



## Multifunktionseinheit KS 98-1

# KS 98-1 MODBUS

**KS98-1 Modbus**

**Beschreibung der Funktionalität  
anhand von Beispielen**

**9499-040-88718**

Gültig ab: 06/2007

---

## Erklärung der Symbole:



**Information allgemein**



**Warnung allgemein**



**Achtung: ESD-gefährdete Bauteile**

---

MODBUS®

ist ein eingetragenes Warenzeichen der Modbus-IDA Organisation

BluePort® und BlueControl®

sind eingetragene Warenzeichen der PMA Prozeß- und Maschinen-Automation GmbH

© 2007 PMA Prozeß- und Maschinen-Automation GmbH • Printed in Germany • Alle Rechte vorbehalten • Ohne vorhergehende schriftliche Genehmigung ist der Nachdruck oder die auszugsweise fotomechanische oder anderweitige Wiedergabe dieses Dokumentes nicht gestattet.

Dies ist eine Publikation von  
PMA Prozeß- und Maschinen Automation  
Postfach 310229  
D-34058 Kassel  
Germany

# Inhaltsverzeichnis

<b>1. Allgemeines</b>	<b>5</b>
<b>2. Busprotokoll</b>	<b>6</b>
2.1 Aufbau eines Übertragungsbytes	6
2.2 Genereller Nachrichtenaufbau	6
2.2.1 CRC	7
2.2.2 Endekennung	7
2.3 Sendeprozess	7
2.4 Funktionscodes	7
2.4.1 Lesen von mehreren Werten	8
2.4.2 Schreiben eines einzelnen Wertes	8
2.4.3 Schreiben mehrerer Werte	9
2.4.4 Lesen und Schreiben von Daten im Blockformat	10
2.5 Fehlerprotokoll	11
2.5.1 Fehlercodes	11
<b>3. Modbus-Definitionen</b>	<b>12</b>
3.1 Implementierte Modbus-Adressen	12
3.2 Implementierte Modbus-Funktionscodes	12
<b>4. Datenstrukturen</b>	<b>13</b>
4.1 Gerätedaten	13
4.1.1 Definitionen	13
4.1.2 Beispiele	17
4.2 L1WRIT / L1READ	18
4.2.1 Struktur für L1WRIT	18
4.2.2 Struktur für L1READ	18
4.2.3 Beispiel	19
4.3 MBDATA-Strukturen	19
4.3.1 Definitionen	19
4.3.2 Beispiel	19
4.4 Modbusadressen für Regler	20
4.4.1 Struktur für CONTR, CONTR+, PIDMA	20
4.4.2 Beispiel	22
4.5 Modbusadressen für Programmgeber	22
4.5.1 Struktur für APROG und DPROG	22
4.5.2 Beispiel	23
<b>5. 'B'-Schlüssel Nachrichten</b>	<b>24</b>
5.1 Funktionsblockprotokoll für I/O- und VTREND-Daten: Code B1	24
5.2 Funktionsblockprotokoll für Parameter: Code B2	25
5.3 Funktionsblockprotokoll für Anzeigetexte: Code B2	26
5.4 Funktionsblockprotokoll für Konfigurationsdaten: Code B3	27



**1****Allgemeines**

In diesem Dokument werden die Möglichkeiten und die Benutzung der neuen Modbus-Schnittstelle des KS98-1 anhand von Beispielen erläutert. Dabei werden nur die Möglichkeiten beschrieben, die nach einem erfolgreichen Engineering-Download notwendig sind.

Zuerst wird die Verteilung der Daten auf den verfügbaren Modbus-Adressraum beschrieben, anschliessend werden die direkt adressierbaren Daten erläutert und der Zugriff mit Beispielen für den Lese- und Schreibzugriff beschrieben.

Zuletzt wird der Zugriff auf die nicht einzeln adressierbaren Daten der Funktionsblöcke eines Engineering beschrieben. Hierzu wurde für die so genannten ‚B‘-Schlüssel-Nachrichten des ISO-Protokolls eine spezielle Blockübertragung des Modbusprotokolls verwendet.

In den Beispielen wird von einer Kommunikation über die Feldschnittstelle mit Adresse = 1 ausgegangen. Bei Verwendung der Frontschnittstelle muss die Adresse = 0 verwendet werden (diese gilt auf der Frontschnittstelle nicht als Broadcastadresse!).

## 2 Busprotokoll

### 2.1 Aufbau eines Übertragungsbytes

Das MODBUS - Protokoll wurde ursprünglich zur Kommunikation zwischen einem Leitsystem und der Modicon®-Steuerung definiert. Genutzt wird eine Master-Slave-Struktur, in der nur ein Gerät (Master) Datentransaktionen (Queries) auslösen kann. Weitere Geräte (Slaves) beantworten die Nachricht (Response) des Masters mit den angefragten Daten.

Der Master kann gezielt einen Slave über dessen MODBUS-Adresse ansprechen oder über eine allgemeine Nachricht (Broadcast) alle angeschlossenen Slaves benachrichtigen.

Das MODBUS-Protokoll bestimmt das Übertragungsformat der Datenanfrage und der Datenantwort. Die Funktionscodes definieren die auszuführende Aktionen in den Slaves.

Im Gerät wird das MODBUS-Protokoll im RTU (Remote Terminal Unit)- Mode genutzt, d. h. jedes gesendete Nachrichtenbyte enthält zwei hexadezimale Zeichen (0..9, A..F).

Der Aufbau eines Bytes im RTU-Protokoll ist folgendermaßen:

<b>Startbit</b>	<b>8-Datenbits</b>	<b>Paritäts-/Stoppbit</b>	<b>Stoppbit</b>
-----------------	--------------------	---------------------------	-----------------

### 2.2 Genereller Nachrichtenaufbau

Die Nachricht wird in einen Datenbuffer mit einer maximal definierten Länge eingelesen. Längere Nachrichten werden nicht akzeptiert. Es erfolgt keine Antwort durch das Gerät.

Die Nachricht setzt sich aus folgenden Elementen zusammen:

Geräteadresse	Funktionscode	Data	CRC	Endekennung
1 Byte	1 Byte	N * 1 Byte	2 Bytes	

- **Geräteadresse (Addr)**  
Die Geräteadresse spezifiziert das Gerät. Geräteadressen können im Bereich von 1 - 127 vergeben werden. Die Geräteadresse 0 wird als Broadcast-Message verwendet. Eine Broadcast-Message kann für Schreibaufträge vergeben werden. Sie werden von allen Geräten am Bus ausgeführt. Da alle Geräte den Auftrag ausführen, erfolgt keine Antwort durch die Geräte.
- **Funktionscode**  
Der Funktionscode definiert den Typ einer Nachricht. Die MODBUS-Spezifikation definiert über 17 verschiedene Funktionscodes. Die von dem Gerät unterstützten Funktionscodes werden im Kapitel "Funktionscodes" (→ S. 7) beschrieben.
- **Data**  
Der Datenblock beinhaltet die weitere Spezifikation der Aktion, die mit dem Funktionscode definiert wird. Die Länge des Datenblocks ist abhängig vom Funktionscode.
- **CRC**  
Als weitere Fehlererkennung (nach der Paritätsbiterkennung) wird ein 16 Bit Cyclical Redundancy Check (CRC) durchgeführt. Der CRC-Code stellt sicher, das Übertragungsfehler erkannt werden können. Weitere Informationen siehe Kapitel "CRC" (→ S. 7).
- **Endekennung**  
Das Ende einer Nachricht wird definiert durch eine Zeit von 3,5 Zeichen, in der kein Datentransfer stattgefunden hat. Weitere Informationen siehe Kapitel "Endekennung" (→ S. 7).

 Weitere Informationen sind in den in [1] genannten Dokumenten oder unter <http://www.modbus.org> zu finden.

## 2.2.1 CRC

Bei dem CRC handelt es sich um ein 16-Bit Wert, der der Nachricht angehängt wird. Er dient zur Feststellung, ob die Übertragung einer Nachricht fehlerfrei erkannt wurde. Zusammen mit der Paritätskontrolle sollten alle möglichen Übertragungsfehler erkannt werden.

 Wird beim Empfang ein Paritätsfehler erkannt, so wird keine Antwortnachricht generiert.

