

CAN 300 / CAN 400 Anwendungsbeispiel DS402 ERL

CAN Kommunikations-Baugruppe für S7-300/S7-400
Anwendungsbeispiel für CANopen DS402 mit ERL PCS Motoren

Handbuch

Ausgabe 2 / 06.11.2008



Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung dieses Handbuches, oder Teilen daraus, vorbehalten. Kein Teil des Handbuches darf ohne schriftliche Genehmigung der Systeme Helmholtz GmbH in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, oder unter Verwendung elektronischer Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden. Alle Rechte für den Fall der Patenterteilung oder Gebrauchsmustereintragung vorbehalten.

Copyright © 2008 by

Systeme Helmholtz GmbH

Hannberger Weg 2, 91091 Großenseebach

Hinweis:

Der Inhalt dieses Handbuches ist von uns auf die Übereinstimmung mit der beschriebenen Hard- und Software überprüft worden. Da dennoch Abweichungen nicht ausgeschlossen sind, können wir für die vollständige Übereinstimmung keine Gewährleistung übernehmen. Die Angaben in diesem Handbuch werden jedoch regelmäßig überprüft und notwendige Korrekturen sind in den nachfolgenden Ausgaben enthalten. Für Verbesserungsvorschläge sind wir Ihnen dankbar.

Inhaltsverzeichnis

1	Übersicht	5
1.1	Anwendung und Funktionsbeschreibung	5
1.2	Aufbau SPS	6
1.3	Aufbau ERL Motor	7
1.4	Aufbau der verwendeten PDOs (Mapping)	8
2	Projektierung der CAN Baugruppe	9
2.1	Einstellung der CAN-Bus Baudrate	9
2.2	Einstellung der Übertragungsart (Protokoll)	9
2.3	Akzeptanzmasken	10
2.4	Netzmanagement	10
2.5	Timer	11
3	Programmierung in der SPS	12
3.1	Übersicht	12
3.2	Init Drive (FB 37)	12
3.3	Homing (FB 30, FB 31)	13
3.4	Profile Position (FB 32, FB 33)	14
3.5	Profile Velocity (FB 34, FB 35)	15
3.6	Fault Reset (FB 36)	16
3.7	Beispiel FC 1 (ERL_Test)	16
3.8	Verwendung mit mehreren Antrieben	17
3.9	Inhalt des Statuswortes	18

1 Übersicht

CANopen

1.1 Anwendung und Funktionsbeschreibung

Dieses Handbuch beschreibt das Anwendungsbeispiel (Hantierungsbausteine) für die Ansteuerung eines ERL PCS Motors mit einer S7-300 unter Verwendung einer CAN 300 Baugruppe nach dem CANopen Profil DS402.

Das Anwendungsbeispiel liegt auch für die CAN 400 Baugruppe für die S7-400 vor. Die Funktionalitäten sind übertragbar.

Es ist als Ergänzung zum Handbuch „CAN 300“ bzw. „CAN 400“ zu verwenden. Die Informationen aus diesen Handbüchern werden als bekannt vorausgesetzt, insbesondere die Beschreibung des CANopen Protokolls und der CANopen Hantierung.



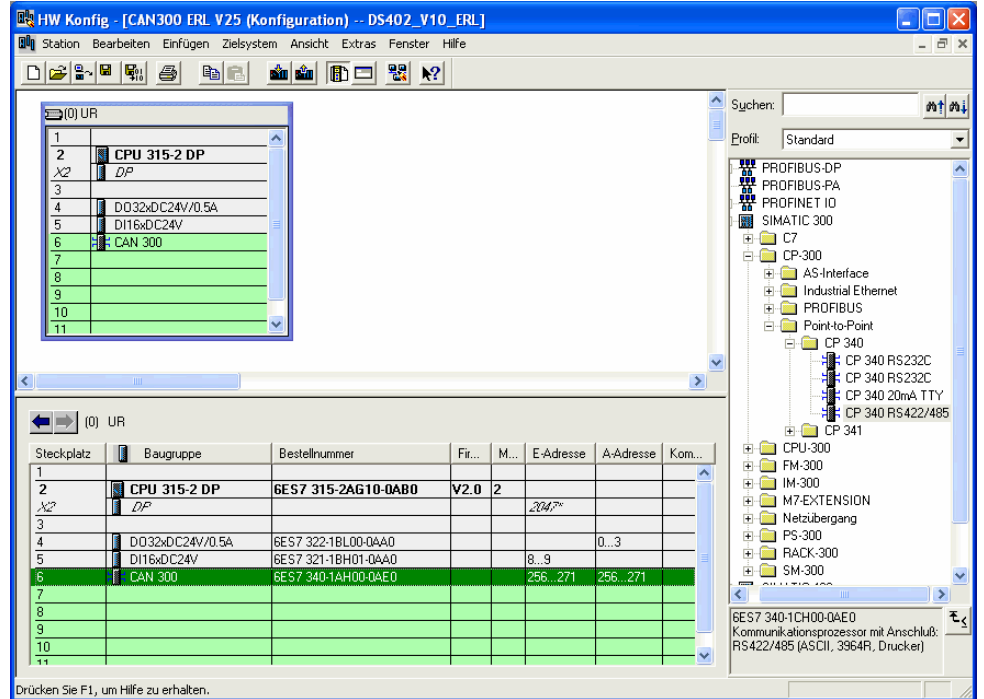
Ergänzend hierzu ist die CANopen Profilbeschreibung „DSP-402 Device Profile for Drives and Motion“, sowie die Beschreibung von ERL „PCS CANopen Handbuch“, insbesondere das Kapitel „Object Dictionary“ sehr hilfreich.

