37383D

easYgen-3000 Series (Package P1)
Genset Control

Interface

Software Version: 1.10xx
Part Numbers: 8440-1816 / 8440-1817 / 8440-1818 / 8440-1831

Manual 37383D

WARNING

Read this entire manual and all other publications pertaining to the work to be performed before installing, operating, or servicing this equipment. Practice all plant and safety instructions and precautions. Failure to follow instructions can cause personal injury and/or property damage.

The engine, turbine, or other type of prime mover should be equipped with an overspeed (overtemperature, or overpressure, where applicable) shutdown device(s), that operates totally independently of the prime mover control device(s) to protect against runaway or damage to the engine, turbine, or other type of prime mover with possible personal injury or loss of life should the mechanical-hydraulic governor(s) or electric control(s), the actuator(s), fuel control(s), the driving mechanism(s), the linkage(s), or the controlled device(s) fail.

Any unauthorized modifications to or use of this equipment outside its specified mechanical, electrical, or other operating limits may cause personal injury and/or property damage, including damage to the equipment. Any such unauthorized modifications: (i) constitute "misuse" and/or "negligence" within the meaning of the product warranty thereby excluding warranty coverage for any resulting damage, and (ii) invalidate product certifications or listings.

CAUTION

To prevent damage to a control system that uses an alternator or battery-charging device, make sure the charging device is turned off before disconnecting the battery from the system.

Electronic controls contain static-sensitive parts. Observe the following precautions to prevent damage to these parts.

- **Discharge body static before handling the control (with power to the control turned off, contact a grounded surface and maintain contact while handling the control).**
- **Avoid all plastic, vinyl, and Styrofoam (except antistatic versions) around printed circuit boards.**
- **Do not touch the components or conductors on a printed circuit board with your hands or with conductive devices.**

OUT-OF-DATE PUBLICATION

This publication may have been revised or updated since this copy was produced. To verify that you have the latest revision, be sure to check the Woodward website:

http://www.woodward.com/pubs/current.pdf

The revision level is shown at the bottom of the front cover after the publication number. The latest version of most publications is available at:

http://www.woodward.com/publications

If your publication is not there, please contact your customer service representative to get the latest copy.

Important definitions

WARNING

Indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION

Indicates a potentially hazardous situation that, if not avoided, could result in damage to equipment.

NOTE

Provides other helpful information that does not fall under the warning or caution categories.

Revision History

Rev.	Date	Editor	Changes
NEW	06-12-08	TP	Release
A	07-06-29	TP	Update to reflect the new functionality
B	08-02-07	TP	Minor corrections; application examples, remote control and easYgen-3100 sections extended/added
C	08-07-25	TP	Minor corrections; update to reflect the new functionality
D	09-10-23	TE	Minor corrections

Contents

Figures and Tables

Figures

Tables

Chapter 1.
General Information

Document Overview

Type		English	German
easYgen-3000 Series			
easYgen-3000 - Installation		37223	GR37223
easYgen-3000 - Configuration		37224	GR37224
easYgen-3000 - Operation		37225	GR37225
easYgen-3000 - Application		37226	-
easYgen-3000 - Interfaces	this manual ð	37383	-
easYgen-3200 - Brief Operation Information		37399	GR37399
easYgen-3100 - Brief Operation Information		37409	-

Table 1-1: Manual - overview

Intended Use The unit must only be operated in the manner described by this manual. The prerequisite for a proper and safe operation of the product is correct transportation, storage, and installation as well as careful operation and maintenance.

NOTE

This manual has been developed for a unit fitted with all available options. Inputs/outputs, functions, configuration screens, and other details described, which do not exist on your unit, may be ignored.

The present manual has been prepared to enable the installation and commissioning of the unit. Due to the large variety of parameter settings, it is not possible to cover every combination. The manual is therefore only a guide. In case of incorrect entries or a total loss of functions, the default settings may be taken from the list of parameters enclosed in the configuration manual 37224 or from ToolKit and the respective *.SID file.

Abbreviations

The following abbreviations are frequently used in this documents:

- PDO Process Data Object
- RPDO Receive PDO
- TPDO Transmit PDO
- SDO Service Data Object
- SSDO Server SDO
- MSB Most Significant Bit
- LSB Least Significant Bit

Interface Overview

Depending on the respective model and package, the easYgen-3000 provides up to 3 CAN interfaces, 3 serial interfaces and 2 Ethernet interfaces. Table 1-2 indicates the interface set up of respective model and package.

Interface(s)	CAN	Serial RS-232	Serial RS-485	Ethernet
easYgen-3100	2	1	1	0
easYgen-3200	2	1	1	0

Table 1-2: Interfaces - overview

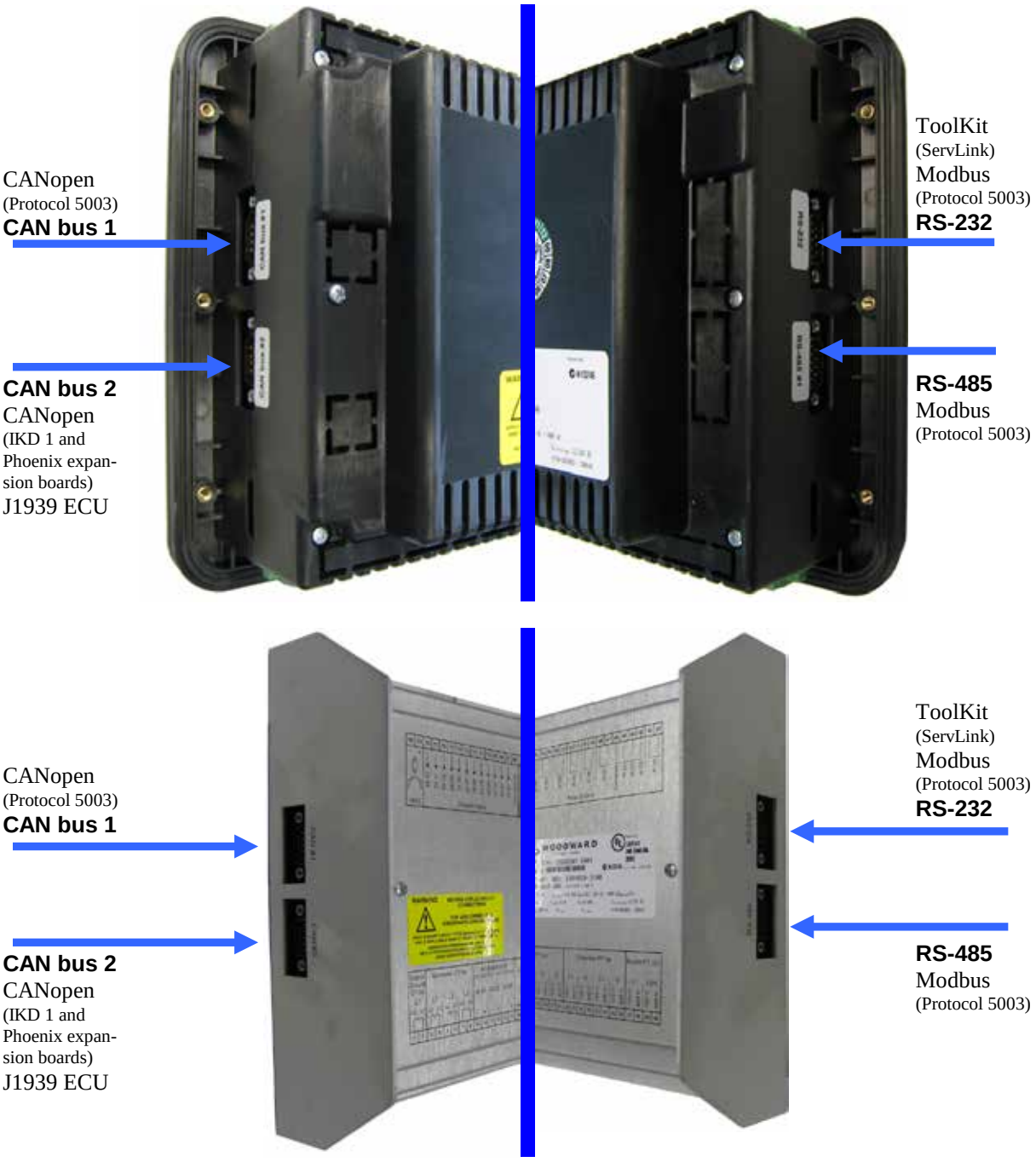

Figure 1-1: easYgen - interface overview

CAN Interfaces

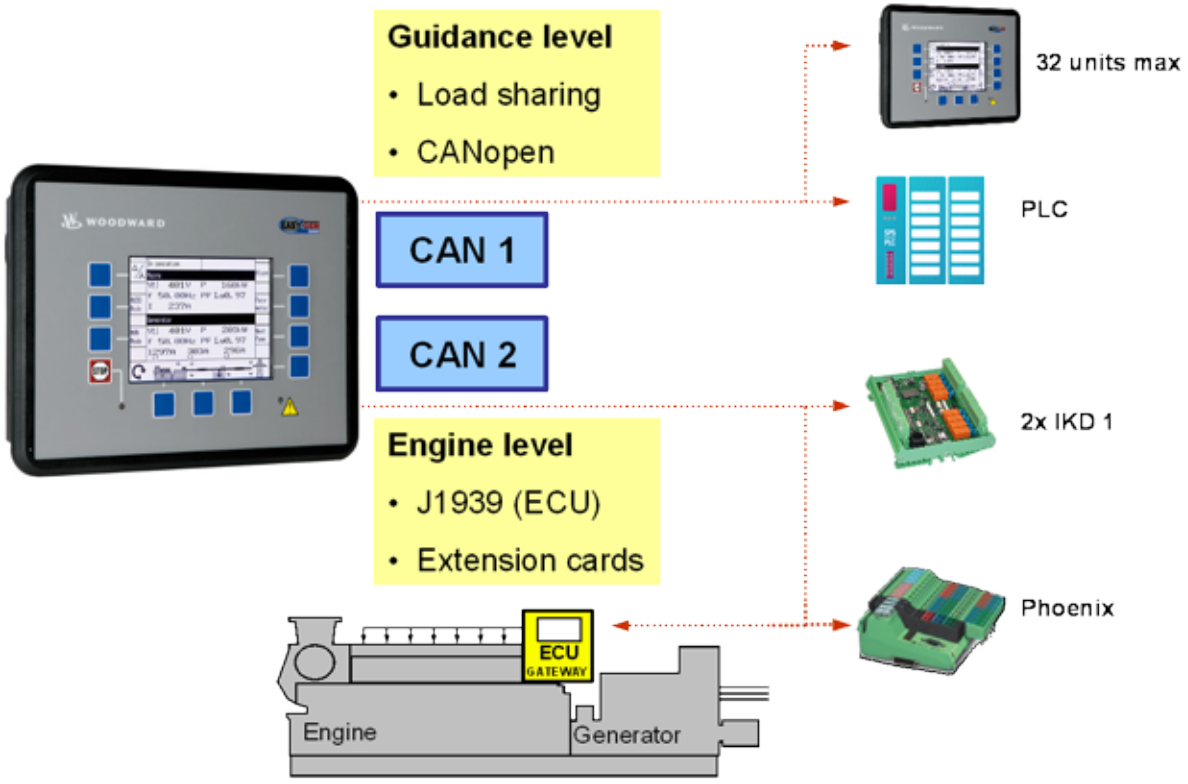

Figure 1-2: Interface overview - CAN interfaces

CAN Interface 1 – Freely Configurable CANopen Interface

CAN interface 1 is a freely configurable CANopen interface with 3 RPDOs (receive boxes), 4 TPDOs (send boxes), and 4 additional Server SDOs.

CAN Interface 2 (Engine Bus)

The CAN interface 2 supports the CANopen and J1939 protocol simultaneously.

Pre-Configured CANopen Interface

CAN interface 2 is pre-configured for several expansion units. These include the I/O expansion boards Woodward IKD 1, Phoenix BK 16DiDo, and Phoenix Co 16DiDo.
The following combinations of I/O expansion boards are possible:

Function RPDO 1 \ 2	OFF	IKD 1 #1	IKD 1 #2	BK 16DiDo	Co 16DiDo
OFF	YES	YES	YES	YES	YES
IKD 1 #1	YES	NO	YES	NO	NO
IKD 1 #2	YES	YES	NO	NO	NO
BK 16DiDo	YES	NO	NO	YES	NO
Co 16DiDo	YES	NO	NO	NO	YES

Table 1-3: CAN interface 1 - I/O board combination matrix

RPDO 1 and RPDO 2 of CAN interface 2 the communication with the configured expansion units) may be monitored individually. Refer to the Configuration Manual 37224 for more information about this monitoring function.

J1939 Interface

The J1939 protocol enables to connect different Engine Control Units (ECUs) with the easYgen on the CAN bus. The following ECUs are supported:

- Woodward EGS
- Scania S6
- MTU ADEC
- Deutz EMR2
- Volvo EMS2
- Volvo EDC4
- MAN MFR/EDC7
- SISU EEM2/EEM3

NOTE

If a Volvo EDC4 or a Deutz EMR3 ECU is utilized, all settings are to be configured as described for the Deutz EMR2 ECU in this manual.

Serial Interfaces

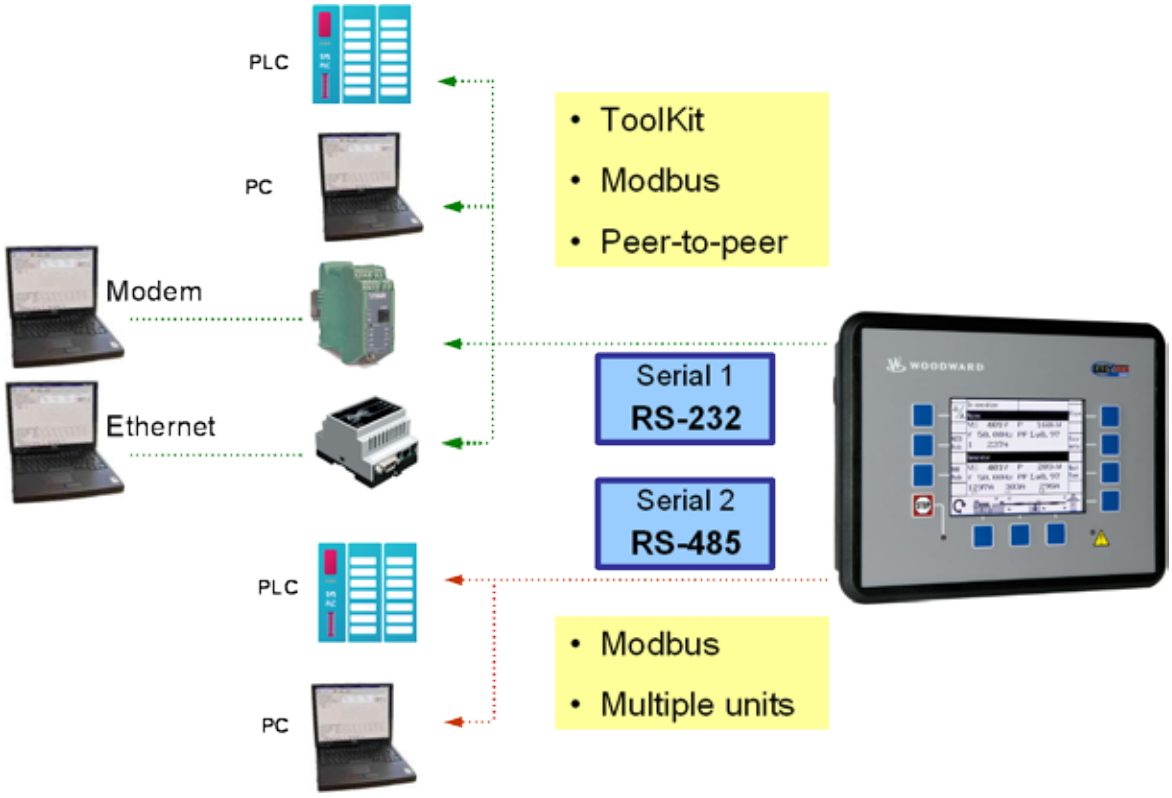

Figure 1-3: Interface overview - serial interfaces

Serial Interface 1 – RS-232

A freely configurable RS-232 interface is provided to serve as a local service interface for configuring the unit and visualize measured data. It is possible to connect a modem for remote control and alarm signaling. The serial interface 1 provides ServLink and Modbus protocol simultaneously.

Serial Interface 2 – RS-485

A freely configurable RS-485 Modbus RTU Slave interface is provided to add PLC connectivity. It is also possible to configure the unit, visualize measured data and alarm messages, and control the unit remotely.

RS-485 Modbus Half/Full Duplex Application

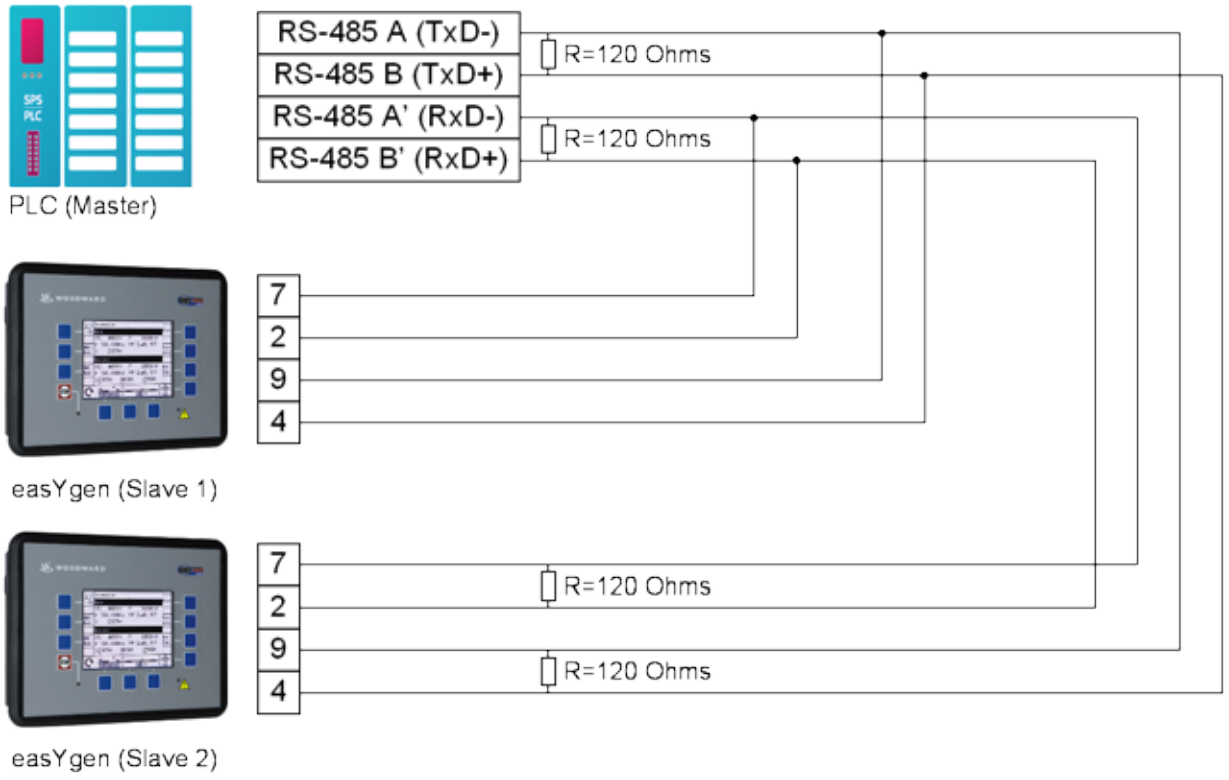

Figure 1-4: Interface overview - serial interface Modbus full-duplex

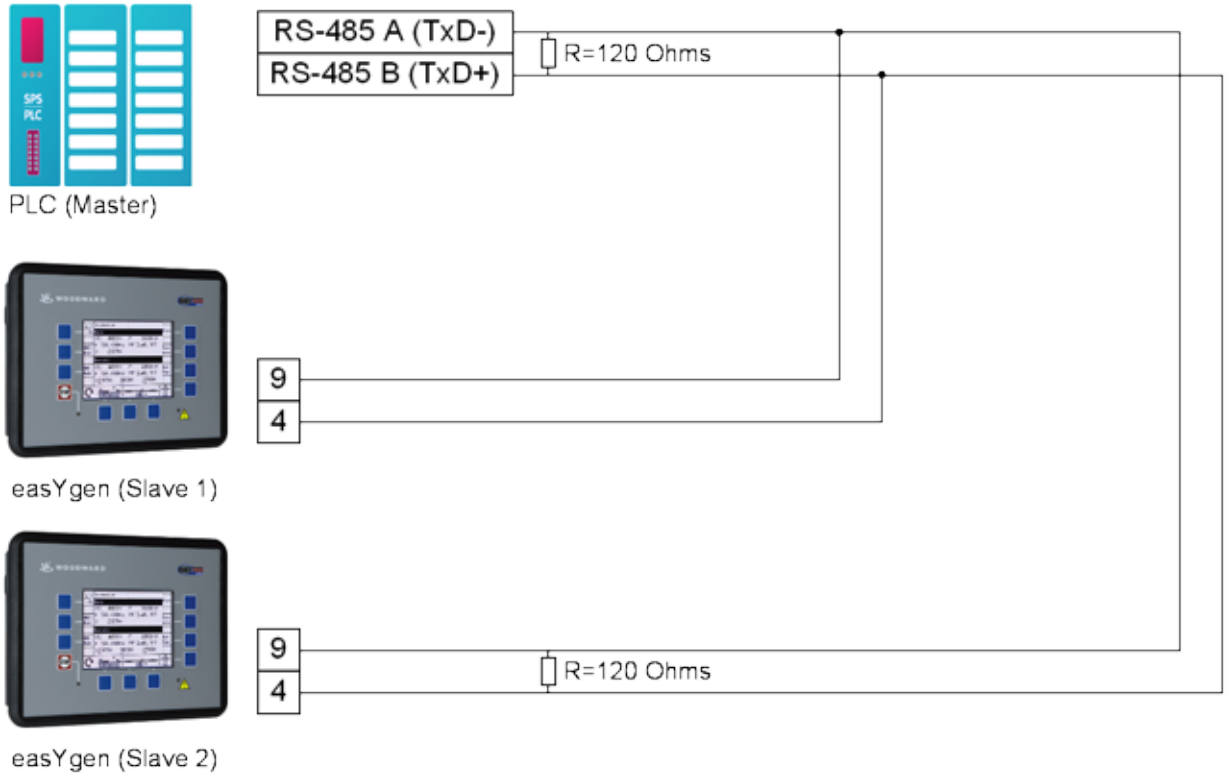

Figure 1-5: Interface overview - serial interface Modbus half-duplex

Chapter 2.
CAN Bus Configuration

CAN Interface Parameters

NOTE

The parameters in the following section are an excerpt of the Configuration Manual 37224. Refer to this manual for all parameters.

Configure CAN Interface 1

EN Baudrate
DE Baudrate
CL2 3156 {0} Ü {1o} Ü {1oc} Ü {2oc} Ü

CAN bus 1: Baud rate **20 / 50 / 100 / 125 / 250 / 500 / 800 / 1,000 kBaud**

This parameter defines the used Baud rate. Please note, that all participants on the CAN bus must use the same Baud rate.

EN Node-ID CAN-Bus 1
DE Node-ID CAN-Bus 1
CL2 8950 {0} Ü {1o} Ü {1oc} Ü {2oc} Ü

CAN bus 1: Node ID **1 to 127**

A number that is unique to the control must be set in this parameter so that this control unit can be correctly identified on the CAN bus. This address number may only be used once on the CAN bus. All additional addresses are calculated based on this unique device number.

EN CANopen Master
DE CANopen Master
CL2 8993 {0} Ü {1o} Ü {1oc} Ü {2oc} Ü

CAN bus 1: CANopen Master **Default Master / On / Off**

One bus participant must take over the network management and put the other participants into "operational" mode. The easYgen is able to perform this task.

Default Master The unit starts up in "operational" mode and sends a broadcast message (Start_Remote_Node) after a short delay (the delay is the Node ID (parameter 8950) in seconds, i.e. if the Node ID is configured to 2, the message will be sent after 2 seconds). If more than one easYgen is configured to Default Master, the unit with the lower Node ID will take over control. Therefore, the CAN bus devices, which are intended to act as Default Master should be assigned a low Node ID. No other device on the CAN bus (except the easYgens) may operate as Master).

On..... The unit is the CANopen Master and automatically changes into operational mode and sends broadcast messages (Start_Remote_Node), which cause all other units to change into operational mode as well.

Off ..... The unit is a CANopen Slave. An external Master must change into operational mode.

NOTE

If CANopen Master (parameter 8993) is configured to "Off", the Master controller (for example a PLC) must send a "Start_Remote_node" message to initiate the load share message transmission of the easYgen.

If no "Start_Remote_node" message would be sent, the complete system would not be operational.

EN Producer heartbeat time
DE Producer heartbeat time
CL2 {0} {1o} {1oc} {2oc}
9120 Ü Ü Ü Ü

CAN bus 1: Producer heartbeat time **0 to 65530 ms**

Independent from the CANopen Master configuration, the unit transmits a heartbeat message with this configured heartbeat cycle time. If the producer heartbeat time is equal 0, the heartbeat will only be sent as response to a remote frame request. The time configured here will be rounded up to the next 20 ms step.

EN COB ID SYNC Message
DE COB ID SYNC Message
CL2 {0} {1o} {1oc} {2oc}
9100 Ü Ü Ü Ü

CAN bus 1: COB ID SYNC Message **1 to FFFFFFFF**

This parameter defines whether the unit generates the SYNC message or not.

Complies with CANopen specification: object 1005, subindex 0; defines the COB ID of the synchronization object (SYNC). The structure of this object is shown in the following tables:

UNSIGNED 32 The bits in question are found on the following position

31 (MSB)	30	29	28-11	10-0 (LSB)
X	0/1	X	00000000000000000	11 bit identifier

The meaning of the bits is as follows

bit number	value	meaning
31 (MSB)	X	N/A
30	0 1	Unit does not generate SYNC message Unit generates SYNC message
29	X	N/A
28-11	0	always
10-0 (LSB)	X	bits 10-0 of SYNC COB ID

Producer SYNC Message time
DE Producer SYNC Message time
CL2 {0} {1o} {1oc} {2oc}
8940 Ü Ü Ü Ü

CAN bus 1: Sending time for SYNC Message **0 to 65000 ms**

This is the cycle time of the SYNC message. If the unit is configured for this function (parameter 9100) it will send the SYNC message with this interval. The time configured here will be rounded up to the next 10 ms step.

Additional Server SDOs (Service Data Objects)

NOTE

The CAN bus is a field bus and subject to various disturbances. Therefore, it cannot be guaranteed that every request will be answered. We recommend to repeat a request, which is not answered within reasonable time.

NOTE

The first Node ID is the standard Node ID of CAN interface 1 (parameter 8950).

EN 2. Node-ID
DE 2. Node-ID
CL2 {0} {1o} {1oc} {2oc}
33040 Ü Ü Ü Ü

CAN bus 1: Additional Server SDOs - 2. Node ID **0 to 127**

In a multi-master application, each Master needs its own identifier (Node ID) from the unit. in order to send remote signals (i.e. remote start, stop, or acknowledge) to the unit. The additional SDO channel will be made available by configuring this Node ID to a value different than zero. This is the additional CAN ID for the PLC.

EN 3. Node-ID
DE 3. Node-ID
CL2 {0} {1o} {1oc} {2oc}
33041 Ü Ü Ü Ü

CAN bus 1: Additional Server SDOs - 3. Node ID **0 to 127**

In a multi-master application, each Master needs its own identifier (Node ID) from the unit. in order to send remote signals (i.e. remote start, stop, or acknowledge) to the unit. The additional SDO channel will be made available by configuring this Node ID to a value different than zero. This is the additional CAN ID for the PLC.

EN 4. Node-ID
DE 4. Node-ID
CL2 {0} {1o} {1oc} {2oc}
33042 Ü Ü Ü Ü

CAN bus 1: Additional Server SDOs - 4. Node ID **0 to 127**

In a multi-master application, each Master needs its own identifier (Node ID) from the unit. in order to send remote signals (i.e. remote start, stop, or acknowledge) to the unit. The additional SDO channel will be made available by configuring this Node ID to a value different than zero. This is the additional CAN ID for the PLC.

EN 5. Node-ID
DE 5. Node-ID
CL2 {0} {1o} {1oc} {2oc}
33043 Ü Ü Ü Ü

CAN bus 1: Additional Server SDOs - 5. Node ID **0 to 127**

In a multi-master application, each Master needs its own identifier (Node ID) from the unit. in order to send remote signals (i.e. remote start, stop, or acknowledge) to the unit. The additional SDO channel will be made available by configuring this Node ID to a value different than zero. This is the additional CAN ID for the PLC.

Receive PDOs (Process Data Objects) 1 to 3

EN COB-ID
DE COB-ID
CL2 {0} {1o} {1oc} {2oc}
9300 Ü Ü Ü Ü
9310
9320

CAN bus 1: Receive PDO 1 - COB ID **1 to FFFFFFFH**

This parameter contains the communication parameters for the PDOs, the device is able to receive.

Complies with CANopen specification: object 1400 (for RPDO 1, 1401 for RPDO 2, and 1402 for RPDO 3), subindex 1. The structure of this object is shown in the following tables:

UNSIGNED 32 The bits in question are found on the following position

31 (MSB)	30	29	28-11	10-0 (LSB)
0/1	X	X	00000000000000000	11 bit identifier

The meaning of the bits is as follows

bit number	value	meaning
31 (MSB)	0 1	PDO exists / is valid PDO does not exist / is not valid
30	X	N/A
29	X	N/A
28-11	0	always
10-0 (LSB)	X	bits 10-0 of COB ID

PDO valid / not valid allows to select, which PDOs are used in the operational state.

EN Number of Mapped Objects
DE Anzahl der Mapped Objekte
CL2 {0} {1o} {1oc} {2oc}
9910 Ü Ü Ü Ü
33855
33860

CAN bus 1: Receive PDO 1 - Number of mapped objects **0 to 4**

This parameter defines the number of valid entries within the mapping record. This number is also the number of the application variables, which shall be received with the corresponding PDO.

