

**SIEMENS**

*Ingenuity for life*



# SINUMERIK ONE MCP commissioning

SINUMERIK ONE

<https://support.industry.siemens.com/cs/ww/en/view/109794252>

Siemens  
Industry  
Online  
Support



## Legal information

### Using the application examples

The application examples illustrate the solution of automation tasks with the interaction of several components in the form of text, graphics and/or software blocks. The application examples are a free service provided by Siemens AG and/or a subsidiary company of Siemens AG ("Siemens"). They are non-binding and do not claim to be complete and functional with regard to configuration and equipment. The application examples do not represent customer-specific solutions, but rather, only provide help for typical jobs. You yourself are responsible for the proper and safe operation of products in compliance with the applicable regulations. You must check the function of the respective application example and adapt it specifically to your system.

Siemens grants you the non-exclusive, non-sub-licensable and non-transferable right of use of the application examples by professionally trained personnel. Any change to the application examples is made at your own risk. Transfer to third parties or duplication of the application examples or extracts thereof is only permitted in combination with your own products. The application examples do not necessarily undergo the usual tests and quality checks of a paid product, may contain functional and performance defects and may be subject to errors. You are responsible for ensuring that the application examples are used in such a way that any malfunctions do not lead to property damage or personal injury.

### Disclaimer of liability

Siemens accepts no liability, regardless of the legal grounds, in particular for the usability, availability, completeness and correctness of the application examples, as well as associated notes and configuration or performance data and damage resulting from this. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence or culpable death, injury or health impairment, non-compliance with a guarantee, fraudulent non-disclosure of a defect or culpable breach of material contractual obligations. The claims for compensation for the breach of essential contractual obligations are, however, limited to the foreseeable damage typical for the type of contract, except in the event of intent or gross negligence or death or injury or health impairment. The above provisions do not entail a change in the burden of proof to your disadvantage. You release Siemens from any existing or pending claims of third parties in this context, unless Siemens has mandatory liability. By using the application examples, you acknowledge that Siemens cannot be made liable for any claims beyond the liability clause described.

### Additional notes

Siemens reserves the right to make changes to these application examples at any time without prior notice. In cases of discrepancies between the suggestions in the application examples and other Siemens publications, such as catalogs, the content of the other documentation shall take precedence.

In addition to this, the Siemens terms and conditions of use apply (<https://support.industry.siemens.com>).

### Security information

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

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

Customers are solely responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place. Additionally, Siemens recommendations on appropriate security measures should be taken into account. For additional information on Industrial Security, please visit:

<https://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens expressly recommends that updates are carried out as soon as they become available and that only the current product version is ever used. Use of (older) product versions that are no longer supported, and failure to apply the latest updates, may increase the customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: <https://www.siemens.com/industrialsecurity>.

# Table of contents

<b>Legal information</b> .....	<b>2</b>
<b>1 Introduction</b> .....	<b>4</b>
1.1 Task.....	4
1.2 Principle of operation.....	4
1.3 Components used .....	4
<b>2 Engineering</b> .....	<b>5</b>
2.1 Hardware configuration .....	5
2.2 Configuration .....	5
2.2.1 Configuring PLC data types for the input and output image .....	5
2.2.2 Assignment of the MCP inputs/outputs .....	8
Assignment via a global data block .....	8
Assignment via the PLC tag table .....	9
2.2.3 Configuring "LBP_ConfigBP" in OB "Startup" .....	10
2.2.4 Configuring "ModMCPCtrlMilling" .....	11
2.2.5 Information about input parameters .....	13
2.2.6 Configuring block LBP_Powerride .....	15
<b>3 Appendix</b> .....	<b>16</b>
3.1 Service and Support.....	16
3.2 Links and references .....	17
3.3 Change documentation .....	17

# 1 Introduction

## 1.1 Task

Integrating SINUMERIK ONE MCP 1900, MCP 2200, MCP 2200c, MCP 2400 or customized MCP - connected to the Industrial Ethernet - in the PLC of SINUMERIK ONE.

