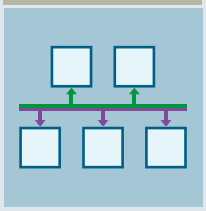
SIEMENS

Communication



Function Manual

SINAMICS

SINAMICS G115D, G120, G120P, G120C, G120D

Fieldbuses

Edition 02/2023

www.siemens.com/drives

SIEMENS

SINAMICS

SINAMICS G115D, G120, G120P, G120C, G120D Fieldbuses

Function Manual

Edition 02/2023, Firmware V4.7 SP14

02/2023, FW V4.7 SP14
A5E34229197B AH

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

A5E34229197B AH
Ⓟ 03/2023 Subject to change

Preface

About this manual

This manual describes the settings and preconditions that are required to communicate with a higher-level control system with the subsequently listed fieldbus systems.

Fieldbuses for SINAMICS G120

- PROFIBUS DP
- PROFINET
- EtherNet/IP
- USS
- Modbus RTU
- CANopen

Additional fieldbuses for SINAMICS G120P

- BACnet MS/TP
- P1

Fieldbuses for SINAMICS G115D

- PROFINET
- EtherNet/IP
- AS-Interface

What is the meaning of the symbols in the manual?

 Reference to further information in the manual

 Download from the Internet

 DVD that can be ordered

End of a handling instruction.
❒

Table of contents

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.

1.2 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.

1.3 Security information

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

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

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

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

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

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

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
https://www.siemens.com/cert.

Further information is provided on the Internet:

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

⚠ WARNING

Unsafe operating states resulting from software manipulation

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

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

General information

2

Communication with the control, even when the line voltage is switched off

So that communication with the control system in your plant or system continues to function even when the line voltage is switched off, you must externally supply the converter/Control Unit with 24 V DC. To do this, use terminals 31 and 32 or connector X01 (X01/X02 with G115D). You can find additional details in the operating instructions for the converter or the Control Unit.

2.1 Ethernet and PROFINET protocols that are used

The converter supports the protocols listed in the following tables. The address parameters, the relevant communication layer as well as the communication role and the communication direction are specified for each protocol.

You require this information to set the appropriate safety measures to protect the automation system, e.g. in the firewall.

As the security measures are limited to Ethernet and PROFINET networks, no PROFIBUS protocols are listed in the table.

Table 2-1 PROFINET protocols

Protocol	Port number	Layer (2) Link layer (4) Transport layer	Function/description
DCP: Discovery and configuration protocol	Not relevant	(2) Ethernet II and IEEE 802.1Q and Ethertype 0x8892 (PROFINET)	**Accessible stations, PROFINET Discovery and configuration** DCP is used by PROFINET to determine PROFINET devices and to make basic settings. DCP uses the special multicast MAC address: xx-xx-xx-01-0E-CF, xx-xx-xx = Organizationally Unique Identifier
LLDP: Link Layer Discovery Protocol	Not relevant	(2) Ethernet II and IEEE 802.1Q and Ethertype 0x88CC (PROFINET)	**PROFINET Link Layer Discovery protocol** LLDP is used by PROFINET to determine and manage neighborhood relationships between PROFINET devices. LLDP uses the special multicast MAC address: 01-80-C2-00-00-0E
MRP: Media Redundancy Protocol	Not relevant	(2) Ethernet II and IEEE 802.1Q and Ethertype 0x88E3 (PROFINET)	**PROFINET medium redundancy** MRP enables the control of redundant routes through a ring topology. MRP uses the special multicast MAC address: xx-xx-xx-01-15-4E, xx-xx-xx = Organizationally Unique Identifier

Protocol	Port number	Layer (2) Link layer (4) Transport layer	Function/description
PTCP Precision Transparent Clock Protocol	Not relevant	(2) Ethernet II and IEEE 802.1Q and Ethertype 0x8892 (PROFINET)	**PROFINET send clock and time synchronization, based on IEEE 1588** PTC is used to implement send clock synchronization and time synchronization between RJ45 ports, which are required for IRT operation. PTCP uses the special multicast MAC address: xx-xx-xx-01-0E-CF, xx-xx-xx = Organizationally Unique Identifier
PROFINET IO data	Not relevant	(2) Ethernet II and IEEE 802.1Q and Ethertype 0x8892 (PROFINET)	**PROFINET Cyclic IO data transfer** The PROFINET IO telegrams are used to transfer IO data cyclically between the PROFINET IO controller and IO devices via Ethernet.
PROFINET Context Manager	34964	(4) UDP	**PROFINET connection less RPC** The PROFINET context manager provides an endpoint mapper in order to establish an application relationship (PROFINET AR).

Table 2-2 EtherNet/IP protocols

Protocol	Port number	Layer (2) Link layer (4) Transport layer	Function/description
Implicit messaging	2222	(4) UDP	Used for exchanging I/O data. This is inactive when delivered. Is activated when selecting EtherNet/IP.
Explicit messaging	44818	(4) TCP (4) UDP	Used for parameter access (writing, reading). This is inactive when delivered. Is activated when selecting EtherNet/IP.

Table 2-3 Connection-oriented communication protocols

Protocol	Port number	Layer (2) Link layer (4) Transport layer	Function/description
ISO on TCP (according to RFC 1006)	102	(4) TCP	**ISO-on-TCP protocol** ISO on TCP (according to RFC 1006) is used for the message-oriented data exchange to a remote CPU, WinAC or devices of other suppliers. Communication with ES, HMI, etc. is activated in the factory setting, and is always required.
SNMP Simple Network Management Protocol	161	(4) UDP	**Simple network management protocol** SNMP enables network management data to be read out and set (SNMP managed objects) by the SNMP manager. It is activated in the factory setting, and is always required
Reserved	49152 ... 65535	(4) TCP (4) UDP	Dynamic port area that is used for the active connection endpoint if the application does not specify the local port.

Communication via PROFIBUS and PROFINET 3

3.1 PROFIDRIVE profile - Cyclic communication

Depending on the Control Unit or converter, there are different telegrams for communication via PROFIBUS DP or PROFINET IO. The structure of the individual telegrams are listed below.

The Startdrive commissioning tool or an operator panel only list the telegrams for selection that are possible with your particular converter.

Commissioning the converter and selecting a telegram are described in the operating instructions.

Overview of the manuals (Page 217)

Communication telegrams if "basic positioner" has been configured

The converter has the following telegrams if you have configured the "Basic positioner" function:

- Standard telegram 7, PZD-2/2
- Standard telegram 9, PZD-10/5
- SIEMENS telegram 110, PZD-12/7
- SIEMENS telegram 111, PZD-12/12
- Telegram 999, free interconnection

Telegrams 7, 9, 110 and 111 are described in the "Basic positioner" Function Manual

Overview of the manuals (Page 217)

Communication telegrams for speed control

The send and receive telegrams of the converter for closed-loop speed control are structured as follows:

Telegram 1

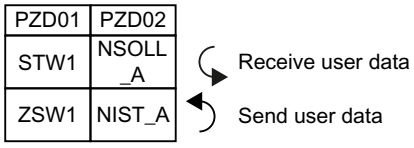


Figure 3-1 16-bit speed setpoint

Telegram 2

PZD01	PZD02	PZD03	PZD04
STW1	NSOLL_B		STW2
ZSW1	NIST_B		ZSW2

Figure 3-2 32-bit speed setpoint

Telegram 3

PZD01	PZD02	PZD03	PZD04	PZD05	PZD06	PZD07	PZD08	PZD09
STW1	NSOLL_B		STW2	G1_ STW				
ZSW1	NIST_B		ZSW2	G1_ ZSW	G1_XIST1		G1_XIST2	

Figure 3-3 32-bit speed setpoint with 1 position encoder

Telegram 4

PZD01	PZD02	PZD03	PZD04	PZD05	PZD06	PZD07	PZD08	PZD09	PZD10	PZD11	PZD12	PZD13	PZD14
STW1	NSOLL_B		STW2	G1_ STW	G2_ST W								
ZSW1	NIST_B		ZSW2	G1_ ZSW	G1_XIST1		G1_XIST2		G2_ ZSW	G2_XIST1		G2_XIST2	

Figure 3-4 32-bit speed setpoint with 2 position encoders

Telegram 20

PZD01	PZD02	PZD03	PZD04	PZD05	PZD06
STW1	NSOLL_ A				
ZSW1	NIST_A GLATT	IAIST_ GLATT	MIST_ GLATT	PIST_ GLATT	MELD_ NAMUR

Figure 3-5 16-bit speed setpoint for VIK-Namur

Telegram 350

PZD01	PZD02	PZD03	PZD04
STW1	NSOLL _A	M_LIM	STW3
ZSW1	NIST_A GLATT	IAIST_ GLATT	ZSW3

Figure 3-6 16-bit speed setpoint with torque limiting

Telegram 352

PZD01	PZD02	PZD03	PZD04	PZD05	PZD06
STW1	NSOLL _A	Freely assignable			
ZSW1	NIST_A GLATT	IAIST_ GLATT	MIST_ GLATT	WARN_ CODE	FAULT_ CODE

Figure 3-7 16-bit speed setpoint for PCS7

Telegram 353

	PZD01	PZD02
PKW	STW1	NSOLL _A
	ZSW1	NIST_A GLATT

Figure 3-8 16-bit speed setpoint with PKW range to read and write parameters

Telegram 354

	PZD01	PZD02	PZD03	PZD04	PZD05	PZD06
PKW	STW1	NSOLL _A	Freely assignable			
	ZSW1	NIST_A GLATT	IAIST_ GLATT	MIST_ GLATT	WARN_ CODE	FAULT_ CODE

Figure 3-9 16-bit speed setpoint for PCS7 with PKW range to read and write parameters

Telegram 999

PZD01	PZD02	PZD03	PZD04	PZD05	PZD06	PZD07	PZD08	PZD09	PZD10	PZD11	PZD12	PZD13 ... PZD17
STW1	Telegram length for the receive data											
ZSW1	Telegram length for the transmit data											

Figure 3-10 Telegram with free interconnection and length

Abbreviation	Explanation	Abbreviation	Explanation
PZD	Process data	PKW	Parameter channel
STW	Control word	PIST_GLATT	Actual active power value, smoothed
ZSW	Status word	M_LIM	Torque limit
NSOLL_A	Speed setpoint 16-bit	FAULT_CODE	Fault code
NSOLL_B	Speed setpoint 32-bit	WARN_CODE	Alarm code
NIST_A	Actual speed value 16-bit	MELD_NAMUR	Message according to the VIK-NAMUR definition
NIST_B	Actual speed value 32-bit	G1_STW / G2_STW	Control word for encoder 1 or encoder 2
IAIST	Current actual value	G1_ZSW / G2_ZSW	Status word for encoder 1 or encoder 2
IAIST_GLATT	Current actual value, smoothed	G1_XIST1 / G2_XIST1	Position actual value 1 from encoder 1 or encoder 2
MIST_GLATT	Torque actual value, smoothed	G1_XIST2 / G2_XIST2	Position actual value 2 from encoder 1 or encoder 2

Interconnection of the process data

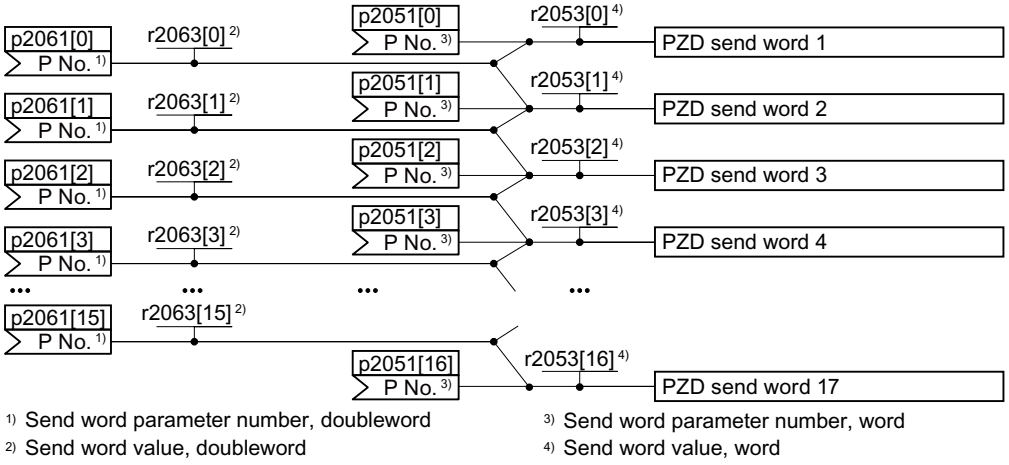


Figure 3-11 Interconnection of the send words

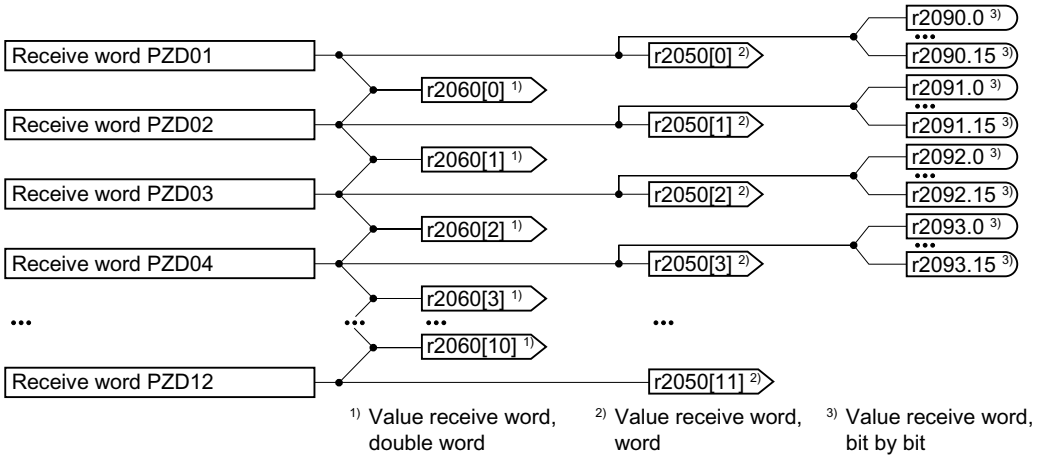


Figure 3-12 Interconnection of the receive words

The telegrams use - with the exception of telegram 999 (free interconnection) - the word-by-word transfer of send and receive data (r2050/p2051).

If you require an individual telegram for your application (e.g. for transferring double words), you can adapt one of the predefined telegrams using parameters p0922 and p2079. For details, please refer to the List Manual, function diagrams 2420 and 2472.

3.1.1 Assigning control and status words

Assigning control and status of words is specified in part by the definitions in the PROFIdrive profile, Version 4.2 for the "Closed-loop speed control" operating mode; the other part is assigned depending on the particular manufacturer.

A more detailed description of the individual control and status words is provided in the following sections.

If you require an individual assignment for your application, you can adapt one of the existing control and status words using p0922 and p2079.

Extend telegrams and change signal interconnection (Page 28)

3.1.1.1 Control and status word 1

Control word 1 is preassigned as follows:

- Telegrams 1, 2, 3 and 4:
  - Bits 0 ... 10 corresponding to the PROFIdrive profile,
  - Bits 11... 15 manufacturer-specific
- Telegrams 7 and 9:
  - Bits 0 ... 11 corresponding to the PROFIdrive profile,
  - Bits 12 ... 15 manufacturer-specific
- Telegram 20 (VIK/NAMUR):
  - Bits 0 ... 11 corresponding to the PROFIdrive profile
  - Bits 12 ... 14 reserved
  - Bit 15 corresponding to the PROFIdrive profile

Status word 1 is preassigned as follows:

- Telegrams 1, 2, 3 and 4:
  - Bits 0 ... 10 corresponding to the PROFIdrive profile,
  - Bits 11... 15 manufacturer-specific
- Telegrams 7 and 9:
  - Bits 0 ... 13 corresponding to the PROFIdrive profile,
  - Bits 14 ... 15 manufacturer-specific
- Telegram 20 (VIK/NAMUR):
  - Bits 0 ... 11 corresponding to the PROFIdrive profile
  - Bit 12 reserved
  - Bits 13 ... 15 corresponding to the PROFIdrive profile

Control word 1 (STW1)

Bit	Meaning		Explanation	Signal interconnection in the converter
	Telegram 20	All other telegrams		
0	0 = OFF1		The motor brakes with the ramp-down time p1121 of the ramp-function generator. The converter switches off the motor at standstill.	p0840[0] = r2090.0
	0 → 1 = ON		The converter goes into the "ready" state. If, in addition, bit 3 = 1, the converter switches on the motor.	
1	0 = OFF2		Switch off the motor immediately, the motor then coasts down to a standstill.	p0844[0] = r2090.1
	1 = No OFF2		The motor can be switched on (ON command).	

Bit	Meaning		Explanation	Signal interconnection in the converter
	Telegram 20	**All other telegrams**		
2	0 = Quick stop (OFF3)		Fast stopping The motor brakes with the OFF3 ramp-down time p1135 down to standstill.	p0848[0] = r2090.2
	1 = No quick stop (OFF3)		The motor can be switched on (ON command).	
3	0 = Inhibit operation		Immediately switch-off motor (cancel pulses).	p0852[0] = r2090.3
	1 = Enable operation		Switch-on motor (pulses can be enabled).	
4	0 = Disable RFG		The converter immediately sets its ramp-function generator output to 0.	p1140[0] = r2090.4
	1 = Do not disable RFG		The ramp-function generator can be enabled.	
5	0 = Stop RFG		The output of the ramp-function generator stops at the actual value.	p1141[0] = r2090.5
	1 = Enable RFG		The output of the ramp-function generator follows the setpoint.	
6	0 = Inhibit setpoint		The converter brakes the motor with the ramp-down time p1121 of the ramp-function generator.	p1142[0] = r2090.6
	1 = Enable setpoint		Motor accelerates with the ramp-up time p1120 to the setpoint.	
7	0 → 1 = Acknowledge faults		Acknowledge fault. If the ON command is still active, the converter switches to the "switching on inhibited" state.	p2103[0] = r2090.7
8, 9	Reserved			
10	0 = No control via PLC		Converter ignores the process data from the fieldbus.	p0854[0] = r2090.10
	1 = Control via PLC		Control via fieldbus, converter accepts the process data from the fieldbus.	
11	1 = Direction reversal		Invert setpoint in the converter.	p1113[0] = r2090.11
12	Not used			
13	---[1)]	1 = MOP up	Increase the setpoint saved in the motorized potentiometer.	p1035[0] = r2090.13
14	---[1)]	1 = MOP down	Reduce the setpoint saved in the motorized potentiometer.	p1036[0] = r2090.14
15	CDS bit 0	Reserved	Changes over between settings for different operation interfaces (command data sets).	p0810 = r2090.15

[1)] If you change over from another telegram to telegram 20, then the assignment of the previous telegram is kept.

Status word 1 (ZSW1)

Bit	Meaning		Remarks	Signal interconnection in the converter
	Telegram 20	All other telegrams		
0	1 = Ready for switching on		Power supply switched on; electronics initialized; pulses locked.	p2080[0] = r0899.0
1	1 = Ready		Motor is switched on (ON/OFF1 = 1), no fault is active. With the command "Enable operation" (STW1.3), the converter switches on the motor.	p2080[1] = r0899.1
2	1 = Operation enabled		Motor follows setpoint. See control word 1, bit 3.	p2080[2] = r0899.2
3	1 = Fault active		The converter has a fault. Acknowledge fault using STW1.7.	p2080[3] = r2139.3
4	1 = OFF2 inactive		Coast down to standstill is not active.	p2080[4] = r0899.4
5	1 = OFF3 inactive		Quick stop is not active.	p2080[5] = r0899.5
6	1 = Switching on inhibited active		It is only possible to switch on the motor after an OFF1 followed by ON.	p2080[6] = r0899.6
7	1 = Alarm active		Motor remains switched on; no acknowledgement is necessary.	p2080[7] = r2139.7
8	1 = Speed deviation within the tolerance range		Setpoint / actual value deviation within the tolerance range.	p2080[8] = r2197.7
9	1 = Master control requested		The automation system is requested to accept the converter control.	p2080[9] = r0899.9
10	1 = Comparison speed reached or exceeded		Speed is greater than or equal to the corresponding maximum speed.	p2080[10] = r2199.1
11	1 = current or torque limit reached	1 = torque limit reached	Comparison value for current or torque has been reached or exceeded.	p2080[11] = r0056.13 / r1407.7
12	---[1)]	1 = Holding brake open	Signal to open and close a motor holding brake.	p2080[12] = r0899.12
13	0 = Alarm, motor overtemperature		--	p2080[13] = r2135.14
14	1 = Motor rotates clockwise		Internal converter actual value > 0	p2080[14] = r2197.3
	0 = Motor rotates counter-clockwise		Internal converter actual value < 0	
15	1 = CDS display	0 = Alarm, converter thermal overload	--	p2080[15] = r0836.0 / r2135.15

[1)] If you change over from another telegram to telegram 20, then the assignment of the previous telegram is kept.

3.1.1.2 Control and status word 2

Control word 2 is preassigned as follows:

- Bits 0 ... 11 manufacturer-specific
- Bits 12 ... 15 corresponding to the PROFIdrive profile

Status word 2 is preassigned as follows:

- Bits 0 ... 11 manufacturer-specific
- Bits 12 ... 15 corresponding to the PROFIdrive profile

Control word 2 (STW2)

Bit	Meaning		Signal interconnection in the converter
	Telegrams 2, 3 and 4	**Telegrams 9, 110 and 111**	
0	1 = drive data set selection DDS bit 0		p0820[0] = r2093.0
1	1 = drive data set selection DDS bit 1		p0821[0] = r2093.1
2...6	Reserved		
7	1 = parking axis is selected		p0897 = r2093.7
8	1 = travel to fixed stop active	Reserved	p1545[0] = r2093.8
9...11	Reserved		
12	1 = master sign-of-life bit 0		p2045 = r2050[3]
13	1 = master sign-of-life bit 1		
14	1 = master sign-of-life bit 3		
15	1 = master sign-of-life bit 4		

Status word 2 (ZSW2)

Bit	Meaning	Signal interconnection in the converter
0	1 = Drive data set DDS effective, bit 0	p2081[0] = r0051.0
1	1 = Drive data set DDS effective, bit 1	p2081[1] = r0051.1
2...4	Reserved	
5	1 = Alarm class bit 0	p2081[5] = r2139.11
6	1 = alarm class bit 1	p2081[6] = r2139.12
7	Reserved	
8	1 = travel to fixed stop active	p2081[8] = r1406.8
9	Reserved	
10	1 = pulses enabled	p2081[10] = r0899.11
11	Reserved	
12	Device sign-of-life bit 0	Internally interconnected
13	Device sign of life bit 1	
14	Device sign of life bit 2	
15	Device sign of life bit 3	

3.1.1.3 Control and status word 3

Control word 3 is preassigned as follows:

- Bits 0 ... 15 manufacturer-specific

Status word 3 is preassigned as follows:

- Bits 0 ... 15 manufacturer-specific

Control word 3 (STW3)

Bit	Meaning Telegram 350	Explanation	Signal interconnection in the converter [1]
0	1 = fixed setpoint bit 0	Selects up to 16 different fixed setpoints.	p1020[0] = r2093.0
1	1 = fixed setpoint bit 1		p1021[0] = r2093.1
2	1 = fixed setpoint bit 2		p1022[0] = r2093.2
3	1 = fixed setpoint bit 3		p1023[0] = r2093.3
4	1 = DDS selection bit 0	Changes over between settings for different motors (drive data sets).	p0820 = r2093.4
5	1 = DDS selection bit 1		p0821 = r2093.5
6	Not used		
7	Not used		
8	1 = technology controller enable	--	p2200[0] = r2093.8
9	1 = enable DC braking	--	p1230[0] = r2093.9
10	Not used		
11	1 = Enable droop	Enable or inhibit speed controller droop.	p1492[0] = r2093.11
12	1 = torque control active 0 = speed control active	Changes over the control mode for vector control.	p1501[0] = r2093.12
13	1 = no external fault 0 = external fault is active (F07860)	--	p2106[0] = r2093.13
14	Not used		
15	1 = CDS bit 1	Changes over between settings for different operation interfaces (command data sets).	p0811[0] = r2093.15

[1] If you switch from telegram 350 to a different one, then the converter sets all interconnections p1020, ... to "0". Exception: p2106 = 1.

Status word 3 (ZSW3)

Bit	Meaning	Description	Signal interconnection in the converter
0	1 = DC braking active	--	p2051[3] = r0053
1	1 = |n_act | > p1226	Absolute current speed > stationary state detection	
2	1 = |n_act | > p1080	Absolute actual speed > minimum speed	
3	1 = i_act ≧ p2170	Actual current ≥ current threshold value	
4	1 = |n_act | > p2155	Absolute actual speed > speed threshold value 2	
5	1 = |n_act | ≦ p2155	Absolute actual speed < speed threshold value 2	
6	1 = |n_act | ≧ r1119	Speed setpoint reached	
7	1 = DC link voltage ≦ p2172	Actual DC link voltage ≦ threshold value	
8	1 = DC link voltage > p2172	Actual DC link voltage > threshold value	
9	1 = ramp-up or ramp-down completed	Ramp-function generator is not active.	
10	1 = technology controller output at the lower limit	Technology controller output ≦ p2292	
11	1 = technology controller output at the upper limit	Technology controller output > p2291	
12	Not used		
13	Not used		
14	Not used		
15	Not used		

3.1.2 NAMUR message word

Function description

Fault word according to the VIK-NAMUR definition (MELD_NAMUR)

Bit	Significance	P No.
0	1 = Control Unit signals a fault	p2051[5] = r3113
1	1 = line fault: Phase failure or inadmissible voltage	
2	1 = DC link overvoltage	
3	1 = Power Module fault, e.g. overcurrent or overtemperature	
4	1 = converter overtemperature	
5	1 = ground fault/phase fault in the motor cable or in the motor	
6	1 = motor overload	
7	1 = communication error to the higher-level control system	
8	1 = fault in a safety-relevant monitoring channel	
10	1 = fault in the internal converter communication	
11	1 = line fault	
15	1 = other fault	

3.1.3 Control and status word, encoder

Telegrams 3 and 4 allow the higher-level control system to directly access the encoder.

Direct access is necessary, if the higher-level control is responsible for the closed-loop position control for the drive.

If you enable the "Basic positioner" position control in the converter, then telegrams 3 and 4 cannot be selected, and the converter handles the encoder control.

Control word encoder (G1_STW and G2_STW)

Bit	Meaning	Explanation		Signal interconnection in the converter
		Bit 7 = 0	Bit 7 = 1	
0	Function 1	1 = search for reference cam 1 with a positive start direction	1 = request flying referencing to the rising edge of reference cam 1	Telegram 3: Encoder 1: p0480[0] = r2050[4] Telegram 4: Encoder 1: p0480[0] = r2050[4] encoder 2: p0480[1] = p2050[9] Telegram 102: Encoder 1: p0480[0] = r2050[5] Telegram 103: Encoder 1: p0480[0] = r2050[5] encoder 2: p0480[1] = p2050[10]
1	Function 2	1 = search for reference cam 1 with a negative start direction	1 = request flying referencing to the falling edge of reference cam 1	
2	Function 3	1 = search for reference cam 2 with a positive start direction	1 = request flying referencing to the rising edge of reference cam 2	
3	Function 4	1 = search for reference cam 2 with a negative start direction	1 = request flying referencing to the falling edge of reference cam 2	
4	Command bit 0	1 = activate the function requested using bit 0 ... 3		
5	Command bit 1	1 = read the value requested using bit 0 ... 3		
6	Command bit 2	Reserved		
7	Mode	1 = flying referencing 0 = search for reference cams		
8 ... 12	Reserved	---		
13	Cyclic absolute value	1 = request for the cyclic transfer of the position actual value in G1_XIST2 or G2_XIST2		
14	Parking	1 = request to park the encoder		
15	Acknowledge	0 → 1 = acknowledge encoder fault		

Status word encoder (G1_ZSW and G2_ZSW)

Bit	Meaning	Explanation		Signal interconnection in the converter
		Bit 7 = 0	Bit 7 = 1	
0	Function 1	1 = search for reference cam 1 is active	1 = flying referencing to the rising edge of reference cam 1 is active	Telegram 3: Encoder 1: p2051[4] = r0481[0] Telegram 4: Encoder 1: p2051[4] = r0481[0] encoder 2: p2051[9] = r0481[1] Telegram 102: Encoder 1: p2051[5] = r0481[0] Telegram 103: Encoder 1: p2051[5] = r0481[0] encoder 2: p2051[10] = r0481[0]
1	Function 2	1 = search for reference cam 1 is active	1 = flying referencing to the falling edge of reference cam 1 is active	
2	Function 3	1 = search for reference cam 2 is active	1 = flying referencing to the rising edge of reference cam 2 is active	
3	Function 4	1 = search for reference cam 2 is active	1 = flying referencing to the falling edge of reference cam 2 is active	
4	Status value 1	1 = position actual value is at reference cam 1	1 = flying referencing to the rising edge of reference cam 1 has been completed	
5	Status value 2	1 = position actual value is at reference cam 1	1 = flying referencing to the falling edge of reference cam 1 has been completed	
6	Status value 3	1 = position actual value is at reference cam 2	1 = flying referencing to the rising edge of reference cam 2 has been completed	
7	Status value 4	1 = position actual value is at reference cam 2	1 = flying referencing to the falling edge of reference cam 2 has been completed	
8	Reference cam 1	1 = reference cam 1 supplies a high signal 0 = reference cam 1 supplies a low signal		
9	Reference cam 2	1 = reference cam 2 supplies a high signal 0 = reference cam 2 supplies a low signal		
10	Reserved	---		
11	Acknowledge	1 = acknowledge encoder fault is active		
12	Reserved	---		
13	Cyclic absolute value	1 = the position actual value is in G1_XIST2 or G2_XIST2.		
14	Parking	1 = the encoder is parked		
15	Fault	1 = the encoder indicates its actual fault in r0483		

3.1.4 Position actual value of the encoder

G1_XIST1 and G2_XIST1

In the factory setting, the converter transfers the encoder position actual value with a fine resolution of 11 bits to the higher-level control system.

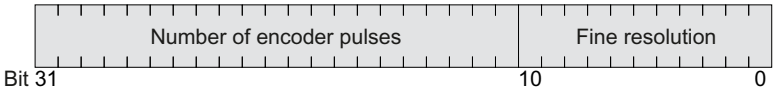


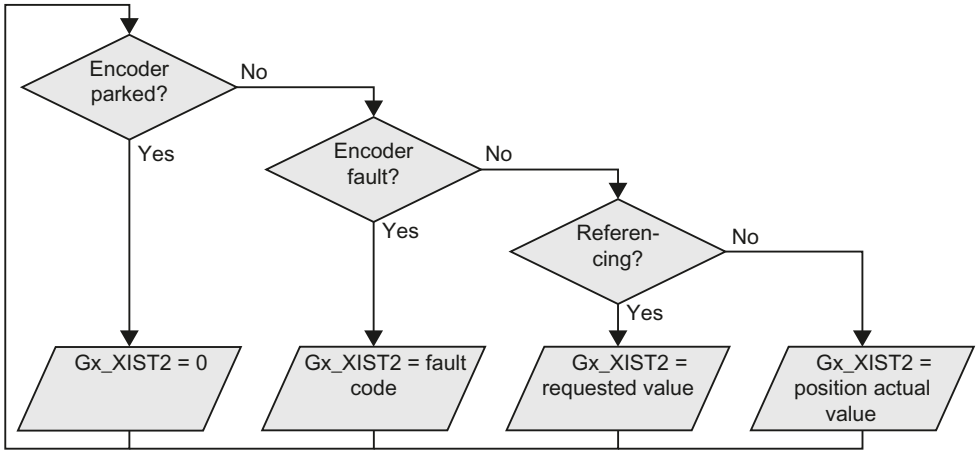
Figure 3-13 G1_XIST1 and G2_XIST1

The transferred encoder signal has the following properties:

- After the converter power supply has been switched on, the encoder signal = 0.
- The higher-level control must be able to handle a counter overflow of the encoder signal.

G1_XIST2 and G2_XIST2

In G1_XIST2 or G2_XIST2, the converter transfers different values to the higher-level control system:



Encoder x parked	Gx_ZSW.14 = 1
Encoder fault x	Gx_ZSW.15 = 1
Referencing encoder x	Gx_ZSW.4 = 1 or Gx_ZSW.5 = 1 or Gx_ZSW.6 = 1 or Gx_ZSW.7 = 1

Figure 3-14 G1_XIST2 and G2_XIST2

The converter transfers the position values in the same format (encoder pulse number and fine resolution) the same as G1_XIST1 and G2_XIST1.

Table 3-1 Fault code

No.	Explanation	Possible cause
1	Encoder fault	One or more encoder faults. Observe the converter message.
2	Zero-mark monitoring	---

No.	Explanation	Possible cause
3	Encoder parking canceled	Parking was already requested.
4	Search for reference canceled	• Encoder has no zero mark (reference mark). • Reference mark 2, 3 or 4 was requested. • Switchover to "Flying measurement" was requested during search for reference. • Command "Read value x" requested during search for reference mark. • Inconsistent position measured value with distance-coded reference marks.
5	Retrieve reference value canceled	• More than four values were requested. • No value requested. • Requested value is not available.
6	Flying referencing canceled	• Reference cam has not been configured • During "Flying referencing" a changeover was made to search for reference. • During "Flying referencing" a request was issued "Read value x".
7	Retrieve measured value canceled	• More than one value was requested. • No value requested. • Requested value is not available. • Encoder is parked.
8	Position actual value transfer canceled	• No absolute encoder available. • Alarm bit in the absolute value protocol set.
3841	Encoder does not support the function	---

3.1.5 Extend telegrams and change signal interconnection

Overview

When you have selected a telegram, the converter interconnects the corresponding signals with the fieldbus interface. Generally, these interconnections are locked so that they cannot be changed. However, with the appropriate setting in the converter, the telegram can be extended or even freely interconnected.

Function description

Interconnection of send data and receive data

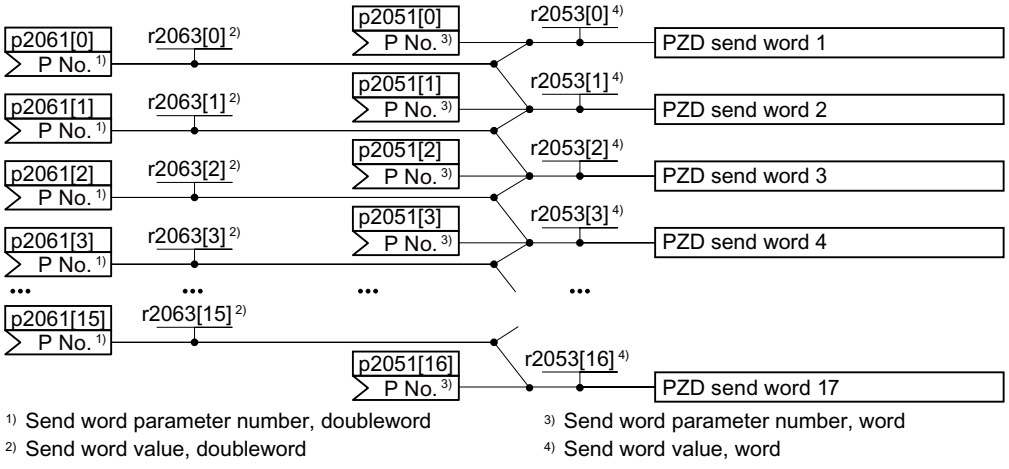


Figure 3-15 Interconnection of the send data

In the converter, the send data are available in the "Word" format (p2051) - and in the "Double word" format (p2061). If you set a specific telegram, or you change the telegram, the converter automatically interconnects parameters p2051 and p2061 with the appropriate signals.

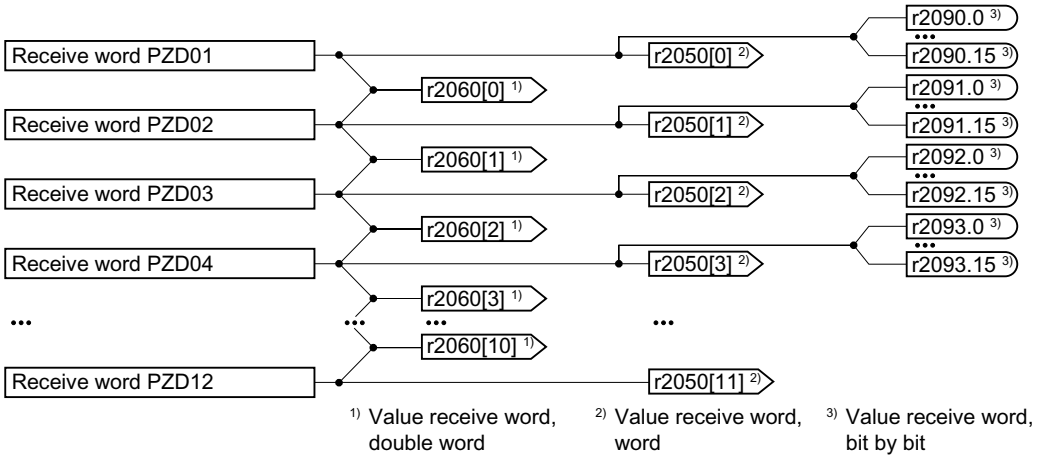


Figure 3-16 Interconnection of the receive data

The converter saves the receive data as follows:

- "Word" format in r2050
- "Double word" format in r2060
- Bit-by-bit in r2090 ... r2093

Extending a telegram: Procedure

1. Set p0922 = 999.
2. Set parameter p2079 to the value of the corresponding telegram.
3. Interconnect additional send words and receive words with signals of your choice via parameters r2050 and p2051.

You have extended a telegram.
❒

Freely interconnecting signals in the telegram: Procedure

1. Set p0922 = 999.
2. Set p2079 = 999.
3. Interconnect additional send words and receive words with signals of your choice via parameters r2050 and p2051.

You have freely interconnected a telegram.
❒

Example

You wish to extend telegram 1 to 6 send words and 6 receive words. You want to test the extension by initiating that the converter returns each receive word back to the higher-level control system.

Procedure

1. p0922 = 999
2. p2079 = 1
3. p2051[2] = r2050[2]
4. ...
5. p2051[5] = r2050[5]
6. Test the telegram length for received and sent words:
   - r2067[0] = 6
   - r2067[1] = 6

You wish to extend telegram 1 to 6 send words and 6 receive words.
❒

Parameter

Number	Name	Factory setting
p0922	PROFIdrive PZD telegram selection	1
r2050[0...11]	CO: PROFIdrive PZD receive word	-
p2051[0...16]	CI: PROFIdrive PZD send word	0 or dependent on the converter
r2053[0...16]	PROFIdrive diagnostics send PZD word	-
r2060[0...10]	CO: PROFIdrive PZD receive double word	-
p2061[0...15]	CI: PROFIdrive PZD send double word	0
r2063[0...15]	PROFIdrive diagnostics PZD send double word	-
r2067	PZD maximum interconnected [0] Receive (r2050, r2060) [1] Send (p2051, p2061)	-
p2079	PROFIdrive PZD telegram selection extended	1
p2080[0...15]	BI: Binector-connector converter, status word 1	[0] 899 [1] 899.1 [2] 899.2 [3] 2139.3 [4] 899.4 [5] 899.5 [6] 899.6 [7] 2139.7 [8] 2197.7 [9] 899.9 [10] 2199.1 [11] 1407.7 [12] 0 [13] 2135.14 [14] 2197.3 [15] 2135.15
r2090.0...15	BO: PROFIdrive receive PZD1 bit by bit	-
r2091.0...15	BO: PROFIdrive PZD2 receive bit-serial	-
r2092.0...15	BO: PROFIdrive PZD3 receive bit-serial	-
r2093.0...15	BO: PROFIdrive PZD4 receive bit-serial	-

Parameter

Parameter	Description	
p0922	**PROFIdrive telegram selection**	
	999:	Free telegram (message frame) configuration

Parameter	Description	
p2079	**PROFIdrive PZD telegram selection extended**	
	The following values apply if you have still not enabled the "Basic positioner" function in the converter:	
	1: 2: 3: 4: 20: 350: 352: 353: 354: 999:	Standard telegram 1, PZD-2/2 Standard telegram 2, PZD-4/4 Standard telegram 3, PZD-5/9 Standard telegram 4, PZD-6/14 Standard telegram 20, PZD-2/6 SIEMENS telegram 350, PZD-4/4 SIEMENS telegram 352, PZD-6/6 SIEMENS telegram 353, PZD-2/2, PKW-4/4 SIEMENS telegram 354, PZD-6/6, PKW-4/4 Free telegram configuration
	The following values apply if you have enabled the "Basic positioner" function in the converter:	
	7: 9: 110: 111: 999:	Standard telegram 7, PZD-2/2 Standard telegram 9, PZD-10/5 SIEMENS telegram 110, PZD-12/7 SIEMENS telegram 111, PZD-12/12 Free telegram configuration
r2050[0...11]	**PROFIdrive PZD receive word** Received PZD (setpoints) in the word format	
p2051[0...16]	**PROFIdrive PZD send word** Sent PZD (actual values) in the word format	

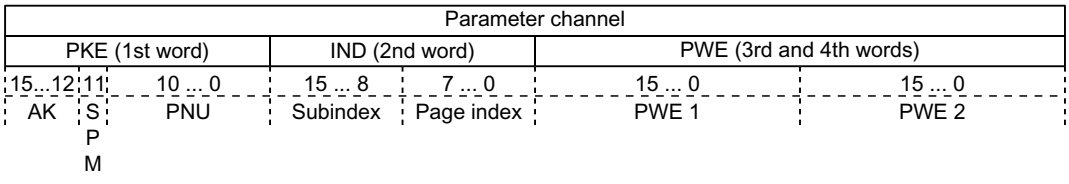
For further information about receive and send words, refer to the function block diagrams 2468 and 2470 in the List Manual.

Overview of the manuals (Page 217)

3.1.6 Data structure of the parameter channel

Overview

The parameter channel allows parameter values to be cyclically read and written to.



Structure of the parameter channel:

- PKE (1st word)
  - Type of task (read or write).
  - Bit 11 is reserved and is always assigned 0.
  - Parameter number
- IND (2nd word)
  - Parameter index
- PWE (3rd and 4th word)
  - Parameter value

Function description

AK: Request and response ID

Table 3-2 Request identifiers, control → converter

AK	Description	Response identifier	
		positive	negative
0	No request	0	7 / 8
1	Request parameter value	1 / 2	7 / 8
2	Change parameter value (word)	1	7 / 8
3	Change parameter value (double word)	2	7 / 8
4	Request descriptive element [1]	3	7 / 8
6 [2]	Request parameter value (field) [1]	4 / 5	7 / 8
7 [2]	Change parameter value (field, word) [1]	4	7 / 8
8 [2]	Change parameter value (field, double word) [1]	5	7 / 8
9	Request number of field elements	6	7 / 8

[1] The required element of the parameter is specified in IND (2nd word).

