

System Manual
ECUcore-5208

User Manual
Version 1.00

Ausgabe August 2009

Dokument-Nr.: L-1202d_1

SYS TEC electronic GmbH August-Bebel-Straße 29 D-07973 Greiz
Telefon: +49 (3661) 6279-0 Telefax: +49 (3661) 6279-99
Web: <http://www.systemec-electronic.com> Mail: info@systemec-electronic.com

Status/Änderungen

Status: Veröffentlicht

Datum/Version	Abschnitt	Änderung	Bearbeiter
2009/08/12	Alle	Erstellung	A. Stein

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

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

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

© Copyright 2009 SYS TEC electronic GmbH, D-07973 Greiz.

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

Informieren Sie sich:

Kontakt	Direkt	Ihr Lokaler Distributor
Adresse:	SYS TEC electronic GmbH August-Bebel-Str. 29 D-07973 Greiz GERMANY	Sie finden eine Liste unserer Distributoren unter http://www.systec-electronic.com/distributors
Angebots-Hotline:	+49 (0) 36 61 / 62 79-0 info@systec-electronic.com	
Technische Hotline:	+49 (0) 36 61 / 62 79-0 support@systec-electronic.com	
Fax:	+49 (0) 36 61 / 6 79 99	
Webseite:	http://www.systec-electronic.com	

1. Ausgabe August 2009

Inhalt

1	Einleitung	6
2	Übersicht / Wo finde ich was?	7
3	Produktbeschreibung	9
4	Development Kit ECUcore-5208	11
4.1	Übersicht.....	11
4.2	Anschlüsse und elektrische Verbindungen	12
4.3	Bedienelemente des Development Kits ECUcore-5208	13
4.4	Optionales Zubehör: USB-RS232 Adapter Kabel	14
5	Anwendung und Administration des ECUcore-5208	15
5.1	Systemvoraussetzungen und erforderliche Softwaretools	15
5.2	Systemstart des ECUcore-5208	16
5.2.1	uClinux-Autostart aktivieren bzw. deaktivieren.....	16
5.2.2	Autostart für Anwendersoftware	17
5.3	Kommandoprompt des Bootloaders "CoLilo"	18
5.4	Ethernet-Konfiguration des ECUcore-5208	19
5.5	Anmeldung am ECUcore-5208.....	22
5.5.1	Anmeldung an der Kommando-Shell.....	22
5.5.2	Der Internet Service Daemon	22
5.5.3	Anmeldung am Telnet-Server.....	22
5.5.4	Anmeldung am FTP-Server	24
5.6	Vordefinierte Nutzerkonten.....	26
5.7	Anlegen und Löschen von Nutzerkonten	26
5.8	Passwort eines Nutzerkontos ändern.....	27
5.9	Setzen der Systemzeit.....	27
5.10	Auslesen und Anzeigen von "CoLilo"-Konfigurationsdaten.....	28
5.11	Anzeigen der installierten uClinux-Version.....	29
5.12	Dateisystem des ECUcore-5208	30
5.13	Vorinstallierte Files im Verzeichnis "/home"	31
5.14	Nutzung des HTTP-Servers	32
5.15	Update des uClinux-Images	33
5.16	Update des Bootloaders "CoLilo"	35
6	VMware-Image mit Linux-Entwicklungssystem	36
6.1	Überblick.....	36
6.2	Installation des Linux-VMware-Images	36
6.3	Starten des Linux VMware-Images	36
6.4	Benutzerkonten zur Anmeldung am Linux-Entwicklungssystem.....	38
6.5	Bestimmung der IP-Adresse des Linux-Images	39
6.6	Zugriff auf das Linux-Entwicklungssystem von einem Windows-PC.....	39
6.6.1	Zugriff über die Windows-Netzwerkumgebung.....	39
6.6.2	Zugriff über Telnet-Client	40
6.7	Persönliche Konfiguration und Aktualisierung des Linux-VMware-Images.....	41
6.7.1	Anpassung von Tastaturlayout und Zeitzone	41
6.7.2	Anpassen der Desktopgröße	44
6.7.3	Festlegen einer statischen IP-Adresse für das Linux-VMware-Image	44
6.7.4	Systemaktualisierung des Linux-VMware-Images	46
6.7.5	Ändern des Computer-Namens in der Windows-Netzwerkumgebung.....	47
6.7.6	Schrumpfen des VMware-Images	47
7	Softwareentwicklung für das ECUcore-5208	48
7.1	Notwendige Schritte nach dem 1. Start.....	48
7.2	Softwarestruktur für das ECUcore-5208	48

7.3	Makefile und Umgebungsvariablen zum Erstellen von Projekten	49
7.4	I/O-Treiber für das ECUcore-5208	50
7.4.1	Einbindung des I/O-Treibers in eigene Anwenderprojekte.....	50
7.4.2	I/O-Treiber Demoprojekt.....	51
7.5	CAN-Treiber für das ECUcore-5208	52
7.5.1	Einbindung des CAN-Treibers in eigene Anwenderprojekte	52
7.5.2	CAN-Treiber Demoprojekt	52
7.6	Übertragen von Programmen auf dem ECUcore-5208	54
7.6.1	Verwendung von NFS.....	55
7.6.2	Verwendung von FTP	56
7.7	Übersetzen und Ausführen des Demoprojektes "demo"	58
7.7.1	Verwendung von "make"	58
7.7.2	Verwendung der grafischen IDE "Eclipse"	60
7.8	Konfigurieren und Übersetzen von uClinux-Image und CoLilo	68
8	Testen der Hardwareanschaltung.....	70
9	Tipps & Tricks im Umgang mit Linux bzw. uClinux	71
	Anhang A: GNU GENERAL PUBLIC LICENSE.....	74
	Index.....	79

1 Einleitung

Vielen Dank, dass Sie sich für das SYS TEC ECUcore-5208 entschieden haben. Mit diesem Produkt verfügen Sie über einen innovativen und leistungsfähigen Einplatinenrechner mit uClinux-Betriebssystem. Aufgrund seiner zahlreichen Schnittstellen auf besonders kleiner Baugröße sowie wegen seiner geringen Leistungsaufnahme eignet er sich besonders gut als Kommunikations- und Steuerrechner für Embedded Anwendungen.

Bitte nehmen Sie sich etwas Zeit, dieses Manual aufmerksam zu lesen. Es beinhaltet wichtige Informationen zur Inbetriebnahme, Konfiguration und Programmierung des ECUcore-5208. Es wird Ihnen helfen, sich mit dem Funktionsumfang und der Anwendung des ECUcore-5208 vertraut zu machen. Dieses Dokument wird ergänzt durch weitere Manuals, beispielsweise zur Hardware des Moduls. Eine Auflistung der relevanten Manuals zum ECUcore-5208 beinhaltet Tabelle 1 im Abschnitt 2. Bitte beachten Sie auch diese ergänzenden Dokumentationen.

Für weiter führende 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 ECUcore-5208

Das ECUcore-5208 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 ECUcore-5208 ist ESD empfindlich und darf nur an ESD geschützten Arbeitsplätzen von geschultem Fachpersonal ausgepackt und gehandhabt bzw. betrieben werden.

Das ECUcore-5208 ist ein Modul für den Bereich Automatisierungstechnik. Durch die Programmierbarkeit unter uClinux 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.

2 Übersicht / Wo finde ich was?

Das vorliegende Dokument beschreibt die Inbetriebnahme des ECUcore-5208 auf Basis des Development Kit ECUcore-5208 sowie die prinzipielle Vorgehensweise bei der Softwareentwicklung für dieses Modul. Für die Hardwarekomponenten wie das ECUcore-5208 selbst, die Developmentboards sowie Referenzschaltungen existieren eigene Hardware-Manuals. Softwareseitig wird das ECUcore-5208 mit einem vorinstallierten uClinux (eine spezielle Linux-Variante für Prozessoren ohne MMU) ausgeliefert. Anwendungen, die auf diesem Modul ausgeführt werden sollen, sind also dementsprechend als Linux-Programme zu entwickeln. Das Kit beinhaltet ein komplett eingerichtetes Linux-Entwicklungssystem in Form eines VMware-Images und ermöglicht so einen leichten Einstieg in die Softwareentwicklung für das ECUcore-5208. Das VMware-Image kann dabei unverändert unter verschiedenen Host-Systemen benutzt werden. Tabelle 1 listet die für das ECUcore-5208 relevanten Manuals auf.

Tabelle 1: Übersicht relevanter Manuals zum ECUcore-5208

Informationen über...	In welchem Manual?
Grundlegende Informationen zum ECUcore-5208 (Konfiguration, Administration, Anschlussbelegung, Softwareentwicklung, Referenzdesigns usw.)	In diesem Manual
Hardware-Beschreibung zum ECUcore-5208, Referenzdesigns usw.	Hardware Manual ECUcore-5208 (Manual-Nr.: L-1137)
Developmentboard für ECUcore-Module (gemeinsame Minimalversion für alle ECUcore-Module)	Hardware Manual Developmentboard ECUcores (Manual-Nr.: L-1179)
CAN-Treiber	CAN Treiber Software Manual (Manual-Nr.: L-1023)
Geeignete Nachschlagewerke zur Linux-Programmierung	<ul style="list-style-type: none"> • Advanced Programming in the UNIX Environment, Stevens Rago, Addison-Wesley • Linux/Unix Systemprogrammierung, Helmut Herold, Addison-Wesley • Linux-UNIX-Programmierung, Jürgen Wolf, Galileo Computing • GNU Function List: http://www.silicontao.com/ProgrammingGuide/GNU_function_list

- Abschnitt 4** dieses Manuals beschreibt zunächst die **elektrische Inbetriebnahme** des ECUcore-5208 auf Basis des Development Kit ECUcore-5208.
- Abschnitt 5** erläutert **Details zur Anwendung des ECUcore-5208**, so z.B. die Konfiguration und Administration des Moduls, die Anmeldung am System, Ethernetkonfiguration, Startprozess und Dateisystem.
- Abschnitt 6** beschreibt das **VMware-Image mit dem Linux-Entwicklungssystem**.
- Abschnitt 7** behandelt die Thematik der **Softwareentwicklung** für das ECUcore-5208 und erläutert dazu die **Einbindung des I/O-Treibers** in eigene Applikationen, die Vorgehensweise zum **Übersetzen** von Anwenderprogrammen, deren **Übertragung** auf das Modul sowie das **Debugging**.
- Abschnitt 9** vermittelt **Tipps & Tricks**, die den Umgang mit Linux vereinfachen helfen. Dieser Abschnitt ist insbesondere für Quereinsteiger in Linux hilfreich.

3 Produktbeschreibung

Das ECUcore-5208 erweitert die Produktpalette der Firma SYS TEC electronic GmbH im Steuerungsbereich um ein weiteres innovatives Produkt. In Form eines Aufsteckmoduls ("Core") stellt dem Anwender einen vollständigen, unter uClinux programmierbaren Einplatinenrechner mit zahlreichen Erweiterungsschnittstellen zur Verfügung. Dank der CAN- und Ethernet-Schnittstellen ist das ECUcore-5208 optimal zur Realisierung dezentraler Steuerungsaufgaben geeignet.

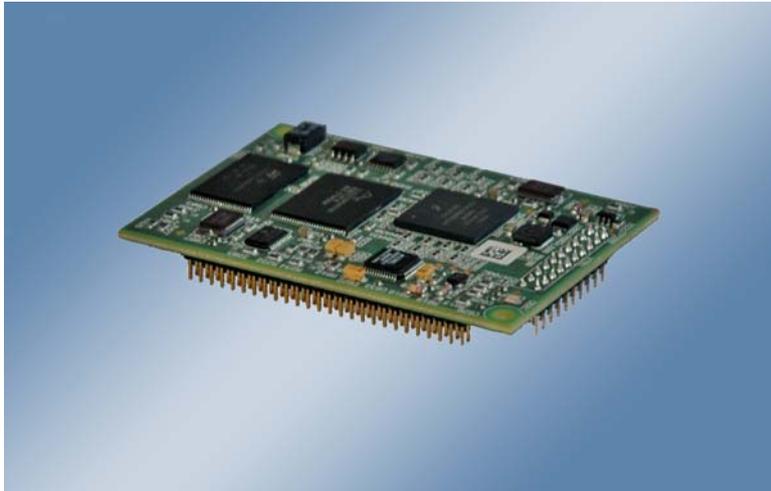


Bild 1: Ansicht des ECUcore-5208

Einige herausragende Merkmale des ECUcore-5208 sind:

- leistungsfähiger CPU-Kern (Freescale 32-Bit MCF5208 ColdFire, 166 MHz CPU Takt, 159 MIPS)
- 32 MByte SDRAM Memory, 4 MByte NOR-FLASH Memory, 64 MByte NAND-Flash Memory
- 1x 10/100 Mbps Ethernet LAN Interface (mit On-board PHY)
- 1x CAN 2.0B Interface
- 3x Asynchronous Serial Ports (UART)
- 6 digitale Eingänge, 5 digitale Ausgänge
- Queued SPI oder SPI-über-I²C und I²C extern nutzbar
- On-board Peripherie: RTC, Temperatursensor, Watchdog, Power-fail Eingang
- Betriebssystem: uClinux
- geringe Abmaße (72 * 51 mm)

Die Verfügbarkeit eines vollständigen Einplatinenrechners als aufsteckbares "Core" und die damit verbundenen geringen Abmessungen reduzieren den Aufwand und die Kosten bei der Entwicklung anwenderspezifischer Industriesteuerungen enorm. Gleichzeitig eignet sich das ECUcore-5208 besonders als intelligenter Netzwerkknoten zur dezentralen Verarbeitung von Prozesssignalen (CANopen und UDP/TCP), für Applikationen im Bereich Motion Control sowie als Basiskomponente von Spezialbaugruppen in den industriellen Bereichen.

Das ECUcore-5208 bietet 6 digitale Eingänge (DI0...DI5, 3.3V-Pegel), 5 digitale Ausgänge (DO0...DO5, 3.3V-Pegel). Die Ablage des Anwenderprogramms in der On-board Flash-Disk des Moduls ermöglicht den autarken Wiederanlauf nach einer Spannungsunterbrechung.

Das ECUcore-5208 basiert auf einem uClinux als Betriebssystem. Dadurch ist es möglich, mehrere anwenderspezifische Programme simultan abuarbeiten.

Das auf dem ECUcore-5208 eingesetzte uClinux ist unter der GNU General Public License, Version 2 lizenziert. Der entsprechende Lizenztext ist im Anhang A aufgeführt. Die vollständigen Quellen des verwendeten LinuxBSP sind im VMware-Image des Linux-Entwicklungssystems (SO-1096) 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

4 Development Kit ECUcore-5208

4.1 Übersicht

Das Development Kit ECUcore-5208 ermöglicht durch das im Kit enthaltene Developmentboard eine schnelle Inbetriebnahme des ECUcore-5208 und vereinfacht den Aufbau von Prototypen anwenderspezifischer Applikationen, die auf diesem Modul basieren. Das Developmentboard besitzt u.a. verschiedene Möglichkeiten der Spannungszufuhr, Ethernet-Schnittstelle, Anbindung an einen CAN-Bus, 6 Taster und 5 LEDs als Bedienelemente für die digitalen Ein- und Ausgänge sowie ein Potentiometer für den analogen Eingang. Die an den Steckverbindern des ECUcore-5208 verfügbaren Signale sind auf Stiftheisten geführt und ermöglichen so einen einfachen Anschluss eigener Peripherie-Schaltungen. Damit bildet das Developmentboard gleichzeitig eine ideale Experimentier- und Testplattform für das ECUcore-5208.

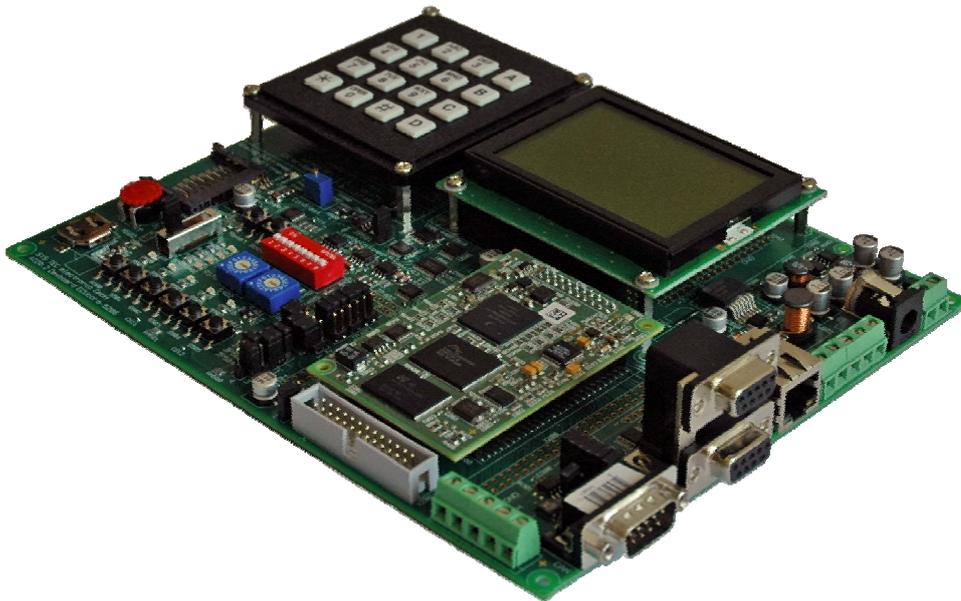


Bild 2: Development Kit ECUcore-5208

Das Development Kit ECUcore-5208 gewährleistet eine rasche und problemlose Inbetriebnahme des ECUcore-5208. Es vereint dazu alle notwendigen Hard- und Software-Komponenten, die zur Erstellung eigener Applikationen erforderlich sind: als Kernkomponente das ECUcore-5208 selbst, das zugehörige Developmentboard mit I/O-Peripherie und zahlreichen Schnittstellen, das Linux-Entwicklungssystem sowie weiteres Zubehör. Damit bildet das Developmentkit die ideale Plattform zur Entwicklung anwenderspezifischer Applikationen auf Basis des ECUcore-5208.

Das Development Kit ECUcore-5208 beinhaltet folgende Komponenten:

- ECUcore-5208
- Developmentboard für ECUcore-5208
- 24V Steckernetzteil
- Ethernet-Kabel
- RS232-Kabel
- DVD mit Linux-Entwicklungssystem, Beispielen, Dokumentation und weiteren Tools

Als Software-Entwicklungsplattform sowie Debugumgebung für das ECUcore-5208 dient das im Kit enthaltene Linux-Entwicklungssystem. In Form eines VMware-Images kann das Entwicklungssystem unverändert unter verschiedenen Host-Systemen verwendet werden. Abschnitt 6 beschreibt exemplarisch den Umgang mit dem VMware-Image unter Windows.

4.2 Anschlüsse und elektrische Verbindungen

Das für den Betrieb des Development Kit ECUcore-5208 notwendige Steckernetzteil sowie die erforderlichen Ethernet- und RS232-Kabel sind bereits im Lieferumfang des Kits enthalten. Für die Inbetriebnahme des Kit sind mindestens die Anschlüsse für Versorgungsspannung (X600/X601), COM0 (X304 oben) und ETH0 (X302) erforderlich. Einen vollständigen Überblick über die Anschlüsse des Development Kit ECUcore-5208 vermittelt Tabelle 2.

Tabelle 2: Anschlüsse des Development Kit ECUcore-5208

Anschluss	Bezeichnung auf Demoboard	Bemerkung
Versorgungsspannung	X600 oder X601	Das im Lieferumfang enthaltene Steckernetzteil ist zum direkten Anschluss an X601 vorgesehen.
ETH0 (Ethernet)	X302	Die Schnittstelle dient zur Kommunikation mit dem Linux-Entwicklungssystem (Programmier-PC) sowie zur freien Verwenden durch das Anwenderprogramm.
COM0 (RS232)	X304 / oben	Die Schnittstelle wird zur Konfiguration der Baugruppe (z.B. Setzen der IP-Adresse) und für Diagnose bzw. Debugging benötigt und ist im normalen Betrieb für das Anwenderprogramm frei verfügbar.
COM1 (RS232)	X304 / unten	Die Schnittstelle ist für das Anwenderprogramm frei verfügbar.
CAN (CAN0)	X303	Die Schnittstelle ist für das Anwenderprogramm frei verfügbar.

Bild 3 zeigt die Lage der Anschlüsse auf der Basisplatine für das Development Kit ECUcore-5208. Die Stromversorgung des Kits kann optional auch über X600 mit einer externen Quelle von 24V/1A erfolgen.

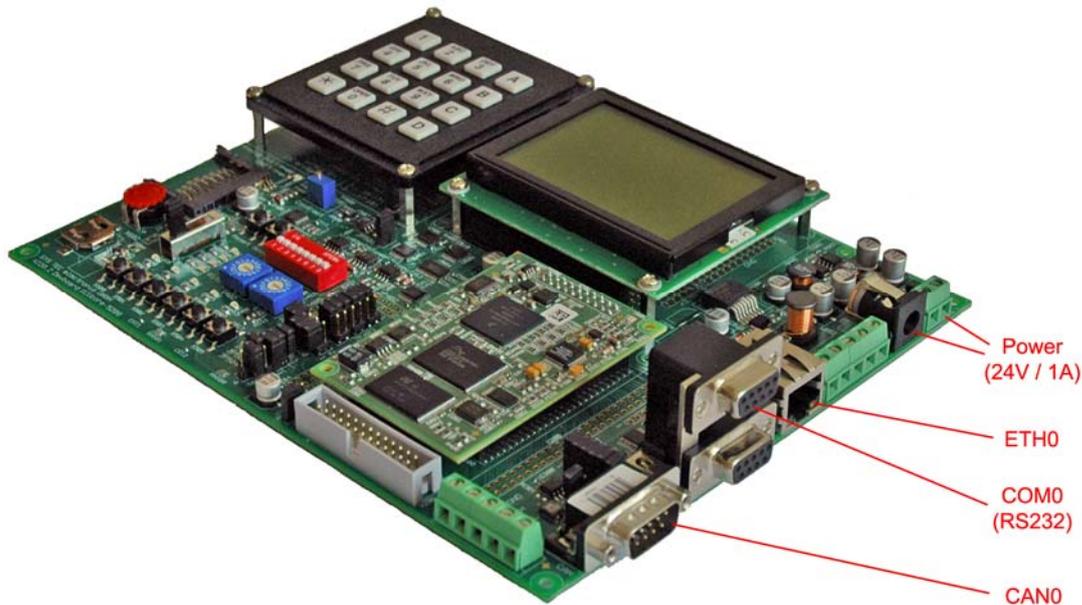


Bild 3: Lage der Anschlüsse auf der Basisplatine für das Development Kit ECUcore-5208

Hinweis: Bei der Inbetriebnahme sind zunächst die Kabel für Ethernet (ETH0, X302) und RS232 (COM0, X304 oben) anzuschließen, erst danach ist die Versorgungsspannung (X600 / X601) zu aktivieren.

4.3 Bedienelemente des Development Kits ECUcore-5208

Das Development Kit ECUcore-5208 ermöglicht eine einfache Inbetriebnahme des ECUcore-5208. Es verfügt über verschiedene Bedienelemente sowohl zur Konfiguration des Moduls als auch zur Simulation von Ein- und Ausgängen für den Einsatz des ECUcore-5208 als Kernkomponente einer Industriesteuerung. Tabelle 3 listet die einzelnen Bedienelemente des Developmentboard auf und beschreibt deren Bedeutung.

Tabelle 3: Bedienelemente des Developmentboard für das ECUcore-5208

Bedienelement	Bezeichnung	Bedeutung
Taster 0	S500	Digitaler Eingang DI0
Taster 1	S501	Digitaler Eingang DI1
Taster 2	S502	Digitaler Eingang DI2
Taster 3	S503	Digitaler Eingang DI3
Taster 4	S504	Digitaler Eingang DI4
Taster 5	S505	Digitaler Eingang DI5
LED 0	D500	Digitaler Ausgang DO0
LED 1	D501	Digitaler Ausgang DO1
LED 2	D502	Digitaler Ausgang DO2
LED 3	D503	Digitaler Ausgang DO3
LED 4	D504	Digitaler Ausgang DO4

Poti (ADC)	R407	Analoger Eingang AI0
Dimm-LED (DAC)	D405	Analoger Ausgang AO0
Run/Stop-Schalter	S405	Bedienelement ist für das Anwenderprogramm frei verfügbar (z.B. Run / Stop für Abarbeitung des Steuerprogramms)
Run-LED	D403	Bedienelement ist für das Anwenderprogramm frei verfügbar (z.B. Anzeige Aktivitätszustand des Steuerprogramms)
Error-LED	D402	Bedienelement ist für das Anwenderprogramm frei verfügbar (z.B. Anzeige Fehlerzustand des Steuerprogramms)
Hexcodier-Schalter	S402/S401	Bedienelement ist für das Anwenderprogramm frei verfügbar (z.B. Konfiguration der Knotenadresse CAN0)
DIP-Schalter	S400	Bedienelement ist für das Anwenderprogramm frei verfügbar (z.B. Konfiguration Bitrate und Master-Modus CAN0)

4.4 Optionales Zubehör: 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 und Diagnose des ECUcore-5208 von PCs, wie z.B. Laptops, die selber keine physikalische RS232-Schnittstelle mehr besitzen (siehe Abschnitt 5).



Bild 4: SYS TEC USB-RS232 Adapter Kabel

5 Anwendung und Administration des ECUcore-5208

5.1 Systemvoraussetzungen und erforderliche Softwaretools

Zur Administration des ECUcore-5208 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), das eine entsprechende RS232-Schnittstelle über einen USB-Port zur Verfügung stellt.

Alle in diesem Manual aufgeführten Beispiele beziehen sich auf die Administration des ECUcore-5208 von einem Windows-PC aus. Das Vorgehen auf einem Linux-PC ist analog.

Zur Administration des ECUcore-5208 sind folgende Softwaretools erforderlich:

Terminalprogramm Ein Terminalprogramm ermöglicht die Kommunikation mit der **Kommando-Shell** des ECUcore-5208 über eine **serielle RS232-Verbindung an COM0 des ECUcore-5208**. Diese ist Voraussetzung für die im Abschnitt 5.4 beschriebene Ethernet-Konfiguration des ECUcore-5208. 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. die Open-Source-Programme "*TeraTerm*" (Download unter: <http://ttssh2.sourceforge.jp>), "*PuTTY*" (Download unter: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>) bzw. das im Lieferumfang von Windows bereits enthaltene "*HyperTerminal*".

Telnet-Client Ein Telnet-Client ermöglicht die Kommunikation mit der **Kommando-Shell** des ECUcore-5208 über eine **Ethernet-Verbindung an ETH0 des ECUcore-5208**. Voraussetzung für die Verwendung eines Telnet-Clients ist eine abgeschlossene Ethernet-Konfiguration des ECUcore-5208 gemäß Abschnitt 5.4. Alternativ zum Telnet-Client können auch sämtliche Kommandos über ein Terminalprogramm (an COM0 des ECUcore-5208) eingegeben werden.

Als Telnet-Client eignet sich z.B. ebenfalls "*PuTTY*", was neben Telnet auch SSH unterstützt, oder "*TeraTerm*" bzw. das im Lieferumfang von Windows bereits enthaltene "*Telnet*".

FTP-Client Ein FTP-Client ermöglicht den Austausch von Dateien zwischen dem ECUcore-5208 (ETH0) und dem PC. Dies erlaubt beispielsweise das **Editieren von Konfigurationsdateien**, indem diese zunächst vom ECUcore-5208 auf den PC übertragen, dort mit einem Editor bearbeitet und anschließend wieder zurück auf das ECUcore-5208 geschrieben werden. Der Download von Dateien auf das ECUcore-5208 ist aber auch zum **Update von Softwarekomponenten** 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 zum Update des uClinux-Images auf dem ECUcore-5208 erforderlich. Standardmäßig ist bereits ein entsprechend konfigurierter

TFTP-Server im VMware-Image des Linux-Entwicklungssystems entahnten. Um alternativ ein Update des uClinux-Images von einem Windows-PC durchführen zu können, 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.

Wichtig: Es ist zu beachten, dass die Programme für das ECUcore-5208 nur innerhalb des VMware-Linux erstellt werden können.

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 Desktop-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. Gegebenenfalls ist dabei der Netzwerkadministrator zu kontaktieren.

5.2 Systemstart des ECUcore-5208

5.2.1 uClinux-Autostart aktivieren bzw. deaktivieren

Im Standardbetriebsmodus startet der Bootloader "CoLilo" bei Reset (bzw. Power-on) autark das uClinux-Betriebssystem des Moduls, das dann wiederum das Laden aller weiteren Softwarekomponenten bis hin zur Ausführung der Anwenderprogramme übernimmt (siehe Abschnitt 5.2.2). Für Servicezwecke wie beispielsweise die Konfiguration der Ethernet-Schnittstelle (siehe Abschnitt 5.4) oder zum Update des uClinux-Images (siehe Abschnitt 5.15) ist es erforderlich, diesen Autostart-Mechanismus zu unterbinden und stattdessen zum "CoLilo" Kommandoprompt zu wechseln (Konfigurationsmodus).

