



**KS 816**  
**Multi-Meßumformer**  
**Multi-Temperaturregler**

**CANopen**

**KS**

**Schnittstellenprotokoll**  
**CANopen**  
**9499 040 56018**  
gültig ab: 8355

©PMA Prozeß- und Maschinen-Automation GmbH 1999. Printed in Germany  
Alle Rechte vorbehalten. Ohne vorherige schriftliche  
Genehmigung ist der Nachdruck oder die auszugsweise  
fotomechanische oder anderweitige Wiedergabe dieses Dokumentes  
nicht gestattet.

Dies ist eine Publikation der PMA Prozeß- und Maschinen-Automation GmbH

Bei Änderungen erfolgt keine gesonderte Mitteilung.

PMA Prozeß- und Maschinen-Automation GmbH  
P.O. Box 31 02 29  
D 34058 Kassel  
Germany

Einschränkung der Gewährleistung:

Es wird keine Gewähr für die vollständige Richtigkeit des Inhalts übernommen, da sich trotz aller  
Sorgfalt Fehler nie ganz vermeiden lassen. Für Hinweise sind wir jederzeit dankbar.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>5</b>
<b>2</b>	<b>Service-Daten-Objekte</b> .....	<b>7</b>
<b>3</b>	<b>Prozeß-Daten-Objekte</b> .....	<b>9</b>
	3.1 Information Record .....	11
	3.2 Control Record .....	12
	3.3 PDO Übertragungsverhalten .....	13
	3.3.1 Asynchrones Sende PDO .....	13
	3.3.2 Synchrones Sende PDO .....	14
	3.3.3 Asynchrones Empfangs PDO .....	14
	3.3.4 Synchrones Empfangs PDO .....	15
<b>4</b>	<b>OP-Mode / Pre-OP-Mode</b> .....	<b>16</b>
<b>5</b>	<b>SYNC</b> .....	<b>17</b>
<b>6</b>	<b>Nodeguarding / Lifeguarding</b> .....	<b>18</b>
<b>7</b>	<b>LMT-Dienste</b> .....	<b>19</b>
	7.1 Switch Mode Services .....	19
	7.1.1 Switch Mode Global .....	19
	7.1.2 Switch Mode Selective .....	19
	7.2 Configuration Services .....	20
	7.2.1 Configure Module ID .....	20
	7.2.2 Configure Bit Timing .....	20
	7.3 Inquire LMT Address .....	21
	7.3.1 Inquire Manufacturer Name .....	21
	7.3.2 Inquire Product Name .....	21
	7.3.3 Inquire Serial Number .....	22
	7.3.4 Serial Number Format .....	22
	7.4 Identification Services .....	22
	7.4.1 LMT Identify Remote Slaves .....	22
	7.4.2 LMT Identify Slave .....	23
	7.5 Herstellerspezifische LMT-Dienste .....	23
	7.6 Inquire Node-ID .....	23
	7.6.1 Configure Module-ID .....	24
	7.6.2 Configure Bit Timing .....	24
	7.6.3 Inquire Bit Timing .....	25
<b>8</b>	<b>Objektverzeichnis</b> .....	<b>26</b>
	8.1 Übersicht Objektverzeichnis herstellerepezifischer Teil .....	26
<b>9</b>	<b>Applikationsbeispiele</b> .....	<b>37</b>
	9.1 Definition der Beispiele .....	37
	9.2 Beispiel 1 (Heizen/Kühlen-Regler) .....	37
	9.3 Beispiel 2 (H/K-Regler einstellen und Selbstoptimierung aktivieren) .....	38
<b>10</b>	<b>CAN Physical Layer</b> .....	<b>40</b>
	10.1 ISO 11898-2 Knoten: .....	40
	10.2 Baudraten und Buslängen .....	41
	10.3 Praktische Buslängen .....	41
	10.4 Kabel-Parameter .....	42



## 1 Einleitung

### Allgemeines:

Der KS 816 besitzt eine CAN-Schnittstelle, über die dem Regler alle zum Betrieb notwendigen Daten (Parameter- und Konfigurationsdaten) gesendet werden können. Ebenso kann ein entsprechender Master (PC oder Steuerung) alle Prozeßdaten abrufen.

Gemäß "CANopen" sind die Zugriffe in SDO (Service Data Objects) und PDO (Process Data Objects) gegliedert. Dabei sind SDOs zur Konfiguration und Parametrisierung von Busteilnehmern vorgesehen und PDOs für die üblichen Betriebswerte.

### Betrieb des KS 816 am CAN Bus:

Nach Aufstart des KS 816 befindet sich das Gerät im "Pre Operational Mode". Man kann über die CAN-Schnittstelle mittels SDOs kommunizieren. Nach der Bus- und Gerätekonfiguration muß der Master den KS 816 in den "Operational Mode" setzen (NMT Start) um auch PDOs nutzen zu können.

Wie PDOs genutzt werden, muß vorab konfiguriert werden. Man unterscheidet "asynchrone PDOs" und "synchrone PDOs". Asynchrone PDOs werden bei Statusänderungen wie Toleranzbandverletzungen, Fühlerbruch oder ähnlichem gesendet. Synchrone PDOs werden auf Anforderung durch den Master gesendet. Dazu sendet der Master einen SYNC Frame. In der Default Einstellung des KS 816 sind asynchrone PDOs freigegeben. (Details siehe: Prozeß-Daten-Objekte)

Die im KS 800 implementierten Kommunikationsdienste basieren auf dem "CANopen Communication Profile for Industrial Systems" (CIA Draft Standard 301). Die Funktionalität entspricht dem "Minimum Capability Device" mit "Predefined Master/Slave Connection Set", ergänzt um "LMT-Dienste" jedoch ohne "Time-Stamp-" und "Store-Parameter\_" Objekte.

Typ	Adresse (dez.)	Adresse (hex.)
NMT_Commands	0	0x0
SYNC	128	0x80
LMT-RX	2021	0x7E5
LMT-TX	2020	0x7E4
EMERGENCY	128 + NodeID *)	0x80 + NodeID *)
PDO1-TX	384 + NodeID *)	0x180 + NodeID *)
PDO1-RX	512 + NodeID *)	0x200 + NodeID *)
PDO2-TX	640 + NodeID *)	0x280 + NodeID *)
PDO2_RX	768 + NodeID *)	0x300 + NodeID *)
SDO-TX	1408 + NodeID *)	0x580 + NodeID *)
SDO-RX	1536 + NodeID *)	0x600 + NodeID *)
NODE_GUARD	1769 + NodeID *)	0x6E0 + NodeID *) **)

\*) NodeID-Bereich von 1 ... 127

\*\*) KS 816 entspricht nicht 301-Standard

### **Netzwerkmanagement:**

Zur Überwachung der Kommunikation bietet der KS 816 die Möglichkeit des "Nodeguardings". Damit ist es dem Master möglich zu erkennen, ob ein KS 816 am Bus angeschlossen ist und ob die Kommunikation zum Gerät sichergestellt ist. Die zur Identifikation eines CAN Teilnehmers nötige Node-ID (Knoten-Adresse) sowie die CAN-Baudrate werden mit Hilfe des KS 816 Engineering Tools über die Diagnoseschnittstelle eingestellt.

Eine Einstellung via CAN-Bus ist ebenfalls möglich (siehe Kapitel 7 Objektverzeichnis, Index 200E/200F)

## 2 Service-Daten-Objekte (SDO's)

Ein SDO ist logisch festgelegt auf genau zwei Partner - 1 Master und 1 Slave. Es ist nicht möglich, von zwei unterschiedlichen Mastern gleichzeitig auf den KS 816 zuzugreifen, da für "tx" als auch für "rx" nur ein SDO-Kanal zur Verfügung steht.

Mittels SDOs können maximal 4 Byte Nutzdaten übertragen werden. Die restlichen 4 Byte werden wie folgt genutzt:

- 1 Byte für den Command: Art der Übertragung
- 2 Byte für den Index: Objektkennung  
(z.B. 0x2213 für Wvol - siehe Objektverzeichn.)
- 1 Byte für den Subindex: Kanalnummer

Im folgenden wird jeweils ein Beispiel eines SDO-Datenframes für Lesen und für Schreiben gegeben.

### Beispiel 1. Ein Master sendet einen Sollwert an einen KS 816 mit Knotenadresse 4:

Der Aufbau der Nachricht ist wie folgt:

- ⇒ COB\_ID = SDO an Knoten 4 = 0x600 + 4 = 0x604
- ⇒ CMD = Schreibzugriff = 0x2B = 0x2B
- ⇒ INDEX = Sollwert Wvol = 0x2213 = 0x2213
- ⇒ SUBINDEX = z.B. Regler 1 = 01 = 0x01
- ⇒ WERT = 30,0 C \*) = 300 \*) = 0x12C

\*) Fixedpoint1-Darstellung, d.h. eine 1 feste Nachkommastelle.

8 data bytes										
	COB-ID	LEN	Cmd	Index		Sub.	Data			
Anforderung (Master)	0x604	0x08	0x2B	0x13	0x22	0x01	0x2C	0x01	0x00	0x00
Antwort (KS816)	0x584	0x08	0x60	0x13	0x22	0x01	0x00	0x00	0x00	0x00

**Beispiel 2. Ein Master liest einen Istwert von einem KS 816 mit Knotenadresse 2:**

Der Aufbau der Nachricht ist wie folgt:

- ⇒ COB\_ID = SDO an Knoten 2 =  $0x600 + 2$  = 0x602
- ⇒ CMD = Lesezugriff = 0x40 = 0x40
- ⇒ INDEX = Istwert Xeff = 0x2202 = 0x2202
- ⇒ SUBINDEX = z.B. Regler 3 = 03 = 0x03

8 data bytes

	COB-ID	LEN	Cmd	Index		Sub.	Data			
Anforderung (Master)	0x602	0x08	0x40	0x02	0x22	0x03	0x00	0x00	0x00	0x00
Antwort (KS816)	0x582	0x08	0x4B	0x02	0x22	0x03	0xFA	0x00	0x00	0x00

⇒ WERT = 0x00FA = 250 = 25,0 C

Mit Hilfe der SDOs können Konfiguration und Parametrisierung des Reglers über Einzelzugriffe auf alle Parameter (siehe Objektverzeichnis) eingestellt werden. Aber auch alle Bediendaten, die üblicherweise über PDOs übertragen werden, können über SDOs angesprochen werden.

