

**Logoline 500**

B 95.3530.2  
Schnittstellenbeschreibung  
12.01/00339580

# Inhalt

---

## 1 Einleitung

1.1	Vorwort .....	3
1.2	Typografische Konventionen .....	4
1.2.1	Warnende Zeichen .....	4
1.2.2	Hinweisende Zeichen .....	4
1.2.3	Darstellungsarten .....	4

## 2 Protokollbeschreibung

2.1	Master-Slave-Prinzip .....	5
2.2	Übertragungsmodus (RTU) .....	5
2.3	Geräteadresse .....	6
2.4	Zeitlicher Ablauf der Kommunikation .....	6
2.4.1	Zeitlicher Ablauf einer Datenanfrage .....	8
2.4.2	Kommunikation während der internen Bearbeitungszeit des Slaves .....	9
2.4.3	Kommunikation während der Antwortzeit des Slaves .....	9
2.5	Aufbau der Datenblöcke .....	9
2.6	Fehlerbehandlung .....	9
2.7	Unterscheidung MOD-Bus/J-Bus .....	10
2.8	Checksumme (CRC16) .....	11
2.9	Schnittstelle .....	12

## 3 Funktionen

3.1	Lesen von n Bits .....	14
3.2	Lesen von n Worten .....	15
3.3	Schreiben eines Bits .....	16
3.4	Schreiben eines Wortes .....	17
3.5	Schreiben von n Bits .....	18
3.6	Schreiben von n Worten .....	19

## 4 Datenfluß

4.1	Azyklische Daten vom Gerät empfangen .....	21
4.2	Azyklische Daten zum Gerät übertragen .....	23
4.3	Übertragungsformat .....	26

## 5 Fehlermeldungen

5.1	Fehlermeldungen der Schnittstelle .....	27
5.2	Fehlermeldungen bei ungültigen Werten .....	27
5.3	System- und Laufzeitfehler .....	28

## 6 Adresstabellen

6.1	Zyklische Daten .....	31
-----	-----------------------	----

---

# Inhalt

---

6.1.1	Gerätedaten .....	31
6.1.2	Prozeßdaten .....	31
6.2	Azyklische Daten .....	33
6.2.1	Text zum Ausdrucken .....	33
6.2.2	Text zum Anzeigen im Display .....	34
6.2.3	Flags für die Anzeigesteuerung des Displays .....	34
6.2.4	float-Werte für das Mathematikmodul .....	35
6.2.5	Reset-Erkennung .....	36

---

# 1 Einleitung

---

## 1.1 Vorwort

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

Bitte unterstützen Sie uns, diese Betriebsanleitung zu verbessern.

Für Ihre Anregungen sind wir dankbar.

Telefon (06 61) 60 03-7 25

Telefax (06 61) 60 03-6 81



Alle erforderlichen Informationen zum Betrieb der Schnittstelle sind in der vorliegenden Betriebsanleitung beschrieben. Sollten bei der Inbetriebnahme trotzdem Schwierigkeiten auftreten, bitten wir Sie, keine unzulässigen Manipulationen vorzunehmen. Sie können Ihren Garantieanspruch gefährden!

Bitte Setzen Sie sich mit der nächsten Niederlassung oder mit dem Stammhaus in Verbindung.



Bei Rücksendungen von Geräteeinschüben, Baugruppen oder Bauelementen sind die Regelungen nach DIN EN 100 015 „Schutz von elektrostatisch gefährdeten Bauelementen“ einzuhalten. Verwenden Sie nur dafür vorgesehene ESD-Verpackungen für den Transport.

Bitte beachten Sie, daß für Schäden, die durch ESD verursacht werden, keine Haftung übernommen werden kann.

ESD= Elektrostatische Entladungen

# 1 Einleitung

---

## 1.2 Typografische Konventionen

### 1.2.1 Warnende Zeichen

Die Zeichen für Vorsicht und Achtung werden in dieser Betriebsanleitung unter folgenden Bedingungen verwendet:



**Vorsicht**

Dieses Zeichen wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu Personenschäden kommen kann!



**Achtung**

Diese Zeichen wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu Beschädigungen von Geräten oder Daten kommen kann!



**Achtung**

Diese Zeichen wird benutzt, wenn Vorsichtsmaßnahmen bei der Handhabung elektrostatisch entladungsgefährdeter Bauelemente zu beachten sind.

### 1.2.2 Hinweisende Zeichen



**Hinweis**

Dieses Zeichen wird benutzt, wenn Sie auf etwas Besonderes aufmerksam gemacht werden sollen.



**Verweis**

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

abc<sup>1</sup>

**Fußnote**

Fußnoten sind Anmerkungen, die auf bestimmte Textstellen Bezug nehmen. Fußnoten bestehen aus zwei Teilen:

Kennzeichnung im Text und Fußnotentext.

Die Kennzeichnung im Text geschieht durch hochstehende fortlaufende Zahlen.

Der Fußnotentext (2 Schriftgrade kleiner als die Grundschrift) steht am unteren Seitenende und beginnt mit einer Zahl und einem Punkt.

### 1.2.3 Darstellungsarten

0x0010

**Hexadezimalzahl**

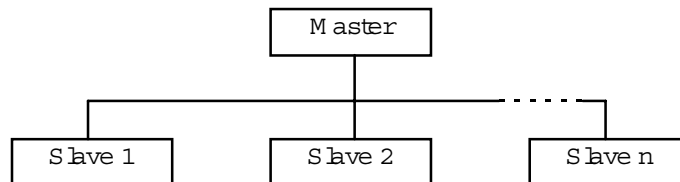
Eine Hexadezimalzahl wird durch ein vorgestelltes „0x“ gekennzeichnet (hier: 16 dezimal).

## 2 Protokollbeschreibung

---

### 2.1 Master-Slave-Prinzip

Die Kommunikation zwischen einem PC (Master) und einem Gerät (Slave) mit MOD-/J-Bus 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. Es können maximal 255 Slaves angesprochen werden.

### 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, 16 Bits bei Integerwerten und 32 Bits bei Floatwerten. Das LSB (least significant bit, engl. das niederwertigste Bit) wird zuerst übertragen. Die Betriebsart ASCII-Modus wird nicht unterstützt.

