

JUMO tecLine...

... Cl₂ (202630), TC (202631), ClO₂+O₃ (202634),
H₂O₂+PAA (202636), Br (202637), Cl₂ OM (202681)



Schnittstellenbeschreibung Modbus



20263000T92Z000K000

V2.00/DE/00718249/2022-03-24

1	Sicherheitshinweise	5
1.1	Warnende Zeichen	5
1.2	Hinweisende Zeichen	5
2	Modbus-Protokollbeschreibung	6
2.1	Master-Slave-Prinzip	6
2.2	Übertragungsmedien für Modbus	7
2.3	Aufbau eines RTU-Modbus-Telegramms	7
2.4	Funktionscodes	7
2.4.1	Lesen von n Worten	7
2.4.2	Schreiben eines Wortes	8
2.4.3	Schreiben von n Worten	9
2.5	Datentypen	10
2.6	Beispiele für die Übertragung von Daten	11
2.6.1	Integer-Werte (16 Bit)	11
2.6.2	Long Integer-Werte (32 Bit)	11
2.6.3	Float-Werte	12
2.7	Checksumme (CRC16)	13
2.8	Fehlermeldungen	14
2.8.1	Modbus-Fehlercodes	14
3	Modbus über serielle Schnittstelle	15
3.1	Modbus-Slave-Betrieb über serielle Schnittstelle RS 485	15
4	Schnittstellen	16
4.1	Schnittstellenbelegung der JUMO tecLine Sensoren (20263x)	16
5	Schnittstellen konfigurieren	17
5.1	Einstellungen für die serielle Schnittstelle	17
6	Modbus-Adresstabellen	18
7	Anhang	20
7.1	Ergänzende Informationen	20
7.2	Funktionsweise der Historie	20
7.3	Kalibrierung der Sensorsteilheit	21
7.3.1	Überprüfung der Sensorsteilheit an der Messstelle	21
7.3.2	Speichern der Kalibrierdaten im Sensor	21
7.3.3	Überprüfung der Kalibrierung	22
7.3.4	Regelmäßige Überprüfung	22

Inhalt

7.4	Kalibrierung des Sensor-Nullpunktes	23
7.4.1	Messung des Zellenstromes bei Abwesenheit von Desinfektionsmittel	23
7.5	Wiederherstellung der Nennsteilheit.	24
7.5.1	Einstellen der Standardwerte für Nennsteilheit und Nullpunkt	24
7.5.2	Überprüfung des übertragenen Wertes für die Sensorsteilheit.	24

1.1 Warnende Zeichen



VORSICHT!

Dieses Zeichen in Verbindung mit dem Signalwort weist darauf hin, dass ein **Sachschaden oder ein Datenverlust** auftritt, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



DOKUMENTATION LESEN!

Dieses Zeichen – angebracht auf dem Gerät – weist darauf hin, dass die zugehörige **Gerätedokumentation** zu **beachten** ist. Dies ist erforderlich, um die Art der potenziellen Gefährdung zu erkennen und Maßnahmen zu deren Vermeidung zu ergreifen.

1.2 Hinweisende Zeichen



HINWEIS!

Dieses Zeichen weist auf eine **wichtige Information** über das Produkt oder dessen Handhabung oder Zusatznutzen hin.



VERWEIS!

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



WEITERE INFORMATION!

Dieses Zeichen wird in Tabellen verwendet und weist auf **weitere Informationen** im Anschluss an die Tabelle hin.



ENTSORGUNG!

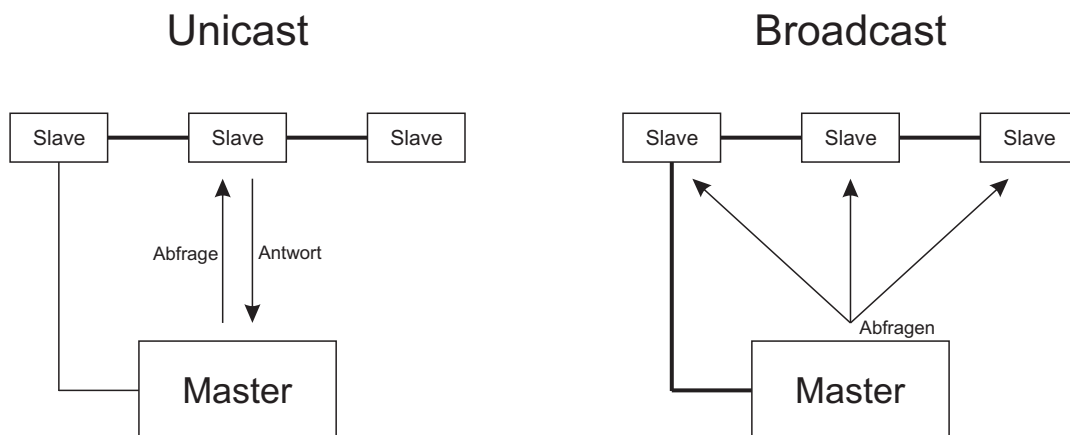
Dieses Gerät und, falls vorhanden, Batterien gehören nach Beendigung der Nutzung nicht in die Mülltonne! Bitte lassen Sie sie ordnungsgemäß und **umweltschonend entsorgen**.

2 Modbus-Protokollbeschreibung

2.1 Master-Slave-Prinzip

Die Kommunikation zwischen einem Master (z. B. SCADA-System oder SPS) und einem JUMO tecLine-Sensor als Slave in einem Modbus findet nach dem Master-Slave-Prinzip in Form von Datenanfrage/Anweisung - Antwort statt. Slaves werden anhand ihrer Geräteadresse identifiziert. Master-Geräte benötigen keine Adresse.

⇒ Kapitel 3 „Modbus über serielle Schnittstelle“, Seite 15



Der Master steuert den Datenaustausch durch zyklische Anfragen an die Slaves im gesamten Bus. Die Slaves (z. B. JUMO tecLine-Sensoren) haben lediglich Antwortfunktion. Der Master kann dabei schreibend und lesend auf die Slaves zugreifen. Auf diese Weise können Daten in Echtzeit zwischen Master und Slave-Geräten kommuniziert werden. Slaves können nicht direkt miteinander kommunizieren. Um Daten von Slave zu Slave zu übertragen, muss der Master die Daten aus dem einen Slave auslesen und dann an den nächsten übertragen.

