[T No])							
DBWn + 34	Original location of new tool (corresponds to NC variables $A_MYMLN[T No])							
DBWn + 36 - DBWn + 48	Reserved							

10.4.21.2 LBP_TMSpindleMT [DB1072], interface for the spindle: Multitool

Variables in "LBP_TMSpindleMT [DB1072]"

MTIF (DB1072 from DBB0)

Description	Interface signals for tool change on spindles: Spindle 1: Index 1 Spindle 2: Index 2 ... Spindle 16: Index 16
Signal flow	NC → PLC
Data type	ARRAY[1..16] of STRUCT STRUCT Load/unload MTIF for spindle: Multitool (Page 578)
Update	Job-controlled

10.4.21.3 STRUCT Load/unload MTIF for spindle: Multitool

The variable structure "MTIF" describes interface signals for loading/unloading multitools in the spindle.

Variables in "MTIF"

DistanceCoding (DB1072 DBW0, 50, ..., 750)

Description	Distance coding
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Type of distance coding of the multitool (corresponds to $TC_MTP_KD) 1 = location number 2 = distance 3 = angle

Places (DB1072 DBW2, 52, ..., 752)

Description	Number of multitool locations
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Number of locations of the multitool

Distance (DB1072 DBD4, 54, ..., 754)

Description	Multitool location distance
Signal flow	NC → PLC
Data type	REAL
Update	Job-controlled
Value	Distance of the MT location to be positioned from the reference location (real value), corresponding to the distance coding

NewMTNo (DB1072 DBW8, 58, ..., 758)

Description	Multitool number (new tool) Start addresses of the spindles: Spindle 1: n = 0 Spindle 2: n = 50 Spindle 3: n = 100
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value FALSE	No internal T number specified.
Value TRUE	Internal T number of the new multitool.

NewPlaceNo (DB1072 DBW10, 60, ..., 760)

Description	Multitool location number
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location number within the multitool (where the new tool is located).

OldMTNo (DB1072 DBW12, 62, ..., 762)

Description	Multitool number (new tool)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Internal T number of the (old) multitool. The T number is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder). It is identical to "MTIF.NewMTNo".

OldPlace (DB1072 DBW14, 64, ..., 764)

Description	Multitool location number (old tool)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location number within the multitool (where the old tool is located). The location number - where the old tool is located - is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder).

LocType (DB1072 DBW16, 66, ..., 766)

Description	Tool new: Location type
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location type of the new tool (of the programmed tool in the multitool).
Corresponds to	"MTIF.SLeft" "MTIF.SRight" "MTIF.STop" "MTIF.Bottom"

SLeft (DB1072 DBW18, 68, ..., 768)

Description	Tool new: Size left
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **left** in half locations for the new spindle tool.
Corresponds to	"MTIF.LocType" "MTIF.SRight" "MTIF.STop" "MTIF.Bottom"

SRight (DB1072 DBW20, 70, ..., 770)

Description	Tool new: Size right
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size to the **right** in half locations of the new tool (of the programmed tool in the multitool)
Corresponds to	"MTIF.LocType" "MTIF.SLeft" "MTIF.STop" "MTIF.Bottom"

STop (DB1072 DBW22, 72, ..., 772)

Description	Tool new: Size top
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size to the **top** in half locations of the new tool (of the programmed tool in the multitool)
Corresponds to	"MTIF.LocType" "MTIF.SLeft" "MTIF.SRight" "MTIF.Bottom"

SBottom (DB1072 DBW24, 74, ..., 774)

Description	Tool new: Size bottom
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size to the **bottom** in half locations of the new tool (of the programmed tool in the multitool)
Corresponds to	"MTIF.LocType" "MTIF.SLeft" "MTIF.SRight" "MTIF.Top"

TInBuffer, IgnoreDisabled, Unload, Load, MasterTool, Exchange_1to1, HandTool, Active, Enabled, Measured, Prewarn, Change, FixedLoc, Used

(DB1072 from DBX26.0, 76.0, ..., 776.0)

Description	Tool status of the new tool
Signal flow	NC → PLC
Data type	BOOL
Update	Job-controlled

Status bits	TInBuffer (DBX26.0)	Tool in the buffer magazine
	IgnoreDisabled	Disabled, but ignore
	Unload	Tool to be unloaded
	Load	Tool to be loaded
	MasterTool	Master tool
	Exchange_1to1	1:1 exchange
	HandTool	Manual tool
	Active	Active tool
	Enabled	Enable tool
	Disabled	Tool disabled
	Measured	Measure tool
	Prewarning	Prewarning limit reached
	Change	Tool is being changed
	FixedLoc	Tool is fixed-location-coded
	Used (DBX27.7)	Tool was in use

TNo (DB1072 DBW28, 78, ..., 778)

Description	Tool new: Internal T number of the NC
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Display the internal T number of the NC for the new spindle tool (of the programmed tool in the multitool).

Tool holder (DB1072 DBW30, 80, ..., 780)

Description	Tool holder
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Spindle or tool holder number to which the change refers (the multitool that is to be changed).

MyMag (DB1072 DBW32, 82, ..., 782)

Description	Original magazine of the new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Owner magazine of the new tool Corresponds to NC variables $A_MYMN[T no.] If the new tool is located in the magazine, then this value is identical with "MTIF.BufferIdent". If the new tool is in the buffer magazine (e.g. gripper), the magazine no. of the magazine from which the tool originally came is entered here.

MyPlace (DB1072 DBW34, 84, ..., 784)

Description	Original location of new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location that owns the new tool. Corresponds to NC variables $A_MYMLNT[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer magazine (e.g. gripper), the magazine no. of the magazine from which the tool originally came is entered here. The new tool has the same owner location as the multitool onto which it is loaded.

10.4.22 LBP_TMTurretMT [DB1073] - Turret: Multitool

10.4.22.1 Overview

LBP_TMTurretMT [DB1073], interface for the turret: Multitool

Table 10-53 Interface for the turret: Multitool

LBP_TMTurretMT [DB1073]	**Turret (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 0	Type of distance coding of the multitool (corresponds to $TC_MTP_KD)							
DBWn + 2	Number of multitool locations							
DBWn + 4	Multitool location distance							
DBWn + 8	Multitool number (new tool)							
DBWn + 10	Multitool location number (new tool)							
DBWn + 12	Multitool number (old tool)							
DBWn + 14	Multitool location number (old tool)							
DBWn + 16	Location type							
DBWn + 18	Tool new: Size left							
DBWn + 20	Tool new: Size right							
DBWn + 22	Tool new: Size top							
DBWn + 24	Tool new: Size bottom							
DBWn + 26	Tool status for new tool (corresponds to parameter $TC_TP8[T_No])							
DBWn + 28	Tool new: Internal T number of the NC							
DBWn + 30	Tool holder							
DBWn + 32	Original magazine of new tool (corresponds to NC variables $A_MYMN[T No])							

LBP_TMTurretMT [DB1073]	**Turret (NC → PLC)**							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
DBWn + 34	Original location of new tool (corresponds to NC variables $A_MYMN[T No])							
DBWn + 36 - DBWn + 48	Reserved							

10.4.22.2 LBP_TMTurretMT [DB1073], interface for the turret: Multitool

Variables in "LBP_TMTurretMT [DB1073]"

MTIF (DB1073 from DBB0)

Description	Interface signals for tool change on spindles: Resolver 1: Index 1 Resolver 2: Index 2 ... Resolver 16: Index 16
Signal flow	NC → PLC
Data type	ARRAY[1..16] of STRUCT STRUCT MTIF for turret: Multitool (Page 584)
Update	Job-controlled

10.4.22.3 STRUCT MTIF for turret: Multitool

The variable structure "MTIF" describes the interface signals for a turret with multitools.

Variables in "MTIF"

DistanceCoding (DB1073 DBW0, 50, ..., 750)

Description	Distance coding
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Type of distance coding of the multitool (corresponds to $TC_MTP_KD) 1 = location number 2 = distance 3 = angle

Places (DB1073 DBW2, 52, ..., 752)

Description	Number of multitool locations
Signal flow	NC → PLC
Data type	INT

Update	Job-controlled
Value	Number of locations of the multitool
Edge evaluation	Signal(s) updated: Conditional

Distance (DB1073 DBD4, 54, ..., 754)

Description	Multitool location distance
Signal flow	NC → PLC
Data type	REAL
Update	Job-controlled
Value	Distance of the MT location to be positioned from the reference location (real value), corresponding to the distance coding

NewMTNo (DB1073 DBW8, 58, ..., 758)

Description	Multitool number (new tool)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Internal T number of the new multitool.

NewPlace (DB1073 DBW10, 60, ..., 760)

Description	Multitool location number (new tool)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location number within the multitool (where the new tool is located).

OldMTNo (DB1073 DBW12, 62, ..., 762)

Description	Multitool number (new tool)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Internal T number of the (new) multitool. The T number is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder). It is identical to "MTIF.NewMTNo".

OldPlace (DB1073 DBW14, 64, ..., 764)

Description	Multitool location number (old tool)
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location number within the multitool (where the old tool is located). The location number - where the old tool is located - is entered here if the preparation or the change to a tool is realized within the same multitool (which, due to a previous change, is located on the tool holder).

LocType (DB1073 DBW16, 66, ..., 766)

Description	Tool new: Location type
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location type of the new tool (of the programmed tool in the multitool).
Corresponds to	"MTIF.SLeft" "MTIF.SRight" "MTIF.STop" "MTIF.Bottom"

SLeft (DB1073 DBW18, 68, ..., 768)

Description	Tool new: Size left
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size **left** in half locations for the new spindle tool.
Corresponds to	"MTIF.LocType" "MTIF.SRight" "MTIF.STop" "MTIF.Bottom"

SRight (DB1073 DBW20, 70, ..., 770)

Description	Tool new: Size right
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size to the **right** in half locations of the new tool (of the programmed tool in the multitool)
Corresponds to	"MTIF.LocType" "MTIF.SLeft" "MTIF.STop" "MTIF.Bottom"

STop (DB1073 DBW22, 72, ..., 772)

Description	Tool new: Size top
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled

Value	Specification of the tool size to the **top** in half locations of the new tool (of the programmed tool in the multitool)
Corresponds to	"MTIF.LocType" "MTIF.SLeft" "MTIF.SRight" "MTIF.Bottom"

SBottom (DB1073 DBW24, 74, ..., 774)

Description	Tool new: Size bottom
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Specification of the tool size to the **bottom** in half locations of the new tool (of the programmed tool in the multitool)
Corresponds to	"MTIF.LocType" "MTIF.SLeft" "MTIF.SRight" "MTIF.Top"

TInBuffer, IgnoreDisabled, Unload, Load, MasterTool, Exchange_1to1, HandTool, Active, Enabled, Measured, Prewarn, Change, FixedLoc, Used

(DB1073 from DBX.26.0, 76.0, ..., 776.0)

Description	Tool status of the new tool	
Signal flow	NC → PLC	
Data type	BOOL	
Update	Job-controlled	
Status bits	TInBuffer (DBX26.0)	Tool in the buffer magazine
	IgnoreDisabled	Disabled, but ignore
	Unload	Tool to be unloaded
	Load	Tool to be loaded
	MasterTool	Master tool
	Exchange_1to1	1:1 exchange
	HandTool	Manual tool
	Active	Active tool
	Enabled	Enable tool
	Disabled	Tool disabled
	Measured	Measure tool
	Prewarning	Prewarning limit reached
	Change	Tool is being changed
	FixedLoc	Tool is fixed-location-coded
	Used (DBX27.7)	Tool was in use

TNo (DB1073 DBW28, 78, ..., 778)

Description	Tool new: Internal T number of the NC
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Display the internal T number of the NC for the new spindle tool (of the programmed tool in the multitool).

Tool holder (DB1073 DBW30, 80, ..., 780)

Description	Tool holder
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Spindle or tool holder number to which the change refers (the multitool that is to be changed).

MyMag (DB1073 DBW32, 82, ..., 782)

Description	Original magazine of the new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Owner magazine of the new tool Corresponds to NC variables $A_MYMN[T no.] If the new tool is located in the magazine, then this value is identical with "MTIF.BufferIdent". If the new tool is in the buffer magazine (e.g. gripper), the magazine no. of the magazine from which the tool originally came is entered here.

MyPlace (DB1073 DBW34, 84, ..., 784)

Description	Original location of new tool
Signal flow	NC → PLC
Data type	INT
Update	Job-controlled
Value	Location that owns the new tool. Corresponds to NC variables $A_MYMLNT[T no.] If the new tool is located in the magazine, then this value is identical with DB72 DBW(n+20). If the new tool is in the buffer magazine (e.g. gripper), the magazine no. of the magazine from which the tool originally came is entered here. The new tool has the same owner location as the multitool onto which it is loaded.

10.5 Input and output signals

10.5.1 General

The basic program library provides PLC data types (UDT) for the machine control panel. Use this for symbolic programming.

See also

LBP_typeMcp* (Page 95)

10.5.2 Input signals

10.5.2.1 M version, signals from the MCP: Input image

Table 10-54 M version, signals from the MCP: Input image

	Signals from the MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Spindle override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDI	AUTO
IB n + 1	Machine function							
	REPOS	REF	INCvar	INC10000	INC1000	INC100	INC10	INC1
IB n + 2	Key-operated switch position 0	Key-operated switch position 2	Spindle start	*Spindle stop	Feed start	*Feedrate stop	NC start	*NC stop
IB n + 3		Key-operated switch position 1		Feedrate override				
	Reset		Single block	E	D	C	B	A
IB n + 4	Direction keys			Key-operated switch position 3	Axis selection			
	+R15	-R13	Rapid traverse R14		X R1	4th axis R4	7th axis R7	R10
IB n + 5	Axis selection							
	Y R2	Z R3	5th axis R5	Travel command MCS/WCS	R11	9th axis R9	8th axis R8	6th axis R6
IB n + 6	Freely assignable customer keys							
	T9	T10	T11	T12	T13	T14	T15	
IB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

10.5.2.2 T version, signals from the MCP: Input image

Table 10-55 T version, signals from the MCP: Input image

	Signals from the MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Spindle override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDI	AUTO
IB n + 1	Machine function							
	REPOS	REF	INCvar	INC10000	INC1000	INC100	INC10	INC1
IB n + 2	Key-operated switch position 0	Key-operated switch position 2	Spindle start	*Spindle stop	Feed start	*Feedrate stop	NC start	*NC stop
IB n + 3		Key-operated switch position 1		Feedrate override				
	Reset		Single block	E	D	C	B	A
IB n + 4	Direction keys			Key-operated switch position 3	Direction keys			
	R15	R13	R14		+Y R1	-Z R4	-C R7	R10
IB n + 5	Direction keys							
	+X R2	+C R3	Rapid traverse override R5	Travel command MCS/ WCS R12	R11	-Y R9	-X R8	+Z R6
IB n + 6	Freely assignable customer keys							
	T9	T10	T11	T12	T13	T14	T15	
IB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

10.5.2.3 Slimline version, signals from the MCP: Input image

Table 10-56 Slimline version, signals from the MCP: Input image

	Signals from the slimline MCP (switches and keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Spindle override				Operating mode			
	*NC stop	SP -	SP 100%	SP +	SINGLEB	JOG	MDI	AUTO
IB n + 1	Spindle				Key-operated switch	Machine function		
	NC start	SP CW	*SP stop	SP CCW	IF 3	REF	REPOS	TEACH IN
IB n + 2	Feedrate			Key-operated switch	Machine functions			
	Start	*Stop	INCvar	IF 0	INC1000	INC100	INC10	INC1
IB n + 3		Key-operated switch		Feedrate override				
	Reset	IF 2	IF 1	E	D	C	B	A

	Signals from the slimline MCP (switches and keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 4	Direction keys			Optional customer keys				
	+R15	-R13	Rapid traverse R14	KT4	KT3	KT2	KT1	KT0
IB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
IB n + 6	Freely assignable customer keys				MCS/WCS	Freely assignable customer keys		
	T9	T10	T11	T12		T14	T15	T16
IB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

10.5.2.4 Signals from the handheld unit: Input image

Table 10-57 Signals from the handheld unit: Input image

	Signals from the handheld unit (keys) (HT 2 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB m + 0	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
IB m + 1	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
IB m + 2	Feed start	Free T2 key	AUTO	NC stop	Spindle stop	Feed stop	Free T1 key	JOG
IB m + 3	Free T3 key	Handwheel	4th axis	Z	Y	X	NC start	Spindle start
IB m + 4	Direction key	Rapid traverse override	Direction key +	Free T4 key				
IB m + 5	Acknowledgement Digital display	Key-operated switch	Rapid traverse/feedrate override switch					
			E	D	C	B	A	

10.5.2.5 Signals from the handheld unit HT 8: Input image

Table 10-58 Signals from the handheld unit HT 8: Input image

	Signals from the MCP simulation (HT 8 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	WCS/MCS
IB n + 1	Function key block							
	CPF (U key)	U4	U3	BigFct	U2	U1	INC	REPOS

	Signals from the MCP simulation (HT 8 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 2		Change-over axes (HMI Advanced only)	Traversing keys (JOG) positive direction					
		Ax7-Ax12 instead of Ax1-Ax6	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 3	Traversing keys (JOG) negative direction							
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 4								
	U16	U15	U14	U13	U12	U11	U10	U9
IB n + 5								
		U8	U7	U6	U5	SBL		
IB n + 6			Start key block					
	Reserved	HT 8	SF2	SF1	SF4	SF3	Start	Stop
IB n + 7	Feedrate override							
				E	D	C	B	A

10.5.2.6 Signals from the handheld unit HT 10: Input image

Table 10-59 Signals from the handheld unit HT 10: Input image

	Signals from the MCP simulation (HT 10 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	WCS/MCS
IB n + 1	Function key block							
	CPF (U key)	U4	U3	BigFct	U2	U1	INC	REPOS
IB n + 2		Change-over axes (HMI Advanced only)	Traversing keys (JOG) HT 8, positive direction					
		Ax7-Ax12 instead of Ax1-Ax6	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 3	JOG keys HT 10		Traversing keys (JOG) HT 8, negative direction					
	Plus	Minus	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
IB n + 4								
	U16	U15	U14	U13	U12	U11	U10	U9
IB n + 5								
		U8	U7	U6	U5	SBL		Rapid traverse

	Signals from the MCP simulation (HT 10 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 6			Start key block					
	HT 10	HT 8	SF2	SF1	SF4	SF3	Start	Stop
IB n + 7	Feedrate override							
				E	D	C	B	A

10.5.3 Output signals

10.5.3.1 M version, signals to the MCP: Output image

Table 10-60 M version, signals to the MCP: Output image

	Signals to the MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Machine function				Operating mode			
	INC1000	INC100	INC10	INC1	JOG	TEACH IN	MDI	AUTO
QB n + 1	Feed start	*Feedrate stop	NC start	*NC stop	Machine function			
					REPOS	REF	INCvar	INC10000
QB n + 2	Direction key -R13	Axis selection				Single block	Spindle start	*Spindle stop
		X R1	4th axis R4	7th axis R7	R10			
QB n + 3	Axis selection							Direction key +R15
	Z R3	5th axis R5	Travel command MCS/WCS R12	R11	9th axis R9	8th axis R8	6th axis R6	
QB n + 4	Freely assignable customer keys							Y R2
	T9	T10	T11	T12	T13	T14	T15	
QB n + 5	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

10.5.3.2 T version, signals to the MCP: Output image

Table 10-61 T version, signals to the MCP: Output image

	Signals to the MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Machine function				Operating mode			
	INC1000	INC100	INC10	INC1	JOG	TEACH IN	MDI	AUTO
QB n + 1	Feed start	*Feed stop	NC start	*NC stop	Machine function			
					REPOS	REF	INCvar	INC10000

	Signals to the MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 2	Direction keys					Single block	Spindle start	*Spindle stop
	R13	+Y R1	-Z R4	-C R7	R10			
QB n + 3	Direction keys							
	R3	R5	Travel command MCS/WCS	R11	-Y R9	-X R8	+Z R6	R15
QB n + 4	Freely assignable customer keys							Direction key +X R2
	T9	T10	T11	T12	T13	T14	T15	
QB n + 5	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

10.5.3.3 Slimline version, signals to the MCP: Output image

Table 10-62 Slimline version, signals to the MCP: Output image

	Signals to the slimline MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
IB n + 0	Spindle override				Operating mode			
	NC stop	SP -	SP 100%	SP +	SINGLEB	JOG	MDI	AUTO
IB n + 1	Spindle					Machine function		
	NC start	SP CW	SP stop	SP CCW	Not assigned	REF	REPOS	TEACH IN
IB n + 2	Feedrate				Machine functions			
	Start	Stop	INCvar	Not assigned	INC1000	INC100	INC10	INC1
IB n + 3	Not assigned							
IB n + 4	Direction keys			Optional customer keys				
	+R15	-R13	Rapid traverse R14	KT4	KT3	KT2	KT1	KT0
IB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
IB n + 6	Freely assignable customer keys					Freely assignable customer keys		
	T9	T10	T11	T12	MCS/WCS	T14	T15	T16
IB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

10.5.3.4 Signals to the handheld unit: Output image

Table 10-63 Signals to the handheld unit: Output image

	Signals to the handheld unit (LED) (PLC → HT 2)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB m + 0	Always 1				Free T4 key	Free T3 key	Free T2 key	Free T1 key
QB m + 1	New data for selected line						Selecting the line	
							3 and 4	1 and 2
QB m + 2	Feed start	Rapid traverse override	AUTO	NC stop	Spindle stop	Feed stop	Direction key +	JOG
QB m + 3	Direction key +	Handwheel	4th axis	Z	Y	X	NC start	Spindle start
	Digital display of the handheld unit							
QB m + 4	Default setting of 1st character (right) of selected line							
QB m + 5	Default setting of 2nd character of selected line							
QB m + 6	Default setting of 3rd character of selected line							
QB m + 7	Default setting of 4th character of selected line							
QB m + 8	Default setting of 5th character of selected line							
QB m + 9	Default setting of 6th character of selected line							
QB m + 10	Default setting of 7th character of selected line							
QB m + 11	Default setting of 8th character of selected line							
QB m + 12	Default setting of 9th character of selected line							
QB m + 13	Default setting of 10th character of selected line							
QB m + 14	Default setting of 11th character of selected line							
QB m + 15	Default setting of 12th character of selected line							
QB m + 16	Default setting of 13th character of selected line							
QB m + 17	Default setting of 14th character of selected line							

	Signals to the handheld unit (LED) (PLC → HT 2)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB m + 18	Default setting of 15th character of selected line							
QB m + 19	Default setting of 16th character (left) of selected line							

10.5.3.5 Signals to handheld terminal HT 8: Output image

Table 10-64 Signals to handheld terminal HT 8: Output image

	Signals to the MCP simulation (PLC → HT 8)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	WCS/MCS
QB n + 1	Function key block							
		U4	U3		U2	U1	INC	REPOS
QB n + 2		Axes 7 - n selected	Traversing keys (JOG) positive direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 3	Traversing keys (JOG) negative direction							
	For WCS: no machine axes		Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
QB n + 5								
		U8	U7	U6	U5	SBL		
QB n + 6	Start key block							
	Display traversing keys		SF2	SF1	SF4	SF3	Start	Stop
QB n + 7								

10.5.3.6 Signals to handheld terminal HT 10: Output image

Table 10-65 Signals to handheld terminal HT 10: Output image

	Signals to the MCP simulation (PLC → HT 10)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	QUIT	Reset	WCS/MCS
QB n + 1	Function key block							
		U4	U3		U2	U1	INC	REPOS

	Signals to the MCP simulation (PLC → HT 10)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
QB n + 2	JOG+ (HT 10)	Axes 7 - n selected	Traversing keys (JOG) HT 8, positive direction					
			Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 3	Traversing keys (JOG) HT 8, negative direction							
	For WCS: no machine axes	JOG- (HT 10)	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
QB n + 4								
	U9	U10	U11	U12	U13	U14	U15	U16
QB n + 5								
		U8	U7	U6	U5	SBL		Rapid traverse
QB n + 6	Start key block							
	Display traversing keys		SF2	SF1	SF4	SF3	Start	Stop
QB n + 7								

10.6 Function blocks (FB)

10.6.1 LBP_ReadVar [FB2]: Read NC variable

10.6.1.1 LBP_ReadVar [FB2]: Read NC variable

Function

The function block "LBP_ReadVar [FB2]" reads variables from the NC area.

Request for reading NC variables

Call of the "LBP_ReadVar [FB2]" with positive edge: Parameter "Req" = 0 → 1

S7 names of the NC variables: Parameter "Addr1" to "Addr8" = <DB name>.<S7 name>. Unused addresses must be assigned "NULL".

Completion of the read request

Read request successfully completed: Parameter "NDR" == 1.

Read request completed with error: Parameter "Error" == 1, error cause in parameter "State"

Requirement

A data block for access to the NC variables has been generated, e.g. "ncvar [DB120]". See also Generating a data block for accessing NC variables (Page 604).

In this data block, the NC variable is specified in the parameters "Addr<x>", e.g. Addr1: = "ncvar".<NC-Variable>

General conditions

- "LBP_ReadVar [FB2]" has multi-instance capability.
  Every call of "LBP_ReadVar [FB2]" must be assigned a separate instance or an instance DB from the user area.
- When **channel-specific** variables are read, in a "LBP_ReadVar [FB2]" call no variables from different channels may be addressed via "Addr1" to "Addr8".

Note

Error case
When variables from different channels are read, an error message is output:

- "Error" == TRUE
- "State" == W#16#02

Variable addressing

For some NC variables, it is necessary to select "Area no." and/or "Line" or "Column" in the NC VAR selector.

It is possible to select a basic type, i.e. "Area no.", "Line" and "Column" are preassigned "0". The values of the "Area no.", "Line" and "Column" specified by the NC VAR selector are checked for a "0" in "LBP_ReadVar [FB2]". If an NC VAR selector value == "0", the corresponding value of the parameter is adopted. For this, the parameters "Unit<n>", "Column<n>", and "Line<n>" (<n> = 1, 2, ... - 8) must be written before "LBP_ReadVar [FB2]" is called.

The following example illustrates the use of "Area no.", "Line" and "Column" using the R-parameters. 100 R-parameters are available for each channel. If you want to read several of these in its PLC program or it is not clear at the time of programming which R-parameters are actually to be read during runtime, it can be very time-consuming to generate an entry for referencing for each R-parameter with the aid of the NC Var selector. Instead, only one entry is created for the variable R-parameter "Basic type".

If in the NC VAR selector, "1" is set as "Area no." and "0" specified as "Line", an entry is created for all R-parameters of channel 1. The "Line" parameter specifies which R-parameter is to be read when "LBP_ReadVar [FB2]" is called.

If in the NC VAR selector, "0" is specified for "Area no." and for "Line", an entry is created for all R-parameters in all channels. The "Unit" parameter specifies the channel and the "Line" parameter specifies the index of the R-parameter to be read when "LBP_ReadVar [FB2]" is called. Note that the value Index +1 must be specified as "Line" parameter:

R-parameter R[0] from channel 1: Unit = 1, Line = 1

R-parameter R[10] from channel 2: Unit = 2, Line = 11

Further explanations can also be found in the online help of the NC VAR selector.

Table 10-66 Parameter match

"LBP_ReadVar [FB2]" parameter	NC VAR selector
Unit	Area no.
Column	Column
Line	Line

Variables from different areas can be combined as follows in a call:

Table 10-67 Call combinations

Call	Area				
1	C[1]	N	B	A	T
2	C[2]	N	B	A	T
The same rules apply for channels 3 to 10 as for call 1 and call 2 shown in the example: A = axis-specific data, area no. corresponds to the axis number B = mode group data, area no. corresponds to the mode group number C = channel data, area no. corresponds to the channel number N = NC data, area no. is irrelevant (cannot be changed in the NC VAR selector) T = tool management data, area no. corresponds to the TO area					

Note

The number of usable variables can be less than eight, depending on the string length, when simultaneously reading several variables of the "String" type. This is because maximum 230 bytes can be transferred with one call. Nothing is read and the following error message is output:

- "Error" == TRUE
- "State" == W#16#03

Description of formal parameters

Parameter	Cate-gory	Type	Value range	Meaning
Req	I	BOOL	---	Job start with positive signal edge
NumVar	I	INT	1 ... 8	Number of variables to be read: "Addr1" - "Addr8"
Addr1 Addr2 ... Addr8	I	VARIANT	<DB name>. <S7 name>	Variable identifiers from **NC Var selector**
Unit1 Unit2 ... Unit8	I	BYTE	---	Area address, optional for variable addressing

Parameter	Category	Type	Value range	Meaning
Column1 Column2 ... Column8	I	WORD	---	Column address, optional for variable addressing
Line1 Line2 ... Line8	I	WORD	---	Line address, optional for variable addressing
Error	O	BOOL	---	TRUE: Negative acknowledgement of job or job could not be executed
NDR	O	BOOL	---	TRUE: Job successfully executed. Data is available
State	O	WORD	---	See paragraph "Error codes"
RD1 RD2 ... RD8	I/O	VARIANT	---	Target area for read data, e.g. DB area or bit memory of the same data type as the variable to be read

Error codes

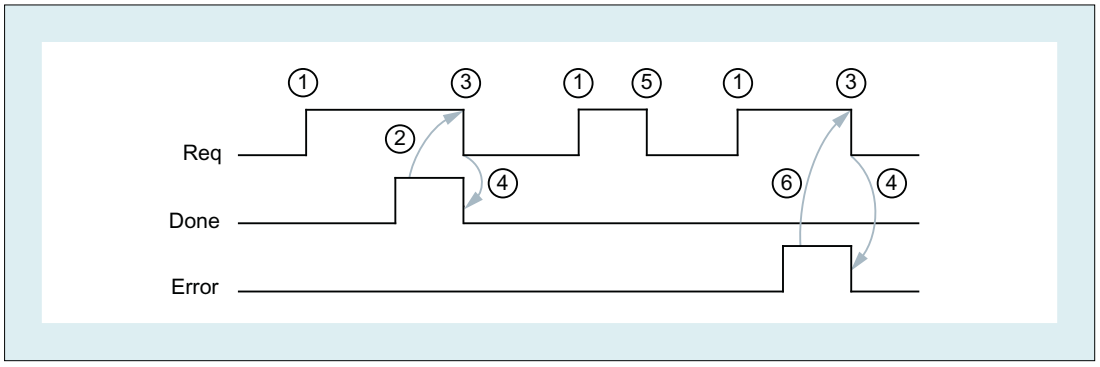
NC variables

State		Meaning	Note
High byte [1]	Low byte		
1 - 8	1	Access error	The variable cannot be accessed. Possible causes: • Variable does not exist in the NCK (possibly machine data must first be set) • The "Line" parameter is wrong (0 is only allowed for certain variables, for Word variables the parameter must be even)
0	2	Error in job	Incorrect compilation of variables in a job
0	3	Negative acknowledgement, job not executable	Internal error, possible remedy: • Check job data • NC reset • Fewer string variables in one call
1 - 8	4	Insufficient local user memory available	Data type of the read variable is greater than specified in "RD1" - "RD8"
1 - 8	5	Format conversion error	Error on conversion of var. type
0	6	FIFO full	Job must be repeated since queue is full
1 - 8	10	Error in variable addressing	"Unit" or "Column"/"Line" contains value 0
0	11	Address of variable invalid	Check "Addr" (or variable name), "Area", "Unit"
0	12	NumVar == 0	Check parameter "NumVar"
1) High byte > 0 ⇒ Number of the variable in which the error occurred			

Proceed as follows to read NC variables:

- Select variables with the NC VAR selector
- Save selected variables in a *.VAR file.
- Create the STL source
- Create a data block with the associated address data by integrating the STL source as external source in the TIA Portal project
- The name of the data block can be changed by the user as well as the S7 name contained therein
- Parameterization of "LBP_ReadVar [FB2]"

Pulse diagram



① User: Set request: Req = 0 → 1
② "LBP_ReadVar [FB2]": Reading successfully completed: NDR == 1
③ User: IF NDR == 1 THEN reset request: Req = 1 → 0
④ "LBP_ReadVar [FB2]": Reset job confirmation: NDR == 0
⑤ User: IF NDR == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ "LBP_ReadVar [FB2]": Reading completed with error: Error == 1
User: Reset request, IF Error == 1 THEN Req = 0, possible further error handling according to status

Call example

Read three machine data whose address data is stored in a "ncvar [DB120]".

Specification of the data

The data are selected with the "NC-Var-Selector" tool and the "ncvar [DB120]" data block is generated from it. See also Generating a data block for accessing NC variables (Page 604).

Area	Block	Name	Type	No.	Byte	S7 symbol
C[1]	M	MD20070 $MC_AXCONF_MACHAX_USED[1]	CHAR	20070	1	C1_M_AXCONF_MA-CHAX_USED1
C[1]	M	MD20070 $MC_AXCONF_MACHAX_USED[2]	CHAR	20070	1	C1_M_AXCONF_MA-CHAX_USED2
N	M	MD10000 $MN_AXCONF_MA-CHAX_NAME_TAB[1]	STRING	10000	1	N_M_AXCONF_MA-CHAX_NAME_1

Data block "ncvar" must be transferred to the PLC. The read values are written here to the "DB_Read", whereby this must be done for the correct type. This means that a CHAR value to be read must be written to a data area of the CHAR type in "DB_Read".

Parameterization of "LBP_ReadVar [FB2]" with instance DB “LBP_ReadVar_DB”:

```
// Test "LBP_ReadVar"
// 2x char, 1x string
"LBP_ReadVar_DB"(Req :="Tag_1",          // Bool
            NumVar := 3,
            Addr1 := "ncvar".C1_M_AXCONF_MACHAX_USED1,
            Addr2 := "ncvar".C1_M_AXCONF_MACHAX_USED2,
            Addr3 := "ncvar".N_M_AXCONF_MACHAX_NAME_1,
            Addr4 := NULL,
            Addr5 := NULL,
            Addr6 := NULL,
            Addr7 := NULL,
            Addr8 := NULL,
            Error => "Tag_2",            // Bool
            NDR => "Tag_3",              // Bool
            State => "Tag_4",            // Word
            RD1 := "DB_Read".CharValue[0],
            RD2 := "DB_Read".CharValue[1],
            RD3 := "DB_Read".StringValue,
            RD4 := NULL,
            RD5 := NULL,
            RD6 := NULL,
            RD7 := NULL,
            RD8 := NULL;
```

In this SCL example, "LBP_ReadVar [FB2]" is activated by setting Tag_1.

The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0.

The read values are written with the correct type to the "DB_Read".

Example: Variable addressing

Reading of two R-parameters from channel 1, whose address data are stored in the data block "ncvar" as "Basic type".

The R-parameter number is parameterized via parameter "Line<x>".

The R-parameter definition in data block "ncvar" is created by the "NC VAR selector":

```
C1_RP_rpa0_1:
   STRUCT
   SYNTAX_ID :           BYTE := B#16#82;
   area_and_unit :       BYTE := B#16#41;
   column :              WORD := W#16#1;
   line :                WORD := W#16#0;         // variable addressing
   block type :          BYTE := B#16#15;
   NO. OF LINES :        BYTE := B#16#1;
   type :                BYTE := B#16#F;
   length :              BYTE := B#16#8;
   END_STRUCT;


// Test "LBP_ReadVar"
// R Parameter
   "LBP_ReadVar_DB"(Req := "Tag_1",            // Bool
               NumVar := 2,
               Addr1 := "ncvar".C1_RP_rpa0_0,
               Line1 := 1                      // Reading R0
               Addr2 := "ncvar".C1_RP_rpa0_1,
               Line3 := 2                      // Reading R1
               Addr3 := NULL,
               Addr4 := NULL,
               Addr5 := NULL,
               Addr6 := NULL,
               Addr7 := NULL,
               Addr8 := NULL,
               Error => "Tag_2",               // Bool
               NDR => "Tag_3",                 // Bool
               State => "Tag_4",               // Word
               RD1 := "DB_Read".RealValue[0],
               RD2 := "DB_Read".RealValue[1],
               RD3 := NULL,
               RD4 := NULL,
               RD5 := NULL,
               RD6 := NULL,
               RD7 := NULL,
               RD8 := NULL;
```

In this SCL example, "LBP_ReadVar [FB2]" is activated by setting "Tag_1". The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0. The read values are written with the correct type to "DB_Read".

Classification of data types

NC data type	S7 data type
DOUBLE	LREAL
FLOAT	REAL
LONG	DINT
INTEGER	DINT
UINT_32	DWORD
INT_16	INT
UINT_16	WORD
UNSIGNED	WORD
CHAR	CHAR or BYTE
STRING	STRING
BOOL	BOOL
DATETIME	DATE_AND_TIME

Example

For example, to be able to read an NC variable of the type DOUBLE without adapting the format, a data area of the REAL type must be specified in the destination area "RDx" (e.g.: "DB_Read".RealValue).

10.6.1.2 Generating a data block for accessing NC variables

Function blocks LBP_ReadVar [FB2]: Read NC variable (Page 597) and LBP_WriteVar [FB3]: Write NC variable (Page 605) allow the user program to access NC variables. For this purpose, these function blocks require a correspondingly structured data block.

You have two options for generating the data block. Generation via the TIA Portal is the preferred option.

Creating a data block in TIA Portal

To generate a data block for accessing NC variables, proceed as follows:

1. Add a new user-defined variable table in the project navigation, e.g. under "CNC_1 > NCK > Configured NC variables".
2. Copy the desired entries from the "List of available NC variables" into the user-defined variable table.
3. Right-click on the user-defined variable table and select the command "Create PLC data block".

A detailed instruction may be found in the PLC manual under NC variables > Configure NC variables.

Creating a data block with the "NC-VAR-Selector" tool

To generate a data block for accessing NC variables, proceed as follows:

1. Open the "NC-VAR-Selector" tool.
2. Select the relevant NC variables with the "Insert in Project" button.
3. Press the "Generate" button to generate an STL source, e.g. "ncvar.awl".
4. In the project navigator of the TIA Portal, open the folder "External Sources".
5. Double-click the entry "Add new external file".
6. Select the file with the generated STL source, e.g. "ncvar.awl".
   The data block generated with "NC-VAR-Selector" is imported as an external source.
7. Select shortcut menu item "Generate blocks from source" for the imported source.
   A data block called, for example, "DB120 [DB120]" is generated from the imported source in the folder "Program Blocks".
8. Assign a symbolic name to the generated data block, e.g. "ncvar".

10.6.2 LBP_WriteVar [FB3]: Write NC variable

Function

The function block "LBP_WriteVar [FB3]" writes variables from the NC area.

Request for writing NC variables

Call of "LBP_WriteVar [FB3]" with positive edge, parameter "Req" = 0 → 1

S7 names of the NC variables: Parameter "Addr1" to "Addr8" = <DB name>.<S7 name>. Unused addresses must be assigned "NULL".

Variables from which the NC variable values to be written are read:

Parameter "SD1" to "SD8" . Unused parameters "SD<n>" must be assigned "NULL".

Completion of the write request

Write request successfully completed: Parameter "Done" == 1.

Write request completed with error: Parameter "Error" == 1, error cause in parameter "State"

Requirement

A data block for access to the NC variables has been generated, e.g. "ncvar [DB120]". See also Generating a data block for accessing NC variables (Page 604).

In this data block, the NC variable is specified in the parameters "Addr<x>", e.g. Addr1: = "ncvar".<NC-Variable>

General conditions

- "LBP_WriteVar [FB3]" has multi-instance capability.
  Every call of "LBP_WriteVar [FB3]" must be assigned a separate instance or an instance DB from the user area.
- In order to define machine data and GUD without a password, the protection level of the data you want to access must be redefined to the lowest level.
  References:
  - Programming Manual NC Programming; Section "Work Preparation"
- When **channel-specific** variables are written, no variables from different channels may be addressed via "Addr1" to "Addr8" when "LBP_WriteVar [FB3]" is called.

Note

Error case
When variables from different channels are written, an error message is output:

- "Error" == TRUE
- "State" == W#16#02

Variable addressing

For some NC variables, it is necessary to select "Area no." and/or "Line" or "Column" in the NC VAR selector. It is possible to select a basic type, i.e. "Area no.", "Line" and "Column" are preassigned "0". The values of the "Area no.", "Line", and "Column" specified by the NC VAR selector are checked for a "0" in "LBP_WriteVar [FB3]". If an NC-VAR selector value == "0", the corresponding value of the "LBP_WriteVar [FB3]" parameter is adopted. For this, the parameters "Unit<n>", "Column<n>" and "Line<n>" must be written with <n> = = 1, 2, ... 8 before "LBP_WriteVar [FB3]" is called.

Further explanations on the use of basic types can be found in the online help of the NC VAR selector.

"LBP_WriteVar [FB3]" parameter	NC VAR selector
Unit<xn	Area no.
Column<n>	Column
Line<n>	Line

NC variables from different areas can be combined as follows in a call:

Call	Area				
1	C[1]	N	B	A	T
2	C[2]	N	B	A	T
The same rules apply for channels 3 to 10 as for call 1 and call 2 shown in the example. A = axis-specific data, area no. corresponds to the axis number B = mode group data, area no. corresponds to the mode group number C = channel data, area no. corresponds to the channel number N = NC data, area no. is irrelevant (cannot be changed in the NC VAR selector) T = tool management data, area no. corresponds to the TO area					

Note

The number of usable variables can be less than eight, depending on the string length, when simultaneously writing several variables of the "String" type.

This is because maximum 230 bytes can be transferred with one call. Nothing is written and the following error message is output, for example:

- "Error" == TRUE
- "State" == W#16#060F (error occurred at the sixth variable, user data too long)

Description of formal parameters

Parameter	Category	Type	Value range	Meaning
Req	I	BOOL	---	Job start with positive signal edge
NumVar	I	INT	1 ... 8	Number of variables to be read: "Addr1" - "Addr8"
Addr1 Addr2 ... Addr8	I	VARIANT	<DB name>. <S7 name>	Variable identifiers from **NC Var selector**
Unit1 Unit2 ... Unit8	I	BYTE	---	Area address, optional for variable addressing
Column1 Column2 ... Column8	I	WORD	---	Column address, optional for variable addressing
Line1 Line2 ... Line8	I	WORD	---	Line address, optional for variable addressing
Error	O	BOOL	---	TRUE: Negative acknowledgement of job or job could not be executed
Done	O	BOOL	---	TRUE: Job successfully executed.
State	O	WORD	---	See paragraph "Error codes"
SD1 SD2 ... SD8	I/O	VARIANT	---	Data to be written, e.g. DB area or bit memory of the same data type as the variable to be written

Error codes

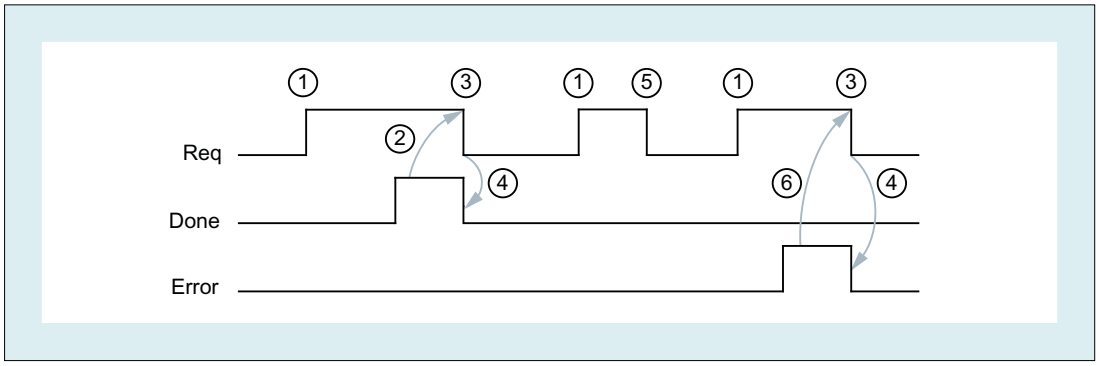
NC variables

State		Meaning	Note
High byte [1)]	**Low byte**		
1 - 8	1	Access error	The variable cannot be accessed. Possible causes: • Variable does not exist in the NCK (possibly machine data must first be set) • The "Line" parameter is wrong (0 is only allowed for certain variables, for Word variables the parameter must be even) • An attempt is made to write an invalid value
0	2	Error in job	Incorrect compilation of variables in a job
0	3	Negative acknowledgement, job not executable	Internal error, possible remedy: • Check job • NC reset
1 - 8	4	Data areas or data types do not match or string is empty	Check data in "SD1" - "SD8"
1 - 8	5	Format conversion error	Error on conversion of var. type
0	6	FIFO full	Job must be repeated since queue is full
1 - 8	10	Error in variable addressing	"Unit" or "Column"/"Line" contains value 0
0	11	Variable address invalid or variable is read-only	Check "Addr" (or variable name), "Area", "Unit"
0	12	NumVar == 0	Check parameter "NumVar"
1 - 8	15	User data too long	Remedy: Write fewer variables per job or use shorter string variables
1) High byte > 0 ⇒ Number of the variable in which the error occurred			

Configuration steps

To write NC variables, the same configuration steps are required as for reading NC variables. It is useful to store the address data of all NC variables to be read or written in a DB.

Pulse diagram



① User: Set request, Req = 0 → 1
② "LBP_WriteVar [FB3]": Writing successfully completed, Done = 1
③ User: IF Done == 1 THEN reset request: 1 → 0
④ "LBP_WriteVar [FB3]": Reset job confirmation, Done = 0
⑤ User: IF Done == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ "LBP_WriteVar [FB3]": Writing completed with error, Error = 1
User: Reset request, IF Error == 1 THEN Req = 0, possible further error handling according to status

Call example

Write three machine data whose addresses are stored in a "ncvar [DB120]".

Specification of the data

The data are selected with the "NC-Var-Selector" tool and the "ncvar [DB120]" data block is generated from it. See also Generating a data block for accessing NC variables (Page 604).

Area	Block	Name	Type	S7 name
C[1]	RP	rpa[5]	REAL	C1_RP_rpa5_1
C[1]	RP	rpa[11]	REAL	C1_RP_rpa11_1
C[1]	RP	rpa[14)	REAL	C1_RP_rpa14_1

Data block "ncvar" must be transferred to the PLC. The values to be written are read here from the "DB_Write", whereby this must be done for the correct type. This means that a REAL value to be written must be read from a data area of the REAL type in "DB_Write".

Call and parameterization of "LBP_WriteVar [FB3]" with instance DB LBP_WriteVar_DB

In this SCL example, "LBP_WriteVar [FB3]" is activated by setting "Tag_1". The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0. The values from "DB_Write" are written to the corresponding R-parameters.

Parameterization of "LBP_WriteVar [FB3]" with instance DB “LBP_WriteVar_DB”:

```
// Test "LBP_WriteVar"
// 3x real
"LBP_WriteVar_DB"(Req :="Tag_1",          // Bool
```

```
NumVar := 3,
Addr1 := "ncvar".C1_RP_rpa5_1,
Addr2 := "ncvar".C1_RP_rpa11_1,
Addr3 := "ncvar".C1_RP_rpa14_1,
Addr4 := NULL,
Addr5 := NULL,
Addr6 := NULL,
Addr7 := NULL,
Addr8 := NULL,
Error => "Tag_2",           // Bool
Done => "Tag_3",            // Bool
State => "Tag_4",           // Word
SD1 := "DB_Write".RealValue[0],
SD2 := "DB_Write".RealValue[1],
SD3 := "DB_Write".RealValue[2],
SD4 := NULL,
SD5 := NULL,
SD6 := NULL,
SD7 := NULL,
SD8 := NULL;
```

In this SCL example, "LBP_WriteVar [FB3]" is activated by setting "Tag_1".

The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0.

The values from "DB_Write" are written to the corresponding R-parameters.

Example: Variable addressing

In contrast to the previous example, the address data of the R-parameters from channel 1 are only stored as basic type in the "ncvar". The R-parameter number is parameterized via the parameter Line<n>.

The R-parameter definition in "ncvar" is created by the "NC VAR selector":

```
C1_RP_rpa0_1:
STRUCT
   SYNTAX_ID :          BYTE := B#16#82;
   area_and_unit :      BYTE := B#16#41;
   column :             WORD := W#16#1;
   line :               WORD := W#16#0;       // variable addressing
   block type :         BYTE := B#16#15;
   NO. OF LINES :       BYTE := B#16#1;
   type :               BYTE := B#16#F;
   length :             BYTE := B#16#8;
   END_STRUCT ;
```

```
// Test "LBP_WriteVar"
// 3x real, variabel adressiert
"LBP_WriteVar_DB"(Req :="Tag_1",          // Bool
             NumVar := 3,
             Addr1 := "ncvar".C1_RP_rpa0_1,
             Line1 := 5,
             Addr2 := "ncvar".C1_RP_rpa0_1,
             Line2 := 11,
             Addr3 := "ncvar".C1_RP_rpa0_1,
             Line3 := 14,
             Addr4 := NULL,
             Addr5 := NULL,
             Addr6 := NULL,
             Addr7 := NULL,
             Addr8 := NULL,
             Error => "Tag_2",            // Bool
             NDR => "Tag_3",              // Bool
             State => "Tag_4",            // Word
             SD1 := "DB_Write".RealValue[0],
             SD2 := "DB_Write".RealValue[1],
             SD3 := "DB_Write".RealValue,
             SD4 := NULL,
             SD5 := NULL,
             SD6 := NULL,
             SD7 := NULL,
             SD8 := NULL;
```

In this SCL example, "LBP_WriteVar [FB3]" is activated by setting "Tag_1".

The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0.

The values from "DB_Write" are written to the corresponding R-parameters.

10.6.3 LBP_ReadGUD [FB5]: Read GUD variable

Function

The function block "LBP_ReadGUD [FB5]" "LBP_ReadGUD" is used for reading global user data (GUD) in the NC or channel area.

For the range types, refer below to the table of the call combinations, as well as the online help of the NC-VAR selector.

When "LBP_ReadGUD [FB5]" is called, the variable name of the NC variable is transferred in the form of a string: Parameter "Addr" = <DB name>.<S7 name>.

Request for reading NC variables

Call of "LBP_ReadGUD [FB5]" with positive edge, parameter "Req" = 0 → 1

S7 name of the NC variable: Parameter "Addr" = <DB name>.<S7 name>

Parameters "Area", "Unit", "Index1", and "Index2": For additional information for addressing the variables, see the explanations in Chapter: LBP_ReadVar [FB2]: Read NC variable (Page 597) and the online help of the NC VAR selector.

