31100307

I 488GB07_16

LOVATO ELECTRIC S.P.A.
24020 GORLE (BERGAMO) ITALIA
VIA DON E. MAZZA, 12
TEL. 035 4282111
TELEFAX (Nazionale): 035 4282200
TELEFAX (International): +39 035 4282400
Web www.LovatoElectric.com
E-mail info@LovatoElectric.com

GB

VLB3

VARIABLE SPEED DRIVES

COMMUNICATION PROTOCOLS (CANOPEN, MODBUS, PROFIBUS)

1 General network settings

Checking the network option

At switch-on, the inverter checks whether the parameter settings saved in the memory module match the network option available in the control unit. If this is not the case, the inverter assumes the error status. The error status can only be exited by accepting the currently available network option as new hardware. For this purpose, 0x2022:027 *(PAR 700/027)* must be set to "1". When this device command is executed, all parameters for the new network option are initialised.

Diagnostic parameters:

- The network option currently configured in the inverter is displayed in 0x231F:001 *(PAR 500/001)*.
- The network option currently available in the inverter is displayed in 0x231F:002 *(PAR 500/002)*.

Enabling the network as control source

In order to be able to control the inverter via network, a trigger must first be assigned in 0x2631:037 *(PAR 400/037)* of the "Network enable" function. This trigger can for instance be the constant value "TRUE" or a digital input. If the trigger assigned is TRUE, the inverter changes to the network control mode. The inverter now responds to the start and stop commands received via network.

In network control mode the following functions are still active:

- 0x2631:001 *(PAR 400/001)*: Controller enable
- 0x2631:002 *(PAR 400/002)*: Start enable
- 0x2631:003 *(PAR 400/003)*: Quick stop
- 0x2631:004 *(PAR 400/004)*: Reset error
- 0x2631:005 *(PAR 400/005)*: DC braking
- 0x2631:037 *(PAR 400/037)*: Network enable

All other functions that can be configured via 0x2631 (PAR 400) are deactivated in network control mode.

Network control word and status word

For establishing a simple network connection, the inverter provides predefined control and status words for device profile CiA402, AC drive profile as well as in LOVATO Electric format. By meansof data mapping to a network register, each of these words can be transferred as process data via network.

For the assignment of the predefined control and status words see the following chapters:

- Device profile CiA402.
- AC Drive Profile.
- LOVATO Electric profile.

If an individual control word format is to be implemented, the NETwordIN1 data word is provided for this purpose.

- The NETwordIN1 data word features the mapping address 0x40080100 (index 0x4008:001).
- The functions which are to be triggered via bits 1 ... 16 of the NETwordIN1 data word are defined in 0x400E:001 *(PAR 505/001)*.

If an individual status word format is to be implemented, the NETwordOUT1 data word is provided for this purpose.

- The NETwordOUT1 data word features the mapping address 0x400A0100 (index 0x400A: 001).
- The trigger for bits 1 ... 16 of the NETwordOUT1 data word is defined in 0x2634:010 *(PAR 420/010)* ... 0x2634:025 *(PAR 420/025)*.

General information about the process of data mapping can be found in the chapter of the same name for the corresponding network.

Network setpoint

- In order to generally set the network as standard setpoint source, go to 0x2860:001 *(PAR 201/001)* and select "Network [5]".
- If a bit of the mappable NETWordIN1 data word is to be used for activation, use 0x400E:001 *(PAR 505/001)* to assign the function "Setpoint = Network [17]" to the corresponding bit of NETWordIN1.
- There is no specific function with a selectable trigger for a setpoint change-over to the network.

Parameter	Name / value range / [default setting]		Info
0x231F:001 *(PAR 500/001)*	Module ID: Active module ID *(Module ID: Active module ID)* • Read only • Default setting depending on the size.		Display of the network options currently configured in the inverter. • When the "Load default settings" device command 0x2022:001 *(PAR 700/001)*or "Accept new inverter hardware" 0x2022:027 *(PAR 700/027)* is executed, the module ID is stored in the memory module. • With the help of this module ID, the keypad only shows the communication parameters relevant to the respective network.
	0	No network	
	67	CANopen	
	80	PROFIBUS	
	87	Modbus	
0x231F:002 *(PAR 500/002)*	Module ID: Module ID connected *(Module ID: Module ID conn.)* • Read only		Display of the network options currently connected in the inverter.
	0	No network	
	67	CANopen	
	80	PROFIBUS	
	87	Modbus	
0x400E:001 *(PAR 505/001)*	NETWordIN1 function assignment: Bit 0 *(NETWordIN1 config.: NETWordIN1.00)* • Setting can only be changed if controller inhibit is active.		Definition of the function that is to be triggered via bit 0 of the mappable NETWordIN1 data word.
	0	**Not active**	Trigger bit without any function.
	1	Controller inhibit	Trigger bit = 0-1 edge: The inverter is inhibited. Starting the drive is not possible. Trigger bit = 0: The inverter is enabled (unless there is another cause for controller inhibit). Notes: • In all device states, a 0-1 edge causes an immediate change to the inhibited state with one exception: If the inverter is in the error status and the error condition still exists, the inverter remains in the error status. • Changing to the inhibited state causes an immediate stop of the drive, regardless of the stop method set in 0x2838:003 *(PAR 203/003)*. The drive coasts down as a function of the mass inertia of the machine. • After deactivating the controller inhibit, a new start command is required to restart the drive. • The causes that are active for controller inhibit are shown in 0x282A:001 *(PAR 126/001)*.
	2	Stop	Trigger bit = 1: Drive is stopped. Trigger bit = 0: No action / Deactivate stop again. Notes: • The stop method can be selected in 0x2838:003 *(PAR 203/003)*.
	3	Quick stop	Trigger bit = 1: "Quick stop" function activated. Trigger bit = 0: no action / deactivate function again. Notes: • The "Quick stop" function brings the motor to a standstill within the deceleration time set in 0x291C *(PAR 225)*. • The "Quick stop" function has a higher priority than the "Start enable" function.

Parameter	Name / value range / [default setting]		Info
	4	Reset error	Trigger bit = 0-1 edge: active error is reset (acknowledged) if the error cause has been eliminated. Trigger bit = 0: no action. Notes: • After resetting the error, a new enable/start command is required to restart the drive.
	5	DC braking	Trigger bit = 1: "DC braking" function activated. Trigger bit = 0: no action / deactivate function again.
	8	Forward run (CW)	Trigger bit = 0-1 edge: drive is started in forward rotating direction (CW). Trigger bit = 1-0 edge: drive is stopped again. Notes: • In order to start the drive with this function, the "Stop [2]" function has to be assigned to a bit in order to provide a stop command. • The stop method can be selected in 0x2838:003 *(PAR 203/003)*. • In the case of a bipolar setpoint selection (e.g ±10 V), the function is executed irrespective of the rotating direction. The rotating direction is determined by the sign of the setpoint. • The function also serves to realise an automatic start after switch-on. ▶ Starting performance • The "Reversal [13]" function can be used in connection with this function.
	9	Backward run (CCW)	Trigger bit = 0-1 edge: drive is started in backward rotating direction (CCW). Trigger bit = 1-0 edge: drive is stopped again. Notes: • In order to start the drive with the function, the "Stop [2]" function has to be assigned to a bit in order to provide a stop command. • The stop method can be selected in 0x2838:003 *(PAR 203/003)*. • In the case of a bipolar setpoint selection (e.g ±10 V), the function is executed irrespective of the rotating direction. The rotating direction is determined by the sign of the setpoint. • The function also serves to realise an automatic start after switch-on. ▶ Starting performance • The "Reversal [13]" function can be used in connection with this function.
	13	Reversal	Trigger bit = 1: the setpoint specified is inverted (i. e. the sign is inverted). Trigger bit = 0: no action / deactivate function again.
	14	Setpoint = AI1	Trigger bit = 1: analog input 1 is used as setpoint source (if the trigger bit assigned has the highest setpoint priority). Trigger bit = 0: no action / deactivate function again.
	15	Setpoint = AI2	Trigger bit = 1: analog input 2 is used as setpoint source (if the trigger bit assigned has the highest setpoint priority). Trigger bit = 0: no action / deactivate function again.
	17	Setpoint = Network	Trigger bit = 1: the network is used as setpoint source (if the trigger bit assigned has the highest setpoint priority). Trigger bit = 0: no action / deactivate function again.
	18	Preset val. selection bit 0	Selection bit with the valency 2^0 for the bit-coded selection and activation of a parameterisable setpoint (preset value).
	19	Preset val. selection bit 1	Selection bit with the valency 2^1 for the bit-coded selection and activation of a parameterisable setpoint (preset value).
	20	Preset val. selection bit 2	Selection bit with the valency 2^2 for the bit-coded selection and activation of a parameterisable setpoint (preset value).
	21	Preset val. selection bit 3	Selection bit with the valency 2^3 for the bit-coded selection and activation of a parameterisable setpoint (preset value).
	39	Activation of ramp 2	Trigger bit = 1: activate acceleration time 2 and deceleration time 2 manually. Trigger bit = 0: no action / deactivate function again.

Parameter	Name / value range / [default setting]		Info
	40	Load parameter set	Trigger bit = 0-1 edge: parameter change-over to the value set selected via "Parameter set selection bit 0" and "Parameter set selection bit 1". Trigger bit = 0: no action. Notes: • The activation method for the "Parameter change-over" function can be selected in 0x4046 *(PAR 755)*.
	41	Parameter set selection bit 0	Selection bit with the valency 2^0 for "Parameter change-over" function.
	42	Parameter set selection bit 1	Selection bit with the valency 2^1 for "Parameter change-over" function.
	43	User-defined fault 1	Trigger bit = 1: trigger user-defined fault 1. Trigger bit = 0: no action. Notes: • When the fault has been triggered, the inverter changes to the "Fault" state. • After resetting the fault, a new enable/start command is required to restart the drive.
	44	User-defined fault 2	Trigger bit = 1: trigger user-defined fault 2. Trigger bit = 0: no action. Notes: • When the fault has been triggered, the inverter changes to the "Fault" state. • After resetting the fault, a new enable/start command is required to restart the drive.
	45	Process controller off	Trigger bit = 1: if process controller mode is active, ignore PID control and actuate the drive in speed-controlled manner. Trigger bit = 0: if process controller mode is active, actuate the drive with PID control. Notes: • PID control can be selected in 0x4020:001 *(PAR 600/001)*.
	46	Set process controller output to 0	Trigger bit = 1: if process controller mode is active, the I component and the output of the process controller are set to 0 and the internal control algorithm is stopped. Process controller mode remains active. Trigger bit = 0: no action / deactivate function again.
	47	Inhibit process controller I-component	Trigger bit = 1: if process controller mode is active, the I component of the process controller is set to 0 and the integration process is stopped. Trigger bit = 0: no action / deactivate function again. Notes: • The reset time can be set in 0x4049 *(PAR 602)*.
	48	Activate process controller influence ramp	Trigger bit = 1: the influence of the process controller is shown by means of a ramp. Trigger bit = 0 or not connected: the influence of the process controller is shown by means of a ramp. Notes: • The influence of the process controller is always active (not only in process controller mode). • Acceleration time for showing the influence of the process controller can be set in 0x404C:001 *(PAR 607/001)*. • Deceleration time for hiding the influence of the process controller can be set in 0x404C:002 *(PAR 607/002)*.
0x400E:002 *(PAR 505/002)*	NETWordIN1 function assignment: Bit 1 *(NETWordIN1 config.: NETWordIN1.01)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.		Definition of the function that is to be triggered via bit 1 of the mappable NETWordIN1 data word.
	00	**Not active**	Trigger bit without any function.

Parameter	Name / value range / [default setting]	Info
0x400E:003 *(PAR 505/003)*	NETWordIN1 function assignment: Bit 2 *(NETWordIN1 config.: NETWordIN1.02)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 2 of the mappable NETWordIN1 data word.
	03 **Quick stop**	Trigger bit = 1: "Quick stop" function activated. Trigger bit = 0: no action / deactivate function again. Notes: • The "Quick stop" function brings the motor to a standstill within the deceleration time set in 0x291C *(PAR 225)*. • The "Quick stop" function has a higher priority than the "Start enable" function.
0x400E:004 *(PAR 505/004)*	NETWordIN1 function assignment: Bit 3 *(NETWordIN1 config.: NETWordIN1.03)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 3 of the mappable NETWordIN1 data word.
	8 **Forward run (CW)**	Trigger bit = 0-1 edge: drive is started in forward rotating direction (CW). Trigger bit = 1-0 edge: drive is stopped again. Notes: • In order to start the drive with this function, the "Stop [2]" function has to be assigned to a bit in order to provide a stop command. • The stop method can be selected in 0x2838:003 *(PAR 203/003)*. • In the case of a bipolar setpoint selection (e.g ±10 V), the function is executed irrespective of the rotating direction. The rotating direction is determined by the sign of the setpoint. • The function also serves to realise an automatic start after switch-on. ▸ Starting performance • The "Reversal [13]" function can be used in connection with this function.
0x400E:005 *(PAR 505/005)*	NETWordIN1 function assignment: Bit 4 *(NETWordIN1 config.: NETWordIN1.04)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 4 of the mappable NETWordIN1 data word.
	13 **Reversal**	Trigger bit = 1: the setpoint specified is inverted (i. e. the sign is inverted). Trigger bit = 0: no action / deactivate function again.
0x400E:006 *(PAR 505/006)*	NETWordIN1 function assignment: Bit 5 *(NETWordIN1 config.: NETWordIN1.05)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 5 of the mappable NETWordIN1 data word.
	05 **DC braking**	Trigger bit = 1: "DC braking" function activated. Trigger bit = 0: no action / deactivate function again.
0x400E:007 *(PAR 505/007)*	NETWordIN1 function assignment: Bit 6 *(NETWordIN1 config.: NETWordIN1.06)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 6 of the mappable NETWordIN1 data word.
	00 **Not active**	Trigger bit without any function.

Parameter	Name / value range / [default setting]	Info
0x400E:008 *(PAR 505/008)*	NETWordIN1 function assignment: Bit 7 *(NETWordIN1 config.: NETWordIN1.07)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 7 of the mappable NETWordIN1 data word.
	04 **Reset error**	Trigger bit = 0-1 edge: active error is reset (acknowledged) if the error cause has been eliminated. Trigger bit = 0: no action. Notes: • After resetting the error, a new enable/start command is required to restart the drive.
0x400E:009 *(PAR 505/009)*	NETWordIN1 function assignment: Bit 8 *(NETWordIN1 config.: NETWordIN1.08)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 8 of the mappable NETWordIN1 data word.
	18 **Preset val. selection bit 0**	Selection bit with the valency 2^0 for the bit-coded selection and activation of a parameterisable setpoint (preset value).
0x400E:010 *(PAR 505/010)*	NETWordIN1 function assignment: Bit 9 *(NETWordIN1 config.: NETWordIN1.09)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 9 of the mappable NETWordIN1 data word.
	19 **Preset val. selection bit 1**	Selection bit with the valency 2^1 for the bit-coded selection and activation of a parameterisable setpoint (preset value).
0x400E:011 *(PAR 505/011)*	NETWordIN1 function assignment: Bit 10 *(NETWordIN1 config.: NETWordIN1.10)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 10 of the mappable NETWordIN1 data word.
	00 **Not active**	Trigger bit without any function.
0x400E:012 *(PAR 505/012)*	NETWordIN1 function assignment: Bit 11 *(NETWordIN1 config.: NETWordIN1.11)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 11 of the mappable NETWordIN1 data word.
	00 **Not active**	Trigger bit without any function.
0x400E:013 *(PAR 505/013)*	NETWordIN1 function assignment: Bit 12 *(NETWordIN1 config.: NETWordIN1.12)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 12 of the mappable NETWordIN1 data word.
	00 **Not active**	Trigger bit without any function.
0x400E:014 *(PAR 505/014)*	NETWordIN1 function assignment: Bit 13 *(NETWordIN1 config.: NETWordIN1.13)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 13 of the mappable NETWordIN1 data word.
	00 **Not active**	Trigger bit without any function.
0x400E:015 *(PAR 505/015)*	NETWordIN1 function assignment: Bit 14 *(NETWordIN1 config.: NETWordIN1.14)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.	Definition of the function that is to be triggered via bit 14 of the mappable NETWordIN1 data word.
	00 **Not active**	Trigger bit without any function.

