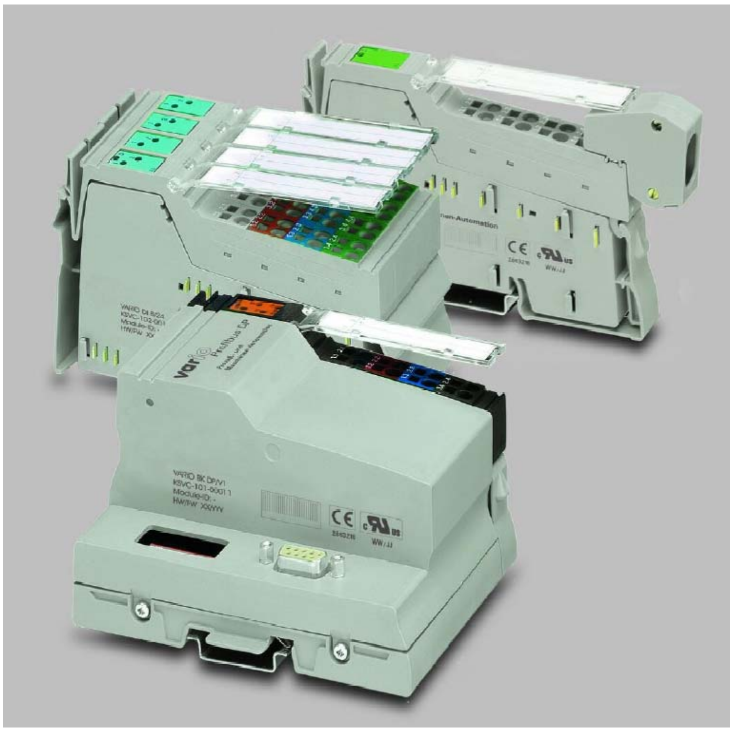
CD Automation SRL

MODBUS Buscoupler



Interface description

Modbus 9499 040 69611

07/2004

Content

1 General 5
2 Interface connection, indicator LED signification 6
2.1 RS 485/422 interface version. 6
2.2 Signification of indicator LEDs 6
3 General information on the RTU Modbus protocol. 7
3.1 General message structure. 7
3.2 Baudrates (bAud) 7
3.3 Parity (PrtY) 7
3.4 CRC 8
3.5 End identier 8
3.6 Delayed reply (dELY) 8
4 Modbus function format 9
4.1 Modbus addresses 9
4.2 Function codes. 13
4.2.1 Read process data data, parameter or configuration data 13
4.2.2 Write a single datum (process data, parameter or configuration). 14
4.3 Diagnosis 15
4.3.1 Returning a received message. 16
4.3.2 Communication restart (terminates the listen-only mode) 16
4.3.3 Returning the diagnosis register. 16
4.3.4 Changing to listen-only mode 16
4.3.5 Deleting the counters and diagnosis register 16
4.3.6 Returning the message counter 16
4.3.7 Returning the counter of faulty message transmissions 16
4.3.8 Returning the counter of the messages replied with error message 17
4.3.9 Returning the counter of messages for this slave 17
4.3.10 Returning the counter of non-replied messages. 17
4.3.11 Returning the counter of messages replied with NAK 17
4.3.12 Returning the counter of messages replied with busy 17
4.3.13 Returning the counter with parity error 17
4.3.14 Returning the counter with framing error 17
4.3.15 eturning the counter of too long messages 18
4.4 Writing several process data data, parameter and configuration data 18
4.5 read-out and specification of data in float format 19
5 Error report 20
6 Hints on operation. 21
6.1 Connecting the interface 21
6.1.1 Cable installation. 21
6.2 System construction. 21
6.2.1 **Minimum equipment of a MODBUS system** 21
6.2.2 **Maximum equipment of a Modbus system** 21
6.2.3 Cable run inside buildings 22

1. General

Various fieldbus interfaces can be connected to Modular Controller System CD VARIO. For this, the relevant bus coupler is used as a central station for the controller system.
One of these bus couplers serves to support the Modbus protocol via a front-panel RS485/422 interface, which permits transmission of all process, parameter and configuration data.
The serial communication interface can be used for communication with supervisory systems, visualization tools, etc.
Another interface, which is always provided as standard, is on the CD VARIO controller modules. This full RS232 interface is used for connection of the ‘BlueControl’ tool, which runs on a PC.
This interface also uses the Modbus protocol. Communication is according to the master/slave principle.

CD VARIO is always slave.