Complies with CANopen specification: object 1600 (for RPDO 1, 1601 for RPDO 2, and 1602 for RPDO 3), subindex 0

EN 1. Mapped Object
DE 1. Mapped Objekt
CL2 {0} {1o} {1oc} {2oc}
9911 Ü Ü Ü Ü
33856
33861

CAN bus 1: Receive PDO 1 - 1. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1600 (for RPDO 1, 1601 for RPDO 2, and 1602 for RPDO 3), subindex 1

EN 2. Mapped Object
DE 2. Mapped Objekt
CL2 {0} {1o} {1oc} {2oc}
9912 Ü Ü Ü Ü
33857
33862

CAN bus 1: Receive PDO 1 - 2. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1600 (for RPDO 1, 1601 for RPDO 2, and 1602 for RPDO 3), subindex 2

EN 3. Mapped Object
DE 3. Mapped Objekt
CL2 {0} {1o} {1oc} {2oc}
9913 Ü Ü Ü Ü
33858
33863

CAN bus 1: Receive PDO 1 - 3. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1600 (for RPDO 1, 1601 for RPDO 2, and 1602 for RPDO 3), subindex 3

EN 4. Mapped Object
DE 4. Mapped Objekt
CL2 {0} {1o} {1oc} {2oc}
9914 Ü Ü Ü Ü
33859
33864

CAN bus 1: Receive PDO 1 - 4. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1600 (for RPDO 1, 1601 for RPDO 2, and 1602 for RPDO 3), subindex 4

Figure 2-1 shows the principle of PDO mapping.

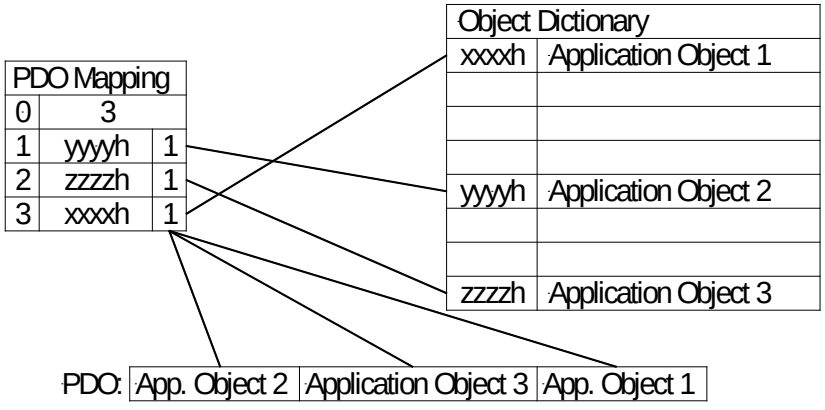

Figure 2-1: Interfaces - principle of PDO mapping

Transmit PDOs (Process Data Objects) 1 to 4

EN COB-ID
DE COB-ID
CL2 {0} {1o} {1oc} {2oc}
9600 Ü Ü Ü Ü
9610
9620
9630

CAN bus 1: Transmit PDO 1 - COB ID — 1 to FFFFFFFH

This parameter contains the communication parameters for the PDOs the unit is able to transmit. The unit transmits data (i.e. visualization data) on the CAN ID configured here.

Complies with CANopen specification: object 1800 for (TPDO 1, 1801 for TPDO 2, 1802 for TPDO 3, and 1803 for TPDO 4), subindex 1. The structure of this object is shown in the following tables:

UNSIGNED 32 The bits in question are found on the following position

31 (MSB)	30	29	28-11	10-0 (LSB)
0/1	X	X	00000000000000000	11 bit identifier

The meaning of the bits is as follows

bit number	value	meaning
31 (MSB)	0	PDO exists / is valid
	1	PDO does not exist / is not valid
30	X	N/A
29	X	N/A
28-11	0	always
10-0 (LSB)	X	bits 10-0 of COB ID

PDO valid / not valid allows to select, which PDOs are used in the operational state.

EN Transmission type
DE Transmission type
CL2 {0} {1o} {1oc} {2oc}
9602 Ü Ü Ü Ü
9612
9622
9632

CAN bus 1: Transmit PDO 1 - Transmission type — 0 to 255

This parameter contains the communication parameters for the PDOs the unit is able to transmit. It defines whether the unit broadcasts all data automatically (value 254 or 255) or only upon request with the configured address of the COB ID SYNC message (parameter 9100).

Complies with CANopen specification: object 1800 (for TPDO 1, 1801 for TPDO 2, 1802 for TPDO 3, and 1803 for TPDO 4), subindex 2. The description of the transmission type is shown in the following table:

transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0	will not be sent				
1-240	X		X		
241-251	will not be sent				
252	will not be sent				
253	will not be sent				
254				X	
255				X	

A value between 1 and 240 means that the PDO is transferred synchronously and cyclically. The transmission type indicating the number of SYNC, which are necessary to trigger PDO transmissions. Receive PDOs are always triggered by the following SYNC upon receiption of data independent of the transmission types 0 to 240. For TPDOs, transmission type 254 and 255 means, the application event is the event timer.

EN Event-timer
DE Event-timer
CL2 {0} {1o} {1oc} {2oc}
9604 Ü Ü Ü Ü
9614
9624
9634

CAN bus 1: Transmit PDO 1 - Event timer — 0 to 65500 ms

This parameter contains the communication parameters for the PDOs the unit is able to transmit. The broadcast cycle for the transmitted data is configured here. The time configured here will be rounded up to the next 5 ms step.

Complies with CANopen specification: object 1800 (for TPDO 1, 1801 for TPDO 2, 1802 for TPDO 3, and 1803 for TPDO 4), subindex 5

EN Number of Mapped Objects
DE Anzahl der Mapped Objekte

	{0}	{1o}	{1oc}	{2oc}
CL2				
9609	Ü	Ü	Ü	Ü
9619				
9629				
9639				

CAN bus 1: Transmit PDO 1 - Number of mapped objects **0 to 4**

This parameter contains the mapping for the PDOs the unit is able to transmit. This number is also the number of the application variables, which shall be transmitted with the corresponding PDO.

Complies with CANopen specification: object 1A00 (for TPDO 1, 1A01 for TPDO 2, 1A02 for TPDO 3, and 1A03 for TPDO 4), subindex 0

EN 1. Mapped Object
DE 1. Mapped Objekt

	{0}	{1o}	{1oc}	{2oc}
CL2				
9605	Ü	Ü	Ü	Ü
9615				
9625				
9635				

CAN bus 1: Transmit PDO 1 - 1. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1A00 (for TPDO 1, 1A01 for TPDO 2, 1A02 for TPDO 3, and 1A03 for TPDO 4), subindex 1

EN 2. Mapped Object
DE 2. Mapped Objekt

	{0}	{1o}	{1oc}	{2oc}
CL2				
9606	Ü	Ü	Ü	Ü
9616				
9626				
9636				

CAN bus 1: Transmit PDO 1 - 2. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1A00 (for TPDO 1, 1A01 for TPDO 2, 1A02 for TPDO 3, and 1A03 for TPDO 4), subindex 2

EN 3. Mapped Object
DE 3. Mapped Objekt

	{0}	{1o}	{1oc}	{2oc}
CL2				
9607	Ü	Ü	Ü	Ü
9617				
9627				
9637				

CAN bus 1: Transmit PDO 1 - 3. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1A00 (for TPDO 1, 1A01 for TPDO 2, 1A02 for TPDO 3, and 1A03 for TPDO 4), subindex 3

EN 4. Mapped Object
DE 4. Mapped Objekt

	{0}	{1o}	{1oc}	{2oc}
CL2				
9608	Ü	Ü	Ü	Ü
9618				
9628				
9638				

CAN bus 1: Transmit PDO 1 - 4. mapped object **0 to 65535**

This parameter contains the information about the mapped application variables. These entries describe the PDO contents by their index. The sub-index is always 1. The length is determined automatically.

Complies with CANopen specification: object 1A00 (for TPDO 1, 1A01 for TPDO 2, 1A02 for TPDO 3, and 1A03 for TPDO 4), subindex 4

NOTE

CANopen allows to send 8 byte of data with each Transmit PDO. These may be defined separately if no pre-defined data protocol is used.

All data protocol parameters with a parameter ID may be sent as an object with a CANopen Transmit PDO.

In this case, the data length will be taken from the data byte column (refer to Appendix B: Data Protocols on page 79):

- **1,2 UNSIGNED16 or SIGNED16**
- **3,4 UNSIGNED16 or SIGNED16**
- **5,6 UNSIGNED16 or SIGNED16**
- **1,2,3,4 UNSIGNED32 or SIGNED32**
- **3,4,5,6 UNSIGNED32 or SIGNED32**
- **etc.**

The object ID is identical with the parameter ID when configuring via front panel or ToolKit.

Configure CAN Interface 2

EN Baudrate
DE Baudrate
CL2 3157 {0} Ü {1o} Ü {1oc} Ü {2oc} Ü

CAN bus 2: Baud rate **20 / 50 / 100 / 125 / 250 / 500 / 800 / 1,000 kBaud**

This parameter defines the used Baud rate. Please note, that all participants on the CAN bus must use the same Baud rate.

EN Function for RPDO 1
DE Funktion für RPDO 1
CL2 9055 {0} Ü {1o} Ü {1oc} Ü {2oc} Ü

CAN bus 2: Function for RPDO 1 **refer to selection below**

The unit provides pre-configured CAN bus settings for the connection of different units. The unit to be connected must be selected here.

no func. .....No external unit is selected for connection. The CAN bus is disabled. Values are not sent or received.
1st IKD.....The unit is pre-configured for the connection of a Woodward IKD 1 expansion board.
2nd IKD .....The unit is pre-configured for the connection of a second Woodward IKD 1 expansion board.
BK 16 DIDO The unit is pre-configured for the connection of a Phoenix Contact BK 16 DIDO expansion board.
Co 16 DIDO.The unit is pre-configured for the connection of a Phoenix Contact Co 16 DIDO expansion board.

EN Function for RPDO 2
DE Funktion für RPDO 2
CL2 9056 {0} Ü {1o} Ü {1oc} Ü {2oc} Ü

CAN bus 2: Function for RPDO 2 **refer to selection below**

The unit provides pre-configured CAN bus settings for the connection of different units. The unit to be connected must be selected here.

no func. .....No external unit is selected for connection. The CAN bus is disabled. Values are not sent or received.
1st IKD.....The unit is pre-configured for the connection of a Woodward IKD 1 expansion board.
2nd IKD .....The unit is pre-configured for the connection of a second Woodward IKD 1 expansion board.
BK 16 DIDO The unit is pre-configured for the connection of a Phoenix Contact BK 16 DIDO expansion board.
Co 16 DIDO.The unit is pre-configured for the connection of a Phoenix Contact Co 16 DIDO expansion board.

J1939 Interface

EN J1939 device addresses
DE J1939 Geräte-Adresse
CL2 {0} {1o} {1oc} {2oc}
15106 Ü Ü Ü Ü

J1939 Interface: Device address **0 to 255**

The easYgen sends J1939 request and control messages with this ID. It must be changed for different ECU types according to the following table. The ECU listens only to to control messages, if they are sent on the correct address.

Scania S6	EMR2 Deutz	EMS2 Volvo	MTU ADEC	Woodward EGS	MAN EDC7	SISU EEM2/3
39	3	17	1	234	253	n/a

Details may be found in the manual of the genset control.
Note: Changing this parameter becomes only effective after restarting the unit.

EN Engine control address
DE Adresse Motorsteuerung
CL2 {0} {1o} {1oc} {2oc}
15107 Ü Ü Ü Ü

J1939 Interface: Engine control address **0 to 255**

Configures the address of the J1939 device, which is controlled.

Scania S6	EMR2 Deutz	EMS2 Volvo	MTU ADEC	Woodward EGS	MAN EDC7	SISU EEM2/3
0	0	0	128	0	0	0 / (1)

EN Reset previous act. DTCs - DM3
DE Quittieren passiver Fehler DM3
CL2 {0} {1o} {1oc} {2oc}
15108 Ü Ü Ü Ü

J1939 Interface: Reset previously active DTCs - DM3 **YES / NO**

If this parameter is set YES, a DM3 message "Acknowledge passive faults" is sent. After that this parameter is reset automatically to NO.
As a result alarms (DM2) which no longer apply are cleared.

EN SPN version
DE SPN Version
CL2 {0} {1o} {1oc} {2oc}
15103 Ü Ü Ü Ü

J1939 Interface: SPN version **Version 1 / Version 2 / Version 3**

The J1939 protocol provides 4 different versions for formatting Suspect Parameter Number. This is important for a correct display of the alarm messages.
With this parameter it is defined if formatting occurs according to Version 1, Version 2, or Version 3. Formatting according to Version 4 is identified automatically. Details may be found in the engine control unit (ECU) J1939 manual.

EN Device type
DE Betriebsmodus
CL2 {0} {1o} {1oc} {2oc}
15102 Ü Ü Ü Ü

J1939 Interface: Device type **refer to selection below**

The J1939 interface of this device may be operated with different engine control units. This parameter determines the operating mode of the used ECU.
Off.....The J1939 interface is disabled. No messages will be sent/received.
Standard.....Standard J1939 messages will be sent or received.
S6 Scania.....Standard J1939 messages plus special S6 Scania messages.
EMR2 Deutz Standard J1939 messages plus special Deutz EMR2 messages.
EMS2 Volvo Standard J1939 messages plus special Volvo EMS2 messages.
ADEC MTU Standard J1939 messages plus special MTU ADEC messages.
EGS.....Standard J1939 messages plus special Woodward EGS messages.
MAN.....Standard J1939 messages plus special MAN EDC7 messages.
SISU EEM...Standard J1939 messages plus special SISU EEM2/3 messages.

EN ECU remote controlled
DE Fernsteuern der ECU über J1939
CL2 {0} {1o} {1oc} {2oc}

J1939 Interface: ECU remote control via J1939 **ON / OFF**

ON.....The unit sends J1939 control messages to the ECU. Depending on

15127	ü	ü	ü	ü	the selected device type (Parmater 15102), contains a specific selection of commands. Available messages are speed deviation and droop for all ECUs as well as engine start/stop, enable idle mode, rated speed switch and preglow for some ECUs. Refer to Appendix A: Supported J1939 ECUs & Remote Control Messages on page 77 for the commands, the different ECUs are able to receive. **OFF** ..... The ECU remote control via the J1939 protocol will be disabled.

EN					Speed deviation ECU
DE					Drehzahlhub
CL2	{0}	{1o}	{1oc}	{2oc}	
5537	Ü	Ü	Ü	Ü	

J1939 Interface: Speed deviation **0 to 1400 rpm**

> i This parameter is only visible if ECU remote controlled (Parameter 15127) is configured to "ON".

This parameter adjusts the range of the speed deviation around the rated speed, which is sent to the ECU.
It relates to the engine rated speed (parameter 1601). There are two methods of sending the speed set point to the ECU: With a speed offset and a speed setpoint. The frequency and power control must be configured to "PID".

Speed offset: Scania S6, Volvo EMS2, EGS
The easYgen sends a speed offset with a range of 0 to 100% (every 20 ms). 50% = rated speed. There is also an internal speed offset configured in the ECU, this parameter determines what corresponds with 0% or 100%. If there is a positive and a negative speed offset, they should be symmetrical in the ECU. We recommend to have the same speed offset configured in the ECU and in this parameter here. A different setting will result in an additional "controller gain".
How to test this parameter during commissioning:
Isolated operation: Disable the frequency controller and change parameter 5508 for the initial state between 0 and 100%, the engine should change the speed as follows:
0 = rated speed – negative speed offset from ECU
50 = rated speed
100 = rated speed + positive speed offset from ECU
Mains parallel operation: Check with the set point in the display if the engine is able to deliver the full power.

Speed set point: Deutz EMR, MTU ADEC, EGS, SISU, Standard
The easYgen sends a speed set point in rpm (every 10 ms) that varies around the rated speed in the range of +/- the speed deviation.
How to test this parameter during commissioning:
Isolated operation: Disable the frequency controller and change parameter 5508 for the initial state between 0 and 100%, the engine should change the speed as follows:

0	= rated speed – speed deviation ECU	e.g. 1500 – 120	= 1380rpm
50	= rated speed	e.g.	= 1500rpm
100	= rated speed + speed deviation ECU	e.g. 1500 + 120	= 1620rpm

Note: Keep this value as small as possible, i.e. do not enter a speed deviation of 500, if the engine varies only between 1400 and 1600rpm.
Mains parallel operation: Check with the set point in the display if the engine is able to deliver the full power.

NOTE

The Wodward EGS ECU supports both types of speed deviation control and may be configured either to "Speed offset" or "Speed set point".

In mains parallel operation, the EGS can be configured to receive a real power set point from the easYgen to control the power. In this case, real power control must be disabled in the easYgen.

CAN Bus Load Sharing

General Load Share Information

The maximum number of participating easYgen-3000 for load sharing is 32. The CANopen bus load increases with the number of units participating in load sharing.

The following parameters effect the bus load:

- Number of CAN participants
- Baud rate
- Transfer rate of load share messages
- Visualization

We recommend to consider whether all data has to be sent on the CAN bus when planning the CAN bus. It is also possible to send visualization data via RS-485 for example.

Typical Load Share Applications

Load Sharing of up to 8 Gensets + Visualization Data Display

Table 2-1 shows the standard setting for load sharing. These settings allow for load sharing of up to 8 gensets with the fastest transfer rate for the load sharing message and simultaneous transmission of all visualization data.

Index	Parameter	Setting	Comment
3156	Baudrate	250 kBd	
9921	Transfer rate LS fast message	0.10 s	
9602	Transmission type	255	
9604	Event-timer	20 ms	
9609	Number of mapped objects	1	
9605	1. Mapped Object	16382	Protocol 5003

Table 2-1: CAN interface 1 - standard values for load sharing

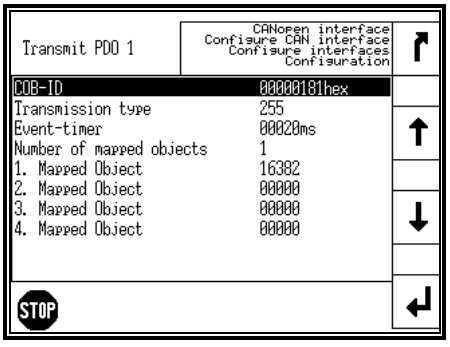

Figure 2-2: CAN interface 1 - standard values for load sharing

Load Sharing of up to 16 Gensets + Visualization Data Display

If you want to perform load sharing with up to 16 gensets with the fastest transfer rate for the load sharing message and simultaneous transmission of all visualization data, the transfer rate of the visualization data (parameter 9604) should be increased to 150 ms.

Load Sharing of up to 32 Gensets without Visualization Data Display

If you want to perform load sharing with up to 32 gensets with the fastest transfer rate for the load sharing message you should disable the simultaneous transmission of visualization data by setting the transmission type (parameter 9602) to "0". The minimum baud rate (parameter 3156) for this application is 250 kBd.

NOTE

Above settings are only recommendations and may vary depending on your specific application.

Measures for Reducing the Bus Load

If you need to reduce the bus load of the load share CAN bus, the following measured may be used:

- Increase the baud rate (parameter 3156) under consideration of the bus length (refer to Installation Manual 37223)
- Reduce the transfer rate of the load share message (parameter 9921)
- Reduce the transfer rate of the visualization message, i.e. the event timer (parameter 9604)
- Disable the transmission visualization data on the CAN bus and use the RS-485 interface to transmit visualization data

Configure Load Share Parameters

EN	Load share Interface			
DE	Schnittstelle Lastverteilung			
CL2	{0}	{1o}	{1oc}	{2oc}
9923	---	---	---	Ü

CAN Interface: load share interface **CAN #1 / OFF**

The interface, which is used for transmitting the load share data is configured here.

EN	Transfer rate LS fast message			
DE	Sendetakt der Lastverteilung			
CL2	{0}	{1o}	{1oc}	{2oc}
9921	---	---	---	Ü

CAN Interface: transfer rate load share fast message **0.10 to 0.30 s**

The transfer rate defines the time delay between two fast CAN messages.
In case of CAN systems with a high bus load (e.g. long distance between the units with low baud rate), a shorter transfer rate (higher time setting) helps to reduce the bus load.

EN	Load Share CAN-ID			
DE	Lastverteilungs CAN-ID			
CL2	{0}	{1o}	{1oc}	{2oc}
9920	---	---	---	Ü

CAN Interface: load share CAN ID **2xx Hex / 3xx Hex / 4xx Hex / 5xx Hex**

The first digit of the CAN ID or the range (i.e. 2xx means 200 through 2FF) is configured here. The last two digits will be assigned by the control with the settings from the device number.

Multi-Master Principle

It is important to know that the load share and load-dependent start/stop functionality is subject to a multi-master principle. This means that there is no dedicated master and slave function. Each easYgen decides for itself how it has to behave. The benefit is that the failure of a master control may cause a complete loss of this functionality. Each control is also responsible for controlling common breakers like a mains circuit or generator group breaker.

Definition of CANopen Protocol Descriptions

If a data protocol is used, a CAN message looks like this:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
MUX	Data byte	Data byte	Data byte	Data byte	Data byte	Data byte	internal

The MUX byte is counted up, the meaning of the data byte changes according to the value of the MUX byte.
In the protocol tables is listed which parameter at which MUX on which position is transmitted.
The meaning of the parameter can be taken by means of the number of the parameter description ("CANopen Mapping parameter").

Example:

MUX	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
1	118				147		internal

In MUX 1 (byte 1 has got value 1) the value of parameter 118 is included in the byte 2 up to byte 5 (mains voltage 1-2).
In byte 6 up to byte 7 the value of parameter 147 is included (mains frequency).
Byte 8 includes internal definitions and can be ignored.
The data format is low Byte/high Byte (compare with CiA draft standard 01 on page 26).

Unsigned Integer

UNSIGNED type data has positive integers as values. The range is between 0 and 2n-1. The data is shown by the bit sequence of length n.

Bit sequence $b = b_0$ to b_{n-1}

shows the value $UNSIGNEDn(b) = b_{n-1}*2^{n-1}+....+b_1*2^1+b_0*2^0$

NOTE

Please note that the bit sequence starts on the left with the least significant byte.

Example: Value 266 = 10A hex of type UNSIGNED16 is transmitted on the bus in two octets, first 0A hex and then 01 hex.

The following UNSIGNED data types are transmitted as follows:

Octet Number	1.	2.	3.	4.	5.	6.	7.	8.
UNSIGNED8	b_7 to b_0							
UNSIGNED16	b_7 to b_0	b_{15} to b_8						
UNSIGNED24	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}					
UNSIGNED32	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}				
UNSIGNED40	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}			
UNSIGNED48	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}	b_{47} to b_{40}		
UNSIGNED56	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	
UNSIGNED64	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	b_{63} to b_{56}

Table 2-2: CAN bus - transfer syntax for data type UNSIGNEDn

Signed Integer

SIGNED type data has integers as values. The range is between 0 and 2^n-1. The data is shown by the bit sequence of length n.

Bit sequence $b = b_0$ to b_{n-1}

shows the value $SIGNEDn(b) = b_{n-2}*2^{n-2}+...+b_1*2^1+b_0*2^0$ if $b_{n-1} = 0$

and with two's complement $SIGNEDn(b) = SIGNEDn(\hat{}b)-1$ if $b_{n-1} = 1$

NOTE

Please note that the bit sequence starts on the left with the least significant byte.

Example: The value -266 = FEF6 hex of type SIGNED16 is transmitted in two octets, first F6 hex and then FE hex.

The following SIGNED data types are transmitted as follows:

Octet Number	1.	2.	3.	4.	5.	6.	7.	8.
SIGNED8	b_7 to b_0							
SIGNED16	b_7 to b_0	b_{15} to b_8						
SIGNED24	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}					
SIGNED32	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}				
SIGNED40	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}			
SIGNED48	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}	b_{47} to b_{40}		
SIGNED56	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	
SIGNED64	b_7 to b_0	b_{15} to b_8	b_{23} to b_{16}	b_{31} to b_{24}	b_{39} to b_{32}	b_{47} to b_{40}	b_{55} to b_{48}	b_{63} to b_{56}

Table 2-3: CAN bus - transfer syntax for data type INTEGERn

J1939 Protocol Display Messages

Messages of a device (for example an ECU) are received on the CAN bus according to J1939 protocol and are shown on the display.
This function can be used via the CAN interface parallel to the CANopen protocol or to ToolKit.
The Baud rate is similar for all devices connected to the CAN bus independent of the selected protocol.

Displayed Messages

DM1/DM2

The first 10 active alarm messages (Active Diagnostic Trouble Codes - DM1) and the first 10 unacknowledged alarm messages (Previously Active Diagnostic Trouble Codes - DM2) with SPN, FMI, and OC are displayed. The state of the lamps DM1/2 is always displayed.

SPN (= Suspect Parameter Number) indicates the measuring value that the alarm code is referring (e.g. SPN = 100 corresponds to oil pressure).

FMI (= Failure Mode Indicator) specifies the alarm more precisely (e.g. FMI = 3 means: value is valid but higher than the standard value.)

OC: (Occurrence Count) indicates how often an alarm occurred. IF OC = 0, no alarm is present

PGN (= Parameter Group Number) defines a particular combination of SPNs.

Refer to the J1939 specification for a list of all SPNs.

Standard Messages

Suspect Parameter Number	Parameter Group Number	Description	Resolution	Display with defective sensor in ToolKit	Display with missing sensor value in ToolKit
91	61443	Throttle position	0.1%	6553.4%	6553.5%
92	61443	Load at current speed	1%	65534%	65535%
98	65263	Engine oil level	0.1%	6553.4%	6553.5%
100	65263	Engine oil pressure	1kPa	65534kPa	65535kPa
102	65270	Boost pressure	1kPa	65534kPa	65535kPa
105	65270	Intake manifold temperature	1°C	32766°C	32767°C
108	65269	Barometric pressure	0.1kPa	65534kPa	65535kPa
110	65262	Engine coolant temperature	°C	32766°C	32767°C
111	65263	Coolant level	0.1%	6553.4%	6553.5%
172	65269	Air inlet temperature	1°C	32766°C	32767°C
173	65270	Exhaust gas temperature	0.01°C	21474836.46°C	21474836.47°C
174	65262	Fuel temperature	1°C	32766°C	32767°C
175	65262	Engine oil temperature	0.01°C	21474836.46°C	21474836.47°C
183	65266	Fuel rate	0.01 l/h	21474836.46 L/h	21474836.47 L/h
190	61444	Engine speed	0.1rpm	214748364.6rpm	214748364.7rpm
247	65253	Total engine hours [1]	1 h	2147483646h	2147483647h
513	61444	Actual engine torque	1%	32766%	32767%

[1] If the total engine hours sent by the ECU exceed 419,000 hrs, the display in the unit is not correct anymore

Table 2-4: J1939 protocol - standard messages

Data transmission by Engine Control Unit (ECU)

- If the sent values exceed the limits of the specification, the displayed value is not defined.
- If a value of the ECU is not sent or sent as not available or defective, the value will be displayed as indicated above.

Special EMR Messages

Suspect Parameter Number	Parameter Group Number	Description
Engine stop	65301 (FF15h)	As Type 0 to 9

Type	Message acc. to EMR manual	Display in unit	Display in ToolKit
0	Engine stop information	Type 0	no stop
1	Engine safety	Type 1	Type 1. Engine safety
2	CAN message engine stop request	Type 2	Type 2: CAN message engine stop request
3	Low oil pressure	Type 3	Type 3: low oil pressure
4	Low oil level	Type 4	Type 4: low oil level
5	High coolant temp	Type 5	Type 5: high coolant temp
6	Low coolant level	Type 6	Type 6: low coolant level
7	Intake manifold temp	Type 7	Type 7: intake manifold temp
8	Reserved (Stop via SAE-J1587)	Type 8	Type 8: reserved (Stop via SAE-J1587)
9	Reserved (Stop via VP2)	Type 9	Type 9: reserved (Stop via VP2)

Table 2-5: J1939 protocol - special EMR messages

Special S6 Messages

Suspect Parameter Number	Parameter Group Number	Description	Display in unit	Display in ToolKit
DLN2-Proprietary	65409 (FF81h)	Assessed messages: Low engine oil level High engine oil level Low oil pressure High coolant temperature	NO ---- YES	NO Sensor defect YES

Table 2-6: J1939 protocol - special S6 messages

If DLN2 does not transmit, the following is valid:

- the screens in the device are suppressed
- "missing" is displayed in ToolKit

Remote Control via CAN

Remote Start/Stop and Acknowledgement

Refer to the Performing Remote Start/Stop and Acknowledgement section in the Special Application Examples section of the application manual 37226 for detailed information.

The easYgen may be started, stopped, or acknowledged with CAN/Modbus. Therefore, two logical command variables have to be configured with the *LogicsManager*:

04.13 Remote request
04.14 Remote acknowledge

Two different methods to perform a remote start/stop/acknowledgement are detailed in the following. These are "Remote start/stop/acknowledgement via RPDO" and "Remote start/stop/acknowledgement via default SDO communication channel". The advantages and the disadvantages of these two methods are as follows:

Comparison of the Two Methods

Start/Stop/Acknowledgement via RPDO

- Classical communication for CANopen devices
- One message
- **No** validation of the received answer
- Only working in operational mode

Start/Stop/Acknowledgement via Default SDO Communication Channel

- Configuration process
- Two messages
- Validation answer, if message has been received by the unit
- May take longer in case of communication with two messages

Remote Start/Stop/Acknowledgement via RPDO

Configuration of CAN Interface 1

Be sure to enable CAN-Open Master if there is no PLC taking over the master function.

Navigate to the "Set up CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Set up CAN interface 1

NOTE

The display field in the upper right corner indicates the path to the displayed screen by displaying the last four display screen levels, cf. Figure 2-3 with above navigation path.

