SIEMENS

Solutions for Powertrain

Software Guide
PLC Programming Guide
SINUMERIK Integrate
Run MyHMI /PRO (S7-1500)

2018 Edition

siemens.com/TRANSLINE

SIEMENS

Solutions for Powertrain

TRANSLINE sl - Software Guide PLC Programming Guide SINUMERIK Integrate Run MyHMI /PRO (S7-1500)

Manual

Valid for:
SIMATIC S7-1500

Edition 2018

SINUMERIK® documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the "Remarks" column.

Status codes in the "Remarks" columns:

A .... New documentation.
B .... Unrevised reprint with new order number.
C .... Revised edition with new status.

Edition	Order No.	Comment
2018	DF MC SIOS Extranet	**A**

Further information is available on the Internet under:
http://www.automation.siemens.com

Tested Siemens quality for software and training in accordance with DIN ISO 9001, Reg. No. 2160-01

Although it is possible that other functions not described in this documentation can run in the controller, there can be no claim for these functions for a new delivery or in a service case.

We have checked the content of the publication for agreement with the described hardware and software. Discrepancies, however, cannot be excluded. The information in this publication is reviewed regularly and any necessary corrections included in subsequent editions. We are grateful for improvement suggestions.

We reserve the right to make technical changes.

Available at: DF MC-SIOS Extranet
Printed in the Federal Republic of Germany

Siemens Aktiengesellschaft

Contents

Figures

Tables

1

1 General information

Uniform software structures across the TRANSLINE Solution for Powertrain assist machine manufacturers, maintenance engineers and service personnel when carrying out commissioning, troubleshooting and maintenance work.
The Software Guide describes the software structures for PLC and CNC-controlled machines, as well as agreed responses to operator actions.

The Manual includes the description and handling of the HMI PRO standard blocks for use with HMI PRO sl RT.

1.1 Preconditions

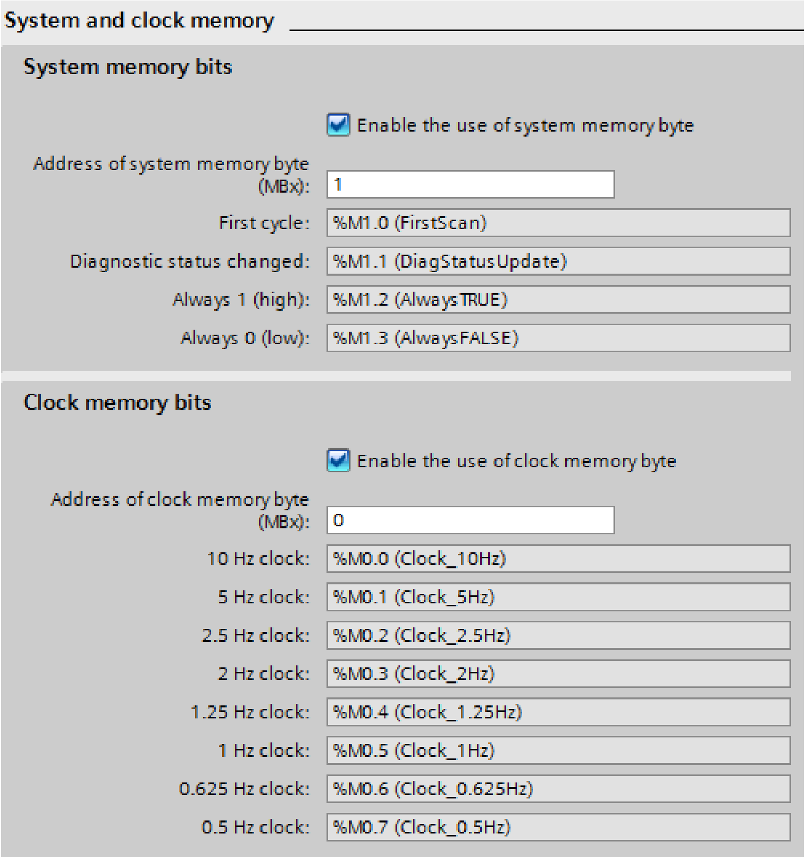
Validity

The descriptions and examples are applicable for the following versions:

SINUMERIK Operate	from V04.08.x.x
TRANSLINE library from	V2.1.x
TIA Portal	from V14 SP1
HMI PRO sl RT	from V08.x
HMI PRO CS	from V08.x

System and clock cycle flags

The system and clock cycle flag bits have been taken from the TIA standard. They must be activated in the **Properties of the CPU** under **General.**



System and clock memory

System memory bits

☑ Enable the use of system memory byte

Address of system memory byte (MBx):	1
First cycle:	%M1.0 (FirstScan)
Diagnostic status changed:	%M1.1 (DiagStatusUpdate)
Always 1 (high):	%M1.2 (AlwaysTRUE)
Always 0 (low):	%M1.3 (AlwaysFALSE)

Clock memory bits

☑ Enable the use of clock memory byte

Address of clock memory byte (MBx):	0
10 Hz clock:	%M0.0 (Clock_10Hz)
5 Hz clock:	%M0.1 (Clock_5Hz)
2.5 Hz clock:	%M0.2 (Clock_2.5Hz)
2 Hz clock:	%M0.3 (Clock_2Hz)
1.25 Hz clock:	%M0.4 (Clock_1.25Hz)
1 Hz clock:	%M0.5 (Clock_1Hz)
0.625 Hz clock:	%M0.6 (Clock_0.625Hz)
0.5 Hz clock:	%M0.7 (Clock_0.5Hz)

Fig. 1-1: System and clock cycle flags

In the TIA default setting, MB1 is used for the system flag and MB0 is used for the clock cycle flag. By using the system clock cycle flag bits, it is no longer necessary to program the symbols **AlwaysTrue** etc., which in STEP 7 Classic still had to be programmed in OB1.

1.2 TRANSLINE library

The HMI PRO standard blocks for using HMI PRO together with an S7-1500 controller are included in the TIA-TRANSLINE library **LTLP_Transline**. The library comprises types and copy templates.
An overview and brief information about the blocks is provided in Chapter 2 Block overview - the use and description of the individual blocks and copy templates, in the relevant chapters.

1.3 Blocks that are absolutely necessary when using HMI PRO

The standard blocks required for HMI PRO running under SINUMERIK Operate are saved in the TRANSLINE library under Copy templates in directories **840evo PLC BP** and **840evo PLC BP data types**.

LBP_NC [DB10] and LBP_HMI [DB19]

From the basic program, you must use **LBP_NC** [DB10] when using an NC, and **LBP_HMI** [DB19] when using a SIMATIC PLC. It is not permissible that the numbers of these blocks are changed, as SINUMERIK Operate and HMI PRO directly access these blocks.

LBP_OpUnitComm [FB25000] and LBP_ConfigData [DB7]

You must call the **LBP_OpUnitComm** [FB25000] in the cyclic program if you connect an HT8 or a TCU. It is appropriately parameterized via the **LBP_ConfigData** [DB7].

Communication

For communication with the HT8, the DNS configuration must be parameterized in the properties of the CPU.

A more precise description of how **LBP_ConfigData** is supplied and using the **LBP_OpUnitComm** is provided in Chapter 4.2 Connecting an IPC with PCU Base and additional control elements.

LTLP_HmiProInterfaceData [DB501]

This data block represents the data interface between HMI PRO and the PLC user program. It is saved in the TRANSLINE library under copy templates in the **common** directory
In exceptional cases, the data block can also be stored in the controller under a different number. This must be entered in the configuration tool HMI PRO CS V8 under **Configuration** ▸ **Default settings** ▸ **Control via PLC** ▸ **User-defined standard DB** and subsequently activated.

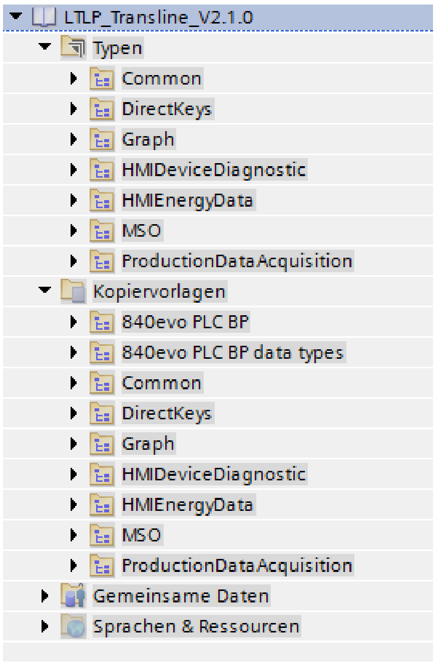
■

2

2 Block overview

2.1 Structure of the TRANSLINE library LTLP_Transline

The TRANSLINE library comprises types and copy templates.



- LTLP_Transline_V2.1.0
  - Typen
    - Common
    - DirectKeys
    - Graph
    - HMIDeviceDiagnostic
    - HMIEnergyData
    - MSO
    - ProductionDataAcquisition
  - Kopiervorlagen
    - 840evo PLC BP
    - 840evo PLC BP data types
    - Common
    - DirectKeys
    - Graph
    - HMIDeviceDiagnostic
    - HMIEnergyData
    - MSO
    - ProductionDataAcquisition
  - Gemeinsame Daten
  - Sprachen & Ressourcen

Fig. 2-1: TRANSLINE library blocks in the TIA Portal

Types

If you are using the TRANSLINE library for the first time in your TIA project: Open the library under **Global libraries**, and copy the **Types** directory into the **Program blocks** directory of your TIA project. The folder structure from the library is kept.

If you are already using blocks from the TRANSLINE library in your TIA project, and wish to use a newer version:
Update the **Types** using the context menu ▸ **Update project** in the actual library - or copy the **Types** directory to the **Program blocks** directory of your TIA project. The dependencies are accepted in both cases.

Copy templates

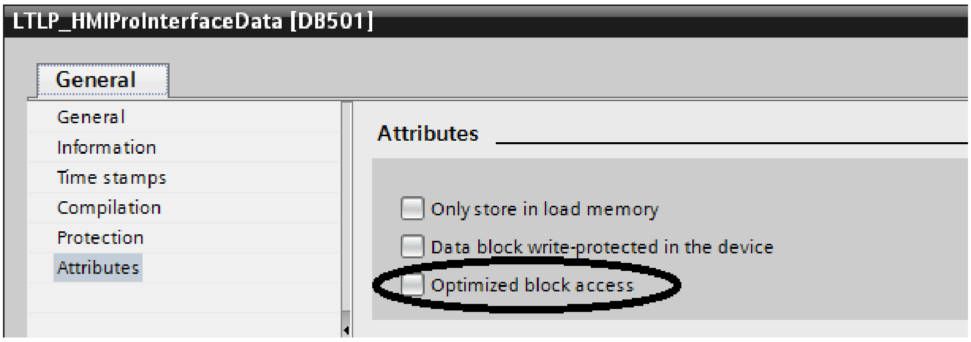
The blocks from the copy templates are not accepted in the project when updating the types. They must be separately copied into the project.

Copy the following copy templates into your TIA project:

- Blocks: all required directories, with the exception of **840evo PLC BP data types,** in the **Program blocks** directory
- Data types: Directory **840evo PLC BP data types** in the **PLC data types** directory

Attention

In the TIA Portal – for all data blocks – on the **HMI PRO accesses** (direct and indirect), the **Optimized block access** setting must be deselected.



LTLP_HMIProInterfaceData [DB501]

Attributes

☐ Only store in load memory
☐ Data block write-protected in the device
☐ Optimized block access

For **GRAPH** blocks, setting **Optimized block access** is set as default and cannot be changed.

2.2 Basic program for SINUMERIK Operate

Copy templates

The copy templates under **840evo PLC BP** and **840evo PLC BP data types** are always required if an S7 controller is being used.

Copy the following copy templates into your TIA project:

- **840evo PLC BP** in the **program blocks** directory
- **840evo PLC BP data types** in the **PLC data types** directory.

A more precise description of using these blocks is provided in Chapter 4.2 Connecting an IPC with PCU Base and additional control elements.

2.2.1 840evo PLC BP

Symbolic name	Block number	Description
DB		
LBP_AlarmMsgs	DB2	PLC alarms/messages: Interface between user program and **LBP_GenAlarmMsgs** [FC10], contains bit arrays for blocking and stop signals as well as error and operating messages.
LBP_ConfigData	DB7	Basic program initialization: Parameters to configure and initialize the basic program
LBP_HMI	DB19	Operator panel signals: Interface between user program and operator panel, contains signals from/to the operating software.
LBP_NC	DB10	NC signals: Interface between user program and NC, contains signals from/to the NC, PLC and operating software.
LBP_ParamAlarmMsgs	DB5	PLC alarms/messages: 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 error or operating message.
FB		
LBP_OpUnitComm	FB25000	Communication with handheld terminals and direct keys
FC		
LBP_GenerateAlarmMsgs	FC10	Evaluation of the signals entered in **LBP_AlarmMsgs** [DB2], generation of incoming and outgoing operating messages, acknowledgment of error messages

2.2.2 840evo PLC BP data types

Symbolic name	Description
LBP_typeAlarmMsgs	PLC alarms/messages: User data type for block LBP_AlarmMessage [DB2]
LBP_typeConfigData	Basic program initialization: User data type for block **LBP_ConfigData** [DB7]
LBP_typeHMI	M to N operating unit switchover User data type for block **LBP_HMI** [DB19]
LBP_typeInternal1	Only for internal use
LBP_typeNC	NC interface: User data type for block **LBP_NC** [DB10]

2.3 Common

General types and blocks required when using HMI PRO are contained under **Common**.

Types

Symbolic name	Block number	Description
LTLP_typeHMIProInterface	UDT501	Structure
LTLP_typeWorkpieceCount	UDT502	Structure for the workpiece counter

Copy templates

Symbolic name	Block number	Description
LTLP_HmiProInterfaceData	DB501	Interface between HMI PRO and PLC (HMI-DB)

2.4 DirectKeys

Blocks that are required for the direct key screens of HMI PRO are contained under **DirectKeys**.

Types

Symbolic name	Block number	Description
LTLP_DirectKey	FC511	Direct key screens: Evaluation of the signals for the plus/minus keys Sets the plus/minus signals and the interface signals
LTLP_DirectKeyGetMoveNo	FC512	Direct key screens: Sets the corresponding bit of the currently selected line
LTLP_typeDirectKeyGetMoveNo	UDT512	Contains the structure, which is used to determine which line is selected in which direct key screen

Copy templates

Symbolic name	Block number	Description
LTLP_DirectKeyGetMovNoData	DB512	DB to evaluate the currently selected line of a direct key screen

2.5 Graph

Blocks for step sequence processing with GRAPH are contained under **Graph**.

Types

Symbolic name	Block number	Description
LTLP_GraphManualInputParam	FC563	Data transfer from HMI DB to the GRAPH extension DB
LTLP_GraphManualOutputParam	FC564	Data transfer from GRAPH extension DB to the HMI DB
LTLP_GraphManualExecutable	FB562	Setup screens: Executability and execution of manual movement in GRAPH
LTLP_typeGraphManualCell	UDT565	Structure: A manual movement with GRAPH
LTLP_typeGraphManualInputParam	UDT561	Input parameter, Extended FB
LTLP_typeGraphManualInternalData	UDT566	Internal data, extended FB
LTLP_typeGraphManualMovement	UDT563	Structure: Manual movement of a setup line with GRAPH
LTLP_typeGraphManualOutputParam	UDT562	Output parameter, Extended FB
LTLP_typeGraphManualSetup	UDT564	Structure: Manual movements for setup screens 1 and 2

Copy templates

Symbolic name	Block number	Description
LTLP_GraphManualSetup1Data	DB564	DB for manual selection plus and minus (setup screen 1)
LTLP_GraphManualSetup2Data	DB565	DB for manual selection plus and minus (setup screen 2)

2.6 HMIDeviceDiagnostic

Blocks for device diagnostics of HMI PRO are contained under **HMIDeviceDiagnostic**.

Types

Symbolic name	Block number	Description
LTLP_DeviceDiagnosticAlarm	FB597	Not used
LTLP_SLDiagnosticCommand	FB599	HMI device diagnostics screens

Copy templates

Symbolic name	Block number	Description
LTLP_SlDiagnosticCommandDB	DB599	Instance DB

2.7 HMIEnergyData

Blocks that are required for the energy data screen of HMI PRO are contained under **HMIEnergyData**.

Types

Symbolic name	Block number	Description
LTLP_EnergyConsmptCounter	FB551	Energy consumption data processing for measuring with pulse counter
LTLP_EnergyConsmptMeasure	FB552	Energy consumption data processing for measurement using a measuring instrument (PAC4200)
LTLP_typeHMIEnergyData	UDT521	Structure to display the energy data in the HMI screen

Copy templates

Symbolic name	Block number	Description
LTLP_EnergyConsmptCounterDB	DB551	Instance DB
LTLP_EnergyConsmptMeasureDB	DB552	Instance DB

2.8 MSO

The operating mode blocks for HMI devices and switching over operating modes are contained under **MSO**.

Types

Symbolic name	Block number	Description
LTLP_AxisSelectScreen	FC545	U keys screen: Processing of the signals of the U keys menu on the operator panel
LTLP_ControlPanelMCP	FC541	MCP: Processing key signals and controlling LEDs
LTLP_ControlPanelMPP	FC540	MPP: Processing key signals and controlling lamps and LEDs
LTLP_ControlPanelMPP1500	FC1540	For MPP1500: Processing key signals and controlling LEDs
LTLP_FunctionModeNC	FC549	Selection of the NC functions
LTLP_FunctionModeTransline	FC548	Selection of the TRANSLINE functions
LTLP_HandHeldTerminal2	FC543	HT2: Processing key signals and controlling the LCD display (only for SINUMERIK control)
LTLP_HandHeldTerminal8	FC542	HT8: Processing key signals and controlling LEDs
LTLP_OperatingMode	FC547	Switches over the operating modes taking into account Mode of Safe Operation and Transline function types
LTLP_typeAxisDataFromNC	UDT574	Axis-specific data from the NC to HMI PRO
LTLP_typeAxisDataToNC	UDT575	Axis-specific data from HMI PRO to the NC
LTLP_typeAxisSelectScreen	UDT545	Structure for parameterizing the U key screen
LTLP_typeModeGroupDataFromNC	UDT570	Mode-group-specific data from the NC to HMI PRO
LTLP_typeModeGroupDataToNC	UDT571	Mode-group-specific data from HMI PRO to the NC
LTLP_typeChannelDataFromNC	UDT572	Channel-specific data from NC to HMI PRO

Symbolic name	Block number	Description
LTLP_typeChannelDataToNC	UDT573	Channel-specific data from HMI PRO to NC
LTLP_typeControlPanelMCP	UDT541	Structure for parameterizing an MCP483
LTLP_typeControlPanelMPP	UDT540	Structure for parameterizing an MPP483
LTLP_typeControlPanelMPP1500	UDT544	Structure for parameterizing an MPP1500
LTLP_typeDeviceInterface	UDT546	Structure of the MSO interface
LTLP_typeEnableFucntions	UDT552	Help structure: functions that are always enabled
LTLP_typeHandHeldTerminal2	UDT543	Structure for parameterizing an HT2 (only for SINUMERIK control)
LTLP_typeHandHeldTerminal8	UDT542	Structure for parameterizing an HT8
LTLP_typeHT8InputSignals	UDT557	Help structure: HT8 input signals
LTLP_typeHT8OutputSignals	UDT558	Help structure: HT8 output signals
LTLP_typeMcpFunctions	UDT551	Help structure: MCP functions
LTLP_typeMppFunctions	UDT550	Help structure: MPP functions
LTLP_typeNcHmiProInterface	UDT549	Structure: NC/HMI interface
LTLP_typeOffsetsMcpFunction	UDT553	Help structure: Offset address, MCP functions
LTLP_typeOffsetsMppFunction	UDT552	Help structure: Offset address, MPP functions
LTLP_typeOperatingMode	UDT547	Structure: Modes of Safe Operation/TL function types
LTLP_typeParametrizationMcp Function	UDT555	Help structure: Parameterizing MCP functions
LTLP_typeParametrizationMpp Function	UDT554	Help structure: Parameterizing MPP functions
LTLP_typeParamMPP1500Output	UDT559	Help structure: Parameterizing an output of the MPP1500

Copy templates

Symbolic name	Block number	Description
LTLP_AxisSelectScreenData	DB545	Parameterizing DB U key screen
LTLP_ControlPanelMCPData	DB541	Parameterizing DB MCP
LTLP_ControlPanelMPP1500Data	DB544	Parameterizing DB MPP1500
LTLP_ControlPanelMPPData	DB540	Parameterizing DB MPP
LTLP_DAGControlPanelMPP1500EData	---	Only Daimler AG: Parameterizing DB MPP1500E
LTLP_DeviceInterfaceData	DB546	Parameterizing DB
LTLP_HandHeldTerminal2Data	DB543	Parameterizing DB HT2 (only for SINUMERIK control)
LTLP_HandHeldTerminal8Data	DB542	Parameterizing DB HT8
LTLP_NcHMIProInterfaceData	DB549	Parameterizing DB for screen axis selection, interface between axis selection, HMI PRO, Mode of Safe Operation and the NC control
LTLP_OperatingModeData	DB547	Parameterizing DB Modes of Safe Operation
UPLT_HT2DisplayLines	DB1543	Structure for parameterizing user texts in the HT2 display (only for SINUMERIK control)

2.9 ProductionDataAcquisition

Blocks for PDA functions lifetime overview, workpiece count, cycle times are contained under **ProductionDataAcquisition**.

Types

Symbolic name	Block number	Description
LTLP_CycleTime	FB523	Cycle time acquisition
LTLP_ToolLifeTime	FB521	Lifetime overview: Calculating tool lifetimes
LTLP_WorkpieceCount	FB522	Workpiece count

Copy templates

Symbolic name	Block number	Description
LTLP_CycleTimeDB	DB523	Parameterizing DB cycle time acquisition
LTLP_ToolLifeTimeDB	DB521	Parameterizing DB lifetime overview
LTLP_WorkpieceCountDB	DB522	Parameterizing DB workpiece count
UPTL_ExternalWorkpieceData		DB template for workpiece types>32

■

For notes

3

3 Program structure

PLC program structures when using a SIMATIC S7-1500 controller are subsequently described.

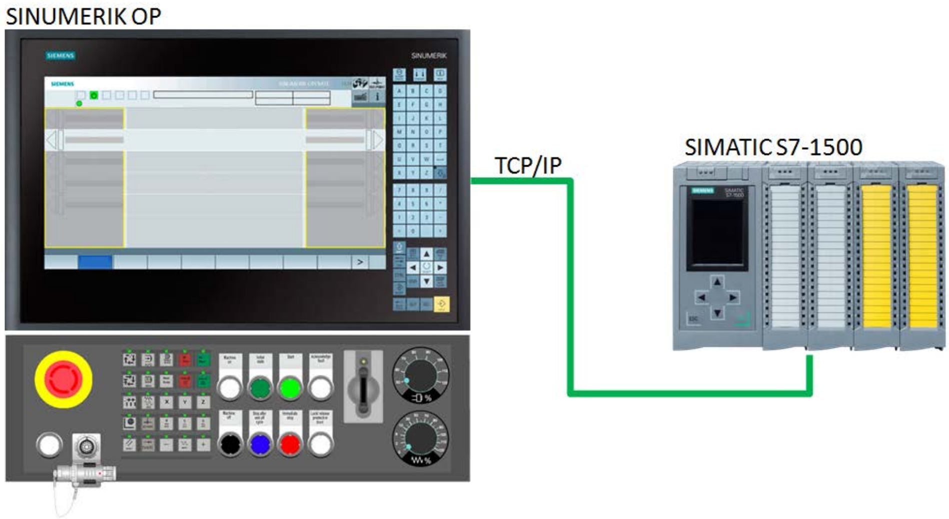


Fig. 3-1: S7-1500 controller with operator panel

The software structure must be adapted to the required conditions. As a rule, unused functions and options are omitted.

Machine manufacturers can also call up their application-specific blocks from the management blocks, resulting in a clear and uniform program structure. If deviations are required in the structure, they must be noted in the PLC documentation.

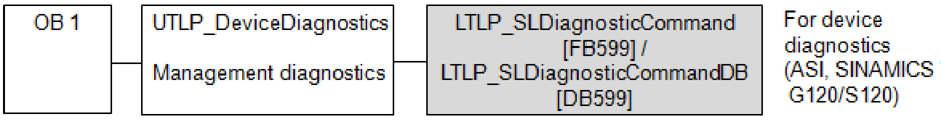
Solutions for Powertrain HMI PRO sl

Interfaces DB **LTLP_HMIProInterfaceData** [DB501] is used for HMI PRO display functions. The user program addresses the DB in the corresponding range in order to implement appropriate display functions. The description of the interface bits can be found in the online help for the HMI PRO CS configuration tool.

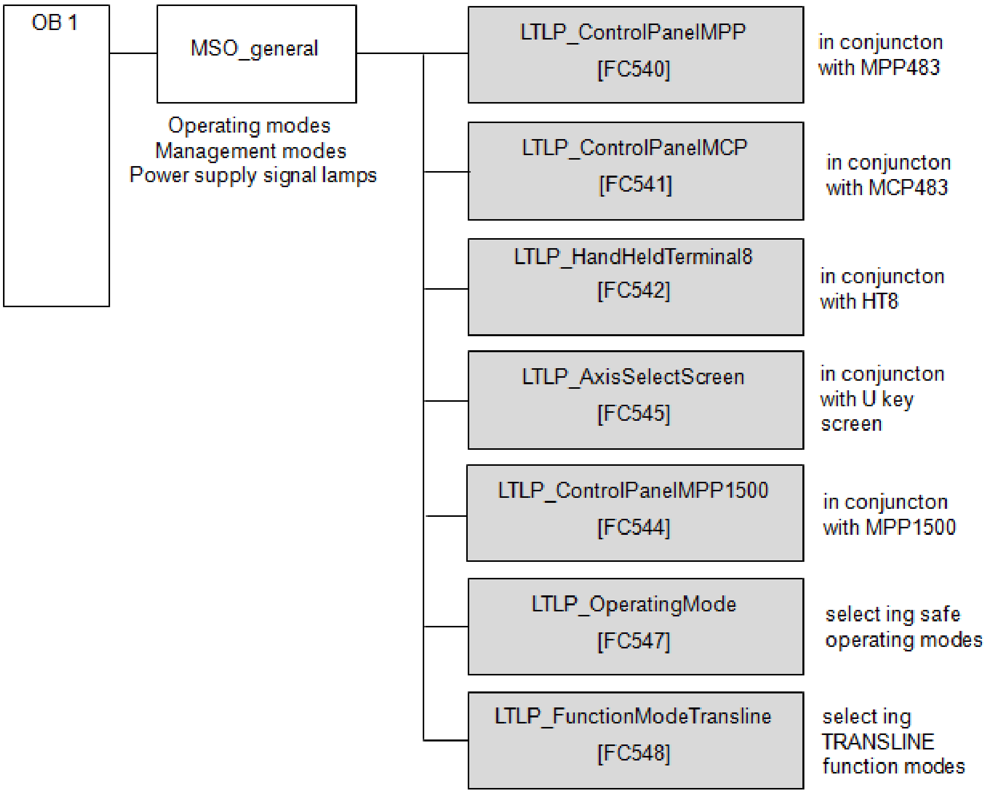
The blocks depicted in the program structure have the following meanings:

- Box with white background: Blocks must be supplemented or created by the user.
- Box with gray background: Standard blocks for TRANSLINE projects.

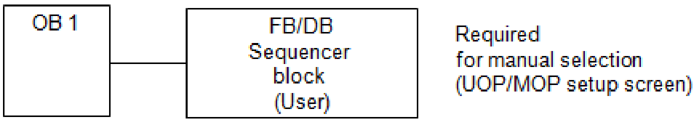
3.1 Device diagnostics


OB 1
UTLP_DeviceDiagnostics
Management diagnostics
LTLP_SLDiagnosticCommand [FB599] / LTLP_SLDiagnosticCommandDB [DB599]
For device diagnostics (ASI, SINAMICS G120/S120)

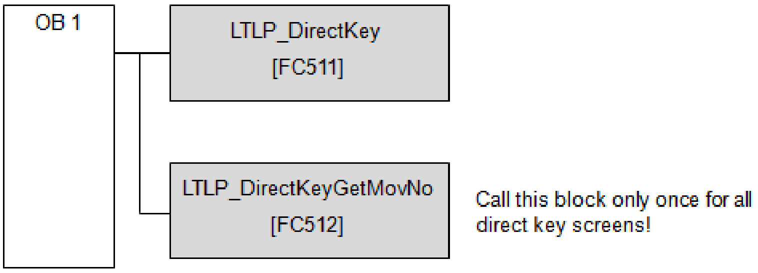
3.2 Modes of Safe Operations/HMI devices



3.3 Programming with GRAPH

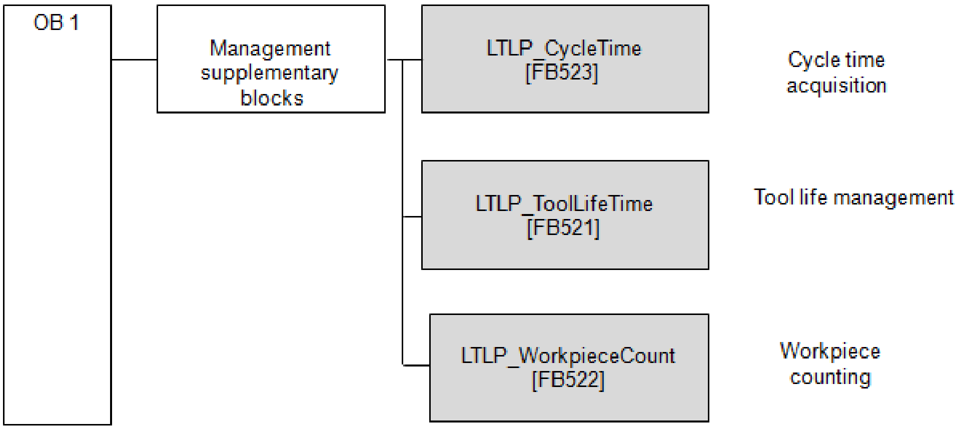


3.4 Direct keys

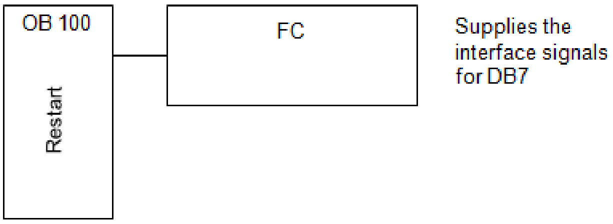


Call this block only once for all direct key screens!

3.5 PDA functions



3.6 Basic program



■

For notes

4

4 Connecting the control elements to the controllers

This chapter includes information on how you connect IPCs (this involves the IPC427 and IPC477 hardware released for SINUMERIK) with HT8, TCU and MPP as well as a SIMATIC S7-1500 in a common network.

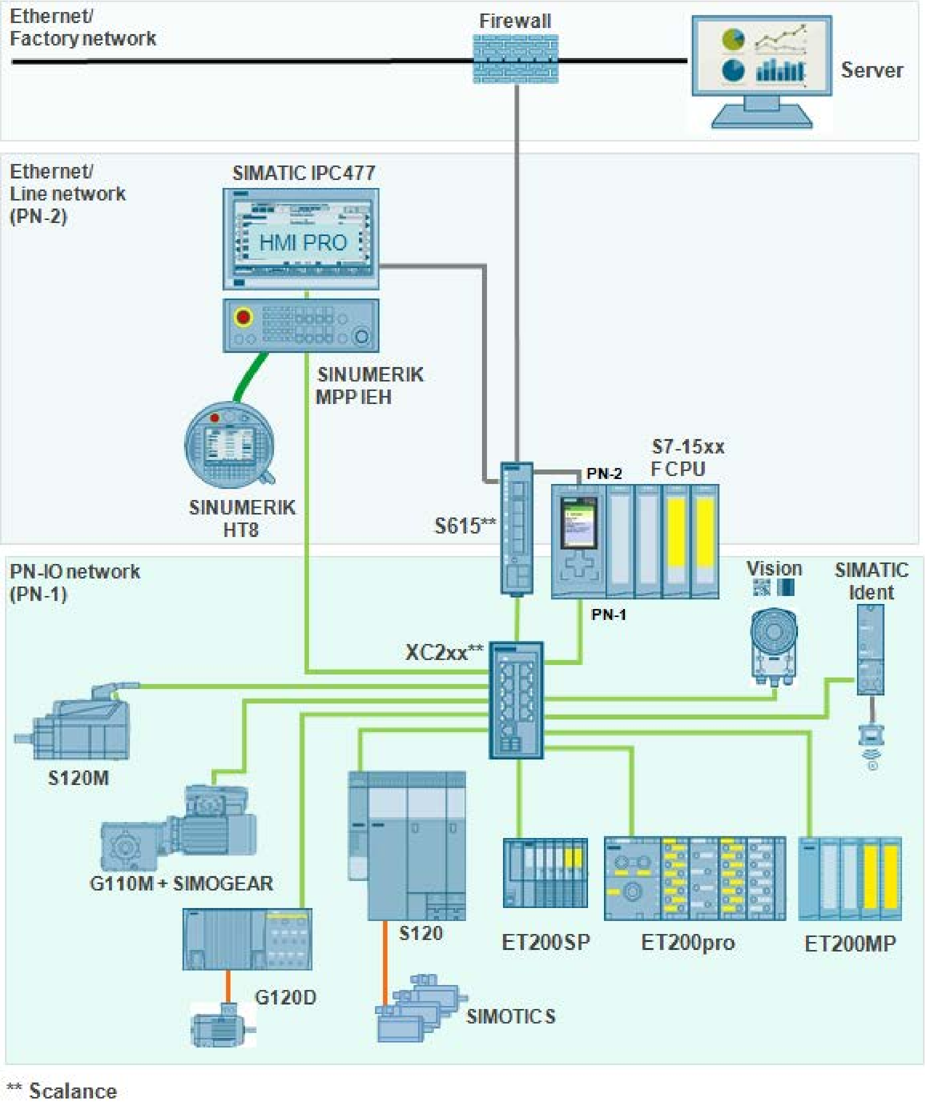


Fig. 4-1: Connecting an IPC and control elements to a SIMATIC S7-1500 controller

The solution line environment mentions 3 network structures:

Ethernet/factory/production network

Connecting the production plants and systems to the IT landscape of an end customer.

Ethernet/system network (PN-2)

The network between machines/stations is known as the Ethernet/plant network; PLC data can be exchanged via this network. In addition, the network is available for other standard TCP communications (e.g. quality data, connection to the master computer, OPC UA). Communication of safety-related and non-safety-related signals is realized via the PROFINET interface of the controller being used. The PLC can be programmed centrally for all machines via this network.

PN-IO network

The PN-IO network connects typical field devices with the controller of the particular machine/station. When using intelligent field devices, by appropriately configuring the S615, it is possible to address field devices from the system network or the factory network.

4.1 Communication levels with PROFINET and Industrial Ethernet

The system network and factory network should be physically separated. Every unit that needs to be visible on the factory network must be directly integrated into this network. This is required, for example, if unit operating data is to be acquired or if ePS Network Services is to be used.

PLC communication between the header-end and the machining units as well as between the individual machining units is via the system network PN **I-Device, T-Send/Recv**.

If the visualization data for transfer lines and flexible lines is transferred over the system network as T:M:N applications, the system network and the PN system network should be connected with one another using unmanaged switches. If visualization is implemented as observation of the individual machines, it makes sense to separate the system network and the PN system network. It is then possible to structure each unit or cell identically; each cell has its own system network. Only the PN system network covers the entire plant.

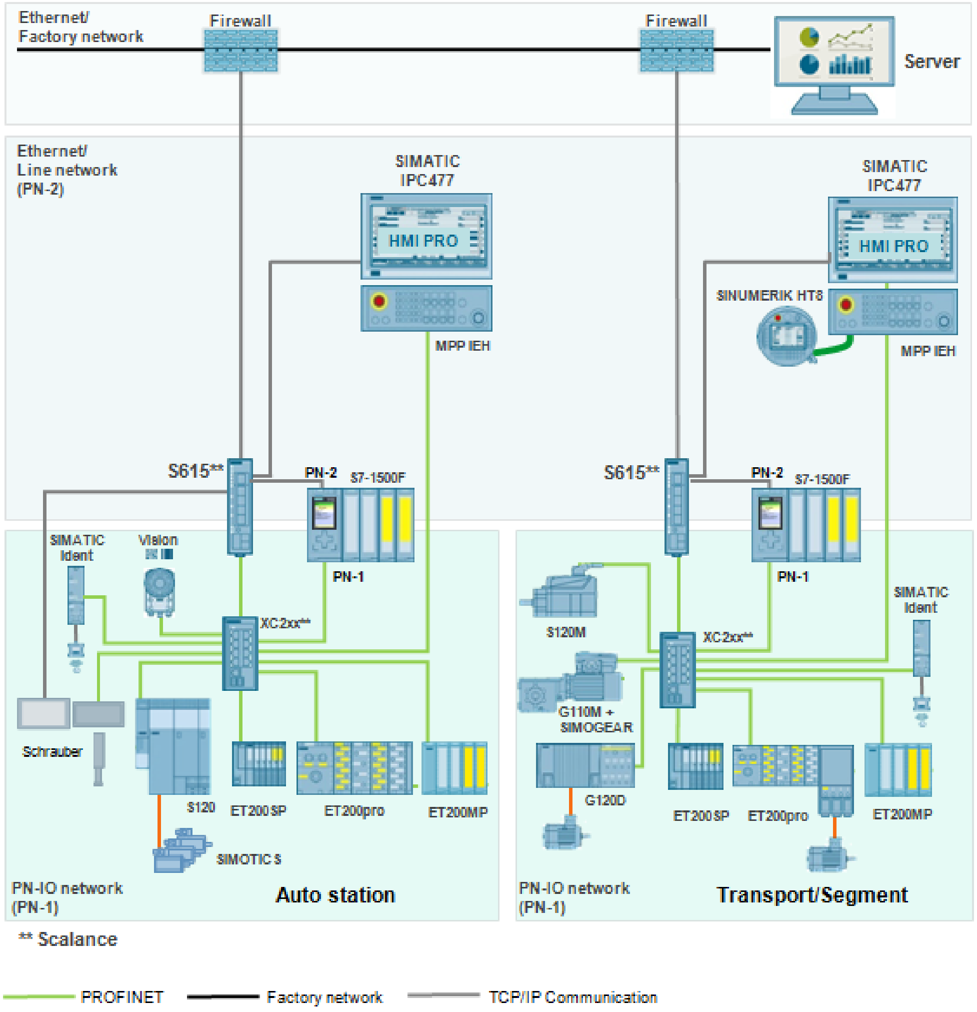


Fig. 4-2: Assembly line, IPC-based (HMI PRO)

4.2 Connecting an IPC with PCU Base and additional control elements to a SIMATIC PLC via the PN interface

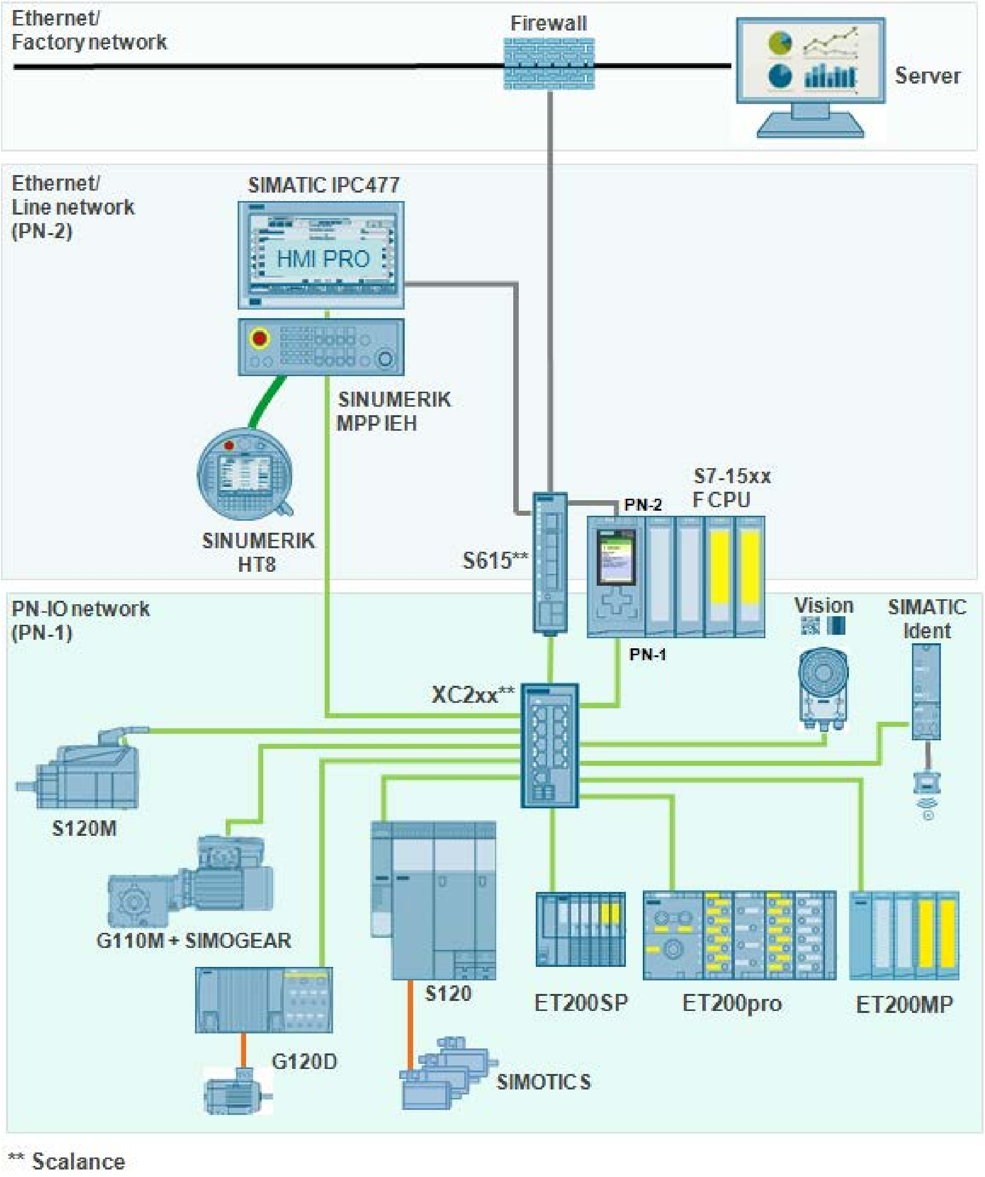


Fig. 4-3: Connecting an IPC and control elements to a SIMATIC PLC via the PN-2 interface

All control elements are configured in the plant/system network and the PN-2 network of the S7-1500.

