

DigiTrace TCONTROL-CONT-03

Compact microprocessor controller
Régulateur compact géré par microprocesseur
Kompakter Mikroprozessorregler



INSTALL-160 Rev. 1

Modbus Manual
Notice Modbus
Modbus-Anleitung

2013-05-21/00521759

Contents

1	Introduction	5
1.1	Preface	5
1.2	Typographical conventions	5
2	Protocol description	7
2.1	Master-Slave principle	7
2.2	Transmission mode (RTU)	7
2.3	Device address	8
2.4	Timing of the communication	8
2.5	Structure of the data blocks	11
2.6	Function codes	12
2.6.1	Read n words	12
2.6.2	Write one word	13
2.6.3	Write n words	14
2.7	Transmission format (integer, float and text values)	15
2.8	Checksum (CRC16)	17
2.9	Error processing	18
3	RS485 interface	19
3.1	Connection diagram	19
3.2	Configuration	20
4	Modbus addresses	21
4.1	Process data	21
4.2	Set point values	22
4.3	Controller parameters	23
4.4	Configuration	23
4.5	Commands	24
4.6	RAM memory	25

1.1 Preface

This operating manual is addressed to the system manufacturer with adequate technical background and PC related knowledge.

Please read this operating manual prior to commissioning the device. Keep the manual in a place accessible to all users at all times. Your comments are appreciated and may assist us in improving this manual.

All necessary settings are described in this operating manual. Should problems be encountered during commissioning, please refrain from carrying out any manipulations that are not described in the manual. Any such intervention will jeopardize your warranty rights. Please contact the nearest subsidiary or the head office.

1.2 Typographical conventions

Warning signs:

Caution



This symbol is used when there may be **damage to equipment or data** if the instructions are ignored or not followed correctly!

Note signs:

Note



This symbol is used when your **special attention** is drawn to a remark.

Reference



This symbol refers to **further information** in other manuals, chapters or sections.

Number types:

Hexadecimal number

0x0010

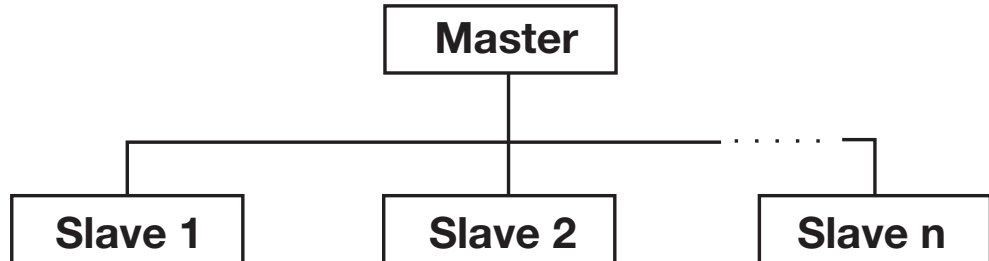
A hexadecimal number is identified by „0x“ preceding the actual number (here: 16 decimal).

1 Introduction

2 Protocol description

2.1 Master-Slave principle

Communication between a master (e.g. PC) and a slave (e.g. measuring and control system) using Modbus takes place according to the master-slave principle, in the form of data request/instruction - response.



The master controls the data exchange, the slaves only have a response function. They are identified by their device address.

2.2 Transmission mode (RTU)

The transmission mode used is the RTU mode (Remote Terminal Unit). Data is transmitted in binary format (hexadecimal) with 8 bits. The LSB (least significant bit) is transmitted first. The ASCII operating mode is not supported.

Data format

The data format describes the structure of a character transmitted. The following data format options are available:

Data word	Parity bit	Stop bit 1/2 bit	Number of bits
8 bits	—	1	9
8 bits	even	1	10
8 bits	odd	1	10
8 bits	—	2	10

2 Protocol description

2.3 Device address

The device address of the slave can be set between 0 and 254. Device address 0 is reserved.



A maximum of 31 slaves can be addressed via the RS485 interface.

There are two different forms of data exchange:

Query

Data request/instruction by the master to a slave via the corresponding device address.

The slave addressed responds.

Broadcast

Instruction by the master to all slaves via the device address 0 (e.g. to transmit a specific value to all slaves).

The connected slaves do not respond. In such a case, the correct acceptance of the value by the slaves should be checked by a subsequent readout at each individual slave.

Data request with the device address 0 is meaningless.

2.4 Timing of the communication

Start and end of a data block are marked by transmission pauses. The maximum permitted interval between two consecutive characters is three times the transmission time required for a single character.

The character transmission time (time required to transmit one single character) depends on the baud rate and the data format used (stop bits and parity bit).

For a data format of 8 data bits, no parity bit and one stop bit, this is:

character transmission time [ms] = 1000 * 9 bit/baud rate

For the other data formats, this is:

**character transmission time [ms]
= 1000 * (8 bits+parity bit+stop bit(s)) bit/baud rate**

2 Protocol description

Timing

Data request from master transmission time = $n \text{ characters} * 1000 * x \text{ bit/baud rate}$
Marker for end of data request $3 \text{ characters} * 1000 * x \text{ bit/baud rate}$
Processing of data request by the slave ($\leq 250\text{ms}$)
Response of the slave transmission time = $n \text{ characters} * 1000 * x \text{ bit/baud rate}$
Marker for end of response $3 \text{ characters} * 1000 * x \text{ bit/baud rate}$

Example

Marker for end of data request or end of response for a 10/9 bit data format

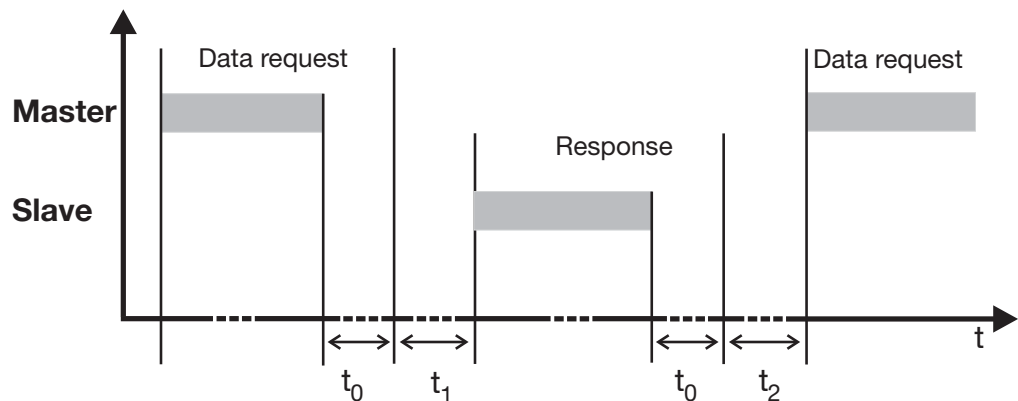
Waiting time = $3 \text{ characters} * 1000 * 10 \text{ bit/baud rate}$

Baud rate [baud]	Data format [bit]	Waiting time [ms] (3 characters)
38400	10	0.79
	9	0.71
19200	10	1.57
	9	1.41
9600	10	3.13
	9	2.82

2 Protocol description

Timing scheme

A data request runs according to the following timing scheme:



- t_0 End marker = 3 characters
(time depending on the baud rate)
- t_1 This time depends on the internal processing.
The maximum processing time is 250 ms.



A minimum response time can be set in the device under the menu point item „Interface“. This preset time is the minimum waiting time before an answer is transmitted (0...500 ms). If a smaller value is set, then the response time may be longer than the preset value (internal processing takes longer), the device answers as soon as internal processing is completed. The preset time of 0 ms means that the device responds at the maximum possible speed.

The minimum response time, which can be set is required by the RS485 interface in the master, to be able to switch over the interface drivers from transmit to receive.

- t_2 This time is needed by the slave to change from transmit back to receive. The master has to observe this waiting time before presenting a new data request. This time must always be observed, even when the new data request is directed to another device.

RS485 interface: $t_2 = 10\text{ms}$

No data requests from the master are permitted during t_1 and t_2 and during the slave response time. Data requests made during t_1 and t_2 are ignored by the slave. Data requests during the response time will result in the invalidation of all data currently on the bus.

2.5 Structure of the data blocks

All data blocks have the same structure:

Data structure

Slave address	Function code	Data field	Checksum CRC16
1 byte	1 byte	x byte	2 bytes

Each data block contains four fields:

Slave address	device address of a specific slave
Function code	function selection (read, write words)
Data field	contains the following information: <ul style="list-style-type: none">- word address- number of words- word value(s)
Checksum	detection of transmission errors

2 Protocol description

2.6 Function codes

The functions described in the following are available for the readout of measured values, device and process data as well as to write specific data.

Function-overview

Function number	Function	Limitation
0x03 or 0x04	Read n words	max. 32 words (64 bytes)
0x06	Write one word	max. 1 word (2 bytes)
0x10	Write n words	max. 32 words (64 bytes)



Please refer to Chapter 2.9 Error processing, Page 18 if the device does not react to these functions or emits an error code.

2.6.1 Read n words

This function is used to read n ($n \leq 32$) words starting from a specific address.

Data request

Slave address	Function	Address first word	Number of words (max. 32)	Checksum CRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Response

Slave address	Function	Number of bytes read	Word value(s)	Checksum CRC16
1 byte	1 byte	1 byte	x byte	2 byte

Example

Reading the W1 and W2 set point values (2 words each)

Address of first word = 0x3100 (W1 set point value)

Data request:

01	03	3100	0004	4AF5
----	----	------	------	------

Response (values in the Modbus float format):

01	03	08	0000	41C8	0000	4120	4A9E
			Set point value W1 (25.0)	Set point value W2 (10.0)			

2 Protocol description

2.6.2 Write one word

For the Write Word function, the data blocks for instruction and response are identical.

Instruction

Slave address	Function 0x06	Word address	Word value	Checksum CRC16
1 byte	1 byte	2 byte	2 bytes	2 byte

Response

Slave address	Function 0x06	Word address	Word value	Checksum CRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Example

Write the limit value AL of limit comparator 1 = 275.0
(Value = 0x80004389 in the Modbus float format)

Word address = 0x0057

Instruction: Write the first part of the value

01	06	0057	8000	59DA
----	----	------	------	------

Response (as instruction):

01	06	0057	8000	59DA
----	----	------	------	------

Instruction: Write the second part of the value (next word address)

01	06	0058	4389	F88F
----	----	------	------	------

Response (as instruction):

01	06	0058	4389	F88F
----	----	------	------	------

2 Protocol description

2.6.3 Write n words

This function is used to write n ($n \leq 32$) words starting from a specific address.

Instruction

Slave address	Function 0x10	Address first word	Number of words (max. 32)	Number of bytes	Word value(s)	ChecksumC RC16
1 byte	1 byte	2 byte	2 byte	1 byte	x byte	2 byte

Response

Slave address	Function 0x10	Address first word	Number of words	Checksum CRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Example

Writing the W1 and W2 set point values (2 words each)

Word address = 0x3100 (W1 set point value)

Instruction:

01	10	3100	0004	08	0000	41C8	0000	4120	2A42
					Set point value W1 (25.0)		Set point value W2 (10.0)		

Response:

01	10	3100	0004	CF36
----	----	------	------	------

2 Protocol description

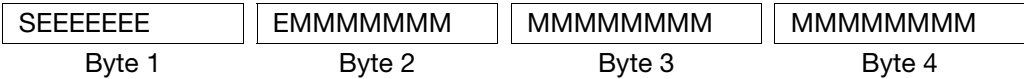
2.7 Transmission format (integer, float and text values)

Integer values Integer values are transmitted via the Modbus in the following format:
The high byte first, followed by the low byte.

Example Request of the integer value of address 0x0021, if value "4" (word value 0x0004) is written under this address.
Request: 01 03 0021 0001 (+ 2 bytes CRC16)
Response: 01 03 02 **0004** (+ 2 bytes CRC16)

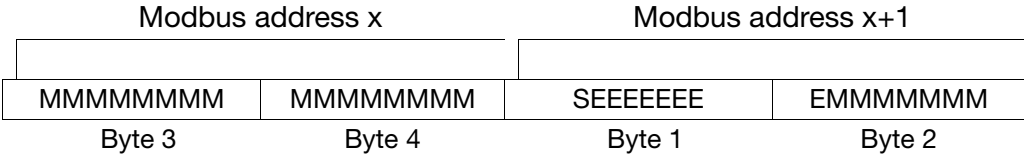
Float values In the case of float values, the Modbus operates with the IEEE-754 standard format (32bits), the only difference being that byte 1 and 2 are changed over with byte 3 and 4.