#### Datenformat

Mit dem Datenformat wird der Aufbau eines übertragenen Bytes beschrieben. Es sind folgende Möglichkeiten des Datenformats gegeben:

Datenwort	Paritätsbit	Stoppbit 1/2 Bit	Bitanzahl
8 Bit	—	1	9
8 Bit	—	2	10
8 Bit	gerade (even)	1	10
8 Bit	ungerade (odd)	1	10
8 Bit	immer 0	1	10

## 2 Protokollbeschreibung

---

### 2.3 Geräteadresse

Die Geräteadresse des Slaves ist zwischen 1 und 255 einstellbar. Die Geräteadresse 0 ist reserviert.



Über die RS422-/RS485-Schnittstelle können maximal 31 Slaves angesprochen werden.

### 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.

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

$$\text{Zeichenübertragungszeit [ms]} = 1000 * 9 \text{ Bits/Baudrate}$$

Bei den anderen Datenformaten ergibt sich:

$$\text{Zeichenübertragungszeit [ms]} = 1000 * 10 \text{ Bits/Baudrate}$$

#### Ablauf

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

## 2 Protokollbeschreibung

---

### Beispiel

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

Wartezeit = 3 Zeichen \* 1000 \* 10 Bits/Baudrate[

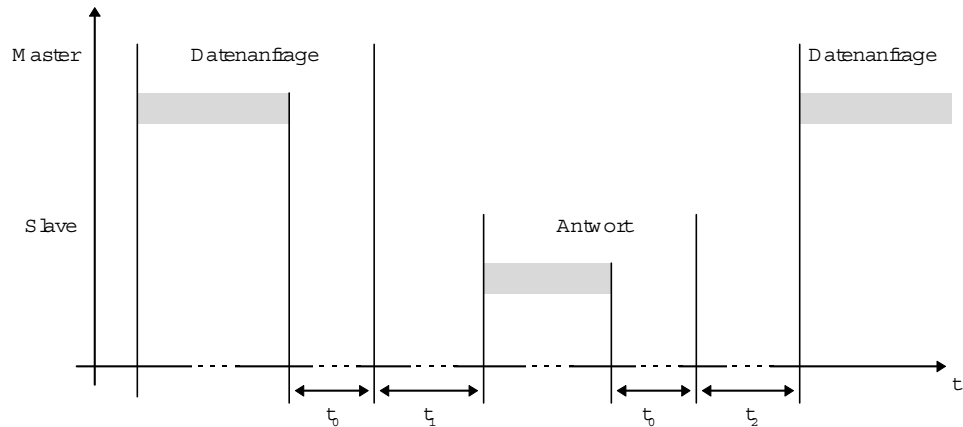
Baudrate [Baud]	Datenformat [Bit]	Wartezeit [ms]
187k	10	0,160
	9	0,144
125k	10	0,240
	9	0,216
38400	10	0,781
	9	0,703
19200	10	1,563
	9	1,406
9600	10	3,125
	9	2,813
4800	10	6,250
	9	5,625
2400	10	12,500
	9	11,250
1200	10	25,000
	9	22,500
600	10	50,000
	9	45,000
300	10	100,000
	9	90,000
150	10	200,000
	9	180,000



## 2 Protokollbeschreibung

### 2.4.1 Zeitlicher Ablauf einer Datenanfrage

**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.



In dem Gerät kann unter dem Menüpunkt „Schnittstelle“ eine minimale Antwortzeit eingestellt werden. Diese eingestellte Zeit wird mindestens eingehalten, bevor eine Antwort gesendet wird (0...500 ms). Wird ein kleiner Wert eingestellt, so kann die Antwortzeit größer sein als der eingestellte Wert (die interne Bearbeitungszeit ist länger), der Gerät antwortet dann unmittelbar nachdem die interne Bearbeitung abgeschlossen ist. Eine eingestellte Zeit von 0 ms bedeutet, daß der Gerät mit der maximal möglichen Geschwindigkeit antwortet.

Die minimal einstellbare Antwortzeit wird bei der RS485-Schnittstelle vom Master benötigt, um die Schnittstellentreiber von Senden auf Empfangen umzustellen. Bei der RS422-Schnittstelle wird dieser Parameter nicht benötigt.

- $t_2$  Diese Zeit braucht der Gerät, 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.

RS422-Schnittstelle:  $t_2 = 1\text{ms}$

RS485-Schnittstelle:  $t_2 = 10\text{ms}$

## 2 Protokollbeschreibung

---

### 2.4.2 Kommunikation während der internen Bearbeitungszeit des Slaves

Während der internen Bearbeitungszeit des Slaves dürfen vom Master keine Datenanfragen gestellt werden. In dieser Zeit gestellte Datenanfragen werden vom Slave ignoriert.

### 2.4.3 Kommunikation während der Antwortzeit des Slaves

Während der Antwortzeit des Slaves dürfen vom Master keine Datenanfragen gestellt werden. In dieser Zeit gestellte Datenanfragen führen dazu, daß alle gerade auf dem Bus befindlichen Daten ungültig werden.

## 2.5 Aufbau der Datenblöcke

Alle Datenblöcke haben die gleiche Struktur:

### Datenstruktur

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

Jeder Datenblock enthält vier Felder:

**Slave-Adresse** Geräteadresse eines bestimmten Slaves

**Funktionscode** Funktionsauswahl (Lesen, Schreiben, Bit, Wort)

**Datenfeld** Enthält die Informationen:  
-Bitadresse (Wortadresse)  
-Bitanzahl (Wortanzahl)  
-Bitwert (Wortwert)

**Checksumme** Erkennung von Übertragungsfehlern

## 2.6 Fehlerbehandlung

**Fehlercodes** Es existieren fünf Fehlercodes:

- 1 ungültige Funktion
- 2 ungültige Parameteradresse
- 3 Parameterwert außerhalb Wertebereich
- 4 Slave nicht bereit
- 8 Schreibzugriff auf Parameter verweigert

## 2 Protokollbeschreibung

### 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 gesetzt, d. h., das MSB (most significant bit, engl. das höchstwertige Bit) wird auf 1 gesetzt.

### Beispiel

Datenanfrage:

01	02	00	00	00	00	CRC16
----	----	----	----	----	----	-------

Antwort:

01	82	01	CRC16
----	----	----	-------

### Sonderfälle

In folgenden Fehlerfällen antwortet der Slave nicht:

- die Checksumme (CRC16) ist nicht korrekt
- die Anweisung des Masters ist unvollständig oder überdefiniert
- die Anzahl der zu lesenden Worte oder Bits ist Null
- das Gerät wird momentan über Tastatur konfiguriert

## 2.7 Unterscheidung MOD-Bus/J-Bus

Das MOD-Bus-Protokoll ist zu dem J-Bus-Protokoll kompatibel. Die Struktur der Datenblöcke ist identisch.



