
POSITIONIER- UND BAHNSTEUERUNG

APCI-8008 – ETHERCAT

(APCI-8008 / APCLe-8008 / APCLe-8008-EC)

Stand: 21.09.2021, ab Disk V2.53VO
Karten-Revision: APCI-8008 Rev. C
APCLe-8008 Rev. A
APCLe-8008-EC Rev. 0

Rev. 05/052022

www.addi-data.de

1 EtherCAT-Master	5
1.1 Kurzbeschreibung	5
2 Hardware-Voraussetzungen	6
2.1 APcLe-8008-EC	6
2.2 APCI-8008 und APcLe-8008	6
2.3 Lage der Stecker auf BOB-3100 und APCI-8008	7
3 Software-Voraussetzungen	9
4 Inbetriebnahme des EtherCAT-Systems	10
4.1 Programm fwsetup.exe	10
4.1.1 fwsetup Monitor-Screen – HVC0-Tab	10
4.1.1.1 RWMOS-Version und Optionen anzeigen	10
4.1.1.2 Weitere fwsetup-Diagnosemöglichkeiten	10
4.1.1.2.1 Alias-Adresse zuweisen	10
4.1.2 fwsetup EtherCAT-Tab	11
4.1.2.1 EtherCAT-Seite: Configuration	12
4.1.2.2 EtherCAT-Seite: SAP	13
4.1.2.2.1 Erläuterung der grafischen Bediensymbole	14
4.1.2.2.2 Erläuterung der Bildschirmtext-Ausgabe EtherCAT SAP	15
4.1.2.3 EtherCAT Seite: Preferences	16
4.1.3 EtherCAT-SAP-Programmierung	16
4.1.3.1 EtherCAT-Zugriffe auf Slave-Module per universellem Objekt-Interface	16
4.1.3.2 Zugriffe auf Service-Daten-Objekte (SDOs)	17
4.1.3.3 EtherCAT-Fehlerbehandlung	17
4.1.4 EtherCAT-PCAP-Programmierung	17
4.1.4.1 Besonderheiten bei Zugriffen auf Service-Daten-Objekte (SDO)	17
4.2 SDO-Initialisierung mit Hilfe von Konfigurationsdaten in einer Datei	18
4.2.1 Aufbau der Konfigurationsdatei	18
4.2.1.1 Erste Zeile von sdos.conf	18
4.2.1.2 Folgezeilen in sdos.conf	18
4.2.1.3 Kommentare	18
4.2.2 Speichern einer Konfigurationsdatei im Dateisystem der APcIx-8008-EC	19
4.3 EtherCAT-Diagnose per universellem Objekt-Interface	19
4.3.1 Working Counter und Anzahl der erkannten Slaves im Bus	19
4.3.2 Lost Frame Counter	20
4.3.3 Status der Slaves im Bus	20
4.4 Lesen von Bus- und Slave-Informationen	20
4.4.1 Generelle Informationen der Slaves im Bus ermitteln	20
4.4.1.1 Dev.# 700 hex	21

4.4.2	EtherCAT-Master-Informationen lesen	21
4.4.2.1	Dev.# 900 hex	21
4.4.3	SDOs der Busteilnehmer lesen und schreiben	22
4.5	Manuelles Lesen und Beschreiben von SDOs im Slave-Verzeichnis.....	22
4.6	Inbetriebnahme von Achsen in mcfg.exe	23
4.6.1	Vergabe der Motorachsen.....	23
4.6.2	Einstellung der Istwert-Auflösung.....	23
4.7	Sonstige EtherCAT-spezifische Handhabung des Systems	24
4.7.1	Restart der Karten per BootFile	24
4.7.2	Übertragen des dmesg-Protokolls ins Windows-Dateisystem	24
5	Antriebsspezifische Eigenschaften	25
5.1	Omron-Servoantriebe.....	25
5.1.1	Accurax G5-Antriebe	25
5.1.1.1	Betriebsarten	25
5.1.1.2	Digitale Eingänge	25
5.1.2	Digitale Ausgänge	25
5.1.3	Touch Probe-Funktion / Latch-Funktionalität	25
5.1.4	Sysmac 1S-Servoantriebe.....	26
5.2	Panasonic MADHT1505.....	26
5.2.1	Betriebsarten	26
5.2.2	Digitale Eingänge	26
5.2.3	Digitale Ausgänge	26
5.2.4	Touch Probe-Funktion / Latch-Funktionalität	26
5.2.5	Quick Stop Deceleration (Object 6085 hex).....	26
6	Konfiguration und Diagnose in der Konsole der APCI-8008.....	27
6.1	Anpassung von herstellereigenen XML-Dateien für die Verwendung mit der APCI-8008 ...	27
6.1.1	SM=??	27
6.1.2	PDOs ergänzen.....	27
6.2	EtherCAT-Modul neu starten	27
6.3	EtherCAT-SDOs.....	28
7	Anhang.....	30
7.1	Fallstricke bei der Einrichtung des Systems	30
7.2	Implementierte Geräte	31
7.2.1	E/A-Module.....	31
7.2.2	Antriebe	31
7.3	Abbildungsverzeichnis	32
7.4	Tabellenverzeichnis	32

1 EtherCAT-Master

Die APCI-8008-EC stellt dem Anwender die bekannte Funktion der APCI-8008 mit der Möglichkeit der Anschaltung von EtherCAT-Slave-Komponenten zur Verfügung. Diese können Ein-/Ausgabemodule oder Antriebselemente sein.

In der Familie der Karte APCLx-8008 gibt es drei Produkte, mit denen eine EtherCAT-Anbindung möglich ist:

APCI-8008 und BOB-3100
APCLe-8008 und BOB-3100
APCLe-8008-EC.

1.1 Kurzbeschreibung

Die im EtherCAT-Bus zu verwendenden Komponenten (Slaves) müssen in der vorliegenden Systemsoftware implementiert sein. Deshalb muss vor jeder Verwendung die Eignung der vorgesehenen EtherCAT-Komponenten geprüft und gegebenenfalls vom Hersteller der APCLx-8008 in die Systemsoftware integriert werden.

Bei den Karten APCI-8008 und APCLe-8008 ist auch eine Mischversion von EtherCAT-Komponenten und herkömmlichen Antriebskomponenten möglich. Das heißt, die Standardschnittstellen (Analog Out, Inkrementalencoder, I/O-Interface) können gleichzeitig mit EtherCAT-Komponenten auf demselben System betrieben werden.

Die Zykluszeit der Steuerung ist fest eingestellt auf 1 ms und kann nicht verändert werden.

2.3 Lage der Stecker auf BOB-3100 und APCI-8008

Abbildung 2-2: Anschlüsse BOB-3100

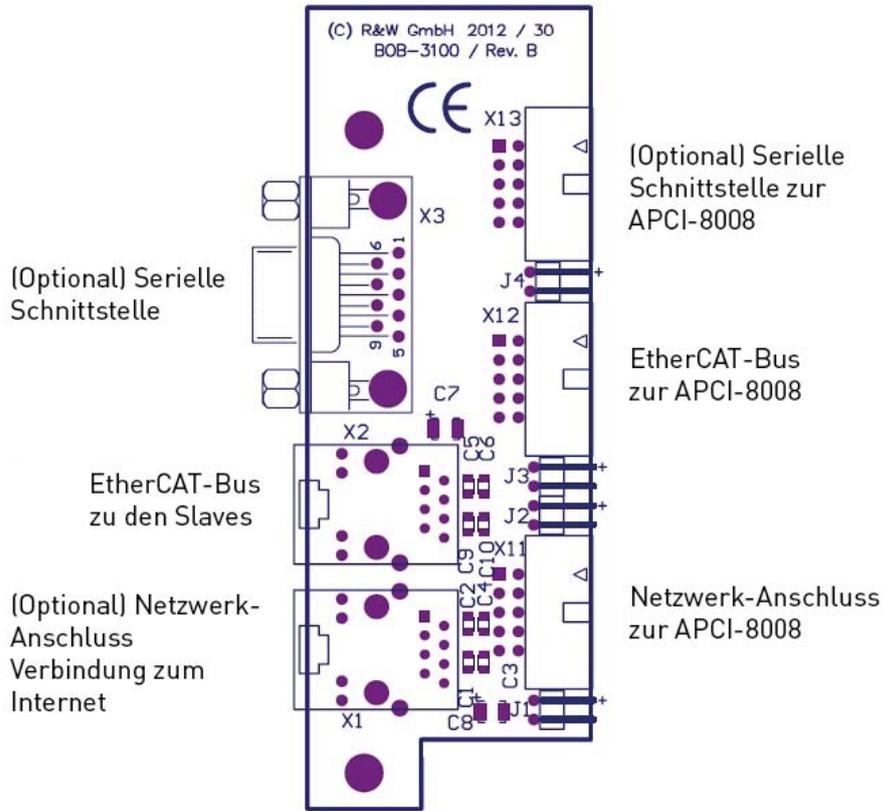
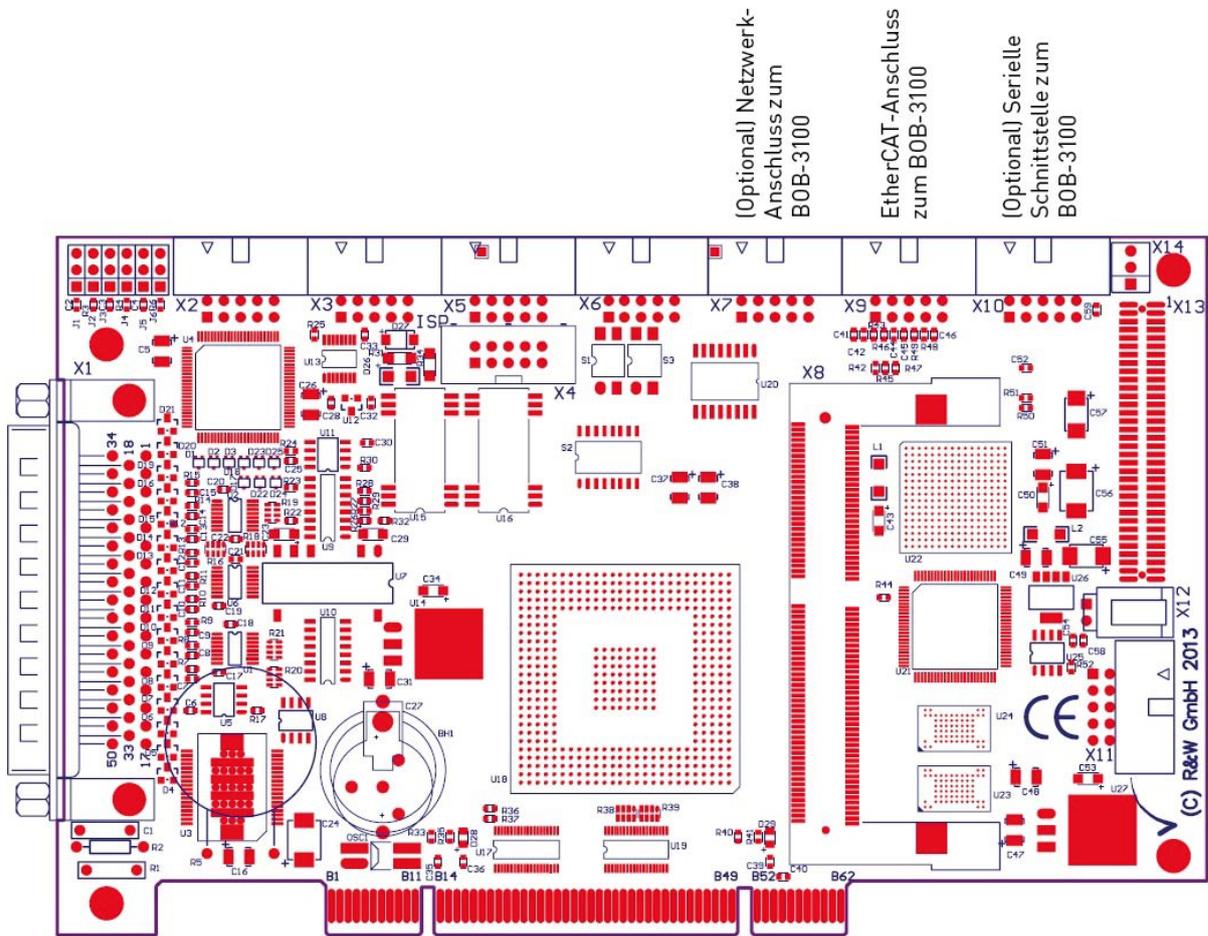


