Heinzmann
GmbH & Co. KG
Speed Governors

Am Haselbach 1
D-79677 Schönau (Black Forest)
Germany

Phone (0 76 73) 82 08-0
Fax (0 76 73) 82 08-188

HEINZMANN®

Digital electronic speed governors

HEINZMANN-CAN
Customer Module

DG 05 007-e / 02-06

 **Achtung**	Read through the relevant manuals before installing, putting into operation and maintenance. All instructions relating to the system and its safety must be scrupulously observed.
 **Gefahr**	Failure to observe the instructions may lead to personal injuries and/or material damage.
 **Achtung! Hoch-spannung**  **Gefahr**	**Before putting into operation:** Turn off the power before installing! Use cable shielding and power connections that comply with the European Directive on Electromagnetic Compatibility. Check the functionality of existing protection and monitoring devices.
 **Gefahr**	**The following protective and monitoring devices must be mounted to prevent personal injuries and material damages:** overspeed device independent from speed governor protection against overheating **For systems with generators additionally:** overload protection protection against faulty synchronization when frequency, tension or phase differences are too high reverse power protection
 **Achtung**	**Possible reasons for overspeed:** power failure failure of the controller or peripheral units attached to it failure of servo components stiffness or sluggishness of linkage assembly
	All examples, data and other information in this manual are provided

	exclusively for the purpose of instruction and should not be used for a specific application, for which the user has to carry out independent tests and controls beforehand.
 **Gefahr**	Independent testing and control are particularly important for all applications whose faulty functioning can cause personal injuries or material damage.
	HEINZMANN excludes any implicit or explicit guarantee that the examples, data and other information provided in this manual are free of error, correspond to industrial standards or are suitable to the requirements of a specific application**.**
	HEINZMANN explicitly denies the tacit guarantee for marketability or suitability for a specific purpose, even if HEINZMANN has been alerted to such specific purpose or the manual contains a reference to such purpose**.**
	HEINZMANN declines all responsibility for direct and indirect damages, as well as concurrent or consequential damages resulting from an application of the examples, data or other information contained in this manual.
	HEINZMANN declines all responsibility for the concept and design of the technical system as a whole. This is responsibility of the operator and its planning and engineering personnel. Their range of responsibility includes the verification if the output of the devices we deliver is sufficient for the intended purpose. The operator is also responsible for putting the technical installation as a whole into operation correctly.

About this manual

Version	Changes effected	Date	Author
1.00	written	07.12.05	Sz
1.01	Sensors Slidereduction, TractionVoltage, TractionCurrent, TractionPower implemented	24.11.06	Sz

Contents

Page

1 HEINZMANN-CAN protocol

The HEINZMANN-CAN protocol is based on the CAN specification 2.0B with a 29 bit identifier.

The identifier contains information about sender and receiver and the command code. The maximum 8 data bytes are therefore available completely for operative data.

1.1 Identifier structure

Bit range	Code	Meaning	Value
28..27	p	priority	always 2
26..23	d	type of receiving device (destination)	0..15
22..18	m	destination node number	0, 1..31
17	r	reserved	always 0
16..13	s	type of sending device (source)	0..15
12..8	n	source node number	1..31
7..0	c	command	0..255

Each connection in a HEINZMANN-CAN network is therefore a point-to-point connection. A telegram is sent by a unique source to a unique destination. An exception is the transmission of a command to all units of the same type using node number 0, but this command is not used in the context of the customer module.

1.2 Node types

Sources and destinations are subdivided in node type (device type) and a node number. The following device types are configured for communication with the customer module.

Device type	Code	Control device
0	DC	speed governor conventional or direct injector
1	GC	Theseus

The customer module itself is of the following type:

Device type	Code	Control device
6	CM	Customer Module

Communication in CAN is established only if the respective device type is authorized:

4400 *CanCommDCOn* = 1 must be set in the speed governor

4401 *CanCommGCOn* = 1 must be set in Theseus

4406 *CanCommCMOn* = 1 must be set in the control device, if a communication to a customer module must be established.

1.3 Node numbers

For each device type in the HEINZMANN-CAN network each node number in the range between 1 and 31 must be assigned only once. Node number 0 is not allowed for a single device, since it is used as the number for messages to all nodes of a certain type.
The node number of the control device (DC or GC) is defined in 401 *CanMyNodeNumber* and the node number of the customer module with which the control device must work is entered in 403 *CanCMNodeNumber*.

1.4 Commands

The possible command codes of the telegrams and the respective data content are described at length in ↑ *3* Telegram structure. The command codes always relate both to the connection and the direction between two device types, which means that the same command numbers may have a different meaning when used for connections between different types and in different directions.

1.5 Identifier for the customer module

The basic structure of the 29-bit form of the identifier in connection with the customer module is as follows. mmmmm are the five bits for the node number of the receiving device, nnnnn is the node number of the sending device and cccccccc is the command code.

Direction	**p**	**d**	**m**	**r**	**s**	**n**	**c**	**Identifier**
CM→DC	2	0	m	0	6	n	c	10 0000 mmmmm 0 0110 nnnnn cccccccc
CM→GC	2	1	m	0	6	n	c	10 0001 mmmmm 0 0110 nnnnn cccccccc
DC→CM	2	6	m	0	0	n	c	10 0110 mmmmm 0 0000 nnnnn cccccccc
GC→CM	2	6	m	0	1	n	c	10 0110 mmmmm 0 0001 nnnnn cccccccc

1.6 Baud rate

The selected baud rate can be set in the HEINZMANN device, the differences between the various types of CAN controller must be taken into account.

The control devices HELENOS, PRIAMOS, KRONOS 30, DARDANOS I (MVC 01-10/20) and THESEUS work with the Intel or Philips CAN-controller.

The control devices ARCHIMEDES, GMU, HELIOS 2, KRONOS 20 and PANDAROS use a Mitsubishi M16C controller.

The control devices DARDANOS II (MVC03), DARDANOS III (MVC04), ARIADNE and APOLLON use the TOUCAN controller.

The function parameter 4416 *CanSegmentOrBaudrate* determines whether to work with the baud rate 416 *CanBaudrate* (4416 *CanSegmentOrBaudrate* = 0) or with the segment settings in the parameters 410 *CanPrescaler* to 415 *CanPropSegment* (4416 *CanSegmentOrBaudrate* = 1).

In 416 *CanBaudrate,* only the four indicated values are admissible as baud rates, for every other entry 250 kBaud will be used. To these four values the segment settings listed in the following tables are stably assigned.

If another baud rate should be necessary or the segment settings have to be changed because of the sampling moment or the cable length, you will have to work with the segment settings (4416 *CanSegmentOrBaudrate* = 1).

Parameter	**125 kbaud**			**250 kbaud**		
	Intel/ Philips	**M16C**	**TOUCAN**	**Intel/ Philips**	**M16C**	**TOUCAN**
410 *CanPrescaler*	3	3	27	1	1	13
411 *CanSyncJumpWidth*	2	2	0	2	2	0
412 *CanSamplingMode*	0	0	0	0	0	0
413 *CanPhaseSegment1*	10	2	3	10	2	3
414 *CanPhaseSegment2*	3	3	7	3	3	7
415 *CanPropSegment*	-	7	2	-	7	2
416 *CanBaudrate*	125			250		
4416 *CanSegmentOrBaudrate*	0/1			0/1		

Parameter	500 kBaud			1 MBaud		
	Intel/ Philips	M16C	TOUCAN	Intel/ Philips	M16C	TOUCAN
410 *CanPrescaler*	0	0	6	0	0	3
411 *CanSyncJumpWidth*	2	2	0	0	0	0
412 *CanSamplingMode*	0	0	0	0	0	0
413 *CanPhaseSegment1*	10	2	3	4	0	1
414 *CanPhaseSegment2*	3	3	7	1	1	7
415 *CanPropSegment*	-	7	2	-	3	2
416 *CanBaudrate*	500			1000		
4416 *CanSegmentOrBaudrate*	0/1			0/1		

1.7 Monitoring the CAN communication

The communication is constantly monitored. After the control device is switched on, a certain amount of time may pass before an error message is originated. This delay is set in parameter 400 *CanStartTimeOutDelay*. All devices connected to the CAN network should have the same delay setting. The whole network must be fed tension during this time, in order not to trigger an error message during start-up.

The parameters 2422 *CanCMNodeState31to16* and 2423 *CanCMNodeState15to01* show if a connection between the control device and a customer module has been established. The bit corresponding to the node number of the customer module is activated in the process.

The following general error messages are generated:

3070 *ErrCanBus* Error CAN bus

3071 *ErrCanComm* Error CAN communication

In case of a CAN bus error, the CAN controller emits errors such as BusStatus, ErrorStatus oder DataOverrun. Even if the controller is initialized again, the errors are not eliminated durably. The reason usually is faulty cabling, lack of termination or different baud rates used by the users connected to the network. The control device tries to reach an error-free state by keeping to initialize the CAN controller.

