SIEMENS

Application description • 08/2014

SINAMICS S: Positioning an S120 with S7-1x00 (TIA Portal) via PROFINET with HMI

SINAMICS S120, TIA Portal

http://support.automation.siemens.com/WW/view/en/98961635

Warranty and liability

Note

The application examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The application examples do not represent specific customer solutions; they are intended only as support for typical tasks. You are responsible for the proper operation of the described products. These application examples do not relieve you of the responsibility in safely and professionally using, installing, operating and servicing equipment. By using these application examples, you agree that Siemens cannot be made liable for possible damage beyond the liability clause described. We reserve the right to make changes to these application examples at any time and without prior notice. If there are any differences between the suggestions made in these application examples and other Siemens publications, such as catalogs, the contents of the other document(s) take priority.

We give no guarantee that the information contained in this document is complete, accurate, or up-to-date.

We accept no liability for any damage or loss caused by the examples, information, programs, planning data, or performance data described in this application example, irrespective of the legal basis for claims arising from such damage or loss, unless liability is mandatory. For example, according to the product liability law, in cases of malfeasance, gross negligence, due to endangerment of life, body or health, due to assumption of a guarantee for the properties of a product, due to malicious concealment of a defect or due to violation of basic contractual obligations. Any compensation for violation of basic contractual obligations, however, shall be limited to the foreseeable damage or loss which is typically envisaged in contracts unless there has been gross negligence or unless liability is mandatory due to endangerment of life, body, or health. Any change to the burden of proof to your disadvantage is not covered hereby.

Any form of duplication of these application examples or excerpts hereof is not permitted without the express consent of Siemens Industry Sector.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, devices, and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Any third-party products that may be in use must also be taken into account. For more information about industrial security,
visit http://www.siemens.com/industrialsecurity

To receive information about product updates on a regular basis, register for our product newsletter. For more information,
visit http://support.automation.siemens.com.

Table of contents

1 Task

Overview of the automation task

The following diagram provides an overview of the automation task.

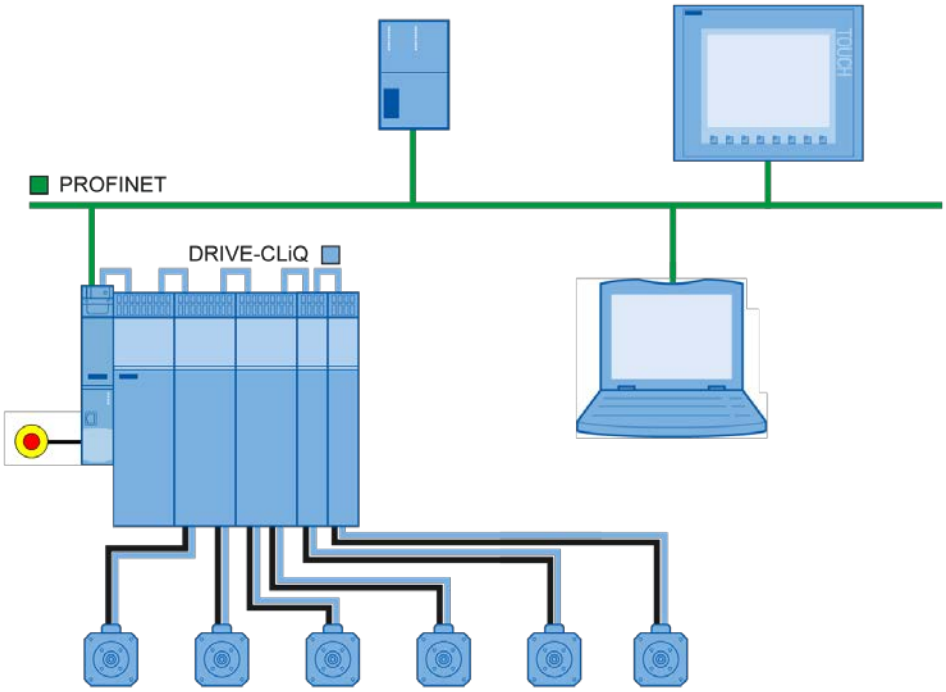


Fig. 1-1

Description of the automation task

The objective of the application is to show conceivable applications involving the SINA_POS(FB284) and SINA_PARA(FB286) function blocks to integrate basic positioner technology (EPOS). This is realized in conjunction with an S7-1x00 in the TIA Portal environment. An appropriately preconfigured HMI is used to operate the application.

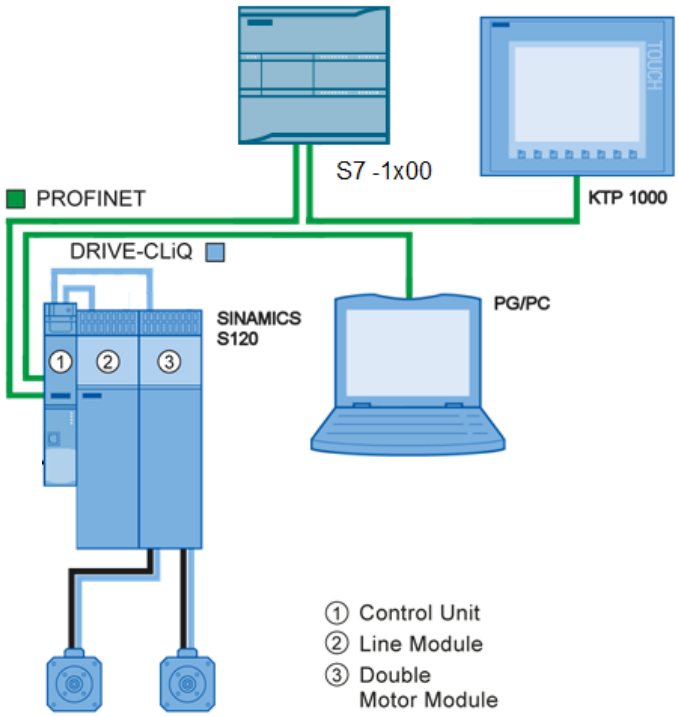
2 Solution

2.1 Overview

Schematic

The following schematic diagram shows the most important components of the solution:

Fig. 2-1



Advantages

The application described here offers you the following advantages:

- Sequential control of up to 32(*1)(*2) EPOS axes
- n-1 (n=number of axes) of the necessary instance data blocks are not required
- Non-cyclic reading/writing of up to 16 arbitrary parameters
- The new blocks FB284 / FB286 can be displayed on the HMI
- HMI demonstration of an absolute encoder adjustment

(1) The limit to 32 can, when required, be adapted in the function block of the application

(2) On the hardware side, the number of EPOS axes is also dependent on the S7-CPU used regarding available I/O area and number of connectable I/O devices (e.g. drives,, ET200, …).

Demarcation

This application does not contain a description of:

- Basic drive commissioning
- EPOS configuration

It is assumed that readers have basic knowledge of these topics.

Knowledge required

It is assumed that readers have basic knowledge of STARTER / Step7 V12/13.

2.2 Description of the core functionality

The new function blocks SINA_POS(FB284) and SINA_PARA(FB286) form the basis of the application.

The application allows the link, provided by block SINA_POS, to be actively controlled from the HMI, and any number of EPOS axes to be sequentially controlled.

The application also allows the acyclic communication to be graphically operated using SINA_PARA.

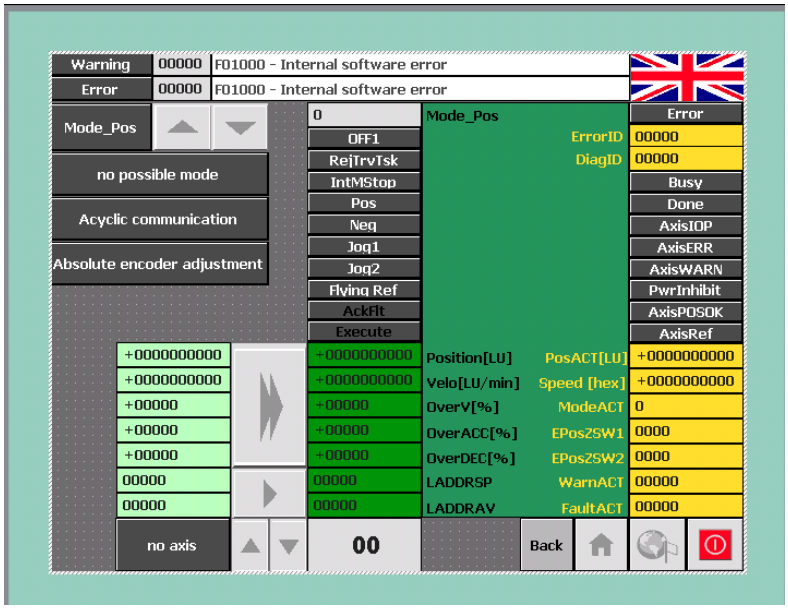
If the HMI access is deactivated, and instead, a PLC program is used, then it is possible to further develop the functionality of the application within the context of your own/dedicated application.

Further, it is possible to expand the number of axes operating in parallel. This is done by using the function block as well as the application data block x number of times.

Note If it is not possible to sequentially control EPOS axes in the application, then the framework, comprising FB and application DB can also be omitted, and communication directly established using FB284.

Overview and description of the user interface

Fig. 2-2 – Cyclic communication with FB284



Sequence of the core functionality

Fig. 2-3

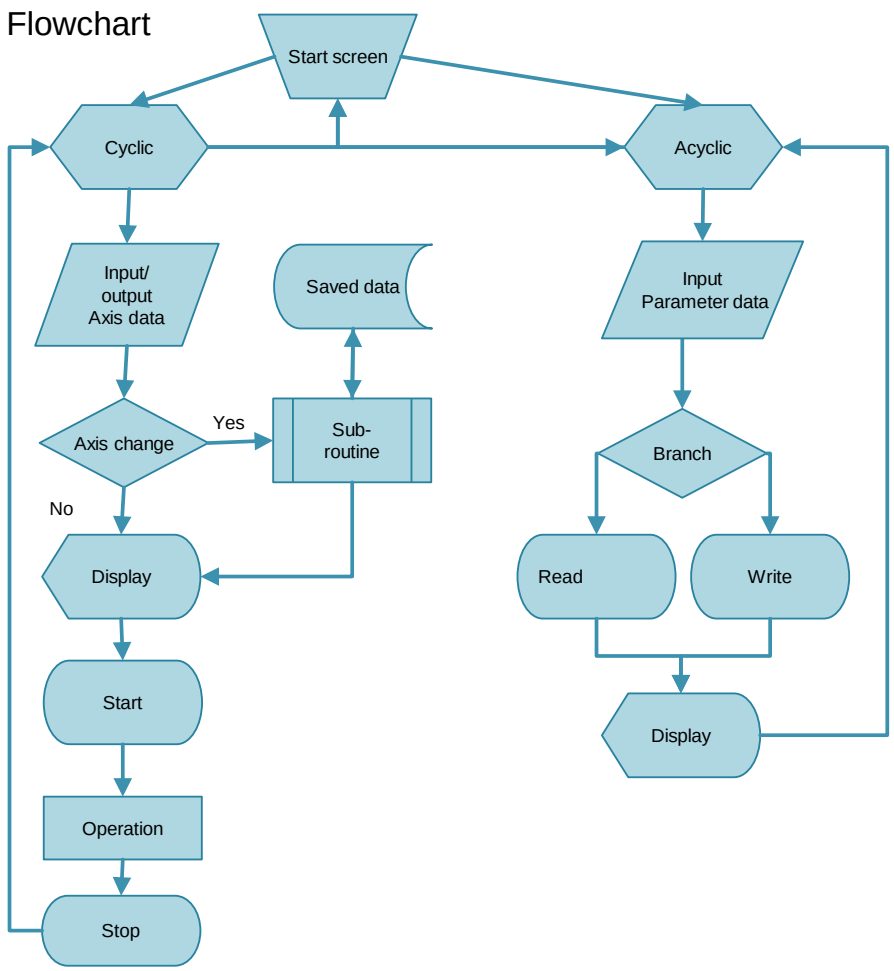


Table 2-1

	Action	Note
1.	HMI starts with the start screen	
2.	Either cyclic or acyclic communication is selected	It is possible to toggle between acyclic communication and cyclic communication at any time by pressing a button
3.	Cyclic: Make a setting for the axes and traverse the axis Acyclic: Any arbitrary parameter can be written to or read	The axis data are not retentively saved when using the "Default setting"

2.3 Hardware and software components

2.3.1 Validity

This application is valid for

- STEP 7 from V12SP1
- S7-1200 from FW 2.x
- S7-1500 from FW 1.1

2.3.2 Components used

The application was created with the following components:

Hardware components

Table 2-2

Component	Qty.	Order number	Note
• S7-1214 DC/DC/DC	1	6ES7214-1AG40-0XB0	V2.2
• S7-1516 3PN/DP	1	6ES7516-3AN00-0AB0	V1.5
• SINAMICS S120 V4.6	1	6SL3040-1LA01-0AA0	CU310-2 PN
• KTP1000 basic color	1	6AV6647-0AF11-3AX0	

Software components

Table 2-3

Component	Qty.	Order number	Note
• Step7-V12SP1	1	6ES7822-1AA02-0YA5	
• WINCC V12SP1	1	6AV2102-0AA02-0AA5	
• STARTER 4.3.3.0	1		http://support.automation.siemens.com/WW/view/en/26233208

Size of the blocks used – example of a S7-1214 DC/DC/DC V2.2

Table 2-4

Block	Size in the work memory	Size in the load memory	Note
FBHmiSinaPos	5789B	37853B	
Inst_DB_FBHmiSinaPos	788B	5855B	
Sina_POS	7157B	84200B	
Sina_POS_DB	118B	11938B	
Sina_PARA	20799B	133977B	
Sina_PARA_DB	674B	8259B	
DBApplicationData	92B	3356B	

Sample files and projects

The list below contains all the files and projects used in this example.