### 3 Prozeß-Daten-Objekte (PDOs)

Mittels SDOs können maximal 4 Byte Nutzdaten zwischen zwei Partnern übertragen werden. Die restlichen 4 Byte werden für den Command, den Index und den SubIndex genutzt. Im Gegensatz dazu können mit einem PDO maximal 8 Byte Nutzdaten übertragen werden.

Vom KS 816 werden zwei Sende- und zwei Empfangs-PDOs unterstützt, die über die Konfiguration entweder als "Synchrone PDOs" oder als "Asynchrone PDOs" definiert werden können. Der Datentransfer über PDOs ist zum Beispiel dann sinnvoll, wenn Istwerte und Statuswerte des Reglers abhängig von z.B. Fehlermeldungen (Statusänderungen) zum Master gesendet werden müssen. In diesem Fall wird man asynchrone PDOs verwenden, die bei Eintreten des Events auf dem Regler automatisch zum Master gesendet werden. Ein anderes Beispiel wäre die zyklische Anforderung von Istwerten durch eine Bedieneinheit. In diesem Fall kann der KS 816 PDOs synchronisiert zu einem Messtakt (SYNC-EVENT) senden.

PDOs sind, im Gegensatz zu SDOs, grundsätzlich **nicht** bestätigte Dienste. Sie können von mehreren Teilnehmern empfangen werden. Dies gilt, wie oben schon erwähnt, nicht für SDOs.

In der Default Konfiguration des KS 816 sind je ein asynchrones Sende- und Empfangs-PDO mit folgenden Identifiern freigeschaltet.

- ⇒ COB-ID des asynchronen Empfangs-PDOs: 0x200 + Node-ID
- ⇒ COB-ID des asynchronen Sende-PDOs: 0x180 + Node-ID

Eine Umkonfiguration der PDOs erfolgt mittels SDO Schreibzugriffen auf folgende Indizes:

- ⇒ Index 1400: erstes Empfangs-PDO
- ⇒ Index 1401: zweites Empfangs-PDO
- ⇒ Index 1800: erstes Sende-PDO
- ⇒ Index 1801: zweites Sende-PDO

Bei einem Zugriff auf eine dieser Adressen bestimmt der Subindex die Konfigurationsart des PDOs.

Subindex = 0x01: Der Message-Identifizier (COB-ID) wird konfiguriert und das PDO freigegeben oder gesperrt. Folgende Daten werden übergeben:

- ⇒ PDO - COB-ID (low byte) data[ 1]
- ⇒ PDO - COB-ID (high byte) data[ 2]
- ⇒ Freigabe data[ 4] (0x00 = frei, 0x80 = gesperrt)

Subindex = 0x02: Der Transfer-Typ wird konfiguriert. Folgende Daten werden übergeben:

- ⇒ Transfertyp data[ 1] (0x01 = sync, 0xFF = async)

**Beispiel:**

**Das erste Sende-PDO wird freigegeben und als synchrones PDO konfiguriert**

Der Aufbau der Nachricht für die Freigabe ist wie folgt:

- ⇒ COB\_ID = SDO an Knoten 5 =  $0x600 + 5$  = 0x0605
- ⇒ CMD = Schreibzugriff = 0x2B = 0x2B
- ⇒ INDEX = erstes Sende-PDO = 0x1816 = 0x1816
- ⇒ SUBINDEX = COB-ID konfigurieren = 0x01 = 0x01
- ⇒ PDO - COB\_ID = PDO, Knoten 5 =  $0x280 + 5$  = 0x0285
- ⇒ Dummy = reserviert (data[3]) = 0x00 = 0x00
- ⇒ Freigabe = freigeben = 0x00 = 0x00

Der Aufbau der Nachricht für die Transfer-Typ-Konfiguration ist wie folgt:

- ⇒ COB\_ID = SDO an Knoten 5 =  $0x600 + 5$  = 0x0605
- ⇒ CMD = Schreibzugriff = 0x2B = 0x2B
- ⇒ INDEX = erstes Sende-PDO = 0x1816 = 0x1816
- ⇒ SUBINDEX = Transfer-Typ konfigurieren = 0x02 = 0x02
- ⇒ Transfer-Typ = synchron = 0x01 = 0x01

	8 data bytes									
	COB-ID	LEN	Cmd	Index		Sub.	Data			
Anforderung (Master)	0x605	0x08	0x2B	0x00	0x18	0x01	0x85	0x02	0x00	0x00
Antwort (KS816)	0x585	0x08	0x60	0x00	0x18	0x01	0x00	0x00	0x00	0x00

### 3.1 Information Record (Sende-PDO vom KS 816):

Der Information Record ist wie folgt aufgebaut:

Index	Subindex	Field	Data Type
42	0	Number of supported entries in the record	unsigned8
	1	channel number	unsigned8
	2	Xeff	FixedPoint1
	3	device status information	unsigned8
	4	channel status information	unsigned16
	5	Ypid (Reglerausgangsgröße)	FixedPoint1

Ein Send-PDO vom KS 816 enthält im Detail folgende Daten:

Kanal	→	data[ 0]
Xeff	→	data[ 1, 2]
low byte		data[ 1]
high byte		data[ 2]
Geräte-Statusinformation:	→	data[ 3]
Bit 0 *)	-	offline[0], online [1]
Bit 1 *)	-	Alarm_Signal_1
Bit 2 *)	-	Alarm_Signal_2
Bit 3 *)	-	Alarm_Signal_3
Kanal-Statusinformation	→	data[ 4, 5]
Bit 0	-	Alarm HH
Bit 1	-	Alarm H
Bit 2	-	Alarm L
Bit 3	-	Alarm LL
Bit 4	-	Alarm Sensor Fail
Bit 5	-	
Bit 6	-	
Bit 7	-	
Bit 8	-	W2 aktiv
Bit 9	-	Wint aktiv
Bit 10	-	Wanfah aktiv
Bit 11	-	Optimierung aktiv
Bit 12	-	Fehler bei Optimierung
Bit 13	-	Regler A/M
Bit 14	-	Regler abgeschaltet (coff)
Bit 15	-	Toggelt bei jeder Änderung von Y_pid
Y_pid	→	data[ 6,7]
low byte		data[ 6]
high byte		data[ 7]

\*) Eine Änderung erzeugt einen asynchronen PDO-Event

### 3.2 Control Record (Empfangs-PDO des KS 816):

Der Control Record ist wie folgt aufgebaut:

Index	Subindex	Field	Data Type
43	0	Nuber of supported entries in the record	unsigned8
	1	channel number	unsigned8
	2	set point	FixedPoint1
	3	manual controller output	Fixedpoint1
	4	control byte	unsigned8
	5	update byte	unsigned8

Über Empfangs-PDOs können im Detail folgende Daten übernommen werden:

Kanal	→	data[ 0]
Wvol	→	data[ 1, 2]
low byte		data[ 1]
high byte		data[ 2]
Reglerausgangswert	→	data[ 3, 4]
low byte		data[ 3]
high byte		data[ 4]
Control Information:	→	data[ 5]
Bit 0 - A[0] / M [0/1]		
Bit 1 - COFF [1]		
Bit 2 - W/W2 [0/1]		
Bit 3 - Wext/Wint [0/1]		
Bit 4 - Ostart [1]		
Update Information:	→	data[ 6]
Bit 0 - A/M [0/1]		
Bit 1 - COFF [1]		
Bit 2 - W/W2 [0/1]		
Bit 3 - Wext/Wint [0/1]		
Bit 4 - Ostart [1]		
Bit 6 - Yman [1]		
Bit 7 - Wvol [1]		

Über die Updateinformation kann selektiert werden, welche Daten aus dem Control-Record vom KS 816 übernommen werden. Ist eines der Bits gesetzt, werden die zugehörigen Daten auf dem KS 816 aktualisiert.

### 3.3 PDO Übertragungsverhalten

#### 3.3.1 Asynchrones Sende PDO

Wenn ein PDO als asynchrones PDO konfiguriert ist, wird bei Änderung von Statusinformationen eines Kanals ein PDO gesendet. Die Reihenfolge der Meldungen ergibt sich wie folgt: Die Überprüfung ob gesendet werden muß beginnt mit dem ersten Kanal. Wurde seit dem letzten Senden eine Statusänderung registriert (Flag fSendEvent gesetzt), wird ein PDO erzeugt, gesendet und das Flag zurückgesetzt. Nach erfolgreichem Senden wird der nächste Kanal bearbeitet. Sind beide Sende PDOs als asynchrone PDOs konfiguriert, werden zwei Kanäle gleichzeitig überprüft und gesendet.

#### Beispiel:

```

Kanal 1: fSendEvent = TRUE  → PDO für Kanal 1 senden
Kanal 2: fSendEvent = FALSE
Kanal 3: fSendEvent = TRUE  → PDO für Kanal 3 senden
Kanal 4: fSendEvent = FALSE
Kanal 5: fSendEvent = FALSE
Kanal 6: fSendEvent = FALSE
Kanal 7: fSendEvent = TRUE  → PDO für Kanal 7 senden
      .
      .
      .

Kanal 16: fSendEvent = FALSE
Kanal 1: fSendEvent = FALSE

```

Ein asynchrones PDO wird vom KS 816 immer bei Zustandswechsel eines der Status-Bits gesendet. Die vom KS 816 gesendete Nachricht sieht bei Default-Konfiguration wie folgt aus:

#### Beispiel:

Nachricht vom KS 816 (ASYNC-PDO):

#### 8 data bytes

COB-ID	LEN	Channel	Xeff	Dev.st.	Channel st.	Y_pid
0x180+no- de	8	0x05	0xBC 0x02	0x00	0x00 0x42	0xF4 0x01

```

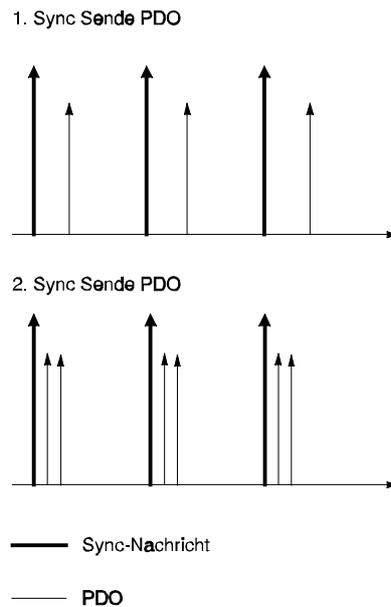
⇒ Actual Value    = 0x02BC= 700 = 70,0 C
⇒ Device Status   = 0x00      = keine Fehler
⇒ Channel Status  = 0x4200     = Coff = 1, Wint = 1
⇒ Y_pid           = 0x01F4 = 500 = 50,0%

```

### 3.3.2 Synchrones Senden PDO

Der Master sendet zum Refresh von Istwerten SYNC-Impulse. Nach wieviel Sync-Nachrichten der KS 816 reagieren soll, kann vom Master konfiguriert werden. Die Defaulteinstellung ist "1".

