

Technical Dokumentation / Instruction Manual

romutec[®] 19"-Manual Override Modules with MODBus RTU Interface

Frontpanel-mounted System *BUS-tec Mod RTU*



romutec[®]
Steuer- u. Regelsysteme GmbH
Jochsberger Straße 39
D-91592 Buch am Wald
Telefon: +49 (0) 98 67/ 97 90-0
Telefax: +49 (0) 98 67/ 97 90-90
E-Mail: info@romutec.de
Home: www.romutec.de

Inhaltsverzeichnis

Romutec Manual Override Modules with MODBus RTU interface	3
General.....	3
Signaling.....	3
MODBus Adjustments	3
Monitoring / Control of MODBus Activity.....	3
Configuration	4
Hardware	4
Connection to the MODBus.....	5
Default Setting	5
Configuring the BZK1000MOD using the pushbuttons	5
Readout the type of interface setup.....	6
Addressing when reading and writing data via MODBus.....	6
Addressing binary In-/Outputs using MODBus WORD function codes	6
Module BLM1001	7
Module BLM1000	8
Module BDH1401	9
Module BDH1400	11
Module BDH2201	13
Module BDH2200	15
Module BDH4800	17
Module BAH4000.....	19
Module BZK1000MOD	22

Romutec Manual Override Modules with MODBus RTU-Interface

General

The System can be connected to controllers providing a MODBus interface, optionally RS232 or RS485. The protocol used is MODBus RTU. The interface type RS232 is applicable only for short distances. Several interface devices with manual override modules connected can be operated at one serial bus by giving the stations different addresses.

The terminal connections of the interface module BZK is as follows:

RS485	Rx-Tx /Rx-/Tx GND	Terminal: 5 4 3
RS232	TxD RxD GND	Terminal: 5 (Transmitted data) 4 (Received data) 3

Signaling

LED I ² C	Off Short (ca. 50 msec) Long (ca. 300 msec)	no exchange of data exchange of data via I ² C-Bus device not present or illegal parameters in the message
LED MOD-BUS	Blinking , slowly Short (ca. 100 msec)	ready for operation active during exchange of data, when module is addressed

In the configuration menu the LEDs will signalize the parameter's values that have to be adjusted.

MODBus Adjustments

Data Bits:	8
Stop Bits:	1
Parity:	none
Handshake:	off

Monitoring/Control of MODBus Activity

The monitoring/control of bus-activity will reset all outputs of the manual override modules if no valid message is received within the time configured. Whenever a valid MODBus message is received by the BZK module, the timer of the monitoring routine is started again. Configuring this time is done in the registers 42 and 43:

	Modbus Type	Register address	Range of values
Monitoring of MODBus activity	Word	42 43*)	1...250 seconds other values will deactivate monitoring

*) To put an intended change of value into effect, register 43 must contain the binary complement of the value in register 42, e.g.:

Time = 160 sec. → Word 42 = 00 A0 h
Word 43 = FF 5F h

Configuration

- **Hardware**

Maximum number of expansion modules per interface module

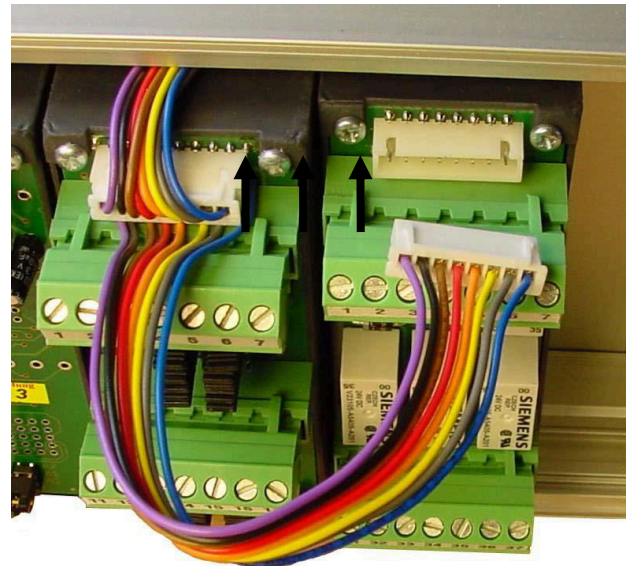
There can be connected up to 9 expansion modules to one MODBus RTU interface module BZK1000MOD. However, the maximum number of 8 digital respectively 8 analogue modules at the same time must not be exceeded. Apart from that, any combination of expansion modules is possible, there are no constraints concerning the sequential arrangement. Solely the rules of addressing (see below) have to be observed.

