

# JUMO Quantrol LC100/LC200/LC300

## Universal-PID-Reglerserie



**B 702030.2.0**  
Schnittstellenbeschreibung Modbus





---

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Gerätedokumentation	5
1.2	Sicherheitshinweise	5
<b>2</b>	<b>Protokoll</b>	<b>7</b>
2.1	Master-Slave-Prinzip	7
2.2	Übertragungsmodus (RTU)	7
2.3	Geräteadresse	7
2.4	Zeitlicher Ablauf der Kommunikation	8
2.5	Aufbau der Datenblöcke	9
2.6	Funktionscodes	10
2.6.1	Lesen von n Worten	10
2.6.2	Schreiben eines Wortes	11
2.6.3	Schreiben von n Worten	12
2.7	Übertragungsformat	13
2.7.1	Integer-Werte	13
2.7.2	Float-Werte	13
2.8	Checksumme (CRC16)	14
2.9	Fehlerbehandlung	15
<b>3</b>	<b>RS485-Schnittstelle</b>	<b>17</b>
3.1	Anschlussplan	17
3.2	Konfiguration	17
<b>4</b>	<b>Modbus-Adressen</b>	<b>19</b>
4.1	Datentyp und Zugriffsart	19
4.2	Prozessdaten	19
4.3	Sollwerte	21
4.4	Reglerparameter	21
4.5	Konfiguration	21
4.6	Kommandos	22
4.7	RAM-Speicher	22



## 1.1 Gerätedokumentation

### Typenblatt T 702030 (als PDF-Dokument)

Das Typenblatt liefert allgemeine Informationen über das Gerät und bildet die Grundlage für Planung und Kaufentscheidung.

### Kurzanleitung B 702030.7 (gedruckt im Format DIN A6)

Die Kurzanleitung enthält die wichtigsten Informationen über die Montage, den elektrischen Anschluss sowie die Bedienung, Parametrierung und Konfiguration des Geräts. Die Kurzanleitung ist jedem Gerät beigelegt. Für weitergehende Informationen steht die Betriebsanleitung B 702030.0 als PDF-Dokument zur Verfügung.

### Betriebsanleitung B 702030.0 (als PDF-Dokument)

Die Betriebsanleitung enthält alle Informationen über die Montage, den elektrischen Anschluss, sowie die Bedienung, Parametrierung und Konfiguration des Geräts.

### Schnittstellenbeschreibung B 702030.2.0 (als PDF-Dokument)

Sie liefert Informationen über die RS485-Schnittstelle, das Modbus-Protokoll und die Kommunikation mit anderen Geräten.

Alle PDF-Dokumente stehen unter [www.jumo.net](http://www.jumo.net) zum Download bereit.

## 1.2 Sicherheitshinweise

Diese Anleitung enthält Hinweise, die Sie zur eigenen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Diese Hinweise sind durch Zeichen unterstützt und werden in dieser Anleitung wie gezeigt verwendet.

Lesen Sie diese Anleitung, bevor Sie das Gerät in Betrieb nehmen. Bewahren Sie die Anleitung an einem für alle Benutzer jederzeit zugänglichen Platz auf.

Alle erforderlichen Einstellungen sind in der vorliegenden Anleitung beschrieben. Durch Manipulationen, die nicht in der Anleitung beschrieben oder ausdrücklich verboten sind, gefährden Sie Ihren Anspruch auf Gewährleistung.