Table 2-5

Component	Note
98961635_S120_at_S7-1500_EPOS_v10.zip	This zipped file contains a STEP 7 V12 SP1 with connection of the SINAMICS S120 via PROFINET.
98961635_S120_at_S7-1200_EPOS_v10.zip	This zipped file contains a STEP 7 V12 SP1 with connection of the SINAMICS S120 via PROFINET.
98961635_S120_at_S7-1x00_EPOS_DOKU_v10.pdf	This document

3 Basic principles

3.1 Cyclic communication

The process data is transferred cyclically, i.e. in each bus cycle. Depending on the bus system used, isochronous or non-isochronous data transfer is possible. In principle, the cyclic communication is a time-critical application.

The SIMATIC S7 controller sends control words and setpoints to the SINAMICS drive and receives status words and actual values from the SINAMICS drive.

With regard to use in the SINAMICS drive, the telegram structure is set by means of predefined standard telegrams according to PROFIdrive profile or manufacturer-specific telegrams.

Depending on the telegram type, a different number of setpoints or actual values or extended control or status words are transferred. The telegram length as well as the links in the SINAMICS drive are fixed in during operation and cannot be changed.

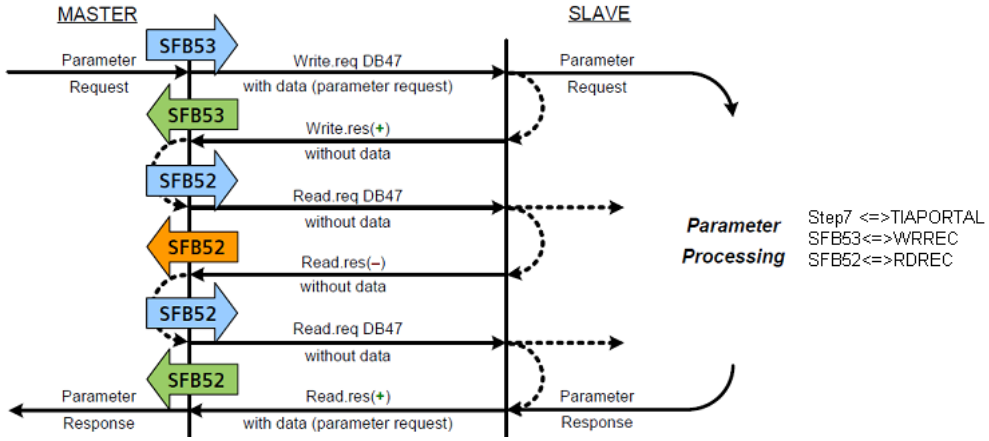
- On the SIMATIC S7 controller side, the process data is provided as peripheral input or output words.
- Which control word bits and which data should be sent to the SIMATIC S7 controller is defined in the SINAMICS drive by the parameterization.
- Various standard functions and function blocks are available for the data exchange in the SIMATIC controllers.

Note

A detailed description of cyclic communication can be found in the **Function Manual, (FH1), 01/2012, 6SL3097-4AB00-0AP2** of the SINAMICS S120 in Chapter 10. (/3/)

3.2 Acyclic communication – data block 47

Fig. 3-1



It is possible to acyclically transfer the parameter area when required, without creating a permanent communication load (communication overhead). The acyclic transfer takes significantly longer than the cyclic transfer of the processed data, however, larger data quantities can be transferred.

- In the SIMATIC controller, read and write jobs are initiated via the standard function blocks SFB52/53.
- A read job always starts with a write job which informs the addressed node which values are to be determined. The actual read job is then performed.
- No special action is required on the SINAMICS drive side.

Decisive for a functioning acyclic communication is the creation of a job profile corresponding to the data block used.

The response to write and read jobs must also be transferred in appropriate data block structures and evaluated.

With unchanging write and read jobs, the structure can be defined beforehand. However, if the jobs vary and the contents are different, this can only be mapped in a general structure and must be evaluated separately by the user.

Note

A detailed description of cyclic communication can be found in the Function Manual, (FH1), 01/2012, 6SL3097-4AB00-0AP2 of the SINAMICS S120 in Chapter 10. (/3/)

Further information with regard to data block 47 can be found in the PROFIdrive Manual, Edition 2006.

3.3 Fundamentals of the basic positioner (EPOS)

The basic positioner (EPOS) is a very comprehensive and powerful function module for position-controlled traversing of an electric drive.

It is used for absolute and relative positioning of linear and rotary axes (modulo) with motor encoders (indirect measuring system) or machine encoders (direct measuring system).

It can be activated in various drives of the SINAMICS S/G converter series as a function module.

User-friendly configuration, commissioning and diagnostic functions for the EPOS functionality are also available in the STARTER or Startdrive parameterization software.

The position controller is also activated when activating the basic positioner. This is performed automatically via the drive wizard. Further, the necessary "internal interconnections" (BICO technology) are automatically established, which are required between the EPOS and position controller (e.g. setpoints from the EPOS for closed-loop position control, axis cycle correction, etc.).

The closed-loop position control essentially comprises the following parts:

- Actual position value processing (including the lower-level measuring input evaluation and reference mark search)
- Position controller (including limits, adaptation and precontrol calculation)
- Monitoring functions (standstill, positioning and dynamic following error monitoring, cam signals)

In addition, the following functions can be carried out using the basic positioner:

Mechanical system:

- Backlash compensation
- Modulo correction
- Position tracking / limits
- Velocity/acceleration/deceleration limits
- Software limit switches (traversing range limitation using position setpoint evaluation)
- Stop cams (traversing range limitation using hardware limit switch evaluation)
- Positioning/standstill monitoring
- Following error monitoring
- Two cam switching signals

Note Detailed descriptions can be found in the Basic Positioner Function Manual, 01/2013, FW V4.6, A5E31759509A AA

3.4 Fundamentals of SINA_POS and SINA_PARA

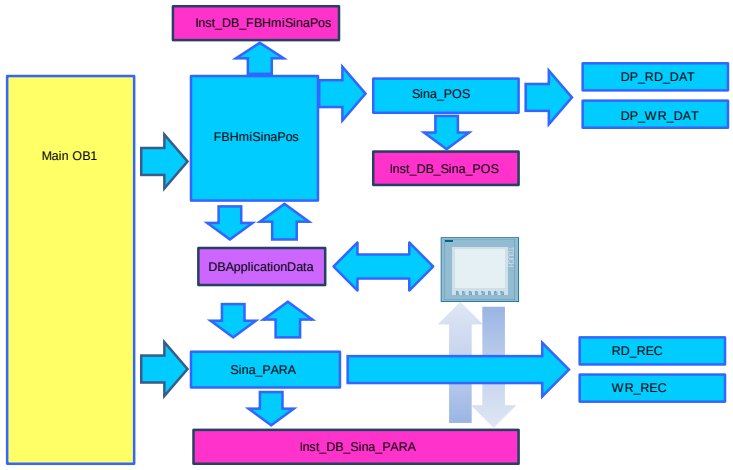
Detailed documentation is available regarding using/operating SINA_POS and SINA_PARA (SINAMICS_blocks_TIAP_V12SP1_V13.pdf) at the following link in the Siemens Industry Online Support:

http://support.automation.siemens.com/WW/view/en/68034568

4 Principle of operation

4.1 General overview

Fig. 4-1



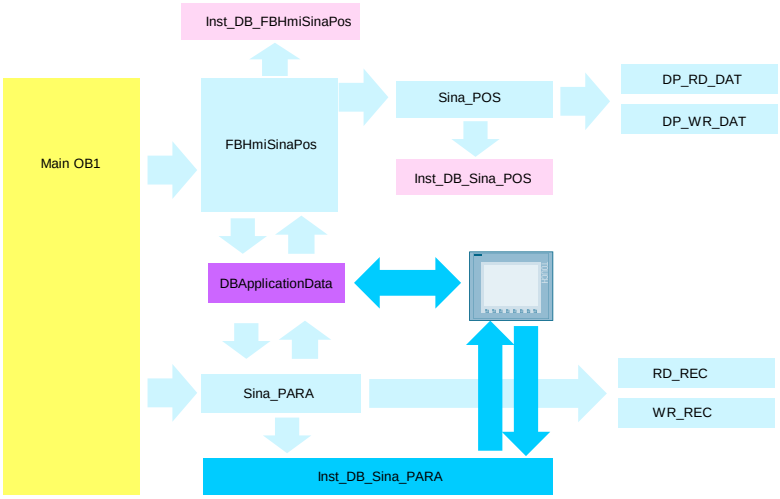
Application function block FBHmiSinaPos is called in a cyclic OB – e.g. the OB1.

The data that the user has entered at the HMI – as well as the display of the various axis states – are collected via the DBApplicationData application data block, and exchanged with the application FB. The HMI provided graphically forms the interface of both blocks.

Using the integrated function blocks SINA_POS (FB284) and SINA_PARA (FB286), cyclic and acyclic data are exchanged with the SINAMICS drive.

4.2 Functionality of the HMI connection

Fig. 4-2

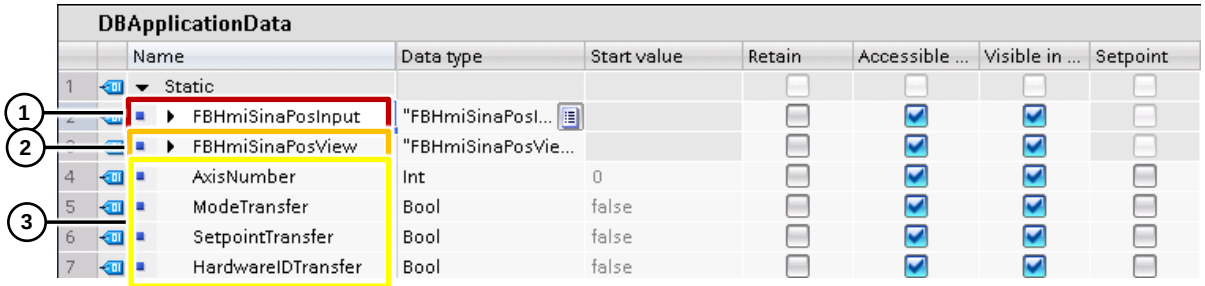


4.2.1 Program details for SIMATIC PLC <-> HMI data exchange

- Data is exchanged between the application and the HMI using a user-defined data block. This block – "DBApplicationData" – has the following functions:

1. The user inputs from the HMI regarding SINA_POS and operator inputs SINA_PARA,
2. Displays the actual values as well as the actual axis control signals on the HMI,
3. And handles all of the additional control commands

Fig. 4-3 Complete overview of DBApplicationData



- On the other hand, the parameter data for the cyclic communication are directly transferred to the structure of the instance data block of SINA_PARA.