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

Im Detail werden durch den Bootloader "CoLilo" die in Tabelle 4 aufgelisteten Voraussetzungen geprüft, von denen alle Bedingungen für ein autarkes Booten des uClinux-Images erfüllt sein müssen.

Tabelle 4: Voraussetzungen zum Booten von uClinux

Nr.	Bedingung	Bemerkung
1	"kfl=1" und "auto=1"	In der CoLilo-Konfiguration müssen das im Flash gespeicherte uClinux-Image als gültig deklariert und der Autostart des Images nach Reset aktiviert sein (zugehörige CoLilo-Kommandos siehe Abschnitt 5.4)
2	Kein Abbruch der Autoboot-Prozedur über COM0 des ECUcore-5208	Sind alle vorangegangenen Bedingungen erfüllt, überprüft der CoLilo nach Reset für ca. 1 Sekunde die serielle Schnittstelle COM0 des ECUcore-5208 auf den Empfang eines SPACE-Zeichens (ASCII 0x20). Wird innerhalb dieser Zeit ein entsprechendes Zeichen empfangen, unterbindet der CoLilo den uClinux-Bootvorgang und aktiviert stattdessen seinen eigenen Kommandoprompt.

Gemäß Tabelle 4 wird unter folgenden Voraussetzungen der uClinux-Bootvorgang nach Reset (z.B. Taster S403 am Developmentboard) unterbunden und stattdessen der CoLilo-Kommandoprompt aktiviert:

- (1) Empfang eines SPACE-Zeichens (ASCII 20H) innerhalb 1 Sekunde nach Reset
- ODER -
- (2) In der CoLilo-Konfiguration: kfl=0
- ODER -
- (3) In der CoLilo-Konfiguration: auto=0

Nach dem Betätigen des Reset-Tasters (z.B. Taster S403 am Developmentboard) meldet sich der "CoLilo" Kommandoprompt.

Die Kommunikation mit dem Bootloader "CoLilo" erfolgt ausschließlich über die serielle Schnittstelle COM0 des ECUcore-5208. Als Gegenstelle ist auf dem PC ein beliebiges Terminalprogramm zu starten (siehe Abschnitt 5.1) und wie folgt zu konfigurieren (siehe Bild 5):

- 19200 Baud
- 8 Daten-Bits
- 1 Stopp-Bit
- keine Parität
- keine Flusskontrolle

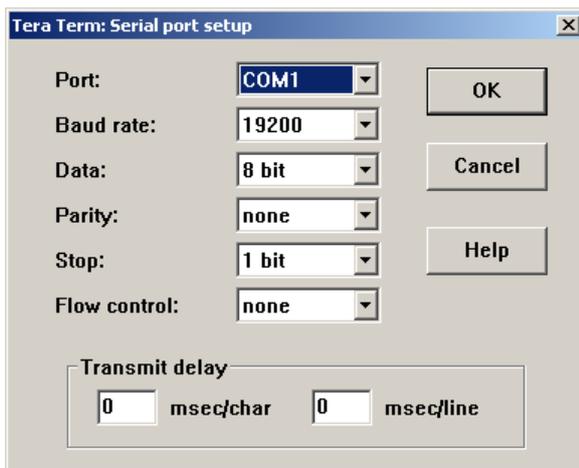


Bild 5: Terminaleinstellungen am Beispiel von "Terra Term"

5.2.2 Autostart für Anwendersoftware

Standardmäßig startet das ECUcore-5208 nach Power-on bzw. Reset das uClinux-Betriebssystem und dieses lädt wiederum alle notwendigen Softwarekomponenten zur Ausführung der Anwendersoftware. Damit eignet sich das ECUcore-5208 für den Einsatz in autarken Steuerungssystemen, die nach einer Spannungsunterbrechung selbständig und ohne Benutzeraktionen die Ausführung des Steuerprogramms wieder aufnehmen. Bild 6 zeigt den Systemstart im Detail.

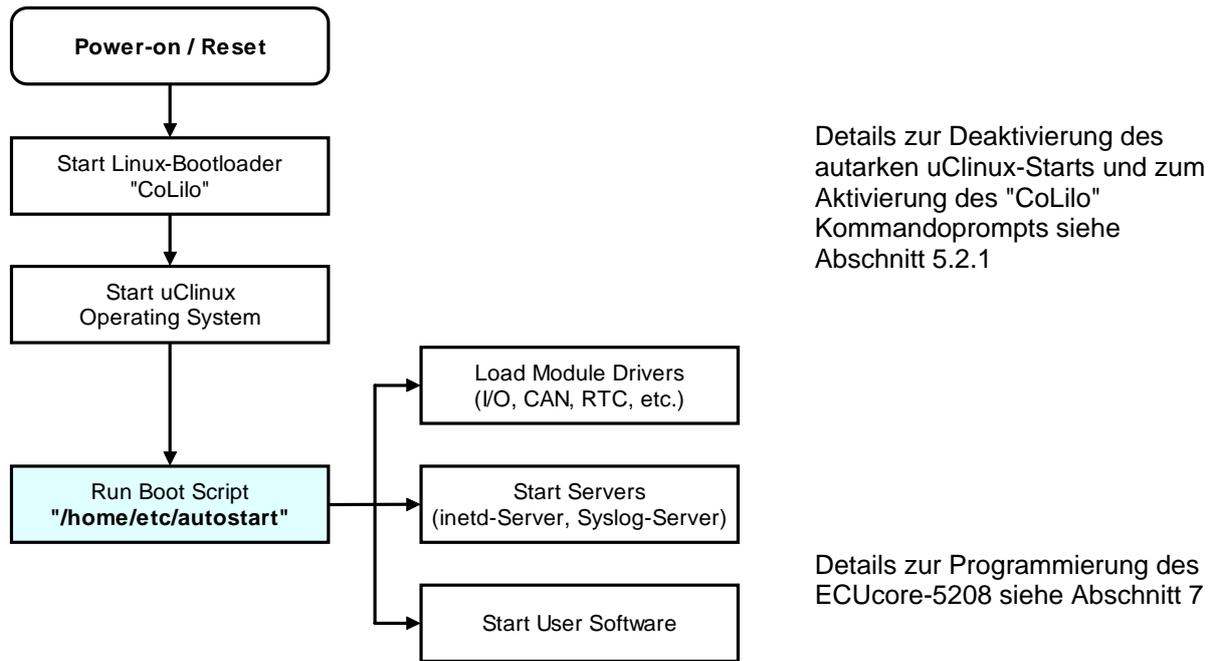


Bild 6: Systemstart des ECUcore-5208

Das ECUcore-5208 kann so konfiguriert werden, dass nach einem Reset die Anwendersoftware automatisch startet. Die dazu notwendigen Kommandos sind in dem Startskript **"/home/etc/autostart"** zu hinterlegen. Ebenso können hier bei Bedarf die notwendigen Umgebungsvariablen gesetzt sowie erforderliche Treiber geladen werden.

Das Startskript **"/home/etc/autostart"** ist entsprechend der gewünschten Funktionalität anzupassen. Das Skript lässt sich im FTP-Client *"WinSCP"* (siehe Abschnitt 5.1) mit Hilfe der Taste *"F4"* bzw. der Schaltfläche *"F4 Edit"* direkt auf dem ECUcore-5208 bearbeiten.

Dabei ist zu beachten, dass das Starten von Hintergrundprozessen mit **&** am Ende des Befehls zwar funktioniert, aber diese Prozesse beim Beenden von **"/home/etc/autostart"** als Kindprozesse auch beendet werden.

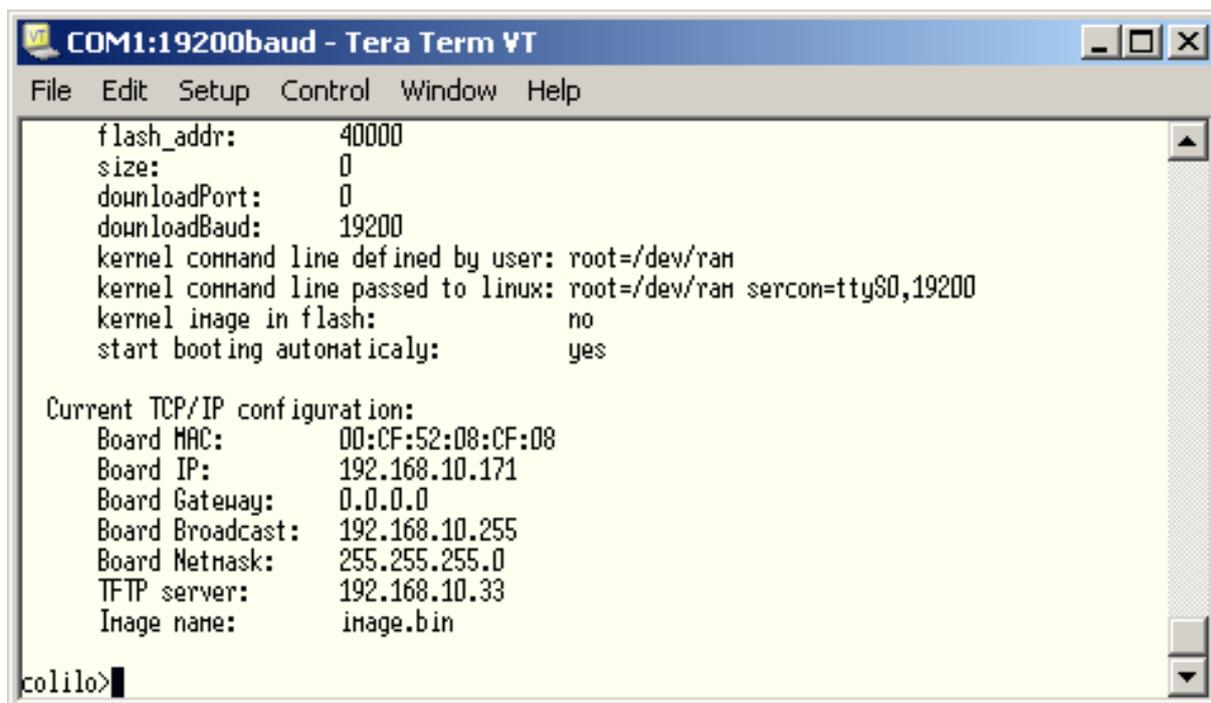
Um Programme im Hintergrund zu starten, die sich nicht selbst vom Vaterprozess trennen, kann das Programm `start-stop-daemon` verwendet werden.

5.3 Kommandoprompt des Bootloaders "CoLilo"

Der Bootloader "CoLilo" wird als erste Softwarekomponenten unmittelbar nach einem Hardware-Reset des ECUcore-5208 gestartet. Im Standardbetriebsmodus lädt der Bootloader autark das uClinux-Betriebssystem, das dann wiederum das Ausführen aller weiteren Anwenderapplikationen veranlasst. Innerhalb des Bootloader "CoLilo" werden zentrale Konfigurationseinstellungen für das ECUcore-5208 festgelegt. Primär wird der Kommandoprompt des Bootloaders "CoLilo" für folgende Aufgaben benötigt:

- Ethernet-Konfiguration des ECUcore-5208 (siehe Abschnitt 5.4) sowie
- Update des uClinux-Images (siehe Abschnitt 5.15)

Die Vorgehensweise zum Aktivieren des "CoLilo" Kommandoprompts beschreibt Abschnitt 5.2.1. Durch Eingabe von **"?"** werden eine Hilfe zu allen verfügbaren Kommandos sowie die derzeitigen Modul-Konfiguration angezeigt (siehe Bild 7).



```
COM1:19200baud - Tera Term VT
File Edit Setup Control Window Help
flash_addr:      40000
size:           0
downloadPort:    0
downloadBaud:    19200
kernel command line defined by user: root=/dev/ram
kernel command line passed to linux: root=/dev/ram sercon=ttyS0,19200
kernel image in flash:      no
start booting automaticly:  yes

Current TCP/IP configuration:
Board MAC:      00:CF:52:08:CF:08
Board IP:       192.168.10.171
Board Gateway:  0.0.0.0
Board Broadcast: 192.168.10.255
Board Netmask:  255.255.255.0
TFTP server:    192.168.10.33
Image name:     image.bin

colilo>
```

Bild 7: Anzeigen der aktuellen Modulkonfiguration im Bootloader "CoLilo"

5.4 Ethernet-Konfiguration des ECUcore-5208

Die zentrale Ethernet-Konfiguration des ECUcore-5208 erfolgt im Bootloader "CoLilo" und wird von hier für alle anderen Softwarekomponenten übernommen (uClinux, Anwendersoftware, HTTP-Server usw.) Die Ethernet-Konfiguration erfolgt über die serielle Schnittstelle COM0. **Dazu ist wie im Abschnitt 5.2.1 der CoLilo-Kommandoprompt zu aktivieren.** Tabelle 5 listet die zur Ethernet-Konfiguration des ECUcore-5208 notwendigen CoLilo-Kommandos auf.

Tabelle 5: "CoLilo" Kommandos zur Konfiguration des ECUcore-5208

Einstellung	Kommando	Bemerkung
MAC-Adresse	set mac <xx:xx:xx:xx:xx:xx>	Die MAC-Adresse ist eine weltweit eindeutige Kennung des Moduls, die bereits vom Hersteller vergeben wird. Diese sollte vom Anwender normalerweise nicht verändert werden.
IP-Adresse	set ip <xxx.xxx.xxx.xxx>	Dieses Kommando setzt die lokale IP-Adresse des ECUcore-5208. Die IP-Adresse ist vom Netzwerkadministrator festzulegen.
Netzwerkmaske	set netmask <xxx.xxx.xxx.xxx>	Dieses Kommando setzt die Netzwerkmaske des ECUcore-5208. Die Netzwerkmaske ist vom Netzwerkadministrator festzulegen.
Gateway-Adresse	set gw <xxx.xxx.xxx.xxx>	Dieses Kommando definiert die IP-Adresse des vom ECUcore-5208 zu benutzenden Gateways. Die Gateway-Adresse ist vom Netzwerkadministrator festzulegen. Hinweis: Befinden sich ECUcore-5208 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.
IP-Adresse des Linux-Entwicklungssystems	set server <xxx.xxx.xxx.xxx>	Dieses Kommando definiert die IP-Adresse des Linux-Entwicklungssystems. Sie wird beispielsweise zum Update des uClinux-Images (siehe Abschnitt 5.15) sowie zum Einbinden des Linux-Entwicklungssystems per NFS in das lokale Filesystem des ECUcore-5208 (siehe Abschnitt 7.6.1) benutzt.
Speichern der Konfiguration	config save	Dieses Kommando speichert die aktuellen Einstellungen im Flash des ECUcore-5208.

Die modifizierten Einstellungen können durch die Eingabe von "?" am "CoLilo" Kommandoprompt nochmals überprüft werden. Die aktuellen Einstellungen werden durch das Kommando

```
config save
```

persistent im Flash des ECUcore-5208 gespeichert. Die Änderungen werden mit dem nächsten Reset des ECUcore-5208 übernommen.

```

COM1:19200baud - Tera Term VT
File Edit Setup Control Window Resize Help
ECUcore-5208 boot...
Press SPACE key to abort autoboot procedure... Autoboot aborted!

colilo>set ip 192.168.10.171
IP address: '192.168.10.171'
colilo>set netmask 255.255.255.0
Netmask: '255.255.255.0'
colilo>config save
Burning 5312 bytes to flash 0x20000 from 0x8000296c
Starting sector: 1
1 128K sector(s) will be erased starting from sector 1, address 0x20000...
Erasing sector 1... ok
Programing...
...
5312 bytes written
colilo>

```

Bild 8: Speichern der Modulkonfiguration für das ECUcore-5208

Hinweis zur Verwendung von DHCP:

Das uClinux des ECUcore-5208 ist in der Lage, über DHCP eine dynamische IP-Adresse anzufordern. Dazu ist die lokale IP-Adresse als "0.0.0.0" zu konfigurieren:

```
set ip 0.0.0.0
```

Bei der Verwendung von DHCP sind folgende Punkte zu beachten:

- DHCP wird nur von uClinux unterstützt, nicht aber vom Bootloader "CoLilo". Um beispielsweise ein Update des uClinux-Images vorzunehmen (siehe Abschnitt 5.15) ist in jedem Fall die Zuweisung einer (temporären) statischen IP-Adresse an das Modul notwendig.
- Um eine Telnet- oder FTP-Verbindung zum ECUcore-5208 aufzubauen, muss dessen IP-Adresse bekannt sein. Die momentane, über DHCP dynamisch zugewiesene Adresse kann mit Hilfe eines **Terminalprogramms** (z.B. HyperTerminal oder TeraTerm, siehe Abschnitt 5.1) über die serielle Schnittstelle **COM0** des ECUcore-5208 ermittelt werden. Hierzu ist zunächst die im Abschnitt 5.5.1 beschriebene Anmeldung an der Kommando-Shell des ECUcore-5208 durchzuführen. Anschließend lässt sich die derzeitige IP-Adresse mit Hilfe des Kommandos "ifconfig" abfragen.
- Der eingesetzte DHCP-Client hat die Einschränkung, dass er beim Starten während des Bootvorgangs nicht als Dienst im Hintergrund weiterlaufen kann. Demnach wird nur einmalig beim Start eine IP-Adresse über DHCP bezogen, die nach Ablauf der Lease-Zeit **nicht** erneuert wird. Es muss daher anderweitig sicher gestellt werden, dass dieselbe IP-Adresse nicht noch einmal an einen anderen DHCP-Client vergeben wird.

Nach Abschluss der Konfiguration sind gemäß Abschnitt 5.2.1 die Voraussetzungen für ein uClinux-Autostart wieder herzustellen.

Nach Reset (z.B. Taster S403 am Developmentboard) startet das Modul mit den aktuellen Einstellungen.

Hinweis: Nach Abschluss der Konfiguration ist die serielle Verbindung zwischen PC und ECUcore-5208 nicht mehr erforderlich.

5.5 Anmeldung am ECUcore-5208

5.5.1 Anmeldung an der Kommando-Shell

Zur Administration sowie für die Softwareentwicklung ist die Eingabe von Linux-Kommandos in der Kommando-Shell des ECUcore-5208 erforderlich. Dazu ist eine direkte Anmeldung am Modul zwingend notwendig. Dies kann auf zwei alternativen Wegen erfolgen:

- mit Hilfe eines **Terminalprogramms** (siehe Abschnitt 5.1) über die serielle Schnittstelle **COM0** des ECUcore-5208, analog zum Vorgehen bei der im Abschnitt 5.4 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!
- oder alternativ mit Hilfe eines **Telnet-Clients** (siehe Abschnitt 5.1) über die Ethernet-Schnittstelle **ETH0** des ECUcore-5208 (Siehe Abschnitt 5.5.3).

5.5.2 Der Internet Service Daemon

Der Internet Service Daemon (kurz inetd) kann für das Starten bestimmter Netzdienste verwendet werden, falls eine eingehende Verbindung hergestellt wird. Im Fall des ECUcore-5208 wird über den inetd der Telnet- und der FTP-Service bereitgestellt. Standardmäßig sind aber alle inetd-Dienste außer Telnet aus Performance- und Sicherheitsgründen deaktiviert. Um FTP oder Telnet nutzen zu können, muss jeweils der Dienst in der inetd-Konfigurationsdatei `/home/etc/inetd.conf` aktiviert sein. Dazu werden im Verzeichnis `/home/bin` einige Hilfsskripte mitgeliefert, die die vorgefertigten Einträge für FTP und Telnet bearbeiten.

Z.B. kann der FTP-Dienst mit folgendem Kommando aktiviert werden:

```
enable_ftp.sh
```

Damit die Änderungen auch aktiv werden, muss der "inetd" neugestartet werden. Dazu ist das folgende Kommando notwendig:

```
/etc/rc.d/init.d/inetd.sh restart
```

Mit einer Raute (#) auskommentierte Zeilen in der Datei `/home/etc/inetd.conf` werden als Kommentare angesehen und ignoriert. Dienste, deren Zeile eine Raute vorangestellt ist, sind damit deaktiviert. Falls der FTP-Dienst aktiviert wurde, kann auch über FTP (bei einem Login mit entsprechenden Rechten wie z.B. root) die Datei `/home/etc/inetd.conf` bearbeitet werden.

Neben Telnet und FTP wird noch discard mitgeliefert, was für TCP- und UDP-Tests eingesetzt werden kann. Es werden alle Pakete empfangen und ggf. quittiert, aber die Nutzdaten sofort verworfen. Da durch eine hohe Last am "discard"-Dienst das System sehr stark ausgelastet werden kann, ist es ratsam, diesen Dienst generell zu deaktivieren oder nur zu Tests der Netzwerkverbindung zu benutzen.

5.5.3 Anmeldung am Telnet-Server

Der Telnet-Server wird indirekt über den Internet-Daemon "*inetd*" gestartet. Damit Telnet zur Verfügung steht, muss, falls nicht schon geschehen, in der inetd-Konfiguration telnet aktiviert werden. Das Vorgehen hierzu ist im Abschnitt 5.5.2 beschrieben.

Um sich über PuTTY am ECUcore-5208 anzumelden, ist die in Abschnitt 5.4 festgelegte IP-Adresse für das ECUcore-5208 einzutragen und bei "Connection type" "Telnet" auszuwählen. Durch das Betätigen der Schaltfläche "Open" wird eine Telnetverbindung hergestellt (siehe Bild 9).

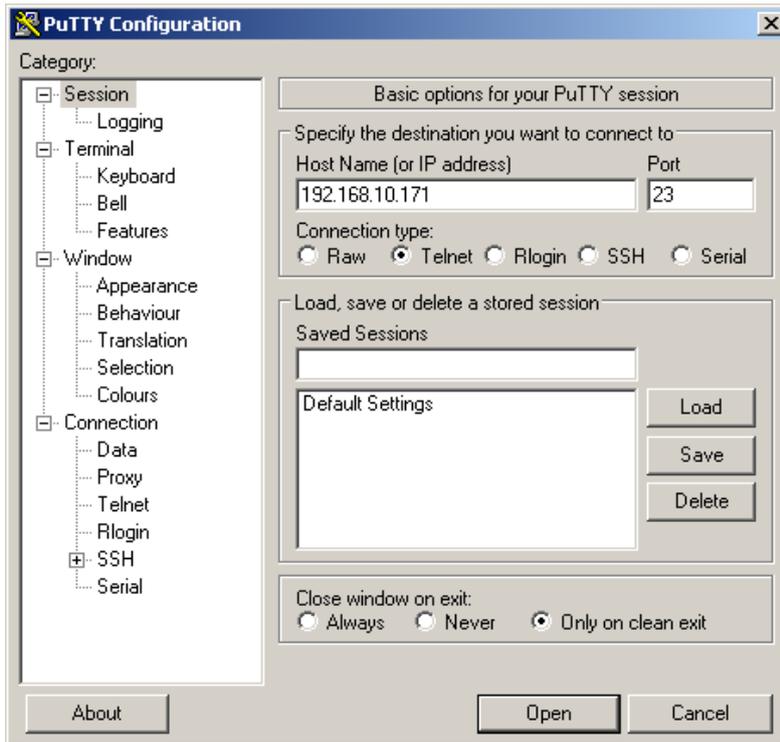


Bild 9: PuTTY-Telnet-Konfiguration für das ECUcore-5208

Innerhalb des Telnet-Fensters (bei Verwendung von ETH0) ist die Anmeldung am ECUcore-5208 möglich. Bei Auslieferung des Moduls ist zur Administration des ECUcore-5208 folgendes Nutzerkonto vorhanden:

User: *PlcAdmin*
 Passwort: *Plc123*

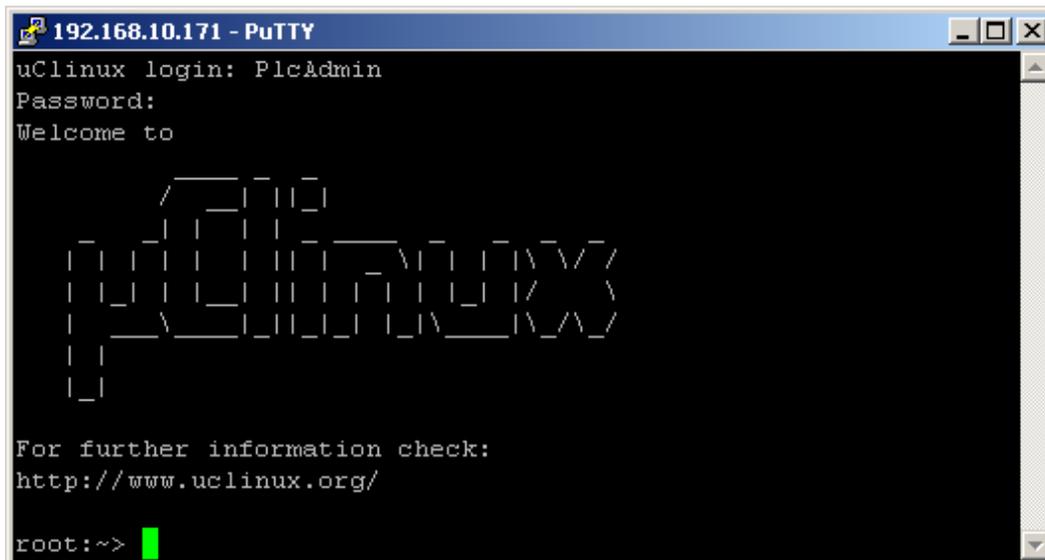


Bild 10: Anmeldung am ECUcore-5208 mit PuTTY

Um sich alternativ über den in Windows standardmäßig enthaltenen Telnet-Client am ECUcore-5208 anzumelden, ist der Befehl *"telnet"* unter Angabe der in Abschnitt 5.4 festgelegten IP-Adresse für das ECUcore-5208 aufzurufen, z.B.

```
telnet 192.168.10.171
```

Die Anmeldung selbst erfolgt analog wie in der oben genannten Methode mit PuTTY.

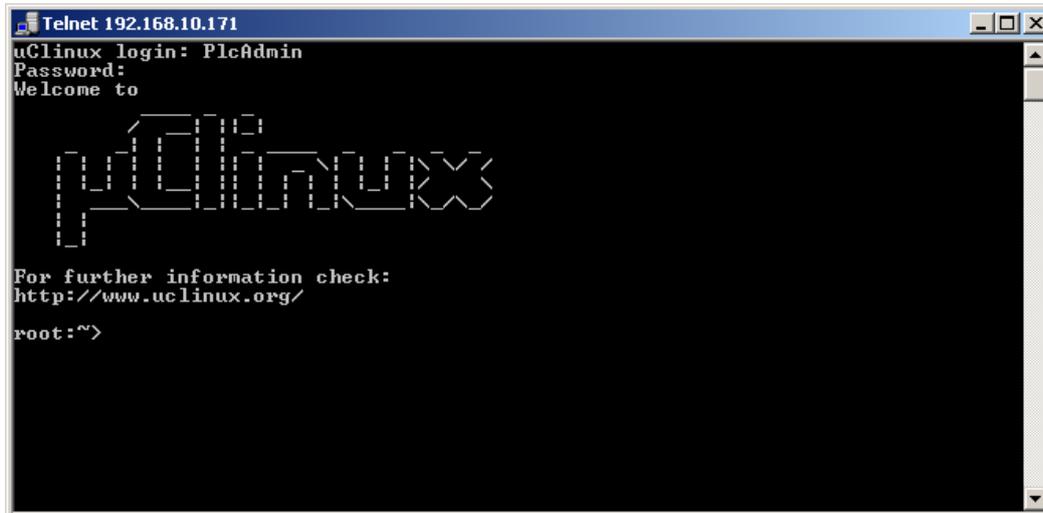


Bild 11: Anmeldung am ECUcore-5208 mit dem Windows-Telnet-Client

Hinweis: Die Datenübertragung erfolgt bei Telnet im Klartext, d.h. dass Name und Passwort unverschlüsselt übertragen werden.

5.5.4 Anmeldung am FTP-Server

Das ECUcore-5208 verfügt über einen FTP-Dienst (FTP Deamon), der den Austausch von Dateien mit einem beliebigen FTP-Client ermöglicht (z.B. Up- und Download von Dateien zu bzw. von einem PC). Aus Sicherheits- und Performance-Gründen ist dieser FTP-Dienst jedoch standardmäßig deaktiviert und muss bei Bedarf aktiviert werden. Da der FTP-Dienst indirekt über den Internet-Daemon "*inetd*" gestartet wird, muss in dessen Konfiguration *ftp* aktiviert werden. Das Vorgehen hierzu ist im Abschnitt 5.5.2 beschrieben.