Der Unterschied zwischen MOD-Bus und J-Bus besteht darin, daß die absoluten Adressen der Daten verschieden sind. Die Adressen des MOD-Bus sind gegenüber denen des J-Bus um eins verschoben.

Absolute Adresse	Adresse J-Bus	Adresse MOD-Bus
1	1	0
2	2	1
3	3	2
...	...	...

## 2 Protokollbeschreibung

### 2.8 Checksumme (CRC16)

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

#### Berechnungs- schema

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);	



Das Low-Byte der Checksumme wird zuerst übertragen.

#### Beispiel 1

Datenanfrage: Lesen von zwei Worten ab Adresse 1 (CRC16 = 0x0E97)

14	03	00	01	00	02	97	0E
----	----	----	----	----	----	----	----

Antwort: (CRC16 = 0x953E)

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

#### Beispiel 2

Anweisung: Setze Bit auf Adresse 24 (CRC16 = 0xF80E)

14	05	00	18	FF	00	0E	F8
----	----	----	----	----	----	----	----

Antwort (wie Anweisung):

14	05	00	18	FF	00	0E	F8
----	----	----	----	----	----	----	----

## 2 Protokollbeschreibung

### 2.9 Schnittstelle

Konfiguration der Schnittstelle am Gerät.

Anzeige	editieren	Auswahl/Eingabe	mit Ta- sten	übernehmen	weiter mit
SCHNITTSTELLE	<input type="text" value="ENTER"/>	PROTOKOLL: <u>J-BUS</u> Protokoll einstellen: J-BUS MOD-BUS	<input type="button" value="▲"/> , <input type="button" value="▼"/>	<input type="text" value="ENTER"/>	⇒ 2
	⇒ 2	BAUD: <u>9,6</u> kBaud Baudrate wählen (Anga- ben in kBaud): 0.15, 0.3, 1.2, 2.4, 4.8, 9.6, 19.2, 38.4, 12.5, 187.5kBaud	<input type="button" value="▲"/> , <input type="button" value="▼"/>	<input type="text" value="ENTER"/>	⇒ 3
	⇒ 3	DATENFORMAT: <u>8/1/KEINE</u> Datenformat wählen (Da- tenbits/Stopbits/Parität): 8/1/KEINE, 8/1/UNGER., 8/1/GERADE, 8/2 KEINE, 8/1/NULL	<input type="button" value="▲"/> , <input type="button" value="▼"/>	<input type="text" value="ENTER"/>	⇒ 4
	⇒ 4	ADRESSE: <u>001</u> Adresse wählen: 1 ... 255	<input type="button" value="▲"/> , <input type="button" value="▼"/> , <input type="button" value="◀"/> , <input type="button" value="▶"/>	<input type="text" value="ENTER"/> Überprü- fung: $1 \leq \text{Adresse}$ $\leq 255$	⇒ 5
	⇒ 5	MIN. ANTWORTZEIT: <u>000</u> ms minimale Antwortzeit wählen: 0 ... 999ms	<input type="button" value="▲"/> , <input type="button" value="▼"/> , <input type="button" value="◀"/> , <input type="button" value="▶"/>	<input type="text" value="ENTER"/> Überprü- fung: $0 \leq$ Antw.zeit $\leq$ 999	<input type="button" value="▲"/> vorwärts <input type="button" value="▼"/> rückwärts



Bei der Skalenvariante Logoline 500 wird die Schnittstelle über das Setup-Programm konfiguriert.

### 3 Funktionen

---

Die folgenden Funktionen stehen für das Gerät zur Verfügung:

<b>Funktionsnummer</b>	<b>Funktion</b>	
0x01/0x02	Lesen von n Bits	(max. 256 Bits)
0x03/0x04	Lesen von n Worten	(max. 80 Worte)
0x05	Schreiben eines Bits	
0x06	Schreiben eines Worts	
0x0F	Schreiben von n Bits	(max. 256 Bits)
0x10	Schreiben von n Worten	(max. 80 Worte)

Für die Systemvariablen sind keine gesonderten Bereiche für Bit und Wort vorhanden. Die Bereiche Bit und Wort sind überlappend und können sowohl als Bit-Bereich als auch als Wort-Bereich gelesen und geschrieben werden.

Die Bit-Adresse wird wie folgt berechnet:

$$\text{Bit-Adresse} = \text{Wort-Adresse} * 16 + \text{Bitnummer}$$

## 3 Funktionen

### 3.1 Lesen von n Bits

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

#### Datenanfrage

Slave-Adresse	Funktion 0x01 oder 0x02	Adresse erstes Bit	Bitanzahl	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

#### Antwort

Slave-Adresse	Funktion 0x01 oder 0x02	Anzahl gelesener Bytes	Bitwerte	Checksumme CRC16
1 Byte	1 Byte	1 Byte	x Byte(s)	2 Bytes

#### Beispiel

Lesen der Stellung der ersten 4 binären Eingänge (Prozeßdaten),  
⇒ Kapitel 6.1.2 „Prozeßdaten“

$$\begin{aligned}\text{Bitadresse} &= (\text{Basisadresse} + \text{Prozeßdatenadresse}) * 16 \\ &= (0x002F + 0x0008) * 0x10 = 0x0370\end{aligned}$$

#### Datenanfrage

0A	01	03	70	00	04	CRC16
----	----	----	----	----	----	-------

#### Antwort

0A	01	01	0F	CRC16
----	----	----	----	-------



Es werden immer, unabhängig von der Anzahl der zu lesenden Bits, mindestens 8 Bits (1 Byte) gelesen, da die Antwort in Bytes erfolgt.

In obigem Beispiel bedeutet das, daß die Bits 0x0370...0x0377 gelesen werden.

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

8 Bits = 1 Byte

Alle nicht relevanten Bits (0x0374...0x0377) werden mit dem Wert 0 beantwortet.