Im synchronen Betrieb wird nach jeder Sync-Nachricht ein Kanal, beginnend mit dem ersten, gesendet. Danach werden alle weiteren Kanäle der Reihe nach gesendet. Nach dem letzten Kanal wird beim ersten ein neuer Durchlauf gestartet. Sind beide Senden PDOs als synchrone PDOs konfiguriert, werden zwei Kanäle gleichzeitig übertragen. Das folgende Diagramm erläutert die Zusammenhänge zwischen Sync-Nachricht und PDOs.



### 3.3.3 Asynchroner Empfang PDO

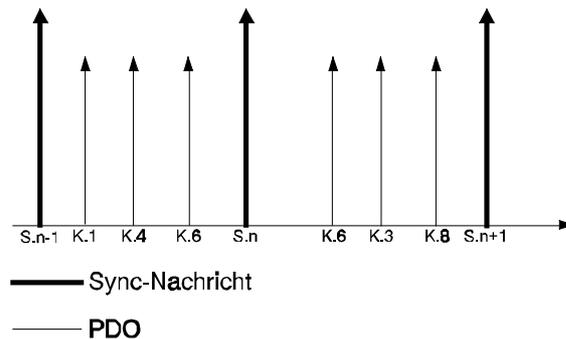
Wenn ein asynchrones PDO empfangen wird, werden die enthaltenen Daten sofort an das Reglermodul weitergereicht. Die Daten werden entsprechend der Kanalnummer in den Übergabebereich geschrieben. Dabei wird das Flag fReceived gesetzt, dies signalisiert dem Regler, daß neue Daten für ihn bereitstehen. Das Flag fReceived wird durch den Regler zurückgesetzt.

**Achtung:** Das Kommunikationsmodul überprüft nicht, ob der Regler die zuletzt empfangenen Daten bereits übernommen hat, wenn es neue Daten an den Regler übergibt. (Abtastrate beachten!) D.h. PDO's können überschrieben werden, ältere Informationen gehen verloren!

### 3.3.4 Synchrones Empfangs PDO

Wird ein synchrones PDO empfangen, verbleiben die Daten zunächst im Rx-Buffer (CAN-Controller) und werden noch nicht an den Regler übergeben. Bei Empfang der nächsten gültigen Sync-Nachricht werden sie an den Regler übergeben. Dazu wird das Flag fReceived für den entsprechenden Kanal gesetzt. Dies signalisiert dem Regler das neue Daten für ihn bereitstehen. Das Flag fReceived wird durch den Regler zurückgesetzt. Folgendes Diagramm zeigt den Zusammenhang zwischen Sync-Nachricht und Synchronen PDOs.

**Achtung:** Wird ein weiterer synchroner PDO vor der nächsten Sync-Nachricht empfangen, so wird das zuletzt empfangene PDO an den Regler übergeben. D.h. PDO's können überschrieben werden, ältere Informationen gehen verloren!



Erklärung des Diagramms: 800

Wenn nur 1 PDO auf sync konfiguriert ist, wird Kanal 6 (S.n), bzw. Kanal 8 (S.n+1) übergeben.

Bei 2 sync PDOs sind es Kanal 4+6, bzw. 3+8. Der Erste der 3er Gruppe geht immer verloren, da der/die Rx-Buffer überschrieben werden.

Erklärung des Diagramms: 816

Zum Zeitpunkt S.n werden die Daten der PDOs die im Zeitraum S.n-1 bis S.n empfangen wurden, an den Regler übergeben. Dies sind die Daten für Kanal 1, 4 und 6.

Zum Zeitpunkt S.n+1 werden die Daten der PDOs die im Zeitraum S.n bis S.n+1 empfangen wurden, an den Regler übergeben. Dies sind die Daten für Kanal 3, 6 und 8.

## 4 OP-Mode / Pre-OP-Mode

Nach dem Aufstart des CAN-Netzwerkes ist der KS 816 im Pre-OP-Mode. Das heißt, daß er nur über SDOs angesprochen werden kann. Wird der KS 816 in den OP-Mode gesetzt, kann auch über die freigegebenen PDOs kommuniziert werden. Bei Default Konfiguration sind die asynchronen PDOs nach Umschalten in den OP-Mode aktiv, synchrone PDOs müssen, falls erforderlich, über entsprechende SDO-Nachrichten freigegeben werden.

KS 816 in OP-Mode setzen:

COB-ID	LEN	2 data bytes	
		Command	Knoten
0x00	2	0x01	<Nummer> <sup>*)</sup>

Wird als Knotenadresse "0" angegeben, werden alle im Netz befindlichen Teilnehmer in den OP-Mode gesetzt. Die Regler senden keine Antwort auf diese Nachricht.

KS 816 in Pre-OP-Mode setzen:

COB-ID	LEN	2 data bytes	
		Command	Knoten
0x00	2	0x01	<Nummer> <sup>*)</sup>

<sup>\*)</sup> Knoten-Nummer 1 ... 127

Wird als Knotenadresse "0" angegeben, werden alle im Netz befindlichen Teilnehmer in den Pre-OP-Mode gesetzt. Die Regler senden keine Antwort auf diese Nachricht.

Weitere NMT (Netzwerk-Management) - Kommandos

Reset Node:

COB-ID	LEN	2 data bytes	
		Command	Knoten
0x00	2	0x81	<Nummer>

Reset Communication:

COB-ID	LEN	2 data bytes	
		Command	Knoten
0x00	2	0x82	<Nummer>

**Achtung:** Reset-Node führt zu einem Hardware-Reset des KS 816, d.h. er ist erst nach einigen Sekunden wieder über den Bus erreichbar. Alle Command-Parameter werden auf die Default-Werte zurückgesetzt.

## 5 SYNC

Wie bereits beschrieben werden synchrone PDO-Daten mittels SYNC-Nachrichten vom KS 816 ausgewertet oder gesendet. Die entsprechenden KS 816 müssen im OP-Mode sein und synchrone PDOs müssen konfiguriert sein. (Siehe 3. Prozeß-Daten-Objekte)

Eine SYNC-Nachricht ist ein sogenannter "Broadcast" (Nachricht an alle Teilnehmer im Netz) und wird wie folgt vom Master gesendet:

		0 data bytes	
COB-ID	LEN		
0x00	2		

Antwort vom KS 816:

**z.B. folgender PDO**

		8 data bytes				
COB-ID	LEN	Channel	Xeff	Dev. st.	Chan. st.	Ypid
0x280+node <sup>^</sup> )	8	0x05	0xBC 0x02	0x00	0x00 0x42	0xF4 0x01

- ⇒ Xeff = 0x02BC = 700 = 70,0°C
- ⇒ Device Status = 0x00 = keine Fehler
- ⇒ Channel Status = 0x4200 = Coff = 1, Wint = 1
- ⇒ Y\_pid = 0x01F4 = 500 = 50,0%

Nach einem erneuten SYNC vom Master werden die Daten des nächsten Kanals, in diesem Fall des Kanals 6, übertragen usw. Nach Übertragung des Kanals 16 wird wieder bei Kanal 1 begonnen. Sind mehrere KS 816 im Netz, werden pro SYNC-Event von jedem Regler die Daten eines Kanals gesendet. (Max. 2 Kanäle, wenn 2 sync. PDOs konfiguriert.)

<sup>^</sup>) Der Default-Wert des COB-ID von 0x80 lässt sich mit dem Objekt 1005 ändern.

## 6 Nodeguarding / Lifeguarding

Mittels "Nodeguarding" testet ein Master zyklisch, ob die Kommunikation zu den im Netz konfigurierten Slaves (z.B. KS 816) noch möglich ist. Bei "Lifeguarding" prüft ein Slave ob er noch regelmäßig von einem Master angesprochen wird.

Über die Parameter "guard time \* life time" kann festgelegt werden, in welchen Zeitabständen der Master die Nodeguard-Nachricht aktivieren muß (Timeout für den Master). Ist "guard time \* life time" = "0", wird kein Lifeguarding durchgeführt und der Master ist an kein Zeitraster gebunden. Ist "guard time \* life time" definiert und der KS 816 wird nicht im angegebenen Zeitraster abgefragt, fällt der KS 816 vom OP-Mode in den Pre-OP-Mode zurück. Es werden dann keine PDOs mehr übertragen.

Beim Nodeguarding antwortet der KS 816 durch "toggeln" des höchstwertigen Bits in der Antwortnachricht:

- Ist der KS 816 im Op-Mode, antwortet er wie folgt: 0x05 → 0x85 → 0x05 → 0x85 → ...
- Ist der KS 816 im Pre-Op-Mode, antwortet er wie folgt: 0x7F → 0xFF → 0x7F → 0xFF → ...

