Heinzmann GmbH & Co. KG
Engine & Turbine Controls
Am Haselbach 1
D-79677 Schönau (Schwarzwald)
Germany
Phone +49 7673 8208-0
Fax +49 7673 8208-188
E-mail info@heinzmann.com
www.heinzmann.com
V.A.T. No.: DE145551926

HEINZMANN®

Digital Speed Governors

CANopen Implementation

DG 06 002-e / 10-06

 **Achtung**	Read this entire manual and all other publications appertaining to the work to be performed before installing, operating or servicing your equipment. All instructions relating to the system and its safety must be scrupulously observed.
 **Gefahr**	Failure to follow instructions may result in personal injury and/or damage to property. HEINZMANN declines all responsibility for damages resulting from failure to observe the instructions.
 **Achtung! Hoch-spannung** 	**Before installing:** Always turn off the power before beginning to install! Be sure to use cable shielding and power supply connections meeting the requirements of the *European Directive Concerning EMI.* Check the functionality of existing protection and monitoring devices.
Gefahr  **Gefahr**	**The following protective and monitoring devices must be mounted to prevent personal injuries and material damages:** overspeed protection acting independently from speed governor overtemperature protection HEINZMANN declines all responsibility for damages resulting from missing or insufficient overspeed protection. **Generator installation will in addition require:** overcurrent protection protection against faulty synchronization due to excessive frequency, voltage or phase differences reverse power protection
	Overspeeding can be caused by: failure of the voltage supply failure of the actuator, control unit or of any accessory device sluggish and blocking linkage

 **Achtung**	**For electronically controlled injection (MVC):** In **common rail** systems each injector pipe must be fitted with a separate mechanical flow limiter. For **Pump-Pipe-Nozzle (PPN)** systems and **Pump-Nozzle (PNU)** systems the injection valves must be designed so as to enable fuel release only when the control piston of the magnetic valve is moving. This inhibits fuel delivery to the individual injection pump when the control piston is stalling in some position.
 **Achtung**	The examples, data and any other information contained in this manual are intended exclusively as instruction aids and should not be used in any particular application without independent testing and verification by the person making the application.
 **Gefahr**	Independent testing and verification are especially important in any application in which malfunction might result in personal injury or damage to property.
	HEINZMANN make no warranties, express or implied, that the examples, data, or other information in this volume are free of error, that they are consistent with industry standards, or that they will meet the requirements for any particular application.
	HEINZMANN expressly disclaim the implied warranties of merchantability and of fitness for any particular purpose, even if **HEINZMANN** have been advised of a particular purpose and even if a particular purpose is indicated in the manual.
	HEINZMANN also disclaim all liability for direct, indirect, incidental or consequential damages that result from any use of the examples, data, or other information contained in this manual.
	HEINZMANN make no warranties for the conception and engineering of the technical installation as a whole. This is the responsibility of the user and of his planning staff and specialists. It is also their responsibility to verify whether the performance features of our devices will meet the intended purposes. The user is also responsible for correct commissioning of the whole installation.

About this manual

Version	Changes effected	Date	Author
1.00	written	29.08.06	Sz

Contents

1 Introduction

CAN protocols may be implemented for all HEINZMANN control units featuring at least one CAN controller. This manual does not describe the CANopen protocol itself, but its implementation in the firmware. The basis for software implementation is the document "CANopen Application Layer and Communication Profile“, CiA Draft Standard 301 Version 4.01, dated June 1st 2000.

The HEINZMANN control unit functions as slave in the 11-bit CAN open pre-defined master-slave connection set.

Four RPDOs and four TPDOs in the standard data set contain the pre-defined identifiers that can be modified. The device node number is entered automatically in these identifiers, in order to avoid having to change them whenever the node number is adapted. Further 12 TPDOs can be implemented on request. All pre-defined transmission modes are supported for the TPDOs. The RPDOs are expected in asynchronous mode. When transmission by the other party is regular, the receipt may be monitored.

Heartbeat producer, heartbeat consumer und node/life guarding have both been implemented according to standard. Each of the two monitoring modes may be selected.

The emergency object is linked and transmits both the error messages recognized by the control unit and their correction. The EMCY object may be enabled separately.

Exactly one service data object SDO has been implemented. The implemented services are described in ↑*5* Service data object.

For each CANopen implementation an EDS file describing the implementation is provided. It should be noted, that the implementation parameters may be modified by the customer by means of DcDesk 2000, especially the assignment of PDOs. In such cases, it is responsibility of the operator to adapt the EDS file accordingly.

2 General CAN parameters

The baud rate is entered in 21750 *CanOp:Baudrate*. Only the four values 125 kBaud, 250 kBaud, 500 kBaud and 1000 kBaud are valid, for every other value 250 kBaud will be used.

The personal node number is transmitted by 21751 *CanOp:MyNodeNo*, the node number of the master/partner in the CANopen network by 21752 *CanOp:PartnerNodeNo*. These node numbers will be automatically added to the respective telegram identifiers, in order to simplify parameter setting.

After the start of the control unit an interval is defined in 21753 *CanOp:TimeOutDelay* within which there is no timeout monitoring of the RPDOs. Within this interval, the control unit should have switched in operational mode and the agreed RPDOs should have been transmitted.

Note

Whenever CANopen parameters are changed, the changes should be saved and followed by a reset of the control unit.

In 23759 *CanOp:RxIRCount* all received telegrams are counted. This serves only as a rapid check to see if the bus is functioning.

CAN bus errors are indicated in 3070 *ErrCanBus/ErrCanBus1* or 3072 *ErrCanBus2*. This error reports either the bus off status of the CAN controller or indicates that no telegram could be placed on the bus within 100 s. In case of a bus error, the control unit automatically goes in pre-operational state, if before it was in operational or stopped state.

The error 3071 *ErrCanComm/ErrCanComm1* or 3073 *ErrCanComm2* indicates that the CAN bus is working correctly but there have been errors in content. The most important content errors are indicated in 23757 *CanOp:ErrLifeSign* and 23758 *CanOp:ErrRPDOTimeOut*. The complete CANopen error status is transmitted with error 3071/3073 via the EMCY object.

