

# **CANopen Einstiegskit KIT-152**

**Hinweise zur Erstinbetriebnahme**

**Auflage Mai 2011**

Im Buch verwendete Bezeichnungen für Erzeugnisse, die zugleich ein eingetragenes Warenzeichen darstellen, wurden nicht besonders gekennzeichnet. Das Fehlen der © Markierung ist demzufolge nicht gleichbedeutend mit der Tatsache, dass die Bezeichnung als freier Warenname gilt. Ebenso wenig kann anhand der verwendeten Bezeichnung auf eventuell vorliegende Patente oder einen Gebrauchsmusterschutz geschlossen werden.

Die Informationen in diesem Handbuch wurden sorgfältig überprüft und können als zutreffend angenommen werden. Dennoch sei ausdrücklich darauf verwiesen, dass die Firma SYS TEC electronic GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf den Gebrauch oder den Inhalt dieses Handbuches zurückzuführen sind. Die in diesem Handbuch enthaltenen Angaben können ohne vorherige Ankündigung geändert werden. Die Firma SYS TEC electronic GmbH geht damit keinerlei Verpflichtungen ein.

Ferner sei ausdrücklich darauf verwiesen, dass SYS TEC electronic GmbH weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgeschäden übernimmt, die auf falschen Gebrauch oder falschen Einsatz der Hard- bzw. Software zurückzuführen sind. Ebenso können ohne vorherige Ankündigung Layout oder Design der Hardware geändert werden. SYS TEC electronic GmbH geht damit keinerlei Verpflichtungen ein.

© Copyright Mai 2011 SYS TEC electronic GmbH, D-07973 Greiz.

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form ohne schriftliche Genehmigung der Firma SYS TEC electronic GmbH unter Einsatz entsprechender Systeme reproduziert, verarbeitet, vervielfältigt oder verbreitet werden.

Informieren Sie sich:

	EUROPA	NORD AMERIKA
Adresse:	SYS TEC electronic GmbH August-Bebel-Str. 29 D-07973 Greiz GERMANY	PHYTEC America LLC 255 Ericksen Avenue NE Bainbridge Island, WA 98110 USA
Angebots Hotline:	+49 (3661) 6279-0 <a href="mailto:info@systec-electronic.com">info@systec-electronic.com</a>	+1 (800) 278-9913 <a href="mailto:order@phytec.com">order@phytec.com</a>
Technische Hotline:	+49 (3661) 6279-0 <a href="mailto:support@systec-electronic.com">support@systec-electronic.com</a>	+1 (800) 278-9913 <a href="mailto:support@phytec.com">support@phytec.com</a>
Fax:	+ 49 (3661) 6279-99	+1 (206) 780-9135
Web Seite:	<a href="http://www.systec-electronic.com">http://www.systec-electronic.com</a>	<a href="http://www.phytec.com">http://www.phytec.com</a>

5. Auflage Mai 2011

---

<b>KIT-152 - CANopen Einstiegskit .....</b>	<b>1</b>
<b>1 Lieferumfang .....</b>	<b>2</b>
1.1 Hardware .....	2
1.2 Dokumentation: .....	2
1.3 Software: .....	2
<b>2 Aufbau und Inbetriebnahme der Hardware.....</b>	<b>3</b>
2.1 Inbetriebnahme DIPmodul F40.....	3
2.2 Inbetriebnahme CANopen IO-Box .....	4
2.3 Anschluss der Spannungsversorgung.....	5
2.4 Einstellung RS 232 Schnittstelle.....	6
2.5 Verdrahtung des CAN-Netzwerkes.....	7
2.5.1 Anschluss des CAN-Kabelsatzes .....	8
<b>3 Installation der Tool Software .....</b>	<b>11</b>
3.1 Installation der Entwicklungsumgebung Softune .....	11
3.1.1 Installation des Fujitsu Flash MCU Programmers.....	12
3.2 Installation des Accemic MDE Debugger.....	13
3.3 Installation der CANopen Einstiegskit Software .....	15
3.3.1 Installation der CANopen Configuration Suite Demo..	16
3.3.2 Installation der ODBuilder Demo .....	16
3.3.3 Installation der USB-CANmodul Utility Disk.....	17
<b>4 Inbetriebnahme der CANopen Software.....</b>	<b>19</b>
4.1 Weitere Hinweise .....	25
4.2 Einschränkungen der CANopen Kit Software .....	26
4.3 Hinweise zu den Demo Applikationen .....	26
4.3.1 Die Initialisierungsstruktur.....	27
4.3.2 Die Variablen-tabelle.....	27
4.3.3 Die PDO Tabelle.....	28
4.3.4 Verwendung der PDO-Variablen.....	30
4.3.5 Die NMT Callback-Funktion .....	32
<b>Index.....</b>	<b>35</b>

## **Bild- und Tabellenverzeichnis**

Bild 1:	Basisplatine für DIPmodul F40 .....	3
Bild 2:	Jumperbelegung für die DIP-Schalter an der CANopen IO-Box4	
Bild 3:	Anschluss der Spannungsversorgung an die Basisplatine.....	5
Bild 4:	prinzipielle CAN-Bus Anschaltung (ohne galvanische Entkopplung) .....	8
Bild 5:	Einleitung des Burn-In-Modus beim DIPmodul F40 .....	13
Tabelle 1:	Pinbelegung der Spannungsversorgung an der CANopen IO-Box.....	5
Tabelle 2:	RS 232 Verbindungskabel .....	7
Tabelle 3:	Verdrahtung des CAN-Busses .....	7
Tabelle 4:	empfohlene Leitungsparameter für das CAN-Kabel .....	8
Tabelle 5:	festes PDO Mapping nach DS-401 V2.1 .....	30

## **KIT-152 - CANopen Einstiegskit**

Das CANopen Einstiegskit ist der kostengünstige Einstieg in die Entwicklung eigener CANopen Produkte und demonstriert die Möglichkeiten der CANopen Software und CAN-Hardware. Es enthält alle Komponenten eines CANopen Netzwerkes, wie Mikrokontroller-Baugruppen, einen Adapter vom PC zum CAN-Bus, CANopen IO-Box, CANopen Slave und Master Software und CANopen Slave und Master Software für Windows 2000/XP und ein CANopen Konfigurationstool für das Objektverzeichnis. Die Softwarekomponenten sind voll lauffähige Demos mit begrenztem Funktionsumfang. Die Anwenderschnittstelle ist identisch zur Vollversion.

Zur Nutzung des CANopen Einstiegskits benötigen Sie folgende Voraussetzungen:

- MS-Windows 2000/XP oder neuer
- 1 freie COM-Schnittstelle
- 1 freie USB-Schnittstelle
- Microsoft Visual C/C++ ab Version 7.1

## 1 Lieferumfang

### 1.1 Hardware

1x MM-217	DIPmodul F40 mit Basisplatine
1x 3204000	USB-CANmodul1
1x phyPS-451-Y	CANopen IO-Box
1x WK054	CAN Kabelsatz
1x WK800	RS 232-Kabel
1x SV007	Netzteil 9V/500mA

### 1.2 Dokumentation:

1x L-1000	Basisplatine Developmentboard DIPmodul F40
1x L-1062	CANopen ChipF40 System Manual
1x L-487	USB-CANmodul Hard- und Software Manual
1x L-1048	IO-Box System Manual
1x L-1020	CANopen Software Manual
1x L-1078	CANopen Einstiegskit KIT-152, Hinweise zur Erstinbetriebnahme (dieses Manual)
1x L-1055	CANopen Configuration Manager User Manual
1x L-1056	CANopen Device Monitor User Manual
1x L-1022	ODBuilder Manual

### 1.3 Software:

1x SO-387	USB-CANmodul Utility Disk
1x SO-1018	CANopen Kit-Library für KIT-152
1x SO-1007	ODBuilder Demo
1x SO-1008	CANopen-Demo für Windows
1x SO-1047	CANopen Configuration Suite

## 2 Aufbau und Inbetriebnahme der Hardware

### 2.1 Inbetriebnahme DIPmodul F40

Entnehmen Sie die Einzelkomponenten der Verpackung. Achten Sie dabei auf eine ESD-gerechte Umgebung.

Die Basisplatine ist bereits entsprechend dem aufgesetztem DIPmodul F40 eingestellt worden. Prüfen Sie dennoch die Einstellungen der Jumper auf der Basisplatine, die weiter unten beschrieben werden.

Das DIPmodul F40 wird auf die Basisplatine auf den 40 poligen Steckplatz (siehe Bild 1) aufgesteckt.