Node-Guarding Identifier → Objekt 100E Defaukt 6E0 + node ID

### Beispiel:

Nodeguard-Nachricht vom Master:

		0 data bytes	
COB-ID	LEN		
0x6E0+node	0		

Antwort vom KS 816 (Pre-OP-Mode):

		1 data bytes	
COB-ID	LEN	Answer	
0x6E0+node	1	0x7F	
0x6E0+node	1	0x7F	
0x6E0+node	1	0x7F	

Antwort vom KS 816 (OP-Mode):

		0 data bytes	
COB-ID	LEN	Answer	
0x6E0+node	1	0x05	
0x6E0+node	1	0x05	
0x6E0+node	1	0x05	

**Achtung:** Reset-Node führt zu einem Hardware-Reset des KS 800, d.h. er ist erst nach einigen Sekunden wieder über den CAN-Bus erreichbar. Alle Command-Parameter werden auf die Default-Werte zurückgesetzt.

## 7 LMT-Dienste (CMS-Standard und Erweiterungen)

Es stehen die LMT-Dienste gemäß CiA/DS205 Version 1.1 zur Verfügung. Einschränkungen, bzw. spezielle Erweiterungen sind den nachfolgenden Service-Beschreibungen zu entnehmen. Die zugehörigen COB-ID's sind 2021 (Master => Slave), bzw. 2020 (Slave => Master), wobei der KS 816 jeweils Slave ist.

### 7.1 Switch Mode Services

#### 7.1.1 Switch Mode Global

M => S

0	1	2	3	4	5	6	7
cs = 04	mode	r	r	r	r	r	r

cs           LMT command specifier  
mode        0: switches to operation mode  
              1: switches to configuration mode  
r            reserved

#### 7.1.2 Switch Mode Selective

M => S

0	1	2	3	4	5	6	7
cs = 01	m1	m2	m3	m4	m5	m6	m7

0	1	2	3	4	5	6	7
cs = 02	p1	p2	p3	p4	p5	p6	p7

0	1	2	3	4	5	6	7
cs = 02	s1	s2	s3	s4	s5	s6	s7

cs           LMT command specifier  
m1..7       manufacturer name part of LMT address  
p1..7       product name part of LMT address  
s1..7       serial number part of LMT address

## 7.2 Configuration Services

Configure NMT Address besteht aus den Diensten Configure Module ID und Configure Modul Name. Der Service Configure Modul Name wird nicht unterstützt. Ein entsprechender Request liefert error\_code: 255 und specific\_error\_code: 01 (service not supported).

### 7.2.1 Configure Module ID

M => S

0	1	2	3	4	5	6	7
cs = 17	MId	r	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 17	error code	spec error	r	r	r	r	r

cs                   LMT command specifier  
MId                   new module\_id to configure  
error\_code           0:    successful  
                      1..254:   reserved  
                      255:   implementation specific error  
specific\_error       only valid if error\_code equal 255  
                      3:    MId out of range  
                      4:    command execution error  
r                     reserved

### 7.2.2 Configure Bit Timing

M => S

0	1	2	3	4	5	6	7
cs = 19	table selector	table index	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 19	error code	spec error	r	r	r	r	r

cs                   LMT command specifier  
table\_selector       0:    standard CiA bit timing table  
table\_index          0:    1 Mbit  
                      1:    800 kbit  
                      2:    500 kbit  
                      3:    250 kbit  
                      4:    125 kbit  
                      5:    50 kbit  
                      6:    20 kbit  
                      7:    10 kbit  
error\_code           0:    successful  
                      1:    bit timing not supported  
                      2..254: reserved  
                      255: implementation specific error  
specific\_error       only valid if error\_code equal 255  
                      3:    value out of range  
                      4:    command execution error  
r                     reserved

Der Service *Activate Bit Timing* wird nicht unterstützt (unconfirmed service !), die Aktivierung erfolgt über einen Reset (siehe Adress-/BR-Schalter für die 3 Reset-Möglichkeiten).

Der Service *Store Configuration* wird nicht unterstützt Ein entsprechender Request liefert error\_code: 01 (service not supported).

### 7.3 Inquire LMT Address

Inquire LMT Address besteht aus den folgenden 3 Diensten:

#### 7.3.1 Inquire Manufacturer Name

M => S

0	1	2	3	4	5	6	7
cs = 36	r	r	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 36	m1	m2	m3	m4	m5	m6	m7

cs LMT command specifier  
 m1..m7 manufacturer name of the module (if m1 is a valid alpha-num character).  
 If m1 is 255, m2 contains error\_code and m3 optional error-reason.  
 (liefert "Firmen-Name" (PMA))

#### 7.3.2 Inquire Product Name

M => S

0	1	2	3	4	5	6	7
cs = 37	r	r	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 37	p1	p2	p3	p4	p5	p6	p7

cs LMT command specifier  
 p1..p7 product name of the module (if p1 is a valid alpha-num character).  
 If p1 is 255, p2 contains error\_code and p3 optional error-reason.  
 (liefert "System-Name" (Popen))  
 r reserved

### 7.3.3 Inquire Serial Number

M => S

0	1	2	3	4	5	6	7
cs = 38	r	r	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 38	s1	s2	s3	s4	s5	s6	s7

cs            LMT command specifier  
s1..s7        serial number of the module (if s1 is a valid BCD-pair).  
              If s1 is 255, s2 contains error\_code and s3 optional error-reason.  
              (liefert Geräte-Code-Nummer)  
r             reserved

### 7.3.4 Serial Number Format (14 BCD-digit => 7 byte)

0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	m1	m2	m3	m4	c1	c2	c3	c4	c5	c6	c7	c8

m1..m4      FabMonth, Herstellungsmonat in verschlüsselter Schreibweise  
c1..c8        FabCount, "unique" Zählnummer, einmalige Vergabe

## 7.4 Identification Services

### 7.4.1 LMT Identify Remote Slaves

M => S

0	1	2	3	4	5	6	7
cs = 05	manufact.name						

M => S

0	1	2	3	4	5	6	7
cs = 06	productname						

M => S

0	1	2	3	4	5	6	7
cs = 07	serialnumber low						

M => S

0	1	2	3	4	5	6	7
cs = 08	serialnumber high						

cs            LMT command specifier  
manufact.\_name    manufacturer name part of LMT address  
product\_name      product name part  
serial\_number\_low    lower boundary of requested range  
serial\_number\_high    higher boundary  
                      ('boundaries' sind im Intervall enthalten)

### 7.4.2 LMT Identify Slave

S => M

0	1	2	3	4	5	6	7
cs = 09	r	r	r	r	r	r	r

cs            LMT command specifier  
r             reserved

**Hinweis:** Bei mehreren Geräten am Bus kann es hier zu Problemen, speziell dem Verlust von Frames, kommen ! (bedingt durch die Anzahl der gleichzeitig antwortenden Teilnehmer mit identischen Frames)

### 7.5 Herstellerspezifische LMT-Dienste

Der Service *Activate Bit Timing* (cs = 131) wird nicht unterstützt (siehe auch gleichnamigen Std-Service).

### 7.6 Inquire Node-ID

M => S

0	1	2	3	4	5	6	7
cs = 128	CAN	r	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 128	error code	spec error	Node ID	r	r	r	r

cs            LMT command specifier  
CAN           number of CAN-controller (must be 0)  
error\_code    0:        successful  
              1..254: reserved  
              255:    implementation specific error  
specific\_error only valid if error\_code equal 255  
              6:        illegal CAN-controller selected  
Node ID       module Node ID

### 7.6.1 Configure Module-ID

M => S

0	1	2	3	4	5	6	7
cs = 129	Mid	CAN	r	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 129	error code	spec error	r	r	r	r	r

cs LMT command specifier  
 CAN number of CAN-controller (must be 0)  
 Mid new module\_id to configure  
 error\_code 0: successful  
 1..254: reserved  
 255: implementation specific error  
 specific\_error only valid if error\_code equal 255  
 3: Mid out of range  
 4: command execution error  
 6: illegal CAN-controller selected  
 r reserved

### 7.6.2 Configure Bit Timing

M => S

0	1	2	3	4	5	6	7
cs = 130	table selector	table index	CAN	r	r	r	r

S => M

0	1	2	3	4	5	6	7
cs = 130	error code	spec error	r	r	r	r	r

cs LMT command specifier  
 CAN number of CAN-controller (must be 0)  
 table\_selector 0: standard CiA bit timing table  
 table\_index 0: 1 Mbit  
 1: 500 kbit  
 2: 250 kbit  
 3: 125 kbit  
 4: -  
 5: 50 kbit  
 6: 20 kbit  
 7: 10 kbit  
 8: 800 kbit  
 error\_code 0: successful  
 1: bit timing not supported  
 2..254: reserved  
 255: implementation specific error  
 specific\_error only valid if error\_code equal 255  
 3: value out of range  
 4: command execution error  
 6: illegal CAN-controller selected  
 r reserved

## 7.6.3 Inquire Bit Timing

M =&gt; S

0	1	2	3	4	5	6	7
cs = 132	r	r	r	r	r	r	r

S =&gt; M

0	1	2	3	4	5	6	7
cs = 132	error code	spec error	table selector	table index	CAN	BTR0	BTR1

cs LMT command specifier  
 CAN number of CAN-controller (always 0)  
 BTR0/BTR1 bit timing register 0 and 1 of CAN-controller  
 table\_selector standard CiA bit timing table (always 0)  
 table\_index 0: 1 Mbit  
 1: 500 kbit  
 2: 250 kbit  
 3: 125 kbit  
 4: -  
 5: 50 kbit  
 6: 20 kbit  
 7: 10 kbit  
 8: 800 kbit  
 error\_code 0: successful  
 1: bit timing not supported  
 2..254: reserved  
 specific\_error only valid if error\_code equal 255  
 r reserved

## 8 Objektverzeichnis

Im folgenden werden die herstellerspezifischen listenorientierten Einträge des Objektverzeichnisses erläutert.

### 8.1 Übersicht Objektverzeichnis herstellerspezifischer Teil

Hinweis: Das gesamte Objektverzeichnis ist doppelt ausgelegt (ab Index 2001; ab Index 3001). Dies ist vorgesehen um gleiche Objekte mit unterschiedlichen Datentypen zu ermöglichen.

Aufbau der Indextabelle:

Der Index ist eine 4-stellige Zahl in hexadezimaler Notation mit folgendem Aufbau.