Abbildung 2-3: Anschlüsse APCI-8008



(Optional) Netzwerk-Anschluss zum BOB-3100

EtherCAT-Anschluss zum BOB-3100

(Optional) Serielle Schnittstelle zum BOB-3100

3 Software-Voraussetzungen

Die APCI-8008-EC muss so konfiguriert sein, dass auf der Karte Linux und die Linux-Variante des APCI-8008 RWMOS.ELF ausgeführt werden. Diese Variante bootet selbständig nach dem Einschalten des PCs bzw. nach einem System-Reset in fwsetup.exe.

Der Aufruf der DLL-Funktion BootFile ist hier nicht erlaubt und führt zu einer Systemstörung.

Wie bei der Standardausführung der APCI-8008 existieren auch für die EC-Version unterschiedliche RWMOS-Varianten. Hier ist ein Update über einen FTP-Server möglich. Dazu muss X7 der APCI-8008-EC mit X11 des BOB-3100 verbunden sein. Der Netzwerkanschluss für den Internetzugang erfolgt am Stecker X1 des BOB-3100 (RJ-45).

Hinweis: Beim Hochfahren des Linux-Kerns der APCI-8008-EC muss die Netzwerkverbindung bereits bestehen; ansonsten ist in der jeweiligen Session kein Internet-Update möglich.

4 Inbetriebnahme des EtherCAT-Systems

4.1 Programm fwsetup.exe

4.1.1 fwsetup Monitor-Screen – HVC0-Tab

Gegenüber dem sogenannten Monitor-Screen bei der Standardvariante der APCI-8008 steht hier in fwsetup die Registerkarte HVC0 zur Verfügung. Diese repräsentiert eine Linux-Konsole, mit der auf den Linux-Kern der APCI-8008-EC zugegriffen werden kann. Alternativ kann auch mit einem Terminalprogramm per Telnet auf die Karte zugegriffen werden. Die erforderliche IP-Adresse kann auf der Registerkarte HVC0 per ifconfig ermittelt werden. Per User root kann mit dem Standard-Passwort „mcu“ der Login erfolgen.

Bei einem Terminalprogramm hat man die Möglichkeit einer umfangreicheren Anzeige, weil in fwsetup Zeichen außerhalb des Anzeigebereichs abgeschnitten werden. Dennoch ist der Zugriff per fwsetup für viele Anwendungsfälle ausreichend.

Hinweis: Man kann die abgeschnittenen Bereiche sichtbar machen durch Übernahme des Monitorinhalts in die Zwischenablage (per Maustaste: „Copy to ClipBoard“) und Einfügen der Zwischenablage in einen Editor.

4.1.1.1 RWMOS-Version und Optionen anzeigen

Die Diagnose der verwendeten RWMOS-Variante kann mit folgendem Kommando erfolgen:

```
/opt/mcu/info.sh
```

Dies ergibt z.B. folgende Ausgabe:

```
Fri Sep 14 10:05:05 CEST 2018
```

```
This is rwmoslkm with build number 2.5.3.144-5e77b1c, created at Sep 14 2018 09:26:29.
```

```
Compiled in options are: APCI-8008-LINUX [3C5, 18A, EV, SSI, RESOURCE, SCANNER, ELCAM, GEAR, SPLINE, TC, LANGHAM, EXTENDED_DIAG, PIDFIDLE, MULTICONTROL, EC].
```

```
Up to 5 axis are supported, cycle time is: 1000 micro seconds.
```

Kennzeichnend und Voraussetzung für die Option EtherCAT ist hier die Betriebssystem-Option „EC“.

4.1.1.2 Weitere fwsetup-Diagnosemöglichkeiten

An dieser Stelle gibt es viele Eingabemöglichkeiten zur Systemkonfiguration und Diagnose, wie z.B. die EtherCAT-Statusanzeige:

```
ethercat diag
```

Eine gruppierte Beschreibung dieser Möglichkeiten befindet sich in Kapitel 6.

4.1.1.2.1 Alias-Adresse zuweisen

Jedem Slave kann eine eindeutige Alias-Adresse zugewiesen werden. Dieser Wert ist dann auf dem jeweiligen Slave resident gespeichert und kann beim Start der Applikation abgefragt / verifiziert werden. Damit kann z.B. die Verkabelung des Busaufbaus überprüft werden.

Beispiel:

```
ethercat alias -p0 101
```

-p0 bedeutet, dass die Alias-Adresse dem 1. Slave im Bus zugewiesen wird.

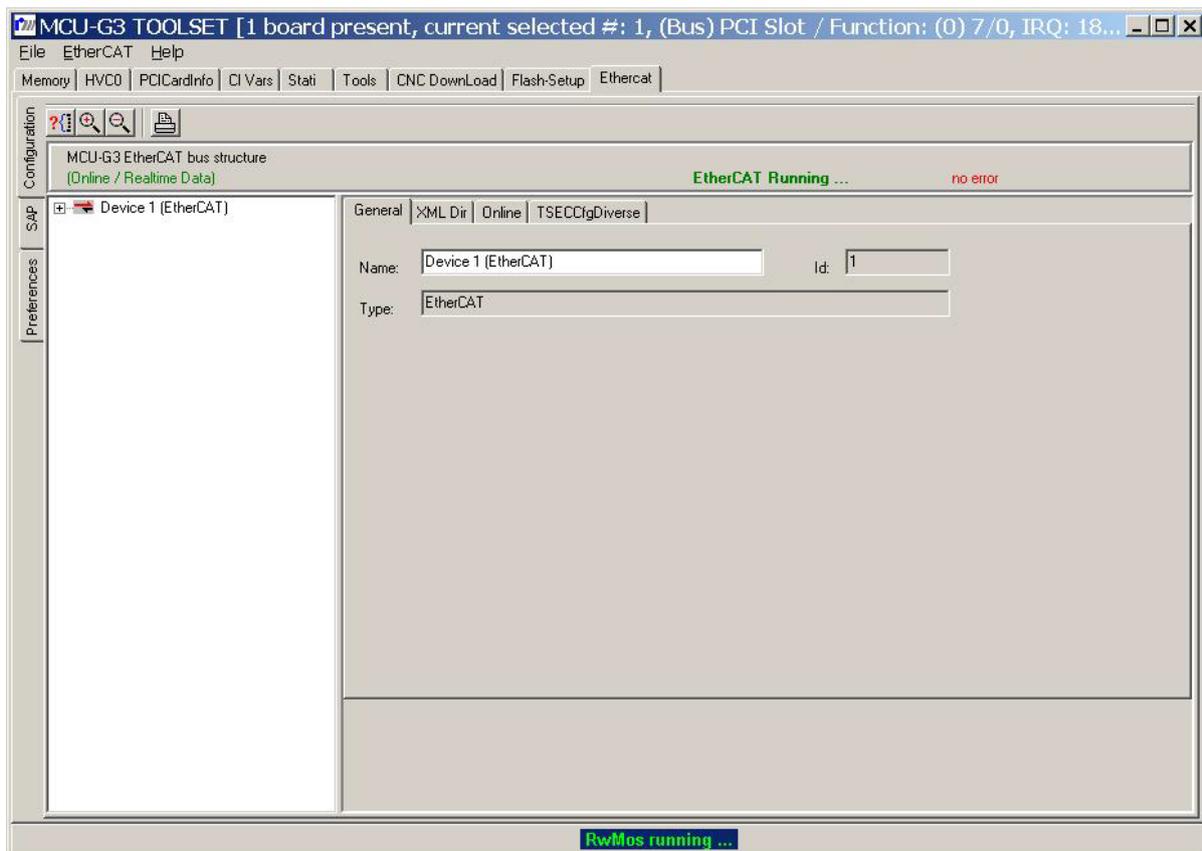
101 ist in diesem Beispiel die Alias-Adresse. Diese wird per ethercat diag in der 2. Spalte angezeigt. Das Lesen der Alias-Adresse zur Laufzeit wird per universellem Objekt-Interface mit der Ressource Dev.# 700 hex Subindex 1 adressiert. Die ab 0 fortlaufende Slave-Position im Bus muss im Index angegeben werden.

4.1.2 fwsetup EtherCAT-Tab

Mit Hilfe des TOOLSET-Programms *fwsetup.exe* kann der an der APCI-8008 angeschlossene EtherCAT-Busaufbau automatisch ermittelt werden, sofern sich der angeschaltete Bus in einem ordnungsgemäßen Zustand befindet. Auf der Registerkarte [Ethercat] können die zusätzlichen Projektierungsmöglichkeiten für den Bus vorgenommen werden. Zur Zeit gibt es drei verschiedene Seiten, auf denen wichtige Informationen über den aktuellen Busaufbau und Zustand ermittelt werden können. Verschiedene hierin enthaltene Daten werden für die SAP- oder PCAP-Programmierung benötigt.

Durch Klicken auf den Button  wird der Busaufbau gelesen. Dies kann einige Zeit in Anspruch nehmen. Durch Klicken auf das Plus-Zeichen des Knotens Device 1 kann der Busaufbau expandiert werden. Device 1 repräsentiert den EtherCAT-Master der APCI-8008.