As opposed to that, the CAN communication error 3071 *ErrCanComm* relates to an error in content, i.e. there is no physical error in the network and communication is in principle possible. Information about the communication errors of the HEINZMANN-CAN bus is shown in the following parameters:

2401	*CanTxBufferState*	state of source buffer
2402	*CanRxBufferState*	state of destination buffer
2403	*CanRxTimeout*	state of destination timeout monitoring
2404	*CanTypeMismatch*	state of device numbers

The values of the parameters 2401 to 2404 are in binary code and the bit number relates to the device type. An activation of these parameters triggers off a 3071 *ErrCanComm* error.

Source and destination buffer are monitored for overflow and indicated in parameters 2401 *CanTxBufferState* und 2402 *CanRxBufferState.* The receipt of messages must be completed within a fixed amount of time, otherwise error 2403 *CanRxTimeout* is activated. The error 2404 *CanTypeMismatch* finally indicates a configuration error. A second user with the same device number and the same device type is connected to the network. In case of source or destination buffer overflow, the overflow is only indicated and the communication proceeds but for one or several messages that cannot be received or sent. If too many messages cannot be received, error 2403 *CanRx-Timeout* is triggered. If a source buffer overflow causes messages to be withheld, the other side indicates the timeout error.

The error 2403 *CanRxTimeout* is generally activated when the other side cannot be reached. In this case messages keep getting sent to the other side, but the content switches to certain emergency procedures.

Whether the control device is generally ready to communicate via CAN is indicated by parameter 2405 *CanOnline* .

2 Customer Module

To the control devices, the functions described in the *Basic Informations* apply. As an additional extension, the CAN bus allows to exchange data between a HEINZMANN control device and a customer module. Besides the pre-defined telegrams containing pre-defined data there are three more parametrizable telegrams, which allow to send freely chosen data from the HEINZMANN device to the customer module.

Note: Parameter changes relating to the customer module become active only after saving and reset of the control device.

In order to send and receive telegrams between a HEINZMANN control device and a customer module, the parameter 4406 *CanCommCMOn* = 1 must be set.

2.1 Receiving data

The delay time for timeout monitoring for arriving telegrams is set in 21950 *CMRxTel10Timeout* to 21954 *CMRxTel23Timeout*. If a telegram was not received during this span of time, error 3071 *ErrCanComm* is activated. If the timeout parameters are set to 0, there is no specific timeout monitoring for the relative telegram.

Besides that, there is always a general timeout monitoring for all CAN communication between control device and customer module. The customer module must, just as the control device, transmit a signal at least once every second.

2.1.1 Switch functions

All switch functions defined in the control device may be received by way of a dedicated hardware input or by way of the customer module's telegram 10 (↑ *3.1.1 Switch functions)*. The receipt path must be communicated to the control device.

In order for the control device to be able to use the switch functions received by way of telegram 10, in 24810 *ChanTyp...* to 24849 *ChanTyp...*, the value 8 must be entered to indicate the chosen channel type. If the receipt is to happen exclusively by way of the hardware, channel type 0 must be chosen.

If channel type 8 (customer module) is chosen, the bit number in telegram 10 must be indicated in the corresponding parameter 20810 *Comm...* to 20849 *Comm....* Up to 32 different switch functions may be transmitted with telegram 10. They are chosen and assigned by the programmer of the customer module.

If one of the switch functions received by the customer module is to be given additional cabling, the number of the digital input used for the purpose must be indicated in parameter 810 *Funct...* to 849 *Funct...* If this parameter is set to 0, the switch function is received only via CAN.

If channel type 0 (own hardware) is chosen, only the number of the digital input used for the purpose must be indicated in parameter 810 *Funct...* to 849 *Funct...* The input number 0 amounts to saying "not used".

8xx *Funct...* = DI-Nr.	<> 0: redundant cabling, 0: no cabling
248xx *ChanTyp...* = 8	switch is received via CAN customer module
208xx *Comm...* = Bitnr.	bit number in telegram 10 (0, 1…32)

The bit number counts bytewise, i.e. the first data byte of the telegram contains the bits 1…8 (LSB..MSB), the second the bits 9..16 (LSB..MSB), and so on. The bit number 0 amounts to saying "not used".

A switch function is activated if it addressed by at least one of the two possible sources (digital input or telegram 10).

The value "1" in telegram 10 switches a function ON, the value "0" switches it OFF: Switch functions serving as toggle commands are defined as follows: "1" for the state indicated to the left of "OR" in the name and "0" for the state to the right of "OR". Example: In switch function 2827 *SwitchSetpoint2Or1* the transmission of "1" activates speed set point 2, "0" activates speed set point 1.

Note: DcDesk 2000 version 5.06 upwards supports the configuration of switch functions by communication modules via menu item "Control unit/Adjustment/Switch functions".

2.1.1.1 Error in the configuration or in CAN receipt of switch functions

If the switch functions 248xx *ChanTyp....* are set = 8, but the customer module is not activated with 4406 *CanCommCMOn* = 1, all switch functions are reset to 0 and the error message 3000 *ConfigurationError* is sent.

If there is a CAN error, either a bus error, downtime of the customer module or a timeout of telegram 10, all switch functions assigned to the CAN customer module are equally returned to 0. If the telegram is received again, switch functions are determined again by way of CAN.

2.1.1.2 Switch function motor stop in the speed governor (DC)

In case of a CAN error, the switches determined via CAN are deleted or reset to zero. If in this case a "motor stop signal" transmitted via CAN is to lead to a motor stop in any case, the parameter 4810 *StopImpulseOrSwitch* must be set to 1. This parameter allows to define whether an external stop command remains active only during the time the command is explicitly active or if an impulse is sufficient to keep the command active until the motor has stopped.

4810 *StopImpulseOrSwitch* = 1	motor stop active only if the stop command is explicitly active
4810 *StopImpulseOrSwitch*= 0	a single switch impulse is sufficient to keep the command active until the motor has stopped.

2.1.2 Sensors

Each sensor defined in the control device may be received either by way of a dedicated hardware input or by way of the customer module's telegrams 20 to 23 (↑ *3.1.2 Sensors*). The receipt path must be communicated to the control device.

In order for the control device to be able to use the sensor values received by way of the telegrams 20 to 23, the value 8 must be set in 4900 *ChanTyp...* to 4924 *ChanTyp...* to indicate the chosen channel type. For receipt by way of an analog input on the hardware, channel type "0" must be entered and for receipt by way of a PWM input on the hardware, channel type "1" must be used.

In the corresponding parameter 900 *AssignIn_...* to 924 *AssignIn_...* the number of the input channel must be entered. Channel number 0 amounts to saying "not used".

49xx *ChanTyp...* = 8 sensor is received via CAN customer module

9xx *AssignIn_...* = channel number channel number in telegrams 20..23 (0, 1..16)

Up to 16 different sensors may be transmitted with telegrams 20..23. They are chosen and assigned by the programmer of the customer module. The channel numbers in the telegrams count word for word, i.e. the first word in telegram 20 defines channel 1, the second word channel 2, and so on. The fourth word of telegram 23 has channel number 16.

Note: DcDesk 2000 version 5.06 upwards supports the configuration of sensors by communication modules via menu item "Control unit/Adjustment/Sensors".

Parametrizing example for DC

You want current boost pressure and coolant temperature to be received every 50 ms via words 1 and 2 of telegram 20.

Number	Parameter	Value	Unit
904	*AssignIn_BoostPressure*	1	
907	*AssignIn_CoolantTemp*	2	
4406	*CanCommCMOn*	1	
4904	*ChanType_BoostPress*	8	
4907	*ChanType_CoolantTemp*	8	
21951	*CMRxTel20Timeout*	0,05	s

2.1.2.1 Error in the configuration or in CAN receipt of sensors

If the sensors 49xx *ChanTyp...* are set = 8 but the customer module is not activated with 4406 *CanCommCMOn* = 1, all sensor values are reset to zero and the error message 3000 *ConfigurationError* is transmitted.

If there is a CAN error, either a bus error, downtime of the customer module or a timeout of one of the telegrams 20..23, all sensors assigned to the CAN customer module are equally returned to 0. If the telegram is received again, sensors values are transmitted again by way of CAN.

The effective value of sensors in case of error depends on the settings in 5000 *SubstOrLast...* to 5024 *SubstOrLast...* and 5040 *HoldOrReset...* to *5064 HoldOrRese....*

500x *SubstOrLast...* = 1 substitution value 1000 *Subst...* is used

500x *SubstOrLast...* = 0 last valid value is used

After the return of the CAN signal, the sensor error caused by a CAN error may be kept until the error is reset or until the error disappears, depending on the setting of parameter 504x *HoldOrReset..*

504x *HoldOrReset...* = 1 sensor error kept for error reset

504x *HoldOrReset...* = 0 sensor error deletes itself when error cause disappears.