Der Algorithmus zur Erzeugung des CRC ist folgendermaßen:

- ① CRC-Register mit  $FFFF_{hex}$  laden.
- ② Exklusiv- ODER Verknüpfung des ersten Sende/Empfangsbyte mit dem Low-Teil des CRC-Registers; Ergebnis in CRC speichern
- ③ CRC-Register um 1 Bit nach rechts schieben; MSB mit 0 auffüllen
- ④ Wenn das hinausgeschobene Bit eine 0 ist, dann Schritt 3 wiederholen.  
Wenn das hinausgeschobene Bit eine 1 ist, dann CRC-Register mit dem Wert  $A001_{hex}$  Exklusiv-ODER verknüpfen.
- ⑤ Schritt 3 und 4 für die anderen 7 Datenbits wiederholen.
- ⑥ Schritt 2 bis 5 für alle weiteren Sende/Empfangsbyte wiederholen.
- ⑦ Ergebnis des CRC-Registers an die Nachricht anhängen. Zuerst den Low-Teil, dann den High-Teil.  
Bei der Kontrolle einer Empfangsnachricht ergibt sich im CRC-Register eine 0, wenn die Nachricht inklusive des CRC bearbeitet wird.

## 2.2.2 Endekennung

Die Endekennung einer Nachricht ist spezifiziert als Ruhesituation auf dem MODBUS mit einer Länge von 3,5 Zeichen. Nach dem Verstreichen dieser Zeit darf ein Slave frühestens mit seiner Antwort beginnen oder ein Master frühestens eine neue Nachricht aussenden.

Die Auswertung einer Nachricht darf bereits beginnen, wenn erkannt wird, dass die Ruhebedingung auf den MODBUS für mehr als 1,5 Zeichen aufgetreten ist. Eine Antwort wird jedoch frühestens nach 3,5 Zeichen gestartet.

## 2.3 Sendeprozess

Es werden bei MODBUS zwei Übertragungsmodi angewandt:

- **Unicast - Modus**
- **Broadcast - Modus**

Im Unicast-Modus adressiert der Master ein individuelles Gerät, das nach Empfang der Nachricht diese abarbeitet und eine Antwort erzeugt. Die Geräteadresse kann von 1 bis 247 variieren. Eine Nachricht besteht immer aus einer Anfrage (request) und einer Antwort (response). Trifft keine Antwort innerhalb einer festzulegenden Zeit ein, so wird Timeout erkannt.

Im Broadcast-Modus sendet der Master einen Schreibbefehl (request) an alle Teilnehmer am Bus, die jedoch keine Antwort generieren. Die Adresse 0 ist für Broadcast-Nachrichten reserviert.

## 2.4 Funktionscodes

Funktionscodes dienen zur Ausführung von Befehlen. Folgende Funktionscodes werden vom Gerät unterstützt:

Funktionscode		Bezeichnung	Bedeutung
hex	dez		
0x03	3	Read Holding (Output) Register	Lesen von Prozessdaten, Parameters und Konfigurationsdaten
0x04	4	Read Input Register	Lesen von Prozessdaten, Parameters und Konfigurationsdaten
0x06	6	Preset Single Register (Output)	Wortweises Schreiben eines Wertes (Prozesswert, Parameter oder Konfiguration)
0x10	16	Preset Multiple Register (Output)	Wortweises Schreiben mehrerer Daten (Prozessdaten, Parameter oder Konfiguration)
0x17	23	Read/Write Multiple Register	Lesen und Schreiben von Daten im Blockformat

Die Funktionscodes 3 und 4 sind in ihrem Verhalten identisch.  
 In den nachfolgenden Kapiteln ist der Nachrichtenaufbau exemplarisch dargestellt.

### 2.4.1 Lesen von mehreren Werten

Nachrichten mit dem Funktionscodes 3 oder 4 dienen zum (wortweisen) Lesen von Prozessdaten, Parametern oder Konfigurationsdaten. Zum Lesen von Daten vom Typ Float müssen jeweils 2 Werte angefordert werden.  
 Der Aufbau einer Lese-Nachricht hat folgende Struktur:

Anfrage:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	03 oder 04	Lesen von Prozessdaten, Parametern oder Konfigurationsdaten
Startadresse High Startadresse Low	02 8A	Anfangsadresse 650
Anzahl der Werte	00 02	2 Daten (2 Worte)
CRC	CRC-Byte1 CRC-Byte2	

Antwort:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	03 oder 04	Lesen von Prozessdaten, Parametern oder Konfigurationsdaten
Anzahl der Bytes	04	Es werden 4 Datenbytes geschickt
Wort 1	00 DE	Prozessdaten, Parameter/Konfigurationsdate Adresse 650= 222
Wort 2	01 4D	Prozessdaten, Parameter/Konfigurationsdate Adresse 651= 333
CRC	CRC-Byte1 CRC-Byte2	



**Eine Broadcast - Nachricht ist für die Funktionscodes 3 und 4 nicht möglich.**



**Ist der erste adressierte Wert nicht definiert, so wird eine Fehlermeldung "ILLEGAL DATA ADDRESS" erzeugt.  
 Sind in dem auszulesenden Bereich nach dem ersten Wert andere nicht definiert, so werden diese mit dem Wert "NOT DEFINED VALUE" eingetragen. Dieses dient dazu, Bereiche mit Lücken mit einer Nachricht auslesen zu können.**

### 2.4.2 Schreiben eines einzelnen Wertes

Nachrichten mit dem Funktionscode 6 dienen zum wortweisen Schreiben von Prozessdaten, Parametern oder Konfigurationsdaten im Integerformat. Dieser Zugriff eignet sich nicht zum Schreiben von Daten vom Typ Float.  
 Der Aufbau einer Schreib-Nachricht hat folgende Struktur:

Anfrage:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	06	Schreiben eines einzelnen Wertes (Prozessdatum, Parameter oder Konfiguration)
Schreibadr. High Schreibadr. Low	02 8A	Schreibadresse 650

Wert	00 7B	Wertvorgabe = 123
CRC	CRC-Byte1 CRC-Byte2	

Antwort:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	06	Schreiben einer einzelnen Date (Ebene1, Parameter oder Konfiguration)
Schreibadr. High Schreibadr. Low	02 8A	Schreibadresse 650
Wert	00 7B	Wertvorgabe = 123
CRC	CRC-Byte1 CRC-Byte2	

Die Antwortnachricht entspricht bei Fehlerfreiheit exakt der Vorgabe.



**Die Geräte können diese Nachricht auch als Broadcast mit der Adresse 0 empfangen.**



**Eine Vorgabe im Datenformat Real ist nicht möglich, da als Wert nur 2 Byte übergeben werden können.**



**Ist ein Wert ausserhalb des einstellbaren Bereichs, so wird die Fehlermeldung "ILLEGAL DATA VALUE" erzeugt. Die Date bleibt unverändert.**

**Kann die Date nicht beschrieben werden (z.B. Konfigurationsdate und das Gerät befindet sich in Online), so wird eine Fehlermeldung "ILLEGAL DATA VALUE" erzeugt.**

### 2.4.3 Schreiben mehrerer Werte

Nachrichten mit dem Funktionscode 16 dienen zum (wortweisen Schreiben von Prozessdaten, Parametern oder Konfigurationsdaten. Zum Schreiben von Daten vom Typ Float müssen jeweils 2 Werte gesendet werden. Der Aufbau einer Schreib-Nachricht hat folgende Struktur:

Anfrage:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	10	Schreiben mehrerer Prozesswerte, Parameter oder Konfigurationsdaten
Startadresse High Startadresse Low	02 8A	Schreibadresse 650
Anzahl der Werte	00 02	2 Werte
Anzahl der Bytes	04	Es werden 4 Datenbytes geschickt
Wort 1	00 DE	Prozesswerte, Parameter oder Konfigurationsdate Adresse 650 = 222
Wort 2	01 4D	Prozess, Parameter oder Konfigurationsdate Adresse 651 = 333
CRC	CRC-Byte1 CRC-Byte2	

Antwort:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	10	Schreiben mehrerer Prozesswerte, Parameter oder Konfigurationsdaten
Startadresse High Startadresse Low	02 8A	Schreibadresse 650
Anzahl der Werte	00 02	2 Prozesswerte, Parameter/Konfigurationsdaten
CRC	CRC-Byte1 CRC-Byte2	



**Die Geräte können diese Nachricht auch als Broadcast mit der Adresse 0 empfangen.**



**Ist der erste Wert nicht definiert, so wird eine Fehlermeldung "ILLEGAL DATA ADDRESS" erzeugt. Kann der erste Wert nicht beschrieben werden (z.B. Konfiguration und Gerät ist in Online), so wird eine Fehlermeldung "ILLEGAL DATA VALUE" erzeugt.**

Sind in dem vorgegebenen Bereich nach dem ersten Wert andere nicht definiert oder momentan nicht beschreibbar, so werden diese überlesen. Daten werden an diesen Stellen nicht verändert. Dieses dient dazu Bereiche mit Lücken bzw. momentan nicht beschreibbaren Daten mit einer Nachricht verändern zu können. Es wird keine Fehlermeldung ausgegeben.

