JUMO dTRON 304/308/316
JUMO dTRON 304/308/316 plast

Compact Controller with program function
Compact Controller for the plactics industry

B 70.3041.2

Interface Description
Modbus

JUMO

2009-10-20/00442143

Contents

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 instrument. 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.

Warranty

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.

Electrostatic discharge

When accessing the inner parts of the unit and returning plug-in modules, assemblies or components, please observe the regulations according to EN 61340-5-1 and EN 61340-5-2 „Protection of electrostatic sensitive devices“. Only use **ESD** packaging for transport.

Please note that we cannot accept any liability for damage caused by ESD.

ESD=**E**lectro **S**tatic **D**ischarge

1.2 Typographical conventions

1.2.1 Warning signs

The symbols for **Danger** and **Caution** are used in these operating instructions under the following conditions:

Danger This symbol is used when there may be **danger to personnel** if the instructions are ignored or not followed correctly.

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

Caution This symbol is used where special care is required when handling components liable to damage through electrostatic discharge.

1.2.2 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.

abc[1] **Footnote** Footnotes are remarks that refer to specific points in the text. Footnotes consist of two parts:

A marker in the text, and the footnote text.

The markers in the text are arranged as continuous superscript numbers.

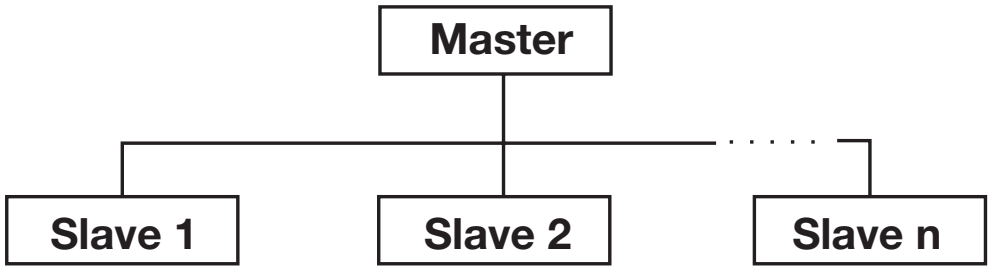
The footnote text (in smaller typeface) is placed at the bottom of the page and starts with a number and a full stop.

1.2.3 Representation

0x0010 **Hexadecimal number** A hexadecimal number is identified by being preceded by an „0x“ (here: 16 decimal).

2.1 Master-slave principle

The communication between a PC (master) and a device (slave) using Modbus protocol takes place according to the master-slave principle, in the form of a 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 are 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 format options are available:

Data word	Parity bit	Stop bit 1/2 bit	Bit number
8 bit	—	1	9
8 bit	even	1	10
8 bit	odd	1	10
8 bit	—	2	10

2 Protocol description

2.3 Device address

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

A maximum of 31 slaves can be addressed via the RS422/485 interface.

Two forms of data exchange can be distinguished:

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. The connected slaves do not respond. A specific setpoint can, for example, be transmitted to all slaves. In such a case, the correct acceptance of the values by the slaves should be checked by a subsequent readout of the setpoint.
A data request with 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 of a single character.

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

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

character transmission time [msec] = 1000 * 9 bits/(baud rate)

For the other data formats it is:

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

Timing

Data request from master transmission time = n characters * 1000 * x bits/(baud rate)
Marker for end of data request 3 characters * 1000 * x bits/(baud rate)
Processing of the data request by the slave (max. 250msec)
Response of slave transmission time = n characters * 1000 * x bits/(baud rate)
Marker for end of response 3 characters * 1000 * x bits/(baud rate)