Single-float format (32bit) as per IEEE 754 standard



S - sign bit
E - exponent (two's complement)
M - 23bits normalized mantissa

Modbus float format

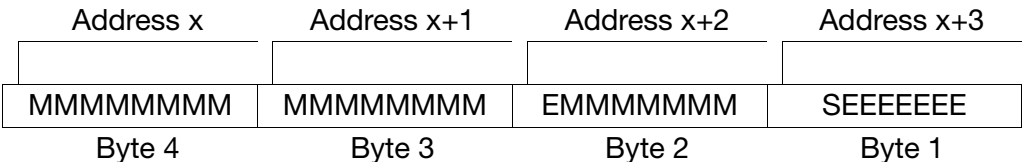


Example Request of the float value of address 0x0035, if value "550.0" (0x44098000 in IEEE-754 format) is written under this address.
Request: 01 03 0035 0002 (+ 2 bytes CRC16)
Response: 01 03 04 **8000 4409** (+ 2 bytes CRC16)

Once transmission from the device is completed, the bytes of the float value need to be changed over accordingly.

A large number of compilers (e.g. Microsoft Visual C++) store the float values in the following order:

Float value



Please find out the way float values are stored in your application. After the request, it might be necessary to change the bytes over in the interface program you are using.

2 Protocol description

Character strings (texts)

Character strings (texts) are transmitted in the ASCII format.



To mark the end, the last character to be transmitted can be a "\0" (ASCII code 0x00). Characters after this mark are without significance.

The address tables show the max. possible number of characters in a data type, e.g. "TEXT24" (24 characters). When an end mark is used, then only 23 readable characters are available for the text in this example.

If no end mark is used, the use of the maximum number of characters (e.g. TEXT8 = 8 characters) indicated in the data type is required. This prevents characters still contained in the memory from being appended to the text.

Knowing that the transmission of texts takes place word by word (16 bits), 0x00 is additionally appended where an odd number of characters is used (incl. "\0").

Example for data type TEXT4

Read the text (here: "AbC ") under address 0x0067 (a max. of 4 characters can be saved)

ASCII code for "AbC " (with one space at the end):
0x41, 0x62, 0x43, 0x20

Request: 01 03 0067 0002 (+ 2 byte CRC16)

Slave address = 01

Function = 03, i.e. read n words

Address = 0067

Number of words to be read = 0002, because of the maximum of 4 characters

Response: 01 03 04 **41 62 43 20** (+ 2 byte CRC16)

Slave address = 01

Function = 03, i.e. read n words

Number of bytes read = 04

Variant:

ASCII code for "Ab" (without a space at the end):

0x41, 0x62, 0x00

ASCII 0x00 ("\0") marks the end of the character string.

During transmission, **0x00** is additionally appended to obtain an even number of characters.

Response in this case: 01 03 04 **41 62 00 00** (+ 2 byte CRC16)

2.8 Checksum (CRC16)

The checksum (CRC16) serves to recognize transmission errors. If an error is identified during evaluation, the device concerned does not respond.

Calculation scheme

CRC = 0xFFFF	
CRC = CRC XOR ByteOfMessage	
For (1 to 8)	
CRC = SHR(CRC)	
if (flag shifted right = 1)	
then	else
CRC = CRC XOR	
0xA001	
while (not all ByteOfMessage processed);	



The low byte of the check sum is the first to be transmitted, then the high byte.

Example

Data request: Read two words, starting at address 0x00CE
(CRC16 = 0x92A5)

07	03	00	CE	00	02	A5	92
						CRC16	

Response: (CRC16 = 0xF5AD)

07	03	04	00	00	41	C8	AD	F5
			Word 1		Word 2		CRC16	

2 Protocol description

2.9 Error processing

Error codes The following error codes exist:

- 1 invalid function
- 2 invalid parameter address or too many words are to be read or written
- 8 write access to parameter denied

Response in the event of an error

Slave address	Function XX OR 80h	Error code	Checksum CRC16
1 byte	1 byte	1 byte	2 bytes

0x80 is used to set the function code to its OR status, i.e. the MSB (most significant bit) is set to 1.

Example

Data request:

01	03	40	00	00	04	CRC16
----	----	----	----	----	----	-------

Response (with error code 2):

01	83	02	CRC16
----	----	----	-------

Special cases

The slave not responding can have the following causes:

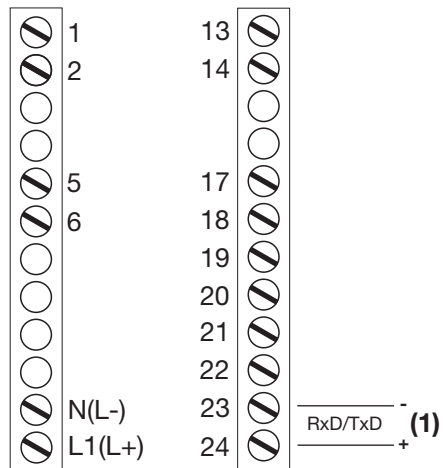
- the baud rate and/or data format of Master and Slave are not compatible
- the device address used does not coincide with that of the slave address
- the checksum (CRC16) is not correct
- the instruction from the Master is incomplete or over-defined
- The number of words to be read is zero.

In these cases the data request should be transmitted again once the timeout time (2 s) has elapsed.

3.1 Connection diagram



This device can be ordered with an RS485 interface as an option. Please refer to the Operating and Installation manual INSTALL-148 for the type designation.



(1) RS485 interface



Connect the interface line shielding to earth on one side in the switch cabinet.

3 RS485 interface

3.2 Configuration

The following table shows the possible Modbus interface settings to be carried out in the configuration level (*CONF* → *INTF*) and/or in the setup program.



For more detailed information about configuration, please refer to the Operating and Installation manual INSTALL-148.

	Symbol	Value/ Selection	Description
Baud rate	<i>bdr</i>	0	9600 bps
		1	19200 bps
		2	38400 bps
Data format	<i>dft</i>	0	8 data bits, 1 stop bit, no parity
		1	8 data bits, 1 stop bit, odd parity
		2	8 data bits, 1 stop bit, even parity
		3	8 data bits, 2 stop bits, no parity
Device address	<i>Adr</i>	0...1 ...255	Address in data network Addresses 0 and 255 are reserved for specific purposes and should not be used here.
Minimum response time	(Setup)	0 ...500ms	Minimum time period that elapses between the request of a device in the data network and the response of the controller (can only be adjusted via the setup program).

Factory settings are shown **bold**.



When the communication takes place via the setup interface, the RS485 interface is inactive.

4 Modbus addresses

Data type, type of access The following tables contain specifications of all process and device data including their addresses, data type and type of access.

Meaning:

R/O	Read only access
W/O	Write only access
R/W	Read/write access
INT	Integer (8 or 16 bit)
Bit x	Bit No. x (bit 0 is always the bit with the lowest value)
LONG	Long integer (4 byte)
FLOAT	Float value (4 bytes) as per IEEE 754
TEXT4	Text 4 characters



Write operations to R/W parameters result in them being saved to the EEPROM. These memory modules only have a limited number of write cycles (approx. 100000). For this reason, this function can be switched off in the case of frequent programming. The parameter values are then only saved in the volatile memory (RAM) and will be lost after a supply failure.

⇒ *Setup program (undocumented parameters -> Bit parameters -> Set parameter 2)*

4.1 Process data

Address	Data type/ bit number	Access	Signal designation
0x001F	INT	R/O	Internal binary values
	Bit 12		Binary value L1 (= 0x1000)
	Bit 13		Binary value L2 (= 0x2000)
0x0020	INT	R/O	Controller status
	Bit 12		Manual mode active (=0x1000)
	Bit 15		Self-optimization active (=0x8000)
0x0021	INT	R/O	Binary outputs 1...4 (Switching states 0 = off / 1 = on)
	Bit 0		Output K1: Relay (= 0x0001)
	Bit 1		Output K2: Relay (= 0x0002)
	Bit 2		Output K3: Logic (= 0x0004)
	Bit 3		Output K4: Relay (= 0x0008)
0x0023	INT	R/O	Binary inputs 1 and 2 (Switching states 0 = open / 1 = closed)
	Bit 0		Input 1 (= 0x0001)
	Bit 1		Input 2 (= 0x0002)

4 Modbus addresses

Address	Data type/ bit number	Access	Signal designation
0x0024	INT	R/O	Limit comparators 1...2
	Bit 0		Limit comparator 1 (= 0x0001)
	Bit 1		Limit comparator 2 (= 0x0002)
0x0025	INT	R/W	Control of the binary outputs (individual)
	Bit 0 + Bit 8		Output K1 (= 0x0101)
	Bit 1 + Bit 9		Output K2 (= 0x0202)
	Bit 2 + Bit 10		Output K3 (= 0x0404)
	Bit 3 + Bit 11		Output K4 (= 0x0808)
0x0026	FLOAT	R/O	Analog input [mV]
0x0028	FLOAT	R/O	Internal Pt100 [Ohm]
0x002A	INT	R/O	Sampling cycle time
0x002B	FLOAT	R/O	Analog input [displayed value]
0x002D	FLOAT	R/O	Internal analog value 1
0x002F	FLOAT	R/O	Internal analog value 2
0x0031	FLOAT	R/O	Controller, ramp limit value
0x0033	FLOAT	R/O	Controller, actual value, FILTERED
0x0035	FLOAT	R/O	Controller, actual value, UNFILTERED
0x0037	FLOAT	R/W	Controller set point value
0x0039	FLOAT	R/O	Controller, output value display
0x003B	FLOAT	R/O	Controller, output value, HEATING
0x003D	FLOAT	R/O	Controller, output value, COOLING
0x003F	FLOAT	R/O	Controller, control difference
0x0041	FLOAT	R/O	Controller, control deviation
0x0043	INT	R/O	Controller, switching state, HEATING
0x0044	INT	R/O	Controller, switching state, COOLING
0x0046	INT	R/O	Output value, manual mode
0x0047	LONG	R/O	Timer run time
0x0049	LONG	R/O	Residual timer time
0x004B	INT	R/O	Timer status
	Bit 1		Timer stopped (= 0x0002)
	Bit 5		Timer runs (= 0x0020)
	Bit 6		Timer end (= 0x0040)
	Bit 15		Timer signal (= 0x8000)

4.2 Set point values

Address	Data type/ bit number	Access	Signal designation
0x3100	FLOAT	R/W	Set point value W1
0x3102	FLOAT	R/W	Set point value W2

4.3 Controller parameters

Address	Data type/ bit number	Access	Signal designation
0x3000	FLOAT	R/W	Controller parameter XP1
0x3002	FLOAT	R/W	Controller parameter XP2
0x3004	FLOAT	R/W	Controller parameter TV
0x3006	FLOAT	R/W	Controller parameter TN
0x300C	FLOAT	R/W	Controller parameter CY1
0x300E	FLOAT	R/W	Controller parameter CY2
0x3010	FLOAT	R/W	Controller parameter XSH
0x3012	FLOAT	R/W	Controller parameter XD1
0x3014	FLOAT	R/W	Controller parameter XD2
0x3016	INT	R/W	Controller parameter TT
0x3017	INT	R/W	Controller parameter Y0
0x3018	INT	R/W	Controller parameter Y1
0x3019	INT	R/W	Controller parameter Y2

4.4 Configuration

Address	Data type/ bit number	Access	Signal designation
0x0053	FLOAT	R/W	Ramp function, ramp rate
0x0055	FLOAT	R/W	Filter time constant (digital filter)
0x0057	FLOAT	R/W	Limit comparator 1 Alarm value AL
0x0059	FLOAT	R/W	Limit comparator 1 Hysteresis
0x005D	FLOAT	R/W	Limit comparator 2 Alarm value AL
0x005F	FLOAT	R/W	Limit comparator 2 Hysteresis
0x0063	LONG	R/W	Timer value
0x0065	LONG	R/W	Service limit value
0x0067	TEXT4	R/W	Alarm text
0x0069	LONG	R/W	Service counter

