Heinzmann GmbH & Co. KG
Engine & Turbine Management

Am Haselbach 1
D-79677 Schönau
Germany

Phone +49 7673 8208 - 0
Fax +49 7673 8208 - 188
E-mail info@heinzmann.de
www.heinzmann.com

V.A.T. No.: DE145551926

HEINZMANN®

Engine & Turbine Management

HEINZMANN CAN bus

Basic information

<table>
<tr>
<td>



</td>
<td>

The appropriate manuals must be thoroughly studied before installation, initial start-up and maintenance.

All instructions pertaining to the system and safety must be followed in full. Non-observance of the instructions may lead to injury to persons and/or material damage.

HEINZMANN shall not be held liable for any damage caused through non-observance of instructions.

Independent tests and inspections are of particular importance for all applications in which a malfunction could result in injury to persons or material damage.

All examples and data, as well as all other information in this manual are there solely for the purpose of instruction and they may not be used for special application without the operator running independent tests and inspections beforehand.

HEINZMANN does not guarantee, neither expressly nor tacitly, that the examples, data or other information in this manual is free from error, complies with industrial standards or fulfils the requirements of any special application.

</td>
</tr>
<tr>
<td>


</td>
<td>

To avoid any injury to persons and damage to systems, the following monitoring and protective systems must be provided:

– Overspeed protection independent of the rpm controller

HEINZMANN shall not be held liable for any damage caused through missing or insufficiently rated overspeed protection.

– thermal overload protection

The following must also be provided for alternator systems:

– Overcurrent protection

– Protection against faulty synchronisation for excessively-large frequency, voltage or phase difference

– Directional contactor

The reasons for overspeeding may be:

– Failure of positioning device, control unit or its auxiliary devices

– Linkage sluggishness and jamming

</td>
</tr>
<tr>
<td>


</td>
<td>

The following must be observed before an installation:

– Always disconnect the electrical mains supply before any interventions to the system.

– Only use cable screening and mains supply connections that correspond with the *European Union EMC Directive*

– Check the function of all installed protection and monitoring systems

</td>
</tr>
</table>

	**Please observe the following for electronically controlled injection (MVC):** – For **common rail** systems each injector line must be equipped with a separate mechanical flow-rate limiter – For **unit pump** (PLD) and **pump-injector unit** (PDE) systems, the fuel enable is first made possible by the solenoid valve’s control plunger motion. This means that in the event of the control plunger sticking, the fuel supply to the injection valve is stopped.
	As soon as the positioning device receives power, it can actuate the controller output shaft automatically at any given time. The range of the controller shaft or control linkage must therefore be secured against unauthorised access.
	HEINZMANN expressly rejects any implied guarantee pertaining to any marketability or suitability for a special purpose, including in the event that **HEINZMANN** was notified of such a special purpose or the manual contains a reference to such a special purpose.
	HEINZMANN shall not be held liable for any indirect and direct damage nor for any incidental and consequential damage that results from application of any of the examples, data or miscellaneous information as given in this manual.
	HEINZMANN shall not provide any guarantee for the design and planning of the overall technical system. This is a matter of the operator its planners and its specialist engineers. They are also responsible for checking whether the performances of our devices match the intended purpose. The operator is also responsible for a correct initial start-up of the overall system.

Version information

Version	Description of changes	Date	Edited by
02-13	Created	21/06/2013	ScB
11-13	DC12 periphery module introduced	18/11/2013	ScB
05-14	Chap. 5.2.10 Detection of both variants for the allocation of the digital inputs of the periphery module in the master	13/05/2014	ScB
	Chap. 5.2.11 Assignment of frequency inputs added		
	General renaming of digital input/output as binary input/output		
	MVC 01-3G introduced		
	Unit terminating resistors moved to the end of the brochure		

Table of Contents

1 CAN bus

The CAN bus is a 2-wire bus system. All participants are connected in parallel to this bus. Only stub lines featuring short line lengths are permitted. A terminating resistor of 120 Ω must be provided at both ends of the bus in order to prevent reflections. The cable lengths in the following table are stated in metres (m).

Baud rate kBaud	Max. cable length	Max. cable length with galvanic isolation	Max. cable length per stub line	Max. cable length of all stub lines
125	390	350	16	80
250	190	150	8	40
500	90	50	4	20
1000	40	-	2	10

Tab. 1: Cable lengths

These line lengths represent the theoretical maximum. The lengths themselves should serve only as a guide. However, it is not guaranteed that the system will function perfectly in the event that the maximum possible lengths are used.

1.1 CAN cables

A particular CAN cable with a characteristic impedance of 108 to 132 Ω is required. Correct function cannot be ensured if other cables are used.

In addition, all participants must be connected to the reference potential CAN ground. The only exception to this rule is if all participants are fed by the same supply and the network spread is low.

The following cables are recommended by HEINZMANN:

1. HELU CAN bus 1×2×0.22 shielded
   Item no. 81286
2. BELDEN CAN
   Item no. 9841
3. Huber+Suhner 2×0.5+0.5 mm²
   Heinzmann order no. 010-02-426-00 (marine approved)

1.2 Galvanic isolation

Galvanic isolation is always required in the event that the participants are not fed by the same supply. This is particularly applicable if the communication is meant to take place between a control unit and a PC/laptop.

The use of galvanic isolation limits the maximum possible length of all lines and stub lines (↑*Tab. 1:* Cable lengths).

1.3 Stub lines

In the CAN bus, all participants need to be connected in parallel. This means that stub lines are permitted only in exceptional cases, as they cause reflections. The length of an individual stub line and the overall length of all stub lines is limited ↑ *Tab. 1:* Cable lengths.

1.4 Terminating resistors

To prevent reflections, each end of the CAN bus must be supplied with a terminating resistor of 120 Ω. This is also a requirement for very short line lengths. The terminating resistor must not be attached to stub lines.

Split termination can also be used for the purpose of improved electromagnetic compatibility. This involves 2×60 Ω terminating resistors with a capacitor featuring 10 nF to ground connected between them.

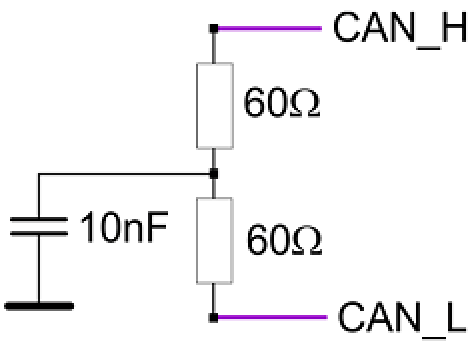


Fig. 1: Terminating resistor 2×60 Ω

An ohmmeter can be used to check the termination and the physical connection of the CAN network at any point in the network. All bus participants must be isolated from the mains prior to the measurement, otherwise the measurement may be distorted.

Measurement between	Measured value	Meaning
CAN L and CAN H	approx. 60 Ω	OK, 2 terminating resistors in bus
GND and CAN L	Infinite	OK
GND and CAN H	Infinite	OK
GND and CAN L	0	Short circuit between GND and CAN L
GND and CAN H	0	Short circuit between GND and CAN H
CAN L and CAN H	approx. 120 Ω	Only 1 terminating resistor in bus
CAN L and CAN H	< 50 Ω	More than 2 terminating resistors in bus

Tab. 2: Verification of terminating resistor

It is often the case that terminating resistors cannot be positioned correctly or are not positioned at all. However, it is forbidden to perform work using the CAN if the physical CAN line is not structured correctly. This is particularly true if important data such as the speed, measured power or values for gas releases are to be transferred. If a correct physical CAN connection is not in place, it is also impossible to ensure a correct logical connection.

For each HEINZMANN control unit, ↑ 9 *Terminating resistors* states whether or not terminating resistors are fitted as standard and where these are to be attached or which are to be removed.

1.5 Baud rate

Preference should be given to a baud rate that is as low as possible. However, the selection of the baud rate depends on the bus load and the line length. The bus load results from the number of participants on the bus and the number of CAN telegrams. A bus load of 60% should not be exceeded where possible. In exceptional cases, contact HEINZMANN in order to determine the correct baud rate.

The parameterisation for the selected baud rate can be performed in the HEINZMANN unit – you must pay attention to the differences relating to the CAN controller in use.

Only the 4 values of 125, 250, 500 and 1000 kBaud are valid baud rates in 416 *CanBaudrate*; 250 kBaud is used for every other entry. The segment settings for the CAN controller relating to these 4 values are permanently stored in the control unit. These segment settings are calculated for a scan time of 75%. This enables the maximum line lengths in ↑*Tab. 1:* Cable lengths to be achieved.

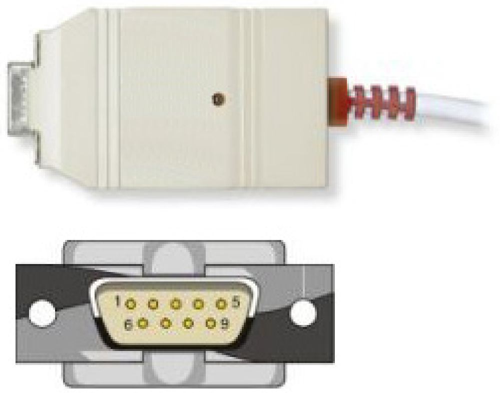
i

In principle, a baud rate of 1000 kBaud is only permitted for test installations.

1.6 PC-CAN adapter

A CAN adapter is required for communication between the control unit and the PC/laptop, as PCs in themselves do not contain a CAN controller. This adapter must be galvanically isolated, as the reference potential of the PC/laptop is usually not the same as that of the control units.

The PC-CAN adapter with galvanic isolation and a USB connection (Heinzmann order number 010-02-563-01) is stipulated for the use of DcDesk 2000/CAN.



Pin	Assignment
1	n.c.
2	CAN-L
3	CAN-GND / n.c.
4	n.c.
5	n.c.
6	CAN-GND / n.c.
7	CAN-H
8	n.c.
9	n.c.

Fig. 2: PC-CAN adapter, pin assignment

1.6.1 Position of the PC-CAN adapter in the CAN bus

A 9-pin (female) plug must be provided for connecting the PC-CAN adapter. This plug must always be connected to 1 end of the bus and issued with 120 Ω. This is the only way to ensure the option of connecting or not connecting the PC-CAN adapter and therefore DcDesk 2000/CAN, as is standard in a service case. It is not permitted to route a stub line to this plug.

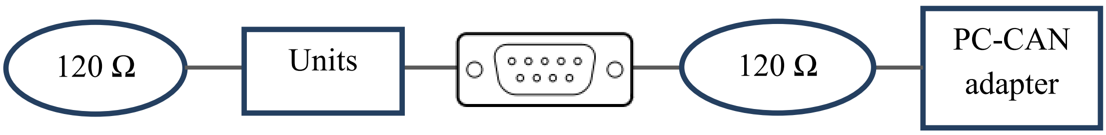


Fig. 3: PC-CAN adapter, positioned in CAN bus

CAN is not a plug and play bus system!

If the PC-CAN adapter is connected to a system that is in operation, this generally leads to communication errors. It is therefore recommended that users pause the system (e.g. by stopping the engine) if they need to connect the PC-CAN adapter.

In the event that the PC-CAN adapter has to be connected to a system that is in operation, the following sequence causes the fewest errors:

1. Connect the PC-CAN adapter to a USB connection on the PC/laptop, but do not connect to the CAN bus yet
2. Start DcDesk 2000/CAN (display in header: "Searching for baud rate")
3. Connect the PC-CAN adapter to the CAN bus

2 General

The HZM-CAN protocol is based on the CAN specification 2.0B with a 29-bit identifier. The protocol must be released separately when creating unit software, as this is the only way for a unit of this kind to be integrated into the CAN bus. This means that the HZM-CAN is not automatically available in each item of unit software.

i

The parameterisation of the HZM-CAN network is supported by DcDesk 2000 in the special window 'Control unit/Settings/HZM-CAN configuration'. Following configuration, the parameters must be saved and a reset must be performed.

The sender and the recipient can be any digital HEINZMANN units or an external unit into which a customer has integrated the HZM-CAN protocol.

2.1 Additional information

This publication describes functions and setting parameters for the implementation of the HZM-CAN bus in the speed governor, additional modules and periphery modules.

The publication

HEINZMANN-CAN customer module, publication no. DG 05 007-e

features a detailed description of the implementation and use of a customer module.

Please see

Control device for conventional injection with actuators, publication no. DG 07 001-e

with regard to general configuration of speed governors for controlling actuators and see

DARDANOS Basic Information, publication no. MV 09 001-e

with regard to speed governors for controlling solenoid valves.

Additional modules are described in the documents

ARIADNE, publication no. DG 06 001-e

ICS_PHLOX_II, publication no. DG 13 004-e

All of these manuals can be downloaded from www.heinzmann.com following registration.

2.2 Example for an HZM-CAN bus system

In a theoretical generator system, the following HEINZMANN units should be connected to one another and communicate with one another:

Generator 1: Diesel V-engine (HELENOS, DC 2 PE, THESEUS)

Generator 2: Dual-fuel engine (PRIAMOS, MVC01 PE, THESEUS, ARIADNE)

Generator 3: Gas V-engine (PANDAROS, 2 * DC 10 PE, THESEUS, ARIADNE)

This section only clarifies the physical structure. The associated parameterisation is described in ↑8 *Parameterisation* – this comes after the following chapters that describe the structure of the HZM-CAN protocol in more detail.

2.2.1 Physical connection

The individual units are connected via the CAN line within the switch cabinets as follows:

Switch cabinet 1: DC 2 PE – HELENOS – THESEUS 1

Switch cabinet 2: MVC01 PE – PRIAMOS – ARIADNE 1 – THESEUS 2

Switch cabinet 3: DC 10 PE 1 – DC 10 PE 2 – PANDAROS – ARIADNE 2 – THESEUS 3

You then reconnect the switch cabinets to one another.

Switch cabinet 1 – switch cabinet 2 – switch cabinet 3

The result of this is that the 120 Ω terminating resistors must be connected to DC 2 PE and THESEUS 3 and nowhere else, ↑*1.4 Terminating resistors*.

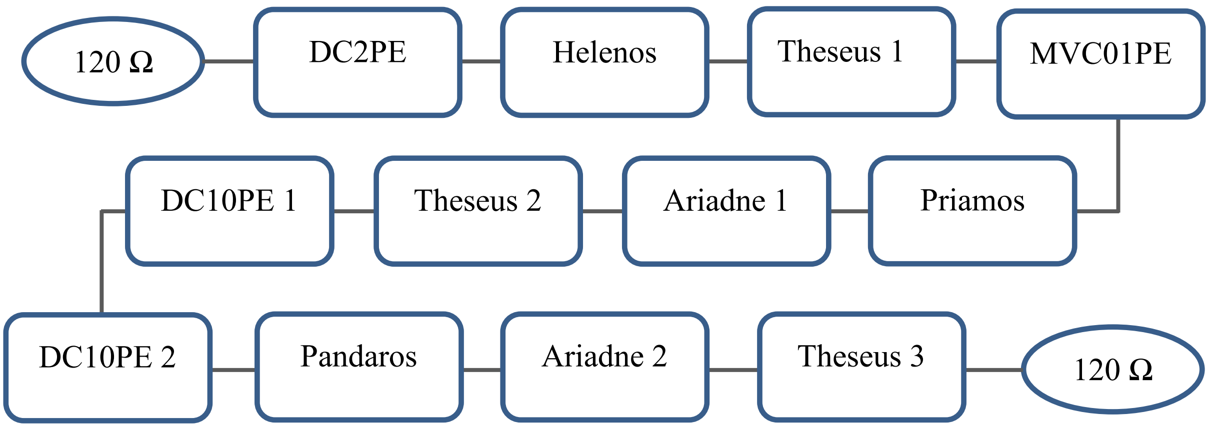


Fig. 4: Example of physical connection

3 Definition of nodes

Each individual unit in the HZM-CAN bus is uniquely identified by means of the defined node type *(↑ 3.1 Node type)* and a node number that is assigned during parameterisation *(↑3.2 Node number)*.

3.1 Node type

The node type defines a certain class of HEINZMANN units that have equal tasks and send or receive largely the same data via the CAN bus.

The following table states the node type as well as the unit type or connection type, depending on whether the description relates to the CAN node, the HEINZMANN unit or the connection between the units.

The following unit types are defined:

Type	ID	Meaning	Control unit
0	DC	Digital Control	Speed governors
1	GC	Generator Control	Generator control units
2	PE	Peripheral Extension	Extension modules
5	AC	Accessory Control	Additional units
6	CM	Customer Module	Customer units (third-party units)
7	PC	Personal Computer	Communication modules
8..14	-	-	Free for extensions
15	ALL	All types except PE and PC	Used for transfers to all

Tab. 3: Node types

In the case of all parameters that relate to the respective unit type, the parameter name features the specified character code, e.g. 404 *Can**PE**NodeNumber*.

Of the parameter numbers for certain unit types that are described in this brochure, only those for which the unit type is released in the corresponding software will be displayed. This means that within a specific software version – even if the HZM-CAN is generally implemented – only those types that are required for the function in question are released. For example, a vehicle does not require a connection to generator control units.

The easiest way to determine what is actually implemented is to refer to the connection types, as these must be released separately by the user ↑4 *Connection between the nodes*.

3.1.1 DC: speed governors (Digital Control)

Speed governors include those with actuators and those with valve control. Examples of speed governors: HELENOS, PANDAROS, MVC01, MVC03, KRONOS 30.

3.1.2 GC: generator control units (Generator Control)

There is currently only 1 generator control unit, the THESEUS.

3.1.3 PE: extension modules (Peripheral Extension)

Periphery modules are units that send the values for their inputs to a master; this is also where they receive values for their outputs. Periphery modules do not function independently, but serve exclusively to increase the number of inputs and outputs on the master. A master can be connected to up to 3 periphery modules as standard, whereas a periphery module can only have 1 master. The master may be of any type except PE and PC, meaning that a periphery module can never itself be the master of another periphery module.

i

The connection between the periphery module and the master does not relate to content. Rather, the periphery module only extends the number of inputs/outputs of the master. The assignment of the transferred values to sensors and switching functions is performed exclusively in the master.

The options made available by a periphery module are determined by the hardware in use as different forms of hardware provide different numbers of inputs and outputs. There are also control units controlled by actuators and others controlled by solenoid valves. In order to uniquely define the periphery module, a number is specified for each periphery module type and must be used when configuring the master.

The actual available number of inputs/outputs varies as a result of the different physical ports, as some of these can be configured. The following tables each state the maximum number. However, the specific system documents make it clear how freely configurable inputs and outputs are used in exceptional cases. The channel number is retained, even if the reconfiguration of ports leads to gaps.

The 2 tables below list the periphery module types that are available, the identifier with which they have been issued and the maximum number of inputs/outputs they can each provide.

PE module		Sensors				Other		
Type	Hardware	Analogue	Temp	PWM	Frequency	Binary	Speed	StG/ fuel
0	DC 2	4	2	4	-	8	2	1
1	DC 6	3	1	3	-	5	2	1
2	ELEKTRA	9	3	3	-	5	-	1
3	DC 1-03	5	2	2	-	12	2	1
4	DC 1-04	8	2	1	-	11	2	3
5	SMC	6	1	1	-	7	-	1
6	MVC01	4	5	1	-	11	2	1
7	ANALOG-IN	6	8	-	-	-	-	-
8	XIOS	117 variable ports					0-2	3
11	DIGITAL-I/O	-	-	-	-	8	-	-
12	MVC03	11	5	2	-	9	2	1
13	DC 11	5	1	3	-	6	2	1
14	DC 10	6	1	1	-	6	2	1
15	DC 8	6	2	4	-	8	2	1
16	MVC04	11	5	1	-	17	2	1
17	PHLOX	1	-	2	-	2	2	-
18	DC 12	3	2	-	-	2	1	1

Tab. 4: Periphery module type and number of inputs

PE module		Outputs			
Type	Hardware	Analogue	PWM	Binary	StG/fuel
0	DC 2	4	4	5	1
1	DC 6	2	2	2	1
2	ELEKTRA	2	2	2	1
3	DC 1-03	2	3	3	1
4	DC 1-04	2	3	5	3
5	SMC	1	0	1	1
6	MVC01	2	2	8	1
7	ANALOG-IN	-	-	-	-
8	XIOS	117 variable ports			3
11	DIGITAL-I/O	-	-	8	-
12	MVC03	-	8	13	1
13	DC 11	2	2	2	1
14	DC 10	1	-	1	1
15	DC 8	1	1	13	1
16	MVC04	0	9	9	1
17	PHLOX	0	2	2	-
18	DC 12	1	-	1	1

Tab. 5: Periphery module type and number of outputs

i

The PE module type SMC includes E-LES SMC and GMA SMC. The PHLOX-PE module must only be used for PHLOX masters.

3.1.3.1 Master of a periphery module

The creation of an item of control unit software involves specifying whether the unit can cooperate with periphery modules at all and, if so, the possible number and type of periphery modules. A unit such as this featuring the corresponding firmware can then be used as a master for periphery modules. It is necessary to restrict this to the software creation stage in order to prevent unnecessarily taking up memory in the master control unit: this is particularly important in the case of small units.

In order to obtain an overview of what is possible with a particular unit, 2489 *PEModulesMax* states the maximum number of periphery modules that can be connected to this control unit. The field elements of 2490 *PEModulesMaxType()* indicate how many of the respective type can be stipulated. The field index corresponds to the periphery module type as listed in ↑ *Tab. 4:* Periphery module type and number of inputs . However, the field size for 2490 *PEModulesMaxType()* is determined based on the maximum approved type in the master, meaning that it may be smaller than the number in the above table.

Even if the master software permits the connection of periphery modules, it is possible at any time to use no modules or fewer modules. This is done by setting the node number in the field 404 *CanPENodeNumber*() to 0 (↑ *3.2 Node number*) or by performing a general switch-off of the connection to the periphery modules (↑ *4.1 Connection*).

In the master, the periphery module type must be parameterised in order for the correct data to be exchanged with the correct unit. This means that in the case of all periphery modules for which a value other than 0 was configured in the master for the node number 404 *CanPENodeNumber()*, the corresponding type must be entered in the same field index in 407 *CanPENodeType()*.

3.1.3.2 Example

If a quantity of 2 is specified for each of the periphery modules DC2 PE, DC6 PE, DC11 PE and DC10 PE,

2490 *PEModulesMaxType*(0) = 2 → max. 2 * DC2 PE
2491 *PEModulesMaxType*(1) = 2 → max. 2 * DC6 PE
2503 *PEModulesMaxType*(13) = 2 → max. 2 * DC11 PE
2504 *PEModulesMaxType*(14) = 2 → max. 2 * DC10 PE,

but all others are set to a quantity of 0 and a maximum of 2 instances are specified (2489 *PEModulesMax* = 2), then only 2 of the possible periphery modules can be connected. 404 *CanPENodeNumber*() and 407 *CanPENodeType()* therefore consist of exactly 2 elements each. The 2 periphery modules may be of the same type, but this is not essential.

2490 *PEModulesMaxType*() has 15 elements in this example.

The variability therefore relates only to the periphery module types that can be used in an application. In this example, the decision depends on the number of additional inputs and outputs that are required and the actuator in use.

3.1.4 AC: additional modules (Accessory Control)

Unlike periphery modules, additional modules are any units that perform independent functional tasks, e.g. Phlox as an ignition control unit, Ariadne as a knock detection module, Kronos 20 as an AFR control unit or different gas positioner variants.

A unit can be connected to up to 5 additional modules as standard, and the unit itself may also be an additional module. Each type – DC, GC, AC and CM – can communicate content with the additional modules. Even in this case, periphery modules are only available for the purpose of extending the inputs and outputs of the additional module.

Additional modules implement various tasks, therefore these units must be issued with an add

+6itional module type featuring a set definition that is listed in ↑*Tab. 6:* Additional module types.

It is only possible to form a point-to-point connection with the additional modules that feature a real type entry (not 0) in the following table and only using the destinations that are specified, ↑*4.1 Connection between exactly 2 units: Point-to-point connection.*

AC module		**Point-to-point connection**
Type	**Hardware**	**Possible destinations**
0	KRONOS 20 (also as SMC) E_LES (also as SMC) GMA (also as SMC) Gas positioners (DC 6-13, DC 6-22, MVC01) ELEKTRA AFRControl ARIADNE PHLOX	Only multipoint connection permitted
1	ELEKTRA GasFlowControl	DC: Speed governors

Tab. 6: Additional module types

Type 0 AC modules either work independently or only use the options for sending or receiving data via multipoint connections ↑*4.2 Sending to all/receiving from all: Multipoint connection* and ↑6 *Multipoint connection.* It is not possible for them to be used in point-to-point connections.

For example, the ARIADNE provides the bus with information on light or hard knocking at the dual-fuel engine. A duel-fuel speed governor or gas positioner that reads this information will subsequently reduce the quantity of gas or cut off the gas completely.

3.1.4.1 Destination of additional modules

In units that want to communicate with an additional module – whether via a point-to-point connection or a multipoint connection – the additional module type must be stated in order for the correct data to be exchanged.

This means that for all additional modules for which a node number 430 *CanAC-NodeNumber()* is stated, the same field index in 435 *CanACNodeType()* must feature an entry for the corresponding type from ↑*Tab. 6:* Additional module types.

In order to quickly detect the capability of a particular item of software, 2549 *AC-ModulesMax* states how many additional modules can be connected to this control unit. The field elements of 2550 *ACModulesMaxType()* indicate how many of the respective type can be stipulated in the process. These maximum figures are fixed in the software and can only be changed by means of a software update. However, it is possible at all times to use fewer modules than specified; the node numbers are set to 0.

The field size of 2550 *ACModulesMaxType*() is determined by the maximum permissible additional module type in the control unit, meaning that it can contain 1 or 2 elements.

Users should also refer to the example in ↑*3.1.3.2 Example*.

3.1.5 CM: customer units (Customer Module)

A customer module is a third-party unit that can be integrated into the HZM-CAN bus in order to read out measured values and potentially set parameters in the HEINZMANN control units. The respective manufacturer integrates a subset of the HZM-CAN protocol into their software for this purpose (normally a PLC). A customer module can theoretically communicate with any other unit type, even enabling communication with a periphery module for the purpose of using its inputs/outputs and actuator/valve controls.

