JUMO dTRANS T06

Multifunctional Four-Wire Transmitter in Mounting Rail Case

Interface Description

70707100T92Z001K000

V1.01/EN/00704068

Contents

Contents

General

This manual contains information that must be observed in the interest of your own safety and to avoid material damage. This information is supported by symbols which are used in this manual as indicated.

Please read this manual before starting up the device. Store this manual in a place that is accessible to all users at all times.

If difficulties occur during startup, please do not intervene in any way that could jeopardize your warranty rights!

1.1 Warning symbols

DANGER!

This symbol indicates that **personal injury from electrocution** may occur if the appropriate precautionary measures are not taken.

WARNING!

This symbol in connection with the signal word indicates that **personal injury** may occur if the respective precautionary measures are not carried out.

CAUTION!

This symbol in connection with the signal word indicates that **material damage or data loss** will occur if the respective precautionary measures are not taken.

CAUTION!

This symbol indicates that **components could be destroyed** by electrostatic discharge (ESD = Electro Static Discharge) if the respective cautionary measures are not taken.

Only use the ESD packages intended for this purpose to return device inserts, assembly groups, or assembly components.

READ THE DOCUMENTATION!

This symbol, which is attached to the device, indicates that the associated **documentation for the device** must be **observed**. This is necessary to identify the nature of the potential hazard, and to take measures to prevent it.

1.2 Note symbols

NOTE!

This symbol refers to **important information** about the product, its handling, or additional benefits.

REFERENCE!

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

FURTHER INFORMATION!

This symbol is used in tables and indicates that **further information** is provided after the table.

1 Safety information

DISPOSAL!

At the end of its service life, the device and any batteries present do not belong in the trash! Please ensure that they are **disposed of** properly and in an **environmentally friendly** manner.

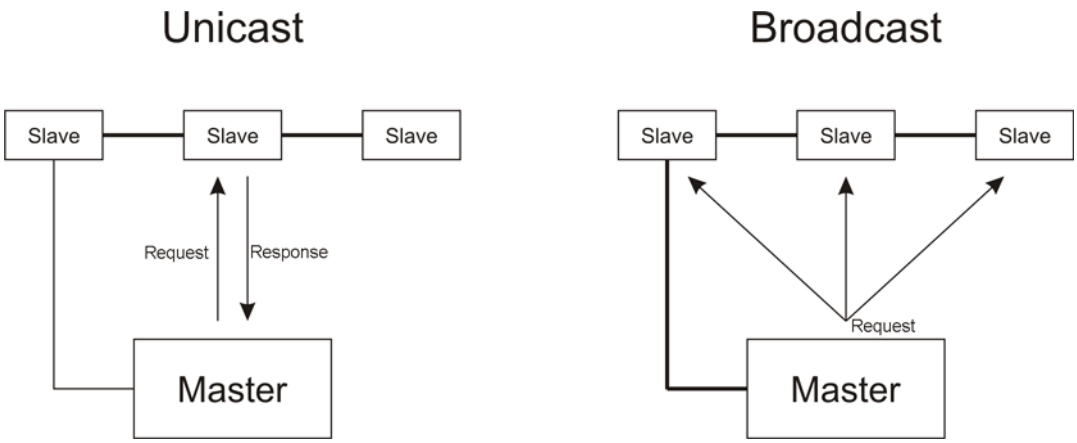
2.1 Master-slave principle

Communication between a master (e.g. a SCADA system or PLC) and a device type 707071 as slave takes place on the basis of the master-slave principle in the form of a data request – response. The bus participants (master and slaves) are addressed as follows depending on the transmission technology.

- For **Modbus via a serial interface**, all slaves are identified by their device address (1 to 254). Master devices do not need an address.

⇨ chapter 4 "Modbus over a serial interface", page 19

The master controls data exchange by cyclically requesting the slaves in the overall bus. The slaves (for example, device type 707071) have a response function only. The master has write and read access to the slaves. This enables data to be communicated between master and slave devices in real-time. Slaves cannot communicate with one another directly. In order to transmit data from slave to slave, the master must extract data from one slave and transfer it to the next.