[2] The following request IDs are identical: 1 ≡ 6, 2 ≡ 7 and 3 ≡ 8.
We recommend that you use identifiers 6, 7 and 8.

Table 3-3 Response identifiers, converter → control

AK	Description
0	No response
1	Transfer parameter value (word)
2	Transfer parameter value (double word)
3	Transfer descriptive element [1]
4	Transfer parameter value (field, word) [2]
5	Transfer parameter value (field, double word) [2]
6	Transfer number of field elements

AK	Description
7	Converter cannot process the request. In the most significant word of the parameter channel, the converter sends an error number to the control, refer to the following table.
8	No master controller status / no authorization to change parameters of the parameter channel interface

[1] The required element of the parameter is specified in IND (2nd word).

[2] The required element of the indexed parameter is specified in IND (2nd word).

Table 3-4 Error numbers for response identifier 7

No.	Description
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 subindex 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 error value that cannot be changed)
0B hex	**No master control** (change request but with no master control, see also p0927)
0C hex	**Keyword missing**
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)
65 hex	**Parameter number is currently deactivated** (depending on the mode of the converter)
66 hex	**Channel width is insufficient** (communication channel is too small for response)
68 hex	**Illegal parameter value** (parameter can only assume certain values)
6A hex	**Request not included / task is not supported** (the valid request identifications can be found in table "Request identifications controller → converter")
6B hex	**No change access for a controller that is enabled**. (The operating state of the converter prevents a parameter change)
86 hex	**Write access only for commissioning (p0010 = 15)** (operating state of the converter 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 converter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

PNU (parameter number) and page index

Parameter number	PNU	Page index
0000 ... 1999	0000 ... 1999	0 hex
2000 ... 3999	0000 ... 1999	80 hex
6000 ... 7999	0000 ... 1999	90 hex
8000 ... 9999	0000 ... 1999	20 hex
10000 ... 11999	0000 ... 1999	A0 hex
20000 ... 21999	0000 ... 1999	50 hex
30000 ... 31999	0000 ... 1999	F0 hex
60000 ... 61999	0000 ... 1999	74 hex

Subindex

For indexed parameters, the parameter index is located in subindex as hexadecimal value.

PWE: Parameter value or connector

Parameter values or connectors can be located in the PWE.

Table 3-5 Parameter value or connector

	PWE 1	PWE 2	
Parameter value	Bit 15 ... 0	Bit 15 ... 8	Bit 7 ... 0
	0	0	8-bit value
	0	16-bit value	
	32-bit value		
Connector	Bit 15 ... 0	Bit 15 ... 10	Bit 9 ... 0
	Number of the connector	3F hex	The index or bit field number of the connector

Examples

Read request: Read out serial number of the Power Module (r7841[2])

To obtain the value of indexed parameter r7841, you must fill the parameter channel with the following data:

- **PKE, Bit 12 ... 15 (AK): = 6** (request parameter value (field))
- **PKE, Bit 0 ... 10 (PNU): = 1841** (parameter number without offset)
  Parameter number = PNU + offset (page index)
  (7841 = 1841 + 6000)
- **IND, bit 8 ... 15 (subindex): = 2** (index of parameter)

- **IND, bit 0 ... 7 (page index): = 90 hex** (offset 6000 corresponds to 90 hex)
- Because you want to read the parameter value, words 3 and 4 in the parameter channel for requesting the parameter value are irrelevant. They should be assigned a value of 0, for example.

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 8	7 ... 0
AK		Parameter number	Subindex	Page index	Parameter value	Parameter value	Parameter value
0 1 1 0	0	1 1 1 0 0 1 1 0 0 0 1	0 0 0 0 0 0 1 0	1 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Figure 3-17 Parameter channel for read request from r7841[2]

Write request: Change restart mode (p1210)

The restart mode is inhibited in the factory setting (p1210 = 0). In order to activate the automatic restart with "acknowledge all faults and restart for an ON command", p1210 must be set to 26:

- **PKE, bit 12 ... 15 (AK): = 7** (change parameter value (field, word))
- **PKE, bit 0 ... 10 (PNU): = 4BA hex** (1210 = 4BA hex, no offset, as 1210 < 1999)
- **IND, bit 8 ... 15 (subindex): = 0 hex** (parameter is not indexed)
- **IND, bit 0 ... 7 (page index): = 0 hex** (offset 0 corresponds to 0 hex)
- **PWE1, bit 0 ... 15: = 0 hex**
- **PWE2, Bit 0 ... 15: = 1A hex** (26 = 1A hex)

Parameter channel						
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 0
AK		Parameter number	Subindex	Page index	Parameter value (bit 16 ... 31)	Parameter value (bit 0 ... 15)
0 1 1 1	0	1 0 0 1 0 1 1 1 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0

Figure 3-18 A parameter channel to activate the automatic restart with p1210 = 26

Write request: Assign digital input 2 with the function ON/OFF1 (p0840[1] = 722.2)

In order to link digital input 2 with ON/OFF1, you must assign parameter p0840[1] (source, ON/OFF1) the value 722.2 (DI 2). To do this, you must fill the parameter channel as follows:

- **PKE, bit 12 ... 15 (AK): = 7 hex** (change parameter value (field, word))
- **PKE, bit 0 ... 10 (PNU): = 348 hex** (840 = 348 hex, no offset, as 840 < 1999)
- **IND, bit 8 ... 15 (subindex): = 1 hex** (CDS1 = Index 1)
- **IND, bit 0 ... 7 (page index): = 0 hex** (offset 0 corresponds to 0 hex)
- **PWE1, Bit 0 ... 15**: = **2D2 hex** (722 = 2D2 hex)

- **PWE2, Bit 10 ... 15: = 3F hex** (drive object - for SINAMICS G120, always 63 = 3f hex)
- **PWE2, Bit 0 ... 9: = 2 hex** (Index of Parameter (DI 2 = 2))

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 10	9 ... 0
AK		Parameter number	Subindex	Page index	Parameter value	Drive Object	Index
0111	0	01101001000	00000001	00000000	0000001011010010	111111	0000000010

Figure 3-19 Parameter channel to assign digital input 2 with ON/OFF1

Function description

AK: Request and response ID

Table 3-6 Request identifiers, control → converter

AK	Description	Response identifier	
		positive	**negative**
0	No request	0	7 / 8
1	Request parameter value	1 / 2	7 / 8
2	Change parameter value (word)	1	7 / 8
3	Change parameter value (double word)	2	7 / 8
4	Request descriptive element 1)	3	7 / 8
6 2)	Request parameter value (field) 1)	4 / 5	7 / 8
7 2)	Change parameter value (field, word) 1)	4	7 / 8
8 2)	Change parameter value (field, double word) 1)	5	7 / 8
9	Request number of field elements	6	7 / 8

1) The required element of the parameter is specified in IND (2nd word).

2) The following request IDs are identical: 1 ≡ 6, 2 ≡ 7 and 3 ≡ 8.
We recommend that you use identifiers 6, 7 and 8.

Table 3-7 Response identifiers, converter → control

AK	Description
0	No response
1	Transfer parameter value (word)
2	Transfer parameter value (double word)
3	Transfer descriptive element 1)
4	Transfer parameter value (field, word) 2)
5	Transfer parameter value (field, double word) 2)
6	Transfer number of field elements

AK	Description
7	Converter cannot process the request. In the most significant word of the parameter channel, the converter sends an error number to the control, refer to the following table.
8	No master controller status / no authorization to change parameters of the parameter channel interface

[1] The required element of the parameter is specified in IND (2nd word).

[2] The required element of the indexed parameter is specified in IND (2nd word).

Table 3-8 Error numbers for response identifier 7

No.	Description
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 subindex 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 error value that cannot be changed)
0B hex	**No master control** (change request but with no master control, see also p0927.)
0C hex	**Keyword missing**
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)
65 hex	**Parameter number is currently deactivated** (depending on the mode of the converter)
66 hex	**Channel width is insufficient** (communication channel is too small for response)
68 hex	**Illegal parameter value** (parameter can only assume certain values)
6A hex	**Request not included / task is not supported** (the valid request identifications can be found in table "Request identifications controller → converter")
6B hex	**No change access for a controller that is enabled**. (The operating state of the conerter prevents a parameter change)
86 hex	**Write access only for commissioning (p0010 = 15)** (operating state of the converter 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 converter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

PNU (parameter number) and page index

Parameter number	PNU	Page index
0000 ... 1999	0000 ... 1999	0 hex
2000 ... 3999	0000 ... 1999	80 hex
6000 ... 7999	0000 ... 1999	90 hex
8000 ... 9999	0000 ... 1999	20 hex
10000 ... 11999	0000 ... 1999	A0 hex
20000 ... 21999	0000 ... 1999	50 hex
29000 ... 29999	0000 ... 1999	70 hex
30000 ... 31999	0000 ... 1999	F0 hex
60000 ... 61999	0000 ... 1999	74 hex

Subindex

For indexed parameters, the parameter index is located in subindex as hexadecimal value.

PWE: Parameter value or connector

Parameter values or connectors can be located in the PWE.

Table 3-9 Parameter value or connector

	PWE 1	PWE 2	
Parameter value	Bit 15 ... 0	Bit 15 ... 8	Bit 7 ... 0
	0	0	8-bit value
	0	16-bit value	
	32-bit value		
Connector	Bit 15 ... 0	Bit 15 ... 10	Bit 9 ... 0
	Number of the connector	3F hex	The index or bit field number of the connector

Examples

Read request: Read out serial number of the Power Module (r7841[2])

To obtain the value of the indexed parameter r7841, you must fill the telegram of the parameter channel with the following data:

- **PKE, Bit 12 ... 15 (AK): = 6** (request parameter value (field))
- **PKE, Bit 0 ... 10 (PNU): = 1841** (parameter number without offset)
  Parameter number = PNU + offset (page index)
  (7841 = 1841 + 6000)
- **IND, bit 8 ... 15 (page index): = 2** (index of parameter)

- **IND, bit 0 ... 7 (subindex): = 90 hex** (offset 6000 corresponds to 90 hex)
- Because you want to read the parameter value, words 3 and 4 in the parameter channel for requesting the parameter value are irrelevant. They should be assigned a value of 0, for example.

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 8	7 ... 0
AK		Parameter number	Page index	Subindex	Parameter value	Parameter value	Parameter value
0 1 1 0	0	1 1 1 0 0 1 1 0 0 0 1	0 0 0 0 0 0 1 0	1 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Figure 3-20 Telegram for a read request from r7841[2]

PNU (parameter number) and page index

Parameter number	PNU	Page index
0000 ... 1999	0000 ... 1999	0 hex
2000 ... 3999	0000 ... 1999	80 hex
6000 ... 7999	0000 ... 1999	90 hex
8000 ... 9999	0000 ... 1999	20 hex
10000 ... 11999	0000 ... 1999	A0 hex
20000 ... 21999	0000 ... 1999	50 hex
30000 ... 31999	0000 ... 1999	F0 hex
60000 ... 61999	0000 ... 1999	74 hex

Subindex

For indexed parameters, the parameter index is located in subindex as hexadecimal value.

PWE: Parameter value or connector

Parameter values or connectors can be located in the PWE.

Table 3-10 Parameter value or connector

	PWE 1	PWE 2	
Parameter value	Bit 15 ... 0	Bit 15 ... 8	Bit 7 ... 0
	0	0	8-bit value
	0	16-bit value	
	32-bit value		
Connector	Bit 15 ... 0	Bit 15 ... 10	Bit 9 ... 0
	Number of the connector	3F hex	The index or bit field number of the connector

3.1.6.1 Application examples

Read request: Read out serial number of the Power Module (p7841[2])

To obtain the value of the indexed parameter p7841, you must fill the telegram of the parameter channel with the following data:

- **PKE, Bit 12 ... 15 (AK): = 6** (request parameter value (field))
- **PKE, Bit 0 ... 10 (PNU): = 1841** (parameter number without offset)
  Parameter number = PNU + offset (page index)
  (7841 = 1841 + 6000)
- **IND, bit 8 ... 15 (subindex): = 2** (index of parameter)
- **IND, bit 0 ... 7 (page index): = 90 hex** (offset 6000 corresponds to 90 hex)
- Because you want to read the parameter value, words 3 and 4 in the parameter channel for requesting the parameter value are irrelevant. They should be assigned a value of 0, for example.

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 10	9 ... 0
AK		Parameter number	Subindex	Page index	Parameter value	Drive object	Index
0 1 1 0	0	1 1 1 0 0 1 1 1 0 0 0 1	0 0 0 0 0 0 1 0	1 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

Figure 3-21 Telegram for a read request from p7841[2]

Write request: Change restart mode (p1210)

The restart mode is inhibited in the factory setting (p1210 = 0). In order to activate the automatic restart with "acknowledge all faults and restart for an ON command", p1210 must be set to 26:

- **PKE, bit 12 ... 15 (AK): = 7** (change parameter value (field, word))
- **PKE, bit 0 ... 10 (PNU): = 4BA hex** (1210 = 4BA hex, no offset, as 1210 < 1999)
- **IND, bit 8 ... 15 (subindex): = 0 hex** (parameter is not indexed)
- **IND, bit 0 ... 7 (page index): = 0 hex** (offset 0 corresponds to 0 hex)
- **PWE1, bit 0 ... 15: = 0 hex**
- **PWE2, Bit 0 ... 15: = 1A hex** (26 = 1A hex)

Parameter channel						
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 0
AK		Parameter number	Subindex	Page index	Parameter value (bit 16 ... 31)	Parameter value (bit 0 ... 15)
0 1 1 1	0	1 0 0 1 0 1 1 1 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0

Figure 3-22 Telegram, to activate the automatic restart with p1210 = 26

Write request: Assign digital input 2 with the function ON/OFF1 (p0840[1] = 722.2)

In order to link digital input 2 with ON/OFF1, you must assign parameter p0840[1] (source, ON/OFF1) the value 722.2 (DI 2). To do this, you must populate the telegram of the parameter channel as follows:

- **PKE, bit 12 ... 15 (AK): = 7 hex** (change parameter value (field, word))
- **PKE, bit 0 ... 10 (PNU): = 348 hex** (840 = 348 hex, no offset, as 840 < 1999)

- **IND, bit 8 ... 15 (subindex): = 1 hex** (CDS1 = Index 1)
- **IND, bit 0 ... 7 (page index): = 0 hex** (offset 0 corresponds to 0 hex)
- **PWE1, Bit 0 ... 15**: = **2D2 hex** (722 = 2D2 hex)
- **PWE2, Bit 10 ... 15: = 3F hex** (drive object - for SINAMICS G120, always 63 = 3f hex)
- **PWE2, Bit 0 ... 9: = 2 hex** (Index of Parameter (DI 2 = 2))

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 10	9 ... 0
AK		Parameter number	Subindex	Page index	Parameter value	Drive Object	Index
0 1 1 1	0	0 1 1 0 1 0 0 1 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0	1 1 1 1 1 1	0 0 0 0 0 0 0 0 1 0

Figure 3-23 Telegram, to assign DI 2 with ON/OFF1

Example

Application example, "Read and write to parameters"

Further information is provided on the Internet:

Reading and writing parameters cyclically via PROFIBUS (https://support.industry.siemens.com/cs/ww/en/view/29157692)

3.1.7 Device-to-device communication

"Direct data exchange" is sometimes called "device-to-device communication" or "data exchange broadcast". Here, devices exchange data without any direct involvement of the master.

Example: A converter uses the actual speed value of another converter as its speed setpoint.

Definitions

- **Publisher:** Device, which sends data for direct data exchange.
- **Subscriber:** Device, which receives the data for direct data exchange from the publisher.
- **Links and access points** define the data that is used for direct data exchange.

Restrictions

- Direct data exchange in the current firmware version is only possible for converters with PROFIBUS communication.
- A maximum of 12 PZDs are permissible for each drive.
- A maximum of four links are possible from one subscriber to one or several publishers.

Configuring device-to-device communication

Procedure

1. In the control, define:
   - Which converters operate as publisher (sender) or subscriber (receiver)?
   - Which data or data areas do you use for direct data exchange?
2. In the converter, define:
   How does the subscriber process the data transferred using direct data exchange?

You have now configured device-to-device communication.
❒

3.2 PROFIDRIVE profile - Acyclic communication

The converter supports the following types of acyclic communication:

- For PROFIBUS:
  acyclic communication via data set 47
- For PROFINET:
  acyclic communication via B02E hex and B02F hex

The maximum data length per request is 240 bytes.

Note

Values in italics

Values in italics in the following tables mean that you have to adjust these values for a specific request.

Reading parameter values

Table 3-11 Request to read parameters

Data block	Byte n	Bytes n + 1	n
Header	**Reference** *01 hex ... FF hex*	**01 hex**: Read job	0
	01 hex (ID of drive objects, at G120 always = 1)	**Number of parameters (m)**	2
Address, parameter 1	**Attribute** *10 hex:* Parameter value *20 hex:* Parameter description	**Number of the indices** *00 hex ... EA hex* (For parameters without index: 00 hex)	4
	Parameter number *0001 hex ... FFFE hex*		6
	Number of the 1st index *0000 hex ... FFFE hex* (for parameters without index: 0000 hex)		8
	...		...
Address, parameter 2	...		...
...	...		...
Address, parameter m	...		...

Table 3-12 Converter response to a read request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a read request)	**01 hex**: Converter has executed the read request. **81 hex**: Converter was not able to completely execute the read request.	0
	01 hex (ID of drive objects, at G120 always = 1)	**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 *0A hex*: OctetString *0D hex*: TimeDifference *34 hex*: TimeOfDay without date indication *35 hex*: TimeDifference with date indication *36 hex*: TimeDifference without date indication *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 3-13 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 (ID of drive objects, at G120 always = 1)	**Number of parameters (m)** *01 hex ... 27 hex*	2
Address, parameter 1	**10 hex**: Parameter value	**Number of indices** *00 hex ... EA hex* (00 hex and 01 hex are equivalents)	4
	Parameter number *0001 hex ... FFFF hex*		6
	Number of the 1st index *0000 hex ... FFFE 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 *0A hex:* Octet String *0D hex:* Time Difference *34 hex*: TimeOfDay without date indication *35 hex*: TimeDifference with date indication *36 hex*: TimeDifference without date indication *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 3-14 Response, if the converter has executed the change request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a change request)	**02 hex** (change request successful)	0
	01 hex (ID of drive objects, at G120 always = 1)	**Number of parameters** (identical to a change request)	2

Table 3-15 Response if the converter 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:** (Converter was not able to completely execute the write request)	0
	01 hex (ID of drive objects, at G120 always = 1)	**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 01 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	...		

Data block	Byte n	Bytes n + 1	n
...	...		...
Values, parameter m	...		

Error values

Table 3-16 Error value in the parameter response

Error value 1	Significance
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)
20 hex	**Parameter text cannot be changed**
21 hex	**Service is not supported** (illegal or not support request ID).
6B hex	**A change request for a controller that has been enabled is not possible**. (The converter rejects the change request because the motor is switched on. Observe the "Can be changed" parameter attribute (C1, C2, U, T) in the parameter list. Manuals and technical support (Page 217)
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 converter is ready (p0010 = 0).**
72 hex	**Change request is only possible for a parameter reset (restore to factory setting) (p0010 = 30).**

Error value 1	Significance
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 during download.**
81 hex	**Change request is not possible during download.**
82 hex	**Accepting the master control is inhibited via 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	**Converter does not accept a change request** (converter is busy with internal calculations. See parameter r3996 in the parameter list. Manuals and technical support (Page 217)
85 hex	**No access methods defined**.
86 hex	**Write access only during commissioning of the data records (p0010 = 15)** (operating status of the converter 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 converter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

3.3 PROFIdrive profile - Diagnostic channels

The converters provide the diagnostics standardized for PROFIBUS and PROFINET. This means that it is possible to directly output faults and alarms at an HMI (control system screen).

Here, PROFINET offers more functions than PROFIBUS

- PROFIBUS: Faults without component assignment
- PROFINET: Faults and alarms with component assignment

The fault and alarm messages are saved in the converter in the following parameters

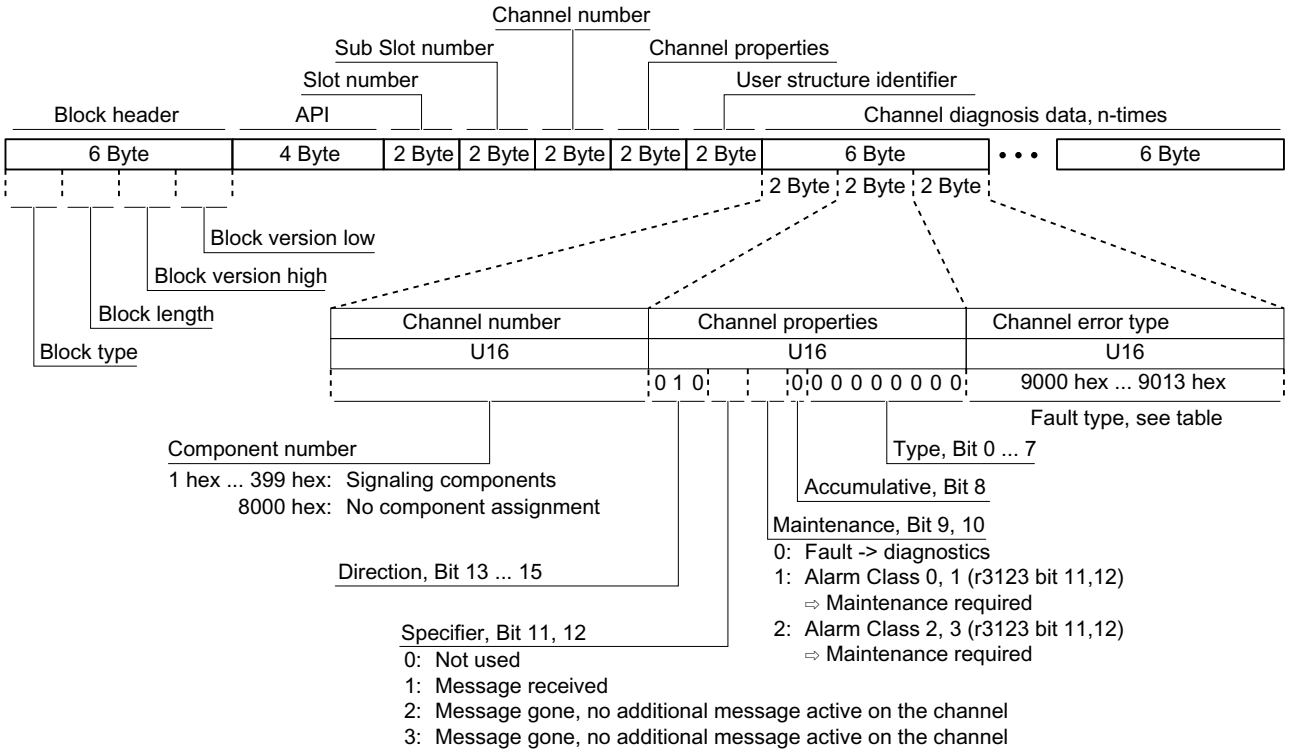
- r0947[0 ... 63]: Fault number
- r2122[0 ... 63]: Alarm code
- r3120[0 ... 63]: Components which are involved with the fault (only for PROFINET)
- r3121[0 ... 63]: Components which are involved with the alarm (only for PROFINET)

The converter transfers the messages in the sequence in which they occurred

The control generates the time stamp when the messages are received

3.3.1 Diagnostics with PROFINET

PROFINET uses the channel diagnostics to transfer PROFIdrive message classes.



9000 hex	Hardware/software error	900A hex	Position/speed actual value incorrect or not available
9001 hex	Network fault	900B hex	Internal (DRIVE-CLiQ) communication error
9002 hex	Supply voltage fault	900C hex	Infeed faulted
9003 hex	DC link overvoltage	900D hex	Braking module faulted
9004 hex	Power electronics faulted	900E hex	Line filter faulted
9005 hex	Overtemperature of the electronic components	900F hex	External measured value / signal state outside the permissible range
9006 hex	Ground fault / inter-phase short circuit	9010 hex	Application / technological function faulted
9007 hex	Motor overload	9011 hex	Error in the parameterization / configuration / commissioning procedure
9008 hex	Communication error to the higher-level control system	9012 hex	General drive fault
9009 hex	Safety monitoring channel has identified an error	9013 hex	Auxiliary unit faulted

Figure 3-24 Structure of the channel diagnostics

Reading out diagnostics data

The control requests the diagnostics data from the converter using "Read data set", e.g. using a read record with index 800C hex.

The following rules apply:

- 1 Message block (=ChannelDiagnosisData)
  if (one or several) faults of the same message class are detected at the converter
- n message blocks
  if at the converter, n faults of different message classes are detected

Further information is provided in the Internet: To access this link, you must be a member of PROFIBUS and PROFINET International (PI).

PROFINET IO specification (http://www.profibus.com/nc/download/specifications-standards/downloads/profinet-io-specification/display/)

3.3.2 Diagnostics with PROFIBUS

The following objects belonging to a diagnostics message in PROFIBUS

- **Standard diagnostics**
  - Sequence: Always at the first position of the message
  - Length is always 6 bytes
- **Identifier-related diagnostics**
  - Sequence: At the second, third or fourth position
  - Identification using the header,
  - For SINAMICS G120, the length is always 2 bytes
- **Status messages/module status**
  - Sequence: At the second, third or fourth position
  - Identification using the header
  - Length for SINAMICS G120:
    - 5 bytes for configuration using GSD
    - 6 bytes when configuring using the object library
- **Channel-related diagnostics**
  - Sequence: At the second, third or fourth position
  - Identification using the header
  - Length is always 3 bytes
- **Diagnostics alarm with DS0 / DS1**
  - Sequence: Always at the last position of the message
  - Slot-specific The current state of the slot responsible for the message is transferred.

Note

Precondition for diagnostics via PROFIBUS

The master must operate in the DPV1 mode for diagnostics via PROFIBUS.

Standard diagnostics

Byte No.	Name	Bit							
		7	6	5	4	3	2	1	0
1	Station status 1	Master_ Lock = 0	Prm_ Fault	0	Not supported	Ext_Diag	Cfg_Fault	Station_ not_ Ready	Station_ Non_ Exist = 0
2	Station status 2	0	0	Sync_ Mode	Freeze_ Mode	WD_ON	0	Start_ Diag = 0	Prm_Req
3	Station status 3	Ext_Diag_ Overflow	0	0	0	0	0	0	0
4		Master_Add							
5		Ident_Number (HighByte) of the slave							
6		Ident_Number (LowByte) of the slave							

The following values are decisive for the diagnostics:

- **Ext_Diag**: Group signal for diagnostics in the device:
  - 0: No fault is active
  - 1: At least one alarm or fault is active
- **Ext_Diag_Overflow:**
  Display for the diagnostics overflow in the device (for more than 240 bytes)

Identifier-related diagnostics

Byte No.	Name	Bit							
		7	6	5	4	3	2	1	0
1	Header byte	Header 0	1	Block length 2 ... 32 For SINAMICS G120 always = 2					
2	Bit structure	KB_7	KB_6	KB_5	KB_4	KB_3	KB_2	KB_1	KB_0
• • •					• • •				• • •
m	Bit structure	...	...	KB_n+1	KB_n	...	...	...	...

The identifier-related diagnostics provides a bit (KB_n) for each slot allocated when configuring the device. If a diagnostics message is active at a slot, then its KB_n = 1.

For G120 only one slot is allocated:

- KB_0 when configuring with the GSD
- KB_3 when configuring with the object manager

Status messages, module status

Byte No.	Name	Bit 7	6	5	4	3	2	1	0
1	Header byte	Header 0	0	Block length 2 ... 32 For SINAMICS G120 = 5 or 6					
2	Module status	82 hex (status block)							
3	Slot	0							
4	Specifier	0							
5	Slot structure	Slot_4		Slot_3		Slot_2		Slot_1	
• • •				• • •				• • •	
m	Slot structure	...		Slot n		...		...	

For G120, independent of the status, for all slots "00" is always output, i.e. valid user data.

Channel-related diagnostics

Byte No.	Name	Bit 7	6	5	4	3	2	1	0
n	Header byte	Header 1	0	Module number 0 ... 63					
n+1	Bit structure	Input / Output 1	1	0 - no component assignment					
n+2	Bit structure	Channel type - unspecific 0	0	0	Message classes, see the following table				

2	Undervoltage	22	Motor overload
3	Overvoltage	23	Commun. with controller faulted
9	Error	24	Safety monit. Detected an error
16	Hardware/software error	25	Act. Position/speed value error
17	Line supply/filter faulted	26	Internal communication faulted
18	DC link overvoltage	27	Infeed faulted
19	Power electronics faulted	28	Braking controller faulted
20	Electronic component overtemp.	29	External signal state error
21	Ground/phase fault detected	30	Application/function faulted

When multiple faults are allocated at one converter with the same message class, then only one message is displayed.

Diagnostics alarm with DS0 / DS1

Byte No.	Name	Bit							
		7	6	5	4	3	2	1	0
1	Header bytes	Header 0	0	Block length = 15					
2		0	Diagnostics alarm = 1						
3		0	Slot number 0 ... 244 For SINAMICS G120 1 or 4: Configuration via GSD = 1 Configuration via library = 4						
4		0	0 ... 31, sequence number				Add_Ack	Alarm Specifier [1]	
5	DS0 byte 0	0	0	0	0	0 [2]	0	0 [3]	0 [4]
6	DS0 byte 1	0	0	0	1 [5]	0 [6]	0 [6]	0 [6]	0 [6]
7	DS0 byte 2	0	0	0	0	0	0	0	0
8	DS0 byte 3	0	0	0	0	0	0	0	0
9	Info byte 1	Mixed	= 45 hex (ChannelTypeID = SINAMICS)						
10	Info byte 2	= 24 (diagnostic bits / channel)							
11	Info byte 3	= 1 (one channel signals)							
12	Channel Error Vector	0	0	0	0	0	0	0	1
13	Channel-related diagnostics (channel 0)	Err 7	Err 6	Err 5	Err 4	Err 3	Err 2	Err 1	Err 0
14		Err 15	Err 14	Err 13	Err 12	Err 11	Err 10	Err 9	Err 8
15		0	0	0	0	Err 19	Err 18	Err 17	Err 16

1) Alarm specifier
   1: Fault is active and the slot is not OK
   2: Fault is resolved and the slot is OK
   3: Fault is resolved and the slot is not OK

2) Channel fault present
   0: No fault is active
   1: Fault is active

3) Internal fault
   0: No fault is active
   1: Fault is active

4) Module fault
   0: No fault is active
   1: Fault is active

5) Channel information present
   1: DS1 exists

6) Type of class module = 0011 (distributed)

A table with the message classes is provided in the List Manual of the converter.

Overview of the manuals (Page 217)

3.4 Identification & maintenance data (I&M)

I&M data

The converter supports the following identification and maintenance (I&M) data.

I&M data	Format	Explanation	Associated parameters	Example for the content
I&M0	u8[64] PROFIBUS u8[54] PROFINET	Converter-specific data, read only	-	See below
I&M1	Visible String [32]	Plant/system identifier	p8806[0 ... 31]	"ak12-ne.bo2=fu1"
	Visible String [22]	Location code	p8806[32 ... 53]	"sc2+or45"
I&M2	Visible String [16]	Date	p8807[0 ... 15]	"2013-01-21 16:15"
I&M3	Visible String [54]	Any comment	p8808[0 ... 53]	-
I&M4	Octet String[54]	Check signature to track changes for Safety Integrated. This value can be changed by the user. The test signature is reset to the value generated by the machine if p8805 = 0 is used.	p8809[0 ... 53]	Values of r9781[0] and r9782[0]

When requested, the converter transfers its I&M data to a higher-level control or to a PC/PG with installed STEP 7 or TIA Portal.

I&M0

Designation	Format	Example for the content	Valid for PROFINET	Valid for PROFIBUS
Manufacturer-specific	u8[10]	00 ... 00 hex	---	✓
MANUFACTURER_ID	u16	42d hex (=Siemens)	✓	✓
ORDER_ID	Visible String [20]	"6SL3246-0BA22-1FA0"	✓	✓
SERIAL_NUMBER	Visible String [16]	"T-R32015957"	✓	✓
HARDWARE_REVISION	u16	0001 hex	✓	✓
SOFTWARE_REVISION	char, u8[3]	"V" 04.70.19	✓	✓
REVISION_COUNTER	u16	0000 hex	✓	✓
PROFILE_ID	u16	3A00 hex	✓	✓
PROFILE_SPECIFIC_TYPE	u16	0000 hex	✓	✓
IM_VERSION	u8[2]	01.02	✓	✓
IM_SUPPORTED	bit[16]	001E hex	✓	✓

3.5 S7 communication

Communication via the S7 protocol facilitates the following:

- Access to the converter with Startdrive.
- Remote maintenance of the converter with Startdrive across network boundaries.
  Remote maintenance across network boundaries (https://support.industry.siemens.com/cs/ww/en/view/97550333)
- Control of the converter directly via SIMATIC Panels via PROFIBUS or PROFINET without higher-level control.
  Directly accessing a SINAMICS G120 converter from a SIMATIC panel (Page 55)

Note

Number of S7 protocol connections

The converter supports four S7 protocol connections. Two of these are required for Startdrive. Each of the remaining two are available for access to the converter via SIMATIC Panels.

3.5.1 Directly accessing a SINAMICS G120 converter from a SIMATIC panel

Example of direct access to the converter via a SIMATIC panel

You want to use the SIMATIC panel to do the following:

- Switch the converter on and off
- Enter a setpoint
- Display the actual value and status

Requirements

You have installed the following software packages on your computer and made the following settings:

- WINCCflex 2008 SP1 or higher
- Startdrive
- You have now configured the converter in Startdrive.
- Converter and panel are connected with one another via PROFIBUS or PROFINET.
- The same baud rates are set in the converter and in the panel.
- The bus address configured in WinCC flexible matches the bus address of the converter.

Adjusting settings in the converter

Procedure

1. **Make the following settings and enables so that the converter can accept commands from the panel:**
   - Set the two signal sources for OFF2 (p0844 and p0845) to 1:
     p0844 = 1
     p0845 = 1
   - Set the two signal sources for OFF3 (p0848 and p0849) to 1:
     p0848 = 1
     p0849 = 1
   - Set the enables for the ramp-function generator:
     p1140 = 1
     p1141 = 1
   - Set the setpoint enable:
     p1142 = 1
2. **Adjust the parameters for the ON/OFF1 command from the SIMATIC panel**
   - Set p0840[0] = 2094.0
     In doing so, you connect up the ON/OFF1 command using the Bit 0 of the BICO transformer 2094. The signal source for this parameter is p2099.
   - Set p2099[0] = p2900.
     In doing so, you give the ON/OFF1 command by setting P2900 = 1 (ON) or 0 (OFF1)
3. **Set parameters for the setpoint default**
   - Set:
     p1070 = 1001 (fixed setpoint 1 as setpoint)
     p1016 = 1 (direct selection of the speed setpoint)
     p1020 = 1 (fixed speed setpoint selection, bit 0)
4. **Actual value and status word**
   No further settings are required on the converter side for displaying the actual speed value (r0021) and the status word (r0052).

You have now made the settings in the converter.
❒

Settings at the SIMATIC panel

Procedure

1. **Configure the connection using WinCC flexible**
   - Enter a name for the connection
   - Set the value in the "Active" column to "On".
   - Select **"SIMATIC S7 300/400"** as the communication driver.
   - Set the value in the "Online" column to "On".
2. **Make the following settings for the configured connection:**
   - Select the interface (IF1 B for PROFIBUS, "Ethernet" for PROFINET).
   - Set the baud rate for PROFIBUS.
   - Assign a bus address (PROFIBUS) or an IP address (PROFINET).
   - Select S7ONLINE as the access point.
   - If no other control is connected to the converter, select "Only master on bus".
   - Select cyclical operation.
3. **ON/OFF1:**
   - Create a variable for the parameter p2900, which refers to the address "Data block 2900 with the data word DBD 0 (data type double word)":
     DB2900.DBD 0
     You can switch ON/OFF1 on the panel using one or two buttons.
4. **Setpoint**
   - Create a variable for the parameter 1001, which refers to the address "Data block 1001 with the data word DBD 0 (data type real)":
     DB1001.DBD 0
     You can display it through an I/O field.
5. **Actual value display**
   - Create a variable for the parameter r0021, which refers to the address "Data block 21 with the data word DBD 0 (data type real)":
     DB21.DBD 0
     You can display it through an I/O field.
6. **Status display**
   - Create a variable for the parameter r0052, which refers to the address "Data block 52 with the data word DBW 0 (data type word)":
     DB52.DBW 0
     You can display it through an I/O field with a binary display, for example.

You have now made the most important settings in the SIMATIC panel.
❒

General information for accessing converter parameters

You must create a variable with the following structure for each parameter that you want to display or change using the SIMATIC panel: DBX DBY Z

- X: Data block number ≙ Parameter number
- Y: Data type (can be found in the parameter list)
- Z: Data block offset ≙ Parameter index

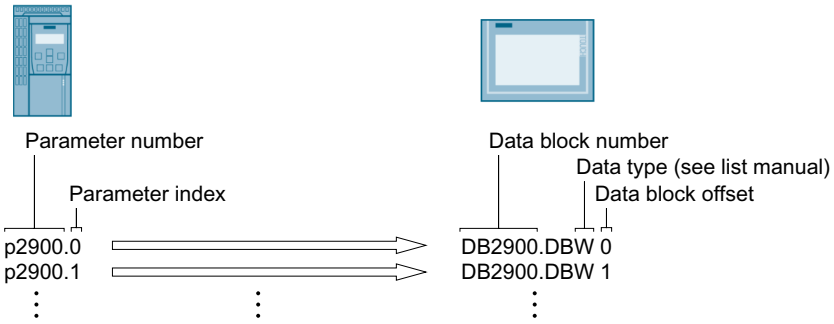


Figure 3-25 Accessing converter parameters using a SINAMICS G120 as an example

3.6 Communication via PROFINET

You can either integrate the converter in a PROFINET network or communicate with the converter via Ethernet.

The converter in PROFINET IO operation

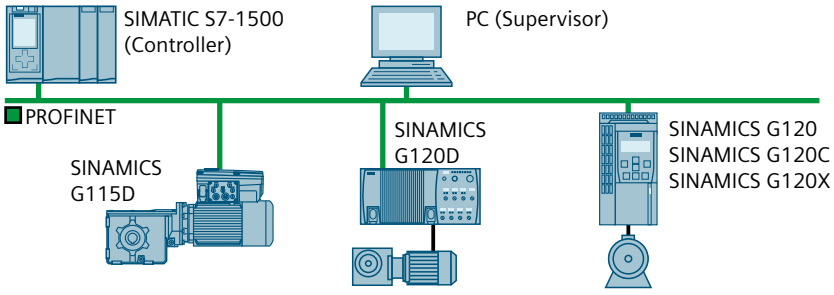


Figure 3-26 The converter in PROFINET IO operation (examples)

The converter supports the following functions:

- RT
- IRT: The converter forwards the clock synchronism, but does not support clock synchronism.
- MRP: Media redundancy, impulsed with 200 ms. Precondition: Ring topology
  With MRP, you get an uninterrupted switchover if you set the failure monitoring time to a value > 200 ms.
- MRPD: Media redundancy, bumpless. Precondition: IRT and the ring topology created in the control

- Diagnostic alarms in accordance with the error classes specified in the PROFIdrive profile.
- Device replacement without removable data storage medium: The replacement converter is assigned the device name from the IO controller, not from its memory card or from the programming device.
- Shared Device for converters that support PROFIsafe.

The converter as Ethernet node

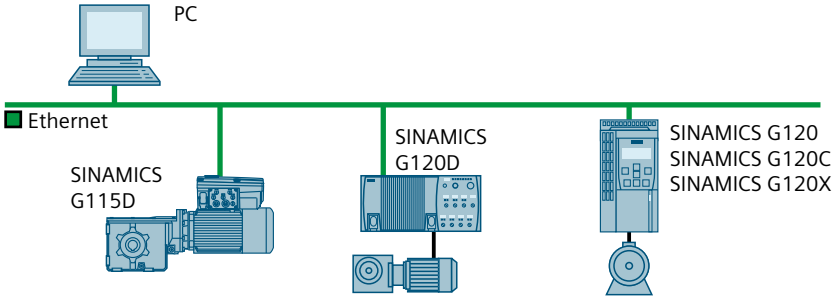


Figure 3-27 The converter as Ethernet node (examples)

Further information on PROFINET

Further information on PROFINET can be found on the Internet:

- PROFINET – the Ethernet standard for automation (http://w3.siemens.com/mcms/automation/en/industrial-communications/profinet/Pages/Default.aspx)
- PROFINET system description (https://support.industry.siemens.com/cs/ww/en/view/19292127)

3.6.1 Converter with PROFINET interface

The pin assignment and the connectors that you require for your converter are listed in the following tables.

You can implement either a ring or line-type topology using the two sockets at the converter. You only require one of the two sockets at the beginning and end of a line.

You can use switches to realize other topologies.

Table 3-17 Assignment table

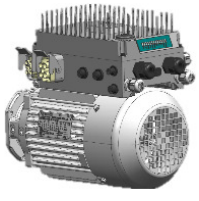
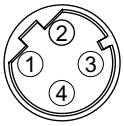
Converter/Control Unit		Connection via		
		X150 P1/ X150 P2 (RJ45)  8 ... 1	X03/X04 (RJ45)  1 ... 8	X03/X04 (M12) 
	G120			
	• CU230P-2 PN	x		
	• CU240E-2 PN	x		
	• CU240E-2 PN-F	x		
	• CU250S-2 PN	x		
	G120C			
	• G120C PN	x		
	G120D			
	• CU240D-2 PN			x
	• CU240D-2 PN-F			x
	• CU250D-2 PN-F			x
	• CU240D-2 PN-F [PP]		x	
	• CU250D-2 PN-F [PP]		x	
	G115D			
	• G115D PN			x (X150 P1/P2)

Table 3-18 Connector pin assignments

Signal		X150 P1/ X150 P2 (RJ45)  8 ... 1	X03/X04 (RJ45)  1 ... 8	X03/X04, X150 P1/P2 (M12) 
TX-	Transmit data -	1	1	1
RX+,	Receive data +	3	2	2
TX+	Transmit data +	2	3	3
RX-	Receive data -	6	6	4
---		4	4	---
---		5	5	---
---		7	7	---
---		8	8	---

Recommended connector

RJ45, IP20: 6GK1901-1BB10-2Ax0

Information for assembling the SIMATIC NET Industrial Ethernet FastConnect RF45 plug 180 can be found on the Internet:

Assembly instructions for the SIMATIC NET Industrial Ethernet FastConnect RJ45 plug (http://support.automation.siemens.com/WW/view/en/37217116/133300)

3.6.2 Integrating converters into PROFINET

Note

PROFINET interface X150

The network with which interface X150 is connected must be separated from the rest of the plant network in accordance with the Defense in Depth concept. Manual access to cables and any open connections must be protected as in a control cabinet.