For a detailed description of the implementation and use of a customer module, see 'DG 05 007-d 05-12 HZM-CAN Customer-Modul.pdf'.

PANOPTES displays function as customer modules when integrated into the HZM-CAN bus. This means that they do not communicate via the PC type, unlike DcDesk 2000/CAN and ARGOS/CAN.

3.1.6 PC: communication modules (Personal Computer)

ARGOS/CAN and DcDesk 2000/CAN are PC modules in the context of the HZM-CAN bus. Depending on the maximum possible baud rate, the transfer is faster than in the serial connection. Unlike the serial connection, the PC or ARGOS can communicate with every unit in the bus without needing to change the connection of the communication cable.

ARGOS is normally installed in switch cabinet doors and is therefore firmly integrated into the CAN bus. If the unit is connected to one end of the bus, it must be provided with a 120 Ω terminating resistor (↑*1.4 Terminating resistors*).

A PC/laptop can only be integrated into the CAN bus via the PC-CAN adapter (↑*1.6 PC-CAN adapter)*. The issue of whether the PC can actually be used to access the CAN on the control unit is determined by the release of DcDesk 2000/CAN in the dongle (separate order number).

The respective control unit must of course feature software with HZM-CAN in order for ARGOS or DcDesk 2000/CAN to communicate with it. The node type of the control unit is not significant in this regard.

2 PC units cannot communicate with one another and it is only possible for 1 of them to be actively connected to the control unit. However, multiple instances of DcDesk 2000/CAN can be started on the computer. This enables communication with multiple control units simultaneously on 1 PC. The node number of the DcDesk 2000/CAN and the Argos/CAN does not need to be assigned, as it is automatically detected by the respective program.

In principle, a control unit featuring multiple CAN interfaces with HZM-CAN protocol can communicate with DcDesk 2000/CAN on all of these interfaces. However, it is only possible for communication to be performed 1 interface at a time. This means that, if the control unit communicates with DcDesk 2000/CAN on a CAN interface, it is not possible to establish DcDesk 2000/CAN communication on the other interfaces. If the DcDesk 2000/CAN communication should be changed from 1 interface to another for a control unit such as this, it is recommended that you actively end communication (using F5) prior to the change. Failure to do this means that the other interfaces for DcDesk 2000/CAN will remain blocked until the communication enters timeout. This timeout was introduced to the control unit software on 13 December 2010 and is activated after 5 minutes. Control units featuring software from before this date (see 3858 *CompileDate* and 3859 *CompileYear*) do not feature a timeout, meaning that the communication interface will never be released unless disconnection is performed correctly.

3.1.7 ALL: all node types except PE and PC

Specific senders use this to make certain data generally available. All units of the types DC, GC, AC and CM (i.e. all except PE and PC) can send such data and obtain this data from the bus as a recipient if they require the data for their own functions. To a certain extent, this operating principle corresponds to the data transfer in SAE J1939 CAN communication, ↑6 *Multipoint connection.*

3.2 Node number

Every node number from 1 to 31 may be assigned a maximum of once for each unit type in the HZM-CAN network. Identical node numbers are permitted only if the node type is different.

i *The node number 0 is not permitted and must not be assigned.*

If the parameters for a node number are set to 0, this means that a node generally provided by the firmware should not be used. For example, up to 3 periphery modules can be connected to 1 unit. However, if only 1 of these is required in a particular application then the node numbers of the 2 other periphery modules must be set to 0.

In order to receive data from multipoint connections, the node type and node number must be stated separately for the respective data that is to be evaluated ↑*6.2 Data receipt*. If the senders are additional modules, then the node numbers of these additional modules must also be parameterised ↑*3.2.5 AC destinations.*

For point-to-point connections to other control units in the bus, the node numbers of the other control units must be entered in the type-dependent parameters.

3.2.1 Own node number

The parameter

401 *CanMyNodeNumber*

defines the node number of the user's own control unit.

3.2.2 DC destination

402 *CanDCNodeNumber*	node number of a speed governor
402 *CanOtherNodeNumber*	node number of the other speed governor in ship projects with master/slave engines (2*DC) or with 2 motors on 1 control lever

397 PartnerDCNodeNumber	
398 ThirdDCNodeNumber	
399 FourthDCNodeNumber	node numbers of the other speed governors in ship projects with 2 control levers, with each control lever featuring 2 engines respectively

3.2.3 GC destination in speed governors and generator control units

The generator control unit has the same node number as the speed governor with which it cooperates, meaning that a different node number does not need to be issued in this case. However, 401 *CanMyNodeNumber* must be set such that it is identical in both control units (DC and GC).

A group THESEUS does not feature an assigned speed governor. In order to distinguish between a group THESEUS and a THESEUS with a speed governor, it is established that node numbers from 1 to 20 can only be assigned to 1 THESEUS (and therefore also the speed governor). This represents a deviation from the standard. A group THESEUS receives a node number from 21 to 29. This means that a single CAN bus system can feature a maximum of 20 THESEUS units and 9 group THESEUS units.

3.2.4 PE destinations

404 *CanPENodeNumber()*	node numbers of up to 3 periphery modules

3.2.5 AC destinations

430 *CanACNodeNumber()*	node number of up to 5 additional modules

There are applications that are intended to communicate with multiple type 0 additional modules, which is only possible via a multipoint connection. For this purpose, the index from field 430 *CanACNodeNumber()* pertaining to the module in question must be assigned to a functional parameter. For example, it may be necessary to define which of the additional modules is ARIADNE. In dual-fuel applications, the parameter 10056 *KnockModulACIndex* must be assigned with the field index pertaining to ARIADNE.

3.2.6 CM destination

403 *CanCMNodeNumber* node number of the customer module

3.2.7 PC destination

DcDesk 2000/CAN and ARGOS/CAN assign the node number to themselves and communicate this to the control unit when connection is established. Therefore, the PC node does not need to be configured in the control unit and there is no parameter for this.

3.2.8 Master destination in periphery modules

Periphery modules feature exactly 1 master to which they provide their input values and from which they receive output values. The parameter

403 *CanMasterNodeNumber*

must stipulate the master node number. The node type of this master is determined via ↑4 *Connection between the nodes*.

4 Connection between the nodes

The connection between the individual nodes in the CAN network can be established from exactly 1 unit to exactly 1 other unit, or from 1 unit to all others or a subset. The first connection type is called point-to-point connection and the second is called multipoint connection.

i *This relates only to the logical type of data transfer, not the establishment of a physical network. CAN is a line with exactly 2 endpoints and 120 Ω terminating resistors at both ends.*

Before information can be exchanged between units, the connection to the node type of the other unit must be established. The release is performed using the following parameters, which activate the respective connection type:

DC:	4400 *CanCommDCOn* = 1	connection to speed governors
GC:	4401 *CanCommGCOn* = 1	connection to generator control units
PE:	4402 *CanCommPEOn* = 1	connection to periphery modules
AC:	4405 *CanCommACOn* = 1	connection to additional modules
CM:	4406 *CanCommCMOn* = 1	connection to customer modules
PC:	DcDesk 2000/CAN or ARGOS/CAN released automatically.	
ALL:	4415 *CanCommAllOn* = 1	release of multipoint connection

It is essential for point-to-point connections to be activated on both sides. In the case of multipoint connections, it is only necessary to release the sender type in the recipient but the multipoint connection must be activated in both the sender and the recipient.

You do not need to activate a connection to your own unit type as you do not need a connection to yourself. However, the activation is necessary in the event that a point-to-point connection to another unit of the same type is required.

i *Not all of the connection types mentioned above are implemented in a particular application; instead, only those that are required for the unit function in question are implemented.*

4.1 Connection between exactly 2 units: Point-to-point connection

In this connection type, which is called point-to-point, a unit sends information to exactly 1 other unit in the bus. The logical connection to the recipient unit type must be released in the sender and the logical connection to the sender unit type must be released in the recipient ↑4 *Connection between the nodes*. The sender must also recognise the node number of the target unit and the recipient must recognise the node number of the sender.

Therefore, if data is to be exchanged between a speed governor and a generator control unit, then the GC connection type must be activated in the speed governor and the DC connection type must be activated in the generator control unit. If another periphery module is connected to the speed governor at the same time, then the PE connection type must also be activated in the speed governor and the DC connection type must be activated in the periphery module. If the generator control unit is also intended to communicate with other generator control units, then the GC connection type must be released in all of these.

Whereas the data transfer in the point-to-point connection between the generator control unit and the speed governor has a fixed structure and does not require external configuration, the connection between a customer module and a control unit needs to be parameterised. See 'DG 05 007-d 05-12 HZM-CAN Customer-Modul.pdf'.

The use of a periphery module to extend the number of inputs and outputs of your control unit also requires precise configuration on both sides, ↑ *5.3 Master* and ↑ *5.2 Periphery module*.

The communication with and between additional modules takes place almost exclusively via the multipoint connection.

4.2 Sending to all/receiving from all: Multipoint connection

If a unit generally stores certain data on the bus and if this data is specifically evaluated by recipients, then the connection is designated as a multipoint connection ↑ 6 *Multipoint connection*.

This means that a unit transfers the information without a specific target. In terms of content, it stores specifically defined data telegrams on the bus ↑ *6.1 Data sending* and other units (all except PE and PC) retrieve only that information that is of interest to them ↑ *6.2 Data receipt*.

To a certain extent, this operating principle corresponds to the data transfer in SAE J1939 CAN communication.

The sender and the interested recipients must feature 4415 *CanCommAllOn* = 1 as a setting.

Whereas the sender can send a telegram to all without having to specifically define the recipients to which the transfer should be sent, the recipient must always release the connection to the node type of the sender from which it wants to receive something ↑ 4 *Connection between the nodes*. If this is not the case, it cannot receive the transfer. The recipient can check whether the sender is issuing stable transfers; if this is not the case, the recipient will issue a timeout error ↑ *7.2 Monitoring CAN communication*.

It is even possible to have a situation in which different senders send identical commands 'to all', but the commands contain differing information and must therefore be evaluated accordingly.

In this way, for example, the speed and measured power can be sent as data of general interest. If an additional module measures the measured power and a speed governor measures the speed, both of these telegrams 'to all' can be saved to the bus. At this point, the AC connection type must be released in a recipient that wants to evaluate the measured power and the DC connection type must be released in a recipient that requires the speed.

4.3 Display of the connection status

In the parameters

2410 *CanDCNodeState31to16*	2411 *CanDCNodeState15to01*
2412 *CanGCNodeState31to16*	2413 *CanGCNodeState15to01*
2414 *CanPENodeState31to16*	2415 *CanPENodeState15to01*
2420 *CanACNodeState31to16*	2421 *CanACNodeState15to01*
2422 *CanCMNodeState31to16*	2423 *CanCMNodeState15to01*
2424 *CanPCNodeState31to16*	2425 *CanPCNodeState15to01*

the current connection status of your control unit to the different unit types is displayed bit by bit. Here, the activated bits correspond to the node number (1 to 31) for the specific module of the respective type from which stable telegrams are received. This can either be a point-to-point connection or the reception of a send procedure from a multipoint connection as, in the latter case, the sender is also identified by the node type and node number and can be indicated.

DcDesk 2000/CAN and ARGOS/CAN assign the node number to themselves; 2424 *CanPCNodeState31to16* and 2425 *CanPCNodeState15to01* may therefore display different node numbers each time a new connection is established with a laptop or the ARGOS.

i *The current connection status is displayed in the special window 'Control unit/Settings/HZM-CAN configuration' in DcDesk (serial and CAN).*

5 Master and periphery module

The connection between the master and periphery module is a point-to-point connection. Multiple periphery modules can be connected to a single master, but each periphery module only ever has exactly 1 master. The master communicates separately with its periphery modules, for which it requires their node numbers ↑ *3.2.4 PE destinations*. The following chapters describe the configuration of the data transfer between the master and the periphery module.

5.1 Send rate

The data is transferred within a set time frame that can be specified via a parameterisable send rate.

If '(change)' is stated with regard to the send rate in the following chapters, this means that the corresponding values are sent each time a change is made and at the latest after the send rate has elapsed.

If '(inhibit)' is stated with regard to the send rate, the values are not sent at a more frequent rate than that specified by the send rate (known as inhibit time). Instead, the values are only sent if at least 1 value has changed from the last send procedure. In this case, a send rate of 0 s therefore means that the respective telegram is sent only in the event of changes.

If '(time)' is stated for the send rate, the data is sent within a set time frame for the send rate. This is performed regardless of whether changes have been made.

To avoid increasing the bus load unnecessarily, take care when determining the send rate.

If there is no parameter for the send rate or if it is set to 0 in the event of '(change)' or '(time)', it corresponds to the main loop frequency of the control unit, which depends on the processor being used. The data is then sent with each passage of the main loop. The following table lists the main loop frequency of each control unit.

Control unit	**64 Hz** **15.625 ms**	**62.5 Hz** **16 ms**	**100 Hz** **10 ms**
DC 1/PRIAMOS	x		
DC 2/HELENOS	x		
KRONOS 30	x		
DC 5/ARCHIMEDES		x	
DC 6/PANDAROS		x	
KRONOS 20		x	
KASSANDRA		x	
ELEKTRA		x	
E-LES		x	
GMA		x	
DC 7		x	
DC 8		x	

Control unit	64 Hz 15.625 ms	62.5 Hz 16 ms	100 Hz 10 ms
DC 10		x	
DC 11		x	
DC 12		x	
DigitalIO		x	
AnalogIn		x	
THESEUS	x		
MVC 01/DARDANOS I	x		
MVC01-3G			x
MVC03/DARDANOS III			x
MVC04/DARDANOS IV			x
MVC03-01			x
ARIADNE			x
PHLOX			x
XIOS			x

Tab. 7: Control units: main loop frequency

5.2 Periphery module → master

The periphery module only communicates with its master via a point-to-point connection. The master control unit may be of type DC, GC, AC and CM. Each periphery module may have only 1 master.

The periphery module can transfer the values of all its inputs to the master, where the values are subject to targeted evaluation.

5.2.1 Actuator/filling

Transfer of the current control unit value to the master (MVC periphery modules send back the received filling value).

Send rate (inhibit): 440 *CanActPosSendRate*

Transfer values:

If the PE module features at least 1 actuator:

2300 *Act(1)Pos*
2302 Act2Pos
2303 Act3Pos

5910 *Actuator(1)On*
5930 Actuator2On
5940 Actuator3On

Otherwise (MVC periphery modules):

	2350 *FuelQuantity*	
Master display:	2305 *PEActPos(x)*	x = 0..2
	2320 *PEActuatorOn(x)*	x = 0..2
Activation:	4440 *CanTelActuatorPosOn*	

5.2.2 Binary inputs

Transfer of the values from the binary inputs of the periphery module.

Send rate (inhibit):	441 *CanDigInSendRate*
Transfer value:	2811 *DigitalIn1* onwards
Master display:	2810 *SwitchEngineStop* onwards, depending on assignment in 810 *FunctEngineStop* onwards or 20810 *CommEngineStop* onwards and 24810 *ChanTypEngineStop* and 24910 *PEIxEngineStop*
Activation:	4441 *CanTelDigitalInOn*

The binary inputs that are configured in this way can be assigned to the respective switching functions in the master ↑*5.2.10 Configuring the switching function and the sensor in the master*.

5.2.3 Sensor inputs: XIOS periphery module

The XIOS periphery module uses this method to transfer all configured analogue, temperature, PWM and frequency inputs – that is to say, all values that can be assigned to the sensors in the master ↑*5.2.10 Configuring the switching function and the sensor in the master.*

Send rate (inhibit):	442 *CanSensorInSendRate*
Master display:	2900 *Setpoint1Ext* onwards, depending on 900 *AssignIn_Setp1Ext* onwards, 4900 *ChanTypSetp1Ext* onwards and 4950 *PEIxSetp1Ext* onwards
Transfer values:	32000 *P089_(MC.DI1)_FI_FreqIn* up to 32392 *P117_(MD.AI8)*
Activation:	automatic, provided at least 1 of these inputs is configured

5.2.4 Sensor inputs: other periphery modules

Other periphery modules as XIOS send the values from analogue, temperature and PWM inputs via different telegrams. As a result, they also have to be activated separately.

5.2.4.1 Analogue inputs

Transfer of the values of the analogue inputs of the periphery module, provided at least 1 sensor is connected. The parameters from 4900 *CanTelAnalogIn1Used* onwards are used to specify which analogue inputs are used and should be sent. The inputs that are configured in this way can be assigned to the respective sensors in the master ↑*5.2.10 Configuring the switching function and the sensor in the master*.

Send rate (inhibit):	442 *CanAnalogInSendRate*
Master display:	2900 *Setpoint1Ext* onwards, depending on 900 *AssignIn_Setp1Ext* onwards, 4900 *ChanTypSetp1Ext* onwards and 4950 *PEIxSetp1Ext* onwards
Transfer values:	3510 *AnalogIn1_Percent* onwards
Channel release:	4900 *CanTelAnalogIn1Used* onwards
Activation:	4442 *CanTelAnalogInOn*

5.2.4.2 Temperature inputs

Transfer of the values for the temperature inputs of the periphery module, provided 1 temperature sensor is connected. The issues of whether temperature inputs are used, which temperature inputs are used and which are to be sent are determined using the parameters from 4920 *CanTelTempInUsed* onwards. The temperature inputs that are configured in this way can be assigned to the respective sensors in the master ↑ *5.2.10 Configuring the switching function and the sensor in the master*.

Send rate (inhibit):	443 *CanTempInSendRate*
Master display:	2900 *Setpoint1Ext* onwards, depending on 900 *AssignIn_Setp1Ext* onwards, 4900 *ChanTypSetp1Ext* onwards and 4950 *PEIxSetp1Ext* onwards
Transfer values:	3550 *TempIn1* onwards
Channel release:	4920 *CanTelTempInUsed* onwards
Activation:	4443 *CanTelTempInOn*

5.2.4.3 PWM inputs

Transfer of the values of the PWM inputs of the periphery module, provided at least 1 PWM input channel is used. The issues of whether PWM inputs are used, which PWM inputs are used and which are to be sent are determined using the parameters from 4940 *CanTelPWMIn1Used* onwards. The PWM inputs that are configured in this way can be assigned to the respective sensors in the master ↑*5.2.10 Configuring the switching function and the sensor in the master*.

Send rate (inhibit):	444 *CanPWMInSendRate*
Master display:	2900 *Setpoint1Ext* onwards, depending on 900 *AssignIn_Setp1Ext* onwards, 4900 *ChanTypSetp1Ext* onwards and 4950 *PEIxSetp1Ext* onwards
Transfer values:	3500 *PWMIn1* onwards
Channel release:	4940 *CanTelPWMIn1Used* onwards
Activation:	4444 *CanTelPWMInOn*

5.2.5 Speed

Transfer of the speed(s) measured by the periphery module.

Send rate (inhibit):	445 *CanSpeedSendRate*
Transfer value:	2001 *SpeedPickUp1* 2002 *SpeedPickUp2*
Activation:	4445 *CanTelSpeedOn*

5.2.6 Vehicle speed

Transfer of the vehicle speed measured by the periphery module.

Send rate (inhibit):	446 *CanVelocitySendRate*
Transfer value:	3300 *Velocity*
Activation:	4446 *CanTelVelocityOn*

5.2.7 Particular measured values

The measured values are sent each time a change is made. This command varies according to the type of periphery module.

Send rate (change):	fixed, 0
Activation:	4444 *CanTelMeasurementsOn*

DC 11 PE:

Transfer value:	2050 *SpeedVariance*
Master display:	2050 *SpeedVariance*

MVC01 PE, MVC03 PE, MVC04 PE:

Transfer value:	Transfer of the minimum, average and maximum value of the exhaust temperatures of all cylinders. 12570 *ExhaustTempAverage* 12572 *ExhaustTempMin* 12573 *ExhaustTempMax*
Master display:	2911 *ExhaustTempMax* 2912 *ExhaustTempMin* 2913 *ExhaustTempAverage*

PHLOX PE:

Transfer value:	13650 *SparkDuration* onwards
Master display:	13650 *SparkDuration* onwards

5.2.8 Error status

Transfer of the periphery module error status, along with the PHLOX status in the case of PHLOX. The status is sent every second, but an additional send procedure is performed whenever a change is made.

Send rate (change):	fixed, 1 s
Master display:	2440 *CanPEError(x)* x = 0...2 3049 *ErrPECommonAlarm* 3089 *ErrPEFatalError* if the master displays the errors as bits or error counters
or	23002 *ErrCanPE(x)* x = 0...2 if the master displays the errors as an error status

The index in the error parameters of the master corresponds precisely to the index that pertains to the respective node number in 404 *CanPENodeNumber()*.

5.2.9 Output configuration

Rather than configuring its outputs itself, the periphery module receives the relevant values from the master. However, the periphery module must feature a definition regarding how an output is to be configured. This is also the case, for example, if a port has to be configured as an output first or if it can optionally be a current output or a voltage output.

The publications

Control device for conventional injection with actuators, publication no. DG 07 001-e

and

DARDANOS Basic Information, publication no. MV 09 001-e

feature a detailed description of the configuration of variable ports for each control unit type.

Once the fundamental output configuration has been completed, the output range still needs to be defined. For PWM outputs, the output value range is used to define the minimum and maximum duty cycle, e.g. [10, 90]%. For analogue outputs, the definition states whether e.g. only [4, 20] mA of the maximum possible current range may be used.

5.2.9.1 XIOS

5.2.9.1.1 Analogue outputs

The modules A2, A5 and A6 are available for analogue outputs. They can be connected anywhere, meaning that ports 1 to 84 can be used.

A maximum of 2 galvanically isolated analogue outputs can be connected to modules A5 and A6. Therefore, only the first 2 ports of the slot that is in use can be configured in this case – the other 6 ports must receive 0 as the configuration value.

The example shows parameterisation as performed on port 1.

Parameter	**Meaning**
30020 *P001_(1.1)_IO_RefLow*	Lower reference value, e.g. 0.5 V or 4 mA
30021 *P001_(1.1)_IO_RefHigh*	Upper reference value, e.g. 4.5 V or 20 mA
24000 *P001_(1.1)_Config*	Selection of the analogue output type 29: 0..5 V 30: 0..24 mA

Tab. 8: XIOS: Analogue outputs

5.2.9.1.2 PWM outputs

The modules A2 and A8 are available for PWM outputs. They can be connected anywhere, meaning that ports 1 to 84 can be used.

A maximum of 2 PWM outputs can be connected to module A8. Therefore, only the first 2 ports of the respective slot can be used in this case – the other 6 ports must receive 0 as the configuration value.

The example shows parameterisation as performed on port 1.

Parameter	Meaning
30020 *P001_(1.1)_IO_RefLow*	Lower duty cycle
30021 *P001_(1.1)_IO_RefHigh*	Upper duty cycle
30025 *P001_(1.1)_PO_Freq*	Output frequency
24000 *P001_(1.1)_Config*	Selection of the PWM output type 33: high side 34: low side 35: half bridge

Tab. 9: XIOS: PWM outputs

The duty cycle can be [0, 100]%, but [10, 90]% of the PWM ratio is normally used in order to enable monitoring at the destination. The output frequency can be set between 126 Hz and 15126 Hz.

5.2.9.1.3 Binary outputs

The modules C, D, A2, A5 and A8 are available for binary outputs. The A modules can be connected anywhere, meaning that ports 1 to 88, 91, 92, 108 and 109 can be used.

A maximum of 2 binary outputs can be connected to modules A5 and A8. Therefore, only the first 2 ports of the slot that is in use can be configured in this case.

The example shows parameterisation as performed on port 1.

Parameter	Meaning
24000 *P001_(1.1)_Config*	Selection of the binary output type 31: high side 32: low side

Tab. 10: XIOS: Binary outputs

5.2.9.2 Other control units

5.2.9.2.1 Analogue outputs

The analogue outputs can be designed as current outputs or voltage outputs. However, for the voltage outputs in particular, it is usually not the maximum output range of approx. 0..22 mA that is desired but the standard output range of 4..20 mA.

The parameters from

1641 AnalogOut1_RefLow or 1641 CurrentOut1_RefLow

1642 AnalogOut1_RefHigh or 1642 CurrentOut1_RefHigh

and above are available for adjusting the output range. The value to be entered relates to the maximum output value and must be stated as a percentage in the periphery modules of type DC 1 (PE type 3 and 4) and DC 2 (PE type 0). In all other control units, the output range can be parameterised in the electrical unit itself.

It is not possible to change the definition of the connection type (current or voltage) during operation. The parameters must therefore be stored following configuration and the control unit must then be reset.

5.2.9.2.2 PWM outputs

A PWM ratio of between 10 and 90% is normally desired. The parameters as of 1601 *PWMOut1_RefLow* and 1602 *PWMOut1_RefHigh* should be used to adjust the output range. The limits are entered directly as a percentage of the PWM ratio.

The frequency of the PWM signals can be set for all outputs with the parameter 1625 *PWMOutFrequency*.

The frequency of the power output (PWM output 5) of the type DC 2 (PE type 0) periphery module is defined separately using the parameter 1626 *PowerOutFrequency*.

5.2.10 Configuring the switching function and the sensor in the master

The values received by the periphery module from analogue, PWM and binary inputs are assigned in the master to the corresponding sensors and switching functions. The publications

Control device for conventional injection with actuators, publication no. DG 07 001-e

and

DARDANOS Basic Information, publication no. MV 09 001-e

feature a detailed description of the parameterisation.

5.3 Master → periphery module

The master and the periphery module communicate with one another exclusively via a point-to-point connection. The master control unit may be of type DC, GC, AC or CM. Each master can communicate with multiple periphery modules. However, a periphery module may have precisely 1 master only.

The master can determine values for all outputs of the periphery module and send them there. Of course, it only makes sense to send values for outputs that are also present and configured accordingly in the periphery module with which communication is being performed ↑ *Tab. 5: Periphery module type and number of outputs* and ↑ *5.2.9 Output configuration*. This means that the only aspect defined in the master is the value that should be transferred to an output of the periphery module. The configuration of the output port in itself (current/voltage, range), i.e. the configuration of the hardware output, is performed in the periphery module.