In der Regel richtet der Master seine Anfragen gezielt an einzelne Slaves. Dazu muss er die jeweiligen Slaves mit ihrer Unicast-Adresse ansprechen. Anfragen können aber auch als Rundsendungsnachricht an alle Slaves im Bus gerichtet werden. Hierfür wird als Slave-Adresse die Broadcast-Adresse verwendet. Broadcast-Anfragen werden von den Slaves nicht beantwortet. In seriellen Bussystemen würden sonst Datenkollisionen entstehen. Aus diesem Grund ist die Verwendung von Broadcast-Adressen nur mit Funktionscodes zum Schreiben von Daten sinnvoll. Broadcasts können nicht mit Funktionscodes zum Lesen von Daten verwendet werden.

2 Modbus-Protokollbeschreibung

2.2 Übertragungsmedien für Modbus

Serielle Schnittstelle

Die Modbus-Spezifikation sieht für die Datenkommunikation über **serielle Schnittstelle** die Übertragungsmodi **RTU-Modus (Remote Terminal Unit)** und ASCII-Modus (Übertragung der Daten im ASCII-Format) vor. JUMO tecLine-Sensoren unterstützen nur den **RTU-Modus**. Hierbei werden die Daten im Binärformat über den seriellen Bus (RS485) übertragen.

⇒ Kapitel 3 „Modbus über serielle Schnittstelle“, Seite 15

2.3 Aufbau eines RTU-Modbus-Telegramms

Modbus-Telegramme sind nach folgendem Muster aufgebaut:

Slave-Adresse	Funktionscode	Datenfeld	Checksumme CRC
1 Byte	1 Byte	n Bytes	2 Bytes

Jedes Telegramm enthält vier Felder:

Slave-Adresse	Geräteadresse eines bestimmten Slaves
Funktionscode	Funktionsauswahl (Lesen/Schreiben von Worten)
Datenfeld	Enthält die Informationen (je nach Funktionscode) - Wortadresse/Bitadresse - Wortanzahl/Bitanzahl - Wortwert(e)/Bitwert(e)
Checksumme	Erkennung von Übertragungsfehlern

2.4 Funktionscodes

Funktionsübersicht

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

Funktionscode		Funktion	Begrenzung
Hex	Dez.		
03 oder 04	3 oder 4	Lesen von n Worten	Max. 125 Worte (250 Bytes)
06	6	Schreiben eines Wortes	Max. 1 Wort (2 Bytes)
10	16	Schreiben von n Worten	Max. 125 Worte (250 Bytes)

2.4.1 Lesen von n Worten

Mit dieser Funktion werden n Worte ab einer bestimmten Adresse gelesen.

Datenanfrage

Slave-Adresse	Funktion 0x03	Adresse erstes Wort	Wortanzahl n	Checksumme CRC
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

Antwort

Slave-Adresse	Funktion 0x03	Anzahl gelesener Bytes	Wortwert(e)	Checksumme CRC
1 Byte	1 Byte	1 Byte	n × 2 Bytes	2 Bytes

2 Modbus-Protokollbeschreibung

Beispiel

Lesen des Temperaturmesswertes. Es handelt sich hier im Beispiel um den Temperaturmesswert (im Beispiel 24,091 °C), welcher in allen JUMO tecLine-Sensoren 20263x an der Startadresse 0x0004 ausgelesen werden kann. Als Fließkommazahl (float) weist der Wert eine Länge von 4 Bytes auf. Es müssen also zwei Worte (Modbus-Register) gelesen werden (vgl. „Prozessdaten Messwerte“, Seite 19).

Hex-Code der Datenanfrage:

01	03	00 04	00 02	85 CA
Slave	Funktion	Adresse 1. Wort	Wortanzahl	CRC

Hex-Code der Antwort (Werte im Byte-Format):

01	03	04	BA 2F 41 C0	DE E2
Slave	Funktion	Bytes gelesen	Temperaturwert 24,091 °C (float)	CRC

2.4.2 Schreiben eines Wortes

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



VORSICHT!

Schreiboperationen auf manche R/W-Parameter bewirken ein Abspeichern im EEPROM oder Flash-Speicher. Diese Speicherbausteine haben nur eine begrenzte Anzahl von Schreibzyklen (ca. 100.000 bzw. 10.000).

Häufiges Beschreiben entsprechender Variablen kann daher dazu führen, dass ein Speicherfehler auftritt.

- ▶ Die Anzahl der Schreibvorgänge sollte daher möglichst klein gehalten werden.

Anweisung

Slave-Adresse	Funktion 0x06	Wort-Adresse	Wort-Wert	Checksumme CRC
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

Antwort

Slave-Adresse	Funktion 0x06	Wort-Adresse	Wort-Wert	Checksumme CRC
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

Beispiel

In diesem Beispiel soll ein Kommando für die Einstellung der Baudrate auf 38400 Bd an einen JUMO tecLine CIO2 (202634) gesendet werden. Die Slave-Adresse des Gerätes ist hier 1, die Wortadresse ist 0x0401 (⇒ „Gerätedaten Modbus“, Seite 18) und der zu schreibende Wert ist „4“ (Wert für das Setzen der Baudrate auf 38400 Bd).

Hex-Code der Anweisung:

01	06	04 01	00 04	D8 F9
Slave	Funktion	Wortadresse	Wert	CRC

Hex-Code der Antwort:

01	06	04 01	00 04	D8 F9
Slave	Funktion	Wortadresse	Wert	CRC

2 Modbus-Protokollbeschreibung

2.4.3 Schreiben von n Worten

Mit dieser Funktion werden n Worte ab einer bestimmten Adresse geschrieben.

Anweisung

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortanzahl n	Byte-Anzahl n × 2	n Wort- wert(e)	Check- summe CRC
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	n × 1 Byte	2 Bytes

Antwort

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortanzahl n	Checksumme CRC
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

Beispiel

Schreiben der Sensorsteilheit 153 nA/ppm an die Modbus-Adresse 0x0208 bei einem JUMO tecLine H2O2+O3 (202634).