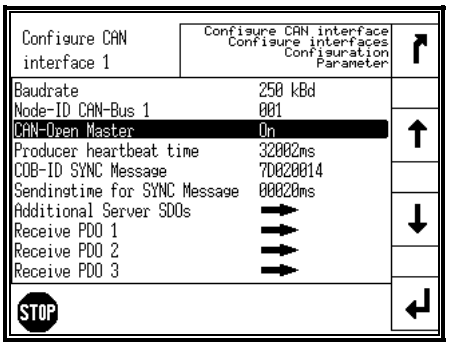

Figure 2-3: Display screen - configure CAN interface 1

Navigate to the parameter "CAN-Open Master" by using the ↓ and ↑ softkeys. Press ↵ to change the parameter. Change the parameter value to "On" using the + and − softkeys. Confirm the change with the ↵ softkey.

Configuration of the RPDO

Press ↱ until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the ↓ and ↑ softkeys and press ↵ to enter the "Receive PDO 1" screen.

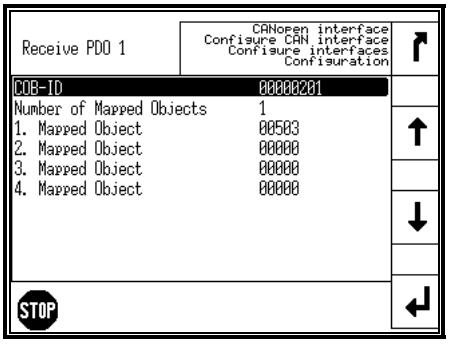

Figure 2-4: Display screen - Receive PDO 1

Configure the following values for the "Receive PDO 1" parameters using the ↓ and ↑ as well as →, + and − softkeys and Confirm the change by pressing the ↵ softkey:

COB-ID	00000201 (hex)
Number of Mapped Objects	1
1. Mapped Object	00503

Setting the COB-ID to 201 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameter 503 of the device as mapped object 1.

NOTE

Refer to Chapter 3.Appendix B: Additional Data Protocol Parameters starting on page 96 for a list of additional parameter groups.

CANopen Request

Figure 2-5 shows exemplary request data for the device on the CANopen bus. The data (hex) shows the state of parameter 503 to achieve the required control.

Figure 2-5: CANopen request data

Remote Start/Stop via Default SDO Communication Channel

Another possibility for a remote start/stop/acknowledgement is to send the request via default SDO communication channel.
The device listens to the CAN ID 600 (hex) + Node ID internally to perform the desired control, the reply is on CAN ID 580 (hex) + Node ID.

The following examples show the request format on CANopen with different Node IDs.

The request on the bus is sent via the control parameter 503 of the device.
The hexadecimal value 2000 is calculated internally.
503(decimal) -- 1F7 (hexadecimal)
1F7+2000 (hexadecimal) = 21F7

Please note that high and low byte are exchanged in the sent address.
The data (hex) shows the state of parameter 503 to achieve the required control.

Node ID 1 standard

Figure 2-6 shows exemplary request data for the device on the CANopen bus.

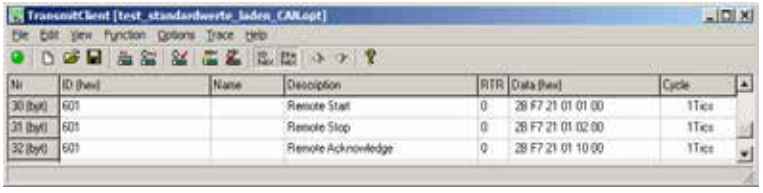

Figure 2-6: CANopen request data for Node ID 1

Node ID (not standard value)

If the Node ID of the device is intended to be different from the standard value, the parameter "Node-ID CAN-Bus 1" must be configured accordingly. Node ID 2 is used in the following example.

Press ↱ until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

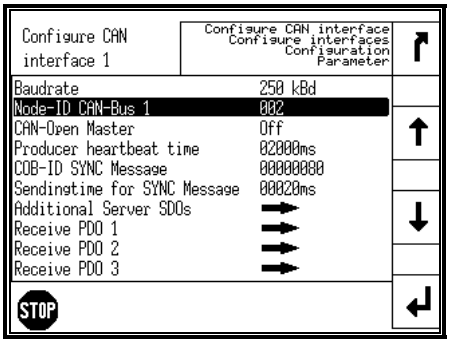

Figure 2-7: Display screen - Configure CAN interface 1

Configure the following value for the "Node-ID CAN-Bus 1" parameter using the ↓ and ↑ as well as →, + and − softkeys and Confirm the change by pressing the ↵ softkey:

Node-ID CAN-Bus 1 002

With this setting, the Node ID of the CAN interface 1 is set to 002.

The request on the bus is sent via the control parameter 503 of the device.
The hexadecimal value 2000 is calculated internally.
503(decimal) -- 1F7 (hexadecimal)
1F7+2000 (hexadecimal) = 21F7

Please note that high and low byte are exchanged in the sent address.
The data (hex) shows the state of parameter 503 to achieve the required control.

Figure 2-8 shows exemplary request data for the device on the CANopen bus.

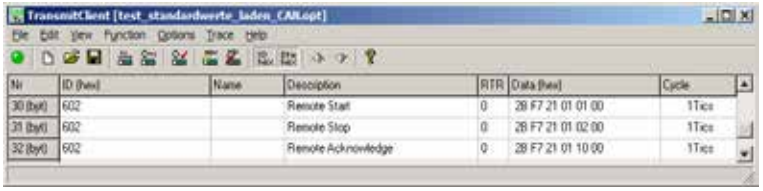

Figure 2-8: CANopen request data for Node ID 2

Additional SDO Communication Channels

It is also possible to allow several PLCs to start/stop/acknowledge the unit in addition to the default SDO communication channel. Four additional SDO communication channels are provided for this. The additional SDO 127 (decimal) is used in the following example.

Press [↱] until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Additional Server SDOs" by using the [↓] and [↑] softkeys and press [↵] to enter the "Receive PDO 1" screen.

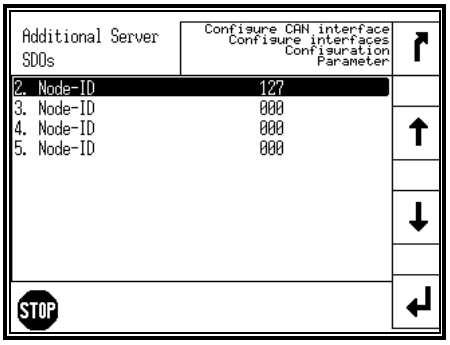

Figure 2-9: Display screen - Additional Server SDOs

Configure the following value for the "2. Node-ID" parameter using the [↓] and [↑] as well as [→], [+] and [−] softkeys and Confirm the change by pressing the [↵] softkey:

2. Node-ID 127 = 7F (hex)

With this setting, an additional SDO communication channel is configured to 127.

The control request is equal to the request via default SDO communication channel, but the device will listen to messages including the configured address as well.

The device listens to the CAN ID 600 (hex) + 2. Node ID internally to perform the desired control, the reply from the easYgen is sent on CAN ID 580 (hex) + 2. Node ID.
Receive CAN ID 67F (hex) (600 (hex) + 7F (hex))
Receive CAN ID 5FF (hex) (580 (hex) + 7F (hex))
The same is valid for the additional SDO communication channels 3, 4, and 5.

Figure 2-10 shows exemplary request data for the device on the CANopen bus.

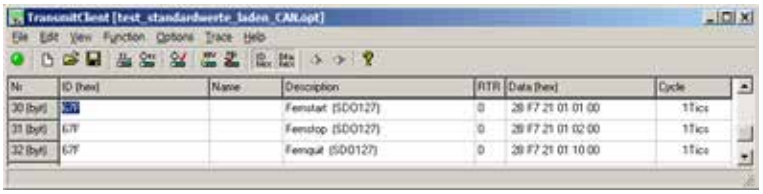

Figure 2-10: CANopen request data for additional Server SDO

NOTE

If parameters are written or read via two or more SDO communication channels at the same time (before the first has answered), the second one will be refused.

Transmitting a Frequency Set Point via CANopen

It is possible to transmit a frequency set point value via the CANopen interface. Prerequisite for the use of a frequency set point via an interface is the configuration of the frequency set point source (parameter ID 5518 for frequency set point 1 source or parameter ID 5519 for frequency set point 2 source; refer to the Configuration Manual 37224 for detailed information). The respective frequency set point source is to be configured to 05.03 "Interface freq.setp.".

Two different methods to transmit a frequency set point via CANopen interface are detailed in the following. Refer to Comparison of the Two Methods on page 28 for the basic differences of these methods.

Transmitting a Frequency Set Point via RPDO

Configuration of CAN Interface 1

Be sure to enable CAN-Open Master if there is no PLC taking over the master function. Refer to Configuration of CAN Interface 1 on page 28 for the configuration of this parameter.

Configuration of the RPDO

Press ↱ until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the ↓ and ↑ softkeys and press ↵ to enter the "Receive PDO 1" screen.

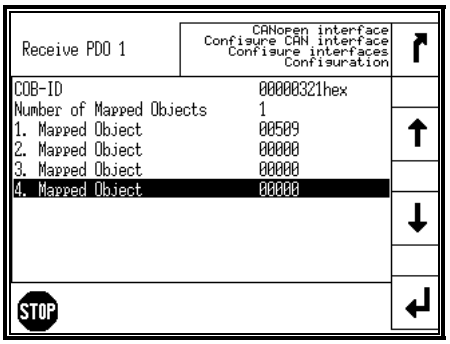

Figure 2-11: Display screen - Receive PDO 1 for frequency set point

Configure the following values for the "Receive PDO 1" parameters using the ↓ and ↑ as well as →, + and − softkeys and Confirm the change by pressing the ↵ softkey:

COB-ID	00000321 (hex)
Number of Mapped Objects	1
1. Mapped Object	00509

Setting the COB-ID to 321 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameter 509 of the device as mapped object 1.

NOTE

Refer to Chapter 3.Appendix B: Additional Data Protocol Parameters starting on page 96 for a list of additional parameter groups.

CANopen Request

Figure 2-12 shows exemplary send data for the device on the CANopen bus. A frequency set point of 50.60 Hz is transmitted (5060 (dec) = 13C4 (hex) -> C4 13 according to the CANopen protocol).

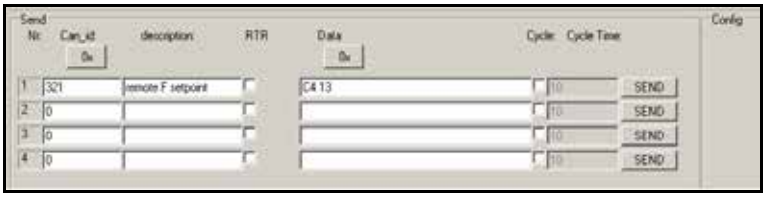

Figure 2-12: CANopen send data for frequency set point

Transmitting a Frequency Set Point via Default SDO Communication Channel

Another possibility for transmitting a frequency set point is to send the value via default SDO communication channel.
The device listens to the CAN ID 600 (hex) + Node ID internally to perform the desired control, the reply is on CAN ID 580 (hex) + Node ID.

The following example shows the send format on CANopen with Node ID 1.

The value is sent on the bus via the control parameter 509 of the device.
The hexadecimal value 2000 is calculated internally.
509(decimal) -- 1FD (hexadecimal)
1FD+2000 (hexadecimal) = 21FD

Please note that high and low byte are exchanged in the sent value.
The data (hex) shows the state of parameter 509 to achieve the required control.

Figure 2-13 shows exemplary send data for the device on the CANopen bus.

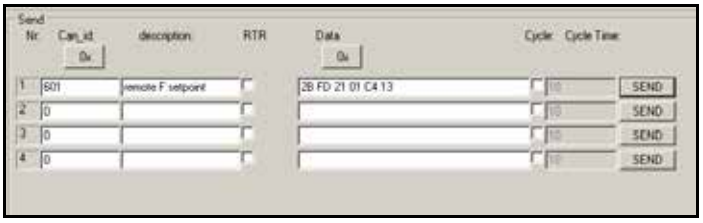

Figure 2-13: CANopen send data for Node ID 1 for frequency set point

Transmitting a Voltage Set Point via CANopen

It is possible to transmit a voltage set point value via the CANopen interface. Prerequisite for the use of a voltage set point via an interface is the configuration of the voltage set point source (parameter ID 5618 for voltage set point 1 source or parameter ID 5619 for voltage set point 2 source; refer to the Configuration Manual 37224 for detailed information). The respective voltage set point source is to be configured to 05.09 "Interface volt.setp.".

Two different methods to transmit a voltage set point via CANopen interface are detailed in the following. Refer to Comparison of the Two Methods on page 28 for the basic differences of these methods.

Transmitting a Voltage Set Point via RPDO

Configuration of CAN Interface 1

Be sure to enable CAN-Open Master if there is no PLC taking over the master function. Refer to Configuration of CAN Interface 1 on page 28 for the configuration of this parameter.

Configuration of the RPDO

Press [↱] until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the [↓] and [↑] softkeys and press [↵] to enter the "Receive PDO 1" screen.

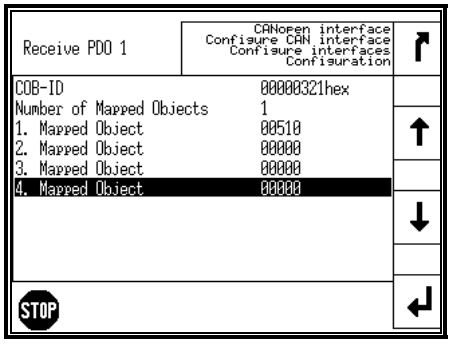

Figure 2-14: Display screen - Receive PDO 1 for voltage set point

Configure the following values for the "Receive PDO 1" parameters using the [↓] and [↑] as well as [→], [+] and [−] softkeys and Confirm the change by pressing the [↵] softkey:

COB-ID	00000321 (hex)
Number of Mapped Objects	1
1. Mapped Object	00510

Setting the COB-ID to 321 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameter 510 of the device as mapped object 1.

NOTE

Refer to Chapter 3.Appendix B: Additional Data Protocol Parameters starting on page 96 for a list of additional parameter groups.

CANopen Request

Figure 2-15 shows exemplary send data for the device on the CANopen bus in line 1. A voltage set point of 412 V is transmitted (412 (dec) = 019C (hex) -> 9C 01 according to the CANopen protocol).

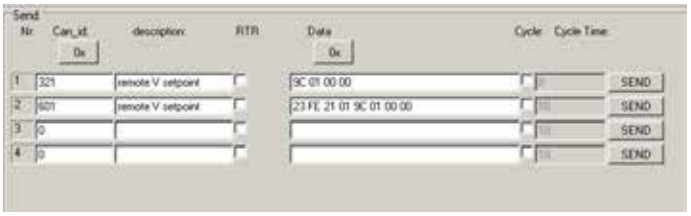

Figure 2-15: CANopen send data for voltage set point

Transmitting a Voltage Set Point via Default SDO Communication Channel

Another possibility for transmitting a voltage set point is to send the value via default SDO communication channel.
The device listens to the CAN ID 600 (hex) + Node ID internally to perform the desired control, the reply is on CAN ID 580 (hex) + Node ID.

The following example shows the send format on CANopen with Node ID 1.

The value is sent on the bus via the control parameter 510 of the device.
The hexadecimal value 2000 is calculated internally.
510(decimal) -- 1FE (hexadecimal)
1FE+2000 (hexadecimal) = 21FE

Please note that high and low byte are exchanged in the sent value.
The data (hex) shows the state of parameter 510 to achieve the required control.

Figure 2-15 shows exemplary send data for the device on the CANopen bus in line 2.

Transmitting a Power Factor Set Point via CANopen

It is possible to transmit a power factor set point value via the CANopen interface. Prerequisite for the use of a power factor set point via an interface is the configuration of the power factor set point source (parameter ID 5638 for power factor set point 1 source or parameter ID 5639 for power factor set point 2 source; refer to the Configuration Manual 37224 for detailed information). The respective power factor set point source is to be configured to 05.03 "Interface PF setp.".

Two different methods to transmit a power factor set point via CANopen interface are detailed in the following. Refer to Comparison of the Two Methods on page 28 for the basic differences of these methods.

Transmitting a Power Factor Set Point via RPDO

Configuration of CAN Interface 1

Be sure to enable CAN-Open Master if there is no PLC taking over the master function. Refer to Configuration of CAN Interface 1 on page 28 for the configuration of this parameter.

Configuration of the RPDO

Press ↱ until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the ↓ and ↑ softkeys and press ↵ to enter the "Receive PDO 1" screen.

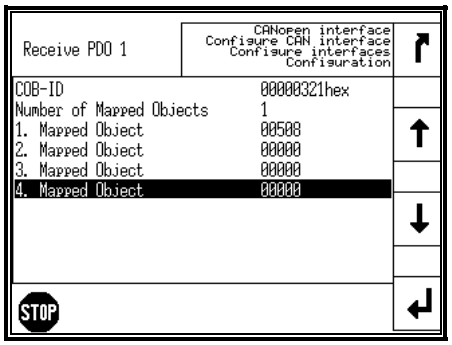

Figure 2-16: Display screen - Receive PDO 1 for power factor set point

Configure the following values for the "Receive PDO 1" parameters using the ↓ and ↑ as well as →, + and − softkeys and Confirm the change by pressing the ↵ softkey:

COB-ID	00000321 (hex)
Number of Mapped Objects	1
1. Mapped Object	00508

Setting the COB-ID to 321 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameter 508 of the device as mapped object 1.

NOTE

Refer to Chapter 3.Appendix B: Additional Data Protocol Parameters starting on page 96 for a list of additional parameter groups.

CANopen Request

Figure 2-17 shows exemplary send data for the device on the CANopen bus.
A power factor set point of 0.85 capacitive/leading is transmitted (64689 (dec) [65536-850] = FCAE (hex) -> AE FC according to the CANopen protocol) in line 1. Please note that negative (capacitive or leading) power factor values are deducted from 65536 (dec) or FFFF (hex).
A power factor set point of 0.9 inductive/lagging is transmitted (900 (dec) = 0384 (hex) -> 84 03 according to the CANopen protocol) in line 2.
A power factor set point of 1.0 is transmitted (1000 (dec) = 03E8 (hex) -> E8 03 according to the CANopen protocol) in line 3.

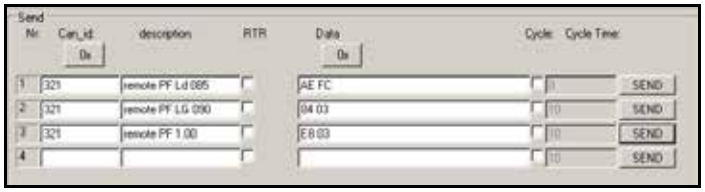

Figure 2-17: CANopen send data for power factor set point

Transmitting a Power Factor Set Point via Default SDO Communication Channel

Another possibility for transmitting a power factor set point is to send the value via default SDO communication channel.
The device listens to the CAN ID 600 (hex) + Node ID internally to perform the desired control, the reply is on CAN ID 580 (hex) + Node ID.

The following example shows the send format on CANopen with Node ID 1.

Thevalue is sent on the bus via the control parameter 508 of the device.
The hexadecimal value 2000 is calculated internally.
508(decimal) -- 1FC (hexadecimal)
1FC+2000 (hexadecimal) = 21FC

Please note that high and low byte are exchanged in the sent value.
The data (hex) shows the state of parameter 508 to achieve the required control.

Figure 2-18 shows exemplary send data for the device on the CANopen bus.

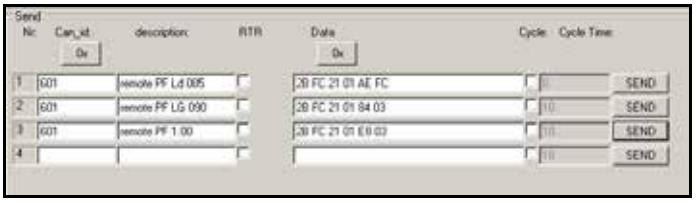

Figure 2-18: CANopen send data for Node ID 1 for power factor set point

Transmitting a Power Set Point via CANopen

It is possible to transmit a power set point value via the CANopen interface. Prerequisite for the use of a power set point via an interface is the configuration of the power set point source (parameter ID 5539 for power set point 1 source or parameter ID 5540 for power set point 2 source; refer to the Configuration Manual 37224 for detailed information). The respective power set point source is to be configured to 05.09 "Interface pow. setp.".
Please note that the type of the power set point (Constant, Import, or Export) must also be defined (parameter ID 5526 for load set point 1 or parameter ID 5527 for load set point 2).

Two different methods to transmit a voltage set point via CANopen interface are detailed in the following. Refer to Comparison of the Two Methods on page 28 for the basic differences of these methods.

Transmitting a Power Set Point via RPDO

Configuration of CAN Interface 1

Be sure to enable CAN-Open Master if there is no PLC taking over the master function. Refer to Configuration of CAN Interface 1 on page 28 for the configuration of this parameter.

Configuration of the RPDO

Press ↱ until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the ↓ and ↑ softkeys and press ↵ to enter the "Receive PDO 1" screen.

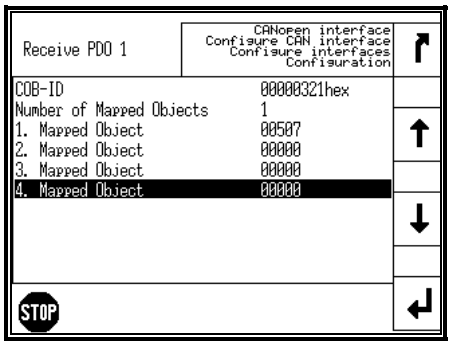

Figure 2-19: Display screen - Receive PDO 1 for power set point

Configure the following values for the "Receive PDO 1" parameters using the ↓ and ↑ as well as →, ✚ and ━ softkeys and Confirm the change by pressing the ↵ softkey:

COB-ID	00000321 (hex)
Number of Mapped Objects	1
1. Mapped Object	00507

Setting the COB-ID to 321 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameter 507 of the device as mapped object 1.

NOTE

Refer to Chapter 3.Appendix B: Additional Data Protocol Parameters starting on page 96 for a list of additional parameter groups.

CANopen Request

Figure 2-20 shows exemplary send data for the device on the CANopen bus in line 1. A power set point of 1000.0 kW is transmitted (10000 (dec) = 2710 (hex) -> 10 27 according to the CANopen protocol).

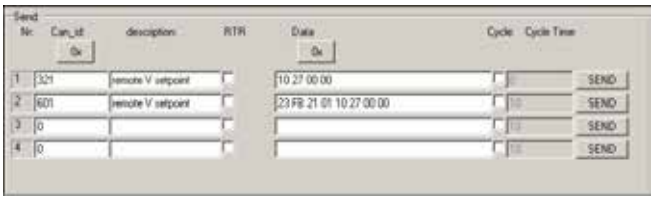

Figure 2-20: CANopen send data for power set point

Transmitting a Power Set Point via Default SDO Communication Channel

Another possibility for transmitting a power set point is to send the value via default SDO communication channel.
The device listens to the CAN ID 600 (hex) + Node ID internally to perform the desired control, the reply is on CAN ID 580 (hex) + Node ID.

The following example shows the send format on CANopen with Node ID 1.

The value is sent on the bus via the control parameter 507 of the device.
The hexadecimal value 2000 is calculated internally.
507(decimal) -- 1FB (hexadecimal)
1FB+2000 (hexadecimal) = 21FB

Please note that high and low byte are exchanged in the sent value.
The data (hex) shows the state of parameter 507 to achieve the required control.

Figure 2-20 shows exemplary send data for the device on the CANopen bus in line 2.

Transmitting Multiple Set Points via CANopen

It is possible to transmit multiple objects with one RPDO. The receive PDO can be used for four objects with 16 bytes. If larger objects (for example 32 bytes, like for voltage and power set points) are used, the maximum number of objects is reduced.

Configuration of the RPDO

Press ↱ until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the ↓ and ↑ softkeys and press ↵ to enter the "Receive PDO 1" screen.

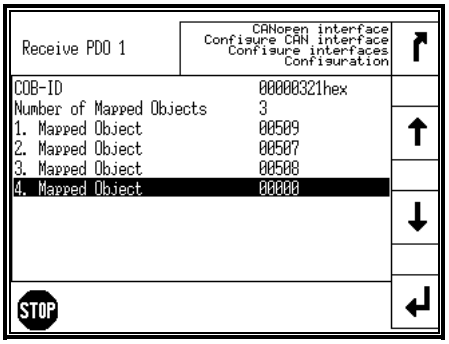

Figure 2-21: Display screen - Receive PDO 1 for multiple set points

Configure the following values for the "Receive PDO 1" parameters using the ↓ and ↑ as well as →, + and − softkeys and Confirm the change by pressing the ↵ softkey:

COB-ID	00000321 (hex)
Number of Mapped Objects	3
1. Mapped Object	00509
2. Mapped Object	00507
3. Mapped Object	00508

Setting the COB-ID to 321 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameters 509, 507, and 508 of the device as mapped object 1.

NOTE

Refer to Chapter 3.Appendix B: Additional Data Protocol Parameters starting on page 96 for a list of additional parameter groups.

CANopen Request

Figure 2-22 shows exemplary send data for the device on the CANopen bus in line 1. The following set points are transmitted:

- Frequency 50.6 Hz (5060 (dec) = 13C4 (hex) -> C4 13 according to the CANopen protocol)
- Power 1000 kW (10000 (dec) = 2710 (hex) -> 10 27 according to the CANopen protocol)
- Power factor 0.9 lagging (900 (dec) = 0384 (hex) -> 84 03 according to the CANopen protocol)

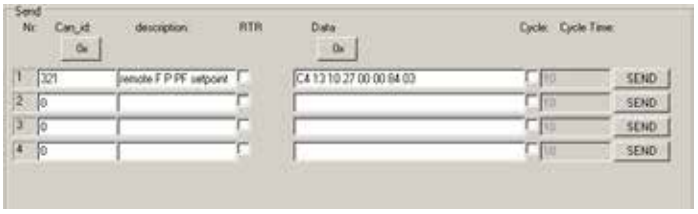

Figure 2-22: CANopen send data for multiple set points

Remotely Changing the Set Point via CANopen

It is possible to remotely change a set point value via the CANopen interface using the parameter ID 504 (refer to Remote Control Word 2 - Object 21F8h (Parameter ID 504) on page 97). In order to enable the set point 2 of the respective control function, different bits of parameter ID 504 must be enabled:

- [04.37] Remote voltage set point 2 bit 4 10 00 (hex) must be sent to parameter ID 504
- [04.38] Remote frequency set point 2 bit 5 20 00 (hex) must be sent to parameter ID 504
- [04.39] Remote Power Factor set point 2 bit 6 30 00 (hex) must be sent to parameter ID 504
- [04.40] Remote power set point 2 bit 7 80 00 (hex) must be sent to parameter ID 504

NOTE

For remotely changing the control set points, it is necessary to use the interface set points instead of the internal set points. In order to use the interface set points, the set point source parameters (for example "Frequency setpoint 1 source" or "Frequency setpoint 2 source") of the controller must be configured to the respective interface set points as data source (refer to the Configure Controller section in the Parameters chapter of the configuration manual 37224 for detailed information).

Two different methods for changing a set point via CANopen interface are detailed in the following. Refer to Comparison of the Two Methods on page 28 for the basic differences of these methods.

Changing a Set Point via RPDO

Configuration of CAN Interface 1

Be sure to enable CAN-Open Master if there is no PLC taking over the master function. Refer to Configuration of CAN Interface 1 on page 28 for the configuration of this parameter.

Configuration of the RPDO

Press [↱] until you return to the start screen.
Navigate to the "Configure CAN interface 1" screen by pressing the following softkeys in this sequence:
Parameter -> Configuration -> Configure interfaces -> Configure CAN interface -> Configure CAN interface 1

Navigate to the entry "Receive PDO 1" by using the [↓] and [↑] softkeys and press [↵] to enter the "Receive PDO 1" screen.

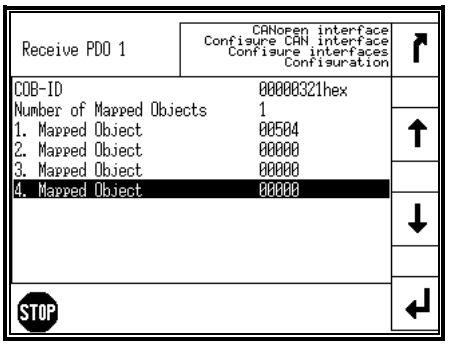

Figure 2-23: Display screen - Receive PDO 1 for changing the set point

Configure the following values for the "Receive PDO 1" parameters using the [↓] and [↑] as well as [→], [+] and [−] softkeys and Confirm the change by pressing the [↵] softkey:

COB-ID	00000321 (hex)
Number of Mapped Objects	1
1. Mapped Object	00504

Setting the COB-ID to 321 is exemplary; usually, the PDOs are in the range of 181 (hex) to 57F (hex).
With this setting, the Receive PDO is set to the address, for which the device is listening on the bus. The number of mapped objects is 1 since 1 mapped object is used. The request on the bus is sent with the control parameter 504 of the device as mapped object 1.

CANopen Request

Figure 2-24 shows exemplary send data for the device on the CANopen bus. The respective bits are enabled by sending the data of the respective lines.

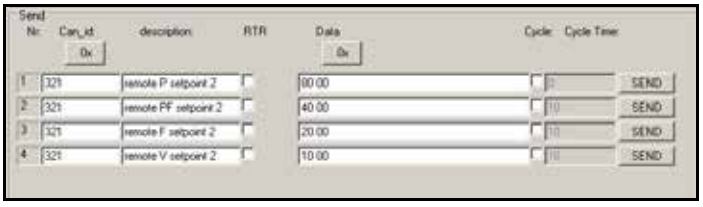

Figure 2-24: CANopen send data for changing the set point

Changing a Set Point via Default SDO Communication Channel

Another possibility for changing a set point is to enable the bit via default SDO communication channel.
The device listens to the CAN ID 600 (hex) + Node ID internally to perform the desired control, the reply is on CAN ID 580 (hex) + Node ID.

The following example shows the send format on CANopen with Node ID 1.

