

***System Manual***  
***PLCcore-iMX35 CODESYS***

**Manual für CODESYS-User**  
**Version 5**

**Ausgabe Mai 2016**

Dokument-Nr.: L1573d\_5

SYS TEC electronic GmbH Am Windrad 2 D-08468 Heinsdorfergrund  
Telefon: +49 (3765) 38600-0 Telefax: +49 (3765) 38600-4100  
Web: <http://www.systemec-electronic.com> Mail: [info@systemec-electronic.com](mailto:info@systemec-electronic.com)

## Status/Änderungen

Datum/Version	Abschnitt	Änderung	Bearbeiter
2014/09/12 Version 1	alle	Erstellung	Dr. N. Prang
2016/03/01 Version 2	alle	Überarbeitete Kopfzeile	T. Volckmann
2016/03/24 Version 3	Anhang A	Korrektur der Grafiken des Referenzdesigns	T. Volckmann
2016/05/12 Version 4	Abschnitt 3	Produktbeschreibung überarbeitet	T. Volckmann
2016/05/17 Version 5	Abschnitt 4.2	Anpassung der Spezifikation des Steckernetzteil	T. Volckmann
	Abschnitt 6.2 Abschnitt 7.3 Abschnitt 7.10 Abschnitt 7.11.1 Abschnitt 7.12.1 Abschnitt 7.12.2	Rechtschreibkorrektur	T. Volckmann

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 Handbuchs 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 2016 SYS TEC electronic GmbH, D-08468 Heinsdorfergrund.  
 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:

Kontakt	Direkt	Ihr Lokaler Distributor
Adresse:	SYS TEC electronic GmbH Am Windrad 2 D-08468 Heinsdorfergrund GERMANY	Sie finden eine Liste unserer Distributoren unter  <a href="http://www.systec-electronic.com/distributors">http://www.systec-electronic.com/distributors</a>
Angebots-Hotline:	+49 (0) 3765 / 38600-0 <a href="mailto:info@systec-electronic.com">info@systec-electronic.com</a>	
Technische Hotline:	+49 (0) 3765 / 38600-0 <a href="mailto:support@systec-electronic.com">support@systec-electronic.com</a>	
Fax:	+49 (0) 3765 / 38600-4100	
Webseite:	<a href="http://www.systec-electronic.com">http://www.systec-electronic.com</a>	

5. Auflage Mai 2016

## Inhalt

<b>1</b>	<b>Einleitung</b> .....	<b>4</b>
<b>2</b>	<b>Übersicht / Wo finde ich was?</b> .....	<b>5</b>
<b>3</b>	<b>Produktbeschreibung</b> .....	<b>6</b>
<b>4</b>	<b>Development Kit PLCcore-iMX35</b> .....	<b>8</b>
4.1	Übersicht.....	8
4.2	Elektrische Inbetriebnahme des Development Kit PLCcore-iMX35 .....	9
4.3	Bedienelemente des Development Kit PLCcore-iMX35.....	10
4.4	Optionales Zubehör .....	11
4.4.1	USB-RS232 Adapter Kabel .....	11
<b>5</b>	<b>Anschlussbelegung des PLCcore-iMX35</b> .....	<b>12</b>
<b>6</b>	<b>SPS-Funktionalität des PLCcore-iMX35</b> .....	<b>15</b>
6.1	Übersicht.....	15
6.2	Systemstart des PLCcore-iMX35 .....	15
6.3	Programmierung des PLCcore-iMX35 .....	16
6.4	Prozessabbild des PLCcore-iMX35.....	16
6.4.1	Lokale Ein- und Ausgänge.....	16
6.5	Kommunikationsschnittstellen .....	17
6.5.1	Serielle Schnittstellen .....	17
6.5.2	CAN-Schnittstellen.....	17
6.5.3	Ethernet-Schnittstellen.....	17
6.6	Nutzung der CAN-Schnittstellen mit CANopen .....	17
6.6.1	Konfiguration von CAN0 für CANopen unter CODESYS .....	19
6.6.2	Zusätzliche CAN-Schnittstellen .....	22
6.7	Target-Visualisierung.....	22
<b>7</b>	<b>Konfiguration und Administration des PLCcore-iMX35</b> .....	<b>23</b>
7.1	Systemvoraussetzungen und erforderliche Softwaretools .....	23
7.2	Linux-Autostart aktivieren bzw. deaktivieren .....	24
7.3	Ethernet-Konfiguration des PLCcore-iMX35 .....	25
7.4	Boot-Konfiguration des PLCcore-iMX35 .....	27
7.5	Vordefinierte Nutzerkonten .....	28
7.6	Anmeldung am PLCcore-iMX35.....	28
7.6.1	Anmeldung an der Kommando-Shell.....	28
7.6.2	Anmeldung am FTP-Server .....	29
7.7	Anlegen und Löschen von Nutzerkonten .....	31
7.8	Passwort eines Nutzerkontos ändern.....	32
7.9	Setzen der Systemzeit.....	33
7.10	Dateisystem des PLCcore-iMX35 .....	34
7.11	Kalibrierung des Touchscreen.....	35
7.11.1	Automatische Überprüfung der Touchscreen-Kalibrierung .....	35
7.11.2	Manuelle Kalibrierung des Touchscreen .....	35
7.12	Software-Update des PLCcore-iMX35 .....	36
7.12.1	Update der SPS-Firmware.....	36
7.12.2	Update des Linux-Images.....	38
	<b>Anhang A: Referenzdesigns zum PLCcore-iMX35</b> .....	<b>42</b>
	<b>Anhang B: GNU GENERAL PUBLIC LICENSE</b> .....	<b>49</b>
	<b>Index</b> .....	<b>54</b>

# 1 Einleitung

Vielen Dank, dass Sie sich für das SYS TEC PLCcore-iMX35 entschieden haben. Mit der Produktvariante mit integriertem CODESYS-Laufzeitsystem verfügen Sie über einen innovativen und leistungsfähigen SPS-Kern. Aufgrund seiner integrierten Target-Visualisierung, hohen Performance sowie wegen seiner umfangreichen on-board Peripherie eignet er sich besonders gut als Kommunikations- und Steuerrechner für HMI Anwendungen.

Bitte nehmen Sie sich etwas Zeit, dieses Manual aufmerksam zu lesen. Es beinhaltet wichtige Informationen zur Inbetriebnahme, Konfiguration und Programmierung des PLCcore-iMX35. Es wird Ihnen helfen, sich mit dem Funktionsumfang und der Anwendung des PLCcore-iMX35 vertraut zu machen. Dieses Dokument wird ergänzt durch weitere Manuals der SYS TEC electronic GmbH und des CODESYS-Herstellers 3S Smart Software Solutions GmbH. Eine Auflistung der relevanten Manuals zum PLCcore-iMX35 beinhaltet Tabelle 1 im Abschnitt 2. Bitte beachten Sie auch diese ergänzenden Dokumentationen.

Für weiterführende Informationen, Zusatzprodukte, Updates usw. empfehlen wir den Besuch unserer Website unter: <http://www.systec-electronic.com>. Der Inhalt dieser Webseite wird periodisch aktualisiert und stellt Ihnen stets die neuesten Software-Releases und Manual-Versionen zum Download bereit.

## Anmerkungen zum EMV-Gesetz für das PLCcore-iMX35



Das PLCcore-iMX35 ist als Zulieferteil für den Einbau in ein Gerät (Weiterverarbeitung durch Industrie) bzw. als Entwicklungsboard für den Laborbetrieb (zur Hardware- und Softwareentwicklung) bestimmt.

Nach dem Einbau in ein Gerät oder bei Änderungen/Erweiterungen an diesem Produkt muss die Konformität nach dem EMV-Gesetz neu festgestellt und bescheinigt werden. Erst danach dürfen solche Geräte in Verkehr gebracht werden.

Die CE-Konformität gilt nur für den hier beschriebenen Anwendungsbereich unter Einhaltung der im folgenden Handbuch gegebenen Hinweise zur Inbetriebnahme! Das PLCcore-iMX35 ist ESD empfindlich und darf nur an ESD geschützten Arbeitsplätzen von geschultem Fachpersonal ausgepackt und gehandhabt bzw. betrieben werden.

Das PLCcore-iMX35 ist ein Modul für den Bereich Automatisierungstechnik. Durch die Programmierbarkeit nach IEC 61131-3 und die Verwendung von CAN-Bus und Ethernet-Standard-Netzwerkschnittstelle für verschiedenste Automatisierungslösungen, sowie dem standardisierten Netzwerkprotokoll CANopen ergeben sich geringere Entwicklungszeiten bei günstigen Kosten der Hardware. Durch die on-board Realisierung der SPS-Funktionalität mit optionaler CANopen-Netzwerkschicht ist die Erstellung einer Firmware durch den Anwender überflüssig geworden.

## 2 Übersicht / Wo finde ich was?

Das PLCcore-iMX35 basiert auf der Hardware des ECUcore-iMX35 und erweitert dieses um SPS-spezifische Funktionalitäten (SPS-Firmware, Target-Visualisierung). Für die Hardwarekomponenten wie das ECUcore-iMX35 bzw. das PLCcore-iMX35 selbst (die Hardware beider Module ist identisch), die Developmentboards sowie Referenzschaltungen existieren eigene Hardware-Manuals. Softwareseitig wird das PLCcore-iMX35 mit der IEC 61131-3-konformen Programmierumgebung CODESYS programmiert. Zu CODESYS existieren ebenfalls eigene Handbücher, die den Umgang mit dem Programmierwerkzeug beschreiben. Diese sind Bestandteil des Software-Paketes "CODESYS". Tabelle 1 listet die für das PLCcore-iMX35 relevanten Manuals auf.

Tabelle 1: Übersicht relevanter Manuals zum PLCcore-iMX35

Informationen über...	In welchem Manual?
Grundlegende Informationen zum PLCcore-iMX35 (Konfiguration, Administration, Prozessabbild, Anschlussbelegung, Firmwareupdate, Referenzdesigns usw.)	In diesem Manual
Entwicklung anwenderspezifischer C/C++ Applikationen für das ECUcore-iMX35 / PLCcore-iMX35, VMware-Image des Linux-Entwicklungssystems	System Manual ECUcore-iMX35 (Manual-Nr.: L-1569)
Hardware-Beschreibung zum ECUcore-iMX35 / PLCcore-iMX35, Referenzdesigns usw.	Hardware Manual ECUcore-iMX35 (Manual-Nr.: L-1570)
Developmentboard zum ECUcore-iMX35 / PLCcore-iMX35, Referenzdesigns usw.	Hardware Manual Developmentboard iMX35 (Manual-Nr.: L-1571)
Vollständige Beschreibung zum IEC 61131-Programmiersystem CODESYS, Grundlagen der SPS-Programmierung nach IEC 61131-3	Siehe: <a href="http://www.3S-software.com">http://www.3S-software.com</a>

**Abschnitt 4** dieses Manuals erläutert die **Inbetriebnahme des PLCcore-iMX35** auf Basis des Development Kit für das PLCcore-iMX35.

**Abschnitt 5** beschreibt die **Anschlussbelegung** des PLCcore-iMX35.

**Abschnitt 6** erklärt Details zur **Anwendung des PLCcore-iMX35**, so z.B. den **Aufbau des Prozessabbildes**, die **Bedeutung der Bedienelemente** und vermittelt grundlegende Informationen zur Programmierung des Moduls. Weiterhin werden hier Informationen zur Nutzung der CAN-Schnittstellen in Verbindung mit **CANopen** gegeben.

**Abschnitt 7** beschreibt **Details zur Konfiguration des PLCcore-iMX35**, so z.B. die Konfiguration von Ethernet- und CAN-Schnittstellen und den Linux-Autostartvorgang. Weiterhin wird hier die **Administration des PLCcore-iMX35** erläutert, so z.B. die Anmeldung am System, die Nutzerverwaltung und die Durchführung von Softwareupdates.

### 3 Produktbeschreibung

Das PLCcore-iMX35 erweitert die Produktpalette der Firma SYS TEC electronic GmbH im Steuerungsbereich um ein weiteres innovatives Produkt. In Form eines Aufsteckmoduls ("Core") stellt es dem Anwender eine vollständige Kompakt-SPS mit integrierter Target-Visualisierung zur Verfügung. Dank der CAN- und Ethernet-Schnittstellen ist das PLCcore-iMX35 optimal zum Aufbau von anwenderspezifischen HMI (**H**uman **M**achine Interface) Applikationen geeignet.



Bild 1: Ansicht des PLCcore-iMX35

Einige herausragende Merkmale des PLCcore-iMX35 sind:

- leistungsfähiger CPU-Kern (ARM 32-Bit ARM1136JF-S, 532 MHz CPU Takt, 740 MIPS)
- 128 MByte SDRAM Memory, 128 MByte FLASH Memory
- LCD Controller mit Unterstützung für Displays bis max. 800x600 Pixel Auflösung bei 24 Bit Farbtiefe
- Unterstützung für Scrollwheel und 4x4 Matrixtastatur
- 1x 10/100 Mbps Ethernet LAN Interface (mit on-board PHY)
- 2x CAN 2.0B Interface, nutzbar als CANopen-Manager (CiA 302 konform)
- 3x Asynchronous Serial Ports (UART)
- 16 digitale Eingänge, 10 digitale Ausgänge
- SPI und I<sup>2</sup>C extern nutzbar
- On-board Peripherie: RTC, Watchdog, Power-fail Eingang
- On-board Software: Linux, SPS-Firmware, CANopen-Master, HTTP- und FTP-Server Programmierbar nach IEC 61131-3 und in C/C++
- Support SPS-typischer Bedienelemente (z.B. Run/Stop-Schalter, Run-LED, Error-LED)
- Linux-basiert, dadurch weitere beliebige Anwenderprogramme parallel ausführbar
- Remote Login über Telnet
- Geringe Abmaße (78 \* 54 mm)

Die Verfügbarkeit einer SPS als aufsteckbares "Core" und die damit verbundenen geringen Abmessungen reduzieren den Aufwand und die Kosten bei der Entwicklung anwenderspezifischer Steuerungen enorm. Gleichzeitig eignet sich das PLCcore-iMX35 besonders als Basiskomponente für anwenderspezifische HMI Baugruppen sowie als intelligenter Netzwerkknoten zur dezentralen Verarbeitung von Prozesssignalen (CANopen und UDP).

Die On-Board-Firmware des PLCcore-iMX35 beinhaltet die gesamte Target-Visualisierung sowie die SPS-Laufzeitumgebung einschließlich der CANopen-Anbindung mit CANopen-Masterfunktionalität.

Dadurch ist das Modul in der Lage, sowohl Aufgaben der Mensch-Maschine-Kommunikation als auch Steuerungsaufgaben wie die Verknüpfung von Ein- und Ausgängen oder die Umsetzung von Regelalgorithmen vor Ort durchzuführen. Über das CANopen-Netzwerk, über Ethernet (UDP-Protokoll) sowie über die seriellen Schnittstellen (UART) können Daten und Ereignisse mit anderen Knoten (z.B. übergeordnete Zentralsteuerung, I/O-Slaves usw.) ausgetauscht werden. Darüber hinaus ist die Anzahl der Ein- und Ausgänge sowohl lokal als auch dezentral mit Hilfe von CANopen-Geräten erweiterbar. Ideal geeignet ist hierfür der CANopen-Chip, der ebenfalls als Aufsteckmodul für den Einsatz in anwenderspezifischen Applikationen konzipiert wurde.

Das PLCcore-iMX35 bietet 16 digitale Eingänge (DI0...DI15, 3.3V-Pegel), 10 digitale Ausgänge (DO0...DO9, 3.3V-Pegel) sowie Unterstützung für das Scrollwheel. Die Ablage des SPS-Programms in der On-board Flash-Disk des Moduls ermöglicht den autarken Wiederanlauf nach einer Spannungsunterbrechung.

Die Programmierung des PLCcore-iMX35 erfolgt nach IEC 61131-3 mit dem Programmiersystem *CODESYS* der Firma 3S Smart Software Solutions GmbH (<http://www.3s-software.com>). Damit kann das PLCcore-iMX35 sowohl grafisch in KOP/FUP, AS und CFC als auch textuell in AWL oder ST programmiert werden. Der Download des SPS-Programms auf das Modul erfolgt über Ethernet. Die Adressierung der Ein- und Ausgänge – und damit der Aufbau des Prozessabbildes – wurde an das Schema der SYS TEC-Kompaktsteuerungen angelehnt. Analog zu allen anderen SYS TEC-Steuerungen unterstützt auch das PLCcore-iMX35 als Debugfunktionalität das Beobachten und Setzen von Variablen, Einzelzyklus, Breakpoints und Einzelschritt.

Die integrierte Target-Visualisierung des PLCcore-iMX35 basiert auf der Visualisierungs-Komponente von *CODESYS* der Firma 3S Smart Software Solutions GmbH (<http://www.3s-software.com>). Sie ermöglicht sowohl die Anzeige von Prozesswerten aus der SPS als auch die Weiterleitung von Benutzeraktionen an die SPS (z.B. Eingaben über Touchscreen und Scrollwheel).

Das PLCcore-iMX35 basiert auf einem Embedded Linux als Betriebssystem. Dadurch ist es möglich, simultan zur SPS-Firmware noch weitere, anwenderspezifische Programme abzuarbeiten.

Das auf dem PLCcore-iMX35 eingesetzte Embedded Linux ist unter der GNU General Public License, Version 2 lizenziert. Der entsprechende Lizenztext ist im Anhang B aufgeführt. Die vollständigen Quellen des verwendeten LinuxBSP sind im Softwarepaket **SO-1121** ("VMware-Image des Linux-Entwicklungssystems für das ECUCore-iMX35") enthalten. Sollten Sie die Quellen des LinuxBSP unabhängig vom VMware-Image des Linux-Entwicklungssystems benötigen, setzen Sie sich bitte mit unserem Support in Verbindung:

[support@systec-electronic.com](mailto:support@systec-electronic.com)

Das verwendete SPS-System sowie die vom Anwender entwickelten SPS- und C/C++ Programme unterliegen **nicht** der GNU General Public License!



## 4 Development Kit PLCcore-iMX35

### 4.1 Übersicht

Das Development Kit PLCcore-iMX35 ist ein leistungsfähiges Komplettpaket zu einem besonders günstigen Preis, um auf Basis einer Kompakt-SPS mit integrierter Target-Visualisierung eigene, anwenderspezifische HMI-Geräte zu entwickeln.



Bild 2: Development Kit PLCcore-iMX35

Das Development Kit PLCcore-iMX35 gewährleistet eine rasche und problemlose Inbetriebnahme des PLCcore-iMX35. Es vereint dazu alle notwendigen Hard- und Software-Komponenten, die zur Erstellung eigener HMI Applikationen erforderlich sind: als Kernkomponente das PLCcore-iMX35 selbst, das zugehörige Developmentboard mit QVGA LCD Display, I/O-Peripherie und zahlreichen Schnittstellen, das IEC 61131-Programmiersystem CODESYS sowie weiteres Zubehör. Damit bildet das Developmentkit die ideale Plattform zur Entwicklung anwenderspezifischer HMI-Applikationen auf Basis des PLCcore-iMX35. Ebenso ermöglicht das Kit den kostengünstigen Einstieg in die Welt der dezentralen Automatisierungstechnik. Bereits die im Kit enthaltenen Komponenten ermöglichen die Erweiterung der Ein- und Ausgänge des PLCcore-iMX35 durch CANopen-I/O-Baugruppen. Damit kann das Development Kit auch für Projekte eingesetzt werden, die eine SPS mit Netzwerkanbindung erfordern.

