SIEMENS

Blocks for activating the SINAMICS with SIMATIC S7-1200/1500 in the TIA Portal

Operating Manual

09/2023

Legal information

Warning notice system

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

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

NOTICE

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

⚠ WARNING

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ⓟ 09/2023 Subject to change

Table of contents

Safety information 1

1.1 Fundamental safety instructions

1.1.1 General safety instructions

⚠ WARNING

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

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

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

⚠ WARNING

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

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

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

1.1.2 Warranty and liability for application examples

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

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

1.1.3 Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity 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 cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit
https://www.siemens.com/global/en/products/automation/topic-areas/industrial-cybersecurity.html.

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

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
https://new.siemens.com/global/en/products/services/cert.html.

Further information is provided on the Internet:

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

⚠ WARNING

Unsafe operating states resulting from software manipulation

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

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

Introduction

2

2.1 Overview

Introduction

The function blocks for the cyclic and acyclic communication are used for the simple connection of various SINAMICS S/G/V converter systems.

Each communication block can be used for an axis of a SINAMICS S120 multi-axis system or a SINAMICS S110, SINAMICS V90 or G120x converter system.

The supported communication paths are intended for PROFIBUS and PROFINET bus systems.

Overview of the automation task

The following diagram provides an overview of the automation task.

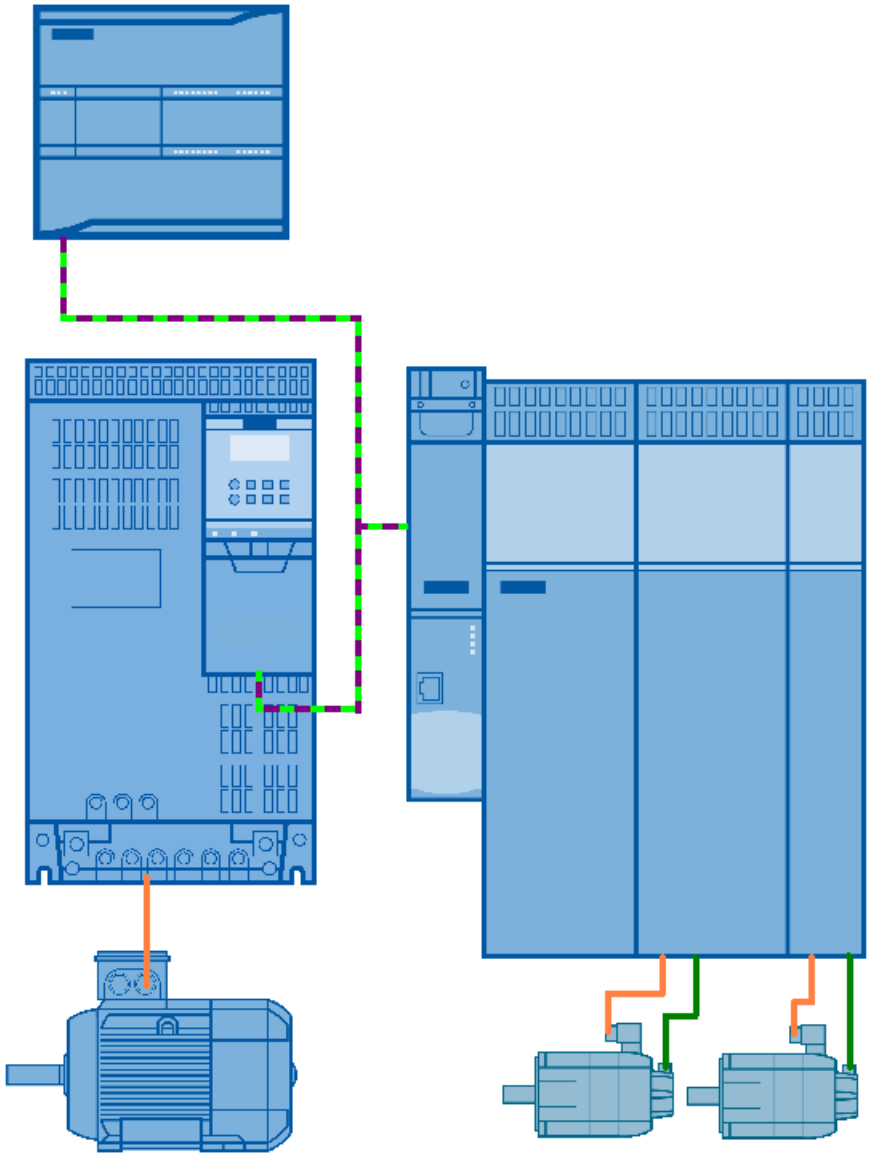


Figure 2-1 Overview of the automation task

Description of the automation task

Depending on the type and use of the data, the data exchange between a SIMATIC S7 controller and a SINAMICS drive is performed **cyclically** – for process data – or **acyclically** – for adjustable parameters.

2.2 Requirements

Requirements of the automation task

Requirement	Explanation
Cyclic transfer: Process data transfer	• Fixed telegram length • No structural change during runtime • "Fast" data transfer
Acyclic transfer: Transferring configuration data Commissioning interface Diagnostics	• Variable telegram length • Variable structural change • "Slow" data transfer • All parameters can be read

2.3 Blocks for S7-1200

The blocks in the "DriveLib_S71200" library provide the following functionality:

- Parameterization of a drive in the cyclic data exchange (max. 1 double word per job)
- Parameterization of a drive in the acyclic data exchange (max. 117 words per job)
- Reading the fault buffer of a SINAMICS drive
- Upload/download of drive parameters stored in a job block
- Report communication fault (slave failed or disabled)
- Cyclic and acyclic communication with drives
- Control of the basic positioner and position controller of a SINAMICS drive

2.4 Blocks for S7-1500

The blocks in the "DriveLib_S71500" library provide the following functionality:

- Send control words and setpoints with selectable length (1 to 32 words) to the drive
- Receive status words and actual values with selectable length (1 to 32 words) from the drive
- Parameterization of a drive in the cyclic data exchange (max. 1 double word per job)
- Parameterization of a drive in the acyclic data exchange (max. 117 words per job)
- Reading the fault buffer of a SINAMICS drive
- Upload/download of drive parameters stored in a job block
- Report communication fault (slave failed or disabled)
- Cyclic and acyclic communication with drives
- Control of the basic positioner and position controller of a SINAMICS drive

Configuration of the communication 3

3.1 Overview

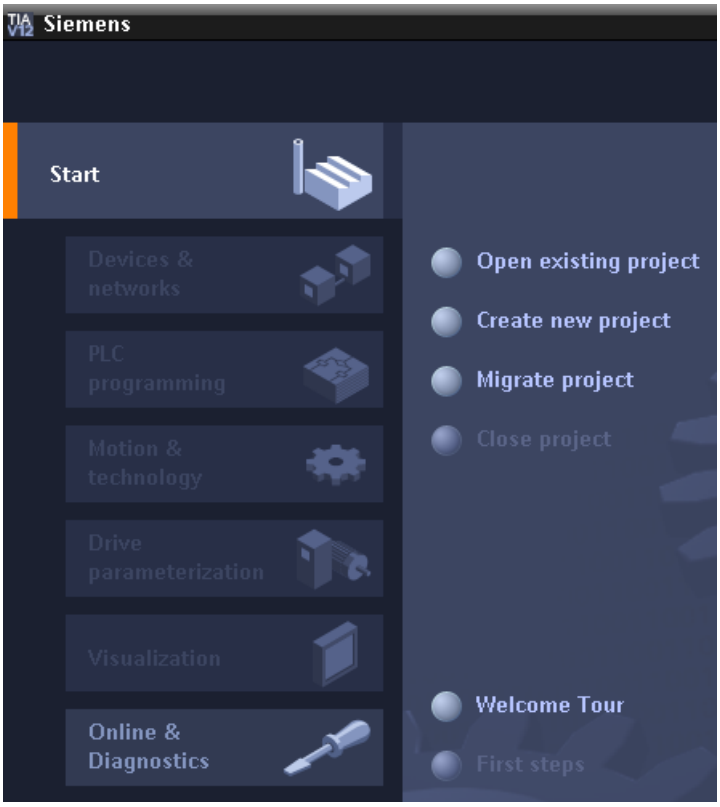
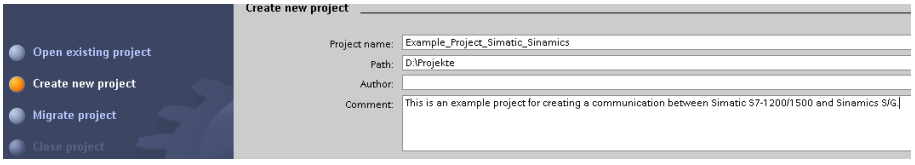
The following configuration steps are required before starting up PROFIBUS communication:

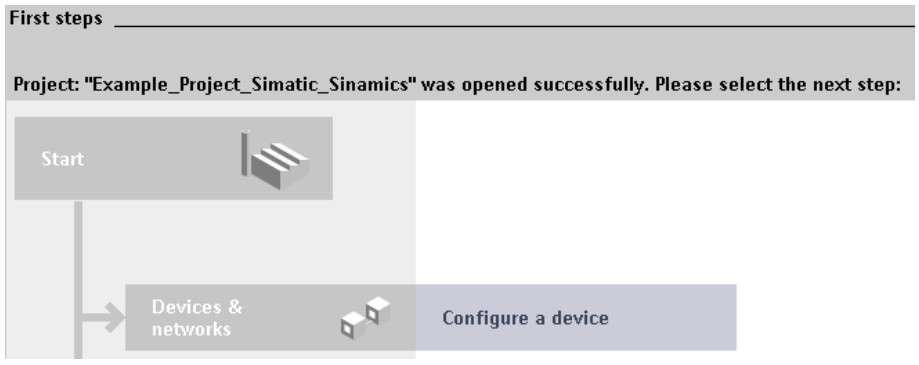
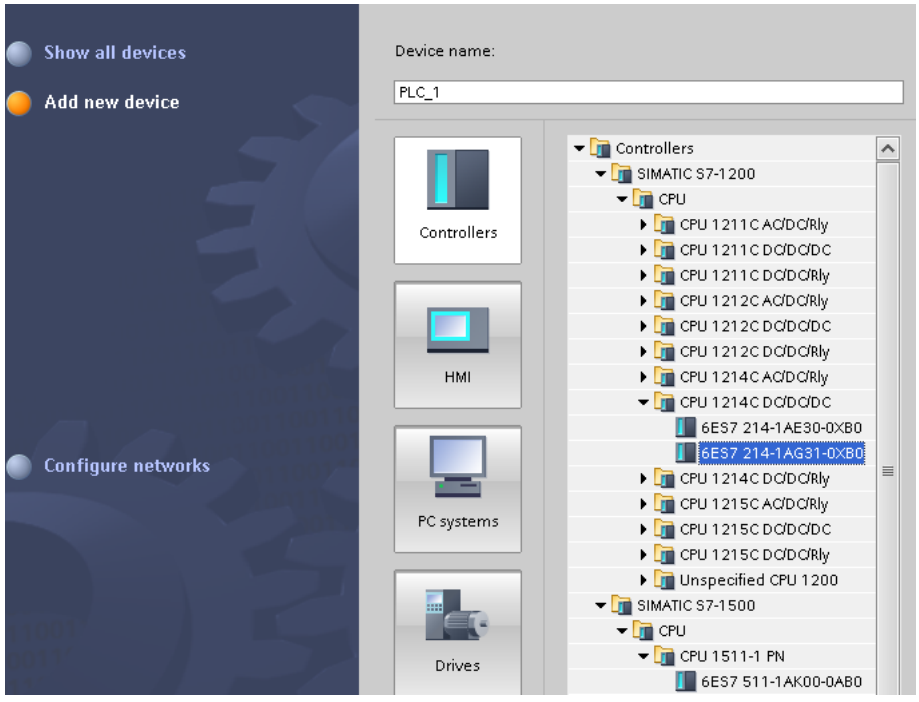
- Configuration and project engineering (Page 16)
- Creating the communications program (Page 38)
- Parameterization of the drives (Page 39)

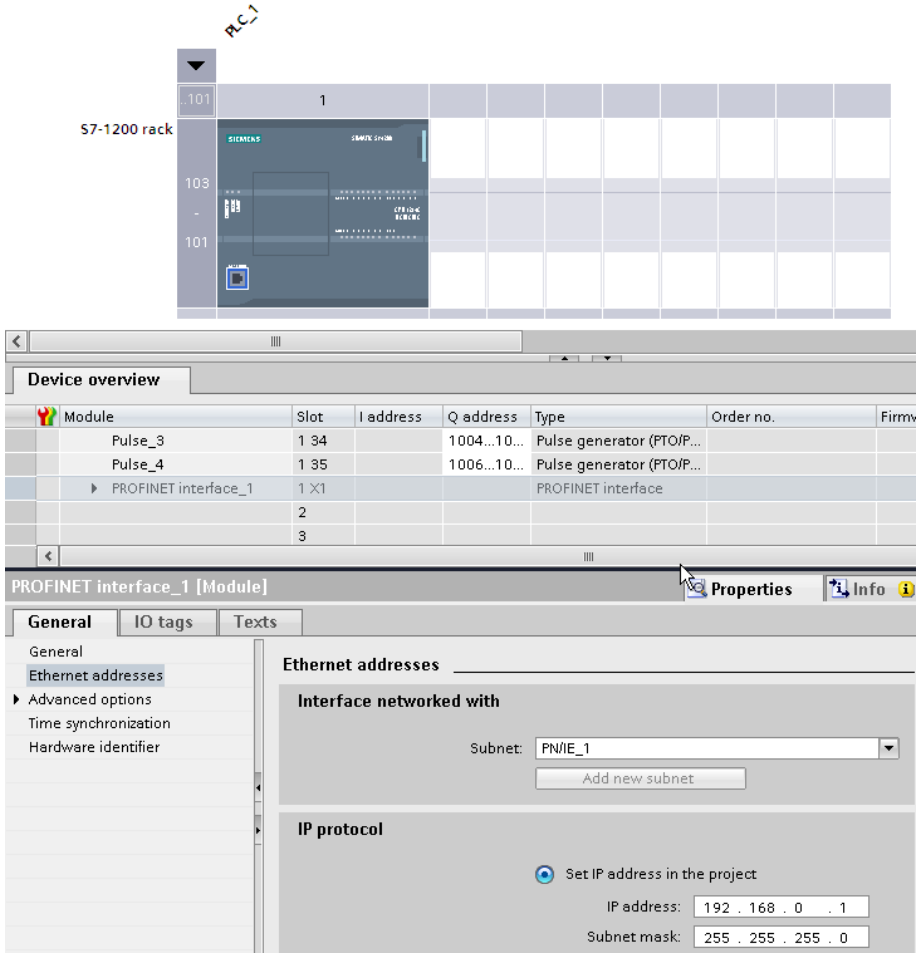
3.2 Configuration and project engineering

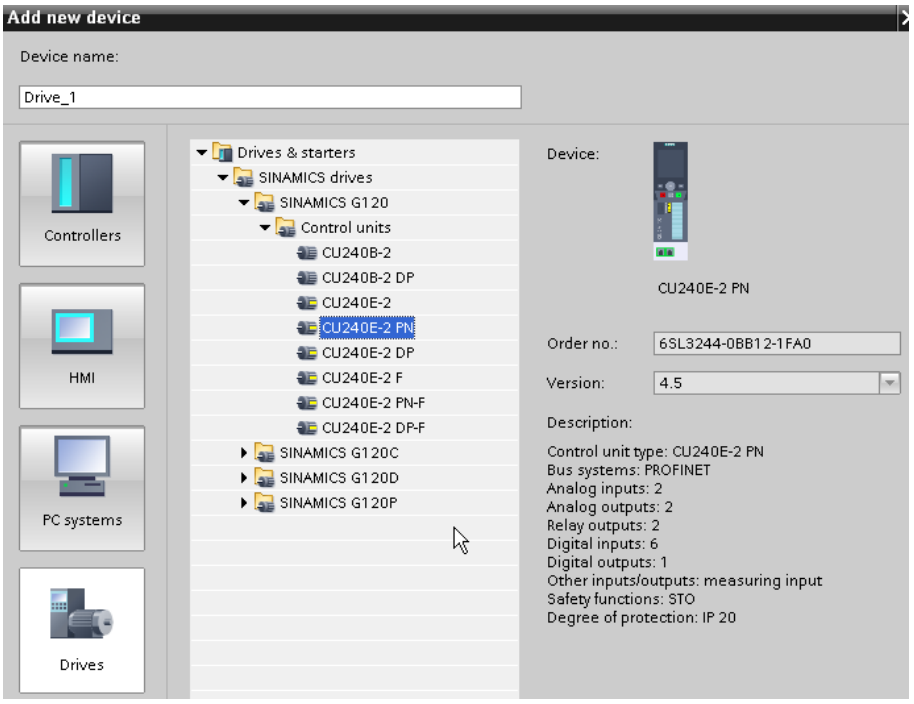
3.2.1 Configuration of the SIMATIC controller S7-1200/1500 with SINAMICS G120 (Startdrive configuration)

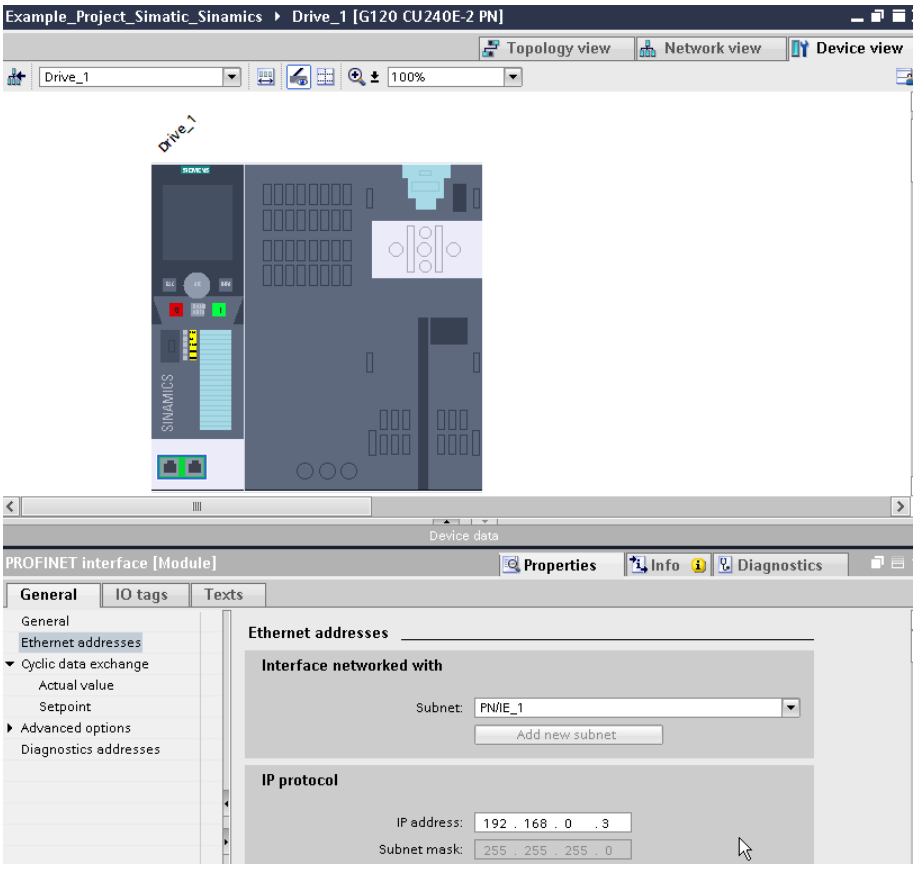
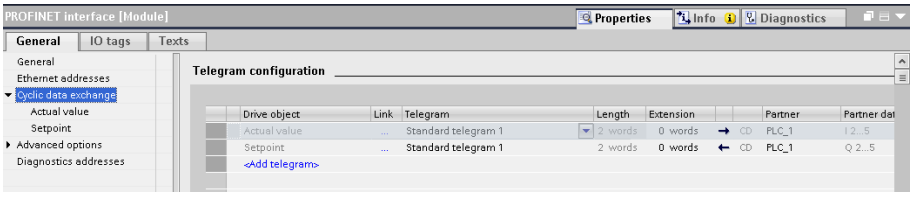
Configuration of the SIMATIC controller S7-1200/1500 with SINAMICS G120 (Startdrive configuration)

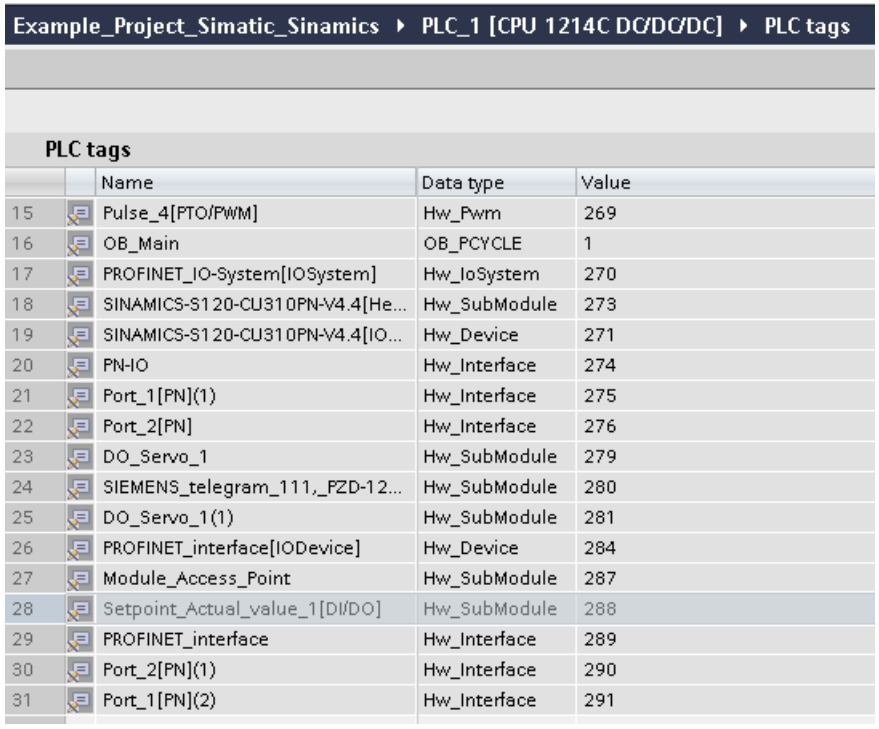
No.	Action	Comment
1.	Start the TIA Portal	
2.	Creating a new project	

No.	Action	Comment
3.	Selection "New device"	
4.	Select the available SIMATIC S7 controller	

No.	Action	Comment
5.	Change to the device view and parameterize the interface as well as the IP/DP address	

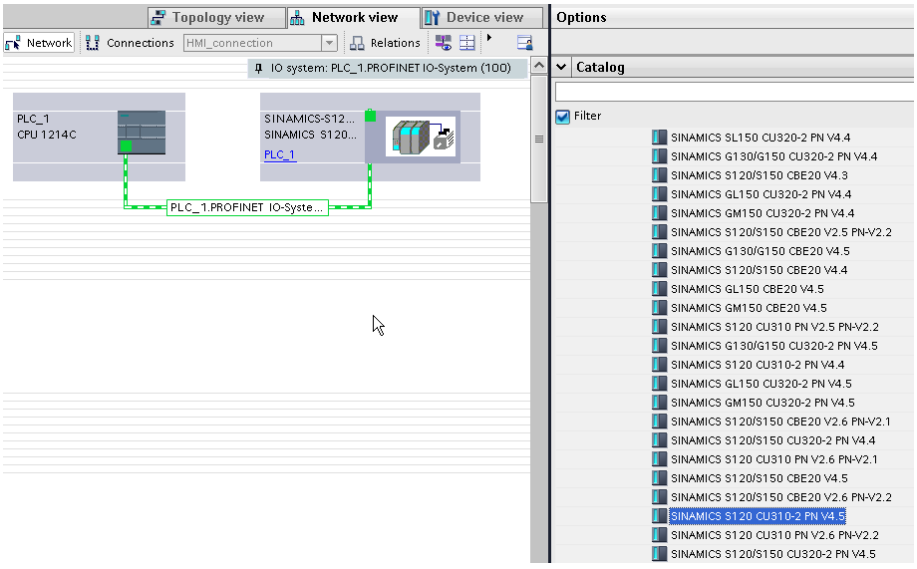
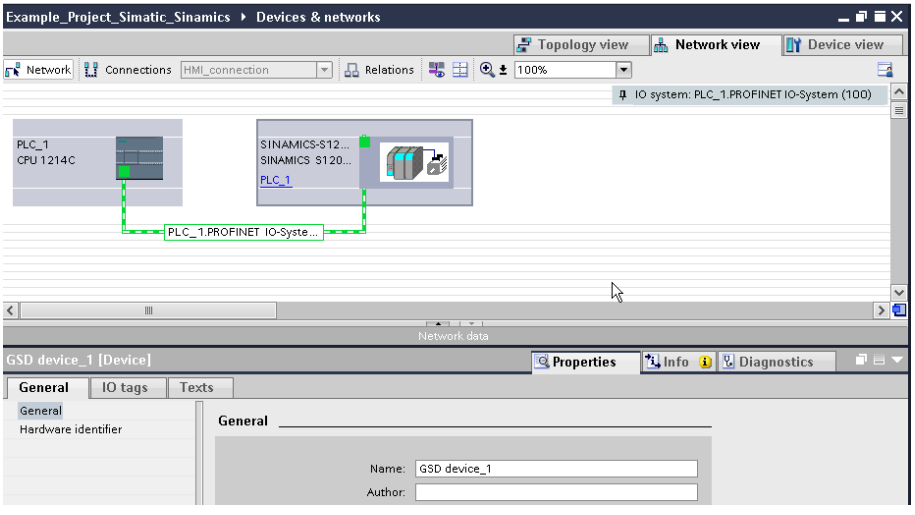
No.	Action	Comment
6.	Create the SINAMICS G120	

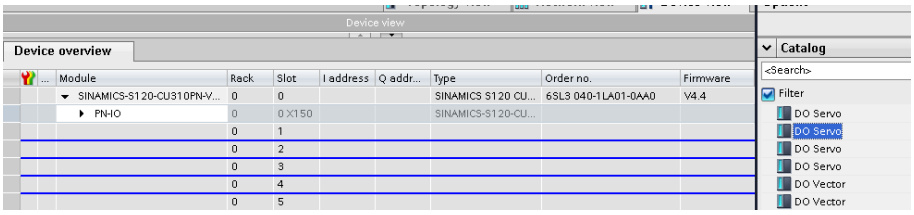
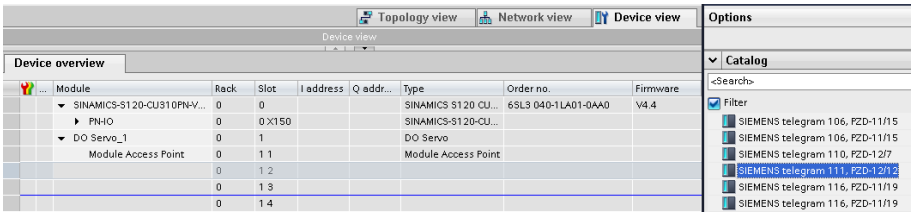
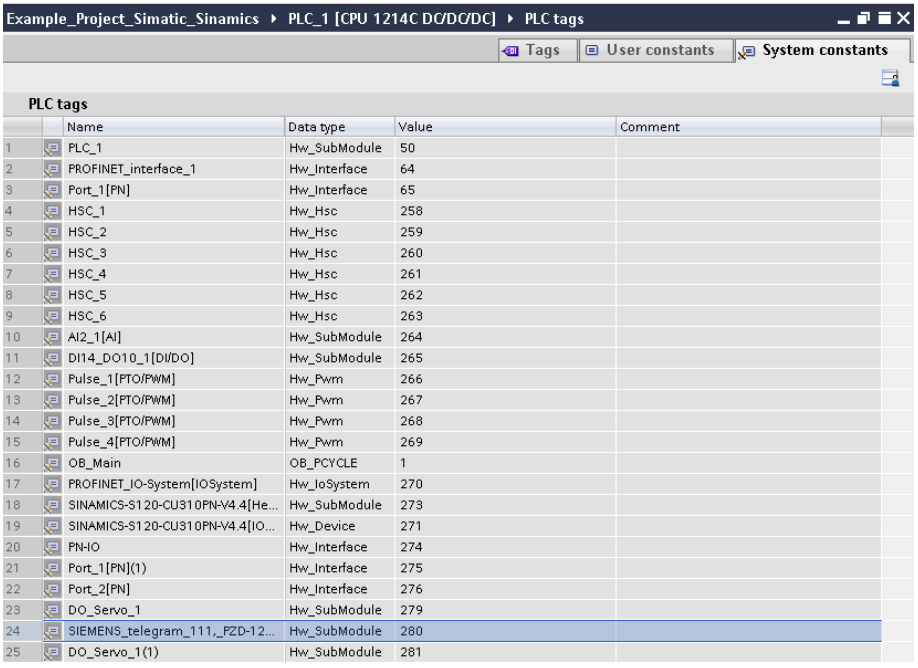
No.	Action	Comment
7.	Change to the device view and parameterize the power unit as well as the Ethernet address of the SINAMICS G120 **Important:** PROFINET nodes configured with Startdrive are linked as standard with the **detailed device name**. For the communication to function, this name must be adapted with the device name assigned by the user.	
8.	**Online commissioning of the SINAMICS drive**	**The commissioning of the SINAMICS drive is not considered in this document. Use the SIEMENS product and information pages for further information, see Chapter** References (Page 212)**. (/1/)**
9.	**After the online basic commissioning, the selected telegram is displayed in the device view.**	

No.	Action	Comment
10.	After the compilation of the hardware: Determine the hardware ID of the telegram slot	
11.	Configuration of the blocks using the hardware ID	See Chapter Selection of the correct hardware submodules (Page 24)

	Name	Data type	Value
15	Pulse_4[PTO/PWM]	Hw_Pwm	269
16	OB_Main	OB_PCYCLE	1
17	PROFINET_IO-System[IOSystem]	Hw_IoSystem	270
18	SINAMICS-S120-CU310PN-V4.4[He...	Hw_SubModule	273
19	SINAMICS-S120-CU310PN-V4.4[IO...	Hw_Device	271
20	PN-IO	Hw_Interface	274
21	Port_1[PN](1)	Hw_Interface	275
22	Port_2[PN]	Hw_Interface	276
23	DO_Servo_1	Hw_SubModule	279
24	SIEMENS_telegram_111,_PZD-12...	Hw_SubModule	280
25	DO_Servo_1(1)	Hw_SubModule	281
26	PROFINET_interface[IODevice]	Hw_Device	284
27	Module_Access_Point	Hw_SubModule	287
28	Setpoint_Actual_value_1[DI/DO]	Hw_SubModule	288
29	PROFINET_interface	Hw_Interface	289
30	Port_2[PN](1)	Hw_Interface	290
31	Port_1[PN](2)	Hw_Interface	291

3.2.2 Configuration of the SIMATIC controller S7-1200/1500 with SINAMICS S120 (GSD configuration)

No.	Action	Comment
12.	Steps 1 to 5 of Chapter Configuration of the SIMATIC controller S7-1200/1500 with SINAMICS G120 (Startdrive configuration) (Page 16)	Create the SIMATIC controller
13.	Change to the network view and configure the communication partners, such as S120, via GSD **Important:** The drive must be assigned to an appropriate PLC so that HW IDs can be created on the CPU	
14.	**IMPORTANT:** PROFINET nodes that have been configured with the GSD file are linked as standard with **GSD_DEVICE_x**! This name must be adapted for a functioning communication with the assigned device name!	

No.	Action	Comment
15.	Configure the drive object in the device view of the SINAMICS drive (GSD configuration) **Important:** An element can be inserted as soon as the editing area is displayed in the center with blue lines!	
16.	Telegram selection	
17.	After the compilation of the hardware → determine the hardware ID of the telegram slot	
18.	Configuration of the blocks using the hardware ID	See Chapter Selection of the correct hardware submodules (Page 24)

Note

Use the SIEMENS product and information pages for more mation on the commissioning of the SINAMICS S120 / G120, see Chapter References (Page 212). (/1/)

3.2.3 Selection of the correct hardware submodules

Selection of the correct hardware submodules

Note

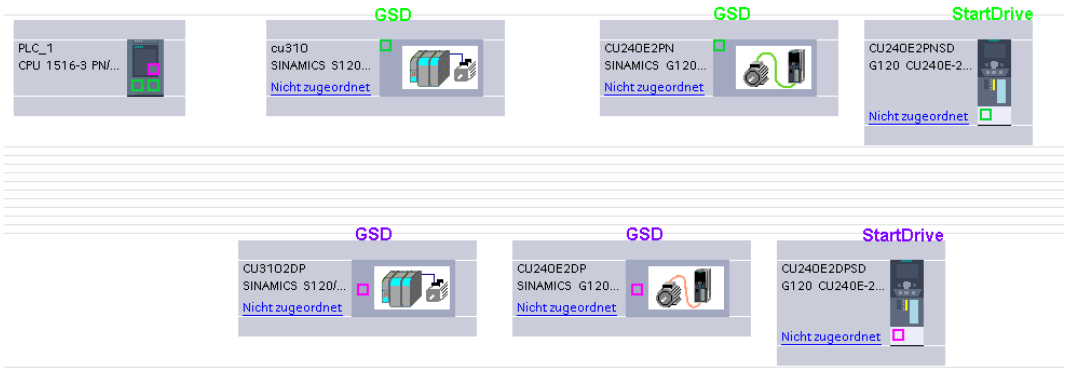
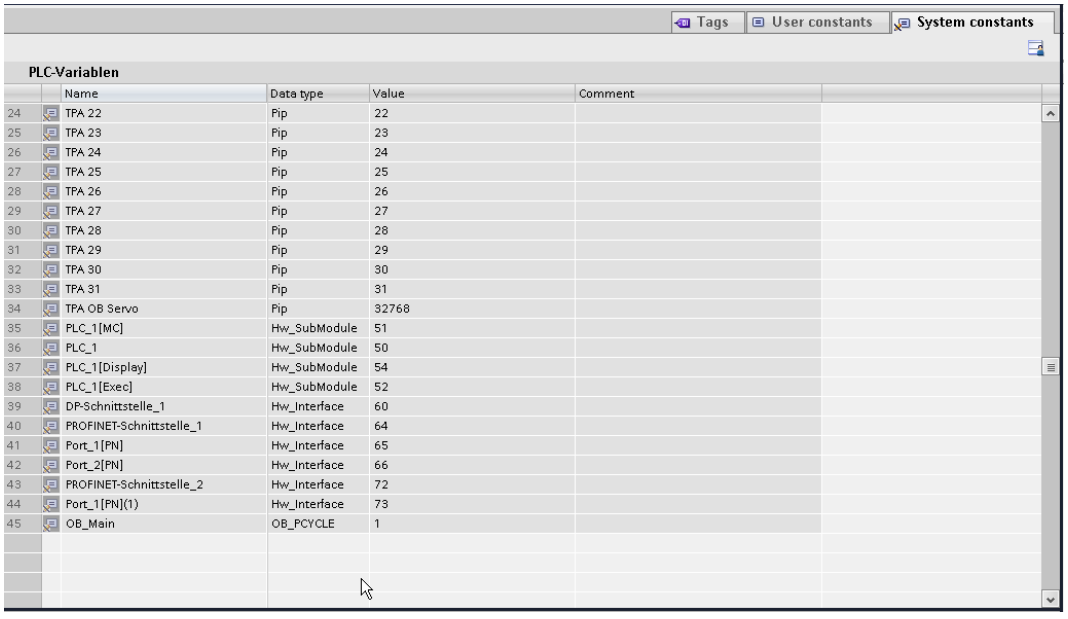
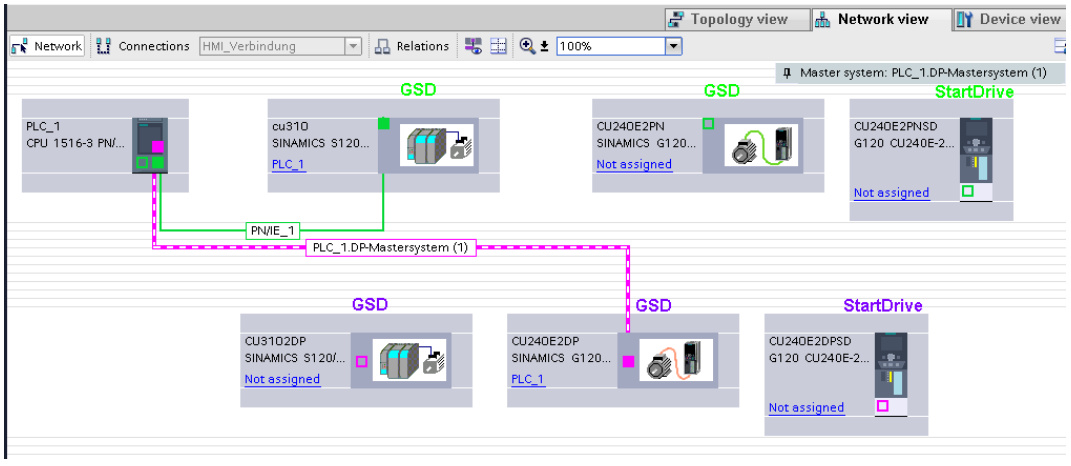
The following screenshots clearly illustrate which hardware IDs are to be used for the communication blocks.

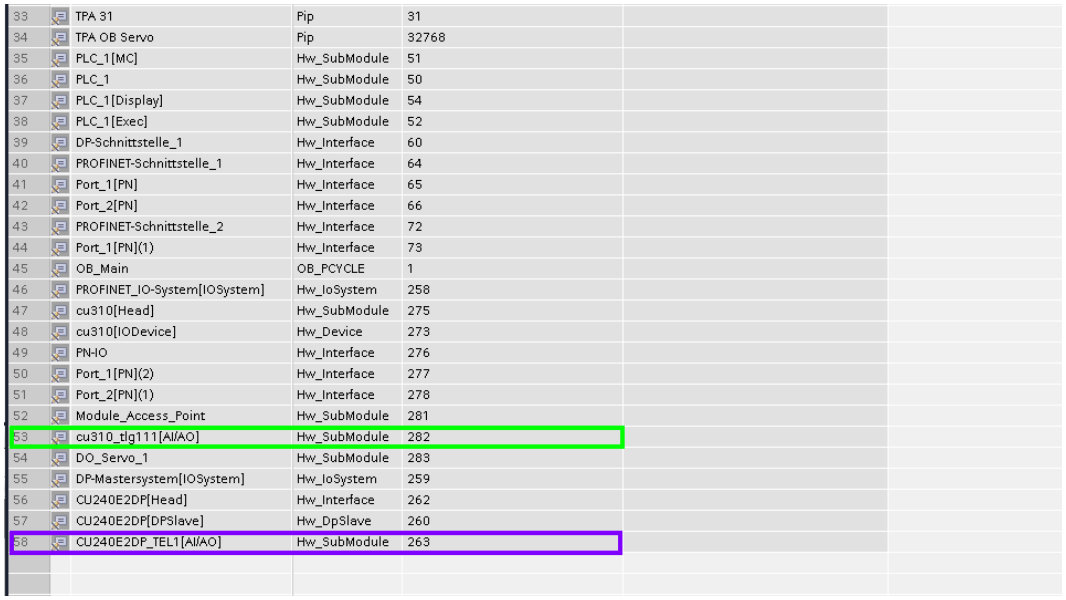
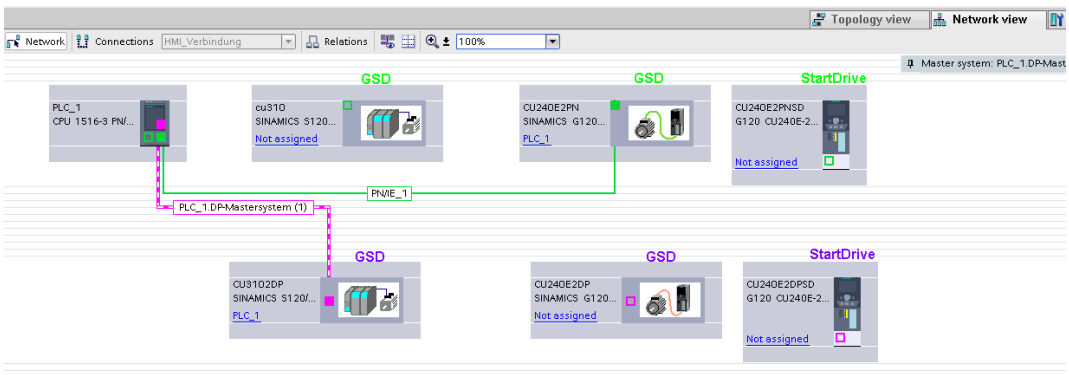
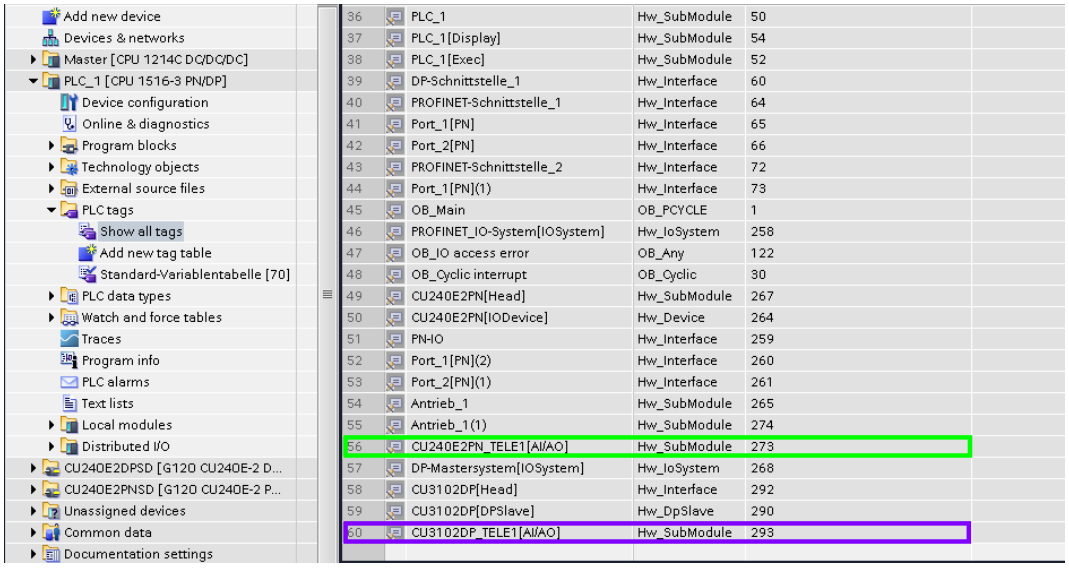
For all variants with only one telegram slot or one ID, this value must be entered at both the HWIDSTW and HWIDZSW inputs. For the variant with two assigned IDs, the appropriate ID must be entered at the corresponding input of the cyclic (!) blocks.

Note

It is possible to individually adapt the telegram name for a GSD configuration. This makes it easier to find the correct hardware ID in the list of the system constants.

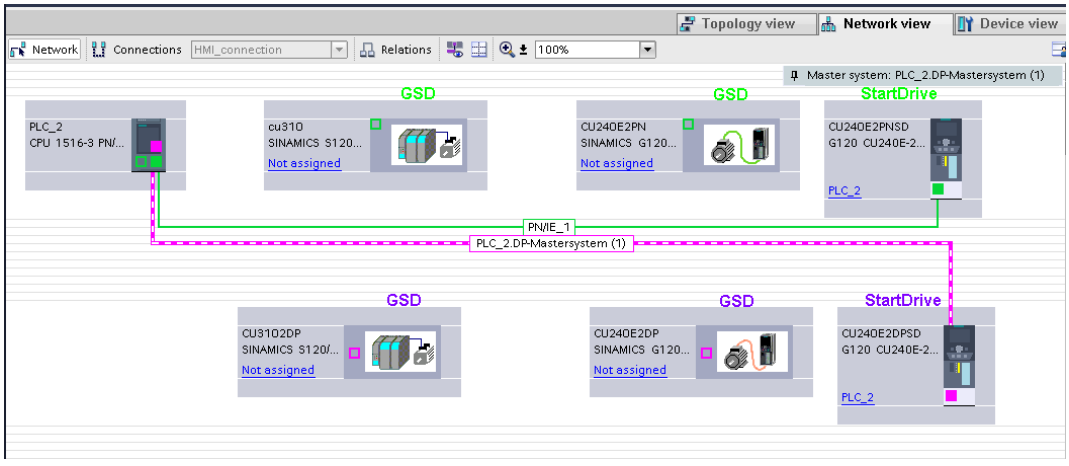
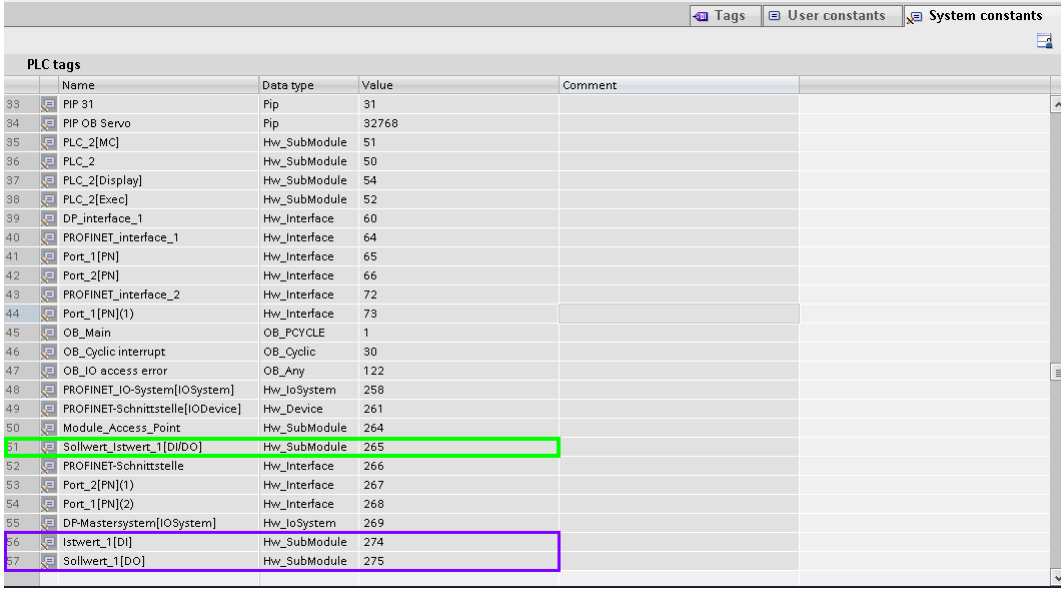
When configuring an S120 multi-axis system, proceed as for the CU310-2 with GSD configuration.