The value is sent on the bus via the control parameter 504 of the device.
The hexadecimal value 2000 is calculated internally.
509(decimal) -- 1F8 (hexadecimal)
1F8+2000 (hexadecimal) = 21F8

Please note that high and low byte are exchanged in the sent value.
The data (hex) shows the state of parameter 504 to achieve the required control.

Figure 2-25 shows exemplary send data for the device on the CANopen bus.

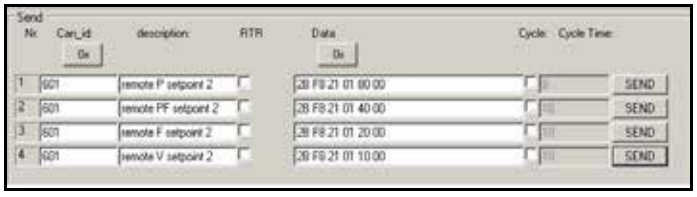

Figure 2-25: CANopen send data for Node ID 1 for changing the set point

Sending a Data Protocol via TPDO

Cyclically Sending of Data

This is a configuration example for sending an object with the index 16382 (data protocol 5003) on CAN ID 2AEh every 20 ms on TPDO1. For this, TPDO1 must be configured as follows:

COB-ID	2AE (hex)	
Transmission type	255	
Event-timer	20 ms	
Number of Mapped Objects	1	(there is only one object to be transmitted)
1. Mapped Object	16382	(display value, the object with the index 16382)
2. Mapped Object	0	(will not be used)
3. Mapped Object	0	(will not be used)
4. Mapped Object	0	(will not be used)

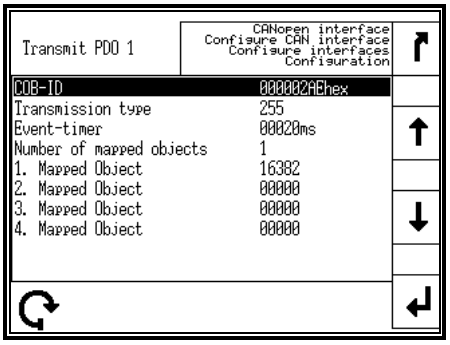

Figure 2-26: Cyclical sending of data - TPDO configuration

Sending of Data on Request

The data to be sent (Mapped Objects) may be provided on request by configuring the Sync Message (Parameter 9100) and the Transmission Type (Parameter 9602, 9612, 9622, or 9632) of a TPDO.
The unit is requested to send its data by sending a Sync Message.

The number of required Sync Messages is determined by the setting of the Transmission Type.

If the data is to be sent on request, Bit 31 of the Sync Message (Parameter 9100) must be configured to "1" and the CANopen Master (Parameter 8993) function must be configured to "Off".

The Transmission Type of TPDO 1 is configured to "2" in the following example (refer to Figure 2-27).
This means that a message of the configured TPDO is sent by the unit after two Sync Messages have been sent to the unit.

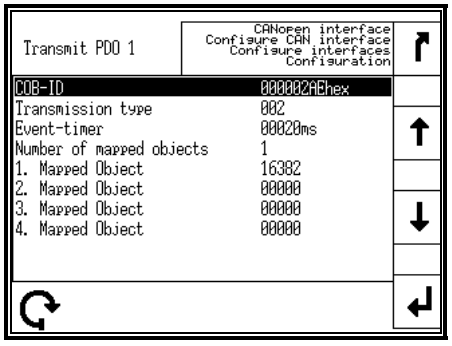

Figure 2-27: Sending of data on request - TPDO configuration

The recorded data shows that the data of the Mapped Object (in this example Mux 5) is sent (refer to Figure 2-29) after sending the Sync Message twice (refer to Figure 2-28).

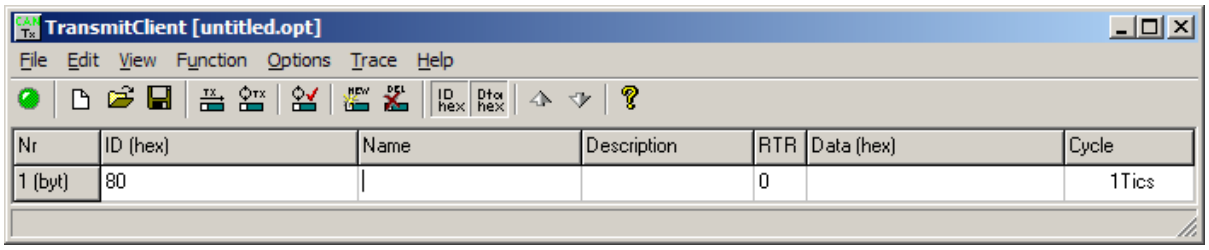

Figure 2-28: Cyclical sending of data - Sync Message request

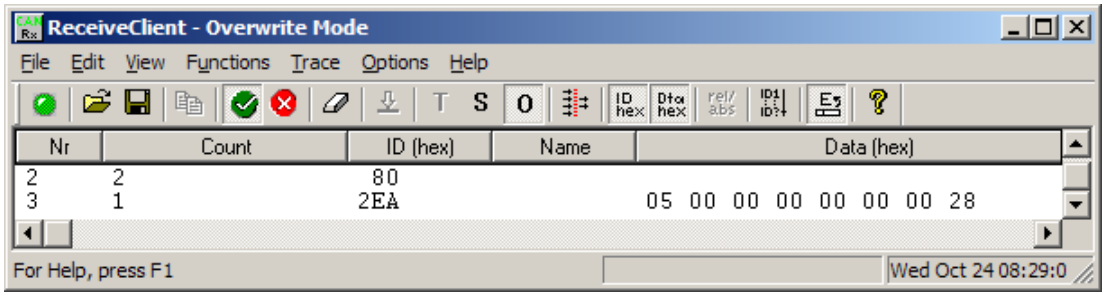

Figure 2-29: Cyclical sending of data - reply

External DOs for an IKD 1

This is a configuration example for sending objects with the index 8001 and 3 x 8000 on CAN ID 181h every 20 ms on TPDO1. This is used to send messages to an external device. For this, TPDO1 must be configured as follows:

COB-ID	181 (hex)	
Transmission type	255	
Event-timer	20 ms	
Number of Mapped Objects	4	(there are four objects to be transmitted)
1. Mapped Object	8001	(display value, the object with the index 8001)
2. Mapped Object	8000	(display value, the object with the index 8000)
3. Mapped Object	8000	(display value, the object with the index 8000)
4. Mapped Object	8000	(display value, the object with the index 8000)

NOTE

This is an example of how to configure an IKD 1 on CAN interface 1. Usually, IKDs may be configured to CAN interface 2 much easier.

Receiving Data from an IKD 1

This is a configuration example for an RPDO configuration. The data received on CAN ID 201h is interpreted as object with the index 8011 (external DIs 1 to 8). For this, RPDO must be configured as follows:

COB-ID	201 (hex)	
Number of Mapped Objects	1	(there is only one object to be received)
1. Mapped Object	8011	(display value, the object with the index 8011)
2. Mapped Object	0	(will not be used)
3. Mapped Object	0	(will not be used)
4. Mapped Object	0	(will not be used)

NOTE

This is an example of how to configure an IKD 1 on CAN interface 1. Usually, IKDs may be configured to CAN interface 2 much easier.

Chapter 3.
Modbus Communications

General Information

Modbus is a serial communications protocol published by Modicon in 1979 for use with its programmable logic controllers (PLCs). It has become a de facto standard communications protocol in industry, and is now the most commonly available means of connecting industrial electronic devices. The easYgen-3000 supports a Modbus RTU Slave module. This means that a Master node needs to poll the easYgen slave node. Modbus RTU can also be multi-dropped, or in other words, multiple Slave devices can exist on one Modbus RTU network, assuming that the serial interface is a RS-485. Detailed Information about the Modbus protocol are available on the following website:
http://www.modbus.org/specs.php

There are also various tools available on the internet. We recommend to use ModScan32 which is a Windows application designed to operate as a Modbus Master device for accessing data points in a connected Modbus Slave device. It is designed primarily as a testing device for verification of correct protocol operation in new or existing systems. It is possible to download a trial version from the following website:
http://www.win-tech.com/html/modscan32.htm

Address Range

The easYgen Modbus Slave module distinguishes between visualization data and configuration & remote control data. The different data is accessible over a split address range and can be read via the "Read Holding Register" function. Furthermore, easYgen parameters and remote control data can be written with the "Preset Single Registers" function or "Preset Multiple Registers" (refer to Table 3-1).

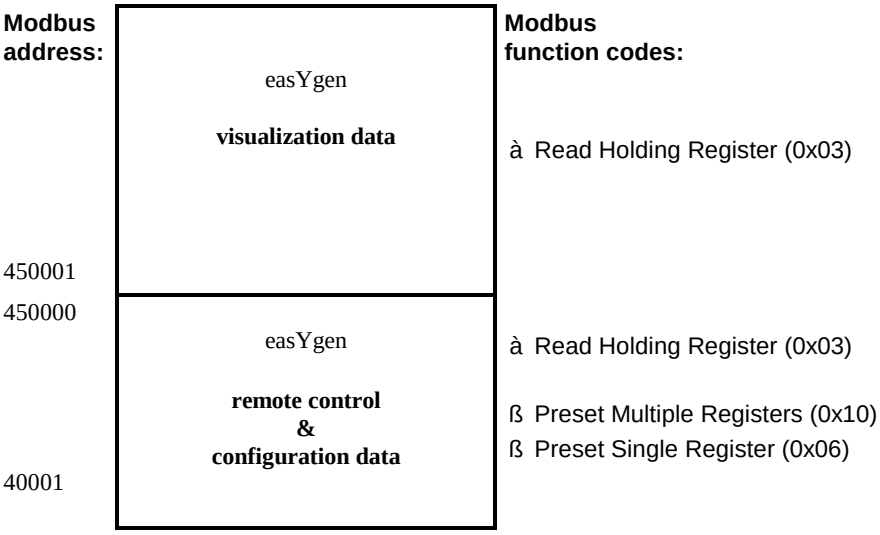

Table 3-1: Modbus - address range

NOTE

All addresses in this document comply with the Modicon address convention. Some PLCs or PC programs use different address conventions depending on their implementation. Then the address must be increased and the leading 4 may be omitted.

Please refer to your PLC or program manual for more information. This determines the address sent over the bus in the Modbus telegram. The Modbus starting address 450001 of the visualization data may become bus address 50000 for example.

Visualization

The visualization over Modbus is provided in a very fast data protocol where important system data like alarm states, AC measurement data, switch states and various other information may be polled. According to the easYgen Modbus addressing range, the visualization protocol can be reached on addresses starting at 450001. On this address range it is possible to do block reads from 1 up to 128 Modbus registers at a time.

Modbus Read Addresses	Description	Multiplier	Units
450001	Protocol-ID, always 5003		--
450002	Pickup speed	1	rpm
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
450268	Exhaust Gas Temp.	0.01	°C

Table 3-2: Modbus - address range block read

NOTE

Table 3-2 is only an excerpt of the data protocol. It conforms to the data protocol 5003 that is also used by CAN bus. Refer to Appendix B: Data Protocol 5003 on page 79 for the complete protocol.

The following ModScan32 screenshot shows the configurations made to read the visualization protocol with a block read of 128 registers.

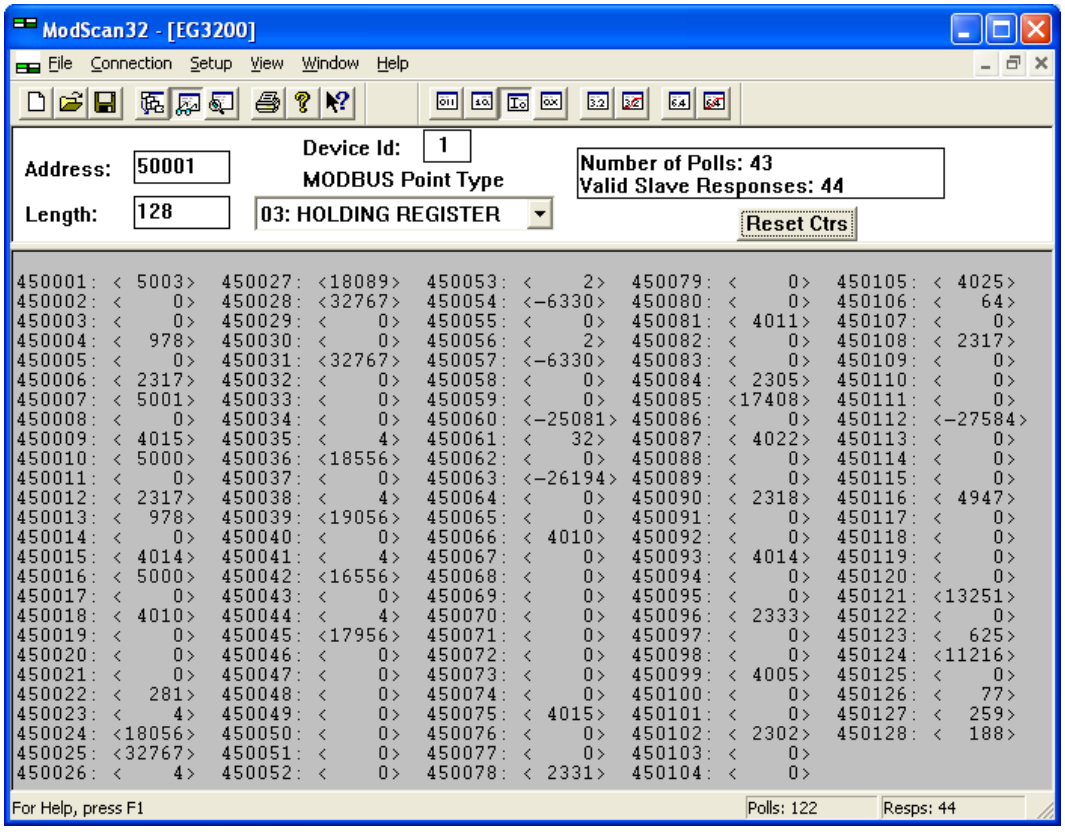

Figure 3-1: Modbus - visualization configurations

Configuration

The Modbus interface can be used to read/write parameters of the easYgen. According to the easYgen Modbus addressing range for the configuration addresses, the range starts at 40001 and ends at 450000. You can always access only one parameter of the system in this address range. The Modbus address can be calculated depending on the parameter ID as illustrated below:

	Parameter ID < 10000	Parameter ID >= 10000
Modbus address =	40000 + (Par. ID+1)	400000 + (Par. ID+1)

Table 3-3: Modbus - address calculation

Block reads in this address range depend on the data type of the parameter. This makes it important to set the correct length in Modbus registers which depends on the data type (UNSIGNED 8, INTEGER 16, etc.). Refer to Table 3-4 for more information.

easYgen types	Modbus registers
UNSIGNED 8	1
UNSIGNED 16	1
INTEGER 16	1
UNSIGNED 32	2
INTEGER 32	2
LOGMAN	7
TEXT/X	X/2

Table 3-4: Modbus - data types

Parameter Setting

NOTE

The example tables below are excerpts of the parameter list in the Configuration Manual 37224. Please refer to this manual for the complete parameter list.

NOTE

Be sure to enter the password for code level 2 or higher for the corresponding interface to get access for changing parameter settings.

NOTE

The new entered value must comply with the parameter setting range when changing the parameter setting.

Example 1: Addressing the password for serial interface1:

Par. ID.	Parameter	Setting range	Data type
10401	Password for serial interface1	0000 to 9999	UNSIGNED 16

Modbus address = 400000 + (Par. ID + 1) = 410402
Modbus length = 1 (UNSIGNED 16)

The following Modscan32 screenshot shows the configurations made to address parameter 10401.

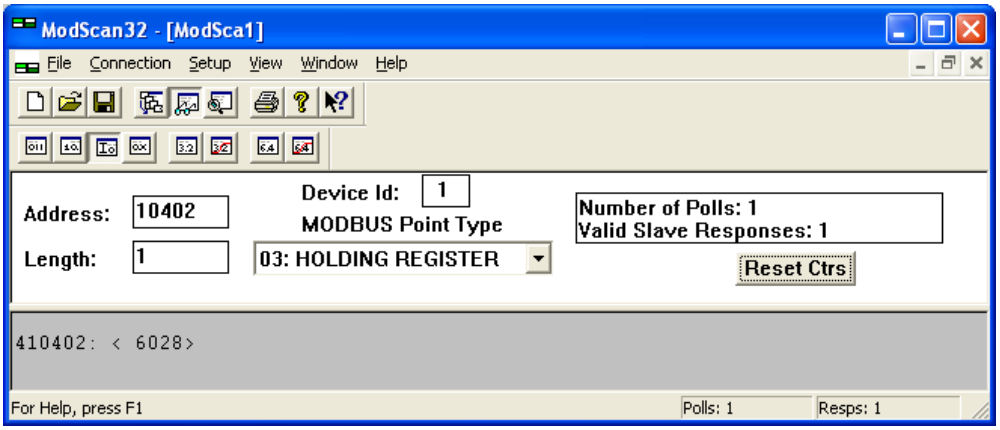

Figure 3-2: Modbus - configuration example 1

Example 2: Addressing the generator rated voltage:

Par. ID.	Parameter	Setting range	Data type
1766	Generator rated voltage	50 to 650000 V	UNSIGNED 32

Modbus address = 40000 + (Par. ID + 1) = 41767
Modbus length = 2 (UNSIGNED 32)

The following Modscan32 screenshot shows the configurations made to address parameter 1766.

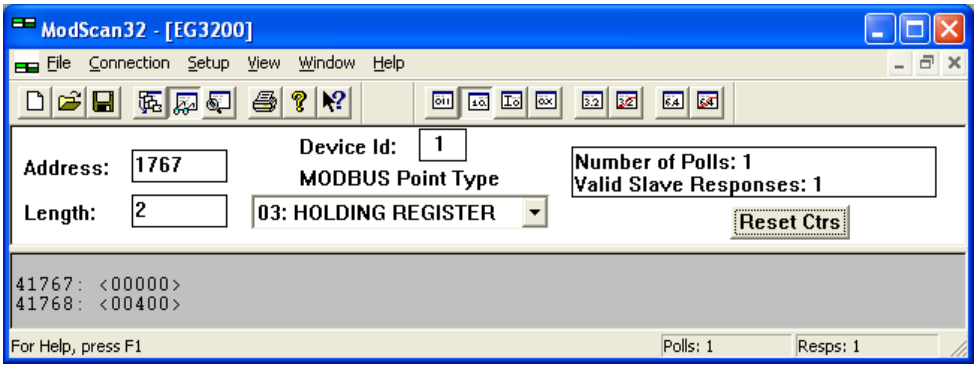

Figure 3-3: Modbus - configuration example 2

NOTE

The parameters of above examples are an excerpt of the parameter list in the appendix of the Configuration Manual 37224. Refer to this manual for the complete parameter list.

Example 3: Addressing the generator voltage measuring:

Par. ID.	Parameter	Setting range	Data type
1851	Generator voltage measuring	3Ph 4W {0} 3Ph 3W {1} 1Ph 2W {2} 1Ph 3W {3}	UNSIGNED 16

Modbus address = 40000 + (Par. ID + 1) = 41852
Modbus length = 1 (UNSIGNED 16)

NOTE

If the setting range contains a list of parameter settings like in this example, the parameter settings are numbered and start with 0 for the first parameter setting. The number corresponding with the respective parameter setting must be configured.

The following Modscan32 screenshot shows the configurations made to address parameter 1851, which is configured to "3Ph 4W".

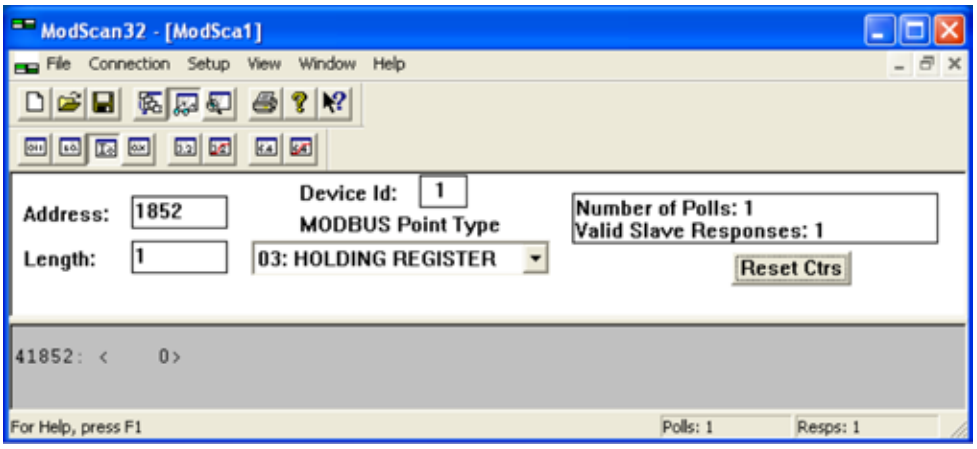

Figure 3-4: Modbus - configuration example 3

Set Point Setting

For a remote setting of the control set points, it is necessary to use the interface set points instead of the internal set points. In order to use the interface set points, the set point source parameters of the controller must be configured to the respective interface set points (refer to the Configure Controller section in the Parameters chapter of the configuration manual 37224 for detailed information). Figure 3-5 shows an exemplary configuration of the load set point 1 source. All other set point sources are configured accordingly.

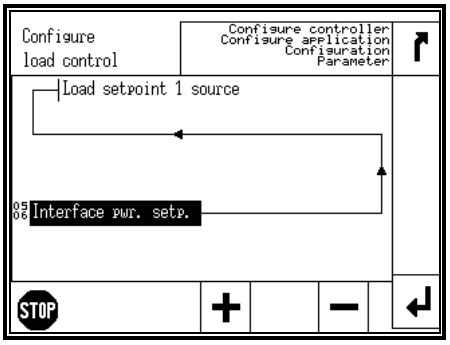

Figure 3-5: Set point source configuration

The interface set points may be set using the objects for active power, power factor, frequency, and voltage (refer to Additional Data Protocol Parameters on page 96 for detailed information).

Par. ID.	Parameter	Setting range	Unit	Data type
507	Active Power Setpoint	0 to 999999	1/10 [kW]	INTEGER 32
508	Power Factor Setpoint	-710 to 1000 to 710	-	INTEGER 16
509	Frequency Setpoint	0 to 7000	1/100 [Hz]	UNSIGNED 16
510	Voltage Setpoint	50 to 650000	[V]	UNSIGNED 32

Example 1: Active Power Interface Set Point

The active power set point value must be written to object 21FB (hex), i.e. parameter ID 507

Example: A power value of 50 kw = 500 (dec) =01F4 (hex) is to be transmitted.
Modbus address = 40000 + (Par. ID + 1) = 40508
Modbus length = 2 (INTEGER 32)
The high word is to be written to the lower address and the low word is to be written to the higher address.

Figure 3-6 through Figure 3-9 show how to set the parameter address 507 in ModScan32.

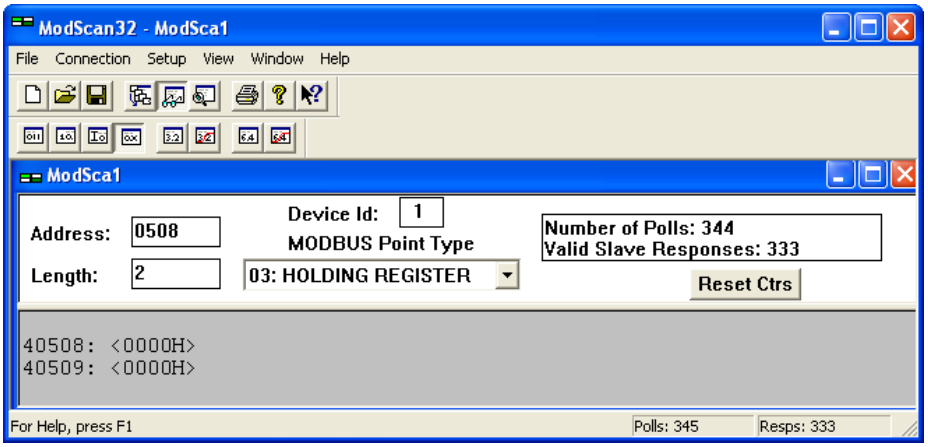

Figure 3-6: Modbus - configuration example 4 - active power

Open the Preset Multiple registers window by selecting Setup -> Extended -> Preset Regs from the menu.

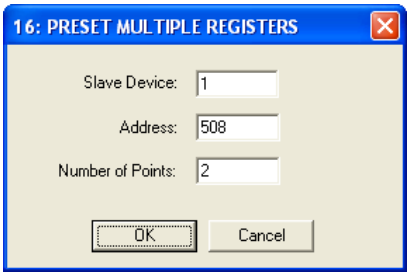

Figure 3-7: Modbus - configuration example 4 - active power

Select OK and enter the desired values.

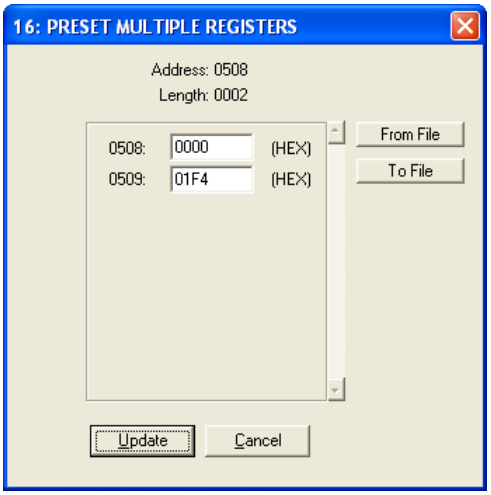

Figure 3-8: Modbus - configuration example 4 - active power

Select Update to take over the entered values.

Figure 3-9: Modbus - configuration example 4 - active power

Example 2: Power Factor Interface Set Point

The power factor set point value must be written to object 21FC (hex), i.e. parameter ID 508

Example: A power factor of 1 = 1000 (dec) =03E8 (hex) is to be transmitted.
Modbus address = 40000 + (Par. ID + 1) = 40509
Modbus length = 1 (UNSIGNED 16)

Figure 3-10 shows the settings made to parameter address 508 in ModScan32.

Figure 3-10: Modbus - configuration example 4 - power factor

Example 3: Frequency Interface Set Point

The frequency set point value must be written to object 21FD (hex), i.e. parameter ID 509

Example: A frequency value of 50.00 Hz = 5000 (dec) =1388 (hex) is to be transmitted.
Modbus address = 40000 + (Par. ID + 1) = 40510
Modbus length = 1 (UNSIGNED 16)

Figure 3-11 shows the settings made to parameter address 509 in ModScan32.

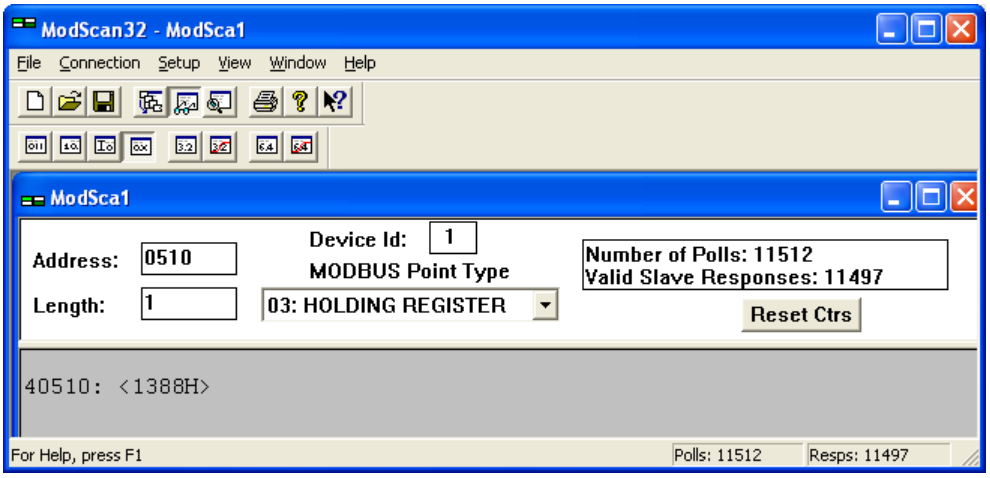

Figure 3-11: Modbus - configuration example 4 - frequency

Example 4: Voltage Interface Set Point

The voltage set point value must be written to object 21FE (hex), i.e. parameter ID 510

Example: A voltage value of 400 V = 400 (dec) =0190 (hex) is to be transmitted.
Modbus address = 40000 + (Par. ID + 1) = 40511
Modbus length = 2 (UNSIGNED 32)
The high word is to be written to the lower address and the low word is to be written to the higher address.

Figure 3-12 shows the settings made to parameter address 510 in ModScan32.

Figure 3-12: Modbus - configuration example 4 - voltage

Configuration of the *LogicsManager* Functions via Modbus

Besides HMI and ToolKit, it is also possible to configure the *LogicsManager* functions via modbus.

Used *LogicsManager* Functions

The following *LogicsManager* functions are used for remote access:

12120 Start req. in AUTO: this *LogicsManager* function is used for remote request start/stop

12490 Ext. acknowledge: this *LogicsManager* function is used for remote acknowledge

12540 Start w/o load: this *LogicsManager* functionr is used for start without load

12510 Operat. mode AUTO: this *LogicsManager* function is used for AUTOMATIC mode

Modbus Encoding of a *LogicsManager* Function

The following section describes how to configure a *LogicsManager* function via Modbus. A *LogicsManager* function is defined by several parameters, like delays, commands, signs, or operators.