GUD variables can also be read using "LBP_ReadVar [FB2]" and written using "LBP_WriteVar [FB3]". For this purpose, an address token is required which is generated on activation of parameter "CnvtToken" and used as a variable address when "LBP_ReadVar [FB2]" and "LBP_WriteVar [FB3]", parameter "Addr1"..."Addr8" = <Token> are called. If the GUD variable contains an array with several values, the parameter "Line1" ... "Line8" = <Index> must also be assigned the array index.

Completion of the read request

Read request successfully completed: Parameter "Done" == 1.

Read request completed with error: "Parameter "Error" == 1, error cause in "State"

General conditions

- "LBP_ReadGUD [FB5]" has multi-instance capability.
  Every call of "LBP_ReadGUD [FB5]" must be assigned a separate instance or an instance DB from the user area.
- Reading of a GUD variable ("LBP_ReadGUD [FB5]" call with Req = 1) is only permitted in the cyclic part of the PLC basic program ("Main [OB1]"). If the job is not started ("LBP_ReadGUD [FB5]" call with Req = 0), the parameters can also be written in the startup part of the PLC basic program ("Startup [OB100]"). The job can then be executed using the previously written parameters in the cyclic part of the PLC basic program ("Main") by calling "LBP_ReadGUD [FB5]" with Req = 1.
- Only capital letters may be used for the names of GUD variables.
- Reading of a GUD variable generally extends over several PLC cycles.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Req	I	BOOL	---	Job start with positive signal edge
Addr	I	VARIANT	"<DBName>".<VarName>	Variable name in a variable of the type STRING
Area	I	BYTE	0, 2	NC area 0 Channel area: 2
Unit	I	BYTE	1, 2, ... 10	Area == NC: 1 Area == Channel: Channel number
Index1	I	INT	0, 1, 2, ... <max. array index>	Array index 1 For variables without array index: 0

Signal	Category	Type	Value range	Meaning
Index2	I	INT	0, 1, 2, ... <max. array index>	Array index 2 For variables without 2nd array index: 0
CnvtToken	I	BOOL	---	Activate generation of a 10-byte variable token
VarToken	I	VARIANT	"<DB name>". <S7 name>	Structure of a 10byte token (see example)
Error	O	BOOL	---	TRUE: Negative acknowledgement of job or job could not be executed
Done	O	BOOL	---	TRUE: Job successfully executed.
State	O	WORD	---	See paragraph "Error codes"
RD	I/O	VARIANT		Target area for read data, e.g. DB area or bit memory of the same data type as the variable to be read

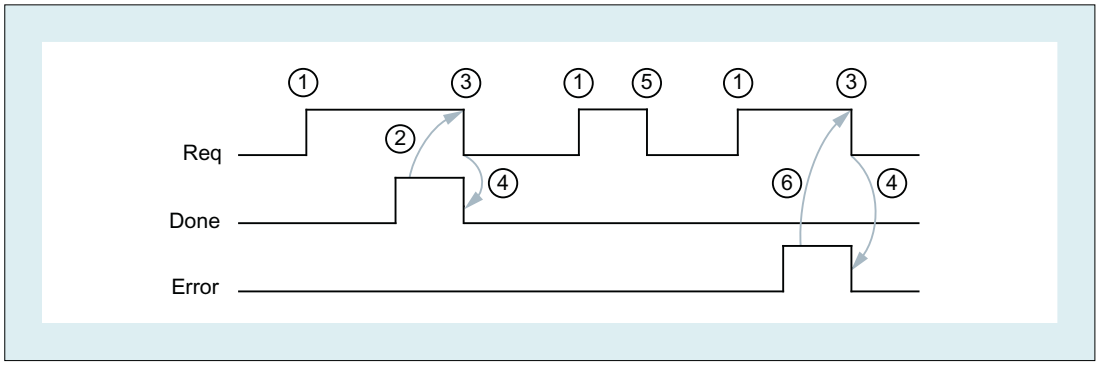
Error codes

State	Meaning	Note
WORD L		
1	Access error	
2	Error in job	Incorrect compilation of Var. in a job
3	Negative acknowledgement, job not executable	Internal error, possible remedy: NC RESET
4	Data areas or data types do not match	Check data to be read in RD
4	Insufficient local user memory available	The variable read is longer than specified in RD
5	Format conversion error	Addr parameter must be transferred as string.
6	FIFO full	Job must be repeated, since queue is full
10	Error in addressing	Unit contains value 0
11	Address of variable invalid	Check addr (or variable name), area, unit
15	String more than 32 characters	GUD variable name too long

Configuration steps

To be able to read a GUD variable, its name must be stored in a string variable.

Flow diagram



① User: Set request, Req = 0 → 1
② "LBP_ReadGUD [FB5]": Reading successfully completed, Done = 1
③ User: IF Done == 1 THEN reset request: 1 → 0
④ "LBP_ReadGUD [FB5]": Reset job confirmation, Done = 0
⑤ User: IF Done == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ "LBP_ReadGUD [FB5]": Reading service completed with error, Error = 1
User: Reset request, IF Error == 1 THEN Req = 0, possible further error handling

Call example 1

Reading of a global variable from global area:

- Name "_SC_NCK_ROU_R[0]"
- Type: REAL
- Filling of a 10-byte structure with the VarToken, e.g. for later use when writing the variable with "LBP_WriteVar [FB3]"

Reading the GUD variables: "LBP_ReadGUD [FB5]" with instance DB "LBP_ReadGUD_DB"

In this example, the global DB "DB_GUD" contains the name of the variable to be read, a structure for the "VarToken" which is filled by the call and a REAL value to take the read value.

```
DATA_BLOCK "DB_GUD"
VERSION : 0.1
NON_RETAIN
   VAR
   GUD_real : String
   Token_real : Struct
    SyntaxID : Byte;
    AreaUnit : Byte;
    ColIndex : Word;
    RowIndex : Word;
    BlockType : Byte;
    NumRows : Byte;
```

```
    S7Typ : Byte;
    NcLength : Byte;
   END_STRUCT;
Value_real : Real;
END_VAR
BEGIN
    GUD_real := '_SC_NCK_ROU_R';
END_DATA_BLOCK


// Test "LBP_ReadGUD"
// real, NC GUD
"LBP_ReadGUD_DB"(Req :="Tag_1",   // Bool
     Addr := "DB_GUD".GUD_real,
     Area := 0,                   // Global GUD, from area NC
     Addr2 := "NCVAR".C1_RP_rpa11_1,
     Unit := 1,
     Index1 := 0,
     Index2 := 0,
     CnvtToken := TRUE,
     VarToken := "DB_GUD".Token_real,
     Error => "Tag_2",            // Bool
     Done => "Tag_3",             // Bool
     State => "Tag_4",            // Word
     RD := "DB_GUD".Value_real);
```

In this SCL example, "LBP_ReadGUD [FB5]" is activated by setting "Tag_1". The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0. The "Token_real" structure in "DB_GUD" is filled as follows by "LBP_ReadGUD [FB5]":

SyntaxID = 16#82

AreaUnit = 16#01

ColIndex = 16#0002

RowIndex = 16#0000

BlockType = 16#36

NumRows = 16#01

S7Typ = 16#0F

NcLength = 16#08

The value from _SC_NCK_ROU_R[0] is written to "DB_GUD".Value_real.

Call example 2

Read a GUD variable from channel 1:

- Name "SC_CONT_NAME[0]"
- Type: STRING[32]
- Filling of a 10-byte structure with the VarToken, e.g. for later use when writing the variable with "LBP_WriteVar [FB3]".

Reading the GUD variables: "LBP_ReadGUD [FB5]" with instance DB "LBP_ReadGUD_DB"

In this example, the global DB "DB_GUD" contains the name of the variable to be read, a structure for the VarToken which is filled by the call and a string value to take the read value.

```
DATA_BLOCK "DB_GUD"
VERSION : 0.1
NON_RETAIN
   VAR
   GUD_string : String
   Token_string : Struct
    SyntaxID : Byte;
    AreaUnit : Byte;
    ColIndex : Word;
    RowIndex : Word;
    BlockType : Byte;
    NumRows : Byte;
    S7Typ : Byte;
    NcLength : Byte;
   END_STRUCT;
Value_string : String
END_VAR
BEGIN
    GUD_string := '_SC_CONT_NAME';
END_DATA_BLOCK


// Test "LBP_ReadGUD"
// string, channel GUD
"LBP_ReadGUD_DB"(Req :="Tag_1",   // Bool
     Addr := "DB_GUD".GUD_string,
     Area := 2,                   //channel GUD
     Unit := 1,                   // GUD from channel 1
     Index1 := 0,                 // _SC_CONT_NAME[0]
     Index2 := 0,
     CnvtToken := TRUE,
     VarToken := "DB_GUD".Token_string,
     Error => "Tag_2",            // Bool
```

```
    Done => "Tag_3",              // Bool
    State => "Tag_4",             // Word
    RD := "DB_GUD".Value_string);
```

In this SCL example, "LBP_ReadGUD [FB5]" is activated by setting Tag_1. The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0. The "Token_string" structure in "DB_GUD" is filled as follows by "LBP_ReadGUD [FB5]":

SyntaxID = 16#82

AreaUnit = 16#41

ColIndex = 16#001A

RowIndex = 16#0000

BlockType = 16#36

NumRows = 16#01

S7Typ = 16#13

NcLength = 16#8D

The value from _SC_CONT_NAME[0] is written to "DB_GUD".Value_real.

Writing the GUD variables: "LBP_WriteVar [FB3]" with instance DB "LBP_WriteVar_DB"

GUD variable token from "LBP_ReadGUD [FB5]", parameter: "VarToken" for writing with "LBP_WriteVar [FB3]", parameter "Addr1"

```
// Test "LBP_WriteVar"
// 1x string
"LBP_WriteVar_DB"(Req :="Tag_5",          // Bool
            NumVar := 1,
            Addr1 := "DB_GUD".Token_string,
            Line1 := 1                    // _SC_CONT_NAME[0]
            Addr2 := NULL,
            Addr3 := NULL,
            Addr4 := NULL,
            Addr5 := NULL,
            Addr6 := NULL,
            Addr7 := NULL,
            Addr8 := NULL,
            Error => "Tag_6",             // Bool
            Done => "Tag_7",              // Bool
            State => "Tag_8",             // Word
            SD1 := "DB_GUD".Value_string,
            SD2 := NULL,
            SD3 := NULL,
            SD4 := NULL,
            SD5 := NULL,
```

```
SD6 := NULL,
SD7 := NULL,
SD8 := NULL;
```

In this SCL example, "LBP_WriteVar [FB3]" is activated by setting "Tag_1". Here, VarToken "DB_GUD".Token_string" previously filled with "LBP_ReadGUD [FB5]" is used. The function is executed without errors, i.e. "Tag_2" = false, "Tag_3" = true, "Tag_4" = 0. The value from "DB_GUD".Value_string is written to the variable SC_CONT_NAME[0].

Note

Parameter index

When used in "LBP_ReadGUD [FB5]", the parameter Index1 starts at 0, in order to read SC_CONT_NAME[**0**].

Whereas, the parameter Line<n> starts at 1, in order to write SC_CONT_NAME[**0**] now.

10.6.4 LBP_ReqPIService [FB7]: Request PI service

Function

The function block "LBP_ReqPIService [FB7]" is used to start PI services.

The available PI services are described in the following chapters with their specific parameters. An overview of the available PI services can be found in the following "List of available PI services".

Start of a PI service

Request to start a PI service: Call of "LBP_ReqPIService [FB7]" with positive edge, parameter "Req" = 0 → 1

Completion of a PI service

PI service successfully completed Parameter "Done" == 1.

PI service completed with error: Parameter "Error" == 1, error cause in parameter "State"

Prerequisites

- The data interface of the PI service is provided as global DB "LBP_PIServices [DB16]" by the basic program library. The PI service to be requested is specified at "LBP_ReqPIService [FB7]" parameter "PIService", e.g. PIService:= "LBP_PIServices".<PI service>

General conditions

- "LBP_ReqPIService [FB7]" has multi-instance capability.
  Every call of "LBP_ReqPIService [FB7]" must be assigned a separate instance or and instance DB from the user area.
- A PI service must only be started ("LBP_ReqPIService [FB7]" call with "Req" = 1) in the cyclic part of the PLC basic program ("Main [OB1]").
- The execution of a PI service extends over several PLC cycles.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Req	I	BOOL	---	Job request
PIService	I	VARIANT	"LBP_PIServices". <PI service>	Requested PI service LBP_PIServices: Global DB supplied by the basic program library
Unit	I	INT	1, 2, 3 ... 10	Area number See the following description of the individual PI services.
Addr1 to Addr4	I	VARIANT	"<DBName>". <VarName>	Reference to a string Meaning depending on selected PI service
WVar1 to WVar16	I	WORD	-32768 ... 32767 8000_H - $7FFF_H$	INTEGER or WORD variable Meaning depending on selected PI service
Error	O	BOOL	---	Error status 1: Negative acknowledgement of job or job could not be executed. See State for cause of the error.
Done	O	BOOL	---	Job status 1: Job successfully executed
State	O	WORD	See table below	Error code Only relevant if "Error" == 1

The meaning of the error codes are explained in the the table below

State	Meaning	Note
3	Negative acknowledgement, job not executable	Internal error, possible remedy through an NC RESET
4	Wrong data type, Addr1 to Addr4 must be a string	Data types of the transfer parameters check Addr1 to Addr4
5	Copy error	Check the parameters
6	FIFO full	Repeat the command
12	PI service not specified.	Parameter "PIService" must not be NULL

State	Meaning	Note
14	PI service is unknown	The PI service specified in the parameter "PIService" is unknown. → Check the notation.
15	Addr1.. Addr4: String too long	The entire length from Addr1 to Addr4 must not exceed 230 bytes

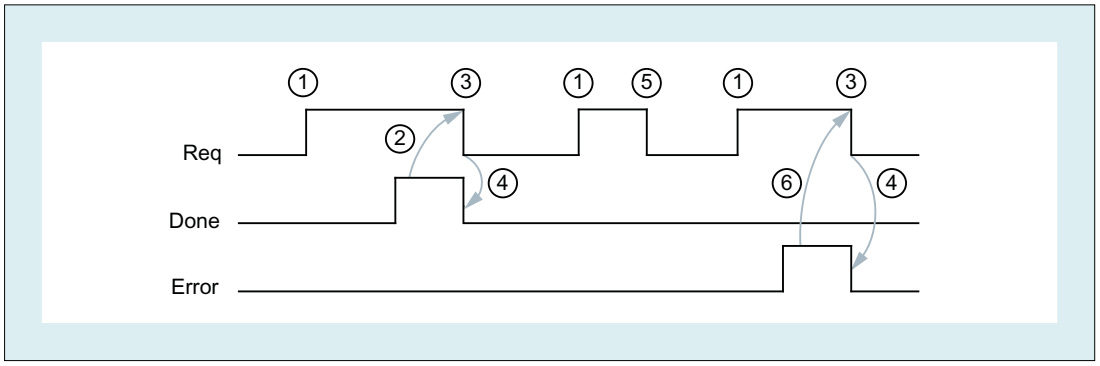
Call example

Function: Program selection in channel 1 (PI service: SELECT)

```
// Select program TEST.MPF in channel 1
//
// #LBP_ReqPIService_Instance: "LBP_ReqPIService";
// #statReq: Bool;
// #statPathName: String;
// #statProgramName: String;
// #tempError: Bool;
// #tempDone: Bool;
// #tempState: Word;
//
// #statPathName := "/_N_MPF_DIR/";
// #statProgramName := "_N_TEST_MPF";

#LBP_ReqPIService_Instance(Req := #statReq,
      PIService := "LBP_PIServices".SELECT,
      Unit := 1,                                    // Kanal 1
      Addr1 := #statPathName,
      Addr2 := #statProgramName,
      Error => #tempError,
      Done => #tempDone,
      State => #tempState);
IF #tempDone OR #tempError THEN
       #statReq := FALSE;
END_IF;
```

Pulse diagram



① User: Set request, Req = 0 → 1
② "LBP_ReqPIService [FB7]": PI service successfully completed, Done = 1
User: Reset request, IF Done == 1 THEN Req = 0
③ User: IF Done == 1 THEN reset request: 1 → 0
④ "LBP_ReqPIService [FB7]": Reset job confirmation, Done = 0
⑤ User: IF Done == 0 AND Error == 0 THEN reset request Req = 1 → 0 **not permissible**
⑥ "LBP_ReqPIService [FB7]": PI service completed with error, Error = 1
User: Reset request, IF Done == 1 OR Error == 1 THEN Req = 0, possible further error handling

10.6.4.1 List of available PI services

General PI services

PI service	Function
ASUB (Page 622)	Assign interrupt
CANCEL (Page 623)	Execute "Cancel"
CONFIG (Page 624)	Reconfiguration of tagged machine data
DIGIOF (Page 624)	Digitizing off
DIGION (Page 625)	Digitizing on
FINDBL (Page 625)	Activate block search
LOGIN (Page 626)	Activate password
LOGOUT (Page 626)	Reset password
NCRES (Page 627)	Trigger NC-RESET
RETRAC (Page 627)	Retraction of the tool in the tool direction
SELECT (Page 628)	Select program for processing for one channel
SETUDT (Page 629)	Sets the current user data to active
SETUFR (Page 629)	Activate user frame

PI services of tool management

PI service	Function
CRCEDN (Page 630)	Create a tool cutting edge with specification of the T number
CREACE (Page 630)	Create a tool cutting edge with the next higher/free D number

PI service	Function
CREATO (Page 631)	Create a tool with specification of a T number.
DELECE (Page 631)	Delete a tool cutting edge
DELETO (Page 632)	Delete tool
MMCSEM (Page 632)	Semaphores for various PI services
TMCRTO (Page 633)	Create a tool with specification of a name, a duplo number
TMFDPL (Page 634)	Empty location search for loading
TMFPBP (Page 635)	Empty location search
TMGETT (Page 637)	T number for the specified tool name with duplo number
TMMVTL (Page 637)	Prepare magazine location for loading, unload tool
TMPOSM (Page 639)	Position magazine location or tool
TMPCIT (Page 640)	Set increment value for workpiece counter
TMRASS (Page 640)	Reset active status
TRESMO (Page 641)	Reset monitoring values
TSEARC (Page 641)	Complex search using search screen forms
TMCRMT (Page 644)	Create multitool
TMDLMT (Page 645)	Delete multitool
POSMT (Page 646)	Position multitool
FDPLMT (Page 647)	Search/check an empty location within the multitool

10.6.4.2 PI service: ASUP

Function: Assign interrupt

An interrupt number and a priority in the specified channel is assigned to a part program stored in the control, identified by a path name and a program name.

The PI service ASUB has the same effect as the program statement SETINT (or as CLRINT if the call of the PI service is performed without specification of the path name).

In contrast to SETINT (or CLRINT), the PI service ASUB also remains effective after the end of the program (M30 or channel reset). An assignment from the PI service ASUB is only deleted after a warm restart.

For detailed information about program management, path and file names, see:

References:
Programming Manual NC Programming; Section "Work Preparation" > "File and Program Management" > "Program Memory"

Possible file types

Block types	
Workpiece directory	WPD
Main program	MPF
Subprogram	SPF
Cycles	CYC
Asynchronous subprograms	ASP
Binary files	BIN

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".ASUP	Assign interrupt [1]
Unit	INT	1, 2, 3, ... 10	Channel number
Addr1	STRING	"/_N_xxx_yyy/"	Path name xxx: File type yyy: DIR or WPD
Addr2	STRING	"_N_xxx_yyy"	Program name xxx: maximum 24 characters yyy: File type
WVar1	WORD	1, 2, 3, ... 8	Interrupt number
WVar2	WORD	1, 2, 3, ... 8	Priority Priority 1 is the highest priority
WVar3	WORD	0, 1	LIFTFAST [2] Fast retraction from the contour
WVar4	WORD	0, 1	BLSYNC [3] Execution of the interrupt routine not to start until the next block change

[1] As an alternative for the interrupt assignment, the SETINT command can be used. See [2]

[2] **References:**
Programming Manual NC Programming; Section "Work preparation" > "Interrupt routine (ASUB)" > "Fast retraction from the contour (SETINT, **LIFTFAST**, ALF)"

[3] **References:**
Programming Manual NC Programming; Section "Work preparation" > "Interrupt routine (ASUB)" > "Assign and start interrupt routine (SETINT, PRIO, **BLSYNC**)"

Note

The ASUB PI service may only be executed when the specified channel is in the RESET state. An ASUB prepared with "LBP_ReqPIService [FB7]" can be subsequently initiated with "LBP_ReqASUP [FC9]".

References:
Programming Manual NC Programming; Section "Work Preparation" > "Interrupt routine (ASUB)"

See also

List of available PI services (Page 621)

10.6.4.3 PI service: CANCEL

Function: Execute "Cancel"

Triggers the "Cancel" function equivalent to the corresponding "Cancel alarm" button on the user interface (operator panel front).

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".CANCEL	Cancel
Unit	INT	1	NC

See also

List of available PI services (Page 621)

10.6.4.4 PI service: CONFIG

Function: Reconfiguration

Machine data of effectiveness level "NEW_CONF" (cf) that were entered sequentially by the operator or the PLC, are activated immediately.

The command can only be activated when the controller is in RESET state or the program is interrupted (NC stop at block limit). An error checkback signal is output if this condition is not fulfilled "LBP_ReqPIService [FB7]" (state = 3).

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".CONFIG	Reconfiguration
Unit	INT	1	NC
WVar1	INT	1	Classification

See also

List of available PI services (Page 621)

10.6.4.5 PI service: DIGIOF

Function: Digitizing off

Deactivate digitizing in the parameterized channel.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".DIGIOF	Digitizing off
Unit	INT	1, 2, 3, ... 10	Channel

See also

List of available PI services (Page 621)

10.6.4.6 PI service: DIGION

Function: Digitizing on

Select digitizing in the parameterized channel.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".DIGION	Digitizing on
Unit	INT	1, 2, 3, ... 10	Channel

See also

List of available PI services (Page 621)

10.6.4.7 PI service: FINDBL

Function: Activate block search

A channel is switched to block search mode as a result of which a positive acknowledgment (Done == TRUE) is output. The block search is then executed immediately by the NC. The block search pointer must already be active in the NC at this time. The block search can be interrupted at any time by an NC RESET. Once the block search is successfully completed, the normal processing mode is reactivated automatically. NC start then takes effect from the located search target.

It is the sole responsibility of the operator to ensure a collision-free approach path.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".FINDBL	Block search
Unit	INT	1, 2, 3 ... 10	Channel
WVar1	WORD	1, 2, 4, 5	Preprocessing mode 1: Without calculation 2: With calculation, with approach 4: With calculation, without approach 5: With program test

See also

List of available PI services (Page 621)

10.6.4.8 PI service: LOGIN

Function: Create password

Transfers the parameterized password to the NC. The password consists of at least 8 characters. For shorter passwords, the string can be supplemented with empty spaces to attain 8 characters.

Note

Visibility of the password in the PLC project

The password for the NCK access level must be stored in the PLC for the use of the PI service LOGIN. This makes the password for the NCK level visible to anyone who has access to the PLC user program.

Example

Password: STRING := 'SafePassword.123';

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".LOGIN	Create password
Unit	INT	1	NC
Addr1	STRING	8 ... 32 characters	Password

See also

List of available PI services (Page 621)

10.6.4.9 PI service: LOGOUT

Function: Reset password

The password last transferred to the NC is reset.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".LOGOUT	Reset password
Unit	INT	1	NC

See also

List of available PI services (Page 621)

10.6.4.10 PI service: NCRES

Function: Trigger NC-RESET

Triggers an NC-RESET.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".NCRES	Trigger NC-RESET
Unit	INT	1	NC

See also

List of available PI services (Page 621)

10.6.4.11 PI service: RETRAC

Function: Select JOG retract

Selects the JOG retract mode. The retraction axis, i.e. the geometry axis, with which the retraction is executed can be determined by the NC automatically or specified explicitly.

The mode remains active until it is ended with RESET.

Note

The RETRAC PI service can only be activated in JOG mode in the "Reset" state.

Automatic determination

For automatic determination, the geometry axis is selected as a retraction axis, which is perpendicular (orthogonal) to the currently selected working plane:

- G17: Retraction axis ⇒ 3rd geometry axis (Z)
- G18: Retraction axis ⇒ 2nd geometry axis (Y)
- G19: Retraction axis ⇒ 1st geometry axis (X)

Note

OPI variable retractState

The active retraction axis can be read via the OPI variable retractState.Bit 2/3.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".RETRAC	Select JOG retract mode
Unit	INT	1, 2, 3, ... 10	Channel

Signal	Type	Value range	Meaning
WVar1	WORD	0, 1, 2, 3	Retraction axis 0: Automatic determination of the retraction axis by the NC. 1: Retraction axis is the 1st geometry axis of the WCS 2: Retraction axis is the 2nd geometry axis of the WCS 3: Retraction axis is the 3rd geometry axis of the WCS
WVar2	WORD	0	Reserved. The value must be pre-assigned with 0.

See also

List of available PI services (Page 621)

10.6.4.12 PI service: SELECT

Function: Select processing for a channel

A program stored on the NC is selected for one channel for processing. The program must be executable. The path and program name must be stated in full.

For detailed information, please refer to:

References

Programming Manual NC Programming; Section "Work preparation" > "File and program management" > "Program memory"

Possible file types

Block types	
Workpiece directory	WPD
Main program	MPF
Subprogram	SPF
Cycles	CYC
Asynchronous subprograms	ASP
Binary files	BIN

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".SELECT	Program selection
Unit	INT	1, 2, 3, ... 10	Channel

Signal	Type	Value range	Meaning
Addr1	STRING	"/_N_xxx_yyy/"	Path name xxx: File type yyy: DIR or WPD
Addr2	STRING	"_N_xxx_yyy"	Program name xxx: maximum 24 characters yyy: File type

See also

List of available PI services (Page 621)

10.6.4.13 PI service: SETUDT

Function: Set current user data active

The current user data, such as tool offsets, basic frames and settable frames are set to active in the next NC block (only in stop state).

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".SETUDT	Activate user data
Unit	INT	1, 2, 3, ... 10	Channel
WVar1	WORD	1, 2, 3, 4, 5	User Data Type 1: Active tool offset 2: Active basic frame 3: Active settable frame 4: Active global basic frame 5: Active global settable frame
WVar2	WORD	0	Reserved
WVar3	WORD	0	Reserved

See also

List of available PI services (Page 621)

10.6.4.14 PI service: SETUFR

Function: Activate user frames

User frames are loaded to the NC. All required frame values must first be transferred to the NC with "LBP_WriteVar [FB3]".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".SETUFR	Activate user frames
Unit	INT	1, 2, 3, ... 10	Channel

See also

List of available PI services (Page 621)

10.6.4.15 PI service: CRCEDN

Function: Creates new cutting edge

If the T number of an existing tool is entered in the "T number" parameter of the PI service, then a tool cutting edge with the stated D number is created for this tool.

If a T number is specified as a parameter and the tool for the entered T number does not exist, the PI service is aborted.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".CRCEDN	Create new cutting edge
Unit	INT	1 ... 10	TOA
WVar1	INT	1, 2, 3 ... 32000	T number of the tool for which the tool cutting edge must be created.
WVar2	INT	0, 1, 2 ...	Edge number of tool cutting edge 0: D number is assigned by the NC

See also

List of available PI services (Page 621)

10.6.4.16 PI service: CREACE

Function: Create tool cutting edge

Create cutting edge with next free D number for the tool with the transferred T number. If the tool for the stated T number does not exist, the PI service fails.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".CREACE	Create tool cutting edge
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	1, 2, 3, ... 32000	T number

See also

List of available PI services (Page 621)

10.6.4.17 PI service: CREATO

Function: Create tool

Create a tool with specification of a T number. If a tool with the stated T number already exists, the PI service fails.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".CREATO	Create tool
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	1, 2, 3, ... 32000	T number

See also

List of available PI services (Page 621)

10.6.4.18 PI service: DELECE

Function: Delete a tool cutting edge

If the T number of an existing tool is entered in the "T number" parameter of the PI service, the cutting edge with the specified D number is deleted from this tool. If a T number is entered as a parameter and the tool for the specified T number does not exist, the PI service is aborted. If the specified cutting edge does not exist, the PI service is aborted.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".DELECE	Delete cutting edge
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	1, 2, 3, ... 32000	T number of the tool for which the tool cutting edge is to be deleted.
WVar2	INT	0, 1, 2, ...	D number of the tool cutting edge that must be deleted. 0: The tool is deleted

See also

List of available PI services (Page 621)

10.6.4.19 PI service: DELETO

Function: Delete tool

Deletes the tool assigned to the transferred T number together with all cutting edges. If the tool for the stated T number does not exist, the PI service fails.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".DELETO	Delete tool
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	1, 2, 3, ... 32000	T number

See also

List of available PI services (Page 621)

10.6.4.20 PI service: MMCSEM

Function: Semaphores for various PI services

The PI service is used for testing, setting and resetting of channel-specific semaphores from HMI through the PLC. 10 semaphores are available per channel for protecting critical data areas.

Functions (PI services) are assigned to semaphores 1 to 6. The semaphores 7 to 10 can be freely used.

"LBP_ReqPIService [FB7]" return values

- "Done" == TRUE
  The semaphore was set, the critical function can be called.
- "Error" == TRUE AND "State" == 3
  The semaphore was already set, currently the critical function cannot be called.

Schematic sequence for free semaphore

```
Testing and setting the semaphore
IF semaphore == FREE
THEN
   Writing/reading of critical data
   Resetting the semaphore
ELSE // Semaphore is blocked
...
ENDIF
```

NOTICE
Resetting the semaphore After blocking the critical data area by setting the semaphore and subsequent reading or writing of the data, the critical data area must be enabled once again by resetting the semaphore, otherwise subsequent blocking will not be possible.

Schematic sequence for blocked semaphore

```
Testing and setting the semaphore
IF semaphore == FREE
THEN
   ...
ELSE // Semaphore is blocked
   Set bit memory for "Function could not be executed, repeat
necessary"
ENDIF
```

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".MMCSEM	Edit semaphore
Unit	INT	1, 2, 3 ... 10	Channel
WVar1	INT	1)	PI service-specific number of the semaphore
WVar2	WORD	0, 1	Job type 0: Reset semaphore 1: Test and set semaphore
1)	**Number**	**PI service**	
	1	TMCRTO (create tool)	
	2	TMFDPL (search for empty location for loading) (Available for SINUMERIK ONE)	
	3	TMMVTL (prepare magazine location for loading, unload tool) (Available for SINUMERIK ONE)	
	4	TMFPBP (search for location) (Available for SINUMERIK ONE)	
	5	TMGETT (search for tool number)	
	6	TSEARC (search for tool) (Available for SINUMERIK ONE)	
	7 ... 10	Freely usable	

See also

List of available PI services (Page 621)

10.6.4.21 PI service: TMCRTO

Function: Create tool

Creates a tool by specifying a name, e.g. with $TC_TP2[y] = "<tool name>" or a duplo number, e.g. with $TC_TP1[y] = <duplo number>. Or optionally by a T number, e.g. with y = <T number>.

After execution of the PI service, the T number of the tool created is available in the NC variable **TnumWZV**.

Note

Before and after this PI service, PI service MMCSEM must be called with parameter "WVar1" =1 (TMCRTO). See Chapter "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMCRTO	Create tool
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	-1, 1, 2, 3, ... 32000	T number The stated T number is specified. -1: the NC assigns the T number
WVar2	INT	0, 1, 2, ... 32000	Duplo number 0: Duplo number is assigned by the NC
Addr1	STRING	Max. 31 characters	Tool name

See also

List of available PI services (Page 621)

10.6.4.22 PI service: TMFDPL

Function: Empty location search for loading

Search for empty location for loading, depending on the parameter assignment:

- Location number (target) = -1 AND magazine number (target) = -1
  Searches all magazines in the specified area (= channel) for an empty location for the tool specified with a T number.
- Location number (target) = -1 AND magazine number (target) = magazine number
  An empty location for the tool specified with a T number is searched for in the specified magazine.
- Location number (target) = location number AND magazine number (target) = magazine number
  The specified location is checked to ensure that it is free for loading of the tool.

After execution of the PI service, the magazine and location numbers are stated in the NC variables **magCMCmdPar1** (magazine number) and **magCMCmdPar2** (location number).

Location number (reference) and magazine number (reference) can be set as search criteria or with -1. The PI service is acknowledged positively or negatively depending on the search result.

Note

Before and after this PI service, PI service MMCSEM must be called with parameter "WVar1" =2 (TMFDPL).
See Chapter "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBR_PIService".TMFDPL	Empty location for loading
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	1, 2, 3, ...	T number
WVar2	INT	-1, 1, 2, 3...	Location number (target)
WVar3	INT	-1, 1, 2, 3...	Magazine number (target)
WVar4	INT	-1, 1, 2, 3...	Location number (reference)
WVar5	INT	-1, 1, 2, 3...	Magazine number (reference)

See also

List of available PI services (Page 621)

10.6.4.23 PI service: TMFPBP

Function: Empty location search

The PI service searches the specified magazines for an empty location which satisfies the criteria for tool size and location type.

If the search is successful, the result can be read in the NC variables:

- magCMCmdPar1 (magazine number)
- magCMCmdPar2 (location number)

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 4 for TMFPBP. See Chapter "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMFPBP	Empty location search
Unit	INT	1, 2 , 3, ... 10	TOA
WVar1	INT	-1, 1, 2 , 3...	**Magazine number** of the magazine **from** which the search is to be performed

Signal	Type	Value range	Meaning
WVar2	INT	-1, 1, 2 , 3...	**Location number** of the location in the magazine from "WVar1"
WVar3	INT	-1, 1, 2 , 3...	**Magazine number** of the magazine up to which the search is to be performed
WVar4	INT	-1, 1, 2 , 3...	**Location number** of the location in the magazine from "WVar3"
WVar5	INT	-1, 1, 2 , 3...	Magazine number reference
WVar6	INT	-1, 1, 2 , 3...	Location number reference
WVar7	INT	0, 1, 2 ... 7	Number of required half locations to left
WVar8	INT	0, 1, 2 ... 7	Number of required half locations to right
WVar9	INT	0, 1, 2 ... 7	Number of required half locations in upward direction
WVar10	INT	0, 1, 2 ... 7	Number of required half locations in downward direction
WVar11	INT	1, 2 , 3...	Number of required location type
WVar12	INT	0, 1, 2, 3	Specifies the required search direction 0: Search strategy as set in $TC_MAMP2 1: Forward 2: Backward 3: Symmetrical

Examples: Setting the search range

From	Location	To	Location	Description
WVar1	WVar2	WVar3	WVar4	
#M1	#L1	#M1	#L1	Only location #L1 in magazine #M1 is checked
#M1	#L1	#M2	#L2	Locations starting at magazine #M1, location #L1 up to magazine #M2, location #L2 are searched
#M1	-1	#M1	-1	All locations in magazine #M1 are searched
#M1	-1	-1	-1	All locations starting at magazine #M1 are searched
#M1	#L1	-1	-1	All locations starting at magazine #M1 location #L1 are searched
#M1	#L1	#M1	-1	Locations in magazine #M1 starting at location #L1 are searched
#M1	#L1	#M2	-1	Locations starting at magazine #M1 location #L1 up to and including magazine #M2 are searched
#M1	-1	#M2	#L2	Locations starting at magazine #M1 up to and including magazine #M2 location #L2 are searched
#M1	-1	#M2	-1	All locations starting at magazine #M1 up to and including magazine #M2 are searched
-1	-1	-1	-1	All magazine locations are searched

See also

LBP_ReqPIService [FB7]: Request PI service (Page 618)

List of available PI services (Page 621)

10.6.4.24 PI service: TMGETT

Function: Determine T number for the specified tool name with duplo number

The PI service determines the T number of a tool via the tool name and duplo number.

The result is written to these NC variables:

- resultNrOfTools
  - resultNrOfTools == 0: The specified tool was not found
  - resultNrOfTools == 1: The specified tool was found
- resultToolNr: T number of the specified tool with "resultNrOfTools" == 1

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 5 for TMGETT. See Chapter "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMGETT	Determining the T number
Unit	INT	1, 2, 3, ... 10	TOA
Addr1	STRING	Max. 31 characters	Tool name
WVar1	INT	1, 2, 3, ... 32000	Duplo number

See also

List of available PI services (Page 621)

10.6.4.25 PI service: TMMVTL

Function: Prepare magazine location for loading, unload tool

The PI service loads, unloads and relocates tools:

1. Loading and unloading: loading location ↔ magazine
2. Loading and unloading: loading location ↔ buffer storage, e.g. spindle
3. Relocation within a magazine
4. Relocation between different magazines
5. Relocation between magazine and buffer storage
6. Relocation within buffer storage

Case 1, 3, 4 and 5: the following NC variables are written:

- magCmd (areaNo = TOA, line = magazine number)
- magCmdState ← "acknowledgement"

Case 2 and 6: the following NC variables are written:

- magCBCmd (areaNo = TOA)
- magCBCmdState ← "acknowledgement"

Loading

- "WVar2" location number (source), "WVar3" magazine number (source)
  The tool location of the magazine is moved to the loading station for loading and the tool is loaded.
- "WVar4" location number (target) == -1
  First, an empty location for the tool is searched for in the magazine. Then the empty location of the magazine is moved to the loading station for loading and the tool is loaded.

Unloading

- "WVar4" location number (target), "WVar5" magazine number (target)
  The tool location of the magazine is moved to the loading station for loading and the tool is unloaded.

In the TOA, the magazine location of the removed tool is designated as free.

Addressing the tool

The tool can be addressed either using a T number or by means of the location and magazine numbers. The value -1 is to be assigned to unused parameters.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" must be called with function number 3 for TMMVTL. See Chapter "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMMVTL	Prepare magazine location for loading, unload tool
Unit	INT	1, 2, 3 ... 10	TOA
WVar1	INT	-1, 1, 2, 3 ... 32000	T number
WVar2	INT	-1, 1, 2, 3...	Location number (source)
WVar3	INT	-1, 1, 2, 3...	Magazine number (source)
WVar4	INT	-1, 1, 2, 3...	Location number (target)
WVar5	INT	1, 2, 3, ...	Magazine number (target)

See also

List of available PI services (Page 621)

10.6.4.26 PI service: TMPOSM

Function: Position magazine location or tool

This PI service can traverse a magazine to position a magazine location at a specified position (e.g. at a loading location). The magazine location can be specified directly or via a tool at the location.

The destination, e.g. the loading location, is specified in the parameters:

- "WVar5" location number (reference)
- "WVar6" magazine number (reference)

The magazine location to be positioned is specified in the following parameters, depending on the respective specification:

- "WVar1" T number of the tool
  The following parameters are not evaluated:
  - "Addr1" tool name = ""
  - "WVar2" duplo number = -1
  - "WVar3" location number (source) = -1
  - "WVar4" magazine number (source) = -1
- "Addr1" tool name, "WVar2" duplo number
  The following parameters are not evaluated:
  - "WVar1" T number of the tool = -1
  - "WVar3" location number (source) = -1
  - "WVar4" magazine number (source) = -1
- "WVar3" location number (source), "WVar4" magazine number (source)
  The following parameters are not evaluated:
  - "Addr1" tool name = ""
  - "WVar1" T number of the tool = -1
  - "WVar2" duplo number = -1

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMPOSM	Position magazine location or tool
Unit	INT	1, 2, 3, ... 10	TOA
Addr1	STRING	Max. 31 characters	Tool name
WVar1	INT	-1, 1, 2, 3, ... 32000	T number of the tool
WVar2	INT	-1, 1, 2, 3, ... 32000	Duplo number
WVar3	INT	-1, 1, 2, 3...	Location number (source)
WVar4	INT	-1, 1, 2, 3...	Magazine number (source)
WVar5	INT	1, 2, 3, ...	Location number (reference)
WVar6	INT	1, 2, 3, ...	Magazine number (reference)

See also

List of available PI services (Page 621)

10.6.4.27 PI service: TMPCIT

Function: Set increment value for workpiece counter

Incrementing the workpiece counter of the spindle tool

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMPCIT	Set increment value for workpiece counter
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	WORD	0, 1, 2, ... max.	Spindle number
WVar2	WORD	0 ... max.	Increment value; indicates the number of spindle revolutions after which the workpiece counter is incremented

See also

List of available PI services (Page 621)

10.6.4.28 PI service: TMRASS

Function: Resetting the tool status "active"

The PI service sets the status to "inactive" for all tools with a tool status "active" and "blocked".

The following events are sensible times for resetting the tool status:

- a negative edge of the NC/PLC interface signal "tool disable ineffective"
- End of program
- Channel reset.

The PI service is intended for the PLC, since it is known here when the disabled tool is no longer to be used.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService". TMRASS	Reset active status
Unit	INT	1, 2, 3, ... 10	TOA

See also

List of available PI services (Page 621)

10.6.4.29 PI service: TRESMO

Function: Reset monitoring values

This PI service resets the monitoring values of the designated cutting edges of the designated tools to their setpoints (initial values).
This is only performed for tools with active monitoring.

See also the RESETMON command.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TRESMO	Reset monitoring values
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	WORD	0 1 ... 32000 -1 ... -32000	Tool number 0: Process all tools > 0: Process specified tool < 0: Process all replacement tools of the specified tool
WVar2	WORD	0, 1, 2, 3, ...	D number > 0: Reset the monitoring of the specified cutting edge 0: Reset the monitoring of all cutting edges
WVar3	WORD	0, 1, 2 ... 15	Monitoring mode that is to be reset (all combinations are possible):
			Bit 0 = 1: Tool-life monitoring
			Bit 1 = 1: Quantity monitoring
			Bit 2 = 1: Wear monitoring
			Bit 3 = 1: Sum-offset monitoring
			0: Reset all active tool-monitoring functions ($TC_TP9).

See also

List of available PI services (Page 621)

10.6.4.30 PI service: TSEARC

Function: Complex search using search screen form, depending on the parameter assignment

The PI service searches for tools with specific properties within a search range in one or more magazines, beginning with a specific location, up to a specific location.

Note

The PI service is only available if tool management is activated.

Specification options

- Search direction
- Search for next tool with the specified property
- Search for all tools with the specified property

Logic operations

For filtering properties, only one AND link is available as a linking option. An OR link must be achieved by the user via several calls of the PI service and subsequent evaluation of the individual results.

Parameterization of the tool properties

The properties of the tools being searched for are set in the the following variables:

- "parMask<X>" parameterization mask
- "parData<X>" comparison values

With <X> = TAD, TAO, TAS, TD

Result list

After completion of the PI service without errors, the search result will be in the following variables:

- "resultCuttingEdgeNrUsed" D numbers of cutting edges used since last quantity count
- "resultNrOfCutEdgesUsed" Number of cutting edges since last quantity count
- "resultNrOfTools" Number of found tools
- "resultToolNr" T numbers of found tools
- "resultToolNrUsed" T numbers of cutting edges used since last quantity count

If no tool was found, the number of found tools is zero ("resultNrOfTools" == 0).

Search range specifications

From mag. number **WVar1**	From location number **WVar2**	To mag. number **WVar3**	To location number **WVar4**	Description **The following magazine locations are searched:**
#M<a>	#L<b>	#M<x>	#L<y>	From: Magazine #M<a>, location #L<b> to: Magazine #M<x>, location #L<y>
#M<a>	-1	#M<a>	-1	From: Magazine #M<a>, first location to: Magazine #M<a>, last location
#M<a>	-1	-1	-1	From: Magazine #M<a>, first location to: Last magazine, last location
#M<a>	#L<b>	-1	-1	From: Magazine #M<a>, location #L<b> to: Last magazine, last location
#M<a>	#L<b>	#M<a>	-1	From: Magazine #M<a>, location #L<b> to: Magazine #M<a>, last location

From mag. number **WVar1**	From location number **WVar2**	To mag. number **WVar3**	To location number **WVar4**	Description **The following magazine locations are searched:**
#M<a>	#L<b>	#M<x>	-1	From: Magazine #M<a>, location #L<b> to: Magazine #M<x>, last location
#M<a>	-1	#M<x>	#L<y>	From: Magazine #M<a>, first location to: Magazine #M<x>, location #L<y>
#M<a>	-1	#M<x>	-1	From: Magazine #M<a>, first location to: Magazine #M<x>, last location
-1	-1	-1	-1	From: First magazine, first location to: Last magazine, last location

Symmetrical search

A symmetrical search, relative to a magazine location, can only be performed if the following conditions have been met:

- The search range must encompass only one magazine: "WVar1" (from: Magazine number) == "WVar3" (to: Magazine number)
- Specification of a reference location, i.e. a magazine location for which a symmetrical search is to be performed: "WVar5" (number of the reference magazine) and "WVar6" (number of the reference location)
- For the reference location, a multiple assignment to the magazine to be searched must have been configured in the TPM block.
- "WVar7" (search direction) = 3

The reference location is a buffer magazine location, i.e. a tool location from the buffer magazine or from the internal loading magazine, e.g. change position, gripper, loading station. The symmetrical search is made in relation to the magazine location in front of the reference location.

If the magazine location is upstream of the reference location outside of the search range, the PI service will respond as if no suitable location had been found.

Note

Before and after this PI service, PI service MMCSEM must be called up with parameter "WVar1":=6 (TSEARC). See Chapter "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TSEARC	Complex search using search screen forms
Unit	INT	1, 2, 3, ... 10	TOA
WVar1	INT	-1, 1, ...	From: Magazine number
WVar2	INT	-1, 1, ...	From: Location number
WVar3	INT	-1, 1, ...	to: Magazine number
WVar4	INT	-1, 1, ...	to: Location number

Signal	Type	Value range	Meaning
WVar5	INT	-1, 1, ...	Number of the reference magazine (only relevant for symmetrical search: Search direction ==3)
WVar6	INT	-1, 1, ...	Number of the reference location (only relevant for symmetrical search: Search direction ==3)
WVar7	INT	1, 2, 3	Search direction: 1: Forwards from the first location of the search range 2: Backwards from the last location of the search range 3: Symmetrical to the real magazine location, which is located **upstream** of the location specified at "WVar5" (number of the reference magazine) and "WVar6" (number of the reference location)
WVar8	INT	0, 1, 2, 3	Properties of the tools: 0: All tools, cutting edge-specific 1: First tool, cutting edge-specific 2: All tools, via all cutting edges 3: First tool, via all cutting edges

See also

List of available PI services (Page 621)

10.6.4.31 PI service: TMCRMT

Function: Create multitool

The PI service creates a new multitool with a defined identifier, an optionally specifiable multitool number, the number of tool locations, and the type of distance coding.

Note

Before and after this PI service, the PI service MMCSEM with parameter "WVar1" =1 (TMCRTO) must be called. See Chapter: "PI service: MMCSEM (Page 632)".

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMCRMT	Create multitool
Unit	INT	1, 2, 3, ... 10	TOA
Addr1	STRING	Max. 31 characters	Multitool identifier
WVar1	INT	0	

Signal	Type	Value range	Meaning
WVar2	INT	-1, 1, 2, ... 32000	Multitool number -1: Automatic assignment of the multi-tool number by NC
WVar3	INT	2, 3, 4, ...	Number of tool locations See MD17504 $MN_MAX_TOOLS_PER_MULTITOOL
WVar4	INT	1, 2, 3	Type of distance coding 1: Location number 2: Length 3: Angle

See also

List of available PI services (Page 621)

10.6.4.32 PI service: TMDLMT

Function: Delete multitool

The PI service is used to delete a multitool in the stated TOA. Tools equipped in multitool are not subsequently equipped or loaded, however, continue to be defined if they are not also to be deleted.

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".TMDLMT	Delete multitool
Unit	INT	1, 2, 3, ... 10	TOA
Addr1	STRING	Max. 31 characters	Multitool identifier
WVar1	INT	0	
WVar2	INT	-1, 1, 2, ... 32000	Multitool number -1: Delete the multitool with the name specified in "Addr1"
WVar3	INT	0, 1	Tools contained in the multitool: 0: Do not delete 1: Delete

See also

List of available PI services (Page 621)

10.6.4.33 PI service: POSMT

Function: Position multitool

The PI service positions a multitool at the programmed location or alternatively at the programmed tool, which is located in one of the locations of the multitool. The tool itself can either be specified using its T number or with its name and duplo number. A multitool can only be positioned if it is at a tool holder (e.g. spindle) and if no tool offset with regard to this tool carrier is active.

Position specification

Position specification can be specified as one of three variants:

No.	Addr1	WVar1	WVar2	WVar3	WVar4
1	Empty string	Number of the tool holder	Tool number	-1	-1
2	Tool name	Number of the tool holder	-1	Duplo number	-1
3	Empty string	Number of the tool holder	-1	-1	Multitool location number

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".POSMT	Position multitool
Unit	INT	1, 2, 3, ... 10	TOA
Addr1	STRING	Max. 31 characters	Tool name of the tool to be positioned in the multitool
WVar1	INT	1, 2, 3, ...	Number of the tool holder
WVar2	INT	-1, 1, 2, 3 ... 32000	Tool number (T number) of the tool to be positioned in the multitool -1: "Addr1" (tool name) and "WVar3" (duplo number) are used
WVar3	INT	-1, 1, 2, 3 ... 32000	Duplo number of the tool to be positioned in the multitool -1: "WVar2" (T number) is used
WVar4	INT	-1, 1, 2, 3 ...	Multitool location number of the tool holder location to which the system should be positioned -1: "WVar2" or "Addr1" and "WVar3" are used

See also

List of available PI services (Page 621)

10.6.4.34 PI service: FDPLMT

Function: Searching for or checking an empty location within the multitool

The PI service searches for a free tool location in a multitool to accommodate the specified tool or for checking whether the specified tool location in the multitool for accommodating the specified tool is free. The tool can be specified using the T number or the identifier and the duplo number.

Note

Before and after this PI service, PI service MMCSEM is called with parameter "WVar1" :=2 (TMFDPL). See Chapter "PI service: MMCSEM (Page 632)".