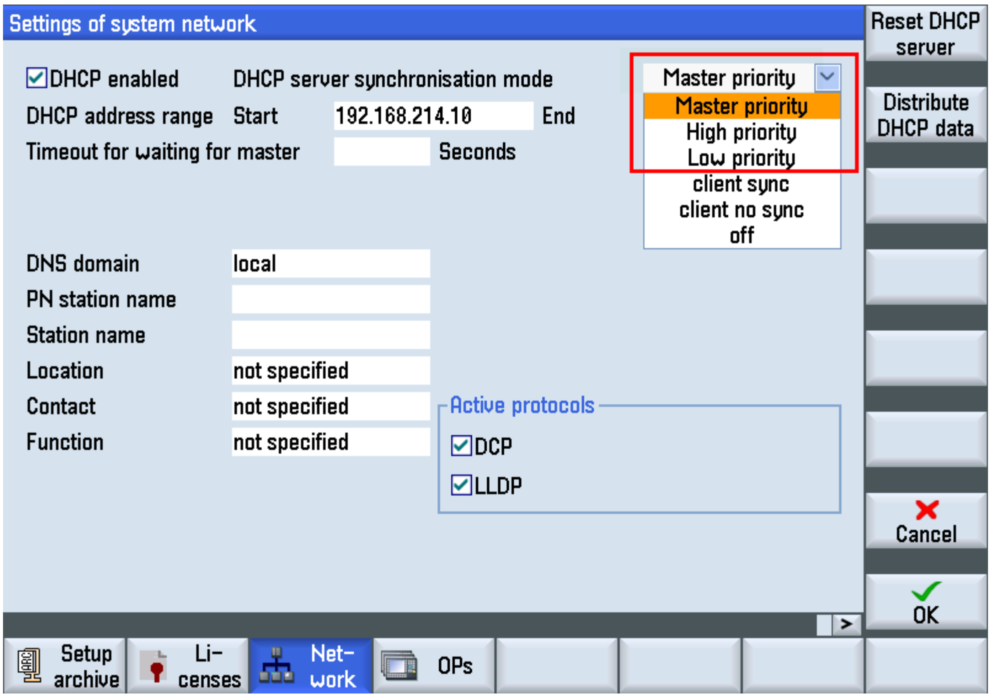
- IPC, SIMATIC S7-1500 are assigned fixed addresses.
- Normally, the MPP is set as PROFINET node, and is therefore supplied with a fixed address via the PN-2 network of the SIMATIC.
- HT8, TCUs - and possibly also the MPP - are connected via the SINUMERIK DHCP server of the IPC. An area should be set for the IP addresses on the IPC under **Commissioning ▸Network ▸System network**.

Input data (e.g. from keys) of the control components is transferred to the PLC, and output data (e.g. LEDs) is transferred from the PLC to the control components. This data is processed by the **LTLP_ControlPanelMPP**, **LTLP_HandHeldTerminal8 …** blocks (see Chapter 6 Modes of Safe Operation).
The memory areas in which input and output data are saved must be defined in the user program.

4.2.1 Procedure when integrating an MPP or HT8 via the Ethernet interface (SINUMERIK DHCP client)

The HT8 **must** be integrated as SINUMERIK DHCP client; MPP IE **can** be integrated as SINUMERIK DHCP client.
The HT8 is connected with the connection box or the MPP IE via Ethernet at the X2P1 interface of the IPC (system network) and at PN-2 of the S7-1500.

When connecting an HT8 or an MPP to a SIMATIC PLC in the operating area **Commissioning ▶ Network ▶ System network,** the DHCP server of the IPC must be set to "Low priority", "High priority" or "Master priority". (You require as a minimum access level 3 in order to change the settings (user).) The settings are accepted using the **Distribute DHCP data**.



Settings of system network

☑ DHCP enabled

DHCP server synchronisation mode: Master priority

DHCP address range: Start 192.168.214.10 End

Timeout for waiting for master: Seconds

DNS domain: local

PN station name:

Station name:

Location: not specified

Contact: not specified

Function: not specified

Active protocols

☑ DCP

☑ LLDP

Fig. 4-4: Setting the DHCP server of the PCU for connecting HT8 to a SIMATIC PLC

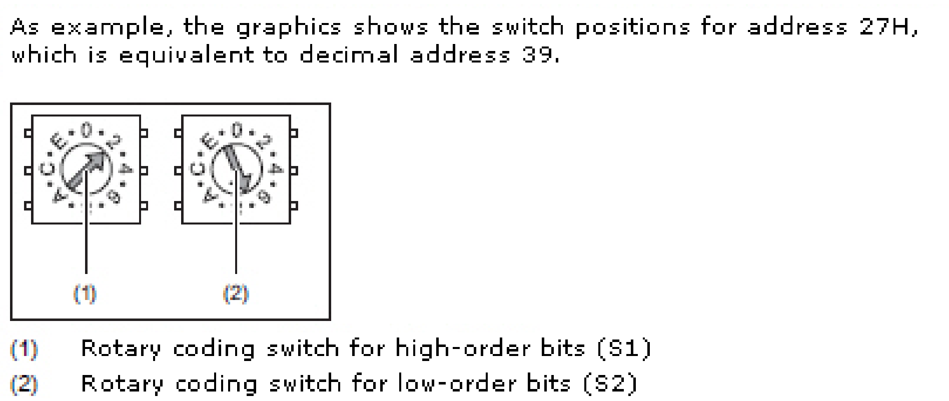
LBP_OPUnitComm [FB25000] handles the actual communication. As a minimum, the block should be called every 40 ms so that the transfer intervals to and from control components can be maintained.

The MPP addresses are set using switch S2.

Table 4-1: Setting the addresses of the MPP for IE addressing

1	2	3	4	5	6	7	8	9	10	Significance
on	on	on	on	on	on	on	on	off	off	MCP address 255
										…
on	off	off	off	off	off	off	off	off	off	MCP address 1

The address for the HT8 must be set at the two mini rotary switches of the HT8 connection box (addressing range 1..254) – and transferred when first inserting the HT8. The default setting is: address 14, i.e. S1=0 and S2=E.



As example, the graphics shows the switch positions for address 27H, which is equivalent to decimal address 39.

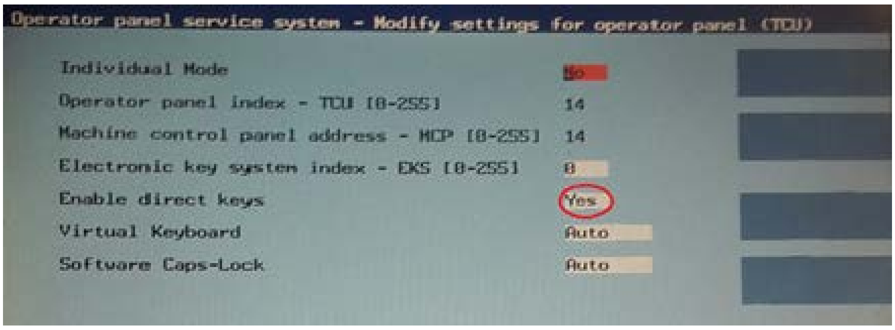
(1) Rotary coding switch for high-order bits (S1)

(2) Rotary coding switch for low-order bits (S2)

Fig. 4-5: Switches S1 and S2 at the HT8

General sequence

- Connect the HT8 to a connection module
- TCU settings
  - Leave HT8 individual mode at NO
  - Accept the TCU and MCP indices
  - Leave the EKS index
  - EnableDirect keys = YES



Operator panel service system - Modify settings for operator panel (TCU)

Individual Mode	No
Operator panel index - TCU [0-255]	14
Machine control panel address - MCP [0-255]	14
Electronic key system index - EKS [0-255]	0
Enable direct keys	Yes
Virtual Keyboard	Auto
Software Caps-Lock	Auto

Fig. 4-6: Making the TCU settings at the HT8

- No more than four HT8 handheld units can be connected or, in mixed operation, no more than 4 nodes, consisting of TCU and HT8

4.2.2 Procedure when integrating an MPP IE as PROFINET I/O node

Switch S2 (address range 1..127) is used to set the addresses of the MPP. If you are using the Ethernet version (MPP IE), this operator panel is connected as PROFINET IO module to the IPC via Ethernet interface 2 (system network). **No LBP_OpUnitComm** [FB25000] should be used.

Table 4-2: Setting the address at the MPP, switch S2

1-8	9	10	Significance
See the following tables for the significance of the switch positions	on	on	PROFINET
	off	off	IE (default)

The switch positions 9 and 10 guarantee the PROFINET functionality of the module and must always be switched **on**.

DCP mode

No default device name is available in this mode. The device name must be set by an "initialization procedure" and remains saved on the I/O module. It is only deleted after a reset to factory settings, e.g. with STEP 7.

Table 4-3: Addressing the devices using the "logical" device names

1	2	3	4	5	6	7	8	9	10	Significance
on	on	on	on	on	on	on	on	on	on	DCP mode/PROFINET

Default device names at the MPP

The device names are structured as follows: <mcp-pn>" + switch code for switches S2-1 to S2-8, e.g. mcp-pn7 for S2-1 to S2-3 **on** and S2-4 to S2-8 **off**.

Table 4-4: Setting the device name at the MPP

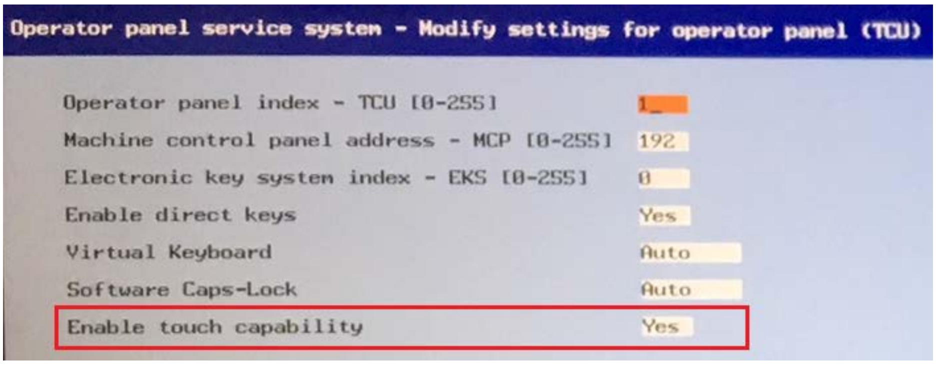
1	2	3	4	5	6	7	8	9	10	Significance
on	on	on	off	off	off	off	off	on	on	Default device name: mcp-pn7 (PROFINET)

4.2.3 Procedure when integrating an OP015 Black or TCU + OP

A direct key module is no longer required for the OP015 Black or TCU + OP

TCU settings

- Set the TCU and MCP indices
- Leave the EKS index -> within the EKS application, the EKS index should be entered onto the IPC, so that a data backup is available within HMI PRO.
- Set EnableDirect Keys = YES
- Enable touch capability = Yes
  so that touch operation at the OP015 Black is activated


Operator panel service system - Modify settings for operator panel (TCU)

Operator panel index - TCU [0-255]	1_
Machine control panel address - MCP [0-255]	192
Electronic key system index - EKS [0-255]	0
Enable direct keys	Yes
Virtual Keyboard	Auto
Software Caps-Lock	Auto
Enable touch capability	Yes

Fig. 4-7: TCU settings: Enable touch capability = Yes

4.2.4 Communication of the control elements with the S7-1500: LBP_ConfigData [DB7] and LBP_OPUnitComm [FB25000]

General settings

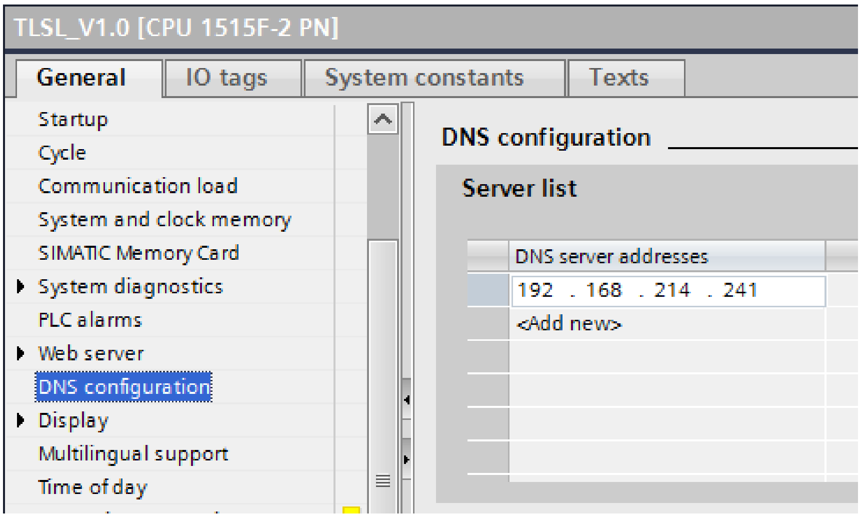
The standard blocks required for HMI PRO running under SINUMERIK Operate are saved in the TRANSLINE library under Copy templates in directories **840evo PLC BP** and **840evo PLC BP data types**.

Copy the following copy templates into your TIA project:

- **840evo PLC BP** in the **program blocks** directory
- **840evo PLC BP data types** in the PLC data types directory.

DNS configuration

The logical device names of the control components must be resolved internally into the appropriate IP addresses. To do this, the PLC requires data relating to the DNS server The IP address of the DNS server must be entered in the **device configuration** in the **properties** of the CPU ▶**General** ▶**DNS configuration**. The IP address of the plant/system interface of the IPC (X2P1) is entered as DNS server.



DNS configuration

Server list

DNS server addresses
192 . 168 . 214 . 241
<Add new>

Fig. 4-8: Setting the DNS configuration for the PLC

The PROFINET interface PN-2 is used as PLC interface.

Connection identifier and UDP ports

If, on the PLC, no different communication functions are used, then ideally you should use the suggested values. These values should be accepted without any adaptation.

Table 4-5: Connection identifier and UDP ports - default values

Operator panel (HMI device)	**Connection identifier**
MCP1ConnectionId	1001
MCP2ConnectionId	1002
I HTConnectionId	1003
Op1KeyConnectionId	1004
Op2KeyConnectionId	1005
IdentConnectionId	1006
Operator panel (HMI device)	**UDP port**
MCP1LocalUdpPort	16001
MCP2LocalUdpPort	16002
HTLocalUdpPort	16003
Op1KeyLocalUdpPort	16004
Op2KeyLocalUdpPort	16005

Note

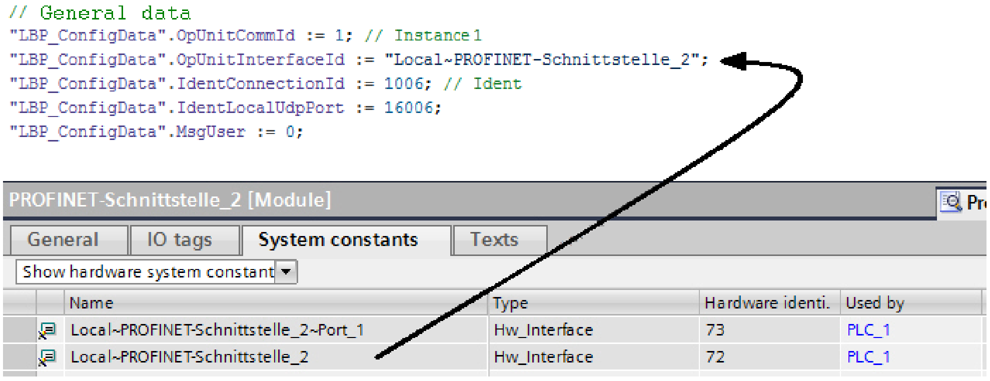
It is not permissible to utilize the connection identifiers being used for other communication connections in the PLC project.
It is not permissible to utilize the local UDP ports being used for other UDP connections at this network interface in the PLC project.

General data in the LBP_ConfigData [DB7]

LBP_ConfigData must be parameterized in the OB1.

Table 4-6: General data in the DB7

Parameter	**Description**
OpUnitCommId	Number of the instance, this must be unique
OpUnitInterfaceID	Hardware identifier of the interface, which is connected with the PCU/IPC (for example: hardware identifier of the CP interface)
IdentConnectionId	= 1006, ident of the connection
IdentLocalUdpPort	= 16006, connection port



```
// General data
"LBP_ConfigData".OpUnitCommId := 1; // Instance 1
"LBP_ConfigData".OpUnitInterfaceId := "Local~PROFINET-Schnittstelle_2";
"LBP_ConfigData".IdentConnectionId := 1006; // Ident
"LBP_ConfigData".IdentLocalUdpPort := 16006;
"LBP_ConfigData".MsgUser := 0;
```

PROFINET-Schnittstelle_2 [Module]
Pr
General
IO tags
System constants
Texts
Show hardware system constant

Name	Type	Hardware identi.	Used by
Local~PROFINET-Schnittstelle_2~Port_1	Hw_Interface	73	PLC_1
Local~PROFINET-Schnittstelle_2	Hw_Interface	72	PLC_1

Fig. 4-9: Parameterization example for the general data in the DB7

Note

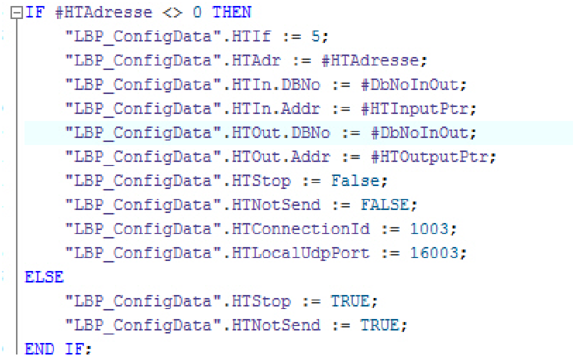
It is only permissible to change the data listed above when the controller restarts.

Integrating an HT8

Table 4-7: Parameters for the HT8 connection in the DB7

Parameter	Description
HTIf	= 5 (specification, if HT8 available)
HTAddr	TCU index of the HT8 connection point Numerical part of the logical name. The logical name is set via switches at the MPP or connection box (mini rotary switches).
HTIn	Memory area, which contains the input data, sent from the HT8 to the controller Input address of the HT8 For a data block, the following must be entered: HTIn.DBNo number of the DB HTIn.Addr pointer P#DBX0.0 as constant, otherwise as P#E0.0
HTOut	Memory area, which contains the output data, sent from the controller to the HT8 Output address of the HT8 For a data block, the following must be entered: HTOut.DBNo number of the DB HTOut.Addr pointer P#DBX0.0 as constant, otherwise as P#A0.0
HTConnectionId	= 1003, connection identifier of the HT8
HTLocalUdpPort	= 16003, connection port

Supply of the **LBP_ConfigData** [DB7] in the OB100:



```
IF #HTAdresse <> 0 THEN
    "LBP_ConfigData".HTIf := 5;
    "LBP_ConfigData".HTAdr := #HTAdresse;
    "LBP_ConfigData".HTIn.DBNo := #DbNoInOut;
    "LBP_ConfigData".HTIn.Addr := #HTInputPtr;
    "LBP_ConfigData".HTOut.DBNo := #DbNoInOut;
    "LBP_ConfigData".HTOut.Addr := #HTOutputPtr;
    "LBP_ConfigData".HTStop := False;
    "LBP_ConfigData".HTNotSend := FALSE;
    "LBP_ConfigData".HTConnectionId := 1003;
    "LBP_ConfigData".HTLocalUdpPort := 16003;
ELSE
    "LBP_ConfigData".HTStop := TRUE;
    "LBP_ConfigData".HTNotSend := TRUE;
END_IF;
```

Fig. 4-10: OB100: HT8 machine control panel

Connecting HT8 direct keys or OP 015 Black direct keys

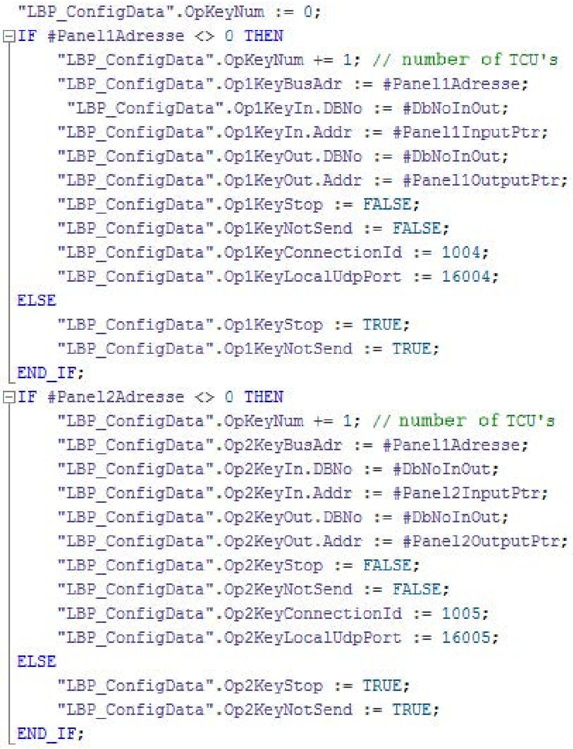
Communication is realized via the **LBP_OPUnitComm** [FB25000]. The parameterization for coupling the direct keys via Ethernet – this involves for example, the HT8 or OP 015 Black – is realized via the parameters in the DB7.

Table 4-8: Parameters for coupling direct keys in the DB7

Parameter	Description
OpKeyNum	Number of panels being used with direct keys
Op1KeyBusAdr/ Op2KeyBusAdr	TCU index: OP: Index, which you specify when newly installing the OP or under **Commissioning ▶OPs**. HT8: Numerical part of the logical name. The logical name is set via switches at the MPP or connection box (mini rotary switches).
OP1KeyConnectionId/ OP2KeyConnectionId	Unique connection identifiers that are used by **LBP_OPUnitComm** [FB25000] to communicate with the control components. Connection identifier to the corresponding device, see Table 4-5
OP1KeyLocalUdpPort/ OP2KeyLocalUdpPort	Unique UDP ports that are used by **LBP_OPUnitComm** [FB25000] to communicate with the control components. UDP port of the connection, see Table 4-5
Op1KeyIn/ OP2KeyIn	Memory area, which contains the input data, sent from the operator component to the controller Input address of the direct keys of the panel/HT8 For a data block, the following must be entered: OP1KeyIn.DBNo number of the DB OP1KeyIn.Addr pointer P#DBX0.0 as constant, otherwise as P#E0.0
OP1KeyOut/	Memory area, which contains the output data, sent

Parameter	Description
OP2KeyOut	from the controller to the operator component Output address of the direct keys of the panel/HT8 For a data block, the following must be entered: OP1KeyOut.DBNo number of the DB OP1KeyOut.Addr pointer P#DBX0.0 as constant, otherwise P#A0.0

Supply of the **LBP_ConfigData** [DB7] in the OB100:



```
 "LBP_ConfigData".OpKeyNum := 0;
IF #Panel1Adresse <> 0 THEN
    "LBP_ConfigData".OpKeyNum += 1; // number of TCU's
    "LBP_ConfigData".Op1KeyBusAdr := #Panel1Adresse;
     "LBP_ConfigData".Op1KeyIn.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op1KeyIn.Addr := #Panel1InputPtr;
    "LBP_ConfigData".Op1KeyOut.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op1KeyOut.Addr := #Panel1OutputPtr;
    "LBP_ConfigData".Op1KeyStop := FALSE;
    "LBP_ConfigData".Op1KeyNotSend := FALSE;
    "LBP_ConfigData".Op1KeyConnectionId := 1004;
    "LBP_ConfigData".Op1KeyLocalUdpPort := 16004;
ELSE
    "LBP_ConfigData".Op1KeyStop := TRUE;
    "LBP_ConfigData".Op1KeyNotSend := TRUE;
END_IF;
IF #Panel2Adresse <> 0 THEN
    "LBP_ConfigData".OpKeyNum += 1; // number of TCU's
    "LBP_ConfigData".Op2KeyBusAdr := #Panel1Adresse;
    "LBP_ConfigData".Op2KeyIn.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op2KeyIn.Addr := #Panel2InputPtr;
    "LBP_ConfigData".Op2KeyOut.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op2KeyOut.Addr := #Panel2OutputPtr;
    "LBP_ConfigData".Op2KeyStop := FALSE;
    "LBP_ConfigData".Op2KeyNotSend := FALSE;
    "LBP_ConfigData".Op2KeyConnectionId := 1005;
    "LBP_ConfigData".Op2KeyLocalUdpPort := 16005;
ELSE
    "LBP_ConfigData".Op2KeyStop := TRUE;
    "LBP_ConfigData".Op2KeyNotSend := TRUE;
END_IF;
```

Fig. 4-11: OB100: HT8 or OP 015 Black direct keys

Communication sequence

Communication to the machine control panels (MCP, MPP), handheld terminals (HT) and direct key modules (OpKey) is realized via block **LBP_OpUnitComm** [FB25000]. This supports simultaneous operation of up to two machine control panels, one handheld terminal and two direct key modules via Industrial Ethernet (IE).
Parameterization of the components is performed, as shown above, in the OB100. Dedicated parameter sets exist for each control component. The user must define pointers for the input and output data in these parameter sets.

4.2.5 Switchover between machine control panel, handheld terminal and direct keys

Control signals

Communication to the individual components can be stopped using parameters **MCP1Stop**, **MCP2Stop**, **HTStop**, **Op1KeyStop** and **Op2KeyStop** . You can stop or activate communication in an active cycle.

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 parameters **MCP1Stop**, **MCP2Stop**, **HTStop**, **Op1KeyStop** and **Op2KeyStop** are set, then alarms are suppressed or cleared.

Switching over 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. Stop the communication of the control component to be disconnected:
   Parameter **MCP1Stop**, **MCP2Stop**, **HTStop**, **Op1KeyStop** or **Op2KeyStop** = TRUE
2. Communication is stopped, if in DB10 the following applies:
   –MCP: **E_MCP1ready** or **E_MCP2ready** = FALSE
   –HT: **E_HTready** = FALSE
   –OpKey: **E_Op1KeyReady** or **E_Op2KeyReady** = FALSE
3. Change the bus address in DB7:
   –MCP: **MCP1BusAdr** or **MCP2BusAdr** = <*bus address of the new MCP*>
   –HT: **HTAddr** = <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 pointe 3):
   Parameter **MCP1Stop**, **MCP2Stop**, **HTStop**, **Op1KeyStop** or **Op2KeyStop** = FALSE
5. The communication with the new control component is active if, in DB10 the following applies:
   –MCP: **E_MCP1ready** or **E_MCP2ready** = TRUE
   –HT: **E_HTready** = TRUE
   –OpKey: **E_Op1KeyReady** or **E_Op2KeyReady** = TRUE

Example (SCL)

```
// Change the configuration for MCP 1, to do this, stop MCP 1.
LBP_ConfigData.MCP1Stop := TRUE;
//Wait at least one cycle, set new parameter and start MCP 1.
LBP_ConfigData.MCP1BusAdr := 192;
LBP_ConfigData.MCP1Stop := FALSE;
// Check as to whether the connection is re-established.
IF LBP_NC.E_MCP1ready THEN
// Connection is re-established
```

LBP_OpUnitComm [FB25000]

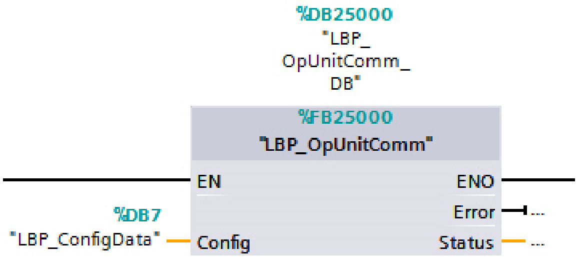
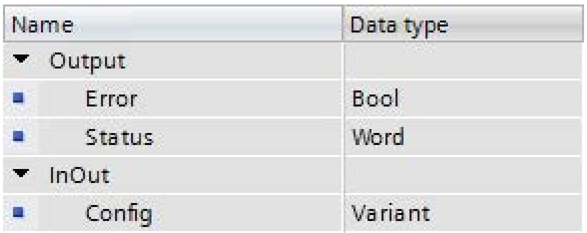


Fig. 4-12: LBP_OpUnitComm [FB25000]



Name	Data type
Output	
Error	Bool
Status	Word
InOut	
Config	Variant

Fig. 4-13: LBP_OPUnitComm parameter

Application

This block is required for communication with the SINUMERIK operator panels (HMI devices).

Description of output parameters

Table 4-9: Output-Parameter LBP_OPUnitComm

Parameter	Data type	Description
Error	Bool	An error has occurred
Status	Word	1 The variable at parameter Config is not a LBP_typeConfigData type 2 A connection-specific error has occurred, see the status of the connections in the configuration block at parameter Config, generally DB7 LBP_ConfigData connection-specific error 10 Error when opening or establishing the communication connection 20 Error when sending the identify request to the operator component 21 Timeout when receiving the identify response from the operator component 22 The operator components does not have an IPv4 address **30** The parameterization data for these control components was not able to be determined 31 The parameterization data is not valid 32 Error when sending the parameterization request to the operator component 33 The parameterization has been rejected by the operator component 34 Timeout when receiving the parameterization response from the operator component 40 Timeout when receiving the data 41 The pointer for the input data from the operator component is invalid 42 The pointer for the output data to the control component is invalid 43 Error when storing the input data from the operator component 44 Error when copying the output data to the operator component 50 Error when disconnecting or closing the connection

Description of InOut parameters

Table 4-10: InOut parameter LBP_OPUnitComm

Parameter	Data type	Description
Config	Variant	LBP_ConfigData DB7

■

For notes

5

5 Relationship between the axis selection screen and HMI PRO

5.1 Axis selection screen in HMI PRO

When using blocks for the Mode of Safe Operation, the **Axis selection** screen must always be integrated into HMI PRO. For applications with HT8 or when using a PCU/IPC without MPP, the screen can be used for selecting TRANSLINE functions.
When using an HT8 or a PCU/IPC without MPP, then the softkeys for the TRANSLINE functions (AUT, SM, SSM etc.) are displayed.
In HMI PRO CS, under **Configuration for SIMATIC PLC,** the setting **HT8: Display U key menu only** must be selected. Further, the labeling of the softkeys on the U keys screen can be changed.

Axis selection screen call on an HT8 or on an IPC

The **Axis selection** screen is opened as follows:

- HT8: With the U key
- IPC: with <Ctrl> + <U>
- IPC with MPP1500: with <Ctrl> + <U> or
  Tip/click on the active area to open the system menu, followed by softkey U key menu
  (Requirement: Configuration in HMI PRO CS under **Default settings ▸ Operator panel ▸ HMI PRO header** not hidden)

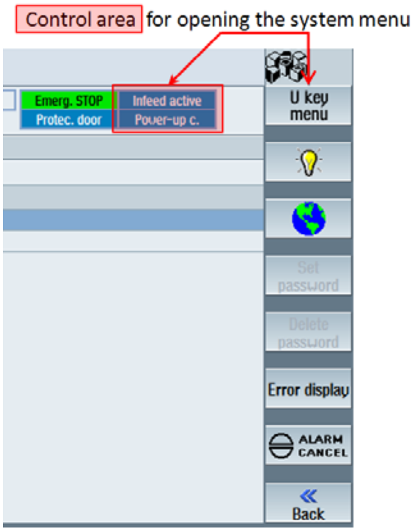
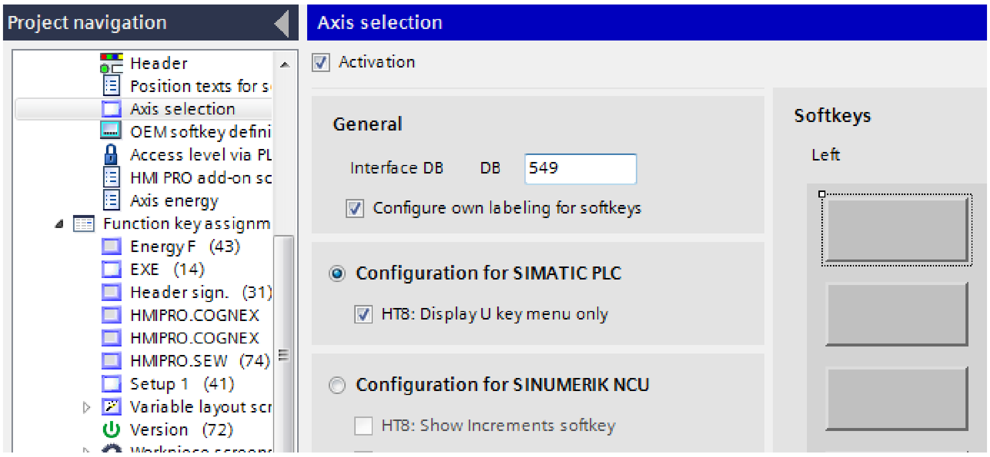


Fig. 5-1: Axis selection call on an IPC with MPP1500

5.2 Configuring in HMI PRO CS when connected to a PLC controller

When connected to a PLC controller, the **axis selection screen** can be used to select TRANSLINE functions. In the following, this screen is called **U keys screen**.



Axis selection

☑ Activation

General

Interface DB DB 549

☑ Configure own labeling for softkeys

● Configuration for SIMATIC PLC

☑ HT8: Display U key menu only

○ Configuration for SINUMERIK NCU

□ HT8: Show Increments softkey

Fig. 5-2: Configuring the axis selection in HMI PRO CS (PLC connection)

Activating

Activates the axis selection screen in HMI PRO RT → must be activated

General information

Entering the data block interfaces for axis selection. This is the **LTLP_NcHmiProInterfaceData** [DB549] as default setting.

Configuring your own softkey inscription → optional
For HT8 – or when using the axis selection screen without MPP – allows users to label U keys using their own text. The softkeys used for labeling are displayed. If this function is used, then **LTLP_HandHeldTerminal8** or **LTLP_AxisSelectScreen** must be appropriately adapted.

Configuration for SIMATIC PLC

HT8: display U key menu only → must be activated
Deactivates access to NCU data. This means that in the axis selection screen, the TRANSLINE functions (U keys) can be selected.

■

6

6 Modes of Safe Operations

TRANSLINE does not support any "safe blocks". The blocks for acquiring data from operator panels MPP, HT8 and U keys screen as well as the signal-processing blocks **LTLP_OperatingMode** and **LTLP_FunctionModeTransline** support OEMs in applying Modes of Safe Operations.

Block definitions

The operating mode blocks provide a similar interface for each operator panel for the following keys and LEDs: MPP, axis selection keys and LEDs and keys/LEDs that OEMs can freely use.
There is an FC and parameterization data block for each operator panel.
The shared interface is the **LTLP_DeviceInterfaceData** [DB546].

The assignment of the keys to TL and NC functions is defined in each case for the associated parameterization data block.

Operating mode blocks

The operating mode blocks are included in the TRANSLINE library V14 under **Types ▸MSO**.

• LTLP_ControlPanelMPP	→ for the MPP483IE
• LTLP_ControlPanelMCP	→ for the MCP483
• LTLP_ControlPanelMPP1500	→ for the MPP1500
• LTLP_HandHeldTerminal8	→ for the HT8
• LTLP_AxisSelectionScreen	→visualization of function types
• LTLP_OperatingMode	→ for selecting operating modes
• LTLP_FunctionModeTransline	→ for selecting TRANSLINE-function types

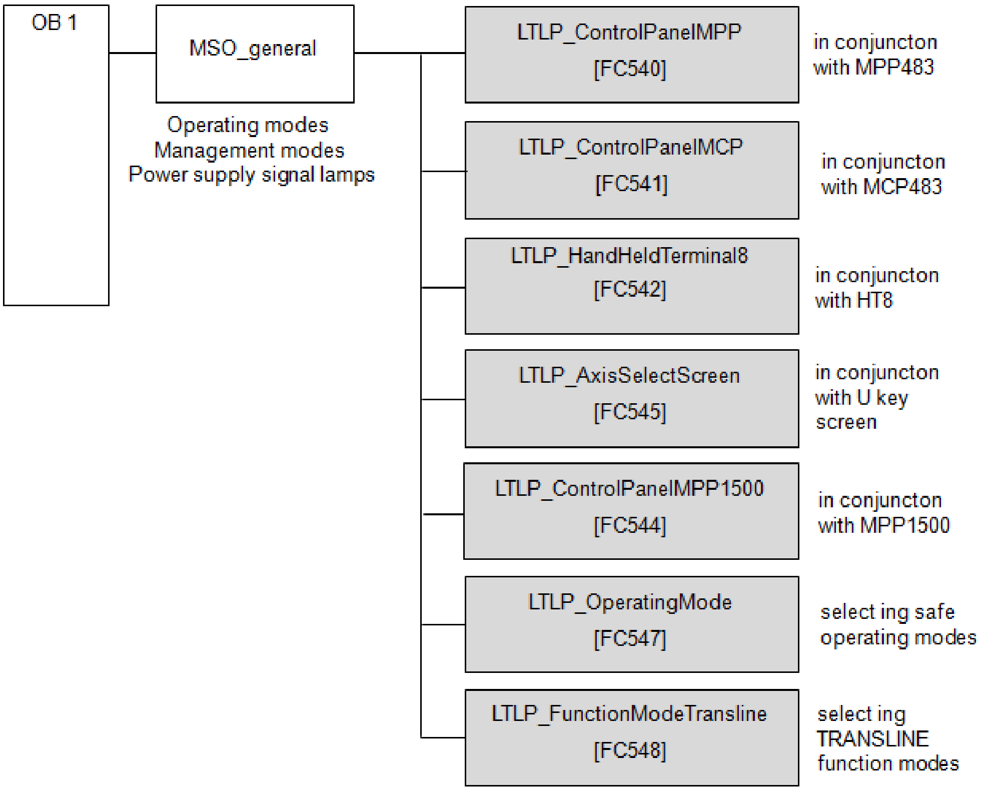


Fig. 6-1: Program structure when using a MPP, MCP, HT8 or the U keys screen with a SIMATIC PLC

6.1 MPP483 - LTLP_ControlPanelMPP [FC540]

This block is required when using an MPP483 as machine control panel.
When creating the **RESTART_PLC** signal, the **LTLP_ControlPanelMPP** reads the assignment of the keys once from the **LTLP_ControlPanelMPPData** [DB540], calculates the new addresses and enters them in data area **inputsAssigned** or **outputsAssigned**.
The FC transfers all key signals to the **LTLP_DeviceInterfaceData** [DB546], area **inputs** as long as at input parameter **ENABLE**=TRUE. It is possible to forward selected keys to the **LTLP_DeviceInterfaceData** even when there is no global enable (ENABLE = FALSE). To do this, in the **LTLP_ControlPanelMPP,** area **alwaysEnabled** enable the appropriate signal.
The lamps and LEDs are always read from the **LTLP_DeviceInterfaceData, outputsAssigned** area and then displayed.
With the input parameter **LAMPTEST** = TRUE, all lamps and LEDs on the MPP are activated. This is also realized if, for a locked MPP, a key is pressed. Keys which were enabled in the **alwaysEnabled** area, are the exception.

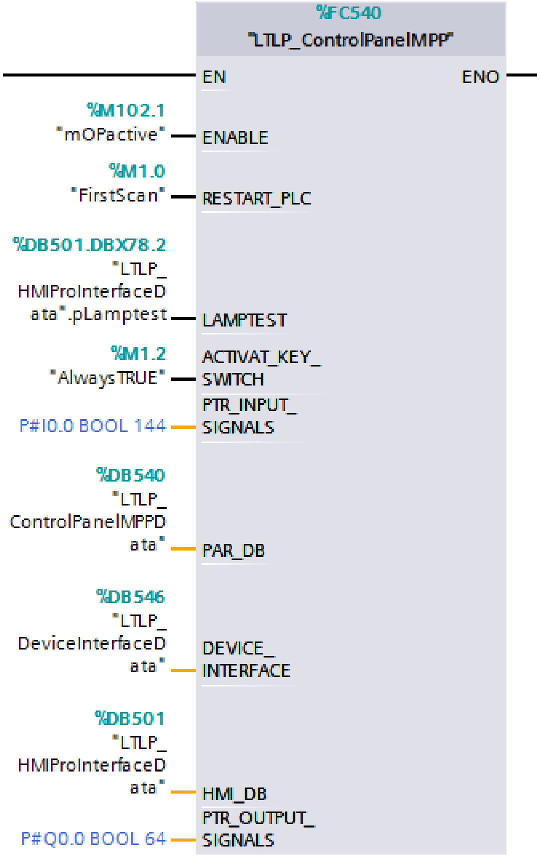
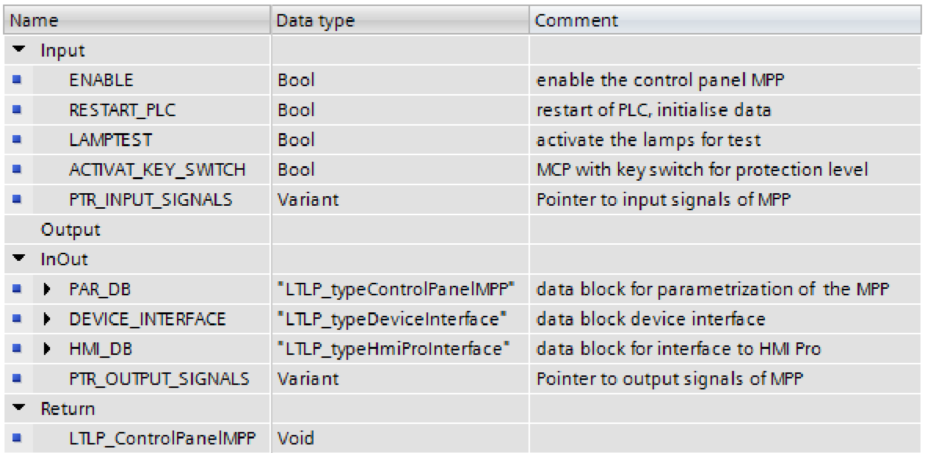


Fig. 6-2: LTLP_ControlPanelMPP [FC540]



Name	Data type	Comment
▼ Input		
■ ENABLE	Bool	enable the control panel MPP
■ RESTART_PLC	Bool	restart of PLC, initialise data
■ LAMPTEST	Bool	activate the lamps for test
■ ACTIVAT_KEY_SWITCH	Bool	MCP with key switch for protection level
■ PTR_INPUT_SIGNALS	Variant	Pointer to input signals of MPP
Output		
▼ InOut		
■ ▶ PAR_DB	"LTLP_typeControlPanelMPP"	data block for parametrization of the MPP
■ ▶ DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block device interface
■ ▶ HMI_DB	"LTLP_typeHmiProInterface"	data block for interface to HMI Pro
■ PTR_OUTPUT_SIGNALS	Variant	Pointer to output signals of MPP
▼ Return		
■ LTLP_ControlPanelMPP	Void	

Fig. 6-3: LTLP_ControlPanelMPP parameter

Description of input parameters

Table 6-1: Input parameter LTLP_ControlPanelMPP

Parameter	Data type	Description
ENABLE	Bool	Global release for reading in the function and key signals of the MPP483. This must be set to TRUE if this is the active operating panel.
RESTART_PLC	Bool	First block cycle following a CPU restart. This must be done after every reparameterization of the key interface in **LTLP_ControlPanelMPPData** [DB540]. According to the TRANSLINE standard, the **FirstScan** [M1.0] flag is specified here.
LAMPTEST	Bool	Lamp test, all outputs (lamps and LEDs) are controlled with this signal. This is the **LTLP_HMIProInterfaceData.pLamptest** as default setting.
ACTIVAT_KEY_SWITCH	Bool	TRUE for MPP with protection level key-operated switch
PTR_INPUT_SIGNALS	Variant	Pointer to the input signals of the MPP483.

