**Heinzmann GmbH & Co. KG**
**Engine & Turbine Management**

Am Haselbach 1
D-79677 Schönau/Germany

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

V.A.T. No.: DE145551926

# HEINZMANN®

**Engine & Turbine Controls**

# MODBUS

# Operating Instructions


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

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

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

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

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

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


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

– Overspeed protection independent of the rpm controller

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

– thermal overload protection

**The following must also be provided for alternator systems:**

– Overcurrent protection
– Protection against faulty synchronisation for excessively-large frequency, voltage or phase difference
– Directional contactor

The reasons for overspeeding may be:

– Failure of positioning device, control unit or its auxiliary devices
– Linkage sluggishness and jamming


**The following must be observed before an installation:**

– Always disconnect the electrical mains supply before any interventions to the system.
– Only use cable screening and mains supply connections that correspond with the *European Union EMC Directive*
– Check the function of all installed protection and monitoring systems

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

# Version Information

| Version | Description of Changes | Date | Edited by |
|---|---|---|---|
| 02-05 | created | 2004-11-30 | DeM |
| | Supplemented by chapter 3 Installation | 2005-02-03 | ShD |
| 11-07 | Chapter 3.3 RS485-two-wire-system corrected | 2007-11-19 | ShD |
| | Front page and page header updated | 2007-11-20 | DeM |
| | Chapter Order Specifications for Manuals, appendixes FAX Replay and Subsidiaries deleted | 2007-11-20 | DeM |
| | Minor changes | 2007-11-20 | DeM |
| 08-14 | Chapter 4.2 Supported function codes and 0 Supported exception codes updated | 2014-07-08 | EnJ |
| | Parameter for Modbus Extended in Chapter 5.1 Configuration of interface added | 2014-07-08 | EnJ |
| | Chapter 5.3 Write data and subchapter revised | 2014-07-08 | EnJ |
| | Chapter Overview Table deleted | 2014-08-08 | DeM |
| | Front page and warning notices updated | 2014-08-08 | DeM |
| | Minor changes | 2014-08-08 | DeM |

# Table of Contents

**Page**

# 1 Safety instructions and related symbols

This publication offers wherever necessary practical safety instructions to indicate inevitable residual risks when operating the engine. These residual risks imply dangers to

- Personnel
- Product and machine
- The environment

**The primary aim of the safety instructions is to prevent personal injury!**

The signal words used in this publication are specifically designed to direct your attention to possible damage extent!

DANGER ***DANGER indicates a hazardous situation the consequence of which could be fatal or severe injuries if it is not prevented.***

WARNING ***WARNING indicates a hazardous situation which could lead to fatal injury or severe injuries if it is not prevented.***

CAUTION ***CAUTION indicates a hazardous situation which could lead to minor injuries if it is not prevented.***

NOTICE ***NOTICE indicates possible material damage.***


***Safety instructions are not only denoted by a signal word but also by hazard warning triangles. Hazard warning triangles can contain different symbols to illustrate the danger. However, the symbol used is no substitute for the actual text of the safety instructions. The text must therefore always be read in full!***

***This symbol does not refer to any safety instructions but offers important notes for better understanding the functions that are being discussed. They should by all means be observed and practiced.***

## 1.1 Basic safety measures for normal operation

- The installation may be operated only by authorized persons who have been duly trained and who are fully acquainted with the operating instructions so that they are capable of working in accordance with them.
- Before turning the installation on please verify and make sure that
- - only authorized persons are present within the working range of the engine;
- - nobody will be in danger of suffering injuries by starting the engine.
- Before starting the engine always check the installation for visible damages and make sure it is not put into operation unless it is in perfect condition. On detecting any faults please inform your superior immediately!
- Before starting the engine remove any unnecessary material and/or objects from the working range of the installation/engine.
- Before starting the engine check and make sure that all safety devices are working properly!

## 1.2 Basic safety measures for servicing and maintenance

- Before performing any maintenance or repair work make sure the working area of the engine has been closed to unauthorized persons. Put on a sign warning that maintenance or repair work is being done.
- Before performing any maintenance or repair work switch off the master switch of the power supply and secure it by a padlock! The key must be kept by the person performing the maintenance and repair works.
- Before performing any maintenance and repair work make sure that all parts of engine to be touched have cooled down to ambient temperature and are dead!
- Refasten loose connections!
- Replace at once any damaged lines and/or cables!
- Keep the cabinet always closed. Access should be permitted only to authorized persons having a key or tools.
- Never use a water hose to clean cabinets or other casings of electric equipment!

## 1.3 Before putting an installation into service after naintenance and repair works

- Check on all slackened screw connections to have been tightened again!
- Make sure the control linkage has been reattached and all cables have been reconnected.
- Make sure all safety devices of the installation are in perfect order and are working properly!

# 2 Introduction

Modbus is a serial Master-Slave-protocol. This protocol is open and therefore freely accessible for any user. It provides two transmission modes RTU and ASCII and can be used on RS232 and RS485 or RS422.

For application of the Modbus protocol, the HEINZMANN control units of the THESEUS and HELENOS series must be equipped with an add-on pc board. The HEINZMANN Connection Module uses exclusively RTU mode. Throughout this document, it will therefore be only this mode that is made reference to. As a connection an RS485 or RS422 interface is provided. Hence an additional level converter will be required for connection to an RS232 interface.

Any details concerning the design and operation of the interface relate to the below specifications which can be downloaded from www.modbus.org.

- Hardware reference: Modbus over Serial Line Specification and Implementation Guide V1.02 – Dec 20, 2006
- Protocol reference: Modbus Application Protocol Specification V1.1b3 – April 26, 2012

The following sections give a general overview of the functionality, concept and structure of the Modbus protocol. Experienced Modbus users may skip them and continue with chapter ↑3 *Installation*.

## 2.1 Message format of data transmission

Modbus data transmission always consists of a request/reply cycle. Request and reply are always executed sequentially, never simultaneously. To separate the individual messages from each other transmission pauses are required as delimiters.

The following Figure 1 shows a typical request-reply-cycle.

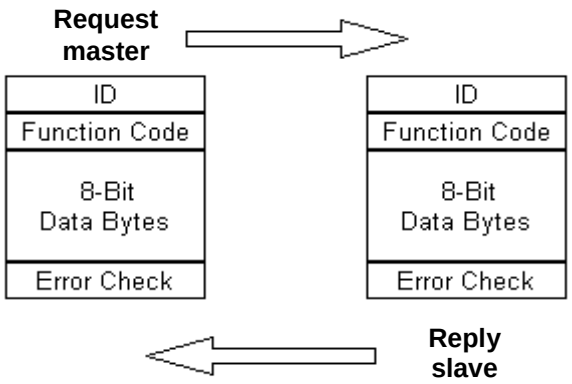



**Figure 1: Request-reply-cycle**

The Modbus protocol defines a simple Protocol Data Unit (PDU) independent of the type of network used and includes a function code indicating by which of the diverse actions the data is to be processed.

As additional information, the Application Data Unit (ADU) includes the participant address (ID) allowing Master and Multi-Slave operation, and a so-called Error Check defined as the check-sum over the characters to be transmitted. The address and the error check are added before and behind the character string as shown in Figure 2 and are transmitted together with it.

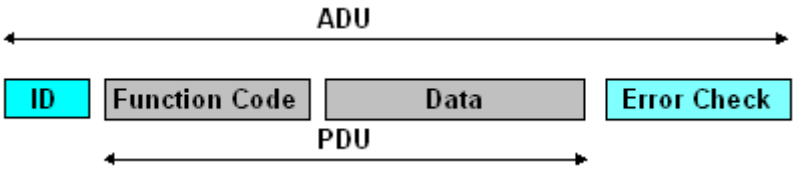



**Figure 2: Protocol frame**

In RTU Mode the error check is created by a well-defined procedure (Cyclical Redundancy Check, in short CRC). The error check comprises two bytes.

Besides the error check that is to be evaluated for the totality of a message's characters, it is also possible to set and perform a parity check for each individual byte of a message.

If checking any message for parity or CRC results in an error, processing the message is aborted and no response is returned.

In contrast to errors that are detected by examining the error checks (parity, CRC) and that are not responded to, there may occur errors caused by the user when the specifications for the data to be transmitted exceed the internal address and data ranges of the device or when a function code is not supported. In this case, the request is answered by an exception message indicating the cause by means of an exception code.

To obtain an assessment of the quality of the data exchange, a number of diagnostic counter have been implemented that are assigned to the various causes. These 16-bit counters are incremented on the occurrence of the respective event and can be read out both via Modbus and directly by a HEINZMANN diagnostics tool.

## 2.2 Example of a data transmission

The function code 0x03 *"Read Holding Registers"* is used to explain the sequence of Master request and Slave reply / Slave exception reply by the following tables (Table 1, Table 2 and Table 3). These tables show the formal structure of the transmitted character strings. The last column contains a numeric example in hexadecimal notation.

| | | | |
|---|---|---|---|
| ID | 0 to 247 (decimal) | Slave address | 01 |
| Function | 1 to 255 (decimal) | Function to be performed | 03 |
| Data | Starting Address High | Starting address of data register (high byte) | 00 |
| | Starting Address Low | Starting address of data register (low byte) | 00 |
| | No. of Registers High | Number of data (high byte) | 00 |
| | No. of Registers Low | Number of data (low byte) | 01 |
| CRC | CRC Low | Error check CRC (low byte) | 84 |
| | CRC High | Error check CRC (high byte) | 0A |

**Table 1: Master request**

| | | | |
|---|---|---|---|
| ID | 0 to 247 (decimal) | Slave address | 01 |
| Function | 1 to 255 (decimal) | Function to be performed | 03 |
| ByteCount | 2 to 255 (decimal) | Number of data bytes | 02 |
| Data | Register Value High | Data value (high byte) | 00 |
| | Register Value Low | Data value (low byte) | 00 |
| CRC | CRC Low | Error check CRC (low byte) | B8 |
| | CRC High | Error check CRC (high byte) | 44 |

**Table 2: Slave reply**

| | | | |
|---|---|---|---|
| ID | 0 to 247 (decimal) | Slave-Adresse | 01 |
| Function | 1 to 255 (decimal) | Function to be performed + 0x80 | 83 |
| Data | Data Byte | Exception code | 02 |
| CRC | CRC Low | Error check CRC (low byte) | C0 |
| | CRC High | Error check CRC (high byte) | F1 |

**Table 3: Slave exception reply**

# 3 Installation

The installation of a Modbus data bus system requires particular care, especially with regard to electromagnetic compatibility as well as to the data transmission rate to be obtained in dependence on type and length of the conductor. In order to avoid errors in the pre-commissioning phase, the most important points concerning the installation will be discussed in advance. More details will be dealt with in later chapters or are to be found in the Modbus Specification, see section ↑2 *Introduction*.