2.1.3 Requesting parameter values

With the request telegram 80 (↑ *3.1.3 Requesting parameter values)* the customer module can request the one-time-only transmission of up to four parameter values. This results in the transmission of up to four parameter numbers, that must exist in the control device and cannot have an agreed level higher than 4. The values of these parameters

are then sent back in answer telegram 80 (↑*3.2.6 Answer to request of parameter values*).

The answer telegram is a sensible option when data are not required continuously, such as hours of operation or certain parameter settings.

2.1.4 Request of a send telegram

The request telegram 81 (↑*2.1.4 Request of a send telegram*) allows to transmit the number of one of the defined send telegrams of the control device. This telegram is then sent once. This is useful when the data is not required continuously and the bus load can be reduced using this procedure. The requested telegram is sent even if it has not been activated via 25960 *CMTxTel20On* .. 25987 *CMTxTel50*..

2.2 Data transmission

For each telegram to be sent, the respective activation parameter in the range 25960 *CMTxTel20On* .. 25987 *CMTxTel52On* must be set.:

259xx *CMTxTel..On* = 1 send telegram

259xx *CMTxTel..On* = 0 do not send telegram

All telegrams except the error telegrams are send regularly at the time interval set in the transmission rate 21960 *CMTxTel20SendRate* to 21987 *CMTxTel52SendRate.* If "0" is set, telegrams are sent at every pass, i.e. with Intel- or Philips-CAN controllers each 15,625 ms, with M16C systems each 16 ms and with TOUCAN systems each 10 ms. Because of the possibly resulting high bus load it should be considered with care if and for which telegrams this is really necessary.

2.2.1 Current errors

The up to five error telegrams 41 to 45 (↑ *3.2.5 Configurable telegrams*) transmit the current error states in the control device. After the first transmission occurring immediately after the connection is established, they are sent again only if at least one error state changes in the corresponding telegram.

2.2.2 Freely configurable telegrams

For the three freely configurable telegrams 50, 51 and 52 (↑ *3.2.4 Current errors*) up to four parameter numbers of data each can be defined for regular transmission to the customer module using the parameters 29800 *CMTel50ParamSet()* to 29813 *CMTel52ParamSet()*. The four field elements must be filled one after the other, beginning with index 0. The current values of these parameters are sent back in the same places in send telegram 80. Basically words are transmitted, even when a single parameter contains only a byte or bit value.

The input of parameter number 0 results in an entry of 0 in the corresponding place in the send telegram. Send telegram length results from the number of valid telegram entries, meaning that the data transmission ends with the first invalid, non-existing parameter number.

2.2.3 Transmission of bit values

To transmit several bit values in compressed form, a bit collection may be assembled using field 29900 *BitCollParamSet()*. Here you enter the parameter numbers of bit values of the control unit that can assume only the values 0 or 1. In each field index a positive or negative parameter number or 0 may be entered.

With the values of the indicated parameters words are formed which are shown in 23720 *BitCollection()*. The first 16 entries of 29900 *BitCollParamSet()* form the first word of 23720 *BitCollection()*; the second 16 entries form the second word, and so on. The value of the parameter in index 0 yields bit 0, the value in parameter 1 becomes bit 1, and so on.

When the parameter number is positive, the value of the parameter is included in the bit collection. When the parameter number is negative, the negated parameter value is included in the bit collection. If a zero is entered, the value 0 appears in the bit collection.

Parameter numbers starting from 23720 *BitCollection()* can be entered again in an element of the fields 29800 *CMTel50ParamSet()* to 29813 *CMTel52ParamSet()*, in order to send a bit collection to the customer module.

Note: DcDesk 2000 version 5.06 upwards supports the configuration of bit collections for communication modules via menu item "Control unit/Adjustment/Bit collection".

Parametrizing example for DC

Boost pressure, oil temperature, rotational speed and the causes of a capacity limitation are to be transmitted each 100 ms by way of a freely parametrizable telegram.

Number	Parameter	Value	Parameter
21971	*CMTxTel50SendRate*	0.1 s	
4406	*CanCommCMOn*	1	
25971	*CMTxTel50On*	1	
29800	*CMTel50ParamSet(0)*	2904	*BoostPressure*
29801	*CMTel50ParamSet(1)*	2909	*OilTemp*
29802	*CMTel50ParamSet(2)*	2000	*Speed*
29803	*CMTel50ParamSet(3)*	23720	*BitCollection*
29900	*BitCollParamSet(0)*	2711	*FuelLimitMaxActive*
29901	*BitCollParamSet(1)*	2712	*StartLimitActive*
29902	*BitCollParamSet(2)*	2713	*SpeedLimitActive*
29903	*BitCollParamSet(3)*	2714	*BoostLimitActive*
29904	*BitCollParamSet(4)*	2715	*ForcedLimitActive*

3 Telegram structure

The following section describes in detail all the telegrams used.

3.1 Received telegrams (CM → DC/GC)

A control device of the type speed governor (DC) or Theseus (GC) can receive the following telegrams from the customer module.

3.1.1 Switch functions

Command: 10

Data bytes: 1, 2, 3 or 4

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 00001010
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 00001010

Timeout: 21950 *CMRxTel10Timeout*

Activation: automatic if 24810ff *ChanTyp*... = 8 receives a switch function by way of HZM-CAN CM

		byte 0	byte 1	byte 2	byte 3
bit	7	Switch function 8	Switch function 16	Switch function 24	Switch function 32
	6	Switch function 7	Switch function 15	Switch function 23	Switch function 31
	5	Switch function 6	Switch function 14	Switch function 22	Switch function 30
	4	Switch function 5	Switch function 13	Switch function 21	Switch function 29
	3	Switch function 4	Switch function 12	Switch function 20	Switch function 28
	2	Switch function 3	Switch function 11	Switch function 19	Switch function 27
	1	Switch function 2	Switch function 10	Switch function 18	Switch function 26
	0	Switch function 1	Switch function 9	Switch function 17	Switch function 25

Each bit in this telegram corresponds to the value of a switch function currently to be set.

Switch functions may be assigned to the single bits by the programmer of the customer module. It is important that during the configuration of the control device the correct bit number is assigned to the respective parameter in 20810 *Comm*... to 20849 *Comm*... (↑ *2.1.1 Switch functions*).

Of the maximum of four data bytes only the part necessary for the switch functions must be transmitted.

3.1.2 Sensors

Sensor values are received by the customer module by way of up to four telegrams with four values each, amounting to 16 sensor values available in total.

The various sensors may be assigned to the data words by the programmer of the customer module. It is important to remember that during the configuration of the control device the correct channel number must be assigned to the respective parameter in 900 *AssignIn_...* to *924 AssignIn_...* (↑ *2.1.2 Sensors*).

Each sensor value must be sent within the internal value range of the control device (↑ *4.1 Value range of sensors*). Of the available maximum of four data words only the required ones must be sent (2, 4, 6 or 8 bytes).

Command: 20

Data bytes: 2, 4, 6 or 8

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 00010100
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 00010100

Timeout: 21951 *CMRxTel20Timeout*

Activation: automatic if 4900ff*ChanTyp...* = 8 and 900 *AssignIn_...* = 1..4 receives one of these sensors by way of HZM-CAN CM

	byte 0	**byte 1**	**byte 2**	**byte 3**	**byte 4**	**byte 5**	**byte 6**	**byte 7**
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Channel	Sensor 1		Sensor 2		Sensor 3		Sensor 4	

Command: 21

Data bytes: 2, 4, 6 or 8

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 00010101
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 00010101

Timeout: 21952 *CMRxTel21Timeout*

Activation: automatic if 4900ff*ChanTyp...* = 8 and 900 *AssignIn_...* = [5..8] receives one of these sensors by way of HZM-CAN CM

	byte 0	**byte 1**	**byte 2**	**byte 3**	**byte 4**	**byte 5**	**byte 6**	**byte 7**
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Channel	Sensor 5		Sensor 6		Sensor 7		Sensor 8	

Command: 22

Data bytes: 2, 4, 6 or 8

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 00010110
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 00010110

Timeout: 21953 *CMRxTel22Timeout*

Activation: automatic if 4900ff*ChanTyp*... = 8 and 900 *AssignIn_*... = [9..12] receives one of these sensors by way of HZM-CAN CM

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Channel	Sensor 9		Sensor 10		Sensor 11		Sensor 12	

Command: 23

Data bytes: 2, 4, 6 or 8

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 00010111
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 00010111

Timeout: 21953 *CMRxTel23Timeout*

Activation: automatic if 4900ff*ChanTyp*... = 8 and 900 *AssignIn_*... = [13..16] receives one of these sensors
by way of HZM-CAN CM

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Channel	Sensor 13		Sensor 14		Sensor 15		Sensor 16	

3.1.3 Requesting parameter values

Command: 80

Data bytes: 2, 4, 6 or 8

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 01010000
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 01010000

With a request by way of command 80 (↑ *2.1.3 Requesting parameter values*) up to four parameters may be requested by indicating their parameter number. The parameter can be shorter, if less than four parameter numbers are required.