Description of InOut parameters

Table 6-2: InOut parameter LTLP_ControlPanelMPP

Parameter	Data type	Description
PAR_DB	LTLP_typeControl PanelMPP	Data block for parameterizing the pushbuttons and function keys on the MPP. This is the **LTLP_ControlPanelMPPData** [DB540] as default setting.
DEVICE_ INTERFACE	LTLL_typeDevice Interface	The interface between the operator panels and the blocks for acquiring the operating modes, TRANSLINE functions and NC functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
PTR_OUTPUT_ SIGNALS	Variant	Pointer to the output signals of the MPP483.

Description of the return value

No value is returned.

Dependencies

none

LTLP_ControlPanelMPPData [DB540]: Parameterization of the keys of the MPP

The keys, lamps and LEDs are assigned via the **LTLP_ControlPanelMPPData**. Function keys are identified with F, pushbuttons with S, LEDs with HF and lamps with HS.

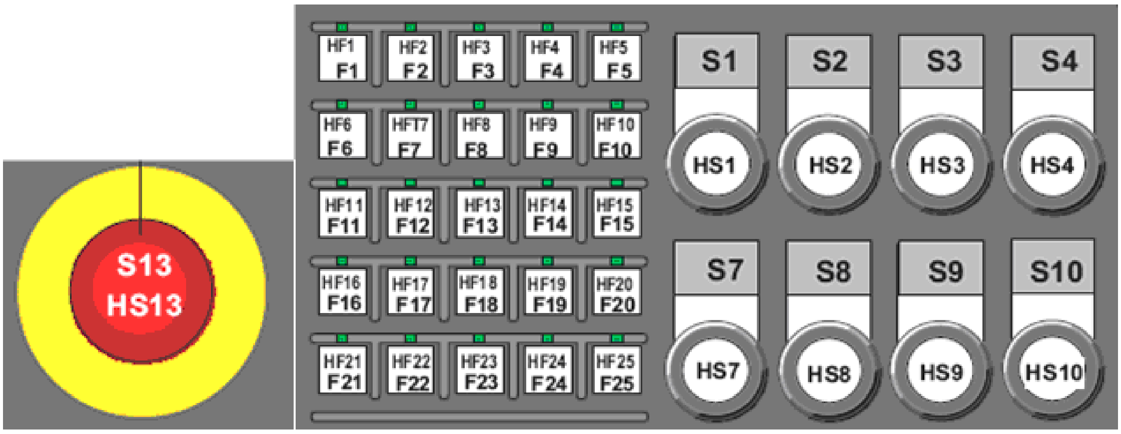


Fig. 6-4: Keys, lamps and LEDs of an MPP

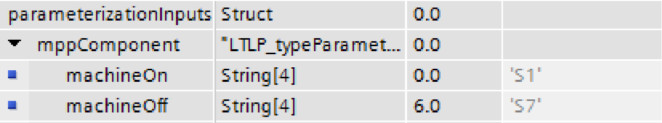
Possible variables
Inputs: **parameterizationInputs** area
Outputs **parameterizationOutputs** area

mppComponent:		
machineOn	STRING[4]	TRANSLINE function unit/ Machine on
machineOff	STRING[4]	TRANSLINE function unit/ Machine off
mediaOn	STRING[4]	TRANSLINE function media ON
mediaOff	STRING[4]	TRANSLINE function media off
basicPosition	STRING[4]	TRANSLINE function initial state
startJog	STRING[4]	TRANSLINE function start/jog single step
acknowledgeFault	STRING[4]	TRANSLINE function acknowledge fault
faultWillBeCorrected	STRING[4]	Message to TRANSLINE Collect/VW Master Interface/PRISMA
allUnitsBack	STRING[4]	TRANSLINE function all units back
stopAfterEndOfCycle	STRING[4]	TRANSLINE function stop after end of cycle
immediateStop	STRING[4]	TRANSLINE function immediate stop
lockRelProtectDoors	STRING[4]	TRANSLINE function lock/release protective doors
emergencyStop	STRING[4]	Emergency Off
automaticMode	STRING[4]	TRANSLINE function automatic mode (AUT)
singleMode	STRING[4]	TRANSLINE function single mode SM
singleStepMode	STRING[4]	TRANSLINE function single step mode SSM
setupMode	STRING[4]	TRANSLINE function setup mode SET

mppComponent:		
auto	STRING[4]	NC function AUTO
mda	STRING[4]	NC function MDA
teach	STRING[4]	NC function TEACH
jog	STRING[4]	NC function JOG
singleBlock	STRING[4]	NC function SINGLE BLOCK
increment1	STRING[4]	NC function increment 1
increment10	STRING[4]	NC function increment 10
increment100	STRING[4]	NC function increment 100
increment1000	STRING[4]	NC function increment 1000
increment10000	STRING[4]	NC function increment 10000
incrementVariable	STRING[4]	NC function variable increments
gotoRefpoint	STRING[4]	NC function goto reference point
repos	STRING[4]	NC function REPOS
ncStop	STRING[4]	NC function NC STOP
ncStart	STRING[4]	NC function NC start
feedStop	STRING[4]	NC function feed stop
feedStart	STRING[4]	NC function feed start
spindleStop	STRING[4]	NC function spindle stop
spindleStart	STRING[4]	NC function spindle start
axisSelectionByHMI	STRING[4]	NC fct. axis selection screen HMI PRO
mcsWcs	STRING[4]	NC function switchover request MCS/WCS

reset	STRING[4]	NC function reset
rapid	STRING[4]	NC function rapid traverse
motionPlus	STRING[4]	NC function motion plus direction
motionMinus	STRING[4]	NC function motion minus direction
axis:		
axis[1]	STRING[4]	NC function axis selection
.		
.		
axis[31]	STRING[4]	NC function axis selection
oem:		
oem[1]	STRING[4]	Unassigned function key for OEM
.		
.		
oem[100]	STRING[4]	Unassigned function key for OEM

Assigning keys and functions is realized by entering the key designation from Fig. 6-4 into the string, which is assigned to the function:



parameterizationInputs	Struct	0.0	
▼ mppComponent	"LTLP_typeParamet...	0.0	
■ machineOn	String[4]	0.0	'S1'
■ machineOff	String[4]	6.0	'S7'

Fig. 6-5: Assignment of keys and function (example, MPP)

General release for acquisition of the keys
In the **alwaysEnabled** area, individual keys can be enabled via a bit interface, despite the lack of a global **ENABLE**. These individual releases are checked cyclically.

6.2 MCP483 - LTLP_ControlPanelMCP [FC541]

This block is required when using an MCP483 as machine control panel.
When creating the **RESTART_PLC** signal, the **LTLP_ControlPanelMCP** reads the assignment of the keys once from the **LTLP_ControlPanelMCPData** [DB541], calculates the new addresses and enters them in data area **inputsAssigned** or **outputsAssigned**.
The FC transfers all key signals to the **LTLP_DeviceInterfaceData** [DB546], area **inputs** as long as at input parameter **ENABLE**=TRUE. It is possible to forward selected keys to the **LTLP_DeviceInterfaceData** even when there is no global enable (ENABLE = FALSE). To do this, in the **LTLP_ControlPanelMCP,** area **alwaysEnabled** enable the appropriate signal.
The LEDs are always read from the **LTLP_DeviceInterfaceData, outputsAssigned** area and then displayed.
All LEDs on the MCP are activated with input parameter **LAMPTEST** = TRUE. This is also realized if, for a locked MCP, a key is pressed. Keys which were enabled in the **alwaysEnabled** area, are the exception.

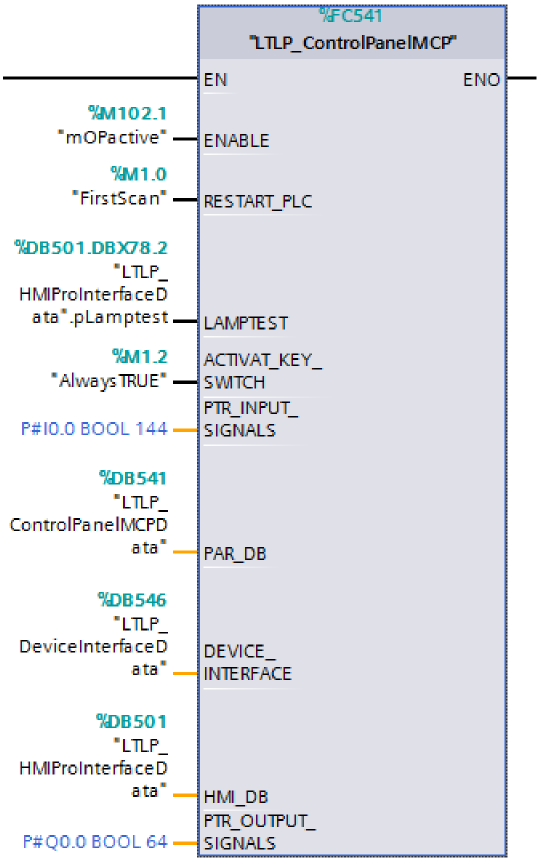
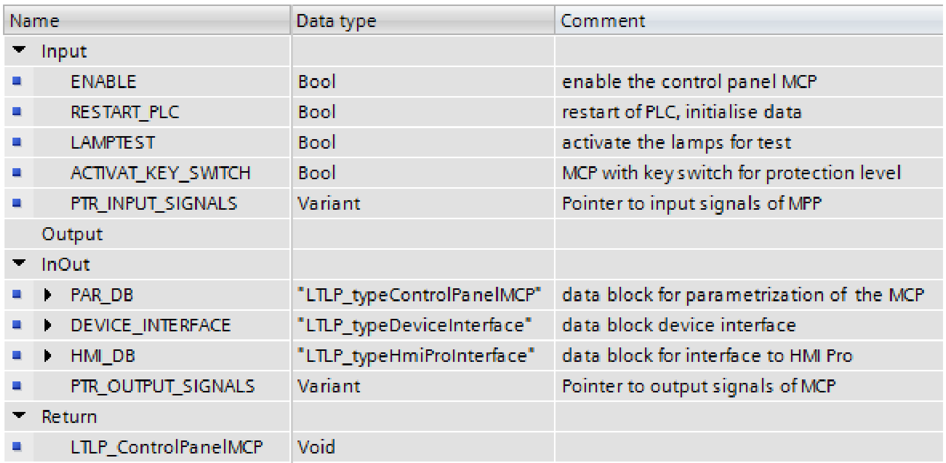


Fig. 6-6: LTLP_ControlPanelMCP [FC541]



Name	Data type	Comment
▼ Input		
ENABLE	Bool	enable the control panel MCP
RESTART_PLC	Bool	restart of PLC, initialise data
LAMPTEST	Bool	activate the lamps for test
ACTIVAT_KEY_SWITCH	Bool	MCP with key switch for protection level
PTR_INPUT_SIGNALS	Variant	Pointer to input signals of MPP
Output		
▼ InOut		
▶ PAR_DB	"LTLP_typeControlPanelMCP"	data block for parametrization of the MCP
▶ DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block device interface
▶ HMI_DB	"LTLP_typeHmiProInterface"	data block for interface to HMI Pro
PTR_OUTPUT_SIGNALS	Variant	Pointer to output signals of MCP
▼ Return		
LTLP_ControlPanelMCP	Void	

Fig. 6-7: LTLP_ControlPanelMCP parameter

Description of input parameters

Table 6-3: Input parameter LTLP_ControlPanelMCP

Parameter	Data type	Description
ENABLE	Bool	Global release for reading in the function and key signals of the MCP483. This must be set to TRUE if this is the active operating panel.
RESTART_PLC	Bool	First block cycle following a CPU restart. This must be done after every reparameterization of the key interface in **LTLP_ControlPanelMCPData** [DB541]. According to the TRANSLINE standard, the **FirstScan** [M1.0] flag is specified here.
LAMPTEST	Bool	Lamp test, all outputs (lamps and LEDs) are controlled with this signal. This is the **LTLP_HMIProInterfaceData.pLamptest** as default setting.
ACTIVAT_KEY_SWITCH	Bool	TRUE for MCP with protection level key-operated switch
PTR_INPUT_SIGNALS	Variant	Pointer to the input signals of the MCP483.

Description of InOut parameters

Table 6-4: InOut parameter LTLP_ControlPanelMCP

Parameter	Data type	Description
PAR_DB	LTLP_typeControl PanelMPP	Data block for parameterizing the pushbuttons and function keys on the MPP. This is the **LTLP_ControlPanelMCPData** [DB541] as default setting.
DEVICE_ INTERFACE	LTLL_typeDevice Interface	The interface between the operator panels and the blocks for acquiring the operating modes, TRANSLINE functions and NC functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
PTR_OUTPUT_SIGNALS	Variant	Pointer to the output signals of the MCP483.

Description of the return value

No value is returned.

Dependencies

none

LTLP_ControlPanelMCPData [DB541]: Parameterizing the keys of the MCP

The keys and LEDs are assigned via **LTLP_ControlPanelMCPData**. Function keys are designated with F, T or R, LEDs with HF, HAT or HR.

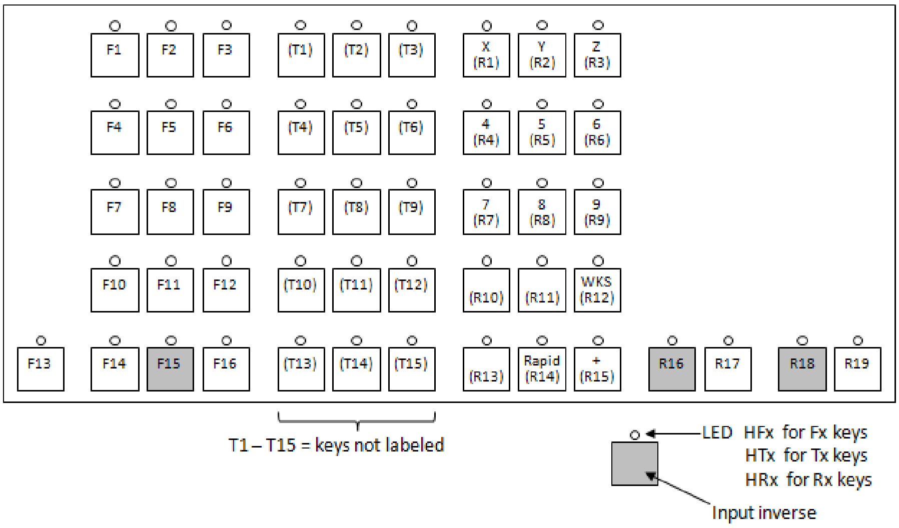


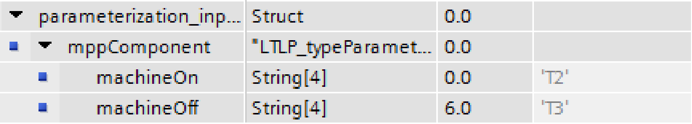
Fig. 6-8: Keys and LEDs of an MCP

Possible variables
Inputs: **parameterizationInputs** area
Outputs **parameterizationOutputs** area

mppComponent:		
machineOn	STRING[4]	TRANSLINE function unit/ Machine on
machineOff	STRING[4]	TRANSLINE function unit/ Machine off
mediaOn	STRING[4]	TRANSLINE function media ON
mediaOff	STRING[4]	TRANSLINE function media off
basicPosition	STRING[4]	TRANSLINE function initial state
startJog	STRING[4]	TRANSLINE function start/jog single step
acknowledgeFault	STRING[4]	TRANSLINE function acknowledge fault
faultWillBeCorrected	STRING[4]	Message to TRANSLINE Collect/VW Master Interface/PRISMA
allUnitsBack	STRING[4]	TRANSLINE function all units back
stopAfterEndOfCycle	STRING[4]	TRANSLINE function stop after end of cycle
immediateStop	STRING[4]	TRANSLINE function immediate stop
lockRelProtectDoors	STRING[4]	TRANSLINE function lock/release protective doors
emergencyStop	STRING[4]	Emergency Off
automaticMode	STRING[4]	TRANSLINE function automatic mode (AUT)
singleMode	STRING[4]	TRANSLINE function single mode SM

singleStepMode	STRING[4]	TRANSLINE fct. single step mode SSM
setupMode	STRING[4]	TRANSLINE function setup mode SET
mppComponent:		
auto	STRING[4]	NC function AUTO
mda	STRING[4]	NC function MDA
teach	STRING[4]	NC function TEACH
jog	STRING[4]	NC function JOG
singleBlock	STRING[4]	NC function SINGLE BLOCK
increment1	STRING[4]	NC function increment 1
increment10	STRING[4]	NC function increment 10
increment100	STRING[4]	NC function increment 100
increment1000	STRING[4]	NC function increment 1000
increment10000	STRING[4]	NC function increment 10000
incrementVariable	STRING[4]	NC function variable increments
gotoRefpoint	STRING[4]	NC function goto reference point
repos	STRING[4]	NC function REPOS
ncStop	STRING[4]	NC function NC STOP
ncStart	STRING[4]	NC function NC start
feedStop	STRING[4]	NC function feed stop
feedStart	STRING[4]	NC function feed start
spindleStop	STRING[4]	NC function spindle stop
spindleStart	STRING[4]	NC function spindle start
axisSelectionByHMI	STRING[4]	NC fct. axis selection screen HMI PRO
mcsWcs	STRING[4]	NC function switchover request MCS/WCS
reset	STRING[4]	NC function reset
rapid	STRING[4]	NC function rapid traverse
motionPlus	STRING[4]	NC function motion plus direction
motionMinus	STRING[4]	NC function motion minus direction
axis:		
axis[1]	STRING[4]	NC function axis selection
.		
.		
axis[31]	STRING[4]	NC function axis selection
oem:		
oem[1]	STRING[4]	Unassigned function key for OEM
.		
.		
oem[100]	STRING[4]	Unassigned function key for OEM

Assigning keys and functions is realized by entering the key designation from Fig. 6-8 into the string, which is assigned to the function:



▼ parameterization_inp...	Struct	0.0	
▼ mppComponent	"LTLP_typeParamet...	0.0	
machineOn	String[4]	0.0	'T2'
machineOff	String[4]	6.0	'T3'

Fig. 6-9: Assignment of keys and function (example, MCP)

General release for acquisition of the keys

In the **alwaysEnabled** area, individual keys can be enabled via a bit interface, despite the lack of a global **ENABLE**. These individual releases are checked cyclically.

6.3 MPP1500 - LTLP_ControlPanelMPP1500 [FC544]

This block is required when using an MPP1500 as machine control panel.
When creating the **RESTART_PLC** signal, the **LTLP_ControlPanelMPP1500** reads the assignment of the keys once from the **LTLP_ControlPanelMPP1500Data** [DB544], calculates the new addresses and enters them in data area **inputsAssigned** or **outputsAssigned**.
The FC transfers all key signals to area **LTLP_DeviceInterfaceData** [DB546], **parameterizationInputs** area until at the input parameter **ENABLE** = TRUE.
The LEDs are always read from the **LTLP_DeviceInterfaceData, outputsAssigned** area and then displayed.
All LEDs on the operator panel are activated with input parameter **LAMPTEST** = TRUE.

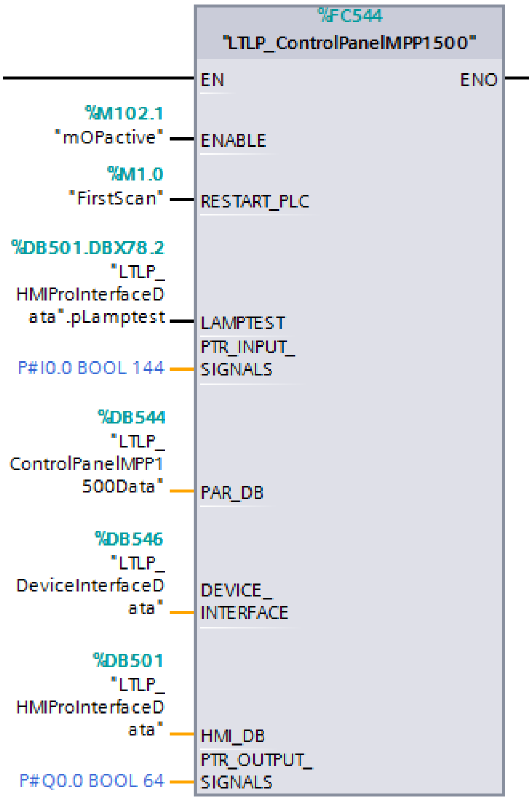
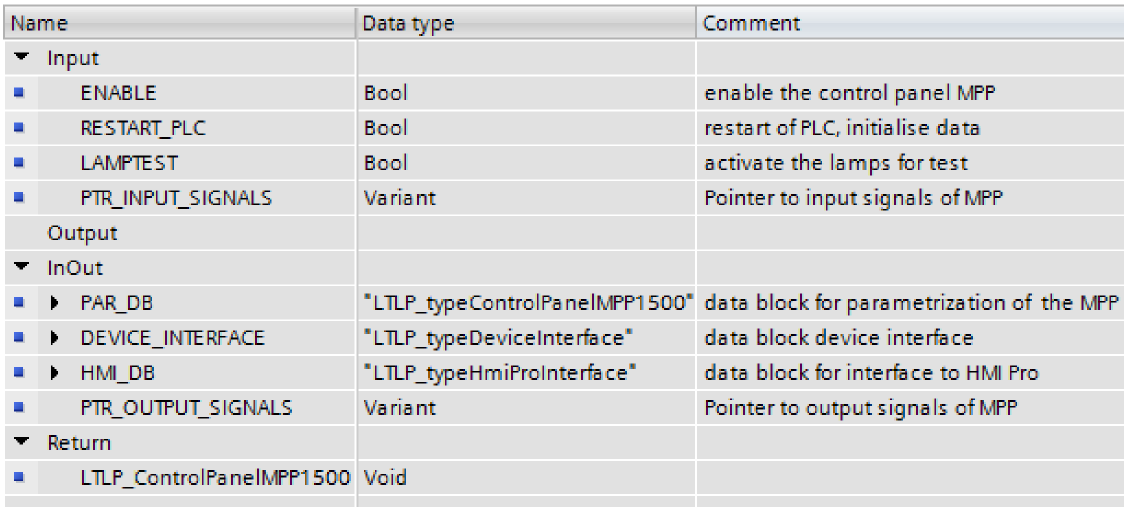


Fig. 6-10: LTLP_ControlPanelMPP1500 [FC544]



Name	Data type	Comment
▼ Input		
▪ ENABLE	Bool	enable the control panel MPP
▪ RESTART_PLC	Bool	restart of PLC, initialise data
▪ LAMPTEST	Bool	activate the lamps for test
▪ PTR_INPUT_SIGNALS	Variant	Pointer to input signals of MPP
Output		
▼ InOut		
▪ ▸ PAR_DB	"LTLP_typeControlPanelMPP1500"	data block for parametrization of the MPP
▪ ▸ DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block device interface
▪ ▸ HMI_DB	"LTLP_typeHmiProInterface"	data block for interface to HMI Pro
▪ PTR_OUTPUT_SIGNALS	Variant	Pointer to output signals of MPP
▼ Return		
▪ LTLP_ControlPanelMPP1500	Void	

Fig. 6-11: LTLP_ControlPanelMPP1500 parameter

Description of input parameters

Table 6-5: Input parameter LTLP_ControlPanelMPP1500

Parameter	Data type	Description
ENABLE	Bool	Global release for reading in the function and key signals of the MPP1500. This must be set to TRUE if this is the active operating panel.
RESTART_PLC	Bool	First block cycle following a CPU restart. This must be done after every reparameterization of the key interface in **LTLP_ControlPanelMPP1500Data** [DB544]. According to the TRANSLINE standard, the **FirstScan** [M1.0] flag is specified here.
LAMPTEST	Bool	Lamp test, all outputs (lamps and LEDs) are controlled with this signal.
PTR_INPUT_SIGNALS	Variant	Pointer to the input signals of the MPP1500.

Description of InOut parameters

Table 6-6: InOut parameter LTLP_ControlPanelMPP1500

Parameter	Data type	Description
PAR_DB	LTLP_typeControlPanelMPP1500	Data block to parameterize the pushbuttons and function keys of the HT8. This is the **LTLP_typeControlPanelMPP1500Data** [DB544] as default setting.
DEVICE_INTERFACE	LTLL_typeDeviceInterface	The interface between the operator panels and the blocks for acquiring the operating modes, TRANSLINE functions and NC functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
HMI_DB	LTLP_typeHMIProInterface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
PTR_OUTPUT_SIGNALS	Variant	Pointer to the output signals of the MPP1500

Description of the return value

No value is returned.

Dependencies

none

LTLP_ControlPanelMPP1500Data [DB544]: Parameterizing the keys of the MPP1500

The keys, lamps and LEDs are assigned via **LTLP_ControlPanelMPP1500Data**. The following identifiers are permitted for parameterizing inputs or outputs:

- Inputs: S1 to S54 and KT1 to KT9
- Outputs: are HS1 to HS12, HS14 to HS17, HS41 to HS50 and HK1 to HKT6

Notes on using the HS1 to HS12 identifiers

For these identifiers, in addition, the required LED color red/green and/or blue must be specified by setting the appropriate color bit. If a bit is not set and the function is active, then no output is set.

Table 6-7: Standard assignment of the MPP1500 inputs and outputs

Function	Key	LED	Color	DeviceInterface
Machine on	S1	HS1	r,g,b	machineOn
Machine off	S7	HS7		machineOff
Initial state	S2	HS2	g	basicPosition
Acknowledge fault	S4	HS4	r,g,b	acknowledgeFault
Stop after end of cycle	S8	HS8	b	stopAfterEndOfCycle
Immediate stop	S9	HS9	r	immediateStop
Protective door	S10	HS10	g,b	lockReleaseProtectDoors
MSO1	S5	HS5	g	oemFunctions[1]
MSO2	S11	HS11	g	oemFunctions [2]
MSO3	S6	HS6	g	oemFunctions [3]
MSO5	S12	HS12	g	oemFunctions [5]

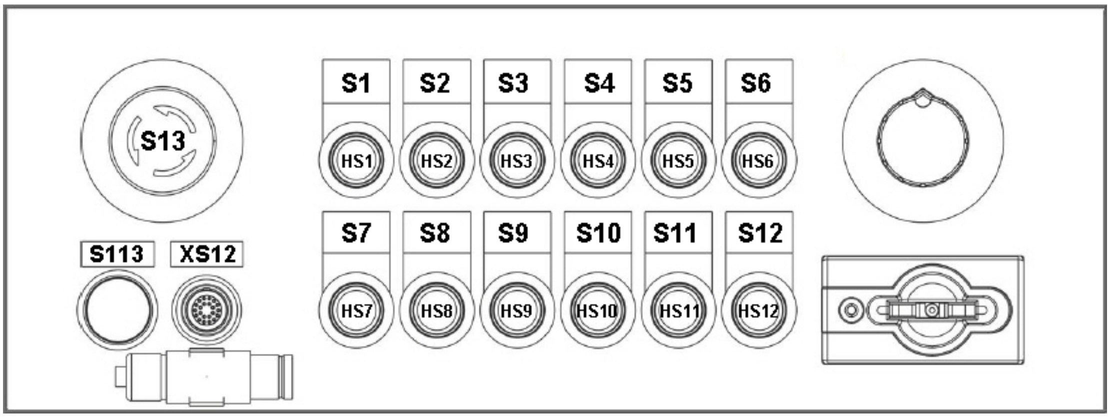


Fig. 6-12: Possible keys, lamps and LEDs of an MPP1500

Possible variables
Inputs: **parameterizationInputs** area

mppComponent:		
machineOn	STRING[4]	TRANSLINE function unit/ Machine on
machineOff	STRING[4]	TRANSLINE function unit/ Machine off
mediaOn	STRING[4]	TRANSLINE function media on
mediaOff	STRING[4]	TRANSLINE function media off
basicPosition	STRING[4]	TRANSLINE function initial state
startJog	STRING[4]	TRANSLINE fct. start/jog single step
acknowledgeFault	STRING[4]	TRANSLINE function acknowledge fault
faultWillBeCorrected	STRING[4]	Message to TRANSLINE Collect/VW Master Interface/PRISMA
allUnitsBack	STRING[4]	TRANSLINE function all units back
stopAfterEndOfCycle	STRING[4]	TRANSLINE function stop after end of cycle
immediateStop	STRING[4]	TRANSLINE function immediate stop
lockRelProtectDoors	STRING[4]	TRANSLINE function lock/release protective doors
emergencyStop	STRING[4]	Emergency Off
automaticMode	STRING[4]	TRANSLINE function automatic mode (AUT)
singleMode	STRING[4]	TRANSLINE function single mode SM
singleStepMode	STRING[4]	TRANSLINE function single step mode SSM
setupMode	STRING[4]	TRANSLINE function setup mode SET
mppComponent:		
auto	STRING[4]	NC function AUTO
mda	STRING[4]	NC function MDA
teach	STRING[4]	NC function TEACH
jog	STRING[4]	NC function JOG
singleBlock	STRING[4]	NC function SINGLE BLOCK
increment1	STRING[4]	NC function increment 1
increment10	STRING[4]	NC function increment 10
increment100	STRING[4]	NC function increment 100
increment1000	STRING[4]	NC function increment 1000
increment10000	STRING[4]	NC function increment 10000
incrementVariable	STRING[4]	NC function variable increments
gotoRefpoint	STRING[4]	NC function goto reference point
repos	STRING[4]	NC function REPOS
ncStop	STRING[4]	NC function NC STOP
ncStart	STRING[4]	NC function NC start
feedStop	STRING[4]	NC function feed stop
feedStart	STRING[4]	NC function feed start
spindleStop	STRING[4]	NC function spindle stop
spindleStart	STRING[4]	NC function spindle start
axisSelectionByHMI	STRING[4]	NC function axis selection screen HMI PRO
mcsWcs	STRING[4]	NC function switchover request MCS/WCS
reset	STRING[4]	NC function reset

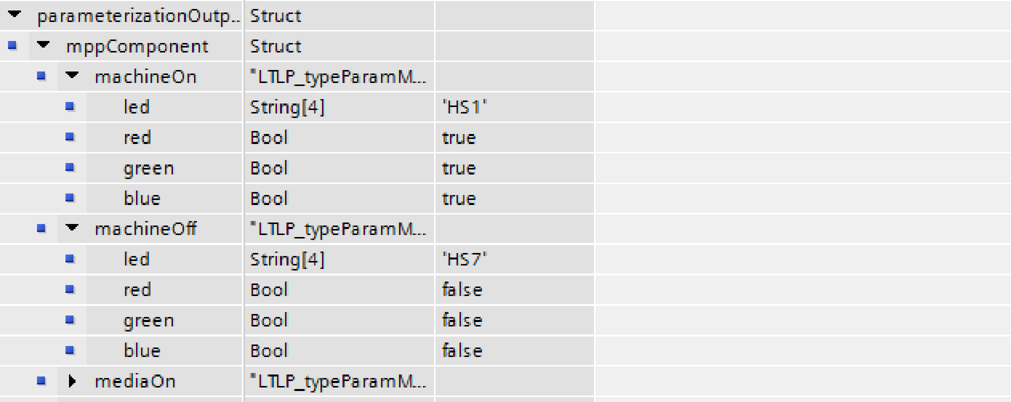
rapid	STRING[4]	NC function rapid traverse
motionPlus	STRING[4]	NC function motion plus direction
motionMinus	STRING[4]	NC function motion minus direction
axis:		
axis[1]	STRING[4]	NC function axis selection
.		
.		
axis[31]	STRING[4]	NC function axis selection
oem:		
oem[1]	STRING[4]	Unassigned function key for OEM
.		
.		
oem[100]	STRING[4]	Unassigned function key for OEM

Outputs **parameterizationOutputs** area

The **parameterizationOutputs** area contains the same elements as the **parameterizationInputs** area, however these are type **LTLP_typeParamMPP1500Output**

mppComponent:		
machineOn	LTLP_typeParamMPP1500Output	TRANSLINE function unit/ Machine on
machineOff	LTLP_typeParamMPP1500Output	TRANSLINE function unit/ Machine off
mediaOn	LTLP_typeParamMPP1500Output	TRANSLINE function media ON
mediaOff	LTLP_typeParamMPP1500Output	TRANSLINE function media off
basicPosition	LTLP_typeParamMPP1500Output	TRANSLINE function initial state
:	**:**	**:**
:	**:**	**:**

Keys and functions are assigned to one another by entering the key designation from Fig. 6-12 into the string to which the function is assigned:



parameterizationOutp...	Struct	
mppComponent	Struct	
machineOn	"LTLP_typeParamM...	
led	String[4]	'HS1'
red	Bool	true
green	Bool	true
blue	Bool	true
machineOff	"LTLP_typeParamM...	
led	String[4]	'HS7'
red	Bool	false
green	Bool	false
blue	Bool	false
mediaOn	"LTLP_typeParamM...	

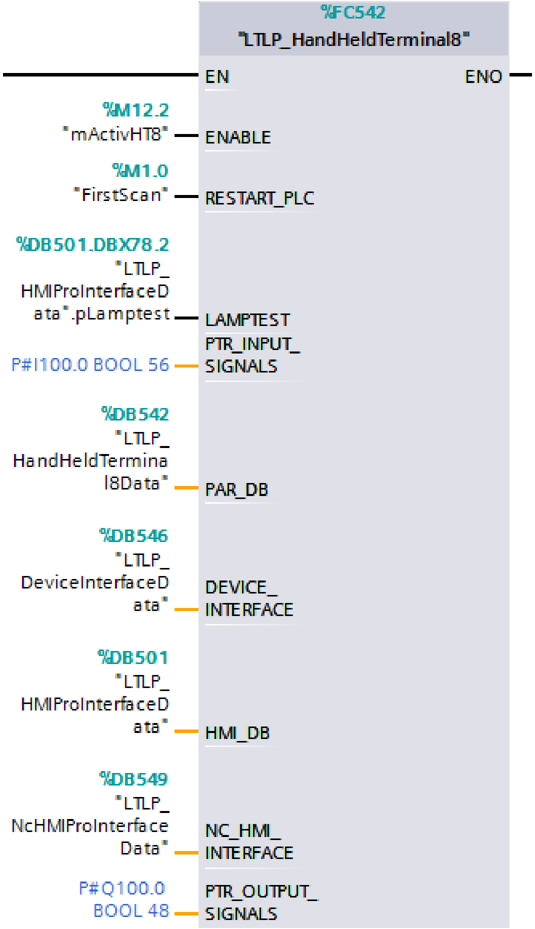
Fig. 6-13: Assignment of keys and function (example: MPP1500)

General release for acquisition of the keys

In the **alwaysEnabled** area, individual keys can be enabled via a bit interface, despite the lack of a global **ENABLE**. These individual releases are checked cyclically.

6.4 HT8 - LTLP_HandHeldTerminal8 [FC542]

This block is required when using an HT8 as handheld terminal.
When creating the **RESTART_PLC** signal, the **LTLP_HandHeldTerminal8** reads the assignment of the keys once from the **LTLP_HandHeldTerminal8Data** [DB542], calculates the new addresses and enters them in data area **inputsAssigned** or **outputsAssigned**.
The FC transfers all key signals to **LTLP_DeviceInterfaceData** [DB546], **parameterizationInputs** area until at the input parameter **ENABLE** = TRUE.
The LEDs are always read from the **LTLP_DeviceInterfaceData, outputsAssigned** area and then displayed.
All LEDs on the HT8 are activated with input parameter **LAMPTEST** = TRUE.



%FC542
"LTLP_HandHeldTerminal8"

EN — ENO

%M12.2 "mActivHT8" — ENABLE

%M1.0 "FirstScan" — RESTART_PLC

%DB501.DBX78.2 "LTLP_HMIProInterfaceData".pLamptest — LAMPTEST

P#I100.0 BOOL 56 — PTR_INPUT_SIGNALS

%DB542 "LTLP_HandHeldTerminal8Data" — PAR_DB

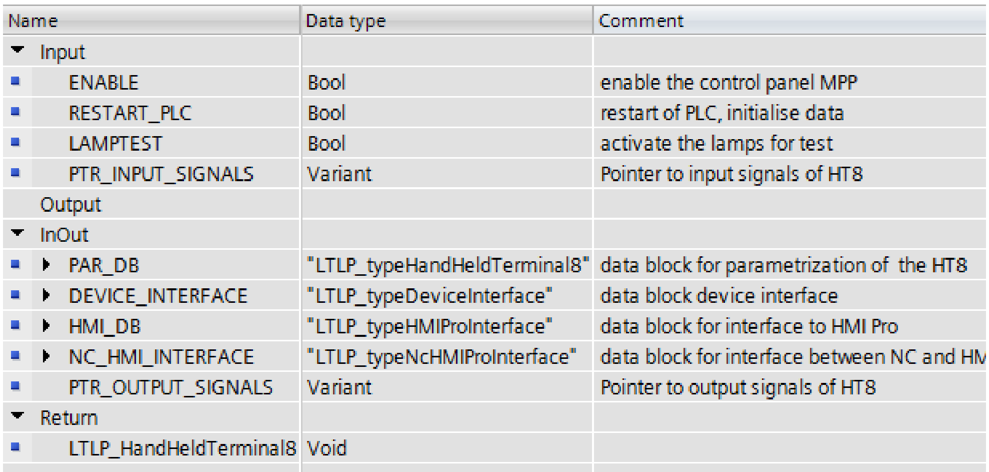
%DB546 "LTLP_DeviceInterfaceData" — DEVICE_INTERFACE

%DB501 "LTLP_HMIProInterfaceData" — HMI_DB

%DB549 "LTLP_NcHMIProInterfaceData" — NC_HMI_INTERFACE

P#Q100.0 BOOL 48 — PTR_OUTPUT_SIGNALS

Fig. 6-14: LTLP_HandHeldTerminal8 [FC542]



Name	Data type	Comment
▼ Input		
■ ENABLE	Bool	enable the control panel MPP
■ RESTART_PLC	Bool	restart of PLC, initialise data
■ LAMPTEST	Bool	activate the lamps for test
■ PTR_INPUT_SIGNALS	Variant	Pointer to input signals of HT8
Output		
▼ InOut		
■ ▶ PAR_DB	"LTLP_typeHandHeldTerminal8"	data block for parametrization of the HT8
■ ▶ DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block device interface
■ ▶ HMI_DB	"LTLP_typeHMIProInterface"	data block for interface to HMI Pro
■ ▶ NC_HMI_INTERFACE	"LTLP_typeNcHMIProInterface"	data block for interface between NC and HM
■ PTR_OUTPUT_SIGNALS	Variant	Pointer to output signals of HT8
▼ Return		
■ LTLP_HandHeldTerminal8	Void	

Fig. 6-15: LTLP_HandHeldTerminal8 parameter

Description of input parameters

Table 6-8: Input parameter LTLP_HandHeldTerminal8

Parameter	Data type	Description
ENABLE	Bool	Global release for reading in the function and key signals of the HT8. This must be set to TRUE if this is the active HHU.
RESTART_PLC	Bool	First block cycle following a CPU restart. This must be done after every reparameterization of the key interface in **LTLP_HandHeldTerminal8Data** [DB542]. According to the TRANSLINE standard, the **FirstScan** [M1.0] flag is specified here.
LAMPTEST	Bool	Lamp test, all outputs (LEDs) are controlled with this signal.
PTR_INPUT_SIGNALS	Variant	Pointer to the input signals of the HT8.

Description of InOut parameters

Table 6-9: InOut parameter LTLP_HandHeldTerminal8

Parameter	Data type	Description
PAR_DB	LTLP_typeHand HeldTerminal8	Data block to parameterize the pushbuttons and function keys of the HT8. This is the **LTLP_HandHeldTerminal8Data** [DB542] as default setting.
DEVICE_ INTERFACE	LTLL_typeDevice Interface	The interface between the operator panels and the blocks for acquiring the operating modes, TRANSLINE functions and NC functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
NC_HMI_ INTERFACE	LTLP_typeNcHmi ProInterface	Interface between HMI PRO and PLC. This is the **LTLP_NcHMIProInterfaceData** [DB549] as default setting.
PTR_OUTPUT_ SIGNALS	Variant	Pointer to the output signals of the HT8

Description of the return value

No value is returned.

Dependencies

none

LTLP_HandHeldTerminal8Data parameterization of the keys of the **HT8 [DB542]**

The U keys and LEDs are assigned via data block **LTLP_HandHeldTerminal8Data**. U keys are identified with U, LEDs with LU. If the standard key assignment is to be changed, the axis selection screen in the HMI PRO must be customized. The NC keys cannot be changed because there are no slide-in labels for this case.