Die hier beschriebenen Hantierungsbaustein (FBs) stellen folgende Grundfunktionen zur Verfügung:

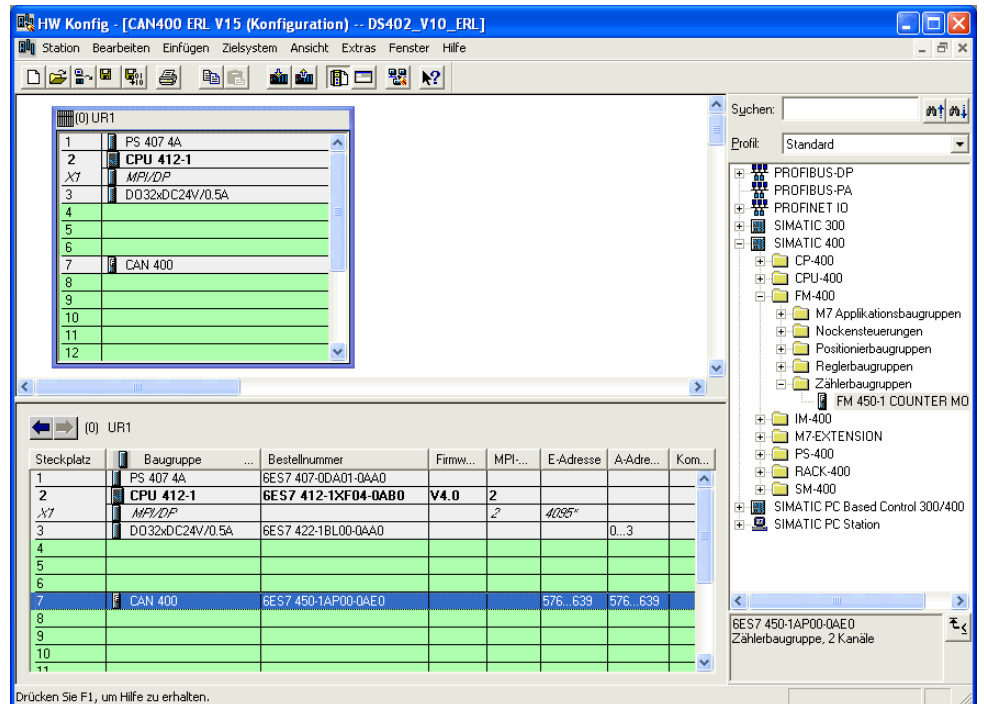
- Grundinitialisierung eines Antriebs
- Profile Velocity Mode
- Profile Position Mode
- Homing Mode
- Fault Reset

1.2 Aufbau SPS

Zur Verwendung kommt eine S7-300 CPU31x, eine 32Bit Digitale Ausgabe-, eine 16Bit Digitale Eingabebaugruppe sowie eine CAN 300 Baugruppe.



Für die CAN 400 sieht die Hardwarekonfiguration wie folgt aus:



1.3 Aufbau ERL Motor

Das Hantierungsbeispiel wurde mit einem ERL PCSI100P erstellt und getestet. Der Motor ist als Node 2 angeschlossen.