## 1.2 Principle of operation

The hardware is integrated using special, specified data types, which are transferred to a function block (ModMCPCtrlMilling), which in turn ensures that the signals are distributed to the HMI, axis and channel interface.

## 1.3 Components used

This application example was created with these hardware and software components:

Table 1-1

Component	Number	Article number	Note
TIA Portal V16	1		
SINUMERIK ONE STEP 7 Toolbox V16	1		
SINUMERIK ONE 1760	1	6FC5317-6AA00-0AA0	
MCP 2400c	1	6FC5303-0AP51-0BA0	
24" Touch Panel	1		

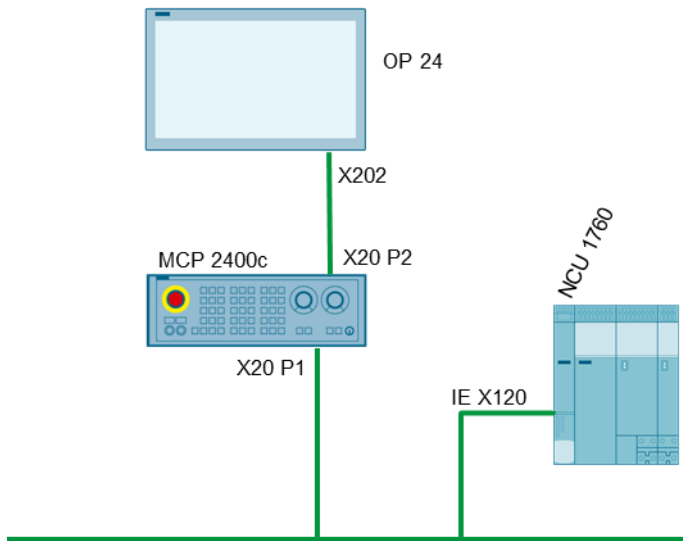
This application example consists of the following components:

Table 1-2

Component	File name	Note
TIA library		
TIA project example		

# 2 Engineering

## 2.1 Hardware configuration



## 2.2 Configuration

### 2.2.1 Configuring PLC data types for the input and output image

Create new data types in the project navigation in the TIA Portal under "PLC data types". This is shown as example for mcpIn2400c and mcpOut2400c in the following diagram.