To connect the converter to a control system via PROFINET, proceed as follows:

Procedure

1. Integrate the converter in the bus system (e.g. ring topology) of the control using PROFINET cables and the two PROFINET sockets X150-P1 and X150-P2 or X03 and X04.
   The position of the sockets is given in the operating instructions for the converter.
   Pin assignment:  Converter with PROFINET interface (Page 60).
   The maximum permitted cable length from the previous station and to the subsequent one is 100 m.
2. Externally supply the converter with 24 V DC through terminals 31 and 32 or via X01 (X01/X02 with G115D).
   The external 24 V supply is only required if communications with the control system should also operate when the line voltage is switched off.

You have connected the converter to the control using PROFINET.

3.6.3 PROFINET IO operation

3.6.3.1 What do you have to set for communication via PROFINET?

Check the communication settings using the following table. If you answer "Yes" to these questions, you have correctly set the communication settings and can control the converter via the fieldbus.

Questions	Answer/description
Is the converter correctly connected to the bus network?	 Integrating converters into PROFINET (Page 61)
Do the IP address and device name in the converter and control match?	 Configuring communication to the control (Page 63)
Is the same telegram set in the converter as in the higher-level control?	Setting the telegram in the control
Are the signals that the converter and the control exchange via PROFINET correctly interconnected?	Interconnect signals in the converter in conformance with PROFIdrive.  PROFIDRIVE profile - Cyclic communication (Page 13)  PROFIDRIVE profile - Acyclic communication (Page 42)

Communication with the control, even when the line voltage is switched off

So that communication with the control system in your plant or system continues to function even when the line voltage is switched off, you must externally supply the converter/Control Unit with 24 V DC. To do this, use terminals 31 and 32 or connector X01 (X01/X02 with G115D). You can find additional details in the operating instructions for the converter or the Control Unit.

3.6.3.2 Configuring communication to the control

Configuring the communication using SIMATIC S7 control

If the converter is not included in the hardware library, you have the following options:

- Install the current Startdrive version.
- Install the GSDML of the converter using "Options/Manage general station description (GSD)" in the components catalog.

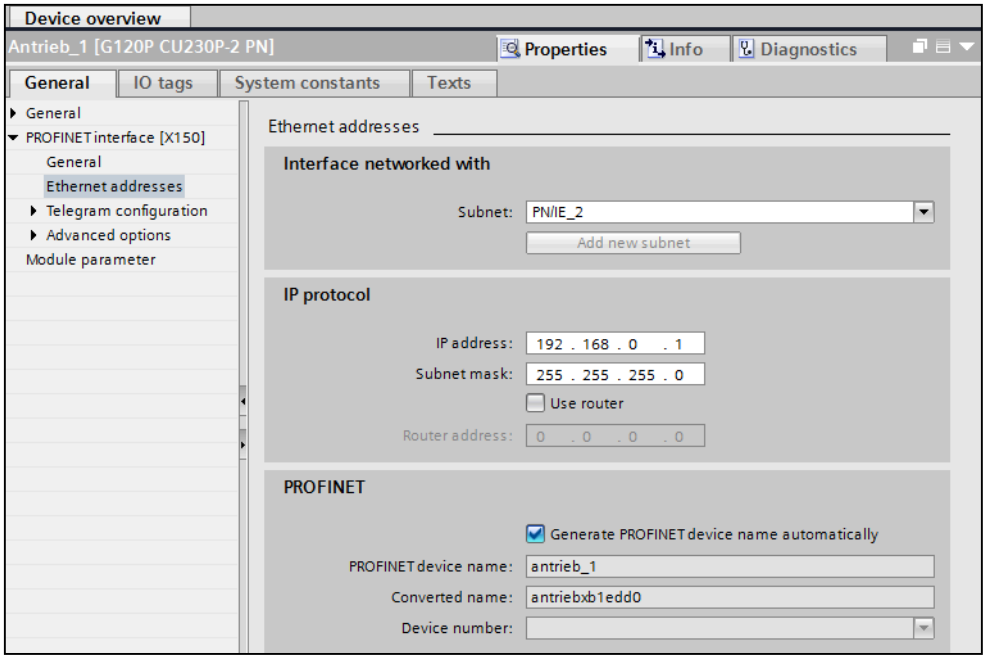
Configuring the communication using a non-Siemens control

1. Import the device file (GSDML) of the converter into the calculation tool for your control system.
2. Configure the communication.

Configuring communication with Startdrive

Proceed as follows to make the settings for communication with the control system.

- Activate the following windows in Startdrive: "View/Project tree" and "View/Inspector window".
- Open the drive in the project tree and double click on "Device configuration".
  This opens the dialog in the inspector window for setting the PROFINET interface.
- Click on "Ethernet addresses".
- Enter the appropriate values.



You have configured communication with the control system.
❒

You can enter or read out data directly via the parameter view. To do this, select the "Communication" parameter group and the "Show advanced parameters" option.

3.6.3.3 Installing GSDML

Procedure

1. Save the GSDML to your PC.
   - With Internet access:
     GSDML (https://support.industry.siemens.com/cs/ww/en/ps/13222/dl)
   - Without Internet access:
     Insert a memory card into the converter.
     Set p0804 = 12.
     The converter writes the GSDML as a zipped file (*.zip) into directory /SIEMENS/SINAMICS/DATA/CFG on the memory card.
2. Unzip the GSDML file on your computer.
3. Import the GSDML into the engineering system of the controller.

You have now installed the GSDML in the engineering system of the controller.

3.6.3.4 Activating diagnostics via the control

The converter provides the functionality to transmit fault and alarm messages (diagnostic messages) to the higher-level control according to the PROFIdrive error classes.

The functionality must be selected in the higher-level controller and activated by powering up.

3.6.4 PROFIenergy

3.6.4.1 Energy-saving mode

Overview

PROFIenergy is a standard based on PROFINET. PROFIenergy is certified and described in the PROFIenergy profile of the PNO.

The higher-level controller transfers the control commands and status queries in acyclic operation via data record 80A0 hex.

The converter supports PROFIenergy profile V1.1 and function unit class 3.

G115D, G120 and G120C converters support energy-saving mode 2.

G120D converters support energy-saving mode 1.

Function description

Behavior of the converter with active energy-saving mode 2:

- The converter outputs alarm A08800.
- The RDY LED flashes green: 500 ms on, 3 000 ms off.

- The converter does not send any diagnostic interrupts.
- If the higher-level controller goes to stop or the bus connection to the higher-level controller is interrupted, the converter exits the energy-saving mode and resumes normal operation.

Behavior of the converter with active energy-saving mode 1:

- The converter switches off the supply voltage for its digital outputs if they are not interconnected with r5613.x (displays the energy-saving mode) or are being used as safety-relevant outputs.
- The converter switches off the supply voltage of its encoders unless they are HTL encoders assigned to the position controller.

Example

You can find an application example for PROFIenergy on the Internet:

PROFIenergy - saving energy with SIMATIC S7 (https://support.industry.siemens.com/cs/ww/en/view/41986454)

Parameters

Number	Name	Factory setting
r5600	Pe energy-saving mode ID	-
r5613	CO/BO: Pe energy-saving active/inactive	-

3.6.4.2 Control commands

Function description

Command	Explanation
Start_Pause	Switches to the energy-saving mode depending on the pause duration.
Start_Pause_with_time_response	Switches to the energy-saving mode depending on the pause duration and also specifies the transition times in the command response
End_Pause	Switches from the energy-saving mode to the operating state. Cancels switching from the operating state to energy-saving mode.

Settings

- Minimum pause time: p5602
  - When the pause time sent using the command "Start_Pause" is equal to or greater than the value in p5602[1], the converter goes to energy-saving mode.
  - If the pause time is less than p5602[1], the converter rejects the command "Start_Pause" with 50 hex (no appropriate pause mode).

  If the controller sends the command "End_Pause" or "Start_Pause" with a pause time of 0, the motor cannot be switched on. An OFF1/ON command is required to switch the motor on again.
- Maximum pause time: p5606
- Disable PROFIenergy
  If you set p5611.0 = 1, you disable the response of the converter to PROFIenergy control commands. In this case, the converter rejects the "Start_Pause" command with 50 hex (no appropriate pause mode).
- Transition to energy-saving mode
  - With p5611.2 = 0, you enable the transition to energy-saving mode from operating state S1 (switching on inhibited) or S2 (ready to switch on).
  - With p5611.2 = 1, you enable the transition to energy-saving mode from operating states S3 (ready for operation) and S4 (operation).

  To do this, you must also set the following:
  - p5611.1 = 1: With the transition to energy-saving mode, the converter triggers an OFF1 command and enters the switching on inhibited state (S1).
  - p5611.1 = 0: You use p5614 to interconnect a signal source that you use to switch off the converter and place it in switching on inhibited state (S1).

3.6.4.3 Status queries

Function description

Command	Explanation
List_Energy_Saving_Modes	Returns all supported energy-saving modes
Get_Mode	Returns information about the selected energy-saving mode
PEM_Status	Returns the current PROFIenergy status
PEM_Status_with_CTTO	Returns the current PROFIenergy status together with the regular transition time to the operating state
PE_Identify	Returns the supported PROFIenergy commands
Query_Version	Returns the implemented PROFIenergy profile
Get_Measurement_List	Returns the measured value IDs that can be accessed using the "Get_Measurement_Values" command
Get_Measure-ment_List_with_object_number	Returns the measured value IDs and the associated object number that can be accessed using the "Get_Measurement_Values_with_object_number" command.

Command	Explanation
Get_Measurement_Values	Returns the measured values requested via the measured value ID
Get_Measurement_Values_with_object_number	Returns the measured values requested via the measured value ID and the object number. The object number corresponds to the drive object ID.

3.6.4.4 Error values and measured values

Function description

Table 3-19 Error values in the parameter response

Error value 1	Meaning
001 hex	Invalid Service_Request_ID
03 hex	Invalid Modifier
04 hex	Invalid Data_Structure_Identifier_RQ
06 hex	No PE energy-saving mode supported
07 hex	Response too long
08 hex	Invalid block header
50 hex	No suitable energy-saving mode available
51 hex	Time is not supported
52 hex	Impermissible PE_Mode_ID
53 hex	No switch to energy saving mode because of state operate
54 hex	Service or function temporarily not available

Table 3-20 Measured values

PROFIenergy					SINAMICS source parameters		Value range
Measured value		Accuracy		Unit			
ID	Name	Domain	Class		Number	Name	
34	Active power	1	12	W	r0032	Active power smoothed	r2004
166	Power factor	1	12	1	r0038	Power factor smoothed	0 ... 1
200	Active energy import	2	11	Wh	r0039[1]	Energy drawn	-

3.6.5 The converter with PROFINET interface as Ethernet node.

As default setting, the converter is set for PROFINET IO communication. Alternatively, you have the option of integrating the converter into an Ethernet network via the PROFINET interface.

This means that from any location in a network, you can use Startdrive to make diagnostic queries, change parameters or carry out commissioning work.

PROFINET I/O communication is not possible with the converter as Ethernet node.

Integrating a converter into an Ethernet network (assigning an IP address)

Procedure

1. Set p8924 (PN DHCP mode) = 2 or 3
   - p8924 = 2
     The DHCP server assigns the IP address based on the MAC address of the converter
   - p8924 = 3
     The DHCP server assigns the IP address based on the device name of the converter
2. Save the settings with p8925 = 2. The next time that the converter switches on, it retrieves the IP address. After this, you can address the converter as an Ethernet node.

Note

Immediate switchover without restart

The switchover to DHCP is performed immediately and without a restart if the change is carried out with the EtherNet/IP command "Set Attribute Single" (class F5 hex, attribute 3). The following options are available:

- Via an EtherNet/IP controller
- Via an EtherNet/IP commissioning tool

You have now integrated the converter into Ethernet

Displays

r8930: Device name of the converter

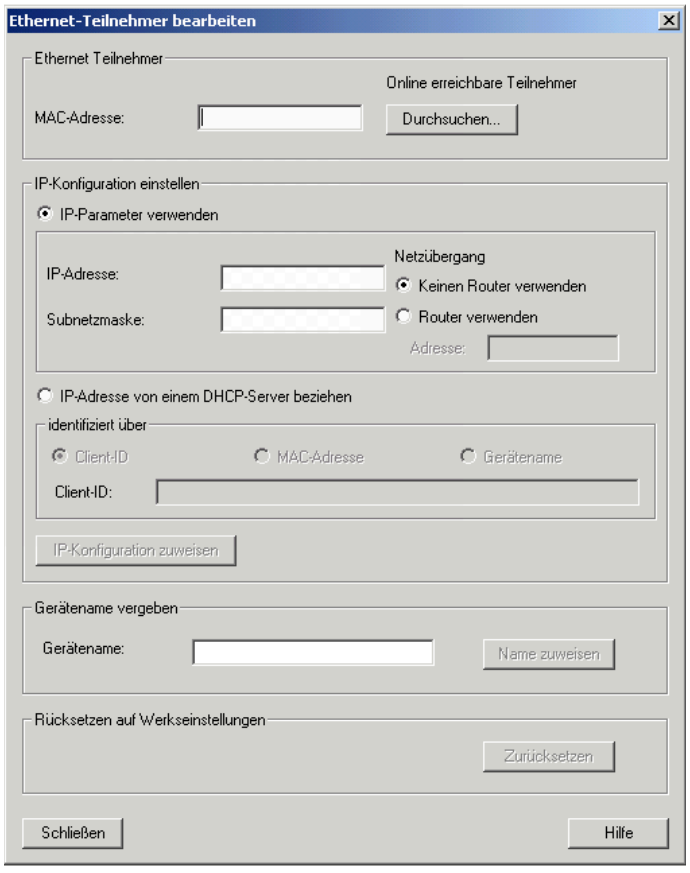
r8934: Operating mode, PN or DHCP

r8935: MAC address

Additional options of integrating converters into Ethernet

You also have the option of integrating the converter into Ethernet using Proneta or STEP 7, for example.

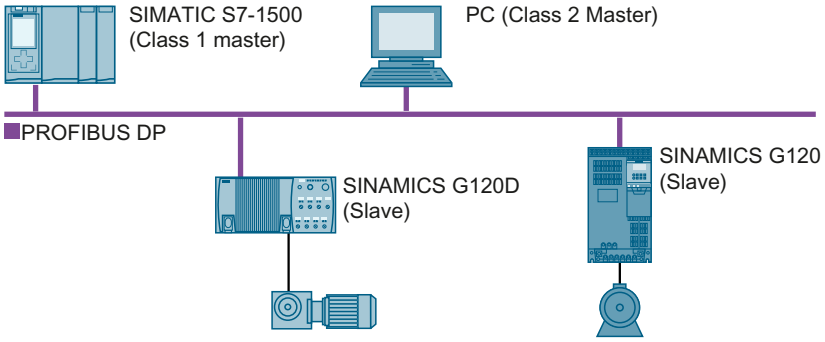
Here is the example of the "Edit Ethernet station" screen form from Step 7, which you can use to make the required settings.



See also

Overview of the manuals (Page 217)

3.7 Communication via PROFIBUS



The PROFIBUS DP interface has the following functions:

- Cyclic communication
- Acyclic communication
- Diagnostic alarms

General information on PROFIBUS DP can be found in the Internet:

- PROFIBUS information (https://support.industry.siemens.com/cs/ww/en/view/1971286)
- Installation guidelines of the PNO (http://www.profibus.com/downloads/installation-guide/)

3.7.1 Converters with PROFIBUS interface

You can find the connectors and the connector assignments of the PROFIBUS DP interface in the following tables.

You can implement a line-type topology using the two connectors at the converter. You can use switches to realize other topologies.

Table 3-21 Assignment table - connectors

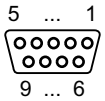
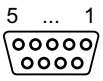
Converter/Control Unit		Connection via		
		X126 (D Sub - socket) 	X03, on (M12) 	X04, off (M12) 
	G120			
	• CU230P-2 DP	x		
	• CU240B-2 DP	x		
	• CU240E-2 DP	x		
	• CU240E-2 DP-F	x		
	• CU250S-2 DP	x		
	G120C			
	• G120C DP	x		
	G120D			
	• CU240D-2 DP		x	x
	• CU240D-2 DP-F		x	x
	• CU250D-2 DP-F		x	x

Table 3-22 Connector pin assignments

Signal	X126 (D Sub - socket) 	X03, on (M12) 	X04, off (M12) 
Shield, ground connection	1	5	5
---	2	1	1
RxD/TxD-P, receive and transmit (B/B')	3	4	4

CNTR-P, control signal	4	---	---
DGND, reference potential for data (C/C')	5	---	---
VP, supply voltage	6	---	---
---	7	3	3
RxD/TxD-N, receive and transmit (A/A')	8	2	2
---	9	---	---

Recommended PROFIBUS connectors

We recommend connectors with the following article numbers for connecting the PROFIBUS cable:

- 6GK1500-0FC10
- 6GK1500-0EA02

3.7.2 What do you have to set for communication via PROFIBUS?

Configuring PROFIBUS communication

You require the appropriate engineering system to configure PROFIBUS communication in the PROFIBUS master.

If required, load the GSD file of the converter into the engineering system.

Configuring communication to the control system (Page 75)

Setting the address

Set the address of the PROFIBUS device.

Set the PROFIBUS address (Page 77)

Setting the telegram

Set the same telegram in the converter as in the PROFIBUS master. Interconnect the telegrams in the control program of the PROFIBUS master with the signals of your choosing.

PROFIDRIVE profile - Cyclic communication (Page 13)

Application examples

You can find application examples for PROFIBUS communication on the Internet:

Controlling the speed of a SINAMICS G110M/G120/G120C/G120D with S7-300/400F via PROFINET or PROFIBUS, with Safety Integrated (via terminal) and HMI (https://support.industry.siemens.com/cs/ww/en/view/60441457)

Controlling the speed of a SINAMICS G110M / G120 (Startdrive) with S7-1500 (TO) via PROFINET or PROFIBUS, with Safety Integrated (via terminal) and HMI (https://support.industry.siemens.com/cs/ww/en/view/78788716)

3.7.3 Integrating converters into PROFIBUS

To connect the converter to a control system via PROFIBUS DP, proceed as follows:

1. Integrate the converter into the bus system (e.g. line topology) of the control using PROFIBUS cables.
   - Converters with IP20 degree of protection using socket X126
   - Converters with IP65 degree of protection (CU240D/CU250D) via X03 and X04

   The position of the socket is given in the operating instructions for the converter.
   Pin assignment: Converters with PROFIBUS interface (Page 72).
   The maximum permitted cable length to the previous station and the subsequent one is 100 m at a baud rate of 12 Mbps. You can achieve a maximum cable length of 400 m by using a maximum of 3 repeaters.
2. Externally supply the converter with 24 V DC through terminals 31 and 32 or via X01.
   The external 24 V supply is only required if communications with the control system should also operate when the line voltage is switched off.

You have now connected the converter to the control system using PROFIBUS DP.

3.7.4 Configuring communication to the control system

Configure the communication in the control system after you have connected the converter to the bus.

3.7.4.1 Configuring the communication using SIMATIC S7 control

- If the converter is listed in the component catalog in the TIA Portal, you can configure the communication in the SIMATIC control.
- If the converter is not listed in the hardware library, you can either install the newest Startdrive version or install the GSD of the converter via "Extras/GSD-Install file" in HW-Config.

3.7.4.2 Configuring the communication with a third-party control system

If you are working with a third-party control system, you must install the device file (GSD) of the converter in the control before you configure the communication.

Installing the GSD (Page 76) .

If you have installed the GSD, configure the communication. To do this, follow the documentation of your control system.

3.7.4.3 Installing the GSD

Procedure

1. Save the GSD on your PC using one of the following methods.
   - With Internet access:
     GSD (http://support.automation.siemens.com/WW/view/en/22339653/133100)
   - Without Internet access:
     Insert a memory card into the converter.
     Set p0804 = 12.
     The converter writes the GSD as zipped file (*.zip) into directory /SIEMENS/SINAMICS/DATA/CFG on the memory card.
2. Unzip the GSD file on your computer.
3. Import the GSD in the engineering system of the controller.

You have now installed the GSD file in the engineering system of the controller.

3.7.5 Set the PROFIBUS address

Valid address area: 1 ... 125

You have the following options for setting the address:

- Using the address switch on the Control Unit:

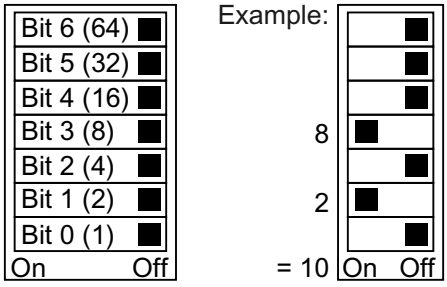


Figure 3-28 Address switch with example for bus address 10

The address switch has priority over the other settings.

- With a commissioning tool, e.g. an operator panel, via parameter p0918 (factory setting: p0918 = 126).
  It is only possible to change p0918 if an invalid address is set in the address switch.

You can find the position of the address switch in the operating instructions for the converter.

Manuals and technical support (Page 217)

Activating the changed bus address

Procedure

1. Set the address as described above.
2. Switch off the converter power supply.
3. Wait until all LEDs on the converter are dark.
4. Switch on the converter power supply again.
   Your settings become effective after switching on.

You have now set the bus address.
❒

3.8 Select telegram

Requirement

In the basic commissioning you have selected the control using PROFIBUS or PROFINET.

Telegrams for SINAMICS G120 converters

The following table shows all of the telegrams for the G120 converter.

In your converter, you have a list of telegrams for selection that are available for your particular converter.

Value p0922

- 1: Standard telegram 1, PZD-2/2 (factory setting, exceptions: CU250D and CU250S)
- 2: Standard telegramm 2, PZD-4/4
- 3: Standard telegram 3, PZD-5/9
- 4: Standard telegram 4, PZD-6/14
- 7: Standard telegram 7, PZD 2/2 (factory setting CU250D)
- 9: Standard telegram 9, PZD-10/5
- 20: Standard telegram 20, PZD-2/6
- 110: SIEMENS telegram 110, PZD-12/7
- 112: SIEMENS telegram 111, PZD-12/12
- 350: SIEMENS telegram 350, PZD-4/4
- 352: SIEMENS telegram 352, PZD-6/6
- 353: SIEMENS telegram 353, PZD-2/2, PKW-4/4
- 354: SIEMENS telegram 354, PZD-6/6, PKW-4/4
- 999: Free telegram

  Extend telegram/change signal interconnection (Page 28) (factory setting, CU250S)

For further information about telegrams:

PROFIDRIVE profile - Cyclic communication (Page 13).

Telegrams for SINAMICS G115D converters

The list of telegrams for the SINAMICS G115D converters can be found in the G115D operating instructions. See:

Manuals and technical support (Page 217)

PROFIsafe telegram selection

The settings for the PROFIsafe telegram selection are described in the "Safety Integrated" Function Manual.

Communication via EtherNet/IP

4

Overview

EtherNet/IP is real-time Ethernet, and is mainly used in automation technology.

The following options are available for integrating SINAMICS G converters into EtherNet/IP:

- You can use the SINAMICS profile.
- You can use the ODVA AC/DC drive profile.
- You can define the assemblies for the process data using the objects that are supported by the converter.

See also

Configuring communication (Page 82)

The pin assignment and the connectors that you require for your converter are listed in the following tables.

You can implement a line-type topology using the two sockets at the converter. You only require one of the two sockets at the beginning and end of a line.

You can use switches to realize other topologies.

4.1 Converters with Ethernet/IP interface

Table 4-1 Assignment table

Converter/Control Unit		Connection via		
		X150 P1/ X150 P2 (RJ45)  8 ... 1	X03/X04 (RJ45)  1 ... 8	X03/X04 (M12) 
	G120			
	• CU230P-2 PN	x		
	• CU240E-2 PN	x		
	• CU240E-2 PN-F	x		
	• CU250S-2 PN	x		

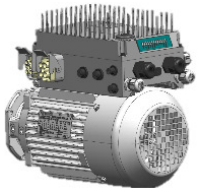
	G120C			
	• G120C PN	x		
	G120D			
	• CU240D-2 PN			x
	• CU240D-2 PN-F			x
	• CU250D-2 PN-F			x
	• CU240D-2 PN-F [PP]		x	
	• CU250D-2 PN-F [PP]		x	
	G115D			
	• G115D PN			x (X150 P1/P2)

Table 4-2 Connector pin assignments

Signal		X150 P1/ X150 P2 (RJ45)  8 ... 1	X03/X04 (RJ45)  1 ... 8	X03/X04, X150 P1/P2 (M12) 
TX-	Transmit data -	1	1	1
RX+,	Receive data +	3	2	2
TX+	Transmit data +	2	3	3
RX-	Receive data -	6	6	4
---		4	4	---
---		5	5	---
---		7	7	---
---		8	8	---

Recommended connector

RJ45, IP20: 6GK1901-1BB10-2Ax0

Information for assembling the SIMATIC NET Industrial Ethernet FastConnect RF45 plug 180 can be found on the Internet:

Assembly instructions for the SIMATIC NET Industrial Ethernet FastConnect RJ45 plug (https://support.industry.siemens.com/cs/ww/en/ps/15251/man)

4.2 Connect converter to EtherNet/IP

Overview

To connect the converter to a control system via Ethernet, proceed as follows:

Procedure

1. Connect the converter to the control system via an Ethernet cable.
2. Create an object for data exchange.
   You have the following options:
   - Load the EDS file into your controller if you want to use the ODVA profile.
     You can find the EDS file on the Internet:
     EDS (https://support.industry.siemens.com/cs/ww/de/view/78026217)
   - If your controller does not accept the EDS file, or if you wish to use the SINAMICS profile, you must create a generic module in your controller:
     Create generic I/O module (Page 97)

You have connected the converter to the control system via EtherNet/IP.
❒

Example

You can find an example showing how to connect a converter to the control system via Ethernet/IP on the Internet:

Application example (https://support.industry.siemens.com/cs/ww/en/view/82843076)

See also

EtherNet/IP (http://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/EtherNetIPLibrary/tabid/76/lng/en-US/Default.aspx)

Manuals and technical support (Page 217)

4.3 What do you need for communication via EtherNet/IP?

Check the communication settings using the following questions. If you answer "Yes" to the questions, you have correctly set the communication settings and can control the converter via the fieldbus.

- Is the converter correctly connected to the EtherNet/IP?
- Is the EDS file installed in your control system?
- Have the bus interface and IP address been correctly set?
- Have the signals that the converter and the control system exchange been correctly interconnected?

4.4 Configuring communication

Overview

EtherNet/IP is realtime Ethernet, and is mainly used in automation technology.

Function description

You must set the following parameters to configure the converter communication via EtherNet/IP:

Procedure

1. p2030 = 10
2. The following parameters must match your EtherNet configuration:
   - p8921 = IP address
   - p8922 = standard gateway
   - p8923 = subnet mask
   - p8920 = station name
3. p8925 = 2
4. Select the EtherNet/IP profile:

SINAMICS profile	ODVA AC/DC drive profile
p8980 = 0	p8980 = 1
Select the appropriate telegram using p0922. PROFIDRIVE profile - Cyclic communication (Page 13)	p0922 = 1: The converter communicates using telegram 1. Other telegrams are not possible. However, when required you can extend telegram 1. Extend telegrams and change signal interconnection (Page 28)
	When required, set the following parameters: • p8981 • p8982 • p8983

5. Switch off the converter power supply.
6. Wait until all LEDs on the converter are dark.
7. Switch on the converter power supply again.

You have now configured the converter for communication via EtherNet/IP.
❒

Parameter

Number	Name	Factory setting
p2030	Fieldbus interface protocol selection 0: no protocol ... 10: EtherNet/IP	Dependent on the converter
p8920	PN Name of Station	-
p8921	PN IP Address	0
p8922	PN Default Gateway	0
p8923	PN Subnet Mask	0
p8925	Activate PN interface configuration 0: No function 1: Reserved 2: Activate the configuration and save 3: Delete configuration	0
r8931	PN IP Address actual	-
r8932	PN Default Gateway actual	-
r8933	PN Subnet Mask actual	-
p8980	EtherNet/IP profile 0: SINAMICS 1: ODVA AC/DC	0
p8981	EtherNet/IP ODVA STOP mode 0: OFF1 1: OFF2	0
p8982	EtherNet/IP ODVA speed scaling 123: 32 124: 16 ... 128: 1 129: 0.5 ... 133: 0.03125	128
p8983	EtherNet/IP ODVA torque scaling Values the same as p8982	128

More information

EtherNet/IP objects and assemblies of the converter:

Supported objects (Page 84)

See also

Overview of the manuals (Page 217)

4.5 Supported objects

Overview

Object class		Object name	Objects required	ODVA objects	SINAMICS objects
hex	dec				
1 hex	1	Identity object	x		
4 hex	4	Assembly Object	x		
6 hex	6	Connection Manager Object	x		
28 hex	40	Motor Data Object		x	
29 hex	41	Supervisor Object		x	
2A hex	42	Drive Object		x	
32C hex	812	Siemens Drive Object			x
32D hex	813	Siemens Motor Data Object			x
F5 hex	245	TCP/IP Interface Object [1]	x		
F6 hex	246	Ethernet Link Object [1]	x		
300 hex	768	Stack Diagnostic Object		x	x
302 hex	770	Adapter Diagnostic Object		x	x
303 hex	771	Explicit Messages Diagnostic Object		x	x
304 hex	772	Explicit Message Diagnostic List Object		x	x
401 hex	1025	Parameter object		x	x

[1] These objects are part of the EtherNet/IP system management.

Identity Object, Instance Number: 1 hex

Supported services

Class
- Get Attribute all
- Get Attribute single

Instance
- Get Attribute all
- Get Attribute single
- Reset

Table 4-3 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-4 Instance Attribute

No.	Service	Type	Name	Value/explanation
1	get	UINT16	Vendor ID	1251
2	get	UINT16	Device Type - ODVA AC Drive - Siemens Drive	 02 hex 12 hex
3	get	UINT16	Product code	r0964[1]
4	get	UINT16	Revision	The versions should match the EDS file
5	get	UINT16	Status	See the following table
6	get	UINT32	Serial number	bits 0 ... 19: consecutive number; bits 20 ... 23: Production identifier bits 24 ... 27: Month of manufacture (0 = Jan, B = Dec) Bits 28 ... 31: Year of manufacture (0 = 2002)
7	get	Short String	Product name	Max. length 32 bytes

Table 4-5 Explanation of No. 5 of the previous table

Byte	Bit	Name	Description
1	0	Owned	0: Converter is not assigned to any master 1: Converter is assigned to a master
	1		Reserved
	2	Configured	0: Ethernet/IP basic settings 1: Modified Ethernet/IP settings For G120, always = 1
	3		Reserved
	4 ... 7	Extended Device Status	0: Self-test or status not known 1: Firmware update active 2: At least one I/O connection with error 3: No I/O connections 4: Incorrect configuration in the ROM 5: Fatal fault 6: At least one I/O connection is active 7: All I/O connections in the quiescent state 8 ... 15: Reserved
2	8 ... 11		Not used
	12 ... 15		Reserved

Assembly Object, Instance Number: 4 hex

Supported services

Class
- Get Attribute single

Instance
- Get Attribute single
- Set Attribute single

Table 4-6 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-7 Instance Attribute

No.	Service	Type	Name	Value/explanation
3	set	Array of UINT8	Assembly	1 byte array Supported ODVA AC/DC assemblies (Page 96)

Connection Manager Object, Instance Number: 6 hex

Supported services

Class
- Get Attribute all
- Get Attribute single

Instance
- Forward open
- Forward close
- Get Attribute single
- Set Attribute single

Table 4-8 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-9 Instance Attribute

No.	Service	Type	Name	Value/explanation
1	get	UINT16	OpenReqs	Counters
2	get	UINT16	OpenFormat Rejects	Counters
3	get	UINT16	OpenResource Rejects	Counters
4	get	UINT16	OpenOther Rejects	Counters
5	get	UINT16	CloseReqs	Counters
6	get	UINT16	CloseFormat Rejects	Counters
7	get	UINT16	CloseOther Rejects	Counters
8	get	UINT16	ConnTimeouts	Counters Number of bus errors

Motor Data Object, Instance Number 28 hex

Supported services

Class
- Get Attribute single

Instance
- Get Attribute single
- Set Attribute single

Table 4-10 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-11 Instance Attribute

No.	Service	Type	Name	Value/explanation
3	get, set	USINT	Motor Type	p0300 motor type, see the following table
6	get, set	UINT16	Rated Current	p0305 rated motor current
7	get, set	UINT16	Rated Voltage	p0304 rated motor voltage
8	get, set	UINT32	Rated Power	p0307 rated motor power
9	get, set	UINT16	Rated Frequency	p0310 rated motor frequency
10	get, set	UINT16	Rated Temperature	p0605 motor temperature threshold
11	get, set	UINT16	Max Speed	p0322 maximum motor speed
12	get, set[1)]	UINT16	Pole Count	p0314 value of p0314*2
13	get, set[2)]	UINT32	Torque Constant	p0316 motor torque constant
14	get, set	UINT32	Inertia	p0341 motor moment of inertia
15	get, set	UINT16	Base Speed	p0311 motor rated speed

1) G120C and G120P: Only "get" possible.

2) G115D: Only "get" possible.

Value in p0300		Ethernet/IP motor data object	
0	no motor	0	Non-standard motor
1	Induction motor	7	Squirrel-cage induction motor
2	Synchronous motor	3	PM synchronous motor
10	1LE1 induction motor	7	Squirrel-cage induction motor
13	1LG6 induction motor	7	Squirrel-cage induction motor
17	1LA7 induction motor	7	Squirrel-cage induction motor
19	1LA9 induction motor	7	Squirrel-cage induction motor
100	1LE1 induction motor	7	Squirrel-cage induction motor
104	1PH4 induction motor	3	PM synchronous motor
107	1PH7 induction motor	0	Non-standard motor

Value in p0300		Ethernet/IP motor data object	
108	1PH8 induction motor	5	Switched reluctance motor
200	1PH8 synchronous motor	0	Non-standard motor
204	1LE4 synchronous motor	3	PM synchronous motor
237	1FK7 synchronous motor	0	Non-standard motor
10000	Motor with DRIVE-CLiQ	0	Non-standard motor
10001	Motor with DRIVE-CLiQ 2nd D	0	Non-standard motor

Supervisor Object, Instance Number: 29 hex

Supported services

Class
- Get Attribute single

Instance
- Get Attribute single
- Set Attribute single

Table 4-12 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-13 Instance Attribute

No.	Service	Type	Name	Value/explanation
3	get, set	Bool	Run1	STW.0 operation, clockwise rotation
5	get, set	Bool	Net Control	Internal 0: Local 1: Network
6	get	UINT8	State	0: Vendor Specific 1: Startup 2: Not_Ready 3: Ready 4: Enabled 5: Stopping 6: Fault_Stop 7: Faulted
7	get	Bool	Running1	ZSW1:2 1: - (Enabled and Run1) or - (Stopping and Running1) or - (Fault_Stop and Running1) 0 = Other state

No.	Service	Type	Name	Value/explanation
9	get	Bool	Ready	ZSW1:0 1: - Ready or - Enabled or - Stopping 0 = Other state
10	get	Bool	Fault	ZSW1:3 drive fault
11	get	Bool	Warning	ZSW1:7 alarm active
12	get, set	Bool	Fault reset	STW.7 acknowledge fault
13	get	UINT16	Fault Code	r945[0] error code
14	get	UINT16	Warning Code	r2122[0] alarm code
15	get	Bool	CtlFromNet	Display from Net Control 1: Control from network 0: Local control

Drive Object, Instance Number: 2A hex

Supported services

Class
- Get Attribute single

Instance
- Get Attribute single
- Set Attribute single

Table 4-14 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-15 Instance Attribute

No.	Service	Type	Name	Value/explanation
3	get	Bool	At reference	r2197.4 (for G115D: r2197.7) 1: Speed setp - act val deviation in tolerance t_off 0: Otherwise
4	get, set	Bool	Net_reference	Internal 0: Local 1: Network (for G115D: "get" only)
6	get	UINT8	Drive_Mode	p1300 manufacturer-specific, see following table
7	get	INT	Speed Actual	Main actual value, see speed units
8	get, set	INT	Speed Ref	Main setpoint, see speed units
9	get	INT	Current Actual	r0027 absolute current actual value, smoothed

No.	Service	Type	Name	Value/explanation
10	get, set[1)]	INT	Current limit	p0323 maximum motor current
15	get	INT	Power Actual	r0032 actual active power smoothed
16	get	INT	Output voltage	r0025 output voltage smoothed
17	get	INT	Output voltage	r0072 output voltage
18	get, set	UINT16	AccelTime	p1120 ramp-function generator ramp-up time
19	get, set	UINT16	DecelTime	p1121 ramp-function generator, ramp-down time
20	get, set	UINT16	Low Speed Lim	p1080 minimum speed
21	get, set	UINT16	High Speed Lim	p1082 maximum speed
22	get, set	SINT	Speed Scale	p8982 Ethernet/IP ODVA speed scaling
29	get	Bool	Ref From Net	Internal - display of Net_Reference 0: Local 1: Network

1) G115D: Only "get" possible.

