37393B

easYgen-1000
Interfaces

Interface Description
Software Version 2.1xxx

EASY GEN 1000

Manual 37393B

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OUT-OF-DATE PUBLICATION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	07-02-06	TP	Release based on 37262D
A	08-05-21	TP	Minor corrections
B	08-07-02	TP	Parameter descriptions moved to Configuration Manual

Contents

Chapter 1.
General Information

Related Documents

Type		English	German
easYgen-1000 Series			
easYgen-1000 - Installation		37390	GR37390
easYgen-1000 - Configuration		37391	GR37391
easYgen-1000 - Operation		37392	GR37392
easYgen-1000 - Interfaces	this manual ⇨	37393	GR37393
easYgen-1000 - Application		37394	GR37394
Additional Manuals			
IKD 1 - Manual		37135	GR37135
Discrete expansion board with 8 discrete inputs and 8 relay outputs that can be coupled via the CAN bus to the control unit. Evaluation of the discrete inputs as well as control of the relay outputs is done via the control unit.			
LeoPC1 - User Manual		37146	GR37146
PC program for visualization, configuration, remote control, data logging, language upload, alarm and user management, and management of the event recorder. This manual describes the set up of the program and interfacing with the control unit.			
LeoPC1 - Engineering Manual		37164	GR37164
PC program for visualization, configuration, remote control, data logging, language upload, alarm and user management, and management of the event recorder. This manual describes the configuration and customization of the program.			
GW 4 - Manual		37133	GR37133
Gateway for transferring the CAN bus to any other interface or bus.			
ST 3 - Manual		37112	GR37112
Control to govern the air fuel ratio of a gas engine. The ratio will be directly measured though a Lambda probe and controlled to a configured value.			

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 37391.

Interface Overview

The easYgen-1000 provides the following communication interfaces:

- **Serial interface (DPC)**
  LeoPC1 or Modbus Protocol
- **CAN interface**
  CANopen, CAN CAL, or CAN J1939 Protocol

NOTE

ECU data in J1939 format may be sent simultaneously with data in CANopen format on the same bus. However, the baud rate must be the same.

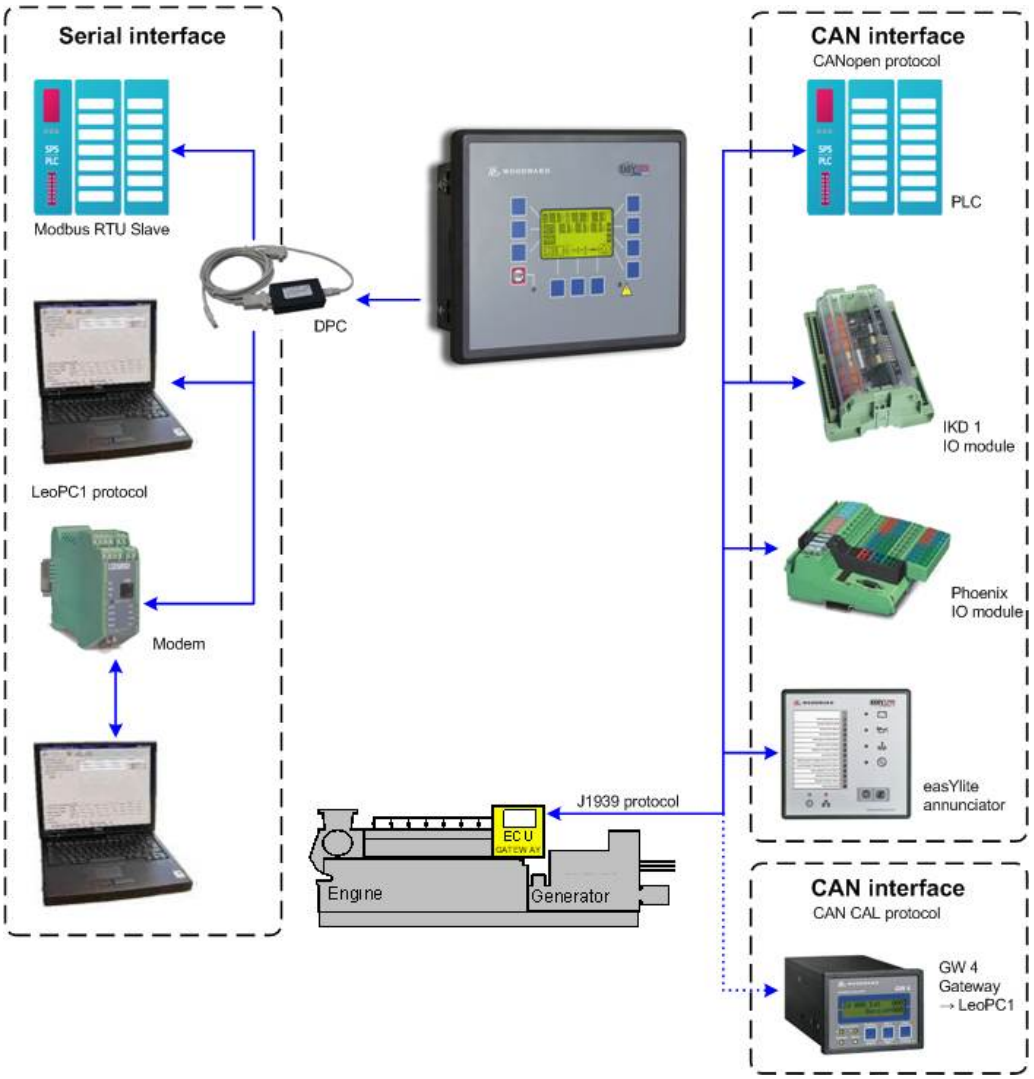


Figure 1-1: Interface overview

WARNING

When connecting the direct configuration interface, the Woodward DPC with RJ45 connector must be used. Failure to do so may destroy the unit.

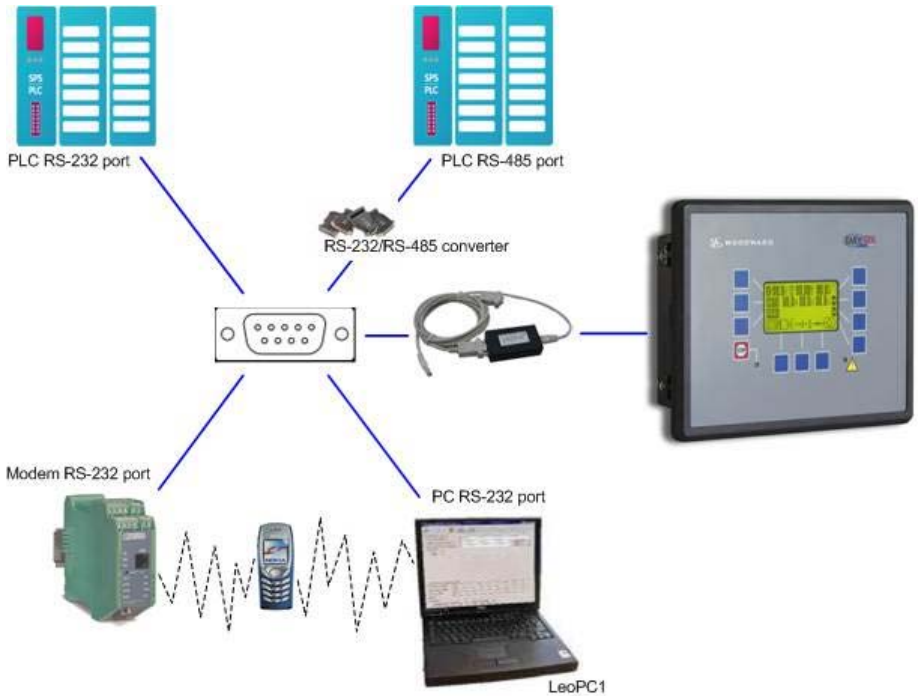


Figure 1-2: Interface overview - serial interface

Modbus Half/Full Duplex Application

NOTE

Not every RS-232/RS-485 converter is suitable for this application when using more than one easYgen because the converter has to control the data of the connected easYgen units.

The following model of the company Amplicon meets this requirements: Magic 485F9

This converter has also been verified for single unit applications.

Please refer to the manufacturer's homepage for further information about the suitable Amplicon RS-485/RS-232 converter under http://www.amplicon.co.uk.

Full-Duplex Wiring on RS-485

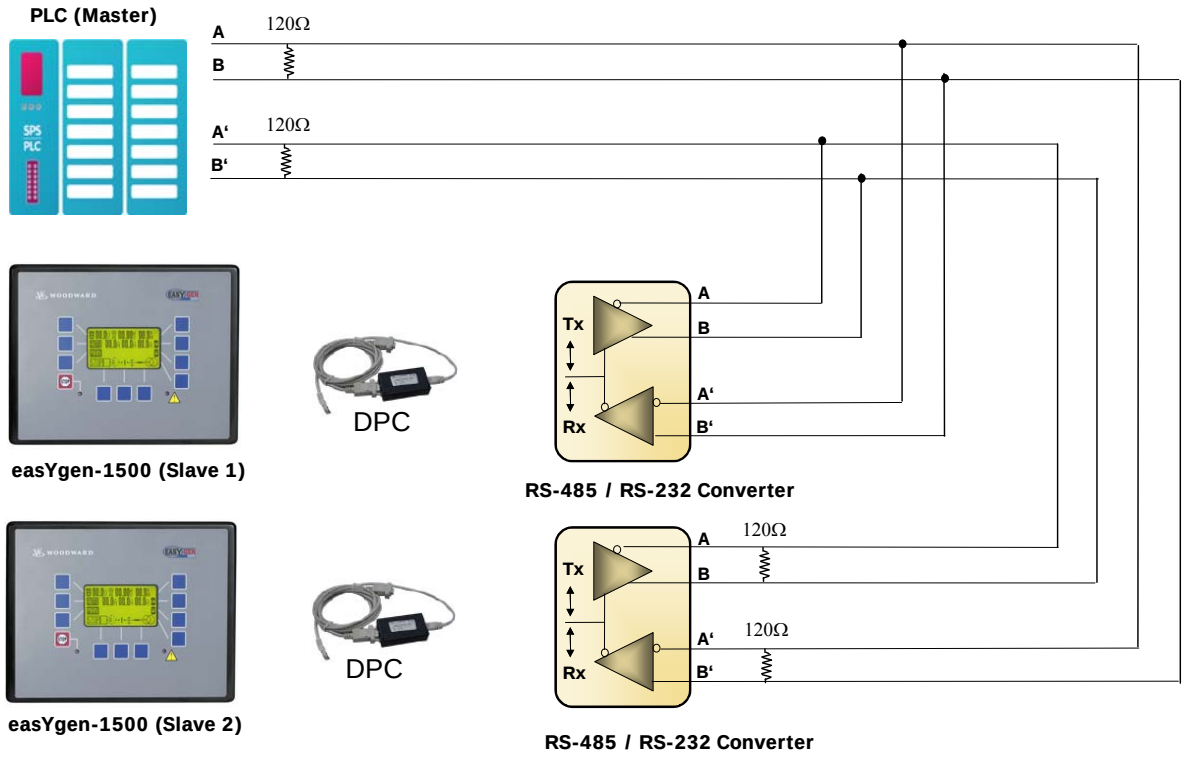


Figure 1-3: Interface overview - serial interface Modbus full-duplex

Half-Duplex Wiring on RS-485

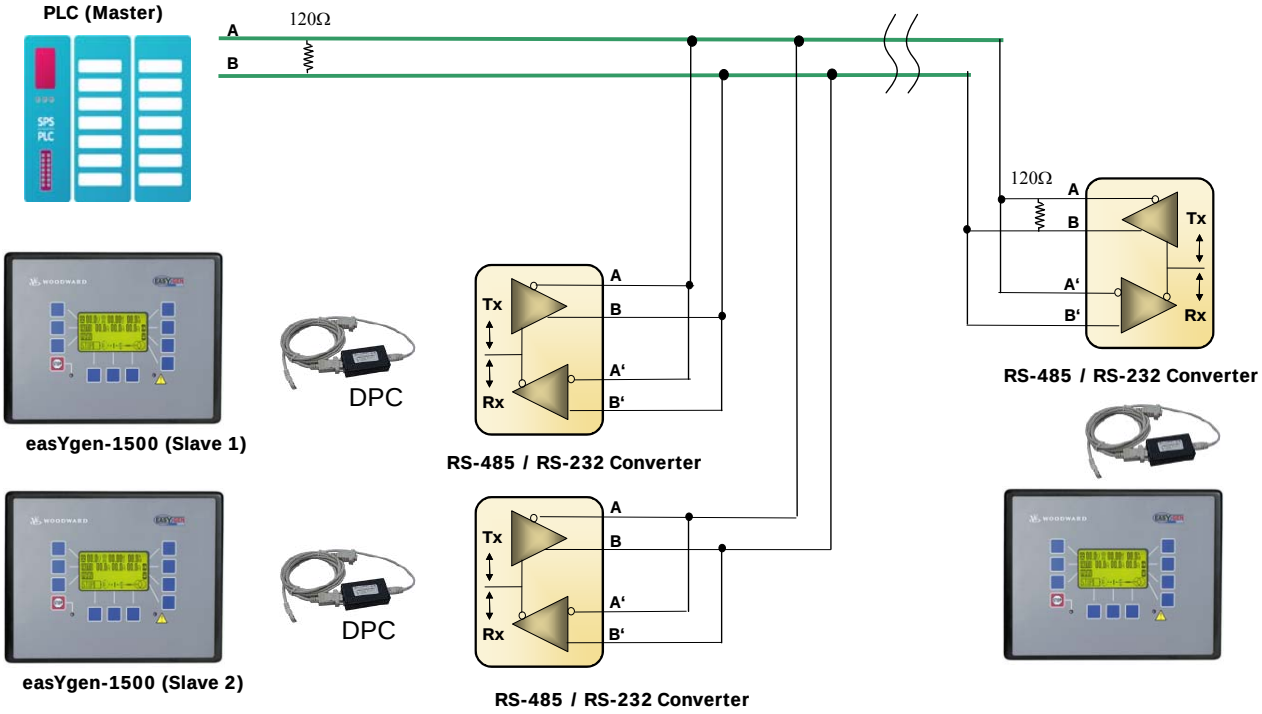


Figure 1-4: Interface overview - serial interface Modbus half-duplex

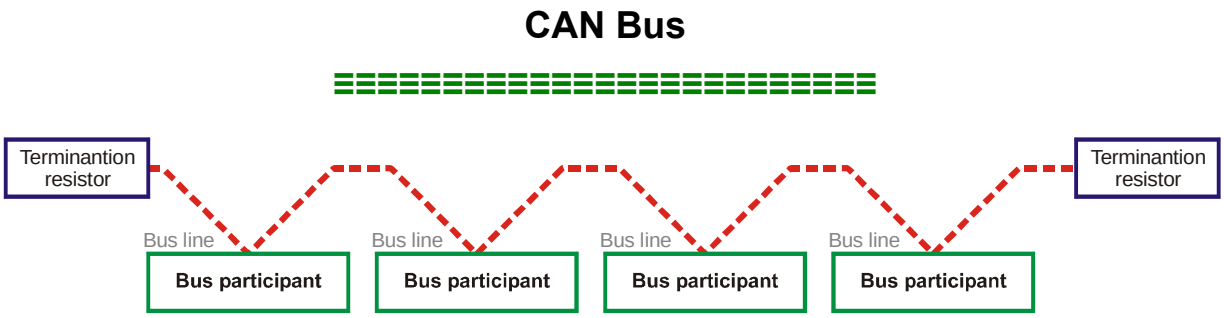


Figure 1-5: CAN bus topology - CAN CAL

Characteristics of the CAN interface used by Woodward:

- Standard: Compatible with ISO 11898
- Electrically isolated: Isolation voltage 1,500 V_{DC}

NOTE

Please note that the CAN bus must be terminated with an impedance which corresponds to the wave impedance of the cable (e.g. 120 Ohm, ¼ W). The CAN bus is terminated between CAN-H and CAN-L.

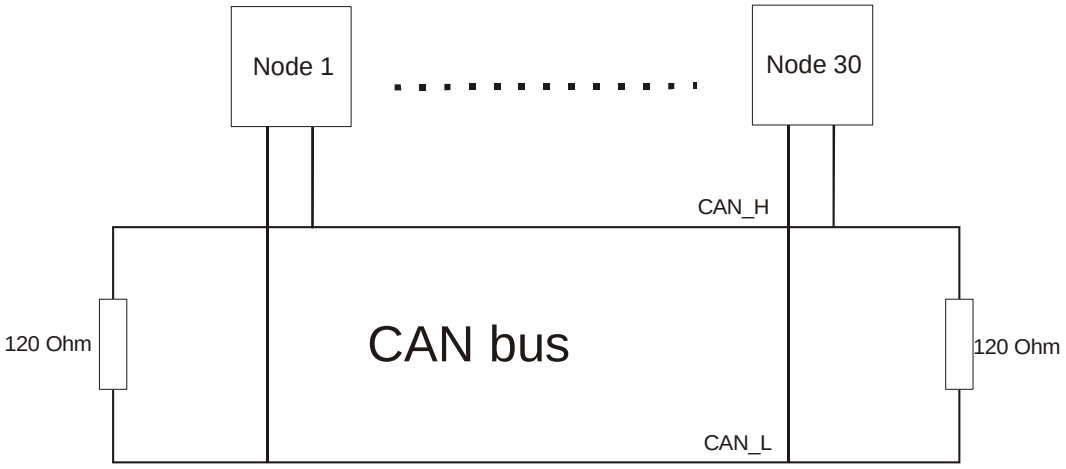


Figure 1-6: Interface - The CAN bus loop

Chapter 2.
Data Telegrams

Interface Monitoring

It is possible to monitor the interface for received data. Refer to the configuration manual for more information about this monitoring function.

Transmit Telegram

The transmit telegram provides all measuring and status data of the easYgen. The data have different addresses and will be transmitted in the respective format depending on the selected interface.

Modbus

Data transmission in Modbus format is performed in the order of the transmit telegram (refer to Appendix A: Transmission Telegram on page 41). The data addresses may be taken from the respective column of the transmit telegram.

CAN (CAL)

The easYgen sends its data via cyclic CAN messages. If a GW 4 is used, the baud rate must be configured to 125 kBaud.

NOTE

Instead of using a GW 4, a CAN to USB (or RS-232) converter may be used.

CANopen

Using the mapped objects, which are described in detail starting on page 25, enables you to send data by setting the object ID 2C76h on the basis of the CANopen protocol.
This document contains tables of further mapped objects, which may be configured in refer to Appendix A: Transmission Telegram on page 41.

NOTE

When using the mapped objects listed in the appendix instead of the complete transmit telegram, the refresh rate of the single messages may be reduced.

Receive Telegram

Starting with V2.xxx the genset may be started, stopped, and acknowledged via the interface.

In order to execute the desired command, a rise of the pulse of the respective signal from Low to High is required.
It must be observed that the combination of remote start and remote stop create the logical command variable 04.13. This command variable may be used as input of the *LogicsManager* in the easYgen.

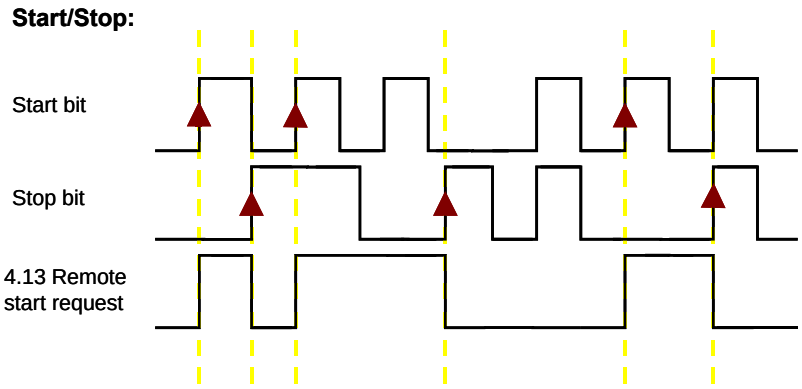


Figure 2-1: Data telegrams - remote control signals

An acknowledgement command must be sent twice to acknowledge a fault, which is not present anymore. The first rise of the pulse resets the horn. The second rise of the pulse acknowledges the unit, if the fault is not present anymore.

NOTE

Please note that the respective remote control parameters must be configured in the *LogicsManager* of the unit. Refer to the application manual 37394 for more detailed information about this.

Modbus

It is possible to remote control the easYgen using the bits 0 to 4 of control word 1 on address 503. The Remote Control Telegram in Appendix A on page 53 is valid for both, CANopen as well as Modbus, and indicates the arrangement of the remote control bits.

CAN (CAL)

The Woodward LeoPC1 software may be used to remote control the easYgen via a connected PC. After selecting the desired remote control command, the remote control command must be confirmed by selecting the "Set" button.

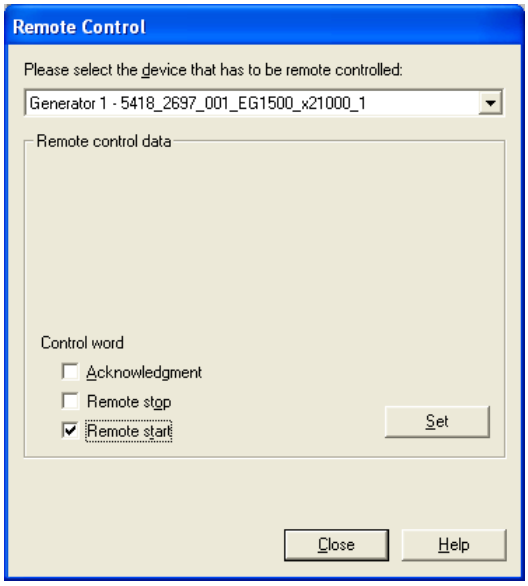


Figure 2-2: Data telegrams - remote control via CAN

CANopen

It is possible to remote control the easYgen using the bits 0 to 4 of control word 1 on address 503. The Remote Control Telegram in Appendix A on page 53 is valid for both, CANopen as well as Modbus, and indicates the arrangement of the remote control bits.