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	Parameter number 1		Parameter number 2		Parameter number 3		Parameter number 4	

The control device sends the requested parameter values with the answer telegram 80 (↑ *3.2.6 Answer to request of parameter values*). Only the parameters with an existing number and a level defined as not higher than 4 are accepted.

The customer module can send a new request telegram only after having received the answer to the preceding one.

3.1.4 Request of a send telegram

Command: 81

Data bytes: 1

Identifier: CM → DC: 10 0000 mmmmm 0 0110 nnnnn 01010001
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 01010001

	byte 0
Value	Telegram number

With the request telegram 81 (↑ *3.1.4 Request of a send telegram*) the customer module transmits one of the send telegram numbers of the control device and in doing so activates a single sending of the corresponding telegram.

3.2 Send telegrams (DC/GC → CM)

A control device of the type speed governor (DC) or Theseus (GC) can send the following telegrams to the customer module.

3.2.1 Sensors

A maximum of four telegrams with a total of 16 pre-defined sensor values may be sent to the customer module.

Each sensor value must be sent within the internal value range (↑ *4.1 Value range of sensors*). In place of sensor values not available in the specific control device, the value 0 is transmitted.

Command: 20

Data bytes: 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00010100

Sender rate: 21960 *CMTxTel20SendRate*

Activation: 25960 *CMTxTel20On* = 1

	byte 0	**byte 1**	**byte 2**	**byte 3**	**byte 4**	**byte 5**	**byte 6**	**byte 7**
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	2900 Setpoint1		2901 Setpoint2		2918 MeasuredPower		2919 PowerSetpoint	

Command: 21

Data bytes: 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00010101
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00010101

Sender rate: 21961 *CMTxTel21SendRate*

Activation: 25961 *CMTxTel21On* = 1

	byte 0	**byte 1**	**byte 2**	**byte 3**	**byte 4**	**byte 5**	**byte 6**	**byte 7**
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	Boost Pressure		OilPressure		Ambient Pressure		Coolant Pressure	

Command: 22

Data bytes: 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00010110
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00010110

Sender rate: 21962 *CMTxTel22SendRate*

Activation: 25962 *CMTxTel22On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	CoolantTemp		ChargeAirTemp		OilTemp		ExhaustTemp	

Command: 23

Data bytes: 2

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00010111

Sender rate: 21963 *CMTxTel23SendRate*

Activation: 25963 *CMTxTel23On* = 1

	byte 0	byte 1
	high-byte	low-byte
Value	FuelTemp	

3.2.2 Rotational speed and fuel quantity

Command: 30

Data bytes: 6 or 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00011110
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00011110

Sender rate: 21964 *CMTxTel30SendRate*

Activation: 25964 *CMTxTel30On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	2000 Speed		2031 Speed Setpoint		2350 FuelQuantity		2300 ActPos	

All values are sent within the internal value range (↑ *4.2 Value range of measured and indicated values)*.

The position of actuators is sent only by conventional spe1ed governors, electrovalve-guided systems and Theseus transmit only 6 bytes.

If other measurement values are not available in the specific control device, the value 0 is transmitted.

3.2.3 Alarm and motor state

Command: 40

Data bytes: 1 or 2

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00101000
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00101000

Sender rate: 21965 *CMTxTel40SendRate*

Activation: 25965 *CMTxTel40On* = 1

		byte 0	byte 1
bit	7		
	6		
	5		
	4		3806 EngineReleased
	3		3805 EngineRunning
	2		3804 EngineStarting
	1	3801 Common-Alarm	3803 EngineStopped
	0	3800 Emergency-Alarm	3802 EngineStopRequest

The speed governor (DC) transmits both bytes. Theseus (GC) transmits the second byte only if working with the integrated motor speed governor.

3.2.4 Current errors

After connection is established by way of telegram 97, all activated error telegrams with the current state of the corresponding error bits are sent once (↑ *2.2.1 Current errors*). Subsequently they are transmitted only if at least one error bit has changed sinde the last transmission.

The meaning of the error bits is described in the manuals of the respective control device. In place on non-existing error numbers a 0 is sent.

Command: 41

Data bytes: 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00101001
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00101001

Sender rate: 21966 *CMTxTel41SendRate*

Activation: 25966 *CMTxTel41On* = 1

		byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
bit	7	free	3087	3079	3071	3063	3055	3047	3039
	6	3094	3086	3078	3070	3062	3054	3046	3038
	5	3093	3085	3077	3069	3061	3053	3045	3037
	4	3092	3084	3076	3068	3060	3052	3044	3036
	3	3091	3083	3075	3067	3059	3051	3043	3035
	2	3090	3082	3074	3066	3058	3050	3042	3034
	1	3089	3081	3073	3065	3057	3049	3041	3033
	0	3088	3080	3072	3064	3056	3048	3040	3032

Command: 42

Data bytes: 4 or 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00101010
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00101010

Sender rate: 21967 *CMTxTel42SendRate*

Activation: 25967 *CMTxTel42On* = 1

Bytes 4 to 7 are sent only if the control device has defined error numbers in the range 13000 to 13095, otherwise the telegram has length 4.

		byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
bit	7	3031	3023	3015	3007	13095	13087	13079	13071
	6	3030	3022	3014	3006	13094	13086	13078	13070
	5	3029	3021	3013	3005	13093	13085	13077	13069
	4	3028	3020	3012	3004	13092	13084	13076	13068
	3	3027	3019	3011	3003	13091	13083	13075	13067
	2	3026	3018	3010	3002	13090	13082	13074	13066
	1	3025	3017	3009	3001	13089	13081	13073	13065
	0	3024	3016	3008	3000	13088	13080	13072	13064

Command: 43

Data bytes: 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00101011
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00101011

Sender rate: 21968 *CMTxTel43SendRate*

Activation: 25968 *CMTxTel43On* = 1

This telegram is sent only if the control device has defined error numbers in the range 13000 to 13095.

		byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
bit	**7**	13063	13055	13047	13039	13031	13023	13015	13007
	6	13062	13054	13046	13038	13030	13022	13014	13006
	5	13061	13053	13045	13037	13029	13021	13013	13005
	4	13060	13052	13044	13036	13028	13020	13012	13004
	3	13059	13051	13043	13035	13027	13019	13011	13003
	2	13058	13050	13042	13034	13026	13018	13010	13002
	1	13057	13049	13041	13033	13025	13017	13009	13001
	0	13056	13048	13040	13032	13024	13016	13008	13000

Command: 44

Data bytes: 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00101100
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00101100

Sender rate: 21969 *CMTxTel44SendRate*

Activation: 25969 *CMTxTel44On* = 1

This telegram is sent only if the control device has defined error numbers in the range 23000 to 23095.

		byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
bit	7	23095	23087	23079	23071	23063	23055	23047	23039
	6	23094	23086	23078	23070	23062	23054	23046	23038
	5	23093	23085	23077	23069	23061	23053	23045	23037
	4	23092	23084	23076	23068	23060	23052	23044	23036
	3	23091	23083	23075	23067	23059	23051	23043	23035
	2	23090	23082	23074	23066	23058	23050	23042	23034
	1	23089	23081	23073	23065	23057	23049	23041	23033
	0	23088	23080	23072	23064	23056	23048	23040	23032

Command: 45

Data bytes: 4

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00101101
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00101101

Sender rate: 21970 *CMTxTel45SendRate*

Activation: 25970 *CMTxTel45On* = 1

This telegram is sent only if the control device has defined error numbers in the range 23000 to 23095.

		byte 0	byte 1	byte 2	byte 3
bit	7	23031	23023	23015	23007
	6	23030	23022	23014	23006
	5	23029	23021	23013	23005
	4	23028	23020	23012	23004
	3	23027	23019	23011	23003
	2	23026	23018	23010	23002
	1	23025	23017	23009	23001
	0	23024	23016	23008	23000

3.2.5 Configurable telegrams

Configurable telegrams send the value of the parameters entered in 29800 CMTel50ParamSet(0) to 29812 CMTel52ParamSet(3) (↑ *2.2.2 Freely configurable telegrams*).

All values are sent in the external value range. This range is defined for each parameter and in shown in the basic information for the specific control device and in DcDesk 2000. If places after the decimal point have been defined, for the transmission these are solved by multiplication with powers of ten.