Fig. 4-4

HMI tags

Name	Tag table	Data type	Connection	PLC name	PLC tag	Address	Access mode
SINA_PARA_DB_sxParameter[1].siIndex	Default tag table	Int	HMI_Verbindung_2	PLC_2	SINA_PARA_DB.sxParameter[1].siIndex		<symbolic access>
SINA_PARA_DB_sxParameter[1].siParaNo	Default tag table	Int	HMI_Verbindung_2	PLC_2	SINA_PARA_DB.sxParameter[1].siParaNo		<symbolic access>
SINA_PARA_DB_sxParameter[1].srValue	Default tag table	Real	HMI_Verbindung_2	PLC_2	SINA_PARA_DB.sxParameter[1].srValue		<symbolic access>
SINA_PARA_DB_sxParameter[1].swErrorNo	Default tag table	Word	HMI_Verbindung_2	PLC_2	SINA_PARA_DB.sxParameter[1].swErrorNo		<symbolic access>
SINA_PARA_DB_sxParameter[1].syFormat	Default tag table	Byte	HMI_Verbindung_2	PLC_2	SINA_PARA_DB.sxParameter[1].syFormat		<symbolic access>

3 user-defined data types exist to create the data block:

- FBHmiSinaPosInputType

Table 4-1

Signal	Type	Default	Input/output
SINA_POS			
ModEPOS	Int	0	I
Off1	Bool	false	I
Rejecttraversingtask	Bool	false	I
Intermediatestop	Bool	false	I
Positive	Bool	false	I
Negative	Bool	false	I
Jog1	Bool	false	I
Jog2	Bool	false	I
Flyingreference	Bool	false	I
Acknowledgefault	Bool	false	I
Execute	Bool	false	I
MDIposition	DInt	0	I
MDIvelocity	DInt	0	I
Velocityoverride	Int	0	I
Acceleration	Int	0	I
Deceleration	Int	0	I
LaddrSP	Word	16#0	I
LaddrAV	Word	16#0	I
SINA_PARA			
Sinaparastart	Bool	false	I
Sinaparareadwrite	Bool	false	I
Sinaparanumberparameter	Int	0	I
SinaParaLaddr	Word	16#0	I
Sinaparaaxisnumber	Bytes	16#0	I
Sinaparaerror	Bool	false	O
SinaparaerrorID	DWord	16#0	O
Sinaparabusy	Bool	false	O
Sinaparadone	Bool	false	O
SinaparadiagID	Word	16#0	O

- FBHmiSinaPosViewType

Table 4-2

Signal	Type	Default
SINA_POS		
FBHmiSinaPosSetpointsAct	FBHmiSinaPosSetpointType	
ModEPOSAct	INT	o
ErrorAct	Bool	false
ErrorIDAct	Word	16#0
DiagIDAct	Word	16#0
BusyAct	Bool	false
DoneAct	Bool	false
AxisInOperationAct	Bool	false
AxisErrorAct	Bool	false
AxisWarningAct	Bool	false
AxisPositionOkAct	Bool	false
AxisReferencedAct	Bool	false
VelocityAct	DInt	0
PositionAct	DInt	0
PowerInhibitAct	Bool	false
EPOSStatusWord1Act	Word	16#0
EPOSStatuswordAct	Word	16#0
WarningAct	Int	0
FaultAct	Int	0

- FBHmiSinaPosSetpointType

Table 4-3

OFF1Act	Bool	false
RejectTraversingTaskAct	Bool	false
IntermediateStopAct	Bool	false
PositiveAct	Bool	false
NegativeAct	Bool	false
Jog1Act	Bool	false
Jog2Act	Bool	false
FlyingReferenceAct	Bool	false
AcknowledgeFaultAct	Bool	false
ExecuteAct	Bool	false
MdiPositionAct	DInt	0
MdiVelocityAct	DInt	0
VelocityOverrideAct	Int	0
AccelerationAct	Int	0
DecelerationAct	Int	0
LaddrSPAct	WORD	0
LaddrAVAct	WORD	0

As a result of the data management required for the various EPOS axes in the FBHmiSinaPos function block, user input data and actual values must be exchanged via the application data block. This facilitates a clear demarcation between input data and display data on the HMI side; however, the most important reason is the axis change.

4.2.2 Configuring information

In the HMI example that has been prepared, the HMI variables are already directly linked with the application data block.

To understand the significance of the HMI variables, when creating the HMI variables, the same names as the SIMATIC PLC variables were used:

Fig. 4-5

HMI tags

Name	Tag table	Data type	Connection	PLC name	PLC tag
ABS_JUSTAGE_STEPS	Default tag table	Int	<Internal tag>		<Undefined>
DBApplicationData_FBHmiSinaPosView.AxisErrorAct	Default tag table	Bool	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.AxisErrorAct
DBApplicationData_FBHmiSinaPosView.AxisInOperationAct	Default tag table	Bool	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.AxisInOperationAct
DBApplicationData_FBHmiSinaPosView.AxisPositionOkAct	Default tag table	Bool	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.AxisPositionOkAct
DBApplicationData_FBHmiSinaPosView.AxisReferencedAct	Default tag table	Bool	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.AxisReferencedAct
DBApplicationData_FBHmiSinaPosView.AxisWarningAct	Default tag table	Bool	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.AxisWarningAct
DBApplicationData_FBHmiSinaPosView.BusyAct	Default tag table	Bool	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.BusyAct
DBApplicationData_FBHmiSinaPosView.DiagIDAct	Default tag table	Word	HMI_Verbindung_2	PLC_2	DBApplicationData.FBHmiSinaPosView.DiagIDAct

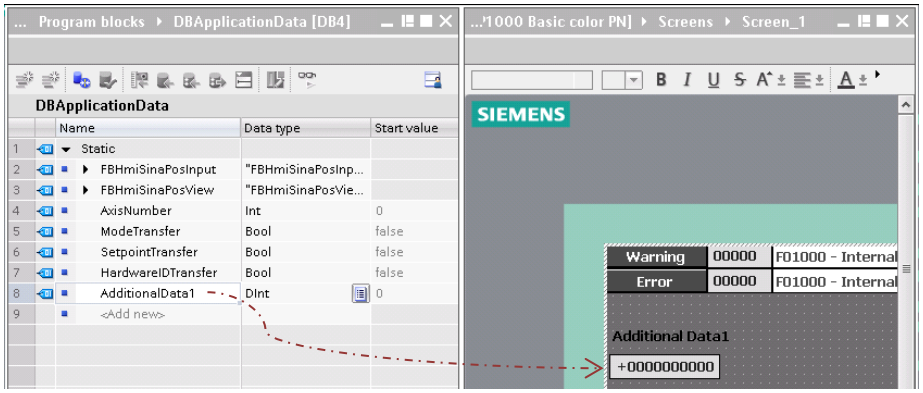
The SIMATIC PLC data types as well as the application data block can be expanded at any time to expand the connection with the appropriate values. Linking by simply dragging & dropping is possible.

Fig. 4-6

DBApplicationData

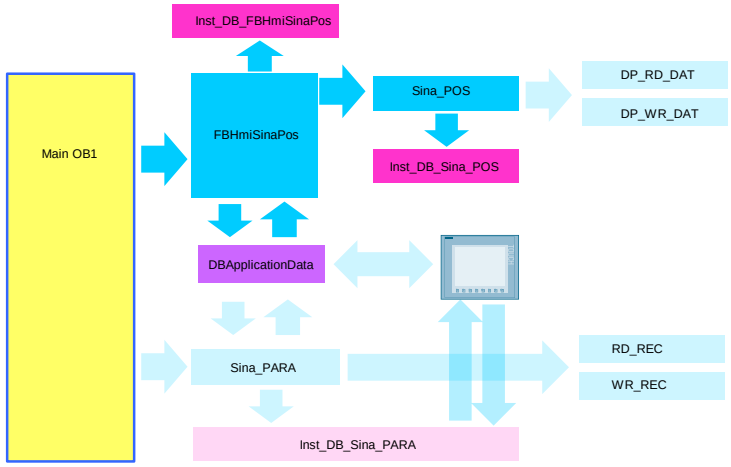
	Name	Data type	Start value	Retain	Accessible ...	Visible in ...	Setpoint
1	Static						
2	FBHmiSinaPosInput	"FBHmiSinaPosInp...					
3	FBHmiSinaPosView	"FBHmiSinaPosVie...					
4	AxisNumber	Int	0				
5	ModeTransfer	Bool	false				
6	SetpointTransfer	Bool	false				
7	HardwareIDTransfer	Bool	false				
8	AdditionalData1	Real	0.0				
9	<Add new>						

Fig. 4-7



4.3 Functionality of the FBHmiSinaPos block

Fig. 4-8



Program details about block FBHmiSinaPos

Application block FBHmiSinaPos internally processes the user inputs acquired via data block DBApplicationData – and provides the appropriate status signals for the cyclic communication as well as the status of the particular axis.

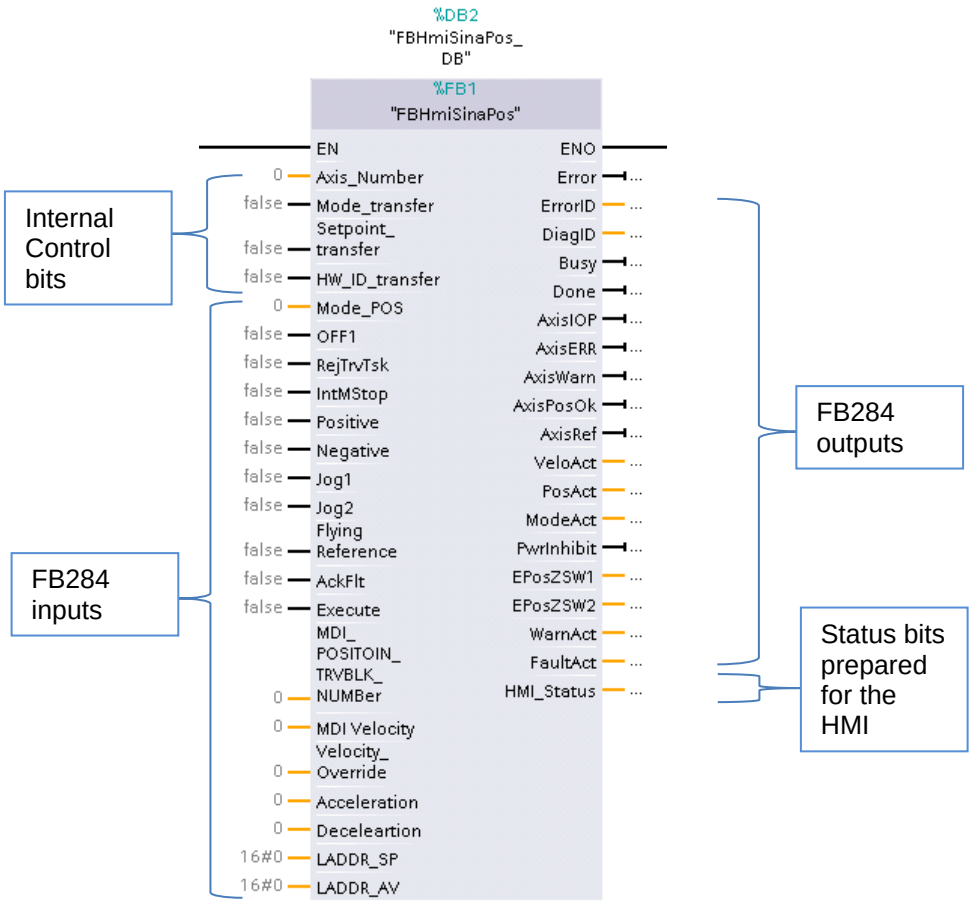
The principle structure of block FBHmiSinaPos is similar to that of block SINA_POS regarding inputs and outputs. This is necessary, as FB284 is internally called in FBHmiSinaPos, and therefore the setpoints and/or actual values must be made available via the block interface.

The additional inputs and outputs available are used to control the internal logic of the axis change and/or their display at the HMI.

Note

If it is not necessary to have an axis change function, internal control bits can be preassigned with "1" or "True". The block then behaves in precisely the same fashion as the internally called FB284.