Configuration without connections	
No drive IDs available	
Selecting S120 CU310-2PN and CU240E-2DP via **GSD**	

Selecting S120 CU310-2PN and CU240E-2DP via **GSD**	
Selecting S120 CU310-2DP and CU240E-2PN via **GSD**	
Selecting S120 CU310-2DP and CU240E-2PN via **GSD**	

33	TPA 31	Pip	31
34	TPA OB Servo	Pip	32768
35	PLC_1[MC]	Hw_SubModule	51
36	PLC_1	Hw_SubModule	50
37	PLC_1[Display]	Hw_SubModule	54
38	PLC_1[Exec]	Hw_SubModule	52
39	DP-Schnittstelle_1	Hw_Interface	60
40	PROFINET-Schnittstelle_1	Hw_Interface	64
41	Port_1[PN]	Hw_Interface	65
42	Port_2[PN]	Hw_Interface	66
43	PROFINET-Schnittstelle_2	Hw_Interface	72
44	Port_1[PN](1)	Hw_Interface	73
45	OB_Main	OB_PCYCLE	1
46	PROFINET_IO-System[IOSystem]	Hw_IoSystem	258
47	cu310[Head]	Hw_SubModule	275
48	cu310[IODevice]	Hw_Device	273
49	PN-IO	Hw_Interface	276
50	Port_1[PN](2)	Hw_Interface	277
51	Port_2[PN](1)	Hw_Interface	278
52	Module_Access_Point	Hw_SubModule	281
53	cu310_tlg111[AI/AO]	Hw_SubModule	282
54	DO_Servo_1	Hw_SubModule	283
55	DP-Mastersystem[IOSystem]	Hw_IoSystem	259
56	CU240E2DP[Head]	Hw_Interface	262
57	CU240E2DP[DPSlave]	Hw_DpSlave	260
58	CU240E2DP_TEL1[AI/AO]	Hw_SubModule	263

36	PLC_1	Hw_SubModule	50
37	PLC_1[Display]	Hw_SubModule	54
38	PLC_1[Exec]	Hw_SubModule	52
39	DP-Schnittstelle_1	Hw_Interface	60
40	PROFINET-Schnittstelle_1	Hw_Interface	64
41	Port_1[PN]	Hw_Interface	65
42	Port_2[PN]	Hw_Interface	66
43	PROFINET-Schnittstelle_2	Hw_Interface	72
44	Port_1[PN](1)	Hw_Interface	73
45	OB_Main	OB_PCYCLE	1
46	PROFINET_IO-System[IOSystem]	Hw_IoSystem	258
47	OB_IO access error	OB_Any	122
48	OB_Cyclic interrupt	OB_Cyclic	30
49	CU240E2PN[Head]	Hw_SubModule	267
50	CU240E2PN[IODevice]	Hw_Device	264
51	PN-IO	Hw_Interface	259
52	Port_1[PN](2)	Hw_Interface	260
53	Port_2[PN](1)	Hw_Interface	261
54	Antrieb_1	Hw_SubModule	265
55	Antrieb_1(1)	Hw_SubModule	274
56	CU240E2PN_TELE1[AI/AO]	Hw_SubModule	273
57	DP-Mastersystem[IOSystem]	Hw_IoSystem	268
58	CU3102DP[Head]	Hw_Interface	292
59	CU3102DP[DPSlave]	Hw_DpSlave	290
60	CU3102DP_TELE1[AI/AO]	Hw_SubModule	293

Selecting G120 CU240E-2DP and CU240E-2PN via **Startdrive**	
Selecting G120 CU240E-2DP and CU240E-2PN via **Startdrive** *DP actual value slot* *DP setpoint slot*	

	Name	Data type	Value	Comment
33	PIP 31	Pip	31	
34	PIP OB Servo	Pip	32768	
35	PLC_2[MC]	Hw_SubModule	51	
36	PLC_2	Hw_SubModule	50	
37	PLC_2[Display]	Hw_SubModule	54	
38	PLC_2[Exec]	Hw_SubModule	52	
39	DP_interface_1	Hw_Interface	60	
40	PROFINET_interface_1	Hw_Interface	64	
41	Port_1[PN]	Hw_Interface	65	
42	Port_2[PN]	Hw_Interface	66	
43	PROFINET_interface_2	Hw_Interface	72	
44	Port_1[PN](1)	Hw_Interface	73	
45	OB_Main	OB_PCYCLE	1	
46	OB_Cyclic interrupt	OB_Cyclic	30	
47	OB_IO access error	OB_Any	122	
48	PROFINET_IO-System[IOSystem]	Hw_IoSystem	258	
49	PROFINET-Schnittstelle[IODevice]	Hw_Device	261	
50	Module_Access_Point	Hw_SubModule	264	
51	Sollwert_Istwert_1[DI/DO]	Hw_SubModule	265	
52	PROFINET-Schnittstelle	Hw_Interface	266	
53	Port_2[PN](1)	Hw_Interface	267	
54	Port_1[PN](2)	Hw_Interface	268	
55	DP-Mastersystem[IOSystem]	Hw_IoSystem	269	
56	Istwert_1[DI]	Hw_SubModule	274	
57	Sollwert_1[DO]	Hw_SubModule	275	

Note

A drive created with Startdrive for PROFIBUS creates two (!) slots for actual value and setpoint.

The appropriate HWIDSTW / HWIDZSW must be used for these two slots on the cyclic blocks FB284, FB285! FB286 (SINA_PARA) works with the actual value slot ID!

3.2.4 Configuration of the blocks

3.2.4.1 Notes on installing the block library

Note

The DriveLib library is automatically installed along with the installation of Startdrive.

A SIOS entry (109475044) is available to update the library, from where the current versions of the library can be downloaded.

Note

The installation routine changes from TIA / Startdrive V14. See Installing the block library up to and including TIA Portal V14 (Page 34).

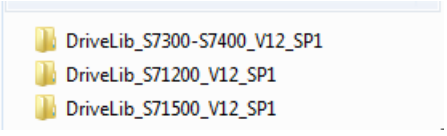
3.2.4.2 Installing the block library up to and including TIA Portal V13 SP1

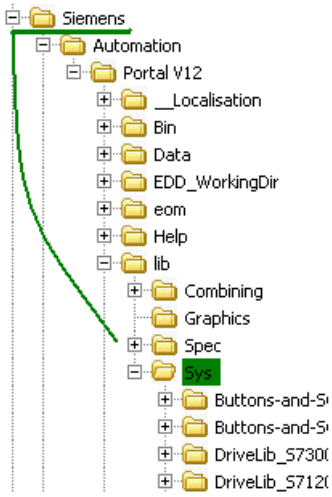
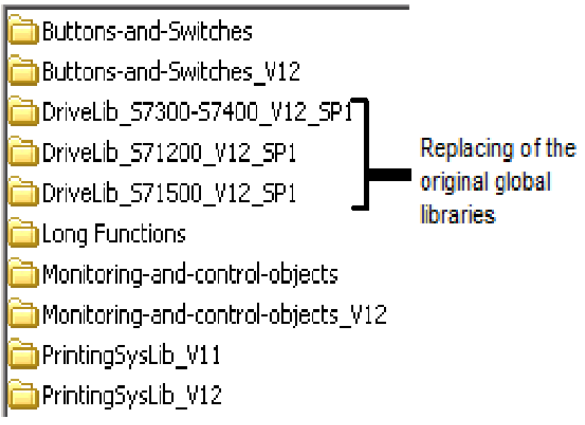
Note

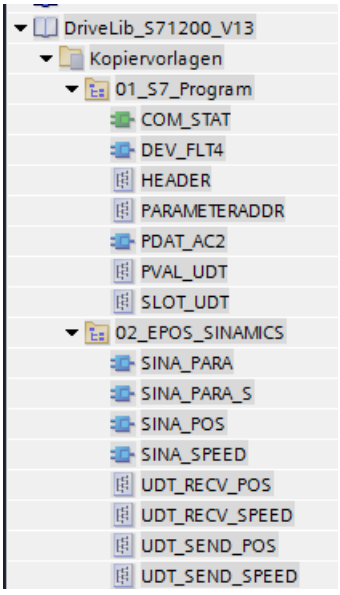
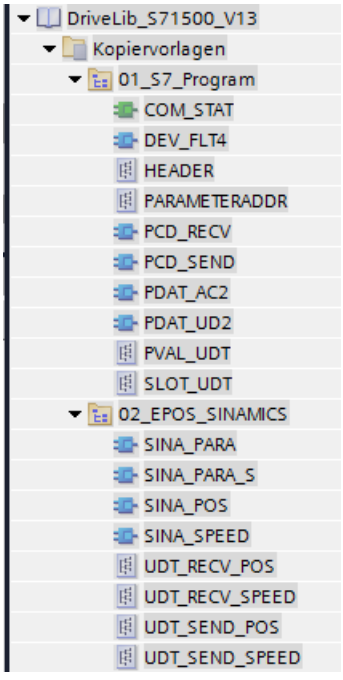
When using the blocks, the library can be downloaded from the Internet free of charge from the SIEMENS product and information pages.

The blocks have been released as of TIA Portal version V12 SP1 and can be used.

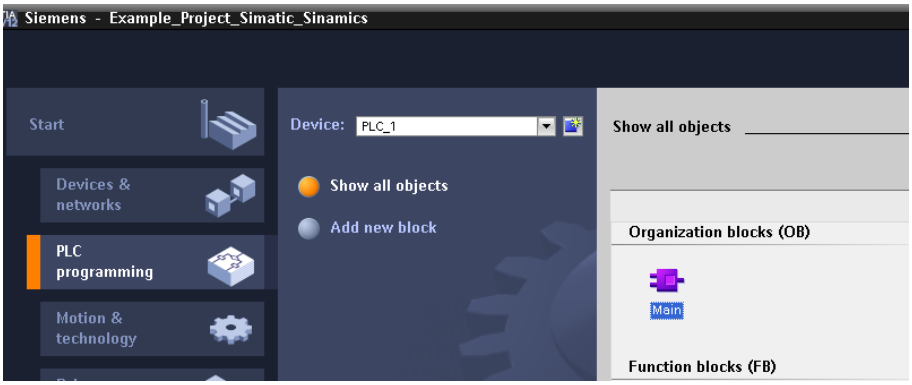
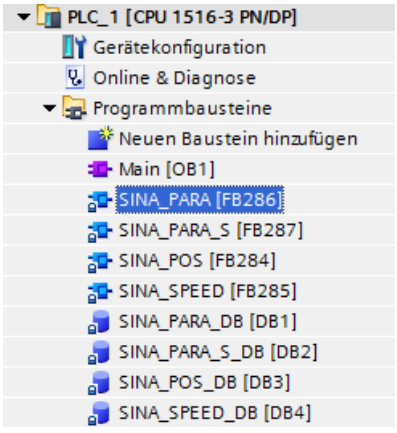
Installing the Drive Library S7-1200/1500 up to and including TIA Portal V13SP1

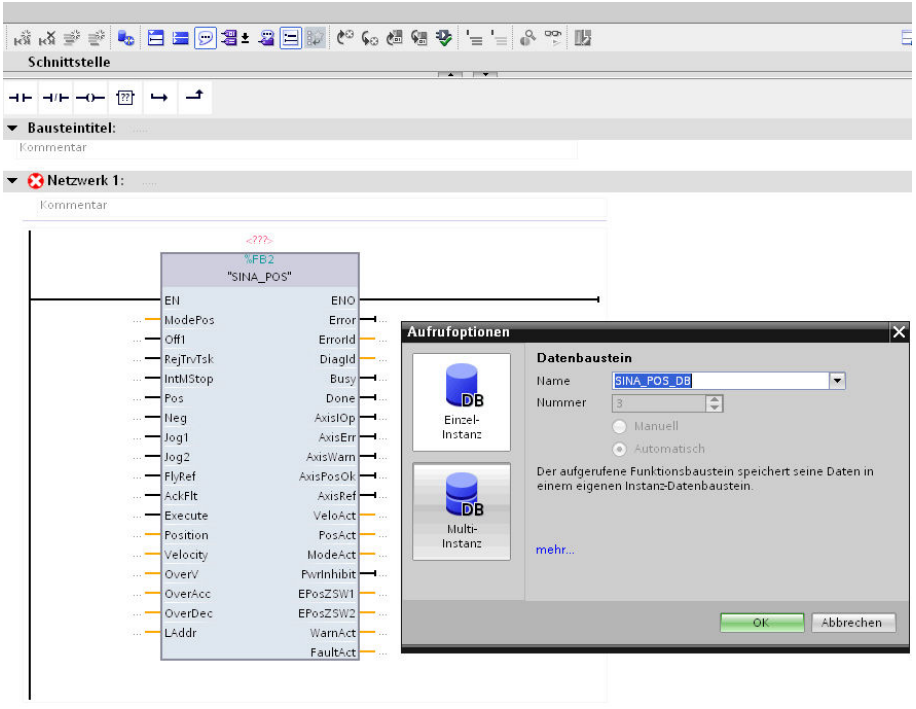
1.	Download the library from the SIEMENS product and information pages and unzip the library to an arbitrary directory	https://support.industry.siemens.com/cs/ww/en/view/109475044
2.	Copy the unzipped directories...	

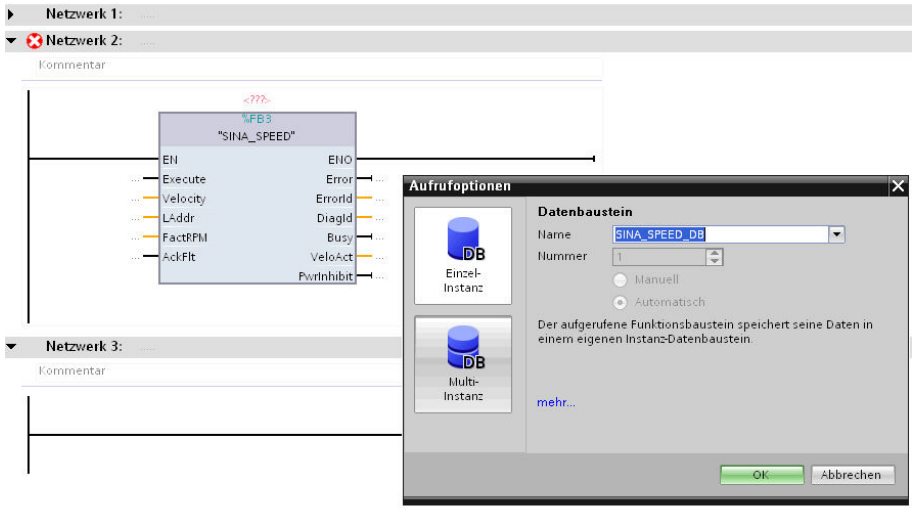
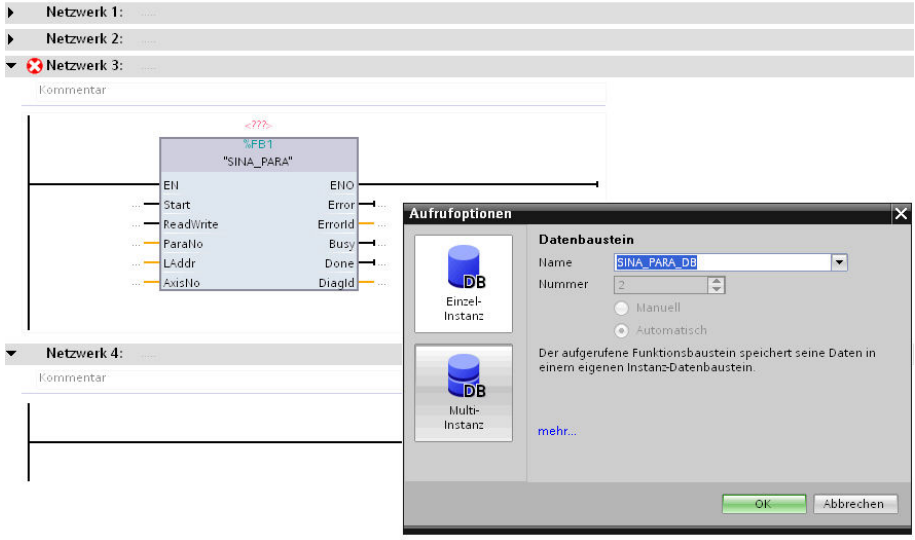
3.	...to the "Sys" installation folder of the TIA Portal installation. Tip: The "Sys" folder of the TIA Portal can be found, for example, using the Windows search for [***.as12**].	
4.	View of the "Sys" folder after copying the libraries	

5.	**View of the installed libraries for an S7-1200**	
6.	**View of the installed libraries for an S7-1500**	

3.2.4.3 Inserting the blocks in the project

1.	Switch to portal view / project view	
2.	Change to the library and select the blocks to be used for the respective SIMATIC S7-CPU	See Chapter Installing the block library up to and including TIA Portal V13 SP1 (Page 28)
3.	Integrate the blocks in the block folder	

4.	Insert the SINA_POS block, for example, in the main (OB1) block Important: When creating, an instance data block is generated automatically, whereby a name and a number must be assigned for the data block in this step	

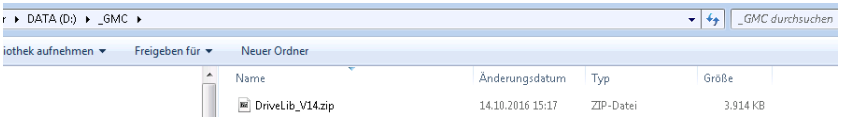
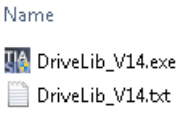
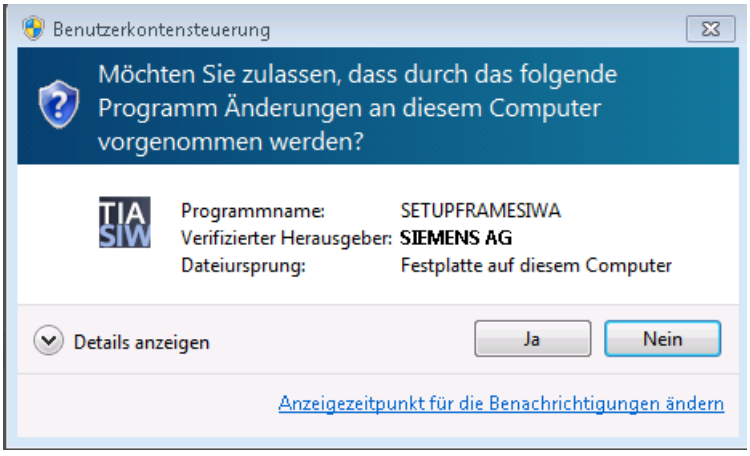
5.	Insert the SINA_SPEED block, for example, in the main (OB1) block Important: When creating, an instance data block is generated automatically, whereby a name and a number must be assigned for the data block in this step	
6.	Insert the SINA_PARA block, for example, in the main (OB1) block Important: When creating, an instance data block is generated automatically, whereby a name and a number must be assigned for the data block in this step	

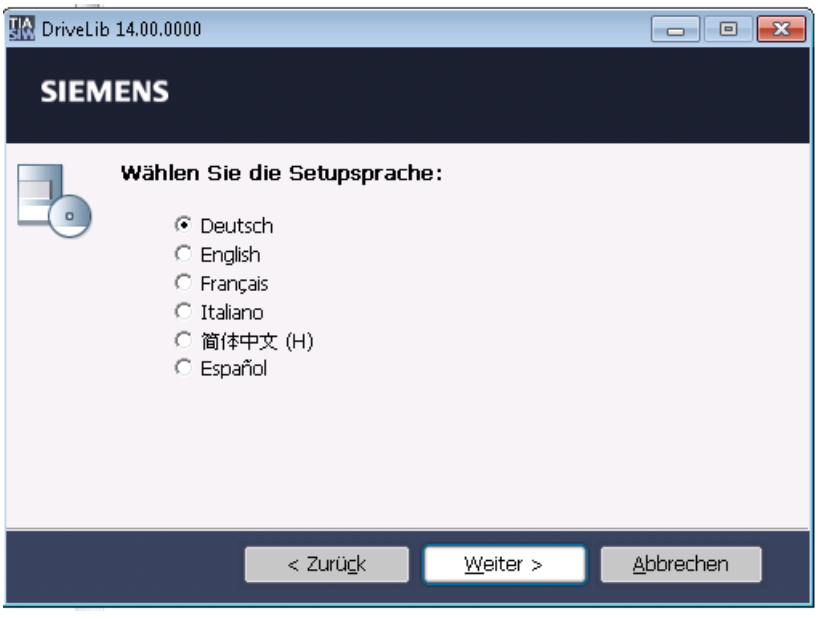
Note

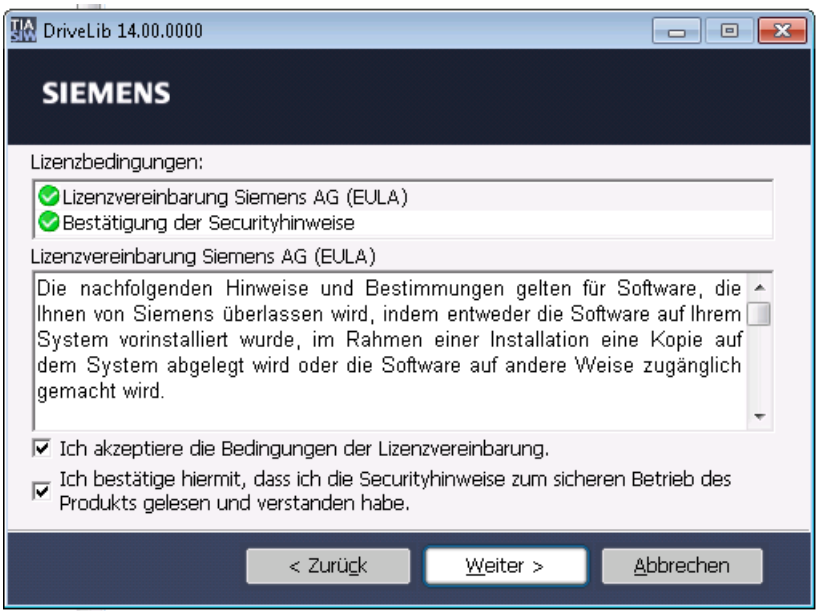
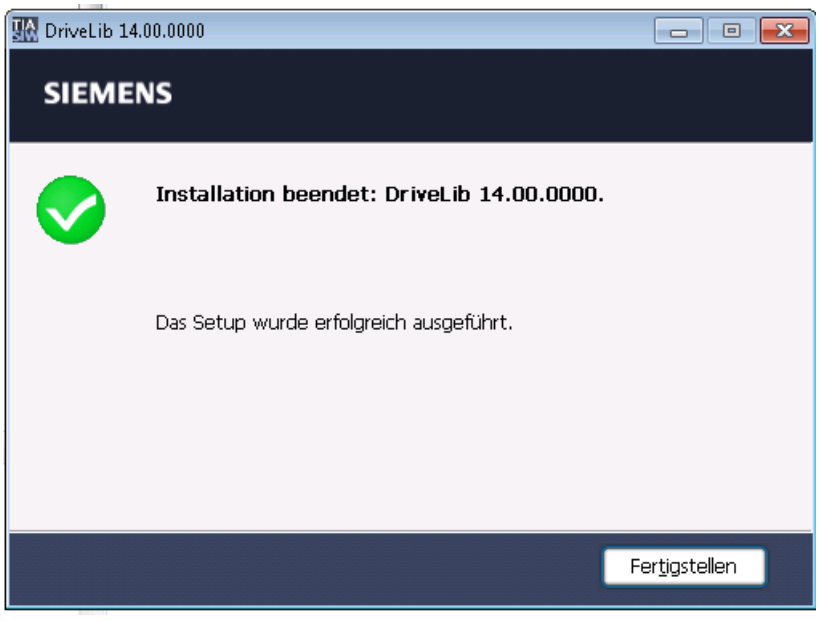
The parameterization of the input and output signals of each block depends on the type of the respective input or output – see also Chapters Function block SINA_POS (FB284) (Page 90)**,** Function block SINA_SPEED (FB285) (Page 109),Function block SINA_PARA (FB286) (Page 113),Function block SINA_PARA_S (FB287) (Page 122) and Function block SINA_INFEED (FB288) (Page 127)**.**

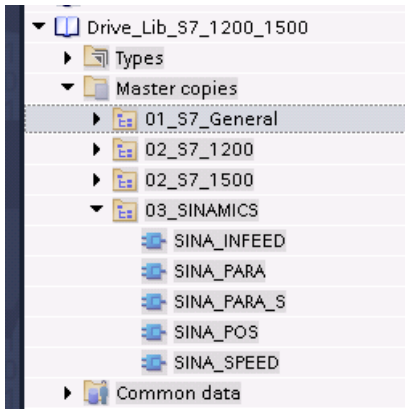
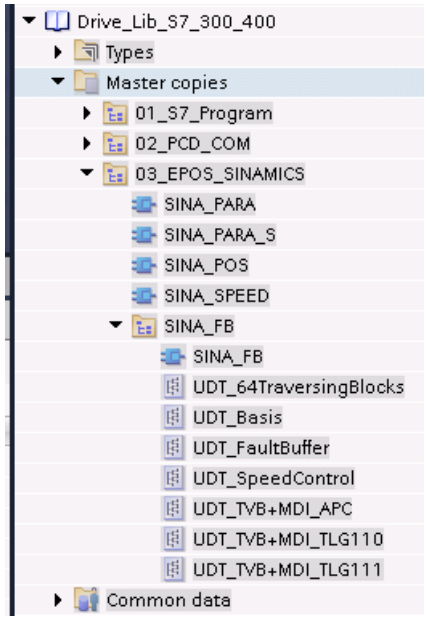
The blocks are created with standard values so that signals that are not required do not have to be interconnected by the user!

3.2.4.4 Installing the block library up to and including TIA Portal V14

No.	Action	Comment
1.	Download the library from the SIEMENS product and information pages and unzip the library to an arbitrary directory	https://support.industry.siemens.com/cs/ww/en/view/109475044 
2.	Run the DriveLib_V14.exe provided **IMPORTANT:** A library installed by Startdrive V14 is automatically overwritten with the installation	
3.	Confirm installation step 1 with "Yes"	

No.	Action	Comment
4.	Confirm installation step 2 with "Next"	
5.	Confirm installation step 3 with "Next"; select the setup language	

No.	Action	Comment
6.	Confirm installation step 4 with "Next"; confirm the license terms as well as the security notes	
7.	**Concluding the installation**	

No.	Action	Comment
8.	Content of Drive-Lib with the SINA_XXX blocks	
9.	Content of Drive-Lib with the SINA_XXX blocks	

3.2.4.5 Installing the block library up to and including TIA Portal V16

DriveLib V600 for SIMATIC STEP 7 Professional V16 (archived user library)

The procedure for using archived user libraries is described in the online help of the TIA Portal in the section "Retrieving global libraries".

3.3 Creating the communications program

The communications program can be created using the STEP 7 programming tools (LAD, FBD, STL, SCL, CFC) by calling up and parameterizing the individual functions (function blocks) or even better through the use of a block shell (call-up block for the individual functions for each drive).

In each case, a data block DRIVDBx must first be created and preassigned with the configuration data of all drives from the STEP 7 hardware configuration (see S7-1500 data blocks (Page 134)).

Creation steps when calling up and parameterizing the		**Creation steps when using a**	
S7-1200 single blocks	**S7-1500 single blocks**	**S7-1200 block shell**	**S7-1500 block shell**
Copy all blocks from the DriveLib_S71200 library to the current project (also UDTs!).	Copy all blocks from the DriveLib_S71500 library to the current project (also UDTs!).	Copy all blocks from the DriveLib_S71200 library to the current project (also UDTs!).	Copy all blocks from the DriveLib_S71500 library to the current project (also UDTs!).
Generate the data block DRIVDBx with the configuration data of the drive slave according to the hardware configuration.			
Call up the standard blocks FB36, FB39, FC60, and FB284 to 286 depending on the desired functionality in the user program (e.g. OB1) and parameterize them (parameter CFG_DATA: reference to the corresponding SLOT_UDT in DRIVDBx).	Call up the standard blocks FB31 to FB42, FC60, and FB284 to 286 depending on the desired functionality in the user program (e.g. OB1) and parameterize them (parameter CFG_DATA: reference to the corresponding SLOT_UDT in DRIVDBx).	Call up the standard blocks FB36, FB39, and FB284 to 286 needed for each drive as a multi-instance and FC60 in a block shell and parameterize them (also see the TIA Portal online help on "multi-instance").	Call up the standard blocks FB31 to FB42 and FB284 to 286 needed for each drive as a multi-instance and FC60 in a block shell and parameterize them (also see the TIA Portal online help on "multi-instance").
Connect the outboxes and inboxes in the instance data blocks to the control program.		If a standard block is needed more often (e.g. because several actual value slots are configured), this must be integrated several times into the tag declaration.	
Load the program into the CPU.		Connect the outboxes and inboxes in the multi-instance DB to the control program.	
-	-	Load the program into the CPU.	

Advantages of using a block shell

The following table shows the creation steps when using single blocks and block shells:

- All of the necessary block calls for a drive are united in a block ⇒ better structuring of the program
- The multi-instances (e.g. FB31 - FB42) called up in a function block do not need any instance DBs ⇒ saving of DBs

Note

To transfer the data configured in the hardware configuration of STEP 7, the following FB calls are required:

- FB31 per setpoint slot with its own instance
- FB32 per actual value slot with its own instance
- FB31 and FB32, 1x each for a combined setpoint/actual value slot with its own instance

3.4 Parameterization of the drives

The drives are connected to the PROFIBUS DP via the integrated PROFIBUS interface in the drives.

The parameterization of these interfaces in the devices can be found in the operating instructions for the respective module or device.

The drives are connected to PROFINET IO via PROFINET-capable CUs from the SINAMICS family of devices.

The parameterization of these CUs can be found in the operating instructions of the respective module or device.

Blocks for communication and EPOS

4

4.1 Blocks - overview

Use of the blocks

The following figure shows the various calls of the different blocks – see the sample documentation SINAMICS S120, Chapter 4/5

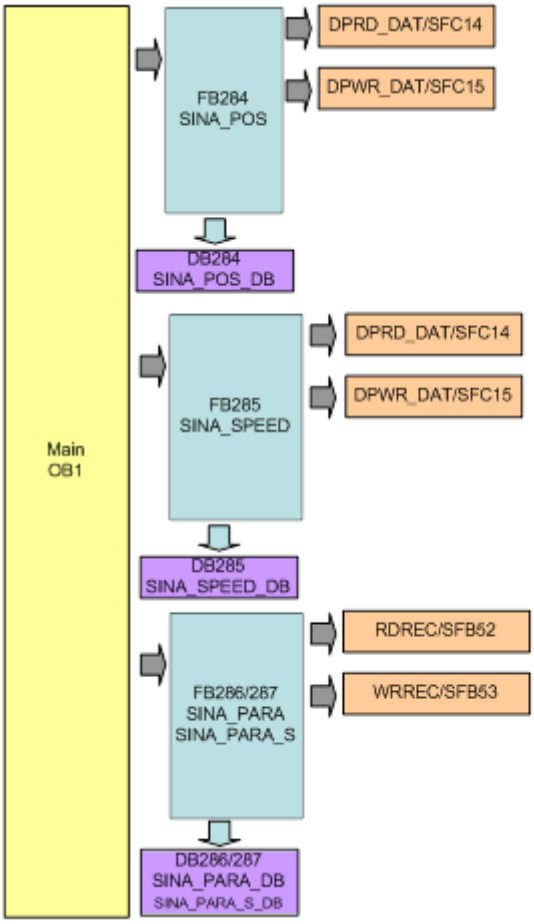


Figure 4-1 Overview of processing the blocks FB284, FB285 and FB286/FB287

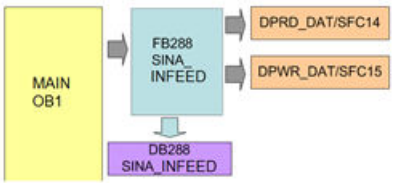


Figure 4-2 SINA_INFEED

The SIMATIC S7-1x00 program is comprised of the following sections:

1. Cyclic process data exchange – SINA_POS (FB284), SINA_SPEED (FB285), SINA_INFEED (FB288):
   In this section, the process data is sent to the SINAMICS S/G (e.g. ON command and position setpoint) or received (status and actual values).
2. Acyclic parameter access – SINA_PARA/SINA_PARA_S (FB286/287):
   In this section, the parameters of the SINAMICS S/G are accessed (e.g. write or read traversing blocks).

4.2 Memory required by the blocks

Required memory for SINA_SPEED, SINA_POS, SINA_INFEED and SINA_PARA

Block	Load memory	Work memory
SINA_SPEED	24113 bytes	959 bytes
SINA_POS	110417 bytes	6693 bytes
SINA_INFEED	20977 bytes	870 bytes
SINA_PARA	203898 bytes	16690 bytes
SINA_PARA_S	115528 bytes	6888 bytes

4.3 Solution

4.3.1 Overview of the overall solution

Schematic

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

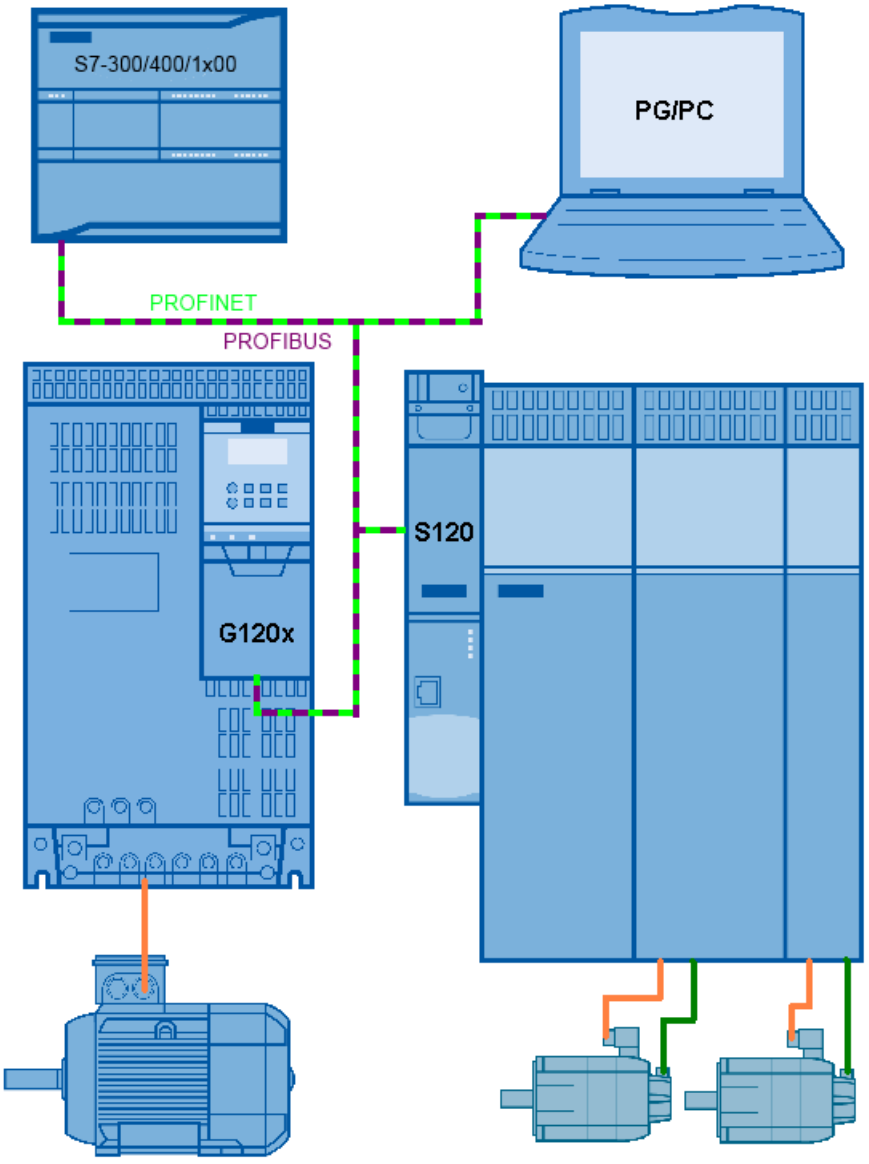


Figure 4-3 Diagram of the most important components

Structure

The configuration of the function blocks is performed in the TIA Portal.

The configuration and parameter settings for the drives is realized as follows

1. For SINAMICS G/S, using Startdrive (or using GSD and STARTER).
2. For SINAMICS V90PN using the V wizard and corresponding GSD.

Advantages

This software package offers you the following advantages:

- The SIMATIC S7-PLC can simply use the EPOS functionality
- Simpler parameter access from the SIMATIC S7-PLC
- A speed-controlled axis can be simply controlled
- Blocks can be intuitively interconnected
- Preconfigured function and data blocks
- Modular software package that can be adapted by the customer

Limitation

This block documentation does not contain a description of

- The drive commissioning/optimization
- The commissioning/selection of the PG/PC interface
- The use of technology objects by the SIMATIC S7-1200/1500

Knowledge required

Basic knowledge of the TIA Portal, SINAMICS commissioning in Startdrive (STARTER) as well as the basic positioner (EPOS) is required.

4.3.2 Description of the core functionality

Description of the core functionality

The software package is divided into 5 function blocks, which provide the various communication paths to the different technology axes on a SINAMICS drive system.

The speed-controlled and position-controlled axes are integrated by means of predefined telegrams including preconfigured instance data blocks:

1. The integration of a speed-controlled axis by means of standard telegram 1 in the SINA_SPEED function block (FB285).
2. The integration of a position-controlled axis by means of standard telegram 111 in the SINA_POS function block (FB284).

3. The integration of an infeed device (BLM / SLM / ALM – only S120) connected via DRIVE-CLiQ by means of standard telegram 370 in the SINA_INFEED function block (FB288).
4. The acyclic communication is established according to the PROFIdrive profile using data block 47, and is implemented in the SINA_PARA (FB286) or SINA_PARA_S (FB287) function block.

Function block FB284 (SINA_POS) has an input and output interface from the application view. The function block provides the available operating modes of the EPOS via a predefined interface. The main focus is on a useful limitation of the displayed variables of telegram 111, thus not all variables of the telegram are **individually** displayed at the block interface. However, at the same time, access to the entire **setpoint interface** of telegram 111 is always possible via the **input range**.

The speed block FB285 (SINA_SPEED) has an input and output interface for simple speed input / evaluation. The user must provide the function block with the rated speed (p2000) set in the SINAMICS drive. However, at the same time, access to the entire setpoint interface of telegram 1 is always possible via the input range.

The infeed block FB288 (SINA_INFEED) has an input and output interface to simply control and evaluate an infeed unit connected via DRIVE-CLiQ. Telegram 370 is used for control. However, at the same time, access to the entire setpoint interface of telegram 370 is always possible via the input range.

The acyclic communication block FB286 (SINA_PARA) provides the user with a predefined interface for simply reading and writing 16 arbitrary SINAMICS drive parameters. The user only has to specify the parameter numbers, a possible index and – for writing – a parameter value(*1). Job processing is performed autonomously after the job is started.

The acyclic communication block FB287 (SINA_PARA_S) provides the user with a predefined interface for simply reading and writing any arbitrary SINAMICS drive parameters. The user only has to specify the parameter numbers, a possible index and – for writing – a parameter value(*1). Job processing is performed autonomously after the job is started.

Note

(*1) Within the scope of the Startdrive V14 update, the SINA_PARA and SINA_PARA_S blocks are assigned an additional input and output field in the DINT format for each job field. This is realized in addition to the previous request slot into the REAL format. With the expansion, it is now possible to transfer parameters in the DINT format without rounding limitations. This is primarily necessary when reading and writing and for BICO parameters.

The external (logic) connection of the function blocks must be performed by the user. This includes, for example, the mode selection for FB284 (SINA_POS), the speed setpoint for FB285 (SINA_SPEED) as well as the filling/evaluation of the data interface of FB286 (SINA_PARA).

Sequence of the core functionality

Simplified state diagram for the EPOS mode selection – FB284 (SINA_POS)

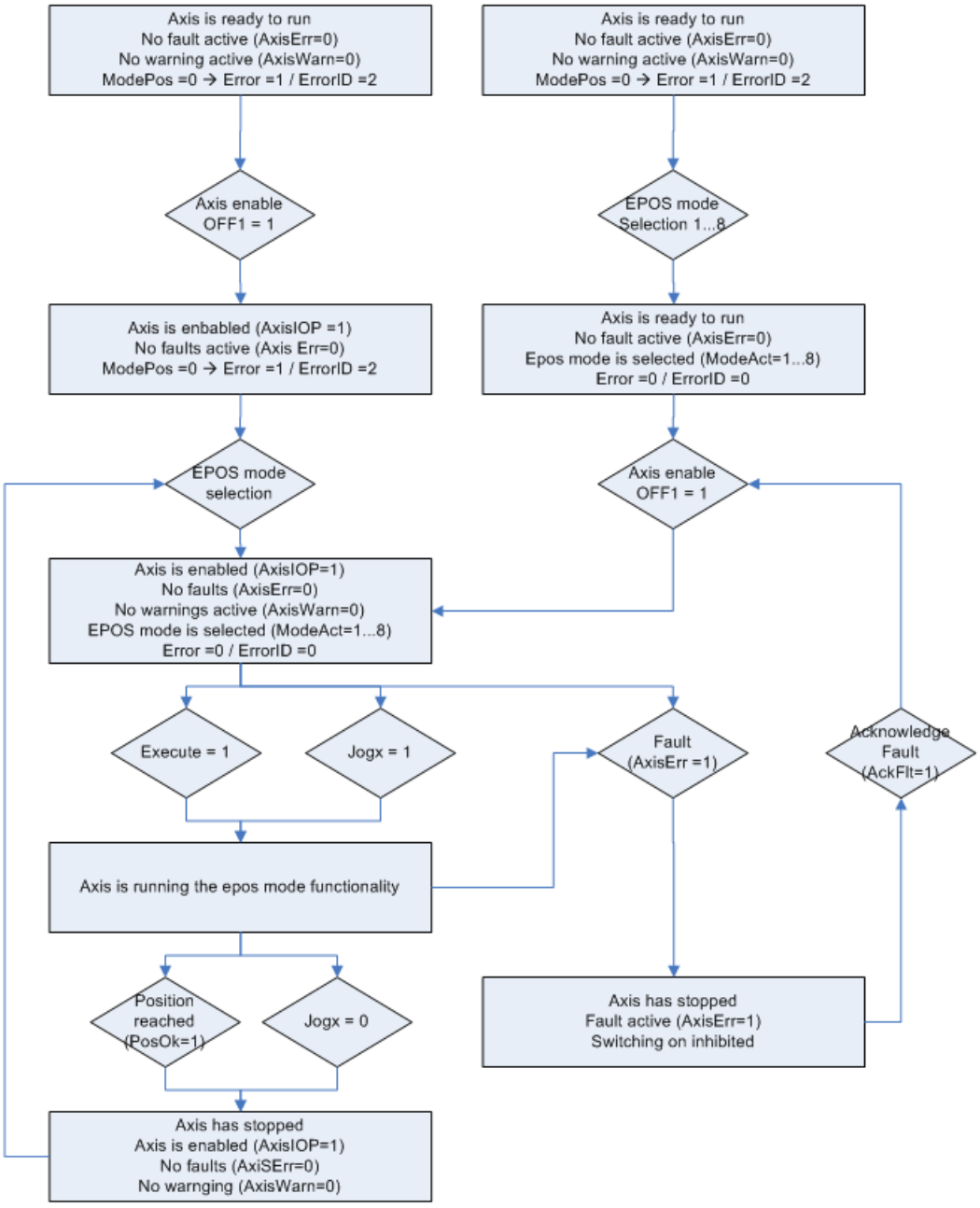


Figure 4-4 Simplified state diagram for the mode selection of the EPOS FB284 (SINA_POS)

	Action	Note
1.	Switching on the axis or selecting the EPOS operating mode	An active fault must not be present / an active alarm should not be present
2.	Start selected operating mode	• Traversing blocks, positioning and referencing use the "Execute" input • Jog mode uses Jog1 or Jog2
3.	Operating mode is performed and then terminated	End of the operating mode when the position setpoint is reached / termination through reject traversing task / deselection of the input "Jog"

General status diagram for speed block FB285 (SINA_SPEED)

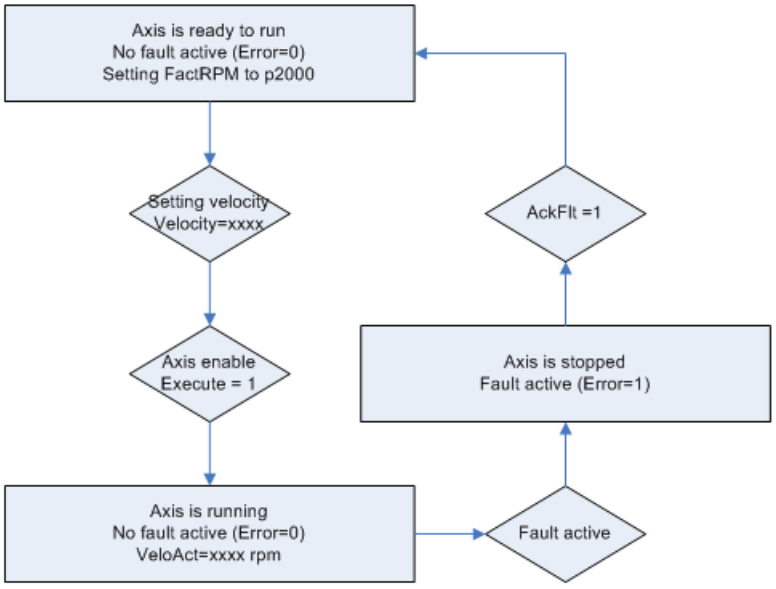


Figure 4-5 General status diagram for speed block FB285 (SINA_SPEED)

Action	Note
Entry of the scaling speed (see p2000 in the SINAMICS drive)	Specification of the real speed setpoint as block input is possible
Speed setpoint input	Input of the speed setpoint
Axis is switched on using "EnableAxis" =1	No fault active / axis is traversed

General status diagram for the acyclic block FB286/287 (SINA_PARA or SINA_PARA_S)

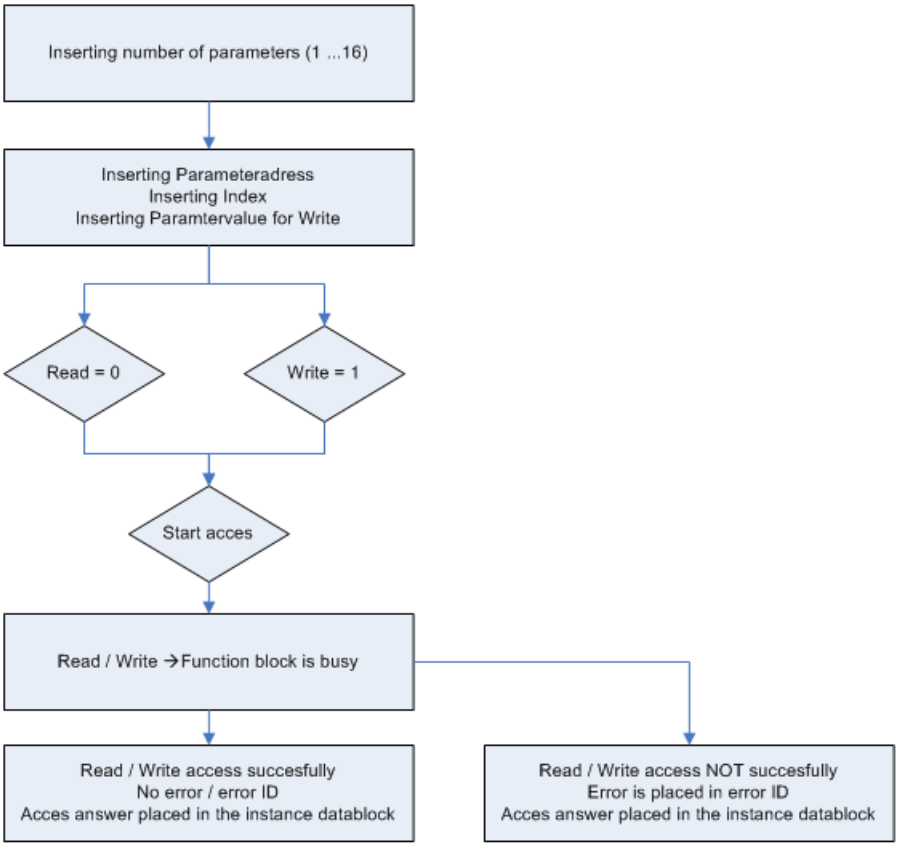


Figure 4-6 General status diagram for the acyclic block FB286/287 (SINA_PARA or SINA_PARA_S)

Action	Note
Entry of the number of parameters	1 to 16 parameters are possible
Entry of the parameter numbers, index, parameter value	Entry in the intended area of the instance data block
Read or write	Read = 0, write = 1
Start of the job	Edge from 0 → 1
Evaluation of the job response	With incorrect jobs, there is an "Error bit" and an "Error ID"

4.3.3 Minimum requirements for the hardware/software

Minimum requirements for the hardware/software

Note

- The block library can only be used as of software version **TIA Portal V12 SP1** including **STEP 7 V12 SP1**.
- The firmware of the S7-1200 MUST be **at least 2.x**.
- The firmware of the S7-1500 MUST be **at least 1.1**.

