

Quick Start Instructions

Linux-Kit

phyCORE-Z500PT

Using Eclipse and the GNU Cross Development Tool Chain

Note: The PHYTEC Linux-Z500PT-Disc includes the electronic version of the English phyCORE-Z500PT Hardware Manual.

Edition: August 2009

A product of a PHYTEC Technology Holding company

In this manual copyrighted products are not explicitly indicated. The absence of the trademark (™) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, PHYTEC Messtechnik GmbH assumes no responsibility for any inaccuracies. PHYTEC Messtechnik GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. PHYTEC Messtechnik GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages that might result.

Additionally, PHYTEC Messtechnik GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. PHYTEC Messtechnik GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

© Copyright 2009 PHYTEC Messtechnik GmbH, D-55129 Mainz.

Rights - including those of translation, reprint, broadcast, photomechanical or similar reproduction and storage or processing in computer systems, in whole or in part - are reserved. No reproduction may be made without the explicit written consent from PHYTEC Messtechnik GmbH.

	EUROPE	NORTH AMERICA
Address:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 55129 Mainz GERMANY	PHYTEC America LLC 203 Parfitt Way SW, Suite G100 Bainbridge Island, WA 98110 USA
Ordering Information:	+49 (800) 0749832 order@phytec.de	1 (800) 278-9913 sales@phytec.com
Technical Support:	+49 (6131) 9221-31 support@phytec.de	1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	1 (206) 780-9135
Web Site:	http://www.phytec.de	http://www.phytec.com

1st Edition: August 2009

Chapter 1 Introduction..... 1

1.1	Rapid Development Kit Documentation	1
1.2	Professional Support Packages Available	1
1.3	Overview of this Quick Start Manual.....	2
1.4	Conventions Used in this Quick Start Manual	3
1.5	System Requirements	4
1.6	Software Development Tool Chains	4
	Eclipse.....	4
	The GNU Cross Development Tool Chain	5



TIME



5 min**Chapter 2 Getting Started..... 7**

2.1	Requirements of the Host Platform	7
2.2	Configuring the Host Platform	8
	Installing Software Packages.....	8
	Set Up Network Card Configuration.....	14
	Disabling the Firewall	15
	Set Up TFTP Server	16
2.3	Linux-Z500PT-Kit Setup.....	18
	Starting the Setup.....	19
2.4	Advanced Configuration Information	26
2.5	Creating a bootable USB flash disk.....	27
2.6	Connecting the Host with the Target.....	28
2.7	Copying an Example to the Target.....	34
	Copying a Program to the Target	35
	Using Telnet to Execute a Program on the Target	36
	Using SSH to Execute a Program on the Target	37
2.8	Advanced Information.....	40
	Copying a Program to the Target with the Command Line	40
	Executing a Program on the Target	40
	Executing a Program directly on the Target using SSH.....	40



TIME

35 min

Chapter 3	Getting More Involved	41	
3.1	Configuring and Compiling the Image	41	TIME
	Writing the Image into the Target's solid state disk	46	70 min
3.2	Opening an Existing Project.....	49	
	Copying the <i>HelloWorld</i> Project.....	49	
	Starting Eclipse and Importing the Example Project	50	
3.3	Creating a New Project.....	54	
3.4	Changing the Demo Application.....	64	
	Executing the Program on the Target using Microcom	65	
3.5	Starting a Program out of Eclipse on the Target	66	
3.6	Automatically Starting the Program when Booting the Target	68	
Chapter 4	Debugging an Example Project	73	
4.1	Starting the GDB Server on the Target	73	TIME
4.2	Configuring and Starting the Debugger in Eclipse	74	20 min
4.3	Setting a Breakpoint	80	
4.4	Stepping and Watching Variable Contents	80	
4.5	Changing Variable Values	82	
4.6	Using the Memory Monitor.....	84	
Chapter 5	Further Information	87	
Chapter 6	Summary.....	88	
Appendix A	Updating BIOS on SPI	89	

Chapter 1 Introduction

**5 min**

In this Quick Start you can find general information on the PHYTEC phyCORE-Z500PT, and an overview of the Eclipse software development tool and the GCC C/C++ cross development tool chain. You will also find instructions on how to run example programs on the phyCORE-Z500PT, mounted on the PHYTEC Development Board, in conjunction with the Eclipse development tool.

Please refer to the phyCORE-Z500PT Hardware Manual for specific information on board-level features, such as [jumper configuration](#), [memory mapping](#), and [pin layout](#).