Parameter	Name / value range / [default setting]		Info
0x400E:016 *(PAR 505/016)*	NETWordIN1 function assignment: Bit 15 *(NETWordIN1 config.: NETWordIN1.15)* • Setting can only be changed if controller inhibit is active. • For possible settings see description for 0x400E:001 *(PAR 505/001)*.		Definition of the function that is to be triggered via bit 15 of the mappable NETWordIN1 data word.
	00	**Not active**	Trigger bit without any function.
0x2022:027 *(PAR 700/027)*	Device commands: Accept new inverter hardware *(Device commands: Accept new HW)* 0 ... [**0**] ... 1 • Setting can only be changed if controller inhibit is active.		1 = initialise parameters for a new network option.
0x2631:037 *(PAR 400/037)*	Function assignment: Network enable *(Function list: Network enable)* • For possible settings see description for 0x2631:001 *(PAR 400/001)*.		Assignment of a trigger for the "Network enable" function. Trigger = TRUE: activate network control mode. Trigger = FALSE: no action / deactivate network control mode again.
	00	**Not connected**	No trigger assigned (trigger is constantly FALSE).
0x2860:001 *(PAR 201/001)*	Standard setpoint sources: Frequency setpoint source *(Standard setpoint: Frequency setp.)* • For possible settings see description for 0x2860:001 *(PAR 201/001)*.		Selection of the standard setpoint source for speed mode. • The standard setpoint source is always active in speed mode if no setpoint change-over to another setpoint source via corresponding triggers/functions is active. ▸ Setpoint change-over
	2	**Analog input 1**	The setpoint is specified analogously via X3/AI1. ▸ Analog input 1
0x2860:002 *(PAR 201/002)*	Standard setpoint sources: Process controller setpoint source *(Standard setpoint: PID setpoint)* • For possible settings see description for 0x2860:002 *(PAR 201/002)*.		Selection of the standard setpoint source for process controller mode. • The standard setpoint source is always active in process controller mode if no setpoint change-over to another setpoint source via corresponding triggers/functions is active.
	1	**Keypad**	The setpoint is specified locally by the keypad. • Default setting: 0x2601:002 *(PAR 202/002)* • Use the ↑ and ↓ navigation keys to change the keypad setpoint (also during running operation).
0x2860:003 *(PAR 201/003)*	Standard setpoint sources: Torque setpoint source *(Standard setpoint: Torque setp.)*		Selection of the standard setpoint source for torque mode. • The standard setpoint source is always active in torque mode if no setpoint change-over to another setpoint source via corresponding triggers/functions is active.
	1	Keypad	The setpoint is specified locally by the keypad. • Use the ↑ and ↓ navigation keys to change the keypad setpoint (also during running operation).
	2	**Analog input 1**	The setpoint is specified analogously via X3/AI1. ▸ Analog input 1
	3	Analog input 2	The setpoint is specified analogously via X3/AI2. ▸ Analog input 2
	50	Motor potentiometer	The setpoint is generated by the "Motor potentiometer" function. ▸ Motor potentiometer setpoint source

2 Predefined process data words

Process data are exchanged via cyclic data exchange between the network master and the inverter.

Details

For the cyclic data exchange, the inverter is provided with 24 network registers.

- 12 network registers are provided as input registers for data words from the network master to the inverter.
- 12 network registers are provided as output registers for data words from the inverter to the network master.
- Each network register is provided with a corresponding code that defines which parameters (or other data codes) are mapped to the network register.
- The input and output registers are divided into three blocks (A, B, C) in each case, featuring 4 successive data words, respectively:

Network register	
Input register	**Output register**
Network IN A0	Network OUT A0
Network IN A1	Network OUT A1
Network IN A2	Network OUT A2
Network IN A3	Network OUT A3
Network IN B0	Network OUT B0
Network IN B1	Network OUT B1
Network IN B2	Network OUT B2
Network IN B3	Network OUT B3
Network IN C0	Network OUT C0
Network IN C1	Network OUT C1
Network IN C2	Network OUT C2
Network IN C3	Network OUT C3

The terms "input" and "output" refer to the point of view of the inverter:

- Input data are transmitted by the network master and received by the inverter.
- Output data are transmitted by the inverter and received by the network master.

The exact assignment of the network registers and the number of data words that can be transmitted cyclically varies according to the network/communication protocol. You can find some detailed information in the documentation for the respective communication protocol.

Data mapping

For establishing a simple network connection, the inverter provides predefined control and status words for device profile CiA402, AC drive profile as well as in LOVATO Electric format. By means of data mapping to a network register, each of these words can be transferred as process data via network. Additionally, further mappable data words to individually control the inverter are provided. The mappable data words are described in detail in the following subchapters.

Data mapping cannot be applied to all parameters.

2.1 Device profile CiA402

For control via CiA402 device profile, the parameters listed in the following can be mapped to network registers.

Details

- The Controlword features the mapping address 0x60400000.
- The Statusword features the mapping address 0x60410000.
- General information about the process of data mapping can be found in the chapter of the same name for the corresponding network.

Parameter	Name / value range / [default setting]		Info
0x6040	Controlword 0 ... [**0**] ... 65535		Mappable CiA402 control word with bit assignment in compliance with CiA402 device profile.
	Bit 0	Switch on	
	Bit 1	Enable voltage	
	Bit 2	Quick stop	
	Bit 3	Enable operation	
	Bit 4	Operation mode specific	
	Bit 5		
	Bit 6		
	Bit 7	Fault reset	
	Bit 8	n/a	Bit is not supported.
	Bit 9	Operation mode specific	
	Bit 10	Reserved	
	Bit 11	Override coast	
	Bit 12	AutoInit	
	Bit 13	Reserved	
	Bit 14	Release holding brake	
	Bit 15	Reserved	
0x6041 *(PAR 780)*	Statusword *(Statusword)* • Read only		Mappable CiA402 status word with bit assignment in compliance with CiA402 device profile.
	Bit 0	Ready to switch on	
	Bit 1	Switched on	
	Bit 2	Operation enabled	
	Bit 3	Fault active	
	Bit 4	Voltage enabled	
	Bit 5	Quick stop	
	Bit 6	Switch on disabled	
	Bit 7	Warning active	
	Bit 8	Deactivate RPDOs	1 ≡ cyclic PDOs have been deactivated.
	Bit 9	Remote	1 ≡ inverter can receive commands via network.
	Bit 10	Target reached	1 ≡ the actual position is in the window.
	Bit 11	Internal limit active	1 ≡ internal limitation of a setpoint active.
	Bit 12	Operation mode active	
	Bit 13	Following error	
	Bit 14	Holding brake released	
	Bit 15	Safe torque off (STO) not active	

2.2 AC Drive Profile

For control via AC drive profile, the parameters listed in the following can be mapped to network registers.

Details

- The AC Drive control word features the mapping address 0x400B0100.
- The AC Drive status word features the mapping address 0x400C0100.
- General information about the process of data mapping can be found in the chapter of the same name for the corresponding network.

Parameter	Name / value range / [default setting]		Info
0x400B:001 *(PAR 592/001)*	Predefined process input data: AC Drive control word *(Legacy NetWordIN: AC control word)* 0x0000 ... [**0x0000**] ... 0xFFFF		Mappable control word with bit assignment in compliance with EtherNet/IP™ AC drive profile.
	Bit 0	Run forward	
	Bit 1	Run reverse	
	Bit 2	Reset error (0-1 edge)	
	Bit 3	Reserved	
	Bit 4		
	Bit 5	Control from Network	
	Bit 6	Reference from Network	
	Bit 7	Reserved	
	Bit 8		
	Bit 9		
	Bit 10		
	Bit 11		
	Bit 12	Controller inhibit	
	Bit 13	Quick stop	
	Bit 14	Process controller off	
	Bit 15	DC braking	
0x400C:001 *(PAR 593/001)*	Predefined process output data: AC Drive status word *(Old netw. off: AC drive status word)* • Read only		Mappable status word with bit assignment in compliance with EtherNet/IP™ AC drive profile.
	Bit 0	Fault/Trip	
	Bit 1	Reserved	
	Bit 2	Running Forward	
	Bit 3	Running Reverse	
	Bit 4	Ready	
	Bit 5	Control from Network	
	Bit 6	Reference from Network	
	Bit 7	At Reference	
	Bit 8	Reserved	
	Bit 9		
	Bit 10		
	Bit 11		
	Bit 12	Process controller active	
	Bit 13	Torque mode active	
	Bit 14	Current limit reached	
	Bit 15	DC braking active	

2.3 LOVATO Electric profile

For connection to inverters with a LOVATO control word (C135) and LOVATO status word (C150), the parameters listed in the following can be mapped to network registers.

Details

- The LOVATO control word (C135) features the mapping address 0x400B0200.
- The LOVATO status word (C150) features the mapping address 0x400C0200.
- General information about the process of data mapping can be found in the chapter of the same name for the corresponding network.

Parameter	**Name / value range / [default setting]**		**Info**
0x400B:002 *(PAR 592/002)*	Predefined process input data: LOVATO control word (C135) *(Legacy NetWordIN: C0135 control word)* 0x0000 ... [**0x0000**] ... 0xFFFF		Mappable control word with bit assignment in compliance with code C135.
	Bit 0	Setpoint Selection bit 0	
	Bit 1	Setpoint Selection bit 1	
	Bit 2	Reversal	
	Bit 3	Quick stop	
	Bit 4	Reserved	
	Bit 5		
	Bit 6		
	Bit 7		
	Bit 8		
	Bit 9	Controller inhibit	
	Bit 10	User-defined fault	
	Bit 11	Reset error (0-1 edge)	
	Bit 12	Reserved	
	Bit 13		
	Bit 14	DC braking	
	Bit 15	Reserved	
0x400C:002 *(PAR 593/002)*	Predefined process output data: LOVATO status word (C150) *(Old netw. off: C0150 status word)* • Read only		Mappable status word with bit assignment in compliance with code C150.
	Bit 0	Active parameter set (0 = set 1 or 3; 1 = set 2 or 4)	
	Bit 1	Power section inhibited	
	Bit 2	Current or Torque limit reached	
	Bit 3	Frequency setpoint reached	
	Bit 4	Ramp generator (input = output)	
	Bit 5	Frequency < frequency threshold	
	Bit 6	Actual frequency = 0	
	Bit 7	Controller inhibit	
	Bit 8	Coded status bit 0	
	Bit 9	Coded status bit 1	
	Bit 10	Coded status bit 2	
	Bit 11	Coded status bit 3	
	Bit 12	Overtemperature warning	
	Bit 13	DC-bus overvoltage	
	Bit 14	Reversal	
	Bit 15	Ready for Operation	

2.4 Further process data

The parameters listed in the following can also be mapped to network registers, in order to transmit control and status information as well as setpoints and actual values as process data.

Details

- The following parameters are always available irrespective of the network option.
- The use of these parameters for the transmission of process data is optional. It is also possible to only use part of the parameters. For the transmission of the frequency setpoint and actual value, for instance several parameters with a different resolution can be selected.
- Via the parameters, at the same time the general network activity can be diagnosed.

NetWordIN1 ... NetWordIN4

These four mappable data words are provided to individually control the inverter:

- NetWordIN1: for the implementation of an individual control word format.
- NetWordIN2: for control of the digital outputs via network.
- NetWordIN3 and NetWordIN4: for control of the analog outputs via network.

NetWordOUT1 and NetWordOUT2

These two mappable data words are provided to output status messages to the network master:

- NetWordOUT1: for the implementation of an individual status word format.
- NetWordOUT2: for the output of messages of the "Sequencer" function (in preparation).

Parameter	Name / value range / [default setting]		Info
0x4008:001 *(PAR 590/001)*	Process input words: NETWordIN1 *(NETWordIN stat: NETWordIN1)* 0x0000 ... [**0x0000**] ... 0xFFFF		Mappable data word for flexible control of the inverter via network. Assignment of the functions: • 0x400E:001 *(PAR 505/001)*: assignment of functions to bit 0 ... 15.
	Bit 0	Mapping bit 0	
	Bit 1	Mapping bit 1	
	Bit 2	Mapping bit 2	
	Bit 3	Mapping bit 3	
	Bit 4	Mapping bit 4	
	Bit 5	Mapping bit 5	
	Bit 6	Mapping bit 6	
	Bit 7	Mapping bit 7	
	Bit 8	Mapping bit 8	
	Bit 9	Mapping bit 9	
	Bit 10	Mapping bit 10	
	Bit 11	Mapping bit 11	
	Bit 12	Mapping bit 12	
	Bit 13	Mapping bit 13	
	Bit 14	Mapping bit 14	
	Bit 15	Mapping bit 15	
0x4008:002 *(PAR 590/002)*	Process input words: NETWordIN2 *(NETWordIN stat: NETWordIN2)* 0x0000 ... [**0x0000**] ... 0xFFFF		Mappable data word for optional control of the digital outputs via network. Assignment of the digital outputs: • 0x2634:001 *(PAR 420/001)* = 34 ... 49: assignment of the relay output to bit 0 ... 15. • 0x2634:002 *(PAR 420/002)* = 34 ... 49: assignment of digital output 1 to bit 0 ... 15. • 0x2634:003 *(PAR 420/003)* = 34 ... 49: assignment of digital output 2 to bit 0 ... 15.
	Bit 0	Mapping bit 0	
	Bit 1	Mapping bit 1	
	Bit 2	Mapping bit 2	
	Bit 3	Mapping bit 3	
	Bit 4	Mapping bit 4	
	Bit 5	Mapping bit 5	
	Bit 6	Mapping bit 6	
	Bit 7	Mapping bit 7	

Parameter	Name / value range / [default setting]		Info
	Bit 8	Mapping bit 8	
	Bit 9	Mapping bit 9	
	Bit 10	Mapping bit 10	
	Bit 11	Mapping bit 11	
	Bit 12	Mapping bit 12	
	Bit 13	Mapping bit 13	
	Bit 14	Mapping bit 14	
	Bit 15	Mapping bit 15	
0x4008:003 *(PAR 590/003)*	Process input words: NETWordIN3 *(NETWordIN stat: NETWordIN3)* 0.0 ... [**0.0**] ... 100.0 %		Mappable data word for optional control of an analog output via network. Assignment of the analog outputs: • 0x2639:002 *(PAR 440/002)* = 20: analog output 1 or • 0x263A:002 *(PAR 441/002)* = 20: analog output 2
0x4008:004 *(PAR 590/004)*	Process input words: NETWordIN4 *(NETWordIN stat: NETWordIN4)* 0.0 ... [**0.0**] ... 100.0 %		Mappable data word for optional control of an analog output via network. Assignment of the analog outputs: • 0x2639:002 *(PAR 440/002)* = 21: analog output 1 or • 0x263A:002 *(PAR 441/002)* = 21: analog output 2
0x400A:001 *(PAR 591/001)*	Process output words: NetWordOUT1 *(NetWordOUT Stat: NetWordOUT1)* • Read only		Mappable data word for the output of status messages of the inverter via network. Assignment of the status messages: • 0x2634:010 *(PAR 420/010)*: assignment of status message to bit 0. • 0x2634:011 *(PAR 420/011)*: assignment of status message to bit 1. • ... • 0x2634:025 *(PAR 420/025)*: assignment of status message to bit 15.
	Bit 0	Mapping bit 0	
	Bit 1	Mapping bit 1	
	Bit 2	Mapping bit 2	
	Bit 3	Mapping bit 3	
	Bit 4	Mapping bit 4	
	Bit 5	Mapping bit 5	
	Bit 6	Mapping bit 6	
	Bit 7	Mapping bit 7	
	Bit 8	Mapping bit 8	
	Bit 9	Mapping bit 9	
	Bit 10	Mapping bit 10	
	Bit 11	Mapping bit 11	
	Bit 12	Mapping bit 12	
	Bit 13	Mapping bit 13	
	Bit 14	Mapping bit 14	
	Bit 15	Mapping bit 15	
0x400A:002 *(PAR 591/002)*	Process output words: NetWordOUT2 *(NetWordOUT Stat: NetWordOUT2)* • Read only		Mappable data word for the output of messages of the "Sequencer" function via network.
	Bit 0	Mapping bit 0	
	Bit 1	Mapping bit 1	
	Bit 2	Mapping bit 2	
	Bit 3	Mapping bit 3	
	Bit 4	Mapping bit 4	
	Bit 5	Mapping bit 5	
	Bit 6	Mapping bit 6	
	Bit 7	Mapping bit 7	
	Bit 8	Mapping bit 8	
	Bit 9	Mapping bit 9	
	Bit 10	Mapping bit 10	
	Bit 11	Mapping bit 11	
	Bit 12	Mapping bit 12	
	Bit 13	Mapping bit 13	
	Bit 14	Mapping bit 14	
	Bit 15	Mapping bit 15	

Parameter	Name / value range / [default setting]	Info
0x400B:003 *(PAR 592/003)*	Predefined process input data: Network setpoint frequency [0.1 Hz] *(Legacy NetWordIN: Netwfreq. 0.1Hz)* 0.0 ... [**0.0**] ... 599.0 Hz	Mappable parameter for specifying the frequency setpoint in [0.1 Hz] via network. • The specification is made without sign (irrespective of the rotating direction). • The rotating direction is specified via the control word. • Example: 456 ≡ 45.6 Hz
0x400B:004 *(PAR 592/004)*	Predefined process input data: Network setpoint speed [r/min] *(Legacy NetWordIN: Netwspeed r/min)* 0 ... [**0**] ... 50000 rpm	Mappable parameter for specifying the setpoint as speed in [rpm] via network. • The specification is made without sign (irrespective of the rotating direction). • The rotating direction is specified via the control word. • Example: 456 ≡ 456 rpm
0x400B:005 *(PAR 592/005)*	Predefined process input data: Network setpoint frequency [0.01 Hz] *(Legacy NetWordIN: Netwfreq. 0.01Hz)* 0.00 ... [**0.00**] ... 599.00 Hz	Mappable parameter for specifying the frequency setpoint in [0.01 Hz] via network. • The specification is made without sign (irrespective of the rotating direction). • The rotating direction is specified via the control word. • Example: 456 ≡ 4.56 Hz
0x400C:003 *(PAR 593/003)*	Predefined process output data: Actual frequency [0.1 Hz] *(Old netw. off: Actual frequency Hz)* • Read only: x.x Hz	Mappable parameter for the output of the actual frequency value in [0.1 Hz] via network. • The output is effected without sign (irrespective of the rotating direction). • The rotating direction is specified via the status word. • Example: 456 ≡ 45.6 Hz
0x400C:004 *(PAR 593/004)*	Predefined process output data: Actual motor speed [r/min] *(Old netw. off: Act. speed r/min)* • Read only: x rpm	Mappable parameter for the output of the actual value as speed in [rpm] via network. • The output is effected without sign (irrespective of the rotating direction). • The rotating direction is specified via the status word. • Example: 456 ≡ 456 rpm
0x400C:005 *(PAR 593/005)*	Predefined process output data: Drive status *(Old netw. off: Drive status)* • Read only	Mappable status word (Modbus Legacy Register 2003).
	0 Error (non-resettable)	
	1 Error	
	2 Waiting for start	
	3 Identification not executed	
	4 Controller inhibit	
	5 Stop	
	7 Identification	
	8 Running	
	9 Acceleration	
	10 Deceleration	
	11 Deceleration override	
	12 DC braking	
	13 Flying Start	
	14 Current limit reached	
	16 Process controller idle state	
0x400C:006 *(PAR 593/006)*	Predefined process output data: Actual frequency [0.01 Hz] *(Old netw. off: Act. freq. 0.01Hz)* • Read only: x.xx Hz	Mappable parameter for the output of the actual frequency value in [0.01 Hz] via network. • The output is effected without sign (irrespective of the rotating direction). • The rotating direction is specified via the status word. • Example: 456 ≡ 4.56 Hz

3 Acyclic data exchange

The acyclic data exchange is normally used for transmitting parameter data the transmission of which is not time-critical. Such parameter data are for example operating parameters, motor data, and diagnostic information.