For the purpose of activating a value transfer, the following chapters each specify 3 parameter numbers that relate to the 3 assignable periphery modules in 404 *CanPENodeNumber()* and 407 *CanPENodeType()* in the same order. In each case, the specified data transfer rate (send rate) is the same for all 3 periphery modules.

The configuration of outputs can be performed in the control unit in 2 different ways. Either direct communication is performed with each output of a periphery module via the defined output parameter, as is the case for all dedicated items of hardware in the master, or the configuration is performed via common outputs.

i *The easiest option is to perform the configuration via 'Control unit/Settings' in the PC tool DcDesk 2000. This features separate menu items for binary, analogue and PWM outputs as well as common outputs.*

5.3.1 Output configuration without common outputs

In this case, the master features parameters as of 450 *PEDigOut1_Assign* and 9120 *PEDigOut1:Param*() for configuring each individual (explicitly intended) binary output on the periphery modules. In the same way, the assignment parameters as of 480 *PEAnaOut1_Assign* are used to communicate with analogue outputs on the periphery modules and the assignment parameters as of 455 *PEPWMOut1_Assign* are used to communicate with PWM outputs on the periphery modules.

When using multiple periphery modules in the system, the parameters mentioned above contain the outputs of the second periphery module after those of the first, etc. See ↑ *Tab. 5:* to determine how many outputs of the respective type are featured in a particular periphery module. In this regard, it does not matter whether or not variable ports have been configured for this type. The numbers remain the same, even if this causes gaps.

Communication is therefore performed with a highly specific output via its number in the parameter name. It is only necessary to configure the source from which it should receive its values.

5.3.2 Output configuration with common outputs

This case involves parameters as of 11400 *Out1:Assign* and 15400 *Out1:Destination* in which the source and target are configured respectively. The number in the parameter name Out1, Out2, etc. is only a serial number and does not make any reference to the output that is actually in use. This is only defined upon the definition of the target.

The following parameterisation enables communication with a particular output on a periphery module:

Transfer value:	11400 *Out1:Assign* 11401 *Out1:ValueMin* 11402 *Out1:ValueMax* 11403 *Out1:DigOutBitMask*
Display value:	13400 *Out1:Value*
Target unit:	15400 *Out1:Destination* = 2 for periphery module 15401 *Out1: PEIx* = 0..2 Index of periphery module in 404 *CanPENodeNumber()*

Channel type in target unit: 15402 *Out1:OutputType*
0: analogue output
1: PWM output
2: binary output
3: filling output (actuator or valve control)

Channel number in target unit: 15403 *Out1:OutputNo* = 1

↑ *Tab. 5*: shows how many outputs of the respective type are featured in a particular periphery module, therefore indicating the maximum possible channel number.

See the brochure 'DG 07 001-d' for the meaning of the parameters 11401 to 11403.

5.3.3 Filling setpoint

When a setpoint is transferred to the periphery module, this value is used in the periphery module to perform either actuator control or valve control. Potentially necessary linearisation curves that transfer a filling setpoint to an actuator target value must be parameterised at the point where the actuator is connected – in this case, this is performed in the periphery module. The same applies for the conversion of the filling level to control times for a solenoid valve.

i

HEINZMANN recommends the value 0 as the send rate for the filling setpoint. This ensures that the setpoints reach the target in the fastest possible time.

You must generally use

Send rate (time): 440 *PEFuelSetpSendRate* = 0

Activation: 4440/4450/4490 *PExFuelSetpointOn* (x = 1..3)

to activate the transfer. The configuration of the other parameters depends on whether or not the common outputs are present.

5.3.3.1 Output configuration without common outputs

The displayed transfer values as of 2355 *PEFuelQuantity* are determined by the assignment of the field 9700 *PEFuelOut:Assign()*. In this field, you must enter the parameter number of the value that should be transferred to the periphery module as the filling setpoint.

Transfer value: 2355 *PEFuelQuantity*

Source configuration: 9700 *PEFuelOut:Assign()*

The first field element normally applies to the first periphery module from 404 *CanPENodeNumber()* with actuator or valve control. The second field element then applies to the second periphery module, and the third applies to the third. However, if a periphery module is not used to control actuators or solenoid valves, then it does not receive an assignment parameter here either. On the other hand, if a periphery module features multiple actuators, these also consecutively occupy multiple assignment spots. The values for the next periphery module shift accordingly in both cases. Therefore, depending on the periphery modules that are connected, the field index of 9700 *PEFuelOut:Assign()* is not always the same as their indices in 404 *CanPENodeNumber()*. In other words, it is only identical if each of the agreed periphery modules features exactly 1 actuator or participates in valve control.

In order to check the values that are to be transferred at a glance, the first value assigned in 9700 *PEFuelOut:Assign(0)* is displayed in 2355 *PEFuelQuantity*, the second in 2356 *PEFuelQuantity2* and so on.

2355 *PEFuelQuantity* is transferred to the first periphery module with actuator or valve control that has received a node number > 0 in the field 404 *CanPENodeNumber()*, 2356 *PEFuelQuantity2* is transferred to the second periphery module with actuator or valve control and so on.

5.3.3.2 Output configuration with common outputs

The following parameterisation is used to transfer the setpoint for a certain actuator or the filling level for solenoid valves on a periphery module:

Transfer value:	13400 *Out1:Value*
Source configuration:	11400 *Out1:Assign*
Target configuration:	15400 *Out1:Destination* = 2 15401 *Out1:PEIx* index of periphery module in 404 *CanPENodeNumber()* 15402 *Out1:OutputType* = 3

Valve control or if only 1 control unit can be connected to the periphery module
15403 *Out1:OutputNo* = 1

Multiple actuators can be connected to the periphery module
15403 *Out1:OutputNo* = actuator number

The displayed transfer value as of 13400 *Out1:Value* is determined by the assignment of 11400 *Out1:Assign*. Here, you must enter the parameter number of the value that should be transferred to the periphery module as the filling setpoint.

5.3.3.3 Examples

See below for several usage examples for the actuator of a connected periphery module and the corresponding parameterisation for this purpose.

5.3.3.3.1 Second actuator in V-engines

If the periphery module is used in order to transfer the same filling setpoint to a second actuator on the V-engine as was transferred to the first actuator on the speed governor control unit, then the following entries must be performed:

Without common outputs:	9700 *PEFuelOut:Assign(0)* = 2350
With common outputs:	11400 *Out1:Assign* = 2350 15400 *Out1:Destination* = 2 15402 *Out1:OutputType* = 3 15403 *Out1:OutputNo* = 1

This is because 2350 *FuelQuantity* is the filling level generated by the speed governor circuit for the entire engine. In this case, 2355 *PEFuelQuantity* and 13400 *Out1:Value* are identical to 2350 *FuelQuantity*.

5.3.3.3.2 Gas throttle valve in dual-fuel engines

If the periphery module is used in order to operate a gas actuator on a dual-fuel engine, then the following parameterisation must be performed:

Without common outputs:	9700 *PEFuelOut:Assign(0)* = 12022
With common outputs:	11400 *Out1:Assign* = 12022 15400 *Out1:Destination* = 2 15402 *Out1:OutputType* = 3 15403 *Out1:OutputNo* = 1

This is because 12022 *GasFuelQuantity* is the gas filling level. In this case, 2355 *PEFuelQuantity* and 13400 *Out1:Value* are identical to 12022 *GasFuelQuantity*.

5.3.3.3.3 Bypass control

If the periphery module is used to control a bypass, then the following parameterisation must be performed:

Without common outputs:	9700 *PEFuelOut:Assign(0)* = 2686
With common outputs:	11400 *Out1:Assign* = 2686 15400 *Out1:Destination* = 2 15402 *Out1:OutputType* = 3 15403 *Out1:OutputNo* = 1

This is because 2686 *BypassValve* is the setpoint for the bypass. In this case, 2355 *PEFuelQuantity* or 13400 *Out1:Value* is identical to 2686 *BypassValve*.

5.3.3.3.4 Wastegate control

If the periphery module is used to control a Wastegate, then the following parameterisation must be performed:

Without common outputs:	9700 *PEFuelOut:Assign(0)* = 2685
With common outputs:	11400 *Out1:Assign* = 2685 15400 *Out1:Destination* = 2 15402 *Out1:OutputType* = 3 15403 *Out1:OutputNo* = 1

This is because 2685 *WasteGate* is the setpoint for the Wastegate. In this case, 2355 *PEFuelQuantity* and 13400 *Out1:Value* are identical to 2685 *WasteGate*.

5.3.4 Binary outputs

Definition and transfer of the values for the binary outputs of the periphery module.

i *HEINZMANN recommends the value 0 as the send rate for the binary outputs. This ensures that the setpoints reach the target in the fastest possible time.*

You must generally use

Send rate (change):	441 *PEDigOutSendRate*
Activation:	4441/4451/4491 *PExDigOutOn* (x = 1..3)

to activate the transfer. The configuration of the other parameters depends on whether or not the common outputs are present.

5.3.4.1 Output configuration without common outputs

Transfer value:	2470 *PEDigitalOut1*
Source configuration:	450 *PEDigOut1_Assign*

5.3.4.2 Output configuration with common outputs

Transfer value:	13400 *Out1:Value*
Source configuration:	11400 *Out1:Assign*
Target configuration:	15400 *Out1:Destination* = 2 15401 *Out1:PEIx* index of periphery module in 404 *CanPENodeNumber()* 15402 *Out1:OutputType* = 2 15403 *Out1:OutputNo* = no. of binary output

5.3.5 Analogue outputs

Definition and transfer of the values for the analogue outputs of the periphery modules. You must generally use

Send rate (inhibit):	442 *PEAnalogOutSendRate*
Activation:	4442/4452/4492 *PExAnalogOutOn* (x = 1..3)

to activate the transfer. The configuration of the other parameters depends on whether or not the common outputs are present.

5.3.5.1 Output configuration without common outputs

Transfer value:	2480 *PEAnaOut1*
Source configuration:	480 *PEAnaOut1_Assign*

5.3.5.2 Output configuration with common outputs

Transfer value:	13400 *Out1:Value*
Source configuration:	11400 *Out1:Assign*
Target configuration:	15400 *Out1:Destination* = 2 15401 *Out1:PEIx* index of periphery module in 404 *CanPENodeNumber()* 15402 *Out1:OutputType* = 0 15403 *Out1:OutputNo* = no. of analogue output

5.3.6 PWM outputs

Definition and transfer of the values for the PWM outputs of the periphery module. You must generally use

Send rate (inhibit):	443 *PEPWMOutSendRate* (default: 100 ms)
Activation:	4443/4453/4493 *PExPWMOutOn* (x = 1..3)

to activate the transfer. The configuration of the other parameters depends on whether or not the common outputs are present.

5.3.6.1 Output configuration without common outputs

Transfer value:	2475 *PEPWMOut1*
Source configuration:	455 *PEPWMOut1_Assign*

5.3.6.2 Output configuration with common outputs

Transfer values:	13400 *Out1:Value*
Source configuration:	11400 *Out1:Assign*
Target configuration:	15400 *Out1:Destination* = 2 15401 *Out1:PEIx* index of periphery module in 404 *CanPENodeNumber()* 15402 *Out1:OutputType* = 1 15403 *Out1:OutputNo* = no. of PWM output

5.3.7 Deleting errors

This involves the 'ErrorReset' request that was received by a communication module such as DcDesk 2000, hand programmer, ARGOS or the switching function 2828 *SwitchErrorReset* being forwarded to the periphery module. The errors are deleted on the master and periphery module simultaneously.

Send rate:	As required
Activation:	4444/4454/4494 *PExErrorResetOn* (x = 1..3)

5.3.8 Triggering AutoReset

This involves the 'AutoReset' request that was received by a communication module such as DcDesk 2000, hand programmer or ARGOS being forwarded to the periphery module. The master and the periphery module boot simultaneously.

Send rate:	As required
Activation:	4445/4455/4495 *PExAutoResetOn* (x = 1..3)

6 Multipoint connection

The following chapters describe the configuration of senders and recipients in multipoint connections so that the data from all control units (except PE and PC modules) can be used.

6.1 Data sending

4415 *CanCommAllOn* = 1 must be set in the sender. The only parameterisation that is required concerns the specific data that the sender should save to the bus. The send rate is set at 50 ms for all data for which sending is activated, regardless of whether or not values have changed.

It is permitted for different senders to issue identical data types – the selection is made when configuring the recipient.

6.1.1 Speed

It is only permitted to send the speed once it has been measured. This means that this command can only be activated in speed governors and in certain additional modules.

4330 *AllSendSpeedOn* = 1	2000 *Speed* should be saved to the CAN bus

6.1.2 Relative measured power

It is only permitted to send the relative measured power once the power itself has been measured. This means that this command may only be activated in speed governors and in certain additional modules.

4332 *AllSendPowPercentOn* = 1	3232 *RelativePower* should be saved to the CAN bus

6.1.3 Error status of additional modules

As this relates to the error status of additional modules, the command can only be sent from these additional modules.

4334 *AllSendErrorStatusOn* = 1	save own error status to the CAN bus

The error status is sent every 50 ms, as well as whenever a change is made. The structure of the error status is described in detail in ↑*6.2.4 Status of additional modules or speed governor modules*.

6.1.4 Status of additional modules or speed governor modules

This command can be sent by additional modules or speed governors.

4335 *AllSendStatusOn* = 1 save own status to the CAN bus

The status is sent every 50 ms, as well as whenever a change is made. The structure of the error status is described in detail in ↑ *6.2.4 Status of additional modules or speed governor modules*.

Parameter 4335 AllSendStatusOn can be viewed only in those modules that are able to provide the status data on the basis of their function.

6.1.5 Setpoint speed of speed governor modules

The speed setpoint itself must be measured and determined, meaning that this command can only be sent by speed governors.

4336 *AllSendSpeedSetpOn* = 1 2031 *SpeedSetp* should be saved to the CAN bus

6.1.6 Boost pressure

The boost pressure itself must be measured. This means that this command can only be activated in speed governors and in certain additional modules.

4338 *AllSendBoostPressOn* = 1 2904 *BoostPressure* should be saved to the CAN bus.

6.1.7 Exhaust temperature

The exhaust temperature itself must be measured.

Save 4342 *AllSendExhaustTempOn* = 1 2911 *ExhaustTemp* to the CAN bus if only 1 sensor is measured
Save 12573 *ExhaustTempMax* if multiple exhaust temperature values are measured (per cylinder or per bank)

6.1.8 Filling/injection quantity/actuator position

This value can be sent by all units that determine a filling or actuator position, e.g. speed governors or gas positioners.

4344 *AllSendFuelSetpOn* = 1 2350 *FuelQuantity* should be saved to the CAN bus

6.1.9 AutoReset and ErrorReset

Here, commands received by a communication module such as DcDesk 2000, hand programmer or ARGOS are forwarded to all other units (except PE and PC modules). The 'ErrorReset' command can also be received via the switching function 2828 *SwitchErrorReset*.

4340 *AllSendAutoResetOn* = 1	save 'AutoReset' request to the CAN bus
4341 *AllSendErrorResetOn* =1	save 'ErrorReset' request to the CAN bus

The error deletion process and the boot process are performed simultaneously in all modules that are recipients of ALL messages.

6.2 Data receipt

In the case of senders, it is only necessary to define the data that they should generally save to the bus. In contrast, it is necessary for the user to define both the specific data that should be evaluated by each individual recipient and the specific sender that sends this data. The user must know the node number and the node type of the send unit in order to do this.

As with the sender, the recipient must feature the setting 4415 *CanCommAllOn* = 1. The connection type of the sender must also be released in the recipient, e.g. 4405 *CanCommACOn* = 1 (↑4 *Connection between the nodes)*.

In order to receive and evaluate data that is generally connected to the bus, 3 parameters per receive value are also required:

Node number of the sender
Node type of the sender
Activation of the function, i.e. actual use of the value

If either the node number is assigned as 0 and/or the connection type is not activated and/or the activation function switches off automatically, then the respective value is not read by the CAN – it remains at the initialisation value of 0 as a result.

However, the value is always taken into consideration in the functional software when the corresponding activation function is switched on. If the data is not read from the CAN due to the node number or the connection type not being assigned, processing continues with the initialisation value 0.

Data is sent every 50 ms, with the receive timeout being set to a value 4 times greater as standard (i.e. 200 ms). If 1 of the values is not received within this period, the value itself is designated as erroneous. The reaction to this depends on the unit and/or the application in question.

6.2.1 Speed

330 *CanSpeedSrcNodeNo*	node number of the sender of the actual speed
331 *CanSpeedSrcNodeType*	node type of the sender of the actual speed
2006 *SpeedCan*	display of the received CAN speed
3003 *ErrPickUpCan*	error display (2006 *SpeedCan* = 0 rpm)
4009 *CanSpeedOn*	speed is retrieved from HZM-CAN
4400 *CanCommDCOn* = 1	or
4405 *CanCommACOn* = 1	release connection type

The redundancy sequence applies even if the speed is received via the CAN bus:

Speed of pickup 1 → Speed of pickup 2 → Camshaft adjuster speed → Speed from generator frequency → CAN speed,

which of course requires the respective measured value to be present and released (otherwise it will be skipped in the redundancy sequence).

6.2.2 Relative measured power

332 *CanPowerSrcNodeNo*	node number of the sender of the relative measured power
333 *CanPowerSrcNodeType*	node type of the sender of the relative measured power
3025 *ErrPowerPercentCan*	error display
3232 *RelativePower*	display of received measured power The last received value is retained in the event of an error
4400 *CanCommDCOn* = 1	or
4405 *CanCommACOn* = 1	release connection type
5232 *CanPowerPercentOn*	relative measured power is retrieved from HZM-CAN

6.2.3 Error status of additional modules

If the error status is expected in the case of certain additional modules, then their node numbers must be announced at minimum. The precise additional module type is not significant for this function.

430 *CanACNodeNumber()*	node numbers of the additional modules
2443 *CanACError()*	hexadecimal display of all error statuses (if recipient displays individual errors)
23006 *ErrCanAC()*	hexadecimal display of all error statuses (if recipient displays error status)
4405 *CanCommACOn* = 1	release connection type
4430 *ReceiveACErrorOn*	error status of all connected additional modules is retrieved by HZM-CAN

The index of 2443 *CanACError()* and 23006 *ErrCanAC()* corresponds to the index occupied by the respective node number in 430 *CanACNodeNumber()*.

The error status is displayed in hexadecimal form and is structured as follows. The bits that are actually used depend on the function of the respective sender.

Bit 8:	Sub-quantity of the fatal error that may be significant to the recipient, currently 3004 *ErrOverSpeed* (if the additional module measures the speed)
Bit 7:	CAN error 3070 *ErrCanBus* and/or 3071 *ErrCanComm* in additional module
Bit 6:	Knocking (ARIADNE only) Heavy knocking if bit 0 is set Light knocking if bit 0 is reset
Bit 4:	System error (ELEKTRA only)
Bit 1:	General error in additional module Corresponds to 3801 *CommonAlarm*
Bit 0:	Fatal error in additional module, corresponds to 3800 *EmergencyAlarm*

6.2.4 Status of additional modules or speed governor modules

If the status of certain AC or DC modules is expected, then their node numbers and their connection type must be announced.

402 *CanDCNodeNumber*	node number of the DC module
430 *CanACNodeNumber()*	node numbers of the additional modules
2540 *CanDCStatus*	hexadecimal display of the DC module status
2541 *CanACStatus()*	hexadecimal display of all additional module statuses
4400 *CanCommDCOn* = 1	release DC connection type and/or
4405 *CanCommACOn* = 1	release AC connection type
4431 *ReceiveStatusOn*	status of all connected AC and DC modules is retrieved from HZM-CAN

The index of 2540 *CanACStatus()* corresponds to the index assigned by the respective node number in 430 *CanACNodeNumber()*. The precise additional module type is not significant for this function.

The status is displayed in hexadecimal form and is structured as follows. The bits that are actually used depend on the function of the respective sender.

Bit 2:	An (engine) stop request is active in the sender
Bit 1:	Gas is being used in dual-fuel mode Reception only from duel-fuel speed governors (DC) or gas positioners (AC)
Bit 0:	Dual-fuel mode is active in the sender Reception only from duel-fuel speed governors (DC) or gas positioners (AC)

6.2.5 Setpoint speed of speed governor modules

336 *CanSpSetpSrcNodeNo*	node number for sender of setpoint speed
337 *CanSpSetpSrcNodeType*	node type for sender of setpoint speed
2031 *SpeedSetp*	display of received setpoint speed
3041 *ErrSpeedSetpCan*	error display (2031 *SpeedSetp* = 0 rpm)
4031 *CanSpeedSetpOn*	setpoint speed is retrieved from HZM-CAN
4400 *CanCommDCOn* = 1	release connection type

6.2.6 Boost pressure

The reception of the boost pressure sensor is treated as if this sensor were performing reception via an analogue input or another communication module. If reception should be performed via the multipoint connection, then the sensor channel type HZM_CAN_ALL must be stated.

338 *CanBoostSrcNodeNo*	node number for sender of boost pressure
339 *CanBoostSrcNodeType*	node type for sender of boost pressure
904 *AssignIn_BoostPress* = 1	sensor activation (0 = deactivated)
2904 *BoostPressure*	display of received boost pressure
3009 *ErrBoostPressure*	error display
4400 *CanCommDCOn* = 1	or
4405 *CanCommACOn* = 1	release connection type
4904 *ChanTypBoostPress* = 14	sensor channel type HZM_CAN_ALL

6.2.7 Exhaust temperature

The reception of the exhaust temperature sensor is treated as if this sensor were performing reception via an analogue input or another communication module. If reception should be performed via the multipoint connection, then the sensor channel type HZM_CAN_ALL must be stated.

342 *CanExhTempSrcNodeNo*	node number for sender of exhaust temperature
343 *CanExhTmpSrcNodeType*	node type for sender of exhaust temperature
911 *AssignIn_ExhaustTemp* = 1	sensor activation (0 = deactivated)
2911 *ExhaustTemp*	display of received exhaust temperature
3016 *ErrExhaustTemp*	error display
4400 *CanCommDCOn* = 1	or
4405 *CanCommACOn* = 1	release connection type
4911 *ChanTypExhaustTemp* = 14	sensor channel type HZM_CAN_ALL

6.2.8 Filling/injection quantity/actuator position

The reception of the filling/injection quantity/actuator position of a speed governor or a gas positioner is treated as if this sensor were performing reception via an analogue input or another communication module. If reception should be performed via the multipoint connection, then the sensor channel type HZM_CAN_ALL must be stated.

344 *CanFuelSetpSrcNodeNo*	node number for sender of filling
345 *CanFuelSpSrcNodeType*	node type for sender of filling
911 *AssignIn_(Gas/Diesel)FuelSetp* = 1	
	sensor activation (0 = deactivated)
2911 *(Gas/Diesel)FuelSetp*	display of received filling value
3016 *Err(Gas/Diesel)FuelSetp*	error display
4400 *CanCommDCOn* = 1	or
4405 *CanCommACOn* = 1	release connection type
4911 *ChanTyp(Gas/Diesel)FuelSetp* = 14	
	sensor channel type HZM_CAN_ALL

6.2.9 AutoReset and ErrorReset

The AutoReset and ErrorReset commands do not need to be released separately. They are always received and the connection to all modules with 4415 *CanCommAllOn* = 1 is released in the recipient. If at least 1 module connects 1 of these commands to the bus, then the command is performed by all recipients of ALL messages. The error deletion process and the boost process are performed simultaneously.

7 Monitoring

7.1 Configuration errors

As previously stated in ↑2 *General*, modified parameters must be stored and a reset must subsequently be performed. After booting, a check is run to establish whether the CAN configuration is correct. If there is a configuration error, this is displayed in 3092 *ErrConfiguration* and 3000 *ConfigurationError* will also feature information on the erroneous configuration.

The following tables list the significance of a configuration error according to the node type of the control unit. These errors are also the subject of a verbal indication in the error window in DcDesk 2000.

No.	Meaning
903	Sensor channel number too large for PE sensor
912	A sensor should be received by a PE node that is not configured or does not feature any sensors
1100	DualFuel: PE node number for gas setpoint not defined or incorrect node type
1101	DualFuel: PE node number for gas setpoint defined, but own hardware activated for this
1102	DualFuel: PE node number for gas setpoint defined, but setpoint telegram not activated
11001	Periphery module type not supported by master
11002	Number of possible nodes for a periphery module type exceeded
11003	Duplicate assignment of node number for periphery module
11004	No node number defined for periphery module, although function requested

Tab. 11: Configuration errors: PE master

No.	Meaning
11000	No master activated for periphery module
11005	Multiple masters defined for periphery module

Tab. 12: Configuration errors: PE module

No.	Meaning
12001	Additional module type not supported
12002	Number of possible nodes for an additional module type exceeded
12003	Duplicate assignment of node number for additional module

Tab. 13: Configuration errors: AC destination

No.	Meaning
12000	No destination activated for additional module
11005	Multiple destinations defined for additional module

Tab. 14: Configuration errors: AC module

No.	Meaning
859	A switching input should be received by a CM node that is not configured or does not feature any binary inputs
909	CM sensor input too large
21950	CM connection not activated, but values requested from this

Tab. 15: Configuration errors: CM destination

No.	Meaning
916	Sensor reception via HZM_CAN_ALL not supported
917	Node number of sensor sender not assigned
12100	Reception of DC node type not supported
12101	Reception of GC node type not supported
12102	Reception of PE node type not supported
12105	Reception of AC node type not supported
12106	Reception of CM node type not supported
12107	Unknown sender node type

Tab. 16: Configuration errors: ALL reception

7.2 Monitoring CAN communication

Communication is constantly monitored. Once the control unit is switched on, it may take a certain amount of time before an error message is triggered. This time delay is entered in the parameter 400 *CanStartTimeOutDelay*. All participants on the CAN network should undergo parameterisation with the same time delay. Within this period, the entire network must be supplied with voltage and all units must complete their boot procedure in order to prevent an error message being triggered upon switching on.

The following general error messages are generated:

3070 *ErrCanBus1*/3072 *ErrCanBus2*	error in CAN bus 1 or 2
3071 *ErrCanComm1*/3073 *ErrCanComm2*	error in CAN communication 1 or 2

In the event of a CAN bus error, the CAN controller issues errors such as bus status, error status or data overrun. Despite reinitialisation of the controller, it is not possible to permanently eliminate the error. The cause of this is normally incorrect cabling, missing termination or differing baud rates in the individual network participants. The control unit performs ongoing initialisation of the CAN controller in attempting to achieve an error-free connection status.