Position specification

The tool to be positioned in the multitool can be specified as one of the following variants:

- "Addr1" = <Tool name>, "WVar1" = -1 and "WVar2" = <Duplo number>
- "Addr1" = <Empty string>, "WVar1" = <Tool number> and "WVar2" = -1

Position output

If the search is successful, the result can be read in the NC variables:

- magCMCmdPar1 (multitool number)
- magCMCmdPar2 (multitool location number)

Description of formal parameters

Signal	Type	Value range	Meaning
PIService	VARIANT	"LBP_PIService".FDPLMT	Search/check an empty tool location within a multitool
Unit	INT	1, 2, 3, ... 10	TOA
Addr1	STRING	Max. 31 characters	Tool name of the tool to be positioned in the multitool
WVar1	INT	-1, 1, 2, 3, ... 32000	Tool number (T number) of the tool to be positioned in the multitool -1: "Addr1" (tool name) and "WVar2" (duplo number) are used.
WVar2	INT	-1, 1, 2, 3, ... 32000	Duplo number of the tool to be positioned in the multitool -1: "WVar1" (T number) is used

Signal	Type	Value range	Meaning
WVar3	INT	-1, 1, 2, 3, ... 32000	Number of the multitool -1: Search across all multitools for an empty location or check across all multitools whether the specified tool location is free in one of them for accommodating the tool.
WVar4	INT	-1, 1, 2, 3, ...	Multitool location number of the tool location to which the system should be positioned -1: Search for any empty tool location within the multitool

See also

List of available PI services (Page 621)

10.6.5 LBP_Powerride [FB21]: rotary/push control dial

The "LBP_Powerride" function block integrates a rotary/push control dial (SINUMERIK Powerride) into the SINUMERIK environment.

With the Powerride, the operator can change the feed rate in percent with immediate effect on the machine. To do this, multiply the override factor of the Powerride by the programmed feed rates.

Interfaces for the feed rate change:

- Individually in the channel DB
- Individually in the axis DB

The hardware of the Powerride is described in the Operating Components Manual Modular MCPs.

Querying and adjusting properties of the Powerride

At the parameters of the FB and in the data records of the DB "LBP_PowerrideConfig" you can query data and adjust the behavior of the Powerride.

Summary of the query and setting options:

- Query current switch position (between 1 and 36)
  "switchPos" parameter
- Query override factor for the current switch position
  "ovrFactor" and "grayCode" parameters
- Define override factor for the respective switch position
  "configuration.switchPos[x].value" parameter
- Evaluate whether switch is pressed
  "pushButton" parameter

- Colors and limit values for controlling the LEDs
  "pushButtonLED" and "configuration.switchPos[x].ledColor" parameters
- Vibration pattern
  "configuration.switchPos[x].vibrationIndex" parameter
- Change value for the current switch position (between 1 and 36)
  "preset" or "forceSwitchPos1" parameter
- Set switch positions for 100 % and the end position
  "configuration.pos100" and "configuration.posMax" parameters
- Time delay for exceeding 100 %
  "configuration.delay100" parameter

Using default settings or configuring individually

Ready-to-use configurations in the "LBP_PowerrideConfig" data block permit you to use the Powerride immediately. Transfer the desired configuration to the FB at the "configuration" parameter. Create your own data block with your configuration settings for individual adaptations.

If you want to program the behavior of the Powerride yourself, you can query and control the signals directly with the parameters "deviceIn" and "deviceOut".

Behavior of the Powerride

The following list briefly describes the behavior of the Powerride using the example of the configuration "LBP_PowerrideConfig.configFeed":

- An edge transition 0 → 1 of the parameter "enable" triggers the configuration check. An error-free configuration is copied into the instance DB.
- Turning clockwise increases the values of the parameters "switchPos", "grayCode" and "ovrFactor".
- Turning counterclockwise decreases the values of the parameters "switchPos", "grayCode" and "ovrFactor".
- The value in the "switchPos" parameter is never less than 1 or higher than "configuration.posMax".
- When counterclockwise rotation reaches position 1, no further reduction takes place. Vibration index 1 becomes noticeable.
- Between position 1 and "configuration.pos100", the LED strip lights up white as far as the current position.
- The number of activated LEDs corresponds to the value in the "switchPos" parameter.
- When clockwise rotation reaches position "configuration.pos100", the LED strip turns green, the position value is recorded, and vibration index 2 becomes noticeable.
- An increase of the value of 100 % is possible again after a time delay "configuration.delay100".

- Between position "configuration.pos100" and "configuration.posMax", the LED color is yellow.
- When clockwise rotation reaches position "configuration.posMax", the position value is recorded. Vibration index 3 becomes noticeable and the LED strip lights up red.

Description of formal parameters

Parameter	Category	Type	Value range	Meaning
enable	I	BOOL	---	Edit Powerride function Only set the parameter to TRUE after Power On (control startup) has been completed. **TRUE** Configuration data are checked and processing of the FBs is started if the check is successful. With Power On, the last "switchPos" becomes active if the instance DB is retentive. **FALSE** No processing of the FB; output values are retained.
preset	I	BOOL	---	Change "switchPos" On the positive edge, "presetValue" is accepted as the current "switchPos". The "presetValue" parameterization must be valid for this. The displays in "grayCode" and "ovrFactor" are updated.
forceSwitchPos1	I	BOOL	---	Set switch to position 1 On the positive edge, the switch is set to position 1 and the override factor "configuration.switchPos[1].value" is output.
pushButtonLED	I	USINT	0 ... 7	Trigger LED of the switch See LED color palette (Page 653)

Parameter	Category	Type	Value range	Meaning
presetValue	I	USINT	1 ... 36	New value for "switchPos" With a positive edge of "preset", the value is accepted as the new "switchPos". **Special case 1** "presetValue" is between "configuration.pos100" and "configuration.posMax" and "switchPos" (current position) is less than "configuration.pos100" For safety reasons, in this case the value of "configuration.pos100" is accepted with "preset". **Special case 2** "presetValue" is greater than "configuration.posMax" and less than 36 With "preset", the value of "configuration.posMax" is accepted.
valid	O	BOOL	---	Display whether the block is being processed and the configuration is valid TRUE if "enable" = TRUE and "error" = FALSE if "enable" = FALSE or "error" = TRUE
error	O	BOOL	---	Indication whether an error is present The cause of the error is displayed in the parameter "status". TRUE if an error occurs, in which case the other outputs are reset FALSE if "enable" = FALSE or "valid" = TRUE
pushButton	O	BOOL	---	Indication whether the switch is pressed The output is supplied if "valid" = TRUE.
ovrFactor[1)]	O	USINT	0 ... 200	Display of the override factor as integer The value parameterized under "configuration.switchPos[1..36].value" is output for the current switch position. The output is supplied if "valid" = TRUE.
grayCode[1)]	O	USINT	---	Display of the override factor as Gray code The value parameterized under "configuration.switchPos[1..36].value" is output for the current switch position. The output is supplied if "valid" = TRUE.
switchPos	O	USINT	1 ... configuration.posMax	Display of the current switch position of the Powerride The output is supplied if "valid" = TRUE.
status	O	WORD	---	Status and error IDs (Page 657)

Parameter	Category	Type	Value range	Meaning
deviceIn	I/O	LBP_type-PowerrideInput (Page 550)	---	Signals to the PLC See Signal transfer between Powerride and PLC (Page 653)
deviceOut	I/O	LBP_type-PowerrideOutput (Page 551)	---	Signals from the PLC to the Powerride See Signal transfer between Powerride and PLC (Page 653)
configuration	I/O	LBP_type-PowerrideConfig (Page 553)	---	Configuration data

1) You set the interpretation of the override factor in the machine date (e.g.: MD12020 $MN_OVR_FEED_IS_GRAY_CODE).

See also

LBP_PowerrideConfig [DB78] - preconfigured behavior for Powerride (Page 547)

10.6.5.1 Signal transfer between Powerride and PLC

The signals of the Powerride are transferred via a USB interface. The PLC is connected via an MCP or its COM board.

Signals to the PLC

The following table lists signals that are transferred from the Powerride to the PLC:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	-	-	Switch pressed = TRUE
1	Position of the rotary wheel [0..255] (modulo, 256 notches at 36 notches per revolution)							

Signals to the Powerride

The following table lists signals that are transferred from the PLC to the Powerride:

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	-	-	-	-	-	-	-	LED strip[1] 24
1	LED strip[1] 23	LED strip[1] 22	LED strip[1] 21	LED strip[1] 20	LED strip[1] 19	LED strip[1] 18	LED strip[1] 17	LED strip[1] 16
2	LED strip[1] 15	LED strip[1] 14	LED strip[1] 13	LED strip[1] 12	LED strip[1] 11	LED strip[1] 10	LED strip[1] 9	LED strip[1] 8
3	LED strip[1] 7	LED strip[1] 6	LED strip[1] 5	LED strip[1] 4	LED strip[1] 3	LED strip[1] 2	LED strip[1] 1	LED strip[1] 0
4	-	-	Color of the switch LED (color combinations (Page 653) possible)			Color of the LED strip (color combinations (Page 653) possible)		
			Blue	Green	Red	Blue	Green	Red
5	-	-	-	-	-	Vibration pattern [0..7] (at 0 no vibration)		

1) With TRUE the respective LED of the LED strip is lit

10.6.5.2 LED color palette

The following table shows the possible color combinations depending on the individual colors blue, green and red:

Combined color	Blue (bit 2)	Green (bit 1)	Red (bit 0)
-	-	-	-
Red	-	-	x
Green	-	x	-
Yellow	-	x	x
Blue	x	-	-

Combined color	Blue (bit 2)	Green (bit 1)	Red (bit 0)
Magenta	x	-	x
Cyan	x	x	-
White	x	x	x

10.6.5.3 Examples

Integrating Powerride into the startup using an MCP

In the example the Powerride is a hardware component of a modular MCP. This MCP is used as second MCP. The global data block "ModularMcpData" is available for the input and output image. The block is not included in the library. Using "ModularMCPData" as an example, it is shown how the input and output image can be mapped for a modular MCP using data types supplied in the library, and how this DB can be used.

ModularMCPData

	Name	Data type	Offset	Start value	Retain	Comment
1	Static					
2	in	Struct	0.0			
3	base	"LBP_typeModularMcpBaseInput"	0.0			Input image of base signals for modular MCP
4	keypad1	"LBP_typeModularMcpUniversalInput"	6.0			Input image of universal keypad for modular MCP
5	keypad2	"LBP_typeModularMcpUniversalInput"	10.0			Input image of universal keypad for modular MCP
6	keypad3	"LBP_typeModularMcpUniversalInput"	14.0			Input image of universal keypad for modular MCP
7	keypad4	"LBP_typeModularMcpUniversalInput"	18.0			Input image of universal keypad for modular MCP
8	powerride1	"LBP_typePowerrideInput"	22.0			Input image of powerride
9	powerride2	"LBP_typePowerrideInput"	24.0			Input image of powerride
10	powerride3	"LBP_typePowerrideInput"	26.0			Input image of powerride
11	handwheel1	"LBP_typeHandwheel"	28.0			
12	handwheel2	"LBP_typeHandwheel"	32.0			
13	out	Struct	36.0			
14	base	"LBP_typeModularMcpBaseOutput"	36.0			Output image of base signals for modular MCP
15	keypad1	"LBP_typeModularMcpUniversalOutput"	38.0			Output image of universal keypad for modular MCP
16	keypad2	"LBP_typeModularMcpUniversalOutput"	42.0			Output image of universal keypad for modular MCP
17	keypad3	"LBP_typeModularMcpUniversalOutput"	46.0			Output image of universal keypad for modular MCP
18	keypad4	"LBP_typeModularMcpUniversalOutput"	50.0			Output image of universal keypad for modular MCP
19	powerride1	"LBP_typePowerrideOutput"	54.0			Output image of powerride
20	powerride2	"LBP_typePowerrideOutput"	60.0			Output image of powerride
21	powerride3	"LBP_typePowerrideOutput"	66.0			Output image of powerride

The following table shows an example of the input and output addresses:

	Byte numbers of the data
Input data	"ModularMcpData".in.powerride_1 and "ModularMcpData".in.powerride_2
Output data	"ModularMcpData".out.powerride1 and "ModularMcpData".out.powerride2

Note

The addresses of the Powerride data in the input and output image of the MCP can change with the firmware version. The current address assignment is described in the Operating Components Manual Modular MCPs.

The code example shows the relevant part of the call of "LBP_ConfigBP [FC1]" in the startup OB, e.g. OB100:

```
"LBP_ConfigBP" MCPNum:=2,
 (
                MCP1In:=P#I0.0,
```

```
MCP1Out:=P#Q0.0,
MCP1BusAdr:=192,
MCP1Timeout:=S5T#700MS,
MCP2In:="ModularMcpData".in,
MCP2Out:="ModularMcpData".out,
MCP2BusAdr:=193,
MCP2Timeout:=S5T#700MS,
MCP1Stop:=FALSE,
MCP2Stop:=FALSE,
MCP1NotSend:=FALSE,
MCP2NotSend:=FALSE,
MCPBusType:=B#16#55,
…
```

Calling LBP_PowerRide

The example shows the call for "LBP_Powerride". The input and output image of the MCP must be assigned previously during startup and is described in example "Integrating Powerride into the startup using an MCP (Page 654)". The global data block "ModularMcpData" contains the input and output image in the example. You can also find information on this in the Operating Components Manual Modular MCPs.

The following code shows the call for "LBP_Powerride" in "Main [OB1]":

```
// #tempEnable: Bool;
// #tempPreset: Bool;
// #tempForceSwitchPos1: Bool;
// #tempPushButtonLED: USInt;
// #tempPresetValue: USInt;

// #tempValid: Bool;
// #tempError: Bool;
// #tempPushBotton: Bool;
// #tempOvrFactor: USInt;
// #tempGrayCode: USInt;
// #tempSwitchPos: USInt;
// #tempStatus: Word;

// "ModularMcpData".in.powerride_1: LBP_typePowerrideInput;
// "ModularMcpData".out.powerride_1: LBP_typePowerrideOutput;

"LBP_Powerride" (    enable := #tempEnable,
                     preset := #tempPreset,
                     forceSwitchPos1 := #tempForceSwitchPos1,
                     pushButtonLED := #tempPushButtonLED,
                     presetValue := #tempPresetValue,
                     valid => #tempValid,
                     error => #tempError,
                     pushButton => #tempPushButton,
                     ovrFactor => #tempOvrFactor,
                     grayCode => #tempGrayCode,
                     switchPos => #tempSwitchPos,
                     status => #tempStatus,
                     deviceIn := "ModularMcpData".in.powerride_1,
                     deviceOut := "ModularMcpData".out.powerride_1,
                     configuration := "LBP_PowerrideConfig".configFeed);
```

10.6.5.4 Status and error IDs

The following table lists the status and error IDs of the "LBP_Powerride".

Number	Type[1)]	Explanation
0000	-	No error
0007	M	Configuration is valid
7000	M	"enable" parameter is FALSE
7005	M	Settings are checked
7104	F	"pushButtonLED" parameter is outside the valid range (0 ... 7)
7105	M	"configuration.switchPos[x].ledColor" parameter is outside the valid range (0 ... 7) in one or more data records
7106	M	"configuration.switchPos[x].vibrationIndex" parameter is outside the valid range (0 ... 7) in one or more data records
8102	F	"presetValue" parameter is greater than 36
8103	F	"presetValue" parameter is less than 1
8212	F	Configuration error: "posMax" parameter is greater than 36
8221	F	Configuration error: "pos100" parameter is less than 1
8222	F	Configuration error: "posMax" parameter is less than 1
8223	F	Configuration error: "pos100" parameter is greater than "posMax"
8231	F	"configuration.switchPos[x].value" parameter is outside the valid range (0 ... 200%) in one or more data records

1) M = message, F = fault

10.6.6 LBP_OpUnitComm [FB25000]: Operator component communication

Function

Note

This function is not available in the virtual SINUMERIK!

The block is used for communication with operator components via Industrial Ethernet (IE), e.g. machine control panels (MCP), handheld units (HT) and direct key modules (OpKey).

The basic program library does not contain an instance DB for "LBP_OpUnitComm [FB25000]". Create an instance DB in the project in order to use the block.

The parameters for the configuration can be set in the startup OB (e.g. "Startup [OB100]") via "LBP_ConfigBP [FC1]". Additional factors or a changed configuration of the connections can be set in the cycle via the parameters in "LBP_ConfigData [DB7]".

The connections to the operator components are created automatically based on the configuration in "LBP_ConfigData [DB7]" and monitored. The block writes the input data received from the operator components to the configured pointers from "LBP_ConfigData [DB7]". The block reads the output data sent to the operator components from the configured pointers from "LBP_ConfigData [DB7]".

Status signals of the connections to the operator components are written to "LBP_NC [DB10]".

Alarms are generated when an operator component fails or a connection to the operator component is interrupted. The block also returns error information. Connection-specific error information is stored in parameters in "LBP_ConfigData [DB7]".

The operator components can be identified via the parameters of the Ident interface in "LBP_ConfigData [DB7]". In this way, the type of an operator component and the required length of the input and output data can be determined.

For details on the configuration and application, see "Configuration of the machine control panel, handheld unit, direct keys".

Cyclic call

The block must be called cyclically. The configuration of the connections to the operator components is in "LBP_ConfigData [DB7]", which is specified by "LBP_OpUnitComm [FB25000]" at the block call.

The call in the PLC cycle should be realized before calling "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]", "LBP_MCPCtrlTurning [FC25]", "LBP_HTCtrlHT8 [FC26]" and "LBP_HTCtrlHT2 [FC13]" - or before accessing the input data, for example, after calling "LBP_MainBP [FC2]". This means that actual operator component data is available for subsequent program parts.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Config	I/O	VARIANT	"LBP_ConfigData [DB7]"	Configuration DB "LBP_ConfigData [DB7]" of type "LBP_typeConfigData"
Error	O	BOOL	---	An error has occurred, see Status
Status	O	WORD	---	See Section "Status and errors"

Status and errors

Status	Description
16#0000	No error
16#8001	The variable on the parameter "Config" is not of the type "LBP_typeConfigData"
16#8002	A connection-specific error has occurred. See status of the connections in the configuration block on parameter "Config", usually "LBP_ConfigData [DB7]"
Connection-specific errors	
16#0000	No error
16#8101	Error when opening or establishing the communication connection
16#8201	Error when sending the identify request to the operator component
16#8202	Timeout when receiving the identify response from the operator component
16#8203	The operator components does not have an IPv4 address
16#8301	The parameterization data for this operator component could not be determined
16#8302	The parameterization data is invalid

Status	Description
16#8303	Error when sending the parameterization request to the operator component
16#8304	The parameterization has been rejected by the operator component
16#8305	Timeout when receiving the parameterization response from the operator component
16#8401	Timeout when receiving the data
16#8402	The pointer for the input data from the operator component is invalid
16#8403	The pointer for the output data to the operator component is invalid
16#8404	Error when storing the input data from the operator component
16#8405	Error when copying the output data to the operator component
16#8501	Error when disconnecting or closing the connection

Call example (SCL)

Call with instance DB "OpUnitComm_DB" and configuration DB "LBP_ConfigData [DB7]". The temporary variables "tempError" and "tempStatus" store the error information and can be evaluated later.

```
// #tempError: Bool;
// #tempStatus: Word;

OpUnitComm_DB(Config := LBP_ConfigData,
      Error => #tempError,
      Status => #tempStatus);
```

The assignment of the output parameters "Error" and "Status" is optional. The following example shows the call when the error information is not required.

```
OpUnitComm_DB(Config := LBP_ConfigData);
```

10.6.7 LBP_OpUnitWriteRecord [FB25002]: Writing a data record to MCP

Function

Note

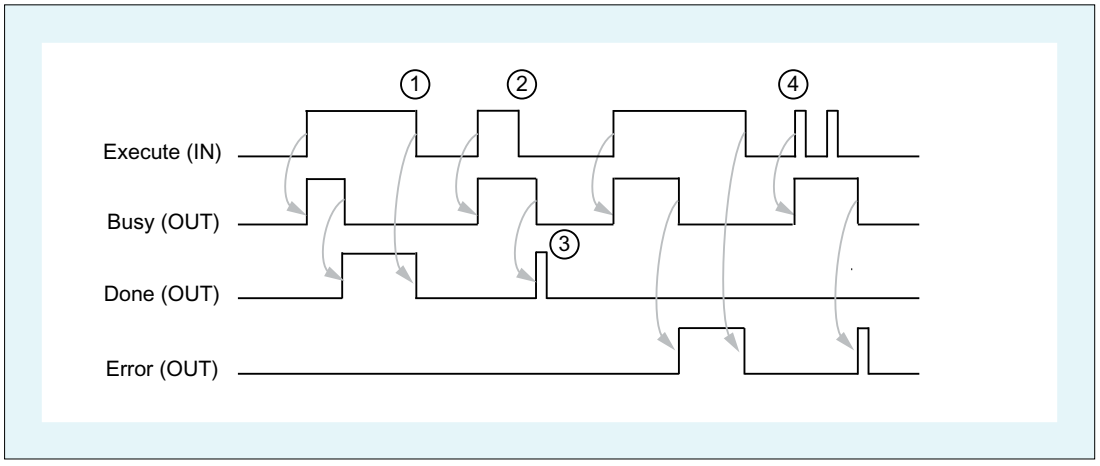
This function is not available in the virtual SINUMERIK!

The block writes a data record to a modular MCP (analog WR_REC for PROFINET devices).

Description of formal parameters

Parameter	Category	Type	Value range	Meaning
execute	I	BOOL	---	Write data record, see pulse diagram
mcpNumber	I	UINT	1 ... 2	Number of the MCP to be written to.
slotNumber	I	UINT	---	Slot of the MCP to which the transmitted data record relates.
recordNumber	I	UINT	---	Data record number of the data record to be written. 128 for configuration data of the modular MCP
busy	O	BOOL	---	Data record is written, see pulse diagram
done	O	BOOL	---	Writing completed, see pulse diagram
error	O	BOOL	---	Indication whether an error is present The cause of the error is displayed in the parameter "status". TRUE on error
status	O	WORD	---	See table Status and error IDs
recordData	I/O	VARIANT	---	Data record to be transmitted
opUnitComm	I/O	LBP_OpUnitComm	---	Instance of "LBP_OpUnitComm [FB25000]" Communication is performed via this connection to the MCP.

Pulse diagram



The job is started on a rising edge at parameter "execute". The values at the input parameters are applied and remain valid for the entire duration of the job.

The block must be called until it reports "done" or "error".

The job cannot be aborted.

When the job is over, a positive edge on "Execute" is required to start a new job.

Multiple, parallel calls with different instances of "LBP_OpUnitWriteRecord" can be started. The block processes the calls internally one after the other, not in parallel.

① "Done" and "error" are reset on "execute" with a falling edge.

② The functionality of the FB is not stopped with a falling edge on "execute". If the input "execute" is reset before output "done" is set, output "done" is only set for one cycle.

③ If "execute" is already FALSE, "done" and "error" are only active for one cycle.

④ A new block is requested with a rising edge on "execute" while the block is still being processed ("busy" = TRUE).
The old job is ended with the parameters active at the beginning of the job.

Application example: Setting the key color of a modular MCP

Colors that depend on the status can be assigned to the keys of a modular MCP. Possible statuses are active and inactive.

The data record passed to parameter "recordData" must be of type LBP_typeModularMcpKeypadRecord (Page 96).

To assign a color to the keys, follow these steps:

- Pass the value 128 at parameter "recordNumber".
- Configure the variables of the data record that you pass to parameter "recordData".
  "Version" must be set to 16#10
  Specify keys using the index referring to specific keypad
  Colors for the relevant key as RGB code (0 to 255)
- Specify the number of the keypad that you want to configure at parameter "slotNumber".
- Configure the remaining parameters of the FB.

A successfully transferred configuration remains active during operation of the MCP.

If more than 21 keys are to be configured, the configuration can be transferred in multiple calls with up to 21 keys.

Status and error IDs

The following table lists the status and error IDs of the block.

Number	Type[1)]	Explanation
16#0000	S	No error
16#7001	S	Waiting for start of job
16#7002	S	Job in progress
16#7003	S	Data transfer
16#8001	F	Invalid MCP number (parameter "mcpNumber")
16#8002	F	MCP is not configured (parameter "mcpNumber")
16#8003	F	MCP is not configured as IE (parameter "mcpNumber")
16#8004	F	Record data too long (parameter "recordData")
16#8005	F	Illegal record version
16#8006	F	Illegal record number (parameter "recordNumber")
16#8100 - 16#81FE	F	For "recordNumber" :=128: impermissible configuration for key The last two digits of the error number correspond to the impermissible key index.
16#81FF	F	Impermissible configuration
16#8401	F	MCP did not respond within the specified time
16#8404	F	Data record could not be serialized
16#8601	F	Internal error

1) S = Status, F = Fault or error

10.7 Functions (FC)

10.7.1 LBP_ConfigBP [FC1]: Basic program, startup section

10.7.1.1 STRUCT "LBP_typeTMMagazines" magazine data

The variable structure "LBP_typeTMMagazines" describes magazine data that can be set with "LBP_ConfigBP [FC1]".

Variables in "LBP_typeTMMagazines"

MaxSpindleNumber

Description	Maximum spindle number
Data type	UINT

NumberOfMagazines

Description	Number of all of the magazines, including the loading and buffer magazines
Data type	UINT

Magazines

Description	Magazine data
Data type	ARRAY[1..64] of STRUCT STRUCT "Magazines" magazine data (Page 663)

10.7.1.2 STRUCT "Magazines" magazine data

The variable structure "Magazines" describes magazine data.

Variables in "Magazines"

Number

Description	Magazine number
Data type	UINT

Type

Description	Magazine type
Data type	UINT

Places

Description	Number of locations in the magazine
Data type	UINT

TMNumber

Description	Tool management number
Data type	UINT

Function

"LBP_ConfigBP [FC1]" must only be called in the startup OB (see online Help for programming example).

The synchronization of the NC and the PLC is performed during startup. The data blocks for the NC/PLC user interface are initialized based on the NC configuration defined via machine data, and the basic program parameters are verified for plausibility. In the event of an error, function block "LBP_ConfigBP [FC1]" passes an error code to the diagnostic buffer and switches the PLC to the stop state.

"Restart" startup mode

The integrated PLC only supports the startup mode "Restart". After basic system initialization, the organization block "Startup" is always run through first, followed by "Main [OB1]".

Input parameters

The block interface is already assigned predefined actual parameters when "LBP_ConfigBP [FC1]" is inserted in the user program. To parameterize the basic program, only the relevant parameters of "LBP_ConfigBP [FC1]" have to be written with user-specific values. The input parameters of "LBP_ConfigBP [FC1]" are mirrored in the global data block "LBP_ConfigData [DB7]".

Output parameters

The output parameters of "LBP_ConfigBP [FC1]" are transferred to "LBP_ConfigData [DB7]" and can be read by the cyclic program section of the basic program:

1. Direct access in "LBP_ConfigData [DB7]" in symbolic form.
   Example: "LBP_ConfigData.MaxChan" (DB7.DBW168)

Note

For assigning the "LBP_ConfigBP [FC1]" parameter for MCP and HT, see "General information on configuration and commissioning (Page 751)".

Supply and access to parameters and further variables

The input parameters are copied from "LBP_ConfigBP [FC1]" to "LBP_ConfigData [DB7]". The output parameters should be transferred to the appropriate positions in "LBP_ConfigData [DB7]". This is already set as default setting in the predefined actual parameters. In addition to the input and output parameters, "LBP_ConfigData [DB7]" contains further variables that can only be accessed in cyclic operation ("Main"). The following table lists all parameters.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
MCPNum	I	INT	0, 1, 2	Number of active MCPs 0: No MCP available (default) 1: 1 MCP 2: 2 MCPs
MCP1In MCP2In	I	POINTER	I0.0 ... I120.0 or M0.0 ... M248.0 or DBn DBX0.0 ... DBXm.0	Start address for the input signals of the relevant machine control panel

Signal	Category	Type	Value range	Meaning
MCP1Out MCP2Out	I	POINTER	A0.0 ... A120.0 or M0.0 ... M248.0 or DBn DBX0.0 ... DBXm.0	Start address for the output signals of the relevant machine control panel
MCP1BusAdr	I	INT	1, 2, 3 ... 126	DP slave: PROFIBUS address
MCP2BusAdr			192, 193, 194 ... 223	Ethernet MCP: DIP setting
MCP1Stop MCP2Stop	I	BOOL	---	0: Start transfer of machine control panel signals 1: Stop transfer of machine control panel signals
				DP slave: Slave deactivated
MCP1NotSend MCP2NotSend	I	BOOL	---	0: Send and receive operation activated 1: Receive machine control panel signals only
MCPBusType	I	BYTE	3, 4, 5, 6	b#16#yx: • Bus type MCP1: lower nibble (x) • Bus type MCP2: upper nibble (y) 3: PROFIBUS 4: PROFIBUS on the MPI/DP port 5: Ethernet 6: PROFINET Mixed operation is possible (see Section "General information on configuration and commissioning (Page 751)")
HTIf	I	INT	0, 5	Handheld unit interface: 0: No HT 5: HT on Ethernet
HTIn	I	POINTER	I0.0 ... I124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Start address PLC receive data from the handheld unit
HTOut	I	POINTER	A0.0 ... A124.0 or M0.0 ... M252.0 or DBn DBX0.0 ... DBXm.0	Start address of the PLC send data to the handheld unit
HTAdr	I	INT	HT default: 2	Ethernet DIP switch
HTStop	I	BOOL	---	0: Start transmission of handheld unit signals 1: Stop transmission of handheld unit signals
HTNotSend	I	BOOL	---	0: Send and receive operation activated 1: Receive handheld unit signals only
NCCyclTimeout	I	S5time	Recommendation: 200 ms	Cyclic sign-of-life monitoring NCK

Signal	Category	Type	Value range	Meaning
NCRunupTimeout	I	S5time	Recommendation: 50 s	Power-up monitoring NCK
ListMDecGrp	I	INT	0, 1, 2 ... 16	Activation of expanded M group decoding 0: Not active 1 ... 16: Number of M groups
MMCToIF	I	BOOL	---	Transmission of HMI signals to interface (modes, program control, etc.) 1: Active
HWheelMMC	I	BOOL	---	0: Handwheel selection via HMI 1: Handwheel selection via user program
MCP_IF_TCS	I	BOOL	---	"LBP_MCPCtrlMilling [FC19]": Evaluation R11 key 0: No evaluation 1: R11 key acts as manual travel in tool orientation (see Section "Cartesian manual travel" in Section "LBP_MCPCtrlMilling [FC19]: Transfer of the MCP signals on the interface (Page 712)")
ExtendChanAxMsg	I	BOOL	---	Activation of all areas for error and system status messages of the "LBP_GenerateAlarmMsgs [FC10]" (not available on every SINUMERIK PLC)
HMIAxisSel	I	BOOL	---	SmartOperate/HT10 0: No evaluation of the axis selection from the HMI 1: Axis selection from the HMI possible Default value: 0
MsgUser	I	INT	0, 1, 2 ... 64	Number of user areas ("LBP_AlarmMsgs [DB2]", not available on every SINUMERIK PLC)
IRAuxfuT	I	BOOL	---	Evaluate T function in "Hardware interrupt [OB40]"
IRAuxfuH	I	BOOL	---	Evaluate H function in "Hardware interrupt"
IRAuxfuE	I	BOOL	---	Evaluate DL function in "Hardware interrupt"
UserVersion	I	POINTER	---	Pointer to string variable, which is displayed in the version display of the user interface The string variable has the following notation (max. 41 characters): "<**Name**> <version **xx.xx.xx**> <date **yy/mm/dd**>" Example: "Test version 07.06.02 13/06/04"
OpKeyNum	I	INT	0, 1, 2	Number of active direct control key modules 0: No Ethernet direct keys available
Op1KeyIn Op2KeyIn	I	POINTER	---	Start address for the input signals of the affected direct control key modules
Op1KeyOut Op2KeyOut	I	POINTER	---	Start address for the output signals of the affected direct control key modules
Op1KeyBusAdr Op2KeyBusAdr	I	INT	1, 2, 3 ... 254	Direct control keys via Ethernet: TCU index
Op1KeyStop Op2KeyStop	I	BOOL	---	0: Start transmission of direct control key signals 1: Stop transmission of direct control key signals

Signal	Category	Type	Value range	Meaning
Op1KeyNotSend Op2KeyNotSend	I	BOOL	---	0: Send and receive operation activated 1: Receive direct key signals only
OpUnitCommId	I	UINT	1, 2, 3 ...	ID for OpUnitComm (The parameter is not relevant for virtual Sinumerik.)
OpUnitInterfaceId	I	HW_ANY	---	Hardware ID of Ethernet interface for OpUnitComm (The parameter is not relevant for virtual Sinumerik.)
MCP1ConnectionId	I	CONN_OUC	1, 2, 3 ... 4095	Connection identifier for MCP1 (The parameter is not relevant for virtual Sinumerik.)
MCP2ConnectionId	I	CONN_OUC	1, 2, 3 ... 4095	Connection identifier for MCP2 (The parameter is not relevant for virtual Sinumerik.)
HTConnectionId	I	CONN_OUC	1, 2, 3 ... 4095	Connection identifier for HT (The parameter is not relevant for virtual Sinumerik.)
Op1KeyConnectionId	I	CONN_OUC	1, 2, 3 ... 4095	Connection identifier for direct keys 1 (The parameter is not relevant for virtual Sinumerik.)
Op2KeyConnectionId	I	CONN_OUC	1, 2, 3 ... 4095	Connection identifier for direct keys 2 (The parameter is not relevant for virtual Sinumerik.)
IdentConnectionId	I	CONN_OUC	1, 2, 3 ... 4095	Connection identifier for identification (The parameter is not relevant for virtual Sinumerik.)
MCP1LocalUdpPort	I	UINT	1, 2, 3 ... 49151	Local UDP port for MCP1 (The parameter is not relevant for virtual Sinumerik.)
MCP2LocalUdpPort	I	UINT	1, 2, 3 ... 49151	Local UDP port for MCP2 (The parameter is not relevant for virtual Sinumerik.)
HTLocalUdpPort	I	UINT	1, 2, 3 ... 49151	Local UDP port for handheld terminal (The parameter is not relevant for virtual Sinumerik.)
Op1KeyLocalUdpPort	I	UINT	1, 2, 3 ... 49151	Local UDP port for direct keys 1 (The parameter is not relevant for virtual Sinumerik.)
Op2KeyLocalUdpPort	I	UINT	1, 2, 3 ... 49151	Local UDP port for direct keys 2 (The parameter is not relevant for virtual Sinumerik.)
IdentLocalUdpPort	I	UINT	1, 2, 3 ... 49151	Local UDP port for identification (The parameter is not relevant for virtual Sinumerik.)
MaxModeGroup	O	INT	1, 2, 3 ... 10	Number of active mode groups

Signal	Category	Type	Value range	Meaning
MaxChan	O	INT	1, 2, 3 ... 10	Highest active channel number
MaxAxis	O	INT	1, 2, 3 ... 31	Highest active axis number
ActiveChan	O	ARRAY[1..10] of BOOL	---	Bit string for active channels
ActiveAxis	O	ARRAY[1..31] of BOOL	---	Bit string for active axes
MaxNumUserDataInt	O	INT	0, 1, 2, ... 256	Number of integer machine data
MaxNumUserDataHex	O	INT	0, 1, 2, ... 256	Number of hexadecimal machine data The parameter may contain a fill byte, if applicable. (If an uneven number of hexadecimal machine data has been defined, the next higher even value is displayed here.) See also information in PLC machine data (Page 669).
MaxNumUserDataReal	O	INT	0, 1, 2, ... 64	Number of real machine data
UserDataIntArray	O	VARIANT	---	Points to an array of type "INT" or structure in any user DB that holds the integer machine data. See also PLC machine data (Page 669). Default value: NULL
UserDataHexArray	O	VARIANT	---	Points to an array of type "BYTE" or structure in any user DB that holds the hexadecimal machine data. See also PLC machine data (Page 669). Default value: NULL
UserDataRealArray	O	VARIANT	---	Points to an array of type "REAL" or structure in any user DB that holds the real machine data. See also PLC machine data (Page 669). Default value: NULL
TMMagazines	O	VARIANT	Variable of the type STRUCT "LBP_typeTMMagazines" magazine data (Page 662)	Parameter assignment with NULL: The magazine data are not set
				Parameterization with variable: The magazine data are set This variable contains the magazine data. Default value: NULL

See also

LBP_ConfigData [DB7] (Page 121)

Switchover of machine control panel, handheld unit (Page 770)

Configuration machine control panel, handheld unit, direct keys (Page 751)

10.7.1.3 PLC machine data

The user can store PLC-specific machine data in the NC. This machine data can then be processed by the user during power-up of the PLC ("Startup [OB100]") in order, for example, to implement user options, machine extensions or machine configurations.

The interface to read this data is defined by the user and specified via the output parameters "UserDataIntArray", "UserDataHexArray", and "UserDataRealArray" when "LBP_ConfigBP [FC1]" is called.

Definition of the PLC machine data to be transferred

The PLC machine data to be transferred is defined on the NC side through the following general machine data:

- Number of possible PLC machine data:
  - MD14504 $MN_MAXNUM_USER_DATA_INT (number of user data of the type INT)
    Maximum value: 256
  - MD14506 $MN_MAXNUM_USER_DATA_HEX (number of user data of the type HEX)
    Maximum value: 256
  - MD14508 $MN_MAXNUM_USER_DATA_FLOAT (number of user data of the type FLOAT)
    Maximum value: 64
- Values of the PLC machine data to be transferred
  - MD14510[<n>] $MN_USER_DATA_INT (user data of the type INT)
  - MD14512[<n>] $MN_USER_DATA_HEX (user data of the type HEX)
  - MD14514[<n>] $MN_USER_DATA_FLOAT (user data of the type FLOAT)

Parameters for the transfer of the PLC machine data

An "Array of Int" must be specified to take the integer data, an "Array of Byte" to take the hexadecimal data and an "Array of Real" to take the real data. Alternatively, structures or PLC data types can be used.

The size of the array or structure that is used to hold all the available PLC machine data must at least correspond to the number of available data according to MD14504, MD14506 or MD14508 but must not be larger than the number of maximum settable PLC machine data.

If an array or a structure is smaller than the number of existing PLC machine data of the corresponding type, only some of the data are transferred (see also the note in the following example).

If the available array or structure is larger than the number of existing PLC machine data of the corresponding type, all of the data existing for that data type are transferred to the PLC. Data in the other additional elements in arrays or structures are not kept.

Note

The user decides into which data areas or variables the three data fields are transferred and whether the data are to be kept in shared or separate areas.

Note

The user machine data is always handled internally as an array. ARRAY variables occupy the memory in the PLC up to the next word limit, i.e. at a byte with an even address. Therefore, a fill byte is supplemented internally when there is an uneven number in MD14506 $MN_MAXNUM_USER_DATA_HEX. This fill byte is also counted in the initial parameter "LBP_ConfigData.MaxNumUserDataHex" (DB7.DBW180) (number of hexadecimal machine data) by "LBP_ConfigBP [FC1]".

Example

The project requires four integer values, two hexadecimal arrays with bit information and one real value.

Machine data:

MD14504 $MN_MAXNUM_USER_DATA_INT = 4
MD14506 $MN_MAXNUM_USER_DATA_HEX = 2
MD14508 $MN_MAXNUM_USER_DATA_FLOAT = 1
...
MD14510[0] $MN_USER_DATA_INT = 123
MD14510[1] $MN_USER_DATA_INT = 456
MD14510[2] $MN_USER_DATA_INT = 789
MD14510[3] $MN_USER_DATA_INT = 1011
...
MD14512[0] $MN_USER_DATA_HEX = 12
MD14512[1] $MN_USER_DATA_HEX = AC
...
MD14514[0] $MN_USER_DATA_FLOAT = 123.456

Three gloal DBs are created, for example, to take the data in the PLC project:

- "DB_IntData" contains a variable "UserDataInt" of the type "Array[0..255] of Int".
- "DB_HexData" contains a variable "UserDataHex" of the type "Array[0..255] of Byte".
- "DB_FloatData" contains a variable "UserDataReal" of the type "Array[0..63] of Real".

NOTICE
Incomplete transfer of the PLC machine data If an array is smaller than the number of available PLC machine data of the corresponding type, only part of the data is transferred. This would be the case, for example, if a variable "UserDataInt" of the type "Array[0..2] of Int" were contained in the example project "DB_IntData". The variable could then only take three integer values with the result that "DB_IntData".UserDataInt[3] = 1011 could not be transferred. Remedy: The size of the used array is arbitrary, but should at least correspond to the number of available data according to MD14504, MD14506 or MD14508. **Alignment when structures are used** Structures can contain filler bytes if the structure contains different data types (see Alignment of variables and data types). Data in filler bytes cannot be symbolically accessed. Avoid using structures with filler bytes.

When "LBP_ConfigBP [FC1]" is called in "Startup", the corresponding output parameters must be linked to the arrays in these global DBs:

```
UserDataIntArray => "DB_IntData".UserDataInt,
UserDataHexArray => "DB_HexData".UserDataHex,
UserDataRealArray => "DB_FloatData".UserDataReal,
```

The three global DBs are then written with the corresponding values during system run-up:

"DB_IntData".UserDataInt[0] = 123
"DB_IntData".UserDataInt[1] = 456
"DB_IntData".UserDataInt[2] = 789
"DB_IntData".UserDataInt[3] = 1011
...
"DB_HexData".UserDataHex[0] = 12
"DB_HexData".UserDataHex[1] = AC
...
"DB_FloatData".UserDataReal[0] = 123.456

10.7.2 LBP_MainBP [FC2]: Basic program, cyclic section

Function

The NC/PLC interface is processed by the basic program in cyclic mode ("Main [OB1]"). You must call up this cyclic part in the cycle OB (see Online help for programming example).

To keep the runtime to a minimum, only the control and status signals are cyclically transferred. The transfer of auxiliary and G commands is processed only on request from the NC.

Furthermore, the data for handwheel selection, modes, and other operating signals are transferred from the operator panel (HMI) to the NC/PLC interface so that the modes support the selection from the MCP or HMI as required.

The transfer of HMI signals to the NC/PLC interface can be deactivated by setting the value of the parameter "MMCToIF" to "FALSE" when "LBP_ConfigBP [FC1]" in "Startup [OB100]" ("LBP_ConfigData [DB7]") is called.

Handwheel selection signals

Requirement: "LBP_ConfigBP [FC1]", parameter "HWheelMMC == "TRUE"

The handwheel selection signals from the HMI are decoded and activated in the interface in the respective machine or geometry axis of the respective handwheel.

Description of formal parameters

The function has no parameters.

10.7.3 LBP_NCKProcessIRT [FC3]: Basic program, interrupt-driven section

Function

Block-synchronized transfers from the NCK to the PLC (auxiliary and G commands) are processed in the interrupt-driven part of the basic program. You must configure this alarm-controlled part in the properties of the NCK (see Online help for programming example) and insert the call in a hardware interrupt OB.

Auxiliary functions are subdivided into normal and highspeed auxiliary functions.

The highspeed functions of an NC block are buffered and the transfer acknowledged to the NC. These are transferred to the user interface at the start of the next "Main [OB1]" cycle.

Highspeed auxiliary functions programmed immediately one after the other, are not lost for the user program. This is ensured by a mechanism in the basic program.

Normal auxiliary functions are acknowledged to the NC only after one completed cycle duration. This allows the user to issue a read-in disable to the NC.

The **G commands** are evaluated immediately and passed to the user interface.

The function provides a series of output parameters that can control the further processing in the hardware interrupt OB. To do this, the parameter "Flags" must be linked to a variable of the type "LBP_typeNCKProzessIRTFlags". The meaning of the available flags is described in the following sections.

NC hardware interrupt

The bit IRFromNCK is set in the output flags when the interrupt is triggered by the NC. To be able to implement highspeed, jobcontrolled processing of the user program for the machine, the following NC functions are available in the hardware interrupt processing routine (process alarm OBs 40 to 47, or 123 to 32767) for the PLC user program:

- Selected **auxiliary functions** (T, H and DL functions)
- **Tool-change function** for tool-management option
- **Position reached** for positioning axes, indexing axes and spindles with activation via PLC

The functions listed above must be evaluated by the user program in a hardware interrupt OB, e.g. "Hardware interrupt [OB40]", in order to initiate high-speed reactions on the machine. As an example, the turret switching mechanism can be activated when a T command is programmed on a turning machine.

Auxiliary functions

Generally, high-speed or acknowledging auxiliary functions are processed with or without interrupt control independently of any assignment.

The parameters IRAuxfuT, IRAuxfuH and IRAuxfuE in "LBP_ConfigData [DB7]" can be set to select which auxiliary functions (T, H, DL) must be processed solely on an interruptdriven basis by the user program.

Functions which are not assigned via interrupts are only made available by the cyclic basic program as in earlier versions. The change signals of these functions are available in a PLC cycle.

Even if the selection for the auxiliary function groups (T, H, DL) is made using interrupt control, only one interrupt can be processed by the user program for the selected functions.

If an interrupt is triggered by an auxiliary function, the corresponding bit is set in the flag "AuxFunction". For example, the set bit AuxFunction[1] means that an auxiliary function is available for the 1st channel.

In the related channel-DB the change signal and the function value are available for the user. The change signal of this interrupt-driven function is reset to zero in the cyclic basic program part after execution of at least one full "Main" cycle (max. approx. two "Main" cycles).

Tool change

With the tool management option, the tool change command for turret and the tool change in the spindle is supported by an interrupt. For this purpose, the bit TM is set in the output flags of "LBP_NCKProcessIRT [FC3]". The PLC user program can thus check the tool management DB ("LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]") for the tool change function and initiate the tool change operation.

Position reached

If the function "LBP_CtrlAxisSpindle [FC18]" (spindle control, positioning axis, indexing axis) has been activated for an axis or spindle, an instantaneous evaluation of the "InPos" signal of this "LBP_CtrlAxisSpindle [FC18]" can be implemented. The corresponding bit is set in the flag "InPosition". For example, the set bit InPosition[1] means that the axis or spindle 1 has reached its position. This feature can be used, for example, to obtain immediate activation of clamps for an indexing axis.

Description of formal parameters

Signal	Category	Type	Value range	Structure	Data type	Meaning
Flags	A	VARIANT	NULL	...	---	Flags are not set.
			LBP_typeNCKProcessIRTFlags	...		Flags are set for further evaluation in the hardware interrupt OB
				IRFromNck	BOOL	Interrupt by the NC
				TM	BOOL	Tool change command is pending
				InPosition	ARRAY[1..31] of BOOL	Axis has reached position (see "LBP_CtrlAxisSpindle [FC18]")
				Auxfunction	ARRAY[1..10] of BOOL	Channel in which the auxiliary function is available

10.7.4 LBP_TMTransferMT [FC6]: Transfer block for tool management and multitool

Function

The block acknowledges a tool change request from "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]". The block covers the same functionality as "LBP_TMTransfer [FC8]" and can also communicate the positioning of a multitool on the spindle.

Positioning of the multitool is only possible and sensible if the tool is located on a spindle. The multitool position is therefore only specified with the tool change command for the "new" tool. In all other cases, MtoolPlaceNum := 0 is used.

See also LBP_TMTransfer [FC8]: Acknowledge tool change (Page 678)

The additional function for multitools is described below.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Start	I	BOOL	---	TRUE: Start acknowledgement
TaskIdent	I	BYTE	1 ... 5	Identifier of the interface or job 1: Loading/unloading location 2: Spindle change position 3: Turret change position 4: Asynchronous transfer 5: Asynchronous transfer with location reservation
TaskIdentNo	I	BYTE	1 ... 16	TaskIdent := 1, 2, 3: associated interface 1 ... 16 TaskIdent := 4, 5: Channel number 1 ... 10
NewToolMag	I	INT	-1 0, 1, 2 ...	Current magazine number of new tool -1: Tool remains at its location
NewToolLoc	I	INT	-1 0, 1, 2 ...	Current location number of new tool -1: Tool remains at its location
OldToolMag	I	INT	-1 0, 1, 2 ...	Current magazine number of tool to be replaced -1: Tool remains at its location
OldToolLoc	I	INT	-1 0, 1, 2 ...	Current location number of tool to be replaced -1: Tool remains at its location
Status	I	INT	1 ... 7, 10 103, 104, 105	see "Status of job acknowledgement" 1: Job successfully completed 2: Reject new tool 3: Error 4: Suggestion of target position of old tool 5: Tool is positioned 6: Job completed, with location reservation 7: Repeat prepare tool change 10: Job completed without magazine movement 103: Accept new tool 104: Tool is positioned 105: Communicate intermediate position
MultitoolLoc	I	INT	-1 0, 1, 2 ...	Selected position in multitool for new tool 0: Position not relevant or no multitool -1: No change of position

Signal	Category	Type	Value range	Meaning
Ready	O	BOOL	---	TRUE: Acknowledgement completed
Error	O	INT		0: No error has occurred 1: "TaskIdent" is not permitted 2: Interface is not permitted ("TaskIdentNo") 3: Interface is not active ("TaskIdentNo") otherwise: Error identifier of NCK

10.7.5 LBP_TMTransferTurret [FC7]: Acknowledge tool change with turret

Function

The block acknowledges a tool change request from "LBP_TMTurret [DB73]".

The block must be called after tool change is completed for a turret with Start := TRUE. Parameter ChgdRevNo specifies the interface from "LBP_TMTurret [DB73]" via which the request for a tool change is output.

After a successful call (Ready == TRUE), the interface in "LBP_TMTurret [DB73]" is deactivated. The block does not have to be called again or the parameter must be set to Start := FALSE.

If an error occurs (Error == TRUE), the block must be called again with Start := TRUE.

Note

The "Start" parameter does not need a signal edge for a subsequent job. After "Ready" == TRUE, the block can be called again with "Start" := TRUE.

General conditions

- The block must only be started with parameter Start := TRUE if there is a request for tool change with activated interface in "LBP_TMTurret [DB73]".

References

- For detailed information about tool management, see Function Manual, Tool Management.