⇒ „Prozessdaten Parameter“, Seite 19

Hex-Code der Anweisung:

01	10	02 08	00 02	04	00 00 43 19	1B 93
Slave	Funktion	Adresse 1. Wort	Wortan- zahl	Byte-An- zahl	Steilheit 153 nA/ppm (float)	CRC

Hex-Code der Antwort:

01	10	02 08	00 02	C1 B2
Slave	Funktion	Adresse 1. Wort	Wortanzahl	CRC

2 Modbus-Protokollbeschreibung

2.5 Datentypen

Datentyp	Beschreibung	Zugriff	Mögliche Funktionscodes	Anzahl Modbus-Register																																
Float	<p>2 Wörter als 32-Bit-Fließkommazahl mit Kodierung nach IEEE 754, wobei zu beachten ist, dass die Reihenfolge, in der die Bytes übertragen werden, von der Modbus-Implementierung eines Gerätes abhängig ist. JUMO tecLine-Sensoren (20263x) übertragen die Bytes nicht in der nach IEEE 754 Standard-Kodierung festgelegten Reihenfolge, sondern in der typischen Reihenfolge für Float-Variablen (Byte 3, Byte 4, Byte 1, Byte 2).</p> <p>S = Vorzeichenbit E = Exponent (2er-Komplement) M = 23 Bit normalisierte Mantisse</p> <p style="text-align: center;">IEEE 754 Standard-Kodierung</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="background-color: #ffff00;">Byte 1</td> <td style="background-color: #ffff00;">Byte 2</td> <td style="background-color: #ffff00;">Byte 3</td> <td style="background-color: #ffff00;">Byte 4</td> </tr> <tr> <td style="background-color: #ffff00;">SE7EEEEEE1</td> <td style="background-color: #ffff00;">E0M23MMMMMM16</td> <td style="background-color: #ffff00;">M15MMMMMMM8</td> <td style="background-color: #ffff00;">M7MMMMMMM0</td> </tr> </table> <p style="text-align: center;">typische Modbus-Kodierung von Float-Variablen</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="2" style="background-color: #00b0f0; color: white;">Adresse des 1. Modbus-Registers der Variablen</td> <td colspan="2" style="background-color: #ffff00;">Adresse des 2. Modbus-Registers der Variablen</td> </tr> <tr> <td style="background-color: #00b0f0; color: white;">Byte 3</td> <td style="background-color: #00b0f0; color: white;">Byte 4</td> <td style="background-color: #ffff00;">Byte 1</td> <td style="background-color: #ffff00;">Byte 2</td> </tr> <tr> <td style="background-color: #00b0f0; color: white;">M15MMMMMMM8</td> <td style="background-color: #00b0f0; color: white;">M7MMMMMMM0</td> <td style="background-color: #ffff00;">SE7EEEEEE1</td> <td style="background-color: #ffff00;">E0M23MMMMMM16</td> </tr> </table> <p>Beim Erstellen kundeneigener Applikationen ist die korrekte Byte-Reihenfolge im Ablageformat zu überprüfen. Viele Compiler nutzen folgendes Ablageformat:</p> <p style="text-align: center;">Compiler-Kodierung</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="background-color: #ffff00;">Byte 4</td> <td style="background-color: #ffff00;">Byte 3</td> <td style="background-color: #ffff00;">Byte 2</td> <td style="background-color: #ffff00;">Byte 1</td> </tr> <tr> <td style="background-color: #ffff00;">MMMMMMMM</td> <td style="background-color: #ffff00;">MMMMMMMM</td> <td style="background-color: #ffff00;">EMMMMMMM</td> <td style="background-color: #ffff00;">SEEEEEEE</td> </tr> <tr> <td style="background-color: #ffff00;">Adresse x</td> <td style="background-color: #ffff00;">Adresse x+1</td> <td style="background-color: #ffff00;">Adresse x+2</td> <td style="background-color: #ffff00;">Adresse x+3</td> </tr> </table>	Byte 1	Byte 2	Byte 3	Byte 4	SE7EEEEEE1	E0M23MMMMMM16	M15MMMMMMM8	M7MMMMMMM0	Adresse des 1. Modbus-Registers der Variablen		Adresse des 2. Modbus-Registers der Variablen		Byte 3	Byte 4	Byte 1	Byte 2	M15MMMMMMM8	M7MMMMMMM0	SE7EEEEEE1	E0M23MMMMMM16	Byte 4	Byte 3	Byte 2	Byte 1	MMMMMMMM	MMMMMMMM	EMMMMMMM	SEEEEEEE	Adresse x	Adresse x+1	Adresse x+2	Adresse x+3	read only	03, 04	2
		Byte 1	Byte 2	Byte 3	Byte 4																															
SE7EEEEEE1	E0M23MMMMMM16	M15MMMMMMM8	M7MMMMMMM0																																	
Adresse des 1. Modbus-Registers der Variablen		Adresse des 2. Modbus-Registers der Variablen																																		
Byte 3	Byte 4	Byte 1	Byte 2																																	
M15MMMMMMM8	M7MMMMMMM0	SE7EEEEEE1	E0M23MMMMMM16																																	
Byte 4	Byte 3	Byte 2	Byte 1																																	
MMMMMMMM	MMMMMMMM	EMMMMMMM	SEEEEEEE																																	
Adresse x	Adresse x+1	Adresse x+2	Adresse x+3																																	
read/write	03, 04, 16																																			
int	<p>Wort (16 Bit) als vorzeichenloser ganzzahliger Wert.</p> <p>Das höherwertige Byte (MSB) wird vor dem niederwertigen Byte (LSB) übertragen.</p> <p>Wertebereich: 0 bis 65.535</p>	read only	03, 04	1																																
		read/write	03, 04, 16																																	
unsigned longint	<p>Doppelwort (32 Bit) als vorzeichenloser ganzzahliger Wert.</p> <p>MSB wird vor LSB übertragen.</p> <p>Wertebereich: 0 bis 4.294.967.295</p>	read only	03, 04	2																																
		read/write	03, 04, 16																																	
char [...]	<p>Zeichen/Byte (8 Bit) als vorzeichenloser ganzzahliger Wert.</p> <p>2 Zeichen sind in 1 Wort enthalten.</p> <p>Das 1. Zeichen wird im MSB übertragen, das 2. Zeichen im LSB.</p> <p>Wertebereich: 0 bis 255</p>	read only	03, 04	1																																

2 Modbus-Protokollbeschreibung

2.6 Beispiele für die Übertragung von Daten

Zum Auslesen von Integer-, Float- und Text-Werten wird die Funktion 0x03 (Einlesen von n Worten) verwendet.