As a rule, the master directs its requests specifically to individual slaves. For this, it must communicate with the individual slaves by means of their unicast address. Requests can also be directed to all slaves on the bus in the form of a broadcast. In this case, the broadcast address is used as the slave address. To avoid data collisions, slaves do not respond to broadcast requests. For this reason, function codes are only used for writing data. Broadcasts cannot be used with function codes for reading data.

2.2 Transmission media for Modbus

Serial interface

For data communication via a serial interface, the Modbus specification provides the transmission modes with RTU mode (Remote Terminal Unit) and ASCII mode (transmission of data in ASCII format). Device type 707071 supports the RTU mode only. Here, the data is transmitted via the serial bus in binary format (RS485).

⇨ chapter 4 "Modbus over a serial interface", page 19

2 Modbus protocol description

2.3 Structure of a Modbus telegram

Data structure

All telegrams have the same structure:

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

Each telegram has four fields:

Slave address Device address of a specific slave

Function code Function selection (read/write words)

Data field Contains information (according to the function code)

- Word address/bit address
- Number of words/bits
- Word value(s)/bit value(s)

Checksum Detection of transmission errors

2.4 Function codes

Function overview

The Modbus standard functions described in the following are available for the extraction of measured values, device data, and process data, as well as for writing specific data.

Function code		Function	Limit
Hex.	Dec.		
03 or 04	3 or 4	Reading n words	Max. 26 words (52 bytes)
06	6	Writing one word	Max. 1 word (2 bytes)

NOTE!

If the device does not respond to these functions (see requirements for communication) or outputs an error code (see error codes): the information in chapter 2.7.1 "Modbus error codes", page 14 may help during troubleshooting.

2.4.1 Reading n words

This function is used to read n words starting at a specific address.

Data request

Slave address	Function 0x03 or 0x04	Address of first word	Number of words	Checksum CRC
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 CRC
1 byte	1 byte	1 byte	x bytes	2 bytes

Example

The example deals with reading the installation date (day, month, and year). Starting from address 0x0212, it is stored in two words.

⇨ chapter 5 "Modbus address tables", page 21

Hex code for the data request:

01	03	0212	00 02	65 B6
Slave	Function code	Address of 1st word	Number of words	CRC

Hex code for the response:

01	03	04	1E 08	14 11	B2 D5
Slave	Function	Byte read	30. 08.	20 17	CRC
			Month/day	Year	

2.4.2 Writing one word

The data blocks for the data request and response are identical when writing a word.

Data request

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

Response

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

Example

In this example, the drag indicator for the minimum measured value should be reset.
The slave address of the device is 1 here, the word address is 0x0FFF and the value to be written should be the value 10d = 0x0A = service, resetting min. measured value.

⇨ chapter 5.2.5 "Services/commands", page 25

Hex code for the data request:

01	06	0F FF	00 0A	3A E9
Slave	Function code	Word address	Value	CRC

Hex code for the response:

01	06	0F FF	00 0A	3A E9
Slave	Function code	Word address	Value	CRC

2 Modbus protocol description

2.5 Examples of data transmission

The function 0x03 (reading) or 0x04 (reading one word) is used to extract integers, floating-point values, and text values.

Data request

Slave address	Function code 0x03 or 0x04	Address of first word	Number of words	Checksum CRC
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Integer values are transmitted over Modbus in the following format:
First the high, then the low byte.

Response

Slave address	Function code 0x03 or 0x04	Number gelesener Byte	Word value(s)	Checksum CRC
1 byte	1 byte	1 byte	x bytes	2 bytes

2.5.1 Integer values

Example

In this example, the signal type of the analog output at address 0x006B is to be read.