Chapter 3.
Serial Interface

Overview

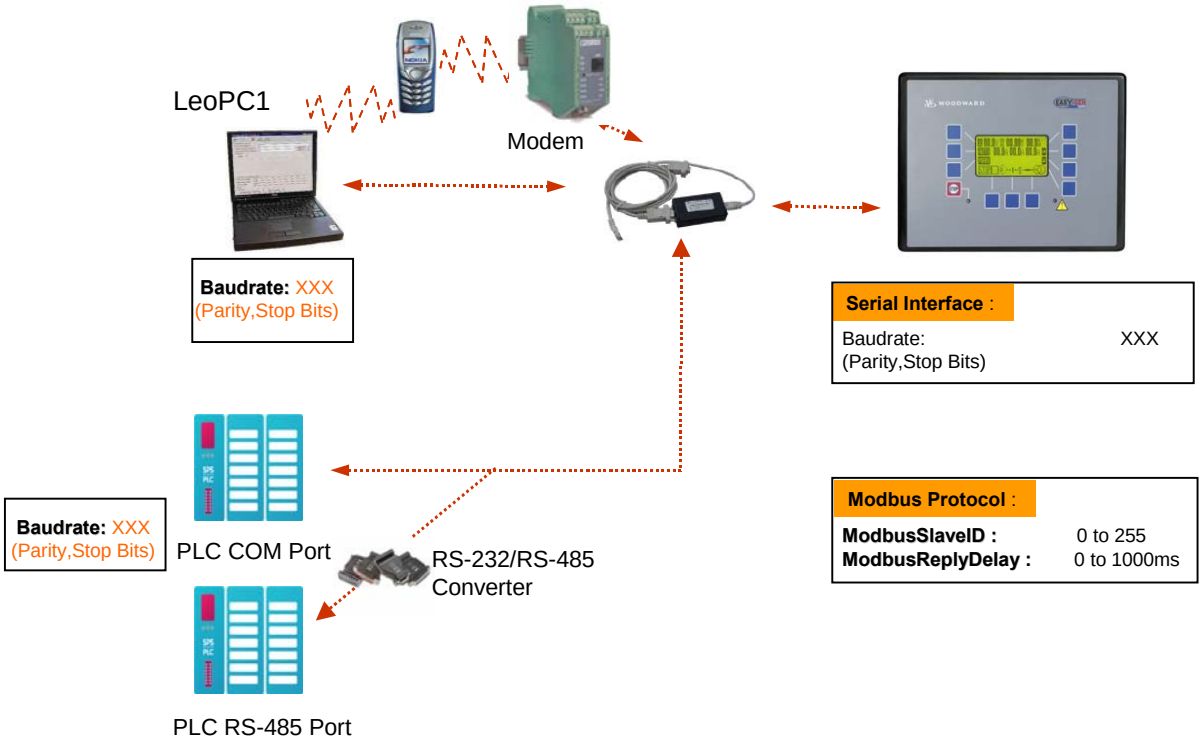


Figure3-1: Serial interface - overview

Modbus RTU Slave

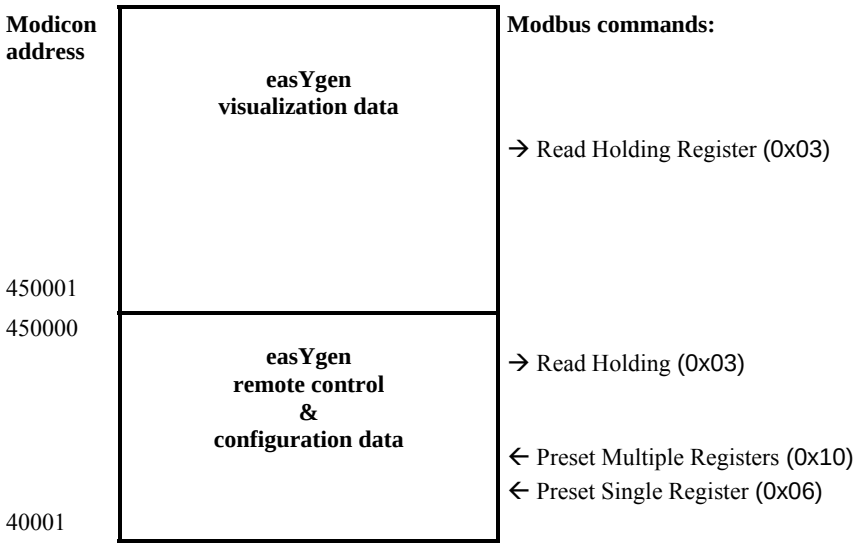
General Information

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 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

Modbus Addressing and Data Model

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 may 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 figure below).



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		--
450002	Generator: Voltage V_{L12}	0.1	V
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
450184	Failure codes from MTU ADEC ECU	-	-

Table 3-1: Modbus - address range block read

NOTE

Table 3-1 is only an excerpt of the data protocol. It conforms to the data protocol, that is also used by CAN bus. Refer to Appendix A: Transmission Telegram on page 41 for the complete protocol.

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

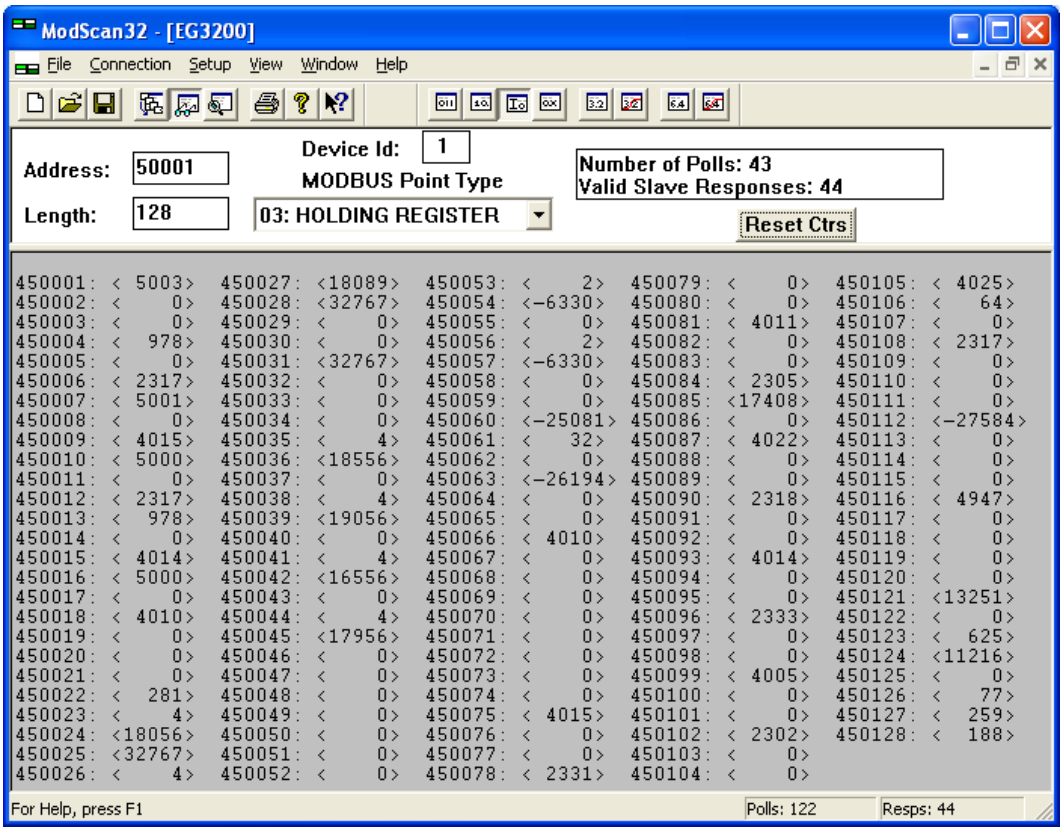


Figure 3-2: Modbus - visualization configurations

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 3-2: 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 3-3 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 3-3: Modbus - data types

NOTE

The parameters of the following examples are an excerpt of the parameter list in the appendix of the Configuration Manual 37391. 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.

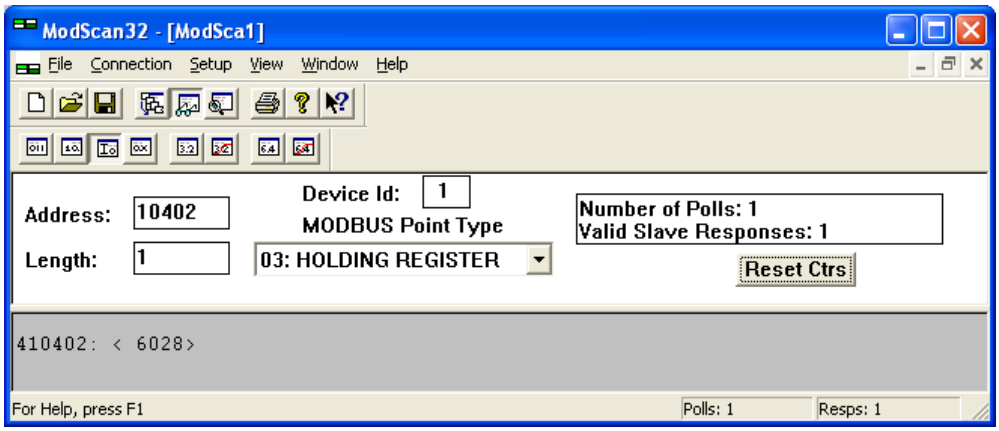


Figure 3-3: Modbus - configuration example 1

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.

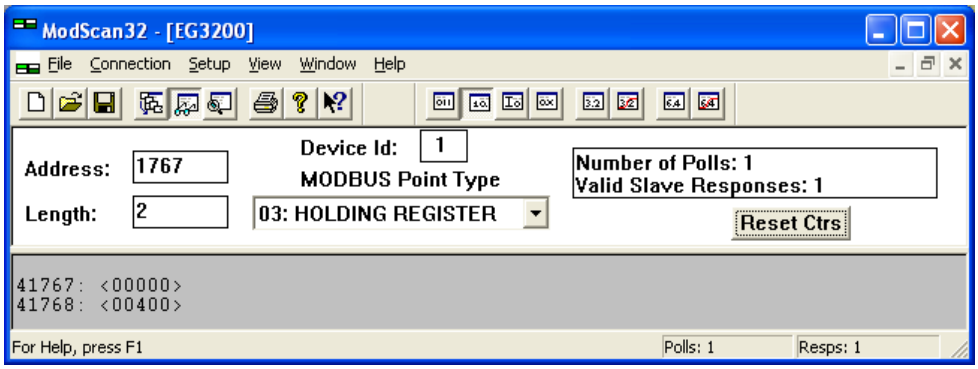


Figure 3-4: Modbus - configuration example 2

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".

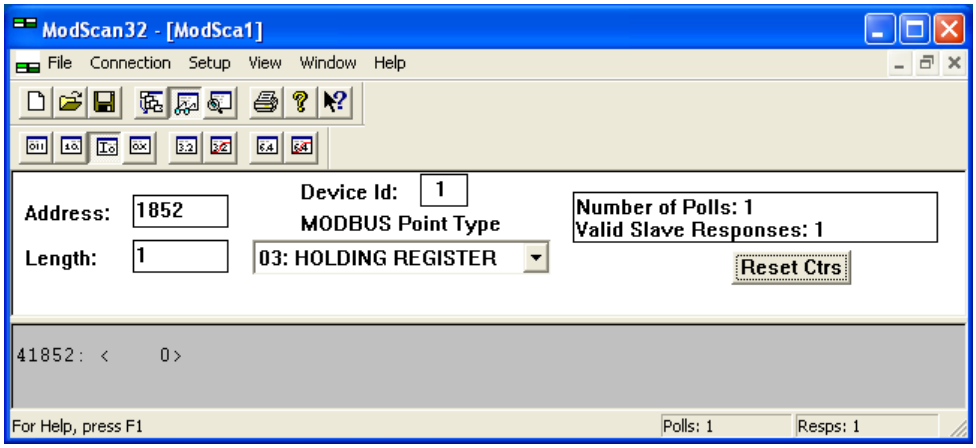


Figure 3-5: Modbus - configuration example 3

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 3-4 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 3-4: Modbus - exception responses

Chapter 4.
CAN (CAL)

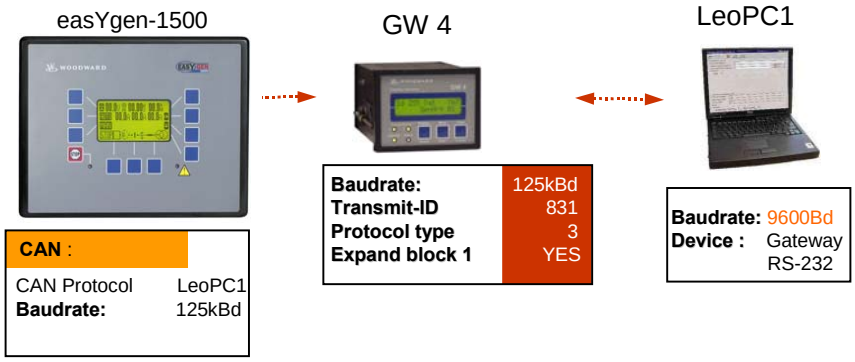


Figure 4-1: CAN (CAL) interface - overview

NOTE

The transmission rate is configurable (default: 125 kBaud). If a GW 4 is used for data transfer, a transmission rate of 125 kBaud must be configured.

The CAN ID, on which the easYgen is transmitting is calculated as follows:

CAN-ID = d‘800 + Item number (or H‘320 + item number)

(The item number is an adjustable parameter in the easYgen, which directly influences the CAN ID that the unit sends the visualization message).

A visualization message which is send out of an easYgen has got 8 Byte and is built as follows:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
H‘DD	MUX number	Data word 1 High-Byte	Data word 1 Low Byte	Data word 2 High-Byte	Data word 2 Low Byte	Data word 3 High-Byte	Data word 3 Low Byte

The byte 0 is always used to show the hexadecimal value H'DD in a visualization message. This defines the message as a visualization message. As the complete transmission telegram of the easYgen includes more than three words byte 1 sends additionally a MUX number starting with 0. Therefore it is theoretically possible to send (256 × 3 = 768) words via the CAN ID. The whole telegram is built up as follows:

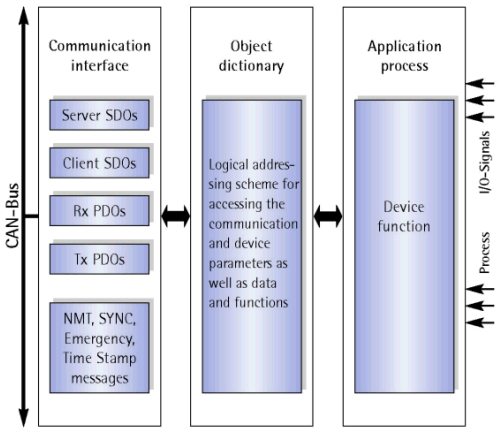
line 1: MUX number 0, word 1
line 2: MUX number 0, word 2
line 3: MUX number 0, word 3
line 4: MUX number 1, word 1
line 5: MUX number 1, word 2
.
line (n): MUX number (n-1/3), word 1
line (n+1): MUX number (n-1/2), word 2
line (n+2): MUX number (n-1/1), word 3

(n) depends on the total length of the unit special telegram and can not be larger than H’FF.
Refer to Appendix A for the interface telegram.

Chapter 5.
CANopen

Introduction

Extract from: Controller Area Network; Basics, Protocols, Chips and Applications; By Prof. Dr.-Ing. K. Etschberger; ISBN: 3-00-007376-0; also see IXXAT GmbH (http://www.ixxat.de)



The CANopen family profile defines a standardized application for distributed industrial automation systems based on CAN as well as the communication standard CAN CAL. CANopen is a standard of CAN-in-Automation (CiA) that after its release, found a broad acceptance, especially in Europe. CANopen can be considered the leading standard for CAN based industrial and embedded system solutions.

The CANopen family profile is based on a "Communication Profile", which specifies the basic communication mechanisms and their description.

The most important device types such as digital and analog I/O modules, drives, operating devices, controllers, programmable controls or encoders, are described by "Device Profiles". The device profiles define the functionality, parameters, and access to process data corresponding to the types of standard devices. These standardized profiles permit devices from different manufacturers to be accessed via the bus in exactly the same manner.

The fundamental element of the CANopen standard is the description of the device functionality through an object dictionary (OD). The object dictionary is divided into two sections. The first section contains general device information like device identification, manufacturer name, etc., as well as communication parameters. The second section describes the specific device functionality.

A 16-Bit index and an 8-Bit sub-index identify the entry ("object") in the object dictionary. Each entry in the object dictionary provide a basis for a standardized network access to the "Application Objects" of a device, such as input and output signals, device parameters, device functions or network variables.

The functionality and characteristics of a CANopen device can be described by means of an "Electronic Data Sheet" (EDS) using the ASCII-format. The EDS acts as a kind of template that describes the data and features, which are accessible via the network. The "Device Configuration File" (DCF) describes the actual device settings. EDS and DCF can be provided in the form of a data carrier, which can be downloaded from the Internet or stored inside the device.

Similar to other well-known field bus systems CANopen also distinguishes two basic data transfer mechanisms: The high-speed exchange of small process data portions through "Process Data Objects (PDO)" as well as the access to entries in the object dictionary through "Service Data Objects (SDO)". The latter ones are primarily used for the transmission of parameters during the device configuration as well as in general for the transmission of larger data portions. Process data object transmissions are generally event triggered, cyclic or requested as broadcast objects without the additional protocol overhead. A PDO can be used for the transmission of a maximum of 8 data bytes. In connection with a synchronization message, the transmission as well as the acceptance of PDOs can be synchronized through the entire network ("Synchronous PDOs"). The assignment of application objects to a PDO (Transmission Object) is adjustable through a structure description ("PDO Mapping") which is stored in the object dictionary, thus allowing the adjustment of a device to the corresponding application requirements.

The transmission of SDOs is performed as a confirmed data transfer with two CAN objects in form of a peer-to-peer connection between two network nodes. The addressing of the corresponding object dictionary entries is accomplished by specifying the index and the sub-index of the entry. Transmitted messages can be unlimited in length. The transmission of SDO messages involves an additional protocol overhead.

Standardized event-triggered "Emergency Messages" of high priority are reserved to report device malfunctions. A common system time can be provided through a central timing message (not included yet).

Management functionality like controlling and monitoring the communication status of the nodes is accomplished by a network management protocol (NMT) organized according to a logical master-slave relationship. Two alternative mechanisms ("Node-Guarding" and "Heartbeat-messages") are available to implement node-monitoring functionality.

The assignment of CAN message identifiers to PDOs and SDOs is possible by direct modifications of entries inside the data structure of the object dictionary or, for simple system structures, through the use of pre-defined identifiers.

Server Data Objects (SDO) - Communication

As already mentioned in the introduction, each CANopen device has an object directory.
All parameters, status variables, measurement values, and input values of the device are stored in this object directory. These parameters are called objects in the CANopen protocol description.
The single objects may contain up to 254 values. If an object has more than one value, these contain a sub-index.

Example: Object 1017h with One Value

Name of the object: Producer Heartbeat Time
Contains a value, which may be read and written.

Example: Object 1200h with Several Values

Name of the object: Server SDO parameter
Sub-index 0 contains the number of sub-indices.
Sub-index 1 contains the COB-ID Client -> Server (rx)
Sub-index 2 contains the COB-ID Server -> Client (tx)

Reading out and changing these objects is performed using an SDO.
This data exchange will be implemented using at least two CAN telegrams, where each on is using an own CAN identifier.

The CAN identifiers of the default service data object are fixed in the object 1200h and are changed using the Node ID.
The values are:
CAN identifier for the reception (Client -> Server): Node ID + 1536 (600h)
CAN identifier for the reply (Server -> Client): Node ID + 1408 (580h)

Some applications (e.g. easYgen + easYlite) require that several SDO clients access one SDO server. To ensure a proper communication, the SDO server must provide several service data objects.
These are described in the objects 1201h to 127Fh.

The easYgen-1500 provides five additional service data objects.

These may be configured under the point "Additional S-SDO".
2 to 5 Client->Server COP-ID (tx)
CAN-IDs, on which SDO requests are received.

2 to 5 Server->Client COP-ID (rx)
CAN-IDs, on which SDO replies are sent.

If a unit is not only intended to work as a server, but also as a client (this is the easYlite in the application easYgen-1500 + easYlite), it requires client service data objects.

These may be configured under the point "Additional C-SDO (client SDO)" for the easYlite.
1. Client->Server COP-ID (rx)
CAN-IDs, on which SDO requests are sent.

1. Server -> Client COP-ID (tx)
CAN-IDs, on which SDO replies are received.

By entering 80000000h (2147483648 dec) for the CAN ID, the CAN identifiers can be disabled if they are not necessary.

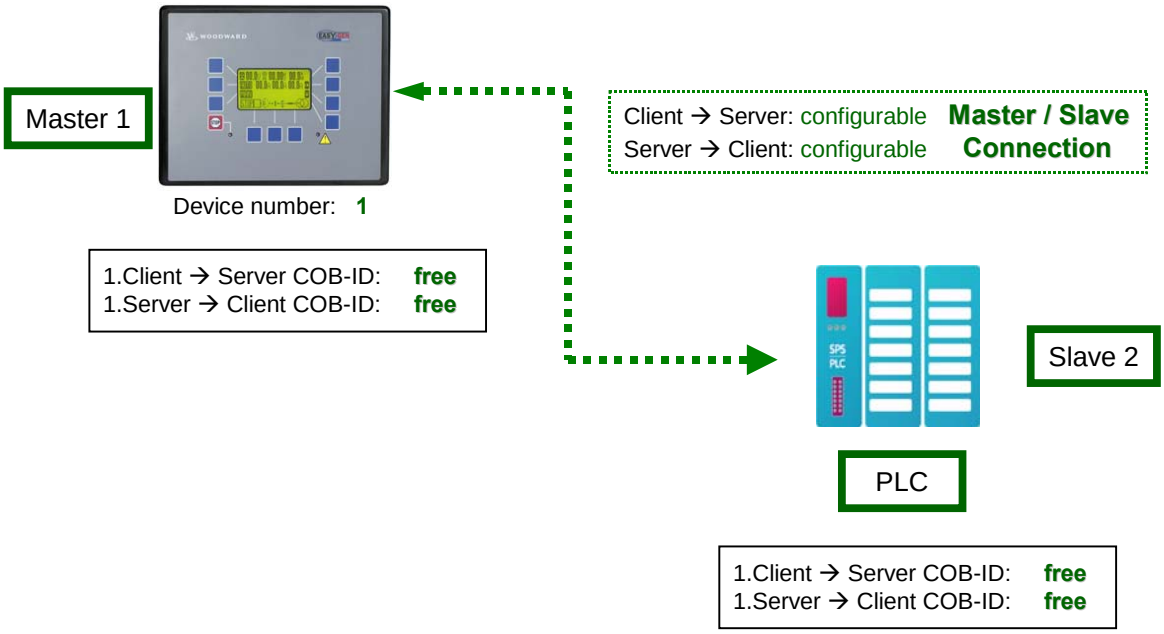


Figure 5-1: CANopen interface - overview

NOTE

If the easYgen-1500 is configured to CAN-Open Master = "Yes" and one external terminal, it sends configuration messages to the default service data objects to the connected terminal as SDO client.