Command: 50

Data bytes: 2, 4, 6 or 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00110010
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00110010

Sender rate: 21971 *CMTxTel50SendRate*

Activation: 25971 *CMTxTel50On* = 1

	byte 0		byte 1		byte 2		byte 3		byte 4		byte 5		byte 6		byte 7	
	high-byte		low-byte		high-byte		low-byte		high-byte		low-byte		high-byte		low-byte	
Value	Value of 29800 CMTel50ParamSet(0)				Value of 29801 CMTel50ParamSet(1)				Value of 29801 CMTel50ParamSet(1)				Value of 29801 CMTel50ParamSet(1)			

Command: 51

Data bytes: 2, 4, 6 or 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00110011
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00110011

Sender rate: 21972 *CMTxTel51SendRate*

Activation: 25972 *CMTxTel51On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	Value of 29805 CMTel51ParamSet(0)		Value of 29805 CMTel51ParamSet(0)		Value of 29805 CMTel51ParamSet(0)		Value of 29805 CMTel51ParamSet(0)	

Command: 52

Data bytes: 2, 4, 6 or 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 00110100
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00110100

Sender rate: 21973 *CMTxTel52SendRate*

Activation: 25973 *CMTxTel52On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	Value of 29810 CMTel52ParamSet(0)		Value of 29810 CMTel52ParamSet(0)		Value of 29810 CMTel52ParamSet(0)		Value of 29810 CMTel52ParamSet(0)	

3.2.6 Answer to request of parameter values

Only the parameters of the request telegram 80 (↑ *3.1.3 Requesting parameter values)* with an existing number and a level defined as not higher than 4 are accepted. The answer telegram 80 transmits the current values of these parameters.

All values are sent in the external value range. This range is defined for each parameter and in shown in the basic information for the specific control device and in DcDesk 2000. If places after the decimal point have been defined, for the transmission these are solved by multiplication with powers of ten.

Command: 80

Data bytes: 2, 4, 6 or 8

Identifier: DC → CM: 10 0110 mmmmm 0 0000 nnnnn 01010000
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 01010000

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	Value of parameter nr. 1		Value of parameter nr.2		Value of parameter nr.3		Value of parameter nr.4	

3.3 Theseus send telegrams (GC → CM)

All values are sent within the internal value range (↑ *4.2 Value range of measured and indicated values*).

3.3.1 Mains frequency

Command: 60

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00111100

Sender rate: 21980 *CMTxTel60SendRate*

Activation: 25980 *CMTxTel60On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	12001 FrequencyNet_L1		12002 FrequencyNet_L2		12003 FrequencyNet_L3	

3.3.2 Generator frequency

Command: 61

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00111101

Sender rate: 21981 *CMTxTel61SendRate*

Activation: 25981 *CMTxTel61On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	12011 FrequencyGeneratorL1		12012 FrequencyGeneratorL2		12013 FrequencyGeneratorL3	

3.3.3 Bus voltage

Command: 62

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00111110

Sender rate: 21982 *CMTxTel62SendRate*

Activation: 25982 *CMTxTel62On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	12107 VoltageBusPrim_1_2		12108 VoltageBusPrim_2_3		12109 VoltageBusPrim_3_1	

3.3.4 Generator voltage

Command: 63

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 00111111

Sender rate: 21983 *CMTxTel63SendRate*

Activation: 25983 *CMTxTel63On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	12127 VoltageGenPrim_1_2		12128 VoltageGenPrim_2_3		12129 VoltageGenPrim_3_1	

3.3.5 Primary current

Command: 64

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 01000000

Sender rate: 21984 *CMTxTel64SendRate*

Activation: 25984 *CMTxTel64On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	12147 CurrentPrim_L1		12148 CurrentPrim_L2		12149 CurrentPrim_L3	

3.3.6 Power

Command: 65

Data bytes: 8

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 01000001

Sender rate: 21985 *CMTxTel65SendRate*

Activation: 25985 *CMTxTel65On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	12208 PowerPrim		12209 PowerReactivePrim		12210 PowerApparenPrim		12203 cosPhi	

3.3.7 Active power

Command: 66

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 01000010

Sender rate: 21986 *CMTxTel66SendRate*

Activation: 25986 *CMTxTel66On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	13700 Power_GWh		13701 Power_MWh		13702 Power_kWh	

3.3.8 Reactive power

Command: 67

Data bytes: 6

Identifier: GC → CM: 10 0110 mmmmm 0 0001 nnnnn 01000011

Sender rate: 21987 *CMTxTel67SendRate*

Activation: 25987 *CMTxTel67On* = 1

	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5
	high-byte	low-byte	high-byte	low-byte	high-byte	low-byte
Value	13704 PowerReactive_GWh		13705 PowerReactive_MWh		13706 PowerImpulseReac_kWh	

3.4 Special telegrams

3.4.1 Connection establishment

Command: 97

Data bytes: none

Identifier:
CM → DC: 10 0000 mmmmm 0 0110 nnnnn 1100001
CM → GC: 10 0001 mmmmm 0 0110 nnnnn 1100001
DC → CM: 10 0110 mmmmm 0 0000 nnnnn 1100001
GC → CM: 10 0110 mmmmm 0 0001 nnnnn 1100001

After the duplicate ID check (command 98) is concluded, both the customer module and the counterpart keep on sending command 97 without data bytes (with data length 0) until any command is received from the other side. This has the purpose to ensure commands with real data are sent to the other side only after the other side is initialized and has started the communication.

3.4.2 Duplicate ID check

Command: 98

Data bytes: 1 byte, value 1

Identifier: CM → CM: 10 0000 mmmmm 0 0110 nnnnn 1100010

To test the CAN bus parameters, during the phase of initialization each node sends a duplicate ID check telegram command 98 with data byte =1 to its own node type and to its own node number.

3.4.3 Answer to duplicate ID check

Command: 98

Data bytes: 1 byte, value 0

Identifier: CM → CM: 10 0000 mmmmm 0 0110 nnnnn 1100010

Each connected device receiving a duplicate ID check telegram because it is of the same type and has the same node number as the sender must reply to it with command 98 and data byte = 0. As a result, both devices – the sender and the receiver – leave the bus to prevent errors.

3.4.4 Life sign

Command: 99

Data bytes: none

Identifier:

CM → DC:	10 0000 mmmmm 0 0110 nnnnn 1100011
CM → GC:	10 0001 mmmmm 0 0110 nnnnn 1100011
DC → CM:	10 0110 mmmmm 0 0000 nnnnn 1100011
GC → CM:	10 0110 mmmmm 0 0001 nnnnn 1100011

To reduce bus load to a minimum, telegrams should be sent only when new information has to be transmitted. If no other telegram has to be sent, the life sign must be sent every second, thereby giving the other side the possibility to recognize if there is a downtime.

3.5 Overview of receipt telegrams

Command	Telegram	from	to	Reference
10	switches 1…32	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
20	sensors 1…4	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
21	sensors 5…8	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
22	sensors 9…12	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
23	sensors 13…16	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
80	Requesting parameter values	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
81	Request of a telegram	CM	DC, GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**

3.6 Overview of send telegrams

Command	Telegram	from	to	Reference
20	Nominal values and power	DC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
21	Pressure sensors	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
22	Temperature sensors	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
23	Temperature sensors	DC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
30	Rotational speed and fuel quantity	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweis-**

				quelle konnte nicht gefunden werden.
40	Alarm and motor state	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
41	Error codes	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
42	Error codes	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
43	Error codes	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
44	Error codes	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
45	Error codes	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
50	Configurable	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
51	Configurable	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
52	Configurable	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
60	Net frequency	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
61	Generator frequency	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
62	Voltage	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
63	Generator voltage	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
64	Phase current	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**

65	Power	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
66	Wattmeter	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
67	Idle-current wattmeter	GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
80	Answer to request of parameter values	DC, GC	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**

3.7 Overview of special telegrams

Command	Telegram	from	to	Reference
97	Connection establishment	DC GC CM CM	CM CM DC GC	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
98	Duplicate ID check	CM	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
98	Answer to duplicate ID check	CM	CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**
99	LifeSign	CM CM DC GC	DC GC CM CM	**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.**

4 Parameter description

4.1 Value range of sensors

Sensor values are transmitted from the control device to the customer module and in the opposite direction always in the internal value range of the control device. The correspondence of the internal value range to the used range is shown in the following tables. It must be borne in mind that the used value range of several parameters is itself parametrizable.