Das Development Kit PLCcore-iMX35 beinhaltet folgende Komponenten:

- PLCcore-iMX35
- Developmentboard für PLCcore-iMX35, incl.:
  - 320x240 Pixel QVGA LCD Display
  - Scrollwheel (on-board)
  - 12V – 1,5A Steckernetzteil
- Ethernet-Kabel
- RS232-Kabel
- RS485-Adapter
- CD mit Programmiersoftware, Beispielen, Dokumentation und weiteren Tools

Das im Kit enthaltene Developmentboard ermöglicht eine schnelle Inbetriebnahme des PLCcore-iMX35 und vereinfacht den Aufbau von Prototypen anwenderspezifischer HMI Applikationen, die auf

diesem Modul basieren. Das Developmentboard besitzt u.a. verschiedene Möglichkeiten der Spannungszufuhr, ein 320x240 Pixel QVGA LCD Display, Ethernet-Schnittstelle, CAN-Schnittstelle, 4 Taster und 4 LEDs als Bedienelemente für die digitalen Ein- und Ausgänge sowie Scrollwheel und Anschluss für eine 4x4 Matrixtastatur. Die an den Steckverbindern des PLCcore-iMX35 verfügbaren Signale sind auf Stiftleisten geführt und ermöglichen so einen einfachen Anschluss eigener Peripherie-Schaltungen. Damit bildet das Developmentboard gleichzeitig eine ideale Experimentier- und Testplattform für das PLCcore-iMX35.

Als Software-Entwicklungsplattform sowie Debugumgebung für das PLCcore-iMX35 dient das im Kit enthaltene IEC 61131-Programmiersystem *CODESYS*. Dadurch kann das Modul sowohl grafisch in KOP/FUP, AS und CFC als auch textuell in AWL oder ST programmiert werden. Der Download des SPS-Programms auf das PLCcore-iMX35 erfolgt über Ethernet. Leistungsfähige Debug-Funktionalitäten wie das Beobachten und Setzen von Variablen, Einzelzyklus, Breakpoints und Einzelschritt erleichtern die Entwicklung und Inbetriebnahme von Anwendersoftware für das Modul.

## 4.2 Elektrische Inbetriebnahme des Development Kit PLCcore-iMX35

Das für den Betrieb des Development Kit PLCcore-iMX35 notwendige Steckernetzteil sowie die erforderlichen Ethernet- und RS232-Kabel sind bereits im Lieferumfang des Kit enthalten. Für die Inbetriebnahme des Kit sind mindestens die Anschlüsse für Versorgungsspannung (X100/X101), COM0 (X701A) und ETH0 (X702) erforderlich. Darüber hinaus wird der Anschluss von CAN0 (X801A) empfohlen. Einen vollständigen Überblick über die Anschlüsse des Development Kit PLCcore-iMX35 vermittelt Tabelle 2.

Tabelle 2: Anschlüsse des Development Kit PLCcore-iMX35

Anschluss	Bezeichnung auf Developmentboard	Bemerkung
Versorgungsspannung	X100 oder X101	Das im Lieferumfang enthaltene Steckernetzteil ist zum direkten Anschluss an X101 vorgesehen
ETH0 (Ethernet)	X702	Schnittstelle dient zur Kommunikation mit Programmier-PC sowie zum Programmdownload, darüber hinaus für das Anwenderprogramm frei verfügbar
COM0 (RS232)	X701A	Schnittstelle wird zur Konfiguration der Baugruppe (z.B. Setzen der IP-Adresse) benötigt und ist im normalen Betrieb für das Anwenderprogramm frei verfügbar
COM1 (RS232)	X701B	Schnittstelle ist für das Anwenderprogramm frei verfügbar.
COM2 (RS485)	X700	Schnittstelle ist für das Anwenderprogramm frei verfügbar.
CAN0 (CAN)	X801A	Schnittstelle dient zur Kommunikation mit Programmier-PC und ist darüber hinaus für das Anwenderprogramm frei verfügbar
CAN1	X801B	Schnittstelle ist für das Anwenderprogramm frei verfügbar.

Bild 3 zeigt die Lage der wichtigsten Anschlüsse auf dem Developmentboard für das PLCcore-iMX35. Anstelle des mitgelieferten Steckernetzteiles kann die Stromversorgung des Kit optional auch über X100 mit einer externen Quelle von 12V/1,5A erfolgen.

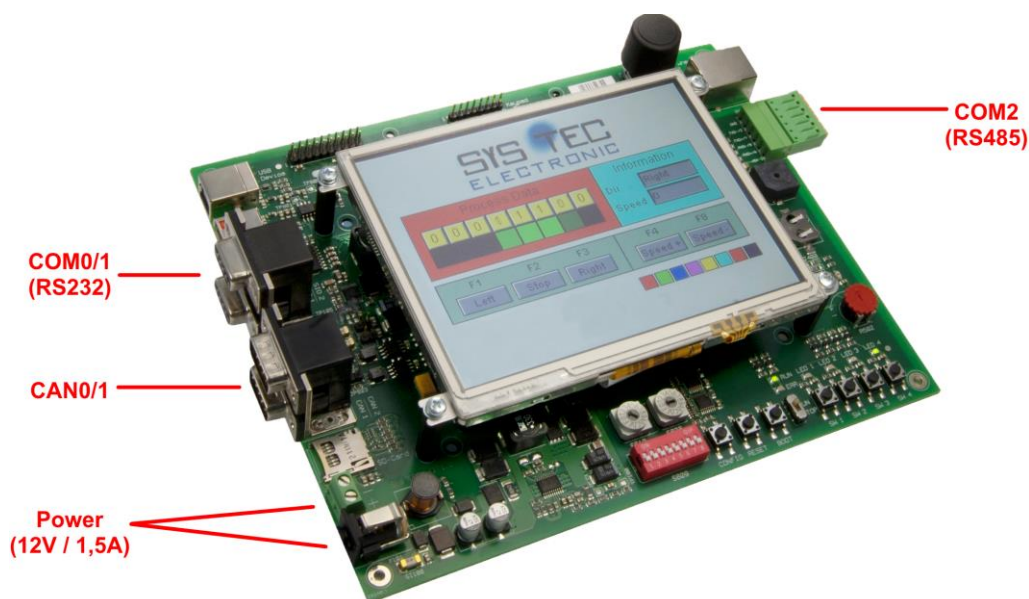


Bild 3: Lage der wichtigsten Anschlüsse auf dem Developmentboard für das PLCcore-iMX35

**Hinweis:** Bei der Inbetriebnahme sind zunächst die Kabel für Ethernet (ETH0, X702) und RS232 (COM0, X701A) anzuschließen, erst danach ist die Versorgungsspannung (X100 / X101) zu aktivieren.

### 4.3 Bedienelemente des Development Kit PLCcore-iMX35

Das Development Kit PLCcore-iMX35 ermöglicht eine einfache Inbetriebnahme des PLCcore-iMX35. Es verfügt über verschiedene Bedienelemente sowohl zur Konfiguration des Moduls als auch zur Simulation von Ein- und Ausgängen für den Einsatz des PLCcore-iMX35 als SPS-Kern. Tabelle 3 listet die einzelnen Bedienelemente des Developmentboard auf und beschreibt deren Bedeutung.

Tabelle 3: Bedienelemente des Developmentboard für das PLCcore-iMX35

Bedienelement	Bezeichnung	Bedeutung
Taster 0	S604	Digitaler Eingang DI0 (Prozessabbild: %IX0.0)
Taster 1	S605	Digitaler Eingang DI1 (Prozessabbild: %IX0.1)
Taster 2	S606	Digitaler Eingang DI2 (Prozessabbild: %IX0.2)
Taster 3	S607	Digitaler Eingang DI3 (Prozessabbild: %IX0.3)
LED 0	D602	Digitaler Ausgang DO0 (Prozessabbild: %QX0.0)
LED 1	D603	Digitaler Ausgang DO1 (Prozessabbild: %QX0.1)
LED 2	D604	Digitaler Ausgang DO2 (Prozessabbild: %QX0.2)
LED 3	D605	Digitaler Ausgang DO3 (Prozessabbild: %QX0.3)
Run/Stop-Schalter	S603	Run / Stop für Abarbeitung des SPS-Programms, Rücksetzen der Steuerung

Bedienelement	Bezeichnung	Bedeutung
Run-LED	D600	Anzeige Aktivitätszustand der SPS
Error-LED	D601	Anzeige Fehlerzustand der SPS
Hexcodier-Schalter	S608/S610	Konfiguration der Knotenadresse CAN0
DIP-Schalter	S609	Konfiguration Bitrate und Master-Modus CAN0

Eine vollständige Auflistung des Prozessabbildes enthält Tabelle 6 im Abschnitt 6.4.1.

## 4.4 Optionales Zubehör

### 4.4.1 USB-RS232 Adapter Kabel

Das SYS TEC USB-RS232 Adapter Kabel (Bestellnummer 3234000) stellt eine RS232-Schnittstelle über einen USB-Port des PC zur Verfügung. In Verbindung mit einem Terminalprogramm ermöglicht es die Konfiguration des PLCcore-iMX35 von PCs, wie z.B. Laptops, die selber keine physikalische RS232-Schnittstelle mehr besitzen (siehe Abschnitt 7.1).

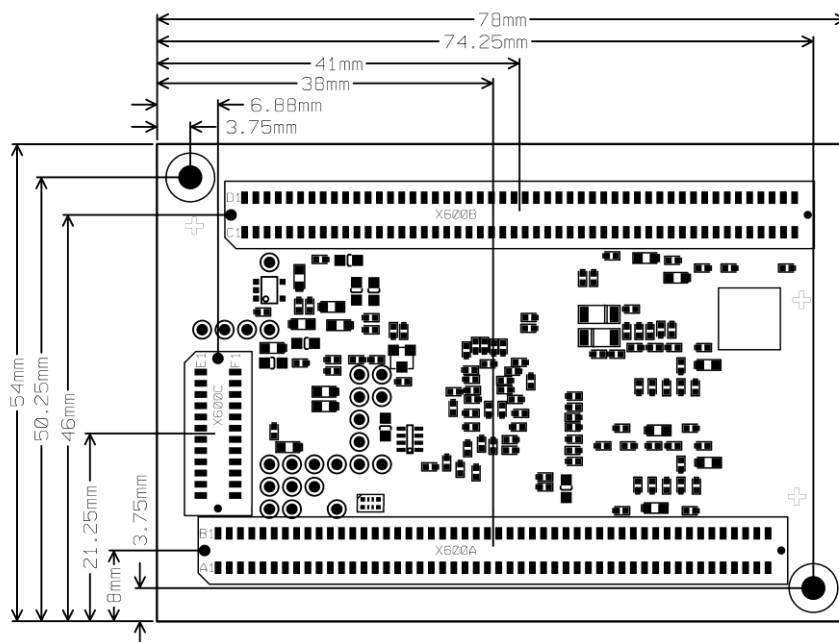


*Bild 4: SYS TEC USB-RS232 Adapter Kabel*

## 5 Anschlussbelegung des PLCcore-iMX35

Die Anschlüsse des PLCcore-iMX35 sind über zwei auf der Modulunterseite montierte, doppelreihige Buchsenleisten nach außen geführt (X800A/B, siehe Bild 5). Die entsprechenden Steckverbinder als Gegenstücke zur Aufnahme des PLCcore-iMX35 sind bei der Firma "W + P" unter folgender Bezeichnung lieferbar:

W+P-Bezeichnung: SMT-Stiftleisten RM 1,27mm, stehend, 2-reihig - 1,00mm  
 W+P-Bestellnummer: 7072-100-10-00-10-PPST (auch in anderen Höhen lieferbar)



SYS TEC electronic GmbH  
 4348.0 - ECUcore iMX35

Bild 5: Pinout des PLCcore-iMX35 - top view

Bild 5 verdeutlicht die Lage der Buchsenleisten (X600A/B) auf dem PLCcore-iMX35, Tabelle 4 listet die vollständige Anschlussbelegung des Moduls auf. Die im Bild 5 zusätzlich aufgeführte Buchsenleiste X600C dient zum Anschluss eines JTAG-Interfaces und ist nur auf speziellen Entwicklungsmodulen bestückt. Für die Verwendung des PLCcore-iMX35 als SPS-Kern ist das JTAG-Interface ohne Bedeutung. Eine ausführliche Beschreibung der Modulanschlüsse beinhaltet das Hardware Manual ECUcore-iMX35 (Manual-Nr.: L-1570). Referenzdesigns zur Anwendung des PLCcore-iMX35 in anwenderspezifischen Applikationen enthält Anhang A.

Tabelle 4: Anschlussbelegung des PLCcore-iMX35, vollständig, sortiert nach Anschluss-Pin

Signal	Pin	Pin	Signal	Signal	Pin	Pin	Signal
GND	A01	B01	GND	GND	C01	D01	2V5_EPHY
/BOOT	A02	B02	/MR	Eth_Tx-	C02	D02	GND
/BOOTSTRAP_1	A03	B03	/RESET_IN	Eth_Tx+	C03	D03	Speed
VSTBY	A04	B04	/PFI	Eth_Rx+	C04	D04	Link/Act
/BOOTSTRAP_0	A05	B05	WDI	Eth_Rx-	C05	D05	GND
GND	A06	B06	/PFO	GND	C06	D06	GPIO1_6

Signal	Pin	Pin	Signal	Signal	Pin	Pin	Signal
RXD1	A07	B07	GND	GPIO1_0	C07	D07	GPIO1_5
TXD1	A08	B08	RTS2	GPIO1_1	C08	D08	GPIO1_4
RTS1	A09	B09	CTS2	GND	C09	D09	GND
CTS1	A10	B10	RTS3	SD2_DATA0	C10	D10	GPIO1_3
GPIO2_12	A11	B11	CTS3	SD2_DATA1	C11	D11	USBOTG_OC
GND	A12	B12	GND	SD2_DATA2	C12	D12	USBOTG_PWR
TXD2	A13	B13	TXD3	SD2_DATA3	C13	D13	USBPHY1_VBUS
RXD2	A14	B14	RXD3	SD2_CLK	C14	D14	SD2_CMD
NVCC_3V3	A15	B15	GPIO1_26	GND	C15	D15	GPIO1_31
NVCC_3V3	A16	B16	GPIO2_18	GPIO1_28	C16	D16	GND
GND	A17	B17	Nicht verwendet	Nicht verwendet	C17	D17	CAPTURE
USBPHY2_DP	A18	B18	Nicht verwendet	GPIO2_12	C18	D18	GPIO1_25
USBPHY2_DM	A19	B19	GND	GPIO2_13	C19	D19	COMPARE
USBPHY1_UID	A20	B20	USBPHY1_DP	GPIO2_14	C20	D20	CLKO
Nicht verwendet	A21	B21	USBPHY1_DM	GPIO2_15	C21	D21	GPIO2_26
GND	A22	B22	GND	GPIO2_17	C22	D22	GPIO2_28
I2C2_DAT	A23	B23	CAN1_TX	GPIO2_25	C23	D23	GND
I2C2_CLK	A24	B24	CAN1_RX	GND	C24	D24	BACKL_EN
GND	A25	B25	GPIO1_24	GPIO1_2	C25	D25	GPIO2_31
GPIO3_25	A26	B26	GND	LCD_CONTRAST	C26	D26	GPIO2_29
GND3_26	A27	B27	/RESET	GPIO2_30	C27	D27	GPIO2_19
Nicht verwendet	A28	B28	/PORESET	GPIO2_20	C28	D28	GND
/EN_IO3V3	A29	B29	Nicht verwendet	GND	C29	D29	GPIO2_21
Nicht verwendet	A30	B30	CAN2_RX	GPIO2_22	C30	D30	Nicht verwendet
GND	A31	B31	CAN2_TX	LCD_TXout0+	C31	D31	LCD_TXout0-
CSPI1_SS0	A32	B32	GND	LCD_TXout1+	C32	D32	LCD_TXout1-
CSPI1_SS2/PWMO	A33	B33	CSPI1_SS1	LCD_TXout2+	C33	D33	GND
CSPI1_MOSI	A34	B34	CSPI1_MISO	LCD_TXout2-	C34	D34	LCD_TXoutCLK+
CSPI1_SS3	A35	B35	CSPI1_SCLK	GND	C35	D35	LCD_TXoutCLK-
Nicht verwendet	A36	B36	Nicht verwendet	LCD_R0	C36	D36	LCD_R1
GND	A37	B37	SD1_CLK	LCD_R2	C37	D37	LCD_R3
SD1_CMD	A38	B38	GND	LCD_R4	C38	D38	LCD_R5
SD1_DATA0	A39	B39	SD1_DATA1	LCD_G0	C39	D39	GND
SD1_DATA2	A40	B40	SD1_DATA3	LCD_G1	C40	D40	LCD_G2
MATRIX_C1	A41	B41	MATRIX_C0	GND	C41	D41	LCD_G3
MATRIX_C3	A42	B42	MATRIX_C2	LCD_G4	C42	D42	LCD_G5
GND	A43	B43	MATRIX_R0	LCD_B0	C43	D43	LCD_B1
MATRIX_R1	A44	B44	GND	LCD_B2	C44	D44	LCD_B3
MATRIX_R3	A45	B45	MATRIX_R2	LCD_B4	C45	D45	GND
GPIO1_8	A46	B46	GPIO1_9	LCD_B5	C46	D46	/LVDS_PWD
GPIO1_12	A47	B47	GPIO1_13	GND	C47	D47	GPIO1_15
VBAT	A48	B48	GPIO1_14	LCD_DEN	C48	D48	LCD_DCLK
GND	A49	B49	GND	LCD_HSYNC	C49	D49	LCD_VSYNC
+3V3	A50	B50	+3V3	GND	C50	D50	GND

Tabelle 5 beinhaltet als Untermenge von Tabelle 4 nur alle Ein- und Ausgänge des PLCcore-iMX35, sortiert nach deren Funktion.

Tabelle 5: Anschlussbelegung des PLCcore-iMX35, nur I/O, sortiert nach Funktion

Connector	I/O-Pin	PLC Function 1	PLC Function 2 A=alternative, S=simultaneous
D27	GPIO2_19	DI0 [Switch0]	
C28	GPIO2_20	DI1 [Switch1]	
D29	GPIO2_21	DI2 [Switch2]	
C30	GPIO2_22	DI3 [Switch3]	
B25	GPIO1_24	DI4	
D18	GPIO1_25	DI5	
B15	GPIO1_26	DI6	
D15	GPIO1_31	DI7	
D14	GPIO2_0 (SD2_CMD)	DI8	
A47	GPIO1_12	DI9	
B16	GPIO2_18	DI10	
B9	GPIO3_13	DI11	
A26	GPIO3_25	DI12	
A27	GPIO3_26	DI13	
D21	GPIO2_26	DI14	
C25	GPIO1_2	DI15	
D22	GPIO2_28	DO0 [LED0]	
D26	GPIO2_29	DO1 [LED1]	
C27	GPIO2_30	DO2 [LED2]	
D25	GPIO2_31	DO3 [LED3]	
C7	GPIO1_0	DO4	
C8	GPIO1_1	DO5	
D10	GPIO1_3	DO6	
D6	GPIO1_6	DO7	
C14	GPIO2_1 (SD2_CLK)	DO8	
C21	GPIO2_15	DO9	
B41	MATRIX_C0	MATRIX_C0	
A41	MATRIX_C1	MATRIX_C1	
B42	MATRIX_C2	MATRIX_C2	
A42	MATRIX_C3	MATRIX_C3	
B43	MATRIX_R0	MATRIX_R0	
A44	MATRIX_R1	MATRIX_R1	
B45	MATRIX_R2	MATRIX_R2	
A45	MATRIX_R3	MATRIX_R3	
D17	CAPTURE	Scrollwheel DIR	
D19	COMPARE	Scrollwheel CLK	
C22	GPIO2_17	Scrollwheel Button	
A46	GPIO1_8	Error-LED	

Ist für den Einsatz des PLCcore-iMX35 auf einer anwenderspezifischen Basisplatine kein Run/Stop-Schalter vorgesehen, muss an den Modulanschlüssen die Codierung für "Run" fest verdrahtet sein (siehe dazu auch Referenzdesign im Anhang A).