## 3 Funktionen

---

### 3.2 Lesen von n Worten

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

#### Datenanfrage

Slave-Adresse	Funktion 0x03 oder 0x04	Adresse erstes Wort	Wortanzahl	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(s)	2 Bytes

#### Beispiel

Lesen der 3 Meßeingänge

⇒ Kapitel 6.1.2 „Prozeßdaten“

Wortadresse = Basisadresse + Prozeßdatenadresse  
= 0x002F + 0x0002 = 0x0031

Datenanfrage:

14	03	00	31	00	06	CRC16
----	----	----	----	----	----	-------

Antwort:

14	03	10	1999	4348	4CCC	4348	2666	4396	CRC16
			Meßwert 1 200,1		Meßwert 2 200,3		Meßwert 3 300,3		



## 3 Funktionen

### 3.3 Schreiben eines Bits

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

#### Anweisung

Slave-Adresse	Funktion 0x05	Bitadresse	Bitwert XX 00	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

#### Antwort

Slave-Adresse	Funktion 0x05	Bitadresse	Bitwert XX 00	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes



Für den Bitwert gilt: FF00 = Bit setzen  
0000 = Bit löschen

#### Beispiel

Setze das Status Bit 0 des Datenblocks „Text zum Ausdrucken“  
⇒ Kapitel 6.2.1 „Text zum Ausdrucken“

Bitadresse = (Basisadresse + Adresse „Status der Datenstruktur“) \* 16 + Bitnummer  
= (0x0075 + 0x0) \* 0x10 + 0x0  
= 0x0750

Anweisung:

14	05	07	50	FF	00	CRC16
----	----	----	----	----	----	-------

Antwort (wie Anweisung):

14	05	07	50	FF	00	CRC16
----	----	----	----	----	----	-------

## 3 Funktionen

---

### 3.4 Schreiben eines Worts

Bei der Funktion Wortschreiben sind die Datenblöcke für Anweisung und Antwort 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

Schreibe Flag 1 für die Anzeigesteuerung (= 0x0001)

⇒ Kapitel 6.2.3 „Flags für die Anzeigesteuerung des Displays“

Wortadresse= Basisadresse + Adresse Flag 1

= 0x00E9 + 0x0002 = 0x00EB

Anweisung:

14	06	00	EB	00	01	CRC16
----	----	----	----	----	----	-------

Antwort (wie Anweisung):

14	06	00	EB	00	01	CRC16
----	----	----	----	----	----	-------

## 3 Funktionen

---

### 3.5 Schreiben von n Bits

#### Anweisung

Slave-Adresse	Funktion 0x0F	Adresse erstes Bit	Bit- anzahl	Byte- anzahl	Bitwert(e)	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	x Byte(s)	2 Bytes

#### Antwort

Slave-Adresse	Funktion 0x0F	Adresse erstes Bit	Bitanzahl	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes

#### Beispiel

Setze das Status Bit 0, Bit 1 und Bit 2 des Datenblocks „Text zum Ausdrucken“

Status Bit 0 = 1, Status Bit 1 = 0, Status Bit 2 = 1

⇒ Kapitel 6.2.1 „Text zum Ausdrucken“

Bitadresse = (Basisadresse + Adresse „Status der Datenstruktur“) \* 16 + Bitnummer  
= (0x0075 + 0x0) \* 0x10 + 0x0 = 0x0750

Anweisung:

14	0F	07	50	00	02	01	01	CRC16
----	----	----	----	----	----	----	----	-------

Antwort:

14	0F	07	50	00	02	CRC16
----	----	----	----	----	----	-------

## 3 Funktionen

---

### 3.6 Schreiben von n Worten

#### Anweisung

Slave-Adresse	Funktion 0x10	Adresse erstes Wort	Wortan- zahl	Byte- anzahl	Wort- wert(e)	Checksumme CRC16
1 Byte	1 Byte	2 Bytes	2 Bytes	1 Byte	x Byte(s)	2 Bytes

#### Antwort

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

#### Beispiel

Schreibe „Text zum Ausdrucken“  
(2 Worte: „ABC“ = 0x4142, 0x4300)  
⇒ Kapitel 6.2.1 „Text zum Ausdrucken“

Wortadresse= Basisadresse + Prozeßdatenadresse  
= 0x0075 + 0x0002 = 0x0077

Anweisung:

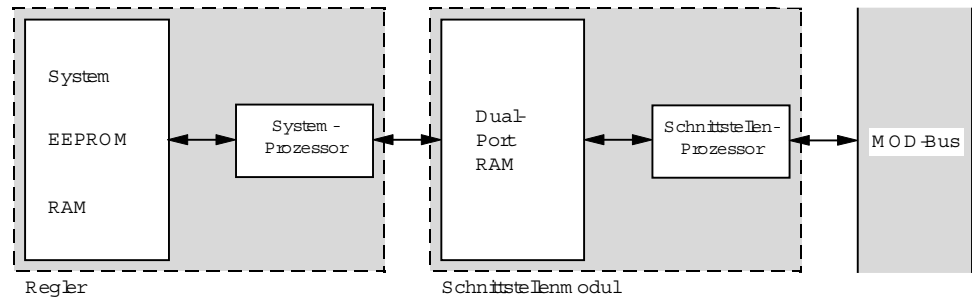
14	10	00	77	00	02	04	41	42	43	00	CRC16
----	----	----	----	----	----	----	----	----	----	----	-------

Antwort:

14	10	00	77	00	02	CRC16
----	----	----	----	----	----	-------

## 4 Datenfluß

---



Für die Datenübertragung zum MOD-Bus werden die Prozeßwerte in einem Dual-Port-RAM vom Systemprozessor bereitgestellt. Es werden nicht alle in dem Gerät vorhandenen Systemvariablen zyklisch im Dual-Port-RAM aufgefrischt. Das Dual-Port-RAM wird in zwei Bereiche unterteilt:

### System-variablen

Diese Variablen können vom MOD-Bus-Treiber direkt gelesen und geschrieben werden (zyklische Daten).

Diese Daten werden zyklisch (innerhalb der Abtastzeit) im Dual-Port-RAM aktualisiert.

### Daten nach Datenanfrage

Dieser Bereich wird nicht zyklisch von dem Systemprozessor aktualisiert (azyklische Daten).

Variablen aus diesem Datenbereich muß der MOD-Bus-Treiber anfordern.

Sie liegen erst nach der Bearbeitung durch den Systemprozessor bereit.