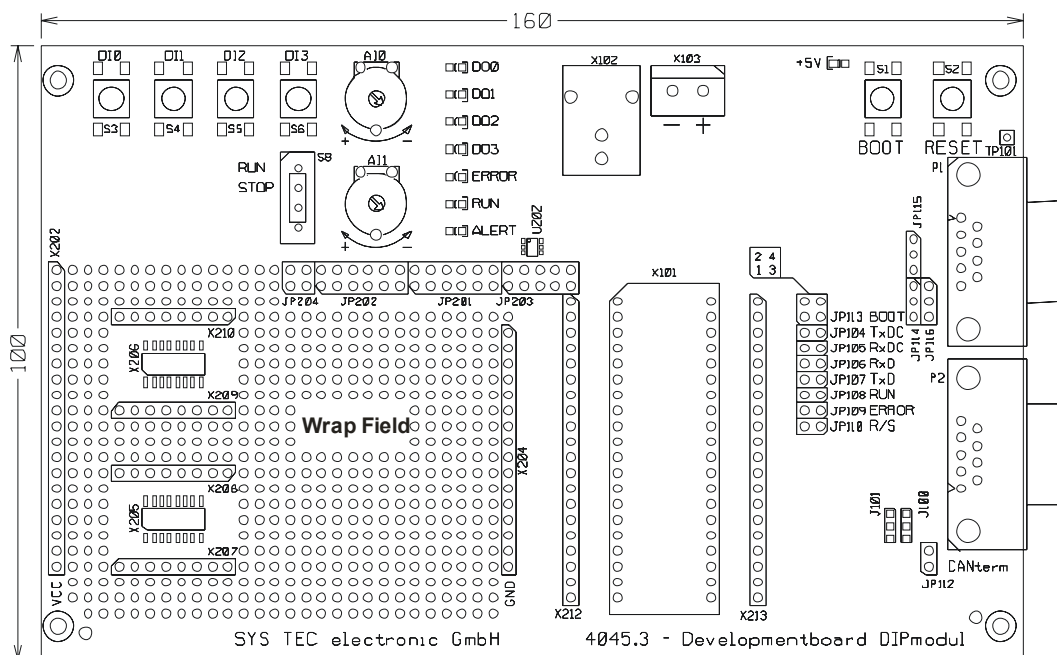


Bild 1: Basisplatine für DIPmodul F40

<b>Jumper</b>	<b>Funktion</b>	<b>Zustand</b>
JP104	TxD CAN	offen
JP105	RxD CAN	offen
JP106	RxD RS232	geschlossen
JP107	TxD RS232	geschlossen
JP108	Run-LED	offen
JP109	Error-LED	offen
JP110	R/S	offen
JP112	CAN Termination	nach Bedarf
JP113	Boot	(1-3) (2-4)
JP114	onboard RS232 enable	(1-2)
JP115	onboard RS232 enable	(1-2)
JP116	onboard RS232 enable	(1-2)

Die genaue Bedeutung der Jumper auf der Basisplatine entnehmen Sie bitte dem Manual für die Basisplatine (*L-1000*).

## 2.2 Inbetriebnahme CANopen IO-Box

Die CANopen IO-Box enthält im Gehäuse ein fertig programmiertes DIPmodul F40 mit CANopen Software.

Am DIP-Schalter der CANopen IO-Box wird die Knotenadresse und die Bitrate eingestellt. Prüfen Sie bitte, ob diese so eingestellt sind, wie es in *Bild 2* beschrieben ist. Ist dies nicht der Fall, dann ändern Sie bitte die Einstellung nach *Bild 2*.

<b>DIP1</b>	<b>DIP2</b>	<b>DIP3</b>	<b>DIP4</b>	<b>DIP5</b>	<b>DIP6</b>	<b>DIP7</b>	<b>DIP8</b>
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF

*Bild 2: Jumperbelegung für die DIP-Schalter an der CANopen IO-Box*

Die genaue Bedeutung der Jumper entnehmen Sie bitte dem Manual für den CANopen ChipF40 (*L-1062*).



## 2.3 Anschluss der Spannungsversorgung

Auf der Basisplatine für das DIPmodul F40 befindet sich der Anschluss der Spannungsversorgung an der Kleinspannungs-Buchse X102. Verwenden Sie bitte dafür das mitgelieferte Netzteil SV007 mit 9V/500mA. Die Leuchtdiode D102 (grün, neben dem Boot-Taster, *siehe Bild 1*) zeigt an, dass die Spannung an der Basisplatine für das DIPmodul F40 anliegt.

### Hinweis:

Vermeiden Sie, das DIPmodul F40 bzw. die Jumperbelegung bei anliegender Spannung zu wechseln!

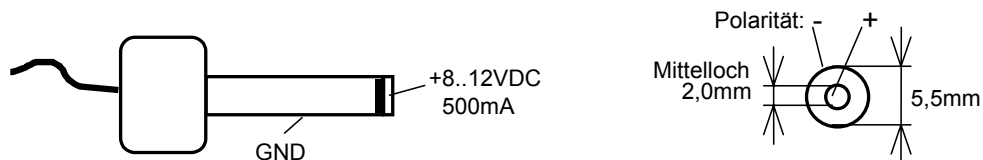


Bild 3: Anschluss der Spannungsversorgung an die Basisplatine

Die Spannungsversorgung der CANopen IO-Box wird über die CAN Buchse angeschlossen. Verwenden Sie bitte dafür den mitgelieferten Abschlusswiderstand mit den 2 Kabeln. Verschrauben Sie diese 2 Kabelenden mit der Schraubklemme X103 auf der Basisplatine des DIPmodul F40. *Tabelle 1* zeigt die Anschlussbelegung der Spannungsversorgung. Verbinden Sie den SUB-D 9 Stecker mit einer freien Buchse am Ende des CAN-Bus Kabels.

Pin am SUB-D 9	Bezeichnung	Farbe des Kabels am Abschlusswiderstand	Pin an X103 auf Basisplatine
3	GND	schwarz	-
9	VCC (+8..12VDC)	rot	+

Tabelle 1: Pinbelegung der Spannungsversorgung an der CANopen IO-Box

## 2.4 Einstellung RS 232 Schnittstelle

Die serielle RS 232 Schnittstelle dient zur Kommunikation mit dem PC und zum Download der Software auf das Microcontroller Board. Alle printf()-Anweisungen werden innerhalb der CANopen Kit Software auf diese Schnittstelle ausgegeben. Die Verbindung erfolgt mit einem RS 232 Verlängerungskabel, das im Lieferumfang des CANopen Einstiegskit enthalten ist. Verbinden Sie das Kabel mit der oberen D-SUB Buchse (P1) auf der Basisplatine des DIPmodul F40 und Ihrem PC.

Für die Darstellung der Ausgaben der CANopen Software kann das Tool HyperTerminal verwendet werden, welches im Windows Betriebssystem unter *Programme* → *Zubehör* → *Kommunikation* zu finden ist. Erstellen Sie im HyperTerminal eine Verbindung mit der verwendeten COM-Schnittstelle mit folgenden Einstellungen: 9600 Bit/s, 8 Datenbits, keine Parität, 1 Stopbit und keine Flusststeuerung.

Die benötigten Signalleitungen sind *TxD*, *RxD* und *GND*.

Steckerbelegung bei IBM PC (9 poliger Stecker)

RxD Pin 2 Empfangsdatenleitung

TxD Pin 3 Sendedatenleitung

GND Pin 5 Ground (GND)

Steckerbelegung bei IBM PC (25 poliger Stecker)

TxD Pin 2 Sendedatenleitung

RxD Pin 3 Empfangsdatenleitung

GND Pin 7 Ground (GND)

Steckerbelegung auf der Basisplatine für das DIPmodul: (9 polige Buchse)

TxD Pin 2 Sendedatenleitung

RxD Pin 3 Empfangsdatenleitung

GND Pin 5 Ground (GND)

Damit ergibt sich für das Verbindungskabel zwischen PC und Basisplatine folgende Belegung:

<b>D-SUB-9 Buchse</b>		<b>D-SUB-9 Stecker</b>	
<b>(PC-Seite)</b>		<b>(MC-Seite)</b>	
Bezeichnung	Pin-Nr.	Bezeichnung	Pin-Nr.
RxD	2	TxD	2
TxD	3	RxD	3
GND	5	GND	5

*Tabelle 2: RS 232 Verbindungskabel*

## 2.5 Verdrahtung des CAN-Netzwerkes

Der D-SUB-9 Stecker "P2" an der Basisplatine mit dem DIPmodul F40 und der D-SUB-9 Stecker "CAN" an der CANopen IO-Box stellen die Anbindung zum CAN-Bus dar.

