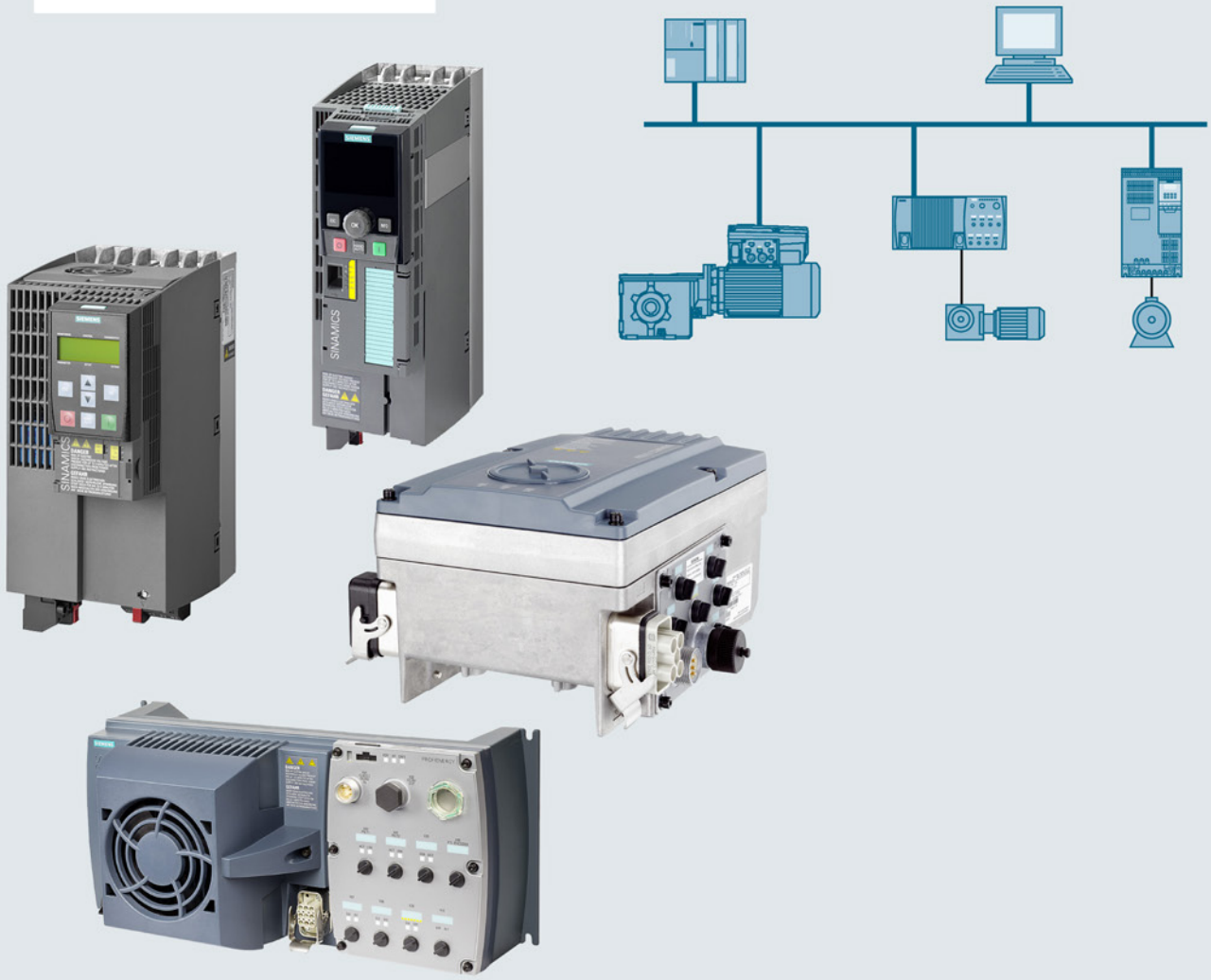
SIEMENS



SINAMICS

SINAMICS G110M, G120, G120P, G120C, G120D inverters

Fieldbus systems: PROFIBUS, PROFINET, EtherNet/IP, CANopen, USS, Bacnet, Modbus

Function Manual

Edition 04/2014

Answers for industry

SIEMENS

SINAMICS

SINAMICS G120, G120P, G120C, G120D, G110M Fieldbuses

Function Manual

Edition 04/2014, Firmware V4.7

04/2014, FW V4.7

A5E34229197B AA

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

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
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: A5E34229197B AA
℗ 07/2014 Subject to change

Table of contents

Fundamental safety instructions

1

1.1 General safety instructions

⚠WARNING

Risk of death if the safety instructions and remaining risks are not carefully observed

If the safety instructions and residual risks are not observed in the associated hardware documentation, 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

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

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

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

1.2 Industrial security

Note

Industrial security

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

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

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit Hotspot-Text (http://support.automation.siemens.com).

⚠WARNING

Danger as a result of unsafe operating states resulting from software manipulation

Software manipulation (e.g. by viruses, Trojan horses, malware, worms) can cause unsafe operating states to develop in your installation which can result in death, severe injuries and/or material damage.

- Keep the software up to date.

  You will find relevant information and newsletters at this address (http://support.automation.siemens.com).
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.

  You will find further information at this address (http://www.siemens.com/industrialsecurity).
- Make sure that you include all installed products into the holistic industrial security concept.

Introduction 2

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

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

If, in your plant or system, communication with the control system should continue to function even when the line voltage is switched off, then you must externally supply the inverter/Control Unit with 24 V DC. To do this, use terminals 31 and 32 – or connector X01. You can find additional details in the operating instructions for the inverter or the Control Unit.

What is the meaning of the symbols in the manual?

An operating instruction starts here.

This concludes the operating instruction.

2.1 Ethernet and PROFINET protocols that are used

The inverter 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
PTCP Precision Transparent Clock Protocol	Not relevant	(2) Ethernet II and IEEE 802.1Q and Ethertype 0x8892 (PROFINET)	**PROFINET send clock and time synchronisation, 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 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

3.1.1 Cyclic communication

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

The telegrams that are possible for your particular inverter are listed for selection in STARTER or on operator panel.

How you set the telegram is described in Section Additional manuals for your inverter (Page 182).

Communication telegrams if "basic positioner" has been configured

The inverter 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

These telegrams are described in the "Basic positioner and technology" Function Manual. See also the section Additional manuals for your inverter (Page 182).

Communication telegrams for speed control

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

PKW	PZD01	PZD02	PZD03	PZD04	PZD05	PZD06	PZD07	PZD08	PZD09	PZD10	PZD11	PZD12	PZD13	PZD14

Telegram 1, closed-loop speed control

STW1	NSOLL_A
ZSW1	NIST_A

Telegram 2, closed-loop speed control

STW1	NSOLL_B	STW2
ZSW1	NIST_B	ZSW2

Telegram 3, closed-loop speed control, 1 position encoder

STW1	NSOLL_B	STW2	G1_STW		
ZSW1	NIST_B	ZSW2	G1_ZSW	G1_XIST1	G1_XIST2

Telegram 4, closed-loop speed control, 2 position encoders

STW1	NSOLL_B	STW2	G1_STW	G2_STW				
ZSW1	NIST_B	ZSW2	G1_ZSW	G1_XIST1	G1_XIST2	G2_ZSW	G2_XIST1	G2_XIST2

Telegram 20, closed-loop speed control, VIK/NAMUR

STW1	NSOLL_A				
ZSW1	NIST_A GLATT	IAIST_ GLATT	MIST_ GLATT	PIST_ GLATT	MELD_ NAMUR

Telegram 350, closed-loop speed control

STW1	NSOLL_A	M_LIM	STW3
ZSW1	NIST_A GLATT	IAIST_ GLATT	ZSW3

Telegram 352, closed-loop speed control for PCS7

STW1	NSOLL_A	Process data for PCS7				
ZSW1	NIST_A GLATT	IAIST_ GLATT	MIST_ GLATT	WARN_ CODE	FAULT_ CODE	

Telegram 353, closed-loop speed control - with PKW area to read and write parameters

PKW	STW1	NSOLL_A
	ZSW1	NIST_A GLATT

Telegram 354, closed-loop speed control for PCS7 - with PKW area to read and write parameters

PKW	STW1	NSOLL_A	Process data for PCS7			
	ZSW1	NIST_A GLATT	IAIST_ GLATT	MIST_ GLATT	WARN_ CODE	FAULT_ CODE

Telegram 999, free interconnection

STW1	Telegram length for receive data can be configured
ZSW1	Telegram length for transmit data can be configured

Abbreviation	Explanation	Abbreviation	Explanation
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 number
NSOLL_B	Speed setpoint 32 bit	WARN_CODE	Alarm number
NIST_A	Speed actual value 16 bit	MELD_NAMUR	Control word according to the VIK-NAMUR definition
NIST_B	Speed actual 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

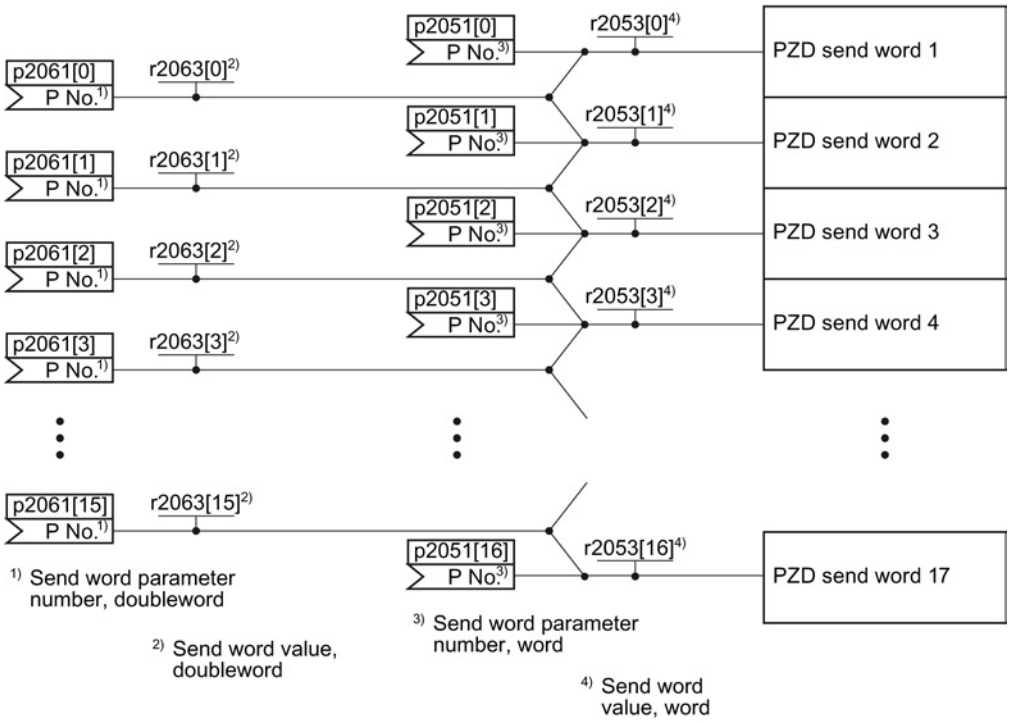


Figure 3-1 Interconnection of the send words

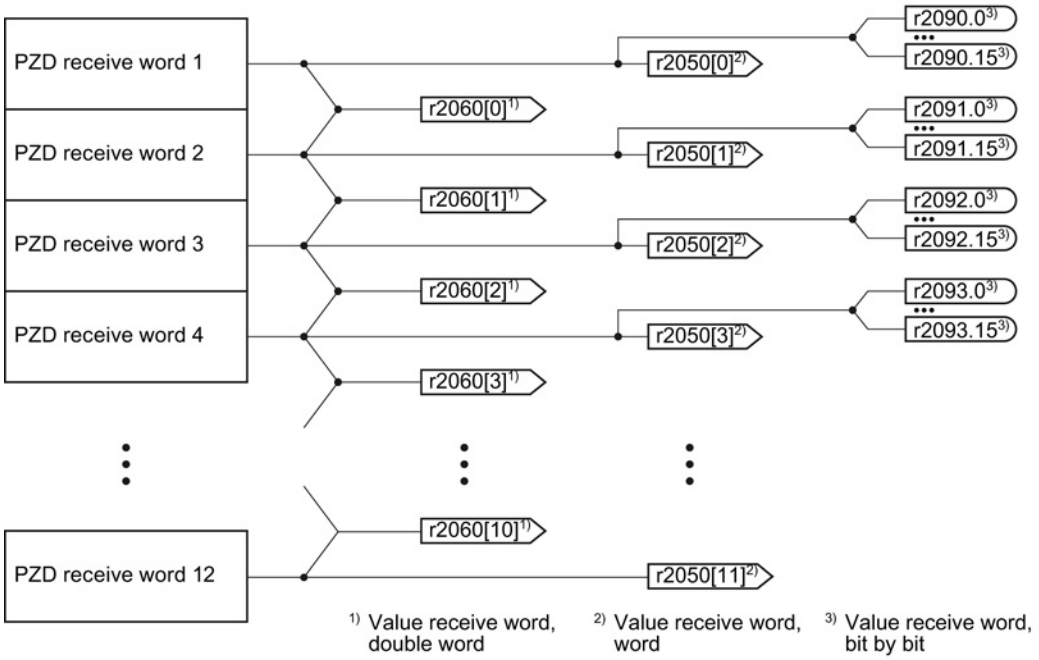


Figure 3-2 Interconnection of the receive words

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.

3.1.1.1 Assigning control and status words

Assigning control and status words

Assigning control and status of words is specified in part by the definitions in the PROFIdrive profile, Version 4.1 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. You can find details about this in Section Extend telegrams and change signal interconnection (Page 26).

Control word 1 (STW1)

The control word 1 is pre-assigned as follows.

- Telegram 20 (VIK/NAMUR):
  - Bit 0 … 11 corresponds to PROFIdrive profile,
  - Bit 12… 15 manufacturer-specific
- Other telegrams:
  - Bit 0 … 10 corresponds to PROFIdrive profile,
  - Bit 11… 15 manufacturer-specific

Bit	Significance		Explanation	Signal interconnection in the inverter
	Telegram 20	All other telegrams		
0	0 = OFF1		The motor brakes with the ramp-down time p1121 of the ramp-function generator. The inverter switches off the motor at standstill.	p0840[0] = r2090.0
	0 → 1 = ON		The inverter goes into the "ready" state. If, in addition bit 3 = 1, then the inverter 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 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 inverter 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 inverter 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 inverter switches to"closing lockout" state.	p2103[0] = r2090.7
8, 9	Reserved			
10	0 = No control via PLC		Inverter ignores the process data from the fieldbus.	p0854[0] = r2090.10
	1 = Control via PLC		Control via fieldbus, inverter accepts the process data from the fieldbus.	
11	0 = Direction reversal		Invert setpoint in the inverter.	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)

The status word 1 is pre-assigned as follows.

- Bit 0 … 10 corresponds to PROFIdrive profile
- Bit 11… 15 manufacturer-specific

Bit	Significance		Comments	Signal interconnection in the inverter
	Telegram 20	All other telegrams		
0	1 = Ready to start		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 inverter 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 inverter 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 = Closing lockout 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 inverter 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 inverter actual value > 0	p2080[14] = r2197.3
	0 = Motor rotates counterclockwise		Internal inverter actual value < 0	
15	1 = CDS display	0 = Alarm, inverter 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.

Control word 2 (STW2)

The control word 2 is pre-assigned as follows.

- Bit 0… 11 manufacturer-specific
- Bit 12 … 15 corresponds to the PROFIdrive profile

Bit	Value	Meaning Telegrams 2, 3 and 4	Explanation	Signal interconnection in the inverter [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	DC braking enable	--	p1230[0] = r2093.9
10	–	Not used		
11	1	1 = Enable droop	Enable or inhibit speed controller droop.	p1492[0] = r2093.11
12	1	Torque control active	Changes over the control mode for vector control.	p1501[0] = r2093.12
	0	Closed-loop speed control active		
13	1	No external fault	--	p2106[0] = r2093.13
	0	External fault is active (F07860)		
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 inverter sets all interconnections p1020, … to "0". Exception: p2106 = 1.

Status word 2 (ZSW2)

Status word 2

- Bit 0… 11 manufacturer-specific
- Bit 12 … 15 corresponds to the PROFIdrive profile

Bit	Value	Meaning	Description	Signal interconnection in the inverter
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	Ramping completed	Ramp-function generator is not active.	
10	1	Technology controller output at lower limit	Technology controller output ≦ p2292	
11	1	Technology controller output at upper limit	Technology controller output > p2291	
12		Not used		
13		Not used		
14		Not used		
15		Not used		

Control word 3 (STW3)

The control word 3 is pre-assigned as follows.

- Bit 0… 15 manufacturer-specific

Bit	Value	Significance Telegram 350	Explanation	Signal interconnection in the inverter [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	DC braking enable	--	p1230[0] = r2093.9
10	–	Not used		
11	1	1 = Enable droop	Enable or inhibit speed controller droop.	p1492[0] = r2093.11
12	1	Torque control active	Changes over the control mode for vector control.	p1501[0] = r2093.12
	0	Closed-loop speed control active		
13	1	No external fault	--	p2106[0] = r2093.13
	0	External fault is active (F07860)		
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 inverter sets all interconnections p1020, … to "0". Exception: p2106 = 1.

Status word 3 (ZSW3)

The status word 3 is pre-assigned as follows.

- Bit 0… 15 manufacturer-specific

Bit	Value	Significance	Description	Signal interconnection in the inverter
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	Ramping completed	Ramp-function generator is not active.	
10	1	Technology controller output at lower limit	Technology controller output ≦ p2292	
11	1	Technology controller output at upper limit	Technology controller output > p2291	
12		Not used		
13		Not used		
14		Not used		
15		Not used		

3.1.1.2 Extend telegrams and change signal interconnection

When you have selected a telegram, the inverter interconnects the corresponding signals with the fieldbus interface. Generally, these interconnections are protected so that they cannot be changed. With the appropriate inverter settings, these interconnections can be changed.

Extend telegram

Every telegram can be extended, by "attaching" additional signals.

Procedure

Proceed as follows to extend a telegram:

1. Using STARTER or an operator panel, set parameter p0922 = 999.
2. Set parameter p2079 to the appropriate value of the corresponding telegram.
3. Interconnect additional PZD send words and PZD receive words with signals of your choice via parameters r2050 and p2051.

You have extended the telegram.

Parameter	Description	
p0922	**PROFIdrive telegram selection**	
	999:	Free telegram (message frame) configuration
p2079	**PROFIdrive PZD telegram selection extended**	
	The following values apply if you have still not enabled the "Basic positioner" function in the inverter:	
	1: 2: 3: 4: 20: 350: 352: 353: 354:	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
	The following values apply if you have enabled the "Basic positioner" function in the inverter:	
	7: 9: 110: 111:	Standard telegram 7, PZD-2/2 Standard telegram 9, PZD-10/5 SIEMENS telegram 110, PZD-12/7 SIEMENS telegram 111, PZD-12/12
r2050[0…11]	**PROFIdrive PZD receive word** Connector output to interconnect the PZD (setpoints) in the word format received from the PROFIdrive controller.	
p2051[0…16]	**PROFIdrive PZD send word** Selection of the PZD (actual values) in the word format to be sent to the PROFIdrive controller.	

Selection of the PZD (actual values) in the word format to be sent to the PROFIdrive controller. For further information refer to the function block diagrams 2468 and 2470 of the List Manual.

Freely selecting the signal interconnection of the telegram

The signals in the telegram can be freely interconnected.

Procedure

Proceed as follows to change the signal interconnection of a telegram:

1. Using STARTER or an operator panel, set parameter p0922 = 999.
2. Using STARTER or an operator panel, set parameter p2079 = 999.
3. Interconnect additional PZD send words and PZD receive words with signals of your choice via parameters r2050 and p2051.

You have freely interconnected the signals transferred in the telegram.

Parameter	Description
p0922	**PROFIdrive telegram selection**
	999: Free telegram (message frame) configuration
p2079	**PROFIdrive PZD telegram selection extended**
	999: Free telegram (message frame) configuration
r2050[0…11]	**PROFIdrive PZD receive word** Connector output to interconnect the PZD (setpoints) in the word format received from the PROFIdrive controller.
p2051[0…16]	**PROFIdrive PZD send word** Selection of the PZD (actual values) in the word format to be sent to the PROFIdrive controller.

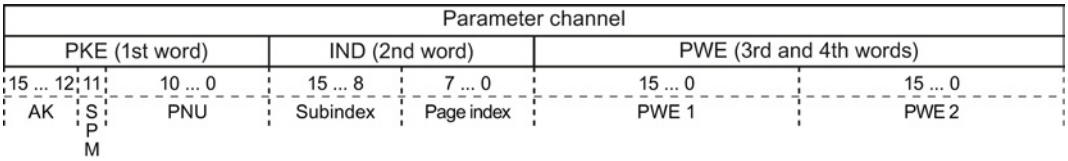
For further information refer to the function block diagrams 2468 and 2470 of the List Manual.

3.1.1.3 Data structure of the parameter channel

Structure of the parameter channel

The parameter channel consists of four words. 1. and 2nd word transfer the parameter number and index as well as the type of job (read or write) The 3rd and 4th word contains the parameter contents. The parameter contents can be 16-bit values (such as baud rate) or 32-bit values (e.g. CO parameters).

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



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

Request and response IDs

Bits 12 to 15 of the 1st word of the parameter channel contain the request and response identifier.

Table 3- 1 Request identifiers, control → inverter

Request identifier	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 3 ≡ 8.
We recommend that you use identifiers 6, 7, and 8.

Table 3- 2 Response identifiers, inverter → control

Response identifier	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	Inverter cannot process the request. In the most significant word of the parameter channel, the inverter 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- 3 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 inverter)
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 → inverter")
6B hex	**No change access for a controller that is enabled**. (operating status of the inverter prevents a parameter change)
86 hex	**Write access only for commissioning (p0010 = 15)** (operating status of the inverter prevents a parameter change)
87 hex	**Know-how protection active, access locked**
C8 hex	**Change request below the currently valid limit** (change request to a value that lies within the "absolute" limits, but is however below the currently valid lower limit)
C9 hex	**Change request above the currently valid limit** (example: a parameter value is too large for the inverter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

Offset and page index of the parameter numbers

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 7 … 0).

Parameter number	Offset	Page index								
		Hex	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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
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 15 … 8).

Parameter contents

Parameter contents can be parameter values or connectors.

Table 3- 4 Parameter values in the parameter channel

PWE, 3rd word	PWE, 4th word	
Bit 15 … 0	Bit 15 … 8	Bit 7 … 0
0	0	8-bit value
0	16-bit value	
32-bit value		

Table 3- 5 Connectors in the parameter channel

PWE, 3rd word	PWE, 4th word	
Bit 15 … 0	Bit 15 … 10	Bit 9 … 0
Number of the connector	3F hex	The index or bit field number of the connector

Telegram 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 the parameter)
- **IND, bit 0 … 7 (page index): = 90 hex** (offset 6000 ≙ 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 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 3-3 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 3-4 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 (subindex): = 1 hex** (CDS1 = index1)
- **IND, bit 0 … 7 (page index): = 0 hex** (offset 0 ≙ 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 3-5 Telegram, to assign DI 2 with ON/OFF1

"Reading and writing parameters" application example

See: Reading and writing parameters via PROFIBUS (http://support.automation.siemens.com/WW/view/en/8894584).

3.1.1.4 Slave-to-slave communication

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

Example: An inverter uses the actual speed value of another inverter as its speed setpoint.

Definitions

- **Publisher:** Slave, which sends data for direct data exchange.
- **Subscriber:** Slave, 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 inverters with PROFIBUS communication.
- A maximum of 12 PZDs are permissible for each drive.
- To a publisher, a maximum of 4 links are possible.

Procedure

To configure direct data exchange, proceed as follows:

1. In the control, define:
   - Which inverters operate as publisher (sender) or subscriber (receiver)?
   - Which data or data areas do you use for direct data exchange?
2. In the inverter, define:

   How does the subscriber process the data transferred using direct data exchange?

You have configured direct data exchange.

You can find more information in: "Manuals for your inverter in the Fieldbus communications manual (Page 180)".

3.1.2 Acyclic communication

The inverter supports the following types of acyclic communication:

- For PROFIBUS:

  Acyclic communication via data set47:
  up to 240 bytes per write or read request

- For PROFINET:

  Acyclic communication via B02E hex and B02F hex

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

See also: STEP 7 program example for acyclic communication (Page 176).