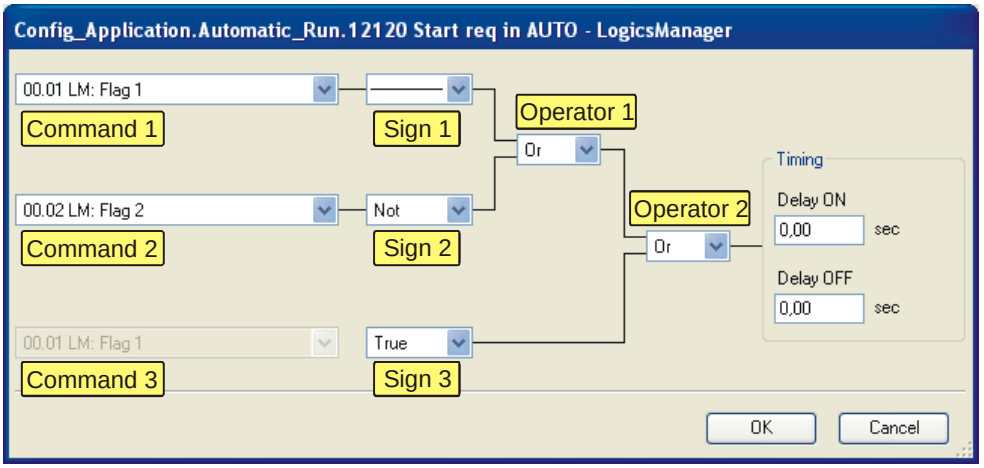

Figure 3-13: LogicsManager - Modbus encoding

The definition for a *LogicsManager* function consists of 7 data words:

Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Delay ON	Delay OFF	Logic equation 1	Logic equation 2	Command 1	Command 2	Command 3

NOTE

The configuration of a *LogicsManager* function via Modbus requires to reverse the byte order of double-byte words, i.e. low byte before high byte.

The delays are configured as hundredths of a second, i.e. 500 must be configured for a delay of 5 seconds.

The logic equation (0xFFFF) contains the information of one operator in each nibble.

Logic equation 1:

Word 2			
High byte		Low byte	
Highest nibble	Second highest nibble	Third highest nibble	Lowest nibble
Sign 2	Operator 2	Sign 1	Operator 1

Logic equation 2:

Word 3			
High byte		Low byte	
Highest nibble	Second highest nibble	Third highest nibble	Lowest nibble
not used	not used	Sign 3	not used

Definition of the nibbles:

Signs:

0x00	negate value of this element with 'NOT'
0x10	keep value of this element with '—'
0x20	force value of this element to 'TRUE'
0x30	force value of this element to 'FALSE'

Operators:

0x00	'AND' with following element
0x01	'NAND' with following element
0x02	'OR' with following element
0x03	'NOR' with following element
0x04	'XOR' with following element
0x05	'NOT-XOR' with following element

The commands are defined by configuring the ID of the respective command variable. Refer to the Logical Command Variables section of the *LogicsManager* appendix of the configuration manual 37224 for the command variable IDs.

Remote Control via Modbus

The easYgen-3000 controller may be configured to perform start/stop/acknowledgement functions remotely through the Modbus. The required procedure is detailed in the following steps. Please also refer to the Application Manual 37226.

Remote Control via Modbus

The easYgen controller may be configured to perform start/stop/acknowledgement functions remotely through the Modbus. The required procedure is detailed in the following steps.

NOTE

The following descriptions refer to the remote control parameter 503 as described under Remote Control Word 1 - Object 21F7h (Parameter ID 503) on page 96.

It may be necessary to shift the address by 1 depending on the used PC software. In this case, the address would be 504 for example.

Be sure to check both possibilities in case of remote control problems.

Par. ID.	Parameter	Setting range	Data type
503	Remote control word 1	0 to 65535	UNSIGNED 16

Modbus address = 40000 + (Par. ID +1) = 40504
Modbus length = 1 (UNSIGNED 16)

The following Modscan32 screenshot shows the configurations made to remote control parameter ID 503. It is possible to set the format to binary to view single bits using the "display options".

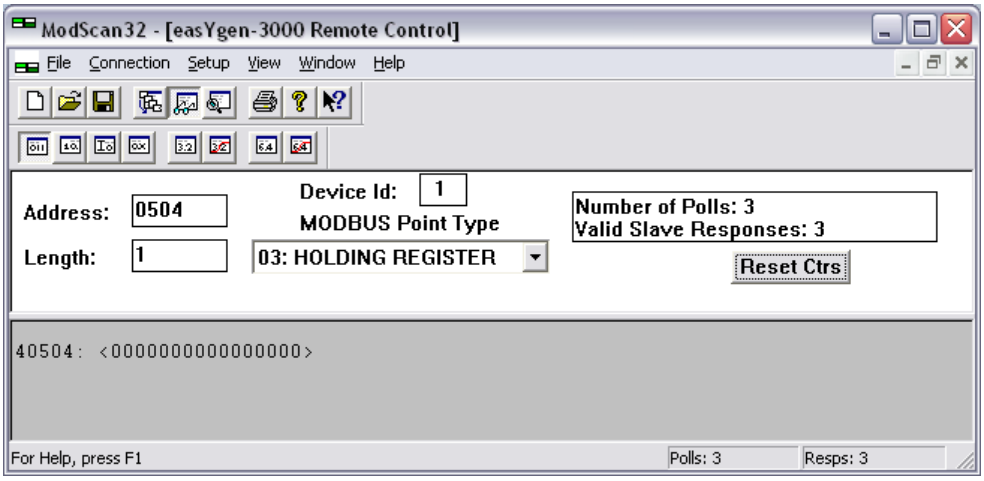

Figure 3-14: Modbus - remote control parameter 503

Example 1: Stop Request

By double-clicking the address, a Write Register command may be issued. Figure 3-15 shows how bit 0 is set using the ModScan32 Software.

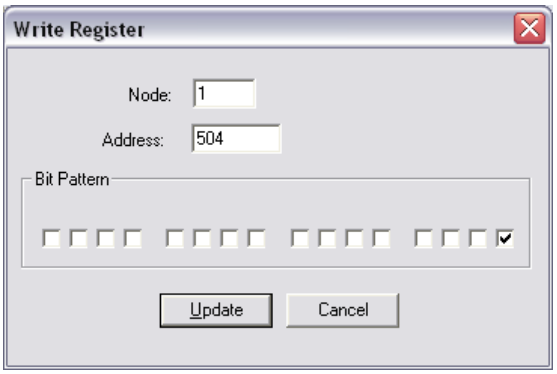

Figure 3-15: Modbus - write register - stop request

Example 2: Start Request

By double-clicking the address, a Write Register command may be issued. Figure 3-15 shows how bit 1 is set using the ModScan32 Software.

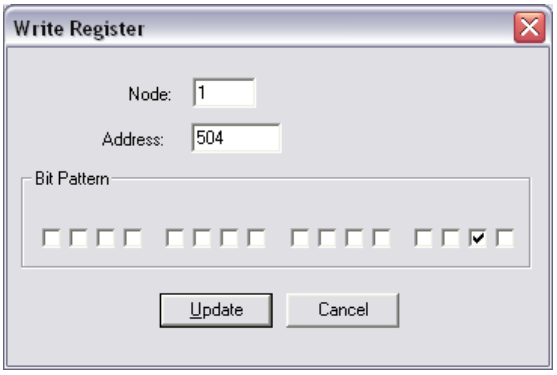

Figure 3-16: Modbus - write register - start request

Example 3: External Acknowledge

By double-clicking the address, a Write Register command may be issued. Figure 3-15 shows how bit 4 is set using the ModScan32 Software.

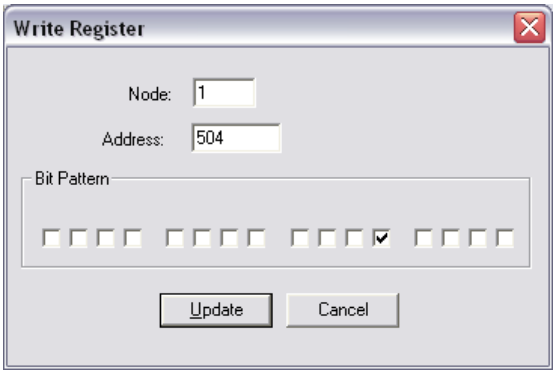

Figure 3-17: Modbus - write register - external acknowledge

Remotely Changing the Set Point

Active Power Set Point

It is possible to remotely change the active power set point through the Modbus using the parameter ID 504 (refer to Remote Control Word 2 - Object 21F8h (Parameter ID 504) on page 97). The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
504	Remote control word 2	YES / NO	UNSIGNED 16

In order to enable the active power set point 2, bit 7 of object 21F8 (hex), i.e. parameter ID 504, must be enabled.

Example:

The active power set point 2 is to be enabled.
Modbus address = 40000 + (Par. ID + 1) = 40505
Modbus length = 1 (UNSIGNED 16)

Figure 3-18 shows the settings made to parameter ID 504 in ModScan32. It is possible to set the format to binary to view single bits using the "display options".

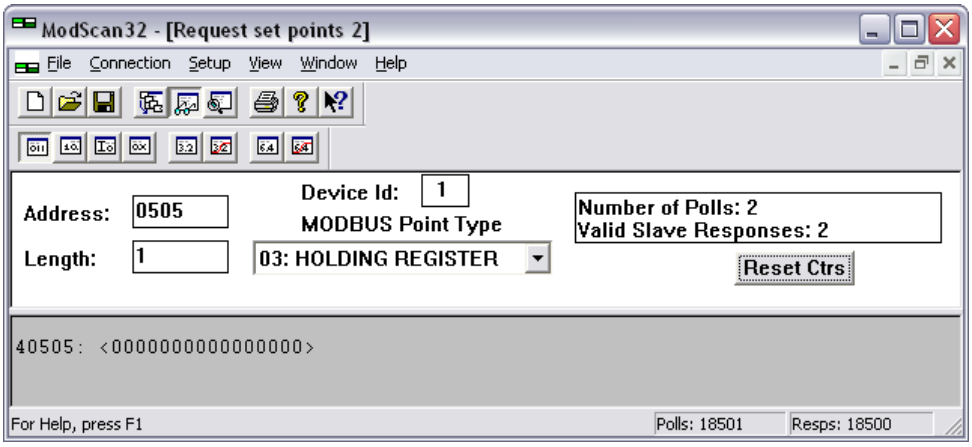

Figure 3-18: Modbus - remote control parameter 504

By double-clicking the address, a Write Register command may be issued. Figure 3-19 shows how bit 7 is set using the ModScan32 Software.

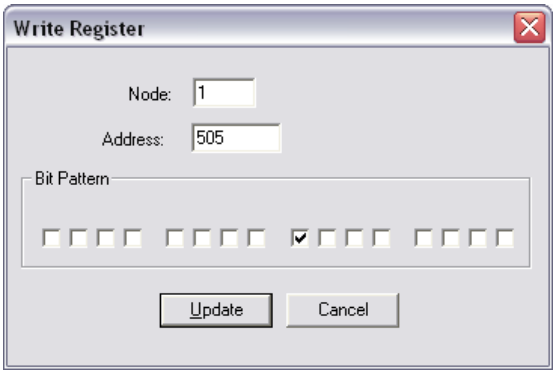

Figure 3-19: Modbus - write register - enable active power set point 2

Power Factor Set Point

It is possible to remotely change the power factor set point through the Modbus using the parameter ID 504 (refer to Remote Control Word 2 - Object 21F8h (Parameter ID 504) on page 97). The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
504	Remote control word 2	YES / NO	UNSIGNED 16

In order to enable the power factor set point 2, bit 6 of object 21F8 (hex), i.e. parameter ID 504, must be enabled.

Example:

The power factor set point 2 is to be enabled.
Modbus address = 40000 + (Par. ID + 1) = 40505
Modbus length = 1 (UNSIGNED 16)

Figure 3-20 shows the settings made to parameter ID 504 in ModScan32. It is possible to set the format to binary to view single bits using the "display options".

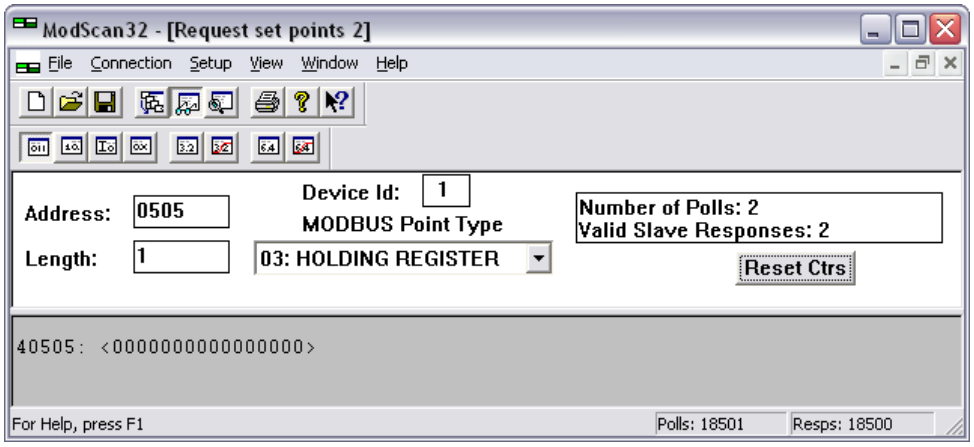

Figure 3-20: Modbus - remote control parameter 504

By double-clicking the address, a Write Register command may be issued. Figure 3-21 shows how bit 6 is set using the ModScan32 Software.

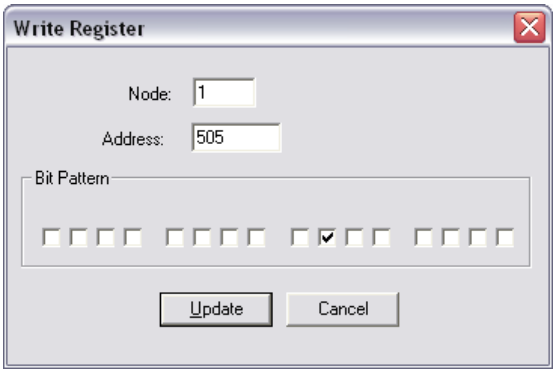

Figure 3-21: Modbus - write register - enable power factor set point 2

Frequency Set Point

It is possible to remotely change the frequency set point through the Modbus using the parameter ID 504 (refer to Remote Control Word 2 - Object 21F8h (Parameter ID 504) on page 97). The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
504	Remote control word 2	YES / NO	UNSIGNED 16

In order to enable the frequency set point 2, bit 5 of object 21F8 (hex), i.e. parameter ID 504, must be enabled.

Example:

The frequency set point 2 is to be enabled.
Modbus address = 40000 + (Par. ID + 1) = 40505
Modbus length = 1 (UNSIGNED 16)

Figure 3-22 shows the settings made to parameter ID 504 in ModScan32. It is possible to set the format to binary to view single bits using the "display options".

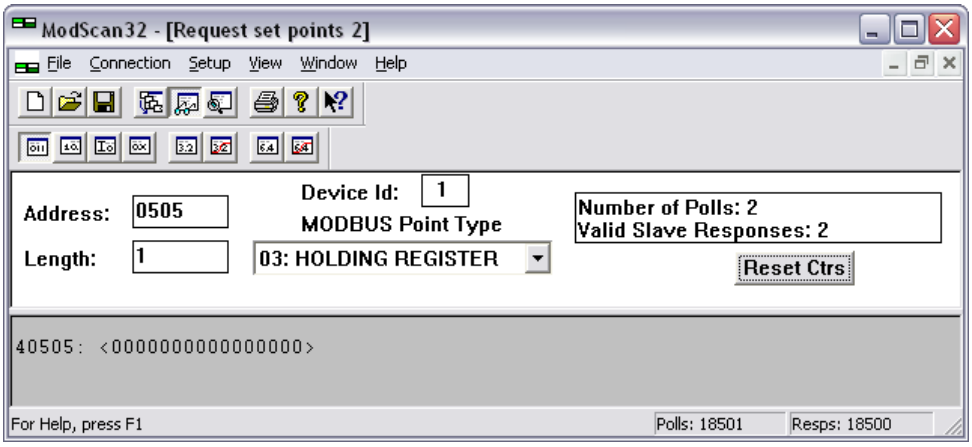

Figure 3-22: Modbus - remote control parameter 504

By double-clicking the address, a Write Register command may be issued. Figure 3-23 shows how bit 5 is set using the ModScan32 Software.

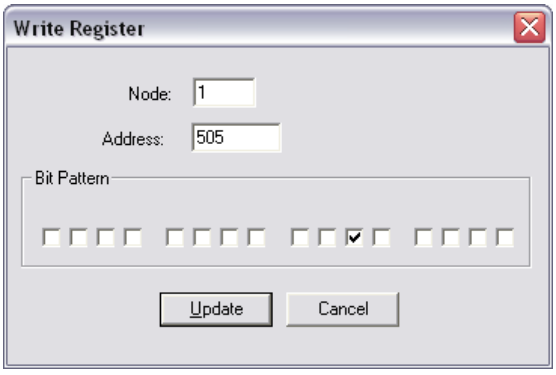

Figure 3-23: Modbus - write register - enable frequency set point 2

Voltage Set Point

It is possible to remotely change the voltage set point through the Modbus using the parameter ID 504 (refer to Remote Control Word 2 - Object 21F8h (Parameter ID 504) on page 97). The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
504	Remote control word 2	YES / NO	UNSIGNED 16

In order to enable the voltage set point 2, bit 4 of object 21F8 (hex), i.e. parameter ID 504, must be enabled.

Example:

The voltage set point 2 is to be enabled.
Modbus address = 40000 + (Par. ID + 1) = 40505
Modbus length = 1 (UNSIGNED 16)

Figure 3-22 shows the settings made to parameter ID 504 in ModScan32. It is possible to set the format to binary to view single bits using the "display options".

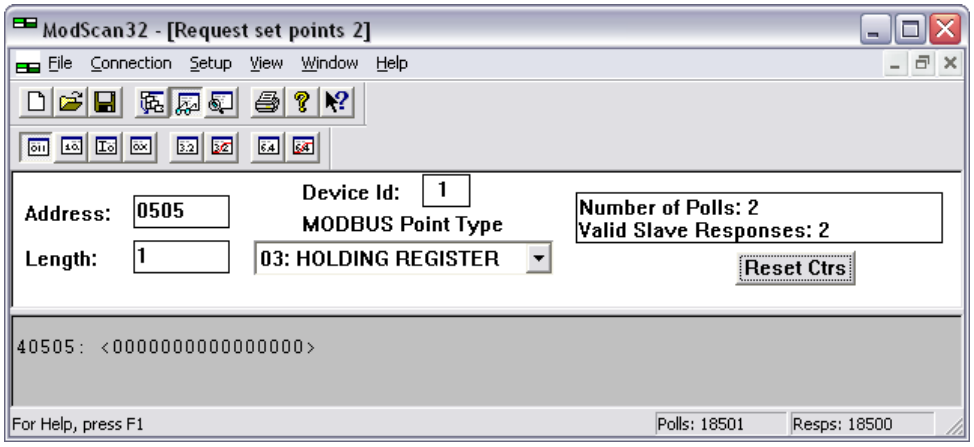

Figure 3-24: Modbus - remote control parameter 504

By double-clicking the address, a Write Register command may be issued. Figure 3-23 shows how bit 4 is set using the ModScan32 Software.

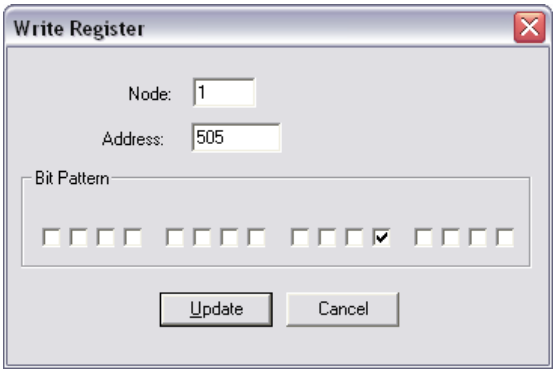

Figure 3-25: Modbus - write register - enable voltage set point 2

Changing Parameter Settings via Modbus

NOTE

Be sure to enter the password for code level 2 or higher for the corresponding interface to get access for changing parameter settings.

Operating Modes

Two operating modes may be used with remote control:

1. STOP
2. AUTOMATIC

It is possible to fix the operating mode using the *LogicsManager* function 00.16 "Operat. mode AUTO" (parameter ID 12510).

Configuration of the LogicsManager Operation Mode AUTO

The Operat. mode AUTO *LogicsManager* function (parameter ID 12510) can be configured in two different ways:

1. Automatic operating mode is always enabled
2. Automatic operating mode is enabled via discrete input

Refer to the Performing Remote Start/Stop and Acknowledgement section of the Application Manual 37226 for a detailed configuration of the *LogicsManager* via HMI or ToolKit.

Example:

The following Modbus message must be sent to the easYgen to configure the *LogicsManager* function accordingly.

Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Delay ON	Delay OFF	Logic equation 1	Logic equation 2	Command 1	Command 2	Command 3
0000 (hex)	0000 (hex)	1020 (hex)	2000 (hex)	0F02 (hex)	0000 (hex)	0000 (hex)

The detailed composition of this message is as follows:

Delay ON = 0.00 s -> word 0 = 0000 (hex)
Delay OFF = 0.00 s -> word 1 = 0000 (hex)
Logic eq. 1: sign 1 = '—'; Operator 1 = 'AND'; Sign 2 = 'TRUE', Operator 2 = 'AND' -> word 2 = 1020 (hex)
Logic equation 2: sign 3 = 'TRUE' -> word 3 = 2000 (hex)
Command 1 = 09.09 Discrete Input 9 = 0 (dec) = 0000 (hex) -> word 4 = 0F02 (hex)
Command 2 = 00.01 Flag 1 (default) = 0 (dec) = 0000 (hex) -> word 5 = 0000 (hex)
Command 3 = 00.01 Flag 1 (default) = 0 (dec) = 0000 (hex) -> word 6 = 0000 (hex)

The complete message of 7 words must be copied to address 12511 ff (12510+1) in one step. This is shown in Figure 3-26 using the ModScan32 software.

Figure 3-26: Modbus configuration - Operat. mode AUTO

NOTE

If an shutdown alarm of alarm class C through F occurs in AUTOMATIC operating mode, the control does not return to STOP operating mode if the alarm is cleared after acknowledgement. This means that a restart is initiated.

Remote Start/Stop and Acknowledgement

Refer to the Performing Remote Start/Stop and Acknowledgement section in the Special Application Examples section of the application manual 37226 for detailed information.

The easYgen may be started, stopped, or acknowledged with CAN/Modbus. Therefore, two logical command variables have to be configured with the *LogicsManager*:

04.13 Remote request
04.14 Remote acknowledge

Configuration of the LogicsManager Function Start Request in AUTO

The Start req. in AUTO *LogicsManager* function (parameter ID 12120) can be configured in a way that a start request in AUTOMATIC operating mode is enabled as soon as a remote request is issued. Refer to the Performing Remote Start/Stop and Acknowledgement section of the Application Manual 37226 for a detailed configuration via HMI or ToolKit.
The remote request may be enabled by setting bit 0 (start) of the remote control word 503 to HIGH and may be disabled by setting bit 1 (stop) of the remote control word 503 to HIGH (refer to Remote Control Word 1 - Object 21F7h (Parameter ID 503) on page 96).

Example:

The following Modbus message must be sent to the easYgen to configure the *LogicsManager* function accordingly.

Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Delay ON	Delay OFF	Logic equation 1	Logic equation 2	Command 1	Command 2	Command 3
0000 (hex)	0000 (hex)	1232 (hex)	1000 (hex)	0802 (hex)	0700 (hex)	FB00 (hex)

The detailed composition of this message is as follows:

Delay ON = 0.00 s -> word 0 = 0000 (hex)
Delay OFF = 0.00 s -> word 1 = 0000 (hex)
Logic equation 1: sign 1 = '—'; Operator 1 = 'OR'; Sign 2 = 'FALSE', Operator 2 = 'OR' -> word 2 = 1232 (hex)
Logic equation 2: sign 3 = '—' -> word 3 = 1000 (hex)
Command 1 = 09.02 Discrete input 2 = 520 (dec) = 0208 (hex) -> word 4 = 0802 (hex)
Command 2 = 00.08 Flag 8 = 0 (dec) = 0000 (hex) -> word 5 = 0700 (hex)
Command 3 = 04.13 Remote request = 251 (dec) = 00FB (hex) -> word 6 = FB00 (hex)

The complete message of 7 words must be copied to address 12121 ff (12120+1) in one step. This is shown in Figure 3-27 using the ModScan32 software.

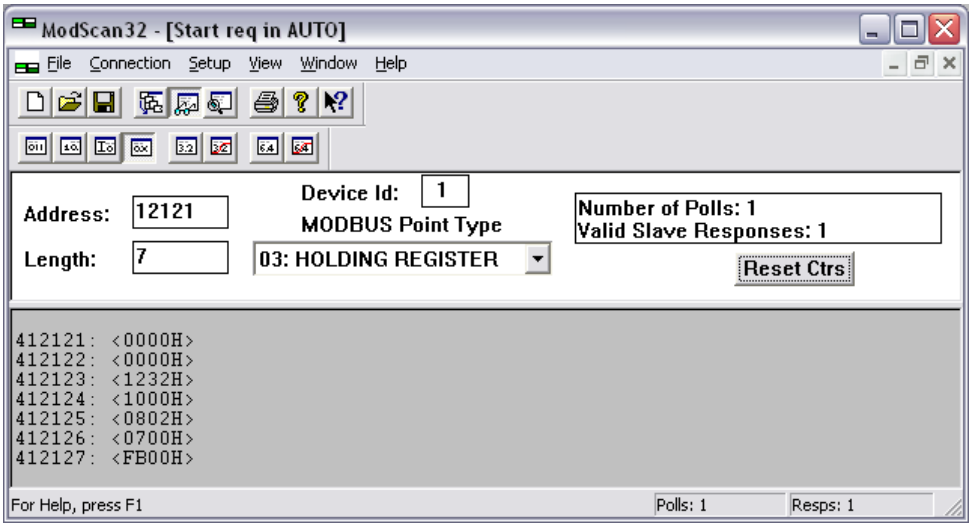

Figure 3-27: Modbus configuration - Start req in AUTO

Configuration of the LogicsManager Function External Acknowledge

The Ext. acknowledge *LogicsManager* function (parameter ID 12490) can be configured in a way that an external acknowledgement is performed as soon as the remote acknowledge signal is enabled. Refer to the Performing Remote Start/Stop and Acknowledgement section of the Application Manual 37226 for a detailed configuration via HMI or ToolKit.
External acknowledge may be enabled by setting bit 4 (external acknowledge) of the remote control word 503 to HIGH (refer to Remote Control Word 1 - Object 21F7h (Parameter ID 503) on page 96).

Example:

The following Modbus message must be sent to the easYgen to configure the *LogicsManager* function accordingly.

Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Delay ON	Delay OFF	Logic equation 1	Logic equation 2	Command 1	Command 2	Command 3
0000 (hex)	0000 (hex)	1212 (hex)	3000 (hex)	0B02 (hex)	FC00 (hex)	0000 (hex)

The detailed composition of this message is as follows:

Delay ON = 0.00 s -> word 0 = 0000 (hex)
Delay OFF = 0.00 s -> word 1 = 0000 (hex)
Logic equation 1: sign 1 = '—'; Operator 1 = 'OR'; Sign 2 = '—', Operator 2 = 'OR' -> word 2 = 1212 (hex)
Logic equation 2: sign 3 = 'FALSE' -> word 3 = 3000 (hex)
Command 1 = 09.05 Discrete input 5 = 523 (dec) = 020B (hex) -> word 4 = 0B02 (hex)
Command 2 = 04.14 Remote acknowledge = 252 (dec) = 00FC (hex) -> word 5 = FC00 (hex)
Command 3 = 00.01 Flag 1 (default) = 0 (dec) = 0000 (hex) -> word 6 = 0000 (hex)

The complete message of 7 words must be copied to address 12491 ff (12490+1) in one step. This is shown in Figure 3-28 using the ModScan32 software.

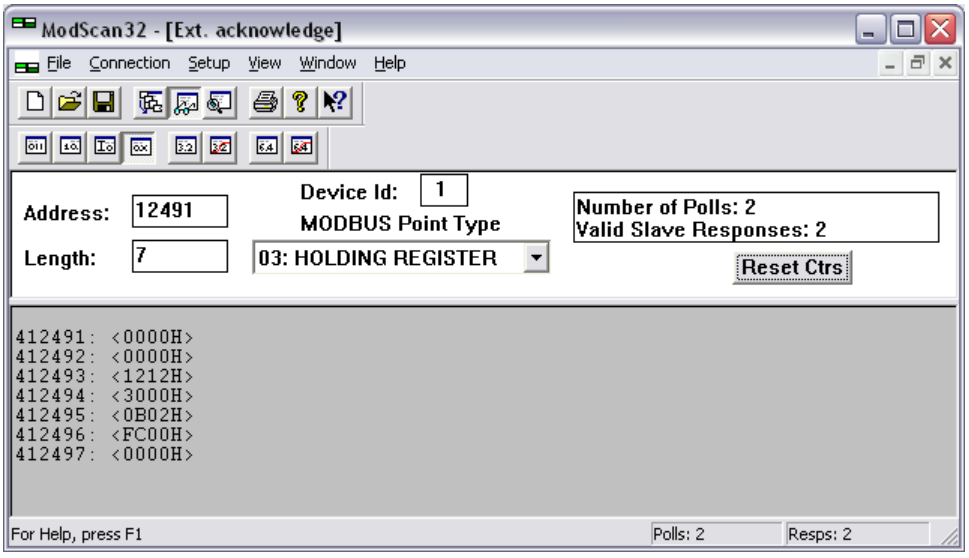

Figure 3-28: Modbus configuration - External acknowledge

Configuration of the LogicsManager Function Start w/o Load

The Start w/o load *LogicsManager* function (parameter ID 12540) can be configured in a way that it is always enabled. Refer to the Performing Remote Start/Stop and Acknowledgement section of the Application Manual 37226 for a detailed configuration via HMI or ToolKit.

Example:

The following Modbus message must be sent to the easYgen to configure the *LogicsManager* function accordingly.