Sind Werte ausserhalb der einstellbaren Grenzen, so wird die Fehlermeldung "ILLEGAL DATA VALUE" erzeugt. Die Auswertung der nachfolgenden Daten wird nicht durchgeführt. Bereits fehlerfrei übernommenen Daten sind aktiv.

## 2.4.4 Lesen und Schreiben von Daten im Blockformat

Nachrichten mit dem Funktionscode 17 dienen zum Lesen und Schreiben von Daten im Blockformat. Der Dateninhalt besteht beim KS 98-1 grundsätzlich aus ASCII-Daten.

Anfrage:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	17	Lesen und Schreiben von Daten im Blockformat
Leseadresse High Leseadresse Low	0 0	Bei KS 98-1 ohne Bedeutung
Anzahl Lesedaten	0 1	=1. Bei KS 98-1 ohne Bedeutung, da sich die Anzahl der Lesedaten aus dem Inhalt der Schreibdaten ableitet
Schreibadresse High Schreibadresse Low	0 0	Bei KS 98-1 ohne Bedeutung
Anzahl Schreibdaten	0 n	Anzahl 'n' der Datenworte in den Schreibdaten
Anzahl Schreibdaten	2*n	Anzahl '2*n' der Datenworte in den Schreibdaten
Schreibdaten 1...n	x x ...	Datenblock der Schreibdaten
CRC	CRC-Byte1 CRC-Byte2	

Antwort:

Feldname	Wert (hex)	Bedeutung
Adresse	11	Adresse 17
Funktion	17	Lesen und Schreiben von Daten im Blockformat
Anzahl Antwortdaten	2*n	Anzahl '2*n' der Datenworte in den Schreibdaten

Feldname	Wert (hex)	Bedeutung
Antwortdaten 1...n	x x ...	Datenblock der Antwortdaten
CRC	CRC-Byte1 CRC-Byte2	



Eine Broadcast-Nachricht ist für Funktionscode 0x17 nicht möglich.

## 2.5

### Fehlerprotokoll

Das Fehlerprotokoll wird erzeugt, wenn eine Nachricht fehlerfrei empfangen wurde, die Interpretation der Nachricht oder die Änderung einer Date jedoch nicht möglich ist.



Wird ein Übertragungsfehler festgestellt, so wird keine Antwort erstellt. Der Master muss die Nachricht erneut abzusenden.

Erkannte Übertragungsfehler sind:

- Paritätsfehler
- Framing-Fehler (Kein Stoppbit empfangen)
- Overrun-Fehler (Empfangsbuffer ist übergelaufen oder Daten konnten nicht schnell genug vom UART abgeholt werden)
- CRC-Fehler

Der Datenaufbau des Fehlerprotokolls ist wie folgt:

Feldname	Wert	Bedeutung
Adresse	11	Adresse 17
Funktion	90	Fehlerprotokoll für die Nachricht Schreiben mehrerer Parameter/Konfigurationsdaten Aufbau: 80 <sub>hex</sub> + Funktionscode
Fehlercode	02	ILLEGAL DATA ADDRESS
CRC	CRC-Byte1 CRC-Byte2	

Im Feld Funktion wird das höchstwertigste Bit gesetzt.  
Im darauf folgenden Byte wird der Fehlercode übertragen.

### 2.5.1 Fehlercodes

Folgende Fehlercodes sind definiert:

Code	Name	Bedeutung
01	ILLEGAL FUNCTION	Der empfangen Funktionscode ist im Gerät nicht definiert.
02	ILLEGAL DATA ADDRESS	Die empfangene Adresse ist im Gerät nicht definiert oder der Wert ist nicht schreibberechtigt (read only). Werden mehrere Daten gleichzeitig gelesen (Funktionscode 01, 03, 04) oder geschrieben (Funktionscode 0F, 10), so wird dieser Fehler nur erzeugt, wenn die erste Date nicht definiert ist.
03	ILLEGAL DATA VALUE	Der empfangene Wert liegt ausserhalb der Einstellgrenzen oder kann momentan nicht beschrieben werden (Gerät befindet sich nicht im Konfigurationsmode). Werden mehrere Daten gleichzeitig geschrieben (Funktionscode 0F, 10), so wird dieser Fehler nur erzeugt, wenn die erste Date nicht beschrieben werden kann.
04	SLAVE DEVICE FAILURE	Es werden mehr Werte angefordert als der Übertragungsbuffer zulässt.

Weitere im Modbusprotokoll definierte Fehlercodes werden jedoch nicht unterstützt.

## 3 Modbus-Definitionen

### 3.1 Implementierte Modbus-Adressen

Der Modbusadressbereich umfasst die Adressen für den Zugriff auf die Daten im Integer- und Floatformat. Der Bereich 0x0001...0x3FFF steht für Daten im Integerformat zur Verfügung und der Bereich 0x8000...0xFFFF für Daten im Floatformat. Die Adresse einer Date im Floatformat berechnet sich aus der Adresse für das Integerformat \* 2 + 0x8000.

Folgende generelle Verteilung der Adressen ist festgelegt worden:

0x0001...0x004F	Allgemeine Geräte-Level-1-Daten und Geräteparameter
0x0050...0x0103	Level-1-Daten der L1READ / L1WRIT-Blöcke der Blocknummern 1...20 (je 9 Adressen)
0x0110...0x015F	5 Funktionsblöcke MBDATA (neu) mit je 16 Adressen)
0x0160...0x0687	30 Bereiche für die Level-1-Daten von Reglerfunktionsblöcken (CONTR / CONTR+ / PIDMA). (je 44 Adressen)
0x0688...0x09F7	40 Bereiche für die Level-1-Daten von Programmgeberfunktionsblöcken (APROG / DPROG). (je 22 Adressen)
0x8000...0xFFFF	Die im Bereich 0x0001...0x3FFF definierten Daten im Floatformat.

Die in einer Nachricht angegebene Modbusadresse muss immer definiert sein. Folgeadressen bei Nachrichten mit mehreren Daten müssen nicht aktiv sein. Beim Lesen wird dann der Abschaltwert (-32000 / -1.5e37) übertragen. Beim Schreiben werden nicht vorhandene Folge-Modbusadressen ignoriert.

### 3.2 Implementierte Modbus-Funktionscodes

Mit den Standardnachrichten werden einzelne oder mehrere Daten übertragen, denen direkt eine Modbusadresse zugeordnet werden kann. Dies sind die oben beschriebenen Geräte- und Level-1-Daten deren Modbusadresse über die Basis-Modbusadresse definiert wird.

Hierzu werden die auch in den anderen PMA-Geräten implementierten Modbus-Funktionscode benutzt, deren Aufbau dort bereits beschrieben ist.

3 bzw. 4	:	Lesen einzelner oder mehrerer Daten
6	:	Schreiben einzelner Daten
16	:	Schreiben mehrerer Daten

Für die Übertragung der im ISO1745-Mode per Funktionsblockprotokoll übertragenen Codes B1...B4 wird der Modbus-Funktionscode 23 (0x17) benutzt, der eine kombinierte Schreib/Lesenachricht bietet. Dieser Nachrichtentyp wird im Allgemeinen nur vom Engineeringtool benutzt.

