JUMO mTRON

Communication module

70.4040

System Manual Part 9

Contents

1.1 Preface

The system manual is addressed to equipment manufacturers and users with the appropriate technical know-how. It describes the range of functions of the JUMO mTRON automation system with its modules, and provides the information required for project design and start-up.

This Part 9 of the System Manual “JUMO mTRON communication module” contains all the module-specific information.

Part 1 of the System Manual “General section” summarises the information which applies to all modules.

Part 2 of the System Manual “JUMO mTRON-iTOOL project design software” describes the project design software for the JUMO mTRON automation system.

1.2 Type designation

The type designation contains all factory settings of the outputs (1) and the supply (2). The supply must correspond to the voltage shown on the label. The label is affixed to the housing.

	(1)		(2)
704040/0-	..	-	..

(1) Outputs

Outputs	Code
RS232 interface	51
RS422 interface	52
RS485 interface	53

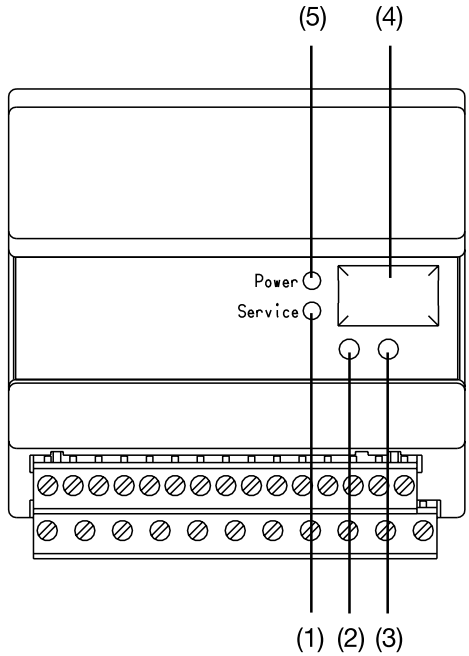
(2) Supply

Type	Code
110 — 240 V +10/-15 % AC 48 — 63 Hz	23
20 — 53 V AC/DC 48 — 63 Hz	22

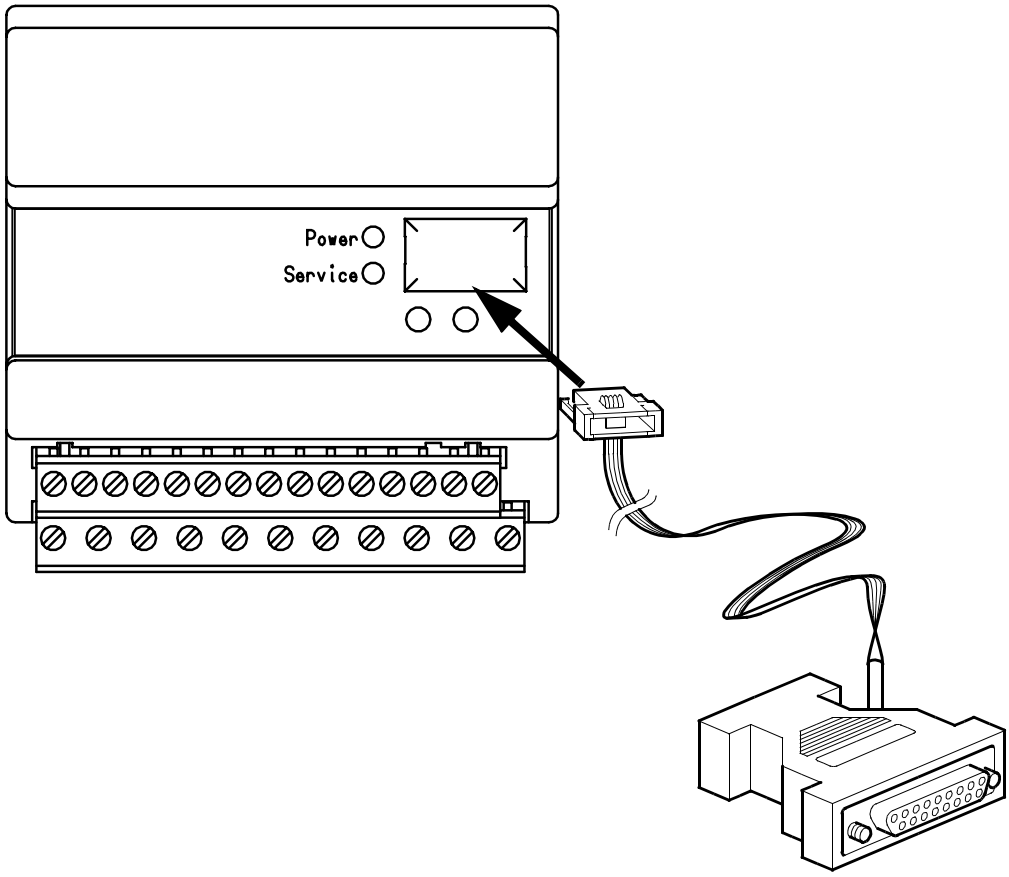
Neuron-ID

Each module has a 12-digit number, by which it can be clearly identified in the JUMO mTRON-iTOOL project design software.

It can be found next to the label.



(1)	**Service LED (red)** - lights up/blinks continuously at one second intervals on operating fault ✶ replace module - blinks at one second intervals for 10 seconds when the network connection from the JUMO mTRON-iTOOL project design software or the operating unit to the module is being tested by a test signal (“wink”). - long blink pulses (3 sec on, 1sec off) when there is a Plug & Play error ⇨ Section 5.6 “Setup data storage on the communication module” - long blink pulses (2sec on, 2sec off) when the unit is in the calibration mode
(2)	**Switches (termination resistance)** ⇨ System Manual Part 1 “General section”, Section 4.2 “Network connection”
(3)	**Installation key** the module reports to the JUMO mTRON-iTOOL project design software or the operating unit
(4)	**Setup interface** for the setup interface line which links the module to the PC. The parameters can be set via this connector not only for the communication module but also **for all the modules connected to the LON bus**. When the setup interface line is connected, the module performs the function of a PC-LON interface converter. The RS232/422/485 interface is disconnected.
(5)	**Power LED (green)** lights up when the supply is switched on



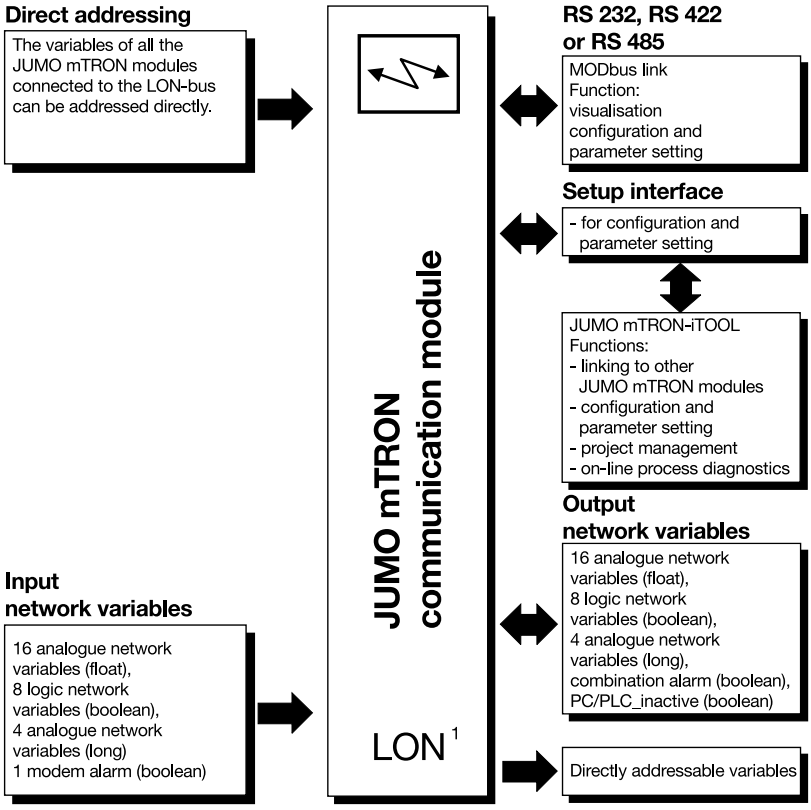
When the interface line is connected, the module performs the function of a PC-LON interface converter. The RS232/422/485 interface is disconnected.

The RS232/422/485 can be used to transmit MODbus commands and to make the link to the JUMO mTRON-iTOOL project design software.

The protocol is recognised automatically.

It is advisable not to plug the setup connector into the communication module but into another mTRON module, since up/downloads take longer when the communication module is plugged in.

3.1 Overview



1. LON® = **L**ocal **O**perating **N**etwork
Registered trademark of the ECHELON Corporation

3.2 Outputs

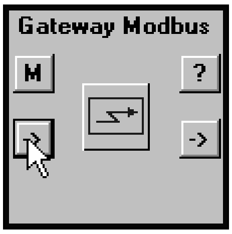
The communication module is accessed in MODbus operation via the RS232 or RS422/485.

3 Integrated functions

After a reset all network variables are set to their default settings.

4.1 Input network-variables

List of input network-variables



Input network-variables can be used to transfer values and operating signals from other modules to the communication module via the network.

Name	Type	Default setting
Alarm_In01	Boolean	0
Bool_In01…08	Boolean	0
Real_In01…16	Real	not programmed (1.5 E38)
Long_In01…04	Long	0

If there is no communication, the input network-variables are set to their default settings and the combination alarm is activated.

Notes on Alarm_In01

The communication module can operate a modem (⇨Section 5.4 “Modem”) via the fieldbus interface (⇨Section 5.3 “Fieldbus”). When the input network-variable Alarm_In01 in the communication module becomes active (TRUE), the communication module will attempt for 30 sec to establish a modem connection. Any existing modem connection will be interrupted first.