Value in p1300		Ethernet/IP motor data object	
0	U/f with linear characteristic	1	Open loop speed (frequency)
1	U/f with linear characteristic and FCC	0	Vendor-specific mode
2	U/f with parabolic characteristic		
3	U/f with parameterizable characteristic		
4	U/f with linear characteristic and ECO		
5	U/f for drives requiring a precise frequency (e.g. in the textile sector)		
6	U/f for drives requiring a precise frequency and FCC		
7	U/f for parabolic characteristic and ECO		
19	U/f with independent voltage setpoint		
20	Speed control (without encoder)	2	Closed-loop speed control
22	Torque control (without encoder)	3	Torque control

Siemens Drive Object, Instance Number: 32C hex

Supported services

Class
- Get Attribute single

Instance
- Get Attribute single
- Set Attribute single

Table 4-16 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-17 Instance Attribute

No.	Type	Service	Name	Value/explanation
2	INT16	get, set	Commissioning state	p0010 commissioning parameter filter
3 ... 18	WORD	get	STW1	STW1 bit-by-bit access: Attr.3 = STW1.0 Attr.18 = STW1.15
19	WORD	get	Main setpoint	Main setpoint
20 ... 35	WORD	get	ZSW1	ZSW1 bit-by-bit access: Attr.20 = ZSW1.0 Attr.35 = ZSW1.15
36	WORD	get	Actual Frequency	Main actual value (actual frequency)
37	REAL	get, set	Ramp Up Time	p1120[0] ramp-function generator ramp-up time
38	REAL	get, set	Ramp Down Time	p1121[0] ramp-function generator ramp-down time
39	REAL	get, set	Current Limit	p0640[0] current limit
40	REAL	get, set	Frequency MAX Limit	p1082[0] maximum speed
41	REAL	get, set	Frequency MIN Limit	p1080[0] minimum speed
42	REAL	get, set	OFF3 Ramp Down Time	p1135[0] OFF3 ramp-down time
43	UINT32 / BOOL	get, set	PID Enable	p2200[0] technology controller enable
44	REAL	get, set	PID Filter Time Constant	p2265 technology controller actual value filter time constant
45	REAL	get, set	PID D Gain	p2274 technology controller differentiation time constant
46	REAL	get, set	PID P Gain	p2280 technology controller proportional gain
47	REAL	get, set	PID I Gain	p2285 technology controller integral time
48	REAL	get, set	PID Up Limit	p2291 technology controller maximum limiting
49	REAL	get, set	PID Down Limit	p2292 technology controller minimum limiting
50	REAL	get	Speed setpoint	r0020 speed setpoint
51	REAL	get	Output Frequency	r0024 output frequency
52	REAL	get	Output Voltage	r0025 output voltage
53	REAL	get	DC Link Voltage	r0026[0] DC-link voltage
54	REAL	get	Actual Current	r0027 current actual value
55	REAL	get	Actual Torque	r0031 torque actual value
56	REAL	get	Output power	r0032 actual active power value
57	REAL	get	Motor Temperature	r0035[0] motor temperature
58	REAL	get	Power Unit Temperature	r0037[0] power unit temperature
59	REAL	get	Energy kWh	r0039 energy display
60	UINT8	get	CDS Eff (Local Mode)	r0050 active command data set
61	WORD	get	Status Word 2	r0053 status word 2
62	WORD	get	Control Word 1	r0054 control word 1

No.	Type	Service	Name	Value/explanation
63	REAL	get	Motor Speed (Encoder)	r0061 actual speed value
64	UINT32	get	Digital Inputs	r0722 digital inputs status
65	UINT32	get	Digital Outputs	r0747 digital outputs status
66	REAL	get	Analog Input 1	r0752[0] analog input 1
67	REAL	get	Analog Input 2	r0752[1] analog input 2
68	REAL	get	Analog Output 1	r0774[0] analog output 1
69	REAL	get	Analog Output 2	r0774[1] analog output 2
70	UINT16	get	Fault Code 1	r0947[0] fault number 1
71	UINT16	get	Fault Code 2	r0947[1] fault number 2
72	UINT16	get	Fault Code 3	r0947[2] fault number 3
73	UINT16	get	Fault Code 4	r0947[3] fault number 4
74	UINT16	get	Fault Code 5	r0947[4] fault number 5
75	UINT16	get	Fault Code 6	r0947[5] fault number 6
76	UINT16	get	Fault Code 7	r0947[6] fault number 7
77	UINT16	get	Fault Code 8	r0947[7] fault number 8
78	REAL	get	Pulse Frequency	r1801 pulse frequency
79	UINT16	get	Alarm Code 1	r2110[0] alarm number 1
80	UINT16	get	Alarm Code 2	r2110[1] alarm number 2
81	UINT16	get	Alarm Code 3	r2110[2] alarm number 3
82	UINT16	get	Alarm Code 4	r2110[3] alarm number 4
83	REAL	get	PID setpoint Output	r2260 technology controller setpoint after the ramp-function generator
84	REAL	get	PID Feedback	r2266 technology controller actual value after the filter
85	REAL	get	PID Output	r2294 technology controller output signal

Siemens Motor Data Object, Instance Number: 32D hex

Supported services

Class
- Get Attribute single

Instance
- Get Attribute single
- Set Attribute single

Table 4-18 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-19 Instance Attribute

No.	Service	Type	Name	Value/explanation
2	get, set	UINT16	Commissioning state	p0010
3	get	INT16	Motor Type	p0300
6	get, set	REAL	Rated Current	p0305
7	get, set	REAL	Rated Voltage	p0304
8	get, set	REAL	Rated Power	p0307
9	get, set	REAL	Rated Frequency	p0310
10	get, set	REAL	Rated Temperature	p0605
11	get, set	REAL	Max Speed	p0322
12	get, set	UINT16	Pole pair number	p0314
13	get, set	REAL	Torque Constant	p0316
14	get, set	REAL	Inertia	p0341
15	get, set	REAL	Base Speed	p0311
19	get, set	REAL	Cos Phi	p0308

TCP/IP Interface Object, Instance Number: F5 hex

Supported services

Class
- Get Attribute all
- Get Attribute single

Instance
- Get Attribute all
- Get Attribute single
- Set Attribute single

Table 4-20 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-21 Instance Attribute

No.	Service	Type	Name	Value/explanation
1	get	UINT32	Status	Fixed value: 1 hex 1: Configuration acknowledged, by DHCP or saved values
2	get	UINT32	Configuration Capability	Fixed value: 94 hex 4 hex: DHCP supported 10 hex: Configuration can be adjusted 80 hex: ACD-capable

No.	Service	Type	Name	Value/explanation
3	get, set	UINT32	Configuration Control	1 hex: Saved values 3 hex: DHCP
4	get	UINT16	Path Size (in WORDs)	Fixed value: 2 hex
		UINT8	Path	20 hex, F6 hex, 24 hex, 05 hex, where 5 hex is the number of instances of F6 hex (four physical ports plus one internal port).
5	get, set	STRING	Interface Configuration	r61000 Name of Station
		UINT32		r61001 IP address
6	get, set	UINT16	Host Name	Host Name Length
		STRING		
10	get, set	UINT8	Select ACD	local OM flash: 0: Disabled, 1: Enabled
11	get, set	UINT8	Last Conflict Detected	local OM flash ACD Activity
		UINT8		local OM flash Remote MAC
		UINT8		local OM flash ARP PDU

Link Object, Instance Number: F6 hex

Supported services

Class
- Get Attribute all
- Get Attribute single

Instance
- Get Attribute all
- Get Attribute single
- Set Attribute single

Table 4-22 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Table 4-23 Instance Attribute

No.	Service	Type	Name	Value/explanation
1	get	UINT32	Interface Speed	0: link down 10: 10 Mbps 100: 100 Mbps
2	get		Interface Flags	Bit 1: Link-Status Bit 2: Duplex Mode (0: Half duplex, 1 duplex) Bit 3 ... 5: Automatic state identification Bit 6: Reset required Bit 7: Local hardware fault (0 = ok)

No.	Service	Type	Name	Value/explanation
3	get	ARRAY	Physical Address	r8935 Ethernet MAC address
4	get_and_clear	Struct of	Interface Counters	Optional; required if the Media Counters attribute is implemented
		UINT32	In Octets	Received octets
		UINT32	In Ucast Packets	Received Unicast packets
		UINT32	In NUcast Packets	Received non-Unicast packets
		UINT32	In Discards	Incoming packets, not processed
		UINT32	In Errors	Incoming packets with errors
		UINT32	In Unknown Protos	Incoming packets with unknown protocol
		UINT32	Out Octets	Sent octets
		UINT32	Out Ucast Packets	Sent Unicast packets
		UINT32	Out NUcast packets	Sent non-Unicast packets
		UINT32	Out Discards	Outgoing packets, not processed
		UINT32	Out Errors	Outgoing packets, with errors
5	get_and_clear	Struct of	Media Counters	Media-specific counters
		UINT32	Alignment Errors	Structure received, which does not match the number of octets
		UINT32	FCS Errors	Structure received, which does not pass the FCS check
		UINT32	Single Collisions	Structure successfully transmitted, precisely one collision
		UINT32	Multiple Collisions	Structure successfully transmitted, multiple collisions
		UINT32	SQE Test Errors	Number of SQE errors
		UINT32	Deferred Transmissions	First transmission attempt delayed
		UINT32	Late Collisions	Number of collisions that occurred delayed by 512 bit timers to the request
		UINT32	Excessive Collisions	Transmission unsuccessful. Reason: Intensive collision
		UINT32	MAC Transmit Errors	Transmission unsuccessful. Reason: An internal MAC sublayer receiving error
		UINT32	Carrier Sense Errors	Times that the carrier sense condition was lost or never asserted when attempting to transmit a frame
		UINT32	Frame Too Long	Structure too large
		UINT32	MAC Receive Errors	Transmit unsuccessful. Reason: An internal MAC sublayer receiving error
6	get, set	Struct of	Interface Control	-
		UINT16	Control Bits	-
		UINT16	Forced Interface Speed	-
10	get	String	Interface_Label	Interface-Label

Parameter Object, Instance Number: 401 hex

Supported services

Class
- Get Attribute all

Instance
- Get Attribute all
- Set Attribute single

Table 4-24 Class Attribute

No.	Service	Type	Name
1	get	UINT16	Revision
2	get	UINT16	Max Instance
3	get	UINT16	Num of Instances

Cyclic communication is established via parameter object 401.

Example: Read parameter 2050[10] (connector output to interconnect the PZD received from the fieldbus controller)

Get Attribute single function with the following values:

- Class = 401 hex
- Instance = 2050 = 802 hex corresponds to the parameter number
- Attribute = 10 = A hex corresponds to index 10

Example: Parameter 1520[0] writing (upper torque limit)

Set Attribute single function with the following values:

- Class = 401 hex
- Instance = 1520 = 5F0 hex corresponds to the parameter number
- Attribute = 0 = 0 hex corresponds to index 0
- Data = 500.0 (value)

4.5.1 Supported ODVA AC/DC assemblies

Overview

Number		required/ optional	Type	Name
hex	dec			
14 hex	20	Required	Sending	Basic Speed Control Output
46 hex	70	Required	Receiving	Basic Speed Control Input

Assembly Basic Speed Control, Instance Number: 20, type: Output

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0						Fault Reset		RUN Forward
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							

Assembly Basic Speed Control, Instance Number: 70, type: Input

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0						Running Forward		Faulted
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							

4.6 Create generic I/O module

Overview

For certain controllers, or if you wish to use the SINAMICS profile, you cannot use the EDS file provided by Siemens. In these cases, you must create a generic I/O module in the control system for the cyclic communication.

Function description

Procedure

1. In your control, create a generic device with Ethernet/IP functionality.
2. In the control, enter the lengths for the process data for cyclic communication in the new device which you set in the converter:
   r2067[0] (input), r2067[1] (output), e.g.: Standard telegram 2/2
   4 ms is supported as the minimum value for RPI (Requested Packet Interval).
3. In the converter, set the same values for IP address, subnet mask, default gateway and name of the station as in the control.
   Configuring communication (Page 82)

You have created a generic I/O module for cyclic communication with the converter.
❒

Further information

You can find a detailed description of how to create a generic I/O module on the Internet:
Generating an EDS file (http://support.automation.siemens.com/WW/view/en/82843076)

4.7 The converter as an Ethernet station

Integrating a converter into an Ethernet network (assigning an IP address)

Procedure

1. Set p8924 (PN DHCP mode) = 2 or 3
   - p8924 = 2
     The DHCP server assigns the IP address based on the MAC address of the converter
   - p8924 = 3
     The DHCP server assigns the IP address based on the device name of the converter
2. Save the settings with p8925 = 2. The next time that the converter switches on, it retrieves the IP address. After this, you can address the converter as an Ethernet node.

Note

Immediate switchover without restart

The switchover to DHCP is performed immediately and without a restart if the change is carried out with the EtherNet/IP command "Set Attribute Single" (class F5 hex, attribute 3). The following options are available:

- Via an EtherNet/IP controller
- Via an EtherNet/IP commissioning tool

You have now integrated the converter into Ethernet

Displays

r8930: Device name of the converter

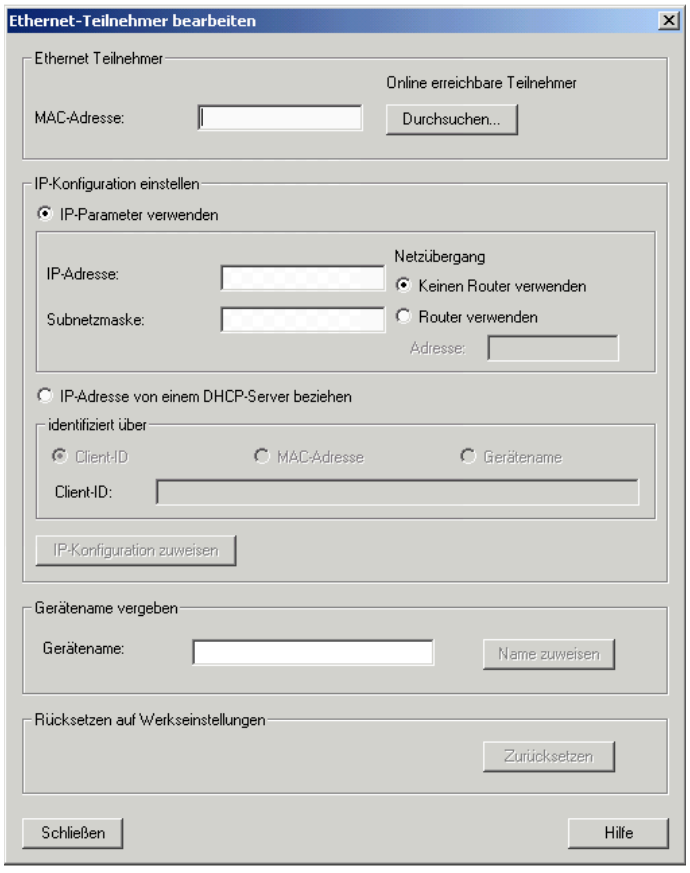
r8934: Operating mode, PN or DHCP

r8935: MAC address

Additional options of integrating converters into Ethernet

You also have the option of integrating the converter into Ethernet using Proneta or STEP 7, for example.

Here is the example of the "Edit Ethernet station" screen form from Step 7, which you can use to make the required settings.



See also

Overview of the manuals (Page 217)

You can find the required settings for the converter as Ethernet node in section "The converter with PROFINET interface as Ethernet node. (Page 69)".

Communication via RS485

5

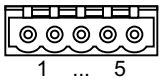
Table 5-1 Assignment table - fieldbus systems via RS485

Converter/Control Unit		Fieldbus connection for			
		USS	Modbus RTU	BACnet MS/TP	P1
	G120				
	• CU230P-2 HVAC	✓	✓	✓	✓
	• CU230P-2 BT	✓	✓	✓	✓
	• CU240B-2	✓	✓	---	---
	• CU240E-2	✓	✓	---	---
	• CU240E-2 F	✓	✓	---	---
	• CU250S-2	✓	✓	---	---
	G120C				
	• G120C USS/MB	✓	✓	---	---

5.1 Converter with RS485 interface

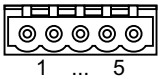
You can find the connectors and the connector assignments of the RS485 interface in the following tables.

Table 5-2 Assignment table

Converter/Control Unit	Connection via		
	X128	X03, in (M12)	X04, out (M12)
			

	G120			
	• CU230P-2 HVAC	x		
	• CU230P-2 BT	x		
	• CU240B-2	x		
	• CU240E-2	x		
	• CU240E-2 F	x		
	• CU250S-2	x		
	G120C			
	• G120C USS/MB	x		

Table 5-3 Pin assignment

Signal	X128 	X03, in (M12) 	X04, out (M12) 
Not assigned	5	1/3	1/3
RS485N, receive and transmit (-)	3	---	---
RS485N, receive	---	2	---
RS485N, transmit (-)	---	---	2
RS485P, receive and transmit (+)	2	---	---
RS485P, receive	---	4	---
RS485P, transmit (+)	---	---	4
0 V, reference potential	1	5	5
Cable shield	4	---	---

5.2 Integrating converters into a bus system via the RS485 interface

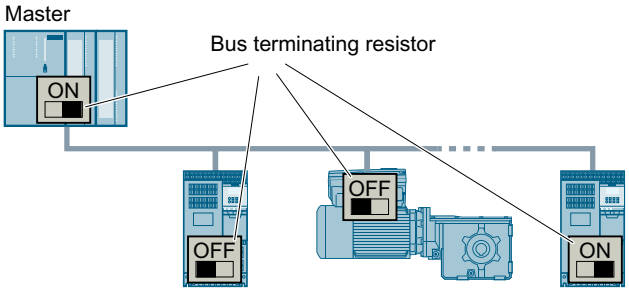
Connecting to a network via RS485

Connect the converter to the fieldbus via the RS485 interface.

The RS485 connector has short-circuit proof, isolated pins.

You must switch-in the bus-terminating resistor for the first and last nodes.

You can find the position of the RS485 connector and the bus terminating resistor in the operating instructions for the converter or the Control Unit.



The precondition for error-free communications is that the first and last station are supplied with power.

Communications are maintained if you withdraw individual devices from the bus without interrupting the cable (this is not possible for converters with a high degree of protection).

Communication with the control, even when the line voltage is switched off

So that communication with the control system in your plant or system continues to function even when the line voltage is switched off, you must externally supply the converter/Control Unit with 24 V DC. To do this, use terminals 31 and 32 or connector X01 (X01/X02 with G115D). You can find additional details in the operating instructions for the converter or the Control Unit.

5.3 Communication via USS

The USS protocol is a serial data link between a master and up to a maximum of 31 devices.

A master is, for example:

- A programmable logic controller (e.g. SIMATIC S7-200)
- A PC

The converter is always a device.

The maximum cable length is:

- 1200 m for a baud rate up to 38400 bit/s and maximum of 32 nodes
- 1000 m for a baud rate of 187500 bit/s and a maximum of 30 nodes

Additional information on how to connect the converter to a USS fieldbus: Integrating converters into a bus system via the RS485 interface (Page 103).

5.3.1 Basic settings for communication

Overview

Depending on the converter, the following options are available for setting communication via the USS:

- For all converters with an RS485 interface:
  21 "USS Fieldbus"
- For converters with a CU230P-2 HVAC / CU230P-2 BT
  108 "BT Mac 8: USS fieldbus"
  For additional information, please refer to the operating instructions of your converter.
  Overview of the manuals (Page 217).

Procedure with default setting 21 "USS Fieldbus"

Proceed as follows to set communication via USS:

1. Activate communication via the RS485 interface using one of the following options:
   - With Startdrive during commissioning in step "Default setting of setpoint/command sources":
     21: USS fieldbus
   - With the BOP-2 during basic commissioning under step "MAc PAr P15":
     FB USS
   - Via the parameter number:
     p0015 = 21
2. Set the bus protocol via p2030:
   p2030 = 1
3. Set the converter address.
4. Make additional changes based on the parameters listed in the following section.
5. If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You have now made the settings for communication via USS.

5.3.1.1 Setting the address

Valid address area: 0 ... 31

You have the following options for setting the address:

- Using the address switch on the Control Unit:

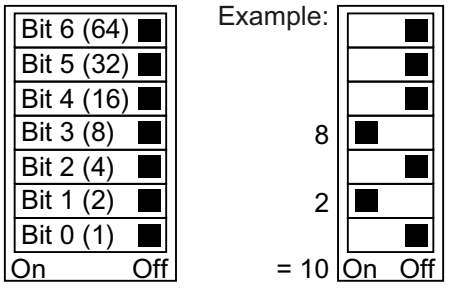


Figure 5-1 Address switch with example for bus address 10

The address switch has priority over the other settings.

- Using Startdrive or an operator panel via parameter p2021 (default setting: p2021 = 0) it is only possible to change p2021 if an invalid address is set in the address switch. If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You can find the position of the address switch in the operating instructions for the converter.

Manuals and technical support (Page 217)

Activating the changed bus address

Procedure

1. Set the address as described above.
2. Switch off the converter power supply.
3. Wait until all LEDs on the converter are dark.
4. Switch on the converter power supply again.
   Your settings become effective after switching on.

You have now set the bus address.
❒

5.3.1.2 Parameters to set communication via USS

Fieldbus protocol selection p2030 = 1 (USS)

Baud rate p2020 = 8, 38400 bit/s

Setting range: 2400 bit/s ... 187500 bit/s

Fieldbus analog outputs p0791[0 ... 1]

Parameter to interconnect the analog outputs for control via the fieldbus

Fieldbus interface USS PZD number p2022 = 2

Setting the number of 16-bit words in the PZD part of the USS telegram
Setting range: 0... 8 (0 ... 8 words)

Fieldbus interface USS PKW number, p2023 = 127

Setting the number of 16-bit words in the PKW part of the USS telegram
Setting range:

- 0, 3, 4: fixed length with 0, 3 or 4 words
- 127: variable lengths

Fieldbus error statistics r2029

Displaying receive errors at the fieldbus interface

Fieldbus monitoring time p2040 = 100 ms

Setting range: 0 ms ... 1999999 ms

The more devices that are connected in the network, the longer the fieldbus monitoring time must be.

If process data is not transferred within one cycle of the fieldbus monitoring time, then the converter shuts down with fault F01910.

p2040 = 0 ⇒ bus monitoring deactivated.

5.3.2 Telegram structure

Overview

A USS telegram comprises a series of elements with a defined sequence. Each element contains 11 bits.

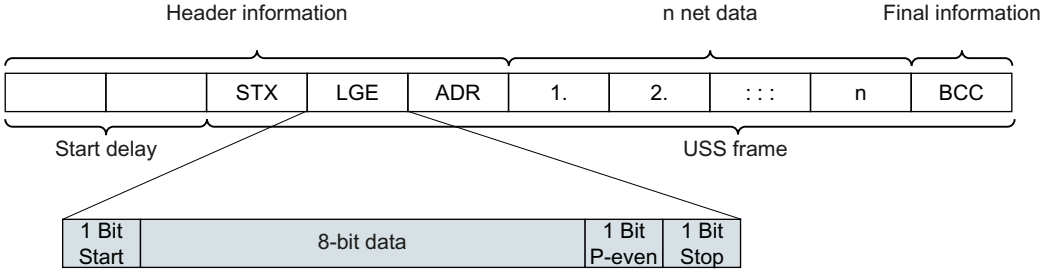


Figure 5-2 Structure of a USS telegram

Telegram part	Description
Start delay / response delay	There is always a start / response delay between two telegrams. Telegram monitoring (Page 117)
STX	An ASCII character (02 hex) indicates the beginning of the message.
LGE	The telegram length "LGE" is calculated as follows: LGE = user data (n bytes) + ADR (1 byte) + BCC (1 byte)

Telegram part	Description
ADR	7 Special telegram; 6 Mirror telegram; 5 Broadcast bit; 4 3 2 1 0 Address • Bit 7 = 0: Normal data exchange. Bit 7 =1, to transfer telegrams that require a net data structure different from the device profile. • Bit 6 = 0: Normal data exchange. Bit 6 = 1: Testing the bus connection: The converter returns the telegram unchanged to the master. • Bit 5 = 0: Normal data exchange. (Bit 5 = 1: Not supported in the converter.) • Bits 0 ... 4: Address of the converter.
User data	Specify user data of telegram (Page 107).
BCC	Checksum (exclusive or) across all telegram bytes – with the exception of BCC.

5.3.3 Specify user data of telegram

Overview

The user data of the telegram consist of the following elements:

- Parameter channel (PIV) for writing and reading parameter values
- Process data (PZD) for controlling the drive

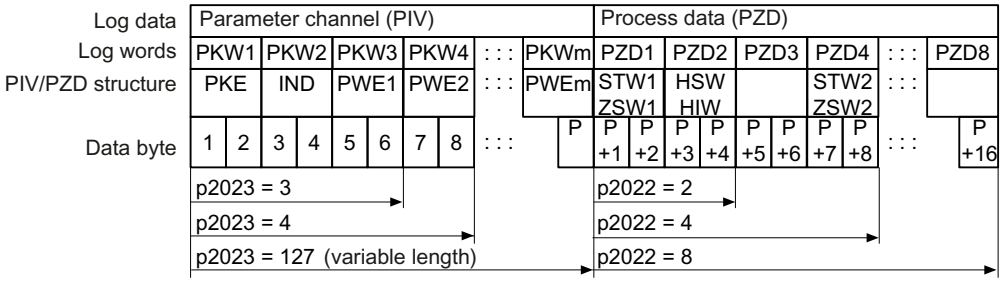


Figure 5-3 USS telegram - user data structure

Function description

Parameter channel

You specify the length of the parameter channel in parameter p2023:

- p2023 = 0
  With this setting, no parameter values are transferred.
- p2023 = 3
  You can select this setting if you only want to read or write 16-bit data or alarm signals.

- p2023 = 4:
  If you want to read or write 32-bit values (for example indexed parameters or bit parameters, e.g. r0722.2), then this setting is required. In this case, the send or receive telegram always contains four words, even if only three would be required. The values are right-justified in the 4th word.
- p2023 = 127:
  If you set p2023 = 127 (variable length), the send and response telegrams are exactly as long as the task requires.

Process data

Parameter p2022 defines the length for the process data. You can transfer up to eight process data items in one telegram (p2022 = 0 ... 8). For p2022 = 0, no process data is transferred.

Parameters

Parameter	Description	Factory setting
p2022	Fieldbus interface USS PZD number	2
p2023	Fieldbus interface USS PKW number	127

5.3.4 USS parameter channel

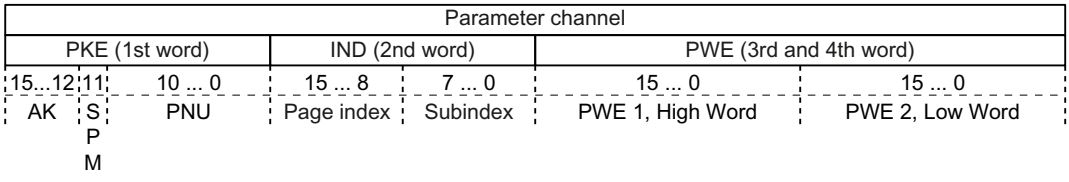
Structure of the parameter channel

Depending on the setting in p2023, the parameter channel has a fixed length of three or four words, or a variable length, depending on the length of the data to be transferred.

1. and 2nd word contain the parameter number and index as well as the type of job (read or write). The other words of the parameter channel contain parameter contents. The parameter contents can be 8-bit values, 16-bit values (such as baud rate) or 32-bit values (e.g. CO parameters). The parameter contents are entered right justified in the word with the highest number. Words that are not required are assigned 0.

Bit 11 in the 1st word is reserved and is always assigned 0.

The diagram shows a parameter channel that is four words long.



You can find examples of telegrams at the end of this section.

Function description

AK: Request and response ID

Table 5-4 Request identifiers, control → converter

AK	Description	Response identifier	
		positive	negative
0	No request	0	7 / 8
1	Request parameter value	1 / 2	7 / 8
2	Change parameter value (word)	1	7 / 8
3	Change parameter value (double word)	2	7 / 8
4	Request descriptive element [1]	3	7 / 8
6 [2]	Request parameter value (field) [1]	4 / 5	7 / 8
7 [2]	Change parameter value (field, word) [1]	4	7 / 8
8 [2]	Change parameter value (field, double word) [1]	5	7 / 8
9	Request number of field elements	6	7 / 8

[1] The required element of the parameter is specified in IND (2nd word).

[2] The following request IDs are identical: 1 ≡ 6, 2 ≡ 7 and 3 ≡ 8.
We recommend that you use identifiers 6, 7 and 8.

Table 5-5 Response identifiers, converter → control

AK	Description
0	No response
1	Transfer parameter value (word)
2	Transfer parameter value (double word)
3	Transfer descriptive element [1]
4	Transfer parameter value (field, word) [2]
5	Transfer parameter value (field, double word) [2]
6	Transfer number of field elements
7	Converter cannot process the request. In the most significant word of the parameter channel, the converter sends an error number to the control, refer to the following table.
8	No master controller status / no authorization to change parameters of the parameter channel interface

[1] The required element of the parameter is specified in IND (2nd word).

[2] The required element of the indexed parameter is specified in IND (2nd word).

Table 5-6 Error numbers for response identifier 7

No.	Description
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 subindex 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 error value that cannot be changed)
0B hex	**No master control** (change request but with no master control, see also p0927.)
0C hex	**Keyword missing**
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)
65 hex	**Parameter number is currently deactivated** (depending on the mode of the converter)
66 hex	**Channel width is insufficient** (communication channel is too small for response)
68 hex	**Illegal parameter value** (parameter can only assume certain values)
6A hex	**Request not included / task is not supported** (the valid request identifications can be found in table "Request identifications controller → converter")
6B hex	**No change access for a controller that is enabled**. (The operating state of the conerter prevents a parameter change)
86 hex	**Write access only for commissioning (p0010 = 15)** (operating state of the converter 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 converter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

PNU (parameter number) and page index

Parameter number	PNU	Page index
0000 ... 1999	0000 ... 1999	0 hex
2000 ... 3999	0000 ... 1999	80 hex
6000 ... 7999	0000 ... 1999	90 hex
8000 ... 9999	0000 ... 1999	20 hex
10000 ... 11999	0000 ... 1999	A0 hex
20000 ... 21999	0000 ... 1999	50 hex
29000 ... 29999	0000 ... 1999	70 hex

Parameter number	PNU	Page index
30000 ... 31999	0000 ... 1999	F0 hex
60000 ... 61999	0000 ... 1999	74 hex

Subindex

For indexed parameters, the parameter index is located in subindex as hexadecimal value.

PWE: Parameter value or connector

Parameter values or connectors can be located in the PWE.

Table 5-7 Parameter value or connector

	PWE 1	PWE 2		
Parameter value	Bit 15 ... 0	Bit 15 ... 8		Bit 7 ... 0
	0	0		8-bit value
	0	16-bit value		
	32-bit value			
Connector	Bit 15 ... 0	Bit 15 ... 10	Bit 9 ... 0	
	Number of the connector	3F hex	The index or bit field number of the connector	

Examples

Read request: Read out serial number of the Power Module (r7841[2])

To obtain the value of the indexed parameter r7841, you must fill the telegram of the parameter channel with the following data:

- **PKE, Bit 12 ... 15 (AK): = 6** (request parameter value (field))
- **PKE, Bit 0 ... 10 (PNU): = 1841** (parameter number without offset)
  Parameter number = PNU + offset (page index)
  (7841 = 1841 + 6000)
- **IND, bit 8 ... 15 (page index): = 2** (index of parameter)
- **IND, bit 0 ... 7 (subindex): = 90 hex** (offset 6000 corresponds to 90 hex)
- Because you want to read the parameter value, words 3 and 4 in the parameter channel for requesting the parameter value are irrelevant. They should be assigned a value of 0, for example.

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 8	7 ... 0
AK		Parameter number	Page index	Subindex	Parameter value	Parameter value	Parameter value
0 1 1 0	0	1 1 1 0 0 1 1 0 0 0 1	0 0 0 0 0 0 1 0	1 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Figure 5-4 Telegram for a read request from r7841[2]

Parameter number

Parameter numbers < 2000	PNU = parameter number. Write the parameter number into the PNU (PKE bit 10 ... 0).
Parameter numbers ≥ 2000	PNU = parameter number - offset. Write the parameter number minus the offset into the PNU (PKE bit 10 ... 0). Write the offset in the page index (IND bit 15 ... 8).

Table 5-8 Offset and page index of the parameter numbers

Parameter number	Offset	Page index								
		Hex	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0000 ... 1999	0	0 hex	0	0	0	0	0	0	0	0
2000 ... 3999	2000	80 hex	1	0	0	0	0	0	0	0
6000 ... 7999	6000	90 hex	1	0	0	1	0	0	0	0
8000 ... 9999	8000	20 hex	0	0	1	0	0	0	0	0
10000 ... 11999	10000	A0 hex	1	0	1	0	0	0	0	0
20000 ... 21999	20000	50 hex	0	1	0	1	0	0	0	0
29000 ... 29999	28000	70 hex	0	1	1	1	0	0	0	0
30000 ... 31999	30000	F0 hex	1	1	1	1	0	0	0	0
60000 ... 61999	60000	74 hex	0	1	1	1	0	1	0	0

Indexed parameters

For indexed parameters, you must write the index as hex value into the subindex (IND bit 7 ... 0).

Parameter contents

Parameter contents can be parameter values or connector parameters. You require two words for connector parameters. You can find more information on interconnecting connector parameters in the operating instructions of the converter in the section "Interconnecting signals in the converter".

Enter the parameter value in the parameter channel right-justified as follows:

- 8-bit values: Low word, bits 0 ... 7, bits 8 ... 15 are zero.
- 16-bit values: Low word, bits 0 ... 15,
- 32-bit values: Low word and high word

Enter a connector parameter right-justified as follows:

- Number of the connector parameter: High word
- Drive object of the connector parameter: Low word, bits 10 ... 15
- The index or bit field number of the connector parameter: Low word, bits 0 ... 9

5.3.4.1 Telegram examples, length of the parameter channel = 4

Read request: Read out serial number of the Power Module (r7841[2])

To obtain the value of the indexed parameter r7841, you must fill the telegram of the parameter channel with the following data:

- **PKE, bit 12 ... 15 (AK): = 6** (request parameter value (field))
- **PKE, bit 0 ... 10 (PNU): = 1841** (parameter number without offset)
  Parameter number = PNU + offset (page index)
  (7841 = 1841 + 6000)
- **IND, bit 8 ... 15 (page index): = 90 hex** (offset 6000 corresponds to 90 hex)
- **IND, bit 0 ... 7 (subindex): = 2** (index of parameter)
- Because you want to read the parameter value, words 3 and 4 in the parameter channel for requesting the parameter value are irrelevant. They should be assigned a value of 0, for example.

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 8	7 ... 0
AK		Parameter number	Page index	Subindex	Parameter value	Parameter value	Parameter value
0 1 1 0	0	1 1 1 0 0 1 1 0 0 0 1	0 0 0 0 0 0 1 0	1 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Figure 5-5 Telegram for a read request from r7841[2]

Write request: Changing the automatic restart mode (p1210)

Parameter p1210 defines the automatic restart mode:

- **PKE, bit 12 ... 15 (AK): = 7** (change parameter value (field, word))
- **PKE, bit 0 ... 10 (PNU): = 4BA hex** (1210 = 4BA hex, no offset, as 1210 < 1999)
- **IND, bit 8 ... 15 (page index): = 0 hex** (offset 0 corresponds to 0 hex)
- **IND, bit 0 ... 7 (subindex): = 0 hex** (parameter is not indexed)
- **PWE1, bit 0 ... 15: = 0 hex**
- **PWE2, bit 0 ... 15: = 1A hex** (26 = 1A hex)

Parameter channel						
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 0
AK		Parameter number	Page index	Subindex	Parameter value (bit 16 ... 31)	Parameter value (bit 0 ... 15)
0 1 1 1	0	1 0 0 1 0 1 1 1 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0

Figure 5-6 Telegram, to activate the automatic restart with p1210 = 26

Write request: Assign digital input 2 with the function ON/OFF1 (p0840[1] = 722.2)

In order to link digital input 2 with ON/OFF1, you must assign parameter p0840[1] (source, ON/OFF1) the value 722.2 (DI 2). To do this, you must fill the telegram of the parameter channel as follows:

- **PKE, bit 12 ... 15 (AK): = 7 hex** (change parameter value (field, word))
- **PKE, bit 0 ... 10 (PNU): = 348 hex** (840 = 348 hex, no offset, as 840 < 1999)
- **IND, bit 8 ... 15 (page index): = 0 hex** (offset 0 corresponds to 0 hex)

- **IND, bit 0 ... 7 (subindex): = 1 hex** (command data set CDS1 = index1)
- **PWE1, bit 0 ... 15**: = **2D2 hex** (722 = 2D2 hex)
- **PWE2, bit 10 ... 15: = 3f hex** (drive object - for SINAMICS G120 always 63 = 3f hex)
- **PWE2, bit 0 ... 9: = 2 hex** (index or bit number of the parameter: DI 2 = r0722.2)

Parameter channel							
PKE, 1st word			IND, 2nd word		PWE1 - high, 3rd word	PWE2 - low, 4th word	
15...12	11	10 ... 0	15 ... 8	7 ... 0	15 ... 0	15 ... 10	9 ... 0
AK		Parameter number	Page index	Subindex	Parameter value	Drive Object	Index
0 1 1 1	0	0 1 1 0 1 0 0 1 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0	1 1 1 1 1 1	0 0 0 0 0 0 0 0 1 0

Figure 5-7 Telegram, to assign DI 2 with ON/OFF1

5.3.5 USS process data channel (PZD)

Function description

The process data channel (PZD) contains the following data depending on the transmission direction:

- Control words and setpoints for the device.
- Status words and actual values for the master.

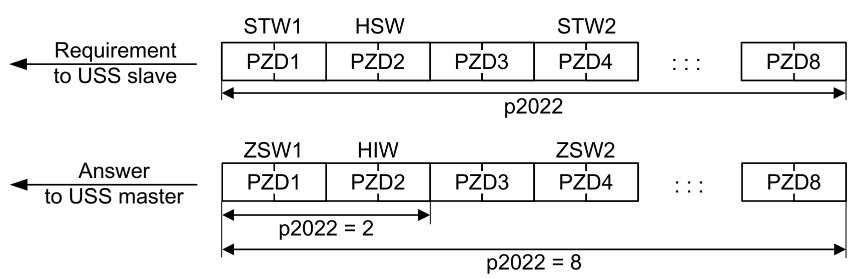


Figure 5-8 Process data channel

The first two words are:

- Control 1 (STW1) and main setpoint (HSW)
- Status word 1 (ZSW1) and main actual value (HIW)

If p2022 is greater than or equal to 4, then the converter receives the additional control word (STW2).

Control word 1 (STW1)

Bit	Significance	Explanation	Signal interconnection in the converter
0	0 = OFF1	The motor brakes with the ramp-down time p1121 of the ramp-function generator. The converter switches off the motor at standstill.	p0840[0] = r2090.0
	0 → 1 = ON	The converter goes into the "ready" state. If, in addition bit 3 = 1, then the converter switches on the motor.	
1	0 = OFF2	Switch off the motor immediately, the motor then coasts down to a standstill.	p0844[0] = r2090.1
	1 = No OFF2	The motor can be switched on (ON command).	
2	0 = Quick stop (OFF3)	Quick stop: The motor brakes to a standstill with the OFF3 ramp-down time p1135.	p0848[0] = r2090.2
	1 = No quick stop (OFF3)	The motor can be switched on (ON command).	
3	0 = Inhibit operation	Immediately switch-off motor (cancel pulses).	p0852[0] = r2090.3
	1 = Enable operation	Switch-on motor (pulses can be enabled).	
4	0 = Disable RFG	The converter immediately sets its ramp-function generator output to 0.	p1140[0] = r2090.4
	1 = Do not disable RFG	The ramp-function generator can be enabled.	
5	0 = Stop RFG	The output of the ramp-function generator stops at the actual value.	p1141[0] = r2090.5
	1 = Enable RFG	The output of the ramp-function generator follows the setpoint.	
6	0 = Inhibit setpoint	The converter brakes the motor with the ramp-down time p1121 of the ramp-function generator.	p1142[0] = r2090.6
	1 = Enable setpoint	Motor accelerates to the setpoint with the ramp-up time p1120.	
7	0 → 1 = Acknowledge faults	Acknowledge fault. If the ON command is still active, the converter switches to the "switching on inhibited" state.	p2103[0] = r2090.7
8, 9	Reserved		
10	0 = No control via PLC	Converter ignores the process data from the fieldbus.	p0854[0] = r2090.10
	1 = Control via PLC	Control via fieldbus, converter accepts the process data from the fieldbus.	
11	1 = Direction reversal	Invert setpoint in the converter.	p1113[0] = r2090.11
12	Reserved		
13	1 = MOP up	Increase the setpoint saved in the motorized potentiometer.	p1035[0] = r2090.13
14	1 = MOP down	Reduce the setpoint saved in the motorized potentiometer.	p1036[0] = r2090.14
15	Reserved		

Status word 1 (ZSW1)

Bit	Significance	Remarks	Signal interconnection in the converter
0	1 = Ready for switching on	Power supply switched on; electronics initialized; pulses locked.	p2080[0] = r0899.0
1	1 = Ready	Motor is switched on (ON/OFF1 = 1), no fault is active. With the command "Enable operation" (STW1.3), the converter switches on the motor.	p2080[1] = r0899.1
2	1 = Operation enabled	Motor follows setpoint. See control word 1, bit 3.	p2080[2] = r0899.2
3	1 = Fault active	The converter has a fault. Acknowledge fault using STW1.7.	p2080[3] = r2139.3
4	1 = OFF2 inactive	Coast down to standstill is not active.	p2080[4] = r0899.4
5	1 = OFF3 inactive	Quick stop is not active.	p2080[5] = r0899.5
6	1 = Switching on inhibited active	It is only possible to switch on the motor after an OFF1 followed by ON.	p2080[6] = r0899.6
7	1 = Alarm active	Motor remains switched on; no acknowledgement is necessary.	p2080[7] = r2139.7
8	1 = Speed deviation within the tolerance range	Setpoint / actual value deviation within the tolerance range.	p2080[8] = r2197.7
9	1 = Master control requested	The automation system is requested to accept the converter control.	p2080[9] = r0899.9
10	1 = Comparison speed reached or exceeded	Speed is greater than or equal to the corresponding maximum speed.	p2080[10] = r2199.1
11	1 = Torque limit not reached	Fallen below comparison value for current or torque.	p2080[11] = r0056.13 / r1407.7
12	Reserved		p2080[12] = r0899.12
13	0 = Alarm, motor overtemperature	--	p2080[13] = r2135.14
14	1 = Motor rotates clockwise	Internal converter actual value > 0.	p2080[14] = r2197.3
	0 = Motor rotates counter-clockwise	Internal converter actual value < 0.	
15	0 = Alarm, converter thermal overload		p2080[15] = r2135.15

5.3.6 Telegram monitoring

Function description

You require the telegram runtimes in order to set the telegram monitoring. The character runtime is the basis of the telegram runtime:

Table 5-9 Character runtime

Baud rate in bit/s	Transmission time per bit	Character run time (= 11 bits)
9600	104.170 µs	1.146 ms
19200	52.084 µs	0.573 ms
38400	26.042 µs	0.286 ms
57600	17.361 µs	0.191 ms
115200	8.681 µs	0.095 ms

The telegram runtime is longer than just purely adding all of the character runtimes (=residual runtime). You must also take into consideration the character delay time between the individual characters of the telegram.

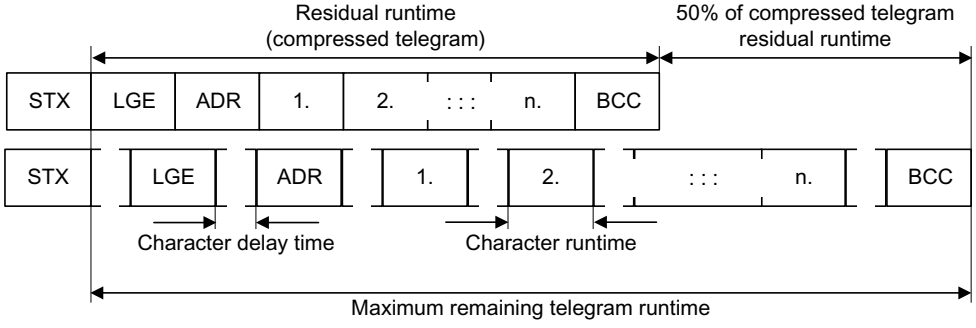


Figure 5-9 Telegram runtime as the sum of the residual runtime and character delay times

The total telegram runtime is always less than 150% of the pure residual runtime.

Before each request telegram, the master must maintain the start delay. The start delay must be > 2 × character runtime.

The device only responds after the response delay has expired.

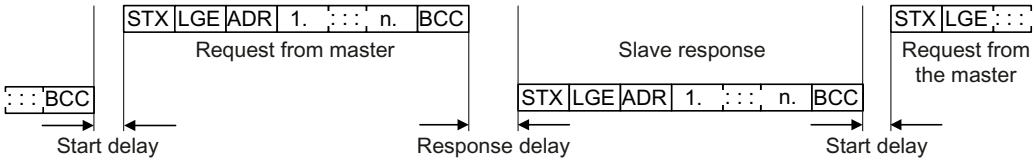


Figure 5-10 Start delay and response delay

Table 5-10 Start delay

Baud rate in bit/s	Transmission time per character (= 11 bits)	Min. start delay
9600	1.146 ms	> 2.291 ms
19200	0.573 ms	> 1.146 ms

Baud rate in bit/s	Transmission time per character (= 11 bits)	Min. start delay
38400	0.286 ms	> 0.573 ms
57600	0.191 ms	> 0.382 ms
115200	0.095 ms	> 0.191 ms

The character delay time must be shorter than the start delay.

Telegram monitoring of the master

With your USS master, we recommend that the following times are monitored:

- Response delay:
  Response time of the device to a request from the master
  The response delay must be < 20 ms, but longer than the start delay
- Telegram runtime:
  Transmission time of the response telegram sent from the device

Telegram monitoring of the converter

The converter monitors the time between two requests of the master. Parameter p2040 defines the permissible time in ms. If a time p2040 ≠ 0 is exceeded, then the converter interprets this as telegram failure and responds with fault F01910.

150% of the residual runtime is the guide value for the setting of p2040, i.e. the telegram runtime without taking into account the character delay times.

For communication via USS, the converter checks bit 10 of the received control word 1. If the bit is not set when the motor is switched on ("Operation"), the converter responds with fault F07220.

Parameters

Parameter	Description	Factory setting
p2040	Fieldbus interface monitoring time	1 000 ms

5.4 Communication using Modbus RTU

Overview of communication using Modbus

The Modbus protocol is a communication protocol based on a client/server architecture. Selected parameters and process data are exchanged in a cyclic access via the Modbus register.

Modbus offers three transmission modes:

- **Modbus ASCII -** via a serial interface
  data in the ASCII code. The data throughput is lower compared to RTU.
- **Modbus RTU -** via a serial interface
  data in the binary format. The data throughput is greater than in ASCII code.
- **Modbus TCP -** via Ethernet
  Data are transferred as TCP/IP packages TCP port 502 is reserved for Modbus TCP.

General information about communication using Modbus RTU

Communication using Modbus RTU takes place over the RS485 interface with a maximum of 247 devices.

- The maximum cable length is 1200 m.
- To polarize the receive and transmit lines, there are two 100 kΩ resistors, which you can switch in or switch out using the DIP switch next to the fieldbus interface.