Fig. 2-1

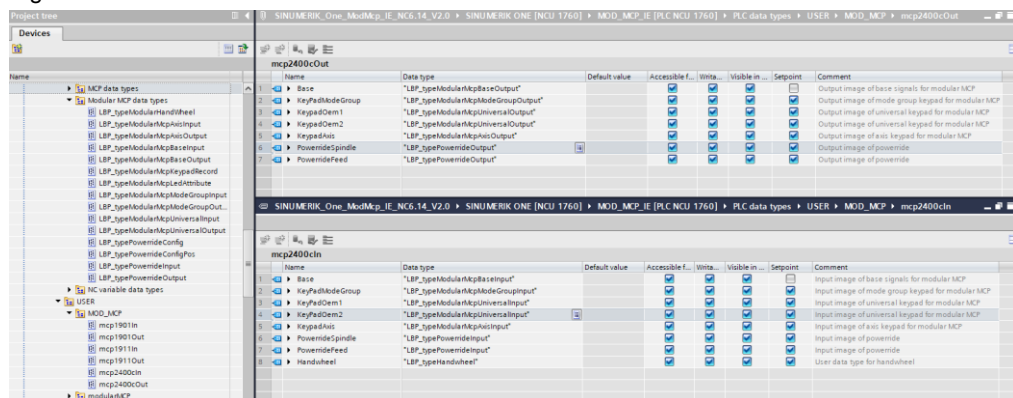
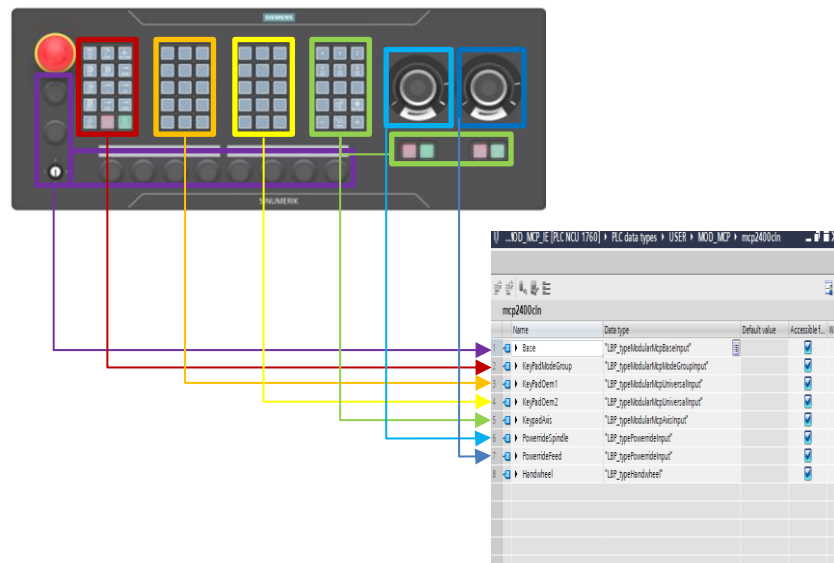




Fig. 2-2



The data types for the input/output image always reflect the existing hardware.

The input image always comprises

- The Base Input Block
- The individual keypads
- The powerrides
- The handwheels

This sequence is fixed, the number of keypads and powerrides is variable. The input images of the handwheels are always transferred, even if there are no handwheels. See Fig. 2-2 for the assignment of the existing hardware to the PLC data types.

The output image always comprises

- The Base Output Block
- The individual keypads
- The powerrides

This sequence is also fixed, the number of keypads and powerrides is variable. There are no handwheels in the output image. The assignment of the existing hardware is analogous to the input image, see Fig. 2-2.

There are no gaps in the PLC data type or placeholders if there are only 2 or 3 keypads or only one powerride.

In the example, data types mcpIn2400c and mcpOut2400c comprise the "Modular MCP Data Types" provided with the base program.

Fig. 2-3

Name	Data type	Default value	Accessible
1 Base	"LBP_typeModularMcpBaseInput"		<input checked="" type="checkbox"/>
2 KeyPadModeGroup	"LBP_typeModularMcpModeGroupInput"		<input checked="" type="checkbox"/>
3 KeyPadOem1	"LBP_typeModularMcpUniversalInput"		<input checked="" type="checkbox"/>
4 KeyPadOem2	"LBP_typeModularMcpUniversalInput"		<input checked="" type="checkbox"/>
5 KeyPadAxis	"LBP_typeModularMcpAxisInput"		<input checked="" type="checkbox"/>
6 PowerrideSpindle	"LBP_typePowerrideInput"		<input checked="" type="checkbox"/>
7 PowerrideFeed	"LBP_typePowerrideInput"		<input checked="" type="checkbox"/>
8 Handwheel	"LBP_typeHandwheel"		<input checked="" type="checkbox"/>

The names of the individual data type components can be freely selected. Names used in the example

KeypadModeGroup	keypad to control JOG, AUTO, REF etc.
KeypadOEM1	keypad to be used by the OEM
KeypadOEM2	keypad to be used by the OEM
KeypadAxis	keypad to select axes and traversing keys etc.
Powerride1	1 Powerride
Powerride2	2 Powerride
handwheel	1 and 2 handwheels

## 2.2.2 Assignment of the MCP inputs/outputs

To be able to use the inputs/outputs of the MCP in the PLC program they still have to be assigned.