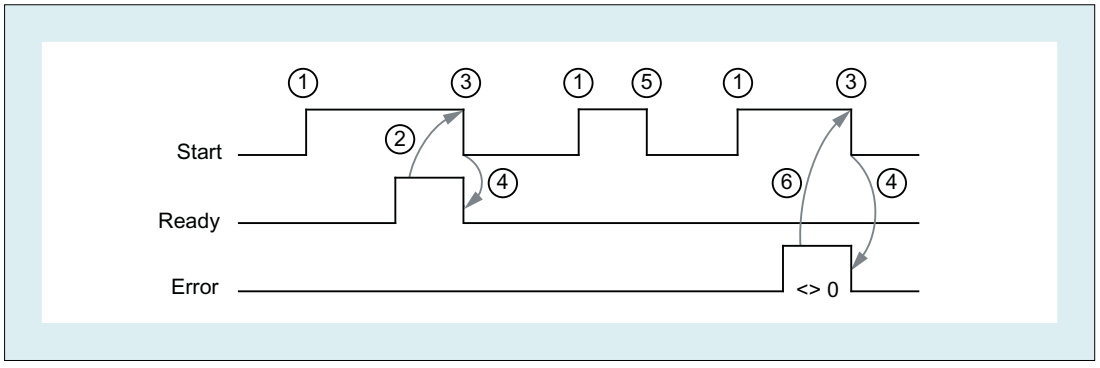
Manual turret switching

If the turret is rotated in manual operation, neither a tool change nor an offset selection is associated with this operation. The first step is the removal of the tool from the tool holder back to its location in the turret. An asynchronous transfer (TaskIdent := 4) can be used for this. If the turret is now turned to any position, the tool must be activated at this location. This is most easily achieved by T programming in the part program. Alternatively, an asynchronous subprogram can be started by transferring the turret position. The tool can then be activated at this location in the asynchronous subprogram.

Description of formal parameters

Signal	Cate-gory	Type	Value range	Meaning
Start	I	BOOL	---	TRUE: Start acknowledgement
ChgdRevNo	I	BYTE	1 ... 16	Number of the turret interface from "LBP_TMTurret [DB73]"
Ready	O	BOOL	---	TRUE: Acknowledgement completed
Error	O	INT	0 ... 3	0: No error has occurred 1: No turret present 2: Turret interface is not permitted 3: Turret interface is not active

Pulse diagram



① Requirement: Start := TRUE
② Positive feedback signal: Tool change acknowledged
③ Reset request: Start := FALSE
④ Feedback signal is reset
⑤ Impermissible reset of start
⑥ Negative feedback signal: Tool change acknowledgment rejected

Call example

```
// #statFC7Start : Bool
// #tempReady : Bool
// #tempError : Int
IF #statFC7Start THEN
    LBP_TMTransferTurret(   Start := TRUE,
                            ChgdRevNo := 1,  // Interface 1 from
                            LBP_TMTurret
                            Ready => #tempReady,
                            Error => #tempError);
    IF #tempReady OR #tempError <> 0 THEN
```

```
            #statFC7Start := FALSE;
        END_IF;
    END_IF;
```

See also

LBP_TMTransfer [FC8]: Acknowledge tool change (Page 678)

LBP_TMTransferMT [FC6]: Transfer block for tool management and multitool (Page 674)

10.7.6 LBP_TMTransfer [FC8]: Acknowledge tool change

Function

The block acknowledges a tool change request from "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]".

- Loading locations
- Spindle change positions
- Turret change positions
- Asynchronous transfer (with/without location reservation)

The block must be called after tool change is completed with Start := TRUE. Parameters TaskIdent and TaskIdentNo specify the interface from "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]" via which the request for a tool change is output.

The current tool positions are specified via the parameters NewToolMag, NewToolLoc, OldToolMag, and OldToolLoc. The block transfers the current positions to the NCK.

If a tool is changed without a socalled "old tool" (e.g. on loading), parameters "OldToolMag" and "OldToolLoc" must be assigned the value 0.

If a tool is changed without a socalled "new tool" (e.g. on loading), parameters "NewToolMag" and "NewToolLoc" must be assigned the value 0.

After a successful call (Ready == TRUE), the block no longer has to be called or the parameter must be set to Start := FALSE. If a value less than 100 is used for the parameter status, the interface in "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]" is deactivated.

If an error occurs (Error == TRUE), the block must be called again with Start := TRUE.

Note

The "Start" parameter does not need a signal edge for a subsequent job. After "Ready" == TRUE, the block can be called again with "Start" := TRUE.

Supplementary condition

The block must only be started with parameter Start := TRUE if there is a request for tool change with activated interface "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]". An exception are asynchronous transfers.

Asynchronous transfer

To ensure that position changes of a tool are independently signaled from the PLC to the tool management (e.g. independent position changes by the PLC), the block is called with "TaskIdent" := 4. For this call, no interface has to be activated in "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]".

The NC channel is specified via parameter "TaskIdentNo".

The previous location of the tool is specified in the parameters "OldToolMag" and "OldToolLoc". The current location of the tool is specified in the parameters "NewToolMag", "NewToolLoc". "Status" := 1 specified.

An asynchronous transfer should also be used after a magazine location has been positioned against a load location or a buffer. In this way, the tool management is told where a particular location of the magazine is located. The tool remains at the position in the magazine ("NewToolMag", "NewToolLoc"). This location is positioned against the load location or buffer ("OldToolMag", "OldToolLoc"). Status := 5 is used for this.

Asynchronous transfer with location reservation

If parameter "TaskIdent" := 5, in addition to changing the position, the tool management also reserves the location. The location is only reserved if the tool has been transported from a real magazine to a buffer storage.

References

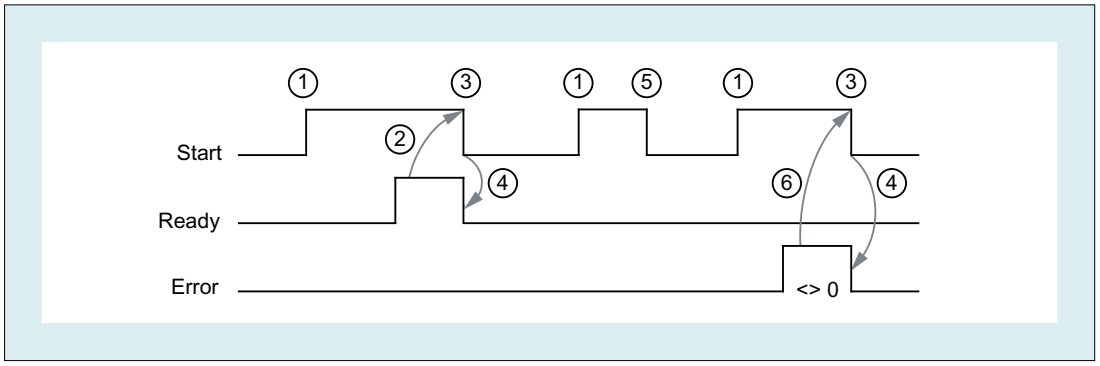
For detailed information about tool management, see Function Manual, Tool Management.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Start	I	BOOL	---	TRUE: Start acknowledgement
TaskIdent	I	BYTE	1 ... 5	Identifier of the interface or job 1: Loading/unloading location 2: Spindle change position 3: Turret change position 4: Asynchronous transfer 5: Asynchronous transfer with location reservation
TaskIdentNo	I	BYTE	1 ... 16	TaskIdent := 1, 2, 3: associated interface 1 ... 16 TaskIdent := 4, 5: Channel number 1 ... 10
NewToolMag	I	INT	-1 0, 1, 2 ...	Current magazine number of new tool -1: Tool remains at its location

Signal	Category	Type	Value range	Meaning
NewToolLoc	I	INT	-1 0, 1, 2 ...	Current location number of the new tool -1: Tool remains at its location
OldToolMag	I	INT	-1 0, 1, 2 ...	Current magazine number of tool that is to be replaced -1: Tool remains at its location
OldToolLoc	I	INT	-1 0, 1, 2 ...	Current location number of tool to be replaced -1: Tool remains at its location
Status	I	INT	1 ... 7, 10 102 ... 105	see "Status of job acknowledgement" 1: Job successfully completed 2: Reject new tool 3: Error 4: Suggestion of target position of old tool 5: Tool is positioned 6: Job completed, with location reservation 7: Repeat prepare tool change 10: Job completed without magazine movement 103: Accept new tool 104: Tool is positioned 105: Communicate intermediate position
Ready	O	BOOL	---	TRUE: Acknowledgement completed
Error	O	INT		0: No error has occurred 1: "TaskIdent" is not permitted 2: Interface is not permitted ("TaskIdentNo") 3: Interface is not active ("TaskIdentNo") otherwise: Error identifier of NC

Pulse diagram



① Requirement: Start := TRUE
② Positive feedback signal: Tool change acknowledged
③ Reset request: Start := FALSE
④ Feedback signal is reset
⑤ Impermissible reset of start
⑥ Negative feedback signal: Tool change acknowledgment rejected

Status of job acknowledgement

Status	Description
1	**Job successfully completed** Only permissible for: Loading, unloading, relocating, changing, preparing, changing The actual positions ("NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc") of the tools must be specified. The positions correspond to the target positions from the request from "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]". With **Prepare change**, the signaled positions may deviate from the target positions. • **Load, Unload, Relocate** The tool has arrived at the requested target address. • **Prepare change** The new tool is now ready. The tool may, for example, be positioned in a buffer (gripper). Possibly, the target (magazine, location) of the old tool was also moved to the tool change position after the new tool was placed in a buffer. However, the old tool still remains in the spindle. After this acknowledgement, the **Change** command can be received. • **Change** The tools have arrived at the requested target addresses. Status := 5 must be used for a request to **position**.
2	**The "new" tool cannot be made available** Only permissible for: Prepare change As a result, tool change of the suggested tool is prevented. The new tool is disabled in the NCK. The positions ("NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc") correspond to the original positions of the tools. Then the request is repeated with the same tool. For this, a new tool search is performed. This can only be acknowledged with Status := 2 if no intermediate acknowledgment has been output.
3	**An error has occurred** The tool positions must not be changed. If the positions of the tools have changed in the meantime, the positions must, e.g. first be communicated by calling the block with status := 105 so that the positions are taken into account.
4	**Suggestion for target position of old tool** Only permissible for: **Prepare change** A new target position ("OldToolMag", "OldToolLoc") for the "old" tool is suggested, the magazine location must be free. The positions for the "new" tool ("NewToolMag", "NewToolLoc") are ignored. Then a new tool change request is output that takes account of the suggested positions. There is no new tool search, the positions of the "new" tool ("NewToolMag", "NewToolLoc") are taken over from the original request.

Status	Description
5	**Tool is positioned** Only permissible for: loading, unloading, relocating, positioning. preparing tool change The "new" tool has reached the specified position ("NewToolMag", "NewToolLoc"). The specified tool is not really in this position but still in the magazine location. However, this magazine location has been moved to this position (e.g. load position or spindle). The status enables the tool management function to adjust the current position of a magazine and to improve the search strategy for subsequent commands. • **Loading, reloading**: On loading or reloading, a location for the tool is already reserved in the NCK. The machine operator must then insert the tool at the target location. Notice: The location reservation is canceled when the control system is switched on again. • **Positioning** A request to position is always acknowledged with Status := 5, Status := 1 is not permitted. • **Tool change preparation**: Outstanding tool movements are not executed until the tool has been changed. The old tool is still in the spindle ("OldToolMag" and "OldToolLoc").
6	**The job has been completed successfully, location reservation for "new" tool** Only permissible for: Relocating This status has the same function as Status := 1. In addition, the original location is reserved in case the target location is in a buffer magazine.
7	**Repetition of "Prepare Tool" request** Only permissible for: **Prepare change** This status is intended for use if the "new" tool has changed its position (e.g. due to an asynchronous command for the "new" tool). The positions ("NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc") correspond to the original positions of the tools. Then the request is repeated with the same tool. For this, a new tool search is performed.
10	**The tool change request was completed successfully without movement of the magazine** Only permissible for: Load, Unload, Relocate This status has the same function as Status := 1. However, the magazine involved must not be moved, the positioning of the magazine has not changed.

Intermediate acknowledgement

If one of the following values is used for the parameter status, the tool change request remains active, the interface in "LBP_TMLoad [DB71]", "LBP_TMSpindle [DB72]" or "LBP_TMTurret [DB73]" is not activated. Further acknowledgements are needed.

Status	Description
103	**The "new" tool can be inserted** Only permissible for: Prepare change This status is only permitted if the PLC is allowed to reject the new tool (see MD20310 Bit 4). The tool positions have remained unchanged. Preprocessing in the NCK can then continue without having to stop machining unnecessarily.
104	**The "new" tool is positioned** Only permissible for: Prepare change The "new" tool has reached the specified position ("NewToolMag", "NewToolLoc"). The specified tool is not really in this position but still in the magazine location. However, this magazine location has been moved to this position (e.g. load position or spindle). The status enables the tool management function to adjust the current position of a magazine and to improve the search strategy for subsequent commands. The "old" tool is still in the spindle ("OldToolMag" and "OldToolLoc").
105	**Communicate intermediate positions of the tools** The tools are located in the specified positions ("NewToolMag", "NewToolLoc", "OldToolMag", "OldToolLoc"). With this status, the intermediate tool positions reached are communicated if the job has not yet been completed.

Call examples

Loading

Acknowledge job to load a tool in magazine 1, location 3.

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
              TaskIdent := 1,      // Job from "LBP_TMLoad"
              TaskIdentNo := 1,    // Interface 1 in "LBP_TMLoad"
              NewToolMag := 1,     // Load to magazine 1, location 3
              NewToolLoc := 3,
              OldToolMag := 0,     // No old tool during loading
              OldToolLoc := 0,
              Status := 1,         // Job completed
              Ready =>
              #tempReady,
              Error =>
              #tempError);
```

Unloading

Acknowledge job to unload a tool via load location 1.

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
               TaskIdent := 1,      // Job from "LBP_TMLoad"
               TaskIdentNo := 1,    // Interface 1 in "LBP_TMLoad"
               NewToolMag := 0,     // No new tool during unloading
               NewToolLoc := 0,
               OldToolMag := 9999, // Loading magazine, load location 1
               OldToolLoc := 1,
               Status := 1,         // Job completed
               Ready =>
               #tempReady,
               Error =>
               #tempError);
```

Positioning

Acknowledge job to position a magazine at load location 1.

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
               TaskIdent := 1,      // Job from "LBP_TMLoad"
               TaskIdentNo := 1,    // Interface 1 in "LBP_TMLoad"
               NewToolMag := 9999, // Position to load location 1
               NewToolLoc := 1,
               OldToolMag := 0,     // No old tool during positioning
               OldToolLoc := 0,
               Status := 5,         // Positioning
               Ready =>
               #tempReady,
               Error =>
               #tempError);
```

Tool change preparation and tool change

Acknowledge job to prepare change.

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
               TaskIdent := 2,      // Job from "LBP_TMSpindle"
               TaskIdentNo := 1,    // Interface 1 in "LBP_TMSpindle"
```

```
                NewToolMag := 9998, // New tool to change position 1
                NewToolLoc := 1,
                OldToolMag := 9998, // Old tool still in spindle
                OldToolLoc := 1,
                Status := 5,        // Positioning
                Ready =>
                #tempReady,
                Error =>
                #tempError);
```

Acknowledge job to change

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
                TaskIdent := 2,     // Job from "LBP_TMSpindle"
                TaskIdentNo := 1,   // Interface 1 in "LBP_TMSpindle"
                NewToolMag := 9998, // New tool in spindle
                NewToolLoc := 1,
                OldToolMag := 1,    // Old tool in magazine, location 8
                OldToolLoc := 8,
                Status := 1,        // Job completed
                Ready =>
                #tempReady,
                Error =>
                #tempError);
```

Asynchronous transfer

Magazine 1, location 3 is positioned at change location to buffer location 1.

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
                TaskIdent := 4,     // Asynchronous transfer
                TaskIdentNo := 1,   // Channel 1
                NewToolMag := 1,    // Magazine 1, location 3
                NewToolLoc := 3,
                OldToolMag := 9998, // Buffer, location 1
                OldToolLoc := 1,
                Status := 5,        // Positioning
                Ready =>
                #tempReady,
                Error =>
                #tempError);
```

Relocate tool from magazine 1, location 5 to location 7.

```
// tempReady : Bool
// tempError : Int
"LBP_TMTransfe Start := TRUE,
r"(
                TaskIdent := 4,      // Asynchronous transfer
                TaskIdentNo := 1,    // Channel 1
                NewToolMag := 1,     // Magazine 1, location 7
                NewToolLoc := 7,
                OldToolMag := 1,     // Magazine 1, location 5
                OldToolLoc := 5,
                Status := 1,         // End
                Ready =>
                #tempReady,
                Error =>
                #tempError);
```

See also

LBP_TMTransferTurret [FC7]: Acknowledge tool change with turret (Page 676)

LBP_TMTransferMT [FC6]: Transfer block for tool management and multitool (Page 674)

10.7.7 LBP_ReqASUP [FC9]: Start asynchronous subprogram (ASUP)

Function

With the function "LBP_ReqASUP [FC9]", predefined subprograms (ASUB) can be started in the NC. Before starting, the ASUB in the NC program must be assigned an interrupt number and a priority. Alternatively this assignment can be performed by calling the PI service "ASUB" for the channel via "LBP_ReqPIService [FB7]". After the assignment, the ASUB can be called as often as required.

The NC program running in the channel is interrupted by the ASUB. Several ASUBs can be started in one PLC cycle, the ASUBs are executed in succession in the NC.

A new job to start an ASUB is started with "Start" := TRUE. Channel number and interrupt number must be specified in accordance with the prepared ASUBs. A global or static variable of the WORD type must be created for each job an initialized with 0. This variable is used a parameter "Ref" for internal purposes when the function "LBP_ReqASUP [FC9]" is called and must not be changed. If several jobs are started in parallel, a separate variable must be created for each job.

The job to start an ASUB is completed when "Done" == TRUE or "Error" == TRUE. The request must then be withdrawn with "Start" := FALSE.

General conditions

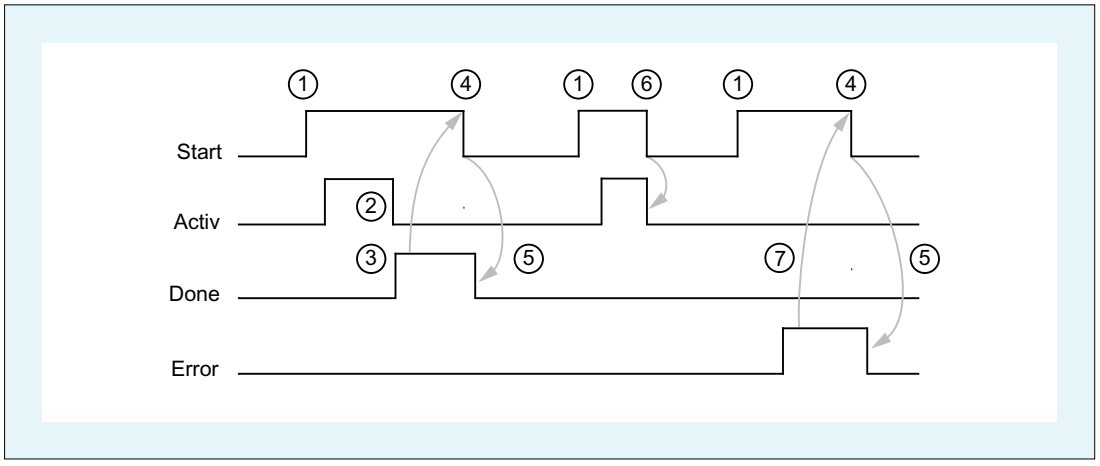
- Before "LBP_ReqASUP [FC9]" can be started, the call of the PI service "ASUB" via "LBP_ReqPIService [FB7]" must be completed.
- No jobs can be started during an emergency stop ("LBP_NC.A_EMERGENCY" (DB10.DBX56.1)).
- No jobs can be started in a channel with active channel reset.

Description of formal parameters

The following table shows all the formal parameters of the function "LBP_ReqASUP [FC9]":

Signal	Category	Type	Value range	Meaning
Start:	I	BOOL	---	Job start with positive signal edge
ChanNo:	I	INT	1, 2, 3 ... 10	Channel number
IntNo:	I	INT	1, 2, 3 ... 8	Interrupt number
Ref:	I/O	WORD	Global variable (MW, DBW, ...)	Internally used job data
Active:	O	BOOL	---	TRUE: ASUB is active
Done:	O	BOOL	---	FALSE: ASUB completed
Error:	O	BOOL	---	TRUE: Job could not be completed successfully, e.g. because the channel is in reset.
StartErr:	O	BOOL	---	TRUE: Interrupt number or channel number invalid

Pulse diagram



(1) Job start with "Start" := TRUE

(2) ASUB is active

(3) Positive acknowledgement: ASUB completed

(4) Reset of "Start" = FALSE after receipt of the acknowledgement

(5) Reset of the output values

(6) Reset of "Start" not permitted before receipt of the acknowledgement: Output values are not set. The ASUB is still completed.

(7) Negative acknowledgement "Error" == TRUE: An error has occurred

Alarm messages

The following alarms are defined for the diagnostics of parameterization errors:

- 404015: "Parameter ChanNo not permitted for "LBP_ReqASUP [FC9]""
- 404016: "Parameter IntNo not permitted for "LBP_ReqASUP [FC9]""

An alarm is reset when "LBP_ReqASUP [FC9]" is called again with "Start" := FALSE or with valid values for the parameters "ChanNo" and "IntNo".

Call example

```
// Starting an asynchronous subprogram in channel 1 with interrupt number 1
// #statStart: Bool;
// #statRef: Word;
// #tempActive: Bool;
// #tempDone: Bool;
// #tempError: Bool;
// #tempStartError: Bool;
// Initial values: #statRef := 0
// Job start: #statStart := TRUE
```

```
"LBP_ReqASUP"(Start := #statStart,
     ChanNo    := 1,
     IntNo     := 1,
     Ref       := #statRef,
     Activ     => #tempActive,
     Done      => #tempDone,
     Error     => #tempError,
     StartErr  => #tempStartError);
IF #tempDone OR #tempError THEN
     #statStart := FALSE;
END_IF;
```

10.7.8 LBP_GenerateAlarmMsgs [FC10]: error and operating messages

Function

The signals entered in "LBP_AlarmMsgs [DB2]" are evaluated with "LBP_GenerateAlarmMsgs [FC10]" and entered in the PLC alarm buffer via internal alarm functions when the alarms occur. They are displayed as incoming or outgoing error and system status messages via a connected HMI device (e.g. SINUMERIK Operate). Error messages are displayed in the alarm list, system status messages in the message list.

Whether a message is issued as an error or system status message can be parameterized via a bit in "LBP_ParamAlarmMsgs [DB5]" byte by byte in "LBP_AlarmMsgs [DB2]".

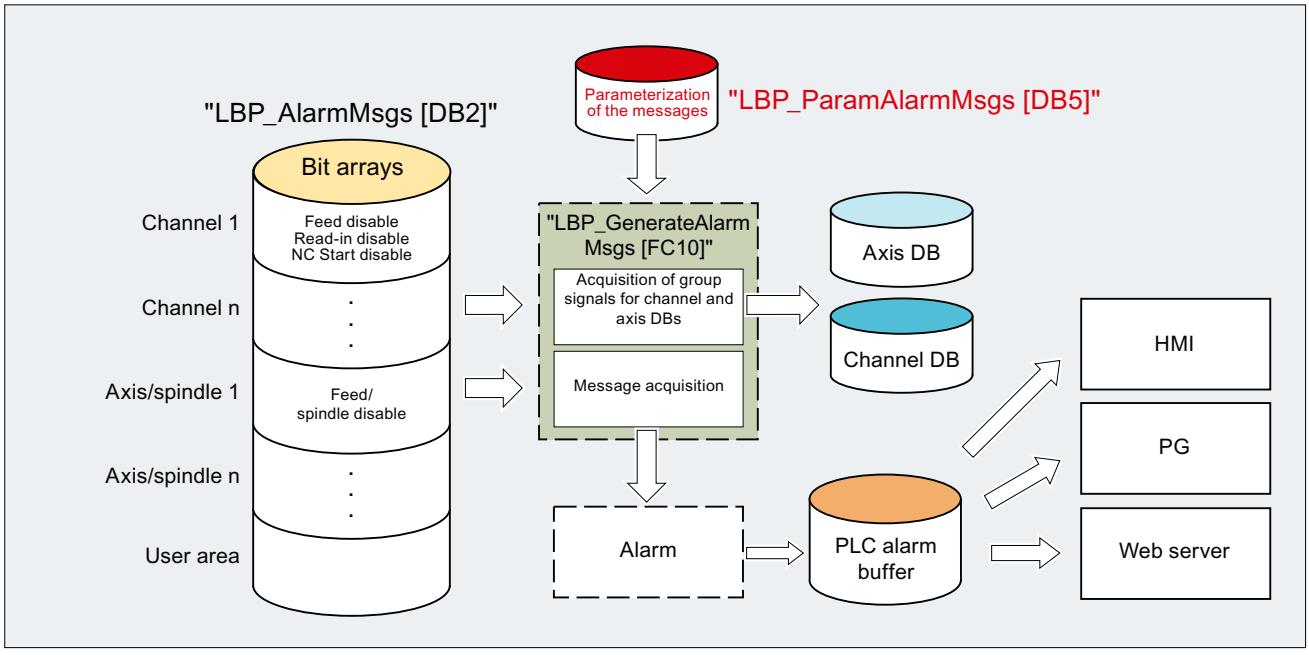
The incoming signals (positive edge) are displayed immediately in the case of both error and system status messages.

Outgoing signals (negative edge) are deleted immediately only for system status messages. In the event of error messages, the messages that are no longer pending are only deleted with the set parameter "Quit", i.e. errors remain displayed on the HMI until they have been acknowledged by the user even if the signals are no longer pending.

The "ToUserIF" parameter can be used to transfer the group signals for the feed, read and NC start disabling signals and feed stop signal to the existing axis, spindle and channel interfaces. The group signals are transferred to the user interface directly from the status information in "LBP_AlarmMsgs [DB2]" irrespective of alarm acknowledgement.

Acquisition of the bit arrays of "LBP_AlarmMsgs [DB2]"

The following figure shows the multiple acquisition and signaling of PLC alarms in "LBP_GenerateAlarmMsgs [FC10]":



The "LBP_AlarmMsgs [DB2]" bit arrays are evaluated multiply with "LBP_GenerateAlarmMsgs [FC10]":

- **Evaluation 1**: Acquisition of **group signals**
  A group signal is generated for each group of signals if at least one bit signal is set to "1". If the "LBP_GenerateAlarmMsgs [FC10]" parameter "ToUserIF" is set to "TRUE", the group signals are led to the associated inhibit signals of the NC/PLC interface. The group signals are acquired completely in cycles.
- **Evaluation 2**: Acquisition of **error and system status messages**
  An error or system status message is generated at a signal state change from "0" to "1" in a bit array of a group. The message class of the alarm can be parameterized via "LBP_ParamAlarmMsgs [DB5]".
  See also "LBP_ParamAlarmMsgs [DB5] (Page 117)".

Signal groups of "LBP_AlarmMsgs [DB2]"

The bit arrays for alarms related to the interface signals are combined with the bit arrays for alarms/messages in "LBP_AlarmMsgs [DB2]".

The signals of "LBP_AlarmMsgs [DB2]" are divided into several signal groups:

Area	Data type	Offset	Comment
Chan	Array[1..10] of Struct	0.0	Bit arrays for channel signals without message generation: • Feedrate disable • Read-in disable • NC Start disable
Axis	Array[1..31] of Struct	120.0	Bit arrays for axis signals without message generation: • Feedrate disable
AddValUserA0	Array[0..63] of INT	182.0	16-bit integer values for alarms/messages in user area 0 (UserA[0]) of the messages 700.000 to 700.063
ChanA	Array[1..10] of Struct	310.0	Bit arrays for channel signals with corresponding message generation: • Feedrate disable • Read-in disable • NC Start disable
AxisA	Array[1..31] of Struct	490.0	Bit arrays for axis signals with corresponding message generation: • Feedrate disable
UserA	Array[0..63] of Struct	554.0	Bit arrays for alarms/messages in the user area

See also "LBP_AlarmMsgs [DB2] (Page 112)".

Acknowledgement

The following acknowledgement procedures are implemented for error and system status messages:

- **System status messages** are intended for the display of normal operating states as information for the operator / machine setter. Acknowledgement signals are, therefore, not required for this type of message. Alarms are acquired both when they arise and when they are cleared and an entry is made in the alarm buffer. The HMI maintains an up-to-date log of existing operating messages using the identifiers "system status message arrived" and "system status message gone".
- **Error messages** display error states on the machine which generally lead to a machine stoppage. If several errors occur simultaneously, it is important to be able to distinguish their order of occurrence for troubleshooting purposes. This is indicated, on the one hand, by the order in which they are entered in the alarm buffer and on the other, by the time stamp which is assigned to every entry.
  If the cause of the error disappears, the associated alarm message is only deleted if the user has acknowledged it (e.g. by pressing a key on the MCP). If the input "Quit" on "LBP_GenerateAlarmMsgs [FC10]" changes to "TRUE", all error bits that are no longer pending (corresponding bit in "LBP_AlarmMsgs [DB2]" is "FALSE") are signaled as cleared. This enables the HMI to also maintain an up-to-date log of pending alarm messages. The time of day indicating the time at which the error occurred is maintained for messages, which are still pending.

Explanation of the parameters of "LBP_GenerateAlarmMsgs [FC10]"

Parameter	Category	Type	Range of values	Meaning
ToUserIF	IN	BOOL	---	TRUE: Signals transferred to user interface in each cycle
Quit	IN	BOOL	---	TRUE: Acknowledgement of error messages
NumActA-larmMsgs	OUT	UINT	0 ... 65535	Number of active alarms/messages
Error	OUT	BOOL	---	TRUE: Means that an error has occurred during the processing. The possible cause of the error is shown in the "StatusId" and "Status" parameters.
StatusId	OUT	UINT	0 ... 65535	Status identifier
Status	OUT	WORD	16#0 ... 16#FFFF	Status information

Parameter "ToUserIF"

If parameter "ToUserIF" = "TRUE", all channel and axis signals are sent to the user interface as a group signal in each case. The user PLC program can therefore influence these signals only via "LBP_AlarmMsgs [DB2]". Depending on the area selection, this can be performed in conjunction with an error or system status message. The appropriate information is overwritten in the user interface.

If parameter "ToUserIF" = "FALSE", there is no transfer of the signals to the user interface. In this case, users must take measures in their PLC program to ensure that these signals are influenced in the interface (appropriate channel and axis DBs).

Parameter "Quit"

If a positive edge is detected on the parameter "Quit", all errors no longer pending are acknowledged (corresponding bit in "LBP_AlarmMsgs [DB2]" = "FALSE") and removed from the alarm buffer. This procedure can take several PLC cycles. This is performed automatically by "LBP_GenerateAlarmMsgs [FC10]" when a positive edge is applied at the parameter.

Parameter "NumActAlarmMsgs"

The number of all active error and system status messages in the current state are shown that can be issued successfully by setting a bit in "LBP_AlarmMsgs [DB2]".

Parameters "Error", "StatusId" and "Status"

The following table contains all the specific error information that is output by the parameters Error, StatusId and Status.

Error	StatusId	Status	Explanation
FALSE	1	16#0	No error
FALSE	2	16#7003	The maximum number of messages per program cycle have been sent.
FALSE	3	16#0085	Message of type "Information only"
FALSE	3	16#70C1	The CPU is currently not generating any messages because initialization routines are running. (This is the case after download during RUN, for example.)

Error	StatusId	Status	Explanation
FALSE	3	16#70C2	The maximum number of messages per time unit have been sent.
FALSE	3	16#70C3	All dynamic message instances are being used.
FALSE	3	16#70C4	Message is being output and cannot be overwritten.
FALSE	4	16#70C1	"LBP_GenerateAlarmMsgs [FC10]" is currently not generating any messages because initialization routines are running.
TRUE	1	16#8232	Data block is not available: "LBP_AlarmMsgs [DB2]"
TRUE	1	16#8233	Data block is not available: "LBP_InternalDB3 [DB3]"
TRUE	1	16#8235	Data block is not available: "LBP_ParamAlarmMsgs [DB5]"
TRUE	1	16#8262	DB access not possible because of attribute "optimized": "LBP_AlarmMsgs [DB2]"
TRUE	1	16#8263	DB access not possible because of attribute "optimized": "LBP_InternalDB3"
TRUE	1	16#8265	DB access not possible because of attribute "optimized": "LBP_ParamAlarmMsgs [DB5]"
TRUE	1	16#8242	Data block too small: "LBP_AlarmMsgs [DB2]"
TRUE	1	16#8243	Data block too small: "LBP_InternalDB3"
TRUE	1	16#8245	Data block too small: "LBP_ParamAlarmMsgs [DB5]"
TRUE	1	16#8252	DB access not possible because of write protection: "LBP_AlarmMsgs [DB2]"
TRUE	1	16#8253	DB access not possible because of write protection: "LBP_InternalDB3"
TRUE	1	16#8255	DB access not possible because of write protection: "LBP_ParamAlarmMsgs [DB5]"
TRUE	2	16#8267	The value of the parameter "MsgUser" in "LBP_ConfigData [DB7]" is outside the valid range of 0 ... 64.
TRUE	2	16#8277	The value of parameter "MaxChan" in "LBP_ConfigData [DB7]" is outside the valid range of 0 ... 10.
TRUE	2	16#8287	The value of parameter "MaxAxis" in "LBP_ConfigData [DB7]" is outside the valid range of 0 ... 31.
TRUE	2	16#8290	"ToUserIF" = "TRUE" is not supported on the hardware used.
TRUE	2	16#87FF	System error. Contact the SINUMERIK hotline of the SIEMENS AG and specify the error.
TRUE	3	16#87FF	Invalid static message information

Note

There is an upper limit for the number of alarms and messages that can be pending at the same time. This is specified in the catalog data of the PLC and is, for example, 1000 messages for the SINUMERIK-integrated PLC1500.

There is also an upper limit of 20 simultaneously issued alarms per PLC cycle. If more than 20 alarms are to be issued simultaneously, the user must ensure that the relevant bits in "LBP_AlarmMsgs [DB2]" are also set in the subsequent PLC cycles. It is recommended that the issuing of several alarms (> 20 alarms) is distributed over several PLC cycles.

Influencing the "LBP_AlarmMsgs [DB2]" evaluation

"LBP_ConfigBP [FC1]" parameter "ExtendChanAxMsg"

The scope of the acquired bit arrays for the axis and channel area is specified by default as the respective number of values defined in the machine (variables "MaxChan" and "MaxAxis" in "LBP_ConfigData [DB7]"), but can be adapted to the maximum values via the basic program parameter "ExtendChanAxMsg" when "LBP_ConfigBP [FC1]" is called.

When this parameter is activated, the entire channel and axis area in "LBP_AlarmMsgs [DB2]" is active for the acquisition of alarms and messages. The parameter is evaluated only when the "LBP_GenerateAlarmMsgs [FC10]" parameter "ToUserIF" is deactivated. In this way, group signals cannot be transferred to the user interface.

"LBP_ConfigBP [FC1]" parameter "MsgUser"

The number of user areas used (UserA area) is entered in the "LBP_AlarmMsgs [DB2]" via this parameter.

The scope of bit arrays for the "LBP_AlarmMsgs [DB2]" user alarms is set by default to 10 areas of 8 bytes each. Up to 64 areas can be parameterized via the "LBP_ConfigBP [FC1]" parameter "MsgUser" depending on the functional application. The actual size of "LBP_AlarmMsgs [DB2]" remains unchanged.

This parameter influences the number of bit arrays in "LBP_AlarmMsgs [DB2]" to be checked and therefore affects the cycle time of the PLC.

"LBP_ConfigBP [FC1]" output variable "MaxChan"

The scope of the acquired bit arrays for the channel area is defined by the "LBP_ConfigBP [FC1]" output variable "MaxChan".

"LBP_ConfigBP [FC1]" output variable "MaxAxis"

The scope of the acquired bit arrays for the axes area is defined by the "LBP_ConfigBP [FC1]" output variable "MaxAxis".

"LBP_ConfigBP [FC1]" output variable "ActiveChan"

The transfer of the channel signals to the respective channel DBs is only performed when the respective bit is set to "TRUE" in the bit array "ActiveChan". Error and system status messages are also generated for channel gaps, that is, for channels whose "ActiveChan" bit is set to "FALSE", but whose number is less than or equal to "MaxChan".

"LBP_ConfigBP [FC1]" output variable "ActiveAxis"

The transfer of the axis signals to the respective axis DBs in the user interface is only performed when the respective bit is set to "TRUE" in the bit array "ActiveAxis". Error and system status messages are also generated for the axis gaps, that is for axes whose "ActiveAxis" bit is set to "FALSE", but whose number is less than or equal to "MaxAxis".

Additional information

When issuing an alarm for user area 0 (UserA[0]), you can also transfer a 16-bit wide associated value in integer format in addition to the alarm number. The user must write this additional information in the array variable "AddValUserA0" of "LBP_AlarmMsgs [DB2]" before setting an alarm bit. You can display this in the HMI using the syntax @<Number of the associated vlaue> %<Format specification>@. In this case, the number of the associated value is 2. For each bit in user area 0 there is an integer value.

When an alarm is issued from the Chan, Axis, ChanA or AxisA area, then the associated value of the alarm automatically contains the number of the channel or the axis.

The table shows the integration of the associated value depending on the trigger bit and alarm number:

Trigger bit from the element in "LBP_AlarmMsgs". ...	**SINUMERIK alarm numbers**	**Syntax for HMI [1]** **Display associated value in alarm texts**	**Meaning of the associated value**
Chan, ChanA	5xx.xxx	@1%d@	SINUMERIK alarm number
Axis, AxisA	6xx.xxx		
UserA	7xx.xxx		
Chan, ChanA	5xx.xxx	@4%u@	Signal number, i.e. the last two digits of the alarm number or the index into the array in "LBP_AlarmMsgs [DB2]"
Axis, AxisA	6xx.xxx		
UserA	7xx.xxx		
Chan, ChanA	5xx.xxx	@3%u@	Channel number
Axis, AxisA	6xx.xxx	@3%u@	Axis number
UserA[0]	700.0xx (xx = 0..63)	@2%d@	Associated values from the element "LBP_AlarmMsgs [DB2]".AddValUserA0[xx] (integer, 16 bit)

1) The format specification is described in the online help of the TIA Portal under "Structure of the associated values".

10.7.9 LBP_CallBackAuxFunc [FC12]: Call interface for users for auxiliary functions

Function

The "LBP_CallBackAuxFunc [FC12]" block is called on an event-driven basis in the basic program when new auxiliary functions are available. PLC users can extend this FC with program instructions for processing their auxiliary function. This avoids the cyclic polling of the channel DBs and permits auxiliary functions to be processed on a jobcontrolled basis.

The "LBP_CallBackAuxFunc [FC12]" is provided as an empty SCL block with the basic program. The channel for which auxiliary functions are available is informed on the input parameter "Chan". The new auxiliary functions can be determined by the auxiliary-function change signals in the channel concerned. The "LBP_CallBackAuxFunc [FC12]" is called for each channel for which auxiliary functions are available.

You can also use your own "LBP_CallBackAuxFunc [FC12]", e.g. if you have migrated an 840D sl project, or want to program in another language. However, it is then important to retain the symbolic name (LBP_CallBackAuxFunc) and define the input parameter Chan of the byte type. Assigning block number 12 ("LBP_CallBackAuxFunc [FC12]") is also recommended. The block can have the attribute "Optimized block access".

Explanations of the formal parameters

Signal	Category	Type	Value range	Meaning
Chan:	I	BYTE	0, 1, 2, ... 9	Index of the channel = channel number -1

Example

In the following SCL example, first the channel in which M auxiliary functions have occurred is determined in the "LBP_CallBackAuxFunc [FC12]". The values for the occurrence of M functions are then read from the corresponding channel DB and written to bit memory Tag_1 and Tag_2.

```
// #Chan: Byte;
// Tag_1: DWord;
// Tag_2: Word;

// Consider all relevant channels, e.g. channel 1 and channel 2
IF (#Chan = 0) THEN
// If M1Change is set, for example, query M1 and ExtM1 and write to bit
memory
     IF ("LBP_Chan1".M1Change) THEN
               "Tag_1" := DINT_TO_DWORD"LBP_Chan1".M1);
               "Tag_2" := INT_TO_WORD("LBP_Chan1".ExtM1);
     END_IF;
END_IF;

IF (#Chan = 1) THEN
// If M1Change is set, for example, query M1 and ExtM1 and write to bit
memory
     IF ("LBP_Chan2".M1Change) THEN
               "Tag_1" := DINT_TO_DWORD("LBP_Chan2".M1);
               "Tag_2" := INT_TO_WORD("LBP_Chan2".ExtM1);
     END_IF;
END_IF;
```

10.7.10 LBP_HTCtrlHT2 [FC13]: Display control for HT 2

Function

The "LBP_HTCtrlHT2 [FC13]" block controls the text display on the display of the HT 2 handheld unit. The text to be displayed is passed as a parameter when the block is called.

The display can display 64 characters spread over four lines per 16 characters. The text is transferred line by line and the transfer of each line takes several PLC cycles.

The block can optionally insert variable proportions into the text to be displayed. Different output data types are supported. The conversion of numerical values results in a fixed length and the converted value is right-justified at the desired position in the text.

For text transfer, the memory areas "HTIn" and "HTOut" are required that are specified when "LBP_ConfigBP [FC1]" is called. For the HT 2, it is necessary to specify "HTIf" := 5.

Explanations of the formal parameters

Signal	Category	Type	Value range	Meaning
Row	I	BYTE	16#0 ... 16#F (2#0000 ... 2#1111)	The lines to be transferred, coded as a bit mask: Bit 0 (2#0001) for 1st line, bit 1 (2#0010) for 2nd, bit 2 (2#0100) for 3rd, bit 3 (2#1000) for 4th line 16#F: all lines 16#0: no lines, or only conversion
ChrArray	I/O	STRING	---	Text that will be transferred to the display of the HT 2
Convert	I	BOOL	---	TRUE: Conversion requested FALSE: No conversion requested, parameters "Addr", "DataType", "StringAddr", "Digits" are not relevant and can be set to 0
Addr	I	VARIANT	---	Value to be converted NULL: If no conversion requested
DataType	I	BYTE	16#**1** ... 16#**8**, 16#**13**, 16#**30**	Only relevant if conversion requested; for possible values, see the following table
StringAddr	I	INT	1 ... 64	Position at which the converted text is to be inserted right-justified Existing text will be overwritten
Digits	I	BYTE	16#0 ... 16#A (0 ... 10)	Number of decimal places, only relevant if floating-point values are converted 0 ... 7 for data type Real 0 ... 10 for data type LReal
Error	O	BOOL	---	TRUE: An error has occurred

Data types and value ranges

Data-Type	Data type	Digits	Character	Area that can be displayed
1	BOOL		1	0, 1
2	BYTE, USINT		3	0 ... 255
3	CHAR		1	Any single character
4	WORD, UINT		5	0 ... 65535
5	INT		6	-32768 ... 32767
6	DWORD, UDINT		7	0 ... 9999999
7	DINT		8	-9999999 ... 99999999
8	Real (7 significant places)	0	9	-9999999 ... 9999999
		1		-999999.9 ... 999999.9
		2		-99999.99 ... 99999.99
		3		-9999.999 ... 9999.999
		4		-999.9999 ... 999.9999
		5		-99.99999 ... 99.99999
		6		-9.999999 ... 9.999999
		7		0 ... 0.9999999
16#13	STRING		1 ... 64	Text up to 64 characters long
16#30	LREAL	0	12	-99999999999 ... 999999999999
		1		-999999999.9 ... 9999999999.9
		2		-99999999.99 ... 999999999.99
		...		---
		8		-99.99999999 ... 999.99999999
		9		-9.999999999 ... 99.999999999
		10		0 ... 9.9999999999

Special features

The text is automatically wrapped round over the four lines of the display:
characters 1–16 on line 1, characters 17–32 on line 2, characters 33–48 on line 3, and characters 49–64 on line 4.

If the string contains more than 64 characters, the following characters are ignored. If the string is shorter than 64 characters, the string is filled with blanks up to 64 characters.

Conversion

The block can convert variable values of different types (see previous table) before transfer and insert them into the text.

The conversion of a numerical value always results in a fixed length that depends on the data type. The numerical value is filled with blanks from the left up to this length.

In the case of floating-point numbers, the number of decimal places to be displayed can be specified via the "Digits" parameter. The total number of decimal places output is not influenced by this. Only the distribution of the number of places before and after the decimal point is changed.

Multiple calls

If more than one converted value is to be inserted in the text, the block must be called multiple times.

Each call inserts the converted partial text into the input text specified by the parameter "ChrArray". Transfer with the parameter "Row" ≠ 0 must only be activated in the last call in the cycle. All previous calls must be made with the parameter "Row" := 0. As a result, no text is transferred, but the result of the conversion is stored.

Remedying errors

Check parameters for input/output addresses "HTIn" and "HTOut", bus address "HTAdr" in the call of "LBP_ConfigBP [FC1]".

Make sure that the data type specified under "DataType" and the type of the value referenced by "Addr" match.

Make sure that the value to be converted is in the displayable range.

The number of decimal places must be in the range specified for the type.

The converted value must not extend beyond the start or end of the text.

Call examples

Single text transfer

```
// #statText: String[64] = '<-- 1. Row --><-- 2. Row --><-- 3. Row --><-- 4.
Row -->';
// #tempError: Bool;

"LBP_HTCtrlHT2"(Row := B#16#F,
            ChrArray := #statText,
            Convert := FALSE,
            Addr := NULL,
            DataType := 0,
            StringAddr := 0,
            Digits := 0,
            Error => #tempError);
```

Table 10-68 Output

```
<-- 1. Row -->
<-- 2. Row -->
<-- 3. Row -->
<-- 4. Row -->
```

Text transfer with single conversion

```
// #statText: String[64] = '<-- i=____ --><-- 2. Row --><-- 3. Row --><-- 4.
Row -->';
// #statNumber: Int = 42;
// #tempError: Bool;

"LBP_HTCtrlHT2"(Row := B#16#F,
            ChrArray := #statText,
            Convert := TRUE,
            Addr := #statNumber,
            DataType := 5,
            StringAddr := 12,
            Digits := 0,
            Error => #tempError);
```

Table 10-69 Output

```
<-- i=  42 -->
<-- 2. Row -->
<-- 3. Row -->
<-- 4. Row -->
```

Text transfer with multiple conversion

```
// #statText: String[64] = '<-- i=____ --><r=_______ --><-- 3. Row --><-- 4.
Row -->';
// #statInt: Int = -42;
// #statFloat: Real = 2.71828;
// #tempError: Bool;

"LBP_HTCtrlHT2"(Row := B#16#0,
            ChrArray := #statText,
            Convert := TRUE,
            Addr := #statInt,
            DataType := 5,
            StringAddr := 12,
            Digits := 0,
            Error => #tempError);

"LBP_HTCtrlHT2"(Row := B#16#F,
            ChrArray := #statText,
            Convert := TRUE,
```

```
                Addr := #statFloat,
                DataType := 8,
                StringAddr := 28,
                Digits := 5,
                Error => #tempError);
```

Table 10-70 Output

```
<-- i= -42 -->
<r=2.71828 -->
<-- 3. Row -->
<-- 4. Row -->
```

10.7.11 LBP_CtrlAxisSpindle [FC18]: Spindle control

Function

Block "LBP_CtrlAxisSpindle [FC18]" can be used to control spindles and axes from the PLC. The block supports the following functions:

- Position spindle
- Rotate spindle
- Oscillate spindle
- Traverse indexing axis
- Traverse positioning axis

One of the functions listed above is triggered on a positive edge (FALSE → TRUE) for the "Start" or "Stop" signal. The "Start" or "Stop" signal must remain TRUE until the function is either positively ("InPos" = TRUE) or negatively ("Error" = TRUE) acknowledged.

References

- Function Manual Axes and Spindles; spindles
- Function Manual Axes and Spindles; positioning axes
- Function Manual Axes and Spindles; indexing axes

Note

Multiple calls for the same axis or spindle

If multiple block calls are programmed for one and the same axis in the PLC user program, these block calls must be programmed as conditional calls to ensure that the block is never called more than once in the same cycle. The block call must be performed cyclically until a status change of output parameter " InPos " or "Error" from TRUE to FALSE occurs. Only then can another block be called for this axis.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Start	I	BOOL	---	0 → 1: Start the function
Stop	I	BOOL	---	0 → 1: Stop the function
Funct	I	BYTE	1, 2, 3, ... 11	1: Position spindle 2: Rotate spindle 3: Oscillate spindle 4: Position indexing axis 5: Position axis, metric (mm) 6: Position axis, imperial (inch) 7: Position axis, metric (mm) with handwheel override 8: Position axis, imperial (inch) with handwheel override 9: Rotate spindle with automatic gear stage selection 10: Rotate spindle with constant cutting speed, metric (mm) 11: Rotate spindle with constant cutting speed, imperial (inch)
Mode	I	BYTE	0, 1, 2, ... 5	0: Positioning to absolute position 1: Positioning incremental 2: Positioning along the shortest path 3: Positioning absolute, positive approach direction 4: Positioning absolute, negative approach direction 5: Direction of rotation as for M4
AxisNo	I	INT	1, 2, 3, ... 31	Number of machine axis
Pos	I	REAL	∓ 0.1469368 E -38 ... ∓ 0.1701412 E +39	rotary axis: Degrees Indexing axis: Indexing position Linear axis: mm or inch
FRate	I	REAL	∓ 0.1469368 E -38 ... ∓ 0.1701412 E +39	Rotary axis and spindle: [rpm] Linear axes: [m/min] or [inch/min]
InPos	O	BOOL	---	TRUE: Position reached or function executed
Error	O	BOOL	---	TRUE: Error
State	O	BYTE	0, 1, 2, ... 255	Error code

General advice on use of the block

Duration of the call

The block must be called cyclically until either the signal "InPos" or "Error" changes from FALSE to TRUE and additionally until the signal "InPos" or, in case of an error, the signal "Error" produces an edge change from TRUE to FALSE. Subsequently, the block must be called for at least one cycle with "Start" := FALSE and "Stop":= FALSE to finalize the axis interchange.

Typically, the block call continues cyclically after the end of the job with "Start" : FALSE and "Stop" := FALSE until the beginning of the next job unless multiple calls are programmed for this axis.