Format	Funktionsblock	Funktion	Parameter-Nr.
2 = Fixed Point Format	0 = Gerät 1 = Input 2 = Controller 3 = Alarm	laufende Nummer	laufende Nummer

Konfigurationsmode:

Änderungen der Konfiguration wie Ausgangszuordnungen oder Messbereichsendwerte werden nicht Online übernommen. Zur Änderung solcher Parameter muß der KS 816 in den Konfigurationsmode geschaltet werden. Die Mode-Umschaltung erfolgt über einen SDO-Zugriff.

Index 2008 (OPMod):	Wert =	0	Konfigurationsmode
	Wert =	1	Normalmode

**Zur Änderung von im Objektverzeichnis mit rw\* gekennzeichneten Daten muß OPMod auf 0 gesetzt werden. Die neuen Parameter werden beim anschließenden Rückändern des OPMod von 0 -> 1 übernommen.**

Anmerkung: Die Objekte, die in KS 816 gegenüber KS 800 entfallen sind, sind als durchgestrichen markiert!

#### Verwendete Datenformate:

Unsigned8: 8-Bit-Wert, ohne Vorzeichen, Bereich 0...256

Unsigned 16: 16-Bit-Wert, ohne Vorzeichen, Bereich 0...65535

Fixedpoint1: 16-Bit Integer-Wert mit 1 festen Kommastelle, Bereich -32767...+32767. Bei der Interpretation dieser Werte wird die letzte Stelle als Nachkommastelle gewertet. z.B. 12345 bedeutet 1234,5 (°C), oder 873 sind 87,3 (%).

Index (hex)	Objekt (Symbolic name)	Name	Type	Attribut
<b>FB Gerät</b>				
<b><i>Prozeßdaten Fktnr.0</i></b>				
2001	VAR	Status 1 Unit-State	Unsigned8	ro
2002	VAR	Basic Hardware Option HWbas	Unsigned16	ro
2003	VAR	SW Options SWOpt	Unsigned16	ro
2004	VAR	SW Code-Nr. SWCode	Unsigned16	ro
2005	VAR	SW Version SWVersion	Unsigned16	ro
2006	VAR	Bedienversion OPVers	Unsigned16	ro
2007	VAR	Versionsstand des EEPROMs EEPVers	Unsigned16	ro
2008	VAR	Konfigurationsmethode Umschaltung OPMoD	Unsigned8	rw
2009	VAR	Stoppen/Starten der RSE aller Gruppenregler OStartg	Unsigned8	rw
200A	VAR	Zurücksetzen des lokalen Datenänderungsflags	Unsigned8	rw
<b><i>Parameter und Konfigurationsdaten Fktnr. 0</i></b>				
200B	VAR	Baudrate COM1 C900	Unsigned16	rw*
200C	VAR	Geräteadresse Adr1	Unsigned16	rw*
200D	VAR	Netzfrequenz 50/60	Unsigned8	rw*
200E	VAR	Baudrate COM2 C900 (CAN Baudrate)	Unsigned16	rw*
200F	VAR	Geräteadresse ADR2 (CAN NODE ID)	Unsigned16	rw*
2010	VAR	Freigabe der Kühlenfunktion für Wasserkühlung	FixedPoint	-rw*
<b><i>Prozeßdaten Fktnr. 2</i></b>				
2020	VAR	<del>Status Alarmausgänge State_alarm_out</del>	Unsigned8	-ro
2021	VAR	<del>Status dig. Ein /Ausgängen State_dio</del>	Unsigned8	-ro
<b><i>Parameter und Konfigurationsdaten Fktnr. 2</i></b>				
2022	VAR	Hauptkonfig. C500	Unsigned16	rw*
2023	VAR	Hauptkonfig. C530	Unsigned16	rw*
2024	VAR	<del>Zuordnung HG/Leckstrom C151</del>	Unsigned16	-rw*
2025	VAR	<del>Heizstromzyklus Heeyel</del>	Unsigned16	-rw*
2026	VAR	<del>Meßbereichsende für HG Eingang HG 100</del>	FixedPoint1	-rw*
2027	VAR	<del>Forced digital output OUT1...OUT8</del>	Unsigned8	-rw*
2028	VAR	<del>Forced digital output OUT9...OUT16</del>	Unsigned8	-rw*
2029	VAR	<del>Forced digital output OUT17...OUT19</del>	Unsigned8	-rw*

<b>FB Input</b>				
<i>Prozeßdaten Fktnr. 0</i>				
2100	ARRAY	Signal Input Fail Input_X_Failed	Unsigned8	ro
2101	ARRAY	Hauptregelgröße	FixedPoint1	ro
2102	ARRAY	Rohmeßwert vor Meßwertkorrektur INP1	FixedPoint1	ro
<i>Parameter und Konfigurationsdaten Fktnr. 1</i>				
2110	ARRAY	Meßwertkorrektur X1 Input X1in	FixedPoint1	rw*
2111	ARRAY	Meßwertkorrektur X1 Output X1out	FixedPoint1	rw*
2112	ARRAY	Meßwertkorrektur X2 Input X2in	FixedPoint1	rw*
2113	ARRAY	Meßwertkorrektur X2 Output X2 out	FixedPoint1	rw*
2114	ARRAY	Sensortyp (T,H) C200	Unsigned16	rw*
2115	ARRAY	Fail: Fühlerbruch (T) C205	Unsigned16	rw*
2116	ARRAY	phys. Wert bei 0% X0	FixedPoint1	rw*
2117	ARRAY	phys. Wert bei 100% X100	FixedPoint1	rw*
2118	ARRAY	Ersatzwert bei Sensorfail XFail	FixedPoint1	rw*
2119	ARRAY	Filterzeitkonstante Meßwertverarbeitung Tfm	FixedPoint1	rw*
211A	ARRAY	angenommene TK Tkref	FixedPoint1	rw*
211B	ARRAY	Signalzuordnung dig. Signale C190	Unsigned16	rw*
<b>FB Controller</b>				
<i>Prozeßdaten Fktnr. 0</i>				
2200	ARRAY	Status 1	Unsigned8	ro
2201	ARRAY	eff. Sollwert Weff	FixedPoint1	ro
2202	ARRAY	eff. Istwert Xeff	FixedPoint1	ro
2203	ARRAY	wirksame stellgröße Ypid	FixedPoint1	ro
2204	ARRAY	Regelabweichung xw	FixedPoint1	ro
2205	ARRAY	Automatik-/Handumschaltung	Unsigned8	rw
2206	ARRAY	Starten der RSE OStart	Unsigned8	rw
2207	ARRAY	umschaltung Wext/Wint	Unsigned8	rw
2208	ARRAY	Umschalten W/W2	Unsigned8	rw
2209	ARRAY	Regler ein/aus Coff	Unsigned8	rw

<b>Parameter und Konfigurationsdaten Fktnr. 0</b>				
220A	ARRAY	Hauptkonfiguration 1, Regelung C100	Unsigned16	rw*
220B	ARRAY	Hauptkonfiguration 2, Regelung C101	Unsigned16	rw*
220C	ARRAY	Konfiguration Tuning C700	Unsigned16	rw*
220D	ARRAY	Signalzuordnung anal. C180	Unsigned16	rw*
<b>Prozeßdaten Fktnr. 1</b>				
2210	ARRAY	Sollwerte WState	Unsigned8	ro
2211	ARRAY	wirksamer interner Sollwert Wint	FixedPoint1	ro
2212	ARRAY	Int Sollwert, nicht flüchtig, Wnvol	FixedPoint1	rw
2213	ARRAY	Int Sollwert, flüchtig, Wvol	FixedPoint1	rw
<b>Parameter und Konfigurationsdaten Fktnr. 1</b>				
2214	ARRAY	Untere Sollwertgrenze für Weff W0	FixedPoint1	rw
2215	ARRAY	obere Sollwertgrenze für Weff W100	FixedPoint1	rw
2216	ARRAY	Zusatzsollwert W2	FixedPoint1	rw
2217	ARRAY	Sollwertgradient plus Grw+	FixedPoint1	rw
2218	ARRAY	Sollwertgradient minus Grw-	FixedPoint1	rw
2219	ARRAY	Sollwertgradient W2 Grw2	FixedPoint1	rw
<b>Parameter und Konfigurationsdaten Fktnr. 3</b>				
2230	ARRAY	Neutrale Zone Xsh	FixedPoint1	rw
2231	ARRAY	Mindestimpulslänge Tpuls	FixedPoint1	rw
2232	ARRAY	Motorlaufzeit des Stellmotors Tm	FixedPoint1	rw
2233	ARRAY	Schaltdifferenz Signalgerät Xsd1	FixedPoint1	rw
2234	ARRAY	Schaltpunktabstand Zusatzkontakt LW	FixedPoint1	rw
2235	ARRAY	Schaltdifferenz Zusatzkontakt Xsd2	FixedPoint1	rw
2236	ARRAY	Neutrale Zone Xsh1	FixedPoint1	rw
2237	ARRAY	Neutrale Zone Xsh2	FixedPoint1	rw
<b>Prozeßdaten Fktnr. 4</b>				
2240	ARRAY	Differenz Stellgrößenvorgabe dYman	FixedPoint1	rw
2241	ARRAY	absolute Stellgrößenvorgabe Yman	FixedPoint1	rw
2242	ARRAY	inrem. Stellgrößenverstellung Yinc	Unsigned8	rw
2243	ARRAY	decrem. Stellgrößenverstellung Ydec	Unsigned8	rw
2244	ARRAY	Geschwindigkeit für increm. und decrem. Stellgrößenverstellung Ygrw_is	Unsigned8	rw