## 6 SPS-Funktionalität des PLCcore-iMX35

### 6.1 Übersicht

Das PLCcore-iMX35 realisiert eine vollständige, Linux-basierte Kompakt-SPS in Form eines Aufsteckmoduls ("Core"). Das PLCcore-iMX35 basiert dabei auf der Hardware des ECUcore-iMX35 und erweitert dieses um SPS-spezifische Funktionalitäten (SPS-Firmware, Target-Visualisierung). Beide Module, sowohl ECUcore-iMX35 als auch PLCcore-iMX35, benutzen dasselbe Embedded Linux als Betriebssystem. Folglich sind auch Konfiguration und C/C++ Programmierung des PLCcore-iMX35 weitestgehend identisch zum ECUcore-iMX35.

### 6.2 Systemstart des PLCcore-iMX35

Standardmäßig lädt das PLCcore-iMX35 nach Power-on bzw. Reset alle notwendigen Firmware-Komponenten und startet anschließend die Abarbeitung des SPS-Programms. Damit eignet sich das PLCcore-iMX35 für den Einsatz in autarken Steuerungssystemen, die nach einer Spannungsunterbrechung selbständig und ohne Benutzeraktionen die Ausführung des SPS-Programms wieder aufnehmen. Bild 6 zeigt den Systemstart im Detail.

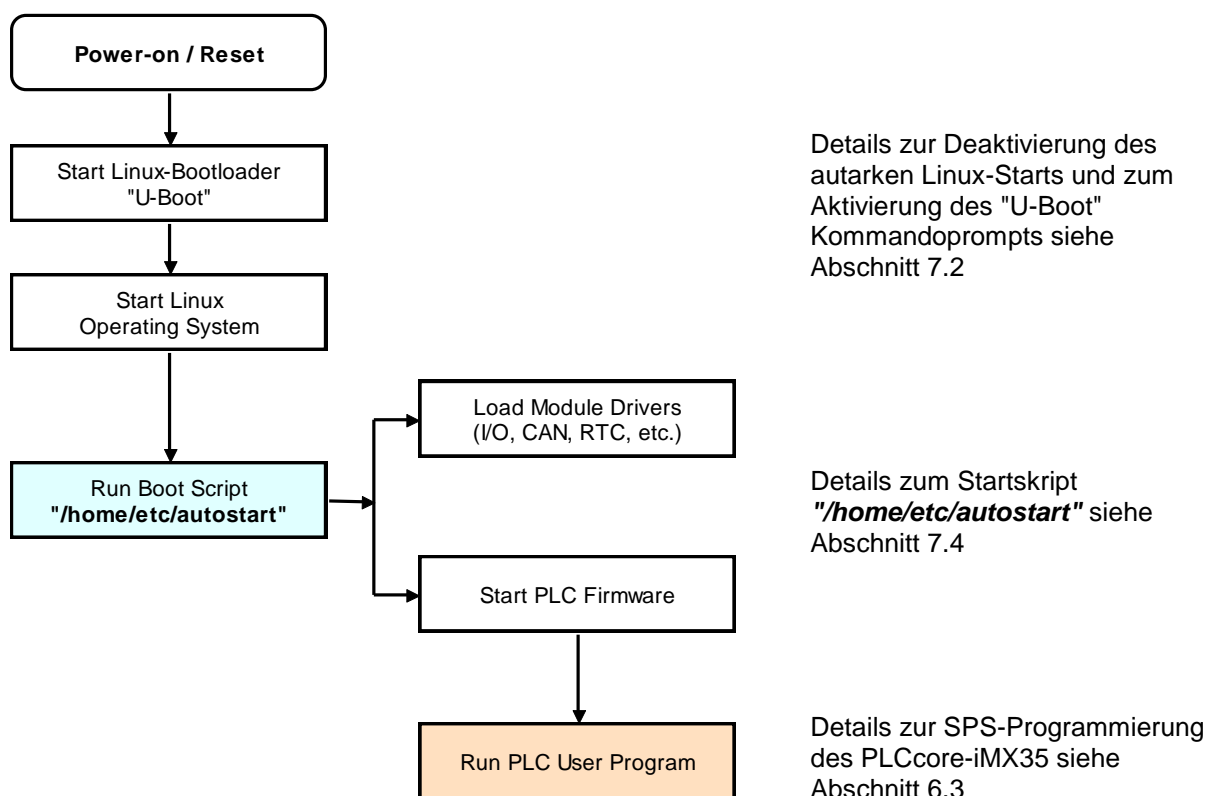


Bild 6: Systemstart des PLCcore-iMX35



## 6.3 Programmierung des PLCcore-iMX35

Das PLCcore-iMX35 wird mit der IEC 61131-3-konformen Programmierumgebung *CODESYS* programmiert. Zu *CODESYS* existieren eigene Handbücher, die den Umgang mit diesem Programmierwerkzeug beschreiben. Tabelle 1 listet die für das PLCcore-iMX35 relevanten Manuals auf.

Die Firmware des PLCcore-iMX35 basiert auf der Standardfirmware für CODESYS-Laufzeitsysteme der Firma 3S Smart Software Solutions GmbH (<http://www.3S-software.com>). Dies betrifft insbesondere den Aufbau des Prozessabbildes (siehe Abschnitt 6.4) sowie die Funktionsweise der Bedienelemente (Hexcodier-Schalter, DIP-Schalter usw.).

Detaillierte Informationen zu Nutzung der CAN-Schnittstellen in Verbindung mit CANopen beschreibt Abschnitt 6.6.

## 6.4 Prozessabbild des PLCcore-iMX35

### 6.4.1 Lokale Ein- und Ausgänge

Im Unterschied zu den anderen SYS TEC-Kompaktsteuerungen wird der Aufbau des Prozessabbildes von der Struktur des Geräte-Baumes des anwenderspezifischen CODESYS-Projektes bestimmt. Vom PLCcore-iMX35 werden die in Tabelle 6 aufgelisteten Ein- und Ausgänge unterstützt.

Die in Tabelle 6 angegebenen Adressen im Prozessabbild gelten für den Fall, dass der Treiber der lokalen Ein- und Ausgänge als erster in der Hierarchie des Gerätebaums eingehängt ist. Verschiebungen der Adressen der Ein- und Ausgänge können vorkommen, wenn diese Hierarchie verletzt wird (bspw. durch vorheriges Einbinden von CANopen-Peripherie).

Tabelle 6: Zuordnung der Ein- und Ausgänge zum Prozessabbild des PLCcore-iMX35

I/O des PLCcore-iMX35	Adresse und Datenformat im Prozessabbild
DI0 ... DI7	<b>%IB0.0</b> als Byte mit DI0 ... DI7 <b>%IX0.0 ... %IX0.7</b> als einzelne Bits für jeden Eingang
DI8 ... DI15	<b>%IB1.0</b> als Byte mit DI8 ... DI15 <b>%IX1.0 ... %IX1.7</b> als einzelne Bits für jeden Eingang
AIN0	<b>%IW8.0</b> 15Bit + sign(0 ... + 32767)
AIN1	<b>%IW10.0</b> 15Bit + sign(0 ... + 32767)
AIN2	<b>%IW12.0</b> 15Bit + sign(0 ... + 32767)
AIN3	<b>%IW14.0</b> 15Bit + sign(0 ... + 32767)
On-board Temperatursensor	<b>%ID72.0</b> 31Bit + Vorzeichen als 1/10000 °C
DO0 ... DO7	<b>%QB0.0</b> als Byte mit DO0 ... DO7 <b>%QX0.0 ... %QX0.7</b> als einzelne Bits für jeden Ausgang
DO8 ... DO9	<b>%QB1.0</b> als Byte mit DO8 ... DO9 <b>%QX1.0</b> als einzelne Bits für jeden Ausgang

Die Ein- und Ausgänge des PLCcore-iMX35 werden nicht negiert im Prozessabbild verwaltet, d.h. ein H-Pegel an einem Eingang führt zum Wert "1" an der entsprechenden Adresse im Prozessabbild und umgekehrt führt der Wert "1" im Prozessabbild zu einem H-Pegel am zugehörigen Ausgang.

## 6.5 Kommunikationsschnittstellen

### 6.5.1 Serielle Schnittstellen

Das PLCcore-iMX35 besitzt 3 serielle Schnittstellen. Das sind COM0 und COM1, die als RS-232 ausgeführt sind, und COM2, die als RS485-Schnittstelle ausgeführt ist. Details zur Hardwareanschaltung beschreibt das *"Hardware Manual Development Board ECUcore-iMX35"* (Manual-Nr.: L-1571).

**COM0:** Die Schnittstelle COM0 dient primär als Serviceschnittstelle zur Administration des PLCcore-iMX35. Sie wird standardmäßig im Boot-Skript *"/etc/inittab"* dem Linux-Prozess *"getty"* zugeordnet und als Linux-Konsole zur Administration des PLCcore-iMX35 benutzt.

**COM1/2:** Die Schnittstellen COM1/2 sind frei verfügbar und dienen in der Regel zum Datenaustausch zwischen PLCcore-iMX35 und anderen Feldgeräten unter Kontrolle des SPS-Programms.

### 6.5.2 CAN-Schnittstellen

Das PLCcore-iMX35 besitzt 2 CAN-Schnittstellen (CAN0, CAN1). Details zur Hardwareanschaltung beschreibt das *"Hardware Manual Development Board ECUcore-iMX35"* (Manual-Nr.: L-1571).

Die CAN-Schnittstellen ermöglichen den Datenaustausch mit anderen Geräten über Netzwerkvariablen.

Detaillierte Informationen zur Nutzung der CAN-Schnittstellen in Verbindung mit CANopen beschreibt Abschnitt 6.6.

### 6.5.3 Ethernet-Schnittstellen

Das PLCcore-iMX35 besitzt 1 Ethernet-Schnittstelle (ETH0). Details zur Hardwareanschaltung beschreibt das *"Hardware Manual Development Board ECUcore-iMX35"* (Manual-Nr.: L-1571).

Die Ethernet-Schnittstelle dient als Serviceschnittstelle zur Administration des PLCcore-iMX35 und als Programmierschnittstelle zum Download von CODESYS-Applikationen sowie zum Monitoring und Debugging mit dem Programmiersystem von CODESYS vom PC aus.

## 6.6 Nutzung der CAN-Schnittstellen mit CANopen

Das PLCcore-iMX35 besitzt 2 CAN-Schnittstellen (CAN0, CAN1), die unter CODESYS alternativ als CANopen-Manager nutzbar sind (konform zum CiA Draft Standard 302). Es kann nur jeweils eine CAN-Schnittstelle als CANopen-Manager fungieren. Die andere ist dann noch verfügbar als Kommunikations-Schnittstelle über CAN-Layer2. Die Konfiguration einer Schnittstelle (aktiv/inaktiv, Knotennummer, Bitrate, Master an/aus) beschreibt Abschnitt 6.6.1.

Die CAN-Schnittstelle ermöglicht den Datenaustausch mit anderen Geräten über Netzwerkvariablen. Dazu sind der CAN-Bus, der CODESYS-CANopen-Manager und die über den CAN-Bus angeschlossenen CANopen-Feldgeräte im Geräte-Baum des CODESYS-Projekts einzuhängen (vgl. Abschnitt 6.6.1).

CANopen stellt mit den beiden Diensten **PDO** (*Process Data Objects*) und **SDO** (*Service Data Objects*) zwei unterschiedliche Mechanismen für den Datenaustausch zwischen den einzelnen Feldbusgeräten zur Verfügung. Die von einem Knoten gesendeten Prozessdaten (**PDO**) stehen als Broadcast gleichzeitig allen interessierten Empfängern zur Verfügung. Durch die Realisierung als quittungslose Broadcast-Nachrichten sind PDOs auf 1 CAN-Telegramm und damit auf maximal 8 Byte Nutzdaten limitiert. Im Gegensatz dazu basieren **SDO**-Transfers auf logischen Punkt-zu-Punkt-Verbindungen ("Peer to Peer") zwischen zwei Knoten und ermöglichen den quitierten Austausch von Datenpaketen, die auch größer als 8 Bytes sein können. Diese Datenpakete werden intern durch eine entsprechende Anzahl quittierter CAN-Telegramme übertragen. Beide Dienste sind für die Schnittstelle CAN0 des PLCcore-iMX35 nutzbar.

Für den PDO-basierten Datenaustausch werden ausschließlich Netzwerkvariablen verwendet. Netzwerkvariablen stellen die einfachste Form des Datenaustausches mit anderen CANopen-Knoten dar. In einem SPS-Programm erfolgt der Zugriff auf Netzwerkvariablen in derselben Form wie auf interne, lokale Variablen der SPS. Aus Sicht des SPS-Programmierers ist es somit völlig unbedeutend, ob z.B. eine Input-Variable einem lokalen Eingang der Steuerung zugeordnet ist oder einen Eingang eines dezentralen Erweiterungsmoduls repräsentiert. Die Verwendung von Netzwerkvariablen basiert auf der Einbindung von EDS-Dateien bzw. mit dem CANopen-Konfigurator vorgefertigte DCF-Dateien, die als Gerätebeschreibungs-Datei in das CODESYS-Gerätorepository aufgenommen und als Gerät in den Gerätebaum eingehängt werden (siehe auch Abschnitt 6.6.1). Mit den CODESYS-spezifischen Konfigurationstools für die CAN-Schnittstelle werden solche Einstellungen wie die Kommunikationsparameter eines Gerätes (CAN Identifier, usw.) und die Zuordnung der Netzwerkvariablen auf die einzelnen Bytes eines CAN-Telegramms (Mapping) vorgenommen. Die Anwendung von Netzwerkvariablen erfordert nur allgemeine Grundkenntnisse über CANopen.

Der Austausch von PDOs erfolgt in einem CANopen-Netzwerk nur im Zustand "*OPERATIONAL*". Befindet sich das PLCcore-iMX35 nicht in diesem Zustand, verarbeitet es keine PDOs (weder sende- noch empfangsseitig) und aktualisiert folglich auch nicht den Inhalt der Netzwerkvariablen. Das Setzen der Betriebszustände "*OPERATIONAL*", "*PRE-OPERATIONAL*" usw. ist Aufgabe des CANopen-Managers (meist auch als "CANopen-Master" bezeichnet). In typischen CANopen-Netzwerken wird für den CANopen-Manager ein programmierbarer Knoten - meist in Form einer SPS - verwendet. Das PLCcore-iMX35 kann optional die Aufgabe des CANopen-Managers übernehmen. Die Aktivierung des Managers beschreibt Abschnitt 6.6.1.

Als CANopen-Manager ist das PLCcore-iMX35 zudem in der Lage, die am CAN-Bus angeschlossenen CANopen I/O-Geräte ("CANopen-Slaves") zu parametrieren, indem es die vom CODESYS-CANopen-Konfigurator erstellten DCF-Dateien beim Systemstart per SDO auf die jeweiligen Knoten überträgt.



2. Diese dann unter den CANopen-Manager hängen (rechte Maustaste – Gerät anhängen)

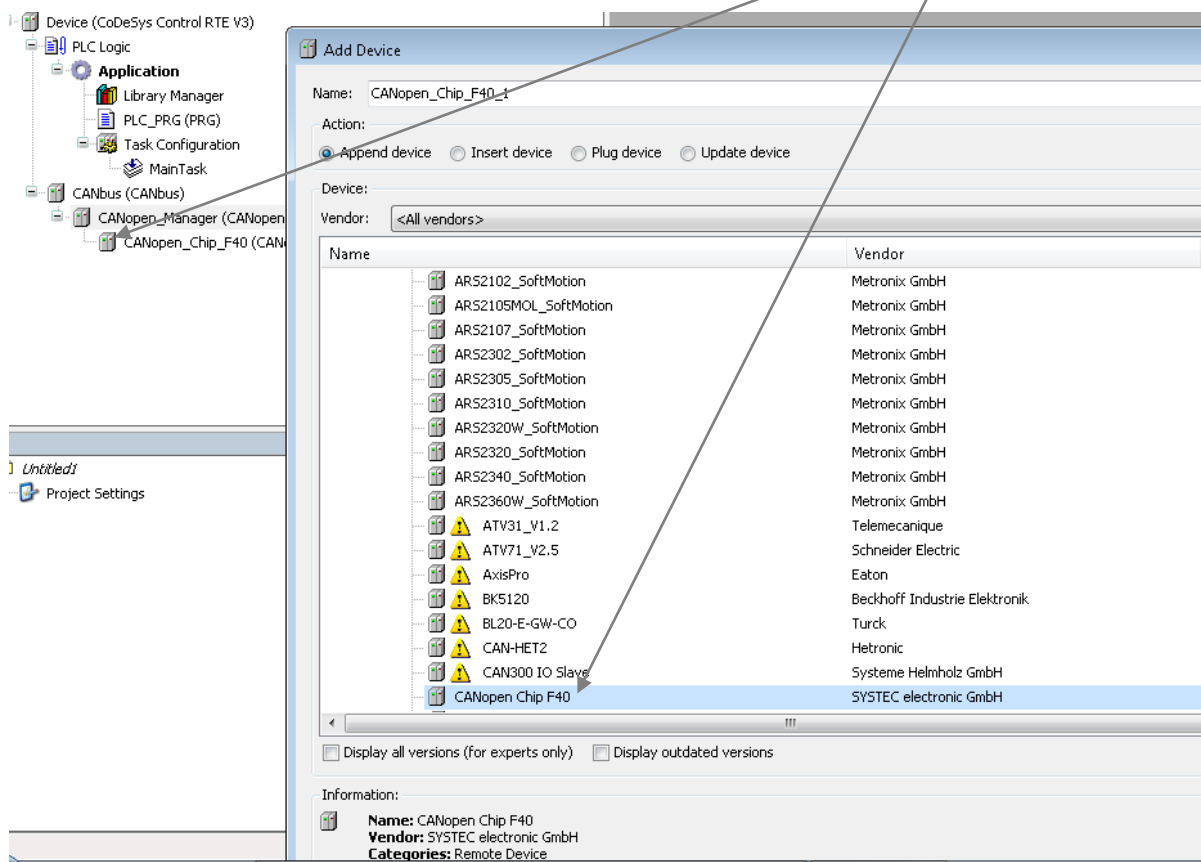


Bild 8: Anhängen des F40-Moduls als Gerät unterhalb des CANopen-Managers

Dann sieht die Konfiguration des Slaves so aus:

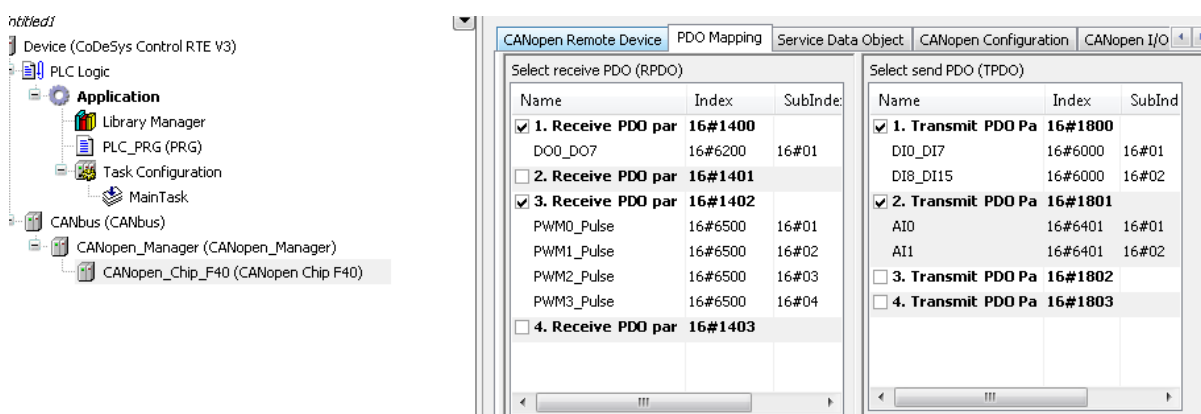


Bild 9: Ansicht des PDO-Mappings für das F40-Modul

### 6.6.1.2 Variablen-Mapping

Über den Reiter "CANopen I/O Abbild" ist dann noch das Mapping der Netzwerk-Variablen durchzuführen. Dazu gibt es 2 Möglichkeiten:

- Netzwerkvariable(n) als globale Variable im Projekt bekanntmachen – dazu die Spalte "**Variable**" benutzen
- Netzwerkvariable(n) auf eine im Projekt vorhandene Variable mappen – dazu die Spalte "**Mapping**" benutzen

Eine gleichzeitige Verwendung beider Möglichkeiten für ein und dieselbe Variable wird in CODESYS nicht zugelassen.