Als FTP-Client für einen Windows-PC eignet sich beispielsweise das als Open-Source verfügbare "*WinSCP*" (siehe Abschnitt 5.1), 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 12), in dem folgende Einstellungen vorzunehmen sind:

File protocol:	FTP
Host name:	die im Abschnitt 5.4 festgelegte IP-Adresse für das ECUcore-5208
User name:	<i>PlcAdmin</i> (für vorkonfiguriertes Nutzerkonto, siehe Abschnitt 5.6)
Password:	<i>Plc123</i> (für vorkonfiguriertes Nutzerkonto, siehe Abschnitt 5.6)

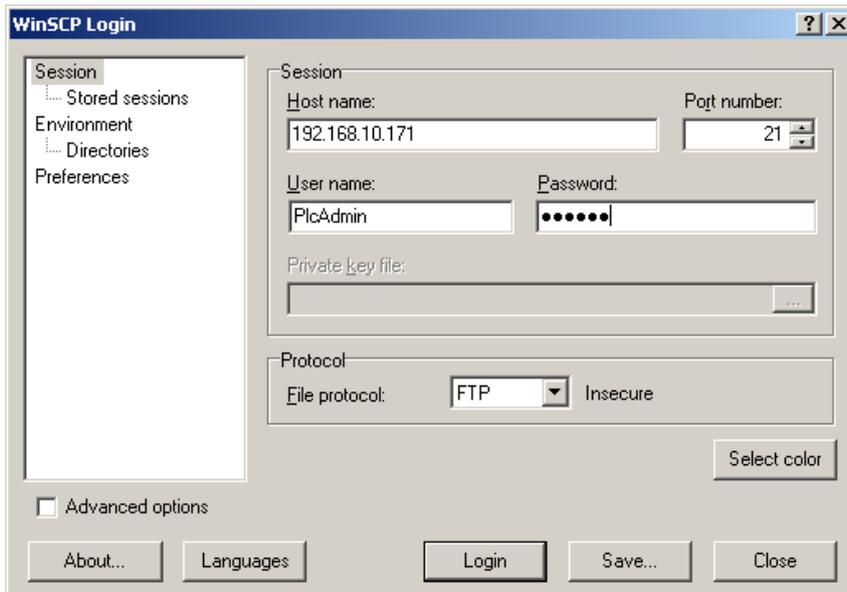


Bild 12: Login-Einstellungen für WinSCP

Nach dem Betätigen der Schaltfläche "Login" meldet sich der FTP-Client am ECUcore-5208 an und listet im rechten Fenster den aktuellen Inhalt des Verzeichnisses "/home" auf. Bild 13 zeigt den FTP-Client "WinSCP" nach der erfolgreichen Anmeldung am ECUcore-5208.

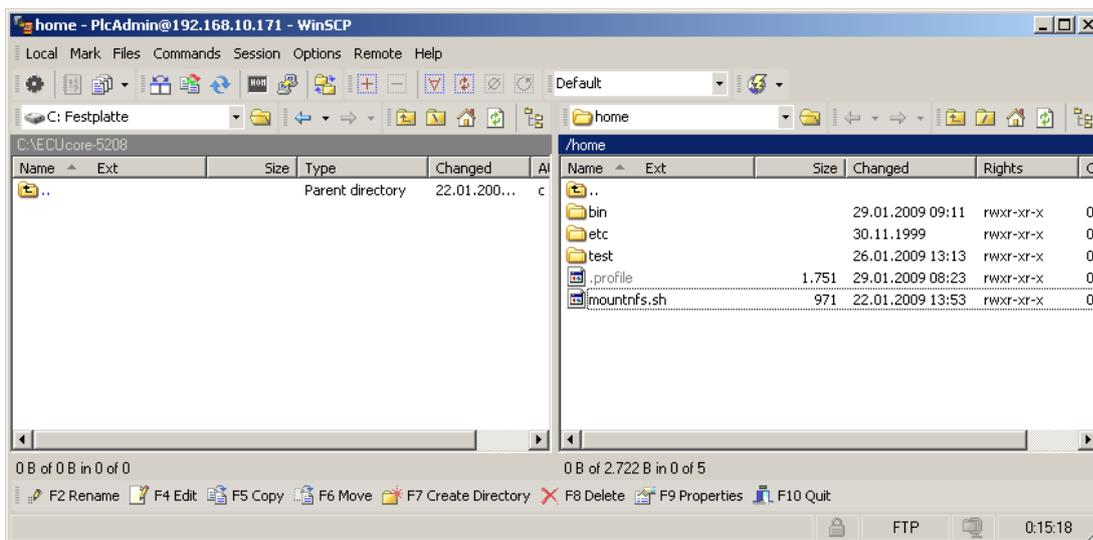


Bild 13: FTP-Client für Windows "WinSCP"

Nach einer erfolgreichen Anmeldung lassen sich innerhalb des FTP-Clients "WinSCP" mit Hilfe der Taste "F4" bzw. der Schaltfläche "F4 Edit" Konfigurationsdateien auf dem ECUcore-5208 bearbeiten (dazu Transfermodus "Text" selektieren). Mit Hilfe der Taste "F5" bzw. der Schaltfläche "F5 Copy" können Dateien zwischen dem PC und dem ECUcore-5208 transferieren werden, um beispielsweise Datensicherungen des ECUcore-5208 durchzuführen oder um Software-Updates auf das Modul zu übertragen (dazu Transfermodus "Binary" selektieren).

Hinweis: Die Datenübertragung erfolgt bei FTP im Klartext, d.h. dass Name und Passwort unverschlüsselt übertragen werden.

5.6 Vordefinierte Nutzerkonten

Bei der Auslieferung des ECUcore-5208 sind die in Tabelle 6 aufgelisteten Benutzerkonten vordefiniert. Diese erlauben eine Anmeldung an der Kommando-Shell (serielle RS232-Verbindung oder Telnet) und am FTP-Server des ECUcore-5208.

Tabelle 6: Vordefinierte Benutzerkonten des ECUcore-5208

Benutzername	Passwort	Bemerkung
PlcAdmin	Plc123	vordefiniertes Benutzerkonto zur Administration des ECUcore-5208 (Konfiguration, Nutzerverwaltung, Softwareupdates usw.)
root	Sys123	Haupt-Benutzerkonto ("root") des ECUcore-5208

5.7 Anlegen und Löschen von Nutzerkonten

Das Anlegen und Löschen von Nutzerkonten erfordert zunächst die Anmeldung am ECUcore-5208 wie in Abschnitt 5.5.1 beschrieben.

Das **Anlegen** eines neuen Nutzerkontos erfolgt mit Hilfe des Linux-Kommandos "*adduser*". Da es bei einem Embedded System wie dem ECUcore-5208 nicht unbedingt sinnvoll ist, für jeden Benutzer ein eigenes Verzeichnis anzulegen, wird mit dem Parameter "*-H*" das Erstellen eines neuen Verzeichnisses zu unterbunden. Stattdessen kann dem neuen Benutzer über den Parameter "*-h /home*" das bereits vorhandene Verzeichnis "*/home*" zugeordnet werden. Zum Anlegen eines neuen Nutzerkontos auf dem ECUcore-5208 ist das Linux-Kommandos "*adduser*" wie folgt anzuwenden:

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

Bild 14 verdeutlicht am Beispiel das Anlegen eines neuen Kontos auf dem ECUcore-5208 für den Nutzer "*admin2*".

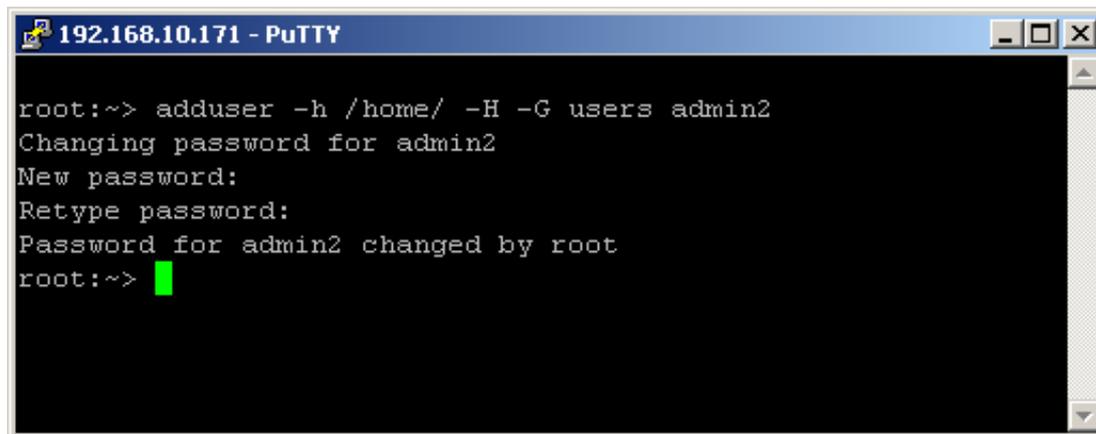


Bild 14: Anlegen eines neuen Nutzerkontos

Zum **Löschen** eines vorhandenen Nutzerkontos vom ECUcore-5208 ist das Linux-Kommando "*deluser*" unter Angabe des Benutzernamens zu verwenden:

```
deluser <username>
```

Falls beim Löschen eines Benutzers die Meldung

```
deluser: can't find <username> in /home/etc/group
```

erscheint, kann diese ignoriert werden.

5.8 Passwort eines Nutzerkontos ändern

Das Ändern von Passwörtern erfordert zunächst die Anmeldung am ECUcore-5208 wie in Abschnitt 5.5.1 beschrieben.

Zum Ändern des Passwortes für ein auf dem ECUcore-5208 vorhandenes Nutzerkonto ist das Linux-Kommando `"passwd"` unter Angabe des Benutzernamens zu verwenden:

```
passwd <username>
```

Bild 15 verdeutlicht am Beispiel das Ändern des Passwortes für den Nutzer `"PlcAdmin"`. Wird der Benutzername weggelassen, wird das Passwort für den aktuellen Benutzer aktualisiert.

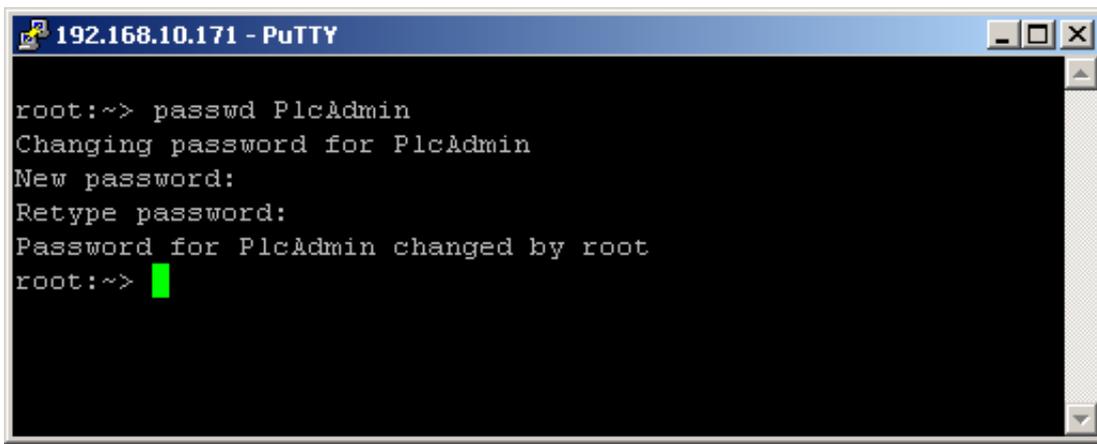


Bild 15: Passwort eines Nutzerkontos ändern

5.9 Setzen der Systemzeit

Das Setzen der Systemzeit erfordert zunächst die Anmeldung am ECUcore-5208 wie in Abschnitt 5.5.1 beschrieben.

Das Setzen der Systemzeit für das ECUcore-5208 erfolgt in zwei Schritten. Zuerst sind Datum und Uhrzeit mit Hilfe des Kommandos `"date"` auf die aktuellen Werte zu setzen. Anschließend wird die Systemzeit durch das Kommando `"hwclock -w"` in den RTC-Baustein des ECUcore-5208 übernommen.

Das Kommando `"date"` besitzt folgenden Aufbau:

```
date [options] [MMDDhhmm[[CC]YY][.ss]]
```

Beispiel:

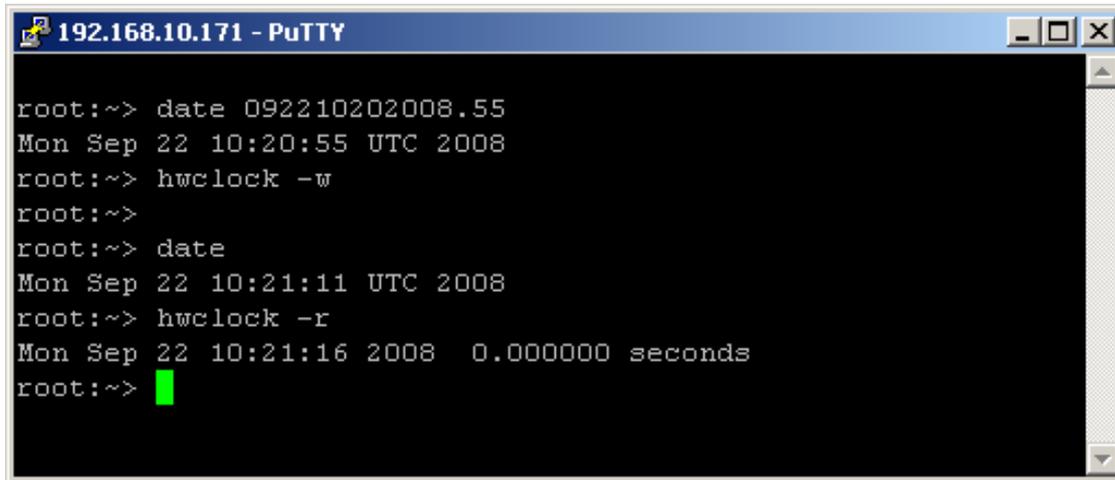
```
date    09  22  10  20  2008.55
        |   |   |   |   |   |
        |   |   |   |   |   +--- Sekunden
        |   |   |   |   +----- Jahr
        |   |   |   +----- Minute
        |   |   +----- Stunde
        |   +----- Tag
        +----- Monat
```

Die Leerzeichen innerhalb der Parameterliste sind hier nur zur Verbesserung der Übersichtlichkeit eingefügt und müssen bei der realen Verwendung des Kommandos entfallen. Um die aktuelle

Systemzeit des ECUcore-5208 wie im obigen Beispiel dargestellt auf den 2008/09/22 um 10:20:55 zu setzen, ist folgende Befehlssequenz notwendig:

```
date 092210202008.55
hwclock -w
```

Die aktuelle Systemzeit wird durch Eingabe des Kommandos "date" (ohne Parameter) angezeigt, die aktuellen Werte der RTC lassen sich durch das Linux-Kommando "hwclock -r" abfragen. Bild 16 verdeutlicht das Setzen und Anzeigen der Systemzeit am Beispiel.

The image shows a terminal window titled "192.168.10.171 - PuTTY". The terminal output is as follows:

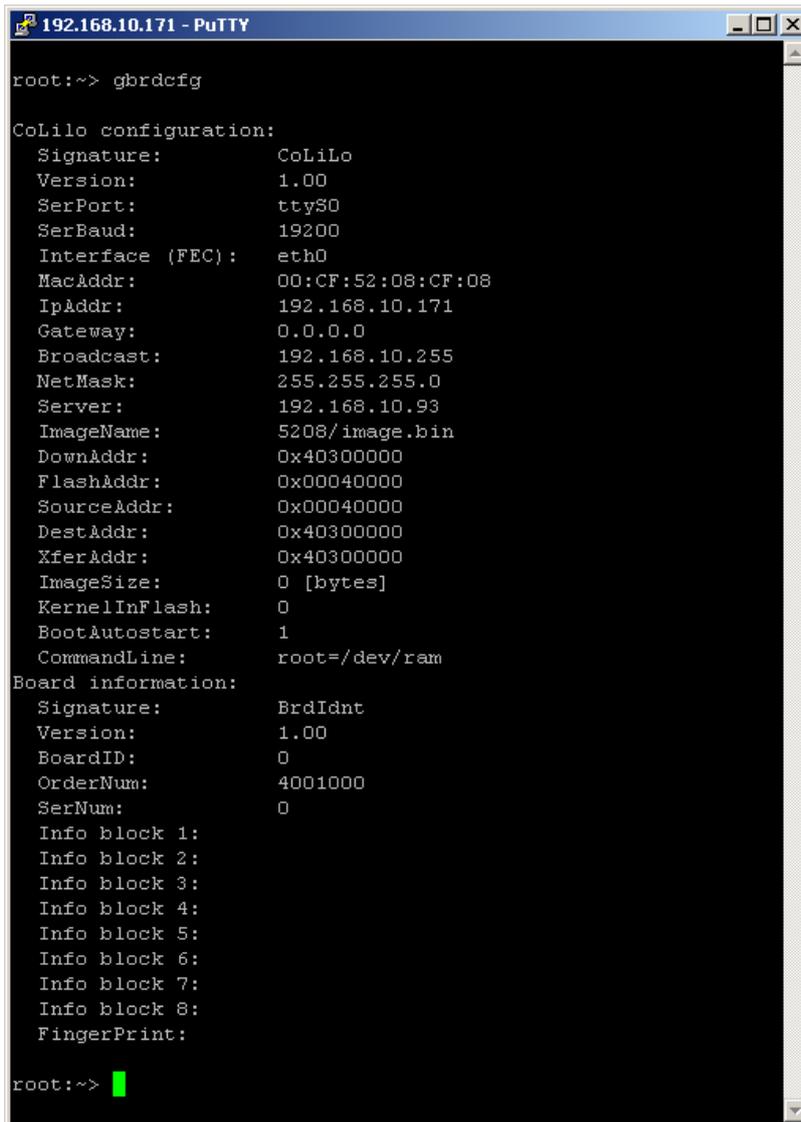
```
root:~> date 092210202008.55
Mon Sep 22 10:20:55 UTC 2008
root:~> hwclock -w
root:~>
root:~> date
Mon Sep 22 10:21:11 UTC 2008
root:~> hwclock -r
Mon Sep 22 10:21:16 2008  0.000000 seconds
root:~>
```

Bild 16: Setzen und Anzeigen der Systemzeit

Beim Starten des ECUcore-5208 übernimmt der uClinux-Kernel für das Modul Datum und Uhrzeit der RTC als aktuelle Systemzeit.

5.10 Auslesen und Anzeigen von "CoLilo"-Konfigurationsdaten

Das Kommando "gbrdcfg" ermöglicht unter uClinux den Zugriff auf die Konfigurationsdaten des Bootloaders "CoLilo". Es wird beispielsweise genutzt, um im Startscript "/etc/rc.d/rcS" die im "CoLilo" für das ECUcore-5208 festgelegte Ethernet-Konfiguration (siehe Abschnitt 5.4) zur Parametrierung des Interfaces "eth0" unter uClinux weiter zu verwenden. Gleiches gilt auch für die Übernahme der im "CoLilo" definierten Server-Adresse in den Shell-Skripten "/home/mountnfs.sh" und "/home/debug.sh".



```

root:~> gbrdcfg

CoLilo configuration:
  Signature:      CoLiLo
  Version:        1.00
  SerPort:        ttyS0
  SerBaud:        19200
  Interface (FEC): eth0
  MacAddr:        00:CF:52:08:CF:08
  IpAddr:         192.168.10.171
  Gateway:        0.0.0.0
  Broadcast:      192.168.10.255
  NetMask:        255.255.255.0
  Server:         192.168.10.93
  ImageName:      5208/image.bin
  DownAddr:       0x40300000
  FlashAddr:      0x00040000
  SourceAddr:     0x00040000
  DestAddr:       0x40300000
  XferAddr:       0x40300000
  ImageSize:      0 [bytes]
  KernelInFlash: 0
  BootAutostart:  1
  CommandLine:    root=/dev/ram

Board information:
  Signature:      BrdIdnt
  Version:        1.00
  BoardID:        0
  OrderNum:       4001000
  SerNum:         0
  Info block 1:
  Info block 2:
  Info block 3:
  Info block 4:
  Info block 5:
  Info block 6:
  Info block 7:
  Info block 8:
  FingerPrint:

root:~> █

```

Bild 17: Anzeigen der "CoLilo"-Konfigurationsdaten unter uClinux mittels "gbrdcfg"

Der Aufruf von "gbrdcfg" ohne Angabe von Parametern zeigt alle "CoLilo"-Konfigurationsdaten an (siehe Bild 17). Der Aufruf von "gbrdcfg -help" listet die Parameter zur gezielten Abfrage einzelner Konfigurationsinformationen auf. Die oben angeführten Skripte verdeutlichen die Anwendung von "gbrdcfg", demonstrieren die Zuweisung der abgefragten Informationen an Umgebungsvariablen und zeigen die Auswertung der Informationen in einem Shell-Script.

5.11 Anzeigen der installierten uClinux-Version

Die auf dem ECUcore-5208 installierte uClinux-Version wird durch Aufruf des Kommandos "version" angezeigt (siehe Bild 18).

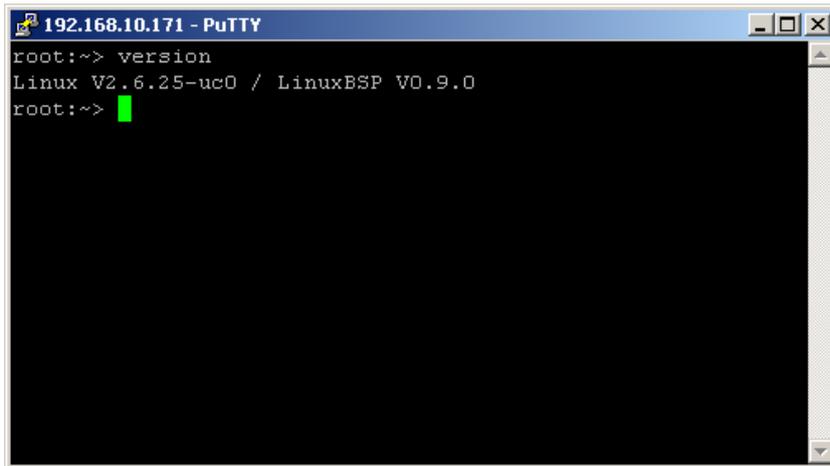


Bild 18: Anzeigen der installierten uClinux-Version

5.12 Dateisystem des ECUcore-5208

Das auf dem ECUcore-5208 vorinstallierte uClinux stellt Teile des Systemspeichers in Form eines Dateisystems zur Verfügung. Wie bei Embedded Systemen üblich, ist dabei der überwiegende Teil des Dateisystems read-only, d.h., dass Änderungen an diesem Teil nur durch Neuerstellung des uClinux-Images für das ECUcore-5208 auf dem Entwicklungs-PC vorgenommen werden können. Der große Vorteil besteht hier in der Resistenz des read-only Dateisystems gegen Beschädigungen beim Ausfall der Spannungsversorgung, wie sie bei Embedded Systemen häufig auftreten. Embedded Systeme werden in der Regel einfach ausgeschaltet, ohne dass zuvor ein Shutdown-Befehl ausgeführt wird.

Die für den Anwender zur Laufzeit beschreibbaren Zweige des Dateisystems listet Tabelle 7 auf. Hinter dem Verzeichnis **"/home"** verbirgt sich eine Flash-Disk, die einen Teil des on-board Flash-Speichers des ECUcore-5208 als Dateisystem zur Verfügung stellt. In diesem Pfad werden sämtliche vom Anwender modifizierbaren und updatefähigen Files wie beispielsweise Konfigurationsdateien und die auf das Modul geladenen Anwenderprogramme abgelegt. Generell sollte für Tests während der Entwicklungsphase das RAM-Disk-Verzeichnis **"/tmp"** benutzt werden, falls nicht ohnehin Teile des Entwicklungs-PCs via NFS eingebunden sind (siehe Abschnitt 7.6.1).

Tabelle 7: Dateisystemkonfiguration des ECUcore-5208

Zweig	Nutzbare Größe	Beschreibung
/tmp	von /var bestimmt	Teil einer RAM-Disk, gut geeignet zum Testen von Programmen, aber kein Datenerhalt bei Spannungsausfall
/var	2048 kByte	RAM-Disk, für Dateien die zur Laufzeit erstellt werden, wie z.B. Logfiles; kein Datenerhalt bei Spannungsausfall
/home	65536 kByte	Flash-Disk, zur persistenten Ablage von Anwenderprogrammen, Konfigurationsdateien usw., Datenerhalt bei Spannungsausfall ist gegeben
/mnt		Ziel für die Einbindung von Remote-Verzeichnissen anderer Systeme via NFS, siehe dazu auch Abschnitt 7.6.1

Hinweis: Die Größe für den Dateisystemzweig **"/var"** kann über die uClinux-Konfiguration geändert werden. Dazu muss ein anderer Wert für **"RAMFS Image"** (unterhalb von **"Miscellaneous Configuration --->"**) eingestellt werden. Die Größe für **"/tmp"** muss nicht extra eingestellt werden, da es ein Teil von **"/var"** ist. Dies erfordert jedoch die Neuerstellung des uClinux-Images für das ECUcore-5208.

Die aktuell konfigurierten, sowie die jeweils noch verfügbaren Größen der einzelnen Dateisystemzweige können mit Hilfe des Kommandos **"df"** ("DiskFree") ermittelt werden. Bild 19 zeigt

ein Beispiel für die Anwendung des Kommandos. Da "/tmp" ein Teil von "/var" ist, wird es bei "df" auch gar nicht aufgelistet.

```

root:~> df
Filesystem      1k-blocks    Used Available Use% Mounted on
/dev/mtdblock0    1275      1275         0 100% /
/dev/ram1         1979         8     1971   0% /var
/dev/mtdblock4   2048      236     1812  12% /home
root:~>

```

Bild 19: Beispiel für die Anwendung des Kommandos "df"

5.13 Vorinstallierte Files im Verzeichnis "/home"

An das Verzeichnis "/home" ist eine Flash-Disk gebunden ("mounted"), die einen Teil des on-board Flash-Speichers des ECUcore-5208 als Dateisystem zur Verfügung stellt. Dieser Pfad ist zur Laufzeit beschreibbar und dient zur persistenten Ablage modifizierbarer und updatefähiger Dateien wie Konfigurationsdateien oder Anwenderprogramme (siehe auch Abschnitt 5.12). In seinem Auslieferungszustand enthält das ECUcore-5208 im Verzeichnis "/home" folgende Dateien:

/home	
+ - etc	
+ - autostart	Skript zum automatisierten Start von Anwendersoftware (siehe Abschnitt 5.2.2)
+ - rc.usr	systemweites Konfigurationskript, welches beim Start ausgeführt wird
+ - profile	Systemweites Konfigurationskript wird bei jeder Login-Shell ausgeführt, ermöglicht z.B. das Setzen von Umgebungsvariablen.
+ - resolv.conf	DNS-lookup-Konfigurationsdatei, ist entsprechend anzupassen, um dem ECUcore-5208 DNS-Namensauflösungen zu ermöglichen
	Folgende Dateien werden nach einem Reset neu angelegt, falls sie nicht existieren (z.B. beim 1. Start)
+ - passwd	Dateien zur Nutzerverwaltung, werden von Programmen wie <i>login</i> , <i>adduser</i> , <i>deluser</i> , <i>passwd</i> , <i>ftpd</i> usw. benutzt und sollten nur indirekt über diese Programme bearbeitet werden
+ - group	
+ - inetd.conf	Konfigurationsdatei für inetd
+ - motd	Message of the Day. Wird nach jedem Shell-Login ausgegeben
+ - bin	
+ - pc5208drv.ko	Kernelspace-Modul des I/O-Treibers (siehe Abschnitt 7.4.1)
+ - 5208-spidev.ko	Kernelspace-Modul für den I/O-Treiber
+ - candrv.ko	Kernelspace-Modul des CAN-Treibers (siehe Abschnitt 7.5.1)

+ - http	Demo-Dateien für HTTP-Server (siehe Abschnitt 5.14)
+ - mountnfs.sh	Skript zum vereinfachten Einbinden von NFS-Verzeichnissen in das Filesystem des ECUcore-5208 (siehe Abschnitt 7.6.1)
+ - debug.sh	Skript zum vereinfachten Starten des Demoprogramms unter Kontrolle des Debugservers (siehe Abschnitt 7.7.2.3)
+ - .profile	Anwenderspezifisches Konfigurationskript wird bei jeder Login-Shell ausgeführt

Bei Bedarf kann der Auslieferungszustand aller Dateien im Verzeichnis `/home` durch Ausführen des Setup-Skripts `setup-ecucore-5208.sh` wieder restauriert werden. Das Setup-Skript `setup-ecucore-5208.sh` befindet sich im Verzeichnis `SetupFlashdisk_ECUcore-5208` der DVD "SO-1096". Im Abschnitt 7.6 sind verschiedene alternative Möglichkeiten zur Übertragung des Files auf das ECUcore-5208 beschrieben.