Table 3- 6 Request to read parameters

Data block	Byte n	Bytes n + 1	n
Header	**Reference** *00 hex ... FF hex*	**01 hex**: Read job	0
	01 hex (ID of drive objects, at G120 always = 1)		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 ... FFFF hex*		6
	Number of the 1st index *0000 hex ... FFFF hex* (for parameters without index: 0000 hex)		8
	…		…
Address, parameter 2	…		…
…	…		…
Address, parameter m	…		…

Table 3- 7 Inverter response to a read request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a read request)	**01 hex**: Inverter has executed the read request. **81 hex**: Inverter was not able to completely execute the read request.	0
	01 hex (ID of drive objects, at G120 always = 1)	**Number of parameters (m)** (identical to the read request)	2
Values, parameter 1	**Format** 02 hex: Integer8 03 hex: Integer16 04 hex: Integer32 05 hex: Unsigned8 06 hex: Unsigned16 07 hex: Unsigned32 08 hex: FloatingPoint 10 hex OctetString 13 hex TimeDifference 41 hex: Byte 42 hex: Word 43 hex: Double word 44 hex: Error	**Number of index values** or - for a negative response - **number of error values**	4
	Value of the 1st index or - for a negative response - **error value 1** You can find the error values in a table at the end of this section.		6
	…		…
Values, parameter 2	…		
…	…		
Values, parameter m	…		

Changing parameter values

See also: STEP 7 program example for acyclic communication (Page 176).

Table 3- 8 Request to change parameters

Data block	Byte n	Bytes n + 1	n
Header	**Reference** *00 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 *0001 hex ... FFFF hex*		8
	…		…
Address, parameter 2	…		
…	…		…
Address, parameter m	…		
Values, parameter 1	**Format** *02 hex:* Integer 8 *03 hex:* Integer 16 *04 hex:* Integer 32 *05 hex:* Unsigned 8 *06 hex:* Unsigned 16 *07 hex:* Unsigned 32 *08 hex:* Floating Point *10 hex* Octet String *13 hex* Time Difference *41 hex:* Byte *42 hex:* Word *43 hex:* Double word	**Number of index values** *00 hex ... EA hex*	
	Value of the 1st index		
	…		
Values, parameter 2	…		
…	…		
Values, parameter m	…		

Table 3- 9 Response, if the inverter has executed the change request

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

Table 3- 10 Response if the inverter was not able to completely execute the change request

Data block	Byte n	Bytes n + 1	n
Header	**Reference** (identical to a change request)	**82 hex**	0
	01 hex (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 or 02 hex	4
	Only for "Error" - **error value 1** You can find the error values in the table at the end of this section.		6
	Only for "Error" - **error value 2** Error value 2 is either zero, or it contains the number of the first index where the error occurred.		8
Values, parameter 2	...		
...	…		…
Values, parameter m	...		

Error values

Table 3- 11 Error value in the parameter response

Error value 1	Meaning
00 hex	**Illegal parameter number** (access to a parameter that does not exist)
01 hex	**Parameter value cannot be changed** (change request for a parameter value that cannot be changed)
02 hex	**Lower or upper value limit exceeded** (change request with a value outside the value limits)
03 hex	**Incorrect subindex** (access to a parameter index that does not exist)
04 hex	**No array** (access with a subindex to non-indexed parameters)
05 hex	**Incorrect data type** (change request with a value that does not match the data type of the parameter)
06 hex	**Setting not permitted, only resetting** (change request with a value not equal to 0 without permission)
07 hex	**Descriptive element cannot be changed** (change request to a descriptive element that cannot be changed)
09 hex	**Description data not available** (access to a description that does not exist, parameter value is available)
0B hex	**No master control** (change request but with no master control)
0F hex	**Text array does not exist** (although the parameter value is available, the request is made to a text array that does not exist)
11 hex	**Request cannot be executed due to the operating state** (access is not possible for temporary reasons that are not specified)
14 hex	**Inadmissible value** (change request with a value that is within the limits but which is illegal for other permanent reasons, i.e. a parameter with defined individual values)
15 hex	**Response too long** (the length of the actual response exceeds the maximum transfer length)
16 hex	**Illegal parameter address** (illegal or unsupported value for attribute, number of elements, parameter number, subindex or a combination of these)
17 hex	**Illegal format** (change request for an illegal or unsupported format)

Error value 1	Meaning
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 inverter rejects the change request because the motor is switched on. Please observe the "Can be changed" parameter attribute (C1, C2, U, T) in the List Manual. See also section: Additional manuals for your inverter (Page 182))
6C hex	**Unknown unit**.
6E hex	**Change request is only possible when the motor is being commissioned (p0010 = 3).**
6F hex	**Change request is only possible when the power unit is being commissioned (p0010 = 2).**
70 hex	**Change request is only possible for quick commissioning (basic commissioning) (p0010 = 1).**
71 hex	**Change request is only possible if the inverter is ready (p0010 = 0).**
72 hex	**Change request is only possible for a parameter reset (restore to factory setting) (p0010 = 30).**
73 hex	**Change request possible only during commissioning of the safety functions (p0010 = 95).**
74 hex	**Change request is only possible when a technological application/unit is being commissioned (p0010 = 5).**
75 hex	**Change request is only possible in a commissioning state (p0010 ≠ 0).**
76 hex	**Change request is not possible for internal reasons (p0010 = 29).**
77 hex	**Change request is not possible during download.**
81 hex	**Change request is not possible during download.**
82 hex	**Transfer of the control authority (master) is inhibited by BI: p0806.**
83 hex	**Desired interconnection is not possible** (the connector output does not supply a float value although the connector input requires a float value)
84 hex	**Inverter does not accept a change request** (inverter is busy with internal calculations. See parameter r3996 in the inverter List Manual. See also Section: Additional manuals for your inverter (Page 182))
85 hex	**No access methods defined**.
86 hex	**Write access only during commissioning of the data records (p0010 = 15)** (operating status of the inverter prevents a parameter change.)
87 hex	**Know-how protection active, access locked**
C8 hex	**Change request below the currently valid limit** (change request to a value that lies within the "absolute" limits, but is however below the currently valid lower limit)
C9 hex	**Change request above the currently valid limit** (example: a parameter value is too large for the inverter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

3.2 Identification & maintenance data (I&M)

I&M data

The inverter 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	Inverter-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 is p8805 = 0 is used.	p8809[0 … 53]	Values of r9781[0] and r9782[0]

When requested, the inverted transfers its I&M data to a higher-level control or to a PC/PG with installed STEP 7, STARTER 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	✓	✓

3.3 S7 communication

S7 communication allows the following:

- The inverter is controlled from a SIMATIC panel, also without control system by directly accessing the inverter via PROFIBUS or PROFINET.
- Remote maintenance by accessing the inverter with STARTER or Startdrive across network boundaries.

In the following example, we describe how you can switch the inverter on and off through a SIMATIC panel, how to specify a setpoint and be shown the actual value on the panel.

You can find a description about accessing inverters across network boundaries in the application document 25339612.

3.3.1 Direct access to a SIMATIC G120 inverter using a SIMATIC panel

Example of direct access to the inverter via a SIMATIC panel

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

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

Conditions

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

- WINCCflex 2008 SP1 or higher
- STARTER plus SSP for 4.7 or higher
- You have configured the inverter in STARTER
- Inverter and panel are connected with one another via PROFIBUS or PROFINET.
- The same baud rates are set in the inverter and in the panel.
- The bus address configured in WinCC flexible matches the bus address of the inverter.

Settings in the inverter

Procedure

To adjust the settings in the inverter, proceed as follows:

1. **Make the following settings and releases so that the inverter 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 releases for the ramp-up encoder
     p1140 = 1
     p1141 = 1
   - Set the setpoint release
     p1142 = 1
2. **Adjust the parameters for the ON/OFF1 command from the SIMATIC panel**
   - Set p0840[0] = 2094.0
     In doing so, you interconnect the ON/OFF1 command with bit 0 of the BiCo converter 2094. The signal source of this parameter is p2099.
   - Now 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 (speed fixed setpoint selection, Bit 0)
4. **Actual value and status word**

   No further settings are required by the inverter for displaying the speed list value (r0021) and the status word (r0052).

You have now made the settings in the inverter.

Settings at the SIMATIC panel

Procedure

To adjust the settings for the panel, proceed as follows:

1. **Configure the connection using WINCCflex**
   - 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 inverter, select "Only master on bus"
   - Select cyclical operation.
3. **ON/OFF1:**
   - Create a variable for the parameter p2900, which refers to the address "Data module 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 module 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 module 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 module 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 important settings in the SIMATIC panel.

General information for accessing inverter 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

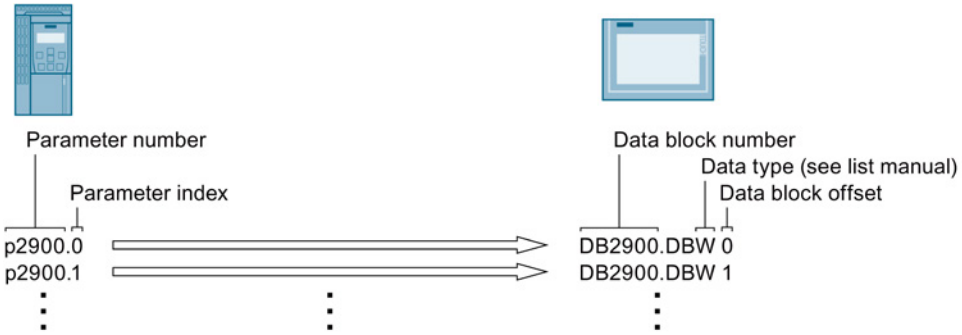
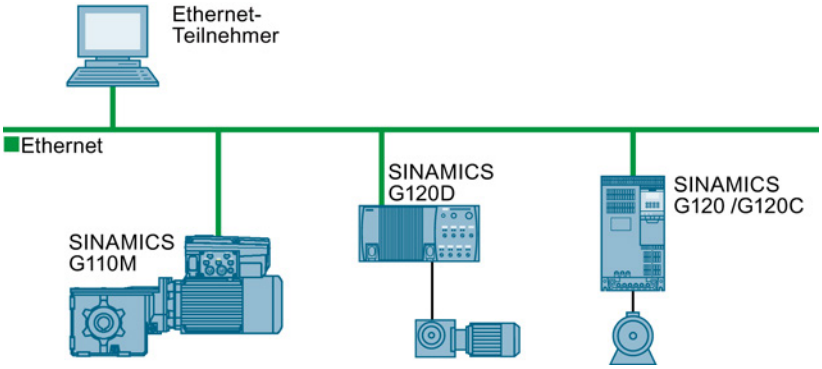


Figure 3-6 Accessing inverter parameters using a SINAMICS G120 as example

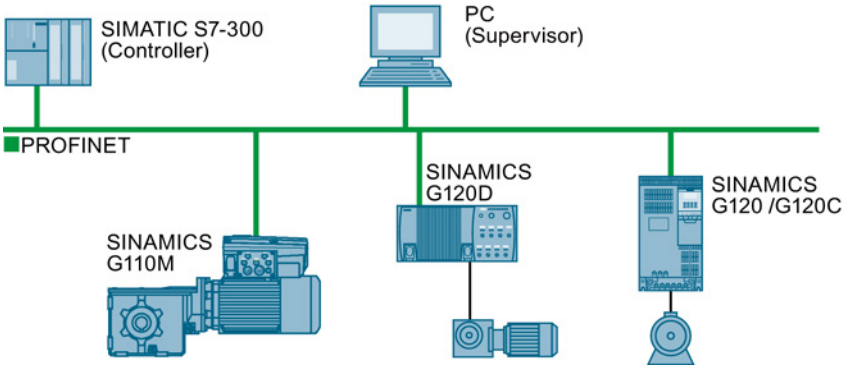
3.4 Communication via PROFINET

You can either communicate via Ethernet using the inverter, or integrate the inverter in a PROFINET network.

- The inverter as an Ethernet station (Page 51)



- PROFINET IO operation (Page 47)



In PROFINET IO operation, the inverter supports the following functions:

- RT
- IRT
  The inverter transmits the clock synchronism but does not support clock synchronism.
- MRP
  Media redundancy, impulsed with 200 ms
  Requirement: Ring topology
- MRPD
  Media redundancy, impulse-free
  Requirement: IRT and the ring topology created in the control
- Diagnostic alarm
  in accordance with the fault classes specified in the PROFIdrive profile. See Activating diagnostics via the control (Page 48).
- Device replacement without removable medium
  Requirement: Topology created in the control
- Shared device
  only in the case of control units with fail-safe functions (see Safety function manual)

Further information on PROFINET can be found on the Internet using the following links:

- General information about PROFINET can be found at Industrial Communication (http://www.automation.siemens.com/mcms/automation/en/industrial-communications/profinet/Pages/Default.aspx).
- The configuration of the functions is described in the PROFINET system description (http://support.automation.siemens.com/WW/view/en/19292127) manual.

3.4.1 Inverter with PROFINET interface

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

You can implement either a ring or line-type topology using the two sockets at the inverter. 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- 12 Assignment table

Inverter/Control Unit		Connection via		
		X150 P1/ X150 P2 (RJ45, IP20)  8 ... 1	X03/X04 (RJ45, IP57)  1 ... 8	X03/X04 (M12, IP57) 
	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	
	G110M			
	• CU240M PN			x

Table 3- 13 Connector pin assignments

Signal	X150 P1/ X150 P2 (RJ45, IP20) 	X03/X04 (RJ45, IP57) 	X03/X04 (M12, IP57) 
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 connectors

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

  Information for assembling the SIMATIC NET Industrial Ethernet FastConnect RF45 Plug 180 can be found on the Internet at "Assembly instructions for SIMATIC NET Industrial Ethernet FastConnect RJ45 Plug (http://support.automation.siemens.com/WW/view/en/37217116/133300)".

3.4.2 Integrating inverters into PROFINET

Procedure

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

1. Integrate the inverter 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 und X04.

   The position of the sockets is available in the operating instructions for the inverter. Pin assignment, see Inverter with PROFINET interface (Page 45).

   The maximum permitted cable length from the previous station and to the subsequent one is 100 m.

2. Externally supply the inverter 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 connected the inverter to the control using PROFINET.

3.4.3 PROFINET IO operation

3.4.3.1 What do you need for communication via PROFINET?

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

Questions	Answer/description	Example
Is the inverter correctly connected to the bus network?	See: Integrating inverters into PROFINET (Page 46)	
Do the IP address and device name in the converter and controller match?	See Configuring communication to the control (Page 47)	See manuals for your inverter, fieldbus communication manual (Page 166)
Is the same telegram set in the converter as in the higher-level controller?	Set the telegram in the converter, see: Select telegram (Page 59)	
Are the signals that the converter and the controller exchange via PROFINET correctly interconnected?	PROFIdrive-compliant interconnection in the converter, see: PROFIdrive profile (Page 15)	

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

If, in your plant or system, communication with the control system should continue to function even when the line voltage is switched off, then you must externally supply the inverter/Control Unit with 24 V DC. To do this, use terminals 31 and 32 – or connector X01. You can find additional details in the operating instructions for the inverter or the Control Unit.

3.4.3.2 Configuring communication to the control

Configuring the communication using SIMATIC S7 control

- If the inverter is included in the hardware library of HW-Config, you can configure the inverter.
- You have the following options, if the inverter is not included in the hardware library:
  - Install the more up to date STARTER version
  - Install the GSDML of the inverter via "Install options/GSDML file" in HW Config.

Additional information on this topic is provided in the "Fieldbuses" Function Manual, also see Configuring PROFINET communication (Page 166).

Configuring the communication using a non-Siemens control

1. Import the device file (GSDML) of the inverter into the configuring tool of your control system.
2. Configure the communication.

Installing GSDML

Procedure

To install the GSDML of the inverter into the configuring tool of your control system, proceed as follows:

1. Load the GSDML to your PC.
   - On the Internet: GSDML (http://support.automation.siemens.com/WW/view/en/22339653/133100).
   - From your inverter:

     Insert a memory card into the converter.

     Set p0804 = 12.

     The inverter writes the GSDML as zipped file (*.zip) into directory /SIEMENS/SINAMICS/DATA/CFG on the memory card.
2. Unzip the GSDML file to a folder on your computer.
3. Import the GSDML into the configuring tool of your control system.

You have now installed the GSDML.

3.4.3.3 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.

You must select the functionality in the higher-level control (see Activate diagnostic messages via STEP 7 (Page 169)) and activate it by booting up.

3.4.3.4 PROFIenergy profile

PROFIenergy is an energy management standard for production plants, based on the PROFINET communication protocol. The functionality is certified and described in the PROFIenergy profile of the PNO.

The converters support the PROFIenergy profile V1.1 and the function unit class 3.

PROFIenergy functions of the converter

The control transfers the PROFIenergy commands in acyclic operation to the converter in data set 80A0 hex.

Parameters r5600 to p5614 are reserved for PROFIenergy functions in the converter.

The SINAMICS G120D supports PROFIenergy hibernation mode 1:

- The converter switches off the supply voltage of its digital inputs unless they are interconnected with r5613.x (display hibernation) or are used as fail-safe inputs.
- The converter switches off the supply voltage of its encoders unless they are HTL encoders assigned to the position controller.

The converters SINAMICS G110M, G120 and G120C support PROFIenergy hibernation mode 2.

The following commands and queries are available for the control:

PROFIenergy control commands

- Start_Pause
  Switches from operating state S1 or S2 (start inhibit or start readiness) to energy-saving mode independently of the break duration.
- START_Pause_with_time_response
  Switches from operating state S1 or S2 (start inhibit or start readiness) to energy-saving mode independently of the break duration and also specifies the transition times in the command response.
- End_Pause
  Switches from energy-saving mode to the operating state independently of the break duration.
  Cancels switching from the operating state to the energy-saving mode.

PROFIenergy status requests

- List_Energy_Saving_Modes
  Determines all supported energy-saving modes.
- Get_Mode
  Determines information about the selected energy-saving mode.
- PEM_Status
  Determines the current PROFIenergy status.
- PEM_Status_with_CTTO
  Determines the current PROFIenergy status, such as the PEM_Status, together with the regular transition time to the operating state.
- PE_Identify
  Determines the supported PROFIenergy commands.
- Query_Version
  Shows the implemented PROFIenergy profile.
- Get_Measurement_List
  This command returns the measured value IDs that can be accessed using the "Get_Measurement_Values" command.
- Get_Measurement_List_with_object_number
  This command returns the measured value IDs and the associated object number that can be accessed using the "Get_Measurement_Values_with_object_number" command.
- Get_Measurement_Values
  The command returns the requested measured value using the measured value ID
- Get_Measurement_Values_with_object_number
  The command returns the requested measured values using the measured value ID and the object number. The object number corresponds to the drive object ID.

Displaying

- r5600: Current energy-saving mode
- r5613: Interconnectable display of the PROFIenergy status
- A08800: PROFIenergy energy-saving mode active

General settings for PROFIenergy

- Minimum pause time: p5602
  - when the pause time sent with the "Start_Pause" command is greater than or equal to the value for p5602[1], the converter enters energy-saving mode.
  - If the pause time is less than p5602[1], the converter ignores the command.
- Maximum pause time: p5606

Block PROFIenergy

If you set p5611.0 = 1, you block the response of the converter to PROFIenergy control commands. In this case, the converter ignores the PROFIenergy control commands.

Transition to the energy-saving mode from the PROFIdrive operating state (S4)

If you set p5611.2 = 1, you enable transition to the energy-saving mode from the PROFIdrive operating state (S4)

To do so, you must set one of the following settings:

- p5611.1 = 1: With the transition to the energy-saving mode, the converter issues an OFF1 command and enters the start-inhibit state (S1).
- p5611.1 = 0: You can use p5614 to interconnect a signal source that switches the converter off and places it in the start-inhibit state (S1).

If the control sends the command "End_Pause" or "Start_Pause" with a pause time of 0, the converter continues to run automatically – if the enables are still set.

PROFIenergy measured values

PROFIenergy					SINAMICS source parameters		Range of values
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	Smoothed power factor	0 … 1
200	Active energy import	2	11	Wh	r0039[1]	Energy accepted	-

3.4.4 The inverter as an Ethernet station

Integrating inverters with PROFINET/EtherNET/IP interface via DHCP in Ethernet

For the case that the inverter is not controlled in cyclic operation from a higher-level control system, you have the option of integrating the inverter in the network as Ethernet station. In this case, you integrate the inverter into the Ethernet network via DHCP. As a consequence, you have the option, using a PC on which STARTER is installed, to carry out diagnostics, commission or make parameter changes from any location within the Ethernet network.

You can identify the inverter either using the MAC address (r8935) or using the device name of the inverter (r8930).

In the factory setting, the inverter is set for PROFINET IO communication. In order that you can integrate the inverter into Ethernet via DHCP, you must change the inverter settings as subsequently described.

Integrating inverters into an Ethernet network using the parameter settings

Procedure

Proceed as follows to integrate an inverter into the Ethernet via DHCP:

1. Set p8924 (PN DHCP Mode) = 2 or 3
   - With p8924 = 2, address is assigned by the DHCP server based on the MAC address of the inverter.
   - With p8924 = 3, address is assigned by the DHCP server based on the device name of the inverter.
2. Set p8925 = 2 or 3
   - With p8925 = 2, the inverter saves the actual configuration. The settings become effective at the next power on.
   - With p8925 = 3, you reset the PROFINET interface to the factory setting (PROFINET IO communication). The settings become effective at the next power on.

You have integrated the inverter into the Ethernet via DHCP.

Displays

The actual PN-DHCP mode is displayed in r8934, the MAC address in r8935.

Additional settings

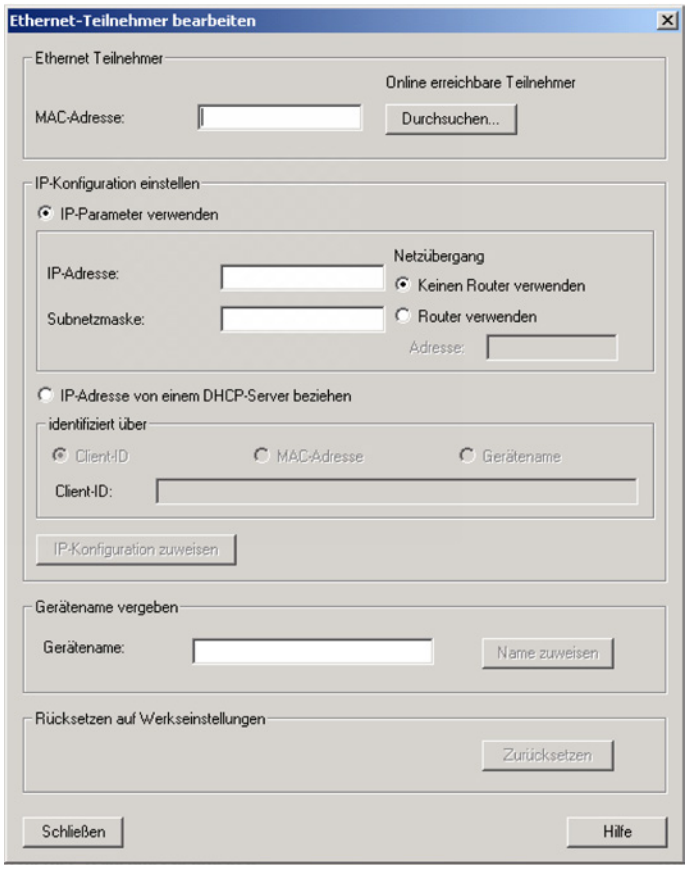
You can find additional setting options and messages (A08565) in the List Manual (Page 182).

With p8924 = 0, you inhibit the 'DHCP mode

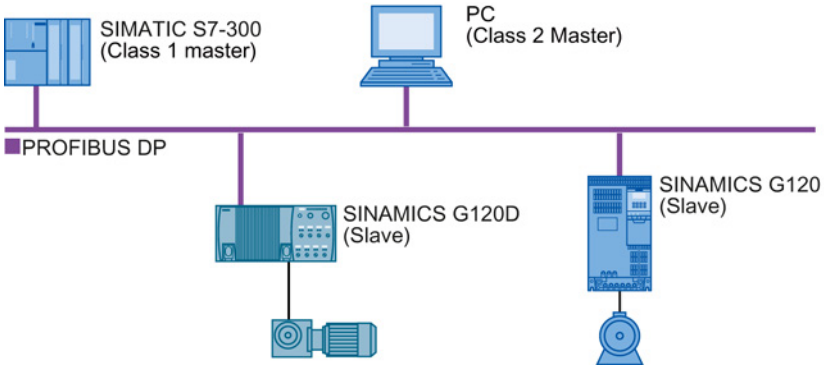
Additional options of integrating inverters into Ethernet

Other options of integrating inverters into the Ethernet involve, e.g. Proneta or STEP 7.

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



3.5 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 on the Internet at the following links:

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

3.5.1 Inverters 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 inverter. You can use switches to realize other topologies.