Fig. 4-9



The axis change functionality is realized as follows (and is controlled in the application example from the HMI):

4. Configuration (several axes)

   a. Setting axis number "1" (or "x") and pre-assignment of the setpoints, the ID addresses and the axis operating mode (ModEPOS)
   b. HW ID / setpoints / ModEPOS accepted with a signal edge(*1) at the currently selected axis number (in case 1)
   c. Continue with step 2 or configure another axis
   d. Setting axis number "2" and pre-assignment of the setpoints as well as the hardware ID and axis operating mode
   e. HW ID / setpoints / ModEPOS accepted with a signal edge(*1) at the currently selected axis number (in case 2)

(*1) *When true is permanently 1 for these inputs, each setpoint change is accepted in the block for the actual data set, however, only when FB284 is activated – functionality also implemented in the SINAMICS drive*

5. Selecting axes based on the axis number (1 or 2 or …)

a. The actual status of the axis is in the outputs – AS WELL AS outputs HMI_status, which contain the status of the input bit.
b. When the control bit is changed, this results in a corresponding axis response (e.g. start of traversing motion)
c. When changing the ModEPOS / the setpoints, these must be accepted using a signal edge(*1)
d. If a new axis change is made, in the FBHmiSinaPos, FB284 is switched over to the "new" hardware ID and the last saved setpoints / ModEPOS of the axis, and at the same time, control and status bits are updated and transferred (to the SINAMICS drive / HMI)

Table 4-4 Overview of the input data FBHmiSinaPos

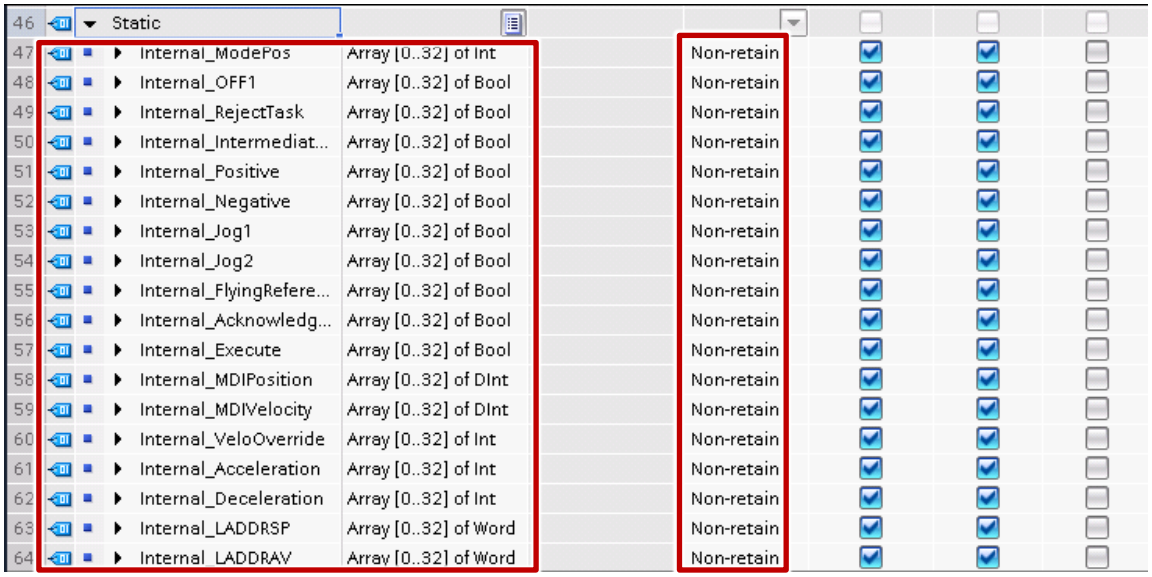
Input signal	Type	Default[…]	Meaning
Axis_number	INT	0	Selecting the axis/axis data
Mode_Transfer	BOOL	0	0 = ModEPOS is not accepted; 1 = ModEPOS is accepted and is activated
Setpoint_Transfer	BOOL	0	0 = setpoint not accepted; 1= accepted setpoint is activated
HW_ID_Transfer	BOOL	0	0=hardware ID is not accepted; 1 = hardware ID is accepted in the axis data set
ModEPOS	INT	0	Mode: 1 = relative positioning 2 = absolute positioning 3 = positioning as setup 4 = reference point approach 5 = set reference point 6 = traversing block 0 – 15/63 (G120/S120) 7 = jog mode 8 = incremental jogging
Off1	BOOL	0	Switching command: 0 = OFF1, 1 = ON
RejTrvTsk	BOOL	1	0 = reject active traversing task, 1 = do not reject
IntMStop	BOOL	1	Intermediate STOP, 0 = active motion command is interrupted, 1 = no intermediate stop
Pos	BOOL	0	Positive direction
Neg	BOOL	0	Negative direction
Jog1	BOOL	0	Jog signal source 1
Jog2	BOOL	0	Jog signal source 2
FlyRef	BOOL	0	0 = deselect flying referencing, 1 = select flying referencing
AckFlt	BOOL	0	Acknowledge faults
Execute	BOOL	0	Activate traversing task / setpoint acceptance / activate reference function
Position	DINT	0[LU]	Position setpoint in [LU] for direct setpoint specification / MDI mode OR traversing block number for traversing block mode
Velocity	DINT	0[LU/min]	Velocity in [LU/min] for MDI mode
OverV	INT	100[%]	Velocity override active for all modes: 0-199%
OverAcc	INT	100[%]	Acceleration override active 0-100%
OverDec	INT	100[%]	Deceleration override active 0-100%
LAddr**SP**	HW_IO INT	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the setpoint slot (**S**et**P**oint)
LAddr**AV**	HW_IO INT	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the actual value slot (**A**ctual **V**alue)

Table 4-5 FBHmiSinaPos output signals

Output signal	Type	Default[…]	Meaning
Error	BOOL	0	1 = general fault active
ErrorId	INT	0	Mode fault / block fault: 0 = no fault active 1 = communication fault active 2 = incorrect mode selected 3 = incorrect parameterization of the setpoints 4 = invalid traversing block number 5 = drive fault active 6 = Closing lockout active 7 = flying referencing could not be started
Busy	BOOL	0	Mode is being executed or enabled
Done	BOOL	0	Mode has been executed error-free
PwrInhibit	BOOL	0	Switching on inhibited active → PwrInhibit =1
AxisIOp	BOOL	0	Drive is ready and switched on
AxisErr	BOOL	0	Drive is faulted
AxisWarn	BOOL	0	Drive alarm active
AxisPosOk	BOOL	0	Target position of the axis reached
AxisRef	BOOL	0	Reference point set
PwrInhibit	BOOL	0	Switching on inhibited
VeloAct	DINT	0[LU/min]	Actual velocity in [LU/min]
PosAct	DINT	0[LU]	Current position in LU
ModeAct	INT	0	Currently active mode
EPOSZSW1	WORD	0	Status of the EPOS ZSW1 (bit-granular)
EPOSZSW2	WORD	0	Status of the EPOS ZSW2 (bit-granular)
WarnAct	WORD	0	Actual warning number
FaultAct	WORD	0	Actual fault number
DiagId	WORD	0	Extended communication error → error during SFB call
HMI_Status	FBHmiSinaPos SetpointType	[]	Status of the input bit after an axis change

For the internal data management of the various axes, a special data area was created in FBHmiSinaPos. Using various arrays, this buffers the statuses/contents of the setpoints/control signals of the particular axis. As a result of the selected size, only up to 33 different axes are possible (only 32 axes when an HMI is being used → Arry[0] per default cannot be defined as axis 0 in the HMI)

Fig. 4-10

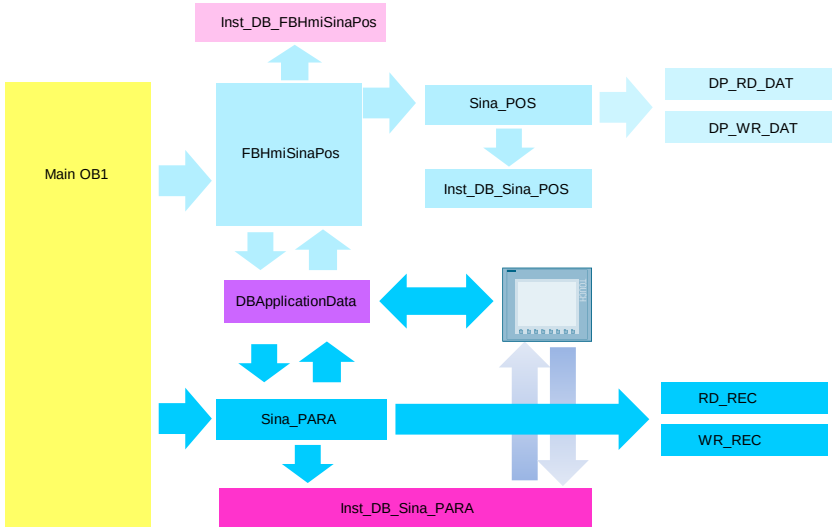


The output data of the actual axis are not the buffered; the reason for this is that when the axis is switched over, a switchover is made to a "new" valid hardware ID, and its status data can be output at the FB284 or FBHmiSinaPos.

NOTICE	**Axis data are lost after a power off/on**
	Regarding the setpoint direction/control bits/hardware IDs, in the current status of the application, axis data are only "Non-retain" – i.e. they are lost for power off/on. When saved in a non-volatile fashion, the required variables can be changed over to "retain" and are saved in a non-volatile fashion.

4.4 Functionality HMI <----> SINA_PARA (FB286)

Fig. 4-11



Program details for SINA_PARA

Communication between the HMI and SINA_PARA (FB286) is realized via 2 different channels.

1. On one hand, the HMI exchanges the necessary control commands with the application data block to correspondingly start acyclic jobs.
2. On the other hand, the HMI directly accesses the parameter structure of the instance data block of SINA_PARA to transfer the data of write jobs – or order to display the data of read jobs on the HMI. This is realized using HMI variables predefined in the application example.

Fig. 4-12 Control commands for SINA_PARA

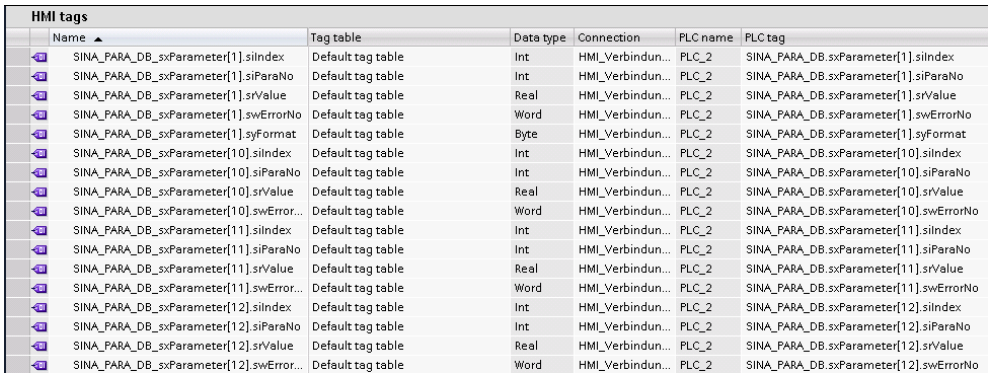
DBApplicationData

	Name	Data type	Start value	Retain	Accessible ...	Visible in ...	Setpoint
21	Sinaparastart	Bool	false	☐	☑	☑	☐
22	Sinaparareadwrite	Bool	false	☐	☑	☑	☐
23	Sinaparanumberp...	Int	0	☐	☑	☑	☐
24	SinaParaLaddr	Word	16#0	☐	☑	☑	☐
25	Sinaparaaxisnum...	Byte	16#0	☐	☑	☑	☐
26	Sinaparaerror	Bool	false	☐	☑	☑	☐
27	SinaparaerrorID	DWord	16#0	☐	☑	☑	☐
28	Sinaparabusy	Bool	false	☐	☑	☑	☐
29	Sinaparadone	Bool	false	☐	☑	☑	☐
30	SinaparadiagID	Word	16#0	☐	☑	☑	☐
31	SinaparaDOnumb...	Byte	16#0	☐	☑	☑	☐