Datenanfrage

Slave-Adresse	Funktion 0x03	Adresse erstes Wort	Wortanzahl	Checksumme CRC
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

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

Antwort

Slave-Adresse	Funktion 0x03	Anzahl gelesener Bytes	Wertwert(e)	Checksumme CRC
1 Byte	1 Byte	1 Byte	n Bytes	2 Bytes

2.6.1 Integer-Werte (16 Bit)

Beispiel

In diesem Beispiel soll der Stand der Firmware eines JUMO tecLine CI2 (202630) an Adresse 0x0309 (⇒ „Gerätedaten“, Seite 18) ausgelesen werden. Der Wert in diesem Beispiel soll 1410 (1.410), also Hexadezimal 582 sein.

Datenanfrage:

01	03	03 09	00 01	54 4C
Slave	Funktion	Adresse 1. Wort	Wortanzahl	CRC

Antwort (Wert im Modbus int-Format):

01	03	02	05 82	3B 75
Slave	Funktion	Bytes gelesen	Integer-Wert	CRC

2.6.2 Long Integer-Werte (32 Bit)

Beispiel

In diesem Beispiel soll der Zeitstempel der neuesten im Kalibrierspeicher befindlichen Kalibrierung (Kalibrierung 0) eines JUMO tecLine CI2 (202630) an Adresse 0x0214 (⇒ „Prozessdaten Parameter“, Seite 19) ausgelesen werden. Der Wert in diesem Beispiel soll dem Datum 08.03.2019, 13:10 Uhr entsprechen, also dezimal 1903081310 und somit hexadezimal 716EB75E sein.

Datenanfrage:

01	03	02 14	00 02	85 B7
Slave	Funktion	Adresse 1. Wort	Wortanzahl	CRC

Antwort (Wert im Modbus int-Format):

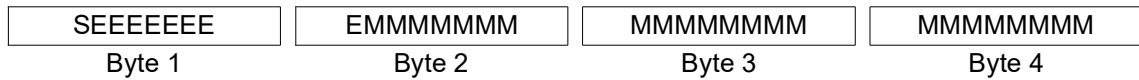
01	03	04	71 6E B7 5E	76 DA
Slave	Funktion	Bytes gelesen	Long Integer- Wert	CRC

2 Modbus-Protokollbeschreibung

2.6.3 Float-Werte

JUMO tecLine-Sensoren (20263x) arbeiten bei Float-Werten mit einer vom IEEE-754-Standard-Format (32 Bit) **abweichenden** Reihenfolge.

Single-Float-Format (32 Bit) nach Standard IEEE 754

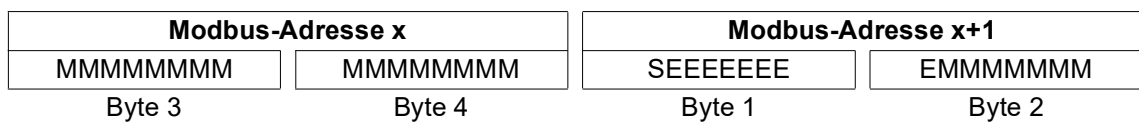


S - Vorzeichen-Bit

E - Exponent (2er-Komplement)

M - 23 Bit normalisierte Mantisse

Modbus-Float-Format bei JUMO tecLine-Sensoren (20263x)



Beispiel

In diesem Beispiel soll der Konzentrationswert des Desinfektionsmittels an Adresse 0x0000 des Gerätes ausgelesen werden. Der Wert soll hier 0,168 ppm (0x08313E2C im umgekehrten IEEE-754-Format) sein.

Datenanfrage:

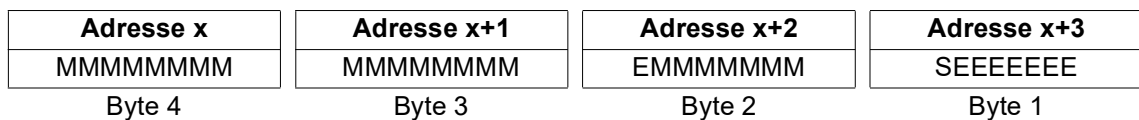
01	03	00 00	00 02	C4 0B
Slave	Funktion	Adresse 1. Wort	Wortanzahl	CRC

Antwort (Werte im Modbus-Float-Format):

01	03	04	08 31 3E 2C	B8 21
Slave	Funktion	Bytes gelesen	Float-Wert	CRC

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

Float-Wert



HINWEIS!

Die Reihenfolge der Bytes hängt davon ab, wie Float-Werte in der betreffenden Anwendung gespeichert werden. Eventuell müssen die Bytes im Schnittstellenprogramm entsprechend vertauscht werden.

2 Modbus-Protokollbeschreibung

2.7 Checksumme (CRC16)

Berechnungsschema

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

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



HINWEIS!

Das Low-Byte der Checksumme wird zuerst übertragen!

Beispiel: Die CRC16-Checksumme DB 25 wird in der Reihenfolge 25 DB übertragen und dargestellt.

Beispiel

Einheit der Ausgabe des Wertes für Konzentration an Adresse 0x0200 abfragen:

Anweisung: Lese ein Wort von Adresse 0x0200

01	03	02 00	00 01	85 B2
Slave	Funktion	Adresse	Ein Wort lesen	CRC

Antwort (CRC16 = 0x45F8)

01	03	02	00 03	F8 45
Slave	Funktion	Anzahl Bytes	Wort 1	CRC

Wort 1 = 3 steht für die Einheit „ppm“.

2 Modbus-Protokollbeschreibung

2.8 Fehlermeldungen

2.8.1 Modbus-Fehlercodes

Voraussetzungen für die Modbus-Kommunikation

Folgende Bedingungen müssen erfüllt sein, damit ein Slave Anfragen empfangen, bearbeiten und beantworten kann:

- Baudrate und Datenformat von Master und Slave müssen übereinstimmen.
- In der Anfrage muss die korrekte Slave-Adresse verwendet werden.
- Slave-Geräte antworten nur bei erfolgreichem Prüfsummen-Check der Anfrage durch den Slave. Anderenfalls wird die Anfrage vom Slave verworfen.
- Die Anweisung des Masters muss vollständig und konform zum Modbus-Protokoll sein.
- Die Anzahl der zu lesenden Worte muss größer 0 sein.