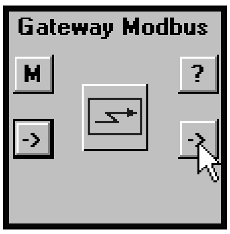
If no modem connection can be set up, a retry will be made after a wait of 2 min. This is repeated until a connection has been made successfully **or** Alarm_In01 is no longer active.

If a modem connection has been successfully established, it will continue to be maintained until no data transfer has occured for at least 30 sec. After that, the connection is automatically broken.

If Alarm_In01 is still active after disconnection, no new connection will be made.

4.2 Output network-variables

List of output network-variables



Using the output network-variables, values and operating signals can be transferred from the communication module to other modules via the network.

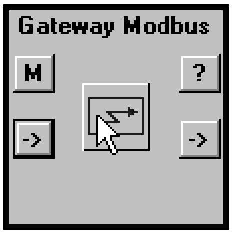
Name	Type	Default setting
Bool_Out01…08	Boolean	0
Real_Out01…16	Real	not programmed (1.5 E38)
Long_Out01…04	Long	0
CombAlarm	Boolean	0
PC_inactive	Boolean	0

The output network-variables are only output if there is communication with a PC/PLC (PC_inactive = 0).

If there is no communication, the output network-variables are set to their default settings and PC_inactive is set to 1.

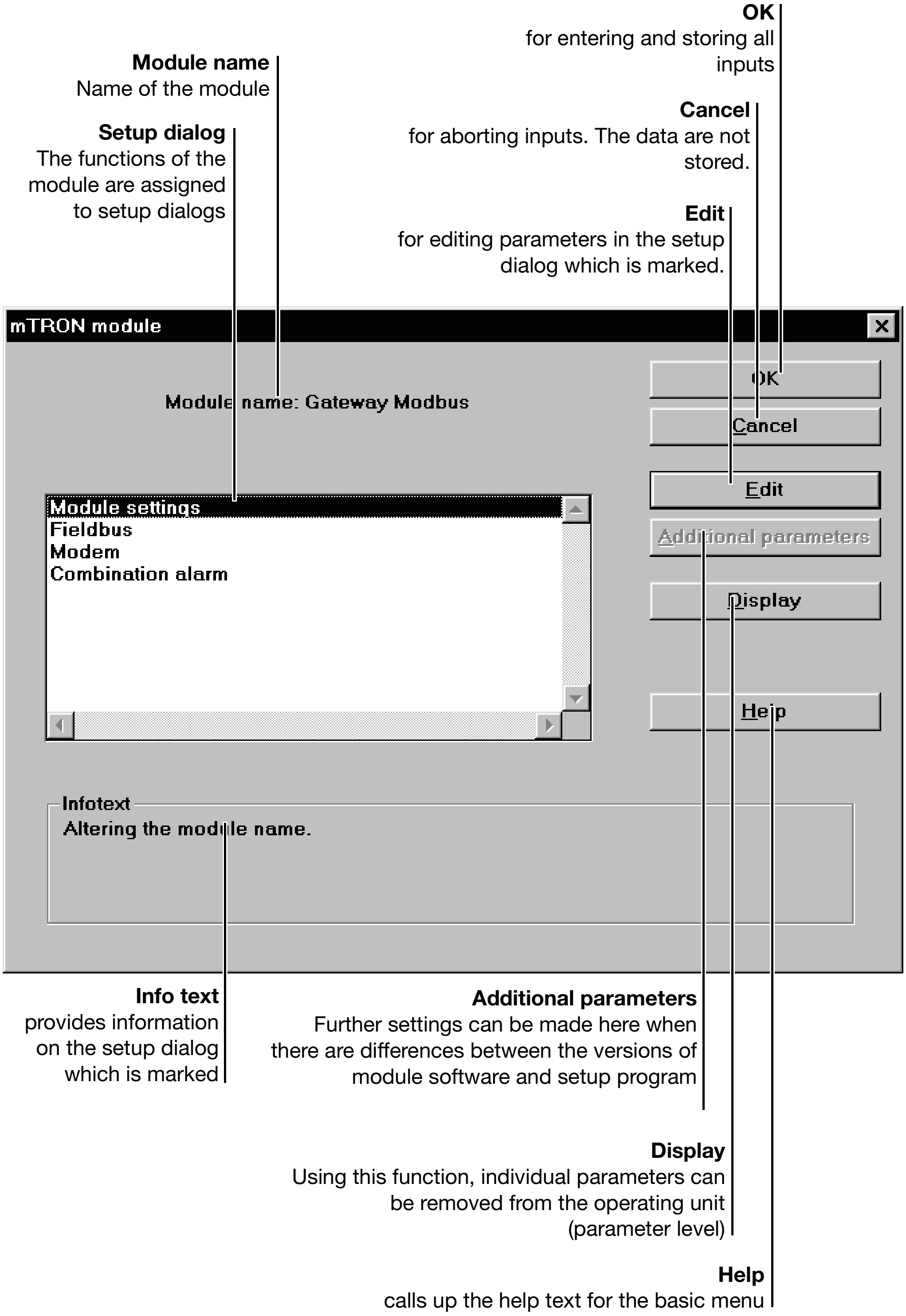
The communication between a PC/PLC and the communication module is monitored at a fixed time interval of 2 min.

5.1 Setup program



The parameters are set with the setup program under JUMO mTRON-iTOOL. The setup program is started by double clicking on the symbol of the communication module in the module window.

Double clicking on an entry in the list of the setup dialogs opens the particular dialog. Alternatively, the dialog can be started by selecting a list entry (single click) and then single clicking on *Edit.*



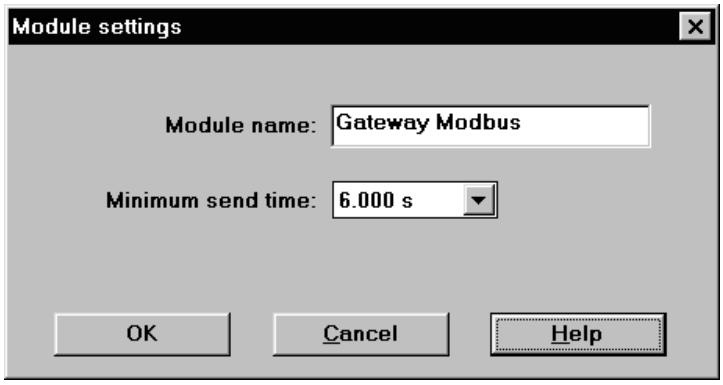
☞ From the iTOOL, the setup for the communication module must not be carried out by download if the communication module is operated as a gateway (with active MODbus) and the setup is not carried out directly via the communication module, but via another mTRON module.

5.2 Module settings

A characteristic designation for the task of the module in the process is assigned here and the time interval of sending repeats of network variables is determined.

Setup dialog

Edit



Parameters

Parameter	Selection/settings	Explanation
Module name [Modulname]	(Text) Gateway Modbus	Name of module (16 characters)
Min Send Time [MinSendT]	n x 200 ms max. time = 8.4 s 6.000 s	Determines the time intervals at which float-type network variables are transmitted over the network. Float-type output network-variables are transmitted without repeat at the time interval MinSendTime. Binary-type output network-variables are transmitted immediately at a change of status (0 → 1, 1 → 0) and are repeated twice. If the status has not changed after 6 sec, there is an automatic safety output via the network to the signal targets.

= factory setting [ ] = short name in the operating unit

5.3 Fieldbus

Automatic protocol recognition

The fieldbus interface incorporates an automatic protocol recognition (MOD/Jbus **or** communication with the JUMO mTRON-iTOOL project design software)

RS232, RS422 and RS485 interface:

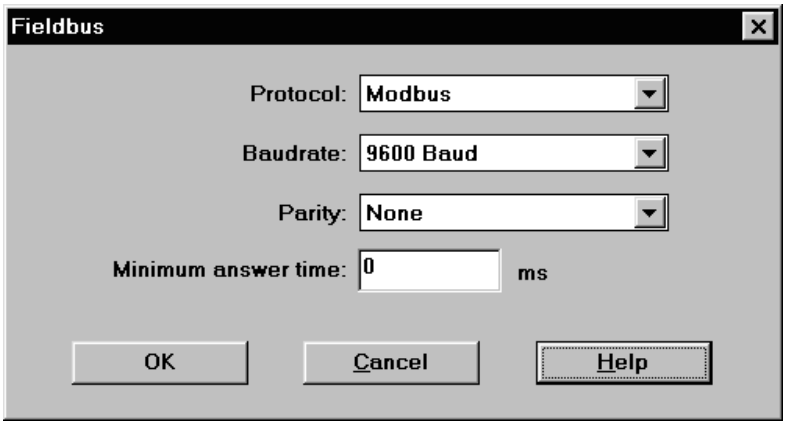
A connection to the JUMO mTRON- iTOOL project design software can be made and MODbus commands can be transmitted.