Example

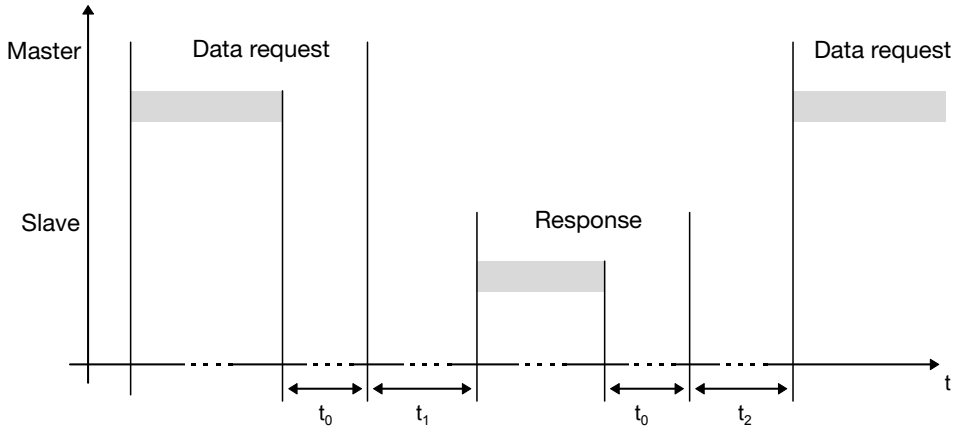
Marker for end of data request or end of response for 10/9 bit data format

Waiting time = 3 characters * 1000 * 10 bits/(baud rate)

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

Timing scheme

A data request runs according to the following timing scheme:



t_0 End marker = 3 characters
(the time depends on the baud rate)

t_1 This time depends on the internal processing.
The maximum processing time is 250 msec.

A minimum response time can be set in the controller, under the menu item “Interface”. This preset time is the minimum time which will be waited before an answer is transmitted (0 — 500 msec). If a smaller value is set, then the response time may be longer than the preset value (because the internal processing time is longer), the controller answers as soon as the internal processing is completed. A preset time of 0 msec means that the controller answers with the maximum possible speed.

The minimum response time which can be set is required by the RS485 interface in the master, in order to switch over the interface driver from transmit to receive. This parameter is not required for the RS422 interface.

t_2 This time is needed by the controller, to switch over from transmit back to receive. This is the waiting time which the master has to observe before presenting a new data request. This time must always be observed, even when the new data request is directed to another device.

RS422 interface: t_2 = 1msec
RS485 interface: t_2 = 10msec

No data requests from the master are permitted during t_1, t_2 and during the response time of the slave. 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(s)	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 information:

- word address
- word number
- word value

Checksum detection of transmission errors

2.6 Error handling

Error codes There are three error codes:

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

The function code is ORed with 0x80, which means that the MSB (most significant bit) is set to 1.

Example

Data request:

01	03	40	00	00	04	CRC16

Response:

01	83	02	CRC16

Special cases

The slave not responding can have the following origins:

- 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.

2.7 Checksum (CRC16)

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

Calculation scheme

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

The checksum is transmitted with the low byte first, followed by 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

2.8 Interface

2.8.1 Configuration

The *plast* series of controllers have expanded settings, which are marked. In addtion, these controllers can be fitted with a current-loop interface

Modbus r422 →

	Symbol	Value/Selection	Description
Protocol	Prot	**0** 1	**Modbus** Modbus integer Modbus master (only for *plast* version) Arburg[1] (only for *plast* version)
Baud rate	bdrt	**0** 1 2 3	**9600 bps** 19200 bps 38400 bps 4800 bps (only for *plast* version)
Data format	dFt	**0** 1 2 3	**8 data bits, 1 stop bit, no parity** 8 data bits, 1 stop bit, odd parity 8 data bits, 1 stop bit, even parity 8 data bits, 2 stop bits, no parity
Device address	Adr	0 — **1** — 255	Address in data network
Min. response time	(setup)	**0** — 500ms	Minimum time that elapses between the request of a device in the data network and the response of the controller.

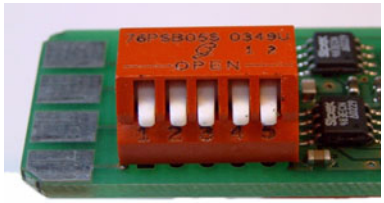
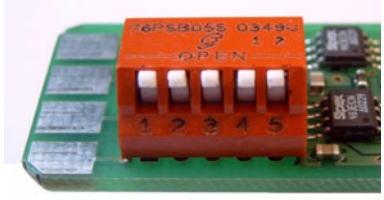
Factory settings are shown **bold**.

[1] Arburg is a registered trademark of Arburg GmbH Co, Loßburg, Germany

2.8.2 Termination resistor for the RS422/485 serial interface

Fault-free operation of several devices in a linear structure requires that the internal terminations resistors are activated for the first and last devices in the bus line.

* Press on the ribbed surfaces and pull out the module.
* Use a ballpoint pen to push all the white switch elements in the same direction.