<b>Parameter und Konfigurationsdaten Fktnr. 4</b>				
2245	ARRAY	untere Stellgrößenbegrenzung Ymin	FixedPoint1	rw
2246	ARRAY	obere Stellgrößenbegrenzung Ymax	FixedPoint1	rw
2247	ARRAY	Arbeitspunkt für Stellgröße Y=	FixedPoint1	rw
2248	ARRAY	Max. Mittelwert der Stellgröße Yhm	FixedPoint1	rw
2249	ARRAY	Grenze für Mittelwertbildung LYh	FixedPoint1	rw
<b>Prozeßdaten Fktnr. 5</b>				
2250	ARRAY	Status tuning State_Tune1	Unsigned8	ro
2251	ARRAY	Eff. Parameterzusatznummer ParNeff	Unsigned8	ro
2252	ARRAY	Parameterzusatznummer wirksam ParNr	Unsigned8	rw
2253	ARRAY	Verzugszeit Heizen Tu1	FixedPoint1	r0
2254	ARRAY	Anstiegsgeschwindigkeit Heizen Vmax1	FixedPoint1	ro
2255	ARRAY	Prozeßverstärkung Heizen Kp1	FixedPoint1	ro
2256	ARRAY	Fehlercode der RSE Heizen MSG1	Unsigned	ro
2257	ARRAY	Verzugszeit Kühlen Tu2	FixedPoint1	ro
2258	ARRAY	Anstiegsgeschwindigkeit Kühlen Vmax	FixedPoint1	ro
2259	ARRAY	Prozeßverstärkung Kühlen Vmax	FixedPoint1	ro
225A	ARRAY	Fehlercode der RSE Kühlen MSG2	Unsigned8	ro
<b>Paramter und Konfigurationsdaten Fktnr. 5</b>				
225B	ARRAY	Stellgröße bis Prozeß in Ruhe YOptm	FixedPoint1	rw
225C	ARRAY	Sprunghöhe bei Identifikation dYopt	FixedPoint1	rw
225D	ARRAY	Parametersatz, der optimiert werden soll POpt	Unsigned8	rw
225E	ARRAY	Hysterese bei Parameterumschaltung OXsd	FixedPoint1	rw
225F	ARRAY	Umschaltpunkt 1 Trig1	FixedPoint1	rw
<b>Parameter und Konfigurationsdaten Fktnr. 6</b>				
2260	ARRAY	Proportionalbereich 1 Xp1_1	FixedPoint1	rw
2261	ARRAY	Nachstellzeit 1Tn1_1	FixedPoint1	rw
2262	ARRAY	Vorhaltzeit 1 Tv1_1	FixedPoint1	rw
2263	ARRAY	min. Periodendauer 1 T1_1	FixedPoint1	rw
2264	ARRAY	Proportionalbereich 2 Xp2_1	FixedPoint1	rw
2265	ARRAY	Nachstellzeit 2 Tn2_1	FixedPoint1	rw
2266	ARRAY	Vorhaltzeit 2 Tv2_1	FixedPoint1	rw
2267	ARRAY	min. Periodendauer 2 T2_1		

<b>Parameter und Konfigurationsdaten Fktnr. 7</b>				
2270	ARRAY	Proportionalbereich 1 Xp1_2	FixedPoint1	rw
2271	ARRAY	Nachstellzeit 1 Tn1_2	FixedPoint1	rw
2272	ARRAY	Vorhaltezeit 1 Tv1_2	FixedPoint1	rw
2273	ARRAY	min. Periodendauer 1 T1_2	FixedPoint1	rw
2274	ARRAY	Proportionalbereich 2 Xp2_2	FixedPoint1	rw
2275	ARRAY	Nachstellzeit 2 Tn2_2	FixedPoint1	rw
2276	ARRAY	Vorhaltezeit 2 Tv2_2	FixedPoint1	rw
2277	ARRAY	min. Periodendauer 2 T2-2	FixedPoint1	rw
<b>Paramter und Konfigurationsdaten Fktnr. 10</b>				
22A0	ARRAY	Max. Stellwert Ya	FixedPoint1	rw
22A1	ARRAY	Anfahrswert Wa	FixedPoint1	rw
22A2	ARRAY	Anfahrhaltezeit TPa	FixedPoint1	rw
<b>FB Alarm</b>				
<b>Prozeßdaten Fktnr. 0</b>				
2300	ARRAY	Alarmstatus 1 Status_AI1	Unsigned8	ro
2301	ARRAY	Heizstromalarm-Status_AI2	Unsigned8	-ro
2302	ARRAY	Heizstrommeßwert HG	FixedPoint1	-ro
<b>Parameter und Konfigurationsdaten Fktnr. 0</b>				
2303	ARRAY	unterer Voralarm LimL	FixedPoint1	rw
2304	ARRAY	obererVoralarm LimH	FixedPoint1	rw
2305	ARRAY	Schaltdifferenz Voralarme xsd_2	FixedPoint1	rw
2306	ARRAY	unterer Hauptalarm LimLL	FixedPoint1	rw
2307	ARRAY	oberer Hauptalarm LimHH	FixedPoint1	rw
2308	ARRAY	Heizstromgrenzwert LimHG	FixedPoint1	rw
2309	ARRAY	Src: Signalquelle (T,H) C600 Fnc: Funktion (Z)	Unsigned16	rw*
230A	ARRAY	Alarmziel C601	Unsigned16	-rw*

Index (hex)	Objekt (Symbolic name)	Name	Type	Attribut
<b>FB Gerät</b>				
<b>Prozeßdaten Fktnr. 0</b>				
3001	VAR	Status 1 Unit-State 1	Unsigned8	ro
3002	VAR	Basic Hardware Options HWbas	Unsigned16	ro
3003	VAR	SW Optionen SWopt	Unsigned16	ro
3004	VAR	SW Code-Nr. SWCode	Unsigned16	ro
3005	VAR	SW Version SWVersion	Unsigned16	ro
3006	VAR	Bedienversion OPVers	Unsigned16	ro
3007	VAR	Versionsstand des EEPROMs EEPVers	Unsigned16	ro
3008	VAR	Konfigurationsmode Umschaltung OPMoD	Unsigned8	rw
3009	VAR	Stoppen/Starten der RSE aller Gruppenregler OStartg	Unsigned8	rw
300A	VAR	Zurücksetzen des lokalen Datenänderungsflags	Unsigned8	rw
<b>Parameter und Konfigurationsdaten Fktnr. 0</b>				
300B	VAR	Baudrate COM 1 C900	Unsigned16	rw*
300C	VAR	Geräteadresse Adr1	Unsigned16	rw*
300D	VAR	Netzfrequenz 50/60	Unsigned8	rw*
300E	VAR	Baudrate COM2 C900 (CAN Baudrate)	Unsigned16	rw*
300F	VAR	Geräteadresse Adr2 (CAN Node ID)	Unsigned16	rw*
<b>Prozeßdaten Fktnr. 2</b>				
<del>3020</del>	<del>VAR</del>	<del>Status Alarmausgänge State_alarm_out</del>	<del>Unsigned8</del>	<del>ro</del>
<del>3021</del>	<del>VAR</del>	<del>Status dig. Ein/Ausgänge State_die</del>	<del>Unsigned8</del>	<del>ro</del>
<b>Parameter und Konfigurationsdaten Fktnr. 2</b>				
3022	VAR	Hauptkonfi. C500	Unsigned16	rw*
3023	VAR	Hauptkonfig. C530	Unsigned16	rw*
<del>3024</del>	<del>VAR</del>	<del>Zuordnung HC/Leckstrom C151</del>	<del>Unsigned16</del>	<del>rw*</del>
<del>3025</del>	<del>VAR</del>	<del>Heizstromzyklus Hccycl</del>	<del>Unsigned16</del>	<del>rw*</del>
3026	VAR	Meßbereichsende für HC Eingang HC100	Float	rw*

<b>FB Input</b>				
<i>Prozeßdaten Fktnr: 0</i>				
3100	ARRAY	Signal Input Fail Input_X_Failed	Unsigned8	ro
3101	ARRAY	Hauptregelgröße X1	Float	ro
3102	ARRAY	Rohmeßwert vor Meßwertkorrektur INP1	Float	ro
<i>Parameter und Konfigurationsdaten Fktnr. 1</i>				
3110	ARRAY	Meßwertkorrektur X1 Input X1in	Float	rw*
3111	ARRAY	Meßwertkorrektur X1 Output X1out	Float	rw*
3112	ARRAY	Meßwertkorrektur X2 Xsin	Float	rw*
3113	ARRAY	Meßwertkorrektur X2 X2out	Float	rw*
3114	ARRAY	Sensor typ (T,H) C200	Unsigned16	rw*
3115	ARRAY	Fail: Fühlerbruch (T) C205	Unsigned16	rw*
3116	ARRAY	phys. Wert bei 0% X100	Float	rw*
3117	ARRAY	phys. Wert bei 100% X100	Float	rw*
3118	ARRAY	Ersatzwert bei Sensorfail XFail	Float	rw*
3119	ARRAY	Filterzeitkonstante Meßwertverarbeitung Tfm	Float	rw*
311A	ARRAY	angenommene TK Tkref	Float	rw*
311B	ARRAY	<del>Signalzuordnung dig-Signale G190</del>	Unsigned	<del>rw*</del>
<b>FB Controller</b>				
<i>Prozeßdaten Fktnr. 0</i>				
3200	ARRAY	Status 1	Unsigned8	ro
3201	ARRAY	eff. Sollwert Weff	Float	ro
3202	ARRAY	eff. Istwert Xeff	Float	ro
3203	ARRAY	wirksame Stellgröße Ypideff	Float	ro
3204	ARRAY	Regelabweichung xw	Float	ro
3205	ARRAY	Automatik/Hand Umschaltung	Unsigned8	rw
3206	ARRAY	Starten der RSE OStart	Unsigned8	rw
3207	ARRAY	UmschaltenWext/Wint	Unsigned8	rw
3208	ARRAY	Umschalten W/W2	Unsigned8	rw
3209	ARRAY	Regler ein/aus Coff	Unsigned8	rw