Word 0	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Delay ON	Delay OFF	Logic equation 1	Logic equation 2	Command 1	Command 2	Command 3
0000 (hex)	0000 (hex)	2020 (hex)	2000 (hex)	0000 (hex)	0000 (hex)	0000 (hex)

The detailed composition of this message is as follows:

Delay ON = 0.00 s -> word 0 = 0000 (hex)
Delay OFF = 0.00 s -> word 1 = 0000 (hex)
Logic eq. 1: sign 1 = 'TRUE'; Operator 1 = 'AND'; Sign 2 = 'TRUE', Operator 2 = 'AND' -> word 2 = 2020 (hex)
Logic equation 2: sign 3 = 'TRUE' -> word 3 = 2000 (hex)
Command 1 = 00.01 Flag 1 (default) = 0 (dec) = 0000 (hex) -> word 4 = 0000 (hex)
Command 2 = 00.01 Flag 1 (default) = 0 (dec) = 0000 (hex) -> word 5 = 0000 (hex)
Command 3 = 00.01 Flag 1 (default) = 0 (dec) = 0000 (hex) -> word 6 = 0000 (hex)

The complete message of 7 words must be copied to address 12541 ff (12540+1) in one step. This is shown in Figure 3-29 using the ModScan32 software.

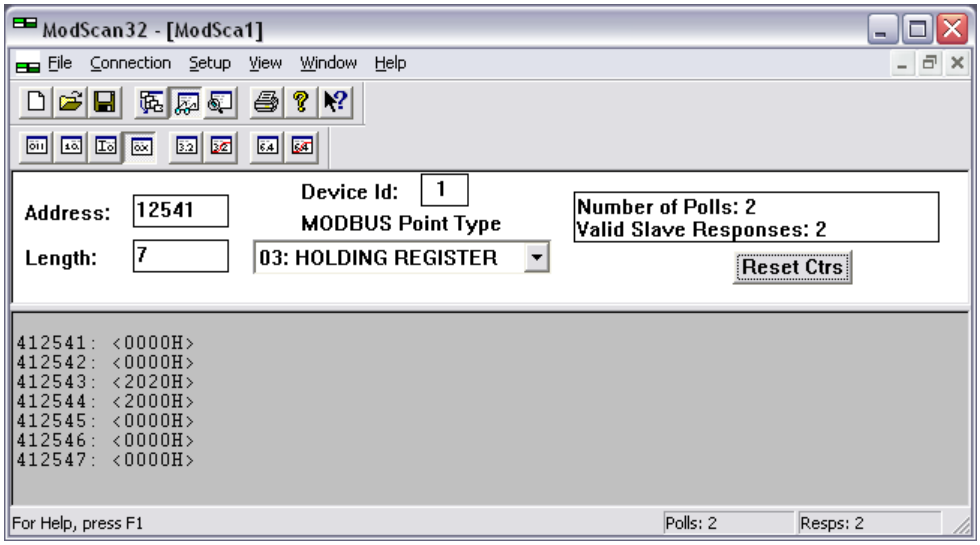

Figure 3-29: Modbus configuration - Start w/o load

Remote Acknowledging Single Alarm Messages

It is possible to remotely acknowledge single alarm messages through the Modbus by sending the respective parameter ID of the alarm to be acknowledged on ID 522. The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
522	Reset alarm list	0 to 65535	UNSIGNED 16

The parameter ID of the alarm to be acknowledged must be written to object 220A (hex), i.e. parameter ID 522.

Example:

A "Mains undervoltage 1" alarm (parameter ID 3012) is to be acknowledged (refer to Data Protocol 5003 starting on page 79 or the alarm list in the Operation Manual 37225).
Modbus address = 40000 + (Par. ID + 1) = 40523
Modbus length = 1 (UNSIGNED 16)

Figure 3-30 shows the settings made to parameter ID 522 in ModScan32. It is possible to set the format to decimal to view the value using the "display options".

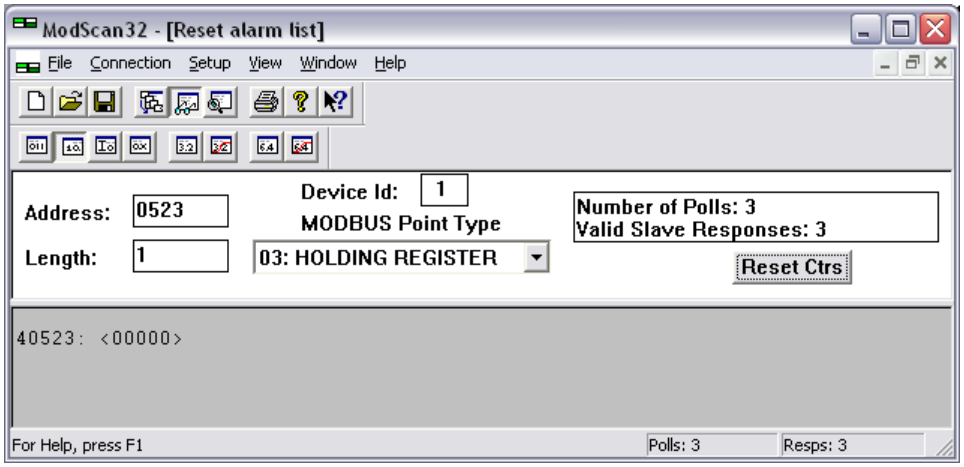

Figure 3-30: Modbus - remote control parameter 522

By double-clicking the address, a Write Register command may be issued. Figure 3-31 shows how the parameter ID of the alarm to be acknowledged is written using the ModScan32 Software.

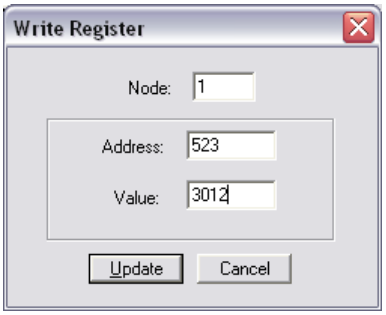

Figure 3-31: Modbus - write register - acknowledge alarm message

Remotely Clearing The Event History

It is possible to remotely clear the event history through the Modbus. The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
1706	`Clear eventlog`	YES / NO	UNSIGNED 16

In order to clear the event history, bit 0 of object 26AA (hex), i.e. parameter ID 1706, must be enabled.

Example:

The event history is to be cleared.
Modbus address = 40000 + (Par. ID + 1) = 41707
Modbus length = 1 (UNSIGNED 16)

Figure 3-32 shows the settings made to parameter ID 1706 in ModScan32. It is possible to set the format to binary to view single bits using the "display options".

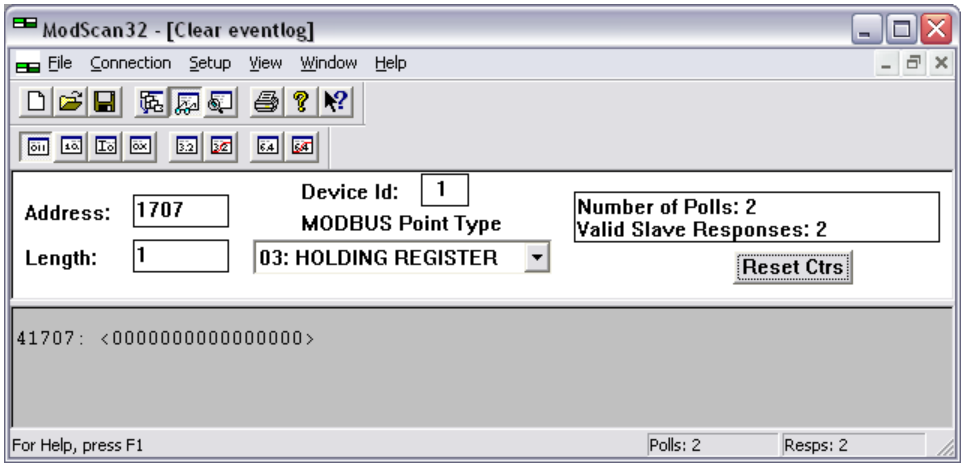

Figure 3-32: Modbus - remote control parameter 1706

By double-clicking the address, a Write Register command may be issued. Figure 3-33 shows how bit 0 is enabled using the ModScan32 Software.

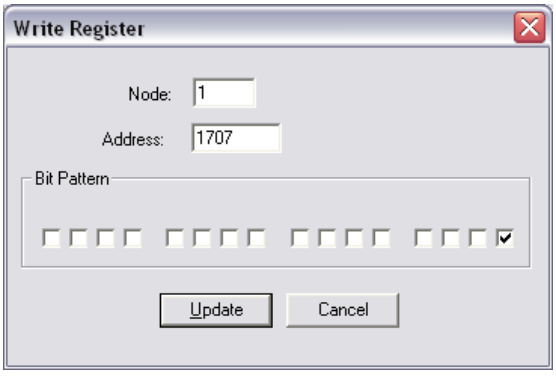

Figure 3-33: Modbus - write register - clear event history

Remotely Resetting the Default Values

Modbus via RS-232 (Serial Interface 1)

It is possible to remotely reset the unit to its default values through the Modbus (via RS-232) using the parameter IDs 1704 and 1701. The required procedure is detailed in the following steps.

Par. ID.	Parameter	Setting range	Data type
1704	Factory settings via RS-232	YES / NO	UNSIGNED 16
1701	Reset factory default values	YES / NO	UNSIGNED 16

In order to enable the resetting procedure, parameter ID 1704 must be enabled.

Example:

The resetting procedure via RS-232 is to be enabled.
Modbus address = 40000 + (Par. ID + 1) = 41705
Modbus length = 1 (UNSIGNED 16)

Figure 3-34 shows the settings made to parameter ID 1704 in ModScan32. It is possible to set the format to decimal to view the value using the "display options".

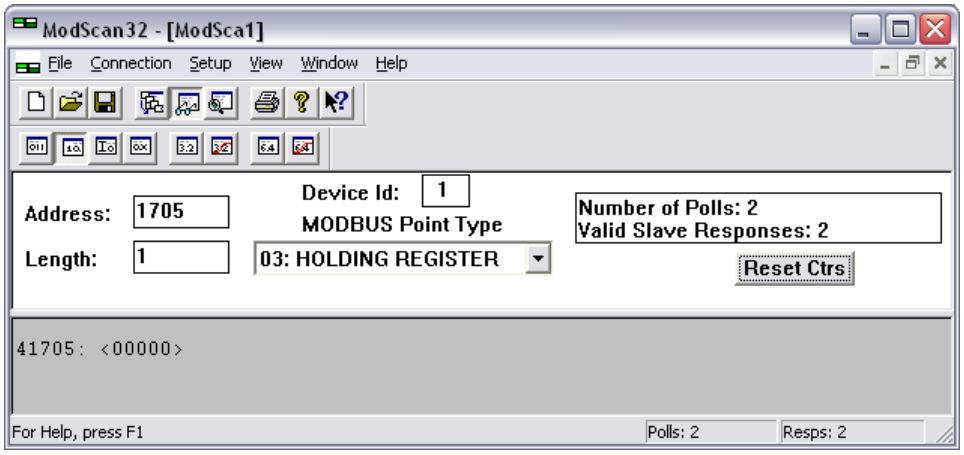

Figure 3-34: Modbus - remote control parameter 1704

By double-clicking the address, a Write Register command may be issued. Figure 3-35 shows how the parameter is enabled using the ModScan32 Software. The value must be set to "1" to enable the parameter.

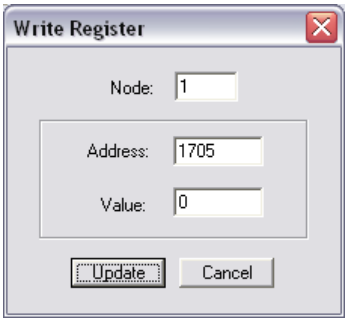

Figure 3-35: Modbus - write register - enable the resetting procedure via RS-232

In order to reset the default values, parameter ID 1701 must be enabled.

Example:

The default values are to be reset.
Modbus address = 40000 + (Par. ID + 1) = 41702
Modbus length = 1 (UNSIGNED 16)

Figure 3-36 shows the settings made to parameter ID 1701 in ModScan32. It is possible to set the format to decimal to view the value using the "display options".

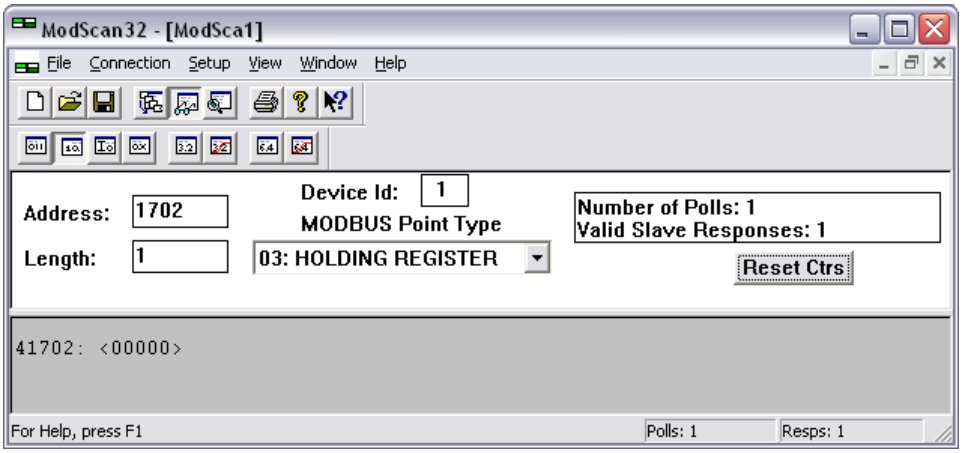

Figure 3-36: Modbus - remote control parameter 1701

By double-clicking the address, a Write Register command may be issued. Figure 3-37 shows how the parameter is enabled using the ModScan32 Software. The value must be set to "1" to enable the parameter.

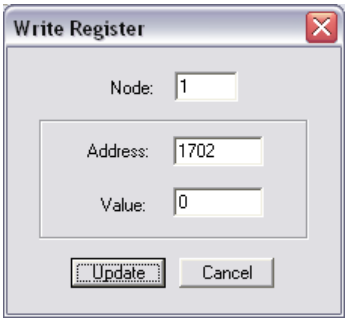

Figure 3-37: Modbus - write register - resetting the default values

Modbus via RS-485 (Serial Interface 2)

It is possible to remotely reset the unit to its default values through the Modbus (via RS-485) using the parameter IDs 1743 and 1701. The required procedure is the same as described under Modbus via RS-232 (Serial Interface 1) on page 73; however the parameter ID 1743 is used instead of parameter ID 1704.

Par. ID.	Parameter	Setting range	Data type
1743	Factory settings via RS-485	YES / NO	UNSIGNED 16
1701	Reset factory default values	YES / NO	UNSIGNED 16

Exception Responses

The easYgen Modbus interface has multiple exception responses to show that a request could not be executed. Exception responses can be recognized if the response telegram contains the request function code with an offset of 128 (0x80 hex).

Table 3-5 explains possible reasons for an exception response that occurred.

easYgen Modbus Exception Responses		
Code	**Name**	**Reason**
01	ILLEGAL FUNCTION	· The sent request function code is not supported by the easYgen Modbus interface.
02	ILLEGAL ADDRESS	· Permission to read/write the parameter is denied. · The amount of requested registers is wrong to read/write this registers.
03	ILLEGAL DATA VALUE	· The data value exceeds the min. and max. limitations of the parameter upon a write request. · There is no parameter on the requested address.

Table 3-5: Modbus - exception responses

Modbus Parameters

The following parameters are available for configuring the Modbus modules on the Serial Interfaces.

Serial Interface 1:

EN Modbus slave ID
DE Modbus Slave ID
CL2 3185 {0} {1o} {1oc} {2oc}

Serial interface 1 Modbus slave ID **0 to 255**

Modbus Slave ID for serial interface 1.

[0] means : Modbus is disabled
[1 to 255] : These are Slave IDs (default = 1)

EN Reply delay time
DE Zeitverzögerung der Antwort
CL2 3186 {0} {1o} {1oc} {2oc}

Serial interface 1 Modbus reply delay time **0 to 100 (1/10ms)**

Serial interface 1 Modbus reply delay time.

Adjustable in 10 ms steps.
Default = 0 ms

Serial Interface 2:

EN Modbus slave ID
DE Modbus Slave ID
CL2 3188 {0} {1o} {1oc} {2oc}

Serial interface 2 Modbus slave ID **0 to 255**

Modbus Slave ID for serial interface 2.

[0] means : Modbus is disabled
[1 to 255] : These are Slave IDs (default = 1)

EN Reply delay time
DE Zeitverzögerung der Antwort
CL2 3189 {0} {1o} {1oc} {2oc}

Serial interface 2 Modbus reply delay time **0 to 100 (1/10ms)**

Serial interface 2 Modbus reply delay time.

Adjustable in 10 ms steps.
Default = 0 ms

NOTE

Above parameters are an excerpt of the Configuration Manual 37224. Refer to this manual for all parameters.

Appendix A.
Supported J1939 ECUs & Remote Control Messages

The following table lists all ECUs, which are supported by the easYgen beyond the J1939 standard with the appropriate settings. We recommend the standard setting for all ECUs, which are not listed here. All other parameters shall be clarified with the ECU manufacturer.

Setting / ECU	Device type (ID 15102)	J1939 Device address (ID 15106)	Engine control address (ID 15107)	SPN Version (ID 15103)	Comment
Woodward EGS	EGS	234	0	n/a	
MTU ADEC	ADEC MTU	1	128	n/a	The easYgen is connected with the SAM via CAN. The SAM communicates with the ADEC using an own bus.
Deutz EMR2 Volvo EDC4	EMR2 Deutz	3	0	Version 1	
Volvo EMS2 Volvo EMS1 Volvo EDC3	EMS2 Volvo	17	0	n/a	The rated speed of the EMS1 and EDC3 cannot be switched via the easYgen.
Scania S6	S6 Scania	39	0	n/a	
MAN MFR/EDC7	MAN	253	0	n/a	The easYgen is connected with the MFR via CAN. The MFR communicates with the EDC7 using an own bus.
SISU EEM2 SISU EEM3	SISU	n/a	0 or 1	n/a	
Standard ECUs	Standard	234	0	n/a	

The following data is only transmitted to the corresponding ECU, if parameter "ECU remote controlled" is configured to "On", and parameter "Device type" is configured to one of the available ECU modes (if "Off" is configured, no J1939 remote control messages will be sent as well).

NOTE

Please note that some ECU manufacturers require that this functionality must be enabled first. In some cases, this is only possible by the manufacturer. Please consider this when ordering the ECU.

Remote control parameter	Woodward EGS	Scania S6	Deutz EMR2 Volvo EDC4	Volvo EMS2	Volvo EMS1/ EDC3	MTU ADEC	MAN	SISU EEM2/ EEM3	Std.	Comment
Engine Start	No	Yes	No	Yes	Yes	Yes	Yes	Only EEM3	No	If an engine start command is initiated by the easYgen, this information is transmitted in the form of a J1939 message bit to an ECU. If ignition speed is reached, this bit will be reset (*LogicsManager* command variable 03.02. "Starter").
Engine Stop	Yes	Yes	No	Yes	Yes	Yes	Yes	Only EEM3	Yes	This J1939 bit information is set, if a "Stop" command in automatic or manual mode is present in the easYgen. The "Stop" bit information remains set, until ignition speed is fallen below. After ignition speed has been fallen below, the "Stop" bit will be reset (*LogicsManager* command variable 03.27. "Stopping solenoid").
Droop mode	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	This J1939 bit information is set, if a "Start" command in automatic or manual mode is initiated by the easYgen. The bit remains set until the engine has been stopped. **Important:** This message is only sent, if the *LogicsManager* output 00.25 "Frequency droop active" is TRUE.
Idle Mode	No	Yes	No	Yes	Yes	No	No	No	No	This J1939 bit information is set, if "Idle" mode is active (*LogicsManager* command variable 04.15. "Idle run active" is TRUE). The bit will be reset, if "Idle" mode is no longer active (*LogicsManager* command variable 04.15. "Idle run active" is FALSE).
50/60 Hz switch	Yes	Yes	No	Yes*[2]	No	Yes	No*[1]	No	No	The J1939 information for 50 or 60 Hz mode is sent to the ECU depending on the "Rated system frequency" parameter setting (ID 1750) within the easYgen .
Speed bias	Yes	Yes offset	Yes absolute	Yes offset	Yes	Yes absolute	Yes absolute	Yes	Yes absolute	Refer to parameter 5537 in the Configuration Manual 37415 for detailed information.
Preglow	No	No	No	Yes	Yes	No	No	No	No	This J1939 bit information is set, if the easYgen is in "Preglow" mode (*LogicsManager* command variable 03.04. "Preglow/Ignition" is TRUE). The bit will be reset, if the "Preglow" phase has been expired or aborted.
Override	No	Yes	No	No	No	Yes	No	No	Yes	This J1939 bit information is set, if the easYgen is in critical mode (*LogicsManager* command variable 04.27. "Critical mode" is TRUE). The bit will be reset, if the critical mode has been expired or aborted.

*[1] Please contact manufacturer to clarify whether both frequencies (50/60 Hz) may be controlled by the speed bias.
*[2] In case the rated speed of the easYgen and the ECU don´t match, please make sure that the CAN connections works and change parameter 1750 of the easYgen once.

Appendix B.
Data Protocols

Data Protocol 5003

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
450001	450000	0	1,2		Protocol ID, always 5003		--
450002	450001	0	3,4	10100	Pickup speed	1	rpm
450003	450002	0	5,6	-	Control mode (STOP/AUTO/MANUAL) 1=AUTO 2=STOP 4=MANUAL	Mask:000Fh	(enum.)
450004	450003	1	1,2	160	Gen. powerfactor	0.001	
450005	450004	1	3,4,5,6	170	Av. Gen. Wye-Voltage	0.1	V
450007	450006	2	1,2	144	Gen. frequency	0.01	Hz
450008	450007	2	3,4,5,6	171	Av. Gen. Delta-Voltage	0.1	V
450010	450009	3	1,2	147	Mains frequency	0.01	Hz
450011	450010	3	3,4,5,6	173	Av. Mains Wye-Voltage	0.1	V
450013	450012	4	1,2	208	Mains power factor	0.001	
450014	450013	4	3,4,5,6	174	Av. Mains Delta-Voltage	0.1	V
450016	450015	5	1,2	209	Busbar 1: Frequency	0.01	Hz
450017	450016	5	3,4,5,6	216	Av. Busbar1 Delta-Voltage	0.1	V
450019	450018	6	1,2		internal		
450020	450019	6	3,4		internal		
450021	450020	6	5,6		internal		
450022	450021	7	1,2	10110	Battery voltage	0.1	V
450023	450022	7	3,4,5,6	207	Av. Mains Current	0.001	A
450025	450024	8	1,2	10111	Analog input 1	changeable	
450026	450025	8	3,4,5,6	185	Av. Gen. Current	0.001	A
450028	450027	9	1,2	10112	Analog input 2	changeable	
450029	450028	9	3,4,5,6	161	Meas. ground current	0.001	A
450031	450030	10	1,2	10115	Analog input 3	changeable	
450032	450031	10	3,4,5,6	159	Calculated ground current	0.001	A
450034	450033	11	1,2		internal		
450035	450034	11	3,4,5,6	111	Gen. current 1	0.001	A
450037	450036	12	1,2		internal		
450038	450037	12	3,4,5,6	112	Gen. current 2	0.001	A
450040	450039	13	1,2		internal		
450041	450040	13	3,4,5,6	113	Gen. current 3	0.001	A
450043	450042	14	1,2		internal		
450044	450043	14	3,4,5,6	134	Mains current L1	0.001	A
450046	450045	15	1,2		internal		
450047	450046	15	3,4		internal		
450048	450047	15	5,6		internal		
450049	450048	16	1,2		internal		
450050	450049	16	3,4		internal		
450051	450050	16	5,6		internal		
450052	450051	17	1,2		internal		
450053	450052	17	3,4,5,6	135	Total gen. power	1	W
450055	450054	18	1,2		internal		
450056	450055	18	3,4,5,6	140	Total mains power	1	W
450058	450057	19	1,2		internal		
450059	450058	19	3,4,5,6	136	Total gen. reactive power	1	var
450061	450060	20	1,2	10159	AI Auxiliary excitation D+	0.1	V
450062	450061	20	3,4,5,6	150	Total mains reactive power	1	var
450064	450063	21	1,2	2112	Overspeed 1 latched	Mask: 8000h	Bit
				2113	Overspeed 2 latched	Mask: 4000h	Bit

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
				2162	Underspeed 1 latched	Mask: 2000h	Bit
				2163	Underspeed 2 latched	Mask: 1000h	Bit
				2652	Unintended stop latched	Mask: 0800h	Bit
				2457	Speed det. alarm latched	Mask: 0400h	Bit
				2504	Shutdwn malfunct. latched	Mask: 0200h	Bit
				2603	GCB fail to close latched	Mask: 0100h	Bit
				2604	GCB fail to open latched	Mask: 0080h	Bit
				2623	MCB fail to close latched	Mask: 0040h	Bit
				2624	MCB fail to open latched	Mask: 0020h	Bit
				10017	CAN-Fault J1939 latched	Mask: 0010h	Bit
				3325	Start fail latched	Mask: 0008h	Bit
				2560	Mainten. days exceeded latched	Mask: 0004h	Bit
				2561	Mainten. hours exceeded latched	Mask: 0002h	Bit
					internal	Mask: 0001h	Bit
450065	450064	21	3,4,5,6	182	Busbar 1: voltage L1-L2	0.1	V
450067	450066	22	1,2	3064	GCB syn. timeout latched	Mask: 8000h	Bit
				3074	MCB syn. timeout latched	Mask: 4000h	Bit
				3084	GGB Timeout latched	Mask: 2000h	Bit
				4056	Charge alt. low voltage (D+) latched	Mask: 1000h	Bit
				2944	Ph.rotation mismatch latched	Mask: 0800h	Bit
				10084	no data receive at RPDO3 at CAN Interface 1	Mask: 0400h	
				10083	no data receive at RPDO2 at CAN Interface 1	Mask: 0200h	
				10082	no data receive at RPDO1 at CAN Interface 1	Mask: 0100h	
				10086	no data receive at RPDO2 (function 1) at CAN Interface 2	Mask: 0080h	
				10085	no data receive at RPDO1 (function 1) at CAN Interface 2	Mask: 0040h	
				4073	Parameter Alignment	Mask: 0010h	
				4064	Missing members on CAN	Mask: 0008h	
				1714	EEPROM failure latched	Mask: 0004h	Bit
				15125	Red stop lamp latched	Mask: 0002h	Bit
				15126	Amber warning lamp latched	Mask: 0001h	Bit
450068	450067	22	3,4		internal		
450069	450068	22	5,6		internal		
450070	450069	23	1,2		internal		
450071	450070	23	3,4		internal		
450072	450071	23	5,6		internal		
450073	450072	24	1,2	1912	Gen.overfreq. 1 latched	Mask: 8000h	Bit
				1913	Gen.overfreq. 2 latched	Mask: 4000h	Bit
				1962	Gen.underfreq. 1 latched	Mask: 2000h	Bit
				1963	Gen.underfreq. 2 latched	Mask: 1000h	Bit
				2012	Gen.overvolt. 1 latched	Mask: 0800h	Bit
				2013	Gen.overvolt. 2 latched	Mask: 0400h	Bit
				2062	Gen.undervolt. 1 latched	Mask: 0200h	Bit
				2063	Gen.undervolt. 2 latched	Mask: 0100h	Bit
				2218	Gen. overcurr. 1 latched	Mask: 0080h	Bit
				2219	Gen. overcurr. 2 latched	Mask: 0040h	Bit
				2220	Gen. overcurr. 3 latched	Mask: 0020h	Bit
				2262	Gen. Rv/Rd pow.1 latched	Mask: 0010h	Bit
				2263	Gen. Rv/Rd pow.2 latched	Mask: 0008h	Bit
				2314	Gen. Overload IOP 1 latched	Mask: 0004h	Bit
				2315	Gen. Overload IOP 2 latched	Mask: 0002h	Bit
					internal	Mask: 0001h	Bit
450074	450073	24	3,4,5,6	108	Gen. voltage L1-L2	0.1	V
450076	450075	25	1,2	2412	Unbal. load 1 latched	Mask: 8000h	Bit
				2413	Unbal. load 2 latched	Mask: 4000h	Bit
				3907	Gen. Asymmetry latched	Mask: 2000h	Bit
				3263	Ground fault 1 latched	Mask: 1000h	Bit
				3264	Ground fault 2 latched	Mask: 0800h	Bit
				3955	Gen. phase rot. misw. Latched	Mask: 0400h	Bit
				2924	Gen act.pwr mismatch Latched	Mask: 0200h	Bit
				3124	Gen. unloading fault Latched	Mask: 0100h	Bit
				4038	Inv.time ov.curr. Latched	Mask: 0080h	Bit
				2664	Operating range failed, latched	Mask: 0040h	Bit