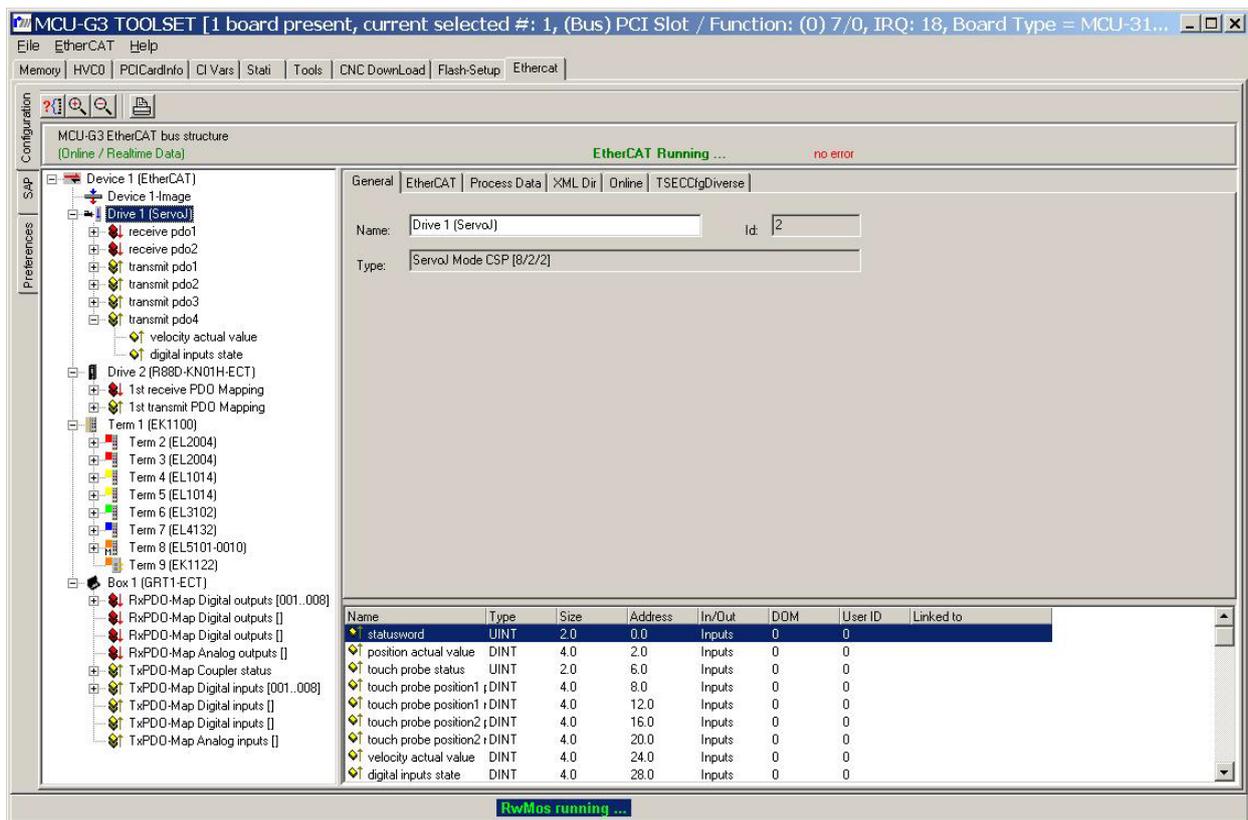
Abbildung 4-1: EtherCAT Configuration



4.1.2.1 EtherCAT-Seite: Configuration

Auf der Seite „Configuration“ kann der am BOB-3100 angeschlossene Busaufbau grafisch in Baumform dargestellt werden (siehe nachfolgender Screenshot). In der obersten Hierarchie-Ebene werden die Buskoppler (Busknoten) mit Gerätekenung und Klartextbeschreibung aufgelistet. Darunter werden die einzelnen Module mit ihren spezifischen Eigenschaften wie Gerätekenung, Beschreibung, Zugriffstyp und Wortbreite angezeigt.

Abbildung 4-2: EtherCAT Configuration: Baumform



Um diese Bildschirmseite aktivieren zu können, muss das Betriebssystem `rwmos.elf` bereits auf der Karte gestartet und der Bus angeschlossen sein (Online-Mode, `RwMos running`).

In der folgenden Tabelle werden die grafischen Bediensymbole (Buttons) des obenstehenden Screenshots erläutert.

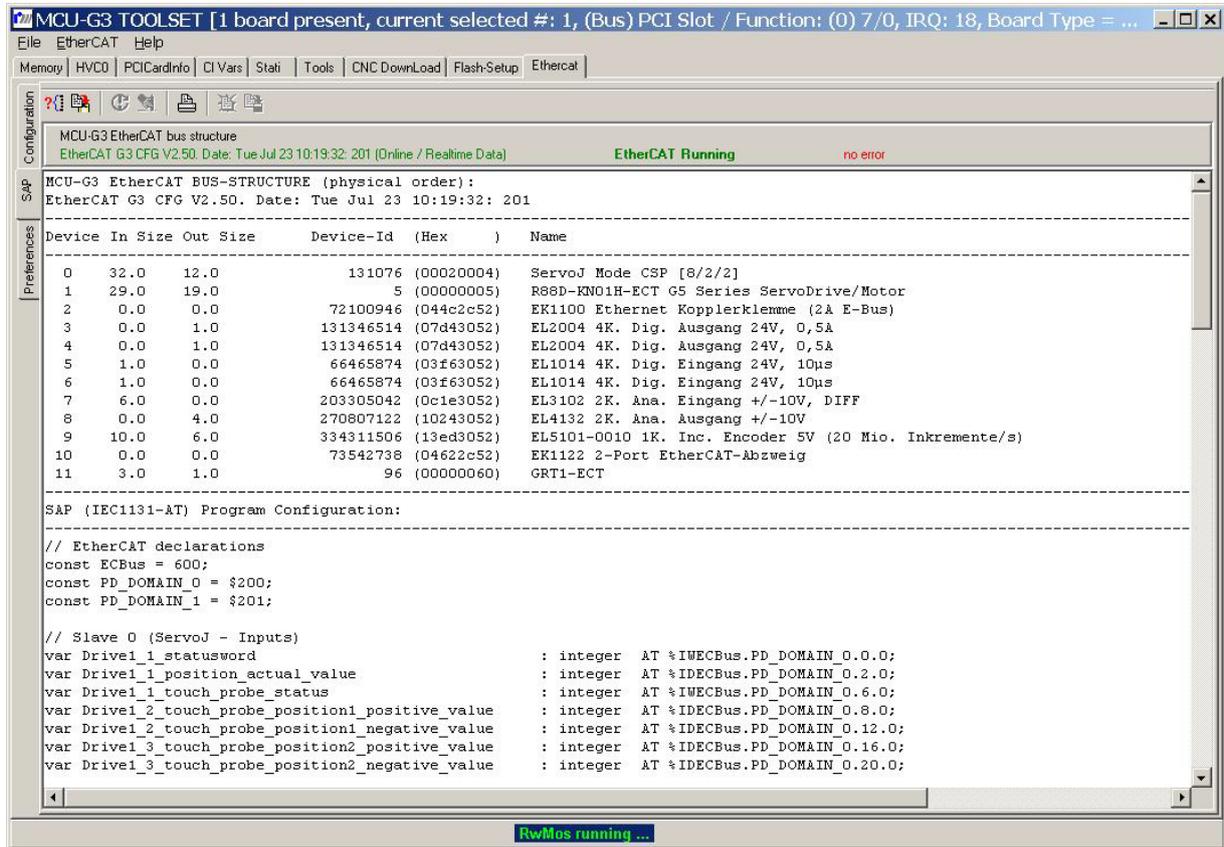
Tabelle 4-1: Bedeutung der Symbole auf der Seite [EtherCAT][Configuration]

Symbol	Bedeutung
	Zuletzt ermittelte EtherCAT-Konfiguration der APCI-8008 einlesen. Es werden die Daten ausgewertet, welche bei der letzten EtherCAT-Initialisierung ermittelt wurden. Hinweis: Dieser Button darf erst betätigt werden, wenn die Initialisierung des EC-Busses abgeschlossen ist. Dies ist im Allgemeinen der Fall, wenn sich alle Teilnehmer im Operational Mode befinden. Je nach Anzahl der Teilnehmer kann das bis zu einer Minute oder länger dauern. Bei Nichtbeachtung ist es möglich, dass Fehlermeldungen auftauchen oder Variablen-Namen falsch (z.B. „unknown“) angezeigt werden. In diesem Fall sollte <code>fwsetup.exe</code> beendet und neu gestartet werden.
	Der aktuelle Geräte-Verzeichnisbaum wird erweitert. Es werden alle verfügbaren Informationen angezeigt.
	Der aktuelle Geräte-Verzeichnisbaum wird zusammengeklappt. Es werden nur noch Busknoten angezeigt.
	Die aktuelle Konfiguration kann für Dokumentationszwecke auf einem Drucker ausgegeben werden.

4.1.2.2 EtherCAT-Seite: SAP

Auf der Seite „SAP“ kann der am BOB-3100 angeschlossene Busaufbau durch Klicken auf  in Textform dargestellt werden (siehe folgender Screenshot).

Abbildung 4-3: EtherCAT SAP



Um diese Bildschirmseite aktivieren zu können, muss das Betriebssystem `rwmos.elf` bereits auf der Karte gestartet sein (Online-Mode, `RwMos running`).

4.1.2.2.1 Erläuterung der grafischen Bediensymbole

In der folgenden Tabelle werden die grafischen Symbole (Speed-Buttons) des in Abbildung 4-3 gezeigten Screenshots erläutert.

Tabelle 4-2: Bedeutung der Symbole auf der Seite [EtherCAT][SAP]

Symbol	Bedeutung
	siehe Tabelle 4-1
	<p>Mit Hilfe dieses Buttons können die nach IEC1131 eingeführten AT-Spezifizierer in die Windows-Zwischenablage kopiert werden zur weiteren Verwendung in einem SAP-Programm. Nach Betätigen dieses Buttons enthält die Windows-Zwischenablage für obenstehenden Busaufbau z.B. folgende Informationen:</p> <pre>// EtherCAT declarations const ECBus = 600; const PD_DOMAIN_0 = \$200; const PD_DOMAIN_1 = \$201; // Slave 0 (ServoJ - Inputs) var Drive1_1_MPRO_402_Status : integer AT %IWECEBus.PD_DOMAIN_9.10.0; var Drive1_1_MPRO_402_ActualTo : integer AT %IWECEBus.PD_DOMAIN_9.12.0; var Drive1_1_MPRO_402_PositionActua : integer AT %IDECEBus.PD_DOMAIN_9.14.0; var Drive1_2_MPRO_INPUT_S : integer AT %IDECEBus.PD_DOMAIN_9.18.0; // Slave 1 (R88D-KN01H-ECT - Inputs) var Drive2_1_Error_code : integer AT %IWECEBus.PD_DOMAIN_9.34.0; var Drive2_1_Statusword : integer AT %IWECEBus.PD_DOMAIN_9.36.0; var Drive2_1_Position_actual_value : integer AT %IDECEBus.PD_DOMAIN_9.38.0; var Drive2_1_Torque_actual_value : integer AT %IWECEBus.PD_DOMAIN_9.42.0; ...</pre> <p>Diese Definitionen können zur Variablen-Deklaration in einem SAP-Programm verwendet werden.</p>
	Die aktuelle Konfiguration kann für Dokumentationszwecke auf einem Drucker ausgegeben werden.