Mounting and Bus connection

For mounting the modules, racks of different sizes are available. They provide space for 5 up to 12 devices. The bus connections between expansion modules among each other and the BZK1000MOD (I²C-/internal bus) are realized pluggable. They are done with the 8-pole cable which is a fixed part of every expansion module (see picture on the right).

Caution: Plugging or loosening the 8-pole bus while modules are running can cause damage to all connected modules!

Furthermore, via these bus connections the power supply of the digital expansion modules is realized. In contrast, for power supply of analogue modules the connector blocks have to be used.



Rules of Addressing

The adjustment of the address that is used by the interface module communicating to the expansion modules is done by the use of jumpers. With 3 jumpers per module, a range of 0...7 for the binary coded addresses can be represented, as the following table demonstrates:

Jumper (0=contact open, 1=connected)			Address value
J1	J2	J3	
0	0	0	0
1	0	0	1
0	1	0	2
1	1	0	3
0	0	1	4
1	0	1	5
0	1	1	6
1	1	1	7

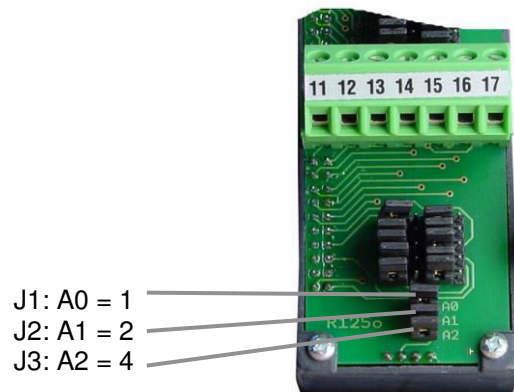


Fig.: The showed address is 7 (=1+2+4)

The topmost jumper J1 represents the least significant bit (1), the lowermost one represents the most significant bit (4) of the address, as figure 2 shows.

Following has to be considered when coding the addresses of the expansion modules:

- Generally a double assignment of addresses is not permissible. Each address may be assigned only once per interface module.
- Exception: Respectively one digital and one analogue module connected to a BZK1000MOD can have seemingly the same address, because they differ in a fixed internal address bit which is hidden to the user.

- **Connection to the MODBus**

Default Setting

Each BZK1000MOD will be delivered having a factory-provided default setting (parameters see below). Normally, with this setup the device can be put into service instantly. However, the parameters of the default setting can be changed at any time. This could be required, e.g., when more than one interface module BZK1000MOD shall run at one controller. In this case, the station address of one interface module has to be changed. There are two options for modifying the default settings, which is to change the parameters directly using the push buttons of the BZK on the one hand, or, on the other hand, using an already working Modbus communication.

Configuring the BZK1000MOD using the push-buttons

When switching on the power supply, the push-button „Reset“ („Störung Entsperren“, [3]) has to be applied for at least 2 seconds to enter the configuration menu. This will be indicated by the upper LED (I²C) blinking fast (approx. 10 Hz).

When in the config menu, pressing the push-button „Reset“ [3] will cause a jump to the next parameter. Thereby the frequency of the blinking upper LED will decrease with each step to the next point. If the last point of the menu is reached the LED will be illuminated continuously. After the last point in the config menu there will be a jump back to the first one, when pushing the button [3] again.

Meanwhile the lower LED (Modbus-Status) indicates the values of the according parameters by the means of series of pulses; e.g. three pulses means that the parameter's value is 3.

Pressing the push-button „Hupe quittieren“ [4] will increase the value step by step until the required adjustment is reached.

When in the config menu, pressing the push-button at the rear side of the module (beside the green LED) will recall the default settings. These are:

Station's number	160
Mode	Modbus RTU
Type of interface	RS 485
Rate of transmission	9600 Baud