On the other hand, the CAN communication error is a content-related network error. This means that there is no physical error present and communication is possible in principle. The following parameters provide more information on the communication error:

2401 *CanTxBufferState*	status of send buffer
2402 *CanRxBufferState*	status of receive buffer
2403 *CanRxTimeout*	status of receive timeout monitoring
2404 *CanTypeMismatch*	status of unit numbers

These values feature binary encoding, whereby the bit number corresponds to the unit type in ↑*Tab. 3:* Node types.

If an entire error status corresponds to each error number in the control unit (all MVCs, ARIADNE, PHLOX, XIOS), then the occurrence of this error is also displayed in 3071 *ErrCanComm1*/3073 *ErrCanComm2*:

Bit 3: bCanTypeMismatch	status of unit numbers
Bit 2: bCanTxBufOverflow	status of send buffer
Bit 1: bCanRxBufOverflow	status of receive buffer
Bit 0: bCanRxTimeout	status of receive timeout monitoring

Send and receive buffers are monitored for exceedance for each unit type, with the parameters

2401 *CanTxBufferState* and 2402 *CanRxBufferState*

displaying this information. A 1 at the corresponding bit indicates that the procedure of sending to a specific recipient node type (regardless of the node number) could not be transferred and/or that the receive buffer for telegrams has been exceeded by a particular sender node type.

The reception of messages must take place within a certain time frame, otherwise the error 2403 *CanRxTimeout* is set for the respective connection.

2404 *CanTypeMismatch* indicates an error if a second participant featuring the same unit number and the same unit type is connected to the network.

If the send or receive buffer is exceeded, this is only displayed: communication continues, but of course 1 or more messages will not have been sent or read. If too many messages are not received, the error 2403 *CanRxTimeout* is set. If the messages cannot be transferred in the event that the send buffer is exceeded, the destination (recipient) displays the timeout error.

The error 2403 *CanRxTimeout* is generally set if the destination does not report. In this case, messages will continue to be sent to the destination but there will be a shift to certain emergency procedures.

The parameters 2405 *Can1Online* / 2407 *Can2Online* display information regarding whether the control unit is generally ready to communicate via CAN.

8 Parameterisation

Example for an HZM-CAN bus system

In a theoretical generator system, the following HEINZMANN units should be connected to one another and communicate with one another:

Generator 1: Diesel V-engine (HELENOS, DC 2 PE, THESEUS)

Generator 2: Dual-fuel engine (PRIAMOS, MVC01 PE, THESEUS, ARIADNE)

Generator 3: Gas V-engine (PANDAROS, 2 * DC 10 PE, THESEUS, ARIADNE)

The example only covers the configuration of the nodes and the connection between them. It does not cover the specific data that is transferred. The physical structure was previously described in ↑ *2.2 Example for an HZM-CAN bus system*.

8.1 Node types and node numbers

Once the physical structure has been defined, the logical connections must then be defined. This requires a unique identifier to be issued to each unit, ↑3 *Definition of nodes.* Whereas the node type is specified by the control unit type, the node number must be parameterised.

Control unit		Node definition		
		Type	Sub-type	No.
Speed governors	HELENOS	DC		1
	PRIAMOS	DC		2
	PANDAROS	DC		3
Generator control units	THESEUS 1	GC		1
	THESEUS 2	GC		2
	THESEUS 3	GC		3
Periphery modules	DC2PE	PE	0	1
	MVC01OE	PE	6	2
	DC10PE 1	PE	14	3
	DC10PE 2	PE	14	4
Additional modules	ARIADNE 1	AC	0	2
	ARIADNE 2	AC	0	3

Tab. 17: Example: node definition

8.2 Logical connections

The speed governors and their respective THESEUS units transfer the necessary values via point-to-point connections that feature fixed definitions of content.

A parameterisable point-to-point connection is established between the speed governors and their periphery modules in each case, ↑5 *Master and periphery module*.

The THESEUS units exchange their information with one another via a specifically defined multipoint connection for generator control units.

Both ARIADNE units save their data to the bus via a parameterisable multipoint connection, ↑6 *Multipoint connection*. The speed governors of the dual-fuel engine and the gas engine can therefore react to any knocking in gas operation.

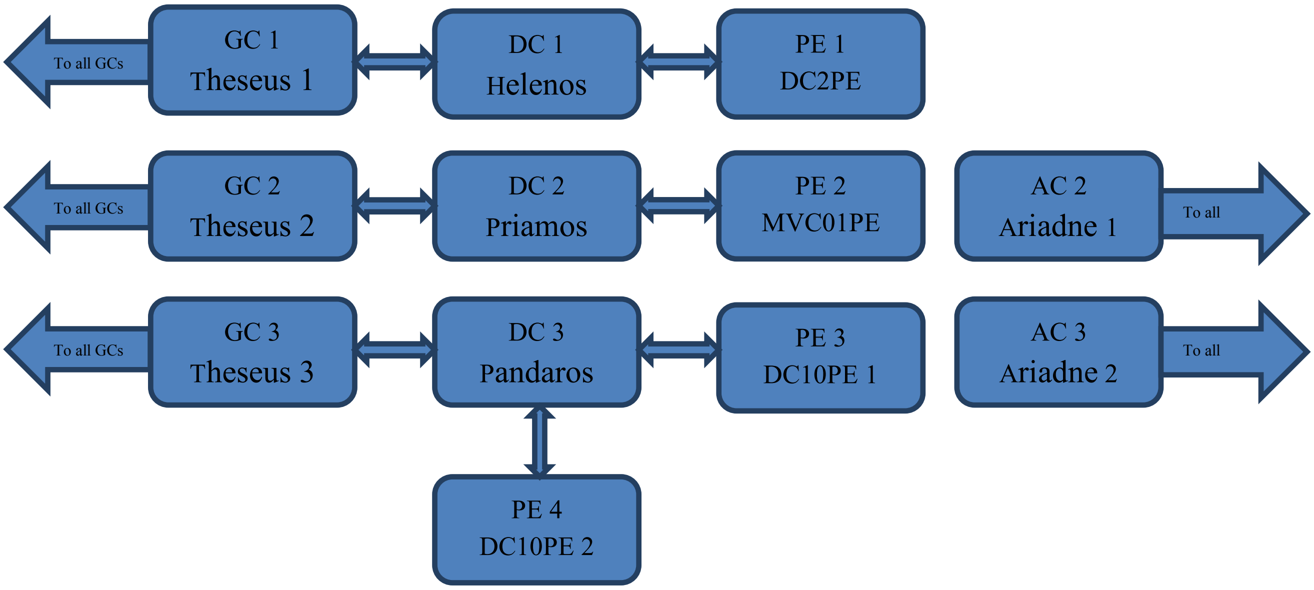


Fig. 5: Example of logical connection

8.3 Parameterisation

The following parameterisation is performed when determining the node numbers and the point-to-point or multipoint connections being used:

HELENOS:

401 *CanMyNodeNumber* = 1
404 *CanPENodeNumber*[0] = 1
405 *CanPENodeNumber*[1] = 406 *CanPENodeNumber*[2] = 0
407 *CanPENodeType*[0] = 0
416 *CanBaudrate* = 500
430 *CanACNodeNumber*[0] …434 *CanACNodeNumber*[4] = 0
4401 *CanCommGCOn* = 1
4402 *CanCommPEOn* = 1

PRIAMOS:

401 *CanMyNodeNumber* = 2
404 *CanPENodeNumber*[0] = 2
405 *CanPENodeNumber*[1] = 406 *CanPENodeNumber*[2] = 0
407 *CanPENodeType*[0] = 6
416 *CanBaudrate* = 500
430 *CanACNodeNumber*[0] = 2
431 *CanACNodeNumber*[1] …434 *CanACNodeNumber*[4] = 0
435 *CanACNodeType*[0] = 0
4401 *CanCommGCOn* = 1
4402 *CanCommPEOn* = 1
4405 *CanCommACOn* = 1
4415 *CanCommAllOn* = 1
4430 *ReceiveACErrorOn* = 1

PANDAROS:

401 *CanMyNodeNumber* = 3
404 *CanPENodeNumber*[0] = 3
405 *CanPENodeNumber*[1] = 4
407 *CanPENodeType*[0] = 14
408 *CanPENodeType*[1] = 14
416 *CanBaudrate* = 500
430 *CanACNodeNumber*[0] = 3
431 *CanACNodeNumber*[1] …434 *CanACNodeNumber*[4] = 0
435 *CanACNodeType*[0] = 0
4401 *CanCommGCOn* = 1
4402 *CanCommPEOn* = 1
4405 *CanCommACOn* = 1
4415 *CanCommAllOn* = 1
4430 *ReceiveACErrorOn* = 1

THESEUS 1:
401 *CanMyNodeNumber* = 1
416 *Can1Baudrate* = 500
4400 *CanCommDCOn* = 1
4401 *CanCommGCOn* = 1

THESEUS 2:
401 *CanMyNodeNumber* = 2
416 *Can1Baudrate* = 500
4400 *CanCommDCOn* = 1
4401 *CanCommGCOn* = 1

THESEUS 3:
401 *CanMyNodeNumber* = 3
416 *Can1Baudrate* = 500
4400 *CanCommDCOn* = 1
4401 *CanCommGCOn* = 1

DC2PE:
401 *CanMyNodeNumber* = 1
403 *CanMasterNodeNumber* = 1
416 *CanBaudrate* = 500
4400 *CanCommDCOn* = 1

MVC01PE:
401 *CanMyNodeNumber* = 2
403 *CanMasterNodeNumber* = 2
416 *CanBaudrate* = 500
4400 *CanCommDCOn* = 1

DC10PE 1:
401 *CanMyNodeNumber* = 3
403 *CanMasterNodeNumber* = 3
416 *CanBaudrate* = 500
4400 *CanCommDCOn* = 1

DC10PE 2:
401 *CanMyNodeNumber* = 4
403 *CanMasterNodeNumber* = 3
416 *CanBaudrate* = 500
4400 *CanCommDCOn* = 1

ARIADNE 1:
401 *CanMyNodeNumber* = 2
416 *Can1Baudrate* = 500
4334 *AllSendErrorStateOn* = 1
4415 *CanCommAllOn* = 1

ARIADNE 2:
401 *CanMyNodeNumber* = 3
416 *Can1Baudrate* = 500
4334 *AllSendErrorStateOn* = 1
4415 *CanCommAllOn* = 1

9 Terminating resistors

The following sections feature information on each control unit, stating whether they are supplied with or without a terminating resistor as standard and where this should be positioned or removed. Control units that use different items of software or use hardware modifications to represent various unit types in the CAN bus are listed only once (e.g. DC 6 and DC 6_200 as speed governor, DC 6 PE as periphery module, DC 6 as (gas) positioner).

9.1 DC 1

All DC1 variants feature a CAN controller on the CAN additional board (without galvanic isolation).

Configuration of CAN termination

Standard configuration: Termination activated

Otherwise: CAN termination according to customer order.

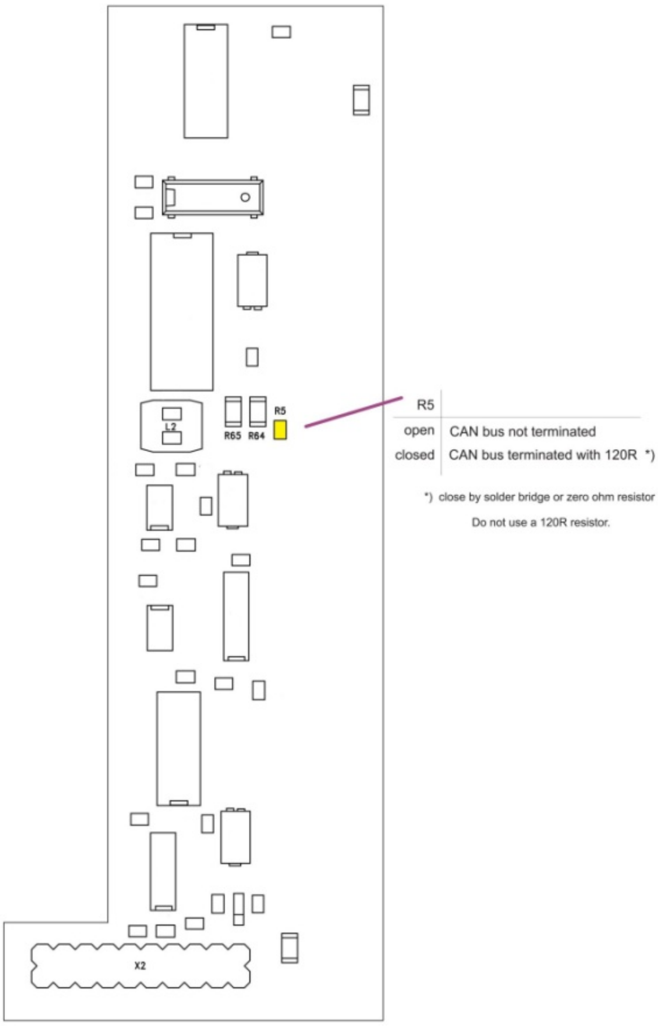


Fig. 6: DC1, position of terminating resistor

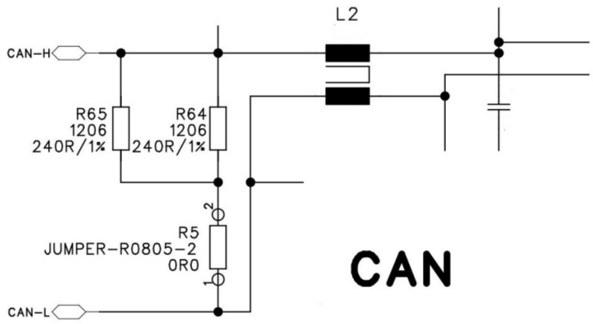


Fig. 7: DC 1, circuit

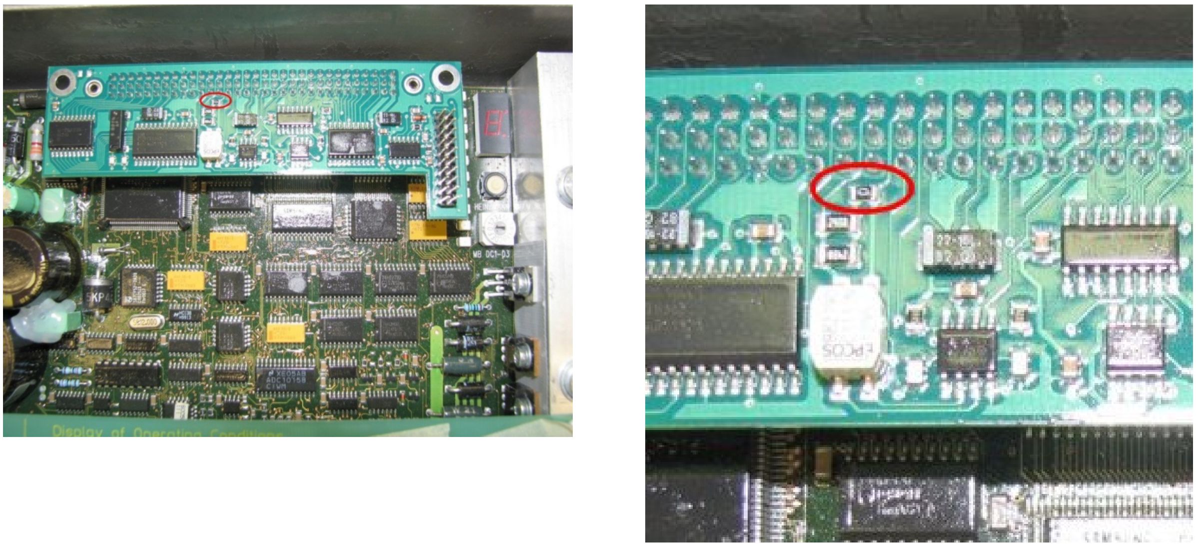


Fig. 8: DC 1-03, position of terminating resistor (photo)

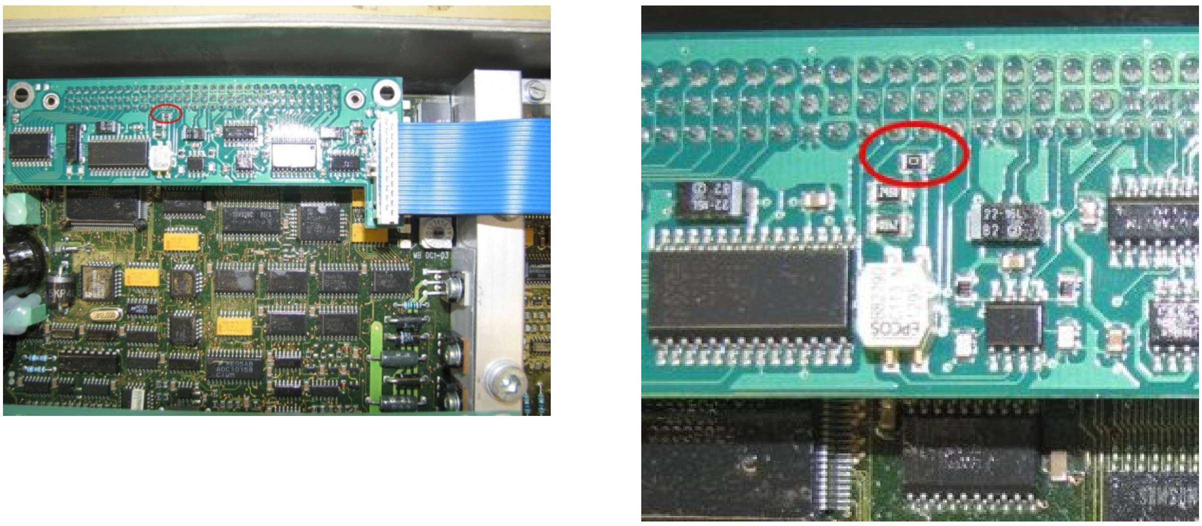


Fig. 9: DC 1-04, position of terminating resistor (photo)

9.2 DC 2

HELENOS features a CAN controller on an additional board. Users must distinguish between the different development statuses of the control unit and the CAN printed circuit boards:

9.2.1 DC2-01 with small CAN printed circuit board without galvanic isolation

The DC2-01 is already hard-wired with 220 Ω.

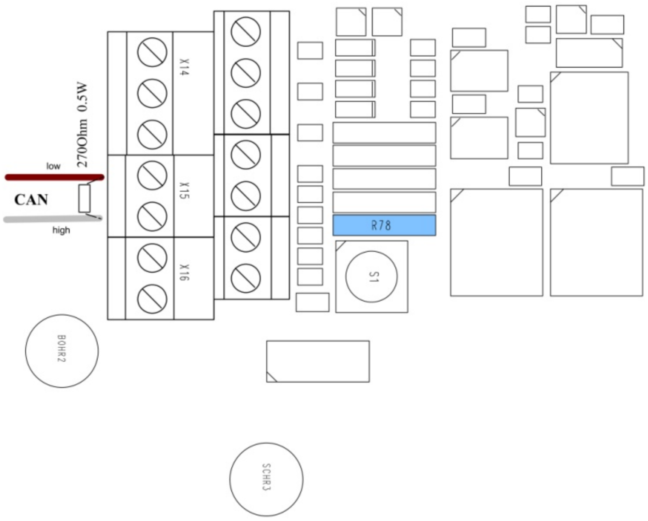


Fig. 10: DC 2-01, hard-wired with 220 Ω

The control unit is a bus participant and an end unit (in a topological sense). The 120 Ω termination is implemented through parallel connection of 270 Ω/0.5W and 220 Ω internally.

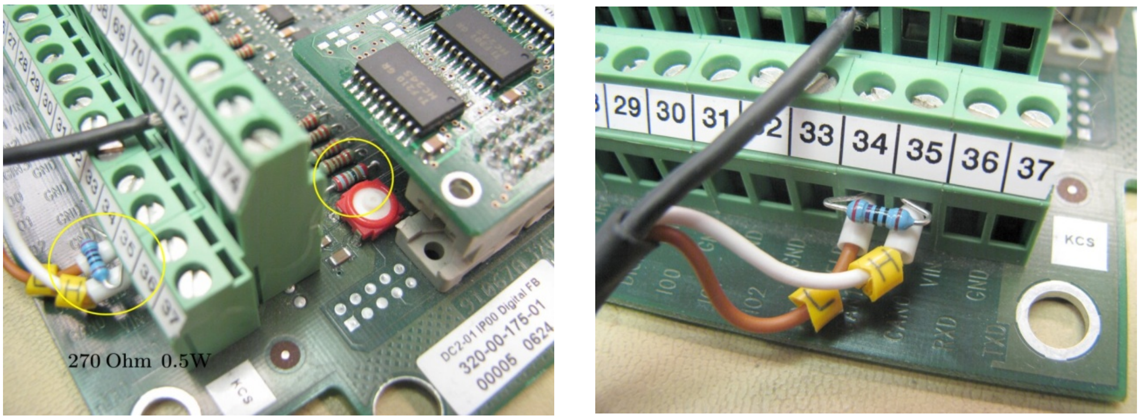


Fig. 11: DC2-01 with small CAN printed circuit board, termination

The control unit is a bus participant, but not an end unit (in a topological sense). No termination, internal resistance removed.

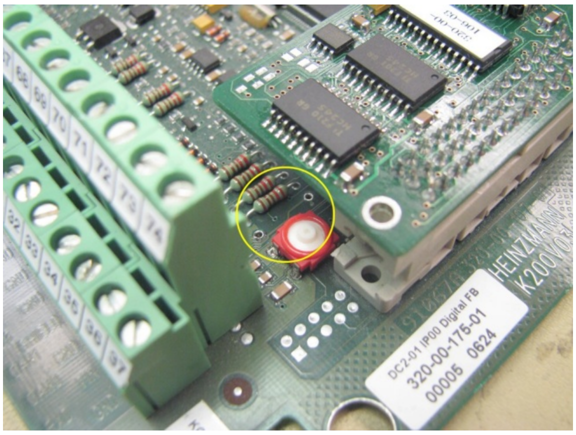


Fig. 12: DC2-01 with small CAN printed circuit board, no termination

As an alternative to both of these solutions, leave the internal 220 Ω resistance in place. Depending on the bus length and the baud rate, 220 Ω termination at the end point may be sufficient. In the case of applications that do not require any termination, the bus drivers of the other bus participants can still absorb the additional parallel connection of the 220 Ω.

9.2.2 DC2-02 with small CAN printed circuit board without galvanic isolation

Circuit context:

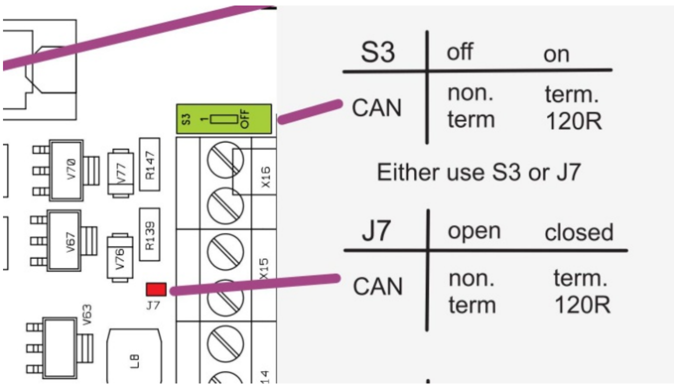


Fig. 13: DC2-01 with small CAN printed circuit board without galvanic termination

In the DC2-02, the CAN bus termination can be activated using a switch (default) or a soldering jumper. The soldering jumper is an additional option that has not previously been used. The termination is active when the switch and/or the soldering jumper are connected.

The control unit is a bus participant and an end unit (in a topological sense):

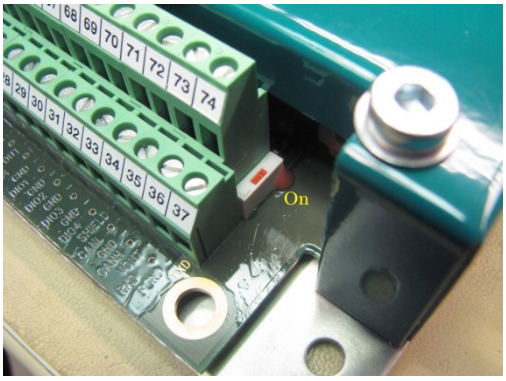


Fig. 14: DC 2 with HCX73.01, termination

The control unit is a bus participant and NOT an end unit (in a topological sense):

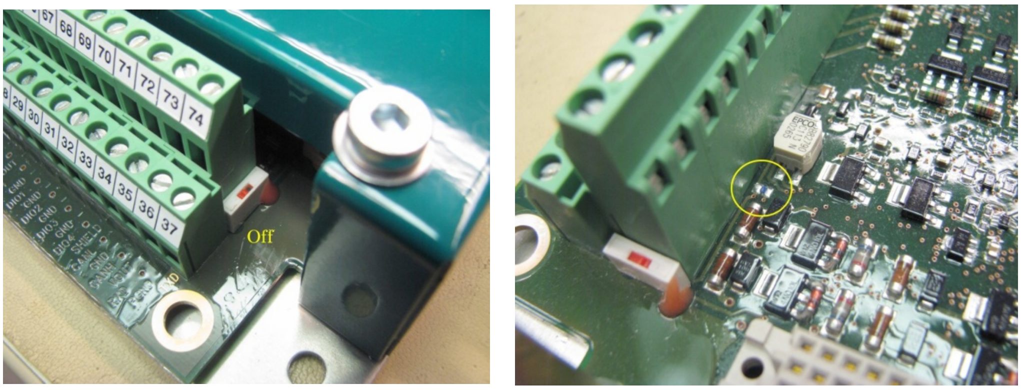


Fig. 15: DC 2 with HCX73.01, no termination (left), position of soldering jumper 'J7' (right)

9.2.3 DC2 with HCX73.01 with galvanic isolation and termination

The control unit is the bus participant located at the end of the bus. Termination: 120 Ω/0.5W on the CAN module. All photos feature DC2-01 as an example, but they also apply for the DC2-02.