Details

- The acyclic data exchange enables access to all parameters of the inverter.
- For all communication protocols except Modbus, the parameter is addressed directly via the index and subindex.
- The parameter attribute list contains a list of all inverter parameters. This list in particular includes some information that is relevant to the reading and writing of parameters via the network.

4 CANopen

CANopen is an internationally approved communication protocol which is designed for commercial and industrial automation applications. High data transfer rates in connection with efficient data formatting provide for the coordination of motion control devices in multi-axis applications.

Preconditions

Control unit (CU) of the inverter is provided with CANopen.

4.1 CANopen introduction

- The LOVATO Electric implementation of the CANopen communication profile (CiA DS301, version 4.02) enables baud rates from 20 kbps to 1 Mbps.
- For establishing a simple network connection, the inverter provides predefined control and status words for device profile CiA402, AC drive profile and in LOVATO electric format. Additionally, further mappable data words are provided to individually control the inverter.
- The inverter control is preconfigured via a CiA402-compliant control word.

4.2 CANopen node address

Each network node must be provided with a unique node address.

Details

- The node address of the inverter can be optionally set in 0x2301:001 *(PAR 510/001)* or using the DIP switches on the device labelled with "1" ... "64".
- The setting that is active when the inverter is switched on is the effective setting.
- The labelling of the DIP switches corresponds to the values of the individual DIP switches for determining the node address (see the following example).
- The active node address is shown in 0x2302:001 *(PAR 511/001)*.

Example of how the node address is set via the DIP switches

DIP switch	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Setting	OFF	OFF	**ON**	OFF	**ON**	**ON**	**ON**
Value	0	0	**16**	0	**4**	**2**	**1**
Node address	= sum of all values = 16 + 4 + 2 + 1 = **23**						

Parameter	**Name / value range / [default setting]**	**Info**
0x2301:001 *(PAR 510/001)*	CANopen settings: Node ID *(CANopen sett.: Node ID)* 1 ... [**1**] ... 127	Optionally setting of the node address (instead of setting via DIP switches 1 ... 64). • The node address set here only becomes effective if DIP switches 1 ... 64 have been set to OFF before mains switching. • A change in the node address will not be effective until a CAN Reset Node is performed.
0x2302:001 *(PAR 511/001)*	Active CANopen settings: Node ID *(CANopen status: Node ID)* • Read only	Display of the active node address.
0x2303 *(PAR 509)*	DIP switch position *(DIP switch)* • Read only	Display of the DIP switch setting at the last mains power-on.

4.3 CANopen baud rate

All network nodes must be set to the same baud rate.

Details

- The baud rate can be optionally set in 0x2301:002 *(PAR 510/002)* or using the DIP switches on the device labelled with "a" ... "d" (see the following table).
- The setting that is active when the inverter is switched on is the effective setting.
- The active baud rate is shown in 0x2302:002 *(PAR 511/002)*.

d	c	b	a	Baud rate
OFF	ON	OFF	ON	20 kbps
OFF	OFF	ON	ON	50 kbps
OFF	OFF	ON	OFF	125 kbps
OFF	OFF	OFF	ON	250 kbps
OFF	**OFF**	**OFF**	**OFF**	Baud rate set in 0x2301:002 *(PAR 510/002)* (**500 kbps**)
OFF	ON	OFF	OFF	1 Mbps
When a combination is set that is not in the list, the baud rate is set to 500 kbps.				

Parameter	Name / value range / [default setting]		Info
0x2301:002 *(PAR 510/002)*	CANopen settings: Baud rate *(CANopen sett.: Baud rate)*		Optionally, setting of the baud rate (instead of setting via DIP switches a ... d). • The baud rate parameterised is only effective if DIP switches a ... d and 1 ... 64 were set to before mains switching. • A change in the baud rate will not be effective until a CAN reset node is performed.
	1	20 kbps	
	2	50 kbps	
	3	125 kbps	
	4	250 kbps	
	5	**500 kbps**	
	6	800 kbps	
	7	1 Mbps	
0x2302:002 *(PAR 511/002)*	Active CANopen settings: Baud rate *(CANopen status: Baud rate)* • Read only		Display of the active baud rate.
	0	Automatic	
	1	20 kbps	
	2	50 kbps	
	3	125 kbps	
	4	250 kbps	
	5	500 kbps	
	6	800 kbps	
	7	1 Mbps	

4.4 CANopen initialisation

If the initialisation of the CANopen network and the associated status change from "Pre-Operational" to "Operational" is not effected by a higher-level host system, the inverter can instead be defined as a "quasi" master to execute this task.

Details

Configuration of the inverter as CAN master is carried out in 0x2301:003 *(PAR 510/003)*. As CAN master, the controller sets all nodes connected to the bus (broadcast telegram) to the "Operational" communication state using the "Start remote node" NMT telegram. Only this communication state enables data exchange via the process data objects.

A change in master/slave operation only becomes effective by repeated mains switching of the inverter or by transmitting the "Reset Node" or "Reset Communication" NMT telegram to the inverter. As an alternative to the "Reset Node" NMT telegram, the "Reset network node" 0x2022:016 *(PAR 700/016)* device command is provided for reinitialisation of the CAN-specific device parameters.

Parameter	Name / value range / [default setting]		Info
0x2301:003 *(PAR 510/003)*	CANopen settings: Slave/Master *(CANopen sett.: Slave/Master)*		1 = after mains switching, inverter starts as CAN master.
	0	**Slave**	
	1	Mini-master	
0x2301:004 *(PAR 510/004)*	CANopen settings: Start of remote deceleration *(CANopen sett.: Start rem. dec.)* 0 ... [**3000**] ... 65535 ms		If the inverter has been defined as CAN master, a delay time can be set here, which has to elapse after mains switching before the inverter deposits the "Start Remote Node" NMT telegram on the CAN bus.

4.5 CANopen diagnostics

For the purpose of diagnostics, the inverter provides several status words via which the CAN bus status, the CAN bus controller status, and the status of different time monitoring functions can be queried.

Parameter	Name / value range / [default setting]		Info
0x2307 *(PAR 515)*	CANopen time-out status *(Time-out status)* • Read only		Bit-coded status display of the CAN time monitoring functions.
	Bit 0	RPDO1-Timeout	1 ≡ RPDO1 was not received within the monitoring time or not with the sync configured. • Status is reset automatically after the RPDO has been received again. • Setting of monitoring time for RPDO1 in 0x1400:005 *(PAR 540/005)*.
	Bit 1	RPDO2-Timeout	1 ≡ RPDO2 was not received within the monitoring time or not with the sync configured. • Status is reset automatically after the RPDO has been received again. • Setting of monitoring time for RPDO2 in 0x1401:005 *(PAR 541/005)*.
	Bit 2	RPDO3-Timeout	1 ≡ RPDO3 was not received within the monitoring time or not with the sync configured. • Status is reset automatically after the RPDO has been received again. • Setting of monitoring time for RPDO3 in 0x1402:005 *(PAR 542/005)*.
	Bit 3	Reserved	-
	Bit 4		
	Bit 5		
	Bit 6		
	Bit 7		
	Bit 8	Heartbeat-Timeout Consumer 1	1 ≡ within the "Heartbeat Consumer Time", no heartbeat telegram was received from node 1 to be monitored. • Status can only be reset by mains switching. • "Heartbeat Consumer Time" setting in 0x1016:001 *(PAR 520/001)*.
	Bit 9	Heartbeat-Timeout Consumer 2	1 ≡ within the "Heartbeat Consumer Time", no heartbeat telegram was received from node 2 to be monitored. • Status can only be reset by mains switching. • "Heartbeat Consumer Time" setting in 0x1016:002 *(PAR 520/002)*.

Parameter	Name / value range / [default setting]		Info
	Bit 10	Heartbeat-Timeout Consumer 3	1 ≡ within the "Heartbeat Consumer Time", no heartbeat telegram was received from node 3 to be monitored. • Status can only be reset by mains switching. • "Heartbeat Consumer Time" setting in 0x1016:003 *(PAR 520/003)*.
	Bit 11	Heartbeat-Timeout Consumer 4	1 ≡ within the "Heartbeat Consumer Time", no heartbeat telegram was received from node 4 to be monitored. • Status can only be reset by mains switching. • "Heartbeat Consumer Time" setting in 0x1016:004 *(PAR 520/004)*.
	Bit 12	Reserved	-
	Bit 13		
	Bit 14		
	Bit 15		
	Bit 16		
	Bit 17		
	Bit 18		
	Bit 19		
	Bit 20		
	Bit 21		
	Bit 22		
	Bit 23		
	Bit 24		
	Bit 25		
	Bit 26		
	Bit 27		
	Bit 28		
	Bit 29		
	Bit 30		
	Bit 31		
0x2308 *(PAR 516)*	CANopen status *(CANopen status)* • Read only		Display of the current CAN bus state.
	0	Initialisation	CAN bus initialisation active. • The initialisation is started automatically at mains connection. During this phase, the inverter us not involved in the data exchange process on the CAN bus. • The standard values are re-written to all CAN-relevant parameters. • When the initialisation process has been completed, the inverter automatically adopts the "Pre-Operational" state.
	1	Reset node	"Reset Node" NMT command active. • Initialisation of all CAN-relevant parameters with the values stored.
	2	Reset communication	"Reset Communication" NMT command active. • Initialisation of all CAN-relevant parameters with the values stored.
	4	Stopped	Only network management telegrams can be received.
	5	Operational	Parameter data and process data can be received.
	127	Pre-Operational	Parameter data can be received, process data are ignored.
0x2309 *(PAR 517)*	CANopen controller status *(Controller status)* • Read only		Status display of the internal CANopen controller.
	1	Error active	The inverter is a fully-fledged communication node at the CANopen network. It is able to transmit and receive data and to report faults.
	2	Error passive	The inverter can only passively indicate faulty reception via the ACK field.
	3	Bus off	The inverter is electrically separated from the CANopen network. In order to exit this state, the CANopen interface must be reset.

4.6 CANopen emergency telegram

If the error status changes when an internal device error occurs or is remedied, an emergency telegram is sent to the NMT master once.

Details

- The identifier for the emergency telegram is fixedly defined and is shown in 0x1014.
- In 0x1015, a blocking time can be set, in order to limit the bus load in the case of emergency telegrams following quickly in succession.

Parameter	Name / value range / [default setting]	Info
0x1014	COB-ID EMCY • Read only	Display of the identifier for emergency telegrams.
0x1015	Inhibit time EMCY 0.0 ... [**0.0**] ... 6553.5 ms	Blocking time which can be set in order to limit the bus load in the case of emergency telegrams following quickly in succession.

4.7 CANopen heartbeat protocol

The heartbeat protocol can be used for node monitoring purposes within a CAN network.

Details

Basic procedure:

1. A heartbeat producer cyclically sends a heartbeat telegram to one or several receivers (consumers).
2. The consumer(s) monitor(s) the heartbeat telegram for arrival on a regular basis.

The inverter can be configured as producer or as consumer to monitor up to four other nodes.

Parameter	Name / value range / [default setting]	Info
0x1016:000 *(PAR 520/000)*	Consumer heartbeat time: Highest subindex *(Cons heartbeat: Highest subindex)* • Read only	Number of nodes to be monitored.
0x1016:001 *(PAR 520/001)*	Consumer heartbeat time: Node 1 *(Cons heartbeat: Node 1)* 0x00000000 ... [**0x00000000**] ... 0x00FFFFFF	Node ID and heartbeat time of node 1 which is to be monitored. • Format: 0x00nnhhhh (nn = node ID, hhhh = heartbeat time in [ms])
0x1016:002 *(PAR 520/002)*	Consumer heartbeat time: Node 2 *(Cons heartbeat: Node 2)* 0x00000000 ... [**0x00000000**] ... 0x00FFFFFF	Node ID and heartbeat time of node 2 which is to be monitored. • Format: 0x00nnhhhh (nn = node ID, hhhh = heartbeat time in [ms])
0x1016:003 *(PAR 520/003)*	Consumer heartbeat time: Node 3 *(Cons heartbeat: Node 3)* 0x00000000 ... [**0x00000000**] ... 0x00FFFFFF	Node ID and heartbeat time of node 3 which is to be monitored. • Format: 0x00nnhhhh (nn = node ID, hhhh = heartbeat time in [ms])
0x1016:004 *(PAR 520/004)*	Consumer heartbeat time: Node 4 *(Cons heartbeat: Node 4)* 0x00000000 ... [**0x00000000**] ... 0x00FFFFFF	Node ID and heartbeat time of node 4 which is to be monitored. • Format: 0x00nnhhhh (nn = node ID, hhhh = heartbeat time in [ms])
0x1017 *(PAR 522)*	Producer heartbeat time *(Prod heartbeat)* 0 ... [**0**] ... 65535 ms	Time interval for the transmission of the heartbeat telegram to the consumer(s). • The heartbeat telegram is sent automatically as soon as a time > 0 ms is set. • The time set is rounded down to an integer multiple of 5 ms.

4.8 CANopen process data objects

Process data objects (PDOs) are used for the cyclic transmission of (process) data via CANopen. PDOs only contain data and an identifier. They do not contain any information about the sender or receiver and are therefore very efficient.

Details

- Process data objects which the inverter receives via the network are referred to as "Receive PDOs" (RPDOs).
- Process data objects which the inverter sends via the network are referred to as "Transmit PDOs" (TPDOs).
- The maximum length of a PDO is 8 bytes (4 data words).
- Each PDO requires a unique identifier ("COB-ID") for the purpose of identification within the network.
- Furthermore the transmission type must be defined for TPDOs (see the following section).
- Communication parameters such as the transmission type and cycle time for each PDO can be set freely and independently of the settings of other PDOs

Transmission type

Process data objects can be transmitted in an event-controlled or time-controlled manner. The below table shows that it is possible to combine the different methods by means of logic operations (AND, OR):

- Event-controlled: The PDO is sent if a special device-internal event has occurred, for instance, if the data contents of the TPDO have changed or if a transmission cycle time has elapsed.
- Synchronous transmission: Transmission of a TPDOs or reception of an RPDO is effected after the inverter has received a sync telegram (COB-ID 0x80).
- Cyclic transmission: The cyclic transmission of PDOs is effected when the transmission cycle time has elapsed.
- Polled via RTR: Transmission of a TPDO is carried out on request by another device via data request frame (RTR remote transmit request). For this, the data requester (e.g. master) sends the data request frame with the COB-ID of the TPDO that is to be requested to transmit. The receiver recognises the RTR and carries out the transmission.

Transmission type	PDO transmission			Logic combination of different transmission types
	cyclic	synchronous	event-controlled	
0		●	●	AND
1 ... 240		●		-
254, 255	●		●	OR

Transmission type	Description
0	Synchronous and acyclic • The PDO is transmitted on an event-controlled basis with every sync (e.g. when a bit change occurs in the PDO).
1 ... 240	Synchronous and cyclic (sync-controlled with a response) • Selection n = 1: The PDO is transmitted with every sync. • Selection 1 < n ≤ 240: The PDO is transmitted with every n-th sync.
241 ... 251	Reserved
252	Synchronous - RTR only
253	Asynchronous - RTR only
254, 255	Asynchronous - manufacturer-specific / device profile-specific • If one of these values is entered, the PDO is transferred in an event-controlled or cyclic manner. (The values "254" and "255" are equivalent). • For a cyclic transmission, a cycle time must be entered for the respective PDO. In this case, cyclic transmission takes place in addition to event-controlled transmission.

