JUMO Quantrol LC100/LC200/LC300

Universal PID Controller Series

B 702030.2.0

Interface Description Modbus

2013-05-27/00600589

1.1 Device documentation

Data sheet T 702030 (as a PDF document)
The data sheet contains general information about the device and forms the basis for planning and purchase decision.

Brief instructions B 702030.7 (printed in DIN A3 format)
These brief instructions contain the most important information about installation, the electrical connection as well as operation, parameterization and configuration of the device. The brief instructions are supplied with every device. For further information, please refer to the operating manual B 702030.0 available as a PDF document.

Operating manual B 702030.0 (as PDF document)
The operating manual contains all information about the installation, electrical connection and the operation, parameter setting and configuration of the device.

Interface description B 702030.2.0 (as a PDF document)
It contains information about the RS485 interface, the Modbus protocol and the communication with other devices.

All PDF documents can be downloaded under www.jumo.net.

1.2 Safety information

This manual contains information that must be observed in the interest of your own safety and to avoid damage to assets. This information is supported by symbols which are used in this manual as follows.
Please read this manual before starting up the device. Keep the manual in a place accessible to all users at all times.
All necessary settings are described in this manual. Manipulations not described in this manual or expressly forbidden will jeopardize your warranty rights.

Warning signs

CAUTION!
This symbol in combination with the signal word indicates that **damage to assets or data loss** will occur if suitable precautions are not taken.

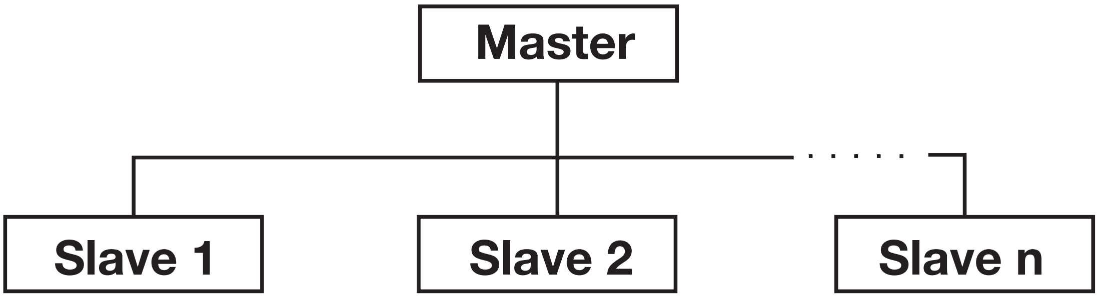
Note signs

TIP!
This symbol refers to **Important information** about the product or its handling or additional use.

REFERENCE!
This symbol refers to **Further information** in other sections, chapters or manuals.

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.

Data word	Parity bit	Stop bit	Number of bits
8 Bits	---	1	9

2.3 Device address

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

TIP!
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

Timing

Data request from master transmission time = n characters * 1000 * x bit/baud rate

Marker for end of data request 3 characters * 1000 * x bit/baud rate

Processing of data request by the slave (£ 250ms)

Response of the slave transmission time = n characters * 1000 * x bit/baud rate

Marker for end of response 3 characters * 1000 * x bit/baud rate

Example

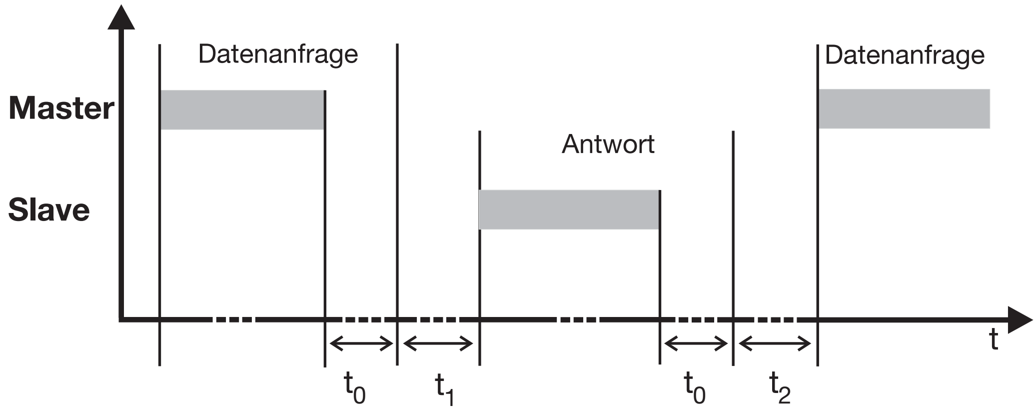
Marker for end of data request or end of response for a 10/9 bit data format