Table 3- 14 Assignment table - connectors

Inverter/Control Unit		Connection via		
		X126 (D Sub socket, IP20) 	X03, on (M12, IP57) 	X04, off (M12, IP57) 
	G120			
	• CU230P-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
	G110M			
	• CU240M DP		x	x

Table 3- 15 Connector pin assignments

Signal	X126 (D Sub socket, IP20) 	X03, on (M12, IP57) 	X04, off (M12, IP57) 
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 order numbers for connecting the PROFIBUS cable:

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

3.5.2 What do you need for communication via PROFIBUS?

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

Questions	Description	Examples
Is the inverter correctly connected to the PROFIBUS?	See Section: Integrating inverters into PROFIBUS (Page 56).	---
Have you configured the communication between the inverter and the higher-level controller?	See Section: Configuring the communication using SIMATIC S7 control (Page 56)	See Configuring PROFIBUS communication (Page 163)
Do the addresses in the inverter and the higher-level controller match?	See Section: Setting the address (Page 58).	
Is the same telegram set in the higher-level controller and in the inverter?	Adapt the telegram in the inverter. See Section: Select telegram (Page 59).	
Are the signals that the inverter and the controller exchange via PROFIBUS correctly interconnected?	Adapt the interconnection of the signals in the controller to the inverter. For the PROFIdrive-compliant interconnection in the inverter, see also Section: PROFIdrive profile (Page 15).	

3.5.3 Integrating inverters into PROFIBUS

Procedure

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

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

   The position of the socket is explained in the operating instructions for the inverter. Pin assignment, see Inverters with PROFIBUS interface (Page 54).

   The maximum permitted cable length to the previous station and the subsequent one is 100 m at a baud rate of 12 Mbit/s. You can achieve a maximum cable length of 400 m by using a maximum of 3 repeaters.
2. Externally supply the inverter 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 inverter to the control system using PROFIBUS DP.

3.5.4 Configuring communication to the control system

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

3.5.4.1 Configuring the communication using SIMATIC S7 control

- If the inverter is listed in the hardware library of HW-Conifg, you can configure the communication in the SIMATIC control.
- If the inverter is not listed in the hardware library, you can either install the newest STARTER version or install the GSD of the inverter through "Extras/GSD-Install file" in HW-Config. See also Installing the GSD (Page 57).

  When you have installed the GSD, configure the communication in the SIMATIC control.

Example of configuring the communication with a SIMATIC control system

You can find an example of configuring the communication with a SIMATIC control system in the Section "Configuring PROFIBUS communication (Page 163)".

3.5.4.2 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 inverter in the control before you configure the communication. You will find details in "Installing the GSD (Page 57)".

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

3.5.4.3 Installing the GSD

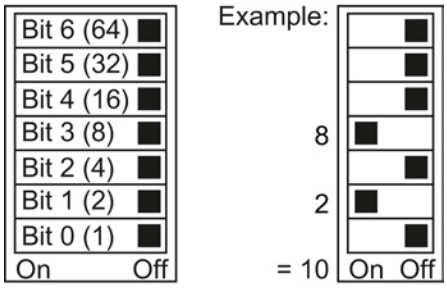
Procedure

To load the GSD of the inverter in the control, proceed as follows:

1. Obtain the GSD
   - either in the Internet at (GSD (http://support.automation.siemens.com/WW/view/en/22339653/133100)).
   - or from your inverter. To do this, insert a memory card in the inverter, and set p0804 = 12. This means that you save the GSD as zipped file (DPGSD.ZIP) in directory /SIEMENS/SINAMICS/DATA/CFG on the memory card.
2. Unzip the GSDfile in a folder on your computer.
3. Import the GSDinto the configuration tool of your control.

You have now installed the GSD file.

3.5.5 Setting the address



You set the PROFIBUS address of the inverter using the address switch on the Control Unit, in parameter p0918 or in STARTER.

In parameter p0918 (factory setting: 126) or in STARTER, you can only set the address, if all address switches are set to "OFF" (0) or "ON" (1).

If you have specified a valid address with the address switches, this address will always be the one that takes effect and parameter p0918 cannot be changed.

Valid address range: 1 … 125

The position of the address switch is described in the operating instructions of the inverter.

Procedure

1
2

To change the bus address, proceed as follows:

1. Set the address using one of the subsequently listed options:
   - using the address switch
   - from an operator panel using parameter p0918
   - in STARTER using screen form "Control Unit/Communication/PROFIBUS" – or using the expert list in parameter p0918

     After you have changed the address in STARTER, carry out RAM to ROM ( ).
2. Switch on the inverter power supply and, if available, the 24 V power supply for the Control Unit.
3. Switch on the voltages again after all LEDs at the inverter have gone dark.

You have now changed the bus address.

3.6 Select telegram

Precondition

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

Telegrams for SINAMICS G120 inverters

The following table shows all of the telegrams that are available for the G120 inverter.

p0922 =
- 1: Standard telegram 1, PZD-2/2 (factory setting, exceptions: CU250D and
- 2: CU250S)
- 3: standard telegram 2, PZD-4/4
- 4: standard telegram 3, PZD-5/9
- 7: standard telegram 4, PZD-6/14
- 9: standard telegram 7, PZD-2/2 (factory setting CU250D)
- 20: standard telegram 9, PZD-10/5
- 110: standard telegram 20, PZD-2/6
- 112: SIEMENS telegram 110, PZD-12/7
- 350: SIEMENS telegram 111, PZD-12/12
- 352: SIEMENS telegram 350, PZD-4/4
- 353: SIEMENS telegram 352, PZD-6/6
- 354: SIEMENS telegram 353, PZD-2/2, PKW-4/4
- 999: SIEMENS telegram 354, PZD-6/6, PKW-4/4
  free telegram (extend telegram/change signal interconnection (Page 26))
  (factory setting CU250S)

If you change parameter p0922, then you can only select the telegrams that are available for your particular inverter.

A more detailed depiction of the individual telegrams can be found in Section Cyclic communication (Page 15).

PROFIsafe telegram selection

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

Communication via EtherNet/IP

4

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

You have the following options of integrating SINAMICS G120 inverters into EtherNet/IP:

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

You can find details on the various options in Section Configuring communication via EtherNet/IP (Page 65).

4.1 Inverters with Ethernet/IP interface

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

You can implement either a ring or line-type topology using the two sockets at the inverter. 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 4- 1 Assignment table

Inverter/Control Unit		Connection via		
		X150 P1/ X150 P2 (RJ45, IP20) 	X03/X04 (RJ45, IP57) 	X03/X04 (M12, IP57) 
	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	
	G110M			
	• CU240M PN			x

Table 4- 2 Connector pin assignments

Signal	X150 P1/ X150 P2 (RJ45, IP20) 	X03/X04 (RJ45, IP57) 	X03/X04 (M12, IP57) 
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 connectors

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

  Information for assembling the SIMATIC NET Industrial Ethernet FastConnect RF45 Plug 180 can be found on the Internet at "Assembly instructions for SIMATIC NET Industrial Ethernet FastConnect RJ45 Plug (http://support.automation.siemens.com/WW/view/en/14293080/133300)".

4.2 Connect converter to Ethernet/IP

Procedure

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

1. Connect the inverter to the control system via an Ethernet cable.
2. You create an object for data exchange.

   You have the following options:

   – You load the EDS file into your control. You can find the file in the Internet at: (http://support.automation.siemens.com/WW/view/en/48351511).

   – If your control does not accept the EDS file, then create a generic I/O module (Page 76) in your control.

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

Routing and shielding Ethernet cables

Information can be found on the Internet: EtherNet/IP guidelines (http://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP/EtherNetIPLibrary/tabid/76/lng/en-US/Default.aspx).

Commissioning the inverter in an EtherNet/IP network

To commission the inverter, connect the inverter via the USB interface with your computer on which STARTER, Version > 4.2 has been installed.

You can find details on this in the Operating instructions (Page 182) of the inverter or the Control Unit.

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 inverter via the fieldbus.

- Is the inverter correctly connected to the EtherNet/IP?
- Is the EDS file (http://support.automation.siemens.com/WW/view/en/48351511) installed in your control system?
- Have the bus interface and IP address been correctly set?
- Have the signals that the inverter and the control system exchange been correctly interconnected?

4.4 Configuring communication via EtherNet/IP

Basic settings

Procedure

Make the following settings in order to communicate with a higher-level control via EtherNet/IP:

1. p2030: set a value of 10: Fieldbus interface protocol selection Ethernet/IP:
2. p8921: Enter the IP address. You can find the currently valid address in r8931.
3. p8923: Enter the subnet mask. You can find the currently valid subnet mask in r8933.
4. p8922: Enter the standard gateway. You can find the currently valid Default Gateway in r8932.
5. p8920: Enter the station name.
6. p8925: Set a value of 2: Save and activate PN interface configuration
7. Switch off the inverter, and then any possibly existing external 24 V power supply.
8. Switch on the inverter again.

You have configured the inverter for communication via Ethernet/IP.

Parameters p8921 … p8925 apply if p2030 = 10 is set, for EtherNet/IP, even if the parameter names indicates PROFINET.

Communication settings

You set the communication using parameter p8980. You have the following options

Communication via the SINAMICS profile

- The SINAMICS profile is a drive profile for EtherNet/IP defined by Siemens, based on PROFIdrive, and is factory set in the inverters.
  Setting: p8980 = 0
  With the SINAMICS profile, you can use each of the telegrams listed in parameter p0922

Communication via the ODVA AC/DC drive profile

- The ODVA AC/DC drive profile is a drive profile defined by the ODVA organization
  Setting: p8980 = 1
  With the AC/DC profile of ODVA, you select the standard telegram, p0922 = 1

Communication settings via EtherNet/IP objects and assemblies

You cannot work with the EDS file if you wish to use the assemblies described in Section Supported objects (Page 67). In this case, you must integrate the inverter into your control system. Details on this topic can be found in the documentation of your control system.

Setting the bus monitoring time

You set the bus monitoring using parameter p8840 in the inverter.

If you set this parameter to 0, the inverter continues to operate even if the bus develops a fault condition. If you set a time ≠ 0, then the inverter switches off with F08501 "Setpoint timeout" if the control system does not issue any signals within this time.

4.5 Additional settings if you are working with the AC/DC profile

If you change the following settings in the inverter by accessing the appropriate parameters, you must switch-off the inverter and switched it on again in order that these changes become effective. The changes become immediately effective when making the changes via the control system with objects 90 hex or 91 hex.

Setting the off response for the motor

You set the standard off response for the inverter using parameter p8981:

- p8981 = 0: OFF1 (factory setting), also corresponds to the setting in the SINAMICS profile
- p8981 = 1: OFF2

You can find details about OFF1 and OFF2 in the operating instructions of the Control Unit in the Section "Switching on and switching off a motor".

Setting the speed and torque scaling

You scale the speed and torque display using parameter p8982 or p8983. Setting range: 2^5 to 2^{-5}.

Displaying the maximum process data that can be transferred (PZD)

- r2067[0] maximum interconnected PZD length - receiving
- p2067[1] maximum interconnected PZD length - sending

4.6 Supported objects

EtherNet/IP objects supported by the G120

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	30	Motor Data Object		x	
29 hex	31	Supervisor Object		x	
2A hex	42	Drive Object		x	
32C hex	44	Siemens Drive Object			x
32D hex	45	Siemens Motor Data Object			x
90 hex	144	Parameter object			x
91 hex	145	Parameter object free access (DS47)			x
F5 hex	245	TCP/IP Interface object 1)	x		
F6 hex	246	Ethernet Link object 1)	x		
401 hex … 43E hex	1025 … 1086	Parameter object			x

1) these objects are part of the EtherNet/IP system management.

ODVA AC/DC assembly

Number		required/ optional	Type	Name
hex	dec			
14 hex	20	Required	Sending	Basic Speed Control Output
15 hex	21	Optional	Sending	Extended Speed Control Output
16 hex	22	Optional	Sending	Speed and Torque Control Output
17 hex	23	Optional	Sending	Extended Speed and Torque Control Output
18 hex	24	Optional	Sending	Process Control Output
19 hex	25	Optional	Sending	Extended Process Control Output
46 hex	70	Required	Receiving	Basic Speed Control Input
47 hex	71	Optional	Receiving	Extended Speed Control Input
48 hex	72	Optional	Receiving	Speed and Torque Control Input
49 hex	73	Optional	Receiving	Extended Speed and Torque Control Input
4A hex	74	Optional	Receiving	Process Control Input
4B hex	75	Optional	Receiving	Extended Process 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)							

Assembly Basic Speed Control with parameter assembly, Instance Number: 120, 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)							
4	Data Out 1 Value (Low Byte)							
5	Data Out 1 Value (High Byte)							
6	Data Out 2 Value (Low Byte)							
7	Data Out 2 Value (High Byte)							
8	Data Out 3 Value (Low Byte)							
9	Data Out 3 Value (High Byte)							
10	Data Out 4 Value (Low Byte)							
11	Data Out 4 Value (High Byte)							
12	Data Out 5 Value (Low Byte)							
13	Data Out 5 Value (High Byte)							
14	Data Out 6 Value (Low Byte)							
15	Data Out 6 Value (High Byte)							
16	Data Out 7 Value (Low Byte)							
17	Data Out 7 Value (High Byte)							
18	Data Out 8 Value (Low Byte)							
19	Data Out 8 Value (High Byte)							
20	Data Out 9 Value (Low Byte)							
21	Data Out 9 Value (High Byte)							
22	Data Out 10 Value (Low Byte)							
23	Data Out 10 Value (High Byte)							

Assembly Basic Speed Control with parameter assembly, Instance Number: 170, 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	Data In 1 Value (Low Byte)							
5	Data In 1 Value (High Byte)							
6	Data In 2 Value (Low Byte)							
7	Data In 2 Value (High Byte)							
8	Data In 3 Value (Low Byte)							
9	Data In 3 Value (High Byte)							
10	Data In 4 Value (Low Byte)							
11	Data In 4 Value (High Byte)							
12	Data In 5 Value (Low Byte)							
13	Data In 5 Value (High Byte)							
14	Data In 6 Value (Low Byte)							
15	Data In 6 Value (High Byte)							
16	Data In 7 Value (Low Byte)							
17	Data In 7 Value (High Byte)							
18	Data In 8 Value (Low Byte)							
19	Data In 8 Value (High Byte)							
20	Data In 9 Value (Low Byte)							
21	Data In 9 Value (High Byte)							
22	Data In10 Value (Low Byte)							
23	Data In 10 Value (High Byte)							

Assembly Extended Speed Control, Instance Number: 21, type: Output

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

Assembly Extended Speed Control, Instance Number: 71, type: Input

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	CtrL From Net	Ready	Running Reverse	Running Forward	Warning	Faulted
1	Drive State							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							

Assembly Extended Speed Control with parameter assembly, Instance Number: 121, type: Output

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0		NetRef	Net CtrL			Fault Reset	RUN Reverse	RUN Forward
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Data Out 1 Value (Low Byte)							
5	Data Out 1 Value (High Byte)							
6	Data Out 2 Value (Low Byte)							
7	Data Out 2 Value (High Byte)							
8	Data Out 3 Value (Low Byte)							
9	Data Out 3 Value (High Byte)							
10	Data Out 4 Value (Low Byte)							
11	Data Out 4 Value (High Byte)							
12	Data Out 5 Value (Low Byte)							
13	Data Out 5 Value (High Byte)							
14	Data Out 6 Value (Low Byte)							
15	Data Out 6 Value (High Byte)							
16	Data Out 7 Value (Low Byte)							
17	Data Out 7 Value (High Byte)							
18	Data Out 8 Value (Low Byte)							
19	Data Out 8 Value (High Byte)							
20	Data Out 9 Value (Low Byte)							
21	Data Out 9 Value (High Byte)							
22	Data Out 10 Value (Low Byte)							
23	Data Out 10 Value (High Byte)							

Assembly Extended Speed Control with parameter assembly, Instance Number: 171, type: Input

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Ref From Net	Ready	Running Reverse	Running Forward	Warning	Faulted
1	Drive State							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Data In 1 Value (Low Byte)							
5	Data In 1 Value (High Byte)							
6	Data In 2 Value (Low Byte)							
7	Data In 2 Value (High Byte)							
8	Data In 3 Value (Low Byte)							
9	Data In 3 Value (High Byte)							
10	Data In 4 Value (Low Byte)							
11	Data In 4 Value (High Byte)							
12	Data In 5 Value (Low Byte)							
13	Data In 5 Value (High Byte)							
14	Data In 6 Value (Low Byte)							
15	Data In 6 Value (High Byte)							
16	Data In 7 Value (Low Byte)							
17	Data In 7 Value (High Byte)							
18	Data In 8 Value (Low Byte)							
19	Data In 8 Value (High Byte)							
20	Data In 9 Value (Low Byte)							
21	Data In 9 Value (High Byte)							
22	Data In 10 Value (Low Byte)							
23	Data In 10 Value (High Byte)							

Assembly Basic Speed and Torque Control , Instance Number: 22, 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)							
4	Torque Reference (High Byte)							
5	Torque Reference (High Byte)							

Assembly Basic Speed and Torque Control , Instance Number: 72, type: Input

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

Assembly Basic Speed and Torque Control with parameter assembly , Instance Number: 122, 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)							
4	Torque Reference (High Byte)							
5	Torque Reference (High Byte)							
6	Data Out 1 Value (Low Byte)							
7	Data Out 1 Value (High Byte)							
8	Data Out 2 Value (Low Byte)							
9	Data Out 2 Value (High Byte)							
10	Data Out 3 Value (Low Byte)							
11	Data Out 3 Value (High Byte)							
12	Data Out 4 Value (Low Byte)							
13	Data Out 4 Value (High Byte)							
14	Data Out 5 Value (Low Byte)							
15	Data Out 5 Value (High Byte)							
16	Data Out 6 Value (Low Byte)							
17	Data Out 6 Value (High Byte)							
18	Data Out 7 Value (Low Byte)							
19	Data Out 7 Value (High Byte)							
20	Data Out 8 Value (Low Byte)							
21	Data Out 8 Value (High Byte)							
22	Data Out 9 Value (Low Byte)							
23	Data Out 9 Value (High Byte)							
24	Data Out 10 Value (Low Byte)							
25	Data Out 10 Value (High Byte)							

Assembly Basic Speed and Torque Control with parameter assembly , Instance Number: 172, type: Input

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
						Running Forward		Faulted
1								
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (High Byte)							
5	Torque Actual (High Byte)							
6	Data In 1 Value (Low Byte)							
7	Data In 1 Value (High Byte)							
8	Data In 2 Value (Low Byte)							
9	Data In 2 Value (High Byte)							
10	Data In 3 Value (Low Byte)							
11	Data In 3 Value (High Byte)							
12	Data In 4 Value (Low Byte)							
13	Data In 4 Value (High Byte)							
14	Data In 5 Value (Low Byte)							
15	Data In 5 Value (High Byte)							
16	Data In 6 Value (Low Byte)							
17	Data In 6 Value (High Byte)							
18	Data In 7 Value (Low Byte)							
19	Data In 7 Value (High Byte)							
20	Data In 8 Value (Low Byte)							
	Data In 8 Value (High Byte)							
22	Data In 9 Value (Low Byte)							
23	Data In 9 Value (High Byte)							
24	Data In 10 Value (Low Byte)							
25	Data In 10 Value (High Byte)							

Extended Speed and Torque Control, Instance Number: 23, type: Output

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

Extended Speed and Torque Control, Instance Number: 73, type: Input

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Crtl From Net	Ready	Running Reverse	Running Forward	Warning	Faulted
1	Drive State							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (High Byte)							
5	Torque Actual (High Byte)							

Basic Speed and Torque Control with parameter assembly, Instance Number: 123, type: Output

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0		NetRef	Net CtrL			Fault Reset	RUN Reverse	RUN Forward
1								
2	Speed Reference (Low Byte)							
3	Speed Reference (High Byte)							
4	Torque Reference (High Byte)							
5	Torque Reference (High Byte)							
6	Data Out 1 Value (Low Byte)							
7	Data Out 1 Value (High Byte)							
8	Data Out 2 Value (Low Byte)							
9	Data Out 2 Value (High Byte)							
10	Data Out 3 Value (Low Byte)							
11	Data Out 3 Value (High Byte)							
12	Data Out 4 Value (Low Byte)							
13	Data Out 4 Value (High Byte)							
14	Data Out 5 Value (Low Byte)							
15	Data Out 5 Value (High Byte)							
16	Data Out 6 Value (Low Byte)							
17	Data Out 6 Value (High Byte)							
18	Data Out 7 Value (Low Byte)							
19	Data Out 7 Value (High Byte)							
20	Data Out 8 Value (Low Byte)							
21	Data Out 8 Value (High Byte)							
22	Data Out 9 Value (Low Byte)							
23	Data Out 9 Value (High Byte)							
24	Data Out 10 Value (Low Byte)							
25	Data Out 10 Value (High Byte)							

Basic Speed and Torque Control with parameter assembly, Instance Number: 173, type: Input

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	At Reference	Ref From Net	Crtl From Net	Ready	Running Reverse	Running Forward	Warning	Faulted
1	Drive State							
2	Speed Actual (Low Byte)							
3	Speed Actual (High Byte)							
4	Torque Actual (High Byte)							
5	Torque Actual (High Byte)							
6	Data In 1 Value (Low Byte)							
7	Data In 1 Value (High Byte)							
8	Data In 2 Value (Low Byte)							
9	Data In 2 Value (High Byte)							
10	Data In 3 Value (Low Byte)							
11	Data In 3 Value (High Byte)							
12	Data In 4 Value (Low Byte)							
13	Data In 4 Value (High Byte)							
14	Data In 5 Value (Low Byte)							
15	Data In 5 Value (High Byte)							
16	Data In 6 Value (Low Byte)							
17	Data In 6 Value (High Byte)							
18	Data In 7 Value (Low Byte)							
19	Data In 7 Value (High Byte)							
20	Data In 8 Value (Low Byte)							
21	Data In 8 Value (High Byte)							
22	Data In 9 Value (Low Byte)							
23	Data In 9 Value (High Byte)							
24	Data In 10 Value (Low Byte)							
25	Data In 10 Value (High Byte)							

4.7 Create generic I/O module

For certain controllers, 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.

Procedure

To do this, proceed as follows:

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 have selected in STARTER, r2067[0] (input), r2067[1] (output), for example: Standard telegram 2/2.
3. In STARTER, set the same values for IP address, Subnet Mask, Default Gateway and the Name of Station as in the control system (see Configuring communication via EtherNet/IP (Page 65))

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

4.8 The inverter as an Ethernet station

Integrating inverters with PROFINET/EtherNET/IP interface via DHCP in Ethernet

For the case that the inverter is not controlled in cyclic operation from a higher-level control system, you have the option of integrating the inverter in the network as Ethernet station. In this case, you integrate the inverter into the Ethernet network via DHCP. As a consequence, you have the option, using a PC on which STARTER is installed, to carry out diagnostics, commission or make parameter changes from any location within the Ethernet network.

You can identify the inverter either using the MAC address (r8935) or using the device name of the inverter (r8930).

In the factory setting, the inverter is set for PROFINET IO communication. In order that you can integrate the inverter into Ethernet via DHCP, you must change the inverter settings as subsequently described.

Integrating inverters into an Ethernet network using the parameter settings

Procedure

Proceed as follows to integrate an inverter into the Ethernet via DHCP:

1. Set p8924 (PN DHCP Mode) = 2 or 3
   - With p8924 = 2, address is assigned by the DHCP server based on the MAC address of the inverter.
   - With p8924 = 3, address is assigned by the DHCP server based on the device name of the inverter.
2. Set p8925 = 2 or 3
   - With p8925 = 2, the inverter saves the actual configuration. The settings become effective at the next power on.
   - With p8925 = 3, you reset the PROFINET interface to the factory setting (PROFINET IO communication). The settings become effective at the next power on.

You have integrated the inverter into the Ethernet via DHCP.

Displays

The actual PN-DHCP mode is displayed in r8934, the MAC address in r8935.

Additional settings

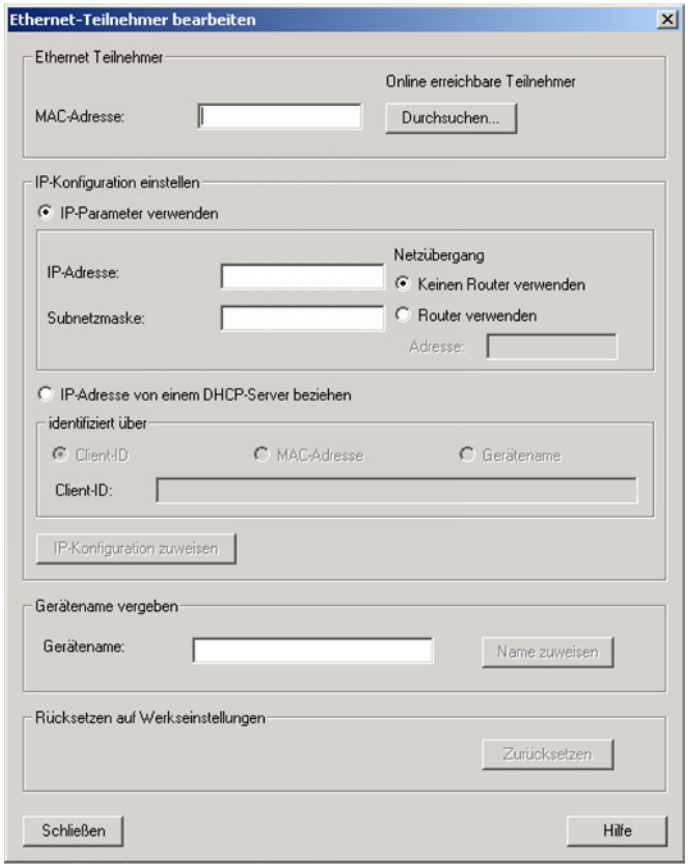
You can find additional setting options and messages (A08565) in the List Manual (Page 182).