4.1.1 Speed governor (DC)

		Value range				
Sensor		maximum		used		internal
N.	Indicated value		Unit	N.	Reference parameter	
2900	Setpoint1Extern	0,0..100,0	%		0,0 100,0	0 65535
2901	Setpoint1Extern	0,0..100,0	%		0,0 100,0	0 65535
2904	BoostPressure	0,00..5,00	bar	982 983	BoostPressSensorLow BoostPressSensorHigh	0 65535
2905	OilPressure	0,00..20,00	bar	980 981	OilPressSensorLow OilPressSensorHigh	0 65535
2906	AmbientPressure	0..2000	mbar	984 985	AmbPressSensorLow AmbPressSensorHigh	0 65535
2907	CoolantTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2908	ChargeAirTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2909	OilTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2910	FuelTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2911	ExhaustTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2914	SlideExcitReduction	0,0..100,0	%		0,0 100,0	0 65535
2915	SlideSpeedReduction	0, 0..4000,0	Min^{-1}	 991	0 SpeedRedSensorHigh	0 65535
2916	CoolantPressure	0,00..5,00	bar	978 979	CoolPressSensorLow CoolPressSensorHigh	0 65535
2918	MeasuredPower	0,0..100,0	%		0,0 100,0	0 65535
		0,0..2500,0	kW	992 993	MeasPowerSensorLow MeasPowerSensorHigh	0 65535
2919	PowerSetpoint	0,0..100,0	%		0,0 100,0	0 65535
		0,0..2500,0	kW	994 995	PowerSetpSensorLow PowerSetpSensorHigh	0 65535

2920	TractionVoltage	0,0..870,0	V	996	0,0 TractVoltSensorHigh	0 65535
2921	TractionCurrent	0..7500	A	997	0 TractCurrSensorHigh	0 65535
2922	TractionPower	0,0..6525,0	kW	998	0,0 TractPowerSensorHigh	0 65535

4.1.2 Theseus (GC)

		Value range				
Sensor		**maximum**		**used**		**internal**
N.	**Indicated value**		**Unit**	**N.**	**Reference parameter**	
2900	PowerSetpoint	-200,0..200,0	%	980 981	PowerSetpointLow PowerSetpointHigh	0 65535
2901	PFSetpoint	0,00..1,00		982 983	PFSetpointLow PFSetpointHigh	0 65535
2902	LoadLimitExt	0,0..200,0	%	984 985	LoadLimitExtLow LoadLimitExtHigh	0 32767
2903	AnalogLSLineIn	0,0..200,0	%	986 987	LSLLow LSLHigh	0 32767
2911	OilTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2912	OilPressure	0,00..20,00	bar	988 989	OilPressSensorLow OilPressSensorHigh	0 65535
2913	CoolantTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2914	CoolantLevel	0,0..100,0	%		0,0 100,0	0 65535
2915	FuelLevel	0,0..100,0	%		0,0 100,0	0 65535
2916	ExhaustTemp	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2921	GenTempStator_1	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2922	GenTempStator_2	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2923	GenTempStator_3	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2924	GenTempRotor_1	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2925	GenTempRotor_1	-100,0..1000,0	°C		-100,0 1000,0	0 65535
2926	GenTempRotor_1	-100,0..1000,0	°C		-100,0 1000,0	0 65535

4.2 Value range of measured and indicated values

N.	Indicated value	External value range	Unit	Internal value range
2000	Speed	0,0..4000,0	1/Min	0..65535
2031	SpeedSetp	0,00..1,00		0..65535
2350	FuelQuantity	0,0..100,0 0.. 500,0	% mm³	0..65535
2300	ActPos	0,0..100,0	%	0..65535
12001	FrequencyNet_L1	0,00..100,00	Hz	0..65535
12002	FrequencyNet_L2	0,00..100,00	Hz	0..65535
12003	FrequencyNet_L3	0,00..100,00	Hz	0..65535
12011	FrequencyGeneratorL1	0,00..100,00	Hz	0..65535
12012	FrequencyGeneratorL2	0,00..100,00	Hz	0..65535
12013	FrequencyGeneratorL3	0,00..100,00	Hz	0..65535
12107	VoltageBusPrim_1_2	0..60000	V	0..65535
12108	VoltageBusPrim_2_3	0..60000	V	0..65535
12109	VoltageBusPrim_3_1	0..60000	V	0..65535
12127	VoltageGenPrim_1_2	0..60000	V	0..65535
12128	VoltageGenPrim_2_3	0..60000	V	0..65535
12129	VoltageGenPrim_3_1	0..60000	V	0..65535
12147	CurrentPrim_L1	0..10000	A	0..65535
12148	CurrentPrim_L2	0..10000	A	0..65535
12149	CurrentPrim_L3	0..10000	A	0..65535
12203	cosPhi	-1,00..1,00		-32768..32767
12208	PowerPrim	-30000..30000	kW	-32768..32767
12209	PowerReactivePrim	-30000..30000	kVAr	-32768..32767
12210	PowerApparentPrim	-30000..30000	kVA	-32768..32767
13700	Power_GWh	0..65535	GWh	0..65535
13701	Power_MWh	0..999	MWh	0..999
13702	Power_kWh	0..999	kWh	0..999
13704	Power_Reactive_GWh	0..65535	GWh	0..65535
13705	Power_Reactive_MWh	0..999	MWh	0..999
13706	Power_ImpulseReac_kWh	0..999	kWh	0..999

4.3 Overview table

The following table shows only the parameters relevant for the customer module of the single parameter groups one beside the other.

N.	Parameter	N.	Measurement values	N.	Functions	N.	Curves
400	CanStartTimeOutDelay						
401	CanMyNodeNumber	2401	CanTxBufferState				
		2402	CanRxBufferState				
403	CanCMNodeNumber	2403	CanRxTimeout				
		2404	CanTypeMismatch				
		2405	CanOnline				
				4406	CanCommCMOn		
410	CanPrescaler						
411	CanSyncJumpWidth						

N.	Parameter	N.	Measurement values	N.	Functions	N.	Curves
412	CanSamplingMode						
413	CanPhaseSegment1						
414	CanPhaseSegment2						
415	CanPropSegment						
416	CanBaudrate			4416	CanSegmentOrBaudrate		
		2422	CanCMNodeState31to16				
		2423	CanCMNodeState15to01				
810	FunctEngineStop						
849	Funct…						
900	AssignIn_Setp1Ext			4900	ChanTypSetpoint1Ext		
924	AssignIn_…			4924	ChanTyp…		
20810	CommEngineStop						
20849	Comm…						
21950	CMRxTel10Timeout						
21951	CMRxTel20Timeout			14600	CMTxTel20On		
21952	CMRxTel21Timeout			14601	CMTxTel21On		
21953	CMRxTel22Timeout			14602	CMTxTel22On		
21954	CMRxTel23Timeout			14603	CMTxTel23On		
		23720	BitCollection				
				24810	ChanTypEngineStop		
				24849	ChanTyp…		
21960	CMTxTel20SendRate			25960	CMTxTel20On		
21961	CMTxTel21SendRate			25961	CMTxTel21On		
21962	CMTxTel22SendRate			25962	CMTxTel22On		
21963	CMTxTel23SendRate			25963	CMTxTel23On		
21964	CMTxTel30SendRate			25964	CMTxTel30On		
				25965	CMTxTel40On		
				25966	CMTxTel41On		
				25967	CMTxTel42On		
				25968	CMTxTel43On		
				25969	CMTxTel44On		
				25970	CMTxTel45On		
21971	CMTxTel50SendRate			25971	CMTxTel50On		
21972	CMTxTel51SendRate			25972	CMTxTel51On		
21973	CMTxTel52SendRate			25973	CMTxTel52On		
21980	CMTxTel60SendRate			25980	CMTxTel60On		
21981	CMTxTel61SendRate			25981	CMTxTel61On		
21982	CMTxTel62SendRate			25982	CMTxTel62On		
21983	CMTxTel63SendRate			25983	CMTxTel63On		
21984	CMTxTel64SendRate			25984	CMTxTel64On		
21985	CMTxTel65SendRate			25985	CMTxTel65On		
21986	CMTxTel66SendRate			25986	CMTxTel66On		
21987	CMTxTel67SendRate			25987	CMTxTel67On		
						29800	CMTel50ParamSet
						29805	CMTel51ParamSet
						29810	CMTel52ParamSet
						29900	BitCollParamSet

The following table shows only the parameters relevant for the customer module and their respective meaning. For other parameters of the control device please see the corresponding basic information.