- Either via a global data block
- Or directly assigned via the PLC tag table

### Assignment via a global data block

Fig. 2-4

The screenshot shows the SIMATIC Manager interface. On the left, the 'Project tree' displays the hierarchy: SINUMERIK ONE [NCU 1760] > MOD\_MCP\_IE [PLC NCU 1760] > Program blocks > User > myMCP [DB64]. The main window shows the 'myMCP' data block configuration table.

Name	Data type	Offset	Start value	Retain
Static				
in	"mcp2400cin"	0.0		
Base	"LBP_typeModularMcpBaseInput"	0.0		
KeyPadModeGroup	"LBP_typeModularMcpModeGroupInput"	6.0		
KeyPadOem1	"LBP_typeModularMcpUniversalInput"	10.0		
KeyPadOem2	"LBP_typeModularMcpUniversalInput"	14.0		
KeypadAxis	"LBP_typeModularMcpAxisInput"	18.0		
PowerideSpindle	"LBP_typePowerideInput"	22.0		
PowerideFeed	"LBP_typePowerideInput"	24.0		
Handwheel	"LBP_typeHandwheel"	26.0		
out	"mcp2400cout"	30.0		
Base	"LBP_typeModularMcpBaseOutput"	30.0		
KeyPadModeGroup	"LBP_typeModularMcpModeGroupOutput"	32.0		
KeyPadOem1	"LBP_typeModularMcpUniversalOutput"	36.0		
KeyPadOem2	"LBP_typeModularMcpUniversalOutput"	40.0		
KeypadAxis	"LBP_typeModularMcpAxisOutput"	44.0		
PowerideSpindle	"LBP_typePowerideOutput"	48.0		
PowerideFeed	"LBP_typePowerideOutput"	54.0		

In the example, a global data block with the name "myMCP" was created under "Program Blocks" in subgroup "User". It is not permissible that this data block is optimized from a memory perspective. An "in" range and an "out" range are created in the data block. This range is then assigned to the data type that was created.

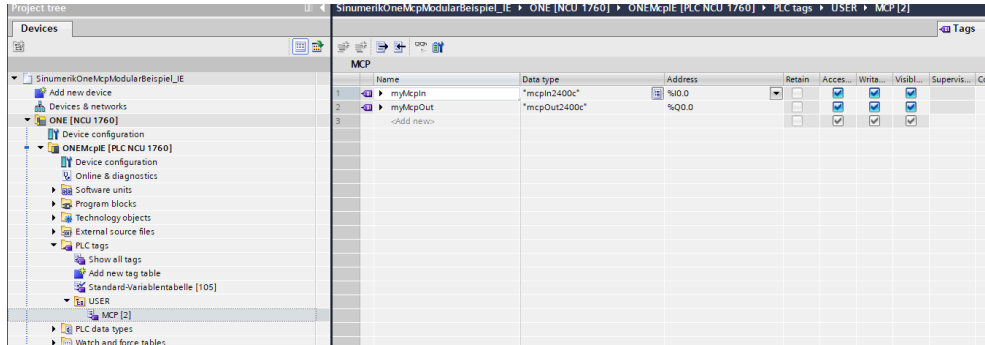
"in" -> mcp2400cin and "out" -> mcp2400cout.

You can view the complete assignment by opening ranges "in" / "out".