Es gilt folgende Belegung des Steckers.

<b>Pin</b>	<b>Bezeichnung</b>
2	CAN_L (dominant low)
3	CAN_GND
6	CAN_GND (optional)
7	CAN_H (dominant high)
9	CAN_VCC (+7..+13VDC)

*Tabelle 3: Verdrahtung des CAN-Busses*

Für die Nutzung des CANopen Einstiegskits sind die Signale CAN\_L und CAN\_H relevant. Im *Bild 4* ist die prinzipielle Busanschaltung dargestellt.

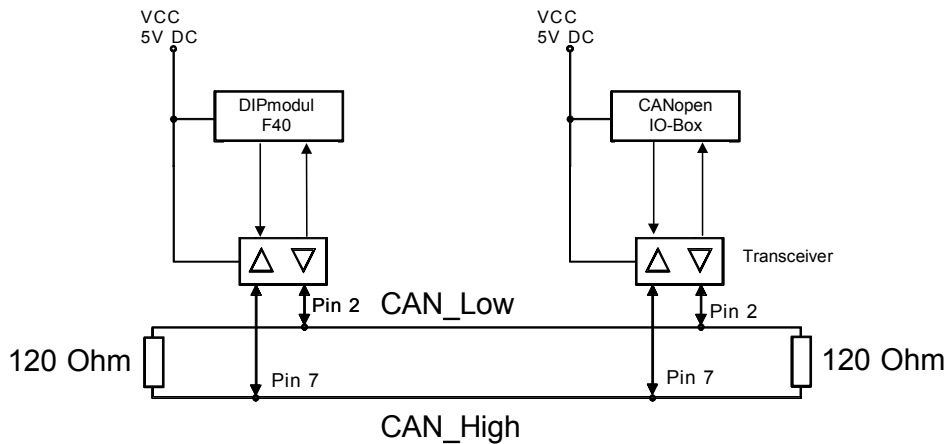


Bild 4: prinzipielle CAN-Bus-Anschaltung (ohne galvanische Entkopplung)

Als CAN-Bus-Kabel kann eine verdrehte Zweidrahtleitung eingesetzt werden, die an beiden Enden mit  $120\Omega$  abzuschließen ist. Der Wellenwiderstand des Kabels soll  $120\Omega$  betragen. Der Querschnitt richtet sich dabei nach der Leitungslänge (siehe Tabelle 4).

max. Leitungslänge [m]	max. Bitrate [kBit/s]	spezifischer Widerstand [ $m\Omega/m$ ]	Leitungsquerschnitt [ $mm^2$ ]
30	1000	70	0,25..0,34
100	500	<60	0,34..0,60
500	100	<40	0,50..0,60
1000	20	<26	0,75..0,80

Tabelle 4: empfohlene Leitungsparameter für das CAN-Kabel

### 2.5.1 Anschluss des CAN-Kabelsatzes

Dem CANopen Einstiegskit liegt ein Kabelsatz WK054 bei. Dieser besteht aus einem 3m langem 9-poligem Flachbandkabel mit 7 D-SUB-Buchsen und einem D-SUB-Stecker (zum Verlängern des Kabels), sowie 2 Steckern im Gehäuse mit den  $120\Omega$  Abschlusswiderständen. Das Kabel übernimmt ebenfalls die Spannungsversorgung der CANopen IO-Box.

Der Kabelsatz ist für den Laborbetrieb, den schnellen Einstieg und Testzwecke geeignet. Für den Dauereinsatz ist ein oben beschriebenes Kabel einzusetzen.

Die Stecker mit den Abschlusswiderständen sind an die Enden des CAN-Busses zu stecken. Die Basisplatine mit dem DIPmodul F40, die CANopen IO-Box und das USB-CANmodul können anschließend an den freien Buchsen des CAN-Busses angeschlossen werden. Verbinden Sie jedoch das USB-CANmodul noch nicht mit Ihrem PC.

An einem der beiden Stecker mit den Abschlusswiderständen enden zwei Kabel. Diese dienen normalerweise der Einspeisung von CAN\_VCC (rot, +7 bis +13 VDC) und CAN\_GND (schwarz) bei einer galvanischen Entkopplung. Eine galvanische Entkopplung ist im CANopen Einstiegskit nicht vorhanden. Die zwei Kabel müssen hier für die Einspeisung der Versorgungsspannung der CANopen IO-Box verwendet werden (*siehe dazu Kapitel 2.3*).

Sind alle Baugruppen ordnungsgemäß eingestellt und der CAN-Bus angeschlossen, kann die Spannungsversorgung zugeschaltet werden.

**Achtung:**

Alle Änderungen an den Einstellungen dürfen nur im spannungslosen Zustand erfolgen!



## **3 Installation der Tool Software**

### **3.1 Installation der Entwicklungsumgebung Softune**

Mit der Auslieferung erhalten Sie die FUJITSU Microcontrollers DVD Version 5.2 mit der Software Workbench für die F2MC-16LX Familie. Legen Sie diese DVD in Ihr DVD Laufwerk ein, starten Sie die Installation und folgen Sie allen Anweisungen. Bevor Sie diese Software verwenden können, müssen Sie diese Software Entwicklungsumgebung bei FUJITSU registrieren. Europäische Kunden können diese Registrierung direkt über folgendes Online-Formular durchführen:

[https://mcu.emea.fujitsu.com/cusreg/htm/cusreg\\_form.htm](https://mcu.emea.fujitsu.com/cusreg/htm/cusreg_form.htm)

Wenn sich Ihr Standort außerhalb Europas befindet, dann kontaktieren Sie bitte Ihren zuständigen Distributor.

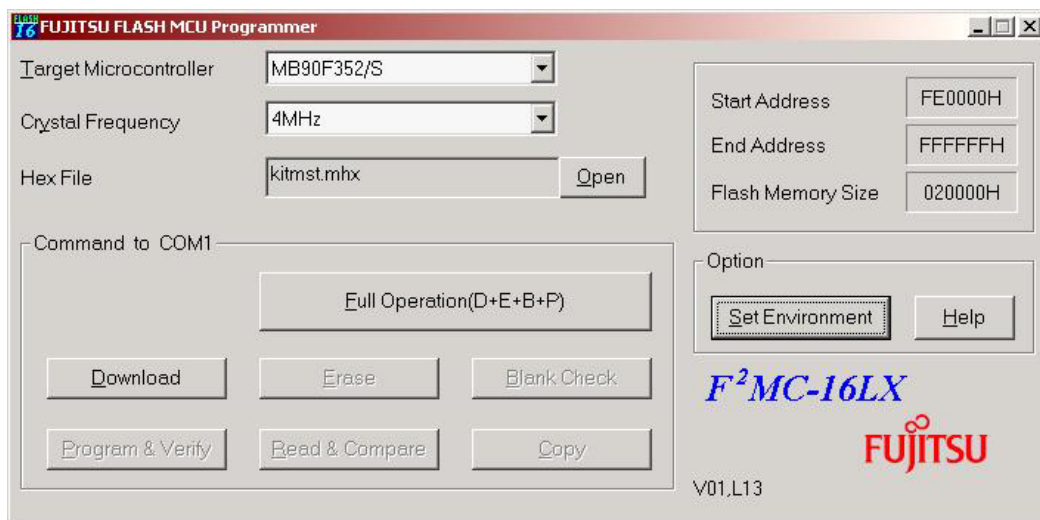
### 3.1.1 Installation des Fujitsu Flash MCU Programmers

Sie benötigen den Fujitsu Flash MCU Programmer, um ein Kontrollprogramm in den internen Flash des Microcontrollers zu programmieren, bevor Sie den Accemic Debugger (*siehe Kapitel 3.2*) in das DIPmodul F40 laden können.

Der Flash Programmer ist Freeware und liegt auf der SYS TEC Produkt-CD bei. Wechseln Sie dort in das Verzeichnis "*CDROM:\Tools\FUJITSU\Flasher\_for\_F2MC-16LX<Version>*" und führen Sie das Setup *Pcw16setup.exe* aus.

Folgen Sie allen Anweisungen bei der Installation.

Starten Sie nach der Installation den Flash Programmer aus der Programmgruppe "*FUJITSU FLASH MCU Programmer*". Dann stellen Sie den *Target Microcontroller* MB90F352/S ein. Setzen Sie unter "*Set Environment*" die verwendete COM Schnittstelle an Ihrem Rechner.