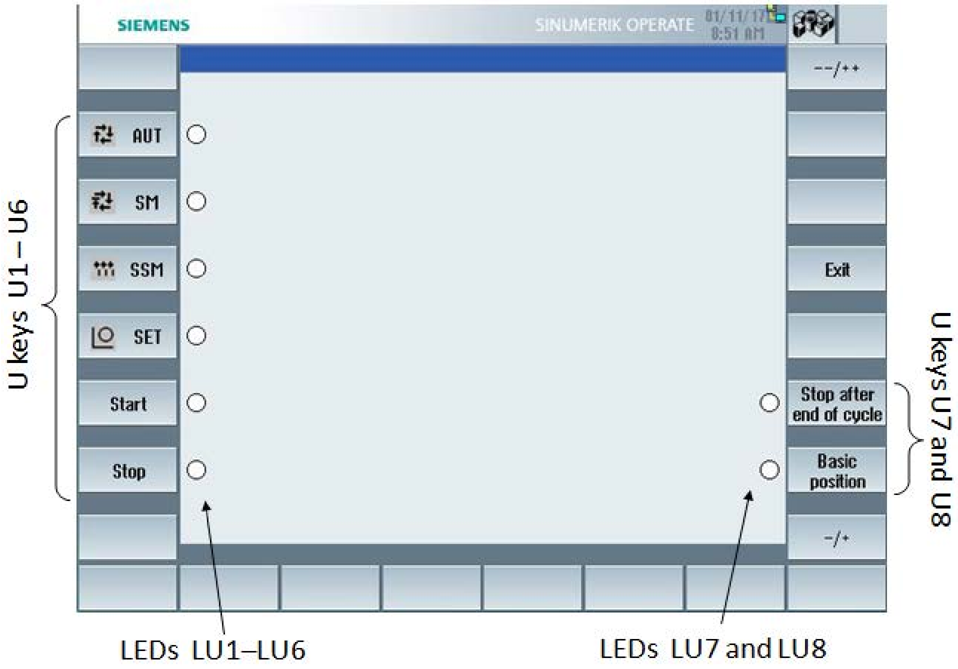


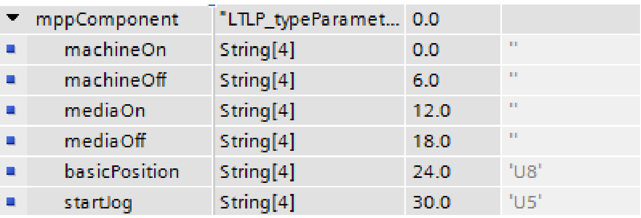
Fig. 6-16: Possible keys and LEDs of an HT8

Possible variables
Inputs: **parameterizationInputs** area
Outputs **parameterizationOutputs** area

mppComponent:

machineOn	STRING[4]	TRANSLINE function unit/ Machine on
machineOff	STRING[4]	TRANSLINE function unit/ Machine off
mediaOn	STRING[4]	TRANSLINE function media ON
mediaOff	STRING[4]	TRANSLINE function media off
basicPosition	STRING[4]	TRANSLINE function initial state
startJog	STRING[4]	TRANSLINE function start/jog single step
acknowledgeFault	STRING[4]	TRANSLINE function acknowledge fault
faultWillBeCorrected	STRING[4]	Message to TRANSLINE Collect/VW Master Interface/PRISMA
allUnitsBack	STRING[4]	TRANSLINE function all units back
stopAfterEndOfCycle	STRING[4]	TRANSLINE function stop after end of cycle
immediateStop	STRING[4]	TRANSLINE function immediate stop
lockRelProtectDoors	STRING[4]	TRANSLINE function lock/release protective doors
emergencyStop	STRING[4]	Emergency Off
automaticMode	STRING[4]	TRANSLINE function automatic mode (AUT)
singleMode	STRING[4]	TRANSLINE function single mode SM
singleStepMode	STRING[4]	TRANSLINE function single step mode SSM
setupMode	STRING[4]	TRANSLINE function setup mode SET

Assigning keys and functions is realized by entering the key designation from Fig. 6-16 into the string, which is assigned to the function:



▼ mppComponent	"LTLP_typeParamet...	0.0	
■ machineOn	String[4]	0.0	''
■ machineOff	String[4]	6.0	''
■ mediaOn	String[4]	12.0	''
■ mediaOff	String[4]	18.0	''
■ basicPosition	String[4]	24.0	'U8'
■ startJog	String[4]	30.0	'U5'

Fig. 6-17: Assignment of keys and function (example, HT8)

6.5 LTLP_AxisSelectScreen [FC545]: Transline function types on the operator panel

This block is used if the operator panel does not have any keys to select TRANSLINE functions, e.g. when using an IPC477 with MPP1500.
The block processes the keys and LED signals of the U key screen. This is configured as axis selection screen, and has no function to select axes; however, the U keys to select TRANSLINE functions.
LTLP_AxisSelectScreen processes the keys and LED signals of the U keys screen. The U key screen is configured as an axis selection screen, which does not contain display and selection options of the axes, but it does contain the U keys for performing the TRANSLINE functions. (Configuring, see Chapter 5.2 Configuring in HMI PRO CS when connected to a PLC controller).
When creating the **RESTART_PLC** signal, the **LTLP_AxisSelectScreen** reads the assignment of the keys once from the **LTLP_AxisSelectScreenData** [DB545], calculates the new addresses and enters them in data area **inputsAssigned** or **outputsAssigned** as long as at the input parameter **ENABLE** = TRUE.
The LEDs shown in the diagram are always read from the **LTLP_DeviceInterfaceData**, **outputsAssigned** area and then displayed.
All LEDs in the screen are activated with input parameter **LAMPTEST** = TRUE.

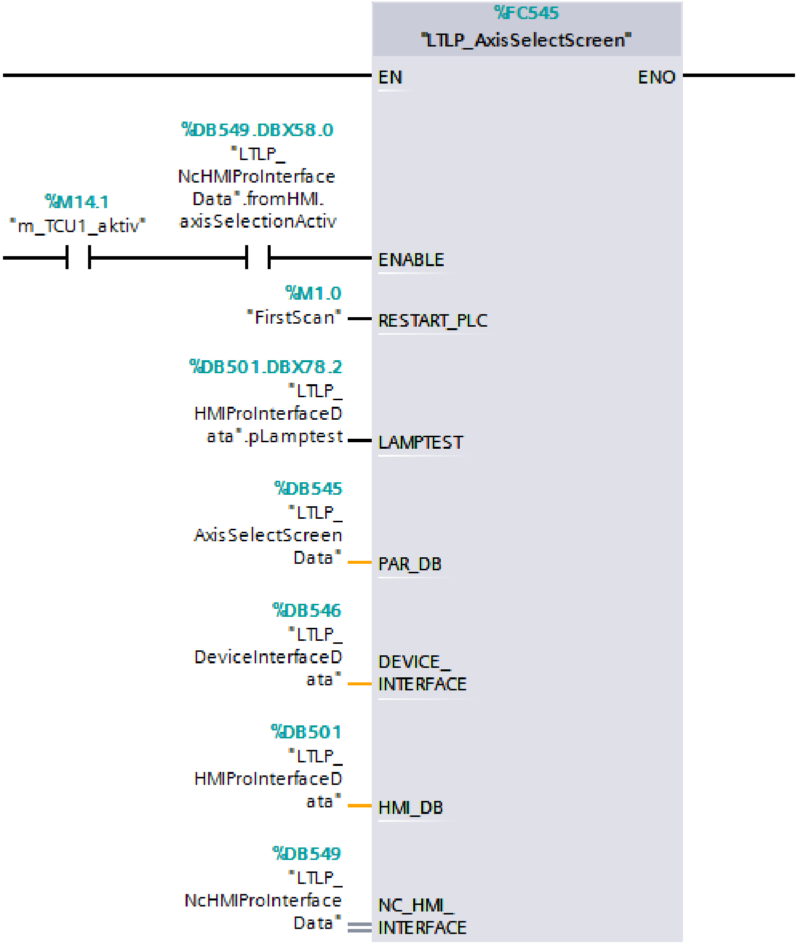
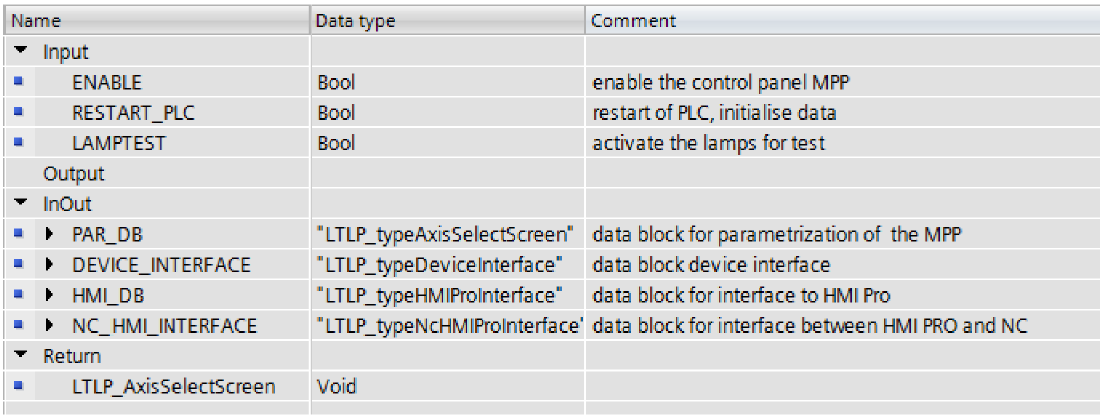


Fig. 6-18: LTLP_AxisSelectScreen [FC545]



Name	Data type	Comment
Input		
ENABLE	Bool	enable the control panel MPP
RESTART_PLC	Bool	restart of PLC, initialise data
LAMPTEST	Bool	activate the lamps for test
Output		
InOut		
PAR_DB	"LTLP_typeAxisSelectScreen"	data block for parametrization of the MPP
DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block device interface
HMI_DB	"LTLP_typeHMIProInterface"	data block for interface to HMI Pro
NC_HMI_INTERFACE	"LTLP_typeNcHMIProInterface'	data block for interface between HMI PRO and NC
Return		
LTLP_AxisSelectScreen	Void	

Fig. 6-19: LTLP_AxisSelectScreen parameter

Description of input parameters

Table 6-10: Input parameter LTLP_AxisSelectScreen

Parameter	Data type	Description
ENABLE	Bool	Global enable to read-in the function and key signals of the U keys screen. This should be set to TRUE if this screen is active.
RESTART_PLC	Bool	First block cycle following a CPU restart. This must be done after every reparameterization of the key interface in **LTLP_AxisSelectScreen** [DB545]. According to the TRANSLINE standard, the **FirstScan** [M1.0] flag is specified here.
LAMPTEST	Bool	Lamp test, all outputs (LEDs shown in the screen) are activated using this signal.

Description of InOut parameters

Table 6-11: InOut parameter LTLP_AxisSelectScreen

Parameter	Data type	Description
PAR_DB	LTLP_typeAxis SelectScreen	Data block to parameterize the keys of the U keys screen. This is the **LTLP_AxisSelectScreenData** [DB545] as default setting.
DEVICE_ INTERFACE	LTLL_typeDevice Interface	The interface between the operator panels and the blocks for acquiring the operating modes, TRANSLINE functions and NC functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
NC_HMI_ INTERFACE	LTLP_typeNcHMI ProInterface	Interface between HMI and the PLC. This is the **LTLP_NcHmiProInterfaceData** [DB549] as default setting.

Description of the return value

No value is returned.

Dependencies

none

LTLP_AxisSelectScreenData [DB545]: Parameterizing the U keys

The U keys and LEDs are assigned via data block **LTLP_AxisSelectScreenData**. U keys are identified with U, LEDs with LU. If the default key assignments is to be changed, the screen must be adapted in HMI PRO (see Chap. 5.2 Configuration in HMI PO CS when connected to a PLC controller).

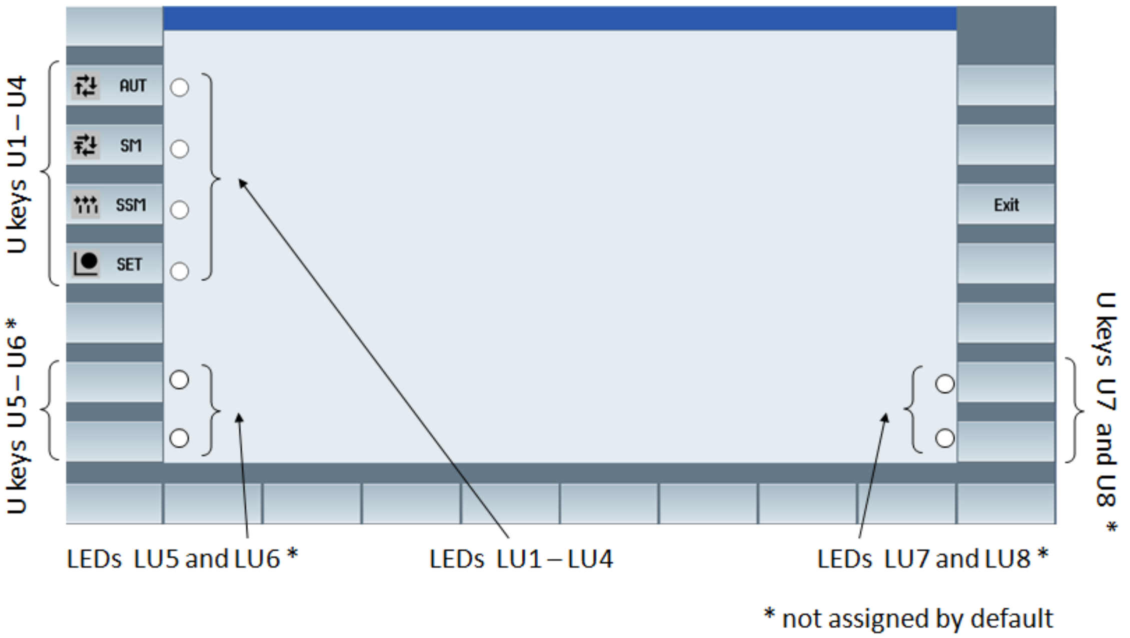


Fig. 6-20: Possible U keys and LEDs of the U keys screen

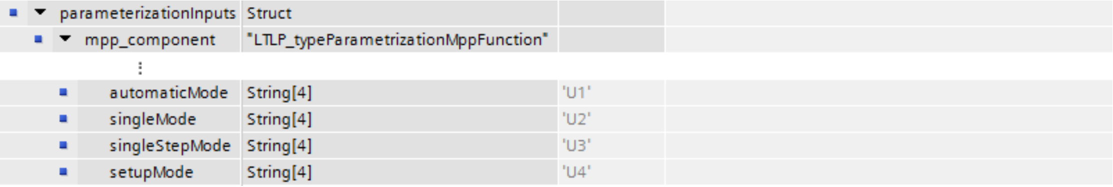
Possible variables
Inputs: **parameterizationInputs** area
Outputs **parameterizationOutputs** area

mppComponent:

machineOn	STRING[4]	TRANSLINE function unit/ Machine on
machineOff	STRING[4]	TRANSLINE function unit/ Machine off
mediaOn	STRING[4]	TRANSLINE function media ON
mediaOff	STRING[4]	TRANSLINE function media off
basicPosition	STRING[4]	TRANSLINE function initial state
startJog	STRING[4]	TRANSLINE function start/jog single step
acknowledgeFault	STRING[4]	TRANSLINE function acknowledge fault
faultWillBeCorrected	STRING[4]	Message to TRANSLINE Collect/VW Master Interface/PRISMA
allUnitsBack	STRING[4]	TRANSLINE function all units back
stopAfterEndOfCycle	STRING[4]	TRANSLINE function stop after end of cycle
immediateStop	STRING[4]	TRANSLINE function immediate stop
lockRelProtectDoors	STRING[4]	TRANSLINE function lock/release protective doors

emergencyStop	STRING[4]	Emergency Off
automaticMode	STRING[4]	TRANSLINE function automatic mode (AUT)
singleMode	STRING[4]	TRANSLINE function single mode SM
singleStepMode	STRING[4]	TRANSLINE function single step mode SSM
setupMode	STRING[4]	TRANSLINE function setup mode SET

Assigning keys and functions is realized by entering the key designation from Fig. 6-20 into the string, which is assigned to the function:



▼ parameterizationInputs	Struct	
▼ mpp_component	"LTLP_typeParametrizationMppFunction"	
⋮		
automaticMode	String[4]	'U1'
singleMode	String[4]	'U2'
singleStepMode	String[4]	'U3'
setupMode	String[4]	'U4'

Fig. 6-21: Assignment of keys and function

6.6 LTLP_OperatingMode [FC547]

Attention

This block is not secure in terms of "Safety".
It does not release the OEM from its obligations defined in the machinery directive.

The **LTLP_OperatingMode** must be called in the operating mode block **MSO_general** between the operator panel blocks **LTLP_ControlPanelMPP**, **LTLP_HandHeldTerminal8** or **LTLP_AxisSelectScreen** and the block for the TRANSLINE functions **LTLP_FunctionModeTransline**.
Depending on the selected Mode of Safe Operation**, LTLP_OperatingMode** preselects a TRANSLINE function type. An attempt is always made to retain the existing function. The block only influences the selection of the TRANSLINE function type. Users must program activating the TRANSLINE function as well as the feedback signal (e.g. **pSetupMode**, **setupMode**).

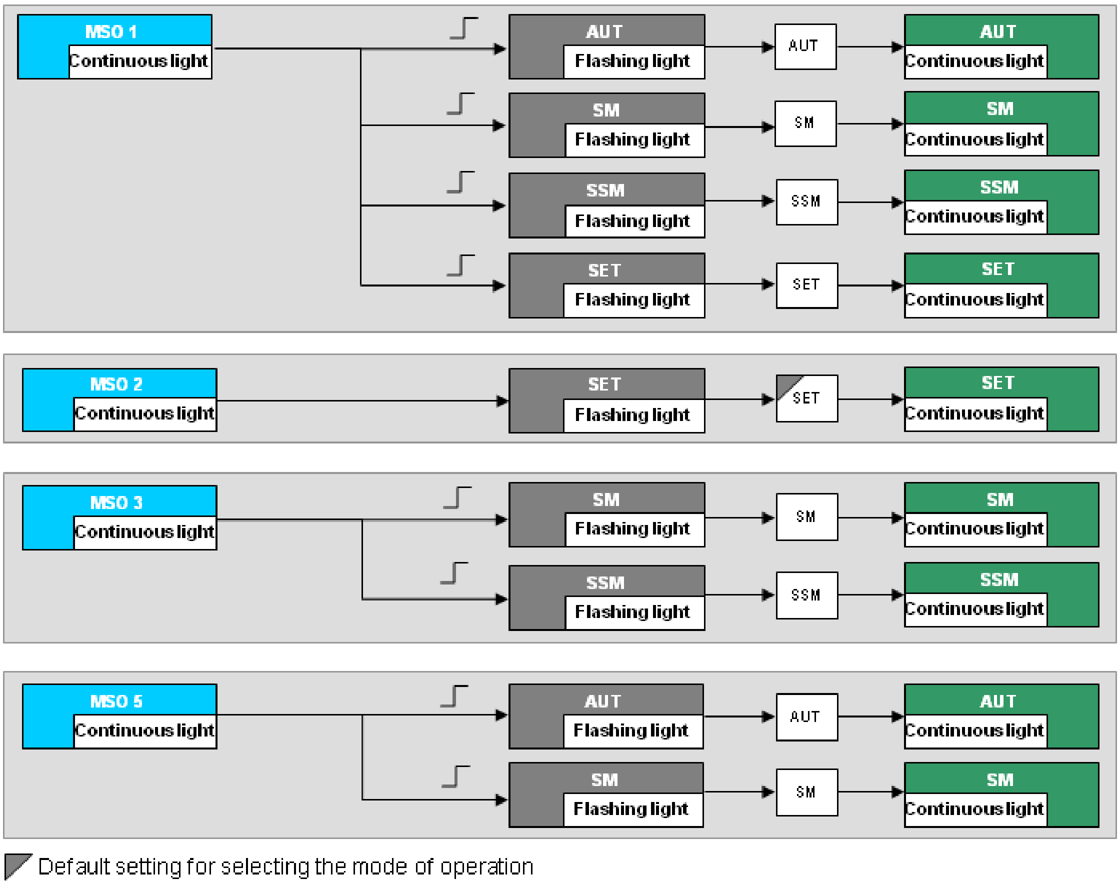


Fig. 6-22: Diagram of TRANSLINE Modes of Safe Operations

The permitted key functions are defined via the **LTLP_OperatingModeData** [DB547], taking the Mode of Safe Operations and the selected TRANSLINE function into consideration.

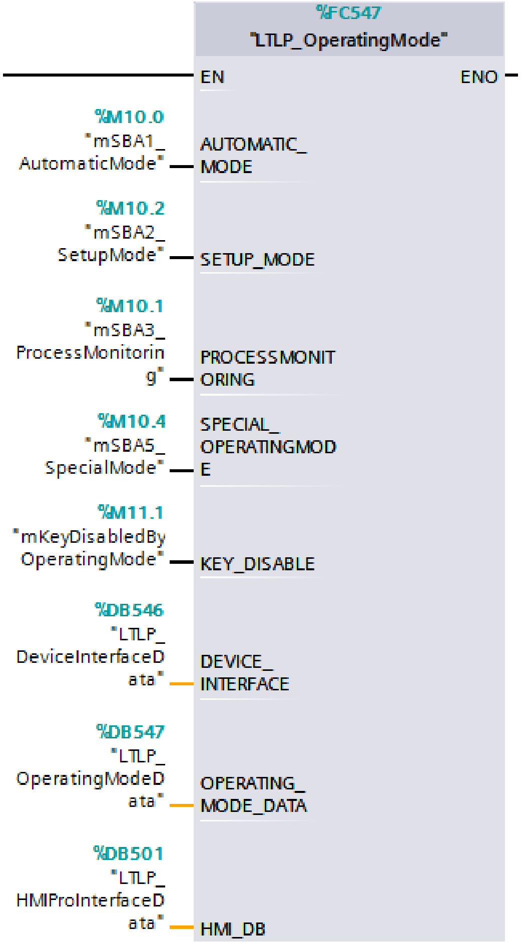
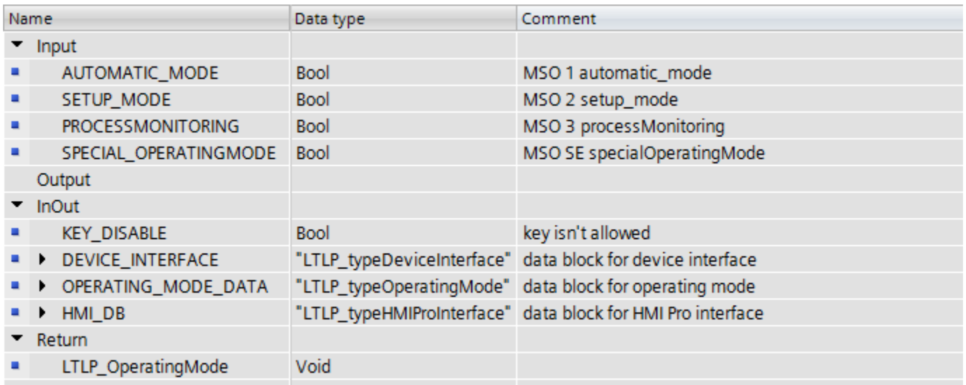


Fig. 6-23: LTLP_OperatingMode [FC547]



Name	Data type	Comment
Input		
AUTOMATIC_MODE	Bool	MSO 1 automatic_mode
SETUP_MODE	Bool	MSO 2 setup_mode
PROCESSMONITORING	Bool	MSO 3 processMonitoring
SPECIAL_OPERATINGMODE	Bool	MSO SE specialOperatingMode
Output		
InOut		
KEY_DISABLE	Bool	key isn't allowed
DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block for device interface
OPERATING_MODE_DATA	"LTLP_typeOperatingMode"	data block for operating mode
HMI_DB	"LTLP_typeHMIProInterface"	data block for HMI Pro interface
Return		
LTLP_OperatingMode	Void	

Fig. 6-24: LTLP_OperatingMode parameter

Description of input parameters

Table 6-12: Input parameter LTLP_OperatingMode

Parameter	Data type	Description
AUTOMATIC_MODE	Bool	TRUE=MSO 1 or SBA 1 production mode, i.e. traversing with closed protection area
SETUP_MODE	Bool	TRUE = MSO 2 or SBA 2 setup mode, i.e. traversing with open protection area
PROCESS_MONITORING	Bool	TRUE = MSO 3 or SBA 3 process monitoring, i.e. traversing with open protection area
SPECIAL_OPERATINGMODE	Bool	TRUE = MSO SE or SBA 4/5 process monitoring, i.e. traversing with open protection area

Description of InOut parameters

Table 6-13: InOut parameter LTLP_OperatingMode

Parameter	Data type	Description
KEY_DISABLE	Bool	Signal is set if the operator presses a locked key
DEVICE_ INTERFACE	LTLL_typeDevice Interface	The interface between the operator panels and the blocks for acquiring the operating modes and the TRANSLINE functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
OPERATING_ MODE_DATA	LTLP_typeOperation Mode	Supplies the interlock interface **LTLP_OperatingModeData** [DB547]
HMI_DB	LTLP_typeHMIPro Interface	Supplies the TRANSLINE functions within the HMI PRO data block. **LTLP_HMIProInterfaceData** [DB501] is the default setting.

Description of the return value

No value is returned.

Dependencies

None

LTLP_OperatingModeData [DB547]

The locking of the key functions takes place in this data block at the device interface **LTLP_DeviceInterfaceData** [DB546]; i.e. impermissible keys are suppressed by this block. There are interlocks from

AUTOMATIC_MODE	**// MSO1 or SBA1**
TRANSLINE function	Linked mode (automaticMode)
TRANSLINE function	Single mode (singleMode)
TRANSLINE function	Single step mode (singleStepMode)
TRANSLINE function	Setup mode (setupMode)
SETUP_MODE	**// MSO2 or SBA2**
TRANSLINE function	Setup mode (setupMode)
PROCESS_MONITORING	**// MSO3 or SBA3**
TRANSLINE function	Single mode (singleMode)
TRANSLINE function	Single step mode (singleStepMode)
SPECIAL_OPERATINGMODE	**// MSO SE or SBA4/5**
TRANSLINE function	Linked mode (automaticMode)
TRANSLINE function	Single mode (singleMode)

Description for one of the TRANSLINE functions, e.g.

automaticMode		MSO 1 / Mode of Safe Operation 1
tlFunctionAutomaticMode		Automatic mode
mppFunctions		
machineOn	Bool	TRANSLINE function unit /Machine on
machineOff	Bool	TRANSLINE function unit/ Machine off
mediaOn	Bool	TRANSLINE function media ON
mediaOff	Bool	TRANSLINE function media off
basicPosition	Bool	TRANSLINE function initial state
startJog	Bool	TRANSLINE function start/jog single step
acknowledgeFault	Bool	TRANSLINE function acknowledge fault
faultWillBeCorrected	Bool	Message to TRANSLINE Collect/VW Master Interface/PRISMA
allUnitsBack	Bool	TRANSLINE function all units back
stopAfterEndOfCycle	Bool	TRANSLINE function stop after end of cycle
immediateStop	Bool	TRANSLINE function immediate stop
lockReleaseProtectDoors	Bool	TRANSLINE function lock/release protective doors
emergencyStop	Bool	Emergency Off
automaticMode	Bool	TRANSLINE function automatic mode AUT
singleMode	Bool	TRANSLINE function single mode SM
singleStepMode	Bool	TRANSLINE function single step mode SSM
setupMode	Bool	TRANSLINE function setup mode SET

mcpFunctions		
auto	Bool	NC function AUTO
mda	Bool	NC function MDA
teach	Bool	NC function TEACH
jog	Bool	NC function JOG
singleBlock	Bool	NC function SINGLE BLOCK
increment1	Bool	NC function increment 1
increment10	Bool	NC function increment 10
increment100	Bool	NC function increment 100
increment1000	Bool	NC function increment 1000
increment10000	Bool	NC function increment 10000
incrementVariable	Bool	NC function variable increments
gotoRefpoint	Bool	NC function goto reference point
repos	Bool	NC function REPOS
ncStop	Bool	NC function NC STOP
ncStart	Bool	NC function NC start
feedStop	Bool	NC function feed stop
feedStart	Bool	NC function feed start
spindleStop	Bool	NC function spindle stop
spindleStart	Bool	NC function spindle start
axisSelectionByHMI	Bool	NC function axis selection screen HMI PRO
mcsWcs	Bool	NC function switchover request MCS/WCS
reset	Bool	NC function reset
rapid	Bool	NC function rapid traverse
motionPlus	Bool	NC function motion plus direction
motionMinus	Bool	NC function motion minus direction
nextAxis	Bool	key next axis
previousAxis	Bool	key previous axis
handwheelActive	Bool	Handwheel selected
axes		
axis[1]	Bool	NC function axis selection
.		
.		
axis[31]	Bool	NC function axis selection
oemFunctions		
oemFunctions[1]	Bool	Unassigned function key for OEM
.		
.		
oemFunctions[100]	Bool	Unassigned function key for OEM

6.7 LTLP_FunctionModeTransline [FC548]

LTLP_FunctionModeTransline must be called in the cyclic sequence after **LTLP_OperatingMode**.
LTLP_FunctionModeTransline reads and writes signals according to TRANSLINE specifications, as subsequently described. It reads the signals from the **LTLP_DeviceInterfaceData**, **inputs.mppFunctions** area, and enters these into **LTLP_HMIProInterfaceData**.
Further, it reads the signals from **LTLP_HMIProInterfaceData** and writes these to the **LTLP_DeviceInterfaceData** area **outputs.mppFunctions**.

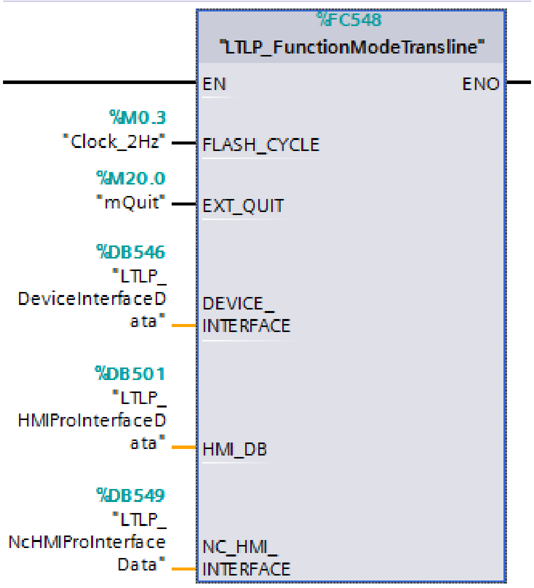
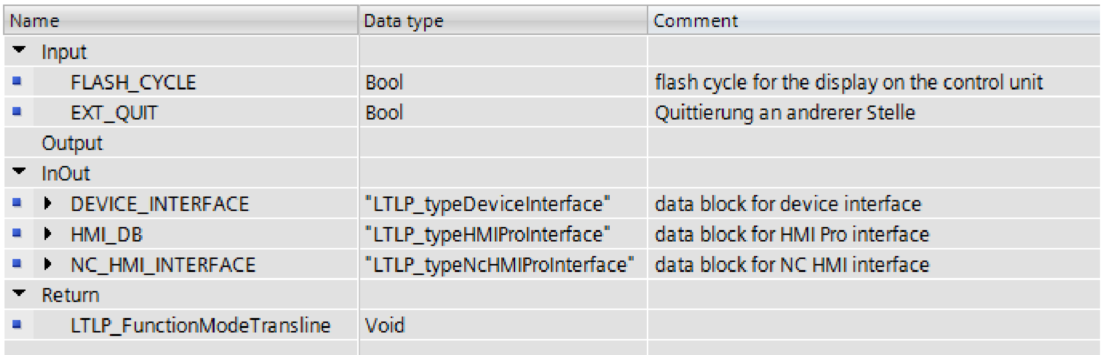


Fig. 6-25: LTLP_FunctionModeTransline [FC548]



Name	Data type	Comment
Input		
FLASH_CYCLE	Bool	flash cycle for the display on the control unit
EXT_QUIT	Bool	Quittierung an andrerer Stelle
Output		
InOut		
DEVICE_INTERFACE	"LTLP_typeDeviceInterface"	data block for device interface
HMI_DB	"LTLP_typeHMIProInterface"	data block for HMI Pro interface
NC_HMI_INTERFACE	"LTLP_typeNcHMIProInterface"	data block for NC HMI interface
Return		
LTLP_FunctionModeTransline	Void	

Fig. 6-26: LTLP_FunctionModeTransline parameter

Description of input parameters

Table 6-14: Input parameter LTLP_FunctionModeTransline

Parameter	Data type	Description
FLASH_CYCLE	Bool	Input for flashing flag clock. According to the TRANSLINE standard, here, the bit memory **clock_2Hz** (M0.3) is specified
EXT_ACKN	Bool	External acknowledgment -> The external acknowledgment is OR'ed with the Acknowledge key of the MPP when acknowledging via MPP.

Description of InOut parameters

Table 6-15: InOut parameter LTLP_FunctionModeTransline

Parameter	Data type	Description
DEVICE_ INTERFACE	LTLL_typeDevice Interface	The interface between the operator panels and the blocks for acquiring the operating modes and the TRANSLINE functions. This is the **LTLP_DeviceInterfaceData** [DB546] as default setting.
HMI_DB	LTLP_typeHMIPro Interface	Supplies the TRANSLINE functions within the HMI PRO data block. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
NC_HMI_INTERFACE	LTLP_typeNcHmiProInterface	Interface between HMI and the PLC. This is the **LTLP_NcHmiProInterfaceData** [DB549] as default setting.

Description of the return value

No value is returned.

Dependencies

None

Result of the LTLP_FunctionModeTransline [FC548]

The signals described below are only controlled by the block if the necessary signals in the **LTLP_DeviceInterfaceData** are provided by the operator panel FCs.

- **Automatic**
  When **automaticMode** is selected in the **LTLP_DeviceInterfaceData**, **inputs** area, then **LTLP_FunctionModeTransline** sets data bit **LTLP_HMIProInterfaceData.pAutomaticMode** [DBX76.1].
  With the automatic mode active, the user must acknowledge this bit with bit **LTLP_HMIProInterfaceData.automaticMode** [DBX86.1].
  Bit **automaticMode** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, flashes until bit **LTLP_HMIProInterfaceData.automaticMode** [DBX86.1] assumes a status of 1 - after which time, it is continuously activated.
  When bit **LTLP_HMIProInterfaceData.automaticMode** [DBX86.1] is set then in the header, the symbol ⇅ is activated.

- **Single mode**
  When **singleMode** is selected in the **LTLP_DeviceInterfaceData**, **inputs** area, then **LTLP_FunctionModeTransline** sets data bit **LTLP_HMIProInterfaceData.pSingleMode** [DBX76.2].
  With single mode active, the user must acknowledge this bit with bit **LTLP_HMIProInterfaceData.singleMode** [DBX86.2].
  Bit **singleMode** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, flashes until bit **LTLP_HMIProInterfaceData.singleMode** [DBX86.2] assumes a status of 1 - after which time, it is continuously activated.
  When bit **LTLP_HMIProInterfaceData.singleMode** [DBX86.2] is set then in the header, the symbol ⇅ is activated.

- **Single step**
  When **singleStepMode** is selected in the **LTLP_DeviceInterfaceData**, **inputs** area, then **LTLP_FunctionModeTransline** sets data bit **LTLP_HMIProInterfaceData.pSingleStepMode** [DBX76.3].
  With single step mode active, the user must acknowledge this bit with bit **LTLP_HMIProInterfaceData.singleStepMode** [DBX86.3].
  Bit **pSingleStepMode** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, flashes until bit **LTLP_HMIProInterfaceData.singleStepMode** [DBX86.3] assumes a status of 1 – after which time, it is continuously activated.
  When bit **LTLP_HMIProInterfaceData.singleStepMode** [DBX86.3] is set then in the header, the symbol ⇈ is activated.

- **Setup**
  When **setupMode** is selected in the **LTLP_DeviceInterfaceData**, **inputs** area, then **LTLP_FunctionModeTransline** sets data bit **LTLP_HMIProInterfaceData.pSetupMode** [DBX76.4].
  With setup active, the user must acknowledge this bit with bit **LTLP_HMIProInterfaceData.setupMode** [DBX86.4].
  Bit **setupMode** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, flashes until bit **LTLP_HMIProInterfaceData.setupMode** [DBX86.4] assumes a status of 1 - after which time, it is continuously activated.
  When bit **LTLP_HMIProInterfaceData.setupMode** [DBX86.4] is set then in the header, the symbol ∟○ is activated.

- **Machine/unit on**
  Using bit **machineOn** in the **LTLP_DeviceInterfaceData**, **inputs** area, with the **LTLP_FunctionModeTransline** bit **LTLP_HMIProInterfaceData.pMachineOn** [DBX77.7] is set, bit **LTLP_HMIProInterfaceData.pMachineOff** [DBX76.6] is reset. **machineOn** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area flashes.
  Once the machine has been switched on, the user sets bit **LTLP_HMIProInterfaceData.machineUnitOn** [DBX87.6].
  machineOn in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, now displays a continuous one

- **Machine/unit off**
  Using the bit **machineOff** in the **LTLP_DeviceInterfaceData**, **inputs** area, the **LTLP_FunctionModeTransline** bit **LTLP_HMIProInterfaceData.pMachineOff** [DBX76.6] is set, bit **LTLP_HMIProInterfaceData.pMachineOn** [DBX77.7] is reset.
  Once the machine has been switched off, the user sets bit **LTLP_HMIProInterfaceData.machineUnitOff** [DBX87.7].

- **Initial state**
  With bit **basicPosition** in the **LTLP_DeviceInterfaceData**, **inputs** area, **LTLP_HMIProInterfaceData.pBasicPosition** [DBX77.6] is set using the **LTLP_FunctionModeTransline** bit.
  The machine manufacturer sets bit **LTLP_HMIProInterfaceData.gotoBasicPosition** [DBX86.6] as response. By setting this bit, **basicPosition** in the **LTLP_DeviceInterfaceData**, **outputs** area starts to flash. Using bit **LTLP_HMIProInterfaceData.reachedBasicPosition** [DBX86.7], the continuous signal **basicPosition** signals that the machine is in basic position.
  Bits **LTLP_HMIProInterfaceData.pBasicPosition** and **LTLP_HMIProInterfaceData.gotoBasicPosition** must be reset.

- **Start/jog**
  The signal **start_jog** in the **LTLP_DeviceInterfaceData**, **inputs** area, is used for starting the **interlinked mode**, **single mode** or **single step mode**.
  Signal **startJog** controls bit **LTLP_HMIProInterfaceData.pStart** [DBX77.5].
  Signal **startJog** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, starts to flash by controlling bits
  LTLP_HMIProInterfaceData.readyAutomaticStart [DBX89.4] or
  LTLP_HMIProInterfaceData.selStopAfterEndOfCycle [DBX87.2]. If bit
  LTLP_HMIProInterfaceData.startJog [DBX87.2], is set, then the signal is permanently one:

- **Acknowledge fault**
  Signal **LTLP_HMIProInterfaceData.pGroupAcknowledge** [DBX78.0] can be reached using signal **acknowledgeFault** in the **LTLP_DeviceInterfaceData**, **inputs.mppFunctions** area, as well as using input parameter **EXT_ACKN**.
  When faults such as **LTLP_HMIProInterfaceData.completeMachineFault** [DBX88.5] or **LTLP_HMIProInterfaceData.generalFault** [DBX88.2] or **LTLP_HMIProInterfaceData.executionFault** [DBX88.3] occur, then the signal **acknowledgeFault** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, starts to flash.

- **Immediate stop**
  With the request Bit **immediateStop** in the **LTLP_DeviceInterfaceData**, **inputs** area, bit **LTLP_HMIProInterfaceData.pImmediateStop** [DBX81.1] is set.
  The user acknowledges this using bit
  LTLP_HMIProInterfaceData.immediateStop [DBX87.1].
  As long as this bit is equal to 1, then **immediateStop** in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions** area, is controlled.

- **Emergency stop**
  Using bit **emergencyStop** in the **LTLP_DeviceInterfaceData**, **inputs** area, bit **LTLP_HMIProInterfaceData.pEmergencyStop** [DBX77.3] is set to 1.

- **Protective doors**
  With request **lockRelProtectDoors** in the **LTLP_DeviceInterfaceData**, **inputs** area, bit **LTLP_HMIProInterfaceData.pLockReleaseProtectDoors** [DBX81.4] is set to 1.
  If it is possible to lock or unlock the protective doors, the user must acknowledge the selection with bit **LTLP_HMIProInterfaceData.selectProtectionDoors [DBX70.0]**. If bit **LTLP_HMIProInterfaceData.selectProtectionDoors** [DBX70.0] has a status of 1, then the **lockReleaseProtectDoors** signal in the **LTLP_DeviceInterfaceData**, **outputs.mppFunctions a**rea flashes.
  If the user sets bit **LTLP_HMIProInterfaceData.lockReleaseProtectDoors** [DBX95.6] to 1 (protective doors are released), then the lamp/LED is continually lit.
  After the execution – or when execution is not possible – the user must reset the **LTLP_HMIProInterfaceData.selectProtectionDoors** (DBX70.0) bit.

- **Stop after end of cycle**
  Using bit **stopAfterEndOfCycle** in the **LTLP_DeviceInterfaceData**, **inputs** area, then bit **LTLP_HMIProInterfaceData.pStopAfterEndOfCycle** [DBX81.0] is controlled.
  The user acknowledges this by controlling bit **LTLP_HMIProInterfaceData.selStopAfterEndOfCycle** [DBX87.2]. By controlling this bit up to controlling **LTLP_HMIProInterfaceData.reachedStopAfterEndOfCycle** [DBX87.3], then signal **stopAfterEndOfCycle** in the **LTLP_DeviceInterfaceData** area, flashes and then the signal is continuously one.

- **U keys screen selection/deselection**
  Bit **axisSelectionByHMI** in the **LTLP_DeviceInterfaceData**, **inputs** area, increments byte **LTLP_HMIProInterfaceData.axisSelectionScreen** [DBB82]. As a consequence HMI PRO is initiated to open or to close the U keys screen.

■

7

7 Direct key screens

The following screens are associated with the direct key screens:

Setup screen

The machine unit can be controlled manually in the Setup function mode using **Setup screen1** or **Setup screen2**. Several screens, type **Setup screen1** or **Setup screen2** can be integrated in HMI PRO. Each setup screen can manage 256 setup movements; a maximum of 6 movements can be displayed on each screen page. For special functions it is possible to display individual setup lines in the **Variables Layout** screen.

Power-up conditions

These are conditions that must be met in order to run the entire machine in automatic mode. Up to 48 switch-on conditions are displayed (maximum of 6 per page).

Cycle types

The various cycle types of the plant are activated and deactivated in this screen (e.g. single cycle, traversing without part, no-load traversing of machine etc.). Up to 16 cycle types are displayed (max. 6 per page).

Nut runner selection/deselection

For assembly lines, the screen permits individual nut runners to be selected and deselected using direct key functions on the operator panel. The following functions are available: Up to 80 nut runner spindles (maximum of 6 per screen) are displayed in 2 areas.

Group selection/deselection

For assembly lines, this screen permits nut runner groups to be selected and deselected. Up to 40 nut runner groups are displayed (maximum of 6 per page).

7.1 General information

Prerequisite for the use of the direct key screens, is that the direct keys are connected to the PLC. This can be achieved in a number of ways.