4.1.2.2.2 Erläuterung der Bildschirmtext-Ausgabe EtherCAT SAP

Abbildung 4-4: Bildschirmtext-Ausgabe EtherCAT SAP

```

MCU-G3 EtherCAT BUS-STRUCTURE (physical order):
EtherCAT G3 CFG V2.50. Date: Tue Jul 23 10:19:32: 201
-----
Device In Size Out Size      Device-Id (Hex      ) Name
-----
0  32.0  12.0          131076 (00020004)  ServoJ Mode CSP [8/2/2]
1  29.0  19.0           5 (00000005)  R88D-KN01H-ECT G5 Series ServoDrive/Motor
2  0.0   0.0          72100946 (044c2c52)  EK1100 Ethernet Kopplerklemme (2A E-Bus)
3  0.0   1.0          131346514 (07d43052)  EL2004 4K. Dig. Ausgang 24V, 0,5A
4  0.0   1.0          131346514 (07d43052)  EL2004 4K. Dig. Ausgang 24V, 0,5A
5  1.0   0.0          66465874 (03f63052)  EL1014 4K. Dig. Eingang 24V, 10µs
6  1.0   0.0          66465874 (03f63052)  EL1014 4K. Dig. Eingang 24V, 10µs
7  6.0   0.0          203305042 (0c1e3052)  EL3102 2K. Ana. Eingang +/-10V, DIFF
8  0.0   4.0          270807122 (10243052)  EL4132 2K. Ana. Ausgang +/-10V
9  10.0  6.0          334311506 (13ed3052)  EL5101-0010 1K. Inc. Encoder 5V (20 Mio. Inkremente/s)
10 0.0   0.0          73542738 (04622c52)  EK1122 2-Port EtherCAT-Abzweig
11 3.0   1.0           96 (00000060)  GRT1-ECT
-----
SAP (IEC1131-AT) Program Configuration:
-----
// EtherCAT declarations
const ECBus = 600;
const PD_DOMAIN_0 = $200;
const PD_DOMAIN_1 = $201;

// Slave 0 (ServoJ - Inputs)
var Drive1_1_statusword          : integer AT %IWECEBus.PD_DOMAIN_0.0.0;
var Drive1_1_position_actual_value : integer AT %IDECEBus.PD_DOMAIN_0.2.0;
var Drive1_1_touch_probe_status  : integer AT %IWECEBus.PD_DOMAIN_0.6.0;
var Drive1_2_touch_probe_position1_positive_value : integer AT %IDECEBus.PD_DOMAIN_0.8.0;
var Drive1_2_touch_probe_position1_negative_value : integer AT %IDECEBus.PD_DOMAIN_0.12.0;
var Drive1_3_touch_probe_position2_positive_value : integer AT %IDECEBus.PD_DOMAIN_0.16.0;
var Drive1_3_touch_probe_position2_negative_value : integer AT %IDECEBus.PD_DOMAIN_0.20.0;
var Drive1_4_velocity_actual_value : integer AT %IDECEBus.PD_DOMAIN_0.24.0;
var Drive1_4_digital_inputs_state : integer AT %IDECEBus.PD_DOMAIN_0.28.0;
// Slave 1 (R88D-KN01H-ECT - Inputs)
var Drive2_1_Error_code          : integer AT %IWECEBus.PD_DOMAIN_0.32.0;
var Drive2_1_Statusword          : integer AT %IWECEBus.PD_DOMAIN_0.34.0;
var Drive2_1_Position_actual_value : integer AT %IDECEBus.PD_DOMAIN_0.36.0;
var Drive2_1_Torque_actual_value  : integer AT %IWECEBus.PD_DOMAIN_0.40.0;
var Drive2_1_Modes_of_operation_display : integer AT %IBECEBus.PD_DOMAIN_0.42.0;
var Drive2_1_Touch_probe_status  : integer AT %IWECEBus.PD_DOMAIN_0.43.0;
var Drive2_1_Touch_probe_pos1_pos_value : integer AT %IDECEBus.PD_DOMAIN_0.45.0;
var Drive2_1_Touch_probe_pos2_pos_value : integer AT %IDECEBus.PD_DOMAIN_0.49.0;
var Drive2_1_Digital_inputs      : integer AT %IDECEBus.PD_DOMAIN_0.53.0;
var Drive2_1_Velocity_actual_value : integer AT %IDECEBus.PD_DOMAIN_0.57.0;
// Slave 5 (EL1014 - Inputs)

```

Alle vorhandenen EtherCAT-Geräte erhalten eine fortlaufende Nummer (Spalte *Device*). Diese Nummer entspricht der tatsächlichen Position im Busaufbau. Die Spalte *Device-Id* dient zur Beschreibung der Gerätefunktion. Normalerweise sind die ID-Codes eindeutige Kennungen der EtherCAT-Module in dezimaler und hexadezimaler Schreibweise. Die Spalte *Name* beschreibt das jeweilige Modul im Klartext.

Für die SAP- und PCAP-Programmierung sind ebenfalls wertvolle Informationen in dieser Bildschirmtext-Ausgabe enthalten. Dazu gehören zunächst alle Typdefinitionen, mit deren Hilfe die jeweiligen EtherCAT-Devices angesprochen werden können. Die Codierung der *AT-Specifier* sind die Deklarationen der Systemvariablen für die SAP-Programmierung. Diese stellen gegebenenfalls auch die notwendigen Informationen für den Objektzugriff mit Hilfe der PCAP-Programmierung zur Verfügung. In der SAP-Programmiersprache wird in Anlehnung an die IEC1131-Programmiersprache der Zugriff auf die E/A-Ebene mit Hilfe des Schlüsselwortes AT geregelt (siehe Handbuch „Universelles Objekt-Interface“, Kapitel 2.2).

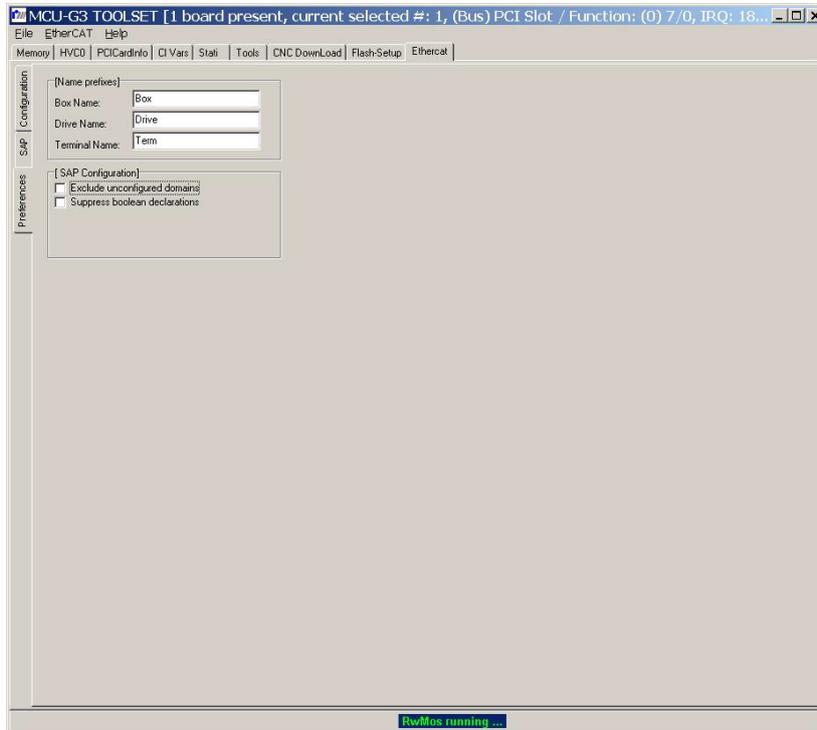
Mit Hilfe des *Access-Type* (erster Buchstabe nach AT % (I oder Q)) kann ermittelt werden, wie auf die jeweilige Systemvariable zugegriffen werden darf, also lesend (I) oder schreibend (Q).

Hinweis: Für den obenstehenden Busaufbau gibt es das SAP-Demoprogramm „SAP_DEMO_EtherCAT.SRC“, in welchem die flexible und einfache EtherCAT-Kommunikation demonstriert wird. Dieses Programm ist auf der Toolset-CD ab V2.53VN zu finden.

4.1.2.3 EtherCAT Seite: Preferences

Auf der Seite „Preferences“ können Vorgaben für die Namensvergabe der SAP-Variablen gemacht werden.

Abbildung 4-5: EtherCAT Diagnostics



4.1.3 EtherCAT-SAP-Programmierung

Die Software-Zusatzfunktionen der Option EC können neben der PCAP-Programmiermethode auch sehr effektiv durch die SAP-Programmiermethode genutzt werden.

Hinweis: Zugriffe auf die Option EC sollten nur aus einer Task erfolgen. Ansonsten muss gewährleistet werden, dass nur sequentiell (nicht gleichzeitig) aus mehreren Tasks auf die Option EC zugegriffen wird. Ist dies nicht gewährleistet, kann dies zu einem unerklärlichen Laufzeitverhalten führen!

4.1.3.1 EtherCAT-Zugriffe auf Slave-Module per universellem Objekt-Interface

Die EtherCAT-Funktion kann durch Lesen und Schreiben der per fwsetup ermittelten Ressourcen genutzt werden. Mit Hilfe dieser Variablen kann direkt mit den einzelnen Slaves kommuniziert werden. Die Definition der einzelnen Variablen ist vom Busaufbau abhängig und nur gültig, solange der Bus nicht verändert wurde. Der Variablenzugriff wird in Anlehnung an die IEC1131-Programmiersprache mit Hilfe des AT-Mechanismus geregelt. Dieser wird ausführlich im Handbuch „Universelles Objekt-Interface“ (Kapitel 2.2) beschrieben. Die hier automatisch generierten AT-Spezifizierer entsprechen den Prozessdatenvariablen (PDOs) des jeweiligen Slaves.

Hinweis: Im SAP-Anwenderprogramm sollte der EtherCAT-Bus überwacht werden, z.B. durch Statusabfragen in einer -Event-Handler-Routine.

4.1.3.2 Zugriffe auf Service-Daten-Objekte (SDOs)

Das universelle Objekt-Interface ermöglicht den Zugriff auf Service-Daten-Objekte der einzelnen Slaves. Die einzelnen AT-Spezifizierer müssen jedoch manuell unter Zuhilfenahme der jeweiligen Produktdokumentation deklariert werden. Hierzu gelten die nachfolgenden Richtlinien.