4

Datenstrukturen

4.1

Gerätedaten

4.1.1

Definitionen

Die Gerätedaten benutzen die Modbusadressen 0x0001...0x004F

Adresse	Datum	Zugriff	Wertebereich
0x0001	Status 1	R	0 ... 63
0x0002	Status 2	R	0 ... 63

Status 1:

Bit 15...6 :	0		
Bit 5 :	Parameter Update	[0] - nein	[1] - ja
Bit 4 :	Power-Fail-Check	[0] - nicht aktiv	[1] - aktiv
Bit 3 :	E <sup>2</sup> PROM-Fehler	[0] - nein	[1] - ja
Bit 2 :	Sicherheitszustand	[0] - nicht aktiv	[1] - aktiv
Bit 1 :	Gerätezustand	[0] - Online	[1] - Konfiguration
Bit 0 (LSB) :	Sensorfehler (Sammelmeldung)	[0] - nein	[1] - ja

Status 2:

Bit 15...6 :	0		
Bit 5 :	Feldschnittstelle	[0] - Read/Write	[1] - Read
Bit 4 :	Hauptmenüanzeige durch Bedienung	[0] - möglich	[1] - blockiert
Bit 3 :	Konfigurationsanz. durch Bedienung	[0] - möglich	[1] - blockiert
Bit 2 :	Parameteranzeige durch Bedienung	[0] - möglich	[1] - blockiert
Bit 1 :	Verdrahtung beendet	[0] - nein	[1] - ja
Bit 0 (LSB) :	Engineering vorh.	[0] - nein	[1] - ja

Adresse	Datum	Zugriff	Wertebereich
0x0005	Gerätezustand	R/W	0 ... 1 / 0 ... 2
0x0006	Sicherheitszustand	R/W	0 ... 1
0x0007	Zurücksetzen der lok. Datenänderung	R/W	0 ... 1 / 0
0x0008	Engineering löschen	R/W	0 ... 1 / 1
0x0009	Verdrahtung beenden	R/W	0 ... 1 / 1
0x000A	Debug-Mode	R/W	0 ... 127
0x000B	Power-Fail-Check aktivieren	R/W	0 ... 1 / 1
0x000C	Schreiberlaubnis für Feldschnittstelle	R/W	0 ... 1

Gerätezustand:	0	-	Online
	1	-	Konfiguration (Offline)
	2	-	Abbruch Konfiguration (Esc) (Nur Schreiben)

Sicherheitszustand:	0	-	nicht aktiv
	1	-	aktiv

Zurücksetzen der lok. Datenänderung:

0	-	Parameter nicht geändert / Reset Flag
1	-	Parameter geändert (Nur Lesen)

Engineering löschen:	0	-	Engineering nicht gelöscht (Nur Lesen)
	1	-	Engineering gelöscht / löschen

Verdrahtung beenden:	0	-	Verdrahtung nicht beendet (Nur lesen)
	1	-	Verdrahtung beendet / beenden

Debug-Mode:	2 <sup>0</sup>	-	AINP1	(0 - aus / 1 - ein)
	2 <sup>2</sup>	-	AINP3	
	2 <sup>3</sup>	-	AINP4	
	2 <sup>4</sup>	-	AINP5	
	2 <sup>5</sup>	-	AINP6	
	2 <sup>6</sup>	-	DINPUT	

Power-Fail-Check aktivieren:

0	-	nicht aktiv (Nur lesen)
1	-	aktiv / aktivieren

Schreiberlaubnis für Feldschnittstelle:

0	-	Lese- und Schreibberechtigung
1	-	Nur Leseberechtigung

Adresse	Datum	Zugriff	Wertebereich
0x0010	Adresse Feldschnittstelle	R/W	1 ... 247
0x0011	Adressänderung-gesperrt-Flag	R/W	0 ... 1

Adresse Feldschnittstelle:

Adressänderung-gesperrt-Flag:

Ein einmaliger Schreibzugriff auf diese Modbus-Adresse sperrt Schreibzugriffe auf Modbus-Adresse 0x0010. Ein neuer Schreibzugriff ist erst wieder möglich, wenn auf die Modbus-Adresse 0x0042 ein Schreibzugriff erfolgte, die Adresse von der Gerätefront verstellt wurde oder die Sperrung durch Löschen dieses Flags aufgehoben wurde.

Adresse	Datum	Zugriff	Wertebereich
0x0014	Passwortmodus	R/W	0 ... 3
0x0015	Passwort-Versuche	R/W	0 ... 99
0x0016	Passwortstatus	R	0 ... 2

Passwort-Modus:

Der Passwort-Modus bestimmt die Zugriffsart auf die Daten des KS98 über die Schnittstellen.

Passwort-Versuche:

Bestimmt die Anzahl der erlaubten Fehlversuche bei der Übertragung des Passwortes. Wenn diese Anzahl überschritten wird, wird das vorhandene Engineering gelöscht.

Passwortstatus:

0	-	kein Passwort vorhanden
1	-	Passwort vorhanden, aber nicht aktiv (eingeloggter Zustand)
2	-	Passwort vorhanden und aktiv (ausgeloggter Zustand)

Adresse	Datum	Zugriff	Wertebereich
0x0020	Basic HW-Optionen: Modul A, P	R	2101 ... 2999
0x0021	Ext. HW-Optionen: Modul B, C	R	0000 ... 9999
0x0022	SW-Optionen	R	0000 ... 9999
0x0023	SW-Codenummer (7.-10.Stelle)	R	7254
0x0024	SW-Version (11.+12. Stelle)	R	0000 ... 0099
0x0025	Bedienversion	R	0000 ... 0099
0x0026	Versionsstand des E <sup>2</sup> PROM's	R	0000 ... 0099
0x0027	HW-Codenummer (6.-9. Stelle)	R	6300 ... 8939
0x0028	Modul 1, Modulare Optionskarte C	R	0, 46-49, 76-78
0x0029	Modul 2, Modulare Optionskarte C	R	0, 46-49, 76-78
0x002A	Modul 3, Modulare Optionskarte C	R	0, 46-49, 76-78
0x002B	Modul 4, Modulare Optionskarte C	R	0, 46-49, 76-78
0x002C	Modulare Optionskarte C	R	0 ... 1
0x002E	Engineeringlänge	R	0 ... 28399

Basic HW-Optionen: Wert = 21xy  
 mit : 21 = Gerätetyp KS98  
 und xy = 01: Relais OUT1, 2, 4, 5  
 bzw. = 21: Strom OUT1, 2  
 Relais OUT 4, 5  
 bzw. = 99: Erweiterung

Ext. HW-Optionen: Wert = abcd  
 mit ab = 00: keine Optionskarte B  
 01: Optionskarte B mit TTL-Schnittstelle, di/do  
 02: Optionskarte B mit RS485/422-Schnittstelle, di/do, Uhr  
 10: Optionskarte B mit Profibus DP-Schnittstelle, di/do  
 11: Optionskarte B mit Interbus S-Schnittstelle  
 99: Sonstige Optionskarten B  
 cd = 00: keine Optionskarte C  
 07: Optionskarte C mit INP3/4, OUT3, di/do  
 08: Optionskarte C, modular  
 99: Sonstige Optionskarten C

SW-Optionen:  
 Dieser Wert wird im KS98 z.Zt. nicht benutzt.

SW-Codenummer:

SW-Version:

Angabe der 7.-10. bzw. 11.+12. Stelle der SW-Codenummer  
 4012 157 254VR

Bedienversion:

Angabe der Bedienversion 1... Die Bedienversion wird unabhängig von der SW-Codenummer gezählt.

Versionsstand des E<sup>2</sup>PROM's:

Dieser Wert wird im KS98 z. Zt. nicht genutzt.

Engineeringlänge:

Genutzte Länge des RAM-Engineeringspeichers in Bytes (Lesbar im Online-Betrieb).