Note

*Connecting a HEINZMANN control unit to a serial data bus system according to EIA/TIA-485/422 Standard for using the Modbus protocol requires that the control unit be equipped with the Modbus connection module.*

## 3.1 Bus topology

The bus topology of the Modbus consists of a straight data link cable as shown in Figure 3. Basically, both point-to-point connections and multipoint connections are allowed.

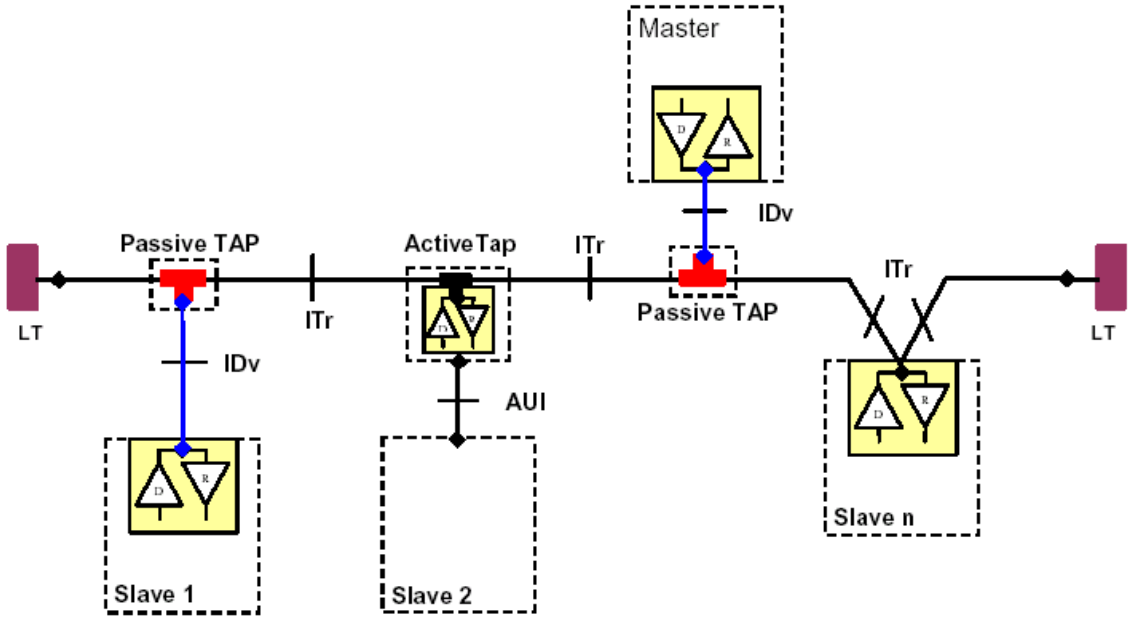



**Figure 3: Bus topology**

A remote data cable (trunk) is running across some longer distance from a Modbus Master to one or more Modbus Slaves. Preferably, the individual Slaves are to be connected to the trunk directly by a linear structure (from node to node, Daisy-Chain structure) or by a short derivation cable.

The elements of the bus are:

**ITr** – Trunk Interface – a bus cable. For different distances, cables with suitable properties must be used, such as characteristic impedance, shield and line construction. The cable lines form a bus of linear structure which is characterized by having two ends. Maximum

length is 1,200 m. For more details see the sections ↑*3.3.1 Bus cable in RS485-two-wire-systems* and ↑*3.4.1 Bus cable in RS485-four-wire-systems*.

**IDv** – Derivation Interface – a derivation cable. By the derivation cable all bus signals, the reference potential and the shield must be conducted from the bus to the participant (Slave or Master). A derivation cable may connect several participants, e.g., the devices in a switch cabinet. Derivation cables must not be longer than 20 m. If there is a local tap with more than one derivation cables connected, the cable length must be limited to 40 m divided by the number of derivation cables. For more details see sections ↑*3.3.2 Derivation cable in RS485-two-wire-systems* and ↑ 3.4.2 *Derivation cable in RS485-four-wire-system*.

**Passive TAP** – a passive tap. The tap connects the bus cable with the derivation cables, i.e., the data lines, the line for the reference potential and the shield. Possibilities of earthing and termination can be provided within the tap.

**Active Tap** – a tap having an integrated bus driver. The rest is very much the same as with the passive tap.

**AUI** – Attachment Unit Interface – a data cable connecting active taps with the participants.

**Slave** – a participant serving as a Slave device. The maximum number of participants of the bus without repeater is 31. For more details see section ↑3.6 *Slave* participant.

**Master** - a participant serving as a Master device. Only one Master may be connected to the bus. For more details see section ↑*3.7 Master participant*.

**LT** – Line Terminator – Bus terminator. At either extremity of the bus a line termination must be provided. For more details see section ↑*3.10 Line termination*.

## 3.2 Bus configuration

The Modbus standard provides three communication configurations:

- according to RS485 in a two-wire-system,
- according to RS485 (or RS422) in a four-wire-system and
- according to RS232 (requiring accessory devices for HEINZMANN units).

The first two configurations provide differential data transmission via a balanced twisted pair (or pairs) of conductors. Communication is supported by a reference potential (Common). Due to EMI, the standard Modbus requires using shielded cables. In these operating modes both multiple connections and a point-to-point connection will be allowed.

With RS232 operation, an unbalanced data transmission is basically given using a separate signal line for either direction, i.e. for transmission from Master to Slave and from Slave to

Master. This will also require an additional line with the reference potential (Common). RS232 operation is provided only for a point-to-point connection.

As communication equipment the following items will be of importance:

- Slave participant,
- Master participant ,
- Bus cable,
- Derivation cable,
- Reference potential,
- Shield,
- Bus earthing,
- Line termination, and
- Line polarization.

## 3.3 RS485-two-wire-system

In a two-wire-system, bi-directional data are being transmitted, viz. Master-Slave and Slave-Master, by a common balanced twisted pair of conductors. There is, in addition, a separate conductor (e.g. a twisted pair of conductors) for the reference potential (Common). The complete cable must have an external shield.

The 2-wire-system constitutes the configuration most frequently used in practice. The layout of the bus must be about the same as shown in **Figure 4**. In a daisy-chain structure, no derivation cables may be used.

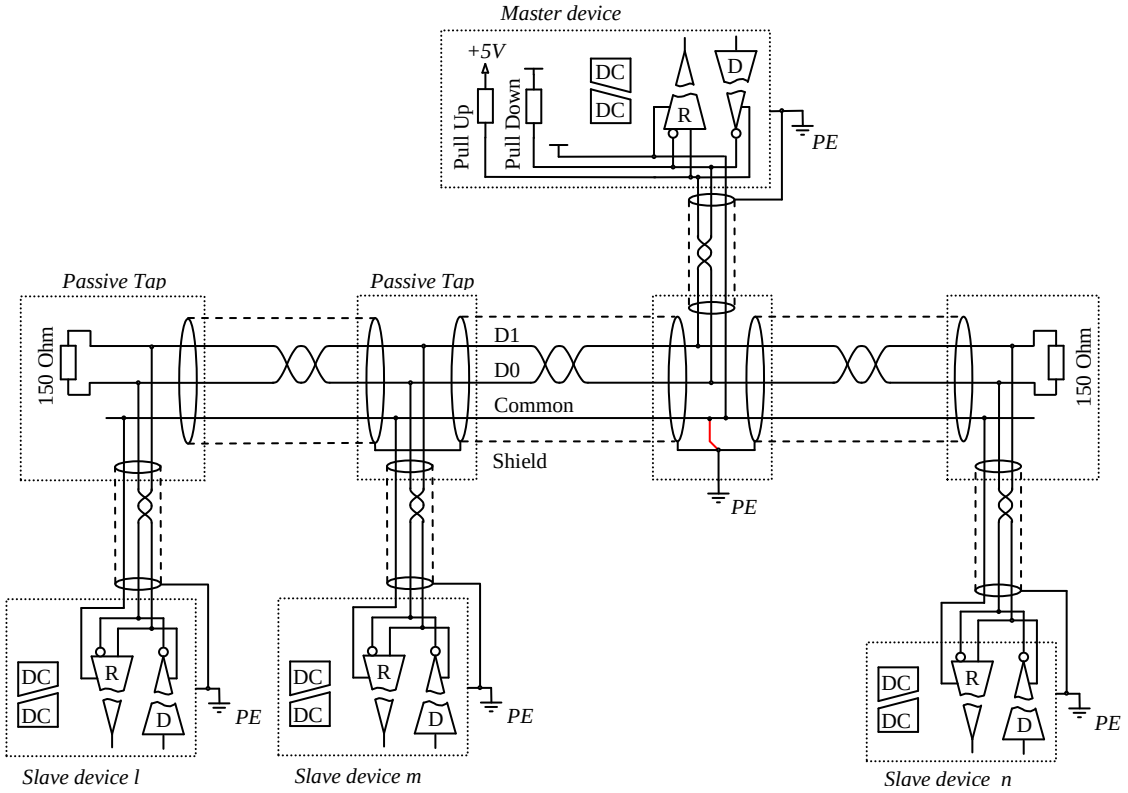



**Figure 4: RS485 – two-wire-system**

The following Table 4 defines the signals as in **Figure 4**.

| Designation | for Device | Contact at Participant | EIA/TIA-485 Designation | Description |
|---|---|---|---|---|
| D1 | I/O | yes | B/B’ | Transmitter-receiver-terminal 1 (+), voltage V1<br>(V1 > V0 for binary 1 and OFF-state respectively) |
| D0 | I/O | yes | A/A’ | Transmitter-receiver-terminal 0 (–), voltage V0<br>(V0 > V1 for binary 0 and ON-state respectively) |
| Common | - | yes | C/C’ | Reference potential<br>and optionally as supply potential |
| Shield | - | yes | - | Cable shield |

**Table 4 Signal definition – two-wire-system**

The connectors A and A' pin no. 8 and 9 respectively and B and B' pin no. 4 and 5 respectively have to be shorted to use the two-wire-system with the Modbus connection module for HEINZMANN devices.