	STEP7 V12 SP1 / V13 / V14 / V15.1	
Block access	Not optimized	Optimized
SINA_POS	≤ V2.9	≥ V4.0
SINA_PARA	≤ V2.9	≥ V4.0
SINA_SPEED	≤ V2.5	≥ V4.0
SINA_PARA_S	-	≥ V4.0
SINA_INFEED	-	≥ V4.3 (STEP7 V14 or higher)

4.4 Cyclic communication

Cyclic communication

The process data is transferred cyclically, i.e. in each bus cycle. Isochronous or non-isochronous data data transmission is possible depending on the bus system used. Basically, cyclic communication is a time-critical application.

The SIMATIC S7 controller sends the control values and setpoints to the SINAMICS and receives status words and actual values from it. With regard to its use in the SINAMICS drive, the telegram format is set by means of pre-defined standard telegrams corresponding to the PROFIdrive profile or manufacturer-specific telegrams.

Depending on the type of telegram, the number of setpoints or actual values or expanded control or status words that are transferred will vary. The length of the telegram and the links in the SINAMICS drive are fixed when the machine is running and cannot be changed.

- At the SIMATIC S7 control end, the process data are provided as I/O input or output words.
- In the SINAMICS drive, the parameter assignment defines which bits of the control word are used and which data is sent to the SIMATIC S7 controller.
- A wide variety of standard functions/function blocks are available to the SIMATIC controllers for the data exchange.

Note

A detailed description of the cyclic communication can be found in the **SINAMICS S120 Function Manual Communication, 06/2019** (*/3/*).

The Manual is also available in the Siemens Industry Online Support portal: SINAMICS S120 Function Manual Communication (https://support.industry.siemens.com/cs/ww/en/view/109771803)

4.5 Acyclic communication

4.5.1 Reading and changing parameters via data set 47

Reading parameter values

Table 4-1 Request to read parameters

Data block	Byte n	Bytes n + 1	n
Header	**Reference** *01 hex ... FF hex*	**01 hex**: Read request	0
	01 hex	**Number of parameters (m)** *01 hex ... 27 hex*	2
Address, parameter 1	**Attribute** *10 hex:* Parameter value *20 hex:* Parameter description	**Number of indexes** *00 hex ... EA hex* (for parameters without index: 00 hex)	4
	Parameter number *0001 hex ... FFFF hex*		6
	Number of the 1st index *0000 hex ... FFFF hex* (for parameters without index: 0000 hex)		8
	...		...
Address, parameter 2	...		...
...	...		...
Address, parameter m	...		...

Table 4-2 Inverter response to a read request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a read request)	**01 hex**: Inverter has executed the read request. **81 hex**: Inverter was not able to completely execute the read request.	0
	01 hex	**Number of parameters (m)** (identical to the read request)	2

Data block	Byte n	Bytes n + 1	n
Values, parameter 1	**Format** 02 hex: Integer8 03 hex: Integer16 04 hex: Integer32 05 hex: Unsigned8 06 hex: Unsigned16 07 hex: Unsigned32 08 hex: FloatingPoint 10 hex OctetString 13 hex TimeDifference 41 hex: Byte 42 hex: Word 43 hex: Double word 44 hex: Error	**Number of index values** or - for a negative response - **number of error values**	4
	Value of the 1st index or - for a negative response - **error value 1** You can find the error values in a table at the end of this section.		6
	...		...
Values, parameter 2	...		
...	...		
Values, parameter m	...		

Changing parameter values

Table 4-3 Request to change parameters

Data block	Byte n	Bytes n + 1	n
Header	**Reference** *01 hex ... FF hex*	**02 hex**: Change request	0
	01 hex	**Number of parameters (m)** *01 hex ... 27 hex*	2
Address, parameter 1	**10 hex**: Parameter value	**Number of indexes** *00 hex ... EA hex* (00 hex and 01 hex have the same significance)	4
	Parameter number *0001 hex ... FFFF hex*		6
	Number of the 1st index *0001 hex ... FFFF hex*		8
	...		...
Address, parameter 2	...		
...	...		...
Address, parameter m	...		

Data block	Byte n	Bytes n + 1	n
Values, parameter 1	**Format** *02 hex:* Integer 8 *03 hex:* Integer 16 *04 hex:* Integer 32 *05 hex:* Unsigned 8 *06 hex:* Unsigned 16 *07 hex:* Unsigned 32 *08 hex:* Floating Point *10 hex* Octet String *13 hex* Time Difference *41 hex:* Byte *42 hex:* Word *43 hex:* Double word	**Number of index values** *00 hex ... EA hex*	
	Value of the 1st index		
	...		
Values, parameter 2	...		
...	...		
Values, parameter m	...		

Table 4-4 Response, if the inverter has executed the change request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a change request)	**02 hex**	0
	01 hex	**Number of parameters** (identical to a change request)	2

Table 4-5 Response if the inverter was not able to completely execute the change request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a change request)	**82 hex**	0
	01 hex	**Number of parameters** (identical to a change request)	2
Values, parameter 1	**Format** 40 hex: Zero (change request for this data block executed) 44 hex: Error (change request for this data block not executed)	**Number of error values** 00 hex or 02 hex	4
	Only for "Error" - **error value 1** You can find the error values in the table at the end of this section.		6
	Only for "Error" - **error value 2** Error value 2 is either zero, or it contains the number of the first index where the error occurred.		8
Values, parameter 2	...		
...	...		...
Values, parameter m	...		

Table 4-6 Error value in the parameter response

Error value 1	Meaning
00 hex	**Illegal parameter number** (access to a parameter that does not exist)
01 hex	**Parameter value cannot be changed** (change request for a parameter value that cannot be changed)
02 hex	**Lower or upper value limit exceeded** (change request with a value outside the value limits)
03 hex	**Incorrect subindex** (access to a parameter index that does not exist)
04 hex	**No array** (access with a subindex to non-indexed parameters)
05 hex	**Incorrect data type** (change request with a value that does not match the data type of the parameter)
06 hex	**Setting not permitted, only resetting** (change request with a value not equal to 0 without permission)
07 hex	**Descriptive element cannot be changed** (change request to a descriptive element that cannot be changed)
09 hex	**Description data not available** (access to a description that does not exist, parameter value is available)
0B hex	**No master control** (change request but with no master control)
0F hex	**Text array does not exist** (although the parameter value is available, the request is made to a text array that does not exist)
11 hex	**Request cannot be executed due to the operating state** (access is not possible for temporary reasons that are not specified)
14 hex	**Inadmissible value** (change request with a value that is within the limits but which is illegal for other permanent reasons, i.e. a parameter with defined individual values)
15 hex	**Response too long** (the length of the actual response exceeds the maximum transfer length)
16 hex	**Illegal parameter address** *(illegal or unsupported value for attribute, number of elements, parameter number, subindex or a combination of these)*
17 hex	**Illegal format** (change request for an illegal or unsupported format)
18 hex	**Number of values not consistent** *(number of values of the parameter data to not match the number of elements in the parameter address)*
19 hex	**Drive object does not exist** (access to a drive object that does not exist)
6B hex	**No change access for a controller that is enabled**.
6C hex	**Unknown unit**.
6E hex	**Change request is only possible when the motor is being commissioned (p0010 = 3).**
6F hex	**Change request is only possible when the power unit is being commissioned (p0010 = 2).**
70 hex	**Change request is only possible for quick commissioning (basic commissioning) (p0010 = 1).**
71 hex	**Change request is only possible if the inverter is ready (p0010 = 0).**
72 hex	**Change request is only possible for a parameter reset (restore to factory setting) (p0010 = 30).**
73 hex	**Change request possible only during commissioning of the safety functions (p0010 = 95).**
74 hex	**Change request is only possible when a technological application/unit is being commissioned (p0010 = 5).**
75 hex	**Change request is only possible in a commissioning state (p0010 ≠ 0).**
76 hex	**Change request is not possible for internal reasons (p0010 = 29).**
77 hex	**Change request is not possible at download.**
81 hex	**Change request is not possible at download.**
82 hex	**Transfer of the control authority (master) is inhibited by BI: p0806.**
83 hex	**Desired interconnection is not possible** (the connector output does not supply a float value although the connector input requires a float value)
84 hex	**Inverter does not accept a change request** (Inverter is busy with internal calculations. See parameter r3996 in the List Manual for the converter)
85 hex	**No access methods defined**.

Error value 1	Meaning
86 hex	**Write access only during commissioning of the data records (p0010 = 15)** (operating status of the inverter prevents a parameter change.)
87 hex	**Know-how protection active, access locked**
C8 hex	**Change request below the currently valid limit** (change request to a value that lies within the "absolute" limits, but is however below the currently valid lower limit)
C9 hex	**Change request above the currently valid limit** (example: a parameter value is too large for the inverter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

4.5.2 Acyclic communication

Data record 47

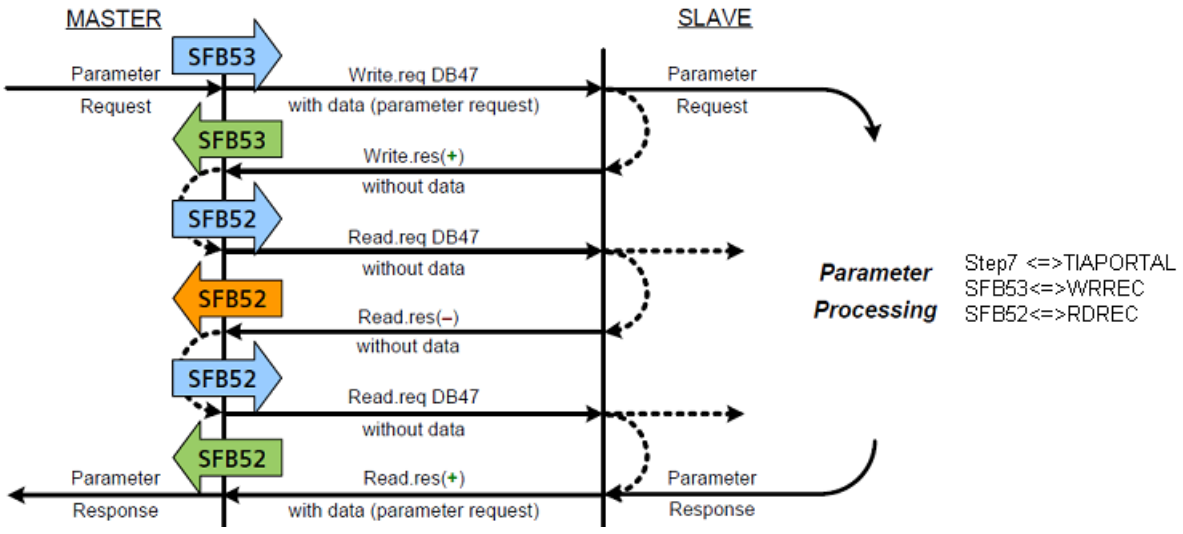


Figure 4-7 Acyclic communication

It is possible to transfer the parameter range acyclically as needed without having to create a permanent communication load.

The acyclic transfer of process data takes considerably longer than the cyclic transfer, but larger quantities of data can be transferred.

- Write and read jobs are triggered via the standard function blocks SFB52/53 in the SIMATIC controller.
- A read job generally begins with a write job, which gives the values to be read to the addressed device. Then the actual read job is executed.
- No special action is required at the SINAMICS drive end.

A decisive factor for functional acyclic communication is the creation of a job profile corresponding to the data record used.

Also, the answer from write and read jobs must be transferred to corresponding data block structures and evaluated by the user.

For write and read jobs that remain constant, the structure can be defined in advance. If, however, different jobs with different contents are involved, this can only be shown in a general structure and must be evaluated separately by the user.

Note

A detailed description of the acyclic communication can be found in the **SINAMICS S120 Function Manual Communication, 06/2019**. (/3/)

The Manual is also available in the Siemens Industry Online Support portal: SINAMICS S120 Function Manual Communication (https://support.industry.siemens.com/cs/ww/en/view/109771803)

Further information on data set 47 can be found in the PROFIdrive Manual, Edition 2006.

4.6 Operating mode selection for EPOS

4.6.1 Basic principles of the basic positioner

Fundamentals

The basic positioner (EPOS) is a very comprehensive and powerful function module for the position-controlled traversing of electrical drives.

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).

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

Furthermore, the parameterization software Startdrive contains convenient configuration, commissioning and diagnostic functions for the EPOS functionality.

When the basic positioner is activated, the position controller is also activated. This is automatically performed via the drive wizard. In addition, the necessary "internal interconnections" (BICO technology), which are needed between the EPOS and the position controller, are automatically established (e.g. setpoints from the EPOS to the closed-loop position control, axis cycle offset, etc.).

The closed-loop position control essentially consists of the following parts:

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

The following additional functions can be implemented using the basic positioner:

Mechanical system:

- Backlash compensation
- Modulo offset
- Position tracking/limitations
- Velocity/acceleration/delay limitations
- Software limit switch (traversing range limitation by means of position setpoint evaluation)
- Stop output cams (traversing range limitation by means of hardware limit switch evaluation)
- Position/standstill monitoring
- Following error monitoring
- Two cam switching signals

Note

A detailed description can be found in the Function Manual Basic Positioner, 04/2018, FW V4.7 SP10, A5E34257659A AF.

4.6.2 Operating mode selection of the basic positioner

Operating mode selection of the basic positioner

The following extract from the List Manual graphically illustrates the operating mode selection of the basic positioner (EPOS):

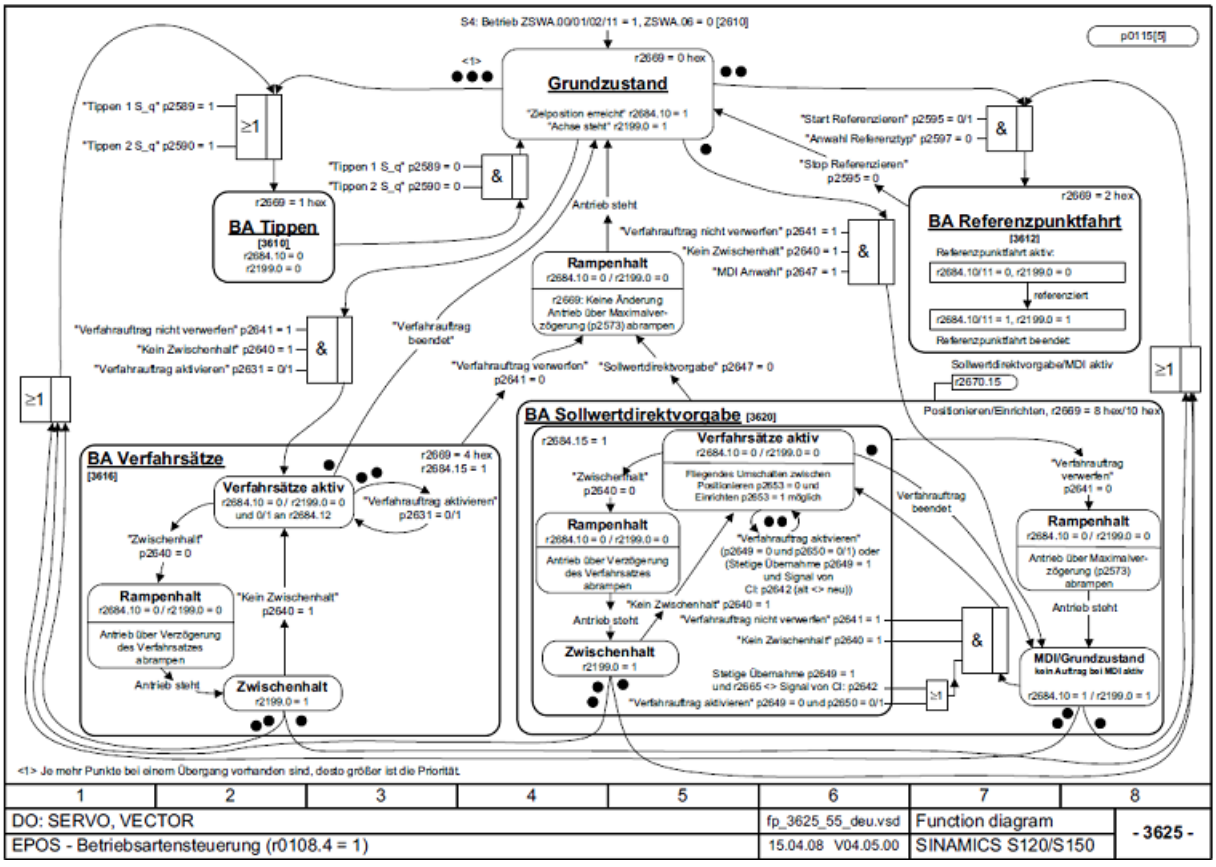


Figure 4-8 Operating mode selection of the basic positioner

The operating mode selection is decisive for carrying out the desired functions. The EPOS operating modes are structured hierarchically and the following sequence applies with simultaneous selection of the functions:

Jog >> Homing procedure >> MDI setpoint specification >> Trav. blocks

Description of the blocks

5

5.1 Area of application of the blocks

5.1.1 Overview

The following blocks can be used for PROFIBUS DP as well as for PROFINET IO.

5.1.2 Which block for which device

The following table shows the available blocks:

FB31	PCD_SEND
FB32	PCD_RECV
FB36	PDAT_AC2
FB39	DEV_FLT4
FB42	PDAT_UD2
FC60	COM_STAT
FB284	SINA_POS
FB285	SINA_SPEED
FB286	SINA_PARA
FB287	SINA_PARA_S
FB288	SINA_INFEED

The following table shows the possible uses of the blocks depending on the CPU type:

Usable blocks / **CPU types**	**FB 31**	**FB 32**	**FB 36**	**FB 39**	**FB 42**	**FC 60**	**FB284**	**FB285**	**FB286**	**FB287**	**FB288**
S7-1200	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
S7-1500	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

The following table shows the possible uses of the blocks depending on the device type:

Usable blocks / **Device types**	**FB 31**	**FB 32**	**FB 36**	**FB 39**	**FB 42**	**FC 60**	**FB284**	**FB285**	**FB286**	**FB287**	**FB288**
SINAMICS G/S	Yes	Yes	**Yes**	Yes	**Yes**	Yes	**Yes**	**Yes**	**Yes**	Yes	
only SINAMICS S120											Yes

Yes (bold) - recommended combination

5.1.3 Which block for which application

The following table shows the applications for which the individual blocks can be used:

Block	Typical application
FB31 PCD_SEND	Send process data (control words and setpoints) to the drive
FB32 PCD_RECV	Receive process data (status words and actual values) from the drive
FB36 PDAT_AC2	Read or write parameters via the 244-byte-wide acyclic telegrams and using the DS47 data set standardized in the PROFIdrive profile V4.1. This function should particularly be used when • An individual, indexed parameter is to be exchanged with several indices. • No management system is required for the automatic processing of several parameters.
FB39 DEV_FLT4	This block is specially programmed for reading the complete diagnostic buffer of a SINAMICS drive. This block reads the fault number and the fault value from the drive.
FB42 PDAT_UD2	This block processes an automatic DOWNLOAD or UPLOAD (updating) of one or even several data blocks using the standardized data set DS47 that is standardized in the PROFIdrive Profile V4.1. This function should be used when • Several parameters, up to an entire parameter set, are to be automatically transferred to the drive. – For re-parameterizing a drive after a device is replaced – For re-parameterizing a drive due to a product, recipe or batch change • The parameter values that are saved in a DB are to be updated with the most current values from the drive. – After parameter changes made locally at the drive – For supplying additional display values going beyond the available PZD data to HMI systems
FC60 COM_STAT	With the aid of the system function SFC 51 RDSYSST, the block evaluates the system status list of the CPU and reports whether the slave to be processed is faulted or disabled.
FB284 SINA_POS	This block cyclically activates drives of the type SINAMICS S/G. This allows you to control the basic positioning and closed-loop position control functionality of drives that support these functions.
FB285 SINA_SPEED	This block cyclically activates drives of the type SINAMICS S/G in order to use closed-loop speed control.
FB286 SINA_PARA	Using this block, up to 16 parameters can be acyclically written or read for a SINAMICS S/G.
FB287 SINA_PARA_S	Using this block, an individual parameter can be acyclically written or read for a SINAMICS S/G.
FB288 SINA_INFEED	With this block you control the infeed unit for a SINAMICS S120.

5.2 Function blocks (S7-1200/1500)

5.2.1 FB PCD_SEND: Write process data - S7-1500

Description

FB 31

Can be used in the following CPUs: S7-1500

Calling OBs

The block can be alternatively installed in the following OBs:

Cyclic task: OB1

Cyclic interrupt OB: e.g. OB32

Description of functions

The block cyclically transfers the process data (control words, setpoints), taking the consistency conditions from the SIMATIC to the drive into consideration.

Exactly one call-up must take place for a combined setpoint/actual value slot.

For multi-axis drives, it is also true that the FB must be called up once for each axis and setpoint slot with its own instance.

PZD setpoint interface

The length of the PZD interface can be freely specified up to a total length of 16 words. The interface is parameterized in HW-Config. Here you can also directly select the standard telegrams PROFIdrive profile Drive Technology or the PPO types 1 to 5.

The first word in the setpoint range (PCD_1) must always be filled with the control word.

I/O bar

The following table shows the input and output parameters:

Parameter	Declaration	Data type	Memory area	Description
Input				
CFG_DATA	INPUT	SLOT_UDT (application-specific)	D, L	Slot-specific configuration data (structure, seeS7-1500 data blocks (Page 134))
PCD_1	INPUT	WORD	E, A, M, D, L, const.	Control word
PCD_2	INPUT	WORD	E, A, M, D, L, const.	Main setpoint

PCD_3	INPUT	WORD	E, A, M, D, L, const.	Setpoint/additional control word
...				
PCD_16	INPUT	WORD	E, A, M, D, L, const.	Setpoint/additional control word
Output				
SFC_ERR	OUTPUT	BOOL	E, A, M, D, L	SFC 15 DPWR_DAT reports fault
CFG_ERR	OUTPUT	BOOL	E, A, M, D, L	Slave or slot is not configured or incorrectly configured

Data area

The following table describes the return value:

Parameter	Declaration	Data type	Description
SFC_RET_VAL	STAT	INT	Return value of the SFC15 DPWR_DAT

Error reactions

The following table describes the error displays:

Output	Error description
CFG_ERR	A configuration error is signaled if no or faulty configuration data is entered. The following data is checked: • Slot type not equal to setpoint or combined setpoint/actual value • (i.e. parameter SLOT_ID <> 1 or 3) • Data length = 0 • Process data address = 0 or > (16 + 1 - data length) or (32 + 1 - data length)
SFC_ERR	SFC error during data transfer with the system function SFC15 DPWR_DAT (return value < 0) The return value is stored in the instance DB (SFC_RET_VAL).

Note

If the PCD_SEND for a setpoint slot with pure data exchange broadcast data is called, the block reports CFG_ERR.

Block call (STL source code)

```
CALL   PCD_SEND, DB_PCD_SEND(
     CFG_DATA   := DRIVDB1.SLAVE_1.SLOT_6,
     PCD_1   := MW0,
     PCD_2   := MW2,
     PCD_3   := MW4,
     PCD_4   := MW6,
     PCD_5   := MW8,
     PCD_6   := MW10,
     PCD_7   := MW12,
     PCD_8   := MW14,
     PCD_9   := MW16,
     PCD_10   := MW18,
     PCD_12   := MW20,
     PCD_13   := MW22,
     PCD_14   := MW24,
     PCD_15   := MW26,
     PCD_16   := MW28,
     SFC_ERR   := M30.0,
     CFG_ERR   := M30.1)
```

5.2.2 FB PCD_RECV: Read process data - S7-1500

Description

FB32

Can be used in the following CPUs: S7-1500

Calling OBs

The block can be alternatively installed in the following OBs:

Cyclic task: OB1

Cyclic interrupt OB: e.g. OB32

Description of functions

The block cyclically receives the process data (status words, actual values), taking the consistency conditions of the drive into consideration.

Exactly one call-up must take place for a combined setpoint/actual value slot.

For multi-axis drives, it is also true that the FB must be called up once for each axis and actual value slot with its own instance.

PZD actual value interface

The length of the PZD interface can be freely specified up to a total length of 16 words. The interface is parameterized in HW-Config. Here you can also directly select the standard telegrams in accordance with the PROFIdrive profile Drive Technology or the PPO types 1 to 5.

For the block to function correctly, it is imperative that the first word in the actual value range (PCD_1) be filled with the status word 1 of the drive (e.g. MASTERDRIVES P734.1 = 32).

I/O bar

The following table shows the input and output parameters:

Parameter	Declaration	Data type	Memory area	Description
Input				
CFG_DATA	INPUT	SLOT_UDT (application-specific)	D, L	Slot-specific configuration data (structure, seeS7-1500 data blocks (Page 134))
Output				
PCD_1	OUTPUT	WORD	E, A, M, D, L	Status word
PCD_2	OUTPUT	WORD	E, A, M, D, L	Main actual value
PCD_3	OUTPUT	WORD	E, A, M, D, L	Actual value/added status word
...				
PCD_32	OUTPUT	WORD	E, A, M, D, L	Actual value/added status word
PLC_CTRL	OUTPUT	BOOL	E, A, M, D, L	PLC control is requested by the slave
DEV_FLT	OUTPUT	BOOL	E, A, M, D, L	Drive signals device malfunction
DEV_WAR	OUTPUT	BOOL	E, A, M, D, L	Drive signals device warning
SFC_ERR	OUTPUT	BOOL	E, A, M, D, L	SFC 14 DPRD_DAT reports error
CFG_ERR	OUTPUT	BOOL	E, A, M, D, L	Slave or slot is not configured or incorrectly configured

Data area

The following table describes the return value:

Parameter	Declaration	Data type	Description
SFC_RET_VAL	STAT	INT	Return value of the SFC14 DPRD_DAT

Error reactions

The following table describes the error displays:

Output	Error description
CFG_ERR	A configuration error is signaled if no or faulty configuration data is entered. The following data is checked: • Slot type not equal to actual value or combined setpoint/actual value (i.e. parameter SLOT_ID <-> 2 or 3) • Data length = 0 • Process data address = 0 or > (16 + 1 - data length) or (32 + 1 - data length)
DEV_FLT	DEV_FLT is set in the status word on the basis of the drive fault bit. This bit can be used as an initiator for the DEV_FLT function block in order to read the fault buffer of the drive.
DEV_WAR	DEV_WAR is set in the status word on the basis of the drive warning bit.
SFC_ERR	SFC error during data transfer with the system function SFC DPRD_DAT (return value < 0) The return value is stored in the instance DB (SFC_RET_VAL). The inbox is emptied.

Block call (STL source code)

```
CALL   PCD_RECV, DB_PCD_RECV(
          CFG_DATA   := DRIVDB1.SLAVE_1.SLOT_5,
          PCD_1   := MW0,
          PCD_2   := MW2,
          PCD_3   := MW4,
          PCD_4   := MW6,
          PCD_5   := MW8,
          PCD_6   := MW10,
          PCD_7   := MW12,
          PCD_8   := MW14,
          PCD_9   := MW16,
          PCD_10   := MW18,
          PCD_12   := MW20,
          PCD_13   := MW22,
          PCD_14   := MW24,
          PCD_15   := MW26,
          PCD_16   := MW28,
          PLC_CTRL   := M30.0,
          DEV_FLT   := M30.1,
          DEV_WAR   := M30.2,
          SFC_ERR   := M30.3,
          CFG_ERR   := M30.4)
```

5.2.3 FB PDAT_AC2: Edit parameters acyclically (DS47) - S7-1200/1500

Description

FB36

Can be used in the following CPUs: S7-1200, S7-1500

For information on parameterization options, see Parameter model (Page 179).

Calling OBs

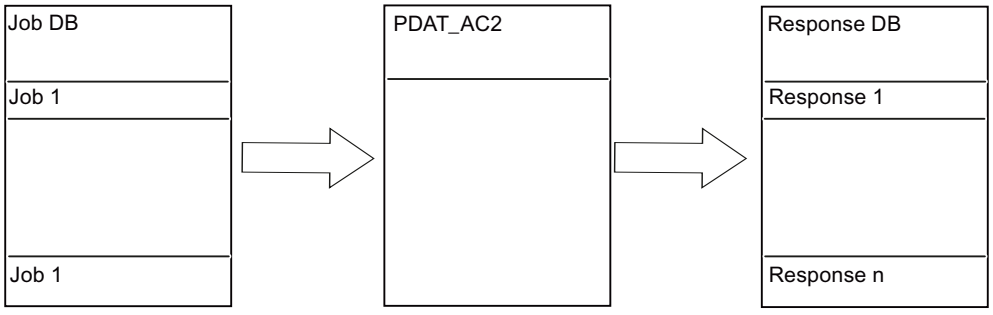
Cyclic task: OB1

Cyclic interrupt OB: e.g. OB32

Description of functions

The block coordinates the acyclic communication between the drive and the S7-CPU for the transfer of parameter data according to PROFIdrive-Profile Drive Technology, version 4.1 The data is exchanged via the S7 communication services "Read/write data set" (data set number 47).

The number of the data set used can be specified via the input DS_NO. If the input is supplied with the value 0, data set 47 is used. For values <> 0, the value that is at the input is viewed as the number of the data set to be used. However, the telegram must continue to be structured as per PROFIdrive-Profile Drive Technology, Version 4.1, because otherwise the block will report an error. This means that any data set which is supported by the device can be used as long as the contents of the telegram correspond to PROFIdrive-Profile Drive Technology, Version 4.1. This function is mainly relevant for drives with PROFINET interface.



Parameter requests are stored in a job data block. Each job consists at least of one "request header" and one "parameter address". A write job also contains a format specification, the number of the values to be written and a value field. A job always has a fixed length of 240 bytes. When the program is initiated, the block PDAT_AC2 transfers an individual job to the drive or, for a multi-axle drive, to an axis of this drive. The block then receives the response data of the drive or the axis of this drive and saves it in a response data block. Each response consists of the "Response header" and a value field. The response data is also always 240 bytes long.

No coordination of the jobs that are requested by various applications is carried out. This must be implemented within the applications.

Example of the structure of the job DB

Name	Data type	Offset	Start value	Retain	Visible in ...	Comment
Static						
REQUEST_1	Struct	0.0		☑	☑	Change parameter value, single
REQUEST_HEADER	"HEADER"	0.0		☑	☑	
PARAMETER_ADDRESS	"PARAMETERADDR"	4.0		☑	☑	
FORMAT	Byte	10.0	B#16#42	☑	☑	Format: WORD
NO_VALUES	Byte	11.0	B#16#1	☑	☑	
VALUES	Array [0 .. 113] of W...	12.0		☑	☑	
REQUEST_2	Struct	240.0		☑	☑	Request parameter value, single
REQUEST_HEADER	"HEADER"	0.0		☑	☑	
PARAMETER_ADDRESS	"PARAMETERADDR"	4.0		☑	☑	
VALUES	Array [0 .. 114] of W...	10.0		☑	☑	
REQUEST_3	Struct	480.0		☑	☑	Change parameter value, multi-parameter
REQUEST_HEADER	"HEADER"	0.0		☑	☑	
PARAMETER_ADDRESS_1	"PARAMETERADDR"	4.0		☑	☑	
PARAMETER_ADDRESS_2	"PARAMETERADDR"	10.0		☑	☑	
FORMAT1	Byte	16.0	B#16#42	☑	☑	Format: WORD
NO_VALUES1	Byte	17.0	B#16#1	☑	☑	singel (array) element
VALUES1	Array [0 .. 0] of Word	18.0		☑	☑	
FORMAT2	Byte	20.0	B#16#42	☑	☑	Format: WORD
NO_VALUES2	Byte	21.0	B#16#5	☑	☑	serveral array elements
VALUES2	Array [0 .. 108] of W...	22.0		☑	☑	
REQUEST_4	Struct	720.0		☑	☑	Request parameter value, smulti-parameter
REQUEST_HEADER	"HEADER"	0.0		☑	☑	
PARAMETER_ADDRESS_1	"PARAMETERADDR"	4.0		☑	☑	
PARAMETER_ADDRESS_2	"PARAMETERADDR"	10.0		☑	☑	
VALUES	Array [0 .. 111] of W...	16.0		☑	☑	

Figure 5-1 FBPDAT AC2 job list

The header and parameter address are best structured as UDT data types. The numbers of the UDT are freely selectable.

Structure of the header UDT (REQUEST_1)

Name	Data type	Offset	Start value	Retain ...	Visible in HMI	Comment
Static						
REQ_REF	Byte	0.0	B#16#1	☑	☑	Request reference
REQ_ID	Byte	1.0	B#16#2	☑	☑	Reqiest ID
AXIS	Byte	2.0	B#16#0	☑	☑	Axis addressing
NO_PARAM	Byte	3.0	B#16#1	☑	☑	Number of parameter

Figure 5-2 FBPDAT AC2 header

Structure of the parameter address UDT (REQUEST_1)

Name	Data type	Offset	Start value	Retain	Visible in ..	Comment
Static						
ATTRIBUTE	Byte	...	B#16#10	☑	☑	Attribute
NO_ELEM	Byte	...	B#16#0	☑	☑	Number of elements
PARA_NO	Word	...	B#16#3	☑	☑	Parameter number
SUBINDEX	Word	...	B#16#0	☑	☑	Subindex

Figure 5-3 FBPDAT AC2 parameters

Example of the structure of the response DB

Name	Data type	Offset	Start value	Retain	Visible in ..	Comment
Static						
RESPONSE_1	Struct	0.0		☑	☑	
RESPONSE_HEADER	"HEADER"	0.0		☑	☑	
VALUE	Array [0..117] of Wo...	4.0		☑	☑	
RESPONSE_2	Struct	240.0		☑	☑	
RESPONSE_HEADER	"HEADER"	0.0		☑	☑	
VALUE	Array [0..117] of Wo...	4.0		☑	☑	
RESPONSE_3	Struct	480.0		☑	☑	
RESPONSE_HEADER	"HEADER"	0.0		☑	☑	
VALUE	Array [0..117] of Wo...	4.0		☑	☑	
RESPONSE_4	Struct	720.0		☑	☑	
RESPONSE_HEADER	"HEADER"	0.0		☑	☑	
VALUE	Array [0..117] of Wo...	4.0		☑	☑	

Figure 5-4 FBPDAT AC2 response

At the input DB_NO_OR of the PDAT_AC2, the number of the job DB is specified. At the input OFFSET_OR, the start address of the job to be transferred is specified. At the input DB_NO_AN, the number of the response DB is specified. At the input OFFSET_AN, the address from which the response of the drive is to be saved is specified. A new job is accepted by means of a positive edge at the START input of the block. The job is then executed exactly once. The START bit must be reset with the BUSY signal. Then, the pointer to the next job can be written to the send interface. This job can only be started with the checkback signal DONE or xyz_ERR.

No coordination of the jobs that are requested by various applications is carried out. This must be implemented within the applications (see Interlocking of blocks with acyclic communication (Page 132)).

For multi-axis drives, addressing of the axis is implemented via the "Axis" byte in the parameter job header.

The block supports the following jobs:

- Read/write parameter value, simple (word/double word)
- Read/write parameter value, one or more array elements (max. 234)
- Read/write parameter value, multiparameter
- Read parameter description, complete or a single description element
- Read parameter texts, individually or several text elements
- Read parameter, multiparameter with different attributes (value, description, text).

Note

The detailed structure of parameter jobs and the associated responses is described under Formulating parameter jobs (data set 47) (Page 146).

I/O bar

The following table shows the input and output parameters:

Parameter	Declaration	Data type	Memory area	Description
Input				
LADDR	INPUT	HW_IO	E, A, M, D, L, const.	Diagnostics address of the slave
START	INPUT	BOOL	E, A, M, D, L	Accept start pulse for job
DS_NO	INPUT	WORD	E, A, M, D, L, const.	Number of data set be read out
DB_NO_OR	INPUT	DINT	E, A, M, D, L, const.	DB no. of the job DB
OFFSET_OR	INPUT	INT	E, A, M, D, L, const.	Start address of the job in the job DB
DB_NO_AN	INPUT	DINT	E, A, M, D, L, const.	DB no. of the response DB
OFFSET_AN	INPUT	INT	E, A, M, D, L, const.	Start address of the area from which the response data can be saved
Output				
BUSY	OUTPUT	BOOL	E, A, M, D, L	Job running
DONE	OUTPUT	BOOL	E, A, M, D, L	Job complete without error
REQ_ERR	OUTPUT	BOOL	E, A, M, D, L	Job was completed with error(s) (response contains error ID)
WDOG_ERR	OUTPUT	BOOL	E, A, M, D, L	Watchdog error, no plausible response data available
SFB_ERR	OUTPUT	BOOL	E, A, M, D, L	SFB 53 WR REC / SFB 52 RDREC reports error
CFG_ERR	OUTPUT	BOOL	E, A, M, D, L	Slave is not or incorrectly configured
DB_ACT	OUTPUT	INT	E, A, M, D, L	Currently processed job DB
DB_ERR	OUTPUT	BOOL	E, A, M, D, L	Job or response DB faulty
DB_ERRNO	OUTPUT	INT	E, A, M, D, L	Error code to DB_ERR
OFFSET_ACT	OUTPUT	INT	E, A, M, D, L	Offset of the job in the job DB

The message bits are valid until the next job is received. The outputs are deleted when the new data is received from the drive and the new data is transferred to the outputs.

Data area

The following table describes the data area:

Parameter	Declaration	Data type	Description
si_NODATA_CYCLE_NO	STAT	INT	The number of cycles that are waited for receiving response data before the job is retried (preassignment = 2500)
si_NODATA_RETRY_NO	STAT	INT	Number of times a job will be retried if no response data has been received (preassignment = 5)
si_SFB52_RET_VAL	STAT	INT	Return value of the SFB 52 RDREC

si_SFB53_RET_VAL	STAT	INT	Return value of the SFB 53 WRREC
si_WDOG_RETRY_NO	STAT	INT	Number of times a job will be retried if no plausible response data has been received (preassignment = 5)

Error reactions

The following table describes the error displays:

Output	Error description
CFG_ERR	A configuration error is signaled if no or faulty configuration data is entered. The following data is checked: • Missing parameterization priority for the drive
DB_ERR	Job or response DB faulty
DB_ERRNO	Error code to DB_ERR 0 = No error has occurred 1 = Job DB does not exist or incorrect length 2 = Response DB does not exist or incorrect length 3 = Number of parameters < 1 or > 39 4 = Number of elements < 0 or > 234
REQ_ERR	Response contains error ID The response also contains a more detailed description of the error (see Formulating parameter jobs (data set 47) (Page 146)).
SFB_ERR	SFB error during data transfer with the system functions SFB RDREC / SFB WRREC (return value < 0) The return value is stored in the instance DB (SFB53_RET_VAL or SFB52_RET_VAL). To learn the meaning of the return value, see the online help for the SFB or see Formulating parameter jobs (data set 47) (Page 146). The inbox is emptied when there is an error reading the data.
WDOG_ERR	No plausible response data within the monitoring time

Note

Display of a communication fault

- In the event of a failed communication connection and initiated job, it is possible that the CPU may not immediately signal SFB_ERR. Instead, the block signals BUSY until the error is either cleared and the job is ended or the cycle monitoring (default setting: NODATA_CYCLE_NO x NODATA_RETRY_NO = 12500 cycles!) activates and SFB_ERR is reported. After the SFB_ERR is cleared, the user must re-initiate his job.
- The immediate display of a communication fault is provided by the FC COM_STAT (FC60) in any case.

Block call (STL source code)

```
Network 1: Check starting conditions of PDAT_AC2

      U   "DI_PDAT_AC2".SFB_ERR //Retrigger of START in case of SFB error
          UN   "DI_PDAT_AC2".START
          =   "DI_PDAT_AC2".START
          U   "DI_PDAT_AC2".SFB_ERR
          SPB   PDAT

          UN   "DI_PDAT_AC2".BUSY // Starting condition after complete
          UN   "DI_PDAT_AC2".DONE // download of the program
          UN   "DI_PDAT_AC2".SFB_ERR
          UN   "DI_PDAT_AC2".WDOG_ERR
          UN   "DI_PDAT_AC2".CFG_ERR
          UN   "DI_PDAT_AC2".REQ_ERR
          S   "DI_PDAT_AC2".START

          U   "DI_PDAT_AC2".BUSY  // Reset of START
          R   "DI_PDAT_AC2".START

          U   "DI_PDAT_AC2".DONE // Start new request
          S   "DI_PDAT_AC2".START

          U   "DI_PDAT_AC2".REQ_ERR
                    // Start new request or error routine
          S   "DI_PDAT_AC2".START

          U   "DI_PDAT_AC2".WDOG_ERR
                    // Start new request or error routine
          S   "DI_PDAT_AC2".START

          U   "DI_PDAT_AC2".CFG_ERR
                    // Start new request or error routine
          S   "DI_PDAT_AC2".START

Network 2: assembling acyclic requests

   UN   "DI_PDAT_AC2".START
   SPB   PDAT  // assemble acyclic requests

Network 3: Call for PDAT_AC2

PDAT: CALL   PDAT_AC2, DI_PDAT_AC2(
          LADDR   :=   "DRIVDB1".SLAVE_5.DADDR,
          START   :=   ,
          DS_NO   :=   MW16,
          DB_NO_OR   :=   MW0,
          OFFSET_OR   :=   MW2,
          DB_NO_AN   :=   MW4,
          OFFSET_AN   :=   MW6,
          BUSY   :=   ,
          DONE   :=   ,
          REQ_ERR   :=   ,
          WDOG_ERR   :=   ,
          SFB_ERR   :=   ,
```

```
CFG_ERR   :=   ,
DB_ERR   :=   M8.0,
DB_ERRNO   :=   MW10,
DB_ACT   :=   MW12,
OFFSET_AC   :=    MW14)

Network 4: interprete answer according to the containing data
```

5.2.4 FB DEV_FLT4: Reading the fault buffer of a SINAMICS G/S - S7-1200/1500

Description

FB39

Can be used in the following CPUs: S7-1200, S7-1500

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

Description of functions

The block reads the last fault from the fault buffer of the drive. Reading is initiated by a positive edge at the START input of the block. The telegram for reading the fault buffer of the drive is saved in the block and its structure corresponds to PROFIdrive-Profile Drive Technology, version 4.1.

The number of the data set used can be specified via the input DS_NO. If the input is supplied with the value 0, data set 47 is used. For values <> 0, the value that is at the input is viewed as the number of the data set to be used. However, the telegram must continue to be structured as per PROFIdrive-Profile Drive Technology, Version 4.1, because otherwise the block will report an error. This means that any data set which is supported by the device can be used as long as the contents of the telegram correspond to PROFIdrive-Profile Drive Technology, Version 4.1. This function is mainly relevant for drives with PROFINET interface.

The START bit must be reset with the BUSY signal. A new block call can only take place with the checkback signal DONE or xyz_ERR. For the data transfer, the FB uses the mechanism of the acyclic communication. No coordination of the block calls, which uses the acyclic communication channel to the same slave, is carried out. This must be implemented within the applications (see Interlocking of blocks with acyclic communication).

I/O bar

The following table shows the input and output parameters:

Parameter	Declaration	Data type	Memory area	Description
Input				
LADDR	INPUT	HW_IO		Diagnostics address of the slave
DS_NO	INPUT	WORD	E, A, M, D, L, const.	Number of data set be read out
AXIS	INPUT	BYTE	E, A, M, D, L, const.	Drive ID for multi-axis drives
START	INPUT	BOOL	E, A, M, D, L	Accept start pulse for job
Output				
BUSY	OUTPUT	BOOL	E, A, M, D, L	Job running
DONE	OUTPUT	BOOL	E, A, M, D, L	Job complete without error
REQ_ERR	OUTPUT	BOOL	E, A, M, D, L	Job complete with errors
WDOG_ERR	OUTPUT	BOOL	E, A, M, D, L	Watchdog error, no plausible response data available
SFB_ERR	OUTPUT	BOOL	E, A, M, D, L	SFB 53 WRREC/SFB 52 RDREC reports errors; you can find information on possible error messages in the TIA Portal online help under the keywords "Parameter STATUS" (search entry).
CFG_ERR	OUTPUT	BOOL	E, A, M, D, L	Slave is not or incorrectly configured
ERR_NO1	OUTPUT	WORD	E, A, M, D, L	Number fault 1 of fault occurrence
ERR_VAL1	OUTPUT	DWORD	E, A, M, D, L	Fault value for fault number of fault 1
...				
ERR_NO8	OUTPUT	WORD	E, A, M, D, L	Number fault 8 of fault occurrence
ERR_VAL8	OUTPUT	DWORD	E, A, M, D, L	Fault value for fault number of fault 8

When the job processing is error-free, the message bits and the data at the block outputs are valid until the next job is initiated.

Data area

The following table describes the data area:

Parameter	Declaration	Data type	Description
NODATA_CYCLE_NO	STAT	INT	The number of cycles that are waited for receiving response data before the job is retried (preassignment = 2500)
NODATA_RETRY_NO	STAT	BYTE	Number of times a job will be retried if no response data has been received (preassignment = 5)
SFB53_RET_VAL	STAT	INT	Return value of the SFB53 WRREC
SFB52_RET_VAL	STAT	INT	Return value of the SFB 52 RDREC
WDOG_RETRY_NO	STAT	BYTE	Number of times a job will be retried if no plausible response data has been received (preassignment = 5)

Error reactions

The following table describes the error displays:

Output	Error description
CFG_ERR	A configuration error is signaled if no or faulty configuration data is entered.
REQ_ERR	Command contains error code (response ID = 7)
SFB_ERR	SFB error during data transfer with the system functions SFB RDREC/WRREC (return value < 0) The return value is stored in the instance DB (SFB53_RET_VAL or SFB52_RET_VAL). To learn the meaning of the return value, see the online help for the SFB or see Formulating parameter jobs (data set 47) (Page 146).
WDOG_ERR	No plausible response data within the monitoring time (default setting: after 5 job retries)

Note

- If an xxx_ERR occurs, no information is displayed at the block outputs for the fault.
- In the event of a failed communication connection and initiated job, it is possible that the CPU may not immediately signal SFB_ERR. Instead, the block signals BUSY until the error is either cleared and the reading of the fault buffer is ended or the cycle monitoring (default setting: NODATA_CYCLE_NO x NODATA_RETRY_NO = 12500 cycles!) activates and SFB_ERR is reported. After the SFB_ERR is cleared, the user must re-initiate the DEV_FLT4.
- The immediate display of a communication fault is provided by the FC COM_STAT (FC60) in any case.
- If you attempt to read the fault buffer of a drive type other than SINAMICS G/S with the DEV_FLT4, this can lead to an erroneous result.
- The fault information is entered into the fault buffer of the SINAMICS with a time delay for reporting via bit 3 of the ZSW1. This must be considered when programming the block call.

Block call (STL source code)

```
U   read device fault
S    "DB_DEV_FLT4".START

CALL    DEV_FLT4, DB_DEV_FLT4(
          LADDR    := DRIVDB1.SLAVE_1.DADDR,
          DS_NO    := MW32,
          START    := ,
          BUSY    := ,
          DONE    := M30.2,
          REQ_ERR    := M30.3,
          WDOG_ERR    := M30.4,
          SFB_ERR    := M30.5,
          CFG_ERR    := M30.6,
          ERR_NO1    := MOTOR.ERR_NO1,
          ERR_VAL1    := MOTOR.ERR_VAL1,
          ERR_NO2    := MOTOR.ERR_NO2,
          ERR_VAL2    := MOTOR.ERR_VAL2,
          ERR_NO3    := MOTOR.ERR_NO3,
          ERR_VAL3    := MOTOR.ERR_VAL3,
          ERR_NO4    := MOTOR.ERR_NO4,
          ERR_VAL4    := MOTOR.ERR_VAL4,
          ERR_NO5    := MOTOR.ERR_NO5,
          ERR_VAL5    := MOTOR.ERR_VAL5,
          ERR_NO6    := MOTOR.ERR_NO6,
          ERR_VAL6    := MOTOR.ERR_VAL6,
          ERR_NO7    := MOTOR.ERR_NO7,
          ERR_VAL7    := MOTOR.ERR_VAL7,
          ERR_NO8    := MOTOR.ERR_NO8,
          ERR_VAL8    := MOTOR.ERR_VAL8)

U    "DB_DEV_FLT4".BUSY
R    "DB_DEV_FLT4".START
```

5.2.5 FB PDAT_UD2: Upload/download of the drive parameters (DS47) - S7-1500

Description

FB42

Can be used in the following CPUs: S7-1500

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

Characteristics

- Can be used for all drives that support "Parameter access with DPV1" (DS47) in accordance with "PROFIBUS Profile Drive Technology, V3.1, November 2002" (see Formulating parameter jobs (data set 47) (Page 146))
- Download / partial download functionality
- Upload / partial upload functionality
- The parameter DB structure must correspond to the definition Structure of the parameter job DB
- The block only functions if the optimized block access of the parameter blocks has been deactivated.