Process Data Objects (PDO)

Process data objects are used to transmit real-time data. No, one, or several recipients are possible with this. Process data objects may be sent cyclically or continuously (other transmission types are not supported by the easYgen), this is configured using the parameter "Transmission Type".
The values 254 and 255 define an asynchronous transmission.
In case of the asynchronous transmission, the PDOs are sent after a certain time. This will be configured using the event timer.

The values 1 to 240 are used for a synchronous transmission. The PDO will be sent as a response to a received SYNC message here. If the value is configured to 1, the PDO will be sent for every received SYNC message, if the value is configured to 2, the PDO will only be sent for every 2nd SYNC message, and so on.

No PDOs will be sent for the remaining values.

Data in the PDO

The data, which is transmitted with the PDO, is to be configured at the unit. The parameters "Mapped Object" are provided for this.

The parameter "Number of Mapped Objects" is used to configure the number of mapped objects.

Then, up to four objects may be entered, whose data is to be transmitted. The identifiers of the objects may be found in the operating instructions.

Setting the Transmit PDO (Examples)

With the TPDOs up to 8 data bytes can be send.

Configuration of a data protocol

Parameter	Value
Number of mapped objects	Parameter no. 1 to 4
1. Mapped Object	for example parameter no. 3191
2. Mapped Object	Parameter no. 0
3. Mapped Object	Parameter no. 0
4. Mapped Object	Parameter no. 0

Configuration of a TPDO message

A TPDO can contain one or more mapped objects with a maximum of 4 data bytes each. The TDPO message has a maximum combined total of 8 bytes.

Example 1

Parameter	Value	Number of bytes
Number of mapped objects	Parameter no. 2	
1. Mapped Object	Parameter no. 108	unsigned32 -> 4byte
2. Mapped Object	Parameter no. 160	unsigned16 -> 2byte – total 6 bytes
3. Mapped Object	Parameter no. 0	
4. Mapped Object	Parameter no. 0	

The TPDO has a length of 6 bytes.

Example 2:

Parameter	Value	Number of bytes
Number of mapped objects	Parameter no. 2	
1. Mapped Object	Parameter no. 108	unsigned32 -> 4Byte
2. Mapped Object	Parameter no. 109	unsigned32 -> 4Byte – total 8 bytes
3. Mapped Object	Parameter no. 0	
4. Mapped Object	Parameter no. 0	

The TPDO has a length of 8 bytes.

Example 3:

Parameter	Value	Number of bytes
Number of mapped objects	Parameter no. 3	
1. Mapped Object	Parameter no. 108	unsigned32 -> 4byte
2. Mapped Object	Parameter no. 109	unsigned32 -> 4byte – total 8 bytes
3. Mapped Object	Parameter no. 110	unsigned32 -> 4byte – total 12 bytes !FAULT!
4. Mapped Object	Parameter no. 0	

The TPDO has a length of 12 bytes, as only 8 bytes are admissible, an idle TPDO is sent.

Configuration of a SYNC message

Parameter	Value	Number of bytes
Number of mapped objects	Parameter no. 0	
1. Mapped Object	Parameter no. 0	
2. Mapped Object	Parameter no. 0	
3. Mapped Object	Parameter no. 0	
4. Mapped Object	Parameter no. 0	

The TPDO has a length of 0 bytes. If the COP ID is configured accordingly for example 80h = 128dez, it is working like a SYNC message. Thereby the easYgen has the possibility to send a SYNC message to the attached devices to arrange a reaction with a PDO, however the time of the transmission is not appraised.

SYNC Message

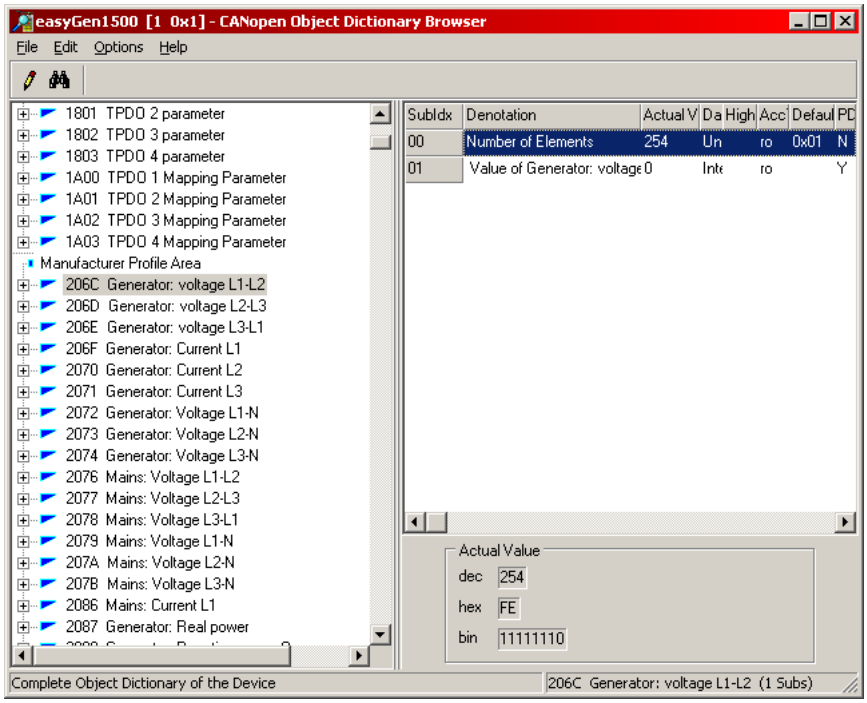
The SYNC message is a CAN message without data. The CAN ID on which the easYgen sends appropriately configured PDOs, is configured with the parameter "COB-ID SYNC Message".

Using a CANopen Configuration Program

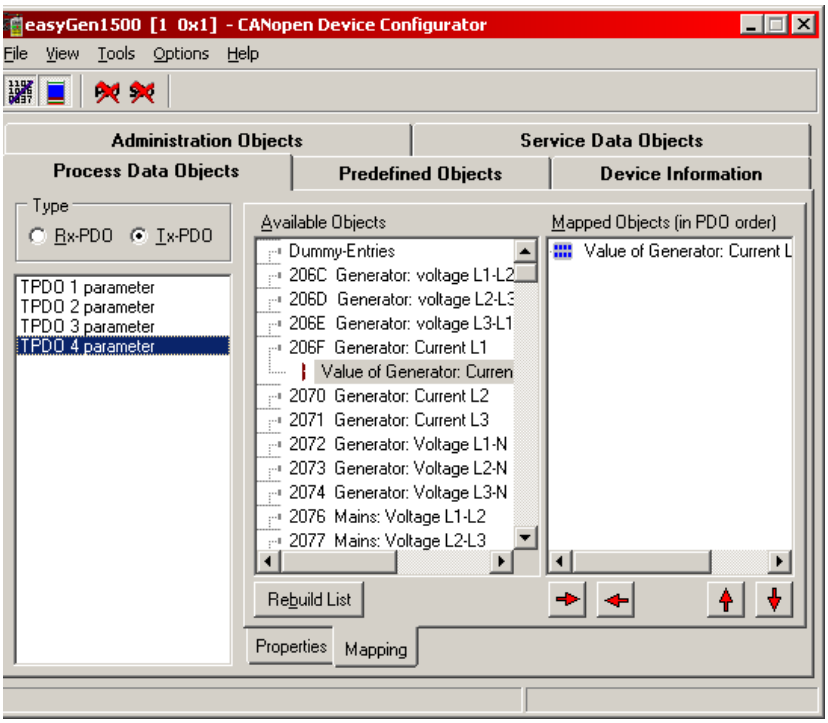
If the easYgen is used as a single unit, the default settings provide useful operation possibilities already. If the easYgen is used together with other CANopen devices, a detailed configuration will be necessary.

An *.eds file is enclosed with the unit for this purpose. An example of this file being used with the CANopen Configuration Studio of IXXAT is shown in the following.

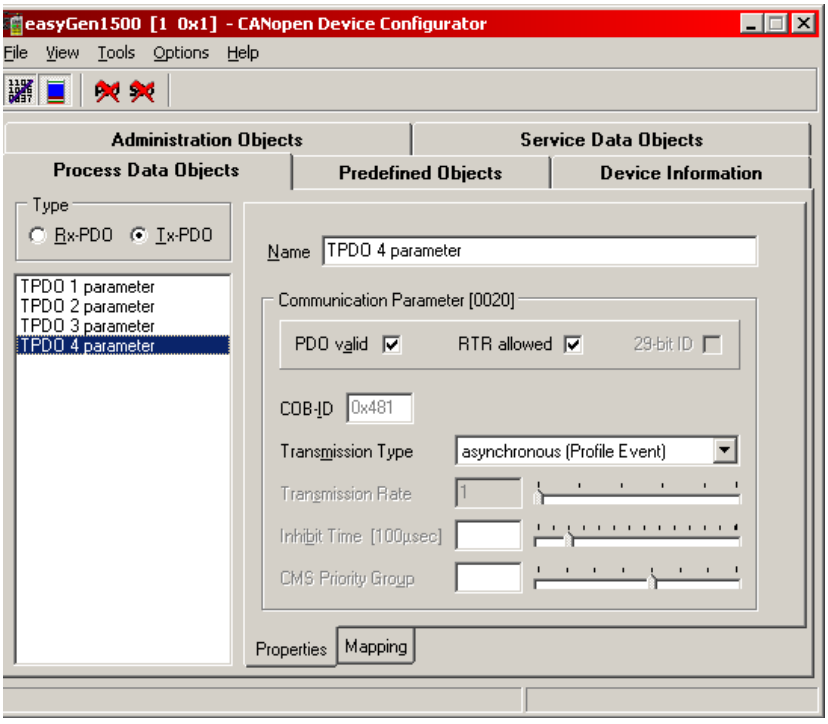
Please refer to IXXAT for a more detailed explanation about this tool.



The easYgen parameters may be changed after loading the *.eds file. The values are only overwritten by the easYgen if the correct password has been entered prior to attempting to make any changes; otherwise, a fault message will be issued, which states that the parameter may not be overwritten.



The configuration of the mapped objects of a send PDO is very clear and easy with this program.



Configuration of the transmission type:
The following transmission types are supported:

- "asynchronous (Profile Event)" and "asynchronous (Manuf. Event)" – both send a message after the event timer has expired
- "synchronous cyclic" with the according transmission rate

Settings for Connection with External Devices

Name	Description
Device number	Determines the node ID for CANopen
Protocol	Determines the protocol – select this for CANopen
Baudrate	Determines the baud rate

NOTE

The standard values of the easYgen enable to connect devices on the basis of the CANopen protocol quickly and easily.

ECU data in J1939 format may be transmitted together with data in CANopen format on the same bus at the same time. The baud rate must be the same for all participants on the bus.

Figure 5-2 shows an overview of the different device combinations, which are possible:

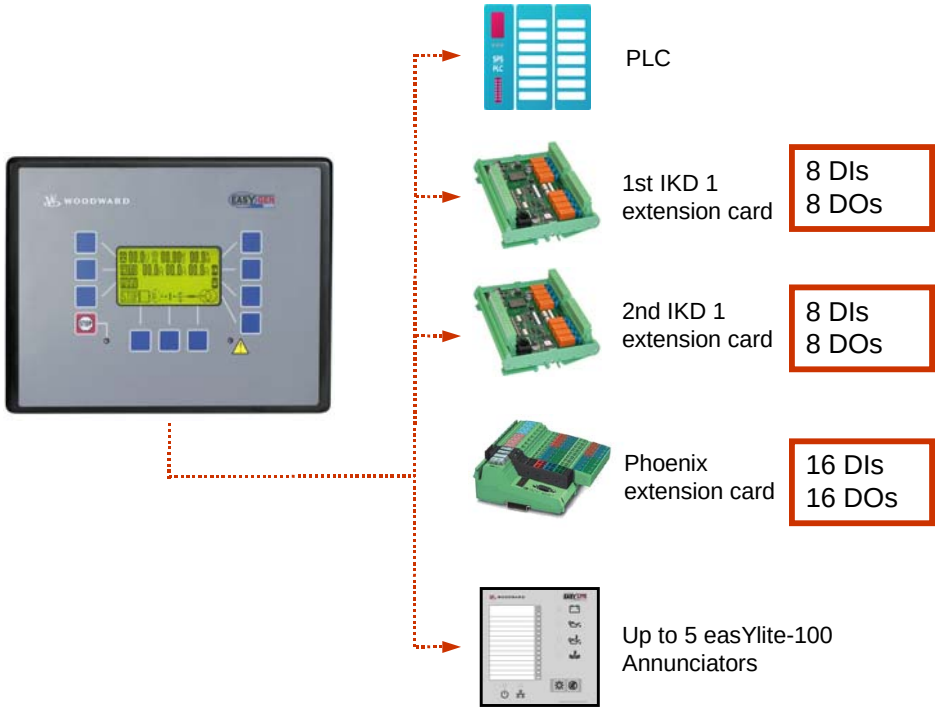


Figure 5-2: CANopen interface - external devices

PLC:	PLC of the plant
IKD 1:	2 extension cards, each for 8 additional external inputs and outputs
Phoenix extension card:	Extension card for 16 additional external inputs and outputs
easYlite:	External display unit

NOTE

The parameters, which are highlighted red in the following figures, must be observed particularly, because these are essential for a communication with the respective device and may differ the default values.

CAUTION

The ID settings are entered in hexadecimal format in the easYgen and are therefore listed in decimal and hexadecimal format in the following tables.

Expansion with One IKD 1 (8 Additional External DI/DO)

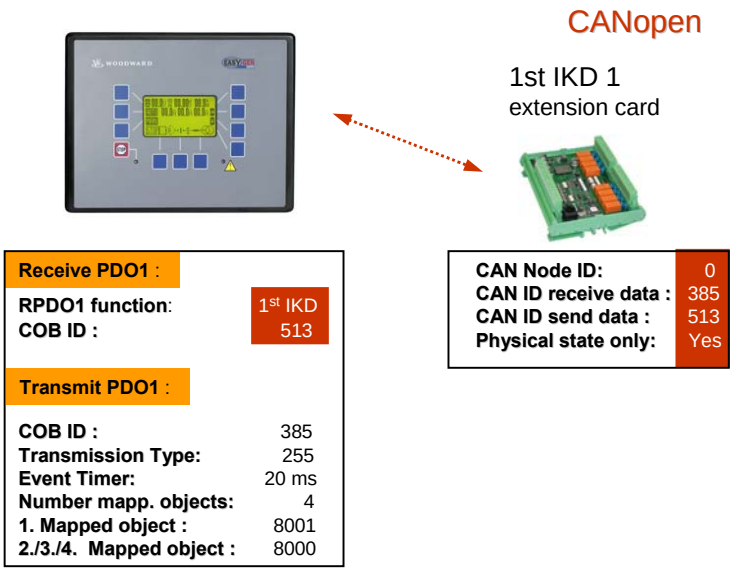


Abbildung 5-3: CANopen Schnittstelle - Einstellungen für externe Geräte

Configuration of the receive PDO 1

Parameter	Value	Comment
COB-ID	201h = 513 Dec	CAN-ID on which the data are received
Function	1. IKD	The data received on the COB-ID were assigned to the external DI 1 to DI 8
Node-ID of the device	2	The IKD is not configured by the easYgen; the suggested value is therefore a default value.
RPDO-COB-ID ext. device 1	282h = 642 Dec	The IKD is not configured by the easYgen; the suggested value is therefore a default value.

Configuration of transmit PDO (e.g. PDO1)

Parameter	Value	Comments
COB-ID	181h = 385 Dec	CAN-ID on which the data was sent
Transmission type	FFh = 255 Dec	The PDO is sent circular
Event-timer	20	The PDO is sent every 20 ms
Number of mapped objects	4	
1. Mapped Object	Parameter no. 8001	DI 1 to 8 is issued
2. Mapped Object	Parameter no. 8000	
3. Mapped Object	Parameter no. 8000	
4. Mapped Object	Parameter no. 8000	

Settings at the IKD

Parameter	Value	Comments
Node-ID	0	So that the entries of the CAN IDs are taken over
CAN-ID transmitting data	201h = 513 Dec	The easYgen receives on this ID.

Settings for DIs on IKD

Parameter	Value	Comments
Physical state	YES	Only the physical state of the inputs is transmitted. (The settings under idle current, tripping delay, revert delay, enabling, self-resetting and acknowledge input are without effect). These settings have to be selected for devices, which include these parameters (e.g. the easYgen-1500).

Check of the settings

Actuate an external DO via the *LogicsManager* and check whether the respective relay at the IKD operates. Scroll the display screens to view the ext. discrete inputs 1 to 8. A set of discrete inputs will be shown that correspond to the IKD. Use the "FAQ CAN Bus" chapter on page 35 to troubleshoot any CAN bus faults.

Expansion with Two IKD 1 (16 Additional External DI/DO)

The first IKD will be adjusted like described above. For the second IKD the following settings must be configured.

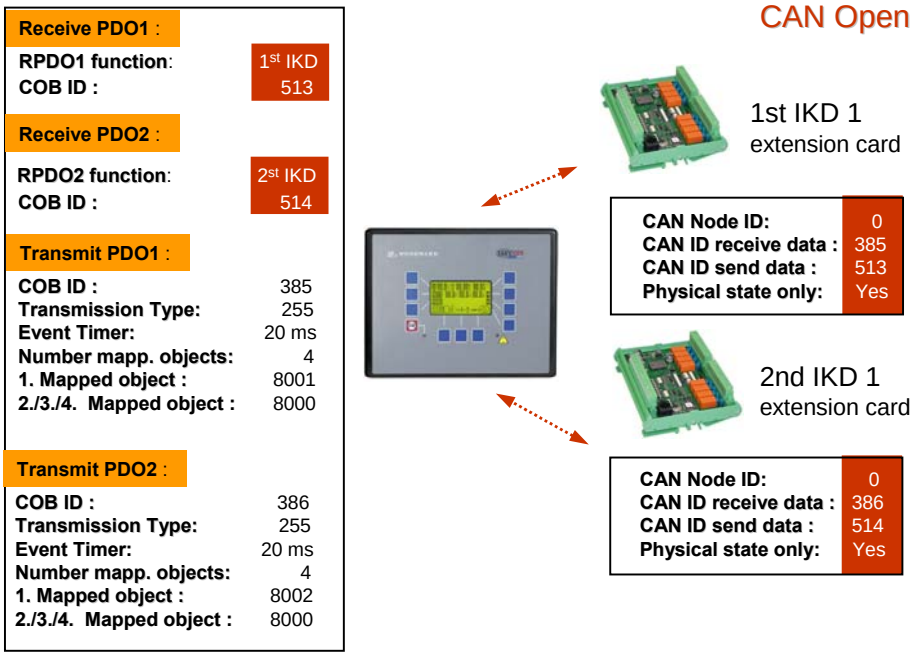


Figure 5-4: CANopen interface - expansion with two IKD 1

Setting of the receive PDO 2

Parameter	Value	Comments
COB-ID	202h = 514 Dec	CAN-ID on which the data are received
Function	2. IKD	The data received on the COB-ID were assigned to the external DI 9 to DI 16
Node-ID of the device	3	The IKD is not configured by the easYgen; the suggested value is therefore a default value.
RPDO-COB-ID ext. device 1	283h = 643 Dec	The IKD is not configured by the easYgen; the suggested value is therefore a default value.

Settings of transmit PDO (e.g. PDO 2)

Parameter	Value	Comments
COB-ID	182h = 386 Dec	CAN-ID on which the data was sent
Transmission type	FFh = 255 Dec	The PDO is sent circular
Event-timer	20	The PDO is sent every 20 ms
Number of mapped objects	4	
1. Mapped Object	Parameter no. 8002	DI 9 to 16 is issued
2. Mapped Object	Parameter no. 8000	
3. Mapped Object	Parameter no. 8000	
4. Mapped Object	Parameter no. 8000	

Settings of DIs on IKD 1 #2

Parameter	Value	Comments
Node-ID	0	That the entries of CAN-IDs are accepted
CAN-ID receiving data	182h = 386 Dec	easYgen receives on this ID
Relay 1 as ready for operation	NO	Otherwise the easYgen cannot be controlled correctly.

Settings on IKD 1 #2

Parameter	Value	Comments
Node-ID	0	So that the entries of the CAN IDs are taken over
CAN-ID transmitting data	202h = 514 Dec	The easYgen receives on this ID.

Settings for DIs on IKD 1 #2

Parameter	Value	Comments
Physical state	YES	Only the physical state of the inputs is transmitted. (The settings under idle current, tripping delay, revert delay, enabling, self-resetting and acknowledgeinput are without effect). These settings have to be selected for devices, which include these parameters (e.g. the easYgen-1500).

Check of the settings

Actuate an external DO via the *LogicsManager* and check whether the respective relay at the IKD operates. Scroll the display screens to view the ext. discrete inputs 9 to 16. A set of discrete inputs will be shown that correspond to the IKD. Use the "FAQ CAN Bus" chapter on page 35 to troubleshoot any CAN bus faults.

Expansion with the Phoenix terminal IL CAN BK / ILB CO 24 16DI 16DO (16 DI/DO)

The specified settings are valid for a Phoenix terminal with Node ID 2.