Data request:

01	03	00 6B	00 01	F5 D6
Slave	Function code	Address of 1st word	Number of words	CRC

Response (values in Modbus integer format):

01	03	02	00 00	B8 44
Slave	Function code	Byte read	Integer value	CRC

2.5.2 Floating-point values

For floating-point values, the device type 707071 functions 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 floating-point format (32-bit) according to standard IEEE 754

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

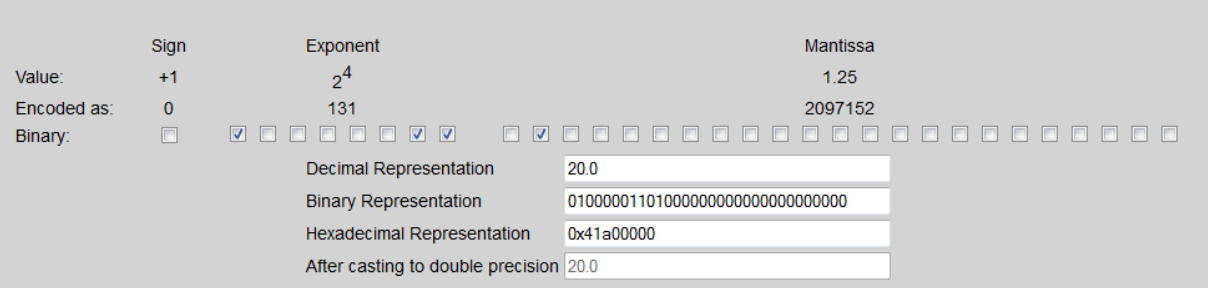
S - Prefix sign bit
E - Exponent (two's complement)
M - 23-bit normalized mantissa

Modbus floating-point format

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

Example

In this example, the value of the analog output at the address is to be extracted. The value here should be 20.0 mA (0x41a00000 in IEEE 754 format).



Data request:

01	03	00 67	00 02	75 D4
Slave	Function code	Address of 1st word	Number of words	CRC

Response (values in Modbus floating-point format):

01	03	04	00 00	41 A0	CA 1B
Slave	Function code	Byte read	Floating-point value in the Modbus floating-point format		CRC

After being transmitted from the device, the bytes for the floating-point value must be interchanged accordingly. Many compilers (for example, Microsoft Visual C++) store the floating-point values in the following sequence:

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

NOTE!

The sequence of the bytes depends on how floating-point values are saved in the application concerned. It may be necessary for the bytes to be interchanged in the interface program accordingly.

2.5.3 Character strings (texts)

Character strings are transmitted in ASCII format.

NOTE!

The maximum lengths specified in the address tables for character strings also include the terminating "/0". This means that, in the case of "Char 20", the text must not be longer than 20 bytes.

Example

Requesting the text from address 0x0208; this address holds the character string for the information text "Anlage Nord" (ASCII code: 0x41 0x6E 0x6C 0x61 0x67 0x65 0x20 0x4E 0x6F 0x72 0x64).

Hex code for the request:

01	03	10 00	00 06	45 B2
Slave	Function code	Address of 1st word	Number of words	CRC

Hex code for the response:

			Word 1		Word 2		Word 3		Word 4		Word 5		Word 6		
01	03	00 0C	41	6E	6C	61	67	65	20	4E	6F	72	64	00	49 1F
Slave	Function code	Bytes read	A	n	l	a	g	e	space	N	o	r	d		CRC

2.6 Checksum (CRC16)

Transmission errors are detected with the aid of the checksum (CRC16). If an error is detected during evaluation, the device concerned does not respond.

Calculation principle

CRC = 0xFFFF			
	CRC = CRC XOR BytesOfMessage		
	For (1 to 8)		
		CRC = SHR(CRC)	
		if (flag shifted to the right = 1)	
		then CRC = CRC XOR 0xA0001	else
while (not all BytesOfMessage processed);			

NOTE!

The low byte of the checksum is transmitted first!
Example: The CRC16 checksum CC DD is transmitted and represented in the sequence DD CC.

2 Modbus protocol description

2.7 Error messages

2.7.1 Modbus error codes

Requirements for Modbus communication

The following conditions must be met for a slave to receive, process, and respond to requests:

- Baud rate and data format of master and slave must match.
- The correct slave address must be used in the request.
- Slave devices respond only after a successful checksum check of the request by the slave. Otherwise, the request is rejected by the slave.
- The data request from the master must be complete and conform to the Modbus protocol.
- The number of words to be read must be greater than 0.