Verbinden Sie nun die Basisplatine mit dem DIPmodul F40 mit einem RS 232 Kabel mit Ihrem Rechner und schalten Sie die Betriebsspannung zur Basisplatine zu. Dann betätigen Sie bitte in der Dialogbox den Button "*Download*". Darauf folgt ein Dialog, der Sie auffordert, einen Reset am Microcontroller auszuführen:





Dazu drücken Sie bitte auf der Basisplatine mit dem DIPmodul F40 die beiden Tasten BOOT und RESET mit folgender zeitlichen Abfolge:

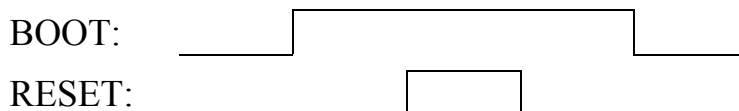


Bild 5: Einleitung des Burn-In-Modus beim DIPmodul F40

Bestätigen Sie die Dialog-Box mit OK. Nun wird das Kontrollprogramm in den Flash des Microcontrollers gebrannt. Drücken Sie nach diesem Vorgang den Reset-Taster auf der Basisplatine des DIPmodul F40 und beenden Sie das Programm. Fahren Sie mit der Installation des Accemic MDE Debuggers im folgenden Kapitel fort.

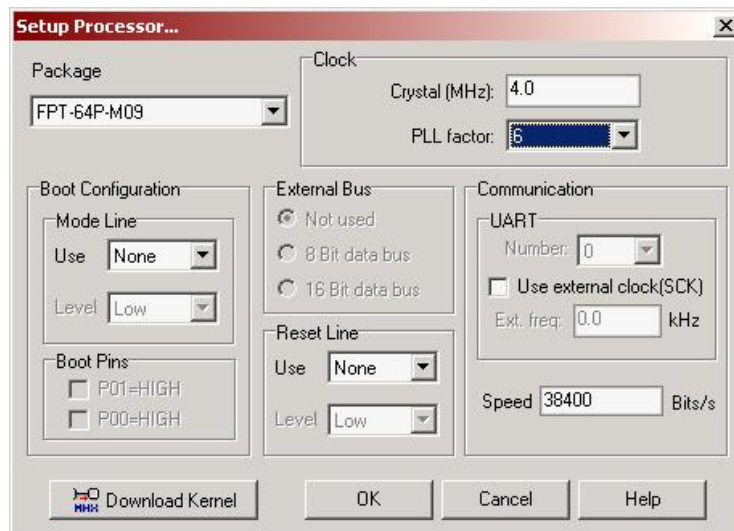
### 3.2 Installation des Accemic MDE Debugger

Der Accemic MDE Debugger kann von der Internetseite Firma Accemic geladen werden, siehe <http://www.accemic.com>. Dazu ist eine Registrierung erforderlich. Führen Sie das Setup für den Debugger aus und folgen Sie allen Anweisungen. Für die unbegrenzte Verwendung des Accemic Debuggers benötigen Sie eine Lizenz, die Sie zusätzlich käuflich erwerben müssen.

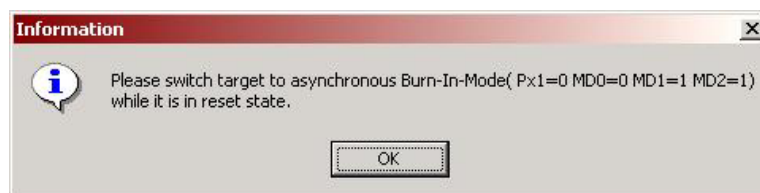
Rufen Sie nach der Installation den *Accemic MDE* aus der gleichnamigen Programmgruppe auf. Beim ersten Start werden Sie gebeten einige Einstellungen vorzunehmen. Im ersten Dialog stellen Sie den Microcontroller-Typ ein:



Bestätigen Sie diese Einstellung mit "Next >". Im nächsten Dialog stellen Sie die Eigenschaften des Microcontrollers ein. Dabei setzen Sie bitte den PLL Faktor auf 6 und tragen die gewünschte Übertragungsgeschwindigkeit unter "Speed" ein:



Verbinden Sie nun die Basisplatine mit dem DIPmodul F40 mit einem RS 232 Kabel mit Ihrem Rechner und schalten Sie die Betriebsspannung zur Basisplatine zu. Dann betätigen Sie bitte in der Dialogbox den Button "Download Kernel". Darauf folgt ein Dialog, der Sie auffordert, in den Burn-In-Modus zu wechseln:



Dazu gehen Sie bitte wie in *Bild 5* in *Kapitel 3.1.1* vor.

Dann bestätigen Sie bitte die Dialog-Box mit OK. Es wird das Kommunikationsmodul für den Accemic Debugger in den Flash kopiert. Dieses Kommunikationsmodul wird erst gestartet, wenn Sie

nach Beendigung des Downloads den RESET Taster an der Basisplatine des DIPmodul F40 betätigen (ohne BOOT).

Nun können Sie diesen Debugger für Ihre Projekte mit dem DIPmodul F40 verwenden. Beachten Sie bitte, dass für diese Projekte folgende Einstellungen in der Software Workbench vorgenommen werden müssen:

- Die Datei *monitor16LX.asm* muss im Projekt enthalten sein.
- Das Define ACC\_MON muss in den Einstellungen für den Assembler gesetzt sein.
- Das Define INIT\_SERIAL muss in den Einstellungen für den C Compiler ausgeschaltet sein.

Nähere Informationen dazu finden Sie im Kapitel 4 in diesem Manual.

**Hinweis:**

Bei Verwendung des Accemic MDE Debuggers können Sie keine printf()-Ausgaben über die serielle Schnittstelle (z.B. über den HyperTerminal) betrachten, da der Accemic MDE Debugger diese Schnittstelle für die Kommunikation mit dem Kommunikationsmodul im Flash des DIPmodul F40 benötigt.

### 3.3 Installation der CANopen Einstiegskit Software

Legen Sie die SYS TEC Produkt-CD in das CD Laufwerk ein und wechseln Sie in das Verzeichnis "*CDROM:\Products\CANopen Starterkit-152\Software\*" und führen Sie die Datei *Kit-152.exe* aus. Dieses Setup installiert automatisch die CANopen Kit Library für das DIPmodul F40 für die Entwicklungsumgebung Softune (SO-1018) und die CANopen Demo für Windows (SO-1008).

Folgen Sie bitte den Anweisungen bei der Installation.

Am Ende der Installation werden Ihnen die automatische Installation der CANopen Configuration Suite Demo SO-1047, der ODBuilder Demo SO-1007 und der USB-CANmodul Utility Disk SO-387

angeboten. Lassen Sie alle diese Zusatzinstallationen zu und drücken Sie auf WEITER.

### **3.3.1 Installation der CANopen Configuration Suite Demo**

Die CANopen Configuration Suite Demo wird automatisch mit der CANopen Kit Software installiert, wenn diese bei der Ausführung der Kit-152.exe auf der SYS TEC Produkt-CD ausgewählt wurde. Sollte diese Installation nicht automatisch durchgeführt worden sein, dann kann sie auch nachträglich installiert werden. Dazu wechseln Sie auf der SYS TEC Product-CD in das Verzeichnis "*CDROM:\Products\CCS\_SO-1047\Software\*" und führen die Datei *CCS\_Setup.exe* aus.

Folgen Sie bitte den Anweisungen bei der Installation.

Die Demo ist auf die Bitrate 125kBit/s und auf die CANopen Knotenadressen 0x20 und 0x40 begrenzt. Durch Installation eines Lizenzschlüssels, wird diese Demo als Vollversion freigeschalten. Diesen Lizenzschlüssel können Sie bei SYS TEC electronic GmbH käuflich erwerben.

### **3.3.2 Installation der ODBuilder Demo**

Auch die Demoversion des ODBuilders wird bei der Ausführung der Kit-152.exe auf der SYS TEC Produkt-CD automatisch installiert. Sie kann jedoch auch nachträglich installiert werden, indem Sie die Datei *SetupDemo.exe* im Verzeichnis "*CDROM:\Products\OD\_Builder\_SO-1007\Software\*" auf der Produkt-CD ausführen.

Folgen Sie bitte den Anweisungen bei der Installation.

Die Demo ist eine voll lauffähige Version. Jedoch können die entstehenden C-Files für das CANopen Objektverzeichnis nicht mit der CANopen Kit Library für das DIPmodul F40 verwendet werden.