1.1 Rapid Development Kit Documentation

This “Rapid Development Kit” (RDK) includes the following electronic documentation on the enclosed “PHYTEC Linux-Z500PT-Disc:”

- The PHYTEC phyCORE-Z500PT Hardware Manual.
- Z500PT Controller [User's Manuals and Data Sheets](#).
- This Quick Start Instructions with general “Rapid Development Kit” description, software installation advice, and an example program enabling quick out-of-the-box start-up of the phyCORE-Z500PT in conjunction with the Eclipse and GCC C/C++ software development tool chain.

1.2 Professional Support Packages Available

This kit comes with free installation support. If you have any questions concerning installation and setup, you are welcome to contact our support department.

For more in-depth questions, we offer a variety of custom-tailored packages with different support options (e-mail, phone, direct contact to the developer) and different reaction times.

Please contact our sales team to discuss the appropriate support option if professional support beyond installation and setup is important to you.

1.3 Overview of this Quick Start Manual

This Quick Start manual gives you a general “Rapid Development Kit” description, as well as software installation advice and an example program enabling quick out-of-the-box start-up of the phyCORE-Z500PT in conjunction with the Eclipse IDE and GCC C/C++ software tools. It is structured as follows:

- The “*Getting Started*” chapter describes the configuration of the host platform and how to setup all the tools used in this manual.
- The “*Getting More Involved*” chapter provides step-by-step instructions on how to configure and build a new kernel, modify an example application, create and build new projects, and copy programs to the phyCORE-Z500PT using Eclipse.
- The “*Debugging*” chapter provides information on how to debug an application with the Eclipse debugging interface.

In addition to the dedicated data for this Rapid Development Kit, the PHYTEC Linux-Z500PT-Disc contains supplemental information on embedded microcontroller design and development in general.

1.4 Conventions Used in this Quick Start Manual

The following is a list of the typographical conventions used in this Quick Start manual:

Italic Used for file and directory names, program and command names, command-line options, menu items, URLs, and other terms that correspond the terms on your desktop.

Bold Used in examples to show commands or other text that should be typed literally by the user.

Pay attention to notes set apart from the text with the following icons:



At this part you might leave the path of this Quick Start.



This is a warning. It helps you to avoid annoying problems.



Provides useful supplementary information about the topic.



At the beginning of each chapter you can find information of the time needed to pass that chapter.



You have successfully passed an important part of this Quick Start manual.



Provides information to solve common problems.

1.5 System Requirements

Use of this “Rapid Development Kit” requires:

- The PHYTEC phyCORE-Z500PT.
- The PHYTEC Development Board with the included DB-9 serial cable, Ethernet cross-over cable and AC adapter supplying 5 VDC (min. 2 A).
- PHYTEC Linux distribution based on OSELAS from Pengutronix.
- An IBM-compatible host PC (586 or higher CPU).
- openSUSE 11.0 (x86) and the KDE desktop.
- Recommended free disk space: at least 2 GB.
- Optional: DVI display/monitor

For more information and updates, please refer to the following sources:



<http://www.phytec.de>

support@phytec.de

1.6 Software Development Tool Chains

Eclipse

The Eclipse platform provides support for C/C++ development. Because the Eclipse platform is only a framework for developer tools, it doesn't support C/C++ directly; instead it uses external plug-ins. This Quick Start shows you how to make use of the *CDT*, a set of plug-ins for C/C++ development in conjunction with the GCC C/C++ tool chain.

The CDT is an open source project (licensed under the Common Public License) implemented purely in Java as a set of plug-ins for the Eclipse SDK platform. These plug-ins add a C/C++ perspective to the Eclipse

Workbench that can now support C/C++ development with a number of views and wizards, along with advanced editing and debugging support.

Due to its complexity, the CDT is broken down into several components, which take the form of separate plug-ins. Each component operates as an autonomous project, with its own set of committers, bug categories, and mailing lists. However, all plug-ins are required for the CDT to work properly. Here is a list of the plug-ins/components:

- **Primary CDT plug-in** is the “framework” for the CDT plug-ins.
- **CDT Feature Eclipse** is the CDT Feature Component.
- **CDT Core** provides Core Model, CDOM, and Core Components.
- **CDT UI** is the Core UI, views, editors, and wizards.
- **CDT Launch** provides the launch mechanism for external tools such as the compiler and debugger.
- **CDT Debug Core** provides debugging functions.
- **CDT Debug UI** provides the user interface for the CDT debugging editors, views, and wizards.
- **CDT Debug MI** is the application connector for MI-compatible debuggers.

The GNU Cross Development Tool Chain

Cross development in general refers to the overall software development process that produces a single application or a complete system running on a platform that is different from the development platform. This is an important concept when the target system doesn't have a native set of compilation tools, or when the host system is faster and has greater resources.