Synchronisation of PDOs via sync telegram

During cyclic transmission, one or more PDOs are transmitted/received in fixed time intervals. An additional specific telegram, the so-called sync telegram, is used for synchronising cyclic process data.

- The sync telegram is the trigger point for the transmission of process data from the slaves to the master and for the acceptance of process data from the master in the slaves.
- For sync-controlled process data processing, the sync telegram must be generated accordingly.
- The response to a sync telegram is determined by the transmission type selected.

Generating the sync telegram:

- 0x1005 can be used to activate the generation of sync telegrams and to write the identifier value.
- Sync telegrams are created when bit 30 (see below) is set to "1".
- The interval between sync telegrams is to be set in 0x1006.

Writing identifiers:

- To receive PDOs, the value 0x80 must be entered in the 11-bit identifier in the LOVATO Electric setting (and in compliance with the CANopen specification). This means that all inverters are set to the same sync telegram by default.
- If sync telegrams are only to be received by specific nodes, their identifiers can be entered with a value of up to and including 0x07FF.
- The identifier can only be changed if the inverter does not send any sync telegrams (0x1005, Bit 30 = "0").

Data telegram assignment

<table>
<tr><th colspan="3">8th byte (data 4)</th><th>7th byte (data 3)</th><th colspan="2">6th byte (data 2)</th><th>5th byte (data 1)</th></tr>
<tr><td>Bit 31</td><td>Bit 30</td><td colspan="3">Bit 29 ... bit 11</td><td colspan="2">Bit 10 ... bit 0</td></tr>
<tr><td>x</td><td>0/1</td><td colspan="3">Extended identifier*</td><td colspan="2">11-bit identifier</td></tr>
<tr><td colspan="7">* The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0".</td></tr>
</table>

Parameter	Name / value range / [default setting]	Info
0x1005	COB-ID SYNC 0x00000000 ... [**0x00000080**] ... 0xFFFFFFFF	Identifier for sync telegram. How to change the identifier: 1. Deactivate Sync: Set bit 30 to "0". 2. Change identifier. 3. Activate Sync: Set bit 30 to "1".
0x1006	Communication cyclic period 0 ... [**0**] ... 65535000 us	Cycle time for sync telegrams. • A setting of "1000" or integer multiples of this settings are possible. • With the setting "0", no sync telegrams are generated.
0x1400:000	RPDO1 communication parameter: Highest subindex • Read only	
0x1400:001 *(PAR 540/001)*	RPDO1 communication parameter: COB-ID *(RPDO1 config.: COB-ID)* 0x00000000 ... [**0x00000200**] ... 0xFFFFFFFF	RPDO1: identifier How to change the identifier: 1. Set PDO to "invalid": Set bit 31 to "1". 2. Change identifier. 3. Reset PDO to "valid": Set bit 31 to "0".
0x1400:002 *(PAR 540/002)*	RPDO1 communication parameter: Transmission type *(RPDO1 config.: Transm. type)* 0 ... [**255**] ... 255	RPDO1: transmission type in compliance with DS301 V4.02
0x1400:005 *(PAR 540/005)*	RPDO1 communication parameter: Event timer *(RPDO1 config.: Event timer)* 0 ... [**100**] ... 65535 ms	RPDO1: time-out for the monitoring of data reception.
0x1401:001 *(PAR 541/001)*	RPDO2 communication parameter: COB-ID *(RPDO2 config.: COB-ID)* 0x00000000 ... [**0x80000300**] ... 0xFFFFFFFF	RPDO2: identifier How to change the identifier: 1. Set PDO to "invalid": Set bit 31 to "1". 2. Change identifier. 3. Reset PDO to "valid": Set bit 31 to "0".
0x1401:002 *(PAR 541/002)*	RPDO2 communication parameter: Transmission type *(RPDO2 config.: Transm. type)* 0 ... [**255**] ... 255	RPDO2: transmission type in compliance with DS301 V4.02

Parameter	Name / value range / [default setting]	Info
0x1401:005 *(PAR 541/005)*	RPDO2 communication parameter: Event timer *(RPDO2 config.: Event timer)* 0 ... [**100**] ... 65535 ms	RPDO2: time-out for the monitoring of data reception.
0x1402:001 *(PAR 542/001)*	RPDO3 communication parameter: COB-ID *(RPDO3 config.: COB-ID)* 0x00000000 ... [**0x80000400**] ... 0xFFFFFFFF	RPDO3: identifier How to change the identifier: 1. Set PDO to "invalid": Set bit 31 to "1". 2. Change identifier. 3. Reset PDO to "valid": Set bit 31 to "0".
0x1402:002 *(PAR 542/002)*	RPDO3 communication parameter: Transmission type *(RPDO3 config.: Transm. type)* 0 ... [**255**] ... 255	RPDO3: transmission type in compliance with DS301 V4.02
0x1402:005 *(PAR 542/005)*	RPDO3 communication parameter: Event timer *(RPDO3 config.: Event timer)* 0 ... [**100**] ... 65535 ms	RPDO3: time-out for the monitoring of data reception.
0x1800:000	TPDO1 communication parameter: Highest subindex • Read only	The value "5" is permanently set.
0x1800:001 *(PAR 550/001)*	TPDO1 communication parameter: COB-ID *(TPDO1 config.: COB-ID)* 0x00000001 ... [**0x40000180**] ... 0xFFFFFFFF	TPDO1: identifier How to change the identifier: 1. Set PDO to "invalid": Set bit 31 to "1". 2. Change identifier. 3. Reset PDO to "valid": Set bit 31 to "0".
0x1800:002 *(PAR 550/002)*	TPDO1 communication parameter: Transmission type *(TPDO1 config.: Transm. type)* 0 ... [**255**] ... 255	TPDO1: transmission type in compliance with DS301 V4.02
0x1800:003 *(PAR 550/003)*	TPDO1 communication parameter: Inhibit time *(TPDO1 config.: Inhibit time)* 0.0 ... [**0.0**] ... 6553.5 ms	TPDO1: minimum time between the transmission of two identical PDOs (see DS301 V4.02).
0x1800:005 *(PAR 550/005)*	TPDO1 communication parameter: Event timer *(TPDO1 config.: Event timer)* 0 ... [**20**] ... 65535 ms	TPDO1: cycle time for PDO transmission with transmission type "254".
0x1801:000	TPDO2 communication parameter: Highest subindex • Read only	The value "5" is permanently set.
0x1801:001 *(PAR 551/001)*	TPDO2 communication parameter: COB-ID *(TPDO2 config.: COB-ID)* 0x00000001 ... [**0xC0000280**] ... 0xFFFFFFFF	TPDO2: identifier How to change the identifier: 1. Set PDO to "invalid": Set bit 31 to "1". 2. Change identifier. 3. Reset PDO to "valid": Set bit 31 to "0".
0x1801:002 *(PAR 551/002)*	TPDO2 communication parameter: Transmission type *(TPDO2 config.: Transm. type)* 0 ... [**255**] ... 255	TPDO2: transmission type in compliance with DS301 V4.02
0x1801:003 *(PAR 551/003)*	TPDO2 communication parameter: Inhibit time *(TPDO2 config.: Inhibit time)* 0.0 ... [**0.0**] ... 6553.5 ms	TPDO2: minimum time between the transmission of two identical PDOs (see DS301 V4.02).
0x1801:005 *(PAR 551/005)*	TPDO2 communication parameter: Event timer *(TPDO2 config.: Event timer)* 0 ... [**0**] ... 65535 ms	TPDO2: cycle time for PDO transmission with transmission type "254".
0x1802:000	TPDO3 communication parameter: Highest subindex • Read only	The value "5" is permanently set.
0x1802:001 *(PAR 552/001)*	TPDO3 communication parameter: COB-ID *(TPDO3 config.: COB-ID)* 0x00000001 ... [**0xC0000380**] ... 0xFFFFFFFF	TPDO3: identifier How to change the identifier: 1. Set PDO to "invalid": Set bit 31 to "1". 2. Change identifier. 3. Reset PDO to "valid": Set bit 31 to "0".
0x1802:002 *(PAR 552/002)*	TPDO3 communication parameter: Transmission type *(TPDO3 config.: Transm. type)* 0 ... [**255**] ... 255	TPDO3: transmission type in compliance with DS301 V4.02
0x1802:003 *(PAR 552/003)*	TPDO3 communication parameter: Inhibit time *(TPDO3 config.: Inhibit time)* 0.0 ... [**0.0**] ... 6553.5 ms	TPDO3: minimum time between the transmission of two identical PDOs (see DS301 V4.02).

Parameter	Name / value range / [default setting]	Info
0x1802:005 *(PAR 552/005)*	TPDO3 communication parameter: Event timer *(TPDO3 config.: Event timer)* 0 ... [**0**] ... 65535 ms	TPDO3: cycle time for PDO transmission with transmission type "254".

4.9 CANopen data mapping

Data mapping serves to define which process data are transmitted cyclically via the process data channels.

Details

Data mapping (in the case of CANopen also referred to as "PDO mapping") is preconfigured for control of the inverter via the CiA402 device profile:

- RPDO1 = Controlword 0x6040 and Target velocity 0x6042 *(PAR 781)*.
- TPDO1 = Statusword 0x6041 *(PAR 780)* and Velocity actual value 0x6044 *(PAR 783)*.

Variable PDO mapping

For individual drive solutions, the inverter supports "variable PDO mapping", providing 8 mapping entries in each case to assign 8-bit, 16-bit, and 32-bit parameters to a PDO in an optional order. The total length of the parameters mapped, however, must not exceed 8 bytes.

The process of PDO mapping cannot be applied to all parameters.

The process of variable PDO mapping only allows the following procedure:

1. Set PDO to "invalid": set bit 31 in the corresponding identifier (0x1400:1 ... 0x1402:1 or 0x1800:1 ... 0x1802:1) to "1".
2. Set PDO mapping to "invalid": set subindex 0 in the mapping parameter (0x1600 ... 0x1602 or 0x1A00 ... 0x1A02) to "0".
3. Set desired PDO mapping via the corresponding mapping entries.
   format: 0xiiiissll (iiii = hexadecimal index, ss = hexadecimal subindex, ll = hexadecimal data length)
4. Set subindex 0 in the mapping parameter (0x1600 ... 0x1602 or 0x1A00 ... 0x1A02) to a valid value (number of parameters mapped).
5. Reset PDO to "valid": set bit 31 in the corresponding identifier (0x1400:1 ... 0x1402:1 or 0x1800:1 ... 0x1802:1) to "0".

Parameter	Name / value range / [default setting]	Info
0x1600:000	RPDO1 mapping parameter: Highest subindex 0 ... [**2**] ... 8	Number of objects mapped in RPDO1.
0x1600:001	RPDO1 mapping parameter: Entry 1 0x00000000 ... [**0x60400010**] ... 0xFFFFFFFF	Mapping entry 1 for RPDO1.
0x1600:002	RPDO1 mapping parameter: Entry 2 0x00000000 ... [**0x60420010**] ... 0xFFFFFFFF	Mapping entry 2 for RPDO1.
0x1600:003	RPDO1 mapping parameter: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for RPDO1.
0x1600:004	RPDO1 mapping parameter: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for RPDO1.
0x1600:005	RPDO1 mapping parameter: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for RPDO1.
0x1600:006	RPDO1 mapping parameter: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for RPDO1.
0x1600:007	RPDO1 mapping parameter: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for RPDO1.
0x1600:008	RPDO1 mapping parameter: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for RPDO1.
0x1601:000	RPDO2 mapping parameter: Highest subindex 0 ... [**0**] ... 8	Number of objects mapped in RPDO2.

Parameter	Name / value range / [default setting]	Info
0x1601:001	RPDO2 mapping parameter: Entry 1 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 1 for RPDO2.
0x1601:002	RPDO2 mapping parameter: Entry 2 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 2 for RPDO2.
0x1601:003	RPDO2 mapping parameter: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for RPDO2.
0x1601:004	RPDO2 mapping parameter: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for RPDO2.
0x1601:005	RPDO2 mapping parameter: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for RPDO2.
0x1601:006	RPDO2 mapping parameter: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for RPDO2.
0x1601:007	RPDO2 mapping parameter: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for RPDO2.
0x1601:008	RPDO2 mapping parameter: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for RPDO2.
0x1602:000	RPDO3 mapping parameter: Highest subindex 0 ... [**0**] ... 8	Number of objects mapped in RPDO3.
0x1602:001	RPDO3 mapping parameter: Entry 1 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 1 for RPDO3.
0x1602:002	RPDO3 mapping parameter: Entry 2 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 2 for RPDO3.
0x1602:003	RPDO3 mapping parameter: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for RPDO3.
0x1602:004	RPDO3 mapping parameter: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for RPDO3.
0x1602:005	RPDO3 mapping parameter: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for RPDO3.
0x1602:006	RPDO3 mapping parameter: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for RPDO3.
0x1602:007	RPDO3 mapping parameter: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for RPDO3.
0x1602:008	RPDO3 mapping parameter: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for RPDO3.
0x1A00:000	TPDO1 mapping parameter: Highest subindex 0 ... [**2**] ... 8	Number of objects mapped in TPDO1.
0x1A00:001	TPDO1 mapping parameter: Entry 1 0x00000000 ... [**0x60410010**] ... 0xFFFFFFFF	Mapping entry 1 for TPDO1.
0x1A00:002	TPDO1 mapping parameter: Entry 2 0x00000000 ... [**0x60440010**] ... 0xFFFFFFFF	Mapping entry 2 for TPDO1.
0x1A00:003	TPDO1 mapping parameter: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for TPDO1.
0x1A00:004	TPDO1 mapping parameter: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for TPDO1.
0x1A00:005	TPDO1 mapping parameter: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for TPDO1.
0x1A00:006	TPDO1 mapping parameter: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for TPDO1.
0x1A00:007	TPDO1 mapping parameter: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for TPDO1.
0x1A00:008	TPDO1 mapping parameter: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for TPDO1.
0x1A01:000	TPDO2 mapping parameter: Highest subindex 0 ... [**0**] ... 8	Number of objects mapped in TPDO2.
0x1A01:001	TPDO2 mapping parameter: Entry 1 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 1 for TPDO2.
0x1A01:002	TPDO2 mapping parameter: Entry 2 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 2 for TPDO2.
0x1A01:003	TPDO2 mapping parameter: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for TPDO2.

Parameter	Name / value range / [default setting]	Info
0x1A01:004	TPDO2 mapping parameter: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for TPDO2.
0x1A01:005	TPDO2 mapping parameter: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for TPDO2.
0x1A01:006	TPDO2 mapping parameter: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for TPDO2.
0x1A01:007	TPDO2 mapping parameter: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for TPDO2.
0x1A01:008	TPDO2 mapping parameter: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for TPDO2.
0x1A02:000	TPDO3 mapping parameter: Highest subindex 0 ... [**0**] ... 8	Number of objects mapped in TPDO3.
0x1A02:001	TPDO3 mapping parameter: Entry 1 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 1 for TPDO3.
0x1A02:002	TPDO3 mapping parameter: Entry 2 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 2 for TPDO3.
0x1A02:003	TPDO3 mapping parameter: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for TPDO3.
0x1A02:004	TPDO3 mapping parameter: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for TPDO3.
0x1A02:005	TPDO3 mapping parameter: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for TPDO3.
0x1A02:006	TPDO3 mapping parameter: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for TPDO3.
0x1A02:007	TPDO3 mapping parameter: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for TPDO3.
0x1A02:008	TPDO3 mapping parameter: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for TPDO3.

4.10 CANopen service data objects

Service data objects (SDOs) make it possible to read and write all parameters of the inverter via CANopen.

Details

- Two independent SDO channels are provided at the same time. SDO channel 1 is always active. SDO channel 2 can be activated via 0x2301:005 *(PAR 510/005)*.
- The identifiers for SDO1 and SDO2 are generated from the basic identifier (in compliance with the "Predefined Connection Set") and the node address set.
- An SDO is always transmitted with confirmation, i. e. the reception of an SDO frame is acknowledged by the receiver.

Structure of the SDO frame user data

The user data are shown in Motorola format:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	**Index**		**Subindex**	**Data 1**	**Data 2**	**Data 3**	**Data 4**
See table below.	LOW byte	HIGH byte		LOW word		HIGH word	
	Address of the parameter to be read or written.			LOW byte	HIGH byte	LOW byte	HIGH byte

The following commands can be transmitted or received for writing and reading the parameters:

Command	1st byte		Data length	Info
	hex	**dec**		
Write request	0x23	35	4 bytes	Writing of a parameter to the inverter.
	0x2B	43	2 bytes	
	0x2F	47	1 byte	
	0x21	33	Block	
Write response	0x60	96	4 bytes	Inverter acknowledges a write request.
Read request	0x40	64	4 bytes	Reading of a parameter from the inverter.

Command	1st byte		Data length	Info
	hex	dec		
Read response	0x43	67	4 bytes	Inverter response to a read request with the current parameter value.
	0x4B	75	2 bytes	
	0x4F	79	1 byte	
	0x41	65	Block	
Error response	0x80	128	4 bytes	Inverter response to the incorrect execution of the read/write request.