### 3.3.1 **Bus cable in RS485-two-wire-systems**

There exist two variants:

- a cable with one balanced twisted pair of conductors, a separate conductor, and with external shield,
- a cable with two balanced twisted pairs of conductors and with external shield.

The conductor size must generally be chosen for lengths of at least 1,000 m. Thus, for all conductors, including the shield, a gauge of at least AWG 24 must be provided. The characteristic impedance must be no less than 100 Ohms in the frequency range up to 100 kHz (for a baud rate of 19,200).

Basically, also cables with several twisted pairs of conductors can be used such as STP category 5 cables (4 pairs).

### 3.3.2 **Derivation cable in RS485-two-wire-systems**

The derivation cable must consist of:

- at least one balanced twisted pair of conductors for the differential data signal,
- a separate conductor or twisted pair of conductors for the reference potential (Common), and
- an external shield.

For derivations, for instance, the same cable can be used as for the bus.

Generally, also cables with several twisted pairs of conductors can be used.

Every derivation cable connects one or more participants to the bus (by passive taps). Therefore, the derivation cable includes

- the connection to the passive tap (connector, screw terminals, etc.), and
- the connection or connections to the participants (connectors, screws, terminals e.a.)

For every participant connection, attention should be paid to the device specific pin and bridge assignments (e.g., 4-wire to 2-wire, see **Figure 4**). The connectors A and A' pin no. 8 and 9 respectively and B and B' pin no. 4 and 5 respectively have to be shorted to use the two-wire-system with the Modbus connection module for HEINZMANN devices.

## 3.4 RS485-Four-wire-system

With 4-wire-systems, data transmission in both directions, viz. Master-Slave and Slave-Master, is made by two separate balanced twisted pairs of conductors. There is, in addition, a separate conductor (e.g., a twisted pair of conductors) for the reference potential (Common). The complete cable must have an external shield.

An example of the bus configuration is shown in **Figure 5**. In the daisy-chain structure no derivation cable may be used.

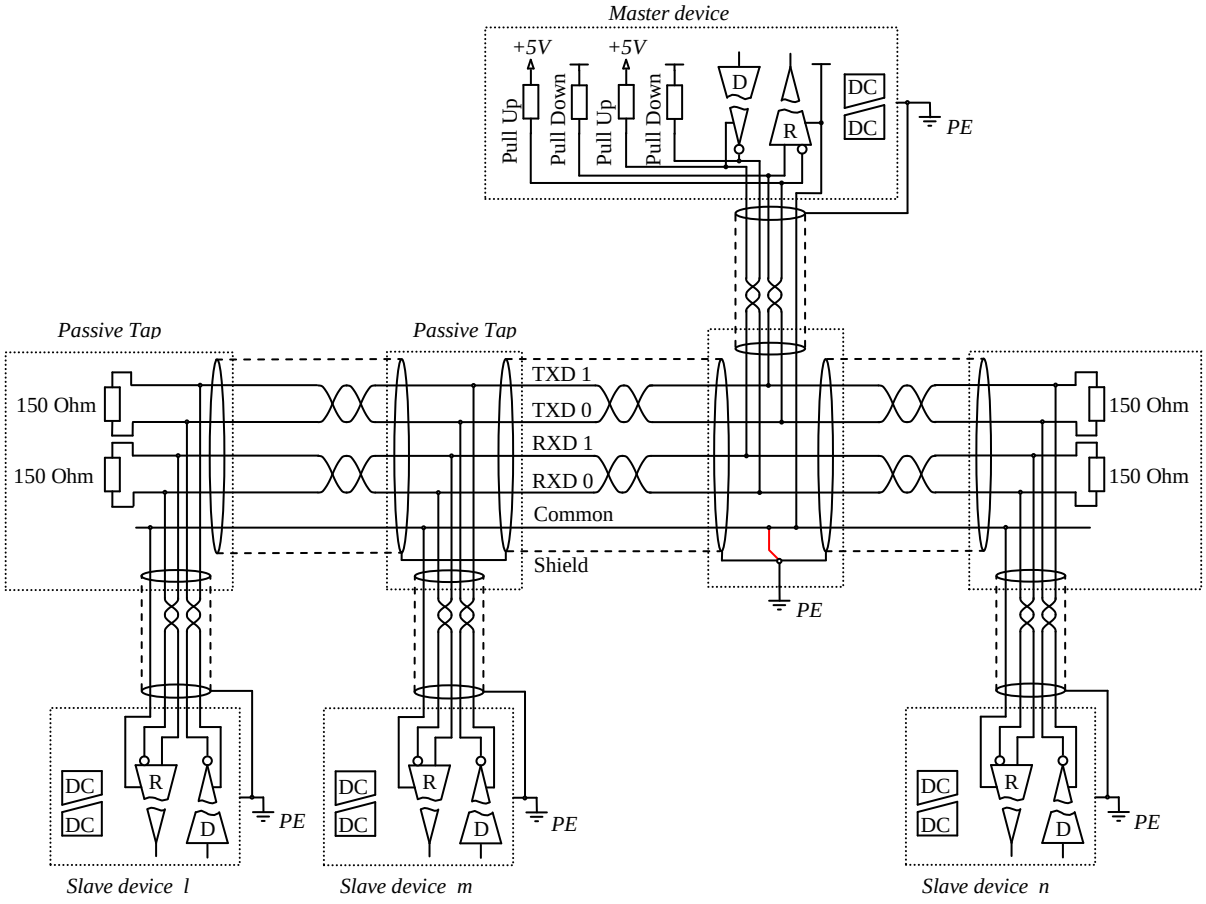



**Figure 5: Signal definition – four-wire-system**

The following Table 5 defines the signals of **Figure 5**.

| Desig-nation | for Device | | Contact at Partici-pant | EIA/TIA-485 Designation | | Description |
|---|---|---|---|---|---|---|
| | Master | Slave | | Master | Slave | |
| TXD 1 | In | Out | yes | B’ | B | Transmitter terminal 1 (+), voltage Vb (Vb > Va for binary 1 and OFF-state respectively) |

| Desig-nation | for Device | | Contact at Partici-pant | EIA/TIA-485 Designation | | Description |
|---|---|---|---|---|---|---|
| | Master | Slave | | Master | Slave | |
| TXD 0 | In | Out | yes | A’ | A | Transmitter terminal 0 (–), voltage Va (Va > Vb for binary 0 and ON- state respectively) |
| RXD 1 | Out | In | yes | B | B’ | Receiver terminal 1 (+), voltage Vb’ (Vb’ > Va’ for binary 1 and OFF-state respectively) |
| RXD 0 | Out | In | yes | A | A’ | Receiver terminal 0 (–), voltage Va’ (Va’ > Vb’ for binary 0 and ON- state respectively) |
| Common | - | - | yes | C/C’ | C/C’ | Reference potential, optionally serving as supply potential |
| Shield | - | - | yes | - | - | Cable shield |

**Table 5: Signal definition – four-wire-system**

### 3.4.1 **Bus cable in RS485-four-wire-systems**

There exist two variants:

- a cable with two balanced twisted pairs of conductors, a separate conductor and external shield,
- a cable with three balanced twisted pairs of conductors and with external shield.

The conductor size must generally be chosen for lengths of at least 1,000 m. Thus, for all conductors including the shield a gauge of at least AWG 24 must be provided. The characteristic impedance must be no less than 100 Ohms in the frequency range up to 100 kHz (with a baud rate of 19,200).

Basically, also cables with several twisted pairs of conductors may be used such as STP category 5 cables (4 pairs).

### 3.4.2 **Derivation cable in RS485-four-wire-system**

The derivation cable must have:

- at least two balanced twisted pairs of conductors for the differential data signal,
- a separate conductor or twisted pair of conductors for the reference potential (common), and
- an external shield.

For the derivation, for instance, the same cable can be used as for the bus.

Generally, also cables with several twisted pairs of conductors can be used.

Every derivation cable connects one or more participants to the bus (by passive taps). Therefore, the derivation cable includes

- the connection to the passive tap (connector, screw terminals, etc.), and
- the connection and connections to the participants (connectors, screw terminals, etc.) respectively.

For every participant connection, particular attention should be paid the device specific pin assignments.

## 3.5 RS232 Operation

With RS232 operation, an unbalanced data transmission is basically given using a separate signal line for either direction, i.e. for transmission from Master to Slave and from Slave to Master. This will also require an additional line with the reference potential (Common).