5.14 Nutzung des HTTP-Servers

Auf dem ECUcore-5208 ist der HTTP-Server `boa` installiert. Dadurch ist der Zugriff auf das Modul mit einem beliebigen WEB-Browser möglich (z.B. Mozilla Firefox, Microsoft Internet Explorer, Opera usw.). Die Konfiguration des Servers erfolgt mit Hilfe der Datei `boa.conf`. Beim Starten des Servers ist das Verzeichnis, das diese Datei enthält, über den Kommandozeilenparameter `-c` anzugeben. Im Auslieferungszustand enthält das ECUcore-5208 im Verzeichnis `/home/http` vorgefertigte Demo-Dateien, mit deren Hilfe sich die Anwendung des HTTP-Servers veranschaulichen lässt. Um diese Demo-Konfiguration zu aktivieren, ist der HTTP-Server `boa` manuell unter Angabe des Demo-Verzeichnisses zu starten. Hierzu ist zunächst die im Abschnitt 5.5.1 beschriebenen Anmeldung an der Kommando-Shell des ECUcore-5208 durchzuführen. Anschließend ist im Telnet- bzw. Terminal-Fenster folgendes Kommando einzugeben:

```
boa -c /home/http &
```

Bild 20 verdeutlicht den Start des HTTP-Servers `boa` am Beispiel der im Auslieferungszustand das ECUcore-5208 enthaltenen, vorgefertigten Demo-Konfiguration.



Bild 20: Starten des HTTP-Servers `boa`

Zum Abrufen der durch den HTTP-Server bereitgestellten Seiten ist in der Adressleiste des WEB-Browser der Präfix `http://` gefolgt von der im Abschnitt 5.4 festgelegte IP-Adresse für das ECUcore-5208 anzugeben, z.B. `http://192.168.10.171`. Bild 21 verdeutlicht die Anzeige von HTML-Seiten des ECUcore-5208 im WEB-Browser.



Bild 21: Anzeige von HTML-Seiten des ECUcore-5208 im WEB-Browser

Hinweis: Durch Einfügen des Startaufrufes für den HTTP-Server (z.B. `"start-stop-daemon -S -b -x boa -- -c /home/http"`) in das Startskript `"/home/etc/autostart"` lässt sich der Aufruf des HTTP-Servers beim Booten des ECUcore-5208 automatisieren (siehe dazu Abschnitt 5.2.2).

5.15 Update des uClinux-Images

Der Update des uClinux-Images erfolgt via TFTP (Trivial **FTP**) innerhalb des Linux-Bootloaders `"CoLilo"`. Auf dem Entwicklungs-PC ist hierfür ein entsprechender TFTP-Server notwendig. Standardmäßig ist ein entsprechend konfigurierter TFTP-Server bereits im VMware-Image des Linux-Entwicklungssystems enthalten.

Um alternativ ein Update des uClinux-Images von einem Windows-PC durchführen zu können, bietet sich beispielsweise die Freeware `"TFTPD32"` an (siehe Abschnitt 5.1). Das Windows-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:\ECUcore-5208"`). In diesem Verzeichnis muss sich das uClinux-Image für das ECUcore-5208 befinden (`image.bin`).

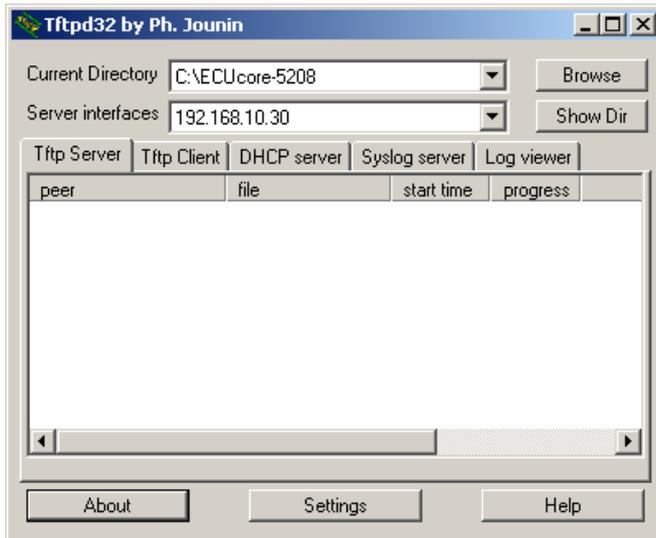


Bild 22: TFTP-Server für Windows "TFTPD32"

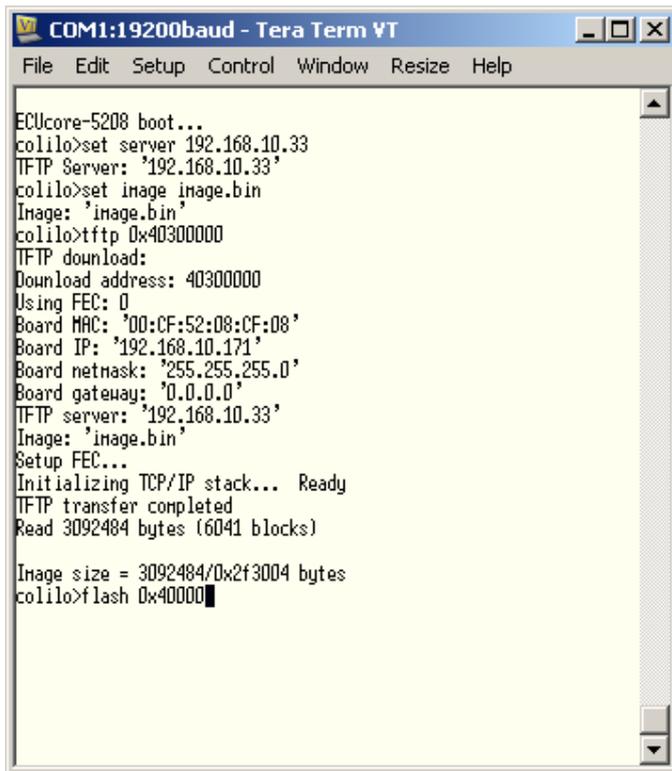
Voraussetzung für den TFTP-Download des uClinux-Images **ist die abgeschlossene Ethernet-Konfiguration** des ECUcore-5208 **gemäß Abschnitt 5.4**. Für das Update des uClinux-Images ist neben der Ethernet-Verbindung noch eine serielle Verbindung zum ECUcore-5208 erforderlich. Hier gelten ebenfalls die im Abschnitt 5.2.1 beschriebenen Einstellungen für das Terminalprogramm (19200 Baud, 8 Datenbits, 1 Stopbit, keine Parität, keine Flusskontrolle).

Ein Update des uClinux-Images auf dem ECUcore-5208 ist nur möglich, wenn uClinux noch nicht gestartet ist. Daher ist vor dem Update der uClinux-Autostart zu unterbinden und stattdessen zum "CoLilo" Kommandoprompt zu wechseln. Die dazu notwendigen Schritte beschreibt Abschnitt 5.2.1.

Nach Reset (z.B. Taster S403 am Developmentboard) meldet sich der "CoLilo"-Kommandoprompt. Hier sind zum Update des uClinux-Images die im Folgenden beschriebenen Kommandos in der aufgeführten Reihenfolge einzugeben:

Tabelle 8: Kommando-Sequenz zum Update des uClinux-Images auf dem ECUcore-5208

Kommando	Bedeutung
<code>set server <host_ip_addr></code>	Setzen der IP-Adresse des TFTP-Servers. Bei Verwendung des TFTP-Servers im VMware-Image des Linux-Entwicklungssystems ist die zu verwendende IP-Adresse mit Hilfe des Linux-Kommandos <code>ifconfig</code> zu ermitteln (siehe Abschnitt 6.5), bei Verwendung des Windows-Programms <code>"TFTPD32"</code> wird diese auf dem PC im Feld "Server Interface" angezeigt
<code>set image image.bin</code>	Setzen der Dateinamen für das zu ladende uClinux-Image
<code>tftp 0x40300000</code>	Download des uClinux-Images vom PC auf das ECUcore-5208
<code>flash 0x40000</code>	Speichern des uClinux-Images im Flash des ECUcore-5208
<code>set kfl 1</code>	Das im Flash des ECUcore-5208 gespeicherte uClinux-Image als gültig deklarieren
<code>set auto 1</code>	Autostart des uClinux-Images nach Reset aktivieren
<code>config save</code>	Speichern der aktuellen Einstellungen im Flash



```

COM1:19200baud - Tera Term VT
File Edit Setup Control Window Resize Help
ECUcore-5208 boot...
colilo>set server 192.168.10.33
TFTP Server: '192.168.10.33'
colilo>set image image.bin
Image: 'image.bin'
colilo>tftp 0x40300000
TFTP download:
Download address: 40300000
Using FEC: 0
Board MAC: '00:CF:52:08:CF:08'
Board IP: '192.168.10.171'
Board netmask: '255.255.255.0'
Board gateway: '0.0.0.0'
TFTP server: '192.168.10.33'
Image: 'image.bin'
Setup FEC...
Initializing TCP/IP stack... Ready
TFTP transfer completed
Read 3092484 bytes (6041 blocks)

Image size = 3092484/0x2f3004 bytes
colilo>flash 0x400000

```

Bild 23: Download des uLinux-Images auf das ECUcore-5208

Nach Abschluss der Konfiguration sind gemäß Abschnitt 5.2.1 die Voraussetzungen für einen uLinux-Autostart wieder herzustellen.

Nach Reset (z.B. Taster S403 am Developmentboard) startet das ECUcore-5208 mit dem aktualisierten uLinux-Image.

Hinweis: Nach Abschluss der Konfiguration ist die serielle Verbindung zwischen PC und ECUcore-5208 nicht mehr erforderlich.

5.16 Update des Bootloaders "CoLilo"

Ein Update des Bootloaders "CoLilo" ist zwar prinzipiell möglich, birgt aber die Gefahr, dass nach einem missglückten Update oder nach dem Einspielen einer fehlerhaften Software das ECUcore-5208 nicht mehr startet. Aus diesem Grund sind auch die Flashsektoren, in denen der "CoLilo" abgelegt ist, gegen versehentliches Löschen geschützt.

Ein Update des Bootloaders "CoLilo" sollte daher nur in dringend begründeten Fällen vorgenommen werden. Bitte setzen Sie sich bei Bedarf dazu mit unserem Support in Verbindung:

support@systec-electronic.com

6 VMware-Image mit Linux-Entwicklungssystem

6.1 Überblick

Das ECUcore-5208 wird mit einem vorinstallierten uClinux ausgeliefert. Anwendungen, die auf diesem Modul ausgeführt werden sollen, sind dementsprechend als Linux-Programme zu entwickeln. Das Kit beinhaltet ein komplett eingerichtetes Linux-Entwicklungssystem in Form eines VMware-Images und ermöglicht so einen leichten Einstieg in die Softwareentwicklung für das ECUcore-5208. Das VMware-Image kann dabei unverändert unter verschiedenen Host-Systemen benutzt werden. Geeignete Nachschlagewerke zur Linux-Programmierung sind in Tabelle 1 im Abschnitt 2 aufgeführt.

Das im VMware-Image des Linux-Entwicklungssystems beinhaltet folgende Softwarekomponenten:

- GNU-Crosscompiler Toolchain für Freescale MCF5208-Prozessoren
- uClinux-Quelltext für das ECUcore-5208
- Eclipse (grafische IDE zur Vereinfachung der Softwareentwicklung)
- Samba-Server (ermöglicht Zugriff "von außen" über Windows-Netzwerkumgebung)
- TFTP-Server (ermöglicht Download des uClinux-Images für das ECUcore-5208 vom Entwicklungs-PC)
- NFS-Server (ermöglicht Einbindung des Entwicklungs-PCs in das lokale Dateisystem des ECUcore-5208)
- FTP-Server (ermöglicht Dateiübertragung zwischen ECUcore-5208 und Entwicklungs-PC)
- Telnet-Server (ermöglicht Benutzung einer Linux-Konsole "von außen")

6.2 Installation des Linux-VMware-Images

Das Development Kit ECUcore-5208 beinhaltet auf der DVD "SO-1096" sowohl das VMware-Image des Linux-Entwicklungssystems für das ECUcore-5208 als auch den "VMware Player" für Windows. Der "VMware Player" ist eine kostenlose Software zur Desktop-Virtualisierung, mit deren Hilfe das VMware-Image des Linux-Entwicklungssystems auf einem Windows- oder Linux-PC ausgeführt werden kann. Bei Bedarf kann die jeweils aktuelle Version des "VMware Players" bzw. Player für andere Hostsysteme auch direkt von der Homepage des Herstellers unter <http://www.vmware.com> herunter geladen werden. Zur Installation des VMware Players ist das entsprechende Setup-Programm auszuführen.

Hinweis: Bei der Installation des "VMware Players" sollte unbedingt die Standardeinstellung "Bridged Mode" für das Ethernet-Interface beibehalten werden, da andernfalls unter Umständen keine Kommunikation zum ECUcore-5208 möglich sein kann.

Das VMware-Image ist auf der DVD als selbstentpackendes Archiv "**SO-1096.exe**" enthalten. Beim Starten von "**SO-1096.exe**" werden sämtliche zum VMware-Image gehörenden Dateien auf die lokale Festplatte entpackt. Das dekomprimierte Image beansprucht ca. 10 GByte.

6.3 Starten des Linux VMware-Images

Auf dem Host-PC ist zunächst der "VMware Player" zu starten. Im Programmfenster des Players ist dann mit Hilfe des "Open"-Symbols die Datei "Xubuntu-ECUcore5208.vmx" zu öffnen (siehe Bild 24).



Bild 24: Programmfenster des VMware Players mit "Open"-Symbol

VMware speichert beim Ausführen eines Images den "Fingerabdruck" des Hostrechners in Form einer UUID in einer *.vmx Datei. Beim Starten des Linux VMware-Images auf einem anderen PC erscheint der in Bild 25 dargestellte Dialog. Wenn dasselbe Linux-Image nicht zeitgleich auf einem weiteren Rechner im selben Netzwerk ausgeführt wird, dann kann die Option **"I moved it"** gewählt werden. Dabei bleibt die bisher verwendete UUID und damit auch die MAC-Adresse der virtuellen Netzwerkkarte im Gastsystem erhalten.

Bei der Voreinstellung **"I copied it"** erzeugt VMware eine neue UUID und damit eine neue MAC-Adresse für das Gastsystem. Falls das VMware-Image auf mehreren Rechnern benutzt werden soll, ist diese Option dringend anzuraten.

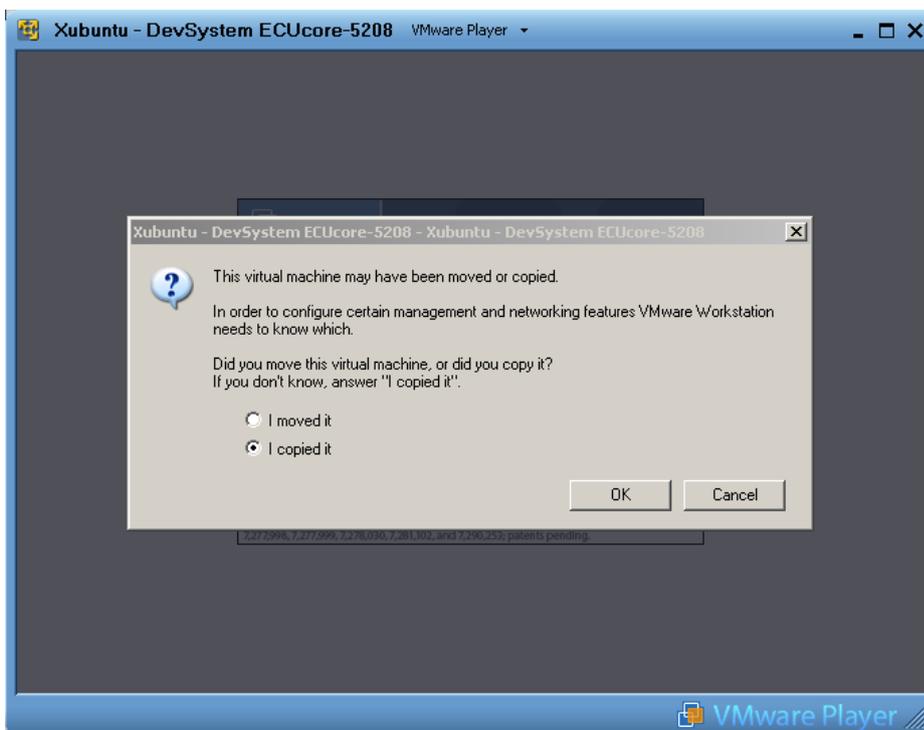


Bild 25: VMware Auswahldialog für eine neue UUID

Das Linux-Image ist so konfiguriert, dass es via DHCP-Client dynamisch eine IP-Adresse anfordert.

Bild 26 zeigt den Desktop des Linux-Entwicklungssystems nach dem Starten des Linux-Images im "VMware Player".

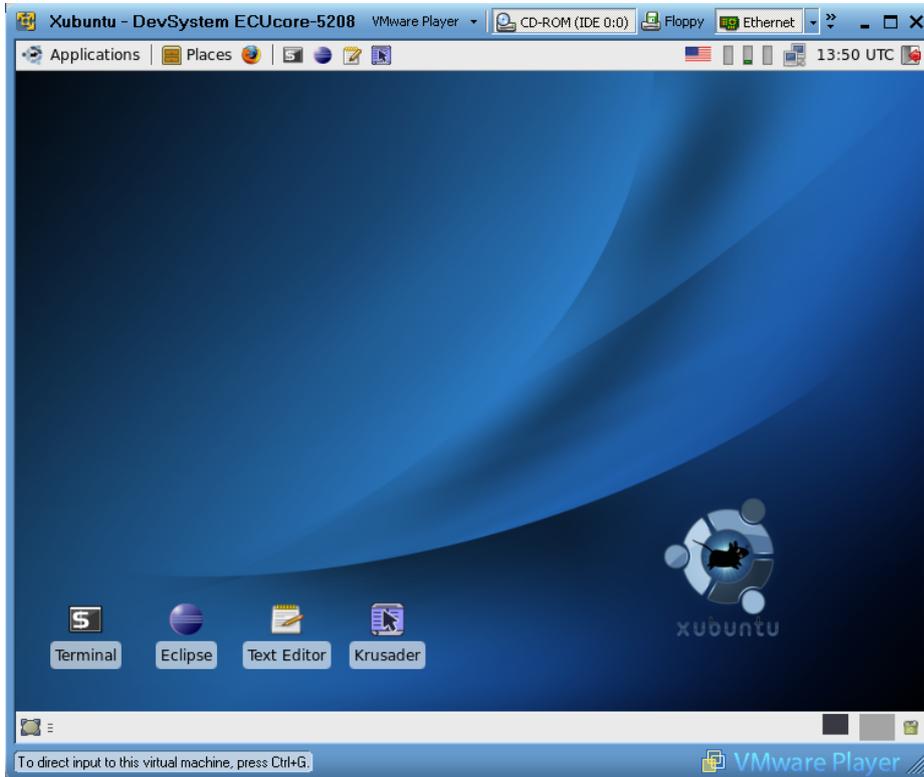


Bild 26: Desktop des Linux-Entwicklungssystems

6.4 Benutzerkonten zur Anmeldung am Linux-Entwicklungssystem

Tabelle 9 listet die zur Anmeldung am Linux-Entwicklungssystem vordefinierten Benutzerkonten auf.

Tabelle 9: Vordefinierte Benutzerkonten des Linux-Entwicklungssystems

Anmeldung	Benutzerdaten	Bemerkung
lokale Konsole / Terminal (normale Nutzerrechte)	User: vmware Passwort: vmware	vordefiniertes Benutzerkonto innerhalb des Linux-Entwicklungssystems
Administratorrechte	Kommando: sudo Passwort: vmware	Das verwendete Linux-Entwicklungssystem unterstützt keine explizite Anmeldung als "root". Bei Bedarf ist dem jeweils mit Administratorrechten auszuführenden Befehl das Linux-Kommando <i>sudo</i> voranzustellen, z.B. <i>"sudo cat /etc/shadow"</i>
Windows-Netzwerkumgebung	Gruppe: Workgroup Computer: vm-xubuntu User: vmware Passwort: vmware	vordefiniertes Benutzerkonto für Zugriff auf das Linux-Entwicklungssystem über die Windows-Netzwerkumgebung (Samba)

Telnet-Zugang	User: vmware Passwort: vmware	vordefiniertes Benutzerkonto zur Anmeldung an das Linux-Entwicklungssystem über einen Telnet-Client (z.B. Telnet-Client unter Windows)
---------------	----------------------------------	--

6.5 Bestimmung der IP-Adresse des Linux-Images

Um die dem Linux-Entwicklungssystem per DHCP zugewiesene IP-Adresse zu ermitteln, ist zunächst innerhalb von Linux ein Konsolenfenster zu starten (Symbol "Terminal"). Nach Eingabe des Befehls **"/sbin/ifconfig"** wird dann u. a. die IP-Adresse des Linux-Images angezeigt (im Bild 27 farblich markiert).

```

vmware@vm-xubuntu:~$ /sbin/ifconfig
eth2      Link encap:Ethernet  HWaddr 00:0c:29:87:f3:5c
          inet addr:192.168.10.33  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe87:f35c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3259 errors:0 dropped:0 overruns:0 frame:0
          TX packets:547 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:408205 (408.2 KB)  TX bytes:173328 (173.3 KB)
          Interrupt:18 Base address:0x10a4

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:78 errors:0 dropped:0 overruns:0 frame:0
          TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5776 (5.7 KB)  TX bytes:5776 (5.7 KB)

vmware@vm-xubuntu:~$

```

Bild 27: IP-Adresse des Linux-Entwicklungssystems ermitteln

6.6 Zugriff auf das Linux-Entwicklungssystem von einem Windows-PC

6.6.1 Zugriff über die Windows-Netzwerkumgebung

Über den im VMware-Image installierten Samba-Server ist der Zugriff auf Dateien im Linux-Entwicklungssystem über die Windows-Netzwerkumgebung möglich. Das erlaubt das Erstellen und Bearbeiten von Quelltexten mit Hilfe eines beliebigen Windows-Editors. Das Dateisystem des Linux-Entwicklungssystem ist in der Windows-Netzwerkumgebung im Zweig **"Workgroup"** unter dem Rechnernamen **"vm-xubuntu"** zugänglich (siehe auch Tabelle 9 im Abschnitt 6.4).

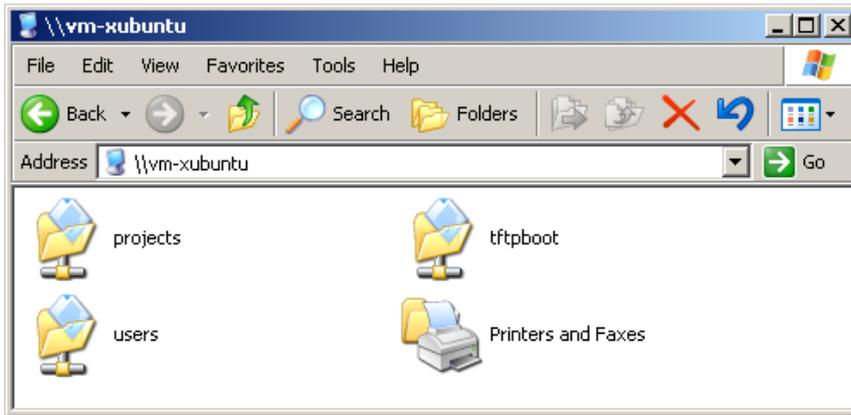


Bild 28: Linux-Entwicklungssystem in der Windows-Netzwerkumgebung

Nach einem Doppelklick auf das Symbol "users" (siehe Bild 28) ist die Anmeldung als **Nutzer "vmware"** mit dem **Passwort "vmware"** möglich (siehe Bild 29). Abschließend besteht Zugriff auf den Pfad "\\vm-xubuntu\users\".



Bild 29: Anmeldung an "vm-xubuntu"

Alternativ (z.B. bei Problemen wegen langer Timeouts beim Durchsuchen der Windows-Netzwerkumgebung oder generell beim Auffinden des virtuellen Rechners) kann das VMware-Image des Linux-Entwicklungssystem auch direkt über den Windows-Befehl "net use" an einen Laufwerksbuchstaben gebunden werden. Dies kann entweder über den symbolischen Namen oder auch direkt über die IP-Adresse des Linux-Entwicklungssystems erfolgen. Für letzteren Fall ist zunächst wie ist im Abschnitt 6.5 beschrieben die IP-Adresse des Linux-Entwicklungssystems zu ermitteln ist. Der Befehl "net use" hat folgende Syntax:

```
net use <local_device> <\\computername\sharename> /user:<username> [options]
```

Um beispielsweise das VMware-Image mit dem Linux-Entwicklungssystem an den lokalen Laufwerksbuchstaben "X:" zu binden, ist der Befehl "net use" wie folgt anzuwenden:

```
net use x: \\vm-xubuntu\users /user:vmware /persistent:NO
```

Alternativ kann statt des symbolischen Namens auch direkt die IP-Adresse des Linux-Entwicklungssystems angegeben werden, z.B.:

```
net use x: \\192.168.10.33\users /user:vmware /persistent:NO
```

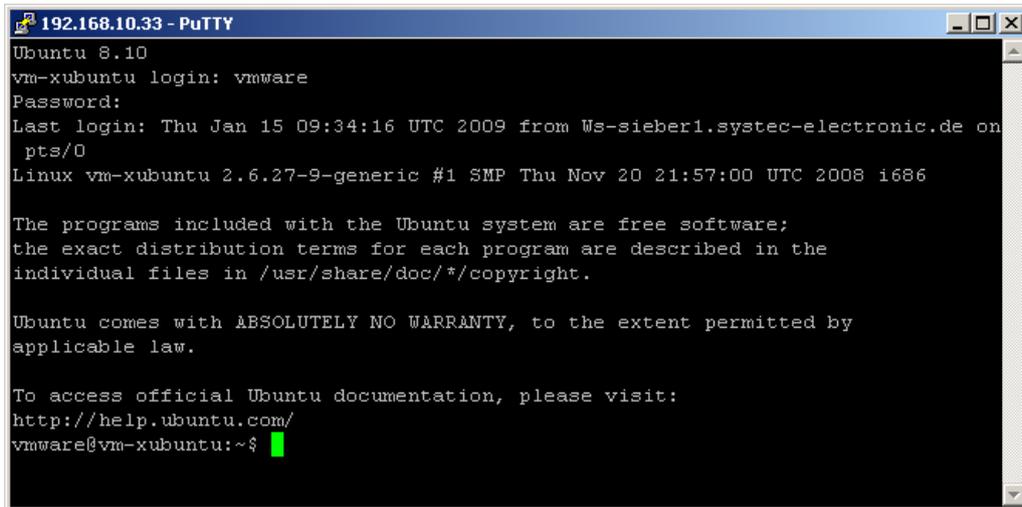
6.6.2 Zugriff über Telnet-Client

Über den im VMware-Image installierten Telnet-Server ist der Zugriff auf eine Konsole des Linux-Entwicklungssystem auch über einen entsprechenden Telnet-Client unter Windows möglich. Das

erlaubt den Aufruf Kommandozeilen-Programmen wie beispielsweise "make" zum Übersetzen von Anwenderprojekten über eine Windows-Oberfläche.

Der Zugriff mittels Telnet-Client erfolgt direkt über die IP-Adresse des Linux-Entwicklungssystems. Abschnitt 6.5 beschreibt, wie die IP-Adresse des Linux-Entwicklungssystems zu ermitteln ist. Um sich über PuTTY am Linux-Entwicklungssystem anzumelden, ist wie in Abschnitt 5.5.3 die IP-Adresse der VMWare anzugeben.

Innerhalb des Telnet-Fensters ist die Anmeldung als **Nutzer "vmware"** mit dem **Passwort "vmware"** möglich (siehe auch Tabelle 9 im Abschnitt 6.4):



```
192.168.10.33 - PuTTY
Ubuntu 8.10
vm-xubuntu login: vmware
Password:
Last login: Thu Jan 15 09:34:16 UTC 2009 from Ws-sieber1.systemec-electronic.de on
pts/0
Linux vm-xubuntu 2.6.27-9-generic #1 SMP Thu Nov 20 21:57:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
vmware@vm-xubuntu:~$
```

Bild 30: Zugriff über Telnet-Client auf das Linux-Entwicklungssystem

Bild 30 verdeutlicht am Beispiel die Anmeldung am Linux-Entwicklungssystem mit Hilfe von PuTTY.