Description of functions

The block transfers the parameterization data of a drive from a data area of the CPU to the drive or reads it back to the CPU. The data can be distributed to several data blocks. The data blocks themselves are located either in the main memory or in the load memory of the CPU. If the download function (READ_EN = 0) is enabled, only write jobs and specially marked read jobs (job IDs 2 and 6). The corresponding job IDs are checked by the block. A job with an impermissible job ID is skipped and is not reported as an error by the block. The jobs can vary in length. The response data is not saved.

If the upload function (READ_EN = 1) is enabled, the jobs contained in the parameter DBs are converted to read jobs (except jobs with the job IDs 1 - 6). This allows the previously written parameters to be read back into the CPU (synchronization of the DB data to changed drive parameters).

The parameter DB can also contain read jobs, which are then carried out during the upload. Ensure that enough space is provided in the parameter DBs for the read jobs in order to be able to save the data that is read back in the DB as well. This data cannot be loaded back into the drive, because the block does not support the conversion of read jobs into write jobs.

The reading back of the parameter values is only possible if all of the parameter DBs are located in the main memory.

The block is also in a position to only write or read back parts of the DB(s). The DB structure must be modified for this purpose (see Example for partial DB transfer (Page 150)).

For the data transfer, the FB uses the mechanism of the acyclic communication. The jobs are formulated according to the PROFIdrive profile, version 4.1 (data set 47).

The number of the data set used can be specified via the input DS_NO. If the input is supplied with the value 0, data set 47 is used. For values <> 0, the value that is at the input is viewed as the number of the data set to be used. However, the telegram must continue to be structured as per PROFIdrive-Profile Drive Technology, Version 4.1, because otherwise the block will report an error. This means that any data set which is supported by the device can be used as long as the contents of the telegram correspond to PROFIdrive-Profile Drive Technology, Version 4.1. This function is mainly relevant for drives with PROFINET interface.

No coordination of the jobs that are requested by various applications is carried out.
This must be implemented within the applications (see Interlocking of blocks with acyclic communication (Page 132)).

For multi-axis drives, addressing is implemented via the "Axis" byte in the parameter job header.

DS47 (PROFIdrive profile)

- Request parameter value, simple (word/double word) job ID 1
- Write parameter value, simple (word/double word) job ID 2
- Request parameter value, several array elements (max. 234)
  Job ID 1
- Write parameter value, several array elements (max. 234)
  Job ID 2
- Request parameter value, multi-parameter job ID 1
- Write parameter value, multi-parameter job ID 2

Migration of the blocks

If you would like to continue using the blocks from existing DRIVE ES projects, you must take the following into consideration:

- If you migrate the project, the blocks can continue to be used.
- If you import sources, the "optimized block access" must be deactivated.
- When used with an S7-1500, the start bit must be reset immediately upon activation of the busy bit.

Data management in the load memory

Data blocks which are programmed as part of an STL program in a source file can be designated as "not relevant to process" (keyword UNLINKED). This means that these data blocks are only saved in the load memory when they are loaded into the CPU. Their content is then copied into the main memory as needed. This function is integrated into the PDAT_UD2 block.

Space can be saved in the main memory by means of data management in the load memory. The expandable load memory is used as a buffer (e.g. for parameter DB: only the parameter DB that is to be processed is loaded into the main memory).

If a data block is created with the "UNLINKED" parameter, the input "DB_UNLINKED" must be set to "TRUE". At the input "DB_NO", the number of the DB is specified in the main memory into which the data is copied. At the DB_NO_LM input, the number of the first parameter DB is located in the load memory. In this case, however, it is not possible to read the data back from the converter, because the data is no longer written back to the load memory.

If the input "DB_UNLINKED" is set to "FALSE", the number of the first parameter DB must be specified in the main memory at the input "DB_NO". The input "DB_NO_LM" is irrelevant in this case.

The copy DB in the main memory and the data blocks in the load memory must be 8192 bytes in size (for an example for generating a corresponding DB, see Tip). If the data is only kept in the main memory, the data blocks can also be larger or smaller than 8192 bytes, depending on the CPU.

If several parameter DBs are created in the load or main memory, reference is made at the end of each of these DBs to the following DB. If no other DB follows, you must enter "0" for the "number of the following DB" in the last data word.

Error logging

The input "LOG_FCT" parameterizes the response of block PDAT_UD2 to an error during download (REQ_ERR):

- If the status of the input is "FALSE", the download is cancelled when the first error occurs. The number of the currently processed parameter DB is displayed at the output "DB_NO_ACT", the number of the parameter is displayed at the output "PA_NO", and the error number is displayed at the output ERR_NO. For multi-parameter jobs, only the first erroneous sub-job is displayed (NOT_TERMINATED = TRUE).
- If the status of the input is "TRUE", the download is not terminated when an "REQ_ERR" occurs. Instead, the error is logged (the log file is located in the instance of the DB under sx_LOG...) and the download continues. The DB number, parameter number, index, and error number are stored in the log. The download is only cancelled for this function after 20 logged errors (NOT_TERMINATED = TRUE).
  Exception: If an error already occurs with the parameter job "Set drive to download mode" for preparing the download, the download is canceled immediately with "NOT_TERMINATED = TRUE".
  The data of the last error (number of parameter DB, parameter number, and error number) are displayed at the outputs "DB_NO_ACT", "PA_NO" and "ERR_NO". For multi-parameter jobs, every erroneous sub-job is entered in the download log.

The download at the end of each parameter job can be cancelled with the input "CANCEL" (NOT_TERMINATED = TRUE).

Do not consider SFB errors

An entry can be made in the instance data of the block (sx_PARA_NO. sw_PARA[0] ... sx_PARA_NO. sw_PARA[4]) as to whether the SFB error is or is not to be displayed for specific parameters. By default, these are the parameters 970, 971, and 972. These can be changed by the user via "Modify tags". "-1" must be entered in the unused memory cells. The use of this function only makes sense if the drive briefly breaks off communication during the writing of these parameters (copy RAM2ROM, PowerOnReset). For write jobs, these parameters should be located at the end of the jobs to be processed, because there is no check to see when the drive is ready to communicate again and thus the subsequent jobs report an SFB error again and the download is then cancelled. This means that only one job from the above list can be contained per download. If several parameters from the above list must be transferred, you must fall back on the functionality of the partial download. For the job "Save parameter in EEPROM", which is created by the "Convert parameter set in DB" tool, the following read job to determine whether saving was successful is also carried out and time-monitored.

Job processing (download)

1. Enter the number of the first parameter DB in the corresponding interface:
   - If the parameter DB(s) is (are) in the main memory, the number of the first parameter DB (parameter DB_UNLINKED = 0) is specified at the input DB_NO.
   - If the parameter DB is located in the load memory, the number of the DB into which the data from the load memory is to be copied is specified at the input DB_NO. The number of the first parameter DB (parameter DB_UNLINKED = 1) in the load memory is specified at the DB_NO_LM input.
2. The start address (the address of the field in which the version ID is saved) of the first job or sub-job is entered at the input START_ADDR.
3. Parameterize the relevant drive family at the DRIVE input.
4. Initiate download with the start bit (START).
5. The block checks whether the specified DB(s) exist on the CPU (if not: DB_ERR = 1).
6. The block imports the first job from the parameter DB. Reads the job ID, enters it into the send buffer, and transfers it to the drive depending on the job ID (BUSY = TRUE).
7. Reset start bit
8. The block checks the receive data from the drive:
   - Response reference (mirrored job reference)
   - Special jobs implemented without error and received data correspond to the comparison values
9. Response in inbox is positive
   If the parameter CANCEL = FALSE, the next job is taken from the parameter DB and transferred. This is repeated until all of the jobs are sent to the drive and processed (DONE = TRUE, BUSY = FALSE).
   To cancel the transfer, the parameter "CANCEL" must be set to "TRUE". If this is the case, the transfer is cancelled when the parameter currently being transferred is terminated (NOT_TERMINATED = TRUE). In addition, the current DB number is displayed at the output DB_NO_ACT and the number of the last processed parameter is displayed at output PA_NO.
10. Response in inbox is negative (response ID = 82hex)
    Job complete with errors (REQ_ERR). The error numbers are located in the parameter value of the response or several error numbers in the parameter values of the sub-jobs (DS47, multiparameter). This error number is displayed in the output parameter ERR_NO. For multi-parameter jobs (DS47), the error number of the first erroneous sub-job is indicated here. If the log function is active, all erroneous sub-jobs are logged.
11. No plausible response to a sent job:
    - Job is transferred to the drive again and receive data is checked
    - Specified number of job retries has been carried out without plausible response: Watchdog error
12. No response data is available after specified number of cycles (SFB reports error 80C0):
    - Job is transferred to the drive again
    - Specified number of job retries has been carried out and there is still no response data: SFB error

13. A group error is signaled at the ERROR block output if one of the following errors occurs: REQ_ERR, WDOG_ERR, SFB_ERR, CFG_ERR or DB_ERR.
14. Once the first job has been processed, the next job is carried out. This happens until the block encounters the next end ID (16#EEEE EEEE). Then it checks the following word to see whether a value < > 0 has been entered (following DB). If yes, the jobs in this DB are processed. Otherwise, the download/partial download is terminated.

Job processing (upload)

1. Enter the number of the first parameter DB in the corresponding interface:
   - If the parameter DB(s) is (are) in the main memory, the number of the first parameter DB (parameter DB_UNLINKED = 0) is specified at the input DB_NO.
   - If the parameter DB(s) is (are) in the load memory, an upload is not possible (DB_ERR = TRUE; DB_ERRNO = 14)
2. At the input START_ADDR, enter the start address (version ID) of the first job or sub-job.
3. Set the input READ_EN = TRUE and parameterize the relevant drive family at the DRIVE input.
4. Initiate download with the start bit (START).
5. The block checks whether the specified DB(s) exist on the CPU (if not: DB_ERR = 1).
6. The block imports the first job from the parameter DB and enters it in the send buffer. Then it converts it to a read job or takes the already formulated read job and transfers it to the drive (BUSY = TRUE). Write jobs without special ID are not converted into a write job (job IDs 1 - 6) are therefore skipped without error message.
7. Reset start bit.
8. The block checks the receive data from the drive:
   - Response reference (mirrored job reference)
   - Special jobs implemented without error and received data correspond to the comparison values
9. Response in inbox is positive
   If the parameter CANCEL = FALSE, the next job is taken from the parameter DB and transferred. This is repeated until all of the jobs are sent to the drive and processed (DONE = TRUE, BUSY = FALSE).
   To cancel the transfer, the parameter "CANCEL" must be set to "TRUE". If this is the case, the transfer is cancelled when the parameter currently being transferred is terminated (NOT_TERMINATED = TRUE). In addition, the current DB number is displayed at the output DB_NO_ACT and the number of the last processed parameter is displayed at output PA_NO.
10. Response in inbox is negative (response ID = 82hex)
    Job complete with errors (REQ_ERR). The error numbers are located in the parameter value of the response or several error numbers in the parameter values of the sub-jobs (DS47, multiparameter). This error number is displayed in the output parameter ERR_NO. For multi-parameter jobs (DS47), the error number of the first erroneous sub-job is indicated here. If the log function is active, all erroneous sub-jobs are logged.

11. No plausible response to a sent job:
    - Job is transferred to the drive again and receive data is checked
    - Specified number of job retries has been carried out without plausible response: Watchdog error
12. No response data is available after specified number of cycles (SFB reports error 80C0):
    - Job is transferred to the drive again
    - Specified number of job retries has been carried out and there is still no response data: SFB error
13. A group error is signaled at the ERROR block output if one of the following errors occurs: REQ_ERR, WDOG_ERR, SFB_ERR, CFG_ERR, or DB_ERR.
14. Once the first job has been processed, the next job is carried out. This happens until the block encounters the next end ID (16#EEEE EEEE). Then it checks the following word to see whether a value < > 0 has been entered (following DB). If yes, the jobs in this DB are processed. Otherwise, the download/partial download is terminated.

I/O bar

The following table shows the input and output parameters:

Parameter	Data type	Type	Description
Input			
LADDR	HW_IO	IN	Diagnostics address of the slave
DS_NO	WORD	IN	Number of data set be read out
START_ADDR	INT	IN	Absolute address after which the data that is to be transferred is saved in the DB (the address of the field in which the version ID is saved) (see Structure of the parameter job DB for FB PDAT_UD2 (Page 148))
START	BOOL	IN	Start download
CANCEL	BOOL	IN	Cancel download process (cancellation takes place after the current parameter job is completed)
DB_NO	INT	IN	DB_UNLINKED = 1: Number of the DB in the main memory into which the data from the load memory is copied DB_UNLINKED = 0: Number of the first parameter DB in the main memory
DB_NO_LM	INT	IN	Number of the first parameter DB in the load memory
DB_UNLINKED	BOOL	IN	=1: Parameter DB(s) is (are) in the load memory
LOG_FCT	BOOL	IN	=1: Errors that occur during the download are logged (max. 20 errors)
READ_EN	BOOL	IN	=1 Read parameter; =0 write parameter
Output			
BUSY	BOOL	OUT	=1: Download in progress (parameters are being transferred)
DONE	BOOL	OUT	=1: Download completed without errors (all of the parameter jobs have been transferred without errors)
ERROR	BOOL	OUT	Group error
WDOG_ERR	BOOL	OUT	Watchdog error
SFB_ERR	BOOL	OUT	SFB 53 WRREC/SFB 52 RDREC reports error
CFG_ERR	BOOL	OUT	Slave is not or incorrectly configured
DB_ERR	BOOL	OUT	Error parameter DB (see DB_ERRNO for explanation)

Parameter	Data type	Type	Description
DB_ERRNO	INT	OUT	Error code to DB_ERR 1 = Specified DB does not exist in the main memory 2 = Specified DB does not exist in the load memory 3 = DB in the main memory < > 8192 bytes 4 = Job > 240 bytes 5 = Number of parameters < 1 or > 39 (DS47) 6 = Number of elements > 234 (DS47) 7 = Incorrect ID for job start or end ID 8 = Start address + minimum job length > DB size 9 = Incorrect start address (start address + 2 <> data set ID) or "optimized block access" for parameter DB active 10 = Start address is not at the word limit or the job length is an uneven number 11 = Read job without 4 bytes of space for data 12 = Address of the data to be copied is outside of the data DB 14 = Read-back of the parameters is not possible because DBs are in the load memory 15 = Number of parameters/elements with system of units > 2 or < 1 16 = Incorrect system of units 17 = Parameter not stored in EEPROM 18 = After write job "Save parameter in EEPROM" no read job follows 19 = The download is not concluded after saving the parameters in the "EEPROM" 20 = The download did not start with the first DL-DB 21 = "End parameter download mode" erroneous 22 = At the minimum a REQ_ERR occurred during the download with activated log function
REQ_ERR	BOOL	OUT	Response contains error ID ("82h")
ERR_NO	INT	OUT	Error feedback of the drive
NOT_TERMINATED	BOOL	OUT	Download canceled by the user • Canceled after the first REQ_ERR (LOG_FCT = FALSE) • Canceled after the 20th REQ_ERR (LOG_FCT = TRUE)
DB_NO_ACT	INT	OUT	Currently of the job DB currently being processed
PA_NO	INT	OUT	Number of the last edited parameter

The message bits and the data at the block outputs are valid until the next download is initiated. They are deleted when the PDAT_DL is restarted.

Data area

The following table describes the data area:

Parameter	Data type	Type	Comment
RD	ARRAY [1..240] of BYTE	STAT	Inbox for acyclic communication
SD	ARRAY [1..240] of BYTE	STAT	Outbox for acyclic communication
si_NODATA_CYCLE_NO	INT	STAT	The number of cycles that are waited for receiving response data before the job is retried (preassignment = 2500)
si_SFB52_RET_VAL	INT	STAT	Return value of the SFB 52 RDREC
si_SFB53_RET_VAL	INT	STAT	Return value of the SFB 53 WRREC
sx_LOG	ARRAY [0..19] of STRUCT	STAT	Logged data of the faulty parameter jobs
sx_LOG.si_DB_NO	INT	STAT	DB no. of the faulty parameter
sx_LOG.si_INDEX	INT	STAT	Index of the faulty parameter
sx_LOG.si_PARA_NO	INT	STAT	Number of the faulty parameter
sx_LOG.sw_ERR_NO	WORD	STAT	Error code of the faulty parameter
sx_PARA_NO. sw_PARA[0]	WORD	STAT	Parameter number for which the SFB error is not evaluated (default: P970,971,972). The user can enter additional or different parameter numbers.
sx_PARA_NO. sw_PARA[1]	WORD	STAT	
sx_PARA_NO. sw_PARA[2]	WORD	STAT	
sx_PARA_NO. sw_PARA[3]	WORD	STAT	
sx_PARA_NO. sw_PARA[4]	WORD	STAT	
sy_NODATA_RETRY_NO	BYTE	STAT	Number of times a job will be retried if no response data has been received (preassignment = 5)
sy_WDOG_RETRY_NO	BYTE	STAT	Number of times a job will be retried if no plausible response data has been received (preassignment = 5)

Error reactions

The following table describes the error displays:

Output	Error description
CFG_ERR	A configuration error is signaled if no or faulty configuration data is entered. The following data is checked: • Missing parameterization priority for the drive (response ID = "0Bh")
DB_ERR	Error parameter DB (see DB_ERRNO for explanation)

Output	Error description
DB_ERRNO	Error code to DB_ERR 1 = Specified DB does not exist in the main memory 2 = Specified DB does not exist in the load memory 3 = DB in the main memory < > 8192 bytes 4 = Job > 240 bytes 5 = Number of parameters < 1 or > 39 (DS47) 6 = Number of elements > 234 (DS47) 7 = Incorrect ID for job start or end ID 8 = Start address + minimum job length > DB size 9 = Incorrect start address (start address + 2 <> data set ID) or "optimized block access" for parameter DB active 10 = Start address is not at the word limit or the job length is an uneven number 11 = Read job without 4 bytes of space for data 12 = Address of the data to be copied is outside of the data DB 14 = Read-back of the parameters is not possible because DBs are in the load memory 15 = Number of parameters/elements with system of units > 2 or < 1 16 = Incorrect system of units 17 = Parameter not stored in EEPROM 18 = After write job "Save parameter in EEPROM" no read job follows 19 = The download is not concluded after saving the parameters in the "EEPROM" 20 = The download did not start with the first DL-DB 21 = "End parameter download mode" erroneous 22 = At the minimum a REQ_ERR occurred during the download with activated log function
ERR_NO	Error feedback of the drive For a description of the error, see Formulating parameter jobs (data set 47) (Page 146)
ERROR	Group error
NOT_TERMINATED	Download canceled by the user • Canceled after the first REQ_ERR (LOG_FCT = FALSE) • Canceled after the 20th REQ_ERR (LOG_FCT = TRUE)
REQ_ERR	Response contains error ID ("82h")

Output	Error description
SFB_ERR	SFB error during data transfer with the system functions SFB RDREC/WRREC (SFB return value < 0) • The return value is stored in the instance DB (SFB53_RET_VAL or SFB52_RET_VAL). • The inbox is emptied when there is an error reading the data. No response data available (SFB return value = 80C0) • The error is reported after the number of job retries with the number of wait cycles. • The return value is stored in the instance DB (SFB53_RET_VAL). To learn the meaning of the return value, see the online help for the SFB or see Formulating parameter jobs (data set 47) (Page 146).
WDOG_ERR	No plausible response data within the monitoring time (job data of job and response do not agree)

Note

Display of a communication fault

- In the event of a failed communication connection and initiated job, it is possible that the CPU may not immediately signal SFB_ERR. Instead, the block signals BUSY until the error is either cleared and the job is ended or the cycle monitoring (default setting: NODATA_CYCLE_NO x NODATA_RETRY_NO = 12500 cycles!) activates and SFB_ERR is reported. After the SFB_ERR is cleared, the user must re-initiate his job.
- The immediate display of a communication fault is provided by the FC COM_STAT (FC60) in any case.

Block call (STL source code)

```
U   START_UPLOAD_DOWNLOAD
S    "DB_PDAT_UD2".START

CALL    "PDAT_UD2" , "DB_PDAT_UD2" (
      LADDR    := "DRIVDB1".SLAVE_3.DADDR,
      DS_NO    .= MW    38,
      START_ADDR    := MW    20,
      START    := ,
      CANCEL    := M     22.1,
      DB_NO    := MW    24,
      DB_NO_LM    := MW    26,
      DB_UNLINKED    := M     22.2,
      LOG_FCT    := M     22.3,
      READ_EN    := M     22.4,
      BUSY    := ,
      DONE    := M     22.6,
      ERROR    := M     22.7,
      WDOG_ERR    := M     23.0,
      SFB_ERR    := M     23.1,
      CFG_ERR    := M     23.2,
      DB_ERR    := M     23.3,
      DB_ERRNO    := MW    30,
      REQ_ERR    := M     23.4,
      ERR_NO    := MW    32,
      NOT_TERMINATED    := M     23.5,
      DB_NO_ACT    := MW    34,
      PA_NO    := MW    36)

U    "DB_PDAT_UD2".BUSY
R    "DB_PDAT_UD2".START
```

Reading back parameters cyclically from the drive

The upload function of the block can also be used with corresponding programming (see following logic) for the cyclic reading of parameters as well.

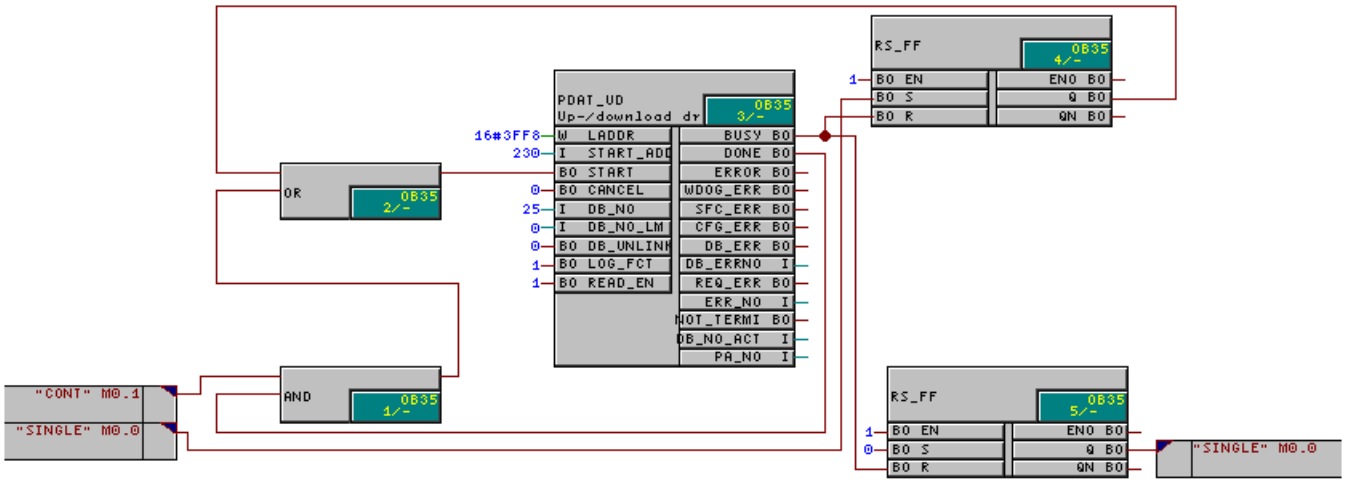


Figure 5-5 Reading back parameters cyclically from the drive

Structure of the parameter job DB (example for complete DB transfer)

See Example for complete DB transfer (Page 148)

Structure of the parameter job DB (example for partial DB transfer)

See Example for partial DB transfer (Page 150)

5.2.6 Function block SINA_POS (FB284)

5.2.6.1 Description

SINA_POS block

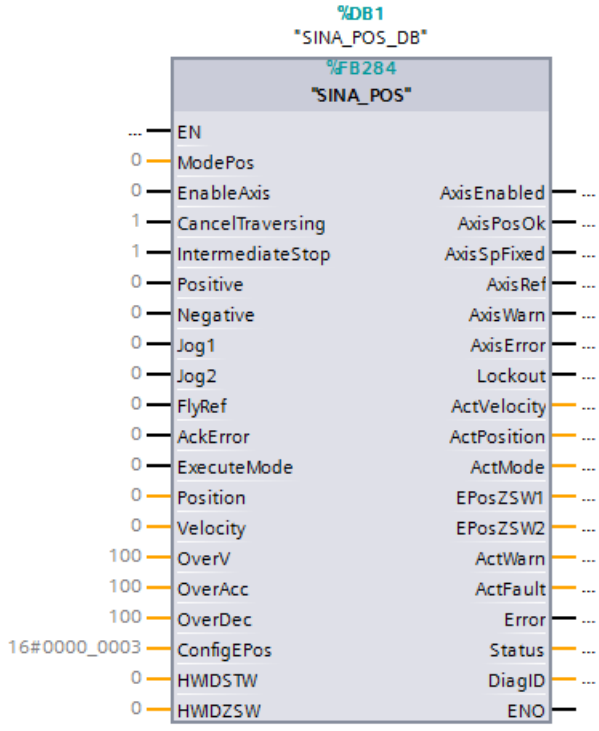


Figure 5-6 SINA_POS 284 S7 1200/1500 CPU

Description

The corresponding instance DB is automatically created when the FB284 (SINA_POS) is integrated.

Can be used in the following CPUs: SIMATIC S7-1200/1500

5.2.6.2 Calling OBs

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

5.2.6.3 Called blocks

Called blocks

DPRD_DAT/SFC14

DPWR_DAT/SFC15

5.2.6.4 Description of functions

Function description - general

The function block can be used to cyclically activate a SINAMICS drive with the SINAMICS S/G basic positioner technology.

Note

Because of the various EPOS modes, there is a special mode input – the input "ModePos". The individual operating modes are selected by means of this input. Due to the structure of the EPOS, it is not possible to select different operating modes simultaneously. It is possible at any time, however, to switch to different modes within an operating mode such as switching from setup mode to absolute positioning.

For detailed information, see Operating mode selection of EPOS with SINA_POS (FB284) (Page 95).

Note

To control all additional bits in the setpoint direction without an explicit input, from TIA Portal/ Startdrive V14 an additional configuration input is available – the input "ConfigEPOS". Using this input, it is now possible to activate basic device functions such as OFF2/OFF3 – or also EPOS functions such as continuous setpoint transfer – **without** having to intervene in the instance data block using a SLICE access.

Note

When configuring the SINAMICS drive, you must ensure that the standard type 111 telegram is selected for communication.

5.2.6.5 Input interface of the SINA_POS

Input interface

The input interface comprises 19 inputs in different data formats.

During the initial configuration of the function block, these are set up with initial values. An overview of the input interface is provided below:

Input signal	Type	Default[...]	Meaning
ModePos	INT	0	Operating mode: 1 = relative positioning 2 = absolute positioning 3 = positioning as setup 4 = homing procedure 5 = set home position 6 = traversing block 0 – 15/63 (G120/S120) 7 = jog 8 = jog incremental
EnableAxis	BOOL	0	Switch command: 0 = OFF1, 1 = ON
CancelTraversing	BOOL	1	0 = reject active traversing job, 1 = do not reject
IntermediateStop	BOOL	1	0 = active traversing command is interrupted, 1 = no intermediate stop
Positive	BOOL	0	Positive direction
Negative	BOOL	0	Negative direction
Jog1	BOOL	0	Jog signal source 1
Jog2	BOOL	0	Jog signal source 2
FlyRef	BOOL	0	0 = deselect flying homing, 1 = select flying homing
AckError	BOOL	0	Acknowledgement of faults
ExecuteMode	BOOL	0	Activate traversing job/setpoint acceptance/activate reference function
Position	DINT	0[LU]	Position setpoint in [LU] for operating mode Direct setpoint specification/MDI OR traversing block number for operating mode Traversing block
Velocity	DINT	0[1000LU/min]	Velocity in [1000LU/min] for MDI operating mode
OverV	INT	100[%]	Velocity override for all operating modes effective: 0-199%
OverAcc	INT	100[%]	Acceleration override effective 0-100%
OverDec	INT	100[%]	Deceleration override effective 0-100%
ConfigEPos	DWORD	3h	Detailed description, see Description of the configuration input "ConfigEPos" (Page 93).
HWIDSTW	HW_IO	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the setpoint slot, see Selection of the right hardware submodules (Page 24).
HWIDZSW	HW_IO	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the actual value slot, see Selection of the right hardware submodules (Page 24).

See also

Operating mode selection of EPOS with SINA_POS (Page 95)

5.2.6.6 Description of the configuration input "ConfigEPos"

Configuration input "ConfigEPos"

ConfigEPos	Meaning	PZD	Interconnection in the drive (telegram 111)	Default
Bit0	OFF2 (1 = no pulse disable)	1	r2090.1 = p 844[0]	1
Bit1	OFF3 (1 = no ramp stop)	1	r2090.2 = p 848[0]	1
Bit2	Software limit switch (active = 1)	3	r2092.14 = p2582	0
Bit3	Stop cams (active = 1)	3	r2092.15 = p2568	0
Bit4	Measuring input edge evaluation	3	r2092.11 = p2511[0]	0
Bit5	Measuring input selection	3	r2092.10 = p2510[0]	0
Bit6	Signal source reference mark	3	r2092.2 = p2612	0
Bit7	External block change (by BUS)	1	r2090.13 = p2633	0
Bit8	Continuous setpoint acceptance MDI (active = 1)	2	r2091.12 = p2649	0
Bit9	DDS BIT0	4	r2093.0 = 820[0]	0
Bit10	DDS BIT1	4	r2093.1 = 821[0]	0
Bit11	DDS BIT2	4	r2093.2 = 822[0]	0
Bit12	DDS BIT3	4	r2093.3 = 823[0]	0
Bit13	DDS BIT4	4	r2093.4 = 824[0]	0
Bit14	Parking axis selection	4	r2093.7 = p897	0
Bit15				
Bit16	Reserve - can be used as required below	1	r2090.14	0
Bit17	Reserve - can be used as required below	1	r2090.15	0
Bit18	Reserve - can be used as required below	2	r2091.6	0
Bit19	Reserve - can be used as required below	2	r2091.7	0
Bit20	Reserve - can be used as required below	2	r2091.11	0
Bit21	Reserve - can be used as required below	2	r2091.13	0
Bit22	Reserve - can be used as required below	3	r2092.3	0
Bit23	Reserve - can be used as required below	3	r2092.4	0
Bit24	Reserve - can be used as required below	3	r2092.6	0
Bit25	Reserve - can be used as required below	3	r2092.7	0
Bit26	Reserve - can be used as required below	3	r2092.12	0
Bit27	Reserve - can be used as required below	3	r2092.13	0
Bit28	Reserve - can be used as required below	4	r2093.5	0
Bit29	Reserve - can be used as required below	4	r2093.6	0
Bit30	Reserve - can be used as required below	4	r2093.8	0
Bit31	Reserve - can be used as required below	4	r2093.9	0

5.2.6.7 Output interface of the SINA_POS

Output interface SINA_POS

The output interface comprises 16 outputs in different data formats.

During the initial configuration of the block, these are set up with initial values. Below the overview of the output interface:

Output signal	Type	Default[...]	Meaning
AxisEnabled	BOOL	0	Drive is ready and switched on
AxisPosOk	BOOL	0	Target position of the axis reached
AxisSpFixed	BOOL	0	1 = Setpoint is stationary (Notice: Information dependent on SINAMICS firmware version: 1. SINAMICS S/G120 FW <4.8 / <4.7.9: Transfer of parameter r2199.0. 2. SINAMICS S/G120 FW ≥ 4.8 / ≥ 4.7.9: Transfer of parameter r2683.2 3. SINAMICS V90 PN Transfer of parameter r2683.2
AxisRef	BOOL	0	Home position set
AxisWarn	BOOL	0	Alarm of the drive effective
AxisError	BOOL	0	Drive is faulted
Lockout	BOOL	0	Switching on inhibited
ActVelocity	DINT	0	Current velocity (standardized 40000000h = 100% p2000)
ActPosition	DINT	0[LU]	Current position in LU
ActMode	INT	0	Currently active operating mode
EPosZSW1	WORD	0	Status of the EPOS ZSW1 (bit-granular)
EPosZSW2	WORD	0	Status of the EPOS ZSW2 (bit-granular)
ActWarn	WORD	0	Current alarm number
ActFault	WORD	0	Current fault number
Error	BOOL	0	1 = group fault present
Status	INT	0	16#7002: No fault – block is working 16#8401: Drive fault 16#8402: Switching on inhibited 16#8403: Flying homing could not be started 16#8600: Error DPRD_DAT 16#8601: Error DPWR_DAT 16#8202: Incorrect operating mode selected 16#8203: Incorrect setpoints parameterized 16#8204: Incorrect traversing block number selected
DiagID	WORD	0	Expanded communication error → SFB call error

5.2.6.8 Operating mode selection of EPOS with SINA_POS

General operating conditions

The axis is switched on via the input bit "EnableAxis" = 1. 1 is preassigned to OFF2 and OFF3 via the input "ConfigEPOS" and do not have to be written for operation.

The axis is ready to be switched on if there is no active error – "AxisError"= "0" – and no switch-on inhibit – "Lockout" = "0". The checkback signal "AxisIEnabled" goes to "1" after switching "EnableAxis".

The input "ModePos" is decisive for the operating mode selection. The desired operating mode is selected by means of this input. It is therefore not possible to select several operating modes at the same time. It is possible, however, to change-over between various lower-level operating modes.

Example: Setup mode ("ModePos"=3) with on-the-fly change-over to absolute positioning ("ModePos"=2).

The input signals "CancelTraversing" and "IntermediateStop" are relevant in all operating modes except for jog and must be set to "1" to operate the EPOS.

1. Setting the "CancelTraversing" bit to "0" leads to a ramp stop with 100% of the set delay. The job data is rejected and the axis can be assigned with a new job from the standstill. In this state, a mode change is possible.
2. Setting the "IntermediateStop" bit to "0" leads to a ramp stoppage of the axis with the currently applicable acceleration values. The job data is NOT rejected, which means that a setting of "1" allows the axis to continue its motion. It is possible to changes modes in a standstill.
3. The flying homing function can be selected and deselected in any operating mode other than the homing procedure mode at any time via the input "FlyRef".

5.2.6.9 Relative positioning

Relative positioning

The "Relative positioning" operating mode is implemented via the drive function "MDI relative positioning". It permits the position-controlled traversing of traversing paths via the integrated position controller of the SINAMICS drive.

1. Requirements:
   - The operating mode is selected with ModePos=1.
   - Device switched on via "EnableAxis".
   - The axis does not have to be referenced or the encoder adjusted.
   - The axis is in standstill if a mode higher than 3 is selected. It is possible to make a change within the MDI operating modes (1,2,3) at any time.
2. Sequence:

The traversing path and dynamic response are specified via the inputs "Position", "Velocity", "OverV" (velocity override), "OverAcc" (acceleration override), "OverDec" (deceleration override).

The velocity override refers to "Velocity".

The operating conditions "CancelTraversing" and "IntermediateStop" must be set to "1". "Jog1" and "Jog2" have no effect and should be set to "0" (false).

In relative positioning, the direction of travel basically results from the sign of the traversing path.

Traversing is started by a positive edge to "ExecuteMode". The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2" (see EPOS telegram 111 (Page 168) for details on the assignment of the status words).

The block acknowledges the successful reaching of the end of the traversing path "AxisPosOk". If a fault occurs during the traversing movement, the "Error" output signal is active.

Note

The currently running command can be replaced on-the-fly by a new command via "ExecuteMode". This is only possible for the operating modes of the "ModePos" 1,2,3.

Example of relative positioning

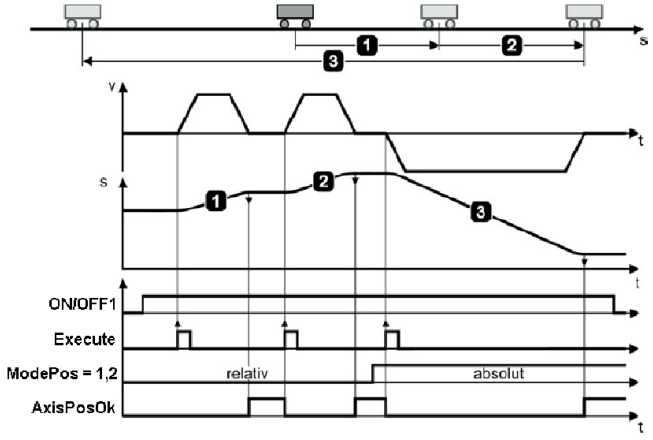


Figure 5-7 Relative positioning

5.2.6.10 Absolute positioning

Absolute positioning operating mode

The **Absolute positioning** operating mode is implemented via the drive function "MDI absolute positioning". It permits the position-controlled approach of absolute positions via the integrated position controller of the SINAMICS drive.

1. Requirements:
   - The operating mode is selected with "ModePos"=2.
   - The device is switched on via "EnableAxis".
   - The axis must be homed or the encoder must be adjusted.
   - The axis is in standstill if a mode higher than 3 is selected. It is possible to make a change within the MDI operating modes (1,2,3) at any time.
2. Sequence:

The traversing path and dynamic response are specified via the inputs "Position", "Velocity", "OverV" (velocity override), "OverAcc" (acceleration override), "OverDec" (deceleration override).

The velocity override refers to "Velocity".

The operating conditions "CancelTraversing" and "IntermediateStop" must be set to "1". Jog1 and Jog2 have no effect and must be set to "0".

In absolute positioning, the direction of travel basically results from the shortest path to the target position. The inputs "Positive " and "Negative" are "0".

Note

If a preferred direction to approach the target position is to be specified for a modulo axis, this can be performed with "Positive" or "Negative".

Simultaneous selection of "Positive" and "Negative" immediately stops the axis with further warnings or faults. For a linear axis, the selection is not effective and is ignored.

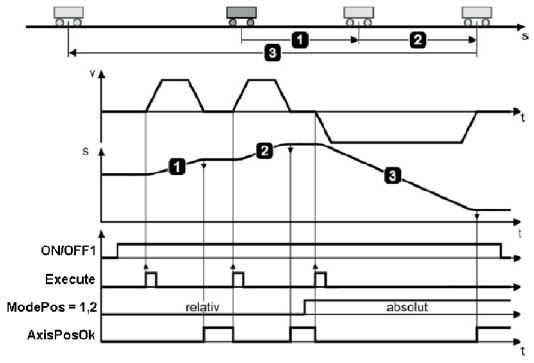
Traversing is started by a positive edge to "ExecuteMode". The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2" (see EPOS telegram 111 (Page 168) for details on the assignment of the status words).

The block acknowledges the successful reaching of the end of the traversing path "AxisPosOk". If a fault occurs during the traversing movement, the "Error" output signal is active.

Note

The currently running command can be replaced on-the-fly by a new command via "ExecuteMode". This is only possible for the operating modes of the "ModePos" 1,2,3.

Example of absolute positioning



5.2.6.11 Setup mode

Setup mode

The setup mode permits the position-controlled traversing of the axis in a positive or negative travel direction at constant speed without specification of a target position by means of the "MDI setup" drive function.

1. Requirements:
   - The operating mode is selected with "ModePos" = 3.
   - Switch on device via "EnableAxis".
   - The axis does **not** have to be referenced or the encoder adjusted.
   - The axis is in standstill if a mode higher than 3 is selected. It is possible to make a change within the MDI operating modes (1,2,3) at any time.
2. Sequence:

The traversing path and dynamic response are specified via the inputs "Position", "Velocity", "OverV" (velocity override), "OverAcc" (acceleration override), "OverDec" (deceleration override).

The operating conditions "CancelTraversing" and "IntermediateStop" must be set. "Jog1" and "Jog2" have no effect and must be set to "0".

The direction of travel is determined by "Positive" and "Negative". Simultaneous selection stops the axis without further alarms or faults.

Traversing is started by a positive edge to "ExecuteMode". The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2" (see EPOS telegram 111 (Page 168) for details on the assignment of the status words).

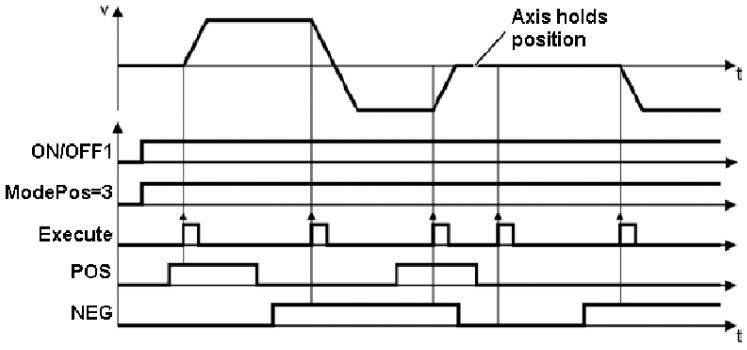
The output signal "AxisPosOk" is set if the setup mode ends with Reject traverse task and the axis has come to a standstill.

If a fault occurs during the traversing movement, the "Error" output signal is active.

Note

The currently running command can be replaced on-the-fly by a new command via "ExecuteMode". This is only possible for the operating modes of the "ModePos" 1,2,3.

Example of setup mode



5.2.6.12 Continuous setpoint acceptance

Description

Note

Continuous setpoint acceptance

The continuous setpoint acceptance is a special function of the preset positioning mode. By means of the parameter p2649 – which can be found in the standard telegram in the EPOS STW1 BIT12 – it is possible to accept these values directly in the EPOS WITHOUT edge triggering MDI setting values (position, speed, etc.).

Access takes place via the input "ConfigEPOS". Example: ConfigEPOS = 3h (Standard) -> ConfigEPos = 103h

259 = (3+(2^8)) (with direct setpoint acceptance) = 103h.

5.2.6.13 Referencing – reference point approach

Referencing – reference point approach

The operating mode allows the homing procedure of the axis in a positive or negative direction of travel with pre-configured velocity and homing mode and is activated via the drive function "Active homing".

1. Requirements:
   - The operating mode is selected with "ModePos"=4.
   - Switch on device via "EnableAxis".
   - The axis is at a standstill
2. Sequence:

The specification of the desired velocity is saved as velocity profile in the SINAMICS drive. Furthermore, the preset acceleration and deceleration values act in the traversing profile of the axis. The velocity override "OverV" effects the preconfigured traversing speed.

The operating conditions "CancelTraversing" and "IntermediateStop" must be set. Jog1 and Jog2 have no effect and must be set to "0".

The direction of travel is determined by "Positive" and "Negative". Simultaneous selection is not permitted and will cause a fault.

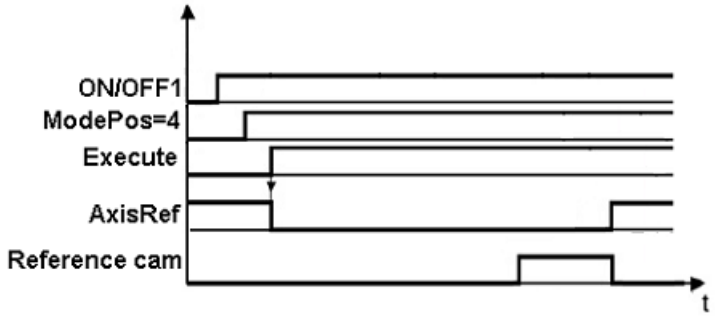
Homing is started with a positive edge to "ExecuteMode".

Traversing is started by a positive edge to "ExecuteMode". The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2", see Appendix (Page 145)).

The "AxisRef" output signal is set when the homing cam is found and evaluated accordingly.

If a fault occurs during the traversing movement, the "Error" output signal is output.

Simplified example of a reference point approach



Note

A detailed graphic representation of the reference point approach can be found in the Function Manual Basic Positioner, 04/2018, FW V4.7 SP10, A5E34257659A AF, and in the SINAMICS S120 List Manual. (/4/)

5.2.6.14 Homing - set home position

Homing – set home position

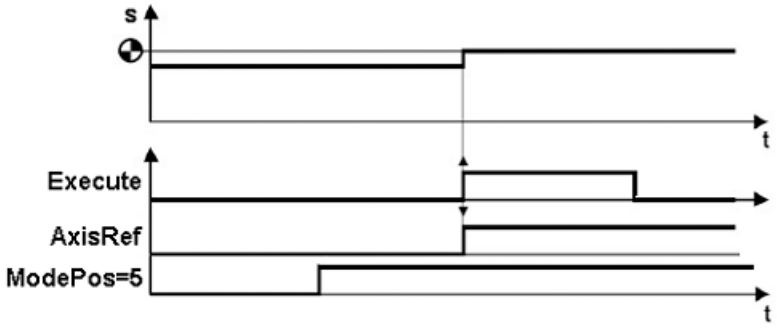
The Referencing – set reference point mode enables the referencing of the axis at an arbitrary position and is performed via the "Set reference point" drive function.

1. Requirements:
   - The operating mode is selected with "ModePos"=5.
   - The axis can be in closed-loop control, but must be at a standstill.
2. Sequence:

Axis is at a standstill and the home position is set by means of a positive edge for "ExecuteMode".

If a fault occurs while setting the home position, the "Error" output signal is output.

Example of set reference point



5.2.6.15 Traversing blocks

Traversing blocks

The Traversing blocks operating mode is implemented via the drive function "Traversing blocks". It permits the creation of automation programs, travel to fixed stop, and setting and resetting of outputs.

1. Requirements:
   - The operating mode is selected with "ModePos"=6.
   - Device switched on via "EnableAxis"
   - The axis is at a standstill
   - The axis must be homed or the encoder must be adjusted.
2. Sequence:

Note

The selection of the traversing job to be started is set via the input "Position". The value can only be between 0 and 63 (S120) or 0 and 15 (G120/S110). If the value is outside this range, an alarm is output at the block.

The job modes, target positions, and dynamic response are specified via the traversing block parameters in the SINAMICS drive. The "OverV" velocity override refers to the setpoint velocity stored in the traversing block.

The operating conditions "CancelTraversing" and "IntermediateStop" must be set to "1". "Jog1" and "Jog2" have no effect and should be set to "0".

The direction of travel that results depends on the job mode and the position setpoint that is set. The inputs "Positive" and "Negative" are not relevant in this case and must be set to "0".

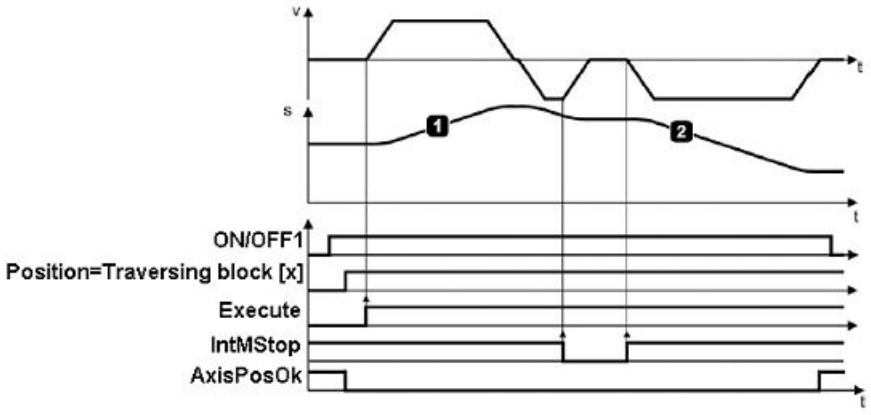
Note

If, in the case of a modulo axis, a preferred direction is specified for the approach of the target position, this can be set by selecting "AbsPos" or "AbsNeg" as the job mode.