In RS232 operation, the Modbus standard is characterized by the following peculiarities:

- RS232 operation is only provided for point-to-point connections.
- Only a shielded cable may be used as a bus line
- Capacitance between the signal line and the Common line may not exceed 2,500 pF. This implies that the length of the bus line will be limited.
- Neither termination nor polarization of the bus.

The Slave devices by HEINZMANN support RS485 in two-wire and four-wire operation. Applications in RS232 operation will require additional RS232-RS485 interface adapters.

## 3.6 Slave participant

The maximum number of devices to be connected to a bus without repeater is up to 31 Slave devices plus one Master device.

All Slave participants must support:

- the 2-wire or 4-wire RS485 standard for two-wire-systems, and
- the 4-wire RS485 standard for four-wire-systems

Every participant must have a potential separation. By this, an electrically isolated communication is established offering these advantages:

- it is suitable for relatively long distances (up to 1,200 m), especially for industrial applications,
- it provides increased immunity against electromagnetic noise, and
- it inhibits larger currents from flowing in case of magnetic disturbances and different earthing potentials.

## 3.7 Master participant

By principle, only one Master participant is allowed on the bus. It must support:

- the 2-wire or 4-wire RS485 standard for two-wire-systems, and
- the 4-wire RS485 standard for four-wire-systems

It must also have a potential isolation. Normally, the Master device implements a bus polarization, see also section ↑*3.11 Line polarization*.

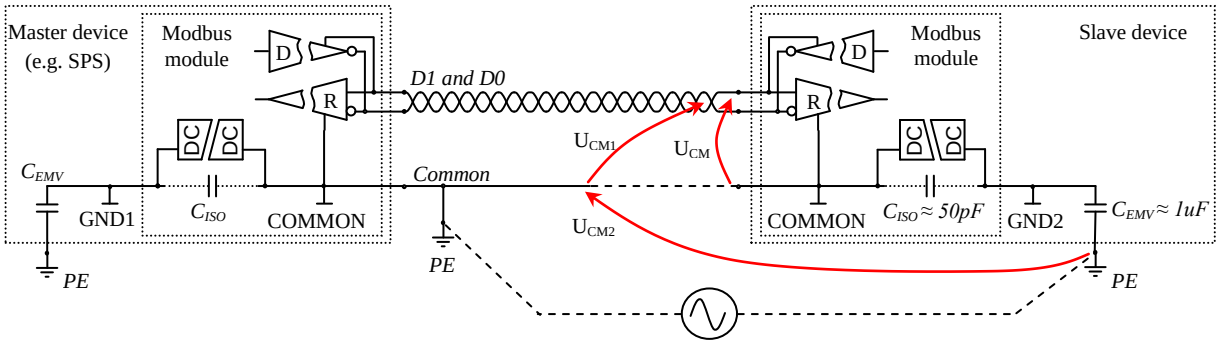
## 3.8 Reference potential

By means of a separate line all bus participants are pulled onto a common reference potential.

The Modbus standard requires this line to be connected with the protective earthing conductor (PE – Protective Earth) preferably on the Master or on its tap. There is only one single earthing point allowed. This rule is observed by the examples in **Figure 4**, **Figure 5**, **Figure 7** and **Figure 8**.


Note

*Although symmetric data transmission as compared with RS232 (asymmetric transmission) shows a relatively greater immunity against interferences and the signal level is to be directly recognized at the data lines, communication is additionally supported by the reference potential (Common).*




$U_{CM1}$ – Common mode voltage from transmitter with interference induced among the conductors within the bus cable.

$U_{CM2}$ – Interference induced between Common line and protective earthing conductor.

**Figure 6: Reference potential**

Note

*The indicated common mode voltage $U_{CM}$ must not exceed the level of -7/+12 V (specific working range). The reference potential (Common) is to be connected to all participants. Then $U_{CM} = U_{CM1}$ will hold for interferences.*

## 3.9 Shield

The Modbus standard requires a shielded cable.

According to this standard, the shield of every cable must be connected at one end to the protective earthing conductor (PE). The shield of any cable is to be connected either to the shield of the other cable or at one of its extremities to protective earthing. The admissible variants are shown in **Figure 4**, **Figure 5**, **Figure 7** and **Figure 8**.

According to our recommendations, the optimum shield configuration will be best as follows:

- A common shield for the bus cables. The only earthing point must preferably be located on the Master or its tap.
- A separate shield for the derivation cables. The only earthing point must be located in the participant's cubicle.

These rules are observed by the examples in **Figure 4**, **Figure 5** and **Figure 8**.

