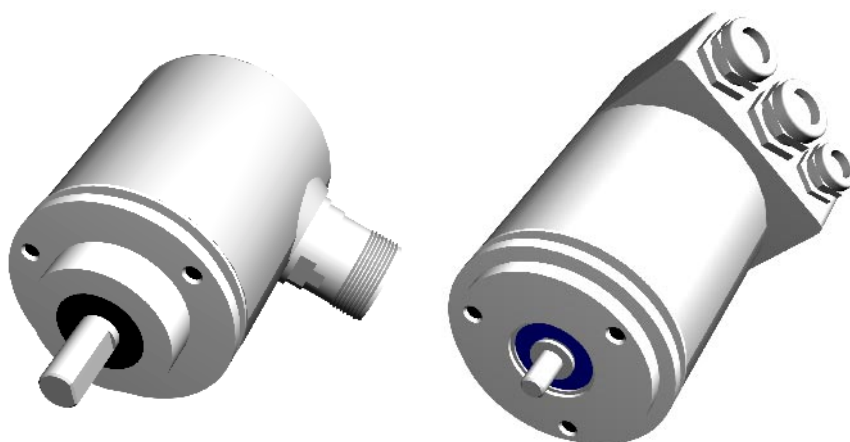


## Anwenderhandbuch



# Inhalt

<b>1. Allgemeines .....</b>	<b>3</b>
<b>2. Installationshinweise für PROFIBUS-DP - RS 485 .....</b>	<b>3</b>
2.1 Anschluß Winkelcodierer mit RS-Stecker .....	4
2.2 Anschluß Winkelcodierer mit Anschlußhaube .....	4
<b>3. Konfigurationsfunktion (DDL_M_Chk_Cfg) .....</b>	<b>6</b>
<b>4. Data Exchange Funktion (DDL_M_Data_Exchange) .....</b>	<b>7</b>
4.1 Positions-Istwert (Input-Data) .....	7
4.2 Referenzwert setzen (Output-Data) .....	7
<b>5. Betriebsparameter für Class 1/2 Winkelcodierer (DDL_M_Set_Prm) /4/ .....</b>	<b>9</b>
5.1 Definition der Programmierparameter .....	10
5.2 Hinweise zum Geschwindigkeitssignal .....	10
<b>6. Diagnosemeldungen (DDL_M_Slave_Diag) .....</b>	<b>13</b>
6.1 Standarddiagnoseinformation (Octet 1-6): .....	13
6.2 Gerätebezogene Diagnose .....	13
6.2.1 Herstellerspezifische Diagnose (Octet 60-63) .....	14
6.2.2 Beispiel für eine Diagnosemeldung .....	14
<b>7. Simatic Step7 .....</b>	<b>16</b>
7.1 Einbinden des TWK-Profibus-Winkelcodierers .....	16
7.1.1 Installation der GSD-Datei .....	16
7.1.2 Installation des TWK-Winkelcodierer Symbols .....	16
7.1.3 Auswahl des TWK-Winkelcodierers aus dem Step7 Hardwarekatalog .....	16
7.1.4 Konfigurieren des Winkelcodierers .....	16
7.1.5 Profibus-Adresse vergeben .....	17
7.1.6 Einstellen der E/A-Adressen (S7-Adressen) .....	18
7.1.7 Parametrieren des Winkelcodierers .....	18
7.1.8 Diagnose-Adresse einstellen .....	19
7.2 Einstellung der Teilnehmeradresse beim Winkelcodierer in Steckerversion .....	19
7.3 Beispielprogramme .....	20
7.3.1 Das Projekt TWKDPCL1 .....	20
7.3.2 Das Projekt TWKDPCL2 .....	21
7.3.3 Installation der Beispielprogramme .....	21
7.3.4 Erläuterungen zu den Beispielprogrammen .....	22
<b>8. Literatur .....</b>	<b>23</b>
<b>Anhang: Begriffe Winkelcodierer .....</b>	<b>23</b>

COPYRIGHT: The Operating Instructions TZY 11109  
is owned by TWK-ELEKTRONIK GMBH and is  
protected by copyright laws and international treaty provisions.

© 2003 by TWK-ELEKTRONIK GMBH  
POB 10 50 63 ■ 40041 Düsseldorf ■ Germany  
Tel. +49/211/96117-0 ■ Fax +49/211/96117-99  
e-mail: [info@twk.de](mailto:info@twk.de) ■ internet: [www.twk.de](http://www.twk.de)

## 1. Allgemeines

Die Winkelcodierer der Modellreihe KBD/ KRD sind Weiterentwicklungen im Hause TWK im Bereich der Feldbus-Komponenten für den Profibus-DP-V0.

Umfangreiche Erfahrungswerte resultierend aus der Modellreihe CRD wurden im Gerätedesign berücksichtigt. Neben der Erhöhung der Auflösung (max. 16 Bit) ist eine verkürzte Diagnose (max. 16 Bit) und optional ein Geschwindigkeitssignal erhältlich. Das Modell KBD ist ein Monotourcodierer (max. 16 Bit) und das Modell KRD ein Multitourcodierer (max. 28 Bit). Im Datenblatt 10941 sind alle relevanten technischen Parameter beschrieben.

Das Anwenderhandbuch beschäftigt sich im ersten Teil mit grundlegenden Voraussetzungen für das Verständnis des Einsatzes eines Winkelcodierers im PROFIBUS DP und im zweiten Teil wird eine Anleitung zum Einsatz unter der Siemens - Step7 - Software inklusive Beispielsoftware gegeben.

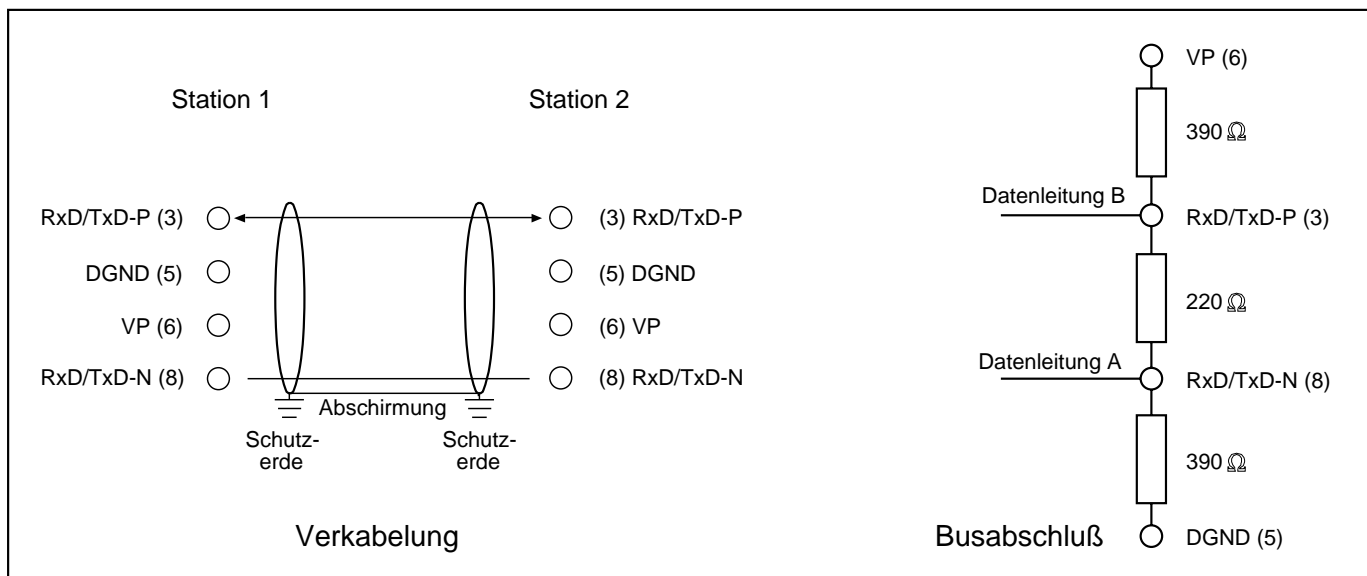
Für das allgemeine Verständnis des Feldbussystems PROFIBUS sowie für weiterführende Literatur möchten wir an dieser Stelle auf die PROFIBUS-Nutzerorganisation PNO ([www.profibus.com](http://www.profibus.com)) verweisen.

## 2. Installationshinweise für PROFIBUS-DP - RS 485

Grundlegende Eigenschaften der RS-485 Übertragungstechnik /2/:

- Netzwerk Topologie: Linearer Bus, Abschlußwiderstände für Busabschluß  
Stichleitungen sind nur bei Baudraten < 1,5 MBit/s zulässig
- Leitung Abgeschirmtes, verdrehtes Kabel
- Stationsanzahl 32 Stationen in jedem Segment ohne Repeater  
Mit Repeatern erweiterbar bis 126.
- Steckverbinder Realisierte Varianten bei der Modellreihe KBD/ KRD:
  - Rundstecker RS 25, 12-polig
  - Anschlußhaube
  - D-SUB 9-polig bzw. andere auf Anfrage  
(Anschlußbelegung nach /1/)

Verkabelung und Busabschluß für PROFIBUS-DP /2/, (Beachte: 9-poliger Sub-D-Stecker)



### Übertragungslänge in Abhängigkeit der Übertragungsgeschwindigkeit für Kabeltyp A

Baudrate [kBit/s]	9,6	19,2	93,75	187,5	500	1.500	12.000
Übertragungslänge [m]	1200	1200	1200	1000	400	200	100

Spezifikation Kabeltyp A:

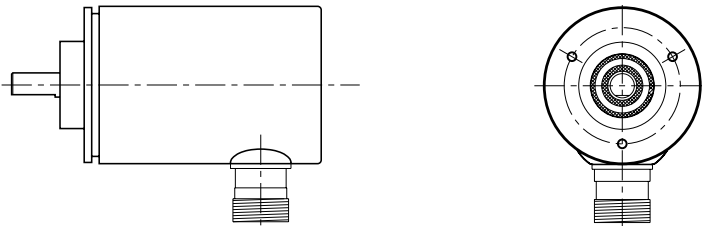
- Wellenwiderstand: 135...165  $\Omega$
- Kapazitätsbelag: < 30 pF/m
- Schleifenwiderstand: 110  $\Omega$ /km
- Aderdurchmesser: 0,64 mm
- Aderquerschnitt: > 0,34 mm<sup>2</sup>

siehe auch: Installation Guideline for PROFIBUS -DP/FMS (Bestell-Nr. 2.111/2 - PNO)  
Implementation Guide to DIN 19245 (Bestell-Nr. 2.001/2 - PNO)