The CANopen error status is structured as follows:

bit 0: missing partner life sign (23757 *CanOp:ErrLifeSign*)

bit 1: at least one RPDO has timed out (23758 *CanOp:ErrRPDOTimeOut*)

bit 2: node reset not possible because engine is running

bit 3:: receiving buffer overflow, received telegram cannot be interpreted

bit 4: sending buffer overflow, send telegram cannot be transmitted

bits 5…7: reserve, always 0

3 Telegram identifiers

Telegram identifiers have been assigned standard values that can be changed by means of the following parameters:

21760 *CanOp:ID_SYNCCons*	identifier SYNC object (80 hex)
21761 *CanOp:ID_EMCYProd*	identifier EMCY object (80 hex)
21762 *CanOp:ID_HBeatCons*	identifier heartbeat consumer (700 hex)
21763 *CanOp:ID_HBeatProd*	identifier heartbeat producer (700 hex)
21764 *CanOp:ID_ClientSDO*	identifier client SDO object (600 hex)
21765 *CanOp:ID_ServerSDO*	identifier server SDO object (580 hex)
21770 *CanOp:RPDOID()*	identifiers of the four RPDOs (200, 300, 400, 500 hex)
21774 *CanOp:TPDOID()*	identifiers of the four TPDOs, on request extendable to16 TPDOs (180, 280, 380, 480 hex, all others 0)

Identifiers 21760 *CanOp:ID_SYNCCons* and 21778 *CanOp:TPDOID(4)* to 21789 *CanOp:TPDOID(15)* are taken over by the control unit as set in the parameters.

In the identifiers of the send objects 21761 *CanOp:ID_EMCYProd*, 21763 *CanOp:ID_HBeatProd*, 21765 *CanOp:ID_ServerSDO*, 21770 *CanOp:RPDOID(0)* to 21773 *CanOp:RPDOID(3)* and 21774 *CanOp:TPDOID(0)* to 21777 *CanOp:TPDOID(3)* the personal node number 21751 *CanOp:MyNodeNo* is automatically entered in the lower seven bits, in order to avoid having to modify it when the node number changes. In this way, data sets may be copied to other control units and only the node number must be adapted.

To the identifier 21762 *CanOp:ID_HBeatCons* the personal node number 21751 *CanOp:MyNodeNo* is added when node/life guarding is active (23756 *CanOp:LifeGuarding* = 1), in all other cases with the partner node number 21752 *CanOp:PartnerNodeNo*.

4 Network management

The current state of the control unit in the CANopen system is indicated by the following parameters:

23750 *CanOp:Init* = 1	initialization
23751 *CanOp:PreOperational* = 1	pre-operational
23752 *CanOp:Operational* = 1	operational
23753 *CanOp:Stopped* = 1	stopped

Only one of these values may be active at any one time.

Layer setting services are not supported. Baud rate and node numbers may be changed by means of DcDesk 2000.

The NMT commands 81 hex (ResetNode) and 82 hex (ResetComm) are answered by an automatic reset of the control unit, if the control unit supports this function. For safety reasons the engine must be stopped. Otherwise a CANopen communication error is generated and transmitted through the EMCY telegram ↑*6* Emergency object.

After an automatic reset, the control unit automatically passes from init state to pre-operational state and is responsive again. The time required by this procedure varies according to control unit type and may be as long as 500 ms.

5 Service data object

Exactly one service data object SDO has been implemented. The identifier of the SDO receipt telegram must be parameterized in 21764 *CanOp:ID_ClientSDO*, the identifier of the SDO send telegram in 21765 *CanOp:ID_ServerSDO*. The personal node number 21751 *CanOp:MyNodeNo* is automatically added to these identifiers, they must therefore not be adapted when the node number changes.

Segmented SDO transfer and block transfer are not supported.

The objects 1000, 1001, 1005, 100C, 100D, 1010, 1011, 1014, 1016, 1017, 1018, 1200, 1400-1403, 1600…1603, 1800…180F, 1A00…1A0F Hex have been implemented.

Objects 1002, 1003, 1006, 1007, 1008, 1009, 100A, 1012, 1013, 1015 and 1280 are not supported.

The following manufacturer-related objects are implemented:

2000 Hex:	This object allows separate reading and writing of switch functions.
2001 Hex:	This object allows to read sensor errors, with additional writing access when RPDO1 is not enabled.
2002 Hex:	This object allows to read out sensor values, with additional writing access when the respective RPDO is not enabled.
2100..210F Hex:	These objects allow to read out specific TPDO parameters, also as bit values.

6 Emergency object

The identifier of the EMCY send telegram must be parameterized in 21761 *CanOp:ID_EMCYProd*. The personal node number 21751 *CanOp:MyNodeNo* is automatically added to this identifier, it must therefore not be adapted when the node number changes. The EMCY object will be transmitted only if it has been enabled with 25751 *CanOp:EMCYOn* = 1.

The emergency object supports both error codes 00xx (Error reset or No error) und 10xx (Generic error).

In data bytes 3 to 6 of the manufacturer specific error field the error code generated by Heinzmann and possibly an additional error status will be transmitted.

byte 3,4 HEINZMANN error codes 3001..3094, 13000..13099, 23000..23099

byte 5,6 HEINZMANN error status, if supported by the control unit (e.g., by DARDANOS III, DARDANOS IV and ARIADNE)

CANopen error status for other control units, only for errors 3071 *ErrCanComm/ErrCanComm1* or 3073 *ErrCanComm2*, depending on which CAN controller is used for the CANopen protocol

0 other

All existing error codes and error states are described in the documentation of the control unit or the software version.

The CANopen error status is structured as follows:

bit 0: missing partner life sign (23757 *CanOp:ErrLifeSign*)

bit 1: at least one RPDO has timed out (23758 *CanOp:ErrRPDOTimeOut*)

bit 2: node reset not possible because engine is running

bit 3: receiving buffer overflow, receive telegram cannot be interpreted

bit 4 sending buffer overflow, send telegram cannot be transmitted

bits 5…7 reserve, always 0

Control unit errors may be deleted via CANopen using the switch function 2828 *SwitchErrorReset*, by setting 24828 *ChanTypErrorReset* to 4 and 20828 *CommErrorReset* to the respective bit number in RPDO1.