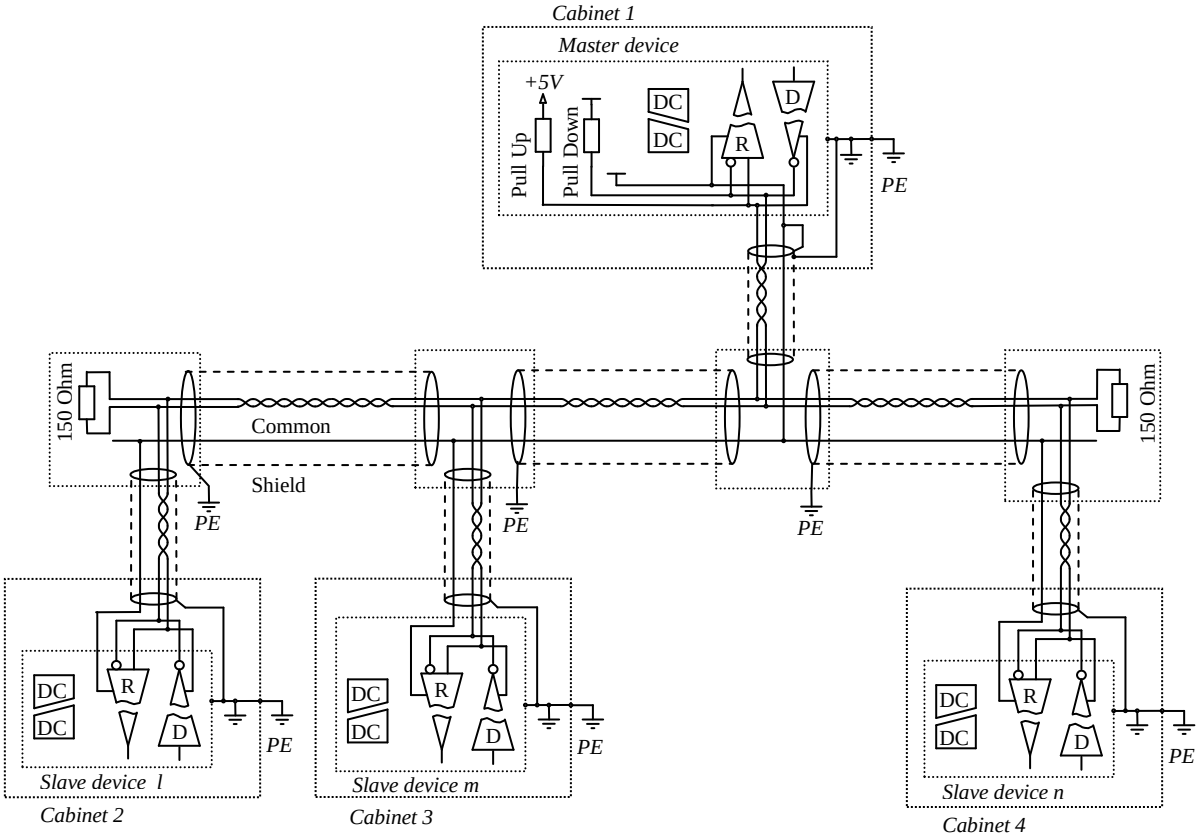



**Figure 7: Reference potential and shield**

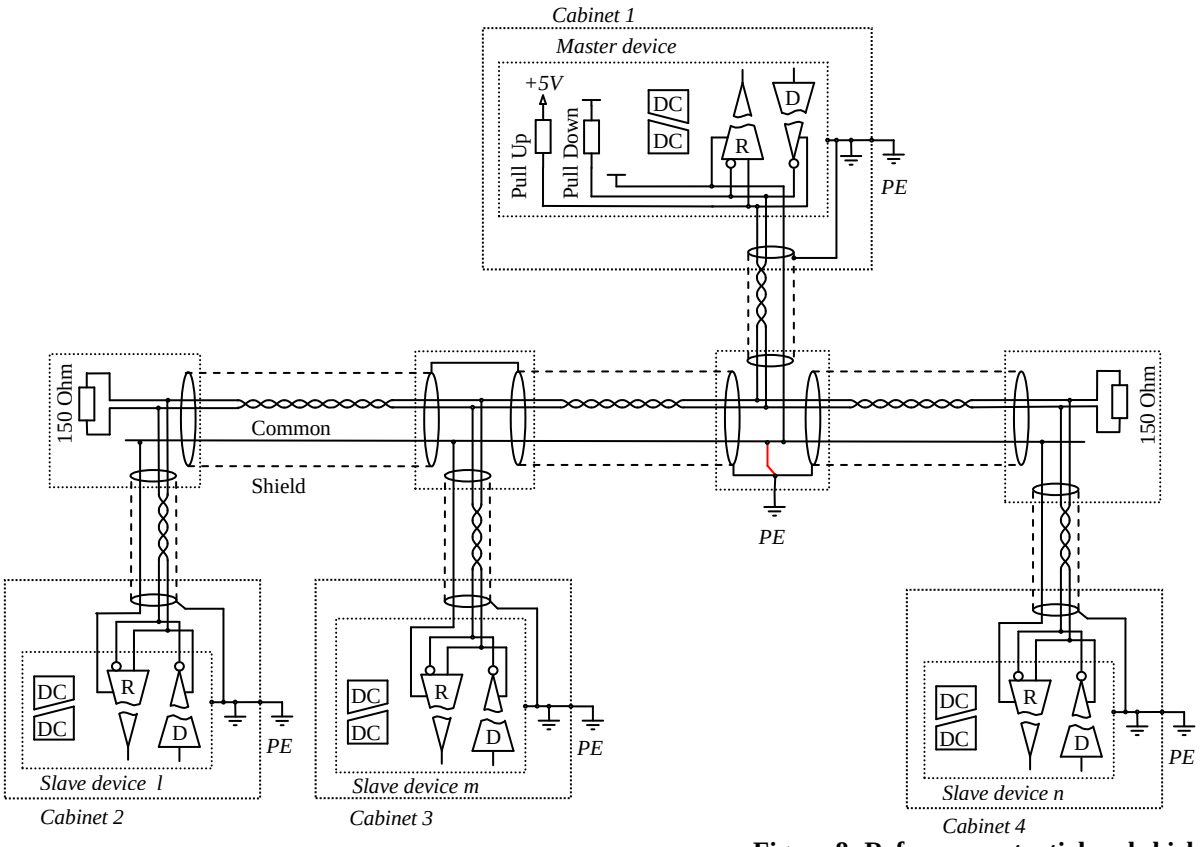
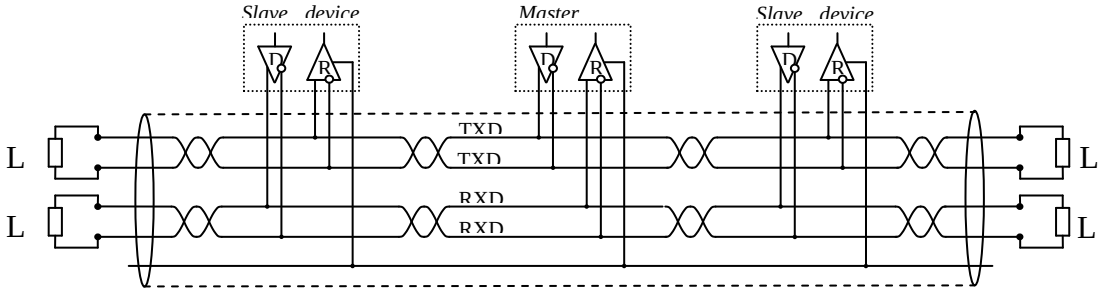



**Figure 8: Reference potential and shield**

The external shield serves as an electrostatic and electromagnetic shield and must not be used as a conductor for the reference potential. The shield must not have a second electric connection to the earthing bus or to the casing.

## 3.10 Line termination

Termination of the data line is a measure recommended according to Modbus standard.




In order to obtain optimum transmission quality and/or signal quality, particularly over long distances, the data line is to have a termination as a twisted data conductor pair at both extremities according to **Figure 9**. For this, termination load must correspond to the characteristic impedance of the data line.

**Figure 9: Line termination in four-wire-system**

The Modbus standard permits two quasi optimum termination methods:

- AC termination – on either end a serial RC element with capacitor: 1 nF, 10 V minimum and with resistor 120 Ohms, 0.25 W.
  Advantage: – maximum signal level up to 5 Vpp.
  Disadvantage: – poor reflection behaviour.
- DC termination – on either end a resistor 150 Ohm, 0.5 W.
  Advantage: – good reflection behaviour.
  Disadvantage: – smaller signal level (e.g., silent state of about 0.27 Vpp).

Preferably, DC termination should be chosen. It is depicted both in **Figure 4** and **Figure 5**.

Termination can be implemented:

- at the bus end in the device connection in the daisy-chain structure, for this see also section ↑*3.1 Bus topology*.
- at the bus end in the participant in the daisy-chain structure if this possibility is provided in the participant,
- in the passive tap at the bus end (see **Figure 4** and **Figure 5**) or
- in the active tap at the bus end.

The termination must not be installed in the derivation cable.

## 3.11 Line polarization

A remote data cable needs a polarization for each pair of data conductors which is to be implemented preferably in the Master device or in its tap. Consequently, there is no need for polarization in the Slave devices.

Polarization is to be built using the following components:

- Pull-up resistors that are to be connected against 5 V DC and to signal + (TXD1, RXD1 or D1), and
- Pull down resistors that are to be connected against the reference potential (Common) and to signal + (TXD1, RXD1 or D1).

For all polarization resistors identical values should be chosen that may range from 450 Ohms to 650 Ohms. Polarization requires a 5 V DC supply. A detailed configuration of the polarization is depicted in **Figure 10** and outlined in **Figure 4** and **Figure 5**.

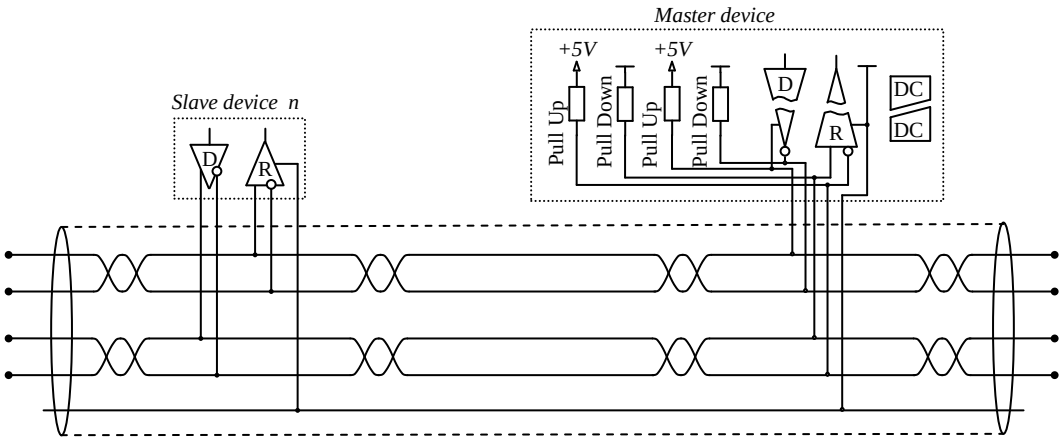



**Figure 10: Line polarization in a four-wire-system**

If the polarization is implemented in the active/passive tap of the Master device the tap must be supplied 5 V DC either from the Master device or from an electrically separated source.

Note

*Line polarization serves to set the OFF-state, where the OFF-state designates the silent state when there is no transmission from any participant. Actually in silent state, however, a polarization current between 3.6 and 5.1 mA DC must be flowing. With a DC termination by 150 Ohms, this will in silent state cause a differential signal of 0.27 to 0.38 V DC. The maximum polarization current will increase interference immunity.*

## 3.12 Bus accessories

The taps constitute an important element of the bus. As a passive tap for Modbus every suitable bus tap can be used (e.g., Fieldbus, CAN, Ethernet).

The possibility of termination and polarization is already provided in certain taps, such as

- Schneider-Electric, TSX SCA 64 connection box,
- Schneider-Electric, TSX SCA 50,
- Phoenix Contact, PSM PTK (passive RS-485 T-tap),
- Weidmüller, Fieldbus components.

## 3.13 Device side connection module

In accordance with the EIA/TIA-485/422 standard, the HEINZMANN connection module is designed as a 4-wire interface with separate transmission and reception lines. Due to this, both connection types as a 2-wire and a 4-wire interface can be used. For 2-wire operation, this can be achieved by simply paralleling two lines RX and TX of identical polarity in the connector.

Electrical isolation between bus connection and control unit is provided. On the communication side, the Slave devices by HEINZMANN have no contacting between the reference potential and protective earthing (PE) nor between the case of the connection and protecting earthing (PE).


**Electric shock hazard**

Contact with communication lines, shield or reference potential might cause electric shock.

> Never touch communication lines, shield or the reference potential

## 3.14 Connector plug and socket

The device side connector socket is a:

- 9 pole Sub-D female connector.

The bus or cable side coupler is a:

- 9 pole Sub-D male connector.

Regarding the specifications concerning the pin assignment of the connectors see section ↑6 *Technical data*.

# 4 Module functions

The HEINZMANN control unit is always defined as a Slave and by request from the connected Master can be made to execute a data transfer as determined by the function code. In accordance with the function code, the reply following this request may consist of a read-out of data, an acknowledgement of data received, an exception response or no response.

## 4.1 Device address

The device addresses of the participants (ID) of the network must be set by the user in the participant device. Every ID may be assigned only once within a bus segment.


Note

*In assigning device addresses it is particularly important to ensure that one and the same address is not assigned to several devices. Such a failure may result in abnormal bus performance so that the Master will not be able to communicate with all Slaves connected to the bus.*


Note

*Broadcast in Multi-Slave operation or in transmission only mode: Requests under Slave address 0 are accepted by the HEINZMANN control unit but will be without any function! The reaction of the control unit will be no response.*

## 4.2 Supported function codes

The following Table 6 lists all supported functions from the Modbus function stock together with a brief description. Whenever an unknown function code is requested by the Master the corresponding exception response will be returned.

| Function Code | Function Designation | Brief Description |
|---|---|---|
| 0x03 | *Read Holding Register* | Read access to one or more data words, for transmission of parameter values or measurements of the control unit |
| 0x06 | *Write Single Register* | Write access to a data word |
| 0x08 | *Diagnostics* | Diagnostic function, e.g., reading out the diagnostics counters |
| 0x10 | *Write Multiple Register* | Write access to one or more data words |

**Table 6: Function codes**

The diagnostics function is confined to the sub-function codes that are specified in Table 7; any other sub-functions are not supported and are rejected through an exception response. The request data field or data word is to be assigned the value 0x0000 except for the sub-function code 0x0000 which supports any length and selectable values of the data words.