### 2.1 Anschluß Winkelcodierer mit RS-Stecker

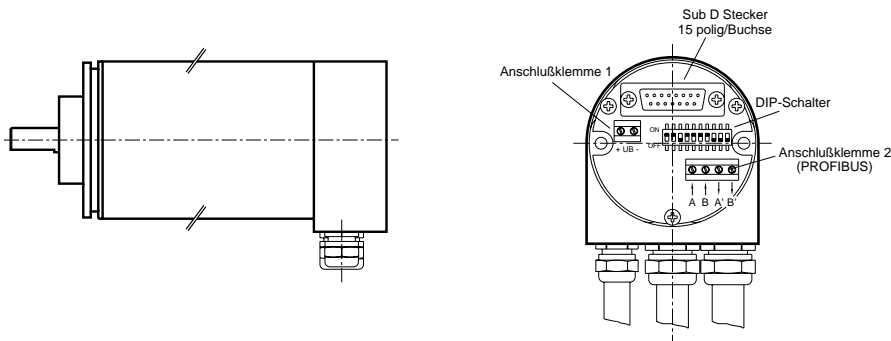
Die Anschlußbelegung für den RS-Stecker 12-polig (Beachte: Nummerierung der Pins im Uhrzeigersinn (Ansicht auf Kontaktseite der Buchse), Winkelcodierer: Buchse) ist konform zur Profildefinition für Encoder /1/.

Die Abschlußwiderstände sind im Gegenstecker bzw. in der Nachfolgeelektronik zu realisieren. Die Adresseinstellung ist im Kapitel 7.2 beschrieben.



Jeder Winkelcodierer mit RS-Stecker hat im Auslieferungszustand die Defaultadresse 123. Über den DP-Master ist es möglich, die Adresse eines DP-Slaves zu ändern. Die neu zu vergebende Slave-Adresse muß im Bereich von 1-126 liegen (DDL\_M\_Set\_Slave\_Add).

### 2.2 Anschluß Winkelcodierer mit Anschlußhaube



Die Anschlußhaube für 3-fach Anschlußtechnik ist ein T-Koppler, der im PROFIBUS installiert wird. Er hat drei PG-Anschlüsse, die wie folgt aufgeteilt sind:

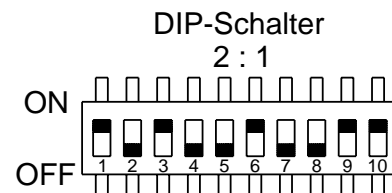
- PG 7: Spannungsversorgung für den Winkelcodierer (24 VDC)
- PG 9: Bus In (Signal-Daten A, B)
- PG 9: Bus Out (Signal-Daten A', B')

Der Winkelcodierer wird über den 15-poligen SUB-D-Stecker angeschlossen. Im Fehlerfall kann der Codierer ohne jeglichen Installationsaufwand ausgetauscht werden. Die Anschlußhaube wird durch Lösen von 2 Befestigungsschrauben vom Winkelcodierer getrennt. (Beachte: Nur im verschraubten Zustand der Anschlußhaube wird die jeweils angegebene Schutzart (IP 65/ IP 66 gewährleistet.)

Die Einstellung der Stations-/Teilnehmeradresse erfolgt über die DIP-Schalter in der Anschlußhaube. Der Adreßbereich liegt zwischen 1 und 126 (Defaultadresse: 123). Die Adresse ist nicht über den Dienst DDLM\_Set\_Slave\_Add änderbar. (Beachte: GSD-Datei entsprechend Winkelcodiererausführung - Stecker- bzw. Anschlußhaube). Die Einstellung der Abschlußwiderstände erfolgt über den 10-fachen DIP-Schalter (9,10) in der Anschlußhaube, die bei Bedarf als Leitungsabschluß zugeschaltet werden können.

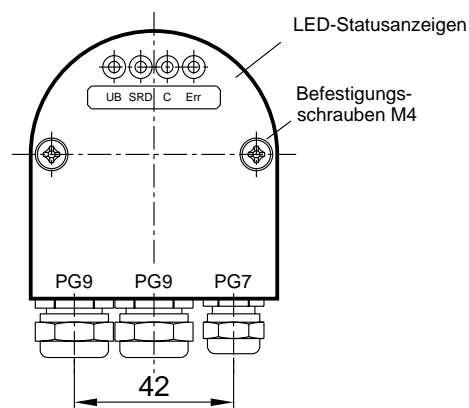
DIP-Schalter - Adresseinstellung/Abschlusswiderstände:

Schalter	1	2	3	4	5	6	7	8	9	10
ON = 1	LSB			MSB				n.c.	Abschlusswiderstände: ein	
OFF = 0	Adresse 1 - 126 einstellbar ( 123: Defaultadresse )								Abschlusswiderstände: aus	



In der Rückwand der Anschlußhaube befinden sich nachfolgende Status-LED's:

	UB	SRD	C	Err
Falsche Konfiguration	x	x		x
Unzulässige Parameter	x		x	x
Codierfehler (siehe Diagnosebytes 62 - 63)	x			x
Class 1 Gerätekonfiguration i.O.	x	x		
Class 2 Gerätekonfiguration i.O.	x	x	x	
UB-Betriebsspannung, SRD-Datentransfer, C-Class-Zugehörigkeit, Err-Fehlermeldung				



#### 3. Konfigurationsfunktion (DDL\_M\_Chk\_Cfg)

Die absoluten Winkelcodierer mit PROFIBUS-DP werden wie folgt klassifiziert:

##### Winkelcodierer mit Class 1 - Funktionalität

Class 1 - Geräte zeichnen sich dadurch aus, daß nur der Positionswert (16 Bit oder 32 Bit) des Winkelcodierers über den Bus übertragen wird. Es erfolgt keine Parametrierung von Winkelcodierer-Parametern. Dabei unterscheidet man die Konfigurationen D0 und D1. Die Konfiguration D0 beinhaltet das Datenformat: 1 Word Input-Daten, Konsistenz und D1 beinhaltet 2 Word Input-Daten, Konsistenz.

##### Winkelcodierer mit Class 2 - Funktionalität

Class 2 - Geräte zeichnen sich dadurch aus, daß sie über den Bus parametrierbar sind. Man unterscheidet die Konfigurationen F0 F1 und F3. Die Konfiguration F0 hat das Datenformat 1 Word Input-Daten, 1 Word Output-Daten, Konsistenz und F1 beinhaltet 2 Word Input-Daten, 2 Word Output-Daten, Konsistenz. Im Rahmen der Weiterentwicklung wurde optional ein Geschwindigkeitssignal, Konfiguration F3 - 4 Word Input-Daten und 4 Word Output-Daten realisiert. Die Klasse 2-Funktionalität bleibt bei der Konfiguration F3 natürlich im vollem Umfang erhalten.

Configuration function (DDL_M_Chk_Cfg)					
Selection	Class	Data	Identifier byte	Comment	Assignment Octet-No. and MSB/LSB
32 Bit In/ Out and 32 Bit Velocity	2	64 Bit In/Output data	F3	KRD, Velocity signal optional	Octet 1/Bit 63: MSB Octet 4/Bit 32: LSB Position value Octet 5/Bit 31: MSB Octet 8/Bit 0: LSB Velocity signal
Class 2 32 Bit In/ Out	2	32 Bit In/ Output data	F1	KRD	Octet 1/Bit 7: MSB Octet 4/Bit 0: LSB
Class 2 16 Bit In/ Out	2	16 Bit In/ Output data	F0	KRD/ KBD	Octet 1/Bit 7: MSB Octet 2/Bit 0: LSB
Class 1 32 Bit In	1	32 Bit Input data	D1	KRD	Octet 1/Bit 7: MSB Octet 4/Bit 0: LSB
Class 1 16 Bit In	1	16 Bit Input data	D0	KRD/ KBD	Octet 1/Bit 7: MSB Octet 2/Bit 0: LSB

### 4. Data Exchange Funktion (DDLML\_Data\_Exchange)

Input-Daten sind Daten, die von den Peripherie-Geräten an den Master bzw. in den Bus geschickt werden. Output-Daten sind Daten, die vom Master an die Slave-Teilnehmer gesendet werden. Als Beispiel für Output-Daten ist an dieser Stelle die Steuerung des Referenzwertes (siehe unten) angeführt.

#### 4.1 Positions-Istwert (Input-Data)

Die Ausgabe des Positions-Istwertes erfolgt im 16-, 32- oder 64-Bit Datenformat (Input-Daten), siehe auch Kennung des Winkelcodierers.