When errors are deleted by means of DcDesk 2000 or of the switch function 2828 *SwitchErrorReset*, the CANopen error status is deleted too.

At most four emergency object telegrams are sent within 10 or 16 ms respectively, the time interval is specific to the control unit, and is always the same.

7 Receive telegrams

The four RPDOs correspond to the standard of the 11-bit CANopen pre-defined master-slave connection set.

The respective four identifiers from 21770 *CanOp:RPDOID()* onward have been assigned the default values 200, 300, 400 and 500 hex. The node number of the control unit 21751 *CanOp:MyNodeNo* is automatically entered in the lower seven bits of the identifiers, in order to avoid having to change them in occasion of node number modification.

The RPDOs allow to transmit the current sensor and switch function values to the control unit. The respective RPDO is expected only if at least one switch function or a sensor has been assigned to it and if the telegram starting from 25770 *CanOp:RPDO1On* = 1 has been generally enabled. Receive telegram length must correspond at least to the expected length, which is output starting from 23770 *RPDOTelLen().*

The RPDOs are expected in asynchronous mode. In case of regular transmission by the master/partner it is also possible to define an interval to monitor the receipt of incoming data separately for each RPDO starting from 29000 *RPDOEvtTim().*

For RPDO1, it is indicated in bytes in the parameters starting from 23760 *CanOp:SwitchMask()* at which bits switch functions or sensor errors are expected.

For RPDO2 to RPDO4, a mask appears in the two bytes starting from 23764 *CanOp:SensorMask(),* in which a bit is set for each place where a sensor is expected. These two masks serve to verify parameter settings on client and server side.

7.1 Telegram structure

RPDO1:	bit mask	23760 *CanOp:SwitchMask(0)*
	telegram length	23770 *CanOp:RPDOTelLen(0)*
	activation	25770 *CanOp:RPDO1On*
	receive rate	29000 *CanOp:RPDOEvtTim(0)*
	byte 0 bits 0..7:	value of switch functions 1..8
	byte 1 bits 0..7:	value of switch functions 9..16
	byte 2 bits 0..7:	value of switch functions 17..24
	byte 3 bits 0..7:	value of switch functions 25..32
	byte 5 bits 0..7:	value of sensor errors 1..8
	byte 6 bits 0..3:	value of sensor errors 9…12
RPDO2:	sensor mask	23764 *CanOp:SensorMask(0),* bits 0..3
	telegram length	23771 *CanOp:RPDOTelLen(1)*
	activation	25771 *CanOp:RPDO2On*
	receive rate	29001 *CanOp:RPDOEvtTim(1)*
	words 0..3:	values of sensors 1..4

respective sensor errors in RPDO1, byte 5, bits 0..3

RPDO3:	sensor mask	23764 *CanOp:SensorMask(0)*, bits 4..7
	telegram length	23772 *CanOp:RPDOTelLen(2)*
	activation	25772 *CanOp:RPDO3On*
	receive rate	29002 *CanOp:RPDOEvtTim(2)*
	words 0..3:	values of sensors 5..8
	respective sensor errors in RPDO1, byte 5, bits 4..7	
RPDO4:	sensor mask	23765 *CanOp:SensorMask(1)*, bits 0..3
	telegram length	23773 *CanOp:RPDOTelLen(3)*
	activation	25773 *CanOp:RPDO4On*
	receive rate	29003 *CanOp:RPDOEvtTim(3)*
	words 0..3:	values of sensors 9..12
	respective sensor errors in RPDO1, byte 6, bits 0..3	

7.2 Receiving data

Telegram RPDO1 transmits the current values of switch functions and sensor error codes. Telegrams RPDO2 to RPDO4 contain current sensor values. Which switch functions are transmitted by which bit of RPDO1 and which sensor is transmitted by which word of RPDO2 to RPDO4 is determined by the manufacturer of the sending module and must be agreed with him.

7.2.1 Switch functions

All switch functions defined in the control device may be received either by way of a dedicated hardware input, by way of the telegram RPDO1 or in both ways. 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 RPDO1, in 24810 *ChanTyp...* to 24849 *ChanTyp...* the value 4 must be entered to indicate the chosen channel type, if the receipt is to happen exclusively by way of hardware and for unused switch functions, channel type 0 must be chosen.

If CANopen is chosen by selecting channel type 4, the bit number in telegram RPDO1 must be indicated in the corresponding parameter 20810 *Comm...* to 20849 *Comm....* Up to 32 different switch functions may be transmitted with the telegram. They are chosen and assigned by the programmer of the CANopen master.

In the parameters starting from 23760 *CanOp:SwitchMask*() it is indicated in bytes at which bits switch functions or sensor errors are expected. This serves to verify parameter settings on client and server side.

If one of the switch functions received through CANopen 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 only) is chosen, 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
208xx *Comm...* = bit no.	bit number in telegram RPDO1 (0, 1…32)
248xx *ChanTyp...* = 4	switch function value is received via CANopen

The bit number counts bytewise, i.e. the first data byte of the telegram contains bits 1…8 (LSB..MSB), the second, bits 9..16 (LSB..MSB), and so on. The bit number 0 amounts to saying "not used". For communication purposes, such a switch function will always have the value 0.

A switch function is activated if it addressed by at least one of the two possible sources. digital input OR RPDO1.

The value "1" in telegram RPDO1 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 setpoint adjuster 2, "0" activates setpoint adjuster 1.

For safety reasons, a function must be activated consciously via a communications module. For this reason, the switch functions addressed by communications modules can be only high-active, i.e. become active on receipt of a "1", as opposed to digital inputs. When the connection to the communication module is interrupted, the switch function automatically adopts the value 0.

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

When value 4 is set for CANopen for the switch functions starting from 24810 *ChanTyp…*, but the protocol is not activated with 25750 *CanOpenOn* = 1, all these switch functions are reset to 0 and the error message 3000 *ConfigurationError* is output.