More precisely, the command byte comprises the following information:

Command	1st byte							
	Command specifier (cs)			Toggle (t)	Length*		e	s
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write request	0	0	1	0	0/1	0/1	1	1
Write response	0	1	1	0	0	0	0	0
Read request	0	1	0	0	0	0	0	0
Read response	0	1	0	0	0/1	0/1	1	1
Error response	1	0	0	0	0	0	0	0
*Bit coding of the length: 00 = 4 bytes, 01 = 3 bytes, 10 = 2 bytes, 11 = 1 byte e: expedited (shortened block service) s: segmented (normal block service)								

More commands are defined in the DS301 V4.02 CANopen specification (e.g. segmented transfer).

Maximally 4 bytes are available for parameter value entries. Depending on the data format, they are assigned as follows:

5th byte	6th byte	7th byte	8th byte
Parameter value (1 byte)	0x00	0x00	0x00
Parameter value (2 bytes)		0x00	0x00
LOW byte	HIGH byte		
Parameter value (4 bytes)			
LOW word		HIGH word	
LOW byte	HIGH byte	LOW byte	HIGH byte

The parameter attribute list in the annex also specifies a scaling factor. The scaling factor is relevant to the transmission of parameter values which are represented with one or several decimal positions in the parameter list. If the scaling factor is > 1, before the transmission, the value must be multiplied with the scaling factor specified, so that the value can be transferred completely (as an integer value). On the SDO-client side, the integer value must then be divided by the scaling factor again, in order to receive the original value with decimal positions.

Parameter	Name / value range / [default setting]	Info
0x1200:000	SDO1 server parameter: Highest subindex • Read only	
0x1200:001	SDO1 server parameter: COB-ID client -> server (RX) • Read only	Display of the receive identifier for SDO server channel 1 (basic SDO channel). • According to DS301 V4.02, the basic SDO channel can neither be changed nor deactivated.
0x1200:002	SDO1 server parameter: COB-ID server -> client (TX) • Read only	Display of the transmit identifier for SDO server channel 1 (basic SDO channel). • According to DS301 V4.02, the basic SDO channel can neither be changed nor deactivated.
0x1201:000	SDO2 server parameter: Highest subindex • Read only	

Parameter	Name / value range / [default setting]	Info
0x1201:001	SDO2 server parameter: COB-ID client -> server (RX) 0x00000000 ... [**0x80000640**] ... 0xFFFFFFFF	Specification of the receive identifier for SDO server channel 2. • If SDO server channel 2 is activated via 0x2301:005 *(PAR 510/005)*, this parameter is set to the value "node address + 0x640". This default setting can be changed.
0x1201:002	SDO2 server parameter: COB-ID server -> client (TX) 0x00000000 ... [**0x800005C0**] ... 0xFFFFFFFF	Specification of the transmit identifier for SDO server channel 2. • If SDO server channel 2 is activated via 0x2301:005 *(PAR 510/005)*, this parameter is set to the value "node address + 0x5C0". This default setting can be changed.
0x1201:003	SDO2 server parameter: SDO client node ID 1 ... [**0**] ... 127	Specification of the node address for the SDO client.
0x2301:005 *(PAR 510/005)*	CANopen settings: Activate SDO2 channel *(CANopen sett.: SDO2 config.)*	1 = activate SDO server channel 2.
	0 **Not active**	
	1 Active	

4.11 CANopen error responses

The response to CANopen errors such as missing PDOs or heartbeat frames can be configured via the following parameters.

Parameter	Name / value range / [default setting]	Info
0x1029:000	Error behavior: Highest subindex • Read only	
0x1029:001	Error behavior: Communication error	Selection of the NMT state to which the inverter is to change automatically if a failure of a CANopen node or an internal error is detected in the "Operational" state. These also include the following communication errors: • Change-over of the CAN interface to the "Bus-off" state. • Occurrence of a "Life Guarding Event". • Occurrence of a "Heartbeat Event".
	0 **Status -> Pre-operational**	In the "Pre-operational" state, network management, sync, and emergency telegrams as well as parameter data can be received; process data, however, are ignored.
	1 No status change	
	2 Status -> Stopped	In the "Stopped" state, only network management telegrams can be received.
0x2857:001	CANopen monitoring: RPDO1-Timeout • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response to triggering the RPDO1 time monitoring.
	3 **Error**	
0x2857:002	CANopen monitoring: RPDO2-Timeout • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response to triggering the RPDO2 time monitoring.
	3 **Error**	
0x2857:003	CANopen monitoring: RPDO3-Timeout • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response to triggering the RPDO3 time monitoring.
	3 **Error**	
0x2857:005	CANopen monitoring: Heartbeat-Timeout Consumer 1 • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response with "Heartbeat Event" in consumer 1.
	3 **Error**	
0x2857:006	CANopen monitoring: Heartbeat-Timeout Consumer 2 • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response with "Heartbeat Event" in consumer 2.
	3 **Error**	
0x2857:007	CANopen monitoring: Heartbeat-Timeout Consumer 3 • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response with "Heartbeat Event" in consumer 3.
	3 **Error**	

Parameter	Name / value range / [default setting]	Info
0x2857:008	CANopen monitoring: Heartbeat-Timeout Consumer 4 • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response with "Heartbeat Event" in consumer 4.
	3 Error	
0x2857:010	CANopen monitoring: "Bus-off" state change • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response to changing to the "Bus off" state.
	2 Trouble	
0x2857:011	CANopen monitoring: Warning	Selection of the response that is executed in the case of too many incorrectly sent or received CAN telegrams (> 96).
	0 No response	
	1 Warning	
	2 Trouble	
	3 Error	

4.12 CANopen diagnostic counter

The following parameters serve to diagnose the communication activities between the inverter and the CANopen network. The counters are free-running, i. e. when the maximum value has been reached, the respective counter starts at 0 again.

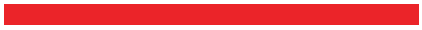
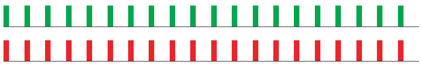
Parameter	Name / value range / [default setting]	Info
0x230A:000	CANopen statistics: Highest subindex • Read only	Number of frame and error counters.
0x230A:001 *(PAR 580/001)*	CANopen statistics: PDO1 received *(CAN counter: PDO1 received)* • Read only	Display of the number of PDO1 telegrams received.
0x230A:002 *(PAR 580/002)*	CANopen statistics: PDO2 received *(CAN counter: PDO2 received)* • Read only	Display of the number of PDO2 telegrams received.
0x230A:003 *(PAR 580/003)*	CANopen statistics: PDO3 received *(CAN counter: PDO3 received)* • Read only	Display of the number of PDO3 telegrams received.
0x230A:005 *(PAR 580/005)*	CANopen statistics: PDO1 transmitted *(CAN counter: PDO1 transmitted)* • Read only	Display of the number of PDO1 telegrams sent.
0x230A:006 *(PAR 580/006)*	CANopen statistics: PDO2 transmitted *(CAN counter: PDO2 transmitted)* • Read only	Display of the number of PDO2 telegrams sent.
0x230A:007 *(PAR 580/007)*	CANopen statistics: PDO3 transmitted *(CAN counter: PDO3 transmitted)* • Read only	Display of the number of PDO3 telegrams sent.
0x230A:009 *(PAR 580/009)*	CANopen statistics: SDO1 telegrams *(CAN counter: SDO1 counter)* • Read only	Display of the number of SDO1 telegrams.
0x230A:010 *(PAR 580/010)*	CANopen statistics: SDO2 telegrams *(CAN counter: SDO2 counter)* • Read only	Display of the number of SDO2 telegrams.
0x230B *(PAR 518)*	CANopen error counter *(CAN error counter)* • Read only	Display of the total number of CAN faults that have occurred.

4.13 CANopen LED status displays

Information about the CAN bus status can be obtained quickly via the "CAN-RUN" and "CAN-ERR" LED displays on the front of the inverter.

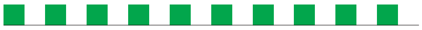
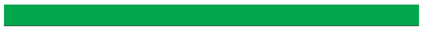
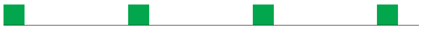
The meaning can be seen from the tables below.

Inverter not active on the CAN bus (yet)

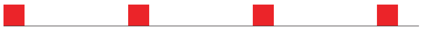
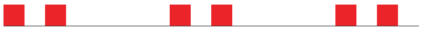
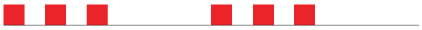
LED display	Meaning
 (CAN-ERR is permanently lit)	Inverter not active on the CAN bus / "Bus Off" state.
 (CAN-RUN and CAN-ERR are flickering)	Automatic baud rate detection active.

Inverter active on the CAN bus

The "CAN-RUN" LED indicates the CANopen state:

LED display	CANopen state
 (CAN-RUN is blinking every 0.2 seconds)	Pre-Operational
 (CAN-RUN and CAN-ERR are flickering)	Operational
 (CAN-RUN blinks every second)	Stopped

The "CAN-ERR" LED indicates a CANopen error:

LED display	CANopen error
 (CAN-ERR blinks once, then goes off for 1 second)	Warning Limit reached
 (CAN-ERR blinks twice, then goes off for 1 second)	Node Guard Event
 (CAN-ERR blinks three times, then goes off for 1 second)	Sync message error (only possible in the "Operational" state)

4.14 Resetting the CANopen interface

The following parameter can be used to restart or stop CAN communication. Optionally it is also possible to reset all CAN parameters to the default state.

Parameter	Name / value range / [default setting]		Info
0x2300 *(PAR 508)*	Activate network *(Activ. network)* • Setting can only be changed if controller inhibit is active.		Restart / stop CAN communication. • After successful execution, the value 0 is shown.
	0	**No action/no error**	Only status feedback.
	1	Restart with current values	Restart CAN communication with the current values.
	2	Restart with standard values	Restart CAN communication with the standard values of the CAN parameters (0x1000 ... 0x1FFF and 0x2301). The standard values of these parameters are saved in the memory module.
	5	Stop network communication	Stop CAN communication. • The "Stop Remote Node" NMT command is executed. After successful execution of this command, only the reception of network management frames is possible.
	10	In progress	Only status feedback.
	11	Action cancelled	
	12	Error	

5 Modbus

Modbus is an internationally approved, asynchronous, serial communication protocol, designed for commercial and industrial automation applications.

Preconditions

Control unit (CU) of the inverter is provided with Modbus.

5.1 Modbus introduction

- The process of data transmission distinguishes between three different operating modes: Modbus ASCII, Modbus RTU, and Modbus TCP. The inverter supports the Modbus RTU operating mode ("Remote Terminal Unit").
- The Modbus protocol is based on a master/slave architecture where the inverter always works as slave.
- The Modbus network only permits one master (at a time) sending commands and requests. The master is also the sole instance to be allowed to initiate Modbus communication. No direct communication takes place between the slaves.
- The physical interface corresponds to TIA/EIA-485-A which is very common and suitable for the industrial environment. This interface enables baud rates from 2400 to 115200 kbps.
- The inverter supports Modbus function codes 3, 6, 16 (0x10) and 23 (0x17).

5.2 Modbus node address

Each network node must be provided with a unique node address.

Details

- The node address of the inverter can be optionally set in 0x2321:001 *(PAR 510/001)* or using the DIP switches on the device labelled with "1" ... "128".
- The setting that is active when the inverter is switched on is the effective setting.
- The labelling of the DIP switches corresponds to the values of the individual DIP switches for determining the node address (see the following example).
- The node address 0 is reserved for messages to all nodes ("Broadcast") .
- The active node address is shown in 0x2322:001 *(PAR 511/001)*.

Example of how the node address is set via the DIP switches

DIP switch	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Setting	OFF	OFF	OFF	**ON**	OFF	**ON**	**ON**	**ON**
Value	0	0	0	**16**	0	**4**	**2**	**1**
Node address	= sum of all values = 16 + 4 + 2 + 1 = **23**							

Parameter	**Name / value range / [default setting]**	**Info**
0x2321:001 *(PAR 510/001)*	Modbus settings: Node ID *(Modbus sett.: Node ID)* 1 ... [**1**] ... 247	Optionally setting of the node address (instead of setting via DIP switches 1 ... 128). • The node address set here only becomes effective if DIP switches 1 ... 128 have been set to OFF before mains switching. • A change in the node address only becomes effective after a restart of Modbus communication.
0x2323 *(PAR 509)*	DIP switch position *(DIP switch)* • Read only	Display of the DIP switch setting at the last mains power-on. • The value displayed corresponds to the sum of the individual DIP switch values 1 ... 64.

5.3 Modbus baud rate

All network nodes must be set to the same baud rate.

Details

- If the DIP switch labelled with "b" is in the OFF position at switch-on, the automatic baud rate detection function is active. If it is in the ON position, the setting in 0x2321:002 *(PAR 510/002)* applies instead.
- If the automatic baud rate detection function is activated, the first 5 ... 10 messages are lost after switch-on.
- The active baud rate is shown in 0x2322:002 *(PAR 511/002)*.

Parameter	Name / value range / [default setting]		Info
0x2321:002 *(PAR 510/002)*	Modbus settings: Baud rate *(Modbus sett.: Baud rate)*		Optionally setting of the baud rate (instead of setting via DIP switch b). • The baud rate set here is only effective if DIP switch b was set to ON before mains switching. Otherwise automatic baud rate detection is active. • A change in the baud rate only becomes effective after a restart of Modbus communication. • If the automatic baud rate detection function is activated, the first 5 ... 10 messages are lost after switch-on.
	0	**Automatic**	
	1	2400 kbps	
	2	4800 kbps	
	3	9600 kbps	
	4	19200 kbps	
	5	38400 kbps	
	6	57600 kbps	
	7	115200 kbps	
0x2323 *(PAR 509)*	DIP switch position *(DIP switch)* • Read only		Display of the DIP switch setting at the last mains power-on. • The value displayed corresponds to the sum of the individual DIP switch values 1 ... 64.

5.4 Modbus data format

Parameter	Name / value range / [default setting]		Info
0x2321:003 *(PAR 510/003)*	Modbus settings: Data format *(Modbus sett.: Data format)*		Definition of the parity and stop bits.
	0	**Automatic**	Automatic data format detection. • With this setting, the first 5 ... 10 messages are lost after switch-on.
	1	8, E, 1	8 data bits, even parity, 1 stop bit
	2	8, O, 1	8 data bits, odd parity, 1 stop bit
	3	8, N, 2	8 data bits, no parity bit, 2 stop bits
0x2323 *(PAR 509)*	DIP switch position *(DIP switch)* • Read only		Display of the DIP switch setting at the last mains power-on. • The value displayed corresponds to the sum of the individual DIP switch values 1 ... 64.

5.5 Modbus time-out monitoring

Parameter	Name / value range / [default setting]		Info
0x2858:001 *(PAR 515/001)*	Modbus monitoring: Response to time-out *(Modbus monit.: Time-out action)* • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.		Selection of the response executed if no valid messages have been received via the Modbus for a longer time than the time-out period set in 0x2858:2.
	3	**Error**	
0x2858:002 *(PAR 515/002)*	Modbus monitoring: Time-out *(Modbus monit.: Timeout)* 0.0 ... [**2.0**] ... 300.0 s		Time-out period for monitoring the message reception via Modbus.

5.6 Modbus diagnostics

The following parameters serve to diagnose the communication activities between the inverter and the Modbus network.

Parameter	Name / value range / [default setting]	Info
0x2322:001 *(PAR 511/001)*	Active Modbus settings: Node ID *(Active sett.: Node ID)* • Read only	Display of the active node address.