Für die Bus-Nummer ist EcBus anzugeben. Bei der Device-Nummer muss der jeweilige Slave-Index eingetragen werden. Maximal 512 Slaves werden unterstützt (0..511). Einerseits ist für die Verwendung dieser Variablen natürlich vorausgesetzt, dass der Busaufbau streng eingehalten und nicht verändert wird. Andererseits hat man hiermit auch die Möglichkeit, den erwarteten Busaufbau zu kontrollieren, indem man die Kennung der einzelnen Teilnehmer nacheinander ausliest und verifiziert.

Bei Index und SubIndex ist der jeweilige SDO-Index / Subindex einzutragen. Hierüber gibt die jeweilige Produktdokumentation Auskunft.

Beim Zugriff auf SDOs ist unbedingt darauf zu achten, dass niemals gleichzeitig von verschiedenen Stellen auf SDOs zugegriffen wird, da die Ausführung dieser Kommandos immer erst im Folgezyklus abgeschlossen werden kann. Wenn einem noch nicht abgeschlossenen Zugriff auf ein SDO in der einen Task ein weiterer Zugriff aus einer anderen Task oder per PCAP folgt, dann ist der Ablauf mit nicht absehbaren Folgen gestört.

Des Weiteren ist unbedingt zu beachten, dass Service-Daten-Objekte im Allgemeinen nicht zyklisch beschrieben werden dürfen. Wenn es sich um sogenannte Backup-Daten handelt, die beim Beschreiben im Flash-Speicher des Slaves resident gehalten werden, ist ansonsten eine Zerstörung der Slave-Baugruppe möglich, weil der Flash-Bereich nicht beliebig oft beschrieben werden darf. Abhilfe kann z.B. geschaffen werden, indem eine Variable gelesen und nur bei ungültigem Wert mit dem jeweiligen Initialisierungswert beschrieben wird.

Außerdem ist Folgendes zu beachten:

Nach dem Starten des EC-Busses oder nach dem Wiederaufbau wird auf alle SDOs im gesamten EC-Bus lesend zugegriffen. Während dieser Phase, die mehrere Minuten dauern kann, ist es praktisch nicht möglich, aus einem Programm heraus lesend oder schreibend auf SDOs zuzugreifen, da die entsprechenden Befehle sequentiell abgearbeitet werden. Somit ist eine Vorinitialisierung von SDOs mit Hilfe von Zugriffen über das universelle Objekt-Interface nicht unbedingt geeignet. Stattdessen gibt es ab RWMOS-Version 2.5.3.154 die Möglichkeit, mit Hilfe einer Konfigurationsdatei eine Vorinitialisierung von SDOs im gesamten Bus vorzunehmen.

4.1.3.3 EtherCAT-Fehlerbehandlung

Zur Fehlerbehandlung von EtherCAT-Fehlermeldungen kann der Eventhandler EVEC verwendet werden. Dieser spricht an, wenn im Interface-Status-Register ein Bus-Error BEF erkannt wird. Dieser Fehler kann, wenn überhaupt, nur mit einem ResetSystem-Kommando (rs) auf PCAP-Ebene quittiert werden. Bei diesem Kommando werden jedoch alle SAP-Tasks gestoppt. Auf SAP-Ebene sollte das rs-Kommando nicht verwendet werden. Nach Ausführung des rs-Kommandos muss gewartet werden, bis das Bus-Error-Flag zurückgenommen wurde. Danach kann der weitere Programmablauf initiiert werden.

4.1.4 EtherCAT-PCAP-Programmierung

Die Zugriffe auf die einzelnen Variablen des EtherCAT-Bereichs werden über das universelle Objekt-Interface abgewickelt. Hierzu ist die Liste der Ressourcen heranzuziehen, die in fwsetup auf der Registerkarte EtherCAT auf dem Tab SAP erzeugt werden kann.

4.1.4.1 Besonderheiten bei Zugriffen auf Service-Daten-Objekte (SDO)

Bei einem Zugriff auf ein SDO aus der PCAP-Programmierung muss stets der Rückgabewert des Aufrufs ausgewertet werden. Solange der Rückgabewert BUSY (2) geliefert wird, ist der Zugriff noch in Bearbeitung und muss so lange wiederholt werden, bis Erfolg (4) oder aber vielleicht ein Fehler angezeigt wird.

Auch hier ist unbedingt zu beachten, dass auf SDOs nur von einer einzigen Stelle aus zugegriffen werden darf.

4.2 SDO-Initialisierung mit Hilfe von Konfigurationsdaten in einer Datei

Um eine zuverlässige Initialisierung von SDO-Konfigurationsdaten vorzunehmen, gibt es die Möglichkeit, die entsprechenden Daten in der Konfigurationsdatei „/etc/sdos.conf“ vorzugeben.

Diese Daten werden schon vor dem Erreichen des Operational Mode auf die Slaves geschrieben und auch nach jedem Wiedereinschalten eines Slaves. Um diese Operationen verwenden zu können, sind bestimmte Mindestversionen der angewendeten Software erforderlich:

rwmos-Kernel-Modul	V2.5.3.154
mcug3.dll	V2.5.3.131
fwsetup.exe	V2.5.3.39

4.2.1 Aufbau der Konfigurationsdatei

Die Konfigurationsdatei ist eine Textdatei, die mit einem handelsüblichen Editor erstellt werden kann. Die Datenelemente sind zeilen- und spaltenweise organisiert. Die Spaltentrennung erfolgt mit Leerzeichen (Blanks oder Tabs). Der Dateiname und der Pfad sind fest vorgegeben. Groß- und Kleinschreibung sind zu beachten.

4.2.1.1 Erste Zeile von sdos.conf

Die erste Zeile enthält eine Kennung SDOSINIT und die Versionsinformation V1.0.

```
SDOSINIT V1.0
```

4.2.1.2 Folgezeilen in sdos.conf

In den Folgezeilen befinden sich die Informationen zu den zu initialisierenden SDOs. Die jeweiligen Zahlenwerte können dezimal oder hexadezimal mit vorangestelltem 0x angegeben werden.

Spalte #	Inhalt	Kommentar	
1	Slave Vendor Id	Identifikation eines EtherCAT-Moduls	
2	Slave Product Code	Identifikation eines EtherCAT-Moduls	
3	Revision Number	Identifikation eines EtherCAT-Moduls; bei -1 bleibt die Revisionsnummer unberücksichtigt	
4	Slave Index	Index eines zu initialisierenden Moduls im EC-Bus (mit 0 beginnend); bei -1 bleibt der Index unberücksichtigt; es werden dann alle identifizierten Slave-Module im Bus initialisiert	
5	Index	SDO-Adresse „Index“	
6	SubIndex	SDO-Adresse „SubIndex“	
7	Byte Size	Wortgröße des SDO in Byte (1, 2 oder 4)	
8	Wert	Zahlenwert, mit welchem das jeweilige SDO initialisiert wird	

Beispiel:

```
0x0000066F 0x51505001 -1 -1 0x6085 0 4 100000000
```

4.2.1.3 Kommentare

Kommentarzeilen können mit dem Zeichen "#" (ohne Anführungszeichen) ausgeblendet werden.

4.2.2 Speichern einer Konfigurationsdatei im Dateisystem der APCLx-8008-EC

Die Konfigurationsdatei wird unter dem Windows-System des Host-PCs erstellt und kann mit dem Programm „fwsetup“ auf die Achsensteuerungskarte übertragen werden. Die Datei wird dabei resident im Flash-Dateisystem der Karte gespeichert. Hierzu muss bei betriebsbereitem EtherCAT-System „fwsetup“ aufgerufen werden. Danach kann unter der Registerkarte „Ethercat/Update“ und hier wiederum auf der Registerkarte „File Transfer“ über die Schaltfläche „sdos.conf Installation“ eine Datei gesucht und installiert werden.

Hinweis: Wenn eine Achsensteuerungskarte ausgetauscht wird oder das Dateisystem einer Karte per Update-Funktion ersetzt wird, muss diese Datei auf der Karte wiederhergestellt werden. Diese Information muss in der Produktdokumentation angegeben werden.

4.3 EtherCAT-Diagnose per universellem Objekt-Interface

Die EtherCAT-Technologie hat sich als äußerst zuverlässiges System erwiesen. Dennoch ist es möglich, dass durch unterschiedliche Ursachen der Datentransfer gestört sein kann. In diesem Kapitel sollen Diagnosemöglichkeiten aufgezeigt werden, um eventuelle Fehler oder Handhabungsprobleme aufspüren zu können.

Beim EtherCAT-System der APCI-8008 wird von einer einzigen Sync-Unit ausgegangen, in der alle Slaves enthalten sind. Der EtherCAT-Master scannt beim ersten Start den EC-Bus ab und ermittelt die angeschlossenen Geräte. Erst wenn dieser Vorgang abgeschlossen ist, kann eine Softwarediagnose vorgenommen werden.

4.3.1 Working Counter und Anzahl der erkannten Slaves im Bus

Nach dem Abscannen des EC-Busses stehen die „Expected Working Counter“ für die Domain 0 (Eingangsinformationen) und die Domain 1 (Ausgangsinformationen) zur Verfügung. Diese können als Erstes mit den von der Applikation erwarteten Werten verglichen werden und müssen übereinstimmen. Wenn dies nicht der Fall ist, stimmt die erwartete Buskonfiguration nicht oder der Bus ist massiv gestört. Dies kann z.B. auftreten, wenn Teile der Anlage zu spät eingeschaltet werden. In diesem Fall muss die Buskonfiguration korrigiert und danach die Achsensteuerungskarte neu gestartet werden.

Das Lesen der Expected Working Counter erfolgt mit der Ressource Device # 900 hex, Index 1, SubIndex 0 für Domain 0 und SubIndex 1 für Domain 1. Per SubIndex 2 können beide Werte gleichzeitig gelesen werden, wobei der Expected Working Counter der Domain 0 in den unteren 16 Bit des Funktionswerts abgelegt ist und der Expected Working Counter der Domain 1 in den oberen 16 Bit (z.B. 0007 0007 hex).

Wenn die Expected Working Counter mit den für die Applikation notwendigen Werten übereinstimmen, können die aktuellen Werte der Working Counter mit der Ressource Device # 900 hex, Index 2, Subindex 0, 1 oder 2 (siehe vorheriger Abschnitt) gelesen werden. Es kann allerdings einige Zeit dauern, bis die aktuellen Werte der Working Counter mit den erwarteten Werten übereinstimmen, weil die Slaves erst in den entsprechenden Zustand gebracht werden müssen.

Wenn der Bus im normalen Betriebsfall unterbrochen wird, z.B. durch Ausfall eines Slave-Moduls oder durch Auftrennen des EC-Busses, dann fallen die aktuellen Werte der Working Counter sofort entsprechend zurück. Die Expected Working Counter bleiben auf den beim Hochlauf ermittelten Werten. Falls sich Servo-Module mit geschlossenen Lageregelkreisen im verbleibenden aktiven Bus befinden, werden diese umgehend geöffnet, da man nun davon ausgehen muss, dass ein ordnungsgemäßer Betrieb des Achssystems nicht mehr gewährleistet werden kann.