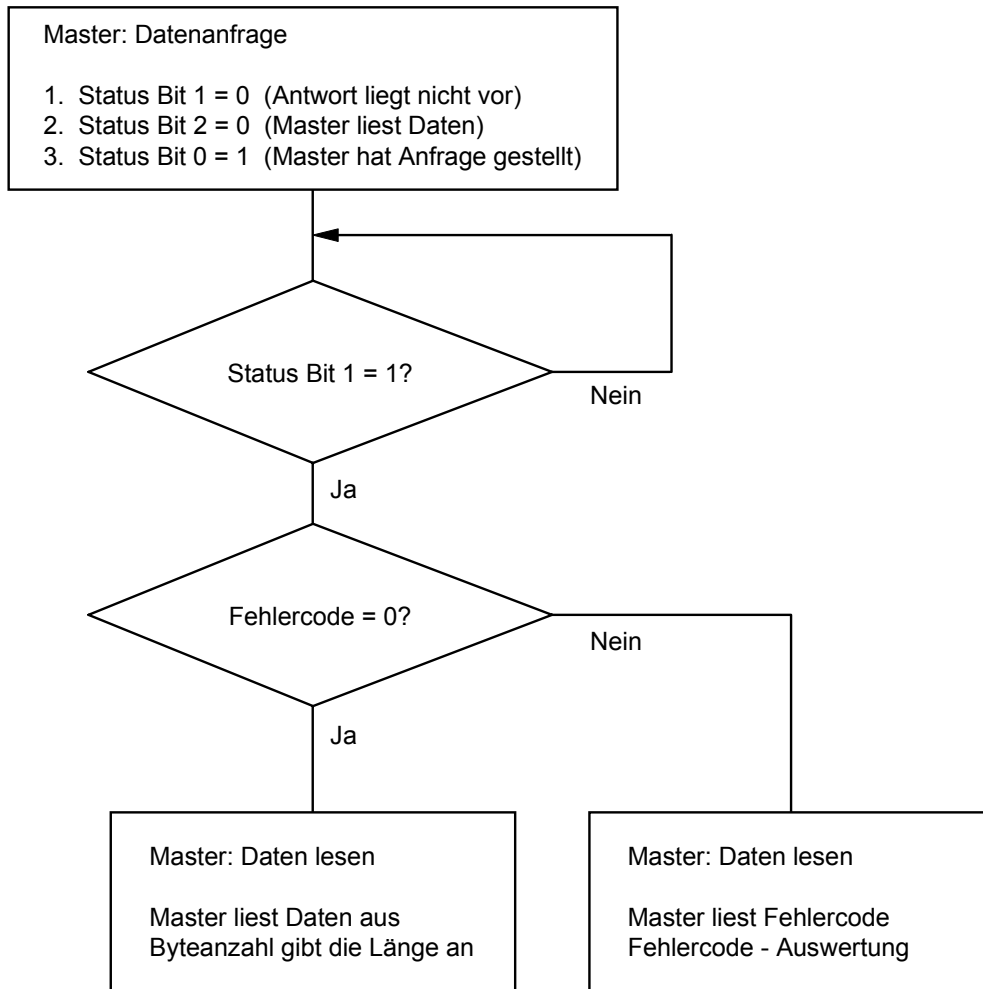


Für die im nachfolgenden aufgeführten Längenangaben von char-Datentypen gilt allgemein, daß es sich jeweils um die Zeichenkettenlänge inklusive Zeichenkettenende-Zeichen „\0“ handelt

## 4 Datenfluß

---

### 4.1 Azyklische Daten vom Gerät empfangen



## 4 Datenfluß

### Beispiel

Lesen des Reserve-Flags für die Anzeigensteuerung

⇒ Kapitel 6.2.3 „Flags für die Anzeigesteuerung des Displays“

**Schritt 1:** Die Datenstruktur „Flags für die Anzeigesteuerung“ wird angefordert

Status Bit 0 = 1, Status Bit 1 = 0 und Status Bit 2 = 0 setzen

MOD-Bus-Befehl: Schreibe 1 Wort

01	06	00	E9	00	01	CRC16
----	----	----	----	----	----	-------

Antwort:

01	06	00	E9	00	01	CRC16
----	----	----	----	----	----	-------

**Schritt 2:** Zyklische Abfrage, ob die entsprechende Datenstruktur zur Verfügung steht (Polling)

Status Bit 1 lesen

MOD-Bus-Befehl: Lese n Bits

01	01	0E	91	00	01	CRC16
----	----	----	----	----	----	-------

Antwort:

01	01	01	00	CRC16
----	----	----	----	-------

Status Bit 1 = 0 (Datenstruktur steht noch nicht zur Verfügung)

01	01	01	01	CRC16
----	----	----	----	-------

Status Bit 1 = 1 (Datenstruktur steht zur Verfügung)

**Schritt 3:** Fehlercode der angeforderten Struktur lesen

MOD-Bus-Befehl: Lese 1 Wort

01	03	00	EA	00	01	CRC16
----	----	----	----	----	----	-------

Antwort:

01	03	02	0000	CRC16
----	----	----	------	-------

Kein Fehler aufgetreten.