With p8924 = 0, you inhibit the 'DHCP mode

Additional options of integrating inverters into Ethernet

Other options of integrating inverters into the Ethernet involve, e.g. Proneta or STEP 7.

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



Communication via RS485

5

Table 5- 1 Assignment table - fieldbus systems via RS485

Inverter/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	✓	✓	---	---
	G110M				
	• CU240M USS	✓	✓	---	---

5.1 Inverter 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

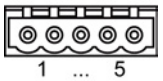
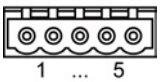
Inverter/Control Unit		Connection via		
		X128 (IP20) 	X03, in (M12, IP57) 	X04, out (M12, IP57) 
	G120			
	• CU230P-2 HVAC	x		
	• CU230P-2 BT	x		
	• CU240B-2	x		
	• CU240E-2	x		
	• CU240E-2 F	x		
	• CU250S-2	x		
	G120C			
	• G120C USS/MB	x		
	G110M			
	• CU240M USS		x	x

Table 5- 3 Pin assignment

Signal	X128 (IP20) 	X03, in (M12, IP57) 	X04, out (M12, IP57) 
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 inverters into a bus system via the RS485 interface

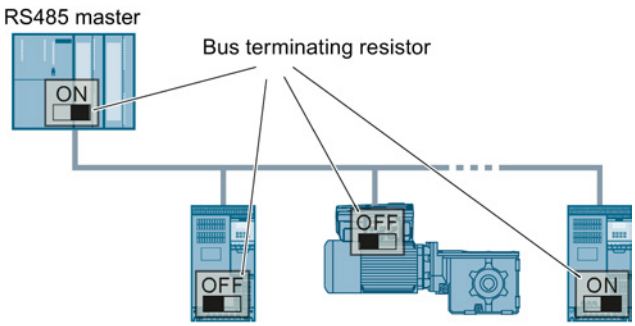
Connecting to a network via RS485

Connect the inverter 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 inverter 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 slaves from the bus without interrupting the cable (this is not possible for inverters with a high degree of protection).

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

If, in your plant or system, communication with the control system should continue to function even when the line voltage is switched off, then you must externally supply the inverter/Control Unit with 24 V DC. To do this, use terminals 31 and 32 – or connector X01. You can find additional details in the operating instructions for the inverter or the Control Unit.

5.3 Communication via USS

The USS protocol is a serial-data connection between one master and one or more slaves. A master is, for example:

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

The inverter is always a slave.

A maximum of 31 slaves is possible.

The maximum cable length is 100 m.

Information about how to connect the inverter to the USS fieldbus is provided in Section Integrating inverters into a bus system via the RS485 interface (Page 81).

5.3.1 Basic settings for communication

Overview

Depending on the particular inverter, you have the following options when setting communication via USS:

- Default setting 21 "USS fieldbus" for all inverters with RS485 interface
- Default setting 108 "BT Mac 8: USS fieldbus" only for CU230P-2 HVAC / CU230P-2 BT. See also Additional manuals for your inverter (Page 182).

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 STARTER under Control Unit/Configuration "Default settings of the setpoint/command sources":
     21: USS fieldbus
   - with BOP-2 for the basic commissioning under Step "MAc PAr P15":
     FB USS
   - using the expert list or parameter number:
     p0015 = 21
2. Set the bus protocol
   - using STARTER under Control Unit/Communication/Fieldbus:
     1 USS
   - using the BOP-2 via parameter p2030:
     p2030 = 1
3. Set the inverter address.
4. Make additional changes based on the parameters listed in the following section.
5. If you are working with STARTER, backup the settings with  .

This means that you have made the settings for communication via USS.

Parameters to adapt the settings for communication via USS

Fieldbus telegram 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 slaves 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 inverter shuts down with fault F01910.

p2040 = 0 ⇒ bus monitoring deactivated.

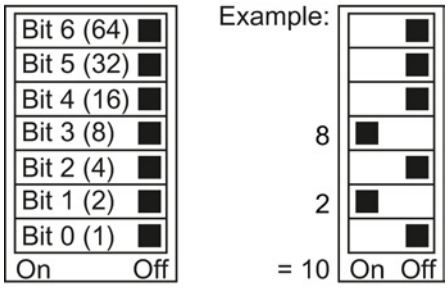
5.3.1.1 Setting the address

You set the bus address of the inverter using the address switches, using parameter p2021 or in STARTER.

Valid address range: 1 … 30

If you have specified a valid address using the address switches, this address will always be the one that takes effect, and parameter p2021 (factory setting: 0) will not be able to be changed.

The position of the address switches can be found in the operating instructions of the Control Unit in the section "Overview of interfaces".



Procedure

To change the bus address, proceed as follows:

1. Set the new address:
   - using the address switches
   - from an operator panel in parameter p2021
   - in STARTER using screen form "Control Unit/Communication/Fieldbus", or using the expert list in parameter p2021
2. Switch on the inverter power supply and, if available, the 24 V power supply for the Control Unit.
3. Switch on the voltages again after all LEDs at the inverter have gone dark.

You have so changed the bus address.

5.3.2 Telegram structure

Overview

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

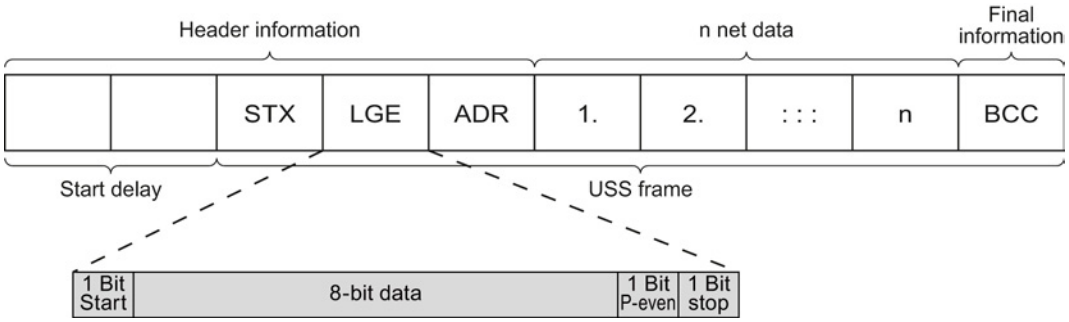
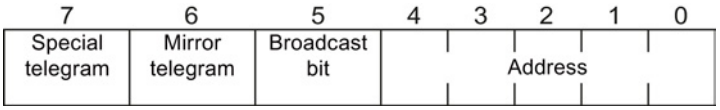


Figure 5-1 Structure of a USS telegram

Telegram part	Description
Start delay / response delay	There is always a start and/or response delay between two telegrams (see alsoTime-out and other errors (Page 92))
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)
ADR	 • 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.
Net data	See section User data range of the USS telegram (Page 85).
BCC	Checksum (exclusive or) across all telegram bytes – with the exception of BCC.

5.3.3 User data range of the USS telegram

The user data area consists of the following elements:

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

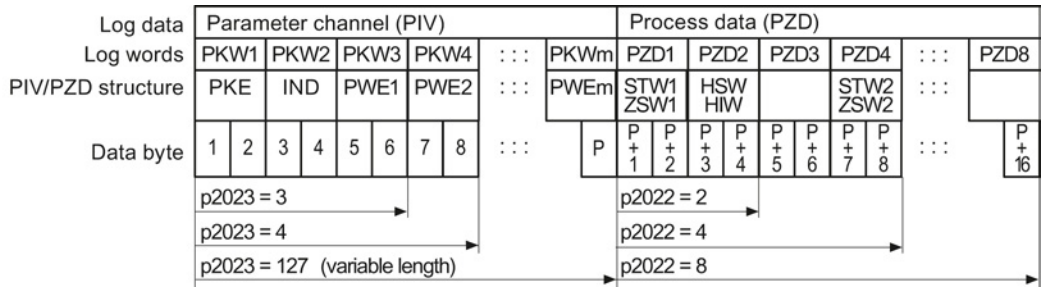


Figure 5-2 USS telegram - user data structure

Parameter channel

In parameter p2023 you specify the parameter channel length.

Parameter channel with fixed and variable length

- 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 enter right-justified in the 4th word.
- p2023 = 127:
  If you set p2023 = 27 (variable length), the send and response telegrams are as long as the task actually 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.

5.3.4 USS parameter channel

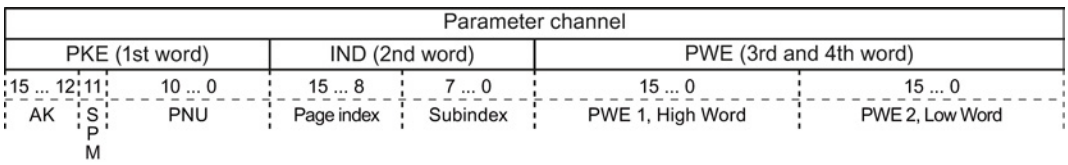
Structure of the parameter channel

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.

Request and response IDs

Bits 12 to 15 of the 1st word of the parameter channel contain the request and response identifier.

Table 5- 4 Request identifiers, control → inverter

Request identifier	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 3 ≡ 8.
We recommend that you use identifiers 6, 7, and 8.

Table 5- 5 Response identifiers, inverter → control

Response identifier	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	Inverter cannot process the request. In the most significant word of the parameter channel, the inverter 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 inverter)
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 → inverter")
6B hex	**No change access for a controller that is enabled**. (operating status of the inverter prevents a parameter change)
86 hex	**Write access only for commissioning (p0010 = 15)** (operating status of the inverter prevents a parameter change)
87 hex	**Know-how protection active, access locked**
C8 hex	**Change request below the currently valid limit** (change request to a value that lies within the "absolute" limits, but is however below the currently valid lower limit)
C9 hex	**Change request above the currently valid limit** (example: a parameter value is too large for the inverter power)
CC hex	**Change request not permitted** (change is not permitted as the access code is not available)

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- 7 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
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 Control Unit in the section "Interconnecting signals in the inverter".

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 (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 (page index): = 90 hex** (offset 6000 ≙ 90 hex)
- **IND, bit 0 … 7 (subindex): = 2** (Index of the 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 ... 10	9 ... 0
AK		Parameter number	Page index	Subindex	Parameter value	Drive Object	Index
0 1 1 0	0	1 1 1 0 0 1 1 0 0 0 1	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 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

Figure 5-3 Telegram for a read request from p7841[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 5-4 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 ≙ 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 5-5 Telegram, to assign DI 2 with ON/OFF1

5.3.5 USS process data channel (PZD)

Description

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

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

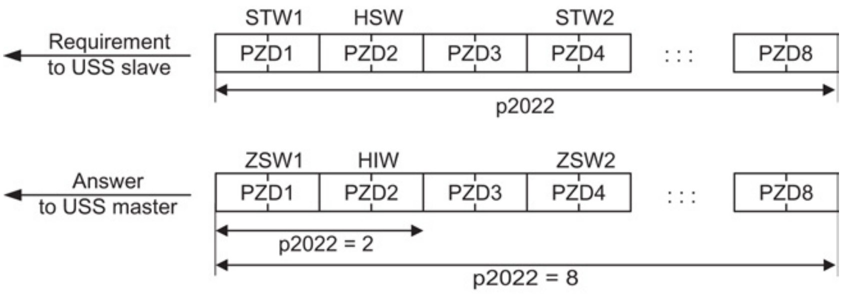


Figure 5-6 Process data channel

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).

You define the sources of the PZD using parameter p2051.

For further information, please refer to the List Manual.

5.3.6 Time-out and other errors

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

Table 5- 8 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.

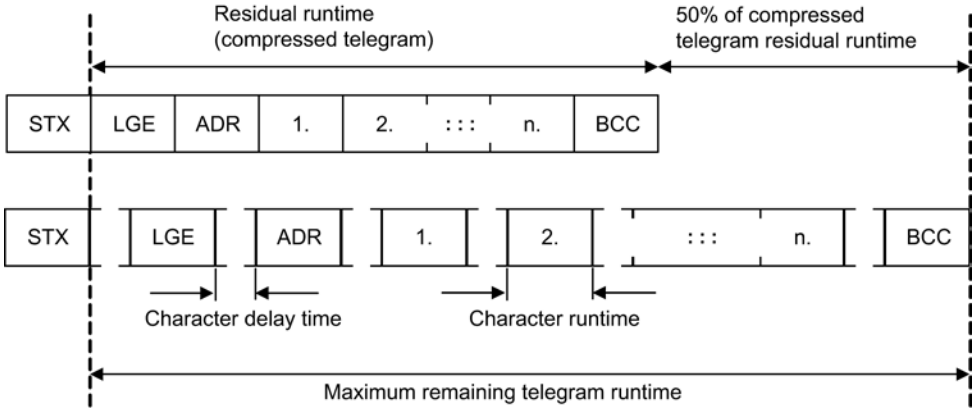


Figure 5-7 Telegram runtime as the sum of the residual runtime and character delay times

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 slave only responds after the response delay has expired.

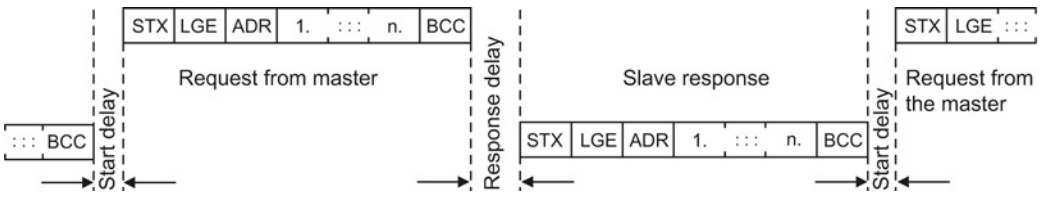


Figure 5-8 Start delay and response delay

The duration of the start delay must at least be as long as the time for two characters and depends on the baud rate.

Table 5- 9 Duration of the 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
38400	0.286 ms	> 0.573 ms
57600	0.191 ms	> 0.382 ms
115200	0.095 ms	> 0.191 ms

Note: 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 slave 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 slave

Telegram monitoring of the inverter

The inverter 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 inverter 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 inverter checks bit 10 of the received control word 1. If the bit is not set when the motor is switched on ("Operation"), the inverter responds with fault F07220.

5.4 Communication using Modbus RTU

Overview of communication using Modbus

The Modbus protocol is a communication protocol with linear topology based on a master/slave architecture.

Modbus offers three transmission modes:

- **Modbus ASCII**
  Data in ASCII code. The data throughput is lower compared to RTU.
- **Modbus RTU (RTU: Remote Terminal Unit)**
  Data in binary format. The data throughput is greater than in ASCII code.
- **Modbus TCP**
  Data as TCP/IP packets. TCP port 502 is reserved for Modbus TCP. Modbus TCP is currently undergoing definition as a standard (IEC PAS 62030 (pre-standard)).

The Control Unit supports Modbus RTU as a slave with even parity.

General information about communication using Modbus RTU

- Communication using Modbus RTU takes place over the RS485 interface with a maximum of 247 slaves.
- The maximum cable length is 100 m.
- Two 100 kΩ resistors are provided to polarize the receive and send cables.

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 particular inverter, you have the following options when setting communication via Modbus RTU:

- Default setting 21 "USS Fieldbus" for all inverters with RS485 interface
- Default setting 109 "BT Mac 9: Modbus RTU Fieldbus" only for CU230P-2 HVAC / CU230P-2 BT. See also Additional manuals for your inverter (Page 182).

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 STARTER under Control Unit/Configuration "Default settings of the setpoint/command sources":
     21: USS fieldbus
   - with BOP-2 for the basic commissioning under Step "MAc PAr P15":
     FB USS
   - using the expert list or parameter number:
     p0015 = 21
2. Set the bus protocol
   - using STARTER under Control Unit/Communication/Fieldbus:
     2 Modbus
   - using the BOP-2 via parameter p2030:
     p2030 = 2
3. Set the inverter address.
4. Make additional changes based on the parameters listed in the following section.
5. If you are working with STARTER, backup the settings with  .

This means that you have made the settings for communication via Modbus.

Parameters to adapt the settings for communication via Modbus RTU

Fieldbus telegram selection p2030 = 2 (Modbus)

Baud rate p2020 = 7, 19200 bit/s

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

Fieldbus analog outputs **p0791[0 … 1]**

Parameter to interconnect the analog outputs for control via the fieldbus

Modbus timing p2024[0 … 2]

(see Section "Baud rates and mapping tables (Page 97)")

- **p2024[0]: Maximum slave telegram processing time:**
  The time after which the slave must have sent a response to the master.
- **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[20]: Inter-telegram delay:**
  maximum permissible time between Modbus telegrams (Modbus standard processing time for 3.5 bytes).

Fieldbus monitoring time p2040 = 100 ms

Setting range: 0 ms … 1999999 ms

The more slaves 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 inverter shuts down with fault F01910.

p2040 = 0 ⇒ bus monitoring deactivated.

Fieldbus error statistics **r2029**

Displaying receive errors at the fieldbus interface

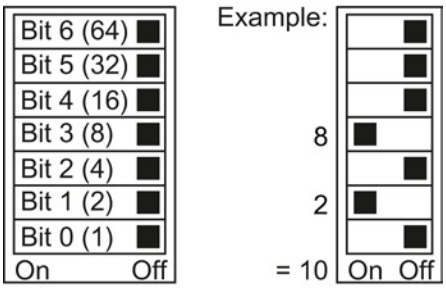
5.4.1.1 Setting the address

You set the bus address of the inverter using the address switches on the Control Unit, using parameter p2021 with the BOP-2 or in STARTER.

Using parameter p2021 (factory setting: 1) or using STARTER, you can only set the address, if all address switches are set to "OFF" (0).

Valid address range: 1 … 247

If you have specified a valid address using the address switches, this address will always be the one that takes effect and parameter p2021 cannot be changed.



Only addresses in the range 1 … 127 can be set via the address switches. If you require an address in the range 128 … 247, set the address switches to 0 and enter the address using p2021.

The position of the address switches can be found in the operating instructions of the Control Unit in the section "Overview of interfaces".

Procedure

To change the bus address, proceed as follows:

1. Set the new address:
   - using the address switches
   - from an operator panel in parameter p2021
   - in STARTER using screen form "Control Unit/Communication/Fieldbus", or using the expert list in parameter p2021
2. Switch on the inverter power supply and, if available, the 24 V power supply for the Control Unit.
3. Switch on the voltages again after all LEDs at the inverter have gone dark.

You have so changed the bus address.

5.4.2 Modbus RTU telegram

Description

For Modbus, there is precisely one master and up to 247 slaves. The master always starts the communication. The slaves can only transfer data at the request of the master. Slave-to-slave communication is not possible. The Control Unit always operates as slave.

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

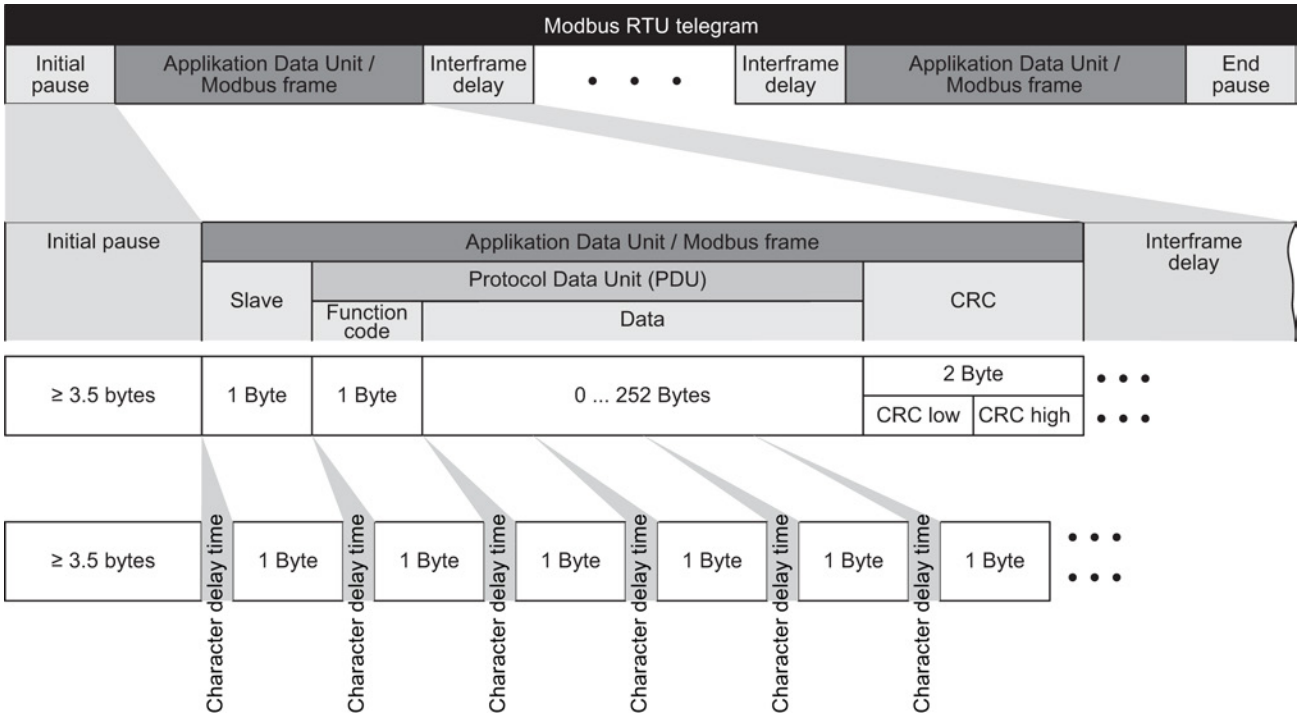


Figure 5-9 Modbus with delay times

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 cases:

- Start detection
- Between the individual frames
- End detection

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- 10 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 specifies the associated values depending on the protocol selection (p2030) or the baud rate.

Modbus register and Control Unit parameters

The Modbus protocol contains register or bit numbers for addressing memory. You must assign the appropriate control words, status words and parameters to these registers in the slave.

The converter supports the following addressing ranges:

Addressing range	Remark
40001 … 40065	Compatible with Micromaster MM436
40100 … 40522	

The valid holding register addressing range extends from 40001 to 40522. Access to other holding registers generates the fault "Exception Code".

The registers 40100 to 40111 are described as process data.

Note

R"; "W"; "R/W" in the column Modbus access stands for read (with FC03); write (with FC06); read/write.

Table 5- 11 Assigning the Modbus register to the parameters of the Control Unit

Modbus Reg. No.	Description	Mod-bus access	Unit	Scaling factor	On/Off text or value range		Data / parameter
Process data							
Control data							
40100	Control word	R/W	--	1			Process data 1
40101	Main setpoint	R/W	--	1			Process data 2
Status data							
40110	Status word	R	--	1			Process data 1
40111	Main actual value	R	--	1			Process data 2
Parameter data							
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
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]
Converter identification							
40300	Powerstack number	R	--	1	0 … 32767		r0200
40301	Converter firmware	R	--	0.0001	0.00 … 327.67		r0018
Converter data							
40320	Rated power of the power unit	R	kW	100	0 … 327.67		r0206
40321	Current limit	R/W	%	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	6.000 … 32767		p2000
Converter diagnostics							
40340	Speed setpoint	R	RPM	1	-16250 … 16250		r0020
40341	Actual speed value	R	RPM	1	-16250 … 16250		r0022

Modbus Reg. No.	Description	Mod-bus access	Unit	Scaling factor	On/Off text or 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	Actual current 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
Fault diagnostics						
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	Failure number, index 2	R	--	1	0 … 32767	r0947 [3]
40404	Failure number, index 3	R	--	1	0 … 32767	r0947 [4]
40405	Failure number, index 4	R	--	1	0 … 32767	r0947 [5]
40406	Failure number, index 5	R	--	1	0 … 32767	r0947 [6]
40407	Failure number, index 6	R	--	1	0 … 32767	r0947 [7]
40408	Alarm number	R	--	1	0 …32767	r2110 [0]
40499	PRM ERROR code	R	--	1	0 …99	--
Technology controller						
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
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
PID diagnostics						
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

5.4.4 Write and read access via FC 03 and FC 06

Function codes used

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

The Control Unit uses the Modbus function code 03, FC 03 (read holding registers) for reading, and the Modbus function code 06, FC 06 (preset single register) for writing.