### 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.

### Hinweisende Zeichen



#### **HINWEIS!**

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



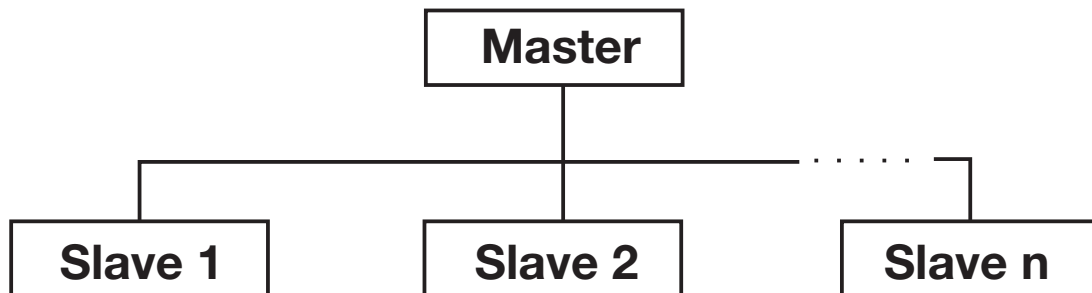
#### **VERWEIS!**

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



## 2.1 Master-Slave-Prinzip

Die Kommunikation zwischen einem Master (z. B. PC) und einem Slave (z. B. Mess- und Regelsystem) mit Modbus findet nach dem Master-Slave-Prinzip in Form von Datenanfrage/Anweisung - Antwort statt.



Der Master steuert den Datenaustausch, die Slaves haben lediglich Antwortfunktion. Sie werden anhand ihrer Geräteadresse identifiziert.

## 2.2 Übertragungsmodus (RTU)

Als Übertragungsmodus wird der RTU-Modus (Remote Terminal Unit) verwendet. Die Übertragung der Daten erfolgt im Binärformat (hexadezimal) mit 8 Bits. Das LSB (least significant bit, engl. das niedrigstwertige Bit) wird zuerst übertragen. Die Betriebsart ASCII-Modus wird nicht unterstützt.

### Datenformat

Mit dem Datenformat wird der Aufbau eines übertragenen Zeichens beschrieben.

Datenwort	Paritätsbit	Stoppbit	Bitanzahl
8 Bits	---	1	9

## 2.3 Geräteadresse

Die Geräteadresse des Slaves ist zwischen 0 und 254 einstellbar. Die Geräteadresse 0 ist reserviert.



### HINWEIS!

Über die RS485-Schnittstelle können maximal 31 Slaves angesprochen werden.

Es gibt zwei Varianten des Datenaustauschs:

### Query

Dies ist eine Datenanfrage/Anweisung des Masters an einen Slave über die entsprechende Geräteadresse.

Der angesprochene Slave antwortet.

## 2 Protokoll

---

### Broadcast

Dies ist eine Anweisung des Masters an alle Slaves über die Geräteadresse 0 (z. B. zur Übertragung eines bestimmten Werts an alle Slaves).

Die angeschlossenen Slaves antworten nicht. Die richtige Übernahme des Werts durch die Slaves sollte in diesem Fall durch anschließendes Auslesen an jedem einzelnen Slave kontrolliert werden.

Eine Datenanfrage mit der Geräteadresse 0 ist nicht sinnvoll.

### 2.4 Zeitlicher Ablauf der Kommunikation

Anfang und Ende eines Datenblocks sind durch Übertragungspausen gekennzeichnet. Zwischen zwei aufeinanderfolgenden Zeichen darf maximal das Dreifache der Zeit zum Übertragen eines Zeichens vergehen.

Die Zeichenübertragungszeit (Zeit für die Übertragung eines Zeichens) ist abhängig von der Baudrate und dem verwendeten Datenformat (Stoppbits und Paritätsbit).

Bei einem Datenformat von 8 Datenbits, keinem Paritätsbit und einem Stoppbit ergibt sich:

**Zeichenübertragungszeit [ms] = 1000 \* 9 bit/Baudrate**

#### Ablauf

<b>Datenanfrage vom Master</b> Übertragungszeit = n Zeichen * 1000 * x bit/Baudrate
Kennzeichen für Datenanfrage-Ende 3 Zeichen * 1000 * x bit/Baudrate
Bearbeitung der Datenanfrage durch den Slave ( $\leq 250\text{ms}$ )
<b>Antwort des Slaves</b> Übertragungszeit = n Zeichen * 1000 * x bit/Baudrate
Kennzeichen für Antwort-Ende 3 Zeichen * 1000 * x bit/Baudrate