4 Modbus addresses

4.5 Commands

Address	Data type/ bit number	Access	Signal designation
0x004D	INT	W/O	Binary functions CONTROLLER
	Bit 0		Self-optimization start (=0x0001)
	Bit 1		Self-optimization abort (=0x0002)
	Bit 2		Manual operation (= 0x0004)
	Bit 3		Automatic operation (= 0x0008)
	Bit 4		Controller off (= 0x0010)
	Bit 5		Manual mode inhibit (= 0x0020)
	Bit 6		Ramp stop (= 0x0040)
	Bit 7		Ramp abort (= 0x0080)
	Bit 8		Ramp restart (= 0x0100)
	Bit 9		Timer start (= 0x0200)
	Bit 10		Timer abort (= 0x0400)
	Bit 11		Timer stop (= 0x0800)
0x004E	INT	W/O	Binary functions OPERATION
	Bit 0		Keyboard inhibit (= 0x0001)
	Bit 1		Configuration and parameter level inhibit (= 0x0002)
	Bit 3		Display OFF (= 0x0008)
	Bit 5		Text display (= 0x0020)
0x004F	INT	W/O	Binary functions TIMER
	Bit 9		Timer start (= 0x0200)
	Bit 10		Timer abort (= 0x0400)
	Bit 11		Timer stop (= 0x0800)
0x0050	INT	R/W	Set point value toggling
	Bit 0		Set point value 1 (= 0x0001)
	Bit 1		Set point value 2 (= 0x0002)

4.6 RAM memory

Address	Data type/ bit number	Access	Signal designation
0x3200	FLOAT	W/O	Controller set point value (writable)
0x3202	FLOAT	W/O	Controller actual value (writable)
0x3204	FLOAT	W/O	Internal analog value 1 (writable)
0x3206	FLOAT	W/O	Internal analog value 2 (writable)
0x3208	INT	W/O	Internal binary values (writable)
	Bit 0 + Bit 7		Binary value L1 (= 0x0081)
	Bit 1 + Bit 7		Binary value L2 (= 0x0082)



Modbus allows direct access to the RAM memory of the controller for writing the controller set point value (0x3200), the controller actual value (0x3202) and the internal analog values (0x3204, 0x3206) as well as the internal binary values (0x3208).

When writing, a range between -1999 and +9999 is available for the controller set point values, controller actual values and the internal analog values. In this case, the data written in the controller is used instead of the original value.

If you wish to use the original value again on the controller, write the value 200001 at the memory location concerned via Modbus.

4 Modbus addresses

DigiTrace TCONTROL-CONT-03

Régulateur compact géré par microprocesseur



INSTALL-160 Rev. 1

Notice Modbus

Sommaire

1	Introduction	5
1.1	Avant-propos	5
1.2	Conventions typographiques	5
2	Description du protocole	7
2.1	Principe maître-esclave	7
2.2	Mode de transmission (RTU)	7
2.3	Adresse-appareil	8
2.4	Déroulement temporel de la transmission	8
2.5	Structure des blocs de données	11
2.6	Code des fonctions	12
2.6.1	Lecture de n mots	12
2.6.2	Ecriture d'un mot	13
2.6.3	Ecriture de n mots	14
2.7	Format de transmission (valeurs entières, flottantes et texte)	15
2.8	Somme de contrôle (CRC16)	17
2.9	Traitement des erreurs	18
3	Interface RS485	19
3.1	Schéma de raccordement	19
3.2	Configuration	20
4	Adresses Modbus	21
4.1	Données de process	21
4.2	Consignes	23
4.3	Paramètres du régulateur	23
4.4	Configuration	23
4.5	Commandes	24
4.6	Mémoire RAM	25

1.1 Avant-propos

Cette notice s'adresse aux constructeurs avec formation spécialisée et possédant des connaissances en PC.

Lisez cette notice avant de mettre en service l'interface.

Conservez cette notice dans un endroit accessible à tout moment à tous les utilisateurs. Aidez-nous à améliorer cette notice en nous faisant part de vos suggestions.

Tous les réglages nécessaires sont décrits dans cette notice. Si vous procédez à des manipulations non décrites dans cette notice ou expressément interdites, vous compromettez votre droit à la garantie. En cas de problèmes, veuillez prendre contact avec nos services.

1.2 Conventions typographiques

Signes d'avertissement :

Attention



Ce symbole est utilisé lorsque la non-observation ou l'observation imprécise des instructions peut **endommager les appareils ou les données** !

Symboles indiquant une remarque :

Remarque



Ce symbole est utilisé pour attirer votre attention sur un **point particulier**.

Renvoi



Ce symbole renvoie à des **informations complémentaires** dans d'autres notices, chapitres ou sections.

Types de représentation:

Nombre hexadécimal

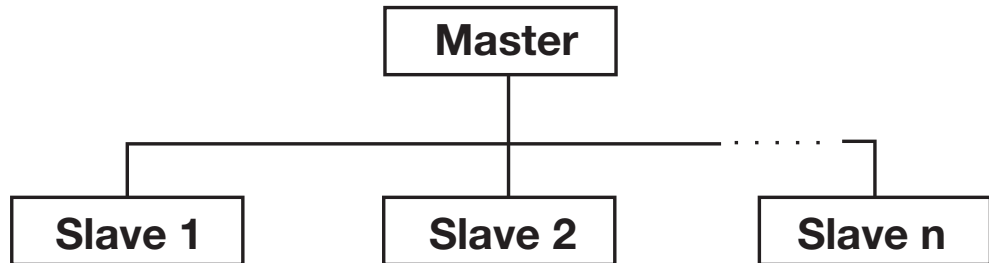
0x0010

Un nombre hexadécimal est précédé de "0x" (ici 16 décimales).

1 Introduction

2.1 Principe maître-esclave

La communication entre un appareil maître (par ex. un PC) et un appareil esclave (par ex. système de mesure et de régulation) avec le protocole ModBus se déroule selon le principe maître/esclave sous la forme demande de données/instruction-réponse.



Le maître contrôle l'échange de données, les esclaves ne donnent que des réponses. Ils sont identifiés par leur adresse d'appareil.

2.2 Mode de transmission (RTU)

Le mode de transmission est le mode RTU (Remote Terminal Unit). La transmission des données s'effectue sous forme binaire (hexadécimale) sur 8 bits. Le bit de poids faible (LSB = least significant bit) est transmis en premier. Le code ASCII n'est pas pris en considération.

Format des données

Le format des données décrit la structure d'un symbole transmis. Les différents formats de données possibles sont les suivants :

Mot de données	Bit de parité	Bit de stop 1/2 bit(s)	Nombre de bits
8 Bit	—	1	9
8 Bit	paire (even)	1	10
8 Bit	impaire (odd)	1	10
8 Bit	—	2	10

2 Description du protocole

2.3 Adresse-appareil

L'adresse appareil de l'esclave est réglable entre 0 et 254.
L'adresse appareil 0 est réservée.



L'interface RS485 permet d'adresser au maximum 31 esclaves.

Il existe deux possibilités d'échange de données :

Query (consultation)

Demande de données / instruction du maître à un esclave au travers d'une adresse d'appareil particulière.

L'esclave adressé répond.

Broadcast (diffusion)

Instruction du maître à tous les esclaves à l'aide de l'adresse d'appareil 0 (par ex. pour la transmission d'une valeur définie à tous les esclaves).

Les esclaves connectés ne répondent pas. La réception correcte de la valeur par les esclaves devra être contrôlée par une lecture ultérieure de la consigne.

Une demande de données avec l'adresse d'appareil 0 n'est pas logique.

2.4 Déroulement temporel de la transmission

Le début et la fin d'un bloc de données sont caractérisés par des pauses de transmission. Entre deux caractères consécutifs, il doit s'écouler au maximum trois fois le temps de transfert d'un caractère.

Le temps de transfert d'un caractère dépend du débit en baud et du format de données utilisé (nombre de bits de stop et parité).

Pour le format de données 8 bits, sans bit de parité et avec un bit de stop, le temps de transfert d'un caractère est égal à :

Temps de transfert d'un caractère [ms] = 1000 * 9 bits / débit en baud

Pour les autres formats de données :

**Temps de transfert d'un caractère [ms] =
1000 * (8 bits + bit de parité + bit(s) de stop) / débit en baud**

2 Description du protocole

Déroulement

Demande de données du maître Temps de transfert = $n \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$
Identificateur de fin de demande de données $3 \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$
Traitement de la demande de données par l'esclave ($\leq 250 \text{ ms}$)
Réponse de l'esclave Temps de transfert = $n \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$
Identificateur de fin de réponse $3 \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$

Exemple

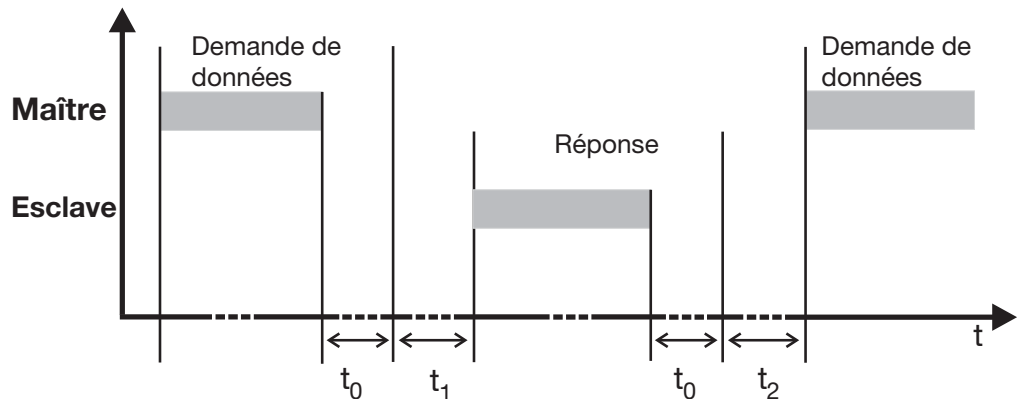
Identificateurs de fin de demande de données et de réponse pour le format 10/9 Bits

Temps d'attente = $3 \text{ caractères} * 1000 * 10 \text{ bits} / \text{débit en baud}$

Débit [bauds]	Format de données [bit]	Temps d'attente [ms] (3 caractères)
38400	10	0.79
	9	0.71
19200	10	1.57
	9	1.41
9600	10	3.13
	9	2.82

2 Description du protocole

Chronogramme Une demande de données se déroule selon le chronogramme suivant :



t_0 Identificateur de fin = 3 caractères.

La durée dépend du débit en baud.

t_1 Cette durée dépend du traitement interne.

La durée maximale de traitement est de 250 ms.



Dans l'appareil, sous le point du menu "Interface", il faut régler le temps de réponse minimal. Il faut laisser s'écouler cette durée avant d'envoyer une réponse (0 à 500 ms). Si on règle une valeur faible, le temps de réponse peut être supérieur à la valeur réglée (le traitement interne dure plus longtemps), l'appareil répond dès que le traitement interne est terminé. Si on règle la valeur sur 0 ms, l'appareil répondra le plus rapidement possible.

Avec une interface RS485, le maître a besoin du temps minimal de réponse (réglable) pour commuter l'interface d'émission en réception.

t_2 L'esclave a besoin de ce temps pour recommuter de l'émission en réception. Le maître laisse s'écouler ce temps avant de poser une nouvelle demande de données. Ce temps doit toujours être respecté, même si la nouvelle demande de données est envoyée à un autre appareil.

Interface RS485 : $t_2 = 10\text{ms}$

Le maître ne peut émettre une demande de données, ni à l'intérieur de t_1 et t_2 , ni pendant le temps de réponse. Les demandes pendant t_1 et t_2 sont ignorées par l'esclave. Les demandes pendant le temps de réponse ont pour conséquence que toutes les données qui se trouvent à ce moment sur le bus deviennent invalides.