Table 4-6

Signal	Type	Default	Meaning
Sinaparastart	BOOL	0	Start of the job
Sinaparareadwrite	BOOL	0	0=read, 1=write
Sinaparanumberparameter	INT	0	Number of parameters → 1 to 16
Sinaparaaxisnumber	INT	0	Axis number / axis ID for multi-axis system
SinaparaLaddr	HW IO INT	0	Hardware ID of the actual value telegram slot
Sinaparaerror	BOOL	0	Group fault active → "Error" =1
SinaparaerrorId	DWORD	0	1st word → which parameter access is faulted in binary code 2nd word: Fault type
Sinaparabusy	BOOL	0	Job being processed with "Busy"=1
Sinaparadone	BOOL	0	Job completed without error means edge change from 0→1
SinaparadiagId	WORD	0	Extended communication error → error during SFB call

Fig. 4-13 Juxtaposition of HMI variables and PLC variables (IDB FB286)



HMI tags

Name	Tag table	Data type	Connection	PLC name	PLC tag
SINA_PARA_DB_sxParameter[1].siIndex	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[1].siIndex
SINA_PARA_DB_sxParameter[1].siParaNo	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[1].siParaNo
SINA_PARA_DB_sxParameter[1].srValue	Default tag table	Real	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[1].srValue
SINA_PARA_DB_sxParameter[1].swErrorNo	Default tag table	Word	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[1].swErrorNo
SINA_PARA_DB_sxParameter[1].syFormat	Default tag table	Byte	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[1].syFormat
SINA_PARA_DB_sxParameter[10].siIndex	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[10].siIndex
SINA_PARA_DB_sxParameter[10].siParaNo	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[10].siParaNo
SINA_PARA_DB_sxParameter[10].srValue	Default tag table	Real	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[10].srValue
SINA_PARA_DB_sxParameter[10].swError...	Default tag table	Word	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[10].swErrorNo
SINA_PARA_DB_sxParameter[11].siIndex	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[11].siIndex
SINA_PARA_DB_sxParameter[11].siParaNo	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[11].siParaNo
SINA_PARA_DB_sxParameter[11].srValue	Default tag table	Real	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[11].srValue
SINA_PARA_DB_sxParameter[11].swError...	Default tag table	Word	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[11].swErrorNo
SINA_PARA_DB_sxParameter[12].siIndex	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[12].siIndex
SINA_PARA_DB_sxParameter[12].siParaNo	Default tag table	Int	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[12].siParaNo
SINA_PARA_DB_sxParameter[12].srValue	Default tag table	Real	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[12].srValue
SINA_PARA_DB_sxParameter[12].swError...	Default tag table	Word	HMI_Verbindun...	PLC_2	SINA_PARA_DB.sxParameter[12].swErrorNo

Note In the application example, the instance data block of SINA_PARA is accessed via HMI variables with a one second refresh time.

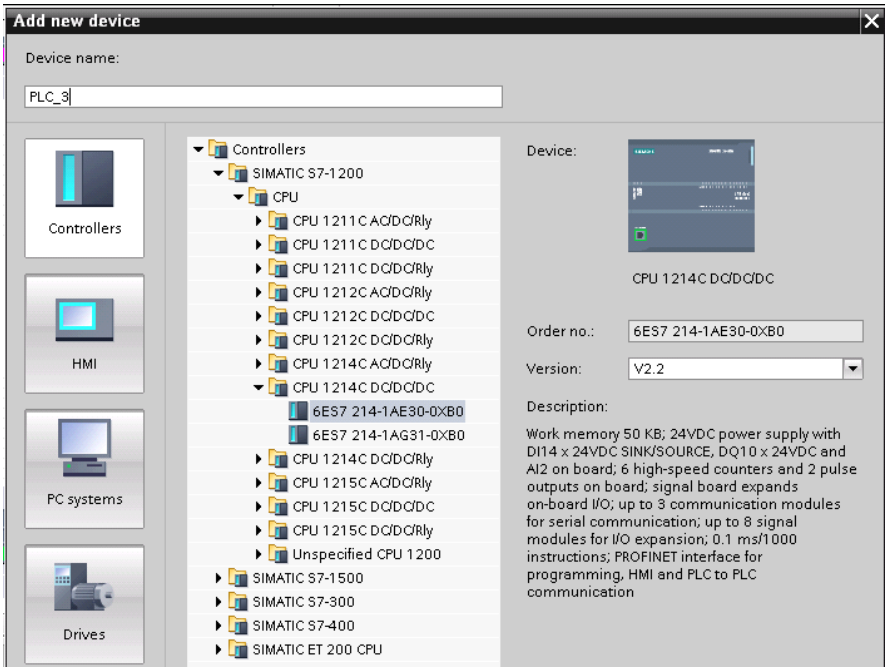
5 Configuration and project engineering

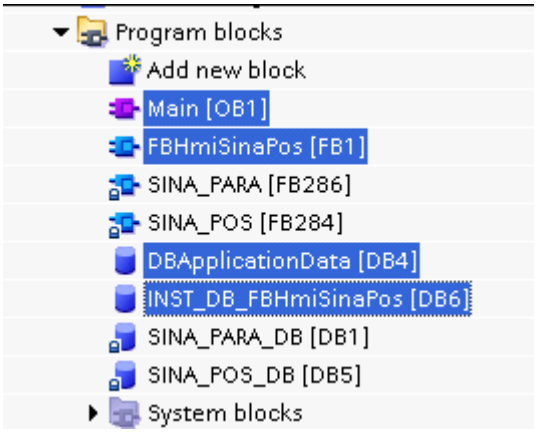
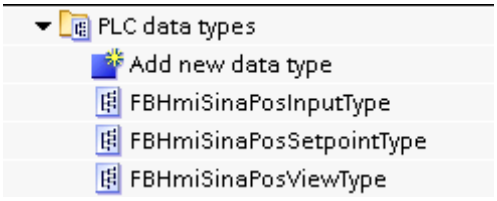
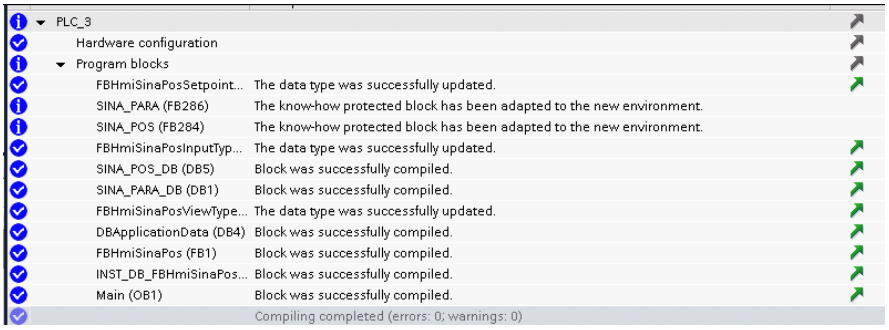
With the prepared hardware configuration (S7-1200 / S7-1500 with SINAMICS S120), the application example can completely function. Further, the supplied HMI can be simulated using the existing WINCC runtime license.

The following steps are required to reparameterize the application:

5.1 Configuring the SIMATIC S7 controller

Table 5-1

No.	Action
1.	Create a S7-1x00 controller 
2.	**Insert blocks SINA_POS(FB284) and SINA_PARA(FB286) from the corresponding library S7-1200 or S7-1500**

No.	Action
3.	Insert the application blocks from the project example: MAIN(OB1), FBHmiSinaPos(FB1), DBApplicationData(DB4), INST_DB_FBHmiSinaPos(DB6) 
4.	Insert the user-defined data types used in the application example: FBHmiSinaPosInputType, FBHmiSinaPosSetpointType, FBHmiSinaPosViewType 
5.	Compile the configuration 