Characteristic data of the cable medium and the physical and electrical interface properties are :

• **Network topology**
Linear bus with bus terminating resistors at both ends.

• **Transfer medium**
Screened, twisted 2 / 5-wire copper cable

• **Baudrates and cable lengths (without repeater)**
The maximum cable length of 1000m must not be exceeded.
The following Baudrates are supported:
38400 Baud
19200 Baud
9600 Baud
4800 Baud
2400 Baud

• **Interface**
RS485 /RS422 with AMP flat-pin connectors; can be mounted at the site

• **Addressing: 0 ... 247**
Address setting via engineering tool

• **32 instruments in a segment. Can be extended to 247 by means of repeaters.**

Interface connection, indicator LED signification

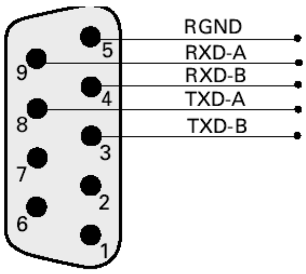
2.1

Both the RS422 and the RS485 interface of the electrical interface can be used. The KS VARIO system Modbus coupler detects the connected version automatically.

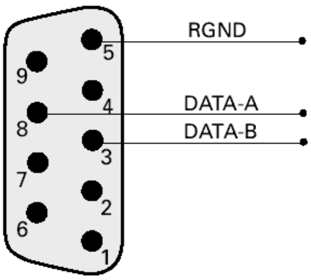
The connection has to be done via a 9-pole Sub-D-female connector:

1. N.C.
2. GND
3. TxD-B (DATA-B)
4. RxD-B
5. Via 100 _ to GND
6. N.C.
7. GND
8. TxD-A (DATA-A)
9. RxD-A

❑ Communication via RS422

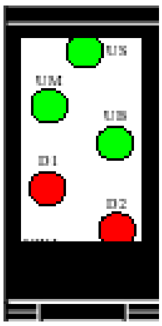


❑ Communication via RS485



2.2 Signification of indicator LEDs

LED no.	LED colour	Function
US	green	Segment voltage U provided
UM	green	Module voltage U provided
UB	green	Coupler voltage U provided
D1	red	Control by signal TxD of KS vario (0 = LED on)
D2	red	Control by signal RxD of KS vario (0 = LED on)



. 3 General information on the RTU Modbus protocol

The MODBUS protocol was defined for communication between a supervisory system and the Modicon control system.
ASCII and RTU protocols were defined. Instrument KS VARIO supports the RTU protocol.
The structure for transfer of a byte in the RTU protocol is:

Start bit	8 data bits	Parity/stop bit	Stop bit

Even or odd parity bit can be selected. Unless a parity bit is selected, an additional stop bit is transferred.

3.1 General message structure

The message is read in into a data buffer with a max. length of 250 bytes. Longer messages are not accepted. The instrument does not reply.
The message is composed of the following elements:

Instrument address	Function code	Data	CRC	End identifier

• Instrument address (Addr)
The instrument address specifies the instrument. Instrument addresses within 1 - 247 can be defined.
Instrument address 0 is used as a broadcast message. A broadcast message can be defined for write orders, which are handled by all instruments on the bus. As all instruments handle the order, no reply by the instruments is given.
• Function code
The function code defines the type of a message. There are 17 defined messages. Which messages are supported is described in chapter ”Data and function control”.
• Data
The data block comprises the further specification of the action defined with the function code. The data block length is dependent of function code. For further information, see chapter ”MODBUS function format” (chapter 4). The internal data buffer includes 256 bytes. I.e., max. 120 integer or 60 real data of a message can be written or read out.
• CRC
The CRC code ensures that transmission errors can be detected. For further information, see chapter ”CRC”.
• End identifier
The end of a message is defined by a time of 3,5 characters without data transfer. For further information, see chapter ”End identifier”.

3.2 Baudrates (bAud

The following Baudrates are supported:
38400 Baud
19200 Baud
9600 Baud
4800 Baud
2400 Baud

3.3 Parity (PrtY)

Even, odd or no parity can be selected.
The parity bit can be used for checking, if there was a single error within a byte during transmission. With even parity, the parity bit is set so that the sum of set bits in the 8-data bits and the parity bit is an even number. This is applicable analogously to odd parity. When detecting a parity error during reception, no reply message is generated.
Unless a parity is selected, 1 or 2 stop bits can be output (determination via configuration).

3.4 CRC

The CRC is a 16-bit value which is appended to the message. This value is used to determine, if the transmission of a
message was detected correctly. In conjunction with parity checking, all possible transmission errors should be
detected.