Zur Verwendung des Hantierungsbeispiels wird davon ausgegangen, daß das PCS betriebsbereit ist (Verkabelung, Tuning, Parameter des verwendeten Motor, CAN-Bus Baudrate, etc.).

Bitte lesen Sie die Dokumentation von ERL zum Aufbau und zur Inbetriebnahme des Motors genau durch.

1.4 Aufbau der verwendeten PDOs (Mapping)

Die Hantierungsbausteine gehen von dem Standard PDO-Mapping des ERL PCS Motors aus.

Das Mapping der PDOs wird über die Objekte 1600 ff. (RxPDOs) und 1A00 ff. (TxPDOs) durchgeführt.

TxPDO1 (wird vom Motor gesendet):	ID 181 _{hex}	
Byte 0-1:	SDO 6041/0	Status Word
TxPDO2 (wird vom Motor gesendet):	ID 281 _{hex}	
Byte 0+1:	SDO 6041/0	Status Word
Byte 2:	SDO 6061/0	Mode of Operation Display
TxPDO3 (wird vom Motor gesendet):	ID 381 _{hex}	
Byte 0+1:	SDO 6041/0	Status Word
Byte 2-5:	SDO 6064/0	Position actual value
TxPDO4 (wird vom Motor gesendet):	ID 481 _{hex}	
Byte 0+1:	SDO 6041/0	Status Word
Byte 2-5:	SDO 606C/0	Velocity actual value
RxPDO1 (wird vom Motor empfangen):	ID 201 _{hex}	
Byte 0+1:	SDO 6040/0	Control Word
RxPDO2 (wird vom Motor empfangen):	ID 301 _{hex}	
Byte 0+1:	SDO 6040/0	Control Word
Byte 2:	SDO 6060/0	Modes of Operation
RxPDO3 (wird vom Motor empfangen):	ID 401 _{hex}	
Byte 0+1:	SDO 6040/0	Control Word
Byte 2-5:	SDO 607A/0	Target Position
RxPDO4 (wird vom Motor empfangen):	ID 501 _{hex}	
Byte 0+1:	SDO 6040/0	Control Word
Byte 2-5:	SDO 60FF/0	Target Velocity

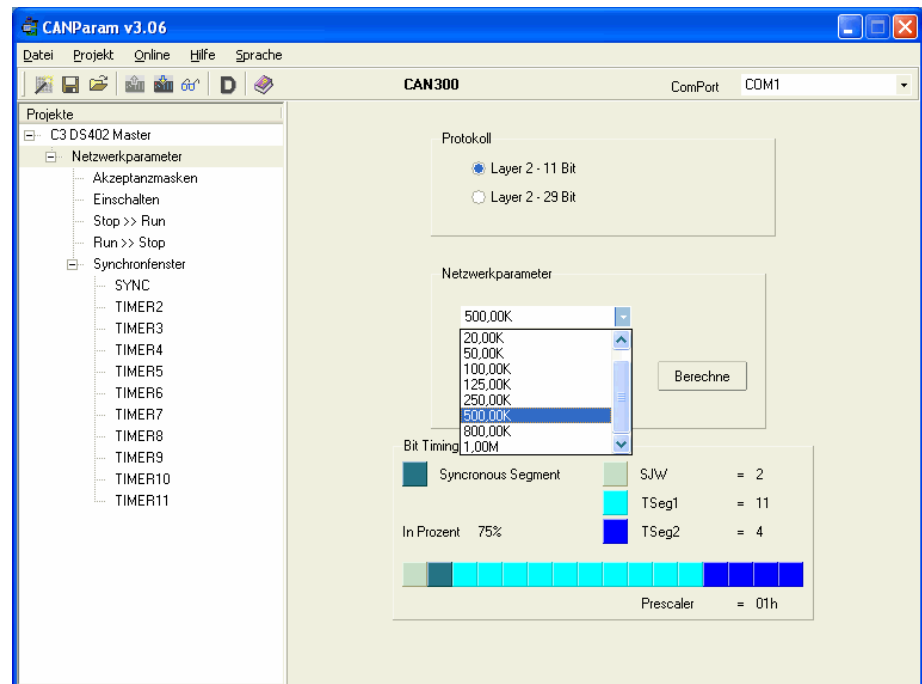
Das Senden der TxPDOs vom Antrieb wird nur durchgeführt wenn er im CANopen Modus „Operational“ ist.

2 Projektierung der CAN Baugruppe

Zur Verwendung des Beispielprogrammes muss das beiliegende CAN-Projekt „C3 DS402 Master.par“ in die CAN 300 Baugruppe eingespielt werden. Für die CAN 400 Baugruppe liegt das Projekt „C4 DS402 Master.par“ bei.