This functionality serves for diagnosing the communication result and allows to individually or globally reading out and reset the counter states that are being incremented with regard to success, error of communication, etc.

| Sub-function code | Designation | Brief Description |
|---|---|---|
| 0x0000 | *Return Query Data* | Request is echoed back |
| 0x0002 | *Return Diagnostics Register* | The data of the sub-functions codes 0x0B to 0x12 are returned in this order |
| 0x000A | *Clear Counters and Diagnostic Registers* | The counter states of the sub-function codes 0x0B to 0x12 are reset to zero |
| 0x000B | *Return Bus Message Count* | Quantity of valid messages received is returned |
| 0x000C | *Return Bus Communication Error Count* | Quantity of errors encountered in receiving messages is returned |
| 0x000D | *Return Bus Exception Error Count* | Quantity of transmitted exception responses is returned |
| 0x000E | *Return Slave Message Count* | Quantity of transmitted messages is returned |
| 0x000F | *Return Slave No Response Count* | Quantity of received messages requiring no response is returned |
| 0x0010 | *Return Slave NAK Count* | Zero is returned |
| 0x0011 | *Return Slave Busy Count* | Zero is returned |
| 0x0012 | *Return Bus Character Overrun Count* | Quantity of messages is returned that due to a character overrun condition were not correctly received |

**Table 7: Sub-functions of the diagnostic function**

## 4.3 Supported exception codes

The possible exception codes that may be contained in an exception response derive from the function codes that are supported. An exception response is returned as a reaction to any request from the Master when the specifications concerning the data to be transmitted exceed the device's internal address and data ranges or when a function code is not supported. Table 8 offers a list of the supported exception codes of a HEINZMANN control unit.

| Exception Code | Designation | Brief Description |
|---|---|---|
| 0x01 | *Illegal Function* | Function code is not supported |
| 0x02 | *Illegal Data Address* | Data address exceeds the internal data range as provided by the device |
| 0x03 | *Illegal Data Value* | Quantity of data exceeds the internal data range as provided by the device |
| 0x04 | *Slave Device Failure* | Device can not process the data |

**Table 8: Exception codes**

# 5 Parameterization and commissioning

Parameterization of the module is to be performed exclusively by means of the software of the HEINZMANN basis device, which will require either DcDesk 2000 as a Windows® programme for visualizing and configuring the input and output data of digital HEINZMANN control devices, or the Hand Programmer HP-03-03 as a diagnostics tool. For more details, please refer to

- *Operation Manual Communication Programme DcDesk 2000, Manual No. DG 00 003–e*,
- *Operating Instructions Hand Programmer HP-03-03, Manual No. DG 04 002-e*,
- *PRIAMOS / HELENOS Basic Information 2000 for Digital Governors Level 6, Manual No. DG 00 001-e* and
- *THESEUS Control Systems for Electronically Controlled Generator Sets in Isolated and Mains Parallel Operation, Manual No. DG 97 002-e*.

## 5.1 Configuration of interface

The configuration of the interface requires parameterizing the baud rate used, the transmission type and the device address within the bus system. The transmission type includes specifications concerning the parity bit and the number of stop bits. In addition, it is necessary to activate the complete Modbus functionality within the HEINZMANN control device. To do so, the user will have to pre-setting the following parameters and function parameters:

| | | |
|---|---|---|
| 21800 | *Modb:Baudrate* | Parameter baud rate, |
| 21801 | *Modb:SlaveID* | Specific device address as Slave, |
| 25800 | *ModbusOn* | On/Off switch for Modbus functionality, |
| 25801 | *Modb:ParityBitOn* | Selection transmission type with or without parity bit, |
| 25802 | *Modb:ParityOddOrEven* | Selection of odd or even parity if parity bit is selected |
| 25803 | *Modb:OneOrTwoStopBit* | Selection of one or two stop bits |
| 25804 | *Modb:ExtendedOn* | Selection of the Modbus functionality: parameter values can also be set in the extended version |

The measurement

| | | |
|---|---|---|
| 23800 | *Modb:Baudrate* | Indication value baud rate |

will indicate whether the device was able to set the baud rate as selected by means of the respective parameter.

Note

*Any changes of the parameter 21800* Modb:Baudrate *or of the function parameters 25800* ModbusOn, *25801* Modb:ParityBitOn, *25802* Modb:ParityOddOrEven *and25803* Modb:OneOrTwoStopBit *will be effective only after saving the changes and resetting the control unit. The function parameter 25804* Modb:ExtendedOn *is only available in the Modbus Extended Version.*

Table 9 gives an overview of the options for setting the transmission types with regard to parity bit and stop bits.

| | No parity with one stop bit | No parity with two stop bits | Even parity with one stop bit | Odd parity with one stop bit |
|---|---|---|---|---|
| 25801 *Modb:ParityBitOn* | 0 | 0 | 1 | 1 |
| 25802 *Modb:ParityOddOrEven* | X | X | 0 | 1 |
| 25803 *Modb:OneOrTwoStopBit* | 1 | 0 | 1 | 1 |

**Table 9: Setting for transmission type**

Example of parameterization:

Modbus functionality is to be activated with the baud rate set to 9600 Baud and the Slave device address to 1. The data transmission type for a character is to be selected as including one bit for even parity and one stop bit. It should be noted that after making modifications to the settings for the baud rate, the activation of the functionality and the transmission type these changes must be saved and the control unit reset.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *21800* | *Modb:Baudrate* | *9.600* | *kBaud* |
| *21801* | *Modb:SlaveID* | *1* | |
| *25800* | *ModbusOn* | *1* | |
| *25801* | *Modb:ParityBitOn* | *1* | |
| *25802* | *Modb:ParityOddOrEven* | *0* | |
| *25803* | *Modb:OneOrTwoStopBit* | *1* | |

Indication

| | | | |
|---|---|---|---|
| *23800* | *Modb:Baudrate* | *9.600* | *kBaud* |

The standard default settings on shipping the device are

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *21800* | *Modb:Baudrate* | *9.600* | *kBaud* |
| *21801* | *Modb:SlaveID* | *247* | |
| *25800* | *ModbusOn* | *1* | |
| *25801* | *Modb:ParityBitOn* | *1* | |
| *25802* | *Modb:ParityOddOrEven* | *0* | |
| *25803* | *Modb:OneOrTwoStopBit* | *1* | |

On request, the shipping state can be modified in dependence of customer wishes and project requirements.

## 5.2 Reading data

A read access to one or more data words serves for the transmission of parameter values or measurements of the control unit to the Master device on the Modbus. For this purpose, the function code 0x03 *"Read Holding Register"* is to be used, see also section *↑4.2 Supported function codes*.

As regards curve parameters, a data array ranging from 29200 to 29299 *Modb:TxParamSet(x)* (or *Modb:ParamSet(x)* for Extended Version) is available for the user and can be accessed via the function code 0x03. The start address specified in the request message corresponds with the index of the data array which is limited to 100 entries. The quantity of data to be transmitted beginning with this index is given by the number of registers specified in the request message.

The data field is to be assigned the parameter numbers whose measuring or parameter values are to be transmitted. Particular care should be taken to enter the numbers consecutively beginning with the index zero and without gaps. Note that from the first invalid parameter number on all subsequent entries will be ignored. The following parameter numbers are defined as invalid:

- parameter number zero,
- any parameter number unknown to the control unit, or
- any parameter number associated with a parameter whose level is higher than the access level for the Modbus which on principle is limited to four.


Note

*By being invariably set to four the access level for the Modbus allows to transmit all measurements and indication values as well as all application specific parameters.*

The user is informed about the maximum number of data words that can be read out by the measurement

23810 *Modb:NoOfTxParams* Number of valid parameter numbers in the data array.

Example of parameterization:

Speed, oil pressure and oil temperature are to be read out via Modbus. A control unit of the HELENOS series is being used.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *29200* | *Modb:TxParamSet(0)* | *2000* | |
| *29201* | *Modb:TxParamSet(1)* | *2905* | |
| *29202* | *Modb:TxParamSet(2)* | *2909* | |
| *29202* | *Modb:TxParamSet(3)* | *0* | |

Indication

| | | | |
|---|---|---|---|
| *23810* | *Modb:NoOfTxParams* | *3* | |
| *2000* | *Speed* | *1500.2* | *1/min* |
| *2905* | *OilPressure* | *3.15* | *bar* |
| *2909* | *OilTemp* | *-10.2* | *°C* |

External Value Ranges

| | | | |
|---|---|---|---|
| *2000* | *Speed* | *0.0..4000.0* | *1/min* |
| *2905* | *OilPressure* | *0.00..10.00* | *bar* |
| *2909* | *OilTemp* | *-100.0..1000.0* | *°C* |

The data are transmitted by their external value ranges. As regards the value ranges of the different parameters the user should refer to the publications describing the HEINZMANN control devices or read them out directly from the control unit using one of the HEINZMANN diagnostics tools. To interpret the data words transmitted correctly, account must be taken of the number of decimal places and of whether the value range can also be negative.

In the above example, the data word for speed will be transmitted by the value 15002 as obtained from $1500.2 \cdot 10$ for one decimal place, that for oil pressure by the value 315 as obtained from $3.15 \cdot 100$ for two decimal places, and that for oil temperature by –102 as obtained from $-10.2 \cdot 10$ for one decimal place.

Since speed and oil pressure can assume positive values only, it is but the number of digital places that must be taken into account as powers of ten. Likewise, if oil temperature is interpreted as already signed, it will also be the number of digital places that is to be evaluated as power of ten.