#### Beispiel

Kennzeichen für Datenanfrage- oder Antwort-Ende bei Datenformat 10/9 bit

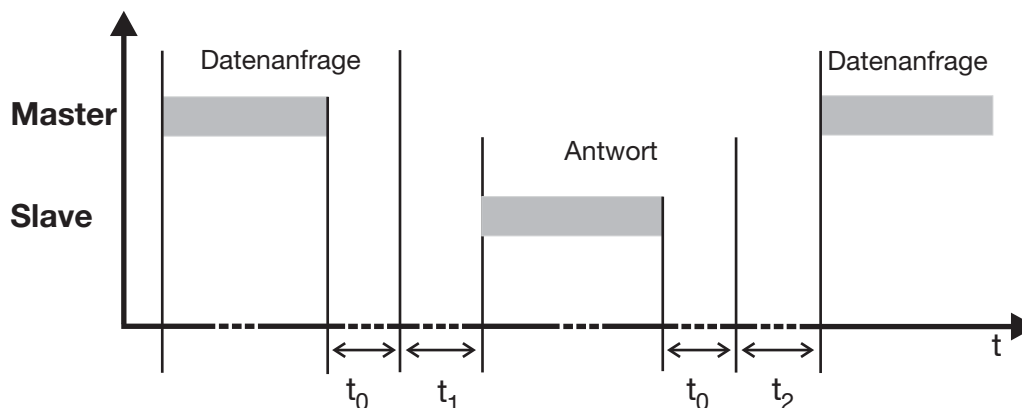
Wartezeit = 3 Zeichen \* 1000 \* 10 bit/Baudrate

Baudrate [Baud]	Datenformat [bit]	Wartezeit [ms] (3 Zeichen)
19200	9	1.41
9600	9	2.82



## Zeitschema

Eine Datenanfrage läuft nach folgendem Zeitschema ab:



$t_0$  Endekennzeichen = 3 Zeichen  
(die Zeit ist von der Baudrate abhängig)

$t_1$  Diese Zeit ist von der internen Bearbeitung abhängig.  
Die maximale Bearbeitungszeit liegt bei 250 ms.  
Die minimal einstellbare Antwortzeit wird bei der RS485-Schnittstelle vom Master benötigt, um die Schnittstellentreiber von Senden auf Empfangen umzustellen.

$t_2$  Diese Zeit braucht der Slave, um von Senden wieder auf Empfangen umzuschalten. Diese Zeit muß der Master einhalten, bevor er eine neue Datenanfrage stellt. Sie muß immer eingehalten werden, auch wenn die neue Datenanfrage an ein anderes Gerät gerichtet ist.  
RS485-Schnittstelle:  $t_2 = 10\text{ms}$

Innerhalb von  $t_1$  und  $t_2$  und während der Antwortzeit des Slaves 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.

## 2.5 Aufbau der Datenblöcke

Alle Datenblöcke haben die gleiche Struktur:

Slave-Adresse	Funktionscode	Datenfeld	Checksumme CRC16
1 Byte	1 Byte	x Byte	2 Bytes

Jeder Datenblock enthält vier Felder:

<b>Slave-Adresse</b>	Geräteadresse eines bestimmten Slaves
<b>Funktionscode</b>	Funktionsauswahl (Lesen, Schreiben von Worten)
<b>Datenfeld</b>	Wortadresse, Wortanzahl, Wortwert(e)
<b>Checksumme</b>	Erkennung von Übertragungsfehlern

# 2 Protokoll

## 2.6 Funktionscodes



**HINWEIS!**

Eine Hexadezimalzahl wird durch ein vorangestelltes „0x“ gekennzeichnet.  
Beispiel: 0x0010 (= 16 dezimal)

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

Funktionsnummer	Funktion	Begrenzung
0x03 oder 0x04	Lesen von n Worten	max. 32 Worte (64 Bytes)
0x06	Schreiben eines Wortes	max. 1 Wort (2 Bytes)
0x10	Schreiben von n Worten	max. 32 Worte (64 Bytes)



**HINWEIS!**

Wenn das Gerät auf diese Funktionen nicht reagiert oder einen Fehlercode ausgibt, siehe Kapitel 2.9 "Fehlerbehandlung", Seite 15.