**Schritt 4:** Reserve-Flag abfragen

MOD-Bus-Befehl: Lese 1 Wort

01	03	00	EE	00	01	CRC16
----	----	----	----	----	----	-------

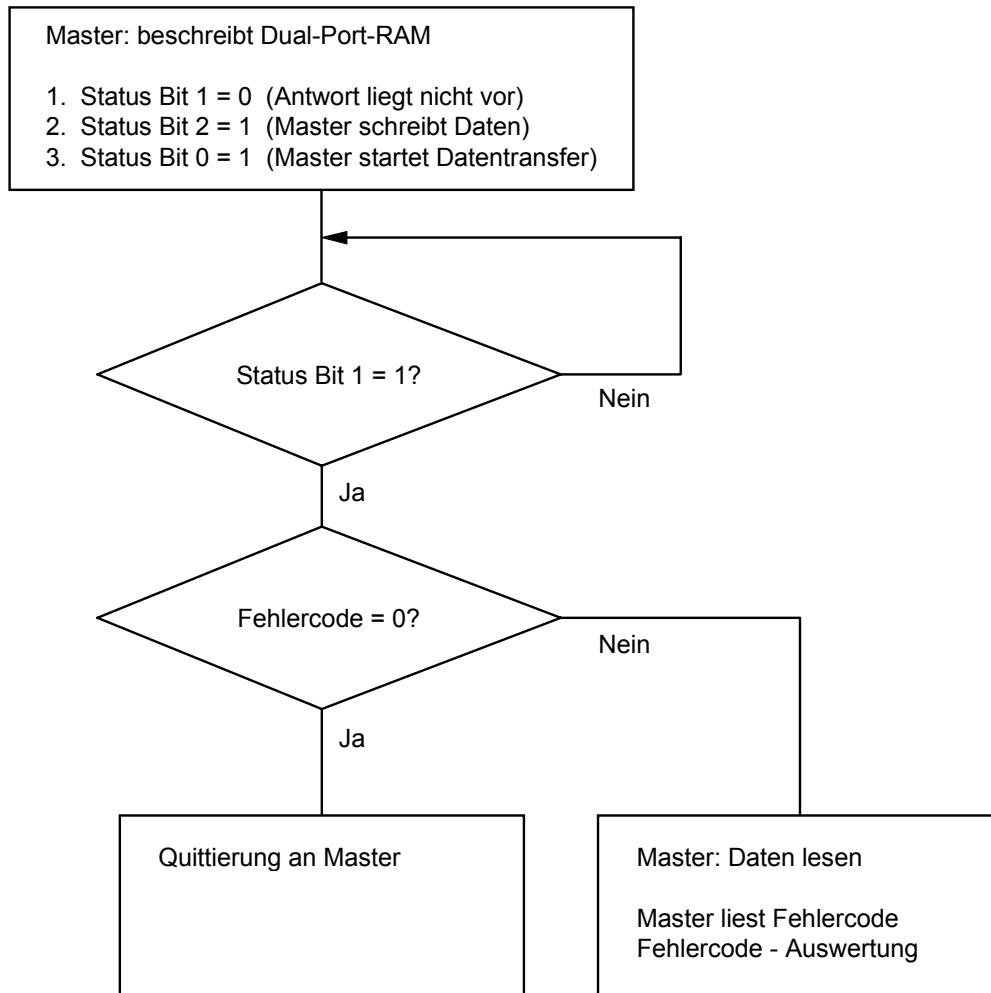
Antwort:

01	03	02	0001	CRC16
			Reserve-Flag ist gesetzt	

## 4 Datenfluß

---

### 4.2 Azyklische Daten zum Gerät übertragen





## 4 Datenfluß

---

### Beispiel

Schreiben eines float-Wertes für das Mathematikmodul  
(float-Wert 1 = 20,32)

**Schritt 1:** Die Datenstruktur wird angefordert

Status setzen: Bit0=1, Bit1 = 0, Bit2 = 0

MOD-BUS Befehl: Schreibe 1 Wort

01	06	00F9	0001	CRC16
----	----	------	------	-------

Antwort:

01	06	00F9	0005	CRC16
----	----	------	------	-------

**Schritt 2:** Zyklische Abfrage ob die entsprechende Datenstruktur zur Verfügung steht

Statusbit1 lesen

MOD-BUS Befehl: Lese n Bits

01	01	0F	91	00	01	CRC16
----	----	----	----	----	----	-------

Antwort:

01	01	01	00	CRC16
----	----	----	----	-------

Statusbit1 = 0 (Datenstruktur steht noch nicht zur Verfügung)

01	01	01	01	CRC16
----	----	----	----	-------

Statusbit1 = 1 (Datenstruktur steht zur Verfügung)

**Schritt 3:** Fehlercode der angeforderten Struktur lesen

MOD-BUS Befehl: Lese 1 Wort

01	03	00FA	0001	CRC16
----	----	------	------	-------

Antwort:

01	03	02	0000	CRC16
----	----	----	------	-------

Kein Fehler aufgetreten.

**Schritt 4:** Auf float-Wert 1 20,32 schreiben  
(20.32 entspricht 0x41A28F5C im IEEE-Format)

MOD-BUS Befehl: Schreibe 2 Worte

01	10	00FB	0002	04	8F5C	41A2	CRC16
----	----	------	------	----	------	------	-------

Antwort:

01	06	00FB	0002	CRC16
----	----	------	------	-------

## 4 Datenfluß

---

**Schritt 5:** Die Datenstruktur wird übertragen

Status setzen: Bit0 = 1, Bit1 = 0, Bit2 = 1

MOD-BUS Befehl: Schreibe 1 Wort

01	06	00F9	0005	CRC16
----	----	------	------	-------

Antwort:

01	06	00F9	0005	CRC16
----	----	------	------	-------

**Schritt 6:** Zyklische Abfrage ob die entsprechende Datenstruktur übertragen wurde

Statusbit1 lesen

MOD-BUS Befehl: Lese n Bits

01	01	00F9	00	01	CRC16
----	----	------	----	----	-------

Antwort:

01	01	01	00	CRC16
----	----	----	----	-------

Statusbit1 = 0 (Datenstruktur wurde noch nicht übertragen)

01	01	01	01	CRC16
----	----	----	----	-------

Statusbit1 = 1 (Datenstruktur wurde übertragen)

**Schritt 7:** Fehlercode der übertragenen Struktur lesen

MOD-BUS Befehl: Lese 1 Wort

01	03	00FA	0002	CRC16
----	----	------	------	-------

Antwort:

01	03	02	0000	CRC16
----	----	----	------	-------

Kein Fehler aufgetreten

## 4 Datenfluß

### 4.3 Übertragungsformat

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

z. B.: Schreiben des int-Wertes 1 (= 0x0001) auf die Adresse 0x00EB:  
010600EB**0001**383E

**Float-Werte** Bei float-Werten wird im MOD-Bus mit dem IEEE-754-Standard-Format (32 bit) gearbeitet, allerdings mit dem Unterschied, daß Byte 1 und 2 mit Byte 3 und 4 vertauscht sind.

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

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

S - Vorzeichen-Bit

E - Exponent (2er-Komplement)

M - 23Bit normalisierte Mantisse

#### Modbus-float-Format

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

z. B.: Schreiben des float-Wertes 550.0  
(= 0x44098000 im IEEE-754-Format) auf die Adresse 0x00FB:  
011000FB000204**80004409**679E

Vor bzw. nach der Übertragung zum bzw. vom Gerät müssen die Bytes des float-Wertes entsprechend vertauscht werden.

Viele Compiler (z. B. Microsoft C++, Turbo C++, Turbo Pascal, Keil C51) legen die float-Werte in folgender Reihenfolge ab:

#### float-Wert

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

# 5 Fehlermeldungen

---

## 5.1 Fehlermeldungen der Schnittstelle

Die Fehlernummern sind in den Datenblöcken der azyklischen Daten unter Fehlercode zu finden.