Bus termination resistor is active:	* Set all 5 switches in the down position. 
No bus termination (factory setting)	* Set all 5 switches in the up position. 

* Plug the module back into the housing.

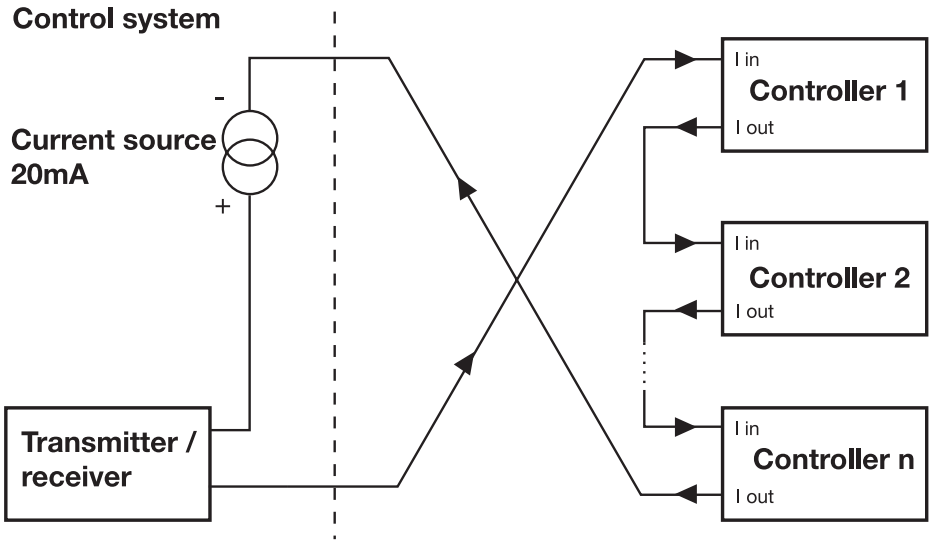
Check

* Press the PGM + ▲ keys

"ON" for an active termination resistor, or "OFF" for an inactive termination resistor, will be shown to the right of the green "VErS" display.

2.8.3 Current-loop interface (only for the plast series)

The "Current-loop interface" option in this controller can interrupt the current flowing in the loop (transmitting) or "listen in" to the switching events in the current loop (receiving).



The following functions are available for the device:

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)

3.1 Read n words

This function reads n (n≤32) words, starting from a defined address.

Data request

Slave address	Function 0x03 or 0x04	Address first word	Word number (max. 32)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

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

Example

Read the two controller setpoints

Word address = 0x3100 (setpoint W1)

Data request:

01	03	31	00	00	04	4AF5

Response:

01	03	08	0000	41C8	0000	4120	4A9E
			Setpoint 1 (25.0)		Setpoint 2 (10.0)		

3 Functions

3.2 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 bytes	2 bytes	2 bytes

Response

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

Example

Write limit for limit comparator 1 = 275

Word address = 0x0077

Instruction: Write first part of the value

01	06	00	77	80	00	5810

Response (as instruction):

01	06	00	77	80	00	5810

Instruction: Write second part of the value

01	06	00	78	43	89	F945

Response (as instruction):

01	06	00	78	43	89	F945

3.3 Write n words

This function writes n (n≤32) words, starting from a defined address.

Instruction

Slave address	Function 0x10	Address of first word	Word number (max. 127)	Byte number	Word value(s)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte(s)	2 bytes

Response

Slave address	Function 0x10	Address of first word	Word number	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Write proportional band Pb1 = 20 of second parameter set

Word address = 0x3014

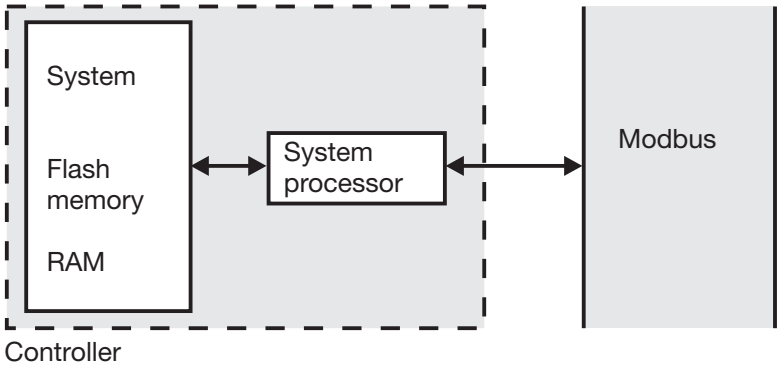
Instruction:

01	10	30	14	00	02	04	00	00	41	A0	9779

Response:

01	10	30	14	00	02	0ECC

3 Functions



The RS422/485 interface is inactive during communication via the setup interface.