If oil temperature is evaluated as not yet signed the above example will yield a data word having the value 65,434. Any values greater than or equal to 32,768 are to be interpreted as negative values provided the external value range admits of negative values. The correct signed value is obtained as the difference between the data word and 65,536, and as a

second step it is but the number of digital places that must be taken into account to calculate the final value.

Note

*If in the above example any request should include a start address equal or greater than two or if the quantity of data to be read exceeds two data words, an exception response will be transmitted.*

## 5.3 Write data

A given data field can be written to by means of the function codes 0x06 "*Write Single Register*" (single write mode) and 0x10 "*Write Multiple Register*" (multiple write mode), see also section ↑*4.2 Supported function* codes. The write access to one or more data words allows to further process received data within the control device as external input quantities.

The implementation of the HEINZMANN Modbus differs fundamentally between the basic and the extended version. In contrast to the extended version, the basic version only has write access rights for the Modbus switching functions and sensors, which are predetermined and fixed. The following Table 10 provides information about the write access rights associated with the different versions.

| Parameter Lists | Basic | Extended |
|---|---|---|
| Parameter | - | X |
| Functions | - | X |
| Curves | - | X |
| Measurements (general) | - | - |
| Measurements (Modbus functions) | - | X |
| Measurements (Modbus binary values) | X | X |
| Measurements (Modbus sensor values) | X | X |

**Table 10: Write access rights**

Measurements (Modbus functions)

23815 *Modb:ResetDevice* Force reset of the control unit

23816 *Modb:SaveParameter* Save parameter values in the control unit

Measurements (Modbus binary values)

23820 *Modb:RxBinary* Binary values 16 bits wide

Measurements (Modbus sensor values)

23821 *Modb:RxSensor(0)* to
23824 *Modb:RxSensor(3)* Sensor values

### 5.3.1 **Basic version**

By standard, the size of the writeable data field is five words. Due to definition, the first word can be used for 16 binary switch input values and every subsequent data word for a sensor input value. This configuration is shown in Table 11 and may on request be extended according to customer or application needs.

| **Data Field** | | | |
|---|---|---|---|
| **Index** | | | **Type** |
| 0 | 8 Bit (high) | 8 Bit (low) | switch |
| 1 | 16 Bit | | sensor |
| 2 | 16 Bit | | sensor |
| 3 | 16 Bit | | sensor |
| 4 | 16 Bit | | sensor |

**Table 11: Write data**

The data received by the control unit can be viewed via these indication values

23820 *Modb:RxBinary* Binary values 16 bits wide, and

23821 *Modb:RxSensor(0)* to
23824 *Modb:RxSensor(3)* Sensor values.

### 5.3.2 **Extended version**

In the extended version, the sensor values and binary values are freely configurable via the 29200 Modb:ParamSet(x) data field. In addition to sensor and switching functions, parameters with an access level of less than or equal to 4 can also be specified. The data field that is used to read out data (see ↑ 5.2 Reading data) should be populated with the corresponding parameter numbers, the parameter values of which should also be specified.

The data field can be activated using the function code 0x06 or 0x10, and the parameter and measured values specific to Modbus can be set. The start address given in the output corresponds to the data field index, for which the data field is limited to 100 entries. The number of data entries that will be transferred from the index is determined by the number of registers that should be populated, which is given in the output. If an array is to be populated, all of the registered parameters in the given array must possess write access. If this requirement is not fulfilled, a corresponding exception code (0x04 Slave Device Failure) will be returned.

Example of parameterization:

As in the above example, the speed, oil pressure and oil temperature must be read via Modbus. It must be possible to specify the PID parameters for the speed governor 100 Gain, 101 Stability and 102 Derivative using Modbus. A control unit of the HELENOS series is being used.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *29200* | *Modb:ParamSet(0)* | *2000* | |
| *29201* | *Modb:ParamSet(1)* | *2905* | |
| *29202* | *Modb:ParamSet(2)* | *2909* | |
| *29203* | *Modb:ParamSet(3)* | *100* | |
| *29204* | *Modb:ParamSet(4)* | *101* | |
| *29205* | *Modb:ParamSet(5)* | *102* | |
| *29206* | *Modb:ParamSet(6)* | *0* | |
| Activation | | | |
| *25804* | *Modb:ExtendedOn* | *1* | |
| Indication | | | |
| *23810* | *Modb:NoOfParams* | *6* | |
| *2000* | *Speed* | *1500.2* | *1/min* |
| *2905* | *OilPressure* | *3.15* | *bar* |
| *2909* | *OilTemp* | *-10.2* | *°C* |

### 5.3.3 **Timeout monitoring**

If binary and/or sensor values are to be transmitted to the HEINZMANN control unit in write mode, it should be noted that these values must be transmitted periodically in cyclical order to detect any failures or damages of the data transmission equipment. For this purpose, a time-out is initiated with the first write access whose time limit is to be set by means of the parameter 21820 *Modb:RxTimeOut*. With every valid reception of a request via the functions codes 0x06 or 0x10 the time-out is re-initialized. If the write access fails to occur within the pre-set reception time limit the error flag 3074 *ErrModbusComm* is set.


Note

*If a failure of Modbus communication occurs the standard reactions of the HEINZMANN control units to sensor errors will be applied for the respective sensor value. The binary values are assumed to be zero, and due to this the assigned switching functions are reset.*

If it is only parameters (e.g. at the start of communication) that are to be specified in write mode, timeout monitoring is not strictly necessary. In this case, timeout monitoring can be deactivated by setting the time limit parameter to zero.

For further details concerning the parameterization of sensors and switching function, refer to the chapters *Sensors* and *Inputs and Outputs* of the respective Basis Information. In this brochure, it is only the specific extension of the particular functionality required for Modbus that will be dealt with.

### 5.3.4 **Assignment of sensor values to sensors**

The assignment of sensor values as inputs to sensors via Modbus is achieved by entering the desired channel numbers, in this case the respective index plus one, in the associated parameters from 900 *Assign…* onward. The selection of the sensor as Modbus sensor value is made by specifying the channel type in the parameters from 4900 *ChanTyp…* onward. For Modbus, it is always the value '6' that is to be specified as channel type.

49xx *ChanTyp…* = 6 Sensor value is being received via Modbus

The transmission of any sensor value from the Master to the HEINZMANN control unit operating as a Slave will on principle be made within the internal value range which is defined as ranging from 0 to 65535.


Note

*The scaling of the sensor value from 0% to 100% is equivalent to values from 0 to 65535 being transmitted via Modbus. In the above example this will yield the value 32767 for 50%.*

Example of parameterization for basic version:

The power set point is to be received via Modbus as second sensor value. The sensor value is to be updated cyclically every two seconds. A control unit of the THESEUS series is being used.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *900* | *Assign_PowerSetpoint* | *2* | |
| *980* | *PowerSetpointLow* | *0.0* | *%* |
| *981* | *PowerSetpointHigh* | *100.0* | *%* |
| *4900* | *ChanType_PowerSetp* | *6* | |
| *21820* | *Modb:RxTimeOut* | *2.5* | *s* |

Indication

| | | | |
|---|---|---|---|
| *2900* | *PowerSetpoint* | *50.0* | *%* |
| *3074* | *ErrModbusComm* | *0* | |
| *23821* | *Modb:RxSensor(0)* | *0.0* | *%* |
| *23822* | *Modb:RxSensor(1)* | *50.0* | *%* |
| *23823* | *Modb:RxSensor(2)* | *0.0* | *%* |
| *23824* | *Modb:RxSensor(3)* | *0.0* | *%* |

Example of parameterization for extended version:

The power set point is to be received via Modbus as second sensor value. The sensor value is to be updated cyclically every two seconds. A control unit of the THESEUS series is being used.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *900* | *Assign_PowerSetpoint* | *2* | |
| *980* | *PowerSetpointLow* | *0.0* | *%* |
| *981* | *PowerSetpointHigh* | *100.0* | *%* |
| *4900* | *ChanType_PowerSetp* | *6* | |
| *21820* | *Modb:RxTimeOut* | *2.5* | *s* |
| *29200* | *Modb:ParamSet(0)* | *23822* | |
| | Activation | | |
| *25804* | *Modb:ExtendedOn* | *1* | |
| | Indication | | |
| *2900* | *PowerSetpoint* | *50.0* | *%* |
| *3074* | *ErrModbusComm* | *0* | |
| *23821* | *Modb:RxSensor(0)* | *0.0* | *%* |
| *23822* | *Modb:RxSensor(1)* | *50.0* | *%* |
| *23823* | *Modb:RxSensor(2)* | *0.0* | *%* |
| *23824* | *Modb:RxSensor(3)* | *0.0* | *%* |

### 5.3.5 **Assignment of binary values to switching functions**

The assignment parameters from parameter number 810 *Funct…* on may be used exclusively for switching functions whose state is to be modified via digital hardware inputs. With respect to the extended functionality of the switching functions the assignment parameters have been doubled by introducing the parameter numbers from 20810 *Comm…* onward. These parameters can now be used for switching functions that are to be set via some communication mode.

A binary value to be transmitted via Modbus can be easily assigned to a switching function by entering the bit number in the corresponding assignment parameter. Similarly to sensor assignment, the channel type of switching functions is to be specified for Modbus by ‘6’.

248xx *ChanTyp…* = 6 Binary value is to be received via Modbus

If the parameterization of a switching functions admits of both the hardware and the selected communication the two states will internally be combined by OR.