2.1 Einstellung der CAN-Bus Baudrate

Die CAN-Bus Baudrate muss passend zur Einstellung des ERL Motors gesetzt werden.



2.2 Einstellung der Übertragungsart (Protokoll)

Als Übertragungsart muss für alle CANopen Anwendungen immer „Layer 2 – 11Bit“ eingestellt sein.

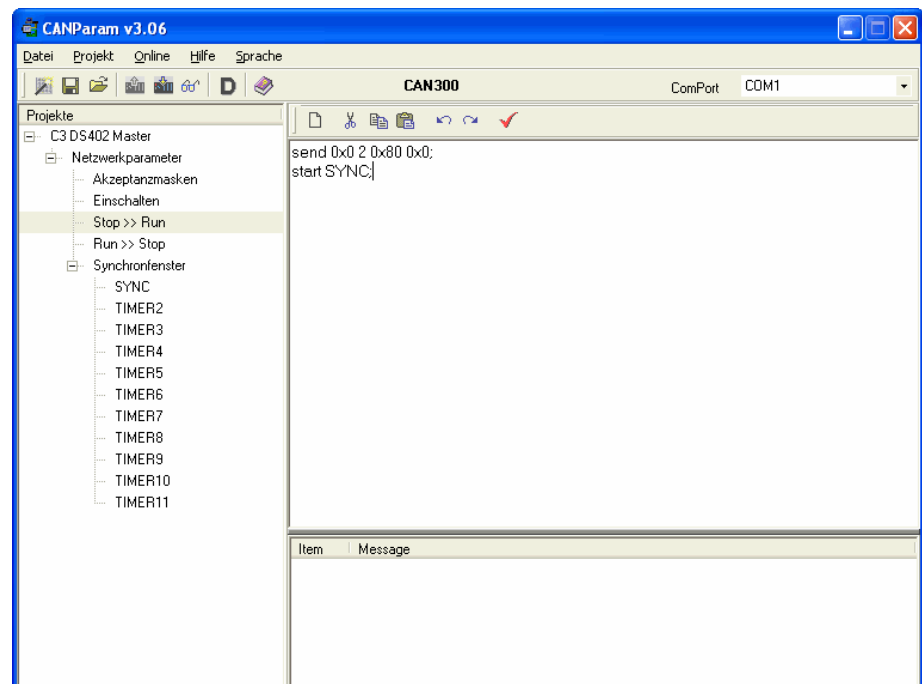
2.3 Akzeptanzmasken

Für CANopen Master Anwendungen werden im Normalfall immer alle CAN-Bus Telegramme an die SPS durchgelassen.

	Anfang	Ende
<input checked="" type="checkbox"/> Maske 1	0x000	0x7FF
<input type="checkbox"/> Maske 2	0x000	0x000
<input type="checkbox"/> Maske 3	0x000	0x000
<input type="checkbox"/> Maske 4	0x000	0x000
<input type="checkbox"/> Express Maske	0x000	0x000
<hr/>		
<input type="checkbox"/> Maske 6	0x000	0x000
<input type="checkbox"/> Maske 7	0x000	0x000
<input type="checkbox"/> Maske 8	0x000	0x000
<input type="checkbox"/> Maske 9	0x000	0x000
<input type="checkbox"/> Maske 10	0x000	0x000
<input type="checkbox"/> Maske 11	0x000	0x000
<input type="checkbox"/> Maske 12	0x000	0x000
<input type="checkbox"/> Maske 13	0x000	0x000
<input type="checkbox"/> Maske 14	0x000	0x000
<input type="checkbox"/> Maske 15	0x000	0x000
<input type="checkbox"/> Maske 16	0x000	0x000

2.4 Netzmanagement

In dem Beispielprojekt werden die Skripte „Stop >> Run“ und „Run >> Stop“ verwendet. Im Anlaufskript wird der Motor in einen definierten Zustand gebracht und der SYNC Timer aktiviert.