##### Positions-Istwert (DDLML\_Data\_Exchange) 16-Bit Datenformat (Kennung F0/D0)

Input-Data		
Octet	1	2
Bit	(MSB) 15 - 8	7 - 0 (LSB)
Data	$2^{15} - 2^8$	$2^7 - 2^0$
	Positionswert	

##### Positions-Istwert (DDLML\_Data\_Exchange) 32-Bit Datenformat (Kennung F1/D1)

Input-Data				
Octet	1	2	3	4
Bit	(MSB) 31 - 24	23 - 16	15 - 8	7 - 0 (LSB)
Data	$2^{31} - 2^{24}$	$2^{23} - 2^{16}$	$2^{15} - 2^8$	$2^7 - 2^0$
	Positionswert			

##### Positions-Istwert (DDLML\_Data\_Exchange) 64-Bit Datenformat (Kennung F3)

Input-Data								
Octet	1	2	3	4	5	6	7	8
Bit	63 - 56	55 - 48	47 - 40	39 - 32	31 - 24	23 - 16	15 - 8	7 - 0
Data	$2^{63} - 2^{56}$	$2^{55} - 2^{48}$	$2^{47} - 2^{40}$	$2^{39} - 2^{32}$	$2^{31} - 2^{24}$	$2^{23} - 2^{16}$	$2^{15} - 2^8$	$2^7 - 2^0$
	MSB			LSB	MSB			LSB
	Positionswert				Geschwindigkeitssignal			

#### 4.2 Referenzwert setzen (Output-Data)

Die Funktion Referenzwert setzen sollte nur bei Stillstand der Winkelcodiererwelle ausgeführt werden! Zum Abgleichen von Maschinenpositionswerten und der absoluten Position des Winkelcodierers ist es in einigen Fällen unumgänglich, den Referenzwert zu setzen. Der Referenzwert ist der Positionswert, der im Referenzpunkt zur Anzeige gebracht wird. Die Möglichkeit den Referenzwert zu setzen, bietet der TWK- Winkelcodierer mit Class 2 - Funktionalität. Durch den Anwender ist zu beachten, daß der Referenzwert innerhalb des Wertebereiches (0 bis Gesamtschrittzahl - 1) liegen muß. Das ist insbesondere bei der Änderung der Gesamtschrittzahl zu berücksichtigen. Der Referenzwert (Binärcode) wird im Data-Exchange Modus durch Setzen des Bits 31 (32-Bit Datenformat) bzw. Bit 15 (16-Bit Datenformat) oder Bit 63 (64-Bit-Datenformat) übertragen. Nachfolgende Darstellungen beziehen sich auf das 32-Bit Datenformat.

##### Referenzwert setzen (DDLML\_Data\_Exchange) z.B. 32-Bit Datenformat

Output-Data				
Octet	1	2	3	4
Bit	31	(MSB)30 - 24	23 - 16	15 - 8
Data	1/0	$2^{30} - 2^{24}$	$2^{23} - 2^{16}$	$2^{15} - 2^8$
	Preset Control	Referenzwert		

Beispiel: Setzen des Referenzwertes (Presetwert) z.B. 32-Bit Datenformat

Output-Data				
Octet	1	2	3	4
Bit	<b>31</b>	30 - 0		
Data	<b>1</b>	00.0000.0000.0000.0000.0000.0000.1000		
	<b>Preset Control</b>	Referenzwert: 8		

Nach Empfang dieser Message wird durch den Winkelcodierer ein Offsetwert (aus aktuellem Positions-Istwert und Referenzwert) berechnet.

Wenn der ausgegebene Positionswert gleich dem Referenzwert ist, kann durch den Master das Bit 31 zurückgesetzt werden, da der Preset Mode beendet ist. Die Timingdiagramme sind in einem gesonderten TY-Blatt spezifiziert. Nach dem Rücksetzen von Bit 31 = 0 arbeitet der Winkelcodierer im Normal operating mode. Der Offsetwert wird in den Diagnosedaten abgespeichert und kann bei Spannungsausfall und Neustart gelesen werden (Siehe auch Diagnosemeldungen [Kapitel 6](#)).



### 5. Betriebsparameter für Class 1/2 Winkelcodierer (DDL\_M\_Set\_Prm) /4/

Die Parametrierdaten setzen sich aus busspezifischen Daten und DP-Slave-spezifischen Daten zusammen.

Busspezifische Daten:                      Octet 1-7      Octet 1 - Stationsstatus  
    Octet 2 - WD\_Fact\_1  
    Octet 3 - WD\_Fact\_2  
    Octet 4 - Min. Station Delay Responder (min  $T_{SDR}$ )  
    Octet 5 - Ident\_Number ) 19  
    Octet 6 - Ident\_Number ) 63 hex  
    Octet 7 - Group\_Ident

DP-Slave-spezifischen Daten:              Octet 8-9      Class 1 Winkelcodierer ( 2 Byte User\_Prm\_Data)  
    Octet 8-29     Class 2 Winkelcodierer (22 Byte User\_Prm\_Data)

<b>Betriebsparameter (DDL_M_Set_Prm)</b>				
Octet Nummer	Parameter	Daten	Geräte-klasse	Bemerkung
9 / Bit 0	Code sequence/ Codeverlauf	0: CW: Increasing clockwise	1,2	
		1: CCW: Increasing counter clockwise		
9 / Bit 1	Class 2 functionality/ Klasse 2 Funktionalität	0: not supported	1,2	
		1: supported		
9 / Bit 2	Commissioning diagnostic control / Diagnose	0: not supported	optional	
9 / Bit 4	Scaling function status/ Freigabe der Skalierung für Auflösung und Gesamtschrittzahl	0: disabled	2	
		1: enabled		
9 / Bit 6	Velocity unit/ Einheit des Geschwindigkeitssignales	0: Steps/ 10ms	2	optional
		1: Steps/ 100ms		
9 / Bit 7	Short diagnostic: 16 Byte/ Verkürzte Diagnose: 16 Byte	0: not supported	2	
		1: supported		
10(MSB) - 13(LSB)	Singleturn resolution/ Auflösung (S/U)	1 to 65536 steps/revolution	2	
14(MSB) - 17(LSB)	Total measuring steps/ Gesamtschrittzahl	KRD: 1 to 268.435.456 steps KBD: 1 to 65536 steps	2	
18 - 29				

**Hinweis:** Zu beachten ist, daß intern im Codierer die Berechnung der Anzahl der Umdrehungen in  $2^n$  - Potenzen erfolgt. Unabhängig von dieser Forderung kann der Anwender die gewünschte Gesamtschrittzahl sowie die gewünschte Auflösung entsprechend der Applikation programmieren. Der Winkelcodierer greift bei der Berechnung bei Bedarf auf die nächst höhere  $2^n$  - Potenz zu. Dabei werden die Werte als tatsächliche Auflösung bzw. als tatsächliche Gesamtschrittzahl bezeichnet und als Ausgabewert angezeigt.

Beispiel: gewünschte Gesamtschrittzahl                      :      20480  
 gewünschte Auflösung     :      4096  
 gewünschte Anzahl von Umdrehungen                         :      5

interne Berechnung Winkelcodierer  
 tatsächliche Gesamtschrittzahl                                 :      32768  
 tatsächliche Auflösung     :      4096

berechnete Anzahl von Umdrehungen                             :      8

(Beachte: Der o. g. Hinweis ist bei nicht reversiblen Betrieb zu berücksichtigen. Bei dem aufgeführten Beispiel ist die Position 0 erst nach 32767 Schritten erreicht und nicht wie gewünscht nach 20479 Schritten.)

### 5.1 Definition der Programmierparameter

Operationsparameter (Festlegung des Betriebszustandes des Winkelcodierers)

- **Codeverlauf:** Der Codeverlauf definiert in welcher Drehrichtung der Positionswert steigenden Werten entspricht (Blickrichtung auf die Welle).
  - CW - clockwise     CCW - counter clockwise
  
- **Klasse 2:** Dieser Operationsparameter dient zur Unterscheidung von Encodern mit Class 1 oder Class 2 - Funktionalität.
  - Class 1: Codeverlauf, Freigabe der Class 2 - Funktionalität
  - Class 2: beinhaltet Class 1 - Funktionalität  
Skalierungsfunktion (siehe unten), verkürzte Diagnose, Geschwindigkeitssignal
  
- **Diagnose - routine:** Die Diagnoseroutine erlaubt eine extensive Überprüfung aller Encoderkomponenten auf fehlerfreie Funktionstüchtigkeit. Wenn Fehler durch die Diagnoseroutine festgestellt werden, kommen diese mit dem Alarmbit zur Anzeige. Die Routine wird bei jedem Einschaltvorgang durchlaufen und muss hier nicht separat eingeschaltet werden.  
Eine umfangreiche Diagnose erfolgt in den Diagnosebytes 62-63 (siehe herstellerspezifische Diagnose).
  
- **Skalierungsfunktion:** Die Skalierungsfunktion gibt die Parametrierung der Auflösung und der Gesamtschrittzahl frei. Diese Funktion wird nur bei Änderung der Parameter Auflösung und Gesamtschrittzahl wirksam. Nach Freigabe der Skalierungsfunktion wird der Positionswert neu berechnet und ausgegeben.
  
- **Geschwindigkeitssignal:** Das Geschwindigkeitssignal wird im Kapitel 5.2 ausführlich behandelt.
  
- **Verkürzte Diagnose:** Bedingt durch Marktanforderungen, nicht jede SPS unterstützt 63 Diagnosebytes, wurde mit der verkürzten Diagnose (16 Bytes) diesem Sachverhalt Rechnung getragen.

### 5.2 Hinweise zum Geschwindigkeitssignal

Allgemeine Angaben zur Konfiguration F3:

- Positionsdaten und Geschwindigkeit
- 64 Bit In/Out - Daten
- Class 2 Funktionalität wird im vollen Umfang unterstützt (siehe auch Konfiguration F1)
- Aktualisierungsrate des Positionswertes ca. 700 us
- Übernahme des Referenzwertes dauert ca. 3 s
- Der Geschwindigkeitswert wird nur dann ausgegeben wenn ein gültiger Parametersatz programmiert wurde.

Die Berechnung des Geschwindigkeitssignal erfolgt unabhängig von den programmierten Parametern für Auflösung und Gesamtschrittzahl! Zur Berechnung werden die tatsächlichen physikalischen Istwert-Positionen zum Zeitpunkt t1 und t1+x verwendet. Codiererintern wird je nach Einstellung des Parameters "Velocity Unit" mit zwei verschiedenen Zeitbasen gearbeitet.

Velocity Unit	Zeitbasis
Steps/10ms	1ms
Steps/100ms	10ms

Das ausgegebene Geschwindigkeitssignal ist das arithmetische Mittel über 16 Meßwerte. Das berechnete Geschwindigkeitssignal wird bei der Zeitbasis T = 1ms alle 16ms und bei der Zeitbasis T = 10 ms alle 160 ms ausgegeben.