Waiting time = 3 characters * 1000 * 10 bit/baud rate

Baud rate [Baud]	**Data format** [bit]	**Waiting time** [ms] (3 charakters)
19200	9	1.41
9600	9	2.82

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.
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 = 10ms

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:

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

Each data block contains four fields:

Slave address	device address of a specific slave
Function code	function selection (read, write words)
Data field	word address, number of words, word value(s)
Checksum	detection of transmission errors

2.6 Function codes

TIP!
A hexadecimal number is identified by „0x“ preceding the actual number.
Example: 0x0010 (= 16 decimal)

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 number	Function	Limitation
0x03 oder 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)

TIP!
Please refer to chapter 2.9 "Error processing", page 15 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 ≤ 32) words starting from a specific address.

Data request

Slave address	Function 0x03 or 0x04	Address first word	Number of words (max. 32)	Checksum CRC16
1 byte	1 byte	2 byte	2 byte	2 byte

Response

Slave-Adresse	Function 0x03 oder 0x04	Number of bytes read	Word value(s)	Checksum CRC16
1 byte	1 byte	1 byte	x byte	2 byte

Example
Reading the SP1 and SP2 setpoints (2 words each)
Address of first word = 0x3100 (SP1 setpoint)
Data request:

01	03	3100	0004	4AF5

Response (values in the Modbus float format)

01	03	08	0000	41C8	0000	4120	4A93
			Setpoint SP1 (25.0)		Setpoint SP2 (10.0)		

2.6.2 Write one word

This function is used to write a single word to a specific address. 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 byte	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 value monitoring 1 = 275.0
(Value = 0x80004389 in the Modbus float format)

Word address = 0x0056

Instruction: Write the first part of the value

01	06	0056	8000	59DA

Response (as instruction):

01	06	0056	8000	59DA

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

01	06	0057	4389	F88F

Response (as instruction):

01	06	0057	4389	F88F

2.6.3 Write n words

This function is used to write n (n ≤ 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)	Checksum CRC16
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 SP1 and SP2 setpoints (2 words each)
Word address = 0x3100 (SP1 setpoint)
Instruction:

01	10	3100	0004	08	0000	41C8	0000	4120	2A42
					Setpoint SP1 (25.0)		Setpoint SP2 (10.0)		

Response:

01	10	3100	0004	CF36

2.7 Transmission format

2.7.1 Integer values

Integer values are transmitted 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)

2.7.2 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 (32 bit) as per IEEE 754 standard

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM
Byte 1	Byte 2	Byte 3	Byte 4

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

Modbus float format

Modbus address x		Modbus address x+1	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
Byte 3	Byte 4	Byte 1	Byte 2

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:

Address x	Address x+1	Address x+2	Address x+3
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
Byte 4	Byte 3	Byte 2	Byte 1

TIP!
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.

2.8 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

<table>
<tr><td colspan="4">CRC = 0xFFFF</td></tr>
<tr><td rowspan="5"></td><td colspan="3">CRC = CRC XOR ByteOfMessage</td></tr>
<tr><td colspan="3">For (1 to 8)</td></tr>
<tr><td rowspan="3"></td><td colspan="2">CRC = SHR(CRC)</td></tr>
<tr><td colspan="2">if (flag shifted right = 1)</td></tr>
<tr><td>then
CRC = CRC XOR 0xA001</td><td>else</td></tr>
<tr><td colspan="4">while (not all ByteOfMessage processed);</td></tr>
</table>

TIP!
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	

2.9 Error processing

Error codes

The following error codes are used:

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 byte

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

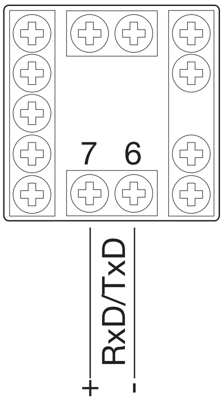
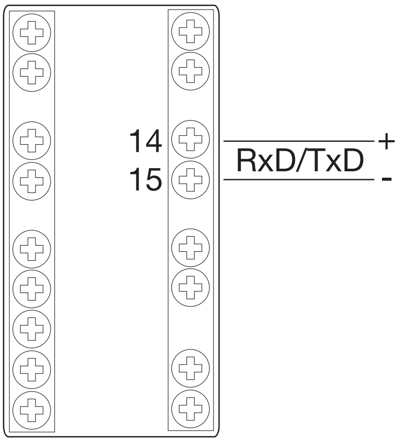
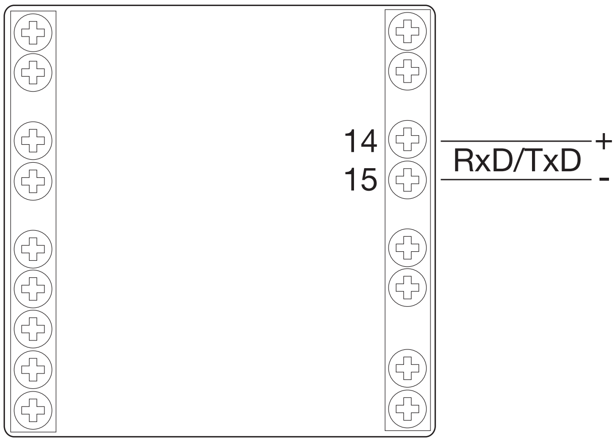
The slave not responding can have the following causes:

- 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

The devices of this controller series can be ordered with an RS485 interface as an option. Information about device version can be found in the data sheet T 702030 (order matrix), the brief instructions B 702030.7 or the operating manual B 702030.0 (identifying the device version).

LC100	LC200	LC300
		

CAUTION!
Electromagnetic fields can interfere with data transmission.
This can lead to transmission errors.
To avoid this, the shielding of the interface cable should be grounded on one side in the electrical cabinet.

3.2 Configuration

The following table shows the possible Modbus interface settings to be carried out on the device (ConF -> IntF) and/or in the setup program.

Parameter	Value	Description
Baud rate bdrt	**0**	**9600 baud**
	1	19200 baud
Device address Adr	0 ... **1** ... 254	Address in data network

TIP!
When the communication takes place via the setup interface (USB), the RS485 interface is inactive.

4.1 Data type and type of access

The following tables contain specifications of all process and device data including their address, 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 byte) as per IEEE 754

CAUTION!
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. 100,000). For this reason, this function can be switched off in the case of frequent programming (please contact Technical support).
The parameter values are then only saved in the volatile memory (RAM) and will be lost after a supply failure.

4.2 Process data

Address	Data type/ bit number	Access	Signal designation
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 ... 5 (Switching states 0 = off / 1 = on)
	Bit 0		Output K1: Relay (= 0x0001)
	Bit 1		Output K2: Relay or Logic (= 0x0002)
	Bit 2		Output K3: Relay or Logic (= 0x0004)
	Bit 3		Output K4: Relay or Logic (= 0x0008)
	Bit 4		Output K5: Relay or Logic (= 0x0010)
0x0023	INT	R/O	Binary input (Switching states 0 = open / 1 = closed)
	Bit 0		Input 1 (= 0x0001)

4 Modbus addresses

Address	Data type/ bit number	Access	Signal designation
0x0024	INT	R/O	Limit value monitoring
	Bit 0		Limit value monitoring 1 (= 0x0001)
	Bit 1		Limit value monitoring 2 (= 0x0002)
0x0025	INT	R/W	Control of the binary outputs
	Bit 0 + Bit 15		Output K1 (= 0x8001)
	Bit 1 + Bit 15		Output K2 (= 0x8002)
	Bit 2 + Bit 15		Output K3 (= 0x8004)
	Bit 3 + Bit 15		Output K4 (= 0x8008)
	Bit 4 + Bit 15		Output K5 (= 0x8010)
0x0026	FLOAT	R/O	Analog input [V], [mA] or [Ohm]
0x0028	FLOAT	R/O	Internal cold junction [Ohm]
0x002A	FLOAT	R/O	Analog input [displayed value]
0x002C	INT	R/O	Sampling time
0x002F	FLOAT	R/O	Controller, ramp end value
0x0031	FLOAT	R/O	Controller, actual value, FILTERED
0x0033	FLOAT	R/O	Controller, actual value, UNFILTERED
0x0035	FLOAT	R/W	Controller, setpoint
0x0037	FLOAT	R/O	Controller, output level display
0x0039	FLOAT	R/O	Output level, controller output 1 (HEATING)
0x003B	FLOAT	R/O	Output level, controller output 2 (COOLING)
0x003D	FLOAT	R/O	Controller, control difference
0x003F	INT	R/O	Switching state, controller output 1 (HEATING)
0x0040	INT	R/O	Switching state, controller output 2 (COOLING)
0x0041	INT	R/O	Output level, manual mode
0x0042	LONG	R/O	Timer run time
0x0044	LONG	R/O	Residual timer time
0x0046	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)