Ohne Mapping-Einträge am unterhalb des CANopen-Managers eingehängten CANopen-Gerät sind die I/Os über die IEC-Adressen des Prozessabbildes (%IBnn, %IWnn, ... %QBnn, %QWnn usw.) erreichbar. Im Variablendeklarationsteil eines SPS-Programms ist dazu wie gewohnt z. B. bei

```
MyVar AT%IBnn : BYTE;
```

für "nn" ist lediglich die entsprechende IEC-Adresse der Netzwerkvariablen anzugeben, wie sie für das CANopen-Gerät gemäß Position im Gerätebaum in der CODESYS-IDE automatisch vergeben worden ist.

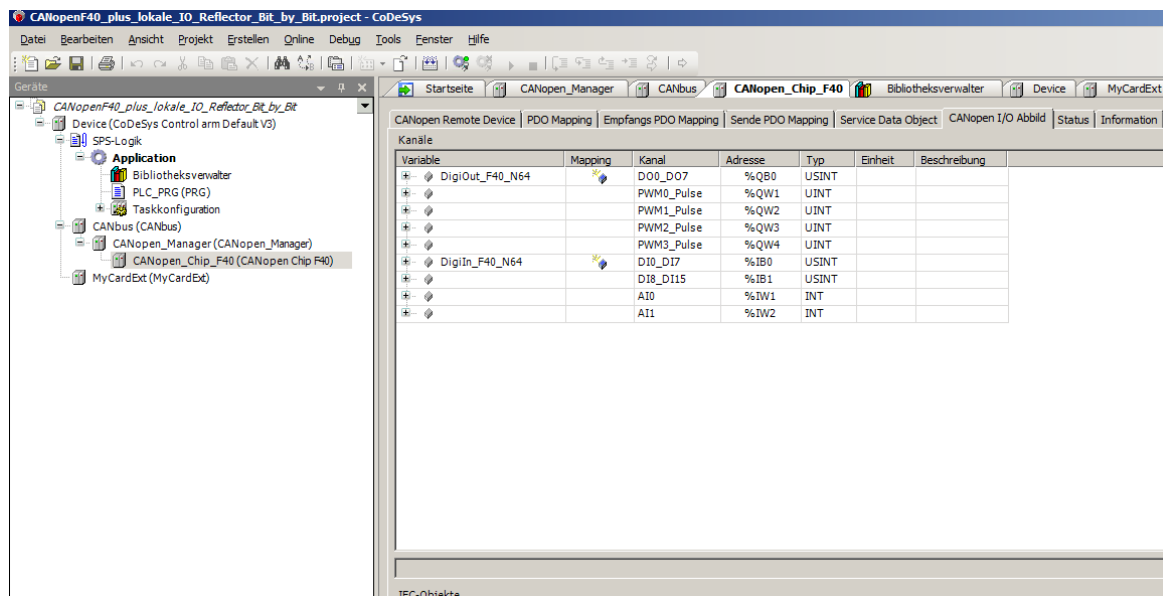


Bild 10: Netzwerkvariablen-Mapping des F40-Moduls

Im angegebenen Beispiel wurde die erste Möglichkeit gewählt.

Damit sind die konfigurierten CANopen-Netzwerkvariablen projektweit verfügbar und können z. B. wie folgt verwendet werden:

```
DigiOut_F40_N64.0 := DigiIn_F40_N64.0;
DigiOut_F40_N64.1 := DigiIn_F40_N64.1;
DigiOut_F40_N64.2 := DigiIn_F40_N64.2;
DigiOut_F40_N64.3 := DigiIn_F40_N64.3;
```

Das angegebene Codebeispiel reflektiert die Bits 0 ... 3 des Eingangsbytes DI0...DI7 des F40-Moduls auf die Bits 0 ... 3 des Ausgangsbytes DO0...DO7 des F40-Moduls.

## 6.6.2 Zusätzliche CAN-Schnittstellen

Die auf dem PLCcore-iMX35 eingesetzte SPS-Firmware ist prinzipiell in der Lage, mehrere CAN-Schnittstellen simultan zu bedienen (analog zu anderen SPS-Typen wie beispielsweise PLCcore-5484 oder PLCmodule-C32).

**Hinweis:** Eine Nutzung von mehr als einer CAN-Schnittstelle für CANopen ist unter CODESYS nicht möglich.

Bei Bedarf können weitere CAN-Schnittstellen extern an das Modul angeschlossen werden. Bitte setzen Sie sich bei Interesse dazu mit unserem Support in Verbindung:

[support@sys-tec-electronic.com](mailto:support@sys-tec-electronic.com)

## 6.7 Target-Visualisierung

Die Visualisierung und Bedienung mittels Touchscreen erfordert das Einhängen des Visualisierungs-Managers unter der Applikation. Unterhalb des Visualisierungs-Managers werden die benötigten Visualisierungs-Objekte eingehängt (je Visualisierungs-Bildschirm ein gesondertes Objekt – eine Hierarchie der Objekte untereinander gibt es hier nicht – siehe Bild 11), die mit Hilfe der zugehörigen Werkzeuge unter der CODESYS-Programmierschnittstelle konfiguriert und programmiert werden. Eine genaue Beschreibung findet man in den entsprechenden Handbüchern hierzu (<http://www.3S-software.com>) und in der Online-Hilfe der Programmierschnittstelle.

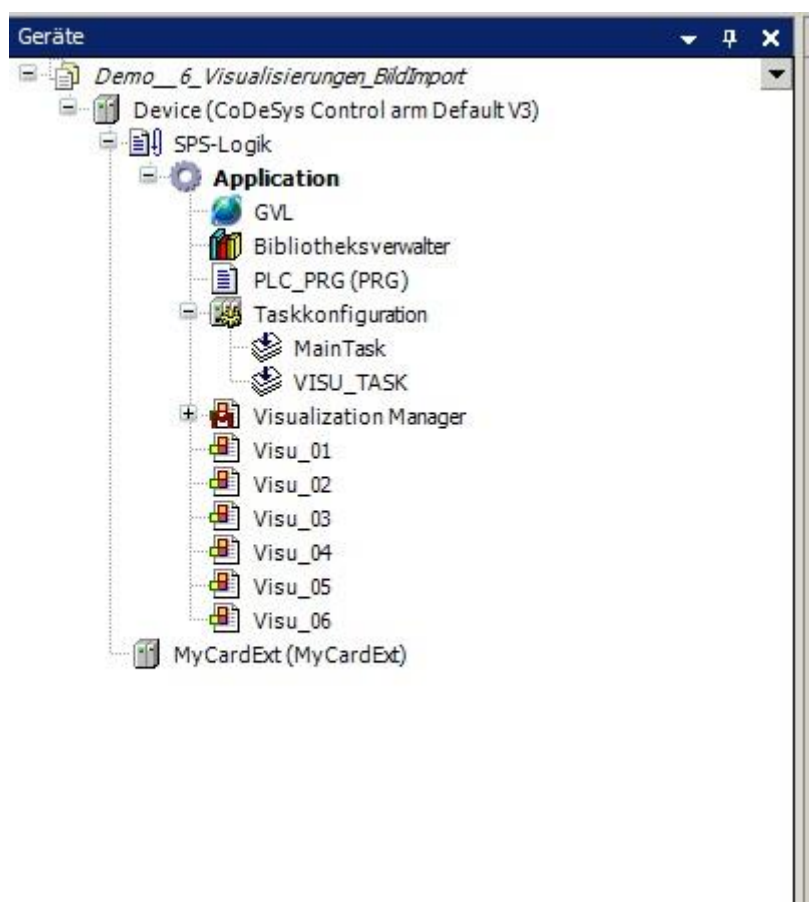


Bild 11: Gerätebaum bei Verwendung der Target-Visualisierung unter CODESYS

## 7 Konfiguration und Administration des PLCcore-iMX35

### 7.1 Systemvoraussetzungen und erforderliche Softwaretools

Zur Administration des PLCcore-iMX35 ist ein beliebiger Windows- oder Linux-PC erforderlich, der über eine Ethernet-Schnittstelle sowie eine serielle Schnittstelle (RS232) verfügt. Als Alternative zur seriellen on-board Schnittstelle eignet sich auch das von SYS TEC angebotenen USB-RS232 Adapter Kabel (Bestellnummer 3234000, siehe Abschnitt 4.4.1), das eine entsprechende RS232-Schnittstelle über einen USB-Port zur Verfügung stellt.

Alle in diesem Manual aufgeführten Beispiele beziehen sich auf die Administration des PLCcore-iMX35 von einem Windows-PC aus. Das Vorgehen auf einem Linux-PC ist analog.

Zur Administration des PLCcore-iMX35 sind folgende Softwaretools erforderlich:

**Terminalprogramm** Ein Terminalprogramm ermöglicht die Kommunikation mit der **Kommando-Shell** des PLCcore-iMX35 über eine **serielle RS232-Verbindung an COM0 des PLCcore-iMX35**. Diese ist Voraussetzung für die im Abschnitt 7.3 beschriebene Ethernet-Konfiguration des PLCcore-iMX35. Nach Abschluss der Ethernet-Konfiguration können alle weiteren Kommandos wahlweise entweder auch weiterhin im Terminalprogramm eingegeben werden oder alternativ dazu in einem Telnet-Client (siehe unten).

Als Terminalprogramm eignen sich z.B. das im Lieferumfang von Windows bereits enthaltenen "*HyperTerminal*" oder für gehobeneren Ansprüche das als Open-Source verfügbare "*TeraTerm*" (Download unter: <http://tssh2.sourceforge.jp>).

**Telnet-Client** Ein Telnet-Client ermöglicht die Kommunikation mit der **Kommando-Shell** des PLCcore-iMX35 über eine **Ethernet-Verbindung an ETH0 des PLCcore-iMX35**. Voraussetzung für die Verwendung eines Telnet-Clients ist eine abgeschlossene Ethernet-Konfiguration des PLCcore-iMX35 gemäß Abschnitt 7.3. Alternativ zum Telnet-Client können auch sämtliche Kommandos über ein Terminalprogramm (an COM0 des PLCcore-iMX35) eingegeben werden.

Als Telnet-Client eignet sich z.B. das im Lieferumfang von Windows bereits enthaltene "*Telnet*" oder ebenfalls "*TeraTerm*", das gleichzeitig auch als Terminalprogramm eingesetzt werden kann (siehe oben).

**FTP-Client** Ein FTP-Client ermöglicht den Austausch von Dateien zwischen dem PLCcore-iMX35 (ETH0) und dem PC. Dies erlaubt beispielsweise das **Editieren von Konfigurationsdateien**, indem diese zunächst vom PLCcore-iMX35 auf den PC übertragen, dort mit einem Editor bearbeitet und anschließend wieder zurück auf das PLCcore-iMX35 geschrieben werden. Der Download von Dateien auf das PLCcore-iMX35 ist aber auch zum **Update der SPS-Firmware** erforderlich. (Hinweis: Der Update der *SPS-Firmware* ist nicht identisch mit dem Update des *SPS-Anwenderprogramms*. Das SPS-Programm wird direkt aus der CODESYS-Programmierungsumgebung heraus auf das Modul übertragen, hierzu ist keinerlei Zusatzsoftware erforderlich.)

Als FTP-Client für den PC eignen sich beispielsweise das als Open-Source verfügbare "*WinSCP*" (Download unter: <http://winscp.net>), das lediglich aus einer einzelnen EXE-Datei besteht, die keine Installation erfordert und sofort



gestartet werden kann. Ebenso geeignet sind aber auch die Freeware "Core FTP LE" (Download unter: <http://www.coreftp.com>) oder der bereits im Dateimanager "Total Commander" integrierte FTP-Client.

### TFTP-Server

Der TFTP-Server ist lediglich zum Update des Linux-Images auf dem PLCcore-iMX35 erforderlich. Als TFTP-Server eignet sich die Freeware "TFTPD32" (Download unter: <http://tftpd32.jounin.net>). Das Programm besteht lediglich aus einer einzelnen EXE-Datei, die keine Installation erfordert und sofort gestartet werden kann.

Bei Programmen die über die Ethernet-Schnittstelle kommunizieren, wie beispielsweise FTP-Client oder TFTP-Server, ist darauf zu achten, dass die entsprechenden Rechte in der Windows-Firewall freigegeben sind. In der Regel melden Firewalls, dass ein Programm Zugriff auf das Netzwerk erlangen möchte und fragen, ob dieser Zugriff erlaubt oder abgeblockt werden soll. Hier ist in jedem Fall der entsprechende Zugriff zu gestatten.

## 7.2 Linux-Autostart aktivieren bzw. deaktivieren

Im Standardbetriebsmodus startet der Bootloader "U-Boot" bei Reset (bzw. Power-on) autark das Linux-Betriebssystem des Moduls, das dann wiederum das Laden aller weiteren Softwarekomponenten bis hin zur Ausführung des SPS-Programms übernimmt (siehe Abschnitt 6.2). Für Servicezwecke wie beispielsweise die Konfiguration der Ethernet-Schnittstelle (siehe Abschnitt 7.3) oder zum Update des Linux-Images (siehe Abschnitt 7.12.2) ist es erforderlich, diesen Autostart-Mechanismus zu unterbinden und stattdessen zum "U-Boot" Kommandoprompt zu wechseln (Konfigurationsmodus).

Der automatische Start des Linux-Betriebssystems ist an die **gleichzeitige Erfüllung** verschiedener Bedingungen geknüpft ("UND-Verknüpfung"). Dementsprechend ist es zur Unterdrückung autarken Linux-Starts ausreichend, eine dieser Bedingungen **nicht zu erfüllen**.

Im Detail werden durch den Bootloader "U-Boot" die in Tabelle 7 aufgelisteten Voraussetzungen geprüft, von denen alle Bedingungen für ein autarkes Booten des Linux-Images erfüllt sein müssen.

Tabelle 7: Voraussetzungen zum Booten von Linux

Nr.	Bedingung	Bemerkung
1	Anschluss "/BOOT" = High (Taster S602 am Development-board nicht gedrückt)	Der Linux-Autostart wird nur freigegeben, wenn Signal "/BOOT" am PLCcore-iMX35 auf dem H-Pegel ("/BOOT" ist nicht aktiv) liegt.  Die die Position des Anschlusses "/BOOT" am Steckverbinder des Moduls ist im Hardware Manual ECUcore-iMX35 (Manual-Nr.: L-1570) definiert.
2	Kein Abbruch der Autoboot-Prozedur über COM0 des PLCcore-iMX35	Sind die vorangegangenen Bedingungen erfüllt, überprüft "U-Boot" nach Reset für ca. 1 Sekunde die serielle Schnittstelle COM0 des PLCcore-iMX35 auf den Empfang eines SPACE-Zeichens (ASCII 20H). Wird innerhalb dieser Zeit ein entsprechendes Zeichen empfangen, unterbindet "U-Boot" den Linux-Bootvorgang und aktiviert stattdessen seinen eigenen Kommandoprompt.



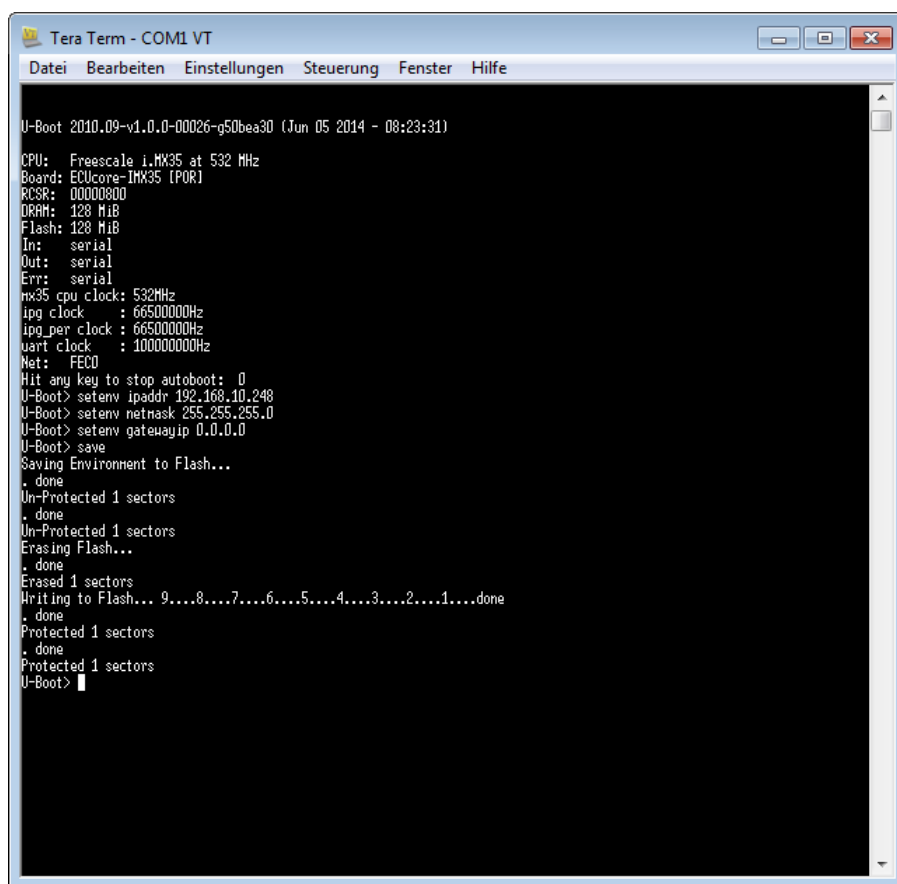
Tabelle 8: "U-Boot" Kommandos zur Konfiguration des PLCcore-iMX35

Einstellung	Kommando	Bemerkung
MAC-Adresse	setenv ethaddr <xx:xx:xx:xx:xx:xx>	Die MAC-Adresse ist eine weltweit eindeutige Kennung des Moduls, die bereits vom Hersteller vergeben wird. <b>Diese sollte vom Anwender normalerweise nicht verändert werden.</b>
IP-Adresse	setenv ipaddr <xxx.xxx.xxx.xxx>	Dieses Kommando setzt die lokale IP-Adresse des PLCcore-iMX35. Die IP-Adresse ist vom Netzwerkadministrator festzulegen.
Netzwerkmaske	setenv netmask <xxx.xxx.xxx.xxx>	Dieses Kommando setzt die Netzwerkmaske des PLCcore-iMX35. Die Netzwerkmaske ist vom Netzwerkadministrator festzulegen.
Gateway-Adresse	setenv gatewayip <xxx.xxx.xxx.xxx>	Dieses Kommando definiert die IP-Adresse des vom PLCcore-iMX35 zu benutzenden Gateways. Die Gateway-Adresse ist vom Netzwerkadministrator festzulegen.  <b>Hinweis:</b> Befinden sich PLCcore-iMX35 und Programmier-PC im selben Sub-Netz, dann kann die Definition der Gateway-Adresse entfallen und stattdessen der Wert "0.0.0.0" verwendet werden.
Speichern der Konfiguration	saveenv	Dieses Kommando speichert die aktuellen Einstellungen im Flash des PLCcore-iMX35.