### **3.3.3 Installation der USB-CANmodul Utility Disk**

Die USB-CANmodul Utility Disk enthält für das Windows Betriebssystem die Hardware-Treiber, die Dokumentation, ein CAN View Programm und ein Demoprojekt für Microsoft Visual Studio C/C++ ab Version 5.0.

Bitte stecken Sie das USB-CANmodul nicht an den USB Port ihres Rechners, bevor Sie die Treiber nicht installiert haben. Die Installation wird ebenfalls von Kit-152.exe mit angeboten. Sollten Sie jedoch diese Installation dort ausgeschaltet haben, dann wechseln Sie auf der SYS TEC Produkt-CD in das Verzeichnis "*CDROM:\Products\USB-CANmodul\_Series\Software\SO-387\*" und führen Sie die Datei *Setup.exe* aus.

Folgen Sie bitte den Anweisungen bei der Installation. Weitere Hinweise zur Installation finden Sie auch im Handbuch *L-487*.

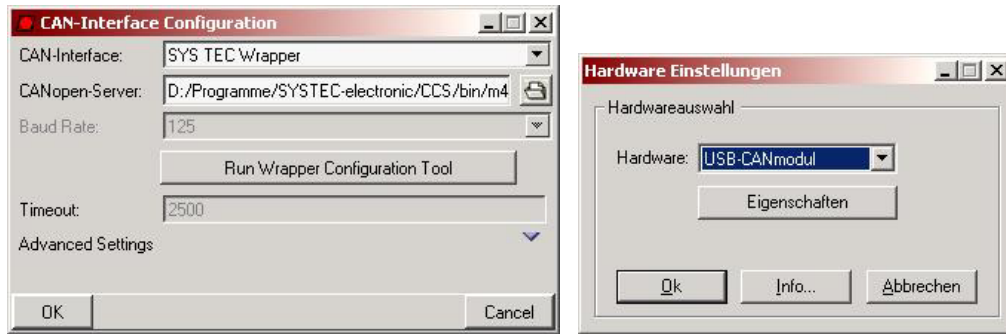


## **4 Inbetriebnahme der CANopen Software**

Nachdem die Hardware wie im *Kapitel 2* beschrieben konfiguriert und die Baugruppen über das CAN-Bus-Kabel verbunden sind sowie die notwendige Software (siehe *Kapitel 3*) installiert wurde, wird die CANopen Einstiegskit Software in das DIPmodul F40 geladen und gestartet.

Bei der Erstinbetriebnahme gehen Sie bitte wie folgt vor:

1. Verbinden Sie die Basisplatine des DIPmodul F40 mit dem mitgeliefertem seriellen Verbindungskabel an die COM Schnittstelle an Ihrem PC.
2. Verbinden Sie alle Komponenten mit dem CAN-Kabel, wie es im *Kapitel 2.5.1* beschrieben ist.
3. Schalten Sie die Spannungsversorgung zu den Komponenten hinzu und verbinden Sie das USB-CANmodul mit Ihrem Rechner.
4. Rufen Sie den CDM ("*CANopen Device Monitor*") aus der CANopen Configuration Suite auf. Beim ersten Start werden Sie gebeten, das CAN-Interface zu konfigurieren. Stellen Sie dabei das CAN-Interface auf "*SYS TEC Wrapper*". Klicken Sie auf den Button "*Run Wrapper Configuration Tool*" und wählen Sie dort die Hardware "*USB-CANmodul*" aus. Unter "*Eigenschaften*" können Sie dort noch die Gerätenummer des USB-CANmoduls einstellen. Sollten Sie nur ein USB-CANmodul an Ihrem Rechner betreiben, dann können Sie die Gerätenummer auf den Wert 255 belassen, wobei der Hardware-Treiber das Modul am Rechner verwendet, welches er als erstes findet. Bestätigen Sie alle Einstellungen mit OK.



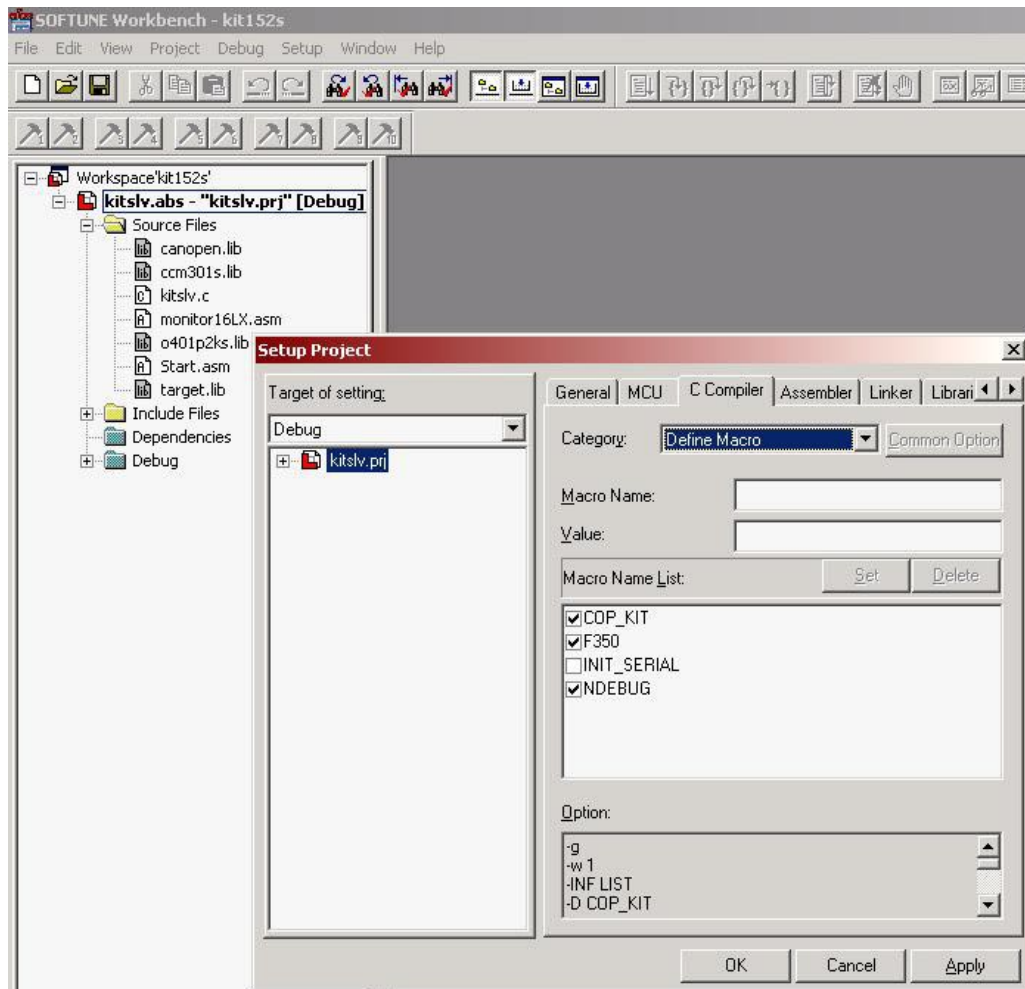
5. Wählen Sie im CDM im Menü *Connection* → *Connect*. Damit sollte die rote LED am USB-CANmodul erlöschen.
6. Rufen Sie die Software Entwicklungsumgebung auf und öffnen Sie den Arbeitsbereich *kit152s.wsp* aus dem Verzeichnis "*C:\systec\cop\target\dipmodul-f40\no\_os\softune\kitslv\*".

Die im Kit enthaltenen Projekte *kit152s.wsp* (CANopen Slave) sowie *kit152m.wsp* (CANopen Master) enthalten jeweils eine Target-Umgebung namens *Debug* und eine namens *Release*. *Debug* ist für die Arbeit mit dem Accemic-Debugger vorbereitet, während *Release* Code erzeugt, der direkt im Flash der CPU ausgeführt wird (ohne Debugger). Die Projektunterschiede sind im folgenden Punkt erläutert.