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

Any valid register address is permitted as the start address.

The controller can access more than one register via FC 03 with a request. The number of addressed registers is contained in bytes 4 and 5 of the read request.

Table 5- 12 Invalid read requests

Read request	Inverter 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)

Table 5- 13 Structure of a read request for slave number 17

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

The response returns the corresponding data set:

Table 5- 14 Slave response to the read request

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

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

Start address is the holding register address.

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

Table 5- 15 Write request and response of the inverter

Write request	Inverter response
Incorrect address (a holding register address does not exist)	Exception code 02
Write to a "read-only" register	Modbus error telegram (exception code 04 - device failure)
Write to a reserved register	

If an incorrect address is entered (a holding register address does not exist), exception code 02 (invalid data address) is returned. An attempt to write to a "read-only" register or a reserved register is replied to with a Modbus error telegram (exception code 4 - device failure). In this instance, the detailed internal error code that occurred on the last parameter access via the holding registers can be read out via holding register 40499.

Table 5- 16 Structure of a write request for slave number 17

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

The response returns the register address (bytes 2 and 3) and the value (bytes 4 and 5) that was written by the higher-level controller to the register.

Table 5- 17 Slave response to the write request

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

5.4.5 Communication procedure

Procedure for communication in a normal case

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

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

Communication error

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

Logical error

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

Table 5- 18 Overview of exception codes

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

Maximum processing time, p2024[0]

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

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 slaves and the baud rate set on the bus.

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 inverter receives control commands and setpoints via service instructions from the control and transmits its status back to the control. The inverter can also send telegrams automatically itself, respectively 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

You will find the Protocol Implementation Conformance Statement (PICS) in the Internet under the following link: BACnet files (http://www.big-eu.org/catalog/siemens/SINAMICS%20PICS%20V4.6.pdf)

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

Overview

Procedure

Proceed as follows to set communication via BACnet:

1. Select the default setting 110.
   - With STARTER: under Control Unit/Configuration "Default settings of the setpoint/command sources":
     110 "BT Mac 10: BACnet MS/TP Fieldbus"
   - with BOP-2 for the basic commissioning under Step "MAc PAr P15":
     P_F bAc
   - using the expert list or parameter number:
     p0015 = 110
2. Set the inverter address.
3. Make additional changes based on the parameters listed in the following sections.
4. If you are working with STARTER, backup the settings with  .

This means that you have made the settings for communication via BACnet.

Settings using "BT Mac 10: BACnet MS/TP Feldbus"

Fieldbus telegram selection p2030 = 5

Baud rate p2020 = 6, 9600 bit/s

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

Fieldbus monitoring time p2040 = 100 ms

Setting range: 0 ms … 1999999 ms

The more slaves 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 inverter shuts down with fault F01910.

p2040 = 0 ⇒ bus monitoring deactivated.

Additional setting options

Additional parameters for adapting communication via BACnet:

Fieldbus analog outputs **p0791[0 … 1]**

Displays the value of the particular analog output as a percentage.

Processing times p2024[0 … 2]

p2024 [0]: 0 ms … 10000 ms, maximum processing time (APDU timeout), factory setting = 1000 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 = 1
- p2025 [2]: 0 … 39: Number of APDU Retries (repeated attempts after fault telegrams), factory setting = 3
- p2025 [3]: 1 … 127: Maximum Master address, factory setting = 127

Setting the COV_Increments p2026[0 … 74]

(COV = change of values) 0 … 4194303.000, factory setting = 1

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 for which inverter value changes an UnConfirmedCOV_Notification or ConfirmedCOV_Notification result is sent.

The factory setting 1 means that the inverter 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

Setting the address

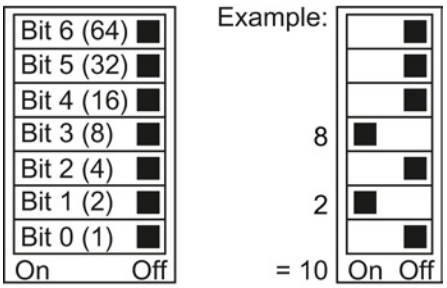
You set the MAC address of the inverter using the address switches on the Control Unit, using parameter p2021 or in STARTER.

Valid address range: 0 … 127.

For address 0, the inverter responds to a broadcast.

If you have specified a valid address ≠ 0 using the address switches, this address will always be the one that takes effect and parameter p2021 cannot be changed.

The position of the address switches can be found in the operating instructions of the Control Unit in the section "Overview of interfaces".



See also

Additional manuals for your inverter (Page 182)

Procedure

To change the bus address, proceed as follows:

1. Set the new address:
   - using the address switches
   - from an operator panel in parameter p2021
   - in STARTER using screen form "Control Unit/Communication/Fieldbus", or using the expert list in parameter p2021
2. Switch on the inverter power supply and, if available, the 24 V power supply for the Control Unit.
3. Switch on the voltages again after all LEDs at the inverter have gone dark.

You have so changed the bus address.

5.5.2 Supported services and objects

BIBBs used by the inverter

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

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

The CU230P-2 HVAC uses the BIBBs listed below.

Overview of the BIBBs used by CU230P-2 HVAC and 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 inverter 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 (AIxx),
- Analog Output (AOxx),
- Analog Value (AVxx),
- Binary Value (BVxx) and
- Multi-state Input (MSIxx)

Note

SubscribeCOV services are not retentive; i.e. the master must re-initiate the SubscribedCOV services when restarting the CU.

Code numbers of the object types supported in BACnet

Object type	Code number for BACnet object type		Object type	Code number for BACnet object type
Device Object	8		Analog Input Object	0
Binary Input Object	3		Analog Output Object	1
Binary Output Object	4		Analog Value Object	2
Binary Value Object	5		Multi-State Input Object	13

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 Master
• 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

Object properties of other object types

Object property	Object type						
	Binary Input	Binary Output	Binary Value	Analog Input	Analog Output	Analog Value	Multi-State Input
Object_Identifier	X	X	X	X	X	X	X
Object_Name	X	X	X	X	X	X	X
Object_Type	X	X	X	X	X	X	X
Present_Value	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X
Status_Flags	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*		X	X*	
Relinquish_Default		X	X*		X	X*	
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

* for command values only (access type C)

Note

Access types are available in the following versions

- C: Executable
- R: Readable
- W: Writable

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 1 - used as DI	ON/OFF	ON/OFF	R	r0722.11
BI8	DI8 ACT	State of AI 2 - used as DI	ON/OFF	ON/OFF	R	r0722.12
BI10	DO0 ACT	State of DO 0 (relay 1)	ON/OFF	ON/OFF	R	read r747.0
BI11	DO1 ACT	State of DO 1 (relay 2)	ON/OFF	ON/OFF	R	read r747.1
BI12	DO2 ACT	State of DO2 (relay 3)	ON/OFF	ON/OFF	R	read r747.2

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

Binary value Objects

Instance ID	Object name	Description	Possible values	Text active	Text inactive	Access type	Parameter
BV0	RUN / STOP ACT	Inverter 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	Inverter fault	FAULT / OK	FAULT	OK	R	r0052.3
BV3	WARN ACT	Inverter warning	WARN / OK	WARN	OK	R	r0052.7
BV4	MANUAL / AUTO ACT	Indicates the source of the manual/auto inverter control	AUTO / MANUAL	AUTO	LOCAL	R	r0052.9
BV7	CTL OVERRIDE ACT	ACT displays the inverter's control unit from the BACnet override control unit via BV93. Note that the operator panel's "Manual" operating mode has a higher priority than the BACnet override control.	ON/OFF	0	1	R	r2032[10]
BV8	AT SET-POINT	Setpoint reached	YES / NO	YES	NO	R	r0052.8

Instance ID	Object name	Description	Possible values	Text active	Text inactive	Access type	Parameter
BV9	AT MAX FREQ	Maximum speed reached	YES / NO	YES	NO	R	r0052.10
BV10	DRIVE READY	Inverter ready	YES / NO	YES	NO	R	r0052.1
BV15	RUN COM ACT	ACT indicates the status of the ON command, regardless of the source	YES / NO	0	1	R	r2032[0]
BV16	HIB MOD ACT	ACT means that the inverter is operating in energy-saving mode.	ON/OFF	0	1	R	r2399[1]
BV17	ESM MOD	ACT means that the inverter is operating in the essential service mode	ON/OFF	0	1	R	r3889[0]
BV20	RUN / STOP CMD	ON command for the inverter (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	Local / Remote	Local / Remote	YES	NO	C	r0054.15
BV26	RUN ENA CMD	Enable inverter operation		ENABLED	DISABLED	C	r0054.3
BV27	OFF2	OFF2 status	RUN / STOP	0	1	C	r0054.1
BV28	OFF3	OFF3 status Note: 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 0 controller	ENABLED / DISABLED	ENABLED	DISABLED	C	p11000
BV52	ENABLE PID 1	Enable technology 1 controller	ENABLED / DISABLED	ENABLED	DISABLED	C	p11100
BV53	ENABLE PID 2	Enable technology 2 controller	ENABLED / DISABLED	ENABLED	DISABLED	C	p11200
BV90	LOCAL LOCK	Use MANUAL (operator panel) to lock inverter control		LOCK	UNLOCK	C	p0806
BV93	CTL OVERRIDE CMD	Inverter control using BACnet override control	ON/OFF	0	1	C	r0054.10

Analog Input Objects

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
AI0	ANALOG INPUT 0	AI0 input signal	V/mA	inverter-dependent	R	r0752[0]
AI1	ANALOG INPUT 1	AI1 input signal	V/mA	inverter-dependent	R	r0752[1]
AI2	ANALOG INPUT 2	AI2 input signal	V/mA	inverter-dependent	R	r0752[2]
AI3	ANALOG INPUT 3	AI3 input signal	V/mA	inverter-dependent	R	r0752[3]
AI10	ANALOG INPUT 0 SCALED	Scaled AI 0 input signal	%	inverter-dependent	R	r0755[0]
AI11	ANALOG INPUT 1 SCALED	Scaled AI 1 input signal	%	inverter-dependent	R	r0755[1]
AI12	ANALOG INPUT 2 SCALED	Scaled AI 2 input signal	%	inverter-dependent	R	r0755[2]
AI13	ANALOG INPUT 3 SCALED	Scaled AI 3 input signal	%	inverter-dependent	R	r0755[3]

Analog Output Objects

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
AO0	ANALOG OUTPUT 0	AI0 value	%	inverter-dependent	C	p0791.0
AO1	ANALOG OUTPUT 1	AI1 value	%	inverter-dependent	C	p0791.1

Analog Value Objects

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
AV0	OUTPUT FREQ_Hz	Output frequency (Hz)	Hz	inverter-dependent	R	r0024
AV1	OUTPUT FREQ_PCT	Output frequency (%)	%	inverter-dependent	R	HIW
AV2	OUTPUT SPEED	Motor speed	RPM	inverter-dependent	R	r0022
AV3	DC BUS VOLT	DC-link voltage.	V	inverter-dependent	R	r0026
AV4	OUTPUT VOLT	Output voltage	V	inverter-dependent	R	r0025
AV5	CURRENT	Motor current	A	inverter-dependent	R	r0027
AV6	TORQUE	Motor torque	Nm	inverter-dependent	R	r0031
AV7	POWER	Motor power	kW	inverter-dependent	R	r0032
AV8	DRIVE TEMP	Heat sink temperature	°C	inverter-dependent	R	r0037
AV9	MOTOR TEMP	Measured or calculated motor temperature	°C	inverter-dependent	R	r0035
AV10	KWH (NR)	Cumulative inverter energy consumption (cannot be reset!)	kWh	inverter-dependent	R	r0039
AV12	INV RUN TIME (R)	Motor's operating hours (is reset by entering "0")	h	0 … 4294967295	W	p0650
AV13	INV Model	Code number of Power Module	---	inverter-dependent	R	r0200
AV14	INV FW VER	Firmware version	---	inverter-dependent	R	r0018
AV15	INV POWER	Rated power of the inverter	kW	inverter-dependent	R	r0206

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
AV16	RPM STPT 1	Reference speed of the inverter	RPM	6.0 … 210000	W	p2000
AV17	FREQ STPT 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	---	inverter-dependent	R	r0947[0]
AV19	PREV FAULT 1	Number of the last fault	---	inverter-dependent	R	r0947[1]
AV20	PREV FAULT 2	Number of the fault before last	---	inverter-dependent	R	r0947[2]
AV21	PREV FAULT 3	Number of the fault third from last	---	inverter-dependent	R	r0947[3]
AV22	PREV FAULT 4	Number of the fault fourth from last	---	inverter-dependent	R	r0947[4]
AV25	Select Setpoint Source	Command to select the setpoint source	---	0 … 32767	W	p1000
AV28	AO1 ACT	Signal from AO 1	mA	inverter-dependent	R	r0774.0
AV29	AO2 ACT	Signal from AO 1	mA	inverter-dependent	R	r0774.1
AV30	MIN SPEED	Minimum speed	RPM	0.000 – 19500.000	W	p1080
AV31	MAX FREQ	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	inverter-dependent	R	p0640
AV39	ACT WARN	Indication of a pending alarm	---	inverter-dependent	R	r2110[0]
AV40	PREV WARN 1	Indication of the last alarm	---	inverter-dependent	R	r2110[1]
AV41	PREV WARN 2	Indication of the last but one alarm	---	inverter-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
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

Instance ID	Object name	Description	Unit	Range	Access type	Parameter
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	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
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 constant	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

Multi-State Input Objects

Instance ID	Object name	Description	Possible values	Access type	Parameter
MSI0	FAULT_1	Fault number 1	See List Manual "List of faults and alarms"	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	Warning number 1		R	r2110[0]
MS9	WARNING_2	Warning number 2		R	r2110[1]
MSI10	WARNING_3	Warning number 3		R	r2110[2]
MSI11	WARNING_4	Warning number 4		R	r2110[3]
MSI12	WARNING_5	Warning number 5		R	r2110[4]
MSI13	WARNING_6	Warning number 6		R	r2110[5]
MSI14	WARNING_7	Warning number 7		R	r2110[6]
MSI15	WARNING_8	Warning number 8		R	r2110[7]

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

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

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

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

Settings in the controller

In the Field Cabinet, for each slave 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.

Overview

Procedure

Proceed as follows to set communication via P1:

1. Select the default setting 114.
   - With STARTER: under Control Unit/Configuration "Default settings of the setpoint/command sources":
     114 "BT Mac 14: Communication P1"
   - with BOP-2 for the basic commissioning under Step "MAc PAr P15":
     P_F _P1
   - using the expert list or parameter number:
     p0015 = 114
2. Set the inverter address.
3. Make additional changes based on the parameters listed in the following sections.
4. If you are working with STARTER, backup the settings with  .

This means that you have made the settings for communication via P1.

Settings using "BT Mac 14: Kommunikation P1"

Fieldbus telegram selection p2030 = 8

Baud rate p2020 = 5: 4800 bit/s
Additional setting option: p2020 = 7 (19200 bits/s)

ON/OFF1: p0840 = 2090.0
The ON/OFF1 command is connected to control word 1 bit 0.

Enable operation: p0852 = 2090.3
The signal for "Enable operation" is connected to control word 1 bit 3

Signal for acknowledging errors: p2103.0 = 2090.7
The signal for "Acknowledge error" is connected to control word 1 bit 7

Additional setting options

Additional parameters for adapting communication via P1:

Main setpoint p1070 = 2050.1

Connect the signal for the **main setpoint** to the communication interface

Sending process data

- Send status word via the communication interface p2051.0 = 52.0
- Send speed actual value via the communication interface p2051.1 = 63.0

Note

Address 99 (to all)

No matter their address, all FLN devices answer telegrams with address 99.

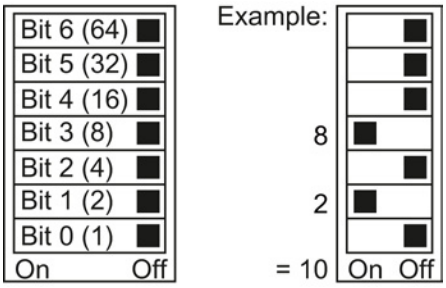
Setting the address

You set the bus address of the inverter using the address switches on the Control Unit, using parameter p2021 or in STARTER.

Valid address range: 1 … 99

If you have specified a valid address using the address switches, this address will always be the one that takes effect, and parameter p2021 (factory setting: 99) will not be able to be changed.

The position of the address switches can be found in the operating instructions of the Control Unit in the section "Overview of interfaces".



See also

Additional manuals for your inverter (Page 182)

Procedure

To change the bus address, proceed as follows:

1. Set the new address:
   - using the address switches
   - from an operator panel in parameter p2021
   - in STARTER using screen form "Control Unit/Communication/Fieldbus", or using the expert list in parameter p2021
2. Switch on the inverter power supply and, if available, the 24 V power supply for the Control Unit.
3. Switch on the voltages again after all LEDs at the inverter have gone dark.

You have so changed the bus address.

Overview

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 on CAN in the CAN Internet pages (http://www.can-cia.org); you can obtain an explanation of CAN terminology in the CANdictionary under CAN downloads (http://www.can-cia.org/index.php?id=6).

Integrating an inverter in a CANopen network

To integrate an inverter in a CANopen network, we recommend the EDS file on the Internet (http://support.automation.siemens.com/WW/view/en/48351511). This file is the description file of the SINAMICS G120 inverter for CANopen networks. In this way, you can use the objects of the DSP 402 device profile.

The following Control Units and inverters have an CANopen interface

G120 CU230P-2 CAN
G120 CU250S-2 CAN

G120C CAN

Table 6- 1 Pin assignment of the connector

Signal	X126 (D Sub connector, IP20) 1 ... 5 6 ... 9
---	1
CAN_L, CAN signal (dominant low)	2
CAN_GND, CAN ground	3
---	4
(CAN _SHLD), optional shield	5
(GND), optional ground	6
CAN_H, CAN signal (dominant high)	7
---	8
---	9

CANopen functions of the inverter

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

SINAMICS G120 inverters 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 inverter operates with the following communication objects:

- **NMT** Network management (NMT service) (Page 125)
  Network management objects for controlling CANopen communication and for monitoring the individual nodes on the basis of a master-slave relationship
- **SDO** SDO services (Page 128)
  Service data objects for reading and changing parameters
- **PDO** PDO services (Page 132)
  Process data objects to transfer process data; TPDO to send, RPDO to receive
- **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 + NAlleode-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

*) In the "Predefined Connection Set", see SectionPredefined connection set (Page 135) .

6.1 Network management (NMT service)

Network management (NMT) is node-oriented and has a master-slave topology.

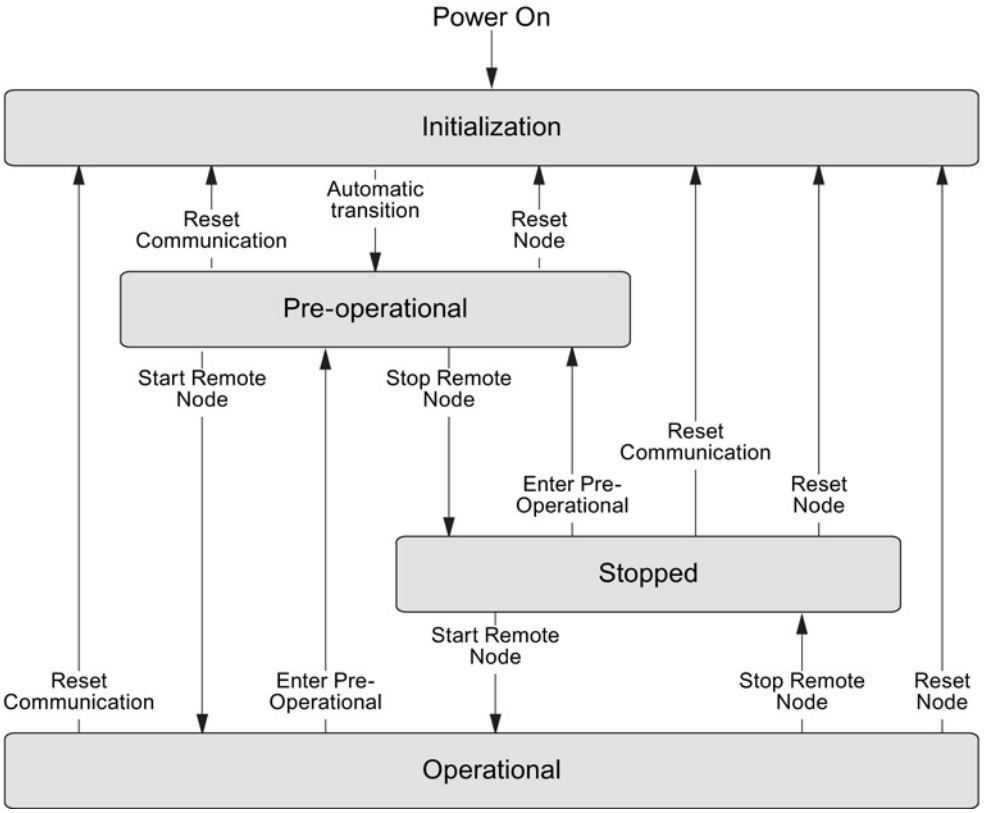
A node is a master or a slave.

The inverter is an NMT slave, and can adopt the following states:

- Boot-up service COB ID = 700hex + 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 at CiA 301 (Application Layer and Communication Profile) (http://www.can-cia.org/index.php?id=specifications).
- Error Control Service COB ID = 700hex + node ID

CANopen state diagram



NMT states

The current state of the node is displayed via p8685. It can also be changed directly using this parameter:

- Initializing, display via p8685 = 0
  The inverter initializes itself after power on. In the factory setting, the inverter 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 inverter does not go into the "Pre-Operational" state, but instead, into the "Stopped" or "Operational" state.
- Pre-operational display/change using p8685 = 127 (factory setting)
  In this state, the node cannot process any process data (PDO). However, the controller can use SDO parameters to change or operate the inverter, which means that you can also enter setpoints via SDO.
- Operational display/change using p8685 = 5
  In this state, the node can process both SDO and PDO.
- Stopped, display/change using p8685 = 4
  In this state, the node can neither process SDO nor PDO. The "Stopped" state terminates one of the following commands:
  - Enter pre-operational, p8685 = 127 (factory setting)
  - Start Remote Node
  - Reset node, p8685 = 128
  - Reset communication, p8685 = 129

You can also change the NMT state in STARTER via "Control_Unit / Communication / CAN" under the "Network-Management" tab.