When detecting a parity error during transmission, no reply message is generated.
The algorithm for generating the CRC is:

- **Load the CRC register with FFFF**
- **Exclusive OR function of the send/receive bytes with the high portion of the CRC register**
- **Shift the CRC register right by 1 bit**
- **If the shifted bit was 1, connect the CRC register with value A001 by an exclusive OR function.**
- **Repeat steps 3 and 4 for the other 7 data bits.**
- **Repeat steps 2 to 5 for all other send/receive bytes.**
- **Append the result of the CRC register to the message, starting with the high portion. When checking a receive message, the result in the CRC register is 0, if the message is handled inclusive of the CRC.**

3.5 End identier

The end identifier of a message is specified as rest situation on the Modbus with a length of 3,5 characters. This time
must elapse, before a slave may start its reply, or before a master can send another message.
The evaluation of a message may start, when the rest condition on the Modbus was met during more than 1,5
characters. However, a reply is started only after 3,5 characters.

3.6 Delayed reply (dELY)

Some instruments have a delay when switching over between send and receive mode. This delay can be adjusted in
ms. The adjusted delay is in addition to the 3,5-character waiting time after the end of a message, before generating a
reply.

4 Modbus function format

The signification of the data range is different dependent of function code. The Modbus protocol defines 17 different functions.
To permit reading and writing of process data, parameters and configuration data with a minimum number of accesses, the relevant ranges are grouped together, whereby process data can be defined several times in different groups.

Example for a transmission

Inquiry

Field name	Value (hex)	Signification
Address	11	Address 17
Function	04	Read parameter/configuration
Start address high Start address low	03 EC	Start address 1004
Number of values	00 03	Number of values 03
CRC	CRC byte1 CRC byte2	

Reply:

Field name	Value	Signification
Address	11	Address 17
Function	04	Read parameter/configuration
Number of bytes	06	6 data bytes are sent
Value1	04 2A	Value1 = 1066
Value2	00 8C	Value2 = 140
Value3	10 3E	Value3 = 4158
CRC	CRC byte1 CRC byte2	

4.1 Modbus addresses

The detailled address-table you find in the document: Parameter List for KS VARIO (9499-040-72911)
The address is coded in 2 bytes. The 2 most significant bits (D15, D14) are used to define the format in which the data are written or read.

The Modbus directory is divided into equal ranges of 512 words (bit D13...D09). Access to all data for one control channel (1...30 channels) is possible via each of these ranges.

There are 2 special ranges. All instrument data are stored in the lower part of the address range (Modbus addr. 0..512).

The most important process data of all 30 channels are stored additionally in the following range (addr. 512...1023).

This range is intended for accesses by visualization tools.
The signification of the individual address bits is:

INTEGER/ FIX-Point Modbus addresses:

MSB		LSB
D15 - D14	**D13 - D09**	**D08 - D00**
Data fomat **00: Integer** **01: Fix Point 1** **1X: Reserved for float**	**Instr., visualization, channel X** **00000: instrument data** **00001: visualization data** **00010: data channel 1** **00011: data channel 2** **....** **11111: data channel 30**	**Relevant datum**

Modbus directory (data format: integer):
Addresses 4000 hex must be added for the Fix Point 1 range.

Addresses	Data
0 511 (1FF hex)	Instrument data
512 (200 hex) 1023 (3FF hex)	Visualization range channel..30
1024 (400 hex) 1535 (5FF hex)	Data channel 1
1536 (600 hex) 2047 (7FF hex)	Data channel 2
....	....
15872 (3E00 hex) 16383 (3FFF hex)	Data channel 30

FLOAT Modbus addresses:

MSB		LSB
D15	**D14 - D10**	**D09 - D00**
Data format **0: Reserved for Integer and Fix Point 1** **1: Float**	**Instr., visualization, channel X** **00000: instrument data** **00001: visualization data** **00010: data channel 1** **00011: data channel 2** **....** **11111: data channel 30**	**relevant datum** (offset 2)

Modbus directory (data format: FLOAT):

Addresses	Data
32768 (8000 hex) 33791 (83FF hex)	Instrument data
33792 (8400 hex) 34815 (87FF hex)	Visualization area channel 1..30
34816 (8800 hex) 35839 (8BFF hex)	Data channel 1
35840 (8C00 hex) 36863 (8FFF hex)	Data channel 2
....	....
64512 (FC00 hex) 65535 (FFFF hex)	Data channel 30

4 bytes are required for storage of these data.