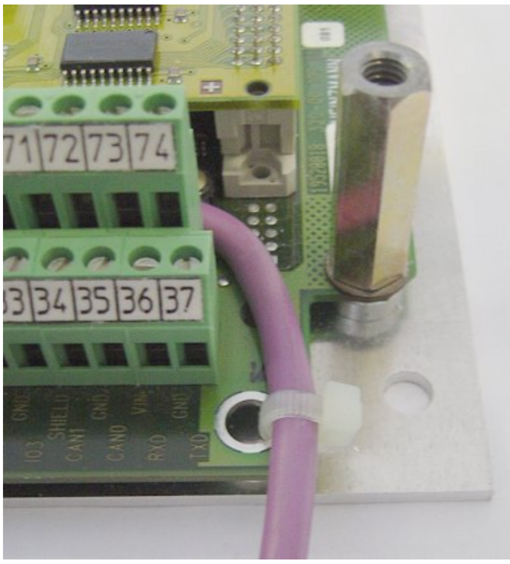


Fig. 16: DC 2, cable infeed

Fig. 17: DC 2 with HCX73.01, termination

The module provides 2 CAN bus participants that are connected in parallel. The right circuit shows the open connections to the main printed circuit board (module variant 320-00-177-03 – the label is incorrect). The 220 Ω resistance on the main printed circuit board DC2-01 does not have any influence.

9.2.4 DC2 with HCX73.01 with galvanic isolation and without termination

The control unit is a bus participant, but not an end unit (in a topological sense). This means that there is no termination. All photos feature DC2-01 as an example, but they also apply for the DC2-02.

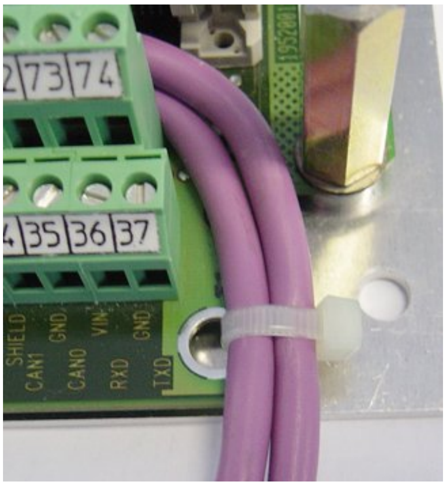


Fig. 18: DC 2: infeed of bus cable into control unit

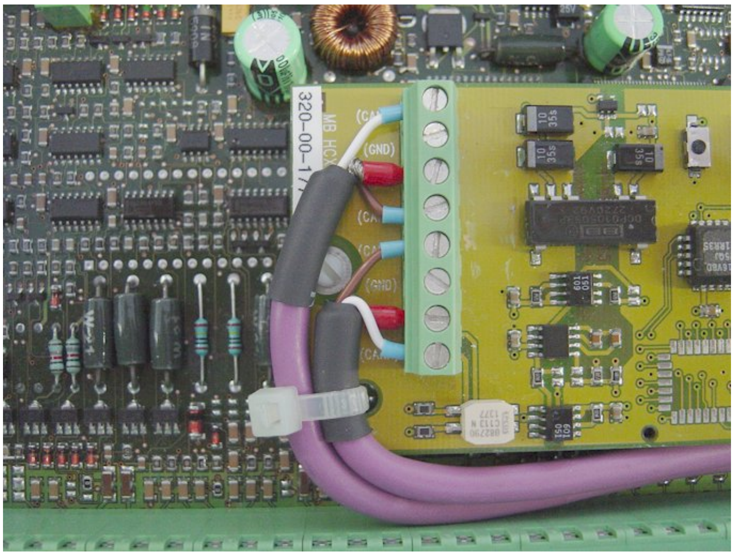


Fig. 19: DC 2, connection of cable to the HCX73.01 module

With regard to the photo, please ignore the colour of the wires. This detail is depicted incorrectly: in the line shown above, brown and white have been mixed up.

9.3 DC 5

ARCHIMEDES has 2 CAN controllers.

The resistances R1 (for CAN0) and R290 (for CAN1) are unequipped as standard. If the unit in the respective CAN bus is located at the end of the line, it can be equipped with 120 Ω. The HZM-CAN is operated on CAN 0 as standard. However, CAN 1 can also be used in certain applications. This is because 1 CAN is located on each of the engine plug and the vehicle plug, and the task dictates the side on which the HZM-CAN is implemented.

Pin assignment DC5 (320-00-167-01)

X1: Connector vehicle side **X2: Connector engine side**

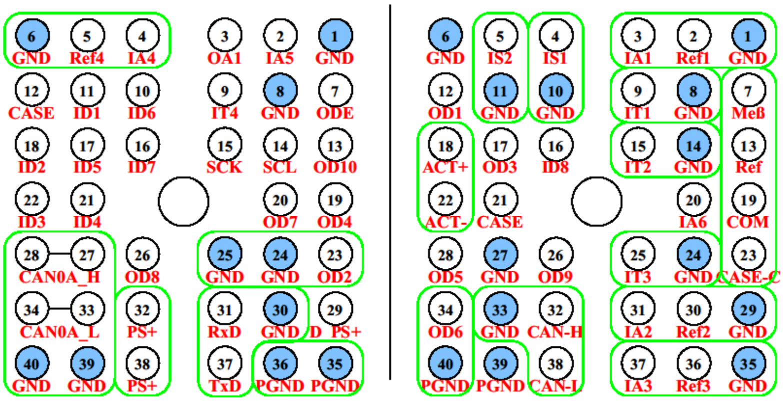


Fig. 20: DC 5, pin assignment

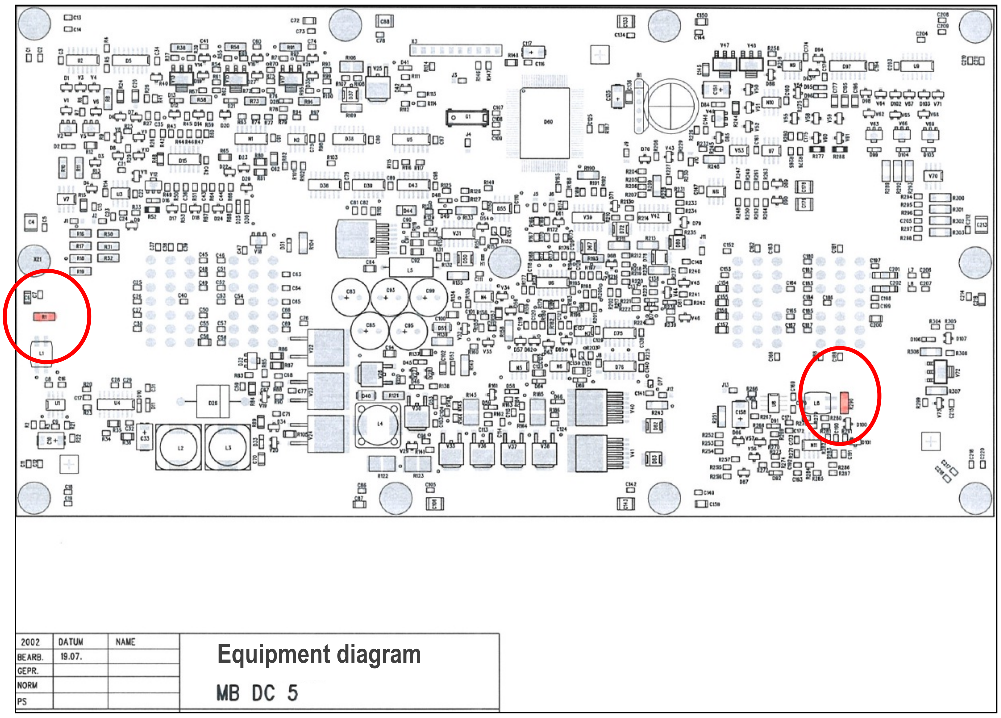


Fig. 21: DC 5, printed circuit board with position of terminating resistor

9.4 DC 6, ELEKTRA, KRONOS 20, KASSANDRA, GMA, E-LES

These units feature a CAN controller on the printed circuit board. These units are supplied without a terminating resistor as standard. When equipping R217 or R212 (both at the same position) with 0 Ω, a 120 Ω terminating resistor is activated.

ELEKTRA also features a second CAN on the plug-in board,

↑*9.5 ELEKTRA plug-in board*.

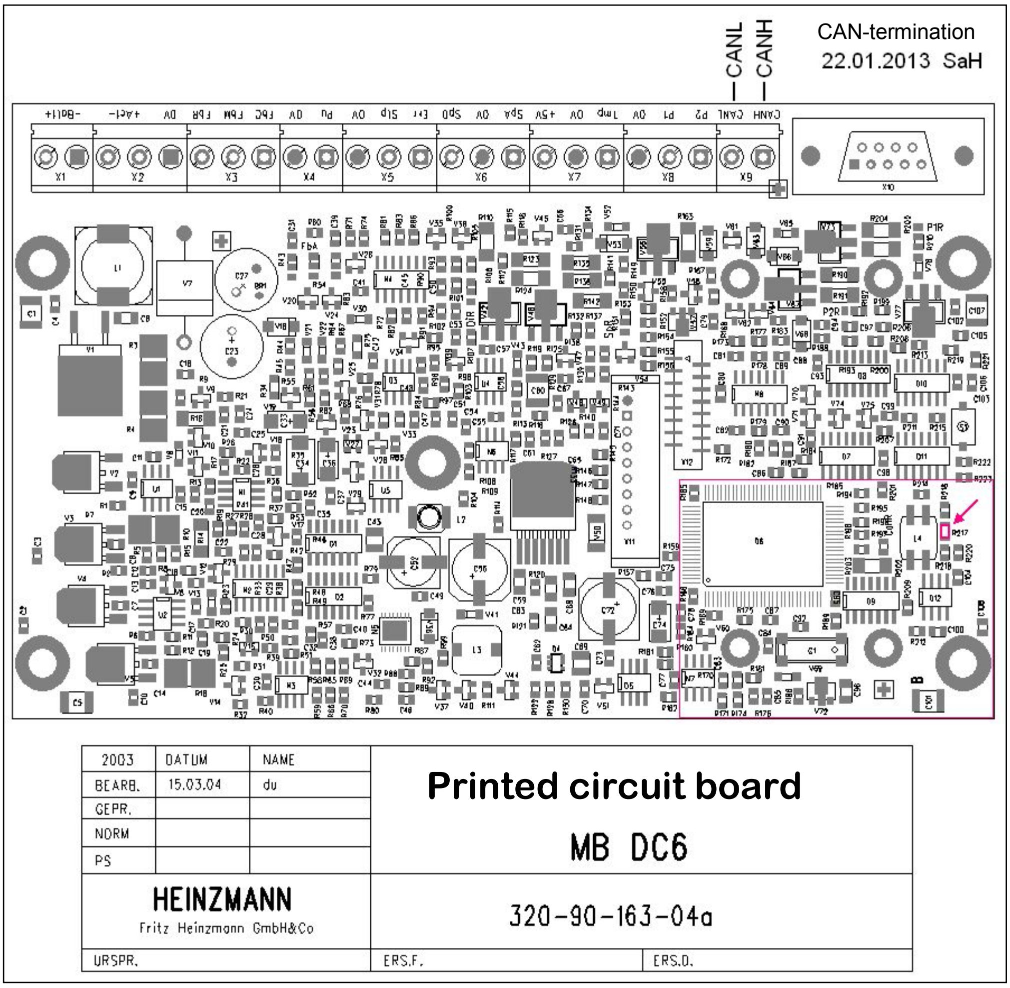


Fig. 22: DC 6, printed circuit board with position of terminating resistor R217

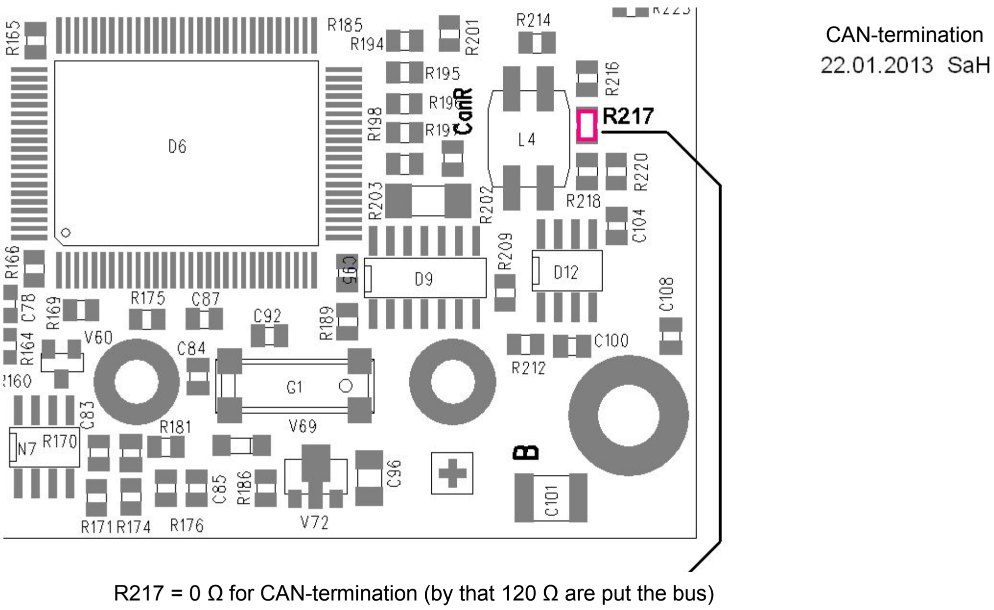


Fig. 23: DC 6, printed circuit board with position of terminating resistor R217 (zoom)

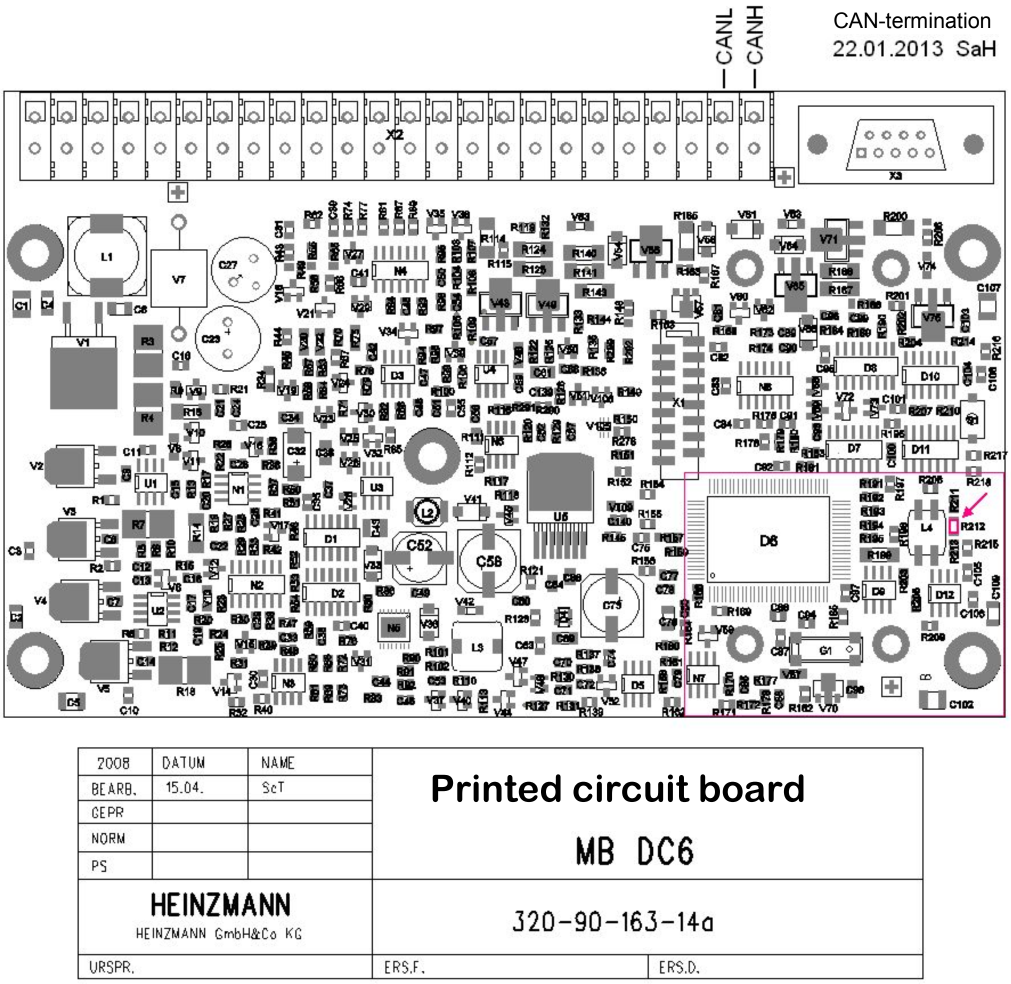


Fig. 24: DC 6, printed circuit board with position of terminating resistor R212

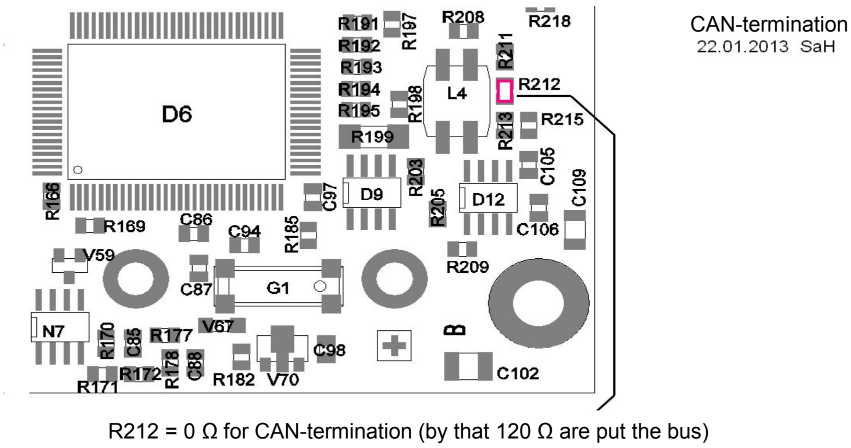


Fig. 25: DC 6, printed circuit board with position of terminating resistor R212 (zoom)

9.5 ELEKTRA plug-in board

ELEKTRA features 2 CAN controllers. CAN 1 is located on the motherboard, ↑*9.4 DC 6, ELEKTRA, KRONOS 20, KASSANDRA, GMA, E-LES*.

CAN 2 is located on the plug-in board. R32 must be equipped with 0 Ω for the purpose of termination. The resistance is also removed as standard, meaning that CAN 2 is not terminated.

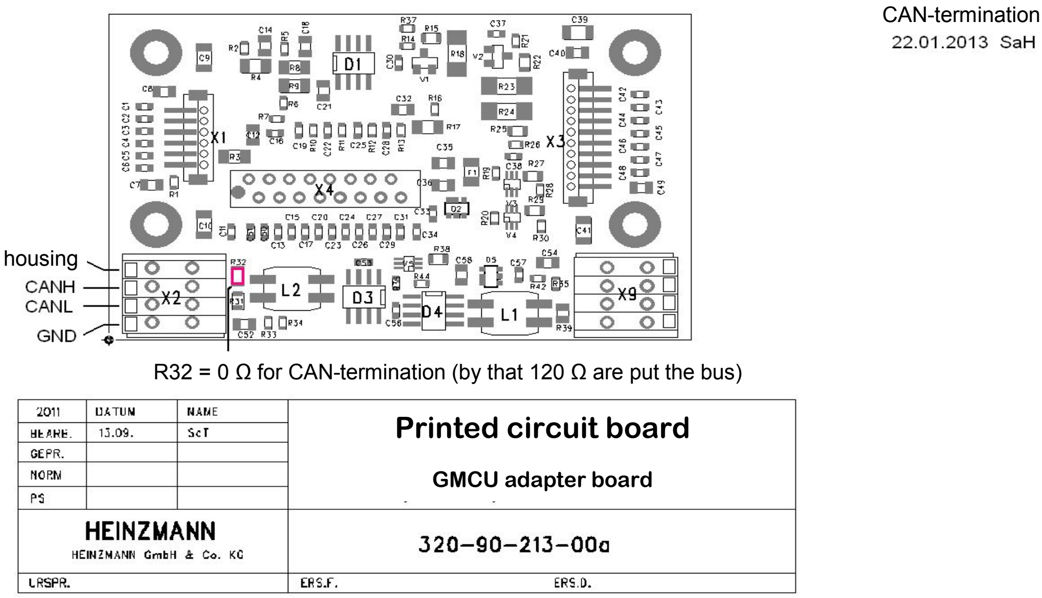


Fig. 26: ELEKTRA plug-in board, printed circuit board with position of terminating resistor

9.6 TWIN-PANDAROS

TWIN-PANDAROS is fitted with 2 DC 6 control units. These are connected to one another via the HZM-CAN. One unit is the speed governor, with the other being a periphery module for controlling the second actuator on a V-engine.

The terminating resistor can be switched on or off separately for each unit using a switch on the internal plug-in board. If the CAN is not laid out, then both terminating resistors must be set. If the CAN is laid out in order to connect additional units to the HZM-CAN bus, then this depends on whether or not the TWIN-PANDAROS is positioned at the end of the bus. In the first instance, only 1 of the 2 units may be terminated with 120 Ω. In the second, neither unit may be terminated. Caution: if the unit is not positioned at the end of the bus, the cable set must be modified such that the CAN line can go in and come out again. Short stub lines to the unit are possible according to the specification ↑1.3 Stub lines.

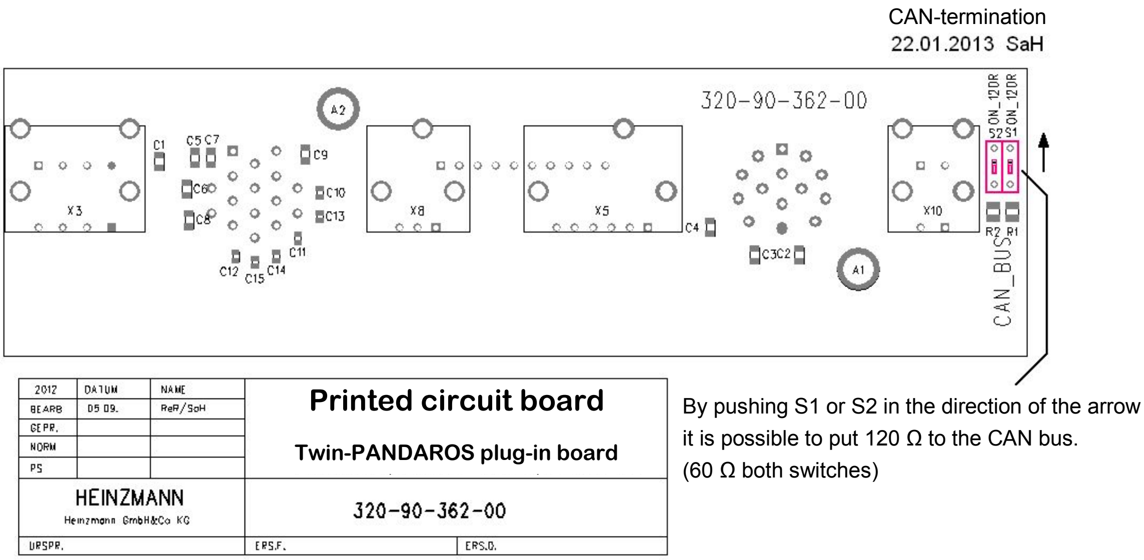


Fig. 27: TWIN-PANDAROS, printed circuit board with position of the terminating resistors

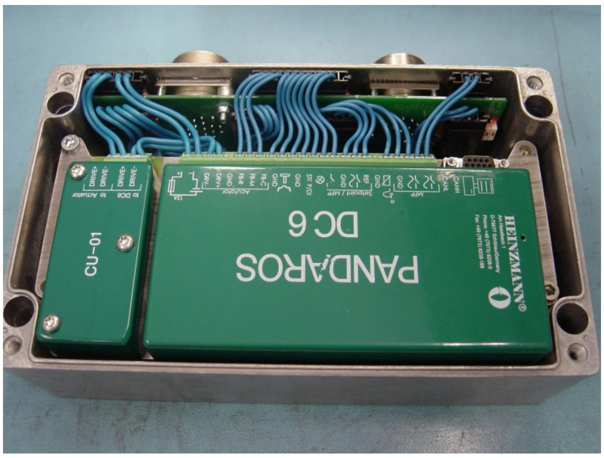


Fig. 28: TWIN-PANDAROS, printed circuit board with position of the terminating resistors (photo)

9.7 DC 7

The DC 7 features a CAN controller that is laid out on the vehicle plug. The terminating resistor is not present in the control unit. It must be attached to the CAN line on the exterior. The CAN is positioned on the vehicle plug, pin 12 CAN-H, pin 13 CAN-L.

Vehicle

Assignment	Pin	Pin	Assignment
CAN-L	13	25	Analogue input 1 - REF
CAN-H	12	24	Analogue input 1 - IN
ISO 9141-K	11	23	Analogue input 1 - GND
ISO 9141-L	10	22	Housing
SAE1708-K (or -L)	9	21	Digital input 2
SAE1708-L (or -K)	8	20	Analogue input 3 - IN
Speed 1	7	19	Digital input 4
Digital input 3	6	18	Digital input 1
Digital output 1	5	17	GND
Digital output, error lamp	4	16	Digital output 7 (speed)
Digital output 2	3	15	Digital output 5
GND	2	14	Battery positive
Battery negative (GND)	1		

Tab. 18: DC 7, vehicle pin assignment

Engine

Assignment	Pin	Pin	Assignment
Speed input 1	13	25	Analogue input 2 - REF
Speed input 1 (GND)	12	24	Analogue input 2 - IN
Speed input 2	11	23	Analogue input 2 - GND
Speed input 2 (GND)	10	22	Analogue input 4 - REF
Temperature input 3 (NTC)	9	21	Analogue input 4 - IN
Temperature input (GND)	8	20	Analogue input 4 - GND
Digital output 7 (PWR)	7	19	Control path input - measurement coil
Speed input 5	6	18	Control path input - reference coil
Temperature input 2 (NTC)	5	17	Control path input - comm.
Temperature input 1 (NTC)	4	16	Housing
Digital output 4	3	15	Control solenoid output (+)
Digital output 3	2	14	Control solenoid output (-)
	1		

Tab. 19: DC 7, engine pin assignment

9.8 DC 8

The DC 8 features 2 CAN controllers, of which only CAN 1 is currently laid out.