Fehlercodes

Wurde die Datenanfrage des Masters vom Slave ohne Übertragungsfehler empfangen, konnte aber nicht bearbeitet werden, antwortet der Slave mit einem Fehlercode. Folgende Fehlercodes können auftreten:

- 01 = ungültige Funktion; Die Funktionscodes, die vom JUMO tecLine (20263x) unterstützt werden, sind im Kapitel 2.4 „Funktionscodes“, Seite 7 aufgeführt.
- 02 = ungültige Adresse oder eine zu große Anzahl von Worten bzw. Bits soll gelesen oder geschrieben werden
- 03 = Das Format der Daten kann nicht gelesen werden.
- 255 = Es liegt ein Kommunikationsproblem vor.

Modbus-Fehlerantworten sind daran erkennbar, dass das MSB des Funktionscodes auf 1 gesetzt wurde.

Antwort im Fehlerfall

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

Der Funktionscode wird mit 0x80 verODERT. Dadurch wird das höchstwertige Bit (msb) auf 1 gesetzt.

Beispiel

Datenanfrage:

01	06	23 45	00 01	52 5B
Slave	Wort schreiben	Wortadresse	Wort-Wert	CRC

Antwort (mit Fehlercode 2):

01	86	02	C3 A1
Slave	Funktion OR	Fehler	CRC

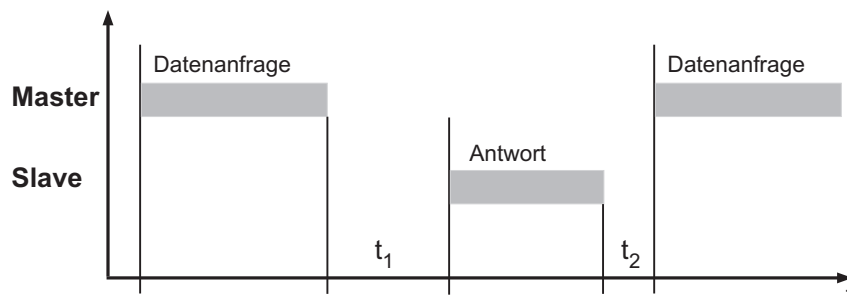
Antwort mit Fehlercode 02, weil die Adresse 0x2345 nicht vorhanden ist.

3 Modbus über serielle Schnittstelle

3.1 Modbus-Slave-Betrieb über serielle Schnittstelle RS 485

Zeitlicher Ablauf der Kommunikation

Ein Abfragezyklus im Bus läuft nach folgendem Zeitschema ab:



t_1	Wartezeit 1, bestehend aus 3,5 Zeichenübertragungszeiten plus der Zeit, die der Slave (Sensor) für die Verarbeitung der Datenabfrage benötigt
t_2	Wartezeit 2, die der Master einhalten muss, bevor er eine neue Datenanfrage startet: 3,5 Zeichenübertragungszeiten



HINWEIS!

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



HINWEIS!

Das Ende-Kennzeichen nach einer Datenanfrage oder Datenantwort ist 1,5 Zeichen lang. Die Dauer dieser 1,5 Zeichen ist abhängig von der Baudrate.

Zeichenübertragungszeit

Anfang und Ende eines Datenblocks sind durch Übertragungspausen gekennzeichnet. Die Zeichenübertragungszeit (Zeit für die Übertragung eines Zeichens) ist abhängig von der Baudrate.

Bei einem Datenformat von 8 Datenbits, keinem Paritätsbit, einem Start- und einem Stopbit ergibt sich:

$$\text{Zeichenübertragungszeit [ms]} = 1000 \times 10 \text{ Bit} \div \text{Baudrate}$$

Bei allen anderen, für den Sensor einstellbaren Datenformaten ergibt sich:

$$\text{Zeichenübertragungszeit [ms]} = 1000 \times 11 \text{ Bit} \div \text{Baudrate}$$

Rechenbeispiel für 9600 Baud:

Baudrate[Baud]	Datenformat[Bit]	Zeichenübertragungszeit[ms]
9600	11	1,146
	10	1,042

Wartezeit

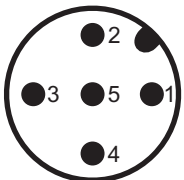
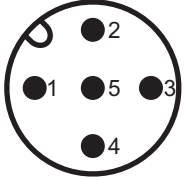
Die Wartezeit beträgt, wie bereits weiter oben in „Zeitlicher Ablauf der Kommunikation“ beschrieben, grundsätzlich 3,5 Zeichenübertragungszeiten. Sie beginnt jeweils nach dem Kennzeichen für Datenanfrage- oder Antwort-Ende.

Rechenbeispiel für 9600 Baud:

Baudrate[Baud]	Datenformat[Bit]	Wartezeit[ms]
9600	11	4,011
	10	3,647

4 Schnittstellen

4.1 Schnittstellenbelegung der JUMO tecLine Sensoren (20263x)

5-poliger M12-Steckanschluss, A-kodiert		
Pin	Potenzial	Symbol
1	nicht angeschlossen	<p>Stecker</p>  <p>Buchse</p> 
2	+24 V	
3	GND	
4	RS485 B (RxD/TxD-)	
5	RS485 A (RxD/TxD+)	
Der Anschluss an der seriellen Schnittstellen eines Mastergerätes mit Schraub- oder Federzugklemmen kann über ein JUMO M12-digiLine-Master-Anschlusskabel erfolgen.		

5 Schnittstellen konfigurieren

Hinweise



VORSICHT!

Durch fehlerhafte Installation oder falsche Einstellungen von Betriebsmitteln können unerwartete Betriebszustände einer Anlage auftreten.

Dies kann Prozesse in ihrer ordnungsgemäßen Funktion beeinträchtigen oder zu Schäden führen.

- ▶ Daher immer vom Gerät unabhängige Sicherheitseinrichtungen vorsehen und die Einstellungen nur von Fachpersonal durchführen lassen.



HINWEIS!