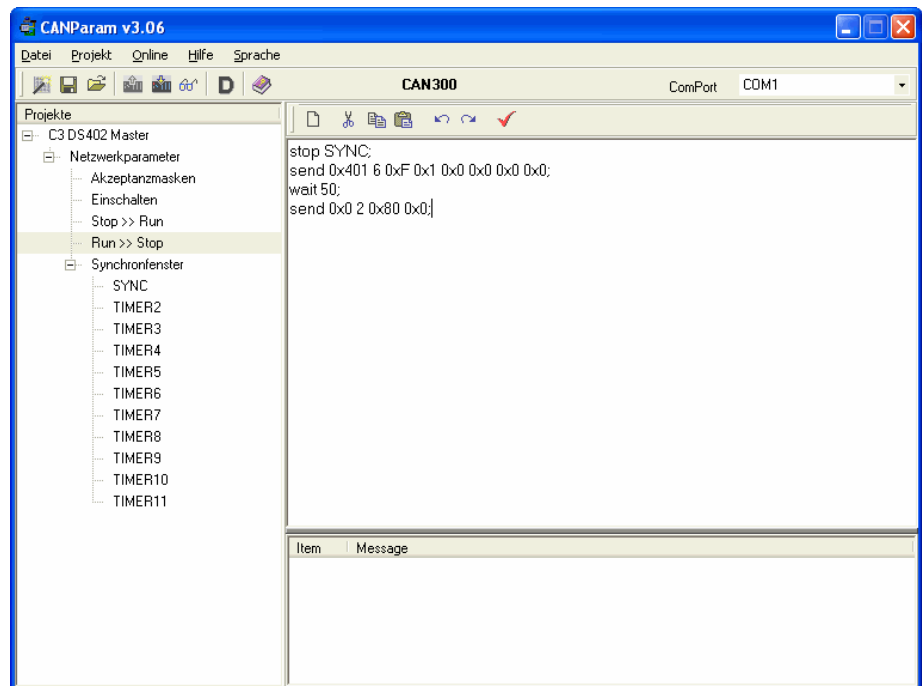


Das Starten des CAN-Bus wird durch die Hantierung durchgeführt.

Wenn die SPS angehalten wird, wird das „Run >> Stop“ Skript ausgeführt. Dieses Skript hält den SYNC-Timer an, sendet an den Motor ein Telegramm, mit dem laufende Bewegungen angehalten werden („Quick Stop“), und stoppt dann den CANopen Bus (NMT Stop all Nodes).

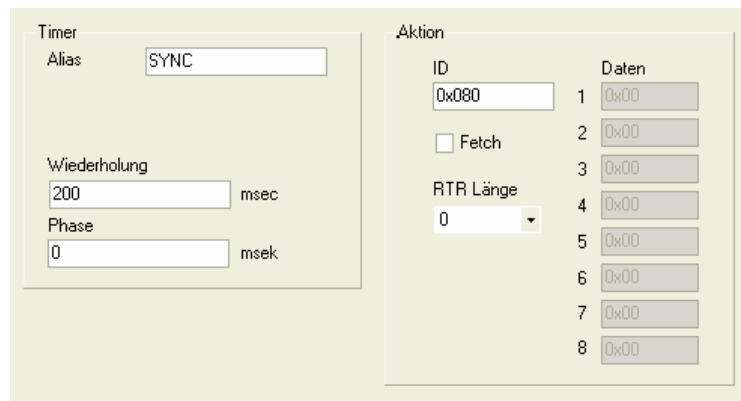


Der Eintrag „send 0x401...“ muß an die jeweilige Nodenummer des verwendeten Antriebs angepasst werden



2.5 Timer

Der SYNC-Timer wird im Skript „Stop >> Run“ gestartet.



3 Programmierung in der SPS

3.1 Übersicht

Das Beispiel enthält die Hantierungsbausteine der CANopen Master Hantierung (FC40 - FC49), die im Handbuch „CAN 300“ bzw. „CAN 400“ ausführlich erläutert sind. Das Anwendungsbeispiel ist für die CAN 400 gleich aufgebaut, die Erläuterungen lassen sich ohne weiteres übertragen.

Die FB30 bis FB37 sind die für die Hantierung des Motors speziell erstellten Bausteine und rufen die Bausteine der CANopen Master Hantierung auf.