Assignment via the PLC tag table

Fig. 2-5

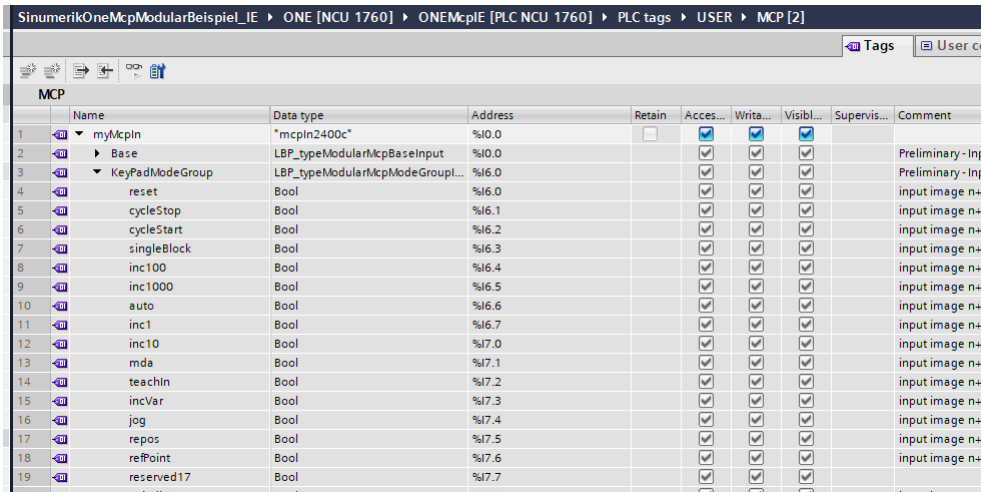


In the example, a USER subgroup was created in the project navigation under "PLC Tags". A PLC tag table "MCP" was created in this subgroup.

In this subgroup, name "myMcpIn", data type "mcpIn2400c" was assigned hardware address "I0.0" – and name "myMcpOut", data type "mcpOut2400c" was assigned hardware address "Q0.0".

You can completely view the assignment by opening the identifier.

Fig. 2-6



### 2.2.3 Configuring "LBP\_ConfigBP" in OB "Startup"

In this configuration, the MCP is operated at X120 (IE = Industrial Ethernet).

Fig. 2-7

```

CALL "LBP_ConfigBP"
MCPNum      :=1
MCP1In      :="myMcpIn"
MCP1Out     :="myMcpOut"
MCP1BusAdr  :=192
MCP1Timeout :=S5T#700MS
MCP2In      :=NULL
MCP2Out     :=NULL
MCP2BusAdr  :=0
MCP2Timeout :=S5T#700MS
MCP1Stop    :=FALSE
MCP2Stop    :=FALSE
MCP1NotSend :=FALSE
MCP2NotSend :=FALSE
MCPBusType  :=B#16#55
HTIf        :=0
HTIn        :=P#T80 0

```

You can see the call of LBP\_ConfigBP in the startup OB in Fig 2-6; this assigns the MCP input/output signals via the PLC tag table.

Fig. 2-8

```

CALL "LBP_ConfigBP"
MCPNum      :=1
MCP1In      :="myMCP".in
MCP1Out     :="myMCP".out
MCP1BusAdr  :=192
MCP1Timeout :=S5T#700MS
MCP2In      :=NULL
MCP2Out     :=NULL
MCP2BusAdr  :=0
MCP2Timeout :=S5T#700MS
MCP1Stop    :=FALSE
MCP2Stop    :=FALSE
MCP1NotSend :=FALSE
MCP2NotSend :=FALSE
MCPBusType  :=B#16#55
HTIf        :=0
HTIn        :=P#T80 0

```

You can see the call of LBP\_ConfigBP in the startup OB in Fig 2-7; this assigns the MCP input/output signals via the global data block.

Interconnecting "LBP\_ConfigBP"

- MCPNum: Number of connected machine control panels.
- MCP1In: The "in" section of the global data block (myMCP.in) or the PLCTag (myMcpIn) is interconnected here for the inputs of the MCP.
- MCP1Out: Just the same as for the input, the "out" section of the global data block (myMCP.Out) or the PLCTag (myMcpOut) for the outputs of the MCP is interconnected here.
- MCP1BusAdr: The address set at the DIP switches of the MCP is specified here. Refer to the MCP documentation for a description of the DIP switches.
- MCPBusType: Bus type 5 IE is selected here
- Regarding the MCP, all other parameters can be left at their preassigned default settings

### 2.2.4 Configuring "ModMCPCtrlMilling"

Block "ModMCPCtrlMilling" is used to transfer data from the machine operator panel (MCP milling version) to the NC / PLC interface. ModMCPCtrlMilling is executed as function block (FB). The block is written in SCL and is not know-how protected; users can adapt the block to address their specific requirements.