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
				2362	Gen. Overload MOP 1 latched	Mask: 0020h	Bit
				2363	Gen. Overload MOP 2 latched	Mask: 0010h	Bit
				2337	Gen. overexcited 1 latched	Mask: 0008h	Bit
				2338	Gen. overexcited 2 latched	Mask: 0004h	Bit
				2387	Gen. underexcited 1 latched	Mask: 0002h	Bit
				2388	Gen. underexcited 2 latched	Mask: 0001h	Bit
450077	450076	25	3,4,5,6	114	Gen. voltage L1-N	0.1	V
450079	450078	26	1,2	2862	Mains ov.freq. 1 latched	Mask: 8000h	Bit
				2863	Mains ov.freq. 2 latched	Mask: 4000h	Bit
				2912	Mains un.freq. 1 latched	Mask: 2000h	Bit
				2913	Mains un.freq. 2 latched	Mask: 1000h	Bit
				2962	Mains ov.volt. 1 latched	Mask: 0800h	Bit
				2963	Mains ov.volt. 2 latched	Mask: 0400h	Bit
				3012	Mains un.volt. 1 latched	Mask: 0200h	Bit
				3013	Mains un.volt. 2 latched	Mask: 0100h	Bit
				3057	Mains phaseshift latched	Mask: 0080h	Bit
				3114	Mains decoupling latched	Mask: 0040h	Bit
					internal	Mask: 0020h	Bit
					internal	Mask: 0010h	Bit
					internal	Mask: 0008h	Bit
				3975	Mains phase rot. misw. Latched	Mask: 0004h	Bit
					internal	Mask: 0002h	Bit
					internal	Mask: 0001h	Bit
450080	450079	26	3,4,5,6	109	Gen. voltage L2-L3	0.1	V
450082	450081	27	1,2	3217	Mains import power 1 latched	Mask: 8000h	Bit
				3218	Mains import power 2 latched	Mask: 4000h	Bit
				3241	Mains export power 1 latched	Mask: 2000h	Bit
				3242	Mains export power 2 latched	Mask: 1000h	Bit
				2985	Mains overexcited 1 latched	Mask: 0800h	Bit
				2986	Mains overexcited 2 latched	Mask: 0400h	Bit
				3035	Mains underexcited 1 latched	Mask: 0200h	Bit
				3036	Mains underexcited 2 latched	Mask: 0100h	Bit
					internal	Mask: 0080h	Bit
				2934	Mns act.pwr mismatch latched	Mask: 0040h	Bit
					internal	Mask: 0020h	Bit
					internal	Mask: 0010h	Bit
					internal	Mask: 0008h	Bit
					internal	Mask: 0004h	Bit
					internal	Mask: 0002h	Bit
					internal	Mask: 0001h	Bit
450083	450082	27	3,4,5,6	115	Gen. voltage L2-N	0.1	V
450085	450084	28	1,2	10600	State Digital Input 1 latched	Mask: 8000h	Bit
				10601	State Digital Input 2 latched	Mask: 4000h	Bit
				10602	State Digital Input 3 latched	Mask: 2000h	Bit
				10603	State Digital Input 4 latched	Mask: 1000h	Bit
				10604	State Digital Input 5 latched	Mask: 0800h	Bit
				10605	State Digital Input 6 latched	Mask: 0400h	Bit
				10607	State Digital Input 7 latched	Mask: 0200h	Bit
				10608	State Digital Input 8 latched	Mask: 0100h	Bit
				10609	State Digital Input 9 latched	Mask: 0080h	Bit
				10610	State Digital Input 10 latched	Mask: 0040h	Bit
				10611	State Digital Input 11 latched	Mask: 0020h	Bit
				10612	State Digital Input 12 latched	Mask: 0010h	Bit
450086	450085	28	3,4,5,6	110	Gen. voltage L3-L1	0.1	V
450088	450087	29	1,2		internal		
450089	450088	29	3,4,5,6	116	Gen. voltage L3-N	0.1	V
450091	450090	30	1,2	16376	State ext. Digital Input 16 latched	Mask: 8000h	Bit
				16375	State ext. Digital Input 15 latched	Mask: 4000h	Bit
				16374	State ext. Digital Input 14 latched	Mask: 2000h	Bit
				16373	State ext. Digital Input 13 latched	Mask: 1000h	Bit
				16372	State ext. Digital Input 12 latched	Mask: 0800h	Bit
				16371	State ext. Digital Input 11 latched	Mask: 0400h	Bit
				16370	State ext. Digital Input 10 latched	Mask: 0200h	Bit
				16369	State ext. Digital Input 9 latched	Mask: 0100h	Bit
				16368	State ext. Digital Input 8 latched	Mask: 0080h	Bit
				16367	State ext. Digital Input 7 latched	Mask: 0040h	Bit

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
				16366	State ext. Digital Input 6 latched	Mask: 0020h	Bit
				16365	State ext. Digital Input 5 latched	Mask: 0010h	Bit
				16364	State ext. Digital Input 4 latched	Mask: 0008h	Bit
				16362	State ext. Digital Input 3 latched	Mask: 0004h	Bit
				16361	State ext. Digital Input 2 latched	Mask: 0002h	Bit
				16360	State ext. Digital Input 1 latched	Mask: 0001h	Bit
450092	450091	30	3,4,5,6	118	Mains voltage L1-L2	0.1	V
450094	450093	31	1,2	10033	Alarm flexible limit 16 latched	Mask: 8000h	Bit
				10032	Alarm flexible limit 15 latched	Mask: 4000h	Bit
				10031	Alarm flexible limit 14 latched	Mask: 2000h	Bit
				10030	Alarm flexible limit 13 latched	Mask: 1000h	Bit
				10029	Alarm flexible limit 12 latched	Mask: 0800h	Bit
				10028	Alarm flexible limit 11 latched	Mask: 0400h	Bit
				10027	Alarm flexible limit 10 latched	Mask: 0200h	Bit
				10026	Alarm flexible limit 9 latched	Mask: 0100h	Bit
				10025	Alarm flexible limit 8 latched	Mask: 0080h	Bit
				10024	Alarm flexible limit 7 latched	Mask: 0040h	Bit
				10023	Alarm flexible limit 6 latched	Mask: 0020h	Bit
				10022	Alarm flexible limit 5 latched	Mask: 0010h	Bit
				10021	Alarm flexible limit 4 latched	Mask: 0008h	Bit
				10020	Alarm flexible limit 3 latched	Mask: 0004h	Bit
				10019	Alarm flexible limit 2 latched	Mask: 0002h	Bit
				10018	Alarm flexible limit 1 latched	Mask: 0001h	Bit
450095	450094	31	3,4,5,6	121	Mains voltage L1-N	0.1	V
450097	450096	32	1,2	10049	Alarm flexible limit 32 latched	Mask: 8000h	Bit
				10048	Alarm flexible limit 31 latched	Mask: 4000h	Bit
				10047	Alarm flexible limit 30 latched	Mask: 2000h	Bit
				10046	Alarm flexible limit 29 latched	Mask: 1000h	Bit
				10045	Alarm flexible limit 28 latched	Mask: 0800h	Bit
				10044	Alarm flexible limit 27 latched	Mask: 0400h	Bit
				10043	Alarm flexible limit 26 latched	Mask: 0200h	Bit
				10042	Alarm flexible limit 25 latched	Mask: 0100h	Bit
				10041	Alarm flexible limit 24 latched	Mask: 0080h	Bit
				10040	Alarm flexible limit 23 latched	Mask: 0040h	Bit
				10039	Alarm flexible limit 22 latched	Mask: 0020h	Bit
				10038	Alarm flexible limit 21 latched	Mask: 0010h	Bit
				10037	Alarm flexible limit 20 latched	Mask: 0008h	Bit
				10036	Alarm flexible limit 19 latched	Mask: 0004h	Bit
				10035	Alarm flexible limit 18 latched	Mask: 0002h	Bit
				10034	Alarm flexible limit 17 latched	Mask: 0001h	Bit
450098	450097	32	3,4,5,6	119	Mains voltage L2-L3	0.1	V
450100	450099	33	1,2	10057	Alarm flexible limit 40 latched	Mask: 0080h	Bit
				10056	Alarm flexible limit 39 latched	Mask: 0040h	Bit
				10055	Alarm flexible limit 38 latched	Mask: 0020h	Bit
				10054	Alarm flexible limit 37 latched	Mask: 0010h	Bit
				10053	Alarm flexible limit 36 latched	Mask: 0008h	Bit
				10052	Alarm flexible limit 35 latched	Mask: 0004h	Bit
				10051	Alarm flexible limit 34 latched	Mask: 0002h	Bit
				10050	Alarm flexible limit 33 latched	Mask: 0001h	Bit
450101	450100	33	3,4,5,6	122	Mains voltage L2-N	0.1	V
450103	450102	34	1,2	1008	Batt.overvolt.2 latched	Mask: 0008h	Bit
				1007	Batt.undervolt.2 latched	Mask: 0004h	Bit
				1006	Batt.overvolt.1 latched	Mask: 0002h	Bit
				1005	Batt.undervolt.1 latched	Mask: 0001h	Bit
450104	450103	34	3,4,5,6	120	Mains voltage L3-L1	0.1	V
450106	450105	35	1,2	10131	(unused)	Mask: 0040h	Bit
					Alarm class F latched	Mask: 0020h	Bit
					Alarm class E latched	Mask: 0010h	Bit
					Alarm class D latched	Mask: 0008h	Bit
					Alarm class C latched	Mask: 0004h	Bit
					Alarm class B latched	Mask: 0002h	Bit
					Alarm class A latched	Mask: 0001h	Bit
450107	450106	35	3,4,5,6	123	Mains voltage L3-N	0.1	V
450109	450108	36	1,2	10014	Analog inp. 1, wire brake or shortcut latched	Mask: 0002h	Bit
				10015	Analog inp. 2, wire brake or shortcut latched	Mask: 0004h	Bit

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
				10060	Analog inp. 3, wire brake or shortcut latched	Mask: 0008h	Bit
450110	450109	36	3,4		internal		
450111	450110	36	5,6		internal		
450112	450111	37	1,2	10107	Digital outputs 1 to 12		
					Relay-Output 1 (inverted)	Mask: 8000h	Bit
					Relay-Output 2	Mask: 4000h	Bit
					Relay-Output 3	Mask: 2000h	Bit
					Relay-Output 4	Mask: 1000h	Bit
					Relay-Output 5	Mask: 0800h	Bit
					Relay-Output 6	Mask: 0400h	Bit
					Relay-Output 7	Mask: 0200h	Bit
					Relay-Output 8	Mask: 0100h	Bit
					Relay-Output 9	Mask: 0080h	Bit
					Relay-Output 10	Mask: 0040h	Bit
					Relay-Output 11	Mask: 0020h	Bit
					Relay-Output 12	Mask: 0010h	Bit
					internal	Mask: 0008h	Bit
					internal	Mask: 0004h	Bit
					internal	Mask: 0002h	Bit
					internal	Mask: 0001h	Bit
450113	450112	37	3,4		internal		
450114	450113	37	5,6	8005	Output to external CAN-I/O Relay 16	Mask DO 16 8000h	Bit
					Relay 15	Mask DO 15 4000h	Bit
					Relay 14	Mask DO 14 2000h	Bit
					Relay 13	Mask DO 13 1000h	Bit
					Relay 12	Mask DO 12 0800h	Bit
					Relay 11	Mask DO 11 0400h	Bit
					Relay 10	Mask DO 10 0200h	Bit
					Relay 9	Mask DO 09 0100h	Bit
					Relay 8	Mask DO 08 0080h	Bit
					Relay 7	Mask DO 07 0040h	Bit
					Relay 6	Mask DO 06 0020h	Bit
					Relay 5	Mask DO 05 0010h	Bit
					Relay 4	Mask DO 04 0008h	Bit
					Relay 3	Mask DO 03 0004h	Bit
					Relay 2	Mask DO 02 0002h	Bit
					Relay 1	Mask DO 01 0001h	Bit
450115	450114	38	1,2	10310	Analog output 1	0,01	%
450116	450115	38	3,4	10311	Analog output 2	0,01	%
450117	450116	38	5,6		internal		
450118	450117	39	1,2		internal		
450119	450118	39	3,4		internal		
450120	450119	39	5,6		internal		
450121	450120	40	1,2	10202	Operation modes 13200 = Auxiliary services postrun 13216 = Idle run active 13201 = Aux. services prerun 13250 = Gen. stable time 13202 = Critical mode 13251 = In operation 13203 = Motor Stop 13252 = Power limited prerun		(enum.)

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
					13204 = Cool down 13253 = AUTO mode ready 13205 = Mains settling 13254 = Ramp to rated 13206 = Start 13255 = GCB open 13207 = Start – Pause 13256 = Unloading generator 13208 = Preglow 13257 = MCB open 13209 = GCB dead bus close 13258 = Loading generator 13210 = MCB dead bus close 13259 = Synchronization GCB 13211 = Emergency run 13260 = Synchronization MCB 13212 = Turning 13261 = GCB -> MCB Delay 13213 = Ignition 13262 = MCB -> GCB Delay 13214 = Crank protect 13263 = Start w/o Load 13215 = Emergency/Critical 13264 = Unloading mains		
450122	450121	40	3,4,5,6	2520	Gen. real energy	0,01	MWh
450124	450123	41	1,2	2540	Engine, number of startrequests	1	
450125	450124	41	3,4,5,6	2522	Gen. positive reactive power	0,01	Mvarh
450127	450126	42	1,2	2558	Hours until next maintenance	1	h
450128	450127	42	3,4,5,6	2568	Gen. hours of operation	0,01	h
450130	450129	43	1,2	5541	Setpoint frequency	0,01	Hz
450131	450130	43	3,4,5,6	5542	Setpoint active power	0,1	kW
450133	450132	44	1,2,3,4	5640	Setpoint voltage	1	V
450135	450134	44	5,6	5641	Setpoint power factor	0,001	
450136	450135	45	1,2	4153	Idle mode monitoring (suppresses undervolt, underfreq,...)	Mask: 8000h	
					Idle mode activ	Mask: 4000h	
					Start without closing GCB	Mask: 2000h	
					Block closing GCB	Mask: 1000h	
					internal	Mask: 0800h	
					internal	Mask: 0400h	
					Cooldown is active	Mask: 0200h	
					Auxiliary services generally active	Mask: 0100h	
					Engine delay timer has expired	Mask: 0080h	
					Breaker delay timer has expired	Mask: 0040h	
					Engine shall run	Mask: 0020h	
					Critical mode is active in auto mode	Mask: 0010h	
					Engine is released	Mask: 0008h	
					Auxiliary services prerun is active	Mask: 0004h	
					Auxiliary services postrun is active	Mask: 0002h	
					Lamp test is active	Mask: 0001h	
450137	450136	45	3,4	4154	Crank (Starter) is active	Mask: 8000h	
					Operating Magnet / Gasrelay is active	Mask: 4000h	
					Preglow / Ignition is active	Mask: 2000h	
					Mains settling timer is running	Mask: 1000h	
					Emergency mode is currently active	Mask: 0800h	
					Emergency breaked by override	Mask: 0400h	
					Emergency Mains overfrequency	Mask: 0200h	
					Emergency Mains underfrequency	Mask: 0100h	
					Emergency Mains overvoltage	Mask: 0080h	
					Emergency Mains undervoltage	Mask: 0040h	
					Stopping Magnet is active	Mask: 0020h	
					internal	Mask: 0010h	
					The genset runs mains parallel	Mask: 0008h	
					internal	Mask: 0004h	
					internal	Mask: 0002h	
					Increment Start Counter	Mask: 0001h	
450138	450137	45	5,6	4155	Dead Band Contr. Freq./Power raise	Mask: 8000h	
					Dead Band Contr. Freq./Power raise	Mask: 4000h	

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
					Dead Band Controller Volt./ReactPow raise	Mask: 2000h	
					Dead Band Controller Volt./ReactPow lower	Mask: 1000h	
					GCB is closed	Mask: 0800h	
					MCB is closed	Mask: 0400h	
					internal	Mask: 0200h	
					Synchronization GCB is active	Mask: 0100h	
					Opening GCB is active	Mask: 0080h	
					Closing GCB is active	Mask: 0040h	
					Synchronization MCB is active	Mask: 0020h	
					Opening MCB is active	Mask: 0010h	
					Closing MCB is active	Mask: 0008h	
					Unloading generator is active	Mask: 0004h	
					Unloading mains is active	Mask: 0002h	
					Power limited prerun	Mask: 0001h	
450139	450138	46	1,2	4156	internal	Mask: 8000h	
					internal	Mask: 4000h	
					internal	Mask: 2000h	
					internal	Mask: 1000h	
					internal	Mask: 0800h	
					Dead busbar closure request for GCB or MCB or GGB	Mask: 0400h	
					Active power load share is active	Mask: 0200h	
					Reactive power load share is active	Mask: 0100h	
					Generator is requested	Mask: 0080h	
					internal	Mask: 0040h	
					internal	Mask: 0020h	
					internal	Mask: 0010h	
					internal	Mask: 0008h	
					internal	Mask: 0004h	
					internal	Mask: 0002h	
					internal	Mask: 0001h	
450140	450139	46	3,4		internal		
450141	450140	46	5,6		internal		
450142	450141	47	1,2		internal		
450143	450142	47	3,4		internal		
450144	450143	47	5,6		internal		
450145	450144	48	1,2		internal		
450146	450145	48	3,4		internal		
450147	450146	48	5,6		internal		
450148	450147	49	1,2		internal		
450149	450148	49	3,4		internal		
450150	450149	49	5,6		internal		
450151	450150	50	1,2		internal		
450152	450151	50	3,4		internal		
450153	450152	50	5,6		internal		
450154	450153	51	1,2		internal		
450155	450154	51	3,4		internal		
450156	450155	51	5,6		internal		
450157	450156	52	1,2		internal		
450158	450157	52	3,4		internal		
450159	450158	52	5,6		internal		
450160	450159	53	1,2		internal		
450161	450160	53	3,4		internal		
450162	450161	53	5,6		internal		
450163	450162	54	1,2		internal		
450164	450163	54	3,4		internal		
450165	450164	54	5,6		internal		
450166	450165	55	1,2		internal		
450167	450166	55	3,4		internal		
450168	450167	55	5,6		internal		
450169	450168	56	1,2		internal		
450170	450169	56	3,4		internal		
450171	450170	56	5,6		internal		
450172	450171	57	1,2		internal		
450173	450172	57	3,4		internal		

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
450174	450173	57	5,6		internal		
450175	450174	58	1,2	15109	J1939 MTU ADEC ECU Failure Codes	1	
450176	450175	58	3,4		internal		
450177	450176	58	5,6		internal		
450178	450177	59	1,2	15304	Engine Stop Information (extracted from DEUTZ-specific J1939-Message; refer to the Deutz documentation for information)	1	(enum.)
450179	450178	59	3,4		internal		
450180	450179	59	5,6		internal		
450181	450180	60	1,2	15305	J1939 DLN2-Message Scania S6		
					Engine Coolant Temperature		
					J1939-Message not available	Mask 8000h	
					Sensor fault	Mask 4000h	
					High Temperature.	Mask 2000h	
					NOT High Temperature	Mask 1000h	
					Engine Oil Pressure		
					J1939-Message not available	Mask 0800h	
					Sensor fault	Mask 0400h	
					Low Pressure	Mask 0200h	
					NOT Low Pressure	Mask 0100h	
					High Engine Oil Level		
					J1939-Message not available	Mask 0080h	
					Sensor fault	Mask 0040h	
					High Level	Mask 0020h	
					NOT High Level	Mask 0010h	
					Low Engine Oil Level		
					J1939-Message not available	Mask 0008h	
					Sensor fault	Mask 0004h	
					Low Level	Mask 0002h	
					NOT Low Level	Mask 0001h	
450182	450181	60	3,4		internal		
450183	450182	60	5,6		internal		
					1. Active Diagnostic Trouble Code (DM1)		
450184	450183	61	1,2,3,4	15400	SPN		
450186	450185	61	5,6	15401	FMT	Mask FF00h	
				15402	OC	Mask 00FFh	
					2. Active Diagnostic Trouble Code (DM1)		
450187	450186	62	1,2,3,4	15403	SPN		
450189	450188	62	5,6	15404	FMT	Mask FF00h	
				15405	OC	Mask 00FFh	
					3. Active Diagnostic Trouble Code (DM1)		
450190	450189	63	1,2,3,4	15406	SPN		
450192	450191	63	5,6	15407	FMT	Mask FF00h	
				15408	OC	Mask 00FFh	
					4. Active Diagnostic Trouble Code (DM1)		
450193	450192	64	1,2,3,4	15409	SPN		
450195	450194	64	5,6	15410	FMT	Mask FF00h	
				15411	OC	Mask 00FFh	
					5. Active Diagnostic Trouble Code (DM1)		
450196	450195	65	1,2,3,4	15412	SPN		
450198	450197	65	5,6	15413	FMT	Mask FF00h	
				15414	OC	Mask 00FFh	
					6. Active Diagnostic Trouble Code (DM1)		
450199	450198	66	1,2,3,4	15415	SPN		
450201	450200	66	5,6	15416	FMT	Mask FF00h	
				15418	OC	Mask 00FFh	
					7. Active Diagnostic Trouble Code (DM1)		
450202	450201	67	1,2,3,4	15419	SPN		
450204	450203	67	5,6	15420	FMT	Mask FF00h	

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
				15421	OC	Mask 00FFh	
					8. Active Diagnostic Trouble Code (DM1)		
450205	450204	68	1,2,3,4	15422	SPN		
450207	450206	68	5,6	15423	FMT	Mask FF00h	
				15424	OC	Mask 00FFh	
					9. Active Diagnostic Trouble Code (DM1)		
450208	450207	69	1,2,3,4	15425	SPN		
450210	450209	69	5,6	15426	FMT	Mask FF00h	
				15427	OC	Mask 00FFh	
					10. Active Diagnostic Trouble Code (DM1)		
450211	450210	70	1,2,3,4	15428	SPN		
450213	450212	70	5,6	15429	FMT	Mask FF00h	
				15430	OC	Mask 00FFh	
					1. Previously Active Diagnostic Trouble Code (DM2)		
450214	450213	71	1,2,3,4	15450	SPN		
450216	450215	71	5,6	15451	FMT	Mask FF00h	
				15452	OC	Mask 00FFh	
					2. Previously Active Diagnostic Trouble Code (DM2)		
450217	450216	72	1,2,3,4	15453	SPN		
450219	450218	72	5,6	15454	FMT	Mask FF00h	
				15455	OC	Mask 00FFh	
					3. Previously Active Diagnostic Trouble Code (DM2)		
450220	450219	73	1,2,3,4	15456	SPN		
450222	450221	73	5,6	15457	FMT	Mask FF00h	
				15458	OC	Mask 00FFh	
					4. Previously Active Diagnostic Trouble Code (DM2)		
450223	450222	74	1,2,3,4	15459	SPN		
450225	450224	74	5,6	15460	FMT	Mask FF00h	
				15461	OC	Mask 00FFh	
					5. Previously Active Diagnostic Trouble Code (DM2)		
450226	450225	75	1,2,3,4	15462	SPN		
450228	450227	75	5,6	15463	FMT	Mask FF00h	
				15464	OC	Mask 00FFh	
					6. Previously Active Diagnostic Trouble Code (DM2)		
450229	450228	76	1,2,3,4	15465	SPN		
450231	450230	76	5,6	15466	FMT	Mask FF00h	
				15467	OC	Mask 00FFh	
					7. Previously Active Diagnostic Trouble Code (DM2)		
450232	450231	77	1,2,3,4	15468	SPN		
450234	450233	77	5,6	15469	FMT	Mask FF00h	
				15470	OC	Mask 00FFh	
					8. Previously Active Diagnostic Trouble Code (DM2)		
450235	450234	78	1,2,3,4	15471	SPN		
450237	450236	78	5,6	15472	FMT	Mask FF00h	
				15473	OC	Mask 00FFh	
					9. Previously Active Diagnostic Trouble Code (DM2)		
450238	450237	79	1,2,3,4	15474	SPN		
450240	450239	79	5,6	15475	FMT	Mask FF00h	
				15476	OC	Mask 00FFh	
					10. Previously Active Diagnostic Trouble Code (DM2)		
450241	450240	80	1,2,3,4	15477	SPN		
450243	450242	80	5,6	15478	FMT	Mask FF00h	
				15479	OC	Mask 00FFh	
450244	450243	81	1,2	15395	**DM1** Lamp Status		
					Malfunction Lamp		

Modbus	Modbus	CAN		Parameter ID	Description	Multiplier	Units
Modicon start addr.	Start addr. (*1)	Data byte 0 (Mux)	Data byte				
					(unused)	Mask 8000h	
					(unused)	Mask 4000h	
					On	Mask 2000h	
					Off	Mask 1000h	
					Red Stop Lamp		
					(unused)	Mask 0800h	
					(unused)	Mask 0400h	
					On	Mask 0200h	
					Off	Mask 0100h	
					Amber Warning Lamp		
					(unused)	Mask 0080h	
					(unused)	Mask 0040h	
					On	Mask 0020h	
					Off	Mask 0010h	
					Protect Lamp		
					(unused)	Mask 0008h	
					(unused)	Mask 0004h	
					On	Mask 0002h	
					Off	Mask 0001h	
450245	450244	81	3,4	15445	**DM2** Lamp Status		
					Malfunction Lamp		
					(unused)	Mask 8000h	
					(unused)	Mask 4000h	
					On	Mask 2000h	
					Off	Mask 1000h	
					Red Stop Lamp		
					(unused)	Mask 0800h	
					(unused)	Mask 0400h	
					On	Mask 0200h	
					Off	Mask 0100h	
					Amber Warning Lamp		
					(unused)	Mask 0080h	
					(unused)	Mask 0040h	
					On	Mask 0020h	
					Off	Mask 0010h	
					Protect Lamp		
					(unused)	Mask 0008h	
					(unused)	Mask 0004h	
					On	Mask 0002h	
					Off	Mask 0001h	
450246	450245	81	5,6		internal		
450247	450246	82	1,2,3,4	15200	Engine Speed (j1939-EEC1)	0,1	rpm
450249	450248	82	5,6	15202	Engine Coolant Temp. (J1939-ET1)	1	°C
450250	450249	83	1,2,3,4	15201	Total engine hours (j1939-HOURS)	1	h
450252	450251	83	5,6	15203	Fuel temperature (j1939-ET1)	1	°C
450253	450252	84	1,2,3,4	15204	Engine Oil Temperature (j1939-ET1)	0,01	°C
450255	450254	84	5,6	15205	Engine Oil Pressure (j1939-EFL/P1)	1	kPa
450256	450255	85	1,2,3,4	15211	Fuel Rate (j1939-LFE)	0,01	L/h
450258	450257	85	5,6	15206	Coolant Level (j1939-EFL/P1)	0,1	%
450259	450258	86	1,2	15207	Throttle position (j1939-EEC2)	0,1	%
450260	450259	86	3,4	15208	Load at current Speed (j1939-EEC2)	1	%
450261	450260	86	5,6	15210	Engine oil level (j1939-EFL/P1)	0,1	%
450262	450261	87	1,2	15214	Boost pressure (j1939-IC1)	1	kPa
450263	450262	87	3,4	15215	Intake Manifold Temp (j1939-IC1)	1	°C
450264	450263	87	5,6	15212	Barometric Pressure (j1939-AMB)	0,1	kPa
450265	450264	88	1,2	15213	Air inlet temperature (j1939-AMB)	1	°C
450266	450265	88	3,4	15209	Actual engine torque (j1939-EEC1)	1	%
450267	450266	88	5,6		internal		
450268	450267	89	1,2,3,4	15216	Exhaust Gas Temp.(J1939-IC1)	0,01	°C
450270	450269	89	5,6		internal		

Data Protocol 5004 (Generator Data)

CAN		Parameter ID	Description	Multiplier	Units
Data byte 0 (Mux)	Data byte				
0	1,2		Protocol ID, always 5004		--
0	3,4	10100	Pickup speed	1	rpm
0	5,6	-	internal		
1	1,2	160	Gen. power factor	0.001	
1	3,4,5,6	170	Av. Gen. Wye-Voltage	0.1	V
2	1,2	144	Gen. frequency	0.01	Hz
2	3,4,5,6	171	Av. Gen. Delta-Voltage	0.1	V
3	1,2	10310	Analog output 1	0,01	%
3	3,4,5,6	185	Av. Gen. Current	0.1	V
4	1,2	10311	Analog output 2	0,01	%
4	3,4,5,6	161	Meas. ground current	0.001	A
5	1,2	2112	Overspeed 1 latched	Mask: 8000h	Bit
		2113	Overspeed 2 latched	Mask: 4000h	Bit
		2162	Underspeed 1 latched	Mask: 2000h	Bit
		2163	Underspeed 2 latched	Mask: 1000h	Bit
		2652	Unintended stop latched	Mask: 0800h	Bit
		2457	Speed det. alarm latched	Mask: 0400h	Bit
		2504	Shutdown malfunct. latched	Mask: 0200h	Bit
		2603	GCB fail to close latched	Mask: 0100h	Bit
		2604	GCB fail to open latched	Mask: 0080h	Bit
		2623	MCB fail to close latched	Mask: 0040h	Bit
		2624	MCB fail to open latched	Mask: 0020h	Bit
		10017	CAN-Fault J1939 latched	Mask: 0010h	Bit
		3325	Start fail latched	Mask: 0008h	Bit
		2560	Mainten. days exceeded latched	Mask: 0004h	Bit
		2561	Mainten. hours exceeded latched	Mask: 0002h	Bit
		-	internal	Mask: 0001h	Bit
5	3,4,5,6	159	Calculated ground current	0.001	A
6	1,2	3064	GCB syn. timeout latched	Mask: 8000h	Bit
		3074	MCB syn. timeout latched	Mask: 4000h	Bit
		3084	GGB Timeout latched	Mask: 2000h	Bit
		4056	Charge alt. low volt latched	Mask: 1000h	Bit
		2944	Ph.rotation mismatch latched	Mask: 0800h	Bit
		10084	no data receive at RPDO3 at CAN Interface 1	Mask: 0400h	
		10083	no data receive at RPDO2 at CAN Interface 1	Mask: 0200h	
		10082	no data receive at RPDO1 at CAN Interface 1	Mask: 0100h	
		10086	no data receive at RPDO2 (function 1) at CAN Interface 2	Mask: 0080h	
		10085	no data receive at RPDO1 (function 1) at CAN Interface 2	Mask: 0040h	
		-	internal	Mask: 0020h	
		4073	Parameter Alignment	Mask: 0010h	
		4064	Missing members on CAN	Mask: 0008h	
		1714	EEPROM failure latched	Mask: 0004h	Bit
		15125	Red stop lamp latched	Mask: 0002h	Bit
		15126	Amber warning lamp latched	Mask: 0001h	Bit
6	3,4,5,6	111	Gen. current 1	0.001	A
7	1,2,3,4	112	Gen. current 2	0.001	A
7	5,6	-	internal		
8	1,2,3,4	113	Gen. current 3	0.001	A
8	5,6	-	internal		
9	1,2,3,4	135	Total gen. power	1	W
9	5,6	1912	Gen.overfreq. 1 latched	Mask: 8000h	Bit
		1913	Gen.overfreq. 2 latched	Mask: 4000h	Bit
		1962	Gen.underfreq. 1 latched	Mask: 2000h	Bit
		1963	Gen.underfreq. 2 latched	Mask: 1000h	Bit
		2012	Gen.overvolt. 1 latched	Mask: 0800h	Bit
		2013	Gen.overvolt. 2 latched	Mask: 0400h	Bit