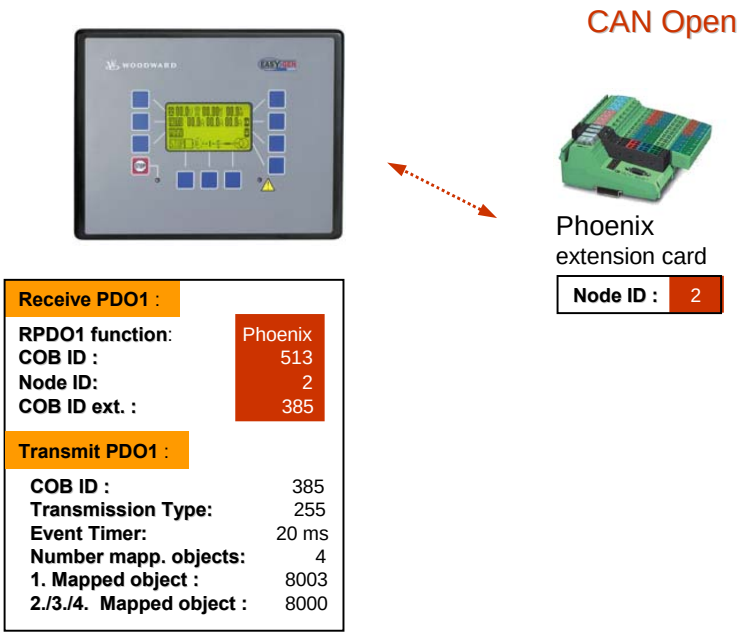


Figure 5-5: CANopen interface - expansion with Phoenix terminal

Parameter	Value	Comments
CAN-open Master	YES	
Max time for reply ext. devices	1.0	
Time for re-init ext. devices	100	If this time is set 0, the attached Phoenix terminal may not be configured correctly.

Setting of the receiving PDO 1

Parameter	Value	Note
COB-ID	201h = 513 Dec	CAN-ID to receive data
Function	BK16DIDO	The received data (via the COB-ID) are copied to the ext. DI 1 to 16
Node-ID of the device	2	According to the setting of the terminals
RPDO-COB-ID ext. device 1	181h = 385 Dec	The Phoenix terminal must be configured in that way that it can receive a PDO on that COB-ID

CAUTION

The 2nd PDO this function must be configured to OFF.

NOTE

The easYgen is the CANopen master.

Settings of the transmitting PDO (i.e. PDO3)

Parameter	Value	Note
COB-ID	381h = 385 Dec	CAN-ID which is used to send data Has to be the same as parameter RPDO-COB-ID of the ext. device 1
Transmission type	FFh = 255 Dec	The PDO is cyclically sent
Event-timer	20	The PDO is sent every 20 ms
Number of mapped objects	1	
1. Mapped Object	Parameter no. 8003	The status of DI 1 to 16 is issued
2. Mapped Object	Parameter no. 0	
3. Mapped Object	Parameter no. 0	
4. Mapped Object	Parameter no. 0	

Check of the settings

Actuate an external DO via the *LogicsManager* and check whether the respective relay at the Phoenix terminal operates.
Scroll the display screens to view the ext. discrete inputs 1 to 8 and ext. discrete inputs 9 to 16. A set of discrete inputs will be shown that correspond to the Phoenix terminal. Use the "FAQ CAN Bus" chapter on page 35 to troubleshoot any CAN bus faults.

Expansion with an easYlite

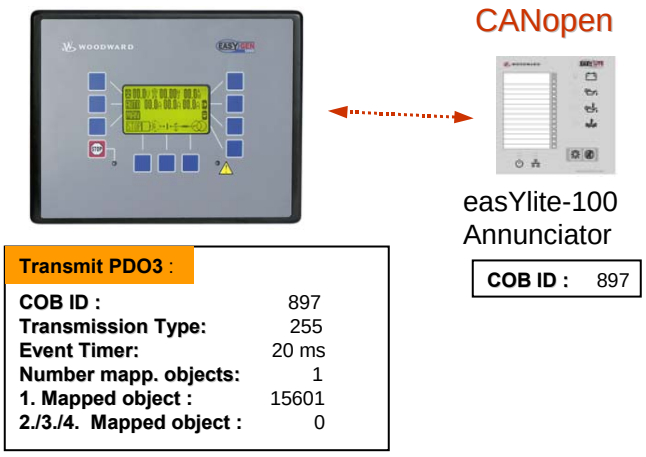


Figure 5-6: CANopen interface - expansion with easYlite

Settings of the transmit PDO (i.e. PDO3)

Parameter	Value	Note
COB-ID	381h = 897 Dec	CAN-ID which is used to send data Has to be the same as parameter RPDO-COB-ID of the ext. device 1
Transmission type	FFh = 255 Dec	The PDO is cyclically sent
Event-timer	20	The PDO is sent every 20 ms
Number of mapped objects	1	
1. Mapped Object	Parameter no. 15601	
2. Mapped Object	Parameter no. 0000	
3. Mapped Object	Parameter no. 0000	
4. Mapped Object	Parameter no. 0000	

FAQ CAN Bus

The following are reason that no data is transmitted:

- T structure bus is utilized
- CAN-L and CAN-H are interchanged
- Not all devices on the bus are using identical Baud rates
- Terminating resistor are missing
- Baud rate to high for wiring length

Recommendations of Woodward

The maximum length of the communication bus wiring is dependent on the configured Baud rate.

Baud rate	Max. length
1000 kbit/s	25 m
800 kbit/s	50 m
500 kbit/s	100 m
125 kbit/s	250 m
50 kbits/s	1000 m
20 kbit/s	2500 m

Source: CANopen; Holger Zeltwanger (Hrsg.); 2001 VDE VERLAG GMBH, Berlin und Offenbach; ISBN 3-8007-2448-0

The maximum specified length for the communication bus wiring might not be achieved if wire of poor quality is utilized, there is high contact resistance, or other conditions exist. Reducing the baud rate may overcome these issues.

Device Combinations and Bus Load

The baud rate has a direct effect on the number of messages, which may be exchanged via the bus per time unit. A bus load should not exceed approx. 40% capacity to prevent long waiting times or loss of messages.

The following information provides clues for reasonable device configurations at certain baud rates. The exact configuration is to be taken from the respective operation manuals.

20 kBaud

easYgen	PLC	easYlite	IKD (8DIDO)
1 PDO every 50ms	only receiver	only receiver	--
2 PDOs every 100 ms	only receiver	only receiver	--
2 PDOs every 150 ms	1 PDO every 150 ms	only receiver	--
2 PDOs every 150 ms	only receiver	only receiver	1 PDO every 160 ms

If a lot of easYlites and PLCs are used, the NMT Error Control may also represent a reasonable bus load.
A solution may be to send the NMT Error Control of the easYlite more rarely using the Producer Heartbeat Time.

If the IKD sends only every 160ms, the respective discrete inputs have a jitter of 160ms, it is recommended to receive two messages, therefore, the delay of the ext. discrete inputs should also be configured greater than 160ms.

50 kBaud

easYgen	PLC	easYlite	BK 16DIDO	IKD (8DIDO)
1 PDO every 20ms (for BK 16DIDO) 1 PDO every 200ms for easYlite 1 PDO every 200ms for PLC	only receiver	only receiver	1 PDO every 20ms	--
1 PDO every 200ms for easYlite 1 PDO every 20ms for PLC (e.g. DOs) 1 PDO every 150ms for PLC (e.g. visu data)	1 PDO every 20 ms	only receiver	Not existing, if the easYgen is the NMT master, set "Time re-init ext. devices" to 0 (off).	--
1 PDO every 20ms (for IKD) 1 PDO every 200ms for easYlite 1 PDO every 200ms for PLC	only receiver	only receiver	---	1 PDO every 20ms
2 PDO every 40ms (for IKD/PLC) 1 PDO every 200ms for easYlite 1 PDO every 200ms for PLC	1 PDO every 40ms (may also be the 2.IKD)	only receiver	---	1 PDO every 40ms

If numerous easYlites and PLCs are used, the NMT Error Control may also represent a reasonable bus load.
A solution may be to send the NMT Error Control of the easYlite more rarely using the Producer Heartbeat Time.

Sometimes the Phoenix CO 16DIDO fails with this baud rate.

100 kBaud

easYgen	PLC	easYlite	IKD (8DIDO)
1 PDO every 20ms for easYlite 1 PDO every 20ms for PLC (e.g. DOs) 1 PDO every 20ms for PLC (e.g. visu data)	1 PDO every 20 ms	only receiver	
1 PDO every 40ms for easYlite 2 PDO every 20ms for PLC (e.g. DOs) 1 PDO every 40ms for PLC (e.g. visu data)	1 PDO every 20ms (may also be the 2.IKD)	only receiver	1 PDO every 20ms

The Phoenix terminals do not support this baud rate.

125 kBaud

easYgen	PLC / Phoenix BK 16 DIDO	easYlite	IKD (8DIDO)
4 PDO every 20ms for easYlite, DO, visualization	1 PDO every 20 ms	only receiver	
4 PDO every 20ms for easYlite, DO, visualization	PLC with 1 PDO every 20ms	only receiver	1 PDO every 20ms

Sometimes the Phoenix CO 16DIDO fails with this baud rate.

250kBaud and above

The maximum load of the CAN bus cannot be reached with combinations of easYgen, easYlite, and external terminals.

A maximum baud rate of 500kBaud may be configured at the IKD.

NOTE

Engine Control ECU with J1939

An engine control unit with J1939 protocol may be connected to the bus in addition to the CANopen components. The J1939 messages load the CAN bus with additional messages so that only half of the original capacity of CAN bus messages is available for first time reads.

Chapter 6.
CAN SAE J1939

Introduction

Messages of a device 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 LeoPC.
The Baud rate is similar for all on CAN connected devices independent of the selected protocol.

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.

The J1939 messages result an additional burden to the bus, so that only the half capacity of the CAN bus is available for the CANopen messages in rough approximation.

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 displayed always.

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.)
"SPN = FMI = 0" or "SPN = 524287, FMI = 31" means: No alarm exists or no alarm codes are available. (refer to the manuals of the engine control being used).

OC: (Occurrence Count) indicates how often an alarm occurred.

DM1/DM2 Lamp State	Display in the device		in LeoPC
Indicator lamp	OFF	ON	missing
Amber warning lamp	OFF	ON	missing
Red stop lamp	OFF	ON	missing
Lamp malfunction	OFF	ON	missing

The DM1/2 messages are displayed as follows:
DM1/2: SPN <xx>
SP:<sssss> F:<ff> OC:<ooo>

The following is valid:
<xx> is the sequential number (0 to 10),
<sssss> is the SPN number
<ff> is the FMI (Fail Mode Identifier)
<ooo> is the OC (Occurrence Count)

NOTE

The DM1/2 messages are only displayed if a message exists.

The lamp status is always displayed.

Standard Messages

Suspect Parameter Number	Parameter Group Number	Description	Setting	Display with defective sensor in Leo PC	Display on missing sensor value in LeoPC
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 h	2147483646h	2147483647h
513	61444	Actual engine torque	1%	32766%	32767%

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, the display in the device is blanked out, in LeoPC it is displayed according to the table "Display at missing sensor value".
- If the value is sent as defect, the device displays "-----". In LeoPC it is displayed according to the table "Display at defect sensor".

NOTE

You must scroll once through the device display to save the current values.

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 LeoPC
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)

Special S6 Messages

Suspect Parameter Number	Parameter Group Number	Description	Display in unit	Display in LeoPC
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

If DLN2 does not transmit, the following is valid:
- the screens in the device are suppressed
- "missing" is displayed in LeoPC1

Monitoring the Interface

The control has a watchdog to monitor if J1939 messages are received. If messages are not received, a fault condition is recognized regardless if the J1939 messages are displayed or not (refer to the Configuration Manual).

Watchdogs

The watchdogs are disabled by default. This is explained in detail about in the device manual in the chapter "Watchdogs", and in the LeoPC1 manual under "Watchdogs Interface" and "Watchdogs J1939".

Appendix A. Telegrams

Transmission Telegram

MUX	No.	Content (words)	Engineering unit	Remark	
0/1	**1**	Protocol number		'4003' = easYgen-1000 Series	
0/2	**2**	Generator: voltage V_{L12}	1/10 V	High word	
0/3	**3**			Low word	
1/1	**4**	Generator: frequency f	1/100 Hz		
1/2	**5**	Generator: voltage V_{L1N}	1/10 V	High word	
1/3	**6**			Low word	
2/1	**7**	Mains: frequency f	1/100 Hz		
2/2	**8**	Generator: voltage V_{L23}	1/10 V	High word	
2/3	**9**			Low word	
3/1	**10**	Generator: power factor cosphi	1/1000, dim.less		
3/2	**11**	Generator: voltage V_{L2N}	1/10 V	High word	
3/3	**12**			Low word	
4/1	**13**	Mains: power factor cosphi	1/1000, dim.less		
4/2	**14**	Generator: voltage V_{L31}	1/10 V	High word	
4/3	**15**			Low word	
5/1	**16**	Engine: speed via Pickup	RPM		
5/2	**17**	Generator: voltage V_{L3N}	1/10 V	High word	
5/3	**18**			Low word	
6/1	**19**	Common: battery voltage	1/10 V		
6/2	**20**	Mains: voltage V_{L12}	1/10 V	High word	
6/3	**21**			Low word	
7/1	**22**	Common: analog input [T1]	dim.less		
7/2	**23**	Mains: voltage V_{L1N}	1/10 V	High word	
7/3	**24**			Low word	
8/1	**25**	Common: analog input [T2]	dim.less		
8/2	**26**	mains: voltage V_{L23}	1/10 V	High word	
8/3	**27**			Low word	
9/1	**28**	Discrete input: state Evaluation of the discrete inputs is done using the physical state (voltage applied = logical "1"; the configured operating type NO/NC will be ignored)		Bit 15	Discrete input [D1]
				Bit 14	Discrete input [D2]
				Bit 13	Discrete input [D3]
				Bit 12	Discrete input [D4]
				Bit 11	Discrete input [D5]
				Bit 10	Discrete input [D6]
				Bit 09	Discrete input [D7]
				Bit 08	Discrete input [D8]
				Bit 07	internal
				Bit 06	internal
				Bit 05	internal
				Bit 04	internal
				Bit 03	internal
				Bit 02	internal
				Bit 01	internal
				Bit 00	internal

MUX	No.	Content (words)	Engineering unit	Remark	
9/2	**29**	Mains: voltage V_{L2N}	1/10 V	High word	
9/3	**30**			Low word	
10/1	**31**	Relay: state		Bit 15	Relay [R1]
				Bit 14	Relay [R2]
				Bit 13	Relay [R3]
				Bit 12	Relay [R4]
				Bit 11	Relay [R5]
				Bit 10	Relay [R6]
				Bit 9	Relay [R7]
				Bit 8	Relay [R8]
				Bit 7	Relay [R9]
				Bit 6	Relay [R10]
				Bit 5	Relay [R11]
				Bit 4	Internal
				Bit 3	Internal
				Bit 2	Internal
				Bit 1	Internal
				Bit 0	Internal
10/2	**32**	Mains: voltage V_{L31}	1/10 V	High word	
10/3	**33**			Low word	
11/1	**34**	System state		Bit 15	Internal
				Bit 14	Internal
				Bit 13	Internal
				Bit 12	Internal
				Bit 11	Internal
				Bit 10	Internal
				Bit 9	Internal
				Bit 8	Operating mode STOP
				Bit 7	Operating mode MANUAL
				Bit 6	Operating mode AUTOMATIC
				Bit 5	Engine is running
				Bit 4	
				Bit 3	Reply: MCB is open
				Bit 2	
				Bit 1	Reply: GCB is open
				Bit 0	
11/2	**35**	Mains: voltage V_{L3N}	1/10 V	High word	
11/3	**36**			Low word	
12/1	**37**	Alarm class		Bit 15	Internal
				Bit 14	Internal
				Bit 13	Internal
				Bit 12	Internal
				Bit 11	Internal
				Bit 10	Internal
				Bit 9	Internal
				Bit 8	Internal
				Bit 7	Internal
				Bit 6	Internal
				Bit 5	Alarm class F
				Bit 4	Alarm class E
				Bit 3	Alarm class D
				Bit 2	Alarm class C
				Bit 1	Alarm class B
				Bit 0	Alarm class A
12/2	**38**	Generator: current I_{L1}	mA	High word	
12/3	**39**			Low word	

MUX	No.	Content (words)	Engineering unit	Remark	
13/1	**40**	Discrete inputs with alarm class The discrete input transmitting telegram bit is logically "1", if the discrete input is configured as an alarm input and has been selected.		Bit 15	Discrete input [D1]
				Bit 14	Discrete input [D2]
				Bit 13	Discrete input [D3]
				Bit 12	Discrete input [D4]
				Bit 11	Discrete input [D5]
				Bit 10	Discrete input [D6]
				Bit 9	Discrete input [D7]
				Bit 8	Discrete input [D8]
				Bit 7	Internal
				Bit 6	Internal
				Bit 5	Internal
				Bit 4	Internal
				Bit 3	Internal
				Bit 2	Internal
				Bit 1	Internal
				Bit 0	Internal
13/2	**41**	Generator: current I_{L2}	mA	High word	
13/3	**42**			Low word	
14/1	**43**	Alarms 1		Bit 15	Overspeed, limit 1
				Bit 14	Overspeed, limit 2
				Bit 13	Underspeed, limit 1
				Bit 12	Underspeed, limit 2
				Bit 11	Unintended stop
				Bit 10	Alarm speed detection
				Bit 9	Stop failure
				Bit 8	Failure during closing of the GCB
				Bit 7	Failure during opening of the GCB
				Bit 6	Failure during closing of the MCB
				Bit 5	Failure during opening of the MCB
				Bit 4	Internal
				Bit 3	Start failure
				Bit 2	Maintenance call "days" expired
				Bit 1	Maintenance call "hours" expired
				Bit 0	Internal
14/2	**44**	Generator: current I_{L3}	mA	High word	
14/3	**45**			Low word	
15/1	**46**	Generator: alarms 1		Bit 15	Generator overfrequency, limit 1
				Bit 14	Generator overfrequency, limit 2
				Bit 13	Generator underfrequency, limit 1
				Bit 12	Generator underfrequency, limit 2
				Bit 11	Generator overvoltage, limit 1
				Bit 10	Generator overvoltage, limit 2
				Bit 9	Generator undervoltage, limit 1
				Bit 8	Generator undervoltage, limit 2
				Bit 7	Generator overcurrent, limit 1
				Bit 6	Generator overcurrent, limit 2
				Bit 5	Generator overcurrent, limit 3
				Bit 4	Generator reverse/reduced power, limit 1
				Bit 3	Generator reverse/reduced power, limit 2
				Bit 2	Generator overload, limit 1
				Bit 1	Generator overload, limit 2
				Bit 0	Internal
15/2	**47**	Mains: current I_{L1}	mA	High word	
15/3	**48**			Low word	

MUX	No.	Content (words)	Engineering unit	Remark	
16/1	**49**	Mains: alarms		Bit 15	Internal
				Bit 14	Internal
				Bit 13	Internal
				Bit 12	Internal
				Bit 11	Internal
				Bit 10	Internal
				Bit 9	Internal
				Bit 8	Internal
				Bit 7	Internal
				Bit 6	Mains overfrequency (for emergency power)
				Bit 5	Mains underfrequency (for emergency power)
				Bit 4	Mains overvoltage (for emergency power)
				Bit 3	Mains undervoltage (for emergency power)
				Bit 2	Mains field rotating wrong direction
				Bit 1	Mains load, limit 1 (*LogicsManager*)
				Bit 0	Mains load, limit 2 (*LogicsManager*)
16/2	**50**	Generator: reactive power Q	var	High word	
16/3	**51**			Low word	
17/1	**52**	Analog input: wire break		Bit 15	Internal
				Bit 14	Internal
				Bit 13	Internal
				Bit 12	Internal
				Bit 11	Internal
				Bit 10	Internal
				Bit 9	Internal
				Bit 8	Internal
				Bit 7	Internal
				Bit 6	Internal
				Bit 5	Internal
				Bit 4	Internal
				Bit 3	Internal
				Bit 2	Wire break analog input [T2]
				Bit 1	Wire break analog input [T1]
				Bit 0	Internal
17/2	**53**	Generator: real power P	W	High word	
17/3	**54**			Low word	
18/1	**55**	Analog inputs: monitoring		Bit 15	Internal
				Bit 14	Internal
				Bit 13	Internal
				Bit 12	Internal
				Bit 11	Internal
				Bit 10	Internal
				Bit 9	Internal
				Bit 8	Internal
				Bit 7	Analog input [T2], limit 2
				Bit 6	Analog input [T2], limit 1
				Bit 5	Analog input [T1], limit 2
				Bit 4	Analog input [T1], limit 1
				Bit 3	Battery: overvoltage, limit 2
				Bit 2	Battery: undervoltage, limit 2
				Bit 1	Battery: overvoltage, limit 1
				Bit 0	Battery: undervoltage, limit 1
18/2	**56**	Mains: real power P	W	High word	
18/3	**57**			Low word	