3.2 Init Drive (FB 37)

Der Funktionsbaustein **INIT_DRIVE (FB 37)** initialisiert alle wichtigen Kommunikationsparameter des Antriebs.

```
CALL FB 37 , DB37
  Activate:=M109.7
  Node :=2
  Busy :=M111.5
  Done :=M111.6
  Error :=M111.7
  ErrorNo :=MW112
```

Parameter	Typ		Funktion
Activate	IN	BOOL	Aktivierungsbit
Node	IN	INT	Node-ID
Busy	OUT	BOOL	Anzeigebit für laufenden Funktion
Done	OUT	BOOL	Anzeigebit Initialisierung fertig
Error	OUT	BOOL	Anzeigebit für aufgetretenen Fehler
ErrorNo	OUT	WORD	Fehlernummer von CANopen Hantierungsbausteinen

Der Funktionsbaustein beschreibt folgende SDOs:

- SDO 1800/2 = 0xFF TPDO1 Transmission Type = Event
- SDO 1801/2 = 0xFF TPDO2 Transmission Type = Event
- SDO 1802/2 = 1 TPDO3 Transmission Type = Sync
- SDO 1803/2 = 1 TPDO4 Transmission Type = Sync

Am Ende des Ablaufes wird die Antrieb per „NMT Start“ in den Operational Mode versetzt.

3.3 Homing (FB 30, FB 31)

Der Funktionsbaustein **INIT_HOMING (FB 30)** aktiviert den Homing Mode des Antriebs und überträgt die notwendigen Parameter.

```

CALL FB 30 , DB30
  Activate      :=M109.0
  Node         :=2
  HomeOffset   :=L#0
  HomingAcceleration:=L#200
  SpeedSwitchSearch :=L#500
  SpeedZeroSearch  :=L#500
  QuickStopDecel  :=L#150
  Busy         :=M111.0
  Done        :=M111.1
  Error       :=M111.7
  ErrorNo    :=MW112
  Statusword :=MW30
  
```

Parameter	Typ		Funktion
Activate	IN	BOOL	Aktivierungsbit
Node	IN	INT	Node-ID
HomeOffset	IN	DWORD	→ SDO 607C/0
HomingAcceleration	IN	DWORD	→ SDO 609A/0
SpeedSwitchSearch	IN	DWORD	→ SDO 6099/1
SpeedZeroSearch	IN	DWORD	→ SDO 6099/2
QuickStopDecel	IN	DWORD	→ SDO 6085/0
Busy	OUT	BOOL	Anzeige bit für laufenden Funktion
Done	OUT	BOOL	Anzeige bit Initialisierung fertig
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler
ErrorNo	OUT	WORD	Fehlernummer von CANopen Hantierungsbausteinen
Statusword	OUT	WORD	Statuswort des Antriebs (aus PDO1)

Der Funktionsbaustein **HOMING (FB 31)** führt einen Hominglauf durch.

```

CALL FB 31 , DB31
  Node      :=2
  HomePosition:=L#0
  HomingMethod:=MW28
  Activate   :=M109.1
  Halt      :=M109.2
  Statusword :=MW30
  Position  :=MD32
  Velocity  :=MD36
  Done     :=M111.2
  Error    :=M110.7
  
```

Parameter	Typ		Funktion
Node	IN	INT	Node-ID
HomePosition	IN	DWORD	Wert für Home Position
HominMethod	IN	INT	Homing Methode (→ SDO 6098/0)
Activate	IN	BOOL	Aktivierungsbit für Homing Start
Halt	IN	BOOL	Aktivierungsbit für Homing Halt
Statusword	OUT	WORD	Statuswort des Antriebs (aus PDO1)
Position	OUT	WORD	Aktuelle Position des Antriebs (aus PDO3)
Velocity	OUT	WORD	Aktuelle Drehzahl des Antriebs (aus PDO4)
Done	OUT	BOOL	Anzeige bit "Homing reached"
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler

3.4 Profile Position (FB 32, FB 33)

Der Funktionsbaustein **INIT_POSITION (FB 32)** aktiviert den Profile Position Mode des Antriebs und überträgt die notwendigen Parameter.

```

CALL FB 32 , DB32
  Activate          :=M108.4
  Node              :=2
  ProfileVelocity   :=L#2000
  ProfileAcceleration:=L#200
  ProfileDeceleration:=L#100
  QuickStopDecel   :=L#150
  Busy              :=M110.4
  Done              :=M110.5
  Error             :=M111.7
  ErrorNo           :=MW112
  Statusword       :=MW30
  
```

Parameter	Typ		Funktion
Activate	IN	BOOL	Aktivierungsbit
Node	IN	INT	Node-ID
ProfileVelocity	IN	DWORD	→ SDO 6081/0
ProfileAcceleration	IN	DWORD	→ SDO 6083/0
ProfileDeceleration	IN	DWORD	→ SDO 6084/0
QuickStopDecel	IN	DWORD	→ SDO 6085/0
Busy	OUT	BOOL	Anzeige bit für laufenden Funktion
Done	OUT	BOOL	Anzeige bit Initialisierung fertig
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler
ErrorNo	OUT	WORD	Fehlernummer von CANopen Handierungsbausteinen
Statusword	OUT	WORD	Statuswort des Antriebs (aus PDO1)