4.4 Parameters

N.	Name		Meaning
400	**CanStartTimeOutDelay**		
	Level:	6	Delay of CAN-connection monitoring after reset.
	Range:	0..100 s	
	Page(s):	37	
401	**CanMyNodeNumber**		
	Level:	6	Own node numbers in CAN network
	Range:	1..31	
	Page(s):	4	
403	**CMNodeNumber**		
	Level:	6	Node number of customer module in CAN network
	Range:	1..31	
	Page(s):	4	
410	**CanPrescaler**		
	Level:	6	Prescaler for CAN baud rate
	Range:	0..63	
	Page(s):	5,5	
411	**CanSyncJumpWidth**		
	Level:	6	Synchronization leap interval for CAN baud rate
	Range:	0..3	
	Page(s):	5	
412	**CanSamplingMode**		
	Level:	6	Sampling mode for CAN baud rate
	Range:	0..1	
	Page(s):	5	
413	**CanPhaseSegment1**		
	Level:	6	Phase segment 1 for CAN baud rate
	Range:	0..15	
	Page(s):	5	
414	**CanPhaseSegment2**		
	Level:	6	Phase segment 2 for CAN baud rate
	Range:	0..7	
	Page(s):	5	
415	**CanPropSegment**		
	Level:	6	Propagation segment for CAN baud rate
	Range:	0..7	
	Page(s):	5,5	
20810 to 20849	**CommEngineStop**		
	Level:	6	Bit number assigned to switch function in telegram 10
	Range:	0..32	
	Page(s):	8,15	
21950	**CMRxTel10Timeout**		
	Level:	4	Time interval within which telegram 10 must be received
	Bereich:	0..100 s	
	Page(s)	8	
21951	**CMRxTel20Timeout**		

N.	Name		Meaning
	Level: Bereich: Page(s)	4 0..100 s 8	Time interval within which telegram 20 must be received
21952	**CMRxTel21Timeout**		
	Level: Bereich: Page(s)	4 0..100 s 8	Time interval within which telegram 21 must be received
21953	**CMRxTel22Timeout**		
	Level: Bereich: Page(s)	4 0..100 s 8	Time interval within which telegram 22 must be received
21954	**CMRxTel23Timeout**		
	Level: Bereich: Page(s)	4 0..100 s 8	Time interval within which telegram 23 must be received
21960	**CMTxTel20SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12	Time interval for sending telegram 20 from the control device to the customer module
21961	**CMTxTel21SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12	Time interval for sending telegram 21 from the control device to the customer module
21962	**CMTxTel22SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12	Time interval for sending telegram 22 from the control device to the customer module
21963	**CMTxTel23SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12	Time interval for sending telegram 23 from the control device to the customer module
21964	**CMTxTel30SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12	Time interval for sending telegram 30 from the control device to the customer module
21971	**CMTxTel50SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12,14	Time interval for sending telegram 50 from the control device to the customer module
21972	**CMTxTel51SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12,14	Time interval for sending telegram 51 from the control device to the customer module
21973	**CMTxTel52SendRate**		
	Level: Bereich: Page(s)	4 0..100 s 12,14	Time interval for sending telegram 52 from the control device to the customer module

N.	Name		Meaning
21980	**CMTxTel60SendRate**		
	Level:	4	Time interval for sending telegram 60 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21981	**CMTxTel61SendRate**		
	Level:	4	Time interval for sending telegram 61 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21982	**CMTxTel62SendRate**		
	Level:	4	Time interval for sending telegram 62 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21983	**CMTxTel63SendRate**		
	Level:	4	Time interval for sending telegram 63 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21984	**CMTxTel64SendRate**		
	Level:	4	Time interval for sending telegram 64 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21985	**CMTxTel65SendRate**		
	Level:	4	Time interval for sending telegram 65 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21986	**CMTxTel66SendRate**		
	Level:	4	Time interval for sending telegram 66 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	
21987	**CMTxTel67SendRate**		
	Level:	4	Time interval for sending telegram 67 from Theseus to the customer module
	Bereich:	0..100 s	
	Page(s)	12	

4.5 Measurement values

N.	Name		Meaning
2401	**CanTxBufferState**		
	Level:	1	State of CAN source buffer
	Range:	0..FFFF Hex	
	Page(s):	7	
2402	**CanRxBufferState**		
	Level:	1	State of CAN destination buffer
	Range:	0..FFFF Hex	
	Page(s):	7	
2403	**CanRxTimeout**		
	Level:	1	State of CAN destination timeout monitoring
	Range:	0..FFFF Hex	
	Page(s):	7	
2404	**CanTypeMismatch**		
	Level:	1	State of CAN device numbers
	Range:	0/1	
	Page(s):	7	
2405	**CanOnline**		
	Level:	1	General state of CAN communication
	Range:	0/1	
	Page(s):	7	
2422	**CanCMNodeState31to16**		
	Level:	1	Connection status to customer modules with
	Range:	0/1	node numers from 16 to 31
	Page(s):	6	
2423	**CanCMNodeState15to01**		
	Level:	1	Connection status to customer modules with
	Range:	0/1	node numers from 1 to 15
	Page(s):	6	
23720	**BitCollection(0)**		
to	Level:	1	Bit collection for CAN transmission
23722	Range:	0/1	
	Page(s):	13	

4.6 Functions

N.	Name		Meaning
4400	**CanCommDCOn**		
	Level:	6	Activation of sending and receiving to/from
	Bereich:	0/1	speed governor via CAN
	Page(s):	3	
4401	**CanCommGCOn**		
	Level:	6	Activation of sending and receiving to/from The-
	Bereich:	0/1	seus via CAN
	Page(s):	3	
4406	**CanCommCMOn**		
	Level:	6	Activation of sending and receiving to/from cus-
	Bereich:	0..1	tomer module via CAN
	Page(s):	4,8,9,14	
4416	**CanSegmentOrBaudrate**		
	Level:	4	Selection of baud rate determination
	Bereich:	0/1	0: 416 *CanBaudrate* is used
	Page(s):	5,5	1: 410 *CanPrescaler* to 415 *CanPropSegment* are used
4810	**ChanType_EngineStop**		
to	Level:	6	Configuration of switch input channel type
4849	Bereich: 0..8	Page(s):	0= digital input
		9	8 = CM and digital input
4900	**ChanType_Setp1Ext**		
bis	Level:	6	Configuration of sensor input channel type
4924	Bereich: 0..8	Page(s):	0 = analog
		10	1 = PWM
			8 = CM
5000	**SubstOrLastSetp1Ext**		
bis	Level:	4	Selection of replacement value for speed set
5024	Bereich:	0/1	point in case of error
	Page(s):	11	(0 = last valid value, 1 = replacement value)
5040	**HoldOrResetSetp1Ext**		
bis	Level:	4	Selects if the error of speed set point 1 is to be deleted or kept after signal returns (0 = error is deleted, 1 = error is kept)
5064	Bereich:	0/1	
	Page(s):	11	
25960	**CMTxTel20On**		
	Level:	4	Activation of send telegram 20
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25961	**CMTxTel21On**		
	Level:	4	Activation of send telegram 21
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25962	**CMTxTel22On**		
	Level:	4	Activation of send telegram 22
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent

25963 **CMTxTel23On**

N.	Name		Meaning
	Level:	4	Activation of send telegram 23
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25964	**CMTxTel30On**		
	Level:	4	Activation of send telegram 30
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25965	**CMTxTel40On**		
	Level:	4	Activation of send telegram 40
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25966	**CMTxTel41On**		
	Level:	4	Activation of send telegram 41
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25967	**CMTxTel42On**		
	Level:	4	Activation of send telegram 42
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25968	**CMTxTel43On**		
	Level:	4	Activation of send telegram 43
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25969	**CMTxTel44On**		
	Level:	4	Activation of send telegram 44
	Bereich:	0/1	0 = telegram is not sent
	Seite(n):	12	1 = telegram is sent
25970	**CMTxTel45On**		
	Level:	4	Activation of send telegram 45
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25971	**CMTxTel50On**		
	Level:	4	Activation of send telegram 50
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25972	**CMTxTel51On**		
	Level:	4	Activation of send telegram 51
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25973	**CMTxTel52On**		
	Level:	4	Activation of send telegram 52
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25980	**CMTxTel60On**		
	Level:	4	Activation of send telegram 60
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25981	**CMTxTel61On**		
	Level:	4	Activation of send telegram 61
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25982	**CMTxTel62On**		
	Level:	4	Activation of send telegram 62

N.	Name		Meaning
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25983	**CMTxTel63On**		
	Level:	4	Activation of send telegram 63
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25984	**CMTxTel64On**		
	Level:	4	Activation of send telegram 64
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25985	**CMTxTel65On**		
	Level:	4	Activation of send telegram 65
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25986	**CMTxTel66On**		
	Level:	4	Activation of send telegram 66
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent
25987	**CMTxTel67On**		
	Level:	4	Activation of send telegram 67
	Bereich:	0/1	0 = telegram is not sent
	Page(s):	12	1 = telegram is sent

4.7 Fields

N.	Name		Meaning
29800	**CMTel50ParamSet(0)**		
to	Level:	4	Parameter values for freely configurable telegrams to the customer module
29803	Bereich:	-29999..29999	
	Page(s):	12,13,24	
29805	**CMTel51ParamSet(0)**		
to	Level:	4	Parameter values for freely configurable telegrams to the customer module
29808	Bereich:	-29999..29999	
	Page(s):	12,13,24	
29810	**CMTel52ParamSet(0)**		
to	Level:	4	Parameter values for freely configurable telegrams to the customer module
29813	Bereich:	-29999..29999	
	Page(s):	12,13,24	
29900	**BitCollParamSet (0)**		
to	Level:	4	Parameter values for bit collection
29902	Bereich:	-29999..29999	
	Page(s):	13	

5 Ordering printed documents

A small amount of our printed documentation may be ordered free of charge.