All process values (variables) together with their addresses, data type and access mode are described below.

References are as follows:

R/O	read access only
R/W	read and write access
char, byte	byte (8 bits)
int	integer (16 bits)
Bit x	bit No. x
long	long integer (4 bytes)
float	float value (4 bytes) according to IEEE 754

Byte sequence

Because of the platform-dependent representation of floating-point numbers and long values, the bytes must be arranged in the sequence that is appropriate for Modbus.

Please find out in which sequence float values are stored in your system (PC, PLC etc.).

Single-float format (32bit) according to the IEEE 754 standard

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM

S - sign bit
E - exponent (complement to base 2)
M - 23bit normalized mantissa

Modbus float format

Modbus address x		Modbus address x+1	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM

Example: Transmission of the floating-point number 3000

PC (master):	00	80	3B	45
Modbus:	80	00	45	3B
Byte	1	2	3	4

Long values

Example: Transmitting the number 66051

PC (master):	03	02	01	00
Modbus:	00	01	02	03
Byte	1	2	3	4

5.1 Process data

Address	Data type/ bit number	Access	Signal designation
0x001F	INT	R/O	Program status
	Bit 1	R/O	Program pause (=0x0002)
	Bit 2	R/O	Basic status/Program stop = 1 (=0x0004)
	Bit 5	R/O	Automatic mode (=0x0020)
	Bit 6	R/O	Program end signal = 1 (=0x0040)
	Bit 7	R/O	Tolerance limit signal = 1 (=0x0080)
0x0020	INT	R/O	Controller status
	Bit 12	R/O	Manual mode = 1 (=0x1000)
	Bit 15	R/O	Autotuning active = 1 (=0x8000)
0x0021	INT	R/O	Binary outputs 1 — 10 (switching states 0 = off/1 = on)
	Bit 0	R/O	Binary output 1 (=0x0001)
	Bit 1	R/O	Binary output 2 (=0x0002)
	Bit 2	R/O	Binary output 3 (=0x0004)
	Bit 3	R/O	Binary output 4 (=0x0008)
	Bit 4	R/O	Binary output 5 (=0x0010)
	Bit 5	R/O	Binary output 6 (=0x0020)
	Bit 6	R/O	Binary output 7 (=0x0040)
	Bit 7	R/O	Binary output 8 (=0x0080)
	Bit 8	R/O	Binary output 9 (=0x0100)
	Bit 9	R/O	Binary output 10 (=0x0200)
0x0023	INT	R/O	Binary inputs 1 — 6 (switching states 0 = open/1 = closed)
	Bit 0	R/O	Binary input 1 (=0x0001)
	Bit 1	R/O	Binary input 2 (=0x0002)
	Bit 2	R/O	Binary input 3 (=0x0004)
	Bit 3	R/O	Binary input 4 (=0x0008)
	Bit 4	R/O	Binary input 6 (=0x0010)
	Bit 5	R/O	Binary input 5 (=0x0020)
	Bit 6	R/O	Binary input 8 (=0x0040)
	Bit 7	R/O	Binary input 7 (=0x0080)
0x0024	INT	R/O	Limit comparator (switching states 0 = off/1 = on)
	Bit 0	R/O	Limit comparator 1 (=0x0001)
	Bit 1	R/O	Limit comparator 2 (=0x0002)
	Bit 2	R/O	Limit comparator 3 (=0x0004)
	Bit 3	R/O	Limit comparator 4 (=0x0008)
0x0025	INT	R/O	Control contacts 1 — 4
	Bit 0	R/O	Control contact 1 (=0x0001)
	Bit 1	R/O	Control contact 2 (=0x0002)
	Bit 2	R/O	Control contact 3 (=0x0004)