Fig. 2-9 ModMCPCtrlMilling

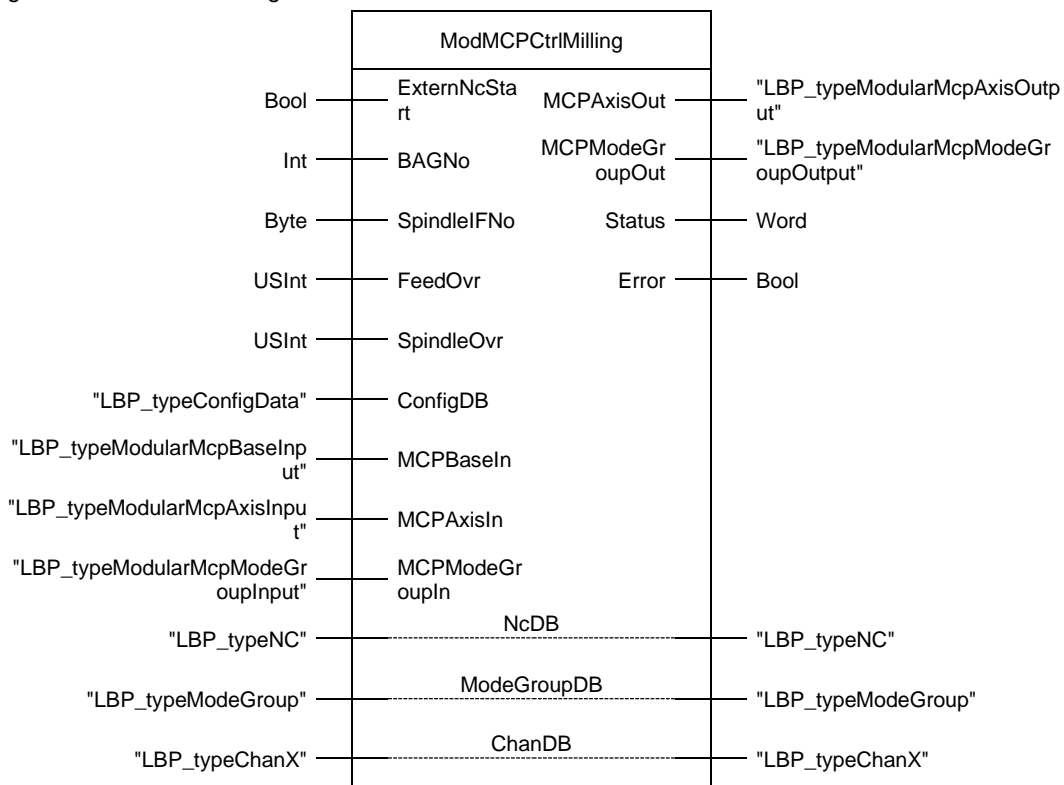


Table 2-1: Parameters of ModMCPCtrlMilling

Name	P type	Data type	Comment
ExternNcStart	IN	Bool	Signal from an external NC start key is internally OR'ed with the NC start key
BAGNo	IN	Int	Number of mode group to which the mode signals are transferred
SpindleIFNo	IN	Byte	Number of the axis/spindle in which the spindle- data is transferred
FeedOvr	IN	USInt	Position of the feed override switch in the Gray code format
SpindleOvr	IN	USInt	Position of the spindle override switch in the Gray code format
ConfigDB	IN	"LBP_typeConfigData"	Data block "LBP_ConfigData" of the base program with the global configuration data.
MCPBaseln	IN	"LBP_typeModularMcpBaselnput"	Basis component of the input image of the SINUMERIK ONE MCP, here in example "myMcpIn.Base"
MCPAxisIn	IN	"LBP_typeModularMcpAxisInput"	Input image of the Axis keypad, here in the example "myMcpIn.KeypadAxis"
MCPModeGroupIn	IN	"LBP_typeModularMcpModeGroupInput"	Input image of the BAG (mode group) keypad, here in the example "myMcpIn.KeyPadModeGroup"
MCPAxisOut	OUT	"LBP_typeModularMcpAxisOutput"	Output image of the Axis keypad, here in the example "myMcpOut.KeypadAxis"
MCPModeGroupOut	OUT	"LBP_typeModularMcpModeGroupOutput"	Output image of the BAG (mode group) keypad, here in the example "myMcpout.KeyPadModeGroup"
Status	OUT	Word	Block error messages as hexadecimal value, see Table xx
Error	OUT	Bool	Block error occurred, evaluate the status.

Name	P type	Data type	Comment
NcDB	IN_OUT	"LBP_typeNC"	Base program data block, NC signals "LBP_NC"
ModeGroupDB	IN_OUT	"LBP_typeModeGroup"	Base program data block in which the channel signals are transferred, here in the example "LBP_Chan1"
ChanDB	IN_OUT	"LBP_typeChanX"	Base program data block in which the channel signals are transferred, here in the example "LBP_Chan1"

Number	Type <sup>1)</sup>	Description
16#7000	S	No error
16#7001	E	Invalid BAG (mode group) number