If there is a CAN error, either a bus error, or a timeout error of telegram RPDO1, the CAN value of all switch functions assigned through CANopen is equally returned to 0. If the telegram is received again, switch functions are determined again by way of CAN.

7.2.1.2 Switch function engine stop

In case of a CAN error, the switches determined via CAN are deleted or reset to zero. If in this case an "engine stop signal" 2810 *SwitchEngineStop* transmitted via CAN is to lead to an engine stop in any case, the parameter 4810 *StopImpulseOrSwitch* must be set to 1. This parameter allows to define whether an external stop command re-

mains active only during the time the command is explicitly active or if an impulse is sufficient to keep the command active until the engine has stopped.

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

Note

For safety reasons, HEINZMANN recommends to connect the engine stop always directly, regardless of a possible additional transmission through a communication module.

7.2.1.3 Value of a switch function

With on-off switches the name is equivalent to the signification *On*. State 1 of the switch function will always define *On* and state 0 *Off*. The names of change-over switches or of parameters selecting between two functions always include an “Or”, where the expression preceding “Or” will be valid when the value of the switch function is 1 and where the expression following “Or” will be valid when the switch function has the value 0.

If no communication module is enabled in the current firmware, the value of the switch function is determined exclusively by digital input. The parameters starting from 20810 *Comm…* and 24810 *ChanTyp…* do not exist.

If, on the other hand, a communication module must be taken into account, then each switch function can be addressed either by a digital input or by the communication module or even by both.

1. Digital input only
   Parameter starting from 24810 *ChanTyp…* must be set to 0.
   If parameter 810 *Funct…* = 0, then the switch function always has the value 0, otherwise it has the current value of the digital input (possibly with inverted activity).

2. Communication module only
   Parameter starting from 810 *Funct…* must be set to 0 and the respective parameter starting from 24810 *ChanTyp…* must be >= 3 (4 for CANopen).
   If parameter 20810 *Comm…* = 0, then the switch function always has the value 0, otherwise it has the current value of the received telegram RPDO1. When the connection to the communication module is interrupted, the switch function automatically adopts the value 0.

3. Both digital input and communication module
   Parameter 810 *Funct…* is not equal 0, parameter starting from 20810 *Comm…* > 0 and parameter starting from 24810 *ChanTyp…* >= 3 (4 for CANopen).

The current value from the digital input (possibly inverted) and from the communications module are combined with OR. The switch function will therefore be = 0 only if both sources send the value 0; it will be = 1 if at least one source sends the value 1. When the connection to the communication module is interrupted, the switch function automatically adopts the value 0 for this transmission path. In this case, the digital input alone decides on the overall value.

Note

On the other hand, HEINZMANN advises never to connect change-over switches that select between two functions (with OR in their identifier) using both signal paths.

7.2.2 Sensors

Each sensor defined in the control device may be received either by way of a dedicated hardware input or by way of the telegrams RPDO2 to RPDO4. The unique receipt path must be communicated to the control device.

In order to use the sensor values received through telegrams RPDO2 to RPDO4 in the control unit, the channel type starting from 4900 *ChanTyp…* must be set to the value "4", for receipt through an analogue input of one's own hardware the channel type must be set to "0", for receipt through a PWM input of one's own hardware the channel type must be set to "1".

The number of the input channel must be entered in the respective parameter starting from 900 *AssignIn_....* Channel number 0 amounts to saying "not used".

49xx *ChanTyp...* = 4	sensor value is received via CANopen
9xx *AssignIn_...* = channel no.	channel number (0, 1…12)

Up to 12 different sensors may be received through telegrams RPDO2 to RPDO4. They are chosen and assigned by the programmer of the CANopen master. The channel numbers in the telegrams count word for word, i.e. the first word in telegram RPDO2 defines channel 1, the second word channel 2, and so on. The fourth word of telegram RPDO4 has channel number 12.

Parameterizing Example:

You want current boost pressure and coolant temperature to be received every 50 ms via words 1 and 2 of telegram RPDO2.

Number	Parameter	Value	Unit
904	*AssignIn_BoostPressure*	1	
907	*AssignIn_CoolantTemp*	2	
4904	*ChanType_BoostPress*	4	
4907	*ChanType_CoolantTemp*	4	

25750	*CanOpenOn*	1	
25771	*CanOp:RPDO2On*	1	
29001	*RPDOEvtTim(1)*	0.05	s

7.2.2.1 Error in the configuration or in CAN receipt of sensors

If the sensors 49xx *ChanTyp...* are set = 4, but CANopen is not activated with 25750 *CanOpenOn* = 1, all these sensor values are set back internally to the value zero and a configuration error 3000 *ConfigurationError* is generated at the same time.

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


Note

The internal value zero identifies a different external physical value depending on the sensor type ↑10.1 Value range of sensors.

The effective value of sensors in case of error depends on the settings entered in the parameters starting from 5000 *SubstOrLast…* and 5040 *HoldOrReset….*

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.

8 Send telegrams

Depending on application requirements, up to 16 TPDOs may be defined. For each TPDO the identifier may be parameterized in 21774 *CanOp:TPDOID()*.

The first four TPDOs always correspond to the standard of the 11-bit CANopen pre-defined master-slave connection set. These four identifiers have been assigned the default values 180, 280, 380 and 480 Hex. The node number of the control unit 21751 *CanOp:MyNodeNo* is automatically entered in the lower seven bits of the identifiers, in order to avoid having to change them whenever a node number is modified. The identifiers of TPDOs 5 to 16 are adopted without changes.

Only the TDPOs activated with 25774 *CanOp:TPDO1On* = 1 are sent.

8.1 Types of transmission

The following types of transmission are supported; they can be defined separately for each TPDO in 29004 *CanOp:TPDOTxType()*. If other values are indicated, the respective telegram is not sent.