Parameter	Name / value range / [default setting]	Info
0x2322:002 *(PAR 511/002)*	Active Modbus settings: Baud rate *(Active sett.: Baud rate)* • Read only	Display of the active baud rate.
0x2322:003 *(PAR 511/003)*	Active Modbus settings: Data format *(Active sett.: Data format)* • Read only	Display of the active data format.
0x232A:001 *(PAR 580/001)*	Modbus statistics: Messages received *(Modbus counter: Messages received)* • Read only	Display of the total number of messages received. • This counter counts both valid and invalid messages. • After the maximum value has been reached, the counter starts again "0".
0x232A:002 *(PAR 580/002)*	Modbus statistics: Valid messages received *(Modbus counter: Val. mess. rec.)* • Read only	Display of the number of valid messages received. • After the maximum value has been reached, the counter starts again "0".
0x232A:003 *(PAR 580/003)*	Modbus statistics: Messages with exceptions *(Modbus counter: Mess. w. exc.)* • Read only	Display of the number of messages with exceptions that have been received. • After the maximum value has been reached, the counter starts again "0".
0x232A:004 *(PAR 580/004)*	Modbus statistics: Messages with errors *(Modbus counter: Mess. with errors)* • Read only	Display of the number of messages received with a faulty data integrity (parity, CRC). • After the maximum value has been reached, the counter starts again "0".
0x232A:005 *(PAR 580/005)*	Modbus statistics: Messages sent *(Modbus counter: Messages sent)* • Read only	Display of the total number of messages sent. • After the maximum value has been reached, the counter starts again "0".
0x232E:001 *(PAR 583/001)*	Modbus diagnostics of last RX data: Offset *(RX data diagnostics: RX data offset)* 0 ... [**0**] ... 240	For purposes of diagnostics, the last message received (max. 16 bytes) is shown in 0x232E:1 ... 16. For longer messages, an offset can be specified here, indicating from which byte of the message the display of the 16 bytes is to start.
0x232E:002 *(PAR 583/002)*	Modbus diagnostics of last RX data: Data byte 0 *(RX data diagnostics: Last RxD byte0)* • Read only	Display of the message received last.
0x232E:003 *(PAR 583/003)*	Modbus diagnostics of last RX data: Data byte 1 *(RX data diagnostics: Last RxD byte2)* • Read only	
0x232E:004 *(PAR 583/004)*	Modbus diagnostics of last RX data: Data byte 2 *(RX data diagnostics: Last RxD byte4)* • Read only	
0x232E:005 *(PAR 583/005)*	Modbus diagnostics of last RX data: Data byte 3 *(RX data diagnostics: Last RxD byte6)* • Read only	
0x232E:006 *(PAR 583/006)*	Modbus diagnostics of last RX data: Data byte 4 *(RX data diagnostics: Last RxD byte8)* • Read only	
0x232E:007 *(PAR 583/007)*	Modbus diagnostics of last RX data: Data byte 5 *(RX data diagnostics: Last RxD byte10)* • Read only	
0x232E:008 *(PAR 583/008)*	Modbus diagnostics of last RX data: Data byte 6 *(RX data diagnostics: Last RxD byte12)* • Read only	
0x232E:009 *(PAR 583/009)*	Modbus diagnostics of last RX data: Data byte 7 *(RX data diagnostics: Last RxD byte14)* • Read only	
0x232E:010 *(PAR 583/010)*	Modbus diagnostics of last RX data: Data byte 8 *(RX data diagnostics: Last RxD byte16)* • Read only	
0x232E:011 *(PAR 583/011)*	Modbus diagnostics of last RX data: Data byte 9 *(RX data diagnostics: Last RxD byte18)* • Read only	
0x232E:012 *(PAR 583/012)*	Modbus diagnostics of last RX data: Data byte 10 *(RX data diagnostics: Last RxD byte20)* • Read only	

Parameter	Name / value range / [default setting]	Info
0x232E:013 *(PAR 583/013)*	Modbus diagnostics of last RX data: Data byte 11 *(RX data diagnostics: Last RxD byte22)* • Read only	
0x232E:014 *(PAR 583/014)*	Modbus diagnostics of last RX data: Data byte 12 *(RX data diagnostics: Last RxD byte24)* • Read only	
0x232E:015 *(PAR 583/015)*	Modbus diagnostics of last RX data: Data byte 13 *(RX data diagnostics: Last RxD byte26)* • Read only	
0x232E:016 *(PAR 583/016)*	Modbus diagnostics of last RX data: Data byte 14 *(RX data diagnostics: Last RxD byte28)* • Read only	
0x232E:017 *(PAR 583/017)*	Modbus diagnostics of last RX data: Data byte 15 *(RX data diagnostics: Last RxD byte30)* • Read only	
0x232F:001 *(PAR 585/001)*	Modbus diagnostics of last TX data: Offset *(TX data diagnostics: TX data offset)* 0 ... [**0**] ... 240	For purposes of diagnostics, the last message sent (max. 16 bytes) is shown in 0x232F:1 ... 16. For longer messages, an offset can be specified here, indicating from which byte of the message the display of the 16 bytes is to start.
0x232F:002 *(PAR 585/002)*	Modbus diagnostics of last TX data: Data byte 0 *(TX data diagnostics: Last TxD byte0)* • Read only	Display of the message sent last.
0x232F:003 *(PAR 585/003)*	Modbus diagnostics of last TX data: Data byte 1 *(TX data diagnostics: Last TxD Byte1)* • Read only	
0x232F:004 *(PAR 585/004)*	Modbus diagnostics of last TX data: Data byte 2 *(TX data diagnostics: Last TxD byte2)* • Read only	
0x232F:005 *(PAR 585/005)*	Modbus diagnostics of last TX data: Data byte 3 *(TX data diagnostics: Last TxD byte3)* • Read only	
0x232F:006 *(PAR 585/006)*	Modbus diagnostics of last TX data: Data byte 4 *(TX data diagnostics: Last TxD byte4)* • Read only	
0x232F:007 *(PAR 585/007)*	Modbus diagnostics of last TX data: Data byte 5 *(TX data diagnostics: Last TxD byte5)* • Read only	
0x232F:008 *(PAR 585/008)*	Modbus diagnostics of last TX data: Data byte 6 *(TX data diagnostics: Last TxD byte6)* • Read only	
0x232F:009 *(PAR 585/009)*	Modbus diagnostics of last TX data: Data byte 7 *(TX data diagnostics: Last TxD byte7)* • Read only	
0x232F:010 *(PAR 585/010)*	Modbus diagnostics of last TX data: Data byte 8 *(TX data diagnostics: Last TxD byte8)* • Read only	
0x232F:011 *(PAR 585/011)*	Modbus diagnostics of last TX data: Data byte 9 *(TX data diagnostics: Last TxD byte9)* • Read only	
0x232F:012 *(PAR 585/012)*	Modbus diagnostics of last TX data: Data byte 10 *(TX data diagnostics: Last TxD byte10)* • Read only	
0x232F:013 *(PAR 585/013)*	Modbus diagnostics of last TX data: Data byte 11 *(TX data diagnostics: Last TxD byte11)* • Read only	
0x232F:014 *(PAR 585/014)*	Modbus diagnostics of last TX data: Data byte 12 *(TX data diagnostics: Last TxD byte12)* • Read only	
0x232F:015 *(PAR 585/015)*	Modbus diagnostics of last TX data: Data byte 13 *(TX data diagnostics: Last TxD byte13)* • Read only	

Parameter	Name / value range / [default setting]	Info
0x232F:016 *(PAR 585/016)*	Modbus diagnostics of last TX data: Data byte 14 *(TX data diagnostics: Last TxD byte14)* • Read only	
0x232F:017 *(PAR 585/017)*	Modbus diagnostics of last TX data: Data byte 15 *(TX data diagnostics: Last TxD byte15)* • Read only	

5.7 Modbus function codes

The mode of access to inverter data (parameters) is controlled via function codes.

Details

The inverter supports the following function codes:

Function code		Function name	Description
3	0x03	Read Holding Registers	Read one or more 16-bit data words.
6	0x06	Preset Single Register	Write a 16-bit data word.
16	0x10	Preset Multiple Registers	Write one or more 16-bit data words.
23	0x17	Read/Write 4X Registers	Write one or more 16-bit data words and read them back.

Addressing

- The function codes listed above exclusively refer to 4X registers in Modbus addressing.
- All data in drives can only be accessed via 4X registers, i.e. via register addresses starting from 40001.
- The 4xxxx reference is implicit, i. e. given by the function code used. In the frame therefore the leading 4 is omitted in the addressing process.
- The numbering of the registers starts with 1; addressing, however, starts with 0. Therefore, for instance, the address 0 is used in the frame when register 40001 is read.

Telegram structure

Communication is established on the basis of the central medium access method. Communication is always started by a master request. The inverter (slave) then either gives a valid response or outputs an error code (provided that the request has been received and evaluated as a valid Modbus frame). Error causes can be invalid CRC checksums, function codes that are not supported, or impermissible data access.

All Modbus frames have the following basic structure:

- A "frame" consists of a PDU (Protocol Data Unit) and an ADU (Application Data Unit).
- The PDU contains the function code and the data belonging to the function code.
- The ADU serves the purposes of addressing and error detection.
- The data are represented in Big Endian format (most significant byte first).

<table>
<tr><th colspan="4">ADU (Application Data Unit)</th></tr>
<tr><td rowspan="2">Slave address</td><td>Function code</td><td>Data</td><td rowspan="2">Checksum (CRC)</td></tr>
<tr><td colspan="2"><b>PDU (Protocol Data Unit)</b></td></tr>
</table>

Error codes

Error code	Name	Cause(s)
0x01	Invalid function code	The function code is not supported by the inverter, or the inverter is in a state in which the request is not permissible or in which it cannot be processed.
0x02	Invalid data address	The combination of a start address and the length of the data to be transmitted is invalid. Example: If you have a slave with 100 registers, the first register has the address 0 and the last register has the address 99. If there is a request of four registers now, from the start address 96, the request can be processed successfully (for registers 96, 97, 98, and 99). If, however, five registers from the start address 96 are queried, this error code is returned, since the slave has no register with the address 100.
0x03	Invalid data value	The cause, however, is not that a (parameter) value is written outside the valid setting range. As a matter of principle, the Modbus protocol has no information on valid setting ranges of individual registers or their meaning.
0x04	Slave device failure	A non-correctable error has occurred while the request was processed in the inverter.

5.8 Modbus data mapping

The process of data mapping is used for defining which Modbus registers read or write to which inverter parameters.

Details

- There are fixedly defined Modbus registers for common control and status words, which are located in coherent blocks, in order to facilitate communication with OPC servers and other Modbus masters. In order to access all relevant data of the inverter, only a minimum number of commands is required.
- In addition, 24 registers are provided for variable mapping, i. e. free assignment to inverter parameters.

Predefined Modbus control registers

- These registers are provided with write and read access.
- The cross-reference in column 2 leads to the detailed parameter description.

Modbus registers	Permanently assigned parameter	
	Address	**Name**
42101	0x400B:001 *(PAR 592/001)*	AC Drive control word
42102	0x400B:005 *(PAR 592/005)*	Network setpoint frequency [0.01 Hz]
42103	0x4008:003 *(PAR 590/003)*	NETWordIN2
42104	0x400B:001 *(PAR 592/001)*	NETWordIN3
42105	0x400B:007 *(PAR 592/007)*	Network process controller setpoint
42106	0x6071	Target torque
42107	0x4008:001 *(PAR 590/001)*	NETWordIN1
42108	0x4008:004 *(PAR 590/004)*	NETWordIN4
42109 ... 42121	-	Reserved

Predefined Modbus status registers

- These registers are only provided with read access.
- The cross-reference in column 2 leads to the detailed parameter description.

Modbus registers	Permanently assigned parameter	
	Address	**Name**
42001	0x400C:001 *(PAR 593/001)*	AC Drive status word
42002	0x400C:006 *(PAR 593/006)*	Actual frequency [0.01 Hz]
42003	0x603F *(PAR 150)*	Error code
42004	0x400C:005 *(PAR 593/005)*	Drive status
42005	0x2D89 *(PAR 106)*	Actual motor voltage
42006	0x2D88 *(PAR 104)*	Actual motor current
42007	0x6078 *(PAR 103)*	Current actual value
42008	0x2DA2:002 *(PAR 108/002)*	Apparent power (42008 = High Word, 42009 = Low Word)
42009		
42010	0x2D84:001 *(PAR 117/001)*	Actual value
42011	0x2D87 *(PAR 105)*	DC-bus voltage
42012	0x60FD *(PAR 118)*	Digital inputs
42013	0x6077 *(PAR 107)*	Torque actual value
42014 ... 42021	-	Reserved

Variable mapping

- Via 0x232B:1 ... 24, 24 registers can be mapped variably to parameters of the inverter.
  Format: 0xiiiiss00 (iiii = hexadecimal index, ss = hexadecimal subindex)
- The display of the internal Modbus register numbers in 0x232C:1 ... 24 is generated automatically. Since 32-bit parameters require two registers, there is no 1:1 assignment.
- For the mappable registers, a CRC (Cyclic Redundancy Check) is executed. The checksum determined is shown in 0x232D *(PAR 532)*. The user can read this "validation code" and use it for comparison in the Modbus master. In this way it can be checked whether the inverter currently queried is configured correctly for the respective application.

Parameter	Name / value range / [default setting]	Info
0x232B:001 *(PAR 530/001)*	Modbus parameter mapping: Parameter 1 *(Para. mapping: Parameter 1)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40103.
0x232B:002 *(PAR 530/002)*	Modbus parameter mapping: Parameter 2 *(Para. mapping: Parameter 2)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40105.
0x232B:003 *(PAR 530/003)*	Modbus parameter mapping: Parameter 3 *(Para. mapping: Parameter 3)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40107.
0x232B:004 *(PAR 530/004)*	Modbus parameter mapping: Parameter 4 *(Para. mapping: Parameter 4)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40109.
0x232B:005 *(PAR 530/005)*	Modbus parameter mapping: Parameter 5 *(Para. mapping: Parameter 5)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40111.
0x232B:006 *(PAR 530/006)*	Modbus parameter mapping: Parameter 6 *(Para. mapping: Parameter 6)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40113.
0x232B:007 *(PAR 530/007)*	Modbus parameter mapping: Parameter 7 *(Para. mapping: Parameter 7)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40115.
0x232B:008 *(PAR 530/008)*	Modbus parameter mapping: Parameter 8 *(Para. mapping: Parameter 8)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40117.
0x232B:009 *(PAR 530/009)*	Modbus parameter mapping: Parameter 9 *(Para. mapping: Parameter 9)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40119.
0x232B:010 *(PAR 530/010)*	Modbus parameter mapping: Parameter 10 *(Para. mapping: Parameter 10)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40121.
0x232B:011 *(PAR 530/011)*	Modbus parameter mapping: Parameter 11 *(Para. mapping: Parameter 11)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40123.
0x232B:012 *(PAR 530/012)*	Modbus parameter mapping: Parameter 12 *(Para. mapping: Parameter 12)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40125.
0x232B:013 *(PAR 530/013)*	Modbus parameter mapping: Parameter 13 *(Para. mapping: Parameter 13)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40127.
0x232B:014 *(PAR 530/014)*	Modbus parameter mapping: Parameter 14 *(Para. mapping: Parameter 14)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40129.
0x232B:015 *(PAR 530/015)*	Modbus parameter mapping: Parameter 15 *(Para. mapping: Parameter 15)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40131.
0x232B:016 *(PAR 530/016)*	Modbus parameter mapping: Parameter 16 *(Para. mapping: Parameter 16)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40133.
0x232B:017 *(PAR 530/017)*	Modbus parameter mapping: Parameter 17 *(Para. mapping: Parameter 17)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40135.
0x232B:018 *(PAR 530/018)*	Modbus parameter mapping: Parameter 18 *(Para. mapping: Parameter 18)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40137.
0x232B:019 *(PAR 530/019)*	Modbus parameter mapping: Parameter 19 *(Para. mapping: Parameter 19)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40139.
0x232B:020 *(PAR 530/020)*	Modbus parameter mapping: Parameter 20 *(Para. mapping: Parameter 20)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40141.
0x232B:021 *(PAR 530/021)*	Modbus parameter mapping: Parameter 21 *(Para. mapping: Parameter 21)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40143.

Parameter	Name / value range / [default setting]	Info
0x232B:022 *(PAR 530/022)*	Modbus parameter mapping: Parameter 22 *(Para. mapping: Parameter 22)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40145.
0x232B:023 *(PAR 530/023)*	Modbus parameter mapping: Parameter 23 *(Para. mapping: Parameter 23)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40147.
0x232B:024 *(PAR 530/024)*	Modbus parameter mapping: Parameter 24 *(Para. mapping: Parameter 24)* 0x00000000 ... [**0x00000000**] ... 0xFFFFFF00	Mapping entry for Modbus register 40149.
0x232C:001 *(PAR 531/001)*	Modbus register assignment: Register 1 *(Reg. assigned: Register 1)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:1 is stored. • For the first parameter mapped, always 2500.
0x232C:002 *(PAR 531/002)*	Modbus register assignment: Register 2 *(Reg. assigned: Register 2)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:2 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:003 *(PAR 531/003)*	Modbus register assignment: Register 3 *(Reg. assigned: Register 3)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:3 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:004 *(PAR 531/004)*	Modbus register assignment: Register 4 *(Reg. assigned: Register 4)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:4 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:005 *(PAR 531/005)*	Modbus register assignment: Register 5 *(Reg. assigned: Register 5)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:5 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:006 *(PAR 531/006)*	Modbus register assignment: Register 6 *(Reg. assigned: Register 6)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:6 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:007 *(PAR 531/007)*	Modbus register assignment: Register 7 *(Reg. assigned: Register 7)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:7 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:008 *(PAR 531/008)*	Modbus register assignment: Register 8 *(Reg. assigned: Register 8)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:8 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:009 *(PAR 531/009)*	Modbus register assignment: Register 9 *(Reg. assigned: Register 9)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:9 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:010 *(PAR 531/010)*	Modbus register assignment: Register 10 *(Reg. assigned: Register 10)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:10 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:011 *(PAR 531/011)*	Modbus register assignment: Register 11 *(Reg. assigned: Register 11)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:11 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:012 *(PAR 531/012)*	Modbus register assignment: Register 12 *(Reg. assigned: Register 12)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:12 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:013 *(PAR 531/013)*	Modbus register assignment: Register 13 *(Reg. assigned: Register 13)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:13 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.