⇒ Kapitel 6.2.1 - Kapitel 6.2.4

Fehlercode	Fehler Setup-Auftragsbearbeitung
0x0014	Auftrags-Busyflag von Master nicht zurückgesetzt
0x0015	Auftrag unzulässig
0x0016	Fehler bei der Datenübernahme
0x0017	keine zyklischen Daten vorhanden
0x0018	Strukturlänge unzulässig
0x0019	Kopf nicht zulässig
0x001C	Schreibfehler in das serielle EEPROM (Kalib)

## 5.2 Fehlermeldungen bei ungültigen Werten

Für Soll-/Istwerte und daraus berechnete Werte gilt die Vereinbarung, daß die Fehlernummer im Wert selbst dargestellt wird, d. h. anstatt des Meßwertes ist die Fehlernummer eingetragen.

Fehler-nummer	Fehler
200000.0	Meßbereichsüber- oder -unterschreitung

# 5 Fehlermeldungen

---

## 5.3 System- und Laufzeitfehler

Die System- bzw. Laufzeitfehler sind Teil der Prozeßdaten (zyklischen Daten).

⇒ Kapitel 6.1.2

Die dort eingetragenen Fehlernummern haben folgende Bedeutung:

### Systemfehler

Tritt einer der folgenden Fehler auf, so wird die Meßwerterfassung und Registrierung unterbrochen. Alle anderen Prozeßdaten sind nicht mehr gültig. Tritt der Fehler Nummer 13 auf, so befindet sich das Gerät noch in der Initialisierungsphase, während der noch keine gültigen Prozeßdaten zur Verfügung stehen. Die Anweisung an das Gerät sollte nach kurzer Zeit wiederholt werden.

Fehler-nummer	Fehler
0	kein Fehler
1	Reserviert
2	Reserviert
3	Reserviert
4	EEPROM defekt
5	Reserviert
6	Reserviert
7	Reserviert
8	A/D-Wandler defekt
9	Reserviert
10	Hallsensor defekt
11	Reserviert
12	Reserviert
13	Initialisierungsphase

## 5 Fehlermeldungen

### Laufzeitfehler

Die Reaktion des Gerätes bei Auftreten von einem oder mehreren der folgenden Fehler lesen Sie bitte in der Betriebsanleitung B 95.3530 nach.

Alle anderen Daten können weiterhin vom Gerät abgefragt werden.

Fehler-nummer	Fehler
0	kein Fehler
1	Reserviert
2	Reserviert
3	Kein Papier
4	Relaisbaugruppe antwortet nicht
5	Uhrzeit muß neu eingestellt werden
6	Batterie leer
7	Reserviert
8	Reserviert



Es können mehrere Laufzeitfehler gleichzeitig auftreten. Es wird immer der Laufzeitfehler mit der höchsten Priorität angezeigt.

Ereignis	Priorität
Kein Papier	höher ↑ ↓ niedriger
Relaisbaugruppe antwortet nicht	
Uhrzeit muß neu eingestellt werden	
Batterie leer	
Kein Fehler	

Wenn z. B. das Papier aufgebraucht und die Batterie leer ist, wird Fehler Nummer 3 (kein Papier) als Laufzeitfehler angegeben.

## 6 Adresstabellen

---

Im folgenden sind alle Prozeßwerte (Variablen) mit ihren Adressen, dem Datentyp und der Zugriffsart beschrieben.

Hierbei bedeutet:

R/O	Zugriff nur lesend
R/W	Zugriff schreibend und lesend
char	ASCII-Zeichen (8 Bits)
byte	Byte (8 Bit)
int	Integer (16 Bits)
char xx	Zeichenkette mit Länge xx; xx = Länge inklusive Zeichenkettenende-Zeichen „\0“
Bit x	Bit Nr. x
float	Float-Wert (4 Byte)

Die Prozeßwerte sind in logische Bereiche unterteilt.

Die absolute MOD-Bus-Adresse ergibt sich aus der Basisadresse des jeweiligen Bereichs und dem Adressoffset.

In den nachfolgenden Adresstabellen ist Bit 0 immer das niederwertigste Bit

## 6 Adresstabellen

---

### 6.1 Zyklische Daten

#### 6.1.1 Gerätedaten

Basisadresse: 0x0000

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/O	int	Gerätegruppe (7)
0x0001	R/O	int	Gerätetyp (0)
0x0002	R/O	char 9	Gerätename („L500/540“)
0x0007	R/O	char 11	Software-Version
0x000D	R/O	char 13	VdN-Nummer
0x0014	R/O	char 10	Fabrikationsnummer
0x0019	R/O	char 15	Datum/Uhrzeit letzte Änderung Konfiguration
0x0021	R/O	char 15	Datum/Uhrzeit letzte Änderung Parameter
0x0029	R/O	12 Byte	Reserve

#### 6.1.2 Prozeßdaten

Basisadresse: 0x002F

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/O	int	Systemfehler ⇒ Kapitel 5 „Fehlermeldungen“
0x0001	R/O	int	Laufzeitfehler ⇒ Kapitel 5.3 „System- und Laufzeitfehler“
0x0002	R/O	float	Meßeingang 1
0x0004	R/O	float	Meßeingang 2
0x0006	R/O	float	Meßeingang 3
0x0008	R/O	int	Stellung der binären Eingänge: 0 = offen / 1 = geschlossen
	R/O	Bit0	Binäreingang 1
	R/O	Bit1	Binäreingang 2
	R/O	Bit2	Binäreingang 3
	R/O	Bit3	Binäreingang 4
	R/O	Bit4	Binäreingang 5



## 6 Adresstabellen

Adresse	Zugriff	Datentyp	Signalbezeichnung
	R/O	Bit5	Binäreingang 6
	R/O	Bit6	Binäreingang 7
	R/O	Bit7	Binäreingang 8
	R/O	Bit8...15	frei
0x0009	R/O	int	Relaisstellungen: 0 = nicht aktiv / 1 = aktiv
	R/O	Bit0	Relaisausgang 1
	R/O	Bit1	Relaisausgang 2
	R/O	Bit2	Relaisausgang 3
	R/O	Bit3	Relaisausgang 4
	R/O	Bit4	Relaisausgang 5
	R/O	Bit5	Relaisausgang 6
	R/O	Bit6	Relaisausgang 7
	R/O	Bit7	Relaisausgang 8
	R/O	Bit8...15	frei
0x000A	R/O	float	Ereigniszähler 1
0x000C	R/O	float	Ereigniszähler 2
0x000E	R/O	int	aktuelle Papiervorschubgeschwindigkeit
0x000F	R/O	int	Stopp-Zustand: 0 = kein Stop / 1 = Stopp
0x0010	R/O	int	Papierende: 0 = Papier vorhanden / 1 = kein Papier
0x0011	R/O	char 20	aktuelles Datum und Uhrzeit („jjjj-mm-dd-hh:mm:ss“)
0x001B	R/O	char 20	letztes Ausschalten des Schreibers (Datum/Uhrzeit)
0x0025	R/O	char 20	letztes Einschalten des Schreibers (Datum/Uhrzeit)
0x002F	R/O	int	Anzahl der Netzabschaltungen
0x0030	R/O	int	Anzahl der Betriebsstunden
0x0031	R/O	int	Polling-Flag für den übertragenen Text zum Ausdrucken: 0 = keine Druckanforderung 1 = Druckanforderung für den Text steht noch an