Blaver setup plug:

Can only be used to connect JUMO mTRON-iTOOL

Setup dialog

Edit



Parameters

Parameter	Selection/settings		Explanation
Protocol [Protocl]	MODbus	[Modbus]	The MODbus protocol is integrated
	Jbus	[J-Bus]	
	No protocol	[NoProt]	
Baud rate [Baudrate]	1200 baud 2400 baud 4800 baud		Determines the speed of the RS232 or RS422/485 interface in the MODbus or Jbus protocol. If the protocol recognition switches the RS232 or RS422/485 interface over to iTOOL protocol, then the correct (permanently-set) parameters baud rate, parity and min. response time will be used automatically for iTOOL operation.
	9600 baud		
	19200 baud 38400 baud		
Parity [Parity]	none	[None]	Determines the parity (error test) If the protocol recognition switches the RS232 or RS422/485 interface over to iTOOL protocol, then the correct (permanently-set) parameters baud rate, parity and min. response time will be used automatically for iTOOL operation
	even	[Even]	
	odd	[Odd]	
Minimum response time [MinTime]	0 ms		Determines the minimum time interval before the communication module responds to a data request ⇨ Section 6.4. “Timing of data request”
	0 — 500 ms		

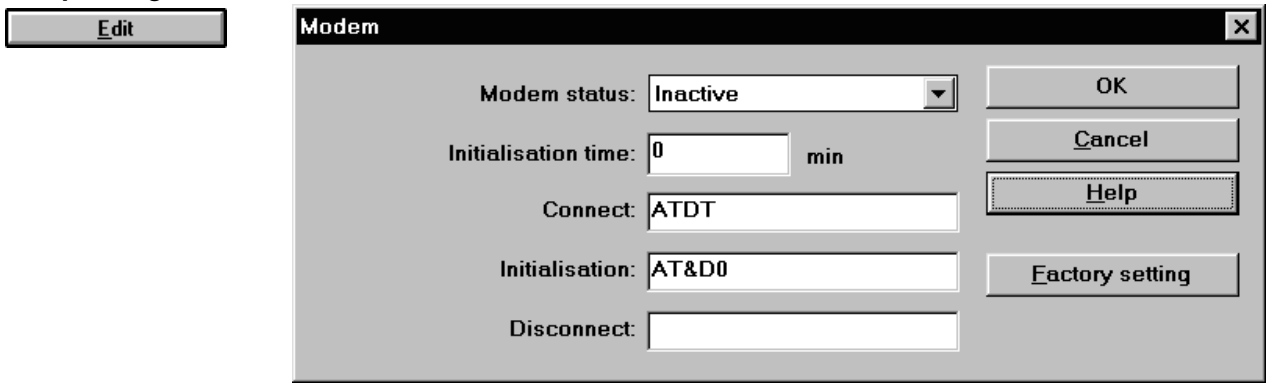
= factory setting [ ] = short name in the operating unit

5.4 Modem

A modem which is connected to the RS232 interface of the communication module can be used to establish a connection to the JUMO mTRON-iTOOL project design software or to any system with a MODbus protocol.

⇨ Section 5.6 “Setup data storage on the communication module”

Setup dialog



Parameters

Parameter	Selection/settings		Explanation
Modem operation [ModemOp]	Inactive	[Inactive]	Switches modem operation on (active) or off (inactive)
	Active	[Active]	
Initialisation time [InitTime]	1…255 min		If modem operation is active, the modem is initialised at the preset time interval, if no data transmission via modem occurs. 0 = initialisation inactive
	0 min		
Connect [Prefix1 … 20]	ATDT		The connection is made, if the input network-variable Alarm_In01 in the communication module becomes (TRUE) and modem operation is active. ⇨ Section 4.1 “Input network-variables” E.g. ATDT 0123456789
	Any modem command		
Initialisation [Init1 … 16]	AT&D0		The command configured here is transmitted to the modem as initialisation command. E. g. ATZ = modem reset For connection to the JUMO mTRON-iTOOL project design software: AT&F0&K0Q1&D0S0=1
	Any modem command		
Disconnect [Suffix1 … 16]	No selection		The command configured here is transmitted to the modem to terminate the connection E.g. ATH = disconnect
	Any modem command		

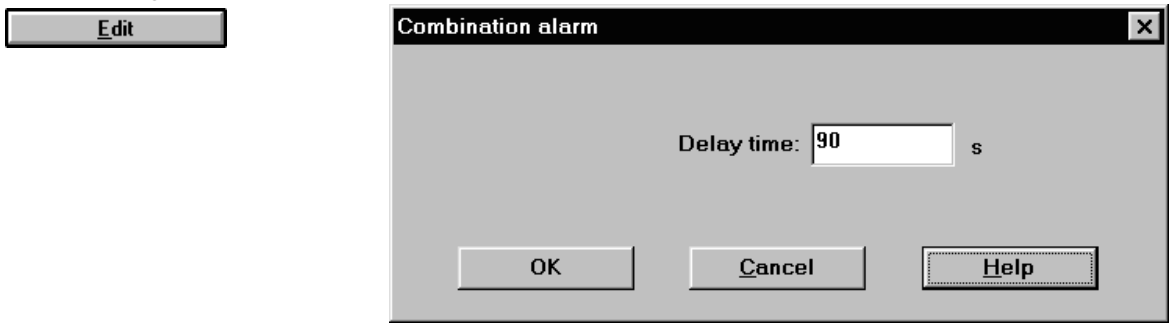
= factory setting [ ] = short name in the operating unit

Modem connection to the JUMO mTRON-iTOOL project design software

The configuration of the JUMO mTRON-iTOOL project design software for a modem connection to the communication module is described in Section “Modem connection” of the System Manual Part 2 “JUMO mTRON-iTOOL project design software”.

5.5 Combination alarm

Setup dialog



Parameters

Parameter	Selection/setting	Explanation
Delay time [Delay]	90 s 0 — 255 s	Delays the output of the combination alarm signal by the set value

■ = factory setting [ ] = short name in the operating unit

⇨ Section 4.1 “Input network-variables”

5.6 Setup data storage on the communication module

After the transfer of the setup data the module must remain switched on for **at least** 90sec so that the setup data can be stored in the non-volatile memory of the module.

If the module was switched off before this waiting time has elapsed, a Plug-&-Play error is indicated when it is switched on next (⇨ Chapter 2 “Indications and controls”).

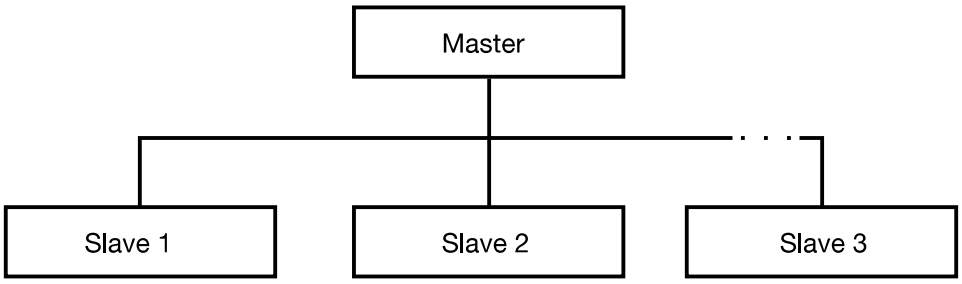
If there is a Plug-&-Play error because this waiting time has not been allowed, it can be rectified by transferring the setup data again followed by the correct waiting time.

After transferring the setup data via another JUMO mTRON module, the communication module must after **at least** 90 seconds be switched off and on again, so that it can read in all modules in the LON network.

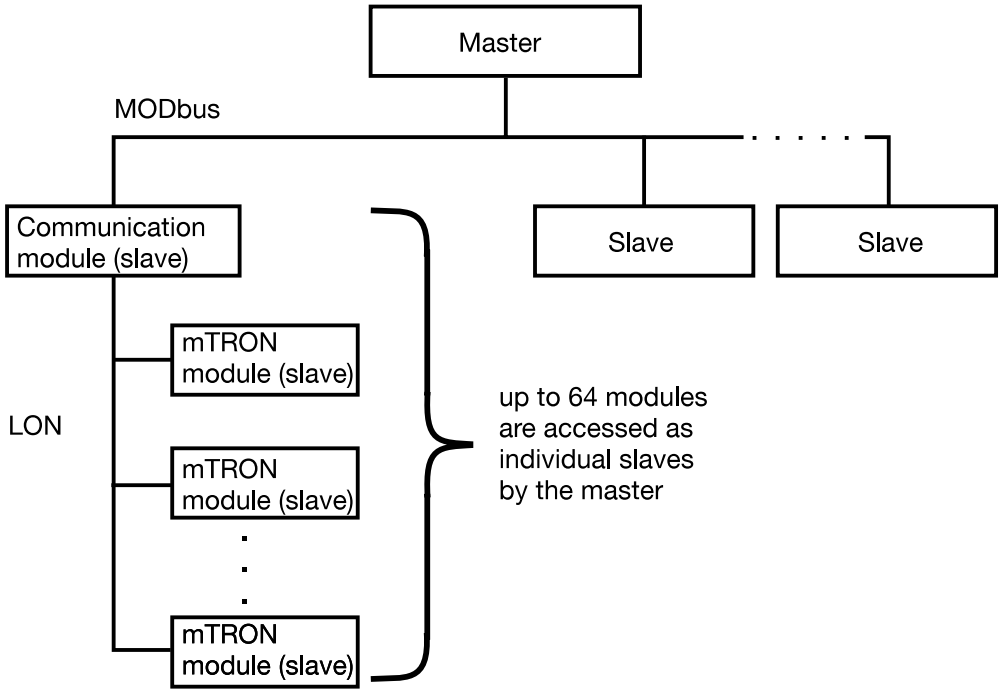
During a setup data transfer there should be no communication via the MODbus interface.

6.1 Master-slave principle

The communication between a PC (PC, PLC …) (master) and a device (slave) using MODbus/Jbus 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. A maximum of 64 slaves can be accessed.



The communication module forms the interface to the LON bus. From the master the individual mTRON modules can be accessed as slave devices. The communication module acts as command converter from MODbus to LON and vice versa. The communication module itself can also be accessed as slave.
Up to 64 modules can be connected to a communication module.

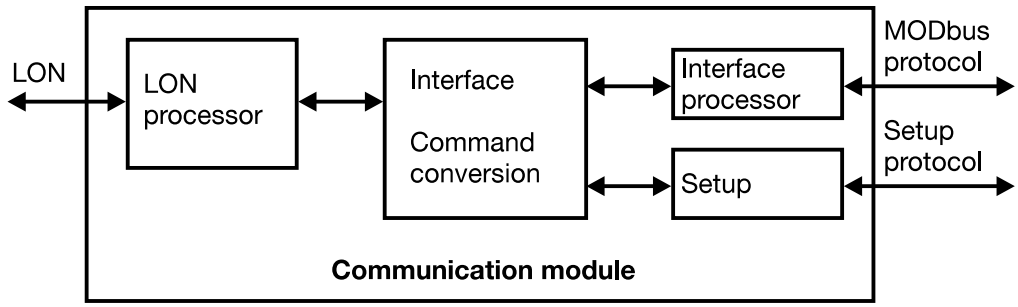
Allocation of the slave device addresses:
The device address is allocated automatically by the JUMO mTRON-iTOOL project design software during the installation of the mTRON installation. These addresses can be read on the project printout:

Example

Module summary

Module name	Type	Version	Address	Neuron ID
Project				
Operating unit	Operating unit	093.01.01	1/ 4	00 01 55 60 27 00
Relay	Relay	090.01.01	1/ 3	00 01 55 50 30 00
Analogue input	Analogue input	089.01.01	1/ 125	00 01 55 69 12 00
Analogue output	Analogue output	088.01.01	1/ 2	00 01 55 50 38 00
Controller	Controller	087.01.01	1/ 1	00 01 54 93 66 00

slave device address



Inside the communication module there is a command conversion from MODbus to LON bus. Each MODbus command is decoded and converted into one or several LON bus commands.