Please order the required printed documents for our control devices at the closest **HEINZMANN** branch/distributor. (Please refer to the list of worldwide distributors on the following pages.)

Please enter the following information:

- your name
- name and address of your firm (just enclose your visiting card).
- delivery address (if different from the above)
- number and title of the required printed manual
- or the technical data of your **HEINZMANN** unit
- the number of copies needed.

To order one or more printed manuals you may use the enclosed fax form.

Please let us know any comments or suggestions to our publications/manuals.

Please send your comments to

HEINZMANN GmbH

Marketing

Am Haselbach 1

D-79677 Schönau

6 Fax form

Order form for HEINZMANN® publications/manuals
Fax hotline +49/7673/8208 194

□ Please send me the following publications:

N° of copies	Publication ID	Title

□ Please send me the latest brochures about

( ) HEINZMANN analog control devices: ..... application:

( ) HEINZMANN digital control devices: ..... application:

Company .....

Contact: .....

Dept./function.....

Road ..... ZIP code/City .....

Tel..... Fax : .....

Line of business.....

Date.....

7 Addresses

7.1 Headquarters

Heinzmann GmbH & Co. KG
Am Haselbach 1
D-79677 Schönau
Germany

Tel. +49 - (0) 7673 - 82 08 - 0
Fax +49 - (0) 7673 - 82 08 - 188
Email info@heinzmann.de www.heinzmann.de/com

7.2 Branches

Australia
HEINZMANN (S.EAsia) Pty. Limited
231 Holt Street
P.O.Box 1415
Eagle Farm QLD 4009

Phone +61 - (0) 7 - 38 68 47 77
Fax +61 - (0) 7 - 38 68 46 66
Email info@govtec.com

Denmark
HEINZMANN DANMARK I/S
Roskildevej 342, Bygning 7
2630 Taastrup

Phone +45 - 43 - 99 92 25 Mobile: 40512377
Fax +45 - 43 - 99 42 23
Email Heinzmann@vip.cybercity.dk

Great Britain
HEINZMANN U.K. LTD.
Teesside Airport
Dinsdale, Darlington
Co. Durham DL2 1PD

Phone +44 - (0)1 - 325 332 805 Mobile: 07831.638.842
Fax +44 - (0)1 - 325 333 631 Gordon.Holt@heinzmannuk.com
Email info@heinzmannuk.com Peter.Walsh@heinzmannuk.com
Chris.Shore@heinzmannuk.com

India
HEINZMANN INDIA Private Limited
SCO-9, SF4, City Plaza, Sector – 16 Market
Faridabad 121 002, Haryana
India

Phone +91 - 129 - 504 6327 Mobile: 981.005.8467
Fax +91 - 129 - 504 6723 981.131.3823
Email hzm_india@vsnl.net Mr. Saraf

Korea
HEINZMANN KOREA Pte. LTD.
473-6 Daebok-Ri
Woongchon-Myon
Ulsan 689-873, Korea

Phone +82 - (0) 52 - 223.2458 Mobile: 019.557.1745
Fax +82 - (0) 52 - 223.2457 016.548.2301
Email hmc4727@kornet.net Mr. Choi

Nederland
HEINZMANN NEDERLAND
Vrijbuiterhof 7
2132 TL Hoofddorp

Phone +31 - 23 - 56 14 729 Mobile: 0655 738123
Fax +31 - 23 - 56 36 831
Email heinzmann.nl@worldmail.nl Mr. Cor de Graaf

Ukraine
HEINZMANN KIEV
ul. Chervonoarmejskaja 84, ap. 14
03150 Kiev 150

Phone +38 - (0) 44 - 227 5531
Fax +38 - (0) 44 - 227 5531
Email heinzmann-kiev@i-c.com.ua Mrs. Elena Galperina

USA
HEINZMANN AMERICA, INC.
8276 Pheasant Run Lane
Wellington, CO 80549

Phone +1 - 970 - 568 0300 Mobile: 970 2139710
Fax +1 - 970 - 568 0700
Email HeinzmannF@aol.com Mr. Jacques van Oppen

7.3 Agencies

Argentina			
ELDI S.A.	Phone	+54 - 11 - 476 827 77	
Calle 136 N 1953/63	Fax	+54 - 11 - 476 431 30	
1653 - Villa Ballester - Pcia. Bs. Aires	Email	eldi@eldi.com.ar	
Brasil			
WS Automacao Industrial Ltd.	Phone	+55 - 19 - 3296 3087	
Rua Angelo Santim, 50 Jardim Nilópolis	Fax	+55 - 19 - 3296 1751	
Campinas - CEP 13089-440			
Sao Paulo	Email	heinzmannbr@aol.com	Mr. Walter Strassburger
China			
Jebsen & Co. LTD.	Phone	+852 - 292 623 36	Mobile: 947.680.11
28/F., Caroline Centre, 28 Yun Ping Road	Fax	+852 - 288 220 17	
Causeway Bay – Hong Kong	Email	hw.sin@mail.jebsen.com.hk	Mr. Sin
France			
DSF Technologies			
Allée Charles-Victor Naudin	Phone	+33 - (0) 4 - 92 38 88 20	
Zone des Templiers, Sophia Antipolis	Fax	+33 - (0) 4 - 92 38 98 89	
06410 Biot	Email	info@dsf-tech.com	
Iran			
Pear Danesh Co. LTD.	Phone	+98 - 21 - 879 50 29	
Kh, S.J. Assadabadi, 16 th St., No 30	Fax	+98 - 21 - 52 52 948	
Tehran 14318- Iran	Email	daneshvar@idehnegar.net.ir	Mr. Daneshvar
Italy			
DSF Tecnologia S.r.l.	Phone	+39 - (0)2 - 91 08 02 09	Mobile: 335.6961.988
Via Ruffini, 3	Fax	+39 - (0)2 - 91 08 03 97	
20030 Paderno Dugnano (MI)	Email	info@dsftecnologia.com	Mr. Cavagnera
Japan			
Summit Link International Inc.	Phone	+81 - (0) 43 - 485 9491	
1568-123 Obukai Sakura-Shi	Fax	+81 - (0) 43 - 489 6061	
Chiba-Ken Japan 285-0836	Email	f.uno@h8.dion.ne.jp	Mr. Uno
Norway			
Data Process Automasjon AS	Phone	+47 - 769 - 610 - 80	
Rombaskvn. 47-E6	Fax	+47 - 769 - 610 - 99	Mr. Karstein Utheim - 81
P.O. Box 336 - 8505 Narvik	Email	dpa@dataprocess.no	Mr. Rolf Richardsen - 82
Singapore			
Siemens Westinghouse	Phone	+65 - 6 - 861 4466	Mobile: 97.345.248
Services Asia Pte. LTD.	Fax	+65 - 6 - 863 1736	
10, Gul Avenue, Jurong	Email	leongmeng.sin@siemens.com	Mr. Sin (section manager)
Singapore 2262		beelim.sim@siemens.com	Mr. Sim (managing director)
Slovakia			
Ing. Imrich Czegledi, CSc.	Phone	+421 - (0) 43 - 41 35 062	Mobile: 0905 750390
Hodzova 16/45	Fax	+421 - (0) 43 - 41 35 062	
036 01 Martin	Email	iczegledi@nextra.sk	Mr. Czegledi
Spain			
Sedni control s.l.	Phone	+34 - 96 - 59 82 178	Mobile: 61.900.8312
C/. BENASAU no. 3, Edifico Alauda	Fax	+34 - 96 - 59 23 067	
03005 – Alicante	Email	sednicontrol@sednicontrol.com	
South Africa			
K H Briegel (PTY) Ltd.	Phone	+27 - (0) 21 - 511 5636	Mobile: 083.702.2379
33 Milner Road, Metro Industrial Township	Fax	+27 - (0) 21 - 511 3535	
Paarden Eiland 7405, Cape Town	Email	briegel@mweb.co.za	Mr. Briegel

Sweden

Mobitron AB	Phone	+46 - (0) 36 - 512 25	Mobile: 0705.85.12.25
P.O Box 241	Fax	+46 - (0) 36 - 511 25	
56123 Huskvarna	Email	soo@mobitron.se	Mr. Sven Olof Olsson

Turkey

	Phone		
	Fax		
	Email	morali@heinzmann.de	Mr. Morali