The platform where the actual development takes place is called the *host platform*. The platform where the final application is tested and run is

called the *target platform*. In this Quick Start we are using an x86-based Linux as the host platform. As the target platform we are using the i586 architecture with an Z500PT CPU.

Building a program for a CPU architecture different from the one used on the machine where the compilation is done is accomplished using a cross compiler tool chain and cross-compiled libraries. In this Quick Start we are using the GNU C/C++ cross development tool chain.

Chapter 2 Getting Started

**35 min**

TIME

In this chapter you will establish the basis to pass the steps in this Quick Start. First you will learn how to configure the host platform. You will install additional software packages and setup the network configuration for connecting your host to the target. After connecting the host to the target, you will copy an application to the target. At the end of this chapter you will be able to start a first demo application on the target.

2.1 Requirements of the Host Platform

To pass the following steps in this Quick Start, you will need a host PC with an installation of openSUSE 11.0 (x86) and the KDE desktop.

When you are installing openSUSE 11.0, you can select *KDE* as *Desktop selection*. The default packages to use openSUSE 11.0 with your host PC will be selected automatically. This default selection will suffice to pass the steps in these Quick Start Instructions. The installation of additional packages and configurations will be described on the following pages.

In the following configuration steps we assume that the host PC is not connected to any other network. The target and host will be connected with a cross-over cable via a peer-to-peer connection. If your host is part of a company's network, we recommend disconnecting your host from such a network.

In these Quick Start Instructions you will have to shutdown the firewall and configure the network card of your host PC. If your host PC is connected to another network, changing the IP address can cause conflicts with existing hosts.

2.2 Configuring the Host Platform

In this passage you will learn how to configure the host platform. You will execute the following steps:

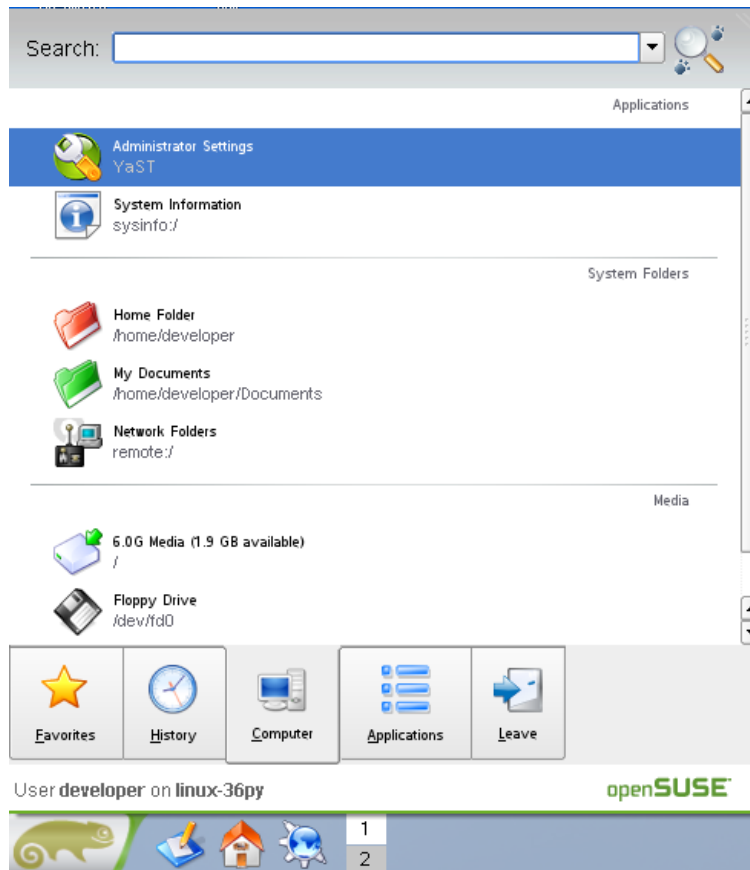
- Install additional software packages. These packages are necessary to accomplish the steps in the Quick Start Instructions.
- Set up the network configuration to use the host PC with your target.
- Disable the firewall. If the firewall is enabled, you will have problems with connecting to the target.
- Set up a TFTP server. You can use a TFTP server to download files (e.g. kernel and root file system images) to the target from within the target's boot loader.