5 Address tables

Address	Data type/ bit number	Access	Signal designation
	Bit 3	R/O	Control contact 4 (=0x0008)
0x0026	INT	R/O	Binary signals
	Bit 0	R/O	Control contact 1 (=0x0001)
	Bit 1	R/O	Control contact 2 (=0x0002)
	Bit 2	R/O	Control contact 3 (=0x0004)
	Bit 3	R/O	Control contact 4 (=0x0008)
	Bit 4	R/O	Timer 1 (=0x0010)
	Bit 5	R/O	Timer 2 (=0x0020)
	Bit 8	R/O	Limit comparator 1 (=0x0100)
	Bit 9	R/O	Limit comparator 2 (=0x0200)
	Bit 10	R/O	Limit comparator 3 (=0x0400)
	Bit 11	R/O	Limit comparator 4 (=0x0800)
	Bit 12	R/O	Logic formula 1 (=0x1000)
	Bit 13	R/O	Logic formula 2 (=0x2000)
0x0027	INT	R/W	Control of the binary outputs
	Bit 0	R/W	Output 1 (=0x8001)
	...	R/W	
	Bit 9	R/W	Output 10 (=0x8200)
	Bit 15	R/W	Activation = 1 (=0x8000)
0x0028	FLOAT	R/O	Analog input 1 [mV]
0x002A	FLOAT	R/O	Analog input 2 [mV]
0x0030	FLOAT	R/O	Internal Pt100 [Ohm]
0x0034	INT	R/O	Sampling cycle time
0x0035	FLOAT	R/O	Analog input 1 [degree]
0x0037	FLOAT	R/O	Analog input 2 [degree]
0x003D	FLOAT	R/O	Math 1
0x003F	FLOAT	R/O	Math 2
0x0041	FLOAT	R/O	Ramp end value (W)
0x0043	FLOAT	R/O	Filtered process value
0x0045	FLOAT	R/O	Unfiltered process value
0x0047	FLOAT	R/W	Setpoint
0x0049	FLOAT	R/O	Output -100 to 100% (displayed value)
0x004B	FLOAT	R/O	Output, heating 0 to 100%
0x004D	FLOAT	R/O	Output, cooling -100 to 0%
0x004F	FLOAT	R/O	Control difference
0x0053	INT	R/O	Switching status, heating (1 = contact closed/ON)
0x0054	INT	R/O	Switching status, cooling (1 = contact closed/ON)
0x0055	INT	R/O	Parameter set number 0 — 1
0x0056	INT	R/W	Manual output
0x0057	INT	R/O	Segment number (1 — 8)
0x0058	INT	R/O	Number of segments (1 — 8)
0x0059	FLOAT	R/O	Program setpoint

Address	Data type/ bit number	Access	Signal designation
0x005B	LONG	R/O	Program run time (in seconds)
0x005D	LONG	R/O	Residual program time (in seconds)
0x0061	LONG	R/O	Segment run time
0x0063	LONG	R/O	Residual segment time
0x0067	LONG	R/O	Timer run time, timer 1
0x0069	LONG	R/O	Timer run time, timer 2
0x006B	LONG	R/O	Timer run time, timer 3
0x006D	LONG	R/O	Timer run time, timer 4

5.2 Setpoints

Address	Data type/ bit number	Access	Signal designation
0x3100	FLOAT	R/W	Setpoint W1
0x3102	FLOAT	R/W	Setpoint W2
0x3104	FLOAT	R/W	Setpoint W3
0x3106	FLOAT	R/W	Setpoint W4

The setpoint limits will not be checked when setpoints are altered via the interface.

Write operations to these parameters result in them being saved to the EEPROM. These memory chips only have a limited amount of write cycles (about 10000), this is why this function can be switched off in the case of frequent programming. The parameter values are then saved to the volatile memory (RAM) only and are lost after a supply failure.

⇨ *Setup/Only Setup/Undocumented parameters/Bit parameters/ Set parameter 2*

5.3 Controller parameters

Address	Data type/ bit number	Access	Signal designation
0x3000	FLOAT	R/W	Parameter set 1: Pb1
0x3002	FLOAT	R/W	Parameter set 1: Pb2
0x3004	INT	R/W	Parameter set 1: dt
0x3005	INT	R/W	Parameter set 1: rt
0x3006	FLOAT	R/W	Parameter set 1: Cy1
0x3008	FLOAT	R/W	Parameter set 1: Cy2
0x300A	FLOAT	R/W	Parameter set 1: db
0x300C	FLOAT	R/W	Parameter set 1: HyS1