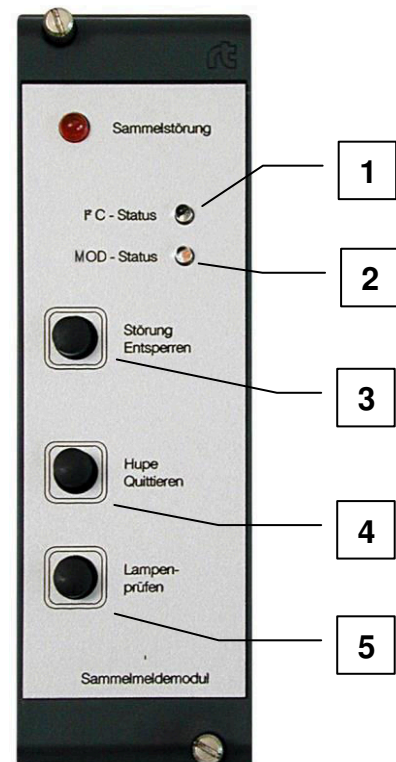


Fig.: The adjacent described control elements

Menu sequence, indicated by the LED I²C-Status

10 Hz	„ones' place“ of the station's number		
3 Hz	„tens' place“ of the station's number		
1 Hz	„hundreds' place“ of the station's number		
0,5 Hz	„Mode“	7 = Modbus RTU	
0,2 Hz	„Type of interface“	2 = RS232, 2-wire	4 = RS485
ON	„Rate of Transmission“	1 = 300 Baud	5 = 4800 Baud
		2 = 600 Baud	6 = 9600 Baud
		3 = 1200 Baud	7 = 19200 Baud
		4 = 2400 Baud	8 = 38400 Baud

Pressing the push-button „Lampen prüfen“ [5] will save the new settings and config menu is left.

Readout the type of interface setup

The **green LED** at the rear side of the interface module BZK1000MOD shows by the means of two blink frequencies whether the MODBus interface is configured as RS232 or as RS485 type (during normal operating mode, push-button not pressed):

RS485	LED briefly illuminated, longer break
RS232	LED longer illuminated, shorter break

Button for recalling
default settings

LED signaling the
type of interface

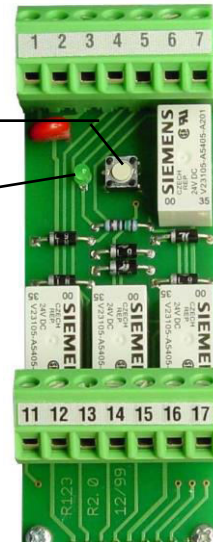


Fig.: LED for indicating the type of MODBus interface setup and push-button for recalling the default settings

Addressing when reading and writing data via MODBus

The MODBus-communication to the Romutec modules is done with standard MODBus message types. Access to data points (inputs or outputs) of the expansion modules has to be carried out for a complete module at a time.

The binary type modules consist of up to 4 groups of 8 In- or Outputs per group (4 x 8 Bit). According to the type of module, they will be addressed with the MODBus function codes **Read Discrete Inputs (0x02)** and **Write Multiple Coils (0x0F)**. Please note that all groups of one module have to be written or read at once. Partial handling (e.g. reading only group 1 out of 3) is not supported.

The analog type modules consist of one group of binary inputs (representing the switches' states Auto and Manual override) and one group of analog Inputs and Outputs as well, each consisting of 4 analog values. These groups will be addressed with the MODBus function codes **Read Discrete Inputs (0x02)**, **Read Input Registers (0x04)** and **Force Multiple Registers (0x10)**.

The starting address for write and read access to the expansion modules is calculated using their jumper coded hardware address. Please observe the following rules concerning the number of elements which are addressed and the starting address. Partial handling of inputs or outputs of an expansion module and invalid starting addresses will be rejected with an error response message.

Addressing binary In-/Outputs using MODBus WORD function codes

Since BZK1000MOD firmware version 3.0 all binary inputs and outputs can be handled using the MODBus function codes **Read Input Registers (0x04)** and **Force Multiple Registers (0x10)** respectively. Even if using these function codes, registers of an expansion module have to be handled **all at once**.

Caution: Read access to registers which are provided for write access only might cause malfunctions like vibrating relays.

Starting addresses for read and write access to the expansion modules are calculated similar to those using the Bit function codes, i.e. by using their jumper coded hardware address. For detailed information see the following examples and the MODBus address tables („Crosslist“) in the attachment.

Module BLM1001

- **Status request of LEDs 1-10**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0 ... 3) of Module address „0“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	00 h	00 h	00 h	20 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
02 h = MODBus function code READ DISCRETE INPUTS
 00 00 h = Starting address 0 (Module address 0 x 32 + 0 = 0)
00 20 h = Number of Bits = 32

Response Message:

Station address	MODBus-function code	Byte Count	Data (Byte 1)	Data (Byte 2)	Data (Byte 3)	Data (Byte 4)	CRC	CRC
A0 h	02 h	04 h	3E h	3D h	15 h	3C h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
02 h = MODBus function code READ DISCRETE INPUTS
04 h = Number of received data bytes = 4
 3E h = Data byte 1 (xx**11 1110**) → LED 1 green = ON (Bits 6 + 7 = xx will be ignored)
 3D h = Data byte 2 (xx**11 1101**) → LED 4 red = ON (Bits 6 + 7 = xx will be ignored)
 15 h = Data byte 3 (xx**01 0101**) → LED 7 - 9 red = ON (Bits 6 + 7 = xx will be ignored)
 3C h = Data byte 4 (xxxx xx**00**) → LED 10 red+green = ON (Bits 2...7 = xxxx xx will be ignored)

For detailed information about which bits are affected by an activated LED and its colour, please see in the attachment (MODBus Crosslist).