Wenn in diesem Fehlerfall die erwartete Busstruktur wiederhergestellt wird, erreichen auch die aktuellen Working Counter nach einiger Zeit wieder die erwarteten Werte. Dann ist ein Weiterbetrieb des gesamten Systems ohne Neustart möglich.

Die aktuellen Werte der Working Counter überschreiten niemals die Anzahl der Expected Working Counter.

Mit Hilfe der Ressource Device # 900 hex, Index 3, SubIndex 0 kann die Anzahl der erkannten Slaves im Bus gelesen werden. Auch diese Information ist ein wichtiger Indikator für das ordnungsgemäße Vorhandensein der Busstruktur. Dieser Wert kann jedoch gegenüber den aktuellen Working Countern verzögert zur Verfügung stehen. Mit diesem Wert kann aber auch erkannt werden, wenn dem Bus zur Laufzeit weitere Slaves hinzugefügt werden, was mit der Überwachung der Working Counter nicht möglich ist.

4.3.2 Lost Frame Counter

Der Lost Frame Counter ist ein Fehlerzähler des Master-Moduls, der als erster Hinweis auf Kommunikationsprobleme der Hardware-Ebene herangezogen werden kann. Der Lost Frame Counter zeigt an, wenn ein gesendetes Datenpaket nicht oder beschädigt zurückkommt. Der Lost Frame Counter wird per Device # 900 hex, Index 4, Subindex 0 gelesen. Der Fehlerzähler ist ein inkrementeller Wert und wird nur durch einen Neustart des Systems zurückgesetzt.

4.3.3 Status der Slaves im Bus

Jeder Slave im EtherCAT-Bus hat maximal 4 EtherCAT-Ports, an denen ankommender oder abgehender Ethernet-Datenverkehr möglich ist. Bei einem Slave im EC-Bus sind normalerweise zwei Ports belegt, zum Master bzw. zum vorherigen Slave und zum nachfolgenden Slave. Beim letzten Slave im Bus bzw. in einem Buszweig ist nur ein Port aktiv. Bei einem Slave im Bus mit einer Busverzweigung sind alle 4 Ports aktiv.

Der EtherCAT-Slave stellt für jeden dieser Ports einen Fehlerzähler für „Link Lost Counter“, „CRC Error Counter“ und „RX Error Counter“ zur Verfügung. Diese Zähler können gelesen werden, wenn auf den Slave Zugriff per EtherCAT besteht. Durch Auswertung dieser Informationen kann eine Fehlerstelle im EC-Bus lokalisiert bzw. eingegrenzt werden. Der Zugriff auf diese Fehlerzähler erfolgt über Device # 900 hex gemäß nachfolgender Tabelle.

4.4 Lesen von Bus- und Slave-Informationen

Auf die nachfolgend beschriebenen Informationen kann über das universelle Objekt-Interface zugegriffen werden. Viele Elemente stehen nur für Leseoperationen zur Verfügung. Der Zugriff kann per SAP- oder PCAP-Programmierung erfolgen.

Die Busnummer für die hier beschriebenen Zugriffe ist immer EDBus (600 dez.). Stets zu beachten ist der angegebene Datentyp. Boolesche, Byte- und Single-Words sind in der SAP-Programmierung als integer zu definieren und in der PCAP-Programmierung per wrOptionInt bzw. rdOptionInt zu bedienen. Mit diesen Informationen kann beispielsweise der angeschlossene Bus überprüft werden.

4.4.1 Generelle Informationen der Slaves im Bus ermitteln

Bei dieser Variablengruppe wird in der Zugriffsvariablen Index stets der Index des anzusprechenden Slave-Geräts angegeben (von 0 beginnend). Dies ist auch der Index, der den Teilnehmern beispielsweise im Kommando "ethercat diag" zugeordnet ist und die Reihenfolge im Bus repräsentiert. Im SubIndex wird eine Funktionsnummer angegeben. Diese Funktionen werden nachfolgend beschrieben.

AccessType: Lesen
DataType: entsprechend dem Datentyp laut folgender Tabelle

4.4.1.1 Dev.# 700 hex

Über diese Device-Nummer können Informationen über die angeschlossenen Slaves gelesen werden. Schreibzugriffe sind hier nicht möglich.

SubIndex	Name Datentyp	Beschreibung
0	EXIST integer 32 bit	Mit dieser Variablen kann ermittelt werden, ob ein Slave im Bus existiert. Wird 0 zurückgeliefert, so ist der Teilnehmer nicht vorhanden. Der Index kann Werte von 0 bis 511 annehmen. Die Busteilnehmer werden ab 0 fortlaufend gelistet. Wenn bei den nachfolgenden Funktionsnummern ein nicht existierender Slave angesprochen wird, liefert die Aufruf-Funktion einen Fehler zurück. Der ermittelte Parameter ist dann ungültig.
1	ALIAS short int 16 bit	Dieser Aufruf liefert die zugewiesene Alias-Adresse eines Slaves zurück. Dieser Wert kann z.B. für die Verifikation des Busaufbaus bzw. zur Kontrolle der Busverkabelung herangezogen werden.
3	Vendor Id. integer	Dieser Aufruf liefert den Hersteller-Code des entsprechenden Slave-Moduls zurück. Dieser Wert kann z.B. für die Verifikation des Busaufbaus herangezogen werden.
4	Product Code integer 32 bit	Dieser Aufruf liefert den Produkt-Code des entsprechenden Slave-Moduls zurück. Dieser Wert kann z.B. für die Verifikation des Busaufbaus herangezogen werden.
8	Slave Online integer 32 bit	Zeigt an, ob ein Slave online ist. In diesem Wert steht der Inhalt des AL-Statusregisters des jeweiligen Slaves. Mit dem Wert -1 wird angezeigt, dass der Wert aktuell nicht abrufbar ist.
64	Name DataBuffer	Mit Hilfe dieser Variablen kann der Modulname des jeweiligen Slaves abgerufen werden. Der Zugriff auf diese Variable muss mit der DLL-Funktion rdOptionBuf erfolgen, wobei der Buffer eine Größe von mindestens 64 Byte haben muss und der Name als nullterminierter String zurückgeliefert wird.

4.4.2 EtherCAT-Master-Informationen lesen

Beim Starten der Steuerung wird der Aufbau des angeschlossenen EC-Busses ermittelt. Unter anderem sind dies die Anzahl der erkannten Slaves und die erwarteten und tatsächlichen Werte für die Working Counter bei Ein- und Ausgangsinformationen.

4.4.2.1 Dev.# 900 hex

Über diese Device-Nummer können Informationen über den Zustand der Working Counter gelesen werden. Schreibzugriffe sind hier nicht möglich.

Index	SubIndex	Name Datentyp	Beschreibung
1	0	32 bit Integer	Expected Working Counter Domain 0 (Inputs)
1	1	32 bit Integer	Expected Working Counter Domain 1 (Outputs)
1	2	32 bit Integer	Expected Working Counter Domain 0+1 (Inputs + Outputs) Lower 16 bit = WC Inputs Higher 16 bit = WC Outputs
2	0	32 bit Integer	Expected Working Counter Domain 0 (Inputs)
2	1	32 bit Integer	Expected Working Counter Domain 1 (Outputs)
2	2	32 bit Integer	Expected Working Counter Domain 0+1 (Inputs + Outputs) Lower 16 bit = WC Inputs Higher 16 bit = WC Outputs
3	-	32 bit Integer	Number of detected slaves

Die Werte der Expected Working Counter werden einmalig ermittelt und ändern sich während der Session nicht mehr. Die aktuellen Werte der Working Counter werden jeweils aktuell angepasst, wenn sich der Busaufbau ändert. Die erwarteten Werte werden jedoch nicht überschritten.

Die Anzahl der aktuellen Slaves wird jeweils aktuell angepasst. Allerdings kann hier eine gewisse Verzögerung durch das Abscannen des Busaufbaus auftreten.

4.4.3 SDOs der Busteilnehmer lesen und schreiben

Bei Slave-Teilnehmern mit einem SDO-Objekt-Interface, welches SDO-Zugriffe erlaubt, kann mit Hilfe des universellen Objekt-Interface lesend und schreibend auf diese SDOs zugegriffen werden. Die Zugriffe können sowohl aus der SAP-Programmierung als auch aus der PCAP-Programmierung erfolgen. Die entsprechenden Objekt-Deskriptoren (bzw. AT-Variable) müssen vom Anwender anhand der Produktdokumentation und der Position der jeweiligen Slave-Module im EtherCAT-Bus erstellt werden.

Falls ein Teilnehmer keine SDO-Kommunikation erlaubt, liefert die Funktion einen Fehler zurück. Der Parameter-Wert bei Lesevorgängen ist in diesem Fall ungültig. Falls auf nicht existierende SDOs oder mit falscher Datentyp-Zuordnung zugegriffen wird, wird ebenfalls ein Fehler zurückgegeben. Im fwsetup-Terminal-Bildschirm kann per dmesg weitere Information über den aufgetretenen Fehler ermittelt werden.

Beim Beschreiben von SDOs werden die geschriebenen Werte im Allgemeinen resident auf dem Slave-Modul gespeichert. Manche Module stellen eine Checksumme zur Verfügung, in welcher sich eine Änderung eines residenten Parameters niederschlägt. In diesen Fällen ist die Überprüfung der korrekten Initialisierung des Slave-Moduls möglich.

Achtung: SDOs dürfen nicht zyklisch in Programmen beschrieben werden, da hierbei in den Flash-Speicher der EtherCAT-Slave-Module geschrieben wird. Da die Anzahl der Schreibvorgänge für diesen Speicher begrenzt ist, können die Module ansonsten schon nach kurzer Zeit irreparabel beschädigt werden.

Die jeweiligen Objekt-Deskriptoren sind wie folgt zu initialisieren:

AccessType: Lesen oder Schreiben
 DataType: entsprechend dem Datentyp laut der Produktdokumentation des Slave-Moduls
 BusNumber: EDBus (600 dez.)
 DeviceNumber: Index des Slave-Moduls im EtherCAT-Bus (laut ethercat diag)
 Index: SDO-Index laut der Produktdokumentation des Slave-Moduls
 SubIndex: SDO-SubIndex laut der Produktdokumentation des Slave-Moduls

Auf diese Weise lassen sich z.B. auch Revisionsnummern und weitere Eigenschaften von EtherCAT-Slave-Modulen überprüfen, ergänzend zu den Bus-Überprüfungen laut Kapitel 4.4.1.

4.5 Manuelles Lesen und Beschreiben von SDOs im Slave-Verzeichnis

Diese Methode ist nicht zu verwechseln mit den Mechanismen in Kapitel 4.4.2. Hier wird beschrieben, wie der Anwender in der Einrichtungsphase per manueller Bedienung des Systems auf interne Variable von EtherCAT-Slave-Modulen zugreifen kann. Diese Bedienung erfolgt im Terminal-Fenster des Programms fwsetup.

```
ethercat upload -p? 0xAAAA B
```

```
ethercat download -p? 0xAAAA B Wert
```

```
Option -t
```

Siehe hierzu auch Kapitel 6.3

Erläuterungen zu obigen Kommandos:

-p?	Das ? repräsentiert den Index des Slave-Moduls im EtherCAT-Bus, der z.B. per ethercat diag angezeigt wird.
0xAAAA	hexadezimale Objektadresse
B	Subindex des Objekts; auch hier ist zu beachten, ob die Angabe dezimal oder hexadezimal vorliegt. Hexadezimalzahlen können auch mit vorangestelltem 0x eingegeben werden.
Wert	Zahlenwert, der gegebenenfalls geschrieben werden soll; dieser kann dezimal oder hexadezimal (mit vorangestelltem 0x) angegeben werden.