Using the MPP

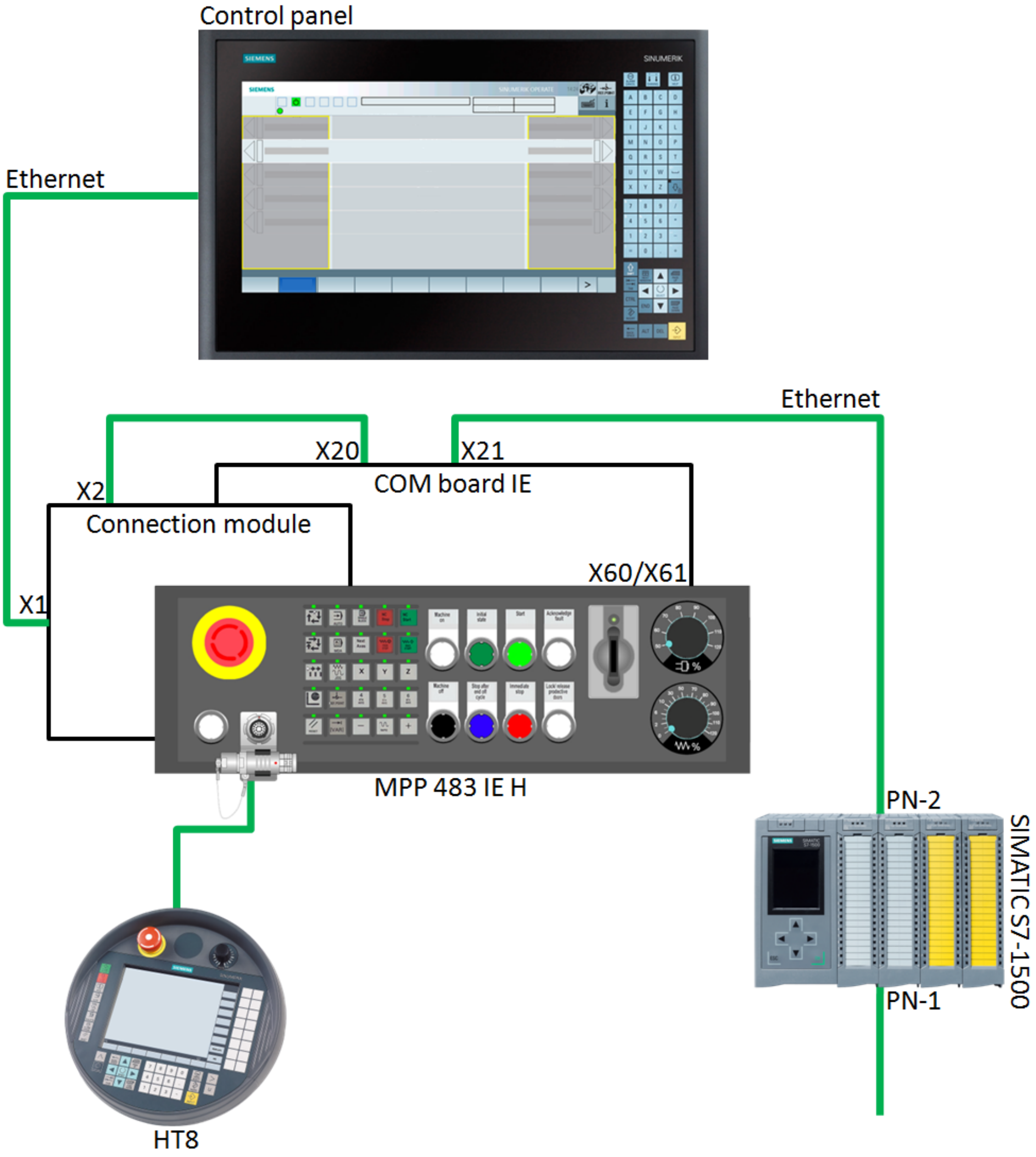


Fig. 7-1: Direct keys: OP015 Black connected to the MPP

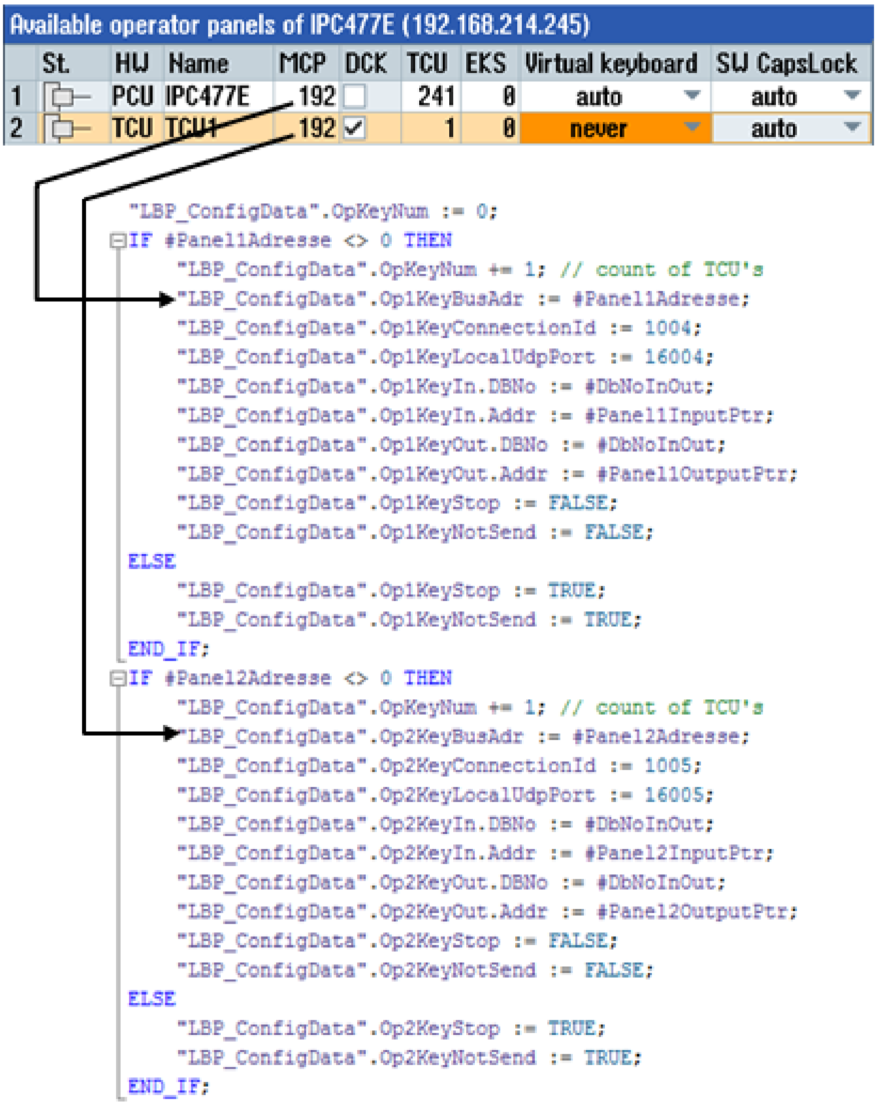
When the **LTLP_DirectKey** [FC511] is called, byte +4 of the specified input address is entered for **MINUS_KEYS** and byte +5 is entered for **PLUS_KEYS.**

Using the HT8

With the connection to the S7-1500, just like every TCU, HT8 is used with the DCK enabled. See: Using the OP015 Black

Using the OP015 Black, other TCUs + OPs or HT8

The direct keys of the TCU must be enabled for these devices.
When used, the modified key arrangement in the PLC must be taken into consideration (see also Chapter 4.2.3).



Available operator panels of IPC477E (192.168.214.245)

	St.	HW	Name	MCP	DCK	TCU	EKS	Virtual keyboard	SW CapsLock
1		PCU	IPC477E	192	□	241	0	auto	auto
2		TCU	TCU1	192	☑	1	0	never	auto

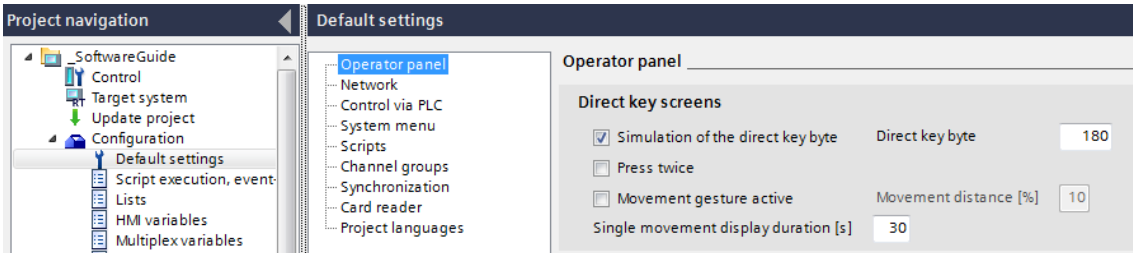
```
"LBP_ConfigData".OpKeyNum := 0;
IF #Panel1Adresse <> 0 THEN
    "LBP_ConfigData".OpKeyNum += 1; // count of TCU's
    "LBP_ConfigData".Op1KeyBusAdr := #Panel1Adresse;
    "LBP_ConfigData".Op1KeyConnectionId := 1004;
    "LBP_ConfigData".Op1KeyLocalUdpPort := 16004;
    "LBP_ConfigData".Op1KeyIn.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op1KeyIn.Addr := #Panel1InputPtr;
    "LBP_ConfigData".Op1KeyOut.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op1KeyOut.Addr := #Panel1OutputPtr;
    "LBP_ConfigData".Op1KeyStop := FALSE;
    "LBP_ConfigData".Op1KeyNotSend := FALSE;
ELSE
    "LBP_ConfigData".Op1KeyStop := TRUE;
    "LBP_ConfigData".Op1KeyNotSend := TRUE;
END_IF;
IF #Panel2Adresse <> 0 THEN
    "LBP_ConfigData".OpKeyNum += 1; // count of TCU's
    "LBP_ConfigData".Op2KeyBusAdr := #Panel2Adresse;
    "LBP_ConfigData".Op2KeyConnectionId := 1005;
    "LBP_ConfigData".Op2KeyLocalUdpPort := 16005;
    "LBP_ConfigData".Op2KeyIn.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op2KeyIn.Addr := #Panel2InputPtr;
    "LBP_ConfigData".Op2KeyOut.DBNo := #DbNoInOut;
    "LBP_ConfigData".Op2KeyOut.Addr := #Panel2OutputPtr;
    "LBP_ConfigData".Op2KeyStop := FALSE;
    "LBP_ConfigData".Op2KeyNotSend := FALSE;
ELSE
    "LBP_ConfigData".Op2KeyStop := TRUE;
    "LBP_ConfigData".Op2KeyNotSend := TRUE;
END_IF;
```

Fig. 7-2: Direct keys: Using the HT8, OP08T, OP015 Black, TCU + OP

When the **LTLP_DirectKey** [FC511] is called, byte +0 of the specified start address (**LBP_ConfigData.Op[x]KeyIn**) is entered for **MINUS_KEYS** and byte +1 is entered for **PLUS_KEYS.**

Using the IPC477

In the IPC477, there is no support for direct keys – this is the reason that simulation by HMI PRO must be used. Make the settings in HMI PRO CS: **Project navigation ▶ Configuration ▶ Default settings ▶ Operator panel** under **Direct key screens**.



Operator panel

Direct key screens

☑ Simulation of the direct key byte   Direct key byte 180

□ Press twice

□ Movement gesture active   Movement distance [%] 10

Single movement display duration [s] 30

Fig. 7-3: Direct keys: Using the IPC477

When the **LTLP_DirectKey** [FC511] is called, byte +0 of the specified input address in the direct key byte is entered into **MINUS_KEYS** and byte +1 is entered into **PLUS_KEYS**.

7.2 General information Configuring direct key screens

Additional information on configuring the direct key screens is provided in the online help for HMI PRO CS.

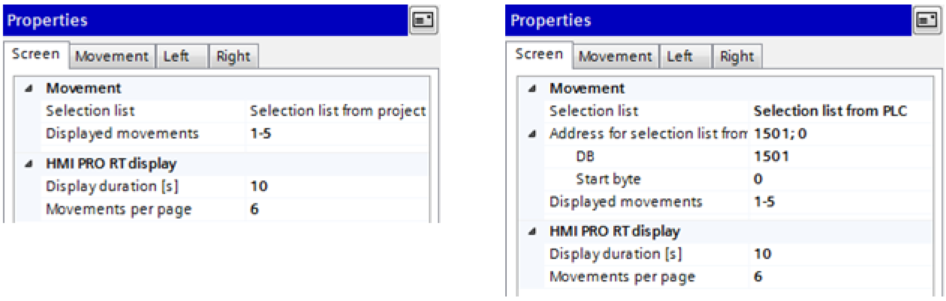
The configuration and sequence are subsequently displayed using a setup function (setup screen 1).

Inserting a setup screen into the function key assignment

This is realized in HMI PRO CS via **Project navigation ▸Function key assignment**. From the **Screen scope ▸Direct keys**, drag **setup screen 1** and drop on the required key in the function key assignment. You can define the designation of the movement group using the right-hand mouse key and the context menu **Change function key labeling.**

Configuring dialog setup screen

Via **Project navigation ▸Function key assignment ▸Setup. 1** or by double-clicking on the function key, open the configuration dialog for direct key screens. On the right-hand side below **Properties ▸Screen** you define how you wish to arrange setup movements.



Movement	
Selection list	Selection list from project
Displayed movements	1-5
HMI PRO RT display	
Display duration [s]	10
Movements per page	6

Movement	
Selection list	**Selection list from PLC**
Address for selection list fron	**1501; 0**
DB	**1501**
Start byte	**0**
Displayed movements	**1-5**
HMI PRO RT display	
Display duration [s]	**10**
Movements per page	**6**

- **Movement**

Selection list from project
Displayed movements are valid for HMI PRO CS and for Runtime

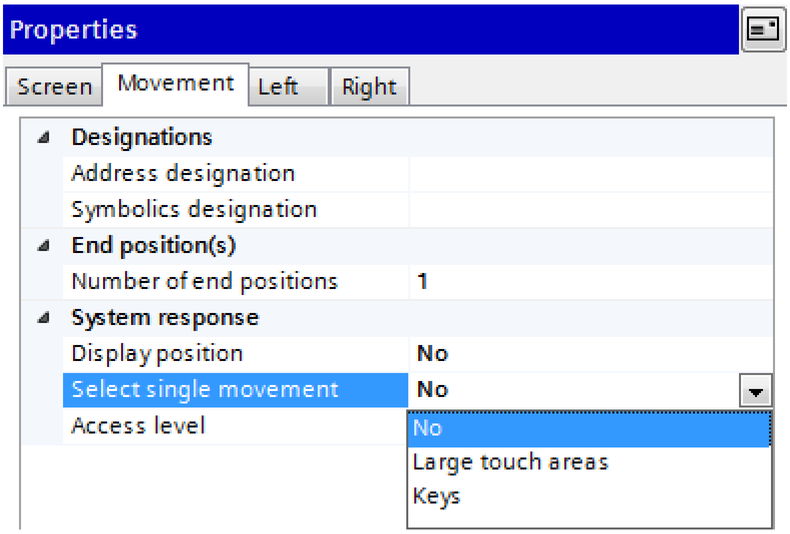
Selection list from PLC
Displayed movements are valid for HMI PRO CS
Selection list from PLC is valid for Runtime

- **HMI PRO RT display**

Display duration: Is valid for the address designation display. It will be switched back to symbols after expiration of the time interval.
Movements per page: Must be configured to 6 for key panels

Fig. 7-4: Configuring setup screens in HMI PRO CS: Properties ▸Screen

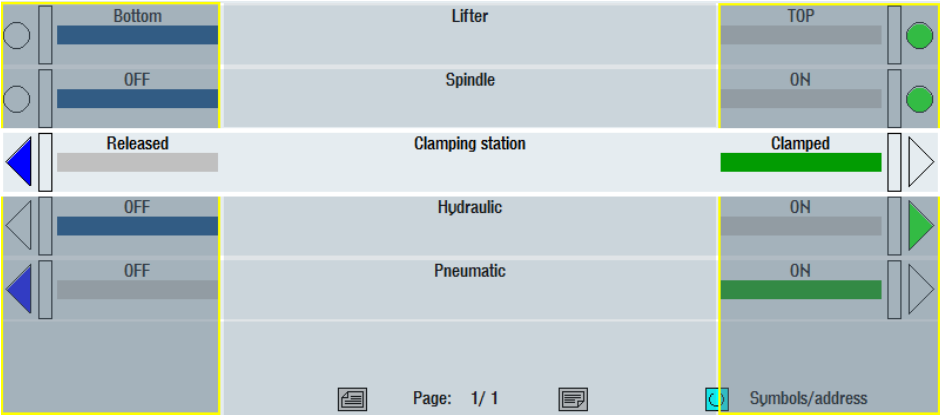
General properties such as name of the movement, symbolic designation and position are entered under **Properties ▸Movement**. For **symbolic** and **address designation** it makes sense to enter the same designation. Setup functions of additional properties regarding movement selection can be assigned via **Select single movement**. These features are available from HMI PRO CS Version ≥ 8.1



Designations	
Address designation	
Symbolics designation	
End position(s)	
Number of end positions	1
System response	
Display position	No
Select single movement	No
Access level	

Fig. 7-5: Configuring setup screens in HMI PRO CS: Movement properties

- Select single movement: Large touch areas

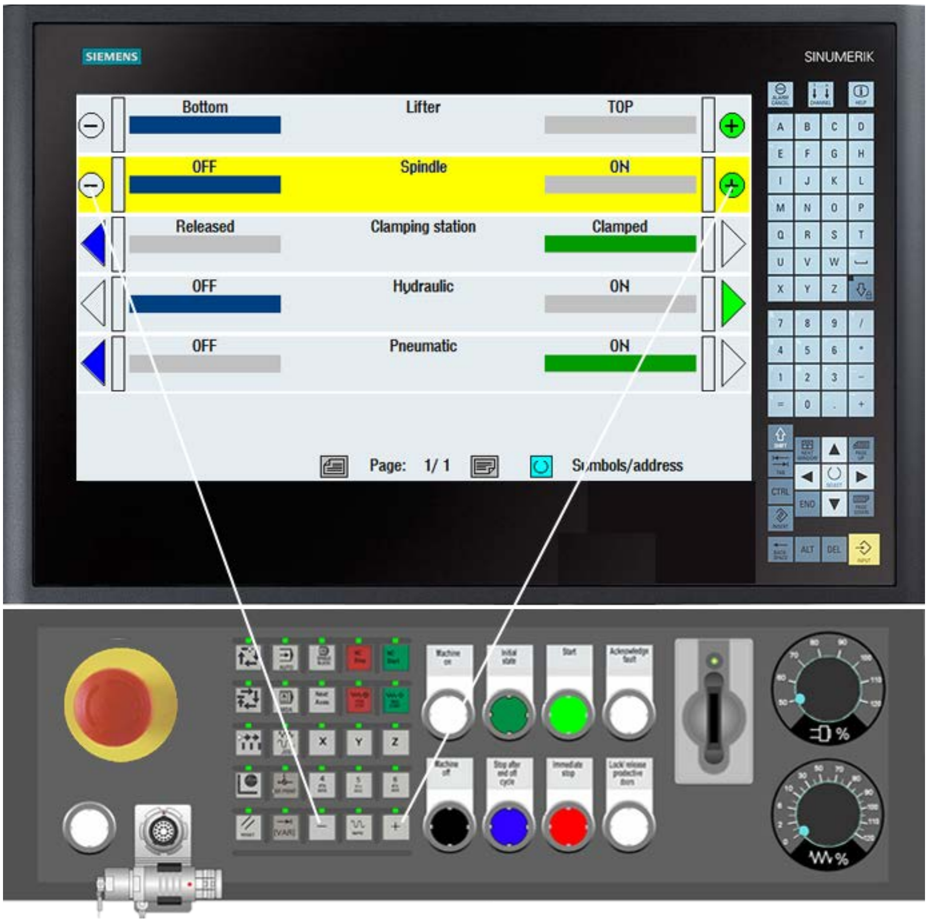


Bottom	Lifter	TOP
OFF	Spindle	ON
Released	Clamping station	Clamped
OFF	Hydraulic	ON
OFF	Pneumatic	ON

Page: 1/ 1 Symbols/address

Fig. 7-6: Large touch areas: Display in HMI PRO sl RT

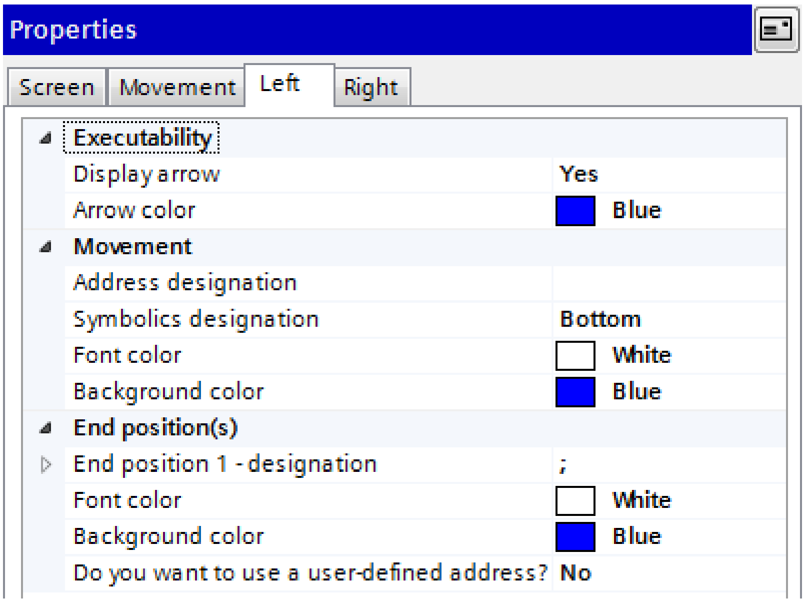
- Select single movement: Keys
  Using the **Keys** configuring, a higher degree of reliability can be achieved that movement is actually initiated using hardware keys.



Bottom	Lifter	TOP
OFF	Spindle	ON
Released	Clamping station	Clamped
OFF	Hydraulic	ON
OFF	Pneumatic	ON

Page: 1/ 1 Symbols/address

The properties for movement active, end positions etc. are defined under **Properties ▸ Left** or **Right**.



Properties

Screen | Movement | Left | Right

◢ Executability	
Display arrow	Yes
Arrow color	🟦 Blue
◢ Movement	
Address designation	
Symbolics designation	Bottom
Font color	⬜ White
Background color	🟦 Blue
◢ End position(s)	
▷ End position 1 - designation	;
Font color	⬜ White
Background color	🟦 Blue
Do you want to use a user-defined address?	No

Fig. 7-7: Configuring setup screens in HMI PRO CS: Properties, left

Triggering a setup function

If one of the vertical direct keys is pressed, or the touch surface for touch HMI devices, then this is signaled to HMI PRO and in the controller using the direct key function of the operator panels, a bit is activated. In the controller, the **LTLP_DirectKey** [FC511] compares the signals that are provided by HMI PRO in the data bar (**LTLP_HMIProInterfaceData**) with the direct key signals. If this comparison is successful within the time which is set as monitoring time in HMI PRO CS under **Project navigation ▸ Configuration ▸ Default settings ▸ Operator panel**, then the following information is set for the movement: Direction bit **LTLP_HMIProInterfaceData.pPlus** for movement towards the initial position (+) - or **LTLP_HMIProInterfaceData.pMinus** for movement in the opposite direction (-). The function code (BCD code of the setup movement) of the pressed key is written to **LTLP_HMIProInterfaceData.noFunctionLow** [DBB79] and **LTLP_HMIProInterfaceData.noFunctionHigh** [DBB80]. Function **LTLP_GetDirectKeyMoveNo** [FC512] decodes the function code in a bit in the **LTLP_GetDirectKeyMoveNoData** [DB512].

7.3 Function sequence for direct key screens

The supply of the direct key screens is described using the **Setup** screen. A solution using LAD/FBD or STL is described in the following section of the block description. This solution can be used for the **Cycle type, Power-on conditions, Nut runner selection/deselection and Group selection/deselection** screens.

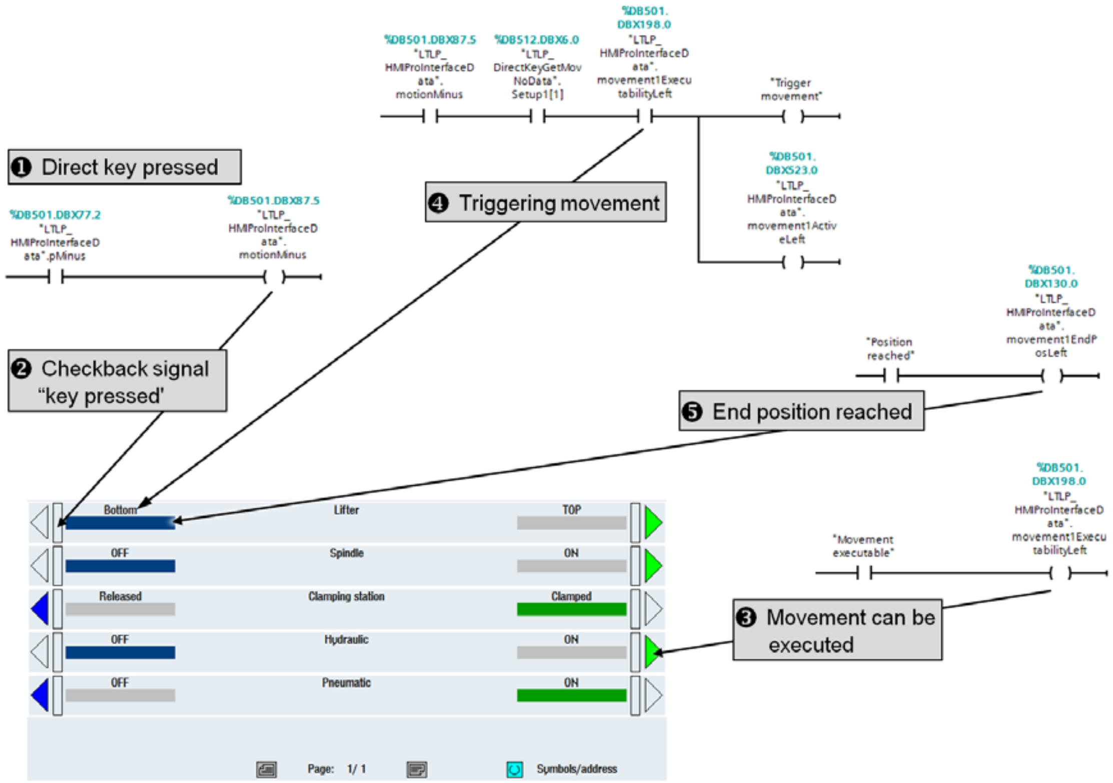

%DB501.DBX87.5
"LTLP_HMIProInterfaceData".motionMinus
%DB512.DBX6.0
"LTLP_DirectKeyGetMovNoData".Setup1[1]
%DB501.DBX198.0
"LTLP_HMIProInterfaceData".movement1ExecutabilityLeft
"Trigger movement"
%DB501.DBX523.0
"LTLP_HMIProInterfaceData".movement1ActiveLeft
❶ Direct key pressed
%DB501.DBX77.2
"LTLP_HMIProInterfaceData".pMinus
%DB501.DBX87.5
"LTLP_HMIProInterfaceData".motionMinus
❹ Triggering movement
❷ Checkback signal "key pressed'
%DB501.DBX130.0
"LTLP_HMIProInterfaceData".movement1EndPosLeft
"Position reached"
❺ End position reached
%DB501.DBX198.0
"LTLP_HMIProInterfaceData".movement1ExecutabilityLeft
"Movement executable"
❸ Movement can be executed
Bottom
Lifter
TOP
OFF
Spindle
ON
Released
Clamping station
Clamped
OFF
Hydraulic
ON
OFF
Pneumatic
ON
Page: 1/ 1
Symbols/address

Fig. 7-8: Supply of the direct key screens (example for the 1st direct key left '-')

Information about keyboard assignment (Fig. 7-8 ❶ ❷)

When pressing a direct key (in the screen above 1st direct key, movement to the left in the '-' direction), then **LTLP_DirectKey** [FC511] sets bit **LTLP_HMIProInterfaceData.pPlus** [DBX77.1] or **LTLP_HMIProInterfaceData.pMinus** [DBX77.2]. As feedback signal, the user must form bits **LTLP_HMIProInterfaceData.motionPlus** [DBX87.4] or **LTLP_HMIProInterfaceData.motionMinus** [DBX87.5]. As 'Key pressed' feedback, the rectangle is highlighted in color (❷).

Information as to whether movement can be executed (Fig. 7-8 ❸)

The arrow symbol represents the enable for the execution of the movement. The enable is realized by the machine manufacturer using bit **LTLP_HMIProInterfaceData.movement*X*ExecutabilityLeft** for movement to the left or **LTLP_HMIProInterfaceData.movement*X*ExecutabilityRight** for movement to the right [from DBX198.0]. The "Movement can be executed" display has to be updated in the Setup mode.
X is the number of the movement (1-256)

Activate movement / movement is being executed (Fig. 7-8 ❹)

The actual execution of the movement depends on:

- movement bit **LTLP_HMIProInterfaceData.pPlus** [DBX77.1] or **LTLP_HMIProInterfaceData.pMinus** [DBX77.2],
- bit **LTLP_DirectKeyGetMovNoData.Setup1[*X*]**, which controls the **LTLP_DirectKeyGetMovNo** [FC512],
- bit **LTLP_HMIProInterfaceData.movement*X*ExecutabilityLeft** or **LTLP_HMIProInterfaceData.movement*X*ExecutabilityRight** [from DBX198.0]

These bits control the bit memories or output for initiating motion as well as bit **LTLP_HMIProInterfaceData.movement*X*ActiveLeft** or **LTLP_HMIProInterfaceData.movement*X*ActiveRight** [from DBX523.0]. If the movement is executed, this movement field is highlighted in color.
X is the number of the movement (1-256)

Information end position reached (Fig. 7-8 ❺)

Once the end position has been reached, the user must supply bit **LTLP_HMIProInterfaceData.movement*X*EndPosLeft** or **LTLP_HMIProInterfaceData.movement*X*EndPosRight** [from DBX130.0]. The field below the movement field is highlighted in color when the end position is reached.
X is the number of the movement (1-256)

7.4 Call sequence in the controller

The standard blocks that are required to connect the direct keys, are stored in the TRANSLINE library under types and under copy templates in the **Common** and **Directkeys** folders.

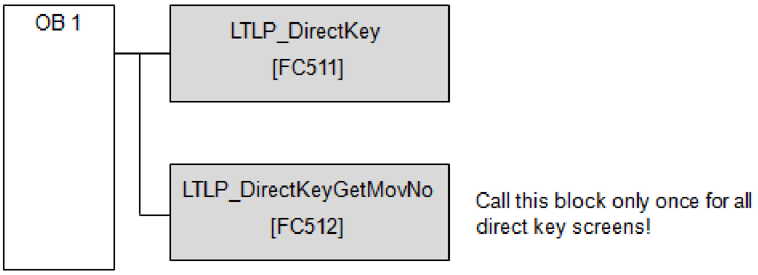


Fig. 7-9: Call sequence for direct key screens in the controller

Block **LTLP_DirectKey** [FC511] must be called once. The formal parameters are supplied depending on the active operator panel. See also Chapter 5.1.

7.4.1 Supplying the LTLP_DirectKey [FC511]

Attention

LTLP_DirectKey [FC511] must be used for direct key screens.

The function block represents the link between the direct keys of the operator panel and the data bar of HMI PRO **LTLP_HMIProInterfaceData**. It is required for all screens that use direct keys.
In conjunction with HMI PRO, this block determines the function code for the appropriate selected movement or direct key function.

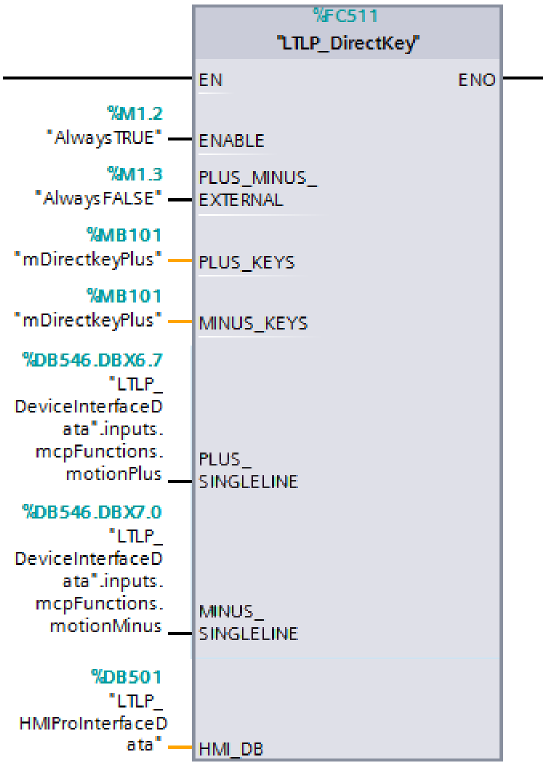
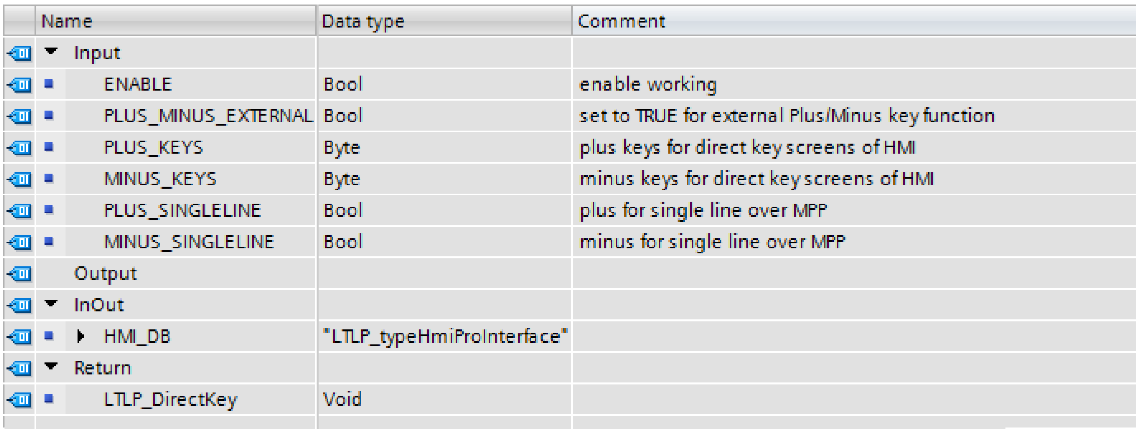


Fig. 7-10: LTLP_DirectKey [FC511]



Name	Data type	Comment
Input		
ENABLE	Bool	enable working
PLUS_MINUS_EXTERNAL	Bool	set to TRUE for external Plus/Minus key function
PLUS_KEYS	Byte	plus keys for direct key screens of HMI
MINUS_KEYS	Byte	minus keys for direct key screens of HMI
PLUS_SINGLELINE	Bool	plus for single line over MPP
MINUS_SINGLELINE	Bool	minus for single line over MPP
Output		
InOut		
HMI_DB	"LTLP_typeHmiProInterface"	
Return		
LTLP_DirectKey	Void	

Fig. 7-11: LTLP_DirectKey parameter

Description of input parameters

Table 7-1: Input parameters LTLP_DirectKey

Parameter	Data type	Description
ENABLE	Bool	Machining enable
PLUS_MINUS_EXTERNAL	Bool	TRUE: When connecting the plus/minus function to external plus/minus keys
PLUS_KEYS	Byte	Input byte for right-hand direct keys on the operator panel
MINUS_KEYS	Byte	Input byte for left-hand direct keys on the operator panel
PLUS_SINGLELINE	Bool	TRUE: selected single line is controlled via the plus key of an MPP (from block version 1.1, HMI PRO CS V08.00.00.04, HMI PRO sl RT V08.00.00.04)
MINUS_SINGLELINE	Bool	TRUE: selected single line is controlled via the minus key of an MPP (from block version 1.1, HMI PRO CS V08.00.00.04, HMI PRO sl RT V08.00.00.04)

Description of InOut parameters

Table 7-2: InOut parameters LTLP_DirectKey

Parameter	Data type	Description
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. Data block **LTLP_HMIProInterfaceData** [DB501] should be used here as variable.

Description of the return value

No value is returned.

Data interface

There is a transfer area for the direct keys in the **LTLP_HMIProInterfaceData** data bar. This area is used to transfer the pressed direct key as well as the bits for the +/- movement to the operator panels and calculate the actual line number. This line number is signaled to the user via DBW79 and a lock bit is set.
This function can be deactivated with parameter **PLUS_MINUS_EXTERN**=TRUE in order to connect the standard plus/minus function to external keys. If **PLUS_MINUS_EXTERNAL** is set to FALSE, then the plus/minus keys are realized with HMI PRO and the **LTLP_DirectKey**.

The following signals of the **LTLP_HMIProInterfaceData** must be used by the user as output parameters:

- **LTLP_HMIProInterfaceData.pPlus** [DBX77.1] movement in the plus direction selected.
- **LTLP_HMIProInterfaceData.pMinus** [DBX77.2] movement in the minus direction selected.

Movement numbers **LTLP_HMIProInterfaceData.noFunctionLow** [DBB79] and **LTLP_HMIProInterfaceData.noFunctionHigh** [DBB80] are only valid for status 1 at signals **LTLP_HMIProInterfaceData.pPlus** and **LTLP_HMIProInterfaceData.pMinus**.

Note

Please note that the movement number must be interpreted as word, is BCD-coded and low/high are interchanged.

The programming resources that implement the functions are only allocated after the **LTLP_DirectKey**. The use of GRAPH is described in Chapter 9.

7.4.2 Supplying LTLP_GetDirectKeyMoveNo [FC512]

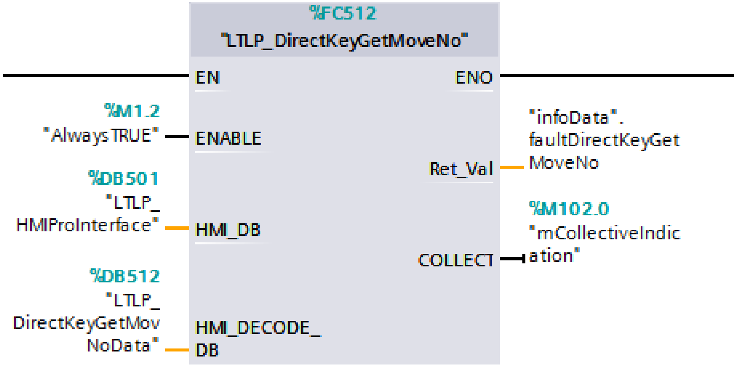
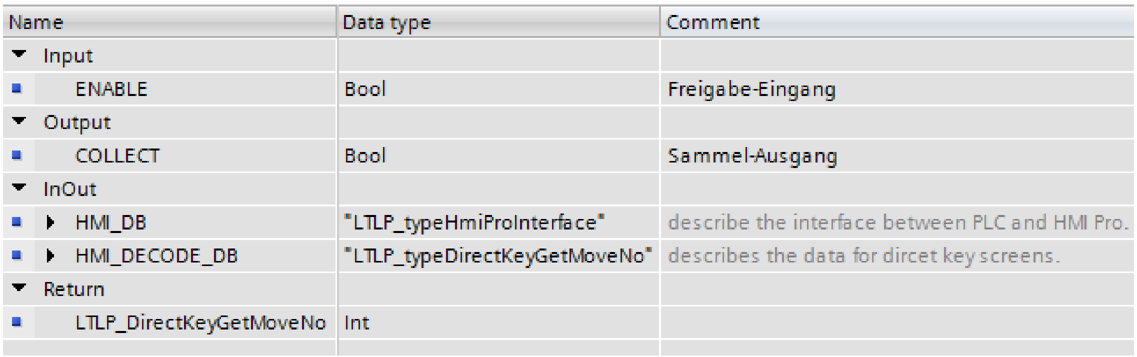


Fig. 7-12: LTLP_GetDirectKeyMoveNo [FC512]



Name	Data type	Comment
Input		
ENABLE	Bool	Freigabe-Eingang
Output		
COLLECT	Bool	Sammel-Ausgang
InOut		
HMI_DB	"LTLP_typeHmiProInterface"	describe the interface between PLC and HMI Pro.
HMI_DECODE_DB	"LTLP_typeDirectKeyGetMoveNo"	describes the data for dircet key screens.
Return		
LTLP_DirectKeyGetMoveNo	Int	

Fig. 7-13: LTLP_GetDirectKeyMoveNo parameter

Application

For the direct key functions, this block is used to convert the function number from HMI PRO **LTLP_HMIProInterfaceData.noFunctionLow** [DBB79] and **LTLP_HMIProInterfaceData.noFunctionHigh** [DBB80] into a bit.

Description of input parameters

Table 7-3: Input parameters LTLP_GetDirectKeyMoveNo

Parameter	Data type	Description
ENABLE	Bool	Machining enable The block can always be enabled.

Description of output parameters

Table 7-4: Output parameters LTLP_GetDirectKeyMoveNo

Parameter	Data type	Description
COLLECT	Bool	The group output is set if a bit is set in the **LTLP_DirectKeyGetMovNoData**.

Description of InOut parameters

Table 7-5: InOut parameters LTLP_GetDirectKeyMoveNo

Parameter	Data type	Description
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
HMI_DECODE_ DB	LTLP_typeDirectK eyGetMoveNo	Bit interface between the user and the direct keys Data block **LTLP_DirectKeyGetMovNoData** [DB512] should be used here as variable.

Description of the return value

No value is returned.

Data interface

From the data bar **LTLP_HMIProInterfaceData,** the movement number comprising **LTLP_HMIProInterfaceData.noFunctionLow** [DBB79] and **LTLP_HMIProInterfaceData.noFunctionHigh** [DBB80] is used. From this number, a data bit in data block **LTLP_DirectKeyGetMovNoData** is determined. This means from the value of the 1st movement of the setup screen 1 (BCD code 101) data bit **LTLP_DirectKeyGetMovNoData.Setup**1[1] is obtained.

■

8

8 PDA functions

Under the general term, PDA functions, for HMI PRO, the following screens are involved:

- Cycle times
- Workpiece count
- Utilization
- Shift model
- Tool life overview
- Tool wear

For the **Cycle times, Workpiece count** and **Lifetime overview** screens the required standard blocks are stored in the TRANSLINE library in the **ProductionDataAcquisition** folder.