Note: When using a BLM1001, activated LEDs will turn the according bit to „0“, whereas a bit value being „1“ means the LED is not activated!

Important: All the 4 data bytes have to be polled at once. Attempting to readout single bytes (e.g. solely data byte #2) will fail and causes an exception response message.

Values printed in bold letters are fixed and must not be changed.

Module BLM1000

- **Activation of LEDs 1-10**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating LEDs (=outputs, group 0 ... 3) von of Module address „1“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte							
A0 h	0F h	00 h	20 h	00 h	20 h	04 h	01 h	02 h	2A h	03 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 20 h = Starting address 32 (Module address 1 x 32 + 0 = 32)
- 00 20 h** = Number of Bits = 32
- 04 h** = Number of data bytes to transmit = 4
- 01 h = Data byte 1 (xx**00 0001**) → LED 1 green = ON (Bits 6 + 7 = xx will be ignored)
- 02 h = Data byte 2 (xx**00 0010**) → LED 4 red = ON (Bits 6 + 7 = xx will be ignored)
- 2A h = Data byte 3 (xx**10 1010**) → LED 7 - 9 red = ON (Bits 6 + 7 = xx will be ignored)
- 03 h = Data byte 4 (xxxx xx**11**) → LED 10 red+green = ON (Bits 2...7 = xxxx xx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	00 h	20 h	00 h	20 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 20 h = Starting address 32 (Module address 1 x 32 + 0 = 32)
- 00 20 h** = Number of Bits = 32

For detailed information about which LED will be activated in which colour by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BLM1000, LEDs will be activated setting the according bit to „0“, whereas a bit value being „1“ means the LED will be not activated!

Important: All the 4 data bytes have to be sent at once. Attempting to write single bytes (e.g. solely data byte #2) will fail and causes an exception response message.

Values printed in bold letters are fixed and must not be changed.

Module BDH1401

- **Status request of LEDs and switches' states (Auto – Manual override)**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0 ... 2) of Module address „2“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	00 h	40 h	00 h	18 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
02 h = MODBus function code READ DISCRETE INPUTS
 00 40 h = Starting address 64 (Module address 2 x 32 + 0 = 64)
00 18 h = Number of Bits = 24

Response Message:

Station address	MODBus-function code	Byte Count	Data (Byte 1)	Data (Byte 2)	Data (Byte 3)	CRC	CRC
A0 h	02 h	03 h	39 h	2A h	3D h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
02 h = MODBus function code READ DISCRETE INPUTS
03 h = Number of received data bytes = 3
 39 h = Data byte 1 (xxxx 1001) → Switches 1 + 4 = manual override (Bits 4...7 = xxxx will be ignored)
 2A h = Data byte 2 (xx**10 1010**) → LED 1, 3, 5 red = ON (Bits 6 + 7 = xx will be ignored)
 3D h = Data byte 3 (xxxx xx**01**) → LED 8 green = ON (Bits 2...7 = xxxx xx will be ignored)

For detailed information about which bits are affected by activated LEDs and switches, please see in the attachment (MODBus Crosslist).

Note: When using a BDH1401, activated LEDs will turn the according bit to „0“, whereas a bit value being „1“ means the LED is not activated!

A switch being in the position 'Automatic' will set the according bit to „0“. When the bit value changes to „1“ it means that the binary output has been overridden (position ON or OFF).

Important: All the 3 data bytes have to be polled at once. Attempting to readout single bytes (e.g. solely data byte #2) will fail and causes an exception response message.

Values printed in bold letters are fixed and must not be changed.

Module BDH1401

- **Activation of Relays**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating relays (=outputs, group 0) of Module address „2“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte				
A0 h	0F h	00 h	40 h	00 h	08 h	01 h	07 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 40 h = Starting address 64 (Module address 2 x 32 + 0 = 64)
- 00 08 h** = Number of Bits = 8
- 01 h** = Number of data bytes to transmit = 1
- 07 h = Data byte (xxxx 0111) → Relays 1...3 = ON (Bits 4...7 = xxxx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	00 h	40 h	00 h	08 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 40 h = Starting address 64 (Module address 2 x 32 + 0 = 64)
- 00 08 h** = Number of Bits = 8

For detailed information about which relay will be activated by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BDH1401, relays will be activated setting the according bit to „1“, whereas a bit value being „0“ means the relay will be not activated!