MODbus read command:

- decode the MODbus command
- read all values from the accessed mTRON module
- generate and transmit MODbus response

MODbus write command:

- decode the MODbus command
- write all values to the accessed mTRON module
- generate and transmit MODbus response

For the master these processes are being carried out without being noticed. The action of the communication module is transparent for the master. It accesses the particular mTRON unit directly.

6.2 Transfer mode (RTU)

The transfer mode used is the RTU (Remote Terminal Unit) MODbus mode. Data are transferred in binary format (hexadecimal) with 8 bits, 16 bits for integers, and 32 bits for float values. The LSB (least significant bit) is transferred first.

Data format

The data format describes the arrangement of a byte being transferred. The data formats can be as follows:

Data word	Parity bit	Stop bit 1/2 bit	Number of bits
8 bits	even	1	10
8 bits	odd	1	10
8 bits	none	1	9

6.3 Timing of the communication

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

The character transmission time depends on the baud rate and on the data format used.

If the data format has 8 data bits, no parity bit and one stop bit, then:

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

For other data formats:

character transmission time [msec] = 1000 * 10 bits/(baud rate)

Sequence

Data request from master
Transmisssion time = n characters * 1000 * x bits/(baud rate)

Marker for data request end
3 characters * 1000 * x bits/(baud rate)

Processing the data request by the slave

Response of slave
Transmission time = n characters * 1000 * x bits/(baud rate)

Marker for response end
3 characters * 1000 * x bits/(baud rate)

Example

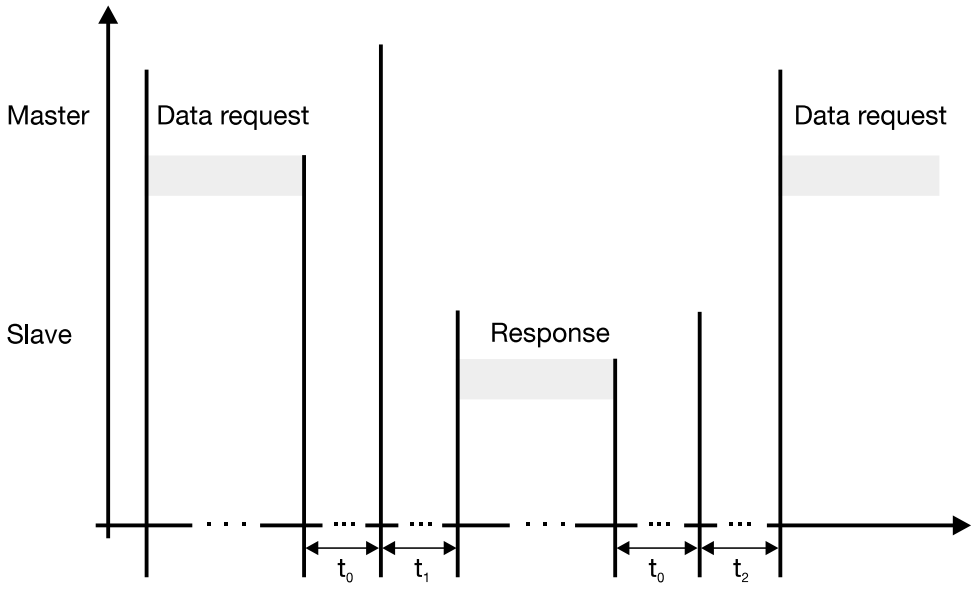
Marker for end of data request or end of response for 10/9 bit data format

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

Baud rate [baud]	**Data format [**bit]	**Waiting time** [msec]
38400	10	0.781
	9	0.703
9600	10	3.125
	9	2.813
4800	10	6.250
	9	5.625

6.4 Timing of a data request

Timing diagram A data request runs according to the following timing diagram:



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

t_1 this time depends on internal processing in the communication module and can take a few seconds.

A minumum response time can be set via setup (⇨ Section 5.3 “Fieldbus”). This time setting is kept as the minimum before a response is sent (0 — 500 msec). If a small value is selected, the response time may be longer than the set value (the internal processing time is longer), the device then responds immediately after the internal processing has been completed. A time setting of 0 msec means that the device responds at the maximum possible speed.

t_2 this is the time required by the device to switch from transmitting back to receiving. The master must wait this time before making a fresh data request. This time must always be maintained, even if the new data request is addressed to a different device.

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

Time-out time in master (PC/PLC)

No exact time can be specified. The response time required by the communication module depends on the following factors:

- length of the MODbus command (number of variables)
- number of LON commands required. A MODbus command may possibly consist of several LON commands
- response time of the accessed JUMO mTRON module
- loading of the LON bus system
- external interference picked up by the bus lines. Such interference may require repetition of commands

When short MODbus commands are being used (few variables), the time-out setting should be 4 seconds. With longer MODbus commands it may be necessary to increase time-out to 7 seconds. In certain cases longer time-outs may become necessary, taking into account the factors listed above. This has to be determined by trial and error. If time-out errors arise in an application, the time-out should be increased.

If a JUMO mTRON module becomes faulty, this is recognised within 3 seconds by the JUMO mTRON communication module. An error message is sent to the master with the next MODbus command.

6.5 Communication with slave

There must be no further command from the master as long as a command sent by the master is being processed by the slave.

6.6 Arrangement 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 consists of four fields:

Slave address	address of a particular slave
Function code	function selection (read, write, bit, word)
Data field	contains the information: -bit address (word address) -number of bits (number of words) -bit value (word value)
Checksum	recognition of transmission errors

6 Protocol description

6.7 Error treatment

Error codes

There are five error codes:

1	invalid function
2	invalid parameter address
3	parameter value outside range of values
4	slave not ready
8	write access to parameter inhibited

Response to error

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

The function code is linked by OR with 0x80, i. e. the MSB (most significant bit) is set to 1.

Example

Data request:

01	02	00	00	00	00	CRC16

Response:

01	82	01	CRC16

Special cases

The slave does not respond for the following errors:

- the checksum (CRC16) is incorrect
- the instruction of the master is incomplete or over-defined
- the number of words or bits to be read is zero

6.8 Distinction MODbus/Jbus

The MODbus protocol is compatible with the Jbus protocol. The structure of the data blocks is identical.

MODbus differs from Jbus in the absolute data addresses. The addresses of MODbus are shifted by one against those of Jbus.

Absolute address	Jbus address	MODbus address
1	1	0
2	2	1
3	3	2
...	...	...

6.9 Checksum (CRC16)

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

Calculation scheme

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

The low byte of the checksum is transmitted first.

Example 1

Data request: read two words from address 1 (CRC16 = 0x0E97)

14	03	00	01	00	02	97	0E

Response: (CRC16 = 0x953E)

14	03	04	03	E8	01	F4	3E	95
			Word 1		Word 2			

Example 2

Instruction: set bit on address 24 (CRC16 = 0xF80E)

14	05	00	18	FF	00	0E	F8

Response (as instruction):

14	05	00	18	FF	00	0E	F8

6 Protocol description

7.1 Write/read functions available

The following functions are available to the device:

Function number	Function	
0x01/0x02	reading n bits	(256 bits max.)
0x03/0x04	reading n words	(35 words max.)
0x05	writing one bit	
0x06	writing one word	
0x0F	writing n bits	(256 bits max.)
0x10	writing n words	(35 words max.)

No separate ranges for bit and word are available for system variables. The ranges for bit and word overlap and can be read and written both as bit range and as word range.

The bit address is calculated as follows:

bit address = word address * 16 + bit number

7.2 Reading n bits

This function reads n bits starting from a defined address.

Data request

Slave address	Function 0x01 or 0x02	Address first bit	Number of bits	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x01 or 0x02	Number of bytes read	Bit values	Checksum CRC16
1 byte	1 byte	1 byte	x byte(s)	2 bytes

Example

Reading 4 bits starting from addr. 0x370

Data request

0A	01	03	70	00	04	CRC16

Response

0A	01	01	0F	CRC16

Reading always covers at least 8 bits (1 byte), irrespective of the number of bits to be read, since the response is in bytes.

In the example above this means that the bits 0x0370 to 0x0377 are being read.

0x0377	0x0376	0x0375	0x0374	0x0373	0x0372	0x0371	0x0370

8 bits = 1 byte

The response for all non-relevant bits (0x0374 to 0x0377) is the value 0.

7.3 Reading n words

This function reads n words starting from a defined address.

Data request

Slave address	Function 0x03 or 0x04	Address first word	Number of words	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

Reading 6 words type 3xFloat starting from addr. 0x031

Data request:

14	03	00	31	00	06	CRC16

Response:

14	03	10	1999	4348	4CCC	4348	2666	4396	CRC16
			Float value 1 200.1		Float value 2 200.3		Float value 3 300.3		

7.4 Writing one bit

In the “writing bit” function the data blocks for instruction and response are identical.