MUX	No.	Content (words)	Engineering unit	Remark	
19/1	**58**	System state		Bit 15	Turning (gas engine)
				Bit 14	Ignition ON (gas engine)
				Bit 13	Dead bus start GCB
				Bit 12	Dead bus start MCB
				Bit 11	Starter/fuel solenoid (diesel engine) Gas valve (gas engine)
				Bit 10	Start pause
				Bit 9	Engine post run
				Bit 8	Engine will be stopped
				Bit 7	Preglow (diesel engine)
				Bit 6	Crank protection
				Bit 5	Emergency power op. (AMF)/critical mode
				Bit 4	Post-run auxiliary services (gas/diesel engine)
				Bit 3	Mains settling
				Bit 2	Pre-run auxiliary services (gas/diesel engine)
				Bit 1	Emergency power operation (AMF)
				Bit 0	Critical mode (Sprinkler) operation
19/2	**59**	Mains: reactive power Q	var	High word	
19/3	**60**			Low word	
20/1	**61**	Generator: power factor cosphi	1/100, dim.less		
20/2	**62**	Mains: power factor cosphi	1/100, dim.less		
20/3	**63**	Mains: reactive power Q	1/10 kvar		
21/1	**64**	Generator: real power P	1/10 kW		
21/2	**65**	Generator: reactive power Q	1/10 kvar		
21/3	**65**	Mains: real power P	1/10 kW		
22/1	**67**	Generator: alarms 2		Bit 15	Generator - load imbalance, limit 1
				Bit 14	Generator - load imbalance, limit 2
				Bit 13	Generator - voltage asymmetry
				Bit 12	Generator - ground fault, limit 1
				Bit 11	Generator - ground fault, limit 2
				Bit 10	Generator field rotating wrong direction
				Bit 9	Generator load - limit 1
				Bit 8	Generator load - limit 2
				Bit 7	Generator - inverse-time overcurrent
				Bit 6	Internal
				Bit 5	Internal
				Bit 4	Internal
				Bit 3	Internal
				Bit 2	Internal
				Bit 1	Internal
				Bit 0	Internal
22/2	**68**	Real energy W	1/100 MWh	High word	
22/3	**69**			Low word	

MUX	No.	Content (words)	Engineering unit	Remark	
23/1	**70**	*LogicsManager*		Bit 15	Flag 1 is TRUE
				Bit 14	Flag 2 is TRUE
				Bit 13	Flag 3 is TRUE
				Bit 12	Flag 4 is TRUE
				Bit 11	Flag 5 is TRUE
				Bit 10	Flag 6 is TRUE
				Bit 9	Flag 7 is TRUE
				Bit 8	Flag 8 is TRUE
				Bit 7	Internal
				Bit 6	Internal
				Bit 5	Internal
				Bit 4	Internal
				Bit 3	Internal
				Bit 2	Internal
				Bit 1	Internal
				Bit 0	Internal
23/2	**71**	Reactive energy W	1/100 Mvarh	High word	
23/3	**72**			Low word	
24/1	**73**	Internal			
24/2	**74**	Generator: ground current	mA	High word	
24/3	**75**			Low word	
25/1	**76**	External discrete inputs with alarm class, status		Bit 15	Discrete input [DEx16]
				Bit 14	Discrete input [DEx15]
				Bit 13	Discrete input [DEx14]
				Bit 12	Discrete input [DEx13]
				Bit 11	Discrete input [DEx12]
				Bit 10	Discrete input [DEx11]
				Bit 9	Discrete input [DEx10]
				Bit 8	Discrete input [DEx09]
				Bit 7	Discrete input [DEx08]
				Bit 6	Discrete input [DEx07]
				Bit 5	Discrete input [DEx06]
				Bit 4	Discrete input [DEx05]
				Bit 3	Discrete input [DEx04]
				Bit 2	Discrete input [DEx03]
				Bit 1	Discrete input [DEx02]
				Bit 0	Discrete input [DEx01]
25/2	**77**	Internal			
25/3	**78**	Internal			

MUX	No.	Content (words)	Engineering unit	Remark	
26/1	**79**	External relay outputs, status		Bit 15	Relay output [REx16]
				Bit 14	Relay output [REx15]
				Bit 13	Relay output [REx14]
				Bit 12	Relay output [REx13]
				Bit 11	Relay output [REx12]
				Bit 10	Relay output [REx11]
				Bit 9	Relay output [REx10]
				Bit 8	Relay output [REx09]
				Bit 7	Relay output [REx08]
				Bit 6	Relay output [REx07]
				Bit 5	Relay output [REx06]
				Bit 4	Relay output [REx05]
				Bit 3	Relay output [REx04]
				Bit 2	Relay output [REx03]
				Bit 1	Relay output [REx02]
				Bit 0	Relay output [REx01]
26/2	**80**	External discrete inputs, status		Bit 15	Discrete input [DEx16]
				Bit 14	Discrete input [DEx15]
				Bit 13	Discrete input [DEx14]
				Bit 12	Discrete input [DEx13]
				Bit 11	Discrete input [DEx12]
				Bit 10	Discrete input [DEx11]
				Bit 9	Discrete input [DEx10]
				Bit 8	Discrete input [DEx09]
				Bit 7	Discrete input [DEx08]
				Bit 6	Discrete input [DEx07]
				Bit 5	Discrete input [DEx06]
				Bit 4	Discrete input [DEx05]
				Bit 3	Discrete input [DEx04]
				Bit 2	Discrete input [DEx03]
				Bit 1	Discrete input [DEx02]
				Bit 0	Discrete input [DEx01]
26/3	**81**	Internal			

MUX	No.	Content (words)	Engineering unit	Remark
Appendix starting from software version 1.0200				
27/1	**82**	Diagnostic Message 1		High word
27/2	**83**	1. SPN number		Low word
27/3	**84**	Diagnostic Message 1 HighByte → 1. FMT LowByte → 1. OC		Refer to page 38 for a description of Diagnostic Messages
28/1	**85**	Diagnostic Message 1		High word
28/2	**86**	2. SPN number		Low word
28/3	**87**	Diagnostic Message 1 HighByte → 2. FMT LowByte → 2. OC		
29/1	**88**	Diagnostic Message 1		High word
29/2	**89**	3. SPN number		Low word
29/3	**90**	Diagnostic Message 1 HighByte → 3. FMT LowByte → 3. OC		
30/1	**91**	Diagnostic Message 1		High word
30/2	**92**	4. SPN number		Low word
30/3	**93**	DM1 message HighByte → 4. FMT LowByte → 4. OC		
31/1	**94**	Diagnostic Message 1		High word
31/2	**95**	5. SPN number		Low word
31/3	**96**	Diagnostic Message 1 HighByte → 5. FMT LowByte → 5. OC		
32/1	**97**	Diagnostic Message 1		High word
32/2	**98**	6. SPN number		Low word
32/3	**99**	Diagnostic Message 1 HighByte → 6. FMT LowByte → 6. OC		
33/1	**100**	Diagnostic Message 1		High word
33/2	**101**	7. SPN number		Low word
33/3	**102**	Diagnostic Message 1 HighByte → 7. FMT LowByte → 7. OC		
34/1	**103**	Diagnostic Message 1		High word
34/2	**104**	8. SPN number		Low word
34/3	**105**	Diagnostic Message 1 HighByte → 8. FMT LowByte → 8. OC		
35/1	**106**	Diagnostic Message 1		High word
35/2	**107**	9. SPN number		Low word
35/3	**108**	Diagnostic Message 1 HighByte → 9. FMT LowByte → 9. OC		
36/1	**109**	Diagnostic Message 1		High word
36/2	**110**	10. SPN number		Low word
36/3	**111**	Diagnostic Message 1 HighByte → 10. FMT LowByte → 10. OC		

MUX	No.	Content (words)	Engineering unit	Remark	
37/1	**112**	190/61444 Engine Speed	0.1 rpm	High word	
37/2	**113**			Low word	
37/3	**114**	110/65262 Engine coolant temperature	°C		
38/1	**115**	247/65253 Total engine hours	1 h	High word	
38/2	**116**			Low word	
38/3	**117**	174/65262 Fuel temperature	1°C		
39/1	**118**	175/65262	0.01°C	High word	
39/2	**119**	Engine oil temperature		Low word	
39/3	**120**	100/65263 Engine oil pressure	1kPa		
40/1	**121**	183/65266 Fuel rate	0.01 l/h	High word	
40/2	**122**			Low word	
40/3	**123**	111/65263 Coolant level	0.1%		
41/1	**124**	91/61443 Throttle position	0.1%		
41/2	**125**	92/61443 Load at current speed	1%		
41/3	**126**	98/65263 Engine oil level	0.1%		
42/1	**127**	102/65270 Boost pressure	1kPa		
42/2	**128**	105/65270 Intake manifold temp.	1°C		
42/3	**129**	108/65269 Barometric pressure	0.1kPa		
43/1	**130**	172/65269 Air inlet temperature	1°C		
43/2	**131**	513/61444 Actual engine torque	1%		
43/3	**132**	from Deutz EMR ECU Motorstopinformation		Bit 0	no stop
				Bit 1	Engine safety
				Bit 2	CAN message engine stop request
				Bit 3	low oil pressure
				Bit 4	low oil level
				Bit 5	high coolant temp
				Bit 6	low coolant level
				Bit 7	intake manifold temp.
				Bit 8	reserved (Stop via SAE-J1587)
				Bit 9	reserved (Stop via VP2)"\
				FEFFh	Sensor defect
				FFFFh	Missing
44/1	**133**	173/65270	0.01°C	High word	
44/2	**134**	Exhaust gas temperature		Low word	
44/3	**135**	Intern			

MUX	No.	Content (words)	Engineering unit	Remark	
45/1	**136**	from S6 DLN2-Proprietary Low Engine Oil Level		Bit 0	Not Low Engine Oil Level
				Bit 1	Low Engine Oil Level
				Bit 2	Sensor defect
				Bit 3	Missing
		DLN2-Proprietary High Engine Oil Level		Bit 4	Not High Engine Oil Level
				Bit 5	High Engine Oil Level
				Bit 6	Sensor defect
				Bit 7	Missing
		DLN2-Proprietary Low Engine Oil Pressure		Bit 8	Not Low Engine Oil Pressure
				Bit 9	Low Engine Oil Pressure
				Bit 10	Sensor defect
				Bit 11	Missing
		DLN2-Proprietary High Engine Coolant Temperature		Bit 12	Not High Engine Coolant Temperature
				Bit 13	High Engine Coolant Temperature
				Bit 14	Sensor defect
				Bit 15	Missing
45/2	**137**	Diagnostic Message 1 Lamp Status Protect Lamp Status			
				Bit 0	Off
				Bit 1	On
				Bit 2	Missing
				Bit 3	Missing
		Amber Warning Lamp Status		Bit 4	Off
				Bit 5	On
				Bit 6	Missing
				Bit 7	Missing
		Red Stop Lamp Status		Bit 8	Off
				Bit 9	On
				Bit 10	Missing
				Bit 11	Missing
		Malfunction Indicator Lamp Status		Bit 12	Off
				Bit 13	On
				Bit 14	Missing
				Bit 15	Missing
45/3	**138**	Diagnostic Message 2 Lamp Status Protect Lamp Status			
				Bit 0	Off
				Bit 1	On
				Bit 2	Missing
				Bit 3	Missing
		Amber Warning Lamp Status		Bit 4	Off
				Bit 5	On
				Bit 6	Missing
				Bit 7	Missing
		Red Stop Lamp Status		Bit 8	Off
				Bit 9	On
				Bit 10	Missing
				Bit 11	Missing
		Malfunction Indicator Lamp Status		Bit 12	Off
				Bit 13	On
				Bit 14	Missing
				Bit 15	Missing

MUX	No.	Content (words)	Engineering unit	Remark
From GW4 no longer transmitted				
46/1	**139**	Diagnostic Message 2 message		High word
46/2	**140**	1. SPN number		Low word
46/3	**141**	Diagnostic Message 2 HighByte → 1. FMT LowByte → 1. OC		
47/1	**142**	Diagnostic Message 2		High word
47/2	**143**	2. SPN number		Low word
47/3	**144**	Diagnostic Message 2 HighByte → 2. FMT LowByte → 2. OC		
48/1	**145**	Diagnostic Message 2		High word
48/2	**146**	3. SPN number		Low word
48/3	**147**	Diagnostic Message 2 HighByte → 3. FMT LowByte → 3. OC		
49/1	**148**	Diagnostic Message 2		High word
49/2	**149**	4. SPN number		Low word
49/3	**150**	Diagnostic Message 2 HighByte → 4. FMT LowByte → 4. OC		
50/1	**151**	Diagnostic Message 2		High word
50/2	**152**	5. SPN number		Low word
50/3	**153**	Diagnostic Message 2 HighByte → 5. FMT LowByte → 5. OC		
51/1	**154**	Diagnostic Message 2		High word
51/2	**155**	6. SPN number		Low word
51/3	**156**	Diagnostic Message 2 HighByte → 6. FMT LowByte → 6. OC		
52/1	**157**	Diagnostic Message 2		High word
52/2	**158**	7. SPN number		Low word
52/3	**159**	Diagnostic Message 2 HighByte → 7. FMT LowByte → 7. OC		
53/1	**160**	Diagnostic Message 2		High word
53/2	**161**	8. SPN number		Low word
53/3	**162**	Diagnostic Message 2 HighByte → 8. FMT LowByte → 8. OC		
54/1	**163**	Diagnostic Message 2		High word
54/2	**164**	9. SPN number		Low word
54/3	**165**	Diagnostic Message 2 HighByte → 9. FMT LowByte → 9. OC		
55/1	**166**	Diagnostic Message 2		High word
55/2	**167**	10. SPN number		Low word
55/3	**168**	Diagnostic Message 2 HighByte → 10. FMT LowByte → 10. OC		
56/1	**169**	Hours until maintenance	h	
56/2	**170**	Hours of operation	h	High word
56/3	**171**	Hours of operation	h	Low word

MUX	No.	Content (words)	Engineering unit	Remark	
57/1	**172**	Days until maintenance	d		
57/2	**173**	Alarm 2			
		Amber warning lamp from ECU via J1939		Bit 0 (will only be enabled if the amber alarm monitoring function is enabled in the easYgen)	
		Red stop lamp from ECU via J1939		Bit 1 (will only be enabled if the red alarm monitoring function is enabled in the easYgen)	
		Internal		Bit 2 to 15	
57/3	**174**	Number of starts			
58/1	**175**	Ground current, measured	mA	High word	
58/2	**176**	Ground current, measured	mA	Low word	
58/3	**177**	Internal			
59/1	**178**	Hours of operation in 1/100h resolution	1/100 h	High word	
59/2	**179**			Low word	
59/3	**180**	Internal			
60/1	**181**	Free adjustable hours counter in 1/100h resolution	1/100 h	High word	
60/2	**182**			Low word	
60/3	**183**	Internal			
61/1	**184**	Failurecodes from MTU ADEC ECU		Failure range	0 to 64768
				FEFFh	Sensor defect
				FFFFh	Missing
61/2	**185**	Internal			
61/3	**186**	Internal			

Remote Control Telegram

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	Remote acknwledgement : reset alarm messages (rise of the pulse)			Transmit first a 0, then a 1 to acknowledge
		Bit 3	Must always be configured to 0			
		Bit 2	Must always be configured to 0			
		Bit 1	Remote stop (rise of the pulse)			Transmit first a 0, then a 1 to stop
		Bit 0	Remote start (rise of the pulse)			Transmit first a 0, then a 1 to start

Remark : Bits may also be used for a different purpose than start/stop.

Bit 0 "Remote start"
If this bit changes from Low to High, the easYgen enables the remote start (logical command variable 04.13). The condition of the start command is stored and may be used as command variable for the *LogicsManager*.

Bit 1 "Remote stop"
If this bit changes from Low to High, the easYgen enables the remote stop (logical command variable 04.13). The condition of the stop command is stored and may be used as command variable for the *LogicsManager*.

Bit 4 "Remote acknowledgement: reset alarm messages"
This bit controls the logical command variable 04.14.
This command must be executed twice.
The first rise of the pulse resets the horn and the second rise of the pulse acknowledges a fault, which is not present anymore.

Appendix B. CANopen

Description of the Common Data Types

Structure of the PDO-COB-ID Entry (UNSIGNED32)

MSB					LSB
Bits	31	30	29	28-11	10 – 0
11 bit ID	0/1	0	0	all 0	11 bit identifier
29 bit ID	0/1	0	1	29 bit identifier	

Description of the PDO-COB-ID entry

Bit number	Value	Description
31 (MSB)	0	PDO exists / is valid
	1	PDO does not exist / is invalid
30	0	Device does not generate SYNC message
	1	Device generates SYNC message
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28 – 11	0 X	If bit 29=0 and if bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	Bits 10-0 of SYNC-COB-ID

Transmission Types (PDO Transmission)

	cyclically	continuously	synchronous	asynchronous	RTR only
0 *	--	X	X	--	--
1-240	X	--	X	--	--
241-251	----------	----------	reserved	----------	----------
252 *	--	--	X		X
253 *	--	--	--	X	X
254	--	--	--	X	--
255	--	--	--	X	--

* not supported

Description of the Object Parameter

Object 1000h: Device Type

This contains information about the type of the participant.

Object description

Index 1000h
Name ..... Device Type
Object code ..... VAR
Data type UNSIGNED32
Category ..... obligatory

Entry description

Access ..... Read Only
PDO figure ..... no
Value range UNSIGNED32
Default value ..... 0 h no standard profile

Object 1001h: Error Register

This object is an error register for the participant.

Object description

Index 1001h
Name ..... Error Register
Object code ..... VAR
Data type UNSIGNED8
Category ..... obligatory

Entry description

Access ..... Read Only
PDO figure ..... no
Value range UNSIGNED8
Default value ..... no

Note

This object is always value 0.

Object 1005h: COB-ID SYNC Message

The index 1005h defines the COB-ID of the synchronization object (SYNC).

Description of the SYNC-COB-ID entry (UNSIGNED32)

MSB					LSB
Bits	31	30	29	28-11	10 – 0
11 Bit-ID	X	0/1	0	all 0	11-bit Identifier
29 Bit-ID	X	0/1	1	29-bit Identifier	

Description of the SYNC-COB-ID entry

Bit number	Value	Description
31 (MSB)	0/1	0 = valid / 1 = invalid
30	0	Device does not generate SYNC message
	1	Device generates SYNC message
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2,0B)
28 – 11	0 X	If bit 29=0 and if bit 29=1: bits 28-11 of 29-bit-SYNC-COB-ID
10-0 (LSB)	X	Bits 10-0 of SYNC-COB-ID

Object description

Index.....1005h
Name.....COB-ID SYNC
Object code.....VAR
Data type.....UNSIGNED32

Entry description

Access.....Read/Write
PDO figure.....no
Value range.....UNSIGNED32
Default value.....80 hex

Note

Bit 31-29 are ignored. Writing these bits does not cause faults. The bit 28-11 should be configured to 0. This parameter can be configured using the parameter COB-ID SYNC Message. If a SYNC message is to be sent the PDO can be configured in that way that it contains no values.

Object 1017h: Producer Heartbeat Time

The object Producer Heartbeat Time defines the heartbeat cycle time in ms. If no Producer Heartbeat (NMT Error Control) is to be sent, this is to be configured to 0.

Object description

Index 1017h
Name ..... Producer Heartbeat Time
Object code ..... VAR
Data type UNSIGNED16

Entry description

Access ..... Read/Write
PDO figure ..... no
Value range UNSIGNED16
Default value 240

Note

The time is extended to the next full 20 ms. If the time is 0, the (NMT Error Control) will be sent as response to a remote frame.

Object 1018h: Identity Object

The object contains common information of one participant.

Object description

Index 1018h
Name ..... Identity Object
Object code ..... RECORD
Data type ..... Identity
Category ..... obligatory

Entry description

Sub index 0h

Description ..... Number of entries
Entry category ..... obligatory
Access ..... Read Only
PDO figure ..... no
Value range 1
Default value 1

Sub index 1h

Description ..... Vendor ID
Entry category ..... obligatory
Access ..... Read Only
PDO figure ..... no
Value range UNSIGNED32
Default value 0

Object 1200h – 1201h: Server SDO Parameter

Objects are not supported.
The receive SDO is: 600h+Node-ID
The transmit SDO for answers is 580h+Node-ID
The Node ID can be entered using the parameter "Unit number".

Object 1400h – 141Fh: Receive PDO Communication Parameter

This object contains the communication parameter for the PDOs that can be received from the participant. The sub index 0h contains the number of valid entries within the communication recording. The sub index 1h contains the COB ID of the PDO. The interpretation of the entry occurs according to the tables "Structure of the PDO-COB-ID entry" and the "Description of the POD-COB-ID entry".

Object description

Index .....1400h — 141Fh
Name.....Receive PDO parameter
Object code.....RECORD
Data type.....PDO CommPar
Category .....conditioned; obligatory for every supported PDO

Entry description

Sub index 0h
Description .....Largest Sub index supported
Entry category.....obligatory
Access.....Read Only
PDO figure.....no
Value range.....2

Sub index 1h

Description .....COB-ID used by PDO
Entry category.....obligatory
Access.....Read Only; Read/Write if variable COB-ID is supported
PDO figure.....no
Value range.....UNSIGNED32 (Table 54)
Default value.....Index 1400h: 200h + Node-ID,
Index 1401h: 300h + Node-ID,
Index 1402h: 400h + Node-ID,
Index 1403h: 500h + Node-ID,
Index 1404h - 15FFh: disabled

Sub index 2h

Description .....Transmission type
Entry category.....obligatory
Access.....Read Only
PDO figure.....no
Value range.....UNSIGNED8 (Table 55)
Default value.....(Device Profile dependent)

Note

The device possesses only two RPDOs. Therefore the objects 1402h-141Fh are not available.

Sub index 1h

The bits 30-29 were ignored. Writing these bits do not cause faults. The bits 28-11 should be configured to 0. This value can be set in the display mask "COB-ID" in sub menu CAN-OPEN RPDO 1 / 2.

Sub index 2h

This value is always set 0xFF.

Object 1600h – 161Fh: Receive PDO Mapping Parameter

Is not used. The receive PDOs can be assigned to defined functions. The corresponding parameter can be set in the display screen "Function" in sub menu CAN-OPEN RPDO 1 / 2.

Object 1800h – 181Fh: Transmit PDO Communication Parameter

Includes the communication parameter for the PDOs that can be sent from the participant.

Object description

Index ..... 1800h — 181Fh
Name ..... Transmit PDO parameter
Object code ..... RECORD
Data type ..... PDO CommPar
Category ..... conditioned; obligatory for every supported PDO

Entry description

Sub index 0h

Description ..... Largest Sub index supported
Entry category ..... obligatory
Access ..... Read Only
PDO figure ..... no
Value range 5

Sub index 1h

Description ..... COB-ID used by PDO
Entry category ..... obligatory
Access ..... Read Only; Read/Write if COB-ID can be configured
PDO figure ..... no
Value range ..... UNSIGNED32 (Figure 65)
Default value: ..... Index 1800h: 181h,
Index 1801h: 281h,
Index 1802h: 381h,
Index 1803h: 481h, because Default value for Node-ID is 1.