Values printed in bold letters are fixed and must not be changed.

Module BDH1400

- **Status request of switches' states (Auto – Manual override)**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0) of Module address „3“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	00 h	60 h	00 h	08 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

00 60 h = Starting address 96 (Module address 3 x 32 + 0 = 96)

00 08 h = Number of Bits = 8

Response Message:

Station address	MODBus-function code	Byte Count	Data Byte	CRC	CRC
A0 h	02 h	01 h	06 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 h = Number of received data bytes = 1

06 h = Data byte (xxxx 0110) → Switches 2 + 3 = manual override (Bits 4...7 = xxxx will be ignored)

For detailed information about which bits are affected by the switches, please see in the attachment (MODBus Crosslist).

Note: When using a BDH1400, a switch being in the position 'Automatic' will set the according bit to „0“. When the bit value changes to „1“ it means that the binary output has been overridden (position ON or OFF).

Values printed in bold letters are fixed and must not be changed.

Module BDH1400

- **Activation of relays and LEDs**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating relays and LEDs (=outputs, group 0...2) of Module address „3“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte 1	Data Byte 2	Data Byte 3	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte						
A0 h	0F h	00 h	60 h	00 h	18 h	03 h	0E h	20 h	06 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
0F h = MODBus function code WRITE MULTIPLE COILS
 00 60 h = Starting address 96 (Module address 3 x 32 + 0 = 96)
00 18 h = Number of Bits = 24
03 h = Number of data bytes to transmit = 3
 0E h = Data byte 1 (xx**00 1110**) → LEDs 1+2 green, 3 red (Bits 6 + 7 = xx will be ignored)
 20 h = Data byte 2 (xx**10 0000**) → LED 8 green (Bits 6 + 7 = xx will be ignored)
 06 h = Data byte 3 (xxxx **0110**) → Relays 2 + 3 = ON (Bits 4...7 = xxxx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	00 h	60 h	00 h	18 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
0F h = MODBus function code WRITE MULTIPLE COILS
 00 60 h = Starting address 96 (Module address 3 x 32 + 0 = 96)
00 18 h = Number of Bits = 24

For detailed information about which relay will be activated by which bit respectively which LED will be activated in which colour by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BDH1400, relays and LEDs will be activated setting the according bit to „1“, whereas a bit value being „0“ means the relay respectively LED will be not activated!

Values printed in bold letters are fixed and must not be changed.

Module BDH2201

- **Status request of LEDs and switches' states (Auto – Manual override)**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0 ... 2) of Module address „4“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	00 h	80 h	00 h	18 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
02 h = MODBus function code READ DISCRETE INPUTS
 00 80 h = Starting address 128 (Module address 4 x 32 + 0 = 128)
00 18 h = Number of Bits = 24

Response Message:

Station address	MODBus-function code	Byte Count	Data (Byte 1)	Data (Byte 2)	Data (Byte 3)	CRC	CRC
A0 h	02 h	03 h	31 h	1E h	3F h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
02 h = MODBus function code READ DISCRETE INPUTS
03 h = Number of received data bytes = 3
 31 h = Data byte 1 (xxxx xx01) → Switch 1 = manual override (Bits 2...7 = xxxx xx will be ignored)
 1E h = Data byte 2 (xx**01 1110**) → LED 1 + 6 = ON (Bits 6 + 7 = xx will be ignored)
 3F h = Data byte 3 (xxxx xxxx) → (Data byte 3 will be ignored, however it has to be polled within the request)

For detailed information about which bits are affected by activated LEDs and switches, please see in the attachment (MODBus Crosslist).

Note: When using a BDH2201, activated LEDs will turn the according bit to „0“, whereas a bit value being „1“ means the LED is not activated!

A switch being in the position 'Automatic' will set the according bit to „0“. When the bit value changes to „1“ it means that the binary output has been overridden (position ON or OFF).

Important: All the 3 data bytes have to be polled at once. Attempting to readout single bytes (e.g. solely data byte #2) will fail and causes an exception response message.

Values printed in bold letters are fixed and must not be changed.

Module BDH2201

- **Activation of Relays**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating relays (=outputs, group 0) of Module address „4“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte				
A0 h	0F h	00 h	80 h	00 h	08 h	01 h	0A h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 80 h = Starting address 128 (Module address 4 x 32 + 0 = 128)
- 00 08 h** = Number of Bits = 8
- 01 h** = Number of data bytes to transmit = 1
- 0A h = Data byte (xxxx 1010) → Relays 2 + 4 = ON (Bits 4...7 = xxxx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	00 h	80 h	00 h	08 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 80 h = Starting address 128 (Module address 4 x 32 + 0 = 128)
- 00 08 h** = Number of Bits = 8

For detailed information about which relay will be activated by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BDH2201, relays will be activated setting the according bit to „1“, whereas a bit value being „0“ means the relay will be not activated!