Installing Software Packages

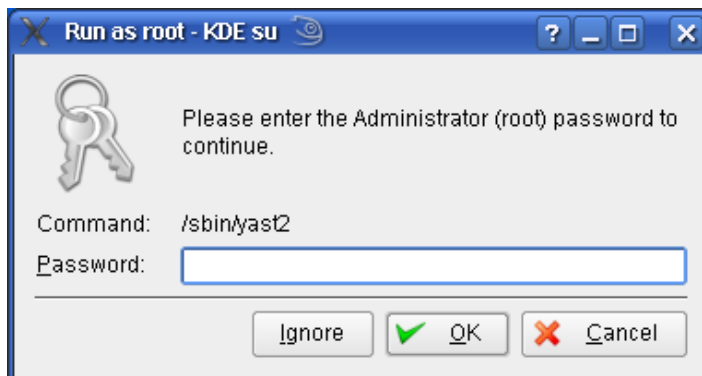
To accomplish the steps in the Quick Start Instructions, you will have to install additional packages.



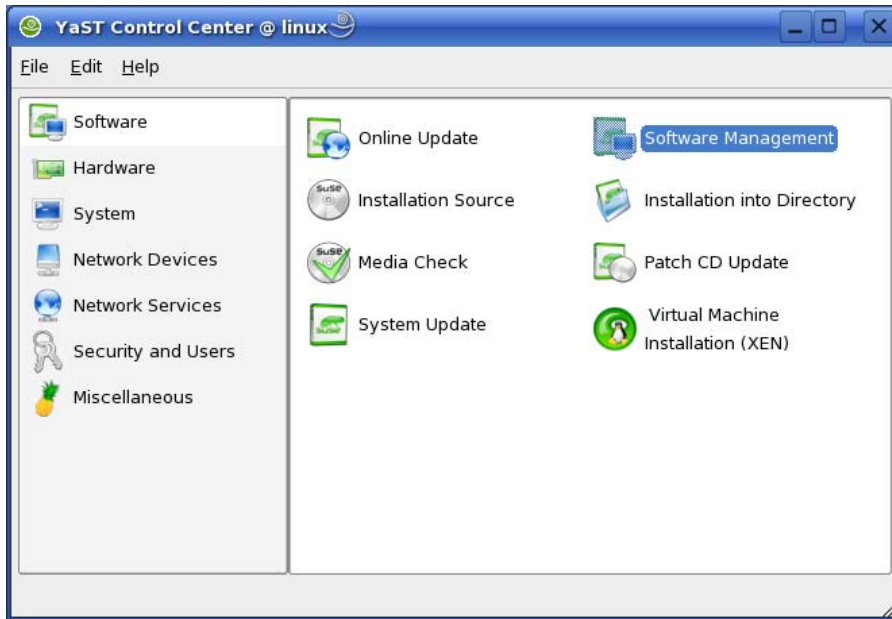
If you don't install all of these packages, the setup may fail or some configuration steps won't work correctly.



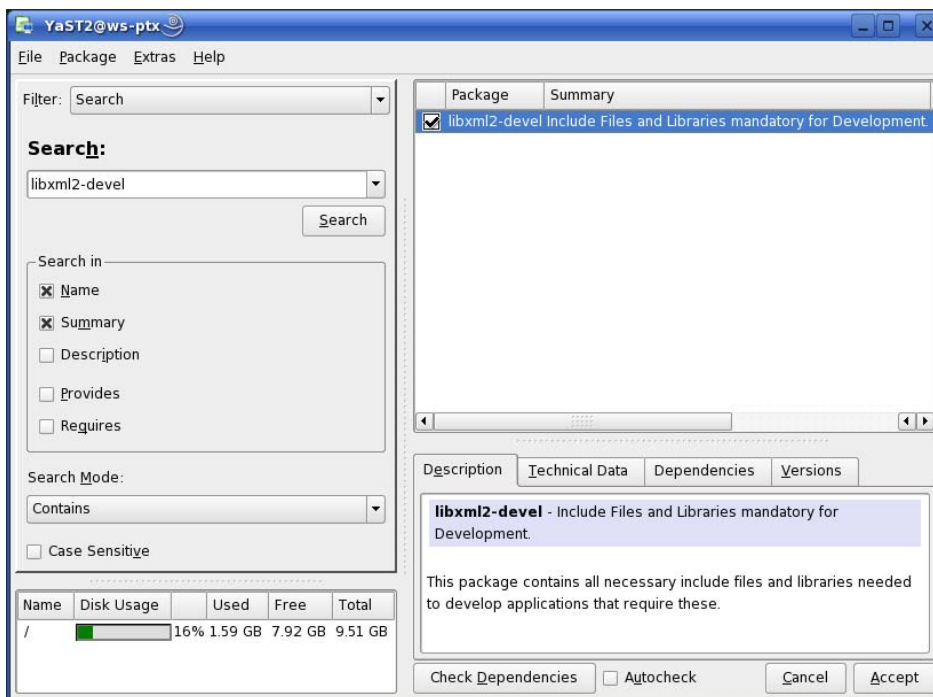
- Open the *K menu* from the lower-left corner of the desktop and click on the tab *Computer*.
- Open the *Administrator Settings / YaST*.