In order to connect a terminating resistor for CAN 1, it is necessary to connect 120 Ω to the plugs specified below. This resistance is not equipped as standard.

CAN 2 is connected via R198 = 0 Ω. This resistance is equipped as standard, as the CAN 2 is used for internal communication with the servo motor. If another actuator is connected, then this CAN can also be used in a different way (submit query).

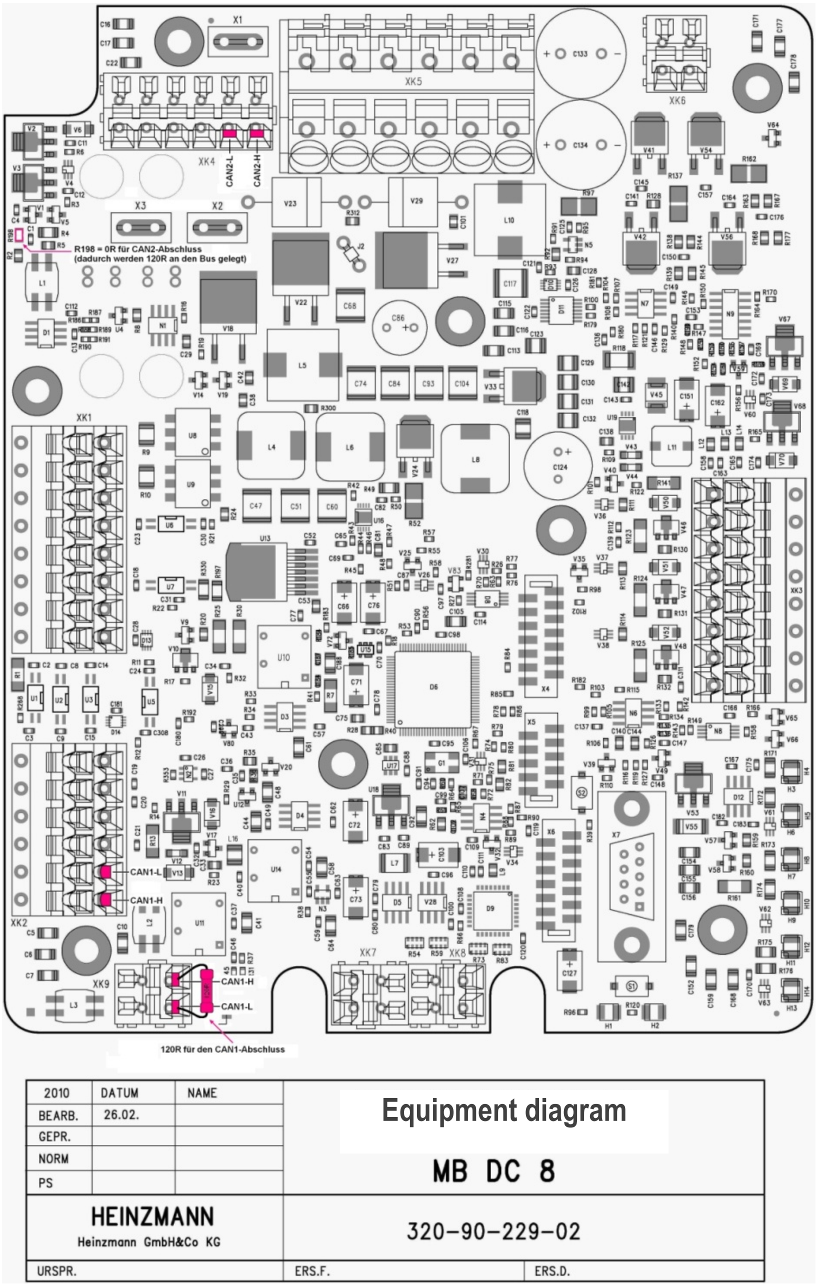


Fig. 29: DC 8, printed circuit board with position of terminating resistors

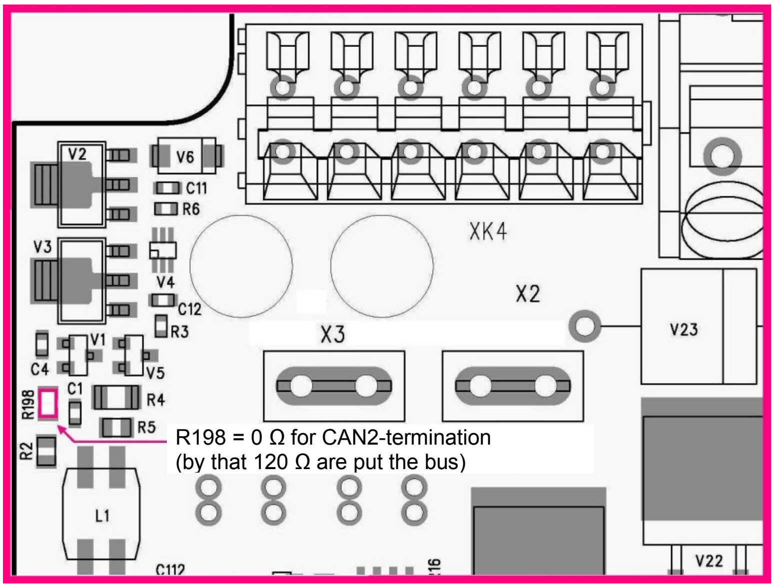


Fig. 30: DC 8, printed circuit board with position of terminating resistor CAN 2 (zoom)

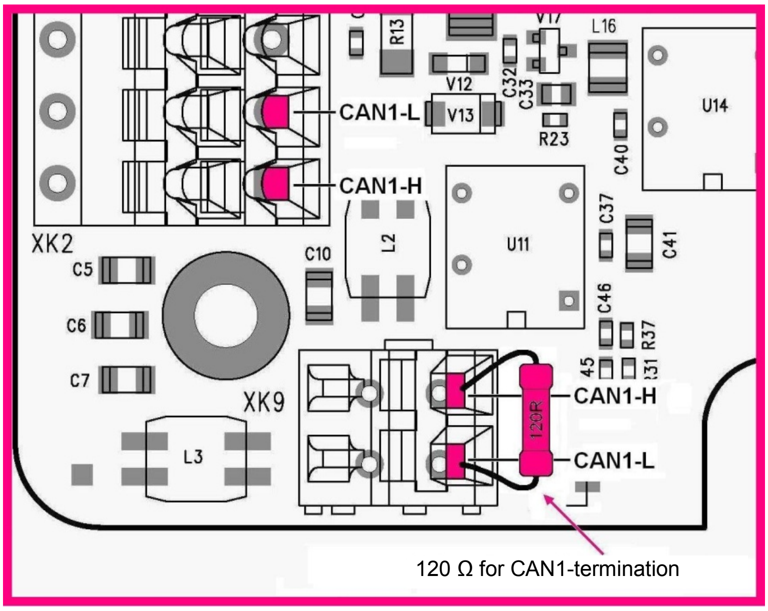


Fig. 31: DC 8, printed circuit board with position of terminating resistor CAN 1 (zoom)

9.9 DC 9

The Orion does not feature a CAN.

9.10 DC 10

The DC 10 is supplied in a number of different hardware variants – the CAN is activated in some of these, but this is not the case in some others. As almost all feature identical software in which the CAN parameters are also visible in the event that the CAN bus cannot be used for hardware reasons, you should use the parameter 3834 *CanBusAvailable* = 1 to check whether a CAN controller is present.

The terminating resistor is not present in the control unit. It must therefore be attached to the CAN line on the exterior.

Pin assignment, hardware variants with CAN (1, 3, 5, 7)		
Pin	**Function**	**Description**
1	PWM input / digital input 5 / speed input 2 or analogue input 5 or temperature input	PWM input, 50 – 500 Hz / digital input 5 / speed input 2, square-wave signal (e.g. hall) or 0 – 5 V, analogue or NTC / Pt1000 / Pt200
2	CAN-Low	CAN, ISO 11898, 125 – 1000 kB/s
3	CAN-High	
4	Battery -	0 V from battery
5	Battery +	Supply voltage (battery +)
6	DcDesk communication	HZM diagnostic interface
7	DcDesk communication	HZM diagnostic interface
8	Analogue/digital input 2 or analogue output	0 – 5 V or 0 – 10 V analogue/digital input 2 0 – 5 V analogue output (max. 10 mA/500 Ω)
9	Speed input 1 or analogue/digital input 6	Speed input (inductive sensor) 0 – 5 V analogue input 6/digital input 6
10	GND	0 V for analogue inputs/digital inputs
11	Analogue/digital input 6	0 – 5 V/4 – 20 mA, analogue input 1 or digital input 1
12	+5 V reference	+5 V reference voltage, max. 10 mA
13	Digital output	Digital output, switches to 0 V, max. 0.3 A, can be configured e.g. as an error output
14	GND	0 V for speed input/digital inputs

Tab. 20: DC 10, pin assignment for CAN variants

9.11 DC 11

The DC 11 features 2 CAN controllers, but only CAN 1 is laid out on the 24-pin plug. The plugs with fewer pins do not feature any CAN lines.

Pin	Port	Function	Comments
1+5	-	Batt+	Voltage supply
2+6	-	Batt-	Voltage supply
3	DO	Digital output	Error lamp
4	P4	Digital/analogue input 4	0 - 5 V
7	0 V	0 V	0 V potential
8	P3	MFP 3	Multifunctional port 3
9	IS1	Speed input	For inductive sensors max. 9 kHz
10	-	Shield/housing	For cable shield
11	HZM comm	ISO K	HZM communication TX
12	HZM comm	ISO L	HZM communication RX
13	P1	MFP 1	Multifunctional port 1
14	P2	MFP 2	Multifunctional port 2
15	0 V	0 V	0 V potential
16	CAN L	CAN Low	Isolated
17	HZM comm	+UBatt output	For HZM communication
18	0 V	0 V	0 V potential
19	+5 V ref 1	+5 V reference 1	5 V reference for P1, P2, P3
20	P5	Digital/analogue input 5	0 – 5 V or 4 – 20 mA
21	CAN H	CAN High	Isolated
22	HZM comm	0 V	HZM comm 0 V potential
23	+5 V ref 2	+5 V reference 2	5 V reference
24	TI	Temperature sensor input	For NTC/PT1000

Tab. 21: DC 11, assignment of 24-pin plug

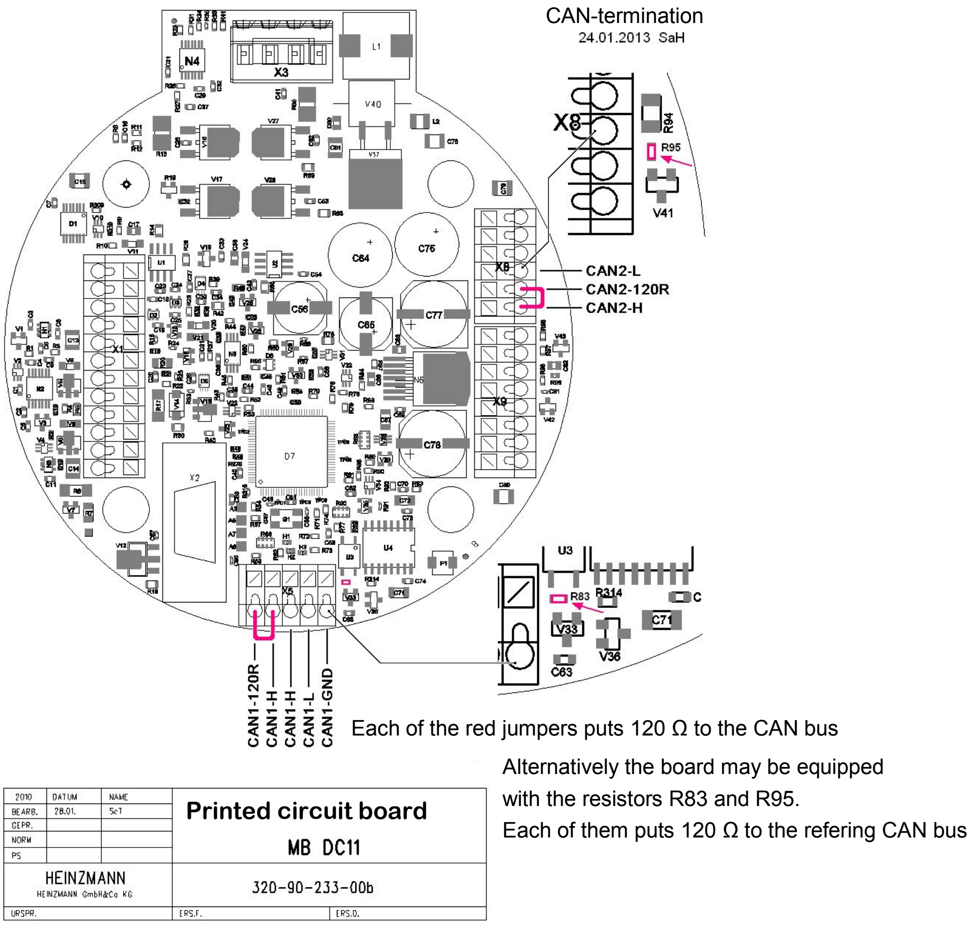


Fig. 32: DC 11, printed circuit board with position of terminating resistors

The terminating resistors are not equipped as standard. As the unit is completely closed, the resistances are connected externally unless the termination is explicitly requested in the production order.

9.12 DC 12

The DC12 does not feature a terminating resistor as standard, meaning that it must be connected externally. If the control unit is used in more large-scale series in which it is clear that it is located at the end of the CAN bus, the terminating resistor can be implemented on the printed circuit board by means of a soldering jumper.

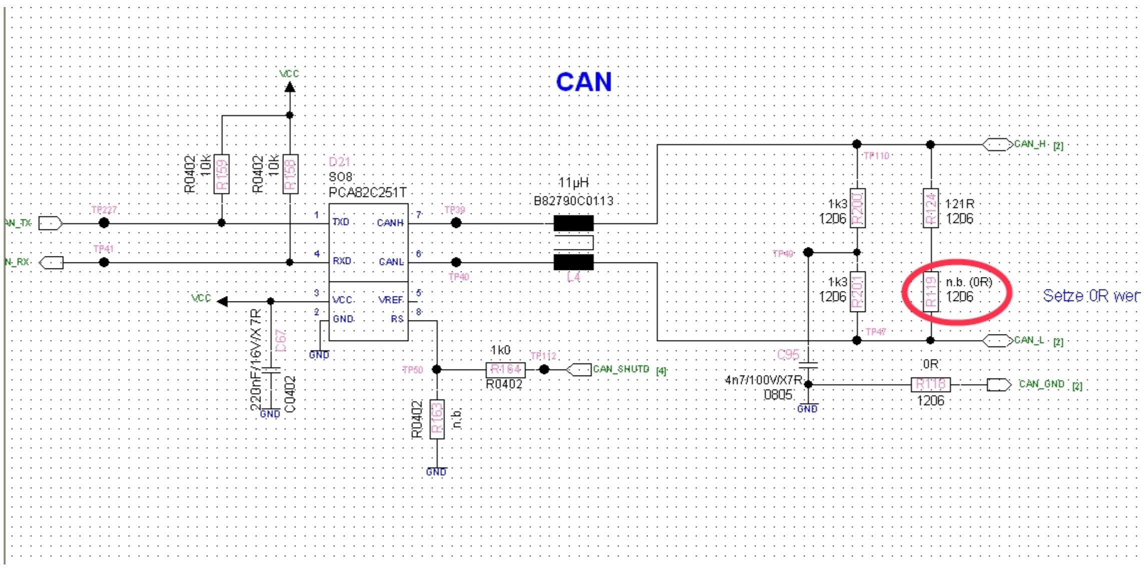


Fig. 33: DC 12, circuit diagram with position of terminating resistor

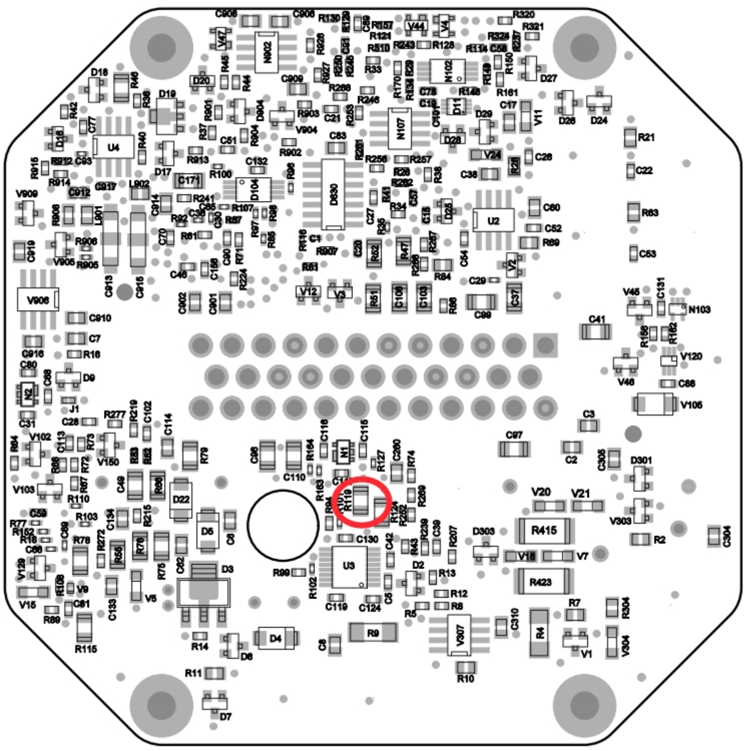


Fig. 34: DC 12, printed circuit board with position of terminating resistors

9.13 MVC 01

The Dardanos I features 2 CAN controllers.

Activate CAN1 terminating resistor:	Set J25 on the printed circuit board
Deactivate CAN1 terminating resistor:	Remove J25 from the printed circuit board
Activate CAN2 terminating resistor:	Set J26 on the printed circuit board
Deactivate CAN2 terminating resistor:	Remove J26 from the printed circuit board

The jumpers are set as standard upon delivery.

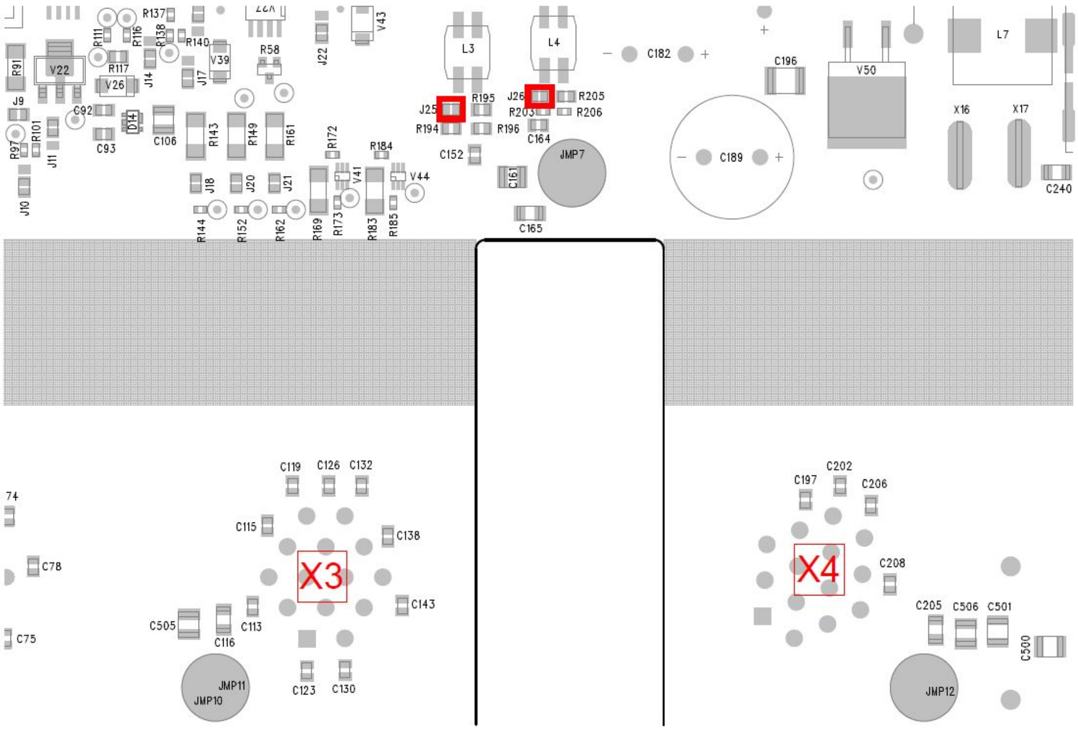


Fig. 35: MVC01, printed circuit board with position of terminating resistors (zoom)

9.14 MVC 01-3G

This control unit features 3 CAN controllers. They are laid out on the following pins:

CAN controller	Pins
CAN1-(H)	A:25 and A:29
CAN1-(L)	A:33 and A:37
CAN1-(G)	A:41 and A:45
CAN1-(R)	A:49
CAN2-(H)	A:26 and A:30
CAN2-(L)	A:34 and A:38
CAN2-(G)	A:42 and A:46
CAN2-(R)	A:50
CAN3-(H)	C:41 and C:45
CAN3-(L)	C:49 and C:53
CAN3-(R)	C:50

Tab. 22: MVC01-3G, CAN pin assignment

CAN1 and CAN2 are galvanically separated, whereas CAN3 is not.

The terminating resistors are not activated as standard. They can be set by connecting the respective CANx-(R) pins to the corresponding CANx-(L) pins.

9.15 MVC 03

DARDANOS III features 2 CAN controllers.

CAN controller	Connector pins
CAN1 HIGH	D:8 and D:45
CAN1 LOW	D:9 and D:40
CAN1 RES	D:54
CAN2 HIGH	D:22 and D:59
CAN2 LOW	D:23 and D:41
CAN2 RES	D:55

Tab. 23: MVC03, CAN pin assignment

The resistances can be pre-shielded in the pins or they can be housed externally in the line. The terminating resistors are not activated as standard. They can be set by connecting the respective CANx RES pins to the corresponding CANx LOW pins.

9.16 MVC 03-01

The 3 CAN controllers are laid out on the following pins:

CAN controller	TYCO pins	DEUTSCH pins
CAN1_H	B:40 and X2:39	B:8 and B:37
CAN1_L	B:44 and X2:43	B:9 and B:23
CAN1_RES	B:48	B:10
CAN2_H	A:16 and A:20 and C:8	A:4 and A:5 and C:2
CAN2_L	A:4 and A:8 and C:4	A:1 and A:2 and C:1
CAN2_RES	A:12	A:3
CAN3_H	B:56 and B:55	B:12 and B:26
CAN3_L	B:52 and B:51	B:11 and B:25
CAN3_RES	B:47	B:24
CAN120_1	C:51	C:25
CAN120_2	C:55	C:26

Tab. 24: MVC03-01, CAN pin assignment

Whereas CAN2 is available on the unit and engine side (pin A and C), CAN1 and CAN3 are only laid out on unit pin B. CAN1 is galvanically isolated, while the other 2 are not.

The terminating resistors are not activated as standard. They can be set by connecting the respective CANx_RES pins to the corresponding CANx_L pins. The CAN120_1 and CAN120_2 pins are simply a 120 Ω resistance. This can optionally be used as a terminating resistor.

9.17 MVC 04

DARDANOS IV features 2 CAN controllers. The unit does not feature a terminating resistor, meaning that this must be attached to the line on the exterior.

CAN controller	Connector pins
CAN1_H	X1_R:8
CAN1_L	X1_R:1
CAN2_H	X1_R:9
CAN2_L	X1_R:2

Tab. 25: MVC04, CAN pin assignment

9.18 THESEUS

THESEUS features 2 CAN controllers. CAN1 is used as standard for the HZM-CAN. In certain applications, CAN2 can also be used for a second HZM-CAN bus.

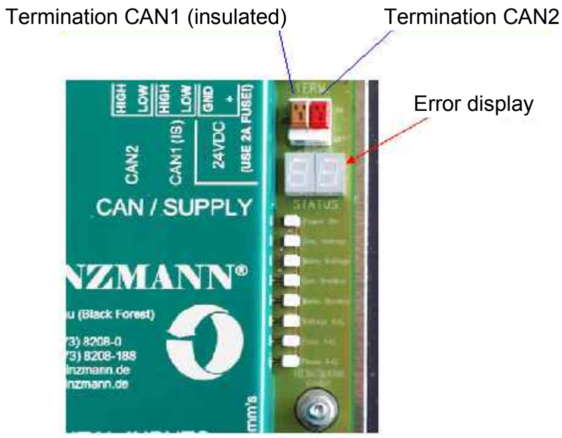


Fig. 36: THESEUS, position of terminating resistors

9.19 ARIADNE, PANTHEON

These control units feature 2 CAN controllers. CAN1 is used as standard for the HZM-CAN.

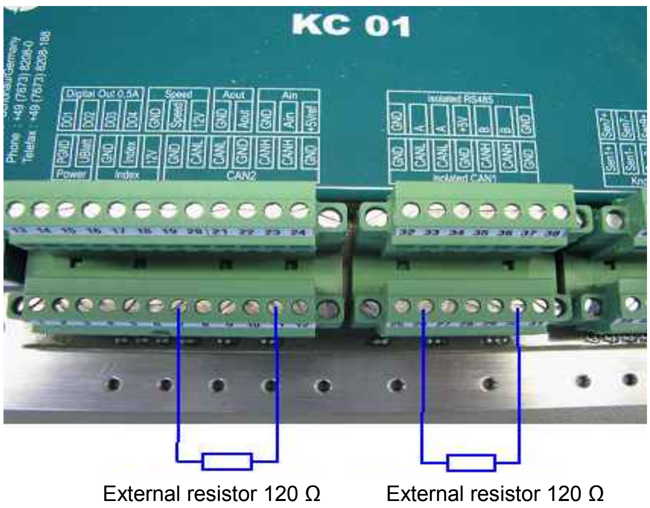


Fig. 37: ARIADNE, position of terminating resistors

9.20 PHLOX

The PHLOX ignition control unit features 2 CAN controllers.