Die modifizierten Einstellungen können durch die Eingabe von *"printenv"* am "U-Boot" Kommandoprompt nochmals überprüft werden. Die aktuellen Einstellungen werden durch das Kommando

**saveenv**

persistent im Flash des PLCcore-iMX35 gespeichert. Die Änderungen werden mit dem nächsten Reset des PLCcore-iMX35 übernommen.



```
Tera Term - COM1 VT
Datei Bearbeiten Einstellungen Steuerung Fenster Hilfe

U-Boot 2010.09-v1.0.0-00026-g50bea30 (Jun 05 2014 - 08:23:31)

CPU: Freescale i.MX35 at 532 MHz
Board: EOLcore-iMX35 (P0R)
RCMR: 00000800
DRAM: 128 MiB
Flash: 128 MiB
In: serial
Out: serial
Err: serial
mx35 cpu clock: 532MHz
ipg clock : 665000000Hz
ipg_per clock : 665000000Hz
uart clock : 100000000Hz
Net: FEC0
Hit any key to stop autoboot: 0
U-Boot> setenv ipaddr 192.168.10.248
U-Boot> setenv netmask 255.255.255.0
U-Boot> setenv gatewayip 0.0.0.0
U-Boot> save
Saving Environment to Flash...
. done
Un-Protected 1 sectors
. done
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash... 9...8...7...6...5...4...3...2...1...done
. done
Protected 1 sectors
. done
Protected 1 sectors
U-Boot> █
```

Bild 13: Ethernet-Konfiguration des PLCcore-iMX35

**Nach Abschluss der Konfiguration sind gemäß Abschnitt 7.2 die Voraussetzungen für einen Linux-Autostart wieder herzustellen.**

Nach Reset (z.B. Taster S601 am Developmentboard) startet das Modul mit den aktuellen Einstellungen.

**Hinweis:** Nach Abschluss der Konfiguration ist die serielle Verbindung zwischen PC und PLCcore-iMX35 nicht mehr erforderlich.

## 7.4 Boot-Konfiguration des PLCcore-iMX35

Das PLCcore-iMX35 ist so konfiguriert, dass nach einem Reset die SPS-Firmware automatisch gestartet wird. Die dazu notwendigen Kommandos sind in dem Startskript *"/home/etc/autostart"* hinterlegt. Hier werden u.a. die notwendigen Umgebungsvariablen gesetzt sowie die erforderlichen Treiber geladen.

Bei Bedarf kann das Startskript *"/home/etc/autostart"* um weitere Einträge ergänzt werden. Hier kann z.B. durch einfügen des Kommandos *"pureftp"* der Aufruf des FTP-Servers beim Booten des PLCcore-iMX35 automatisiert werden. Das Skript lässt sich im FTP-Client *"WinSCP"* (siehe Abschnitt 7.1) mit Hilfe der Taste *"F4"* bzw. der Schaltfläche *"F4 Edit"* direkt auf dem PLCcore-iMX35 bearbeiten.

## 7.5 Vordefinierte Nutzerkonten

Bei der Auslieferung des PLCcore-iMX35 sind die in Tabelle 9 aufgelisteten Benutzerkonten vordefiniert. Diese erlauben eine Anmeldung an der Kommando-Shell (serielle RS232-Verbindung oder Telnet) und am FTP-Server des PLCcore-iMX35.

Tabelle 9: Vordefinierte Benutzerkonten des PLCcore-iMX35

Benutzername	Passwort	Bemerkung
PlcAdmin	Plc123	vordefiniertes Benutzerkonto zur Administration des PLCcore-iMX35 (Konfiguration, Nutzerverwaltung, Softwareupdates usw.)
root	Sys123	Haupt-Benutzerkonto ("root") des PLCcore-iMX35

## 7.6 Anmeldung am PLCcore-iMX35

### 7.6.1 Anmeldung an der Kommando-Shell

Die Administration des PLCcore-iMX35 erfordert in einigen Fällen die Eingabe von Linux-Kommandos in der Kommando-Shell. Dazu ist eine direkte Anmeldung am Modul notwendig. Dies kann auf zwei alternativen Wegen erfolgen:

- Mit Hilfe eines **Terminalprogramms** (z.B. HyperTerminal oder TeraTerm, siehe Abschnitt 7.1) über die serielle Schnittstelle **COM0** des PLCcore-iMX35, analog zum Vorgehen bei der im Abschnitt 7.2 beschriebenen Ethernet-Konfiguration. **Bei der Konfiguration der Terminaleinstellungen ist darauf zu achten, dass als Zeilenendezeichen nur "CR" (carriage return) verwendet wird.** Bei "CR+LF" (carriage return + line feed) ist keine Anmeldung mit Nutzernamen und Passwort möglich!
- Alternativ ist die Anmeldung mit Hilfe eines **Telnet-Clients** (z.B. Telnet oder ebenfalls TeraTerm) über die Ethernet-Schnittstelle **ETH0** des PLCcore-iMX35 möglich.

Um sich über den in Windows standardmäßig enthaltenen Telnet-Client am PLCcore-iMX35 anzumelden, ist der Befehl `telnet` unter Angabe der in Abschnitt 7.2 festgelegten IP-Adresse für das PLCcore-iMX35 aufzurufen, z.B.

```
telnet 192.168.10.248
```

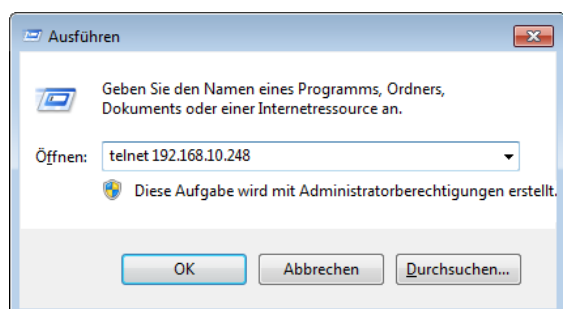


Bild 14: Aufruf des Telnet-Clients unter Windows

Innerhalb des Terminal-Fensters (bei Verbindung über COM0) bzw. des Telnet-Fensters (bei Verwendung von ETH0) ist die Anmeldung am PLCcore-iMX35 möglich. Bei Auslieferung des Moduls ist zur Administration des PLCcore-iMX35 folgendes Nutzerkonto vorkonfiguriert (siehe auch Abschnitt 7.5):

User: *PlcAdmin*  
Passwort: *Plc123*

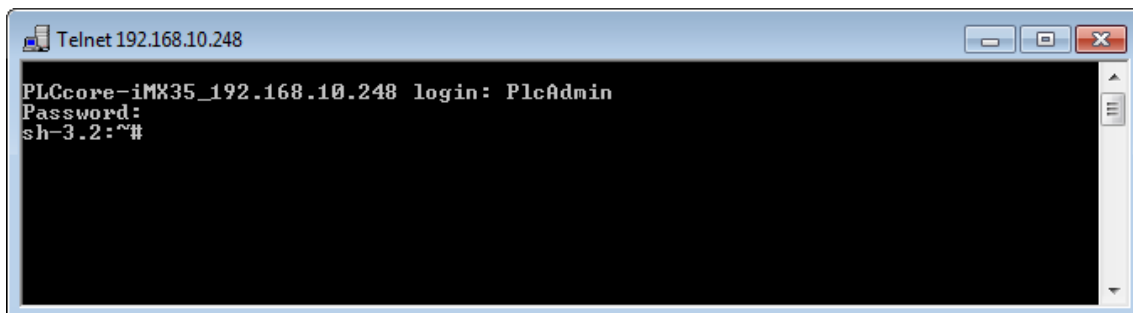


Bild 15: Anmeldung am PLCcore-iMX35

Bild 15 verdeutlicht am Beispiel die Anmeldung am PLCcore-iMX35 mit Hilfe des in Windows standardmäßig enthaltenen Telnet-Clients.

## 7.6.2 Anmeldung am FTP-Server

Das PLCcore-iMX35 verfügt über einen FTP-Server (FTP Daemon), der den Austausch von Dateien mit einem PC ermöglicht (Up- und Download von Dateien). Aus Sicherheits- und Performance-Gründen ist dieser FTP-Server jedoch standardmäßig deaktiviert und muss bei Bedarf zunächst manuell gestartet werden. Hierzu ist zunächst die im Abschnitt 7.6.1 beschriebenen Anmeldung an der Kommando-Shell des PLCcore-iMX35 durchzuführen. Anschließend ist im Telnet- bzw. Terminal-Fenster folgendes Kommando einzugeben:

```
pureftp
```

Bild 16 verdeutlicht am Beispiel das Starten des FTP-Servers.

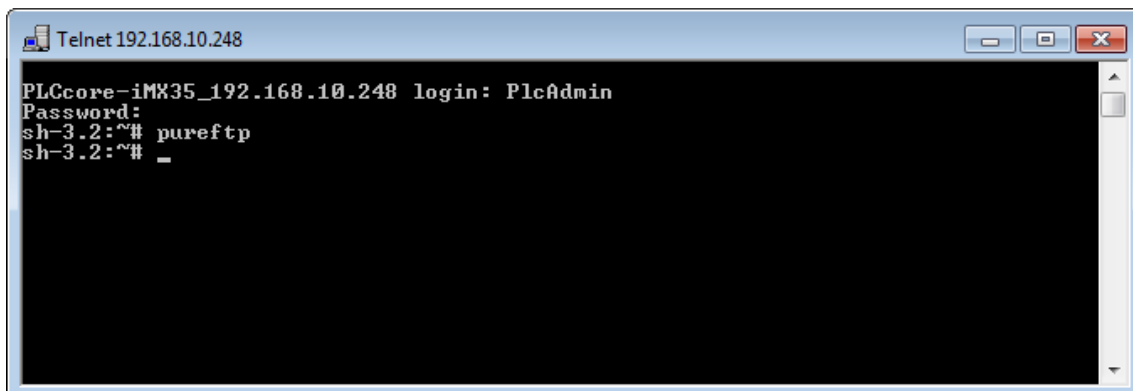


Bild 16: Starten des FTP-Servers

**Hinweis:** Durch Einfügen des Kommandos *"pureftp"* in das Startskript *"/home/etc/autostart"* lässt sich der Aufruf des FTP-Servers beim Booten des PLCcore-iMX35 automatisieren (siehe dazu Abschnitt 7.4).

Als FTP-Client für den PC eignet sich beispielsweise das als Open-Source verfügbare *"WinSCP"* (siehe Abschnitt 7.1), das lediglich aus einer einzelnen EXE-Datei besteht, die keine Installation erfordert und sofort gestartet werden kann. Nach dem Programmstart erscheint zunächst der Dialog *"WinSCP Login"* (siehe Bild 17), in dem folgende Einstellungen vorzunehmen sind:

File protocol: FTP  
Host name: die im Abschnitt 7.3 festgelegte IP-Adresse für das PLCcore-iMX35  
User name: *PlcAdmin* (für vorkonfiguriertes Nutzerkonto, siehe Abschnitt 7.5)  
Password: *Plc123* (für vorkonfiguriertes Nutzerkonto, siehe Abschnitt 7.5)

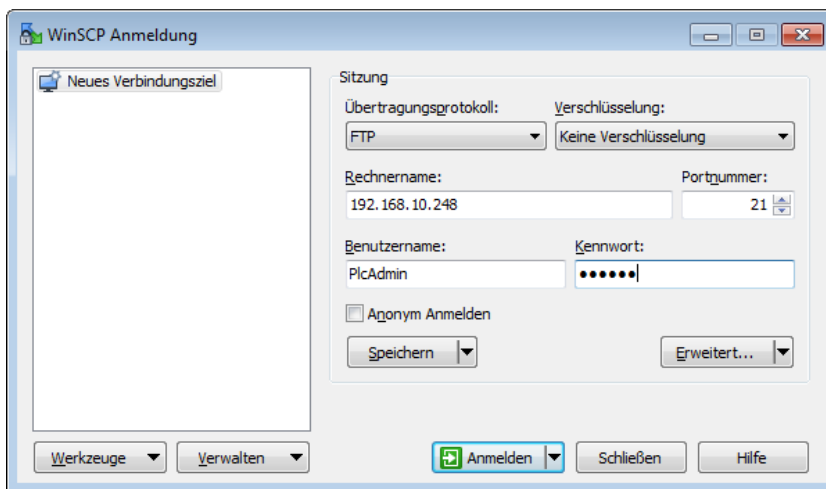


Bild 17: Login-Einstellungen für WinSCP

Nach dem Betätigen der Schaltfläche *"Anmelden"* meldet sich der FTP-Client am PLCcore-iMX35 an und listet im rechten Fenster den aktuellen Inhalt des Verzeichnisses *"/home"* auf. Bild 18 zeigt den FTP-Client *"WinSCP"* nach der erfolgreichen Anmeldung am PLCcore-iMX35.

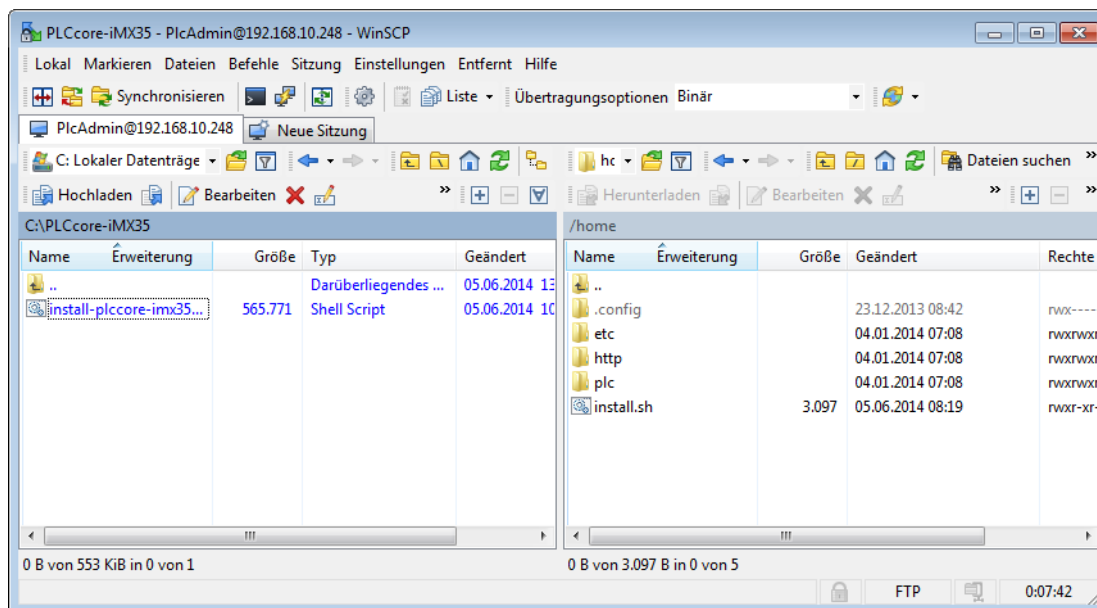


Bild 18: FTP-Client für Windows "WinSCP"

Nach einer erfolgreichen Anmeldung lassen sich innerhalb des FTP-Clients "WinSCP" mit Hilfe der Taste "F4" bzw. der Schaltfläche "F4 Edit" Konfigurationsdateien auf dem PLCcore-iMX35 bearbeiten (dazu Transfermodus "Text" selektieren). Mit Hilfe der Taste "F5" bzw. der Schaltfläche "F5 Copy" können Dateien zwischen dem PC und dem PLCcore-iMX35 transferieren werden, um beispielsweise Datensicherungen des PLCcore-iMX35 durchzuführen oder um Installationsdateien für Firmware-Updates auf das Modul zu übertragen (dazu Transfermodus "Binary" selektieren).

## 7.7 Anlegen und Löschen von Nutzerkonten

Das Anlegen und Löschen von Nutzerkonten erfordert zunächst die Anmeldung am PLCcore-iMX35 wie in Abschnitt 7.6.1 beschrieben.

Das **Anlegen** eines neuen Nutzerkontos erfolgt mit Hilfe des Linux-Kommandos "adduser". Da es bei einem Embedded System wie dem PLCcore-iMX35 nicht sinnvoll ist, für jeden Benutzer ein eigenes Verzeichnis anzulegen, ist mit dem Parameter "-H" das Erstellen eines neuen Verzeichnisses zu unterbinden. Stattdessen wird dem neuen Benutzer über den Parameter "-h /home" das bereits vorhandene Verzeichnis "/home" zugeordnet. Zum Anlegen eines neuen Nutzerkontos auf dem PLCcore-iMX35 ist das Linux-Kommandos "adduser" wie folgt anzuwenden:

```
adduser -h /home -H -G <group> <username>
```

Bild 19 verdeutlicht am Beispiel das Anlegen eines neuen Kontos auf dem PLCcore-iMX35 für den Nutzer "admin2".



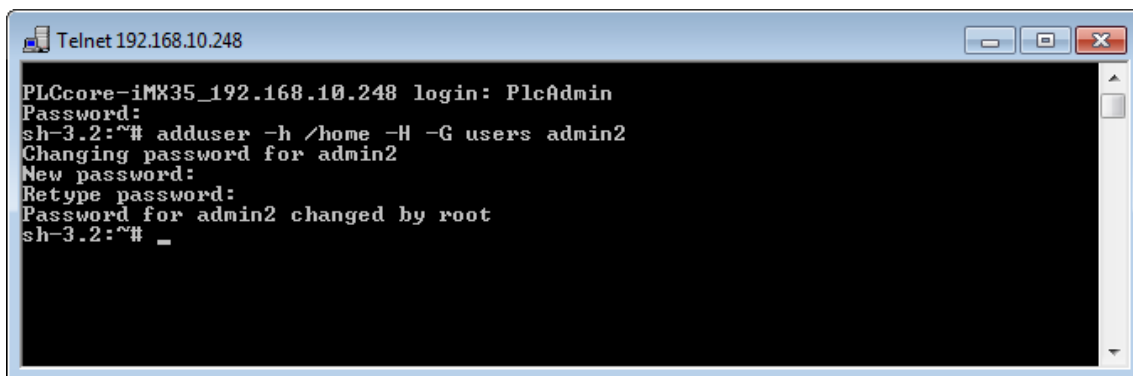


Bild 19: Anlegen eines neuen Nutzerkontos

Zum **Löschen** eines vorhandenen Nutzerkontos vom PLCcore-iMX35 ist das Linux-Kommando "*deluser*" unter Angabe des Benutzernamens zu verwenden:

```
deluser <username>
```

## 7.8 Passwort eines Nutzerkontos ändern

Das Ändern von Passwörtern erfordert zunächst die Anmeldung am PLCcore-iMX35 wie in Abschnitt 7.6.1 beschrieben.