4.6 Inbetriebnahme von Achsen in mcfg.exe

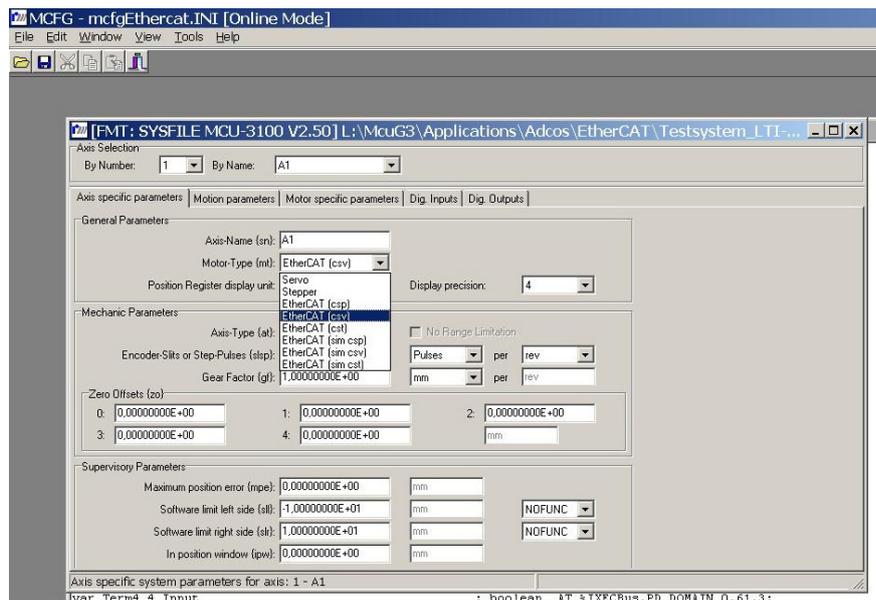
4.6.1 Vergabe der Motorachsen

Zunächst müssen die Achsen des Systems den vorliegenden Achstypen im EtherCAT-Bus zugeordnet werden. Dies erfolgt auf der Seite der Systemdaten, und zwar in der Auswahl „Motor Types {mt}“ auf der Registerkarte „Axis Specific Parameters“. Für EtherCAT-Achsen stehen hier die Einträge „EtherCAT (csp)“, „EtherCAT (csv)“ und „EtherCAT (cst)“ zur Verfügung.

Hierbei ist die jeweilige Fähigkeit der Geräte bzw. der Implementierung in der Betriebssystem-Software der APCI-8008 zu berücksichtigen.

Die Reihenfolge der Vergabe muss der Reihenfolge der zugeordneten Achsen im Bus entsprechen, darf aber in der Vergabe nicht lückenlos sein. Somit ist es auch möglich, EtherCAT-Achsen gemischt mit Standard-Servo- oder Stepperachsen zu betreiben.

Abbildung 4-6: Zuweisung einer EtherCAT-Achse an die erste Achse in mcfg



4.6.2 Einstellung der Istwert-Auflösung

Die Auflösung des Encoder-Systems „Encoder Slits or Step Pulses {slsp}“ muss auch bei EtherCAT-Achsen der Dokumentation entnommen oder manuell vom Bus gelesen und hier eingetragen werden. Die Zählereinheit von {slsp} ist in der Regel „Pulses“, weil die eventuelle elektronische Vervielfachung stets im EtherCAT-Modul stattfindet. Im DS-402-Profil ist die Auflösung des Gebersystems im Objekt 0x608F:1 zu finden. Hierbei handelt es sich um eine 32-Bit-Ganzzahlvariable.

Diese Variable kann laut Kapitel 4.4 z.B. mit folgendem Kommando im fwsetup-Monitor-Screen gelesen werden:

```
ethercat upload -p2 -tint32 0x608f 1
```

-p2 bedeutet, dass vom Slave mit dem Index 2 gelesen wird.

-tint32 bedeutet, dass der Datentyp int32 erwartet wird.

Hinweis: Man muss unterscheiden zwischen der realen Auflösung des Lagegebers und der vom EtherCAT-Slave zur Verfügung gestellten Auflösung. Diese hat im Allgemeinen nicht den gleichen Wert. Genauen Aufschluss, wie die Auflösung zu ermitteln ist, muss die Herstellerdokumentation liefern.

4.7 Sonstige EtherCAT-spezifische Handhabung des Systems

4.7.1 Restart der Karten per BootFile

Seit September 2021 (mcug3.dll V2.5.3.133) ist es möglich, das gesamte System mit Hilfe der DLL-Funktion BootFile () neu zu starten. Hierbei wird ein Reset der Karte durchgeführt, so wie beim Hochfahren des PCs oder mit dem Reset-Button in fwsetup. Dieser Schritt darf aber nur erfolgen, wenn keine anderen Programme auf das System zugreifen.

Der Reboot-Vorgang kann ziemlich lange dauern. Die Arbeit mit der Karte kann erst fortgesetzt werden, wenn InitmcuSystem? den Wert 0 liefert.

4.7.2 Übertragen des dmesg-Protokolls ins Windows-Dateisystem

Ab mcug3.dll V2.5.3.133 und rwmos-Version V2.5.3.159 ist es möglich, per DLL-Funktion das dmesg-Protokoll der Achsensteuerungskarte zu lesen. Der Protokollinhalt auf der Karte wird dadurch verworfen. Somit wird dem Benutzer die Information zur Verfügung gestellt, die ansonsten im fwsetup-Monitor beim Kommando dmesg angezeigt wird. Diese Information kann für die Systemdiagnose verwendet werden.

DLL-Funktion in C:

```
int rdDmesg (char *filename);
```

Parameter: Dateiname mit optionaler Laufwerks- und Pfadangabe

Rückgabewerte:

- >= 0 bei Erfolg; die Anzahl der gelesenen Bytes wird zurückgegeben.
- 1 Funktion in RWMOS nicht verfügbar
- 33 RWMOS ist keine EtherCAT-Variante.
- 38 keine Achsensteuerungskarte im System erkannt
- 53 Fehler beim Öffnen der angegebenen Datei in Windows
- 54 interner Fehler / Speicherproblem

Hinweis: Die einzelnen Zeilen sind per LineFeed getrennt, nicht durch CarriageReturn.

5 Antriebsspezifische Eigenschaften

Bei der APCI-8008-EC gibt es für jedes enthaltene Antriebssystem spezifische Einstellungen, die bei der Verwendung eines Antriebs berücksichtigt werden müssen.

So gibt es keine gemeinsam genutzten digitalen Ein- und Ausgänge mehr. Stattdessen werden die auf einem Antrieb verfügbaren digitalen Ein- und Ausgänge im digi- bzw. digo-Wort der jeweiligen Achse abgebildet. Wie diese Ein-/Ausgänge den einzelnen Bits zugeordnet sind, wird nachfolgend antriebsspezifisch dokumentiert. Unabhängig davon kann auf alle Bits der einzelnen EtherCAT-Register über das universelle Objekt-Interface zugegriffen werden, soweit diese zyklisch per PDO transferiert werden (siehe Kapitel 4.1.2.2.2).

Des Weiteren werden die Touch-Probe Functions, welche einen Positions-Latch erlauben, konfiguriert und der Latch-Funktionalität der APCI-8008-EC zugeordnet. Somit ist im Allgemeinen ein Positions-Latch per digitalem Eingang und per Nullspursignal wie bei den bisherigen Karten möglich.

Hinweis: Eine Signal-Invertierung / Flanken-Invertierung bei Touch Probe-Signalen ist nicht wie bisher möglich!

5.1 Omron-Servoantriebe

5.1.1 Accurax G5-Antriebe

5.1.1.1 Betriebsarten

Dieses Gerät kann in den Betriebsarten csv, csp und cst betrieben werden.

5.1.1.2 Digitale Eingänge

Signal	Bit# digi-Wort	

5.1.2 Digitale Ausgänge

Signal	Bit# digo-Wort

5.1.3 Touch Probe-Funktion / Latch-Funktionalität

Touch Probe channel	Source / Trigger	Flanke	Modus

5.1.4 Sysmac 1S-Servoantriebe

5.2 Panasonic MADHT1505

5.2.1 Betriebsarten

Dieses Gerät kann in den Betriebsarten csv, csp und cst betrieben werden.

5.2.2 Digitale Eingänge

Signal	Bit# digi-Wort	
NOT	0	negative limit switch
POT	1	positive limit switch
HOME	2	home switch
SI-MON1	3	Latch Input EXT1
SI-MON2	4	General Purpose
SI-MON3	5	General Purpose
SI-MON4	6	General Purpose
SI-MON5	7	Emergency Stop E-STOP / General Purpose
INP	8	Positioning Complete

5.2.3 Digitale Ausgänge

Signal	Bit# digo-Wort
EX-OUT1	0

5.2.4 Touch Probe-Funktion / Latch-Funktionalität

Touch Probe channel	Source / Trigger	Flanke	Modus
1	EXT1	positiv	Continuous Triggering
2	Z-Phase	positiv	Trigger First Event

Das Flag NDX (Bit # 16) im digi-word kann bei diesem Gerät nicht verwendet werden, weil der EtherCAT-Bus kein entsprechendes Signal zur Verfügung stellt. Zur Referenzierung kann nur die gelatchte Position herangezogen werden.

5.2.5 Quick Stop Deceleration (Object 6085 hex)

Hier muss per SDO-Zugriff ein sinnvoller Beschleunigungswert in Counts / s² eingetragen werden, welcher gewährleistet, dass die Achse im Fehlerfall nach einer angemessenen Zeit zum Stillstand kommt. Der Standardwert dieses Objekts (1 000 000) ist bei hoher Geberauflösung viel zu niedrig und kann bewirken, dass die Achse viel zu lange braucht, um anzuhalten.

Dieser Wert kommt zum Tragen, wenn der EC-Bus während des Betriebs hinter dem entsprechenden Slave aufgetrennt wird. Dann initiiert die Achsensteuerungskarte einen Quick-Stop (im csv- und cst-Betrieb).

6 Konfiguration und Diagnose in der Konsole der APCI-8008

6.1 Anpassung von herstellerspezifischen XML-Dateien für die Verwendung mit der APCI-8008

6.1.1 SM=??

Diese Angabe definiert den SyncManager, mit dem PDOs zyklisch übertragen werden. Register, die vom EtherCAT-Slave gelesen werden, sind dem SyncManager „3“ zuzuordnen; Register, welche auf den EtherCAT-Slave geschrieben werden sollen, dem SyncManager „2“.