### 2.6.1 Lesen von n Worten

Mit dieser Funktion werden n ( $n \leq 32$ ) Worte ab einer bestimmten Adresse gelesen.

**Datenanfrage**

Slave-Adresse	Funktion 0x03 oder 0x04	Adresse erstes Wort	Wortanzahl (max. 32)	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

**Antwort**

Slave-Adresse	Funktion 0x03 oder 0x04	Anzahl gelesener Bytes	Wert(e)	Checksumme CRC16
1 Byte	1 Byte	1 Byte	x Byte	2 Bytes

**Beispiel**

Lesen der Sollwerte SP1 und SP2 (jeweils 2 Worte)

Adresse erstes Wort = 0x3100 (Sollwert SP1)

Datenanfrage:

01	03	3100	0004	4AF5
----	----	------	------	------

Antwort (Werte im Modbus-Float-Format):

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

## 2.6.2 Schreiben eines Wortes

Mit dieser Funktion wird ein Wort an einer bestimmten Adresse geschrieben. Die Datenblöcke für Anweisung und Antwort sind identisch.

### Anweisung

Slave-Adresse	Funktion 0x06	Wortadresse	Wortwert	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

### Antwort

Slave-Adresse	Funktion 0x06	Wortadresse	Wortwert	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

### Beispiel

Schreiben des Grenzwerts AL der Grenzwertüberwachung 1 = 275.0  
(Wert = 0x80004389 im Modbus-Float-Format)

Wortadresse = 0x0056

Anweisung: Schreiben des ersten Teils des Werts

01	06	0056	8000	59DA
----	----	------	------	------

Antwort (wie Anweisung):

01	06	0056	8000	59DA
----	----	------	------	------

Anweisung: Schreiben des zweiten Teils des Werts (nächste Wortadresse)

01	06	0057	4389	F88F
----	----	------	------	------

Antwort (wie Anweisung):

01	06	0057	4389	F88F
----	----	------	------	------

## 2 Protokoll

### 2.6.3 Schreiben von n Worten

Mit dieser Funktion werden n ( $n \leq 32$ ) Worte ab einer bestimmten Adresse geschrieben.

#### Anweisung

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortanzahl (max. 32)	Byteanzahl	Wertwert(e)	Check- summe CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	x Byte	2 Bytes

#### Antwort

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortanzahl	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Byte	2 Bytes

#### Beispiel

Schreiben der Sollwerte SP1 und SP2 (jeweils 2 Worte)

Wortadresse = 0x3100 (Sollwert SP1)

Anweisung:

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

Antwort:

01	10	3100	0004	CF36
----	----	------	------	------

## 2.7 Übertragungsformat

### 2.7.1 Integer-Werte

Bei Integer-Werten wird zuerst das High-Byte übertragen, dann das Low-Byte.

#### Beispiel

Abfrage des Integer-Werts von Adresse 0x0021; unter dieser Adresse steht der Wert "4" (Wert 0x0004)

Anfrage: 01 03 0021 0001 (+ 2 Bytes CRC16)

Antwort: 01 03 02 0004 (+ 2 Bytes CRC16)

### 2.7.2 Float-Werte

Float-Werte werden im IEEE-754-Standard-Format (32 Bits) übertragen, allerdings mit dem Unterschied, dass Byte 1 und 2 mit Byte 3 und 4 vertauscht sind.