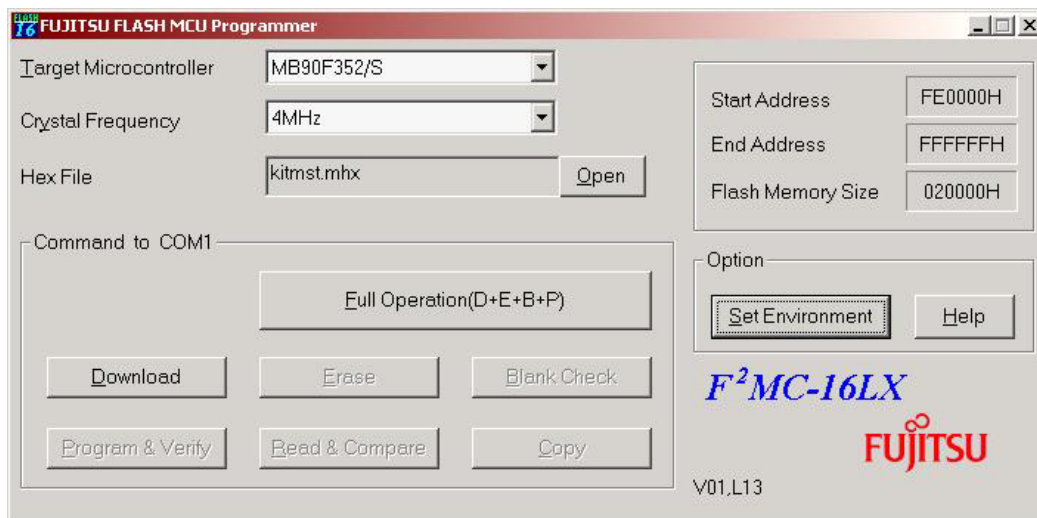
7. Verwenden Sie den Accemic MDE Debugger (d.h. ohne Ansicht der `printf()`-Anweisungen), dann ist es nötig, dass die Datei *monitor16LX.asm* im Projekt enthalten ist. Diese ist Bestandteil der Accemic-Installation. Im Projekt ist die Datei bereits eingetragen, allerdings müssen Sie diese noch aus dem Accemic-Installationsverzeichnis in den Projektpfad kopieren. Kopieren die dazu alle Dateien aus dem Verzeichnis *include* der Accemic-Installation in das Verzeichnis "*systec\cop\target\dipmodul-f40\no\_os\softune\source*". Im Gegensatz zur Target-Umgebung *Release* ist in den Projekteinstellungen (Menü *Project* → *Setup Project...* → *C Compiler* → *Category: Define Macro*) das Define `INIT_SERIAL` ausgeschaltet und das Define `ACC_MON` eingeschaltet. Sollte dies nicht der Fall sein, dann schalten Sie es bitte entsprechend. Ebenfalls muss das Define `ACC_MON` in den



Projekteinstellungen für den Assembler (Menü *Project* → *Setup Project...* → *Assembler* → *Category: Define Macro*) eingeschaltet sein. Übersetzen Sie das Projekt neu und fahren Sie mit *Punkt 17* fort.



8. Verwenden Sie keinen Accemic MDE Debugger, dann entfernen Sie gegebenenfalls die Datei *monitor16LX.asm* aus dem Projekt und setzen Sie das Define `INIT_SERIAL` (C Compiler) und deaktivieren Sie das Define `ACC_MON` (C Compiler und Assembler) in den Projekteinstellungen. Übersetzen Sie das Projekt neu.
9. Rufen Sie den FUJITSU FLASH MCU Programmer FMC16LX auf und stellen Sie unter "*Set Environment*" Ihre verwendete COM Schnittstelle ein.



10. Drücken Sie auf "*Download*". Dabei werden Sie aufgefordert, den Burn-In-Modus zusetzen. Verfahren Sie dabei wie in *Kapitel 3.1.1, Bild 5* beschrieben wurde. Wenn Sie die Aufforderung mit OK betätigen, dann wird das Flash Controller Programm in die Hardware übertragen. Bestätigen Sie anschließend die Meldung mit dem Abschluss der Übertragung mit OK.
11. Drücken Sie auf "*Open*" und stellen Sie die Datei *kitslv.mhx* aus dem Verzeichnis "*C:\systec\cop\target\dipmodul-f40\no\_os\softune\kitslv\Debug\ABS*" oder "*C:\systec\cop\target\dipmodul-f40\no\_os\softune\kitslv\Release\ABS*" ein.
12. Drücken Sie anschließend auf den Button "*Erase*", um den Flash im DIPmodul F40 zu löschen.
13. Drücken Sie auf "*Program & Verify*" um das compilierte Projekt in den Flash des DIPmodul F40 zu programmieren. Beenden Sie danach das Tool FMC16LX.
14. Rufen Sie HyperTerminal auf und konfigurieren Sie ihn für die verwendete COM-Schnittstelle 9600 Bit/s, 8 Datenbits, keine Parität, 1 Stopbit und keine Flusssteuerung. Sollte die Meldung „COM... konnte nicht geöffnet werden,...“ erscheinen, prüfen Sie, ob das Tool FMC16LX beendet wurde.

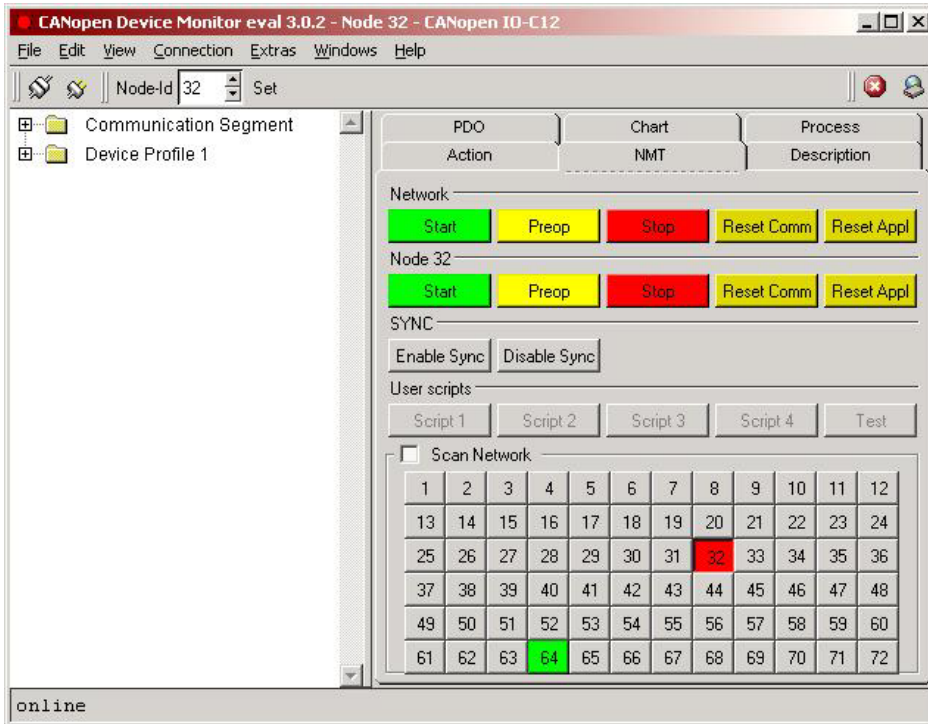
15. Setzen Sie die DIPmodul F40 durch Drücken der Taste RESET auf der Basisplatine zurück. Damit wird das Projekt auf der Hardware gestartet.

Sie sehen im HyperTerminal folgende Ausgabe:

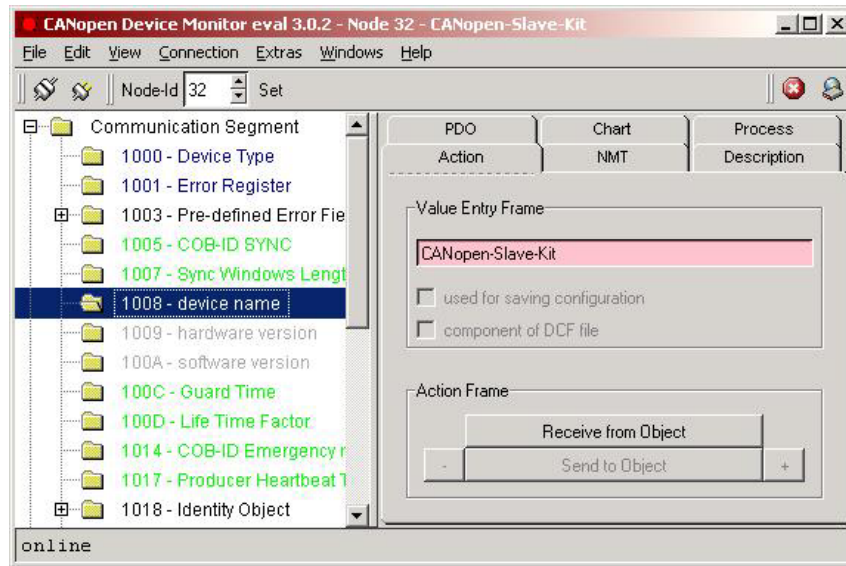
```
*****  
CANopen Slave Kit demo  
(c) SYS TEC electronic GmbH  
CANopen V5.29.0  
*****  
  
init CANopen! Node ID = 0x20  
connect this node to network...  
...
```

Weiterhin sehen Sie die ständig wiederkehrende Ausgabe: „TX-PDO 0 event“. Dieses PDO wird im 500ms Takt über die CAN-Nachricht mit der CAN-ID 0x2C0 gesendet. Das führt dazu, dass die LEDs der CANopen IO-Box blinken.