Berechnung der Drehzahl:

Einstellung: Velocity Unit = Steps/100 ms

Drehzahl [min<sup>-1</sup>] = Angezeigter Wert x 10 x 60 / 65536

Drehzahl	Angezeigter Wert [hex]
4000 min-1	6AAAA
3000 min-1	50000
2000 min-1	35555
1000 min-1	1AAAA
500 min-1	D555
100 min-1	2AAA
10 min-1	444

Einstellung: Velocity Unit = Steps/10 ms

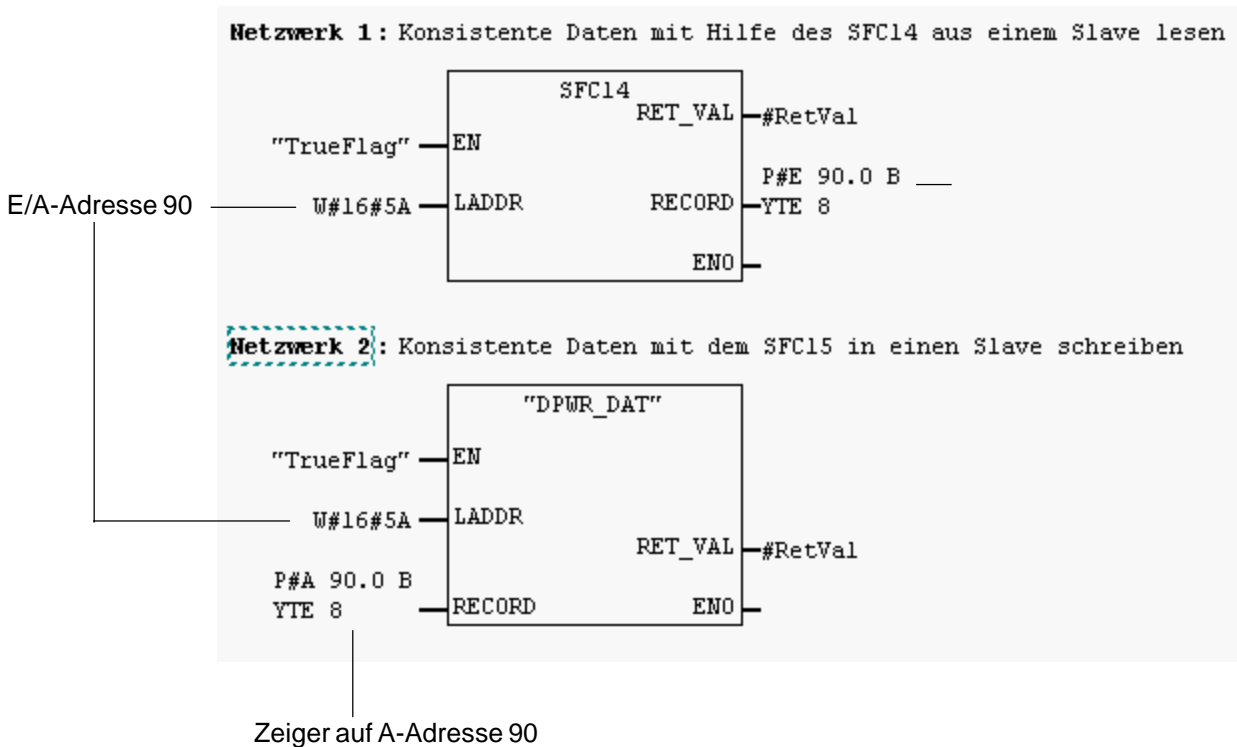
Drehzahl [min<sup>-1</sup>] = Angezeigter Wert x 100 x 60 / 65536

Drehzahl	Angezeigter Wert [hex]
4000 min-1	AAAA
3000 min-1	8000
2000 min-1	5555
1000 min-1	2AAA
500 min-1	1555
100 min-1	444
10 min-1	6D

Overflow = 8888 8888 (32-Bit-Wert) bei Drehzahlen  $\geq 4000 \text{ min}^{-1}$

Zugriff auf die E/A-Daten des Codierers

Da es in Steuerungen wie der S7 von Siemens keine E-/A-Datentypen mit 8 Byte Länge gibt, müssen bei Konfiguration F3 (8 Byte Eingangsdaten und 8 Byte Ausgangsdaten) die Daten mit den Systemfunktionen SFC14 und SFC15 abgeholt werden. Sie sind in den Steuerungen vorhanden und können in den OB's, FB's oder FC's aufgerufen werden. Als Parameter müssen Sie neben einem Enable-Merker (hier: TrueFlag) und einer Variablen für den Returnwert (hier: RetVal), die in der Hardwarekonfiguration vergebene E/A-Adresse übergeben (hier: 90<sub>dez</sub> bzw. 5A<sub>hex</sub>). Die Eingangsdaten stehen dann am Bausteingang "RECORD" des SFC14 zur Verfügung, die Ausgangsdaten müssen am Bausteineingang "RECORD" des SFC15 übergeben werden. Der Parameter RECORD ist vom Typ Any-Pointer und muß ein Zeiger auf ein 8Byte großes Array sein. Gibt man hier wiederum einen Zeiger auf die vergebene E/A-Adresse an, so stehen die Daten anschließend im E/A-Peripheriebereich unter dieser Adresse zur Verfügung. Sie können dann auch in der Variablen-tabelle an diesen Adressen beobachtet werden.



**6. Diagnosemeldungen (DDLML\_Slave\_Diag)**
**6.1 Standarddiagnoseinformation (Octet 1-6):**

Detailbeschreibung siehe DIN 19245 /4/

(Hinweis: Octet 5,6: Herstellerkennung: 1963 hex)

Die Herstellerkennung ist hinterlegt bei der PNO und identifiziert den Teilnehmer als TWK-Winkelcodierer der Modellreihe KBD/ KRD.

**6.2 Gerätebezogene Diagnose**

In dem Bereich Octet 7 bis max. 244 (lt. Standard /4/) kann der DP-Slave seine spezifische Diagnose ablegen.

<b>Diagnoseinformationen (DDLML_Slave_Diag)</b>			
<b>Diagnose Octet Nummer</b>	<b>Parameter</b>	<b>Daten</b>	<b>Geräte - Klasse</b>
8	Alarmmeldungen	4: Speicherfehler	1,2
9	Betriebsart	0: Codeverlauf	1,2
		1: Klasse 2 Funktion	
		2: Diagnoseroutine	
		3: Skalierungsfunktion	
		6: Einheit des Geschwindigkeitssignals	
		7: Kurze Diagnose: 16 Byte	
10	Encodertyp	01 hex: Absoluter Multitourcodierer	1,2
11(MSB) - 14 (LSB)	Auflösung	1 to 65536 (S/U)	1, 2
15 (MSB) - 16 (LSB)	Meßbereich	1 - 4096 Umdrehungen	1, 2
<b>Ende der Diagnosedaten für Klasse 1 Codierer und bei kurzer Diagnose!</b>			
17	Zusätzliche Alarmmeldungen		2
18 - 19	Unterstützte Alarmmeldungen	4: Speicherfehler	2
20 -21	Warnmeldungen		2
22 - 23	Unterstützte Warnmeldungen		2
24 - 25	Profilversion	z. B. 01.00	2
26 - 27	Softwareversion	z. B. 01.00	2
28 - 31	Betriebszeit	FFFF FFFFhex	2
32 - 35	Offset-Wert	00FF 230Fhex	2
36 - 39	Hersteller-Offset-Wert	nicht implementiert	2
40(MSB) - 43(LSB)	Auflösung (S/U)	1 - 65.536 S/U	2
44(MSB) - 47(LSB)	Gesamtschrittzahl	1 - 268.435.456 Schritte	2
48 - 57	Seriennummer	2A2A2A2A2A2A2A2A2A hex	2
58 - 59	Reserviert	00 00 hex	2
60 - 63	Herstellerspezifische Diagnose	siehe unten	2

**6.2.1 Herstellerspezifische Diagnose (Octet 60-63)**

Herstellerspezifische Diagnose - Octet Nummer	Bit	Definiton	Bemerkung	Fehlerbeseitigung
60	0 - 7	reserviert		
61	0 - 7	reserviert		
62	0	ErrEE	EEPROM Fehler	Reset Codierer
	1	ErrMSA	MSA Fehler	Reset Codierer
	2	ErrXRAM	Fehler externes RAM	Reset Codierer
	3	ErrExp	Fehler Anschlusshaube	Reset Codierer
	4	IniFlg	Neuinitialisierung EEPROM	
	5 - 7	reserviert		
63	0	ErrCRCO	CRC0 Fehler	Neuprogrammierung und erneutes Hochfahren des Codierers
	1	ErrCRC1	CRC1 Fehler	Neuprogrammierung und erneutes Hochfahren des Codierers
	2	ErrPar	Falscher Wert für die Anzahl der Umdrehungen	Neuprogrammierung
	3	ErrSkal	Skalierungsfehler	Wird von der Kommunikation abgedeckt
	4	ErrMem	ROM-Code Fehler	Reset Codierer
	5	ErrInt	Interner Controllerfehler	Reset Codierer
	6	ErrPre	Fehler Referenzwert	Eingabe des Referenzwertes* innerhalb des Wertebereiches: 0 bis Gesamtschrittzahl-1
	7	ErrStat	Unbekannter Auftrag von der Kommunikation	Ordnungsgemäßer Auftrag von der Kommunikation

\* Bei Eingabe eines fehlerhaften Presetwertes muss vor dem Setzen eines korrekten Presetwertes das Preset-Control-Bit 31 (siehe Kapitel 4.2) auf Null gesetzt werden.