6.1.2 PDOs ergänzen

Hierzu sind genaue Kenntnisse des herstellerspezifischen Objektverzeichnisses notwendig. Objekte, die auf dem Slave resident gespeichert werden, dürfen auf keinen Fall zyklisch geschrieben werden, da die Baugruppe ansonsten innerhalb kurzer Zeit beschädigt würde.

6.2 EtherCAT-Modul neu starten

RWMOS anhalten mit:
`rwmoslkm stop`

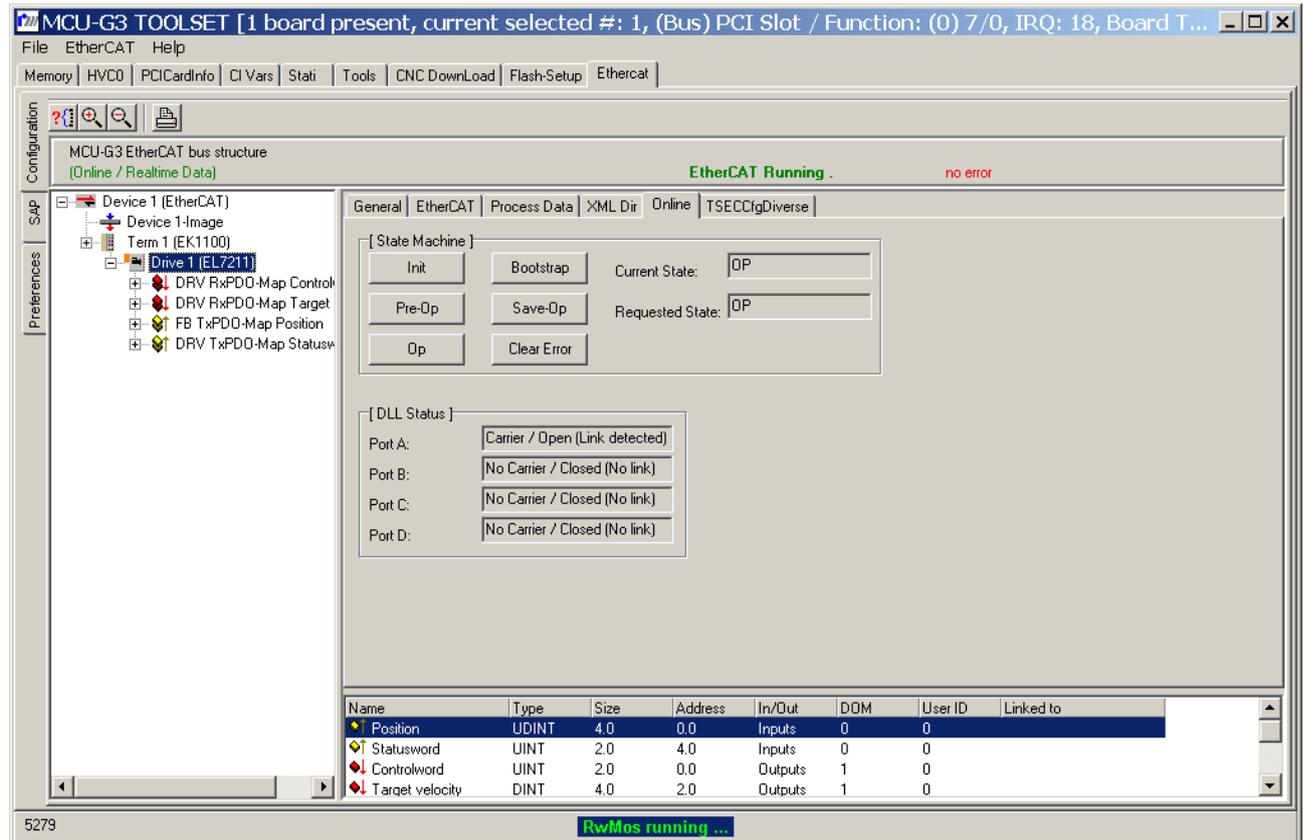
Neustart des EtherCAT-Moduls:
`/etc/init.scripts/ethercat restart`

Danach muss RWMOS wieder gestartet werden:
`rwmoslkm start`

6.3 EtherCAT-SDOs

Die Beschreibung von EtherCAT-SDOs (Service Data Objects) der jeweiligen Slave-Baugruppe ist der entsprechenden Produktdokumentation zu entnehmen. Ein SDO ist durch seinen Index und SubIndex definiert. Das Beschreiben von SDOs ist im Operational State der Slave-Baugruppe unter Umständen nicht möglich. In diesem Fall kann der PreOperational State durch Klicken auf den Button Pre-Op in der gezeigten Ansicht von fwsetup erzwungen werden:

Abbildung 6-1: Button Pre-Op



Das RWMOS-Modul der Achsensteuerungskarte kann innerhalb von Linux angehalten oder neu gestartet werden.

Starten und Stoppen des RWMOS-Moduls:

```
rwmoslkm stop
```

In diesem Fall ist hinterher ein Neustart notwendig, der folgendermaßen initiiert werden kann:

```
rwmoslkm start
```

Ein hexadezimaler Index wird nachfolgend als 0xXXXX dargestellt; ein SubIndex als YY.

Das Lesen und Beschreiben von SDOs wird mit den Kommandos

```
ethercat upload
```

und

```
ethercat download
```

durchgeführt.

Lesen und Anzeigen eines SDO:

```
ethercat upload -p? 0xXXXX YY
```

Das ? stellt hier den Index der anzusprechenden Slave-Komponente dar.

Beispiel für Lesen und Anzeigen:

```
[root@MCU:~]# ethercat upload -p1 0x1c13 1  
0x1a00 6656
```

Der gelesene Wert wird in hexadezimaler und dezimaler Form ausgegeben.

Beispiel für Beschreiben des SDO:

```
[root@MCU:~]# ethercat download -p1 0x1c13 1 0x1a00
```

Hier erfolgt eine Anzeige nur im Fehlerfall.

Optional kann oder muss manchmal ein Datentyp mit dem Parameter -t angegeben werden.

Beispiel:

```
[root@MCU:~]# ethercat download -p1 0x1c13 1 0x1a00 -tint32
```

Eine Liste der verfügbaren Datentypen kann durch Aufruf des Kommandos ohne Parameter angezeigt werden:

```
ethercat upload  
oder  
ethercat download
```

7 Anhang

7.1 Fallstricke bei der Einrichtung des Systems

- Eine oder mehrere Achsen gehen nicht in den Operational State; die EtherCAT-Bus-LED blinkt. Ursache kann hier sein, dass die in mcfg definierten EtherCAT-Achsen nicht mit den tatsächlich im Bus vorhandenen Achsen übereinstimmen. Die nicht zugeordneten Achsen werden dann als nicht gültig initialisiert.
In diesem Fall werden im dmesg-Protokoll Meldungen in folgender Form angezeigt:
EtherCAT WARNING 0-0: CoE Emergency Request received:
Error code 0xFF1E, Error register 0x80, data
- Bei der Einrichtung der Systemdaten muss auf der Registerkarte „Motor specific parameters“ auf die Einstellung in „Maximum / Minimum output Voltage“ geachtet werden. Bei Nicht-EtherCAT-Systemen steht hier gewöhnlich ein Wert von +/- 10 Volt. Bei EtherCAT-Systemen hat dieser Wert eine gänzlich andere Bedeutung, z.B. in der Einheit digits/Sekunde, und kann Werte von z.B. +/-2e23 annehmen. Dies entspricht +8.388.607 und -8.388.608.
- Beim Generieren der SAP-Ressourcenbezeichner werden Elemente mit unerlaubten Domains generiert. Hier ist darauf zu achten, dass sich alle EtherCAT-Slaves bei der Generierung der SAP-Bezeichner im Operational-State befinden.
- In den SAP-Variablendeklarationen enthalten die Namen der AT-Spezifizierer sonderbare, nichtssagende Bezeichnungen, teilweise mit dem Wort „unknown“.
Ursache hierfür ist, dass in den mitgelieferten XML-Dateien der Hersteller nicht alle von der Achsensteuerungskarte benötigten PDOs beschrieben sind. Gegebenenfalls muss eine speziell ergänzte XML-Datei verwendet oder eine bestehende XML-Datei modifiziert werden.
- Beim Start wird im Monitor-Screen von fwsetup folgende Fehlermeldung angezeigt:
Failed to calculate Bus Topologie.
Ursache hierfür kann ein Fehler in der Busverkabelung sein, wenn z.B. ein Buskabel am EtherCAT-Ausgang anstatt am EtherCAT-Eingang eines Slaves angeschlossen wurde.

7.2 Implementierte Geräte

7.2.1 E/A-Module

Beckhoff	EL1014
Beckhoff	EL1804
Beckhoff	EL1809
Beckhoff	EL2004
Beckhoff	EL2202
Beckhoff	EL2809
Beckhoff	EL2828
Beckhoff	EL3064
Beckhoff	EL3102
Beckhoff	EL3255
Beckhoff	EL4002
Beckhoff	EL4132
Beckhoff	EL5101
Beckhoff	EL7211
Beckhoff	EK1814
Beckhoff	EP2316
ADDI-DATA	MSX-EC-1730-16
ADDI-DATA	CS-MAT-3001-16-EC

7.2.2 Antriebe

CopleyControl		AEP_180_18
CopleyControl		XEL_230_13
HIWIN		D1N
KOLLMORGEN		AKD
Lti		ServoJ
Omron	R88D	KN01H_ECT
		KN04H_ECT
		KN08H ECT
		KN15F ECT
		KN20F ECT
		KN30F ECT
		1SN01H ECT
Rexrot IndraDriveCS		HSC01_1E_W0018
Beckhoff	EL7211	
Panasonic	MADHT1505	
Panasonic	MADHT1507	
Panasonic	MADHT2510	

Weitere Geräte können jederzeit kurzfristig hinzugefügt werden.

7.3 Abbildungsverzeichnis

Abbildung 2-1: Anschlüsse APCLe-8008-EC	6
Abbildung 2-2: Anschlüsse BOB-3100	7
Abbildung 2-3: Anschlüsse APCI-8008	8
Abbildung 4-1: EtherCAT Configuration	11
Abbildung 4-2: EtherCAT Configuration: Baumform	12
Abbildung 4-3: EtherCAT SAP	13
Abbildung 4-4: Bildschirmtext-Ausgabe EtherCAT SAP	15
Abbildung 4-5: EtherCAT Diagnostics	16
Abbildung 4-6: Zuweisung einer EtherCAT-Achse an die erste Achse in mcfg	23
Abbildung 6-1: Button Pre-Op	28

7.4 Tabellenverzeichnis

Tabelle 4-1: Bedeutung der Symbole auf der Seite [EtherCAT][Configuration]	12
Tabelle 4-2: Bedeutung der Symbole auf der Seite [EtherCAT][SAP]	14