16#7002	E	Channel is invalid
16#7003		

<sup>1)</sup>S=Status / E= Error

### 2.2.5 Information about input parameters

- **ExternNcStart:**  
Signal from an external NC start key is internally OR'ed with the NC start key. This input can be used for the NC start key in the Powerride. In the project example, a signal was implemented in block "PowerRideLedControl".
- **BAGNo:**  
Number of the mode group in which the mode signals are transferred
- **SpindleIFNo:**  
Number of the axis/spindle to which the spindle data is transferred (number of the associated machine axis)
- **FeedOvr., SpindleOvr:**  
Position of the override switch in the Gray code format  
The Powerrides on the MCP are not evaluated in this block. The Powerrides are parameterized and evaluated in a separate block.
- **ConfigDB:**  
Data block "LBP\_ConfigData" of the base program with the global configuration data.

- **MCPBaseIn:**  
Basis component of the input image of SINUMERIK ONE MCP, in the example "myMcpIn.Base" or "myMcpIn.Base"
- **MCPAxisIn:**  
Input image of the Axis keypad, in the example "myMcp.In.KeypadAxis" or "myMcpIn.KeypadAxis"
- **MCPModeGroupIn:**  
Input image of the BAG (mode group) keypad, in the example "myMcp.In.KeyPadModeGroup" or "myMcpIn.KeyPadModeGroup"
- **MCPAxisOut:**  
Output image of the Axis keypad, in the example "myMcp.Out.KeypadAxis" or "myMcpOut.KeypadAxis"
- **MCPModeGroupOut:**  
Output image of the BAG (mode group) keypad, in the example "myMcp.out.KeyPadModeGroup" or "myMcpout.KeyPadModeGroup"
- **Status:**  
Block error messages as hexadecimal values, see Table xx??
- **Error:**  
Block error occurred, evaluate the status.
- **NcDB :**  
Base program data block, NC signals "LBP\_NC"
- **ModeGroupDB:**  
Data block of the base program, operating mode signals, signals "LBP\_ModeGroup"
- **ChanDB:**  
Base program data block in which the channel signals are transferred, here in the example "LBP\_Chan1"