Instruction

Slave address	Function 0x05	Bit address	Bit value XX 00	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Response

Slave address	Function 0x05	Bit address	Bit value XX 00	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Bit value is as follows: FF00 = set bit
0000 = delete bit

Example

Set 1 bit at bitaddr. 0x750

Instruction:

14	05	07	50	FF	00	CRC16

Response (as instruction):

14	05	07	50	FF	00	CRC16

7.5 Writing one word

In the “writing 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 the word at addr. 0x0EB to the value 0x01

Instruction:

14	06	00	EB	00	01	CRC16

Response (as instruction):

14	06	00	EB	00	01	CRC16

7.6 Writing n bits

Instruction

Slave address	Function 0x0F	Address first bit	Number of bits	Number of bytes	Bit value(s)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte(s)	2 bytes

Response

Slave address	Function 0x0F hex	Address first bit	Number of bits	Checksumme CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Write 2 bits from bitaddr. 0x0750 to the values 1 and 0

Instruction:

14	0F	07	50	00	02	01	01	CRC16

Response:

14	0F	07	50	00	02	CRC16

7.7 Writing n words

Instruction

Slave address	Function 0x10	Address first word	Number of words	Number of bytes	Word value(s)	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	x byte(s)	2 bytes

Response

Slave address	Function 16dec = 0x10hex	Address first word	Number of words	Checksum CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example

Write 2 words from addr. 0x077 to the values 0x4142 and 0x4300

Instruction:

14	10	00	77	00	02	04	41	42	43	00	CRC16

Response:

14	10	00	77	00	02	CRC16

7 Write/read functions

8.1 Transmission format

Integer values

Integer values are transmitted via MODbus in the following format:
first the high byte, then the low byte.

E. g.: Write the int value 1 (= 0x0001) to the address 0x00EB:
010600EB**0001**383E

Float values

With float values, MODbus operates with the IEEE 754 standard format (32 bit), but with the difference that bytes 1 and 2 are interchanged with bytes 3 and 4.

Single float format (32 bit) to IEEE 754 Standard

SEEEEEEE	EMMMMMMM	MMMMMMMM	MMMMMMMM
byte 1	byte 2	byte 3	byte 4

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

Modbus float format

MODbus address x		MODbus address x+1	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
byte 3	byte 4	byte 1	byte 2

E. g.:writing the float value 550.0
(= 0x44098000 in IEEE 754 format) to the address 0x00FB:
011000FB000204**80004409**679E

Before / after transmission to/from the device the bytes of the float value have to be suitably interchanged.

Many compilers, e.g. Mircosoft C++, Turbo C++, Turbo Pascal, Keil C51) store float values in the following order:

float value

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

8 Data flow

9.1 General errors

0	no error

Error list of the program store archival

1	program can not be created
2	program does not exist
3	program can not be deleted
4	section can not be deleted
5	checksum can not be stored
6	checksum can not be read
7	program can not be copied
8	section can not be copied
9	program checksum error
10	program pointer table checksum error
11	program store end
12	section does not exist
13	repeat jump markers can not be configured

Error list of the general input and output

14	please enter by pressing ENTER
15	invalid number of digits
16	input contains illegal characters
17	value outside limits
18	section not programmed correctly
19	password error

Error list of instruction programming

20	busy flag not reset by master
21	illegal instruction
22	error in receiving data
23	no cyclic data
24	illegal structure length
25	illegal header ID

Error list of the key and programming inhibit

26	keys inhibited
27	programming inhibited
28	write error in the serial EEPROM (calib)
29	hardware error: HAND and AUTO inhibited
30	edit not permitted while program is running
31	copying the actual program not permitted
32	HAND not permitted during AUTO delay time
33	section change! Display structure required
34	no DB No. for display structure of PLC
35	no DB No. for process values of PLC
36	printer busy or not ready
37	setpoint 1 not programmed
38	configure printer (config/interface)
39	only possible when device in HAND mode
40	self-optimisation already running
41	time axis expired or not programmed
42	time axis can not be copied
43	no time axis
44	program alteration is inhibited
45	manual operation is inhibited
46	program start is inhibited

9.2 Setpoints/process values and calculated values

1.1e38	first error value
1.1e38	float error value underrange
1.2e38	float error value overrange
1.3e38	division by 0
1.4e38	invalid input value
1.5e38	not programmed
1.6e38	overrange (hardware)

9.3 System and run time errors

System errors

0	no error
1	illegal vector interrupt
2	stack pointer outside the stack range
3	task run time exceeded
4	EEPROM write error
5	division by zero
6	RTC run time error
7	no fuzzy parameters
8	no return message AD converter
9	no return message relay module
10	hall sensor faulty
11	PLC stack overflow
12	no PLC program
13	initialisation phase
14	external error message
15	address conflict
16	reserved

Run time errors

0	no error
1	program data corrupted
2	restart data corrupted
3	no chart
4	no reply from relay module
5	clock time must be set
6	battery discharged
7	config data corrupted
8	AD converter faulty

9 Error messages

The tables below describe all process values (variables) with their MODbus addresses, the data type and the access.

Data type

Data type	Explanation
bit x	bit No. x
word	2 bytes (16 bits)
float	Float value (4 bytes)

Access

Access	Explanation
r	read
w	write
r/w	read/write

10 MODbus address tables

10.1 Controller module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word	r	operating status
2F	word bit0 bit1	r	logic HW inputs logic input 1 logic input 2
30	word bit0 bit1	r	logic HW outputs logic output 1 logic output 2
31	float	r	AnIn1_Meas
33	float	r	AnIn2_Meas
35	float	r	controller setpoint
37	float	r	controller process value
39	float	r	controller output 1
3B	float	r	controller output 2
3D	float	r	analogue output
3F	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 bit8 bit9 bit10 — 15	r r r r r r r r r r r r	logic network outputs AnIn1_Alarm AnIn2_Alarm AnIn1_Warn AnIn2_Warn combination alarm logic input 1 logic input 2 logic output 1 (relay 1) logic output 2 (relay 2) LC not used
40	word bit0 bit1 bit2 bit3 bit4 — 15	r r r r r r	actual controller operating mode not used not used auto mode manual mode not used

MODbus address	Data type	Access	Note
41	word bit0 bit1 bit2 bit3 — 15	r r r r r	self-optimisation status tune inactive tune active tune completed not used
42	float	r/w	analogue input 1 low limit
44	float	r/w	analogue input 2 low limit
46	float	r/w	analogue input 1 high limit
48	float	r/w	analogue input 2 high limit
4A	float	r/w	setpoint 1
4C	float	r/w	setpoint 2
4E	float	r/w	setpoint 3
50	float	r/w	setpoint 4
52	word bit0 bit1 bit2 bit3 bit4 — 15	r/w r/w r/w r/w r/w r	controller operating mode set not used not used auto mode manual mode not used
			parameter set 1
53	float	r/w	proportional band Xp1
55	float	r/w	proportional band Xp2
57	float	r/w	reset time Tn
59	float	r/w	derivative time Tv
5B	float	r/w	working point Y0
			parameter set 2
5D	float	r/w	proportional band Xp1
5F	float	r/w	proportional band Xp2
61	float	r/w	reset time Tn
63	float	r/w	derivative time Tv
65	float	r/w	working point Y0
			parameter set 1
67	float	r/w	XSh

MODbus address	Data type	Access	Note
69	float	r/w	manual output prog. (only if manual output = manual output prog.)
6B	float	r/w	filter time constant df
6D	float	r/w	limit comparator limit
6F	word	r/w	self-optimisation start (only if controller parameter self-optimisation start = operating unit)

10.2 Relay module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word bit0 bit1 bit2 bit3 bit4 bit 5 — 15	r r r r r r r	logic network outputs combination alarm relay 1 relay 2 relay 3 relay 4 not used
2F	float	r/w	relay 1 offset
31	float	r/w	relay 2 offset
33	float	r/w	relay 3 offset
35	float	r/w	relay 4 offset
37	float	r/w	limit comparator 1 low limit
39	float	r/w	limit comparator 2 low limit
3B	float	r/w	limit comparator 3 low limit
3D	float	r/w	limit comparator 4 low limit
3F	float	r/w	limit comparator 1 high limit

MODbus address	Data type	Access	Note
41	float	r/w	limit comparator 2 high limit
43	float	r/w	limit comparator 3 high limit
45	float	r/w	limit comparator 4 high limit

10.3 Analogue input module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word	r	operating status
2F	word bit0 bit1 —15	r r	logic HW inputs logic input not used
30	word	r	counter input
31	float	r	AnIn1_Meas
33	float	r	AnIn2_Meas
35	float	r	AnIn3_Meas
37	float	r	AnIn4_Meas
39	float	r	maths
3B	float	r	linearisation 1
3D	float	r	linearisation 2
3F	float	r	linearisation 3
41	float	r	linearisation 4

MODbus address	Data type	Access	Note
43	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 bit8 bit9 bit10	r r r r r r r r r r r r	logic network outputs AI1_alarm AI2_alarm AI3_alarm AI4_alarm AI1_warning AI2_warning AI3_warning AI4_warning combination alarm LogicIn LC
44	float	r/w	analogue input 1 low limit
46	float	r/w	analogue input 2 low limit
48	float	r/w	analogue input 3 low limit
4A	float	r/w	analogue input 4 low limit
4C	float	r/w	analogue input 1 high limit
4E	float	r/w	analogue input 2 high limit
50	float	r/w	analogue input 3 high limit
52	float	r/w	analogue input 4 high limit
54	float	r/w	limit comparator low limit
56	float	r/w	limit comparator high limit

10.4 Analogue output module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word bit0 bit1 — 15	r r r	logic HW inputs logic input not used
2F	word bit0 bit1 bit 2 — 15	r r r	logic network inputs logic inputs combination alarm not used
30	float	r/w	analogue output 1 low limit 1
32	float	r/w	analogue output 1 low limit 2
34	float	r/w	analogue output 1 high limit 1
36	float	r/w	analogue output 1 high limit 2
38	float	r/w	analogue output 2 low limit 1
3A	float	r/w	analogue output 2 low limit 2
3C	float	r/w	analogue output 2 high limit 1
3E	float	r/w	analogue output 2 high limit 2