Änderungen der in diesem Kapitel beschriebenen Konfigurationseinstellungen von JUMO tecLine-Sensoren (20263x) können am PC mit der JUMO DSM-Software vorgenommen werden. Die Handhabung der JUMO DSM-Software ist in der Betriebsanleitung der JUMO DSM-Software ausführlich beschrieben.

5.1 Einstellungen für die serielle Schnittstelle

Damit alle Teilnehmergeräte in einem Bus miteinander kommunizieren können, müssen ihre Schnittstellen-Einstellungen übereinstimmen. Die folgende Tabelle stellt die Einstellmöglichkeiten für die JUMO tecLine-Sensoren (20263x) dar.

Konfigurationspunkt	Auswahl/Einstellungen	Beschreibung
Baudrate	2400 4800 9600 19200 38400 57600 115200	Übertragungsgeschwindigkeit (Symbolrate) der seriellen Schnittstelle
Datenformat	8 - 2 - none 8 - 1 - even 8 - 1 - odd 8 - 1 - none	Format des Datenwortes Nutzbit - Stoppbit - Parität
Geräteadresse	1 bis 247	eindeutige Kennung eines Busteilnehmers 0 = Broadcast-Adresse ^a 1 bis 247 = Unicast-Adressen ^b

^a Im Modbus-Standard ist die Geräteadressierung festgelegt. Die Broadcast-Adresse darf nicht als Slave-Adresse verwendet werden. Sie ist für Rundsendungsnachrichten vorgesehen.

^b Unicast-Adressen sind für die Verwendung als Slave-Adressen vorgesehen. Sie dienen der eindeutigen Kennung der Slave-Geräte, damit diese vom Master explizit angesprochen werden können.

6 Modbus-Adresstabellen



HINWEIS!

Die in den nachfolgenden Tabellen angegebenen Adressen können, je nach Zugriff, gelesen oder gelesen und beschrieben werden. Für diese Zugriffe ist keine Programmierung des Sensors erforderlich.

Modbus Funktionen

Adresse	Funktion	Parameter
0x03	Read Holding Registers (16 Bit)	Verwendung für alle Parameter
0x04	Read Input Register (16 Bit)	
0x06	Write Single Register	
0x10	Write Multiple Registers	

Gerätedaten

Adresse	Zugriff	Datentyp	Parameter	Beispiel
0x0308	R/O	int	Hardware	1130 (1.130)
0x0309	R/O	int	Firmware	1503 (1.503)
0x030A	R/O	float	Nennsteilheit	7.5
0x030C	R/O	char [20]	F-Nr.	0104714601019120001
0x0317	R/O	char [10]	Teile-Nr.	00705172

Gerätedaten Modbus

Adresse	Zugriff	Datentyp	Parameter	Wertebereich		Default	
0x0400	R/W	int	Slave Adresse	1 bis 247		20	202630 (CI2)
						30	202631 (TC)
						50	202634 (O3)
						80	202634 (CIO2)
						60	202636 (H2O2)
						70	202636 (PAA)
						90	202637 (Br)
						100	202681 (CI2 OM)
0x0401	R/W	int	Baudrate	0	2400	4	
				1	4800		
				2	9600		
				3	19200		
				4	38400		
				5	57600		
				6	115200		
0x0402	R/W	int	Parity/Stopbit	0	none/2	3	
				1	even/1		
				2	odd/1		
				3	none/1		

6 Modbus-Adresstabellen

Adresse	Zugriff	Datentyp	Parameter	Wertebereich	Default	
0x0200	R/O	int	Einheit	0	%	
				1	‰	
				2	g/l	
				3	ppm	
				4	mg/l	
				5	ppb	
0x0201	R/O	int	Nachkommastellen	0	0000	
				1	000,0	
				2	00,00	
				3	0,000	
0x0206	R/W	float	X_Null			
0x0208	R/W	float	X_Span			
0x020A	R/W	unsigned longint	DatumUhrzeit	yymmddhhmm		
Historie						
0x0210	R/O	float	X_Null (0)			
0x0212	R/O	float	X_Span (0)			
0x0214	R/O	unsigned longint	DatumUhrzeit (0)	yymmddhhmm		
0x0216	R/O	float	X_Null (1)			
0x0218	R/O	float	X_Span (1)			
0x021A	R/O	unsigned longint	DatumUhrzeit (1)	yymmddhhmm		
0x021C	R/O	float	X_Null (2)			
0x021E	R/O	float	X_Span (2)			
0x0220	R/O	unsigned longint	DatumUhrzeit (2)	yymmddhhmm		
0x0212	R/O	float	X_Null (3)			
0x0214	R/O	float	X_Span (3)			
0x0226	R/O	unsigned longint	DatumUhrzeit (3)	yymmddhhmm		
0x0228	R/O	float	X_Null (4)			
0x022A	R/O	float	X_Span (4)			
0x022C	R/O	unsigned longint	DatumUhrzeit (4)	yymmddhhmm		
0x022E	R/O	float	Messbereich		20	

Prozessdaten Messwerte

Adresse	Zugriff	Datentyp	Parameter
0x0000	R/O	float	Konzentration (in gewählter Einheit)
0x0002	R/O	float	Zellenstrom (in nA bei 25 °C)
0x0004	R/O	float	Temperatur (in °C)

7 Anhang

7.1 Ergänzende Informationen

Definitionen	<ul style="list-style-type: none">• X_Null (nA) Strom, wenn kein Desinfektionsmittel im zu messenden Medium vorhanden ist. Die Startadresse für diesen Wert ist 0x0206.• X_Span (nA/Einheit) Stromwert der aktuellen Kalibrierung im Verhältnis zur Konzentration des Desinfektionsmittels bei der aktuellen Kalibrierung. Die Startadresse für diesen Wert ist 0x0208. Die Einheit erhält man durch Auslesen der Adresse 0x0200.• Konzentration Die Software berechnet die Konzentration wie folgt: $\frac{\text{aktueller Strom} - X_Null}{X_Span}$
Datum/Zeitstempel	<ul style="list-style-type: none">• Datentyp: unsigned long• größte darstellbare Zahl: $2^{32} = 42 94 96 72 96$, entsprechend (theoretisch) Year: (20)42, Month: 94, Day: 96, Hour: 72, Minute: 96• Letztmögliches Datum: 31.12.2042, 23:59 Uhr (42 12 31 23 59)

7.2 Funktionsweise der Historie

Wie in den Modbus-Adresstabellen des Sensors (⇒Seite 18) ersichtlich, werden die **aktuellen Kalibrierdaten** an den Adressen 0x0206 bis 0x020A abgelegt.