5.2 Configuring the communication

Table 5-2 Integrating the drive

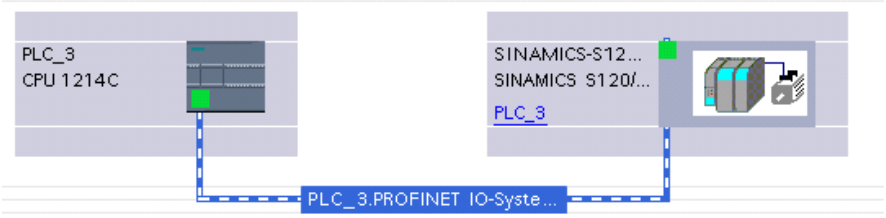
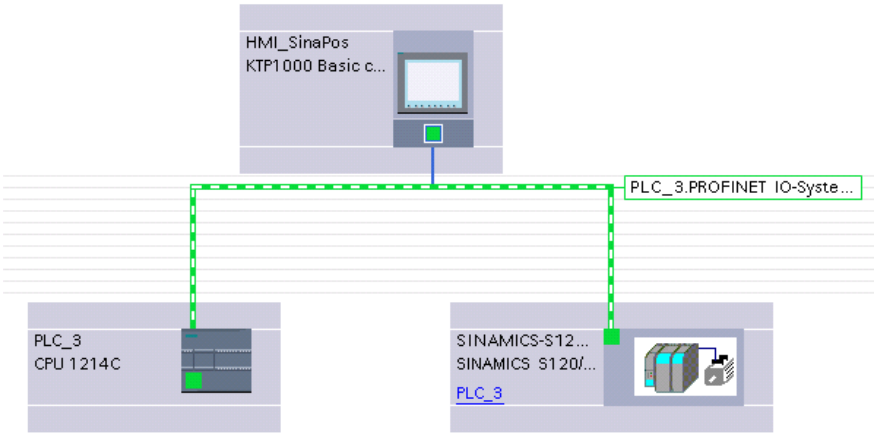
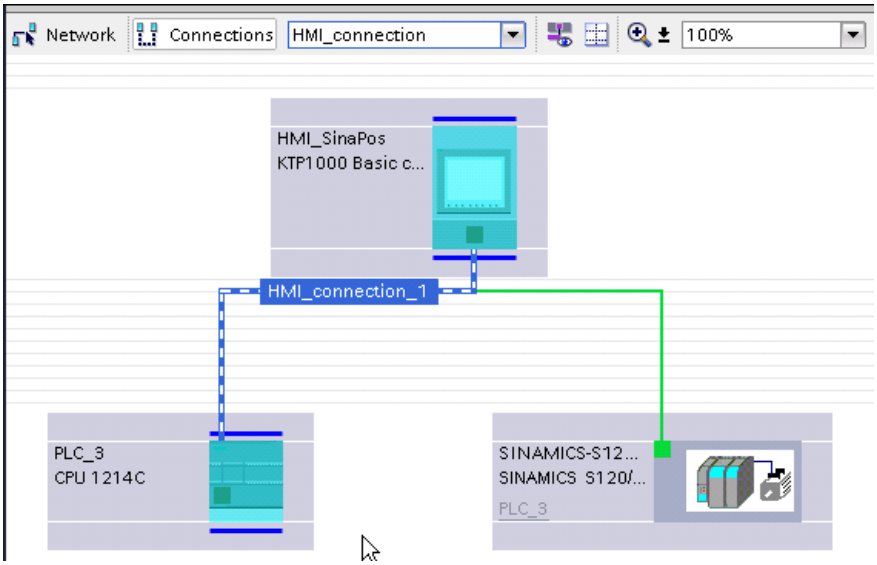
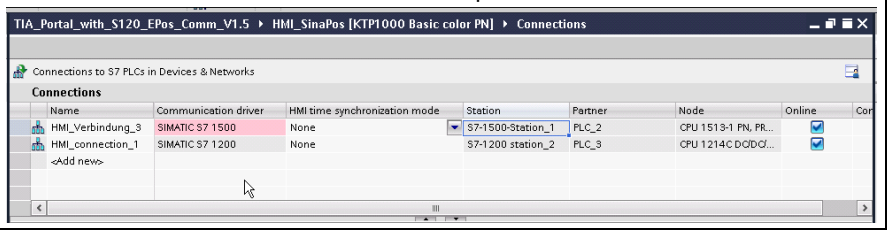
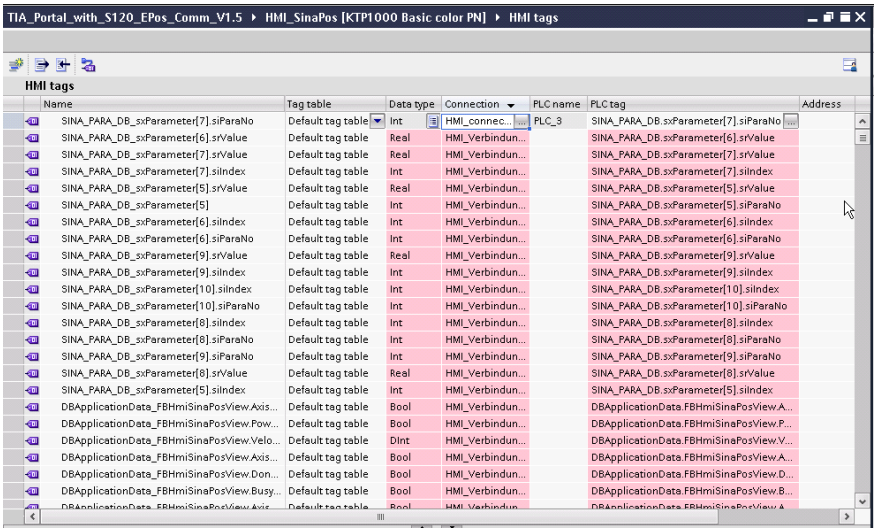
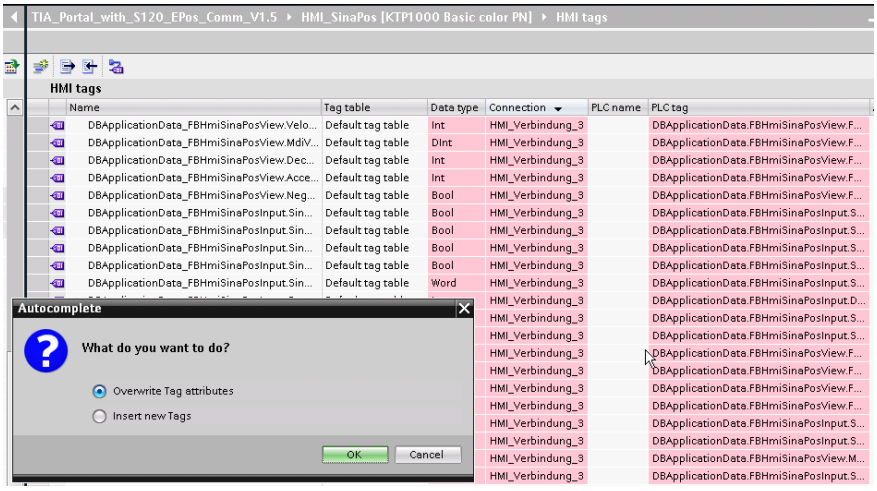
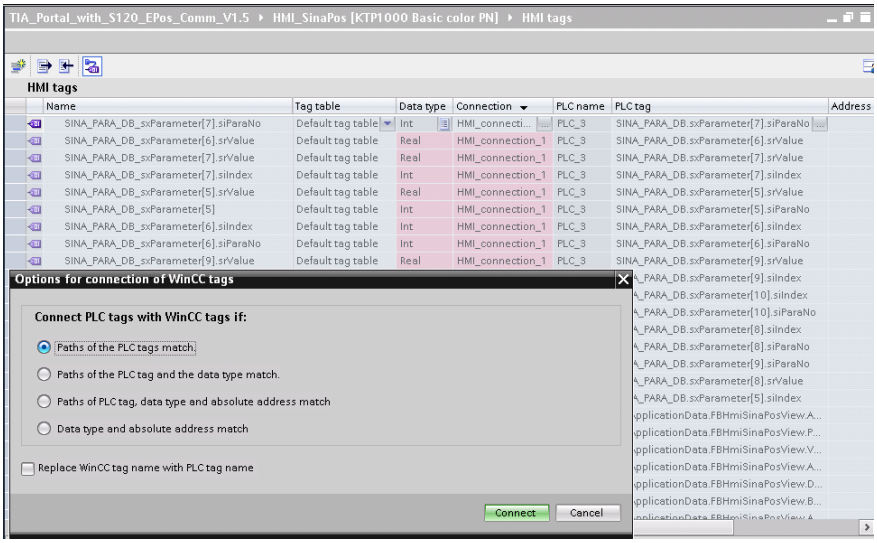
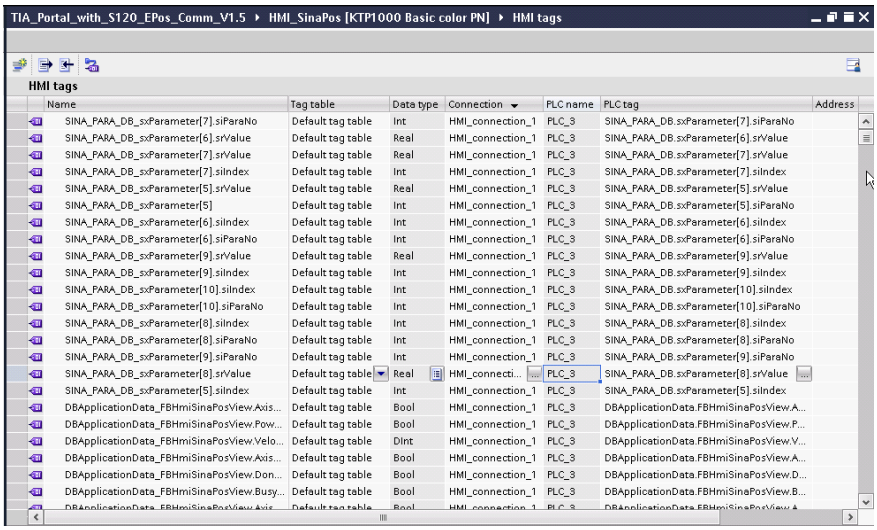
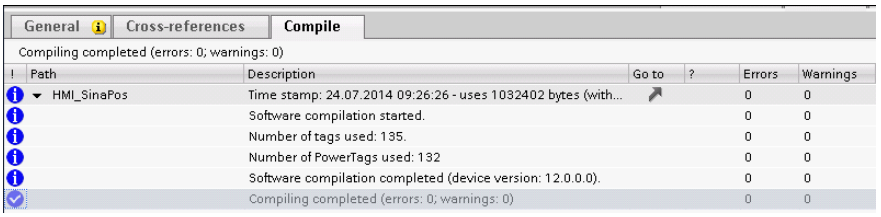
No.	Action
1.	Integrate the drive using the GSD (in the example, S120), and connect to the SIMATIC controller 
2.	**Important: Align the IP address setting and Profinet name with the initialized values**
3.	Create the drive objects as well as the corresponding drive telegram (telegram 111) Device overview Module | ... | Rack | Slot | I address | Q addr... | Type | Order number | Firmware SINAMICS-S120-CU320PN-V... | 0 | 0 | SINAMICS S120/S1... | 6SL3 040-1MA01-0AA0 | V4.6 PN-IO | 0 | 0 X150 | SINAMICS-S120-CU... DO SERVO_1 | 0 | 1 | DO SERVO Module Access Point | 0 | 1 1 | Module Access Point 0 | 1 2 Axis_1 | 0 | 1 3 | 68...91 | 64...87 | SIEMENS telegram... 0 | 1 4 DO SERVO_2 | 0 | 2 | DO SERVO Module Access Point | 0 | 2 1 | Module Access Point 0 | 2 2 Axis_2 | 0 | 2 3 | 92...115 | 88...111 | SIEMENS telegram...
4.	Compile the SIMATIC CPU / hardware configuration to identify subsequently necessary hardware IDs for use in the program / HMI: 21 SINAMICS-S120-CU320PN-V4.6[He... Hw_SubModule 276 22 DO_SERVO_1 Hw_SubModule 277 23 Axis_1[AI/AO] Hw_SubModule 278 24 DO_SERVO_1(1) Hw_SubModule 279 25 DO_SERVO_2 Hw_SubModule 280 26 Axis_2[AI/AO] Hw_SubModule 281 27 DO_SERVO_2(1) Hw_SubModule 282

Table 5-3 Connection of the configured HMI

No.	Action
1.	Copy the HMI from the project example and connect to the newly created SIMATIC S7-CPU: 
2.	Create a new HMI connection by dragging & dropping: 
3.	Remark: Delete the HMI connections not required 

No.	Action
4.	Change to the view of the HMI variables and change the HMI connection for the first variable (sort according to connection type): 
5.	Select the valid HMI connection and auto-complete (select the field and pull down with a right mouse click) the connection for all remaining HMI connections, which up until now were incorrect: 

No.	Action
6.	Select all HMI variables with an HMI connection to the PLC, and reconnect all variables: 
7.	HMI variables are now newly connected to the SIMATIC S7-PLC: 
8.	After the HMI has been compiled, this part of the application is now also ready to run: 

5.3 Configuring the SINAMICS with EPOS topology

Configuring a SINAMICS S120 / G120 with EPOS is not discussed within the scope of this documentation.

You can find information regarding commissioning a SINAMICS with EPOS in the following application examples:

SINAMICS S: Positioning an S110 with S7-300/400 (STEP 7 V5) via PROFINET/PROFIBUS using Safety Integrated (via terminal and PROFIsafe

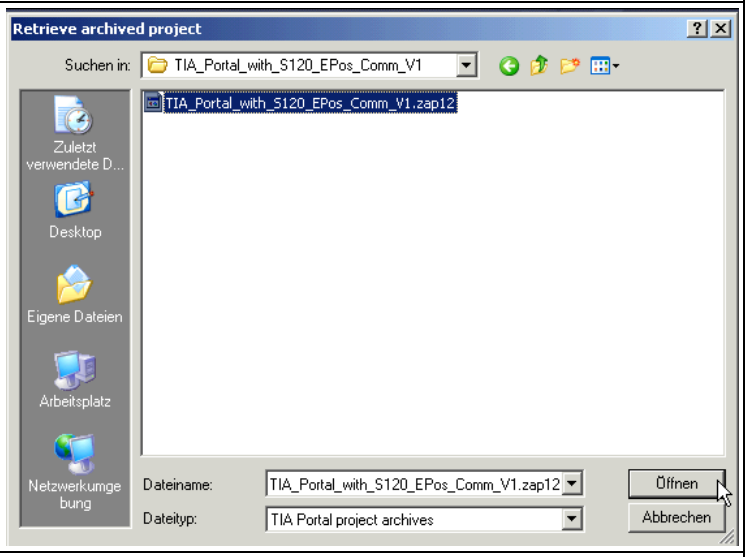
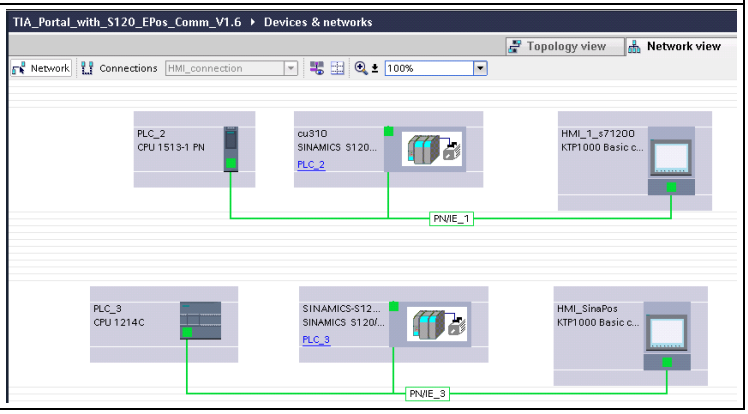
SINAMICS S: Positioning an S120 with S7-300/400 (STEP 7 V5) via PROFIBUS/PROFINET using Safety Integrated (via terminal and PROFIsafe

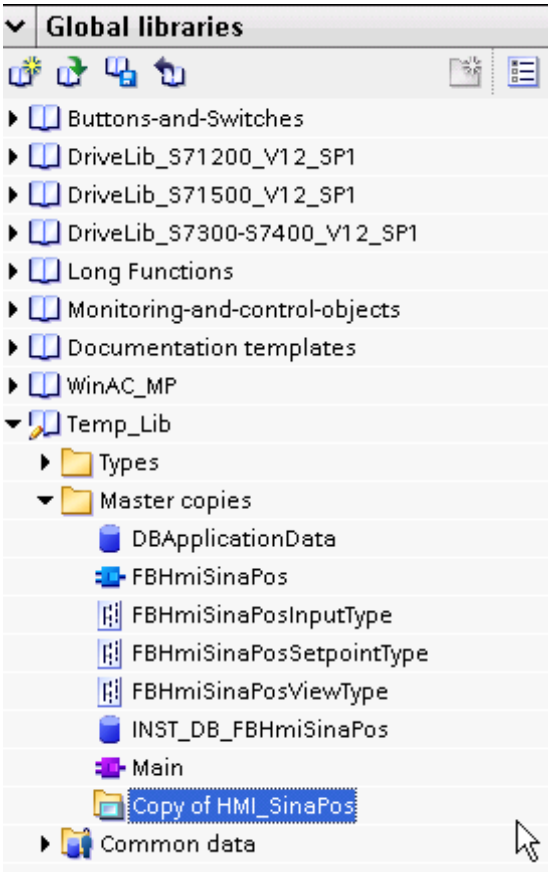
6 Installation and commissioning

Installing the software (download)

This chapter describes the steps required to install the code example.