Transferable values:

Integer: -30000 ... +32000 (resolution: +/-1) Fix
Point 1: -3000.0 ...+3200.0 (resolution: +/- 0,1)
Float: -1.0 E+037...+1.0 E+037 (resolution: +/- 1.4E-045)

The following special values are defined for transmission in **integer format**:

-31000 This datum is not defined. This value is returned by the controller unless a datum within the block is defined when reading a block.

-32000 The function is switched off.

-32768 Corresponds to 0x8000 hex. The value to be transmitted is out of the transferable integer range.

The following special values are defined for transmission in **float format:**

1.5E37 This datum is not defined. This value is returned by the controller unless a datum within a block is defined when reading a block.

In the code tables (chapter 8), the addresses of each parameter for the relevant data format are specified in decimal values (Addr. = integer without digits behind the decimal point; 1 dP = integer with 1 digit behind the decimal point; real = float (IEEE format)).

4.2 Function codes

The following function codes are realized in CD VARIO:

Function code	Signification
0x03	Read process data, parameter or configuration data
0x04	Read process data, parameter or configuration data
0x06	Write a single datum (process data, parameter or configuration)
0x08	Diagnosis
0x10	Write several data (process data, parameter or configuration)

4.2.1 Read process data data, parameter or configuration data

The structure of a message is:

Inquiry:

Field name	Value	Signification
Address	11	Address 17
Function	03 or 04	Read process data data, parameter or configuration data
Start address high Start address low	04 98	Start address 0498 (ti1 / channel 1)
Number of values	00 02	2 data
CRC	CRC byte1 CRC byte2	

Reply:

Field name	Value	Signification
Address	11	Address 17
Function	03 or 04	Read process data data, parameter or configuration data
Number of bytes	04	4 data bytes are sent
Parameter1	00 B4	Process data data, parameter/configuration datum 0498= 180
Parameter ti2	01 4D	Process data data, parameter/configuration datum 0499= 333
CRC	CRC byte1 CRC byte2	

Broadcast is not possible.

Unless the 1st parameter / configuration datum was defined, error message "ILLEGAL DATA ADDRESS" is generated. Unless other values after the 1st value in the area to be read out are defined, these are entered with value "NOT DEFINED VALUE". This can be used for reading out areas with gaps using a message

4.2.2 Write a single datum (process data, parameter or configuration)

The structure of a message is:

Inquiry:

Field name	Value	Signification
Address	11	Address 17
Function	06	Write a single datum (process data, parameter or configuration)
Write address high Write address low	0D 57	Write address 15990 (SetpInterface of channel 30)
Value = 123	00 7B	
CRC	CRC byte1 CRC byte2	

Reply:

Field name	Value	Signification
Address	11	Address 17
Function	06	Write a single datum (process data, parameter or configuration)
Write addr. high Write addr. low	3E 76	Write address 15990 (SetpInterface of channel 30)
Value = 123	00 7B	
CRC	CRC byte1 CRC byte2	

The structure of a correct reply message is exactly as defined.
Broadcast is possible.

Entry in real data format is not possible, because only values of 2 bytes can be transmitted.

If the value is out of the adjustable range, error message ”ILLEGAL DATA VALUE” is generated. The datum remains unchanged.

Unless the datum can be written (e.g. configuration datum and the instrument is on-line), error message ”ILLEGAL DATA VALUE” is generated.

4.3 Diagnosis

Using a diagnosis message, the controller can be caused to return check messages, to go to operating statuses, to output counter states or to reset the counters.
Broadcast is **generally not** possible.
The following functions were defined:

Code	Signification
0x00	Return the received message
0x01	Communication restart (terminates the listen-only mode)
0x02	Status register feedback
0x04	Change to the listen-only mode
0x0A	Delete counters and reset the diagnosis register
0x0B	Return the message counter (all messages on the bus)
0x0C	Return the counter of faulty message transmissions to this slave (parity or CRC error)
0x0D	Return the counter of messages replied with error message
0x0E	Return the counter of messages for this slave
0x0F	Return the counter of non-replied messages
0x10	Return the counter of messages replied with NAK
0x11	Return the counter of messages replied with busy
0x12	Return the counter of too long messages
0x40	Return the parity error counter
0x41	Return the framing error counter (stop bit not detected)
0x42	Return the buffer full counter (message longer than receive buffer)

Inquiry in integer format

When using the setting for integer with digits behind the decimal point with the address (3 most significant address bits), the counter statuses are incremented according to the relevant conversion factor.