If signals "InPos" and "Error" show the value FALSE, a further "Start" or "Stop" can be called for this axis.

Cancel

Once "Start" := TRUE or "Stop" := TRUE has been called, the function cannot be canceled by resetting the parameters "Start" or "Stop". This can only be done by means of the axial interface signals (e.g. delete distance-to-go). The axial interface also returns status signals of the axis, which have to be evaluated (e.g. exact stop, traverse command).

Simultaneity

Several axes can be traversed simultaneously or time-delayed by blocks. In this case, the block must be cyclically called independently for each axis.

Controlling the axis via the PLC

Before an axis can be traversed by the PLC, the axis must be activated as a PLC axis. Then the PLC can take control of the axis (axis interchange from NC channel to PLC).

Note that machine data 30460 changes from bit 4 to bit 5.

This happens automatically when parameter "Start" or "Stop" is activated. The block requests control of the axis from the NC, after return of control with "Start" :=FALSE or "Stop" := FALSE, the axis can be programmed by the NC program again (axis interchange from PLC to NC channel).

For better response of the axis when several calls of this function are programmed in succession, the PLC user program can also request control for the PLC before calling the block. Then the switching operation is omitted in the block. Activation by the PLC user program is performed in the corresponding axis interface.

See also

- "Status axis/spindle exchange (Page 449)"
  "LBP_Axis*.E_NCASpChan*", "LBP_Axis*.E_PLCType", "LBP_Axis*.E_ChPoss", "LBP_Axis*.E_NeutrASp", "LBP_Axis*.E_PLCASp"
  (DB31, ... DBB68)
- "Axis/spindle exchange request (Page 388)"
  "LBP_Axis*.A_NCASpChan*", "LBP_Axis*.A_NCASpStrobe", "LBP_Axis*.A_PLCASp"
  (DB31, ... DBB8)

Axis disable

If axis disable "LBP_AxisX.A_ASpDisable" (DB31, ... DBX1.3) is set, any axis controlled via the block does not move. Only a simulated actual value is generated. This response is the same when traversing the axis with an axis disable from the NC.

Axes in the coupled grouping

Axes that belong to a coupled grouping can only be traversed by the block via the leading axis. If you attempt to control an axis other than the leading axis from the coupled grouping, neither of the output signals "InPos" or "Error" is set.

Axis/spindle functions

Function 1: Position spindle

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Funct	1: Function "Position spindle"
Mode	1: Positioning incremental 2: Positioning along the shortest path 3: Positioning absolute, positive approach direction 4: Positioning absolute, negative approach direction
AxisNo	Number of machine axis
Pos	Position
FRate	Positioning speed 0: Velocity corresponding to MD35300 $MA_SPIND_POSCTRL_VELO
InPos	TRUE: Position reached with "Exact stop fine"
Error	TRUE: Positioning error
State	Error code

Function 2: Rotate spindle

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Stop	FALSE → TRUE: Stop the function
Funct	2: Function "Rotate spindle"
Mode	5: direction of rotation M4 otherwise, direction of rotation M3
AxisNo	Number of machine axis
FRate	Spindle speed
InPos	TRUE: Set speed is output, see also "LBP_AxisX.E_SetRange" (DB31, ... DBX83.5) Spindle in the setpoint range
Error	TRUE: Positioning error
State	Error code

Function 3: Oscillate spindle

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Stop	FALSE → TRUE: Stop the function
Funct	3: Function "Oscillate spindle"

Parameter	Meaning
AxisNo	Number of machine axis
Pos	Set gear stage
InPos	TRUE: Set speed is output, see also "LBP_AxisX.E_SetRange" (DB31, ... DBX83.5) Spindle in the setpoint range
Error	TRUE: Positioning error
State	Error code

Parameterized oscillation speed: MD35400 $MA_SPIND_OSCILL_DES_VELO

The function of parameter "Pos" depends on the setting in MD35010 $MA_GEAR_STEP_CHANGE_ENABLE

MD35010	Pos	Function
0	0, 1, 2, … 5	Oscillating
1	0	Oscillation with gear stage change M40
	1	Oscillation with gear stage change M41
	2	Oscillation with gear stage change M42
	3	Oscillation with gear stage change M43
	4	Oscillation with gear stage change M44
	5	Oscillation with gear stage change M45

Function 4: Traverse indexing axes

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Funct	4: Function "Position indexing axis"
Mode	0: Positioning to absolute position 1: Positioning incremental 2: Positioning along the shortest path 3: Positioning absolute, positive approach direction 4: Positioning absolute, negative approach direction
AxisNo	Number of machine axis
Pos	Indexing position
FRate	Positioning speed 0: Velocity corresponding to MD32060 $MA_POS_AX_VEL
InPos	TRUE: Position reached with "Exact stop fine"
Error	TRUE: Positioning error
State	Error code

The modulo conversion can be compared with approaching the indexing position via POS[AX] = CIC (value) in the part program.

Function 5, 6, 7, 8: Position axes

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Funct	5: Position axis, metric (mm) 6: Position axis, imperial (inch) 7: Position axis, metric (mm) with handwheel override 8: Position axis, imperial (inch) with handwheel override
Mode	0: Positioning to absolute position 1: Positioning incremental 2: Positioning along the shortest path 3: Positioning absolute, positive approach direction 4: Positioning absolute, negative approach direction
AxisNo	Number of machine axis
Pos	Position
FRate	Positioning speed 0: Velocity corresponding to MD32060 $MA_POS_AX_VELO
InPos	TRUE: Position reached with "Exact stop fine"
Error	TRUE: Positioning error
State	Error code

Function 9: Rotate spindle with automatic gear stage selection

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Stop	FALSE → TRUE: Stop the function
Funct	9: Function "Rotate spindle with automatic gear stage selection"
Mode	5: direction of rotation M4 otherwise, direction of rotation M3
AxisNo	Number of machine axis
FRate	Spindle speed
InPos	TRUE: Setpoint speed is output
Error	TRUE: Positioning error
State	Error code

Function 10, 11: Rotate spindle with constant cutting speed

The "Constant cutting speed" function (G96) must be active in the NC.

Parameter	Meaning
Start	FALSE → TRUE: Start the function
Stop	FALSE → TRUE: Stop the function
Funct	10: Function "Rotate spindle with constant cutting speed, metric (mm)" 11: Function "Rotate spindle with constant cutting speed, imperial (inch)"
Mode	5: direction of rotation M4 otherwise, direction of rotation M3

Parameter	Meaning
AxisNo	Number of machine axis
FRate	Cutting speed
InPos	TRUE: Setpoint speed is output
Error	TRUE: Positioning error
State	Error code

Error codes

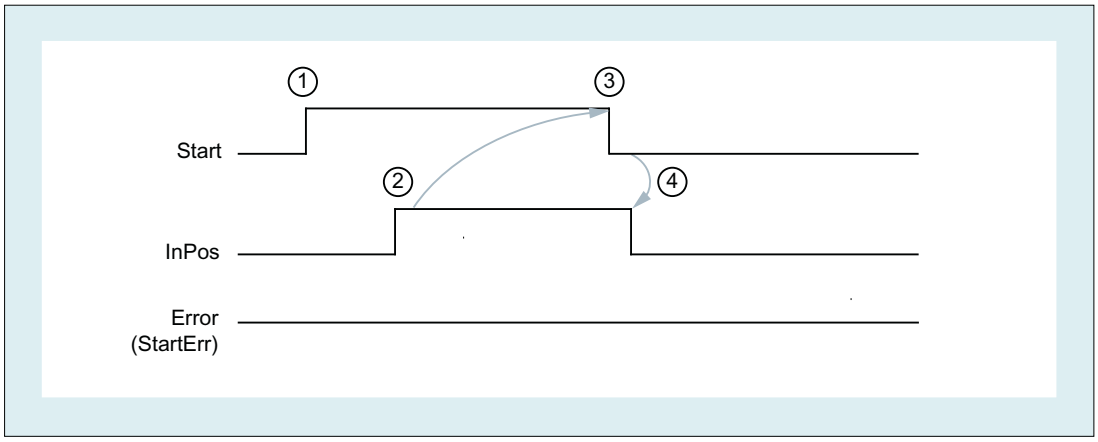
State	Meaning
100	Incorrect position programmed for axis/spindle (corresponds to alarm 16830)
101	Programmed speed is too high
102	Incorrect value range for constant cutting speed (corresponds to alarm 14840)
104	Following spindle: Illegal programming (corresponds to alarm 22030)
105	No measuring system available (corresponds to alarm 16770)
106	Axis positioning still active (corresponds to alarm 22052)
107	Reference mark not found (corresponds to alarm 22051)
108	No transition from speed control to position control (corresponds to alarm 22050)
109	Positioning speed is too high (corresponds to alarm 22055)
110	Velocity/speed is negative
111	Setpoint speed = 0
112	Invalid gear stage
115	Programmed position has not been reached
117	G96/G961 is not active in the NC
118	G96/G961 is still active in the NC
120	Axis is not an indexing axis (corresponds to alarm 20072)
121	Indexing position not permissible (corresponds to alarm 17510)
125	Positioning to shortest distance not possible (corresponds to alarm 16800)
126	Absolute positioning in negative approach direction not possible (corresponds to alarm 16820)
127	Absolute positioning in positive approach direction not possible (corresponds to alarm 16810)
128	No transverse axis available for diameter programming (corresponds to alarm 16510)
130	Software limit switch plus (corresponds to alarm 20070)
131	Software limit switch minus (corresponds to alarm 20070)
132	Working area limit plus (corresponds to alarm 20071)
133	Working area limit minus (corresponds to alarm 20071)
134	Frame not permitted for indexing axis
135	Indexing axis with active "Hirth joint" (corresponds to alarm 17501)
136	Indexing axis with active "Hirth joint" not referenced (corresponds to alarm 17503)
137	Spindle mode not possible for transformed spindle/axis (corresponds to alarm 22290)
138	Axis: Coordinate system-specific working area plus violated (corresponds to alarm 20082)
139	Axis: Coordinate system-specific working area minus violated (corresponds to alarm 20082)
200	System error (corresponds to alarm 1000 and higher, with system error number 450007)

Additional references

Diagnostics Manual

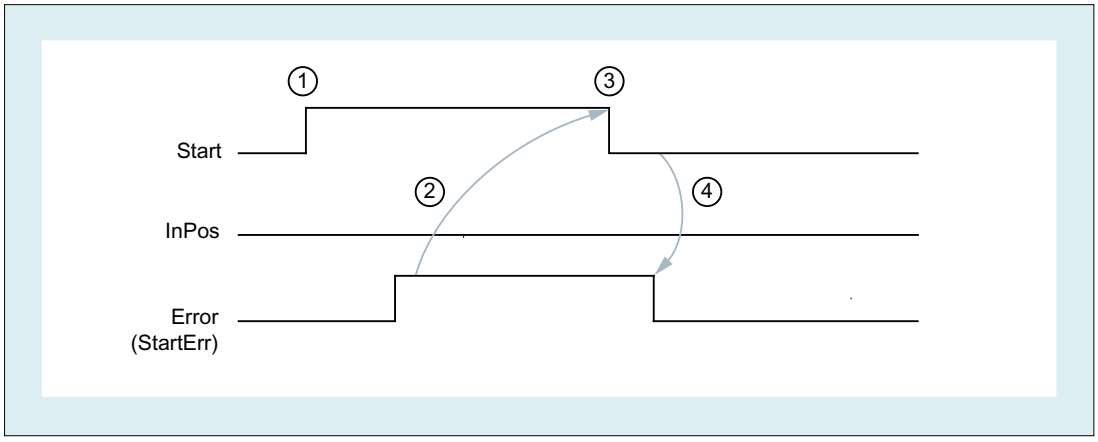
Pulse diagram

Timing without fault



① User: Function start ("Start" or "Stop") with positive edge FALSE → TRUE
② "LBP_CtrlAxisSpindle [FC18]": Positive acknowledgement, function executed or position reached
③ User: Reset "Start" or "Stop" after positive acknowledgement
④ "LBP_CtrlAxisSpindle [FC18]": Reset positive acknowledgement

Timing in case of an error



① User: Function start ("Start" or "Stop") with positive edge FALSE → TRUE
② "LBP_CtrlAxisSpindle [FC18]": Negative acknowledgement, error occurred
③ User: Reset "Start" or "Stop" after negative acknowledgement
④ "LBP_CtrlAxisSpindle [FC18]": Reset negative acknowledgement

Call examples

Position spindle

```
// Position spindle (machine axis 5) to 90°, along the shortest path
// #statStart: Bool;
// #tempPositionReached: Bool;
// #tempError: Bool;
// #tempErrorCode: Byte;
// Job start: #statStart := TRUE;

"LBP_CtrlAxisSpindle"(Start := #statStart,
     Stop      := FALSE,
     Funct     := 1,    // position spindle
     Mode      := 2,    // shortest path
     AxisNo    := 5,    // 5th Machine axis
     Pos       := 90.0,
     FRate     := 600.0,
     InPos     => #tempPositionReached,
     Error     => #tempError,
     State     => #tempErrorCode);
IF #tempPositionReached OR #tempError THEN
     #statStart := FALSE;
END_IF;
```

Rotate spindle

```
// rotate spindle (machine axis 5) at 3000 rpm, start and stop
// #statStart: Bool;
// #statStop: Bool;
// #tempPositionReached: Bool;
// #tempError: Bool;
// #tempErrorCode: Byte;
// job start to start: #statStart := TRUE;
// job start to stop: #statStop := TRUE;

"LBP_CtrlAxisSpindle"(Start := #statStart,
     Stop      := #statStop,
     Funct     := 2,    // rotate spindle
     Mode      := 5,    // direction of rotation as for M4
     AxisNo    := 5,    // 5th Machine axis
     Pos       := 0,
     FRate     := 3000.0,
     InPos     => #tempPositionReached,
     Error     => #tempError,
```

```
     State     => #tempErrorCode);
IF #tempPositionReached OR #tempError THEN
     #statStart := FALSE;
     #statStop := FALSE;
END_IF;
```

Oscillate spindle

```
// oscillate, start and stop spindle (machine axis 5)
// #statStart: Bool;
// #statStop: Bool;
// #tempPositionReached: Bool;
// #tempError: Bool;
// #tempErrorCode: Byte;
// job start to start: #statStart := TRUE;
// job start to stop: #statStop := TRUE;

"LBP_CtrlAxisSpindle"(Start := #statStart,
     Stop      := #statStop,
     Funct     := 3,    // oscillate spindle
     Mode      := 0,
     AxisNo    := 5,    // 5th Machine axis
     Pos       := 0,
     FRate     := 0,
     InPos     => #tempPositionReached,
     Error     => #tempError,
     State     => #tempErrorCode);
IF #tempPositionReached OR #tempError THEN
     #statStart := FALSE;
     #statStop := FALSE;
END_IF;
```

Traverse indexing axis

```
// position indexing axis (machine axis 4) at next indexing position
// #statStart: Bool;
// #tempPositionReached: Bool;
// #tempError: Bool;
// #tempErrorCode: Byte;
// Job start: #statStart := TRUE;

"LBP_CtrlAxisSpindle"(Start := #statStart,
     Stop      := FALSE,
     Funct     := 4,    // position indexing axis
     Mode      := 1,    // position incrementally
```

```
     AxisNo    := 4,    // 4th Machine axis
     Pos       := +1,    // +1 indexing position
     FRate     := 60.0,
     InPos     => #tempPositionReached,
     Error     => #tempError,
     State     => #tempErrorCode);
IF #tempPositionReached OR #tempError THEN
     #statStart := FALSE;
END_IF;
```

Position axis

```
// traverse positioning axis (machine axis 1) by 100mm
// #statStart: Bool;
// #tempPositionReached: Bool;
// #tempError: Bool;
// #tempErrorCode: Byte;
// Job start: #statStart := TRUE;

"LBP_CtrlAxisSpindle"(Start := #statStart,
     Stop      := FALSE,
     Funct     := 5,    // position axis
     Mode      := 1,    // position incrementally
     AxisNo    := 1,    // 1st Machine axis
     Pos       := 100.0,    // +100 mm
     FRate     := 1000.0,
     InPos     => #tempPositionReached,
     Error     => #tempError,
     State     => #tempErrorCode);
IF #tempPositionReached OR #tempError THEN
     #statStart := FALSE;
END_IF;
```

10.7.12 LBP_MCPCtrlMilling [FC19]: Transfer of the MCP signals on the interface

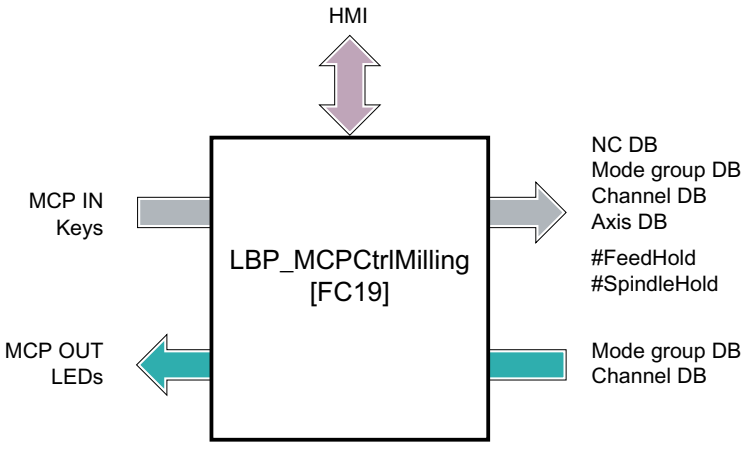
Function

The block "LBP_MCPCtrlMilling [FC19]" transfers signals between a machine control panel (MCP) and the NC/PLC interface (NC DB, Mode group DB, Channel DB, Axis DB). The signals are transferred from the MCP (keys, etc.) to data blocks of the NC/PLC interface as well as signals from data blocks of the NC/PLC interface to the MCP (LEDs).

The block transfers the following signals for the M variant of the MCP 483 machine control panel:

- Keyswitch
- Modes
- Program control
- Axis traversing signals
- Spindle and feedrate override
- WCS/MCS switchover

The following figure shows the signal transfer between the MCP, NC/PLC interface and block outputs:



The memory areas MCP-IN and MCP-OUT are used for the signal transfer as specified in the parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" in the call of "LBP_ConfigBP [FC1]". The parameter "MCPNum" has no effect on the function of the block. See also Section "Operation of several machine control panels" below and Chapter LBP_ConfigBP [FC1]: Basic program, startup section (Page 662).

Feed start/stop and spindle start/stop are not transferred on the NC/PLC interface, but output as output parameters "FeedHold" and "SpindleHold". The state is stored internally, the outputs are retained even after the key is pressed. The block controls the associated LEDs based on the state stored internally.

If the machine control panel fails, all signals of this machine control panel are overwritten with zero and the output parameters "FeedHold" and "SpindleHold" set to TRUE.

Description of formal parameters

Signal	Cate-gory	Type	Value range	Meaning
BAGNo:	I	BYTE	B#16#**0**0 ... B#16#**0**A (0 ... 10)	**1st MCP**: Number of the mode group to which the mode signals are transferred B#16#00: No transfer of mode signals
			B#16#**1**0 ... B#16#**1**A	**2nd MCP**: Number of the mode group to which the mode signals are transferred B#16#10: No transfer of mode signals

Signal	Category	Type	Value range	Meaning
ChanNo:	I	BYTE	0 ... 10	Number of the channel to which the channel signals are transferred 0: No transfer of channel signals
SpindleIFNo:	I	BYTE	0 ... 31	Number of the machine axis to which the spindle data is transferred 0: No transfer of spindle data
FeedHold :	O	BOOL	---	Feed stop from MCP
SpindleHold :	O	BOOL	---	Spindle stop from MCP

Special features

Multiple calls of the block

In order to transfer signals to several mode groups, channels or spindles, the block can be called several times with the corresponding parameters "BAGNo", "ChanNo" and "SpindleIFNo".

At the first call of the block for a machine control panel, all signals are transferred to the data blocks of the NC/PLC interface and the LEDs activated. Now only signals to mode groups and channels are transferred in the following calls in the same PLC cycle.

Operation of several machine control panels

The block can process the signals of up to two machine control panels. To do this, the block must be called for each MCP. The first machine control panel must be called first. Then the second machine control panel called. Key-operated switch position and feedrate override for all axes are only transferred on the first call in the cycle.

Valid input and output areas must be specified as parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" in the call of "LBP_ConfigBP [FC1]" – irrespective of the value of parameter "MCPNum" - for each MCP for which the block is called.

Machine coordinate system and workpiece coordinate system

You can switch between the "machine coordinate system" (MCS) and the "workpiece coordinate system" (WCS) by pressing the MCS/WCS key.

The axis traversing signals are transferred to the data block of the selected machine axis in MCS mode. See the table below "Manual traversing of machine axes (from MCP to NC/PLC interface)".

The axis traversing signals are transferred to the data block of the channel that is specified via the parameter "ChanNo" in WCS mode. See the table below "Manual traversing of geometry axes (from MCP to NC/PLC interface)".

No axis is selected after switchover to MCS or WCS via the machine control panel.

Manual traversing in tool orientation

The R11 direction key on the machine control panel can switch the mode to "Manual traversing in tool orientation" (Tool coordinate system, TCS). Prerequisite is that the parameter "MCP_IF_TCS" := TRUE has been set in the call of "LBP_ConfigBP [FC1]".

With the switchover to TCS, the Z axis (3rd geometry axis) is permanently selected by the block. The axis traversing signals are transferred to the data block of the channel specified via the parameter "ChanNo" in this mode. See the table below "Manual traversing in tool orientation (from MCP to NC/PLC interface)".

Flexible axis configuration

The assignment of axis selection keys to machine axes can be configured flexibly.

Each key can be assigned to an arbitrary machine axis. If more than nine machine axes are to be selected, the assignment can be changed dynamically. The new assignment only takes effect when the axis selection is changed.

The assignment is made via the data block "LBP_NC [DB10]". The number of the machine axis that is to be selected by the ith axis selection key of the first machine control panel is entered in the assignment table "LBP_NC.MCP1AxisTbl[i]".

The assignment is entered in "LBP_NC.MCP2AxisTbl" for the keys of the second machine control panel.

If the value 0 is specified in an entry of the assignment tables, no axis is selected by the associated key.

The number of usable axis selection keys can be limited by the value "LBP_NC.MCP1MaxAxis" or "LBP_NC.MCP2MaxAxis". If the value 0 (default setting) is used, all axis selection keys assigned a machine axis can be used.

Machine axes 1 to 9 are preselected for axis selection keys 1 to 9.

Table 10-71 Example of flexible axis configuration

Key	Assignment	Axis
Direction key X axis	LBP_NC.MCP1AxisTbl[1] := 7	7th machine axis
Direction key Y axis	LBP_NC.MCP1AxisTbl[2] := 2	2nd machine axis
Direction key Z axis	LBP_NC.MCP1AxisTbl[3] := 0	No axis
Direction key 4th axis	LBP_NC.MCP1AxisTbl[4] := 1	1st machine axis

Axis selection from the HMI

In addition to the nine axis selection keys of the MCP, the active axis can also be selected at the HMI. The axis selected from the HMI is only transferred to the MCP if the channel selected at the HMI and at the "LBP_MCPCtrlMilling" match one another. When an axis is selected from the HMI, just the same as when selecting an axis with the axis selection keys of the MCP, initially, an attempt is made to find the assignment between the selected axis number and the actual machine axis in the axis table of the NC-DB. See Section "Flexible axis configuration". This means that when an axis is selected from the HMI, the same machine axis is selected as if the corresponding axis selection key of the MCP had been selected. If an assignment is not possible, then the axis number selected from the HMI is directly used as machine axis. As a consequence it is possible to select axes that cannot be selected via the MCP.

Single block mode

Pressing the Single Block key activates single block mode, pressing the key again deactivates single block mode. If the mode is called several times for one MCP, the block transfers single block mode state from the channel of the first call to the channels of the subsequent calls.

Increment selection

The signals of the increment keys are transferred to the mode group DB of the NC/PLC interface. The NC evaluates the signals from the mode group DB when LBP_NC.A_IncInModeGroup := TRUE is set. This is the default setting after system run-up.

Signals

Control signals from MCP to NC/PLC interface

Table 10-72 Key-operated switch (from MCP to NC/PLC interface)

MCP switch	Interface DB "LBP_NC [DB10]"
Position 0	A_Keyswitch0
Position 1	A_Keyswitch1
Position 2	A_Keyswitch2
Position 3	A_Keyswitch3

Table 10-73 Operating modes and machine functions (from MCP to NC/PLC interface)

MCP key	Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) - Display for mode group 1
AUTOMATIC	A_AUTO
MDI	A_MDA
JOG	A_JOG
REPOS	A_REPOS
REF	A_REF
TEACH IN	A_TEACHIN
INC 1 ... 10 000, INC Var.	A_INC1 ... 10000, A_INCVar

Table 10-74 Manual traversing of machine axes (from MCP to NC/PLC interface)

MCP key	Axis DB: "LBP_Axis1 [DB31], ..." (selected axis)
Direction key +	A_Plus
Direction key -	A_Minus
Rapid traverse override	A_RapidTrOR

Table 10-75 Manual traversing of geometry axes (from MCP to NC/PLC interface)

MCP key	Channel DB: "LBP_Chan1 [DB21], ..." (Parameter ChanNo and selected axis)		
Direction key +	A_Geo[1].Plus	A_Geo[2].Plus	A_Geo[3].Plus
Direction key -	A_Geo[1].Minus	A_Geo[2].Minus	A_Geo[3].Minus
Rapid traverse override	A_Geo[1].RapidTrOR	A_Geo[2].RapidTrOR	A_Geo[3].RapidTrOR

Table 10-76 Manual traversing in tool orientation (from MCP to NC/PLC interface)

MCP keys	**Channel DB: "LBP_Chan [DB21], ..." (parameter ChanNo)**
Direction key R11	E_CART_JOG_MODE
Direction key +	A_Geo[3].Plus
Direction key -	A_Geo[3].Minus
Rapid traverse override	A_Geo[3].RapidTrOR

Table 10-77 Feedrate and spindle speed override (from MCP to NC/PLC interface)

MCP rotary switch	**Channel DB: "LBP_Chan [DB21], ..." (parameter ChanNo)**
Feedrate override	A_FD_OR
MCP rotary switch	**All axis DBs: "LBP_Axis1 [DB31], ..." (from 1st MCP only)**
Feedrate override	A_FD_OR
MCP rotary switch	**Axis DB: "LBP_Axis1 [DB31], ..." (parameter SpindleIFNo)**
Spindle speed override	A_SpOR

Table 10-78 Channel signals (from MCP to NC/PLC interface)

MCP keys	**Channel DB: "LBP_Chan [DB21], ..." (parameter ChanNo)**
NC Start	A_NCStart
NC Stop	A_NCStop
Reset	A_Reset
Single block	A_SBL

Table 10-79 Feed and spindle enable (from MCP to NC/PLC interface)

MCP keys	**Output parameter**
Feed stop Feed enable	Parameter: "FeedHold"
Spindle stop Spindle enable	Parameter: "SpindleHold"

Feedback signals to MCP from NC/PLC interface

Table 10-80 Operating modes and machine functions (to MCP from NC/PLC interface)

MCP LED	Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) - Display for mode group 1
AUTOMATIC	E_AUTO
MDI	E_MDA
JOG	E_JOG
REPOS	E_REPOS
REF	E_REF
TEACH IN	E_TEACHIN
INC 1 ... 10 000, INC Var.	E_INC1 ... 10000, E_INCVar

Table 10-81 Channel signals (to MCP from NC/PLC interface)

MCP LED	Channel DB: "LBP_Chan [DB21], ..." (parameter ChanNo)
NC Start	E_ProgRunning
NC Stop	E_ProgStop and E_ProgInterrupt
Reset	Not available
Single block	A_SBL *(same signal as key)*

Note

LEDs of the direction keys +/- are activated by actuating.

LEDs of the axis keys are activated by selecting an axis or MCS/WCS switchover.

The LEDs of the feed start/stop and spindle start/stop keys are switched after the keys have been pressed.

LEDs of WCS and R11 are switched after actuating.

LEDs of rapid traverse and reset are not activated.

Alarm messages

The following alarms are defined for the diagnostics of parameterization errors:

- 404018: "Parameter BAGNo not permitted for LBP_MCPCtrlMilling [FC19]"
- 404019: "Parameter ChanNo not permitted for LBP_MCPCtrlMilling [FC19]"
- 404026: "Pointer machine control panel %1 incorrect"

Call examples

The following example shows a single call:

```
// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
// #tempFeedHold: Bool;
// #tempSpindleHold: Bool;
"LBP_MCPCtrlMilling"(BAGNo := 1,         // or B#16#01
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);
```

The following example shows a multiple call:

```
// #tempFeedHold: Bool;
// #tempSpindleHold: Bool;

// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
"LBP_MCPCtrlMilling"(BAGNo := 1,           // or B#16#01
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);

// Call of the block for the 1st MCP, without mode group, channel 2,
   without spindle
"LBP_MCPCtrlMilling"(BAGNo := 0,           // or B#16#00
            ChanNo := 2,
            SpindleIFNo := 0,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);

// Call of the block for the 1st MCP, mode group 2, channel 3,
   spindle: Axis 8
"LBP_MCPCtrlMilling"(BAGNo := 2,           // or B#16#02
            ChanNo := 3,
            SpindleIFNo := 8,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);
```

The following example shows a call for several MCPs:

```
// #tempFeedHoldMcp1: Bool;
// #tempSpindleHoldMcp1: Bool;
// #tempFeedHoldMcp2: Bool;
// #tempSpindleHoldMcp2: Bool;

// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
"LBP_MCPCtrlMilling"(BAGNo := B#16#01,
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHoldMcp1,
            SpindleHold => #tempSpindleHoldMcp1);

// Call of the block for the 2nd MCP, without mode group, channel 2,
   without spindle
"LBP_MCPCtrlMilling"(BAGNo := B#16#10,
            ChanNo := 2,
            SpindleIFNo := 0,
            FeedHold => #tempFeedHoldMcp2,
            SpindleHold => #tempSpindleHoldMcp2);
```

10.7.13 LBP_TransferSelData [FC21]: Data exchange NC/PLC

Function

The block "LBP_TransferSelData [FC21]" exchanges data between the PLC and NC. The data are transferred immediately whenever the block is called, irrespective of the cyclic data exchange.

Data transfer is activated by calling the block with parameter "Enable" = TRUE.

The block provides the following functions:

- Synchronized action signals: PLC → NC channel (LBP_Chan*.A_SynDisabl (DB21, ...from DBB300))
- Synchronized action signals: NC channel → PLC (LBP_Chan*.E_SynDisabled (DB21, ...from DBB308))
- Fast data exchange NC → PLC
- Fast data exchange PLC → NC
- Update signals to channel (LBP_Chan*.DBB6" and DBB7")
- Update signals to axes (LBP_Axis*.DBB2 (DB31, ...DBB2))
- Update signals to axes (LBP_Axis*.DBB4 (DB31, ...DBB4))

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Enable	I	BOOL	---	TRUE: Transfer data
Funct	I	BYTE	1 ... 7	1: Synchronized actions to channel
				2: Synchronized actions from channel
				3: Fast data exchange, read
				4: Fast data exchange, write
				5: Control signals to channel
				6: Control signals (part 1) to axis
				7: Control signals (part 2) to axis
S7Var	I	VARIANT	---	Depends on "Funct"
IVAR1	I	INT	---	Depends on "Funct"
IVAR2	I	INT	---	Depends on "Funct"
Error	O	BOOL	---	TRUE: An error has occurred
ErrCode	O	INT	---	1: "Funct" invalid Otherwise: Depends on "Funct"

See also

Function 1, 2: Signals synchronized actions to / from Channel (Page 721)

Function 3, 4: Fast data exchange PLC-NC (Page 722)

Function 5: Update control signals to channel (Page 727)

Function 6: Update control signals to axes (Page 728)

Function 7: Update control signals to axes (Page 728)

10.7.13.1 Function 1, 2: Signals synchronized actions to / from Channel

With these functions, the data areas for synchronized actions are transferred to the channel from the channel interface in the PLC (LBP_Chan*.A_SynDisabl (DB21, ...from DBB300)) or from the channel to the channel interface in the PLC (LBP_Chan*.E_SynDisabled (DB21, ...from DBB308)). Transfer to or from the NC occurs immediately.

Signal	Category	Type	Value range	Meaning
Enable	I	BOOL	---	TRUE: Transfer data
Funct	I	BYTE	1, 2	1: Synchronized actions to channel 2: Synchronized actions from channel
IVAR1	I	INT	1 ... 10	Channel number
Error	O	BOOL	---	TRUE: An error has occurred
ErrCode	O	INT	---	10: Channel number invalid

Call example:

```
// #tempError: Bool
// #tempStatus: Int
"LBP_TransferSelData"(Enable := TRUE,         // transfer data
             Funct := 1,                      // to channel
             S7Var := NULL,                   // not relevant
             IVAR1 := 1,                      // Channel 1
             IVAR2 := 0,                      // not relevant
             Error => #tempError,
             ErrCode => #tempStatus);
```

10.7.13.2 Function 3, 4: Fast data exchange PLC-NC

General

A separate, internal data area is provided to allow the fast exchange of information between the NC and the PLC. This interface encompasses 4096 bytes. The structure of the data in the interface is freely definable.

The data in the interface can be accessed from a NC program as follows:

$A_DBB[x] for byte	Offset x=0,1,2,3 ... 4095
$A_DBW[x] for word	Offset x=0,2,4,6 ... 4094
$A_DBD[x] for DWord	Offset x=0,4,8,12 ... 4092
$A_DBR[x] for Real	Offset x=0,4,8,12 ... 4092

All these variables use the same memory area. Index "x" represents the offset address (0 to 4095) in the data area.

The same offsets are used for access from the PLC

For access from the PLC, it is recommended that the basic data types BYTE, CHAR, WORD, INT, DWORD, DINT or REAL are used if only one value is transferred, and arrays of these data types (Array of ...) if multiple values are to be transferred. In principle, the data interface can be structured in any way (STRUCT). In this case, however, data will always be transferred byte by byte. Any filler bytes in the PLC program resulting from alignments and the necessity to include SWAP commands must be taken into account. See example 3.

Semaphore

A semaphore byte can be used to ensure consistency during data transfer. In this concept, data cannot be read by the other side until they have been written in full.

The semaphore byte is in the same data area (0 to 4095). Therefore, is must not lie outside the area that is used for the transfer of data. The read operation on the PLC must use the same semaphore byte as the associated write operation on the NC. The write operation on the PLC must use the same semaphore byte as the associated write operation on the NC.

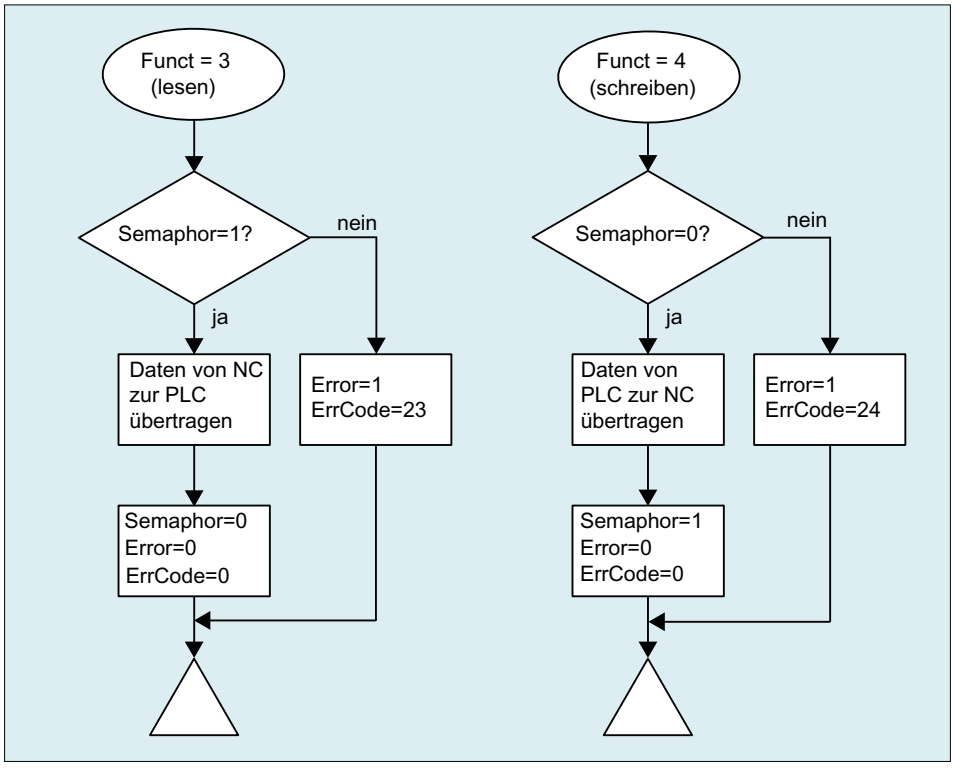
Furthermore, a separate semaphore byte must be used for transfer NC → PLC and transfer PLC → NC. The number of usable semaphore bytes for the various read and write access operations is only limited by the size of the data area.

The block checks the semaphore byte before the read or write operation and writes to it when the read or write operation has been completed.

The semaphore mechanism must be programmed in the NC program (see below).

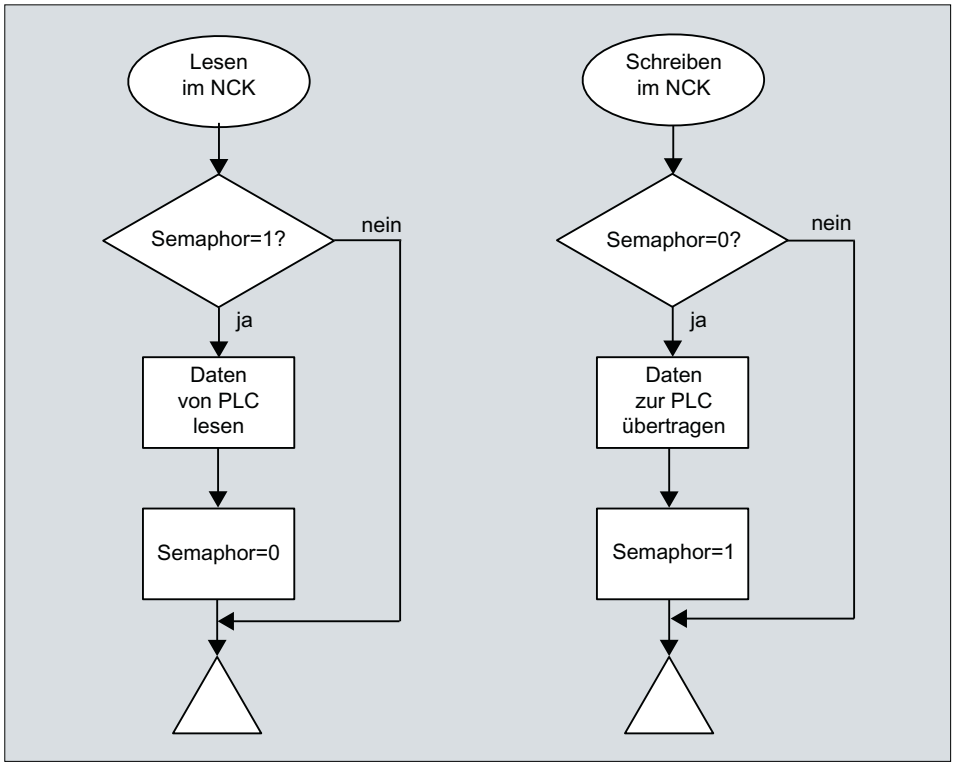
Using the semaphore in the PLC

The block takes over control of and sets the semaphore byte.



Using the semaphore in the NC

Control and setting of the semaphore byte must be performed explicitly.



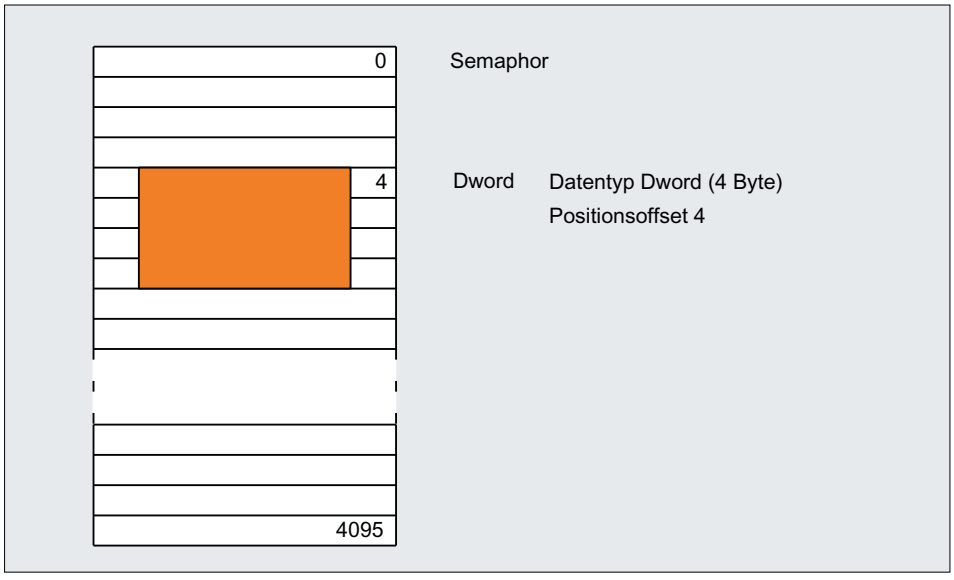
Variable value ranges

Signal	Category	Type	Value range	Meaning
Enable	I	BOOL	---	TRUE: Transfer data
Funct	I	BYTE	3, 4	3: Fast data exchange, read
				4: Fast data exchange, write
S7Var	I	VARIANT	---	Source data area (for "Funct" := 4) Target data area (for "Funct" := 3)
IVAR1	I	INT	0 ... 4095	Offset
IVAR2	I	INT	-1, 0 ... 4095	Offset for semaphore byte -1: Transfer without semaphore
Error	O	BOOL	---	TRUE: An error has occurred

Signal	Category	Type	Value range	Meaning
ErrCode	O	INT	---	20: Alignment error of Word/DWord/Real
				21: Illegal shift offset
				22: Illegal semaphore byte
				23: No new data available for reading (semaphore)
				24: New data cannot be written (semaphore)
				26: S7Var data area invalid (NULL)
				28: S7Var points to data area > 4096 bytes
				30: S7Var shows invalid data area (<> NULL)

Example 1: Read

A DWord from offset 4 is read by the PLC in the memory area for fast data exchange. Byte 0 is used as the semaphore.



PLC programming

```
// #tempData: DWord
// #tempError: Bool
// #tempStatus: Int
"LBP_TransferSelData"(Enable := TRUE,    // transfer data
            Funct := 5,                  // read from NC
            S7Var := #tempData,          // variable to which data is
                                         written
            IVAR1 := 4,                  // read from Offset 4
            IVAR2 := 0,                  // semaphore in byte 0
```

```
            Error => #tempError,
            ErrCode => #tempStatus);
```

Programming the NC with synchronized actions

Writing data to the PLC. Byte 0 must also be used as the semaphore.

```
ID=1 WHENEVER $A_DBB[0] == 0 DO $A_DBD[4] = $R1 $A_DBB[0] = 1
```

Example 2: Write

Ten words from offset 8 are written to the memory area for fast data exchange. Byte 1 is used as the semaphore.

PLC programming

```
// #tempData: Array [1..10] of Word
// #tempError: Bool
// #tempStatus: Int
"LBP_TransferSelData"(Enable := TRUE,    // transfer data
            Funct := 4,                  // write to NC
            S7Var := #tempData,          // variable from which data are
                                         read
            IVAR1 := 8,                  // write from offset 8
            IVAR2 := 1,                  // semaphore in byte 1
            Error => #tempError,
            ErrCode => #tempStatus);
```

NC programming using synchronized actions

Reading data from the PLC, byte 1 must also be used as the semaphore.

```
ID=2 WHENEVER $A_DBB[1] == 1 DO $R1 = $A_DBW[8] $R2 = $A_DBW[10]
$A_DBB[1] = 0
```

Example 3: Reading with STRUCT

In the memory area for fast data exchange, a structure for reading is used.

NC values

$A_DBB[0] = 11, $A_DBB[1] = 22, $A_DBW[2] = B#1a2b, $A_DBD[2] = B#3c4d5e6f

PLC programming

```
// S7Var_Struct : Struct
//           S7Var_Byte : Byte;
//           S7Var_Word : Word;
//           S7Var_DInt : DWord;
// END_STRUCT;
// #tempError: Bool
// #tempStatus: Int
```

```
"LBP_TransferSelData"(Enable := TRUE,    // transfer data
            Funct := 3,                  // read from NC
            S7Var := #S7Var_Struct,      // structure to which data is
                                         written
            IVAR1 := 0,                  // write from offset 0
            IVAR2 := -1,                 // without semaphore
            Error => #tempError,
            ErrCode => #tempStatus);
```

After the function has been executed, the structure is filled as follows:
`$A_DBB[1]` → `#S7Var_Struct.S7Var_Byte = 11`
`$A_DBW[2]` → `#S7Var_Struct.S7Var_Word = 16#2B1A`
`$A_DBD[4]` → `#S7Var_Struct.S7Var_DWord = 16#6F5E4D3C`

`$A_DBB[1]` cannot be addressed symbolically in the PLC but is located in the structure as a filler byte between `#S7Var_Struct.S7Var_Byte` and `#S7Var_Struct.S7Var_Word`.

To transfer `$A_DBW[2]` and `$A_DBD[4]` correctly, the byte sequences must be changed in `#S7Var_Struct.S7Var_Word` and `#S7Var_Struct.S7Var_DWord`:

```
#statS7Var_Struct.S7Var_Word :=
SWAP_WORD(#statS7Var_Struct.S7Var_Word);
#statS7Var_Struct.S7Var_DWord :=
SWAP_DWORD(#statS7Var_Struct.S7Var_DWord);
```

10.7.13.3 Function 5: Update control signals to channel

This function is used for fast transfer of important control signals. The signals "LBP_Chan*.DBW6" (DB21, ... DBX6.0 to DBX7.7) of the channel interface in the PLC are transferred to the NC. Transfer takes place immediately. In this way, for example, the feed disable or read-in disable can be transferred outside the cyclic data exchange (PLC-NC).

Signal	Category	Type	Value range	Meaning
Enable	I	BOOL	---	TRUE: Transfer data
Funct	I	BYTE	5	5: Control signals to channel
IVAR1	I	INT	1 ... 10	Channel number
Error	O	BOOL	---	TRUE: An error has occurred
ErrCode	O	INT	---	10: // channel number invalid

```
// #tempError: Bool
// #tempStatus: Int
"LBP_TransferSelData"(Enable := TRUE,    // transfer data
            Funct := 5,
            S7Var := NULL,               // not relevant
            IVAR1 := 1,                  // channel
```

```
                IVAR2 := 0,                 // not relevant
                Error => #tempError,
                ErrCode => #tempStatus);
```

10.7.13.4 Function 6: Update control signals to axes

This function is used for fast transfer of important control signals. The signals "LBP_Axis*.DBB2" (DB31, ... DBX2.0 to DBX2.7) of the axis interface in the PLC are transferred to the NC. Transfer takes place immediately and for all activated axes. This allows, for example, the controller enable to be transferred outside cyclic data exchange (PLC-NC).

Signal	Category	Type	Value range	Meaning
Enable	I	BOOL	---	TRUE: Transfer data
Funct	I	BYTE	6	6: Control signals (part 1) to axes
Error	O	BOOL	---	TRUE: An error has occurred
ErrCode	O	INT	---	Not used

Call example

```
// #tempError: Bool
// #tempStatus: Int
"LBP_TransferSelData"(Enable := TRUE,    // transfer data
             Funct := 6,
             S7Var := NULL,              // not relevant
             IVAR1 := 0,                 // not relevant
             IVAR2 := 0,                 // not relevant
             Error => #tempError,
             ErrCode => #tempStatus);
```

10.7.13.5 Function 7: Update control signals to axes

This function is used for fast transfer of important control signals. The signals "LBP_Axis*.DBB4" (DB31, ... DBX4.0 to DBX4.7) of the axis interface in the PLC are transferred to the NC. Transfer takes place immediately and for all activated axes. This allows, for example, the feed stop to be transferred outside cyclic data exchange (PLC-NC).

Signal	Category	Type	Value range	Meaning
Enable	I	BOOL	---	TRUE: Transfer data
Funct	I	BYTE	7	7: Control signals (part 2) to axes
Error	O	BOOL	---	TRUE: An error has occurred
ErrCode	O	INT	---	Not used

Call example

```
// #tempError: Bool
// #tempStatus: Int
"LBP_TransferSelData"(Enable := TRUE,    // transfer data
             Funct := 7,
             S7Var := NULL,              // not relevant
             IVAR1 := 0,                 // not relevant
             IVAR2 := 0,                 // not relevant
             Error => #tempError,
             ErrCode => #tempStatus);
```

10.7.14 LBP_MCPCtrlMillingSmall [FC24]: Transfer of the MCP signals on the interface

Function

The block "LBP_MCPCtrlMillingSmall [FC24]" transfers signals between a machine control panel (MCP) and the NC/PLC interface (NC DB, Mode group DB, Channel DB, Axis DB). The signals are transferred from the MCP (keys, etc.) to data blocks of the NC/PLC interface as well as signals from data blocks of the NC/PLC interface to the MCP (LEDs).