4.3 Setpoints

Address	Data type/ bit number	Access	Signal designation
0x3100	FLOAT	R/W	Setpoint SP1
0x3102	FLOAT	R/W	Setpoint SP2

4.4 Controller parameters

Address	Data type/ bit number	Access	Signal designation
0x3000	FLOAT	R/W	Controller parameter XP1 (Pb 1)
0x3002	FLOAT	R/W	Controller parameter XP2 (Pb2)
0x3004	FLOAT	R/W	Controller parameter TV (dt)
0x3006	FLOAT	R/W	Controller parameter TN (rt)
0x300C	FLOAT	R/W	Controller parameter CY1 (CY 1)
0x300E	FLOAT	R/W	Controller parameter CY2 (CY2)
0x3010	FLOAT	R/W	Controller parameter XSH (db)
0x3012	FLOAT	R/W	Controller parameter XD1 (HYS 1)
0x3014	FLOAT	R/W	Controller parameter XD2 (HYS2)
0x3017	INT	R/W	Controller parameter Y0 (Y0)
0x3018	INT	R/W	Controller parameter Y1 (Y 1)
0x3019	INT	R/W	Controller parameter Y2 (Y2)

4.5 Configuration

Address	Data type/ bit number	Access	Signal designation
0x004E	FLOAT	R/W	Ramp function, ramp rate
0x0050	FLOAT	R/W	Filter time constant
0x0052	FLOAT	R/W	Offset analog input
0x0054	LONG	R/W	Timer value
0x0056	FLOAT	R/W	Limit value monitoring 1, Alarm value AL
0x0058	FLOAT	R/W	Limit value monitoring 1, Hysteresis
0x005A	FLOAT	R/W	Limit value monitoring 2, Alarm value AL
0x005C	FLOAT	R/W	Limit value monitoring 2, Hysteresis

4.6 Commands

Address	Data type/ bit number	Access	Signal designation
0x0047	INT	W/O	Binary functions CONTROLLER
	Bit 0		Self-optimization start (= 0x0001)
	Bit 1		Self-optimization abort (= 0x0002)
	Bit 2		Manual mode (= 0x0004)
	Bit 3		Automatic mode (= 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)
0x0048	INT	W/O	Binary functions OPERATION
	Bit 0		Keyboard inhibit (= 0x0001)
	Bit 1		Configuration and parameter level inhibit (= 0x0002)
	Bit 3		Display OFF (= 0x0008)
0x004A	INT	R/W	Setpoint changeover
	Bit 0		Setpoint 1 (= 0x0000)
	Bit 0		Setpoint 2 (= 0x0001)

4.7 RAM memory

Modbus allows direct access to the RAM memory of the device for writing the controller setpoint (0x3200) and the controller actual value (0x3202).

When writing, a range between -1999 and +9999 is available. In this case, the data written in the device is used instead of the original value.

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

Address	Data type/ bit number	Access	Signal designation
0x3200	FLOAT	W/O	Controller setpoint (writable)
0x3202	FLOAT	W/O	Controller actual value (writable)

JUMO GmbH & Co. KG
Street address:
Moritz-Juchheim-Straße 1
36039 Fulda, Germany
Delivery address:
Mackenrodtstraße 14
36039 Fulda, Germany
Postal address:
36035 Fulda, Germany
Phone: +49 661 6003-0
Fax: +49 661 6003-607
E-mail: mail@jumo.net
Internet: www.jumo.net

JUMO Instrument Co. Ltd.
JUMO House
Temple Bank, Riverway
Harlow - Essex CM20 2DY, UK
Phone: +44 1279 63 55 33
Fax: +44 1279 63 52 62
E-mail: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.
6733 Myers Road
East Syracuse, NY 13057, USA
Phone: 315-437-5866
1-800-554-5866
Fax: 315-437-5860
E-mail: info.us@jumo.net
Internet: www.jumousa.com