The **Machine utilization** and **Shift model** screens do not require any blocks.

The **tool wear** screen is a screen from the SINUMERIK standard and is described the.

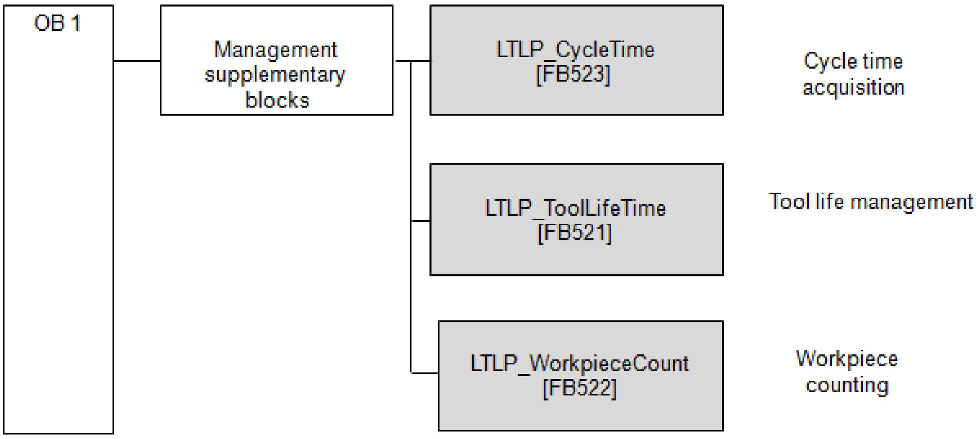
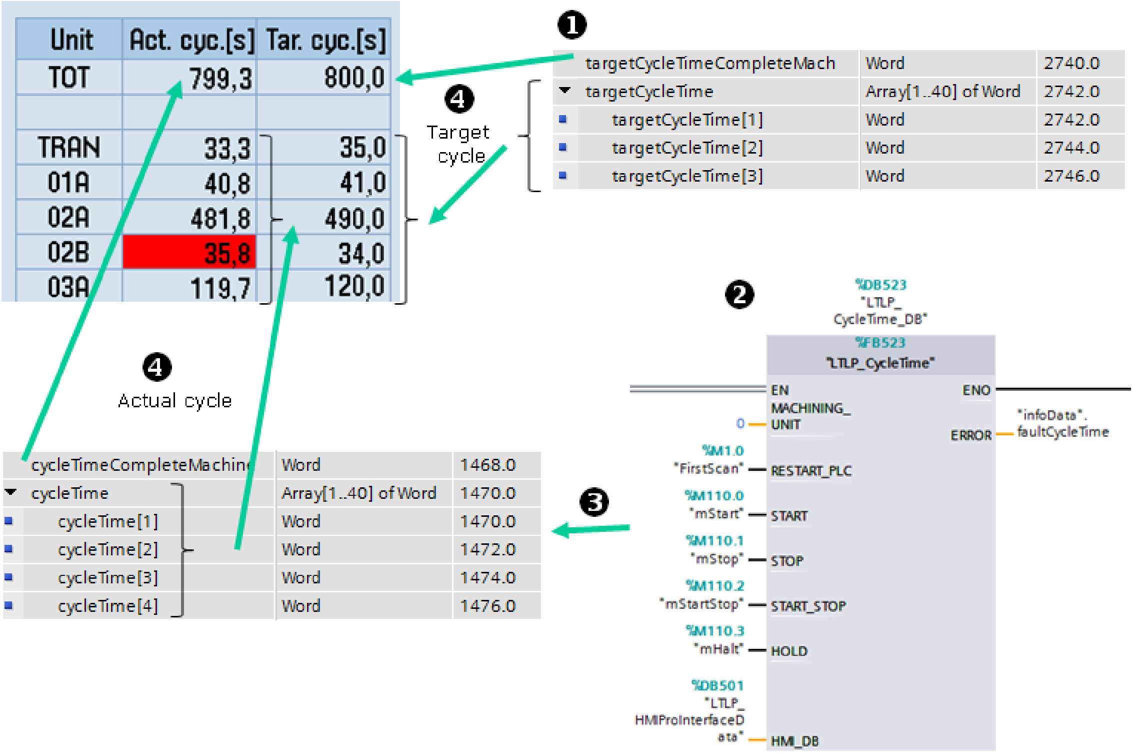


Fig. 8-1: Call sequence of PDA functions

8.1 Cycle times

The **LTLP_CycleTime** [FB523] is required for the HMI PRO **cycle times** screen (see also online help of HMI PRO CS).
It supplies HMI PRO in the **LTLP_HMIProInterfaceData** with the measured actual cycle times. It must be called once for the entire machine and once for each displayed machining unit. A machine can have a maximum of 40 machining units (restriction as a result of the **LTLP_HMIProInterfaceData**).
Cycle times are acquired using **LTLP_CycleTime**. They are saved as multiples of 0.1 seconds in the **LTLP_HMIProInterfaceData**.



Unit	Act. cyc.[s]	Tar. cyc.[s]
TOT	799,3	800,0
TRAN	33,3	35,0
01A	40,8	41,0
02A	481,8	490,0
02B	35,8	34,0
03A	119,7	120,0

targetCycleTimeCompleteMach	Word	2740.0
▼ targetCycleTime	Array[1..40] of Word	2742.0
▪ targetCycleTime[1]	Word	2742.0
▪ targetCycleTime[2]	Word	2744.0
▪ targetCycleTime[3]	Word	2746.0

cycleTimeCompleteMaching	Word	1468.0
▼ cycleTime	Array[1..40] of Word	1470.0
▪ cycleTime[1]	Word	1470.0
▪ cycleTime[2]	Word	1472.0
▪ cycleTime[3]	Word	1474.0
▪ cycleTime[4]	Word	1476.0

Fig. 8-2: Interrelationship between LTLP_CycleTime and HMI PRO

Fig. 8-2 → ❶
The user must enter the target cycle time for the entire machine and the target cycle times for the machining units in the **LTLP_HMIProInterfaceData** in 1/10 seconds ("**LTLP_HMIProInterfaceData".targetCycleTimeCompleteMach**, "**LTLP_HMIProInterfaceData".targetCycleTime[*X*]**).

Fig. 8-2 → ❷
The **LTLP_CycleTime** block must be called once for the entire machine (parameter MACHINING_UNIT = 0) and once for each displayed machining unit (parameter MACHINING_UNIT = ***X***).

Fig. 8-2 → ❸
Depending on parameter MACHINING_UNIT, the FB writes the actual cycle times of the complete machine or the actual cycle times of machining units to **LTLP_HMIProInterfaceData**.
("**LTLP_HMIProInterfaceData".targetCycleTimeCompleteMach** or "**LTLP_HMIProInterfaceData".targetCycleTime [*X*]**).

Fig. 8-2 → ❹
HMI PRO displays the values from **LTLP_HMIProInterfaceData** in the Cycle times screen. If the actual value exceeds the specified setpoint, its color turns red.

X is the number of the machining unit (1-40)

LTLP_CycleTime [FB523]

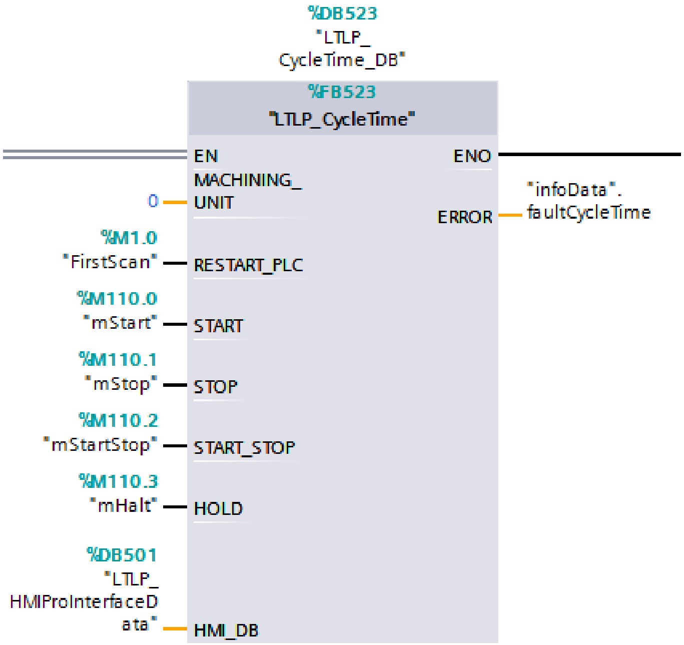
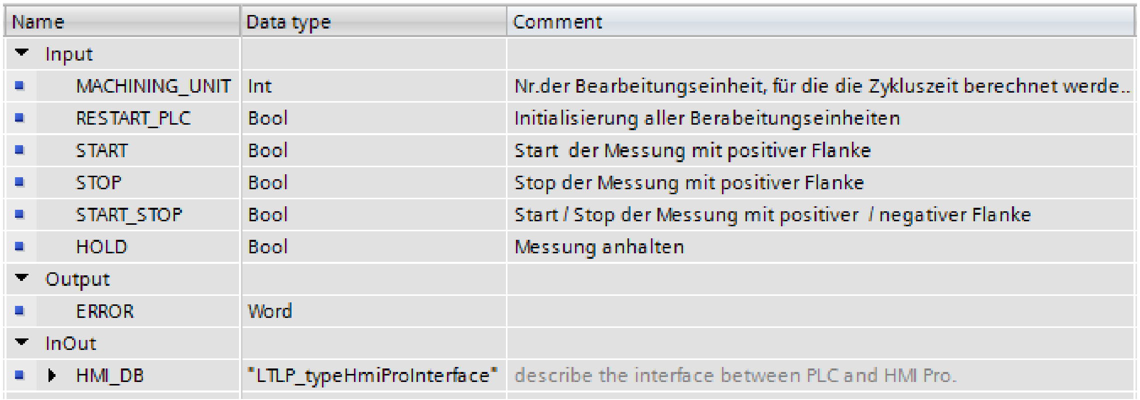


Fig. 8-3: LTLP_CycleTime [FB523]



Name	Data type	Comment
▼ Input		
■ MACHINING_UNIT	Int	Nr.der Bearbeitungseinheit, für die die Zykluszeit berechnet werde..
■ RESTART_PLC	Bool	Initialisierung aller Berabeitungseinheiten
■ START	Bool	Start der Messung mit positiver Flanke
■ STOP	Bool	Stop der Messung mit positiver Flanke
■ START_STOP	Bool	Start / Stop der Messung mit positiver / negativer Flanke
■ HOLD	Bool	Messung anhalten
▼ Output		
■ ERROR	Word	
▼ InOut		
■ ▶ HMI_DB	"LTLP_typeHmiProInterface"	describe the interface between PLC and HMI Pro.

Fig. 8-4: LTLP_CycleTime parameter

Application

This block is required for the **Cycle times** HMI PRO screen.

Data interface

The actual cycle times are supplied in data area **LTLP_HMIProInterfaceData** ("**LTLP_HMIProInterfaceData".targetCycleTimeCompleteMach** to "**LTLP_HMIProInterfaceData".targetCycleTime[*40*]**).

Description of input parameters

Table 8-1: Input parameter LTLP_CycleTime

Parameter	Data type	Description
MACHINING_UNIT	Int	Unit/machine for which the cycle time should be calculated: 1 to 40: Number of the machining unit 0: Complete machine
RESTART_PLC	Bool	Initializes all machining units of the instance; this is applied when the CPU reboots For TRANSLINE: Bit memory FirstScan (M1.0)
START	Bool	If the time is acquired using START/STOP, then the formal START_STOP parameter must be assigned **AlwaysFalse.** Acquisition is started with a positive edge at START. Another positive edge without a previous STOP causes the cycle time acquisition to restart. The time that has accumulated up until then is discarded.

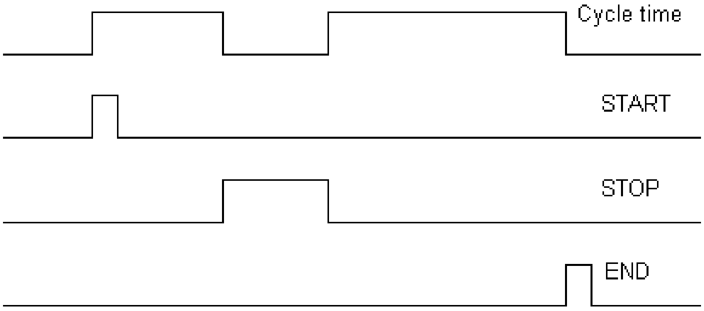
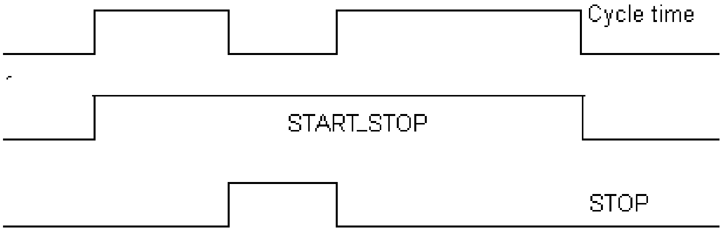
Parameter	Data type	Description
STOP	Bool	If the time is acquired using START/STOP, then the formal START_STOP parameter must be assigned **AlwaysFalse.** The acquisition is stopped with a positive edge if the acquisition was previously started using "START". An additional positive edge without a previous START has no effect. The acquisition of the cycle times is started by the positive edge at "START_END". The measurement is terminated by a negative edge at "START_STOP". Measurement is only performed when HOLD is not set.  Fig. 8-5: Cycle time "START" and "STOP" with HOLD
START_END	Bool	The acquisition is started with a positive edge unless an acquisition is already running via START. The acquisition is stopped on a negative edge. If the time is acquired using the START_STOP parameter, then the START and STOP parameter must be supplied with **AlwaysFalse.**  Fig. 8-6: Cycle time "START_STOP" with HOLD
HOLD	Bool	The acquisition can be interrupted as many times as required by setting this parameter to TRUE. The acquisition of the cycle times is started by the positive edge at "START_END". The measurement is terminated by a negative edge at "START_STOP". Measurement is only performed when HOLD is not set.

Table 8-2: Output parameter LTLP_CycleTime

Parameter	Data type	Description
ERROR	Word	0x0000 No error 0x8001 Station index incorrect

Table 8-3: InOut parameter LTLP_CycleTime

Parameter	Data type	Description
HMI_DB	LTLP_typeHmi ProInterface	Interface between HMI PRO and PLC user program. As default setting, this is the **LTLP_HMIProInterface** [DB501] from the copy template.

Description of the return value

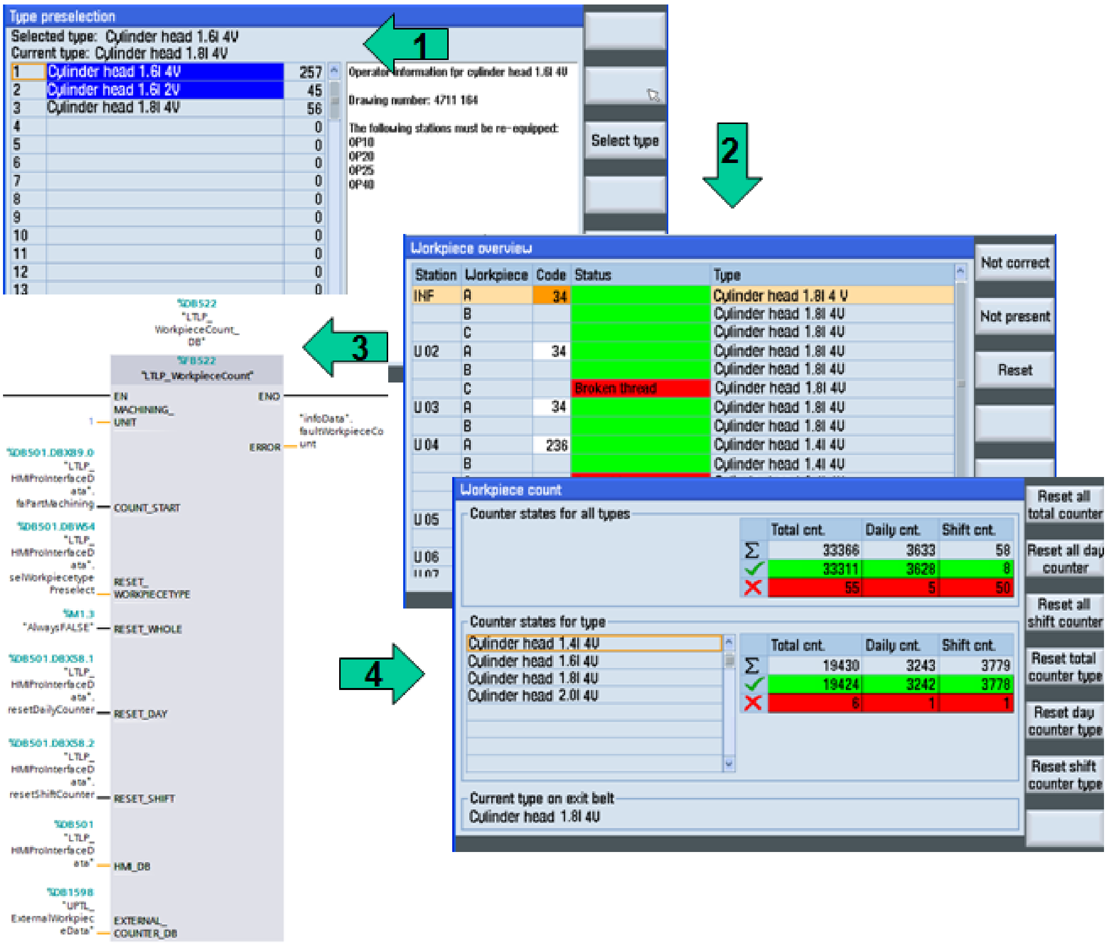
No value is returned.

⚠ **Attention**

Parameters that are not connected **must** be set to **AlwaysFalse.**
If multiple calls are made, the block must always be supplied the same instance data block.

8.2 Type preselection, workpiece overview and part counter

These screens show the workpiece production workflow in the automatic mode. The workpiece types are defined or displayed when entering using type preselection. The workpiece overview is used to display or influence the machining of workpieces as they pass through the machine. The workpieces are counted at the final station.



Selected type: Cylinder head 1.6l 4V
Current type: Cylinder head 1.8l 4V

1	Cylinder head 1.6l 4V	257
2	Cylinder head 1.6l 2V	45
3	Cylinder head 1.8l 4V	56
4		0
5		0
6		0
7		0
8		0
9		0
10		0
11		0
12		0
13		0

Operator information for cylinder head 1.6l 4V

Drawing number: 4711 164

The following stations must be re-equipped:
OP10
OP20
OP25
OP40

Station	Workpiece	Code	Status	Type
INF	A	34		Cylinder head 1.8l 4 V
	B			Cylinder head 1.8l 4V
	C			Cylinder head 1.8l 4V
U 02	A	34		Cylinder head 1.8l 4V
	B			Cylinder head 1.8l 4V
	C		Broken thread	Cylinder head 1.8l 4V
U 03	A	34		Cylinder head 1.8l 4V
	B			Cylinder head 1.8l 4V
U 04	A	236		Cylinder head 1.4l 4V
	B			Cylinder head 1.4l 4V
U 05				
U 06				
U 07				

Counter states for all types

	Total cnt.	Daily cnt.	Shift cnt.
Σ	33366	3633	58
✓	33311	3628	8
✗	55	5	50

Counter states for type

Cylinder head 1.4l 4V
Cylinder head 1.6l 4V
Cylinder head 1.8l 4V
Cylinder head 2.0l 4V

	Total cnt.	Daily cnt.	Shift cnt.
Σ	19430	3243	3779
✓	19424	3242	3778
✗	6	1	1

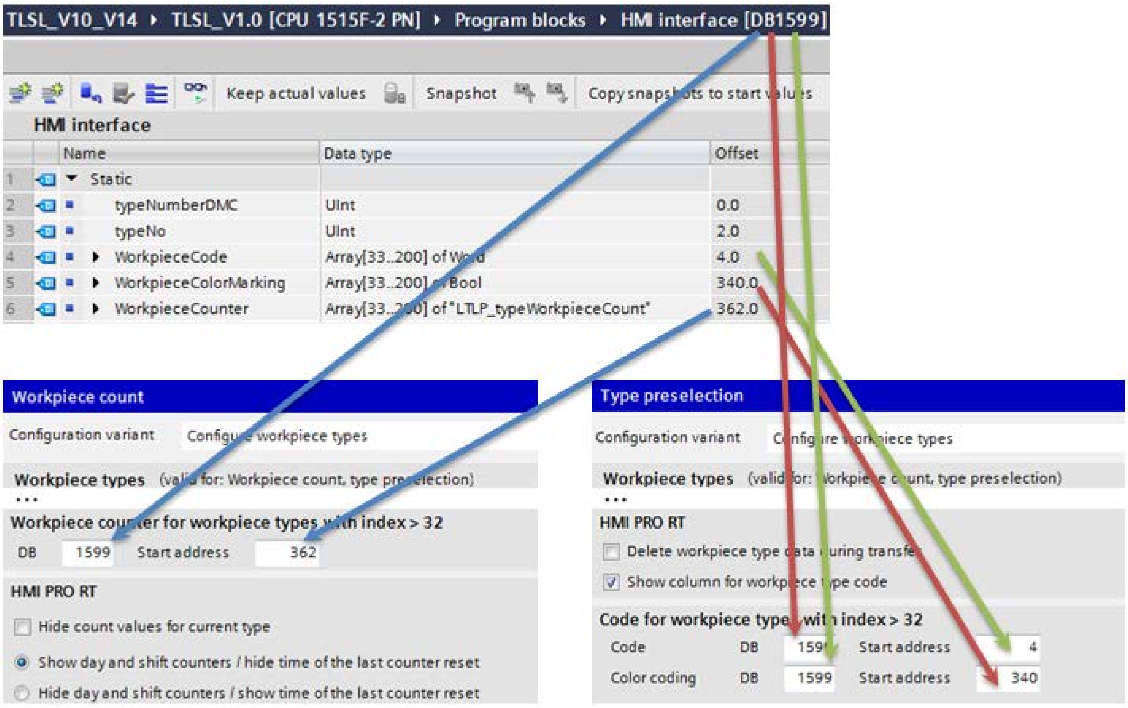
Current type on exit belt
Cylinder head 1.8l 4V

Fig. 8-7: Relationship between type preselection, workpiece overview and part counter

General information about extensions for a type number > 32

The code, the color change, and the counter for types > 32 must be located in a special data block. As HMI PRO can only access non-optimized data blocks, the properties of the data source must be set accordingly.

Only with these settings can you see the relevant absolute addresses that you have to enter in HMI PRO CS.



TLSL_V10_V14 ▸ TLSL_V1.0 [CPU 1515F-2 PN] ▸ Program blocks ▸ HMI interface [DB1599]

HMI interface

	Name	Data type	Offset
1	▾ Static		
2	typeNumberDMC	UInt	0.0
3	typeNo	UInt	2.0
4	▸ WorkpieceCode	Array[33..200] of Word	4.0
5	▸ WorkpieceColorMarking	Array[33..200] of Bool	340.0
6	▸ WorkpieceCounter	Array[33..200] of "LTLP_typeWorkpieceCount"	362.0

Workpiece count

Configuration variant: Configure workpiece types

Workpiece types (valid for: Workpiece count, type preselection)

...

Workpiece counter for workpiece types with index > 32

DB 1599 Start address 362

HMI PRO RT

□ Hide count values for current type

● Show day and shift counters / hide time of the last counter reset

○ Hide day and shift counters / show time of the last counter reset

Type preselection

Configuration variant: Configure workpiece types

Workpiece types (valid for: Workpiece count, type preselection)

...

HMI PRO RT

□ Delete workpiece type data during transfer

☑ Show column for workpiece type code

Code for workpiece types with index > 32

Code DB 1599 Start address 4

Color coding DB 1599 Start address 340

Define data blocks for workpiece types> 32 (blue arrow)

Optional display for the workpiece type code: Define array for code (red arrow)
Array for color change (green arrow)

Fig. 8-8: Configuring more than 32 workpiece types

Type preselection or workpiece overview (types > 32)

For the optionally displayable code **WorkpieceCode**, you use an array [33..maxNoTypes], word type; for the optionally displayable color change in the **Type preselection** screen, you use an array [33..maxNoTypes], bool type.

Workpiece counting (types> 32)

For workpiece counting, you can find the **LTLP_typeWorkpieceCount** data type to be used in your project under **PLC DataType** in the **Common** directory. You parameterize these in a separate DB as array [33..maxNoTypes] of **LTLP_typeWorkpieceCount**.

8.2.1 Type preselection

A standard block is not available for type preselection. The type of workpiece to be machined is identified during type preselection and displayed as the current type on the loading belt.
For automatic detection, the word "**LTLP_HMIProInterfaceData".workpiecetypeLoadingBelt** must be supplied. Note that the word is BCD-coded and the LOW and HIGH bytes are interchanged.
On a manual change, the operator can change the type on the type preselection screen. The type preselected by the operator is transferred using data word "**LTLP_HMIProInterfaceData".selWorkpiecetypePreselect**. The screen ID of the type preselection screen and the plus key are used to trigger acceptance of the type. (Fig. 8-7 →1)

```
IF (( "LTLP_HMIProInterfaceData".noScreenId = 58) AND "LTLP_HMIProInterfaceData".pPlus)
THEN
      "LTLP_HMIProInterfaceData".workpiecetypeLoadingBelt
      :="LTLP_HMIProInterfaceData".selWorkpiecetypePreselect
; END_IF;
```

The color change is read from "**LTLP_HMIProInterfaceData".typePreselectColorChange[1]**. If the workpiece code is to be activated, then for types 1-32, this is read in the **LTLP_HMIProInterfaceData** from **typePreselectWorkpieceCode[1]**.

8.2.2 Workpiece overview

Note

The TRANSLINE blocks and examples only support eight workpieces with the same name per station.

The Workpiece overview screen shows the workpiece types available for each station, plus the status of the workpieces (good/scrap). Each workpiece is visualized using the following displays:

- Workpiece code (can be optionally hidden)
- Workpiece type
- Workpiece correct/not correct
- Workpiece present/not present
- Extended workpiece status (see online help for HMI PRO sl RT)

Note

Regarding the display of workpiece types and codes, please note that LOW and HIGH bytes are swapped for workpiece types, as they are for type preselection. The workpiece type is BCD-coded. If the workpiece type in the stations is machined using the workpiece code, please note that the workpiece code is stored as an integer value.

Transferring the type from type preselection to station 1

Copies the data for the workpiece overview
"**LTLP_HMIProInterfaceData".workpieceTypeStation[1]**
Depending on the number of workpieces with the same name, bits
"**LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAPresent** and
"**LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAOk** must be set.

```
IF ("mipStartTypTransfer")
THEN
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType :=
    "LTLP_HMIProInterfaceData".workpiecetypeLoadingBelt;
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAPresent := true;
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAOk := true;
END_IF;
```

Transferring the type number from a data storage medium

When using the type number from the data storage medium, reading the type of the type identifier no longer applies. The type number from the data storage medium is written to data word
"**LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType**.

Transferring the type from type preselection with type number in the Code field to station 1

Copies the data for the workpiece overview
"**LTLP_HMIProInterfaceData".workpieceTypeStation[1]**
Depending on the number of workpieces of the same name, the bits must be set as code in "**LTLP_HMIProInterfaceData".station[1]**.

```
IF ("mipStartTypTransfer")
THEN
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType :=
    "LTLP_HMIProInterfaceData".workpiecetypeLoadingBelt;
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAPresent := true;
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAOk := true;
    "LTLP_HMIProInterfaceData".station[1] := BCD32_TO_DINT(IN :=
    SWAP("LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType));
END_IF;
```

Transferring the type number from a data storage medium with type number in the Code field

When using the codes from the data storage medium, reading the type of the type identifier no longer applies. The type number from the data storage medium is written to data word "**LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType**.

Transferring the type from type preselection with workpiece code to station 1

When using the code from a data storage medium, the transfer of the type preselection could look like the following.

```
IF ("mipStartTypTransfer")
THEN
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType :=
    "LTLP_HMIProInterfaceData".workpiecetypeLoadingBelt;
    "HMI_interface".typNo := BCD32_TO_DINT(IN :=
    SWAP("LTLP_HMIProInterfaceData".workpieceTypeStation[1].workpieceType));
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAPresent := true;
    "LTLP_HMIProInterfaceData".workpieceTypeStation[1].typeAOk := true;
    // Determine workpiece code
    IF ("HMI_interface".typNo <= 32)
    THEN
        "LTLP_HMIProInterfaceData".station[1] :=
        "LTLP_HMIProInterfaceData".typePreselectWorkpieceCode["HMI_interface ".typNo];
    ELSE
        "LTLP_HMIProInterfaceData".station[1] :=
        "HMI_interface".WorkpieceCode["HMI_interface".typNo]
     END_IF
END_IF;
```

From this point, the user can change workpiece OK / workpiece not OK, available/unavailable or the workpiece code.
Workpiece OK / workpiece not OK and the workpiece code can be changed by the user on the workpiece overview screen. The PLC programmer is informed about the workpiece status (e.g. OK/not OK and workpiece code) using data bytes **noFunctionLow** and **noFunctionHigh** in the **LTLP_HMIProInterfaceData** [DBW79]. Since this data word is used for transferring from several screens, the evaluation of the data word must always be locked using the screen ID or screen number (see HMI PRO CS online help).

Changing from workpiece OK/workpiece not OK and available/not available

Note

In the PLC, this part of the workpiece overview is only possible based on pointer addressing. The data connection to HMI PRO does not allow any other possibility.

The workpiece properties can be changed using the plus key to indicate "workpiece OK" or "available", or the minus key to indicate "workpiece not OK" or "unavailable".

Checking the screen ID

```
    L     "LTLP_HMIProInterfaceData".noScreenId   //(DBW84) screen selected from HMI PRO
    L     23                                      //Screen identifier workpiece overview
    <>I
    JC    n_1

//Check keys

    UN    "LTLP_HMIProInterfaceData“.Pplus        // (DBX77.1)
    UN    "LTLP_HMIProInterfaceData“.Pminus       // (DBX77.2)
    JC    cod1

//Check whether correct/not correct or present/unavailable has been changed

    L     "LTLP_HMIProInterfaceData“.noFunctionLow        //(DBB79) low value of the function
                                                          //value BCD coded
    L     B#16#40                                 //Values < 40 signify OK/not
                                                  //OK selected
    <=I
    JC    ion

//Present/not present has been changed.
//Workpiece number and station number are coded in data byte noFunctionHigh or noFunctionLow
//byte. The workpiece number can be calculated from noFunctionHigh (BCD coded) – 40.
//Likewise, the station number is calculated from noFunctionLow (BCD) –40. In
//the example below, 41 is deducted because the output address corresponds to
//workpiece or station number 1.

    L     "LTLP_HMIProInterfaceData“.noFunctionHigh       //(DBB79) high value of the function
                                                          //value BCD coded
    BTI                                  //Convert to integer
    L     41                             //Identifier for workpiece property change
    -I
    T     #workshop number                        //(tempVar)
    L     2                              //The workpiece properties occupy 2 bits ->
    *I                                   //times 2
    T     #Bit number                    //(tempVar)
    L     "LTLP_HMIProInterfaceData“.noFunctionLow        I/(DBB80) low value of the function
                                                          //value BCD coded
    BTI                                  //Convert to integer
    L     41                             //Identifier for the function from the workpiece overview
    -I
    T     #station number                         //(tempVar)
    JU    bit
ion:      NOP          0
```

// OK/not OK has been changed
//Workpiece number and station number are coded in data byte ***noFunctionHigh*** or ***noFunctionL*ow**
//byte. The workpiece number can be calculated from ***noFunctionHigh*** (BCD coded) – 40.
//The station number is available directly in ***noFunctionL*ow** as a BCD value. In the following
//41 or 1 is deducted because the output address corresponds to workpiece or station number 1.
//The 2nd bit is addressed in the OK / not OK bit - therefore, a value of 1 is added for the
//bit number.

```
    L      "LTLP_HMIProInterfaceData“.noFunctionHigh           //(DBB79) high value of the function
                                                              //value BCD coded
    BTI                                   //Convert to integer
    L      41                             //Identifier for workpiece property change
    -I
    T      #workshop number                        //(tempVar)
    L      2                              //The workpiece properties occupy 2 bits ->
    *I                                    //times 2
    L      1                              //Plus 2 due to 2nd bit as OK/not OK Info
    +I
    T      #Bit number                    //(tempVar)
    L      "LTLP_HMIProInterfaceData“.noFunctionLow            //(DBB80) low value of the function
                                                              //value BCD coded
    BTI                                   //Convert to integer
    L      1                              //As index 1 == start address
    -I
    T      #station number                         //(tempVar)

bit:       NOP          0
```

//Calculate bit number
//2 words are available for each station. Word 1 is for the workpiece type. Word 2 for the
//properties of the workpiece type The STEP 7 address calculation uses bits, which is why each
//station is increased by 8 * 4 bytes == SLW 5

```
    L       #stationNo                    //(tempVar)
    SLW     5                             //*24
    LAR1    P#1558.0                      //Start address workpiece properties
    +AR1                          //Address station workpiece properties
    L       #Bit number
    +AR1                          //Bit address of the workpiece property
```

//Set/reset bit for correct/not correct or present/not present

```
    U      "LTLP_HMIProInterfaceData".Pplus
    S      DBX [AR1,P#0.0]
    U      "LTLP_HMIProInterfaceData".Pminus
    R      DBX [AR1,P#0.0]

cod1:      NOP          0
```

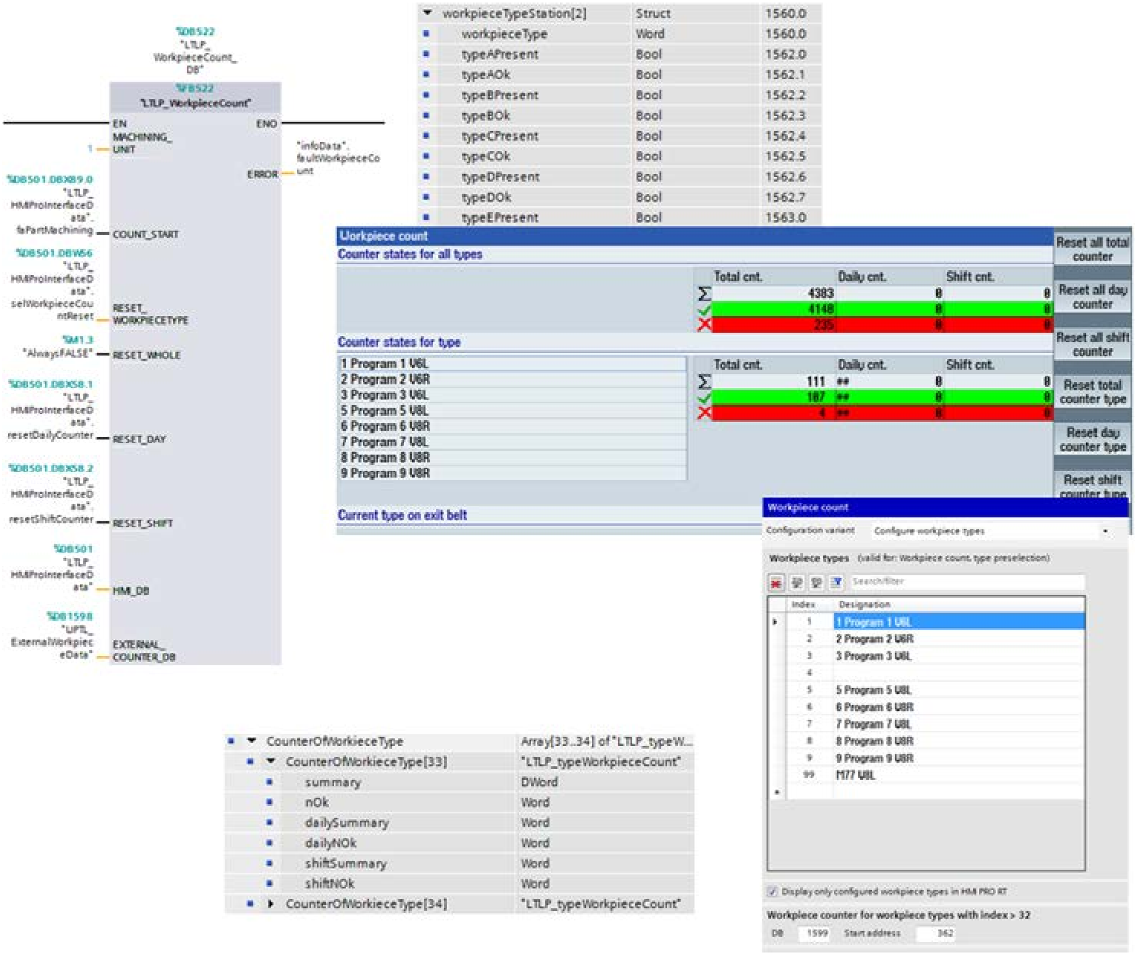
Changing the workpiece code

The workpiece code is selected by the user via the Insert key. After entering the value, using the return key, the value is saved to **"LTLP_HMIProInterfaceData".codeWorkpiece** in the integer format. The station that has been addressed is stored in the BCD format in "**LTLP_HMIProInterfaceData".noFunctionLow** The station number of the modified code is in **"LTLP_HMIProInterfaceData".noFunctionLow** as BCD value. The address of the modified station code is determined from the station number and the start value of the station code.

```
IF ("LTLP_HMIProInterfaceData".noScreenId = 23)
THEN
    IF ("LTLP_HMIProInterfaceData".noFunctionHigh = W#16#61)
    THEN
          // Workpiece code changed
          "LTLP_HMIProInterfaceData".station [(BCD16_TO_INT
         ("LTLP_HMIProInterfaceData".noFunctionLow)-40)] :=
         "LTLP_HMIProInterfaceData".codeWorkpiece ;
    END_IF
END_IF;
```

8.2.3 Workpiece count

Counters are updated using the **LTLP_WorkpieceCount** [FB522] block. The last station's data is documented with the aid of the workpiece count. Transfer to the block is realized using parameter MACHINING_UNIT. The display in the HMI PRO screen is realized by copying type **"LTLP_HMIProInterfaceData".workpieceTypeStation[*MACHINING_UNIT*].workpieceType** into the data word for "workpiece type at the exit belt" **"LTLP_HMIProInterfaceData".workpieceTypeExitBelt**. This variable is also used for the MDA/PDA connection to TRANSLINE Collect.


%DB522
"LTLP_WorkpieceCount_DB"
%FB522
"LTLP_WorkpieceCount"
EN
ENO
1 — MACHINING_UNIT
ERROR — "infoData".faultWorkpieceCount
%DB501.DBX89.0 "LTLP_HMIProInterfaceData".faPartMachining — COUNT_START
%DB501.DBW56 "LTLP_HMIProInterfaceData".selWorkpieceCountReset — RESET_WORKPIECETYPE
%M1.3 "AlwaysFALSE" — RESET_WHOLE
%DB501.DBX58.1 "LTLP_HMIProInterfaceData".resetDailyCounter — RESET_DAY
%DB501.DBX58.2 "LTLP_HMIProInterfaceData".resetShiftCounter — RESET_SHIFT
%DB501 "LTLP_HMIProInterfaceData" — HMI_DB
%DB1598 "UPTL_ExternalWorkpieceData" — EXTERNAL_COUNTER_DB
Reset all total counter
Reset all day counter
Reset all shift counter
Reset total counter type
Reset day counter type
Reset shift counter type

▼ workpieceTypeStation[2]	Struct	1560.0
workpieceType	Word	1560.0
typeAPresent	Bool	1562.0
typeAOk	Bool	1562.1
typeBPresent	Bool	1562.2
typeBOk	Bool	1562.3
typeCPresent	Bool	1562.4
typeCOk	Bool	1562.5
typeDPresent	Bool	1562.6
typeDOk	Bool	1562.7
typeEPresent	Bool	1563.0

Uorkpiece count

Counter states for all types

	Total cnt.	Daily cnt.	Shift cnt.
Σ	4383	0	0
✓	4148	0	0
✗	235	0	0

Counter states for type

1 Program 1 U6L
2 Program 2 U6R
3 Program 3 U6L
5 Program 5 U8L
6 Program 6 U8R
7 Program 7 U8L
8 Program 8 U8R
9 Program 9 U8R

	Total cnt.	Daily cnt.	Shift cnt.
Σ	111	** 0	0
✓	107	** 0	0
✗	4	** 0	0

Current type on exit belt

Workpiece count

Configuration variant: Configure workpiece types

Workpiece types (valid for: Workpiece count, type preselection)

Search/filter

Index	Designation
1	1 Program 1 U8L
2	2 Program 2 U8R
3	3 Program 3 U8L
4	
5	5 Program 5 U8L
6	6 Program 6 U8R
7	7 Program 7 U8L
8	8 Program 8 U8R
9	9 Program 9 U8R
99	M77 U8L

☑ Display only configured workpiece types in HMI PRO RT

Workpiece counter for workpiece types with index > 32

DB: 1599 Start address: 362

▼ CounterOfWorkieceType	Array[33..34] of "LTLP_typeW...
▼ CounterOfWorkieceType[33]	"LTLP_typeWorkpieceCount"
summary	DWord
nOk	Word
dailySummary	Word
dailyNOk	Word
shiftSummary	Word
shiftNOk	Word
▶ CounterOfWorkieceType[34]	"LTLP_typeWorkpieceCount"

Fig. 8-9: Interrelationship between the screen Workpiece count and block LTLP_WorkpieceCount

The parameters of the **LTLP_WorkpieceCount** are explained in the diagram above. (Additional information, see FB description). The workpiece types to be counted are counted depending on the index, which supplies formal parameter **MACHINING_UNIT**. Here, the workpiece type is expected with LOW/HIGH interchanged and BCD-coded. The subsequent 16 bits (see Fig. 8-9 data excerpt above) contain the status of the workpieces for each count cycle: available (typeXPresent = TRUE and OK/not OK (typeXOK = TRUE/FALSE). The **COUNT_START** parameter is used for counting, taking the data described above into account.

LTLP_WorkpieceCount counts the workpiece types 1 to 32 within the **LTLP_HMIProInterfaceData**.
Types > 32 are counted in a data block **UPTL_ExternalWorkpieceData** to be created by the OEM, which is based on the **LTLP_typeWorkpieceCount**. This data block is in the TRANSLINE library under types and under copy templates in the **ProductionDataAcquisition** directory.
Count data can be reset using parameters **RESET_WORKPIECETYPE**, **RESET_WHOLE**, **RESET_DAY** and **RESET_SHIFT**. If, at these parameters, the data from HMI PRO are created, then the operator can reset the total counter, daily counter and shift counter.