Error codes

If the data request from the master has been received by the slave without transmission errors but cannot be processed, the slave responds with an error code. The following error codes may occur:

- 01 = Invalid function; for the function codes that device type 707071 supports, see chapter 2.4 "Function codes", page 8
- 02 = Invalid address or too many words, or bit should be read or written
- 03 = Value is outside the admissible range
- 08 = Value is write-protected

Response in case of error

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

The function code is ORed with 0x80. As a result, the highest value bit (msb) is set to 1.

Example

Data request:

01	06	00 69	00 01	98 16
Slave	Write word	Word address	Word value	CRC

Response (with error code 8):

01	86	08	43 A6
Slave	OR function	Error	CRC

Response with error code 08, because address 0x0069 can only be read.

2.7.2 Error messages for invalid values

For measured values in the floating-point format, the error is displayed in the value itself, i.e. it contains the error code instead of the measured value.

Error code for floating-point values	Error
1.0×10^{37}	Measuring range underflow
2.0×10^{37}	Measuring range overflow
3.0×10^{37}	Not a valid input value
4.0×10^{37}	Division by zero
5.0×10^{37}	Mathematical error
6.0×10^{37}	Invalid terminal temperature of thermocouple
7.0×10^{37}	Probe short circuit
8.0×10^{37}	Probe break
9.0×10^{37}	Timeout during value determination
10.0×10^{37}	No value received

Example

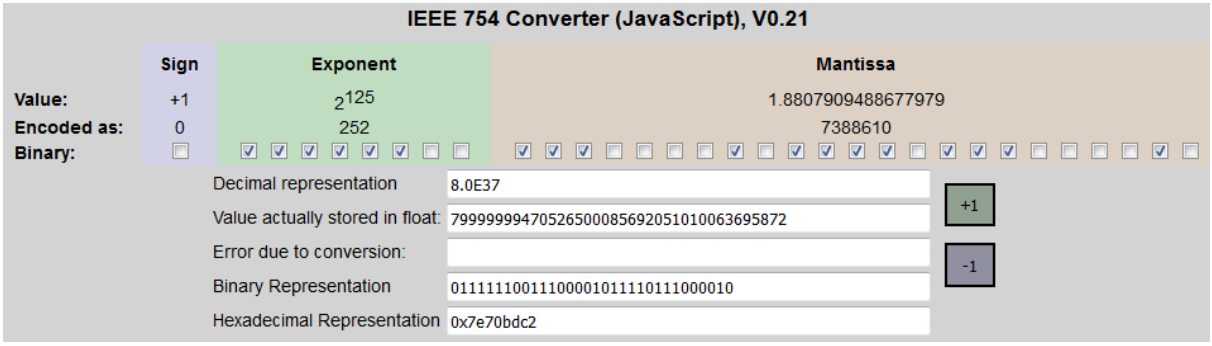
Loading of the measured value at Modbus address 0x0064: data request:

01	03	00 64	00 02	85 D4
Slave	Function code	Word address	Number of words	CRC

Response:

01	03	04	BD C2	7E 70	C8 ED
Slave	Function code	Bytes read		Error code	CRC

The supplied measured value 0x7E70BDC2 (= 8.0×10^{37}) indicates that this is a probe break.



2 Modbus protocol description

3.1 Connection of the interface

The device optionally features an RS485 interface. This is intended for transmission using the Modbus protocol (slave) and must be specified during ordering.

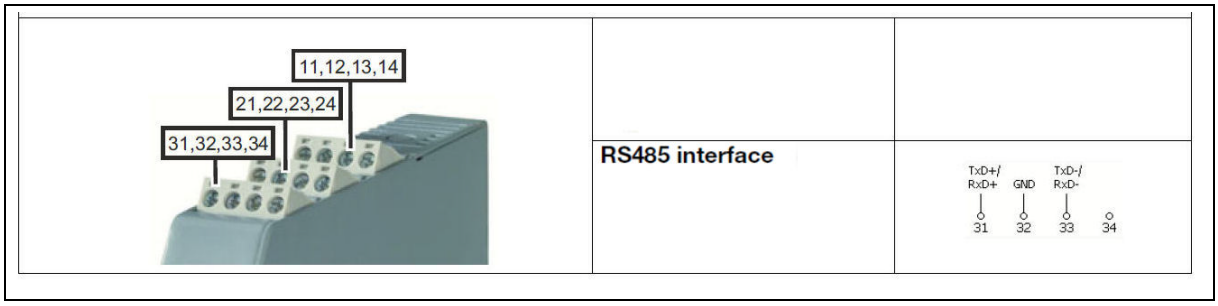
Retrofitting is not possible!