Inquiry in float format

When using the setting for float with the address (3 most significant address bits are 100), the counter statuses are transmitted in IEEE format. The highest value is 65535, because the counters in the instrument are designed as word counters.
In float format, a 4-byte data field is returned when reading the counter statuses. In all other cases, reply is with a 2-byte data field.
When switching over to listen mode (0x04) and with restart after switching the instrument to listen mode, no reply is generated.
When receiving a restart diagnosis message whilst the instrument is not in listen mode, the instrument generates a reply.
The general structure of a diagnosis message is: Inquiry:

Field name	Value	Signification
Address	11	Address 17
Function	0 8	Diagnosis message
Sub-function high Sub-function low	00 YY	Sub-function code
Data field	Byte 1 Byte 2	Further data definitions
CRC	CRC byte1 CRC byte2	

4.3.1 Returning a received message

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 00	2 bytes with any content	Returning received datum

Serves for checking, if the communication works generally.

4.3.2 Communication restart (terminates the listen-only mode)

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 01	00 00	00 00

The slave is requested to initialize its interface and to delete the event counters. Moreover, the instrument shall leave the listen-only mode. If the unit was in listen-only mode, no reply is generated.

4.3.3 Returning the diagnosis register

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 02	00 00	Contents of diagnosis register

The slave sends its 16-bit diagnosis register to the master. Which data are contained in this register can be defined freely. Information can be (faulty EEPROM, defective LED, etc.).

4.3.4 Changing to listen-only mode

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 04	00 00	No reply

The slave is requested to stop executing and replying any messages sent to it. This condition can be removed from the instrument only by means of diagnosis message sub-function 00 01. The normal instrument operating condition can be re-established also by power up.
The purpose of this function is to switch off a module with faulty behaviour on the Modbus, i.e. to enable the bus to continue operating. After receiving this message, the instrument does not generate a reply.

4.3.5 Deleting the counters and diagnosis register

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 0A	00 00	00 00

The slave is requested to delete its event counters and to reset the diagnosis register.

4.3.6 Returning the message counter

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 0B	00 00	Message counter

The slave is asked to return the value of its message counter.
The counter contains the sum of all messages recorded on the bus, whereby all messages which were sent by the master and by other slaves are also counted. Its own replies are not included in this value.

4.3.7 Returning the counter of faulty message transmissions

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 0C	00 00	Counter of faulty message transmissions

The slave is requested to return the value of its counter for faulty message transmission.
The counter contains the sum of all messages sent to the slave, at which an error was detected. These errors can be CRC errors or parity errors.

4.3.8 Returning the counter of the messages replied with error message

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 0D	00 00	Counter of messages replied with error message

The slave is requested to return the value of its counter for messages replied with error message. The counter contains the sum of all messages sent to the slave, which were replied with an error message by the slave.

4.3.9 Returning the counter of messages for this slave

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 0E	00 00	Counter of messages for this slave

The slave is requested to return the value of its message counter for this slave. The counter contains the sum of all messages sent to the slave.

4.3.10 Returning the counter of non-replied messages

Definition of received and returned data

Sub-function	Receiveddata field	Sent data field
00 0F	00 00	Counter of non-replied messages

The slave is requested to return the value of its counter of non-replied messages. The counter contains the sum of all messages sent to the slave, which were not replied due to internal events or detected errors.

4.3.11 Returning the counter of messages replied with NAK

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 10	00 00	Counter of messages replied with NAK

The slave is requested to return the value of its counter of messages replied with NAK. The counter contains the sum of all messages sent to the slave, which were replied with NAK.

4.3.12 Returning the counter of messages replied with busy

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 12	00 00	Counter of messages replied with busy

The slave is requested to return the value of its counter of messages replied with busy. The counter contains the sum of all messages sent to the slave, which were replied with busy.

4.3.13 Returning the counter with parity error

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 40	00 00	Counter of the number of parity errors

The slave is requested to return the value of its counter with the number of parity errors. The counter contains the sum of all messages sent to the slave at which a parity error was detected.

4.3.14 Returning the counter with framing error

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 41	00 00	Counter of the number of framing errors

The slave is requested to return the value of its counter with the number of framing errors. The counter contains the sum of all messages sent to the slave at which a framing error was detected. Framing error is detected, unless the stop bit at the end of a byte is detected.

4.3.15 Returning the counter of too long messages

Definition of received and returned data

Sub-function	Received data field	Sent data field
00 42	00 00	Counter of too long messages

The slave is requested to return the value of its counter of too long messages.
The counter contains the sum of all messages sent to the slave at which there was an overflow of the receive buffer or at which data could not be fetched in time by the UART.