2 Description du protocole

2.5 Structure des blocs de données

Tous les blocs de données ont la même structure :

Structure des données

Adresse de l'esclave	Code de la fonction	Données	Somme de contrôle CRC16
1 octet	1 octet	x octet	2 octets

Chaque bloc de données dispose de 4 champs :

Adresse de l'esclave	Adresse d'appareil d'un certain esclave
Code de la fonction	Choix de la fonction (lecture, écriture de mots)
Données	Contient les informations : <ul style="list-style-type: none">- adresse des mots- nombre de mots- valeur des mots
Somme de contrôle	Détection des erreurs de transmission

2 Description du protocole

2.6 Code des fonctions

Les fonctions suivantes décrites sont disponibles pour la lecture de valeurs de mesure, de données de process et des appareils ainsi que pour l'écriture de données définies.

Aperçu des fonctions

Numéro	Fonction	Limitation
0x03 ou 0x04	Lecture de n mots	max. 32 mots (64 octets)
0x06	Ecriture d'un mot	max. 1 mot (2 octets)
0x10	Ecriture de n mots	max. 32 mots (64 octets)



Lorsque l'appareil ne réagit pas à ces fonctions ou émet un code d'erreur, voir chapitre 2.9 Traitement des erreurs, page 18.

2.6.1 Lecture de n mots

Avec cette fonction, n mots ($n \leq 32$) peuvent être lus à partir d'une adresse définie

Demande de données

Adresse de l'esclave	Fonction 0x03 ou 0x04	Adresse 1er mot	Nombre de mots (max. 32)	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Réponse

Adresse de l'esclave	Fonction 0x03 ou 0x04	Nombre d'octets lus	Valeur mot(s)	Somme de contrôle CRC16
1 octet	1 octet	1 octet	x octet(s)	2 octets

Exemple

Lecture des consignes W1 et W2 (respectivement 2 mots)

Adresse du premier mot = 0x3100 (consigne W1)

Demande de données :

01	03	3100	0004	4AF5
----	----	------	------	------

Réponse (valeurs en format Modbus flottant):

01	03	08	0000	41C8	0000	4120	4A9E
			consigne W1 (25.0)		consigne W2 (10.0)		

2 Description du protocole

2.6.2 Ecriture d'un mot

Pour cette fonction, les blocs de données de l'ordre sont identiques aux blocs de données de la réponse.

Ordre

Adresse de l'esclave	Fonction 0x06	Adresse mot	Valeur mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octet

Réponse

Adresse de l'esclave	Fonction 0x06	Adresse mot	Valeur mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple

Ecriture de la valeur limite AL seuil d'alarme 1 = 275.0
(valeur = 0x80004389 en format Modbus flottant)

Adresse mot = 0x0057

Ordre : écriture de la première partie de la valeur

01	06	0057	8000	59DA
----	----	------	------	------

Réponse (idem ordre) :

01	06	0057	8000	59DA
----	----	------	------	------

Ordre : écriture de la seconde partie de la valeur (prochaine adresse mot)

01	06	0058	4389	F88F
----	----	------	------	------

Réponse (idem ordre) :

01	06	0058	4389	F88F
----	----	------	------	------

2 Description du protocole

2.6.3 Ecriture de n mots

Avec cette fonction, n mots ($n \leq 32$) peuvent être lus à partir d'une adresse définie.

Ordre

Adresse de l'esclave	Fonction 0x10	Adresse 1er mot	Nombre de mot (max. 32)	Nombre d'octets	Valeur(s) mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	1 octet	x octet(s)	2 octets

Réponse

Adresse de l'esclave	Fonction 0x10	Adresse 1er mot	Nombre de mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple

Ecriture des consignes W1 et W2

Adresse mot = 0x3100 (consigne W1)

Ordre :

01	10	3100	0004	08	0000	41C8	0000	4120	2A42
					Consigne W1 (25.0)		Consigne W2 (10.0)		

Réponse:

01	10	3100	0004	CF36
----	----	------	------	------

2 Description du protocole

2.7 Format de transmission (valeurs entières, flottantes et texte)

Valeurs entières Avec le protocole Modbus, les valeurs entières sont transmises sous la forme suivante : d'abord l'octet de poids fort, ensuite l'octet de poids faible.

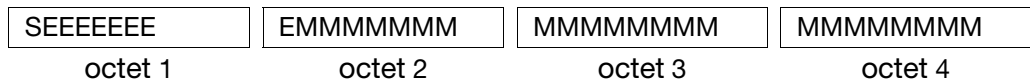
Exemple Consultation de la valeur entière à l'adresse 0x0021 lorsque le contenu à cette adresse est "4" (valeur du mot 0x0004).

Demande : 01 03 0021 0001 (+ 2 octets CRC16)

Réponse : 01 03 02 **0004** (+ 2 octets CRC16)

Valeurs flottantes Le protocole Modbus traite les valeurs flottantes conformément au format standard IEEE 754 (32 bits) ; toutefois il y a une différence : les octets 1 et 2 sont échangés avec les octets 3 et 4.

Format des valeurs flottantes simples (32 bits) suivant le standard IEEE 754

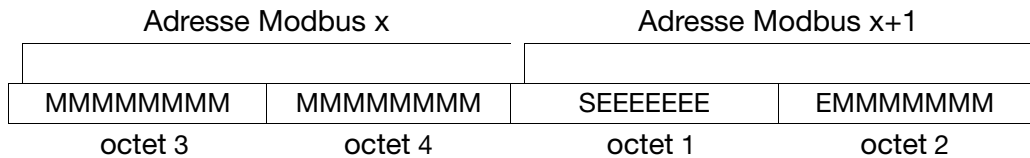


S - Bit de signe

E - Exposant (complément à 2)

M - Mantisse normalisée sur 23 bits

Format flottant Modbus



Exemple Consultation de la valeur flottante à l'adresse 0x0035 lorsque le contenu à cette adresse est 550.0 (0x44098000 au format IEEE 754).

Demande : 01 03 0035 0002 (+ 2 octets CRC16)

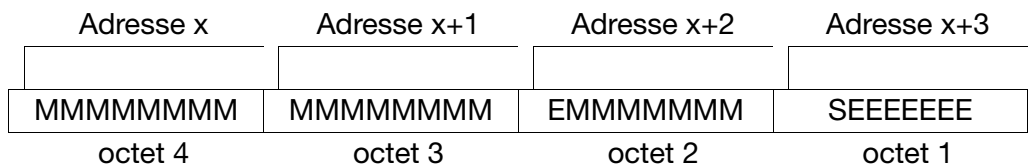
Réponse : 01 03 04 **8000 4409** (+ 2 octets CRC16)

Après le transfert depuis l'appareil, il faut échanger les octets de cette valeur.



De nombreux compilateurs (par ex. Microsoft Visual C++) manipulent les valeurs flottantes dans l'ordre suivant :

Valeur flottante



Déterminez le mode de stockage des valeurs flottantes dans votre application. Le cas échéant, après la consultation de l'enregistreur sans papier, il faudra échanger les octets dans votre programme d'interface.

2 Description du protocole

Chaînes de caractères (textes)

Les chaînes de caractères (textes) sont transmises en format ASCII.



Le dernier caractère (indicateur de fin) doit toujours être un "\0" (code ASCII 0x00). Les caractères qui suivent n'ont aucune importance.

Le nombre de caractères max. dans le type de données est indiqué dans les tableaux d'adresse, par ex. "TEXT24" (24 caractères). Si l'on utilise aucun indicateur de fin, il ne reste dans ce cas de figure que 23 caractères lisibles pour le texte.

Si l'on utilise aucun indicateur de fin, il faut, pour l'écriture que le nombre de caractères max. indiqué dans le type de données (par ex. TEXT8 = 8 caractères) soit utilisé. On évite ainsi que des caractères encore en mémoire soient annexés au texte.

Comme la transmission des textes a lieu également mot par mot (16 bits), il faut envoyer un 0x00 supplémentaire si le nombre de caractères est impair (y compris le caractère "\0").

Exemple pour type de données TEXT4

Lecture du texte (ici : "AbC ") sous l'adresse 0x0067 (4 caractères max. peuvent être mémorisés)

Code ASCII pour "AbC " (avec 1 espace à la fin) :
0x41, 0x62, 0x43, 0x20

Demande : 01 03 0067 0002 (+ 2 octets CRC16)

Adresse esclave = 01

Fonction = 03, c.-à-d. lecture de n mots

Adresse = 0067

Nombre de mots à lire = 0002, 4 caractères max.

Réponse : 01 03 04 **41 62 43 20** (+ 2 octets CRC16)

Adresse esclave = 01

Fonction = 03, c.-à-d. lecture de n mots

Nombre d'octets lus = 04

Variante :

Code ASCII pour "Ab" (sans espace à la fin) :

0x41, 0x62, 0x00

ASCII 0x00 ("\0") signifie que la chaîne de caractères s'arrête ici.

Lors de la transmission un **0x00** supplémentaire est annexé, pour obtenir un nombre pair de caractères.

Réponse dans ce cas : 01 03 04 **41 62 00 00** (+ 2 octets CRC16)

2 Description du protocole

2.8 Somme de contrôle (CRC16)

La somme de contrôle (CRC16) permet de détecter les erreurs de transmission. Si une erreur est détectée lors de l'analyse, l'appareil correspondant ne répond pas.

Mode de calcul

CRC = 0xFFFF	
CRC = CRC XOR ByteOfMessage	
For (1 à 8)	
CRC = SHR(CRC)	
if (drapeau report à droite = 1)	
then	else
CRC = CRC XOR 0xA001	
while (tous les octets du message ne sont pas traités);	



L'octet faible de la somme de contrôle est transféré le premier, suivi de l'octet fort.

Exemple

Demande de données : lecture de 2 mots à partir de l'adresse 0x00CE
(CRC16 = 0x92A5)

07	03	00	CE	00	02	A5	92
							CRC16

Réponse : (CRC16 = 0xF5AD)

07	03	04	00	00	41	C8	AD	F5
				mot 1	mot 2		CRC16	

2 Description du protocole

2.9 Traitement des erreurs

Code d'erreur Codes d'erreur possibles :

- 1 fonction invalide
- 2 Adresse de paramètres invalide ou nombre de mots ou de bits à lire ou à écrire trop élevé
- 8 Paramètre protégé en écriture

Réponse en cas d'erreur

Adresse esclave	Fonction XX OR 80h	Code d'erreur	Somme de contrôle CRC16
1 octet	1 octet	1 octet	2 octets

Le code de la fonction est associé à 0x80 à l'aide d'une fonction OU (OR), c'est-à-dire que le bit de poids fort (MSB = Most Significant Bit) est mis à 1:

Exemple

Demande de données :

01	03	40	00	00	04	CRC16
----	----	----	----	----	----	-------

Réponse (avec code d'erreur 2) :

01	83	02	CRC16
----	----	----	-------

Cas particuliers

Dans les cas suivants, l'esclave ne répond pas :

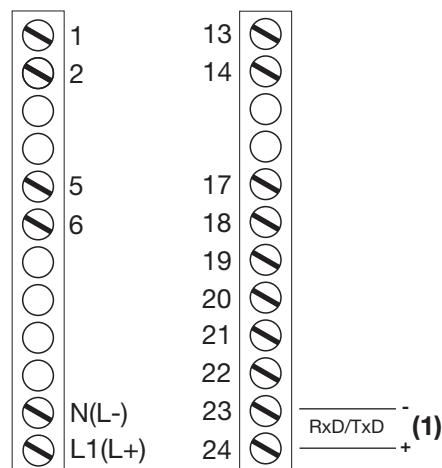
- débit en baud et/ou format de données ne concordent pas pour le maître et pour l'esclave
- L'adresse de l'appareil utilisée ne concorde pas avec l'adresse de l'esclave
- La somme de contrôle (CRC16) est incorrecte.
- L'ordre du maître est incomplet ou contradictoire
- Le nombre de mots à lire est égal à 0

Dans ces cas, la demande de données devrait après écoulement du temps du timeout (2s) être renvoyée.

3.1 Schéma de raccordement



Cet appareil peut être commandé avec une interface RS 485, en option. Vous trouverez des références de commande dans la notice de mise en service et de montage INSTALL-150 (identification du type).



(1) Interface RS485



Le blindage du câble de l'interface doit être mis unilatéralement à la terre dans l'armoire de commande.