Modulare Optionskarte C:

0 - nicht bestückt  
 1 - bestückt

Modul x, Modulare Optionskarte C:

0 - nicht bestückt  
 46 - bestückt mit Thermoelementeingang-Modul  
 47 - bestückt mit Stromausgang-Modul  
 48 - bestückt mit Spannungsausgang-Modul  
 49 - bestückt mit Digital-I/O-Modul  
 76 - bestückt mit Frequenzeingang-Modul  
 77 - bestückt mit Widerstandseingang-Modul  
 78 - bestückt mit Spannungseingang-Modul

Adresse	Datum	Zugriff	Wertebereich
0x0030	Uhrzeit Jahr	R/W	0 ... 99 / 1970...2069
0x0031	Uhrzeit Monat	R/W	1 ... 12
0x0032	Uhrzeit Tag	R/W	1 ... 31
0x0033	Uhrzeit Stunden	R/W	0 ... 23
0x0034	Uhrzeit Minuten	R/W	0 ... 59
0x0035	Parameteranzeige durch Bedienung	R/W	0 (möglich), 1 (blockiert)
0x0036	Konfigurationsanz. durch Bedienung	R/W	0 (möglich), 1 (blockiert)
0x0037	Hauptmenüanzeige durch Bedienung	R/W	0 (möglich), 1 (blockiert)

### Geräteparameter:

Die Modbus-Adressen 0x0035 ...0x0037 haben nur eine Wirkung, falls die Option Echtzeituhr vorhanden ist. Der Wertebereich ist alternativ: 00...69, 70...99 = 2000...2069, 1970...1999 oder 1970...2069.

Die Geräteparameter mit den Modbus-Adressen ... beeinflussen die Möglichkeit der Bedienung vorhandene Einstellungen zu ändern. Ihr Zustand wird mit den entsprechenden digitalen Eingängen der Libraryfunktion STATUS , Typnummer 125, verodert, sofern diese benutzt wird. Die Parameter werden im E<sup>2</sup>PROM gespeichert und sind damit auch nach Power-On vorhanden. Die Ergebnisse der ODER-Verknüpfung haben unterschiedliche Priorität.

- Bei der Blockierung der Hauptmenüanzeige können auch die Menüs für Parameter- und Konfigurationsanzeige nicht mehr erreicht werden.
- Wenn die Anzeige von Parametern blockiert ist, ist ebenfalls die Anzeige von Konfigurationen blockiert.
- Die Blockierung der Konfigurationsanzeige schließt keine weiteren Blockierungen ein.

Blockierung der Konfigurationsanzeige heißt, daß das Gerät den Online-Zustand nicht durch Bedieneingaben, sondern nur durch Schnittstellennachrichten verlassen kann und Anzeige der Konfiguration durch die Bedienung nicht möglich sind.

Blockierung der Parameteranzeige heißt, daß die Parameter nicht angezeigt werden können. Dies hat keine Auswirkung auf die Änderung von Level-1-Daten in den Bedienseiten.

Adresse	Datum	Zugriff	Wertebereich
0x0040	Protokollart	R/W	0 ... 3
0x0041	Baudrate	R/W	(0), 1 ... 4
0x0042	Geräteadresse	R/W	1 ... 247
0x0043	Netzfrequenz	R/W	0, 1
0x0044	Sprache	R/W	0, 2
0x0045	CAN-Node-Id	R/W	1 ... 24
0x0046	CAN-Baudrate	R/W	0 ... 8
0x0047	Ausgangszustände beim Download	R/W	0, 1
0x0048	Einschaltverzögerung CAN	R/W	0 ... 10

Protokollart:

0	-	ISO 1745
1	-	Profibus DP
2	-	Interbus S
3	-	Modbus

Baudrate:

0	-	nicht einstellbar (Keine Feldschnittstelle, Profibus DP, Interbus S)
1	-	2400 Baud
2	-	4800 Baud
3	-	9600 Baud
4	-	19200 Baud

### Geräteadresse:

Hiermit wird die Adresse der Feldschnittstelle festgelegt. Der Einstellbereich geht von 1 ... 247.

Netzfrequenz:

0	-	50 Hz
1	-	60 Hz

Sprache:

0	-	deutsch
1	-	englisch
2	-	französisch

### CAN-Node-Id:

Knotennummer des KS98. Id=1 bedeutet gleichzeitig, dass der KS98 Netzwerkmaster (NMT) ist.

<u>CAN-Baudrate:</u>	0	-	10 KBaud
	1	-	20 KBaud
	2	-	50 KBaud
	3	-	100 KBaud
	4	-	125 KBaud
	5	-	250 KBaud
	6	-	500 KBaud
	7	-	800 KBaud
	8	-	1000 KBaud

Ausgangszustände beim Download:

0	-	alle Ausgänge abschalten nach Umschaltung ONLINE -> OFFLINE RAM-Zustand als zerstört kennzeichnen
1	-	Ausgangszustände auf dem letzten Wert / Zustand einfrieren RAM-Daten bei Umschaltung ONLINE -> OFFLINE erhalten.

### 4.1.2 Beispiele

- Lesen der Gerätestatusinformationen 'Status 1' (0x0001), 'Status 2' (0x0002) und 'Gerätezustand' (0x0005):

=> Lesen von 5 Werten vom Gerät mit der Adresse 1 beginnend ab Modbusadresse 0x0001

Aufbau der Anforderungsnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>CrcH</i>	<i>CrcL</i>
01	03	00	01	00	05	xx	xx

Aufbau der Antwortnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>Anz</i>	<i>Status1</i>	<i>Status 2</i>	<i>32000</i>	<i>-32000</i>	<i>Gerätestatus</i>	<i>CrcH</i>	<i>CrcL</i>					
01	03	0A	00	22	00	03	81	0C	81	0C	00	01	yy	yy

- Umschalten in den Offline-Zustand 'Gerätezustand' (0x0005) = 1:

=> Schreiben von einem Wert an Gerät mit der Adresse 1 an Modbusadresse 0x0005

Aufbau der Vorgabennachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>Anz</i>	<i>Offline</i>	<i>CrcH</i>	<i>CrcL</i>	
01	10	00	05	00	01	02	00	01	xx	xx

Aufbau der Antwortnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>CrcH</i>	<i>CrcL</i>
01	10	00	05	00	01	yy	yy

## 4.2

**L1WRIT / L1READ**

Die Adressen der Daten der Funktionstypen L1WRIT und L1READ sind abhängig von der Blocknummer, die der Funktionsblock belegt.

Berechnung der Startadresse =  $0x0050 + (\text{Blocknummer} - 1) * 0x0009$

4.2.1 **Struktur für L1WRIT**

Offset	Datum	Zugriff	Wertebereich
0x0	Digitale Ausgänge z1...z15	R/W	0 ... 32767
0x1	Analoger Ausgang Y1	R/W	-29999 ... 200000
0x2	Analoger Ausgang Y2	R/W	-29999 ... 200000
0x3	Analoger Ausgang Y3	R/W	-29999 ... 200000
0x4	Analoger Ausgang Y4	R/W	-29999 ... 200000
0x5	Analoger Ausgang Y5	R/W	-29999 ... 200000
0x6	Analoger Ausgang Y6	R/W	-29999 ... 200000
0x7	Analoger Ausgang Y7	R/W	-29999 ... 200000
0x8	Analoger Ausgang Y8	R/W	-29999 ... 200000

Digitaler Ausgang z1 an Bit 0 (LSB).

4.2.2 **Struktur für L1READ**

Offset	Datum	Zugriff	Wertebereich
0x0	Status 1	R	0 ... 63
0x1	Status 2	R	0 ... 63
0x2	Analoger Eingang X1	R	-29999 ... 200000
0x3	Analoger Eingang X2	R	-29999 ... 200000
0x4	Analoger Eingang X3	R	-29999 ... 200000
0x5	Analoger Eingang X4	R	-29999 ... 200000
0x6	Analoger Eingang X5	R	-29999 ... 200000
0x7	Analoger Eingang X6	R	-29999 ... 200000
0x8	Analoger Eingang X7	R	-29999 ... 200000

Status 1:

Bit 15...6 :	0		
Bit 5 :	Zustand d6	[0] - aus	[1] - ein
Bit 4 :	Zustand d5	[0] - aus	[1] - ein
Bit 3 :	Zustand d4	[0] - aus	[1] - ein
Bit 2 :	Zustand d3	[0] - aus	[1] - ein
Bit 1 :	Zustand d2	[0] - aus	[1] - ein
Bit 0 (LSB) :	Zustand d1	[0] - aus	[1] - ein

Status 2:

Bit 15...6 :	0		
Bit 5 :	Zustand d12	[0] - aus	[1] - ein
Bit 4 :	Zustand d11	[0] - aus	[1] - ein
Bit 3 :	Zustand d10	[0] - aus	[1] - ein
Bit 2 :	Zustand d9	[0] - aus	[1] - ein
Bit 1 :	Zustand d8	[0] - aus	[1] - ein
Bit 0 (LSB) :	Zustand d7	[0] - aus	[1] - ein

### 4.2.3 Beispiel

Schreiben des zweiten bis vierten Wertes (3, 4, 5) im Floatformat an die L1WRIT-Funktion an Blocknummer 8:

=> Schreiben von drei Werten an Gerät mit der Adresse 1 an Modbusadresse  $0x8000 + 2*(0x50 + 7*0x9)$

Aufbau der Vorgabennachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>Anz</i>	<i>Wert = 3.0</i>	<i>Wert = 4.0</i>	<i>Wert</i>
01	10	81	1E	00	06	0C	40	40	00
							00	00	40
							80	00	00
							40	80	00
							00	00	40

<i>= 5.0</i>	<i>CrcH</i>	<i>CrcL</i>
A0	00	00
xx	xx	xx

Aufbau der Antwortnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>CrcH</i>	<i>CrcL</i>
01	10	81	1E	00	06	yy	yy

## 4.3 MBDATA-Strukturen

### 4.3.1 Definitionen

Der Adressbereich 0x0110...0x015F wird von 5 Funktionsblöcken MBDATA belegt. Diese Funktionen sind neu und erlauben den Zugriff auf je 16 frei konfigurierbare Parameter des Engineerings analog zu VPARA. Die Parameter können allerdings nur über die Schnittstelle verändert werden, nicht über Eingänge oder eine Bedienung der Funktionen. Sie können im Engineering an die Blocknummern 56...60 gelegt werden. Blocknummer 56 beginnt an Modbusadressen 0x0110. Die anderen Blocknummern jeweils 0x0010 Adressen später.

Spezifikation:

Analoge Eingänge:	keine
Digitale Eingänge:	keine
Analoge Ausgänge:	16, Werte der konfigurierten Parameter bzw. 0
Digitale Ausgänge:	keine
Float-Parameter:	keine
Int-Parameter:	keine
Float-Konfigurationen:	keine
Int-Konfigurationen:	32, 16mal je Datum die Angabe der Blocknummer und der Parameternummer (es werden erst die Integer-, dann die Floatparameter gezählt).
Texte:	1, Titel
Blocknummerbereich:	56...60
Zeitgruppenzuordnung:	alle

### 4.3.2 Beispiel

Lesen der Werte des 4. – 6. Parameters im Floatformat, die über die MBDATA-Funktion an Blocknummer 59 parametrisiert werden können.

=> Lesen von 3 Werten im Floatformat (6 Worte) vom Gerät mit der Adresse 1 beginnend ab Modbusadresse  $0x8000 + 2*(0x110 + 3*0x10 + 3) = 0x8286$

Aufbau der Anforderungsnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>CrcH</i>	<i>CrcL</i>
01	03	82	86	00	06	xx	xx

Aufbau der Antwortnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>Anz</i>	<i>Wert = 3.0</i>	<i>Wert = 4.0</i>	<i>Wert = 5.0</i>	<i>CrcH</i>	<i>CrcL</i>									
01	03	0C	40	40	00	00	40	80	00	00	40	A0	00	00	yy	yy

## 4.4

### Modbusadressen für Regler

Es werden Modbusadressen für die Level-1-Daten von 30 Reglerfunktionsblöcken (CONTR / CONTR+ / PIDMA) vorgehalten. Die Regler-Basis-Modbusadressen 1...30 definieren die Startadressen von 30 Bereichen mit Datenstrukturen für Regler. Der zugehörige Modbusadressbereich umfasst 0x0160...0x0687. Die Modbusadressen der einzelnen Level-1-Daten errechnen sich über den in der unten beschriebenen Definition dieser Datenbereiche angegebenen Offset. Die Zuordnung der Bereiche zu den Funktionsblöcken ergibt sich aus der Reihenfolge der vorhandenen Reglerfunktionsblöcke.

Berechnung der Basisadresse =  $0x0160 + (\text{Bereichsnummer} - 1) * 0x2C$

### 4.4.1

#### Struktur für CONTR, CONTR+, PIDMA

Offset	Datum	Zugriff	Wertebereich
0x00	Status 1	R	0 ... 63
0x01	Status 2	R	0 ... 63
0x02	Status 3 (nicht PIDMA)	R	0 ... 63
0x03	Sollwertstatus	R	0 ... 63
0x04	Status Tuning 1	R	0 ... 63

Status 1:

Bit 15...6	:	0		
Bit 5	:	Sensorfail	[0] - nein	[1] - ja
Bit 4	:	Regler abgeschaltet	[0] - nein	[1] - ja
Bit 3	:	Y/Y2-Umschaltung	[0] - Y	[1] - Y2
Bit 2	:	Automatik/Hand	[0] - Auto	[1] - Hand
Bit 1	:	Schaltausgang 2	[0] - aus	[1] - ein
Bit 0 (LSB)	:	Schaltausgang 1	[0] - aus	[1] - ein

Status 2:

Bit 15...6	:	0		
Bit 5	:	Zustand Regler	[0] - ok	[1] - nicht ok
Bit 4	:	Zustand PI/P	[0] - PI	[1] - P (nicht PIDMA)
Bit 3	:	0		
Bit 2	:	0		
Bit 1	:	0		
Bit 0 (LSB)	:	0		

Status 3: (nicht PIDMA)

Bit 15...3	:	0		
Bit 2	:	OVC+	[0] - aus	[1] - ein
Bit 1	:	OVC-	[0] - aus	[1] - ein
Bit 0 (LSB)	:	0		

Sollwertstatus:

Bit 15...5 :	0		
Bit 4 :	Tracking	[0] - aus	[1] - ein
Bit 3 :	Sollwertgradient unterdrückt	[0] - nein	[1] - ja
Bit 2 :	Weff eingefroren	[0] - nein	[1] - ja
Bit 1 :	Wext/Wint	[0] - ext	[1] - int
Bit 0 (LSB) :	w/W2-Umschaltung	[0] - w	[1] - W2

Status Tuning:

Bit 15...3 :	0		
Bit 2 :	Ergebnis RSE	[0] - Ok	[1] - Fehler
Bit 1 :	Betrieb RSE	[0] - aus	[1] - ein
Bit 0 (LSB) :	Prozeß in Ruhe	[0] - nein	[1] - ja (nicht PIDMA)

Offset	Datum	Zugriff	Wertebereich
0x07	Zusatzstellwert aus / ein	R/W	0 / 1
0x08	Strukturumschaltung (nicht PIDMA)	R/W	0 / 1
0x09	Automatik/Hand-Umschaltung	R/W	0 / 1
0x0A	Starten der RSE	R/W	0 / 1
0x0B	Umschaltung Wext/Wint	R/W	0 / 1
0x0C	Umschaltung w / W2	R/W	0 / 1
0x0D	Regler ein/aus	R/W	0 / 1

Offset	Datum	Zugriff	Wertebereich
0x10	eff. Sollwert	R	-29999 ... 200000
0x11	eff. Istwert	R	-29999 ... 200000
0x12	wirksame Stellgröße	R	-29999 ... 200000
0x13	Regelabweichung	R	-29999 ... 200000
0x14	Hauptregelgröße 1	R	-29999 ... 200000
0x15	Hilfsregelgröße 2	R	-29999 ... 200000
0x16	Hilfsregelgröße 3	R	-29999 ... 200000
0x17	Stellgrößenrückmeldung	R	-29999 ... 200000
0x18	Overridecontrol + (nicht PIDMA)	R	-29999 ... 200000
0x19	Overridecontrol - (nicht PIDMA)	R	-29999 ... 200000
0x1A	ext. Sollwert	R	-29999 ... 200000
0x1B	int. Sollwert, nicht flüchtig	R/W	-29999 ... 200000
0x1C	int. Sollwert, flüchtig	R/W	-29999 ... 200000
0x1D	differenzielle Stellgrößenvorgabe	R/W	-210 ... 210
0x1E	absolute Stellgrößenvorgabe	R/W	-105 ... 105
0x1F	wirksame Parametersatznr. (nicht PIDMA)	R/W	1 ... 6
0x20	Parametersatz, der optimiert werden soll	R	1 ... 6
0x21	Verzugszeit Heizen (nicht PIDMA)	R	0 ... 200000
0x22	Anstiegsgeschwindigkeit Heizen (nicht PIDMA)	R	0 ... 9.999
0x23	Prozeßverstärkung Heizen (nicht PIDMA)	R	0 ... 9.999
0x24	Fehlercode der RSE Heizen (nicht PIDMA)	R	0 ... 8
0x25	Verzugszeit Kühlen (nicht PIDMA)	R	0 ... 200000
0x26	Anstiegsgeschwindigkeit Kühlen (nicht PIDMA)	R	0 ... 9.999
0x27	Prozeßverstärkung Kühlen (nicht PIDMA)	R	0 ... 9.999
0x28	Fehlercode der RSE Kühlen (nicht PIDMA)	R	0 ... 8