Zum Ändern des Passwortes für ein auf dem PLCcore-iMX35 vorhandenes Nutzerkonto ist das Linux-Kommando "*passwd*" unter Angabe des Benutzernamens zu verwenden:

```
passwd <username>
```

Bild 20 verdeutlicht am Beispiel das Ändern des Passwortes für den Nutzer "*PlcAdmin*".

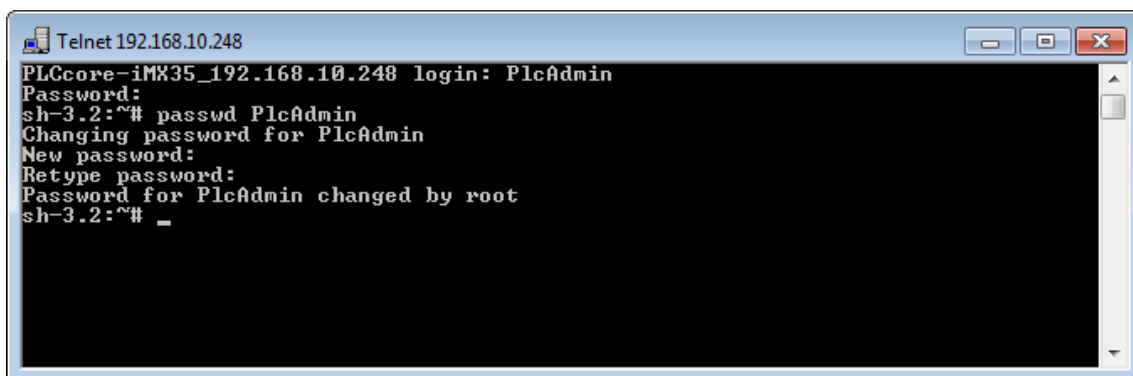


Bild 20: Passwort eines Nutzerkontos ändern

## 7.9 Setzen der Systemzeit

Das Setzen der Systemzeit erfordert zunächst die Anmeldung am PLCcore-iMX35 wie in Abschnitt 7.6.1 beschrieben.

Das Setzen der Systemzeit für das PLCcore-iMX35 erfolgt in zwei Schritten. Zuerst sind Datum und Uhrzeit mit Hilfe des Linux-Kommandos *"date"* auf die aktuellen Werte zu setzen. Anschließend wird die Systemzeit durch das Linux-Kommando *"hwclock -w"* in den RTC-Baustein des PLCcore-iMX35 übernommen.

Das Linux-Kommando *"date"* besitzt folgenden Aufbau:

```
date [options] [YYYY.]MM.DD-hh:mm[:ss]
```

### Beispiel:

```
date 2014.06.05-14:00:35
|   |   |   |   |   |
|   |   |   |   | +--- Sekunde
|   |   |   | +----- Minute
|   |   +----- Stunde
|   +----- Tag
| +----- Monat
+----- Jahr
```

Um die aktuelle Systemzeit des PLCcore-iMX35 wie im obigen Beispiel dargestellt auf den 2014/06/05 um 14:00:35 zu setzen, ist folgende Befehlssequenz notwendig:

```
date 2014.06.05-14:00:35
hwclock -w
```

Die aktuelle Systemzeit wird durch Eingabe des Linux-Kommandos *"date"* (ohne Parameter) angezeigt, die aktuellen Werte der RTC lassen sich durch das Linux-Kommando *"hwclock -r"* abfragen. Mit *"hwclock -s"* werden die aktuellen Werte der RTC als Systemzeit für Linux übernommen (Synchronisation des Kernels auf die RTC). Bild 21 verdeutlicht das Setzen und Anzeigen der Systemzeit am Beispiel.

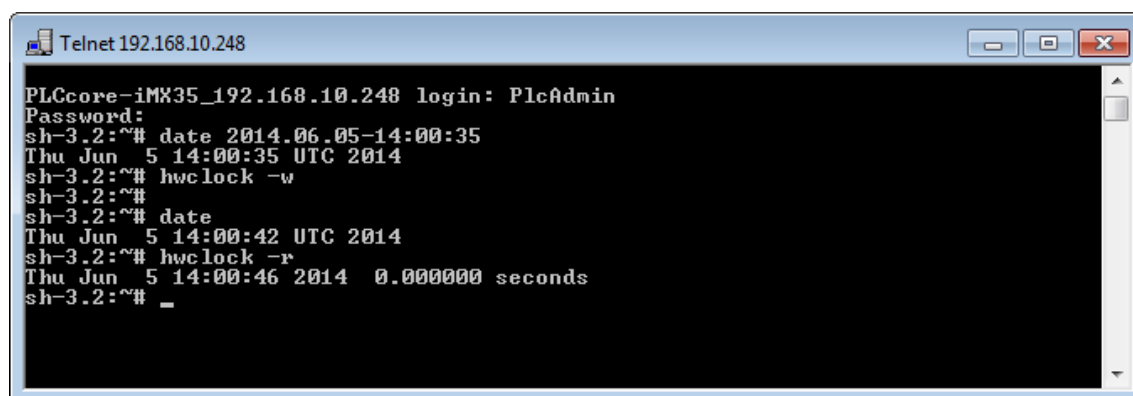


Bild 21: Setzen und Anzeigen der Systemzeit

Beim Starten des PLCcore-iMX35 werden Datum und Uhrzeit der RTC als aktuelle Systemzeit für das Modul übernommen. Das dazu notwendige Linux-Kommando *"hwclock -s"* ist bereits im Startskript *"/etc/init.d/hwclock"* enthalten.

## 7.10 Dateisystem des PLCcore-iMX35

Das auf dem PLCcore-iMX35 vorinstallierte Embedded Linux stellt Teile des Systemspeichers in Form eines Dateisystems zur Verfügung. Wie bei Embedded Systemen üblich ist dabei der überwiegende Teil des Dateisystems read/only, das bedeutet, dass Änderungen an diesem Teil nur durch Neuerstellung des Linux-Images für das PLCcore-iMX35 vorgenommen werden können. Der große Vorteil besteht hier in der Resistenz des read/only Dateisystems gegen Beschädigungen beim Ausfall der Spannungsversorgung. Diese treten bei Embedded Systemen relativ häufig auf, da Embedded Systeme in der Regel einfach nur ausgeschaltet werden, ohne vorherigen Shutdown.

Die zur Laufzeit beschreibbaren Zweige des Dateisystems listet Tabelle 10 auf. Hinter dem Verzeichnis **"/home"** verbirgt sich eine Flash-Disk, die einen Teil des on-board Flash-Speichers des PLCcore-iMX35 als Dateisystem zur Verfügung stellt. In diesem Pfad werden sämtliche vom Anwender modifizierbaren und updatefähigen Files wie beispielsweise Konfigurationsdateien, SPS-Firmware sowie die auf das Modul geladenen SPS-Programm-Dateien abgelegt. Das Verzeichnis **"/tmp"** ist in seiner Größe so dimensioniert, dass es als Zwischenpuffer für den FTP-Download des Firmware-Archives beim Update der SPS-Firmware benutzt werden kann (siehe Abschnitt 7.12.1).

Tabelle 10: Dateisystemkonfiguration des PLCcore-iMX35

Zweig	Nutzbare Größe	Beschreibung
/home	95516 kByte	Flash-Disk, zur persistenten Ablage vom Anwender modifizierbarer und updatefähiger Files (z.B. Konfigurationsdateien, SPS-Firmware, SPS-Programm, Dateien für Target-Visualisierung), Datenerhalt bei Spannungsausfall ist gegeben
/tmp	8192 kByte	RAM-Disk, geeignet als Zwischenpuffer für FTP-Downloads, aber kein Datenerhalt bei Spannungsausfall
/var/log	62640 kByte	RAM-Disk, wird vom System selbst zur Ablage temporärer Dateien benutzt, aber kein Datenerhalt bei Spannungsausfall
/mnt		Ziel für die Einbindung von Remote-Verzeichnissen, wird auf dem PLCcore-iMX35 in der Standardfunktionalität nicht verwendet

Die konfigurierten sowie jeweils aktuell noch verfügbaren Größen der einzelnen Dateisystemzweige können mit Hilfe des Linux-Kommandos "df" ("DiskFree") ermittelt werden (siehe Bild 22).

```

Telnet 192.168.10.248
PLCcore-iMX35_192.168.10.248 login: PlcAdmin
Password:
sh-3.2:~# df
Filesystem            1k-blocks    Used Available Use% Mounted on
rootfs                 16640        16640      0 100% /
/dev/root              16640        16640      0 100% /
devtmpfs               62584         0    62584   0% /dev
/dev/ubi0_0            95516        2256    88424   2% /home
none                   8192          8     8184   0% /tmp
none                   62640         12    62628   0% /run
none                   62640         12    62628   0% /var/log
none                   62640         12    62628   0% /var/run
none                   62640         0    62640   0% /var/lock
none                   62640         0    62640   0% /var/tmp
sh-3.2:~# _

```

Bild 22: Anzeige von Informationen zum Dateisystem

Einzelheiten zur Anmeldung am System und zum Umgang mit der Linux-Kommando-Shell des PLCcore-iMX35 behandelt Abschnitt 7.6.

## 7.11 Kalibrierung des Touchscreen

Das PLCcore-iMX35 besitzt keinen eigenen on-board Touch-Controller. Für den Anschluss resistiver Touchscreens wird dementsprechend ein externer Touch Controller benötigt. Touchscreen und Touch Controller müssen vor der ersten Verwendung aufeinander abgestimmt – kalibriert – werden. Ohne Kalibrierung arbeitet der Touchscreen extrem ungenau, normalerweise ist eine korrekte Bedienung unmöglich.

### 7.11.1 Automatische Überprüfung der Touchscreen-Kalibrierung

Eine erfolgreiche Kalibrierung ist Voraussetzung für die Benutzung des Touchscreens. Die Geräte-Firmware kann beim Booten des SPS Systems prüfen, ob die erforderliche Kalibrierung des Touchscreens bereits durchgeführt wurde. Als Entscheidungskriterium hierfür wird untersucht, ob die Datei `"/home/etc/pointercal"` existiert und größer als 0 Byte ist. Ist diese Bedingung nicht erfüllt, wird noch vor dem Start der SPS-Firmware das entsprechende Kalibrierprogramm `"ts_calibrate"` aufgerufen (siehe auch Abschnitt 7.11.2).

Da das PLCcore-iMX35 sowohl Displays mit als auch ohne Touchscreen unterstützt, lässt sich die automatische Überprüfung der Touchscreen-Kalibrierung durch Konfigurationseinstellungen des Modules wahlweise aktivieren bzw. deaktivieren. Die entsprechende Konfiguration erfolgt über die Umgebungsvariable `"check_tscalibfile"` des Bootloaders "U-Boot". Um diese Variable zu setzen, ist zunächst wie im Abschnitt 7.2 beschrieben der "U-Boot"-Kommandoprompt zu aktivieren. Tabelle 11 listet die Kommandos zum ein- bzw. ausschalten der automatische Überprüfung auf.

Tabelle 11: Konfiguration der automatischen Überprüfung der Touchscreen-Kalibrierung

Kommandosequenz	Einstellung
setenv check_tscalibfile on saveenv	Automatische Überprüfung der Touchscreen-Kalibrierung aktiviert, falls die Datei <code>"/home/etc/pointercal"</code> nicht existiert (oder nur eine Größe von 0 Byte hat), wird automatisch das Kalibrierprogramm <code>"ts_calibrate"</code> aufgerufen
setenv check_tscalibfile off saveenv	Automatische Überprüfung der Touchscreen-Kalibrierung deaktiviert, das Vorhandensein der Datei <code>"/home/etc/pointercal"</code> wird nicht geprüft

**Hinweis:** Das in Tabelle 11 jeweils mit aufgeführte Kommando `"saveenv"` ist erforderlich, um die modifizierte Konfiguration persistent im Flash des PLCcore-iMX35 abzuspeichern.

### 7.11.2 Manuelle Kalibrierung des Touchscreen

Das manuelle Kalibrieren des Touchscreens erfolgt interaktiv, indem der Anwender auf dem Display vorgegebene Markierungen ("Fadenkreuze") anklickt. Das dazu notwendige Kalibrierungs-Programm wird über die Kommandozeile gestartet, daher ist zunächst wie in Abschnitt 7.6.1 beschrieben die Anmeldung an der Kommando-Shell des PLCcore-iMX35 notwendig. Anschließend ist im Telnet- bzw. Terminal-Fenster folgendes Kommando einzugeben:

```
ts_calibrate
```

Im Verlaufe der Kalibrierungssequenz werden auf dem Display nacheinander 5 Markierungen ("Fadenkreuze" in allen 4 Ecken sowie in der Mitte) angezeigt, die vom Anwender anzuklicken sind. Je exakter dabei die dargestellten Markierungen angeklickt werden, umso größer ist die erreichbare Genauigkeit bei der späteren Bedienung des Touchscreen. Es wird daher empfohlen, zur Kalibrierung einen Eingabestift (umgangssprachlich als "Touchpen" oder "Stylus" bezeichnet) zu verwenden, wie er auch für Handhelds, PDAs oder Grafiktablets üblich ist.

Nach Abschluss der Kalibrierung werden die Kalibrierdaten in der Datei *"/home/etc/pointercal"* gespeichert. Geht diese Datei verloren (z.B. durch Neuformatierung der Flash-Disk, ist die Kalibrierung erneut auszuführen.

**Hinweis:** Das Development Kit PLCcore-iMX35 wird bereits fertig kalibriert ausgeliefert. Eine Neukalibrierung ist nur in Ausnahmefällen (z.B. nach Wechsel des Displays mit integriertem Touchscreen) notwendig.

## 7.12 Software-Update des PLCcore-iMX35

Bei Auslieferung ab Werk sind bereits sämtliche für den Betrieb des PLCcore-iMX35 notwendigen Firmwarekomponenten auf dem Modul installiert. Ein Firmwareupdate ist daher nur in Ausnahmefällen erforderlich, um beispielsweise eine aktuellere Software mit neuen Eigenschaften einzuspielen.

### 7.12.1 Update der SPS-Firmware

Als SPS-Firmware wird die Laufzeitumgebung der SPS bezeichnet. Die **SPS-Firmware** kann nur vom Hersteller generiert und modifiziert werden, sie ist nicht identisch mit dem **SPS-Anwenderprogramm**, das vom Nutzer der SPS erstellt wird. Das SPS-Anwenderprogramm wird direkt aus der CODESYS-Programmierungsumgebung heraus auf das Modul übertragen, hierzu ist keinerlei externe Zusatzsoftware erforderlich.

Ein Update der SPS-Firmware erfordert sowohl die Anmeldung an der Kommando-Shell des PLCcore-iMX35 wie in Abschnitt 7.6.1 beschrieben als auch die Anmeldung am FTP-Server gemäß Abschnitt 7.6.2.

Das Update der SPS-Firmware erfolgt durch ein selbst-extrahierendes Firmware-Archiv, das per FTP auf das PLCcore-iMX35 übertragen wird. Nach dem Start des FTP-Servers auf dem PLCcore-iMX35 (Kommando *"pureftp"*, siehe Abschnitt 7.6.2) kann das entsprechende Firmware-Archiv in das Verzeichnis *"/tmp"* des PLCcore-iMX35 übertragen werden (siehe Bild 23).

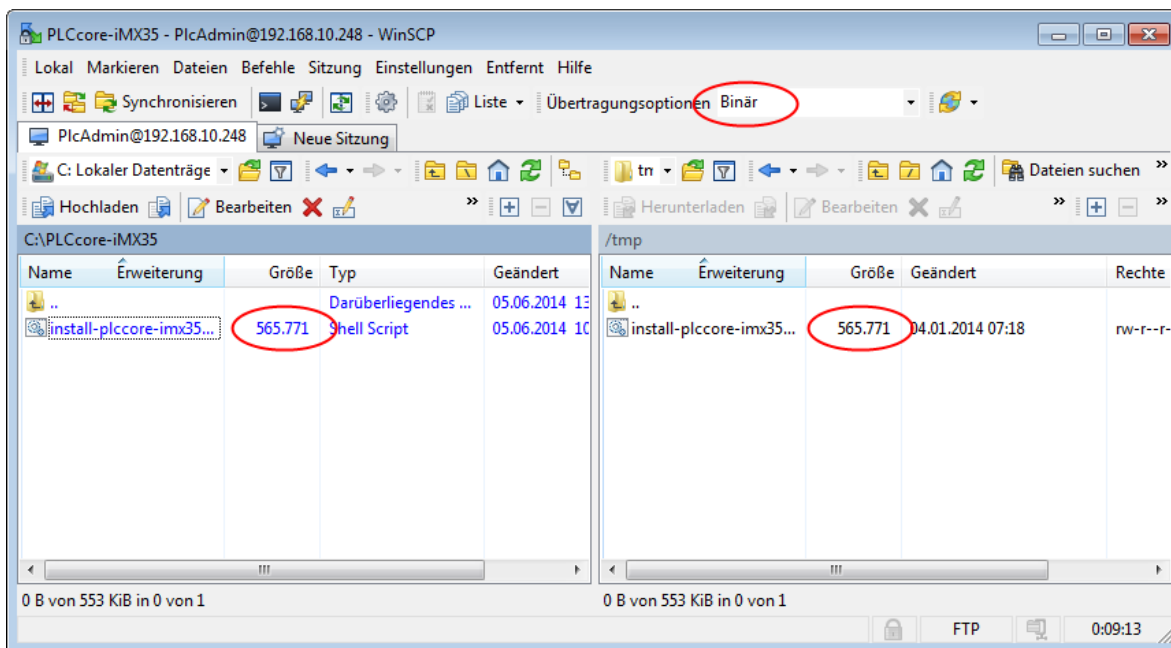


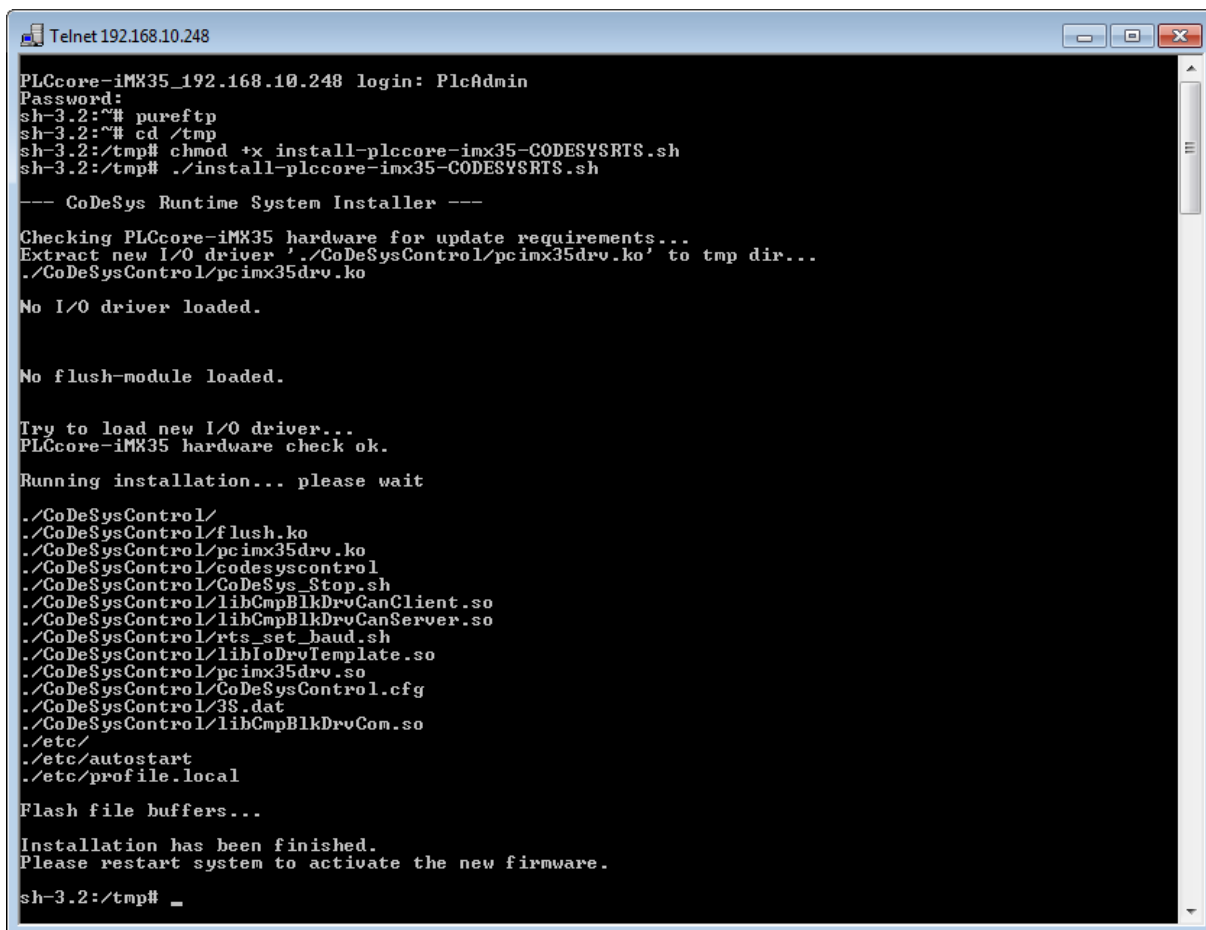
Bild 23: Dateitransfer im FTP-Client "WinSCP"