Traversing is started by a positive edge to "ExecuteMode". The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2" (see EPOS telegram 111 (Page 168) for details on the assignment of the status words).

The block indicates the current processing of the command with "AxisEnabled" and acknowledges the successful reaching of the target position or the ending of the last task step with "AxisPosOk". If a fault occurs during the traversing movement, the "Error" output signal is active.

Example of traversing blocks



Note

The currently running command can be replaced on-the-fly by a new command via "ExecuteMode". This is only possible for the same operating mode.

5.2.6.16 Jog

Jog

The Jog operating mode is implemented via the drive function "Jog". It permits the position-controlled, velocity-dependent traversing of axes via the integrated position controller of the SINAMICS drive.

1. Requirements:
   - The operating mode is selected with "ModePos" = 7.
   - Device switched on via "EnableAxis".
   - The axis is at a standstill
   - The axis does **not** have to be homed or adjusted.
2. Sequence:

The jog speed is specified via the STARTER/Startdrive input screen or the acyclic communication for configuring the operating mode in the SINAMICS drive. For the dynamic response of the axis, the SINAMICS drive uses the set acceleration and delay in the SINAMICS drive.

The velocity override is also effective in the operating mode and is set via "OverV".

The operating conditions "CancelTraversing" and "IntermediateStop" are not relevant for the operating mode and can be set to "1" by default.

Note

"Jog1" and "Jog2" are the signal sources for jog mode in EPOS. The direction of the traversing movement of the respective signal source is configured in the SINAMICS drive and is set by default to Jog1 = negative and Jog2 = positive.

The direction of travel for jogging depends on the velocity setpoint that is set.

The inputs "Positive" and "Negative" are not relevant for the operating mode can be set to "0" by default.

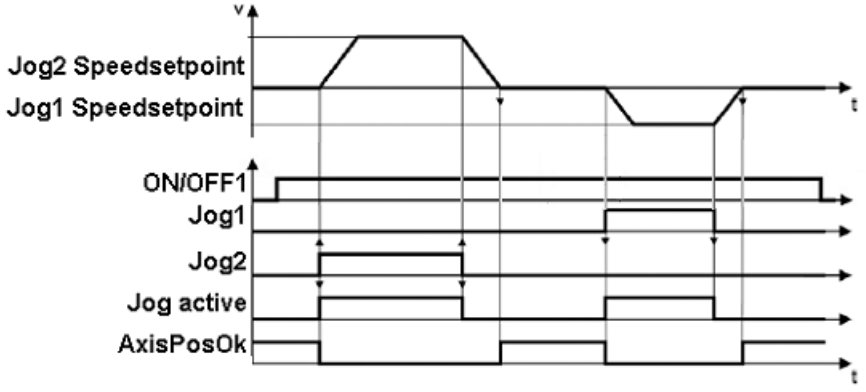
The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2" (see Appendix (Page 145) for details on the assignment of the status words).

The block indicates the current processing of the command with "AxisPosOK" and acknowledges the ending of the jog function (Jog1 or Jog2 = 0) when the axis comes to a standstill with "AxisPosOK". If a fault occurs during the traversing movement, the Error output signal is active.

Note

The currently running command can be replaced on-the-fly by a new command via "Jog1" or "Jog2". This is only possible when you are remaining in one of the jog modes.

Example for "jog" mode



5.2.6.17 Jog - incremental

Jog incremental

The Jog incremental operating mode is implemented via the drive function "Jog". It permits the position-controlled, path-dependent traversing of axes via the integrated position controller of the SINAMICS drive.

1. Requirements:
   - The operating mode is selected with "ModePos" = 8.
   - The device is switched on via "EnableAxis".
   - The axis is at a standstill
   - The axis does not have to be homed or adjusted.
2. Sequence:

The path and velocity are specified via the STARTER/Startdrive input screen or the acyclic communication for configuring the operating mode in the SINAMICS drive. For the dynamic response of the axis, the SINAMICS drive uses the configuration of the acceleration and delay in the SINAMICS drive.

The velocity override is also effective in the operating mode and is set via "OverV".

The operating conditions "CancelTraversing" and "IntermediateStop" are not relevant for the operating mode and can be set to "1" by default.

Note

"Jog1" and "Jog2" are the signal sources for jog mode in EPOS. The direction of the incremental traversing movement of the respective signal source is configured in the SINAMICS drive and is set to 1000LU (length units) in each case for incremental jogging.

The direction of travel for jogging depends on the path setpoint that is set.

The inputs "Positive" and "NEG" are not relevant for the operating mode can be set to "0" by default.

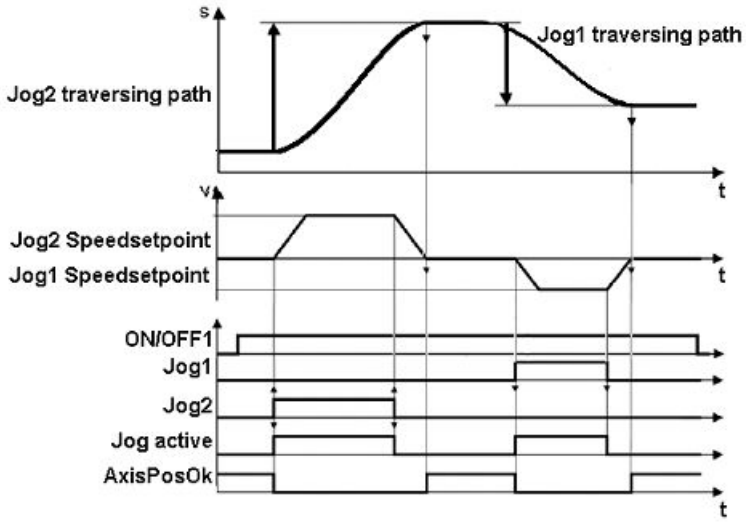
The current status of the active command can be monitored via "EPosZSW1 / EPosZSW2" (see Appendix (Page 145) for details on the assignment of the status words).

The block indicates the current processing of the command with "AxisEnabled" and acknowledges the ending of the jog function ("Jog1" or "Jog2" = 0) when the axis comes to a standstill with the bit AxisPosOk. If a fault occurs during the traversing movement, the "Error" output signal is active.

Note

The currently running command can be replaced on-the-fly by a new command via "Jog1" or "Jog2". This is only possible when you are remaining in one of the jog modes.

Example of incremental jogging



5.2.6.18 Flying homing

Flying homing

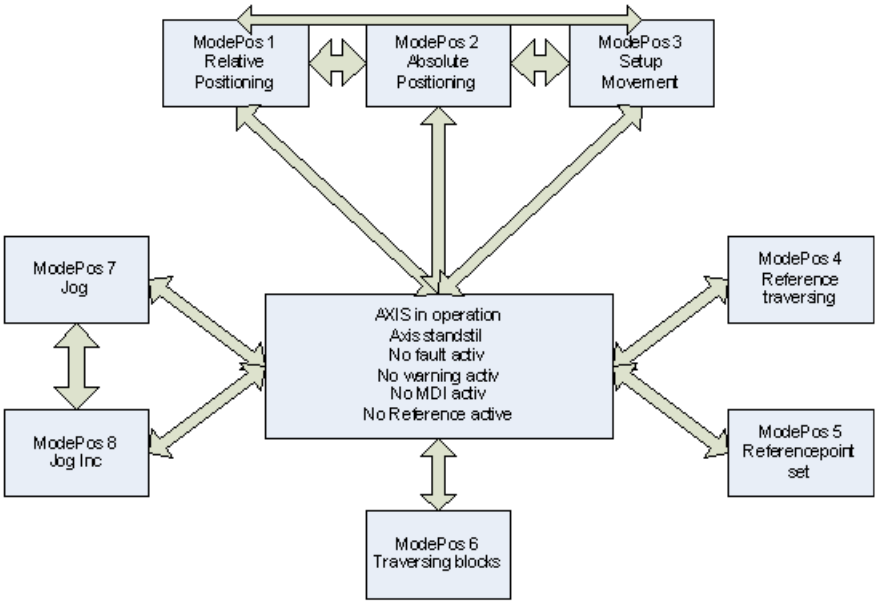
The operating mode Flying homing (passive homing) is implemented via the "Homing" drive function and is subordinate to most modes. It allows the SINAMICS drive to be re-homed during operation.

1. Requirements:
   - The input "FlyRef" is set to "1"
   - No selection of "ModePos" = 4 (homing procedure) and 5 (set home position)
2. Sequence:

The settings/prerequisites of the active operating mode apply. Flying homing can be selected or deselected at any time. When the set homing measuring input is reached, the setpoint and actual value are processed on the fly.

5.2.6.19 Operating mode change based on the ModePos values

Operating mode change based on the "ModePos" values



5.2.6.20 Troubleshooting the SINA_POS function block

Function block error

For fault detection, the group error "Error" is set and the "ErrorID" is set. The following errors that occur are monitored:

Error number Status	Cause	Remedy
16#7002	No error	
16#8600	Interruption of communication to the SINAMICS drive: Error DPRD_DAT	Check the communication connections/ settings (see DiagID)
16#8601	Interruption of communication to the SINAMICS drive: Error DPWR_DAT	Check the communication connections/ settings (see DiagID)
16#8202	Incorrect operating mode selected	Set "ModePos" from 1 to 8
16#8203	Incorrect parameterization of the override inputs	Check the settings of the override inputs
16#8204	Invalid traversing block number	Enter traversing block number from 0 to 63
16#8401	Fault message(s) in the SINAMICS drive	Evaluation of the error code at the output "ActFault"

Error number Status	Cause	Remedy
16#8402	Closing lockout of the SINAMICS drive active	Check whether axis/encoder is parked, safety functions active, parameter p10 ≠ 0
16#8403	Flying homing could not be started	Check for active alarms/errors in the drive,

- The faults of the SINAMICS drive are indicated via the "ActFault" output and can be acknowledged (if possible) via the "AckError" input.
- Active alarms do not have to be acknowledged. They are marked as cleared by the SINAMICS drive once the user has resolved the cause of the alarms.

Note

The meanings of the displayed faults and alarms are described in the List Manual of the relevant SINAMICS drive.

- The fault of the SFB call is displayed at the "DiagID" output and must be checked by the user. Once this fault has been cleared or goes away, the group error "Error" is rescinded as needed and the "Status" output is updated.

Note

If error message 8092(hex) occurs at the DIAGID output, the S7-1x00 firmware must be checked. The following applies:

- S7-1200 -> firmware at least 2.x
- S7-1500 -> firmware at least 1.1

5.2.7 Function block SINA_SPEED (FB285)

5.2.7.1 Description

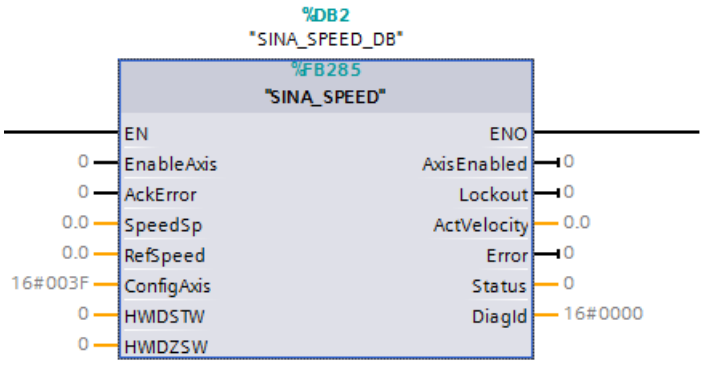


Figure 5-8 SINA_SPEED S7 1200/1500 CPU block

Description

The corresponding instance data block is automatically created when the FB285 (SINA_SPEED) is integrated.

Can be used in the following CPUs: SIMATIC S7-1200/1500

5.2.7.2 Calling OBs

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

5.2.7.3 Called blocks

Called blocks

DPRD_DAT/SFC14

DPWR_DAT/SFC15

5.2.7.4 Function description - general

Description of functions

Using the function block, a SINAMICS drive can be cyclically activated with the standard telegram 1.

Note

When configuring the SINAMICS drive, you must ensure that the standard telegram 1 is selected for communication.

Note

The interface for the block is limited to just a few inputs and outputs. All of the signals of the telegram can be reached in the direction of the setpoint at any time via the input "ConfigAxis". When the block is inserted, the inputs are filled with default values.

The axis is switched on via the input bit "EnableAxis" = 1. "1" is preassigned to OFF2 and OFF3 via the input "ConfigAxis" and they do not have to be written by the user for operation.

The axis is ready to be switched on if there is no active error – "Error"= "0" – and no switch-on inhibit – "Lockout" = "0".

The speed setpoint is specified directly on the block input "SpeedSp" in the REAL format. To undertake the necessary normalization of the setpoint, "RefSpeed" must be entered at the input – this corresponds to the parameter p2000 in the SINAMICS drive. The actual speed value is output at the output "ActVelocity" in the REAL format.

5.2.7.5 Input interface of the SINA_SPEED

Input interface SINA_SPEED

Input signal	Type	Default	Meaning
EnableAxis	BOOL	0	"EnableAxis" = 1 → switching on the drive
AckError	BOOL	0	Acknowledgement of axis fault → "AckFlt"=1
SpeedSp	REAL	0.0 [rpm]	Speed setpoint
RefSpeed	REAL	0.0 [rpm]	Rated speed of the drive → p2000
ConfigAxis	WORD	3	For additional information, see Pre-assignment of the ConfigAxis input (Page 111)
HWIDSTW	HW_IO	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the setpoint slot, see Selection of the right hardware submodules (Page 24)
HWIDZSW	HW_IO	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the actual value slot, see Selection of the right hardware submodules (Page 24)

5.2.7.6 Pre-assignment of the ConfigAxis input

Pre-assignment of the ConfigAxis input

ConfigAxis	Meaning	PZD	Interconnection in the drive	Default
Bit0	OFF2	1	r2090.1 = p 844[0]	1
Bit1	OFF3	1	r2090.2 = p 848[0]	1
Bit2	Inverter enable	1	r2090.3 = p 852[0]	1
Bit3	Ramp-function generator enable	1	r2090.4 = p1140[0]	1
Bit4	Continue ramp-function generator	1	r2090.5 = p1141[0]	1
Bit5	Speed setpoint enable	1	r2090.6 = p1142[0]	1
Bit6	Direction of rotation	1	r2090.11 = p1113[0]	0
Bit7	Holding brake must be opened	1	r2090.12 = p855[0]	0
Bit8	Motorized pot. setpoint higher	1	r2090.13 = p1035[0]	0
Bit9	Motorized pot. setpoint lower	1	r2090.14 = p1036[0]	0
Bit10	Reserve - can be used as required (bit 8)	1	r2091.8	0
Bit11	Reserve - can be used as required (bit 9)	1	r2091.9	0
Bit12	Reserve - can be used as required (bit 15)	1	r2091.15	0
Bit13				0
Bit14				0
Bit15				0

5.2.7.7 Output interface SINA_SPEED

Output interface SINA_SPEED

Output signal	Type	Default	Meaning
AxisEnabled	BOOL	0	Operating mode is executed or enabled
Lockout	BOOL	0	1 = switch-on inhibit active
ActVelocity	REAL	0.0[rpm]	Current velocity → depending on the normalization factor RefSpeed
Error	BOOL	0	1 = group fault present
Status	INT	0	16#7002: No error – block is being processed 16#8401: Error in drive 16#8402: Switching on inhibited 16#8600: Error DPRD_DAT 16#8601: Error DPWR_DAT
DiagID	WORD	0	Expanded communication error →SFB call error

Note

The complete status data of Telegram 1 can be found in Appendix (Page 145).

5.2.7.8 Troubleshooting the SINA_SPEED function block

Troubleshooting the SINA_SPEED function block

The group error "Error" is set if the SINAMICS drive is faulted or the switch-on inhibit of the SINAMICS drive is active or if the call of the SFB reports an error. A corresponding "Status" is also output:

Error number **Status**	**Meaning**	**Remedy**
16#7002	No fault active	
16#8401	Drive fault active	Evaluate active errors of the SINAMICS per acyclic communication
16#8402	Switching on of drive inhibited active	Check for parking axis, safety active, parameter p10 ≠ 0
16#8600 16#8601	SFB call error active	Clearing the communication fault

- The faults of the SINAMICS drive can be acknowledged via the "AcktError" input.
- The fault of the SFB call is displayed at the DiagID output and must be checked by the user. Once this fault has been cleared or goes away, the group error "Error" is rescinded as needed and the error ID "Status" is updated.

Note

If error message 8092(hex) occurs at the DIAGID output, the S7-1x00 firmware must be checked. The following applies:

- S7-1200 Firmware at least 2.x
- S7-1500 Firmware at least 1.1

5.2.8 Function block SINA_PARA (FB286)

5.2.8.1 Description

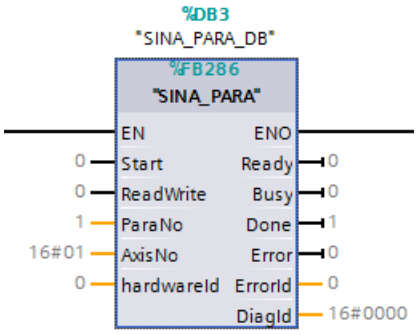


Figure 5-9 SINA_PARA S7 1200/1500 CPU block

Description

The corresponding instance data block is automatically created when the FB286 (SINA_PARA) is integrated.

Can be used in the following CPUs: S7-1200/1500.

5.2.8.2 Calling OBs

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

5.2.8.3 Called blocks

Called blocks

RDREC/SFB52

WRRECSFB53

5.2.8.4 Description of functions

With the aid of the block, up to a maximum of 16 parameters can be read or written acyclically to a SINAMICS S/G drive.

Note

The data is accessed by means of data set 47 according to the PROFIdrive profile. Data set 47 enables global acyclic access to the drive. For this a drive object number (AxisNo) is required.

At the input "ReadWrite" it is specified whether the number specified at the input "ParaNo" is to be written to or read from the SINAMICS drive.

The reading or the writing of the parameters is initiated by the edge-triggered input "Start".

The data of the parameters is stored in a preconfigured, internal structure of the created instance data block "sxParameter". The entire instance data block is freely accessible and can be changed. The data to be written or read is entered or displayed in the REAL or DINT format.

Note

Only the "sxParameter" area is to be adapted by the user or evaluated accordingly in the event of a transmission error. All other areas of the instance data block are needed for internal measures and must **not** be changed.

5.2.8.5 Input interface of SINA_PARA

Input interface of SINA_PARA

Input signal	Type	Default	Meaning
Start	BOOL	0	**Start of the job (0 = no job or cancel job; 1 = start and execute job)**
ReadWrite	BOOL	0	Job type 0=read, 1=write
ParaNo	INT	1	Number of parameters → 1 to 16
AxisNo	BYTE	16#01	Axis number/axis ID/DO number in multi-axis system
hardwareId	HW IO	0	Hardware ID of the module access points/actual value telegram slot of the axis or drive, see Selection of the correct hardware submodules (Page 24)

5.2.8.6 Output interface of SINA_PARA

Output interface of SINA_PARA

Output signal	Type	Default	Meaning
Ready	BOOL	0	Checkback signal for connecting in LacycCom environment; 1 = job ended or job canceled (one cycle long) See Chapter 7.2
Busy	BOOL	0	Job in progress if "Busy"=1
Done	BOOL	0	Job ended means edge change from 0->1
Error	BOOL	0	Active group error -> "Error" =1
Status	DWORD	0	1st word -> binary-coded indicating which parameter access is faulted 2nd word: type of fault
DiagId	WORD	0	Expanded communication error -> SFB call error

5.2.8.7 Data structure of the "sxParameter" area

Data structure of the "sxParameter" area

Job fields to be filled in by the user:

- sxParameter[x].siParaNo := parameter number (value range 1..65535)
- sxParameter[x1].siIndex := parameter index (value range 0..65535)
- sxParameter[x].srValue := parameter value (value range ±1.175 495e-38.. ±3.402823e+38) – is filled in when reading the block.
- sxParameter[x]sdValue := parameter value (value range -214748364810 (-2^31) to +214748364710 (2^31)

44		sxParameter	Array [1..16] of Struct			☑	☑	
45		sxParameter[1]	Struct			☑	☑	
46		siParaNo	Int	0		☑	☑	Number of parameter (Number 1..65535)
47		siIndex	Int	0		☑	☑	Subindex (Number 1..65535)
48		srValue	Real	0.0		☑	☑	Value of parameter
49		sdValue	DInt	0		☑	☑	Value of parameter
50		syFormat	Byte	B#16#00		☑	☑	Format of value (Format 0x40..0x44)
51		swErrorNo	Word	W#16#0000		☑	☑	Error number (see table below)

Figure 5-10 sxParameter

Note

TIA Portal/Startdrive V14

As of TIA Portal/Startdrive V14, the instance data block of the SINA_PARA in the "sxParameter" data structure contains two different input or output fields in the formats REAL and DINT (new!).

All of the parameters of the type DWORD or DINT must be written in the sxParameter[x].sdValue field as of this version. The block logic has been modified to the extent that the job field sxParameter[x].sdValue is used for reading and writing for the automatic detection of the DWORD or DINT format.

For all of the other parameters, the already existing sxParameter[x].srValue field is used as usual.

Note

Older versions

In contrast to older versions, beginning with this block version V4.x, the user must know whether the format of the parameter to be read/written involves DWORD/DINT or the remainder (byte, word, real, INT, etc.).

If this is not observed, problems can occur, especially when writing, because the default value of the DINT field ("0") is transferred here instead of the desired value (which was incorrectly entered in the REAL field).

Likewise, the evaluation of read operations for parameters in the DWORD/DINT format must be carried out via the new job field.

Note

Parameter structure

The parameter structure for the symbolic programming used is also compatible for older programs of the TIA Portal versions V12SP1 or V13SPx.

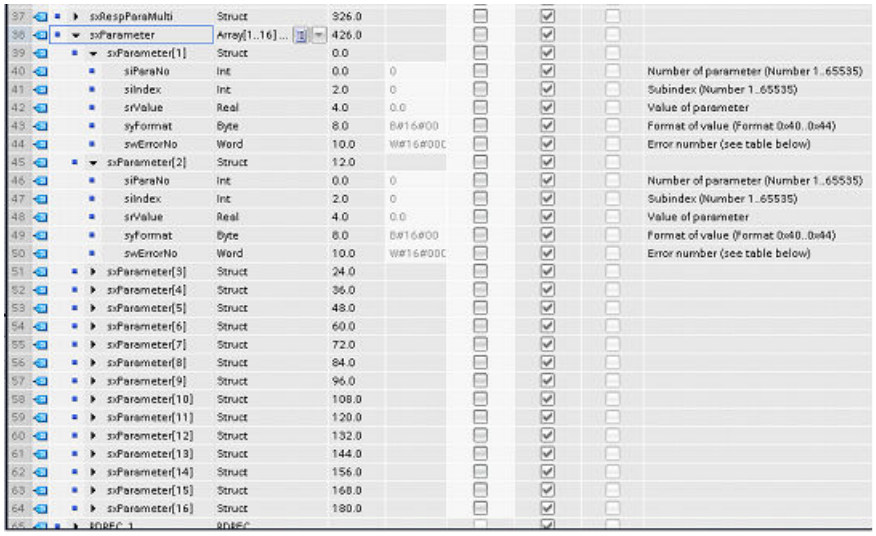
With the aid of the new job field, it is now possible to read/write BICO parameter interconnections without problems.

The different formats of the parameter are determined by the block itself. (Value range 0x40 = Zero,0x41/0x02/0x05 = Byte, 0x42/0x03/0x06 = Word, 0x43/0x04/0x07/0x08 = Dword, 0x44 = Error)

The following job fields are filled by the block:

- sxParameter[x].syFormat := parameter format
- sxParameter[x].swErrorNo := parameter error number (value range 0x0000..0x00FF)

You can find further information in the SINAMICS S120 Function Manual Communication in the section "Error values in parameter responses" on the internet: https://support.industry.siemens.com/cs/ww/en/view/109771803



37		sxRespParaMulti	Struct	326.0				
38		sxParameter	Array[1..16] ...	426.0				
39		sxParameter[1]	Struct	0.0				
40		siParaNo	Int	0.0	0			Number of parameter (Number 1..65535)
41		siIndex	Int	2.0	0			Subindex (Number 1..65535)
42		srValue	Real	4.0	0.0			Value of parameter
43		syFormat	Byte	8.0	B#16#00			Format of value (Format 0x40..0x44)
44		swErrorNo	Word	10.0	W#16#00			Error number (see table below)
45		sxParameter[2]	Struct	12.0				
46		siParaNo	Int	0.0	0			Number of parameter (Number 1..65535)
47		siIndex	Int	2.0	0			Subindex (Number 1..65535)
48		srValue	Real	4.0	0.0			Value of parameter
49		syFormat	Byte	8.0	B#16#00			Format of value (Format 0x40..0x44)
50		swErrorNo	Word	10.0	W#16#00			Error number (see table below)
51		sxParameter[3]	Struct	24.0				
52		sxParameter[4]	Struct	36.0				
53		sxParameter[5]	Struct	48.0				
54		sxParameter[6]	Struct	60.0				
55		sxParameter[7]	Struct	72.0				
56		sxParameter[8]	Struct	84.0				
57		sxParameter[9]	Struct	96.0				
58		sxParameter[10]	Struct	108.0				
59		sxParameter[11]	Struct	120.0				
60		sxParameter[12]	Struct	132.0				
61		sxParameter[13]	Struct	144.0				
62		sxParameter[14]	Struct	156.0				
63		sxParameter[15]	Struct	168.0				
64		sxParameter[16]	Struct	180.0				
65		RDREC_1	RDREC					

5.2.8.8 Writing parameters

Writing parameters

The "Write" action results in the parameter value and the format of the set parameter first being read from the SINAMICS drive and then written into the parameter structure. After being read successfully, the parameter value that was set by the user is then sent to the SINAMICS drive.

While this is taking place, the Busy bit is set to "1".

If the parameter to be written is erroneous, the associated parameter error number is also read and entered into the structure. At the same time, the corresponding error bit in the first word of the double word Status is set.

A successful write process is ended with the edge change "1 → 0" of the Busy bit and an edge change "0 → 1" of the Done bit. The Error bit must NOT be set during this. If this happens, the double word Status is to be evaluated.

5.2.8.9 Reading parameters

Reading parameters

The "Read" action results in the parameter value and the format of the set parameter being read from the SINAMICS drive and then written into the parameter structure. Then the value to be read is saved in the structure.

While this is taking place, the "Busy" bit is set to "1".

If the parameter to be read is erroneous, the associated parameter error number is also read and entered into the structure. At the same time, the corresponding error bit in the first word of the double word Status is set.

A successful read process is ended with the edge change "1->0" of the "Busy" bit and an edge change "0 → 1" of the "Done" bit. The Error bit must **not** be set during this. If this happens, the double word Status is to be evaluated.

5.2.8.10 Error handling of the FB286 function block

Troubleshooting function block SINA_PARA

The PROFIdrive errors that temporarily occur during the communication with the SINAMICS drive are identified and lead to the required action being repeated.

Note

The siErrorCount (current count) and siMaxErrCount parameters are listed in the instance data block. The siMaxErrCount can be edited by the user and specifies the maximum number of attempts to repeat the job when temporary errors occur (default 12500).

Error = 1 is then set and the status is set.

- During an active SFB fault, group error "Error = 1" is set, and an output is realized in the first word of the status as well as at output DiagID. Errors due to the SFB calls must not be acknowledged. Once these faults have been eliminated and a new job is started, the outputs DiagID, Error and ErrorID are taken back.
- If an incorrect value is entered at the input "ParaNo", this value is not taken into consideration, the group error "Error" is set, and the parameterization error is displayed at the output "ErroID".
- The group error "Error" is also set if a "Request" error occurs. For this error, the job is carried out, but one or more parameter accesses were not possible. The errors that occurred due to the access are binary coded and displayed in the second word of the double word "ErrorID". The job is also displayed as completed with "Done" = 1.

Evaluation of the ErrorID output

ErrorID	
ErrorID[1]	ErrorID[2]

ErrorID(1)	**Meaning**
0x000	No fault active
0x001	Internal telegram error active
0x002	Parameterization error active
0x003	SFB call error active
0x004	Cancelation of the job during the active data transfer by resetting the Start input to "0"
0x005	Unknown data type detected; evaluation of the ErrorID[2] shows the parameter with unknown data type in the highest value bit

ErrorID[2]	Meaning
0x00	No fault during parameter access
0x01	1st parameter access faulty For evaluation see swParameter[1].ErrorNo
0x02	2nd parameter access faulty For evaluation see swParameter[2].ErrorNo
0x04	3rd parameter access faulty For evaluation see swParameter[3].ErrorNo
0x08	4th parameter access faulty For evaluation see swParameter[4].ErrorNo
0x10	5th parameter access faulty For evaluation see swParameter[5].ErrorNo
0x20	6th parameter access faulty For evaluation see swParameter[6].ErrorNo
0x40	7th parameter access faulty For evaluation see swParameter[7].ErrorNo
0x80	8th parameter access faulty For evaluation see swParameter[8].ErrorNo
0x100	9th parameter access faulty For evaluation see swParameter[9].ErrorNo
0x200	10th parameter access faulty For evaluation see swParameter[10].ErrorNo
0x400	11th parameter access faulty For evaluation see swParameter[11].ErrorNo
0x800	12th parameter access faulty For evaluation see swParameter[12].ErrorNo
0x1000	13th parameter access faulty For evaluation see swParameter[13].ErrorNo
0x2000	14th parameter access faulty For evaluation see swParameter[14].ErrorNo
0x4000	15th parameter access faulty For evaluation see swParameter[15].ErrorNo
0x8000	16th parameter access faulty For evaluation see swParameter[16].ErrorNo

Note

If the parameter ErrorID[2] contains e.g. the value 0x0003, this means that both the first and the second parameter access is faulted.

5.2.8.11 Connection to the LAcycCom library

Connection to the LAcycCom library

Note

LAcycCom libraries for SIMATIC S7-1200/S7-1500 facilitate collision-free coordination of communication resources in the CPU for acyclic communication using DPV1 services. For this purpose, in the application, instead of the system functions, the corresponding functions in these libraries are used to communicate with external devices.

Note

The LAcycCom library can be accessed at the following SIOS link:

(https://support.industry.siemens.com/cs/ww/en/view/109479553)

Note

For use within the LacycCom environment, function block "LacycCom_ResourceManager", global data block "LacycCom_RequestBuffer" and the PLC variables and PLC data types available in the library are required.

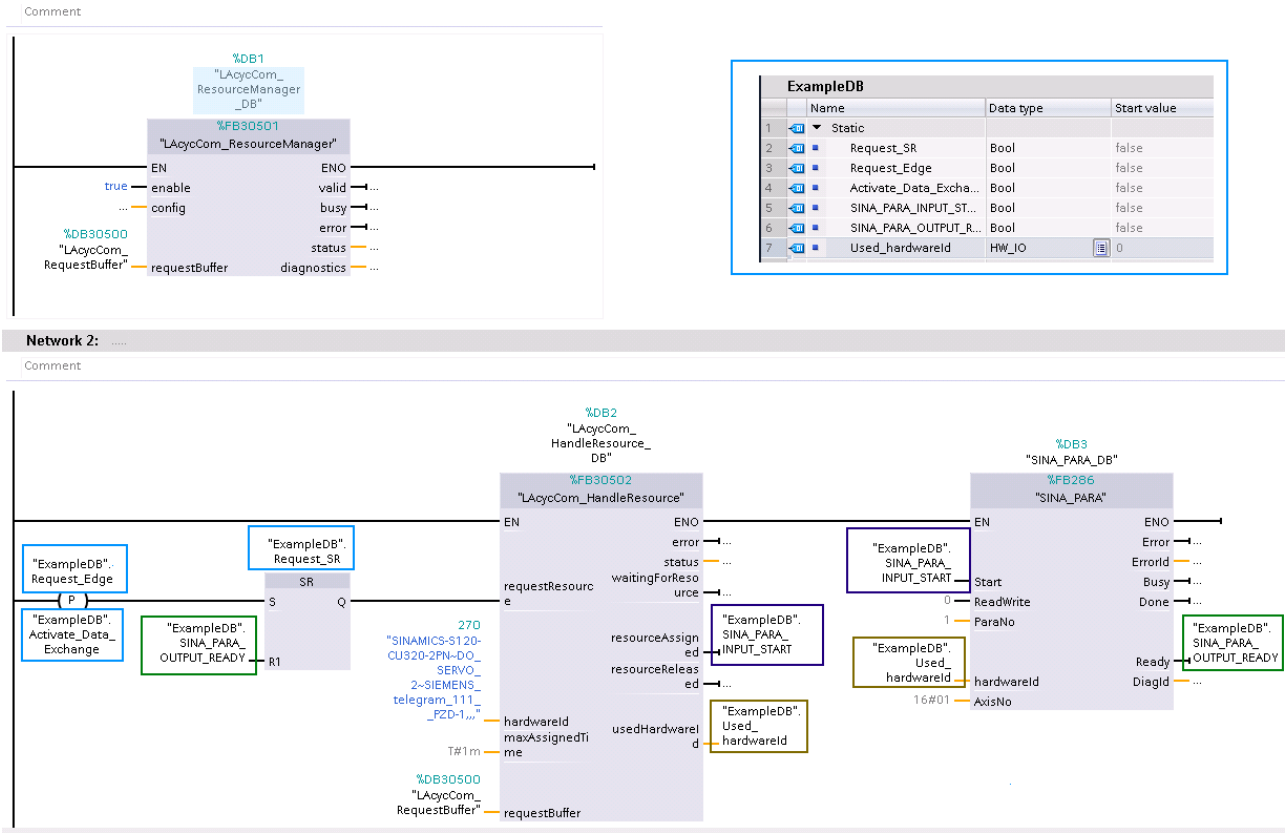


Figure 5-11 Connection to the LAcycCom library

Blocks SINA_PARA and SINA_PARA_S are connected in conjunction with the "LacycCom_HandleResource" block.

The acyclic communication job is transferred to the HandleResource block, and after the release (by the ResourceManager) this controls block SINA_PARA.

After the job has been completed, block SINA_PARA communicates this to the HandleResource block via the Ready output (for one cycle). This can now release the resource again.

To reliably evaluate the start and enable signals, an edge evaluation is used for the start command as well as a memory element (SR flip flop).

Note

Block SINA_PARA_S is connected in the same way.

5.2.9 Function block SINA_PARA_S (FB287)

5.2.9.1 Description

Function block SINA_PARA_S (FB287)

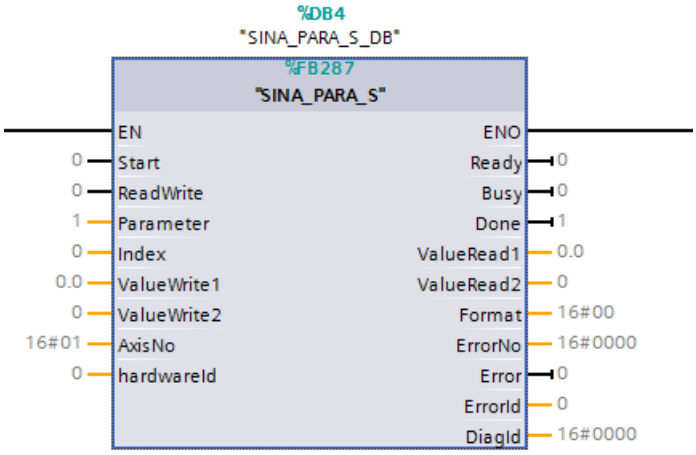


Figure 5-12 FB SINA_PARA_S 1200/1500 CPU

Description

The corresponding instance data block is automatically created when the SINA_PARA_S (FB287) is integrated.

Can be used in the following CPUs: S7-1200/1500

5.2.9.2 Calling OBs

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

5.2.9.3 Description of functions

Function description - general

With the function block, 1 parameter can be read or written acyclically to a SINAMICS S/G drive.

Note

The data is accessed by means of data set 47 according to the PROFIdrive profile. Data set 47 enables global acyclic access to the drive. For this a drive object number (AxisNo) is required.

At the input "ReadWrite" it is specified whether the parameter is to be written to or read from the SINAMICS drive.

The reading or the writing of the parameters is initiated by the edge-triggered input "Start".

5.2.9.4 Input interface of SINA_PARA_S

Input interface of SINA_PARA_S (FB287)

Input signal	Type	Default	Meaning
Start	BOOL	0	**Start of the job (0 = no job or cancel job; 1 = start and execute job)**
ReadWrite	BOOL	0	Job type 0=read, 1=write
Parameter	UINT	1	Parameter number
Index	UINT	0	Index of the parameter
ValueWrite1	REAL	0.0	Value of the parameter in REAL format
ValueWrite2	DINT	0	Value of the parameter in DINT format
AxisNo	BYTE	16#01	Axis number/axis ID/DO number in multi-axis system
hardwareId	HW IO	0	Hardware ID of the module access point/actual value telegram slot of the axis or drive, see Selection of the right hardware submodules (Page 24)

5.2.9.5 Output interface of SINA_PARA_S

Output interface of FB287

Output signal	Type	Default	Meaning
Ready (block S7-1200/1500)	BOOL	0	Checkback signal for connecting in LacycCom environment; 1 = job ended or job canceled (one cycle long)
Busy	BOOL	0	Job in progress if "Busy"=1
Done	BOOL	0	Job completed without errors means edge change from 0 → 1

Output signal	Type	Default	Meaning
ValueRead1	REAL	0.0	Value of the read parameter (REAL format)
ValueRead2	DINT	0	Value of the read parameter (DINT format)
Format	BYTE	16#00	Format of the read parameter
ErrorNo	WORD	16#0000	Error number according to PROFIdrive profile[1)]
Error	BOOL	0	Active group error → "Error" =1
ErrorId	DWORD	0	1st word: Binary-coded indicating which parameter access is faulted 2nd word: type of fault
DiagId	WORD	0	Expanded communication error → SFB call error

1) You can find further information in the SINAMICS S120 Function Manual Communication in the section "Error values in parameter responses" on the internet: https://support.industry.siemens.com/cs/ww/en/view/109771803 (https://support.industry.siemens.com/cs/ww/en/view/109771803)

5.2.9.6 Use of the various parameter inputs and outputs

Use of the various parameter inputs and outputs

Note

TIA Portal/Startdrive V14

From TIA Portal/Startdrive V14 and higher, the input area of SINA_PARA_S contains two different inputs and/or outputs in the REAL and DINT formats (new!).

From this version, all parameters, type DWORD or DINT must be written from this version to field ValueWrite2. This block logic has been changed such that when automatically identifying the DWORD/DINT formats, the job field ValueWrite2 is used for writing or ValueRead2 for reading.

For all other parameters, just as before, the already existing ValueWrite1 or ValueRead1 field is used.

Note

Older versions

In contrast to older versions, beginning with this block version V4.x, the user must know whether the format of the parameter to be read/written involves DWORD/DINT or the remainder (byte, word, real, INT, etc.).

If this is not observed, problems can occur, especially when writing, because the default value of the DINT field ("0") is transferred here instead of the desired value (which was incorrectly entered in the REAL field).

Likewise, the evaluation of read operations for parameters in the DWORD/DINT format must be carried out via the new job field.

Note

The parameter structure for the symbolic programming used is also compatible for older programs of the TIA Portal versions V12SP1 or V13SPx.

With the aid of the new job field, it is now possible to read/write BICO parameter interconnections without problems.

5.2.9.7 Writing parameters

Writing parameters

The "Write" action initially means that the parameter value at the inputs ValueWrite1 and ValueWrite2 is accepted. After the parameter format has been successfully read, the appropriate job field is transferred to the SINAMICS drive.

While this is taking place, the "Busy" bit is set to "1".

If the parameter to be written is erroneous, the associated parameter error number is also read and entered at the output "ErrorNo". At the same time, the corresponding error bit in the first word of the double word "ErrorID" is set.

A successful write process is ended with the edge change "1→0" of the "Busy" bit and an edge change "0→1" of the "Done" bit. The "Error" bit must **not** be set during this. If this happens, the double word "ErrorID" must be evaluated.

5.2.9.8 Reading parameters

Reading parameters

The "Read" action initially means that the parameter at the input parameter is read, and the drive displays the appropriate value at the ValueRead1 or ValueRead2 output.

While this is taking place, the "Busy" bit is set to "1".

If the parameter to be read is erroneous, the associated parameter error number is also output. At the same time, the corresponding error bit in the first word of the double word "ErrorID" is set.

A successful read process is ended with the edge change "1→0" of the "Busy" bit and an edge change "0→1" of the "Done" bit. The "Error" bit must NOT be set during this. If this happens, the double word "ErrorID" must be evaluated.

5.2.9.9 Troubleshooting function block SINA_PARA_S

Troubleshooting function block SINA_PARA_S

The errors that temporarily occurred during the communication with the SINAMICS drive are identified and lead to the required action being repeated.

Note

The siErrorCount (current count) and siMaxErrCount parameters are listed in the instance data block. The siMaxErrCount can be edited by the user and specifies the maximum number of attempts to repeat the job when temporary errors occur (default 12500).

Error = 1 is then set and the status is set.

- During an active SFB fault, the group error "Error = 1" is set and there is an output in the first word of the "ErrorID" and the output "DiagID". Errors due to the SFB calls must not be acknowledged. Once these faults have been eliminated and a new job is started, the outputs "DiagID", "Error" and "ErrorID" are taken back.
- If an incorrect value is entered at the input "ParaNo", this value is not taken into consideration, the group error "Error" is set, and the parameterization error is displayed in the output "ErrorID".
- The group error "Error" is also set if a "Request" error occurs. For this error, the job is carried out, but one or more parameter accesses were not possible. The errors that occurred due to the access are binary coded and displayed in the second word of the double word "ErrorID". The job is also displayed as completed with "Done" = 1.

Evaluation of the output status

ErrorID	
ErrorID[1]	ErrorID[2]

Status[1]	Meaning
0x000	No fault active
0x001	Internal telegram error active
0x002	Parameterization error active
0x003	SFB call error active
0x004	Cancelation of the job during the active data transfer by resetting the Start input to "0"
0x005	Unknown data type detected Evaluation of the ErrorID[2] shows the parameter with unknown data type in the highest value bit

Status[2]	Meaning
0x00	No fault during parameter access
0x01	1st parameter access faulty For evaluation see swParameter[1].ErrorNo

5.2.10 Function block SINA_INFEED (FB288)

5.2.10.1 Description of functions

Function description - general

The hardware ID of the setpoint slot is specified via the input "HWIDSTW" and that of the actual value slot is specified via the input "HWIDZSW".

The infeed can be precharged by setting the input "EnablePrecharging" (STW1.0) and enabled via the input "EnableInfeed" (STW1.3) (by setting the corresponding control bits in STW1).

Note

The functions are only carried out if the infeed is in the status required for this (evaluation of the current ZSW1).

The individual checkback signals (relevant status bits) of the infeed and the complete status word 1 are output via outputs of the block.

Besides the inputs "EnablePrecharging", "EnableInfeed" and "AckError", the user can also make further specifications in control word 1 via the parameter "ConfigAxis" (standard: 3h). For immediate operation, specific bits are preset in the telegram by means of this input.

The "Control request" bit (STW1.10) is cyclically set within the block.

5.2.10.2 Calling OBs

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

5.2.10.3 Description

Function block SINA_INFEED (FB288)

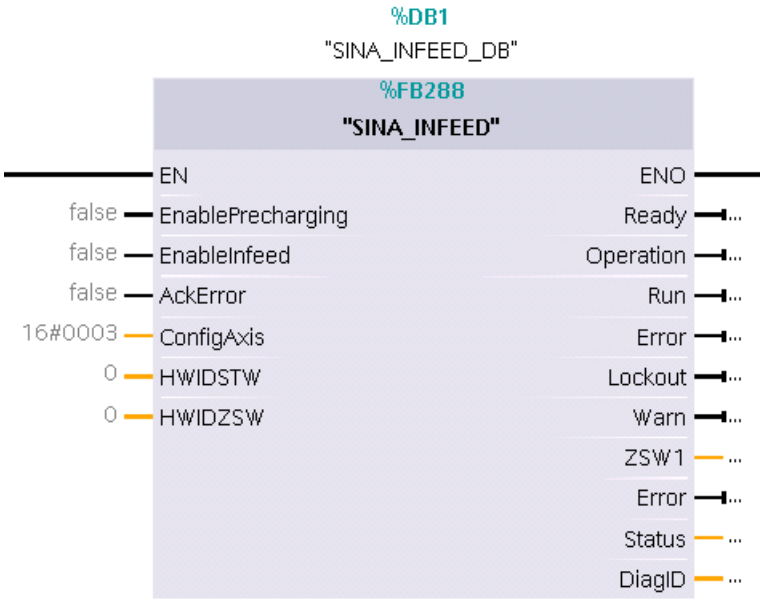


Figure 5-13 FB SINA_INFEED

Description

The block is used to control the infeed unit of a SINAMICS S120. The block is only used for the control word STW1 and evaluates the status word ZSW1 of the infeed (standard telegram 370).

The corresponding instance DB is automatically created when the SINA_INFEED (FB288) is integrated.

Can be used in the following CPUs: S7-1200/1500

5.2.10.4 Called blocks

Called blocks

DPRD_DAT	Read consistent data of a DP standard slave
DPWR_DAT	Write consistent data of a DP standard slave

5.2.10.5 Input interface of SINA_INFEED

Input interface of SINA_INFEED

Input signal	Type	Default	Meaning
EN	BOOL	1	
EnablePrecharging	BOOL	0	Precharge infeed
EnableInfeed	BOOL	0	Switch on infeed
AckError	BOOL	0	Acknowledgement error infeed
ConfigAxis	WORD	16#0003	Acknowledgement error infeed, see Pre-assignment of the Config-Axis input (Page 129).
HWIDSTW	HW_IO	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the set-point slot (**S**et**P**oint), see Selection of the correct hardware sub-module (Page 24)
HWIDZSW	HW_IO	0	Symbolic name or HW ID on the SIMATIC S7-1200/1500 of the actual value slot (**A**ctual **V**alue), see Selection of the correct hardware submodules (Page 24)

5.2.10.6 Pre-assignment of the ConfigAxis input

Pre-assignment of the ConfigAxis input

ConfigAxis	Meaning	PZD	Interconnection in the drive	Default
Bit0	OFF2	1	r2090.1 = p 844[0]	1
Bit1				0
Bit2	1 = disable motor operation	1	r2090.5 = p 3532	0
Bit3	1 = disable generator operation	1	r2090.6 = p 3533	0
Bit4	Reserve - can be used as required (bit 2)	1	r2090.2	0
Bit5	Reserve - can be used as required (bit 4)	1	r2090.4	0
Bit6	Reserve - can be used as required (bit 8)	1	r2090.8	0
Bit7	Reserve - can be used as required (bit 9)	1	r2090.9	0
Bit8	Reserve - can be used as required (bit 11)	1	r2090.11	0
Bit9	Reserve - can be used as required (bit 12)	1	r2090.12	0
Bit10	Reserve - can be used as required (bit 13)	1	r2091.13	0
Bit11	Reserve - can be used as required (bit 14)	1	r2091.14	0
Bit12	Reserve - can be used as required (bit 15)	1	r2091.5	0
Bit13				0
Bit14				0
Bit15				0

5.2.10.7 Output interface of SINA_INFEED

Output interface of SINA_INFEED

Output signal	Type	Default	Meaning
ENO	BOOL	1	
Ready	BOOL	1	Ready for switching on (ZSW1.0)
Operation	BOOL	0	Ready for operation (ZSW1.1)
Run	BOOL	0	Running (ZSW1.2)
Fault	BOOL	0	Error infeed (ZSW1.3)
Lockout	BOOL	0	Infeed blocked (ZSW1.6)
Warning	BOOL	0	Warning infeed (ZSW1.7)
ZSW1	WORD	16#0	Status word 1
Error	BOOL	0	Error
DiagID	WORD	0	Expanded communication error RET_VAL from the system functions DPRD_DAT or DPWR_DAT (see also "Status" parameter)
Status	WORD	16#0	16#7002: No error active 16#7200: Warning infeed 16#8400: Error pre-charging 16#8401: Error infeed 16#8600: Error: DPRD_DAT 16#8601: Error: DPWR_DAT

5.2.10.8 Error handling of the function block SINA_INFEED

Error handling of the function block SINA_INFEED

The output "Error" signals a general error, which is specified more precisely via the output "Status".