### 4.4.2 Beispiel

Schreiben des int. Sollwertes = 100 an den 2. Reglerfunktionsblock im Floatformat

=> Schreiben von 1 Werten im Floatformat (2 Worte) an Gerät mit der Adresse 1 beginnend ab Modbusadresse  $0x8000 + 2 \cdot (0x160 + 0x2C + 0x1C) = 0x8350$

Aufbau der Vorgabennachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>Anz</i>	<i>Wert = 100.0</i>	<i>CrcH</i>	<i>CrcL</i>
01	10	83	50	00	02	04	42 C8 00 00	xx	xx

Aufbau der Antwortnachricht (Hex-Darstellung):

<i>Adr</i>	<i>Mode</i>	<i>ModH</i>	<i>ModL</i>	<i>AnzH</i>	<i>AnzL</i>	<i>CrcH</i>	<i>CrcL</i>
01	10	81	1E	00	02	yy	yy

## 4.5 Modbusadressen für Programmgeber

Es werden Modbusadressen für die Level-1-Daten von 40 Programmgeberfunktionsblöcken (APROG / DPROG) vorgehalten. Die Programmgeber-Basis-Modbusadressen 1...40 definieren die Startadressen von 40 Bereichen mit Datenstrukturen für Programmgeber. Der zugehörige Modbusadressbereich umfasst 0x0688...0x09F7 Die Modbusadressen der einzelnen Level-1-Daten errechnen sich über den in der unten beschriebenen Definition dieser Datenbereiche angegebenen Offset. Die Zuordnung der Bereiche zu den Funktionsblöcken ergibt sich aus der Reihenfolge der vorhandenen Programmgeberfunktionsblöcke.

Berechnung der Basisadresse =  $0x0688 + (\text{Bereichsnummer} - 1) \cdot 0x16$

### 4.5.1 Struktur für APROG und DPROG

Offset	Datum	Zugriff	Wertebereich
0x00	Status 1	R	0 ... 63
0x01	Status 2	R	0 ... 63
0x02	Status 3 (nur DPROG)	R	0 ... 63
0x03	eff. Rezeptnummer	R	1 ... 99
0x04	Programmzeit netto	R	0 ... 959999
0x05	Programmzeit brutto	R	0 ... 959999
0x06	Sollwert (nur APROG)	R	-29999 ... 999999
0x07	Programmzeit Rest	R	0 ... 959999
0x08	Endwert aktuelles Segment (nur APROG)	R	-29999 ... 999999
0x09	Segmentnummer	R	1 ... 999
0x0A	Segmentrestzeit	R	0 ... 959999

Status 1:

Bit 15...5 :	0		
Bit 4 :	Err2	[0] - nein	[1] - ja
Bit 3 :	Err1	[0] - nein	[1] - ja
Bit 2 :	Prog-Reset	[0] - aus	[1] - ein
Bit 1 :	Prog-Ende	[0] - nein	[1] - ja
Bit 0 (LSB) :	Prog-Lauf	[0] - Stop	[1] - läuft

Err1: fehlerhafter Parameterblock

Err2: Unendlichschleife bei Parameterblöcken

Status 2: Akt. Zustände der Spuren (nur DPROG)  
 Bit 15...6 : 0  
 Bit 5 : Spur 6 [0] - aus [1] - ein  
 Bit 4 : Spur 5 [0] - aus [1] - ein  
 Bit 3 : Spur 4 [0] - aus [1] - ein  
 Bit 2 : Spur 3 [0] - aus [1] - ein  
 Bit 1 : Spur 2 [0] - aus [1] - ein  
 Bit 0 (LSB) : Spur 1 [0] - aus [1] - ein

Status 2: Akt. Zustand (nur APROG)  
Status 3: Akt. Zustand (nur DPROG)  
 Bit 15...2 : 0  
 Bit 1 : Prog-Manual [0] - Auto [1] - Manual  
 Bit 0 (LSB) : Prog-Halt [0] - kein Halt [1] - Halt

Offset	Datum	Zugriff	Wertebereich
0x0C	Programm Stop / Run	R/W	0 / 1
0x0D	Programm Continue / Reset	R/W	0 / 1
0x0E	Programmsuchlauf starten (nur APROG)	R/W	0 / 1
0x0F	F-Key-Funktion (A/H-Umschaltung)	R/W	0 / 1
0x10	Programm Auto / Manual	R/W	0 / 1

Offset	Datum	Zugriff	Wertebereich
0x12	Rezeptnummer	R/W	1 ... 99
0x13	Presetwert (Segment / Zeit)	R/W	1...999 / 0...59999
0x14	Sollwertvorgabe (im Manual-Betrieb)	R/W	-29999 ... 999999 / 000000 ... 111111

### 4.5.2 Beispiel

Lesen von Programmzeit netto (0x04), Programmzeit brutto (0x05), Sollwert (0x06) und Programmzeit Rest (0x07) vom 15. Programmgeberfunktionsblock:

=> Lesen von 4 Werten im Floatformat (8 Worte) vom Gerät mit der Adresse 1 beginnend ab Modbusadresse  $0x8000 + 2*(0x0688 + 14*0x16 + 0x04) = 0x87C0$

Aufbau der Anforderungsnachricht (Hex-Darstellung):

*Adr Mode ModH ModL AnzH AnzL CrCH CrCL*

01	03	87	C0	00	08	xx	xx
----	----	----	----	----	----	----	----

Aufbau der Antwortnachricht (Hex-Darstellung):

*Adr Mode Anz Wert = 20.0 Wert = 20.0 Wert = 100.0 Wert*

01	03	10	41	A0	00	00	41	A0	00	00	42	C8	00	00	42	20
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

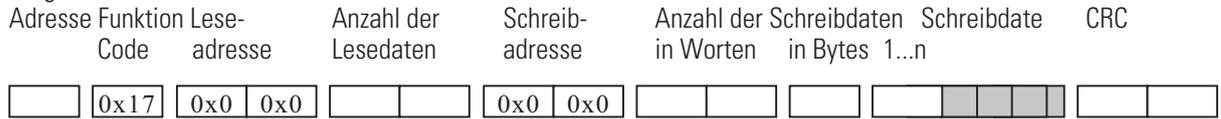
*= 40.0 CrCH CrCL*

00	00	yy	yy
----	----	----	----

## 5 'B'-Schlüssel Nachrichten

Für die Übertragung der im ISO1745-Mode per Funktionsblockprotokoll übertragenen Codes B1...B4 wird der Modbus-Funktionscode 23 (0x17) benutzt, der eine kombinierte Schreib/Lesenachricht bietet. Dieser Nachrichtentyp wird im Allgemeinen nur vom Engineeringtool benutzt.  
Der Aufbau einer solchen Nachricht ist:

Vorgabe:



Antwort:



Da dieser Nachrichtentyp nur verwendet wird, um die bestehenden ISO1745-Funktionsblockprotokolle in den Schreib- bzw. Lesedaten zu übertragen, sind die 'Leseadresse' und die 'Schreibadresse' ohne Bedeutung und werden nicht ausgewertet. Die Bereiche 'Schreibdaten' und 'Lesedaten' enthalten im Gegensatz zu den anderen Bereichen immer ASCII-Werte.