#### Single-float-Format (32 Bits) nach Standard IEEE 754

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

S = Vorzeichen-Bit

E = Exponent (2er-Komplement)

M = Normalisierte Mantisse (23 Bits)

#### Modbus-float-Format

Modbus-Adresse x		Modbus-Adresse x+1	
MMMMMMMM	MMMMMMMM	SEEEEEEE	EMMMMMMM
Byte 3	Byte 4	Byte 1	Byte 2

#### Beispiel

Abfrage des Float-Werts von Adresse 0x0035; unter dieser Adresse steht der Wert 550.0 (0x44098000 im IEEE-754-Format)

Anfrage: 01 03 0035 0002 (+ 2 Bytes CRC16)

Antwort: 01 03 04 **8000 4409** (+ 2 Bytes CRC16)

Nach der Übertragung vom Gerät müssen die Bytes des Float-Werts entsprechend vertauscht werden.

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

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



#### HINWEIS!

Bitte ermitteln Sie, wie in Ihrer Anwendung Float-Werte gespeichert werden. Ggf. müssen die Bytes nach der Abfrage in Ihrem Schnittstellenprogramm entsprechend getauscht werden.

## 2 Protokoll

### 2.8 Checksumme (CRC16)

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

#### Berechnungsschema

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, dann folgt das High-Byte.

#### Beispiel

Datenanfrage: Lesen von zwei Worten ab Adresse 0x00CE (CRC16 = 0x92A5)

07	03	00	CE	00	02	A5	92
						CRC16	

Antwort: (CRC16 = 0xF5AD)

07	03	04	00	00	41	C8	AD	F5
				Wort 1		Wort 2		CRC16

## 2.9 Fehlerbehandlung

### Fehlercodes

Die folgende Fehlercodes werden verwendet:

1	Ungültige Funktion
2	Ungültige Parameteradresse oder zu große Anzahl von Worten soll gelesen oder geschrieben werden
8	Schreibzugriff auf Parameter verweigert

### Antwort im Fehlerfall

Slave-Adresse	Funktion XX OR 80h	Fehlercode	Checksumme CRC16
1 Byte	1 Byte	1 Byte	2 Bytes

Der Funktionscode wird mit 0x80 verODERT. Dadurch wird das MSB (most significant bit, engl. das höchstwertige Bit) auf 1 gesetzt.

### Beispiel

Datenanfrage:

01	03	40	00	00	04	CRC16
----	----	----	----	----	----	-------

Antwort (mit Fehlercode 2):

01	83	02	CRC16
----	----	----	-------

### Sonderfälle

Wenn der Slave nicht antwortet, können folgende Ursachen vorliegen:

- Baudrate und/oder Datenformat stimmen beim Master und beim Slave nicht überein
- die verwendete Geräteadresse stimmt nicht mit der Slaveadresse überein
- die Checksumme (CRC16) ist nicht korrekt
- die Anweisung des Masters ist unvollständig oder überdefiniert
- die Anzahl der zu lesenden Worte ist Null