**Wichtig:** Für den FTP-Transfer des Firmware-Archives ist der Übertragungstyp "Binary" auszuwählen. Bei Verwendung des FTP-Clients "WinSCP" ist hierzu in der Menüleiste der entsprechende Transfermodus zu selektieren. Nach dem Download des Firmware-Archives ist zu kontrollieren, dass die übertragene Datei auf dem PLCcore-iMX35 exakt dieselbe Größe besitzt wie die Originaldatei auf dem PC. Eine abweichende Größe deutet auf einen falschen Übertragungsmodus hin (z.B. "Text"). In diesem Fall ist der Download mit dem Übertragungstyp "Binary" zu wiederholen.

Nach dem Download des selbstextrahierenden Archives muss die SPS-Firmware auf dem PLCcore-iMX35 installiert werden. Dazu sind im Telnet-Fenster nachfolgende Befehle einzugeben:

```
cd /tmp
chmod +x install-plccore-imx35-CODESYSRTS.sh
./install-plccore-imx35-CODESYSRTS.sh
```

**Tipp:** Die Kommando-Shell auf dem PLCcore-iMX35 kann angefangene Namen durch Drücken der Tab-Taste automatisch vervollständigen ("tab completion"), so dass es in der Regel genügt, nur den Anfang des Dateinamens einzugeben und dann vom System automatisch ergänzen zu lassen. So wird z.B. `./ins` durch Drücken der Tab-Taste automatisch zu `./install-plccore-imx35-CODESYSRTS.sh` erweitert.



```
Telnet 192.168.10.248
PLCcore-iMX35_192.168.10.248 login: PlcAdmin
Password:
sh-3.2:~# pureftpd
sh-3.2:~# cd /tmp
sh-3.2:/tmp# chmod +x install-plccore-imx35-CODESYSRTS.sh
sh-3.2:/tmp# ./install-plccore-imx35-CODESYSRTS.sh

--- CoDeSys Runtime System Installer ---

Checking PLCcore-iMX35 hardware for update requirements...
Extract new I/O driver './CoDeSysControl/pcimx35drv.ko' to tmp dir...
./CoDeSysControl/pcimx35drv.ko

No I/O driver loaded.

No flush-module loaded.

Try to load new I/O driver...
PLCcore-iMX35 hardware check ok.

Running installation... please wait

./CoDeSysControl/
./CoDeSysControl/flush.ko
./CoDeSysControl/pcimx35drv.ko
./CoDeSysControl/codesyscontrol
./CoDeSysControl/CoDeSys_Stop.sh
./CoDeSysControl/libCmpBlkDrvCanClient.so
./CoDeSysControl/libCmpBlkDrvCanServer.so
./CoDeSysControl/rts_set_baud.sh
./CoDeSysControl/libIoDrvTemplate.so
./CoDeSysControl/pcimx35drv.so
./CoDeSysControl/CoDeSysControl.cfg
./CoDeSysControl/3S.dat
./CoDeSysControl/libCmpBlkDrvCom.so
./etc/
./etc/autostart
./etc/profile.local

Flash file buffers...

Installation has been finished.
Please restart system to activate the new firmware.
sh-3.2:/tmp# _
```

Bild 24: Installation der SPS-Firmware auf dem PLCcore-iMX35

Bild 24 verdeutlicht die Installation der SPS-Firmware auf dem PLCcore-iMX35. Nach einem Reset startet das Modul anschließend mit der aktualisierten Firmware.

## 7.12.2 Update des Linux-Images

Der Update des Linux-Images erfolgt via TFTP (Trivial FTP) innerhalb des Linux-Bootloaders "U-Boot". Auf dem PC ist hierfür ein entsprechender TFTP-Server notwendig, beispielsweise die Freeware "TFTPD32" (siehe Abschnitt 7.1). Das Programm besteht lediglich aus einer einzelnen EXE-Datei, die keine Installation erfordert und sofort gestartet werden kann. Nach dem Programmstart sollte ein geeignetes Arbeitsverzeichnis ("Current Directory") durch Anklicken der Schaltfläche "Browse" eingestellt werden (z.B. "C:\VPLCcore-iMX35"). In diesem Verzeichnis müssen sich Image-Dateien für das PLCcore-iMX35 befinden ("linuximage" und "root.squashfs").

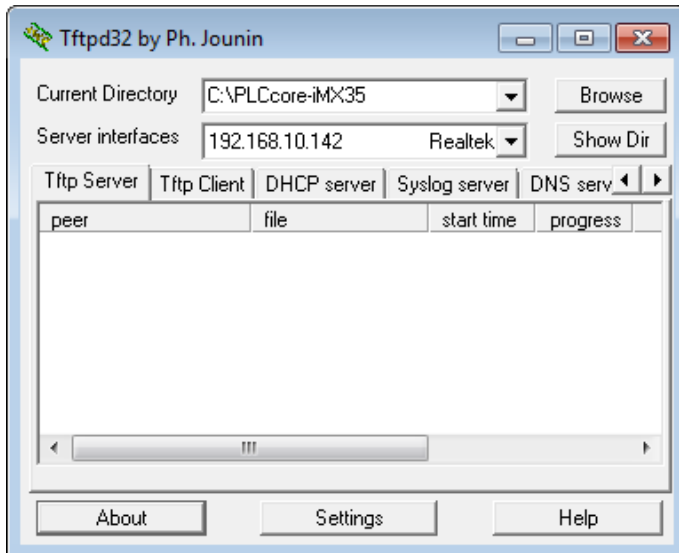


Bild 25: TFTP-Server für Windows "TFTPD32"

**Voraussetzung** für den TFTP-Download der Image-Dateien **ist die abgeschlossene Ethernet-Konfiguration** des PLCcore-iMX35 **gemäß Abschnitt 7.3**. Für das Update des Linux-Images ist neben der Ethernet-Verbindung noch eine serielle Verbindung zum PLCcore-iMX35 erforderlich. Hier gelten ebenfalls die im Abschnitt 7.2 beschriebenen Einstellungen für das Terminalprogramm (115200 Baud, 8 Datenbits, 1 Stopbit, keine Parität, keine Flusskontrolle).

**Ein Update des Linux-Images auf dem PLCcore-iMX35 ist nur möglich, wenn Linux noch nicht gestartet ist. Daher ist vor dem Update der Linux-Autostart zu unterbinden und stattdessen zum "U-Boot" Kommandoprompt zu wechseln. Die dazu notwendigen Schritte beschreibt Abschnitt 7.2.**



Nach Reset (z.B. Taster S601 am Developmentboard) und dem Abbruch des Linux-Autostarts (siehe Abschnitt 7.2) meldet sich der "U-Boot" Kommandoprompt. Hier sind zum Update des Linux-Images die im Folgenden beschriebenen Kommandos in der aufgeführten Reihenfolge einzugeben:

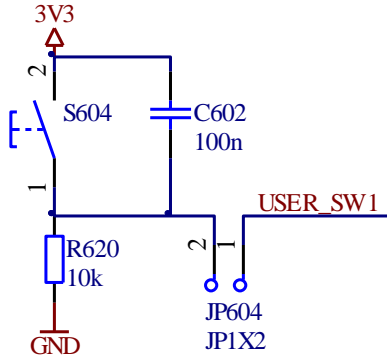
Tabelle 12: Kommando-Sequenz zum Update des Linux-Images auf dem PLCcore-iMX35

Kommando	Bedeutung
<code>setenv serverip &lt;host_ip_addr&gt;</code>	Setzen der IP-Adresse des TFTP-Servers. Bei Verwendung von "TFTP32" wird diese auf dem PC im Feld "Server Interface" angezeigt
<code>ftp linuximage</code>	Download des Linux-Images vom Entwicklungs-PC auf das PLCcore-iMX35
<code>erase nor0,4</code>	Löschen des vom Linux-Image benötigten Flash-Bereiches
<code>cp.b \${fileaddr} 0xa00e0000 \${filesize}</code>	Speichern des Linux-Images im Flash des PLCcore-iMX35
<code>ftp root.squashfs</code>	Download des Root-Filesystems vom Entwicklungs-PC auf das PLCcore-iMX35
<code>erase nor0,5</code>	Löschen des vom Root-Filesystems benötigten Flash-Bereiches
<code>cp.b \${fileaddr} 0xa04e0000 \${filesize}</code>	Speichern des Root-Filesystems im Flash des PLCcore-iMX35

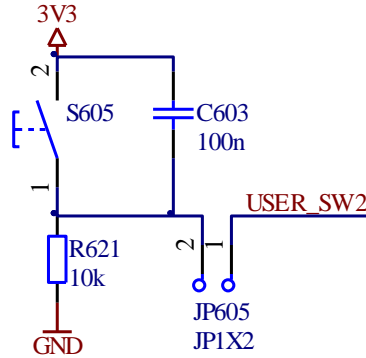


## Anhang A: Referenzdesigns zum PLCcore-iMX35

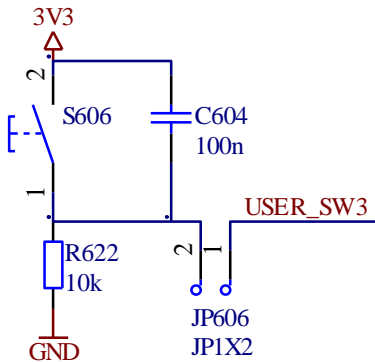
USER-Switch 1



USER-Switch 2



USER-Switch 3



USER-Switch 4

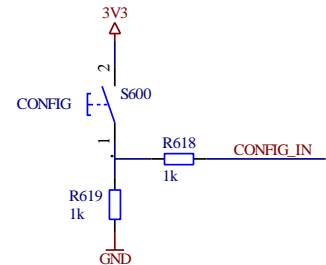
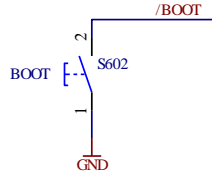
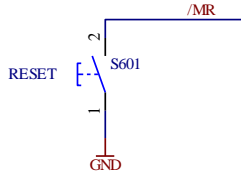
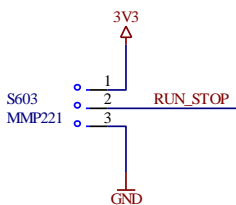
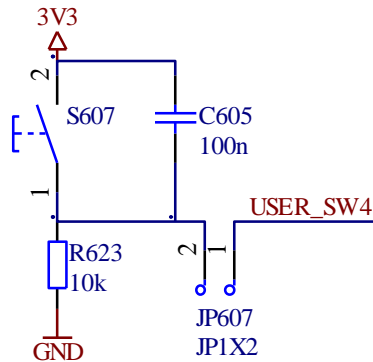


Bild 27: Referenzdesign Bedienelemente

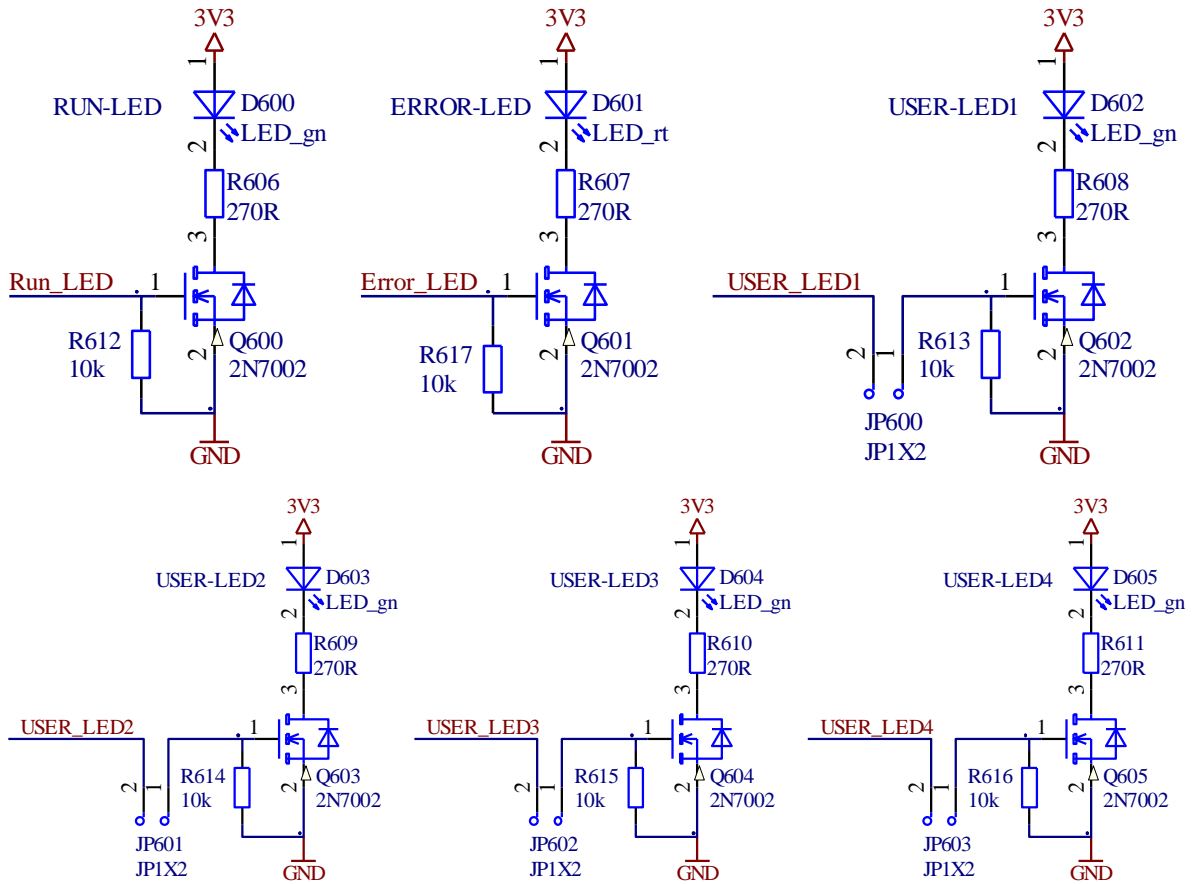
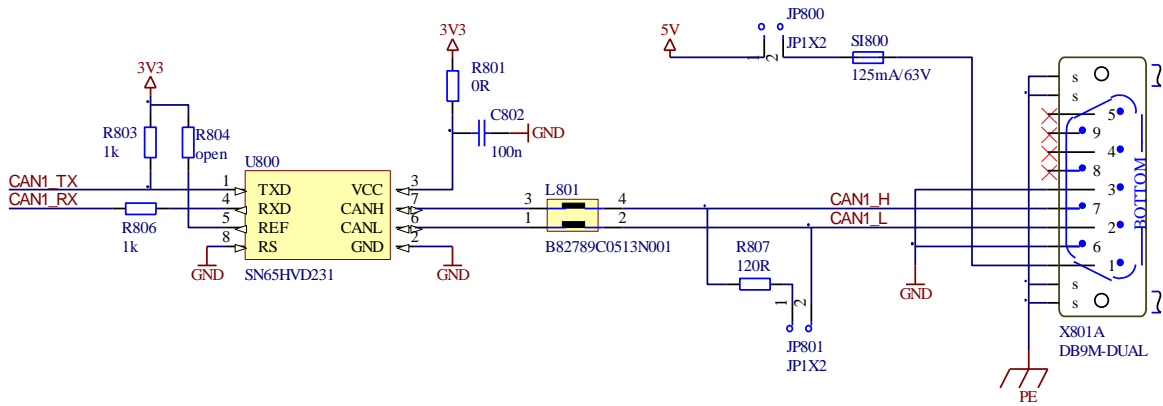