6.7 Persönliche Konfiguration und Aktualisierung des Linux-VMware-Images

6.7.1 Anpassung von Tastaturlayout und Zeitzone

Standardmäßig ist das Linux-VMware-Image auf US-Tastaturlayout und UTC-Zeitzone eingestellt. Über das Ländersymbol in der Taskleiste ist der bequeme Wechsel zu einem alternativen, bereits vorinstallierten Tastaturlayout möglich (siehe Bild 31).

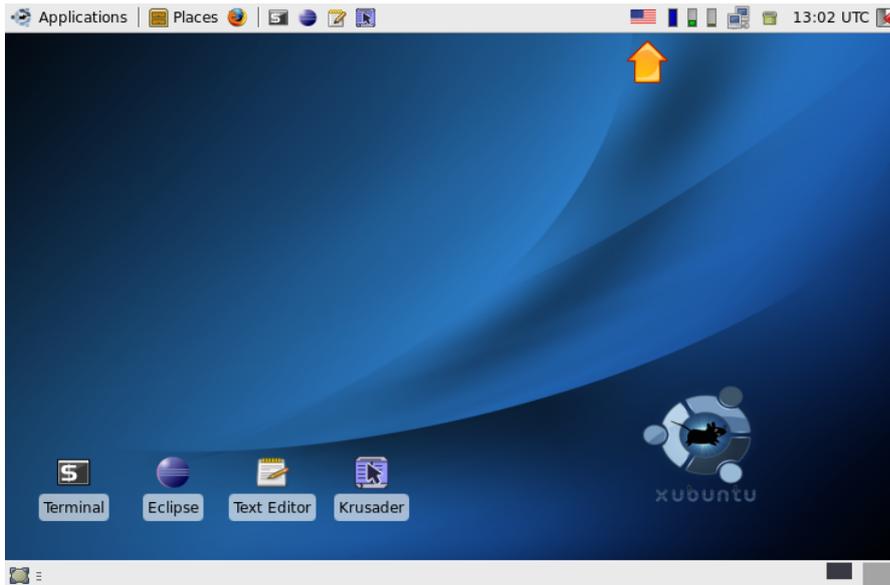


Bild 31: Ländersymbol für Wechsel des Tastaturlayouts

Um ein alternatives **Tastaturlayout permanent auszuwählen**, ist das Ländersymbol in der Taskleiste (siehe Bild 31) mit der **rechten Maustaste** anzuklicken und aus dem Popup-Menü der Eintrag "Properties" aufzurufen. Dabei öffnet sich der Dialog "Configure Keyboard Layout Switcher", hier kann das gewünschte Layout als "Default Layout" festgelegt werden (siehe Bild 32).

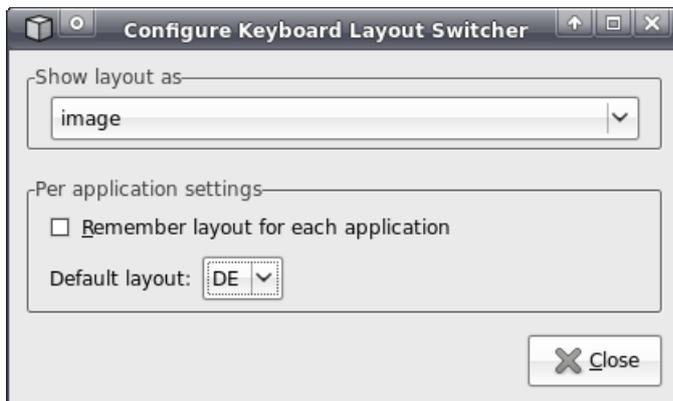


Bild 32: Tastaturlayout permanent auszuwählen

Das **Hinzufügen weiterer Tastaturlayouts** erfolgt mit Hilfe des "Xfce Settings Manager", der direkt über das Startmenü aufrufbar ist: "Applications -> Settings -> Settings Manager" (siehe Bild 33).

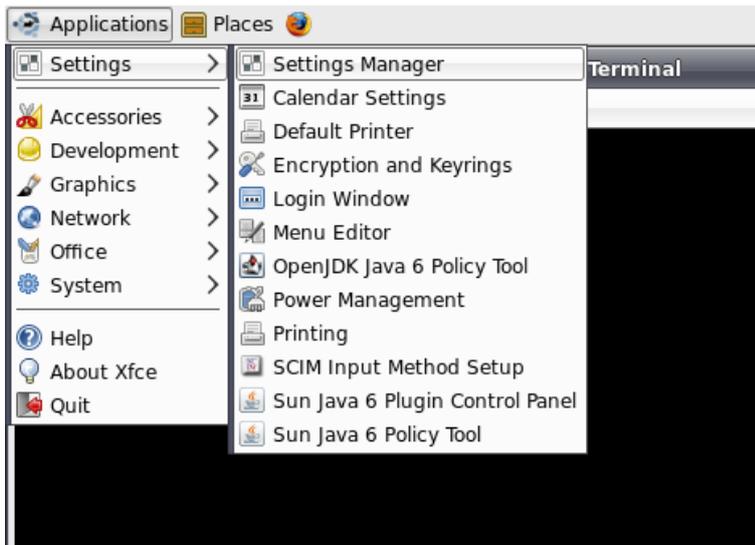


Bild 33: Aufruf des "Xfce Settings Manager" über das Startmenü

Innerhalb des "Xfce Settings Manager" können über die Option "Keyboard" und deren Unterpunkt "Layouts" weitere Tastaturlayouts hinzugefügt bzw. auch wieder entfernt werden (siehe Bild 34).

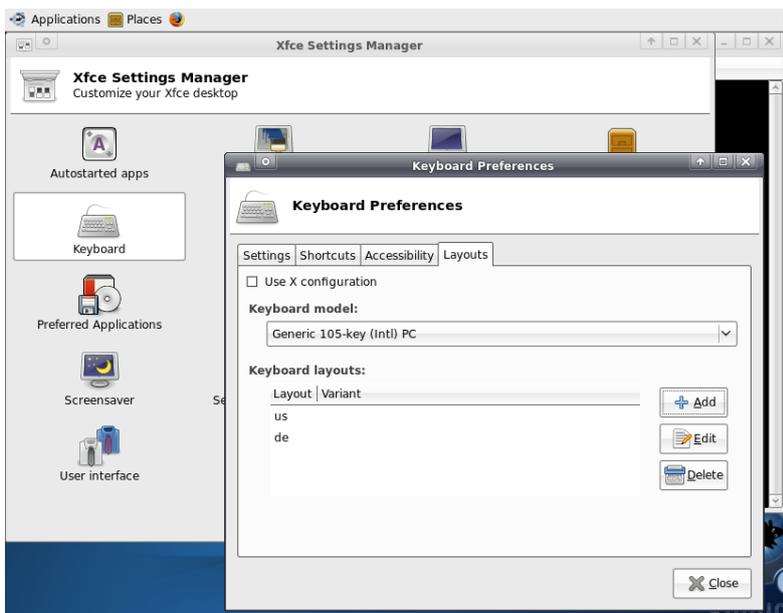


Bild 34: Hinzufügen von Tastaturlayouts im "Xfce Settings Manager"

Die **Einstellung der Zeitzone** erfolgt über ein Control-Pannel der Systemkonfiguration, das ebenfalls direkt über das Startmenü aufrufbar ist: "Applications -> System -> Time and Date" (siehe Bild 35). Da das Ändern der Zeitzone eine administrative Tätigkeit ist, muss der Dialog zunächst durch Betätigen der Schaltfläche "Unlock" freigegeben werden (ebenfalls siehe Bild 35). Dazu wird analog zum Konsolen-Kommando "sudo" das Administratorpasswort abgefragt. Standardmäßig ist hierfür als **Passwort "vmware"** einzugeben (siehe auch Tabelle 9 im Abschnitt 6.4). Anschließend kann über den Punkt "Time zone" die Hauptstadt und damit die jeweilige Zeitzone ausgewählt werden.

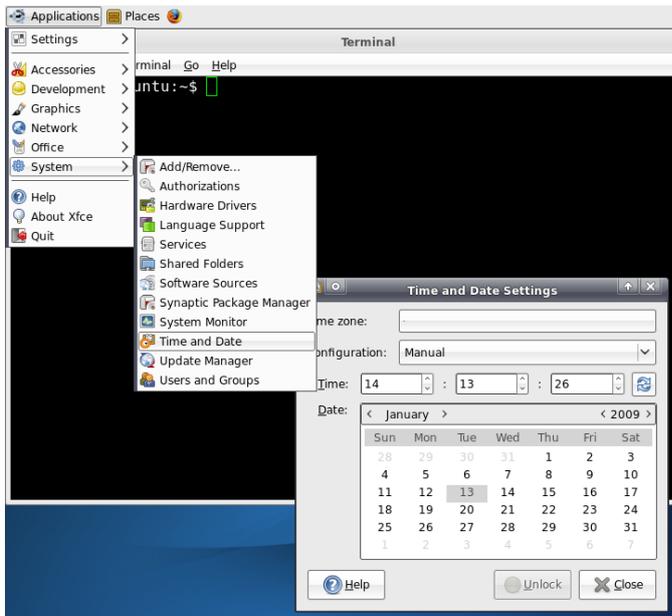


Bild 35: Anpassen der Zeitzone

6.7.2 Anpassen der Desktopgröße

Das Linux-VMware-Image ist in der Lage, die Desktopgröße automatisch an die Fenstergröße des VMware-Players anzupassen. Da sich der VMware-Player wie eine normale Windows-Applikation verhält, muss also nur der Fensterrahmen des VMware-Players mit der Maus auf die gewünschte Größe gezogen werden, um die Desktopgröße des Linux-VMware-Images zu ändern. Die maximal nutzbare Fenstergröße ist in der Konfigurationsdatei "**SO-1096Xubuntu-ECUcore52008.vmx**" des VMware-Players definiert und kann hier bei Bedarf verändert werden:

```
##### display #####
...
svga.maxWidth = "1024"
svga.maxHeight = "768"
...
```

6.7.3 Festlegen einer statischen IP-Adresse für das Linux-VMware-Image

Im Linux-VMware-Image ist standardmäßig die dynamische Konfiguration der IP-Adresse über DHCP aktiviert. Damit kann das Linux-VMware-Image in den meisten Netzwerkeumgebungen ad-hoc benutzt werden, ohne dass zuvor eine manuelle Einstellung von Parametern notwendig ist. In Netzwerken, in denen kein DHCP-Server vorhanden ist, muss dagegen vom Anwender eine statische IP-Adresse für das Linux-VMware-Image festgelegt werden. Andernfalls ist keine Ethernet-basierte Kommunikation mit dem ECUcore-5208 möglich.

Um eine statische IP-Adresse für das Linux-VMware-Image festzulegen, ist das Symbol des "*Network Manager*" in der Taskleiste (siehe Bild 36) mit der **rechten Maustaste** anzuklicken und aus dem Popup-Menü der Eintrag "*Edit Connections*" aufzurufen. Dabei öffnet sich der Dialog "*Network Connections*" (siehe Bild 37).

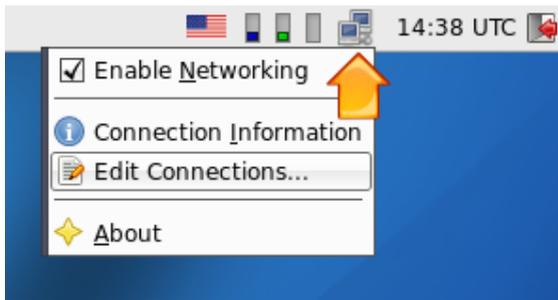


Bild 36: "Network Manager" zur Konfiguration der Ethernet-Schnittstelle

Im Dialog "Network Connections" (siehe Bild 37) kann im Tabsheet "Wired" mit Hilfe der Schaltfläche "Add" eine neue Netzwerkverbindung angelegt werden. Dabei öffnet sich der Dialog "Edit Wired Connection" (siehe Bild 38).



Bild 37: Hinzufügen einer Netzwerkverbindung

Im Dialog "Edit Wired Connection" (siehe Bild 38) ist zunächst im Auswahlfeld "Method" die Voreinstellung "Automatic (DHCP)" auf die alternative Option "Manual" zu ändern. Anschließend können im Tabsheet "IPv4 Settings" die Einstellungen für IP-Adresse, Netzwerkmaske, Gateway, DNS-Server usw. vorgenommen werden.

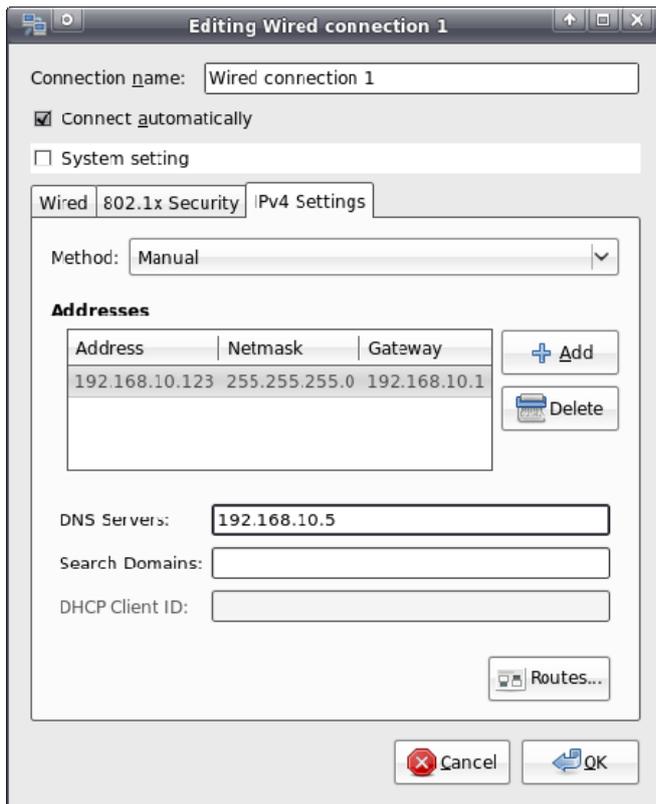


Bild 38: Konfiguration der Netzwerkverbindung

Nach Abschluss der Konfiguration und Schließen aller Dialoge ist wiederum das Symbol des "Network Manager" in der Taskleiste anzuklicken, jedoch diesmal mit der **linken Maustaste** (siehe Bild 39). Hier ist die neu angelegte Netzwerkverbindung mit der statischen IP-Adresse entsprechend als aktive Verbindung auszuwählen.

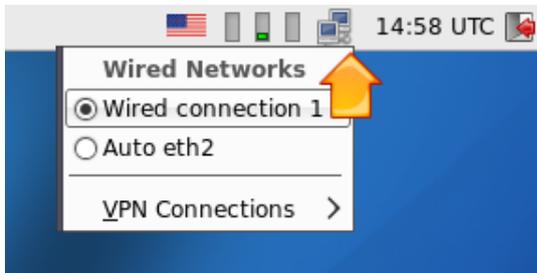


Bild 39: Wechsel der Netzwerkconfiguration im "Network Manager"

6.7.4 Systemaktualisierung des Linux-VMware-Images

Im Linux-VMware-Image ist die automatische Aktualisierung der Paketlisten **deaktiviert**. Damit ist sichergestellt, dass ein definierter Systemstand bestehen bleibt. Es erfolgt jedoch auch keine Information an den Benutzer, falls aktualisierte Pakete verfügbar sind, die eventuelle Sicherheitslücken schließen.

Es ist dennoch möglich, die automatische Aktualisierung der Paketlisten wieder zu aktivieren bzw. das Aktualisieren manuell zu veranlassen. Mögliche Programme dafür sind "Synaptic Package Manager" (aus dem Menü "System") und das Konsolenprogramm "aptitude". Nach der Aktualisierung der Paketlisten ist es mit diesen Programmen auch möglich, neue Paketversionen zu installieren.

Es wird jedoch ausdrücklich darauf hingewiesen, dass nicht gewährleistet werden kann, dass nach einem Update das Linux-Entwicklungssystem für das ECUcore-5208 weiterhin vollständig funktionsfähig bleibt. Es wird daher dringend empfohlen, vor einem Update unbedingt eine Backup-Kopie des Linux-Entwicklungssystems anzufertigen.

6.7.5 Ändern des Computer-Namens in der Windows-Netzwerkumgebung

Das Linux-VMware-Image benutzt standardmäßig den Computer-Namen "*vm-xubuntu*", der so auch in der Windows-Netzwerkumgebung verwendet wird (siehe auch Tabelle 9 im Abschnitt 6.4). Da der Zugriff auf einen Computer im Windows-Netzwerk über dessen Namen gesteuert wird, sind beim parallelen Einsatz des Linux-VMware-Images auf mehreren Rechnern die Computer-Namen eindeutig anzupassen. Damit wird eine Mehrfachverwendung desselben Namens verhindert, was andernfalls zu Kollisionen und Zugriffsfehlern führen würde.

Der Computer-Namen wird über die Datei "*/etc/hostname*" definiert. Um diese zu modifizieren, ist der Befehl "***sudo gedit /etc/hostname***" einzugeben. Der geänderte Name wird dann nach einem Neustart übernommen. Eine sofortige Änderung des Namens bewirkt das Kommando "*hostname*". Dazu wird der neu zu setzende Computer-Name als Parameter angegeben, z.B. "***sudo hostname vm-xubuntu-2***". Die mit diesem Befehl vorgenommene Änderung wirkt jedoch im Gegensatz zur Modifikation der Datei "*/etc/hostname*" nur temporär bis zum nächsten Neustart.

6.7.6 Schrumpfen des VMware-Images

Um das VMware-Image auf die notwendige Minimalgröße zu schrumpfen, ist die VMware-Toolbox durch eingabe des Befehls "***sudo vmware-toolbox***" aufzurufen. Über das Tabsheet "Shrink" kann das VMware-Image auf seine notwendige Minimalgröße geschrumpft werden.

7 Softwareentwicklung für das ECUcore-5208

7.1 Notwendige Schritte nach dem 1. Start

Nach dem ersten Start des VMware-Images muss (**innerhalb** des Images) im Verzeichnis `"/projects/ECUcore-5208/uClinux-dist"` einmalig `"make"` aufgerufen werden.

7.2 Softwarestruktur für das ECUcore-5208

Im VMware-Image des Linux-Entwicklungssystems sind sämtliche Komponenten, die zur Entwicklung von Software für das ECUcore-5208 benötigt werden, im Pfad `"/projects"` abgelegt (respektive `"\\vm-xubuntu\users\projects\"` in der Windows-Netzwerkumgebung). Bild 40 veranschaulicht die Verzeichnisstruktur.

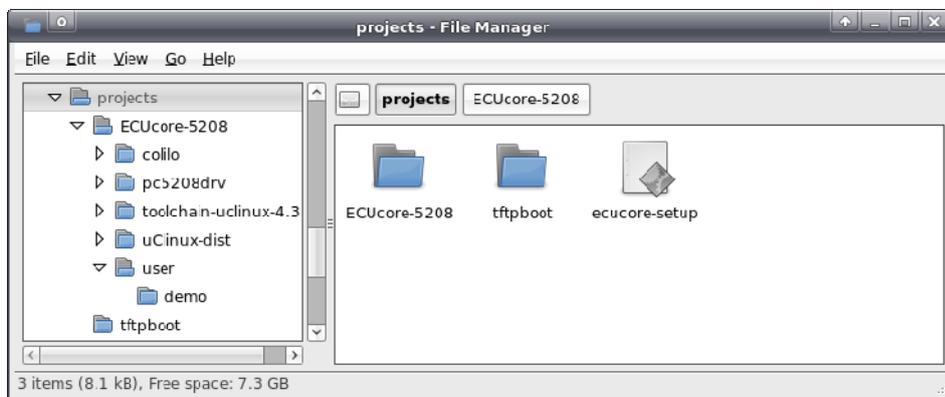


Bild 40: Struktur des Verzeichnisses `"/projects"` im Linux-Entwicklungssystem

<code>/projects</code>	
<code>+ - ECUcore-5208</code>	
<code>+ - colilo</code>	Sourcecode des Bootloaders "CoLilo"
<code>+ - driver</code>	Treiber-Bibliotheken für das ECUcore-5208
<code>+ - pc5208drv</code>	I/O-Treiber zur Einbindung in eigene Anwenderprojekte (incl. Sourcecode und Testapplikation)
<code>+ - candrv</code>	CAN-Treiber zur Einbindung in eigene Anwenderprojekte
<code>+ - toolchain-uclinux-4.2</code>	uClinux-Toolchain für Coldfire 5208
<code>+ - uClinux-dist</code>	Projektpfad zur Erstellung des uClinux-Images für das ECUcore-5208 (uClinux-Kernel und User-Programme)
<code>+ - linux-2.6.x</code>	Sourcecode des uClinux-Kernels für das ECUcore-5208
<code>+ - user</code>	Sourcecode der User-Programme für das ECUcore-5208
<code>... </code>	sonstige Komponenten der uClinux-Distribution
<code>+ - user</code>	Pfad zur Ablage eigener Anwenderprojekte
<code>+ - demo</code>	Referenz- und Demoapplikation für das ECUcore-5208
<code>+ - helloCAN</code>	Referenz- und CAN-Demoprojekt für das ECUcore-5208
<code>+ - tftpboot</code>	wird per NFS in das lokale Filesystem des ECUcore-5208 eingebunden

|
+- `ecucore-setup`

Shell-Script zum Setzen von Umgebungsparametern

Im Pfad `"/projects/ECUcore-5208/uClinux-dist"` befinden sich die gesamten uClinux-Sourcen für das ECUcore-5208.

Das Verzeichnis `"/projects/ECUcore-5208/driver/pc5208drv"` beinhaltet den Sourcecode des I/O-Treibers für das ECUcore-5208 (incl. Testapplikation), als auch **Headerfiles und fertig generierte Bibliotheken des I/O-Treibers** zur Einbindung in eigene Anwenderprojekte (siehe Abschnitt 7.4.1).

Das Verzeichnis `"/projects/ECUcore-5208/driver/candrv"` beinhaltet **Headerfiles und fertig generierte Bibliotheken des CAN-Treibers** zur Einbindung in eigene Anwenderprojekte (siehe Abschnitt 7.5.1).

Im Verzeichnis `"/projects/ECUcore-5208/user/demo"` ist ein Demoprogramm enthalten, das zum einen den Zugriff auf die Ein- und Ausgänge des ECUcore-5208 beschreibt (Details siehe Abschnitt 7.4.1) und zum anderen als Template für eigene Projekte verwendet werden sollte. Alle weiteren Beschreibungen zur Softwareentwicklung beziehen sich im Folgenden auf diese Demoprojekt (speziell im Abschnitt 7.7).

Im Verzeichnis `"/projects/ECUcore-5208/user/helloCAN"` ist ein Demoprogramm enthalten, das zum einen den Zugriff auf das CAN-Interface des ECUcore-5208 beschreibt und zum anderen als Template für eigene Projekte verwendet werden sollte.

Der Pfad `"/projects/tftpboot"` ist das Wurzelverzeichnis für die Einbindung des Linux-Entwicklungssystems in das lokale Filesystem des ECUcore-5208 per NFS (siehe Abschnitt 7.6.1).

Das Shell-Script `"/projects/ecucore-setup"` dient zum Setzen der erforderlichen Umgebungsparameter die zur Ausführung des Build-Systems benötigt werden (siehe Abschnitt 7.3). Dieses Shell-Skript wird automatisch beim Öffnen einer Konsole ("Terminal") über das File `".bashrc"` ausgeführt, ebenso beim Starten der grafischen IDE "Eclipse" über das entsprechende Desktop-Symbol.

7.3 Makefile und Umgebungsvariablen zum Erstellen von Projekten

Die Erstellung von Anwenderprogrammen für das ECUcore-5208 erfordert die Verwendung der GNU-Crosscompiler Toolchain für Freescale MCF52xx-Prozessoren. Diese ist im VMware-Image des Linux-Entwicklungssystems bereits vollständig installiert und konfiguriert. Von zentraler Bedeutung sind dabei die im Shell-Script `"/projects/ecucore-setup"` definierten Umgebungsvariablen:

```
export M68K_LINUX_BASE_PATH=/projects/ECUcore-5208
export M68K_LINUX_BSP_PATH=$M68K_LINUX_BASE_PATH/uClinux-dist
export M68K_PATH=$M68K_LINUX_BASE_PATH/toolchain-ucLinux-4.2/bin

#add toolchains to PATH
export PATH="$M68K_PATH:$PATH"
```

Als Ausgangspunkt für die Entwicklung eigener Anwenderprogramme sollte das im Pfad `"/projects/ECUcore-5208/user/demo"` enthaltene Template verwendet werden (bzw. `"\\vm-xubuntu\projects\ECUcore-5208\user\demo"` in der Windows-Netzwerkumgebung). Von besonderer Bedeutung sind hierbei die im Makefile (`demo/source/Makefile`) bereits definierten Variablen sowie die Verweise auf die GNU-Crosscompiler Toolchain für Freescale MCF52xx-Prozessoren, basierend auf den Definitionen in `"/projects/ecucore-setup"`:

```
CDEFS          = $(M68K_DBG_MODE)
CFLAGS        += -O0 -g -Wall -mcpu=5208 $(INCLUDES) $(CDEFS)
LDFLAGS       += -mcpu=5208 -Wl,-elf2flt

CROSS         = m68k-ucLinux-
CC            = $(CROSS)gcc
AR            = $(CROSS)ar
```

Innerhalb des Makefiles erfolgt der Aufruf der einzelnen Tools über entsprechende Makros wie beispielsweise \$(CC):

```
$(CC) $(CFLAGS) -o $(EXEC) hello.c
```

Ebenfalls im Makefile bereits vorbereitet ist das Umkopieren des erstellten Executables in das Verzeichnis `"/ftplibboot"` bzw. eines der darin enthaltenen Unterverzeichnisse. Hierdurch kann das ausführbare Programm später ohne weitere Zwischenschritte direkt auf dem ECUcore-5208 gestartet werden (siehe dazu auch Abschnitt 7.6.1).

7.4 I/O-Treiber für das ECUcore-5208

7.4.1 Einbindung des I/O-Treibers in eigene Anwenderprojekte

Das Verzeichnis `"/projects/ECUcore-5208/pc5208drv"` des Linux-Entwicklungssystems beinhaltet sowohl den Sourcecode des I/O-Treibers für das ECUcore-5208 (incl. Testapplikation), als auch Headerfiles und bereits fertig generierte Bibliotheken des Treibers zur Einbindung in eigene Anwenderprojekte.

Bild 41 verdeutlicht die Struktur des I/O-Treibers für das ECUcore-5208. Der Treiber untergliedert sich in ein Kernelspace-Modul (`pc5208drv.ko`), das die Zugriffe auf die Hardware realisiert, sowie in eine statische Userspace-Bibliothek (`pc5208drv.a`). Beide Komponenten, sowohl Kernelspace-Modul als auch Userspace-Bibliothek werden zur Ausführung von I/O-Zugriffen benötigt.

Das Erstellen und Verwenden von dynamischen Bibliotheken wird derzeit **nicht** unterstützt.

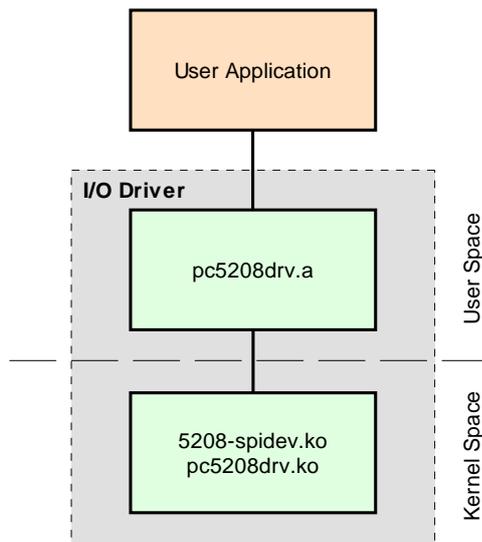


Bild 41: Struktur des I/O-Treibers für das ECUcore-5208

Aus dem Projektverzeichnis des I/O-Treibers sind folgende Dateien zur Einbindung in eigene Anwenderprojekte relevant:

`pc5208drv.h`: `/projects/ECUcore-5208/pc5208drv/include/pc5208drv.h`
Headerfile zur Beschreibung des Treiber-Interfaces
Dieses Headerfile ist mittels `#include` in den Quelltext (`*.c`) des Anwenderprojektes einzubinden.

`pc5208drv.a`: `/projects/ECUcore-5208/pc5208drv/lib/pc5208drv.a`
Userspace-Bibliothek des I/O-Treibers zum statischen Linken an die Anwenderapplikation (vergleichbar mit den statischen Standardbibliotheken der C-Entwicklungsumgebung).