4.4 Writing several process data data, parameter and configuration

The structure of a message is:
Definition:

Field name	Value	Signification
Address	11	Address 17
Function	10	Write several process data data, parameter or configuration data
Start address high Start addresse low	0D 57	Write address 3415
Number of values	00 02	2 values
Number of bytes	04	4 data bytes are sent
Parameter/configuration datum 15	00 DE	Process data datum, parameter or configuration datum 3415 = 222
Parameter/configuration datum 16	01 4D	Process data datum, parameter or configuration datum 3416 = 333
CRC	CRC byte1 CRC byte2	

Reply :

Reply:	**Value**	**Signification**
Field name	11	Address 17
Address	10	Write several process data data, parameter or configuration
Function	0D	Data
Start address high	57	Write address 3415
Start address low	00	2 process data data, parameter/configuration data
Number of values	02	

Broadcast is possible.
Unless the 1st value was defined, an error message ”ILLEGAL DATA ADDRESS” is generated.
Unless the 1st value can be written (configuration and instrument is on-line), an error message ”ILLEGAL DATA VALUE”
is generated.
Unless other values in the defined range after the 1st value are defined or can be written instantaneously, these values are overread. Data in these positions are not changed. The purpose is to change parts with gaps or which cannot be written instantaneously by means of a message. No error message is output.
If values are out of the adjustable limits, error message ”ILLEGAL DATA VALUE” is generated. Evaluation of the following data is omitted. Data which were already stored correctly are active.
The Modbus does not provide information related to the error position in its error report. If this is required, a datum containing the position of the last error must be defined. In case of error, this datum can be read out by the master.

4.5 Read-out and specification of data in float format

Level-1 data, parameter and configuration data in float format can be read out and written. (Function codes 03, 04, 16)
Writing single data in float format with code 06 is not possible, since only 2 bytes for the value of the datum can be transmitted by means of this function.
If data in float format are required, the address of the required datum must be calculated as follows:
Address of the datum in integer format multiplied with factor 2
Addition of an offset of 8000H.
In "Number of values", a value twice as high as with a message for data in integer format is required.
Accordingly, the value in field "Number of data bytes" is twice as high.
All data are always converted into float values. This is also applicable to status or control words.
The data are transmitted in Motorola format (exponent followed by mantissa first).
The float format structure of a message as described in the previous chapter is:

Definition:

Field name	Value	Signification
Address	11	Address 17
Function	10	Write several process data data, parameter or configuration data
Start address high Start address low	9A AE	Write address 2 * 3415 + 8000H for float format
Number of values	00 04	2 values in float format
Number of bytes	08	8 data bytes are sent
Parameter/configuration datum 15	43 5E 00 00	Process data datum, parameter or configuration datum 3415 = 222
Parameter/configuration datum 16	43 A6 80 00	Process data datum, parameter or configuration datum 3416 = 333
CRC	CRC byte1 CRC byte2	

Definition:

Field name	Value	Signification
Address	11	Address 17
Function	10	Write several process data data, parameter or configuration data
Start address high Start address low	9A AE	Write address 2 * 3415 + 8000H for float format
Number of values	00 04	2 process data data, parameter or configuration data in float format
CRC	CRC byte1 CRC byte2	

. 5 Error report

The error report is generated, when interpretation or changing a datum are not possible, although the message was received correctly.
When detecting a transmission error, **no** reply is given. The master must resend the message.

Detected transmission errors are:

- **Parity error**
- **Framing error (no stop bit received)**
- **Overrun error (receive buffer overflow, or data could not be fetched in time by the UART)**
- **CRC error**

The data structure of the error report is:

Field name	Value	Signification
Address	11	Address 17
Function	90	Error report for message Write several parameter/configuration data
Error Code	02	ILLEGAL DATA ADDRESS
CRC	CRC byte1 CRC byte2	

In field Function, the most significant bit is set.
The error code is transmitted in the following byte.
The following error codes are defined:

CODE	Name	Signification
01	ILLEGAL FUNCTION	The received function code is not defined in the instrument.
02	ILLEGAL DATA ADDRESS	The received address is not defined in the instrument. When reading (function code 01, 03, 04) or writing (function code 0F, 10) several data simultaneously, this error is generated only, unless the first datum is defined.
03	ILLEGAL DATA VALUE	The received value is out of the adjusted limits or cannot be written instantaneously (instrument is not in configuration mode). When writing several data simultaneously (function code 0F, 10), this error is generated only, unless the first datum can be written.