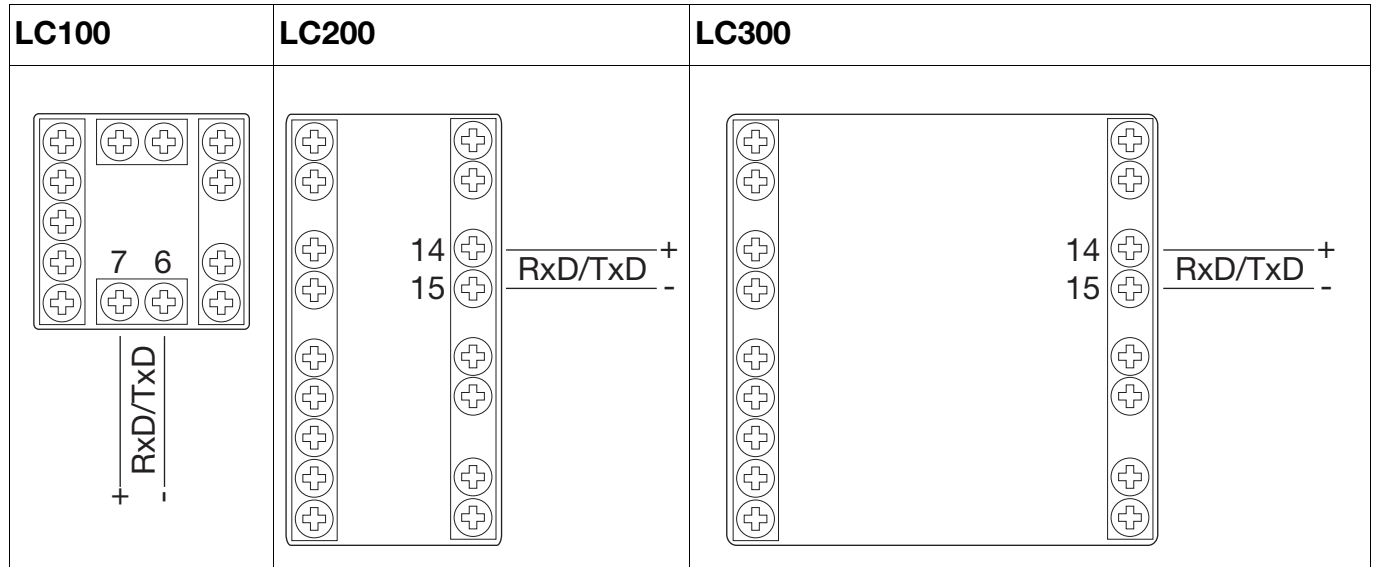
In diesen Fällen sollte die Datenanfrage nach Ablauf der Timeout-Zeit (2s) erneut gesendet werden.





## 3.1 Anschlussplan

Die Geräte dieser Reglerserie können optional mit einer RS485-Schnittstelle bestellt werden. Informationen zur Geräteausführung sind dem Typenblatt T 702030 („Bestellangaben“), der Kurzanleitung B 702030.7 oder der Betriebsanleitung B 702030.0 („Geräteausführung identifizieren“) zu entnehmen.



### VORSICHT!

Elektromagnetische Felder können die Datenübertragung stören. Dadurch kann es zu Übertragungsfehlern kommen.

Zur Vermeidung ist die Schirmung der Schnittstellenleitung einseitig im Schaltschrank zu erden.

## 3.2 Konfiguration

Die folgende Tabelle zeigt die möglichen Einstellungen der Modbus-Schnittstelle, die am Gerät (*Conf* -> *Intf*) bzw. im Setup-Programm vorgenommen werden.

Parameter	Wert	Beschreibung
Baudrate <i>bdr</i>	0	9600 Baud
	1	19200 Baud
Geräteadresse <i>Adr</i>	0 ... 1 ... 254	Adresse im Datenverbund



### HINWEIS!

Bei Kommunikation über die Setup-Schnittstelle (USB) ist die RS485-Schnittstelle inaktiv.



## 4.1 Datentyp und Zugriffsart

In den folgenden Tabellen sind alle Prozess- und Gerätedaten mit ihrer Adresse, dem Datentyp und der Zugriffsart aufgeführt.

Hierbei bedeutet:

R/O	Zugriff nur lesend
W/O	Zugriff nur schreibend
R/W	Zugriff lesend und schreibend
INT	Integer (8 oder 16 Bits)
Bit x	Bit Nr. x (Bit 0 ist das niedrigstwertige Bit)
LONG	Long Integer (4 Byte)
FLOAT	Float-Wert (4 Byte) nach IEEE 754



### VORSICHT!

Schreiboperationen auf R/W-Parameter bewirken ein Abspeichern im EEPROM. Diese Speicherbausteine haben nur eine begrenzte Anzahl von Schreibzyklen (ca. 100.000), weshalb bei häufiger Programmierung diese Funktion abgeschaltet werden kann (bitte Technischen Support kontaktieren). Die Parameterwerte sind dann nur im flüchtigen Speicher (RAM) gespeichert und nach einem Netzausfall verloren.