Note

It is not permitted to change over the units

The "Unit switchover" function – for details see the operating instructions of the Control Unit – is not permissible with this bus system!

5.4.1 Basic settings for communication

Overview

Depending on the converter, the following options are available for setting communication via the Modbus RTU:

- For all converters with an RS485 interface:
  21 "USS Fieldbus"
- For converters with a CU230P-2 HVAC / CU230P-2 BT
  109 "BT Mac 9: Modbus RTU Fieldbus"
  For additional information, please refer to the operating instructions of your converter.
  Overview of the manuals (Page 217).

Procedure with default setting 21 "USS Fieldbus"

Proceed as follows to set communication via Modbus RTU:

1. Activate communication via the RS485 interface using one of the following options:
   - With Startdrive during commissioning step "Default setting of setpoint/command sources":
     21: USS fieldbus
   - With the BOP-2 during the basic commissioning under step "MAc PAr P15":
     FB USS
   - Via parameter number:
     p0015 = 21
2. Set the bus protocol via p2030:
   p2030 = 2
3. Set the converter address.
4. Make additional changes based on the parameters listed in the following section.
5. If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You have now made the settings for communication via Modbus.

5.4.1.1 Setting the address

Valid address area: 1 ... 247

You have the following options for setting the address:

- Using the address switch on the Control Unit from 1 ... 127:

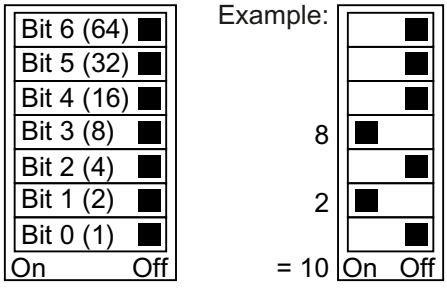


Figure 5-11 Address switch with example for bus address 10

The address switch has priority over the other settings.

- Using Startdrive or an operator panel via parameter p2021 from 1 ... 247 (default setting: p2021 = 1)
  Setting via p2021 is only possible if address 0 is set in the address switch.
  If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You can find the position of the address switch in the operating instructions for the converter.

Manuals and technical support (Page 217)

Activating the changed bus address

Procedure

1. Set the address as described above.
2. Switch off the converter power supply.
3. Wait until all LEDs on the converter are dark.
4. Switch on the converter power supply again.
   Your settings become effective after switching on.

You have now set the bus address.
❒

5.4.1.2 Parameters for setting communication via Modbus RTU

General settings

Fieldbus protocol selection p2030 = 2 (Modbus RTU)

Baud rate p2020 = 7, 19200 bit/s

Setting range: 4800 bit/s ... 187500 bit/s

Parity

In the factory, the converter is set for controllers with "parity even". You can adapt the parity at your controller using p2031:

- p2031 = 0: No parity, 1 stop bit or 2 stop bits
- p2031 = 1: Odd parity, 1 stop bit
- p2031 = 2: Even parity, 1 stop bit
- p2031 = 3: No parity, 1 stop bit

Modbus timing p2024[0 ... 2]

- **p2024[0]: Maximum device telegram processing time:**
  The time after which the device must have sent a response to the master. 0 ms ... 10000 ms, factory setting = 6000 ms.
- **p2024[1]: Character delay time:**
  Character delay time: Maximum permissible time between the individual characters in the Modbus frame. (Modbus standard processing time for 1.5 bytes).
- **p2024 [2]: Inter-telegram delay:**
  maximum permissible time between Modbus telegrams. (Modbus standard processing time for 3.5 bytes).

Values for p2024 [1] and p2024 [2]
Table 5-11 Baud rates, transmission times, and delays (Page 125).

Fieldbus monitoring time p2040 = 1000 ms

Setting range: 0 ms ... 1999999 ms

The more devices that are connected in the network, the longer the fieldbus monitoring time must be.

If process data is not transferred within one cycle of the fieldbus monitoring time, then the converter shuts down with fault F01910.

p2040 = 0 ⇒ bus monitoring deactivated.

Fieldbus error statistics r2029

Displaying receive errors at the fieldbus interface

Interconnecting analog outputs

If you set communication via Modbus (p2030 = 2), then the analog outputs of the converter are internally interconnected with the fieldbus analog outputs:

- p0771[0] = 791[0]
- p0771[1] = 791[1].

The values for p0791[0] and p0791[1] are written via registers 40523 and 40524. Interconnections between parameter p0791 and other sources are rejected.

This means that the control outputs system-specific values via the analog outputs of the converter.

However, if you still wish to display a converter-specific value, you must adapt the appropriate wiring.

Example

- AO 0 should display the value written via the control with register 40523. In this particular case, no other settings are required in the converter.
- AO 1 should display the smoothed actual current value. To do this, you must set p0771[1] = 27 (r0027 smoothed actual current value).
  In this case, a write access via register 40524 to p0791[1] results in a fault message in the control.

Note

Reset to the factory setting for Modbus

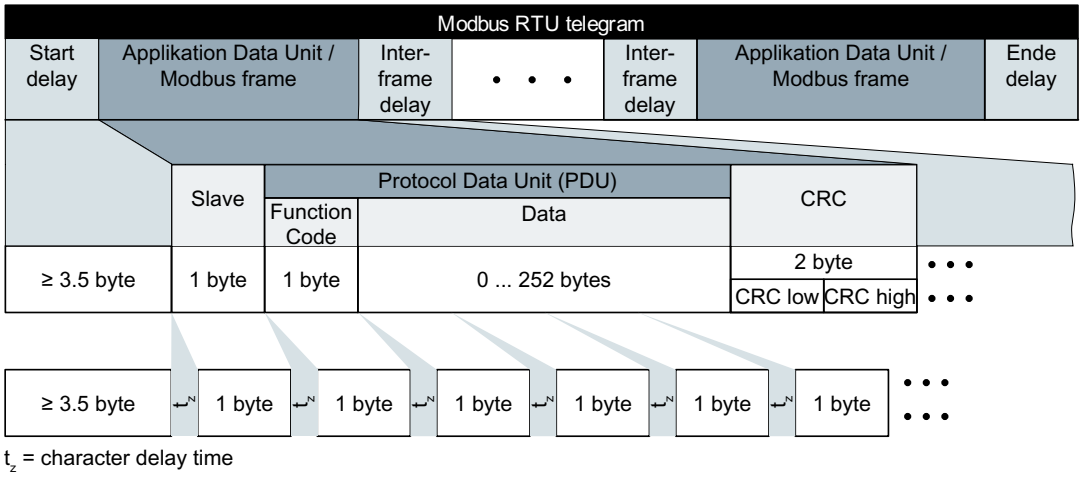
If you have set communication via Modbus (p2030 = 2), when restoring the factory settings, the analog outputs are again interconnected with p0771[0] = 791[0] and p0771[1] = 791[1].

5.4.2 Modbus RTU telegram

Description

For Modbus, there is precisely one master and up to 247 devices. The master always starts the communication. Devices send data when requested to do so by the master. Device-to-device communication is not possible. The converter always operates as device.

The following figure shows the structure of a Modbus RTU telegram.



t_z = character delay time

Figure 5-12 Modbus with delay times

The data area of the telegram is structured according to the mapping tables.

5.4.3 Baud rates and mapping tables

Permissible baud rates and telegram delay

The Modbus RTU telegram requires pauses for the following situations:

- for the start identifier
- for separating the individual frames
- for the end identifier

Minimum duration: Processing time for 3.5 bytes (can be set via p2024[2]).

A character delay time is also permitted between the individual bytes of a frame. Maximum duration: Processing time for 1.5 bytes (can be set via p2024[1]).

Table 5-11 Baud rates, transmission times, and delays

Baud rate in bit/s (p2020)	Transmission time per character (11 bits)	Minimum pause between two telegrams (p2024[2])	Maximum pause between two bytes (p2024[1])
4800	2.292 ms	≥ 8.021 ms	≤ 3.438 ms
9600	1.146 ms	≥ 4.010 ms	≤ 1.719 ms
19200 (factory setting)	0.573 ms	≥ 1.75 ms	≤ 0.859 ms
38400	0.286 ms	≥ 1.75 ms	≤ 0.75 ms
57600	0.191 ms	≥ 1.75 ms	≤ 0.556 ms
76800	0.143 ms	≥ 1.75 ms	≤ 0.417 ms
93750	0.117 ms	≥ 1.75 ms	≤ 0.341 ms
115200	0.095 ms	≥ 1.75 ms	≤ 0.278 ms
187500	0.059 ms	≥ 1.75 ms	≤ 0.171 ms

Note

The factory setting for p2024[1] and p2024[2] is 0. The converter defines the particular values depending on the protocol selection (p2030) or the baud rate.

Modbus register

The converter supports the subsequently listed registers. Error "Exception Code" is output if an attempt is made to access other registers.

Note

Read and write access to converter data

R: read via FC03; W: write via FC06; R/W: read via FC03 or write via FC06

Table 5-12 Assigning the Modbus registers to the process data

Register	Description	Access	Scaling	Data / parameter
40100	Control word	R/W	1	Process data 1
40101	Main setpoint	R/W	1	Process data 2
40110	Status word	R	1	Process data 1
40111	Main actual value	R	1	Process data 2

5.4.4 Mapping tables - converter data

Table 5-13 Assigning the Modbus registers to the parameters - inputs and outputs

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range		Data / parameter
Digital outputs							
40200	DO 0	R/W	--	1	HIGH	LOW	p0730, r747.0, p748.0
40201	DO 1	R/W	--	1	HIGH	LOW	p0731, r747.1, p748.1
40202	DO 2	R/W	--	1	HIGH	LOW	p0732, r747.2, p748.2
Analog outputs							
40220	AO 0	R	%	100	-100.0 ... 100.0		r0774.0
40221	AO 1	R	%	100	-100.0 ... 100.0		r0774.1
40222	AO 2	R	%	100	-100.0 ... 100.0		r0774.2
40523	AO 0	R/W	%	100	-199.99 ... 199.99		p0791.0
40524	AO 1	R/W	%	100	-199.99 ... 199.99		p0791.1
40525	AO 2	R/W	%	100	-199.99 ... 199.99		p0791.2
Digital inputs							
40240	DI 0	R	--	1	HIGH	LOW	r0722.0
40241	DI 1	R	--	1	HIGH	LOW	r0722.1
40242	DI 2	R	--	1	HIGH	LOW	r0722.2
40243	DI 3	R	--	1	HIGH	LOW	r0722.3
40244	DI 4	R	--	1	HIGH	LOW	r0722.4
40245	DI 5	R	--	1	HIGH	LOW	r0722.5
Analog inputs							
40260	AI 0	R	%	100	-300.0 ... 300.0		r0755 [0]
40261	AI 1	R	%	100	-300.0 ... 300.0		r0755 [1]
40262	AI 2	R	%	100	-300.0 ... 300.0		r0755 [2]
40263	AI 3	R	%	100	-300.0 ... 300.0		r0755 [3]

Table 5-14 Assigning the Modbus registers to the parameters - converter data

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range	Data / parameter
40300	Powerstack number	R	--	1	0 ... 32767	r0200
40301	Converter firmware	R	--	1	e.g. 470	r0018 / 10000
40320	Rated power	R	kW	100	0 ... 327.67	r0206
40321	Current limit	R/W	A	10	10.0 ... 400.0	p0640
40322	Ramp-up time	R/W	s	100	0.00 ... 650.0	p1120
40323	Ramp-down time	R/W	s	100	0.00 ... 650.0	p1121
40324	Reference speed	R/W	RPM	1	6 ... 32767	p2000
Converter diagnostics						
40340	Speed setpoint	R	RPM	1	-16250 ... 16250	r0020
40341	Actual speed value	R	RPM	1	-16250 ... 16250	r0022

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range		Data / parameter
40342	Output frequency	R	Hz	100	- 327.68 ... 327.67		r0024
40343	Output voltage	R	V	1	0 ... 32767		r0025
40344	DC link voltage	R	V	1	0 ... 32767		r0026
40345	Current actual value	R	A	100	0 ... 163.83		r0027
40346	Actual torque value	R	Nm	100	- 325.00 ... 325.00		r0031
40347	Actual active power	R	kW	100	0 ... 327.67		r0032
40348	Energy consumption	R	kWh	1	0 ... 32767		r0039
40349	Control priority	R	--	1	HAND	AUTO	r0807

Table 5-15 Assigning the Modbus registers to the parameters - fault diagnostics

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range	Data / parameter
40400	Failure number, index 0	R	--	1	0 ... 32767	r0947 [0]
40401	Failure number, index 1	R	--	1	0 ... 32767	r0947 [1]
40402	Failure number, index 2	R	--	1	0 ... 32767	r0947 [2]
40403	Fault number, index 3	R	--	1	0 ... 32767	r0947 [3]
40404	Fault number, index 4	R	--	1	0 ... 32767	r0947 [4]
40405	Fault number, index 5	R	--	1	0 ... 32767	r0947 [5]
40406	Fault number, index 6	R	--	1	0 ... 32767	r0947 [6]
40407	Fault number, index 7	R	--	1	0 ... 32767	r0947 [7]
40408	Alarm number	R	--	1	0 ... 32767	r2110 [0]
40409	Actual alarm code	R	--	1	0 ... 32767	r2132
40499	PRM ERROR code	R	--	1	0 ... 255	--

Table 5-16 Assigning the Modbus registers to the parameters - technology controller

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range	Data / parameter
40500	Technology controller enable	R/W	--	1	0 ... 1	p2200, r2349.0
40501	Technology controller MOP	R/W	%	100	-200.0 ... 200.0	p2240
Technology controller adjustment						
40510	Time constant for actual value filters of the technology controller	R/W	--	100	0.00 ... 60.0	p2265
40511	Scaling factor for actual value of the technology controller	R/W	%	100	0.00 ... 500.00	p2269
40512	Proportional amplification of the technology controller	R/W	--	1000	0.000 ... 65.000	p2280
40513	Integral time of the technology controller	R/W	s	1	0 ... 60	p2285
40514	Time constant D-component of the technology controller	R/W	--	1	0 ... 60	p2274

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range	Data / parameter
40515	Max. limit of technology controller	R/W	%	100	-200.0 ... 200.0	p2291
40516	Min. limit technology controller	R/W	%	100	-200.0 ... 200.0	p2292

Table 5-17 Assigning the Modbus registers to the parameters - PID diagnostics

Register	Description	Access	Unit	Scaling	ON/OFF text/ value range	Data / parameter
40520	Effective setpoint acc. to internal technology controller MOP ramp-function generator	R	%	100	-100.0 ... 100.0	r2250
40521	Actual value of technology controller after filter	R	%	100	-100.0 ... 100.0	r2266
40522	Output signal technology controller	R	%	100	-100.0 ... 100.0	r2294

Table 5-18 Modbus registers for communication via DS47

Register	Description	Access	Unit	Scaling	Data / parameter
40601	DS47 Control	R/W	--	--	--
40602	DS47 header	R/W	--	--	--
40603	DS47 data 1	R/W	--	--	--
...	...	...			
40722	DS47 data 120	R/W	--	--	--

Table 5-19 Modbus registers for multi-pump control

Register	Last register	Description	Access	Unit	Scaling	ON/OFF text/value range	Data / parameter
40800		Status word	R	--	1	0 ... 65535	p29529
40801		Motor index speed control	R	--	1	0 ... 3	p29538
40802		Status word, service mode	R	--	1	0 ... 65535	p29544
40804	40805	Motor 1 operating hours	R/W	h	10	0 ... 429496729.5	p29530[0]
40806	40807	Motor 2 operating hours	R/W	h	10	0 ... 429496729.5	p29530[1]
40808	40809	Motor 3 operating hours	R/W	h	10	0 ... 429496729.5	p29530[2]
40810	40811	Motor 4 operating hours	R/W	h	10	0 ... 429496729.5	p29530[3]

5.4.5 Acyclic communication via Modbus RTU

Acyclic communication or general parameter access is realized using the Modbus registers 40601 ... 40722.

Acyclic communication is controlled using 40601. 40602 contains the function code (always = 47 = 2F hex) and the number of the following user data. User data are contained in registers 40603 ... 40722.

Overview of acyclic communication

Value in the register				Explanation
40601	**40602**		**40603 ... 40722**	
0	47	...	...	Write values for acyclic access
1	47	Request length [bytes]	Request data	Activate acyclic access
2	47	Response length [bytes]	Response data	Response for a successful request
2	47	0	Error code	Response for an erronous request

Error codes

1 hex: Invalid Length (invalid length)
2 hex: Invalid State (in the current converter state, this action is not permitted)
3 hex: Invalid function code (FC ≠ 2F hex)
4 hex: Response not ready (the response has still not been issued)
5 hex: Internal Error (general system error)

Incorrect access operations to parameters via data set 47 are logged in registers 40603 ... 40722.

5.4.6 Write and read access using function codes

Basic structure of read and write access using function codes

Slave ID	Protocol Data Unit (PDU)		CRC	
	FC	Data	low	high
1 Byte	1 Byte	0 ... 252 Bytes	2 Byte	

Function codes used

For data exchange between the master and device, predefined function codes are used for communication via Modbus.

The converter uses the following Modbus function codes:

- FC 03: Holding register to read data from the converter
- FC 06: Write single register to write to individual register
- FC 16: Write to multiple registers to write to several registers

Structure of a read request via Modbus function code 03 (FC 03)

Any valid register address is permitted as the start address.

Via FC 03, the control can address more than one register with one request. The number of addressed registers is contained in bytes 4 and 5 of the read request.

Table 5-20 Structure of a read request via device number 17, example

Value	Byte	Description
11 h	0	Device address
03 h 00 h 6D h 00 h 02 h	1 2 3 4 5	Function code Register start address "High" (register 40110) Register start address "Low" Number of registers "High" (2 registers: 40110; 40111) number of registers "Low"
xx h xx h	6 7	CRC "Low" CRC "High"

The response returns the corresponding data set:

Table 5-21 Device response to the read request, example

Value	Byte	Description
11 h	0	Device address
03 h 04 h 11 h 22 h 33 h 44 h	1 2 3 4 5 6	Function code Number of bytes (4 bytes are returned) Data first register "High" Data first register "Low" Data second register "High" Data second register "Low"
xx h xx h	7 8	CRC "Low" CRC "High"

Table 5-22 Invalid read request

Read request	Converter response
Invalid register address	Exception code 02 (invalid data address)
Read a write-only register	Telegram in which all values are set to 0.
Read a reserved register	
Controller addresses more than 125 registers	Exception code 03 (invalid data value)
The start address and the number of registers of an address are located outside of a defined register block	Exception code 02 (invalid data address)

Structure of a write request via Modbus function code 06 (FC 06)

Start address is the holding register address.

Via FC 06, with one request, only precisely one register can be addressed. The value, which is written to the addressed register, is contained in bytes 4 and 5 of the write request.

Table 5-23 Structure of a write request for device number 17, example

Value	Byte	Description
11 h	0	Device address
06 h 00 h 63 h 55 h 66 h	1 2 3 4 5	Function code Register start address "High" (write register 40100) Register start address "Low" Register data "High" Register data "Low"
xx h xx h	6 7	CRC "Low" CRC "High"

The response returns register address (bytes 2 and 3) and the value (bytes 4 and 5), which the higher-level control had written to the register.

Table 5-24 Device response to the write request

Value	Byte	Description
11 h	0	Device address
06 h 00 h 63 h 55 h 66 h	1 2 3 4 5	Function code Register start address "High" Register start address "Low" Register data "High" Register data "Low"
xx h xx h	6 7	CRC "Low" CRC "High"

Table 5-25 Invalid write request

Write request	Converter response
Incorrect address (a holding register address does not exist)	Exception Code 02 - invalid data address
Write to a "read-only" register	Exception Code 04 - device failure
Write to a reserved register	

For Exception Code 4, via the holding register 40499, you can read out the internal drive error code, which has occurred for the last parameter access via the holding register.

5.4.7 Acyclically read and write parameter via FC 16

Via FC 16, with one request, up to 122 registers can be written to directly one after the other, while for Write Single Register (FC 06) you must individually write the header data for each register.

Header

In addition to the device address, enter the transfer type, the start address and the number of the following registers in the header.

User data

You control the access in the user data via register 40601.

In register 40602, you define the acyclic access as well as the length of the request data.

Register 40603 contains the request reference - it is defined by the user - and the access type -reading or writing.

Register 40604 contains the number of the drive object (always 1) and the number of parameters that are read or written.

Register 40605 contains the attribute that you use to control whether you read out the parameter value or the parameter attribute. In the number of elements you specify how many indices are read.

See also

PROFIDRIVE profile - Acyclic communication (Page 42)

5.4.7.1 Read parameter

Example: r0002 read acyclically

Table 5-26 Write parameter request: Reading the parameter value of r0002 from device number 17

Value	Byte	Description
11 h	0	Device address
10 h	1	Function code (write multiple)
0258 h	2,3	Register start address
0007 h	4,5	Number of registers to be read (40601 … 40607)
0E h	6	Number of data bytes (7 registers, each 2 bytes = 14 bytes)
0001 h	7,8	40601: DS47 Control = 1 (activate request)
2F0A h	9,10	40602: Function 2F h (47), request length 10 bytes (0A h)
8001 h	11,12	40603: Request reference = 80 h, request identifier = 1 h
0101 h	13,14	40604: DO-Id = 1, number of parameters = 1
1001 h	15,16	40605: Attribute, number of elements = 1
0002 h	17,18	40606: Parameter number = 2
0000 h	19,20	40607: Subindex = 0
xx h	21	CRC "Low"
xx h	22	CRC "High"

Table 5-27 Start parameter request: Reading the parameter value of r0002 from device number 17

Value	Byte	Description
11 h	0	Device address
03 h	1	Function code (read)
0258 h	2,3	Register start address
0007 h	4,5	Number of registers to be read (40601 … 40607)
0010 h	6,7	Number of registers
xx h	8	CRC "Low"
xx h	9	CRC "High"

Table 5-28 Response for successful read operation

Value	Byte	Description
11 h	0	Device address
03 h	1	Function code (read)
20 h	2	Number of following data bytes (20 h: 32 bytes corresponds to 16 registers)
0002 h	3,4	40601: DS47 Control = 2 (the request was executed)
2F08 h	5,6	40602: Function code 2F h (47), response lengths 8 bytes
8001 h	7,8	40603: Request reference mirrored = 80 h, response identifier = 1 (request parameter)
0101 h	9,10	40604: DO-ID = 1, number of parameters = 1
0301 h	11,12	40605: Format, number of elements = 1
001F h	13,14	40606: Parameter value = 1F h (31)
xx h	15	CRC "Low"
xx h	16	CRC "High"

Table 5-29 Response for unsuccessful read operation - read request still not completed

Value	Byte	Description
11 h	0	Device address
03 h	1	Function code (read)
20 h	2	Number of following data bytes (20 h: 32 bytes corresponds to 16 registers)
0001 h	3,4	40601: Check value 1 = request is processed
2F00 h	5,6	40602: Function 2F h(47), response length 0 (fault)
0004 h	7,8	40603: Error code: 0004 Response Not Ready (response has still not been issued)
xx h	9	CRC "Low"
xx h	10	CRC "High"

5.4.7.2 Write parameter

Example: Set p1121 = 12.15

Table 5-30 Write parameter request: Writing the parameter value of p1121 from device number 17

Value	Byte	Description
11 h	0	Device address
10 h	1	Function code (write multiple)
0258 h	2,3	Register start address
000A h	4,5	Number of registers to be written to (40601 … 40610)
14 h	6	Number of data bytes (10 registers, each 2 bytes = 20 bytes)
0001 h	7,8	40601: C1 (activate request)
2F10 h	9,10	40602: Function 2F h (47), request length 16 bytes (10 h)
8002 h	11,12	40603: Request reference = 80 h, request identifier = 2 h (write)
0101 h	13,14	40604: DO-Id = 1, number of parameters = 1
1001 h	15,16	40605: Attribute, number of elements = 1
0461 h	17,18	40606: Parameter number = 1121
0000 h	19,20	40607: Subindex = 0
0801 h	21,22	40608: Format + number of values
4142 h	23,24	40609: Parameter value 12,15
6666 h	25,26	40610: Parameter value
xx h	27	CRC "Low"
xx h	28	CRC "High"

Table 5-31 Start parameter request: Writing the parameter value of p1121 from device number 17

Value	Byte	Description
11 h	0	Device address
06 h	1	Function code (write)
0258 h	2,3	Register start address
0007 h	4,5	Number of registers to be written to (40601 … 40610)
0010 h	6,7	Number of registers
xx h	8	CRC "Low"
xx h	9	CRC "High"

Table 5-32 Response for successful write operation

Value	Byte	Description
11 h	0	Device address
06 h	1	Function code (write)
20 h	2	Number of following data bytes (20 h: 32 bytes corresponds to 16 registers)
0002 h	3,4	40601: DS47 Control = 2 (request was executed)
2F04 h	5,6	40602: Function code 2F h (47), response length 4 bytes
8002 h	7,8	40603: Request reference mirrored = 80 h, response identifier = 2 (change parameter)
0101 h	9,10	40604: DO-ID = 1, number of parameters = 1
xx h	11	CRC "Low"
xx h	12	CRC "High"

Table 5-33 Response for unsuccessful write operation - write request still not completed

Value	Byte	Description
11 h	0	Device address
06 h 20 h 0001 h 2F00 h 0004 h	1 2 3,4 5,6 7,8	Function code (write) Number of following data bytes (20 h: 32 bytes corresponds to 16 registers) 40601: DS47 Control = 1 (request is processed) 40602: Function 2F h(47), response length 0 (fault) 40603: Error code: 0004 Response Not Ready (response has still not been issued)
xx h xx h	9 10	CRC "Low" CRC "High"

5.4.8 Communication procedure

Procedure for communication in a normal case

Normally, the master sends a telegram to a device (address range 1 ... 247). The device sends a response telegram to the master. This response telegram mirrors the function code; the device enters its own address in the telegram and so the device identifies itself with the master.

The device only processes orders and telegrams which are directly addressed to it.

Communication error

If the device detects a communication error on receipt (parity, CRC), it does not send a response to the master, since this can lead to "setpoint timeout".

Logical error

If the device detects a logical error within a request, it responds to the master with an "exception response". In the response, the device sets the highest bit in the function code to 1. If the device receives, for example, an unsupported function code from the master, the device responds with an "exception response" with code 01 (Illegal function code).

Table 5-34 Overview of exception codes

Exception code	Modbus name	Remark
01	Illegal function code	An unknown (unsupported) function code was sent to the device.
02	Illegal Data Address	An invalid address was requested.
03	Illegal data value	An invalid data value was detected.
04	Server failure	device has terminated during processing.

Maximum processing time, p2024[0]

The device-response time is the time in which the Modbus master expects a response to a request. Set the same device-response time (p2024 [0] in the converter) in the master and device.

Process data monitoring time (setpoint timeout), p2040

"Setpoint timeout" (F1910) is issued by the Modbus if p2040 is set to a value > 0 ms and no process data is requested within this time period.

The "Setpoint timeout" only applies for access to process data (40100, 40101, 40110, 40111). The "Setpoint timeout" is not generated for parameter data (40200 ... 40522).

Note

Adjust the time (factory setting = 100 ms) depending on the number of devices and the baud rate set on the bus.

5.4.9 Application example

An application example for MODBUS RTU is provided on the Internet:

Communication via the MODBUS interface (https://support.industry.siemens.com/cs/ww/en/view/35928944)

5.5 Communication via BACnet MS/TP - only CU230P-2 HVAC / BT

BACnet properties

In BACnet, components and systems are considered to be black boxes which contain a number of objects. BACnet objects only stipulate the behavior outside the device, BACnet sets no internal functions.

A range of object types and their instances represent one component.

Each BACnet device has precisely one BACnet device object. An NSAP (Network Service Access Point - comprising network number and MAC address; MAC: **M**edium **A**ccess **C**ontrol) uniquely identifies a BACnet device. This address is BACnet-specific and must not be confused with the Ethernet MAC address.

Data exchange with the client

The converter receives control commands and setpoints via service instructions from the control and transmits its status back to the control. The converter can also autonomously send telegrams or execute services, e.g. COV_Notification.

Communication settings

- The Control Unit supports BACnet via RS485 (BACnet MS/TP),
- Communication supports Unicode, coded with the character set UTF-8
- The maximum cable length is 1200 m (3281 ft).

Protocol Implementation Conformance Statement

The Protocol Implementation Conformance Statement (PICS) is available on the Internet:

PICS (https://support.industry.siemens.com/cs/us/en/view/109760469)

Note

It is not permitted to change over the units

The "Unit switchover" function – for details see the operating instructions of the Control Unit – is not permissible with this bus system!

5.5.1 Basic settings for communication

Setting communication via BACnet

Procedure

1. Select the default setting 110
   - With Startdrive during commissioning step "Default setting of setpoint/command sources":
     110 "BT Mac 10: BACnet MS/TP fieldbus"
   - With the BOP-2 during the basic commissioning under step "MAc PAr P15":
     P_F bAc
   - Via parameter number:
     p0015 = 110
2. Set the converter address.
3. Make additional changes based on the parameters listed in the following sections.
4. If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You have now made the settings for communication via BACnet.
❒

Settings by "BT Mac 10: BACnet MS/TP fieldbus"

Fieldbus protocol selection p2030 = 5

Baud rate p2020 = 8, 38400 bit/s

Setting range: 9600 bit/s ... 76800 bit/s

Fieldbus monitoring time p2040 = 1000 ms

Setting range: 0 ms ... 1999999 ms

The more devices that are connected in the network, the longer the fieldbus monitoring time must be.

If process data is not transferred within one cycle of the fieldbus monitoring time, then the converter shuts down with fault F01910.

p2040 = 0 ⇒ bus monitoring deactivated.

5.5.1.1 Setting the address

Valid address area: 0 ... 127
With address 0, the converter responds to a broadcast.

You have the following options for setting the BACnet address:

- Using the address switch on the Control Unit:

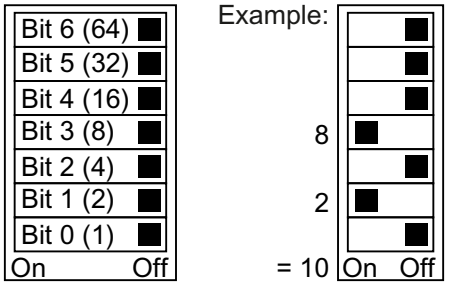


Figure 5-13 Address switch with example for bus address 10

The address switch has priority over the other settings.

- Using Startdrive or an operator panel via parameter p2021 (default setting: p2021 = 0).
  it is only possible to change p2021 if the address switch is set to 0.
  If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You can find the position of the address switch in the operating instructions for the converter.

Manuals and technical support (Page 217)

Activating the changed bus address

Procedure

1. Set the address as described above.
2. Switch off the converter power supply.
3. Wait until all LEDs on the converter are dark.
4. Switch on the converter power supply again.
   Your settings become effective after switching on.

You have now set the bus address.
❒

5.5.1.2 Parameters for setting communication via BACnet

General settings

Processing times p2024[0 ... 2]

p2024[0]: 0 ms ... 10000 ms, maximum processing time (APDU timeout), factory setting = 6000 ms,
p2024 [1 ... 2]: Irrelevant

BACnet communication parameter p2025[0 ... 3]

- p2025 [0]: 0 ... 4194303: Device object instance number,
  Factory setting = 1
- p2025 [1]: 1 ... 10: Maximum Info Frames, factory setting = 5
- p2025 [2]: 0 ... 39: Number of APDU Retries (repeated attempts after fault telegrams), factory setting = 3
- p2025 [3]: 1 ... 127: maximum manager address, factory setting = 32

Setting COV_Increment p2026[0 ... 75]

(COV = change of values) 0 ... 4194303.000, factory setting = 1. A maximum of 32 COVs are permissible.

COV_Increment: Changes the value of the "present value" of an object instance for which the server transfers an UnConfirmedCOV_Notification or ConfirmedCOV_Notification.

You can use these parameters to set the converter value changes for which an UnConfirmedCOV_Notification or ConfirmedCOV_Notification result is sent.

The factory setting 1 means that the converter sends an UnConfirmedCOV_Notification or ConfirmedCOV_Notification if the considered value, e.g. for a range of 0 ... 10 V, changes by an absolute value ≥ 1.
This requires an active SubscribeCOV_Service to send the relevant object instance.

You can also set the COV_Increment via the object property "COV_Increment" of the relevant analog input, analog output or analog value.

BACnet language selection p2027

German/English - only becomes effective after power off/on

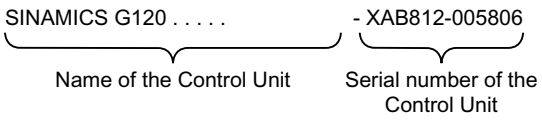
Fieldbus error statistics r2029

Displaying receive errors at the fieldbus interface

Device name - default setting, change, restore factory setting

The converter has a device name in BACnet that uniquely identifies the converter.

The device name is preset at initial power up. It has the following structure:



p7610[0...79] contains the device names in ASCII format.

Changing device names

You can change the device name either in the converter or via the controller:

- Converter: Change p7610
- Controller: Change the "object-name" property via the Write Property Service

Restoring factory settings

The device name is retained when the factory settings are restored.

If you wish to reset the name to the factory setting, original value, set p7610[0] = NULL (ASCII-0).

Interconnecting analog outputs, restoring factory settings

If you set communication via BACnet, the converter switches its analog outputs with the fieldbus.

The control then specifies the values which the converter outputs via its analog outputs.

To display a converter-specific value, you must change the interconnection of the analog output.

Examples:

- AO 0 should display the value which the control specifies in the ANALOG OUTPUT 0 object. In this particular case, no other settings are required in the converter.
- AO 1 should display the smoothed current actual value of the converter (r0027 smoothed current actual value).
  Interconnect p0771[1] with r0027: p0771[1] = 27
  In this case, write access via the object ANALOG OUTPUT 1 results in an error message in the control.

Reset to the factory setting for BACnet

When restoring the factory setting, the converter again uses the fieldbus to switch its analog outputs.

5.5.2 Supported services and objects

BIBBs used by the converter

The BIBBs (BIBB: **B**ACnet **I**nteroperability **B**uilding **B**lock) are a collection of one or several BACnet services. BACnet services are subdivided into A and B devices. An A device operates as client and a B device as server.

The converter is a server and therefore operates as B device, as "BACnet Application Specific Controller" (B-ASC).

It uses the following executed BIBBs.

Overview of the BIBB used and the associated services

Short designation	BIBB	Service
DS-RP-B	Data Sharing-ReadProperty-B	ReadProperty
DS-RPM-B	Data Sharing-ReadMultipleProperty-B	ReadPropertyMultiple
DS-WP-B	Data Sharing-WriteProperty-B	WriteProperty
DM-DDB-B	Device Management-Dynamic Device Binding-B	• Who-Is • I-Am
DM-DOB-B	Device Management-Dynamic Object Binding-B	• Who-Has • I-Have
DM-DCC-B	Device Management-DeviceCommunicationControl-B	DeviceCommunicationControl
DS-COV-B	Data Sharing-COV-B	• SubscribeCOV, • ConfirmedCOVNotification, • UnConfirmedCOVNotification

The converter can simultaneously process up to 32 SubscribeCOV services. These can all refer to the same object instances - or different object instances.

SubscribeCOV monitors the property changes of the following objects:

- Analog Input AI...
- Analog Output AO...
- Analog Value AV...
- Binary Value BV...
- Multi-State Input MSI...

Note

SubscribeCOV services are not retentive; i.e. the manager must re-initiate the SubscribeCOV services when restarting the converter.

Object types in BACnet

Object type	Code digit		Object type	Code digit
Device Object	8		Analog Output AO...	1
Binary Input BI...	3		Analog Value AV...	2
Binary Output BO...	4		Multi-State Input MSI...	13
Binary Value BV...	5		Octet String Values	47
Analog Input AI...	0			

Object properties of the "Device" object type

• Object_Identifier	• Application_Software_Version	• APDU_Timeout
• Object_Name	• Protocol_Version	• Number_Of_APDU_Retries
• Object_Type	• Protocol_Revision	• Max manager
• System_Status	• Protocol_Services_Supported	• Max Info Frames
• Vendor_Name	• Protocol_Object_Types_Supported	• Device Address Binding
• Vendor_Identifier	• Object_List	• Database Revision
• Model_Name	• Max_APDU_Length_Accepted [1]	
• Firmware_Revision	• Segmentation_Supported [2]	

[1] Length = 480, [2] not supported

Properties of the other object types

Property	Object type							
	Binary Input BI...	Binary Output BO...	Binary Value BV...	Analog Input AI...	Analog Output AO...	Analog Value AV...	Multi-State Input MSI...	Octet String values
Object_Identifier	X	X	X	X	X	X	X	X
Object_Name	X	X	X	X	X	X	X	X
Object_Type	X	X	X	X	X	X	X	X
Present_Value	X	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X	
Status_Flags	X	X	X	X	X	X	X	X
Event_State	X	X	X	X	X	X	X	
Out_Of_Service	X	X	X	X	X	X	X	
Units				X	X	X		
Priority_Array		X	X [1]		X	X [1]		
Relinquish_Default		X	X [1]		X	X [1]		
Polarity	X	X						
Active_Text	X	X	X					
Inactive_Text	X	X	X					
COV_Increment				X	X	X		
State_Text							X	
Number_of_States							X	

[1] Only for access type C: Commandable

Note

Language switching

Using parameter p2027, you can switch the language of the BACnet object properties (German, English). Only the English identifiers (e.g. "Object name") are specified in the following tables.

Binary Input Objects

Instance ID	Object name	Description	Possible values	Text active / text inactive	Access type	Parameter
BI0	DI0 ACT	State of DI 0	ON/OFF	ON/OFF	R	r0722.0
BI1	DI1 ACT	State of DI 1	ON/OFF	ON/OFF	R	r0722.1
BI2	DI2 ACT	State of DI 2	ON/OFF	ON/OFF	R	r0722.2
BI3	DI3 ACT	State of DI 3	ON/OFF	ON/OFF	R	r0722.3
BI4	DI4 ACT	State of DI 4	ON/OFF	ON/OFF	R	r0722.4
BI5	DI5 ACT	State of DI 5	ON/OFF	ON/OFF	R	r0722.5
BI7	DI7 ACT	State of AI 0 - used as DI 11	ON/OFF	ON/OFF	R	r0722.11
BI8	DI8 ACT	State of AI 1 - used as DI 12	ON/OFF	ON/OFF	R	r0722.12
BI10	DO0 ACT	State of DO 0 (relay 1)	ON/OFF	ON/OFF	R	read r0747.0
BI11	DO1 ACT	State of DO 1 (relay 2)	ON/OFF	ON/OFF	R	read r0747.1
BI12	DO2 ACT	State of DO2 (relay 3)	ON/OFF	ON/OFF	R	read r0747.2

Binary Output Objects

Instance ID	Object name	Description	Possible values	Text active / text inactive	Access type	Parameter
BO0	DO0 CMD	Controls DO 0 (relay 1)	ON/OFF	ON/OFF	C	p0730
BO1	DO1 CMD	Controls DO 1 (relay 2)	ON/OFF	ON/OFF	C	p0731
BO2	DO2 CMD	Controls DO 2 (relay 3)	ON/OFF	ON/OFF	C	p0732

Analog Input Objects

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
AI0	ANALOG IN 0	AI0 input signal	V/mA	Converter-dependent	R	r0752[0]
AI1	ANALOG IN 1	AI1 input signal	V/mA	Converter-dependent	R	r0752[1]
AI10	AIN 0 SCALED	Scaled AI 0 input signal	%	Converter-dependent	R	r0755[0]
AI11	AIN 1 SCALED	Scaled AI 1 input signal	%	Converter-dependent	R	r0755[1]

Analog Output Objects

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
AO0	ANALOG OUT 0	Value of AO0	%	Converter-dependent	C	p0791.0
AO1	ANALOG OUT 1	Value of AO1	%	Converter-dependent	C	p0791.1

Binary Value BV...

Instance ID	Object name	Description	Possible values	Text active	Text inactive	Access type[1)]	Parameter
BV0	RUN STOPPED	Converter status regardless of command source	RUN / STOP	STOP	RUN	R	r0052.2
BV1	FWD REV	Direction of rotation regardless of command source	REV / FWD	FWD	REV	R	r0052.14
BV2	FAULT ACT	Converter fault	FAULT / OK	FAULT	OK	R	r0052.3
BV3	WARN ACT	Converter warning	WARN / OK	WARN	OK	R	r0052.7
BV4	MANUAL AUTO	Source of Manual/Auto converter control	AUTO / MANUAL	AUTO	LOCAL	R	r0052.9
BV6[2)]	MAINT REQ	Maintenance required	MAINT/OK	MAINT	OK	R	reserved
BV7	HAND CONTROL	Control of the converter from the BACnet override control via BV93 The "Manual" mode of the operator panel has a higher priority than the BACnet override control.	ON/OFF	0	1	R	r2032[10]
BV8	AT SETPOINT	Setpoint reached	YES / NO	YES	NO	R	r0052.8
BV9	AT MAX FREQ	Maximum speed reached	YES / NO	YES	NO	R	r0052.10
BV10	DRIVE READY	Converter ready	YES / NO	YES	NO	R	r0052.1
BV15	HAND RUNNING	Status of the ON command, regardless of the source	YES / NO	0	1	R	r2032[0]
BV16	HIB MOD ACT	Energy saving mode is active	ON/OFF	0	1	R	r2399[1]
BV17	ESM MOD	Essential service mode is active	ON/OFF	0	1	R	r3889[0]
BV20	RUN STOP CMD	ON command for the converter (when controlling via BACnet)	RUN / STOP	0	1	C	r0054.0
BV21	FWD REV CMD	Reverse direction of rotation (when controlling via BACnet)	REV / FWD	0	1	C	r0054.11
BV22	FAULT RESET	Acknowledge fault (when controlling via BACnet)	RESET / NO	0	1	C	r0054.7
BV24	CDS	Changeover drive control	Local / Remote	YES	NO	C	r0054.15
BV26	RUN ENA CMD	Enable converter operation		ENABLED	DISABLED	C	r0054.3
BV27	OFF2	Status OFF2	RUN / STOP	0	1	C	r0054.1

Instance ID	Object name	Description	Possible values	Text active	Text inactive	Access type[1]	Parameter
BV28	OFF3	Status OFF3 BV28 sets the r0054.4, r0054.5, and r0054.6 bits	RUN / STOP	0	1	C	r0054.2
BV50	ENABLE PID	Enable technology controller	ENABLED / DISABLED	ENABLED	DISABLED	C	p2200
BV51	ENABLE PID 0	Enable technology controller 0	ENABLED / DISABLED	ENABLED	DISABLED	C	p11000
BV52	ENABLE PID 1	Enable technology controller 1	ENABLED / DISABLED	ENABLED	DISABLED	C	p11100
BV53	ENABLE PID 2	Enable technology controller 2	ENABLED / DISABLED	ENABLED	DISABLED	C	p11200
BV90	LOCAL LOCK	Use MANUAL (operator panel) to lock converter control		LOCK	UNLOCK	C	p0806
BV91[2]	LOCK PANEL	Interlocking for operator panel and parameter changes	LOCK/UNLO	0	1	W	reserved
BV93	CTL OVERRIDE	Converter control using BACnet override control	ON/OFF	0	1	C	r0054.10

[1] C: Commandable, R: Readable, W: Writable

[2] reserved for future functional expansions

Analog Value AV...