The Modbus protocol includes further defined error codes, which, however, are presently not supported:

CODE	Name	Signification
04	SLAVE DEVICE FAILURE	A non-reproducible error occurred during message processing.
05	ACKNOWLEDG E	The instrument has received an inquiry and handles it. As handling takes a very long time, this reply is output to prevent an interface timeout. The master can poll the diagnosis, to find out if handling is finished.
06	SLAVE DEVICE BUSY	The instrument is busy handling an order. The master must repeat the message subsequently.
07	NEGATIVE	The instrument cannot handle the requested order. This error message can be output for changing a configuration datum, although the instrument is not in configuration mode.
08	ACKNOWLEDG E	Parity error found when reading the memory.

. 6 Hints on operation

6.1 Connecting the interface

The MODBUS is connected to socket B.
Rear serial interface, RS485-based physical signals.
Suitable cables must be provided by the user.

6.1.1 Cable installation

When laying cables, the general wiring hints must be followed:

- **Cable run inside buildings (inside and outside cabinets)**
- **Cable run outside buildings**
- **Potential compensation**
- **Cable screening**
- **Measures against interference voltage**
- **Length of tap line**

Termination of the bus cable at both segment ends by means of terminating resistors ensures that:
a defined rest potential on the cable is adjusted, cable reflections are minimized, and nearly constant load behaviour at the bus is adjusted.

6.2 System construction

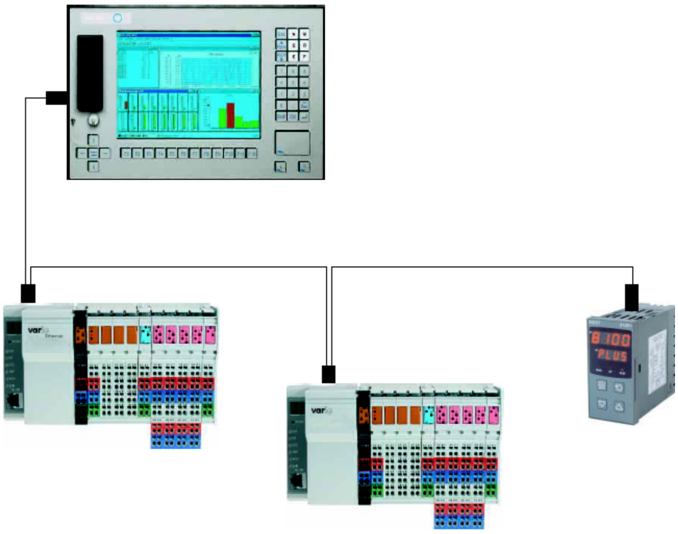
6.2.1 Minimum equipment of a MODBUS system

A Modbus system comprises the following minimum components:
a bus master, which controls the data communication, one or several slave units, which make data available on request by the master, the transfer medium comprising bus cable and bus connector for connecting the individual units, one or several bus segments, which are connected with repeaters.

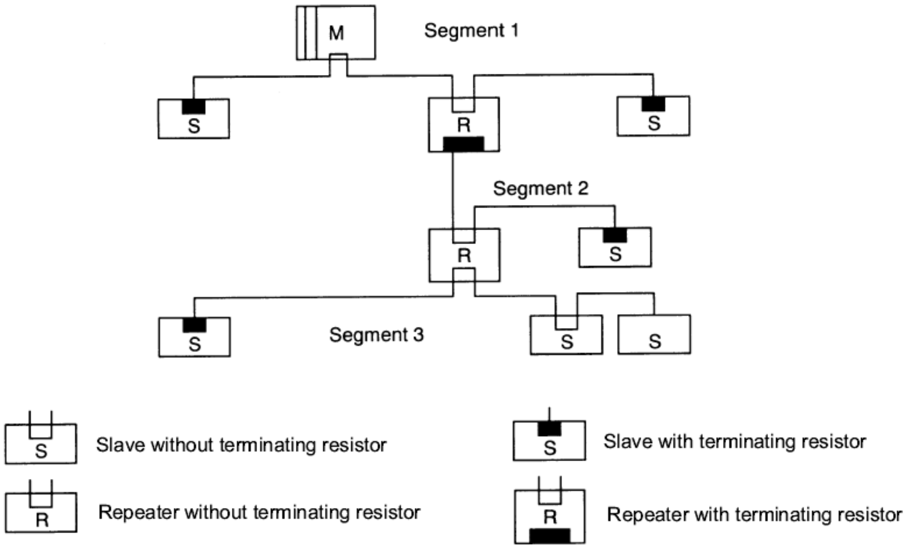
6.2.2 Maximum equipment of a Modbus system

A bus segment comprises max. 32 field instruments (active and passive ones). The maximum number of slave units, which can be operated at a MODBUS master over several segments is determined by the internal memory structure of the master. Therefore you should inform yourself on the master capacity when planning a system. The bus cable can be opened at any point to include another unit by adding a bus connector. At the end of a segment, extending the bus cable up to the specified segment lengths and including new units for extensions are possible. The length of a bus segment is dependent of adjusted Baudrate. The Baudrate is determined mainly by the system constellation (segment length, number of distributed input and outputs) and the required polling intervals of individual units. For all units connected on the bus, the same Baudrate must be selected.