Address	Data type/ bit number	Access	Signal designation
0x300E	FLOAT	R/W	Parameter set 1: HyS2
0x3010	INT	R/W	Parameter set 1: tt
0x3011	INT	R/W	Parameter set 1: y0
0x3012	INT	R/W	Parameter set 1: y1
0x3013	INT	R/W	Parameter set 1: y2
0x3014	FLOAT	R/W	Parameter set 2: Pb1
0x3016	FLOAT	R/W	Parameter set 2: Pb2
0x3018	INT	R/W	Parameter set 2: dt
0x3019	INT	R/W	Parameter set 2: rt
0x301A	FLOAT	R/W	Parameter set 2: Cy1
0x301C	FLOAT	R/W	Parameter set 2: Cy2
0x301E	FLOAT	R/W	Parameter set 2: db
0x3020	FLOAT	R/W	Parameter set 2: HyS1
0x3022	FLOAT	R/W	Parameter set 2: HyS2
0x3024	INT	R/W	Parameter set 2: tt
0x3025	INT	R/W	Parameter set 2: y0
0x3026	INT	R/W	Parameter set 2: y1
0x3027	INT	R/W	Parameter set 2: y2

Write operations to these parameters result in them being saved to the EEPROM. These memory chips only have a limited amount of write cycles (about 10000), this is why this function can be switched off in the case of frequent programming. The parameter values are then saved to the volatile memory (RAM) only and are lost after a supply failure.

⇨ *Setup/Only Setup/Undocumented parameters/Bit parameters/ Set parameter 2*

5.4 Configuration

Address	Data type/ bit number	Access	Signal designation
0x0077	FLOAT	R/W	Limit comparator 1, limit AL
0x0079	FLOAT	R/W	Limit comparator 1, switching differential
0x007B	INT	R/W	Limit comparator 1, switch-on delay
0x007C	INT	R/W	Limit comparator 1, switch-off delay
0x007D	FLOAT	R/W	Limit comparator 2, limit AL
0x007F	FLOAT	R/W	Limit comparator 2, switching differential
0x0081	INT	R/W	Limit comparator 2, switch-on delay
0x0082	INT	R/W	Limit comparator 2, switch-off delay
0x0083	FLOAT	R/W	Limit comparator 3, limit AL
0x0085	FLOAT	R/W	Limit comparator 3, switching differential
0x0087	INT	R/W	Limit comparator 3, switch-on delay

Address	Data type/ bit number	Access	Signal designation
0x0088	INT	R/W	Limit comparator 3, switch-off delay
0x0089	FLOAT	R/W	Limit comparator 4, limit AL
0x008B	FLOAT	R/W	Limit comparator 4, switching differential
0x008D	INT	R/W	Limit comparator 4, switch-on delay
0x008E	INT	R/W	Limit comparator 4, switch-off delay
0x008F	FLOAT	R/W	Ramp function, slope
0x00B9	INT	R/W	Alarm text (1-2)
0x00BA	INT	R/W	Alarm text (3-4)

Write operations to these parameters result in them being saved to the EEPROM. These memory chips only have a limited amount of write cycles (about 10000), this is why this function can be switched off in the case of frequent programming. The parameter values are then saved to the volatile memory (RAM) only and are lost after a supply failure.

⇨ *Setup/Only Setup/Undocumented parameters/Bit parameters/ Set parameter 2*

5.5 Program transmission

Address	Data type/ bit number	Access	Signal designation
0x0091	FLOAT	R/W	Segment setpoint 1
0x0093	LONG	R/W	Segment time 1
0x0095	FLOAT	R/W	Segment setpoint 2
0x0097	LONG	R/W	Segment time 2
0x0099	FLOAT	R/W	Segment setpoint 3
0x009B	LONG	R/W	Segment time 3
0x009D	FLOAT	R/W	Segment setpoint 4
0x009F	LONG	R/W	Segment time 4
0x00A1	FLOAT	R/W	Segment setpoint 5
0x00A3	LONG	R/W	Segment time 5
0x00A5	FLOAT	R/W	Segment setpoint 6
0x00A7	LONG	R/W	Segment time 6
0x00A9	FLOAT	R/W	Segment setpoint 7
0x00AB	LONG	R/W	Segment time 7
0x00AD	FLOAT	R/W	Segment setpoint 8
0x00AF	LONG	R/W	Segment time 8
0x00B1	INT	R/W	Control contacts, segment 1 (1=ON; Bit 0=control contact 1; Bit 3=control contact 4)
0x00B2	INT	R/W	Control contacts, segment 2
0x00B3	INT	R/W	Control contacts, segment 3
0x00B4	INT	R/W	Control contacts, segment 4