If the inputs "EnablePrecharging" and "EnableInfeed" are set and the drive reports a fault, the control bits for the pre-charging and enabling are reset.

If the input "EnableInfeed" is set and "EnablePrecharging" is not set, the output "Error" = 1 and Status = 16#8400 is set. If the input "EnablePrecharging" is then reset to 1, the output "Error" is immediately reset to 0 (no acknowledgment necessary).

Communication between SIMATIC, CPU and infeed takes place via the system blocks "DPRD_DAT" and "DPWR_DAT".

If an error occurs during the processing of the system blocks, the output "Error" is set to 1 and the error message of the system function is output via the output "DiagID".

The output "Status" is set to 16#8600 (DPRD_DAT) or to 16#8601 (DPWR_DAT) depending on which system reports the error.

If an error is active for both system functions, the error message of the DPRD_DAT block is output first and if this is no longer active, the error message of the DPWR_DAT is output, if it is still active after this.

An infeed fault is displayed via the output "Fault" = 1 and "Status" = 16#8401 and can be acknowledged via the input "AckError".

An infeed warning is displayed via the output "Warning" = 1 and "Status" = 16#7200.

If the block operates error-free, "Status" = 16#7002 is displayed at the output.

Note

Input "AckError"

The input "AckError" must be reset by the user because the error acknowledgment is expecting an edge change (0->1).

5.2.11 Interlocking of blocks with acyclic communication

Description

Because the acyclic communications link between the S7-CPU and the drive can only be used by one application at a time, several applications must be locked from one another using acyclic communication, as shown in the following example.

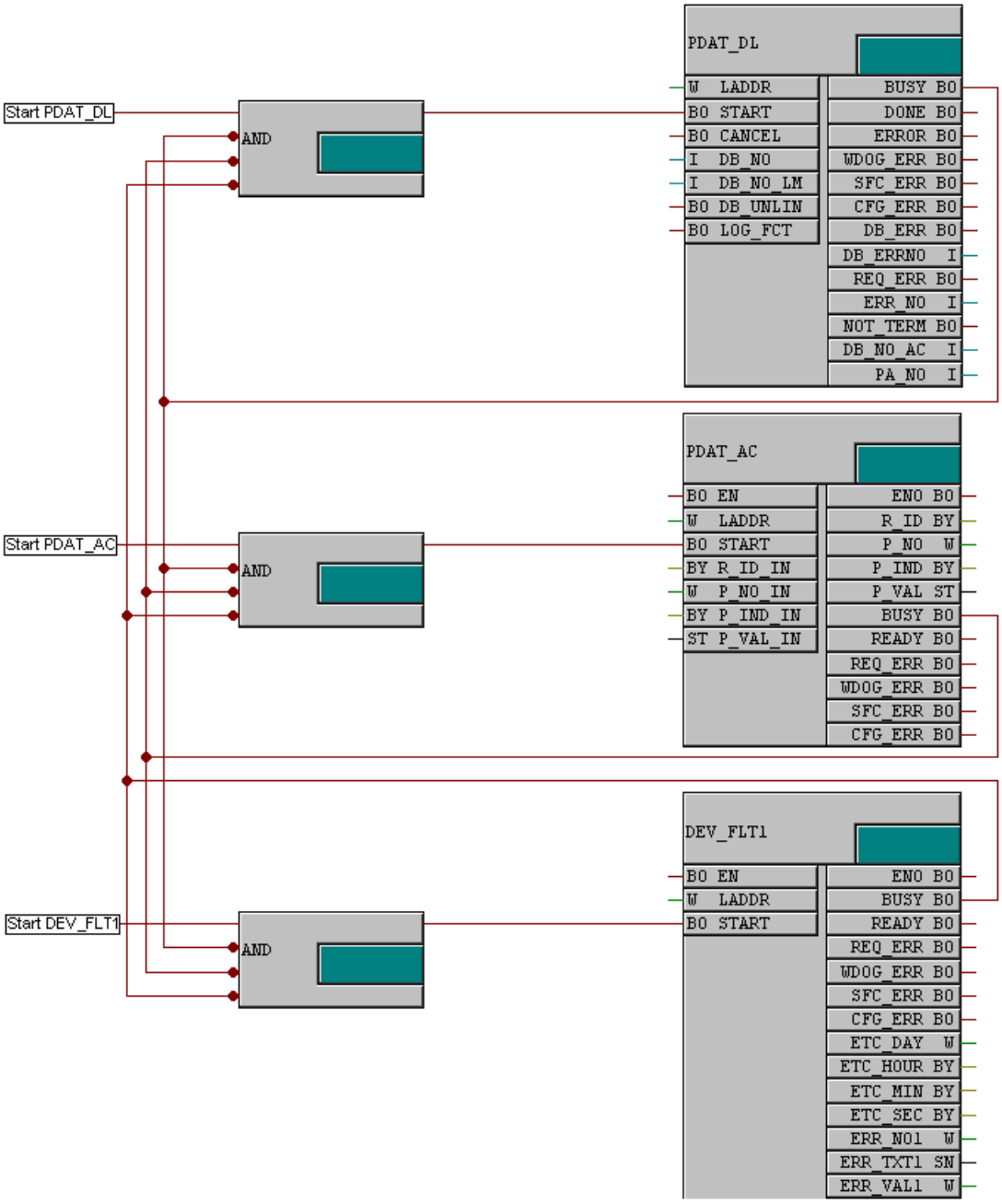


Figure 5-14 Interlocking of blocks with acyclic communication

Note

Reset start bit with BUSY

The start bit must be reset once the output BUSY of the enabled block is "true". Otherwise, the corresponding block will re-initiate again and again (BUSY has the value "false" when a job has ended).

5.3 S7-1500 data blocks

5.3.1 DB DRIVDBx: Configuration data of the drives S7-1500

General procedure

DB DRIVDBx

The block number can be changed.

Can be used in the following CPUs: S7-1500

Description

For the transfer of the configuration data to the communication blocks, a data block DB DRIVDBx (x = consecutive number of the configuration DB) must be provided. All values must be entered in hexadecimal format.

The following table describes the data set of the configuration data:

Parameter		Declaration	Data type	Description
SLAVE_n		STAT	STRUCT	Slave-specific configuration data slave n
	DADDR	STAT	HW_IO	Diagnostics addresses of the slave (from HW configuration)
	DPADDR	STAT	WORD	PROFIBUS address/device number of the slave (from HW configuration, is not evaluated by the blocks)
	SLOT_m	STAT	SLOT_UDT	Slot-specific configuration data Slot m (UDT 31, see below)
	SLOT_m+y	STAT	SLOT_UDT	Slot-specific configuration data slot m + y (UDT 31, see below)
...				
SLAVE_n+1		STAT	STRUCT	Slave-specific configuration data slave n + 1
	DADDR	STAT	HW_IO	Diagnostics addresses of the slave (from HW configuration)
	DPADDR	STAT	WORD	PROFIBUS address/device number of the slave (from HW configuration, is not evaluated by the blocks)
	SLOT_m	STAT	SLOT_UDT	Slot-specific configuration data slot m (UDT 31, see below)
	SLOT_m+y	STAT	SLOT_UDT	Slot-specific configuration data slot m + y (UDT 31, see below)
...				
SLAVE_n+x		STAT	SLOT_UDT	Slave-specific configuration data slave n + x
	DADDR	STAT	HW_IO	Diagnostics addresses of the slave (from HW configuration)

Parameter		Declaration	Data type	Description
	DPADDR	STAT	WORD	PROFIBUS address/device number of the slave (from HW configuration, is not evaluated by the blocks)
	SLOT_m	STAT	SLOT_UDT	Slot-specific configuration data slot m (UDT 31, see below)
	SLOT_m+y	STAT	SLOT_UDT	Slot-specific configuration data slot m + y (UDT 31, see below)

The data block uses the user-defined data type SLOT_UDT:

- UDT 31: (data type for slot-specific configuration data)
  The number of the UDT must **not** be changed!

The following table describes the SLOT_UDT data type:

Parameter		Data type	Description
①	LADDR	HW_IO	HW identifier (identifier of the drive)
②	SLOT_ID	BYTE	ID for slot type (indirect from HW Configuration > Drive > Type) 0 = Slot not assigned 1 = Setpoint slot 2 = Actual value slot 3 = Combined setpoint/actual value slot 4 = PIV slot
③	PCD_ADDR [1]	BYTE	PCD_ADDR: Process data offset address for setpoint or/and actual value slot (from HWCN Configuration > Drive properties > Cyclic data exchange > Actual value or Setpoint > Start address ; value range 1 ... 10h).
④	LENGTH [1]	BYTE	Length of the slot data to be transferred in words (from HW Configuration > Length; value range 1 ... 10h).
	CONSIST [1]	BOOL	Consistency (from HW Configuration > Consistency): • 0 (FALSE) = unit • 1 (TRUE) = total length The consistency is set in the "Setpoint" and "Actual value" screen forms. This parameter is not relevant for PROFINET IO. It must be specified, but the settings will not take effect.
[1] Irrelevant with slot ID = 3, 4			

The number of slaves whose configuration data can be stored in a DRIVDBx depends on the structure of the slaves (number of slots per slave) and on the CPU used.

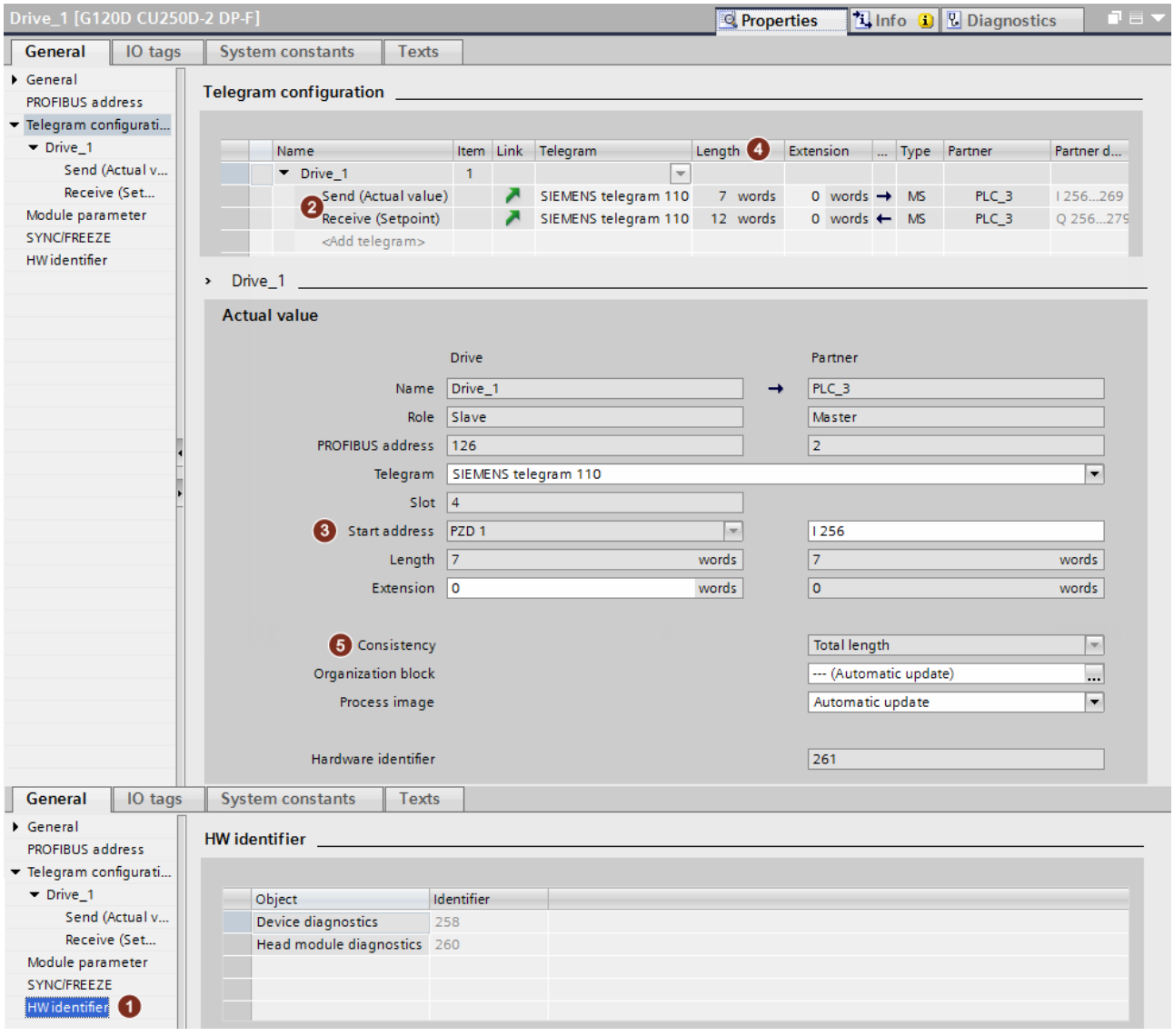


Figure 5-15 DP slave properties

5.4 Functions

5.4.1 FC COM_STAT: Reporting a communication failure

Description

FC 60

The block number can be changed.

Can be used in the following CPUs: S7-1200 and S7-1500

Calling OBs

The block can be alternatively installed in the following OBs:

- Cyclic task: OB1
- Cyclic interrupt OB: e.g. OB32

Description of functions

With the aid of the system function SFC 51 RDSYSST, the block evaluates the system status list of the CPU and reports whether the slave to be processed is faulted or disabled. It is called up once for each slave that is to be monitored.

If the slave can no longer be addressed by the master, this is reported by the "COM_FLT" bit.

Slaves can be enabled or disabled by means of SFC12 while a system is running. When a slave is disabled, no error messages occur, no LEDs illuminate, and repeated telegrams are avoided. The "disabled" status is detected by the block and is signaled in the "DEACTIV" bit.

Note

The system function SFC12 is not available for an S7-1200.

I/O bar

The following table shows the input and output parameters:

Parameter	Declaration	Data type	Memory area	Description
Input				
LADDR	INPUT	HW_DPSLAVE	E, A, M, D, L, const.	Diagnostics address of the slave
Output				
COM_FLT	OUTPUT	BOOL	E, A, M, D, L	DP slave failed
DEACTIV	OUTPUT	BOOL	E, A, M, D, L	DP slave deactivated

Parameter	Declaration	Data type	Memory area	Description
SFC_ERR	OUTPUT	BOOL	E, A, M, D, L	SFC 51 RDSSYST reports error
SFC_FLT	OUTPUT	INT	E, A, M, D, L	Return value of the SFC 51 RDSSYST. Example: 16#8090 (specified LADDR is invalid)

Block call (STL source code)

```
CALL   COM_STAT(
          LADDR     := 16#...,   //Diagnoseadresse,
          COM_FLT   := M30.0,
          DEACTIV   := M30.1,
          SFC_ERR   := M30.2,
          SFC_FLT   := MW32
```

5.5 Using the blocks under PROFINET IO

5.5.1 Overview

Description

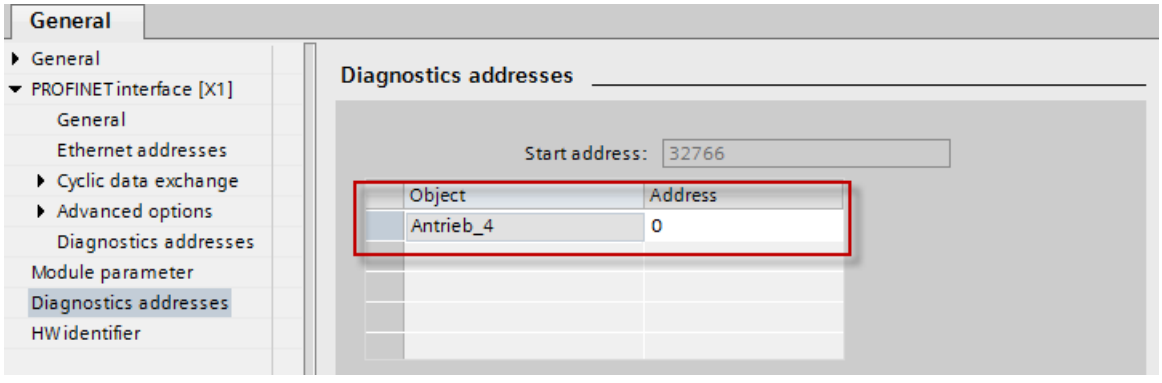
The blocks described in the previous chapters can also be used under PROFINET IO.

5.5.2 Continued use of the S7 program for PROFIBUS with PROFINET IO

Description

To change an existing PROFIBUS configuration to PROFINET IO with having to change the S7 program, the following points must be observed.

- The existing S7 program must be backed up; especially the DrivDBx(s) with the configuration of the drive slaves
- The existing DrivDBx from the PROFIBUS configuration must not be overwritten by a DB that was newly created based on the PROFINET configuration. In this case, the old DrivDBx-DB continues to be used without changes.
- Make a note of the PROFIBUS configuration (diagnostics, IO address, and telegram) of the drives
- Delete the drive slaves on PROFIBUS or remove the DP master system
- Insert the PROFINET IO system
- Insert the drives in the PROFINET IO system and configure them
- The diagnostics address from the PROFIBUS configuration (can also be found in the DrivDBx for the corresponding slave) must be entered in the telegram with the red border



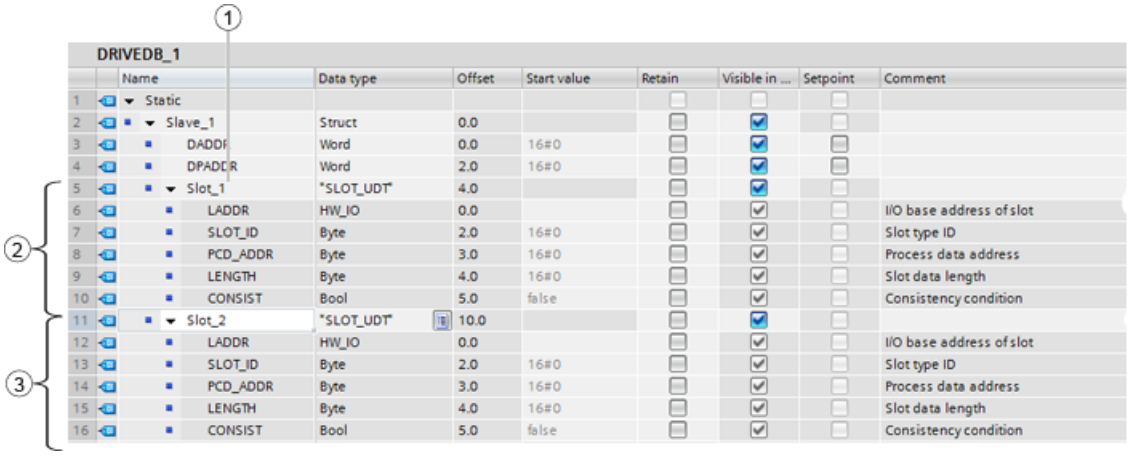
- The IO address and the choice of telegram must be identical to the previous PROFIBUS configuration.
- The parameter "Consistency" has no meaning for PROFINET IO. The blocks contain the parameter, but the settings have no effect.

5.5.3 Migration to PROFINET IO

Description

The configuration of blocks for applications with a PROFINET IO system is described below.

- Creating a PROFINET IO configuration
- Saving the HW-Config
- The DrivDBx for the PROFINET IO system must either be migrated from Classic or created manually.
- Programming the user program using the blocks described in the previous chapters.
- Supplying the blocks with the data from DrivDBx-DB:



DRIVEDB_1

	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint	Comment
1	Static							
2	Slave_1	Struct	0.0					
3	DADDR	Word	0.0	16#0				
4	DPADDR	Word	2.0	16#0				
5	Slot_1	"SLOT_UDT"	4.0					
6	LADDR	HW_IO	0.0					I/O base address of slot
7	SLOT_ID	Byte	2.0	16#0				Slot type ID
8	PCD_ADDR	Byte	3.0	16#0				Process data address
9	LENGTH	Byte	4.0	16#0				Slot data length
10	CONSIST	Bool	5.0	false				Consistency condition
11	Slot_2	"SLOT_UDT"	10.0					
12	LADDR	HW_IO	0.0					I/O base address of slot
13	SLOT_ID	Byte	2.0	16#0				Slot type ID
14	PCD_ADDR	Byte	3.0	16#0				Process data address
15	LENGTH	Byte	4.0	16#0				Slot data length
16	CONSIST	Bool	5.0	false				Consistency condition

① Diagnostics address for the blocks with acyclic communication
② Data for CFG_DATA input for FB32 (PCD_RECV)
③ Data for CFG_DATA input for FB31 (PCD_SEND)

- If you want to continue using an existing PROFIBUS program for PROFINET IO, for which the DrivDBx was newly created manually based on the PROFINET IO configuration, only the pointers to the SLOT-UDTs must be adapted as described. The automatic generation of the DrivDBx is not available in the TIA Portal.
- Loading the program into the CPU

Diagnostics

6

6.1 Diagnostics of the drive

The PCD_RECV block evaluates the "group fault" bit in the status word of the drive (bit 3) and reports a drive fault in the parameter DEV_FLT (to do this, the status word must be configured as Word 1 in the telegram from the drive to the master!). The user can then read the fault buffer of the drive in order to get more detailed information about the cause(s) of the fault.

The complete current fault (fault number, fault text, fault value if applicable) SINAMICS can be read from the fault buffer with the aid of the DEV_FLT4 block.

6.2 DP diagnostics

The slaves can be monitored for station failure or deactivation using the COM_STAT function. If the displays are COM_FLT or DEACTIV = TRUE, the receive data cannot be imported.

6.3 S7 system diagnostics

SIMATIC S7 reports the following slave errors:

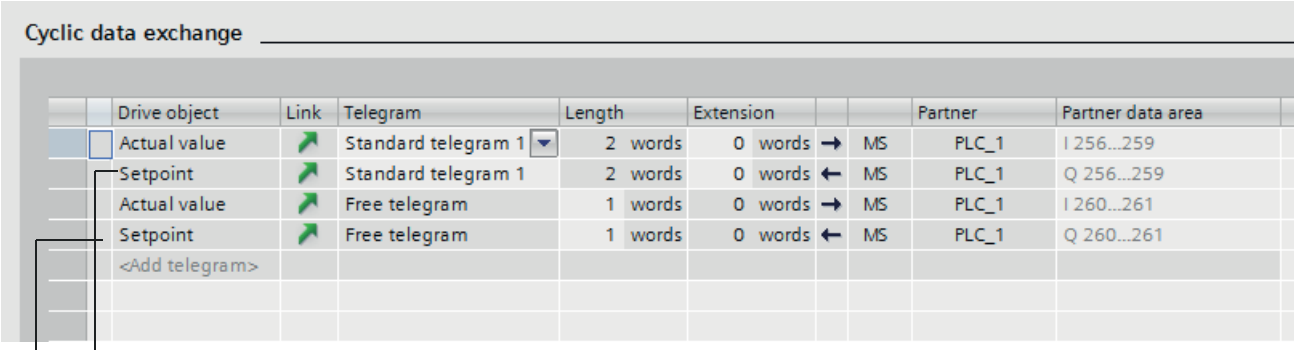
- When a DP station fails, the OB86 (rack failure) is called
- When there is a peripheral access error, the OB122 is called.

Thus, a targeted response to the error can be programmed in these organization blocks. In each case, the corresponding block must at least be available "empty" in the PLC. Otherwise the CPU goes into STOP status in the event of an error.

Appendix

A

A.1 Parameterization of the PCD_SEND with several setpoint slots



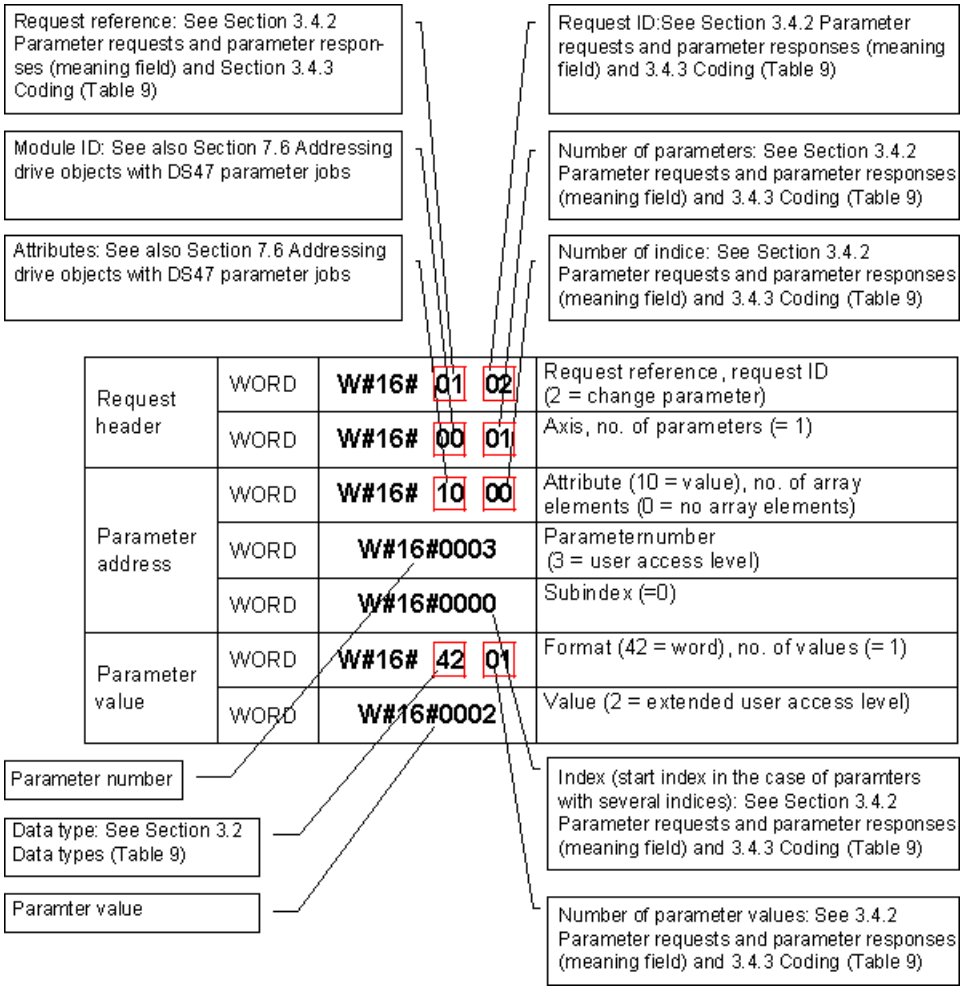

```
CALL PCD_SEND, DB_PCD_SEND(
     CFG_DATA := DRIVDB1.SLAVE_1.SLOT_5,
     PCD_1    := MW0, //PZD 1
     PCD_2    := MW2, //PZD 2
     .....
     SFC_ERR  := M30.0,
     CFG_ERR  := M30.1

CALL PCD_SEND, DB_PCD_SEND(
     CFG_DATA := DRIVDB1.SLAVE_1.SLOT_6,
     PCD_1    := MW0, //Data is not transferred to the drive
     PCD_2    := MW2, //Data is not transferred to the drive
     PCD_3    := MW4, //PZD 3
     PCD_4    := MW6, //PZD 4
     PCD_5    := MW8, //PZD 5
     .....
     SFC_ERR  := M30.0,
     CFG_ERR  := M30.1
```

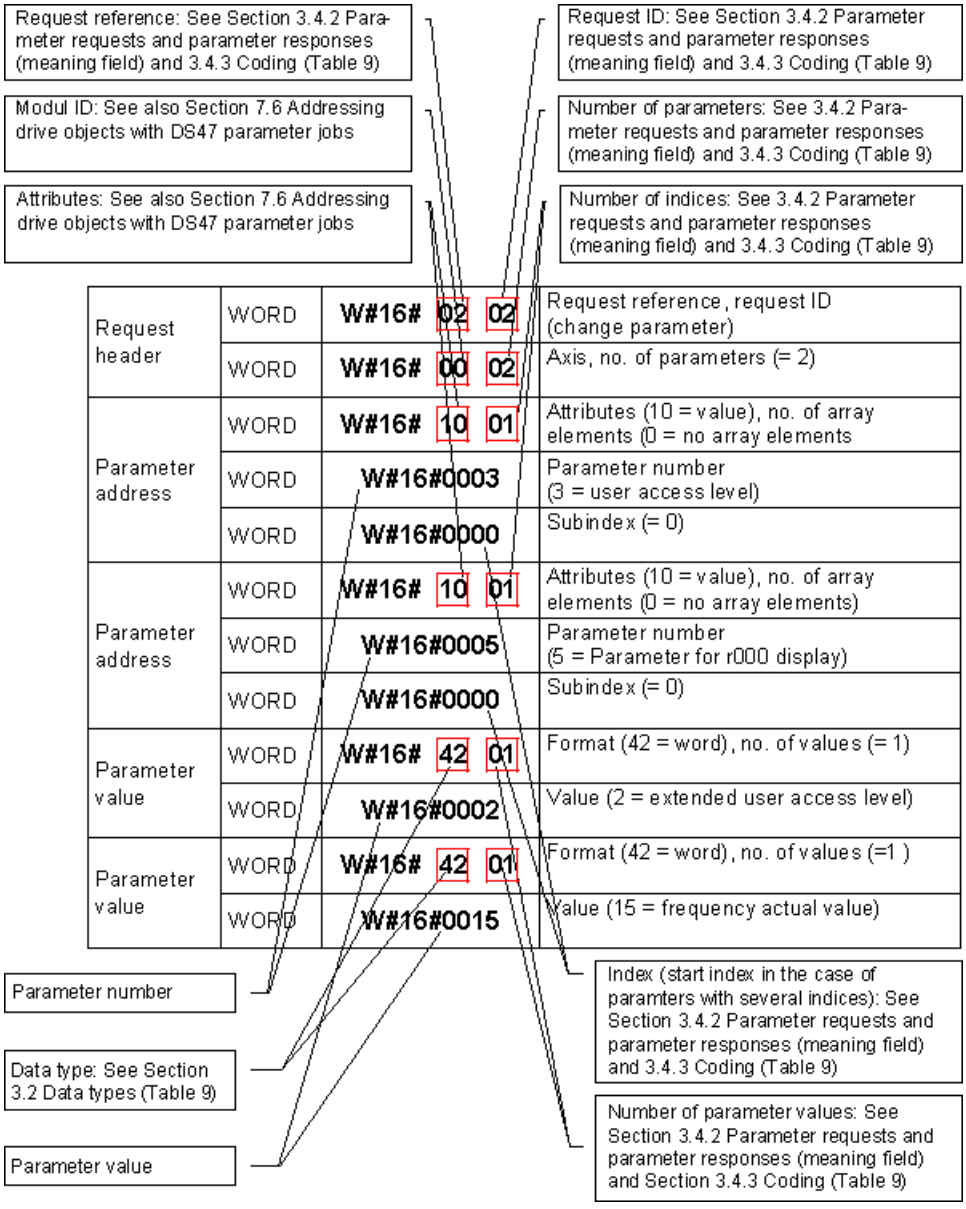
A.2 Formulating parameter jobs (data set 47)

The following tables should serve as a brief explanation of data set 47 jobs. For detailed information, see "Profile Drive Technology PROFIdrive V4.1 May 2006".

Single-parameter job (change parameter value, single)



Multi-parameter job (change parameter value, multi-parameter)



PROFIBUS Profile Drive Technology

The description "Parameter model (Page 179)" is an extract from "Profile Drive Technology PROFIdrive V4.1 May 2006", with the kind permission of the PROFIBUS user organization in Germany.

A.3 Structure of the parameter job DB for FB PDAT_UD2

A.3.1 Example for complete DB transfer

DBB0	Version ID1 converter	Main version	**Start address**
DBB1		Subversion	
DBB2	Version ID2 converter	Service pack	
DBB3		Hotfix	
Bit 4.0	Control information	= 1: ES relevant; =0 ES not relevant	
Bit 4.1		= 1: ES relevant; =0 ES not relevant	
Bit 4.2		No significance	
Bit 4.3		No significance	
Bit 4.4		No significance	
Bit 4.5		No significance	
Bit 4.6		No significance	
Bit 4.7		No significance	
Bit 5.0		No significance	
Bit 5.1		No significance	
Bit 5.2		No significance	
Bit 5.3		No significance	
Bit 5.4		No significance	
Bit 5.5		No significance	
Bit 5.6		=1 Accept comparison value (DBB34)/wait time DBB35) for parameter in EEPROM stored in DBW34	
Bit 5.7		=1 Predecessor DB exists; =0 Predecessor DB does not exist	
DBW6	Detailed information on DBx4.0	Parameter value for ES system (-1, if not used)	
DBW8	Detailed information on DBx4.1	Parameter value for ES system (-1, if not used)	
DBW10	Detailed information on DBx4.2	No significance	
DBW12	Detailed information on DBx4.3	No significance	
DBW14	Detailed information on DBx4.4	No significance	
DBW16	Detailed information on DBx4.5	No significance	
DBW18	Detailed information on DBx4.6	No significance	
DBW20	Detailed information on DBx4.7	No significance	
DBW22	Detailed information on DBx5.0	No significance	
DBW24	Detailed information on DBx5.1	No significance	
DBW26	Detailed information on DBx5.2	No significance	
DBW28	Detailed information on DBx5.3	No significance	

DBW30	Detailed information on DBx5.4	No significance	
DBW32	Detailed information on DBx5.5	No significance	
DBB34	Detailed information on DBx5.6	Comparison value for parameter accepted in EEPROM;	
DBB35		Wait time until parameter must be accepted in EEPROM at the latest; if not, then cancelation with error message	
DBW36	Detailed information on DBx5.7	No. of the predecessor DB	
DBW38	Reserve 1		
	1st job		
DBW40	AA2F	ID DS47	**Separator between two jobs, must not be changed**
DBW42	AA2F	ID DS47	
DBW44	Job ID[1)]	See below for definition	
DBB46	Job reference		
DBB47	Job identifier		
DBB48	Axis		
DBB49	Number of parameters		
DBB50	Attribute		
DBB51	Number of elements		
DBW52	Parameter number		
DBW54	Subindex		
DBB56	Format		
DBB57	Number of values		
DBW58	Value		
	2nd job ...		
DBW60	AA2F		
DBW62	AA2F		
...	...		
	nth job		
DBWn	AA2F	ID DS47	**Separator between two jobs, must not be changed**
DBW(n+2)	AA2F	ID DS47	
DBW(n+4)	Job ID[1)]	See below for definition	
DBB(n+6)	Job reference		
DBB(n+7)	Job identifier		
DBB(n+8)	Axis		
DBB(n+9)	Number of parameters		
DBB(n+10)	Attribute		
DBB(n+11)	Number of elements		

DBW(n+12)	Parameter number		
DBW(n+14)	Subindex		
DBB(n+16)	Format		
DBB(n+17)	Number of values		
DBW(n+18)	Value		
DBW(n+20)	EEEE	End ID	
DBW(n+22)	EEEE	End ID	
DBW(n+24)	No. following DB[2)]		

[1)] See table "Meaning of the job ID"

[2)] If no further data block follows, "0000" must be entered here.

A.3.2 Example for partial DB transfer

DBB0	Version ID1 converter	Main version	**Beginning of the first sub-job (start address)**
DBB1		Subversion	
DBB2	Version ID2 converter	Service pack	
DBB3		Hotfix	
Bit 4.0	Control information	= 1: ES relevant; =0 ES not relevant	
Bit 4.1		= 1: ES relevant; =0 ES not relevant	
Bit 4.2		No significance	
Bit 4.3		No significance	
Bit 4.4		No significance	
Bit 4.5		No significance	
Bit 4.6		No significance	
Bit 4.7		No significance	
Bit 5.0		No significance	
Bit 5.1		No significance	
Bit 5.2		No significance	
Bit 5.3		No significance	
Bit 5.4		No significance	
Bit 5.5		No significance	
Bit 5.6		=1 Accept comparison value (DBB34)/wait time DBB35) for parameter in EEPROM stored in DBW34	
Bit 5.7		=1 Predecessor DB exists; =0 Predecessor DB does not exist	
DBW6	Detailed information on DBx4.0	Parameter value for ES system (-1, if not used)	

DBW8	Detailed information on DBx4.1	Parameter value for ES system (-1, if not used)	
DBW10	Detailed information on DBx4.2	No significance	
DBW12	Detailed information on DBx4.3	No significance	
DBW14	Detailed information on DBx4.4	No significance	
DBW16	Detailed information on DBx4.5	No significance	
DBW18	Detailed information on DBx4.6	No significance	
DBW20	Detailed information on DBx4.7	No significance	
DBW22	Detailed information on DBx5.0	No significance	
DBW24	Detailed information on DBx5.1	No significance	
DBW26	Detailed information on DBx5.2	No significance	
DBW28	Detailed information on DBx5.3	No significance	
DBW30	Detailed information on DBx5.4	No significance	
DBW32	Detailed information on DBx5.5	No significance	
DBB34	Detailed information on DBx5.6	Comparison value for parameter accepted in EE-PROM;	
DBB35		Wait time until parameter must be accepted in EE-PROM at the latest; if not, then cancelation with error message	
DBW36	Detailed information on DBx5.7	No. of the predecessor DB	
DBW38	Reserve 1		
	1st job		
DBW40	AA2F	ID DS47	**Separator between two jobs, must not be changed**
DBW42	AA2F	ID DS47	
DBW44	Job ID[1)]	See below for definition	
DBB46	Job reference		
DBB47	Job identifier		
DBB48	Axis		
DBB49	Number of parameters		
DBB50	Attribute		
DBB51	Number of elements		
DBW52	Parameter number		
DBW54	Subindex		
DBB56	Format		
DBB57	Number of values		
DBW58	Value		
	2nd job ...		
DBW60	AA2F		**Separator between two jobs, must not be changed**
DBW62	AA2F		

...	...		
	nth job		
DBWn	AA2F	ID DS47	**Separator between two jobs, must not be changed**
DBW(n+2)	AA2F	ID DS47	
DBW(n+4)	Job ID[1)]	See below for definition	
DBB(n+6)	Job reference		
DBB(n+7)	Job identifier		
DBB(n+8)	Axis		
DBB(n+9)	Number of parameters		
DBB(n+10)	Attribute		
DBB(n+11)	Number of elements		
DBW(n+12)	Parameter number		
DBW(n+14)	Subindex		
DBB(n+16)	Format		
DBB(n+17)	Number of values		
DBW(n+18)	Value		
DBW(n+20)	EEEE	End ID	
DBW(n+22)	EEEE	End ID	
DBW(n+24)	No. following DB[2)]		**End of the first sub-job**
DBB(n+26)	Version ID1 converter	Main version	**Beginning of the second sub-job (start address)**
DBB(n+27)		Subversion	
DBB(n+28)	Version ID2 converter	Service pack	
DBB(n+29)		Hotfix	

Bit(n+30).0	Control information	= 1: ES relevant; =0 ES not relevant	
Bit(n+30).1		= 1: ES relevant; =0 ES not relevant	
Bit(n+30).2		No significance	
Bit(n+30).3		No significance	
Bit(n+30).4		No significance	
Bit(n+30).5		No significance	
Bit(n+30).6		No significance	
Bit(n+30).7		No significance	
Bit(n+31).0		No significance	
Bit(n+31).1		No significance	
Bit(n+31).2		No significance	
Bit(n+31).3		No significance	
Bit(n+31).4		No significance	
Bit(n+31).5		No significance	
Bit(n+31).6		=1 Accept comparison value (DBB34)/wait time DBB35) for parameter in EEPROM stored in DBW34	
Bit(n+31).7		=1 Predecessor DB exists; =0 Predecessor DB does not exist	
DBW(n+32)	Detailed information on DBx4.0	Parameter value for ES system (-1, if not used)	
DBW(n+34)	Detailed information on DBx4.1	Parameter value for ES system (-1, if not used)	
DBW(n+36)	Detailed information on DBx4.2	No significance	
DBW(n+38)	Detailed information on DBx4.3	No significance	
DBW(n+40)	Detailed information on DBx4.4	No significance	
DBW(n+42)	Detailed information on DBx4.5	No significance	
DBW(n+44)	Detailed information on DBx4.6	No significance	
DBW(n+46)	Detailed information on DBx4.7	No significance	
DBW(n+48)	Detailed information on DBx5.0	No significance	
DBW(n+50)	Detailed information on DBx5.1	No significance	
DBW(n+52)	Detailed information on DBx5.2	No significance	
DBW(n+54)	Detailed information on DBx5.3	No significance	
DBW(n+56)	Detailed information on DBx5.4	No significance	
DBW(n+58)	Detailed information on DBx5.5	No significance	
DBB(n+60)	Detailed information on DBx5.6	Comparison value for parameter accepted in EE-PROM;	
DBB(n+61)		Wait time until parameter must be accepted in EE-PROM at the latest; if not, then cancelation with error message	
DBW(n+62)	Detailed information on DBx5.7	No. of the predecessor DB	
DBW(n+64)	Reserve 1		
	1st job		

DBW(n+66)	AA2F	ID DS47	**Separator between two jobs, must not be changed**
DBW(n+68)	AA2F	ID DS47	
DBW(n+70)	Job ID[1)]	See below for definition	
DBB(n+72)	Job reference		
DBB(n+73)	Job identifier		
DBB(n+74)	Axis		
DBB(n+75)	Number of parameters		
DBB(n+76)	Attribute		
DBB(n+77)	Number of elements		
DBW(n+78)	Parameter number		
DBW(n+80)	Subindex		
DBB(n+82)	Format		
DBB(n+83)	Number of values		
DBW(n+84)	Value		
	2nd job ...		
DBW(n+86)	AA2F		**Separator between two jobs, must not be changed**
DBW(n+88)	AA2F		
...	...		
	mth job		
DBWm	AA2F	ID DS47	**Separator between two jobs, must not be changed**
DBW(m+2)	AA2F	ID DS47	
DBW(m+4)	Job ID[1)]	See below for definition	
DBB(m+6)	Job reference		
DBB(m+7)	Job identifier		
DBB(m+8)	Axis		
DBB(m+9)	Number of parameters		
DBB(m+10)	Attribute		
DBB(m+11)	Number of elements		
DBW(m+12)	Parameter number		
DBW(m+14)	Subindex		
DBB(m+16)	Format		
DBB(m+17)	Number of values		
DBW(m+18)	Value		
DBW(m+20)	EEEE	End ID	

DBW(m+22)	EEEE	End ID	
DBW(m+24)	No. following DB[2)]		**End of the first sub-job**

1) See table "Meaning of the job ID"

2) If no further data block follows, "0000" must be entered here.

Table A-1 Meaning of the job ID

Job ID	Meaning
0	"Normal" job (job is implemented without restrictions, both as read and as write job)
1	Upload-preparing job (required to read parameters from the drive). Jobs with this ID are always executed as write jobs and are never converted into a read job.
2	Read job "System of units (ES) of the target devices" (required to compare ES of target and source) If the ES of the target device differs from the ES of the source, no download takes place and an error is output. If the upload functionality is selected, the ES of the target device is transferred to the DB.
3	Download-preparing job (required to change parameters in the drive) Jobs with this ID are located before "normal" parameter jobs in the DB. They are only implemented in the download direction and skipped in the upload direction.
4	Download-concluding job (concludes a write job) Jobs with this ID are located at the end of the "normal" parameter jobs in the DB. They are only implemented in the download direction and skipped in the upload direction
5	Restore the factory settings Jobs with this ID are only implemented in the download direction and are skipped in the upload direction.
6	Save the parameters in EEPROM/Compact Flash Jobs with this ID are only implemented in the download direction and are skipped in the upload direction. At least two jobs are required in each case. The first job to trigger saving in the EEPROM and the second job to check if saving in the EERPOM was successful (if this is supported by the drive).

Table A-2 Meaning of the control information bits and storage location of the detailed information

Bit	Meaning
4.0	= 1: ES relevant
4.1	= 1: ES relevant
4.2	No significance
4.3	No significance
4.4	No significance
4.5	No significance
4.6	No significance
4.7	No significance
5.0	No significance
5.1	No significance

Bit	Meaning
5.2	No significance
5.3	No significance
5.4	No significance
5.5	No significance
5.6	=1 Accept comparison value (DBB34)/wait time DBB35) for parameter in EEPROM stored in DBW34
5.7	=1 Predecessor DB exists; =0 Predecessor DB does not exist; DB no. stored in DBW36.

Arrangement of the job types in the parameter DB

1. **Upload-preparing jobs:** Required to place the drive in the upload status and to read out the system of units of the target device (if relevant for the drive).
2. **Jobs to read out the system of units of the target device** (if relevant for the drive): The read value is compared to the corresponding value of the source parameter set. A download is only implemented if the ES between target and source agrees. In the event of an upload, the ES of the target system is stored in the parameter DB (see table above).
3. **Download-preparing jobs:** These jobs place the drive into the download status.
4. **"Normal" jobs:** These jobs are executed both in the download and upload direction. In the upload direction, all write jobs are converted into read jobs.
5. **Download-concluding jobs:** These jobs are required to place the drive into the status it was in before the download.
6. **Jobs to save the parameters in the EEPROM/Compact Flash:** Two jobs are required for this purpose. The first one is a write job that triggers the storage of the data in the EEPROM. The second one is a read job to the same parameter to determine if the parameters were successfully stored in the EEPROM. The comparison value for the parameter and the wait time until a message may be output indicating that saving failed are stored in the detailed information on the control information bits.

A.3.3 Note on transferring the parameter value 16#AA2F / 16#AA2FAA2F to the converter

It can occur that the parameter value 16#AA2F is not transferred to the converter. The reason for this is that the ID 16# AA2FAA2F is defined as separator between two jobs. Below it is described how the parameter value 16#AA2F can cause this problem and how to avoid this problem, if necessary.

Table A-3 Use of a data type

Case	Data type	Value	Solution
1	8-bit	Last word in the range parameter value = 16#AA2F	Enter another word with the value 0 between the last value and the separator
2	8-bit	Two back-to-back words in the range parameter value = 16#AA2F	Divide into two jobs and additionally enter another word with the value 0 between the last value and the separator
3	16-bit	Last word in the range parameter value = 16#AA2F	Enter another word with the value 0 between the last value and the separator
4	16-bit	Two back-to-back words in the range parameter value = 16#AA2F	Divide into two jobs and additionally enter another word with the value 0 between the last value and the separator
5	32-bit	Last word in the range parameter value = 16#AA2F	Enter another word with the value 0 between the last value and the separator
6	32-bit	Two back-to-back words in the range parameter value = 16#AA2F	This combination cannot be transferred. The only remedy would be to change the value

Combination of the data types above:

- It must be considered that if the value 16#AA2F is transferred as last value, then it must be separated from the separator by a word with the value 0.
- If in the range of the parameter values the value 16#AA2F is written to two back-to-back words, then it may be possible to prevent that the value 16#AA2FAA2F is created by changing the parameter sequence or by dividing the job into two jobs. The previous issue, however, must continue to be considered.

A.4 Example of download data block

A.4.1 Data block with DS47 jobs

Structure of DS47 jobs see Formulating parameter jobs (data set 47) (Page 146).