16. Drücken Sie eine der 8 Taster auf der CANopen IO-Box. Das führt dazu, dass ein PDO mit der CAN-ID 0x1C0 gesendet wird und im HyperTerminal die Ausgabe „RX-PDO 0 event“ erscheint.
17. Wechseln Sie nun zum CDM, klicken Sie im rechten Teilfenster auf das Sheet "NMT" und wählen Sie die Knotennummer 32 aus. Die Frage „Do you want to load ...“ mit Nein beantworten.



18. Klicken Sie im Menü des CDM auf *File* → *Load EDS* → *Select another file* und wählen Sie im Verzeichnis "*C:\systec\cop\objdicts\o401p2ks\*" die Datei *o401p2ks.eds* aus.
19. Wechseln sie im rechten Teilfenster das Sheet "*Action*" aus und wählen Sie im linken Fenster durch Aufklappen der Baumstruktur "*Communication Segment*" das Objekt 0x1008 (device name) und drücken Sie auf den Button "*Receive from Object*". Nun wird der Gerätenamen des CANopen Einstiegskit gelesen und auf der rechten Seite des Fensters dargestellt: „*CANopen-Slave-Kit*“.



Sie können die CANopen Demo Applikation *kitslv.c* auf Ihre Bedürfnisse anpassen. Beachten Sie jedoch, dass die CANopen Einstiegskit Software die Einschränkungen laut *Kapitel 4.2* hat.

### 4.1 Weitere Hinweise

Neben der CANopen Slave Kit Library ist im Verzeichnis "*C:\systec\cop\target\dipmodul-f40\no\_os\softune\kitmst\*" auch ein Projekt für eine CANopen Master Demo enthalten. Dieses hat die gleichen Einschränkungen wie der Slave, unterstützt jedoch die NMT Master Dienste nach DS-301. In der Demo-Applikation wird dort auch der SDO Client und der Emergency Consumer verwendet.

Für das Windows Betriebssystem sind im Verzeichnis "*C:\systec\cop\target\x86\windows\vc7\*" zwei Demo Projekte (Slave und Master) mit den gleichen Einschränkungen enthalten. Diese können mit dem mitgelieferten USB-CANmodul verwendet werden. Beachten Sie jedoch, dass die Tools der CANopen Configuration Suite nicht gleichzeitig mit diesen Windows Demos auf ein USB-CANmodul zugreifen können.

Die Beschreibung der CANopen Funktionen entnehmen Sie dem CANopen Softwaremanual (*L-1020*). Gleiches gilt für die Bedienung des CANopen Device Monitor CDM (*L-1055*).

## 4.2 Einschränkungen der CANopen Kit Software

Die CANopen Slave Kit Library hat folgende Einschränkungen:

- Verwendbare CAN Bitraten: nur 125kBit/s.
- Einstellbare Knotennummer: nur 0x20 und 0x40.
- Eingeschränktes Objektverzeichnis mit 8 x digital Input 8 Bit, 8 x digital Output 8 Bit, 4 x analog Input 16 Bit und 4 x analog Output 16 Bit.
- Heartbeat Producer aber kein Heartbeat Consumer und kein Emergency Consumer.
- 2 TPDOs und 2 RPDOs, statisches PDO Mapping (definiert nach *DS-401 V2.1 ab Kapitel 6.2.4*), feste Inhibit Time von 500ms, keinen Event Timer (Subindex 5 der Kommunikationsparameter der TPDOs).

Die CANopen Master Kit Library bietet die Möglichkeit, 2 Slave Knoten per Life Guarding zu überwachen. Weiterhin kann der Emergency Consumer für 2 Knoten konfiguriert und ein SDO Client (Objekt 0x1280) verwendet werden. All diese zusätzlichen Dienste beim Master sind auf die Knotennummern 0x20 und 0x40 beschränkt.

## 4.3 Hinweise zu den Demo Applikationen

Die Demo Applikationen *kitslv.c* für den Slave und *kitmst.c* für den Master sind auf die entsprechenden Objektverzeichnisse und den beigefügten Gegenknoten (CANopen IO-Box) zugeschnitten.

### 4.3.1 Die Initialisierungsstruktur

Auch wenn einige Initialisierungsparameter für das CANopen Kit unveränderbar sind, muss eine Initialisierungsstruktur angegeben werden. Werden ungültige Werte an diese Struktur übergeben, dann gibt die Funktion CcmInitCANopen() einen entsprechenden Fehlercode zurück. Eine Liste der möglichen Fehlercodes finden Sie im *Handbuch des CANopen Stacks L-1020*.

```

CONST tCcmInitParam ROM CcmInitParamInst1_g =
{
    NODE_ID,                // NodeId
    BAUDRATE,               // baudrate
    NULL,                   // address of BDI table
    0,                       // size of BDI table
    0xFFFFFFFFL,
    0x00000000L,
    {{0}},                  // hardware parameter

    #if (DEV_SYSTEM != _DEV_WIN32_)
    TgtEnableCanInterrupt1, // for 1st CAN-Controller
    #else
    NULL,
    #endif

    AppCbNmtEvent,         // NMT callback function
    ObdInitRam             // OD init function
};

```

### 4.3.2 Die Variablen-tabelle

Die Variablen-tabelle verknüpft die C-Variablen in der Applikation mit den jeweiligen Objekten im Objektverzeichnis. So ist sichergestellt, dass die Applikation schnell auf die aktuellen Daten im Objektverzeichnis zugreifen kann.

```

CONST tVarParam ROM aVarTab_g[] =
{
    // digital input 8 bit
    {kVarValidAll, 0x6000, 0x01, sizeof (BYTE),
     &abDigitalIn8Bit_g[0], NULL, NULL},
    ...
};

```

Der Inhalt dieser Tabelle darf nicht verändert werden, denn er ist abhängig vom verwendeten Objektverzeichnis, welches beim CANopen Kit unveränderbar ist.

### 4.3.3 Die PDO Tabelle

Die PDO Tabelle beinhaltet die Kommunikationsparameter für das PDO Linking, die Pointer auf die zu verwendenden Daten und die Pointer auf die Callback-Funktionen der PDOs. Diese Parameter können beim CANopen Kit verändert werden. Beachten Sie jedoch, dass es Einschränkungen beim Transmisionstyp, Inhibit Time und beim Event Timer gibt. Ein Eintrag dieser PDO Tabelle hat folgenden Aufbau:

```
typedef struct
{
    WORD           m_wPdoCommuIndex;
    DWORD          m_dwCanId;
    BYTE           m_bTxTyp;
    WORD           m_wInhibitTime;
    WORD           m_wEventTime;
    BYTE           m_bSize;
    void MEM*      m_pVarData;
    tPdoCallback   m_fpCallback;
} tPdoStaticParam;

CONST tPdoStaticParam ROM aPdoTab_g[] =
{
    // 1st RPDO
    {0x1400,
      (0x000001C0L), 255, 0, 0,
      1, &abDigitalOut8Bit_g[0],
      AppCbRxPdo},
    ...
};
```

Parameter	Bedeutung
m_wPdoCommuIndex	Kommunikationsindex des PDO im OD (0x1400-0x15FF für RPDOs, 0x1800-0x19FF für TPDOs)
m_dwCanId	COB-ID des PDO nach Subindex 1 der Kommunikationsparameter
m_bTxTyp	Transmisionstyp des PDO nach Subindex 2 der Kommunikationsparameter (Parameter wird ignoriert)
m_wInhibitTime	Inhibit Time des PDO nach Subindex 3 der Kommunikationsparameter (Parameter wird ignoriert)



Parameter	Bedeutung
m_wEventTime	Event Time des PDO nach Subindex 5 der Kommunikationsparameter (Parameter wird ignoriert)
m_bSize	Anzahl der zu übertragenden Bytes in der CAN-Nachricht des PDO (zwischen 0 und 8)
m_pVarData	Pointer auf bis zu 8 zusammenhängende Datenbytes, die in der CAN-Nachricht des PDOs übertragen werden.
m_fpCallback	Pointer auf die Callback-Funktion, die nach dem Senden oder Empfang des PDO gerufen wird. Dieser Pointer kann NULL sein.