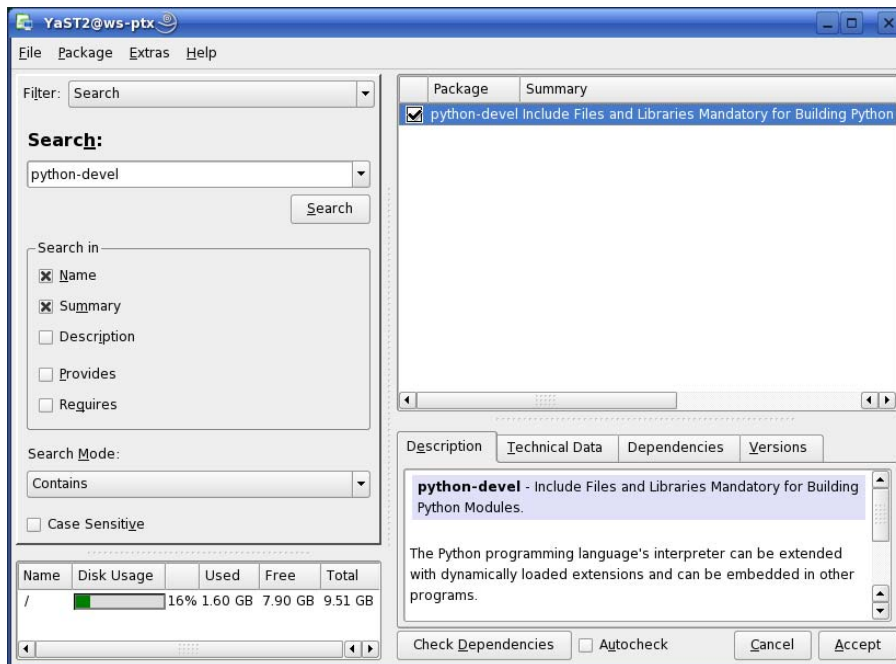
- Enter your root password and click *OK*.



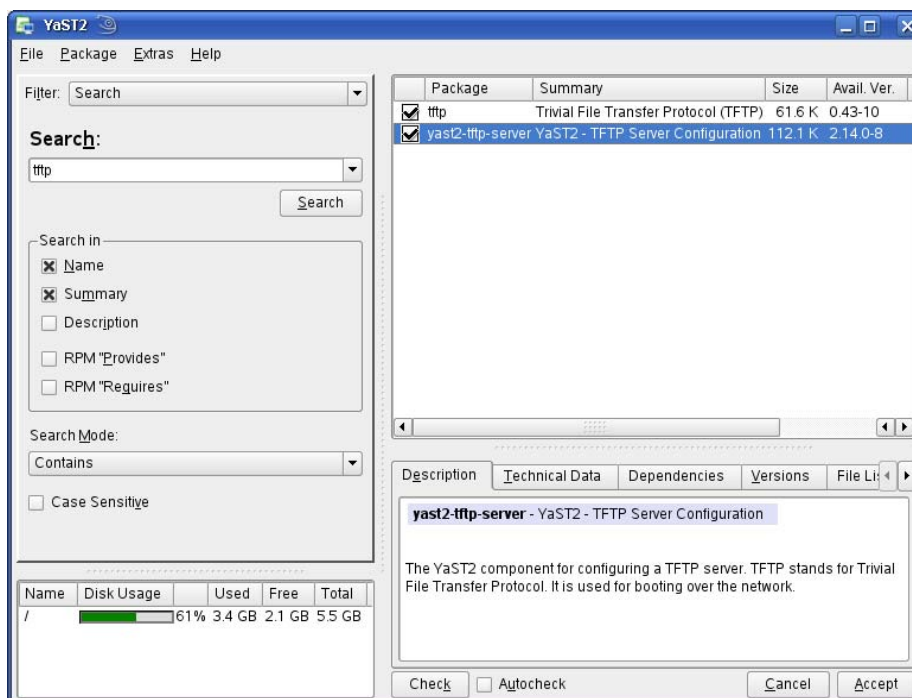
- Open *Software Management* in *Software*.