The block transfers the following signals for the M variant of the MCP 310 machine control panel:

- Keyswitch
- Modes
- Program control
- Axis traversing signals
- Spindle and feedrate override
- WCS/MCS switchover

The following figure shows the signal transfer between the MCP, NC/PLC interface and block outputs:

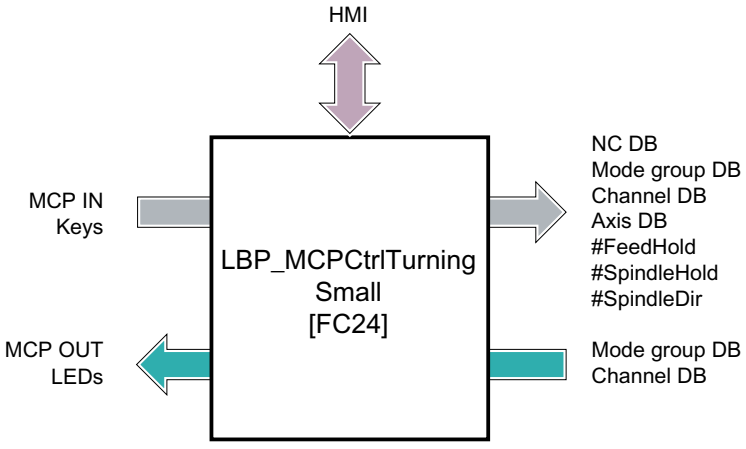


Figure 10-2 Signal transfer FC24

The memory areas MCP-IN and MCP-OUT are used for the signal transfer as specified in parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" when "LBP_ConfigBP [FC1]" is called. Parameter "MCPNum" has no effect on the function of the block. See also Section "Operation of several machine control panels" below and Chapter LBP_ConfigBP [FC1]: Basic program, startup section (Page 662).

Feed start/stop and spindle left/right/stop are not transferred to the NC/PLC interface, but output as output parameters "FeedHold", "SpindleDir" or "SpindleHold". The state is stored internally, the outputs are retained even after the key is pressed. The block controls the associated LEDs based on the state stored internally.

If the machine control panel fails, all signals of this machine control panel are overwritten with zero and the output parameters "FeedHold" and "SpindleHold" are set to TRUE.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
BAGNo:	I	BYTE	B#16#**0**0 - B#16#**0**A	**1st MCP**: Number of the mode group to which the mode signals are transferred B#16#00: No transfer of mode signals
			B#16#**1**0 - B#16#**1**A	**2nd MCP**: Number of the mode group to which the mode signals are transferred B#16#10: No transfer of mode signals
ChanNo:	I	BYTE	0 - 10	Number of the channel to which the channel signals are transferred 0: No transfer of channel signals
SpindleIFNo:	I	BYTE	0 - 31	Number of the machine axis to which the spindle data is transferred 0: No transfer of spindle data
FeedHold:	O	BOOL	---	Feed stop from MCP
SpindleHold:	O	BOOL	---	Spindle stop from MCP
SpindleDir:	O	BOOL	---	TRUE: Spindle direction counterclockwise FALSE: Spindle direction clockwise

Special features

Multiple calls of the block

To transfer signals to multiple mode groups, channels or spindles, the block can be called multiple times with the corresponding parameters "BAGNo", "ChanNo" and "SpindleIFNo".

At the first call of the block for a machine control panel, all signals are transferred to the data blocks of the NC/PLC interface and the LEDs activated. Now only signals to mode groups and channels are transferred in the following calls in the same PLC cycle.

Operation of several machine control panels

The block can process the signals of up to two machine control panels. To do this, the block must be called for each MCP. The first machine control panel must be called first. Then the second

machine control panel called. Key-operated switch position and feedrate override for all axes are only transferred on the first call in the cycle.

For each MCP for which the block is called, valid input and output areas must be specified as parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" in the call of "LBP_ConfigBP [FC1]" – irrespective of the value of parameter "MCPNum".

Machine coordinate system and workpiece coordinate system

You can switch between the "machine coordinate system" (MCS) and the "workpiece coordinate system" (WCS) by pressing the MCS/WCS key.

The axis traversing signals are transferred to the data block of the selected machine axis in MCS mode. See the table below "Manual traversing of machine axes (from MCP to NC/PLC interface)".

The axis traversing signals are transferred to the data block of the channel that is specified via the parameter "ChanNo" in WCS mode. See the table below "Manual traversing of geometry axes (from MCP to NC/PLC interface)".

No axis is selected after switchover to MCS or WCS via the machine control panel.

Flexible axis configuration

The assignment of axis selection keys to machine axes can be configured flexibly.

Each key can be assigned to an arbitrary machine axis. If more than six machine axes are to be selected, the assignment can be changed dynamically. The new assignment only takes effect when the axis selection is changed.

The assignment is made via the data block "LBP_NC [DB10]". The number of the machine axis that is to be selected by the ith axis selection key of the first machine control panel is entered in the assignment table "LBP_NC.MCP1AxisTbl[i]".

The assignment is entered in "LBP_NC.MCP2AxisTbl" for the keys of the second machine control panel.

If the value 0 is specified in an entry of the assignment tables, no axis is selected by the associated key.

The number of usable axis selection keys can be limited by a value "LBP_NC.MCP1MaxAxis" or "LBP_NC.MCP2MaxAxis". If the value 0 (default setting) is used, all axis selection keys assigned a machine axis can be used.

Machine axes 1 to 6 are preselected for axis selection keys 1 to 6.

Table 10-82 Example of flexible axis configuration

Key	Assignment	Axis
Direction key X axis	`LBP_NC.MCP1AxisTbl[1] := 7 (DB10.DBB8)`	7th machine axis
Direction key Y axis	`LBP_NC.MCP1AxisTbl[2] := 2 (DB10.DBB9)`	2nd machine axis

Key	Assignment	Axis
Direction key Z axis	LBP_NC.MCP1AxisTbl[3] := 0 (DB10.DBB10)	No axis
Direction key 4th Axis	LBP_NC.MCP1AxisTbl[4] := 1 (DB10.DBB11)	1st machine axis

Axis selection from the HMI

In addition to the six axis selection keys of the MCP, the active axis can also be selected at the HMI.

If an axis is being selected from the HMI, then no axes can be traversed using the direction keys. The axis selected from the HMI is only transferred to the MCP if the channel selected at the HMI and at the "LBP_MCPCtrlTurning [FC25]" match one another. When an axis is selected from the HMI, just the same as when selecting an axis with the axis selection keys of the MCP, initially, an attempt is made to find the assignment between the selected axis number and the actual machine axis in the axis table of the NC-DB. See Section "Flexible axis configuration". This means that when an axis is selected from the HMI, the same machine axis is selected as if the corresponding axis selection key of the MCP had been selected. If an assignment is not possible, then the axis number selected from the HMI is directly used as machine axis. As a consequence it is possible to select axes that cannot be selected via the MCP.

Single block mode

Pressing the Single Block key activates single block mode, pressing the key again deactivates single block mode. If the mode is called several times for one MCP, the block transfers single block mode state from the channel of the first call to the channels of the subsequent calls.

Increment selection

The signals of the increment keys are transferred to the mode group DB of the NC/PLC interface. The NC evaluates the increment signals from the mode group DB when LBP_NC.A_IncInModeGroup := TRUE is set. This is the default setting after system run-up.

Signals

Control signals from MCP to NC/PLC interface

Table 10-83 Key-operated switch (from MCP to NC/PLC interface)

MCP switch	Interface DB "LBP_NC [DB10]"
Position 0	A_Keyswitch0
Position 1	A_Keyswitch1
Position 2	A_Keyswitch2
Position 3	A_Keyswitch3

Table 10-84 Operating modes and machine functions (from MCP to NC/PLC interface)

MCP key	**Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) - Display for mode group 1**
AUTOMATIC	A_AUTO
MDI	A_MDA
JOG	A_JOG
REPOS	A_REPOS
REF	A_REF
TEACH IN	A_TEACHIN
INC 1 ... 1000, INC Var.	A_INC1 ... 1000, A_INCVar

Table 10-85 Manual traversing of machine axes (from MCP to NC/PLC interface)

MCP key	**Axis DB: "LBP_Axis1 [DB31], ..." (selected axis)**
Direction key +	A_Plus
Direction key -	A_Minus
Rapid traverse override	A_RapidTrOR

Table 10-86 Manual traversing of geometry axes (from MCP to NC/PLC interface)

MCP key	**Channel DB: "LBP_Chan [DB21], ..." (Parameter ChanNo & selected axis)**		
Direction key +	A_Geo[1].Plus	A_Geo[2].Plus	A_Geo[3].Plus
Direction key -	A_Geo[1].Minus	A_Geo[2].Minus	A_Geo[3].Minus
Rapid traverse override	A_Geo[1].RapidTrOR	A_Geo[2].RapidTrOR	A_Geo[3].RapidTrOR

Table 10-87 Feedrate and spindle speed override (from MCP to NC/PLC interface)

MCP rotary switch	**Channel DB: "LBP_Chan [DB21], ..." (parameter ChanNo)**
Feedrate override	A_FD_OR
MCP rotary switch	**all active axis DBs: "LBP_Axis1 [DB31], ..." (from 1st MCP only)**
Feedrate override	A_FD_OR
MCP keys	**Axis DB: "LBP_Axis1 [DB31], ..." (parameter SpindleIFNo)**
Spindle - Spindle 100% Spindle +	A_SpOR

Table 10-88 Channel signals (from MCP to NC/PLC interface)

MCP keys	**Channel DB: "LBP_Chan [DB21], ..." (Parameter ChanNo)**
NC Start	A_NCStart
NC Stop	A_NCStop
Reset	A_Reset
Single block	A_SBL

Table 10-89 Feed and spindle enable (from MCP to NC/PLC interface)

MCP keys	**Output parameter**
Feed stop Feed enable	Parameter: "FeedHold"
Spindle stop Spindle CCW Spindle CW	Parameter: "SpindleHold" Parameter: "SpindleDir"

Feedback signals to MCP from NC/PLC interface

Table 10-90 Operating modes and machine functions (to MCP from NC/PLC interface)

MCP LED	**Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) - Display for mode group 1**
AUTOMATIC	E_AUTO
MDI	E_MDA
JOG	E_JOG
REPOS	E_REPOS
REF	E_REF
TEACH IN	E_TEACHIN
INC 1 ... 1000, INC Var.	E_INC1 ... 1000, E_INCVar

Table 10-91 Channel signals (to MCP from NC/PLC interface)

MCP LED	**Channel DB: "LBP_Chan [DB21], ..." (parameter ChanNo)**
NC Start	E_ProgRunning
NC Stop	E_ProgStop bzw. E_ProgInterrupt
Reset	Not available
Single block	A_SBL *(same signal as key)*

Note

LEDs of the direction keys +/- and rapid traverse are switched by pressing the keys.

LEDs of the axis keys are activated by selecting an axis or MCS/WCS switchover.

The LEDs of the feed start/stop and spindle CCW/CW/stop keys are switched after the keys have been pressed.

The LEDs of spindle speed override keys +/-/100% are switched after the keys have been pressed.

The LEDs of WCS are switched over after the keys have been pressed.

The LED on the Reset key is not switched.

Alarm messages

The following alarms are defined for the diagnostics of parameterization errors:

- 404035: "Parameter BAGNo not permitted for LBP_MCPCtrlMillingSmall [FC24]"
- 404036: "Parameter ChanNo not permitted for LBP_MCPCtrlMillingSmall [FC24]"
- 404026: "Pointer machine control panel %1 incorrect"

Call examples

The following example shows a single call:

```
// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
// #tempFeedHold: Bool;
// #tempSpindleHold: Bool;
// #tempSpindleDir: Bool;
"LBP_MCPCtrlMillingSmall"(BAGNo := 1,    // or B#16#01
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold,
            SpindleDir => #tempSpindleDir);
```

The following example shows a multiple call:

```
// #tempFeedHold: Bool;
// #tempSpindleHold: Bool;
// #tempSpindleDir: Bool;

// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
"LBP_MCPCtrlMillingSmall"(BAGNo := 1,      // or B#16#01
```

```
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold,
            SpindleDir => #tempSpindleDir);

// Call of the block for the 1st MCP, without mode group, channel 2,
   without spindle
"LBP_MCPCtrlMillingSmall"(BAGNo := 0,      // or B#16#00
            ChanNo := 2,
            SpindleIFNo := 0,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold,
            SpindleDir => #tempSpindleDir);

// Call of the block for the 1st MCP, mode group 2, channel 3,
   spindle: Axis 8
"LBP_MCPCtrlMillingSmall"(BAGNo := 2,      // or B#16#02
            ChanNo := 3,
            SpindleIFNo := 8,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold,
            SpindleDir => #tempSpindleDir);
```

The following example shows a call for several MCPs:

```
// #tempFeedHoldMcp1: Bool;
// #tempSpindleHoldMcp1: Bool;
// #tempSpindleDirMcp1: Bool;
// #tempFeedHoldMcp2: Bool;
// #tempSpindleHoldMcp2: Bool;
// #tempSpindleDirMcp2: Bool;

// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
"LBP_MCPCtrlMillingSmall"(BAGNo := B#16#01,
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHoldMcp1,
            SpindleHold => #tempSpindleHoldMcp1,
            SpindleDir => #tempSpindleDirMcp1);

// Call of the block for the 2nd MCP, without mode group, channel 2,
   without spindle
```

```
"LBP_MCPCtrlMillingSmall"(BAGNo :=
B#16#10,
            ChanNo := 2,
            SpindleIFNo := 0,
            FeedHold => #tempFeedHoldMcp2,
            SpindleHold => #tempSpindleHoldMcp2,
            SpindleDir => #tempSpindleDirMcp2);
```

See also

LBP_MCPCtrlMilling [FC19]: Transfer of the MCP signals on the interface (Page 712)

10.7.15 LBP_MCPCtrlTurning [FC25]: Transferring the MCP/OP signals to the interface

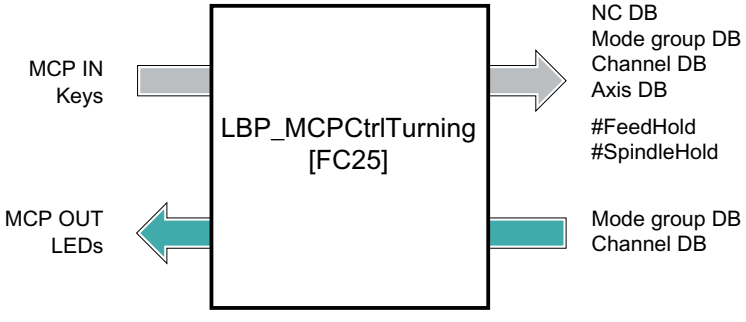
Function

The block "LBP_MCPCtrlTurning [FC25]" transfers signals between a machine control panel (MCP) and the NC/PLC interface (NC DB, Mode group DB, Channel DB, Axis DB). The signals are transferred from the MCP (keys, etc.) to data blocks of the NC/PLC interface as well as signals from data blocks of the NC/PLC interface to the MCP (LEDs).

The block transfers the following signals for the T variant of the MCP 483 machine control panel:

- Keyswitch
- Modes
- Program control
- Axis traversing signals
- Spindle and feedrate override
- WCS/MCS switchover

The following figure shows the signal transfer between the MCP, NC/PLC interface and block outputs:



The memory areas MCP-IN and MCP-OUT are used for signal transfer as specified in parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" when "LBP_ConfigBP [FC1"] is called. Parameter "MCPNum" has no effect on the function of the block. See also Section "Operation of several

machine control panels" below and LBP_ConfigBP [FC1]: Basic program, startup section (Page 662).

Feed start/stop and Spindle CCW/CW/stop are not transferred to the NC/PLC interface, but are output as output parameters "FeedHold" or "SpindleHold". The state is stored internally, the outputs are retained even after the key is pressed. The block controls the associated LEDs based on the state stored internally.

If the machine control panel fails, all signals of this machine control panel are overwritten with zero and output parameters "FeedHold" and "SpindleHold" are set to TRUE.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
BAGNo:	I	BYTE	B#16#**0**0 ... B#16#**0**A (0 ... 10)	**1st MCP**: Number of the mode group to which the mode signals are transferred B#16#00: No transfer of mode signals
			B#16#**1**0 ... B#16#**1**A	**2nd MCP**: Number of the mode group to which the mode signals are transferred B#16#10: No transfer of mode signals
ChanNo:	I	BYTE	0 ... 10	Number of the channel to which the channel signals are transferred 0: No transfer of channel signals
SpindleIFNo:	I	BYTE	0 ... 31	Number of the machine axis to which the spindle data is transferred 0: No transfer of spindle data
FeedHold :	O	BOOL	---	Feed stop from MCP
SpindleHold :	O	BOOL	---	Spindle stop from MCP

Special features

Multiple calls of the block

In order to transfer signals to several mode groups, channels or spindles, the block can be called several times with the corresponding parameters "BAGNo", "ChanNo" and "SpindleIFNo".

At the first call of the block for a machine control panel, all signals are transferred to the data blocks of the NC/PLC interface and the LEDs activated. Now only signals to mode groups and channels are transferred in the following calls in the same PLC cycle.

Operation of several machine control panels

The block can process the signals of up to two machine control panels. To do this, the block must be called for each MCP. The first machine control panel must be called first. Then the second machine control panel called. Key-operated switch position and feedrate override for all axes are only transferred on the first call in the cycle.

Valid input and output areas must be specified as parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" in the call of "LBP_ConfigBP [FC1]" for each MCP for which the block is called, irrespective of the value of parameter "MCPNum".

Machine coordinate system and workpiece coordinate system

You can switch between the "machine coordinate system" (MCS) and the "workpiece coordinate system" (WCS) by pressing the MCS/WCS key.

The axis traversing signals are transferred to the data block of each associated machine axis in MCS mode. See the table below "Manual traversing of machine axes (from MCP to NC/PLC interface)".

The axis traversing signals are transferred to the data block of the channel that is specified via the parameter "ChanNo" in WCS mode. The axis traversing keys +X/-X, +Y/-Y, +Z/-Z, are assigned to each associated geometry axis. See the table below "Manual traversing of geometry axes (from MCP to NC/PLC interface)".

Flexible axis configuration

The assignment of axis traversing keys to machine axes can be configured flexibly.

Each pair of axis traversing keys can be assigned any machine axis. If more than 4 machine axes are to be selected, the assignment can be changed dynamically. The assignment must not be changed while axes are being traversed.

The assignment is made via the data block "LBP_NC [DB10]". The number of the machine axis that is to be selected by the ith axis traversing key pair (X, Z, C, Y) of the first machine control panel is entered in assignment table "LBP_NC.MCP1AxisTbl[i]" (DB10 from DBB8).

The assignment for the key pairs of the second machine control panel is entered in "LBP_NC.MCP2AxisTbl" (DB10 from DBB32).

If the value 0 is specified in an entry of the assignment tables, this means that no axis is selected by the associated key pair.

The number of usable axis traversing key pairs can be limited by the value "LBP_NC.MCP1MaxAxis" (DB10.DBW30) or "LBP_NC.MCP2MaxAxis" (DB10.DBW54). If the value 0 (default setting) is used, all axis traversing key pairs to which a machine axis has been assigned can be used.

Machine axes 1 to 4 are preselected for axis traversing key pairs 1 to 4.

Table 10-92 Example of flexible axis configuration

Key	Assignment	Axis
Key pair +X/-X	LBP_NC.MCP1AxisTbl[1] := 7 (DB10.DBB8)	7th machine axis
Key pair +Z/-Z	LBP_NC.MCP1AxisTbl[2] := 2 (DB10.DBB9)	2nd machine axis
Key pair +C/-C	LBP_NC.MCP1AxisTbl[3] := 1 (DB10.DBB10)	1st machine axis
Key pair +Y/-Y	LBP_NC.MCP1AxisTbl[4] := 0 (DB10.DBB11)	No axis

Increment selection

The signals of the increment keys are transferred to the mode group DB of the NC/PLC interface. The NC evaluates the signals from the mode group DB when LBP_NC.A_IncInModeGroup := TRUE is set. This is the default setting after system run-up.

Signals

Control signals from MCP to NC/PLC interface

Table 10-93 Key-operated switch (from MCP to NC/PLC interface)

MCP switch	NC DB: "LBP_NC [DB10]"
Position 0	A_Keyswitch0 [DBX56.4]
Position 1	A_Keyswitch1 [DBX56.5]
Position 2	A_Keyswitch2 [DBX56.6]
Position 3	A_Keyswitch3 [DBX56.7]

Table 10-94 Operating modes and machine functions (from MCP to NC/PLC interface)

MCP key	Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) - Display for mode group 1
AUTOMATIC	A_AUTO [DBX0.0]
MDI	A_MDA [DBX0.1]
JOG	A_JOG [DBX0.2]
REPOS	A_REPOS [DBX1.1]
REF	A_REF [DBX1.2]
TEACH IN	A_TEACHIN [DBX1.0]
INC 1 ... 10 000, INC Var.	A_INC1 ... 10000, A_INCVar [DBX2.0 ... 2.5]

Table 10-95 Manual traversing of machine axes (from MCP to NC/PLC interface)

MCP key	Axis DB: "LBP_Axis1 [DB31], ..." (selected axis)
Traversing key +X, +Z, +C, +Y	A_Plus [DBX4.7]
Traversing key -X, -Z, -C, -Y	A_Minus [DBX4.6]
Rapid traverse override	A_RapidTrOR [DBX4.5]

Table 10-96 Manual traversing of geometry axes (from MCP to NC/PLC interface)

MCP key	Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)
Traversing key +X	A_Geo[1].Plus [DBX12.7]
Traversing key -X	A_Geo[1].Minus [DBX12.6]
Traversing key +Y	A_Geo[2].Plus [DBX16.7]
Traversing key -Y	A_Geo[2].Minus [DBX16.6]
Traversing key +Z	A_Geo[3].Plus [DBX20.7]

MCP key	**Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)**
Traversing key -Z	A_Geo[3].Minus [DBX20.6]
Rapid traverse override	A_Geo[1].RapidTrOR [DBX12.5] A_Geo[2].RapidTrOR [DBX16.5] A_Geo[3].RapidTrOR [DBX20.5]

Table 10-97 Feedrate and spindle speed override (from MCP to NC/PLC interface)

MCP rotary switch	**Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)**
Feedrate override	A_FD_OR [DBB4]
MCP rotary switch	**Axis DB: "LBP_Axis1 [DB31], ..." for X, Z, C, Y (from 1st MCP only)**
Feedrate override	A_FD_OR [DBB0]
MCP rotary switch	**Axis DB: "LBP_Axis1 [DB31], ..." (parameter SpindleIFNo)**
Spindle speed override	A_SpOR [DBB19]

Table 10-98 Channel signals (from MCP to NC/PLC interface)

MCP keys	**Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)**
NC Start	A_NCStart [DBX7.1]
NC Stop	A_NCStop [DBX7.3]
Reset	A_Reset [DBX7.7]
Single block	A_SBL [DBX0.4]

Table 10-99 Feed and spindle enable (from MCP to NC/PLC interface)

MCP keys	**Output parameter**
Feed stop Feed enable	Parameter: "FeedHold"
Spindle stop Spindle enable	Parameter: "SpindleHold"

Feedback signals to MCP from NC/PLC interface

Table 10-100 Operating modes and machine functions (to MCP from NC/PLC interface)

MCP LED	**Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) - Display for mode group 1**
AUTOMATIC	E_AUTO [DBX6.0]
MDI	E_MDA [DBX6.1]
JOG	E_JOG [DBX6.2]
REPOS	E_REPOS [DBX7.1]
REF	E_REF [DBX7.2]
TEACH IN	E_TEACHIN [DBX7.0]
INC 1 ... 10 000, INC Var.	E_INC1 ... 10000, E_INCVar [DBX8.0 ... 8.5]

Table 10-101 Channel signals (to MCP from NC/PLC interface)

MCP LED	**Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)**
NC Start	E_ProgRunning [DBX35.0]
NC Stop	E_ProgStop [DBX35.2] and E_ProgInterrupt [DBX35.3]
Reset	Not available
Single block	A_SBL [DBX0.4] *(same signal as key)*

Note

The LEDs of axis traversing key pairs X/Z/C/Y are switched when the keys are pressed.

The LEDs of the feed start/stop and spindle start/stop keys are switched after the keys have been pressed.

The LEDs of WCS are switched over after the keys have been pressed.

LEDs of rapid traverse and reset are not activated.

Alarm messages

The following alarms are defined for the diagnostics of parameterization errors:

- 404033: "Parameter BAGNo not permitted for LBP_MCPCtrlTurning [FC25]"
- 404034: "Parameter ChanNo not permitted for LBP_MCPCtrlTurning [FC25]"
- 404026: "Pointer machine control panel %1 incorrect"

Call examples

The following example shows a single call:

```
// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
// #tempFeedHold: Bool;
// #tempSpindleHold: Bool;
"LBP_MCPCtrlTurning"(BAGNo := 1,      // or B#16#01
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);
```

The following example shows a multiple call:

```
// #tempFeedHold: Bool;
// #tempSpindleHold: Bool;

// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
"LBP_MCPCtrlTurning"(BAGNo := 1,       // or B#16#01
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);

// Call of the block for the 1st MCP, without mode group, channel 2,
   without spindle
"LBP_MCPCtrlTurning"(BAGNo := 0,       // or B#16#00
            ChanNo := 2,
            SpindleIFNo := 0,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);

// Call of the block for the 1st MCP, mode group 2, channel 3,
   spindle: Axis 8
"LBP_MCPCtrlTurning"(BAGNo := 2,       // or B#16#02
            ChanNo := 3,
            SpindleIFNo := 8,
            FeedHold => #tempFeedHold,
            SpindleHold => #tempSpindleHold);
```

The following example shows a call for several MCPs:

```
// #tempFeedHoldMcp1: Bool;
// #tempSpindleHoldMcp1: Bool;
// #tempFeedHoldMcp2: Bool;
// #tempSpindleHoldMcp2: Bool;

// Call of the block for the 1st MCP, mode group 1, channel 1,
   spindle: Axis 4
"LBP_MCPCtrlTurning"(BAGNo := 1,
            ChanNo := 1,
            SpindleIFNo := 4,
            FeedHold => #tempFeedHoldMcp1,
            SpindleHold => #tempSpindleHoldMcp1);

// Call of the block for the 2nd MCP, without mode group, channel 2,
   without spindle
"LBP_MCPCtrlTurning"(BAGNo := 10,
            ChanNo := 2,
            SpindleIFNo := 0,
            FeedHold => #tempFeedHoldMcp2,
            SpindleHold => #tempSpindleHoldMcp2);
```

10.7.16 LBP_HTCtrlHT8 [FC26]: Transferring the HT 8 / HT 10 signals to the interface

Function

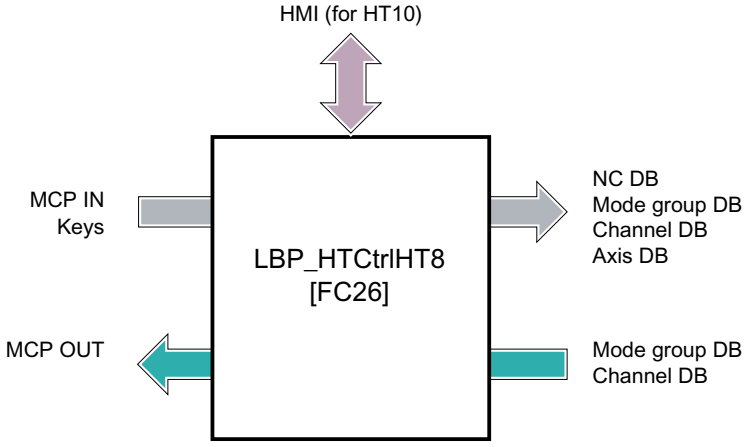
Block "LBP_HTCtrlHT8 [FC26]" transfers signals between the machine control panel (MCP) of the HT 8 or the HT 10 and the NC/PLC interface (NC-DB, mode group DB, channel DB, axis DB). The signals are transferred from the HT 8 or the HT 10 (keys etc.) to the data blocks of the NC/PLC interface and the signals from the data blocks of the NC/PLC interface are transferred to the HT 8 or the HT 10.

The block transfers the following signals for the HT 8 or the HT 10:

- Modes
- Program control
- Axis traversing signals
- Feedrate override
- WCS/MCS switchover

Bit 6.7 in the input image is used to make a differentiation between the HT 8 and HT 10. This bit is set for an HT 10.

The following figure shows the signal transfer between the MCP, NC/PLC interface and block outputs:



The memory areas MCP-IN and MCP-OUT are used for the signal transfer as specified in the parameters "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out" in the call of "LBP_ConfigBP [FC1]". The parameter "MCPNum" has no effect on the function of the block. See also Section "Operation of several machine control panels" and STRUCT "Magazines" magazine data (Page 663) below.

If the HT 8 or the HT 10 fails, all signals of this machine control panel are overwritten with zero.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
BAGNo	I	BYTE	B#16#**0**0 ... B#16#**0**A (0 ... 10)	**1st MCP**: Number of the mode group to which the mode signals are transferred B#16#00: No transfer of mode signals
			B#16#**1**0 ... B#16#**1**A	**2nd MCP**: Number of the mode group to which the mode signals are transferred B#16#10: No transfer of mode signals
ChanNo	I	BYTE	(0 ... 10)	Number of the channel to which the channel signals are transferred 0: channel signals are not transferred (for the HT 10, 0 means that no axis can be traversed in the JOG mode)

Special features

Keys as softkeys

Not all of the keys of the HT8 or the HT 10 described here are physical keys on the device. The keys MCS/WCS, Single Block and Inc. Var are softkeys that are called up with the U key.

Multiple calls of the block

To transfer signals to multiple mode groups, channels or spindles, the block can be called multiple times with the corresponding parameters, "BAGNo" and "ChanNo".

In the first call of the block for an HT 8 or the HT 10, all signals are transferred to the data blocks of the NC/PLC interface and the output signals are controlled. Now only signals to mode groups and channels are transferred in the following calls in the same PLC cycle.

Operation of several machine control panels

The block can process signals from up to two HT 8 or HT 10 devices. This means that a maximum of two devices can be used. Mixed (two different devices) operation is also possible. The block must be called for each HT 8 or HT 10. The HT 8 or the HT 10 must be called at interface MCP1 first. The HT 8 or the HT 10 can then be called at interface MCP2.

Valid input and output ranges must be parameterized for every HT 8 and HT 10 for which the block is called. When you call "LBP_ConfigBP [FC1]", enter the input and output ranges as parameter "MCP1In", "MCP1Out" or "MCP2In", "MCP2Out". These parameter settings are independent of the value of parameter "MCPNum".

Machine coordinate system and workpiece coordinate system

You can switch between the "Machine Coordinate System" (MCS) and "Workpiece Coordinate System" (WCS) by pressing the MCS/WCS softkey.

The axis traversing signals are transferred to the data block of each associated machine axis in MCS mode. See the table below for manual traversing (MCS) of machine axes.

The axis traversing signals are transferred to the data block of the channel that is specified via the parameter "ChanNo" in WCS mode.

For the HT 8, the first three axis traversing key pairs are assigned to the relevant geometry axis. See the table below for manual traversing (WCS) of geometry axes.

For the HT 10, only one geo axis can be traversed, which was selected from the HMI.

In WCS mode, the axis traversing key pairs 1 to 3 control the geometry axes, pairs 4 to 6 control machine axes 4 to 6. This behavior of axis traversing key pairs 4 to 6 in WCS mode can be configured. If at least one of the output signals %A3.7 or %A2.6 is activated in WCS mode, no machine axes can be traversed with axis traversing key pairs 4 to 6.

For the HT 10, the axis traversing signals (+/- and rapid traverse) refer to a machine axis or a geo axis, depending on which axis was selected at the HMI and whether WCS or MCS is active. See also "Flexible axis configuration for the HT 10".

Flexible axis configuration for the HT 8

The assignment of axis traversing keys to machine axes can be configured flexibly.

Each pair of axis traversing keys can be assigned any machine axis. If more than six machine axes are to be used, the assignment can be changed dynamically. The assignment must not be changed while axes are traversing.

The assignment is made in data block "LBP_NC [DB10]". The number of the machine axis that is to be selected by the ith axis traversing key pair of the first HT 8 is entered in the assignment table "LBP_NC.MCP1AxisTbl[i]".

The assignment for the key pairs of the second HT 8 is entered in "LBP_NC.MCP2AxisTbl" (DB10 from DBB8).

If the value 0 is specified in an entry of the assignment tables, this means that no axis is selected by the associated key pair.

The number of usable axis traversing key pairs can be limited by the value "LBP_NC.MCP1MaxAxis" (DB10.DBW30) or "LBP_NC.MCP2MaxAxis" (DB10.DBW54). If the value 0 (default setting) is used, all axis traversing key pairs to which a machine axis has been assigned can be used.

Machine axes 1 to 6 are preselected for axis traversing key pairs 1 to 6.

The following table shows an example of a flexible axis configuration.

Key	Assignment	Axis
1st key pair +/-	"LBP_NC.MCP1AxisTbl[1]" := 7 (DB10.DBB8)	7th machine axis
2nd key pair +/-	"LBP_NC.MCP1AxisTbl[2]" := 2 (DB10.DBB9)	2nd machine axis
3rd key pair +/-	"LBP_NC.MCP1AxisTbl[3]" := 1 (DB10.DBB10)	1st machine axis
4th key pair +/-	"LBP_NC.MCP1AxisTbl[4]" := 0 (DB10.DBB11)	No axis

Flexible axis configuration for the HT 10

Contrary to the HT 8, the HT 10 only has one plus and one minus key for traversing. There is also a rapid key for rapid traversing. The active axis, to which these traversing signals refer, must be selected at the HMI.

If an axis is being selected from the HMI, then no axis can be traversed using the direction keys. The axis selected from the HMI is only transferred to the MCP if the channel selected at the HMI and at the "LBP_HTCtrlHT8" match one another.

If "LBP_HTCtrlHT8" is called for an HT 10 with ChanNo <= 0, then no axes can be traversed as the current axis cannot be determined from the channel DB.

Single block mode

Pressing the softkey activates single block mode, pressing the key again deactivates single block mode. If the mode is called several times for one MCP, the block transfers single block mode state from the channel of the first call to the channels of the subsequent calls.

Increment selection

The signals of the increment softkey are transferred to the mode group DB of the NC/PLC interface. The NC evaluates the increment signals from the mode group DB when "LBP_NC.A_IncInModeGroup" := TRUE (DB10.DBX57.0) is set. This is the default setting after system run-up.

Signals

Control signals from MCP to NC/PLC interface

The following table shows the signals of the operating modes and machine functions (from MCP to NC/PLC interface):

MCP key	Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) – shown for mode group 1
AUTOMATIC	A_AUTO (DBX0.0)
MDI	A_MDA (DBX0.1)
JOG	A_JOG (DBX0.2)
REPOS	A_REPOS (DBX1.1)
REF	A_REF (DBX1.2)
TEACH IN	A_TEACHIN (DBX1.0)
INC Var	A_INCVar (DBX2.5)

The following table shows the signals for manual traversing (MCS) of machine axes (from MCP to NC/PLC interface):

MCP key	Axis DB: "LBP_Axis1 [DB31], ..." (associated axis)
Traversing keys +	A_Plus (DBX4.7)
Traversing keys -	A_Minus (DBX4.6)
Rapid traverse override	A_RapidTrOR (DBX4.5); only HT 10

The following table shows the signals for manual traversing (WCS) of geometry axes (from MCP to NC/PLC interface) for the HT 8

MCP key	Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)
1st traversing key +	A_Geo[1].Plus (DBX12.7)
1st traversing key -	A_Geo[1].Minus (DBX12.6)
2nd traversing key +	A_Geo[2].Plus (DBX16.7)
2nd traversing key -	A_Geo[2].Minus (DBX16.6)
3rd traversing key +	A_Geo[3].Plus (DBX20.7)
3rd traversing key -	A_Geo[3].Minus (DBX20.6)

The following table shows the signals for manual traversing (WCS) of geometry axes (from MCP to NC/PLC interface) for the HT 10

MCP key	Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)
Traversing keys +	A_Geo[n].Plus (DBX12.7, DBX16.7, DBX20.7)
Traversing keys -	A_Geo[n].Minus (DBX12.6, DBX16.6, DBX20.6)
Rapid traverse override	A_Geo[n].RapidTrOR (DBX12.5, DBX16.5, DBX20.5)

The following tables show the signals for feedrate and spindle speed override (from MCP to NC/PLC interface):

MCP rotary switch	Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)
Feedrate override	A_FD_OR (DBB4)

MCP rotary switch	All active axis DBs: "LBP_Axis1 [DB31], ..." (from 1st MCP only)
Feedrate override	A_FD_OR (DBB0)

The following table shows the channel signals (from MCP to NC/PLC interface):

MCP keys	Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)
NC Start	A_NCStart (DBX7.1)
NC Stop	A_NCStop (DBX7.3)
Reset	A_Reset (DBX7.7)
Single block	A_SBL (DBX0.4)

Feedback signals to MCP from NC/PLC interface

The following table shows the operating modes and machine functions (to MCP from NC/PLC interface):

MCP output signal	Mode group DB: "LBP_ModeGroup [DB11]" (parameter BAGNo) – shown for mode group 1
AUTOMATIC	E_AUTO (DBX6.0)
MDI	E_MDA (DBX6.1)
JOG	E_JOG (DBX6.2)
REPOS	E_REPOS (DBX7.1)
REF	E_REF (DBX7.2)
TEACH IN	E_TEACHIN (DBX7.0)
INC Var	E_INCVar (DBX8.5)

The following table shows the channel signals (to MCP from NC/PLC interface):

MCP output signal	Channel DB: "LBP_Chan1 [DB21], ..." (parameter ChanNo)
NC Start	E_ProgRunning (DBX35.0)
NC Stop	E_ProgStop (DBX35.2) and E_ProgInterrupt (DBX35.3)
Reset	E_ChanReset (DBX35.7); only for the HT 10
Single block	A_SBL (DBX0.4) (same signal as softkey)

Note

Unlike other MCP models, the HT 8 does not indicate the feedback messages via output signals with LEDs.

Output signals from the axis traversing key pairs are not switched.

Output signals from the WCS and single block are switched after the keys have been pressed.

The output signal from Reset is not switched. (HT 8)

The LEDs of the direction keys +/- and rapid traverse are controlled by pressing the keys. (HT 10)

Alarm messages

The following alarms are defined for the diagnostics of parameterization errors:

404038: "Parameter BAGNo not permitted for LBP_HTCtrlHT8 [FC26]"
404039: "Parameter ChanNo not permitted for LBP_HTCtrlHT8 [FC26]"
404026: "Pointer machine control panel %1 incorrect"

Call examples

Single call

```
// calls block for 1st MCP, Mode Group 1, Channel 1
"LBP_HTCtrlHT8"(BAGNo := 1,    // or B#16#01
             ChanNo   := 1);
```

Multiple call

```
// calls block for 1st MCP, Mode Group 1, Channel 1
"LBP_HTCtrlHT8"(BAGNo := 1,    // or B#16#01
             ChanNo   := 1);

// calls block for 1st MCP, without Mode Group, Channel 2
"LBP_HTCtrlHT8"(BAGNo := 0,    // or B#16#00
             ChanNo   := 2);

// calls block for 1st MCP, Mode Group 2, Channel 3
"LBP_HTCtrlHT8"(BAGNo := 2,    // or B#16#02
             ChanNo   := 3);
```

Call for multiple MCPs

```
// calls block for 1st MCP, Mode Group 1, Channel 1
"LBP_HTCtrlHT8"(BAGNo := B#16#01,
             ChanNo   := 1);

// calls block for 2nd MCP, without Mode Group, Channel 2
"LBP_HTCtrlHT8"(BAGNo := B#16#10,
             ChanNo   := 2);
```

10.7.17 LBP_Handwheels [FC25002]: transfer handwheel signals

Function

Block "LBP_Handwheels [FC25002]" transfers handwheel signals (rotary pulses from the handwheel) from the PLC to the NCK. Handwheel signals are transferred from a machine control panel (MCP) or a handheld terminal (HT) to the PLC in block "LBP_OpUnitComm [FB25000]". Both blocks must be called cyclically.

"LBP_Handwheels" uses the instance of block "LBP_OpUnitComm" to transfer the handwheel signals and transfers the handwheel signals as soon as new data are received from an MCP or HT.

So that the NCK processes the handwheel signals, machine data MD11350[i] := 6 must be set for every one of the required handwheels i and the assignment of the transfer interfaces 1 to 6 on the NCK to the handwheels must be made in machine data MD11352[i]. For a 1:1 assignment, MD11352[i] := i can be set for handwheels 1 to 6.

Description of formal parameters

Signal	Category	Type	Value range	Meaning
Comm	I	"LBP_OpUnitComm"	---	Instance data of "LBP_OpUnitComm [FB25000]"

Call examples

Single call

```
// OpUnitComm_DB: "LBP_OpUnitComm" (instance block)
// #tempError: Bool;
// #tempStatus: Word;
OpUnitComm_DB(Config := LBP_ConfigData,
              Error    => #tempError,
              Status   => #tempStatus);
LBP_Handwheels(OpUnitComm_DB);
```

10.8 General information on configuration and commissioning

10.8.1 Configuration machine control panel, handheld unit, direct keys

Communication with machine control panels (MCP), handheld units (HT) and direct key modules (OpKey) is realized via PROFINET or Industrial Ethernet (IE). PROFIBUS communication depends on the device and is not possible with all SINUMERIK devices.

The input data from the operator components (e.g. keys) are transferred to the PLC. The output data from the PLC (e.g. LEDs) are transferred to the operator component. These data are processed by the blocks "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]", "LBP_MCPCtrlTurning [FC25]" and "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK-PLC) or by the user program.

In the user program, you define which memory areas the input and output data are to be saved to. You pass the pointers to these memory areas as parameters of the function "LBP_ConfigBP [FC1]" when it is called in "Startup [OB100]".

As soon as new data have been received by the operator component, the input data are written to the memory area that is referenced by the associated pointer to "LBP_ConfigBP [FC1]". Now the input data are available to the user and to the basic program.

The output data in the parameterized memory area are sent cyclically in defined intervals to the operator components.

10.8.1.1 Coupling via PROFINET

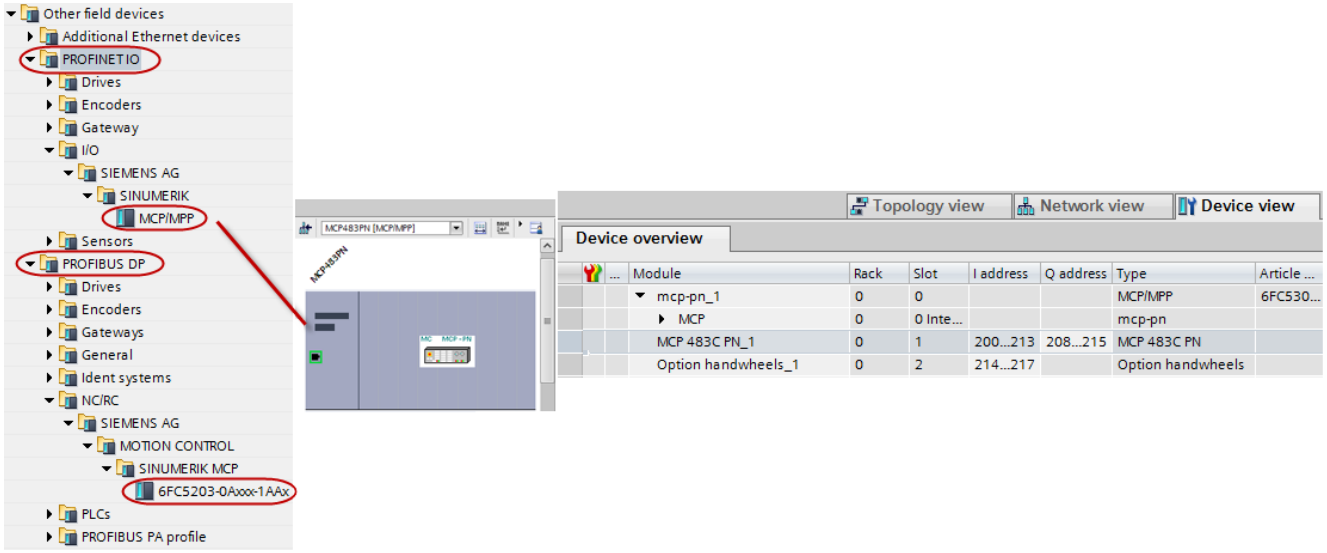
The PLC operating system is used to configure and establish a PROFINET connection to the operating components.

To enable the basic program to access the input and output data of the operator components, the pointers used in the "LBP_ConfigBP [FC1]" parameters must be made known to the basic program.

Example PROFINET-MCP

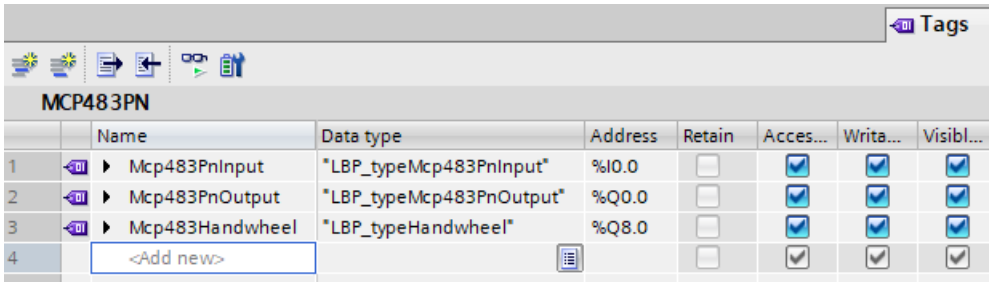
A PROFINET-MCP is created in the device configuration.

The figure below shows the hardware catalog and a configured MCP483PN with handwheel option:



Using a PLC variable table, you can combine the input and output images with the relevant MCP data types. This is a user-friendly method of symbolically addressing the inputs and outputs of the MCP.

The following screenshot shows a variable table for a MCP483PN:



See also

LBP_typeMcp* (Page 95)

10.8.1.2 Coupling via PROFIBUS

The PLC operating system is used to configure and establish a PROFIBUS connection to the operating components.

To enable the basic program to access the input and output data of the operator components, the pointers used in the "LBP_ConfigBP [FC1]" parameters must be made known to the basic program.

Note

In the virtual Sinumerik, a Profibus-MCP is simulated with the standard input and output signals which are permanently configured to I0.0 / Q0.0.

See also

Coupling via PROFINET (Page 752)

10.8.1.3 Connection via Industrial Ethernet

The block "LBP_OpUnitComm [FB25000]" is used for communicating with machine control panels (MCP), handheld terminals (HT) and direct key modules (OpKey) via Industrial Ethernet (IE). This block supports simultaneous operation of up to two machine control panels, one handheld terminal, and two direct key modules via Industrial Ethernet (IE).

Dedicated parameter sets exist for each control component. The user must define pointers for the input and output data in these parameter sets. These input and output pointers are also used in "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]", "LBP_MCPCtrlTurning [FC25]" and "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK-PLC). More information can be found in the operator component manual.

The parameters for the configuration can be set in the startup OB (e.g. "Startup [OB100]") via "LBP_ConfigBP [FC1]". Additional factors or a changed configuration of the connections can be made in the cycle via the parameters in "LBP_ConfigData [DB7]".

The connections to the operator components are created automatically based on the configuration in "LBP_ConfigData [DB7]" and monitored. The block writes the input data received from the operator components to the configured pointers from "LBP_ConfigData [DB7]". The block reads the output data sent to the operator components from the configured pointers from "LBP_ConfigData [DB7]". Status signals of the connections to the operator components are written to "LBP_NC [DB10]".

Cyclic call

"LBP_OpUnitComm [FB25000]" must be called cyclically with the instance block. The connections to the operator components are configured in "LBP_ConfigData [DB7]". Enter this data block when calling a block from "LBP_OpUnitComm [FB25000]" at the Config input parameter.

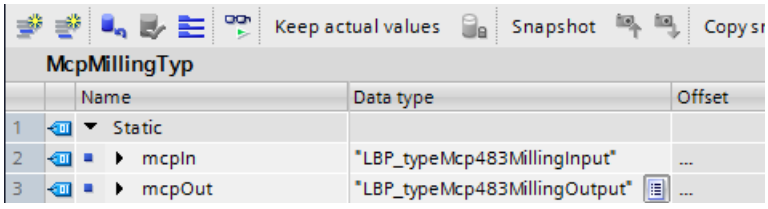
The call of "LBP_OpUnitComm [FB25000]" in the PLC cycle should be performed before calling "LBP_MCPCtrlMilling [FC19]", "LBP_MCPCtrlMillingSmall [FC24]", "LBP_MCPCtrlTurning [FC25]", "LBP_HTCtrlHT8 [FC26]" and "LBP_HTCtrlHT2 [FC13]" or before accessing the input data, for example after calling "LBP_MainBP [FC2]". This means that actual operator component data is available for subsequent program parts.

Example of the use of MCP data types to configure an MCP M version

In this example, the MCP1In and MCP1Out pointers of"LBP_ConfigBP [FC1]"are combined with data block areas that use the MCP data types provided in the basic program library.

To do this, create a global non-optimized DB and pass the initial addresses of the input and output images to the relevant parameters.

The following screenshot shows the input and output image in the DB for MCP 483 (M version) using the data types provided:



Parameter supply at"LBP_ConfigBP [FC1]":

```
"LBP_ConfigBP"  MCPNum:=0,
(
                MCP1In:="McpMillingTyp".mcpIn,
                MCP1Out:="McpMillingTyp".mcpOut,
                MCP1BusAdr:=6,
                MCP1Timeout:=S5T#700MS,
                MCP2In:=NULL,
                MCP2Out:=NULL,
                MCP2BusAdr:=0,
                MCP2Timeout:=S5T#700MS,
                MCP1Stop:=FALSE,
                MCP2Stop:=FALSE,
```

```
MCP1NotSend:=FALSE,
MCP2NotSend:=FALSE,
MCPBusType:=B#16#5,
...
```

Response in the event of a communication error

Alarms are generated when an operator component fails or a connection to the operator component is interrupted. Connection-specific error information is stored in parameters in"LBP_ConfigData [DB7]" . The input signals of the operator component (MCP1In, MCP2In, HTIn, Op1KeyIn, Op2KeyIn) are initialized with 16#00. If it is possible to establish a new connection between the PLC and the operator component, communication is resumed automatically and the error message is deleted.

Identification of operator components

The operator components can be identified via the parameters of the Ident interface in"LBP_ConfigData [DB7]". In this way, the type of an operator component and the required length of the input and output data can be determined.

See also

LBP_typeMcp* (Page 95)

Configuration

The Ethernet connections are parameterized via the operator component parameters in "LBP_ConfigData [DB7]" and can take place during the startup OB (e.g. "Startup [OB100]") on "LBP_ConfigBP [FC1]". The "LBP_ConfigBP [FC1]" parameters and "LBP_ConfigData [DB7]" variables have identical symbols.