Instance ID	Object name	Description	Unit	Range	Access type[1]	Parameter
AV0	OUT FREQ HZ	Output frequency (Hz)	Hz	Converter-dependent	R	r0024
AV1	OUT FREQ PCT	Output frequency (%)	%	Converter-dependent	R	HIW
AV2	OUTPUT SPEED	Motor speed	RPM	Converter-dependent	R	r0022
AV3	DC BUS VOLT	DC-link voltage.	V	Converter-dependent	R	r0026
AV4	OUTPUT VOLT	Output voltage	V	Converter-dependent	R	r0025
AV5	CURRENT	Motor current	A	Converter-dependent	R	r0027
AV6	TORQUE	Motor torque	Nm	Converter-dependent	R	r0031
AV7	POWER	Motor power	kW	Converter-dependent	R	r0032
AV8	DRIVE TEMP	Heat sink temperature	°C	Converter-dependent	R	r0037
AV9	MOTOR TEMP	Measured or calculated motor temperature	°C	Converter-dependent	R	r0035

Instance ID	Object name	Description	Unit	Range	Access type[1)]	Parameter
AV10	KWH NR	Cumulative converter energy consumption (cannot be reset!)	kWh	Converter-dependent	R	r0039
AV12	INV RUN TIME	Motor's operating hours (is reset by entering "0")	h	0 ... 4294967295	W	p0650
AV13	INV MODEL	Code number of Power Module	---	Converter-dependent	R	r0200
AV14	INV FW VER	Firmware version	---	Converter-dependent	R	r0018
AV15	INV POWER	Rated power of the converter	kW	Converter-dependent	R	r0206
AV16	RPM STPT 1	Reference speed of the converter	RPM	6.0 ... 210000	W	p2000
AV17	FREQ SP PCT	Setpoint 1 (when controlling via BACnet)	%	-199.99 ... 199.99	C	HSW
AV18	ACT FAULT	Number of the fault due to be dealt with	---	Converter-dependent	R	r0947[0]
AV19	PREV FAULT 1	Number of the last fault	---	Converter-dependent	R	r0947[1]
AV20	PREV FAULT 2	Number of the fault before last	---	Converter-dependent	R	r0947[2]
AV21	PREV FAULT 3	Number of the fault third from last	---	Converter-dependent	R	r0947[3]
AV22	PREV FAULT 4	Number of the fault fourth from last	---	Converter-dependent	R	r0947[4]
AV25	SEL STPT	Command to select the setpoint source	---	0 ... 32767	W	p1000
AV28	AO1 ACT	Signal from AO 1	mA	Converter-dependent	R	r0774.0
AV29	AO2 ACT	Signal from AO 1	mA	Converter-dependent	R	r0774.1
AV30	MIN Speed	Minimum speed	RPM	0.000 – 19500.000	W	p1080
AV31	MAX Speed	Maximum speed	RPM	0.000 ... 210000.000	W	p1082
AV32	ACCEL TIME	Ramp-up time	s	0.00 ... 999999.0	W	p1120
AV33	DECEL TIME	Ramp-down time	s	0.00 ... 999999.0	W	p1121
AV34	CUR LIM	Current limit	A	Converter-dependent	R	p0640
AV39	ACT WARN	Indication of a pending alarm	---	Converter-dependent	R	r2110[0]
AV40	PREV WARN 1	Indication of the last alarm	---	Converter-dependent	R	r2110[1]
AV41	PREV WARN 2	Indication of the last but one alarm	---	Converter-dependent	R	r2110[2]
AV5000	RAMP UP TIME	Technology controller ramp-up time	s	0 ... 650	W	p2257
AV5001	RAMP DOWN TIME	Technology controller ramp-down time	s	0 ... 650	W	p2258

Instance ID	Object name	Description	Unit	Range	Access type[1)]	Parameter
AV5002	FILTER TIME	Technology controller actual value filter time constant	s	0 ... 60	W	p2265
AV5003	DIFF TIME	Technology controller differentiation time constant	s	0 ... 60	W	p2274
AV5004	PROP GAIN	Technology controller proportional gain	s	0 ... 1000	W	p2280
AV5005	INTEG TIME	Technology controller integral time	s	0 ... 1000	W	p2285
AV5006	OUTPUT MAX	Technology controller maximum limiting	%	- 200 ... 200	W	p2291
AV5007	OUTPUT MIN	Technology controller minimum limiting	%	- 200 ... 200	W	p2292
AV5100	RAMP UP TIME 0	Technology controller 0 ramp-up time	s	0 ... 650	W	p11057
AV5101	RAMP DOWN TIME 0	Technology controller 0 ramp-down time	s	0 ... 650	W	p11058
AV5102	FILTER TIME 0	Technology controller 0 actual value filter time constant	s	0 ... 60	W	p11065
AV5103	DIFF TIME 0	Technology controller 0 differentiation time constant	s	0 ... 60	W	p11074
AV5104	PROP GAIN 0	Technology controller 0 proportional gain	s	0 ... 1000	W	p11080
AV5105	INTEG TIME 0	Technology controller 0 integral time	s	0 ... 1000	W	p11085
AV5106	OUTPUT MAX 0	Technology controller 0 maximum limiting	%	- 200 ... 200	W	p11091
AV5107	OUTPUT MIN 0	Technology controller 0 minimum limiting	%	- 200 ... 200	W	p11092
AV5200	RAMP UP TIME 1	Technology controller 1 ramp-up time	s	0 ... 650	W	p11157
AV5201	RAMP DOWN TIME 1	Technology controller 1 ramp-down time	s	0 ... 650	W	p11158
AV5202	FILTER TIME 1	Technology controller 1 actual value filter time constant	s	0 ... 60	W	p11165
AV5203	DIFF TIME 1	Technology controller 1 differentiation time constant	s	0 ... 60	W	p11174
AV5204	PROP GAIN 1	Technology controller 1 proportional gain	s	0 ... 1000	W	p11180
AV5205	INTEG TIME 1	Technology controller integral time	s	0 ... 1000	W	p11185
AV5206	OUTPUT MAX 1	Technology controller 1 maximum limiting	%	- 200 ... 200	W	p11191
AV5207	OUTPUT MIN 1	Technology controller 1 minimum limiting	%	- 200 ... 200	W	p11192
AV5300	RAMP UP TIME 2	Technology controller 2 ramp-up time	s	0 ... 650	W	p11257

Instance ID	Object name	Description	Unit	Range	Access type[1)]	Parameter
AV5301	RAMP DOWN TIME 2	Technology controller 2 ramp-down time	s	0 ... 650	W	p11258
AV5302	FILTER TIME 2	Technology controller 2 actual value filter time constant	s	0 ... 60	W	p11265
AV5303	DIFF TIME 2	Technology controller 2 differentiation time constants	s	0 ... 60	W	p11274
AV5304	PROP GAIN 2	Technology controller 2 proportional gain	s	0 ... 1000	W	p11280
AV5305	INTEG TIME 2	Technology controller 2 integral time	s	0 ... 1000	W	p11285
AV5306	OUTPUT MAX 2	Technology controller 2 maximum limiting	%	- 200 ... 200	W	p11291
AV5307	OUTPUT MIN 2	Technology controller 2 minimum limiting	%	- 200 ... 200	W	p11292

1) C: Commandable, R: Readable, W: Writable

Multi-State Input MSI...

Instance ID	Object name	Description	Possible values	Access type	Parameter
MSI0	FAULT 1	Fault number 1	See "List of fault codes and alarm codes"	R	r0947[0]
MSI1	FAULT 2	Fault number 2		R	r0947[1]
MSI2	FAULT 3	Fault number 3		R	r0947[2]
MSI3	FAULT 4	Fault number 4		R	r0947[3]
MSI4	FAULT 5	Fault number 5		R	r0947[4]
MSI5	FAULT 6	Fault number 6		R	r0947[5]
MSI6	FAULT 7	Fault number 7		R	r0947[6]
MSI7	FAULT 8	Fault number 8		R	r0947[7]
MSI8	WARNING 1	Alarm number 1		R	r2110[0]
MSI9	WARNING 2	Alarm number 2		R	r2110[1]
MSI10	WARNING 3	Alarm number 3		R	r2110[2]
MSI11	WARNING 4	Alarm number 4		R	r2110[3]
MSI12	WARNING 5	Alarm number 5		R	r2110[4]
MSI13	WARNING 6	Alarm number 6		R	r2110[5]
MSI14	WARNING 7	Alarm number 7		R	r2110[6]
MSI15	WARNING 8	Alarm number 8		R	r2110[7]

1) R: Readable

5.5.3 Acyclic communication (general parameter access) via BACnet

Acyclic communication or general parameter access is realized via BACnet objects DS47IN and DS47OUT.

Acyclic communication uses the octet string value objects OSV0 and OSV1.

Instance ID	Object name	Description	Access type
OSV0	DS47IN	Maximum length 242, of which two bytes header, 240 bytes user data	W
OSV1	DS47OUT		R

The OSV are structured as follows:

Function Code	Request length	User data
2F (1 Byte)	(1 byte)	Maximum 240 bytes

Write parameter request with OSV0 and read with OSV1

To read parameter r0002 write the following values into the present value window of OSV0

Table 5-35 Write parameter request via OSV0

	Byte	Description
2F h	1	Function code 2F h (47),
0A h	2	Request length 10 bytes (0A h)
80 h	3	Request reference = 80 h
01 h	4	Request identifier = 1 h
01 h	5	DO-Id = 1
01 h	6	Number of parameters = 1
10 h	7	Attribute
01 h	8	Number of elements = 1
0002 h	9,10	Parameter number = 2
0000 h	11,12	Subindex = 0

If the request was successfully processed, then you can read out the response precisely once from the present value window of the OSV1:

Table 5-36 Read parameter content via OSV1

	Byte	Description
2F h	1	Function code 2F h (47)
08 h	2	Response length 8 bytes
80 h	3	Request reference = 80 h
01 h	4	Request identifier = 1 h
01 h	5	DO-Id = 1
01 h	6	Number of parameters = 1
10 h	7	Format
01 h	8	Number of elements = 1
001F h	9,10	Parameter value 1F h = 31

If the response is still not available, then you receive the following message via the present value window of the OSV1:

Table 5-37 Read parameter content via OSV1

	Byte	Description
2F h	1	Function code 2F h (47)
00 h	2	Response length 0 (error)
0004 h	3,4	Error code 4 h (response still not available)

If you wish to read the response once more, then you obtain the following message via the present value window of the OSV1:

Table 5-38 Read parameter content again via OSV1

	Byte	Description
2F h	1	Function code 2F h (47)
00 h	2	response length 0 (error)
0002 h	3,4	Error code 2 h (Invalid State)

Overview of the error codes

1 h: Invalid Length (invalid length)

2 h: Invalid State (action is not permitted in the actual converter state)

3 h: Invalid function Code (FC = 2 hex)

4 h: Response not ready (the response has still not been issued)

5 h: Internal Error (general system error)

Incorrect access operations to parameters via data set 47 are logged in objects OSV0 and OSV1.

5.6 Communication via P1 - only CU230P-2 HVAC, CU230P-2 BT

P1 is an asynchronous master-device communication between what is known as a Field Cabinet (master) and the FLN devices (device). FLN stands for "Floor level network".

The master individually addresses the various devices. A device responds only if the master addresses it. Communication between the devices is not possible.

A Field Cabinet can have several FLN ports. You can connect up to 32 FLN devices to each FLN port (devices).

Settings in the controller

In the Field Cabinet, for each device you must install what is known as a "Logical controller (LCTR) point". In addition, in the Field Cabinet you must define the "Point numbers" for communication.

An overview of the "Point Numbers" is provided on the following pages.

5.6.1 Basic settings for communication via P1

Overview

Procedure

Proceed as follows to set communication via P1:

1. Select the default setting 114
   - With Startdrive during commissioning step "Default setting of setpoint/command sources":
     114 "BT Mac 14: Communication P1"
   - With the BOP-2 during basic commissioning under step "MAc PAr P15":
     P_F _P1
   - Via parameter number:
     p0015 = 114

   After selecting default setting 114, the converter automatically sets the following parameters:
   - p2030 = 8: Fieldbus protocol P1
   - p2020 = 5: Baud rate 4800 bit/s
   - p0840 = 2090.0 The ON/OFF1 command is interconnected with control word 1, bit 0
   - p0852 = 2090.3: The signal for "Enable operation" is interconnected with control word 1, bit 3
   - p2103[0] = 2090.7: The signal for "Acknowledge fault" is interconnected with control word 1, bit 7
2. Set the address.
   Irrespective of the address that has been set, every FLN device responds to telegrams with address 99.
3. Make additional changes based on the parameters listed in the following sections.
4. If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You have now set the communication via P1.
❒

Additional parameters for adapting communication via P1

p2020 = 7: Baud rate 19200 bit/s

p1070 = 2050[1]: Receive main setpoint via fieldbus

p2051[0] = 52: Send status word via fieldbus

p2051[1] = 63: Send speed actual value via fieldbus

5.6.2 Setting the address

Valid address area: 1 ... 99

You have the following options for setting the address:

- Using the address switch on the Control Unit:

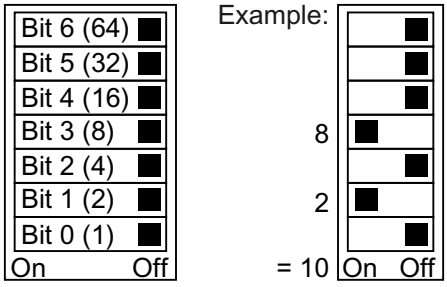


Figure 5-14 Address switch with example for bus address 10

The address switch has priority over the other settings.

- Using Startdrive or an operator panel via parameter p2021 (default setting: p2021 = 99).
  It is only possible to change p2021 if an invalid address is set in the address switch.
  If you are working with Startdrive, back up the settings so they are not lost if the power fails.

You can find the position of the address switch in the operating instructions for the converter.

Manuals and technical support (Page 217)

Activating the changed bus address

Procedure

1. Set the address as described above.
2. Switch off the converter power supply.
3. Wait until all LEDs on the converter are dark.
4. Switch on the converter power supply again.
   Your settings become effective after switching on.

You have now set the bus address.
❒

5.6.3 Point numbers

The subsequently listed "Point Numbers" for communication are defined using P1 in the converter. The values listed in the tables refer to SI units.

Point No.	Descriptor	Default/ factory	Units	Slope	Inter-cept	Subpt. Type	IO Type	On Text / Range	Off Text	CU Param / Word Type
1	CTLR ADDRESS	99	--	1	0	2	LAO_255	0 ... 255		p2021
2	APPLICATION	2767	--	1	0	2	LAO_32k	0 ... 32767		p8998[0]
3	FREQ OUTPUT	0	HZ	0.04	-650	1*)	LAI_32k	-650 ... 650		r0024
5	SPEED	0	RPM	1	-16250	1*)	LAI_32k	-16250 ... 16250		r0022
6	CURRENT	0	A	0.05	0	1*)	LAI_32k	0 ... 1638.4		r0027
7	TORQUE	0	NM	0.2	-3250	1*)	LAI_32k	-3250 ... 3250		r0031
8	ACTUAL PWR	0	KW	0.01	0	1	LAI_32k	0 ... 327.67		r0032
9	TOTAL KWH	0	KWH	1	0	1	LAI_32k	0 ... 32767		r0039
13	DC BUS VOLTS	0	V	1	0		LAI_32k	0 ... 32767		r0026
14	REFERENCE	0	HZ	0.04	-650		LAI_32k	-650 ... 650		r0020
16	RATED PWR	0	KW	0.01	0		LAI_32k	0 ... 327.67		r0206
17	OUTPUT VOLTS	0	V	1	0		LAI_32k	0 ... 32767		r0025
20	OVRD TIME	1	HRS	1	0	2	LAO_255	0 ... 255		p8998[1]
21	AR MAX FREQ	0	--	1	0	1	LDI	MAX	NO	ZSW:10
22	CMD FWD REV	0	--	1	0	1	LDO	REV	FWD	STW:11
23	FWD REV	0	--	1	0	1	LDI	FWD	REV	ZSW:14
24	CMD START	0	--	1	0	1	LDO	START	STOP	STW:0
25	STOP RUN	0	--	1	0	1	LDI	RUN	STOP	ZSW:2
26	CONTROL MODE	1	--	1	0	1	LDI	SERIAL	LOCAL	ZSW:9
28	READY TO RUN	0	--	1	0	1	LDI	READY	OFF	ZSW:1
29	DAY NIGHT	0	--	1	0	1	LDO	NIGHT	DAY	p8998[2]
30	CURRENT LMT	0.0	PCT	0.1	10.0	2	LAO_4k	0 ... 400		p0640
31	ACCEL TIME 1	10.00	SEC	0.02	0	2	LAO_32k	0 ... 650.00		p1120
32	DECEL TIME 1	10.00	SEC	0.02	0	2	LAO_32k	0 ... 650.00		p1121
34	HAND AUTO	0	--	1	0	2	LDI	HAND	AUTO	r0807.0

Point No.	Descriptor	Default/ factory	Units	Slope	Inter-cept	Subpt. Type	IO Type	On Text	Off Text	CU Param / Word Type
								Range		
35	RUN ENABLE	1	--	1	0	1	LDO	ENABLE	OFF	STW:3
36	ENABLED	0	--	1	0	1	LDI	ON	OFF	ZSW:0
40	DIGITAL OUT 1	0	--		0	2	LDO	ON	OFF	p0730 / r747.0
41	DIGITAL OUT 2	0	--	1	0	2	LDO	ON	OFF	p0731 / r747.1
42	DIGITAL OUT 3	0	--	1	0	2	LDO	ON	OFF	p0732 / r747.2
45	ANALOG IN 1	0	PCT	0.1	-300.0	1*)	LAI_32k	-300 ... 300		r0755[0]
46	ANALOG IN 2	0	PCT	0.1	-300.0	1*)	LAI_32k	-300 ... 300		r0755[1]
47	ANALOG OUT 1	0	PCT	0.1	-100.0	1	LAI_32k	-100 ... 100		r0774[0]
48	ANALOG OUT 2	0	PCT	0.1	-100.0	1	LAI_32k	-100 ... 100		r0774[1]
51	FREQ REF	0	PCT	0.006103515	0	1*)	LAO_32k	0 ... 100		HSW
52	FREQ ACTUAL	0	PCT	0.012207031	-100.0	1*)	LAI_32k	-100.0 ... 100.0		HIW
53	FREQ MAX	3000.00	HZ	0.02	1.00	1	LAO_32k	0.10 ... 650.00		p2000 1/min à Hz
55	PID SP REF	0	PCT	0.024414063	-200.0	1	LAO_32k	-200.0 ... 200.0		p2240
56	PID SP OUT	0	PCT	0.012207031	-100.0	1	LAI_32k	-100.0 ... 100.0		r2250
57	PID UP LMT	100.0	PCT	0.024414063	-200.0	1	LAO_32k	-200.0 ... 200.0		p2291
58	PID LO LMT	0	PCT	0.024414063	-200.0	1	LAO_32k	-200.0 ... 200.0		p2292
59	PID OUTPUT	0	PCT	0.012207031	0	1	LAI_32k	-100.0 ... 100.0		r2294
60	PI FEEDBACK	0	PCT	0.012207031	-100.0	1*)	LAI_32k	-100.0 ... 100.0		r2266
61	P GAIN	1.000	--	0.01	0	2	LAO_32k	0 ... 100.00		p2280
62	I GAIN	0	SEC	0.002	0	2	LAO_32k	0 ... 60.00		p2285
63	D GAIN	0	--	0.002	0	2	LAO_32k	0 ... 60.00		p2274
64	ENABLE PID	0	--	1	0	2	LDO	ON	OFF	p2200
66	FEEDBK GAIN	100.0	PCT	0.02	0	2	LAO_32k	0 ... 500.00		p2269
68	LOW PASS	0	--	0.01	0	2	LAO_32k	0 ... 60.00		p2265
71	DIGITAL IN 0	0	--	1	0	1	LDI	ON	OFF	r0722.0

Point No.	Descriptor	Default/ factory	Units	Slope	Inter-cept	Subpt. Type	IO Type	On Text	Off Text	CU Param / Word Type
								Range		
72	DIGITAL IN 1	0	--	1	0	1	LDI	ON	OFF	r722.1
73	DIGITAL IN 2	0	--	1	0	1	LDI	ON	OFF	r722.2
74	DIGITAL IN 3	0	--	1	0	1	LDI	ON	OFF	r722.3
75	DIGITAL IN 4	0	--	1	0	1	LDI	ON	OFF	r722.4
76	DIGITAL IN 5	0	--	1	0	1	LDI	ON	OFF	r722.5
80	WDOG TIME	100	ms	10	0	2	LAO_8k	0 ... 65530		p2040
83	INVERTER VER	Apr 50	--	0.01	0	2	LAI_32k	00.00 ... 99.99		r0018
84	DRIVE MODEL	0	--	1	0	2	LAI_32k	0 ... 32767		r0200
90	ACTIVE FAULT	0	--	1	0	1*)	LAI_32k	0 ...32767		r0947[0]
91	1st FAULT	0	--	1	0	1*)	LAI_32k	0 ...32767		r0947[1]
92	2nd FAULT	0	--	1	0	1*)	LAI_32k	0 ...32767		r0947[2]
93	3rd FAULT	0	--	1	0	1*)	LAI_32k	0 ...32767		r0947[3]
94	FAULT	0	--	1	0	1	LDI	FAULT	OK	ZSW:3
95	FAULT ACK	0	--	1	0	1	LDO	ON	OFF	STW:7
96	WARNING	0	--	1	0	1	LDI	WARN	OK	ZSW:7
97	ACTIVE WARNING	0	--	1	0	1*)	LAI_32k	0 ...32767		r2110[0]
98	RAM TO ROM	0	--	1	0	1	LDO	SAVE	DONE	p971/ p10=30
99	ERROR STATUS	0	--	1	0	1*)	LAI_255	0 ... 255		r947[0]

1*): For reasons of compatibility, these type 1 subpoints can save COV area information. Point Number 98 RAM TO ROM was implemented in order to be able to save these in a non-volatile fashion.

Communication over CANopen

6

General information on CAN

You can find general information about CAN on the Internet:

CAN Internet pages (http://www.can-cia.org)

You can find the CANdictionary with explanations of CAN terminology on the Internet:

CAN downloads (http://www.can-cia.org/index.php?id=6).

Integrating a converter in a CANopen network

You can find the EDS file for integrating the converter in a CANopen network on the Internet:

EDS (http://support.automation.siemens.com/WW/view/en/48351511)

The EDS file is the description file of the SINAMICS G120 converter for CANopen networks. In this way, you can use the objects of the DSP 402 device profile.

The following Control Units and converters have an CANopen interface

G120 CU230P-2 CAN
G120 CU250S-2 CAN

G120C CAN

Table 6-1 Pin assignment of the connector

Signal	D sub connector X126 1 ... 5 6 ... 9
---	1
CAN_L, CAN signal (dominant low)	2
CAN_GND, CAN ground	3
---	4
(CAN _SHLD), optional shield	5
(CAN_GND), optional ground	6

CAN_H, CAN signal (dominant high)	7
---	8
---	9

Grounding the CANopen Control Unit

The CAN ground (pin 3) and the optional ground are electrically isolated from the ground potential of the system.

The optional shield (pin 5) and the connector housing are connected with the ground potential of the system.

CANopen functions of the converter

CANopen is a communication protocol with line-type topology that operates on the basis of communication objects (COB).

SINAMICS G120 converters with CANopen interface comply with the following standards:

- CiA 301 (Application Layer and Communication Profile)
- CiA 303-3 (Indicator Specification)
- CiA 306 (Electronic Data Sheet Specification for CANopen)
- CiA 402 (Device Profile for Drives and Motion Control)

Communication objects (COB)

The converter operates with the following communication objects:

- **NMT** Network management (NMT service) (Page 161)
  Network management objects for controlling CANopen communication and for monitoring the individual nodes on the basis of a manager-device relationship.
- **SDO** SDO services (Page 164)
  Service data objects for reading and changing parameters
- **PDO** PDO services (Page 167)
  Process data objects to transfer process data, TPDO to transmit, RPDO to receive process data
- **SYNC**
  Synchronization objects
- **EMCY**
  Time stamp and fault messages

COB ID

A communication object contains the data to be transferred and a unique 11-bit COB ID. The COB ID also defines the priority for processing the communication objects. The communication object with the lowest COB ID always has the highest priority.

COB ID for individual communication objects

You will find the specifications for the COB IDs of the individual communication objects below:

- **COB ID$_{NMT}$ = 0** Cannot be changed
- **COB ID$_{SYNC}$ = free** Pre-assigned with 80 hex
- **COB ID$_{EMCY}$ = free** 80 hex + node ID = COB ID$_{EMCY}$
- **COB ID$_{TPDO}$ = free** In the free PDO mapping *)
- **COB ID$_{RPDO}$ = free** In the free PDO mapping *)
- **COB ID$_{TSDO}$** = 580 hex + node ID
- **COB-ID$_{RSDO}$** = 600 hex + node ID
- **COB ID$_{Node\ Guarding/Heartbeat}$** = 700 hex + node ID

*) Predefined connection set (Page 172)

6.1 Network management (NMT service)

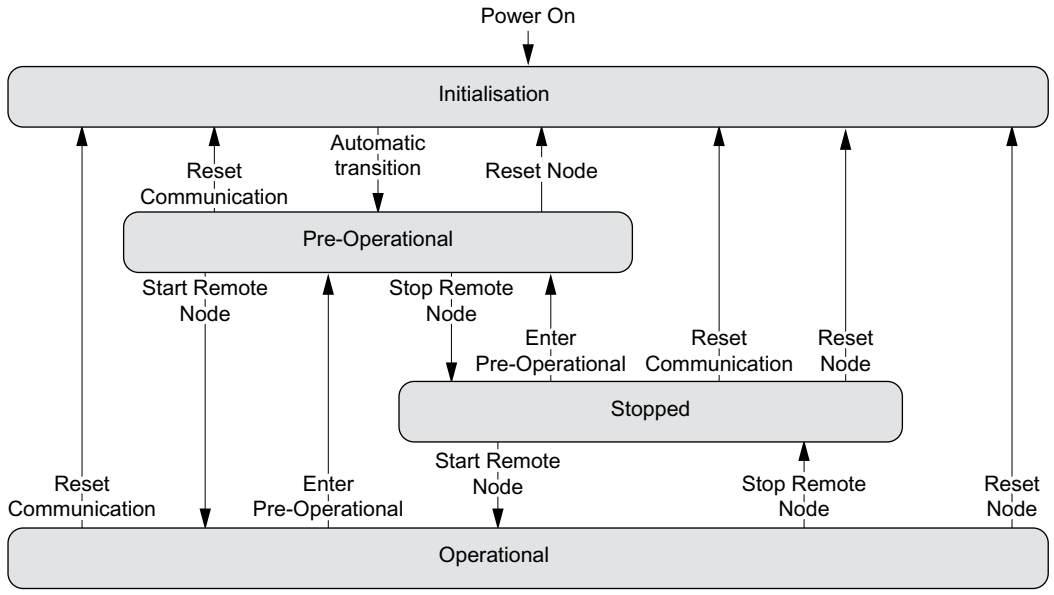
Network management (NMT) is node-oriented and has a manager-device topology.

A node is a manager or a device.

The converter is an NMT device, and can adopt the following states:

- Boot-up service COB-ID = 700 hex + Node-ID
- Node Control Service COB ID = 0 (see CANopen state diagram)
  The transition between two states is realized using NMT services. You can find details on the NMT services on the Internet:
  CiA 301 (Application Layer and Communication Profile) (http://www.can-cia.org/index.php?id=specifications).
- Error Control Service COB-ID = 700 hex + Node-ID

CANopen state diagram



NMT states

The converter state is displayed in p8685.

You can either change the converter state via the control with an NMT telegram, using one of the command specifiers listed below, or in the converter itself using p8685.

- Initialization: p8685 = 0, Command specifier = 0
  The converter initializes itself after power on. In the factory setting, the converter then enters the "Pre-Operational" state, which also corresponds to the CANopen standard.
  Using p8684, you can set that after the bus has booted, the converter does not go into the "Pre-Operational" state, but instead into the "Stopped" or "Operational" state.
- Pre-Operational, p8685 = 127 (factory setting), Command specifier = 128
  In this state, the node cannot process any process data (PDO). However, the controller can use SDO parameters to change or operate the converter, which means that you can also enter setpoints via SDO.
- Operational, p8685 = 5; Command specifier = 1
  In this state, the nodes can process SDO as well as also PDO.
- Stopped, p8685 = 4; Command specifier = 2
  In this state, the nodes can neither process PDO nor SDO. The "Stopped" state terminates one of the following commands:
  - Enter Pre-Operational, p8685 = 127 (factory setting), Command specifier = 128
  - Start Remote Node
  - Reset Node, p8685 = 128, Command specifier = 129
  - Reset Communication, p8685 = 129, Command specifier = 130

Note

Sending an incorrect NMT state

If the control sends an incorrect NMT state to the converter, the converter goes into the "Stopped" state.

Request sent by the manager to one or several devices

The NMT manager can simultaneously direct a request to one or more devices. The following is applicable:

- Requirement of a device:
  The controller accesses the device with its node ID (1 - 127).
- Requirement for all devices:
  Node ID = 0

Boot-up Service

The boot-up protocol indicates the state of the NMT device after it has booted (factory setting "Pre-operational").

Bootup protocol COB ID = 700 hex + node ID
1 data byte with the value 0 is transmitted.

NMT state after power up

Use parameter p8684 to set the state that the converter goes into after powering up:

- p8684 = 4 Stopped
- p8684 = 5 Operational
- p8684 = 127 Pre-Operational (factory setting)

Node Control Service

The Node Control Services control state transitions

- Start Remote Node:
  Command for switching from the "Pre-Operational" communication state to "Operational". The drive can only transmit and receive process data (PDO) in "Operational" state.
- Stop Remote Node:
  Command for switching from "Pre-Operational" or "Operational" to "Stopped". The node only processes NMT commands in the "Stopped" state.
- Enter Pre-Operational:
  Command for switching from "Operational" or "Stopped" to "Pre-Operational". In this state, the node cannot process any process data (PDO). However, the controller can use SDO parameters to change or operate the converter, which means that you can also enter setpoints via SDO.

- Reset Node:
  Command for switching from "Operational", "Pre-Operational" or "Stopped" to "Initialization". When the Reset Node command is issued, the converter resets all the objects (1000 hex - 9FFF hex) to the state that was present after "Power On".
- Reset Communication:
  Command for switching from "Operational", "Pre-Operational" or "Stopped" to "Initialization". When the Reset Communication command is issued, the converter resets all the communication objects (1000 hex - 1FFF hex) to the state that was present after "Power On".

Command specifier and node_ID indicate the transition states and addressed nodes.

Error Control Service

"Node Guarding / Life Guarding" or "Heartbeat" monitor communication.

Setting options and default settings.

Setting the monitoring of the communication (Page 197)

6.2 SDO services

You can access the object directory of the connected drive unit using the SDO services. An SDO connection is a peer-to-peer coupling between an SDO client and a server.

The drive unit with its object directory is an SDO server.

The identifiers for the SDO channel of a drive unit are defined according to CANopen as follows.

Receiving:	Server ⇐ client:	COB ID = 600 hex + node ID
Transmitting:	Server ⇒ client:	COB ID = 580 hex + node ID

Properties

The SDOs have the following properties:

- An SDO connection exists only in the Pre-Operational and Operational states
- Transmission is confirmed
- Asynchronous transmission (matches the acyclical communication via PROFIBUS DB)
- Transmission of values > 4 bytes (normal transfer)
- Transmission of values ≤ 4 bytes (expedited transfer)
- All drive unit parameters can be addressed via SDO

6.2.1 Access to SINAMICS parameters via SDO

You access SINAMICS parameters using the SDO service. To do this, you use objects 2000 hex ... 470F hex of the manufacturer-specific area of the object directory.

Because you cannot directly address all of the parameters using this area, you require for an SDO job always the parameter number itself and the offset dependent on the parameter number.

Selection of parameter range and the associated offset

Parameter range	Offset	Offset value
0 < parameter number < 10000	p8630[2] = 0	0
10000 ≤ parameter number < 20000	p8630[2] = 1	10000
20000 ≤ parameter number < 20000	p8630[2] = 2	20000
30000 ≤ parameter number < 20000	p8630[2] = 3	30000

Calculate object number for an SDO job

The object number for the SDO job is calculated as follows:
object number hex = (number of the converter parameter - offset value) hex + 2000 hex

Examples of object numbers

Parameter	Number of the converter parameter - offset value		Object number
	Decimal	Hexadecimal	
● p0010:	10 dec	A hex	⇒ 200A hex
● p11000:	1000 dec	3E8 hex	⇒ 23E8 hex
● r20001:	1 dec	1 hex	⇒ 2001 hex
● p31020:	1020 dec	3FC hex	⇒ 23FC hex

Selection, index range

A CANopen object can contain a maximum of 255 indexes. For parameters with more than 255 indexes, you must create additional CANopen objects via p8630[1]. Overall, 1024 indexes are possible.

- p8630[1] = 0: 0 ... 255
- p8630[1] = 1: 256 ... 511
- p8630[1] = 2: 512 ... 767
- p8630[1] = 3: 768 ... 1023

Switch-on access to objects of the converter parameters

Access to objects of the converter parameters is activated via p8630[0], where:

- p8630[0] = 0: only access to CANopen objects
- p8630[0] = 1: access to virtual CANopen objects (converter parameters)
- p8630[0] = 2: not relevant for G120 converters

A selection of important manufacturer-specific objects is included in the EDS file.

6.2.2 Access PZD objects via SDO

Access to mapped PZD objects

When you access objects mapped via transmit or receive telegrams, you can access the process data without additional settings.

Overview

Figure 6-1 Access to mapped PZD setpoint objects

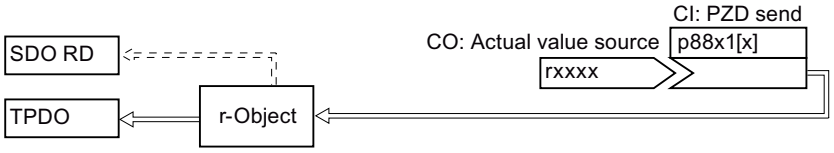


Figure 6-2 Access to mapped PZD actual value objects

Example, access to object 6042 hex

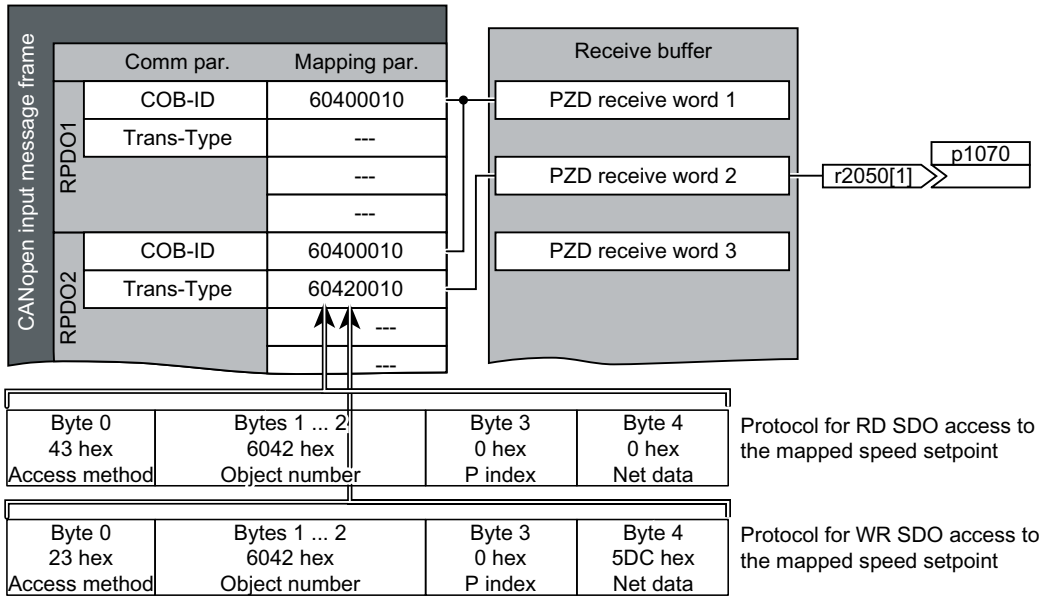


Figure 6-3 Access to the process data

Access to non-mapped PZD objects

When you access objects that are not interconnected via the receive or transmit telegram, you must also establish the interconnection with the corresponding CANopen parameters.

Overview

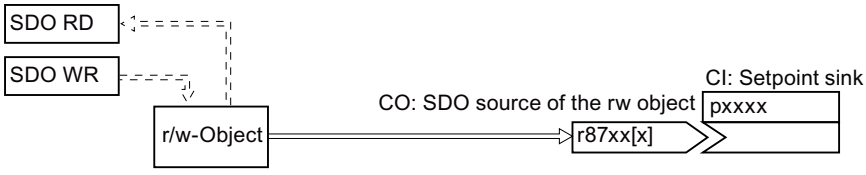


Figure 6-4 Access to non-mapped PZD setpoint objects

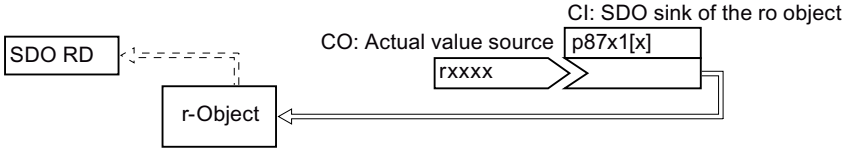


Figure 6-5 Access to non-mapped free PZD actual value objects

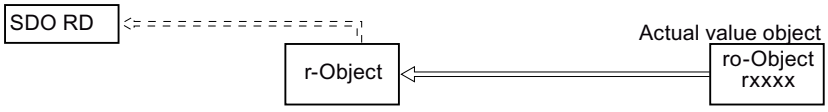


Figure 6-6 Access to non-mapped standardized PZD actual value objects

Example for interconnecting the control word with CANopen parameters:

ON/OFF1	p840[0] = r8795.0
No coast down activated	p0844[0] = r8795.1
No fast stop activated	p0848[0] = r8795.2
Enable operation	p0852[0] = r8795.3
Enable ramp-function generator	p1140[0] = r8795.4
Continue ramp-function generator	p1141[0] = r8795.5
Enable speed setpoint	p1142[0] = r8795.6
Acknowledge fault	p2103[0] = r8795.7
Stop	p8791 = r8795.8

6.3 PDO services

Process data objects (PDO)

CANopen transfers the process data using "Process Data Objects" (PDO). There are send PDOs (TDPO) and receive PDOs (RPDO). CAN controller and converter each exchange up to eight TPDOs and RPDOs.

PDO communication parameters and PDO mapping parameters define a PDO.

Link the PDO with the elements of the object directory that contain the process data.

Free PDO mapping (Page 174)

Predefined connection set (Page 172) .

Parameter area for PDO	RPDO		TPDO	
	In the converter	In CANopen	In the converter	In CANopen
Communication parameters	p8700 ... p8707	1400 hex ... 1407 hex	p8720 ... p8727	1800 hex ... 1807 hex
Mapping parameters	p8710 ... p8717	1600 hex ... 1607 hex	p8730 ... p8737	1A00 hex ...1A07 hex

Structure of the PDO

A PDO consists of communication and mapping parameters. Examples for the structure of the TPDO and RPDO follow.

Values for communication parameters:

Tables in the Section Object directories (Page 183)

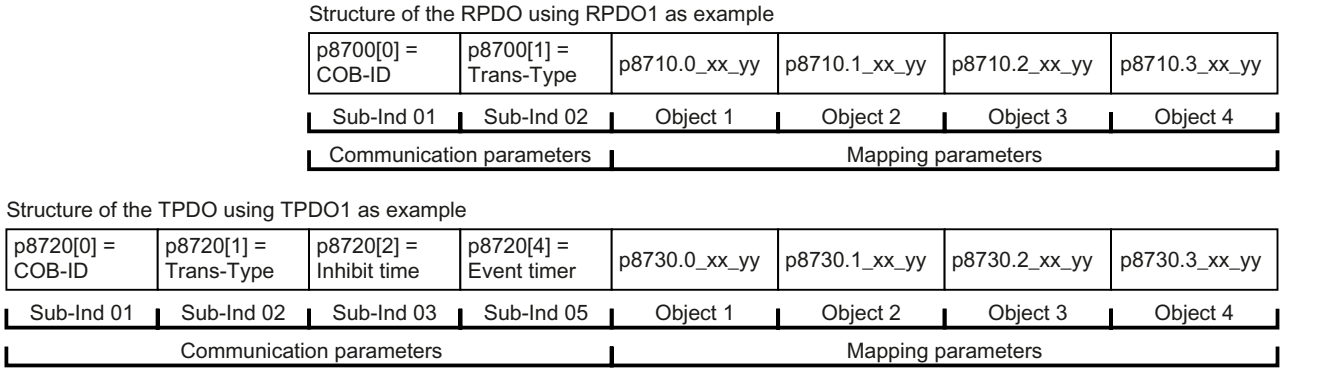


Figure 6-7 Structure of the RPDO and TPDO communication objects

COB ID

Overview: Communication over CANopen (Page 159).

Calculating the COB IDs: Predefined connection set (Page 172)

Transmission type

For process data objects, the following transmission types are available, which you set in index 1 of the communication parameter (p8700[1] ... p8707[1] / p8720[1] ... p8727[1]) in the converter:

- Synchronous cyclic (value range: 1 ... 240)
  - TPDO after each n-th SYNC
  - RPDO after each n-th SYNC
- Acyclic synchronous (value: 0)
  - TPDO when a SYNC is received and a process data has changed in the telegram.
- Cyclic asynchronous (values: 254, 255 + event time)
  - TPDO when a process data has changed in the telegram.