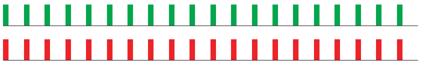
Parameter	Name / value range / [default setting]	Info
0x232C:014 *(PAR 531/014)*	Modbus register assignment: Register 14 *(Reg. assigned: Register 14)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:14 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:015 *(PAR 531/015)*	Modbus register assignment: Register 15 *(Reg. assigned: Register 15)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:15 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:016 *(PAR 531/016)*	Modbus register assignment: Register 16 *(Reg. assigned: Register 16)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:16 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:017 *(PAR 531/017)*	Modbus register assignment: Register 17 *(Reg. assigned: Register 17)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:17 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:018 *(PAR 531/018)*	Modbus register assignment: Register 18 *(Reg. assigned: Register 18)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:18 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:019 *(PAR 531/019)*	Modbus register assignment: Register 19 *(Reg. assigned: Register 19)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:19 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:020 *(PAR 531/020)*	Modbus register assignment: Register 20 *(Reg. assigned: Register 20)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:20 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:021 *(PAR 531/021)*	Modbus register assignment: Register 21 *(Reg. assigned: Register 21)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:21 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:022 *(PAR 531/022)*	Modbus register assignment: Register 22 *(Reg. assigned: Register 22)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:22 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:023 *(PAR 531/023)*	Modbus register assignment: Register 23 *(Reg. assigned: Register 23)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:23 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232C:024 *(PAR 531/024)*	Modbus register assignment: Register 24 *(Reg. assigned: Register 24)* • Read only	Display of the internal Modbus register number starting from which the parameter mapped in 0x232B:24 is stored. • 2500 + offset. The offset results from the data types of the previously mapped parameters.
0x232D *(PAR 532)*	Modbus verification code *(Verification code)* • Read only	
0x6071	Target torque -3276.8 ... [**0.0**] ... 3276.7 %	Setpoint torque • 100 % ≡ Motor rated torque 0x6076 *(PAR 325)*

5.9 Modbus LED status displays

Information about the Modbus status can be obtained quickly via the "MOD-RUN" and "MOD-ERR" LED displays on the front of the inverter.

The meaning can be seen from the tables below.

Inverter not active on the Modbus bus (yet)

LED display	Meaning
 (MOD-ERR is lit permanently)	Internal error
 (MOD-RUN and MOD-ERR are flickering)	Automatic baud rate detection active.

Inverter active on the Modbus

The green "MOD-RUN" LED indicates the communication status:

LED display	Communication status
Off	No reception / no transmission
On	Reception / transmission active

The red "MOD-ERR" LED indicates an error:

LED display	Fault
Off	No fault
Blinking	Communication error

5.10 Reset Modbus interface

Parameter	Name / value range / [default setting]	Info
0x2320 *(PAR 508)*	Activate Modbus network *(Activ. netw. sett.)* 0 ... [**0**] ... 1	1 = activate network options.

5.11 Modbus response time

Parameter	Name / value range / [default setting]	Info
0x2321:004 *(PAR 510/004)*	Modbus settings: Minimum response time *(Modbus sett.: Min. resp. time)* 0 ... [**0**] ... 1000 ms	Minimum time delay between the reception of a valid message and the response of the drive.

5.12 Short setup of Modbus

In the following, the steps required for controlling the inverter via Modbus are described.

Parameterisation required

1. Activate network: 0x2631:037 *(PAR 400/037)* = "1: TRUE"
2. Set network as default setpoint source: 0x2860:001 *(PAR 201/001)* = "5: Network"
3. Set Modbus node address. ▶ Modbus node address
4. Set Modbus baud rate. ▶ Modbus baud rate
   - Default setting: automatic detection.
   - If the automatic baud rate detection function is activated, the first 5 ... 10 messages are lost after switch-on.
5. Set Modbus data format. ▶ Modbus data format
   - Default setting: automatic detection.
   - If the automatic data format detection function is activated, the first 5 ... 10 messages are lost after switch-on.

Digital input DI1 is assigned with the "Start enable" function by default and therefore must be set to HIGH level.

Starting/stopping the drive via Modbus

For starting/stopping the drive, Modbus register 42101 can be used.

- Modbus register 42101 is permanently assigned to the AC Drive control word 0x400B:001 *(PAR 592/001)*.
- In the frame, the leading 4 is omitted in the addressing process. The numbering of the registers starts with 1; addressing, however starts with 0. Therefore the address 2100 (0x0834) is used in the frame when register 42101 is written.

Bits set in the AC Drive control word:
- Bit 0 ≡ forward operation
- Bit 5 ≡ network control
- Bit 6 ≡ network setpoint set

Example of an inverter with the node address 1:

Request frame by the master							
Slave address	Function code	Data				Checksum (CRC)	
		Register address		AC Drive control word			
0x01	0x06	0x08	0x34	0x00	0x61	0x0B	0x8C

If digital input DI1 ("Start enable") is set to HIGH level, the drive should start and the inverter should respond with the same frame as confirmation:

Response message from the inverter							
Slave address	Function code	Data				Checksum (CRC)	
		Register address		AC Drive control word			
0x01	0x06	0x08	0x34	0x00	0x61	0x0B	0x8C

6 PROFIBUS

PROFIBUS is a common fieldbus for the connection of inverters to different control systems in plants.

Preconditions

- Control unit (CU) of the inverter is provided with PROFIBUS.
- For the configuration of PROFIBUS, the PROFIBUS device description file of the inverter must be imported into the master.

6.1 PROFIBUS introduction

The inverter is integrated into a PROFIBUS-DP network as slave. Therefore it is only allowed to receive and acknowledge messages and to respond to requests by a master. The master is also referred to as an active node. Two different types are distinguished:

- Class 1 DP master: central control (PLC or PC) which cyclically exchanges process data with the slave. Acyclic data exchange via a separate transmission channel is also possible.
- Class 2 DP master: engineering, configuration, or operator device (HMI) which only exchanges data with the slave acyclically, e.g. for the purposes of configuration, maintenance, or diagnostics.

6.2 PROFIBUS communication time

The communication time is the time between the start of a request and the arrival of the corresponding response.

The communication times in the PROFIBUS network depend on the ...

- processing time in the inverter
- transmission delay time (baud rate / frame length)
- nesting depth of the network.

In the case of the inverter, the processing time for process data is approx. 2 ... 3 ms, and for parameter data (DPV1) it is approx. 10 ms. There are no interdependencies between parameter data and process data.

6.3 PROFIBUS node address

Each network node must be provided with a unique station address.

Details

- The station address of the inverter can be optionally set via the DIP switches on the device labelled with "1" ... "64" or in 0x2341:001 *(PAR 510/001)*. (The DIP switches have priority.)
- The setting that is active when the inverter is switched on is the effective setting.
- The labelling of the DIP switches corresponds to the values of the individual DIP switches for determining the station address (see the following example).
- The active station address is shown in 0x2342:001 *(PAR 511/001)*.

Example of how the station address is set via the DIP switches

DIP switch	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Setting	OFF	OFF	**ON**	OFF	**ON**	**ON**	**ON**
Value	0	0	**16**	0	**4**	**2**	**1**
Station address	= sum of all values = 16 + 4 + 2 + 1 = **23**						

Parameter	**Name / value range / [default setting]**	**Info**
0x2341:001 *(PAR 510/001)*	PROFIBUS settings: Node ID *(PROFIBUS sett.: Node ID)* 1 ... [**3**] ... 125	Optional setting of the station address (instead of setting via DIP switches 1 ... 64). • The station address set here only becomes effective if DIP switches 1 ... 64 have been set to OFF before mains switching. • A change in the station address only becomes effective after a restart of PROFIBUS communication.
0x2342:001 *(PAR 511/001)*	Active PROFIBUS settings: Node ID *(PROFIBUS active: Node ID)* • Read only	Display of the active station address.
0x2343 *(PAR 509)*	DIP switch position *(Switch position)* • Read only	Display of the DIP switch setting at the last mains power-on. • The displayed value corresponds to the sum of the individual DIP switch values 1 ... 64.

6.4 PROFIBUS baud rate

At the class 1 DP master, the desired baud rate is set. All masters at the bus must be set to the same baud rate.

Details

- The inverter detects the baud rate automatically.
- The active baud rate is shown in 0x2342:002 *(PAR 511/002)*.
- The status of automatic detection is shown in 0x2348:002 *(PAR 516/002)*.

Parameter	**Name / value range / [default setting]**		**Info**
0x2342:002 *(PAR 511/002)*	Active PROFIBUS settings: Baud rate *(PROFIBUS active: Baud rate)* • Read only		Display of the active baud rate.
	0	12 Mbps	
	1	6 Mbps	
	2	3 Mbps	
	3	1.5 Mbps	
	4	500 kbps	
	5	187.5 kbps	
	6	93.75 kbps	
	7	45.45 kbps	
	8	19.2 kbps	
	9	9.6 kbps	
	15	Search	Automatic baud rate detection active.

Parameter	Name / value range / [default setting]		Info
0x2348:002 *(PAR 516/002)*	PROFIBUS Status: Watchdog status *(PROFIBUS Status: Watchdog status)* • Read only		Display of the current state of the watchdog state machine (WD-STATE).
	0	BAUD_SEARCH	The inverter (slave) is able to detect the baud rate automatically.
	1	BAUD_CONTROL	After detecting the correct baud rate, the inverter (slave) status changes to BAUD_CONTROL, and the baud rate is monitored.
	2	DP_CONTROL	The DP_CONTROL state serves to the response monitoring of the master.

6.5 PROFIBUS monitoring

The inverter can give a parameterisable response to the following events:

- Communication to the PROFIBUS master is continuously interrupted.
- Data exchange via PROFIBUS has been terminated.
- The inverter has received invalid configuration data from the master.
- An error has occurred during the initialisation of the PROFIBUS interface.
- The process data received are invalid.

Parameter	Name / value range / [default setting]		Info
0x2342:003 *(PAR 511/003)*	Active PROFIBUS settings: Watchdog time *(PROFIBUS active: Watchdog time)* • Read only		Display of the watchdog monitoring time specified by the master. • Monitoring starts with the arrival of the first telegram. • When a value of "0" is displayed, the monitoring function is deactivated. • A change in the watchdog monitoring time in the master is effective immediately.
0x2348:002 *(PAR 516/002)*	PROFIBUS Status: Watchdog status *(PROFIBUS Status: Watchdog status)* • Read only		Display of the current state of the watchdog state machine (WD-STATE).
	0	BAUD_SEARCH	The inverter (slave) is able to detect the baud rate automatically.
	1	BAUD_CONTROL	After detecting the correct baud rate, the inverter (slave) status changes to BAUD_CONTROL, and the baud rate is monitored.
	2	DP_CONTROL	The DP_CONTROL state serves to the response monitoring of the master.
0x2349 *(PAR 517)*	PROFIBUS error *(PROFIBUS error)* • Read only		Bit-coded display of PROFIBUS errors.
	Bit 0	Watchdog elapsed	Communication with the PROFIBUS master is continuously interrupted, e. g. by cable break or failure of the PROFIBUS master. • No process data are sent to the inverter (slave) in the "Data Exchange" state. • When the watchdog monitoring time specified by the master has elapsed, the response set in 0x2859:001 *(PAR 515/001)* is triggered in the inverter. Preconditions for a response by the inverter (slave): • The slave is in the "Data Exchange" state. • The watchdog monitoring time is configured correctly in the master (1 ... 65535 ms). If one of these preconditions is not met, the response to the absence of cyclic process data telegrams from the master is not executed.
	Bit 1	Data exchange completed	Data exchange via PROFIBUS has been terminated. • The inverter (slave) can be instructed by the master to exit the "Data Exchange" state. • If this state change is to be treated as an error in the inverter, the desired response can be set in 0x2859:002 *(PAR 515/002)*.
	Bit 2	Incorrect configuration data	The inverter (slave) has received invalid configuration data from the master. • The response set in 0x2859:003 *(PAR 515/003)* is effected.
	Bit 3	Initialisation error	An error has occurred during the initialisation of the PROFIBUS interface. • The response set in 0x2859:004 *(PAR 515/004)* is effected.
	Bit 4	Invalid process data	The inverter (slave) has received invalid process data from the master, e.g. no process data or deleted process data are sent by the "Stop" operating status in the master. • The response set in 0x2859:005 *(PAR 515/005)* is effected.

Parameter	Name / value range / [default setting]	Info
0x2859:001 *(PAR 515/001)*	PROFIBUS monitoring: Watchdog elapsed *(PB monitoring: WD elapsed)* • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response to the continuous interruption of communication to the PROFIBUS master, e. g. by cable break or failure of the PROFIBUS master.
	3 Error	
0x2859:002 *(PAR 515/002)*	PROFIBUS monitoring: Data exchange completed *(PB monitoring: Data exch. exited)* • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response to exiting the "Data Exchange" state.
	0 No response	
0x2859:003 *(PAR 515/003)*	PROFIBUS monitoring: Invalid configuration *(PB monitoring: Invalid config)* • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response triggered by the reception of invalid configuration data.
	3 Error	
0x2859:004 *(PAR 515/004)*	PROFIBUS monitoring: Initialisation error *(PB monitoring: Stack init. faulty)* • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response triggered by the occurrence of an error during the initialisation of the PROFIBUS module.
	3 Error	
0x2859:005 *(PAR 515/005)*	PROFIBUS monitoring: Invalid process data *(PB monitoring: Invalid proc. data)* • For possible settings see description for 0x2D45:001 *(PAR 310/001)*.	Selection of the response triggered by the reception of invalid process data. • If the master changes to the "Stop" state, no cyclic process data are sent to the inverter (slave) anymore; the length of the process data then is 0.
	2 Trouble	

6.6 PROFIBUS LED status displays

Information about the PROFIBUS status can be obtained quickly via the "NS" and "NE" LED displays on the front of the inverter.

The meaning can be seen from the table below.

LED "NS" (green)	LED "NE" (red)	Status/meaning
off	off	No supply voltage available, network deactivated, not initialised, or firmware download active.
on		Connected with master, control running, "Data Exchange" state active.
Blinking		Not connected, control stopped, or no data exchange.
Blinking	Blinking	Watchdog monitoring time elapsed.
off	on	Invalid station address set or non-correctable error.
Any	Flashing	PROFIBUS parameterisation error.
	Flashing 2 x	PROFIBUS configuration error.

6.7 PROFIBUS diagnostics

The following parameters serve to diagnose the communication activities between the inverter and the PROFIBUS network.

Parameter	Name / value range / [default setting]	Info
0x2344:001 *(PAR 512/001)*	PROFIBUS Configuration: Extended diagnostic bit *(PROFIBUS Config.: Ext. diag. bit)*	1 = set external diagnostic bit ("Diag Bit"). • The diagnostic bit is sent to the master where it is evaluated separately.
	0 Delete	
	1 Set	

Parameter	Name / value range / [default setting]	Info
0x2348:001 *(PAR 516/001)*	PROFIBUS Status: Bus status *(PROFIBUS Status: Bus status)* • Read only	Display of the current DP state machine state (DP-STATE).
	0 Wait for parameter data (WAIT_PRM)	After the run-up, the inverter (slave) is waiting for parameter data (CHK_PRM) from the master. All other frame types are not processed. Exchanging user data with the master is not possible yet.
	1 Wait for configuration data (WAIT_CFG)	The inverter (slave) is waiting for configuration data (CHK_CFG) from the master that define the structure of the cyclic frames.
	2 Data exchange (DATA_EXCH)	Parameter and configuration data have been received and accepted by the inverter (slave). The inverter is in the "Data Exchange" state. It is now possible to exchange user data with the master.
0x234A:001 *(PAR 580/001)*	PROFIBUS statistics: Data cycles per second *(PROFIBUS counter: Data cycles/sec.)* • Read only	Display of the data cycles per second.
0x234A:002 *(PAR 580/002)*	PROFIBUS statistics: Program events *(PROFIBUS counter: PRM events)* • Read only	Display of the number of parameterisation events.
0x234A:003 *(PAR 580/003)*	PROFIBUS statistics: Configuration events *(PROFIBUS counter: CFG events)* • Read only	Display of the number of configuration events.
0x234A:004 *(PAR 580/004)*	PROFIBUS statistics: Diagnostics events *(PROFIBUS counter: DIAG events)* • Read only	Display of the number of diagnostic telegrams sent.
0x234A:005 *(PAR 580/005)*	PROFIBUS statistics: C1 messages *(PROFIBUS counter: C1 messages)* • Read only	Display of the number of requests by the class 1 DPV1 master.
0x234A:006 *(PAR 580/006)*	PROFIBUS statistics: C2 messages *(PROFIBUS counter: C2 messages)* • Read only	Display of the number of requests by the class 2 DPV1 master.
0x234A:007 *(PAR 580/007)*	PROFIBUS statistics: Watchdog events *(PROFIBUS counter: WD events)* • Read only	Display of the number of watchdog events.
0x234A:008 *(PAR 580/008)*	PROFIBUS statistics: Data exchange aborts *(PROFIBUS counter: DataEx.event)* • Read only	Display of the number of "Data Exchange exited" events.
0x234A:009 *(PAR 580/009)*	PROFIBUS statistics: Total data cycles *(PROFIBUS counter: Total data cycles)* • Read only	Display of the number of cyclic process data received.

6.8 PROFIBUS functions

The inverter supports PROFIBUS DP-V0 (DRIVECOM profile) and PROFIBUS DP-V1 (PROFIdrive profile). PROFIBUS DP-V2 is not supported.