CAN		Parameter ID	Description	Multiplier	Units
Data byte 0 (Mux)	Data byte				
		2062	Gen.undervolt. 1 latched	Mask: 0200h	Bit
		2063	Gen.undervolt. 2 latched	Mask: 0100h	Bit
		2218	Gen. overcurr. 1 latched	Mask: 0080h	Bit
		2219	Gen. overcurr. 2 latched	Mask: 0040h	Bit
		2220	Gen. overcurr. 3 latched	Mask: 0020h	Bit
		2262	Gen. Rv/Rd pow.1 latched	Mask: 0010h	Bit
		2263	Gen. Rv/Rd pow.2 latched	Mask: 0008h	Bit
		2314	Gen. Overload IOP 1 latched	Mask: 0004h	Bit
		2315	Gen. Overload IOP 2 latched	Mask: 0002h	Bit
		-	internal	Mask: 0001h	
10	1,2,3,4	136	Total gen. reactive power	1	var
10	5,6	2412	Unbal. load 1 latched	Mask: 8000h	Bit
		2413	Unbal. load 2 latched	Mask: 4000h	Bit
		3907	Gen. Asymmetry latched	Mask: 2000h	Bit
		3263	Ground fault 1 latched	Mask: 1000h	Bit
		3264	Ground fault 2 latched	Mask: 0800h	Bit
		3955	Gen. phase rot. misw. Latched	Mask: 0400h	Bit
		2924	Gen act.pwr mismatch Latched	Mask: 0200h	Bit
		3124	Gen. unloading fault Latched	Mask: 0100h	Bit
		4038	Inv.time ov.curr. Latched	Mask: 0080h	Bit
		2644	Timeout dead bus op. Latched	Mask: 0040h	Bit
		2362	Gen. Overload MOP 1 latched	Mask: 0020h	Bit
		2363	Gen. Overload MOP 2 latched	Mask: 0010h	Bit
		2337	Gen. overexcited 1 latched	Mask: 0008h	Bit
		2338	Gen. overexcited 2 latched	Mask: 0004h	Bit
		2387	Gen. underexcited 1 latched	Mask: 0002h	Bit
		2388	Gen. underexcited 2 latched	Mask: 0001h	Bit
11	1,2,3,4	108	Gen. voltage L1-L2	0.1	V
11	5,6	10131	control class latched	Mask: 0040h	Bit
			Alarm class F latched	Mask: 0020h	Bit
			Alarm class E latched	Mask: 0010h	Bit
			Alarm class D latched	Mask: 0008h	Bit
			Alarm class C latched	Mask: 0004h	Bit
			Alarm class B latched	Mask: 0002h	Bit
			Alarm class A latched	Mask: 0001h	Bit
12	1,2	4153	Idle mode monitoring (suppresses undervolt, underfreq,...)	Mask: 8000h	
			Idle mode activ	Mask: 4000h	
			Start without closing GCB	Mask: 2000h	
			Block closing GCB	Mask: 1000h	
			A manual START has been requested	Mask: 0800h	
			A manual STOP has been requested	Mask: 0400h	
			Cooldown is active	Mask: 0200h	
			Auxiliary services generally active	Mask: 0100h	
			Engine delay timer has expired	Mask: 0080h	
			Breaker delay timer has expired	Mask: 0040h	
			Engine shall run	Mask: 0020h	
			Critical mode is active in automatic mode	Mask: 0010h	
			Engine is released	Mask: 0008h	
			Auxiliary services prerun is active	Mask: 0004h	
			Auxiliary services postrun is active	Mask: 0002h	
			Lamp test is active	Mask: 0001h	
12	3,4,5,6	114	Gen. voltage L1-N	0.1	V
13	1,2,3,4	109	Gen. voltage L2-L3	0.1	V
13	5,6	-	internal		
14	1,2,3,4	115	Gen. voltage L2-N	0.1	V
14	5,6	-	internal		
15	1,2,3,4	110	Gen. voltage L3-L1	0.1	V
15	5,6	-	internal		
16	1,2,3,4	116	Gen. voltage 3-N	0.1	V
16	5,6	-	internal		
17	1,2,3,4	2522	Positive reactive gen power	0,01	Mvarh
17	5,6	-	internal		
18	1,2	5541	Frequency	0,01	Hz
18	3,4,5,6	5542	Active Power	0,1	kW

CAN		Parameter ID	Description	Multiplier	Units
Data byte 0 (Mux)	Data byte				
19	1,2,3,4	5640	Voltage	1	V
19	5,6	5641	Power Factor	0,001	
20	1,2	4154	Crank (Starter) is active	Mask: 8000h	
			Operating Magnet / Gas relay is active	Mask: 4000h	
			Preglow / Ignition is active	Mask: 2000h	
			Mains settling timer is running	Mask: 1000h	
			Emergency mode is currently active	Mask: 0800h	
			Emergency breaked by override	Mask: 0400h	
			Emergency Mains overfrequency	Mask: 0200h	
			Emergency Mains underfrequency	Mask: 0100h	
			Emergency Mains overvoltage	Mask: 0080h	
			Emergency Mains undervoltage	Mask: 0040h	
			Stopping Magnet is active	Mask: 0020h	
			internal	Mask: 0010h	
			The genset runs mains parallel	Mask: 0008h	
			internal	Mask: 0004h	
			internal	Mask: 0002h	
			Increment Start Counter	Mask: 0001h	
20	3,4	4155	Dead Band Controller Freq./Power raise	Mask: 8000h	
			Dead Band Controller Freq./Power raise	Mask: 4000h	
			Dead Band Controller Volt./ReactPow raise	Mask: 2000h	
			Dead Band Controller Volt./ReactPow lower	Mask: 1000h	
			GCB is closed	Mask: 0800h	
			MCB is closed	Mask: 0400h	
			internal	Mask: 0200h	
			Synchronization GCB is active	Mask: 0100h	
			Opening GCB is active	Mask: 0080h	
			Closing GCB is active	Mask: 0040h	
			Synchronization MCB is active	Mask: 0020h	
			Opening MCB is active	Mask: 0010h	
			Closing MCB is active	Mask: 0008h	
			Unloading generator is active	Mask: 0004h	
			Unloading mains is active	Mask: 0002h	
			Power limited prerun	Mask: 0001h	
20	5,6	4156	GGB is closed	Mask: 8000h	
			GGB is released	Mask: 4000h	
			Synchronization GGB is active	Mask: 2000h	
			Opening GGB is active	Mask: 1000h	
			Closing GGB is active	Mask: 0800h	
			Dead busbar closure request for GCB or MCB or GGB	Mask: 0400h	
			Active power load share is acitve	Mask: 0200h	
			Reactive power load share is acitve	Mask: 0100h	
			Generator is requested	Mask: 0080h	
			internal	Mask: 0040h	
			internal	Mask: 0020h	
			internal	Mask: 0010h	
			internal	Mask: 0008h	
			internal	Mask: 0004h	
			internal	Mask: 0002h	
			internal	Mask: 0001h	

Data Protocol 5005 (Mains Data)

CAN		Parameter ID	Description	Multiplier	Units
Data byte 0 (Mux)	Data byte				
0	1,2	-	Protocol ID (always 5005)		--
0	3,4	10100	Pickup speed	1	rpm
0	5,6	-	internal		
1	1,2	147	Mains frequency	0.01	Hz
1	3,4,5,6	173	Av. Mains Wye-Voltage	0.1	V
2	1,2	208	Mains power factor	0.001	
2	3,4,5,6	174	Av. Mains Delta-Voltage	0.1	V
3	1,2,3,4	207	Av. Mains Current	0.1	V
3	5,6	-	internal		
4	1,2	10111	Analog input 1	(changeable)	
4	3,4,5,6	134	Mains current L1	0.001	A
5	1,2	10112	Analog input 2	(changeable)	
5	3,4,5,6	140	Total mains power	1	W
6	1,2	10115	Analog input 3	(changeable)	
6	3,4,5,6	150	Total mains reactive power	1	var
7	1,2	2862	Mains ov.freq. 1 latched	Mask: 8000h	Bit
		2863	Mains ov.freq. 2 latched	Mask: 4000h	Bit
		2912	Mains un.freq. 1 latched	Mask: 2000h	Bit
		2913	Mains un.freq. 2 latched	Mask: 1000h	Bit
		2962	Mains ov.volt. 1 latched	Mask: 0800h	Bit
		2963	Mains ov.volt. 2 latched	Mask: 0400h	Bit
		3012	Mains un.volt. 1 latched	Mask: 0200h	Bit
		3013	Mains un.volt. 2 latched	Mask: 0100h	Bit
		3057	Mains phaseshift latched	Mask: 0080h	Bit
		3114	Mains decoupling latched	Mask: 0040h	Bit
		-	internal	Mask: 0020h	Bit
		-	internal	Mask: 0010h	Bit
		-	internal	Mask: 0008h	Bit
		3975	Mains phase rot. misw. Latched	Mask: 0004h	Bit
		-	internal	Mask: 0002h	Bit
		-	internal	Mask: 0001h	Bit
7	3,4	3217	Mains import power 1 latched	Mask: 8000h	Bit
		3218	Mains import power 2 latched	Mask: 4000h	Bit
		3241	Mains export power 1 latched	Mask: 2000h	Bit
		3242	Mains export power 2 latched	Mask: 1000h	Bit
		2985	Mains overexcited 1 latched	Mask: 0800h	Bit
		2986	Mains overexcited 2 latched	Mask: 0400h	Bit
		3035	Mains underexcited 1 latched	Mask: 0200h	Bit
		3036	Mains underexcited 2 latched	Mask: 0100h	Bit
		-	internal	Mask: 0080h	Bit
		2934	Mns act.pwr mismatch latched	Mask: 0040h	Bit
		-	internal	Mask: 0020h	Bit
		-	internal	Mask: 0010h	Bit
		-	internal	Mask: 0008h	Bit
		-	internal	Mask: 0004h	Bit
		-	internal	Mask: 0002h	Bit
		-	internal	Mask: 0001h	Bit
7	5,6	-	internal		
8	1,2,3,4	118	Mains voltage L1-L2	0.1	V
8	5,6	-	internal		
9	1,2,3,4	121	Mains voltage L1-N	0.1	V
9	5,6	-	internal		
10	1,2,3,4	119	Mains voltage L2-L3	0.1	V
10	5,6	-	internal		
11	1,2,3,4	122	Mains voltage L2-N	0.1	V
11	5,6	-	internal		
12	1,2,3,4	120	Mains voltage L3-L1	0.1	V
12	5,6	-	internal		
13	1,2,3,4	123	Mains voltage L3-N	0.1	V
13	5,6	-	internal		

Data Protocol 6000 (Load Share Message)

General

The load share message contains all data, which is required for load/var sharing, load-dependent start/stop and dead bus detection.
Further data, which are to be exchanged between the control units concern time synchronization and parameter alignment. Parameter alignment is intended for those parameters, which must be configured identically for all units participating in load sharing, to ensure a proper operation of load sharing or load-dependent start/stop.

In order to lower the bus load, the messages are divided into "fast", "normal", and "slow" refreshed data. The mux is identified accordingly with "F", "N", and "S" (refer to the following tables). The load share message contains one fast, two normal, and four slow messages, which are made up as shown in Table 3-6.

Timing

The time interval between two fast messages (T_{Fast} , i.e. the time for refreshing a fast message) is configured with the parameter "Transfer rate LS fast message" (parameter 9921 on page 23). The time intervals between refreshing a normal or slow messages depend on this parameter as well according to the following sequence:

S0 – F – N0 – F – N1 – F – S1 – F – N0 – F – N1 – F – S2 – F – N0 – F – N1 – F – S3 – F – N0 – F – N1 – F

T_{Fast} = time interval between refreshing the fast message
T_{Normal} = time interval between refreshing a normal message = 3 x T_{Fast}
T_{Slow} = time interval between refreshing a slow message = 12 x T_{Fast}

Example:
The parameter "Transfer rate LS fast message" (parameter 9921 on page 23) is configured to "0.10 s".
The sequence of the sent messages for T_{Fast} = 100 ms (i.e. 0.10 s) is shown in Table 3-6. This means that a new message is sent every 50 ms.

Time [ms]	0	50	100	150	200	250	300	350	400	450	500	550
Sent message	S0	F	N0	F	N1	F	S1	F	N0	F	N1	F
Mux #	0	3	1	3	2	3	4	3	1	3	2	3
Time [ms]	600	650	700	750	800	850	900	950	1000	1050	1100	1150
Sent message	S2	F	N0	F	N1	F	S3	F	N0	F	N1	F
Mux #	5	3	1	3	2	3	6	3	1	3	2	3

Table 3-6: Load share message - example

The maximum length of the CAN bus load share line depends on this parameter as well. The values in Table 3-7 are valid for 32 participants and a bus load of approx. 30 %.

T_{Fast} [ms]	T_{Normal} [ms]	T_{Slow} [ms]	Baud rate	Distance
100	300	1200	250 kBaud	250 m
200	600	2400	125 kBaud	500 m
300	900	3800	50 kBaud	1000 m

Table 3-7: Load share line - max. length

Load share bus communication - "fast" refreshed data				
Mux	**Byte**	**Bit**	**Function**	**Remark**
F	0		**3**	Mux identifier
	1		Generator real load capacity utilization rate, L-Byte	Integer [‰], unsigned
	2		Generator real load capacity utilization rate, H-Byte	
	3		Generator reactive load capacity utilization rate, L-Byte	Integer [‰], unsigned
	4		Generator reactive load capacity utilization rate, H-Byte	
	5	0	Active power load sharing is enabled	
		1	Reactive power load sharing is enabled	
		2	GCB is closed	
		3	MCB is closed	
		4	GGB (generator group breaker) is closed	
		5	Dead bus closure request is active	Dead bus detection
		6	Mains settling time is running	Back synchronization to mains
		7	Shutdown alarm is active (alarm class C,D,E,F)	
	6	0-4	Bus segment / node	Max. 32 nodes possible
		5	Not used	
		6	LDSS: add-on request enabled	Load dependent start / stop
		7	LDSS: add-off request enabled (reserved)	Load dependent start / stop
	7		Not used	

Load share bus communication - "normal" refreshed data				
Mux	**Byte**	**Bit**	**Function**	**Remark**
N0	0		1	Mux identifier
	1		Generator real load, L-Byte, L-Word	Long [W]
	2		Generator real load, H-Byte, L-Word	
	3		Generator real load, L-Byte, H-Word	
	4		Generator real load, H-Byte, H-Word	
	5	0-3	Real load control state	2: Static 3: Isochronous 4: Base load control 5: Export/import control 10:Load share 0, 1, 6, 7, 8, 9, 11, … : internal
		4-7	Reactive load control state	2: Static 3: Isochronous 4: Reactive load control 5: Import/export reactive load 10:Reactive load share 0, 1, 6, 7, 8, 9, 11, … : internal
	6	0-3	Engine state	1: Locked out 2: Off 3: Preglow 4: Crank 5: Run 6: Cool down 7: Spin down 8: Start pause 9: Idle 0, 10, 11, … : internal
		4,5	Operating mode	0: Not available 1: STOP 2: MANUAL 3: AUTOMATIC
		6	Generator request	Generator is in AUTOMATIC mode and able to produce rated active power
		7	Not used	
	7		Not used	

Load share bus communication - "normal" refreshed data				
Mux	**Byte**	**Bit**	**Function**	**Remark**
N1	0		**2**	Mux identifier
	1		Generator reactive load, L-Byte, L-Word	Long [var]
	2		Generator reactive load, H-Byte, L-Word	
	3		Generator reactive load, L-Byte, H-Word	
	4		Generator reactive load, H-Byte, H-Word	
	5	0	Generator voltage and frequency ok	
		1	Busbar voltage and frequency ok	
		2	Mains voltage and frequency ok	
		3	Fourth system voltage and frequency ok	
		4	Not used	
		5	Not used	
		6	Not used	
		7	Not used	
	6		Not used	
	7		Not used	

Load share bus communication - "slow" refreshed data				
Mux	**Byte**	**Bit**	**Function**	**Remark**
S0	0		0	Mux identifier
	1		Protocol-Identifier	
	2			
	3		Generator rated real power, L-Byte, L-Word	Long [0.1 kW]
	4		Generator rated real power, H-Byte, L-Word	
	5		Generator rated real power, L-Byte, H-Word	
	6		Generator rated real power, H-Byte, H-Word	
	7		Not used	
S1	0		4	Mux identifier
	1		Generator rated reactive power, L-Byte, L-Word	Long [0.1 kvar]
	2		Generator rated reactive power, H-Byte, L-Word	
	3		Generator rated reactive power, L-Byte, H-Word	
	4		Generator rated reactive power, H-Byte, H-Word	
	5		Not used	
	6	0-4	Priority	Up to 32
		5-7	Not used	
	7		Not used	
S2	0		5	Mux identifier
	1		Operating hours L-Byte, L-Word	Long [h]
	2		Operating hours H-Byte, L-Word	
	3		Operating hours L-Byte, H-Word	
	4		Operating hours H-Byte, H-Word	
	5	0	Alarm class A occurred	
		1	Alarm class B occurred	
		2	Alarm class C occurred	
		3	Alarm class D occurred	
		4	Alarm class E occurred	
		5	Alarm class F occurred	
		6	Warning alarm class occurred	
		7	Not used	
	6		Not used	
	7		Not used	
S3	0		6	Mux identifier
	1		Remaining days before maintenance, L-Byte	Integer [d]
	2		Remaining days before maintenance, H-Byte	
	3		Remaining operating hours before maintenance, L-Byte	Integer [h]
	4		Remaining operating hours before maintenance, H-Byte	
	5		Checksum parameters L-Byte	Load share and load-dependent start / stop parameters
	6		Checksum parameters H-Byte	
	7		Not used	

Additional Data Protocol Parameters

Remote Control Word 1 - Object 21F7h (Parameter ID 503)

This object is required for remote control. The data type is UNSIGNED16.

The internal parameter 503 of the easYgen must be set to react on the remote control instructions. This is performed by sending rising signals for the respective bits (refer to Figure 3-38 for the priority of start and stop signals).

Parameter no.	Object ID	Name		Unit	Data type	Note
503	21F7h	Control word 1		Bit field	unsigned16	
		Bit 15	Not used			
		Bit 14	Not used			
		Bit 13	Not used			
		Bit 12	Not used			
		Bit 11	Not used			
		Bit 10	Not used			
		Bit 9	Not used			
		Bit 8	Not used			
		Bit 7	Not used			
		Bit 6	Not used			
		Bit 5	Not used			
		Bit 4	Ext. Acknowledge (rising edge) Must be set twice to acknowledge			To acknowledge, a 0 must be written and then a 1
		Bit 3	Must always be set to 0			
		Bit 2	Must always be set to 0			
		Bit 1	Stop bit (rising edge)			To stop, a 0 must be written and then a 1
		Bit 0	Start bit (rising edge)			To start, a 0 must be written and then a 1

Table 3-8: Remote control telegram

Bit 0 Start bit
With the rising edge of the bit, the easYgen activates the remote request command (*LogicsManager* input command variable 04.13). The condition of the start command will be stored and may be used as command variable for the *LogicsManager*.

Bit 1 Stop bit
With the rising edge of the bit, the easYgen deactivates the remote request command (*LogicsManager* input command variable 04.13). The condition of the start command will be stored and may be used as command variable for the *LogicsManager*.

Bit 4 "Reset alarms"
This bit controls the *LogicsManager* input command variable 04.14. The remote acknowledge bit must be set and reset twice to acknowledge an alarm completely. The first rising edge disables the horn and the second rising edge acknowledges the alarm.

Remote start /stop: The command variable "04.13 Remote request" changes to "1" (high) if the start bit is enabled and changes back to "0" (low) if the stop bit is enabled.

Ext. Acknowledge: The command variable "04.14 Remote acknowledge" is the reflection of the control bit. The easYgen deactivates the horn with the first change from "0" to "1" of the logical output "External acknowledge", and acknowledges all alarm messages, which have occurred and are no longer active, with the second change from "0" to "1".

Figure 3-38 shows the reaction of the command variable on the various status changes of the bits:

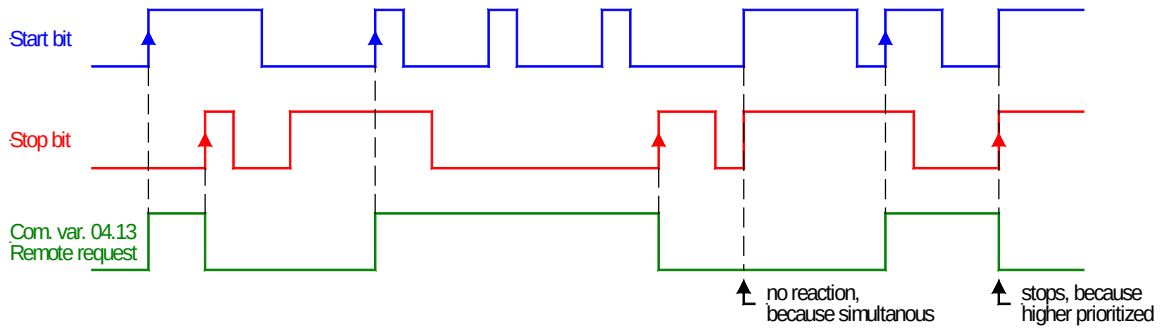

Figure 3-38: Remote control - start/stop priority

ATTENTION

The easYgen does NOT react on the disabling of the start bit, but only on the enabling of the stop bit. This has the advantage that it is not required to maintain the connection established for the whole time in case of a remote start via a modem.

Remote Control Word 2 - Object 21F8h (Parameter ID 504)

This object is required for remote control. The data type is UNSIGNED16.

Bit 15 = 1	
Bit 14 = 1	
Bit 13 = 1	
Bit 12 = 1	
Bit 11 = 1	
Bit 10 = 1	
Bit 9 = 1	
Bit 8 = 1	
Bit 7 = 1	Request active power set point 2 – this bit activates the *LogicsManager* input [04.40] "Remote power set point 2" and is dedicated for switching from active power set point 1 to active power set point 2
Bit 6 = 1	Request power factor set point 2 – this bit activates the *LogicsManager* input [04.39] "Remote PF set point 2" and is dedicated for switching from power factor set point 1 to power factor set point 2
Bit 5 = 1	Request frequency set point 2 – this bit activates the *LogicsManager* input [04.38] "Remote frequency set point 2" and is dedicated for switching from frequency set point 1 to frequency set point 2
Bit 4 = 1	Request voltage set point 2 – this bit activates the *LogicsManager* input [04.37] "Remote voltage set point 2" and is dedicated for switching from voltage set point 1 to voltage set point 2
Bit 3 = 1	
Bit 2 = 1	
Bit 1 = 1	
Bit 0 = 1	

Remote Active Power Set Point - Object 21FBh (Parameter ID 507)

This object is required to transmit the active power set point for active power control. The data type is INTEGER32. The value is scaled in [kW * 10].
Example: 100 kW = 1000 = 03E8h

Remote Power Factor Set Point - Object 21FCh (Parameter ID 508)

This object is required to transmit the power factor set point for power factor control. The data type is INTEGER16. The valid range for this value is [-710 to 1000 to 710].
Example: PF (cosphi) = c0.71 (capacitive) = -710 = FD3Ah
PF (cosphi) = 1.00 = 1000 = 03E8h
PF (cosphi) = i0.71 (inductive) = 710 = 02C6h

Remote Frequency Set Point - Object 21FDh (Parameter ID 509)

This object is required to transmit the frequency set point for frequency control. The data type is UNSIGNED16. The value is scaled in [Hz * 100].
Example: 50.00 Hz = 5000 = 1388h

Remote Voltage Set Point - Object 21FEh (Parameter ID 510)

This object is required to transmit the voltage set point for voltage control. The data type is UNSIGNED32. The value is scaled in [V].
Example: 400 V = 400 = 190h
10000 V = 10000 = 2710h

Remote IKD 1 DI 1 to 8 Request - Object 3F4Bh (Parameter ID 8011)

This object is required to receive the state of the external discrete inputs 1 to 8 of an IKD 1. The data type is UNSIGNED8. This object cannot be read.

Byte 1	Bit 1 to 7	Always 0
	Bit 0	Always 1
Byte 2	Bit 7	External discrete input 8 [DIex08]
	Bit 6	External discrete input 7 [DIex07]
	Bit 5	External discrete input 6 [DIex06]
	Bit 4	External discrete input 5 [DIex05]
	Bit 3	External discrete input 4 [DIex04]
	Bit 2	External discrete input 3 [DIex03]
	Bit 1	External discrete input 2 [DIex02]
	Bit 0	External discrete input 1 [DIex01]
Byte 3 to 8	Bit 0 to 7	Always 0

Remote IKD 1 DI 9 to 16 Request - Object 3F4Ch (Parameter ID 8012)

This object is required to receive the state of the external discrete inputs 9 to 16 of an IKD 1. The data type is UNSIGNED8. This object cannot be read.

Byte 1	Bit 1 to 7	Always 0
	Bit 0	Always 1
Byte 2	Bit 7	External discrete input 16 [DIex16]
	Bit 6	External discrete input 15 [DIex15]
	Bit 5	External discrete input 14 [DIex14]
	Bit 4	External discrete input 13 [DIex13]
	Bit 3	External discrete input 12 [DIex12]
	Bit 2	External discrete input 11 [DIex11]
	Bit 1	External discrete input 10 [DIex10]
	Bit 0	External discrete input 9 [DIex09]
Byte 3 to 8	Bit 0 to 7	Always 0

Remote External DI Request - Object 3F4Dh (Parameter ID 8013)

This object is required to receive the state of the external discrete inputs 1 to 16 (e.g. of a Phoenix expansion card). The data corresponds to the data in the objects 8011 and 8002 and the data type is UNSIGNED16.

Bit 15	External discrete input 16 [DIex16]
Bit 14	External discrete input 15 [DIex15]
Bit 13	External discrete input 14 [DIex14]
Bit 12	External discrete input 13 [DIex13]
Bit 11	External discrete input 12 [DIex12]
Bit 10	External discrete input 11 [DIex11]
Bit 9	External discrete input 10 [DIex10]
Bit 8	External discrete input 9 [DIex09]
Bit 7	External discrete input 8 [DIex08]
Bit 6	External discrete input 7 [DIex07]
Bit 5	External discrete input 6 [DIex06]
Bit 4	External discrete input 5 [DIex05]
Bit 3	External discrete input 4 [DIex04]
Bit 2	External discrete input 3 [DIex03]
Bit 1	External discrete input 2 [DIex02]
Bit 0	External discrete input 1 [DIex01]

Remote IKD 1 Control - Object 34F0h (Parameter ID 8000)

This object is required to assemble a message to the IKD 1(s). The bits in this object are always 0.

Remote IKD 1 DO 1 to 8 Control - Object 34F1h (Parameter ID 8001)

This object is required to control the external outputs (relays) 1 to 8 of an IKD 1. The data type is UNSIGNED16.

Bit 15	External discrete output 8 [Rex08]
Bit 14	External discrete output 7 [Rex07]
Bit 13	External discrete output 6 [Rex06]
Bit 12	External discrete output 5 [Rex05]
Bit 11	External discrete output 4 [Rex04]
Bit 10	External discrete output 3 [Rex03]
Bit 9	External discrete output 2 [Rex02]
Bit 8	External discrete output 1 [Rex01]
Bit 7	Always 0
Bit 6	Always 0
Bit 5	Always 0
Bit 4	Always 0
Bit 3	Always 0
Bit 2	Always 0
Bit 1	Always 0
Bit 0	Always 0

Remote IKD 1 DO 9 to 16 Control - Object 34F2h (Parameter ID 8002)

This object is required to control the external outputs (relays) 9 to 16 of an IKD 1. The data type is UNSIGNED16.

Bit 15	External discrete output 16 [Rex16]
Bit 14	External discrete output 15 [Rex15]
Bit 13	External discrete output 14 [Rex14]
Bit 12	External discrete output 13 [Rex13]
Bit 11	External discrete output 12 [Rex12]
Bit 10	External discrete output 11 [Rex11]
Bit 9	External discrete output 10 [Rex10]
Bit 8	External discrete output 9 [Rex09]
Bit 7	Always 0
Bit 6	Always 0
Bit 5	Always 0
Bit 4	Always 0
Bit 3	Always 0
Bit 2	Always 0
Bit 1	Always 0
Bit 0	Always 0

Remote External DO Control - Object 34F5h (Parameter ID 8005)

This object is required to control the external outputs (relays) 1 to 16 (e.g. of a Phoenix expansion card). The data corresponds to the data in the objects 8001 and 8002 and the data type is UNSIGNED16.

Bit 15	External discrete output 16 [Rex16]
Bit 14	External discrete output 15 [Rex15]
Bit 13	External discrete output 14 [Rex14]
Bit 12	External discrete output 13 [Rex13]
Bit 11	External discrete output 12 [Rex12]
Bit 10	External discrete output 11 [Rex11]
Bit 9	External discrete output 10 [Rex10]
Bit 8	External discrete output 9 [Rex09]
Bit 7	External discrete output 8 [Rex08]
Bit 6	External discrete output 7 [Rex07]
Bit 5	External discrete output 6 [Rex06]
Bit 4	External discrete output 5 [Rex05]
Bit 3	External discrete output 4 [Rex04]
Bit 2	External discrete output 3 [Rex03]
Bit 1	External discrete output 2 [Rex02]
Bit 0	External discrete output 1 [Rex01]

We appreciate your comments about the content of our publications.

Please send comments to: stgt-documentation@woodward.com

Please include the manual number from the front cover of this publication.

Woodward GmbH
Handwerkstrasse 29 - 70565 Stuttgart - Germany
Phone +49 (0) 711-789 54-0 · Fax +49 (0) 711-789 54-100
sales-stuttgart@woodward.com

Homepage

http://www.woodward.com/power

Woodward has company-owned plants, subsidiaries, and branches, as well as authorized distributors and other authorized service and sales facilities throughout the world.

Complete address/phone/fax/e-mail information for all locations is available on our website (www.woodward.com).

2012/06/Stuttgart