**6.2.2 Beispiel für eine Diagnosemeldung**

Diagnose in Hexadezimalformat																
Octet	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
00-15	00	0C	00	02	19	63	39	00	02	01	00	01	00	00	10	00
16-31	00	00	10	00	00	00	00	01	00	01	00	FF	FF	FF	FF	0B
32-47	27	EA	D4	00	00	00	00	00	01	00	00	10	00	00	00	2A
47-63	2A	2A	2A	2A	2A	2A	2A	2A	2A	00	00	00	00	00	00	

Diagnose Octet Nummer	Parameter	Daten		Bemerkung
1 - 6	Standard-Diagnose-Information	01	00 hex	
		02	0C hex	Ansprechüberwachung aktiviert Bit 2 fest auf 1
		03	00 hex	
		04	02 hex	Parametrierung durch Master mit Adresse 2
		05 - 06	1963 hex	Ident-Nummer KBD/ KRD
7	Erweitertes Headerbyte	39 hex		63 Diagnose Bytes
8	Alarmmeldungen	00 hex		Kein Alarm
9	Betriebszustand	0A hex		CW, Klasse 2, Diagnose: nein, Skalierung: ja, Einheit des Geschwindigkeitssignales: steps/ 10ms, keine verkürzte Diagnose
10	Encodertyp	01 hex		Absoluter Multiturn Codierer
11 - 14	Auflösung	10000 hex		65536 Schritte/Umdrehung
15 - 16	Messbereich	1000 hex		4096 Umdrehungen
17	Zusätzliche Alarmmeldungen	00 hex		kein Alarm
18-19	Unterstützte Alarmmeldungen	0010 hex		Speicherfehler wird unterstützt
20-21	Warnmeldungen	0000 hex		wird nicht unterstützt
22-23	Unterstützte Warnmeldungen	0000 hex		wird nicht unterstützt
24-25	Profil - Version	01.00		Hardwareversion: 1.00
26-27	Software - Version	01.00		Softwareversion 1.00
28-31	Betriebszeit	FFFFFFFF hex		
32-35	Offset-Value	0B27EAD4 hex		Offsetwert
36-39	Hersteller Offset Wert	00000000 hex		wird nicht unterstützt
40-43	Auflösung	00010000 hex		65536 S/U
44-47	Gesamtschrittzahl	10000000 hex		268.435.456 Schritte
48-57	Seriennummer	2A2A2A2A2A2A2A2A2A2A hex		
58-59	Reserviert	0000 hex		
60-63	Herstellerspezifische Diagnose	00000000 hex		kein Fehler

**7. Simatic Step7**

Dieses Kapitel erläutert die Vorgehensweise zum Einbinden des TWK-Winkelcodierers in den Profibus einer Siemens-S7-Steuerung, sowie den Aufbau und die Nutzung der Beispielprogramme für Step7. Grundlage der Dokumentation ist die Step 7 Version 5.0.

**7.1 Einbinden des TWK-Profibus-Winkelcodierers**

Voraussetzungen: Sie haben eine Hardwarekonfiguration gemäß Ihres Steuerungsaufbaus erstellt und ein Profibus-Subnetz installiert.

**7.1.1 Installation der GSD-Datei**

- Schließen sie alle Projekte in der Hardwarekonfiguration.
- Legen Sie die von TWK mitgelieferte Diskette in Ihr Laufwerk ein.
- Wählen Sie in der Hardwarekonfiguration unter **Extras, Neue GSD installieren.**
- Wählen Sie die Ihrem Winkelcodierer entsprechende GSD-Datei auf der Diskette aus:  
 Version mit Steckeranschluß: z.B.: KBDL16.GSD, KRDL28.GSD  
 Version mit Anschlußhaube: z.B.: KBDZ16.GSD, KRZ28.GSD
- Aktualisieren Sie den Step7-Hardwarekatalog über **Extras, Katalog aktualisieren.**

**7.1.2 Installation des TWK-Winkelcodierer Symbols**

Durch die Installation des TWK-Winkelcodierer Symbols wird Ihr Codierer in der Hardwarekonfiguration nicht als unbekannter Teilnehmer dargestellt sondern erhält das Aussehen Ihres Winkelcodierers. Für die Funktion des Winkelcodierers ist dies jedoch nicht von Bedeutung.

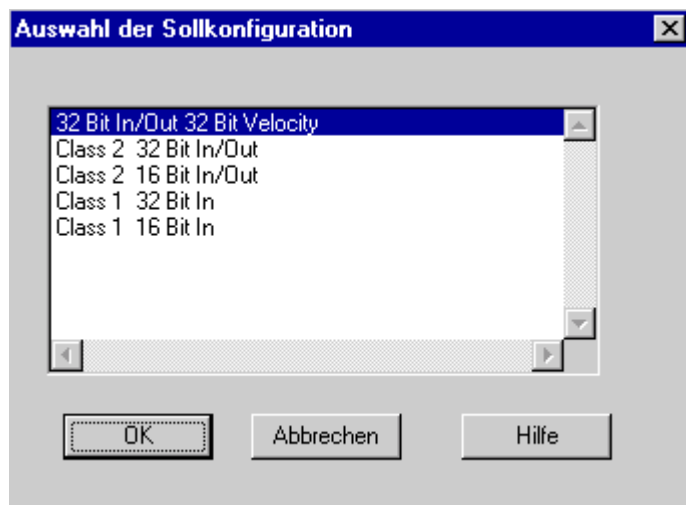
Zur Installation des Symbols (Bitmap) kopieren Sie die Datei: **KBD\_L\_an.BMP** oder **KBD\_Z\_an.BMP** (bzw. **KRD\_L\_an.BMP** oder **KRD\_Z\_an.BMP**) von A:\Bitmaps\ in das Verzeichnis **C:\Siemens\Step7\S7Data\NSBMP\** (falls C: ihr S7-Laufwerk ist). Die Symbole erscheinen erst nach einem Neustart des Simatic Managers.

**7.1.3 Auswahl des TWK-Winkelcodierers aus dem Step7 Hardwarekatalog**

- Nach dem Öffnen des Hardwarekataloges finden Sie unter **Profibus-DP, Weitere Feldgeräte, Encoder** die TWK-Profibus-Winkelcodierer z. B.:  
**TWK KBD/L - 16 Bit**  
**TWK KBD/Z - 16 Bit**  
**TWK KRD/L - 28 Bit**  
**TWK KRD/Z - 28 Bit.**
- Öffnen Sie nun Ihr Projekt, markieren Sie den Bus und binden Sie den Winkelcodierer durch einen Doppelklick auf die entsprechende Zeile im Hardwarekatalog (z.B. TWK KBD/L - 16 Bit) in den Bus ein.

**7.1.4 Konfigurieren des Winkelcodierers**

Nachdem der passende Codierertyp im Hardwarekatalog ausgewählt worden ist, erscheint das folgende Fenster zur Anwahl von Codierereffektivität und Auflösung. Wählen Sie hier entsprechend Ihren Anforderungen Class1 oder Class2 Funktionalität und 16-, 32, 64-Bit Auflösung . (Vergl. Kapitel 3)