Beim statischen Linken wird die Userspace-Bibliothek fest an die Anwenderapplikation gebunden und ist dadurch untrennbar mit dieser verbunden.

```
pc5208drv.ko: /projects/ECUcore-5208/pc5208drv/lib/pc5208drv.ko
5208-spidev.ko: /projects/ECUcore-5208/pc5208drv/lib/5208-spidev.ko
```

Kernelspace-Module des I/O-Treibers für Zugriffe auf Hardware
Diese Module sind vor dem Starten der Anwenderapplikation mit Hilfe des Linux-Kommandos "*insmod*" zu laden:

```
insmod 5208-spidev.ko
insmod pc5208drv.ko
```

Im Beispiel des im Abschnitt 7.7 vorgestellten Demoprogramms wird die Userspace-Bibliothek des I/O-Treibers fest an die Anwenderapplikation gebunden. Dazu ist die statische Bibliothek *pc5208drv.a* beim Aufruf des GCC in die Liste der zu linkenden Objekte aufzunehmen. Im Makefile des Demoprojektes wird dazu die Variable "*LIBS*=" entsprechend gesetzt, die dann wiederum Linkeraufruf (via GCC) übergeben wird:

```
LIBS = ../lib/pc5208drv.a

@$(CC) $(LDFLAGS) -o $@ $(OBJS) $(LIBS) $(LDLIBS)
```

Die vom I/O-Treiber zur Verfügung gestellten Funktionen sind im Headerfile "*pc5208drv.h*" aufgelistet, ihre Anwendung dokumentiert das Demoprogramm unter "*/projects/ECUcore-5208/user/demo*" (siehe auch Abschnitt 7.7).

Im Auslieferungszustand des ECUcore-5208 sind bereits fertig generierte und sofort per "*insmod*" ladbare Kernelmodule des I/O-Treibers im Verzeichnis "*/home/bin*" abgelegt (siehe Abschnitt 5.13):

```
insmod /home/bin/5208-spidev.ko
insmod /home/bin/pc5208drv.ko
```

Das zusätzliche Laden des Moduls *5208-spidev.ko* ist notwendig, damit der I/O-Treiber *pc5208drv.ko* auch die SPI-Geräte über den separaten Bus ansprechen kann. Diese Separierung ist notwendig, damit auch die schnellere CAN-Treibervariante genutzt werden kann (siehe Abschnitt 7.5.1).

Ein Beispiel für die praktische Anwendung des I/O-Treibers auf dem ECUcore-5208 zeigt das Demoprojekt unter "*/projects/ECUcore-5208/user/demo*" (siehe Abschnitte 7.4.2 und 7.7).

7.4.2 I/O-Treiber Demoprojekt

Das Demoprojekt für den I/O-Treiber ist im Verzeichnis "*/projects/ECUcore-5208/user/demo*" des Linux-Entwicklungssystems abgelegt. Es veranschaulicht den Zugriff auf die Ein- und Ausgänge des Moduls. Dazu bedient sich das Demoprogramm der Hilfe des unter "*/projects/ECUcore-5208/pc5208drv*" abgelegten I/O-Treibers. Das Demoprojekt wird im Abschnitt 7.7 als Referenzprojekt zur Softwareentwicklung für das ECUcore-5208 sehr detailliert beschrieben.

Vor dem Start des Demoprogramms ist zunächst der I/O-Treiber mit dem Befehl "*insmod*" explizit zu laden, anschließend kann das Demoprogramm selbst aufgerufen werden:

```
insmod 5208-spidev.ko
insmod pc5208drv.ko
./demo
```

Die Ausführung des Demoprojektes auf dem ECUcore-5208 verdeutlicht Bild 47 im Abschnitt 7.7.1. Das Demoprojekt kann entweder durch setzen des Run/Stop-Schalters in die Position "Res" oder durch Drücken von "Ctrl+C" beendet werden.

7.5 CAN-Treiber für das ECUcore-5208

7.5.1 Einbindung des CAN-Treibers in eigene Anwenderprojekte

Das Verzeichnis ***"/projects/ECUcore-5208/driver/candrv"*** des Linux-Entwicklungssystems befindet sich der CAN-Treiber für das ECUcore-5208. In dem Treiberverzeichnis befinden sich Headerdateien und eine bereits fertig generierte Bibliothek des CAN-Treibers für das ECUcore-5208 zur Einbindung in eigene Anwenderprojekte.

Die Struktur des CAN-Treibers ist identisch zu der im Abschnitt 7.4.1 beschriebenen Struktur des I/O-Treiber. So untergliedert sich der CAN-Treiber ebenfalls in ein Kernespace-Modul (*candrv.ko*), das die Zugriffe auf die CAN-Controller realisiert, sowie in eine Userspace-Bibliothek (*candrv.a*). Beide Komponenten, sowohl Kernespace-Modul als auch Userspace-Bibliothek werden zum Zugriff auf den CAN-Bus benötigt.

Aus dem Projektverzeichnis des CAN-Treibers sind folgende Dateien zur Einbindung in eigene Anwenderprojekte relevant:

cdrvinc.h: */projects/ECUcore-5208/driver/candrv/include/cdrvinc.h*
Headerdatei zur Beschreibung des Treiber-Interfaces
Diese Headerdatei ist mittels *#include* in den Quelltext (*.c) des Anwenderprojektes einzubinden. Diese Headerdatei inkludiert dann alle weiteren Headerdateien im Verzeichnis *"include"* in der richtigen Reihenfolge.

candrv.a: */projects/ECUcore-5208/driver/candrv/lib/candrv.a*
Userspace-Bibliothek des I/O-Treibers zum statischen Linken an die Anwenderapplikation.

candrv.ko: */projects/ECUcore-5208/driver/candrv/lib/candrv.ko*
Kernespace-Modul für den CAN-Treiber zum Zugriff auf den CAN-Controller.
Dieses Modul ist vor dem Starten der Anwenderapplikation mit Hilfe des Linux-Kommandos *"insmod"* zu laden:

```
insmod candrv.ko
```

Das Interface sowie die Nutzung des CAN-Treibers in eigenen Applikationen dokumentiert das CAN Treiber Software Manual (Manual-Nr.: L-1023). Ein Beispiel für die praktische Anwendung des CAN-Treibers auf dem ECUcore-5208 zeigt das Demoprojekt unter *"/projects/ECUcore-5208/user/hellocan"* (siehe Abschnitt 7.5.2).

Im Auslieferungszustand des ECUcore-5208 ist bereits ein fertig generiertes und sofort per *"insmod"* ladbares Kernelmodul des CAN-Treibers im Verzeichnis *"/home/bin"* abgelegt (siehe Abschnitt 5.13):

```
insmod /home/bin/candrv.ko
```

7.5.2 CAN-Treiber Demoprojekt

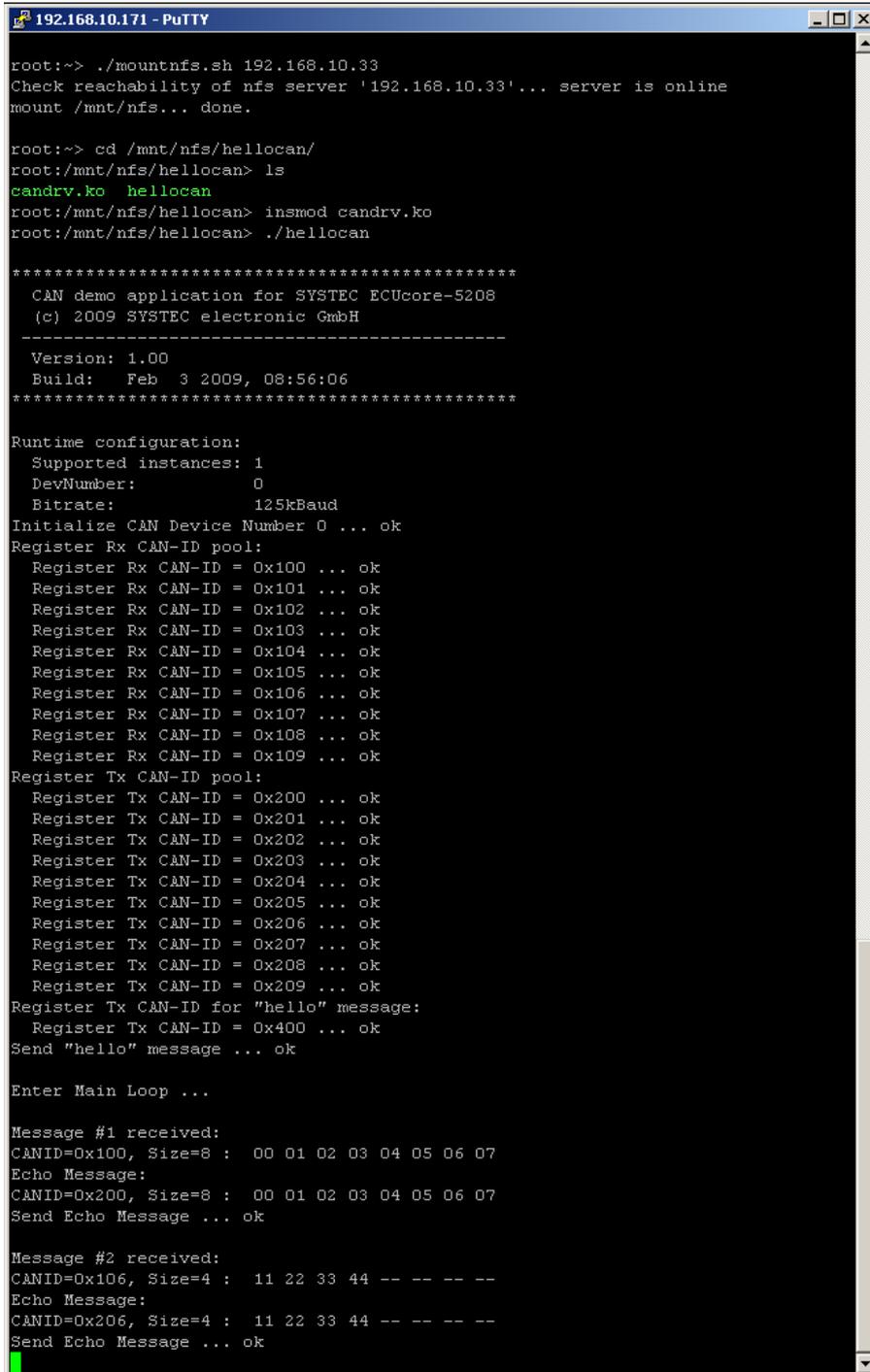
Das Demoprojekt für den CAN-Treiber ist im Verzeichnis ***"/projects/ECUcore-5208/user/hellocan"*** des Linux-Entwicklungssystems abgelegt. Es veranschaulicht den Zugriff auf die CAN-Schnittstellen des Moduls. Dazu bedient sich das Demoprogramm der Hilfe des unter ***"/projects/ECUcore-5208/driver/candrv"*** abgelegten CAN-Treibers.

Zur Demonstration des Nachrichtenaustausches über den CAN-Bus wird eine entsprechende Gegenstelle benötigt. Hierfür bieten sich idealer Weise das CAN-Analysetool *"CAN-REport"* in Verbindung mit einem USB-CANmodul an. Mit Hilfe des *"CAN-REport"* können beliebige CAN-Nachrichten gesendet und empfangen werden (siehe Bild 43).

Das Demoprogramm *"hellocan"* benutzt **125kBit/s** als Bitrate. Es legt nach seinem Start zunächst je 10 Kanäle für den Empfang von CAN-Nachrichten sowie 10 Kanäle zum Versenden an:

Empfangsbereich: CAN-Nachrichten im Identifier-Bereich 0x100 - 0x109
 Sendebereich: CAN-Nachrichten im Identifier-Bereich 0x200 - 0x209

Nach seiner erfolgreichen Initialisierung sendet das Demoprogramm "hellocan" einmalig eine CAN-Nachricht mit dem Identifier 0x400 und dem Dateninhalt "68 61 6C 6C 6f 63 61 6E" (ASCII: "hellocan") auf den Bus. Anschließend geht es in seine Hauptschleife über, in der es auf dem Empfang von CAN-Nachrichten im spezifizierten Identifier-Bereich 0x100 - 0x109 wartet. Beim Empfang einer entsprechenden Nachricht wird diese mit einem um 0x100 erhöhten Identifier (entspricht Identifier-Bereich 0x200 - 0x209) als Echo wieder zurück gesendet. Bild 42 zeigt die Ausführung des Demoprojektes "hellocan" auf dem ECUcore-5208. Das Demoprojekt kann durch Drücken von "Ctrl+C" beendet werden.



```

192.168.10.171 - PuTTY
root:~> ./mountnfs.sh 192.168.10.33
Check reachability of nfs server '192.168.10.33'... server is online
mount /mnt/nfs... done.

root:~> cd /mnt/nfs/hellocan/
root:/mnt/nfs/hellocan> ls
candrv.ko  hellocan
root:/mnt/nfs/hellocan> insmod candrv.ko
root:/mnt/nfs/hellocan> ./hellocan

*****
CAN demo application for SYSTEC ECUcore-5208
(c) 2009 SYSTEC electronic GmbH
-----
Version: 1.00
Build: Feb 3 2009, 08:56:06
*****

Runtime configuration:
Supported instances: 1
DevNumber: 0
Bitrate: 125kBaud
Initialize CAN Device Number 0 ... ok
Register Rx CAN-ID pool:
Register Rx CAN-ID = 0x100 ... ok
Register Rx CAN-ID = 0x101 ... ok
Register Rx CAN-ID = 0x102 ... ok
Register Rx CAN-ID = 0x103 ... ok
Register Rx CAN-ID = 0x104 ... ok
Register Rx CAN-ID = 0x105 ... ok
Register Rx CAN-ID = 0x106 ... ok
Register Rx CAN-ID = 0x107 ... ok
Register Rx CAN-ID = 0x108 ... ok
Register Rx CAN-ID = 0x109 ... ok
Register Tx CAN-ID pool:
Register Tx CAN-ID = 0x200 ... ok
Register Tx CAN-ID = 0x201 ... ok
Register Tx CAN-ID = 0x202 ... ok
Register Tx CAN-ID = 0x203 ... ok
Register Tx CAN-ID = 0x204 ... ok
Register Tx CAN-ID = 0x205 ... ok
Register Tx CAN-ID = 0x206 ... ok
Register Tx CAN-ID = 0x207 ... ok
Register Tx CAN-ID = 0x208 ... ok
Register Tx CAN-ID = 0x209 ... ok
Register Tx CAN-ID for "hello" message:
Register Tx CAN-ID = 0x400 ... ok
Send "hello" message ... ok

Enter Main Loop ...

Message #1 received:
CANID=0x100, Size=8 : 00 01 02 03 04 05 06 07
Echo Message:
CANID=0x200, Size=8 : 00 01 02 03 04 05 06 07
Send Echo Message ... ok

Message #2 received:
CANID=0x106, Size=4 : 11 22 33 44 -- -- -- --
Echo Message:
CANID=0x206, Size=4 : 11 22 33 44 -- -- -- --
Send Echo Message ... ok

```

Bild 42: Ausführung des Demoprojektes "hellocan" auf dem ECUcore-5208

Bild 43 zeigt den Datenaustausch mit dem Demoprogramm "hellocan" im CAN-Bus Analysetool "CAN-REport".

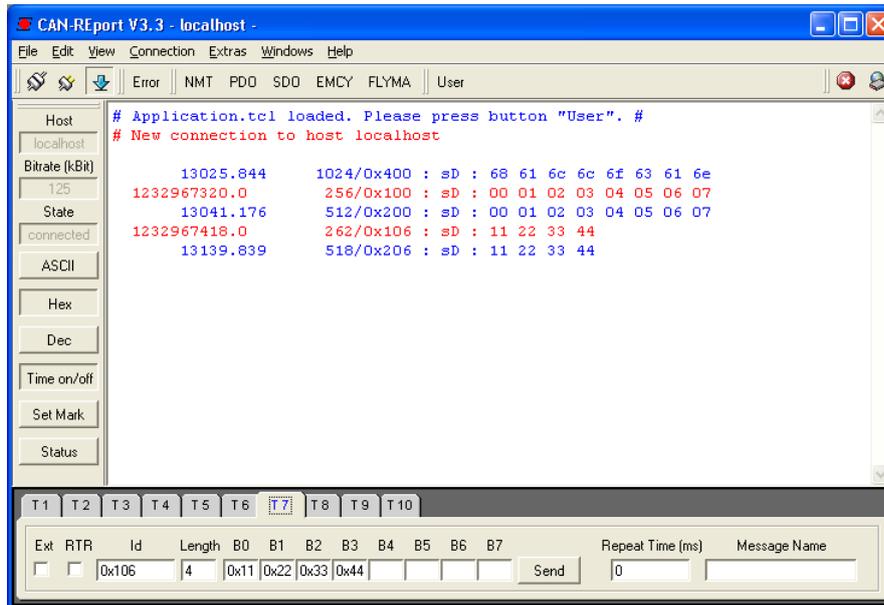


Bild 43: CAN-Analysetool "CAN-REport"

7.6 Übertragen von Programmen auf dem ECUcore-5208

Voraussetzung für das Starten von Anwenderprogrammen auf dem ECUcore-5208 ist zunächst die Konfiguration des Moduls wie im Abschnitt 5.5 beschrieben. Nach dem anschließenden Neustart des Moduls ist im Terminalfenster auf dem ECUcore-5208 die Anmeldung an der Kommando-Shell des ECUcore-5208 entsprechend Abschnitt 5.5.1 notwendig.

Die Übertragung von Programmen auf das ECUcore-5208, bzw. ganz allgemein der Dateiaustausch zwischen Entwicklungssystem (VMware Linux-Image) und dem ECUcore-5208, kann auf zwei verschiedenen Wegen erfolgen, die beide mit Vor- und Nachteilen behaftet sind:

- NFS:** Das "Network File System" (NFS) stellt die einfachste Form dar, ein im Linux-Image übersetztes Anwenderprogramm direkt auf dem ECUcore-5208 zu starten. Dazu wird ein Verzeichnis aus dem Linux-Image in das lokale Dateisystem des ECUcore-5208 eingebunden ("gemountet" oder "montiert"). Zum Starten des Programms muss dann auf dem ECUcore-5208 nur noch der entsprechende Befehlsname eingegeben werden. Die notwendige Dateiübertragung vom Entwicklungssystem auf das ECUcore-5208 erfolgt implizit durch NFS, ohne dass vom Anwender hierzu weitere Kommandos notwendig sind. Es ist also sichergestellt, dass auf dem ECUcore-5208 stets die aktuell auf dem Entwicklungssystem vorliegende Programmversion ausgeführt wird und nicht eine veraltete. Damit empfiehlt sich NFS insbesondere während der Softwareentwicklung. Auch können so transparent Dateien vom ECUcore-5208 auf das VMware-Linux übertragen werden. Nachteilig an NFS ist zum einen die Tatsache, dass hierüber nur Verbindungen zu anderen Linux-Maschinen möglich sind, nicht aber z.B. zu Windows-PCs. Zum anderen lässt NFS nur eine sehr rudimentäre Nutzerverwaltung und damit Zugangskontrolle zu, was beim späteren realen Einsatz der Geräte im Feld oftmals unerwünscht ist. Die Verwendung von NFS beschreibt Abschnitt 7.6.1.
- FTP:** Das "File Transfer Protokoll" ist als standardisiertes und in der Praxis etabliertes Protokoll plattformunabhängig. Sowohl FTP-Server als auch -Clients sind für die verschiedensten Betriebssysteme verfügbar, darunter Linux und Windows. Mit Hilfe von FTP ist es also im Gegensatz zu NFS auch möglich, Dateien von einem Windows-PC auf das ECUcore-5208 zu übertragen (z.B. Programmupdate durch Service-Techniker mit Windows-Laptop). Darüber hinaus erlaubt FTP eine detaillierte Zugangskontrolle durch die Authentifizierung mit

Username und Passwort. Nachteilig an FTP ist die unverschlüsselte Übertragung von Login und Passwort. Weiterhin kann die für jede Dateiübertragung erforderliche Kommandoeingabe gerade während der Entwicklungsphase oft als lästig empfunden bzw. vergessen werden. Auf dem ECUcore-5208 wird dann u. U. unbeabsichtigt und unbemerkt eine veraltete Programmversion ausgeführt. Die Verwendung von FTP beschreibt Abschnitt 7.6.2.

7.6.1 Verwendung von NFS

Die einfachste Form ein im Linux-Image übersetztes Anwenderprogramm auf dem ECUcore-5208 zu starten, besteht darin, ein Verzeichnis aus dem Linux-Image direkt in das lokale Dateisystem des ECUcore-5208 einzubinden ("zu mounten"). Dazu exportiert das VMware-Image des Linux-Entwicklungssystems das Verzeichnis "tftpboot", inklusive aller darin enthaltenen Unterverzeichnisse. Um diesen Verzeichnisbaum des Entwicklungssystems in das lokale Dateisystem des ECUcore-5208 einzubinden, sind folgende Schritte notwendig:

1. Bestimmung der IP-Adresse des Linux-Entwicklungssystems

Die Vorgehensweise zur Bestimmung der IP-Adresse des Linux-Images ist im Abschnitt 6.5 beschrieben.

2. Mounten des Linux-Images auf dem ECUcore-5208

Um das Verzeichnis "/tftpboot" des Linux-Images im lokalen Dateisystem des ECUcore-5208 einzubinden, ist der Befehl "mount" wie folgt zu verwenden:

```
mount -t nfs -o nolock <ip_vmware_image>:/tftpboot /mnt/nfs
```

Um beispielsweise das ECUcore-5208 an das Linux-Entwicklungssystem mit der im Abschnitt 6.5 ermittelten IP-Adresse anzubinden, ist auf dem ECUcore-5208 folgender Befehl einzugeben:

```
mount -t nfs -o nolock 192.168.10.33:/tftpboot /mnt/nfs
```

Um die Nutzung des Mount-Befehls etwas zu vereinfachen, existiert im Home-Verzeichnis des ECUcore-5208 (in diesem befindet man sich unmittelbar nach der Anmeldung am Kommandoprompt) das Skript "mountnfs.sh". Diesem Skript ist als Parameter lediglich die IP-Adresse des Linux-Images zu übergeben:

```
./mountnfs.sh 192.168.10.33
```

Nach der Ausführung des Mount-Befehls ist der gesamte Inhalt des Verzeichnisses "/tftpboot" des Linux-Images (inkl. evtl. dort enthaltener Unterordner) im lokalen Verzeichnis "/mnt/nfs" des ECUcore-5208 zugänglich.

Bild 44 fasst die notwendigen Schritte zum Mounten des Linux-Images in das lokale Dateisystem des ECUcore-5208 zusammen.

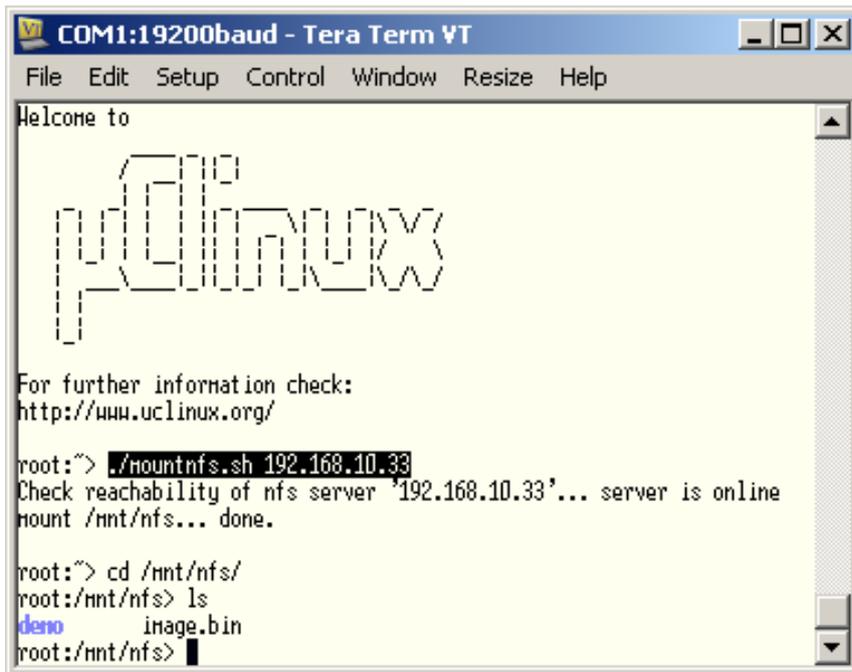


Bild 44: Mouten des Linux-Images in das lokale Dateisystem des ECUcore-5208

7.6.2 Verwendung von FTP

Alternativ zur Einbindung des Linux-Images in das lokale Dateisystem des ECUcore-5208 via NFS können Daten zwischen dem Entwicklungs-PC und dem ECUcore-5208 über FTP in beiden Richtungen transferiert werden. Dabei kann das ECUcore-5208 sowohl als Server als auch als Client fungieren.

Bei der Übertragung von ausführbaren Programmen über eine FTP-Verbindung ist zu beachten, dass hierbei grundsätzlich das "x"-Flag in den Dateiattributen ("eXecuteable") gelöscht wird. Daher ist nach einem FTP-Transfer das "x"-Flag mit Hilfe des Kommandos "chmod" wieder zu setzen (siehe auch Punkt "Aufruf von Programmen" im Abschnitt 9, "Tipps & Tricks im Umgang mit Linux bzw. uClinux"):

```
chmod +x ./mountnfs.sh
```

7.6.2.1 ECUcore-5208 als FTP-Client

Beim Einsatz des ECUcore-5208 als FTP-Client dient das Linux-Entwicklungssystem als Server. Diese Variante hat den Vorteil, dass beim späteren realen Einsatz des Gerätes im Feld kein Sicherheitsrisiko besteht, da auf dem Modul kein Serverdienst aktiviert sein muss und somit eine explizite Nutzerverwaltung nicht notwendig wird. Für den Einsatz des ECUcore-5208 als FTP-Client ist zunächst wieder die dem Linux-Entwicklungssystem per DHCP zugewiesene IP-Adresse zu ermitteln. Das Vorgehen dazu ist im Abschnitt 6.5 beschrieben.

FTP-Download

Der Download von Dateien aus dem Linux-Image auf das ECUcore-5208 erfolgt mit Hilfe des Befehls "ftpget". Die beiden Parameter "-u" für Username und "-p" für Passwort dienen zur Authentifizierung am Hostsystem. Der Befehl "ftpget" besitzt folgende Syntax:

```
ftpget -u <username> -p <password> <ip_vmware_image> <local_file> <remote_file>
```

Um beispielsweise das im Abschnitt 7.3 übersetzte und in das Verzeichnis `/tftpboot/demo` kopierte Programm `demo` per FTP in das lokale Verzeichnis `/tmp` des ECUcore-5208 zu übertragen, ist folgender Befehl notwendig:

```
ftpget -u vmware -p vmware 192.168.10.33 /tmp/demo /tftpboot/demo/demo
```

FTP-Upload

Der Upload von Dateien des ECUcore-5208 in das Linux-Image erfolgt mit Hilfe des Befehls "ftpput". Die beiden Parameter "-u" für Username und "-p" für Passwort dienen zur Authentifizierung am Hostsystem. Der Befehl "ftpput" besitzt folgende Syntax:

```
ftpput -u <username> -p <password> <ip_vmware_image> <remote_file> <local_file>
```

Um bspw. die Logdatei `/var/log/messages` vom ECUcore-5208 per FTP in das Verzeichnis `/tftpboot` des Hostsystems zu übertragen, ist folgender Befehl notwendig:

```
ftpput -u vmware -p vmware 192.168.10.33 /tftpboot/messages /var/log/messages
```

Bild 45 verdeutlicht die Anwendung der Befehle "ftpget" und "ftpput" zum Download und Upload von Dateien über FTP.