NOTE!

The device's nameplate provides information on which optional interfaces were assembled **when delivered**.
Information on this can be found in the chapter "Identifying the device version" in the operating manual 70707100T90Z000K000 or the data sheet 70707100T10Z000K000.

Overview of connections

JUMO dTRANS T06 LCD (707071)



NOTE!

A twisted connecting cable with shielding must be used to connect the RS485 interface; to the greatest extent possible, it should not be laid close to current-carrying components or cables and its shielding should be grounded on one side.
To avoid transmission errors, only the signals listed above may be connected.

3.1.1 Terminating resistors

NOTE!

To ensure fault-free operation, terminating resistors are required at the beginning and end of an RS485 transmission path. With devices that are located at the start or end of a bus system, the terminating resistors have to be activated.

CAUTION!

A high-resistance terminating resistor (in the kilo-ohm range) integrated in the device type 707071 is sufficient for short, interference-free cable lengths.

▶ The bus terminations must be realized with other devices in the event of malfunctions or line reflections.

3.2 Configuring interfaces

CAUTION!

The interface restarts after each change to the configuration.
An existing connection could be disrupted or interrupted by this.

For this reason, configuration changes must never be made while a system is operating.

3 Interfaces

CAUTION!

Incorrect installation or the wrong settings on the device can cause the system to assume an unexpected
operating status.

This can disrupt processes or result in damage. For this reason, only allow settings to be made by qualified personnel.

Operation, configuration, and parameterization procedures are explained in detail in the operating manual for device 707071ooT90Z00K00.

Settings for serial interfaces

For all user devices on a bus to be able to communicate with one another, their interface settings must match. The table below shows the setting options for the serial interfaces on device type 707071.

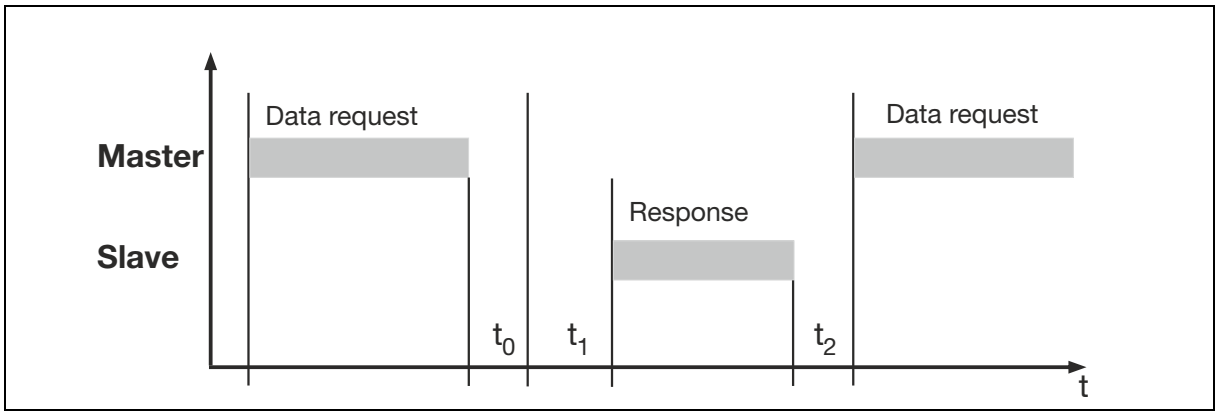
Open:
Device menu > Configuration > Interface >

Configuration item	Selection/settings	Description
Baud rate	9600 19200 38400 57600 115200	Transmission speed (symbol rate) of the serial interface
Data format	8 - 1 - no parity 8 - 1- odd Parity 8 - 1- even Parity 8 - 2- no Parity	Format of the data word Useful bit - stop bit - parity
Device address	1 to 254	Unambiguous specification of a bus participant 1 to 254

4.1 Modbus timing for serial interface RS485

Chronological sequence of communication

A scanning cycle on a bus proceeds with the following timing:



t_0 Final identification, internal waiting period of the device before checking the data request: 10 to 15 ms

t_1 Internal processing time: max. 100 ms

t_2 The master has to observe this waiting period before starting a new data request: min. 10 ms

NOTE!