0 synchronous acyclic
transmission after receipt of a SYNC signal, but only if at least one send value has changed between two SYNC signals

1..240 synchronous cyclic
transmission after receipt of x SYNC signals, x = [1,240]

252 synchronous, RTR only
on receipt of the SYNC telegram the data is latched, transmission only with RTR

253 asynchronous, RTR only
transmission on request by RTR message

254 asynchronous, event manufacturer specific
Transmission after an interval pre-defined in the parameters starting from 29020 *CanOp:TPDOEvtTim()*, but only if at least one value has changed beyond the predefines hysteresis value and not more frequently than defined in parameters starting from 29036 *CanOp:TPDOInhTim()*. Hysteresis may be defined separately for each send parameter starting from 29116 *CanOp:TPDOxHyst()*.

transmission also on request by RTR message

The receipt of the SYNC object is enabled automatically when a synchronous transmission is requested, otherwise it is suppressed to minimize system load.

8.2 Transmission values

The values to send may be defined separately for each TPDO by entering them in the respective parameter numbers starting from 29052 *TPDOxAssign()*. All parameters with a level no higher than 4 may be transmitted. The single parameter numbers must be entered

into the fields consecutively. With the first zero or non-existent or not admitted parameter (e.g., level too high) the transmission field ends. The telegram length as determined by the control unit is indicated in parameters starting from 23774 *CanOp:TPDOTelLen()*. A TPDO is sent only if telegram length is not equal to zero and the telegram has been switched on by means of the parameters starting from 25774 *CanOp:TPDO1On.*

8.2.1 Collection of single bits for compressed transmission

Each parameter is transmitted as word, even if only byte- or bit-sized. In order to make the most of the available space, all bit parameters, i.e. parameters that can take on only the values 0 or 1, may be compressed. The field 29900 *BitCollParamSet()* is provided for this purpose. Here as many parameter numbers relating to bit parameters as desired may be entered. The current values are indicated in the same position (field index = bit number) in 23720 *BitCollection()*. These parameter numbers 23720 ff in turn may be entered into the parameters following 29052 *TPDOxAssign()* to transmit compressed bits. Normally, 29900 *BitCollParamSet()* contains 32 elements, which allow to form two send words. A zero is transmitted for all unassigned columns.

9 Life sign monitoring

HEINZMANN control units allow both heartbeat monitoring and node/life guarding. Both procedures use the same identifiers 21762 *CanOp:ID_HbeatCons* and 21763 *CanOp:ID_HbeatProd*, albeit with a different addition to the consumer identifier.

9.1 Heartbeat monitoring

If parameter 21754 *CanOp:HbeatConsTime* contains a value not equal to zero, the life sign of the master/partner is monitored according to this time interval. This is indicated by 23754 *CanOp:HBeatConsumer* = 1.

When 21755 *CanOp:HbeatProdTime* is not equal to zero, the life sign of the device itself is transmitted at this send rate. This is indicated by 23755 *CanOp:HBeatProducer* = 1.

The identifier of the heartbeat consumer is set in 21762 *CanOp:ID_HbeatCons* and is automatically extended by the partner node number 21752 *CanOp:PartnerNodeNo* whenever life sign monitoring is enabled as described above. The identifier of the heartbeat producer 21763 *CanOp:ID_HbeatProd* on the other hand is automatically extended by its own node number 21751 *CanOp:MyNodeNo.*

9.2 Node/life guarding

If a zero was entered for both values of heartbeat monitoring, the function node/life guarding will be enabled/disabled by 21756 *CanOp:GuardingTime* and 21757 *CanOp:LifeTimeFactor*. The function is enabled only if these two parameters are not equal to zero.

When node guarding is active, 21762 *CanOp:ID_HbeatCons* is extended with its own node number 21751 *CanOp:MyNodeNo.* 23756 *CanOp:LifeGuarding* shows whether node/life guarding is enabled.

10 Parameter description

10.1 Value range of sensors

Sensor values are transmitted from the control device to the CANopen partner (↑ *8.2* Transmission values) and in the opposite direction (↑ *7.2.2* Sensors), always within 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.

10.1.1 Speed governor

		Value range				
Sensor		Maximum		used		internal
No.	Indicated value		Unit	No.	Reference parameter	
2900	Setpoint1Extern	0.0..100.0	%		0.0 100.0	0 65535
2901	Setpoint2Extern	0.0..100.0	%		0.0 100.0	0 65535
2902	LoadControlInput	0.0..100.0	%		0.0 100.0	0 65535
2903	SyncInput	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
2912	RailPressure1	0.0..2000.0	bar	986 987	RailPress1SensorLow RailPress1SensorHigh	0 65535
2913	RailPressure2	0.0..2000.0	bar	988 989	RailPress2SensorLow RailPress2SensorHigh	0 65535
2914	SlideExcitReduction	0.0..100.0	%		0.0 100.0	0 65535
2915	SlideSpeedReduction	0.0..4000.0	rpm	987	0.0 SpeedRedSensorHigh	0 65535

2916	CoolantPressure	0.00..5.00	bar	978 979	CoolPressSensorLow CoolPressSensorHigh	0 65535
2917	AsymmetricLoad	0.0..100.0	%		0.0 100.0	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

10.1.2 Theseus

		Value range				
Sensor		**maximum**		**used**		**internal**
No.	**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

10.1.3 Kronos 20

		Value range				
Sensor		maximum		used		internal
No.	Indicated value		Unit	N.	Reference parameter	
2912	ManifoldPressure	0.00..5.00	bar	986 987	MnfldPressSensorLow MnfldPressSensorHigh	0 65535
2913	ManifoldTemp	-100.0..1000.0	°C		-100.0 1000.0	0 65535
2914	MeasuredPower	0.0..2500.0	kW	988 989	MeasPowerSensorLow MeasPowerSensorHigh	0 65535
2915	LambdaProbe	0.00..5.00	V	990 991	LambdaProbeLow LambdaProbeHigh	0 65535
2916	CH4Content	0.0..100.0	%	992 993	CH4ContentSensorLow CH4ContentSensorHigh	0 65535

10.2 Value range of measured and indicated values

All values are transmitted from the control device to the CANopen partner within the internal value range of the control device (↑ *8.2* Transmission values). Other values as shown in the following tables are possible on request.