Bild 28: Referenzdesign LEDs

# CAN1



# CAN2

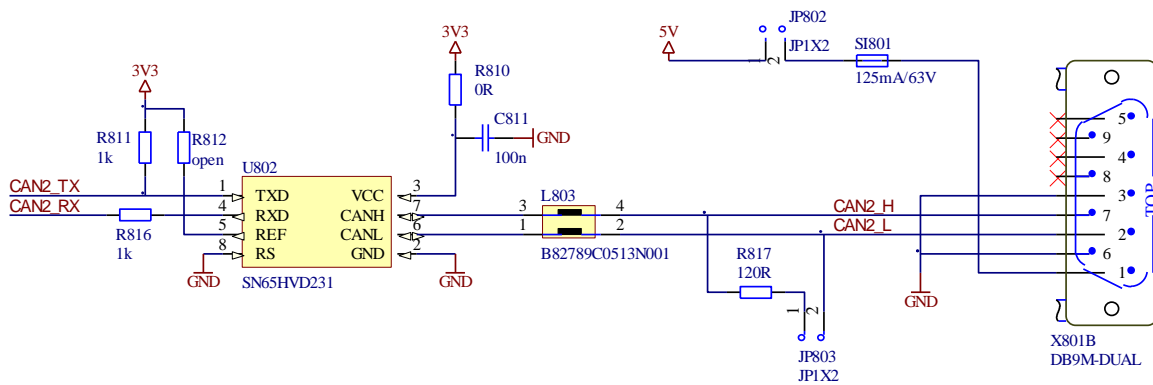
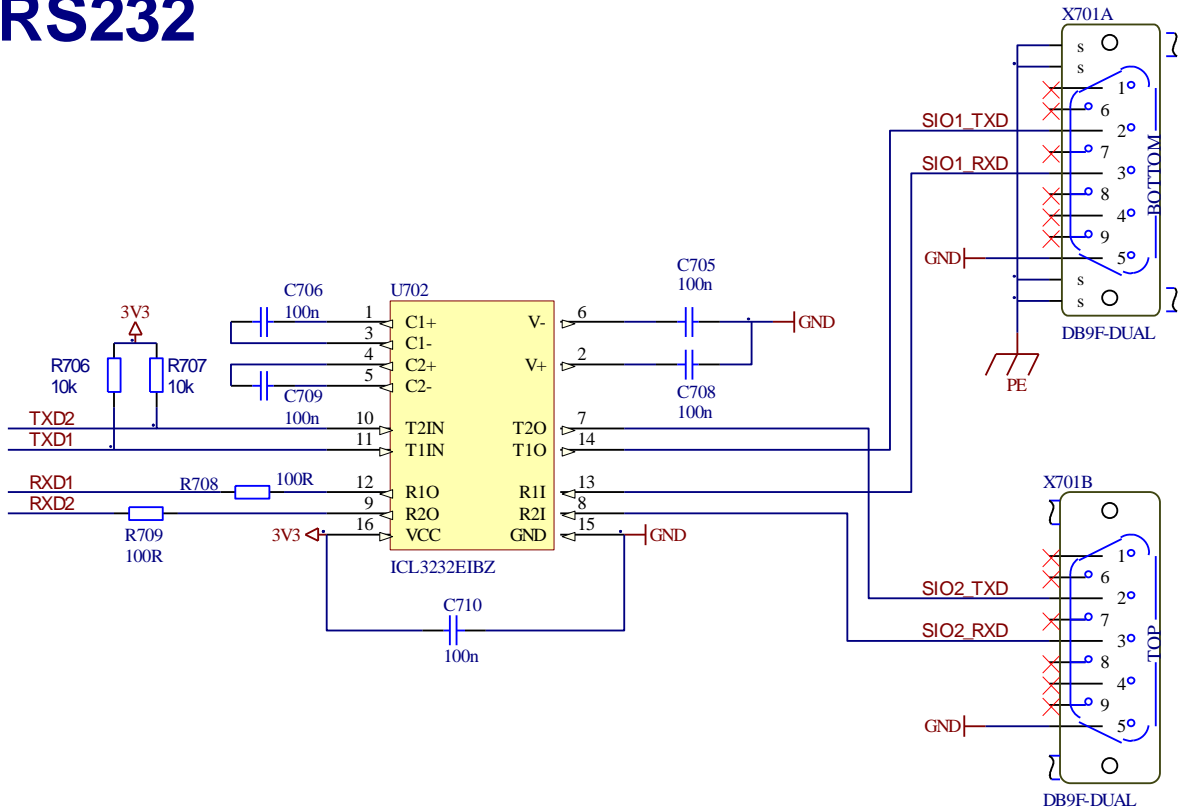


Bild 29: Referenzdesign zur Interface-Beschaltung CAN

# RS232



# RS485

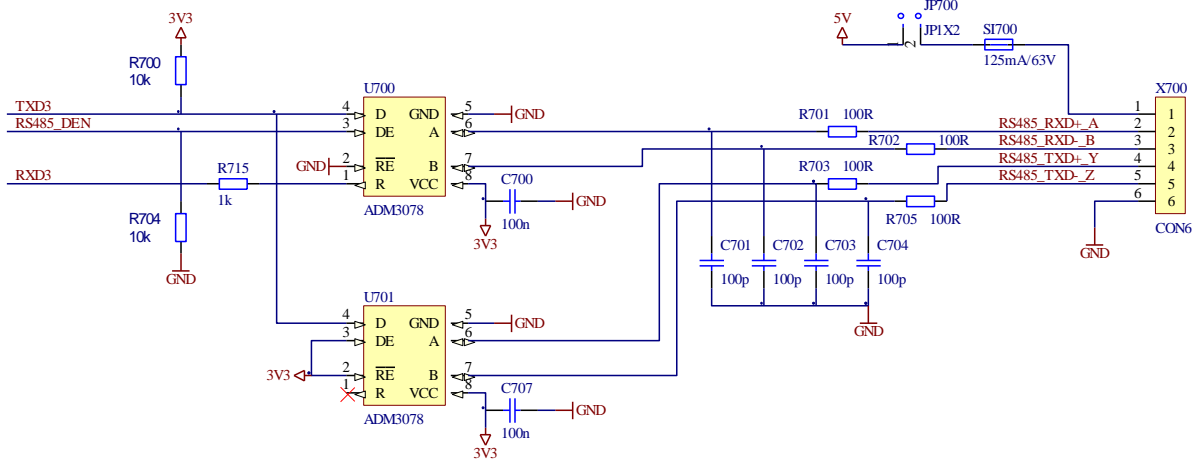


Bild 30: Referenzdesign zur Interface-Beschaltung RS232/485

# LAN

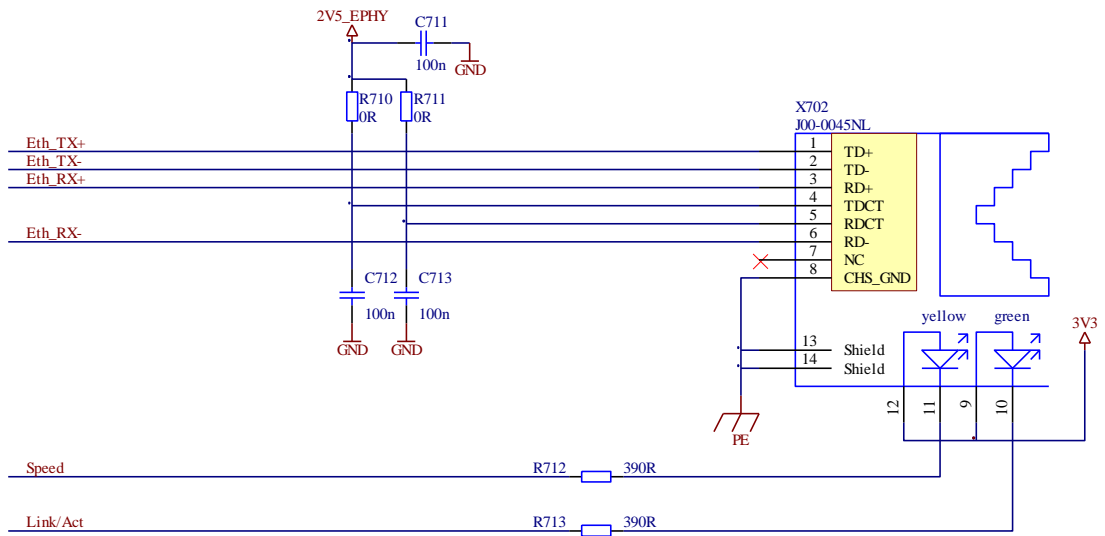
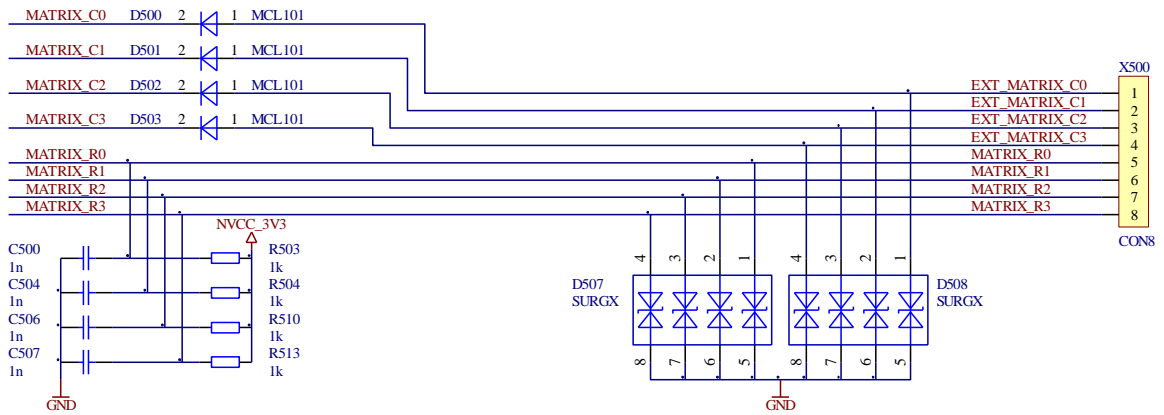


Bild 31: Referenzdesign zur Interface-Beschaltung Ethernet

# Keypad



# Scroll wheel

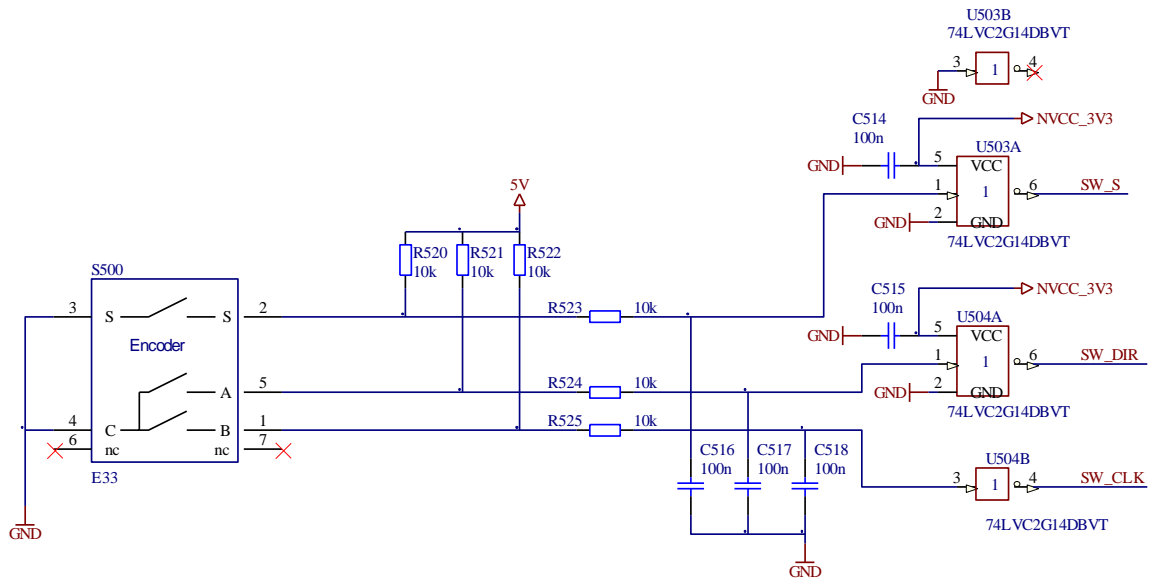


Bild 32:Referenzdesign Matrixtastatur und Scroll Wheel



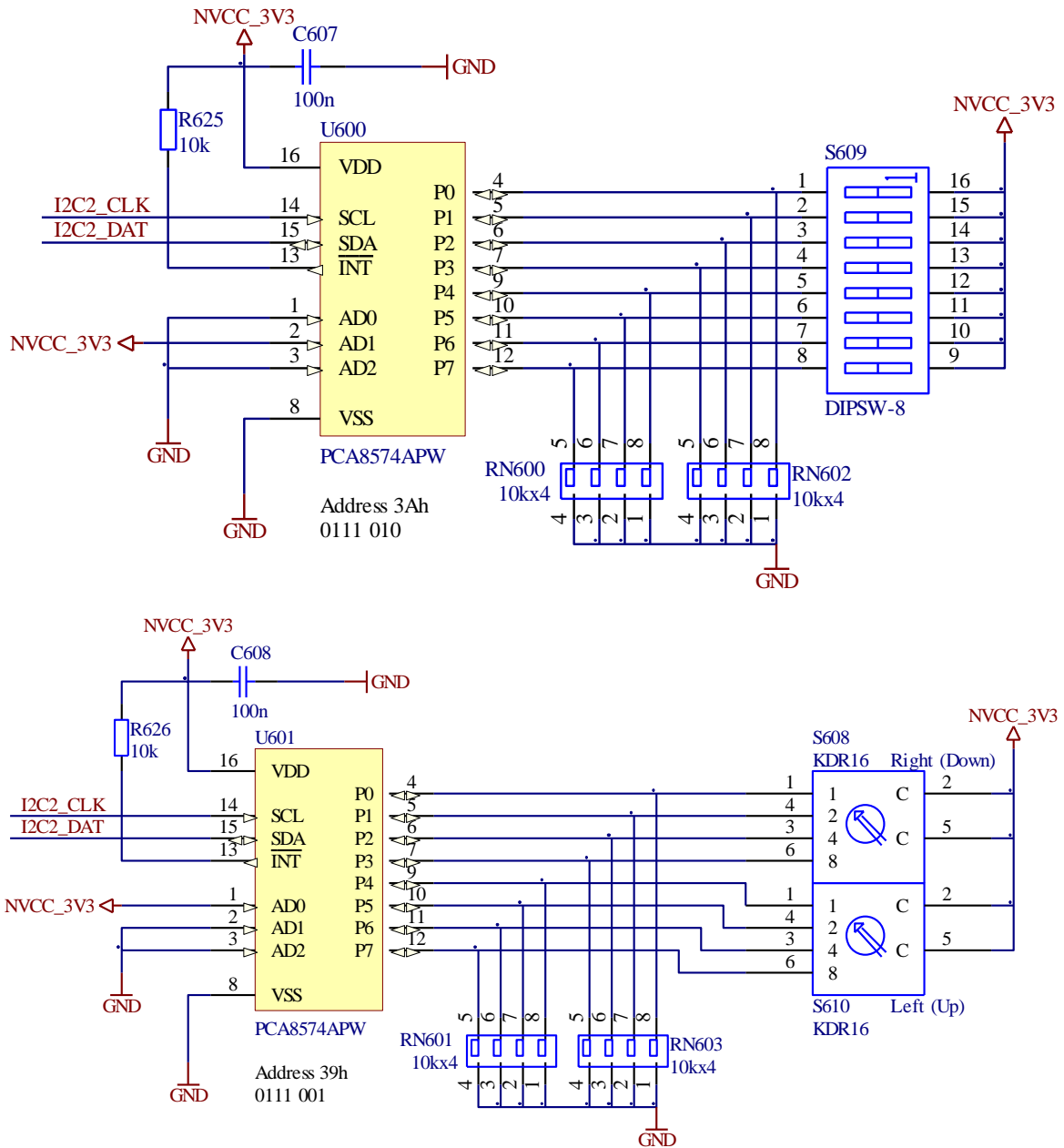


Bild 33:Referenzdesign HEX- und DIP-Switches

## Anhang B: GNU GENERAL PUBLIC LICENSE

Das auf dem PLCcore-iMX35 eingesetzte Embedded Linux ist unter der GNU General Public License, Version 2 lizenziert. Der vollständige Text dieser Lizenz ist nachfolgend aufgeführt. Eine deutsche Übersetzung ist unter <http://www.gnu.de/documents/gpl-2.0.de.html> zu finden. Es handelt sich jedoch dabei nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.

Das verwendete SPS-System sowie die vom Anwender entwickelten SPS- und C/C++ Programme unterliegen **nicht** der GNU General Public License!

### **GNU GENERAL PUBLIC LICENSE Version 2, June 1991**

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software -- to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## **GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

## How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.  
This is free software, and you are welcome to redistribute it under certain conditions;  
type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items -- whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes passes at compilers) written by James Hacker.

```
<signature of Ty Coon>, 1 April 1989  
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## Index

/	
/home.....	34
/home/etc/autostart.....	15, 27
/home/etc/pointercal.....	35, 36
/tmp.....	34, 36
<b>A</b>	
Abmessungen.....	6
adduser.....	31
Administration	
Systemvoraussetzungen.....	23
Anschlussbelegung.....	12
autostart.....	15, 27
AWL.....	7
<b>B</b>	
Boot-Bedingungen.....	24
Bootkonfiguration.....	27
<b>C</b>	
CAN0.....	9, 17
CAN1.....	17
CANopen.....	7, 17
CANopen Master.....	6
CANopen-Chip.....	7
CE-Konformität.....	4
COM.....	17
COM0.....	9, 17
COM1.....	9, 17
COM2.....	9
<b>D</b>	
date.....	33
Dateisystem.....	34
deluser.....	32
Development Kit.....	8
Developmentboard	
Anschlüsse.....	9
Bedienelemente.....	10
df 34	
<b>E</b>	
Embedded Linux.....	7
EMV-Gesetz.....	4
Entwicklungskit.....	8
ETH0.....	9, 17
<b>F</b>	
FTP	
Anmeldung am PLCcore-iMX35.....	29
FTP-Client.....	23
FUB.....	7
<b>G</b>	
GNU.....	7
GPL.....	49
<b>H</b>	
hwclock.....	33
<b>K</b>	
Kommunikationsschnittstellen	
CAN.....	17
COM.....	17
ETH.....	17
Konfiguration	
Kommandos.....	25
Konfigurationsmodus.....	24
KOP.....	7
<b>L</b>	
Linux.....	7
linuximage.....	38
<b>M</b>	
Manuale	
Übersicht.....	5
<b>N</b>	
Nutzerkonten	
Anlegen und Löschen.....	31
Passwort ändern.....	32
vordefinierte.....	28
<b>P</b>	
passwd.....	32
Pinout.....	12
Programmiersystem CODESYS.....	7
Programmierung.....	16
Prozessabbild	
Aufbau und Adressierung.....	16
<b>R</b>	
Referenzdesign.....	42
root.squashfs.....	38
RTC setzen.....	33
Run/Stop-Schalter	
Codierung.....	14
<b>S</b>	
Scrollwheel.....	7
Softwareupdate	
Linux-Image.....	38
SPS-Firmware.....	36
ST.....	7
Systemstart.....	15
Systemzeit setzen.....	33
<b>T</b>	
Telnet	
Anmeldung am PLCcore-iMX35.....	28

Telnet-Client .....	23	Ethernet Konfiguration.....	25
Terminaleinstellungen .....	25	Update Linux-Image.....	40
Terminalprogramm .....	23	USB-RS232 Adapter Kabel .....	11
TFTPD32 .....	38	<b>V</b>	
Touchscreen .....	7	vordefinierte Nutzerkonten.....	28
Kalibrierung .....	35	<b>W</b>	
ts_calibrate .....	35	WinSCP .....	30
<b>U</b>		<b>Z</b>	
U-Boot Kommandoprompt		Zubehör.....	11
Aktivierung .....	24		
Terminaleinstellungen.....	25		
U-Boot Kommandos			



**Dokument:** CODESYS System Manual PLCcore-iMX35

**Dokumentnummer:** L1573d\_5, 5. Auflage Mai 2016

---

**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  
Am Windrad 2  
D - 08468 Heinsdorfergrund  
GERMANY  
Fax : +49 (0) 3765 / 38600-4100  
Email: [info@systec-electronic.com](mailto:info@systec-electronic.com)