3 Interface RS485

3.2 Configuration

Les différents réglages de l'interface Modbus qui sont effectués au niveau de configuration (*CONF* → *INTF*) et/ou dans le logiciel Setup sont décrits dans le tableau ci-dessous.



Vous trouverez des informations complémentaires dans la notice de mise en service et de montage INSTALL-150.

	Symbole	Valeur / Sélection	Description
Débit en baud	<i>bdr</i>	0	9600 Baud
		1	19200 Baud
		2	38400 Baud
Format des données	<i>dft</i>	0	8 bits de données, 1 bit stop, pas de parité
		1	8 bits de données, 1 bit stop, parité impaire
		2	8 bits de données, 1 bit stop, parité paire
Adresse app.	<i>Adr</i>	0 à 1 à 255	Adresse dans le réseau de données Les adresses 0 et 255 sont prévues à des fins définies et ne devraient être utilisées ici.
Temps de réponse min.	(Setup)	0 à 500ms	Laps de temps, entre le moment de la demande d'un appareil dans un réseau de données et la réponse du régulateur (réglable uniquement via le logiciel Setup).

Les réglages d'usine sont en **gras**.



L'interface RS485 est inactive si communication via l'interface Setup.

4 Adresses Modbus

Type de données, type d'accès

Vous trouverez dans le tableau ci-dessous toutes les valeurs de process (variables) avec leurs adresses, le type de données ainsi que le mode d'accès.

Signification :

R/O	Lecture seule
W/O	Ecriture seule
R/W	Ecriture et lecture
INT	Integer (8 ou 16 Bit)
Bit x	Bit No x (Bit 0 est le bit le plus faible)
LONG	Long Integer (4 octets)
FLOAT	Valeur flottante (4 octets) suivant IEEE 754
TEXT4	Texte à 4 caractères



Les opérations d'écriture sur les paramètres R/W provoquent une sauvegarde dans l'EEPROM. Ces modules mémoires n'ont qu'un nombre limité de cycles d'écriture (env. 100 000), c'est pourquoi cette fonction peut être déconnectée en cas de programmation fréquente. Les valeurs de paramètres sont alors seulement enregistrées dans la mémoire volatile (RAM) et sont perdues en cas de panne de secteur.

⇒ *Setup/Uniquement Setup/Paramètre non documenté/Paramètre Bit/Activer paramètre 2*

4.1 Données de process

Adresses	Type de données / numéro de bit	Accès	Désignation du signal
0x001F	INT	R/O	Valeurs binaires internes
	Bit 12		Valeur binaire L1 (= 0x1000)
	Bit 13		Valeur binaire L2 (= 0x2000)
0x0020	INT	R/O	Etat du régulateur
	Bit 12		Mode manuel actif (= 0x1000)
	Bit 15		Auto-optimisation active (= 0x8000)
0x0021	INT	R/O	Sorties binaires 1 à 4 (états de commutation 0 = OFF / 1 = ON)
	Bit 0		Sortie K1: relais (= 0x0001)
	Bit 1		Sortie K2: relais (= 0x0002)
	Bit 2		Sortie K3: logique (= 0x0004)
	Bit 3		Sortie K4: relais (= 0x0008)

4 Adresses Modbus

Adresses	Type de données / numéro de bit	Accès	Désignation du signal
0x0023	INT	R/O	Entrées binaires 1 et 2 (états de commutation 0 = ouverte / 1 = fermée)
	Bit 0		Entrée 1 (= 0x0001)
	Bit 1		Entrée 2 (= 0x0002)
0x0024	INT	R/O	Seuils d'alarme 1 à 2
	Bit 0		Seuil d'alarme 1 (= 0x0001)
	Bit 1		Seuil d'alarme 2 (= 0x0002)
0x0025	INT	R/W	Commande Sorties binaires (séparément)
	Bit 0 + Bit 8		Sortie K1 (= 0x0101)
	Bit 1 + Bit 9		Sortie K2 (= 0x0202)
	Bit 2 + Bit 10		Sortie K3 (= 0x0404)
	Bit 3 + Bit 11		Sortie K4 (= 0x0808)
0x0026	FLOAT	R/O	Entrée analogique [mV]
0x0028	FLOAT	R/O	Pt100 Interne [Ohm]
0x002A	INT	R/O	Cadence de scrutation
0x002B	FLOAT	R/O	Entrée analogique [valeur d'affichage]
0x002D	FLOAT	R/O	Valeur analogique interne 1
0x002F	FLOAT	R/O	Valeur analogique interne 2
0x0031	FLOAT	R/O	Régulateur Valeur fin de rampe
0x0033	FLOAT	R/O	Régulateur Valeur réelle FILTEREE
0x0035	FLOAT	R/O	Régulateur Valeur réelle NON FILTEREE
0x0037	FLOAT	R/W	Régulateur Consigne
0x0039	FLOAT	R/O	Régulateur Affichage du taux de modulation
0x003B	FLOAT	R/O	Régulateur Taux de modulation CHAUFFER
0x003D	FLOAT	R/O	Régulateur Taux de modulation REFROIDIR
0x003F	FLOAT	R/O	Régulateur Différence
0x0041	FLOAT	R/O	Régulateur Ecart de régulation
0x0043	INT	R/O	Régulateur Position du point de contact CHAUFFER
0x0044	INT	R/O	Régulateur Position du point de contact REFROIDIR
0x0046	INT	R/O	Taux de modulation manuel
0x0047	LONG	R/O	Temps d'exécution de la minuterie
0x0049	LONG	R/O	Temps restant de la minuterie
0x004B	INT	R/O	Etat de la minuterie
	Bit 1		Minuterie arrêtée (= 0x0002)
	Bit 5		Minuterie fonctionne (= 0x0020)
	Bit 6		Fin minuterie (= 0x0040)
	Bit 15		Signal de la minuterie (= 0x8000)

4.2 Consignes

Adresses	Type de données / numéro de bit	Accès	Désignation du signal
0x3100	FLOAT	R/W	Consigne W1
0x3102	FLOAT	R/W	Consigne W2

4.3 Paramètres du régulateur

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x3000	FLOAT	R/W	Paramètre du régulateur XP1
0x3002	FLOAT	R/W	Paramètre du régulateur XP2
0x3004	FLOAT	R/W	Paramètre du régulateur TV
0x3006	FLOAT	R/W	Paramètre du régulateur TN
0x300C	FLOAT	R/W	Paramètre du régulateur CY1
0x300E	FLOAT	R/W	Paramètre du régulateur CY2
0x3010	FLOAT	R/W	Paramètre du régulateur XSH
0x3012	FLOAT	R/W	Paramètre du régulateur XD1
0x3014	FLOAT	R/W	Paramètre du régulateur XD2
0x3016	INT	R/W	Paramètre du régulateur TT
0x3017	INT	R/W	Paramètre du régulateur Y0
0x3018	INT	R/W	Paramètre du régulateur Y1
0x3019	INT	R/W	Paramètre du régulateur Y2

4.4 Configuration

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x0053	FLOAT	R/W	Fonction rampe Pente
0x0055	FLOAT	R/W	Constante de temps du filtre
0x0057	FLOAT	R/W	Seuil d'alarme 1 Valeur limite AL
0x0059	FLOAT	R/W	Seuil d'alarme 1 Différentiel de coupure
0x005D	FLOAT	R/W	Seuil d'alarme 2 Valeur limite AL
0x005F	FLOAT	R/W	Seuil d'alarme 2 Différentiel de coupure
0x0063	LONG	R/W	Valeur de la minuterie
0x0065	LONG	R/W	Valeur limite Intervention
0x0067	TEXT4	R/W	Texte d'alarme
0x0069	LONG	R/W	Compteur d'intervention

4 Adresses Modbus

4.5 Commandes

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x004D	INT	W/O	Fonctions binaires REGULATEUR
	Bit 0		Démarrage de l'auto-optimisation (= 0x0001)
	Bit 1		Annulation de l'auto-optimisation (= 0x0002)
	Bit 2		Mode manuel (= 0x0004)
	Bit 3		Mode automatique (= 0x0008)
	Bit 4		Régulateur OFF (= 0x0010)
	Bit 5		Verrouillage du mode manuel (= 0x0020)
	Bit 6		Stop Rampe (= 0x0040)
	Bit 7		Annulation Rampe (= 0x0080)
	Bit 8		Redémarrage Rampe (= 0x0100)
	Bit 9		Démarrage de la minuterie (= 0x0200)
	Bit 10		Annulation de la minuterie (= 0x0400)
	Bit 11		Arrêt de la minuterie (= 0x0800)
0x004E	INT	W/O	Fonctions binaires COMMANDE
	Bit 0		Verrouillage du clavier (= 0x0001)
	Bit 1		Verrouillage des niveaux de configuration et de paramétrage (= 0x0002)
	Bit 3		Affichage OFF (= 0x0008)
	Bit 5		Affichage textes (= 0x0020)
0x004F	INT	W/O	Fonctions binaires MINUTERIE
	Bit 9		Démarrer minuterie (= 0x0200)
	Bit 10		Annuler minuterie (= 0x0400)
	Bit 11		Arrêter minuterie (= 0x0800)
0x0050	INT	R/W	Commutation de la consigne
	Bit 0		Consigne 1 (= 0x0001)
	Bit 1		Consigne 2 (= 0x0002)

4.6 Mémoire RAM

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x3200	FLOAT	W/O	Consigne du régulateur (lecture/écriture)
0x3202	FLOAT	W/O	Valeur réelle du régulateur (lecture/écriture)
0x3204	FLOAT	W/O	Valeur analogique interne 1 (lecture/écriture)
0x3206	FLOAT	W/O	Valeur analogique interne 2 (lecture/écriture)
0x3208	INT	W/O	Valeurs binaires internes (lecture/écriture)
	Bit 0 + Bit 7		Valeur binaire L1 (= 0x0081)
	Bit 1 + Bit 7		Valeur binaire L2 (= 0x0082)



L'accès direct à la mémoire RAM de l'appareil est possible via le Modbus, afin d'écrire la valeur de consigne du régulateur (0x3200), la valeur réelle du régulateur (0x3202) et les valeurs analogiques internes (0x3204, 0x3206) ainsi que les valeurs binaires internes (0x3208).

Une plage comprise entre 1999 et +9999 est disponible pour l'écriture de la valeur du régulateur, la valeur réelle du régulateur et les valeurs analogiques internes. Est alors utilisée la valeur écrite dans l'appareil au lieu de la valeur originale.

Si la valeur originale doit à nouveau être utilisée, il faut écrire la valeur 200001 via le Modbus dans les emplacements mémoires respectifs.

4 Adresses Modbus

DigiTrace TCONTROL-CONT-03

Kompakter Mikroprozessorregler



INSTALL-160 Rev. 1

Modbus-Anleitung

2013-05-21

Inhalt

1	Einleitung	5
1.1	Vorwort	5
1.2	Typografische Konventionen	5
2	Protokollbeschreibung	7
2.1	Master-Slave-Prinzip	7
2.2	Übertragungsmodus (RTU)	7
2.3	Geräteadresse	8
2.4	Zeitlicher Ablauf der Kommunikation	8
2.5	Aufbau der Datenblöcke	11
2.6	Funktionscodes	12
2.6.1	Lesen von n Worten	12
2.6.2	Schreiben eines Wortes	13
2.6.3	Schreiben von n Worten	14
2.7	Übertragungsformat (Integer-, Float- und Text-Werte)	15
2.8	Checksumme (CRC16)	17
2.9	Fehlerbehandlung	18
3	RS485-Schnittstelle	19
3.1	Anschlussplan	19
3.2	Konfiguration	20
4	Modbus-Adressen	21
4.1	Prozessdaten	21
4.2	Sollwerte	23
4.3	Reglerparameter	23
4.4	Konfiguration	23
4.5	Kommandos	24
4.6	RAM-Speicher	25

1.1 Vorwort

Diese Anleitung wendet sich an den Anlagenhersteller mit fachbezogener Ausbildung und PC-Kenntnissen.

Lesen Sie diese Anleitung, bevor Sie mit Ihrer Arbeit am Gerät beginnen. Bewahren Sie die Anleitung an einem für alle Benutzer jederzeit zugänglichen Platz auf. Mit Ihren Anregungen können Sie uns helfen, diese Anleitung zu verbessern.