Der Funktionsbaustein **POSITION (FB 33)** führt eine Positionierung durch.

```

CALL FB 33 , DB33
  Node              :=2
  TargetPosition    :=MD24
  Activate          :=M108.5
  RelativPositioning :=M108.7
  ImmediatePositioning:=FALSE
  Halt              :=M108.6
  Statusword        :=MW30
  Position          :=MD32
  Velocity          :=MD36
  Done              :=M110.6
  Error             :=M111.7
  
```

Parameter	Typ		Funktion
Node	IN	INT	Node-ID
TargetPosition	IN	DWORD	Ziel Position
Activate	IN	BOOL	Aktivierungsbit für Homing Start
RelativPositioning	IN	BOOL	Steuerbit relative/absolute Positionierung
ImmediatePositioning	IN	BOOL	Steuerbit immediate Positionierung
Halt	IN	BOOL	Aktivierungsbit für Positioning Halt
Statusword	OUT	WORD	Statuswort des Antriebs (aus PDO1)
Position	OUT	WORD	Aktuelle Position des Antriebs (aus PDO3)
Velocity	OUT	WORD	Aktuelle Drehzahl des Antriebs (aus PDO4)
Done	OUT	BOOL	Anzeige bit "Position reached"
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler

3.5 Profile Velocity (FB 34, FB 35)

Der Funktionsbaustein **INIT_VELOCITY (FB 34)** aktiviert den Profile Velocity Mode des Antriebs und überträgt die notwendigen Parameter.

```

CALL FB 34 , DB34
  Activate      :=M108.0
  Node         :=2
  MaxProfileVelocity :=L#4500
  ProfileAcceleration:=L#200
  ProfileDeceleration:=L#100
  QuickStopDecel :=L#150
  Busy         :=M110.0
  Done         :=M110.1
  Error        :=M111.7
  ErrorNo      :=MW112
  Statusword   :=MW30
    
```

Parameter	Typ		Funktion
Activate	IN	BOOL	Aktivierungsbit
Node	IN	INT	Node-ID
MaxProfileVelocity	IN	DWORD	→ SDO 607F/0
ProfileAcceleration	IN	DWORD	→ SDO 6083/0
ProfileDeceleration	IN	DWORD	→ SDO 6084/0
QuickStopDecel	IN	DWORD	→ SDO 6085/0
Busy	OUT	BOOL	Anzeige bit für laufenden Funktion
Done	OUT	BOOL	Anzeige bit Initialisierung fertig
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler
ErrorNo	OUT	WORD	Fehlernummer von CANopen Handlingsbausteinen
Statusword	OUT	WORD	Statuswort des Antriebs (aus PDO1)

Der Funktionsbaustein **VELOCITY (FB 35)** aktiviert eine Drehzahl.

```

CALL FB 35 , DB35
  Node         :=2
  TargetVelocity:=MD20
  Activate     :=M108.1
  Halt        :=M108.2
  Statusword   :=MW30
  Position     :=MD32
  Velocity     :=MD36
  Done        :=M110.2
  Error       :=M111.7
    
```

Parameter	Typ		Funktion
Node	IN	INT	Node-ID
TargetVelocity	IN	DWORD	Soll-Drehzahl
Activate	IN	BOOL	Aktivierungsbit für Homing Start
Halt	IN	BOOL	Aktivierungsbit für Positioning Halt
Statusword	OUT	WORD	Statuswort des Antriebs (aus PDO1)
Position	OUT	WORD	Aktuelle Position des Antriebs (aus PDO3)
Velocity	OUT	WORD	Aktuelle Drehzahl des Antriebs (aus PDO4)
Done	OUT	BOOL	Anzeige bit "Velocity reached"
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler

3.6 Fault Reset (FB 36)

Der Funktionsbaustein **FAULT_RESET (FB 35)** setzt Fehler im Antrieb zurück. Ob ein „Fault“ im Antrieb vorliegt, kann aus dem Statusword ersehen werden (s.a. Kap. 3.9).