Sub index 2h

Description ..... Transmission type
Entry category ..... obligatory
Access ..... Read Only; Read/Write if transmission type can be changed
PDO figure ..... no
Value range ..... UNSIGNED8 (Table 54)
Default value 0

Sub index 5h

Description ..... Event timer
Entry category ..... optional
Access ..... Read/Write
PDO figure ..... no
Value range ..... 0 = not used UNSIGNED16
Default value 20

Note

Sub index 1h

The bits 31-29 were ignored. Writing these bits does not cause faults. The bits 28-11 should be configured to 0. This sub index can be set in the display screens "COB-ID" in sub menu CAN-OPEN TPDO 1 / 2 / 3 / 4.

Sub index 2h

Value	Function
0	A PDO will not be sent
1-240	A PDO will be sent as answer to a SYNC message
241-251	A PDO will not be sent
252-253	A PDO will not be sent
254-255	A PDO will be sent cyclically

This sub index does not change the PDO communication parameter screen. This sub index can be set in the display screen "Transmission type" in sub menu CAN-OPEN TPDO 1 / 2 / 3 / 4.

Sub index 5h

The time is rounded up to the next full 5 ms. The sub index can be set in the display screen "Event-timer" in sub menu CAN-OPEN TPDO 1 / 2 / 3 / 4.

Object 1A00h – 1A1Fh: Transmit PDO Mapping Parameter

The mapping for the PDOs, which the participant can send, is located here. An exact description of the entries can be found in the chapter "Parameter description".

CAUTION

The parameter can be configured only if the respective PDO is valid (Object 1800 Sub index 1 Bit 31 is set).

Object description

Index .....1A00h — 1A1Fh
Name.....Transmit PDO mapping
Object code.....RECORD
Data type.....PDO figure
Category .....conditioned; obligatory for every supported PDO

Entry description

Sub index 0h

Description ..... number of mapped application objects in PDO
Entry category ..... obligatory
Access ..... Read Only; Read/Write if dynamic mapping is supported
PDO figure ..... no
Value range 4
Default value 4

Sub index 1h - 4h

Description ..... PDO mapping for the n^{th} application object to be mapped
Entry category ..... conditioned, dependent on the number and size of the objects
Access ..... Read/Write
PDO figure ..... no
Value range UNSIGNED32
Default value ..... (Device profile dependent)

Note

Sub index 0h

The sub index 0 cannot be changed. Writing does not cause fault messages however the value will not be saved. For configuration of the other sub indexes the sub index 0h has to be set **not** 0.

Sub index 1h-4h

You have to enter the object numbers from the EDS file into the sub indexes 1h-4h. The sub indexes 1h-4h can be set in the display masks "1-4 Mapped Object" in sub menu CAN-OPEN TPDO 1 / 2 / 3 / 4.

CAUTION

With configuration over CAN open the object ID is to be used (see EDS file).

With configuration over display/LeoPC the parameter number is to be used (see "CANopen: Mapping-Parameter" after page 70.)

Data Format of Different Functions

Depending on the selected RPDO function a different data format will be expected.

Receiving Messages

1.IKD / 2.IKD

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
01	Bit 0 DI1 Bit 1 DI 2 + + + Bit 7 DI 8	not analyzed	not analyzed	not analyzed	not analyzed	not analyzed	not analyzed

Phoenix16

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Bit 0 DI 1 Bit 1 DI 2 + + + Bit 7 DI 8	Bit 0 DI 9 Bit 1 DI 10 + + + Bit 7 DI 16	not analyzed	not analyzed	not analyzed	not analyzed	not analyzed	not analyzed

CAUTION

Please note for combination of the different functions.

CAUTION

Configuration of the Phoenix terminal, if the easYgen is not CAN open master.

If the discrete inputs of the Phoenix terminal shall be evaluated by the easYgen, it must be configured this way that the corresponding discrete inputs in byte 1 and byte 2 are available for the received PDO. This PDO must be sent independently from the terminal. The easYgen does not pick up PDOs with remote frames.

The receiving PDO of the Phoenix terminal and the corresponding transmitting PDO of the easYgen must be adjusted on both units.

Definition of Protocol Descriptions

If in a PDO a protocol number is entered as 1. Mapped object, a data array with 8x unsigned8 is sent.

The denotation is:

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 = 10Ah of type UNSIGNED16 is transmitted on the bus in two octets, first 0Ah and then 01h.

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}

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 = FEF6h of type SIGNED16 is transmitted in two octets, first F6h and then FEh.

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}

Transmission Telegram

NOTE

When using the listed Mapped Objects instead of the complete transmission telegram, the refresh rate of the individual messages may be reduced.

Data Protocol Parameter No.3190/Object 2C76h

In this protocol the LeoPC display messages were sent:

Parameter 3190, Object 2C76h							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4003		Parameter no. 108				-Internal-
1	Parameter no. 144		Parameter no. 114				-Internal-
2	Parameter no. 147		Parameter no. 109				-Internal-
3	Parameter no. 160		Parameter no. 115				-Internal-
4	Parameter no. 141		Parameter no. 110				-Internal-
5	Parameter no. 10100		Parameter no. 116				-Internal-
6	Parameter no. 10110		Parameter no. 118				-Internal-
7	Parameter no. 10111		Parameter no. 121				-Internal-
8	Parameter no. 10112		Parameter no. 119				-Internal-
9	Parameter no. 10106		Parameter no. 122				-Internal-
10	Parameter no. 10107		Parameter no. 120				-Internal-
11	Parameter no. 10201		Parameter no. 123				-Internal-
12	Parameter no. 10131		Parameter no. 111				-Internal-
13	Parameter no. 10139		Parameter no. 112				-Internal-
14	Parameter no. 10133		Parameter no. 113				-Internal-
15	Parameter no. 10134		Parameter no. 134				-Internal-
16	Parameter no. 10135		Parameter no. 136				-Internal-
17	Parameter no. 10137		Parameter no. 135				-Internal-
18	Parameter no. 10141		Parameter no. 140				-Internal-
19	Parameter no. 10200		Parameter no. 150				-Internal-
20	Parameter no. 10306		Parameter no. 10301		Parameter no. 10305		-Internal-
21	Parameter no. 10302		Parameter no. 10303		Parameter no. 10304		-Internal-
22	Parameter no. 10138		Parameter no. 2520				-Internal-
23	Parameter no. 10140		Parameter no. 2522				-Internal-
24	Parameter no. 10202		Parameter no. 159				-Internal-
25	Parameter no. 10307		Parameter no. 10308				-Internal-
26	Parameter no. 8003		Parameter no. 8013		---		-Internal-

Appendix starting from software version 1.0200					
27	Parameter-No. 15400		P.-No.: 15401	P.-No.: 15402	-Internal-
28	Parameter-No. 15403		P.-No.: 15404	P.-No.: 15405	-Internal-
29	Parameter-No. 15406		P.-No.: 15407	P.-No.: 15408	-Internal-
30	Parameter-No. 15409		P.-No.: 15410	P.-No.: 15411	-Internal-
31	Parameter-No. 15412		P.-No.: 15413	P.-No.: 15414	-Internal-
32	Parameter-No. 15415		P.-No.: 15416	P.-No.: 15418	-Internal-
33	Parameter-No. 15419		P.-No.: 15420	P.-No.: 15421	-Internal-
34	Parameter-No. 15422		P.-No.: 15423	P.-No.: 15424	-Internal-
35	Parameter-No. 15425		P.-No.: 15426	P.-No.: 15427	-Internal-
36	Parameter-No. 15428		P.-No.: 15429	P.-No.: 15430	-Internal-
37	Parameter-No. 15200		P.-No.: 15202		-Internal-
38	Parameter-No. 15201		P.-No.: 15203		-Internal-
39	Parameter-No. 15204		P.-No.: 15205		-Internal-
40	Parameter-No. 15211		P.-No.: 15206		-Internal-
41	Parameter-No. 15207	Parameter-No. 15208	P.-No.: 15210		-Internal-
42	Parameter-No. 15214	Parameter-No. 15215	P.-No.: 15212		-Internal-
43	Parameter-No. 15213	Parameter-No. 15209	P.-No.: 15304		-Internal-
44	Parameter-No. 15216		---		-Internal-
45	Parameter-No. 15305	Parameter-No. 15395	P.-No.: 15445		-Internal-

From GW4 no longer transmitted							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
46	Parameter-No. 15450				P.-No.: 15451	P.-No.: 15452	-Internal-
47	Parameter-No. 15453				P.-No.: 15454	P.-No.: 15455	-Internal-
48	Parameter-No. 15456				P.-No.: 15457	P.-No.: 15458	-Internal-
49	Parameter-No. 15459				P.-No.: 15460	P.-No.: 15461	-Internal-
50	Parameter-No. 15462				P.-No.: 15463	P.-No.: 15464	-Internal-
51	Parameter-No. 15465				P.-No.: 15466	P.-No.: 15467	-Internal-
52	Parameter-No. 15468				P.-No.: 15469	P.-No.: 15470	-Internal-
53	Parameter-No. 15471				P.-No.: 15472	P.-No.: 15473	-Internal-
54	Parameter-No. 15474				P.-No.: 15475	P.-No.: 15476	-Internal-
55	Parameter-No. 15477				P.-No.: 15478	P.-No.: 15479	-Internal-
56	Parameter-No. 2558		P.-No.: 2552				-Internal-
57	Parameter-No. 2556		P.-No.: 10149		P.-No.: 2540		-Internal-
58	Parameter-No. 161				---		-Internal-
59	Parameter-No. 2568				- Internal -	- Internal -	- Internal -
60	Parameter-No. 2570				- Internal -	- Internal -	- Internal -
61	Parameter-No. 15109		- Internal -	- Internal -	- Internal -	- Internal -	- Internal -

NOTE

The GW4 does not transmit the following data-MUX: 46 to 55.

Data Protocol Parameter No.3193/Object 2C79h – J1939 Values

If the object 2C79h is read out, the protocol known value is replaced.
Protocol known value: 4103
This parameter is available starting from software version 1.0200 and includes the standard values of J1939 protocol.

Parameter 3193, Object 2C79h							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4103		---	---	---	---	-Internal-
1	Parameter-No. 15400				P.-No.: 15401	P.-No.: 15402	-Internal-
2	Parameter-No. 15403				P.-No.: 15404	P.-No.: 15405	-Internal-
3	Parameter-No. 15406				P.-No.: 15407	P.-No.: 15408	-Internal-
4	Parameter-No. 15409				P.-No.: 15410	P.-No.: 15411	-Internal-
5	Parameter-No. 15412				P.-No.: 15413	P.-No.: 15414	-Internal-
6	Parameter-No. 15415				P.-No.: 15416	P.-No.: 15418	-Internal-
7	Parameter-No. 15419				P.-No.: 15420	P.-No.: 15421	-Internal-
8	Parameter-No. 15422				P.-No.: 15423	P.-No.: 15424	-Internal-
9	Parameter-No. 15425				P.-No.: 15426	P.-No.: 15427	-Internal-
10	Parameter-No. 15428				P.-No.: 15429	P.-No.: 15430	-Internal-
11	Parameter-No. 15450				P.-No.: 15451	P.-No.: 15452	-Internal-
12	Parameter-No. 15453				P.-No.: 15454	P.-No.: 15455	-Internal-
13	Parameter-No. 15456				P.-No.: 15457	P.-No.: 15458	-Internal-
14	Parameter-No. 15459				P.-No.: 15460	P.-No.: 15461	-Internal-
15	Parameter-No. 15462				P.-No.: 15463	P.-No.: 15464	-Internal-
16	Parameter-No. 15465				P.-No.: 15466	P.-No.: 15467	-Internal-
17	Parameter-No. 15468				P.-No.: 15469	P.-No.: 15470	-Internal-
18	Parameter-No. 15471				P.-No.: 15472	P.-No.: 15473	-Internal-
19	Parameter-No. 15474				P.-No.: 15475	P.-No.: 15476	-Internal-
20	Parameter-No. 15477				P.-No.: 15478	P.-No.: 15479	-Internal-
21	Parameter-No. 15395		Parameter-No. 15445		---		-Internal-
22	Parameter-No. 15200				Parameter-No. 15202		-Internal-
23	Parameter-No. 15201				Parameter-No. 15203		-Internal-
24	Parameter-No. 15204				Parameter-No. 15205		-Internal-
25	Parameter-No. 15211				Parameter-No. 15206		-Internal-
26	Parameter-No. 15207		Parameter-No. 15208		Parameter-No. 15120		-Internal-
27	Parameter-No. 15214		Parameter-No. 15215		Parameter-No. 15212		-Internal-
28	Parameter-No. 15213		Parameter-No. 15209		---		-Internal-
29	Parameter-No. 15216				---		-Internal-

Data Protocol Parameter No.3194/Object 2C7Ah – Scania S6 DLN2 Proprietary Messages

If the object 2C7Ah is read out, the protocol known value is replaced.
Protocol known value: 4104
This parameter is available starting from software version 1.0200 and includes the appendix of the J1939 protocol of Scania S6.

Parameter 3194, Object 2C7Ah							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4104		---	---	---	---	-Internal-
1	Parameter-No. 15305		---		---		-Internal-

Data Protocol Parameter No. 3195/Object 2C7Bh – Deutz EMR Proprietary Messages

If the object 2C7Ch is read out, the protocol known value is replaced.
Protocol known value: 4105
This parameter is available starting from software version 1.0200 and includes the appendix of the J1939 protocol of Deutz EMR.

Parameter 3195, Object 2C7Bh							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4105		---	---	---	---	-Internal-
1	Parameter-No. 15304		---		---		-Internal-

Data Protocol Parameter No. 3199/Object 2C7Fh – Mains values

If the object 2C7Fh is read out, the protocol known value is replaced
Protocol known value : 4107
This protocol replaces "Data Protocol Parameter No. 3192/Object 2C78h" for Software versions higher than V2.1000.

Parameter 3199, Object 2C7Fh							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4107		---	---	---	---	- Internal -
1	Parameter no. 118				Parameter no. 147		- Internal -
2	Parameter no. 119				Parameter no. 141		- Internal -
3	Parameter no. 120				Parameter no. 10100		- Internal -
4	Parameter no. 121				Parameter no. 10110		- Internal -
5	Parameter no. 122				Parameter no. 10111		- Internal -
6	Parameter no. 123				Parameter no. 10112		- Internal -
7	Parameter no. 140				Parameter no. 10135		- Internal -
8	Parameter no. 150				---	---	- Internal -
9	Parameter no. 2520				---	---	- Internal -
10	Parameter no. 2522				Parameter no. 2540		- Internal -
11	Parameter no. 2552				---	---	- Internal -
12	Parameter no. 2568				---	---	- Internal -
13	Parameter no. 2570				---	---	- Internal -

Data Protocol Parameter No. 15600/Object 5CF0h – Generator values

If the object 5CF0h is read out, the protocol known value is replaced
Protocol known value : 4108
This protocol replaces "Data Protocol Parameter No. 3191/Object 2C77h" for Software versions higher than V2.1000.

Parameter 15600, Object 5CF0h							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4108		---	---	---	---	- Internal -
1	Parameter no. 108				Parameter no. 144		- Internal -
2	Parameter no. 109				Parameter no. 160		- Internal -
3	Parameter no. 110				Parameter no. 10134		- Internal -
4	Parameter no. 111				---	---	- Internal -
5	Parameter no. 112				Parameter no. 10131		- Internal -
6	Parameter no. 113				Parameter no. 10132		- Internal -
7	Parameter no. 114				Parameter no. 10133		- Internal -
8	Parameter no. 115				Parameter no. 10141		- Internal -
9	Parameter no. 116				Parameter no. 10137		- Internal -
10	Parameter no. 159				Parameter no. 10200		- Internal -
11	Parameter nn. 135				Parameter no. 10201		- Internal -
12	Parameter no. 136				Parameter no. 10306		- Internal -
13	Parameter no. 10203		Parameter no. 10149		---	---	- Internal -

Data Protocol Parameter No. 15601/Object 5CF1h – easYlite-100 values

If the object 5CF1h is read out, the protocol known value is replaced
Protocol known value : 4109
This protocol replaces "Data Protocol Parameter No. 3196/Object 2C7Ch" for Software versions higher than V2.1000.

Parameter 15601, Object 5CF1h							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4109		---		---		- Internal -
1	Parameter no. 10134		Parameter no. 10138		Parameter no. 10135		- Internal -
2	Parameter no. 10201		Parameter no. 10133		Parameter no. 10131		- Internal -
3	Parameter no. 10137		Parameter no. 10136		Parameter no. 10200		- Internal -
4	Parameter no. 10146		Parameter no. 10147		Parameter no. 10140		- Internal -
5	Parameter no. 10148		Parameter no. 10132		Parameter no. 16377		- Internal -
6	Parameter no. 10149		Parameter no. 10203		---		- Internal -

Data Protocol Parameter No. 15602/Object 5CF2h – ADEC Values

If the object 5CF2h is read out, the protocol known value is replaced
Protocol known value : 4110

Parameter 15602, Object 5CF2h							
MUX	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**	**Byte 8**
0	4110		---		---		- Internal -
1	Parameter no. 15109		---		---		- Internal -

CANopen: Mapping-Parameter

Parameter no.	Object-ID	Name	Unit	Data type	Note
108	206Ch	Generator: Voltage V_{L12}	1/10 V	signed32	
109	206Dh	Generator: Voltage V_{L23}	1/10 V	signed32	
110	206Eh	Generator: Voltage V_{L31}	1/10 V	signed32	
111	206Fh	Generator: Current I_{L1}	mA	signed32	
112	2070h	Generator: Current I_{L2}	mA	signed32	
113	2071h	Generator: Current I_{L3}	mA	signed32	
114	2072h	Generator: Voltage V_{L1N}	1/10 V	signed32	
115	2073h	Generator: Voltage V_{L2N}	1/10 V	signed32	
116	2074h	Generator: Voltage V_{L3N}	1/10 V	signed32	
118	2076h	Mains: Voltage V_{L12}	1/10 V	signed32	
119	2077h	Mains: Voltage V_{L23}	1/10 V	signed32	
120	2078h	Mains: Voltage V_{L31}	1/10 V	signed32	
121	2079h	Mains: Voltage V_{L1N}	1/10 V	signed32	
122	207Ah	Mains: Voltage V_{L2N}	1/10 V	signed32	
123	207Bh	Mains: Voltage V_{L3N}	1/10 V	signed32	
134	2086h	Mains: Current I_{L1}	mA	signed32	
135	2087h	Generator: Real power P	W	signed32	
136	2088h	Generator: Reactive power Q	var	signed32	
140	208Ch	Mains: Real power P_{L1}	W	signed32	
141	208Dh	Mains: Power factor $\cos\varphi_{L1}$	1/1000, dim.less	signed16	
144	2090h	Generator: Frequency	1/100 Hz	signed16	
147	2093h	Mains: Frequency f_{123}	1/100 Hz	signed16	
150	2096h	Mains: Reactive power Q	var	signed32	
159	209Fh	Generator: Calculated ground current	mA	signed32	
160	20A0h	Generator: Power factor $\cos\varphi_{L1}$	1/1000, dimls.	signed16	
161	20A1h	Generator: Measured ground current	mA	signed32	
2520	29D8h	Real energy	1/100 MWh	unsigned32	
2522	29DAh	Reactive energy	1/100 Mvarh	unsigned32	
2540	29ECh	Number of engine starts	---	unsigned16	
2552	29F8h	Hours of operation	h	unsigned32	
2556	29FCh	Days until maintenance	d	unsigned16	
2558	29FEh	Hours until maintenance	h	unsigned16	
2568	2A08h	Hours of operation	1/100 h	unsigned32	
2570	2A0Ah	Free adjustable hours counter	1/100 h	unsigned32	
3190	2C76h	LeoPC	---	unsigned64	Data protocol
3193	2C79h	Standard values J1939		unsigned64	Data protocol
3194	2C7Ah	Additions J1939 for S6		unsigned64	Data protocol
3195	2C7Bh	Additions J1939 for EMR		unsigned64	Data protocol
3199	2C7Fh	Mains values		unsigned64	Data protocol

Parameter no.	Object-ID	Name		Unit	Data type	Note
8000	3F40h	always 0			unsigned16	
8001	3F41h	Output of the 1st IKD1		Bit field	unsigned16	
		Bit 15	Relay output [REx08]			
		Bit 14	Relay output [REx07]			
		Bit 13	Relay output [REx06]			
		Bit 12	Relay output [REx05]			
		Bit 11	Relay output [REx04]			
		Bit 10	Relay output [REx03]			
		Bit 9	Relay output [REx02]			
		Bit 8	Relay output [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 1			
8002	3F42h	Outputs of the 2nd IKD1		Bit field	unsigned16	
		Bit 15	Relay output [REx16]			
		Bit 14	Relay output [REx15]			
		Bit 13	Relay output [REx14]			
		Bit 12	Relay output [REx13]			
		Bit 11	Relay output [REx12]			
		Bit 10	Relay output [REx11]			
		Bit 9	Relay output [REx10]			
		Bit 8	Relay output [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 1			
8003	3F43h	External relay outputs, status		Bit field	unsigned16	
		Bit 15	Relay output [REx16]			
		Bit 14	Relay output [REx15]			
		Bit 13	Relay output [REx14]			
		Bit 12	Relay output [REx13]			
		Bit 11	Relay output [REx12]			
		Bit 10	Relay output [REx11]			
		Bit 9	Relay output [REx10]			
		Bit 8	Relay output [REx09]			
		Bit 7	Relay output [REx08]			
		Bit 6	Relay output [REx07]			
		Bit 5	Relay output [REx06]			
		Bit 4	Relay output [REx05]			
		Bit 3	Relay output [REx04]			
		Bit 2	Relay output [REx03]			
		Bit 1	Relay output [REx02]			
		Bit 0	Relay output [REx01]			