## 2.2.6 Configuring block LBP\_Powerride

Block LBP\_Powerride is included in the SINUMERIK ONE STEP 7 Toolbox. This block is used to process the input/output signals of the Powerride. The configuration of the Powerride is saved in a configuration data block and transferred to the block interface. Additional information is provided in the block documentation in the online help in the TIA Portal.

```
CALL "LBP_Powerride", "LBP_Powerride_DB"
  enable           := "PowerrideDaten".feedOverrideEnable
  preset          := "PowerrideDaten".feedOverridePreset
  forceSwitchPos1 := "PowerrideDaten".feedOverrideForcePos1
  pushButtonLED   := "PowerrideDaten".feedOverridePushButtonLed
  presetValue     := "PowerrideDaten".feedOverridePresetValue
  valid           := "PowerrideDaten".feedOverrideValid
  error           := "PowerrideDaten".feedOverrideError
  pushButton      := "PowerrideDaten".feedOverridePushButton
  ovrFactor       := "PowerrideDaten".feedOverrideOvrFactor
  grayCode        := "PowerrideDaten".feedOverrideGrayCode
  switchPos       := "PowerrideDaten".feedOverrideSwitchPos
  status          := "PowerrideDaten".feedOverrideStatus
  deviceIn        := "myMcpIn".PowerrideFeed
  deviceOut       := "myMcpOut".PowerrideFeed
  configuration    := "LBP_PowerrideConfig".configFeed
```

In the project example, the input-output signals of the block are combined in a "PowerrideData" data block.

Signals that are used in the example:

- enable: Must be supplied with "True" in order to activate Powerride.
- pushButtonLED: Switches on and switches off the LED
- pushButton: Signal as to whether the Powerride was pressed
- grayCode: Output signal of the Powerride position is transferred to block "ModMCPCtrlMilling".
- deviceIn: Input image of the Powerride, here "myMcp.In.PowerrideFeed"
- deviceOut: Output image of the Powerride, here "myMcp.In.PowerrideFeed"
- configuration: configFeed area from data block "LBP\_PowerrideConfig"

Block LBP\_Powerride is called a second time for the spindle override. However, the configuration for a spindle override is used here.

## 3 Appendix

### 3.1 Service and Support

#### Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support provides round-the-clock access to our entire Service and Support know-how and service offers.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs and sample applications – all information is accessible with just a few mouse clicks at:

[support.industry.siemens.com](https://support.industry.siemens.com)

#### Technical Support

The Siemens Industry Technical Support offers you fast and competent support for any technical queries you may have with a number of tailor-made solutions

– From basic support up to individual support contracts.

Use the web form to send queries to Technical Support:

[www.siemens.com/industry/supportrequest](https://www.siemens.com/industry/supportrequest)

#### SITRAIN – Training for Industry

With our globally available training courses for Siemens products and solutions, we provide you with practical support, with innovative learning methods and with a customized training concept.

For more information on our various training courses as well as their locations and dates, go to:

[www.siemens.com/sitrain](https://www.siemens.com/sitrain)

#### Service portfolio

Our service portfolio includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofit and modernization services
- Service programs and contracts

You can find detailed information on our service portfolio in the service catalog:

[support.industry.siemens.com/cs/sc](https://support.industry.siemens.com/cs/sc)

#### Industry Online Support app

With the "Siemens Industry Online Support" app, you can obtain optimum support, even when you are on the move. The app is available for iOS and Android:

[support.industry.siemens.com/cs/ww/en/sc/2067](https://support.industry.siemens.com/cs/ww/en/sc/2067)

## 3.2 Links and references

Table 3-1

No.	Topic
\1\	Siemens Industry Online Support <a href="https://support.industry.siemens.com">https://support.industry.siemens.com</a>
\2\	Link to the entry page of the application example <a href="https://support.industry.siemens.com/cs/ww/en/view/109794252">https://support.industry.siemens.com/cs/ww/en/view/109794252</a>
\3\	

## 3.3 Change documentation

Table 3-2

Version	Date	Revision
V1.0	04/2021	First Edition