Values printed in bold letters are fixed and must not be changed.

Module BDH2200

- **Status request of switches' states (Auto – Manual override)**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0) of Module address „6“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	00 h	C0 h	00 h	08 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

00 C0 h = Starting address 192 (Module address 6 x 32 + 0 = 192)

00 08 h = Number of Bits = 8

Response Message:

Station address	MODBus-function code	Byte Count	Data Byte	CRC	CRC
A0 h	02 h	01 h	03 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 h = Number of received data bytes = 1

03 h = Data byte (xxxx xx11) → both switches 1 + 2 = manual override (Bits 2...7 = xxxx xx will be ignored)

For detailed information about which bits are affected by the switches, please see in the attachment (MODBus Crosslist).

Note: When using a BDH2200, a switch being in the position 'Automatic' will set the according bit to „0“. When the bit value changes to „1“ it means that the binary output has been overridden (position ON or OFF).

Values printed in bold letters are fixed and must not be changed.

Module BDH2200

- **Activation of relays and LEDs**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating relays and LEDs (=outputs, group 0...2) of Module address „6“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte 1	Data Byte 2	Data Byte 3	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte						
A0 h	0F h	00 h	C0 h	00 h	18 h	03 h	01 h	03 h	06 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 C0 h = Starting address 192 (Module address 6 x 32 + 0 = 192)
- 00 18 h** = Number of Bits = 24
- 03 h** = Number of data bytes to transmit = 3
- 01 h = Data byte 1 (xx**00 0001**) → LED 1 red (Bits 6 + 7 = xx will be ignored)
- 03 h = Data byte 2 (xxxx xx**11**) → LEDs 5 + 6 green (Bits 2...7 = xx will be ignored)
- 06 h = Data byte 3 (xxxx 0110) → Relays 2 + 3 = ON (Bits 4...7 = xxxx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	00 h	C0 h	00 h	18 h	XX	XX

Data used in the example:

- A0 h = BZK1000MOD Station address 160
- 0F h** = MODBus function code WRITE MULTIPLE COILS
- 00 C0 h = Starting address 192 (Module address 6 x 32 + 0 = 192)
- 00 18 h** = Number of Bits = 24

For detailed information about which relay will be activated by which bit respectively which LED will be activated in which colour by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BDH2200, relays and will be activated setting the according bit to „1“, whereas a bit value being „0“ means the relay respectively LED will be not activated!

Values printed in bold letters are fixed and must not be changed.

Module BDH4800

- **Status request of push button switches' states**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0) of Module address „5“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	00 h	A0 h	00 h	08 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

00 A0 h = Starting address 160 (Module address 5 x 32 + 0 = 160)

00 08 h = Number of Bits = 8

Response Message:

Station address	MODBus-function code	Byte Count	Data Byte	CRC	CRC
A0 h	02 h	01 h	04 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 h = Number of received data bytes = 1

04 h = Data byte (xxxx 0100) → Push-button switch 3 = pressed (Bits 4...7 = xxxx will be ignored)

For detailed information about which bits are affected by the push-button switches, please see in the attachment (MODBus Crosslist).

Note: When using a BDH4800, a push-button switch being not activated will set the according bit to „0“. When the bit value changes to „1“ it means that the push button currently is being pressed down.

Values printed in bold letters are fixed and must not be changed.

Module BDH4800

- **Activation of relays and LEDs**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating relays and LEDs (=outputs, group 0...2) of Module address „5“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte 1	Data Byte 2	Data Byte 3	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte						
A0 h	0F h	00 h	A0 h	00 h	18 h	03 h	02 h	11 h	00 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
0F h = MODBus function code WRITE MULTIPLE COILS
 00 A0 h = Starting address 160 (Module address 5 x 32 + 0 = 160)
00 18 h = Number of Bits = 24
03 h = Number of data bytes to transmit = 3
 02 h = Data byte 1 (xx00 0010) → LED 1 (!) = ON (Bits 6 + 7 = xx will be ignored)
 11 h = Data byte 2 (xx01 0001) → LED 7 + 10 = ON (Bits 6 + 7 = xx will be ignored)
 00 h = Data byte 3 (xxxx 0000) → all relays = OFF (Bits 4...7 = xxxx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	00 h	A0 h	00 h	18 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160
0F h = MODBus function code WRITE MULTIPLE COILS
 00 A0 h = Starting address 160 (Module address 5 x 32 + 0 = 160)
00 18 h = Number of Bits = 24

For detailed information about which relay respectively which LED will be activated by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BDH4800, relays and LEDs will be activated setting the according bit to „1“, whereas a bit value being „0“ means the relay respectively LED will be not activated!