Pin	Signal Name	Application	Function
2	+BAT	+ Power supply	Supply 24V DC
15	+BAT		
1	-BAT	- Power supply	
14	-BAT		
16	DIO1	Digital / PWM input 1 (high- or low-side configurable) Digital / PWM output 1 (low-side up to 1A).	DIO1
3	DIO2	Digital / PWM input 2 (high- or low-side configurable) Digital / PWM output 2 (low-side up to 1A).	DIO2
4	AI_POW(5VR/24V)	Sensor supply (configurable 5V/ 24V)	AI (C/V): Differential analogue input, configurable: 0 ... 5V oder 0 ... 25mA.
17	AI_SIG(C/V)	Sensor signal (configurable 0 ... 25mA/ 0 ... 5V)	
18	AI_SIG_0V	Signal ground (SIG_0V)	
5	AI_POW_0V	Sensor supply ground (POW_0V)	
6	AI_SHILD	Cable shield	
21	CAN-H	CAN-High	CAN- Interface ISO/DIS 11898 (CAN2.0B)
20	CAN-L	CAN low	
7	CAN-GND	CAN ground and CAN cable shield connection	
9	CAN2-H / MODBUS-A	CAN2-High / MODBUS-A	CAN2 / ModBus-Interface (option) ISO/DIS 11898 (CAN2.0B) RS485 (MODBUS)
8	CAN2-L / MODBUS-B	CAN2-Low / MODBUS-B	
19	CAN2 / MODBUS-GND	CAN2 / MODBUS ground and CAN2 / MODBUS cable shield connection	
11	SPEED_5/12V	Hall Speed pickup power supply	Speed_PickUp, magnetic or Hall
23	SPEED_SIG	Speed pickup input signal	
22	SPEED_0V	Speed pickup ground	
10	SPEED_SHIELD	Speed pickup cable shield	
13	INDEX_5/12V	Hall index pickup power supply	Index_PickUp,magnetic or Hall
25	INDEX_SIG	Index pickup input signal	
24	INDEX_0V	Index pickup ground	
12	INDEX_SHIELD	Index pickup cable shield	

Tab. 26: PHLOX IC8/12/16, Pin assignment Connector X1

Whereas CAN1 is internally terminated by PHLOX 1, it is no longer terminated as of PHLOX 2. The isolated CAN2 (only optionally equipped) is generally terminated. As of PHLOX 2, all terminating resistors must be attached externally.

9.21 SMC, E-LES SMC, GMA SMC

The stepper motor control features 1 CAN controller that is laid out on the pin. The terminating resistor is not present in the control unit. It must therefore be attached to the CAN line on the exterior.

Pin	Function	Range	Conditions	Optional
1	Analogue input	0..5 V	f_g = 15 Hz, R_i = 100 kΩ (0.5..4.5 V recommended)	-
	PWM input	50..500 Hz	U_{Low} < 2 V, U_{High} > 5 V	
	Temperature input	1 kΩ..5 kΩ	NTC/PTC sensors	
2	Digital input	0/1	U_{Low} < 2 V, U_{High} > 5 V	Various CAN protocols
	Analogue input	0..5 V	Ri = 100 kOhm	
	CAN-Low		125..1000 kBaud	
3	Digital input	0/1	U_{Low} < 2 V, U_{High} > 5 V	
	Analogue input	0..5 V	Ri = 100 kOhm	
	CAN-High		125..1000 kBaud	
4/5	Supply voltage	6..33 V	Current input max. 6 A	-
6/7	Diagnostic interface	2.4..57.6 kBaud/s	HZM communication	DcDesk 2000
8	Analogue output	0..5 V	1 kOhm max. load	-
		4..20 mA	250 Ohm max. load	-
	Analogue input	0..5 V	Ri = 100 kOhm	-
	Digital input	0/1	U_{Low} < 2 V, U_{High} > 5 V	-
9	Speed input	15..9000 Hz	Inductive sensor 0.3..30 V U_{pp}	Hall sensor
	Analogue input	0..5 V	Ri = 100 kOhm	-
10/14	GND	0 V	0 V reference for I/O	-
11	Analogue input	0..5 V	f_g = 15 Hz, R_i = 100 kΩ (0.5..4.5 V recommended)	-
		4..20 mA	f_g = 15 Hz, R_i = 200 Ω	
	Digital input	0/1	U_{Low} < 2 V, U_{High} > 5 V	-
12	5 V reference	5.0 V	I_{Ref} < 5 mA, ±1%	-
13	Digital output Error lamp	Max. 0.3 A	Switches to 0 V $U_{Remainder}$ < 1.0 V, I_{Leak} < 1 mA Note minimum current (parallel resistance for LEDs)	Error flash codes
	Digital input	0 / 1	< 1 V / > 5 V	-

Tab. 27: SMC, pin assignment

9.22 XIOS

XIOS features 2 CAN controllers. CAN A is isolated, CAN B is not.

A terminating resistor is equipped on the printed circuit board in each case, but is not activated as standard. A bridge must be set at the pin for the purpose of activation.

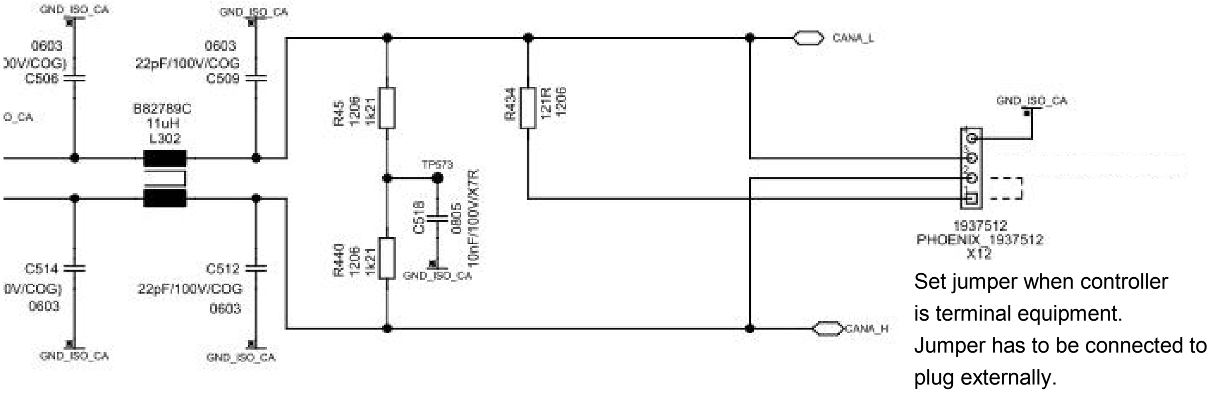


Fig. 38: XIOS, position of terminating resistor

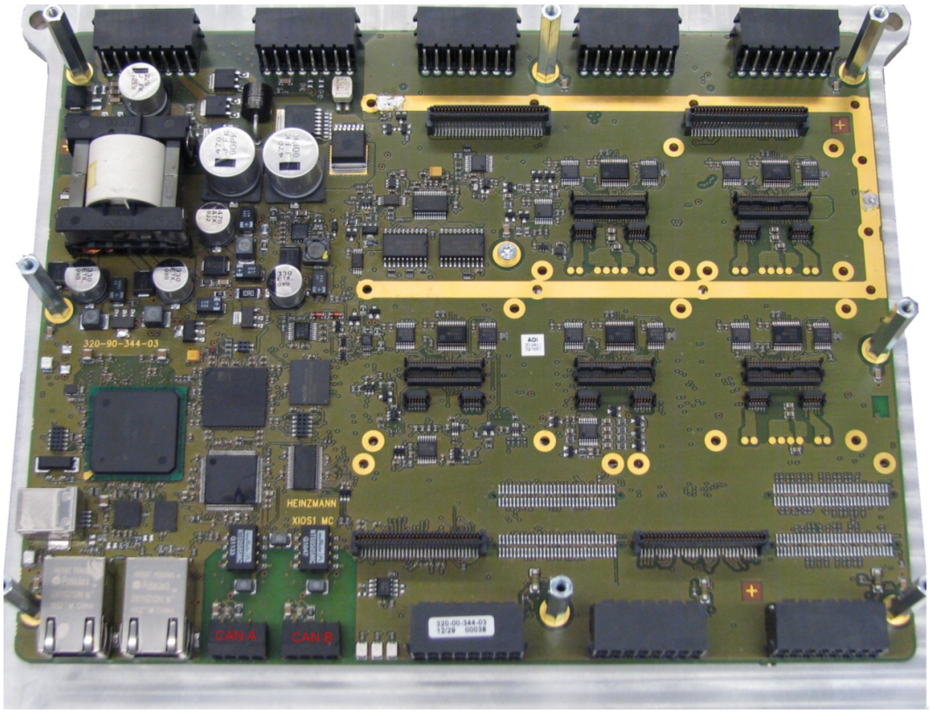


Fig. 39: XIOS, position of CAN on the printed circuit board (photo)

9.23 DigitalIO

DigitalIO features 1 CAN controller. The terminating resistor is activated by the jumper J200. R211, R210 and C207 should always be mounted (stub line connection). R213 should be activated at the ends with J200.

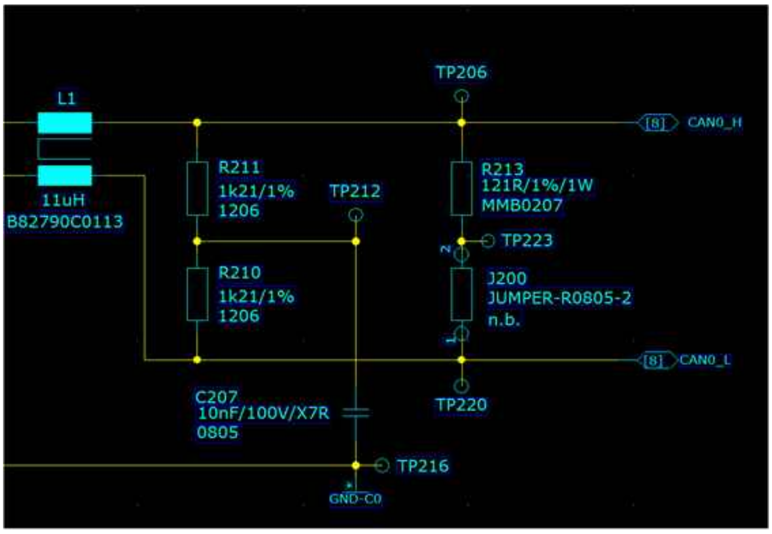


Fig. 40: DigitalIO, position of terminating resistor

9.24 AnalogIn

AnalogIn features 1 CAN controller.

R683, R685 and C670 should always be mounted (stub line connection). R687 should be activated at the ends with J1.

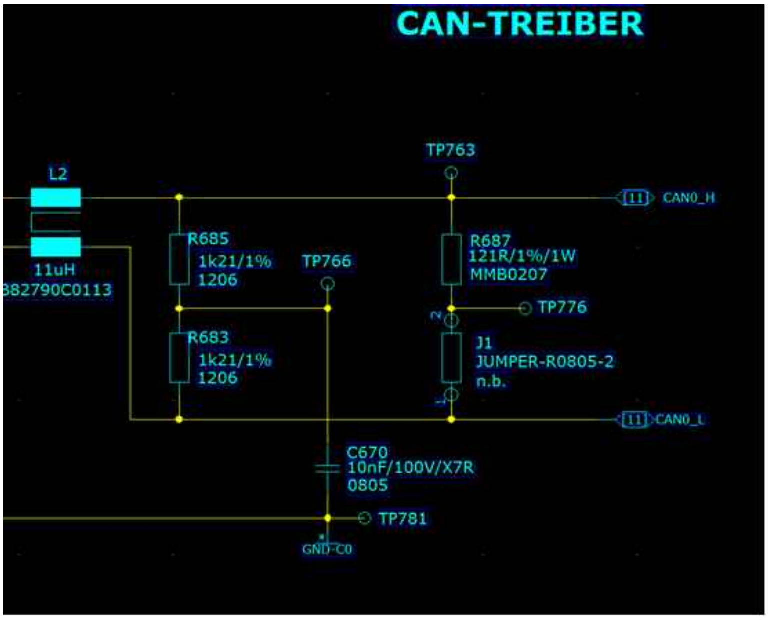


Fig. 41: AnalogIn, position of terminating resistor

9.25 ARGOS

ARGOS features 1 CAN controller. Termination is performed via terminals 7 and 8 on the rear of the housing. There is an internal connection between pin 5 and pin 7, as well as between pin 6 and pin 8.

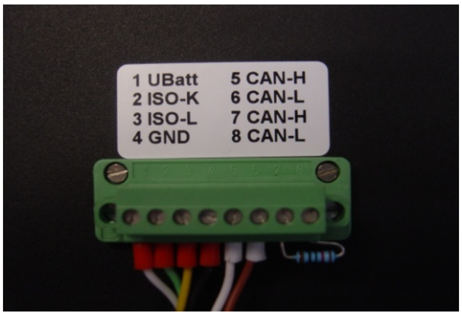


Fig. 42: ARGOS, position of terminating resistor

9.26 PANOPTES

PANOPTES Mobile (Wachendorf Opus A3) features 2 CAN controllers as standard. CAN1 is used for the customer module in HZM-CAN.

There is no internal terminating resistor present, meaning either that the unit must not be positioned at the end of the CAN bus or the terminating resistor must be housed in the pin, which is nearly impossible.

Pin	Name	Description
8	CAN1H	CAN bus 1 high signal
9	CAN1L	CAN bus 1 low signal
10	CAN2H	CAN bus 2 high signal
11	CAN2L	CAN bus 2 low signal

Tab. 28: PANOPTES Mobile (Wachendorf Opus A3), CAN pin assignment

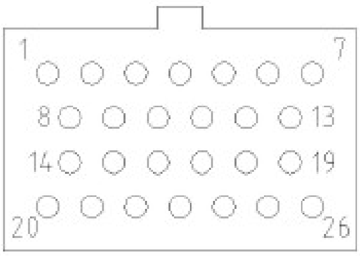
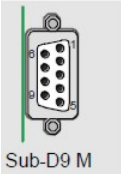


Fig. 43: PANOPTES Mobile (Wachendorf Opus A3), pin assignment

PANOPTES 2 (Berghoff DC1000) features 1 CAN controller.

The terminating resistor is activated or deactivated using a switch on the rear of the unit.



1	NC (Do not connect)
2	**CAN_L**
3	**CAN_GND**
4	NC (Do not connect)
5	NC (Do not connect)
6	NC (Do not connect)
7	**CAN_H**
8	NC (Do not connect)
9	NC (Do not connect)

Fig. 44: PANOPTES 2 (Berghoff DC1000), pin assignment

10 Parameter description

The following chapters contain the parameters relevant to the configuration of the HZM-CAN, along with their individual meaning.

For a detailed description of the parameters used to connect customer modules, see 'DG 05 007-d 05-12 HZM-CAN Customer-Modul.pdf'.

For other parameters of the control units, refer to the corresponding basic information and the following in particular:

> *Steuergeräte mit konventioneller Einspritzung über Stellgeräte [Control units with conventional injection via actuators], publication no. DG 07 001-d*

and

> *DARDANOS Basic Information, publication no. MV 09 001-e*

10.1 Parameter

No.	Name		Meaning
330	**CanSpeedSrcNodeNo**		
	Level:	6	Node number for the sender of the speed via multipoint connection
	Range:	0..31	
	Page(s):	54	
331	**CanSpeedSrcNodeType**		
	Level:	6	Node type for the sender of the speed via multipoint connection
	Range:	0..14	
	Page(s):	54	
332	**CanPowerSrcNodeNo**		
	Level:	6	Node number for the sender of the measured power via multipoint connection
	Range:	0..31	
	Page(s):	54	
333	**CanPowerSrcNodeType**		
	Level:	6	Node type for the sender of the measured power via multipoint connection
	Range:	0..14	
	Page(s):	54	
336	**CanSpSetpSrcNodeNo**		
	Level:	6	Node number for the sender of the setpoint speed via multipoint connection
	Range:	0..31	
	Page(s):	56	
337	**CanSpSetpSrcNodeType**		
	Level:	6	Node type for the sender of the setpoint speed via multipoint connection
	Range:	0..14	
	Page(s):	56	
338	**CanBoostSrcNodeNo**		
	Level:	6	Node number for the sender of the boost pressure via multipoint connection
	Range:	0..31	
	Page(s):	57	

No.	Name		Meaning
339	**CanBoostSrcNodeType**		
	Level:	6	Node type for the sender of the boost pressure via multipoint connection
	Range:	0..14	
	Page(s):	57	
342	**CanExhTempSrcNodeNo**		
	Level:	6	Node number for the sender of the exhaust temperature via multipoint connection
	Range:	0..31	
	Page(s):	57	
343	**CanExhTmpSrcNodeType**		
	Level:	6	Node type for the sender of the exhaust temperature via multipoint connection
	Range:	0..14	
	Page(s):	57	
344	**CanFuelSetpSrcNodeNo**		
	Level:	6	Node number for the sender of the filling via multipoint connection
	Range:	0..31	
	Page(s):	58	
345	**CanFuelSpSrcNodeType**		
	Level:	6	Node type for the sender of the filling via multipoint connection
	Range:	0..14	
	Page(s):	58	
397	**PartnerDCNodeNumber**		
	Level:	6	Node number for the other speed governor on the same control lever (2*DC)
	Range:	0..31	
	Page(s):	27	
398	**ThirdDCNodeNumber**		
	Level:	6	Node numbers for the third speed governor in ship projects on the other control lever (4*DC)
	Range:	0..31	
	Page(s):	27	
399	**FourthDCNodeNumber**		
	Level:	6	Node numbers for the fourth speed governor in ship projects on the other control lever (4*DC)
	Range:	0..31	
	Page(s):	27	
400	**CanStartTimeOutDelay**		
	Level:	6	Delay for the monitoring of the CAN connection after reset
	Range:	0..100 s	
	Page(s):	60	
401	**CanMyNodeNumber**		
	Level:	6	Control unit's own node number in the CAN network
	Range:	0..31	
	Page(s):	26, 65, 66	
402	**CanDCNodeNumber**		If not type DC itself
	CanOtherNodeNumber		If type DC itself (master/slave in ships)
	Level:	6	Control unit's own node number in the CAN network
	Range:	0..31	
	Page(s):	27, 56	
403	**CanCMNodeNumber**		
	Level:	6	Node number for the customer module in the CAN network
	Range:	0..31	
	Page(s):	28	

No.	Name		Meaning
403	**CanMasterNodeNumber**		**Periphery module**
	Level:	6	Node number for the master module in the CAN network
	Range:	0..31	
	Page(s):	28, 66	
404	**CanPENodeNumber**		
ff.	Level:	6	Node numbers for the periphery modules in the CAN
	Range:	0..31	network
	Page(s):	19, 22, 27, 38	
		43,45, 47, 65, 114, 115	
407	**CanPENodeType**		
ff.	Level:	6	Periphery module types for the periphery modules in the
	Range:	0..20	CAN network
	Page(s):	22, 42, 65, 65	
416	**CanBaudrate**		
ff.	**CanxBaudrate**		
	Level:	6	CAN baud rate
	Range:	0..7	CAN controller x
	Page(s):	13, 65, 66	x = 1…2
430	**CanACNodeNumber**		
ff.	Level:	6	Node numbers for the additional modules in the CAN
	Range:	0..31	network
	Page(s):	24, 27, 55, 65, 65, 102	
435	**CanACNodeType**		
ff.	Level:	6	Additional module types for the additional modules in the
	Range:	0..10	CAN network
	Page(s):	24, 65	
440	**PEFuelSetpSendRate**		**Master to periphery module**
	Level:	6	Send rate for the filling level
	Range:	0..100,00 s	
	Page(s):	44	
440	**CanActPosSendRate**		**Periphery module to master**
	Level:	6	Send rate for the actual value of the actuators
	Range:	0..100,00 s	
	Page(s):	34	
441	**PEDigOutSendRate**		**Master to periphery module**
	Level:	6	Send rate for the binary outputs
	Range:	0..100,00 s	
	Page(s):	47	
441	**CanDigInSendRate**		**Periphery module to master**
	Level:	6	Send rate for the binary inputs
	Range:	0..100,00 s	
	Page(s):	35	
442	**PEAnalogOutSendRate**		**Master to periphery module**
	Level:	6	Send rate for the analogue outputs
	Range:	0..100,00 s	
	Page(s):	48	

No.	Name		Meaning
442	**CanAnalogInSendRate**		**Periphery module to master**
	Level:	6	Send rate for the analogue inputs
	Range:	0..100,00 s	
	Page(s):	35	
443	**PEPWMOutSendRate**		**Master to periphery module**
	Level:	6	Send rate for the PWM outputs
	Range:	0..100,00 s	
	Page(s):	48	
443	**CanTempInSendRate**		**Periphery module to master**
	Level:	6	Send rate for the temperature inputs
	Range:	0..100,00 s	
	Page(s):	36	
444	**CanPWMInSendRate**		**Periphery module to master**
	Level:	6	Send rate for the PWM inputs
	Range:	0..100,00 s	
	Page(s):	37	
445	**CanSpeedSendRate**		**Periphery module to master**
	Level:	6	Send rate for the speed
	Range:	0..100,00 s	
	Page(s):	37	
446	**CanVelocitySendRate**		**Periphery module to master**
	Level:	6	Send rate for the travel speed
	Range:	0..100,00 s	
	Page(s):	37	
450 ff.	**PEDigOut1_Assign**		**Master to periphery module (without common outputs)**
	Level:	6	Assignment parameter for a binary output
	Range:	-29999..29999	
	Page(s):	43, 47	
455 ff.	**PEPWMOut1_Assign**		**Master to periphery module (without common outputs)**
	Level:	6	Assignment parameter for a PWM output
	Range:	-29999..29999	
	Page(s):	43, 48	
458 ff.	**PEPWMOut1_ValueMin**		**Master to periphery module (without common outputs)**
	Level:	6	Lower limit value of the assignment parameter, corresponds to 0% at the PWM output
	Range:	0..100,0 %	
	Page(s):		
459 ff.	**PEPWMOut1_ValueMax**		**Master to periphery module (without common outputs)**
	Level:	6	Upper limit value of the assignment parameter, corresponds to 100% at the PWM output
	Range:	0..100,0 %	
	Page(s):		
480 ff.	**PEAnaOut1_Assign**		**Master to periphery module (without common outputs)**
	Level:	6	Assignment parameter for a PWM output
	Range:	-29999..29999	
	Page(s):	43, 48	
483 ff.	**PEAnaOut1_ValueMin**		**Master to periphery module (without common outputs)**
	Level:	6	Lower limit value of the assignment parameter, corresponds to 0% at the analogue output
	Range:	0..100,0 %	
	Page(s):		

No.	Name		Meaning
484	**PEAnaOut1_ValueMax**		**Master to periphery module (without common outputs)**
ff.	Level:	6	Upper limit value of the assignment parameter, corresponds
	Range:	0..100,0 %	to 100% at the analogue output
	Page(s):		
810	**FunctEngineStop**		
ff.	Level:	6	Assigning a channel number to a switching function
	Range:	-xx..xx	0: Switching function not assigned
	Page(s):	35	-xx..xx: channel number
900	**AssignIn_Setp1Ext**		
ff.	Level:	6	Assigning a channel number to a sensor
	Range:	0..xx	0: Sensor not assigned
	Page(s):	35, 37	1..xx: channel number
904	**AssignIn_BoostPress**		
	Level:	6	Assigning a channel number to the boost pressure sensor
	Range:	0..xx	0: Sensor not assigned
	Page(s):	57	1..xx: channel number
911	**AssignIn_ExhaustTemp**		
	Level:	6	Assigning a channel number to the exhaust temperature
	Range:	0..xx	sensor
	Page(s):	57	0: Sensor not assigned
			1..xx: channel number
1601	**PWMOut1_RefLow**		**Not XIOS**
ff.	Level:	4	Minimum value of PWM output 1
	Range:	0..100 %	
	Page(s):	41	
1602	**PWMOut1_RefHigh**		**Not XIOS**
ff.	Level:	4	Maximum value of PWM output 1
	Range:	0..100 %	
	Page(s):	41	
1625	**PWMOutFrequency**		**Not XIOS**
	Level:	4	Frequency of the PWM outputs
	Range:	0..xx Hz	
	Page(s):	41	
1626	**PowerOutFrequency**		**DC 2**
	Level:	6	Frequency of the power output
	Range:	0..xx Hz	
	Page(s):	41	
1641	**AnalogOut1_RefLow**		**Not XIOS**
	CurrentOut1_RefLow		
	VoltOut1_RefLow		
ff.	Level:	4	Minimum value of analogue output 1
	Range:	0..100 %	
		oder 0..22 mA	
		oder 0..5 V	
		oder 0..10 V	
	Page(s):	41	

No.	Name		Meaning
1642	**AnalogOut1_RefHigh**		**Not XIOS**
	CurrentOut1_RefHigh		
	VoltOut1_RefHigh		
ff.	Level:	4	Maximum value of analogue output 1
	Range:	0..100 %	
		oder 0..22 mA	
		oder 0..5 V	
		oder 0..10 V	
	Page(s):	41	
10056	**KnockModulACIndex**		**DUAL-fuel only**
	Level:	6	Field index of ARIADNE 430 *CanACNodeNumber*()
	Range:	0..4	
	Page(s):	27	
11400	**Out1:Assign**		**Common outputs only**
ff.	Level:	6	Assignment parameter for an output (source)
	Range:	0..29999	0: no assignment
	Page(s):	43, 45, 47	
11401	**Out1:ValueMin**		**Common outputs only**
ff.	Level:	6	Lower limit value of the assignment parameter, corresponds
	Range:	0..100,0 %	to 0% at the output
	Page(s):	43	
11402	**Out1:ValueMax**		**Common outputs only**
ff.	Level:	6	Upper limit value of the assignment parameter, corresponds
	Range:	0..100,0 %	to 100% at the output
	Page(s):	43	
11403	**Out1:DigOutBitMask**		**Common outputs and error status only**
ff.	Level:	6	Bit mask if the assignment parameter is an error status
	Range:	0..0xFFFF	
	Page(s):	43	
20810	**CommEngineStop**		
ff.	Level:	6	Assigning a communication channel number to a switching
	Range:	-xx..xx	function
	Page(s):	35	0: Switching function not assigned
			-xx..xx: channel number
30020	**P001_(1.1)_IO_RefLow**		**XIOS**
ff.	Level:	6	Minimum value of the output port
	Range:	0..5 V	
		0..24 mA	
	Page(s):	39	
30021	**P001_(1.1)_IO_RefHigh**		**XIOS**
ff.	Level:	6	Maximum value of the output port
	Range:	0..5 V	
		0..24 mA	
	Page(s):	39	
30025	**P001_(1.1)_PO_Freq**		**XIOS**
ff.	Level:	6	Frequency of the PWM output
	Range:	126..15625 Hz	
	Page(s):	40	