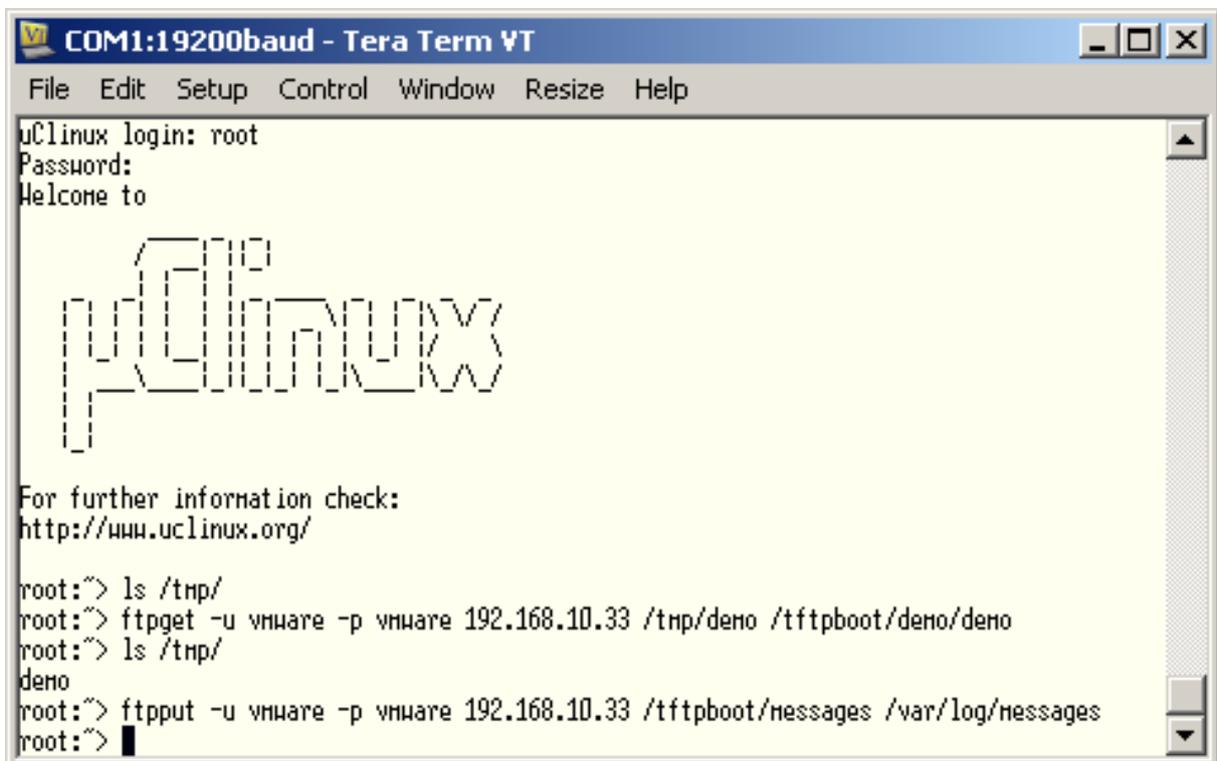


Bild 45: Download und Upload über FTP

7.6.2.2 ECUcore-5208 als FTP-Server

Das ECUcore-5208 verfügt über einen FTP-Dienst (via Internet Service Daemon), der den Austausch von Dateien mit einem beliebigen FTP-Client ermöglicht (z.B. Up- und Download von Dateien zu bzw.

von einem PC). Aus Sicherheits- und Performance-Gründen ist dieser FTP-Dienst jedoch standardmäßig deaktiviert und muss bei Bedarf zunächst manuell aktiviert werden.

Der Vorteil beim Einsatz des FTP-Servers auf dem ECUcore-5208 besteht darin, dass für die Client-Seite komfortable grafische Programme existieren, die einen einfachen Datenaustausch ohne tiefgreifende Kenntnis von Linux-Kommandos ermöglichen. Ein Servicetechniker ist so beispielsweise beim späteren realen Einsatz des Gerätes im Feld in der Lage, mit einem grafischen FTP-Client auf seinem Laptop ein Firmwareupdate auf das ECUcore-5208 zu übertragen oder Logfiles von diesem Modul auszulesen.

Die Aktivierung des FTP-Servers auf dem ECUcore-5208 sowie die Anmeldung am Modul durch einen FTP-Client beschreibt Abschnitt 5.5.4. Geeignete FTP-Client Programme sind im Abschnitt 5.1 aufgeführt.

7.7 Übersetzen und Ausführen des Demoprojektes "demo"

7.7.1 Verwendung von "make"

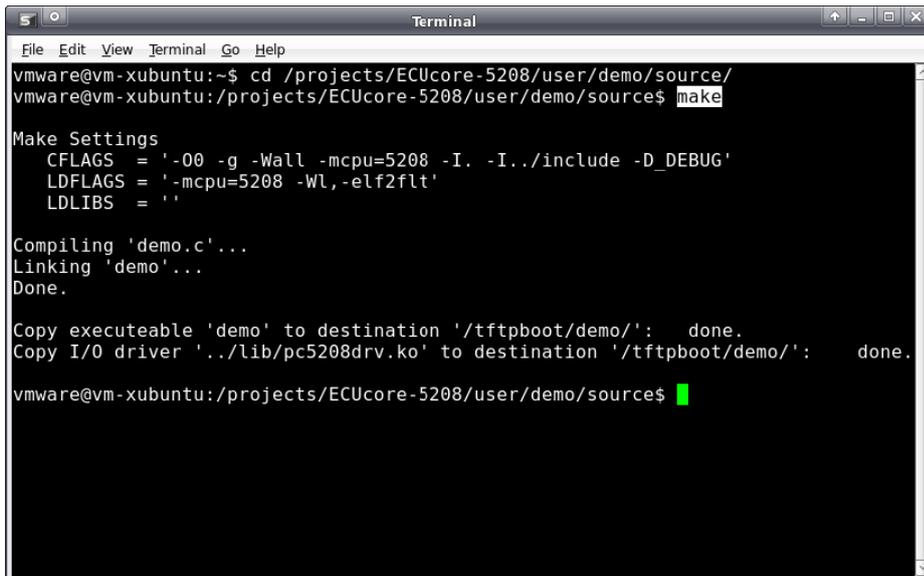
Das VMware-Image des Linux-Entwicklungssystems enthält im Verzeichnis ***"/projects/ECUcore-5208/user/demo"*** ein Demoprogramm, das den Zugriff auf die Ein- und Ausgänge des Moduls veranschaulicht. Das Demoprogramm bedient sich für die I/O-Zugriff der Hilfe des unter ***"/projects/ECUcore-5208/pc5208drv"*** abgelegten I/O-Treibers. Das hier beschriebene Demoprojekt, zumindest aber dessen Makefile, sollte als Vorlage für eigene Projekte verwendet werden.

Während der Quelltext über die Windows-Netzwerkumgebung mit jedem beliebigen Windows-Editor erstellt bzw. bearbeitet werden kann, ist das Übersetzen des Projektes ausschließlich innerhalb des Linux-Entwicklungssystems selbst möglich. Dazu kann entweder ein Konsolenfenster im Linux-Image verwendet werden (dort auch als "Terminal" bezeichnet), oder es ist der Zugriff über einen Telnet-Client "von außen" notwendig, in dem dann alle notwendigen Schritte analog erfolgen. Im Folgenden wird daher auch keine Unterscheidung zwischen Konsolenfenster innerhalb des Linux-Images und Telnet-Zugriff "von außen" getroffen.

Innerhalb des Konsolenfensters ist zunächst mit dem Befehl ***"cd"*** (*cd = change directory*) in das entsprechende Verzeichnis des Projektes zu wechseln, in dem sich das Makefile befindet. Zum Übersetzen des Demoprojektes ist dazu in das Verzeichnis ***"/projects/ECUcore-5208/user/demo/source"*** zu wechseln:

```
cd /projects/ECUcore-5208/user/demo/source
```

Anschließend ist hier der Befehl ***"make"*** zu starten. Bild 46 zeigt die Anwendung der Befehle ***"cd"*** und ***"make"*** am Beispiel des im VMware-Image enthaltenen Demoprogramms.



```

Terminal
File Edit View Terminal Go Help
vmware@vm-xubuntu:~$ cd /projects/ECUcore-5208/user/demo/source/
vmware@vm-xubuntu:/projects/ECUcore-5208/user/demo/source$ make

Make Settings
CFLAGS = '-O0 -g -Wall -mcpu=5208 -I. -I../include -D_DEBUG'
LDFLAGS = '-mcpu=5208 -Wl,-elf2flt'
LDLIBS = ''

Compiling 'demo.c'...
Linking 'demo'...
Done.

Copy executable 'demo' to destination '/tftpboot/demo/': done.
Copy I/O driver '../lib/pc5208drv.ko' to destination '/tftpboot/demo/': done.

vmware@vm-xubuntu:/projects/ECUcore-5208/user/demo/source$

```

Bild 46: Übersetzen des Demoprojektes im Linux-Image

Bei der Abarbeitung des Makefiles wurden nach dem erfolgreichen Abschluss des Build-Prozesses sowohl das erstellte Demoprogramm selbst (Datei "demo") als auch der zu seiner Ausführung benötigte I/O-Treiber (Dateien "5208-spidev.ko" "pc5208drv.ko") in das Verzeichnis "/tftpboot/demo" kopiert (siehe Bild 46). Mit dem Übersetzen und Kopieren sind alle notwendigen Schritte im Linux-Entwicklungssystem abgeschlossen.

Alle weiteren Schritte erfolgen jetzt ausschließlich auf dem ECUcore-5208 selbst. Dazu ist zunächst die Anmeldung an der Kommando-Shell des Moduls notwendig (siehe Abschnitt 5.5.1). Innerhalb des Terminalprogramms bzw. Telnet-Clients sind anschließend folgende Schritte auszuführen:

1. Einbinden des Verzeichnisses "/tftpboot" vom Linux-Entwicklungssystem in das lokale Filesystem des ECUcore-5208 per NFS (siehe Abschnitt 7.6.1):

```
mount -t nfs -o nolock 192.168.10.33:/tftpboot /mnt/nfs
```

2. Wechseln in das NFS-Verzeichnis im lokalen Dateisystem mit Hilfe des Befehls "cd":

```
cd /mnt/nfs/demo
```

Das Verzeichnis "/mnt/nfs" auf dem ECUcore-5208 ist identisch mit dem Verzeichnis "/tftpboot" des Linux-Entwicklungssystems im VMware-Image. Dementsprechend sind auch die vom Makefile auf dem Linux-Entwicklungssystem in das Verzeichnis "/tftpboot/demo" kopierten Dateien für das ECUcore-5208 in seinem Dateisystem im Verzeichnis "/mnt/nfs/demo" zugänglich. Ein expliziter Download der ausführbaren Binärdateien auf das ECUcore-5208 ist somit nicht notwendig.

3. Starten des Demoprogramms auf dem ECUcore-5208.

Da das Demoprogramm auf die Ein- und Ausgänge des ECUcore-5208 zugreift, benötigt es für seine Ausführung den I/O-Treiber "pc5208drv". Dementsprechend ist zunächst der I/O-Treiber mit dem Befehl "insmod" explizit zu laden, anschließend kann das Demoprogramm gestartet werden:

```
insmod 5208-spidev.ko
insmod pc5208drv.ko
./demo
```

```

192.168.10.171 - PuTTY
root:~> mount -t nfs -o nolock 192.168.10.33:/tftpboot /mnt/nfs/
root:~> cd /mnt/nfs/demo/
root:/mnt/nfs/demo> ls
demo          pc5208drv.ko
root:/mnt/nfs/demo> insmod pc5208drv.ko
root:/mnt/nfs/demo> ./demo

*****
I/O demo application for SYSTEC ECUcore-5208
(c) 2009 SYSTEC electronic GmbH
-----
Version: 1.00
Build:   Jan 29 2009, 13:41:21
*****

I/O Driver version:  KernelModule=3.00, UserLib=3.00

Hardware:  CPU Board: 4218.00 (#00H)
           CPU PLD:  0.00  (#00H)
           IO Board: 4220.00 (#00H)
Driver:    Config:   0000H

Start basic I/O main loop...

DI=00-00-00  DO=00-00-01  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-02  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-04  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-08  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-10  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-20  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-40  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-00-80  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-01-00  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-02-00  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
DI=00-00-00  DO=00-04-00  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN
^C
root:/mnt/nfs/demo>

```

Bild 47: Ausführung des Demoprojektes "demo" auf dem ECUcore-5208

Bild 47 verdeutlicht die Ausführung des Demoprojektes auf dem ECUcore-5208. Das Demoprojekt kann entweder durch Setzen des Run/Stop-Schalters in die Position "Res" oder durch Drücken von "Ctrl+C" beendet werden.

7.7.2 Verwendung der grafischen IDE "Eclipse"

Im VMware-Image des Linux-Entwicklungssystems ist "Eclipse" als grafische IDE installiert. Diese erlaubt das Ausführen sämtlicher Arbeitsschritte im Softwareentwicklungsprozess wie Editieren, Übersetzen und Debuggen von Anwenderprogrammen innerhalb einer komfortablen Entwicklungsumgebung. Die folgenden Abschnitte beschreiben den Umgang mit der IDE "Eclipse" am Beispiel des im VMware-Image enthaltenen und im Abschnitt 7.7.1 bereits beschriebenen Demoprojektes im Pfad **"/projects/ECUcore-5208/user/demo"**.

7.7.2.1 Öffnen und Bearbeiten des Demoprojektes

Die grafische IDE "Eclipse" wird durch Anklicken des entsprechenden Desktop-Symbols gestartet. Dabei erscheint zunächst der Dialog "Workspace Launcher", wie im Bild 48 dargestellt. Im Feld "Workspace" ist der Pfad für das Workspace-Verzeichnis innerhalb des Projektbaumes einzutragen. Für das im Linux-Entwicklungssystem enthaltene Demo-Projekt ist dies **"/projects/ECUcore-5208/user/demo/workspace"**.

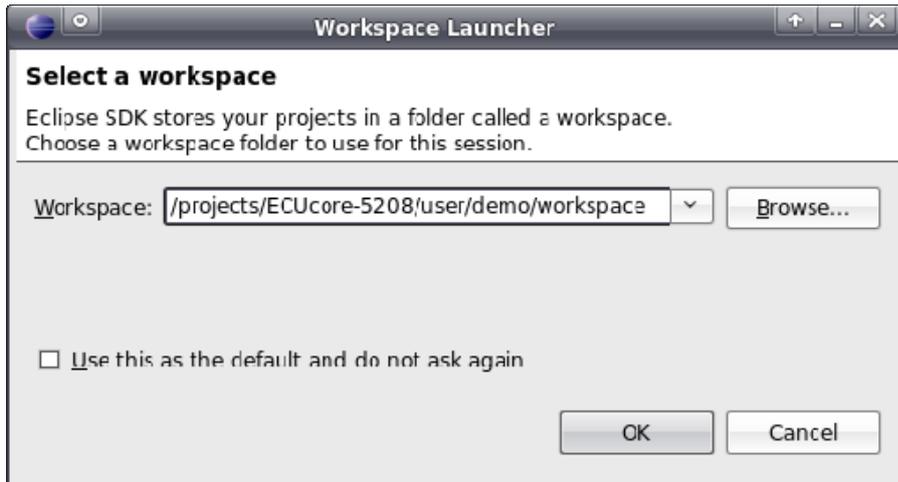


Bild 48: Eclipse-Dialog "Workspace Launcher"

Nach dem Betätigen der Schaltfläche "OK" startet die grafische Oberfläche der IDE und lädt den angegebenen Workspace. Bild 49 zeigt die Gesamtansicht von "Eclipse" nach dem Start.

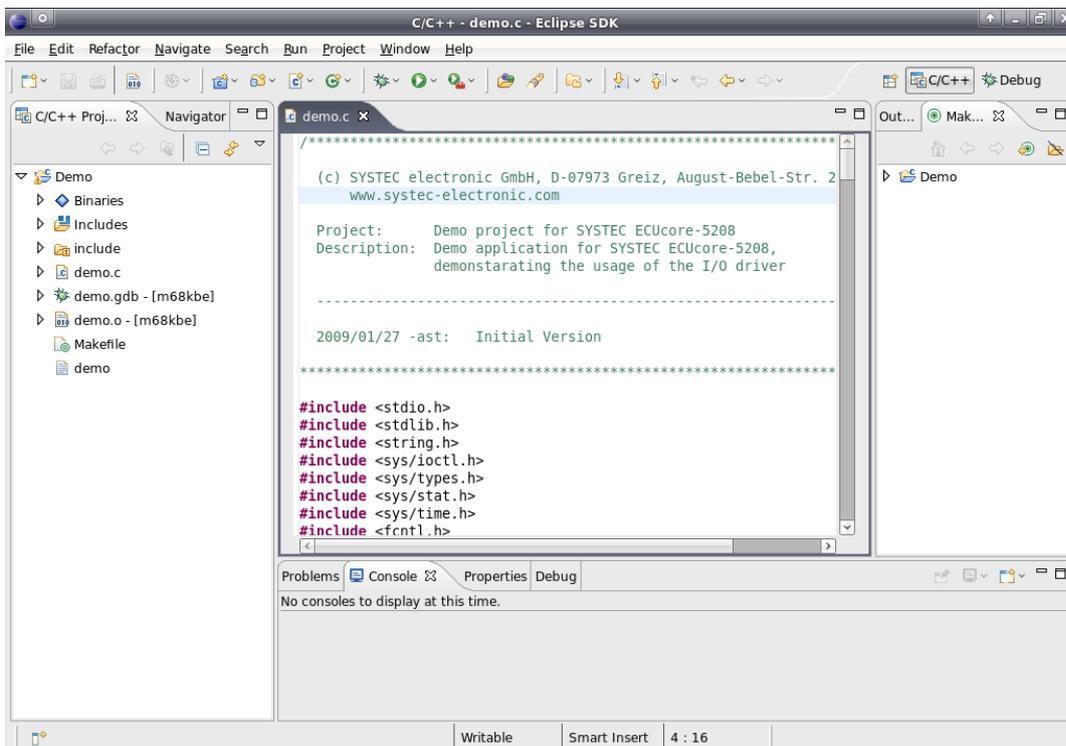


Bild 49: Ansicht der grafischen IDE "Eclipse"

Im Editorfenster kann der Sourcecode des Demoprogramms bearbeitet werden. Durch Doppelklick auf einen Eintrag im Projektbaum (links) lassen sich alle im Projekt enthaltenen Dateien öffnen und bearbeiten.

7.7.2.2 Übersetzen des Demoprojektes

Zum Übersetzen des Demoprojektes ist zunächst im Fenster "Make Targets" der Eintrag "Demo" durch Anklicken des vorangestellten Dreieckes aufzuklappen. Anschließend ist der Übersetzungslauf mit einem Doppelklick auf den Eintrag "Build Project" aufzurufen (siehe Bild 50). Dadurch führt Eclipse das Makefile des Demoprojektes im Verzeichnis "source" aus ("/projects/ECUcore-5208/user/demo/source/Makefile"). Das ist genau dasselbe Makefile, das bereits im Abschnitt 7.7.1

manuell in einem Terminalfenster aufgerufen wurde. Dementsprechend sind auch die im IDE-Fenster "Console" angezeigten Meldungen identisch zu denen im Bild 46. Ebenso werden auch hier wieder sowohl das erstellte Demoprogramm selbst (Datei "demo") als auch der zu seiner Ausführung benötigte I/O-Treiber (Dateien "5208-spidev.ko" "pc5208drv.ko") in das Verzeichnis "/tftpboot/demo" kopiert. Folglich kann auch hier nach dem erfolgreichen Abschluss des Build-Prozesses das Demoprogramm wie im Abschnitt 7.7.1 beschrieben auf dem ECUcore-5208 gestartet werden.

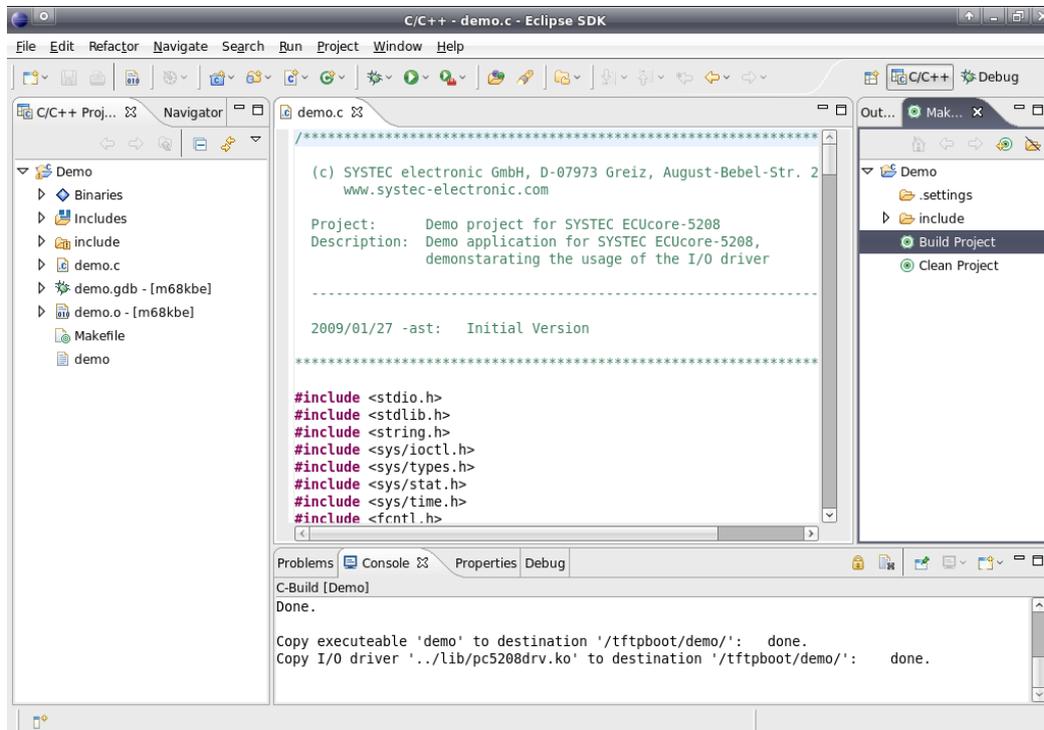


Bild 50: Übersetzen des Demoprojektes in Eclipse

Wenn beim Übersetzen Fehler oder Warnungen auftreten (z.B. infolge von Modifikationen des Demoprojekts durch den Anwender), dann werden diese im IDE-Fenster "Problems" strukturiert dargestellt. Durch einen Doppelklick auf einen Eintrag öffnet die IDE die betreffende Quelltextdatei und markiert die Zeile entsprechend im Editor.

Durch einen Doppelklick auf den Eintrag "Clean Project" (siehe Bild 50) werden alle generierten Dateien wieder gelöscht. Durch Anklicken der Einträge "Build Project" bzw. "Clean Project" mit der rechten Maustaste können die entsprechenden Targets bearbeitet werden (z.B. Änderung des zugehörigen Namens).

7.7.2.3 Debuggen des Demoprojektes in der IDE

Einer der wesentlichen Vorteile bei der Verwendung von Eclipse ist die Möglichkeit, das übersetzte Programm innerhalb der IDE direkt auf Hochsprachenebene debuggen zu können. Dazu zählen z.B. die zeilenweise Programmabarbeitung auf C-Ebene, das Setzen von Breakpoints direkt im Quellcode sowie die Beobachtung von Variablen innerhalb der IDE. Während die Eclipse-IDE auf dem PC mit der Linux-Entwicklungsumgebung läuft, wird das zu debuggende Programm unter der Kontrolle eines Debugservers (gdbserver) direkt auf dem ECUcore-5208 selbst ausgeführt. Bei dieser als "Remote-Debugging" bezeichneten Vorgehensweise wird sowohl Zugriff auf die Linux-Entwicklungsumgebung (Host-PC) als auch auf das ECUcore-5208 (Target) benötigt.

Im Folgenden werden die zum Debuggen einer Anwenderapplikation notwendigen Schritte exemplarisch am Beispiel des im VMware-Image des Linux-Entwicklungssystems enthaltenen Demoprojektes beschrieben:

1. Starten des Debugservers mit dem Anwenderprogramm auf dem ECUcore-5208

Zum Debuggen soll das Demoprogramm auf dem ECUcore-5208 direkt aus dem NFS-Verzeichnis des Linux-Entwicklungssystems gestartet werden. Dazu ist zunächst die Anmeldung an der Kommando-Shell des Moduls notwendig (siehe Abschnitt 5.5.1). Innerhalb des Terminalprogramms bzw. Telnet-Clients sind anschließend folgende Schritte auszuführen:

- 1.1 Einbinden des Verzeichnisses `"/projects/tftpboot"` vom Linux-Entwicklungssystem in das lokale Filesystem des ECUcore-5208 per NFS (siehe Abschnitt 7.6.1):

```
mount -t nfs -o nolock 192.168.10.33:/tftpboot /mnt/nfs
```

- 1.2 Wechseln in das NFS-Verzeichnis im lokalen Dateisystem mit Hilfe des Befehls `"cd"`:

```
cd /mnt/nfs/demo
```

Das Verzeichnis `"/mnt/nfs"` auf dem ECUcore-5208 ist identisch mit dem Verzeichnis `"/tftpboot"` des Linux-Entwicklungssystems im VMware-Image. Dementsprechend sind auch die vom Makefile auf dem Linux-Entwicklungssystem in das Verzeichnis `"/tftpboot/demo"` kopierten Dateien für das ECUcore-5208 in seinem Dateisystem im Verzeichnis `"/mnt/nfs/demo"` zugänglich. Ein expliziter Download der ausführbaren Binärdateien auf das ECUcore-5208 ist somit nicht notwendig.

- 1.3 Da das Demoprogramm auf die Ein- und Ausgänge des ECUcore-5208 zugreift, benötigt es für seine Ausführung den I/O-Treiber `"pc5208drv"`. Dementsprechend ist zunächst der I/O-Treiber mit dem Befehl `"insmod"` explizit zu laden:

```
insmod 5208-spidev.ko  
insmod pc5208drv.ko
```

- 1.4 Starten des Demoprogramms auf dem ECUcore-5208 unter Kontrolle des Debugservers, der dazu notwendige Befehl `"gdbserver"` besitzt folgendes, allgemeines Aufrufformat:

```
gdbserver <ip_vmware_image>:<port> <program> [args ...]
```

Für das oben angeführte Beispiel ergibt sich damit folgender Aufruf:

```
gdbserver 192.168.10.33:10000 ./demo
```

Dabei ist `"192.168.10.33"` die IP-Adresse des Linux-Entwicklungssystems (zur Ermittlung der IP-Adresse siehe Abschnitt 6.5). Die nach dem Doppelpunkt folgende Portnummer (hier: `"10000"`) ist frei wählbar, sie muss jedoch mit der nachfolgend im Punkt 2.3 innerhalb von Eclipse für die Target-Verbindung angegebenen Portnummer korrespondieren (auch siehe Bild 54).

Bild 51 verdeutlicht die auszuführenden Schritte auf dem ECUcore-5208.

2.1 Festlegen des im Debugger auszuführenden Programms (siehe Bild 52)

Dazu ist im Reiter *"Main"* im Feld *"C/C++ Application"* der Name des zu debuggenden Programms anzugeben (für das angegebene Beispiel: *"demo.gdb"*).

Wichtig dabei ist, dass die ELF-Datei mit der Endung **.gdb** ausgewählt wird. Auf dem ECUcore-5208 wird das sogenannte FLAT-Dateiformat verwendet, während der Debugger eine ELF-Datei benötigt. Beide Dateien werden vom C Compiler gcc erzeugt.

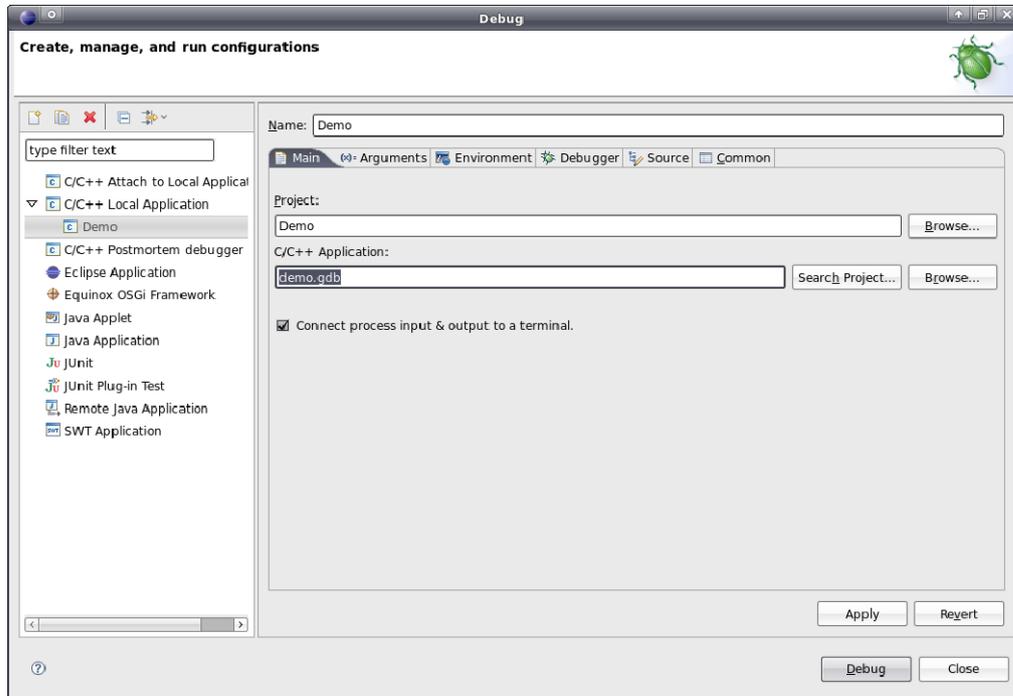


Bild 52: Festlegen der zu debuggenden Applikation

2.2 Auswahl des zu benutzenden GDB Debuggers (siehe Bild 53)

Auf der Host-Seite muss der in der GNU-Crosscompiler Toolchain für Freescale MCF52xx-Prozessoren enthaltene GDB Debugger verwendet werden. Um diesen auszuwählen, ist zunächst der Reiter *"Debugger"* zu aktivieren. In der Sektion *"Debugger Options"* ist im Unterreiter *"Main"* im Feld *"GDB debugger"* (siehe Bild 53) der entsprechende GDB Debugger der Crosscompiler Toolchain mit vollständigem Pfad anzugeben (für das angegebene Beispiel: *"/projects/ECUcore-5208/toolchain-uclinux-4.2/bin/m68k-uclinux-gdb"*).