10.2.1 Speed governor

No.	Indicated value	External value range	Unit	Internal value range
2000	Speed	0.0..4000.0	1/Min	0..65535
2031	SpeedSetp	0.0..4000.0	1/Min	0..65535
2050	SpeedVariance	0..65535		0..65535
2350	FuelQuantity	0.0..100.0 0..500.0	% mm³/str	0..65535
2300	ActPos	0.0..100.0	%	0..65535
2940	BoostPressRelative	0.00..5.00	bar	0..65535
2941	AbsoluteAltitude	-5000..5000	m	-32768..32767

10.2.2 Theseus

No.	Indicated value	External value range	Unit	Internal value range
2000	Speed	0.0..4000.0	1/Min	0..65535
2031	SpeedSetp	0.0..4000.0	1/Min	0..65535
2350	FuelQuantity	0.0..100.0 0..500.0	% mm³/str	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

10.2.3 Kronos 20

No.	Indicated value	External value range	Unit	Internal value range
2000	Speed	0.0..4000.0	1/Min	0..65535
2050	SpeedVariance	0..65535		0..65535
3462	LambdaDesiredValue	0.000..2.500		0..65535

10.3 Overview table

The following table shows only the parameters relevant to the CANopen protocol in the single parameter groups one beside the other.

No.	Parameter	No.	Measurements	No.	Functions	No.	Curves
810 ...	FunctEngineStop	2810...	SwitchEngineStop				
900 ...	AssignIn_Setp1Ext	2900...	Setpoint1Extern	4900...	ChanTypSetp1Ext		
		3070	ErrCanBus1				
		3071	ErrCanComm1				
		3072	ErrCanBus2				
		3073	ErrCanComm2				
20810.	CommEngineStop			24810.	ChanTypeEngineStop		
		23720	BitCollection(0)				
21750	CanOp:Baudrate	23750	CanOp:Init	25750	CanOpenOn		
21751	CanOp:MyNodeNo	23751	CanOp:PreOperational	25751	CanOp:EMCYOn		
21752	CanOp:PartnerNodeNo	23752	CanOp:Operational				
21753	CanOp:TimeOutDelay	23753	CanOp:Stopped				
21754	CanOp:HBeatConsTime	23754	CanOp:HBeatConsumer				
21755	CanOp:HBeatProdTime	23755	CanOp:HBeatProducer				
21756	CanOp:GuardingTime	23756	CanOp:LifeGuarding				
21757	CanOp:LifeTimeFactor	23757	CanOp:ErrLifeSign				
		23758	CanOp:ErrRPDOTimeOut				
		23759	CanOp:RxIRCount				
21760	CanOp:ID_SYNCCons	23760	CanOp:SwitchMask(0)				
21761	CanOp:ID_EMCYProd						
21762	CanOp:ID_HBeatCons						
21763	CanOp:ID_HBeatProd						
21764	CanOp:ID_ClientSDO	23764	CanOp:SensorMask(0)				
21765	CanOp:ID_ServerSDO						
21770	CanOp:RPDOID(0)	23770	CanOp:RPDOTelLen(0)	25770	CanOp:RPDO1On		
				25771	CanOp:RPDO2On		
				25772	CanOp:RPDO3On		
				25773	CanOp:RPDO4On		
21774	CanOp:TPDOID(0)	23774	CanOp:TPDOTelLen(0)	25774	CanOp:TPDO1On		
				25775	CanOp:TPDO2On		
				25776	CanOp:TPDO3On		
				25777	CanOp:TPDO4On		
				25778	CanOp:TPDO5On		
				25779	CanOp:TPDO6On		
				25780	CanOp:TPDO7On		
				25781	CanOp:TPDO8On		
				25782	CanOp:TPDO9On		
				25783	CanOp:TPDO10On		
				25784	CanOp:TPDO11On		
				25785	CanOp:TPDO12On		
				25756	CanOp:TPDO13On		
				25787	CanOp:TPDO14On		
				25788	CanOp:TPDO15On		
				25789	CanOp:TPDO16On		
						29000	CanOp:RPDOEvtTim()
						29004	CanOp:TPDOTxType()
						29020	CanOp:TPDOEvtTim()
						29036	CanOp:TPDOInhTim()
						29052	CanOp:TPDO1Assign()
						29056	CanOp:TPDO2Assign()
						29060	CanOp:TPDO3Assign()
						29064	CanOp:TPDO4Assign()
						29068	CanOp:TPDO5Assign()

No.	Parameter	No.	Measurements	No.	Functions	No.	Curves
						29072	CanOp:TPDO6Assign()
						29076	CanOp:TPDO7Assign()
						29080	CanOp:TPDO8Assign()
						29084	CanOp:TPDO9Assign()
						29088	CanOp:TPDO10Assgn()
						29092	CanOp:TPDO11Assgn()
						29096	CanOp:TPDO12Assgn()
						29100	CanOp:TPDO13Assgn()
						29104	CanOp:TPDO14Assgn()
						29108	CanOp:TPDO15Assgn()
						29112	CanOp:TPDO16Assgn()
						29116	CanOp:TPDO1Hyst()
						29120	CanOp:TPDO2Hyst()
						29124	CanOp:TPDO3Hyst()
						29128	CanOp:TPDO4Hyst()
						29132	CanOp:TPDO5Hyst()
						29136	CanOp:TPDO6Hyst()
						29140	CanOp:TPDO7Hyst()
						29144	CanOp:TPDO8Hyst()
						29148	CanOp:TPDO9Hyst()
						29152	CanOp:TPDO10Hyst()
						29156	CanOp:TPDO11Hyst()
						29160	CanOp:TPDO12Hyst()
						29164	CanOp:TPDO13Hyst()
						29168	CanOp:TPDO14Hyst()
						29172	CanOp:TPDO15Hyst()
						29176	CanOp:TPDO16Hyst()
						29900	BitCollParamSet()

The following table shows only the parameters relevant to the CANopen protocol and their respective meaning. For other parameters of the control device please see the corresponding basic information.

For characteristic curves and maps only the first field parameter is included and the parameter numbers are indicated with the complement "ff" (and following).