Parameter no.	Object-ID	Name		Unit	Data type	Note
8013	3F43h	External discrete inputs, status		Bit field	unsigned16	
		Bit 15	Discrete input [DEx16]			
		Bit 14	Discrete input [DEx15]			
		Bit 13	Discrete input [DEx14]			
		Bit 12	Discrete input [DEx13]			
		Bit 11	Discrete input [DEx12]			
		Bit 10	Discrete input [DEx11]			
		Bit 9	Discrete input [DEx10]			
		Bit 8	Discrete input [DEx09]			
		Bit 7	Discrete input [DEx08]			
		Bit 6	Discrete input [DEx07]			
		Bit 5	Discrete input [DEx06]			
		Bit 4	Discrete input [DEx05]			
		Bit 3	Discrete input [DEx04]			
		Bit 2	Discrete input [DEx03]			
		Bit 1	Discrete input [DEx02]			
		Bit 0	Discrete input [DEx01]			
10100	4774h	Engine speed		RPM	unsigned16	
10106	---	Discrete inputs, status		Bit field	unsigned16	
		Bit 15	Discrete input [D1]			
		Bit 14	Discrete input [D2]			
		Bit 13	Discrete input [D3]			
		Bit 12	Discrete input [D4]			
		Bit 11	Discrete input [D5]			
		Bit 10	Discrete input [D6]			
		Bit 9	Discrete input [D7]			
		Bit 8	Discrete input [D8]			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			
10107	---	Relay outputs, status		Bit field	unsigned16	
		Bit 15	Relay output [R01]			
		Bit 14	Relay output [R02]			
		Bit 13	Relay output [R03]			
		Bit 12	Relay output [R04]			
		Bit 11	Relay output [R05]			
		Bit 10	Relay output [R06]			
		Bit 9	Relay output [R07]			
		Bit 8	Relay output [R08]			
		Bit 7	Relay output [R09]			
		Bit 6	Relay output [R10]			
		Bit 5	Relay output [R11]			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			

Parameter no.	Object-ID	Name		Unit	Data type	Note
10110	477Eh	Battery voltage		1/10 V	unsigned16	
10111	477Fh	Analog input [T1]			unsigned16	
10112	4780h	Analog input [T2]			unsigned16	
10131	4793h	Alarm classes		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	Alarm class F			
		Bit 4	Alarm class E			
		Bit 3	Alarm class D			
		Bit 2	Alarm class C			
		Bit 1	Alarm class B			
		Bit 0	Alarm class A			
10132	4794h	Latched alarm bits discrete input		Bit field	unsigned16	
		Bit 15	Discrete input [DI1]			
		Bit 14	Discrete input [DI2]			
		Bit 13	Discrete input [DI3]			
		Bit 12	Discrete input [DI4]			
		Bit 11	Discrete input [DI5]			
		Bit 10	Discrete input [DI6]			
		Bit 9	Discrete input [DI7]			
		Bit 8	Discrete input [DI8]			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			
10133	4795h	Alarms 1		Bit field	unsigned16	
		Bit 15	Overspeed, limit 1			
		Bit 14	Overspeed, limit 2			
		Bit 13	Underspeed, limit 1			
		Bit 12	Underspeed, limit 2			
		Bit 11	Unintended stop			
		Bit 10	Alarm speed detection			
		Bit 9	Shutdown malfunction			
		Bit 8	GCB fail to close			
		Bit 7	GCB fail to open			
		Bit 6	MCB fail to close			
		Bit 5	MCB fail to open			
		Bit 4	-Internal-			
		Bit 3	Start failure			
		Bit 2	Maintenance call "Days expired"			
		Bit 1	Maintenance call "Hours expired"			
		Bit 0	-Internal-			

Parameter no.	Object-ID	Name		Unit	Data type	Note
10134	4796h	Generator, watchdog 1		Bit field	unsigned16	
		Bit 15	Generator, overfrequency, limit 1			
		Bit 14	Generator, overfrequency, limit 2			
		Bit 13	Generator, underfrequency, limit 1			
		Bit 12	Generator, underfrequency, limit 2			
		Bit 11	Generator, overvoltage, limit 1			
		Bit 10	Generator, overvoltage, limit 2			
		Bit 9	Generator, undervoltage, limit 1			
		Bit 8	Generator, undervoltage, limit 2			
		Bit 7	Generator, overcurrent, limit 1			Time-overcurrent
		Bit 6	Generator, overcurrent, limit 2			Time-overcurrent
		Bit 5	Generator, overcurrent, limit 3			Time-overcurrent
		Bit 4	Generator, rev/red power, limit 1			Rev/red load
		Bit 3	Generator, rev/red power, limit 2			Rev/red load
		Bit 2	Generator, overload, limit 1			
		Bit 1	Generator, overload, limit 2			
		Bit 0	-Internal-			
10135	4797h	Mains, watchdog 1		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			Time-overcurrent
		Bit 6	Mains, overfrequency			For AMF only
		Bit 5	Mains, underfrequency			For AMF only
		Bit 4	Mains, overvoltage			For AMF only
		Bit 3	Mains, undervoltage			For AMF only
		Bit 2	Mains, rotating field failure			
		Bit 1	Mains, load limit 1			*LogicsManager*
		Bit 0	Mains, load limit 2			*LogicsManager*

Parameter no.	Object-ID	Name		Unit	Data type	Note
10136	4798h	Latched alarm bits analog input		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	Alarm bit monitoring analog input 2 threshold 2			
		Bit 6	Alarm bit monitoring analog input 2 threshold 1			
		Bit 5	Alarm bit monitoring analog input 1 threshold 2			
		Bit 4	Alarm bit monitoring analog input 1 threshold 1			
		Bit 3	Alarm bit monitoring battery voltage overvoltage threshold 2			
		Bit 2	Alarm bit monitoring battery voltage undervoltage threshold 2			
		Bit 1	Alarm bit monitoring battery voltage overvoltage threshold 1			
		Bit 0	Alarm bit monitoring battery voltage undervoltage threshold 1			
10137	4799h	Analog inputs, wire break		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	Analog input [T2], wire break			
		Bit 1	Analog input [T1], wire break			
		Bit 0	-Internal-			

Parameter no.	Object-ID	Name		Unit	Data type	Note
10138	479Ah	Generator, watchdog 2		Bit field	unsigned16	
		Bit 15	Generator, load imbalance, limit 1			
		Bit 14	Generator, load imbalance, limit 2			
		Bit 13	Generator, voltage asymmetry			
		Bit 12	Generator, ground fault, limit 1			
		Bit 11	Generator, ground fault, limit 2			
		Bit 10	Generator, rotating field failure			
		Bit 9	Generator, load limit 1			*LogicsManager*
		Bit 8	Generator, load limit 2			*LogicsManager*
		Bit 7	Generator, inverse time overcurrent			Inverse time-o/c
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			
10139	---	Discrete inputs with alarm class		Bit field	unsigned16	The discrete input transmitting telegram bit is logically "1", if the discrete input is configured as an alarm input and has been selected.
		Bit 15	Discrete input [D1]			
		Bit 14	Discrete input [D2]			
		Bit 13	Discrete input [D3]			
		Bit 12	Discrete input [D4]			
		Bit 11	Discrete input [D5]			
		Bit 10	Discrete input [D6]			
		Bit 9	Discrete input [D7]			
		Bit 8	Discrete input [D8]			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			
10140	---	Flag of the *LogicsManager*		Bit field	unsigned16	
		Bit 15	Flag 1 is TRUE			
		Bit 14	Flag 2 is TRUE			
		Bit 13	Flag 3 is TRUE			
		Bit 12	Flag 4 is TRUE			
		Bit 11	Flag 5 is TRUE			
		Bit 10	Flag 6 is TRUE			
		Bit 9	Flag 7 is TRUE			
		Bit 8	Flag 8 is TRUE			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			

Parameter no.	Object-ID	Name		Unit	Data type	Note
10141	---	Analog inputs		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	Analog input [T2], limit 2			
		Bit 6	Analog input [T2], limit 1			
		Bit 5	Analog input [T1], limit 2			
		Bit 4	Analog input [T1], limit 1			
		Bit 3	Battery: overvoltage, limit 2			
		Bit 2	Battery: undervoltage, limit 2			
		Bit 1	Battery: overvoltage, limit 1			
		Bit 0	Battery: undervoltage, limit 1			
10146	47A2h	Internal flags of the *LogicsManager*		Bit field	unsigned16	
		Bit 15	Firing speed			
		Bit 14	Speed			
		Bit 13	Horn output			
		Bit 12	Stopping alarm			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	Daily time set point 1 exceeded			
		Bit 8	Daily time set point 2 exceeded			
		Bit 7	Actual weekday is in group of active weekdays			
		Bit 6	Actual day is active day			
		Bit 5	Actual hour is active hour			
		Bit 4	Actual minute is active minute			
		Bit 3	Actual second is active second			
		Bit 2	Enabled, if operation hours are odd			
		Bit 1	Enabled, if (operation hours)/10 are odd			
		Bit 0	Enabled, if (operation hours)/100 are odd			
10147	47A3h	Internal flags		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	-Internal-			
		Bit 0	-Internal-			

Parameter no.	Object-ID	Name		Unit	Data type	Note
10148	47A4h	Internal flags of the *LogicsManager*		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			
		Bit 6	Idle mode (suppresses undervoltage, underfrequency, underspeed)			
		Bit 5	-Internal-			
		Bit 4	GCB is closed			
		Bit 3	GCB is opened			
		Bit 2	MCB is closed			
		Bit 1	MCB is opened			
		Bit 0	-Internal-			
10149	47A5h	Alarms 2		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	-Internal-			
		Bit 1	Red stop lamp from ECU via J1939			Alarm bits will only be enabled if the respective monitoring function (J1939 amber/red alert) is triggered
		Bit 0	Amber warning lamp from ECU via J1939			

Parameter no.	Object-ID	Name		Unit	Data type	Note
10200	47D8h	System status		Bit field	unsigned16	
		Bit 15	Turning (gas engine)			
		Bit 14	Ignition ON (gas engine)			
		Bit 13	Dead-bus operation GCB			
		Bit 12	Dead-bus operation MCB			
		Bit 11	Starter/fuel solenoid (diesel engine) Gas valve (gas engine)			
		Bit 10	Start pause			
		Bit 9	Engine cool-down			
		Bit 8	Engine will be stopped			
		Bit 7	Preglow (diesel engine)			
		Bit 6	Starter protection			
		Bit 5	Emergency power op./ critical mode			
		Bit 4	Postrun auxiliary services (gas/diesel engine)			
		Bit 3	Mains settling			
		Bit 2	Prerun auxiliary services (gas/diesel engine)			
		Bit 1	Emergency power operation			
		Bit 0	Critical mode operation			

10201	---	System status		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	Operating mode STOP			
		Bit 7	Operating mode MANUAL			
		Bit 6	Operating mode AUTOMATIC			
		Bit 5 Bit 4	Engines is running			Double bit
		Bit 3 Bit 2	MCB is open			Double bit
		Bit 1 Bit 0	GCB is open			Double bit

Parameter no.	Object-ID	Name		Unit	Data type	Note
10203	---	System status		Bit field	unsigned16	
		Bit 15	-Internal-			
		Bit 14	-Internal-			
		Bit 13	-Internal-			
		Bit 12	-Internal-			
		Bit 11	-Internal-			
		Bit 10	-Internal-			
		Bit 9	-Internal-			
		Bit 8	-Internal-			
		Bit 7	-Internal-			
		Bit 6	-Internal-			
		Bit 5	-Internal-			
		Bit 4	-Internal-			
		Bit 3	-Internal-			
		Bit 2	Not in Automatic			
		Bit 1	-Internal-			
		Bit 0	Idle mode active			
10301	---	Mains: power factor cosphi		1/100, dim.less	unsigned16	These variables are necessary to ensure downward compatibility with LeoPC1 V2.1.xxx.
10302	---	Generator: real power P		1/10 kW	unsigned16	
10303	---	Generator: reactive power Q		1/10 kvar	unsigned16	
10304	---	Mains: real power P		1/10 kW	unsigned16	
10305	---	Mains: reactive power Q		1/10 kvar	unsigned16	
10306	---	Generator: power factor cosphi		cos1=100	unsigned16	
10307	---	External discrete inputs with alarm class		Bit filed	unsigned16	
		Bit 15	Discrete input [DEx16]			
		Bit 14	Discrete input [DEx15]			
		Bit 13	Discrete input [DEx14]			
		Bit 12	Discrete input [DEx13]			
		Bit 11	Discrete input [DEx12]			
		Bit 10	Discrete input [DEx11]			
		Bit 9	Discrete input [DEx10]			
		Bit 8	Discrete input [DEx09]			
		Bit 7	Discrete input [DEx08]			
		Bit 6	Discrete input [DEx07]			
		Bit 5	Discrete input [DEx06]			
		Bit 4	Discrete input [DEx05]			
		Bit 3	Discrete input [DEx04]			
		Bit 2	Discrete input [DEx03]			
		Bit 1	Discrete input [DEx02]			
		Bit 0	Discrete input [DEx01]			
15600	5CF0h	Generator values		---	unsigned64	Data protocol
15601	5CF1h	Visualization data for easYlite-100		---	unsigned64	Data protocol

J1939 Measuring values

J1939 Standard Measuring Values

Parameter No.	Object-ID	Name			Unit	Data type	Value with defective sensor	Value with missing sensor value
		SPN	PGN	Description in J1939 protocol				
15200	**5B60h**	190	61444	Engine speed	0.1 rpm	unsigned32	214748364.6rpm	214748364.7rpm
15201	**5B61h**	247	65253	Total engine hours	1 h	unsigned32	2147483646h	2147483647h
15202	**5B62h**	110	65262	Engine coolant temperature	°C	signed16	32766°C	32767°C
15203	**5B63h**	174	65262	Fuel temperature	1°C	signed16	32766°C	32767°C
15204	**5B64h**	175	65262	Engine oil temperature	0.01°C	signed32	21474836.46°C	21474836.47°C
15205	**5B65h**	100	65263	Engine oil pressure	1kPa	unsigned16	65534kPa	65535kPa
15206	**5B66h**	111	65263	Coolant level	0.1%	unsigned16	6553,4%	6553.5%
15207	**5B67h**	91	61443	Throttle position	0.1%	unsigned16	6553.4%	6553.5%
15208	**5B68h**	92	61443	Load at current speed	1%	unsigned16	65534%	65535%
15209	**5B69h**	513	61444	Actual engine torque	1%	signed16	32766%	32767%
15210	**5B6Ah**	98	65263	Engine oil level	0.1%	unsigned16	6553.4%	6553.5%
15211	**5B6Bh**	183	65266	Fuel rate	0.01 l/h	unsigned32	21474836.46 L/h	21474836.47 L/h
15212	**5B6Ch**	108	65269	Barometric pressure	0.1kPa	unsigned16	65534kPa	65535kPa
15213	**5B6Dh**	172	65269	Air inlet temperature	1°C	signed16	32766°C	32767°C
15214	**5B6Eh**	102	65270	Boost pressure	1kPa	unsigned16	65534kPa	65535kPa
15215	**5B6Fh**	105	65270	Intake manifold temp.	1°C	signed16	32766°C	32767°C
15216	**5B70h**	173	65270	Exhaust gas temperature	0.001°C	signed32	21474836.46°C	21474836.47°C

J1939 Messages of DM1 Advise

Parameter No.	Object-ID	Name		Unit	Data type
15395	5C23h	DM1 Lamp State		Digit	unsigned16
		Protect Lamp State			
		Bit 0	Off		
		Bit 1	On		
		Bit 2	Missing		
		Bit 3	Missing		
		Amber Warning Lamp State			
		Bit 4	Off		
		Bit 5	On		
		Bit 6	Missing		
		Bit 7	Missing		
		Red Stop Lamp State			
		Bit 8	Off		
		Bit 9	On		
		Bit 10	Missing		
		Bit 11	Missing		
		Malfunction Indicator Lamp State			
		Bit 12	Off		
		Bit 13	On		
		Bit 14	Missing		
		Bit 15	Missing		

Parameter No.	Object-ID	Name		Unit	Data type
15400	5C28h	1.	SPN-Number	Digit	unsigned32
15401	5C29h	1.	FMI	Digit	unsigned8
15402	5C2Ah	1.	OC	Digit	unsigned8
15403	5C2Bh	2.	SPN-Number	Digit	unsigned32
15404	5C2Ch	2.	FMI	Digit	Unsigned8
15405	5C2Dh	2.	OC	Digit	unsigned8
15406	5C2Eh	3.	SPN-Number	Digit	unsigned32
15407	5C2Fh	3.	FMI	Digit	unsigned8
15408	5C30h	3.	OC	Digit	unsigned8
15409	5C31h	4.	SPN-Number	Digit	unsigned32
15410	5C32h	4.	FMI	Digit	unsigned8
15411	5C33h	4.	OC	Digit	unsigned8
15412	5C34h	5.	SPN-Number	Digit	unsigned32
15413	5C35h	5.	FMI	Digit	unsigned8
15414	5C36h	5.	OC	Digit	unsigned8
15415	5C37h	6.	SPN-Number	Digit	unsigned32
15416	5C38h	6.	FMI	Digit	unsigned8
15418	5C3Ah	6.	OC	Digit	unsigned8
15419	5C3Bh	7.	SPN-Number	Digit	unsigned32
15420	5C3Ch	7.	FMI	Digit	unsigned8
15421	5C3Dh	7.	OC	Digit	unsigned8
15422	5C3Eh	8.	SPN-Number	Digit	unsigned32
15423	5C3Fh	8.	FMI	Digit	unsigned8
15424	5C40h	8.	OC	Digit	unsigned8
15425	5C41h	9.	SPN-Number	Digit	unsigned32
15426	5C42h	9.	FMI	Digit	unsigned8
15427	5C43h	9.	OC	Digit	unsigned8
15428	5C44h	10.	SPN-Number	Digit	unsigned32
15429	5C45h	10.	FMI	Digit	unsigned8
15430	5C46h	10.	OC	Digit	unsigned8

J1939 Messages of DM2 Advise

Parameter No.	Object-ID	Name		Unit	Data type
15445	5C55h	DM2 Lamp Status		Digit	unsigned16
		Protect Lamp Status			
		Bit 0	Off		
		Bit 1	On		
		Bit 2	Missing		
		Bit 3	Missing		
		Amber Warning Lamp Status			
		Bit 4	Off		
		Bit 5	On		
		Bit 6	Missing		
		Bit 7	Missing		
		Red Stop Lamp Status			
		Bit 8	Off		
		Bit 9	On		
		Bit 10	Missing		
		Bit 11	Missing		
		Malfunction Indicator Lamp Status			
		Bit 12	Off		
		Bit 13	On		
		Bit 14	Missing		
		Bit 15	Missing		

Parameter No.	Object-ID	Name		Unit	Data type
15450	5C5Ah	1.	SPN-Number	Digit	unsigned32
15451	5C5Bh	1.	FMI	Digit	unsigned8
15452	5C5Ch	1.	OC	Digit	unsigned8
15453	5C5Dh	2.	SPN-Number	Digit	unsigned32
15454	5C5Eh	2.	FMI	Digit	unsigned8
15455	5C5Fh	2.	OC	Digit	unsigned8
15456	5C60h	3.	SPN-Number	Digit	unsigned32
15457	5C61h	3.	FMI	Digit	unsigned8
15458	5C62h	3.	OC	Digit	unsigned8
15459	5C63h	4.	SPN-Number	Digit	unsigned32
15460	5C64h	4.	FMI	Digit	unsigned8
15461	5C65h	4.	OC	Digit	unsigned8
15462	5C66h	5.	SPN-Number	Digit	unsigned32
15463	5C67h	5.	FMI	Digit	unsigned8
15464	5C68h	5.	OC	Digit	unsigned8
15465	5C69h	6.	SPN-Number	Digit	unsigned32
15466	5C6Ah	6.	FMI	Digit	unsigned8
15467	5C6Bh	6.	OC	Digit	unsigned8
15468	5C6Ch	7.	SPN-Number	Digit	unsigned32
15469	5C6Dh	7.	FMI	Digit	unsigned8
15470	5C6Eh	7.	OC	Digit	unsigned8
15471	5C6Fh	8.	SPN-Number	Digit	unsigned32
15472	5C70h	8.	FMI	Digit	unsigned8
15473	5C71h	8.	OC	Digit	unsigned8
15474	5C72h	9.	SPN-Number	Digit	unsigned32
15475	5C73h	9.	FMI	Digit	unsigned8
15476	5C74h	9.	OC	Digit	unsigned8
15477	5C75h	10.	SPN-Number	Digit	unsigned32
15478	5C76h	10.	FMI	Digit	unsigned8
15479	5C77h	10.	OC	Digit	unsigned8

J1939 Appendix for S6

Parameter No.	Object-ID	Name				Unit	Data type
		SPN	PGN	Description in J1939 protocol			
15305	5BC9h		65409	DLN2-Proprietary Low Engine Oil Level		Digit	unsigned16
				Bit 0	Not Low Engine Oil Level		
				Bit 1	Low Engine Oil Level		
				Bit 2	Sensor defect		
				Bit 3	Missing		
				DLN2-Proprietary High Engine Oil Level			
				Bit 4	Not High Engine Oil Level		
				Bit 5	High Engine Oil Level		
				Bit 6	Sensor defect		
				Bit 7	Missing		
				DLN2-Proprietary Low Engine Oil Pressure			
				Bit 8	Not Low Engine Oil Pressure		
				Bit 9	Low Engine Oil Pressure		
				Bit 10	Sensor defect		
				Bit 11	Missing		
				DLN2-Proprietary High Engine Coolant Temperature			
				Bit 12	Not High Engine Coolant Temperature		
				Bit 13	High Engine Coolant Temperature		
				Bit 14	Sensor defect		
				Bit 15	Missing		

Compare also the documentation of S6

J1939 Appendix for EMR

Parameter No.	Object-ID	Name				Unit	Data type
		SPN	PGN	Description in J1939 protocol			
15304	5BC8h		65301	Motor stop information		Digit	unsigned16
				0	No Stop		
				1	Engine safety		
				2	CAN message Engine Stop Request		
				3	low oil pressure		
				4	low oil level		
				5	high coolant temperature		
				6	low coolant level		
				7	intake manifold temp		
				8	reserved (Stop via SAE-J1587)		
				9	reserved (Stop via VP2)"\		
				FEFFh	Sensor defect		
				FFFFh	Missing		

Compare also the documentation of EMR

J1939 Appendix for MTU ADEC

Parameter No.	Object-ID	Name				Unit	Data type
		SPN	PGN	Description in J1939 protocol			
15109	5B05h	-	65284	ADEC Failure codes		Digit	unsigned 16
				0 to 64768	Failure code enumeration		
				FEFFh	Sensor defect		
				FFFFh	Missing		

Compare also the documentation of MTU ADEC.