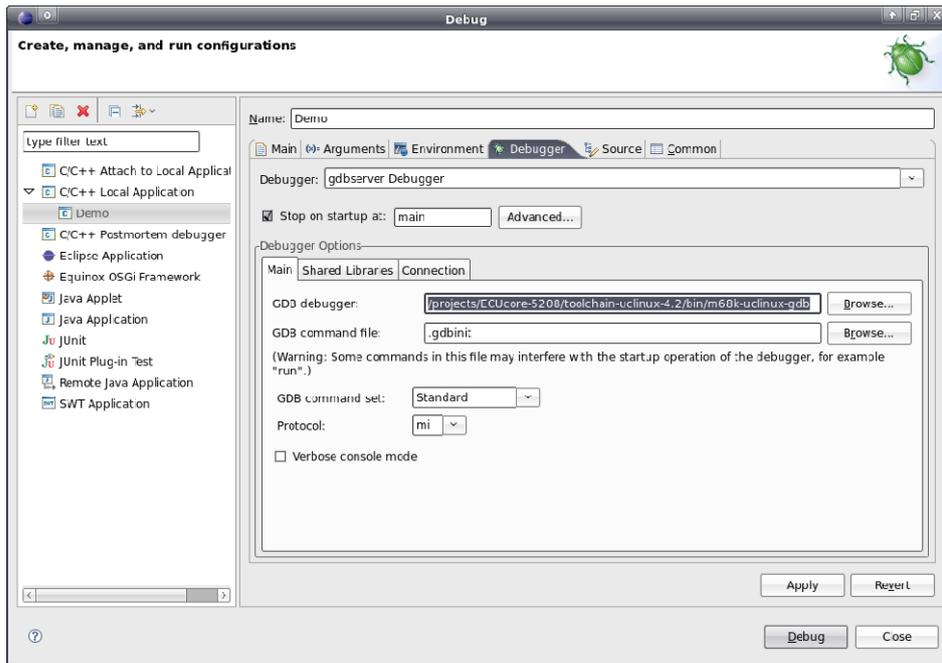


Bild 53: Auswahl des GDB Debuggers

2.3 Konfiguration der Verbindung zum Target (siehe Bild 54)

Die Konfiguration der Verbindung zum Target erfolgt ebenfalls im Reiter "Debugger". Dazu sind in der Sektion "Debugger Options" im Unter-Reiter "Connection" die entsprechenden Informationen einzutragen (siehe Bild 54). Im Feld "Host name or IP address" ist die im Abschnitt 5.4 festgelegten IP-Adresse für das ECUcore-5208 einzutragen (für das angegebene Beispiel: "192.168.10.171"). Im Feld "Port number" ist die beim Aufruf des Befehls "gdbserver" im Punkt 1.4 definierte Portnummer anzugeben (für das angegebene Beispiel: "10000").

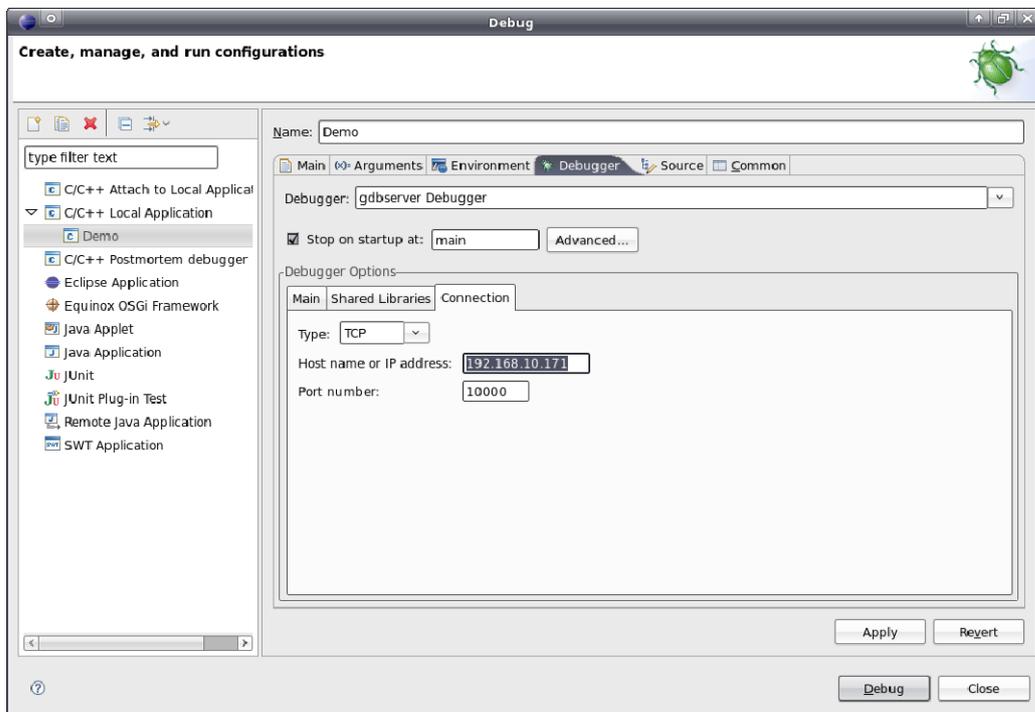


Bild 54: Konfiguration der Verbindung zum Target

Damit ist die Konfiguration des Debuggers abgeschlossen. Durch betätigen der Schaltfläche "Debug" wird der Debugger mit den aktuellen Einstellungen gestartet.

3. Ausführen des Debuggers in der IDE

Nach Abschluss der im Punkt 2 beschriebenen Konfiguration kann der Debugger innerhalb von "Eclipse" über den Menüpunkt "**Run -> Debug Last Launched**" aufgerufen werden. Dabei wechselt die IDE von der "C/C++ Perspektive" in die "Debug Perspektive" (siehe Bild 55). Tabelle 10 listet die wichtigsten Debugger-Kommandos auf.

Tabelle 10: Übersicht wichtiger Debugger-Kommandos

Tastenkürzel	Symbol	Funktion
F5		Step Into
F6		Step Over
F7		Step Return
F8		Run (ggf. bis zum nächsten Breakpoint)
Ctrl+Shift+B	(Doppelklick)	Toggle Line Breakpoint
		Terminate

Bild 55 verdeutlicht das Debuggen des Demoprogramms innerhalb der IDE "Eclipse". Beim Positionieren des Mauszeigers auf eine Variable wird deren aktueller Wert angezeigt.

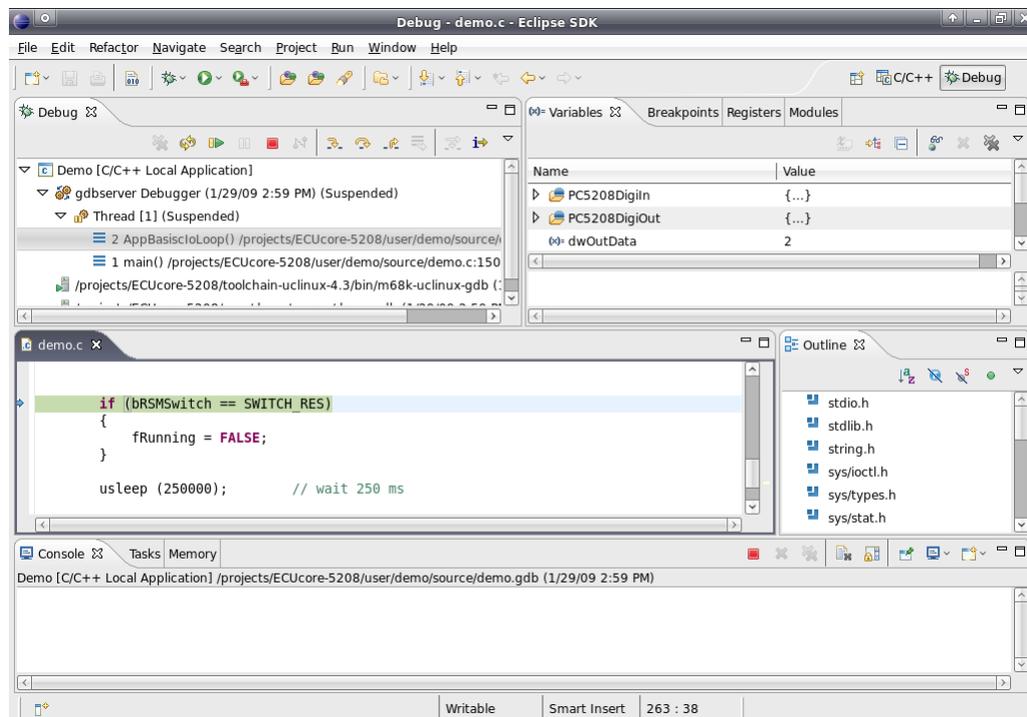


Bild 55: Debuggen des Demoprojektes in Eclipse

Die während des Debuggens auf dem ECUcore-5208 generierten Terminalausgaben zeigt Bild 56.

```

192.168.10.171 - PuTTY
root:~> mount -t nfs -o nolock 192.168.10.33:/tftpboot /mnt/nfs/
root:~> cd /mnt/nfs/demo/
root:/mnt/nfs/demo> ls
demo      pc5208drv.ko
root:/mnt/nfs/demo> insmod pc5208drv.ko
root:/mnt/nfs/demo> gdbserver 192.168.10.33:10000 ./demo
Process ./demo created; pid = 761
Listening on port 10000
Remote debugging from host 192.168.10.33

*****
I/O demo application for SYSTEC ECUcore-5208
(c) 2009 SYSTEC electronic GmbH
-----
Version: 1.00
Build:   Jan 29 2009, 13:41:21
*****

I/O Driver version:  KernelModule=3.00, UserLib=3.00

Hardware:  CPU Board: 4218.00 (#00H)
           CPU PLD:  0.00  (#00H)
           IO Board: 4220.00 (#00H)
Driver:    Config:   0000H

Start basic I/O main loop...

DI=00-00-00  DO=00-00-01  bHexSw=0x00  bDipSw=0x00  R/S/M-Sw=RUN

```

Bild 56: Terminalausgaben des ECUcore-5208 während des Debuggens

7.8 Konfigurieren und Übersetzen von uClinux-Image und CoLilo

Die gesamten uClinux-Quellen für das ECUcore-5208 befinden sich nach dem Entpacken (siehe Abschnitt 7.1) innerhalb des VMware-Images für das Linux-Entwicklungssystem im Pfad **"/projects/ECUcore-5208/uClinux-dist"**. Um die Konfiguration des uClinux-Kernels oder der im Image enthaltenen User-Applikationen zu modifizieren, ist innerhalb eines Konsolenfensters zunächst mit dem Befehl "cd" in das Verzeichnis der uClinux-Distribution zu wechseln und anschließend dort der Befehl "make menuconfig" aufzurufen:

```
cd /projects/ECUcore-5208/uClinux-dist
make menuconfig
```

Bild 57 zeigt die typische Oberfläche zur Konfiguration von uClinux-Kernel und Userspace-Applikationen.

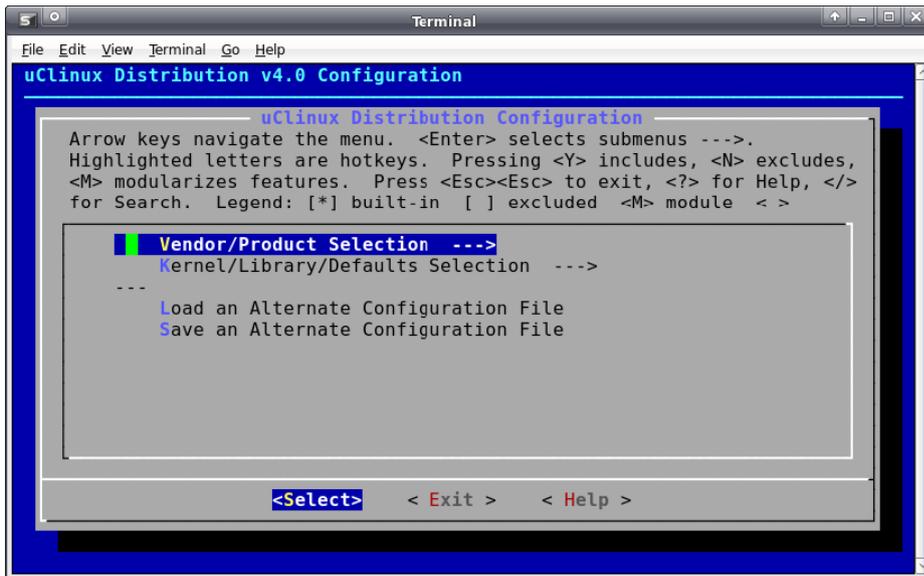


Bild 57: Benutzeroberfläche zur Konfiguration der uClinux-Distribution

Nach Abschluss der Konfiguration werden uClinux-Kernel sowie Userspace-Applikationen durch den Aufruf von "make" übersetzt und anschließend zu einem neuen Image zusammengefasst:

```
cd /projects/ECUcore-5208/uClinux-dist
make
```

Durch den Aufruf des folgenden Kommandos wird der uClinux-Image nach /tftpboot kopiert, von wo es auf das ECUcore-5208 geladen werden kann (siehe Abschnitt 5.15):

```
cp images/image.bin /tftpboot/
```

Um den Bootloader "CoLilo" neu zu erstellen, ist das Kommando "make" aufzurufen:

```
cd /projects/ECUcore-5208/colilo
make
```


9 Tipps & Tricks im Umgang mit Linux bzw. uClinux

Dieser Abschnitt skizziert in Kurzform einige Besonderheiten, die beim Umgang mit Linux und damit auch uClinux zu beachten sind. Hier können jedoch nur einige grundlegende Hinweise gegeben werden, für tiefergreifende Informationen ist ein geeignetes Linux-Nachschlagewerk notwendig.

- **Groß-/Kleinschreibung**

Im Gegensatz zu DOS/Windows beachtet Linux (wie auch andere Unix-Derivate) die Groß- und Kleinschreibung bei Namen von Verzeichnissen, Dateien, Programmen, ... Demzufolge ist bei Eingaben, aber auch Verweise auf z.B. Dateinamen innerhalb der Programme, Shellscripte unbedingt auf die richtige Schreibweise zu achten. Besonders beim externen Zugriff von Windows aus, ist sicherzustellen, dass auf dem Linux-Entwicklungssystem die korrekte Schreibweise angewandt wurde.

- **Zeilenumbruch**

Windows benutzt im Gegensatz zu allen anderen Unix-Derivate (und somit auch Linux) andere Steuerzeichen für den Zeilenumbruch. Unter Windows wird ein CR LF (carriage return & line feed = `\r\n`) verwendet, während Linux nur ein LF (line feed = `\n`) verwendet. Während Linux in den meisten Fällen mit beiden Varianten umgehen kann, kann es unter Windows zu Problemen bei der Darstellung kommen, falls der Texteditor keine andere Variante unterstützt. Besonders wichtig ist das Einstellen des richtigen (und damit Linux-)Zeilenumbruchs beim Erstellen von Shellskripten unter Windows. Shellskripte mit Windowszeilenumbruch werden **nicht** ausgeführt und erzeugen nur eine Fehlermeldung!

- **Versteckte Dateien und Verzeichnisse**

Versteckte Dateisystemeinträge werden durch einen Punkt "." am Anfang des Datei- bzw. Verzeichnisnamens markiert. Diese werden beim normalen Auflisten (z.B. durch das Kommando "ls") nicht angezeigt. Um versteckte Einträge dennoch sehen zu können, ist bei "ls" der Parameter "-a" (all) anzugeben. Ggf. ist dies beim Zugriff über FTP zu beachten.

- **Aufruf von Programmen (Suchpfad)**

Ähnlich wie DOS/Windows durchsucht Linux beim Aufruf eines Kommandos nur die in der Umgebungsvariablen "PATH" definierten Pfade, aber anders nicht aber das aktuelle Verzeichnis. Es sei denn, das aktuelle Verzeichnis "." wurde PATH hinzugefügt, was aber aus Sicherheitsgründen nicht gemacht wird. Um beispielsweise das Programm "demo" aufzurufen, das sich im aktuellen Verzeichnis befindet, muss beim Aufruf explizit der Verweis auf das aktuelle Verzeichnis in Form von "./" mit angegeben werden. Das Programm "demo" ist daher als "./demo" aufzurufen.

Die Standardkommandos wie z.B. "ls" können ohne Pfadangabe aufgerufen werden, da sie in einem der durch die Umgebungsvariable "PATH" definierten Pfade liegen.

- **Aufruf von Programmen (Ausführungsrechte)**

Um ein Programm unter Linux ausführen zu können, muss die zugehörige Datei explizit als ausführbar gekennzeichnet sein. Dafür verantwortlich ist das "x"-Flag in den Dateiattributen ("eXecuteable"), die beim Aufruf von "ls -l" angezeigt werden, z.B.:

```
ls -l ./mountnfs.sh
-rwxr-xr-x  1 1000      users          2236 Jan 21  2009 ./mountnfs.sh
```

Ist dieses Flag nicht gesetzt, ist die Datei als nicht ausführbar gekennzeichnet und das betreffende Kommando kann nicht aufgerufen werden. Dies ist z.B. generell nach dem FTP-Download einer Datei der Fall. Das "x"-Flag kann jedoch mit Hilfe des Kommandos "chmod" (change mode) wieder gesetzt werden:

```
chmod +x ./mountnfs.sh
```

Das Kommando "*chmod*" ermöglicht generell die Beeinflussung aller Dateiattribute (sowohl setzen als auch löschen). Details hierzu liefert der Aufruf "*chmod --help*".

- **Automatische Vervollständigung von Eingaben durch TAB-Taste**

Ein äußerst praktisches Feature unter Linux ist die automatische Vervollständigung von Datei- bzw. Kommandonamen mittels TAB-Taste. Damit werden so viele Zeichen vervollständigt, wie sie eindeutig sind. Falls eine Eingabe noch nicht vervollständigt werden konnte, können durch doppeltes Drücken der TAB-Taste die möglichen Kandidaten angezeigt werden.

Beispiel: Nach der Eingabe von "l" und dem doppelten Betätigen der TAB-Taste werden alle Programme, die mit "l" beginnen aufgelistet. Nach der weiteren Eingabe von "o" wird durch TAB auf "log" erweitert, da es keine andere Möglichkeit gibt (siehe Liste nach 1. Eingabe im Bild 59). Bei der weiteren Eingabe von "g" (Es steht nun "logg" in der Eingabe) vervollständigt die TAB-Taste die Eingabe auf "logger", da es keine weiteren Kandidaten gibt. Bild 59 zeigt das komplette Beispiel.

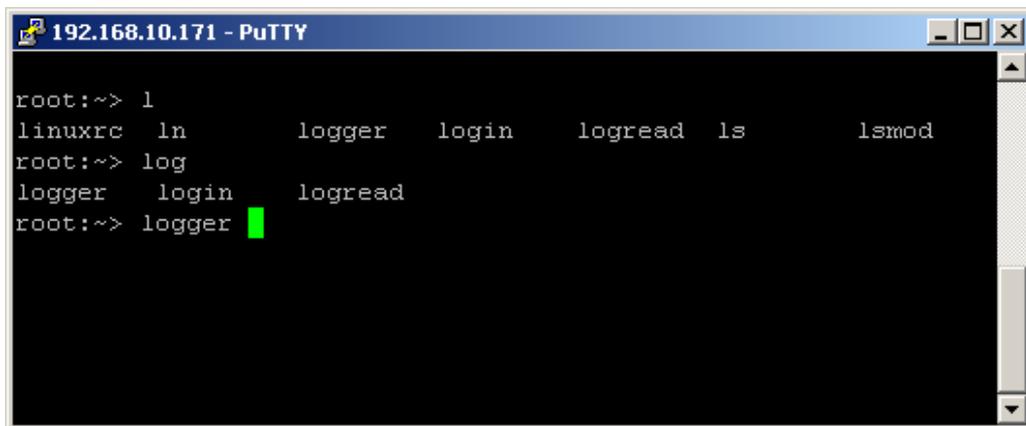


Bild 59: Tabcompletion in der Shell

- **Anzeigen von Umgebungsvariablen**

Die Anzeige von Umgebungsvariablen ist mit Hilfe des Shellkommandos "*echo*" (Ausgabe des nachfolgenden Ausdrucks) möglich. Umgebungsvariablen werden in der Shell durch das Voranstellen des Dollarzeichens "\$" angesprochen. So erfolgt beispielsweise die Anzeige des aktuell konfigurierten Suchpfades mit Hilfe von "*echo \$PATH*".

- **Setzen von Umgebungsvariablen**

Beim Setzen einer Umgebungsvariablen ist zu beachten, dass diese nur innerhalb der aktuellen Shell-Instanz sichtbar ist. Wird eine Umgebungsvariable beispielsweise in einem Shell-Script definiert, ist sie nach dessen Ausführung nicht mehr (neu) definiert. Der Grund hierfür ist, dass zur Ausführung eines Scripts eine neue Shell-Instanz aufgerufen wird, die das Script abarbeitet. Innerhalb dieser Shell-Instanz kann auf die betreffende Umgebungsvariable zugegriffen werden. Nach der Abarbeitung des Scripts wird die eigens hierfür gestartete Shell-Instanz jedoch wieder beendet und damit auch die darin definierte Umgebungsvariable bzw. deren Änderung verworfen.

Eine Möglichkeit, Umgebungsvariablen (z.B. "*TZ=*") dauerhaft zu setzen, besteht darin, diese im Script "*/home/etc/profile*" zu definieren. Dieses Script wird einmalig beim Start der Login-Shell ausgeführt, so dass die darin definierten Variablen während der gesamten Shell-Sitzung erhalten bleiben. Um Umgebungsvariablen an eine Sub-Shell weiterzuleiten, muss diese exportiert werden. Um z.B. die "*TZ*"-Variable an alle Sub-Prozesse weiterzuleiten, muss einmalig "*export TZ*" ausgeführt werden, was auch gleich bei der Wertzuweisung der Variable geschehen kann.

- **Hilfe zu einem Programm**

Unter Linux kann eine kurze Formatbeschreibung zur Parametereingabe in der Regel durch Eingabe des Programmnamens gefolgt von "--help" aufgerufen werden (z.B. "mount --help").

Alternativ kann auf dem Linux-Entwicklungssystem auch die sogenannte Manpage (Manual page) aufgerufen werden. Dazu ist der Befehl "man" gefolgt von dem Programm dessen Manpage geöffnet werden soll, z.B. "man ls". Es kann mit den Pfeiltasten nach oben und unten geblättert werden. Zum Verlassen muss nur die Taste "q" (quit) betätigt werden. Alternativ sind sehr viele Manpages auch im Internet in HTML-Seiten zu finden.

- **Fehlersuche in Shell-Skripten**

Zur Vereinfachung der Fehlersuche bei der Ausführung von Shell-Skripten kann das zu untersuchende Skript mit dem Befehl "sh -x <script_file>" aufgerufen. Die Option "-x" weist die Shell an, jede ausgeführte Zeile auf der Konsole auszugeben. Dadurch kann z.B. leicht nachvollzogen werden, welche Zweige einer bedingten Ausführung ("if", "case", "while" usw.) tatsächlich ausgeführt wurden.

- **Setzen der Zeitzone auf dem ECUcore-5208**

Das Setzen der Zeitzone auf dem ECUcore-5208 erfolgt durch die Definition der Umgebungsvariablen "TZ" im Startscript "/home/etc/profile". Die entsprechende Definition für Deutschland (UTC +1:00) mit Angabe von Beginn und Ende der Sommerzeit ist dort bereits als Kommentar enthalten. Für andere Zeitzonen ist die Definition entsprechend anzupassen.

- **Stackgröße von Programmen**

Die Programme, die für uClinux erstellt werden, haben eine Standardstackgröße von 4 KiB. Da allerdings keine MMU vorhanden ist, besteht keine Möglichkeit mit der Situation umzugehen, falls der Stack zu klein ist.

Falls es zu Abstürzen eines Programmes oder des Betriebssystems kommt, könnte ein zu kleiner Stack einer Anwendung die Ursache sein. Zum Vergrößern des Stacks wird das Programm **flthdr** der Toolchain verwendet. Ein Beispielaufwurf ist:

```
m68k-uclinux-flthdr -s 20480 romfs/bin/busybox
```

Dies setzt die Stackgröße des Programmes busybox aus dem Verzeichnis romfs/bin (relativ zum aktuellen Verzeichnis) auf 20 KiB.

Anhang A: GNU GENERAL PUBLIC LICENSE

Das auf dem ECUcore-5208 eingesetzte uClinux 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.

Zusätzliche SYSTEC-Systemsoftware sowie vom Anwender entwickelte 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/etc/autostart.....	18
A	
Abmessungen.....	9
adduser.....	26
Administration	
Systemvoraussetzungen.....	15
Ausführungsrechte.....	71
autostart.....	17, 18
B	
boa.....	32
Boot-Bedingungen.....	16
Bootkonfiguration.....	18
C	
CAN0.....	12
candrv.....	52
CAN-REport.....	53
CAN-Treiber.....	52
CE-Konformität.....	6
chmod.....	56, 71
CoLilo.....	18
CoLilo Kommandoprompt	
Aktivierung.....	16
Terminaleinstellungen.....	17
CoLilo Kommandos.....	20
Ethernet Konfiguration.....	19
Update uClinux-Image.....	34
COM0.....	12
COM1.....	12
D	
date.....	27
Dateisystem.....	30
debug.sh.....	64
Debugger-Kommandos.....	67
deluser.....	26
Demoprojekt.....	58
CAN-Treiber.....	52
IO-Treiber.....	51
Developmentboard	
Bedienelemente.....	13
DHCP.....	38
VMware-Image.....	44
E	
Eclipse.....	60
Debuggen.....	62
Debugger konfigurieren.....	64
Debugger-Kommandos.....	67
Projekt öffnen.....	60
Projekt übersetzen.....	62
EMV-Gesetz.....	6
ETH0.....	12
F	
FTP.....	56
FTP-Client.....	15
ftpd.....	58
ftpget.....	57
ftpput.....	57
FTP-Server.....	58
G	
gbrdcfg.....	28
gdbserver.....	63
GNU.....	9
GPL.....	74
H	
Hardwareanschaltung testen.....	70
hellocan.....	52
HTTP-Server.....	32
hwclock.....	27
I	
I/O-Treiber.....	50
insmod.....	51, 52
iodrvdemo.....	70
K	
Konfiguration	
auslesen und anzeigen.....	28
Kommandos.....	19
Konfiguration, Kommandos.....	20
Konfigurationsmodus.....	16
L	
Linux-BSP.....	68
M	
make.....	58
Makefile.....	49
Manuale	
Übersicht.....	7
mount.....	55
N	
net use.....	40
NFS.....	55
Nutzerkonten	
Anlegen und Löschen.....	26
Passwort ändern.....	27
vordefinierte.....	26
P	
passwd.....	27
PATH.....	71
pc5208drv.....	50

R

RTC setzen.....27

S

setup-ecucore-5208.sh.....32

SO-1096.exe36

Softwareupdate

 uClinux-Image 33

Suchpfad.....71

Systemstart.....17

Systemzeit setzen27

T

TAB-Completion72

Telnet-Client15

Terminaleinstellungen17

Terminalprogramm15

TFTPD3233

TZ73

U

Umgebungsvariablen

 anzeigen 72

 setzen 72

USB-RS232 Adapter Kabel 14

V

VMware-Image

 Installation 36

 statische IP-Adresse festlegen..... 44

VMware-Player

 Installation 36

vordefinierte Nutzerkonten..... 26

W

WinSCP 24

Z

Zeitzone 73

Zubehör..... 14

Dokument: Developer Manual ECUcore-5208
Dokumentnummer: L-1202d_1, 1. Auflage August 2009

Wie würden Sie dieses Handbuch verbessern?

Haben Sie in diesem Handbuch Fehler entdeckt? Seite

Eingesandt von:

Kundennummer: _____

Name: _____

Firma: _____

Adresse: _____

Einsenden an: SYS TEC electronic GmbH
August-Bebel-Str. 29
D - 07973 Greiz
GERMANY
Fax : +49 (0) 36 61 / 6279-99