The "LBP_OpUnitComm [FB25000]" block is not permanently connected to "LBP_ConfigData [DB7]". Therefore, the data block must be specified at the Config parameter. Currently, only "LBP_ConfigData [DB7]" is supported as a configuration block.

Activation

The respective operator component is activated during the startup OB via the number of the machine control panels (parameter MCPNum), for the handheld terminal via the parameter HTIf and via the number of direct keys (parameter OpKeyNum) on "LBP_ConfigBP [FC1]".

The parameter OpUnitCommId must be permanently set to 1.

The hardware ID of the Ethernet interface must be specified at parameter OpUnitInterfaceId. In "LBP_ConfigData [DB7]" the parameter is preset to 262.

These parameters cannot be changed in the cycle.

Note

You will learn the right hardware ID of the Ethernet interface from the TIA project under "Device configuration > CP > Ethernet interface > HW ID" or from the interface to which the operator components are connected.

The hardware ID of the Ethernet interface must be determined, not the hardware ID of the port. The Ethernet interface of the CP [X120, X130, X127] of an NCU 1750 has the hardware ID "Local~CP~Ethernet-Schnittstelle_1" (262).

DNS configuration

The logical device names of the control components must be resolved internally into the appropriate IP addresses. The resolution is done automatically by the integrated CP.

Coupling of machine control panels and handheld units

Configure the numeric part of the logical name of the operator component in the variables of "LBP_ConfigData [DB7]" in MCP1BusAdr, MCP2BusAdr or HTAdr (corresponds to the bus address of the node). The logical name is defined via switches on the MCP or terminal box.

The variables MCP1In, MCP2In and HTIn must be assigned with pointers that point to the memory area for the input data that are received from the operator component.

Note

Ensure that sufficient memory is available for the input and output data when the pointers are specified, the memory areas do not overlap, and no other data is present in these memory areas. Existing data is overwritten.

The variables MCP1Out, MCP2Out and HTOut must be assigned with pointers that point to the memory area for the output data that are sent to the operator component.

MCP1ConnectionId, MCP2ConnectionId and HTConnectionId are unambiguous connection identifiers that are used by "LBP_OpUnitComm [FB25000]" for communicating with the operator components. The parameters at "LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]" are preset with default values.

The variable MCPBusType specifies which communication connection is used (IE: B#16#5).

MCP1LocalUdpPort, MCP2LocalUdpPort and HTLocalUdpPort are unique UDP ports that are used by "LBP_OpUnitComm [FB25000]" for communicating with the operator components. The parameters at "LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]" are preset with default values.

Note

It is not permissible to utilize the connection identifiers being used for other communication connections in the PLC project.

It is not permissible to utilize the local UDP ports being used for other UDP connections at this network interface in the PLC project.

MCP1Stop, MCP2Stop and HTStop are used to interrupt the connections to the operator components.

With MCP1NotSend, MCP2NotSend and HTNotSend, no output data is sent to the operator components. The connection remains intact, only data is received.

Relevant parameters for operator components ("LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]")		
General		
OpUnitCommId = 1		
OpUnitInterfaceId = "Local~CP~Ethernet interface_1" (262, CP interface)		
MCP		**HT**
MCPNum = 0, 1 or 2 (number of MCPs)		HTIf = 5 (HT via Ethernet)
MCP1In	MCP2In	HTIn
MCP1Out	MCP2Out	HTOut
MCP1BusAdr	MCP2BusAdr	HTAdr
MCP1Stop	MCP2Stop	HTStop
MCP1NotSend	MCP2NotSend	HTNotSend
MCPBusType = B#16#55 for MCP1 and MCP2		
MCP1ConnectionId = 1001	MCP2ConnectionId = 1002	HTConnectionId = 1003
MCP1LocalUdpPort = 16001	MCP2LocalUdpPort = 16002	HTLocalUdpPort = 16003

The status of the respective operator component interfaces can be read in the following data blocks.

"LBP_NC [DB10]":

- E_MCP1ready (MCP1 ready)
- E_MCP2ready (MCP2 ready)
- E_HTready (HT ready)

"LBP_ConfigData [DB7]":

- MCP1Ready (MCP1 ready)
- MCP2Ready (MCP2 ready)
- HTReady (HT ready)

Warning of connection termination

To ensure a quick response to an impending termination of the connection between PLC and MCP, signals are set shortly after the cyclic communication between PLC and MCP has failed. If there is still no cyclic communication between PLC and MCP, the connection is terminated.

The signals for warning of a connection termination are set in "LBP_NC [DB10]":

- E_MCP1LostPackageWarn (MCP1)
- E_MCP2LostPackageWarn (MCP2)
- E_HTLostPackageWarn (HT)

The signals are reset after a connection is terminated or when there is a new connection.

Alarms MCP, HT

An error entry is also made in the PLC alarm buffer for timeouts. This gives rise to the following error messages:

- 404027: Failure of machine control panel %1
- 404028: Failure of manual operating device

The failure of an operator component is also detected after the restart if no data is exchanged between the operator component and the PLC.

Error information can be read in "LBP_ConfigData [DB7]":

- MCP1Error and MCP1Status (MCP1)
- MCP2Error and MCP2Status (MCP2)
- HTError and HTStatus (HT)

Coupling of OP with direct keys

Configure the numeric part of the logical name of the operator component in the variables of "LBP_ConfigData [DB7]" in Op1KeyBusAdr or Op2KeyBusAdr (corresponds to the bus address of the node). The logical name is defined via switches on the MCP or terminal box.

The variables Op1KeyIn and Op2KeyIn must be assigned with pointers that point to the memory area for the input data that are received from the operator component.

Note

Ensure that sufficient memory is available for the input and output data when the pointers are specified, the memory areas do not overlap, and no other data is present in these memory areas. Existing data is overwritten.

The variables Op1KeyOut and Op2KeyOut must be assigned with pointers that point to the memory area for the output data that are sent to the operator component.

Op1KeyConnectionId and Op2KeyConnectionId are unique connection identifiers that are used by "LBP_OpUnitComm [FB25000]" for communicating with the operator components. The parameters at "LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]" are preset with default values.

Op1KeyLocalUdpPort and Op2KeyLocalUdpPort are unambiguous UDP ports that are used by "LBP_OpUnitComm [FB25000]" for communicating with the operator components. The parameters at "LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]" are preset with default values.

Note

It is not permissible to utilize the connection identifiers being used for other communication connections in the PLC project.

It is not permissible to utilize the local UDP ports being used for other UDP connections at this network interface in the PLC project.

Op1KeyStop and Op2KeyStop are used to interrupt the connections to the operator components.

With Op1KeyNotSend and Op2KeyNotSend, no output data is sent to the operator components. The connection remains intact, only data is received.

The transport of the user data of the direct keys runs in the same way as in the case of Ethernet MCP. During the stop phase (Op1KeyStop or Op2KeyStop), the bus address of the direct key module (TCU index or bus address of the MCP) and the input and output pointers can be changed. After resetting Op1KeyStop or Op2KeyStop, a connection to the new bus address is established.

Relevant parameters for operator components ("LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]")	
General	
OpUnitCommId = 1	
OpUnitInterfaceId = "Local~CP~Ethernet interface_1" (262, CP interface)	
Direct keys	
OpKeyNum = 0, 1 or 2 (number of OPs with direct keys)	
Op1KeyIn	Op2KeyIn
Op1KeyOut	Op2KeyOut
OpKey1BusAdr	Op2KeyBusAdr
Op1KeyStop	Op2KeyStop
Op1KeyNotSend	Op2KeyNotSend
Op1KeyConnectionId = 1004	Op2KeyConnectionId = 1005
Op1KeyLocalUdpPort = 16004	Op2KeyLocalUdpPort = 16005

The status of the respective operator component interfaces can be read in the following data blocks.

"LBP_NC [DB10]":

- E_Op1KeyReady (OP1Key ready)
- E_Op2KeyReady (OP2Key ready)

"LBP_ConfigData [DB7]":

- Op1KeyReady (OP1Key ready)
- Op2KeyReady (OP2Key ready)

Bus addresses of direct keys

For the parameter OpKeyBusAdr, the TCU index is normally to be used. This affects the OPs, such as OP 08T, OP 12T, which for the direct keys do not have special cable connection to an Ethernet MCP.

If OPs with direct keys have a special cable connection and these are connected to an Ethernet-MCP, then for the parameter OpKeyBusAdr the bus address of the MCP (DIP-switch setting of the MCP) is to be used. Only the data stream of the direct keys (2 bytes) is transferred via the direct key interface.

Warning of connection termination

To ensure a quick response to an impending termination of the connection between PLC and direct keys, signals are set shortly after the cyclic communicating between PLC and direct keys has failed. If there is still no cyclic communication between PLC and direct keys, the connection is terminated.

The signals for warning of a connection termination are set in "LBP_NC [DB10]".

- E_Op1KeyLostPackageWarn (Op1Key)
- E_Op2KeyLostPackageWarn (Op2Key)

The signals are reset after a connection is terminated or when there is a new connection.

Alarm direct keys

An error entry is also made in the PLC alarm buffer for timeouts. This gives rise to the following error message:

- 404029: Direct keys %1 failed

The failure of an operator component is also detected after the restart if no data is exchanged between the operator component and the PLC.

Error information can be read in "LBP_ConfigData [DB7]":

- Op1KeyError and Op1KeyStatus (Op1Key)
- Op2KeyError and Op2KeyStatus (Op2Key)

Example (SCL)

Two MCPs are configured

- MCP 483C PN with bus address 192, IE mode (Ethernet), as MCP 1
- MCP 310C PN with bus address 193, IE mode (Ethernet), as MCP 2

One HT

- HT8 with bus address 14

Two direct key modules

- Direct keys integrated in the HT, bus address 14 (bus address of the HT)
- Direct keys connected to MCP 2, bus address 193 (bus address of the MCP 2)

20 bytes each are used for input data and output data, even if fewer bytes are needed for the devices. Global inputs and global outputs as of I 0.0 and Q 0.0 are used for this.

	MCP type	**Bus address**	**Length of the input data**	**In pointer**	**Length of the output data**	**Out pointer**
MCP 1	MCP 483C PN	192	14	P#E0.0	8	P#A0.0
MCP 2	MCP 310C PN	193	14	P#E20.0	8	P#A20.0
HT	HT8	14	8	P#E40.0	8	P#A40.0
OpKey 1	Direct keys	14	2	P#E60.0	0	P#A60.0
OpKey 2	Direct keys	193	2	P#E80.0	0	P#A80.0

Examples of the input and output images of the operator components used are listed below.

Table 10-102 MCP 483C PN: Input image

	Signals from the MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Spindle override				Operating mode			
	D	C	B	A	JOG	TEACH IN	MDI	AUTO
EB n + 1	Machine function							
	REPOS	REF	INCvar	INC10000	INC1000	INC100	INC10	INC1
EB n + 2	Key-operated switch position 0	Key-operated switch position 2	Spindle start	Spindle stop	Feedrate start	Feed stop	NC Start	NC stop
EB n + 3		Keyswitch		Feedrate override				
	Reset	Position 1	Single block	E	D	C	B	A
EB n + 4	Arrow keys			Keyswitch	Axis selection			
	+ (plus)	- (minus)	Rapid traverse	Position 3	X	4th axis	7th axis	R10
EB n + 5	Axis selection							
	Y	Z	5th axis	MCS/WCS	R11	9th axis	8th axis	6th axis
EB n + 6	Freely assignable customer keys							
	T9	T10	T11	T12	T13	T14	T15	
EB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
EB n + 8	-	-	-	-	-	-	-	-
EB n + 9	-	-	-	-	-	-	-	-
EB n + 10	KT-IN8	KT-IN7	KT-IN6	KT-IN5	KT-IN4	KT-IN3	KT-IN2	KT-IN1
EB n + 11	-	-	-	-	-	-	-	KT-IN9
EB n + 12	-	-	-	-	-	-	-	-
EB n + 13	-	-	-	X31				
	-	-	-	Pin 6	Pin 7	Pin 8	Pin 9	Pin 10

Table 10-103 MCP 483C PN: Output image

	Signals to the MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Machine function				Operating mode			
	INC 1000	INC 100	INC 10	INC 1	JOG	TEACH IN	MDI	AUTO
AB n + 1	-	-	-	-	Machine function			
	Feedrate start	Feed stop	NC Start	NC stop	REPOS	REF	INC Var	INC 10000
AB n + 2	- (minus)	Axis selection				Single block	Spindle start	Spindle stop
		X	4th axis	7th axis	R10			
AB n + 3	Axis selection							+ (plus)
	Z	5th axis	MCS/WCS	R11	9th axis	8th axis	6th axis	
AB n + 4	Freely assignable customer keys							
	T9	T10	T11	T12	T13	T14	T15	Y
AB n + 5	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
AB n + 6	-	-	-	-	-	-	Reset	R14
AB n + 7	-	-	KT-OUT6	KT-OUT5	KT-OUT4	KT-OUT3	KT-OUT2	KT-OUT1

Table 10-104 MCP 310C PN: Input image

	Signals from the slimline MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Spindle override				Operating mode			
	NC stop	SP -	SP 100%	SP +	Single block	JOG	MDI	AUTO
EB n + 1	Spindle				Keyswitch SS 3	Machine function		
	NC Start	SP right	SP stop	SP left		REF	REPOS	TEACH IN
EB n + 2	Feedrate			Keyswitch SS 0	Machine functions			
	Start	Stop	INC Var		INC1000	INC100	INC10	INC1
EB n + 3		Keyswitch		Feedrate override				
	Reset	SS 2	SS 1	E	D	C	B	A
EB n + 4	Arrow keys			Optional customer keys				
	+ (plus)	- (minus)	Rapid traverse	KT4	KT3	KT2	KT1	KT0
EB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
EB n + 6	Freely assignable customer keys					Freely assignable customer keys		
	T9	T10	T11	T12	MCS/WCS	T14	T15	T16
EB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8
EB n + 8	-	-	-	-	-	-	-	-
EB n + 9	-	-	-	-	-	-	-	-
EB n + 10	KT-IN8	KT-IN7	KT-IN6	KT-IN5	KT-IN4	KT-IN3	KT-IN2	KT-IN1
EB n + 11	-	-	-	-	-	-	-	KT-IN9

	Signals from the slimline MCP (keys) (MCP → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 12	-	-	-	-	-	-	-	-
EB n + 13	-	-	-	X31				
	-	-	-	Pin 6	Pin 7	Pin 8	Pin 9	Pin 10

Table 10-105 MCP 310C PN: Output image

	Signals to the slimline MCP (LED) (PLC → MCP)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Spindle override				Operating mode			
	NC stop	SP -	SP 100%	SP +	Single block	JOG	MDI	AUTO
AB n + 1	Spindle				-	Machine function		
	NC Start	SP right	SP stop	SP left		REF	REPOS	TEACH IN
AB n + 2	Feedrate		Machine functions					
	Start	Stop	INC Var		INC1000	INC100	INC10	INC1
AB n + 3	-	-	-	-	-	-	-	-
AB n + 4	Arrow keys			Optional customer keys				
	+ (plus)	- (minus)	Rapid traverse	KT4	KT3	KT2	KT1	KT0
AB n + 5			Axis selection					
	T17	KT5	6	5	4	Z	Y	X
AB n + 6	Freely assignable customer keys					Freely assignable customer keys		
	T9	T10	T11	T12	MCS/WCS	T14	T15	T16
AB n + 7	Freely assignable customer keys							
	T1	T2	T3	T4	T5	T6	T7	T8

Table 10-106 HT8: Input image

	Signals from the MCP simulation (keys) (HT 8 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	ACKN	Reset	WCS/MCS
EB n + 1	Function key block							
	CPF	U4	U3	BigFct	U2	U1	INC	REPOS
EB n + 2	-	-	Traversing keys (JOG) positive direction					
	-	Ax7-12	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
EB n + 3	-	-	Traversing keys (JOG) negative direction					
	-	-	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
EB n + 4	-							
	U9	U10	U11	U12	U13	U14	U15	U16
EB n + 5	-							
	-	U8	U7	U6	U5	SBL	-	-

	Signals from the MCP simulation (keys) (HT 8 → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 6	Start key block							
	-	HT 8	SF2	SF1	SF4	SF3	Start	Stop
EB n + 7	Feedrate override							
	-	-	-	E	D	C	B	A

Table 10-107 HT8: Output image

	Signals to the MCP simulation (LED) (PLC → HT 8)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AB n + 0	Function key block							
	REF	TEACH	AUTO	MDI	JOG	ACKN	Reset	WCS/MCS
AB n + 1	Function key block							
	-	U4	U3	-	U2	U1	INC	REPOS
AB n + 2	-	-	Traversing keys (JOG) positive direction					
	-	Ax7-12	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
AB n + 3	-	-	Traversing keys (JOG) negative direction					
	-	-	Ax6	Ax5	Ax4	Ax3	Ax2	Ax1
AB n + 4	-							
	U9	U10	U11	U12	U13	U14	U15	U16
AB n + 5	-	U8	U7	U6	U5	SBL	-	-
AB n + 6	Display traversing keys	Start key block						
		-	SF2	SF1	SF4	SF3	Start	Stop
AB n + 7	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-

Table 10-108 Direct keys: Input image

	Signals from direct keys (keys) (OpKey → PLC)							
Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EB n + 0	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1
EB n + 1	DT16	DT15	DT14	DT13	DT12	DT11	DT10	DT9

Initialization in the startup OB

The call of "LBP_ConfigBP [FC1]" is reduced to the parameters that are relevant for operator components.

```
"LBP_ConfigBP"(OpUnitCommId                  // Instance 1
 := 1,
   OpUnitInterfaceId :=                      // Interface X130 262
       "Local~CP~Ethernet-
   Schnittstelle_1"
```

```
MCPNum := 2,                            // Two MCPs
MCPBusType := B#16#55,                  // Ethernet
MCP1In := P#E0.0,                       // MCP 1
MCP1Out := P#A0.0,
MCP1BusAdr := 196,
MCP1Stop := FALSE,
MCP1NotSend := FALSE,
MCP1ConnectionId := 1001,
MCP1LocalUdpPort := 16001,
MCP2In := P#E20.0,                      // MCP 2
MCP2Out := P#A20.0,
MCP2BusAdr := 197,
MCP2Stop := FALSE,
MCP2NotSend := FALSE,
MCP2ConnectionId := 1002,
MCP2LocalUdpPort := 16002
HTIf := 5,                              // One HT8
HTIn := P#E40.0,
HTOut := P#A40.0,
HTAdr := 14,
HTStop := FALSE,
HTNotSend := FALSE,
HTConnectionId := 1003,
HTLocalUdpPort := 16003,
OpKeyNum := 2,                          // Two direct keys
Op1KeyIn := P#E60.0,                    // Direct keys 1
Op1KeyOut := P#A60.0,
Op1KeyBusAdr := 14,
Op1KeyStop := FALSE,
Op1KeyNotSend := FALSE,
Op1KeyConnectionId := 1004,
Op1KeyLocalUdpPort := 16004,
Op2KeyIn := P#E80.0,                    // Direct keys 2
Op2KeyOut := P#A80.0,
Op2KeyBusAdr := 15,
Op2KeyStop := FALSE,
Op2KeyNotSend := FALSE,
Op2KeyConnectionId := 1005,
Op2KeyLocalUdpPort := 16005,
IdentConnectionId := 1006,              // Ident
IdentLocalUdpPort := 16006);
```

Cyclical call (e.g. in "Main [OB1]")

The temporary variables tempError and tempStatus store the error information and can be evaluated later.

```
// #tempError: Bool;
// #tempStatus: Word;

OpUnitComm_DB(Config := LBP_ConfigData,
      Error => #tempError,
      Status => #tempStatus);
```

The following status values in "LBP_NC [DB10]" are updated:

"LBP_NC.E_MCP1ready", "LBP_NC.E_MCP2ready", "LBP_NC.E_HTready", "LBP_NC.E_Op1KeyReady" and "LBP_NC.E_Op2KeyReady"

// MCP 1: MCP 483C PN

```
LBP_MCPCtrlMilling(BAGNo := 1,                  // Mode group 1 and MCP 1
               ChanNo := 1,                     // Channel 1
               SpindleIFNo := 4,                // Axis 4 as spindle
               FeedHold =>
               LBP_Chan1.A_FDdisable,
               SpindleHold =>
               LBP_Axis4.A_FDSpStop);
```

As an example, the input and output data for MCP1 are processed by the call of "LBP_MCPCtrlMilling [FC19]".

For MCP 2 (MCP 310C PN), the block "LBP_MCPCtrlMillingSmall [FC24]" would be used. For the HT 8, the block "LBP_HTCtrlHT8 [FC26]" (not available on every SINUMERIK-PLC) would be used.

Connection identifiers and UDP ports

If, on the PLC, no different communication functions are used, then ideally you should use the suggested values. These values are preset on the "LBP_ConfigBP [FC1]" and in "LBP_ConfigData [DB7]" and can be imported without any adaptations.

- MCP1ConnectionId = 1001
- MCP2ConnectionId = 1002
- HTConnectionId = 1003
- Op1KeyConnectionId = 1004
- Op2KeyConnectionId = 1005
- IdentConnectionId = 1006

and

- MCP1LocalUdpPort = 16001
- MCP2LocalUdpPort = 16002

- HTConnectionId = 16003
- Op1KeyLocalUdpPort = 16004
- Op2KeyLocalUdpPort = 16005
- IdentLocalUdpPort = 16006

The connection identifiers may not be used a multiple number of times. If additional communication functions are used on the PLC, different connection identifiers must be selected. The connection identifiers must be unique. Also see the TCON online help.

The local UDP ports must not be used multiple times on the Ethernet interface that is used. If additional UDP communication functions in the PLC are used on the Ethernet interface configured for the operator components, other local UDP ports must be selected for this. The local UPD ports must be unique for the Ethernet interface.

Error and status

If an error occurs in the communication between PLC and operator component, the respective error bit is set and an error code is saved in the parameters listed in "LBP_ConfigData [DB7]".

- MCP1Error and MCP1Status (MCP1)
- MCP2Error and MCP2Status (MCP2)
- HTError and HTStatus (HT)
- Op1KeyError and Op1KeyStatus (Op1Key)
- Op2KeyError and Op2KeyStatus (Op2Key)
- IdentError and IdentStatus (Ident)

After an error occurs, check for a correct connection of the operator component with the PLC, the correct function of the DNS server, the correct parameterization of the hardware identification of the Ethernet port of the PLC, the local UDP port, the configured bus address of the operator component, and the correct parameterization of the pointers for the input and output data.

Table 10-109 Connection-specific error codes

Status	Description
16#0000	No error
16#8101	Error when opening or establishing the communication connection
16#8201	Error when sending the identify request to the operator component
16#8202	Timeout when receiving the identify response from the operator component
16#8203	The operator components does not have an IPv4 address
16#8301	The parameterization data for this operator component could not be determined
16#8302	The parameterization data is invalid
16#8303	Error when sending the parameterization request to the operator component
16#8304	The parameterization has been rejected by the operator component
16#8305	Timeout when receiving the parameterization response from the operator component
16#8401	Timeout when receiving the data
16#8402	The pointer for the input data from the operator component is invalid

16#8403	The pointer for the output data to the operator component is invalid
16#8404	Error when storing the input data from the operator component
16#8405	Error when copying the output data to the operator component
16#8501	Error when disconnecting or closing the connection

10.8.2 Identification of operator components

Via the identify interface in "LBP_ConfigData [DB7]", it is possible to query the type of the Ethernet component (MCP, HT 2, HT 8, HT 10 or direct keys) in cyclic operation.

The bus address or the TCU index is specified at the parameter IdentMcpBusAdr.

The input parameter IdentMcpProfilNo must normally be set to "0". This parameter must be set to the value "1" for the identification of the direct keys.

IdentConnectionId is an unambiguous connection identifier that is used by "LBP_OpUnitComm [FB25000]" for communication. The parameter at "LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]" is preset with the default value.

IdentMcpBusType specifies which communication connection is used (IE: B#16#5).

IdentLocalUdpPort is a unique UDP port that is used by "LBP_OpUnitComm [FB25000]" for communication. The parameter at "LBP_ConfigBP [FC1]" and "LBP_ConfigData [DB7]" is preset with the default value.

Note

It is not permissible to utilize the connection identifiers being used for other communication connections in the PLC project.

It is not permissible to utilize the local UDP ports being used for other UDP connections at this network interface in the PLC project.

The identification is started by setting the strobe signal IdentMcpStrobe.

After resetting the Strobe signal by the basic program, valid output information becomes available to the user. The identification can take several PLC cycles (up to 2 seconds).

Relevant parameters ("LBP_ConfigData [DB7]")	
General	
OpUnitCommId	
OpUnitInterfaceId (use the predefined constants for the CP interfaces, e.g. "Local~CP~Ethernet-Schnittstelle_120" for X120)	
Ident	
Input	**Output**
IdentMcpStrobe	IdentMcpType
IdentMcpBusAdr	IdentMcpLengthIn
IdentMcpProfilNo	IdentMcpLengthOut
IdentMcpBusType = B#16#5	
IdentConnectionId = 1006	
IdentLocalUdpPort = 16006	

Error information can be found in "LBP_ConfigData [DB7]" in the following variables:

- IdentError and IdentStatus

The output parameters should show the user the size of the data areas for the addressed device. Furthermore, it can be defined here whether an HT 2, HT 8 or an HT 10 or no device is connected to the terminal box. With this information, the channel for the operator component can be activated.

MCP type	IdentMcpType
no device connected	16#00
MCP 483C IE (Compact)	16#80
MCP 483C IE	16#81
MCP 310	16#82
MCP OEM	16#83
MCP DMG	16#84
HT 8	16#85
TCU DT (direct keys)	16#86
MCP MPP	16#87
HT 2	16#88
OP 08T (direct keys)	16#89
MCP 310C PN	16#8A
MCP index	16#8B
RCU/TCU30	16#8C
MCP Interface PN	16#8D
HT 10	16#91
MCP Modular	16#A0

Example (SCL)

The Ident interface is parameterized via the operator component parameters in "LBP_ConfigData [DB7]" and during the startup OB on "LBP_ConfigBP [FC1]".

```
LBP_ConfigData.OpUnitCommId := 1;         // Instance 1
LBP_ConfigData.OpUnitInterfaceId :=       // Interface X130
    "Local~CP~Ethernet-Schnittstelle_130"
LBP_ConfigData.IdentConnectionId := 1006;
LBP_ConfigData.IdentLocalUdpPort :=
16006;
```

One-time start of the identification.

```
LBP_ConfigData.IdentMcpBusAdr := 192;     // Device with bus address 192
LBP_ConfigData.IdentMcpProfilNo := 0;     // Use MCP profile
LBP_ConfigData.IdentMcpBusType := 5;      // Ethernet
LBP_ConfigData.IdentMcpStrobe := TRUE;    // Start identification
```

The block "LBP_OpUnitComm [FB25000]" must continue to be cyclically called, e.g. in "Main [OB1]". The instance DB "OpUnitComm_DB" is used, and the configuration block "LBP_ConfigData [DB7]" at the Config parameter.

```
OpUnitComm_DB(Config := LBP_ConfigData);
```

The identification is ended when IdentMcpStrobe has been reset.

```
IF LBP_ConfigData.IdentMcpStrobe = FALSE
THEN
    // Evaluate IdentMcpType, IdentMcpLengthIn, IdentMcpLengthOut
    // If applicable, check IdentError, IdentStatus
END_IF;
```

Now the type of the MCP and the lengths of the input and output data can be evaluated in "LBP_ConfigData [DB7]". In the event of an error during the identification, IdentError and IdentStatus must be checked.

10.8.3 Switchover of machine control panel, handheld unit

Switchover/deactivation of an operator component (MCP or HT or direct keys) is only supported with Ethernet coupling.

Control signals

Communication to the individual components can be stopped using parameters MCP1Stop, MCP2Stop, HTStop, Op1KeyStop and Op2KeyStop. Stopping and subsequent reactivation of communication is possible in the current cycle.

Since "LBP_ConfigBP [FC1]" can only be called once, all of the changes in the cyclic mode must be made directly in the configuration block "LBP_ConfigData [DB7]".

Example: Stopping transfer from the 1st machine control panel:

```
LBP_ConfigData.MCP1Stop := TRUE;
```

Example: Starting transfer of the 1st machine control panel:

```
LBP_ConfigData.MCP1Stop := FALSE;
```

When the parameters MCP1Stop, MCP2Stop HTStop, Op1KeyStop and Op2KeyStop are set, the bus address, the start addresses of the input and output data, and the bus type (only MCP) can be changed. In addition, alarms are deleted or suppressed for the respective component.

Switchover of the bus address

If an existing communication connection to an operator component is to be cancelled and a new communication connection established to a different operator component with a different bus address - or the pointers for the input and output data are changed - then proceed as follows:

1. Stopping the communication of the operator component to be disconnected:
   Parameter MCP1Stop, MCP2Stop, HTStop, Op1KeyStop or Op2KeyStop := TRUE;
2. The communication is stopped when the following applies in "LBP_NC [DB10]":
   - MCP: E_MCP1ready or E_MCP2_Ready = FALSE
   - HT: E_HTready = FALSE
   - OpKey: E_Op1KeyReady or E_Op2KeyReady = FALSE
3. Change of the bus address in "LBP_ConfigData [DB7]":
   - MCP: MCP1BusAdr or MCP2BusAdr := <Bus address of the new MCP>
   - HT: HTAdr := <Bus address of the new HT>
   - OpKey: Op1KeyBusAdr or Op2KeyBusAdr := <Bus address of the new OpKey>
4. Enable the communication (possible in the same PLC cycle as point 3):
   Parameter MCP1Stop, MCP2Stop, HTStop, Op1KeyStop or Op2KeyStop := FALSE;
5. The communication with the new control component is active if the following applies in "LBP_NC [DB10]":
   - MCP: E_MCP1ready or E_MCP2_Ready = TRUE
   - HT: E_HTready = TRUE
   - OpKey: E_Op1KeyReady or E_Op2KeyReady = TRUE

Connection identifiers and the local UDP ports can be changed using the same procedure.

Example (SCL)

Change the configuration for MCP 1, to do this, stop MCP 1.

```
LBP_ConfigData.MCP1Stop := TRUE;          // Stop communication
```

Wait at least one cycle, set new parameter and start MCP 1.

```
LBP_ConfigData.MCP1BusAdr := 192;         // Device with bus address 192
LBP_ConfigData.MCP1Stop := FALSE;         // Start communication
```

Wait and check as to whether the connection is re-established.

```
IF LBP_NC.E_MCP1ready THEN
    // Connection is re-established
END_IF;
```

10.9 Appendix

10.9.1 References

An exhaustive description of the system data can be found in the following references:

- SINAMICS S210 servo drive system

Appendix

A

A.1 List of abbreviations

A	
A	Output
ASCII	American Standard Code for Information Interchange: American coding standard for the exchange of information
ASIC	Application Specific Integrated Circuit: User switching circuit
ASUP	Asynchronous subprogram
AUTO	Operating mode "Automatic"
AUXFU	Auxiliary Function: Auxiliary function
AWL	Statement List

B	
BAG	Mode group
BCD	Binary Coded Decimals: Decimal numbers encoded in binary code
BICO	Binector Connector
BIN	Binary Files: Binary files
BKS	Basic Coordinate System
BM	Operating alarm
BO	Binector Output
BTSS	Operator Panel Interface

C	
CLC	Clearance control
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
COM	Communication
CP	Communication Processor
CPU	Central Processing Unit: Central processing unit
CST	Configured Stop: Configured stop

D	
DB	Data Block (PLC)
DBB	Data Block Byte (PLC)
DBD	Data Block Double word (PLC)
DBW	Data Block Word (PLC)
DBX	Data block bit (PLC)
DDS	Drive Data Set: Drive data set

D	
DIR	Directory: Directory
DO	Drive Object
DRF	Differential Resolver Function: Differential revolver function (handwheel)
DRY	Dry Run: Dry run feedrate
DW	Data Word
DWORD	Double Word (currently 32 bits)

E	
E	Input
EES	Execution from External Storage
E/A	Input/Output
ESR	Extended Stop and Retract
ETC	ETC key ">"; Softkey bar extension in the same menu

F	
FB	Function Block (PLC)
FC	Function Call: Function Block (PLC)
FDD	Feed Disable: Feed disable
FdStop	Feed Stop: Feed stop
FIFO	First In First Out: Memory that works without address specification and whose data is read in the same order in which they was stored
FM	Error message
FUP	Function Block Diagram (PLC programming method)
FW	Firmware

G	
GEO	Geometry, e.g. geometry axis
GP	Basic program (PLC)
GUD	Global User Data: Global user data

H	
HEX	Abbreviation for hexadecimal number
HiFu	Auxiliary function
HMI	Human Machine Interface: SINUMERIK user interface
HSA	Main Spindle Drive
HT	Handheld Terminal
HW	Hardware

I	
IBN	Commissioning
INC	Increment: Increment
INI	Initializing Data: Initializing data
IPO	Interpolator

J	
JOG	Jogging: Setup mode

K	
KOP	Ladder Diagram (PLC programming method)

L	
LED	Light Emitting Diode: Light-emitting diode
LMS	Position Measuring System
LR	Position controller

M	
Main	Main program: Main program (OB1, PLC)
MCP	Machine Control Panel: Machine control panel
MD	Machine Data
MDA	Manual Data Automatic: Manual input
MDS	Motor Data Set: Motor data set
MELDW	Message Word
MKS	Machine Coordinate System
MM	Motor Module
MPF	Main Program File: Main program (NC)
MPI	Multi Point Interface
MSTT	Machine control panel

N	
NC	Numerical Control: Numerical control with block preparation, traversing range, etc.
NCU	Numerical Control Unit: NC hardware unit
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	Interface Signal
NV	Work offset
NX	Numerical Extension: Axis expansion board

O	
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Operating equipment

P	
PCU	PC Unit: PC box (computer unit)
PG	Programming device
PLC	Programmable Logic Control: Adaptation control
PN	PROFINET
PO	Power On
POS	Position/positioning
PPO	Parameter Process data Object: Cyclic data telegram for PROFIBUS DP transmission and "Variable speed drives" profile
PPU	Panel Processing Unit (central hardware for a panel-based CNC, e.g. SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serial data bus
PRT	Program Test
PTP	Point to Point: Point-To-Point
PZD	Process data: Process data part of a PPO

R	
REF	REFerence point approach function
REPOS	REPOSition function
RESU	Retrace support
RID	Read In Disable
RP	R Parameter, arithmetic parameter, predefined user variable

S	
SA	Synchronized Action
SBL	Single Block: Single block
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum or setting data
SDI	Safe Direction
SERUPRO	Search-Run by Program Test: Block search, program test
SIC	Safety Info Channel
SKP	Skip: Function for skipping a part program block
SLP	Safe Limited Position
SLS	Safely Limited Speed
SMI	Sensor Module Integrated

S	
SOS	Safe Operating Stop
SPF	Sub Program File: Subprogram (NC)
SS1	Safe Stop 1
SS2	Safe Stop 2
STO	Safe Torque Off
STW	Control word
SUG	Grinding Wheel Peripheral Speed
SW	Software

T	
TCU	Thin Client Unit
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TOFF	Online tool length offset
TRANSMIT	Transform Milling Into Turning: Coordination transformation for milling operations on a lathe

U	
UP	Subprogram
USB	Universal Serial Bus

V	
VDI	Internal communication interface between NC and PLC

W	
WKS	Workpiece Coordinate System
WPD	Work Piece Directory: Workpiece directory
WZ	Tool
WZV	Tool Management

Z	
ZSW	Status word (of drive)

A.2 Documentation overview

Comprehensive documentation about the functions provided in SINUMERIK ONE Version 6.13 and higher is provided in the Documentation overview SINUMERIK ONE (https://support.industry.siemens.com/cs/ww/en/view/109768483).



You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Configuration/Setup
- Manufacturer/Service: Safety Integrated
- Information and training
- Manufacturer/Service: SINAMICS

Index

A

Activation
from machine control panel, handheld unit, 755

B

Block
Conflicts during copying, 32
List of all basic program blocks, 87, 89, 91

D

Data type
LBP_typeHandwheel, 95
LBP_typeMcp*, 95
Data type LBP_typeHandwheel
handWheel1, 95
handWheel2, 95
DB19, signals from the operator panel, 214

H

Handheld unit
HT 10, 592, 596
HT 2, 591, 595
HT 8, 591, 596
Hardware catalog
NCU 1750, 19
NCU 1760, 19
NX10.3, 19
NX15.3, 19

I

Inverse signals, 94

L

LBP_AlarmMsgs [DB2]
AddValUserA0, 113
Axis, 112
Axis ranges, 107
AxisA, 113
Chan, 112
ChanA, 113
Channel range 1, 98
Channel range 10, 106
Channel range 2, 100
Channel range 3, 101
Channel range 4, 101
Channel range 5, 102
Channel range 6, 103
Channel range 7, 104
Channel range 8, 104
Channel range 9, 105
User ranges, 108
UserA, 113
LBP_Axis1 [DB31], ...
A_AckFixedStop, 393
A_ActGearA, 423
A_ActGearB, 423
A_ActGearC, 423
A_ASpDisable, 394
A_AutoGearStep, 447
A_AxReset, 441
A_AxResume, 442
A_ChangeRP, 443
A_ChangSetpOutAss0, 437
A_ChangSetpOutAss1, 437
A_Clamp, 406
A_CompContr, 439
A_ContManTravel, 415, 418, 419
A_ContrEnable, 402
A_ControlAx, 437
A_Corr, 440
A_DelayRef, 420
A_DelDTGSpReset, 405
A_DeleteS, 426
A_DelSyncRunCorr, 448
A_DEPBCS, 440
A_DEPMCS, 439
A_Disable, 413
A_DisSync, 447
A_DriveEnable, 392
A_DynBComp, 439
A_EnabTravFixedStop, 408
A_ExtZO, 407
A_FD_OR, 390
A_FD_ORSp, 426
A_FDSpStop, 412
A_FeedDriveCAxEng, 448
A_FixedFD1, 409
A_FixedFD2, 409

A_FixedFD3, 409
A_FixedFD4, 409
A_Follow_upMode, 397
A_FollowAxOv, 439
A_GearChangeOv, 425
A_HIAxMov, 440
A_HoldBrakeToOpen, 434
A_HW1, 410
A_HW2, 410
A_HW3, 410
A_HWLimitMinus, 419
A_HWLimitPlus, 419
A_INC1, 415
A_INC10, 415
A_INC100, 415
A_INC1000, 415
A_INC10000, 415
A_INCVar, 415
A_IntegratDisable, 436
A_InvHWDirOfRot, 416
A_JogFixPPos0, 421
A_JogFixPPos1, 421
A_JogFixPPos2, 421
A_JogToPos, 422
A_LockParSetDef, 418
A_M3M4Inv, 427
A_Minus, 413
A_ModuloLimitEn, 420
A_MotA, 434
A_MotB, 434
A_MotOK, 435
A_MS, 438
A_NCASpChanA, 416
A_NCASpChanB, 416
A_NCASpChanC, 416
A_NCASpChanD, 416
A_NCASpStrobe, 416
A_NoAutoSync, 446
A_NoMonitorGear, 426
A_OEMAxis, 416
A_ORactive, 401
A_OscillAxExtRev, 440
A_OscilPLC, 427
A_OscilSpeed, 428
A_ParA, 434
A_Para_A, 417
A_Para_B, 417
A_Para_C, 417
A_ParB, 434
A_ParC, 434
A_PLCASp, 416
A_PLCAxis, 446
A_Plus, 413
A_PosMeas1, 397
A_PosMeas2, 397
A_PosSpindle, 447
A_PrgtestAxRel, 410
A_ProgtestActivate, 423
A_ProgtestSuppress, 422
A_PulseEnable, 436
A_RapidTrOR, 413
A_RefVal1, 406
A_RefVal2, 406
A_RefVal3, 406
A_RefVal4, 406
A_Resynchronize, 447
A_RotationMonit, 437
A_RUEncQuStop, 433
A_SensorFixedStop, 393
A_SetRotDirectLeft, 429
A_SetRotDirectRight, 429
A_SetRP, 443
A_SI_SetpLimBit0, 448
A_SI_SetpLimBit1, 448
A_Sp1Syn, 425
A_Sp1SynP, 426
A_Sp2Syn, 425
A_Sp2SynP, 426
A_SpeedSetpSmooth, 434
A_SpInsideClamp, 448
A_SpOR, 431
A_SpSpeedDisp, 448
A_StartCCW, 447
A_StartCW, 446
A_StartGantSynRun, 446
A_Stop, 445
A_StopCorr, 440
A_StopDEPBCS, 440
A_StopDEPMCS, 440
A_StopHIAxMove, 440
A_StopRP, 444
A_StopSpindle, 446
A_SWCam, 401
A_SWLimit2Minus, 419
A_SWLimit2Plus, 419
A_TorqComp, 437
A_Tracksync, 448
A_VeloSpeedLimit, 410
CW7.A_DisableCurrReduction, 448
CW7.A_DisablePowerReduction, 448
E_ActRotRight, 481
E_ActTravFStop, 459
E_ActValCoupling, 497
E_AIAxis, 505

E_AnalogSensor, 509
E_AndConditionFulfilled, 510
E_AWL, 499
E_AxContainerRot, 460
E_AxialAlarm, 454
E_AxisAccel, 500
E_AxisCtrl, 494
E_AxisReady, 455
E_AxReset, 460
E_AxSpDisable, 462
E_AxStop, 461
E_BitValCtrlOutChang0, 495
E_BitValCtrlOutChang1, 496
E_BrakeTest, 470
E_ChPoss, 466
E_ClampingState, 508
E_ClampTolerance, 504
E_CLG, 482
E_CollCheckRedSpeed, 476
E_ConstCuttSpeed, 482
E_ContrMode, 484
E_CurrentContr, 457
E_CurrReductionActive, 510
E_CurvePos, 476
E_DriveRunEn, 487
E_DynBComp, 503
E_EmRetr, 499
E_EncMaintenanceRequired, 510
E_EncodeFreq1, 450
E_EncodeFreq2, 451
E_ErrorOscill, 502
E_ExactCoarse, 452
E_ExactFine, 453
E_FixedStop, 460
E_Follow_upMode, 455
E_ForceFixedStop, 460
E_FPosAxis, 476
E_GantAxis, 503
E_GantCOLimit, 503
E_GantGroupSyn, 503
E_GantLeadAxis, 503
E_GantWarn, 503
E_GearChange, 477
E_GeoMonit, 480
E_HoldBrakeOpened, 487
E_HTempWarn, 490
E_HW1, 462
E_HW2, 462
E_HW3, 462
E_HWOverlay, 458
E_I2TLimit, 494
E_INC1, 465
E_IndexAxisPos, 475
E_IntegratDisable, 489
E_InvHWDirOfRot, 466
E_JogFixPPos0, 472
E_JogFixPPos0Act, 472
E_JogFixPPos1, 472
E_JogFixPPos1Act, 472
E_JogFixPPos2, 472
E_JogFixPPos2Act, 472
E_JogPos, 474
E_JogToPos, 473
E_MAR, 501
E_MasterSp, 499
E_MdMdx, 491
E_MeasAct, 459
E_Message, 493
E_MFunct, 485
E_MMC_ProgtestActivate, 506
E_MMC_ProgtestSuppress, 505
E_MMC_REPOSDelay, 471
E_ModLimEnAct, 472
E_MotA, 488
E_MotB, 488
E_MotOK, 488
E_MotorTempModelAlarm, 510
E_MS, 496
E_MSCoarse, 495
E_MSCompContr, 495
E_MSFine, 494
E_MSR, 501
E_MTempWarn, 490
E_NactNmin, 492
E_NactNset, 493
E_NactNx, 492
E_NCASpChanA, 466
E_NCASpChanB, 466
E_NCASpChanC, 466
E_NCASpChanD, 466
E_NCU_Link, 450
E_NCUNumLink3, 468
E_NeutrASp, 466
E_OEMAxis, 465
E_OrConditionFulfilled, 511
E_Oscill, 502
E_OscillAxExtRev, 501
E_OscillMode, 484
E_OscillMotion, 502
E_OverlayMotion, 498
E_ParA, 488
E_ParB, 488
E_ParC, 488
E_ParS_A, 467

E_ParS_B, 467
E_ParS_C, 467
E_PathAxis, 474
E_PLCASp, 466
E_PLCAxDedic, 471
E_PLCCtrlAx, 461
E_PLCType, 466
E_PolePosIdentPassed, 510
E_POS_RESTORED1, 470
E_POS_RESTORED2, 470
E_PosAxis, 475
E_PositContr, 456
E_PosMeas1, 504
E_PosMeas2, 504
E_PosMode, 483
E_PulseEnable, 489
E_RefSyn1, 452
E_RefSyn2, 452
E_ReposDelayQuit, 469
E_ReposSh, 468
E_ReposShValid, 469
E_RevFD, 458
E_RLIact, 487
E_RUComplete, 491
E_RUEncDisabl, 486
E_S1aMeasARod, 507
E_S4dPistonStopP, 507
E_S4PistonStopP, 509
E_S5AngleShaft, 509
E_S5dAngleShaft, 507
E_S6Temp, 508
E_ScratchPulse, 474
E_SensorAv, 506
E_SetpGearA, 476
E_SetpGearB, 476
E_SetpGearC, 476
E_SetRange, 480
E_SetSpeedIncreased, 479
E_SFunct, 486
E_SI_Drv_SICSCC, 470
E_SlaveSp, 500
E_Sp_NA, 449
E_SparkOut, 502
E_SpeedContr, 457
E_SpeedLimit, 478
E_SpeedMonit, 481
E_SpeedSetpSmooth, 487
E_SpindleMonit, 494
E_SpInPosition, 485
E_Stat, 456
E_StopCorr, 462
E_StopDEPBCS, 462
E_StopDEPMCS, 462
E_StopHIAxMove, 462
E_StopOscill, 501
E_SUG, 482
E_SupportAreaViol, 480
E_SWCam, 457
E_SWL, 498
E_Sync, 501
E_SyncMode, 483
E_Syncron2Coarse, 504
E_Syncron2Fine, 504
E_SyncronCoarse, 497
E_SyncronFine, 496
E_SyncRunCorrDOut, 500
E_SyncRunCorrIncl, 504
E_SynRunStart, 503
E_Tapping, 482
E_TCMinus, 464
E_TCPlus, 464
E_ToolDynLimit, 484
E_TorqueLimited, 510
E_TravRequ, 454
E_TReqMinus, 463
E_TReqPlus, 463
E_UDClessAlThreshold, 493
SCC, 510
SIC, 510
Signals from axis/spindle, 384
Signals to the axis/spindle, 381
LBP_CallBackAuxFunc [FC12], 696
LBP_Chan1 [DB21], ...
A_AxisSelTypeActive, 350
A_AxisSetAxisActive, 350
A_CfgStopFunc, 280
A_CLC_OR, 263
A_CLC_Stop, 262
A_ContHW1, 290
A_ContHW2, 290
A_ContHW3, 290
A_ContHWneg, 291
A_ContHWsim, 291
A_CSP, 281
A_Cycle_Sig, 344
A_D1, 333
A_D2, 333
A_D3, 333
A_D4, 333
A_D5, 333
A_D6, 333
A_D7, 333
A_D8, 333
A_D9, 333

A_DeleteDTG, 274
A_DisablWearMon, 289
A_DisablWP_Counter, 288
A_DontDisablTool, 290
A_DRF, 259
A_DRY, 260
A_F_Limit, 275
A_FD_OR, 268
A_FD_ORA, 276
A_FDdisable, 273
A_FixedFD1, 287
A_FixedFD2, 287
A_FixedFD3, 287
A_FixedFD4, 287
A_InKeyG1, 347
A_InKeyG2, 347
A_InKeyG3, 347
A_InKeyG4, 347
A_InKeyG5, 347
A_InKeyG6, 347
A_InKeyG7, 347
A_InKeyG8, 347
A_InKeyGEn1, 347
A_InKeyGEn2, 347
A_InKeyGEn3, 347
A_InKeyGEn4, 347
A_InKeyGEn5, 347
A_InKeyGEn6, 347
A_InKeyGEn7, 347
A_InKeyGEn8, 347
A_InKeyGRunIn1, 347
A_InKeyGRunIn2, 347
A_InKeyGRunIn3, 347
A_InKeyGRunIn4, 347
A_InKeyGRunIn5, 347
A_InKeyGRunIn6, 347
A_InKeyGRunIn7, 347
A_InKeyGRunIn8, 348
A_InvCHWDirOfRot, 295
A_JOG_Circles, 292
A_M01, 260
A_ManRelStroke2, 267
A_ManStrokEnab, 266
A_MRP, 281
A_NCKrelatedM01, 292
A_NCStart, 277
A_NCStartDisabl, 277
A_NCStop, 278
A_NCStopASp, 279
A_NCStopBlock, 278
A_NoToolChangeCmd, 293
A_OEM, 287
A_OEM_TechnoSig, 344
A_Ori, 340
A_PLC_ActCompl, 263
A_ProgAbort, 275
A_ProgJump, 346
A_ProgTest, 264
A_ProtZone, 261
A_PTP_Travel, 288
A_Ref, 261
A_REPOSMode, 294
A_REPOSPM_0, 293
A_REPOSPM_1, 293
A_REPOSPM_2, 293
A_Reset, 280
A_RIdisable, 274
A_RT_OR, 271
A_RT_ORA, 276
A_SBL, 259
A_SKP0, 265
A_SKP1, 265
A_SKP2, 265
A_SKP3, 265
A_SKP4, 265
A_SKP5, 265
A_SKP6, 265
A_SKP7, 265
A_SKP8, 295
A_SKP9, 295
A_SP_Clear, 275
A_Str_Synfrom, 333
A_Str_Synon, 333
A_StrokDelayed, 266
A_StrokEnab, 265
A_StrokStop, 267
A_StrokSup, 266
A_SuppStartLock, 280
A_SynDisabl, 333
A_SynOff, 261
A_TM_TimOut, 262
D1, 324
D1Change, 319
D1Quick, 321
D2, 324
D2Change, 319
D2Quick, 321
D3, 325
D3Change, 319
D3Quick, 321
E_AcknManStrokEnab, 317
E_ActBlock, 297
E_AnyAsup, 346
E_AssM01, 337