10.5 Logic module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C … 012A	word	r/w	255 Marker Bool Marker Bool 1 2C Marker Bool 2 2D Marker Bool 3 2E ... Marker Bool 255 012A
012B … 01AA	word	r/w	128 Marker word Marker word 1 012B Marker word 2 012C Marker word 3 012D ... Marker word 128 01AA
01AB … 022A	float	r/w	64 Marker float Marker float 1 01AB Marker float 2 01AD Marker float 3 01AF ... Marker float 64 0229
022B … 02BA	word	r/w	36 Marker date/time Marker date/time 1 022B Marker date/time 2 022F Marker date/time 3 0233 ... Marker date/time 36 02B7
02BB … 03B9	word	r/w	255 Data block Bool Data block Bool 1 02BB Data block Bool 2 02BC Data block Bool 3 02BD … Data block Bool 255 03B9
03BA … 0439	word	r/w	128 Data block word Data block word 1 03BA Data block word 2 03BB Data block word 3 03BC … Data block word 128 0439
043A… 04B9	float	r/w	64 data block float Data block float 1 043A Data block float 2 043C Data block float 3 043E … Data block float 64 04B8

MODbus address	Data type	Access	Note
04BA … 0549	word	r/w	36 data block date/time Data block date/time 1 04BA Data block date/time 2 04BE Data block date/time 3 04C2 … Data block date/time 36 0546
054A … 0551	word	r	Hardware input 1 — 8
0552 … 0557	word	r	Hardware output 1 — 6
0558	word	r	System error ⇨ Section 9.3 “System and run time errors”
0559	word	r	Run time errors Section 9.3 “System and run time errors”
055A	word bit0 bit1 —15	r r	 Combination alarm not used
055B	word bit0 bit1 bit2 bit3 bit4 —15	r r r r r	Buffer mode Power on Run Stop Breakpoint not used

10.6 Operating unit

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word	r	operating status (const. = 4; auto mode)
2F	word bit0 bit1 bit2 — 15	r r r r	logic HW inputs logic input 1 logic input 2 not used
30	word	r	logic HW output (logic level)
31	word bit0 bit1 bit2 bit3 bit4 — 15	r r r r r r	 Logic inp 1 Logic inp 2 Logic output (logic level) CombAlarm_in not used
32	word bit0 — 15	r r	Alarm window 1 — 16 Alarm window 1 — 16

10.7 Controller operating unit

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time errors ⇨ Section 9.3 “System and run time errors”
2E	word	r	operating status (const. = 4; auto mode)
2F	word bit0 bit1 bit2 — 15	r r r r	logic HW inputs logic input 1 logic input 2 not used
30	word	r	logic HW outputs (logic level)
31	bit0 bit1 bit2	r r r	Logic inp 1 Logic inp 2 Logic output (logic level)

10.8 Communication module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	float	r	Real_In01
30	float	r	Real_In02
32	float	r	Real_In03
34	float	r	Real_In04
36	float	r	Real_In05
38	float	r	Real_In06
3A	float	r	Real_In07
3C	float	r	Real_In08
3E	float	r	Real_In09
40	float	r	Real_In10
42	float	r	Real_In11
44	float	r	Real_In12
46	float	r	Real_In13
48	float	r	Real_In14
4A	float	r	Real_In15
4C	float	r	Real_In16
4E	float	r/w	Real_Out01
50	float	r/w	Real_Out02
52	float	r/w	Real_Out03
54	float	r/w	Real_Out04
56	float	r/w	Real_Out05
58	float	r/w	Real_Out06
5A	float	r/w	Real_Out07
5C	float	r/w	Real_Out08
5E	float	r/w	Real_Out09
60	float	r/w	Real_Out10

MODbus address	Data type	Access	Note
62	float	r/w	Real_Out11
64	float	r/w	Real_Out12
66	float	r/w	Real_Out13
68	float	r/w	Real_Out14
6A	float	r/w	Real_Out15
6C	float	r/w	Real_Out16
6E	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 bit8 — 15	r r r r r r r r r r	logic network inputs Bool_In01 Bool_In02 Bool_In03 Bool_In04 Bool_In05 Bool_In06 Bool_In07 Bool_In08 not used
6F	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 bit8 — 15	r/w r/w r/w r/w r/w r/w r/w r/w r/w r/w	logic network outputs Bool_Out01 Bool_Out02 Bool_Out03 Bool_Out04 Bool_Out05 Bool_Out06 Bool_Out07 Bool_Out08 not used
70	word	r	Long_In01
71	word	r	Long_In02
72	word	r	Long_In03
73	word	r	Long_In04
74	word	r/w	Long_Out01
75	word	r/w	Long_Out02
76	word	r/w	Long_Out03
77	word	r/w	Long_Out04
78	char16	r/w	Modulname

10.9 LPF process module

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	runt time error ⇨ Section 9.3 “System and run time errors”
2E	float	r	Act_Chamber
30	float	r	Act_Humidity
32	float	r	Act_Core
34	float	r	Act_Temperature
36	float	r	Setpt_Chamber
38	float	r	Setpt_Humidity
3A	float	r	Setpt_Temperature
3C	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 — 15	r r r r r r r r r	logic network inputs Combination alarm LimitComp1 LimitComp2 LimitComp3 LimitComp4 LimitComp5 LimitComp6 not used
3D	word	r	Relays 01 — 08
3E	word	r	Relays 09 — 16
3F	word	r	Relays 17 — 25
40	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 — 15	r r r r r r r r	Control states Controller-1 heat Controller-1 cool Controller-2 heat Controller-2 cool Controller-3 heat Controller-3 cool not used
41	float	r	Controller1_Output1
43	float	r	Controller1_Output2
45	float	r	Controller2_Output1
47	float	r	Controller2_Output2
49	float	r	Controller3_Output1
4B	float	r	Controller3_Output2

MODbus address	Data type	Access	Note
4D	float	r/w	Offset X_0 chamber
4F	float	r/w	Offset X_0 humidity
51	float	r/w	Offset X_0 core
53	float	r/w	Offset X_0 temperature
55	float	r/w	Controller 1 Xp1
57	float	r/w	Controller 1 Xp2
59	float	r/w	Controller 1 T_N
5B	float	r/w	Controller 1 T_V
5D	float	r/w	Controller 2 Xp1
5F	float	r/w	Controller 2 Xp2
61	float	r/w	Controller 2 T_N
63	float	r/w	Controller 2 T_V
65	float	r/w	Controller 3 Xp1
67	float	r/w	Controller 3 Xp2
69	float	r/w	Controller 3 T_N
6B	float	r/w	Controller 3 T_V

10.10 LPF operating unit

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 bit7 bit8 bit9 bit10 bit11 bit12 — 15	r r r r r r r r r r r r r r	operating mode not used unit in base status unit in auto mode unit in manual mode unit in programming mode unit in restart mode unit in delay mode auto-manual active ext. auto-manual mode active unit in profile end mode unit in stop mode range condition active not used
2F	float	r	Setpt_ChamberDispl
31	float	r	Setpt_Humidity
33	float	r	Setpt_Core
35	float	r	Sept_Temperature
37	float	r	Act_F_Value
39	float	r	Act_C_Value
3B	float	r	Act_Chamber
3D	float	r	Act_Humidity
3F	float	r	Act_Core
41	float	r	Act_Temperature
43	word	r	bit0 — 7 = operating contacts 01 — 08
44	word	r	bit0 — 7 = operating contacts 09 — 16
45	word	r	bit0 — 7 = operating contacts 17 — 24
46	word	r	bit0 — 7 = operating contacts 25 — 32
47	word	r	bit0 — 3 = operating contacts 33 — 36
48	word	r	current profile No.
49	word	r	current segment No.

MODbus address	Data type	Access	Note
4A	word	r	No. of current process step
4B	word	r	No. of next process step
4C	float	r	residual segment time in sec
4E	word	w	set operating status; bits can only be set individually, do not write as word.
	bit0 — 3	w	not used
	bit4	w	enter temporary change
	bit 5 — 7	w	internally occupied, do not use
	bit8	w	next segment
	bit9	w	internally occupied, do not use
	bit10	w	manual
	bit11	w	internally occupied, do not use
	bit12	w	stop profile
	bit13	w	internally occupied, do not use
	bit14	w	start profile
	bit15	w	internally occupied, do not use
4F	float	r/w	Setpt_Chamber for temporary changes
51	float	r/w	Setpt_Humidity for temporary changes
53	float	r/w	Setpt_Core for temporary changes
55	float	r/w	Setpt_Temperature for temporary changes
57	word	r/w	bit0 — 7 = operating contacts 01 — 08 for temporary changes
58	word	r/w	bit0 — 7 = operating contacts 09 — 16 for temporary changes
59	word	r/w	bit0 — 7 = operating contacts 17 — 24 for temporary changes
5A	word	r/w	bit0 — 7 = operating contacts 25 — 32 for temporary changes
5B	word	r/w	bit0 — 3 = operating contacts 33 — 36 for temporary changes
5C	word	r/w	segment set time for temporary changes MSB = 0 input in sec MSB = 1 input in min
5D valid up to software version 085.01.03	4x word	w	start time for profile start (LPF 200 only) date+time (7 bytes max.)
	word1	w	bit0 — 7 = min bit8 — 15 = sec
	word2	w	bit0 — 7 = day bit8 — 15 = hour
	word3	w	bit0 — 7 = year bit8 — 15 = month

MODbus address	Data type	Access	Note
5D valid from software version 085.02.04	4x word word1 word2 word3 word4	w w w w w	start time for profile start (LPF 200 only) date+time (7 bytes max.) Year bit0 — 7 = day bit8 — 15 = month bit0 — 7 = minute bit8 — 15 = hour bit0 — 7 = second bit8 — 15 = not used
61	word	r/w	profile number for profile start
62	word	r/w	segment number for profile start
63	word	w	delay time (for LPF 100 only) for profile start
64	char8	r/w	batch number from software version 085.02.04
68	char8	r/w	user number from software version 085.02.04
6C	float	r/w	batch number as float from software version 085.03.06
6E	float	r/w	user number as float from software version 085.03.06

10.11 Cooking kettle controller LKR-96

MODbus address	Data type	Access	Note
0 … 2B			internal data
2C	word	r	system error ⇨ Section 9.3 “System and run time errors”
2D	word	r	run time error ⇨ Section 9.3 “System and run time errors”
2E	word	r	operating mode
2F	word	r	profile number
30	word	r	segment number
31	float	r	measurement analogue input 1 (kettle temp.)
33	float	r	measurement analogue input 2 (core temp.)
35	float	r	actual F-value