Beim Schreiben von Daten ist die 'Anzahl der Lesedaten' = 1. Die 'Schreibdaten' enthalten das Funktionsblockprotokoll entsprechend den folgenden Definitionen und damit die Information, welche Daten wohin übertragen werden. Die 'Anzahl der Schreibdaten' zeigt an, wie viele Worte bzw. Bytes an Daten im Bereich 'Schreibdaten' stehen. Die Antwort enthält als 'Anzahl der Bytes' den Wert 2 und 2 'Lesedaten' mit dem Wert 0.

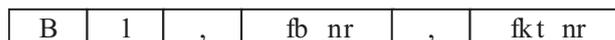
Beim Lesen von Daten ist die 'Anzahl der Lesedaten' ≠ 0. Der genaue Wert ist egal, da das in den 'Schreibdaten' enthaltene Funktionsblockprotokoll die Information enthält, welche Daten von wo gelesen werden sollen. Die 'Anzahl der Schreibdaten' zeigt an, wie viele Worte bzw. Bytes an Daten im Bereich 'Schreibdaten' stehen. Die Antwort enthält als 'Anzahl der Bytes' die Länge der 'Lesedaten'. In den 'Lesedaten' steht die Antwortstruktur entsprechend den folgenden Definitionen.

### 5.1 Funktionsblockprotokoll für I/O- und VTREND-Daten: Code B1

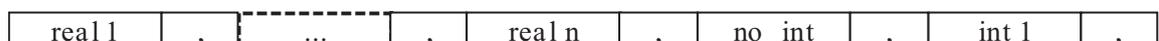
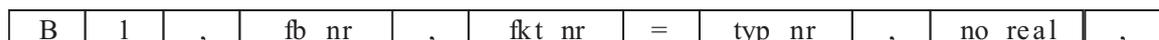
I/O-Daten einer Rechenfunktion und Daten von VTREND können über den B1-Zugriff ausgelesen werden.

Lesen von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:



Aufbau der 'Lesedaten' für die Antwort:



max. Anz. Daten:	Reals:	25	<->	0	Integers
	Reals	0	<->	38	Integers
	bei VTREND immer 25 Realwerte				

fb_nr:	1 ... 450	Funktionsblocknummer
fkt_nr:	0 1 80-84	Inputdaten Outputdaten Trenddatenbereich 1 bis 5
typ_nr:	1...127	Nummer des Funktionstyps
no_real:	0 1 ... 25	keine Real-Werte Anzahl der übertragenen Real-Werte
no_int:	0 1 ... 38	keine Integer-Werte Anzahl der übertragenen Integer-Werte

## 5.2 Funktionsblockprotokoll für Parameter: Code B2

Dieser Zugriff erlaubt das gruppenweise Lesen und Schreiben von Parametern. Parameter sind Level-2-Daten, die permanent gespeichert werden (online).

Lesen von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:

B	2	,	fb_nr	,	fkt_nr
---	---	---	-------	---	--------

Aufbau der 'Lesedaten' für die Antwort:

B	2	,	fb_nr	,	fkt_nr	=	typ_nr	,	no_real	,
---	---	---	-------	---	--------	---	--------	---	---------	---

real 1	,	...	,	real n	,	no_int	,	int 1	,
--------	---	-----	---	--------	---	--------	---	-------	---

...	,	int n
-----	---	-------

Schreiben von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:

B	2	,	fb_nr	,	fkt_nr	=	typ_nr	,	no_real	,
---	---	---	-------	---	--------	---	--------	---	---------	---

real 1	,	...	,	real n	,	no_int	,	int 1	,
--------	---	-----	---	--------	---	--------	---	-------	---

...	,	int n
-----	---	-------

Aufbau der 'Lesedaten' für die Antwort:

Keine Lesedaten!

max. Anz. Daten:	Reals:	25	0	Integers
	Reals:	0	38	Integers

fb_nr:	0 1 ... 450	Gerät Funktionsblocknummer
fkt_nr:	0 1-9 1-10	Allgemein übrige Funktionen, nach Erfordernissen vorhanden
typ_nr:	1...127	Nummer des Funktionstyps
no_real:	0 1 ... 25	keine Real-Werte Anzahl der übertragenen Real-Werte
no_int:	0 1 ... 38	keine Integer-Werte Anzahl der übertragenen Integer-Werte

### 5.3 Funktionsblockprotokoll für Anzeigetexte: Code B2

Dieser Zugriff erlaubt das gruppenweise Lesen und Schreiben von Anzeigentexten. Parameter sind Level-2-Daten, die permanent gespeichert werden (online). Ein Text besteht immer aus 16 Zeichen. Der Datentyp CHAR[n] stellt eine Sonderfunktion dar und wird den Integerwerten zugerechnet.

Lesen von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:

B	2	,	fb_nr	,	8	0
---	---	---	-------	---	---	---

Aufbau der 'Lesedaten' für die Antwort:

B	2	,	fb_nr	,	8	0	=	typ_nr	,	0	,
---	---	---	-------	---	---	---	---	--------	---	---	---

Schreiben von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:

B	2	,	fb_nr	,	8	0	=	typ_nr	,	0	,
---	---	---	-------	---	---	---	---	--------	---	---	---

no_text	,	text 1	,	...	,	text n
---------	---	--------	---	-----	---	--------

...	,	int n
-----	---	-------

Aufbau der 'Lesedaten' für die Antwort:

Keine Lesedaten!

max. Anz. Texte:	13 Texte	
fb_nr:	1 ... 450	Funktionsblock
typ_nr:	1...127	Nummer des Funktionstyps
no_text:	0 1 ... 13	keine Texte Anzahl der übertragenen Texte

**5.4**

**Funktionsblockprotokoll für Konfigurationsdaten: Code B3**

Dieser Zugriff erlaubt das gruppenweise Lesen und Schreiben von Konfigurationsdaten. Konfigurationsdaten sind Level-3-Daten, die permanent nur offline gespeichert werden können.

Lesen von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:

B	3	,	fb_nr	,	fkt_nr
---	---	---	-------	---	--------

Aufbau der 'Lesedaten' für die Antwort:

B	3	,	fb_nr	,	fkt_nr	=	typ_nr	,	no_real	,
---	---	---	-------	---	--------	---	--------	---	---------	---

real 1	,	...	,	real n	,	no_int	,	int 1	,
--------	---	-----	---	--------	---	--------	---	-------	---

...	,	int n
-----	---	-------

Schreiben von Daten:

Aufbau der 'Schreibdaten' für die Anforderung:

B	3	,	fb_nr	,	fkt_nr	=	typ_nr	,	no_real	,
---	---	---	-------	---	--------	---	--------	---	---------	---

real 1	,	...	,	real n	,	no_int	,	int 1	,
--------	---	-----	---	--------	---	--------	---	-------	---

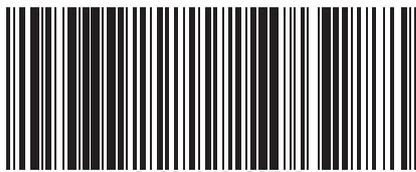
...	,	int n
-----	---	-------

Aufbau der 'Lesedaten' für die Antwort:

Keine Lesedaten!

max. Anz. Daten:	Reals:	25	<->	0	Integers
	Reals	0	<->	38	Integers
fb_nr:	0	Gerät			
	1 ... 450	Funktionsblocknummer			
fkt_nr:	0	Allgemein			
	1-9	übrige Funktionen, nach Erfordernissen vorhanden			
	1-10				
typ_nr:	1...127	Nummer des Funktionstyps			
no_real:	0	keine Real-Werte			
	1 ... 25	Anzahl der übertragenen Real-Werte			
no_int:	0	keine Integer-Werte			
	1 ... 38	Anzahl der übertragenen Integer-Werte			

Um Daten mit dem B3-Schlüssel schreiben zu können, muß vorher das Gerät in den Konfigurationsmodus geschaltet werden. Wirksam werden alle neu eingegebenen Konfigurationsdaten u. Parameter erst, wenn das Gerät wieder auf online umgeschaltet wurde.



9499-040-88718

Subject to alterations without notice  
Änderungen vorbehalten  
Sous réserve de toutes modifications

© PMA Prozeß- und Maschinen-Automation GmbH  
P.O.B. 310 229, D-34058 Kassel, Germany  
Printed in Germany 9499-040-88718 (02/2008)

A4, unibind, SW-Druck, 80g weiß