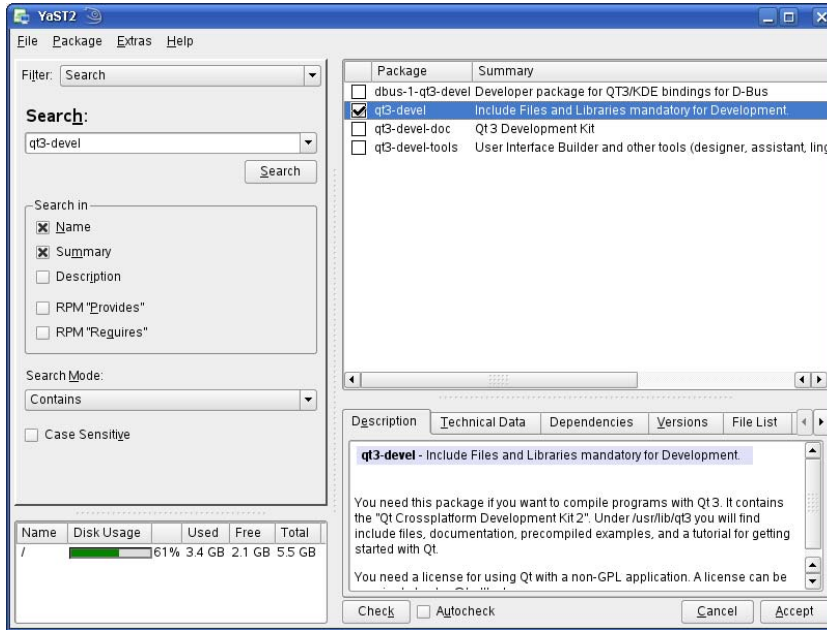
- Select the filter *Search*.
- Type **libxml2-devel** and click the *Search* button.
- Check *libxml2-devel*.



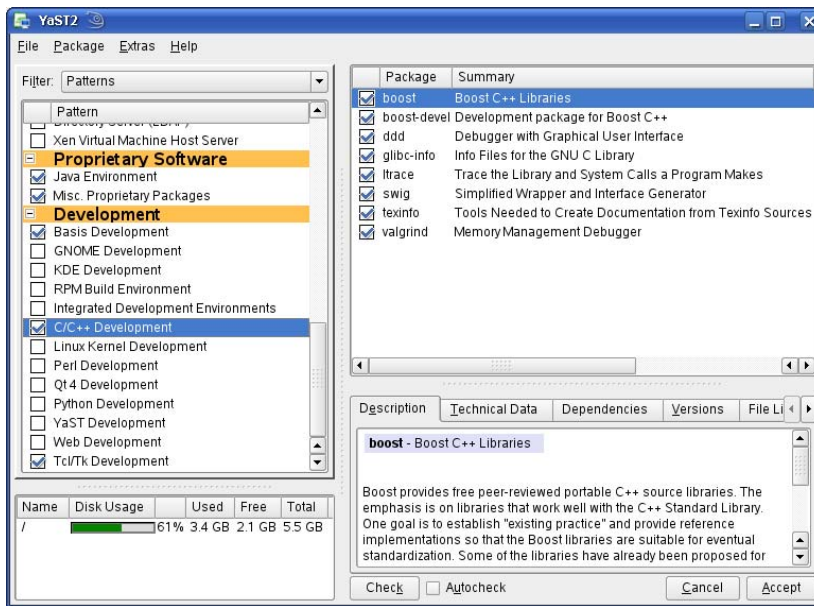
- Type **python-devel** and click the *Search* button.
- Check *python-devel*.



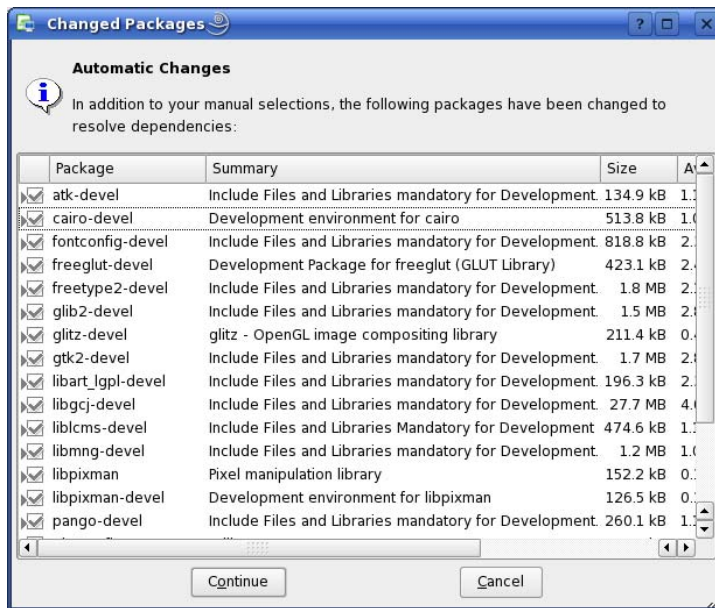
- Type **tftp** and click the *Search* button.
- Check the packages *tftp* and *yast2-tftp-server*.