Alle erforderlichen Einstellungen sind in der vorliegenden Anleitung beschrieben. Sollten bei der Inbetriebnahme Schwierigkeiten auftreten, bitten wir Sie, keine eigenmächtigen Manipulationen vorzunehmen, die nicht in der Anleitung beschrieben sind. Sie gefährden dadurch Ihren Gewährleistungsanspruch. Bitte setzen Sie sich mit der nächsten Niederlassung oder dem Stammhaus in Verbindung.

1.2 Typografische Konventionen

Warnende Zeichen:

Achtung



Dieses Zeichen wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu **Beschädigungen von Geräten oder Daten** kommen kann!

Hinweisende Zeichen:

Hinweis



Dieses Zeichen wird benutzt, wenn Sie auf **etwas Besonderes** aufmerksam gemacht werden sollen.

Verweis



Dieses Zeichen weist auf **weitere Informationen** in anderen Handbüchern, Kapiteln oder Abschnitten hin.

Darstellungsarten:

Hexadezimalzahl

0x0010

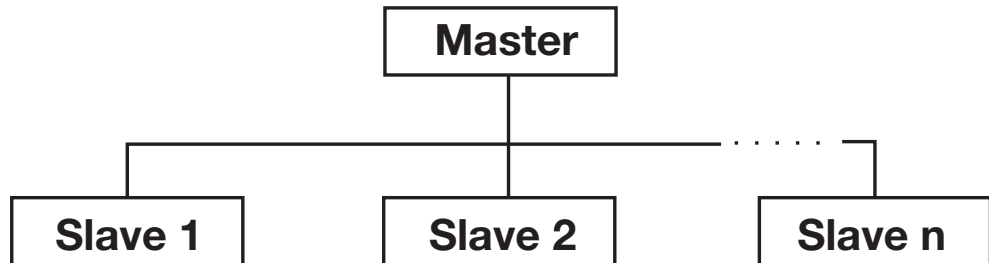
Eine Hexadezimalzahl wird durch ein vorangestelltes „0x“ gekennzeichnet (hier: 16 dezimal).

1 Einleitung

2 Protokollbeschreibung

2.1 Master-Slave-Prinzip

Die Kommunikation zwischen einem Master (z. B. PC) und einem Slave (z. B. Mess- und Regelsystem) mit Modbus findet nach dem Master-Slave-Prinzip in Form von Datenanfrage/Anweisung - Antwort statt.



Der Master steuert den Datenaustausch, die Slaves haben lediglich Antwortfunktion. Sie werden anhand ihrer Geräteadresse identifiziert.

2.2 Übertragungsmodus (RTU)

Als Übertragungsmodus wird der RTU-Modus (Remote Terminal Unit) verwendet. Die Übertragung der Daten erfolgt im Binärformat (hexadezimal) mit 8 Bits. Das LSB (least significant bit, engl. das niederwertigste Bit) wird zuerst übertragen. Die Betriebsart ASCII-Modus wird nicht unterstützt.

Datenformat

Mit dem Datenformat wird der Aufbau eines übertragenen Zeichens beschrieben. Es sind folgende Möglichkeiten des Datenformats gegeben:

Datenwort	Paritätsbit	Stoppbit 1/2 Bit	Bitanzahl
8 Bit	—	1	9
8 Bit	gerade (even)	1	10
8 Bit	ungerade (odd)	1	10
8 Bit	—	2	10

2 Protokollbeschreibung

2.3 Geräteadresse

Die Geräteadresse des Slaves ist zwischen 0 und 254 einstellbar. Die Geräteadresse 0 ist reserviert.



Über die RS485-Schnittstelle können maximal 31 Slaves angesprochen werden.

Es gibt zwei Varianten des Datenaustausches:

Query

Datenanfrage/Anweisung des Masters an einen Slave über die entsprechende Geräteadresse.

Der angesprochene Slave antwortet.

Broadcast

Anweisung des Masters an alle Slaves über die Geräteadresse 0 (z. B. zur Übertragung eines bestimmten Wertes an alle Slaves).

Die angeschlossenen Slaves antworten nicht. Die richtige Übernahme des Wertes durch die Slaves sollte in diesem Fall durch anschließendes Auslesen an jedem einzelnen Slave kontrolliert werden.

Eine Datenanfrage mit der Geräteadresse 0 ist nicht sinnvoll.

2.4 Zeitlicher Ablauf der Kommunikation

Anfang und Ende eines Datenblocks sind durch Übertragungspausen gekennzeichnet. Zwischen zwei aufeinanderfolgenden Zeichen darf maximal das Dreifache der Zeit zum Übertragen eines Zeichens vergehen.

Die Zeichenübertragungszeit (Zeit für die Übertragung eines Zeichens) ist abhängig von der Baudrate und dem verwendeten Datenformat (Stoppbits und Paritätsbit).

Bei einem Datenformat von 8 Datenbits, keinem Paritätsbit und einem Stoppbit ergibt sich:

$$\text{Zeichenübertragungszeit [ms]} = 1000 * 9 \text{ Bit/Baudrate}$$

Bei den anderen Datenformaten ergibt sich:

$$\begin{aligned} \text{Zeichenübertragungszeit [ms]} \\ = 1000 * (8 \text{ Bits} + \text{Paritätsbit} + \text{Stoppbit(s)}) \text{ Bit/Baudrate} \end{aligned}$$

2 Protokollbeschreibung

Ablauf

Datenanfrage vom Master Übertragungszeit = $n \text{ Zeichen} * 1000 * x \text{ Bit/Baudrate}$
Kennzeichen für Datenanfrage-Ende $3 \text{ Zeichen} * 1000 * x \text{ Bit/Baudrate}$
Bearbeitung der Datenanfrage durch den Slave ($\leq 250\text{ms}$)
Antwort des Slaves Übertragungszeit = $n \text{ Zeichen} * 1000 * x \text{ Bit/Baudrate}$
Kennzeichen für Antwort-Ende $3 \text{ Zeichen} * 1000 * x \text{ Bit/Baudrate}$

Beispiel

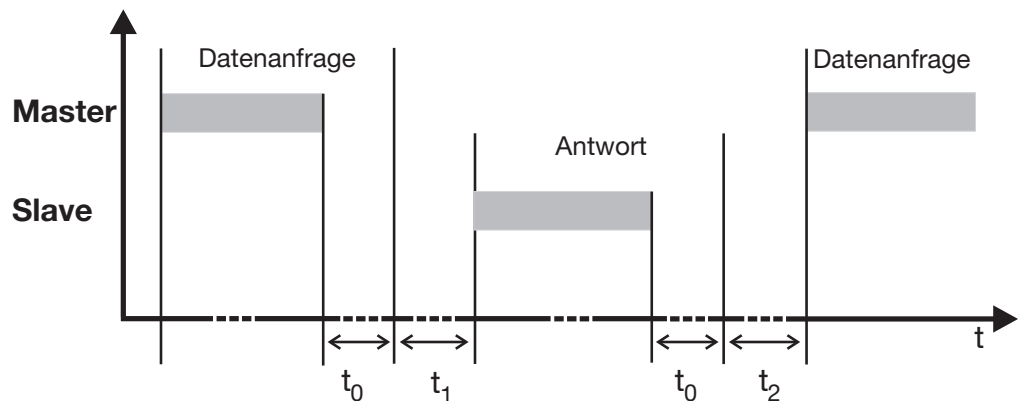
Kennzeichen für Datenanfrage- oder Antwort-Ende bei Datenformat 10/9 Bits
Wartezeit = $3 \text{ Zeichen} * 1000 * 10 \text{ Bit/Baudrate}$

Baudrate [Baud]	Datenformat [Bit]	Wartezeit [ms] (3 Zeichen)
38400	10	0.79
	9	0.71
19200	10	1.57
	9	1.41
9600	10	3.13
	9	2.82

2 Protokollbeschreibung

Zeitschema

Eine Datenanfrage läuft nach folgendem Zeitschema ab:



- t_0 Endekennzeichen = 3 Zeichen
(die Zeit ist von der Baudrate abhängig)
- t_1 Diese Zeit ist von der internen Bearbeitungszeit abhängig.
Die maximale Bearbeitungszeit liegt bei 250 ms.



In dem Gerät kann unter dem Menüpunkt „Schnittstelle“ eine minimale Antwortzeit eingestellt werden. Diese eingestellte Zeit wird mindestens eingehalten, bevor eine Antwort gesendet wird (0...500 ms). Wird ein kleiner Wert eingestellt, so kann die Antwortzeit größer sein als der eingestellte Wert (die interne Bearbeitungszeit ist länger), das Gerät antwortet dann unmittelbar nachdem die interne Bearbeitung abgeschlossen ist. Eine eingestellte Zeit von 0 ms bedeutet, dass das Gerät mit der maximal möglichen Geschwindigkeit antwortet.

Die minimal einstellbare Antwortzeit wird bei der RS485-Schnittstelle vom Master benötigt, um die Schnittstellentreiber von Senden auf Empfangen umzustellen.

- t_2 Diese Zeit braucht der Slave, um von Senden wieder auf Empfangen umzuschalten. Diese Zeit muß der Master einhalten, bevor er eine neue Datenanfrage stellt. Sie muß immer eingehalten werden, auch wenn die neue Datenanfrage an ein anderes Gerät gerichtet ist.

RS485-Schnittstelle: $t_2 = 10\text{ms}$

Innerhalb von t_1 und t_2 und während der Antwortzeit des Slaves dürfen vom Master keine Datenanfragen gestellt werden. Anfragen während t_1 und t_2 werden vom Slave ignoriert. Anfragen während der Antwortzeit führen dazu, dass alle gerade auf dem Bus befindlichen Daten ungültig werden.

2.5 Aufbau der Datenblöcke

Alle Datenblöcke haben die gleiche Struktur:

Datenstruktur

Slave-Adresse	Funktionscode	Datenfeld	Checksumme CRC16
1 Byte	1 Byte	x Byte	2 Bytes

Jeder Datenblock enthält vier Felder:

Slave-Adresse	Geräteadresse eines bestimmten Slaves
Funktionscode	Funktionsauswahl (Lesen, Schreiben von Worten)
Datenfeld	Enthält die Informationen: <ul style="list-style-type: none">- Wortadresse- Wortanzahl- Wortwert(e)
Checksumme	Erkennung von Übertragungsfehlern

2 Protokollbeschreibung

2.6 Funktionscodes

Die nachfolgend beschriebenen Funktionen stehen zum Auslesen von Messwerten, Geräte- und Prozessdaten sowie zum Schreiben von bestimmten Daten zur Verfügung.

Funktionsübersicht

Funktionsnummer	Funktion	Begrenzung
0x03 oder 0x04	Lesen von n Worten	max. 32 Worte (64 Byte)
0x06	Schreiben eines Wortes	max. 1 Wort (2 Byte)
0x10	Schreiben von n Worten	max. 32 Worte (64 Byte)



Wenn das Gerät auf diese Funktionen nicht reagiert oder einen Fehlercode ausgibt, siehe Kapitel 2.9 Fehlerbehandlung, Seite 18.

2.6.1 Lesen von n Worten

Mit dieser Funktion werden n ($n \leq 32$) Worte ab einer bestimmten Adresse gelesen.

Datenanfrage

Slave-Adresse	Funktion x03 oder 0x04	Adresse erstes Wort	Wortanzahl (max. 32)	Checksumme CRC16
1 Byte	1 Byte	2 Byte	2 Byte	2 Byte

Antwort

Slave-Adresse	Funktion 0x03 oder 0x04	Anzahl gelesener Bytes	Wort- wert(e)	Checksumme CRC16
1 Byte	1 Byte	1 Byte	x Byte	2 Byte

Beispiel

Lesen der Sollwerte W1 und W2 (jeweils 2 Worte)

Adresse erstes Wort = 0x3100 (Sollwert W1)

Datenanfrage:

01	03	3100	0004	4AF5
----	----	------	------	------

Antwort (Werte im Modbus-Float-Format):

01	03	08	0000	41C8	0000	4120	4A9E
			Sollwert W1 (25.0)		Sollwert W2 (10.0)		

2 Protokollbeschreibung

2.6.2 Schreiben eines Wortes

Bei der Funktion Wortschreiben sind die Datenblöcke für Anweisung und Antwort identisch.