Request sent by the master to one or several slaves

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

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

Boot-up Service

The boot-up protocol indicates the state of the NMT slave 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

Using parameter p8684, you can set that after powering up, the inverter goes into one of the following states:

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 inverter, 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 inverter 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 inverter 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, see Setting the monitoring of the communication (Page 158)

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 inverter parameter - offset value) hex + 2000 hex

Examples of object numbers

Parameter	Number of the inverter 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 inverter parameters

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

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

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

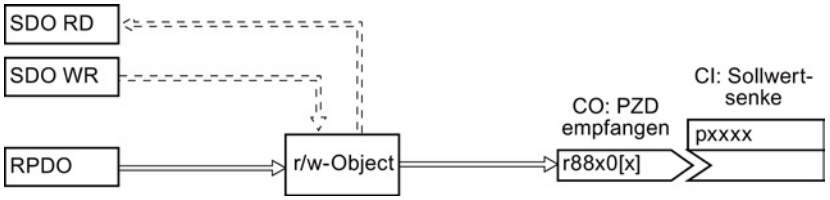


Figure 6-1 Access to mapped PZD setpoint objects

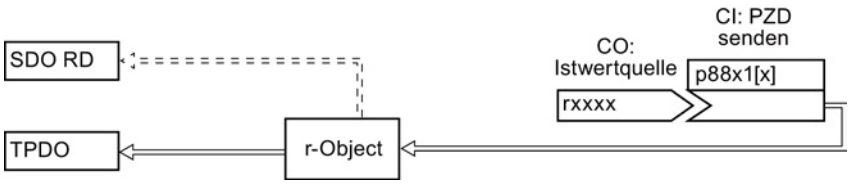


Figure 6-2 Access to mapped PZD actual value objects

Example, access to object 6042 hex

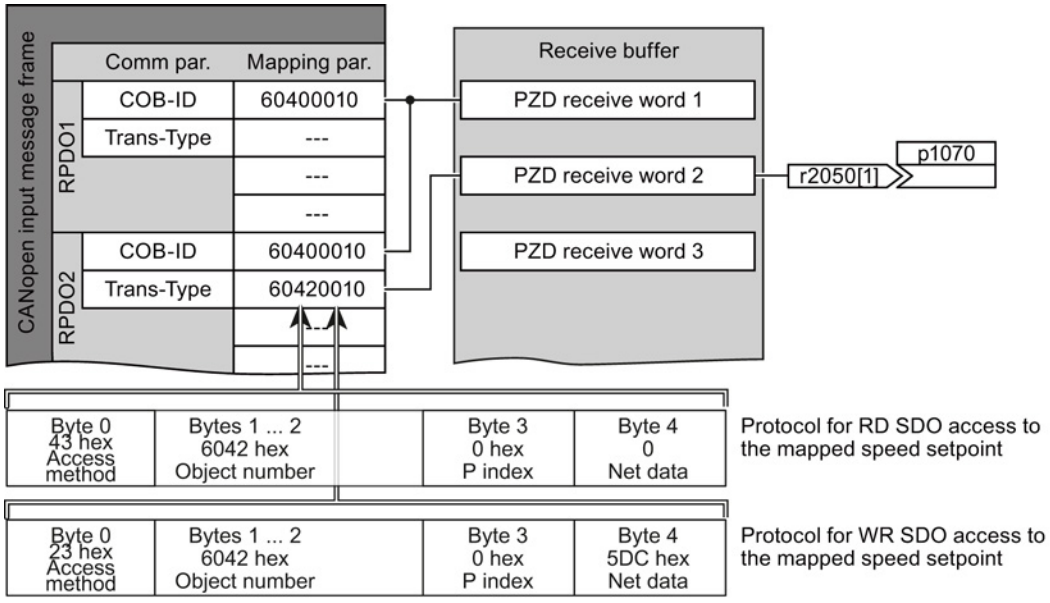


Figure 6-3 Access to the process data

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

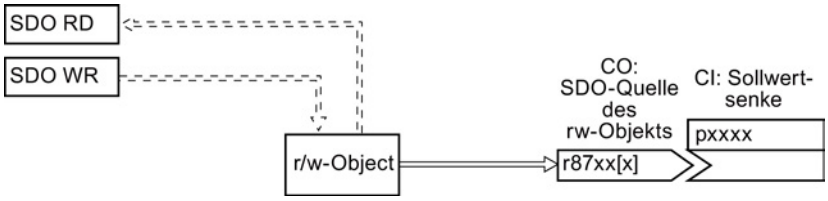


Figure 6-4 Access to non-mapped PZD setpoint objects

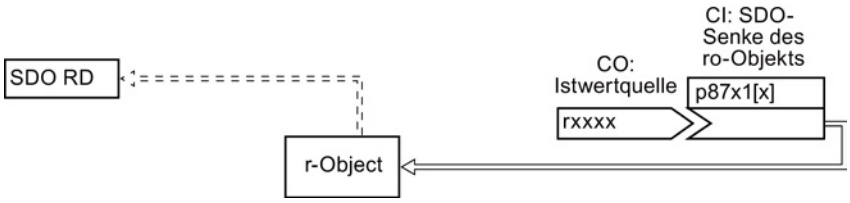


Figure 6-5 Access to non-mapped free PZD actual value objects

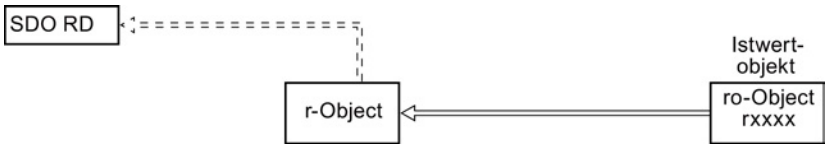


Figure 6-6 Access to non-mapped standardized PZD actual value objects

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 inverter 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. You can use Free PDO mapping (Page 136) or the Predefined connection set (Page 135) to do this.

Parameter area for PDO	RPDO		TPDO	
	In the inverter	In CANopen	In the inverter	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.

The values for communication parameters can be found in the tables in Section Object directories (Page 144)

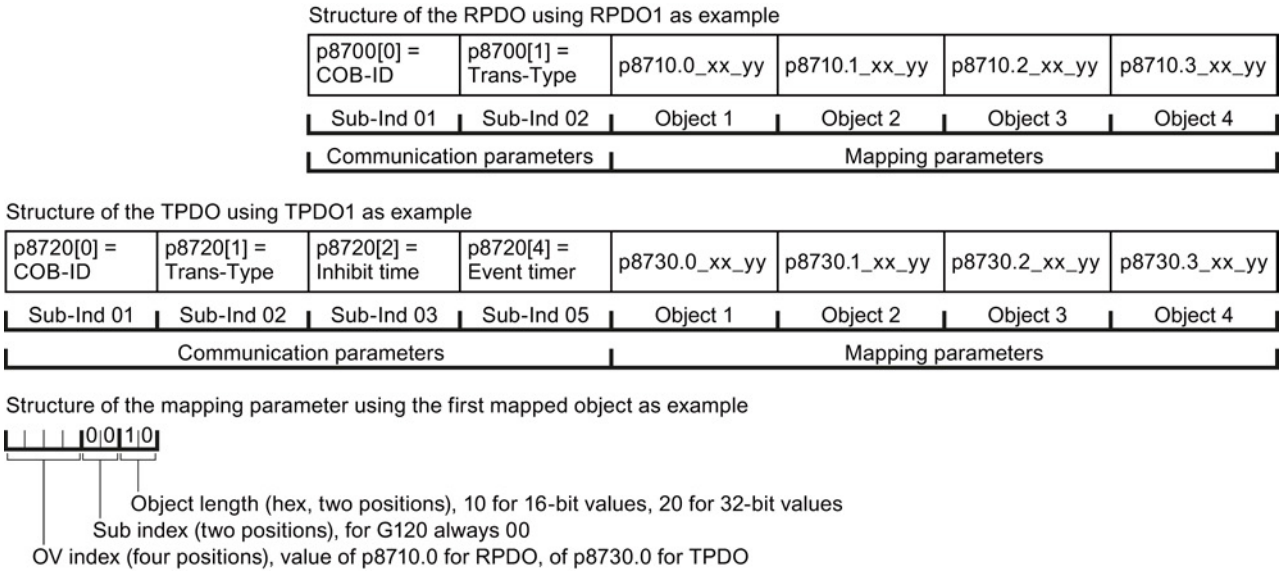


Figure 6-7 Structure of the RPDO and TPDO communication objects

COB ID

An overview of the COB IDs can be found in Section Communication over CANopen (Page 123). Section Predefined connection set (Page 135) explains how the COB IDs are calculated.

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 inverter:

- Cyclic synchronous (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:

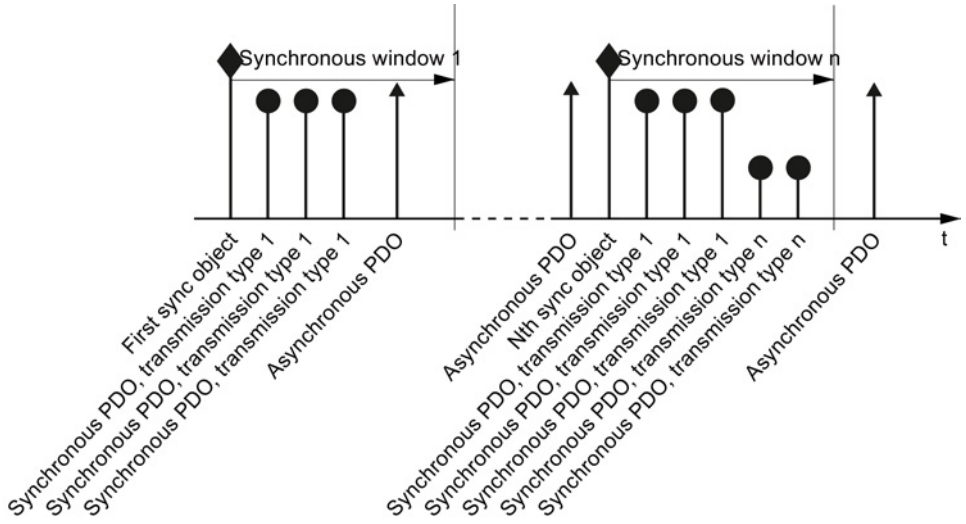


Figure 6-8 Principle of synchronous and asynchronous transmission

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 inverter.

Note

The SYNC signal synchronizes only the communication on the CANopen bus and not functions in the inverter, 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 inverters support the PDO Write protocol

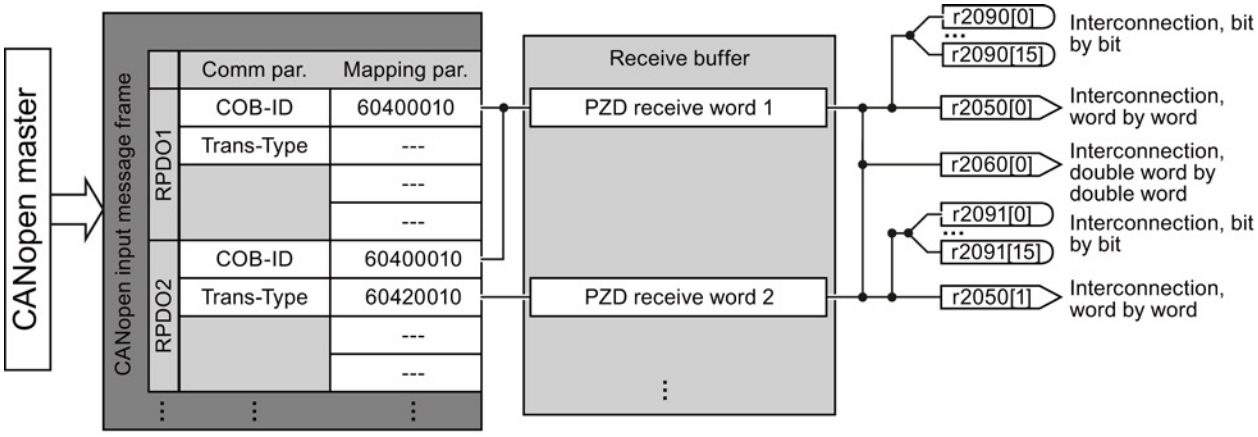
Write PDO

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 inverter using the factory setting in CANopen, the inverter receives the control word and the speed setpoint from the controller. The inverter 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

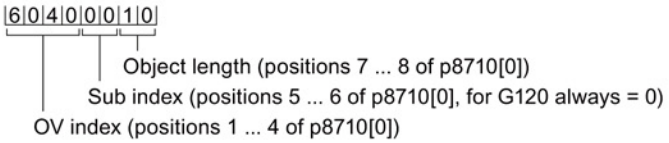
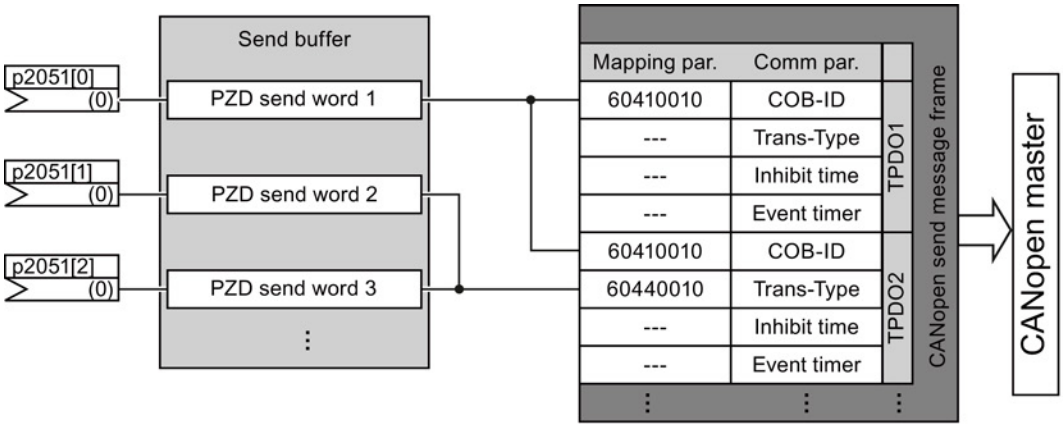


Figure 6-9 RPDO mapping with the Predefined Connection Set



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

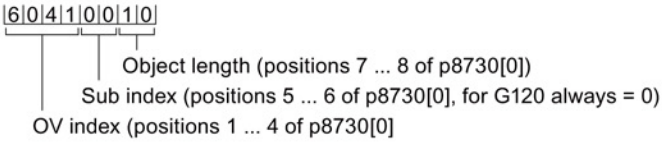


Figure 6-10 TPDO mapping with the Predefined Connection Set

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, or
- as objects of drive profile CiA 402, corresponding to the requirements of your system for the PDO service

This requires that the inverter is set to a free PDO mapping (p8744 = 2) (factory setting).

Configuring and mapping process data using free PDO mapping

Procedure

Proceed as follows to configure and map process data:

1. Define process data,
   examples:
   - Send actual current value (r0068) from the inverter to the controller (TPDO - Transmit Process Data Object)
   - Send additional speed setpoint from the controller to the inverter (RPDO - Receive Process Data Object) and write in p1075
2. Specify objects for transmission of the process data
   - TPDO1 for the actual current value
   - RPDO1 for additional speed setpoint
3. Set communication parameters for RPDO and TPDO
   - Define the communication parameters for RPDO, see RPDO communication parameters (Page 146)
   - Define the communication parameters for TPDO, see TPDO communication parameters (Page 149)
4. Select the OD index for the mapping parameters:
   - Mapping parameters for RPDO, see RPDO mapping parameters (Page 147)
   - Mapping parameters for TPDO, see TPDO mapping parameters (Page 151)
5. Write OV index into the SINAMICS mapping parameters
   - p8710 … p8717 for RPDO
   - p8730 … p8737 for TPDO
   - You will find the OD index in the free objects (Page 153) or in the objects of the CiA 402 drive profile (Page 154)

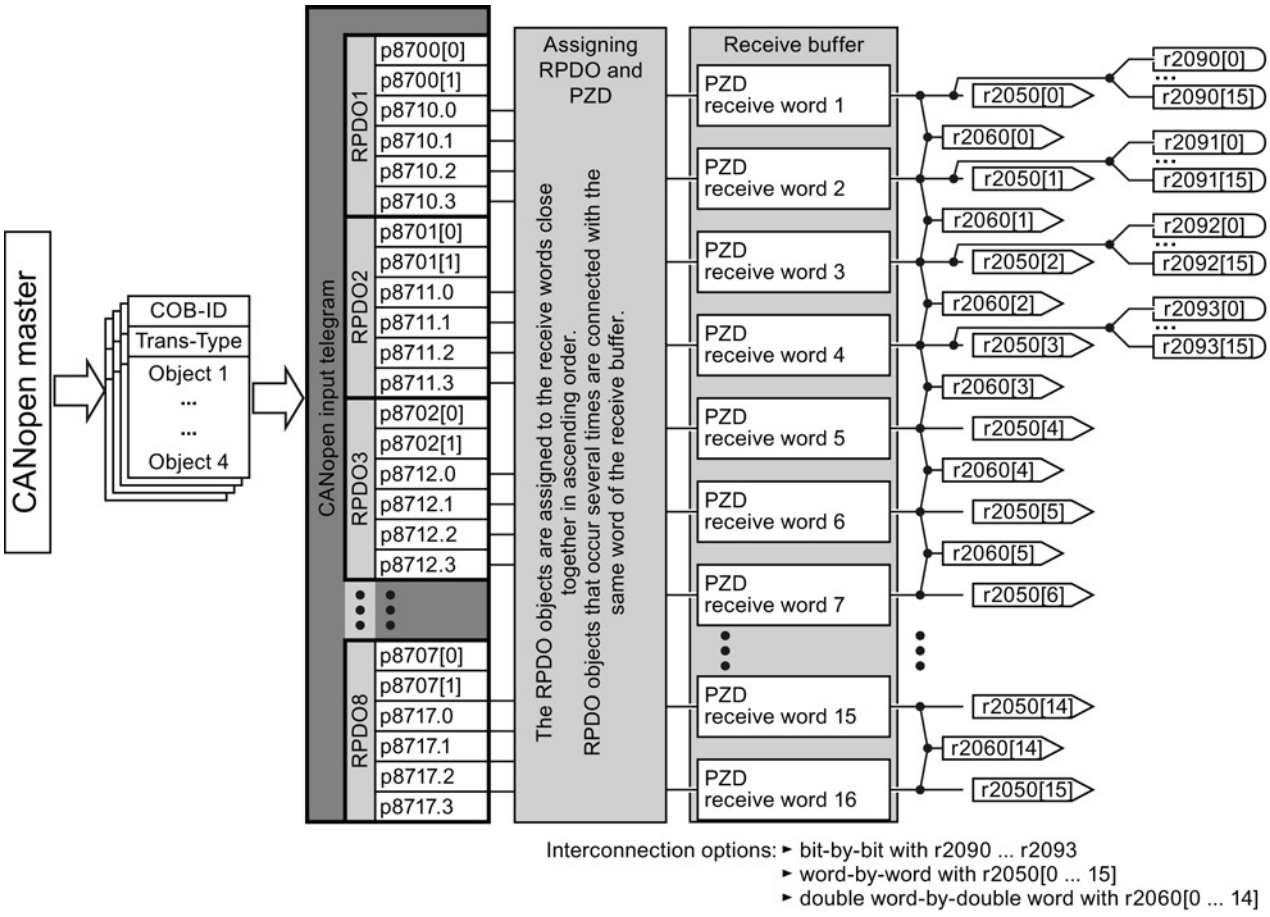
Note

Requirement for changing the OD indexes of the SINAMICS mapping parameters

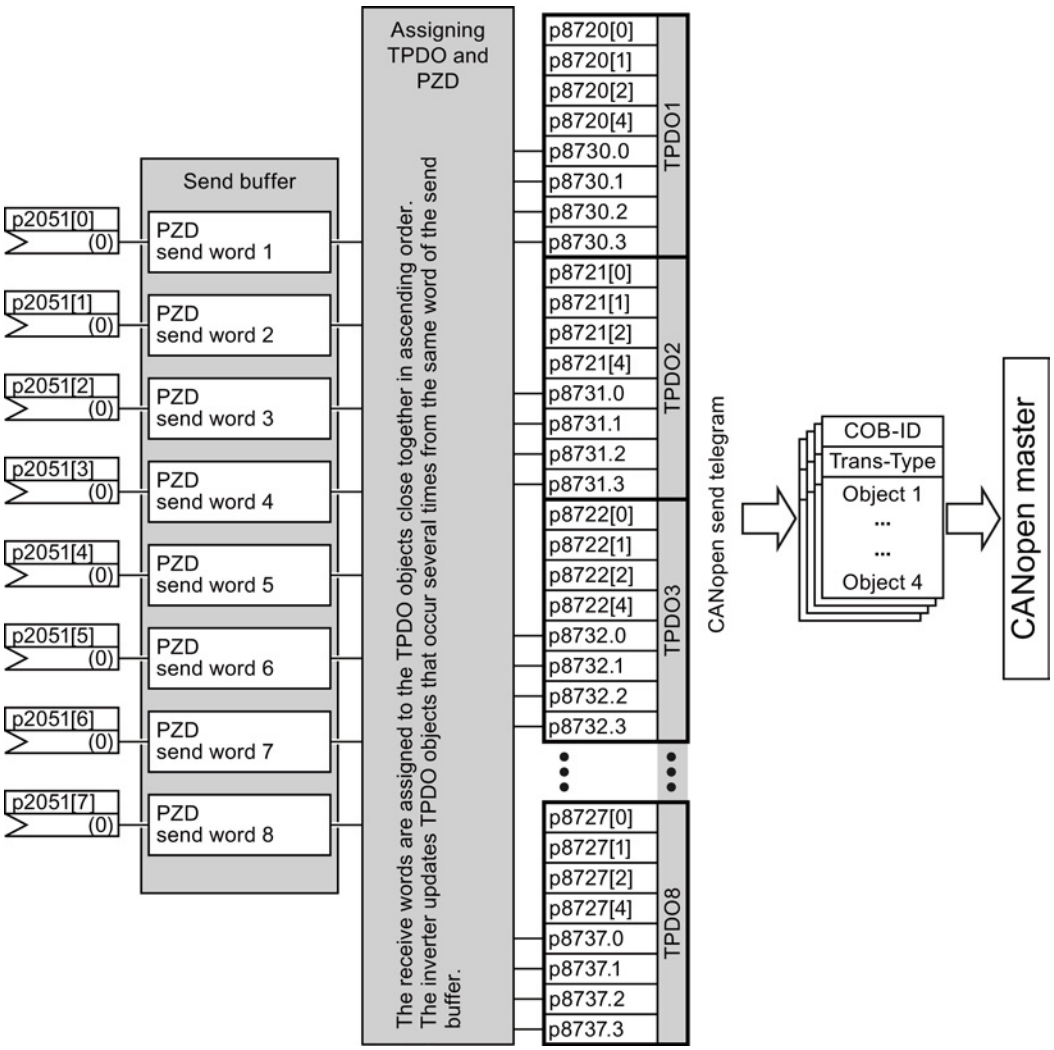
To allow you to change the values of the mapping parameters, you must set the COB ID of the corresponding parameter to invalid. Add the value 80000000 hex to the COB ID. If you have changed the mapping parameter, you must reset the COB ID to the valid value.

- You have configured and mapped the process data.

Free RPDO mapping - Overview



Free TPDO mapping - Overview



See also

General objects from the CiA 301 communication profile (Page 144)

General objects from the CiA 301 communication profile (Page 144)

6.3.3 Interconnect objects from the receive and transmit buffers

Procedure

Proceed as follows to interconnect process data:

1. Create a telegram:
   create PDO (parameterize the PDO Com. Parameters and PDO mapping parameters), see Predefined connection set (Page 135) and Free PDO mapping (Page 136)
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 interconnected the process data.

Interconnecting the receive buffer

The inverter writes the received data in 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 inverter sends 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

Procedure

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

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]

This means you have transferred the actual current value into the communication as the send object.

Mapping the torque limit (p1520) with RPDO1

Procedure

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

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]

This means you have transferred the value for the torque limit into the communication.

6.4 CANopen operating modes

The inverter has the following CANopen operating modes

CANopen operating mode			SINAMICS				
Active operating mode	**Set in 6060 hex 1)**	**Displays the active operating mode in 6502 hex**	**Open-loop/closed-loop control mode**	**Control Unit/inverter**			**Value in p1300**
	Value			**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 of		
		velocity mode	Profile velocity mode	Profile torque mode
Up	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

You save the parameter changes in the EEPROM of the inverter using the CANopen object 1010. This function is identical with the data save function using the button (RAM -> ROM) in STARTER or via p0971 = 1.

Saving data using the CANopen object 1010 is not possible when an MMC is inserted, the inverter is connected with a computer via the USB connector, but STARTER is not accessing the inverter online.

Withdraw the USB cable in order to be able to save data using CANopen object 1010.

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

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written
1027		Module list					
	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

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	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
	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

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
1602		**Receive PDO 3 mapping parameter**				
	0	Number of mapped application objects in PDO		U8	0	r
	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

OD index (hex)	Sub-index (hex)	Name of the object	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written to
	2	PDO mapping for the second application object to be mapped	p8716.1	U32	0	r/w
	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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written
	5	Event timer	p8721.4	U16	0	r/w
1802		**Transmit PDO 3 communication parameter**				
	0	Largest subindex supported		U8	5	r
	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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written
	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)	Sub-index (hex)	Object name	SINAMICS parameters	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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written
	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
	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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Data type	Predefined connection set	Can be read/ written
	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 the process data of the free objects:
  - 16-bit (word): 4000 hex ≙ 100%
  - 32-bit (doubleword) 4000000 hex ≙ 100%
  - For temperature values: 16-bit (word): 4000 hex ≙ 100° C
  - For temperature values: 32-bit (doubleword): 4000000 hex ≙ 100° C

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 values	Can be read/ written	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 inverter.

OD index (hex)	Subindex (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]							

OD index (hex)	Subindex (hex)	Name of the object	SINAMICS parameters	Transmission	Data type	Default setting	Can be read/ written
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
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 inverter can process the objects from the Profile Torque Mode. But they cannot be set nor selected in inverters.

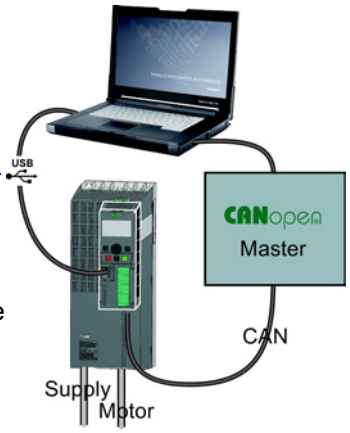
6.7 Integrating the inverter into CANopen

Commissioning

Precondition

- STARTER ≥ version 4.2, is installed on the computer used to commission the system.
- The inverter is connected to a CANopen master.
- The EDS (Electronic Data Sheet) (http://support.automation.siemens.com/WW/view/en/48351511)is installed on your CANopen master.
- In the basic commissioning you have set the inverter interfaces to the CANopen fieldbus.