Table 6-1

No.	Action	Remark
1.	The project example can be downloaded from Siemens Industry Online Support.	http://support.automation.siemens.com/WW/view/en/98961635
2.	Unzip the project example in any directory	
3.	Dearchive the .zap12 file using the TIA Portal	
4.	Open the project – view the configuration example with S7-1200 or S7-1500	

No.	Action	Remark
5.	Adapt the configuration or copy the required components into a new project (blocks / HMI)	User-defined libraries can be used to simply accept the required program sections: 

7 Operating the application

7.1 Overview

The preconfigured HMI is used as a central component of the application example. Using this HMI, the application can be graphically operated using the hardware; simulation is also possible.

Overview and description of the user interface

Fig. 7-1

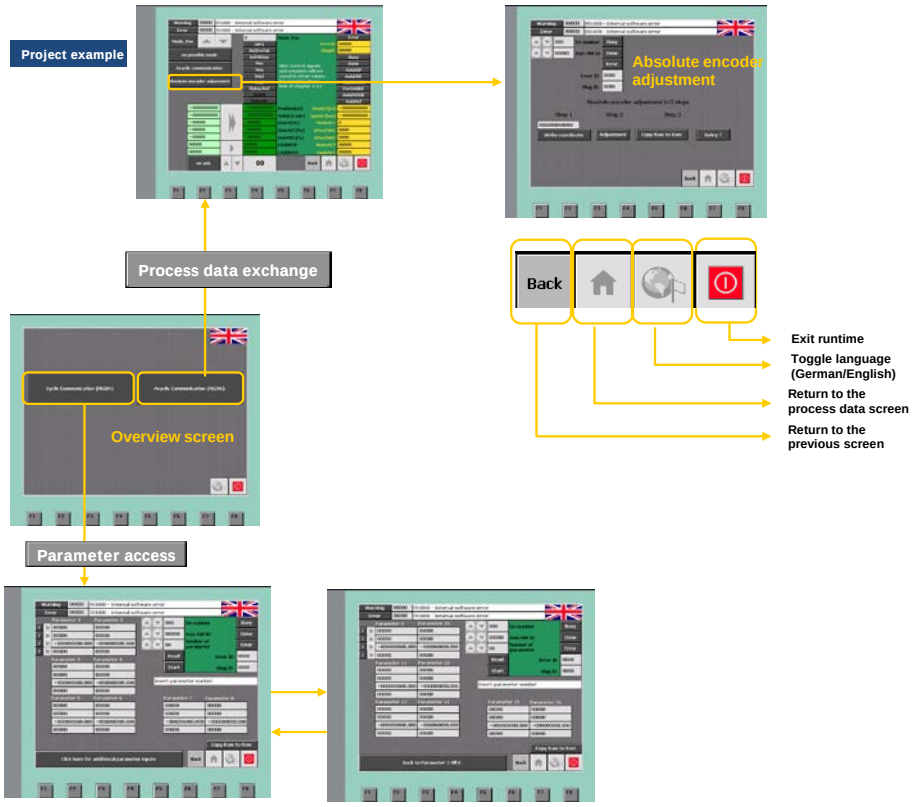
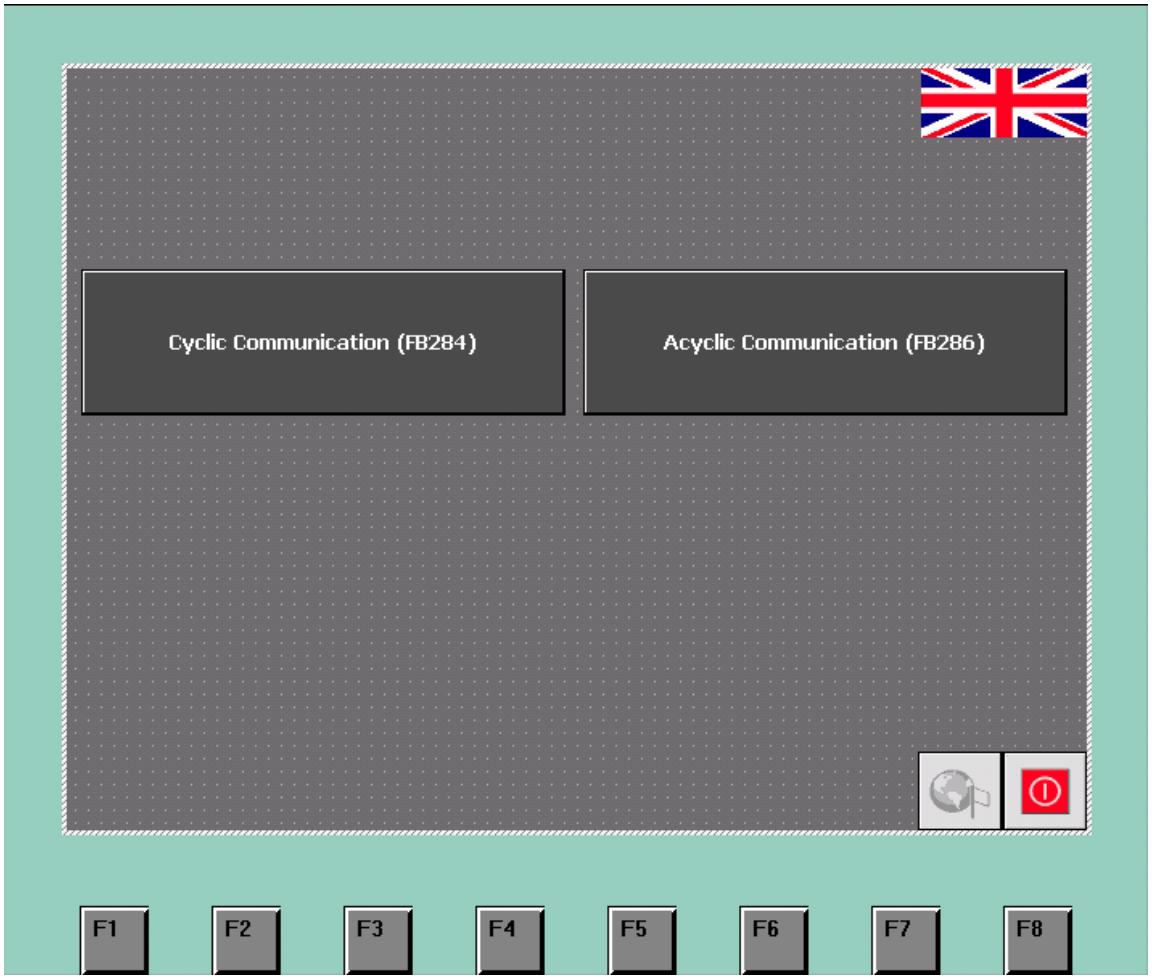


Fig. 7-2 Start screen



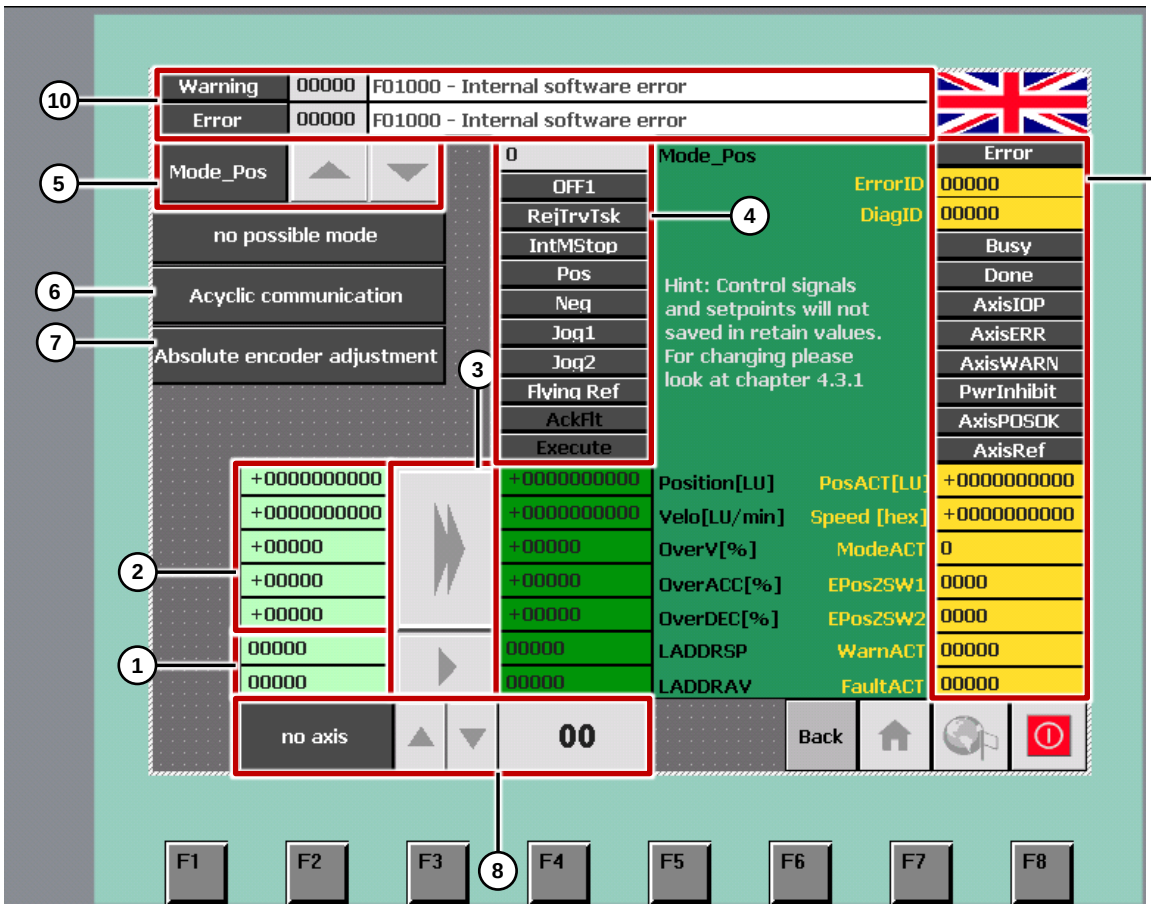
When the HMI is restarted, an overview screen is displayed. Here, it is possible to either start with cyclic or acyclic communication.