Values printed in bold letters are fixed and must not be changed.

Module BAH4000

- **Status request of switches' states (Auto – Manual override)**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs (group 0) of Module address „3“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	01 h	18 h	00 h	08 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 18 h = Starting address 280 (Module address 3 x 8 + 256 = 280)

00 08 h = Number of Bits = 8

Response Message:

Station address	MODBus-function code	Byte Count	Data Byte	CRC	CRC
A0 h	02 h	01 h	04 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 h = Number of received data bytes = 1

04 h = Data byte (xxxx 0100) → switch 3 = manual override (Bits 4...7 = xxxx will be ignored)

For detailed information about which bits are affected by the switches, please see in the attachment (MODBus Crosslist).

Note: When using a BAH4000, a switch being in the position 'Automatic' will set the according bit to „0“. When the bit value changes to „1“ it means that the analogue output has been overridden (position 0%, 100% or Poti).

Values printed in bold letters are fixed and must not be changed.

Module BAH4000

- **Status request of Analogue Inputs (Checkback signal of Analogue Outputs)**

MODBus function code: MODBus [04] → READ INPUT REGISTERS

Example: Reading Analogue Inputs 1...4 of Module address „3“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Words (High Byte)	Number of Words (Low Byte)	CRC	CRC
A0 h	04 h	00 h	0C h	00 h	04 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

04 h = MODBus function code READ INPUT REGISTERS

00 0C h = Starting address 12 (Module address 3 x 4 + 0 = 12)

00 04 h = Number of Words to be received = 4

Response Message:

Station address	MODBus-function code	Byte Count	Word AI 1		Word AI 2		Word AI 3		Word AI 4		CRC	CRC
			High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte		
A0 h	04 h	08 h	00 h	FF h	00 h	7F h	00 h	3F h	00 h	00 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

04 h = MODBus function code READ INPUT REGISTERS

08 h = Number of received data bytes = 8

FF h = AI 1 (FF h ≡ 255) → 100% ≈ 10,0 Volts (High Byte = 00 is unused)

7F h = AI 2 (7F h ≡ 127) → 50% ≈ 5,0 Volts (High Byte = 00 is unused)

3F h = AI 3 (3F h ≡ 63) → 25% ≈ 2,5 Volts (High Byte = 00 is unused)

00 h = AI 4 (00 h ≡ 0) → 0% ≈ 0,0 Volts (High Byte = 00 is unused)

Values printed in bold letters are fixed and must not be changed.

Module BAH4000

- Controlling the Analogue Output Signal

MODBus function code: MODBus [16] (= Hex 10) → WRITE MULTIPLE REGISTERS

Example: Controlling Analogue Output Signals 1...4 of Module address „3“ at Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Words		Byte Count	Word AO 1		Word AO 2		Word AO 3		Word AO 4		CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte		High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte		
A0 h	10 h	00 h	0C h	00 h	04 h	08 h	00 h	FF h	00 h	7F h	00 h	3F h	00 h	00 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

10 h = MODBus function code WRITE MULTIPLE REGISTERS

00 0C h = Starting address 12 (Module address 3 x 4 + 0 = 12)

00 04 h = Number of Words to be transmitted = 4

08 h = Number of data bytes to be transmitted = 8

FFh = AO 1 (FF h ≡ 255) → 100% ≈ 10,0 Volts (High Byte = 00 is unused)

7Fh = AO 2 (7F h ≡ 127) → 50% ≈ 5,0 Volts (High Byte = 00 is unused)

3Fh = AO 3 (3F h ≡ 63) → 25% ≈ 2,5 Volts (High Byte = 00 is unused)

00 h = AO 4 (00 h ≡ 0) → 0% ≈ 0,0 Volts (High Byte = 00 is unused)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Words (High Byte)	Number of Words (Low Byte)	CRC	CRC
A0 h	10 h	00 h	0C h	00 h	04 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

10 h = MODBus function code WRITE MULTIPLE REGISTERS

00 0C h = Starting address 12 (Module address 3 x 4 + 0 = 12)

00 04 h = Number of Words that have been received = 4

Values printed in bold letters are fixed and must not be changed.