- Acyclic asynchronous (values: 254, 255)
  - TPDO sent in the Event Time interval.
  - The controller accepts the RPDO immediately.
- Synchronous data transmission
  A periodic synchronization object (SYNC object) ensures that the devices on the CANopen bus remain synchronized during transmission.
  Each PDO transferred as synchronization object must include a "transmission type" 1 ... n:
  - Transmission type 1: PDO in each SYNC cycle
  - Transmission type n: PDO in every n-th SYNC cycle

  The following diagram shows the principle of synchronous and asynchronous transmission:

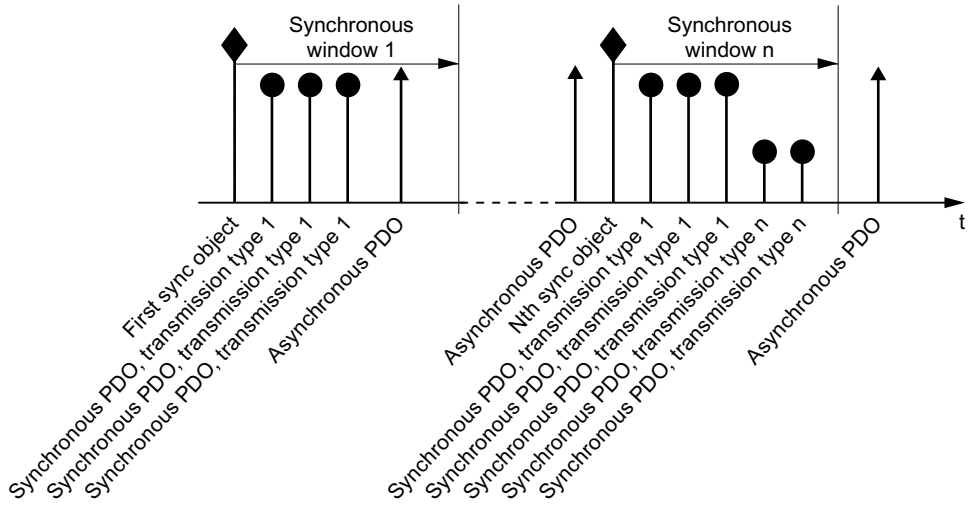


Figure 6-8 Principle of synchronous and asynchronous transmission

For synchronous TPDOs, the transmission mode also identifies the transmission rate as a factor of the SYNC object transmission intervals.
The CAN controller transfers data from synchronous RPDOs that it received after a SYNC signal only after the next SYNC signal to the converter.

Note

The SYNC signal synchronizes only the communication on the CANopen bus and not functions in the converter, e.g. the clock times of the speed control.

Inhibit time

The inhibit time defines the minimum interval between two transmissions.

PDO services

The following services are available for CANopen:

- PDO Write protocol
- PDO Read protocol

SINAMICS converters support the PDO Write protocol

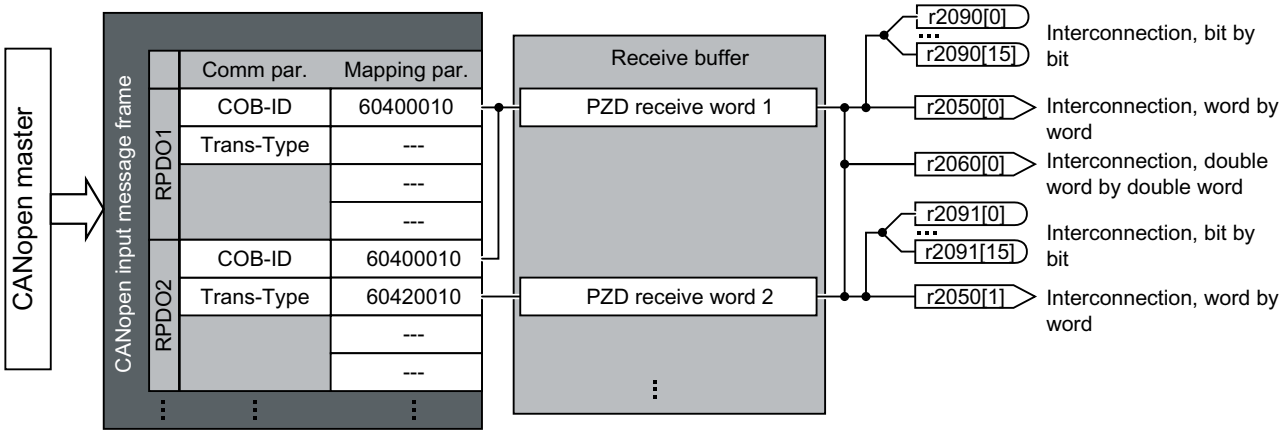
Write PDO

The "PDO Write protocol" service is based on the push model. The PDO has exactly one producer. There can be no consumer, one consumer, or multiple consumers.

Via Write PDO, the producer of the PDO sends the data of the mapped application object to the individual consumer.

6.3.1 Predefined connection set

If you integrate the converter using the factory setting in CANopen, the converter receives the control word and the speed setpoint from the controller. The converter returns the status word and the actual speed value to the controller. These are the settings stipulated in the Predefined Connection Set.



Structure of the communication parameter using the control word in the predefined connections set as example

RPDO1: Communication parameter
- p8700[0] = COB-ID
- p8700[1] = transmission type

Structure of the mapping parameter using the control word in the predefined connections set as example

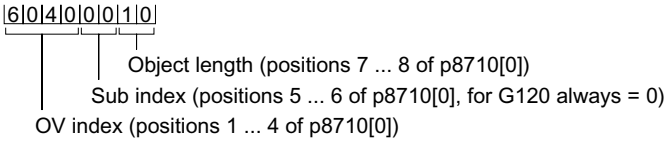
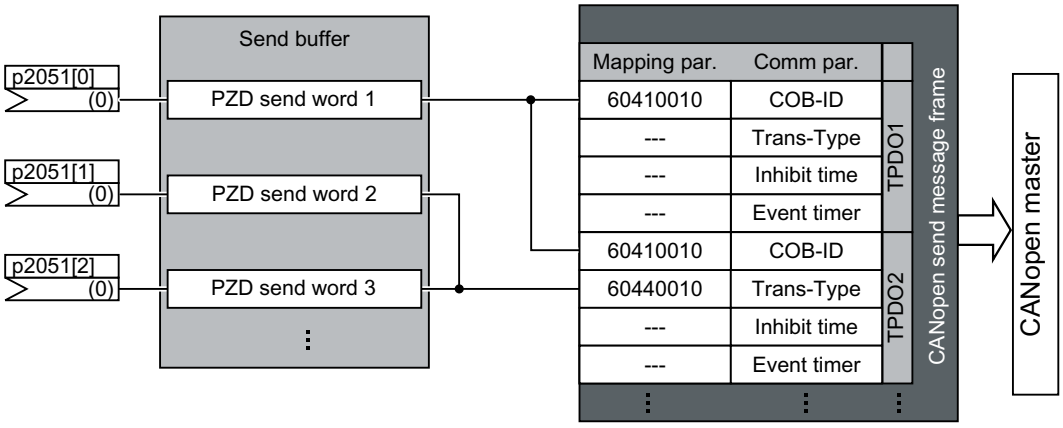


Figure 6-9 RPDO mapping with the Predefined Connection Set



Structure of the communication parameter using the status word in the predefined connections set as example

TPDO1: Communication parameter
- p8720[0] = COB-ID
- p8700[1] = transmission type
- p8700[2] = inhibit time
- p8700[3] = event timer

Structure of the mapping parameter using the control word in the predefined connections set as example

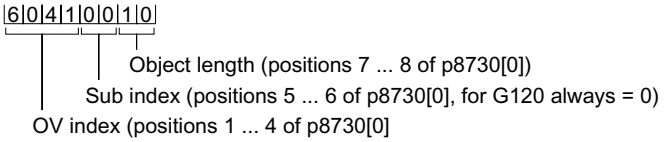


Figure 6-10 TPDO mapping with the Predefined Connection Set

6.3.2 Free PDO mapping

Using the free PDO mapping, you configure and interconnect any process data as required as follows:

- as free objects free objects (Page 192) or
- as objects of drive profile CiA 402, corresponding to the requirements of your system for the PDO service

The precondition is that the converter is set for free PDO mapping (p8744 = 2) (factory setting).

Configuring and mapping process data using free PDO mapping

Procedure

1. Specify the process data.
   Examples:
   - Send current actual value (r0068) from the converter to the controller (TPDO - Transmit Process Data Object)
   - Send additional speed setpoint from the controller to the converter (RPDO - Receive Process Data Object) and write in p1075
2. Specify objects for transmission of the process data
   - TPDO1 for the current actual value
   - RPDO1 for additional speed setpoint
3. Define the communication parameters for RPDO and TPDO
   - Define communication parameters for RPDO.
     RPDO communication parameters (Page 185)
   - Define communication parameters for TPDO.
     See TPDO communication parameters (Page 188)
4. Select the OV index for the mapping parameters
   - Mapping parameters for RPDO.
     RPDO mapping parameters (Page 186)
   - Mapping parameters for TPDO.
     TPDO mapping parameters (Page 190)
5. Write OV index into the SINAMICS mapping parameters
   - p8710 ... p8717 for RPDO
   - p8730 ... p8737 for TPDO

Note

Precondition for changing the OD indexes of the SINAMICS mapping parameters

To allow you to change the values of the mapping parameters, you must set the COB ID of the corresponding parameter to invalid. To do this, add a value of 80000000 hex to the COB-ID. You must reset the COB-ID to a valid value once you changed the mapping parameters.

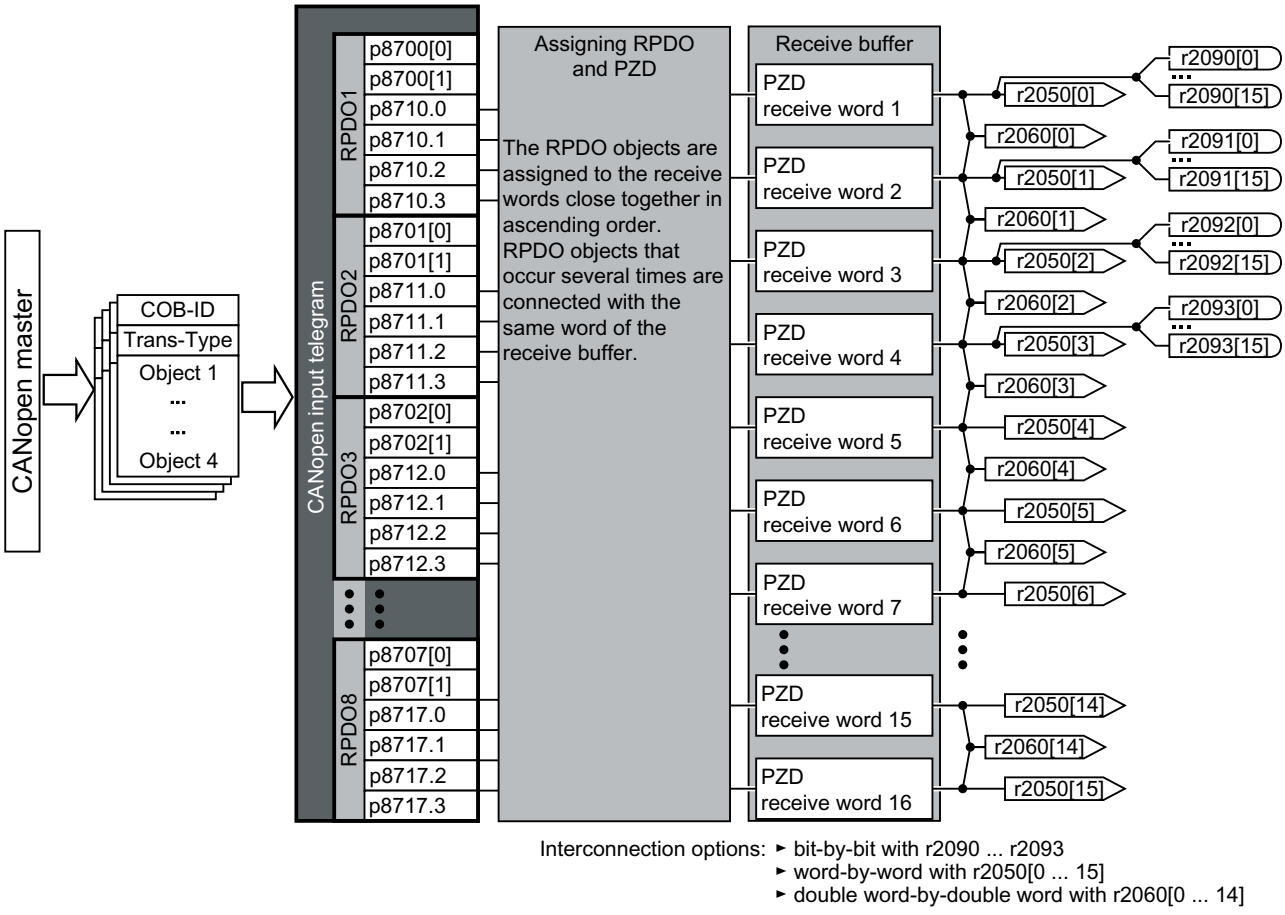
OV index:

Free objects (Page 192)

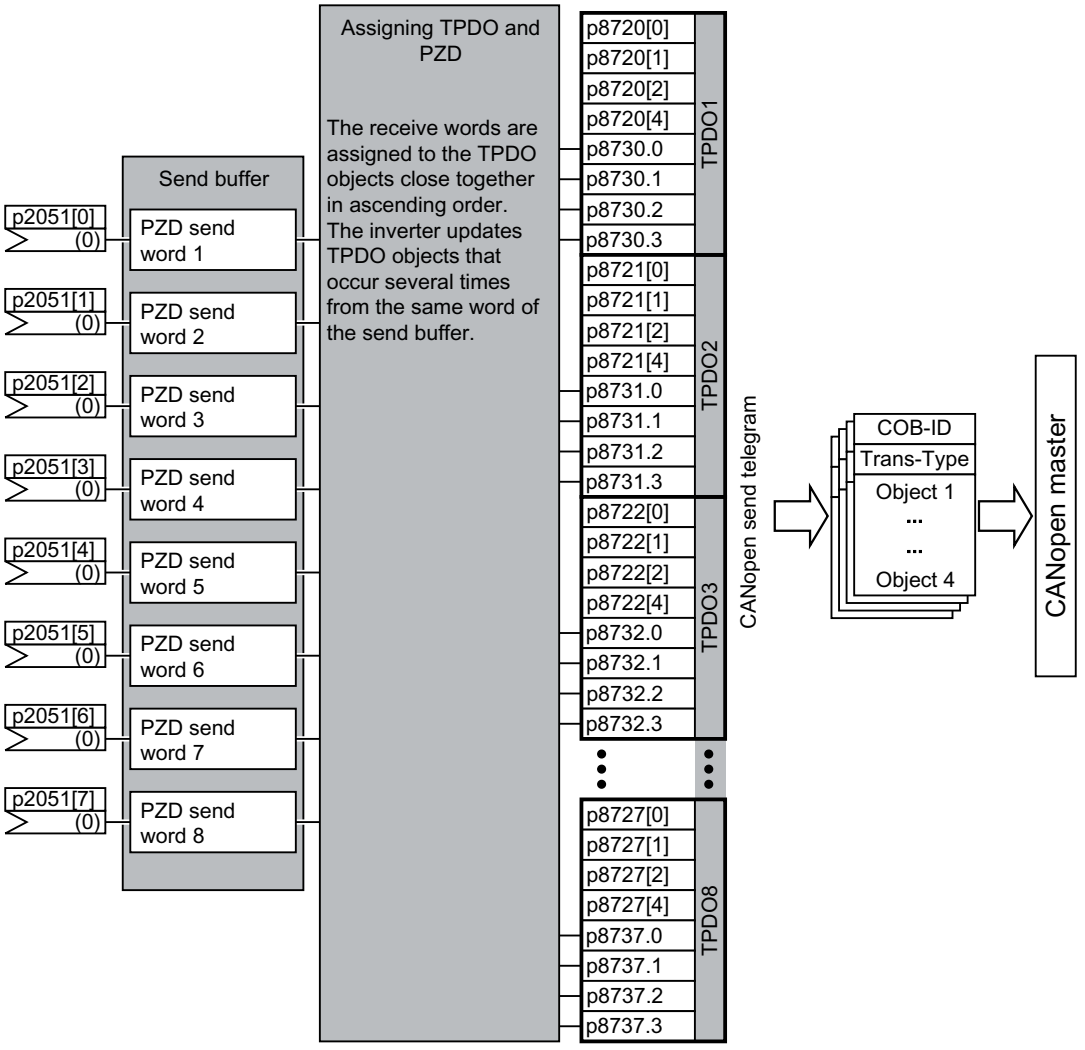
Objects of the drive profile CiA 402 (Page 193)

You have now configured and mapped the process data.

Free RPDO mapping - Overview



Free TPDO mapping - Overview



6.3.3 Interconnect objects from the receive and transmit buffers

To interconnect process data, proceed as follows:

Procedure

1. Create a telegram:
   create PDO (parameterize the PDO Com. Parameters and PDO mapping parameters).
   
   Predefined connection set (Page 172)
   Free PDO mapping (Page 174)
2. Interconnect parameters:
   Interconnect the parameters of the PZD buffer (r2050/r2060, p2051/p2061) corresponding to the mapping of point "Create telegram" using the mapping table r8750/r8760 or r8751/r8761. The mapping table indicates the position of a mapped CANopen object in the PZD buffer.

You have now interconnected the process data.

Interconnecting the receive buffer

The converter writes the received data to the receive buffer:

- PZD receive word 1 ... PZD receive word 12 double word in r2060[0] ... r2060[10].
- PZD receive word 1 ... PZD receive word 12 word in r2050[0] ... r2050[11].
- PZD 1 ... PZD 4 bit-by-bit in r2090.0 ... r2090.15 to r2093.0 ... r2093.15

The position of the mapped objects in the receive buffer is displayed in:

- r8760 for double word switching
- r8750 for word switching

Examples

Object	Mapped receive objects	Receive word r2050	
Control word	r8750[0] = 6040 hex (PZD1)	Interconnect r2050[0] (PZD1) in control word [1)]	p0840.0 = 2090.0 p0844.0 = 2090.1 p08484.0 = 2090.2 p0852.0 = 2090.3 p2130.0 = 2090.7
Torque limit	r8750[1] = 5800 hex (PZD2)	Interconnect r2050[1] (PZD2) in the torque limit:	p1522 = 2050[1]
Speed setpoint	r8750[2] = 6042 hex (PZD3)	Interconnect r2050[2] (PZD3) in the speed setpoint:	p1070 = 2050[2]

1) see also p8790, "Automatic CAN control word interconnection"

Interconnecting the send buffer

The converter transmits the data from the send buffer as follows:

- p2051[0] ... p2051[13] in PZD 1 ... PZD 14 (indication of the actual values in r2053[0 ... 13])
- p2061[0] ... p2061[12] in PZD 1 ... PZD 14 (indication of the actual values in r2063[0 ... 12])

Examples

Object	Mapped send objects	Send word p2051	
Status word	r8751[0] = 6041 hex (PZD1)	Interconnect p2051[0] in PZD1	p2051[0] = r8784
Current actual value	r8751[1] = 5810 hex (PZD2)	Interconnect PZD2 in the current actual value	p2051[1] = r68[1]
Actual speed value	r8751[2] = 6044 hex (PZD3)	Interconnect PZD3 in the speed actual value	p2051[2] = r63[0]

6.3.4 Free PDO mapping for example of the actual current value and torque limit

You integrate the actual current value and torque limit into the communication via the free PDO mapping.

The actual current value and the torque setpoint are transferred in TPDO1 and RPDO1, respectively. TPDO1 and RPDO1 have already been specified by the Predefined Connection Set.

Mapping the actual current value (r0068) with TPDO1

Proceed as follows to accept the current actual value as send object in the communication:

Procedure

1. Set the OV index for the actual current value:
   first free OV index from the send data from the "Free objects" 5810 table
2. Map the OV index for the actual current value with PZD2:
   - Set the COB-ID of TPDO1 to "invalid":
     p8720[0] = 800001B2 hex
   - Link the mapping parameter object 2 of TPDO1 (p8730.1) with the OV index for the actual current value:
     p8730.1 = 58100010 hex (5810 = OV index, 00 = fixed value, 10 ≙ 16 bit value)
   - Set the COB-ID of TPDO1 to "valid":
     p8720[0] = 400001B2 hex

   r8751 shows which object is matched to which PZD:
   PZD2 (r8751[1]) = 5810 (actual current value)
3. link the PZD send word 2 in the send word (p2051) with the actual current value:
   p2051[1] = r0086[0]

You have now transferred the actual current value into the communication as the send object.
❒

Mapping the torque limit (p1520) with RPDO1

Proceed as follows to accept the torque limit value in the communication:

Procedure

1. Set the OV index for the torque limit:
   first free OV index from the receive data from the "Free objects" 5800 table
2. Map the OV index for the torque limit with PZD2
   - Set the COB-ID of RPDO1 to "invalid":
     p8700[0] = 80000232 hex
   - Link the mapping parameter object 2 of RPDO1 (p8710.1) with the OV index for the torque limit:
     p8710.1 = 58000010 hex (5800 = OV index, 00 = fixed value)
   - Set the COB-ID of RPDO1 to "valid":
     p8700[0] = 40000232 hex

   r8750 shows which object is mapped to which PZD:
   PZD2 (r8750[1]) = 5800 (torque limit)
3. Link the PZD receive word 2 in the receive word (p2050) with the torque limit:
   p2050[1] = p1520[0]

You have now transferred the value for the torque limit into the communication.
❒

6.4 CANopen operating modes

The converter has the following CANopen operating modes

CANopen operating mode			SINAMICS				
Active operating mode	Setting in 6060 h: Value	6502 h: Display the active operating mode in	Open-loop/closed-loop control mode	Control Unit / converter			Value in p1300
				CU230P-2 CAN	G120C CAN	CU250S-2 CAN	
Velocity Mode	2	Bit1	U/f control with linear characteristic	x	x	x	0
Manufacturer-specific operating mode 1	-1	Bit16	U/f control with linear characteristic and FCC	x	x	x	1
Manufacturer-specific operating mode 2	-2	Bit17	U/f control with parabolic characteristic	x	x	x	2
Manufacturer-specific operating mode 3	-3	Bit18	U/f control with parameterizable characteristic		X	x	3
Manufacturer-specific operating mode 4	-4	Bit19	U/f control with linear characteristic and ECO	x	x	x	4
Manufacturer-specific operating mode 5	-5	Bit20	U/f control for drive requiring a precise frequency (e.g. textiles)		X	x	5

Manufacturer-specific operating mode 6	-6	Bit21	U/f control for drive requiring a precise frequency and FCC		X	x	6
Manufacturer-specific operating mode 7	-7	Bit22	U/f control with parabolic characteristic and ECO	x	x	x	7
Manufacturer-specific operating mode 8	-15	Bit23	Operation with braking resistor			x	15
Manufacturer-specific operating mode 10	-19	Bit25	U/f control with independent voltage setpoint		x	x	19
Manufacturer-specific operating mode 11	-20	Bit26	Speed control (without encoder)	x	x	x	20
Profile Velocity Mode	3	Bit2	Speed control (with encoder)			x	21
Manufacturer-specific operating mode 12	-22	Bit27	Torque control (without encoder)			x	22
Profile Torque Mode	4	Bit3	Closed-loop torque control (with encoder)			x	23

Switching the CANopen operating modes

		Switching from		
		Velocity mode	Profile velocity mode	Profile torque mode
to	Velocity mode		p1300 < 20 V/f control	p1300 < 20 V/f control
	Profile velocity mode	p1300 = 20 / 21 Speed control		p1500 = 0 (via BiCo), speed control
	Profile torque mode	p1300 = 22 / 23 Speed control	p1500 = 1 (via BiCo), torque control	

Parameter access via SDO Parameter change via PDO

You can also use parameters from other CANopen operating modes, independently from the current effective CANopen operating mode.

6.5 RAM to ROM via the CANopen object 1010

Save the parameters of the converter EEPROM using CANopen object 1010.

The following options are available:

- 1010.1: Save all parameters - identical with p0971 = 1, or back them up so they are not lost if the power fails.
- 1010.2: Save communication parameters - not possible via parameter settings!
- 1010.3: Save application parameters - not possible via parameter settings!

If a memory card is inserted, write the parameter settings via the control with object 1010.1 into the EEPROM and to the memory card. You can carry out series commissioning with the memory card.

For additional information, please refer to the operating instructions, Chapter "Backing up data and series commissioning"

Overview of the manuals (Page 217),

Note

Save data using objects 1010.2 and 1010.3

Although you can write the communication and/or application parameters to the EEPROM using objects 1010.2 and 1010.3, you cannot write them to the memory card. This also means that it is not possible to load the communication data or only the application data from one converter into the next via the memory card.

Note

With the USB cable inserted, save the data in the converter via the control.

If the converter is connected with a computer via USB but Startdrive cannot access the converter online, then you cannot save data in the converter using CANopen object 1010.

Withdraw the USB cable from the converter if you wish to save the parameter setting with object 1010.1 in the converter via the control.

6.6 Object directories

6.6.1 General objects from the CiA 301 communication profile

Overview

The following table lists the drive-independent communication objects. The "SINAMICS parameters" column shows the parameter numbers assigned in the converter.

Table 6-2 Drive-independent communication objects

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written
1000		Device type	r8600	SDO	U32	–	r
1001		Error register	r8601	SDO	U8	–	r
1003	0...52 hex	Predefined error field	p8611[0...82]	SDO	U32	0	r/w
	0	Number of errors	p8611.0	SDO	U32	0	rw
	1	Number of module	p8611.1	SDO	U32	0	r
	2	Number of errors: module 1	p8611.2	SDO	U32	0	r
	3-A	Standard error field: module 1	p8611.3- p8611.10	SDO	U32	0	r
	B	Number of errors: module 2	p8611.11	SDO	U32	0	r
	C-13	Standard error field: module 2	p8611.12- p8611.19	SDO	U32	0	r
	14	Number of errors: module 3	p8611.20	SDO	U32	0	r
	15-1C	Standard error field: module 3	p8611.21- p8611.28	SDO	U32	0	r
	1D	Number of errors: module 4	p8611.29	SDO	U32	0	r
	1E-25	Standard error field: module 4	p8611.30- p8611.37	SDO	U32	0	r
	26	Number of errors: module 5	p8611.38	SDO	U32	0	r
	27-2E	Standard error field: module 5	p8611.39- p8611.46	SDO	U32	0	r
	2F	Number of errors: module 6	p8611.47	SDO	U32	0	r
	30-37	Standard error field: module 6	p8611.48- p8611.55	SDO	U32	0	r
	38	Number of errors: module 7	p8611.56	SDO	U32	0	r
	39-40	Standard error field: module 7	p8611.57- p8611.64	SDO	U32	0	r

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written
	41	Number of errors: module 8	p8611.65	SDO	U32	0	r
	42-49	Standard error field: module 8	p8611.66-p8611.73	SDO	U32	0	r
	4A	Number of Control Unit faults	p8611.74	SDO	U32	0	r
	4B-52	Field Control Unit standard error	p8611.75-p8611.82	SDO	U32	0	r
1005		SYNCH COB ID	p8602	SDO	U32	128	rw
1008		Manufacturer device name		SDO			
100A		Manufacturer software version	r0018	SDO	U32	–	r
100C		Guard time	p8604.0	SDO	U16	0	rw
100D		Lifetime factor	p8604.1	SDO	U16	0	rw
1010		Store parameters	p0971	SDO	U16	0	rw
	0	Largest subindex supported		SDO			
	1	Save all parameters	p0971	SDO	U16	0	rw
	2	Save communication parameters (0x1000-0x1fff)	p0971	SDO	U16	0	rw
	3	Save application-related parameters (0x6000-0x9fff)	p0971	SDO	U16	0	rw
1011		Restore default parameters	p0970	SDO	U16	0	rw
	0	Largest subindex supported		SDO			
	1	Restore all default parameters	p0970	SDO	U16	0	rw
	2	Restore communication default parameters (0x1000-0x1fff)	p0970	SDO	U16	0	rw
	3	Restore application default parameters (0x6000-0x9fff)	p0970	SDO	U16	0	rw
1014		COB ID emergency	p8603	SDO	U32	0	rw
1017		Producer heartbeat time	p8606	SDO	U16	0	rw
1018		Identy Object	r8607[0...3]		U32	–	r
	0	Number of entries		SDO			
	1	Vendor ID	r8607.0	SDO	U32	–	r
	2	Product code	r8607.1	SDO	U32	–	r
	3	Revision number	r8607.2	SDO	U32	–	r
	4	Serial number	r8607.3	SDO	U32	0	r
1027		Module list					

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written
	0	Number of entries	r0102	SDO	U16	–	r
	1-8	Module ID	p0107[0...15]	SDO	I16	0	rw
1029		Error behavior					
	0	Number of error classes		SDO			
	1	Communication Error	p8609.0	SDO	U32	1	rw
	2	Device profile or manufacturer-specific error	p8609.1	SDO	U32	1	rw
1200		1st server SDO parameter					
	0	Number of entries		SDO			
	1	COB ID client -> server (rx)	r8610.0	SDO	U32	–	r
	2	COB ID server -> client (tx)	r8610.1	SDO	U32	–	r

RPDO configuration objects

The following tables list the communication and mapping parameters together with the indexes for the individual RPDO configuration objects. The configuration objects are established via SDO. The "SINAMICS parameters" column shows the parameter numbers assigned in the converter.

Table 6-3 RPDO configuration objects - communication parameters

OD Index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
1400		**Receive PDO 1 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8700.0	U32	200 hex + node ID	r/w
	2	Transmission type	p8700.1	U8	FE hex	r/w
1401		**Receive PDO 2 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8701.0	U32	300 hex + node ID	r/w
	2	Transmission type	p8701.1	U8	FE hex	r/w
1402		**Receive PDO 3 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8702.0	U32	8000 06DF hex	r/w
	2	Transmission type	p8702.1	U8	FE hex	r/w
1403		**Receive PDO 4 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8703.0	U32	8000 06DF hex	r/w
	2	Transmission type	p8703.1	U8	FE hex	r/w
1404		**Receive PDO 5 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8704.0	U32	8000 06DF hex	r/w

OD Index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
	2	Transmission type	p8704.1	U8	FE hex	r/w
1405		**Receive PDO 6 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8705.0	U32	8000 06DF hex	r/w
	2	Transmission type	p8705.1	U8	FE hex	r/w
1406		**Receive PDO 7 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8706.0	U32	8000 06DF hex	r/w
	2	Transmission type	p8706.1	U8	FE hex	r/w
1407		**Receive PDO 8 communication parameter**				
	0	Largest subindex supported		U8	2	r
	1	COB ID used by PDO	p8707.0	U32	8000 06DF hex	r/w
	2	Transmission type	p8707.1	U8	FE hex	r/w

Table 6-4 RPDO configuration objects - mapping parameters

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
1600		**Receive PDO 1 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	1	r
	1	PDO mapping for the first application object to be mapped	p8710.0	U32	6040 hex	r/w
	2	PDO mapping for the second application object to be mapped	p8710.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8710.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8710.3	U32	0	r/w
1601		**Receive PDO 2 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	2	r
	1	PDO mapping for the first application object to be mapped	p8711.0	U32	6040 hex	r/w
	2	PDO mapping for the second application object to be mapped	p8711.1	U32	6042 hex	r/w
	3	PDO mapping for the third application object to be mapped	p8711.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8711.3	U32	0	r/w
1602		**Receive PDO 3 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
	1	PDO mapping for the first application object to be mapped	p8712.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8712.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8712.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8712.3	U32	0	r/w
1603		**Receive PDO 4 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8713.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8713.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8713.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8713.3	U32	0	r/w
1604		**Receive PDO 5 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8714.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8714.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8714.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8714.3	U32	0	r/w
1605		**Receive PDO 6 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8715.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8715.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8715.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8715.3	U32	0	r/w
1606		**Receive PDO 7 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8716.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8716.1	U32	0	r/w

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
	3	PDO mapping for the third application object to be mapped	p8716.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8716.3	U32	0	r/w
1607		**Receive PDO 8 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8717.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8717.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8717.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8717.3	U32	0	r/w

TPDO configuration objects

The following tables list the communication and mapping parameters together with the indexes for the individual TPDO configuration objects. The configuration objects are established via SDO. The "SINAMICS parameters" column shows the parameter numbers assigned in the converter.

Table 6-5 TPDO configuration objects - communication parameters

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written
1800		**Transmit PDO 1 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8720.0	U32	180 hex + node ID	r/w
	2	Transmission type	p8720.1	U8	FE hex	r/w
	3	Inhibit time	p8720.2	U16	0	r/w
	4	Reserved	p8720.3	U8	---	r/w
	5	Event timer	p8720.4	U16	0	r/w
1801		**Transmit PDO 2 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8721.0	U32	280 hex + node ID	r/w
	2	Transmission type	p8721.1	U8	FE hex	r/w
	3	Inhibit time	p8721.2	U16	0	r/w
	4	Reserved	p8721.3	U8	---	r/w
	5	Event timer	p8721.4	U16	0	r/w
1802		**Transmit PDO 3 communication parameter**				
	0	Largest subindex supported		U8	5	r

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written
	1	COB ID used by PDO	p8722.0	U32	C000 06DF hex	r/w
	2	Transmission type	p8722.1	U8	FE hex	r/w
	3	Inhibit time	p8722.2	U16	0	r/w
	4	Reserved	p8722.3	U8	---	r/w
	5	Event timer	p8722.4	U16	0	r/w
1803		**Transmit PDO 4 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8723.0	U32	C000 06DF hex	r/w
	2	Transmission type	p8723.1	U8	FE hex	r/w
	3	Inhibit time	p8723.2	U16	0	r/w
	4	Reserved	p8723.3	U8	---	r/w
	5	Event timer	p8723.4	U16	0	r/w
1804		**Transmit PDO 5 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8724.0	U32	C000 06DF hex	r/w
	2	Transmission type	p8724.1	U8	FE hex	r/w
	3	Inhibit time	p8724.2	U16	0	r/w
	4	Reserved	p8724.3	U8	---	r/w
	5	Event timer	p8724.4	U16	0	r/w
1805		**Transmit PDO 6 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8725.0	U32	C000 06DF hex	r/w
	2	Transmission type	p8725.1	U8	FE hex	r/w
	3	Inhibit time	p8725.2	U16	0	r/w
	4	Reserved	p8725.3	U8	---	r/w
	5	Event timer	p8725.4	U16	0	r/w
1806		**Transmit PDO 7 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8726.0	U32	C000 06DF hex	r/w
	2	Transmission type	p8726.1	U8	FE hex	r/w
	3	Inhibit time	p8726.2	U16	0	r/w
	4	Reserved	p8726.3	U8	---	r/w
	5	Event timer	p8726.4	U16	0	r/w
1807		**Transmit PDO 8 communication parameter**				
	0	Largest subindex supported		U8	5	r
	1	COB ID used by PDO	p8727.0	U32	C000 06DF hex	r/w
	2	Transmission type	p8727.1	U8	FE hex	r/w
	3	Inhibit time	p8727.2	U16	0	r/w
	4	Reserved	p8727.3	U8	---	r/w
	5	Event timer	p8727.4	U16	0	r/w

Table 6-6 TPDO configuration objects - mapping parameters

OD index (hex)	Subin-dex (hex)	Object name	SINAMICS parame-ters	Data type	Predefined connection set	Can be read/ written
1A00		**Transmit PDO 1 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	1	r/w
	1	PDO mapping for the first application object to be mapped	p8730.0	U32	6041 hex	r/w
	2	PDO mapping for the second application object to be mapped	p8730.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8730.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8730.3	U32	0	r/w
1A01		**Transmit PDO 2 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	2	r/w
	1	PDO mapping for the first application object to be mapped	p8731.0	U32	6041 hex	r/w
	2	PDO mapping for the second application object to be mapped	p8731.1	U32	6044 hex	r/w
	3	PDO mapping for the third application object to be mapped	p8731.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8731.3	U32	0	r/w
1A02		**Transmit PDO 3 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r/w
	1	PDO mapping for the first application object to be mapped	p8732.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8732.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8732.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8732.3	U32	0	r/w
1A03		**Transmit PDO 4 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r/w
	1	PDO mapping for the first application object to be mapped	p8733.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8733.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8733.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8733.3	U32	0	r/w
1A04		**Transmit PDO 5 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8734.0	U32	0	r/w

OD index (hex)	Subin-dex (hex)	Object name	SINAMICS parame-ters	Data type	Predefined connection set	Can be read/ written
	2	PDO mapping for the second application object to be mapped	p8734.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8734.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8734.3	U32	0	r/w
1A05		**Transmit PDO 6 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r/w
	1	PDO mapping for the first application object to be mapped	p8735.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8735.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8735.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8735.3	U32	0	r/w
1A06		**Transmit PDO 7 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8736.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8736.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8736.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8736.3	U32	0	r/w
1A07		**Transmit PDO 8 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	1	PDO mapping for the first application object to be mapped	p8737.0	U32	0	r/w
	2	PDO mapping for the second application object to be mapped	p8737.1	U32	0	r/w
	3	PDO mapping for the third application object to be mapped	p8737.2	U32	0	r/w
	4	PDO mapping for the fourth application object to be mapped	p8737.3	U32	0	r/w

6.6.2 Free objects

You can interconnect any process data objects of the receive and transmit buffer using receive and transmit double words.

- Scaling for percentage values:
  - 16-bit (word): 4000 hex ≙ 100%
  - 32-bit (doubleword) 4000000 hex ≙ 100%
- Scaling for values referred to units:
  - 16-bit (word): 4000 hex ≙ value of the corresponding reference parameter for p200x
  - 32-bit (double word) 4000000 hex ≙ Value of the corresponding reference parameter for p200x

  Example:
  - For temperature values: 16-bit (word): 4000 hex ≙ p2006
  - For temperature values: 32-bit (doubleword): 4000000 hex ≙ p2006

The "SINAMICS parameters" column shows the parameter numbers assigned in the converter. The assignment applies to the case in which an object which is not mapped in any PDO is to be accessed via SDO.

OD index (hex)	Description	Data type per PZD	Default setting	Can be written to-/ read	SINAMICS parameters
5800 ... 580F	16 freely-interconnectable receive process data	I16	0	r/w	r8745[0 ... 15]
5810 ... 581F	16 freely-interconnectable transmit process data	I16	0	r	r8746[0 ... 15]
5820 ... 5827	8 freely-interconnectable receive process data	I32	0	r/w	r8747[0 ... 7]
5828 ... 582F	Reserved				
5830 ... 5837	8 freely-interconnectable transmit process data	I32	0	r	r8748[0 ... 7]
5828 ... 582F	Reserved				

6.6.3 Objects from the CiA 402 drive profile

The following table lists the object directory with the index of the individual objects for the drives. The "SINAMICS parameters" column shows the parameter numbers assigned in the converter.

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Transmission	Data type	Default setting	Can be read/ written
Predefinitions							
67FF		Single device type		SDO	U32		r
Common entries in the object dictionary							
6007		Abort connection option code	p8641	SDO	I16	3	r/w
6502		Supported drive modes		SDO	I32		r
6504		Drive manufacturer		SDO	String	SIEMENS	r
Device control							
6040		Control word	r8795	PDO/SDO	U16	–	r/w
6041		Status word	r8784	PDO/SDO	U16	–	r
605D		Halt option code	p8791	PDO/SDO	I16	–	r/w
6060		Modes of operation	p1300	SDO	I8	–	r/w
6061		Modes of operation display	r8762	SDO	I8	–	r
Factor group							
6094		Velocity encoder factor		SDO	U8	-	r
	01	velocity encoder factor numerator	p8798[1]	SDO	U32	1	r/w
	02	velocity encoder factor denumerator	p8798[2]	SDO	U32	1	r/w
Profile velocity mode							
6063		Actual position value	r0482	SDO/PDO	I32	–	r
6069		Velocity sensor actual value	r0061	SDO/PDO	I32	–	r
606B		Velocity demand value	r1170	SDO/PDO	I32	–	r
606C		Velocity actual value Actual velocity	r0063	SDO/PDO	I32	–	r
6083		Profile acceleration	p1082/p1120	SDO	I32	–	r/w
6084		Profile deceleration	p1082/p1121	SDO	I32	0	r/w
6085		Quick stop deceleration	p1082/p1135	SDO	I32	0	r/w
6086		Motion profile type	p1115/p1134	SDO	I32	0	r/w
60FF		**Target** velocity Set velocity	p1155[0] [1] p1072 [2]	SDO/PDO	I32	0	r/w
Profile Torque Mode [3]							
6071		Target torque torque setpoint	r8797	SDO/PDO	I16	–	r/w
6072		Max. torque	p1520	SDO	0	0	
6074		Torque demand value overall torque setpoint	r0079	SDO/PDO	I16	–	r
6077		Torque actual value	r0080	SDO/PDO	I16	–	r

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Transmission	Data type	Default setting	Can be read/ written
Velocity mode							
6042		vl target velocity	r8792	SDO/PDO	I16	–	r/w
6043		vl velocity demand	r1170	SDO/PDO	I16	–	r
6044		vl velocity actual value	r0063	SDO/PDO	I16	–	r
6046	0	vl velocity min./max. amount		SDO	U8	–	r
	1	vl velocity min. amount	p1080	SDO	U32	–	r/w
	2	vl velocity max. amount	p1082	SDO	U32	–	r/w
6048	0	vl velocity acceleration		SDO	U8	–	r
	1	Delta speed	p1082	SDO	U32	–	r/w
	2	Delta time	p1120	SDO	U16	–	r/w

1) Without ramp-function generator

2) With ramp-function generator

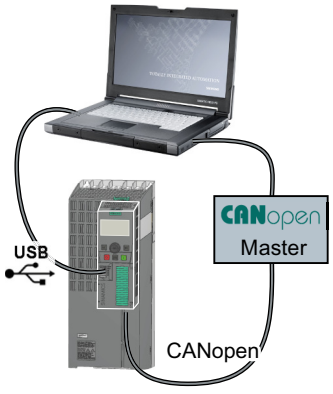
3) The converter can process the objects from the Profile Torque Mode. But they cannot be set nor selected in converters.