## 6 Adresstabellen

### 6.2 Azyklische Daten

#### Status-Bits

Die nachfolgenden Adresstabellen (azyklische Daten) werden erst auf Datenanfrage des MOD-Bus-Treibers gelesen oder geschrieben. Der Zustand dieser Adresstabellen (azyklische Daten) wird im Statuswort angezeigt. Statuswort und Fehlercode liegen jeweils am Anfang der Datenblöcke der azyklischen Daten.

Status			Bedeutung für den Master
Bit 0	Bit 1	Bit 2	
0	0	X	Master hat keine Datenanfrage gestellt
1	0	X	Master hat eine Datenanfrage an das Gerät gestellt
		X	die Datenanfrage wird bearbeitet
0	1	X	Die Bearbeitung ist zu Ende, die Antwort steht
		X	im Puffer für den Master bereit
1	1	X	nicht gültig
		0	Datenübertragung vom Dual-Port-RAM zum Gerät
		1	Datenübertragung vom Gerät zum Dual-Port-RAM
Bit 3... Bit 15 werden nicht verwendet			

#### 6.2.1 Text zum Ausdrucken

Basisadresse: 0x0075

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/W	int	Status der Datenstruktur
	R/W	Bit0	1 = Datenanfrage bearbeiten
	R/W	Bit1	0 = Antwort noch nicht da / 1 = Bearbeitung beendet, Antwort steht im Puffer
	R/W	Bit2	0 = Übertragung Schnittstelle → Schreiber 1 = Übertragung Schreiber → Schnittstelle
		Bit3...15	frei
0x0001	R/W	int	Fehlercode ⇒ Kapitel 5 „Fehlermeldungen“
0x0002	R/W	char36	Text für den Ausdruck auf das Papier

## 6 Adresstabellen

### 6.2.2 Text zum Anzeigen im Display

Basisadresse: 0x009D

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/W	int	Status der Datenstruktur
	R/W	Bit0	1 = Datenanfrage bearbeiten
	R/W	Bit1	0 = Antwort noch nicht da 1 = Bearbeitung beendet, Antwort steht im Puffer
	R/W	Bit2	0 = Übertragung Schnittstelle → Schreiber 1 = Übertragung Schreiber → Schnittstelle
		Bit3...15	frei
0x0001	R/W	int	Fehlercode ⇒ Kapitel 5 „Fehlermeldungen“
0x0002	R/W	char 36	Text 1 zum Anzeigen im Display
0x0014	R/W	char 36	Text 2 zum Anzeigen im Display
0x0026	R/W	char 36	Text 3 zum Anzeigen im Display

### 6.2.3 Flags für die Anzeigesteuerung des Displays

Basisadresse: 0x00E9

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/W	int	Status der Datenstruktur
	R/W	Bit0	1 = Datenanfrage wird bearbeitet
	R/W	Bit1	0 = Antwort noch nicht da 1 = Bearbeitung beendet, Antwort steht im Puffer
	R/W	Bit2	0 = Übertragung Schnittstelle → Schreiber 1 = Übertragung Schreiber → Schnittstelle
		Bit3...15	frei
0x0001	R/W	int	Fehlercode ⇒ Kapitel 5 „Fehlermeldungen“
0x0002	R/W	int	Flag 1: 0 = Text 1 nicht anzeigen 1 = Text 1 anzeigen

## 6 Adresstabellen

---

0x0003	R/W	int	Flag 2: 0 = Text 2 nicht anzeigen 1 = Text 2 anzeigen
0x0004	R/W	int	Flag 3: 0 = Text 3 nicht anzeigen 1 = Text 3 anzeigen
0x0005	R/W	int	Reserve

### 6.2.4 float-Werte für das Mathematikmodul

Basisadresse: 0x00F9

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/W	int	Status der Datenstruktur
	R/W	Bit0	1 = Datenanfrage wird bearbeitet
	R/W	Bit1	0 = Antwort noch nicht da / 1 = Bearbeitung beendet, Antwort steht im Puffer
	R/W	Bit2	0 = Übertragung Schnittstelle → Schreiber 1 = Übertragung Schreiber → Schnittstelle
		Bit3...15	frei
0x0001	R/W	int	Fehlercode ⇒ Kapitel 5 „Fehlermeldungen“
0x0002	R/W	float	float-Wert 1 = frei verwendbar im Mathematikmodul
0x0004	R/W	float	float-Wert 2 = frei verwendbar im Mathematikmodul
0x0006	R/W	float	float-Wert 3 = frei verwendbar im Mathematikmodul

## 6 Adresstabellen

---

### 6.2.5 Reset-Erkennung

Basisadresse: 0x0406

Adresse	Zugriff	Datentyp	Signalbezeichnung
0x0000	R/W	int	Reset-Erkennung der Schnittstellenbaugruppe

Diese Speicherstelle kann für eine Reset-Erkennung der Schnittstellenbaugruppe verwendet werden. Es muß ein Wert ungleich Null eingetragen und dieser dann zyklisch abgefragt werden. Tritt an der Schnittstellenbaugruppe des Schreibers ein Reset auf, so wird das Byte gelöscht (auf Null gesetzt). Durch die zyklische Abfrage kann somit ein Reset erkannt werden.

Ein Reset der Schnittstellenbaugruppe kann folgende Ursachen haben:

- Netz-Aus
- Konfiguration des Gerätes über die Tastatur  
(wenn Codewort richtig eingegeben wurde)
- Setup-Stecker ist am Gerät gesteckt



**MESS- UND REGELTECHNIK**

M. K. JUCHHEIM GmbH & Co

36035 Fulda

Germany

Telefon (06 61) 60 03-7 25

Telefax (06 61) 60 03-6 81

Telex 49701 juf d

email JUMO\_de@e-mail.com