Example of parameterization for basic version:

By setting the binary input three via Modbus, operation is to be by fixed speed 1. The binary value is to be updated cyclically every ten seconds. A control unit of the HELENOS series is being used.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *20815* | *CommSpeedFix1* | *3* | |
| *21820* | *Modb:RxTimeOut* | *11.0* | *s* |
| *24815* | *ChanType_SpeedFix1* | *6* | |

Indication

| | | | |
|---|---|---|---|
| *2815* | *SwitchSpeedFix1* | *1* | |
| *3074* | *ErrModbusComm* | *0* | |
| *23820* | *Modb:RxBinary* | *04* | *Hex* |

Example of parameterization for extended version:

By setting the binary input three via Modbus, operation is to be by fixed speed 1. The binary value is to be updated cyclically every ten seconds. A control unit of the HELENOS series is being used.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *20815* | *CommSpeedFix1* | *3* | |
| *21820* | *Modb:RxTimeOut* | *11.0* | *s* |
| *24815* | *ChanType_SpeedFix1* | *6* | |
| *29200* | *Modb:ParamSet(0)* | *23820* | |

Activation

| | | | |
|---|---|---|---|
| *25804* | *Modb:ExtendedOn* | *1* | |

Indication

| | | | |
|---|---|---|---|
| *2815* | *SwitchSpeedFix1* | *1* | |
| *3074* | *ErrModbusComm* | *0* | |
| *23820* | *Modb:RxBinary* | *04* | *Hex* |

### 5.3.6 **Assignment of Modbus functions (extended version only)**

The extended version offers the facility to carry out specific functions in the control unit. The parameter number must be entered into the data field to enable the relevant function to be carried out. The following Modbus functions are possible:

| | | |
|---|---|---|
| 23815 | *Modb:ResetDevice* | Force reset of the control unit |
| 23816 | *Modb:SaveParameter* | Save parameter values in the control unit |

Example of parameterization:

The parameters must be stored in the control unit using Modbus.

| Number | Parameter | Value | Unit |
|---|---|---|---|
| *29200* | *Modb:ParamSet(0)* | *23815* | |

Activation

| | | | |
|---|---|---|---|
| *25804* | *Modb:ExtendedOn* | *1* | |

Indication

| | | |
|---|---|---|
| *3851* | *LastIdentifier* | *94* |
| *23816* | *Modb:SaveParameter* | *1* |

## 5.4 Diagnostic counter

For the HEINZMANN control devices the following diagnostic counters have been implemented according to Modbus specification. They can also be read out via the function code 0x08 "*Diagnostics*", see also section ↑*4.2 Supported function codes*:

| | | |
|---|---|---|
| 23801 | *Modb:BusMessageCnt* | Quantity of valid messages received, |
| 23802 | *Modb:CommErrorCnt* | Quantity of faulty messages received, |
| 23803 | *Modb:ExceptErrCnt* | Quantity of exception messages transmitted, |
| 23804 | *Modb:SlaveMessageCnt* | Quantity of messages transmitted, |
| 23805 | *Modb:SlaveNoRespCnt* | Quantity of No response messages, |
| 23806 | *Modb:BusCharOvrCnt* | Quantity of messages not correctly received due to character overrun. |

# 6 Technical data

| | |
|---|---|
| Interface: | EIA/TIA-485/422 standard |
| | 4-wire interface |
| Insulation: | electrically separated |
| Insulation voltage: | 1000 $Volts_{(RMS)}$, 1 min |
| Signal voltages against reference potential (Common): | –7 V to +12 V D.C. |
| Interference immunity (ESD): | to EN 61000-4-2 (intended) |
| Connector: | D-Sub 9 (male) |
| Signal assignment: | according to Modbus Specification 4-wire interface |

| Pin | Signal | EIA/TIA-485/422 | Description |
|---|---|---|---|
| 8 | RXD0 | A’ | Receiver connection 0 |
| 4 | RXD1 | B’ | Receiver connection 1 |
| 9 | TXD0 | A | Transmitter connection 0 |
| 5 | TXD1 | B | Transmitter connection 1 |
| 2 | VP | --- | +5 V DC (depending on version) |
| 1 | Common | C/C’ | Signal reference potential |
| Case | --- | --- | potential free |

**Table 12: Signal assignment**

| | |
|---|---|
| Visual error indication: | none |
| Data transmission: | RTU Mode |
| Baud rate: | 9,600, 19,200 Baud |
| Parity: | no parity, even or odd parity |
| Stop bits | one or two |
| Function codes: | see section ↑*4.2 Supported function codes* |
| Exception codes: | see section ↑*4.3 Supported exception codes* |

# 7 Parameter description

## 7.1 List 1: Parameters

| | | | |
|---|---|---|---|
| **21800** | **Modb:Baudrate** | | |
| | Level: | 4 | Modbus |
| | Range: | 9.6..19.2 kBaud | Parameter baud rate, modification only effective after |
| | Page(s): | 32 | reset |
| **21801** | **Modb:SlaveID** | | |
| | Level: | 4 | Modbus |
| | Range: | 1..247 | Unique device address as Slave |
| | Page(s): | 32 | |
| **21820** | **Modb:RxTimeOut** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..60 s | Reception time limit for requests for writing binary |
| | Page(s): | 38 | values or sensor values |

## 7.2 List 2: Measurements

| | | | |
|---|---|---|---|
| **3074** | **ErrModbusComm** | | |
| | Level: | 1 | Modbus |
| | Range: | 0..1 | Error indication Modbus communication |
| | Page(s): | 38 | |
| **3174** | **SErrModbusComm** | | |
| | Level: | 1 | Modbus |
| | Range: | 0..255 or 0..1 | Error Counter or sentinel for the occurrence of |
| | | | 3074 *ErrModbusComm* |
| **23800** | **Modb:Baudrate** | | |
| | Level: | 4 | Modbus |
| | Range: | 9.6..19.2 kBaud | Indication value for the pre-set baud rate |
| | Page(s): | 32 | |
| **23801** | **Modb:BusMessageCnt** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..65535 | Quantity of valid received messages |
| | Page(s): | 42 | |
| **23802** | **Modb:CommErrorCnt** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..65535 | Quantity of faulty received messages |
| | Page(s): | 42 | |

| | | | |
|---|---|---|---|
| **23803** | **Modb:ExceptErrCnt** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..65535 | Quantity of exception messages transmitted |
| | Page(s): | 42 | |
| **23804** | **Modb:SlaveMessageCnt** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..65535 | Quantity of messages transmitted |
| | Page(s): | 42 | |
| **23805** | **Modb:SlaveNoRespCnt** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..65535 | Quantity of messages received requiring no response |
| | Page(s): | 42 | (here: only transmission message from Master) |
| **23806** | **Modb:BusCharOvrCnt** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..65535 | Quantity of messages not correctly received due to |
| | Page(s): | 42 | character overrun |
| **23810** | **Modb:NoOfTxParams** | | |
| | Level: | 4 | Modbus Basic Version |
| | Range: | 0..100 | Number of valid parameter numbers entered in the data |
| | Page(s): | 35 | field for read access |
| **23810** | **Modb:NoOfParams** | | |
| | Level: | 4 | Modbus Extended Version |
| | Range: | 0..100 | Number of valid parameter numbers entered in the data |
| | Page(s): | 35 | field for read access and write access |
| **23815** | **Modb:ResetDevice** | | |
| | Level: | 4 | Modbus Extended Version |
| | Range: | 0..1 | Modbus function: Force reset of the control unit |
| | Page(s): | 36, 41 | |
| **23816** | **Modb:SaveParameter** | | |
| | Level: | 4 | Modbus Extended Version |
| | Range: | 0..1 | Modbus function: Save parameter values in the control |
| | Page(s): | 36, 41 | unit |
| **23820** | **Modb:RxBinary** | | |
| | Level: | 4 | Modbus |
| | Range: | 0000..FFFF Hex | Binary values for write access |
| | Page(s): | 36 | |
| **23821** | **Modb:RxSensor(x)** | | |
| **to** | Level: | 4 | Modbus |
| **23824** | Range: | 0..65535 | Sensor values for write access |
| | Page(s): | 36 | |

## 7.3 List 3: Functions

| | | | |
|---|---|---|---|
| **25800** | **ModbusOn** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..1 | On/Off switch Modbus functionality. Modification will |
| | Page(s): | 32 | be effective after reset only |
| **25801** | **Modb:ParityBitOn** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..1 | Selection transmission mode with or without parity bit. |
| | Page(s): | 32 | Modification will be effective after reset only |
| | | | 0 = no parity bit |
| | | | 1 = with parity bit |
| **25802** | **Modb:ParityOddOrEven** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..1 | Selection odd or even parity with parity bit set. |
| | Page(s): | 32 | Modification will be effective after reset only |
| | | | 0 = even parity |
| | | | 1 = odd parity |
| **25803** | **Modb:OneOrTwoStopBit** | | |
| | Level: | 4 | Modbus |
| | Range: | 0..1 | Selection one or two stop bits. Modification will be |
| | Page(s): | 32 | effective after reset only |
| | | | 0 = two stop bits |
| | | | 1 = one stop bit |
| **25804** | **Modb:ExtendedOn** | | |
| | Level: | 4 | Modbus Extended Version |
| | Range: | 0..1 | Activation of Modbus Extended Version |
| | Page(s): | 32 | |

## 7.4 List 4: Curves and maps

| | | | |
|---|---|---|---|
| **29200** | **Modb:TxParamSet(x)** | | |
| **to** | Level: | 4 | Modbus Basic Version |
| **29299** | Range: | 0..29999 | Data field for read access |
| | Page(s): | 34 | Any parameters or measurement numbers must be |
| | | | entered whose values will be transmitted on request |
| **29200** | **Modb:ParamSet (x)** | | |
| **to** | Level: | 4 | Modbus Extended Version |
| **29299** | Range: | 0..29999 | Data field for read access and write access |
| | Page(s): | 34 | Any parameters or measurement numbers must be |
| | | | entered whose values will be transmitted on request or |
| | | | should be received |

# 8 Index of Figures

# 9 Index of Tables

# 10 Index