Name	Data type	Offset	Start value	Retain	Visible in...	Comment
DATA_0000	Byte	0.0	B#16#44	☑	☑	68
DATA_0001	Byte	1.0	B#16#13	☑	☑	19
DATA_0002	Byte	2.0	0	☑	☑	00
DATA_0003	Byte	3.0	0	☑	☑	00
DATA_0004	Bool	4.0	false	☑	☑	=0: ES not relevant
DATA_0005	Bool	4.1	false	☑	☑	=0: ES not relevant
DATA_0006	Bool	4.2	false	☑	☑	No significance
DATA_0007	Bool	4.3	false	☑	☑	No significance
DATA_0008	Bool	4.4	false	☑	☑	No significance
DATA_0009	Bool	4.5	false	☑	☑	No significance
DATA_0010	Bool	4.6	false	☑	☑	No significance
DATA_0011	Bool	4.7	false	☑	☑	No significance
DATA_0012	Bool	5.0	false	☑	☑	No significance
DATA_0013	Bool	5.1	false	☑	☑	No significance
DATA_0014	Bool	5.2	false	☑	☑	No significance
DATA_0015	Bool	5.3	false	☑	☑	No significance
DATA_0016	Bool	5.4	false	☑	☑	No significance
DATA_0017	Bool	5.5	TRUE	☑	☑	Parameter download mode activated
DATA_0018	Bool	5.6	false	☑	☑	Save to RAM
DATA_0019	Bool	5.7	false	☑	☑	Previous DB does not exist
DATA_0020	Word	6.0	W#16#FFFF	☑	☑	Parameter value for ES (-1, if not used)
DATA_0021	Word	8.0	W#16#FFFF	☑	☑	Parameter value for ES (-1, if not used)
DATA_0022	Word	10.0	0	☑	☑	No significance
DATA_0023	Word	12.0	0	☑	☑	No significance
DATA_0024	Word	14.0	0	☑	☑	No significance
DATA_0025	Word	16.0	0	☑	☑	No significance
DATA_0026	Word	18.0	0	☑	☑	No significance
DATA_0027	Word	20.0	0	☑	☑	No significance
DATA_0028	Word	22.0	0	☑	☑	No significance
DATA_0029	Word	24.0	0	☑	☑	No significance
DATA_0030	Word	26.0	0	☑	☑	No significance
DATA_0031	Word	28.0	0	☑	☑	No significance
DATA_0032	Word	30.0	0	☑	☑	No significance
DATA_0033	Byte	32.0	B#16#0A	☑	☑	Comparison values for detection of parameter
DATA_0034	Byte	33.0	B#16#78	☑	☑	Wait time until parameter download mode is...
DATA_0035	Byte	34.0	0	☑	☑	Comparison value for Transfer parameter to EEP...
DATA_0036	Byte	35.0	0	☑	☑	Delay time for Transfer parameter to EEPROM
DATA_0037	Word	36.0	0	☑	☑	Number of the previous DB (0 = no previous DB)
DATA_0038	Word	38.0	0	☑	☑	Reserved
DATA_0039	Word	40.0	W#16#AA2F	☑	☑	DS47 ID

+40.0	DATA_0038	WORD	W#16#AA2F	DS47 ID
+42.0	DATA_0039	WORD	W#16#AA2F	DS47 ID
+44.0	DATA_0040	INT	3	Job ID = 3: DL preparing job
+46.0	DATA_0041	WORD	W#16#102	Job reference, job ID (write parameter)
+48.0	DATA_0042	WORD	W#16#3	Axis, Number of parameters (= 3)
+50.0	DATA_0043	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+52.0	DATA_0044	WORD	W#16#3	Parameter no. (p3 = User access level)
+54.0	DATA_0045	WORD	W#16#0	Subindex (= 0)
+56.0	DATA_0046	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+58.0	DATA_0047	WORD	W#16#F6E	Parameter no. (p3950 = Access of hidden parameter)
+60.0	DATA_0048	WORD	W#16#0	Subindex (= 0)
+62.0	DATA_0049	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+64.0	DATA_0050	WORD	W#16#A	Parameter no. (p10 = Commissioning parameter filter)
+66.0	DATA_0051	WORD	W#16#0	Subindex (= 0)
+68.0	DATA_0052	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+70.0	DATA_0053	INT	4	Value (4)
+72.0	DATA_0054	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+74.0	DATA_0055	INT	46	Value (46)
+76.0	DATA_0056	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+78.0	DATA_0057	INT	29	Value (29)
+80.0	DATA_0058	WORD	W#16#AA2F	DS47 ID
+82.0	DATA_0059	WORD	W#16#AA2F	DS47 ID
+84.0	DATA_0060	INT	0	Job ID = 0: Normal job
+86.0	DATA_0061	WORD	W#16#202	Job reference, job ID (write parameter)
+88.0	DATA_0062	WORD	W#16#1	Axis, Number of parameters (= 1)
+90.0	DATA_0063	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+92.0	DATA_0064	WORD	W#16#5	Parameter no. (p5 = Display selection)
+94.0	DATA_0065	WORD	W#16#0	Subindex (= 0)
+96.0	DATA_0066	WORD	W#16#601	Format (6 = WORD), Number of values (= 1)
+98.0	DATA_0067	WORD	W#16#15	Value (21)
+100.0	DATA_0068	WORD	W#16#AA2F	DS47 ID
+102.0	DATA_0069	WORD	W#16#AA2F	DS47 ID
+104.0	DATA_0070	INT	0	Job ID = 0: Normal job
+106.0	DATA_0071	WORD	W#16#202	Job reference, job ID (write parameter)
+108.0	DATA_0072	WORD	W#16#2	Axis, Number of parameters (= 2)
+110.0	DATA_0073	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+112.0	DATA_0074	WORD	W#16#2BD	Parameter no. (p701 = Selection digital input1)
+114.0	DATA_0075	WORD	W#16#0	Subindex (= 0)
+116.0	DATA_0076	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+118.0	DATA_0077	WORD	W#16#2BE	Parameter no. (p702 = Selection digital input2)
+120.0	DATA_0078	WORD	W#16#0	Subindex (= 0)
+122.0	DATA_0079	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+124.0	DATA_0080	INT	1	Value (1)
+126.0	DATA_0081	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+128.0	DATA_0082	INT	12	Value (12)

A1: Download-preparing jobs to place the converter (here MICROMASTER 4xx) in download mode

A2: Set display P5 to "Actual frequency"

A3: Multi-parameter job:

- Interconnect digital input 1 to "ON/OFF"
- Interconnect digital input 2 to "Reversing"

The two words with the value "W#16#AA2F" between the parameter jobs (e.g. A1 and A2) are an indicator for the download block to be able to differentiate between the individual jobs. They are not required for recognizing the data set and must not be changed.

+130.0	DATA_0083	WORD	W#16#AA2F	DS47 ID
+132.0	DATA_0084	WORD	W#16#AA2F	DS47 ID
+134.0	DATA_0085	INT	4	Job ID = 4: DL completing job
+136.0	DATA_0086	WORD	W#16#302	Job reference, job ID (write parameter)
+138.0	DATA_0087	WORD	W#16#4	Axis, Number of parameters (= 4)
+140.0	DATA_0088	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+142.0	DATA_0089	WORD	W#16#154	Parameter no. (p340 = Calc motor model and control)
+144.0	DATA_0090	WORD	W#16#0	Subindex (= 0)
+146.0	DATA_0091	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+148.0	DATA_0092	WORD	W#16#A	Parameter no. (p10 = Commissioning parameter filter)
+150.0	DATA_0093	WORD	W#16#0	Subindex (= 0)
+152.0	DATA_0094	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+154.0	DATA_0095	WORD	W#16#3	Parameter no. (p3 = User access level)
+156.0	DATA_0096	WORD	W#16#0	Subindex (= 0)
+158.0	DATA_0097	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+160.0	DATA_0098	WORD	W#16#F6E	Parameter no. (p3950 = Access of hidden parameter)
+162.0	DATA_0099	WORD	W#16#0	Subindex (= 0)
+164.0	DATA_0100	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+166.0	DATA_0101	INT	0	Value (0)
+168.0	DATA_0102	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+170.0	DATA_0103	INT	0	Value (0)
+172.0	DATA_0104	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+174.0	DATA_0105	INT	2	Value (2)
+176.0	DATA_0106	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+178.0	DATA_0107	INT	0	Value (0)
+180.0	DATA_0108	WORD	W#16#AA2F	DS47 ID
+182.0	DATA_0109	WORD	W#16#AA2F	DS47 ID
+184.0	DATA_0110	INT	6	Job ID = 6: Save parameter to EEPROM
+186.0	DATA_0111	WORD	W#16#402	Job reference, job ID (write parameter)
+188.0	DATA_0112	WORD	W#16#1	Axis, Number of parameters (= 1)
+190.0	DATA_0113	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+192.0	DATA_0114	WORD	W#16#3CB	Parameter no. (p971 = Transfer data from RAM to EEPROM)
+194.0	DATA_0115	WORD	W#16#0	Subindex (= 0)
+196.0	DATA_0116	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+198.0	DATA_0117	INT	1	Value (1)
+200.0	DATA_0118	WORD	W#16#AA2F	DS47 ID
+202.0	DATA_0119	WORD	W#16#AA2F	DS47 ID
+204.0	DATA_0120	INT	6	Job ID = 6: Save parameter to EEPROM
+206.0	DATA_0121	WORD	W#16#501	Job reference, job ID (read parameter)
+208.0	DATA_0122	WORD	W#16#1	Axis, Number of parameters (= 1)
+210.0	DATA_0123	WORD	W#16#1001	Attribute (10 = value), Number of elements (= 1)
+212.0	DATA_0124	WORD	W#16#3CB	Parameter no. (p971 = Transfer data from RAM to EEPROM)
+214.0	DATA_0125	WORD	W#16#0	Subindex (= 0)
+216.0	DATA_0126	WORD	W#16#301	Format (3 = INT), Number of values (= 1)
+218.0	DATA_0127	INT	1	Value (1)
+220.0	DATA_0128	WORD	W#16#EEEE	END ID
+222.0	DATA_0129	WORD	W#16#EEEE	END ID
+224.0	DATA_0130	WORD	W#16#0	0 = Number of the following DB (0 - no further DB)
=226.0		END_STRUCT		

A4: Download-concluding jobs to place the converter (here MICROMASTER 4) in operational mode

A5: Save changed parameters in EEPROM

A6: Read job to determine if parameters are stored in the EEPROM

A.5 Addressing of the drive objects for parameter jobs according to data set 47

A.5.1 SINAMICS G120

	Module ID / axis
Single-axis device	1

A.5.2 SINAMICS G130 / G150

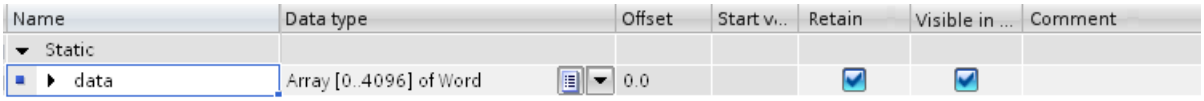
	Module ID / axis
Control unit (CU320)	1
Drive (vector)	2
Customer terminal strip TM31 (-A60)	3
Customer terminal strip TM31 (-A61)	4

A.5.3 SINAMICS S

The module ID of the individual drive objects can be derived from the CU parameter 101 (or 978).

A.6 Tip

Example for generating an 8192-byte copy DB in the main memory or DB in the load memory when using the download function block FB40 or the upload/download function blocks FB41 / FB42.



A.7 Examples of acyclic communication with SINA_PARA (FB286)

A.7.1 Copy RAM to ROM

Copy RAM to ROM

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].srValue	p977 0 1	The Control Unit must be selected as hardware ID.
ReadWrite	1	Select write process
Start	1	Start of the job

A.7.2 Absolute encoder adjustment

Absolute encoder adjustment

Note

Execute the steps one after the other

The steps for adjusting the absolute encoder must be carried out sequentially, i.e. one after the other! For this reason, only the first structure [1] is used in the data block.

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].sdValue	p2599 0 xxxx[LU]	Selection of the hardware ID of the axis Write the reference coordinate in [LU]
ReadWrite	1	Select write process
Start	1	Start of the job

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].srValue	p2507 0 2	Adapt the index if an encoder other than the motor encoder is used.
ReadWrite	1	Select write process
Start	1	Start of the job

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].srValue	p977 0 1	The Control Unit must be selected as hardware ID.
ReadWrite	1	Select write process
Start	1	Start of the job

A.7.3 Writing the acceleration/deceleration ramp of the ramp-function generator

Writing the acceleration/deceleration ramp of the ramp-function generator

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].srValue sParameter[2].siParaNo sParameter[2].siIndex sParameter[2].srValue	p1120 0 xxxx[s] p1121 0 xxxx[s]	Selection of the hardware ID of the drive Adapt the index if you cange the values of another data set.
ReadWrite	1	Select write process
Start	1	Start of the job

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].srValue	p977 0 1	The Control Unit must be selected as hardware ID.
ReadWrite	1	Select write process
Start	1	Start of the job

A.7.4 Jog speed/incremental path

Writing the jog speeds

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].sdValue sParameter[2].siParaNo sParameter[2].siIndex sParameter[2].sdValue	p2585 0 xxxx[1000*LU/min] p2586 0 xxxx[1000*LU/min]	The axis must be selected as hardware ID.
ReadWrite	1	Select write process
Start	1	Start of the job

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].sdValue sParameter[2].siParaNo sParameter[2].siIndex sParameter[2].sdValue	p2587 0 xxxx[LU] p2588 0 xxxx[LU]	The axis must be selected as hardware ID.
ReadWrite	1	Select write process
Start	1	Start of the job

Structure parameter	Data set information	Note
sParameter[1].siParaNo sParameter[1].siIndex sParameter[1].srValue	p977 0 1	The Control Unit must be selected as hardware ID.
ReadWrite	1	Select write process
Start	1	Start of the job

A.7.5 Reading the current fault buffer

Reading the current fault buffer

Structure parameter	Data set information	Note
sParameter[1].siParaNo	r945	The Control Unit must be selected as hardware ID.
sParameter[1].siIndex	0	
sParameter[1].srValue	xxxx	
sParameter[2].siParaNo	r945	
sParameter[2].siIndex	1	
sParameter[2].srValue	xxxx	
sParameter[3].siParaNo	r945	
sParameter[3].siIndex	2	
sParameter[3].srValue	xxxx	
sParameter[4].siParaNo	r945	
sParameter[4].siIndex	3	
sParameter[4].srValue	xxxx	
sParameter[5].siParaNo	r945	
sParameter[5].siIndex	4	
sParameter[5].srValue	xxxx	
sParameter[6].siParaNo	r945	
sParameter[6].siIndex	5	
sParameter[6].srValue	xxxx	
sParameter[7].siParaNo	r945	
sParameter[7].siIndex	6	
sParameter[7].srValue	xxxx	
sParameter[8].siParaNo	r945	
sParameter[8].siIndex	7	
sParameter[8].srValue	xxxx	
sParameter[9].siParaNo	r949	
sParameter[9].siIndex	0	
sParameter[9].srValue	xxxx	
sParameter[10].siParaNo	r949	
sParameter[10].siIndex	1	
sParameter[10].srValue	xxxx	
sParameter[11].siParaNo	r949	
sParameter[11].siIndex	2	
sParameter[11].srValue	xxxx	
sParameter[12].siParaNo	r949	
sParameter[12].siIndex	3	
sParameter[12].srValue	xxxx	
sParameter[13].siParaNo	r949	
sParameter[13].siIndex	4	

Structure parameter	Data set information	Note
sParameter[13].srValue	xxxx	
sParameter[14].siParaNo	r949	
sParameter[14].siIndex	5	
sParameter[14].srValue	xxxx	
sParameter[15].siParaNo	r949	
sParameter[15].siIndex	6	
sParameter[15].srValue	xxxx	
sParameter[16].siParaNo	r949	
sParameter[16].siIndex	7	
sParameter[16].srValue	xxxx	
ReadWrite	0	Select read process
Start	1	Start of the job

Note

The results of the job are stored in the respective parameter of the sxParameter[x]srValue structure.

A.8 EPOS telegram 111

EPOS telegram 111

PZD	Assignment of the process data
PZD1	Control word 1
PZD2	EPosSTW 1
PZD3	EPosSTW 2
PZD4	Control word 2
PZD5	Velocity override for all operating modes effective (4000HEX = 100%)
PZD6	Position setpoint in [LU] for direct setpoint specification/MDI operating mode
PZD7	
PZD8	Velocity setpoint in MDI operating mode
PZD9	
PZD10	Acceleration override for direct setpoint specification/MDI operating mode
PZD11	Deceleration override for direct setpoint specification/MDI operating mode
PZD12	Reserve

Assignment of control word 1

Bit	Abbr.	Designation (description of the HIGH level)	Drive parameters	Function diagram
0	Off1	ON command: 0 = OFF1 active; 1 = ON	p840	2501
1	Off2	0 =: OFF2 active 1 = signal: operating condition **No** coasting active	p844	2501
2	Off3	0 = OFF3 active 1 = operating condition **no** quick stop active	p848	2501
3	Enc	Enable of inverter	p852	2501

Bit	Abbr.	Designation (description of the HIGH level)	Drive parameters	Function diagram
4	RejTrvTsk	Traversing blocks and direct setpoint specification/MDI Reject traversing task 0 = active traversing command is rejected / axis brakes with 100% deceleration override 1 = Do not reject traversing task (axis can be traversed)	p2640	3616
5	IntMStop	Intermediate STOP traversing blocks and MDI/direct setpoint specification intermediate stop 0 = active traversing command is interrupted/axis brakes with specified delay override 1 = no intermediate stop (axis can be traversed)	p2640	3616
6	**TrvStart**	**Activate traversing job** **Setpoint acceptance edge if MdiTyp = 0**	**p2631** **p2650**	**3640** **3620**
7	AckFault	Acknowledgement of fault	p2103	2501
8	Jog1	Jog signal source 1	p2589	3610
9	Jog2	Jog signal source 2	p2590	3610
10	LB	Life bit (control request from PLC)	p854	2501
11	RefStart	Start homing	p2595	3612
12	Bit12	Reserved		
13	Bit13	External block change (0→1)	<not used> (p2633)	
14	Bit14	Reserved		
15	Bit15	Reserved		

Assignment of EPosSTW 1

Bit	Abbr.	Name	Drive parameter	Function diagram
	TrvBit0	Block selection bit 0	p2625	3640
1	TrvBit1	Block selection bit 1	p2626	3640
2	TrvBit2	Block selection bit 2	p2627	3640
3	TrvBit3	Block selection bit 3	p2628	3640
4	TrvBit4	Block selection bit 4	p2629	3640
5	TrvBit5	Block selection bit 5	p2630	3640
6	Bit6	Reserved		
7	Bit7	Reserved		
8	MdiTyp	Positioning type 0 = relative positioning 1 = absolute positioning	p2648	3620
9	MdiPos	Selection of direction for setting up, or absolute positioning of rotary axes, in the positive direction	p2651	3620
10	MdiNeg	Selection of direction for setting up, or absolute positioning of rotary axes, in the negative direction	p2652	3620
11	Bit11	Reserved		
12	MdiTrTyp	Transfer type 0 = Value acceptance through 0 → 1 edge at MdiEdge 1 signal: continuous setpoint acceptance	P2649	3620
13	Bit13	Reserved		

Bit	Abbr.	Name	Drive parameter	Function diagram
14	MdiSetup	Direct setpoint input/MDI – setup selection Selection of MDI setup mode 0 = positioning 1 = set up	p2653	3620
15	MdiStart	Operating mode MDI/direct setpoint specification	p2647	3640

Assignment of EPosSTW 2

Bit	Abbr.	Name	Drive parameters	Function diagram
0	TrkMode	Start follow-up mode	p2655.0	3635
1	SetRefPt	Set home position	p2596	3612
2	ActRefCam	Activate homing cams	p2612	3612
3	Bit3	Activate fixed stop	<not used>	
4	Bit4	Reserved		
5	JogInc	Jog: 0 = continuous traversing 1 = traversing about parameterized path	p2591	3610
6	Bit6	Reserved		
7	Bit7	Reserved		
8	RefTyp	Selection of homing type 0 = reference point approach 1 = flying referencing	p2597	3612
9	RefStDi	Homing procedure start direction 0 = positive start direction 1 = negative start direction	p2604	3612

Bit	Abbr.	Name	Drive parameters	Function diagram
10	RefInpS	Setting the signal source for the selection of the measuring input for flying (passive) homing 0 = measuring input 1 is activated 1 = measuring input 2 is activated	p2510	4010
11	RefEdge	Passive homing: setting of the edge evaluation 0 : Positive edge 1 : Negative edge	p2511	4010
12	Bit12	Reserved		
13	Bit13	Reserved		
14	SftLimAct	Activation of the software limit switch	p2582	3630
15	StpCamAct	Activation of the stop cams	p2568	3630

Assignment of STW2

Bit	Abbr.	Name	Drive parameters	Function diagram
0	DDSBit0	Drive data set bit 0	p820.0	8565
1	DDSBit1	Drive data set bit 1	p821.0	8565
2	DDSBit2	Drive data set bit 2	p822.0	8565
3	DDSBit3	Drive data set bit 3	p823.0	8565
4	DDSBit4	Drive data set bit 4	p824.0	8565
5	GlbStart	Global start	<not used>	
6	ResIComp	Reset I-component of speed controller	<not used>	
7	ActPrkAxis	Activate parking axis	p897	
8	TrvFixedStp	Travel to fixed stop	<not used> (p1545.0)	<not used> (8012)
9	GlbTrgCom	Global trigger command	<not used>	
10	Bit10	Reserved		
11	MotSwOver	Motor switchover completed (0->1)	p828.0	8575
12	MsZykBit0	Master sign-of-life bit 0	<not used>	

Bit	Abbr.	Name	Drive parameters	Function diagram
13	MsZykBit1	Master sign-of-life bit 1	<not used>	
14	MsZykBit2	Master sign-of-life bit 2	<not used>	
15	MsZykBit3	Master sign-of-life bit 3	<not used>	

Setpoint overview

PZD	Abbr.	Setpoint	Parameter	Function diagram
5	OverrideV	Velocity override	p2646	3630
6+7	Position	Position setpoint	p2642	3620
8+9	Velocity	Velocity setpoint	p2643	3618
10	OverrideA	Acceleration override	p2644	3618
11	OverrideD	Deceleration override	p2645	3618
12	Word12	Reserved		

PZD	Assignment of the process data
PZD1	Status word 1
PZD2	EPosZSW 1
PZD3	EPosZSW 2
PZD4	Status word 2
PZD5	MELDW
PZD6	Actual position value [LU]
PZD7	
PZD8	Actual velocity value (refers to reference speed p2000) Note: 40000000HEX = 100%
PZD9	
PZD10	Fault (sending of the active fault number)
PZD11	Alarm (sending of the active alarm number)
PZD12	Reserve

Assignment of status word 1

Bit	Abbr.	Name	Drive parameters	Function diagram
0	RTS	Ready for power-up	r899.0	2503
1	RDY	Ready to operate	r899.1	2503

Bit	Abbr.	Name	Drive parameters	Function diagram
2	IOp	Drive is switched on (condition for the mode selection of the EPOS)	r899.2	2503
3	Fault	Fault active	r2139.3	2548
4	NoOff2Act	OFF2 not activated (partial condition for the switch-on)	r899.4	2503
5	NoOff3Act	OFF3 not activated (partial condition for the switch-on)	r899.5	2503
6	PowInhbt	Switching on inhibited active	r899.6	2503
7	Alarm	Alarm present	r2139.7	2548
8	NoFlwErr	Following error within tolerance	r2684.8	4025
9	LbCr	Control request	r899.9	2503
10	TargPos	Target position reached	r2684.10	4020
11	RefPSet	Home position set	r2684.11	3614
12	TrvTskAck	Traversing block activated acknowledgement	r2684.12	3646
Important note on bit 13: Information dependent on SINAMICS FW version. This bit is used for standstill detection				
13	StndStill	**SINAMICS S/G120 FW ≥ 4.8 / ≥ 4.7.9, V90 PN:** EPOS status word 1: Setpoint stationary	r2683.2	2537
13	StndStill	**SINAMICS S/G120 FW <4.8 / <4.7.9:** Status word monitoring 3: |n_act| < speed threshold value 3	r2199.0	2537
14	Accel	Axis accelerates	r2684.4	3635
15	Decel	Axis decelerates	r2684.5	3635

Assignment of EPosZSW 1

Bit	Abbr.	Name	Drive parameters	Function diagram
0	ActTrvBit0	Active traversing block bit 0	r2670.0	3650
1	ActTrvBit1	Active traversing block bit 1	r2670.1	3650
2	ActTrvBit2	Active traversing block bit 2	r2670.2	3650
3	ActTrvBit3	Active traversing block bit 3	r2670.3	3650

Bit	Abbr.	Name	Drive parameters	Function diagram
4	ActTrvBit4	Active traversing block bit 4	r2670.4	3650
5	ActTrvBit5	Active traversing block bit 5	r2670.5	3650
6	Bit6	Reserved		
6	Bit7	Reserved		
8	StpCamMinAct	STOP cam minus active	r2684.13	3630
9	StpCamPlsAct	STOP cam plus active	r2684.14	3630
10	JogAct	Jog active operating mode	r2094.0 [1)]	2460
11	RefAct	Homing procedure active operating mode	r2094.1 [1)]	2460
12	FlyRefAct	Flying homing active	r2684.1	3630
13	TrvBlAct	Traversing blocks active operating mode	r2094.2 [1)]	2460
14	MdiStupAct	In the direct setpoint specification / MDI operating mode, setup is active	r2094.4 [1)]	2460
15	MdiPosAct	Positioning is active in the direct setpoint specification/MDI operating mode	r2094.3 [1)]	2460

[1)] r2669 (function block diagram 3630) in bit-granular display. For this purpose, p2099[0] = r2699 is connected at the input of the connector-binector converter.

Assignment of EPosZSW 2

Bit	Abbr.	Name	Drive parameters	Function diagram
0	TrkModeAct	Follow-up mode active	r2683.0	3645
1	VeloLimAct	Velocity limitation active	r2683.1	3645
2	SetPStat	Setpoint stationary	r2683.2	3645
3	PrntMrkOut	Registration mark outside outer window	r2684.3	3614
4	FWD	Axis travels forward	r2683.4	3635
5	BWD	Axis travels backward	r2683.5	3635

Bit	Abbr.	Name	Drive parameters	Function diagram
6	SftSwMinAct	Minus software limit switch approached	r2683.6	3635
7	SftSwPlsAct	Plus software limit switch approached	r2683.7	3635
8	PosSmCam1	Actual position value <= cam switching position 1	r2683.8	4025
9	PosSmCam2	Actual position value <= cam switching position 2	r2683.9	4025
10	TrvOut1	Direct output 1 via traversing block	r2683.10	3616
11	TrvOut2	Direct output 2 via traversing block	r2683.11	3616
12	FxStpRd	Fixed stop reached	<not used> (r2683.12)	3645
13	FxStpTrRd	Fixed stop clamping torque reached	<not used> (r2683.13)	3645
14	TrvFxStpAct	Travel to fixed stop active	<not used> (r2683.14)	3645
15	CmdAct	Traversing active	r2683.15	3645

Assignment of status word 2

Bit	Abbr.	Name	Drive parameters	Function diagram
0	ActDDSBit0	Drive data set bit 0	r51.0	8565
1	ActDDSBit1	Drive data set bit 1	r51.1	8565
2	ActDDSBit2	Drive data set bit 2	r51.2	8565
3	ActDDSBit3	Drive data set bit 3	r51.3	8565
4	ActDDSBit4	Drive data set bit 4	r51.4	8565
5	CmdActRelBrk	Open holding brake active	<not used>	
6	TrqContMode	Torque-controlled operation	<not used>	
7	ParkAxisAct	Parking axis selected	r896.0	
8	Bit8	Reserved	r1406.8	
9	GlbTrgReq	Global trigger request	<not used>	
10	PulsEn	Pulses enabled	r899.11	2503
11	MotSwOverAct	Motor data set switchover active	r835.0	8575
12	SlvZykBit0	Slave sign-of-life bit 0	<not used>	

Bit	Abbr.	Name	Drive parameters	Function diagram
13	SlvZykBit1	Slave sign-of-life bit 1	<not used>	
14	SlvZykBit2	Slave sign-of-life bit 2	<not used>	
15	SlvZykBit3	Slave sign-of-life bit 3	<not used>	

Actual value overview

PZD	Abbr.	Actual value	Parameter	Function diagram
5	Word6	Reserved		
6+7	Position	Actual position value	r2521	4010
8+9	Velocity	Actual speed value	r63	4715
10	ErrNr	Error	r2131	8060
11	WarnNr	Alarm	r2132	8065
12	Reserve	Reserved		

A.9 Standard telegram 1

Standard telegram 1

S7 bit representation (drive)	Meaning
STW1 1.0 (bit 0)	OFF1/ON (pulse enable possible)
STW1 1.1 (bit 1)	OFF2/ON (enable possible)
STW1 1.2 (bit 2)	OFF3/ON (enable possible)
STW1 1.3 (bit 3)	Enable or disable operation
STW1 1.4 (bit 4)	Ramp-function generator enable
STW1 1.5 (bit 5)	Continue ramp-function generator
STW1 1.6 (bit 6)	Speed setpoint enable
STW1 1.7 (bit 7)	Acknowledge fault
STW1 0.0 (bit 8)	Reserved
STW1 0.1 (bit 9)	Reserved
STW1 0.2 (bit 10)	Control by PLC
STW1 0.3 (bit 11)	Direction of rotation
STW1 0.4 (bit 12)	Holding brake must be opened
STW1 0.5 (bit 13)	Motorized potentiometer setpoint higher
STW1 0.6 (bit 14)	Motorized potentiometer setpoint lower
STW1 0.7 (bit 15)	Reserved
STW2 (bits 16 to 32)	Speed setpoint

S7 bit representation (drive)	
ZSW1 1.0 (bit 0)	Ready for switching on
ZSW1 1.1 (bit 1)	Ready for operation
ZSW1 1.2 (bit 2)	Operation enabled
ZSW1 1.3 (bit 3)	Fault present
ZSW1 1.4 (bit 4)	No coast down active (OFF2 inactive)
ZSW1 1.5 (bit 5)	No quick stop active (OFF3 inactive)
ZSW1 1.6 (bit 6)	Switching on inhibited active
ZSW1 1.7 (bit 7)	Alarm present
ZSW1 0.0 (bit 8)	Speed setpoint - actual value deviation within tolerance t_off
ZSW1 0.1 (bit 9)	Control requested
ZSW1 0.2 (bit 10)	f or n comparison value reached/exceeded
ZSW1 0.3 (bit 11)	I, M, or P limit not reached
ZSW1 0.4 (bit 12)	Open the holding brake
ZSW1 0.5 (bit 13)	No motor overtemperature alarm
ZSW1 0.6 (bit 14)	1 = Motor rotates forwards (n_act □ 0) 0 = Motor rotates backwards (n_act < 0)
ZSW1 0.7 (bit 15)	No alarm, thermal overload, power unit
ZSW2 (bits 16 to 32)	Bits 16 – 31 → actual speed value

A.10 Parameter model

A.10.1 Parameter definition

Description

A parameter stands for an information memory, which consists of the following elements:

Element	Meaning
Parameter value (PWE)	Contains the information variable(s)
Parameter description (PBE)	Specifies a parameter
Text	Supports visualization and contains a general description of the parameter function or the value.

All the parameters of a drive as a whole clearly describe its behavior or characteristics.

Each parameter is assigned a parameter number. The decimal range from 1 to 65535 is specified as a number range for the parameters. The parameter 0 is not permitted. The decimal ranges from 900 to 999 and from 60000 to 65535 are reserved as profile-specific parameters (see PROFIdrive profile, parameter definition). The profile-specific parameters are to be created precisely according to the definition (see PROFIdrive profile, parameter definition), even if a parameter description is already pre-assigned in the drive.

Access to the parameters (parameter value, parameter description or text) is explained under Basic mode parameter access (Page 185).

Parameter value

The parameter value contains an individual ("simple variable" type) or several similar ("array" type) information variables.

An array consists of n elements of the same data type, which can be addressed individually with subindices from 0 to n-1.

Parameter description

The parameter description contains relevant information on the respective parameter. The following table shows the structure of the parameter description that will be explained in the remainder of this section.

Subindex	Meaning	Data type
1	Identifier (ID)	V2
2	Number of array elements or string lengths	Unsigned 16
3	Standardization factor	Floating point
4	Variable attribute	OctetString 2
5	Reserved	OctetString 4
6	Name	VisibleString 16

Subindex	Meaning	Data type
7	Lower limit	OctetString 4
8	Upper limit	OctetString 4
9	Reserved	OctetString 2
10	ID extension	V2
11	DO–I/O-data reference parameter	Unsigned 16
12	DO–I/O-data standardization	V2
0	Complete description	OctetString 46

Identifier (ID)

Additional parameter characteristics are saved in the ID.

- Bit value = "0" means: "Parameter does not have this attribute."
- Bit value = "1" means: "Parameter has this attribute."

Bit	Meaning	Explanation
0 – 7	Parameter value data type	
8	Standardization factor and variable attribute irrelevant	This bit is set when parameters have data types for which no physical values can be calculated; e.g. the data type string.
9	Parameter unwritable	
10	Additional text array available	
11	Reserved	
12	Parameter factory setting has been changed	This bit is set if the parameter value differs from the factory setting. It is reset if the parameter value corresponds with the factory setting.
13	The parameter value can only be reset	If this bit is set, the associated parameter value is only increased by internal processing, while it can only be set to "0" externally (e.g. "time differences").
14	Array	
15	Reserved	

Number of array elements or string lengths (subindex 2)

For parameters of the "array" data type, the number of elements are entered here. For parameters of the "string" data type, the length of the character sequence is entered here. The data types OctetString or VisibleString correspond to an array of bytes. No arrays of the "string" data type can be formed.

Standardization factor (subindex 3)

Factor for converting the (internal) value into an (external) standardized variable, which together with the unit corresponds to the physical representation of the parameter. The standardization factor is from the data type "floating point".

Variable attribute (subindex 4)

A variable index and a conversion index are saved in the variable attribute:

Octet 1	Octet 2
Variable index	Conversion index (factor A, offset B)

The variable index represents the fixed coding of the physical variable (and therefore the basic unit) of the parameter value. The variable index is from the data type "Unsigned 8".

The conversion index represents the fixed coding of the conversion factor (A) and the offset (B) for a parameter value. With the conversion index the unit can be converted to the basic unit. The conversion index is from the data type "Integer 8".

Name (subindex 6)

"Name" describes the symbolic name of the parameter. The name is from the data type VisibleString with a length of 16 bit.

Lower/upper limit value (subindices 7 and 8)

The "lower/upper limit value" defines the parameter value's valid value range.

The attempt to assign a value outside the parameter's value range is rejected by the drive. The lower and upper limit value are from the same data type as the parameter value; however, the length of the description element is always 4 bytes (file format: right-aligned, big-endian). For parameters whose data types permit no value range (e.g. VisibleString), the content of this description element plays no role.

ID extension (subindex 10)

The ID extension is reserved.

I/O data reference parameters / I/O-data standardization (subindices 11 and 12)

Parameter values can also be transmitted as I/O data (see also PROFIdrive profile, DO-I/O data). For the transmission of standardized variables (data types N2, N4 / X2, X4 / optional Integer16, Integer32, Floating Point), the following is required for calculating the physical values:

- The physical reference value (I/O data reference value)
- The bit (see I/O data standardization), to which the physical reference value refers.

The description elements "I/O data reference parameters" and "I/O data standardization" must be available for parameters of data types X2 and X4.

For parameters of the data types N2 and N4 the description element "I/O data reference parameter" must be available; the description element "I/O data standardization" is optional, as it is specified by the data type.

If parameters of the data types Integer8, Integer16, Integer32, Unsigned8, Unsigned16, Unsigned32, or Floating Point are transmitted as standardized I/O data (with the unit %), the description elements "I/O data reference parameter" and "I/O data standardization" must be

available. If transmitted as non-standardized data these description elements must not be available.

For all other data types these description elements play no role.

Description element	Content
I/O data reference parameter	0 no reference value available 1–65535 parameter number of the reference value
I/O data standardization	Bit 0–5 standardization bit 0–31 (32–63 is reserved) bit 6–14 reserved Bit 15 standardization valid
• NOTE 1: The coding of the standardization bit is set for the parameter of the data types N2 and N4 (14 and 30). • NOTE 2: For the standardized parameters of the floating point data type, the coding of the standardization bit plays no role (=0). NOTE 3: The combination "no reference value available" / "standardization valid" is permitted. • NOTE 4: Parameters used for reference values may not be standardized. NOTE 5: If the entire parameter description is read out in one access, the description elements must be contained (see below).	

Complete description (subindex 0)

The "complete description" contains a whole field of 46 bytes (according to the complete parameter description structure). This length is constant for all parameters (irrespective of data type, etc.).

Text

Text from a text array can be assigned to a parameter as an additional explanation or description. An indexed text line has a length of 16 bytes.

Subindex text array	Text
0	Text 0 (16 bytes)
1	Text 1 (16 bytes)
2 to n	Text 2 to n (per 16 bytes)

The availability of a text array is shown in the parameter description (ID: additional text array available). The text is saved in the object type "array" of the data type "VisibleString 16", which is assigned to the parameter. Text arrays can either be assigned to parameters of the "array" object type (with any data type) or to parameters of the object type "simple variable" (with the data types "Unsigned8/16/32", "Boolean", or "V2"). The individual texts of a text array are assigned to the array elements with parameters of the "array" type and to the values for parameters of the type "simple variables".

Array parameter – text array

Subindex text array == Subindex array parameter

Unsigned8/16/32 – text array

Subindex text array == Parameter value

0 ≤ parameter value ≤ 65535

Boolean – text array

Number of the texts = 2

Subindex text array	Parameter value
0	"incorrect"
1	"true"

V2 – text array

Number of the texts = 32

Two texts are assigned to each bit in the bit sequence, one of the two bit values "0" and "1".

Subindex text array == bit position x 2 + bit value

0 (LSB) ≤ bit position ≤ 15 (MSB), 0 ≤ bit value ≤ 1;

Subindex text array	Parameter value
0	- - - - - - - - - - - - - - - 0
1	- - - - - - - - - - - - - - - 1
2	- - - - - - - - - - - - - - 0 -
3	- - - - - - - - - - - - - - 1 -
4	- - - - - - - - - - - - - 0 - -
:	:
30	0 - - - - - - - - - - - - - - -
31	1 - - - - - - - - - - - - - - -

A.10.2 Global and local parameters

Description

According to the definition a drive unit consists of the drive unit itself as well as one or several drive objects (DO). The drive shafts are assigned axle-type DOs.

With multi-axle units and modular drive units, each DO has its own parameter number range. Two types of parameters with different value ranges are set in the profile:

- **Global parameters**
  Global parameters refer to the entire device (e.g. communication interface parameters). When addressing different DOs of a drive unit, a global parameter always specifies the same value.
- **DO axle-specific parameters**
  These parameters relate to the drive object. The DO axle-specific parameters can have different values in each axle DO (e.g. parameter 967 "control word 1").
  The splitting of the parameters into CU and DO axle-specific parameters is explained in the PROFIdrive profile under parameter definition.

The following image shows an example with global parameter 918 "node address" and the drive-specific parameter 944 "error message counter" for a multi-axle unit or a modular drive unit. A single-axle drive unit is set up in a similar way to the multi-axle drive unit, but only DO1 is present.

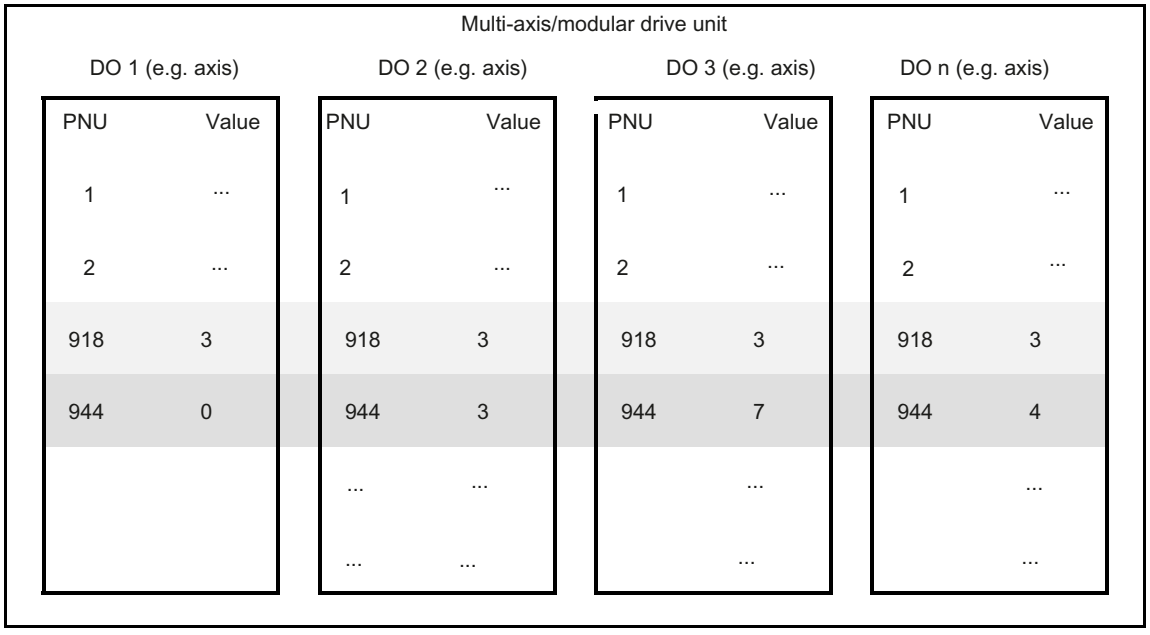


Figure A-1 Global and local parameters of a multi-axle system

The DO ID numbers are in a value range between 0 and 254. With the DO ID 0, the drive unit can be addressed itself (device representative, no axle), and the global parameters can be read. The assignment of the drive axle numbers to the DO is device-specific and can be read from parameter P978 "list of module IDs" (see PROFIdrive profile, DO I/O data).

A.10.3 Basic mode parameter access

Description

In this section, access to parameters in "basic mode" is described. A request language is defined for access. The requests and responses are transmitted acyclically using the communication system's acyclic data exchange mechanism.

Basic mode parameter access should ensure compatibility with older PROFIdrive profiles. Therefore, all drives must support basic mode parameter access (mandatory).

General characteristics

- 16-bit-wide address for parameter number and subindex in each case
- Transmission of complete or partial arrays or the whole parameter description
- Transmission of different parameters in a single access (multi-parameter requests)
- Processing of just one parameter request in each case (no parallel processing)
- A parameter request/response must fit into a data block (standard length 240 bytes). The requests / responses are not split into several data blocks. Depending on device properties or bus configuration, the maximum length of the data blocks can even be less than 240 bytes.
- No spontaneous messages are transmitted.
- Multi-parameter requests are defined for optimal, simultaneous access to different parameters (e.g. user interface image content).
- There are no cyclical parameter requests.
- After power-up the profile-specific parameters must at least be legible in all conditions.

DO addressing modes

The basic mode parameter access is defined as follows with two different addressing modes:

- Basic mode parameter access – local: In this addressing mode, only the local parameters of the DO can be accessed which are linked to the CO that is connected to the parameter access point. Furthermore, access to all global parameters is possible. The DO ID in the parameter request header is "don't care".
- Basic mode parameter access – global: In this addressing mode all parameters of the drive unit can be accessed which are linked to the CO that is connected to the parameter access point. The DO ID in the parameter request is used to access local parameters within the drive unit. For access to global parameters, the DO ID 0 can also be used. This addressing mode is used for ensuring compatibility (PROFIBUS) and may not be used by new PROFINET I/O controllers or supervisor application processes.

A.10.4 Parameter requests and parameter responses

A.10.4.1 Parameter requests and parameter responses

Description

A parameter request consists of three segments:

- **Request header**
  ID for the request and the number of parameters which are being accessed. Multi-axle and module drives, addressing of a single DO.
- **Parameter address**
  Addressing a parameter When addressing several parameters there are also many parameter addresses. The parameter address appears in the request only; not in the response.
- **Parameter value**
  There is a segment for the parameter value for each addressed parameter. Depending on the request ID, parameter values appear either only in the request or in the response.

Words and double words

The following telegram contents are displayed in words (a word or 2 bytes per line). With words or double words, the most significant byte is transmitted first (big-endian).

Word:

Byte 1	Byte 2

Double word:

Byte 1	Byte 2
Byte	Byte 4

The following tables show the structure of the parameter request / the parameter response according to basic mode parameter access.

Parameter request

Block definition	**Byte**	**Byte n**	**n**
Request header	Request reference	Request ID	0
	Axle no. / DO ID	No. of parameters = n	2
1. Parameter address	Attribute	No. of elements	4
	Parameter number (PNU)		.
	Subindex		.
n th parameter address ...			4 + 6 × (n-1)
1. parameter value(s) (only for request	format	No. of values	4 + 6 × n
	values		
"Change parameters").....			

Block definition	Byte	Byte n	n
nth parameter values ...			
			4 + 6 × n +...+ (format_n

Basic mode parameter response

Block definition	Byte	Byte n	n
Response header	Request ref.	Response ID	0
	Axle no. / DO ID	No. of parameters = n	2
1. parameter value(s) (only for request)	Format	No. of values	4
	values or error values		
:			:
nth parameter values	:		:
			4 + ... + (format_n × Qty_n)

Meaning of the fields

Request header

- Request reference
  Unique identification of the request/response pair for the master. The master changes the request reference for each new request (e.g. module 255). The slave mirrors the request reference in the response.
- Request ID
  Two IDs are defined:
  - Request parameter
  - Change parameter

  Depending on the device, a parameter change can be saved onto either a volatile or non-volatile RAM. A changed parameter which is first saved onto a volatile RAM can be stored with parameter P971 on the ROM. The differentiation value/description/test is added to the address as an attribute. The differentiation word/double word is added as a format to the parameter values. For differentiating single/array parameters see "No. of elements" in the parameter address.
- Response ID
  Mirroring of the request ID with supplementary information about whether the request was successful (positive) or not (negative).
  - Request parameter, positive
  - Request parameter, negative (the request could neither be fully nor partially performed)
  - Change parameter, positive
  - Change parameter, negative (the request could neither be fully nor partially performed)

  If the response is negative, error numbers are entered instead of values per partial response.

- Axle no. / DO ID
  For basic mode parameter access – local: Is used for the consistency test. If the DO ID in this field does not match the DO ID of the DO with which this parameter access point (PAP) is linked, the DO parameter manager must respond with error code 0x19 "axle / DO not available" (see Code (Page 190)). For access to global parameters, the DO ID in the parameter request header is "don't care". For basic mode parameter access – global: DO addressing information for multi-axle or modular drives. This enables access to different axles / DOs with one individual range of parameter numbers in each case in the drive through the same PAP.
- No. of parameters
- Specifies the number of the following parameter addresses and/or parameter value ranges for multi-parameter requests. For single parameter requests the no. of Parameter =1st value range 1 ... 39 (limited by the telegram length in PROFIBUS DPV1).
  Important: for a multi-parameter request the PROFIdrive drive unit must arrange the parameter value ranges in the response message into the same sequence as in the associated multi-parameter request message.

Parameter address

- Attribute
  Type of the object being accessed. Value range:
  - Value
  - Description
  - Text
- No. of elements:
  - Number of array elements being accessed or the length of the string being accessed.
  - Value range: 0, 1..234
  - Limited by the telegram length in PROFIBUS DPV1.
  - Special case no. of elements = 0:
    If values are accessed: Recommended for non-indexed parameters
- Parameter number
  Addresses the parameter being accessed. Value range: 1..65535.
- Subindex
  Addresses the first array element of the parameter, the beginning of a string access or the text array, or the description element being accessed. Value range: 0..65535.

Parameter value

- Format
  Format and number define the place in the telegram which will be assigned the following values.
  Value range:
  - Zero (without values as positive partial response to the change request)
  - Data type
  - Error (as negative partial response)
  - In place of a data type the following options are also possible:
  - Byte (for description and text)
  - Word
  - Double word
- No. of values
  Number of the following values or number of the following data type elements (number of the octet for OctetString). In the event of a write request from OctetString the correct length must be provided, otherwise the drive will respond with error code 0x18 "number of values not consistent" (see Code (Page 190)).
- Values
  The values of the parameter
  If the values consist of an odd number of bytes, a zero byte is added on to ensure the word structure of the telegram.
  In the case of a positive partial response, the parameter value contains the following:
  - Format = (data type or byte, word, double word)
  - Number of values
  - The values

  In the case of a negative partial response, the parameter value contains the following:
  - Format = error
  - No. of values = 1
  - Value = error value = error number

  In the case of a negative response, the parameter value may contain the following:
  - Format = error
  - No. of values = 2
  - Value 1 = error value 1: Error number
  - Value 2 = error value 2: Subindex of the first array element in which the error appears
  - (Purpose: following an incorrect write access to an array, not all values should be repeated.)