MODbus address	Data type	Access	Note
37	float	r	setpoint kettle temperature
39	float	r	setpoint core temperature
3B	float	r	setpoint F-value
3D	bit0 bit1 bit2 bit3 bit4	r r r r r	combination alarm output LC1 output LC2 controller1 heating controller1 cooling
3E	word bit0 — 10 bit11 bit12 bit13 bit14 bit15	r/w r/w r/w r/w r/w r/w r/w	set operating status; bits can only be set individually, do not write as word. not used not used stop auto profile not used start profile not used
3F	float	w	setpoint kettle temperature for temporary changes
41	float	w	setpoint core temperature for temporary changes
43	float	w	setpoint F-value for temporary changes
45	float	w	segment set time for temporary changes in sec - time not programmed = $1.5 * 10^{38}$
47	4x word	w	start time for profile start date + time (7 bytes max.)
4B	word	w	profile No. for profile start
4C	word	r	segment number for timing
4D	float	r	continuous control output 1
4F	float	r	continuous control output 2

10.12 Screen recorder LOGOSCREEN

MODbus address	Data type	Access	Note
0 … 2B			internal data
2F	word bit0 — bit7 bit8 bit9 bit10 bit11 bit12 bit13 bit14 — 15	r	logic network outputs not used Bool_Out01 Bool_Out02 Bool_Out03 Bool_Out04 Bool_Out05 Bool_Out06 not used
32	word bit0 bit1 bit2 bit3 bit4 bit5 bit6 —15	r	logic network inputs Bool_In01 Bool_In02 Bool_In03 Bool_In04 Bool_In05 Bool_In06 not used
35	float	r	Real_Out01
37	float	r	Real_Out02
39	float	r	Real_Out03
3B	float	r	Real_Out04
3D	float	r	Real_Out05
3F	float	r	Real_Out06
41	float	r	Real_Out07
43	float	r	Real_Out08
45	float	r	Real_Out09
47	float	r	Real_Out10
49	float	r	Real_Out11
4B	float	r	Real_Out12
55	float	r	counter01 (output)
57	float	r	counter02 (output)
5D	float	r	Real_In01
5F	float	r	Real_In02
61	float	r	Real_In03
63	float	r	Real_In04
65	float	r	Real_In05

MODbus address	Data type	Access	Note
67	float	r	Real_In06
69	float	r	Real_In07
6B	float	r	Real_In08
6D	float	r	Real_In09
6F	float	r	Real_In10
71	float	r	Real_In11
73	float	r	Real_In12
75	float	r	Real_In13
77	float	r	Real_In14
79	float	r	Real_In15
7B	float	r	Real_In16
7D	float	r	Real_In17
7F	float	r	Real_In18
81	float	r	Real_In19
83	float	r	Real_In20
85	float	r	Real_In21
87	float	r	Real_In22
89	float	r	Real_In23
8B	float	r	Real_In24

The two input counters (external counters) of the screen recorder cannot be addressed via the communication module.

A

C

D

E

F

I

J

L

M

N

O

P

R

S

T

Index

M. K. JUCHHEIM GmbH & Co

Delivery address: Mackenrodtstraße 14, 36039 Fulda, Germany
Postal address: 36035 Fulda, Germany
Phone: +49 (0) 661 60 03-7 25
Fax: +49 (0) 661 60 03-6 81
E-Mail: mail@jumo.net
Internet: www.jumo.de

JUMO Instrument Co. Ltd.

JUMO House
Temple Bank, Riverway
Harlow, Essex CM20 2TT, UK

Phone: +44 (0) 12 79 63 55 33
Fax: +44 (0) 12 79 63 52 62
E-Mail: info@jumoinstruments.fsnet.co.uk

JUMO PROCESS CONTROL INC.

735 Fox Chase
Coatesville PA 19320, USA
Phone: 610-380-8002
1-800-554-JUMO
Fax: 610-380-8009
E-Mail: info@JumoUSA.com
Internet: www.JumoUSA.com

for stock items see price list

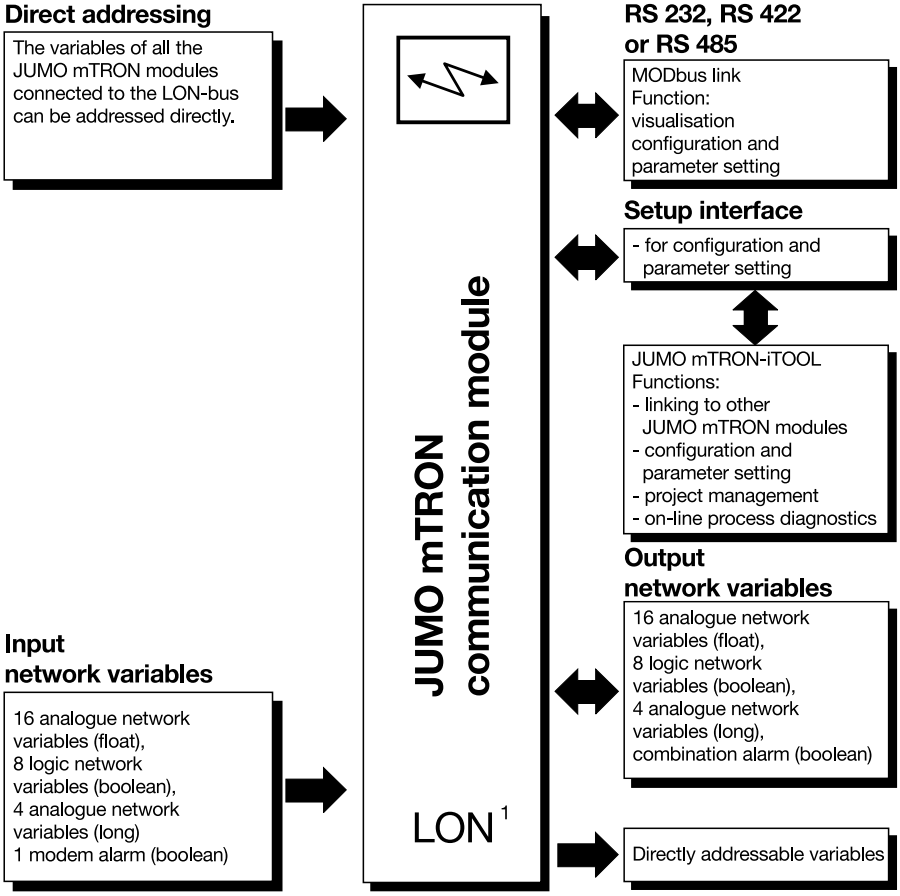
Communication module

Brief description

This unit is a module of the JUMO mTRON control and automation system. The plastic housing measures 91mm x 85.5mm x 73.5mm (W x H x D) and is mounted on a standard rail.
The module is used for communication between the JUMO mTRON modules and higher-level units with MODbus or Jbus interface. The communication module has a LON interface with FTT-10A transceiver for linking to the JUMO mTRON installation and either an RS232, RS422 or RS485 interface for data transmission under the MODbus protocol. A setup interface is provided for parameter setting and configuration of the module via a PC under the JUMO mTRON-iTOOL project design software. The electrical connection is made through plug-in screw terminals.

Type 704040/0-...

Block structure

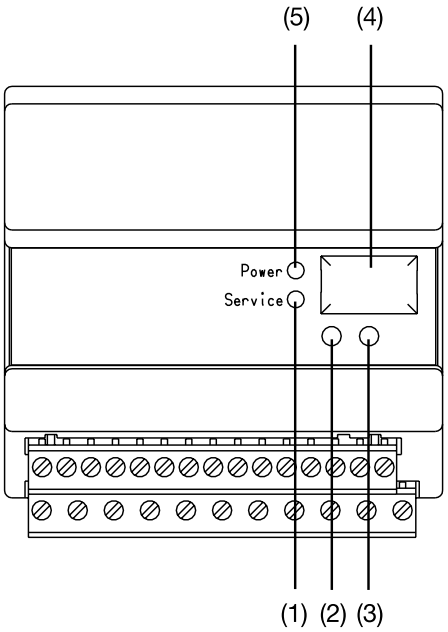


Features

- Visualisation via MODbus
- Connection to PLC via MODbus
- Configuration and parameter setting via JUMO mTRON-iTOOL project design software
- Modem operation for configuration and setting parameters of a JUMO mTRON automation system over any distance
- Several communication modules can operate in a network
- Automatic dialling of a telephone number via modem on alarm in the LON network
- Integral RS232, RS422 or RS485 interface

1. LON® = **L**ocal **O**perating **N**etwork
Registered trademark of the ECHELON Corporation

Displays and controls



(1)	**Service LED, red** - lights up on operating fault - flashes when the mechanical connection to the module from JUMO mTRON-iTOOL or the operating unit is being checked by a test handshake signal (“wink”)	(3)	**Installation key** the module reports to the JUMO mTRON-iTOOL project design software or the operating unit
(2)	**Switch** for the termination resistance of the LON network	(4)	**Setup interface** for the PC interface line which links the module to the PC
		(5)	**Power LED, green** lights up when the supply is switched on

Input network variables

Analogue network variables
– 16 variables “real” type
– 4 variables “long” type

Logic network variables
– 8 variables “bool” type
– 1 modem alarm “bool” type

Function:
They are linked to any network variable of other mTRON modules

Output network variables

Analogue network variables
– 16 variables “real” type
– 4 variables “long” type

Logic network variables
– 8 variables “bool” type

Function:
They can be written as output network variables of the communication module via MODbus.

General data

Environmental conditions to EN 61 010

Operating and ambient temperature:
0 — 55°C
Permitted storage temperature:
-40 to +70°C
Relative humidity: rH 80% max.
Pollution degree 2
Overvoltage category 2

Housing

Material: plastic,
self-extinguishing
Flammability Class: UL 94 VO
Protection: IP20 (to EN 60 529)
Mounting: on standard rail

Supply

110 — 240V AC +10/–15 %, 48 — 63Hz,
or 20 — 53V AC/DC, 48 — 63Hz
Power consumption: 5VA max.