Details

The PROFIBUS DP communication protocol is provided with the following functions:
- DP-V0: cyclic data exchange, diagnostics (all devices).
- DP-V1: acyclic data exchange, process alarm processing (process automation). Note: The inverter does not support any alarm diagnostics.
- DP-V2: cycle synchronisation and time stamp, slave-to-slave communication.

A class 1 DP master connection (DPV1 C1) between a cyclic master and slave is established automatically when the "Data Exchange" state has been established. In byte 7 of the parameterisation frame, the "DPV1_Enable" bit must be set. Furthermore, a class 2 DP master connection (DPV1 C2) with the slave can be defined by another master connected. This connection must be established via the "MSAC2_Initiate" service.

The inverter supports the following acyclic DPV1 services:
- MSAC1_Read/Write: C1 read/write request for a data block.
- MSAC2_Initiate/Abort: connection or disconnection for acyclic data exchange between a class 2 DP master and the slave.
- MSAC2_Read/Write: C2 read/write request for a data block.

6.9 PROFIBUS data mapping

The process of data mapping is used for defining which process data are exchanged cyclically between the master and slave. The data mapping is defined in the hardware configurator and is transferred to the inverter at each restart.

Details

- First the user imports the GSD file into the hardware configurator of the control. By adding the node to the PROFIBUS network configuration, the user can then select the data required for the application.
- After the start-up, the master communicates the structure of the cyclic frames to the inverter (slave) via the configuration frame (CHK_CFG).
- The inverter checks the configuration. If the configuration is accepted, the inverter changes from the "Wait Configuration" state to the "Data Exchange" state. It is now possible to exchange user data with the master.
- Internal mapping of the cyclic data is set in 0x24E0:xxx (master → inverter direction) and 0x24E1:xxx (inverter → master direction).

Parameter	Name / value range / [default setting]	Info
0x24E0:000	Generic RPDO mapping: Highest subindex 0 ... [**2**] ... 16	Number of mapping entries for RPDO.
0x24E0:001	Generic RPDO mapping: Entry 1 0x00000000 ... [**0x60400010**] ... 0xFFFFFFFF	Mapping entry 1 for RPDO.
0x24E0:002	Generic RPDO mapping: Entry 2 0x00000000 ... [**0x60420010**] ... 0xFFFFFFFF	Mapping entry 2 for RPDO.
0x24E0:003	Generic RPDO mapping: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for RPDO.
0x24E0:004	Generic RPDO mapping: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for RPDO.
0x24E0:005	Generic RPDO mapping: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for RPDO.
0x24E0:006	Generic RPDO mapping: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for RPDO.
0x24E0:007	Generic RPDO mapping: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for RPDO.
0x24E0:008	Generic RPDO mapping: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for RPDO.
0x24E0:009	Generic RPDO mapping: Entry 9 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 9 for RPDO.
0x24E0:010	Generic RPDO mapping: Entry 10 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 10 for RPDO.
0x24E0:011	Generic RPDO mapping: Entry 11 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 11 for RPDO.
0x24E0:012	Generic RPDO mapping: Entry 12 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 12 for RPDO.
0x24E0:013	Generic RPDO mapping: Entry 13 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 13 for RPDO.
0x24E0:014	Generic RPDO mapping: Entry 14 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 14 for RPDO.
0x24E0:015	Generic RPDO mapping: Entry 15 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 15 for RPDO.
0x24E0:016	Generic RPDO mapping: Entry 16 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 16 for RPDO.
0x24E1:000	Generic TPDO mapping: Highest subindex 0 ... [**2**] ... 16	Number of mapping entries for TPDO.
0x24E1:001	Generic TPDO mapping: Entry 1 0x00000000 ... [**0x60410010**] ... 0xFFFFFFFF	Mapping entry 1 for TPDO.
0x24E1:002	Generic TPDO mapping: Entry 2 0x00000000 ... [**0x60440010**] ... 0xFFFFFFFF	Mapping entry 2 for TPDO.
0x24E1:003	Generic TPDO mapping: Entry 3 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 3 for TPDO.

Parameter	Name / value range / [default setting]	Info
0x24E1:004	Generic TPDO mapping: Entry 4 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 4 for TPDO.
0x24E1:005	Generic TPDO mapping: Entry 5 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 5 for TPDO.
0x24E1:006	Generic TPDO mapping: Entry 6 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 6 for TPDO.
0x24E1:007	Generic TPDO mapping: Entry 7 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 7 for TPDO.
0x24E1:008	Generic TPDO mapping: Entry 8 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 8 for TPDO.
0x24E1:009	Generic TPDO mapping: Entry 9 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 9 for TPDO.
0x24E1:010	Generic TPDO mapping: Entry 10 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 10 for TPDO.
0x24E1:011	Generic TPDO mapping: Entry 11 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 11 for TPDO.
0x24E1:012	Generic TPDO mapping: Entry 12 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 12 for TPDO.
0x24E1:013	Generic TPDO mapping: Entry 13 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 13 for TPDO.
0x24E1:014	Generic TPDO mapping: Entry 14 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 14 for TPDO.
0x24E1:015	Generic TPDO mapping: Entry 15 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 15 for TPDO.
0x24E1:016	Generic TPDO mapping: Entry 16 0x00000000 ... [**0x00000000**] ... 0xFFFFFFFF	Mapping entry 16 for TPDO.

6.10 PROFIBUS - acyclic data transfer

Data communication with PROFIBUS DP-V0 is characterised by cyclic diagnostics and cyclic process data transfer. An optional service expansion is the acyclic parameter data transfer of PROFIBUS DP-V1. This service does not impair the functionality of the standard services under PROFIBUS DP-V0.

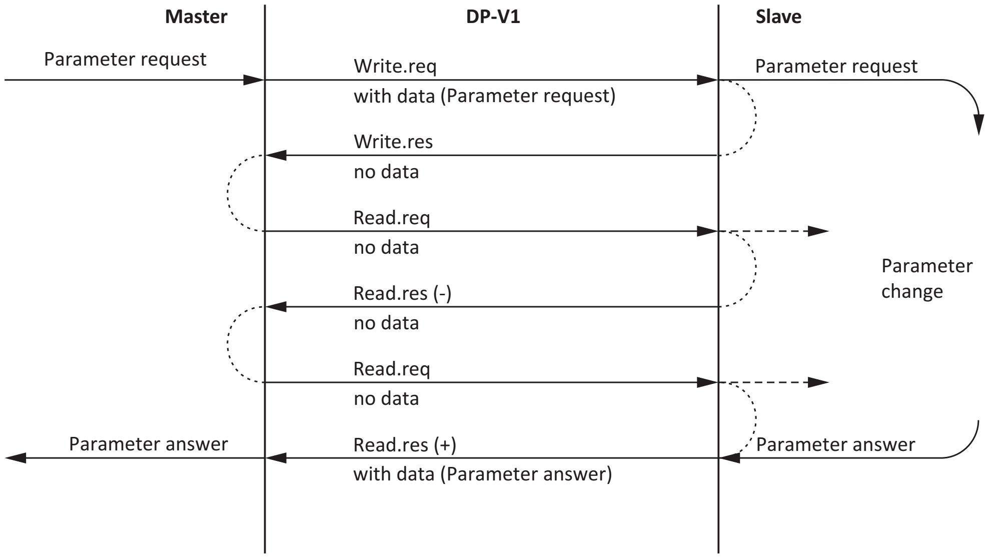
Details

- PROFIBUS DP-V0 and PROFIBUS DP-V1 can be operated simultaneously in the same network. This enables the step-by-step expansion or modification of a system.
- The services of PROFIBUS DP-V1 can be used by the class 1 master (PLC) and the class 2 DP master (diagnostics master, etc.).
- Integration of the acyclic service into the fixed bus cycle depends on the corresponding configuration of the class 1 master:
  - With configuration, a time slot is reserved.
  - Without configuration, the acyclic service is appended when a class 2 DP master acyclically accesses a DP-V1 slave.

Product features

- 16 bits each for addressing the parameter index and subindex.
- Several parameter requests can be combined to one request (multi-parameter requests).
- Only one request is processed at a time (no pipelining).
- A request or response must fit into one data block (max. 240 bytes). Requests or responses cannot be split into several data blocks.
- No spontaneous messages are transferred.
- There are only acyclic parameter requests.
- Profile-specific parameters can be read independently of the slave state.
- A class 1 DP master can always request parameters from a slave if the slave is in the "Data Exchange" state.
- In addition to a class 1 DP master, a class 2 DP master can establish communication with a slave:

Transmission directions for acyclic data transfer



Procedure:

1. A "Write.req" is used to pass the data set (DB47) to the slave in the form of a parameter request.
2. With "Write.res" the master receives the confirmation for the receipt of the message.
3. The master requests the response of the slave with "Read.req".
4. The slave responds with "Read.res (-)" if processing has not been completed yet.
5. After parameter processing, the parameter request is completed by transmitting the parameter response to the master with "Read.res (+)".

Telegram structure

SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	**Data Unit (DU)**	FCS	ED

The Data Unit (DU) contains the DP-V1 header and the parameter request or the parameter response. The DP V1 header consists of the function detection, slot number, data set, and the length of the user data. More information about the DP-V1 header can be found in the corresponding PROFIBUS specification. A detailed description of the parameter request and parameter response can be found in the following subchapters.

Assignment of the user data depending on the data type

Depending on the data type used, the user data are assigned as follows:

Data type	**Length**	**User data assignment**				
		Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Byte ...**
String	x bytes	*Data* (x bytes)				
U8	1 byte	*Data*	0x00			
U16	2 bytes	HIGH byte	LOW byte			
		Data	*Data*			
U32	4 bytes	HIGH word		LOW word		
		HIGH byte	LOW byte	HIGH byte	LOW byte	
		Data	*Data*	*Data*	*Data*	

6.11 PROFIBUS - reading parameter data acyclically

This section describes the request and response for the acyclic reading of a parameter.

Details

- When a read request is processed, no parameter value is written to the slave.
- When a read request is transmitted by multi-parameters, the parameter attribute, index and subindex are repeated.
- A read request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is defined by the master.
Request identification	U8	0x01: Request parameters for reading.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter attribute

Byte 5	Byte 6	
Attribute	Number of subindices	

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
HIGH byte	LOW byte	HIGH byte	LOW byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0000 ... 0x00FF (0 ... 255)

Response to a correctly executed read request

Responses to a read request do not contain parameter attributes, indices and subindices.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request.
Response identification	U8	0x01: Parameter has been read.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6	
Format	Number of values	

Field	Data type	Values
Format	U8	0x02: integer8 (1 byte with sign) 0x03: Integer16 (2 bytes with sign) 0x04: Integer32 (4 bytes with sign) 0x05: Unsigned8 (1 byte without sign) 0x06: Unsigned16 (2 bytes without sign) 0x07: Unsigned32 (4 bytes without sign) 0x09: Visible String (with n characters) 0x0A: Octet String (with n characters) 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of characters (n) for string parameters.

Parameter value

Byte 7	Byte 8	Byte 9	Byte 10
Value (Integer8 / Unsigned8 / byte)			
Value (Integer16 / Unsigned16 / word)			
Value (Integer32 / Unsigned32 / double word)			

Byte 7	Byte 8	Byte 9	Byte ...
String (Visible String / octet string with an optional length)			

Field	Data type	Values
Value	U8/U16/U32	Value range/length depends on the parameter format (see table above).
String	U8	Visible string / octet string with an optional length (n characters = n bytes)

Response to a read error

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one telegram. They have the following data contents:

Correct message

- Format: data type of the value requested
- Number of values: as described above.
- Parameter value: value requested

Faulty message

- Format: 0x44
- Number of values: 0x01 or 0x02
- Error code without additional information (for number of values = 0x01) or error code with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request.
Response identification	U8	0x81: Parameter has not been read. The data in bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6	
Format	Number of values	

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information. 0x02: Error code with additional information.

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
HIGH byte	LOW byte	HIGH byte	LOW byte

Field	Data type	Values
Error code	U16	0x0000 ... 0xFFFF PROFIBUS error codes for acyclic data transfer
Additional information (if available)	U16	

6.12 PROFIBUS - writing parameter data acyclically

This section describes the request and response for the acyclic writing of a parameter.

Details

- When a multi-parameter write request is transmitted, the parameter attribute, index and subindex and then the parameter format and parameter value are repeated "n" times, "n" being the number of parameters addressed.
- A write request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is defined by the master.
Request identification	U8	0x02: Write parameters.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter attribute

Byte 5	Byte 6	
Attribute	Number of subindices	

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
HIGH byte	LOW byte	HIGH byte	LOW byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0000 ... 0x00FF (0 ... 255)

Parameter format

Byte 11	Byte 12	
Format	Number of values	

Field	Data type	Values
Format	U8	0x02: integer8 (1 byte with sign) 0x03: Integer16 (2 bytes with sign) 0x04: Integer32 (4 bytes with sign) 0x05: Unsigned8 (1 byte without sign) 0x06: Unsigned16 (2 bytes without sign) 0x07: Unsigned32 (4 bytes without sign) 0x09: Visible String (with n characters) 0x0A: Octet String (with n characters) 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of characters (n) for string parameters.

Parameter value

Byte 13	Byte 14	Byte 15	Byte 16
Value (Integer8 / Unsigned8 / byte)			
Value (Integer16 / Unsigned16 / word)			
Value (Integer32 / Unsigned32 / double word)			

Byte 13	Byte 14	Byte 15	Byte ...
String (Visible string / octet string with an optional length)			

Field	Data type	Values
Value	U8/U16/U32	Value range/length depends on the parameter format (see table above).
String	U8	Visible string / octet string with an optional length (n characters = n bytes)

Response to a correctly executed write request

With an error-free multi-parameter request, only the response header is transmitted, and the complete data area is omitted.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request.
Response identification	U8	0x02: Parameter has been written.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Response to a write error

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one telegram. They have the following data contents:

Correct message
- Format: 0x40
- Number of values: 0x00

Faulty message
- Format: 0x44
- Number of values: 0x01 or 0x02
- Error code without additional information (for number of values = 0x01) or error code with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request.
Response identification	U8	0x82: Parameter has not been written. The data in bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter format

Byte 5	Byte 6	
Format	Number of values	

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information. 0x02: Error code with additional information.

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
HIGH byte	LOW byte	HIGH byte	LOW byte

Field	Data type	Values
Error code	U16	0x0000 ... 0xFFFF
Additional information (if available)	U16	PROFIBUS error codes for acyclic data transfer

6.13 PROFIBUS error codes for acyclic data transfer

The following table lists all possible error codes for the acyclic data exchange:

Error code	Description	Explanation	Additional information
0x0000	Parameter number impermissible	Access to non-available parameter.	-
0x0001	Parameter value cannot be changed	Change access to a parameter value that cannot be changed.	Subindex
0x0002	Lower or upper value limit exceeded	Change access with value beyond the value limits.	Subindex
0x0003	Subindex impermissible	Access to non-available subindex.	Subindex
0x0004	No array	Access with subindex to non-indicated parameter.	-
0x0005	Incorrect data type	Change access with value that does not match the data type of the parameter.	-
0x0006	No setting permitted (only resettable)	Change access with a non-zero value where it is not permitted.	Subindex
0x0007	Description element cannot be changed	Change access to a description element that cannot be changed.	Subindex
0x0008	Reserved	(PROFIdrive profile V2: PPO-Write requested in IR is not available.)	-
0x0009	Description data not available	Access to non-available description (parameter value is available).	-
0x000A	Reserved	(PROFIdrive profile V2: Wrong access group.)	-
0x000B	No parameter change rights	Change access with missing parameter change rights.	-
0x000C	Reserved	(PROFIdrive profile V2: Wrong password.)	-
0x000D	Reserved	(PROFIdrive profile V2: Text cannot be read in cyclic data transfer.)	-
0x000E	Reserved	(PROFIdrive profile V2: Name cannot be read in cyclic data transfer.)	-
0x000F	No text array available	Access to non-available text array (parameter value is available).	-
0x0010	Reserved	(PROFIdrive profile V2: No PPO-Write.)	-
0x0011	Request cannot be executed due to the operating state	Access is not possible for temporary reasons that are not specified in detail.	-
0x0012	Reserved	(PROFIdrive profile V2: Other error.)	-
0x0013	Reserved	(PROFIdrive profile V2: Date cannot be read in cyclic data transfer.)	-
0x0014	Value impermissible	Change access with the value that is within the value limits but that is impermissible for other permanent reasons (parameters with defined individual values).	Subindex
0x0015	Response too long	The length of the current response exceeds the maximum length transferrable.	-

Error code	Description	Explanation	Additional information
0x0016	Parameter address impermissible	Impermissible value or value which is not supported for the attribute, number of subindexes, parameter number, or subindex, or a combination.	-
0x0017	Format impermissible	Write request: Impermissible or non-supported format of parameter data.	-
0x0018	Number of values not consistent	Write request: Number of parameter data values does not match the number of subindexes in the parameter address.	-
0x0019	Axis impermissible	Access to non-available axis. For double axis, only 0x00 or 0x01 permitted.	-
0x001A	Reserved	-	-
...			
0x00FF			