Das Speichersystem der Historie funktioniert nach folgenden Regeln:

- Eine durchgeführte Kalibrierung wird mit dem Schreiben von „Datum/Uhrzeit“ (Datumstempel) aktiv.
- Hat die aktuelle Kalibrierung einen größeren (neueren) Datumstempel als die letzte Kalibrierung, wird diese an den Adressen für die aktuelle Kalibrierung (0x0206 bis 0x020A) abgelegt.
 - Die Daten der **aktuellen Kalibrierung** werden ebenso an den Adressen 0x0210 bis 0x0214 (X_Null (0), X_Span (0) und Datum/Uhrzeit (0)), **also auf Platz 0 der Historie**, abgelegt.
 - Die Daten der **alten Kalibrierung** werden **vom bisherigen Platz 0 der Historie auf Platz 1** verschoben, auch **alle weiteren Daten** werden jeweils **um einen Platz** weitergeschoben.
 - Die bisher **ältesten** Daten (Platz 4) können nicht weitergeschoben werden, sie werden **gelöscht**.
- Hat die aktuelle Kalibrierung einen Datumstempel, der **kleiner oder gleich dem der letzten Kalibrierung** ist, werden die Daten der letzten Kalibrierung (an den Adressen 0x0206 bis 0x020A) und die Daten auf Platz 0 der Historie (an den Adressen 0x0210 bis 0x0214) **überschrieben**.
Die Daten der Plätze 1 bis 4 in der Historie bleiben in diesem Fall unberührt!



HINWEIS!

Da der Datumstempel die Minute als kleinste Einheit hat, sollte die Zeit zwischen 2 Kalibrierungen mindestens 1 Minute betragen.

7.3 Kalibrierung der Sensorsteilheit

7.3.1 Überprüfung der Sensorsteilheit an der Messstelle



HINWEIS!

Beachten Sie die im Kapitel „Kalibrieren“ der Betriebsanleitung Ihres Sensors enthaltenen Informationen zur Kalibrierung.

1. Stellen Sie sicher, dass an der Messstelle Desinfektionsmittel vorhanden ist und dessen Konzentration stabil ist.
2. Warten Sie, bis das Signal am Sensor stabil ist.



HINWEIS!

Berücksichtigen Sie die im Typenblatt für Ihren Sensor angegebene Einlaufzeit.

3. Lesen sie den Zellenstrom (in nA) an Adresse 0x0002 aus. Notieren Sie den Wert.
4. Entnehmen Sie eine Probe des Messwassers an einer dem Einbauort des Sensors nahegelegenen Stelle.
5. Messen Sie die Konzentration des Desinfektionsmittels so schnell wie möglich mit Hilfe eines zuverlässigen und präzisen Verfahrens, je nach Sensortyp z. B. mit einem Photometer oder durch Titration.



HINWEIS!

Verwenden Sie **kein** Colorimeter für die Messung der Konzentration des Desinfektionsmittels! Ein Colorimeter ist lediglich ein Indikator. Für eine Kalibrierung reicht dessen Genauigkeit keinesfalls aus!

6. Berechnen Sie die Steilheit des Sensors durch Division des zuvor notierten Zellenstromes durch die gemessene Konzentration des Desinfektionsmittels.



HINWEIS!

Falls ein von Null abweichender Strom für den Sensornullpunkt ermittelt wurde, muss dieser vor der Division vom notierten Zellenstrom subtrahiert werden.

7.3.2 Speichern der Kalibrierdaten im Sensor

1. Schreiben Sie den Wert „0“ an die Adresse 0x0206. Im Falle einer Nullpunktkalibrierung schreiben Sie den in Kapitel 7.4 „Kalibrierung des Sensor-Nullpunktes“, Seite 23 ermittelten Stromwert bei desinfektionsmittelfreiem Prozesswasser in die Adresse 0x0206.
2. Schreiben Sie den im Kapitel 7.3.1 „Überprüfung der Sensorsteilheit an der Messstelle“, Seite 21 berechneten Wert für die Steilheit an die Adresse 0x0208.
3. Schreiben Sie den Wert für „Datum/Uhrzeit“ an die Adresse 0x020A.

Die Struktur für diese Variable ist „YYMMDDHHMM“. Ein Beispiel für „Datum/Uhrzeit“ ist 1904101430 (siehe auch Kapitel 7.1 „Ergänzende Informationen“, Seite 20).

Mit dem Schreiben von „Datum/Uhrzeit“ ist die Kalibrierung aktiv.

7 Anhang

7.3.3 Überprüfung der Kalibrierung

Software Überprüfung

Überprüfen Sie die Werte für Nullpunkt und Steilheit durch Auslesen der Werte in den Adressen 0x0210 (X_Null (0)) und 0x0212 (X_Span (0)).

An Adresse 0x0210 sollte der Wert „0“ stehen (oder der Wert aus der Nullpunktkalibrierung aus Kapitel 7.4 „Kalibrierung des Sensor-Nullpunktes“, Seite 23, falls diese durchgeführt wurde).

An Adresse 0x0212 sollte der Wert aus der Berechnung der Steilheit aus Kapitel 7.3.1 „Überprüfung der Sensorsteilheit an der Messstelle“, Seite 21 stehen.

Chemische Überprüfung

Die vom Sensor gemessene Konzentration und die analytisch ermittelte Konzentration sollten weitgehend identisch sein.

7.3.4 Regelmäßige Überprüfung

Überprüfen Sie die Kalibrierung regelmäßig, am besten wöchentlich, durch Ausführen der Schritte ab Kapitel 7.3.1 „Überprüfung der Sensorsteilheit an der Messstelle“, Seite 21.

Wenn Sie feststellen, dass die aktuell ermittelte Steilheit geringer als 30 % der Nennsteilheit (Wert an Adresse 0x030A) ist, befolgen Sie die Anweisungen zur Wartung in der Betriebsanleitung Ihres Sensors.



HINWEIS!

Nutzen Sie das Kalibrierlogbuch (Historie) um die Werte für die letzten 5 Kalibrierungen einzusehen. Beachten Sie, dass die 6. Kalibrierung die bisher älteste Kalibrierung im Kalibrierlogbuch überschreibt.