  In the case of a positive partial response without values, the parameter value contains the following:
  - Format = zero

– Number of values = 0

– (no values)

Not all combinations of attribute, no. of elements, and subindex are permitted (see table below).
A parameter that is not indexed in the profile can be implemented in the drive unit without indices if the response to a parameter access is profile-specific.

Permitted combinations of attribute, number of elements, and subindex

Attribute	No. of elements	Subindex	=> Data	Comment
Value (single parameter)	0	0	The value	The value
Indexed parameter	1	0	The value	The value
	1	0 to n	One value, under subindex	One value, under subindex
	2 to n*)	0 to n	Multiple values, beginning with subindex	Multiple values, beginning with subindex
Description	0 (irrelevant)	0	The whole description	The whole description
	1	1 to n	One description element	One description element
Text (from text array)	1	0 to n	One text (16 bytes), under subindex	One text (16 bytes), under subindex
	2 to n	0 to n	Multiple texts, beginning with subindex	Multiple texts, beginning with subindex
*) If the number of elements available in the device does not tally with the number of elements requested or to be changed, an error is output.				

A.10.4.2 Code

Description

Coding of the fields in parameter requests/responses of the basic mode parameter access

Array	Data type	Values	Comment
Request reference	Unsigned8	0x00 reserved 0x01..0xFF	
Request ID	Unsigned8	0x00 reserved 0x01 request parameter 0x02 change parameter 0x03..0x3F reserved 0x40..0x7F manufacturer-specific 0x80..0xFF reserved	

Array	Data type	Values	Comment
Axle / DO ID	Unsigned8	0x00 Device representative 0x01..0xFE DO ID number 1 to 254 0xFF reserved	Zero is not a DO, but rather the access to the drive unit representatives.
No. of parameters	Unsigned8	0x00 reserved 0x01..0x27 quantity 1 to 39 0x28..0xFF reserved	There may be additional limitations placed by the communication system (telegram length) or optional scalability.
Attribute	Unsigned8	0x00 reserved 0x10 value 0x20 description 0x30 text 0x40..0x70 reserved 0x80..0xF0 manufacturer-specific	The four lower-value bits are reserved for a (future) expansion of the "no. of elements" to 12 bit.
No. of elements	Unsigned8	0x00 special function 0x01..0xEA quantity 1 to 234 0xEB..0xFF reserved	Limitation due to compatibility with PROFIBUS process data ASE telegram lengths
Parameter number	Unsigned16	0x0000 reserved 0x0001... Number 1 to 65535 0xFFFF	
Subindex	Unsigned16	0x0000... Number 0 to 65534 0xFFFE	
Format	Unsigned8	0x00 reserved 0x01 to 0x36 data types 0x37to 0x3F reserved 0x40 zero 0x41 byte 0x42 word 0x43 double word 0x44 error 0x45 to 0xFF reserved	Each slave must at least support the data types byte, word, and double word (mandatory) Master write requests preferably use the "correct" data types. Alternatively the data types byte, word, and double word are also possible. The master must be able to interpret all words / data types.
No. of values	Unsigned8	0x00..0xEA quantity 0 to 234 0xEB..0xFF reserved	Limitation due to data block size of 240 bytes (compatible with older PROFIdrive version 3.1.2)
Error number	Unsigned16	0x0000... Error numbers 0x00FF (see following table)	The higher-value byte is reserved.

During access to reserved values the device outputs an error.

Error numbers in basic mode parameter responses

Error no.	Meaning	Use for	Supplementary info
0x00	Invalid parameter number	Access to unavailable parameter	0
0x01	Parameter value cannot be changed	Change access to a non-changeable parameter value	Subindex
0x02	Lower or upper limit value undershot/exceeded	Change access with value outside the value limits	Subindex
0x03	Incorrect subindex	Access to unavailable subindex or array parameter. May not be used for non- array parameter.	Subindex
0x04	No array	Access with subindex to non-indexed parameter	0
0x05	Incorrect data type	Change access with value which does not match the parameter data type	0
0x06	Setting not permitted (may not be reset)	Change access with value unequal to 0, where this is not permitted	Subindex
0x07	Description element cannot be changed	Change access to a non-changeable description element	Subindex
0x08	Reserved	(PROFIdrive profile V2: PPR-write requested in IR not available)	-
0x09	Description data not available	Access to unavailable description (parameter value is available)	0
0x0A	Reserved	(PROFIdrive profile V2: incorrect access group)	-
0x0B	No operation priority	Change access without change rights for parameter	0
0x0C	Reserved	(PROFIdrive profile V2: incorrect password)	-
0x0D	Reserved	(PROFIdrive profile V2: text for cyclic data transmission illegible)	-
0x0E	Reserved	(PROFIdrive profile V2: name for cyclic data transmission illegible)	-
0x0F	No text array available	Access to unavailable text array (parameter value is available)	0
0x10	Reserved	(PROFIdrive profile V2: PPO-write not available)	-

Error no.	Meaning	Use for	Supplementary info
0x11	Request cannot be performed due to operating state	Access is temporarily not possible due to reasons not specified in detail	0
0x12	Reserved	(PROFIdrive profile V2: other error)	-
0x13	Reserved	(PROFIdrive profile V2: data for cyclic exchange illegible)	-
0x14	Invalid value	Change access with a value which, although within the value limits, is however not permitted due to other long-term reasons (parameter with defined single values)	Subindex
0x15	Response too long	The length of the current response exceeds the maximum transmittable length.	0
0x16	Invalid parameter address	Invalid value or a value which is not supported for the attribute, the number of elements, the parameter number, or the subindex, or a combination of the above.	0
0x17	Invalid format	Write request: invalid format or non-supported parameter data format	0
0x18	Number of inconsistent values	Write request: the number of values of the parameter data does not match the number of elements in the parameter address.	0
0x19	Axle / DO ID not available	Access to unavailable axle/DO	0
0x20	Parameter text element cannot be changed	Change access to a non-changeable parameter text element	Subindex
...			
up to 0x64	Reserved	-	-
0x65 to 0xFF	Manufacturer-specific	-	-

Each PROFIdrive drive unit must always support parameter read and write requests in basic mode with the data types byte, word, and double word (mandatory).

If the PROFIdrive drive unit supports additional data types, it should act as follows:

- In the case of a parameter read request, it should indicate the relevant data type in the read response.
- In the event of a parameter write request, it should check the data type and report an error if there are non-matching parameter types.

If the PROFIdrive drive unit does not support additional data types, it should act as follows:

- It rejects the parameter write request with an error response if the data types do not match.

The error numbers 0x00 to 0x13 come from PROFIdrive profile version 2. Values that cannot be assigned are reserved for future use.

If an error appears with the error number 0x05, 0x16, 0x17, or 0x18 during the processing of a multi-parameter change request, all other parameter changes in the multi-parameter request are canceled.

A.10.4.3 Data flow for basic mode parameter access

Data flow

The basic mode parameter access request is transmitted to the DO/DU parameter manager by writing the request data structure in the PAP data set. When the writing operation is completed, the parameter manager's state machine is triggered according to the following illustration and table.

The basic mode parameter access response is transmitted from the DO/DU parameter manager to the client by reading the response data structure from the PAP data set. The response to the read access is based on the internal state of the parameter manager as per the following illustration and table.

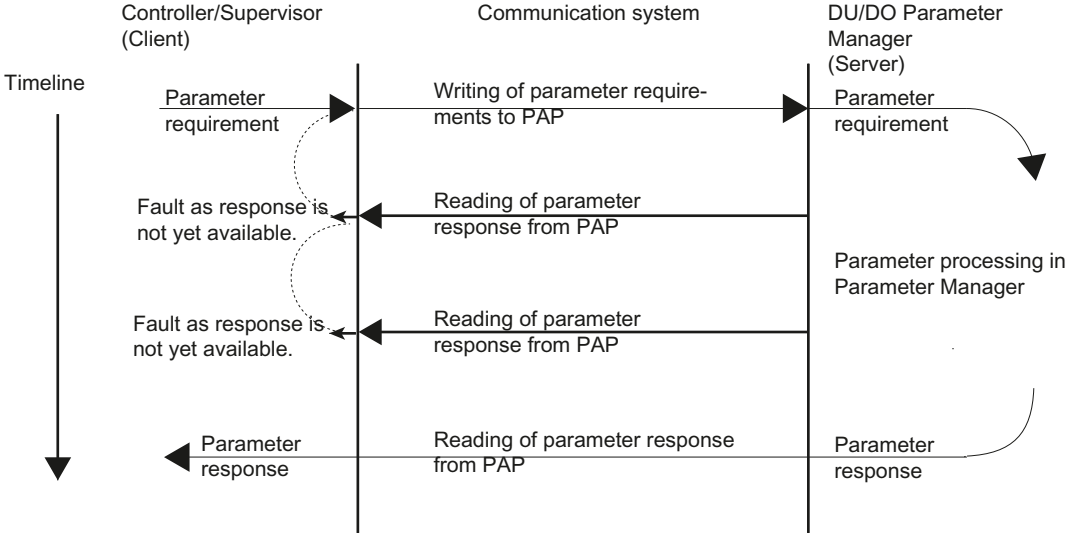


Figure A-2 Data flow

Event (communication)		State			
		Connection interrupted	Idle	Request is processed	Response available
Connection is set up	Solution		Reset processing		
	Successor state	Idle	Idle		
Connection is interrupted	Solution	(Ignore)	Reset processing		
	Successor state	-	Connection interrupted		
Write req.	Solution	(Protocol error) write resp.(-)	Start processing	Write resp.(-) "State conflict"	Reject response Processing
	Successor state	-	Request is processed	-	Request is processed
Read req.	Solution	(Protocol error) read resp. (-)	Read resp.(-)"State conflict"		Read resp.(+)
	Successor state	-	-		Idle
Processing completed	Solution	Reject	Reject		Reject (internal error)
	Successor state	-	-	Response available	-
NOTE 1 meaning: The columns specify the state. The rows explain the event. Each row is divided into two fields. One describes the action; the other the successor state. NOTE 2 This state machine applies to exactly one connection. If multiple connections have been set up, a corresponding number of machine states must be available.					

A.10.5 Telegram sequences for the parameter access

A.10.5.1 Sequence 1: Parameter value request, single

Sequence 1: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 1.	2
Parameter address	Attribute = value	No. of elements = 0	4
	Parameter number		
	Subindex = 0 (irrelevant)		
			10

Sequence 1: Parameter response positive with data from word data-type

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = word	No. of values = 1	4
	Value		6
			8

Sequence 1: Parameter response positive with data from double word data-type

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = double word	No. of values = 1	4
	Value		6
			-
			10

Sequence 1: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter (-)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.2 Sequence 2: Parameter value change, single

Sequence 2: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Change parameter	0
	DO ID = 0	No. of parameters = 1.	2
Parameter address	Attribute = value	No. of elements = 0	4
	Parameter number		
	Subindex = 0 (irrelevant)		

Block definition	Byte n+1	Byte n	n
Parameter value	Format = word	No. of values = 1	10
	Value		12
			14

Sequence 2: Parameter response, positive

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
			4

Sequence 2: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Request ID = Request parameter	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = error	No. of values = 1	4
	Error value		
			8

A.10.5.3 Sequence 3: Parameter value request, multiple array elements

Sequence 3: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 1.	2
Parameter address	Attribute = value	No. of elements = 5	4
	Parameter number		
	Subindex = 0		
			10

Sequence 3: Parameter response, positive

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2

Block definition	Byte n+1	Byte n	n
Parameter value	Format = word	No. of values = 5	4
	Value = 1		6
	Value = 2		
	Value = 3		
	Value = 4		
	Value = 5		
			16

Sequence 3: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(-)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.4 Sequence 4: Parameter value change, multiple array elements

Sequence 4: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Change parameter	0
	DO ID = 0	No. of parameters = 1.	2
Parameter address	Attribute = value	No. of elements = 5	4
	Parameter number		
	Subindex = 125		
Parameter value	Format = word	No. of values = 5	10
	Value = 1		12
	Value = 2		
	Value = 3		
	Value = 4		
	Value = 5		
			22

Sequence 4: Parameter response, positive

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
			4

Sequence 4: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(-)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.5 Sequence 5: Parameter value change, multiple array elements, format byte

Sequence 5: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Change parameter	0
	DO ID = 0	No. of parameters = 1.	2
Parameter address	Attribute = value	No. of elements = 7	4
	Parameter number		
	Subindex = 110		
Parameter value	Format = byte	No. of values = 7	10
	Value = 1	Value = 2	12
	Value = 3	Value = 4	
	Value = 5	Value = 6	
	Value = 7	Dummy byte	18

Sequence 5: Parameter response, positive

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
			4

Sequence 5: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(-)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.6 Sequence 6: Parameter value request, multi-parameter

Sequence 6: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 3.	2
1st parameter address	Attribute = value	No. of elements = 1	4
	Parameter number		
	Subindex = 7		
2nd parameter address	Attribute = value	No. of elements = 100	10
	Parameter number		
	Subindex = 0		
3rd parameter address	Attribute = value	No. of elements = 2	16
	Parameter number		
	Subindex = 13		
			22

Sequence 6: Parameter response (+): All partial access OK

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 3.	2
1st parameter value(s)	Format = word	No. of values = 1	4
	Value		6
2nd parameter value(s)	Format = word	No. of values = 100	8
	Value = 1		10
	Value = 2		
	...		...
	Value = 100		

Block definition	Byte n+1	Byte n	n
3rd parameter value(s)	Format = double word	No. of values = 2	210
	Value = 1		212
	...		
	Value = 2		
	...		
			220

Sequence 6: Parameter response (-): First and third partial access OK, second partial access incorrect

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(-)	0
	DO ID mirrored	No. of parameters = 3.	2
1st parameter value(s)	Format = word	No. of values = 1	4
	Value		6
2nd parameter value(s)	Format = error	No. of values = 1	8
	Error value		10
3rd parameter value(s)	Format = double word	No. of values = 2	12
	Value = 1		14
	...		
	Value = 2		
	...		
			22

A.10.5.7 Sequence 7: Parameter value change, multi-parameter

Sequence 7: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Change parameter	0
	DO ID = 0	No. of parameters = 3.	2
1st parameter address	Attribute = value	No. of elements = 1	4
	Parameter number		
	Subindex = 0		
2nd parameter address	Attribute = value	No. of elements = 100	10
	Parameter number		
	Subindex = 0		

Block definition	Byte n+1	Byte n	n
3rd parameter address	Attribute = value	No. of elements = 2	16
	Parameter number		
	Subindex = 13		
1st parameter value(s)	Format = word	No. of values = 1	22
	Value		24
2nd parameter value(s)	Format = word	No. of values = 100	26
	Value = 1		28
	Value = 2		
...			...
	Value = 100		
3rd parameter value(s)	Format = double word	No. of values = 2	228
	Value = 1		230
	Value = 2		
			238

Sequence 7: Parameter response (+): All partial access OK

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(+)	0
	DO ID mirrored	No. of parameters = 3.	2
			4

Sequence 7: Parameter response (-): First and third partial access OK, second partial access incorrect

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Change parameter(-)	0
	DO ID mirrored	No. of parameters = 3.	2
1st parameter value(s)	Format = zero	No. of values = 0	4
2nd parameter value(s)	Format = error	No. of values = 2	6
	Error value		8
	Incorrect subindex		10
3rd parameter value(s)	Format = zero	No. of values = 0	12
			14

A.10.5.8 Sequence 8: Request description, single

Sequence 8: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 1.	2
parameter address	Attribute = description	No. of elements = 1	4
	Parameter number		
	Subindex = n		
			10

Sequence 8: Parameter response positive with data from word data-type (e.g. ID)

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = word	No. of values = 1	4
	Value		6
			8

Sequence 8: Parameter response positive with text

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = byte	No. of values = 16	4
	Byte 1	Byte 2	6
	...	...	...
	Byte 15	Byte 16	
			22

Sequence 8: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(-)	0
	DO ID mirrored	No. of parameters = 1.	2

Block definition	Byte n+1	Byte n	n
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.9 Sequence 9: Request description, whole

Sequence 9: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 1.	2
parameter address	Attribute = description	No. of Elements = 0 (irrelevant)	4
	Parameter number		
	Subindex = 0 (!)		
			10

Sequence 9: Parameter response, positive

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = byte	No. of Values = (bytes)	4
	ID		6
	(etc.)		
...			...
...			...
			6 + description

Sequence 9: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(-)	0
	DO ID mirrored	No. of parameters = 1.	2

Block definition	Byte n+1	Byte n	n
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.10 Sequence 10: Request text, single

Sequence 10: Parameter request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 1.	2
parameter address	Attribute = text	No. of elements = 1	4
	Parameter number		
	Subindex = n		
			10

Sequence 10: Parameter response, positive

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = byte	No. of values = 16	4
	Byte 1	Byte 2	6
...			...
	Byte 15	Byte 16	
			22

Sequence 10: Parameter response, negative

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request description (-)	0
	DO ID mirrored	No. of parameters = 1.	2
Parameter value	Format = error	No. of values = 1	4
	Error value		6
			8

A.10.5.11 Sequence 11: Parameter request, multi-parameter, different attributes

Sequence 11: Request of values, description, and text in a single request

Block definition	Byte n+1	Byte n	n
Request header	Request reference	Request ID = Request parameter	0
	DO ID = 0	No. of parameters = 3.	2
1st parameter address	Attribute = value	No. of elements = 3	4
	Parameter number		
	Subindex = 0		
2nd parameter address	Attribute = description	No. of elements = 0	10
	Parameter number		
	Subindex = 0		
3rd parameter address	Attribute = text	No. of elements = 3	16
	Parameter number		
	Subindex = 0		
			22

Sequence 11: Parameter response (+): All partial access OK

Block definition	Byte n+1	Byte n	n
Response header	Request ref. mirrored	Response ID = Request parameter(+)	0
	DO ID mirrored	No. of parameters = 3.	2
1st Parameter value(s) (three values)	Format = word	No. of values = 3	4
	Value = 1		6
	Value = 2		
	Value = 3		
2nd parameter value(s) (complete description)	Format = byte	No. of Values = (bytes)	12
	ID		10
	(etc.)		
...			...
3rd Parameter value(s) (three texts)	Format = byte	No. of values = 48	12 + description
	Byte 1	Byte 2	
...			...
	Byte 47	Byte 48	
			62 + description

A.10.6 PROFIdrive-specific data types

Description

A range of data types have been defined for the purpose of using communication that is compliant with PROFIdrive. You will find detailed information on this in the following standards:

- IEC 61800-7-203
- IEC 61800-7-303
- IEC 61158-5

These standards contain detailed descriptions of the data types. The most important data types are listed below.

In order to use drive applications without profile-specific data types, use the data types Integer16 and Integer32 with optional standardization in place of N2, N4/X4, and X4.

PROFIdrive profile-specific data types

Data types used in the PROFIdrive profile	Definition	Coding (dec)
Boolean	Boolean (IEC 61158-5)	1
Integer8	Integer8 (IEC 61158-5)	2
Integer16	Integer16 (IEC 61158-5)	3
Integer32	Integer32 (IEC 61158-5)	4
Unsigned8	Unsigned8 (IEC 61158-5)	5
Unsigned16	Unsigned16 (IEC 61158-5)	6
Unsigned32	Unsigned32 (IEC 61158-5)	7
FloatingPoint32	Float32 (IEC 61158-5)	8
FloatingPoint64	Float64 (IEC 61158-5)	15
VisibleString	VisibleString (IEC 61158-5)	9
OctetString	OctetString (IEC 61158-5)	10
TimeOfDay (with date indication)	TimeOfDay (IEC 61158-5)	11
TimeDifference	TimeDifference (IEC 61158-5)	12
Date	Date (IEC 61158-5)	13
TimeOfDay (without data indication)	TimeOfDay (IEC 61158-5)	52
TimeDifference (with data indication)	TimeDifference (IEC 61158-5)	53
TimeDifference (without data indication)	TimeDifference (IEC 61158-5)	54
Specific data types	**See below for description**	
N2 (normalized value (16-bit))		113
N4 (normalized value (32-bit))		114
V2 bit sequence		115
L2 nibble		116
R2 reciprocal time constant		117
T2 time constant (16-bit)		118
T4 time constant (32-bit)		119

Data types used in the PROFIdrive profile	Definition	Coding (dec)
D2 time constant		120
E2 fixed-point value (16-bit)		121
C4 fixed-point value (32-bit)		122
X2 normalized value, variable (16-bit)		123
X4 normalized value, variable (32-bit)		124

Normalized value N2, N4

Linear normalized value, 0% corresponds to 0 (0x0), 100% corresponds to 2^{12} (0x4,000) for N2, or 2^{28} (0x40,000,000) for N4. The length is 2 or 4 octets.

Coding

Represented in two's complement; MSB (most significant bit) is the first bit after the sign bit (SN) of the first octet.

- SN = 0; positive numbers with 0
- SN = 1; negative numbers

Range of values N2, N4	Resolution N2, N4	Cod. N2, N4 (dec.)	Octet	Bit							
				8	7	6	5	4	3	2	1
-200% ≤ i ≤ (200-2^{-14})%	2^{-12} = 0.0061%	113	1	SN	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
			2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}
-200% ≤ i ≤ (200-2^{30})%	2^{-28} = 9.3 * 10^{-8}%	114	3	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}
			4	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}

Normalized value X2, X4 (example X = 12/28)

Linear normalized value, 0% corresponds to 0 (0x0), 100% corresponds to 2^{x}. These structures are identical to N2 and N4, except that normalization is variable. Normalization can be determined from the parameter descriptions. The length is 2 or 4 octets.

Coding

Represented in two's complement; MSB (most significant bit) is the first bit after the sign bit (SN) of the first octet.

- SN = 0; positive numbers with 0
- SN = 1; negative numbers

Range of values X2, X4	Resolution X2, X4	Cod. X2, X4 (dec.)	Octet	Bit							
				8	7	6	5	4	3	2	1
-800% ≤ i ≤ 800-2^{-12})%	2^{-12}	123	1	SN	2^{2}	2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}
			2	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}

Range of values X2, X4	Resolution X2, X4	Cod. X2, X4 (dec.)	Octet	Bit							
-800% ≤ i ≤ 800-2^{-28})%	2^{-28}	124	3	2^{-13}	2^{-14}	2^{-15}	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}
			4	2^{-21}	2^{-22}	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}

Fixed-point value E2

Linear fixed-point value with four places after the decimal point. 0 corresponds to 0 (0x0), 128 corresponds to 2^{14} (0x4,000). The length is 2 octets.

Coding

Represented in two's complement; MSB (most significant bit) is the first bit after the sign bit (SN) of the first octet.

- SN = 0; positive numbers with 0
- SN = 1; negative numbers

Range of values E2	Resolution	Cod. (dec.)	Octet	Bit							
				8	7	6	5	4	3	2	1
-256+2^{-7} ≤ i ≤ 256-2^{-7}	2^{-7} = 0.0078125	121	1	SN	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}
			2	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

Fixed-point value C4

Linear fixed-point value with four places after the decimal point. 0 corresponds to 0 (0x0), 0.0001 corresponds to 2^{0} (0x0000 0001).

Coding

As with Integer32, the weighting of the bits has been reduced by a factor of 10,000.

Range of values	Resolution	Coding (dec.)	Length
-214,748.3648 ≤ i ≤ 214,748.3648	10^{-4} = 00001	122	4 octets

Bit sequence V2

Bit sequence for checking and representing application functions. 16 Boolean variables are combined to form 2 octets.

Range of values	Resolution	Cod. (dec.)	Octet	Bit							
				8	7	6	5	4	3	2	1
		115	1	15	14	13	12	11	10	9	8
			2	7	6	5	4	3	2	1	0

Nibble (half-byte) L2

Four associated bits make up a nibble. Four nibbles are represented by two octets.

Coding

Range of values	Resolution	Cod. (dec.)	Octet	Bit							
				8	7	6	5	4	3	2	1
-	-	116	1	Nibble 3				Nibble 2			
			2	Nibble 1				Nibble 0			

Time constants T2 and T4

Time data as a multiple of sampling time T_a. Interpreted value = internal value * T_a

Coding

- T2: As with Unsigned16, with a restricted range of values of $0 \leq x \leq 32767$
  When interpreted, internal values that fall outside this range of values are set to 0.
- T4: As with Unsigned32

The values for the time parameters of types D2, T2, T4, and R2 always relate to the specified, constant sampling time T_a. The associated sampling time (parameter p0962) is required to interpret the internal value.

Range of values	Resolution	Coding (dec.)	Length
$0 \leq i \leq 32{,}767 * T_a$	T_a	118	2 octets
$0 \leq i \leq 4{,}294{,}967{,}295 * T_a$	T_a	119	4 octets

Time constant D2

Time data as a fraction of the constant sampling time T_a. Interpreted value = internal value * T_a/16,348

Coding

- T2: As with Unsigned16, with a restricted range of values of $0 \leq x \leq 32767$
  When interpreted, internal values that fall outside this range of values are set to 0.

Range of values	Resolution	Coding (dec.)	Length
$0 \leq i \leq (2\text{-}2\text{-}14) * T_a$	T_a	120	2 octets

Time constant R2

Time data as a reciprocal multiple of the constant sampling time T_a. Interpreted value = 16,348 * T_a/internal value

Coding

- T2: As with Unsigned16, with a restricted range of values of $0 \leq x \leq 16384$
  When interpreted, internal values that fall outside this range of values are set to 16384.

Range of values	Resolution	Coding (dec.)	Length
1 * Ta ≤ i ≤ 16,384 * T_a	T_a	117	2 octets

A.11 References

A.11.1 References

References

	Subject area	Title
1	STEP7 SIMATIC S7-300/400	Automation with STEP 7 in STL and SCL Author: Hans Berger Publicis MCD Verlag ISBN: 978-3-89578-397-5
2	STEP7 SIMATIC S7-300/400	Automation with STEP 7 in LAD and FBD Author: Hans Berger Publicis MCD Verlag ISBN: 978-3-89578-296-1
3	STEP7 SIMATIC S7-300	Automation with SIMATIC S7-300 in the TIA Portal Author: Hans Berger Publicis MCD Verlag ISBN: 978-3-89578-357-9
4	STEP7 SIMATIC S7-400	Automation with SIMATIC S7-400 in the TIA Portal Author: Hans Berger Publicis MCD Verlag ISBN: 978-3-89578-372-2
5	STEP7 SIMATIC S7-1200	Automation with SIMATIC S7-1200 Author: Hans Berger Publicis MCD Verlag ISBN: 978-3-89578-355-5
6	Basic positioner of the G120	Function Manual Basic Positioner 01/2013, FW V4.6, A5E31759509A AA

A.11.2 Internet links

Internet links

	Subject	Link
/1/	Siemens Industry Online Support	http://support.automation.siemens.com
/2/	SIOS entry SINAMICS communication blocks DriveLib for activation in the TIA Portal	https://support.industry.siemens.com/cs/ww/en/view/109475044
/3/	SINAMICS S120 Function Manual Communication	https://support.industry.siemens.com/cs/ww/en/view/109771803
/4/	SIOS entry SINAMICS S120/S150 List Manual 01/2013 FW 4.6	https://support.automation.siemens.com/WW/view/en/68041075
/5/	SIOS entry SINAMICS Startdrive V15.1	https://support.industry.siemens.com/cs/de/en/view/109760845
/6/	SIOS entry SINAMICS Startdrive V16	https://support.industry.siemens.com/cs/ww/en/view/109771710

List of abbreviations

B

B.1 List of abbreviations

Note

The following list of abbreviations includes all abbreviations and their meanings used in the entire SINAMICS family of drives.

A

Abbreviation	Derivation of abbreviation	Meaning
A...	Alarm	Warning
AC	Alternating Current	Alternating current
ADC	Analog Digital Converter	Analog digital converter
AI	Analog Input	Analog input
AIM	Active Interface Module	Active Interface Module
ALM	Active Line Module	Active Line Module
AO	Analog Output	Analog output
AOP	Advanced Operator Panel	Advanced Operator Panel
APC	Advanced Positioning Control	Advanced Positioning Control
AR	Automatic Restart	Automatic restart
ASC	Armature Short-Circuit	Armature short-circuit
ASCII	American Standard Code for Information Interchange	American coding standard for the exchange of information
AS-i	AS-Interface (Actuator Sensor Interface)	AS-Interface (open bus system in automation technology)
ASM	Asynchronmotor	Induction motor
AVS	Active Vibration Suppression	Active load vibration damping
AWG	American Wire Gauge	American Wire Gauge (Standard for cross-sections of cables)

B

Abbreviation	Derivation of abbreviation	Meaning
BB	Betriebsbedingung	Operation condition
BERO	-	Contactless proximity switch
BI	Binector Input	Binector input
BIA	Berufsgenossenschaftliches Institut für Arbeitssicherheit	BG Institute for Occupational Safety and Health
BICO	Binector Connector Technology	Binector connector technology

Abbreviation	Derivation of abbreviation	Meaning
BLM	Basic Line Module	Basic Line Module
BO	Binector Output	Binector output
BOP	Basic Operator Panel	Basic operator panel

C

Abbreviation	Derivation of abbreviation	Meaning
C	Capacitance	Capacitance
C...	-	Safety message
CAN	Controller Area Network	Serial bus system
CBC	Communication Board CAN	Communication Board CAN
CBE	Communication Board Ethernet	PROFINET communication module (Ethernet)
CD	Compact Disc	Compact disc
CDS	Command Data Set	Command data set
CF Card	CompactFlash Card	CompactFlash card
CI	Connector Input	Connector input
CLC	Clearance Control	Clearance control
CNC	Computerized Numerical Control	Computer-supported numerical control
CO	Connector Output	Connector output
CO/BO	Connector Output/Binector Output	Connector/binector output
COB-ID	CAN Object-Identification	CAN Object Identification
CoL	Certificate of License	Certificate of License
COM	Common contact of a change-over relay	Center contact of a change-over contact
COMM	Commissioning	Commissioning
CP	Communication Processor	Communications processor
CPU	Central Processing Unit	Central processing unit
CRC	Cyclic Redundancy Check	Cyclic redundancy check
CSM	Control Supply Module	Control Supply Module
CU	Control Unit	Control Unit
CUA	Control Unit Adapter	Control Unit Adapter
CUD	Control Unit DC	Control Unit DC

D

Abbreviation	Derivation of abbreviation	Meaning
DAC	Digital Analog Converter	Digital analog converter
DC	Direct Current	Direct current
DCB	Drive Control Block	Drive Control Block
DCBRK	DC Brake	DC braking
DCC	Drive Control Chart	Drive Control Chart
DCN	Direct Current Negative	Direct current negative
DCP	Direct Current Positive	Direct current positive

Abbreviation	Derivation of abbreviation	Meaning
DDC	Dynamic Drive Control	Dynamic Drive Control
DDS	Drive Data Set	Drive Data Set
DHCP	Dynamic Host Configuration Protocol	Dynamic Host Configuration Protocol (Communication protocol)
DI	Digital Input	Digital input
DI/DO	Digital Input/Digital Output	Digital input/output, bidirectional
DIN	Deutsches Institut für Normung	Deutsches Institut für Normung (German Institute for Standardization)
DMC	DRIVE-CLiQ Hub Module Cabinet	DRIVE-CLiQ Hub Module Cabinet
DME	DRIVE-CLiQ Hub Module External	DRIVE-CLiQ Hub Module External
DMM	Double Motor Module	Double Motor Module
DO	Digital Output	Digital output
DO	Drive Object	Drive object
DP	Decentralized Peripherals	Distributed I/O
DPRAM	Dual Ported Random Access Memory	Dual-Port Random Access Memory
DQ	DRIVE-CLiQ	DRIVE-CLiQ
DRAM	Dynamic Random Access Memory	Dynamic Random Access Memory
DRIVE-CLiQ	Drive Component Link with IQ	Drive Component Link with IQ
DSC	Dynamic Servo Control	Dynamic Servo Control
DSM	Doppelsubmodul	Double submodule
DTC	Digital Time Clock	Timer

E

Abbreviation	Derivation of abbreviation	Meaning
EASC	External Armature Short-Circuit	External armature short-circuit
EDS	Encoder Data Set	Encoder data set
EEPROM	Electrically Erasable Programmable Read-Only Memory	Electrically Erasable Programmable Read-Only Memory
EGB	Elektrostatisch gefährdete Baugruppen	Electrostatic sensitive devices
EIP	EtherNet/IP	EtherNet Industrial Protocol (real-time Ethernet)
ELCB	Earth Leakage Circuit Breaker	Residual current operated circuit breaker
ELP	Earth Leakage Protection	Ground-fault monitoring
EMC	Electromagnetic Compatibility	Electromagnetic compatibility
EMF	Electromotive Force	Electromotive force
EMK	Elektromotorische Kraft	Electromotive force
EMV	Elektromagnetische Verträglichkeit	Electromagnetic compatibility
EN	Europäische Norm	European standard
EnDat	Encoder-Data-Interface	Encoder interface
EP	Enable Pulses	Pulse enable
EPOS	Einfachpositionierer	Basic positioner
ES	Engineering System	Engineering system
ESB	Ersatzschaltbild	Equivalent circuit diagram

Abbreviation	Derivation of abbreviation	Meaning
ESD	Electrostatic Sensitive Devices	Electrostatic sensitive devices
ESM	Essential Service Mode	Essential service mode
ESR	Extended Stop and Retract	Extended stop and retract

F

Abbreviation	Derivation of abbreviation	Meaning
F...	Fault	Fault
FAQ	Frequently Asked Questions	Frequently Asked Questions
FBLOCKS	Free Blocks	Free function blocks
FCC	Function Control Chart	Function control chart
FCC	Flux Current Control	Flux current control
FD	Function Diagram	Function diagram
F-DI	Failsafe Digital Input	Fail-safe digital input
F-DO	Failsafe Digital Output	Fail-safe digital output
FEPROM	Flash-EPROM	Non-volatile write and read memory
FG	Function Generator	Function generator
FI	-	Fault current
FOC	Fiber-Optic Cable	Fiber-optic cable
FP	Funktionsplan	Function diagram
FPGA	Field Programmable Gate Array	Field Programmable Gate Array
F-PLC	Fail-safe PLC	Fail-safe PLC
FW	Firmware	Firmware

G

Abbreviation	Derivation of abbreviation	Meaning
GB	Gigabyte	Gigabyte
GC	Global Control	Global control telegram (broadcast telegram)
GND	Ground	Reference potential for all signal and operating voltages, usually defined as 0 V (also referred to as M)
GSD	Gerätestammdaten	Device master data: Describe the features of a PROFIBUS slave
GSV	Gate Supply Voltage	Gate supply voltage
GUID	Globally Unique Identifier	Globally Unique Identifier

H

Abbreviation	Derivation of abbreviation	Meaning
HF	High frequency	High frequency
HFD	Hochfrequenzdrossel	Radio frequency reactor
HLA	Hydraulic Linear Actuator	Hydraulic linear actuator

Abbreviation	Derivation of abbreviation	Meaning
HLG	Hochlaufgeber	Ramp-function generator
HM	Hydraulic Module	Hydraulic Module
HMI	Human Machine Interface	Human Machine Interface
HTL	High-Threshold Logic	Logic with high interference threshold
HTTP	Hypertext Transfer Protocol	Hypertext Transfer Protocol (communication protocol)
HTTP	Hypertext Transfer Protocol Secure	Hypertext Transfer Protocol Secure (communication protocol)
HW	Hardware	Hardware

I

Abbreviation	Derivation of abbreviation	Meaning
i. V.	In Vorbereitung	Under development: This property is currently not available
I/O	Input/Output	Input/output
I2C	Inter-Integrated Circuit	Internal serial data bus
IASC	Internal Armature Short-Circuit	Internal armature short-circuit
IBN	Inbetriebnahme	Commissioning
ID	Identifier	Identification
IE	Industrial Ethernet	Industrial Ethernet
IEC	International Electrotechnical Commission	International Electrotechnical Commission
IF	Interface	Interface
IGBT	Insulated Gate Bipolar Transistor	Insulated gate bipolar transistor
IGCT	Integrated Gate-Controlled Thyristor	Semiconductor power switch with integrated control electrode
IL	Impulslöschung	Pulse suppression
IP	Internet Protocol	Internet Protocol
IPO	Interpolator	Interpolator
ISO	Internationale Organisation für Normung	International Standards Organization
IT	Isolé Terre	Non-grounded three-phase line supply
IVP	Internal Voltage Protection	Internal voltage protection

J

Abbreviation	Derivation of abbreviation	Meaning
JOG	Jogging	Jogging

K

Abbreviation	Derivation of abbreviation	Meaning
KDV	Kreuzweiser Datenvergleich	Data cross-check
KHP	Know-how protection	Know-how protection
KIP	Kinetische Pufferung	Kinetic buffering
Kp	-	Proportional gain
KTY84-130	-	Temperature sensor

L

Abbreviation	Derivation of abbreviation	Meaning
L		
L	-	Symbol for inductance
LED	Light Emitting Diode	Light emitting diode
LIN	Linearmotor	Linear motor
LR	Lageregler	Position controller
LSB	Least Significant Bit	Least significant bit
LSC	Line-Side Converter	Line-side converter
LSS	Line-Side Switch	Line-side switch
LU	Length Unit	Length unit
LWL	Lichtwellenleiter	Fiber-optic cable

M

Abbreviation	Derivation of abbreviation	Meaning
M	-	Symbol for torque
M	Masse	Reference potential for all signal and operating voltages, usually defined as 0 V (also referred to as GND)
MB	Megabyte	Megabyte
MCC	Motion Control Chart	Motion Control Chart
MDI	Manual Data Input	Manual data input
MDS	Motor Data Set	Motor data set
MLFB	Maschinenlesbare Fabrikatebezeichnung	Machine-readable product code
MM	Motor Module	Motor Module
MMC	Man-Machine Communication	Man-machine communication
MMC	Micro Memory Card	Micro memory card
MRCD	Modular Residual Current protection Device	Modular Residual Current protection Device
MSB	Most Significant Bit	Most significant bit
MSC	Motor-Side Converter	Motor-side converter
MSCY_C1	Master Slave Cycle Class 1	Cyclic communication between master (class 1) and slave
MSR	Motorstromrichter	Motor-side converter
MT	Messtaster	Probe

N

Abbreviation	Derivation of abbreviation	Meaning
N. C.	Not Connected	Not connected
N...	No Report	No report or internal message
NAMUR	Interessengemeinschaft Automatisierungstechnik der Prozessindustrie	User association of automation technology in the process industry
NC	Normally Closed (contact)	NC contact
NC	Numerical Control	Numerical control
NEMA	National Electrical Manufacturers Association	Standardization association in USA (United States of America)
NM	Nullmarke	Zero mark
NO	Normally Open (contact)	NO contact
NSR	Netzstromrichter	Line-side converter
NTP	Network Time Protocol	Standard for synchronization of the time of day
NVRAM	Non-Volatile Random Access Memory	Non-volatile read/write memory

O

Abbreviation	Derivation of abbreviation	Meaning
OA	Open Architecture	Software component which provides additional functions for the SINAMICS drive system
OAIF	Open Architecture Interface	Version of the SINAMICS firmware as of which the OA application can be used
OASP	Open Architecture Support Package	Expands the commissioning tool by the corresponding OA application
OC	Operating Condition	Operation condition
OCC	One Cable Connection	One-cable technology
OEM	Original Equipment Manufacturer	Original equipment manufacturer
OLP	Optical Link Plug	Bus connector for fiber-optic cable
OMI	Option Module Interface	Option Module Interface

P

Abbreviation	Derivation of abbreviation	Meaning
p...	-	Adjustable parameters
P1	Processor 1	CPU 1
P2	Processor 2	CPU 2
PB	PROFIBUS	PROFIBUS
PcCtrl	PC Control	Master control
PD	PROFIdrive	PROFIdrive
PDC	Precision Drive Control	Precision Drive Control
PDS	Power unit Data Set	Power unit data set
PDS	Power Drive System	Drive system

Abbreviation	Derivation of abbreviation	Meaning
PE	Protective Earth	Protective ground
PELV	Protective Extra Low Voltage	Safety extra-low voltage
PFH	Probability of dangerous failure per hour	Probability of dangerous failure per hour
PG	Programmiergerät	Programming device
PI	Proportional Integral	Proportional integral
PID	Proportional Integral Differential	Proportional integral differential
PLC	Programmable Logical Controller	Programmable logic controller
PLL	Phase-Locked Loop	Phase-locked loop
PM	Power Module	Power Module
PMI	Power Module Interface	Power Module Interface
PMSM	Permanent-magnet synchronous motor	Permanent-magnet synchronous motor
PN	PROFINET	PROFINET
PNO	PROFIBUS Nutzerorganisation	PROFIBUS user organization
PPI	Point to Point Interface	Point-to-point interface
PRBS	Pseudo Random Binary Signal	White noise
PROFIBUS	Process Field Bus	Serial data bus
PS	Power Supply	Power supply
PSA	Power Stack Adapter	Power Stack Adapter
PT1000	-	Temperature sensor
PTC	Positive Temperature Coefficient	Positive temperature coefficient
PTP	Point To Point	Point-to-point
PWM	Pulse Width Modulation	Pulse width modulation
PZD	Prozessdaten	Process data

Q

Abbreviation	Derivation of abbreviation	Meaning
No entries		

R

Abbreviation	Derivation of abbreviation	Meaning
r...	-	Display parameters (read-only)
RAM	Random Access Memory	Memory for reading and writing
RCCB	Residual Current Circuit Breaker	Residual current operated circuit breaker
RCD	Residual Current Device	Residual current device
RCM	Residual Current Monitor	Residual current monitor
REL	Reluctance motor textile	Reluctance motor textile
RESM	Reluctance synchronous motor	Synchronous reluctance motor
RFG	Ramp-Function Generator	Ramp-function generator

Abbreviation	Derivation of abbreviation	Meaning
RJ45	Registered Jack 45	Term for an 8-pin socket system for data transmission with shielded or non-shielded multi-wire copper cables
RKA	Rückkühlanlage	Cooling unit
RLM	Renewable Line Module	Renewable Line Module
RO	Read Only	Read only
ROM	Read-Only Memory	Read-only memory
RPDO	Receive Process Data Object	Receive Process Data Object
RS232	Recommended Standard 232	Interface standard for cable-connected serial data transmission between a sender and receiver (also known as EIA232)
RS485	Recommended Standard 485	Interface standard for a cable-connected differential, parallel, and/or serial bus system (data transmission between a number of senders and receivers, also known as EIA485)
RTC	Real Time Clock	Real-time clock
RZA	Raumzeigerapproximation	Space-vector approximation

S

Abbreviation	Derivation of abbreviation	Meaning
S1	-	Continuous operation
S3	-	Intermittent duty
SAM	Safe Acceleration Monitor	Safe acceleration monitoring
SBC	Safe Brake Control	Safe brake control
SBH	Sicherer Betriebshalt	Safe operating stop
SBR	Safe Brake Ramp	Safe brake ramp monitoring
SBT	Safe Brake Test	Safe brake test
SCA	Safe Cam	Safe cam
SCC	Safety Control Channel	Safety Control Channel
SCSE	Single Channel Safety Encoder	Single-channel safety encoder
SD Card	SecureDigital Card	Secure digital memory card
SDC	Standard Drive Control	Standard Drive Control
SDI	Safe Direction	Safe motion direction
SE	Sicherer Software-Endschalter	Safe software limit switch
SESM	Separately-excited synchronous motor	Separately excited synchronous motor
SG	Sicher reduzierte Geschwindigkeit	Safely limited speed
SGA	Sicherheitsgerichteter Ausgang	Safety-related output
SGE	Sicherheitsgerichteter Eingang	Safety-related input
SH	Sicherer Halt	Safe stop
SI	Safety Integrated	Safety Integrated
SIC	Safety Info Channel	Safety Info Channel
SIL	Safety Integrity Level	Safety Integrity Level
SITOP	-	Siemens power supply system

Abbreviation	Derivation of abbreviation	Meaning
SLA	Safely-Limited Acceleration	Safely limited acceleration
SLM	Smart Line Module	Smart Line Module
SLP	Safely-Limited Position	Safely Limited Position
SLS	Safely-Limited Speed	Safely limited speed
SLVC	Sensorless Vector Control	Sensorless vector control
SM	Sensor Module	Sensor Module
SMC	Sensor Module Cabinet	Sensor Module Cabinet
SME	Sensor Module External	Sensor Module External
SMI	SINAMICS Sensor Module Integrated	SINAMICS Sensor Module Integrated
SMM	Single Motor Module	Single Motor Module
SN	Sicherer Software-Nocken	Safe software cam
SOS	Safe Operating Stop	Safe operating stop
SP	Service Pack	Service pack
SP	Safe Position	Safe position
SPC	Setpoint Channel	Setpoint channel
SPI	Serial Peripheral Interface	Serial peripheral interface
SPS	Speicherprogrammierbare Steuerung	Programmable logic controller
SS1	Safe Stop 1	Safe Stop 1 (time-monitored, ramp-monitored)
SS1E	Safe Stop 1 External	Safe Stop 1 with external stop
SS2	Safe Stop 2	Safe Stop 2
SS2E	Safe Stop 2 External	Safe Stop 2 with external stop
SSI	Synchronous Serial Interface	Synchronous serial interface
SSL	Secure Sockets Layer	Encryption protocol for secure data transfer (new TLS)
SSM	Safe Speed Monitor	Safe feedback from speed monitor
SSP	SINAMICS Support Package	SINAMICS support package
STO	Safe Torque Off	Safe torque off
STW	Steuerwort	Control word

T

Abbreviation	Derivation of abbreviation	Meaning
TB	Terminal Board	Terminal Board
TEC	Technology Extension	Software component which is installed as an additional technology package and which expands the functionality of SINAMICS (previously OA application)
TIA	Totally Integrated Automation	Totally Integrated Automation
TLS	Transport Layer Security	Encryption protocol for secure data transfer (previously SSL)
TM	Terminal Module	Terminal Module
TN	Terre Neutre	Grounded three-phase line supply
Tn	-	Integral time

Abbreviation	Derivation of abbreviation	Meaning
TPDO	Transmit Process Data Object	Transmit Process Data Object
TSN	Time-Sensitive Networking	Time-Sensitive Networking
TT	Terre Terre	Grounded three-phase line supply
TTL	Transistor-Transistor-Logic	Transistor-transistor logic
Tv	-	Rate time

U

Abbreviation	Derivation of abbreviation	Meaning
UL	Underwriters Laboratories Inc.	Underwriters Laboratories Inc.
UPS	Uninterruptible Power Supply	Uninterruptible power supply
USV	Unterbrechungsfreie Stromversorgung	Uninterruptible power supply
UTC	Universal Time Coordinated	Universal time coordinated

V

Abbreviation	Derivation of abbreviation	Meaning
VC	Vector Control	Vector control
Vdc	-	DC link voltage
VdcN	-	Partial DC link voltage negative
VdcP	-	Partial DC link voltage positive
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik	Association of Electrical Engineering, Electronics and Information Technology
VDI	Verein Deutscher Ingenieure	Verein Deutscher Ingenieure [Association of German Engineers]
VPM	Voltage Protection Module	Voltage Protection Module
Vpp	Volt peak to peak	Volt peak to peak
VSM	Voltage Sensing Module	Voltage Sensing Module

W

Abbreviation	Derivation of abbreviation	Meaning
WEA	Wiedereinschaltautomatik	Automatic restart
WZM	Werkzeugmaschine	Machine tool

X

Abbreviation	Derivation of abbreviation	Meaning
XML	Extensible Markup Language	Extensible markup language (standard language for Web publishing and document management)

Y

Abbreviation	Derivation of abbreviation	Meaning
No entries		

Z

Abbreviation	Derivation of abbreviation	Meaning
ZK	Zwischenkreis	DC link
ZM	Zero Mark	Zero mark
ZSW	Zustandswort	Status word

Index

T