Anweisung

Slave-Adresse	Funktion 0x06	Wortadresse	Wortwert	Checksumme CRC16
1 Byte	1 Byte	2 Byte	2 Bytes	2 Byte

Antwort

Slave-Adresse	Funktion 0x06	Wortadresse	Wortwert	Checksumme CRC16
1 Byte	1 Byte	2 Byte	2 Byte	2 Byte

Beispiel

Schreiben des Grenzwertes AL von Limitkomparator 1 = 275.0
(Wert = 0x80004389 im Modbus-Float-Format)

Wortadresse = 0x0057

Anweisung: Schreiben des ersten Teils des Wertes

01	06	0057	8000	59DA
----	----	------	------	------

Antwort (wie Anweisung):

01	06	0057	8000	59DA
----	----	------	------	------

Anweisung: Schreiben des zweiten Teils des Wertes (nächste Wortadresse)

01	06	0058	4389	F88F
----	----	------	------	------

Antwort (wie Anweisung):

01	06	0058	4389	F88F
----	----	------	------	------

2 Protokollbeschreibung

2.6.3 Schreiben von n Worten

Mit dieser Funktion werden n ($n \leq 32$) Worte ab einer bestimmten Adresse geschrieben.

Anweisung

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortanzahl (max. 32)	Byteanzahl	Wortwert(e)	Checksumme CRC16
1 Byte	1 Byte	2 Byte	2 Byte	1 Byte	x Byte	2 Byte

Antwort

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortanzahl	Checksumme CRC16
1 Byte	1 Byte	2 Byte	2 Byte	2 Byte

Beispiel

Schreiben der Sollwerte W1 und W2 (jeweils 2 Worte)

Wortadresse = 0x3100 (Sollwert W1)

Anweisung:

01	10	3100	0004	08	0000	41C8	0000	4120	2A42
					Sollwert W1 (25.0)		Sollwert W2 (10.0)		

Antwort:

01	10	3100	0004	CF36
----	----	------	------	------

2 Protokollbeschreibung

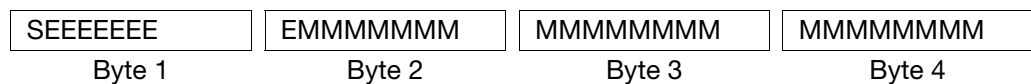
2.7 Übertragungsformat (Integer-, Float- und Text-Werte)

Integer-Werte Integer-Werte werden über Modbus im folgenden Format übertragen:
Zuerst das High-, dann das Low-Byte.

Beispiel Abfrage des Integer-Wertes von Adresse 0x0021, wenn unter dieser Adresse der Wert "4" (Wortwert 0x0004) steht.
Anfrage: 01 03 0021 0001 (+ 2 Byte CRC16)
Antwort: 01 03 02 **0004** (+ 2 Byte CRC16)

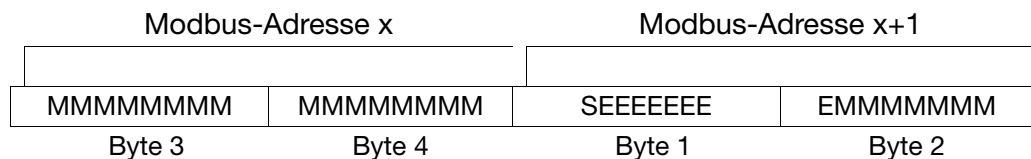
Float-Werte Bei Float-Werten wird im Modbus mit dem IEEE-754-Standard-Format (32bit) gearbeitet, allerdings mit dem Unterschied, dass Byte 1 und 2 mit Byte 3 und 4 vertauscht sind.

Single-float-Format (32bit) nach Standard IEEE 754



S - Vorzeichen-Bit
E - Exponent (2er-Komplement)
M - 23Bit normalisierte Mantisse

Modbus-float-Format

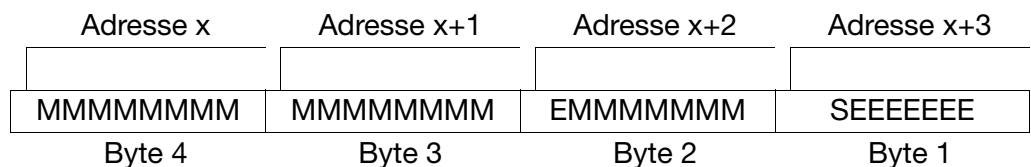



Beispiel Abfrage des Float-Wertes von Adresse 0x0035, wenn unter dieser Adresse der Wert 550.0 (0x44098000 im IEEE-754-Format) steht.
Anfrage: 01 03 0035 0002 (+ 2 Byte CRC16)
Antwort: 01 03 04 **8000 4409** (+ 2 Byte CRC16)

Nach der Übertragung vom Gerät müssen die Bytes des Float-Wertes entsprechend vertauscht werden.

 Viele Compiler (z. B. Microsoft Visual C++) legen die Float-Werte in folgender Reihenfolge ab:

Float-Wert



 Bitte ermitteln Sie, wie in Ihrer Anwendung Float-Werte gespeichert werden. Ggf. müssen die Bytes nach der Abfrage in Ihrem Schnittstellenprogramm entsprechend getauscht werden.

2 Protokollbeschreibung

Zeichenketten (Texte)

Zeichenketten (Texte) werden im ASCII-Format übertragen.



Als letztes Zeichen kann ein "\0" (ASCII-Code 0x00) als Endekennung übertragen werden. Danach folgende Zeichen haben keine Bedeutung.

In den Adresstabellen ist die max. mögliche Zeichenanzahl im Datentyp angegeben, z. B. "TEXT24" (24 Zeichen). Bei Verwendung einer Endekennung stehen bei diesem Beispiel nur noch 23 lesbare Zeichen für den Text zur Verfügung.

Wird keine Endekennung verwendet, muss beim Schreiben die im Datentyp angegebene max. Zeichenanzahl (z. B. TEXT8 = 8 Zeichen) genutzt werden. So wird verhindert, dass noch im Speicher vorhandene Zeichen an den Text angehängt werden.

Da die Übertragung von Texten wortweise (16 Bit) erfolgt, wird bei einer ungeraden Zeichenanzahl (inkl. "\0") zusätzlich 0x00 angehängt.

Beispiel für Datentyp TEXT4

Lesen des Textes (hier: "AbC ") unter Adresse 0x0067 (max. 4 Zeichen können gespeichert werden)

ASCII-Code für "AbC " (mit einem Leerzeichen am Ende):
0x41, 0x62, 0x43, 0x20

Anfrage: 01 03 0067 0002 (+ 2 Byte CRC16)

Slave-Adresse = 01

Funktion = 03, d. h. Lesen von n Worten

Adresse = 0067

Anzahl der zu lesenden Worte = 0002, da max. 4 Zeichen

Antwort: 01 03 04 **41 62 43 20** (+ 2 Byte CRC16)

Slave-Adresse = 01

Funktion = 03, d. h. Lesen von n Worten

Anzahl der gelesenen Bytes = 04

Variante:

ASCII-Code für "Ab" (ohne Leerzeichen am Ende):

0x41, 0x62, 0x00

ASCII 0x00 ("\0") bedeutet, dass die Zeichenkette hier endet.

Bei der Übertragung wird ein zusätzliches **0x00** angehängt, um eine gerade Anzahl von Zeichen zu erhalten.

Antwort in diesem Fall: 01 03 04 **41 62 00 00** (+ 2 Byte CRC16)

2.8 Checksumme (CRC16)

Anhand der Checksumme (CRC16) werden Übertragungsfehler erkannt. Wird bei der Auswertung ein Fehler festgestellt, antwortet das entsprechende Gerät nicht.

Berechnungs- schema

CRC = 0xFFFF	
CRC = CRC XOR ByteOfMessage	
For (1 bis 8)	
CRC = SHR(CRC)	
if (rechts hinausgeschobenes Flag = 1)	
then	else
CRC = CRC XOR	
0xA001	
while (nicht alle ByteOfMessage bearbeitet);	



Das Low-Byte der Checksumme wird zuerst übertragen, dann folgt das High-Byte.

Beispiel

Datenanfrage: Lesen von zwei Worten ab Adresse 0x00CE
(CRC16 = 0x92A5)

07	03	00	CE	00	02	A5	92
						CRC16	

Antwort: (CRC16 = 0xF5AD)

07	03	04	00	00	41	C8	AD	F5
				Wort 1		Wort 2		CRC16

2 Protokollbeschreibung

2.9 Fehlerbehandlung

Fehlercodes Es existieren folgende Fehlercodes:

- 1 ungültige Funktion
- 2 ungültige Parameteradresse oder zu große Anzahl von Worten soll gelesen oder geschrieben werden
- 8 Schreibzugriff auf Parameter verweigert

Antwort im Fehlerfall

Slave-Adresse	Funktion XX OR 80h	Fehlercode	Checksumme CRC16
1 Byte	1 Byte	1 Byte	2 Bytes

Der Funktionscode wird mit 0x80 verODERT, d. h., das MSB (most significant bit, engl. das höchstwertige Bit) wird auf 1 gesetzt.

Beispiel

Datenanfrage:

01	03	40	00	00	04	CRC16
----	----	----	----	----	----	-------

Antwort (mit Fehlercode 2):

01	83	02	CRC16
----	----	----	-------

Sonderfälle

Wenn der Slave nicht antwortet, können folgende Ursachen vorliegen:

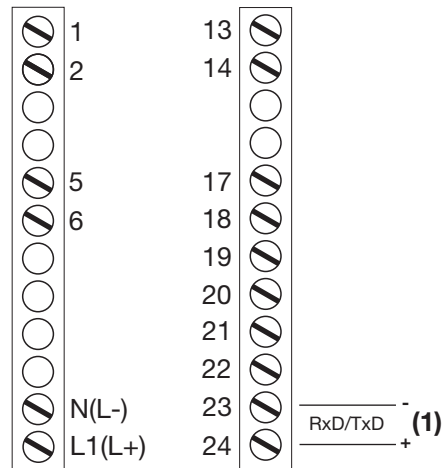
- Baudrate und/oder Datenformat stimmen beim Master und beim Slave nicht überein
- die verwendete Geräteadresse stimmt nicht mit der Slaveadresse überein
- die Checksumme (CRC16) ist nicht korrekt
- die Anweisung des Masters ist unvollständig oder überdefiniert
- die Anzahl der zu lesenden Worte ist Null

In diesen Fällen sollte die Datenanfrage nach Ablauf der Timeout-Zeit (2s) erneut gesendet werden.

3.1 Anschlussplan



Dieses Gerät kann optional mit einer RS485-Schnittstelle bestellt werden. Bestellangaben finden Sie in der in der Betriebs- und Montageanleitung INSTALL-149 (Typenerklärung).



(1) RS485-Schnittstelle



Die Schirmung der Schnittstellenleitung ist einseitig im Schalt-schrank zu erden.

3 RS485-Schnittstelle

3.2 Konfiguration

Die folgende Tabelle zeigt die möglichen Einstellungen der Modbus-Schnittstelle, die in der Konfigurationsebene (CONF → INTF) bzw. im Setup-Programm vorgenommen werden.



Weitere Informationen zur Konfiguration können der Betriebs- und Montageanleitung INSTALL-149 entnommen werden.

	Symbol	Wert/ Auswahl	Beschreibung
Baudrate	<i>bdr</i> t	0	9600 Baud
		1	19200 Baud
		2	38400 Baud
Datenformat	<i>dft</i>	0	8 Datenbits, 1 Stoppbit, keine Parität
		1	8 Datenbits, 1 Stoppbit, ungerade Parität
		2	8 Datenbits, 1 Stoppbit, gerade Parität
		3	8 Datenbits, 2 Stoppbits, keine Parität
Geräteadresse	<i>Adr</i>	0...1...255	Adresse im Datenverbund Die Adressen 0 und 255 sind für bestimmte Zwecke vorgesehen und sollten hier nicht verwendet werden.
Minimale Antwortzeit	(Setup)	0...500ms	Zeitspanne, die von der Anfrage eines Gerätes in einem Datenverbund bis zur Antwort des Reglers mindestens vergeht (nur im Setup-Programm einstellbar).