10.4 Parameters

No.	Name		Meaning
810	**Funct_…**		
ff	Level:	6	Assignment of the digital input at which the value for the indicated switch function is expected.
	Range:	-8..8	
	Page(s):	8, 10	
900	**AssignIn_…**		
ff	Level:	6	Assignment of the channel number at which the value for the indicated sensor is expected.
	Range:	0..12	
	Page(s):	11	
21750	**CanOp:Baudrate**		
	Level:	4	CAN baud rate
	Range:	125..1000 kBaud	
	Page(s):	2	
21751	**CanOp:MyNodeNo**		

No.	Name		Meaning
	Level:	4	Personal node number in CAN network
	Range:	1..127	
	Page(s):	2, 5, 7, 13, 15	
21752	**CanOp:PartnerNodeNo**		
	Level:	4	Node number of master/partner in CAN net-
	Range:	0..127	work
	Page(s):	2, 15	
21753	**CanOp:TimeOutDelay**		
	Level:	4	Delay after control device start.
	Range:	0..100 s	
	Page(s)	2	
21754	**CanOp:HBeatConsTime**		
	Level:	4	Heartbeat receiving rate
	Range:	0..50 s	
	Page(s)	15	
21755	**CanOp:HBeatProdTime**		
	Level:	4	Heartbeat sending rate
	Range:	0..50 s	
	Page(s)	15	
21756	**CanOp:GuardingTime**		
	Level:	4	Node Guarding monitoring interval
	Range:	0..50 s	
	Page(s)	15	
21757	**CanOp:LifeTimeFactor**		
	Level:	4	Factor for Node Guarding monitoring interval
	Range:	0..255	
	Page(s):	15	
21760	**CanOp:ID_SYNCCons**		
	Level:	4	Identifier of SYNC receipt telegram
	Range:	0000..07FF Hex	Standard 80 Hex
	Page(s)::	3	
21761	**CanOp:ID_EMCYProd**		
	Level:	4	Identifier of EMCY send telegram
	Range:	0000..07FF Hex	Standard: 80 Hex
	Page(s)::	3, 6	
21762	**CanOp:ID_HBeatCons**		
	Level:	4	Identifier of heartbeat receipt telegram
	Range:	0000..07FF Hex	Standard: 700 Hex + partner node number (the
	Page(s)::	3, 15	latter is added automatically)
			or
			Identifier of NodeGuarding receipt telegram
			Standard: 700 Hex + own node number
			(the latter is added automatically)
21763	**CanOp:ID_HBeatProd**		
	Level:	4	Identifier of heartbeat send telegram
	Range:	0000..07FF Hex	Standard: 700 Hex + own node number
	Page(s)::	3, 15	(the latter is added automatically)
21764	**CanOp:ID_ClientSDO**		
	Level:	4	Identifier of SDO receipt telegram
	Range:	0000..07FF Hex	Standard 600 Hex + partner node number
	Page(s)::	3, 5	(the latter is added automatically)
21765	**CanOp:ID_ServerSDO**		
	Level:	4	Identifier of SDO send telegram

No.	Name		Meaning
	Range:	0000..07FF Hex	Standard: 580 Hex + own node number
	Page(s)::	3, 5	(the latter is added automatically)
21770	**CanOp:RPDOID(0)**		
ff	Level:	4	Identifiers of max. 4 RPDOs
	Range:	0000..07FF Hex	Standard: 200, 300, 400, 500 hex + own node
	Page(s)::	3, 7	number
			(the latter is added automatically)
21774	**CanOp:TPDOID(0)**		
ff	Level:	4	Identifiers of max. 16 TPDOs
	Range:	0000..07FF Hex	Standard for the first four TPDOs: 180, 280, 380,
	Page(s)::	3, 13	480 hex + own node number
			(the latter is added automatically)
			TPDOs 5 to 16 may be assigned freely
20810	**Comm…**		
ff	Level:	6	Assignment of a bit number in the first 4 bytes
	Range:	0..32	of RPDO1 to which the value for the indicated
	Page(s):	8, 10	switch function is delivered.

10.5 Measurements

No.	Name		Meaning
2810	**SwitchEngineStop**		
	Level:	1	Switch function “EngineStop”
	Range:	0..1	
	Page(s):	9	
2827	**SwitchSetpoint2Or1**		
	Level:	1	Switch function “Setpoint2Or1”
	Range:	0..1	
	Page(s):	9	
2828	**SwitchErrorReset**		
	Level:	1	Switch function “ErrorReset”
	Range:	0..1	
	Page(s):	6	
3070	**ErrCanBus/ErrCanBus1**		
	Level:	1	CAN bus error, CAN controller 1
	Range:	0..1	
	Page(s):	2	
3071	**ErrCanComm/ErrCanComm1**		
	Level:	1	CAN communication error, CAN controller 1
	Range:	0..1	
	Page(s):	2, 6	
3072	**ErrCanBus2**		
	Level:	1	CAN bus error, CAN controller 2
	Range:	0..1	
	Page(s):	2	
3073	**ErrCanComm2**		
	Level:	1	CAN communication error, CAN controller 2
	Range:	0..1	
	Page(s):	2, 6	
23720	**BitCollection(0)**		
ff	Level:	1	Collection of single bits for compressed trans-
	Range:	0000..FFFF Hex	mission in TPDOs