E_ASUP_Stop, 335
E_AxesRef, 309
E_AxesStop, 310
E_AxisSelAxis, 350
E_AxisSelRequest, 350
E_AxisSelType, 349
E_BegBlock, 297
E_BlockSearch, 299
E_CART_JOG_MODE, 349
E_CfgStopAtBlockEnd, 318
E_CfgStopFuncActive, 318
E_ChanActive, 307
E_ChanInterrupt, 308
E_ChanReset, 309
E_ChanRO, 311
E_CHW1, 311
E_CHW2, 311
E_CHW3, 311
E_CLC, 312
E_CLCStopLL, 313
E_CLCStopUL, 314
E_CollCheckStop, 345
E_CSPactivated, 331
E_CSPViolated, 332
E_Cycle_Sig, 344
E_DelayFTS, 339
E_DriveTest, 334
E_DRY, 337
E_ExecExtern, 296
E_ExtLang, 334
E_G00, 334
E_HWOverlay, 299
E_InKeyGIsEn1, 348
E_InKeyGIsEn2, 348
E_InKeyGIsEn3, 348
E_InKeyGIsEn4, 348
E_InKeyGIsEn5, 348
E_InKeyGIsEn6, 348
E_InKeyGIsEn7, 348
E_InKeyGIsEn8, 348
E_InKeyGRunOut1, 348
E_InKeyGRunOut2, 348
E_InKeyGRunOut3, 348
E_InKeyGRunOut4, 348
E_InKeyGRunOut5, 348
E_InKeyGRunOut6, 349
E_InKeyGRunOut7, 349
E_InKeyGRunOut8, 349
E_InvCHWDirOfRot, 318
E_IR_Activ, 310
E_JOG_Circles, 345
E_JogRetract, 345
E_LastActBlock, 298
E_M01, 297
E_M30, 300
E_MMC_CfgStopFunc, 283
E_MMC_DRF, 283
E_MMC_DRY, 285
E_MMC_FD_OR4RT_OR, 286
E_MMC_M01, 284
E_MMC_M01AssocNC, 284
E_MMC_ProgTest, 287
E_MMC_REPOSMode, 286
E_MMC_REPOSPM_0, 285
E_MMC_REPOSPM_1, 285
E_MMC_REPOSPM_2, 285
E_MMC_SKP0, 287
E_MMC_SKP1, 287
E_MMC_SKP2, 287
E_MMC_SKP3, 287
E_MMC_SKP4, 287
E_MMC_SKP5, 287
E_MMC_SKP6, 287
E_MMC_SKP7, 287
E_MMC_SKP8, 287
E_MMC_SKP9, 287
E_MRPactivated, 330
E_MRPViolated, 331
E_NCKalarmChan, 311
E_NCKalarmStop, 311
E_NoToolChangeCmd, 340
E_OEM_TechnoSig, 344
E_OEMChan, 302
E_Ori, 341
E_OrieToolholder, 298
E_Overstore, 337
E_ProgAborted, 306
E_ProgEvent_M30, 344
E_ProgEvent_PowerOn, 344
E_ProgEvent_Reset, 344
E_ProgEvent_SearchRun, 344
E_ProgEvent_Start, 344
E_ProgInterrupt, 305
E_ProgRunning, 302
E_ProgStop, 304
E_ProgTest, 301
E_ProgWait, 303
E_ProtZoneNOK, 317
E_PTP_Travel, 334
E_Ref, 298
E_REPOS_DEFERRA, 339
E_REPOS_EdgeAckn, 337
E_REPOS_PMode0, 338
E_REPOS_PMode1, 338

E_REPOS_PMode2, 338
E_RetractData, 345
E_RevFD, 299
E_RIEnablgnored, 315
E_SblStopAtBlockEnd, 318
E_SearchAct, 335
E_SilentAsup, 346
E_StartReq, 309
E_Stop_NoDelayR, 340
E_StopBlkEndSBLsuppr, 316
E_StopCondition, 345
E_StopDelayed, 336
E_StopReq, 309
E_StrokeEA, 316
E_SynDisabled, 333
E_TLastRepl, 342
E_TLimit, 341
E_TNewRepl, 342
E_TNoLastRepl, 343
E_TNoLimit, 343
E_TNoNewRepl, 343
E_TNoPrewarnLimit, 343
E_TOFF, 336
E_TOFFmovem, 336
E_ToolMissing, 335
E_TPrewarnLimit, 341
E_Transform, 301
E_TransformNo, 348
E_WS_Setp, 334
EChange, 319
EQuick, 321
ExtD1, 324
ExtD2, 324
ExtD3, 324
ExtE, 325
ExtF1, 326
ExtF2, 326
ExtF3, 326
ExtF4, 326
ExtF5, 327
ExtF6, 327
ExtH1, 326
ExtH2, 326
ExtH3, 326
ExtM1, 322
ExtM2, 322
ExtM3, 322
ExtM4, 322
ExtM5, 322
ExtS1, 322
ExtS2, 322
ExtS3, 323
ExtT1, 323
ExtT2, 323
ExtT3, 323
F1, 326
F1Change, 320
F1Quick, 321
F2, 326
F2Change, 320
F2Quick, 321
F3, 326
F3Change, 320
F3Quick, 321
F4, 326
F4Change, 320
F4Quick, 321
F5, 327
F5Change, 320
F5Quick, 321
F6, 327
F6Change, 320
F6Quick, 321
GAct, 329
H1, 326
H1Change, 320
H1Quick, 321
H2, 326
H2Change, 320
H2Quick, 321
H3, 326
H3Change, 320
H3Quick, 321
I, 325
M1, 322
M1Change, 319
M1NDec, 320
M1Quick, 321
M2, 322
M2Change, 319
M2NDec, 320
M2Quick, 321
M3, 322
M3Change, 319
M3NDec, 320
M3Quick, 321
M4, 322
M4Change, 319
M4NDec, 320
M4Quick, 321
M5, 322
M5Change, 319
M5NDec, 320
M5Quick, 321

MDyn, 328
S1, 322
S1Change, 319
S1Quick, 321
S2, 322
S2Change, 319
S2Quick, 321
S3, 323
S3Change, 319
S3Quick, 321
T1, 323
T1Change, 319
T1Quick, 321
T2, 323
T2Change, 319
T2Quick, 321
T3, 323
T3Change, 319
T3Quick, 321
LBP_Chan1 [DB21], ...
A_Geo, 282
E_Geo, 319
LBP_Chan1 [DB21], ...
Change signals from the channel, 246, 247
Control signals from geometry axes, 245, 246
Control signals from the operating software, 244, 245
Control signals from the PLC, 244, 245
Control signals to channel (1), 241, 242
Control signals to the geometry axes, 242, 243
D function signals, 248
DL function signals, 248
F function signals, 249
G function signals, 250, 251
H function signals, 249
Job-controlled signals, 252, 253
M function signals, 247, 248
M function signals, dynamic, 249, 250
Protection area signals from the channel, 251
S function signals, 247, 248
Signals for the tool management functions, 256
Signals from channel, 256, 257
Signals from the orientation axes, 255
Signals from/to the channel, 257, 258
Signals to channel, 257
Signals to the orientation axes, 254
Status signal from the channel, 244, 245
T function signals, 248
LBP_ConfigBP [FC1] (basic program, startup section), 663
LBP_ConfigData [DB7]
ActiveAxis, 127
ActiveChan, 127
ExtendChanAxMsg, 125
GenerateAlarmMsgs, 128
HTAdr, 123
HTConnectionId, 129
HTError, 130
HTIf, 123
HTIn, 123
HTLocalUdpPort, 130
HTNotSend, 124
HTOut, 123
HTReady, 131
HTStatus, 130
HTStop, 123
HWheelMMC, 124
IdentConnectionId, 129
IdentError, 131
IdentLocalUdpPort, 130
IdentMcpBusAdr, 126
IdentMcpBusType, 127
IdentMcpLengthIn, 128
IdentMcpLengthOut, 128
IdentMcpProfilNo, 126
IdentMcpStrobe, 127
IdentMcpType, 128
IdentStatus, 131
IRAuxfuE, 125
IRAuxfuH, 125
IRAuxfuT, 125
LBPVersionInfo, 128
ListMDecGrp, 124
MaxAxis, 127
MaxChan, 127
MaxModeGroup, 127
MaxNumUserDataHex, 127
MaxNumUserDataInt, 127
MaxNumUserDataReal, 128
MCP_IF_TCS, 124
MCP1BusAdr, 122
MCP1ConnectionId, 129
MCP1Error, 130
MCP1In, 122
MCP1LocalUdpPort, 129
MCP1NotSend, 122
MCP1Out, 122
MCP1Ready, 131
MCP1Status, 130
MCP1Stop, 122
MCP2BusAdr, 122
MCP2ConnectionId, 129
MCP2Error, 130
MCP2In, 122

MCP2LocalUdpPort, 129
MCP2NotSend, 122
MCP2Out, 122
MCP2Ready, 131
MCP2Status, 130
MCP2Stop, 122
MCPBusType, 123
MCPNum, 122
MMCToIF, 124
MsgUser, 125
NCCyclTimeout, 124
NCRunupTimeout, 124
Op1KeyBusAdr, 126
Op1KeyConnectionId, 129
Op1KeyError, 131
Op1KeyIn, 125
Op1KeyLocalUdpPort, 130
Op1KeyNotSend, 126
Op1KeyOut, 126
Op1KeyReady, 131
Op1KeyStatus, 131
Op1KeyStop, 126
Op2KeyBusAdr, 126
Op2KeyConnectionId, 129
Op2KeyError, 131
Op2KeyIn, 125
Op2KeyLocalUdpPort, 130
Op2KeyNotSend, 126
Op2KeyOut, 126
Op2KeyReady, 132
Op2KeyStatus, 131
Op2KeyStop, 126
OpKeyNum, 125
OpUnitCommId, 128
OpUnitInterfaceId, 128
UserVersion, 125
UserVersionInfo, 128
LBP_CtrlAxisSpindle [FC18], 702
LBP_CtrlEnergy [DB1000]
Energy-saving profiles, 554
Logs, 556
LBP_GenerateAlarmMsgs [FC10], 690
LBP_Handwheels [FC25002], 751
LBP_HMI [DB19]
A_Active, 240
A_ActiveEnable, 239
A_ActWCS, 217
A_ASpi1percent, 218
A_ASpi2percent, 218
A_AToolMeas, 219
A_Channel, 239
A_ClearCanAl, 217
A_ClearRecAl, 217
A_DeleteTool, 239
A_Done, 240
A_Duplo, 238
A_Error, 240
A_Error_Code, 240
A_FileIndex, 239
A_FuncNo, 231
A_Hardkey, 219
A_HWeelSimOvr, 240
A_Ident, 238
A_K_Code, 239
A_KeyDisable, 216
A_LoadingPlace, 239
A_LoadTool, 239
A_Magazine, 239
A_MagazinePlace, 238
A_MagazinePlaceType, 238
A_MCPChan, 218
A_NCU_Index, 239
A_OEM1, 219
A_OEM2, 219
A_Override, 239
A_Par1, 232
A_Par2, 232
A_Par3, 232
A_PictNo, 230
A_PictOff, 230
A_PictOn, 230
A_PLCExtViewerMode, 218
A_PLCExtViewerReject, 218
A_PPdeload, 219
A_PPload, 220
A_PPsel, 220
A_ReqMMC1, 239
A_ReqMMC2, 239
A_SCDark, 216
A_SCLight, 215
A_SimOvr, 240
A_SimState, 241
A_Subtype, 238
A_TeachDisable, 219
A_Tnumber, 238
A_ToolSize_Down, 238
A_ToolSize_Left, 238
A_ToolSize_Right, 238
A_ToolSize_Upper, 238
A_TPMCounter, 240
A_TPMStatus, 240
A_Unit, 239
A_V24Index1, 220
A_V24Index2, 220

A_V24NoFile1, 221
A_V24NoFile2, 222
E_Active, 239
E_ActivWA, 224
E_ActWCS, 224
E_Cancel, 223
E_CanCleared, 223
E_Chan, 224
E_DataTransf, 225
E_DisplChange, 225
E_Done, 239
E_Duplo, 239
E_Error, 239
E_Error_Code, 240
E_FuncErr, 232
E_Ident, 239
E_Ident_Ready, 239
E_K_CodeError, 240
E_KeybHandl, 225
E_Magazine, 239
E_MagazinePlace, 239
E_MagazinePlaceType, 239
E_NotAct, 231
E_PictAct, 231
E_PictErr, 231
E_PictOff, 231
E_PictOK, 230
E_PictOn, 230
E_PMagazine, 240
E_PMagazinePlace, 240
E_PPAct, 226
E_PPdeload, 227
E_PPErr, 226
E_PPError, 229
E_PPload, 228
E_PPOK, 225
E_PPsel, 228
E_PTnumber, 240
E_RecCleared, 223
E_Req, 240
E_SCDark, 222
E_ScreenNo, 225
E_SimActiv, 224
E_Subtype, 239
E_TDC_Cancel, 240
E_TDC_Read, 240
E_TDC_Read_before_Write, 240
E_TDC_Write, 240
E_Tnumber, 239
E_ToolSize_Down, 239
E_ToolSize_Left, 239
E_ToolSize_Right, 239
E_ToolSize_Upper, 239
E_ToolState, 239
MMC1_ACTIVE_CHANGED, 236
MMC1_ACTIVE_PERM, 236
MMC1_ACTIVE_REQ, 236
MMC1_CHANGED_DENIED, 236
MMC1_CLIENT_IDENT, 235
MMC1_MSTT_ADR, 235
MMC1_MSTT_SHIFT_LOCK, 236
MMC1_SHIFT_LOCK, 236
MMC1_STATUS, 235
MMC1_TCU_SHIFT_LOCK, 236
MMC1_TRANS_DB, 236
MMC1_TYP, 235
MMC1_Z_INFO, 235
MMC2.A_ActWCS, 217
MMC2.A_ASpi1percent, 218
MMC2.A_ASpi2percent, 218
MMC2.A_AToolMeas, 219
MMC2.A_ClearCanAl, 217
MMC2.A_ClearRecAl, 217
MMC2.A_FuncNo, 231
MMC2.A_Hardkey, 219
MMC2.A_KeyDisable, 216
MMC2.A_MCPChan, 218
MMC2.A_OEM1, 219
MMC2.A_OEM2, 219
MMC2.A_Par1, 232
MMC2.A_Par2, 232
MMC2.A_Par3, 232
MMC2.A_PictNo, 230
MMC2.A_PictOff, 230
MMC2.A_PictOn, 230
MMC2.A_PPdeload, 219
MMC2.A_PPload, 220
MMC2.A_PPsel, 220
MMC2.A_SCDark, 216
MMC2.A_SCLight, 215
MMC2.A_TeachDisable, 219
MMC2.A_V24Index1, 220
MMC2.A_V24Index2, 220
MMC2.A_V24NoFile1, 221
MMC2.A_V24NoFile2, 222
MMC2.E_ActivWA, 224
MMC2.E_ActWCS, 224
MMC2.E_Cancel, 223
MMC2.E_CanCleared, 223
MMC2.E_Chan, 224
MMC2.E_FuncErr, 232
MMC2.E_NotAct, 231
MMC2.E_PictAct, 231
MMC2.E_PictErr, 231

MMC2.E_PictOff, 231
MMC2.E_PictOK, 230
MMC2.E_PictOn, 230
MMC2.E_PPAct, 226
MMC2.E_PPdeload, 227
MMC2.E_PPErr, 226
MMC2.E_PPError, 229
MMC2.E_PPload, 228
MMC2.E_PPOK, 225
MMC2.E_PPsel, 228
MMC2.E_RecCleared, 223
MMC2.E_SCDark, 222
MMC2.E_ScreenNo, 225
MMC2.E_SimActiv, 224
MMC2_ACTIVE_CHANGED, 238
MMC2_ACTIVE_PERM, 238
MMC2_ACTIVE_REQ, 237
MMC2_CHANGED_DENIED, 238
MMC2_CLIENT_IDENT, 237
MMC2_MSTT_ADR, 237
MMC2_MSTT_SHIFT_LOCK, 237
MMC2_SHIFT_LOCK, 237
MMC2_STATUS, 237
MMC2_TCU_SHIFT_LOCK, 238
MMC2_TRANS_DB, 238
MMC2_TYP, 237
MMC2_Z_INFO, 237
MtoNAlive, 234
ONL_CONFIRM, 233
ONL_REQUEST, 233
PAR_CLIENT_IDENT, 233
PAR_MMC_TYP, 234
PAR_MSTT_ADR, 234
PAR_STATUS, 234
PAR_Z_INFO, 234
ParOpKeyAdr, 234
ParTcuIndex, 234
Signals from the operator panel, 210
Signals to the operator panel, 209
Signals to/from operator panel, sidescreen, 214
Tcu1Index, 235
Tcu1KeyAdr, 234
Tcu2Index, 235
Tcu2KeyAdr, 234
LBP_HTCtrlHT2 [FC13], 698
LBP_HTCtrlHT8 [FC26], 744
LBP_MainBP [FC2] (basic program, cyclic section), 671
LBP_MCPCtrlMilling [FC19], 712
LBP_MCPCtrlMillingSmall [FC24], 729
LBP_MCPCtrlTurning [FC25], 737
LBP_MFuncDecListConfig [DB75]
MSigGrp, 545
LBP_MFuncDecListSignals [DB76]
MSigGrp1, 546
MSigGrp10, 546
MSigGrp11, 546
MSigGrp12, 546
MSigGrp13, 546
MSigGrp14, 546
MSigGrp15, 546
MSigGrp16, 546
MSigGrp2, 546
MSigGrp3, 546
MSigGrp4, 546
MSigGrp5, 546
MSigGrp6, 546
MSigGrp7, 546
MSigGrp8, 546
MSigGrp9, 546
LBP_ModeGroup [DB11]
A_AUTO, 187
A_INC1, 194
A_INC10, 195
A_INC100, 195
A_INC1000, 196
A_INC10000, 196
A_INCVar, 197
A_JOG, 188
A_MCDisable, 189
A_MDA, 188
A_MGReset, 191
A_MGStop, 189
A_MGStopASp, 190
A_REF, 192
A_REPOS, 192
A_SingleBlock_A, 193
A_SingleBlock_B, 192
A_TEACHIN, 191
E_AUTO, 199
E_ChanReset, 201
E_HMI_ContManTravel, 208
E_HMI_INC1, 205
E_HMI_INC10, 206
E_HMI_INC100, 206
E_HMI_INC1000, 207
E_HMI_INC10000, 207
E_HMI_INCVar, 208
E_INC1, 202
E_INC10, 203
E_INC100, 203
E_INC1000, 204
E_INC10000, 204

E_INCVar, 205
E_JOG, 200
E_MDA, 199
E_MGOK, 200
E_MGreseted, 200
E_MMC_AUTO, 197
E_MMC_JOG, 198
E_MMC_MDA, 197
E_MMC_REF, 199
E_MMC_REPOS, 198
E_MMC_TEACHIN, 198
E_NCKintJOG, 201
E_REF, 202
E_REPOS, 202
E_TEACHIN, 201
Mode signals 1 from the NC, 185
Mode signals 1 from the operating software, 186
Mode signals 2 from the NC, 186
Mode signals 2 to the NC, 186
Mode signals to the NC, 185
LBP_NC [DB10]
", 157
A_CA_DisableAutoMatch, 154
A_CA_DisableAutoTools, 154
A_CA_DisableAutoWorkh, 154
A_CA_DisableAutoWorkp, 154
A_CA_DisableJogMach, 154
A_CA_DisableJogTools, 154
A_CA_DisableJogWorkh, 154
A_CA_DisableJogWorkp, 154
A_CollCheck, 183
A_Disabl_anaIn1, 176
A_Disabl_anaIn2, 176
A_Disabl_anaIn3, 176
A_Disabl_anaIn4, 176
A_Disabl_anaIn5, 176
A_Disabl_anaIn6, 176
A_Disabl_anaIn7, 176
A_Disabl_anaIn8, 176
A_Disabl_anaOut1, 179
A_Disabl_anaOut2, 179
A_Disabl_anaOut3, 179
A_Disabl_anaOut4, 179
A_Disabl_anaOut5, 179
A_Disabl_anaOut6, 179
A_Disabl_anaOut7, 179
A_Disabl_anaOut8, 179
A_Disabl_Inp1, 145
A_Disabl_Inp10, 145
A_Disabl_Inp11, 145
A_Disabl_Inp12, 145
A_Disabl_Inp13, 145
A_Disabl_Inp14, 145
A_Disabl_Inp15, 145
A_Disabl_Inp16, 145
A_Disabl_Inp17, 145
A_Disabl_Inp18, 145
A_Disabl_Inp19, 145
A_Disabl_Inp2, 145
A_Disabl_Inp20, 145
A_Disabl_Inp21, 145
A_Disabl_Inp22, 145
A_Disabl_Inp23, 145
A_Disabl_Inp24, 145
A_Disabl_Inp25, 145
A_Disabl_Inp26, 145
A_Disabl_Inp27, 145
A_Disabl_Inp28, 145
A_Disabl_Inp29, 145
A_Disabl_Inp3, 145
A_Disabl_Inp30, 145
A_Disabl_Inp31, 145
A_Disabl_Inp32, 145
A_Disabl_Inp33, 145
A_Disabl_Inp34, 145
A_Disabl_Inp35, 145
A_Disabl_Inp36, 145
A_Disabl_Inp37, 145
A_Disabl_Inp38, 145
A_Disabl_Inp39, 145
A_Disabl_Inp4, 145
A_Disabl_Inp40, 145
A_Disabl_Inp5, 145
A_Disabl_Inp6, 145
A_Disabl_Inp7, 145
A_Disabl_Inp8, 145
A_Disabl_Inp9, 145
A_Disabl_Out1, 147
A_Disabl_Out10, 147
A_Disabl_Out11, 147
A_Disabl_Out12, 147
A_Disabl_Out13, 147
A_Disabl_Out14, 147
A_Disabl_Out15, 147
A_Disabl_Out16, 147
A_Disabl_Out17, 147
A_Disabl_Out18, 147
A_Disabl_Out19, 147
A_Disabl_Out2, 147
A_Disabl_Out20, 147
A_Disabl_Out21, 147
A_Disabl_Out22, 147
A_Disabl_Out23, 147
A_Disabl_Out24, 147

A_Disabl_Out25, 147
A_Disabl_Out26, 147
A_Disabl_Out27, 147
A_Disabl_Out28, 147
A_Disabl_Out29, 147
A_Disabl_Out3, 147
A_Disabl_Out30, 147
A_Disabl_Out31, 147
A_Disabl_Out32, 147
A_Disabl_Out33, 147
A_Disabl_Out34, 147
A_Disabl_Out35, 147
A_Disabl_Out36, 147
A_Disabl_Out37, 147
A_Disabl_Out38, 147
A_Disabl_Out39, 147
A_Disabl_Out4, 147
A_Disabl_Out40, 147
A_Disabl_Out5, 147
A_Disabl_Out6, 147
A_Disabl_Out7, 147
A_Disabl_Out8, 147
A_Disabl_Out9, 147
A_EMERGENCY, 152
A_EMERGENCY_Ackn, 153
A_IncInModeGroup, 154
A_InMask_anaIn1, 176
A_InMask_anaIn2, 176
A_InMask_anaIn3, 176
A_InMask_anaIn4, 176
A_InMask_anaIn5, 176
A_InMask_anaIn6, 176
A_InMask_anaIn7, 176
A_InMask_anaIn8, 176
A_InMask_anaOut1, 178
A_InMask_anaOut2, 178
A_InMask_anaOut3, 178
A_InMask_anaOut4, 178
A_InMask_anaOut5, 178
A_InMask_anaOut6, 178
A_InMask_anaOut7, 178
A_InMask_anaOut8, 178
A_InMask_Out1, 151
A_InMask_Out10, 151
A_InMask_Out11, 151
A_InMask_Out12, 151
A_InMask_Out13, 151
A_InMask_Out14, 151
A_InMask_Out15, 151
A_InMask_Out16, 151
A_InMask_Out17, 151
A_InMask_Out18, 151
A_InMask_Out19, 151
A_InMask_Out2, 151
A_InMask_Out20, 151
A_InMask_Out21, 151
A_InMask_Out22, 151
A_InMask_Out23, 151
A_InMask_Out24, 151
A_InMask_Out25, 151
A_InMask_Out26, 151
A_InMask_Out27, 151
A_InMask_Out28, 151
A_InMask_Out29, 151
A_InMask_Out3, 151
A_InMask_Out30, 151
A_InMask_Out31, 151
A_InMask_Out32, 151
A_InMask_Out33, 151
A_InMask_Out34, 151
A_InMask_Out35, 151
A_InMask_Out36, 151
A_InMask_Out37, 151
A_InMask_Out38, 151
A_InMask_Out39, 151
A_InMask_Out4, 151
A_InMask_Out40, 151
A_InMask_Out5, 151
A_InMask_Out6, 151
A_InMask_Out7, 151
A_InMask_Out8, 151
A_InMask_Out9, 151
A_Keyswitch0, 153
A_Keyswitch1, 153
A_Keyswitch2, 153
A_Keyswitch3, 153
A_OvMask_anaOut1, 177
A_OvMask_anaOut2, 177
A_OvMask_anaOut3, 177
A_OvMask_anaOut4, 177
A_OvMask_anaOut5, 177
A_OvMask_anaOut6, 177
A_OvMask_anaOut7, 177
A_OvMask_anaOut8, 177
A_OvMask_Out1, 148
A_OvMask_Out10, 148
A_OvMask_Out11, 148
A_OvMask_Out12, 148
A_OvMask_Out13, 148
A_OvMask_Out14, 148
A_OvMask_Out15, 148
A_OvMask_Out16, 148
A_OvMask_Out17, 148
A_OvMask_Out18, 148

A_OvMask_Out19, 148
A_OvMask_Out2, 148
A_OvMask_Out20, 148
A_OvMask_Out21, 148
A_OvMask_Out22, 148
A_OvMask_Out23, 148
A_OvMask_Out24, 148
A_OvMask_Out25, 148
A_OvMask_Out26, 148
A_OvMask_Out27, 148
A_OvMask_Out28, 148
A_OvMask_Out29, 148
A_OvMask_Out3, 148
A_OvMask_Out30, 148
A_OvMask_Out31, 148
A_OvMask_Out32, 148
A_OvMask_Out33, 148
A_OvMask_Out34, 148
A_OvMask_Out35, 148
A_OvMask_Out36, 148
A_OvMask_Out37, 148
A_OvMask_Out38, 148
A_OvMask_Out39, 148
A_OvMask_Out4, 148
A_OvMask_Out40, 148
A_OvMask_Out5, 148
A_OvMask_Out6, 148
A_OvMask_Out7, 148
A_OvMask_Out8, 148
A_OvMask_Out9, 148
A_PC_OSfault, 173
A_RobotStatus, 184
A_Set_Inp1, 146
A_Set_Inp10, 146
A_Set_Inp11, 146
A_Set_Inp12, 146
A_Set_Inp13, 146
A_Set_Inp14, 146
A_Set_Inp15, 146
A_Set_Inp16, 146
A_Set_Inp17, 146
A_Set_Inp18, 146
A_Set_Inp19, 146
A_Set_Inp2, 146
A_Set_Inp20, 146
A_Set_Inp21, 146
A_Set_Inp22, 146
A_Set_Inp23, 146
A_Set_Inp24, 146
A_Set_Inp25, 146
A_Set_Inp26, 146
A_Set_Inp27, 146
A_Set_Inp28, 146
A_Set_Inp29, 146
A_Set_Inp3, 146
A_Set_Inp30, 146
A_Set_Inp31, 146
A_Set_Inp32, 146
A_Set_Inp33, 146
A_Set_Inp34, 146
A_Set_Inp35, 146
A_Set_Inp36, 146
A_Set_Inp37, 146
A_Set_Inp38, 146
A_Set_Inp39, 146
A_Set_Inp4, 146
A_Set_Inp40, 146
A_Set_Inp5, 146
A_Set_Inp6, 146
A_Set_Inp7, 146
A_Set_Inp8, 146
A_Set_Inp9, 146
A_Set_Out1, 150
A_Set_Out10, 150
A_Set_Out11, 150
A_Set_Out12, 150
A_Set_Out13, 150
A_Set_Out14, 150
A_Set_Out15, 150
A_Set_Out16, 150
A_Set_Out17, 150
A_Set_Out18, 150
A_Set_Out19, 150
A_Set_Out2, 150
A_Set_Out20, 150
A_Set_Out21, 150
A_Set_Out22, 150
A_Set_Out23, 150
A_Set_Out24, 150
A_Set_Out25, 150
A_Set_Out26, 150
A_Set_Out27, 150
A_Set_Out28, 150
A_Set_Out29, 150
A_Set_Out3, 150
A_Set_Out30, 150
A_Set_Out31, 150
A_Set_Out32, 150
A_Set_Out33, 150
A_Set_Out34, 150
A_Set_Out35, 150
A_Set_Out36, 150
A_Set_Out37, 150
A_Set_Out38, 150

A_Set_Out39, 150
A_Set_Out4, 150
A_Set_Out40, 150
A_Set_Out5, 150
A_Set_Out6, 150
A_Set_Out7, 150
A_Set_Out8, 150
A_Set_Out9, 150
A_Setval_anaIn1, 177
A_Setval_anaIn2, 177
A_Setval_anaIn3, 177
A_Setval_anaIn4, 177
A_Setval_anaIn5, 177
A_Setval_anaIn6, 177
A_Setval_anaIn7, 177
A_Setval_anaIn8, 177
A_Setval_anaOut1, 180
A_Setval_anaOut2, 180
A_Setval_anaOut3, 180
A_Setval_anaOut4, 180
A_Setval_anaOut5, 180
A_Setval_anaOut6, 180
A_Setval_anaOut7, 180
A_Setval_anaOut8, 180
Analog NC inputs and outputs, 142
Collision avoidance: Activate protection area, 143
Collision avoidance:protection area active, 143
Disabl_AlarmDp, 159
Disabl_AlarmMpiDp, 159
Disabl_AlarmPn, 160
DpOk, 159
E_611Dready, 172
E_ActVal_anaIn1, 180
E_ActVal_anaIn2, 180
E_ActVal_anaIn3, 180
E_ActVal_anaIn4, 180
E_ActVal_anaIn5, 180
E_ActVal_anaIn6, 180
E_ActVal_anaIn7, 180
E_ActVal_anaIn8, 180
E_ActVal_In1, 156
E_ActVal_In10, 156
E_ActVal_In11, 156
E_ActVal_In12, 156
E_ActVal_In13, 156
E_ActVal_In14, 156
E_ActVal_In15, 156
E_ActVal_In16, 156
E_ActVal_In17, 156
E_ActVal_In18, 156
E_ActVal_In19, 156
E_ActVal_In2, 156
E_ActVal_In20, 156
E_ActVal_In21, 156
E_ActVal_In22, 156
E_ActVal_In23, 156
E_ActVal_In24, 156
E_ActVal_In25, 156
E_ActVal_In26, 156
E_ActVal_In27, 156
E_ActVal_In28, 156
E_ActVal_In29, 156
E_ActVal_In3, 156
E_ActVal_In30, 156
E_ActVal_In31, 156
E_ActVal_In32, 156
E_ActVal_In33, 156
E_ActVal_In34, 156
E_ActVal_In35, 156
E_ActVal_In36, 156
E_ActVal_In37, 156
E_ActVal_In38, 156
E_ActVal_In39, 156
E_ActVal_In4, 156
E_ActVal_In40, 156
E_ActVal_In9, 156
E_ATempAlarm, 174
E_ATready, 168
E_Ax_A_HW1, 163
E_Ax_A_HW2, 163
E_Ax_A_HW3, 163
E_Ax_B_HW1, 163
E_Ax_B_HW2, 163
E_Ax_B_HW3, 163
E_Ax_C_HW1, 163
E_Ax_C_HW2, 163
E_Ax_C_HW3, 163
E_Ax_D_HW1, 163
E_Ax_D_HW2, 163
E_Ax_D_HW3, 163
E_Ax_E_HW1, 163
E_Ax_E_HW2, 163
E_Ax_E_HW3, 163
E_BattAlarm, 174
E_CA_DisableAutoMach, 160
E_CA_DisableAutoTools, 160
E_CA_DisableAutoWorkh, 160
E_CA_DisableAutoWorkp, 160
E_CA_DisableJogMach, 160
E_CA_DisableJogTools, 160
E_CA_DisableJogWorkh, 160
E_CA_DisableJogWorkp, 160
E_CancelToolCmd, 170
E_Chan_HW1_A, 162

E_Chan_HW1_B, 162
E_Chan_HW1_C, 162
E_Chan_HW1_D, 162
E_Chan_HW2_A, 162
E_Chan_HW2_B, 162
E_Chan_HW2_C, 162
E_Chan_HW2_D, 162
E_Chan_HW3_A, 162
E_Chan_HW3_B, 162
E_Chan_HW3_C, 162
E_Chan_HW3_D, 162
E_CHW_sel1, 165
E_CHW_sel2, 165
E_CHW_sel3, 165
E_CollCheck, 182
E_CollCheckOff, 170
E_CounterInchMetr, 158
E_DrivesInCylOp, 172
E_EHW1_stands, 183
E_EHW2_stands, 183
E_EHW3_stands, 183
E_EHW4_stands, 183
E_EHW5_stands, 183
E_EHW6_stands, 183
E_EMERGENCY, 170
E_FirstOB1, 169
E_HTempAlarm, 174
E_HTLostPackageWarn, 161
E_HTready, 169
E_HW_sel1, 166
E_HW_sel2, 166
E_HW_sel3, 166
E_HW1_moved, 157
E_HW2_moved, 157
E_HW3_moved, 157
E_HW4_moved, 157
E_HW5_moved, 157
E_HW6_moved, 157
E_InspProbe1, 170
E_InspProbe2, 170
E_LanguageID, 161
E_Mach_Ax1, 167
E_Mach_Ax2, 167
E_Mach_Ax3, 167
E_MCP1AxisValid, 158
E_MCP1AxKeyDisp, 158
E_MCP1LostPackageWarn, 161
E_MCP1ready, 169
E_MCP1WCS, 158
E_MCP2AxisValid, 158
E_MCP2AxKeyDisp, 158
E_MCP2LostPackageWarn, 161
E_MCP2ready, 169
E_MCP2WCS, 158
E_MMC2BTSSready, 171
E_MMC2MPIready, 171
E_MMC2Ready, 171
E_MMCBatt, 169
E_MMCTemp, 168
E_NCKalarm, 173
E_NCKready, 169
E_NCready, 172
E_NCU_LinkActive, 171
E_Op1KeyLostPackageWarn, 161
E_OP1KeyReady, 169
E_Op2KeyLostPackageWarn, 161
E_OP2KeyReady, 169
E_RemDiag, 168
E_RobotStatus, 184
E_Setpoint_Out1, 157
E_Setpoint_Out10, 157
E_Setpoint_Out11, 157
E_Setpoint_Out12, 157
E_Setpoint_Out13, 157
E_Setpoint_Out14, 157
E_Setpoint_Out15, 157
E_Setpoint_Out16, 157
E_Setpoint_Out17, 157
E_Setpoint_Out18, 157
E_Setpoint_Out19, 157
E_Setpoint_Out2, 157
E_Setpoint_Out20, 157
E_Setpoint_Out21, 157
E_Setpoint_Out22, 157
E_Setpoint_Out23, 157
E_Setpoint_Out24, 157
E_Setpoint_Out25, 157
E_Setpoint_Out26, 157
E_Setpoint_Out27, 157
E_Setpoint_Out28, 157
E_Setpoint_Out29, 157
E_Setpoint_Out3, 157
E_Setpoint_Out30, 157
E_Setpoint_Out31, 157
E_Setpoint_Out32, 157
E_Setpoint_Out33, 157
E_Setpoint_Out34, 157
E_Setpoint_Out35, 157
E_Setpoint_Out36, 157
E_Setpoint_Out37, 157
E_Setpoint_Out38, 157
E_Setpoint_Out39, 157
E_Setpoint_Out4, 157
E_Setpoint_Out40, 157

E_Setpoint_Out5, 157
E_Setpoint_Out6, 157
E_Setpoint_Out9, 157
E_SetVal_anaOut1, 181
E_SetVal_anaOut2, 181
E_SetVal_anaOut3, 181
E_SetVal_anaOut4, 181
E_SetVal_anaOut5, 181
E_SetVal_anaOut6, 181
E_SetVal_anaOut7, 181
E_SetVal_anaOut8, 181
E_SWCamMinus, 175
E_SWCamPlus, 175
E_SystemInchDim, 171
epsFromPlc, 159
epsToPlc, 159
External analog NC inputs, 140
External analog NC outputs, 141
External digital NC inputs, 139
External digital NC inputs and outputs, 142
External digital NC outputs, 139
General signals from the NC, 138
General signals to the NC, 135
Handwheel signals from the NC, 144
Inputs and outputs of the NC, 145
LBP_NC [DB10]E_Setpoint_Out7, 157
LBP_NC [DB10]E_Setpoint_Out8, 157
MCP1AxisFromHMI, 159
MCP1AxisTbl, 151
MCP1MaxAxis, 152
MCP2AxisFromHMI, 159
MCP2AxisTbl, 152
MCP2MaxAxis, 152
MpiDpOk, 159
Onboard inputs and outputs from the NC, 135
Onboard NC inputs/outputs, 133
PnOk, 159
Signals from the operating software, 136
Signals from the robot, 144
Signals to the robot, 145
LBP_NCKProcessIRT [FC3] (basic program, interrupt-driven section), 672
LBP_OpUnitComm [FB25000], 657
LBP_OpUnitWriteRecord [FB25002], 659
LBP_ParamAlarmMsgs [DB5]
Axes/spindles, 119
Channels, 118
User data, 120
LBP_Powerride, 648
LBP_PowerrideConfig
configFeed, 549
configSpindle, 549
LBP_ReadGUD [FB5] (read GUD variable), 611
LBP_ReadVar [FB2] (read NC variable), 597
LBP_ReqASUP [FC9], 687
LBP_ReqPIService [FB7] (PI services)
Available PI services, 621
LBP_ReqPIService [FB7] (request PI service), 618
LBP_SentronPac [DB1001]
ActiveEnergyDay, 564
ActiveEnergyItem, 564
ActiveEnergyMonth, 565
ActiveEnergyPrevDay, 565
ActiveEnergyPrevMonth, 565
ActiveEnergyPrevYear, 565
ActiveEnergySentron, 564
ActiveEnergyYear, 565
ActivePowerItem, 564
ActivePowerSentron, 564
CtrlE_AuxDevice, 567
ErrDBD104_NAN, 566
ErrDBD144_NAN, 566
ErrDBD184_NAN, 566
ErrDBD20_NAN, 566
ErrDBD224_NAN, 566
ErrDBD24_NAN, 566
ErrDBD264_NAN, 567
ErrDBD28_NAN, 566
ErrDBD304_NAN, 567
ErrDBD344_NAN, 567
ErrDBD384_NAN, 567
ErrDBD424_NAN, 567
ErrDBD464_NAN, 567
EStateOn, 563
FeedBackEnergyReadAuxil, 563
FeedBackEnergyReadSP, 563
FeedInEnergyReadAuxil, 563
FeedInEnergyReadSP, 563
ManualVal, 562
ManualValue, 564
MeasAct, 563
MeasAuto, 563
ProductionAct, 566
ReactiveEnergyDay, 564
ReactiveEnergyItem, 564
ReactiveEnergyMonth, 565
ReactiveEnergyPrevDay, 565
ReactiveEnergyPrevMonth, 565
ReactiveEnergyPrevYear, 565
ReactiveEnergySentron, 564
ReactiveEnergyYear, 565
SENTRON PAC, 559
SENTRON PAC auxiliary devices, 561
SentronPacMach, 563

LBP_SpindleTempSensor [DB1002]
SP, 572
Spindle temperature sensors, 570
LBP_TMLoad [DB71]
IFNo, 516
Magazine interface, 515
Quit, 517
TIF, 517
LBP_TMLoadMT [DB1071]
MTIF, 575
Multitool magazine interface, 575
LBP_TMSpindle [DB72]
IFNo, 524
Quit, 524
Spindle interface, 522
TIF, 524
LBP_TMSpindleMT [DB1072]
MTIF, 578
Multitool spindle interface, 577
LBP_TMTransfer [FC8], 678
LBP_TMTransferMT [FC6], 674
LBP_TMTransferTurret [FC7], 676
LBP_TMTurret [DB73]
IFNo, 534
Quit, 534
TIF, 535
Turret interface, 532
LBP_TMTurretMT [DB1073]
MTIF, 584
Multitool turret interface, 583
LBP_TransferSelData [FC21], 720
LBP_WriteVar [FB3] (write NC variable), 605

M

M decoding acc. to list, 541
Machine control panel
M version, 589, 593
Slimline version, 590, 594
T version, 590, 593
MD22510, 329
MD35400, 706
Multitool, 644

N

Number range
for user program, 33

P

PLC basic program
Assignment overview, 33
Block listing as table, 87, 89, 91
Blocks with user-specific adaptations, 44
Execution structure, 45
Programming recommendations, 4

S

STRUCT, (A_Geo), (A_Ori), (Axis), (AxisA), (Chan), (ChanA), (CtrlE_AuxDevice), (E_Geo), (E_Ori), (GenerateAlarmMsgs), (HTIn), (HTOut), (MCP1In), (MCP1Out), (MCP2In), (MCP2Out), (MSigGrp), (MSigGrp*), (MTIF), (Op1KeyIn), (Op1KeyOut), (Op2KeyIn), (Op2KeyOut), (Profiles), (SCC), (SIC), (SP), (TIF), (TS), (UserA), (UserVersion)
LBP_typeTMMagazines, 662
Magazines, 663
STRUCT A_Geo
ContManTravel, 357
Disabl, 352
FDStop, 352
HW1, 350
HW2, 350
HW3, 351
INC1, 356
INC10, 356
INC100, 356
INC1000, 356
INC10000, 356
INCVar, 356
InvHWDirOfRot, 358
Minus, 353
OEM, 358
Plus, 354
RapidTrOR, 353
STRUCT A_Ori
Disabl, 368
FDStop, 368
HW1, 366
HW2, 366
HW3, 367
INC1, 372
INC10, 372
INC100, 372
INC1000, 372
INC10000, 372
INCVar, 373

InvHWDirOfRot, 374
Minus, 369
OEM, 374
Plus, 370
RapidTrOR, 368
STRUCT Axis
FdStop, 114
STRUCT AxisA
AAFdStop60AAxx, 117
STRUCT Chan
FDD, 113
FdStop_1, 114
FdStop_2, 114
FdStop_3, 114
NCSD, 114
RID, 114
STRUCT ChanA
FDD_5C00xx, 115
FDD_RID_5C01xx, 115
FdStop_1_5C11xx, 116
FdStop_2_5C12xx, 116
FdStop_3_5C13xx, 116
NCSD_5C03xx, 116
RID_5C02xx, 115
STRUCT CtrlE_AuxDevice
ActiveEnAux, 569
ActiveEnAuxMeasOFF, 569
ActiveEnAuxMeasON, 569
ActivePowerAux, 569
InputMode, 568
MeasAct, 568
MeasMode, 568
ProcessAux, 568
ReActiveEnAux, 569
ReActiveEnAuxMeasOFF, 569
ReadEnergy, 568
ResetAux, 568
STRUCT E_Geo
HW1, 359
HW2, 359
HW3, 360
INC1, 364
INC10, 364
INC100, 364
INC1000, 364
INC10000, 364
INCVar, 365
InvHWDirOfRot, 366
OEM, 365
TCMinus, 362
TCPlus, 363
TReqMinus, 360
TReqPlus, 361
STRUCT E_Ori
HW1, 375
HW2, 375
HW3, 375
INC1, 379
INC10, 379
INC100, 379
INC1000, 379
INC10000, 379
INCVar, 380
InvHWDirOfRot, 380
OEM, 380
TCMinus, 377
TCPlus, 378
TReqMinus, 376
TReqPlus, 376
STRUCT GenerateAlarmMsgs
Error, 133
NumActAlarmMsgs, 132
Status, 133
StatusId, 133
STRUCT LBP_typeTMMagazines
MaxSpindleNumber, 662
NumberOfMagazines, 663
STRUCT Magazines, 663
STRUCT Magazines
Number, 663
Places, 663
TMNumber, 663
Type, 663
STRUCT MSigGrp
MExtAdr, 545
MFirstAdr, 545
MLastAdr, 546
STRUCT MSigGrp*
MSig, 547
STRUCT MTIF
Active, 581, 587
Change, 581, 587
Distance, 576, 579, 585
DistanceCoding, 576, 578, 584
Enabled, 581, 587
Exchange_1to1, 581, 587
FixedLoc, 581, 587
IgnoreDisabled, 581, 587
Load, 581, 587
LocType, 580, 586
Measured, 581, 587
MTNo, 576
MyMag, 582, 588

MyPlace, 583, 588
NewMTNo, 579, 585
NewPlace, 585
NewPlaceNo, 579
OldMTNo, 579, 585
OldPlace, 580, 585
PlaceNo, 577
Places, 576, 579, 584
Prewarn, 581, 587
SBottom, 581, 587
SLeft, 580, 586
SRight, 580, 586
STop, 581, 586
TInBuffer, 581, 587
TNo, 582, 588
Tool holder, 577, 582, 588
Unload, 581, 587
Used, 581, 587
STRUCT profile
ActValT1, 557
ActValT2, 557
CondDataTransfer, 558
CondMasterCompSig, 558
CondMCP, 558
CondNcChan1, 558
CondNcChan10, 558
CondNcChan2, 558
CondNcChan3, 558
CondNcChan4, 558
CondNcChan5, 558
CondNcChan6, 558
CondNcChan7, 558
CondNcChan8, 558
CondNcChan9, 558
CondOpPanel, 557
CondPlcUserSig, 558
CondWindowChange, 558
ESaveIsOn, 557
ESaveOff, 557
ESaveOn, 556
ESaveValid, 557
HmiESaveOn, 556
MasterCompSig, 556
PlcUserSig, 557
PresetT1, 558
PresetT2, 559
ReloadT2, 556
T1Expired, 557
STRUCT SCC
CW1.CloseBrake, 511
CW1.TeststopStart, 511
CW3.Brake_1_2, 511
CW3.BrakeClosed, 511
CW3.BT_Start, 511
CW3.DirOfROT, 511
CW3.SelectionSBT, 511
CW3.Testseq_1_2, 511
STRUCT SIC
SW1.ESR_Req, 512
SW1.InternalEvent, 513
SW1.SF_Message_Act, 512
SW1.SLA_Sel, 512
SW1.SLS_Act, 513
SW1.SLS_LimBit0_Sel, 512
SW1.SLS_LimBit1_Sel, 512
SW1.SLS_Sel, 513
SW1.SOS, 513
SW1.SOS_Act, 512
SW1.SS1_Act, 512
SW1.SS2_Act, 512
SW1.STO_Act, 512
SW2.SDI_N_Sel, 513
SW2.SDI_P_Sel, 513
SW2.SLP_LimBit0_Sel, 513
SW2.SLP_Sel, 513
SW2.TestStopAct, 513
SW2.TestStopReq, 513
SW3.AcceptSLPAct, 514
SW3.AcceptTestSel, 514
SW3.Brake_1_2, 514
SW3.BrakeReq, 514
SW3.BT_Act, 514
SW3.BT_Finished, 514
SW3.BT_OK, 514
SW3.SBT_Sel, 514
SW3.SetpSettingDrive, 514
SW3.SignLoadTorque, 515
SW3.SS2E_ACTIVE, 514
SW3.SS2ESR, 514
V_Limit_B, 513
STRUCT SP
TS, 573
STRUCT TIF
Active, 531, 540
BufferIdent, 528, 537
BufferNo, 528, 537
Change, 531, 540
Channel, 520, 528, 536
Comp, 525, 535
DataInDB.1072, 527
DataInDB.1073, 536
DataInDB1071, 520
Enabled, 531, 540
Exchange_1to1, 531, 540

FixedLoc, 531, 540
IgnoreDisabled, 531, 540
Load, 531, 540
LoadIdent, 520
Loading, 517
LoadNo, 520
LocType, 530, 538
ManTIn, 526
ManTOut, 527, 536
Measured, 531, 540
MyMag, 532
MyPlace, 532, 540
OldTBufferNo, 532
OldTinDBW42, 526
Perform, 525, 535
PerfStatusError, 519, 527, 536
PosByProgram, 519
Positioning, 518
PositioningMT, 519
Prepare, 526
Prewarn, 531, 540
Reloading, 518
SBottom, 531, 539
SLeft, 530, 539
SLoc, 521, 529, 538
SMag, 521, 529, 538
SpiTRSpi, 527
SRight, 530, 539
STop, 530, 539
T0, 526, 536
TInBuffer, 531, 540
TLoc, 521, 529, 538
TMag, 521, 529, 538
TMNo, 520, 528, 537
TNo, 531, 540
Unload, 531, 540
Unloading, 518
Used, 531, 540
VdiTcp0, 528, 537
VdiTcp1, 528
VdiTcp2, 528
WithoutMagMoving, 522
STRUCT TS
ActVal, 573
FaultDurLimitViol, 575
FaultNumLimitViol, 574
FaultThreshold, 574
FaultTsViolDD, 574
FaultTsViolH, 574
FaultTsViolM, 574
FaultTsViolMM, 574
FaultTsViolS, 575
FaultTsViolYY, 574
MountPlace, 573
WarnDurLimitViol, 574
WarnNumLimitViol, 573
WarnThreshold, 573
WarnTsViolDD, 573
WarnTsViolH, 574
WarnTsViolM, 574
WarnTsViolMM, 573
WarnTsViolS, 574
WarnTsViolYY, 573
STRUCT UserA
A70UUxx, 117

U

UDT, (LBP_typePowerrideConfig), (LBP_typePowerrideConfigPos), (LBP_typePowerrideInput), (LBP_typePowerrideOutput)
UDT LBP_typePowerrideConfig
delay100, 553
pos100, 553
posMax, 553
switchPos, 553
UDT LBP_typePowerrideConfigPos
ledColor, 552
value, 552
vibrationIndex, 552
UDT LBP_typePowerrideInput
powerrideCounter, 550
pushButtons, 550
UDT LBP_typePowerrideOutput
buttonBlue, 551
buttonGreen, 551
buttonRed, 551
ledInfo, 551
scaleBlue, 551
scaleGreen, 551
scaleRed, 551
vibrationPattern, 551