  This means that the following signals in the inverter are interconnected corresponding to the Predefined Connection Sets:
  - Speed setpoint and control word
  - Speed actual value and status word



Procedure

Proceed as follows to commission the CANopen interface:

1. Connecting inverter to CAN bus (Page 157)
2. Set the node ID, baud rate and the monitoring of the communication.
   Also see "Setting the node ID and baud rate (Page 157)" as well as "Setting the monitoring of the communication (Page 158)"
3. Interconnecting additional process data
   Set p8744 = 2. You can now interconnect additional process data.

   See also "Free PDO mapping (Page 136)"
4. Signal interconnection of the links created in free PDO mapping.

   See also Interconnect objects from the receive and transmit buffers (Page 139).
5. Exiting commissioning
   Select the "Operational" status in the "Network management" tab in the "Control Unit/Communication/CAN" STARTER screen form and exit the commissioning.

You have commissioned the CANopen interface.

More information about configuring the communication is provided in Sections Communication over CANopen (Page 123) and Object directories (Page 144).

6.7.1 Connecting inverter to CAN bus

Connect the inverter 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 inverter forms the first or last slave 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

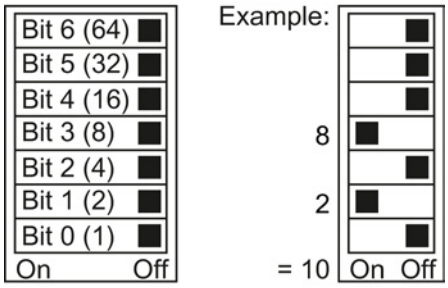
Setting the node ID

You set the node ID of the inverter using the address switches on the Control Unit, using parameter p8620 or in STARTER.

Valid address range: 1 … 127

If you have specified a valid node ID using the address switches, this node ID will always be the one that takes effect, and parameter p8620 (factory setting: 0) will not be able to be changed.

The position of the address switches can be found in the operating instructions of the Control Unit in the section "Overview of interfaces".



Procedure

To change the bus address, proceed as follows:

1
2

- Set the address using one of the subsequently listed options:
  - using the address switches
  - with an operator panel using p8620
  - in STARTER using screen form "Control Unit/Communication/Fieldbus" or using the expert list with p8620 and - only for STARTER - save the settings using (RAM to ROM)

1. Switch on the inverter power supply and, if being used, the 24 V supply for the Control Unit.
2. Switch on the voltages again after all LEDs at the inverter have gone dark.

This means that you have changed the bus address.

Setting the data transfer rate

You set the data transfer rate using parameter p8622 or in the STARTER "Control Unit/Communication/CAN" screen form under the "CAN interface" tab. When you are working with STARTER, you must also save the setting with (RAM to ROM).

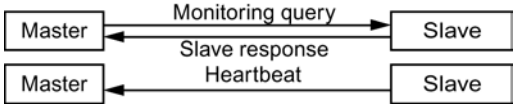
Setting range: 10 kbit/s … 1 Mbit/s. The maximum permissible cable length for1 Mbit/s is 40 m.

In order that a newly set or modified data transfer rate becomes effective, switch on the inverter power supply and, if being used, the 24 V supply for the Control Unit. Switch on the voltages again after all LEDs at the inverter have gone dark.

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 master 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 master.
- 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 master'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 master.

  #### Calculate value for lifetime

  Life time = guard time in milliseconds (p8604.0) * life time factor (p8604.1)

Heartbeat

Principle of operation

The slave periodically sends heartbeat messages. Other slaves and the master can monitor this signal. In the master, 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 master goes to the "Bus OFF" status. In the converter set the response via the p8641 parameter. Factory setting: p8641 = 3 (AUS3). When you have resolved the bus fault, you can confirm the error as follows:

- via OFF/ON: this enables you to remove the Bus OFF state and start communication again.
- via the DI 2 or direct via p3981: the converter remains in the Bus OFF state. To start the communication again, set p8608 = 1.

⚠ WARNING

Stopping the motor after a bus error

If the converter does not enter into a fault for a bus fault (p8641 = 0), the motor cannot be stopped via the controller.

For this case, wire an additional OFF command via terminals.

To stop the motor from the controller, you must resolve the bus fault and restart the communication via p8608 = 1.

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)
  - Inverter-specific error list (predefined error field)
  - CANopen error register (error register)

Display of the operating state using LEDs

In addition to the signal states "ON" and "OFF", the following flashing frequencies are available:

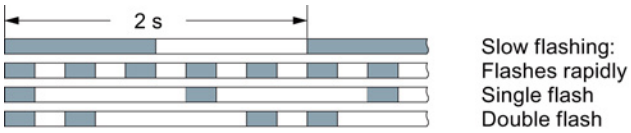


Table 6- 7 Diagnostics of the communication state using the LED

LED		Explanation
BF	RDY	
GREEN - on	Not relevant	Data is being exchanged between the inverter and control ("Operational")
GREEN - fast		Bus state "Pre-Operational"
GREEN - single flash		Bus state "Stopped"
RED - on		No bus available
RED - single flash		Alarm - limit reached
RED - double flash		Error event in control (Error Control Event)
RED - slow	RED - slow	Inverter waits until the power supply is switched off and switched on again after a firmware update
RED - fast	RED - fast	Incorrect memory card or unsuccessful firmware update
YELLOW - variable frequency	RED - on	Firmware update in progress

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 inverters, 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 master.

Inverter-specific error list (predefined error field)

You can read out the inverter-specific error list using the following objects:

- OV index 1003 hex
- Inverter parameter p8611

It includes the alarms and faults present in the inverter 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 inverter-specific error list.

You acknowledge all of the active inverter 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
- Inverter 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 inverter outputs fault F(A)08700(2). (Details are provided in the List Manual, see Additional manuals for your inverter (Page 182)).

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 inverter response in p8641:

- p8641 = 0 No reaction (factory setting)
- p8641 = 1 OFF1
- p8641 = 2 OFF2
- p8641 = 3 OFF3

Appendix

A

A.1 Communication with STEP7

This section describes the communication with a SIMATIC control system using examples.

To configure the control you require the SIMATIC STEP 7 software tool with HW Config.

It is assumed that you are knowledgeable about working with SIMATIC control systems and that you have a sound understanding of the STEP 7 engineering tool.

A.1.1 Configuring PROFIBUS communication

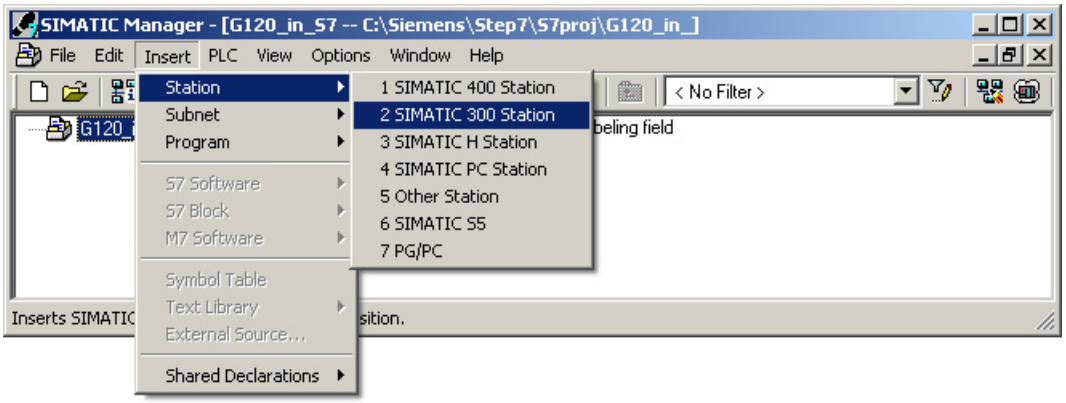
A.1.1.1 Creating a STEP 7 project and network

Procedure

1
2

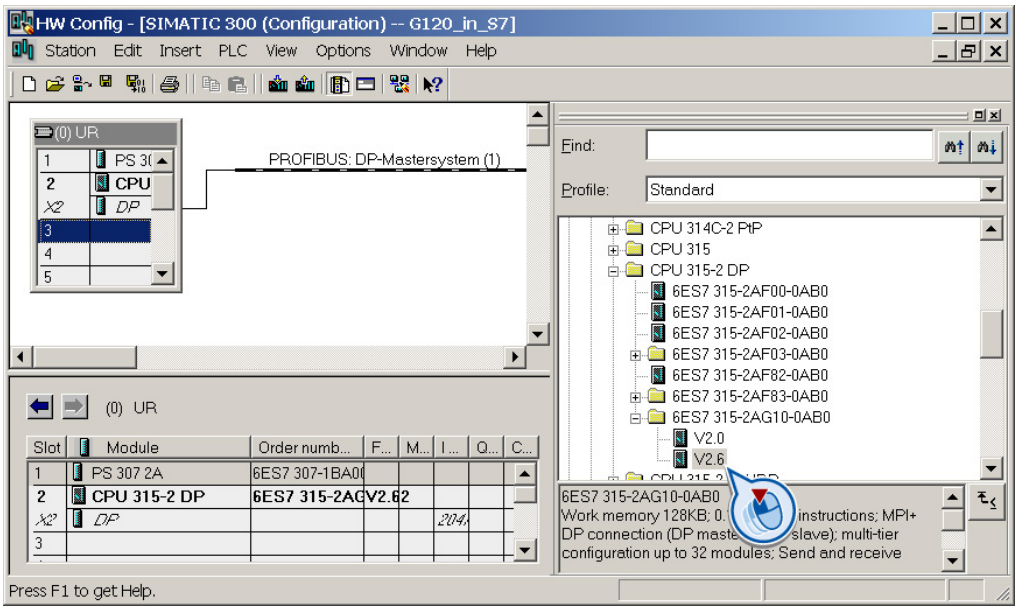
In order to create a STEP 7 project, proceed as follows:

1. Create a new STEP 7 project, e.g. "G120_in_S7".
2. Insert a SIMATIC control S7 300 CPU.



3. Select the SIMATIC 300 station in your project and open HW Config.
4. Insert an S7 300 mounting rail from the hardware catalog into your project with drag & drop.
5. Locate a power supply at slot 1 of the mounting rail and a CPU 315-2 DP control at slot 2.

   When inserting the control, HW Config opens the network setting.
6. Create a PROFIBUS DP network.



You have created a STEP 7 project with a SIMATIC control and a PROFIBUS network.

A.1.1.2 Inserting the inverter into the project

There are two ways to insert an inverter into the project:

- Using the inverter GSD
- Using the STEP 7 object manager
  This more user-friendly method is only available when STARTER is available.

Using an example of a SINAMICS G120 with Control Unit CU240B-2 or CU240E-2, the procedure shows how you insert the inverter into the project using the GSD.

Precondition

You have installed the GSD of the inverter in your PC using HW Config (menu "Options - Install GSD files").

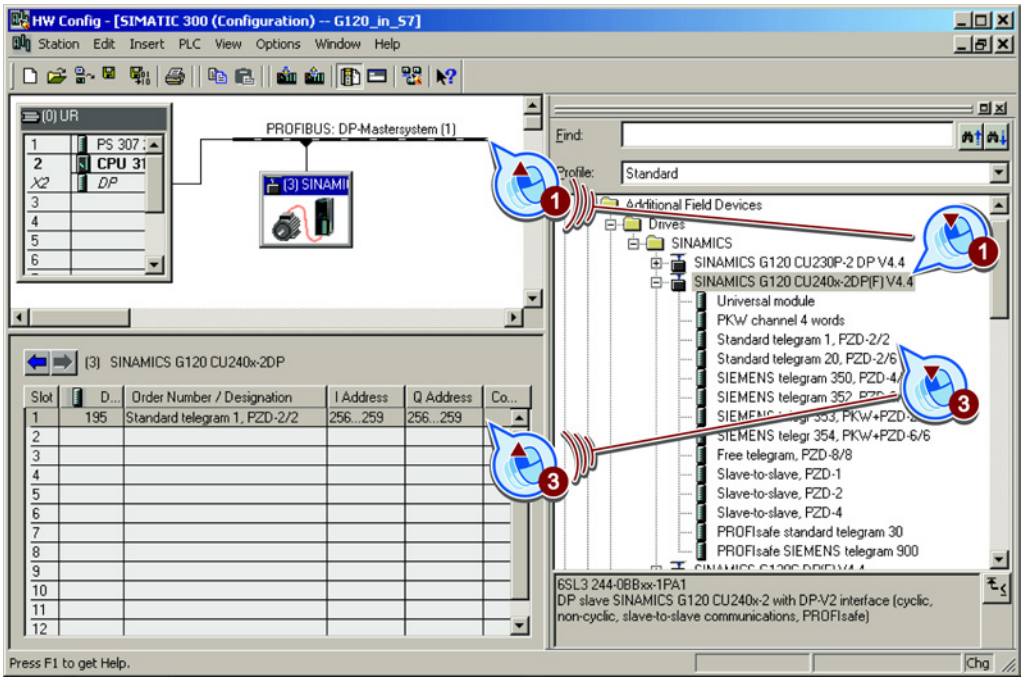
Procedure

1
2

In order to insert an inverter into your project, proceed as follows:

1. Insert the inverter by dragging and dropping it into the PROFIBUS network.
   You can find the inverter under "PROFIBUS DP - Other field devices" in the HW Config hardware catalog.
2. Enter the PROFIBUS address set at the inverter in HW Config.
3. Select the appropriate telegram and insert the telegram into slot 1 of the inverter by dragging and dropping.
   More information on the telegram types can be found in Chapter Cyclic communication (Page 15).
4. If you wish to assign several inverter slots with telegrams, then you must comply with the permitted sequence of slot assignments.

5. Save and compile the project.
6. Download the project data to the S7-CPU.



You have inserted your project into the inverter and loaded your configuration to the CPU.

Permitted sequence for the slot assignment

1. PROFIsafe telegram, if one is being used.
   Information on connecting the inverter via PROFIsafe can be found in the "Safety Integrated Function Manual".
2. PKW channel, if one is used.
3. Standard, SIEMENS or free telegram, if one is used.
4. Direct data exchange

If you do not use one or several of the telegrams 1, 2 or 3, configure your telegrams starting with the 1st slot.

Cyclic communication to the inverter when using the universal module

A universal module with the following properties is not permitted:

- PZD length 4/4 words
- Consistency over the complete length

With these properties, the universal module has the same DP identifier (4AX) as the "PKW channel 4 words". The higher-level control does not establish cyclic communication with the inverter.

Workaround when using the universal module above:

- In the properties of the DP slave, change the PZD length to 8/8 bytes
- Change the consistency to "Unit".

A.1.2 Configuring PROFINET communication

A.1.2.1 Configuring the controller and converter in HW Config

Using an example of a SINAMICS G120 with Control Unit CU240B-2 or CU240E-2, the procedure shows how you insert the inverter into the project.

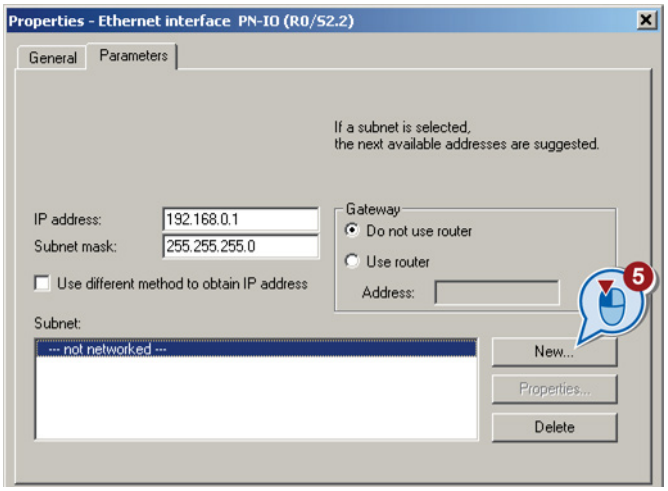
Procedure

Proceed as follows to configure communications between the inverter and the control system via PROFINET:

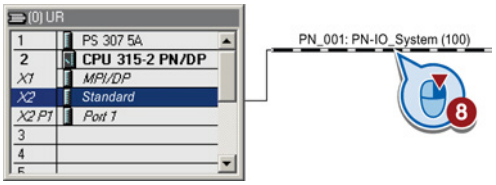
1. Open HW Config in STEP 7 via "Insert/[Station]", and create the components in accordance with your hardware structure. The following example is limited to the components that are absolutely required.
2. Build your station with a rack and power supply unit.
3. Insert the CPU.

   HW Config opens a screen form with suggestions for the next free IP address and a subnet screen form.
4. If you have configured a local area network, and are not working within a larger Ethernet network, use the proposed entries.

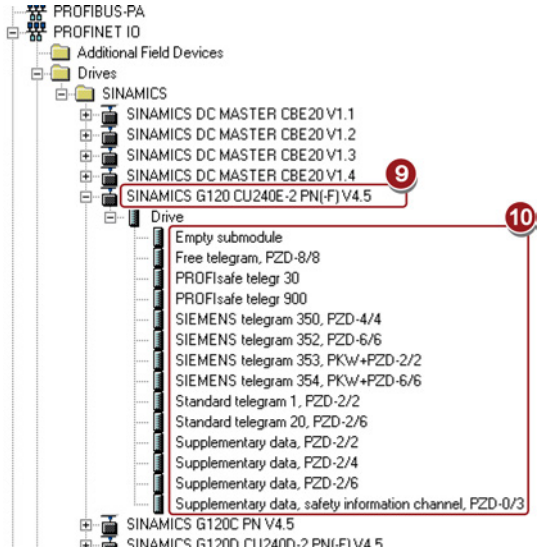
   Otherwise, ask your administrator about the IP addresses for the PROFINET participants and subnet mask. CPU and supervisor must have the same subnet screen form.



5. Use the "New" button to either create a new PROFINET subnet or select an existing one.
6. Assign a name for your PROFINET network.
7. Exit this screen form and the next one with OK.
8. Select your subnet.

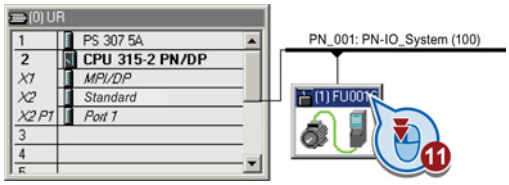


9. Using the hardware catalog, first insert the inverter using drag & drop.
10. Insert the communication telegram.



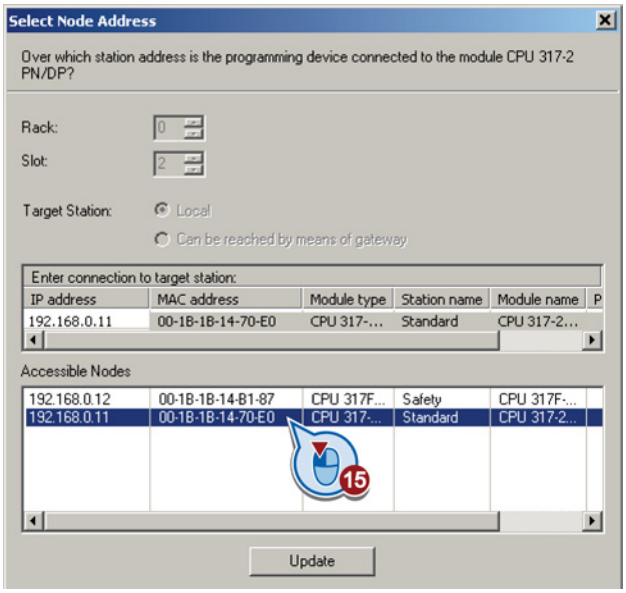
11. Open the properties window of the inverter and enter a unique and descriptive device name for the inverter.

    Using the device name, the PROFINET controller assigns the IP address when starting up.



12. You will also find the proposed IP address in this screen form. If required, you can change the IP address via "Properties".
13. Save your hardware configuration with "Save and compile" ( ).
14. Load the configuration into the control unit via the button.
15. Enter the IP address of the controller.

    If you do not have the IP address readily available, you can display the participants that can be reached by clicking the "Display" button. Select the control from the list of accessible participants, and exit the screen form with OK.



16. If you have installed Drive ES Basic, open the STARTER by double-clicking the inverter symbol in the Hardware Manager and configure the inverter in the STARTER.

    In this case, STARTER automatically accepts the device name and IP address. The approach described in the following section is therefore superfluous.

17. If you are working with the GSDML, close HW Config now and create a reference for STARTER as described in the following section.

You have configured the communication between the inverter and the control unit using PROFINET .

A.1.2.2 Activate diagnostic messages via STEP 7

Procedure

1
2

Proceed as follows to activate the diagnostic messages of the inverter:

1. In HW Config, select the inverter.

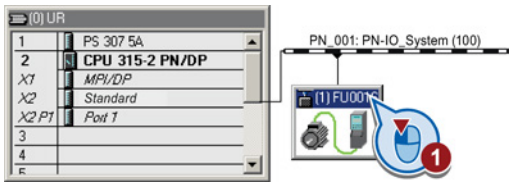
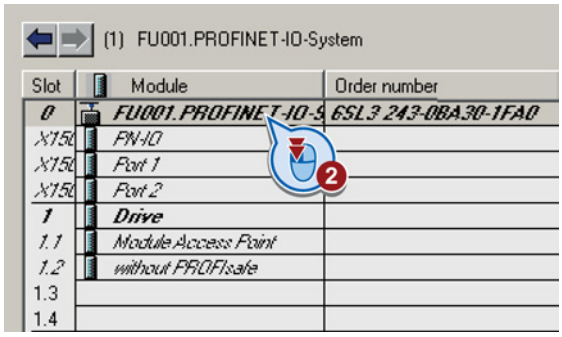
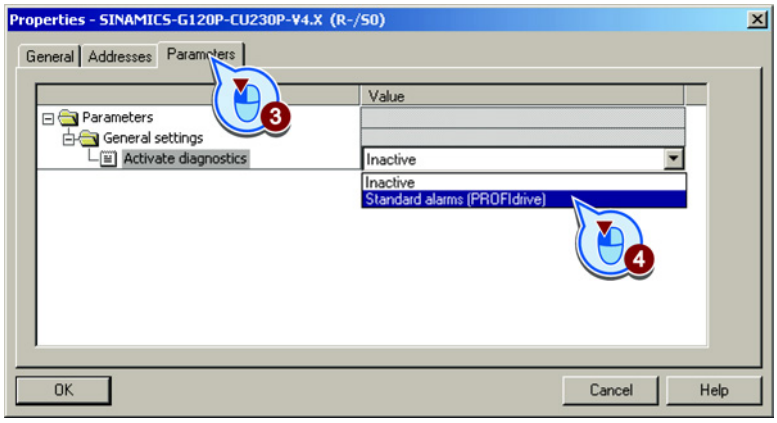


Figure A-1 Highlight inverter in HW Config

2. By double clicking on slot 0 in the station window, open the property window for the inverter's network settings.



3. Select the Parameters tab.
4. Activate the standard alarms.



You have activated the diagnosis messages.

With the next ramp-up of the controller, the diagnostic messages of the inverter are then transferred to the controller.

A.1.2.3 Go online with STARTER via PROFINET

Adapting the PROFINET interface

If you commission the inverter with STARTER via PROFINET, then you must correctly address your PC and allocate STARTER the interface via which it goes online with the inverter.

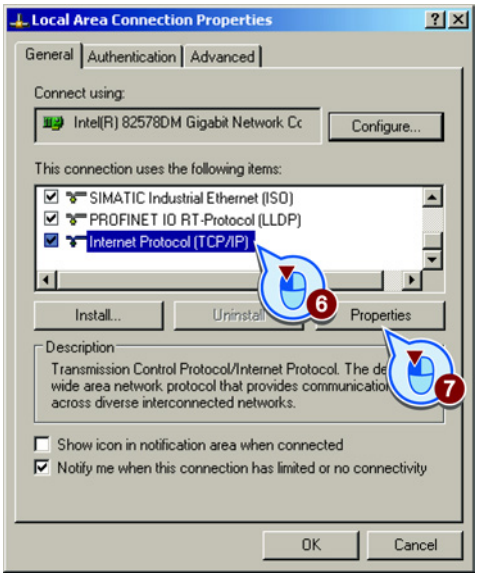
Procedure

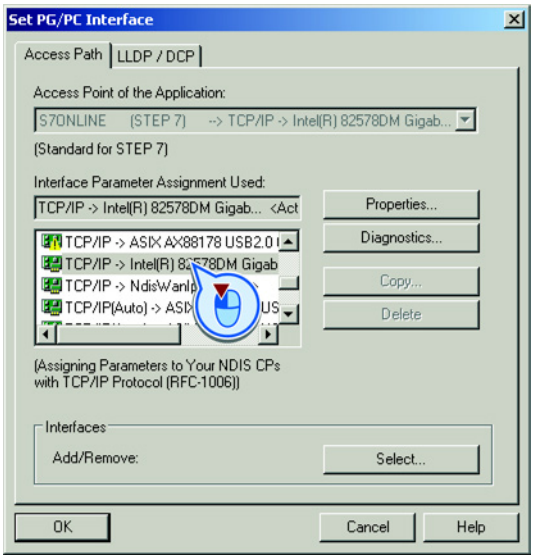
To address the inverter, proceed as follows:

1. Establish the bus connection.

   See Section Integrating inverters into PROFINET (Page 46))
2. From the control panel, assign the IP address and the subnet mask address to your computer:
3. Go to "Start/Settings/Control Panel".
4. Select "Network Connections".
5. Right-click to open the properties window of the LAN connection.
6. In this window, select "Internet Protocol (TCP/IP)".
7. Select "Properties".
8. Set 192.168.0.100 as the IP address of the supervisor and 255.255.255.0 as subnet mask.