<b>Parameter und Konfigurationsdaten Fktnr. 0</b>				
320A	ARRAY	Hauptkonfiguration 1, Regelung C100	Unsigned16	rw*
320B	ARRAY	Hauptkonfiguration 2, Regelung C101	Unsigned16	rw*
320C	ARRAY	Konfiguration Tuning C700	Unsigned16	rw*
320D	ARRAY	Signalzuordnung anal. C180	Unsigned16	rw*
<b>Parameter und Konfigurationsdaten Fktnr. 1</b>				
3210	ARRAY	Sollwertstatus WState	Unsigned8	ro
3211	ARRAY	wirksamer interner Sollwert Wint	Float	ro
3212	ARRAY	int. Sollwert, nicht flüchtig Wnvol	Float	rw
3213	ARRAY	int. Sollwert, flüchtig Wvol	Float	rw
<b>Parameter und Konfigurationsdaten Fktnr. 1</b>				
3214	ARRAY	untere Sollwertgrenze für Weff W0	Float	rw
3215	ARRAY	Obere Sollwertgrenze für Weff W100	Float	rw
3216	ARRAY	Zusatzsollwert W2	Float	rw
3217	ARRAY	Sollwertgradient plus Grw+	Float	rw
3218	ARRAY	Sollwertgradient minus Grw-	Float	rw
3219	ARRAY	Sollwertgradient W2 Grw2	Float	rw
<b>Parameter und Konfigurationsdaten Fktnr. 3</b>				
3230	ARRAY	Neutrale Zone Xsh	Float	rw
3231	ARRAY	Mindestimpulslänge Tpuls	Float	rw
3232	ARRAY	Motorlaufzeit des Stellmotors Tm	Float	rw
3233	ARRAY	Schaltdifferenz Signalgerät Xsd1	Float	rw
3234	ARRAY	Schaltpunktabstand Zusatzkontakt LW	Float	rw
3235	ARRAY	Schaltdifferenz Zusatzkontakt Xsd2	Float	rw
3236	ARRAY	Neutrale Zone Xsh1	Float	rw
3237	ARRAY	Neutrale Zone Xsh2	Float	rw
<b>Prozeßdaten Fktnr. 4</b>				
3240	ARRAY	Differenz Stellgrößenvorgabe dYman	Float	rw
3241	ARRAY	absolute Stellgrößenvorgabe Yman	Float	rw
3242	ARRAY	incred. Stellgrößenvorgabe Yinc	Unsigned8	rw
3243	ARRAY	decrem. Stellgrößenvorgabe Ydec	Unsigned8	rw
3244	ARRAY	Geschwindigkeit für increm. und decrem. Stellgrößenverstellung Ygrw_is	Unsigned8	rw

<b>Parameter und Konfigurationsdaten Fktnr. 4</b>				
3245	ARRAY	untere Stellgrößenbegrenzung Ymin	Float	rw
3246	ARRAY	obere Stellgrößenbegrenzung Ymax	Float	rw
3247	ARRAY	Arbeitspunkt für stellgröße Y0	Float	rw
3248	ARRAY	max. Mittelwert der Stellgröße Yhm	Float	rw
3249	ARRAY	Grenze für Mittelwertbildung LYh	Float	rw
<b>Prozeßdaten Fktnr. 5</b>				
3250	ARRAY	Status Tuning State_Tune1	Unsigned8	ro
3251	ARRAY	eff. Parameterzusatznummer ParNeff	Unsigned8	ro
3252	ARRAY	Parameterzusatznummer wirksam ParNr	Unsigned8	rw
3253	ARRAY	Verzugszeit Heizen Tu1	Float	ro
3254	ARRAY	Anstiegsgeschwindigkeit Heizen Vmax1	Float	ro
3255	ARRAY	Prozeßverstärkung Heizen Kp1	Float	ro
3256	ARRAY	Fehlercode der RSE Heizen MSG1	Unsigned8	ro
3257	ARRAY	Verzugszeit Kühlen Tu2	Float	ro
3258	ARRAY	Anstiegsgeschwindigkeit Kühlen Vmax	Float	ro
3259	ARRAY	Prozeßverstärkung Kühlen Kp2	Float	ro
325A	ARRAY	Fehlercode der RSE Kühlen MSG2	Unsigned8	ro
<b>Parameter und Konfigurationsdaten Fktnr. 5</b>				
325B	ARRAY	Stellgröße bis Prozeß in Ruhe YOptm	Float	rw
325C	ARRAY	Sprunghöhe bei Identifikation dYOpt	Float	rw
325D	ARRAY	Parametersatz, der optimiert werden soll POpt	Unsigned8	rw
325E	ARRAY	Hysterese bei Parameterumschaltung OXsd	Float	rw
325F	ARRAY	Umschaltpunkt 1 Trig1	Float	rw
<b>Parameter und Konfigurationsdaten Fktnr. 6</b>				
3260	ARRAY	Proportionalbereich 1 Xp1_1	Float	rw
3261	ARRAY	Nachstellzeit 1 Tn1_1	Float	rw
3262	ARRAY	Vorhaltzeit 1 Tv1_1	Float	rw
3263	ARRAY	min. Periodendauer 1 Tn1_1	Float	rw
3264	ARRAY	Proportionalbereich 2 Xp2_2	Float	rw
3265	ARRAY	Nachstellzeit 2 Tn2_2	Float	rw
3266	ARRAY	Vorhaltzeit 2 Tv2_2	Float	rw
3267	ARRAY	min. Periodendauer 2 T2_2	Float	rw

<b><i>Parameter und Konfigurationsdaten Fktnr. 7</i></b>				
3270	ARRAY	Proportionalbereich 1 Xp1_2	Float	rw
3271	ARRAY	Nachstellzeit 1 Tn1_2	Float	rw
3272	ARRAY	Vorhaltzeit 1 Tv1_2	Float	rw
3273	ARRAY	min. Periodendauer 1 T1_1	Float	rw
3274	ARRAY	Proportionalbereich 2 Xp2_2	Float	rw
3275	ARRAY	Nachstellzeit 2 Tn2_2	Float	rw
3276	ARRAY	Vorhaltzeit 2 Tv2_2	Float	rw
3277	ARRAY	min. Periodendauer 2 T2_2	Float	rw
<b><i>Parameter und Konfigurationsdaten Fktnr. 10</i></b>				
32A0	ARRAY	max. Stellwert Ya	Float	rw
32A1	ARRAY	Anfahrswert Wa	Float	rw
32A2	ARRAY	Anfahrhaltezeit TPa	Float	rw
<b>FB Alarm</b>				
<b><i>Prozeßdaten Fktnr. 0</i></b>				
3300	ARRAY	Alarmstatus 1 Status_AL1	Unsigned8	ro
3301	ARRAY	Heizstromalarm-Status_AL2	Unsigned8	ro
3302	ARRAY	Heizstrommeßwert HG	Float	ro
<b><i>Parameter und Konfigurationsdaten Fktnr. 0</i></b>				
3303	ARRAY	unterer Voralarm LimL	Float	rw
3304	ARRAY	oberer Voralarm LimH	Float	rw
3305	ARRAY	Schaltdifferenz Voralarme xsd_2	Float	rw
3306	ARRAY	unterer Hauptalarm LimLL	Float	rw
3307	ARRAY	oberer Hauptalarm LimHH	Float	rw
3308	ARRAY	Heizstromgrenzwert LimHG	Float	rw
3309	ARRAY	Src: Signalquelle (T,H) C600 Fnc: Funktion (Z)	Unsigned16	rw <sup>*</sup>
330A	ARRAY	Alarmziel C601	Unsigned16	rw <sup>±</sup>

## 9 Applikationsbeispiele

### 9.1 Definition der Beispiele

Nach Auslieferung des KS 816 ist der Regler mit den in der Reglerbeschreibung angegebenen Default-Parametern eingestellt. Die folgenden Beispiele sollen den Ablauf von Umkonfigurierung und Inbetriebsetzung des Reglers verdeutlichen. In Beispiel 1 soll ein Heizen/Kühlen-Regler mit PT100 Eingang betrieben werden. Während in Beispiel 1 die Default-Regelparameter genutzt werden soll im 2. Beispiel die Selbstoptimierung zu den optimalen Parametern führen.

### 9.2 Beispiel 1 (Heizen/Kühlen-Regler)

Nachfolgend wird der Regler von der Defaulteinstellung (Heizen) auf Heizen/Kühlen umparametrisiert. Dann werden die Istwerte zyklisch (via SDOs) gelesen. Im Beispiel wird Kanal 1 genutzt. Der Subindex ist daher entsprechend der Kanalnummer bei allen Zugriffen auf "1" zu setzen.

- Konfigurationsmode wählen
  - **Index 0x2008** (OPMod=0)
  - Verweis: Prozeßdaten, Einzelcode 31
- Reglerhaupteinstellung C100=0x0300 setzen (H/K-Regler)
  - **Index 0x200A**
  - Verweis: C100 - Reglerbeschreibung
- Haupteinstellung C200=0x2010 setzen (PT100, Einheit: C)
  - **Index 0x2114**
  - Verweis: C200 - Reglerbeschreibung
- Konfigurationsmode verlassen und Werte übernehmen
  - **Index 0x2008** (OPMod=1)
  - Verweis: Prozeßdaten, Einzelcode 31
- Sollwert Weff setzen
  - **Index 0x2201** (Weff=250)
- Regler in AUTO setzen
  - **Index 0x2205** (A/H=0)
  - Verweis: Controller, Einzelcode 33
- Schleife für Istwerte:
  - **for(;;)**
    - {
    - Effektiver Istwert Kanal 1 lesen
      - **Index 0x2202**
    - wait( 1sec)
    - }

### 9.3 Beispiel 2 (H/K-Regler einstellen und Selbstoptimierung aktivieren)

Nachfolgend wird nach der Konfigurierung als H/K-Regler eine Selbstoptimierung gestartet. Nach erfolgreicher Optimierung werden die Istwerte zyklisch (via SDOs) gelesen. Wie in Beispiel 1 ist der Subindex immer auf "1" zu setzen (Regler 1).