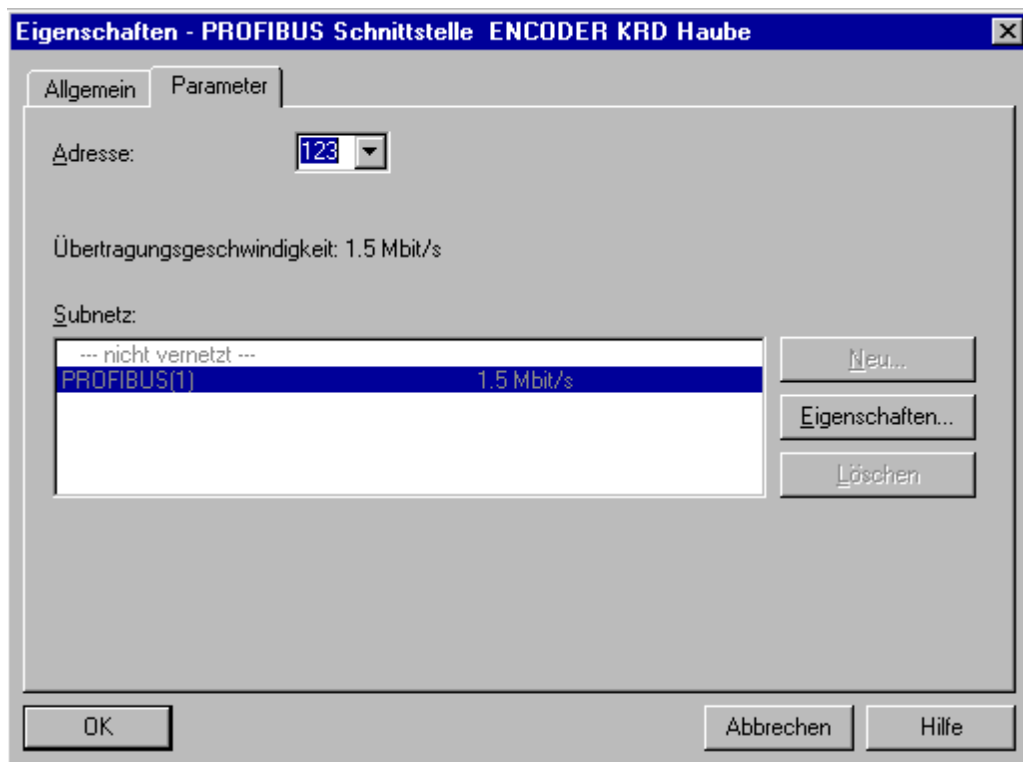


### 7.1.5 Profibus-Adresse vergeben

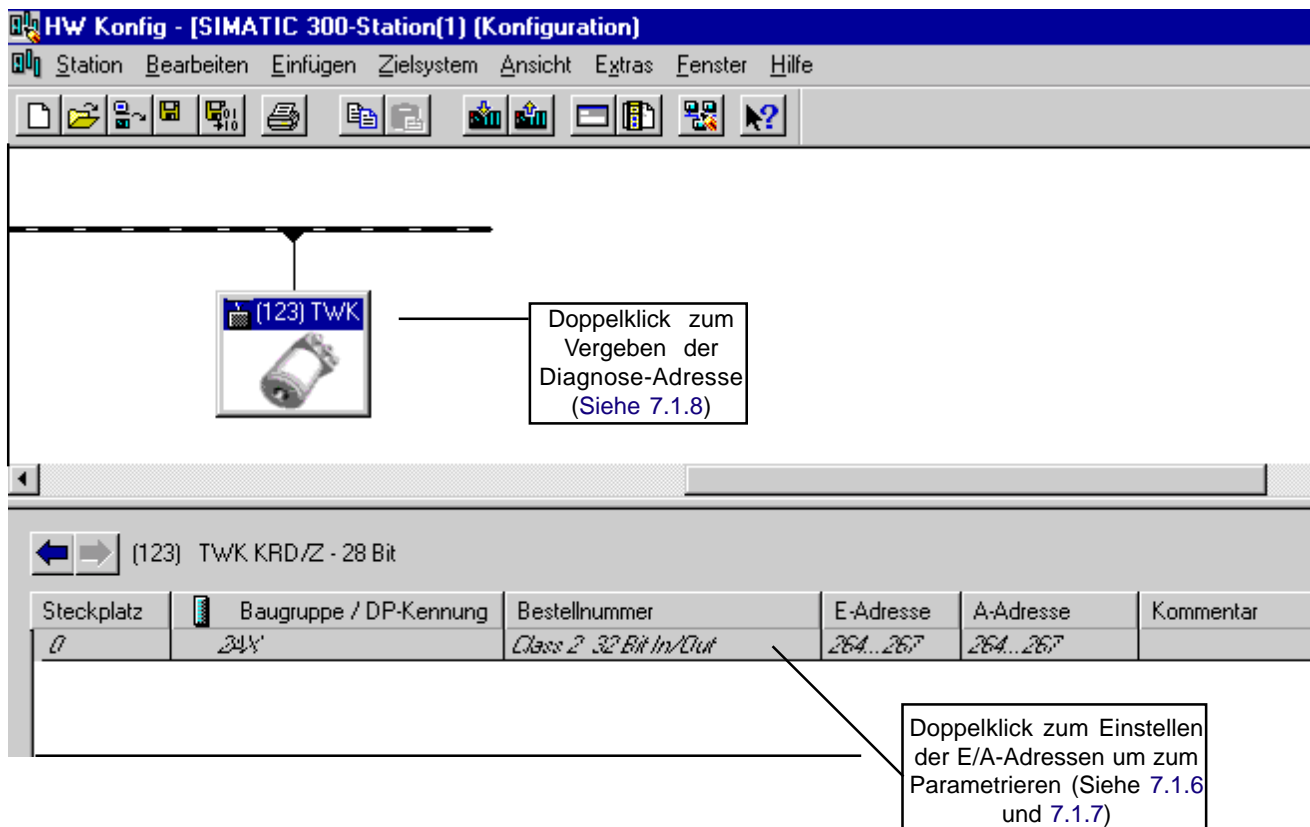
Haben Sie Ihre Sollkonfiguration ausgewählt erscheint das folgende Fenster, in dem Sie die am Codierer eingestellte Profibus-Adresse angeben müssen. Für die Beispielprogramme wählen Sie hier bitte Adresse 123 für den ersten und Adresse 122 für den zweiten Codierer.

**Hinweis:** Die Profibus-Adresse des Codierers wird bei der Haubenversion über Dip-Schalter (Siehe Kapitel 2.2) und bei der Steckversion per Software eingestellt (Siehe Kapitel 7.2).

Wählen Sie außerdem im Feld **Subnetz** Ihren projektierten Profibus an und verlassen Sie das Fenster mit **OK**.



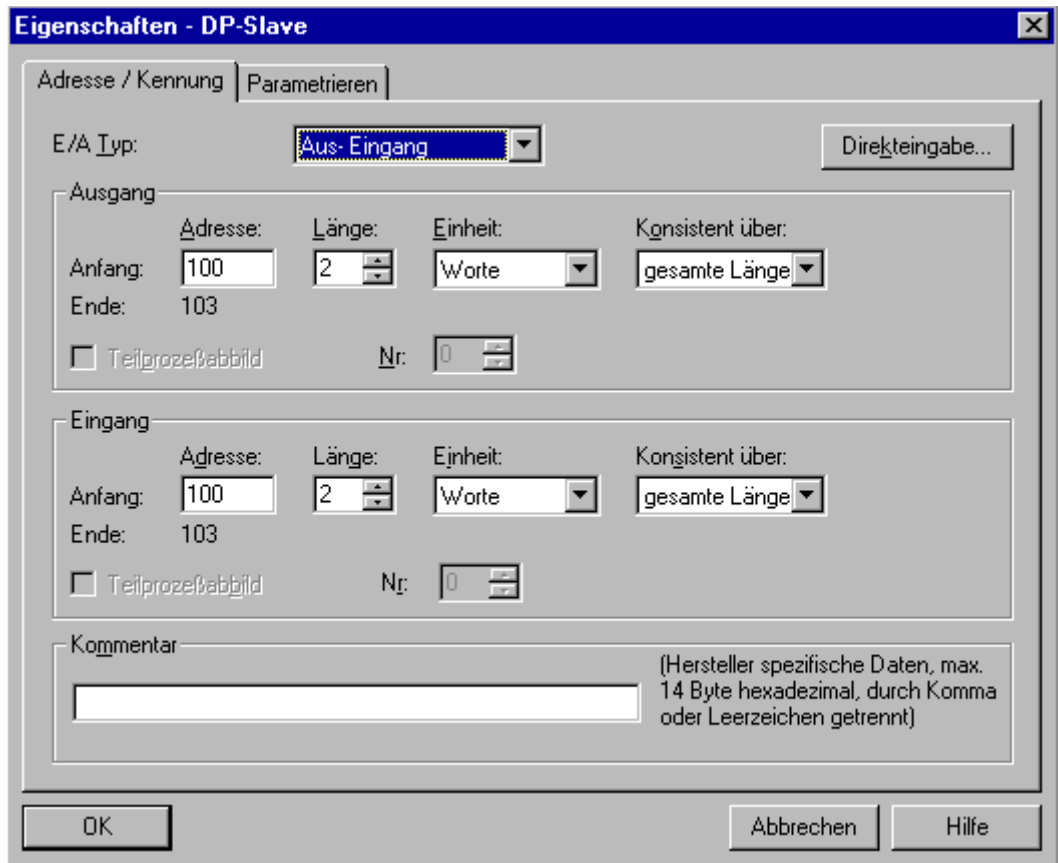
Anschließend erscheint der Codierer als Teilnehmer in Ihrem Profibus. Dies könnte je nach Konfiguration und Adresse so aussehen:



Der Wert für **Baugruppen / DP-Kennung** ergibt sich aus der ausgewählten Konfiguration. Die Werte für E/A-Adresse sind Defaultwerte die je nach Steuerung variieren.

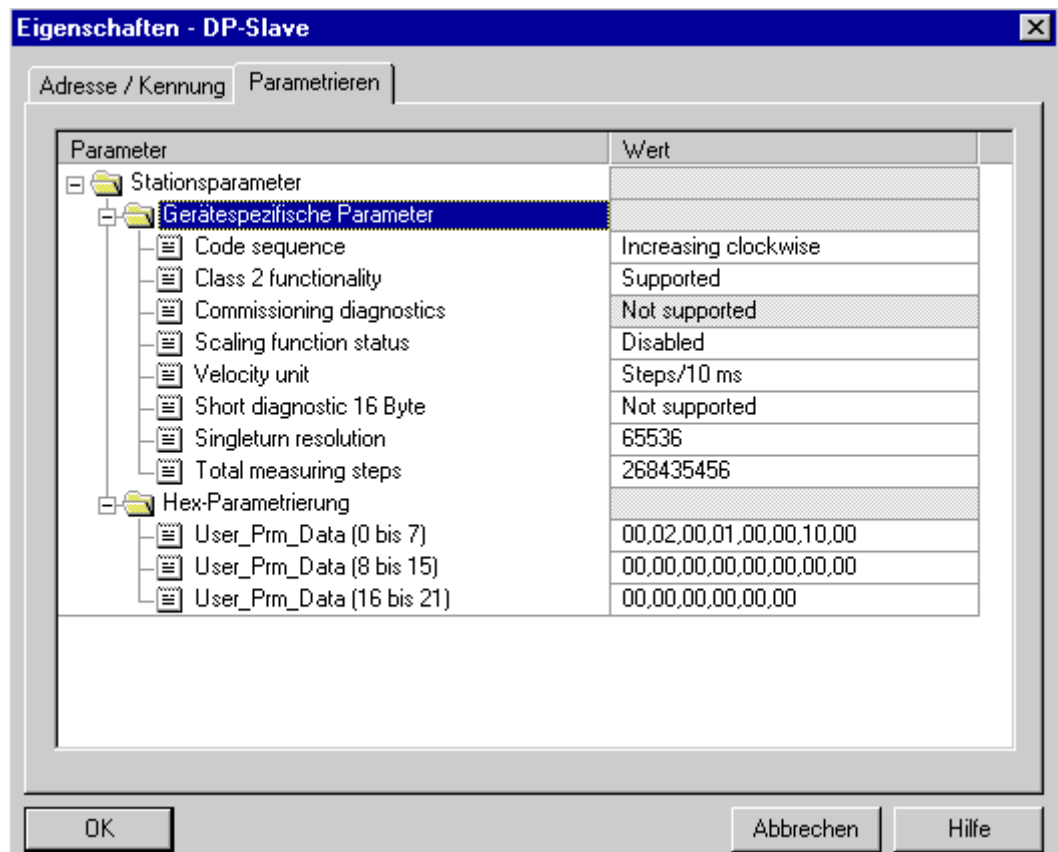
**7.1.6 Einstellen der E/A-Adressen (S7-Adressen)**

Ein Doppelklick auf die Zeile "Steckplatz 0" liefert das Fenster **Eigenschaften DP-Slave** mit den Registern **Adresse / Kennung** und **Parameterisieren**. Im Register **Adresse / Kennung** müssen unter Ausgang (nur bei Class 2 Codierer) und Eingang die Adressen für den Codierer vergeben werden, unter denen dieser in der S7 angesprochen werden soll. Die anderen Einträge dieses Registers sollten nicht verändert werden. Die folgende Abbildung zeigt dieses Register beispielsweise für einen Class 2 Codierer mit 32-Bit Auflösung. Für das Beispielprogramm geben Sie bitte für Ausgänge und Eingänge Adresse 100 ein.