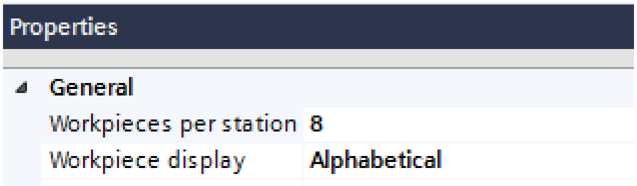
Note

The workpiece counter type (**preselWorkpieceCountReset**), which should be reset, is BCD-coded and LOW/HIGH are interchanged.

Using **"LTLP_HMIProInterfaceData".resetTotalCounter,** the total counter is reset, using **"LTLP_HMIProInterfaceData".resetDailyCounter** the daily counter and using **"LTLP_HMIProInterfaceData".resetShiftCounter** the shift counter.
If, in the function key assignment of HMI PRO the **Shift model** and **Utilization** screens are integrated, then there is an automated version to reset these counters.
The actual shift is stored in data byte **"LTLP_HMIProInterfaceData".shiftNumber**.
This can be used to determine how shift and daily counters are reset.

LTLP_WorkpieceCount [FB522]

This block is required for the **Workpiece count** HMI PRO screen.
The workpieces machined in the machine are counted using this block. Up to 1000 workpiece types with a maximum of 8 workpieces are possible per station.
(Configuring in the **workpiece overview** screen under **Properties**)


Properties

General

Workpieces per station 8

Workpiece display Alphabetical

Fig. 8-10: HMI PRO CS – workpiece overview, workpieces per station

Workpieces are counted each shift, each day and over the entire period. Every counter indicates the sum of all workpieces as well as correct and incorrect workpieces. The counters can be reset.

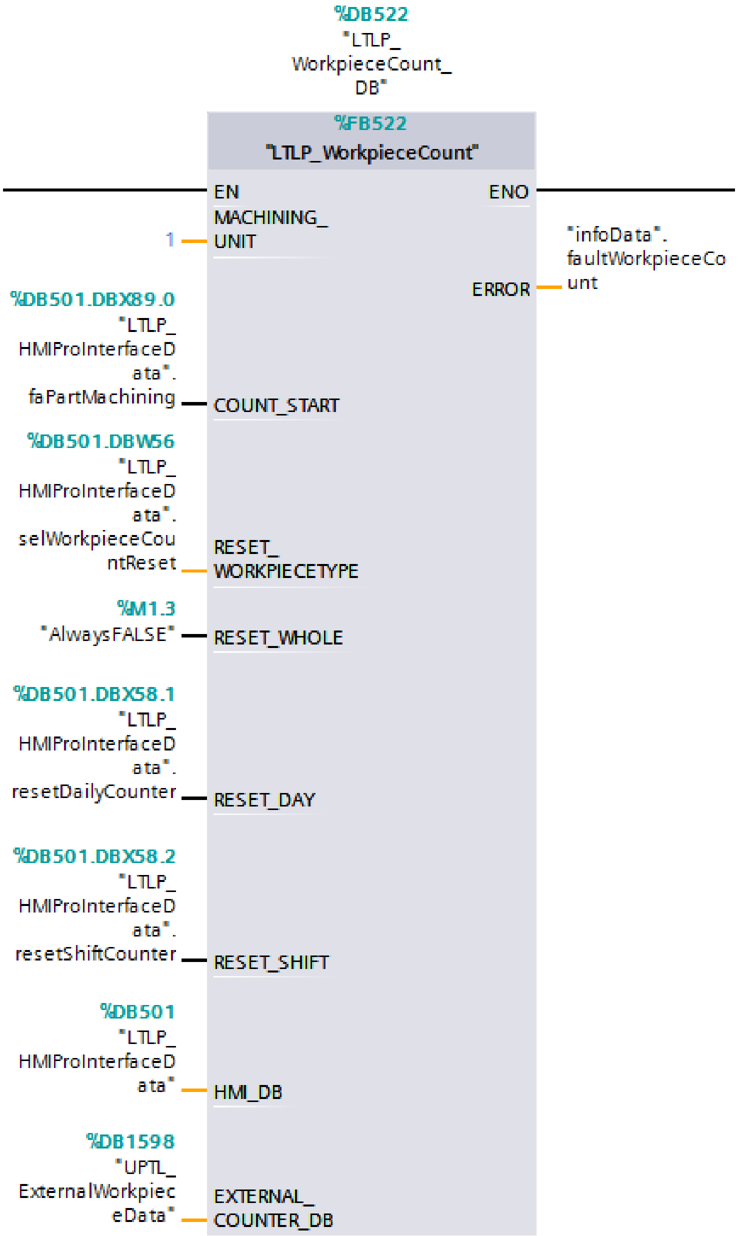
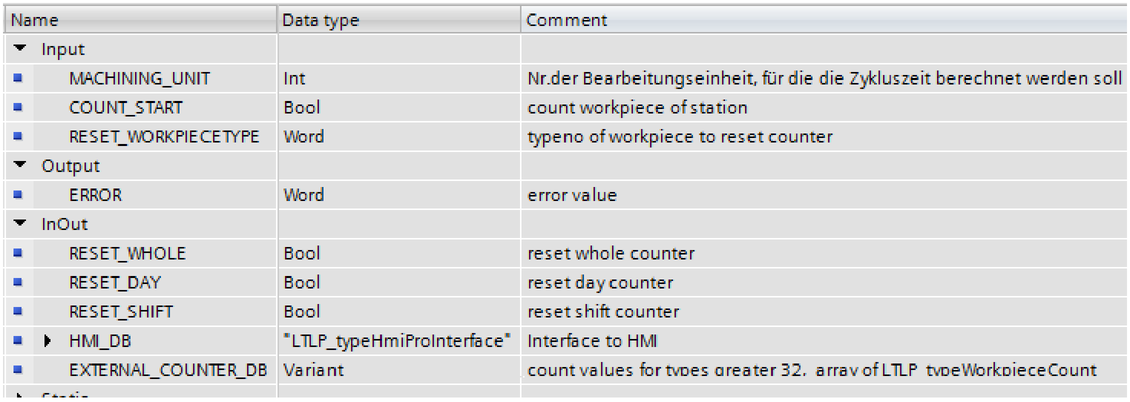


Fig. 8-11: LTLP_WorkpieceCount [FB522]



Name	Data type	Comment
▼ Input		
■ MACHINING_UNIT	Int	Nr.der Bearbeitungseinheit, für die die Zykluszeit berechnet werden soll
■ COUNT_START	Bool	count workpiece of station
■ RESET_WORKPIECETYPE	Word	typeno of workpiece to reset counter
▼ Output		
■ ERROR	Word	error value
▼ InOut		
■ RESET_WHOLE	Bool	reset whole counter
■ RESET_DAY	Bool	reset day counter
■ RESET_SHIFT	Bool	reset shift counter
■ ▸ HMI_DB	"LTLP_typeHmiProInterface"	Interface to HMI
■ EXTERNAL_COUNTER_DB	Variant	count values for types greater 32, array of LTLP_typeWorkpieceCount
▸ Static		

Fig. 8-12: LTLP_WorkpieceCount parameter

Application

This block is required for the **Workpiece count** HMI PRO screen.

Data interface

For a type number <= 32, the data area in **LTLP_HMIProInterfaceData** (from **cntTotalWorkpieces** up to **cntWorkpiecetype[32]**) is supplied. As data source, a structure is taken from **"LTLP_HMIProInterfaceData".workpieceTypeStation[]**, which refers to parameter **STATION_INDEX**. For a type number > 32, parameter **EXTERNAL_COUNTER_DB** must be supplied with an array, type **LTLP_typeWorkpieceCount**.

Description of input parameters

Table 8-4: Input parameter LTLP_WorkpieceCount

Parameter	Data type	Description
MACHINING_UNIT	Int	Number of the machining unit (1 to 40), whose workpiece type should be counted This is a reference to the following structure within the HMI PRO interface **LTLP_HMIProInterfaceData**:

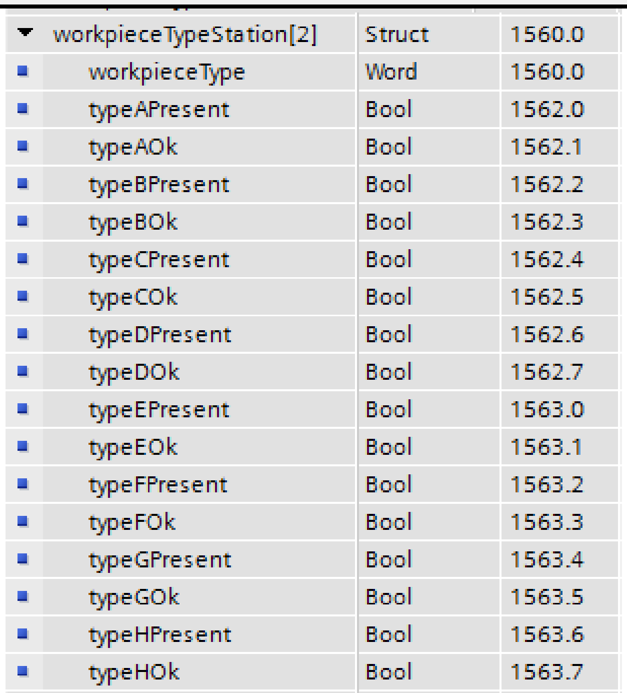
<table>
<tr><th>Parameter</th><th>Data type</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>▼ workpieceTypeStation[2]</td><td>Struct</td><td>1560.0</td></tr>
<tr><td>■ workpieceType</td><td>Word</td><td>1560.0</td></tr>
<tr><td>■ typeAPresent</td><td>Bool</td><td>1562.0</td></tr>
<tr><td>■ typeAOk</td><td>Bool</td><td>1562.1</td></tr>
<tr><td>■ typeBPresent</td><td>Bool</td><td>1562.2</td></tr>
<tr><td>■ typeBOk</td><td>Bool</td><td>1562.3</td></tr>
<tr><td>■ typeCPresent</td><td>Bool</td><td>1562.4</td></tr>
<tr><td>■ typeCOk</td><td>Bool</td><td>1562.5</td></tr>
<tr><td>■ typeDPresent</td><td>Bool</td><td>1562.6</td></tr>
<tr><td>■ typeDOk</td><td>Bool</td><td>1562.7</td></tr>
<tr><td>■ typeEPresent</td><td>Bool</td><td>1563.0</td></tr>
<tr><td>■ typeEOk</td><td>Bool</td><td>1563.1</td></tr>
<tr><td>■ typeFPresent</td><td>Bool</td><td>1563.2</td></tr>
<tr><td>■ typeFOk</td><td>Bool</td><td>1563.3</td></tr>
<tr><td>■ typeGPresent</td><td>Bool</td><td>1563.4</td></tr>
<tr><td>■ typeGOk</td><td>Bool</td><td>1563.5</td></tr>
<tr><td>■ typeHPresent</td><td>Bool</td><td>1563.6</td></tr>
<tr><td>■ typeHOk</td><td>Bool</td><td>1563.7</td></tr>
</table>
Fig. 8-13: Example of station index 2</td></tr>
<tr><td>COUNT_START</td><td>Bool</td><td>Production counter
The appropriate workpiece of the station is counted with the positive edge.
Data bit <b>"LTLP_HMIProInterfaceData".completionReport</b> or <b>completeMachineCompletReport</b> is used for this purpose.</td></tr>
<tr><td>RESET_ WORKPIECETYPE</td><td>Bool</td><td>Number of the workpiece type whose counter values should be cleared
= 0: all counter values should be reset
Parameter is not supplied from HMI PRO:
Value must be specified as BCD number (caution: not high/low interchanged!)
Parameter is supplied from HMI PRO:
the value is supplied using <b>"LTLP_HMIProInterfaceData".selWorkpieceCountReset</b> [DB501.DBW56]
(= workpiece type that was selected by the system supervisor when resetting total, daily and shift counters)</td></tr>
</table>

Table 8-5: Output parameter LTLP_WorkpieceCount

Parameter	Data type	Description	
ERROR	Word	0x0000	No error
		0x8001	Station index incorrect
		0x8002	Workpiece type incorrect
		0x8003	DB for the external count values is incorrect
		0x8004	Array size for external counter values is not adequate

Table 8-6: InOut parameter LTLP_WorkpieceCount

Parameter	Data type	Description
RESET_WHOLE	Int	Reset total counter of the selected workpiece type If this is to be realized via HMI PRO, then **"LTLP_HMIProInterfaceData".resetTotalCounter** should be entered. If the counter is reset, the block sets the parameter to 0.
RESET_DAY	Bool	Reset daily counter of the selected workpiece type If this is to be realized via HMI PRO, then **"LTLP_HMIProInterfaceData".resetDailyCounter** should be entered. If the counter is reset, the block sets the parameter to 0.
RESET_SHIFT	Bool	Reset shift counter of the selected workpiece type If this is to be realized via HMI PRO, then **"LTLP_HMIProInterfaceData".resetShiftCounter** should be entered. If the counter is reset, the block sets the parameter to 0.
HMI_DB	LTLP_typeHmi ProInterface	Interface between HMI PRO and PLC user program. As default setting, this is the **LTLP_HMIProInterface** [DB501] from the copy template.
EXTERNAL_ COUNTER_DB	Variant	For workpiece type count > 32, an ARRAY[33...xx] of LTLP_typeWorkpieceCount must be created for the workpiece types type number 33 and following. This is **UPTL_ExternalWorkpieceData** as default setting. Array [33.. max. number of types] contained in the **UPTL_ExternalWorkpieceData** must have a structure, type **LTLP_typeWorkpieceCount**. For a workpiece type count <= 32, a null flag can be placed here

Description of the return value

No value is returned.

8.3 Tool lifetime with HMI PRO sl

The tool lifetime is evaluated in HMI PRO sl using the **Tool life overview** and **Tool wear** screens.
The following options are available:

- Tool magazine with standard tool management
- Tool group with PLC-based tool management (e.g. in TRANSLINE HMI PRO with **LTLP_HMIProInterfaceData** or machine manufacturer-specific tool management).

8.3.1 HMI PRO sl and Siemens tool management

In HMI PRO CS, insert the **Tool wear** screen in the function key assignment of the HMI PRO text project.
This screen corresponds to the **Tool wear** screen of the SINUMERIK tool management and must be configured in the Parameters operating area in the controller.

8.3.2 HMI PRO sl and PLC-based lifetime determination

In HMI PRO CS, insert the **Tool life overview** screen in the function key assignment of the HMI PRO text project:
Configure the tool groups by entering a name for each group. You can also add further information for each tool group in the Tool info field (see also online help for HMI PRO CS).

The counter values for a maximum of 96 tool groups are managed and stored in the controller. The counter values for the actual value, pre-alarm, alarm and extended alarm are listed in the **LTLP_HMIProInterfaceData** using the **LTLP_ToolLifeTime** (FB521).
Further, the alarm status and the activation of the lifetime extension of a tool group are also displayed.

8.3.3 LTLP_ToolLifeTime [FB521]

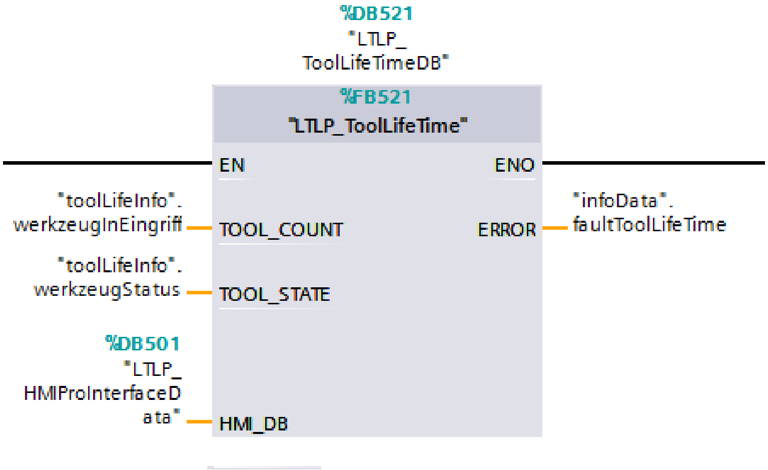
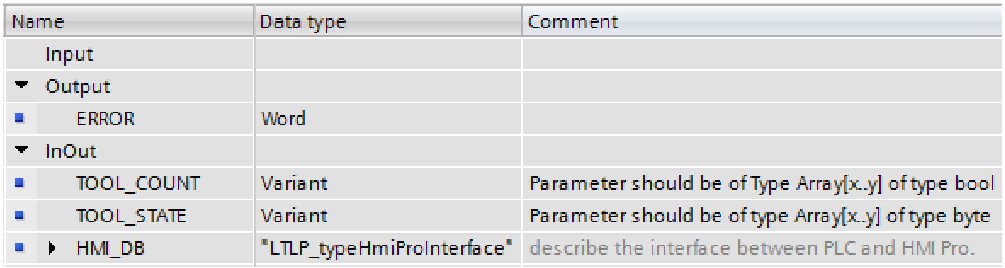


Fig. 8-14: LTLP_ToolLifeTime [FB521]



Name	Data type	Comment
Input		
Output		
ERROR	Word	
InOut		
TOOL_COUNT	Variant	Parameter should be of Type Array[x..y] of type bool
TOOL_STATE	Variant	Parameter should be of type Array[x..y] of type byte
HMI_DB	"LTLP_typeHmiProInterface"	describe the interface between PLC and HMI Pro.

Fig. 8-15: LTLP_ToolLifeTime parameter

Application

This block supplies the lifetime screens in the TRANSLINE environment.

Data interface

The block supplies areas **"LTLP_HMIProInterfaceData".toolGroup** and **"LTLP_HMIProInterfaceData".identification96Extended** of the HMI PRO data area.

Block description

Up to 99,999,999 machining cycles can be counted per tool group. The block processes as many as 96 tool groups. Each tool group receives its own counting pulse and reports the current status of the tool group to the user program.

Description of output parameters

Table 8-7: Output parameter LTLP_ToolLifeTime

Parameter	Data type	Description	
ERROR	Word	0x0000	No error
		0x80yy	Error active, yy can occur OR'ed
		0x8001	Data type for TOOL_COUNT parameter not permissible, not a BOOL array
		0x8002	Data type for TOOL_STATE parameter not permissible, not a BYTE array
		0x8004	Different number groups of TOOL_COUNT and TOOL_STATE
		0x8008	Array size exceeds the maximum number of 96 groups
		0x8010	internal error occurred
		0x8020	The group marked in the HMI does not exist

Description of InOut parameters

Table 8-8: InOut parameter LTLP_ToolLifeTime

Parameter	Data type	Description
TOOL_COUNT	Variant	A bool-type array is expected. The array limits correspond to the number of available tool groups. A positive edge of the corresponding bit specifies that the counter value is incremented for the associated tool group.
TOOL_STATE	Variant	A byte-type array is expected. The array limits correspond to the number of available tool groups. The status of the corresponding tool group is entered here. Prealarm reached 1 Alarm reached 2 Alarm extended 5 Extended alarm reached 6 Tool locked 8
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.

Description of the return value

No value is returned.

Responses to TOOL_STATE

The machine responds to workpiece lifetime alarms in accordance with this table.

Message	Immediate stop	Spindle stop	Travel to initial setting	Stop after end of cycle	Start disable	Warning
Prealarm	-	-	-	-	-	X
Alarm	-	-	-	X	X	X
Extended alarm	-	-	-	-	-	X

■

For notes

9

9 Diagnostics in HMI PRO

9.1 Device diagnostics

9.1.1 General information

The blocks, data types, and example DBs are stored in the TRANSLINE library under **Types/copy template** in the **HMIDeviceDiagnostic** directory.

9.1.2 LTLP_SLDeviceDiagnosticCommand [FB599]

The block is required to determine the diagnostic data of individual ProfiNet IO devices. It is called cyclically, and determines the data requested by the HMI via the system function **RD_REC**.

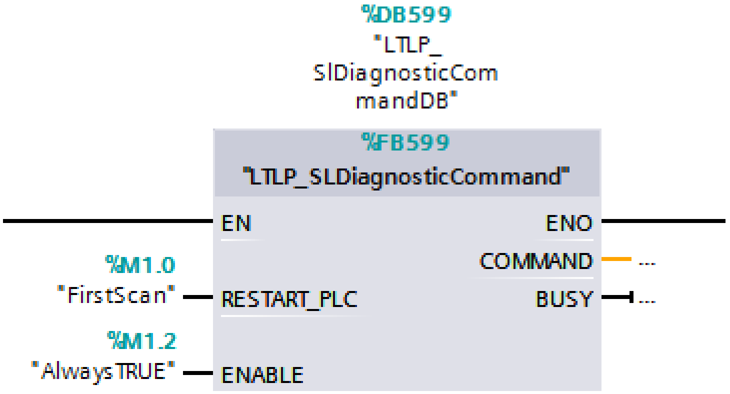
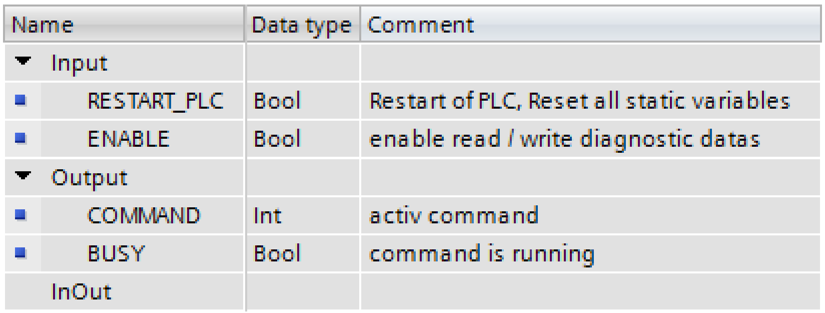


Fig. 9-1: LTLP_SLDiagnosticCommand [FC599]



Name	Data type	Comment
▼ Input		
■ RESTART_PLC	Bool	Restart of PLC, Reset all static variables
■ ENABLE	Bool	enable read / write diagnostic datas
▼ Output		
■ COMMAND	Int	activ command
■ BUSY	Bool	command is running
InOut		

Fig. 9-2: LTLP_SLDiagnosticCommand [FC599]

Description of input parameters

Table 9-1: Input parameter LTLP_SLDeviceDiagnosticCommand

Parameter	Data type	Description
RESTART_PLC	Bool	First block cycle following a CPU new start/restart. In the standard case, the "FirstScan" [M1.0] flag is specified here.
ENABLE	Bool	Enable to execute the task by HMI PRO

Description of InOut parameters

Table 9-2: InOut parameter LTLP_SLDeviceDiagnosticCommand

Parameter	Data type	Description
COMMAND	Int	Number of the active task (3 = read, 6 = write, 5 = reset)
BUSY	Bool	Processing is active.

■

10

10 Special features for HMI PRO with 1:N and virtual stations

10.1 General information

It is assumed that only one operator panel MPP, HT8 is active.

1:1 configuring

For a 1:1 configuration, HMI PRO has a connection with an individual controller.

Standard projects with a 1:1 configuration are identified in the status bar with .

1:N configuration

For a 1:N configuration, it is possible to visualize and operate up to 4 lower-level controllers with one operator panel (i.e. **one IPC** with **one MPP**; (several TCUs or HT8 are permitted)). Virtual stations are not possible!
An 1:N project includes a basic configuration (ROOT) and a control-specific configuration for each controller to be used.

1:N projects are designated with in the status list.

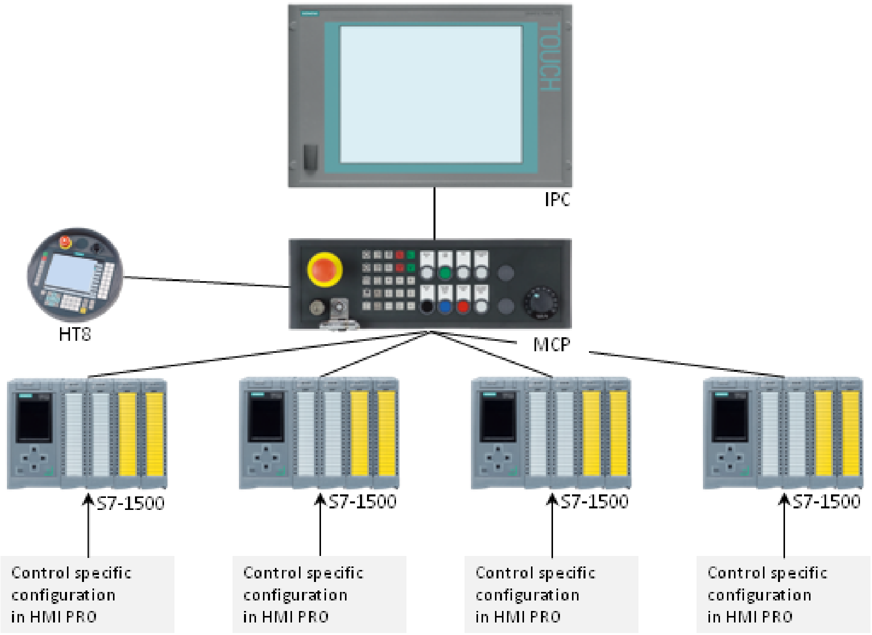

IPC
HT8
MCP
S7-1500
S7-1500
S7-1500
S7-1500
Control specific configuration in HMI PRO
Control specific configuration in HMI PRO
Control specific configuration in HMI PRO
Control specific configuration in HMI PRO

Fig. 10-1: Example of a 1:N configuration with 4 SIMATIC PLCs

Virtual stations

With the aid of the configuration of virtual stations, it is possible to visualize up to 20 stations on the operator panel with one controller. This means that a station is the virtual image of a PLC on the actual SIMATIC PLC controller. A separate HMI PRO DB with a different number is generated for each station to be visualized.
Permissible operator components are: **one IPC** with **one MPP** or **HT8.**
An HMI PRO project with virtual stations contains a basic configuration (ROOT) and, for each station to be visualized, a station-specific configuration in a dedicated data block. As a consequence, virtual stations operate with a multitude of standard HMI PRO DBs on a controller.
Virtual stations are only permitted for a 1:1 connection.

1:N projects are designated with DB in the status list.

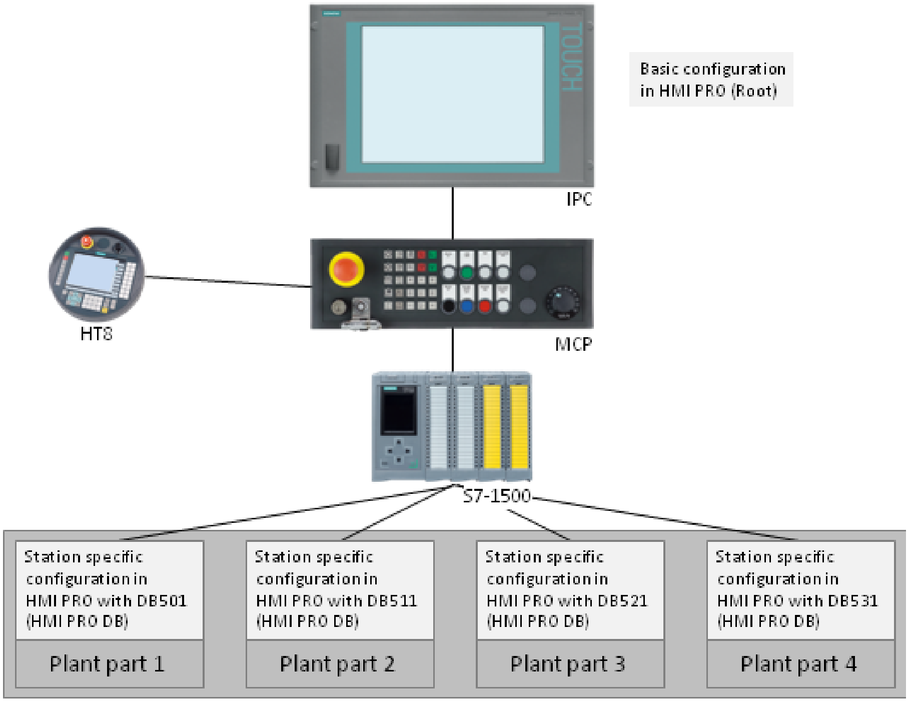

Basic configuration in HMI PRO (Root)
IPC
HT8
MCP
S7-1500
Station specific configuration in HMI PRO with DB501 (HMI PRO DB)
Plant part 1
Station specific configuration in HMI PRO with DB511 (HMI PRO DB)
Plant part 2
Station specific configuration in HMI PRO with DB521 (HMI PRO DB)
Plant part 3
Station specific configuration in HMI PRO with DB531 (HMI PRO DB)
Plant part 4

Fig. 10-2: Example of a configuration with 4 virtual stations

10.2 Configuration in HMI PRO CS

Controller

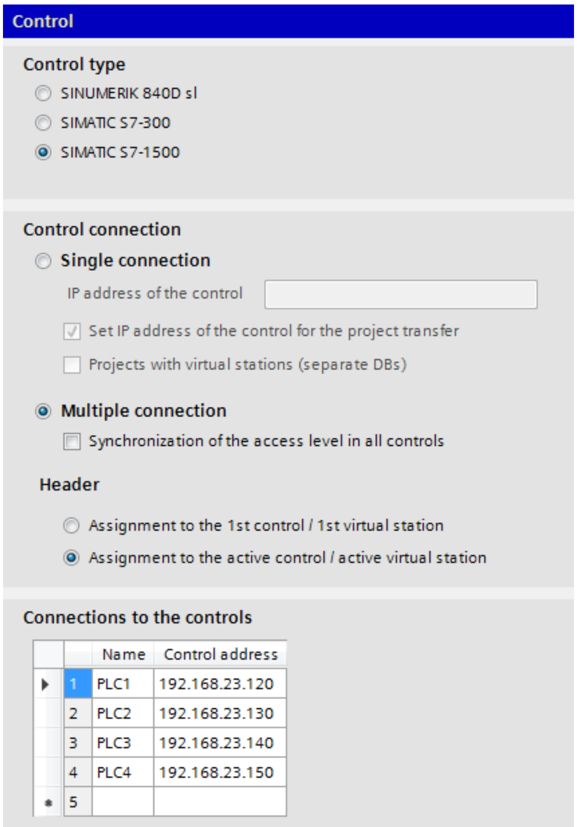
When configuring 1:N projects and projects with virtual stations, navigate in the project under ***Project name*** ▸ **Controller**. There, configure the connections from the HMI PRO to the controllers and/or configure the virtual stations with the HMI-PRO-DBs.
For the configuration, see the online help for HMI PRO CS.

10.2.1 1:N configuration with SIMATIC PLCs

You can parameterize up to 4 connections from HMI PRO to controllers (1:N) for a 1:N configuration.

Configure under **Controller type: SIMATIC S7-1500** and under **Controller connection: Multiple connection.**
Configure the connections from HMI PRO to the controllers by assigning controller names and addresses in the project navigation under ***Project name*** ▸ **Controller** ▸ **Connections to the controllers**.
Enter the following in each case:

- Name: Name of the PLC controller that you use for your 1:N project
- Controller address: IP address of the PLC controller that you use for your 1:N project



Control

Control type

○ SINUMERIK 840D sl
○ SIMATIC S7-300
● SIMATIC S7-1500

Control connection

○ Single connection

IP address of the control

☑ Set IP address of the control for the project transfer
□ Projects with virtual stations (separate DBs)

● Multiple connection

□ Synchronization of the access level in all controls

Header

○ Assignment to the 1st control / 1st virtual station
● Assignment to the active control / active virtual station

Connections to the controls

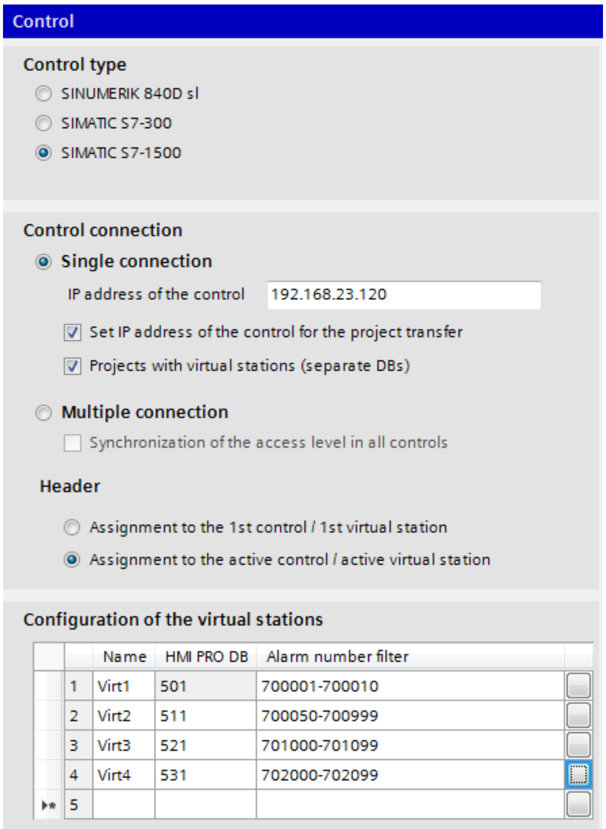
		Name	Control address
▸	1	PLC1	192.168.23.120
	2	PLC2	192.168.23.130
	3	PLC3	192.168.23.140
	4	PLC4	192.168.23.150
*	5		

Fig. 10-3: Configuring 1:N controller connections for SIMATIC-PLCs S7-1500

10.2.2 Configuration of virtual stations connected to a SIMATIC-PLC

You can parameterize up to 20 connections from a controller to virtual stations.

Configure under **Controller type: SIMATIC S7-1500** as well as under **Controller connection**: **Single connection** Enter the **IP address of the controller**, and activate checkbox **Project with virtual stations (separate DBs)**.
Under ***Project name*** ▸ **Controller** ▸ **Configuration of the virtual stations** configure the connections from HMI PRO to the virtual stations by assigning station names and HMI PRO DBs as well as alarm numbers.



Control

Control type

○ SINUMERIK 840D sl
○ SIMATIC S7-300
● SIMATIC S7-1500

Control connection

● Single connection

IP address of the control: 192.168.23.120

☑ Set IP address of the control for the project transfer
☑ Projects with virtual stations (separate DBs)

○ Multiple connection

□ Synchronization of the access level in all controls

Header

○ Assignment to the 1st control / 1st virtual station
● Assignment to the active control / active virtual station

Configuration of the virtual stations

		Name	HMI PRO DB	Alarm number filter	
	1	Virt1	501	700001-700010	□
	2	Virt2	511	700050-700999	□
	3	Virt3	521	701000-701099	□
	4	Virt4	531	702000-702099	□
▸*	5				□

Fig. 10-4: Configuring virtual stations for SIMATIC PLCs S7-1500

Enter the following:

- Name: Station name
- HMI DB: Enter the standard HMI DB (DB501) in the first line (Virt1); in the following lines (Virt2 to Virt20), enter the respective data block that you use for the station.
- Alarm number filter: Alarm numbers or number ranges that belong to the configured station

10.3 HMI PRO screen assignments

Assignment to all controllers

The following screens are independent of the controller selected in HMI PRO RT, and display the data of all connected controls:

- Alarms/messages
- Alarm analysis
- Alarm history
- Password specification by EKS

Fixed assignment to the default controller

Independent of the controller selected in HMI PRO RT, the following screens and functions are permanently assigned to the first controller (first controller, which is configured under ***Project name*** ▸ **Controller** ▸ **Connections to the controllers**):

- Lamp test
- Group acknowledgment
- Selecting a screen via PLC
- Change language via PLC
- Machine utilization (total counter, parts per cycle...)
- Shift model
- Parts tracking

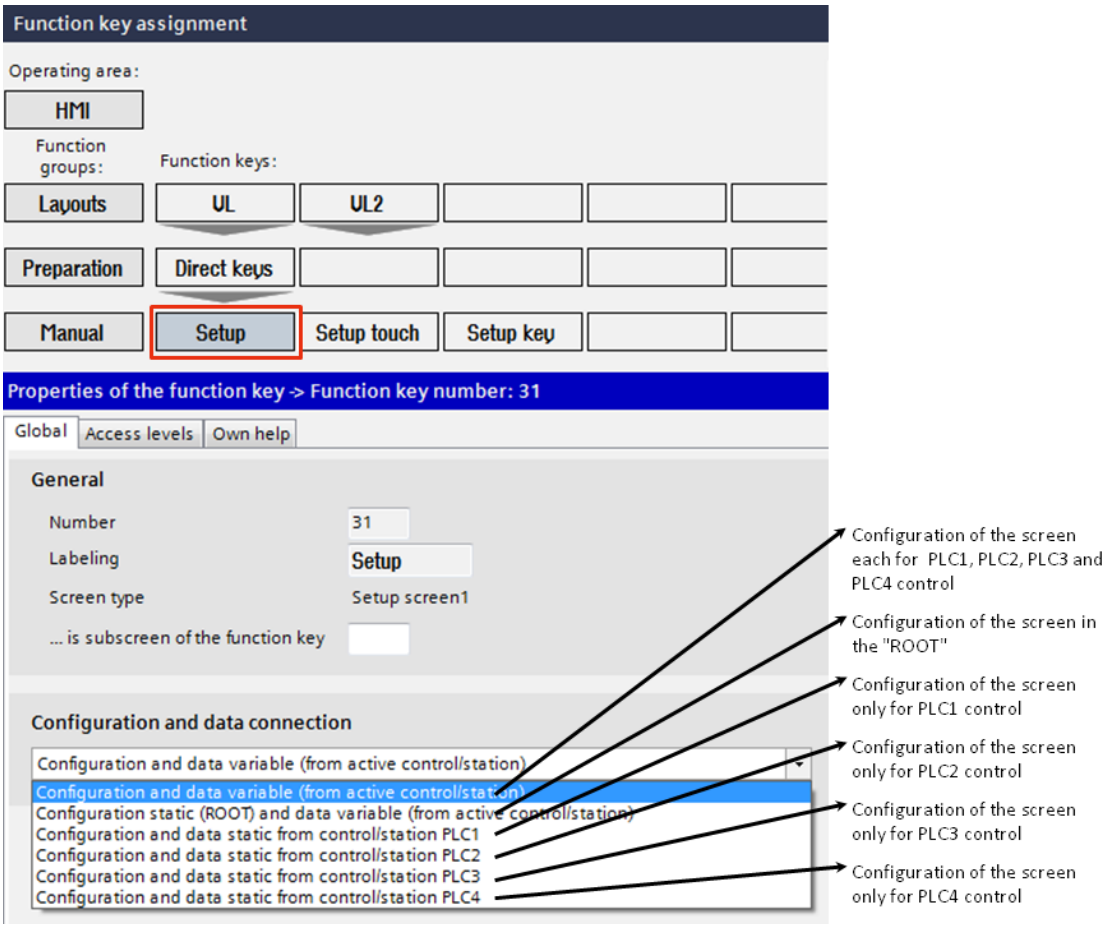
Configuring the assignment in the function key assignment

All other screens can be assigned via the function key menu of a controller or virtual station. The assignment is made within HMI PRO CS in the function key assignment.

Configuration and data variable (from active controller/station)
The screen is configured separately for each controller/virtual station, i.e. each controller/virtual station has its own texts and its own data areas. The screen must be configured for each controller/virtual station (PLC1, PLC2, PLC3, PLC4 in the example below).

Configuration static (ROOT) and data variable (from active controller/station)
The screen texts are only configured once in the root. Each controller/virtual station has its own data areas.