- Konfigurationsmode wählen
  - **Index 0x2008** (OPMod=0)  
Verweis: Prozeßdaten, Einzelcode 31
- Reglerhauptkonfiguration C100=0x0300 setzen (H/K-Regler)
  - **Index 0x200A**  
Verweis: C100 - Reglerbeschreibung
- Hauptkonfiguration C200=0x2010 setzen (PT100, Einheit: C)
  - **Index 0x2114**  
Verweis: C200 - Reglerbeschreibung
- Konfigurationsmode verlassen und Werte übernehmen
  - **Index 0x2008** (OPMod=1)  
Verweis: Prozeßdaten, Einzelcode 31
- Sollwert Weff setzen
  - **Index 0x2201** (Weff=250)
- Optimierung starten
  - **Index 0x2206** (OStart=1)  
Verweis: Controller, Einzelcode 34
  - OPTIMIERUNG = GESTARTET
- Optimierungsstatus auswerten
  - **while(OPTIMIERUNG <> BEENDET)**
    - {
    - Optimierungsstatus lesen
      - **Index 0x2250** (State\_Tune\_1)  
Verweis: Controller, Einzelcode 1
    - fallende Flanke des Orun-bits abwarten
      - if ( State\_Tune\_1[ Orun] == 1)  
          OPTIMIERUNG = AKTIV
      - if ( State\_Tune\_1[ Orun] == 0 AND OPTIMIERUNG ==

AKTIV)

OPTIMIERUNG = BEENDET

- wait( 1sec)
- }

- Fehlermeldungen prüfen
  - **if( State\_Tune1[ Oerr == 0])**
    - {
    - Regler läuft mit neuen Parametern, da Optimierung erfolgreich
    - Schleife für Istwerte:
      - **for(;;)**
        - {

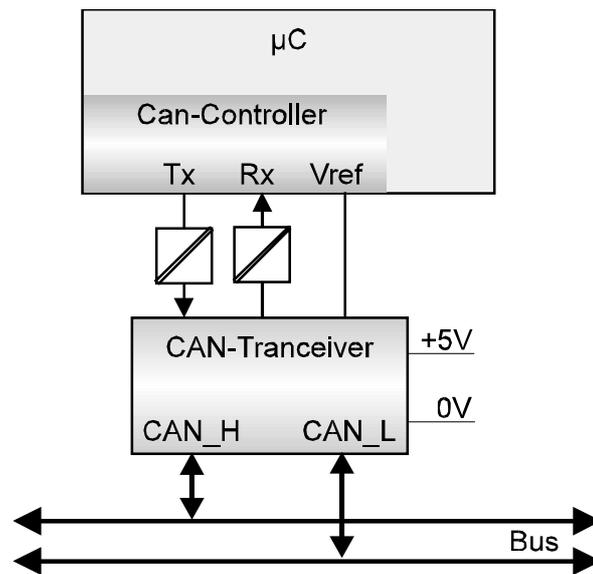
```
-      Effektiver Istwert Kanal 1 lesen  
      →      Index 0x2202  
-      wait( 1sec)  
      }  
}  
else  
{  
-      Fehlermeldung Heizen lesen  
      →      Index 0x2256 (MSG1)  
            Verweis: Controller, Einzelcode 35  
  
-      Fehlermeldung Kühlen lesen  
      →      Index 0x225A (MSG2)  
            Verweis: Controller, Einzelcode 39  
}
```

## 10 CAN Physical Layer

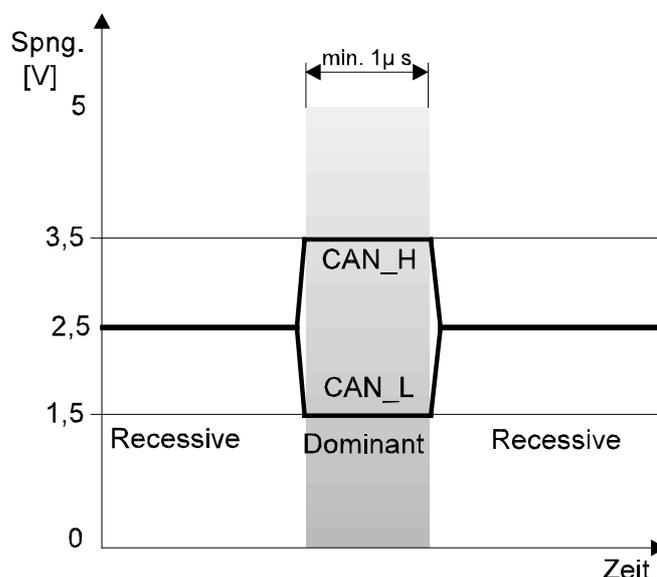
Es gibt eine Reihe von genormten Standards bezüglich des CAN Physical Layers. Der wichtigste für allgemeine Anwendungen ist der "CAN High-Speed Standard ISO 11898-2". Die nachfolgenden Empfehlungen basieren primär auf diesem Standard und gelten unabhängig vom verwendeten CAN-Protokoll (CANopen / DeviceNet).

### 10.1 ISO 11898-2 Knoten:

Ein ISO 11898-2 konformer Knoten besteht aus einem  $\mu\text{C}$  mit CAN-Controller (evtl. auch integriert), der über Rx- und Tx-Line mit einem CAN-Transceiver verbunden ist. Der Transceiver wiederum ist mit den differentiellen CAN-H und CAN-L Leitungen am CAN-Bus angeschlossen. Dieser (Transceiver-) Anschluß ist beim KS 800 galvanisch getrennt ausgeführt.



Die nominellen CAN-Buspegel werden beim CAN-Bus mit "Recessive" (nominelle Spannung von 2,5 V für CAN-H und CAN-L) und "Dominant" (nominell 3,5V für CAN-H und 1,5V für CAN-L) bezeichnet.



## 10.2 Baudraten und Buslängen

Die maximale, nutzbare Buslänge in einem CAN-Netzwerk wird durch eine Vielzahl von Einflüssen bestimmt, vor allem durch die folgenden physikalischen Effekte:

- Verzögerungszeiten der angeschlossenen Bus-Knoten (mit/ohne Opto-Koppler) und Verzögerungszeit des Bus-Kabels (propagation delays)
- unterschiedliche Abtastzeitpunkte innerhalb einer CAN-Bit-Zelle, bedingt durch Oszillatortoleranzen der einzelnen Bus-Knoten
- Signal-Amplituden Dämpfung, bedingt durch den ohmschen Widerstand des Bus-Kabels und den Eingangs-Widerständen der Bus-Knoten

Die im folgenden aufgeführten praktischen Buslängen können bei Verwendung von ISO 11898-2 konformen Transceivern mit Standard Buskabeln erreicht werden.

Bei den hohen Baudraten (1 MBd / 800 kBd) kann es allerdings durch die Anzahl/Geschwindigkeit der evtl. vorhanden Opto-Koppler (galv. Trennung) zu erheblich kürzeren Buslängen kommen !

## 10.3 Praktische Buslängen

CAN-Profil(e)	Baud-Rate	Buslänge	Nominelle Bit-Time
CANopen	1 MBd	30 m	1 $\mu$ s
CANopen	800 kBd	50 m	1,25 $\mu$ s
CANopen/DeviceNet	500 kBd	100 m	2 $\mu$ s
CANopen/DeviceNet	250 kBd	200 m	4 $\mu$ s
CANopen/DeviceNet	125 kBd	500 m	8 $\mu$ s
CANopen	50 kBd	1000 m *)	20 $\mu$ s
CANopen	20 kBd	2500 m *)	50 $\mu$ s
CANopen	10 kBd	5000 m *)	100 $\mu$ s

\*) Bei großen Kabellängen ist der Einsatz von galv. Trennungen und Repeatern zwingend notwendig

Weitere Hinweise zu den Buslängen können auch den Standards CiA „DS-102“ (CANopen) bzw. den ODVA "DeviceNet Specifications Volume I, Release 2.0", speziell Appendix A und B entnommen werden.

10.4 Kabel-Parameter

ISO 11898-2 definiert einige DC- bzw. AC-Parameter für die in CAN-Bus Netzwerken einsetzbaren Kabel (typischerweise kommen paarweise verdrehte Kabel mit definierten elektrischen Eigenschaften zum Einsatz). Die wichtigen AC Parameter sind 120 Ohm Kabelimpedanz und eine nominelle "propagation delay" von 5 ns/m ! Empfehlungen für die zu verwendenden Buskabel und Abschlußwiderstände können der folgenden Tabelle entnommen werden:

Bus-Länge	Buskabel (Z: 120 Ohm, tp: 5ns/m)		Abschlußwiderstand	Max. Bit-Rate
	Spez. Widerstand	Kabelquerschnitt		
0 ..40 m	70 mOhm/m	0,25mm <sup>2</sup> , 0,34mm <sup>2</sup> AWG 23, AWG 22	124 Ohm, 1%	1 MBd @ 40m
40 m .. 300 m	<60 mOhm/m	0,34mm <sup>2</sup> , 0,6mm <sup>2</sup> AWG 22, AWG 20	127 Ohm, 1% *)	> 500 kBd @ 100m
300 m .. 600 m	<40 mOhm/m	0,5mm <sup>2</sup> , 0,6mm <sup>2</sup> AWG 20	127 Ohm, 1% *)	> 100 kBd @ 500m
600 m ..1 km	<26 mOhm/m	0,75mm <sup>2</sup> , 0,8mm <sup>2</sup> AWG 18	127 Ohm, 1% *)	> 50 kBd @ 1 km

\*) Bei großen Kabellängen ist ein höherer Wert für den Abschlußwiderstand (150 .. 300 Ohm) hilfreich, zur Reduzierung der Dämpfung .

Weitere Empfehlungen für CAN Netzwerke (speziell auch mit großer Ausdehnung):

- galv. Trennungen sind notwendig bei großen Längen (z.B. bei 400m Buskabel)
- separate Ground-Leitung ist sinnvoll
- der Spannungseinbruch (Potentialdifferenz) zwischen den Ground-Potentialen der Transceiver sollte gering sein (kleiner 2V). Einspeisung des Netzteils evtl. in der Mitte des Kabels)
- der Gesamt-Eingangswiderstand der Bus-Knoten sollte > 500 Ohm sein
- evtl. notwendige Stichleitungen sollten so kurz wie möglich sein, um Reflektionen zu vermeiden/verringern, z.B. < 6m @ 500kBd (DeviceNet). Bei höheren Baudraten < 1m !

Weitergehende Informationen sind bei der ODVA (DeviceNet), dem CiA (CANopen), den diversen Chip-Herstellern und im Internet zu bekommen.