Module BZK1000MOD

Gateway module between the controller's MODBus interface and the expansion modules, providing the following additional functions:

- **Lamp testing**
- **Collective fault indication**
- **Relay contact for external horn**
- **Pushbutton for acknowledging faults**
- **Pushbutton and relay contact for faults to be cleared externally**

- **Status request of push button switches' states**

MODBus function code: MODBus [02] → READ DISCRETE INPUTS

Example: Reading Inputs of BZK1000MOD with Station address 160

Query Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	02 h	01 h	40 h	00 h	03 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 40 h = Starting address 320

00 03 h = Number of Bits = 3

Response Message:

Station address	MODBus-function code	Byte Count	Data Byte	CRC	CRC
A0 h	02 h	01 h	04 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

02 h = MODBus function code READ DISCRETE INPUTS

01 h = Number of received data bytes = 1

04 h = Data byte (xxxx x100) → Push-button 'Lamp testing' = pressed (Bits 3...7 = xxxx x will be ignored)

For detailed information about which bits are affected by the push-button switches, please see in the attachment (MODBus Crosslist).

Note: When using a BZK1000MOD, a push-button switch being not activated will set the according bit to „0“. When the bit value changes to „1“ it means that the push button currently is being pressed down.

Values printed in bold letters are fixed and must not be changed.

Module BZK1000MOD

- **Activation of Relays**

MODBus function code: MODBus [0F] → WRITE MULTIPLE COILS

Example: Activating relays of BZK1000MOD with Station address 160

Query Message:

Station address	MODBus function code	Starting address		Number of Bits		Byte Count	Data Byte	CRC	CRC
		High Byte	Low Byte	High Byte	Low Byte				
A0 h	0F h	01 h	48 h	00 h	04 h	01 h	08 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

0F h = MODBus function code WRITE MULTIPLE COILS

01 48 h = Starting address 328

00 04 h = Number of Bits = 4

01 h = Number of data bytes to transmit = 1

08 h = Data byte (xxxx 1000) → Relay 'Lamp testing' = ON (Bits 4...7 = xxxx will be ignored)

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Bits (High Byte)	Number of Bits (Low Byte)	CRC	CRC
A0 h	0F h	01 h	48 h	00 h	04 h	XX	XX

Data used in the example:

A0 h = BZK1000MOD Station address 160

0F h = MODBus function code WRITE MULTIPLE COILS

01 48 h = Starting address 328

00 04 h = Number of Bits = 4

For detailed information about which relay will be activated by which bit, please see in the attachment (MODBus Crosslist).

Note: When using a BZK1000MOD, relays will be activated setting the according bit to „1“, whereas a bit value being „0“ means the relay will be not activated!

Values printed in bold letters are fixed and must not be changed.

Module BZK1000MOD

- Configuring the parameters of the MODBus interface using MODBus commands

MODBus function code: MODBus [16] (= Hex 10) → WRITE MULTIPLE REGISTERS

The alterable parameters are listed below:

	Modbus Type	Register Address	Range of Values
Station Address	Word	32 33 ^{*)}	0...249 Default: [160]
Rate of transmission (Baud)	Word	34 35 ^{*)}	1...8 corresponding to 300/600/1200/2400/4800 9600/19200/38400 Baud Default: [6] = 9600 Baud
Modus	Word	36 37 ^{*)}	7 = MODBus RTU Default: [7] = MODBus RTU
Type of interface	Word	38 39 ^{*)}	2 = RS232 4 = RS485 Default: [4] = RS485

^{*)} In order to execute a change of values as intended, the second word of the sent data (marked with a *) has to be the complement of the first word, e.g.:

Bus address = 160 → Word 32 = 00 A0 h
 Word 33 = FF 5F h

Example: Sending a new setup (Station = 150, Rate of Transmission = 9600 Baud, Modus = MODBus RTU, Type of Interface = RS485) to the module having the current station address 160.

Important: The four parameters have to be sent all together as contiguous data with just one single command, even if only one parameter has to be changed!

Query Message:

Station address	MODBus function code	Starting address		Number of Words		Byte Count
		High Byte	Low Byte	High Byte	Low Byte	
A0 h	10 h	00 h	20 h	00 h	08 h	10 h

word 1		NOT word 1		word 2		NOT word 2		word 3		NOT word 3		word 4		NOT word 4		CRC	CRC
High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte		
00 h	96 h	FF h	69 h	00 h	06 h	FF h	F9 h	00 h	07 h	FF h	F8 h	00 h	04 h	FF h	FB h	XX	XX

Response Message:

Station address	MODBus function code	Starting address (High Byte)	Starting address (Low Byte)	Number of Words (High Byte)	Number of Words (Low Byte)	CRC	CRC
A0 h	10 h	00 h	20 h	00 h	08 h	XX	XX