## 4.2 Prozessdaten

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x0020	INT	R/O	Reglerstatus
	Bit 12		Handbetrieb aktiv (= 0x1000)
	Bit 15		Selbstoptimierung aktiv (= 0x8000)
0x0021	INT	R/O	Binärausgänge 1 ... 5 (Schaltzustände 0 = aus / 1 = ein)
	Bit 0		Ausgang K1: Relais (= 0x0001)
	Bit 1		Ausgang K2: Relais oder Logik (= 0x0002)
	Bit 2		Ausgang K3: Relais oder Logik (= 0x0004)
	Bit 3		Ausgang K4: Relais oder Logik (= 0x0008)
	Bit 4		Ausgang K5: Relais oder Logik (= 0x0010)
0x0023	INT	R/O	Binäreingang (Schaltzustände 0 = offen / 1 = geschlossen)
	Bit 0		Eingang 1 (= 0x0001)

## 4 Modbus-Adressen

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x0024	INT	R/O	Grenzwertüberwachung
	Bit 0		Grenzwertüberwachung 1 (= 0x0001)
	Bit 1		Grenzwertüberwachung 2 (= 0x0002)
0x0025	INT	R/W	Ansteuerung Binärausgänge
	Bit 0 + Bit 15		Ausgang K1 (= 0x8001)
	Bit 1 + Bit 15		Ausgang K2 (= 0x8002)
	Bit 2 + Bit 15		Ausgang K3 (= 0x8004)
	Bit 3 + Bit 15		Ausgang K4 (= 0x8008)
	Bit 4 + Bit 15		Ausgang K5 (= 0x8010)
0x0026	FLOAT	R/O	Analogeingang [V], [mA] oder [Ohm]
0x0028	FLOAT	R/O	Interne Vergleichsstelle [Ohm]
0x002A	FLOAT	R/O	Analogeingang [Anzeigewert]
0x002C	INT	R/O	Abtastzeit
0x002F	FLOAT	R/O	Regler Rampenendwert
0x0031	FLOAT	R/O	Regler Istwert GEFILTERT
0x0033	FLOAT	R/O	Regler Istwert UNGEFILTERT
0x0035	FLOAT	R/W	Regler Sollwert
0x0037	FLOAT	R/O	Regler Stellgradanzeige
0x0039	FLOAT	R/O	Stellgrad Reglerausgang 1 (HEIZEN)
0x003B	FLOAT	R/O	Stellgrad Reglerausgang 2 (KÜHLEN)
0x003D	FLOAT	R/O	Regler Regeldifferenz
0x003F	INT	R/O	Schaltstellung Reglerausgang 1 (HEIZEN)
0x0040	INT	R/O	Schaltstellung Reglerausgang 2 (KÜHLEN)
0x0041	INT	R/O	Handstellgrad
0x0042	LONG	R/O	Timer-Laufzeit
0x0044	LONG	R/O	Timer-Restzeit
0x0046	INT	R/O	Timer-Status
	Bit 1		Timer angehalten (= 0x0002)
	Bit 5		Timer läuft (= 0x0020)
	Bit 6		Timer-Ende (= 0x0040)
	Bit 15		Timer-Signal (= 0x8000)

## 4.3 Sollwerte

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x3100	FLOAT	R/W	Sollwert SP1
0x3102	FLOAT	R/W	Sollwert SP2