**7.1.7 Parametrieren des Winkelcodierers**

Über das Register Parametrieren gelangt man in das folgende Fenster, in dem man die Eigenschaften des Winkelcodierers festlegen kann. Gezeigt werden die Parameter eines Class 2 - Slaves. Beim Class 1 - Slave ist hier nur der Parameter Code sequence einstellbar. (Siehe Kapitel 5)



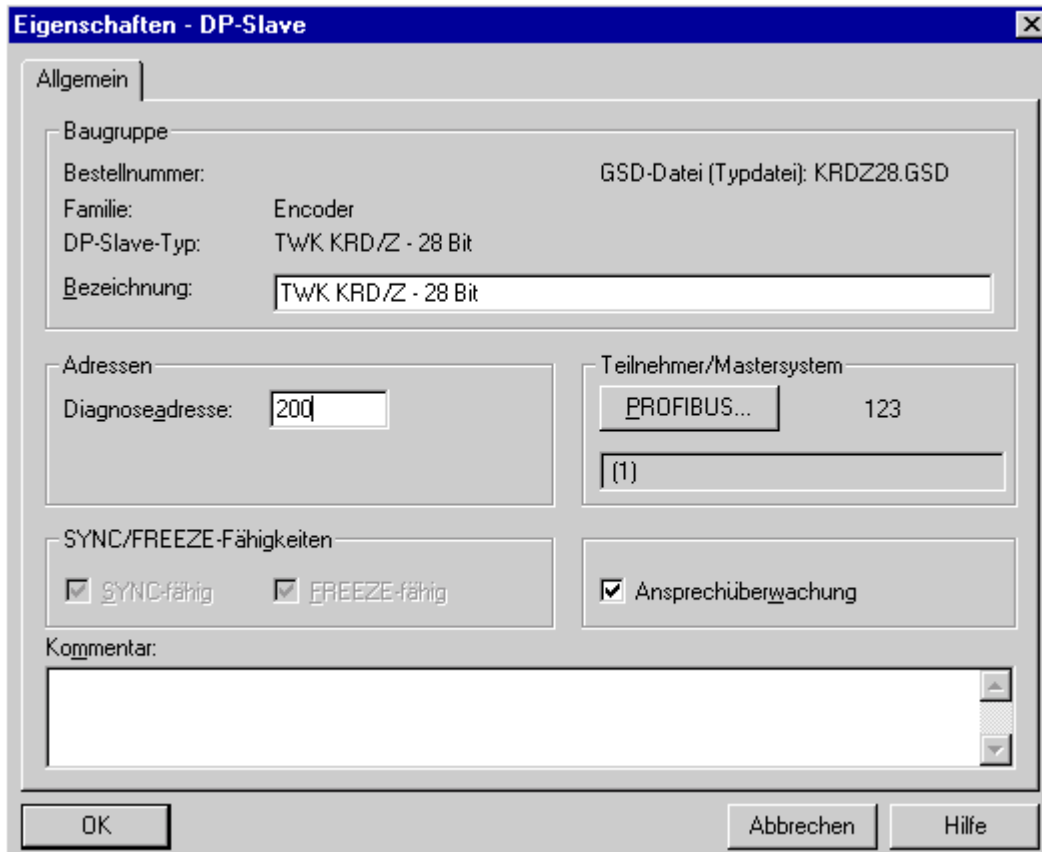
**7.1.8 Diagnose-Adresse einstellen**

Die Einstellung der Diagnoseadresse ist nur erforderlich wenn innerhalb des S7-Programms die speziellen Profibus-diagnosefunktionen genutzt werden.

Damit innerhalb des S7-Programms auf den Diagnosebereich des Winkelcodierers zugegriffen werden kann, ist es erforderlich diesem eine spezielle S7-Diagnoseadresse zu vergeben. Diese kann im gesamten Peripheriebereich der Steuerung liegen. Sie belegt also keine Ein-/Ausgangsadressen.

Durch einen Doppelklick auf das [Codierersymbol](#) erscheint das Fenster **Eigenschaften DP-Slave** mit dem Register **Allgemein**.

Für das Beispielprogramm geben Sie bitte die Diagnoseadresse 200 für den 1.Winkelcodierer und 202 für den 2.Winkelcodierer an.



Nach der Bestätigung mit OK ist der Winkelcodierer konfiguriert und parametrier.

Die Hardwarekonfiguration kann nun übersetzt und in die S7 übertragen werden.

**7.2 Einstellung der Teilnehmeradresse beim Winkelcodierer in Steckerversion**

Eine Einstellung der Adresse des KBD/ KRZ in Steckerausführung ist unter S7 mit einer Profibus-Protokoll-fähigen MPI-Karte von Siemens (z.B.: CP5611) möglich.

Die Voraussetzungen zur Vergabe einer neuen Adresse sind laut Siemens:

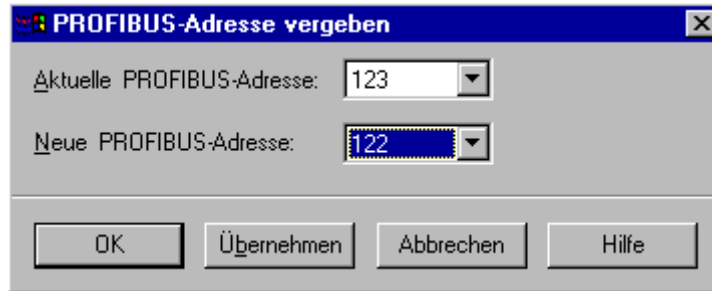
1. Es befindet sich kein DP-Master am PROFIBUS-Netz.
2. Im PROFIBUS-Netz existiert mindestens ein DP-Slave.
3. Sie haben bereits jedem DP-Slave des PROFIBUS-Netzes eine PROFIBUS-Adresse direkt zugewiesen.(entweder durch Software über Direktanschluß PG - DP-Slave oder durch Schaltereinstellung am DP-Slave; Zuweisung ist DP-Slave-abhängig, deshalb siehe Handbuch zum DP-Slave)
4. Sie haben das Programmiergerät über eine Stichleitung an das PROFIBUS-Netz angeschlossen (siehe Handbuch zum DP-Master).

(Siehe Online-Hilfe unter "Zielsystem/Profibus-Adresse vergeben" im Simatic Manager)

D.h. ist der Winkelcodierer nicht einziger Slave an der Profibus/MPI-Karte, so müssen alle im Bus befindlichen Slaves eine eindeutige (nur einmal vergebene) Adresse besitzen und der Master (S7-CPU) muss vom Bus getrennt sein. Es ist also beispielsweise nicht möglich zwei Winkelcodierer mit der Adresse 123 gleichzeitig am Bus anzuschließen und dann einem von beiden eine neue Adresse zuzuweisen.

Ist der Winkelcodierer, dessen Adresse geändert werden soll, vorher schon einmal im Busbetrieb gewesen, so muss dieser einmal spannungslos geschaltet werden, bevor er auf eine neue Adresse eingestellt werden kann.

Die Adresseinstellung wird im Simatic-Manager unter "**Zielsystem, Profibus-Adresse vergeben**" vorgenommen. Ist der Winkelcodierer mit der Profibus/MPI-Schnittstelle Ihres Programmiergerätes verbunden so erscheint der abgebildete Dialog.



Dieser zeigt die momentane Adresse des Codierers und fordert zur Eingabe der neuen Profibus-Adresse auf. Bestätigen Sie diese mit **Übernehmen**, so wird die neue Adresse nullspannungssicher im Codierer abgespeichert. Anschließend zeigt der Dialog die neue Adresse als aktuelle Adresse an.

**7.3 Beispielprogramme**

Auf der von TWK mitgelieferten Diskette befinden sich im Pfad \S7\_BSP\ mehrere S7-Archivdateien die von TWK erstellte S7-Beispielprogramme für das Arbeiten mit dem TWK-Profibus-Winkelcodierer enthalten. Die Programme wurden für eine CPU315-2DP entwickelt und so gehalten, daß keine weitere Peripherie, als ein TWK-Profibus-Winkelcodierer benötigt wird.

Auf der Diskette befindet sich je ein Projekt für Winkelcodierer mit Class 1 Funktionalität und Winkelcodierer mit Class 2 Funktionalität. Jedes Projekt enthält mehrere Programmordner für unterschiedliche Anwendungsfälle. Unterhalb der Programmordner befinden sich die standardmäßigen Ordner "Quellen" und "Bausteine".

Die TWK-Beispiele enthalten nur Bausteine die mit dem KOP/FUP/AWL-Editor erstellt worden sind. Die Erstsprache war FUP. Innerhalb der Bausteine wird umfangreiche Dokumentation anhand von Netzwerkkommentaren zur Verfügung gestellt.

**Für die Funktion dieser Programme auf Kundenanlagen/-steuerungen kann TWK keine Garantie übernehmen.**

Programme in den Archivdateien:

- TWKDPCL1.ARJ : Class 1 Projekt mit den Programmordnern Diagnose und Istwert, Kommentare in deutsch
- TWKDPCL2.ARJ : Class 2 Projekt mit den Programmordnern Diagnose, IstRef und Istwert, Kommentare in deutsch
- DP\_C1\_GB.ARJ : Class 1 Projekt mit den Programmordnern Diagnose und Istwert, Kommentare in englisch
- DP\_C2\_GB.ARJ : Class 2 Projekt mit den Programmordnern Diagnose, IstRef und Istwert, Kommentare in englisch

**7.3.1 Das Projekt TWKDPCL1**

Das folgende Bild zeigt die Programmordner des Class 1 Projektes:



**Programm Istwert** : Das Programm Istwert besteht nur aus einem OB1 und soll kurz zeigen, wie innerhalb des S7-Programms auf den Positionswert des Winkelcodierers zugegriffen wird.

**Programm Diag1**: Das Programm Diag1 beinhaltet zusätzlich zum Programm Istwert die Fehlerbehandlung für einen Class 1 Winkelcodierer. Es enthält u.a. den OB86 zur Erkennung eines Ausfalls des Winkelcodierers und den OB82 zur Erkennung einer Diagnoseanforderung des Winkelcodierers. Zum Auslesen des Diagnosebereichs wird die Step7 Systemfunktion SFC13 verwendet. (Der Diagnosebereich des Class 1 Winkelcodierers beträgt 16 Byte. Siehe Kapitel 6)

**Programm Diag2**: Das Programm Diag2 bietet die gleiche Funktionalität wie Diag1, ist jedoch für zwei Winkelcodierer ausgelegt.