During t_1 and t_2 and during the response time of the slave, no data requests may be generated by the master. Requests during t_1 and t_2 are ignored by the slave. Requests during the response time invalidate all the data currently on the bus.

Character transmission time

The beginning and end of a data block are identified by pauses in transmission. The character transmission time (time required to transmit one single character) depends on the baud rate and the data format used.

The result with 1 start bit, 8 data bits, 0 parity bits, and 1 stop bit is:

Character transmission time [ms] = 1000 × 10 bits / baud rate

For other data formats, the following is the result:

Character transmission time [ms] = 1000 × 11 bits / baud rate

Example

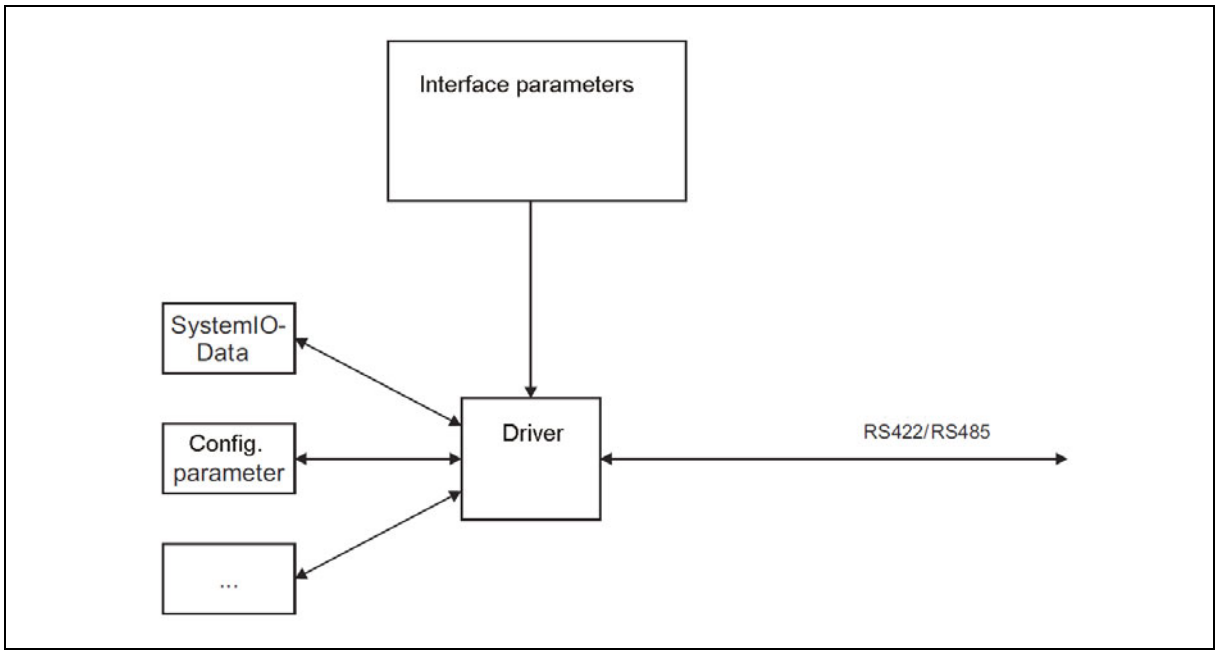
Identifier for end of data request or response for data format 11/10 bits

Waiting period = 3 characters * 1000 * 11 bits/baud rate

Baud rate [Baud]	**Data format** [Bit]	**Character transmission time** [ms]
38400	11	0.286
	10	0.260
19200	11	0.573
	10	0.521
9600	11	1.146
	10	1.042

4 Modbus over a serial interface

4.2 Modbus slave communication over an RS422/485 serial interface



The device type 707071 functions as a slave and responds in the network to Modbus requests from the master.
The master controls the data exchange and the slaves only have a response function. They are identified by their device address.