7.4 Kalibrierung des Sensor-Nullpunktes



HINWEIS!

Grundsätzlich ist eine Kalibrierung des Sensor-Nullpunktes nicht erforderlich und wird von uns nicht empfohlen!

Nach einer langen Einsatzzeit ist es dennoch unter Umständen erforderlich, den Nullpunkt des Sensors zu kalibrieren.

7.4.1 Messung des Zellenstromes bei Abwesenheit von Desinfektionsmittel

Messung des Zellenstromes mit Sensor im Messkreis

Wenn der Messkreis ohne Desinfektionsmittel betrieben werden kann, ist es nicht notwendig, den Sensor aus dem Messkreis auszubauen.



HINWEIS!

Denken Sie daran, dass sich Bakterien und Keime bei Abwesenheit von Desinfektionsmittel vermehren! Halten Sie die Betriebszeit ohne Desinfektionsmittel so kurz wie möglich.

1. Stellen Sie sicher, dass sich der Sensor in einem komplett desinfektionsmittelfreien Medium befindet.
2. Warten Sie, bis das Signal des Sensors stabil ist.
3. Lesen Sie den Zellenstrom (in nA) an Adresse 0x0002 und notieren Sie diesen.
4. Führen Sie eine Steilheitskalibrierung durch wie ab Kapitel 7.3.1 „Überprüfung der Sensorsteilheit an der Messstelle“, Seite 21 beschrieben.

Messung des Zellenstromes mit ausgebautem Sensor

Wenn der Messkreis **nicht** ohne Desinfektionsmittel betrieben werden kann, bauen Sie den Sensor aus und halten ihn in ein mit Wasser gefülltes Becherglas.



HINWEIS!

Tauchen Sie den Sensor nicht komplett unter Wasser!
Stellen Sie den Sensor nicht mit der Membran auf den Boden des Becherglases!
Bewegen Sie den Sensor während der Messung im Becherglas oder verwenden Sie einen Magnetrührer.

1. Stellen Sie sicher, dass sich der Sensor in komplett desinfektionsmittelfreiem Wasser befindet.
2. Stellen Sie sicher, dass das Wasser im Becherglas die gleiche Temperatur wie das im Messkreis enthaltene Wasser hat.
3. Warten Sie, bis das Signal des Sensors stabil ist.
4. Lesen Sie den Zellenstrom (in nA) an Adresse 0x0002 und notieren Sie diesen.
5. Führen Sie eine Steilheitskalibrierung durch wie ab Kapitel 7.3.1 „Überprüfung der Sensorsteilheit an der Messstelle“, Seite 21 beschrieben.

7 Anhang

7.5 Wiederherstellung der Nennsteilheit

7.5.1 Einstellen der Standardwerte für Nennsteilheit und Nullpunkt



HINWEIS!

Der Sensor wird ab Werk unkalibriert ausgeliefert. Wenn der Sensor neu ist und noch nicht kalibriert wurde, ist die nachfolgend beschriebene Einstellung der Nennsteilheit nicht erforderlich.

1. Lesen Sie die Nennsteilheit an der Adresse 0x030A aus.
Beispiele für Nennsteilheiten sind z. B. 22 nA/ppm beim Sensor für freies Chlor mit dem Messbereich 0 bis 200 ppm (202630/53-45) oder 7,5 nA/ppm beim Sensor für Gesamtchlor mit dem Messbereich 0 bis 20 ppm (202631/52-37).
2. Schreiben Sie die Nennsteilheit an die Adresse 0x0208.
Mit diesem Schritt wird die Ersetzung der zuvor durch Kalibrierung ermittelte Steilheit durch die Nennsteilheit und das Ablegen der letzten Kalibrierung in der Historie vorbereitet.
3. Schreiben Sie den Wert „0“ an die Adresse 0x0206. Dies ist der Standardwert für den Nullpunkt.
4. Schreiben Sie den Datumsstempel an die Adresse 0x020A.
Die Struktur des Datumsstempels ist in Kapitel 7.1 „Ergänzende Informationen“, Seite 20 erklärt.
Mit dem Schreiben des Datumsstempels wird die Einstellung der Nennsteilheit und das Ablegen der letzten Kalibrierung in der Historie abgeschlossen.

7.5.2 Überprüfung des übertragenen Wertes für die Sensorsteilheit



HINWEIS!

Wenn der Sensor neu ist und noch nicht kalibriert wurde, ist die nachfolgend beschriebene Überprüfung der Nennsteilheit nicht erforderlich.

1. Überprüfen Sie die Werte an den Adressen 0x0210 (X_Null (0)) und 0x0212 (X_Span (0)). Der Wert an Adresse 0x0210 sollte „0“ sein während an Adresse 0x0212 der Wert für die Nennsteilheit stehen sollte. In diesem Beispiel der Wert „22“ für den Sensor 202630/53-45.



JUMO GmbH & Co. KG

Moritz-Juchheim-Straße 1
36039 Fulda, Germany

Telefon: +49 661 6003-714
Telefax: +49 661 6003-605
E-Mail: mail@jumo.net
Internet: www.jumo.net

Lieferadresse:
Mackenrodtstraße 14
36039 Fulda, Germany

Postadresse:
36035 Fulda, Germany

Technischer Support Deutschland:

Telefon: +49 661 6003-9135
Telefax: +49 661 6003-881899
E-Mail: support@jumo.net

JUMO Mess- und Regelgeräte GmbH

Pfarrgasse 48
1230 Wien, Austria

Telefon: +43 1 610610
Telefax: +43 1 6106140
E-Mail: info.at@jumo.net
Internet: www.jumo.at

Technischer Support Österreich:

Telefon: +43 1 610610
Telefax: +43 1 6106140
E-Mail: info.at@jumo.net

JUMO Mess- und Regeltechnik AG

Laubisrütistrasse 70
8712 Stäfa, Switzerland

Telefon: +41 44 928 24 44
Telefax: +41 44 928 24 48
E-Mail: info@jumo.ch
Internet: www.jumo.ch

Technischer Support Schweiz:

Telefon: +41 44 928 24 44
Telefax: +41 44 928 24 48
E-Mail: info@jumo.ch