## 4.4 Reglerparameter

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x3000	FLOAT	R/W	Regler-Parameter XP1 ( $P_b \dot{t}$ )
0x3002	FLOAT	R/W	Regler-Parameter XP2 ( $P_b \ddot{t}$ )
0x3004	FLOAT	R/W	Regler-Parameter TV ( $d\dot{t}$ )
0x3006	FLOAT	R/W	Regler-Parameter TN ( $r\dot{t}$ )
0x300C	FLOAT	R/W	Regler-Parameter CY1 ( $\dot{Y} \dot{t}$ )
0x300E	FLOAT	R/W	Regler-Parameter CY2 ( $\dot{Y} \ddot{t}$ )
0x3010	FLOAT	R/W	Regler-Parameter XSH ( $d\dot{b}$ )
0x3012	FLOAT	R/W	Regler-Parameter XD1 ( $H\dot{Y} \dot{t}$ )
0x3014	FLOAT	R/W	Regler-Parameter XD2 ( $H\dot{Y} \ddot{t}$ )
0x3017	INT	R/W	Regler-Parameter Y0 ( $\dot{Y} \dot{t}$ )
0x3018	INT	R/W	Regler-Parameter Y1 ( $\dot{Y} \dot{t}$ )
0x3019	INT	R/W	Regler-Parameter Y2 ( $\dot{Y} \ddot{t}$ )

## 4.5 Konfiguration

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x004E	FLOAT	R/W	Rampenfunktion Steigung
0x0050	FLOAT	R/W	Filterzeitkonstante
0x0052	FLOAT	R/W	Offset Analogeingang
0x0054	LONG	R/W	Timer-Wert
0x0056	FLOAT	R/W	Grenzwertüberwachung 1 Grenzwert AL
0x0058	FLOAT	R/W	Grenzwertüberwachung 1 Schaltdifferenz
0x005A	FLOAT	R/W	Grenzwertüberwachung 2 Grenzwert AL
0x005C	FLOAT	R/W	Grenzwertüberwachung 2 Schaltdifferenz

## 4 Modbus-Adressen

### 4.6 Kommandos

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x0047	INT	W/O	Binärfunktionen REGLER
	Bit 0		Selbstoptimierung Start (= 0x0001)
	Bit 1		Selbstoptimierung Abbruch (= 0x0002)
	Bit 2		Handbetrieb (= 0x0004)
	Bit 3		Automatikbetrieb (= 0x0008)
	Bit 4		Regler aus (= 0x0010)
	Bit 5		Verriegelung Handbetrieb (= 0x0020)
	Bit 6		Rampe Halt (= 0x0040)
	Bit 7		Rampe Abbruch (= 0x0080)
	Bit 8		Rampe Neustart (= 0x0100)
	Bit 9		Timer Start (= 0x0200)
	Bit 10		Timer Abbruch (= 0x0400)
	Bit 11		Timer Halt (= 0x0800)
0x0048	INT	W/O	Binärfunktionen BEDIENUNG
	Bit 0		Verriegelung Tastatur (= 0x0001)
	Bit 1		Verriegelung Konfigurations- und Parameterebene (= 0x0002)
	Bit 3		Anzeige AUS (= 0x0008)
0x004A	INT	R/W	Sollwertumschaltung
	Bit 0		Sollwert 1 (= 0x0000)
	Bit 0		Sollwert 2 (= 0x0001)

### 4.7 RAM-Speicher

Über Modbus ist der direkte Zugriff auf den RAM-Speicher des Gerätes möglich, um den Regler-Sollwert (0x3200) und den Regler-Istwert (0x3202) zu schreiben.

Beim Schreiben steht ein Bereich von -1999 bis +9999 zur Verfügung. Statt des Originalwertes wird dann der ins Gerät geschriebene Wert verwendet.

Soll am Gerät wieder der Originalwert verwendet werden, muss der Wert 200001 über Modbus in die jeweilige Speicherstelle geschrieben werden.

Adresse	Datentyp/ Bitnummer	Zugriff	Signalbezeichnung
0x3200	FLOAT	W/O	Regler-Sollwert (beschreibbar)
0x3202	FLOAT	W/O	Regler-Istwert (beschreibbar)





#### **JUMO GmbH & Co. KG**

Moritz-Juchheim-Straße 1  
36039 Fulda, Germany

Telefon: +49 661 6003-727  
Telefax: +49 661 6003-508  
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: service@jumo.net

#### **JUMO Mess- und Regelgeräte Ges.m.b.H**

Pfarrgasse 48  
1232 Wien, Austria

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

Technischer Support Österreich:

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

#### **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