Beachten Sie bitte, dass die Zuordnung der PDO Daten im Parameter m\_pVarData mit dem festen Mapping (*siehe Tabelle 5*) und der Variablentabelle (*siehe Kapitel 4.3.2*) übereinstimmen. Sonst kann es vorkommen, dass die Daten, die evtl. per SDO für das entsprechende Objekt übertragen werden, nicht mit den Daten im PDO übereinstimmen.

Mapping Objekte	Mapping Daten
<b>RPDO1:</b>	
0x1600/0	8 gemappte Objekte
0x1600/1	Objekt 0x6200/1
0x1600/2	Objekt 0x6200/2
0x1600/3	Objekt 0x6200/3
0x1600/4	Objekt 0x6200/4
0x1600/5	Objekt 0x6200/5
0x1600/6	Objekt 0x6200/6
0x1600/7	Objekt 0x6200/7
0x1600/8	Objekt 0x6200/8
<b>RPDO2:</b>	
0x1601/0	4 gemappte Objekte
0x1601/1	Objekt 0x6411/1
0x1601/2	Objekt 0x6411/2
0x1601/3	Objekt 0x6411/3
0x1601/4	Objekt 0x6411/4

Mapping Objekte	Mapping Daten
<b>TPDO1:</b>	
0x1800/0	8 gemappte Objekte
0x1800/1	Objekt 0x6000/1
0x1800/2	Objekt 0x6000/2
0x1800/3	Objekt 0x6000/3
0x1800/4	Objekt 0x6000/4
0x1800/5	Objekt 0x6000/5
0x1800/6	Objekt 0x6000/6
0x1800/7	Objekt 0x6000/7
0x1800/8	Objekt 0x6000/8
<b>TPDO2:</b>	
0x1801/0	4 gemappte Objekte
0x1801/1	Objekt 0x6401/1
0x1801/2	Objekt 0x6401/2
0x1801/3	Objekt 0x6401/3
0x1801/4	Objekt 0x6401/4

Tabelle 5: festes PDO Mapping nach DS-401 V2.1

#### 4.3.4 Verwendung der PDO-Variablen

Die C-Variablen eines PDOs werden mit der PDO Tabelle (*siehe Kapitel 4.3.3*) an das CANopen übergeben. Die Sendevariablen muss die Applikation ändern (z.B. auf Grund eines digitalen Einganges), um diese Variable auf den CAN-Bus zu senden. Wird ein PDO vom CANopen Stack empfangen, dann werden die enthaltenen Werte in die Empfangsvariablen in der Applikation kopiert.

Der CANopen Stack ruft nach dem Senden und nach dem Empfang eines PDOs eine PDO Callback-Funktion, die mit der PDO Tabelle definiert wird. Über diese Funktion können Sie eine spezielle Reaktion auf dieses Ereignis implementieren. Da beim CANopen Kit nur statisches PDO Mapping unterstützt wird, finden Sie zu der aufgerufenen PDO Callback-Funktion leicht die dazugehörigen PDO-Variablen, die Sie bereits mit der PDO Tabelle definiert haben.

Es gibt zwei unterschiedliche Verfahren, die PDO-Variablen zu ändern bzw. zu lesen.

1. Im Hauptprogramm befindet sich eine Prozessschleife, in der Sie die Variablen ständig lesen und schreiben können. Hat sich z.B. der Wert eines digitalen Eingang geändert, dann schreiben sie den neuen Wert in die PDO-Variable und rufen dann die Funktion CcmSignalStaticPdo() mit dem entsprechenden PDO Kommunikationsparameter. Im nächsten Prozessdurchlauf sendet der CANopen Stack dieses PDO (vorausgesetzt die Inhibit Time ist abgelaufen). Die Empfangsvariablen können sofort dazu verwendet werden, einen digitalen Ausgang zu schalten.

Beispiel:

```
// Prozessschleife
while (1)
{
    // hat sich der digitale Eingang geändert, dann senden
    if (abDigitalIn8Bit_g[0] != READ_IO ())
    {
        abDigitalIn8Bit_g[0] = READ_IO ();
        CcmSignalStaticPdo (0x1800);
    }

    // Ausgänge direkt schreiben
    WRITE_IO (abDigitalOut8Bit_g[0]);

    // CANopen Prozess ausführen
    CcmProcess ();
}
```

2. Sie können die PDO-Variablen als Reaktion innerhalb der PDO Callback-Funktion lesen oder schreiben. Das hat z.B. den Vorteil, dass Sie einen Ausgang nur dann setzen, wenn das PDO wirklich empfangen wurde. Oder Sie senden die Werte nach einem vorgegebenen Berechnungsalgorithmus.

Beispiel:

```
// Callback-Funktion für Sende-PDO
tCopKernel PUBLIC AppCbTxPdo (WORD wPdoCommuIndex_p)
{
    // als nächstes den nachfolgenden Wert senden
    abDigitalIn8Bit_g[0] += 1;
    CcmSignalStaticPdo (0x1800);
}

// Callback-Funktion für Empfangs-PDO
tCopKernel PUBLIC AppCbRxPdo (WORD wPdoCommuIndex_p)
```



```
switch (NmtEvent_p)
{
    case kNmtEvEnterInitialising:
        SWITCH_STATUS_LED (OFF);
        ...
        break;

    case kNmtEvResetNode:
        SWITCH_STATUS_LED (OFF);
        break;

    case kNmtEvResetCommunication:
        SWITCH_STATUS_LED (OFF);
        break;

    case kNmtEvEnterPreOperational:
        SWITCH_STATUS_LED (OFF);
        break;

    case kNmtEvEnterOperational:
        SWITCH_STATUS_LED (ON);
        break;

    case kNmtEvEnterStopped:
        SWITCH_STATUS_LED (OFF);
        break;
}

return kCopSuccessful;
}
```



---

**Index**

<b>3</b>		<b>K</b>	
3204000 .....	2	Knotennummer .....	26
<b>A</b>		<b>L</b>	
Accemic MDE Debugger .....	13	L-1000 .....	2, 4
<b>B</b>		L-1020 .....	2, 25
Baudrate .....	26	L-1022 .....	2
<b>C</b>		L-1048 .....	2
CAN_GND .....	7	L-1055 .....	2
CAN_H .....	7	L-1056 .....	2
CAN_L .....	7	L-1062 .....	2, 4
CAN-Busanschaltung .....	7	L-1078 .....	2
CAN-Bus-Kabel .....	8	L-487 .....	2
CAN-Kabelsatz .....	8	Lieferumfang .....	2
CANopen Configuration Suite .	19	<b>M</b>	
CANopen Device Monitor .....	19	Master .....	25, 26
CDM .....	19, 25	MM-217 .....	2
<b>D</b>		<b>O</b>	
Demo .....	26	Objektverzeichnis .....	26
<b>E</b>		<b>P</b>	
Einschränkungen .....	25, 26	PDO .....	26
Emergency .....	26	phyPS-451-Y .....	2
<b>H</b>		<b>R</b>	
Heartbeat .....	26	RS 232 .....	6
HyperTerminal .....	6	<b>S</b>	
<b>I</b>		SDO Client .....	26
Inbetriebnahme .....	19	Slave .....	26
Inhibit Time .....	26	SO-1007 .....	2
Installation .....	11	SO-1008 .....	2
<b>J</b>		SO-1018 .....	2
Jumperbelegung .....	4	SO-1047 .....	2
		SO-387 .....	2
		Softune .....	11
		Software Download .....	6

Spannungsversorgung .....	5	<b>W</b>		
<b>V</b>			Windows .....	25
Voraussetzung .....	1		WK054.....	2, 8
			WK800.....	2



**Dokument: KIT-152**  
**Dokumentnummer: L-1078d\_5, Auflage Mai 2011**

---

**Wie würden Sie dieses Handbuch verbessern?**

---

---

---

---

**Haben Sie in diesem Handbuch Fehler entdeckt?** Seite

---

---

---

---

**Eingesandt von:**

Kundennummer: \_\_\_\_\_

Name: \_\_\_\_\_

Firma: \_\_\_\_\_

Adresse: \_\_\_\_\_

\_\_\_\_\_

**Einsenden an:**

SYS TEC electronic GmbH  
August-Bebel-Str. 29  
D-07973 Greiz, Germany  
Fax : +49 (0) 3661 62 79 99

Published by

**SYS TEC**  
ELECTRONIC

---

© SYS TEC electronic GmbH Mai 2011

Ordering No. L-1078d\_5  
Printed in Germany