Address	Data type/ bit number	Access	Signal designation
0x00B5	INT	R/W	Control contacts, segment 5
0x00B6	INT	R/W	Control contacts, segment 6
0x00B7	INT	R/W	Control contacts, segment 7
0x00B8	INT	R/W	Control contacts, segment 8

Write operations to these parameters result in them being saved to the EEPROM. These memory chips only have a limited amount of write cycles (about 10000), this is why this function can be switched off in the case of frequent programming. The parameter values are then saved to the volatile memory (RAM) only and are lost after a supply failure.

⇨ *Setup/Only Setup/Undocumented parameters/Bit parameters/ Set parameter 2*

5.6 Commands

Address	Data type/ bit number	Access	Signal designation
0x006F	INT	R/W	Binary functions "Program controller"
	Bit 1	R/W	Program pause (=0x0002)
	Bit 2	R/W	Program canceled (=0x0004)
	Bit 3	R/W	Program start (=0x0008)
	Bit 8	R/W	Segment change (=0x0100)
	Bit 9	R/W	Store temporary change (=0x0200)
0x0070	INT	R/W	Binary functions "Controller"
	Bit 1	R/W	Pause program (=0x0002)
	Bit 2	R/W	Cancel ramp (=0x0004)
	Bit 4	R/W	Activate manual mode (=0x0010)
	Bit 5	R/W	Automatic mode (=0x0020)
	Bit 6	R/W	Controller off (=0x0040)
	Bit 7	R/W	Start autotuning (=0x0080)
	Bit 13	R/W	Cancel autotuning (=0x2000)
0x0071	INT	R/W	Binary functions "Operation"
	Bit 4	R/W	Display off with key inhibit (=0x0010)
	Bit 5	R/W	Acknowledge limit comparators (=0x0020)
	Bit 6	R/W	Text display (=0x0040)
0x0072	INT	R/W	Binary functions "Timer"
	Bit 1	R/W	Pause timer 1 (=0x0002) (only via interface)
	Bit 2	R/W	Cancel timer 1 (=0x0004)
	Bit 3	R/W	Start timer 1 (=0x0008)
	Bit 9	R/W	Pause timer 2 (=0x0200) (only via interface)
	Bit 10	R/W	Cancel timer 2 (=0x0400)
	Bit 11	R/W	Start timer 2 (=0x0800)

Address	Data type/ bit number	Access	Signal designation
0x0073	INT	R/W	Setpoint switching (0=off) 1 — 4=W1 — W4
0x0074	INT	R/W	Parameter set switching (0=off) 1=P1 2=P2
0x3200	FLOAT	W/O	Setpoint
0x3202	FLOAT	W/O	Process value
0x3004	FLOAT	W/O	Math 1
0x3006	FLOAT	W/O	Math 2
0x3008	INT	W/O	Logic 1+2
	Bit 0	W/O	Logic 1 (=0x0081)
	Bit 1	W/O	Logic 2 (=0x0082)
	Bit 7	W/O	Activation (=0x0080)

5.7 Device identification

Address	Data type/ bit number	Access	Signal designation
0x4000	INT	R/O	Option slot 1+2
	Bit 0 — 7	R/O	Slot 2
	Bit 8 — 15	R/O	Slot 1
0x4001	INT	R/O	Option slot 3
	Bit 8 — 15	R/O	Slot 3
0x4009	INT	R/O	Device format
	Bit 0	R/O	1=Type 703041/45 (48mm x 48mm)
	Bit 1	R/O	1=Type 703042/43/46 (48mm x 96mm)
	Bit 2	R/O	1=Type 703044/48 (96mm x 96mm)

5 Address tables

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 635533
Fax: +44 1279 635262
E-mail: sales@jumo.co.uk
Internet: www.jumo.co.uk

JUMO Process Control, Inc.

8 Technology Boulevard
Canastota, NY 13032, USA
Phone: 315-697-JUMO
1-800-554-JUMO
Telefax: 315-697-5867
E-mail: info@jumo.us
Internet: www.jumo.us