- Type **qt3-devel** and click the *Search* button.
- Check *qt3-devel*.



- Select the filter *Patterns*.
- Select *Base Development*, *C/C++ Development*, and *Tcl/Tk Development*.
- Click *Accept*.

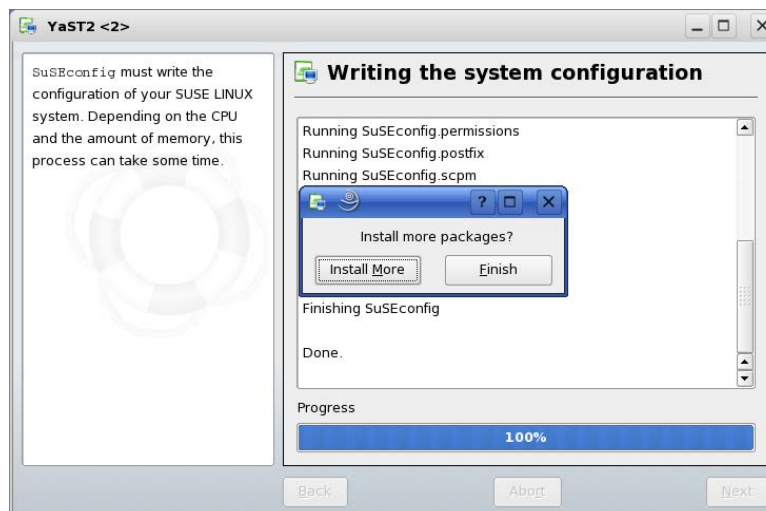


Some additional packages will be selected automatically to resolve any dependencies.



If problems occur while resolving dependencies, we recommend going back to a default configuration.

- Click *Continue* to install the packages.



- Click *Finish*.