The master is usually a PC with a setup or visualization program or another Modbus-capable device. The master can request all device variables according to the Modbus address tables of this slave.

⇨ chapter 5 "Modbus address tables", page 21

The tables in this chapter list process and device data for the device type 707071 with their Modbus address, the data type, and the possible access options (Modbus function codes).

5.1 Data types and access types

Data types

<table>
<tr><th>Data type</th><th>Description</th><th>Number of Modbus registers</th></tr>
<tr><td>BOOL</td><td>Least significant bit in a word as bit value
0000 0000 0000 0001 = 1 or TRUE
0000 0000 0000 0000 = 0 or FALSE</td><td>1</td></tr>
<tr><td>UBYTE</td><td>1 byte = 8 bits; can be read or written as word
Value range: 0 to 255</td><td>1</td></tr>
<tr><td>ENUM</td><td>Enumeration type as an integer value</td><td>1</td></tr>
<tr><td>USHORT16</td><td>Integer value
Unsigned short = 1 word = 2 bytes = 16 bits
Value range: 0 to 65535</td><td>1</td></tr>
<tr><td>FLOAT</td><td>2 words as 32-bit floating-point number with coding according to IEEE 754, whereby it must be kept in mind that bytes 1 and 2 are interchanged with bytes 3 and 4 during transmission
S = Prefix sign bit
E = Exponent (two's complement)
M = 23-bit normalized mantissa
IEEE 754 standard coding:
<table>
<tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td></tr>
<tr><td>SEEEEEEE</td><td>EMMMMMMM</td><td>MMMMMMMM</td><td>MMMMMMMM</td></tr>
</table>
Modbus coding of floating-point variables in device type 707071:
<table>
<tr><td colspan="2">Address of the first Modbus variable register</td><td colspan="2">Address of the second Modbus variable register</td></tr>
<tr><td>Byte 3</td><td>Byte 4</td><td>Byte 1</td><td>Byte 2</td></tr>
<tr><td>MMMMMMMM</td><td>MMMMMMMM</td><td>SEEEEEEE</td><td>EMMMMMMM</td></tr>
</table>
When creating customer-specific applications, the correct byte sequence in storage format needs to be checked. Many compilers use the following storage format:
<table>
<tr><td>Byte 4</td><td>Byte 3</td><td>Byte 2</td><td>Byte 1</td></tr>
<tr><td>MMMMMMMM</td><td>MMMMMMMM</td><td>EMMMMMMM</td><td>SEEEEEEE</td></tr>
<tr><td>Address x</td><td>Address x+1</td><td>Address x+2</td><td>Address x+3</td></tr>
</table></td><td>2</td></tr>
<tr><td>ULONG32</td><td>Unsigned integer value
Unsigned long (4 bytes) = 32 bits = 2 words
Value range: 0 to 4.294.967.295</td><td>2</td></tr>
<tr><td>CHAR [12]</td><td>Character string, e.g. for up to 12 ASCII characters
Note that the character string always has "\0" (ASCII code 0x00) as the terminating code.</td><td>6</td></tr>
<tr><td>BITFELD32</td><td>Bit field 32 bits long
Bit 0 is the bit with the lowest value</td><td>1</td></tr>
</table>

5 Modbus address tables

Access types

R/O	Read only
W/O	Write only
R/W	Read/write

5.2 Process values, configuration data, and parameters

5.2.1 Device name

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x0000 ... 0x0005	0 to 5	Device name	CHAR [12]	R/O

5.2.2 Version data

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x0006 ... 0x000B	6 to 11	Software version, diagnostics channel	CHAR [12]	R/O
0x000C ... 0x0012	12 to 17	Extra version, diagnostics channel	CHAR [14]	R/O
0x0013	19	Internal version counter, diagnostics channel	USHORT16	R/O
0x0024 ... 0x0029	36 to 41	Software version, safety channel	CHAR [12]	R/O
0x002A ... 0x0030	42 to 48	Extra version, safety channel	CHAR [14]	R/O
0x0031	49	Internal version counter, safety channel	USHORT16	R/O