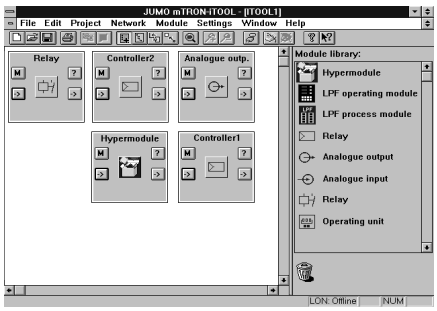
Network (LON interface)

Transceiver: free topology FTT-10A
Topology: ring, star, line or mixed structure
Baud rate: 78 kbaud
Max. lead length
(depending on lead structure):
line: 2700m
star: 500m
ring: 500m
mixed: 500m
Max. number of modules: 64

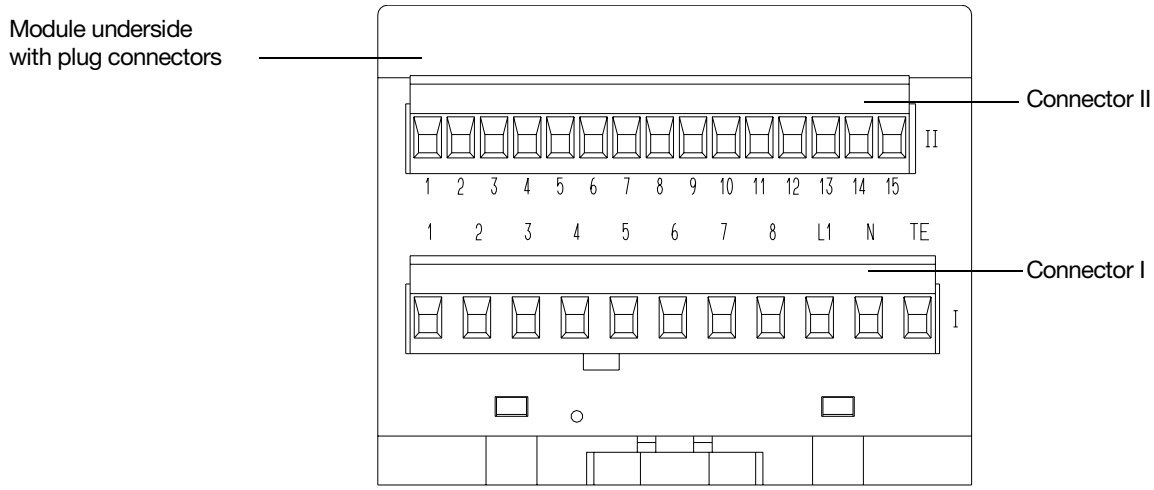
Operation and project design

Operation, parameter setting and configuration of JUMO mTRON modules can be carried out from the JUMO mTRON operating unit.

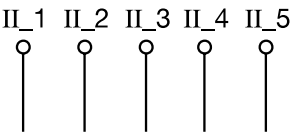
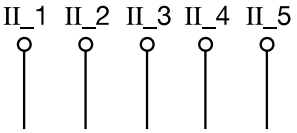
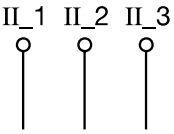
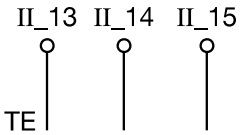
The JUMO mTRON-iTOOL project design software permits convenient design and start-up of a JUMO mTRON system.
The projects can be archived and documented. Individual modules are linked via LON by assigning network variable (NV) names.



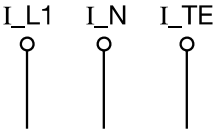
Connection diagram



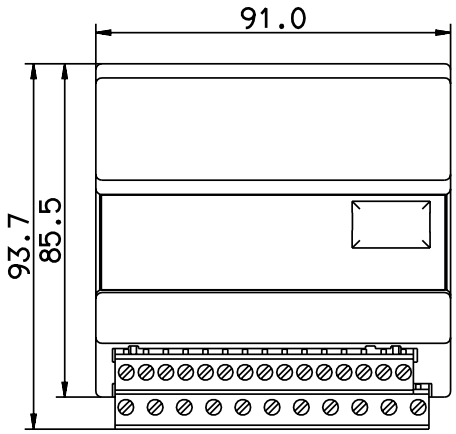
Connector II

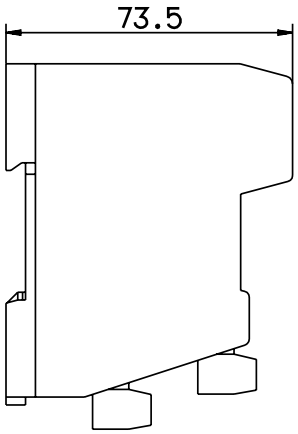
Connection for	Terminals	Notes	Diagram
RS232	II_1 II_2 II_3 II_4 II_5	GND RxD TxD CTS RTS	
RS422	II_1 II_2 II_3 II_4 II_5	GND TxD A TxD B RxD A RxD B	
RS485	II_1 II_2 II_3	GND RxD/TxD A RxD/TxD B	
LON interface	II_13 = TE	screen	
	II_14 = Net_A II_15 = Net_B	any polarity	

Connector I

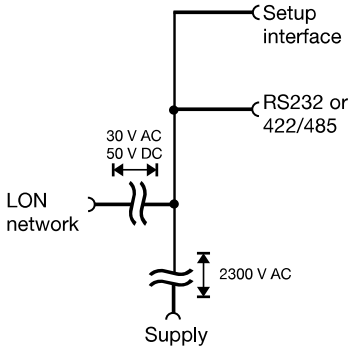
Connection for	Terminals	Notes	Diagram
Supply as label	I_L1 AC I_N I_TE	line neutral technical earth	
	I_L1 DC I_N I_TE	} any polarity technical earth	

Dimensions





Isolation



Ordering details

704040/0- (1) **..** - (2) **..**

(1) Outputs ....

Outputs	Code
Interface RS232	**51**
Interface RS422	**52**
Interface RS485	**53**

(2) Supply ....

Type	Code
110 — 240 V AC +10/-15 %, 48 — 63 Hz	**23**
20 — 53 V AC/DC, 48 — 63 Hz	**22**

Standard accessory

1 Installation Instructions M 70.4040

Accessories

PC interface
with TTL/RS232C converter
for connecting the module to a PC; length 2 m.
Sales No. 70/00301315

Project design software
JUMO mTRON-iTOOL
Using the JUMO mTRON- iTOOL project design software the modules can be designed graphically on the PC. The user is able to link modules of the JUMO mTRON family and to configure the application-specific parameters.

System Manual JUMO mTRON
Documentation of configuration, parameter setting and installation of the modules.
Sales No. 70/00334336

JUMO mTRON modules

Controller module
Data Sheet 70.4010

Relay module
Data Sheet 70.4015

Analogue input module
Data Sheet 70.4020

Analogue output module
Data Sheet 70.4025

Logic module
Data Sheet 70.4030

Operating unit
Data Sheet 70.4035

Communication module
Data Sheet 70.4040

Project design software
JUMO mTRON-iTOOL
Data Sheet 70.4090

Communication module (gateway)

Brief description

The “mTRON to PROFIBUS-DP gateway” is an all-in-one unit with a switched-mode power supply, data transmission module (transceiver), controller (Neuron 3150) and a fieldbus controller in a compact case (with a bracket for mounting on a standard rail, if required). The fieldbus controller acts as a slave to link up with PROFIBUS-DP.
The gateway serves to link an mTRON network with any number of mTRON modules to a PROFIBUS-DP network as a PROFIBUS-DP slave.
The automatic baud rate recognition enables baud rates of 9.6, 19.2, 93.75, 187.5, 500 and 1500 kbps. All PROFIBUS-DP slave services are supported. The software for the fieldbus controller is PNO certified.

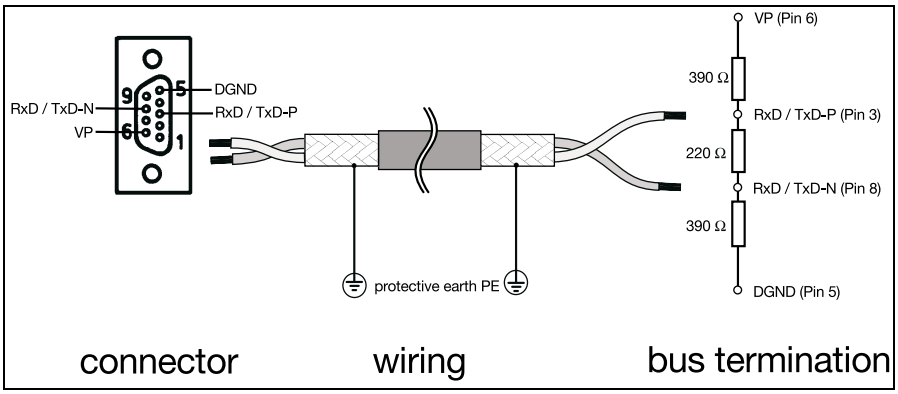
Type 704041/0-64-24

Cabling and bus termination

Supply voltage and LON connection

Pin assignment of the panel-mounting connector	
Pin No.	Function
1	positive supply voltage
2	ground
M/5	shield
3	LON
4	LON

PROFIBUS-DP



Order details

	(1)		(2)
704041/0-	64	-	24

(1) Outputs ....

Outputs	Code
PROFIBUS-DP interface	**64**

(2) Supply ....

Type	Code
18 — 32V DC	**24**

Technical data

Neuron chip	3150 / 10MHz
EEPROM	Flash-EEPROM AT29C512
Supply	18 — 32V DC (unstabilised)
Transceiver	FTT10A
Configuration and data exchange	entirely via network variables
Material	ABS
Dimensions W x H x D	80 x 172 x 43mm