No.	Name		Meaning
	Page(s):	14, 26	Please refer to 29900 *BitCollParamSet(0).*
23750	**CanOp:Init**		
	Level:	1	1: CANopen in init state
	Range:	0..1	
	Page(s):	4	
23751	**CanOp:PreOperational**		
	Level:	1	1: CANopen in pre-operational state
	Range:	0..1	
	Page(s):	4	
23752	**CanOp:Operational**		
	Level:	1	1: CANopen in operational state
	Range:	0..1	
	Page(s):	4	
23753	**CanOp:Stopped**		
	Level:	1	1: CANopen in stopped state
	Range:	0..1	
	Page(s):	4	
23754	**CanOp:HBeatConsumer**		
	Level:	1	1: Heartbeat consumer is active.
	Range:	0..1	
	Page(s):	15	
23755	**CanOp:HBeatProducer**		
	Level:	1	1: Heartbeat producer is active.
	Range:	0..1	
	Page(s):	15	
23756	**CanOp:LifeGuarding**		
	Level:	1	1: Node Guarding is active.
	Range:	0..1	
	Page(s):	3,15	
23757	**CanOp:ErrLifeSign**		
	Level:	1	1: Life sign error detected (heartbeat consumer or Node Guarding)
	Range:	0..1	
	Page(s):	2,6	
23758	**CanOp:ErrRPDOTimeOut**		
	Level:	1	1: at least one RPDO has timed out.
	Range:	0..1	
	Page(s):	2,6	
23759	**CanOp:RxIRCount**		
	Level:	1	Receive telegram counter
	Range:	0..65535	
	Page(s):	2	
23760	**CanOp:SwitchMask(0)**		
ff	Level:	4	Indication of bits occupied in RPDO1 by assigned switch functions
	Range:	00..FF Hex	
	Page(s)::	7	
23764	**CanOp:SensorMask(0)**		
ff	Level:	4	Indication of sensors assigned to RPDO2 to RPDO4.
	Range:	00..FF Hex	
	Page(s)::	7	
23770	**CanOp:RPDOTelLen(0)**		
ff	Level:	4	Ascertained minimum length of the up to 4 RPDOs
	Range:	0..8	
	Page(s):	7	

No.	Name		Meaning
23774	**CanOp:TPDOTelLen(0)**		
ff	Level:	4	Ascertained minimum length of the up to 16
	Range:	0..8	TPDOs
	Page(s):	14	

10.6 Functions

No.	Name		Meaning
4900	**ChanTyp…**		
ff	Level:	6	Assignment of channel type from which the
	Range:	0..4	value for the indicated sensor is expected
	Page(s):	11	0 = analogue input
			1 = PWM input
			4 = CANopen
			(please refer to 900 *AssignIn_...*)
24810	**ChanTyp…**		
ff	Level:	6	Assignment of protocol type from which the
	Range:	0..4	value for the indicated switch function is ex-
	Page(s):	8,10	pected
			0 = only through digital inputs (810 *Funct…*)
			4 = through CANopen too (20810 *Comm…*)
25750	**CanOpenOn**		
	Level:	4	Activation of CANopen protocol
	Range:	0..1	
	Page(s):	9	
25751	**CanOp:EMCYOn**		
	Level:	4	Activation of EMCY send telegram
	Range:	0..1	
	Page(s):	6	
25770	**CanOp:RPDOxOn**		
ff	Level:	4	Activation of the up to 4 RPDOs
	Range:	0..1	
	Page(s):	7	
25774	**CanOp:TPDOxOn**		
ff	Level:	4	Activation of the up to 16 TPDOs
	Range:	0..1	
	Page(s):	13	

10.7 Fields

No.	Name		Meaning
29000	**CanOp:RPDOEvtTim(0)**		
Ff	Level:	4	Event time of the up to 4 RPDOs
	Range:	0..50 s	Index 0…3 = RPDO1..RPDO4
	Page(s)	7	
29004	**CanOp:TPDOTxType(0)**		

No.	Name		Meaning
ff	Level: Range: Page(s):	4 0..255 13	Transmit types for the up to 16 TPDOs except the values 241…251 and 255, which are ignored, all transmit types are possible. Index 0...15 = TPDO1..TPDO16
29020	**CanOp:TPDOEvtTim(0)**		
ff	Level: Range: Page(s):	4 0..50 s 13	Event time of the up to 16 TPDOs, only for transmit type 254 Index 0…15 = TPDO1..TPDO16
29036	**CanOp:TPDOInhTim(0)**		
ff	Level: Range: Page(s):	4 0..50 s 13	Inhibit time of the up to 16 TPDOs, only for transmit type 254 Index 0…15 = TPDO1..TPDO16
29052	**CanOp:TPDOxAssign(0)**		
ff	Level: Range: Page(s):	4 0..29999 13	Assignment of parameter numbers to the resp. max. 4 send parameters of the max. 16 TPDOs x = [1,16] Index 0..3 = word 1..4
29116	**CanOp:TPDOxHyst(0)**		
ff	Level: Range: Page(s):	4 0..100 % 13	Assignment of hysteresis values to the resp. max. 4 send parameters of the max. 16 TPDOs, only for transmit type 254 x = [1,16] Index 0..3 = word 1..4
29900	**BitCollParamSet(0)**		
ff	Level: Range: Page(s):	4 -29999..29999 14	Collection of parameter numbers corresponding to bit values for word transmission in TPDO1…TPD16 Please refer to 23720 *BitCollection(0)*.

11 Order Specifications for Manuals

There is no charge for our technical manuals ordered in reasonable quantities.

Order the necessary manuals on our speed governors from your nearest

HEINZMANN location.

(Please click on “HEINZMANN location” to see the list of our subsidiaries and agents in the world).

Please include the following information:

- your name,
- the name and address of your company (you can simply include your business card),
- the address where you want the manuals sent (if different from above),
- the number(s) (as on front page bottom right) and title(s) of the desired manual(s),
- or the technical data of your HEINZMANN equipment,
- the quantity you want.

You can directly use the following fax-form for ordering one or several manuals.

Most of the manuals are available as acrobat PDF-files, too. On request they can be send via e-mail.

We solicit comments about the content and the presentation of our publications. Please, send your comments to:

HEINZMANN GmbH & Co. KG

Service Department

Am Haselbach 1

D-79677 Schönau

Germany

Fax Reply

Order for HEINZMANN technical manuals
Fax-Hotline +49 7673 8208 194

☐ **Please send me the following manuals:**

Quantity	No. of the manual	Title

☐ **Please send me your new sales documentation about**

( ) the HEINZMANN Analogue Governors Application:.....

( ) the HEINZMANN Digital Governors Application:.....

( ) the HEINZMANN Gas Engine Equipment Application:.....

Company .....

Contact Person.....

Department .....

Address..... Country/Code/Town .....

Phone..... Fax .....

E-Mail.....

Activity.....

Date .....