Parameter No.	Object-ID	Name		Unit	Data type
16377	5FF9h		Latched state of external DI alarm bits	Bit field	unsigned16
		Bit 15	Discrete input [Dex16]		
		Bit 14	Discrete input [Dex15]		
		Bit 13	Discrete input [Dex14]		
		Bit 12	Discrete input [Dex13]		
		Bit 11	Discrete input [Dex12]		
		Bit 10	Discrete input [Dex11]		
		Bit 9	Discrete input [Dex10]		
		Bit 8	Discrete input [Dex09]		
		Bit 7	Discrete input [Dex08]		
		Bit 6	Discrete input [Dex07]		
		Bit 5	Discrete input [Dex06]		
		Bit 4	Discrete input [Dex05]		
		Bit 3	Discrete input [Dex04]		
		Bit 2	Discrete input [Dex03]		
		Bit 1	Discrete input [Dex02]		
		Bit 0	Discrete input [Dex01]		

Appendix C.
Supported Remote Control Messages for ECUs

NOTE

Because of the various different ECU types with different configuration options, this section is only an overview and not binding. 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.

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).

Remote control parameter	Woodward EGS02	Scania S6	Deutz EMR	Volvo EMS2	MTU ADEC	SISU ECU		Standard	Comment
						EEM2	EEM3		
Engine Start	No	Yes	No	Yes	Yes	No	Yes	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.
Engine Stop	Yes	Yes	No	Yes	Yes	No	Yes	No	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.
Droop mode	Yes	Yes	No	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 parameter "ECU set droop mode" is configured to "On".
Idle Mode	No	Yes	No	Yes	No	No	No	No	This J1939 bit information is set, if "Idle" mode is requested via the corresponding *LogicsManager* (equation has to be TRUE) and "Idle" mode is active. The bit will be reset, if the corresponding *LogicsManager* equation becomes FALSE, and "Idle" mode is no longer active.
50/60 Hz switch	Yes	Yes	No	Yes	Yes	No	No	No	If the "Rated system frequency" parameter within the easYgen is set to "50Hz", the J1939 information for 50Hz mode is sent to the ECU. If the "Rated system frequency" parameter within the easYgen is set to "60Hz", the J1939 information for 60Hz mode is sent to the ECU.
Speed offset	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	**Important:** This message is only sent, if parameter "Frequency offset ECU" is configured to "On".
Preglow	No	No	No	Yes	No	No	No	No	This J1939 bit information is set, if the easYgen is in "Preglow" mode. The bit will be reset, if the "Preglow" phase has been expired or aborted.
Reset ADEC Alarm	No	No	No	No	Yes	No	No	No	Any alarm of an MTU ADEC ECU will be acknowledged if the "External Acknowledge" *LogicsManager* becomes TRUE or the "Reset ADEC Alarms" softkey is pressed in the J1939 display screen.

Appendix D.
Application Examples

Remote Control

The easYgen-1000 controller may be configured to perform start/stop/acknowledgement functions remotely through the CAN bus. The required procedure is detailed in the following steps.

NOTE

Refer to the operation manual 37392 for a detailed description of the navigation through the various display screens. A detailed description of the individual parameters may be found in the configuration manual 37391.

Be sure to enter the password for code level 2 or higher to be able to access the required configuration screens.

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* Functions

Open the main menu by pressing the softkey and navigate to "Configure application" by using the softkey. Open the "Configure application" menu by using the softkey. Navigate to "Start req. in AUTO" by using the softkey and enter the "Start req. in AUTO" *LogicsManager* screen by pressing the softkey.

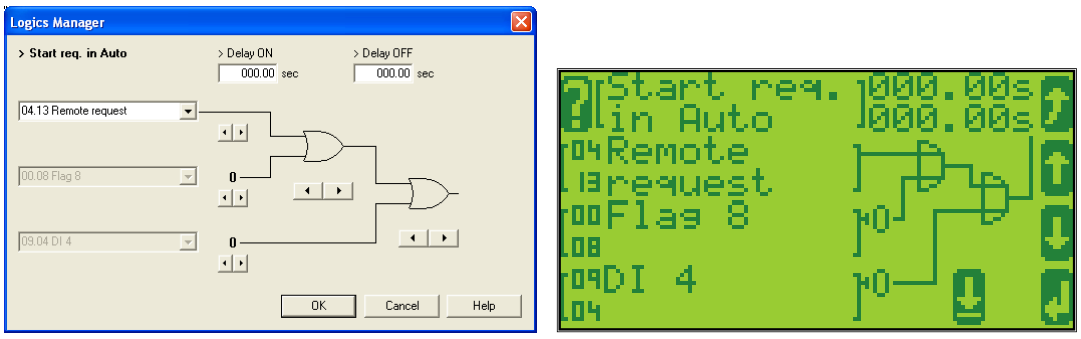


Figure 6-1: Display screen - Start req in AUTO

Configure the respective values for the "Start req. in AUTO" *LogicsManager* function using the and as well as the softkey and Confirm the change by pressing the softkey:

With this setting, the "Start req. in AUTO" *LogicsManager* output becomes TRUE as soon as the remote request signal is enabled.

Press to return to the "Configure application" menu and navigate to "Stop req. in AUTO" by using the softkey and press to enter the "Stop req. in AUTO" *LogicsManager* screen.

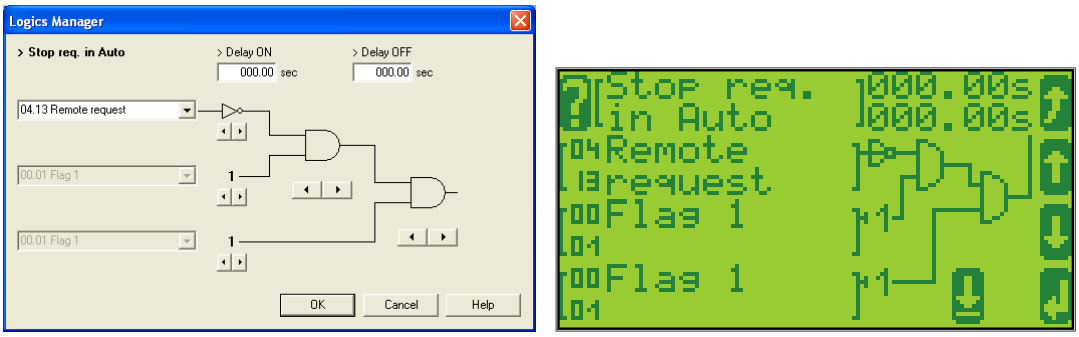


Figure 6-2: Display screen - Stop req in AUTO

Configure the respective values for the "Stop req. in AUTO" *LogicsManager* function using the and as well as the softkey and Confirm the change by pressing the softkey:

With this setting, the "Stop req. in AUTO" *LogicsManager* output becomes TRUE as soon as the remote request signal is disabled.

Press twice to return to the main menu and navigate to "Configure monitoring" screen by using the softkey. Open the "Configure monitoring" menu by using the softkey. Navigate to "External acknowledge" by using the softkey and enter the "External acknowledge" *LogicsManager* screen by pressing the softkey.

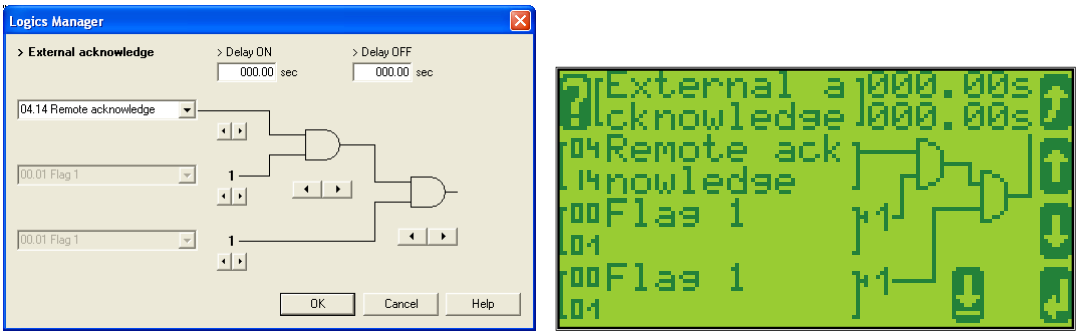


Figure 6-3: Display screen - Ext. acknowledge

Configure the respective values for the "External acknowledge" *LogicsManager* function using the and as well as the softkey and Confirm the change by pressing the softkey:

With this setting, the "External acknowledge" *LogicsManager* output becomes TRUE as soon as the remote acknowledge signal is enabled.

NOTE

The *LogicsManager* commands 2 and 3 may be used to configure additional conditions like discrete inputs, which must be energized to be able to issue the remote command.

Remote Control Telegram

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 6-4 for the priority of start and stop signals).

Refer to the Remote Control Telegram section on page 53 for detailed information about the telegram structure and the control bits.

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

Ext. Acknowledge: The command variable "04.14 Remote acknowledge" is the reflection of the control bit (bit 4). The easYgen deactivates the horn with the first change from "0" to "1" of the logical output "External acknowledge", and acknowledges inactive alarm messages with the second change from "0" to "1".

Figure 6-4 shows the reaction of the command variable on the various status changes of the bits:

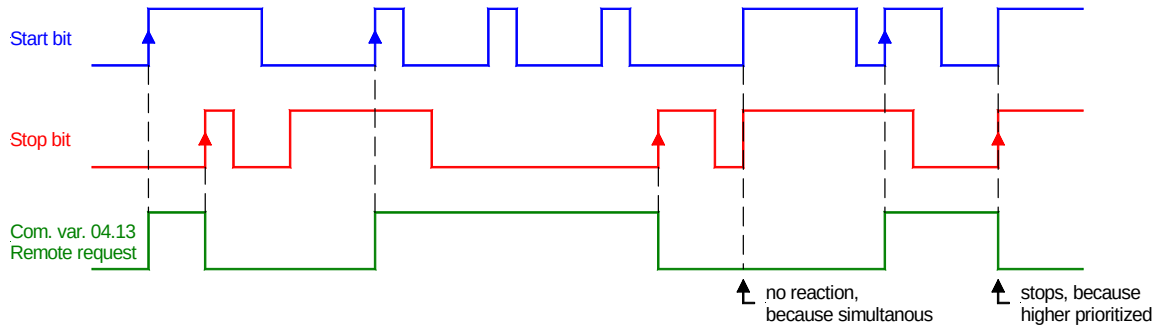


Figure 6-4: Remote control - start/stop priority

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 via CAN

It is possible to perform a remote start/stop/acknowledgement via a default SDO communication channel.

Remote Start/Stop/Acknowledgement

Configuration of CAN Interface

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

Open the main menu by pressing the softkey and navigate to "Set up Comm interfaces" by using the softkey. Open the "Set up Comm interfaces" menu by using the softkey and navigate to "Set up CAN interfaces" by using the softkey. Open the "Set up CAN interfaces" menu by using the softkey and navigate to "CAN Open interface" by using the softkey. Open the "CAN Open interface" menu by using the softkey, navigate to "CAN-Open Master" by using the softkey and enter the "CAN-Open Master" screen by pressing the softkey.

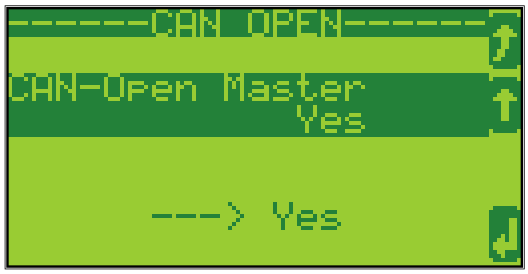


Figure 6-5: Display screen - configure CAN interface

Select "Yes" by using the softkey and confirm your selection by pressing the softkey.

General Information

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

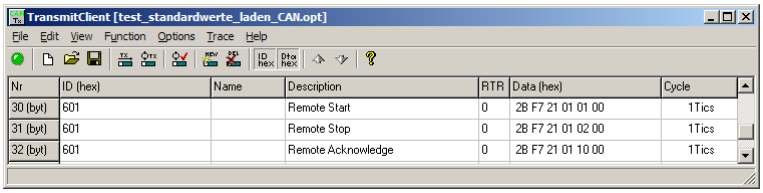


Figure 6-6: CANopen request data for Node ID 1

Node ID (not standard value)

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

Press ↑ until you return to the start screen.

Open the main menu by pressing the ↑ softkey and navigate to "Set up Comm interfaces" by using the ↓ softkey. Open the "Set up Comm interfaces" menu by using the ↵ softkey and navigate to "Device number" by using the ↓ softkey and enter the "Device number" screen by pressing the ↵ softkey.

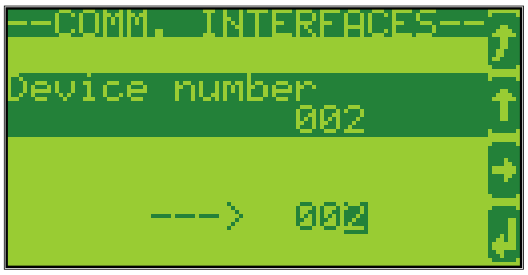


Figure 6-7: Display screen - configure device number

Configure "002" by using the → and ↑ softkeys and confirm your selection by pressing the ↵ softkey.

With this setting, the Node ID of the CAN interface 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 6-8 shows exemplary request data for the device on the CANopen bus.

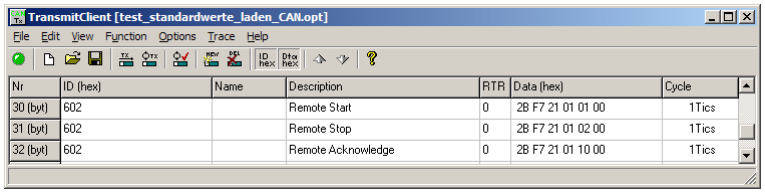


Figure 6-8: CANopen request data for Node ID 2

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) or 7F (hex) is used in the following example.

Press ↑ until you return to the start screen.

Open the main menu by pressing the ↓ softkey and navigate to "Set up Comm interfaces" by using the ↓ softkey. Open the "Set up Comm interfaces" menu by using the ↵ softkey and navigate to "Set up CAN interfaces" by using the ↓ softkey. Open the "Set up CAN interfaces" menu by using the ↵ softkey and navigate to "CAN Open interface" by using the ↓ softkey. Open the "CAN Open interfaces" menu by using the ↵ softkey and navigate to "Additional Server SDOs" by using the ↓ softkey. Enter the "Additional S-SDO" screen by pressing the ↵ softkey.

Navigate to "2nd Client->Server COB-ID (rx)" by using the ↓ softkey and press the ↵ softkey to edit this parameter. Configure "0000067F" by using the → and ↑ softkeys and confirm your entry by pressing the ↵ softkey.

Navigate to "2nd Server->Client COB-ID (tx)" by using the ↓ softkey and press the ↵ softkey to edit this parameter. Configure "000005FF" by using the → and ↑ softkeys and confirm your entry by pressing the ↵ softkey.

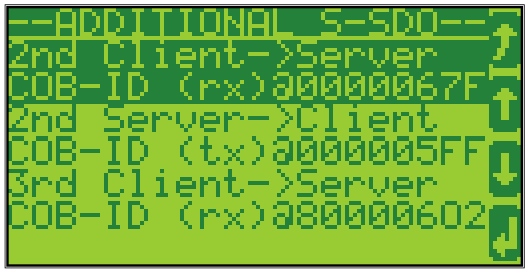


Figure 6-9: Display screen - configure Server SDOs

NOTE

Be sure to remove the leading 8 from the COB-IDs to enable them. For example, change the standard value of "2nd Client-Server COB-ID (rx)", which is "80000601", to "00000067F".

In this example, an additional SDO communication channel is configured to 127 (decimal) or 7F (hex).

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) + Node ID internally to perform the desired control, the reply from the easYgen is sent on CAN ID 580 (hex) + Node ID.
Receive CAN ID 67F (hex) (600 (hex) + 7F (hex))
Transmit CAN ID 5FF (hex) (580 (hex) + 7F (hex))

The same is valid for the additional SDO communication channels 3, 4, and 5.
Figure 6-10 shows exemplary request data for the device on the CANopen bus.

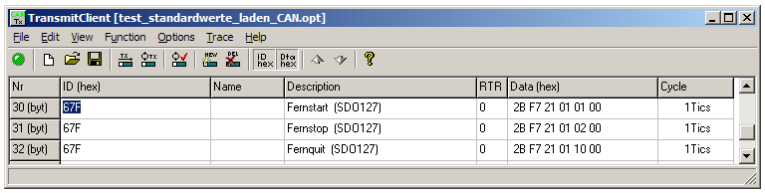


Figure 6-10: CANopen request data for additional Server SDO

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.

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 Telegram on page 89.

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	0 to 65535	UNSIGNED 16

Modbus address = 40000 + (Par. ID +1) = 504
Modbus length = 1 (UNSIGNED 16)

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

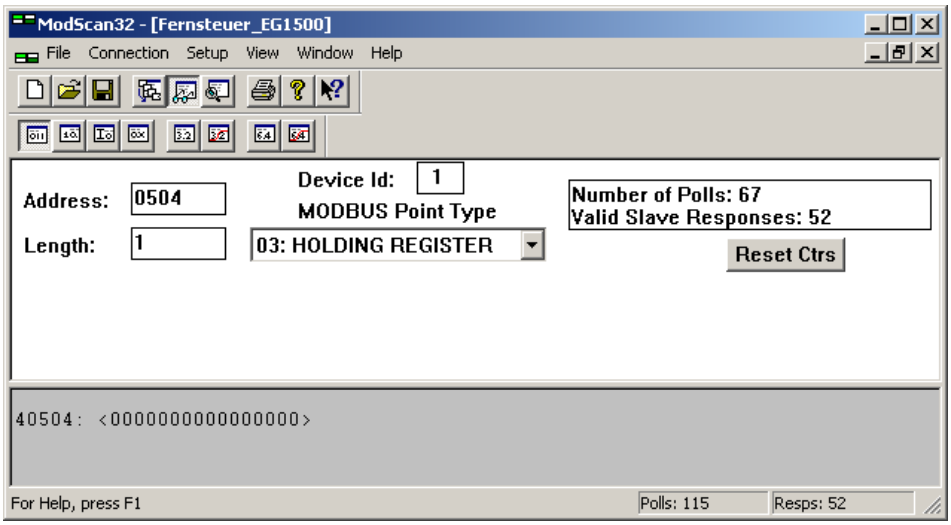


Figure 6-11: Modbus - remote control parameter 503

By double-clicking the address, a Write Register command may be issued. Figure 6-12 shows how bit 0 is set using the ModScan32 Software.

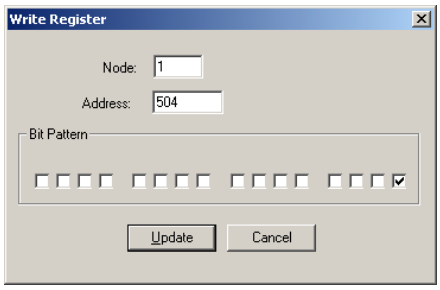


Figure 6-12: Modbus - write register

NOTE

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

Sending a Data Protocol via TPDO

Cyclically Sending of Data

This is a configuration example for sending an object with the index 3190 (data protocol 4003) 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	3190	(display value, the object with the index 3190)
2. Mapped Object	0	(will not be used)
3. Mapped Object	0	(will not be used)
4. Mapped Object	0	(will not be used)

Sending of Data on Request

The data to be sent (Mapped Objects) may be provided on request by configuring the Sync Message and the Transmission Type 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 must be configured to "1" and the CANopen Master function must be configured to "Off".

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

The recorded data shows that the data of the Mapped Object (in this example Mux 5) is sent (refer to Figure 6-14) after sending the Sync Message twice (refer to Figure 6-13).

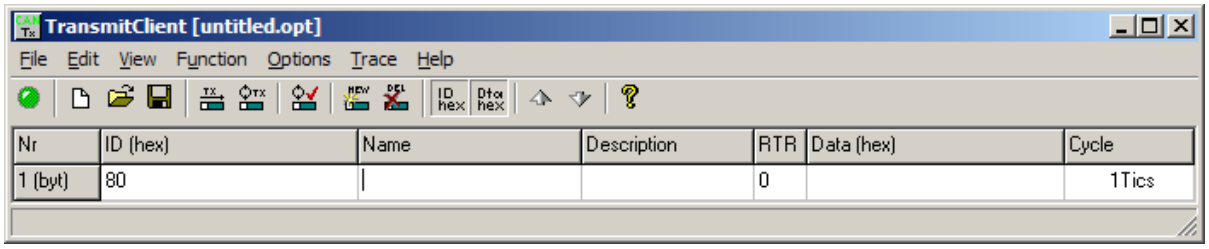


Figure 6-13: Cyclical sending of data - Sync Message request

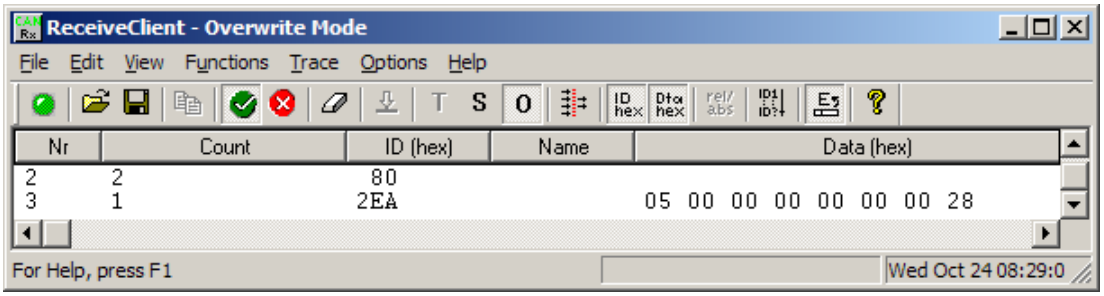


Figure 6-14: Cyclical sending of data - reply

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
stgt-info@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).

2008/07/Stuttgart