Configuration and data static from controllerX/stationX
The screen is configured for the specified controller (PLC1 in the example)



Properties of the function key -> Function key number: 31

General

Number: 31

Labeling: Setup

Screen type: Setup screen1

... is subscreen of the function key:

Configuration and data connection

Configuration and data variable (from active control/station)

- Configuration and data variable (from active control/station)
- Configuration static (ROOT) and data variable (from active control/station)
- Configuration and data static from control/station PLC1
- Configuration and data static from control/station PLC2
- Configuration and data static from control/station PLC3
- Configuration and data static from control/station PLC4

Configuration of the screen each for PLC1, PLC2, PLC3 and PLC4 control

Configuration of the screen in the "ROOT"

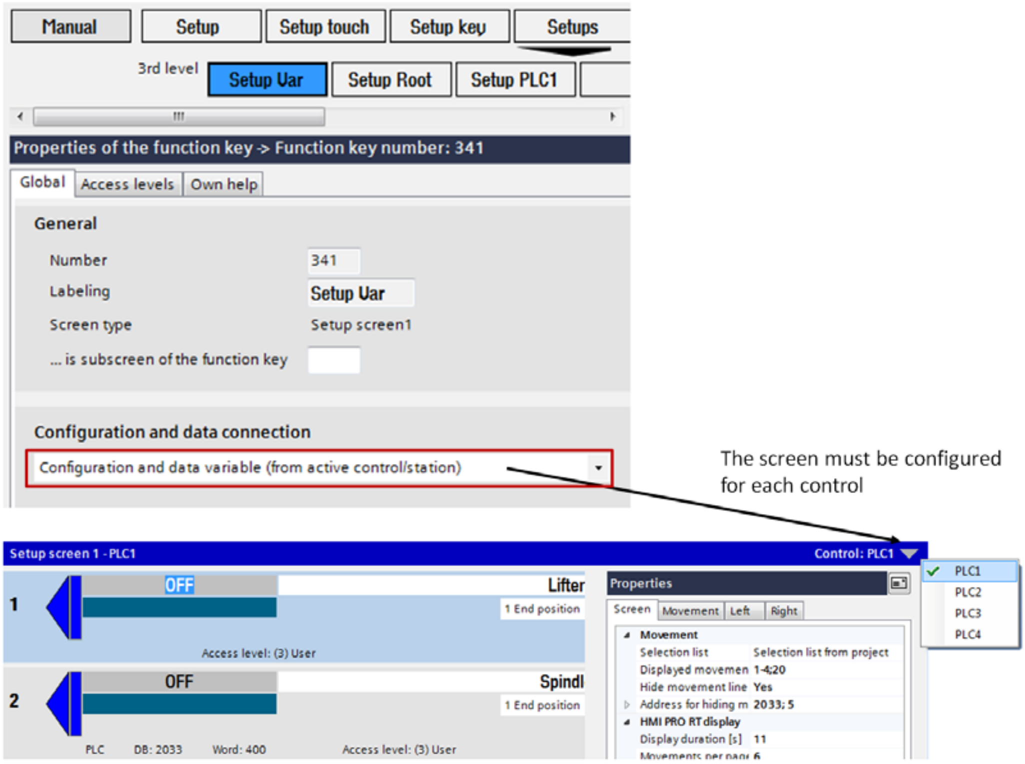
Configuration of the screen only for PLC1 control

Configuration of the screen only for PLC2 control

Configuration of the screen only for PLC3 control

Configuration of the screen only for PLC4 control

Fig. 10-5: Possible configurations of the controller texts in the function key menu



Properties of the function key -> Function key number: 341

General

Number	341
Labeling	Setup Var
Screen type	Setup screen1
... is subscreen of the function key	

Configuration and data connection

Configuration and data variable (from active control/station)

Setup screen 1 - PLC1 Control: PLC1

1 End position

Access level: (3) User

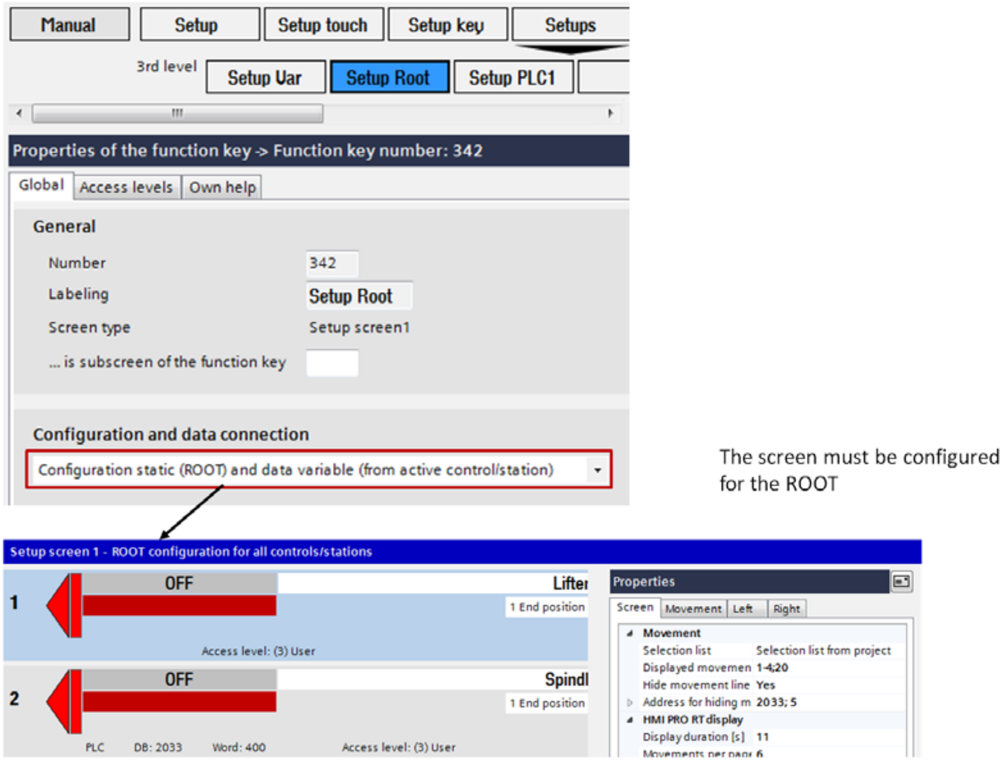
PLC DB: 2033 Word: 400 Access level: (3) User

Properties

Movement	
Selection list	Selection list from project
Displayed movemen	1-4;20
Hide movement line	Yes
Address for hiding m	2033; 5
HMI PRO RT display	
Display duration [s]	11
Movements per page	6

The screen must be configured for each control

Fig. 10-6: Configuration and data variable using setup screen 1 as example



Properties of the function key -> Function key number: 342

General

Number	342
Labeling	Setup Root
Screen type	Setup screen1
... is subscreen of the function key	

Configuration and data connection

Configuration static (ROOT) and data variable (from active control/station)

Setup screen 1 - ROOT configuration for all controls/stations

1 End position

Access level: (3) User

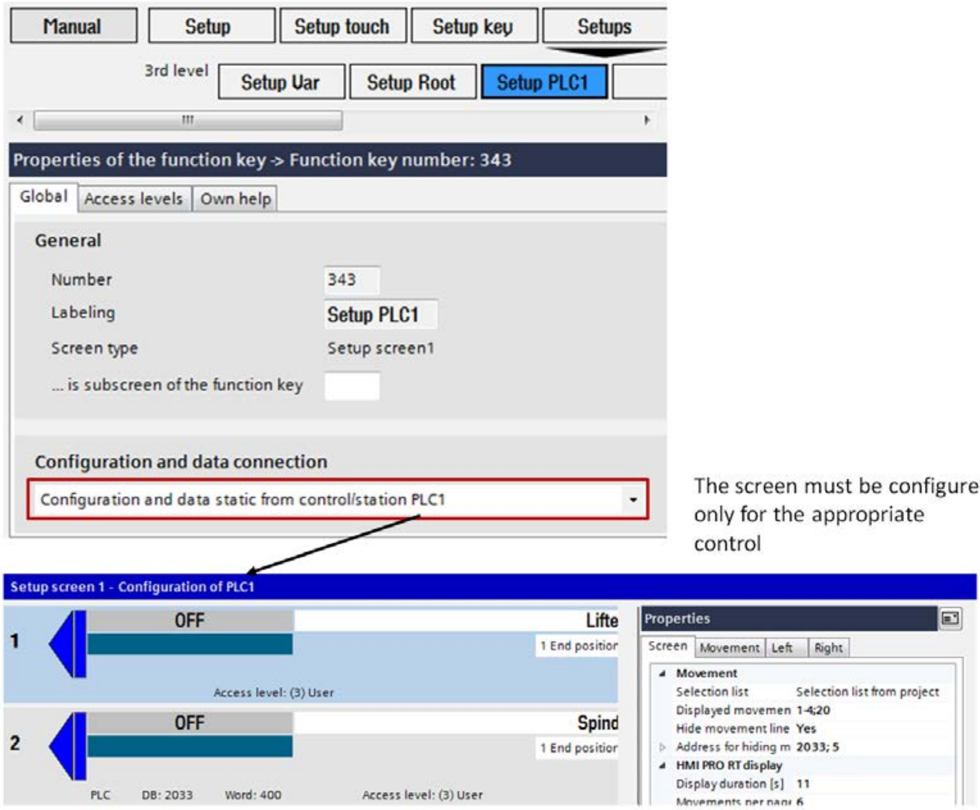
PLC DB: 2033 Word: 400 Access level: (3) User

Properties

Movement	
Selection list	Selection list from project
Displayed movemen	1-4;20
Hide movement line	Yes
Address for hiding m	2033; 5
HMI PRO RT display	
Display duration [s]	11
Movements per page	6

The screen must be configured for the ROOT

Fig. 10-7: Configuration static and data variable using setup screen 1 as example



Properties of the function key -> Function key number: 343

General

Number	343
Labeling	Setup PLC1
Screen type	Setup screen1
... is subscreen of the function key	

Configuration and data connection

Configuration and data static from control/station PLC1

The screen must be configured only for the appropriate control

Setup screen 1 - Configuration of PLC1

1 OFF Lifte 1 End positior Access level: (3) User

2 OFF Spind 1 End positior PLC DB: 2033 Word: 400 Access level: (3) User

Properties

Movement

Selection list	Selection list from project
Displayed movemen	1-4;20
Hide movement line	Yes
Address for hiding m	2033; 5

HMI PRO RT display

Display duration [s]	11
Movements per page	6

Fig. 10-8: Configuration and data from controller PLC1 using setup screen 1 as example

Additional HMI PRO screens

For additional HMI-PRO screens, **configuration static (ROOT) and data variable (from the active controller/station)** should always be set.

Creating a project with virtual stations from a standard project (1:1 assignment)

1. Create a standard project (1:1 assignment)
2. In the project navigation, under **Controller** ▸ **Controller connection** ▸ **Basic connection,** activate checkbox **Project with virtual stations (separate DBs)**.
3. Under **Controller** ▸ **Configuration of the virtual stations** configure all HMI PRO DBs that you wish to use in your project with virtual stations.
   Your project with virtual stations is created, and the standard project that you created in step 1, is accepted as ROOT configuration and is copied to all station-specific configurations.
   The field **Configuration and data connection** is displayed in the window **Properties of the function key** under the **Global** tab.
4. To configure the virtual stations, click on the required function keys one after the other, and then copy these to a free function key.
5. In dialog **Assignment function key 'Screen name',** select the configuration and data connection for the copied screen; i.e. which configuration and which data should be displayed after switching over to another station in HMI PRO RT.
   The following configurations are available in the dialog:

   - Configuration static (ROOT) and data variable (from active station)
   - Configuration and data variable (from active station)
   - Configuration and data static (from station...)

   If you use the controller blocks from TRANSLINE, then you must ensure that parameter **HMI_DB** must be supplied/parameterized with the HMI-PRO DB (number) of the currently selected station.
6. Configure your screens as required.

Interchanging the screen assignment in projects with virtual stations

1. Click on the required function key.
2. Copy the screen to a free function key.
3. In dialog **Assignment function key 'Screen name',** select the configuration and data connection for the copied screen; i.e. which configuration and which data should be displayed after switching over to another station in HMI PRO RT.
   The following configurations are available in the dialog:

   - Configuration static (ROOT) and data variable (from active station)
   - Configuration and data variable (from active station)
   - Configuration and data static (from station...)

   If you use the controller blocks from TRANSLINE, then you must ensure that parameter **HMI_DB** must be supplied/parameterized with the HMI-PRO DB (number) of the currently selected station.
   If required, delete the original screen.

10.4 PLC configuring

10.4.1 1:N

Every PLC must be configured as for a 1:1 connection.
When connecting an MPP, HT8 or HT2, you must observe that only the MPP of the controller whose data is displayed in HMI PRO can be enabled.

Data byte **LTLP_HMIProInterfaceData.net_Index** of the 1st station can be evaluated to determine the active PLC. The controller that is currently selected in HMI PRO (on the HMI PRO interface) is entered here.

Direct key screens

LTLP_DirectKey must always be actively called in each controller. The reason is the life bit, which is evaluated by HMI PRO.

Axis selection, operating mode blocks

You must ensure that the functions of the MPP are only active in the controller selected by HMI PRO/SINUMERIK Operate.

10.4.2 Virtual stations

The active virtual station can be evaluated using data byte **LTLP_HMIProInterfaceData.net_Index** [DB501.DBB49] of the 1st station The station that is currently selected in HMI PRO is entered here.

Direct key screens

LTLP_DirectKey must always be actively called for each virtual station. The reason is the life bit, which is evaluated by HMI PRO.

10.4.3 Configuration and data variable (from the active controller/station) or configuration static (ROOT) and data variable (from the active controller/station)

Depending on the selected virtual station, function blocks (e.g. **LTLP_DirectKey**) are executed to supply the MPP/HT8 etc. Byte **LTLP_HMIProInterfaceData.net_Index** [DB501.DBB49] of the 1st station can be used to identify which station is selected.

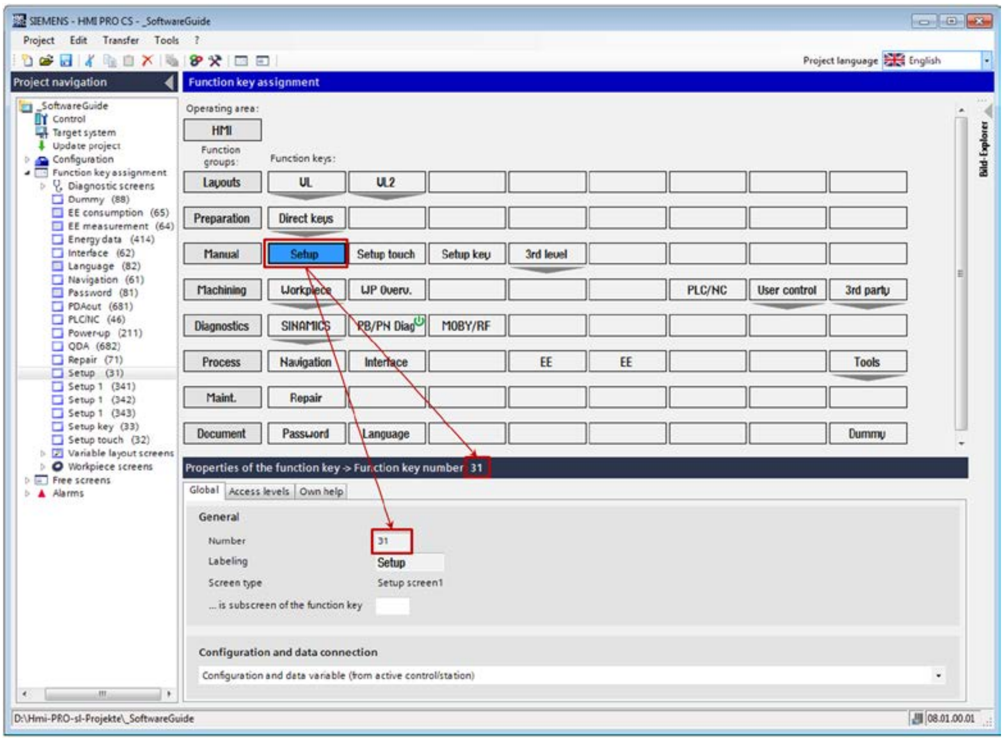
```
          L      DB501.DBB49     //LTLP_HMIProInterfaceData_station_1.net_Index
          L     1
          ==I
          =     M4.0             //"m_HMI1"
          L     DB501.DBB49      //LTLP_HMIProInterfaceData_station_1.net_Index
          L     2
          ==I
          =     M4.1

          U     M4.0             //"m_HMI1"
          SPBN  HMI2
          L     501              //LTLP_HMIProInterfaceData_station_1
          T     DB19.DBW128      //"MMC".MMC1_TRANS_DB
          SPA   screen

HMI2:     U     M4.2
          SPBN  screen
          L     511
          T     DB19.DBW128      //"MMC".MMC1_TRANS_DB
Diagram:        NOP              0
```

10.4.4 Configuration and data static from controllerX/station

Byte **LTLP_HMIProInterfaceData.net_Index** [DB501.DBB49] cannot be used here. Instead, use the screen identifier of the selected screen. Using the function key number you can determine which screen is selected.



Properties of the function key -> Function key number 31

General

Number: 31
Labeling: Setup
Screen type: Setup screen1
... is subscreen of the function key:

Configuration and data connection

Configuration and data variable (from active control/station)

Fig. 10-9: Softkey/function key number of a screen in the function key screen

Example of evaluation of the function key number (softkey) 31:
DB501.DBB82 = 1 corresponds to the 1st column of the horizontal softkey bar (upper and third levels in the function key assignment)
DB501.DBB83 = 3 corresponds to the 3rd line of the vertical softkey bar (lower level in the function key assignment)
This means that for DB501.DBW82 = W#16#0103 setup screen 1 is currently selected.

```
L      DB501.DBW 82          //DB for the selected station
L      w#16#0103
==I
=      M4.1                  //Setup screen 1 "cylinder" selected
```

PLC configuring of screens/dialogs with background function

These include the following screens:

- Workpiece overview
- Workpiece count
- Cycle time
- Tool life

The supply of the screens must be independent of which screen is currently selected in HMI PRO RT.

Axis selection, operating mode blocks

You must ensure that the functions of the MPP are only active in the active controller.

■

11

11 Using GRAPH blocks

Step sequence programming using GRAPH is suitable for programming sequential machine processes in PLC-controlled machines.
This chapter describes the use of GRAPH blocks from the TRANSLINE library for using the manual control via GRAPH in the HMI PRO setup screens.

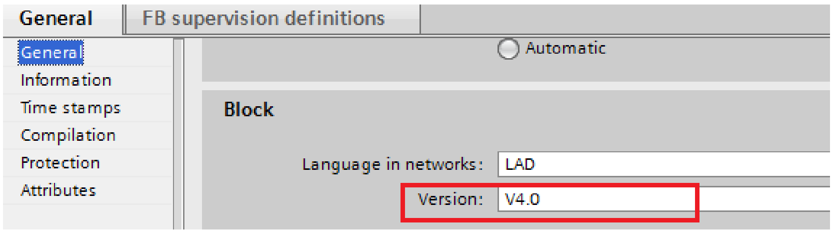
TRANSLINE library

The blocks, data types, and example DBs are stored in the TRANSLINE library under types/copy template in the Graph directory.

11.1 Specifications

In the GRAPH blocks used for the manual control, the setting **"Permanent processing of all interlocks in manual operation"** must be activated so that executability can be accordingly displayed.
The name of the instance of a GRAPH block should not exceed 40 characters in length because internally and in the parameterization no more than 40 characters are reserved. Note that the quotation marks at the beginning and end of the name in the parameterization must be taken into account. This means that only 38 characters are available for the actual instance name.

In the properties of the GRAPH block, under the **General** tab, you must set version V4.0.

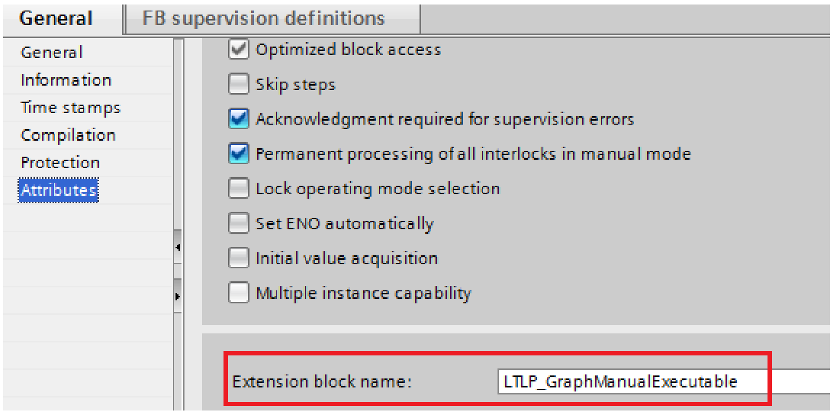


Language in networks: LAD
Version: V4.0

Fig. 11-1: Properties of the GRAPH block: setting the version

11.2 LTLP_GraphManualExecutable [FB562]

If you wish to execute the manual control in HMI PRO setup screens 1 and 2 via GRAPH, then in the properties of your GRAPH block, under the **Attribute** tab under **Name of the extension block** you must enter block **LTLP_GraphManualExecutable**. It is not permissible that the block is called in the PLC program.
For this purpose, in the interface of the GRAPH block, either a single-instance of the FB must be defined as I/O parameter – or a multi-instance in the static area of the GRAPH block.

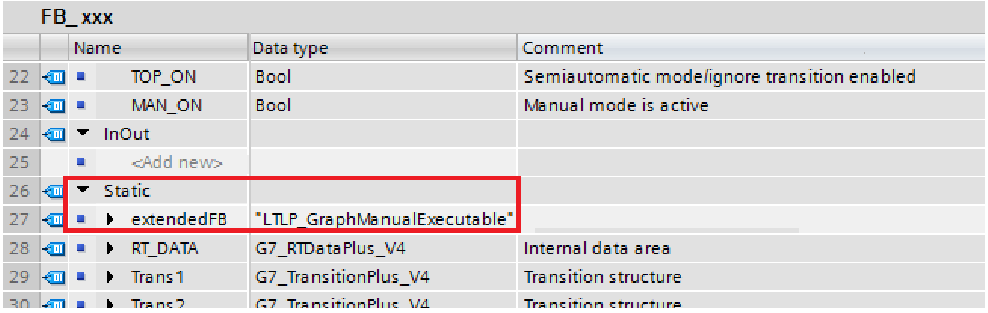


☑ Optimized block access
□ Skip steps
☑ Acknowledgment required for supervision errors
☑ Permanent processing of all interlocks in manual mode
□ Lock operating mode selection
□ Set ENO automatically
□ Initial value acquisition
□ Multiple instance capability

Extension block name: LTLP_GraphManualExecutable

Fig. 11-2: Integration of the LTLP_GraphManualExecutable as extension block

Program example: Integration as multi-instance

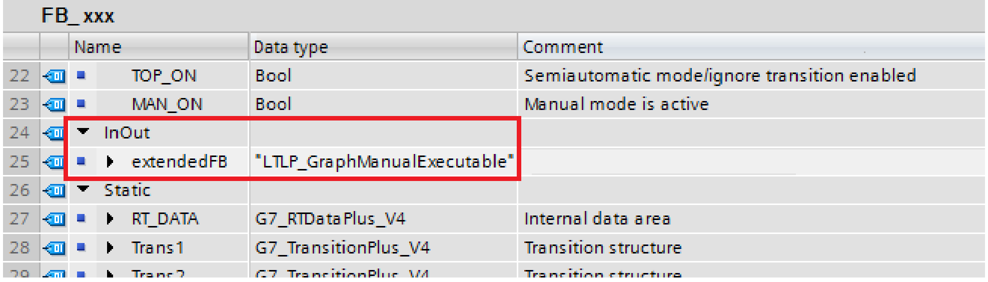


FB_xxx

	Name	Data type	Comment
22	TOP_ON	Bool	Semiautomatic mode/ignore transition enabled
23	MAN_ON	Bool	Manual mode is active
24	InOut		
25	<Add new>		
26	Static		
27	extendedFB	"LTLP_GraphManualExecutable"	
28	RT_DATA	G7_RTDataPlus_V4	Internal data area
29	Trans1	G7_TransitionPlus_V4	Transition structure
30	Trans2	G7_TransitionPlus_V4	Transition structure

Fig. 11-3: Integration as multi-instance of a GRAPH block

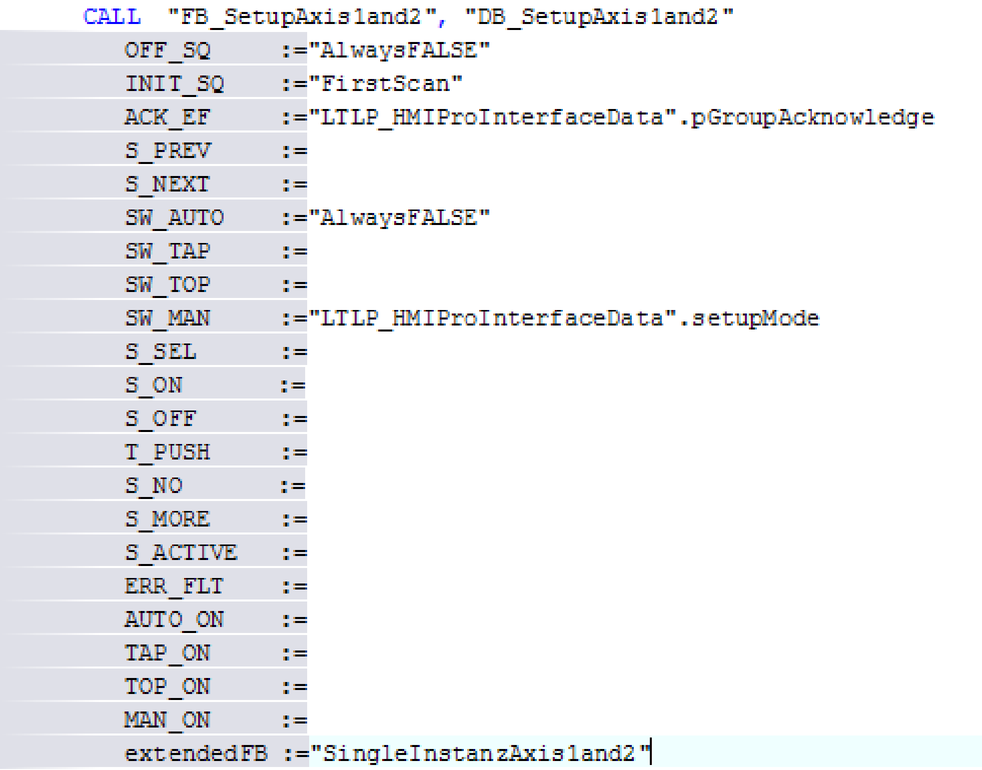
Program example: Integration as single-instance



FB_xxx

	Name	Data type	Comment
22	TOP_ON	Bool	Semiautomatic mode/ignore transition enabled
23	MAN_ON	Bool	Manual mode is active
24	▼ InOut		
25	▶ extendedFB	"LTLP_GraphManualExecutable"	
26	▼ Static		
27	▶ RT_DATA	G7_RTDataPlus_V4	Internal data area
28	▶ Trans1	G7_TransitionPlus_V4	Transition structure
29	▶ Trans2	G7_TransitionPlus_V4	Transition structure

Fig. 11-4: Integration as single-instance of a GRAPH block



```
CALL  "FB_SetupAxis1and2", "DB_SetupAxis1and2"
   OFF_SQ     :="AlwaysFALSE"
   INIT_SQ    :="FirstScan"
   ACK_EF     :="LTLP_HMIProInterfaceData".pGroupAcknowledge
   S_PREV     :=
   S_NEXT     :=
   SW_AUTO    :="AlwaysFALSE"
   SW_TAP     :=
   SW_TOP     :=
   SW_MAN     :="LTLP_HMIProInterfaceData".setupMode
   S_SEL      :=
   S_ON       :=
   S_OFF      :=
   T_PUSH     :=
   S_NO       :=
   S_MORE     :=
   S_ACTIVE   :=
   ERR_FLT    :=
   AUTO_ON    :=
   TAP_ON     :=
   TOP_ON     :=
   MAN_ON     :=
   extendedFB :="SingleInstanzAxis1and2"
```

Fig. 11-5: Example: Calling the GRAPH block as single instance

Parameter description

The block has no parameters.

11.3 LTLP_GraphManualInputParam [FC563]

This block transfers the data required to the **LTLP_GraphManualExecutable**. It must be called in the GRAPH blocks, which are written for the manual control, in the previous instruction.
The **GetInstanceName** standard function must also be called. The return value must be saved in the parameter **input.graphInstanceName** of the instance of the extension FB.

Program example: instance name is cyclically read

In this example, the instance name of the GRAPH block is cyclically read.

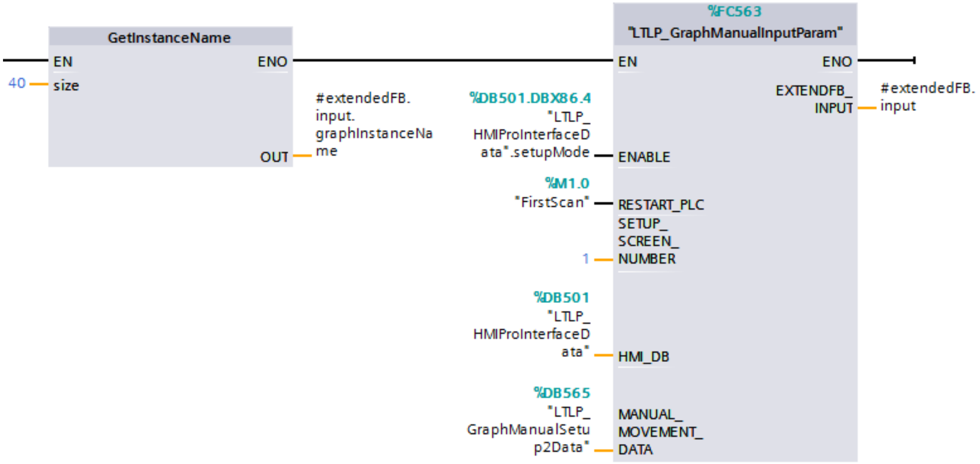


Fig. 11-6: Instance name of the GRAPH block is cyclically read

Program example: Reading the instance name for a PLC restart

In this example, the instance name of the GRAPH block is only read when the PLC restarts.

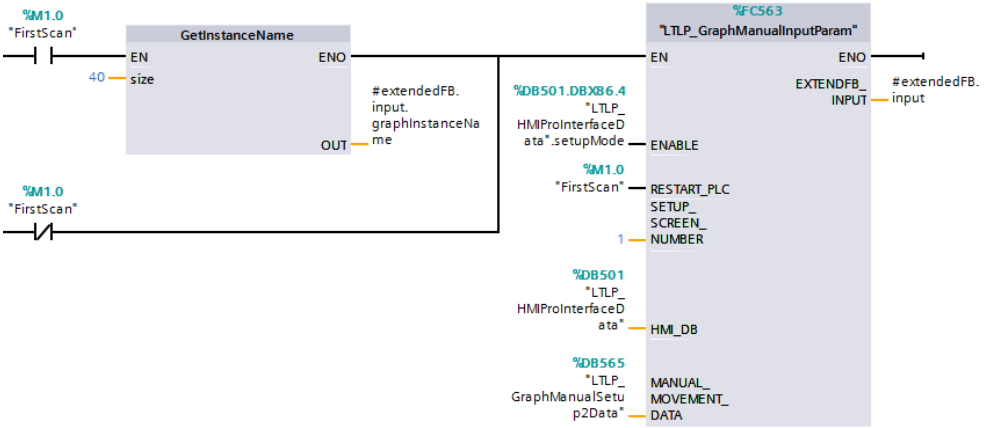
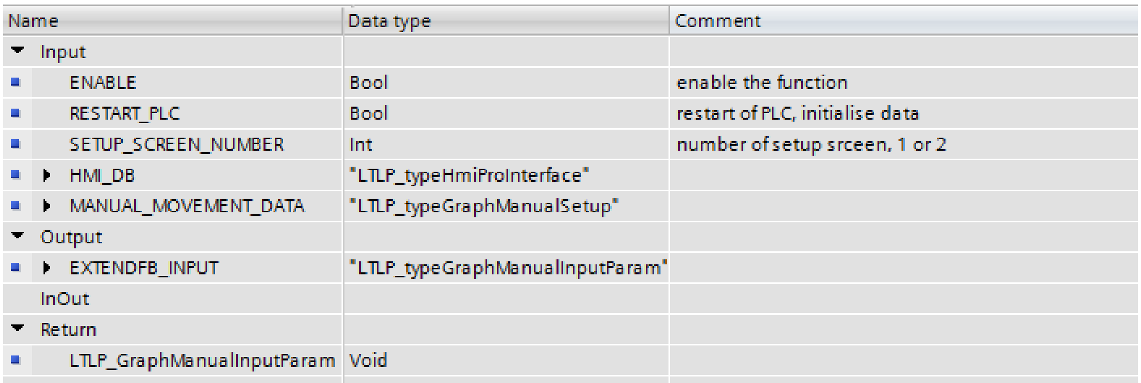


Fig. 11-7: Read instance name of the GRAPH block when the PLC restarts



Name	Data type	Comment
▼ Input		
■ ENABLE	Bool	enable the function
■ RESTART_PLC	Bool	restart of PLC, initialise data
■ SETUP_SCREEN_NUMBER	Int	number of setup srceen, 1 or 2
■ ▶ HMI_DB	"LTLP_typeHmiProInterface"	
■ ▶ MANUAL_MOVEMENT_DATA	"LTLP_typeGraphManualSetup"	
▼ Output		
■ ▶ EXTENDFB_INPUT	"LTLP_typeGraphManualInputParam"	
InOut		
▼ Return		
■ LTLP_GraphManualInputParam	Void	

Fig. 11-8: LTLP_GraphManualInputParam parameter

Description of input parameters

Table 11-1: Input parameter LTLP_GraphManualInputParam

Parameter	Data type	Description
ENABLE	Bool	Enable signal to process the extension block for manual selection.
RESTART_PLC	Bool	Request to restart (FirstScan)
SETUP_SCREEN_NUMBER	Int	Number of the setup screen type used (1 or 2)
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.
MANUAL_MOVEMENT_DATA	LTLP_typeGraph ManualSetup	Contains the parameterization of all movements for manual control of setup screen 1 or 2

Description of output parameters

Table 11-2: Output parameter LTLP_GraphManualInputParam

Parameter	Data type	Description
EXTENDFB_INPUT	LTLP_typeGraphManual InputParam	The static variable input of the instance of the extension FB should be specified here.

Description of the return value

No value is returned.

11.4 LTLP_GraphManualOutputParam [FC564]

This block transfers the required data of the **LTLP_GraphManualExecutable** to HMI PRO. It must be called in the GRAPH blocks, which are written for the manual control, in the next instruction.

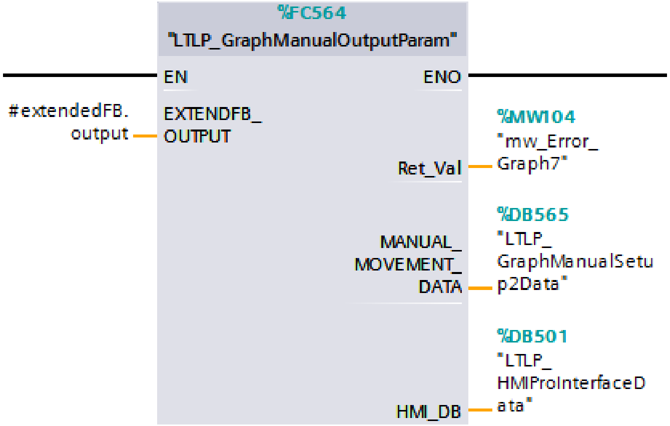
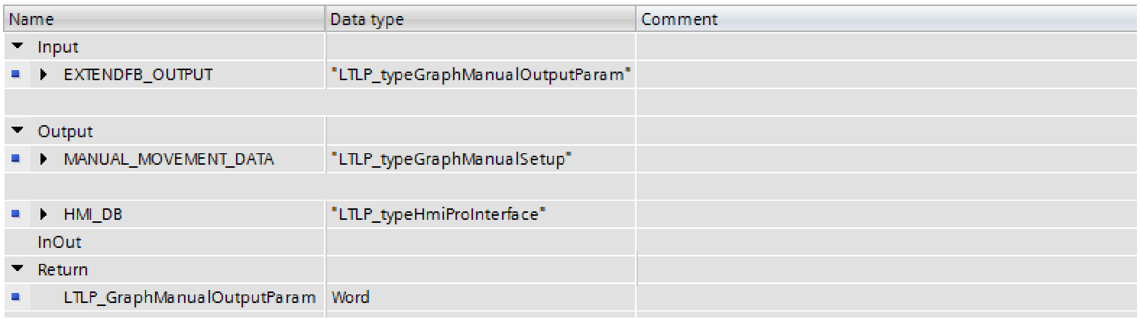


Fig. 11-9: Call LTLP_GraphManualOutputParam



Name	Data type	Comment
▼ Input		
▶ EXTENDFB_OUTPUT	"LTLP_typeGraphManualOutputParam"	
▼ Output		
▶ MANUAL_MOVEMENT_DATA	"LTLP_typeGraphManualSetup"	
▶ HMI_DB	"LTLP_typeHmiProInterface"	
InOut		
▼ Return		
LTLP_GraphManualOutputParam	Word	

Fig. 11-10: LTLP_GraphManualOutputParam parameter

Description of input parameters

Table 11-3: Input parameter LTLP_GraphManualOutputParam

Parameter	Data type	Description
EXTENDFB_OUTPUT	LTLP_typeGraphManual OutputParam	The static variable output of the instance of the extension FB should be specified here.

Description of output parameters

Table 11-4: Output parameter LTLP_GraphManualOutputParam

Parameter	Data type	Description
MANUAL_MOVEMENT_ DATA	LTLP_typeGraphManual Setup	Contains the parameterization of all movements for manual control of setup screen 1 or 2
HMI_DB	LTLP_typeHMIPro Interface	Interface between HMI PRO and PLC user program. **LTLP_HMIProInterfaceData** [DB501] is the default setting.

Description of the return value

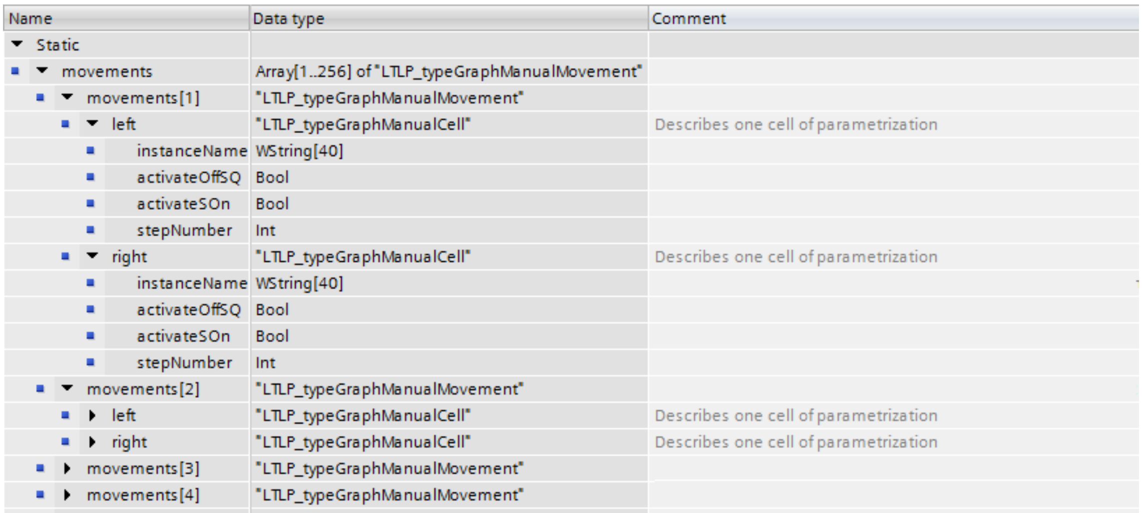
WORD
Return value of the expansion block

0x0000	No error
0x7001	Message: No manual operation
0x0010	Error: GRAPH version not correct (not V4)

11.5 LTLP_GraphManualSetup1Data [DB564] and LTLP_GraphManualSetup2Data [DB565]

These two data blocks are used to parameterized the manual control via GRAPH in setup screens 1 and 2.
For each movement member, which should execute a function, parameterize the movement left **and** the movement right and specify the instance name of the GRAPH block and the corresponding step number.

Parameterization



Name	Data type	Comment
▼ Static		
▼ movements	Array[1..256] of "LTLP_typeGraphManualMovement"	
▼ movements[1]	"LTLP_typeGraphManualMovement"	
▼ left	"LTLP_typeGraphManualCell"	Describes one cell of parametrization
instanceName	WString[40]	
activateOffSQ	Bool	
activateSOn	Bool	
stepNumber	Int	
▼ right	"LTLP_typeGraphManualCell"	Describes one cell of parametrization
instanceName	WString[40]	
activateOffSQ	Bool	
activateSOn	Bool	
stepNumber	Int	
▼ movements[2]	"LTLP_typeGraphManualMovement"	
▶ left	"LTLP_typeGraphManualCell"	Describes one cell of parametrization
▶ right	"LTLP_typeGraphManualCell"	Describes one cell of parametrization
▶ movements[3]	"LTLP_typeGraphManualMovement"	
▶ movements[4]	"LTLP_typeGraphManualMovement"	

Fig. 11-11: Parameterization of the LTLP_GraphManualSetup1Data or LTLP_GraphManualSetup2Data

Description of the parameters

Table 11-5: Parameter LTLP_GraphManualSetup1Data/ LTLP_GraphManualSetup2Data

Parameter	Data type	Description
instanceName	WString[40]	The instance name of the GRAPH block should be specified here. The quotation marks "" should be specified, as the function GetInstanceName supplies quotation marks.
stepNumber	Int	Step number of the GRAPH block that must be activated for the manual function.
activateOffSQ	Bool	
activateSOn	Bool	

■

12

12 Error and operating messages

12.1 DB_HMI_ALARM_SERVER [DB126]

The **DB_HMI_ALARM_SERVER** can manage up to 16000 alarms and messages. Alarm and message texts are configured in the HMI PRO CS in the project navigation under **Configuration ▸ Alarms ▸ Constant alarms and messages (all alarm sources) ▸ Alarms and messages.**

As with the standard alarm system, texts are displayed in the message window on alarm screens and in the system message line. The numbers (700000 to 715999) are displayed corresponding to the bit set in the **DB_HMI_ALARM_SERVER**.

The areas for alarms or messages can be configured with HMI PRO CS. Other supplementary conditions for evaluating errors can also be defined.

For more information about configuring alarms and messages, see the online help for HMI PRO CS.

12.2 Error and operating messages

12.2.1 General information

DB_HMI_ALARM_SERVER is used as the interface for error and operating messages. The block should be created in the TIA Portal as data block with a size that corresponds to that configured in HMI PRO CS. Error and operating messages are configured in the HMI PRO CS in the project navigation under **Configuration ▸ Alarms ▸ Constant alarms and messages (all alarm sources) ▸ Alarms and messages.**
For more information about configuring alarms and messages, see the online help for HMI PRO CS.

12.2.2 Acknowledgment concept

The following acknowledgement concepts are implemented for error and operating messages:

Operating messages

Operating messages are intended for displaying normal operating states as information for the operator. Acknowledgment signals are, therefore, not required. An entry is made in the operating message buffer for incoming and outgoing messages. The operator panel (HMI device) maintains an up-to-date log of existing operating messages using the identifiers **Operating message arrived** and **Operating message gone**.

Error messages

Error messages display error states at the machine, which could result in 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 one hand, by the order in which they are entered in the error message buffer and on the other, by the time stamp, which is assigned to each entry. If the cause of the error has disappeared, the error message is only cleared if the user has acknowledged the message.

■

A

A Appendix, Change directory

A.1 Edition 2018

Document creation for SIMATIC S7-1500

■

I

I Block index

I.1 Symbolic block names

I.2 Block numbers

DB

FB

FC

To
Siemens AG

TRANSLINE Support
P.O. Box 10 60 26

D-70049 Stuttgart (Germany)

Fax +49 (0) 711 / 137-2838 [Documentation]
mailto:transline-support.sdw.iiadt.ger@.siemens.com

Suggestions

Corrections

For Publication/Manual:

Solutions for Powertrain

TRANSLINE sl - Software Guide
PLC Programming Guide
SINUMERIK Integrate Run MyHMI /PRO
(S7-1500)

Manufacturer documentation

Sender

Name:

Address of your Company/Dept.

Street: ________________________________

Zip code: __________ Location: ________________

Telephone: ________ / ____________________

Fax _______ / __________________________

Manual

Available at:
DF MC-SIOS Extranet

Edition 2018

If you come across any misprints in this document, please let us know using this form. Suggestions for improvements are also welcome.

Suggestions and/or corrections

Get more information

Solutions for Powertrain:
www.siemens.com/TRANSLINE

Automotive Manufacturing:
www.siemens.com/automotive

SINUMERIK CNC Automation System:
www.siemens.com/sinumerik

SINAMICS Drive Technology:
www.siemens.com/sinamics

Motion Control Systems and Solutions for Production Machines and Machine Tools:
www.siemens.com/motioncontrol

Industry Online Support (Service and Support):
www.siemens.com/online-support

IndustryMall:
www.siemens.com/industrymall

Siemens AG
Digital Factory
Motion Control
Postfach 3180
91050 Erlangen
Germany

Scan the QR-Code for information regarding Solutions for Powertrain