6.7 Integrating the converter into CANopen

Commissioning

Requirement

- Startdrive is installed on the computer used to commission the system.
- The converter is connected to a CANopen manager.
- The EDS (Electronic Data Sheet) is installed on your CANopen manager.
- The converter interfaces have been set to the CANopen fieldbus during basic commissioning.
  This means that the following signals in the converter are interconnected corresponding to the Predefined Connection Sets:
  - Speed setpoint and control word
  - Speed actual value and status word



You can find the EDS on the Internet:

EDS (electronic data sheet) (http://support.automation.siemens.com/WW/view/en/48351511)

Procedure

1.  Connecting converter to CAN bus (Page 195)
2. Set the node ID, baud rate and the communication monitoring.
    Setting the node ID and baud rate (Page 195)"
    Setting the monitoring of the communication (Page 197)"
3. Interconnect additional process data
   Set p8744 = 2. You can now interconnect other process data.
    Free PDO mapping (Page 174)"
4. Signal interconnection of the links created in free PDO mapping.
    Interconnect objects from the receive and transmit buffers (Page 177).
5. Exit commissioning
   This is carried out in Startdrive in the commissioning wizard.

You have now commissioned the CANopen interface.

Further information about configuring the communication:

 Communication over CANopen (Page 159)

 Object directories (Page 183).

6.7.1 Connecting converter to CAN bus

Connect the converter to the fieldbus via the 9-pin SUB-D pin connector.

The connections of this pin connector are short-circuit proof and isolated. If the converter forms the first or last device in the CANopen network, then you must switch-in the bus-terminating resistor.

For additional information, refer to the operating instructions of the Control Unit.

6.7.2 Setting the node ID and baud rate

Node ID

Valid value range: 1 ... 127

You have the following options for setting the node ID:

- Using the address switch on the Control Unit:

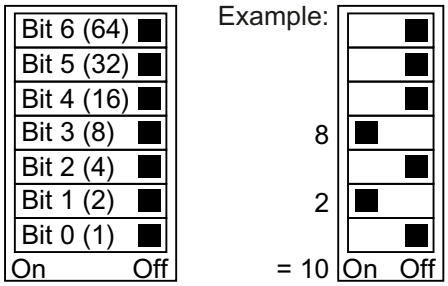


Figure 6-11 Address switch with example for bus address 10

The address switch has priority over the other settings.

- Using Startdrive or an operator panel via parameter p8620 (default setting: p8620 = 126)
  It is only possible to change p8620 if the address 0 is set in the address switch.
  If you are working with Startdrive, back up the settings so they are not lost if the power fails.

The position of the address switch can be found in the operating instructions for the converter.

Manuals and technical support (Page 217)

Setting the baud rate

You can set the baud rate using parameter p8622. If you are working with Startdrive, back up the settings so they are not lost if the power fails.

Setting range: 10 kbps ... 1 Mbps. The maximum permissible cable length for 1 Mbps is 40 m.

Activating node ID or baud rate

Procedure

To activate the changed node ID or baud rate, proceed as follows:

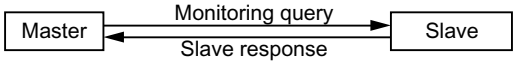
1. Switch off the converter supply voltage.
2. Wait until all LEDs on the converter are dark.
3. Switch the converter supply voltage on again.
   Your settings become effective after switching on.

You have now activated the changed settings.

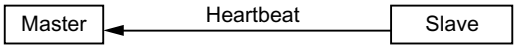
6.7.3 Setting the monitoring of the communication

To monitor the communication, use one of the following methods:

- Node guarding / life guarding



- Heartbeat



Node guarding / life guarding

Principle of operation

- Node guarding:
  Is always active if heartbeat is not activated (p8606 = 0). Node guarding means the manager sends monitoring queries to the converter which then answers.
  The converter does not monitor the communication. Set the responses to a bus failure in the manager.
- Life guarding:
  is active if you use p8604.0 and p8604.1 to set a lifetime ≠ 0.
  Life Guarding means that the converter monitors the manager's monitoring query and reports fault F8700 (A) with fault value 2, if a life guarding protocol (life guarding event) is not received within the lifetime. Set additional responses to a bus failure in the manager.
  Calculate value for lifetime
  Life time = guard time in milliseconds (p8604.0) * life time factor (p8604.1)

Heartbeat

Principle of operation

The device periodically sends heartbeat messages. Other devices and the manager can monitor this signal. In the manager, set the responses for the case that the heartbeat does not come.

Setting value for heartbeat

Set in p8606 the cycle time for the heartbeat in milliseconds.

Converter behavior with a bus fault

With a bus fault, the CAN manager goes to the "Bus OFF" status. In the converter, set the response to the bus error using parameter p8641. Factory setting: p8641 = 3 (AUS3).

If you have resolved the bus error, then you have the following options to restart communication:

- You switch off the converter power supply, wait until all of the LEDs on the converter go dark, and then you switch on the converter power supply again.
  This means that you withdraw the bus state and restart communication.
- You acknowledge the bus error via DI 2 or directly via p3981 and start the communication either
  - Manually by setting p8608[0] = 1. After starting, p8608 is internally set back to 0.
  - Automatically every two seconds. To do this, you must have set p8608[1] to 1 when commissioning.

⚠ WARNING

OFF command not effective as a result of a bus fault

When the bus has a fault condition, the higher-level control cannot access the converter. If, as a response to a bus fault p8641 = 0 (no response) is set, then the motor remains switched-on - even if the higher-level control sends an OFF command to the converter.

- Configure an additional OFF command via terminals.

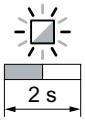
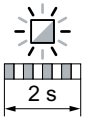
6.8 Error diagnostics

Objects to signal and describe errors and operating states

The following options are available to display errors and operating states:

- Display of the operating state using LEDs
- Display of the operating state using the alarm object (Emergency Object)
  - Converter-specific error list (predefined error field)
  - CANopen error register (error register)

Description of the LED symbols for CANopen

	LED is bright
	LED flashes slowly
	LED flashes quickly

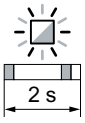
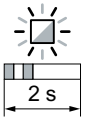
	LED flashes in the "single flash" mode
	LED flashes in the "double flash" mode
	LED flashes with variable frequency

Table 6-7 CANopen fieldbus

BF	Explanation
	Data exchange between the converter and control system is active ("Operational" state)
	Fieldbus is in the "Pre-operational" state
	Fieldbus is in the "Stopped" state
	No fieldbus available
	RDY  When LED RDY flashes simultaneously: Firmware update failed
	Alarm - limit reached
	Error event in the higher-level control system (Error Control Event)
	Converter waits until the power supply is switched off and switched on again after a firmware update
	Incorrect memory card or unsuccessful firmware update
	Firmware update is active

Display of the operating state using the alarm object (Emergency Object)

Error states are displayed using the alarm object (Emergency Object), OV index 1014 in the emergency telegram. It has the following structure:

Byte 0 ... 1	Byte 2	Byte 3 ... 4	Byte 5	Byte 6	Byte 7
CANopen Errorcode	CANopen Error Register	SINAMICS fault number	Drive object (always = 1)	Reserved	Reserved

- Bytes 0 and 1: CANopen error code
- Byte 2: Codes for the CANopen error register
- Byte 5: Number of the drive object. For G120 converters, this is always = 1

Errors trigger an emergency telegram and cause the drive to shut down.

You can suppress the emergency telegram by setting bit 31 in object 1014 hex to 1.

This means that shutdown is not suppressed, however the fault message is not sent to the manager.

Converter-specific error list (predefined error field)

You can read out the converter-specific error list using the following objects:

- OV index 1003 hex
- Converter parameter p8611

It includes the alarms and faults present in the converter in the CANopen alarm number range 8700-8799.

The errors are listed in the order in which they occur using an error code and additional, device-specific information.

As soon as a fault is acknowledged or an alarm is resolved, they are deleted from the converter-specific error list.

You acknowledge all of the active converter errors by setting subindex 0 in the OV index 1003 to 0 or setting p8611[0] = 0.

Table 6-8 CANopen error code

Error code	Meaning	Explanation
0000 hex	No error present	Successful acknowledgement of all errors or all the alarms are cleared in the display.
1000 hex	CAN Error 1	All other SINAMICS faults/errors
1001 hex	CAN Error 2	All other CANopen alarms in the alarm number range F08700 to F08799
8110 hex	CAN overflow, message lost	CBC: Telegram loss (A(N)08751) [alarm]
8120 hex	CAN Error Passive	CBC: Error number for Error Passive exceeded (A08752) [alarm]
8130 hex	CAN Life Guard Error	CBC: Communications error, alarm value 2 F08700(A) [error/alarm]

CANopen error register (error register)

You can read out the error register using the following objects:

- OV index 1001 hex
- Converter parameter r8601

It indicates the error in byte 2 of the emergency telegram.

Table 6-9 CANopen Error Register

Error Register	Meaning	Explanation
Bit 0	Generic error	Set for every alarm that CAN identifies.
Bit 4	Communication error	Is set for CAN communication alarms (alarms in the range 08700 ... 08799).
Bit 7	Manufacturer error	Is set for all alarms outside the range 08700 ... 08799.

Response in the case of an error

For a CAN communication error, e.g. too many telegram failures, the converter outputs fault F(A)08700(2).

For further information, please refer to the List Manual of your converter.

Overview of the manuals (Page 217)).

You set the response of the CAN node in p8609.

- p8609 = 0 Pre-operational
- p8609 = 1 No change (factory setting)
- p8609 = 2 Stopped

You set the converter response in p8641:

- p8641 = 0 No reaction (factory setting)
- p8641 = 1 OFF1
- p8641 = 2 OFF2
- p8641 = 3 OFF3

6.9 CAN bus sampling time

The CAN bus sampling time is 4 ms. The converter can send and receive telegrams within this time frame.

Receive telegrams cycle time

- For cyclic receive telegrams, the cycle time must be greater than twice the sampling time. Telegrams could be lost if the cycle time is any less than this. In this case, warning A08751 appears.
- With receive telegrams for which data does not change faster than twice the sampling time, it is possible to set a cycle time shorter than twice the sampling time if your application permits telegrams to be lost in the process.
  An A08751 warning can be avoided by changing the message type to "No message" via parameters p2118 and p2119.

Communication via AS-i – only for G115D

7

General information

The converter operates based on the extended AS-i specification V3.0.

The signaling is made as Manchester-coded current pulses superimposed on the 28 V supply. Decouple the 28 V supply with inductances so that the receivers can decouple the transferred messages.

The Control Unit power consumption is approx. 90 mA provided you do not use any digital or analog inputs. When you use digital and analog inputs, the power requirement can be as high as 300 mA.

The converter supports the Single device and Dual device modes.

In Single device mode, the converter has an address in the AS-i network over which four bits are transferred. In Dual device mode, each converter has two AS-i addresses over each of which four bits are transferred.

In the Single device mode, communication is realized in accordance with protocol 7.F.E. In the Dual device mode, communication is realized in accordance with protocols 7.A.5 and 7.A.E.

Default settings for commissioning

To configure the communication of the converter via AS-i, the following options are available for commissioning the converter: Using parameter p0015, select which default setting you wish to accept:

Table 7-1 p0015 (macro drive unit)

p0015 =	Designation	Meaning
30	Default setting 30 - single device mode, standard addressing	Single device mode, where the control system specifies a fixed frequency
31	Default setting 31 - dual device mode with fixed setpoints	Dual device mode, where the control system specifies a fixed frequency
32	Default setting 32 - single device mode, modified addressing	Default setting 32 - Single device mode, modified addressing:
34	Default setting 34 - dual device mode with "ON/OFF1", "OFF2"	Dual device mode with "ON/OFF1", "OFF2", speed setpoint from the control system

Details about the default settings are provided in the operating instructions of your converter.

Overview of the manuals (Page 217)

Connection

The following table shows the AS-i plug assignment.

Table 7-2 Pin assignment

X03 AS-i (M12)	Pin	Function	Description
	1	AS-i +	AS-i plus signal
	2	0 V	Reference potential for terminal 4
	3	AS-i -	AS-i minus signal
	4	24 V	24 V auxiliary voltage
	5	Not assigned	

Requirement for applications in the USA and Canada

Use an external 24 V power supply with one of the following specifications:

- NEC Class 2
- Voltage/current-limited

More information

You can find more information about connection in the AS-Interface System Manual.

Overview of the manuals (Page 217)

7.1 Setting the address

As factory setting, all AS-i devices have address 0. Devices with address 0 are not included in the communication.

The addresses must be unique, although they can be mixed as required.

You have the following options when making the address assignment:

- Automatic addressing via the AS-i master
- Addressing via the addressing device
- Addressing via parameters

Before you set the address, you must specify whether the converter is integrated as Single device or Dual devicein the AS-i network.

- p2013 = 0: Single device (factory setting)
- p2013 = 2: Dual device

If for the commissioning you select the default setting 30 or 32 (Single device or 31 or 34 (Dual device), p2013 is assigned the appropriate value.

Note

Changes made to p2012 and p2013

Changes made to the p2012 and p2013 parameters take effect immediately after the change.

If you work with Startdrive, you must back up the changes so they are not lost when the system is switched off and on again.

Automatic addressing via the AS-i master

Single device

For automatic addressing, the address is specified by the AS-i master. For a Single device, the master checks which device has address 0 and assigns it the next free address. This address is also written to parameter p2012. If more than one device has address 0, an automatic addressing is not possible.

Dual device

For automatic addressing, the address is specified by the AS-i master. If both devices have address 0, the second device is hidden and the control assigns a valid address for device 1.

device 2 then becomes visible with address 0 and can be addressed.

Automatic addressing is not always possible for older AS-i masters. In this case, use the manual addressing and set the address from an addressing device, via Startdrive or from an operator panel on the converter.

You can find more information in the AS-Interface System Manual, "Setting the AS-i address" section.

Overview of the manuals (Page 217)

Addressing via the addressing device (e.g. 3RK1904-2AB02)

Addressing via the addressing device is made offline.

You can find more information

- in the AS-Interface System Manual, "Setting the AS-i address" section
- G115D Operating Instructions, "Using an AS-i addressing unit" section

Overview of the manuals (Page 217)

Addressing via parameters

The address assignment is made with the p2012[0] and p2012[1] parameters.

If you assign the address via Startdrive, you must back up the settings so that they are not lost if the power fails.

- Address range for Single device converter, profile 7.F.E
  - p2012[1]: 0 ... 31, range for the A address, 0A ... 31A
- Address range for Dual device converter, profile 7.A.5 or 7.A.E
  - p2012[0]: 0 ... 31, 33 ... 63 for device 1:
  - p2012[1]: 0 ... 31, 33 ... 63 for device 2:

  with
  - 0 ... 31 range for the A address, 0A ... 31A
  - 33 ... 63 range for B address, 1B ... 31B

7.2 Single Device mode

In Single Device mode, four bits are available for the communication between the AS-i master and the converter. The four bits are used to transfer process data. In parallel, the control can start a diagnostic request via AS-i.P0.

The following default settings are available; both work with profile 7.F.E.

- Default setting 30: Standard Single Device mode
- Default setting 32: Modified Single Device mode

Default setting 30: Standard Single Device mode

In standard addressing, the control specifies the speed setpoint via the motor control bits (AS-i.DO0 ... AS-i.DO3).

Control -> converter

- AS-i.DO0 -> p1020 = 2093.0 Fixed speed bit 0
- AS-i.DO1 -> p1021 = 2093.1 Fixed speed bit 1
- AS-i.DO2 -> p1022 = 2093.2 Fixed speed bit 2
- AS-i.DO3 -> p1023 = 2093.3 Fixed speed bit 3

Table 7-3 Fixed speeds via the motor control bits (Page 210).

Converter -> control

If the control specifies the speed setpoint, the converter replies:

- p2080[0] = 53.13 -> AS-i.DI0 Operational enable for PLC
- p2080[1] = 899.11 -> AS-i.DI1 Pulses enabled
- p2080[2] = 722.0 -> AS-i.DI2 State DI0
- p2080[3] = 722.1 -> AS-i.DI3 State DI1

If the control sends a diagnostic request via AS-i.P0, the converter replies with the currently pending fault or alarm messages.

Table 7-6 Alarm and fault messages via RP0 ... RP3 from the converter to the AS-i master (Page 211).

Default setting 32: Modified Single Device mode

In Single Device mode with modified addressing the control specifies the following:

Control -> converter

- AS-i.DO0 -> p3330.0 = 2093.0 ON clockwise / OFF 1
- AS-i.DO1 -> p3331.0 = 2093.1 ON counter-clockwise / OFF 1
- AS-i.DO2 -> p0810 = 2093.2 Speed via potentiometer or AI0
- AS-i.DO3 -> p2104 = 2093.3 Acknowledge errors with a positive edge
  p0852 = 2093.3 Operating enable, if p2093.3 = 1

Converter -> control

The converter sends as response:

- p2080[0] = 899.0 -> AS-i.DI0 Ready for switching on
- p2080[1] = 807.0 -> AS-i.DI1 Control priority
- p2080[2] = 722.0 -> AS-i.DI2 State DI0
- p2080[3] = 722.1 -> AS-i.DI3 State DI1

If an alarm or fault is pending in the converter, it sends a warning or fault message.

Table 7-6 Alarm and fault messages via RP0 ... RP3 from the converter to the AS-i master (Page 211).

Scaling factors for the speed

The scaling factor is specified via AS-i.P0 ... AS-i.P3. A diagnostic request is also performed when AS-i.P0 is sent.

This means, if the control specifies a scaling factor and an alarm or fault is pending in the converter, it sends the current alarm or fault messages and accepts simultaneously the sent value consisting of AS-i.P0 ... AS-i.P3 as new scaling factor.

- AS-i.P0 Scaling factor bit 0
- AS-i.P1 Scaling factor bit 1
- AS-i.P2 Scaling factor bit 2
- AS-i.P3 Scaling factor bit 3

Table 7-4 Scaling of the speed setpoint via AS-i.P0 ... AS-i.P3 (Page 210).

7.3 Dual Device mode

In Dual Device mode, eight bits are available for the communication between the AS-i master and the converter. The eight bits are used to transfer process data. In parallel, the control can start a diagnostic request via AS-i.P0.

The following default settings are possible:

- Default setting 31: Dual Device mode with fixed setpoints
- Default setting 34: Dual Device mode with setpoint via AS-i field bus

Default setting 31: Dual Device mode with fixed setpoints

The control accesses the two devices of the converter each via four bits.

Via device 2, in accordance with profile 7.A.E, the control specifies the speed setpoint via the motor control bits (AS-i.DO0 ... AS-i.DO2).

Via device 1, the control sends data in cyclical or acyclical mode, in accordance with profile 7.A.5.

The control requires one bit per device in order to specify the device.

Default setting 31, device 2 with profile 7.A.E: Control -> converter

- AS-i.DO0 -> p1020.0 = 2093.0 Fixed speed bit 0
- AS-i.DO1 -> p1021.0 = 2093.1 Fixed speed bit 1
- AS-i.DO2 -> p1022.0 = 2093.2 Fixed speed bit 2
- AS-i.DO3 -> Select device A or device B, interconnected internally

Table 7-5 Fixed speeds via the motor control bits and response in the converter (Page 211).

If the control specifies the speed setpoint, the converter replies:

Default setting 31, device 2 with profile 7.A.E: Converter -> control

- p2080[0] = 53.13 PLC ready to switch on -> AS-i.DI0
- p2080[1] = 899.11 Pulses enabled -> AS-i.DI1
- p2080[2] = 722.0 State DI0 -> AS-i.DI2
- p2080[3] = 722.1 State DI1 -> AS-i.DI3

If the control sends a diagnostic request via AS-i.P0, the converter replies with the currently pending fault or alarm messages.

Table 7-6 Alarm and fault messages via RP0 ... RP3 from the converter to the AS-i master (Page 211).

Default setting 31, device 1 with profile 7.A.5: Control -> converter

- AS-i.DO0 -> Time signal for the CTT2 transfer from the AS-i master
- AS-i.DO1 -> Data bit for the CTT2 transfer, four bytes cyclically or acyclically via PIV. The reading and writing of parameters is possible via the PIV. Because data is transferred bit-by-bit, the read and write process is very slow.
- AS-i.DO2 -> p0881 = 2093.4 Override quick stop
- AS-i.DO3 -> Select device A or device B, interconnected internally

Default setting 31, device 1 with profile 7.A.5: Converter -> control

- p2080[4] = 722.2 State DI2 -> AS-i.DI0
- p2080[5] = 722.3 State DI3 -> AS-i.DI1
- Serial data transfer CTT2, four bytes cyclically or acyclically via PIV. The reading and writing of parameters is possible via the PIV. Because data is transferred bit-by-bit, the read and write process is very slow. -> AS-i.DI2
- Time signal for the CTT2 transfer to the AS-i master -> AS-i.DI3

Default setting 34: Dual Device mode with setpoint via AS-i field bus

The control accesses the two devices of the converter each via four bits.

Via device 2, in accordance with profile 7.A.E, the control specifies the commands listed below (AS-i.DO0 ... AS-i.DO2).

Via device 1, the control sends the command for quick stop and the data in cyclical or acyclical mode.

The control requires one bit per device in order to specify the device.

Default setting 34, device 2 with profile 7.A.E: Control -> converter

- AS-i.DO0 -> ON / OFF 1
- AS-i.DO1 -> OFF 2
- AS-i.DO2 -> Acknowledge fault
- AS-i.DO3 -> Select device A or device B, interconnected internally

If the control specifies the speed setpoint, the converter replies:

Default setting 34, device 2 with profile 7.A.E: Converter -> control

- p2080[0] = 53.13 PLC ready to switch on -> AS-i.DI0
- p2080[1] = 899.11 Pulses enabled -> AS-i.DI1
- p2080[2] = 722.0 State DI0 -> AS-i.DI2
- p2080[3] = 722.1 State DI1 -> AS-i.DI3

If the control sends a diagnostic request via AS-i.P0, the converter replies with the currently pending fault or alarm messages.

Table 7-6 Alarm and fault messages via RP0 ... RP3 from the converter to the AS-i master (Page 211).

Default setting 34, device 1 with profile 7.A.5: Control -> converter

- AS-i.DO0 -> Time signal for the CTT2 transfer from the AS-i master
- AS-i.DO1 -> Data bit for the CTT2 transfer, four bytes cyclically or acyclically via PIV. The reading and writing of parameters is possible via the PIV. Because data is transferred bit-by-bit, the read and write process is very slow.
- AS-i.DO2 -> p0881 = 2093.4 Override quick stop
- AS-i.DO3 -> Select device A or device B, interconnected internally

Default setting 34, device 1 with profile 7.A.5: Converter -> control

- p2080[4] = 722.2 State DI2 -> AS-i.DI0
- p2080[5] = 722.3 State DI3 -> AS-i.DI1
- Serial data transfer CTT2, four bytes cyclically or acyclically via PIV. The reading and writing of parameters is possible via the PIV. Because data is transferred bit-by-bit, the read and write process is very slow. -> AS-i.DI2
- Time signal for the CTT2 transfer to the AS-i master -> AS-i.DI3

7.4 Assignment tables

Fixed speeds - Single Device

Table 7-3 Fixed speeds via the motor control bits

AS-i.DO3	AS-i.DO2	AS-i.DO1	AS-i.DO0	Response in the converter
0	0	0	0	OFF1
0	0	0	1	On + fixed speed 1 (factory setting: 1500 rpm)
0	0	1	0	On + fixed speed 2 (factory setting: -1500 rpm)
0	0	1	1	On + fixed speed 3 (factory setting: 300 rpm)
0	1	0	0	On + fixed speed 4 (factory setting: 450 rpm)
0	1	0	1	On + fixed speed 5 (factory setting: 600 rpm)
0	1	1	0	On + fixed speed 6 (factory setting: 750 rpm)
0	1	1	1	On + fixed speed 7 (factory setting: 900 rpm)
1	0	0	0	On + fixed speed 8 (factory setting: 1050 rpm)
1	0	0	1	On + fixed speed 9 (factory setting: 1200 rpm)
1	0	1	0	On + fixed speed 10 (factory setting: 1350 rpm)
1	0	1	1	On + fixed speed 11 (factory setting: 1500 rpm)
1	1	0	0	On + fixed speed 12 (factory setting: 1650 rpm)
1	1	0	1	On + fixed speed 13 (factory setting: 1800 rpm)
1	1	1	0	On + fixed speed 14 (factory setting: 1950 rpm)
1	1	1	1	Acknowledge fault or OFF2

Modified addressing - scaling factors

Table 7-4 Scaling of the speed setpoint via AS-i.P0 ... AS-i.P3

AS-i.P3	AS-i.P2	AS-i.P1	AS-i.P0	Scaling factor	Frequency (Hz)
1	1	1	1	1	50
1	1	1	0	0.9	45
1	1	0	1	0.8	40
1	1	0	0	0.7	35
1	0	1	1	0.6	30

AS-i.P3	AS-i.P2	AS-i.P1	AS-i.P0	Scaling factor	Frequency (Hz)
1	0	1	0	0.5	25
1	0	0	1	0.45	22.5
1	0	0	0	0.4	20
0	1	1	1	0.35	17.5
0	1	1	0	0.3	15
0	1	0	1	0.25	12.5
0	1	0	0	0.2	10
0	0	1	1	0.15	7.5
0	0	1	0	0.1	5
0	0	0	1	0.07	3.5
0	0	0	0	0.05	2.5

Fixed speeds - Dual Device

Table 7-5 Fixed speeds via the motor control bits and response in the converter

AS-i.DO2	AS-i.DO1	AS-i.DO0	Response in the converter
0	0	0	OFF1
0	0	1	On + fixed speed 1 (factory setting: 1500 rpm)
0	1	0	On + fixed speed 2 (factory setting: -1500 rpm)
0	1	1	On + fixed speed 3 (factory setting: 300 rpm)
1	0	0	On + fixed speed 4 (factory setting: 450 rpm)
1	0	1	On + fixed speed 5 (factory setting: 600 rpm)
1	1	0	On + fixed speed 6 (factory setting: 750 rpm)
1	1	1	Acknowledge fault or OFF2

Alarm and fault messages

Table 7-6 Alarm and fault messages via RP0 ... RP3 from the converter to the AS-i master

RP3	RP2	RP1	RP0	AS-i.P0 = 0 -> alarm messages	AS-i.P0 = 1 -> faults
0	0	0	0	No alarm	No fault
0	0	0	1	not used	Temperature rise F30004, F30012, F30013, F30024, F30025, F30036
0	0	1	0	not used	not used
0	0	1	1	No load (A07929)	not used
0	1	0	0	Temperature rise (A05000, A05004, A05006, A07012, A07015)	I^2t overload (F30005, F07936)
0	1	0	1	Overvoltage (A07400, A30502)	Equipment malfunction (F01000 ... F01257)
0	1	1	0	Keyswitch off (A03560)[1)]	not used
0	1	1	1	Undervoltage (A07402, A30016)	Motor-PTC sensor malfunction (F07011, F07016)

RP3	RP2	RP1	RP0	AS-i.P0 = 0 -> alarm messages	AS-i.P0 = 1 -> faults
1	0	0	0	I^2t overload (A07805)	Overvoltage (F30002, F30011)
1	0	0	1	not used	not used
1	0	1	0	LOCAL Mode active (A03561)[1)]	not used
1	0	1	1	not used	Undervoltage (F30003, F07802)
1	1	0	0	not used	Short-circuit at the output (F30001, F30017, F30021, F07801, F07807, F07900)
1	1	0	1	Motor phase loss (A30015)[2)]	Motor phase loss (F30015, F07902)
1	1	1	0	not used	Safety fault (F016xx)
1	1	1	1	Other alarms	Other faults

1) Only for G115D

2) Only if F30015 was reparameterized as an alarm

7.5 Cyclic and acyclic communication via CTT2

Via CTT2 (Combined Transaction Type 2), both cyclical and acyclical communication is performed via AS-i. Because only one channel is available (AS-i.DO1 master -> device or AS-i.DI3 device -> master), a concurrent cyclical and acyclical data exchange is not possible.

The communication type (cyclical or acyclical) is always coded in the first byte in accordance with the following table.

Table 7-7 CTT2 commands

Code (hex)	Explanation/meaning	Followed by
Cyclic communication		
Access to analog values via DS140 ... DS147. See CP 343–2 / CP 343–2 P AS–Interface master (http://support.automation.siemens.com/WW/view/en/5581657), Chapter 4		4 bytes: PWE1, PWE2
		4 bytes: PWE1, PWE2
Acyclic communication - standard		
10 hex	Read request: Master -> device	2 bytes: Index, length
50 hex	Read request OK: Device -> master	Index, data
90 hex	Read request failed: Device -> master	1 byte: Standard error code (3 hex)
11 hex	Write request: Master -> device	Index, length, data
51 hex	Write request OK: Device -> master	
91 hex	Write request failed: Device -> master	1 byte: Standard error code (3 hex)
Acyclic communication - manufacturer-specific		
12 hex	Read request: Master -> device	Index, length
52 hex	Read request OK: Device -> master	Data
92 hex	Read request failed: Device -> master	Fault object
13 hex	Write request: Master -> device	Index, length, data
53 hex	Write request OK: Device -> master	

Code (hex)	Explanation/meaning	Followed by
93 hex	Write request failed: Device -> master	Fault object
1D hex	Exchange request: Master -> device	Index, read length, write length, write data
5D hex	Exchange request OK: Device -> master	PKE, index, n-2 data
9D hex	Exchange request faulty: Device -> master	Fault object

If an acyclical request cannot be executed by the converter, it replies with one of the following error messages.

Error message	Meaning
0	No fault
1	Invalid index
2	Incorrect length
3	Request not implemented
4	Busy (the request could not be processed completely within the time window, retry later)
5	Last acyclical request was not confirmed
6	Invalid subindex
7	"Selective read request" command missing

7.5.1 Cyclic communication

Converter -> master

The converter cyclically transfers the data from p2051[1] and p2051[2] in four bytes to the master. You can process these four bytes in the control as for analog data. Refer to the documentation for the AS-i master for detailed information about access to analog data.

If you selected default setting 31 or 34 during the commissioning, the two indexes are interconnected as follows:

- p2051[1] = 63: Smoothed actual speed value
- p2051[2] = 27: Absolute smoothed actual current value

The values for transfer are normalized in accordance with the Profidrive N2 data type. Using p2051[1] and p2051[2] you can interconnect any other or connector parameters and transfer to the control.

Master -> converter

The master transfers the data in the "Combined Transaction Type 2" (CTT2) to the converter and writes it to r2050[1] and r2050[2].

To process these values in the converter, you must appropriately interconnect r2050[1] and r2050[2] in the converter. This means, when the control sends the speed setpoint, you must

interconnect parameter p1070 (source for the main setpoint) with r2050 as follows:
p1070[0] = 2050[1]

Note

Internal interconnection with default setting 34

If, when commissioning, you select "Default setting 34", then the main setpoint is internally interconnected with r2050[1].

Once a setpoint has been transferred completely, the setpoint present in the control will be transferred as next setpoint. Any setpoint changes made during the transfer are not considered.

7.5.2 Acyclic communication - standard

This type of acyclical communication supports the ID read request and the diagnostic read request. All other requests receive the "request not implemented" message response.

- ID request:
  - Master -> device | 10 hex | 00 hex | nn hex |
  - Device -> master | 50 hex | 00 hex | Manufacturer's ID | Product ID | BB hex |
- Diagnostic request:
  - Master -> device | 10 hex | 01 hex | nn hex |
  - Device -> master no error | 50 hex | 01 hex | 00 hex |;
    Device -> master general error | 50 hex | 01 hex | 99 hex |

The following repsonse is issued for all other write or read requests:

- Read requests | 90 hex | 03 hex |
- Write requests | 91 hex | 03 hex |

7.5.3 Acyclic communication - manufacturer-specific

The manufacturer-specific acyclical communication is performed via data record 47 in PIV format. The PIV format structure is identical with that for the USS parameter channel.

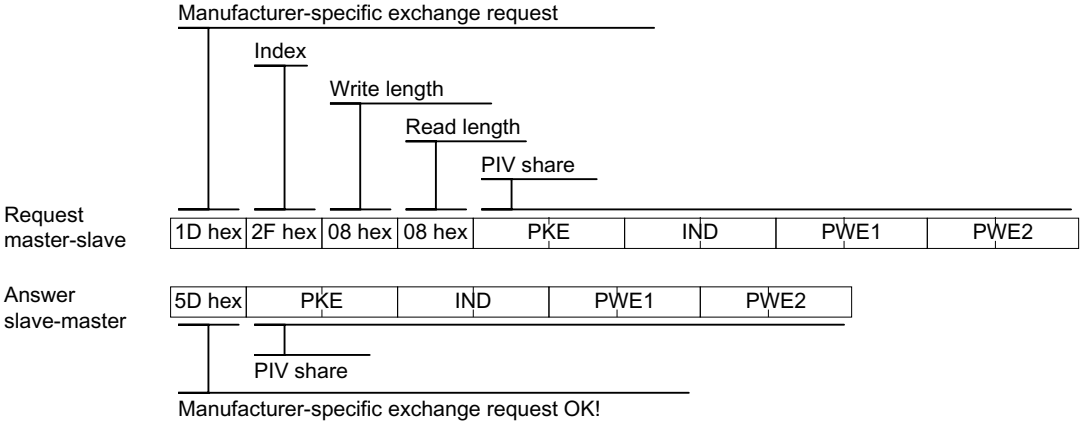
USS parameter channel (Page 108).

To reduce the transfer volume, there is not only the "normal" "data exchange" PIV mechanism, but also the "Read data" and "Write data" commands.

- Data exchange:
  - Control -> converter request
  - Converter -> control response
- Read data:
  The converter sends a read command, and the data of the last exchange request or write request is transferred from the converter to the control.
- Write data
  Write OK: -> 53 hex.

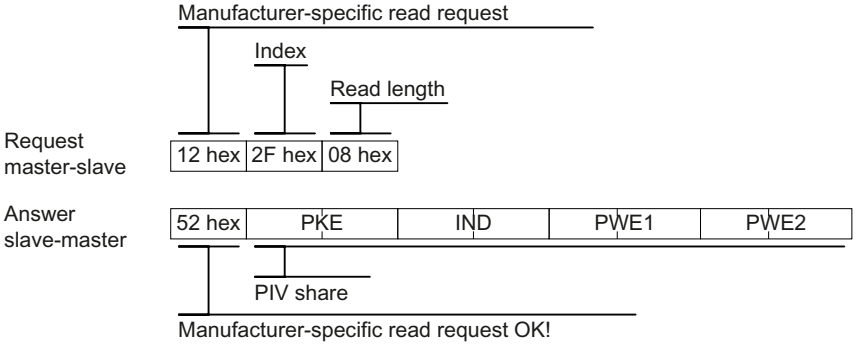
Because the PIV transfer protocol specifies the transfer direction independently, all parameters can be transferred as data exchange request/response. Requests for reading and writing data are included primarily to reduce the transferred data volume for the repeated reading or writing of parameters.

Data exchange

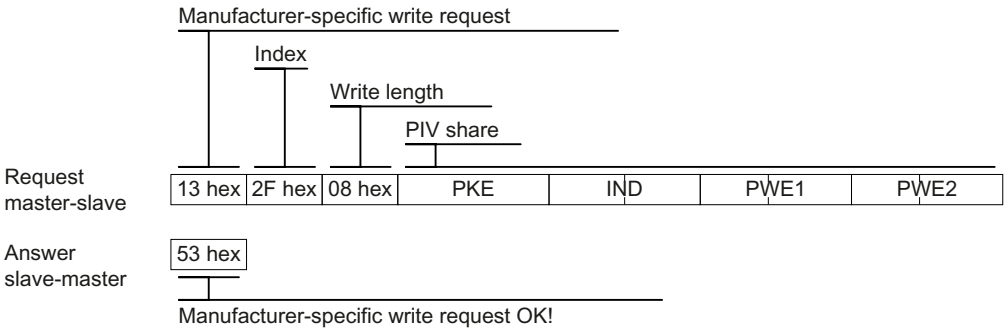


Reading data

The data for the last write or exchange request is read



Writing data



In the event of a fault, the converter sends the following telegram as response to the master:

93 hex	00 hex	PWE1

.

Value for PWE: Fault table from USS parameter channel (Page 108).

Appendix

A

A.1 Application examples for communication with STEP7

Application examples for communication with STEP 7 can be found in the following manual:
"Fieldbuses" function manual, edition 09/2017 (https://support.industry.siemens.com/cs/ww/en/view/109751350)

A.2 Manuals and technical support

A.2.1 Overview of the manuals

You can find manuals here with additional information for downloading

- CU250S-2 operating instructions (https://support.industry.siemens.com/cs/ww/en/view/109782994)
  Installing, commissioning and maintaining the converter. Advanced commissioning
- CU240B/E-2 operating instructions (https://support.industry.siemens.com/cs/ww/en/view/109782865)
  Installing, commissioning and maintaining the converter. Advanced commissioning
- CU230P-2 operating instructions (https://support.industry.siemens.com/cs/ww/en/view/109782866)
  Installing, commissioning and maintaining the converter. Advanced commissioning
- SINAMICS G120C operating instructions. (https://support.industry.siemens.com/cs/ww/en/view/109482993)
  Installing, commissioning and maintaining the converter. Advanced commissioning
- SINAMICS G115D Operating Instructions
  Installing, commissioning and maintaining the converter. Advanced commissioning
- Operating instructions SINAMICS G120D with CU240D-2 (https://support.industry.siemens.com/cs/ww/en/view/109477366)
  Installing, commissioning and maintaining the converter. Advanced commissioning
- Operating instructions SINAMICS G120D with CU250D-2 (https://support.industry.siemens.com/cs/ww/en/view/109477365)
  Installing, commissioning and maintaining the converter. Advanced commissioning

- "Safety Integrated" function manual (https://support.industry.siemens.com/cs/ww/en/view/109751320)
  Configuring PROFIsafe.
  Installing, commissioning and operating failsafe functions of the converter.
- "Fieldbus" function manual (https://support.industry.siemens.com/cs/ww/en/view/109751350)
  Configuring fieldbuses (this manual)
- "Basic positioner" function manual (https://support.industry.siemens.com/cs/ww/en/view/109477922)
  Commissioning the basic positioner
- CU250S-2 List Manual (https://support.industry.siemens.com/cs/ww/en/view/109782287)
  List of all parameters, alarms and faults, graphic function diagrams.
- CU240B/E-2 List Manual (https://support.industry.siemens.com/cs/ww/en/view/109782301)
  List of all parameters, alarms and faults, graphic function diagrams.
- CU230P-2 List Manual (https://support.industry.siemens.com/cs/ww/en/view/109782303)
  List of all parameters, alarms and faults, graphic function diagrams.
- List manual SINAMICS G120D (https://support.industry.siemens.com/cs/ww/en/view/109477255)
  List of all parameters, alarms and faults, graphic function diagrams.
- SINAMICS G120C List Manual (https://support.industry.siemens.com/cs/ww/en/view/109482977)
  List of all parameters, alarms and faults, graphic function diagrams.
- AS-Interface system manual (https://support.industry.siemens.com/cs/ww/en/view/26250840)

Finding the most recent edition of a manual

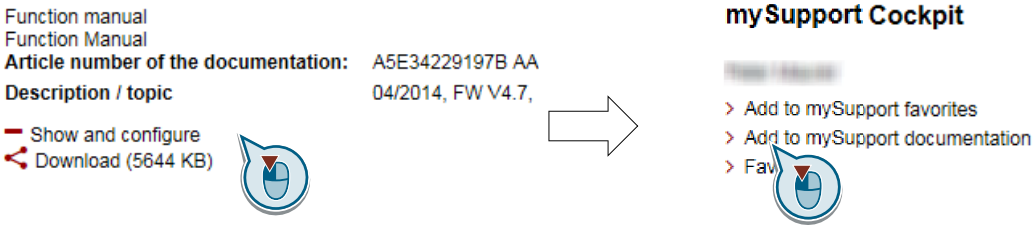
If there a multiple editions of a manual, select the latest edition:

Configuring a manual

Further information about the configurability of manuals is available in the Internet:

MyDocumentationManager (https://www.industry.siemens.com/topics/global/en/planning-efficiency/documentation/Pages/default.aspx).

Select "Display and configure" and add the manual to your "mySupport-documentation":



Not all manuals can be configured.

The configured manual can be exported in RTF, PDF or XML format.

A.2.2 Configuring support

Catalog

Ordering data and technical information for the converters SINAMICS G.

Catalogs for download or online catalog (Industry Mall):

Everything about SINAMICS G120 (www.siemens.en/sinamics-g120)

SIZER

The configuration tool for SINAMICS, MICROMASTER and DYNAVERT T drives, motor starters, as well as SINUMERIK, SIMOTION controllers and SIMATIC technology

SIZER on DVD:

Article number: 6SL3070-0AA00-0AG0

Download SIZER (https://support.industry.siemens.com/cs/ww/en/view/54992004)

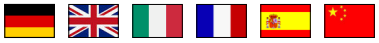
EMC (electromagnetic compatibility) technical overview

Standards and guidelines, EMC-compliant control cabinet design

EMC overview (https://support.industry.siemens.com/cs/ww/en/view/103704610)

EMC Guidelines configuration manual

EMC-compliant control cabinet design, potential equalization and cable routing



EMC installation guideline (http://support.automation.siemens.com/WW/view/en/60612658)

See also

Safety Integrated for novices (https://support.industry.siemens.com/cs/ww/en/view/80561520)

A.2.3 Product Support

Overview

You can find additional information about the product on the Internet:

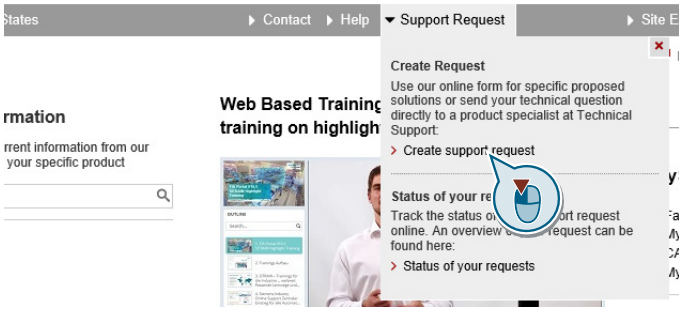
Product support (https://support.industry.siemens.com/cs/ww/en/)

This URL provides the following:

- Up-to-date product information (product announcements)
- FAQs
- Downloads
- The Newsletter contains the latest information on the products you use.
- The Knowledge Manager (Intelligent Search) helps you find the documents you need.
- Users and specialists from around the world share their experience and knowledge in the Forum.

- You can find your local representative for Automation & Drives via our contact database under "Contact & Partner".
- Information about local service, repair, spare parts and much more can be found under "Services".

If you have any technical questions, use the online form in the "Support Request" menu:



Index

O

P

Q

R

S

T

U

Z

Further information

SINAMICS converters:
www.siemens.com/sinamics

Safety Integrated
www.siemens.com/safety-integrated

PROFINET
www.siemens.com/profinet

Siemens AG
Digital Factory
Motion Control
Postfach 3180
91050 ERLANGEN
Germany

Scan the QR code for additional information about SINAMICS G120.