At the start and end of a segment, terminating resistors must be connected to ensure a physically clean signal level.



Modbus instruments must be connected in line structure.
A Modbus installation can be extended by connecting repeaters,
if more than 32 units must be connected
or for connection over longer distances than defined for the Baudrates.

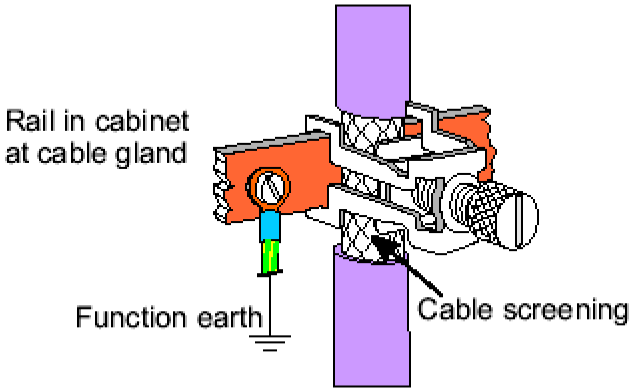


A fully extended MODBUS system can include max. 126 stations with addresses 0 ... 125. Each repeater reduces the maximum number of stations within a segment. No MODBUS unit address isassigned to a passive unit. Nevertheless, its input circuitry is an additional load due to the bus driver power consumption. However, a repeater is without effect on the overall number of stations connected on the bus. The maximum connectable number of repeaters which may be connected in series can differ dependent of manufacturer. Therefore, information on any limitations should be asked for in advance from the manufacturer when planning a system.

6.2.3 Cable run inside buildings

The following hints for cable installation are applicable to a pairwisely twisted, screened two-wire cable. The screening is used for improvement of the electromagnetic compatibility. With type A MODBUS cable, meshed screening and foil screening are integrated into the cable. The following cable screening versions always include the two screening versions (meshed and foil). Using only the foil screening must be omitted, because it is very thin and can be interrupted easily, which may cause interruption of the potential equalization.

Both ends of the cable screening must be connected to the reference potential via a large surface of conducting material. When installing a repeater or a field unit in a cabinet, the cable screening should be connected to a screening rail via cable collar near the cable gland.



The screening must be continued up to the field instrument and connected with the conducting housing and/or the metal connector. Ensure that the ground potential of the instrument housing and of the control cabinet accommodating the field instrument are equal due to large-surface metal contact. Mounting a screening rail on a painted surface is without effect. These measures ensure grounding of high-frequency interference via the meshed screening. With external interference voltage on the data lines despite these measures, increase the voltage potential on the two data lines regularly so that the difference voltage is normally not destroyed. In ordinary cases, safe data transmission is still ensured with a shift of the ground potential by some volts. With higher shifts, a potential equalization lead with a minimum cross section of 100 mm² should be installed in parallel to the bus cable and connected with the reference potential of each field unit. Normally, the field instruments are fitted with a grounding screw. With extreme interference effect, the bus cable can be installed additionally in a steel tube or a tight sheet metal duct. The tube or duct must be grounded correctly.

The minimum distance between bus cable and other cables for voltages exceeding 60 V must be 20 cm. The bus cable should also be kept separate from telephone cables and cables leading into the explosion-hazarded area. In these cases, we recommend using a separate cable duct for the bus cable.

Only conducting materials with regular connection to the reference potential should be used for the cable duct. The bus cables must not be subjected to mechanical stress or obvious damage. Unless this condition is met, special protective measures, e.g. installation in tubes, etc., are necessary.

Non-grounded systems :
The construction of a non-grounded system may be necessary for various reasons. For this purpose, there must be a high-impedance connection between instrument ground and reference potential (e.g. by means of RC protective circuitry). In this case, the system generates its own potential. When connecting bus segments by means of repeaters, we recommend using the non-grounded construction, to prevent transmission of potential differences between bus segments.