7.2 Scenario A - cyclic communication

Table 7-1

No.	Action	Remark
1.	Parameterize the hardware ID	Accept using the button at "3"
2.	Parameterize the MDI setpoints including overrides	Accept using the button at "3"
3.	Accept the specified axis data	Accept the axis setpoints
4.	Control bits of the actual axis	Operating the axis, see Manual FB284
5.	Select the operating mode	Operation/significance of the operating modes, see Manual FB284
6.	Select acyclic communication	Change the HMI screen form
7.	Select absolute encoder adjustment	Change the HMI screen form
8.	Select xth axis	Axis change function
9.	Actual status of the selected axis	
10.	Actual alarm/fault of the actual axis	

Fig. 7-3 Cyclic communication with SINA_POS

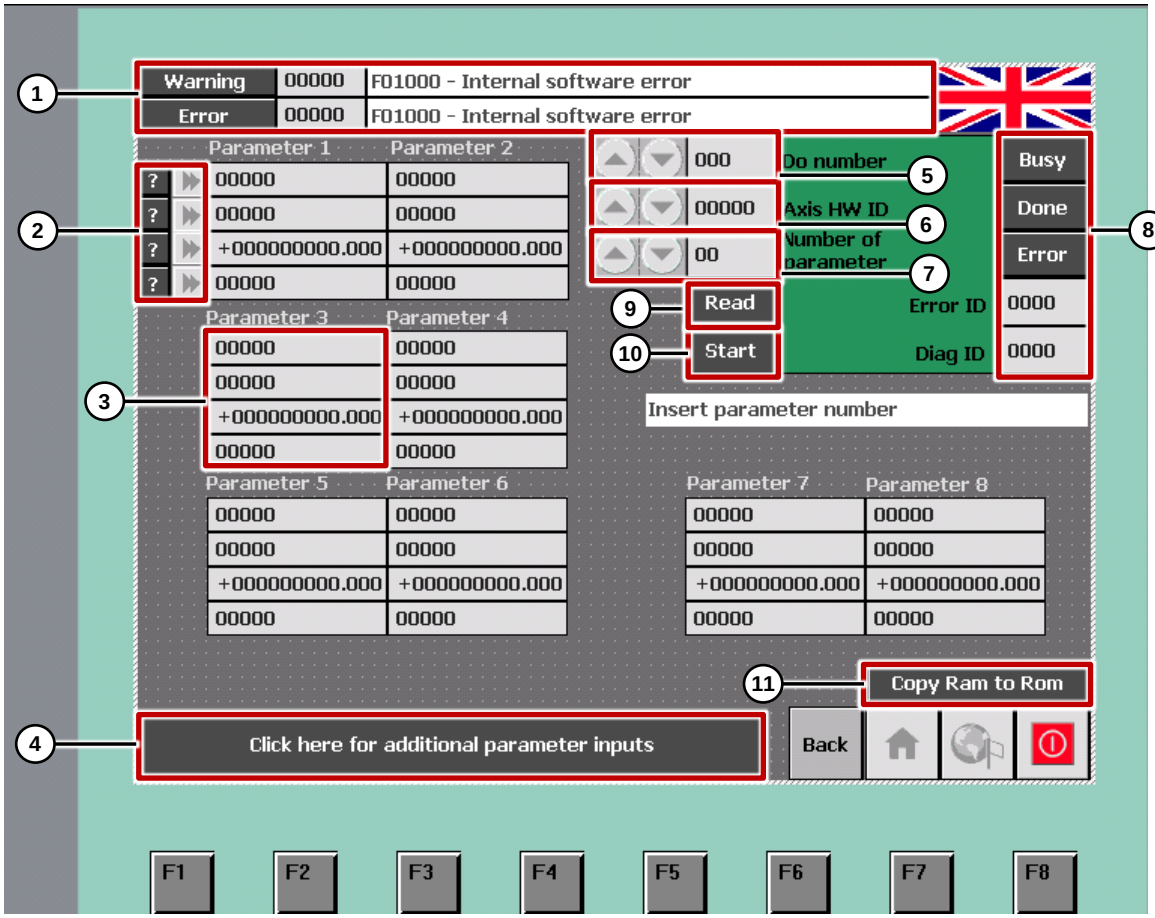


7.3 Scenario B - acyclic communication

Table 7-2

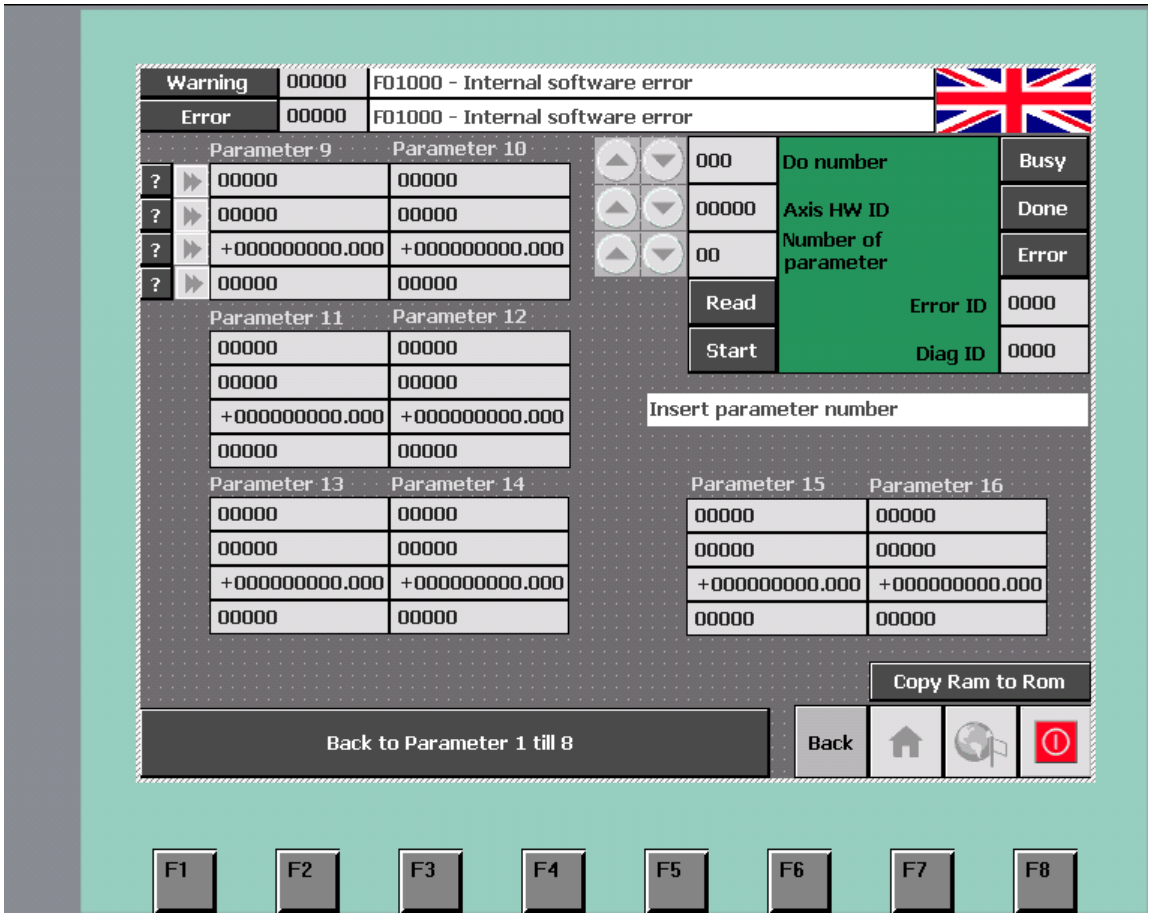
No.	Action	Remark
1.	Actual alarm/fault of the actual axis	
2.	Info buttons	Display information about the values to be entered
3.	Parameter input/output field	
4.	Change to parameter entries 9-16	
5.	Enter the drive object number	See STARTER Configuration DO number
6.	Enter the HW-ID of the (actual value) slot of the axis	
7.	Number of parameters to be read/written	
8.	Status signals SINA_PARA	
9.	Set the job (read=0, write=1)	
10.	Start of the job	
11.	Special job, copy RAM to ROM	

Fig. 7-4 Acyclic communication SINA_PARA Part1



The significance of the buttons corresponds to screen form SINA_PARA Part1.

Fig. 7-5 Acyclic communication SINA_PARA Part2

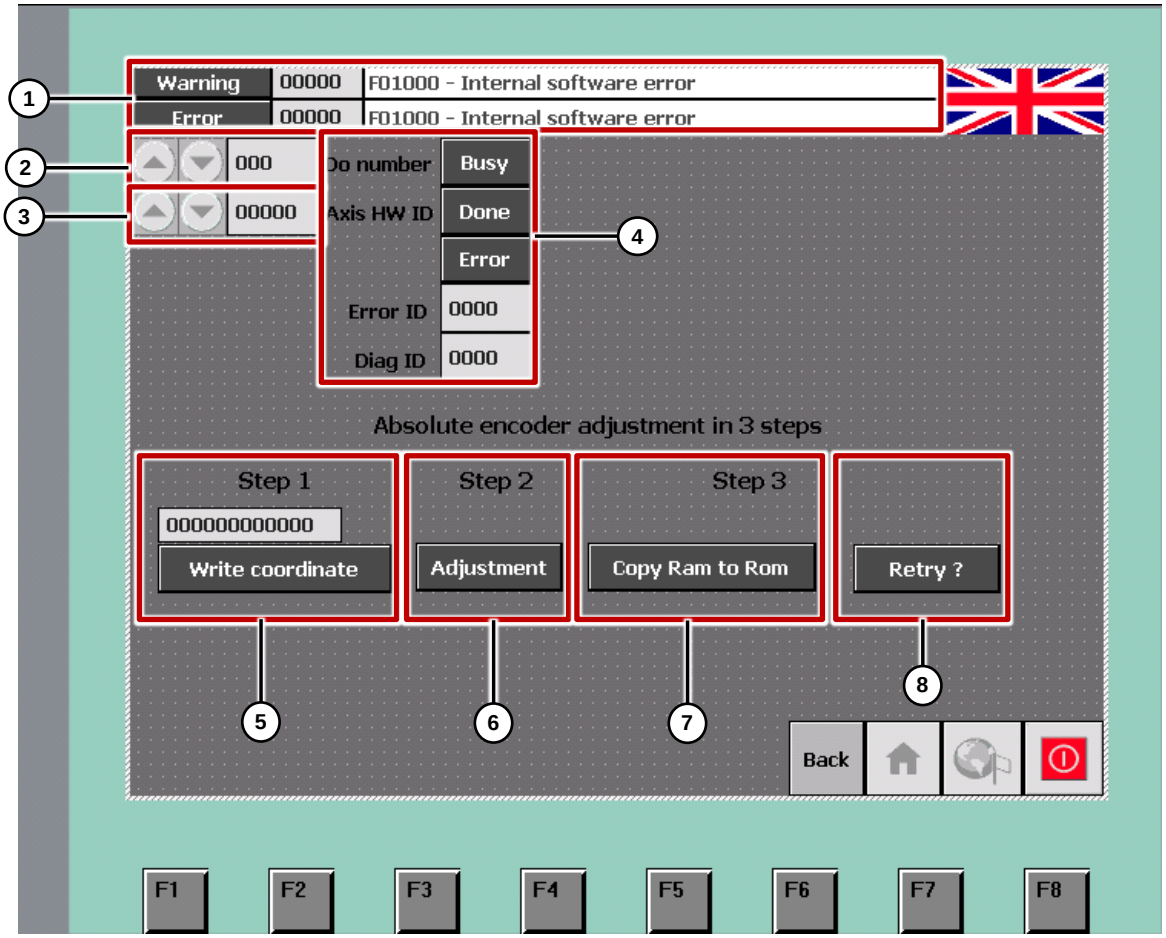


7.4 Scenario C - absolute encoder adjustment

Table 7-3

No.	Action	Remark
1.	Actual faults and alarms	
2.	Select drive object/axis	See STARTER Configuration DO number
3.	Enter the HW-ID of the (actual value) slot of the axis	
4.	Status signals of the SINA_PARA	
5.	Step 1 – write the reference coordinate	
6.	Step 2 – start the adjustment	
7.	Step 3 – copy Ram to Rom	
8.	Step 4 – repeat step chain	

Fig. 7-6 Absolute encoder adjustment



8 References

Note This list does not claim to be complete and only provides a selection of suitable information.

Table 9-1

	Topic	Title
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Download page of the article	http://support.automation.siemens.com/WW/view/en/98961635
\3\	Function Manual for S120	http://support.automation.siemens.com/WW/view/en/59737625
\4\	Download page of the drive libraries	http://support.automation.siemens.com/WW/view/en/68034568
\5\	STARTER	http://support.automation.siemens.com/WW/view/en/26233208
\6\	EPOS reference application	http://support.automation.siemens.com/WW/view/en/58703073
\7\	EPOS reference application	http://support.automation.siemens.com/WW/view/en/67261457
\8\	Additional application examples	http://siemens.com/sinamics-applications

9 Contact person

Siemens AG

Industry Sector
I DT MC PMA APC
Frauenauracher Strasse 80
D - 91056 Erlangen, Germany
E-mail: tech.team.motioncontrol@siemens.com

10 History

Table 11-1

Version	Date	Revision
V1.0	08/2014	First Edition