10.2 Measured values

No.	Name		Meaning
2000	**Speed**		
	Level:	1	Speed
	Range:	0..4000 Min^{-1}	
	Page(s):	51	
2001	**SpeedPickUp1**		**Periphery module to master**
	Level:	1	Speed at pickup 1
	Range:	0..4000 Min^{-1}	
	Page(s):	37	
2002	**SpeedPickUp2**		**Periphery module to master**
	Level:	1	Speed at pickup 2
	Range:	0..4000 Min^{-1}	
	Page(s):	37	
2006	**SpeedCan**		
	Level:	1	Speed received via multipoint connection
	Range:	0..4000 Min^{-1}	
	Page(s):	54	
2031	**SpeedSetp**		
	Level:	1	Independently determined speed or
	Range:	0..4000 Min^{-1}	speed setpoint received via multipoint connection
	Page(s):	52, 56	
2050	**SpeedVariance**		**Periphery module to master**
	Level:	1	MisfireDetection: SpeedVariance, either measured by master
	Range:	0..100 %	itself or display if received by periphery module
	Page(s):	38	
2300	**ActPos**		**Periphery module to master**
ff.	Level:	1	Actuator measured values (feedback)
	Range:	0..100 %	
	Page(s):	34	
2305	**PExActPos**		**Master from periphery module**
ff.	Level:	1	Actuator measured value of the periphery module (feedback)
	Range:	0..100 %	Display in master
	Page(s):	35	
2320	**PExActuatorOn**		**Master from periphery module**
ff.	Level:	1	1 = actuator released in the periphery module
	Range:	0..1	Display in master
	Page(s):	35	
2350	**FuelQuantity**		**Periphery module to master**
ff.	Level:	1	Master filling level is reset
	Range:	0..100 %	
	Page(s):	35, 46, 52	
2355	**PEFuelQuantityx**		**Master to periphery module**
ff.	Level:	1	Filling level
	Range:	0..100 %	
	Page(s):	44, 46	

No.	Name		Meaning
2401	**CanTxBufferState**		
	Level:	1	Status of CAN send buffer
	Range:	0000..FFFF Hex	(Display of send buffer exceedance by unit type)
	Page(s):	61	
2402	**CanRxBufferState**		
	Level:	1	Status of CAN receive buffer
	Range:	0000..FFFF Hex	(Display of receive buffer exceedance by unit type)
	Page(s):	61	
2403	**CanRxTimeout**		
	Level:	1	Status of CAN receive timeout monitoring
	Range:	0000..FFFF Hex	(Display of receive timeout by unit type)
	Page(s):	61	
2404	**CanTypeMismatch**		
	Level:	1	Status of CAN unit type monitoring
	Range:	0/1	(Display of node numbers with duplicate assignment)
	Page(s):	61	
2405	**CanOnline**		
ff.	**CanxOnline**		
	Level:	1	General status of CAN communication
	Range:	0/1	CAN controller x
	Page(s):	62	x = 1…2
2410	**CanDCNodeState31to16**		
	Level:	1	Connection status to speed governors with
	Range:	0000..FFFF Hex	node numbers 16 to 31
	Page(s):	31	
2411	**CanDCNodeState15to01**		
	Level:	1	Connection status to speed governors with
	Range:	0000..FFFF Hex	node numbers 1 to 15
	Page(s):	31	
2412	**CanGCNodeState31to16**		
	Level:	1	Connection status to generator control units with
	Range:	0000..FFFF Hex	node numbers 16 to 31
	Page(s):	31	
2413	**CanGCNodeState15to01**		
	Level:	1	Connection status to generator control units with
	Range:	0000..FFFF Hex	node numbers 1 to 15
	Page(s):	31	
2414	**CanPENodeState31to16**		
	Level:	1	Connection status to periphery modules with
	Range:	0000..FFFF Hex	node numbers 16 to 31
	Page(s):	31	
2415	**CanPENodeState15to01**		
	Level:	1	Connection status to periphery modules with
	Range:	0000..FFFF Hex	node numbers 1 to 15
	Page(s):	31	
2420	**CanACNodeState31to16**		
	Level:	1	Connection status to additional modules with
	Range:	0000..FFFF Hex	node numbers 16 to 31
	Page(s):	31	

No.	Name		Meaning
2421	**CanACNodeState15to01**		
	Level:	1	Connection status to additional modules with
	Range:	0000..FFFF Hex	node numbers 1 to 15
	Page(s):	31	
2422	**CanCMNodeState31to16**		
	Level:	1	Connection status to customer modules with
	Range:	0000..FFFF Hex	node numbers 16 to 31
	Page(s):	31	
2423	**CanCMNodeState15to01**		
	Level:	1	Connection status to customer modules with
	Range:	0000..FFFF Hex	node numbers 1 to 15
	Page(s):	31	
2424	**CanPCNodeState31to16**		
	Level:	1	Connection status to communication modules with
	Range:	0000..FFFF Hex	node numbers 16 to 31
	Page(s):	31	
2425	**CanPCNodeState15to01**		
	Level:	1	Connection status to communication modules with
	Range:	0000..FFFF Hex	node numbers 1 to 15
	Page(s):	31	
2440	**CanPEError**		
ff.	Level:	1	Error status of the connected periphery modules
	Range:	0000..FFFF Hex	If no error status in recipient
	Page(s):	38	
2443	**CanACError**		
ff.	Level:	1	Error status of the connected additional modules
	Range:	0000..FFFF Hex	If no error status in recipient
	Page(s):	55	
2470	**PEDigitalOutx**		**Master to periphery module (without common outputs)**
ff.	Level:	1	Setpoints for binary outputs
	Range:	0..3	
	Page(s):	47	
2475	**PEPWMOutx**		**Master to periphery module (without common outputs)**
ff.	Level:	1	Setpoints for PWM outputs
	Range:	0..3	
	Page(s):	48	
2480	**PEAnaOutx**		**Master to periphery module (without common outputs)**
ff.	Level:	1	Setpoints for analogue outputs
	Range:	0..3	
	Page(s):	48	
2489	**PEModulesMax**		
	Level:	1	Maximum number of periphery modules that can be
	Range:	0..3	connected
	Page(s):	22	
2490	**PEModulesMaxType(x)**		
ff.	Level:	1	Maximum number of type x periphery modules that can be
	Range:	0..3	connected
	Page(s):	22	x = 0..20

No.	Name		Meaning
2540	**CanDCStatus**		
	Level:	1	Status of the DC module via multipoint connection
	Range:	0..FFFF Hex	
	Page(s):	56	
2541 ff.	**CanACStatus**		
	Level:	1	Status of the additional modules via multipoint connection
	Range:	0..FFFF Hex	
	Page(s):	56	
2549	**ACModulesMax**		
	Level:	1	Maximum number of periphery modules that can be connected
	Range:	0..5	
	Page(s):	24	
2550	**ACModulesMaxType(x)**		
	Level:	1	Maximum number of type x periphery modules that can be connected
	Range:	0..5	
	Page(s):	24	x = 0..10
2685	**WasteGate**		
	Level:	1	Filling level for the Wastegate
	Range:	0..100 %	
	Page(s):	47	
2686	**BypassValve**		
	Level:	1	Filling level for the bypass valve
	Range:	0..100 %	
	Page(s):	46	
2810 ff.	**SwitchEngineStop**		
	Level:	1	Status of the switching functions
	Range:	0..3	
	Page(s):	35	
2811 ff.	**DigitalInx**		**Periphery module to master**
	Level:	1	Status of the binary inputs
	Range:	0..3	
	Page(s):	35	
2828	**SwitchErrorReset**		
	Level:	1	Status of the switching function 'Delete errors'
	Range:	0..3	
	Page(s):	49, 53, 112	
2900	**Setpoint1Ext**		
	Level:	1	Display of setpoint adjuster 1
	Range:	0..100 %	
	Page(s):	35, 37	
2904	**BoostPressure**		
	Level:	1	Display of boost pressure
	Range:	0..5 bar	
	Page(s):	52, 57	
2911	**ExhaustTemp**		
	Level:	1	Display of exhaust temperature if measured independently or received via multipoint connection
	Range:	-100..1000 °C	
	Page(s):	52, 57	

No.	Name		Meaning
2911	**ExhaustTempMax**		
	Level:	1	Display of maximum exhaust temperature for multiple cylinders in the master if received by the periphery module
	Range:	-100..1000 °C	
	Page(s):	38	
2912	**ExhaustTempMin**		
	Level:	1	Display of minimum exhaust temperature for multiple cylinders in the master if received by the periphery module
	Range:	-100..1000 °C	
	Page(s):	38	
2913	**ExhaustTempAverage**		
	Level:	1	Display of average exhaust temperature for multiple cylinders in the master if received by the periphery module
	Range:	-100..1000 °C	
	Page(s):	38	
3000	**ConfigurationError**		
	Level:	1	Displays configuration errors
	Range:	0..65535	
	Page(s):	59, 108	
3003	**ErrPickUpCan**		
	Level:	1	Display of error when receiving the speed via multipoint connection
	Range:	0..1 oder 0..FFFF Hex	
	Page(s):	54	
3004	**ErrOverSpeed**		
	Level:	1	Display of overspeed error
	Range:	0..1 oder 0..FFFF Hex	
	Page(s):	55	
3009	**ErrBoostPressure**		
	Level:	1	Display of boost pressure error
	Range:	0..1 oder 0..FFFF Hex	
	Page(s):	57	
3016	**ErrExhaustTemp**		
	Level:	1	Display of exhaust temperature error
	Range:	0..1 oder 0..FFFF Hex	
	Page(s):	57	
3025	**ErrPowerPercentCan**		
	Level:	1	Display of error when receiving the measured power as a percentage
	Range:	0..1 oder 0..FFFF Hex	
	Page(s):	54	
3041	**ErrSpeedSetpCan**		
	Level:	1	Display of error when receiving the speed setpoint
	Range:	0..1 oder 0..FFFF Hex	
	Page(s):	56	
3049	**ErrPECommonAlarm**		**Periphery module to master**
	Level:	1	If no error status in the master
	Range:	0..1	CommonAlarm in the periphery module
	Page(s):	38	
3070	**ErrCanBus**		
ff.	**ErrCanBusx**		
	Level:	1	Error display of the CAN bus
	Range:	0/1	CAN controller x
	Page(s):	55, 61	x = 1…2

No.	Name		Meaning
3071	**ErrCanComm**		
ff.	**ErrCanCommx**		
	Level:	1	Error display of CAN communication
	Range:	0/1	CAN controller x
	Page(s):	55, 61	x = 1…2
3089	**ErrPEFatalError**		**Periphery module to master**
	Level:	1	If no error status in the master
	Range:	0..1	Fatal error in the periphery module
	Page(s):	38	
3092	**ErrConfiguration**		
	Level:	1	0: no configuration error
	Range:	0..1	1: configuration error displayed in 3000 *ConfigurationError*
	Page(s):	59	
3232	**RelativePower**		
	Level:	1	Relative measured power
	Range:	0..200 % oder -200..200 %	
	Page(s):	51, 54	
3300	**Velocity**		**Periphery module to master**
	Level:	1	Speed
	Range:	0..200 km/h	
	Page(s):	37	
3500	**PWMInx**		**Periphery module to master**
ff.	Level:	1	Status of the PWM inputs
	Range:	0..100 %	
	Page(s):	37	
3510	**AnalogIn1_Percent**		**Periphery module to master**
ff.	Level:	1	Status of the analogue inputs
	Range:	0..100 %	
	Page(s):	36	
3550	**TempInx**		**Periphery module to master**
ff.	Level:	1	Status of the temperature inputs
	Range:	-100..1000 °C	
	Page(s):	36	
3800	**EmergencyAlarm**		
	Level:	1	1: fatal error in the control unit
	Range:	0..1	
	Page(s):	55	
3801	**CommonAlarm**		
	Level:	1	1: general error in the control unit
	Range:	0..1	
	Page(s):	55	
3834	**CanBusAvailable**		**DC 10**
	Level:	1	0: no CAN available
	Range:	0..1	1: CAN available
	Page(s):	82	
3858	**CompileDate**		
	Level:	1	Display of month from the software creation date
	Range:	0..1	
	Page(s):	25	

No.	Name		Meaning
3859	**CompileYear**		
	Level:	1	Display of year from the software creation date
	Range:	0..1	
	Page(s):	25	
12022	**GasFuelQuantity**		**Dual-fuel**
	Level:	1	Gas filling level
	Range:	0..100 %	
	Page(s):	46	
12570	**ExhaustTempAverage**		**Periphery module**
	Level:	1	Display of average exhaust temperature for multiple
	Range:	-100..1000 °C	cylinders in the periphery module
	Page(s):	38	
12572	**ExhaustTempMin**		**Periphery module**
	Level:	1	Display of minimum exhaust temperature for multiple
	Range:	-100..1000 °C	cylinders in the periphery module
	Page(s):	38	
12573	**ExhaustTempMax**		**Periphery module**
	Level:	1	Display of maximum exhaust temperature for multiple
	Range:	-100..1000 °C	cylinders in the periphery module
	Page(s):	38	
13400	**Out1:Value**		**Common outputs only**
ff.	Level:	1	Value of output defined in 15400 *Out1:Destination* onwards
	Range:	0..100 %	
	Page(s):	43, 45, 47	
13650	**SparkDuration1**		**Phlox**
ff.	Level:	1	Display of diagnosis time per cylinder
	Range:	0..1,5 ms	
	Page(s):	38	
23002	**ErrCanPE**		**Master from periphery module**
ff.	Level:	1	Error status of the connected periphery modules
	Range:	0000..FFFF Hex	If error status in master
	Page(s):	38	
23006	**ErrCanAC**		If error status in recipient
ff.	Level:	1	Error status of connected additional modules
	Range:	0000..FFFF Hex	
	Page(s):	55	
32000	**P089_(MC.DI1)_FI_FreqIn**		**XIOS**
ff.	Level:	1	Display of measured values at all channels
	Range:	0..500 Hz	
	Page(s):	35	

10.3 Functions

No.	Name		Meaning
4009	**CanSpeedOn**		
	Level:	6	Activates usage of speed received via multipoint connection
	Range:	0/1	
	Page(s):	54	
4031	**CanSpeedSetpOn**		
	Level:	6	Activates usage of speed setpoint received via multipoint connection
	Range:	0/1	
	Page(s):	56	
4330	**AllSendSpeedOn**		
	Level:	6	Activates sending of speed via multipoint connection
	Range:	0/1	
	Page(s):	51	
4332	**AllSendPowPercentOn**		
	Level:	6	Activates sending of relative measured power via multipoint connection
	Range:	0/1	
	Page(s):	51	
4334	**AllSendErrorStatusOn**		**Additional modules**
	Level:	6	Activates sending of error status of additional modules via multipoint connection
	Range:	0/1	
	Page(s):	51, 66	
4335	**AllSendStatusOn**		
	Level:	6	Activates sending of status via multipoint connection
	Range:	0/1	
	Page(s):	52	
4336	**AllSendSpeedSetpOn**		
	Level:	6	Activates sending of speed setpoint via multipoint connection
	Range:	0/1	
	Page(s):	52	
4338	**AllSendBoostPressOn**		
	Level:	6	Activates sending of boost pressure via multipoint connection
	Range:	0/1	
	Page(s):	52	
4340	**AllSendAutoResetOn**		
	Level:	6	Activates sending of auto reset request via multipoint connection
	Range:	0/1	
	Page(s):	53	
4341	**AllSendErrorResetOn**		
	Level:	6	Activates sending of error reset request via multipoint connection
	Range:	0/1	
	Page(s):	53	
4342	**AllSendExhaustTempOn**		
	Level:	6	Activates sending of exhaust temperature via multipoint connection
	Range:	0/1	
	Page(s):	52	

No.	Name		Meaning
4344	**AllSendFuelSetpOn**		
	Level:	6	Activates sending of filling via multipoint connection
	Range:	0/1	
	Page(s):	52	
4400	**CanCommDCOn**		
	Level:	6	Activates connection to speed governor
	Range:	0/1	
	Page(s):	29, 54, 56, 66	
4401	**CanCommGCOn**		
	Level:	6	Activates connection to generator control unit
	Range:	0/1	
	Page(s):	29, 65	
4402	**CanCommPEOn**		
	Level:	6	Activates connection to periphery module
	Range:	0/1	
	Page(s):	29, 65	
4405	**CanCommACOn**		
	Level:	6	Activates connection to additional module
	Range:	0/1	
	Page(s):	29, 53, 56, 57, 65	
4406	**CanCommCMOn**		
	Level:	6	Activates connection to customer module
	Range:	0/1	
	Page(s):	29	
4415	**CanCommALLOn**		
	Level:	6	Activates sending to all or reception via multipoint connection
	Range:	0/1	
	Page(s):	29, 51, 53, 58, 65	
4430	**ReceiveACErrorOn**		
	Level:	6	Activates reception of AC error status via multipoint connection
	Range:	0/1	
	Page(s):	55, 65	
4431	**ReceiveStatusOn**		
	Level:	6	Activates reception of module status via multipoint connection
	Range:	0/1	
	Page(s):	56	
4440	**PEFuelSetpointOn**		**Master to periphery module**
	Level:	6	Activates sending of filling level
	Range:	0/1	
	Page(s):	35	
4440	**CanTelActuatorPosOn**		**Periphery module to master**
	Level:	6	Activates sending of actual value for actuators
	Range:	0/1	
	Page(s):		
4441	**PEDigOutOn**		**Master to periphery module**
	Level:	6	Activates sending of binary outputs
	Range:	0/1	
	Page(s):		

No.	Name		Meaning
4441	**CanTelDigitalInOn**		**Periphery module to master**
	Level:	6	Activates sending of binary inputs
	Range:	0/1	
	Page(s):	35	
4442	**PEAnalogOutOn**		**Master to periphery module**
	Level:	6	Activates sending of analogue outputs
	Range:	0/1	
	Page(s):		
4442	**CanTelAnalogInOn**		**Periphery module to master**
	Level:	6	Activates sending of analogue inputs
	Range:	0/1	
	Page(s):	36	
4443	**PEPWMOutOn**		**Master to periphery module**
	Level:	6	Activates sending of PWM outputs
	Range:	0/1	
	Page(s):		
4443	**CanTelTempInOn**		**Periphery module to master**
	Level:	6	Activates sending of temperature inputs
	Range:	0/1	
	Page(s):	36	
4444	**PEErrorResetOn**		**Master to periphery module**
	Level:	6	Activates sending of ErrorReset request received from
	Range:	0/1	a communication module or the switching function 2828
	Page(s):	49	*SwitchErrorReset*
4444	**CanTelPWMInOn**		**Periphery module to master**
	Level:	6	Activates sending of PWM inputs
	Range:	0/1	
	Page(s):	37	
4445	**PEAutoResetOn**		**Master to periphery module**
	Level:	6	Activates sending of AutoReset request received
	Range:	0/1	from a communication module
	Page(s):		
4445	**CanTelSpeedOn**		**Periphery module to master**
	Level:	6	Activates sending of engine speed values
	Range:	0/1	
	Page(s):	37	
4446	**CanTelVelocityOn**		**Periphery module to master**
	Level:	6	Activates sending of travel speed
	Range:	0/1	
	Page(s):	37	
4450	**PE2FuelSetpointOn**		**Master to second periphery module**
	Level:	6	Activates sending of filling level
	Range:	0/1	
	Page(s):		
4451	**PE2DigOutOn**		**Master to second periphery module**
	Level:	6	Activates sending of binary outputs
	Range:	0/1	
	Page(s):		

No.	Name		Meaning
4452	**PE2AnalogOutOn**		**Master to second periphery module**
	Level:	6	Activates sending of analogue outputs
	Range:	0/1	
	Page(s):		
4453	**PE2PWMOutOn**		**Master to second periphery module**
	Level:	6	Activates sending of PWM outputs
	Range:	0/1	
	Page(s):		
4454	**PE2ErrorResetOn**		**Master to second periphery module**
	Level:	6	Activates sending of ErrorReset request received from
	Range:	0/1	a communication module or the switching function 2828
	Page(s):		*SwitchErrorReset*
4455	**PE2AutoResetOn**		**Master to third periphery module**
	Level:	6	Activates sending of AutoReset request received from
	Range:	0/1	a communication module
	Page(s):		
4490	**PE3FuelSetpointOn**		**Master to third periphery module**
	Level:	6	Activates sending of filling level
	Range:	0/1	
	Page(s):	44	
4491	**PE3DigOutOn**		**Master to third periphery module**
	Level:	6	Activates sending of binary outputs
	Range:	0/1	
	Page(s):	47	
4492	**PE3AnalogOutOn**		**Master to third periphery module**
	Level:	6	Activates sending of analogue outputs
	Range:	0/1	
	Page(s):	48	
4493	**PE3PWMOutOn**		**Master to third periphery module**
	Level:	6	Activates sending of PWM outputs
	Range:	0/1	
	Page(s):	48	
4494	**PE3ErrorResetOn**		**Master to third periphery module**
	Level:	6	Activates sending of ErrorReset request received from
	Range:	0/1	a communication module or the switching function 2828
	Page(s):	49	*SwitchErrorReset*
4495	**PE3AutoResetOn**		**Master to third periphery module**
	Level:	6	Activates sending of AutoReset request received from
	Range:	0/1	a communication module
	Page(s):	49	
4900	**ChanTypSetp1Ext**		
ff.	Level:	6	Channel type of setpoint adjuster 1
	Range:	0..14	= 14: reception via multipoint connection
	Page(s):	35, 37	
4904	**ChanTypBoostPress**		
	Level:	6	Channel type of boost pressure sensor
	Range:	0..14	= 14: reception via multipoint connection
	Page(s):	57	

No.	Name		Meaning
4911	**ChanTypExhaustTemp**		
	Level:	6	Channel type of exhaust temperature sensor
	Range:	0..14	= 14: reception via multipoint connection
	Page(s):	57	
4920	**CanTelTempInUsed**		**Periphery module to master**
	Level:	6	1: release for sending the measured temperatures
	Range:	0..1	to the master
	Page(s):	36	
4940	**CanTelPWMIn1Used**		**Periphery module to master**
	Level:	6	1: release for sending from PWM input 1 to the master
	Range:	0..1	
	Page(s):	37	
4950	**PEIxSetp1Ext**		
ff.	Level:	6	Index of periphery module in 404 *CanPENodeNumber*,
	Range:	0..2	which sends setpoint adjuster 1 to the master
	Page(s):	35, 37	
5232	**CanPowerPercentOn**		
	Level:	6	Activates usage of measured power received
	Range:	0/1	via multipoint connection
	Page(s):	54	
5910	**ActuatorOn**		**Periphery module to master**
5930	Level:	6	Actuator activates; send procedure to master
5940	Range:	0/1	
	Page(s):	34	
15400	**Out1:Destination**		**Only if common outputs**
ff.	Level:	6	Definition of target unit
	Range:	0..13	0: own hardware
	Page(s):	43, 45, 47, 109	2: periphery module
			10: WAGO module
			13: ICENI module
15401	**Out1:PEIx**		**Common output only**
ff.	Level:	6	Index of periphery module from 404 *CanPENodeNumber()*
	Range:	0..2	
	Page(s):	43, 45, 47	
15402	**Out1:OutputType**		**Common output only**
ff.	Level:	6	Output type
	Range:	0..3	0: analogue output
	Page(s):	44, 45, 47	1: PWM output
			2: binary output
			3: actuator output or valve control
15403	**Out1:OutputNo**		**Common output only**
ff.	Level:	6	Channel number of output (depending on type)
	Range:	0..117	
	Page(s):	44, 45, 47	
24000	***P001_(1.1)_Config***		**XIOS**
ff.	Level:	6	Selection of port type
	Range:	0..38	
	Page(s):	39, 40	

No.	Name		Meaning
24810	**ChanTypEngineStop**		
ff.	Level:	6	Assigning the channel type to a switching function
	Range:	0..15	0: own hardware
	Page(s):	35, 115	2: periphery module
			14: multipoint connection
24910	**PEIxEngineStop**		
ff.	Level:	6	Assigning the index of the periphery module in 404
	Range:	0..2	*CanPENodeNumber* to a switching function if channel type 2
	Page(s):	35	was selected in 24810 *ChanTypEngineStop*

10.4 Fields

No.	Name		Meaning
9120	**PEDigOut1:Param**		**Only if not common outputs**
ff.	Level:	6	Assigning a parameter number to a binary output
	Range:	-29999..29999	
	Page(s):	43	
9700	**PEFuelOut:*Assign***		**Only if not common outputs**
ff.	Level:	6	Assigning a parameter number to a filling output
	Range:	-29999..29999	
	Page(s):	44, 46	

11 Tables

12 Figures

13 Download of manuals

Technical manuals can be downloaded in pdf-format from our homepage:

www.heinzmann.com

If the desired manual should not be available there, send an e-mail to:

info@heinzmann.de

or write to:

HEINZMANN GmbH & Co. KG

Technische Redaktion

Am Haselbach 1

D-79677 Schönau/Germany

Please include the following information:

- your name
- name and address of your company
- e-mail or address where you want the manuals to be sent to (if different from above)
- item code (as on front page, bottom right) and title of the desired manual
  or alternatively the technical data of your HEINZMANN equipment

We would be pleased to receive your comments about the contents and presentation of our publications. Send your comments to the e-mail or address shown above please.