Werkseitige Einstellungen sind **fett** dargestellt.



Bei Kommunikation über die Setup-Schnittstelle ist die RS485-Schnittstelle inaktiv.

4 Modbus-Adressen

Datentyp, Zugriffsart

In den folgenden Tabellen sind alle Prozess- und Gerätedaten mit ihren Adressen, dem Datentyp und der Zugriffsart aufgeführt.

Hierbei bedeutet:

R/O	Zugriff nur lesend
W/O	Zugriff nur schreibend
R/W	Zugriff lesend und schreibend
INT	Integer (8 oder 16 Bit)
Bit x	Bit Nr. x (Bit 0 ist das niederwertigste Bit)
LONG	Long Integer (4 Byte)
FLOAT	Float-Wert (4 Byte) nach IEEE 754
TEXT4	Text 4 Zeichen



Schreiboperationen auf R/W-Parameter bewirken ein Abspeichern im EEPROM. Diese Speicherbausteine haben nur eine begrenzte Anzahl von Schreibzyklen (ca. 100000), weshalb bei häufiger Programmierung diese Funktion abgeschaltet werden kann. Die Parameterwerte sind dann nur im flüchtigen Speicher (RAM) gespeichert und nach einem Netzausfall verloren.

⇒ *Setup-Programm (Undokumentierte Parameter -> Bitparameter -> Parameter 2 setzen)*

4.1 Prozessdaten

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x001F	INT	R/O	Interne Binärwerte
	Bit 12		Binärwert L1 (= 0x1000)
	Bit 13		Binärwert L2 (= 0x2000)
0x0020	INT	R/O	Reglerstatus
	Bit 12		Handbetrieb aktiv (= 0x1000)
	Bit 15		Selbstoptimierung aktiv (= 0x8000)
0x0021	INT	R/O	Binärausgänge 1...4 (Schaltzustände 0 = aus / 1 = ein)
	Bit 0		Ausgang K1: Relais (= 0x0001)
	Bit 1		Ausgang K2: Relais (= 0x0002)
	Bit 2		Ausgang K3: Logik (= 0x0004)
	Bit 3		Ausgang K4: Relais (= 0x0008)

4 Modbus-Adressen

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x0023	INT	R/O	Binäreingänge 1...2 (Schaltzustände 0 = offen / 1 = geschlossen)
	Bit 0		Eingang 1 (= 0x0001)
	Bit 1		Eingang 2 (= 0x0002)
0x0024	INT	R/O	Limitkomparatoren 1...2
	Bit 0		Limitkomparator 1 (= 0x0001)
	Bit 1		Limitkomparator 2 (= 0x0002)
0x0025	INT	R/W	Ansteuerung Binärausgänge (einzeln)
	Bit 0 + Bit 8		Ausgang K1 (= 0x0101)
	Bit 1 + Bit 9		Ausgang K2 (= 0x0202)
	Bit 2 + Bit 10		Ausgang K3 (= 0x0404)
	Bit 3 + Bit 11		Ausgang K4 (= 0x0808)
0x0026	FLOAT	R/O	Analogeingang [mV]
0x0028	FLOAT	R/O	Interner Pt100 [Ohm]
0x002A	INT	R/O	Abtastzeit
0x002B	FLOAT	R/O	Analogeingang [Anzeigewert]
0x002D	FLOAT	R/O	Interner Analogwert 1
0x002F	FLOAT	R/O	Interner Analogwert 2
0x0031	FLOAT	R/O	Regler Rampenendwert
0x0033	FLOAT	R/O	Regler Istwert GEFILTERT
0x0035	FLOAT	R/O	Regler Istwert UNGEFILTERT
0x0037	FLOAT	R/W	Regler Sollwert
0x0039	FLOAT	R/O	Regler Stellgradanzeige
0x003B	FLOAT	R/O	Regler Stellgrad HEIZEN
0x003D	FLOAT	R/O	Regler Stellgrad KÜHLEN
0x003F	FLOAT	R/O	Regler Regeldifferenz
0x0041	FLOAT	R/O	Regler Regelabweichung
0x0043	INT	R/O	Regler Schaltstellung HEIZEN
0x0044	INT	R/O	Regler Schaltstellung KÜHLEN
0x0046	INT	R/O	Handstellgrad
0x0047	LONG	R/O	Timer-Laufzeit
0x0049	LONG	R/O	Timer-Restzeit
0x004B	INT	R/O	Timer-Status
	Bit 1		Timer angehalten (= 0x0002)
	Bit 5		Timer läuft (= 0x0020)
	Bit 6		Timer-Ende (= 0x0040)
	Bit 15		Timer-Signal (= 0x8000)

4.2 Sollwerte

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x3100	FLOAT	R/W	Sollwert W1
0x3102	FLOAT	R/W	Sollwert W2

4.3 Reglerparameter

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x3000	FLOAT	R/W	Regler-Parameter XP1
0x3002	FLOAT	R/W	Regler-Parameter XP2
0x3004	FLOAT	R/W	Regler-Parameter TV
0x3006	FLOAT	R/W	Regler-Parameter TN
0x300C	FLOAT	R/W	Regler-Parameter CY1
0x300E	FLOAT	R/W	Regler-Parameter CY2
0x3010	FLOAT	R/W	Regler-Parameter XSH
0x3012	FLOAT	R/W	Regler-Parameter XD1
0x3014	FLOAT	R/W	Regler-Parameter XD2
0x3016	INT	R/W	Regler-Parameter TT
0x3017	INT	R/W	Regler-Parameter Y0
0x3018	INT	R/W	Regler-Parameter Y1
0x3019	INT	R/W	Regler-Parameter Y2

4.4 Konfiguration

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x0053	FLOAT	R/W	Rampenfunktion Steigung
0x0055	FLOAT	R/W	Filterzeitkonstante
0x0057	FLOAT	R/W	Limitkomparator 1 Grenzwert AL
0x0059	FLOAT	R/W	Limitkomparator 1 Schaltdifferenz
0x005D	FLOAT	R/W	Limitkomparator 2 Grenzwert AL
0x005F	FLOAT	R/W	Limitkomparator 2 Schaltdifferenz
0x0063	LONG	R/W	Timer-Wert
0x0065	LONG	R/W	Service-Grenzwert
0x0067	TEXT4	R/W	Alarmtext
0x0069	LONG	R/W	Servicezähler

4 Modbus-Adressen

4.5 Kommandos

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x004D	INT	W/O	Binärfunktionen REGLER
	Bit 0		Selbstoptimierung Start (= 0x0001)
	Bit 1		Selbstoptimierung Abbruch (= 0x0002)
	Bit 2		Handbetrieb (= 0x0004)
	Bit 3		Automatikbetrieb (= 0x0008)
	Bit 4		Regler aus (= 0x0010)
	Bit 5		Verriegelung Handbetrieb (= 0x0020)
	Bit 6		Rampe Halt (= 0x0040)
	Bit 7		Rampe Abbruch (= 0x0080)
	Bit 8		Rampe Neustart (= 0x0100)
	Bit 9		Timer Start (= 0x0200)
	Bit 10		Timer Abbruch (= 0x0400)
	Bit 11		Timer Halt (= 0x0800)
0x004E	INT	W/O	Binärfunktionen BEDIENUNG
	Bit 0		Verriegelung Tastatur (= 0x0001)
	Bit 1		Verriegelung Konfigurations- und Parameterebene (= 0x0002)
	Bit 3		Anzeige AUS (= 0x0008)
	Bit 5		Textanzeige (= 0x0020)
0x004F	INT	W/O	Binärfunktionen TIMER
	Bit 9		Timer starten (= 0x0200)
	Bit 10		Timer abrechnen(= 0x0400)
	Bit 11		Timer anhalten (= 0x0800)
0x0050	INT	R/W	Sollwertumschaltung
	Bit 0		Sollwert 1 (= 0x0001)
	Bit 1		Sollwert 2 (= 0x0002)

4.6 RAM-Speicher

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x3200	FLOAT	W/O	Regler-Sollwert (beschreibbar)
0x3202	FLOAT	W/O	Regler-Istwert (beschreibbar)
0x3204	FLOAT	W/O	Interner Analogwert 1 (beschreibbar)
0x3206	FLOAT	W/O	Interner Analogwert 2 (beschreibbar)
0x3208	INT	W/O	Interne Binärwerte (beschreibbar)
	Bit 0 + Bit 7		Binärwert L1 (= 0x0081)
	Bit 1 + Bit 7		Binärwert L2 (= 0x0082)



Über Modbus ist der direkte Zugriff auf den RAM-Speicher des Gerätes möglich, um den Regler-Sollwert (0x3200), den Regler-Istwert (0x3202) und die Internen Analogwerte (0x3204, 0x3206) sowie die Internen Binärwerte (0x3208) zu schreiben.

Beim Schreiben steht für die Werte von Regler-Sollwert, Regler-Istwert und für die Internen Analogwerte ein Bereich von -1999 bis +9999 zur Verfügung. Statt des Originalwertes wird dann der ins Gerät geschriebene Wert verwendet.

Soll am Gerät wieder der Originalwert verwendet werden, muss der Wert 200001 über Modbus in die jeweilige Speicherstelle geschrieben werden.

4 Modbus-Adressen

BELGIË / BELGIQUE

Tel. +32 16 21 35 02
Fax +32 16 21 36 04
salesbelux@pentair.com

BULGARIA

Tel./fax +359 56 86 68 86
fax +359 56 86 68 86
salesee@pentair.com

ČESKÁ REPUBLIKA

Tel. +420 241 009 215
Fax +420 241 009 219
czechinfo@pentair.com

DANMARK

Tel. +45 70 11 04 00
Fax +45 70 11 04 01
salesdk@pentair.com

DEUTSCHLAND

Tel. 0800 1818205
Fax 0800 1818204
salesde@pentair.com

ESPAÑA

Tel. +34 902 125 307
Fax +34 91 640 29 90
ptm-sales-es@pentair.com

FRANCE

Tél. 0800 906045
Fax 0800 906003
salesfr@pentair.com

HRVATSKA

Tel. +385 1 605 01 88
Fax +385 1 605 01 88
salesee@pentair.com

ITALIA

Tel. +39 02 577 61 51
Fax +39 02 577 61 55 28
salesit@pentair.com

LIETUVA/LATVIJA/EESTI

Tel. +370 5 2136633
Fax +370 5 2330084
info.baltic@pentair.com

MAGYARORSZÁG

Tel. +36 1 253 7617
Fax +36 1 253 7618
saleshu@pentair.com

NEDERLAND

Tel. 0800 0224978
Fax 0800 0224993
salesnl@pentair.com

NORGE

Tel. +47 66 81 79 90
Fax +47 66 80 83 92
salesno@pentair.com

ÖSTERREICH

Tel. 0800 297410
Fax 0800 297409
info-ptm-at@pentair.com

POLSKA

Tel. +48 22 331 29 50
Fax +48 22 331 29 51
salespl@pentair.com

REPUBLIC OF KAZAKHSTAN

Tel. +7 495 926 18 85
Fax +7 495 926 18 86
saleskz@pentair.com

РОССИЯ

Тел. +7 495 926 18 85
Факс +7 495 926 18 86
salesru@pentair.com

SERBIA AND MONTENEGRO

Tel. +381 230 401 770
Fax +381 230 401 770
salesee@pentair.com

SCHWEIZ / SUISSE

Tel. 0800 551308
Fax 0800 551309
info-ptm-ch@pentair.com

SUOMI

Puh. 0800 11 67 99
Telekopio 0800 11 86 74
salesfi@pentair.com

SVERIGE

Tel. +46 31 335 58 00
Fax +46 31 335 58 99
salesse@pentair.com

TÜRKIYE

Tel. +90 530 977 64 67
Fax +32 16 21 36 04
ptm-sales-tr@pentair.com

UNITED KINGDOM

Tel. 0800 969013
Fax 0800 968624
salesthermaluk@pentair.com



WWW.THERMAL.PENTAIR.COM

All Pentair trademarks and logos are owned by Pentair or its global affiliates. Pentair reserves the right to change specifications without prior notice.

© 2013 Pentair.