   In the company network it is possible that the IP address and the subnet mask have different values. You can obtain these values from your network administrator.
9. Open the SIMATIC Manager.
10. Assign the TCP/IP interface to "Intel(R) PRO/100 VE Network Connection" via "Tools/PG/PC interface".





You have allocated your computer the IP address and the address of the subnet mask, and defined the PC interface via which STARTER goes online with the inverter.

Create a reference for STARTERS

If you have configured the inverter via GSDML, in STEP 7, you must create a reference of the inverter for STARTER, so that you can call up STARTER from STEP 7.

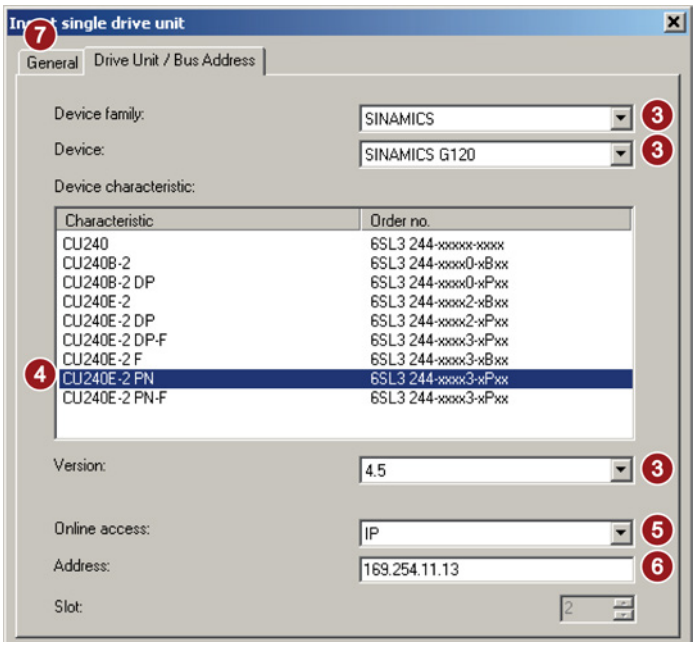
This procedure is described using the example of a SINAMICS G120 with Control Unit CU240B-2 or CU240E-2.

Procedure

1
2

Proceed as follows to create a reference of the inverter for STARTER:

1. Highlight the project in the SIMATIC manager
2. Open the "Insert single drive unit" screen form by right clicking on "Insert New Object/SINAMICS".
3. Under the "Drive device/address" tab, set the device family, device and the firmware version.
4. In the device version, select your inverter.
5. Set the online access.
6. Set the address.
7. Enter the PROFINET device name in the "General" tab.



8. Exit the screen form with OK.
9. The inverter is visible in your project.

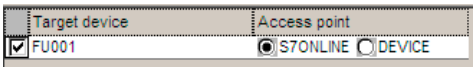
In your project, you have created an inverter reference for STARTER. You can now call STARTER from your STEP 7 project.

Call the STARTER and go online

Procedure

To call STARTER from STEP 7 and establish an online connection to the inverter, proceed as follows:

1. Highlight the inverter in the SIMATIC manager with the right mouse button.
2. Open the STARTER via "Open object".
3. Configure the inverter in STARTER and click on the Online button ( ).
4. In the following window, select the inverter and then the S7ONLINE as access point.



5. Exit the screen form with OK.

You have called STARTER from STEP 7, and have established the online connection to the inverter.

A.1.3 STEP 7 program examples

A.1.3.1 Data exchange via the fieldbus

Data exchange via the fieldbus

Analog signals

The inverter always scales signals that are transferred via the fieldbus to a value of 4000 hex.

Table A- 1 Signal category and the associated scaling parameters

Signal category	4000 hex ≙ …	available in			
		CU230P-2 G120C	CU240D-2 CU250D-2	CU240B-2 CU240E-2	CU250S-2
Speeds, frequencies	p2000	✓	✓	✓	✓
Voltage	p2001	✓	✓	✓	✓
Current	p2002	✓	✓	✓	✓
Torque	p2003	✓	✓	✓	✓
Power	r2004	✓	✓	✓	✓
Angle	p2005	---	✓	✓	✓
Temperature	p2006	✓	✓	✓	✓
Acceleration	p2007	---	✓	✓	✓

Control and status words

Control and status words consist of a high byte and a low byte. A SIMATIC control interprets words differently than the inverter: The higher and lower-order bytes are interchanged when they are transferred. See also the following program example.

A.1.3.2 STEP 7 program example for cyclic communication

Network 1: Control word 1 and setpoint

```
Control word 1: 047E hex
Setpoint: 2500 hex

L    W#16#47E
T    MW    1
L    W#16#2500
T    MW    3
```

Network 2: Acknowledge fault

```
U    E     0.6
=    M     2.7
```

Network 3: Switch the motor on and off

```
U    E     0.0
=    M     2.0
```

Network 4: Write process data

```
L    MW    1
T    PAW   256
L    MW    3
T    PAW   258
```

Network 4: Read process data

```
Status word 1: MW 5
Actual value: MW 7

L    PEW   256
T    MW    5
L    PEW   258
T    MW    7
```

The controller and inverter communicate via standard telegram 1. The control specifies control word 1 (STW1) and the speed setpoint, while the inverter responds with status word 1 (ZSW1) and its actual speed.

In this example, inputs E0.0 and E0.6 are linked to the ON/OFF1 bit or to the "acknowledge fault" bit of STW 1.

Control word 1 contains the numerical value 047E hex. The bits of control word 1 are listed in the following table.

The hexadecimal numeric value 2500 specifies the setpoint frequency of the inverter. The maximum frequency is the hexadecimal value 4000.

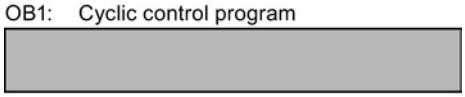
The controller cyclically writes the process data to logical address 256 of the inverter. The inverter also writes its process data to logical address 256. You define the address area in HW Config.

Table A- 2 Assignment of the control bits in the inverter to the SIMATIC flags and inputs

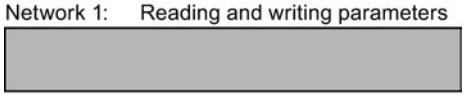
HEX	BIN	Bit in STW1	Significance	Bit in MW1	Bit in MB1	Bit in MB2	Inputs
E	0	0	ON/OFF1	8		0	E0.0
	1	1	OFF2	9		1	
	1	2	OFF3	10		2	
	1	3	Operation enable	11		3	
7	1	4	Ramp-function generator enable	12		4	
	1	5	Start ramp-function generator	13		5	
	1	6	Setpoint enable	14		6	
	0	7	Acknowledge fault	15		7	E0.6
4	0	8	Jog 1	0	0		
	0	9	Jog 2	1	1		
	1	10	PLC control	2	2		
	0	11	Setpoint inversion	3	3		
0	0	12	Irrelevant	4	4		
	0	13	Motorized potentiometer ↑	5	5		
	0	14	Motorized potentiometer ↓	6	6		
	0	15	Data set changeover	7	7		

A.1.3.3 STEP 7 program example for acyclic communication

OB1: Cyclic control program



Network 1: Reading and writing parameters



```
// read parameters
      O(
      U     M     9.2
      UN    M     9.1
      )
      O(
      U     M     9.0
      UN    M     9.1
      )
      R     M     9.3

      SPB   RD

// write parameters
      O(
      U     M     9.3
      UN    M     9.0
      )
      O(
      U     M     9.1
      UN    M     9.0
      )
      R     M     9.2

      SPB   WR
      BEA

RD:   NOP   0
      CALL  FC    1
      BEA
WR:   NOP   0     9.1
      CALL  FC    3
```

M9.0 Starts reading parameters

M9.1 Starts writing parameters

M9.2 Displays the read process

M9.3 Displays the write process

The number of simultaneous requests for acyclic communication is limited. More detailed information can be found under http://support.automation.siemens.com/WW/view/de/15364459 (http://support.automation.siemens.com/WW/view/en/15364459).

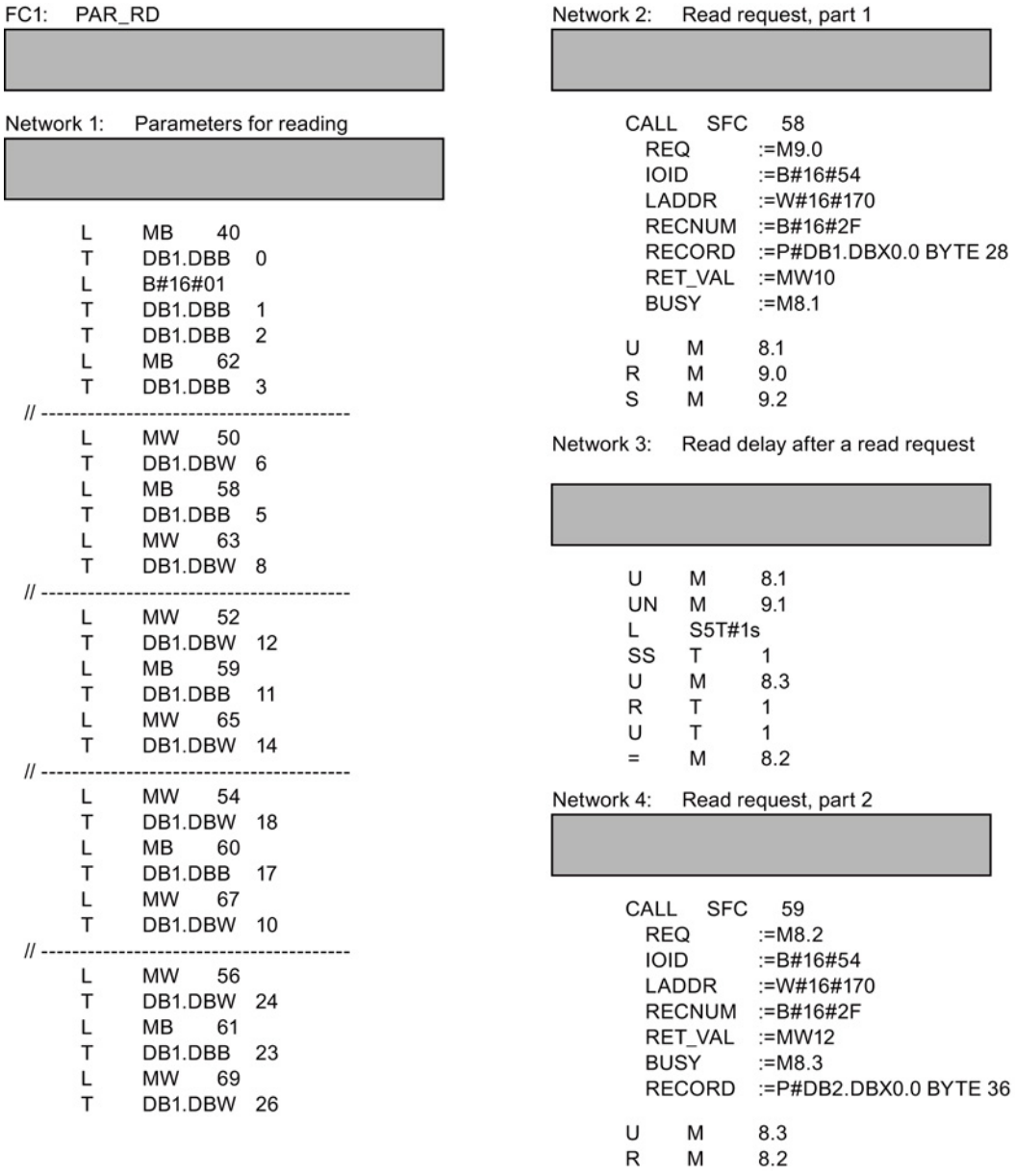


Figure A-2 Reading parameters

Note

With PROFINET standard function blocks (SFB) instead of system functions (SFC)

With acyclic communication via PROFINET, you must replace the system functions with standard function blocks as follows:

- SFC 58 → SFB 53
- SFC 59 → SFB 52

Explanation of FC 1

Table A- 3 Request to read parameters

Data block DB 1	Byte n	Bytes n + 1	n
Header	**Reference** *MB 40*	**01 hex**: Read request	0
	01 hex	**Number of parameters (m)** *MB 62*	2
Address, parameter 1	**Attribute** *10 hex: Parameter value*	**Number of indexes** *MB 58*	4
	Parameter number *MW 50*		6
	Number of the 1st index *MW 63*		8
Address, parameter 2	**Attribute** *10 hex: Parameter value*	**Number of indexes** *MB 59*	10
	Parameter number *MW 52*		12
	Number of the 1st index *MW 65*		14
Address, parameter 3	**Attribute** *10 hex: Parameter value*	**Number of indexes** *MB 60*	16
	Parameter number *MW 54*		18
	Number of the 1st index *MW 67*		20
Address, parameter 4	**Attribute** *10 hex: Parameter value*	**Number of indexes** *MB 61*	22
	Parameter number *MW 56*		24
	Number of the 1st index *MW 69*		26

SFC 58 copies the specifications for the parameters to be read from DB 1 and sends them to the inverter as a read request. No other read requests are permitted while this one is being processed.

After the read request and a waiting time of one second, the controller takes the parameter values from the inverter via SFC 59 and saves them in DB 2.

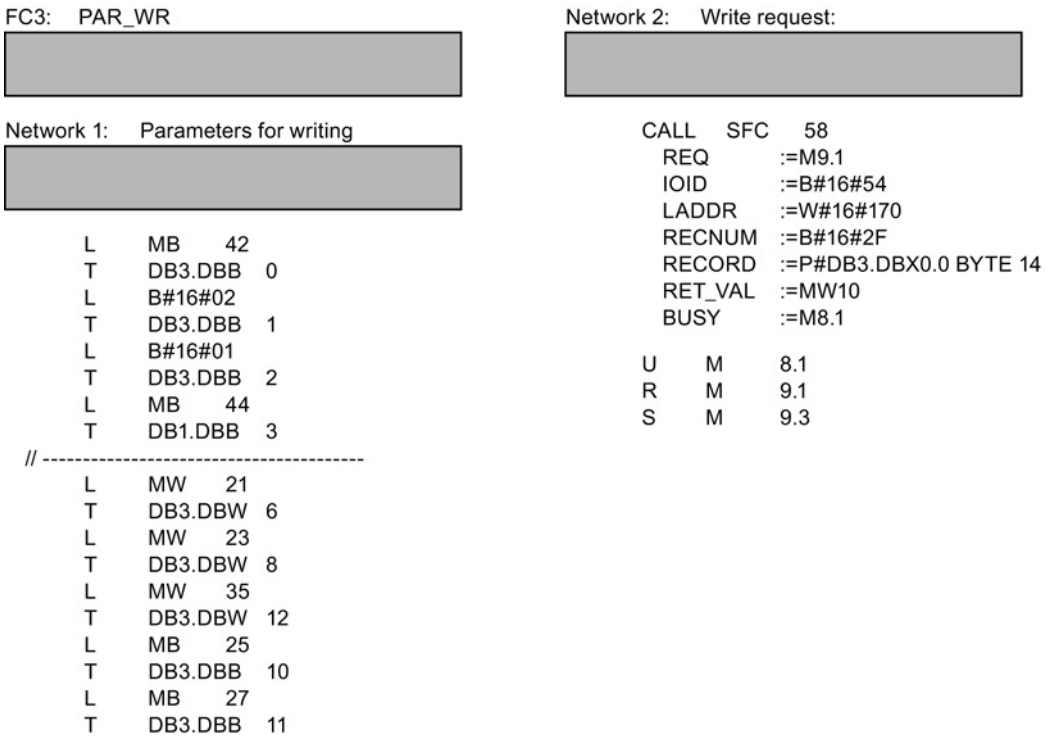


Figure A-3 Writing parameters

Explanation of FC 3

Table A- 4 Request to change parameters

Data block DB 3	Byte n	Bytes n + 1	n
Header	**Reference** *MB 42*	**02 hex**: Change request	0
	01 hex	**Number of parameters** *MB 44*	2
Address, parameter 1	**10 hex**: Parameter value	**Number of indexes** *00 hex*	4
	Parameter number *MW 21*		6
	Number of the 1st index *MW 23*		8
Values, parameter 1	**Format** *MB 25*	**Number of index values** *MB 27*	10
	Value of the 1st index*MW35*		12

SFC 58 copies the specifications for the parameters to be written from DB 3 and sends them to the inverter. The inverter blocks other write jobs while this write job is running.

A.1.4 Configuring slave-to-slave communication in STEP 7

Two drives communicate via standard telegram 1 with the higher-level controller. In addition, drive 2 receives its speed setpoint directly from drive 1 (actual speed).

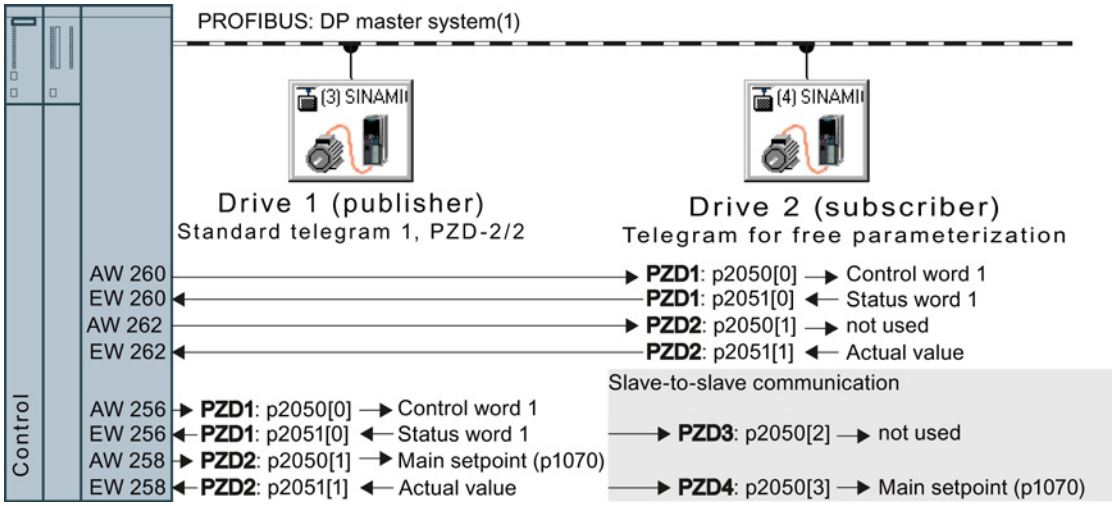


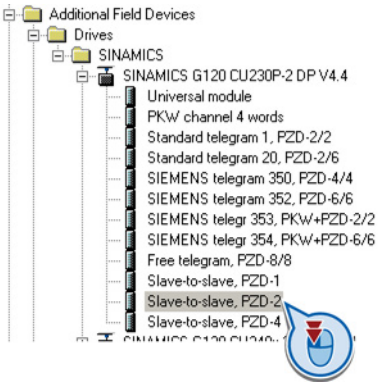
Figure A-4 Communication with the higher-level controller and between the drives with direct data exchange

Setting direct data exchange in the control

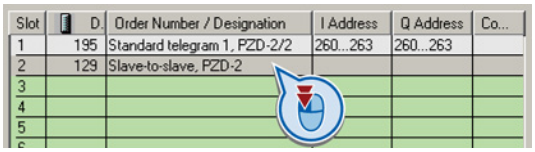
Procedure

Proceed as follows to set direct data exchange in the control:

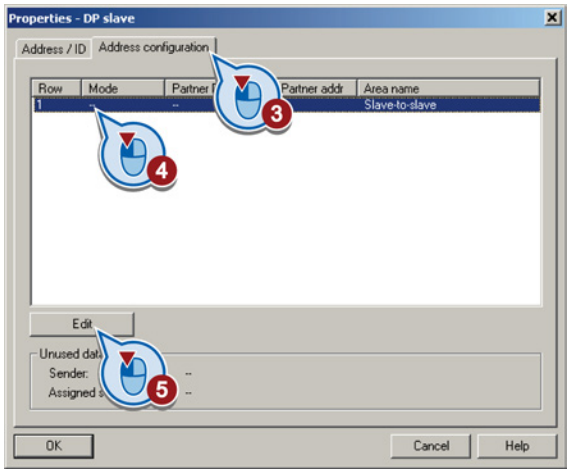
1. In HW Config in drive 2 (subscriber), insert a direct data exchange object, e.g. "Slave-to-slave, PZD2".



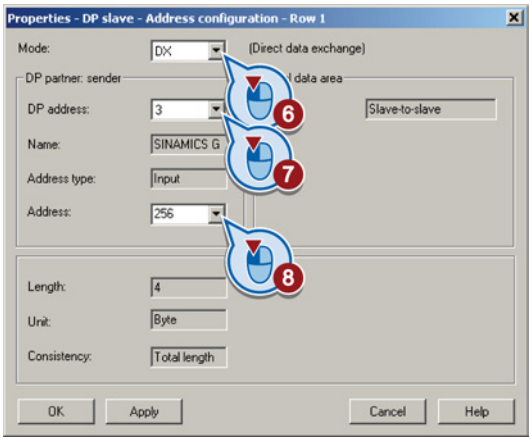
2. With a double-click, open the dialog box to make additional settings for the direct data exchange.



3. Activate the tab "Address configuration".
4. Select line 1.
5. Open the dialog box in which you define the Publisher and the address area to be transferred.



6. Select DX for direct data exchange
7. Select the address of drive 1 (publisher).
8. In the address field, select the start address specifying the data area to be received from drive 1. In the example, these are the status word 1 (PZD1) and the actual speed value with the start address 256.



9. Close both screen forms with OK.

You have now defined the value range for direct data exchange.

In the direct data exchange, drive 2 receives the sent data and writes this into the next available words, in this case, PZD3 and PZD4.

Settings in drive 2 (subscriber)

Drive 2 is preset in such a way that it receives its setpoint from the higher-level controller. In order that drive 2 accepts the actual value sent from drive 1 as setpoint, you must set the following:

- In drive 2, set the PROFIdrive telegram selection to "Free telegram configuration" (p0922 = 999).
- In drive 2, set the source of the main setpoint to p1070 = 2050.3.

The inverter indicates the inverter addresses that are configured for direct data exchange in parameter r2077.

A.2 Additional manuals for your inverter

Table A- 5 Manuals for your inverter

Information depth	Manual	Contents	Available languages	Download or order number
++	**Getting Started Guide**	Installing the inverter and commissioning.	English, German, Italian, French, Spanish, Chinese	Download manuals ((http://support.automation.siemens.com/WW/view/en/22339653/133300) **SINAMICS Manual Collection** Documentation on DVD, order number 6SL3097-4CA00-0YG0
+++	**Operating Instructions** for the SINAMICS G120 inverter with the CU250S-2 Control Unit	Installing the inverter and commissioning. Description of the inverter functions.		
+++	**Function Manual Basic Positioner**	Commissioning the basic positioner.	English, German, Chinese	
+++	**Function Manual for Safety Integrated** for the SINAMICS G110M, G120, G120C, G120D inverters and SIMATIC ET 200pro FC-2 converters	Configuring PROFIsafe. Installing, commissioning and operating fail-safe functions of the inverter.		
+++	**Fieldbus Function Manual** for the SINAMICS G120, G120C and G120D inverters	(this manual)		
+++	**List Manual**	Complete list of all parameters, alarms and faults. Graphic function diagrams.		
+	**Getting Started Guide** for the following SINAMICS G120 Power Modules: • PM240, PM250 and PM260 • PM240-2	Installing the Power Module	English	
+	**Installation Instructions** for reactors, filters and braking resistors	Installing components		
+++	**Hardware Installation Manual** for the following SINAMICS G120 Power Modules: • PM240 • PM240-2 • PM250 • PM260	Installing power modules, reactors and filters. Maintaining power modules.	English, German	
+++	**Operating Instructions** for the following Operator Panels: • BOP-2 • IOP	Operating operator panels, installing door assembly for IOP		

Index

O

P

R

S

T

U

Z

Further information

SINAMICS inverters:
www.siemens.com/sinamics

Siemens AG
Industry Sector
Drive Technologies
Motion Control Systems
Postfach 3180
91050 ERLANGEN
GERMANY

Scan the QR code for additional information on SINAMICS

www.siemens.com drives