Innerhalb eines jeden Programms kann zwischen Funktionen für einen 16- oder 32-Bit breiten Winkelcodierereingang gewählt werden. Hierzu wird einfach entweder die 16- oder die 32-Bit Funktion mit einem "1"-Signal (Eins-Merker M 0.1) am EN - Eingang versorgt und die nicht benötigte Funktion mit einem "0"-Signal (Null-Merker M 0.0) beschaltet.

**7.3.2 Das Projekt TWKDPCL2**

Das Class 2 Projekt enthält die folgenden Programmordner:



**Programm Istwert:** Identisch mit Class-1-Programm

**Programm IstRef:** Das Programm IstRef enthält das Auslesen des Positionswertes und zusätzlich das, bei der Class 2 Funktionalität mögliche, Setzen eines Referenzwertes.

**Programm Diag1:** Das Programm Diag1 enthält analog zu Diag1 aus dem Class 1 Projekt die Fehlerbehandlung eines Class 2 Codierers. Der Umfang des Diagnosebereichs ist hier 63 Byte.

**Programm Diag2:** Diag2 ist dann wieder die auf zwei Winkelcodierer erweiterte Variante von Diag1.

**7.3.3 Installation der Beispielprogramme**

**Vorraussetzungen:**

- Sie haben ein Projekt angelegt und darin eine Steuerung mit ihrer Hardwarekonfiguration eingefügt.

Dies könnte beispielsweise so aussehen:



- Sie haben in der Hardwarekonfiguration einen oder zwei Winkelcodierer mit den folgenden Einstellungen an ein Profibus-Subnetz angeschlossen (Siehe Kapitel 7.1).

Erster Winkelcodierer:                   **Profibusadresse 123**  
   **Ein-/Ausgänge: ab Adresse 100**  
   **Diagnoseadresse: 200**

Evtl.: Zweiter Winkelcodierer:       **Profibusadresse 122**  
   **Ein-/Ausgänge: ab Adresse 110**  
   **Diagnoseadresse: 202**

**Installation:**

- Wählen Sie im Simatic Manager **Datei, Dearchivieren**. Ändern Sie den Dateityp in \*.arj und wählen Sie auf der mitgelieferten Diskette im Verzeichnis \S7\_BSP\ das Class 1 oder Class 2 Projekt aus.
- Im nächsten Fenster geben Sie Ihr Projektverzeichnis an (normalerweise S7proj) .
- Durch Übernahme mit OK wird das Dearchivierungsprogramm gestartet. Nach dessen Beendigung finden Sie in Ihrem S7-Projektverzeichnis Ihr ausgewähltes TWK-Beispielprojekt.
- Wenn Sie nun **Datei, Öffnen , Projekt** anwählen, erhalten Sie eine Liste der auf Ihrem System vorhandenen Projekte. Sollte das Beispielprojekt hier noch nicht vorhanden sein, so wählen Sie **Durchsuchen** und suchen Sie die Datei TWKDPCL1.s7p (bzw. TWKDPCL2.s7p) unter dem TWK-Beispielprojekt.
- Öffnen Sie das Beispielprojekt, so daß Sie nun beide Projekte, Ihr eigenes und das Beispielprojekt, geöffnet haben. Dies könnte dann zum Beispiel wie in der Abbildung auf der nächsten Seite aussehen .
- Wählen Sie einen untergeordneten S7-Programmordner des TWK-Beispielprojektes aus. ( Hier entweder Diag1, Diag2, IstRef oder Istwert. Siehe auch: Kapitel 7.3.1)



- Kopieren Sie alle Bausteine des Behälters Bausteine aus dem ausgewählten Programmordner ( z.B. Diag1 aus TWKDPCL2 ) in den noch leeren Bausteinbehälter Ihres eigenen Projektes ( z.B. S7-Programm(1) aus DP\_CLAS2 ). (Anmerkung: Jeder Bausteinbehälter, auch ein leerer, enthält mindestens einen OB1, dieser ist natürlich ebenfalls leer und kann somit überschrieben werden.)
- Haben Sie einen Class 2, 16-Bit Winkelcodierer installiert und das Programm IstRef oder Diag1/2 ausgewählt, so müssen Sie zum Setzen des Referenzwertes, den FB10 im OB1 freischalten, d.h. den EN-Eingang mit M 0.1 versorgen und den (die) FB11 sperren, also den EN-Eingang mit M 0.0 beschalten.
- Ersetzen Sie eventuell den M 1.0 - Meldung quittieren und den M 10.0 (und M 10.1 bei zwei Winkelcodierern) - Setzen des Referenzwertes, durch Ihre Signale.
- Übertragen Sie alle Bausteine in die Steuerung.
- Rufen Sie nun den OB1 in der Online-Ansicht auf und schalten Sie **Test, Beobachten** ein um sich die aktuellen Werte des Winkelcodierers am Monitor anzeigen zu lassen.
- Nur für Programm IstRef und Diag1/2: Tragen Sie einen Referenzwert in DB100 Datendoppelwort 0 (für den 2.Winkelcodierer DB100 Datendoppelwort 8) ein und setzen Sie diesen mit dem M 10.0 (bzw. M 10.1). Liegt der Referenzwert außerhalb des parametrisierten Meßbereichs des Winkelcodierers, so wird die entsprechende Fehlermeldung im OB1 gesetzt.

### 7.3.4 Erläuterungen zu den Beispielprogrammen

In jedem Programmordner befindet sich eine Symboltabelle, die alle **globalen Variablen** des Maximalausbaus (Class 2 Projekt, Programm Diag2) enthält.

Im Folgenden wird die Programmstruktur dieses Maximalausbaus erläutert. Die Referenzdaten liefern die folgende Übersicht: (In den runden Klammern befindet sich immer der symbolische Name)

<pre> S7-Programm ├── OB1 &lt;maximal: 30&gt; │   ├── DB100(ReferenzData) [26] │   ├── FB10(SetRef16), DB10(SetRef16_IDB1) [28] │   ├── FB11(SetRef32), DB11(SetRef32_IDB1) [30] │   ├── FB11(SetRef32), DB21(SetRef32_IDB2) [30] │   ├── DB120(FaultDB) [26] │   └── DB121(EncoderFaultDB) [26] ├── OB82 &lt;maximal: 116&gt; │   ├── FB13(ReadDiag32), DB13(ReadDiag32_IDB1) │   │   └── SFC13 [116] │   └── DB121(EncoderFaultDB) [30] └── OB86 &lt;maximal: 32&gt;     ├── FC16(SearchSlave) [32]     └── FC16(SearchSlave) [32]         </pre>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <b>Zyklisches Programm:</b>                  - Einlesen des Istwertes                  - Setzen des Referenzwertes (nur Class2) für 16- Bit (FB10) bzw. 32-Bit (FB11)                  - Fehlermeldungen aus OB82                  - Fehlermeldungen aus OB 86             </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">                 Der OB82 wird durchlaufen, sobald ein DP-Slave eine Diagnoseanforderung stellt (bei kommendem und gehendem Ereignis). Im OB82 werden durch den FB13 bzw. SFC13 die Diagnosedaten abgeholt. Gemeldet werden die Fehler im OB1.             </div> <div style="border: 1px solid black; padding: 5px;">                 Der OB86 wird vom System u.a. bei Ausfall eines DP-Slaves (z.B. Stromausfall) aufgerufen (bei kommendem und gehendem Ereignis). Im FC16 wird der ausgefallene Codierer ermittelt und im OB1 zur Anzeige gebracht.             </div>
---	--

Durch die Systemfunktion SFC13 wird immer der gesamte Diagnosebereich des gestörten Winkelcodierers ausgelesen (16 Byte bei Class 1 und 63 Byte bei Class 2). Die Adresse dieses Slave liefert der OB82 in seinen Lokaldaten.

Es werden nur die herstellerepezifischen Fehlermeldebites ausgewertet und von diesen nur diejenigen, die im Data Exchange Mode (im normalen Busbetrieb) auftreten können. Fehler, die nur im Busanlauf auftreten, können durch den OB82 nicht erkannt werden. In diesem Fall müssen die Fehlermeldungen über die Funktion **Hardware diagnostizieren** des Step7 Paketes ausgelesen werden.

**8. Literatur**

- /1/ PROFIBUS Profile for Encoders, Order No. 3.062 1997,  
PROFIBUS Nutzerorganisation e. V.  
Haid-und-Neu-Str. 7  
D-76131 Karlsruhe
- /2/ PROFIBUS, Technische Kurzbeschreibung, Version: April 1997, PROFIBUS Nutzerorganisation
- /3/ DIN 19245 Teil 1, PROFIBUS, Process Field Bus, Übertragungstechnik, Buszugriffs- und Übertragungsprotokoll,  
Dienstschnittstelle zur Anwendungs-Schicht, Management
- /4/ DIN 19245 Teil 3, PROFIBUS, Process Field Bus, Dezentrale Peripherie (DP)
- /5/ SIEMENS, SINEC L2  
SPC 3 Siemens PROFIBUS Controller  
Anwenderbeschreibung  
Bestell-Nr.: 6ES7 195-0BD00-8AA0

**Anhang: Begriffe Winkelcodierer**

Parameter:	Erklärung
Auflösung - Schritte/360°:	Die Auflösung gibt die Anzahl der Schritte pro Umdrehung (360°) an.
Meßbereich:	Der Meßbereich gibt die maximale Anzahl der Umdrehungen an. Die Angabe der Umdrehungen muß in 2 <sup>n</sup> -Potenzen erfolgen.
Gesamtschrittzahl:	Die Gesamtschrittzahl ergibt sich wie folgt: Gesamtschrittzahl = Auflösung x Meßbereich
Codeverlauf:	Der Codeverlauf gibt an, in welcher Drehrichtung der Ausgabecode des Codierers steigenden Werten entspricht. In Abhängigkeit von der Drehrichtung wird unterschieden:  CW - clockwise, Drehrichtung im Uhrzeigersinn CCW - counter clockwise, Drehrichtung entgegen dem Uhrzeigersinn (Blickrichtung auf die Welle)
Referenzwert:	Der Referenzwert ist der Wert , der nach der Funktion Referenzwert setzen als Positionswert des Codierers anliegt.