5.2.3 Process values

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x0064	100	Measured value [°C]	FLOAT	R/O
0x0066	102	Measured value [display unit]	FLOAT	R/O
0x0068	104	Display unit 0 = °C 1 = °F 2 = text	ENUM	R/O
0x0069	105	Analog output value [mA/V] (for signal type see next parameter)	FLOAT	R/O
0x006B	107	Signal type, analog output 0 = current [mA] 1 = voltage [V]	ENUM	R/O

Modbus address		Signal designation	Data type	Access
Hex.	Dec.			
0x006C	108	Switching cycles	UINT32	R/O
0x0070	112	Minimal measured value	FLOAT	R/O
0x0072	114	Maximum measured value	FLOAT	R/O
0x0074	116	Time since occurrence of min. measured value [s]	UINT32	R/O
0x0076	118	Time since occurrence of max. measured value [s]	UINT32	R/O
0x0078	120	Operation time [s]	UINT32	R/O
0x007A	122	Time since last change of configuration	UINT32	R/O
0x0080	128	Duration of operation time at > 70 °C [s]	UINT32	R/O
0x0082	130	Duration of operation time at < -10 °C [s]	UINT32	R/O
0x0084	132	Duration of operation time at -10 to +70 °C [s]	UINT32	R/O
0x0086	134	Terminal temperature [°C]	FLOAT	R/O
0x008C	140	Status, diagnostics channel Bit Meaning 0 Error detected when measuring the 3 V voltage supply 1 Error detected when measuring the 5 V voltage supply 2 Analog output deviates from the specification of the safety channel 3 Relay setting deviates from the specification of the safety channel 4 Error during measurement of the analog output voltage supply 5 SW versions, safety channel and diagnostics channel do not fit 6 Communication between safety channel and diagnostics channel faulty 7 An error has occurred during editing 8 The set relay switching cycles have been exceeded 9 Configuration faulty 10 ROM defective 11 RAM defective 12 Parameter faulty 13 CRC test of configuration in RAM returned errors 14 CRC test of configuration in EEPROM returned errors 15 Program sequence faulty 16 Error when writing/reading the EEPROM 17 Error during communication via USB 18 Error during RS485 communication 19 LCD controller not supported	BITFELD32	R/O

Modbus address Hex.	Modbus address Dec.	Signal designation	Data type	Access
0x0090	144	Status, safety channel Value Meaning 1 Not used 2 Terminal temperature outside the limits or probe faulty 3 Deviation during the reference measurement of the A/D converter 4 Calibration constants faulty 5 Configuration faulty 6 Measured value errors are listed separately, see table in chapter 2.7.2 "Error messages for invalid values", page 15 7 Not used 8 Not used 9 CRC test of calibration returned errors 10 CRC test of configuration returned errors 11 CPU register faulty 12 RAM defective 13 ROM defective 14 Program sequence faulty 15 Watchdog was activated 16 Voltage supply insufficient 17 Deviation of the frequency 18 EEPROM defective 19 Error in the stack 20 Error in the A/D converter 21 Not used 22 Interrupt error 23 Not used 24 Limit value exceeded or fallen short of	ENUM	R/O
0x0091	145	Alarm/relay status active	BOOL	R/O
0x0092	146	Relay option available	BOOL	R/O
0x0093	147	RS485 option available	BOOL	R/O
0x0095	149	SIL mode active	BOOL	R/O

5.2.4 Texts

Modbus address Hex.	Modbus address Dec.	Signal designation	Data type	Access
0x01F4	500	Inspection ID	CHAR[10]	R/O
0x01F9	505	Fabrication number	CHAR[20]	R/O
0x0203	515	TAG number text	CHAR[10]	R/O
0x0208	520	Information text	CHAR[20]	R/O
0x0212	530	Installation date (byte 1: day, byte 2: month, byte 3/4: year)	BYTE[4]	R/O
0x0214	532	Units text	CHAR[10]	R/O

5.2.5 Services/commands

This service only works using the function code 6=writing one word.

Modbus address		Signal designation		Data type	Access
Hex.	Dec.				
0x0FFF	4095	Value	Meaning	USHORT16	W/O
		10	Reset of the min. measured value		
		20	Reset of the max. measured value		

5 Modbus 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
Email: 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 62 50 29
Email: sales@jumo.co.uk
Internet: www.jumo.co.uk

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