CALL	FB	36	, DB36
Node	:	=1	
Activate	:	=M109.6	
Statusword	:	=MW30	
Done	:	=M111.4	
Error	:	=M111.7	

Parameter	Typ		Funktion
Node	IN	INT	Node-ID
Activate	IN	BOOL	Aktivierungsbit für Homing Start
Halt	IN	BOOL	Aktivierungsbit für Positioning Halt
Statusword	OUT	WORD	Statusword des Antriebs (aus PDO1)
Done	OUT	BOOL	Anzeige bit kein Fault Zustand mehr vorhanden
Error	OUT	BOOL	Anzeige bit für aufgetretenen Fehler

3.7 Beispiel FC 1 (ERL_Test)

Die Funktionalitäten der Hantierungsbausteine werden im FC1 durch die Bits der Merkerbytes 8 & 9 ausgelöst.

Am Anfang des FC 1 wird der Zyklusbaustein FC 49 aufgerufen, um über den CAN-Bus empfangene Telegramme zu holen, oder Sendeaufträge abzuarbeiten.

Die Antriebsfunktionen werden mit folgenden Merkern aktiviert:

M 108.0	Init Profile Velocity Mode (FB34)
M 108.1	Set Velocity (FB35)
M 108.2	Halt Velocity (FB35)
MD 20	Target Velocity (FB35)
M 108.4	Init Profile Position Mode (FB32)
M 108.5	Activate Positioning (FB33)
M 108.6	Halt Positioning (FB33)
MD 24	Target Position (FB33)
M 108.7	Relative/Absolut Positioning (FB33)
M 109.0	Init Homing Mode (FB30)
M 109.1	Activate Homing (FB31)
M 109.2	Halt Homing (FB31)
MB 28	Homing Method (FB31)
M 109.6	Fault Reset(FB36)
M 109.7	Init Drive (FB37)

Als Ausgaben liegen folgende Werte vor:

MW 30	Statusword
MD 32	Actual Velocity
MD 36	Actual Position

3.8 Verwendung mit mehreren Antrieben

Die Funktionsbausteine lassen sich mehrfach für verschiedenen Antriebe hintereinander aufrufen. Es muss dann lediglich der Parameter Node angepasst werden. Sollten die Bausteine VELOCITY, POSITION und HOMING für verschiedene Antriebe (Nodes) gleichzeitig aufgerufen werden, dann muss für jeden Node ein eigener Instanzdatenbaustein verwendet werden.



Die „INIT_“ Bausteine dürfen nicht gleichzeitig aktiviert werden.

Zu beachten ist, dass der gleichzeitige Aktivierung der „INIT_...“ Bausteine vermieden werden sollte, da die in den Bausteinen ablaufende Übertragung der SDOs nicht gleichzeitig durchgeführt werden kann. Diese Bausteine dürfen für jeden Antrieb (Node) nur hintereinander aufgerufen werden.

3.9 Inhalt des Statuswortes

Das Statuswort des ERL PCS Antriebes wird im Anwendungsbeispiel im MW30 abgelegt.

	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	
Mode	Home Switch active								Negative Limit Switch active	Positive Limit Switch active							
	Remote									Internal Limit Active							
Profile Position Mode	Target Reached								Internal Limit Active								
Homing Mode	Mode Specific:								Mode Specific:								
Profile Velocity Mode	Setpoint Acknowledge Following Error								Homing Attained Homing Error								
Other Modes	Speed = 0								0								
	X								X								
Mode	Ready to switch On								Negative Limit Switch active								
Profile Position Mode	Switched On								Positive Limit Switch active								
Homing Mode	Operation Enabled								Negative Limit Switch active								
Profile Velocity Mode	Fault								Positive Limit Switch active								
Other Modes	Voltage Disabled								Negative Limit Switch active								
Mode	Quick Stop								Positive Limit Switch active								
Profile Position Mode	Switch On Disabled								Negative Limit Switch active								
Homing Mode	Warning								Positive Limit Switch active								
Profile Velocity Mode	Not Ready to Switch On								Negative Limit Switch active								
Other Modes	Switched On Disabled								Positive Limit Switch active								
Mode	Ready to Switch On								Negative Limit Switch active								
Profile Position Mode	Switched On								Positive Limit Switch active								
Homing Mode	Operation Enabled								Negative Limit Switch active								
Profile Velocity Mode	Fault								Positive Limit Switch active								
Other Modes	Fault Reaction Active								Negative Limit Switch active								
Mode	Quick Stop Active								Positive Limit Switch active								
Profile Position Mode	Warning								Positive Limit Switch active								