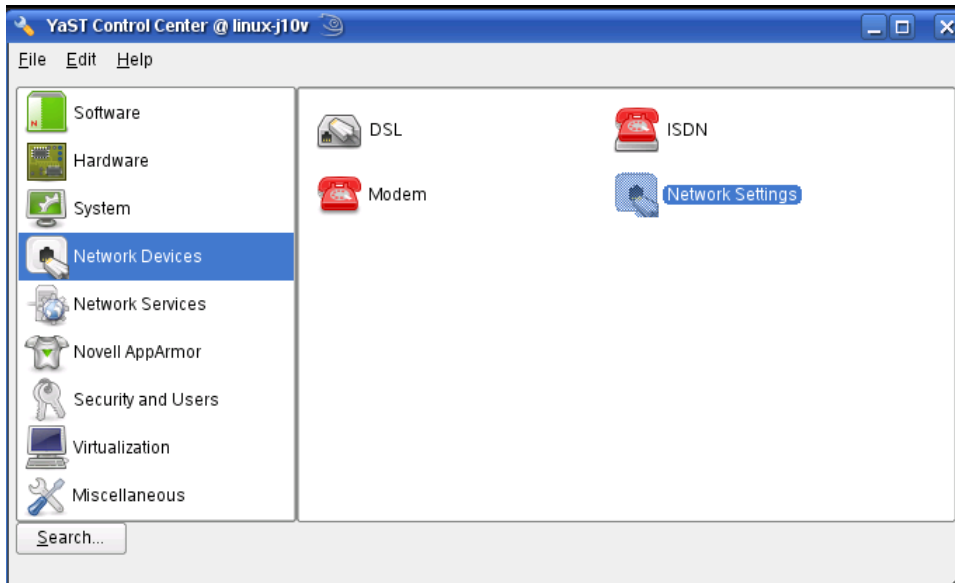
Set Up Network Card Configuration



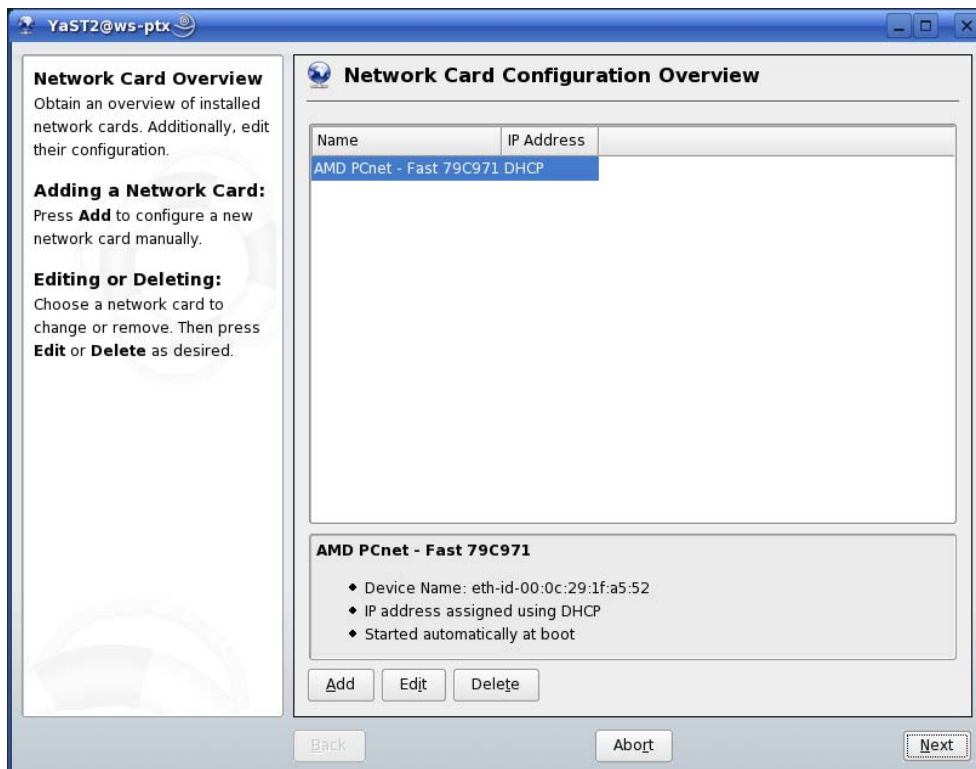
CAUTION

In the following steps you will have to configure the IP address of your host. We recommend disconnecting your host from any other network. If you change the host's IP, chances are that problems may occur with other hosts in the network.

- Open the *YaST Control Center* if it is not already opened.



- Choose *Network Settings* in *Network Devices*.

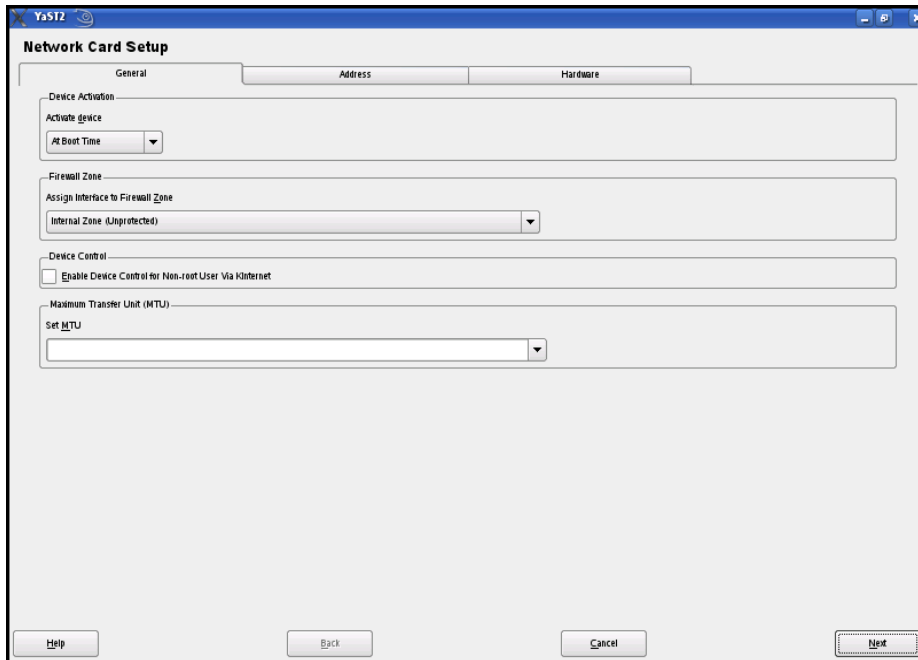


- Select the right network card (if more than one network card is installed on your host).
- Click *Edit* to enter the *Network Card Setup*.
- Choose *Static address setup*.
- Enter IP address **192.168.3.10** and subnet mask **255.255.255.0**

Disabling the Firewall

To ensure that there are no problems with connections to the target, the host's firewall should be disabled.

- Select the *General* tab in the upper-left corner.



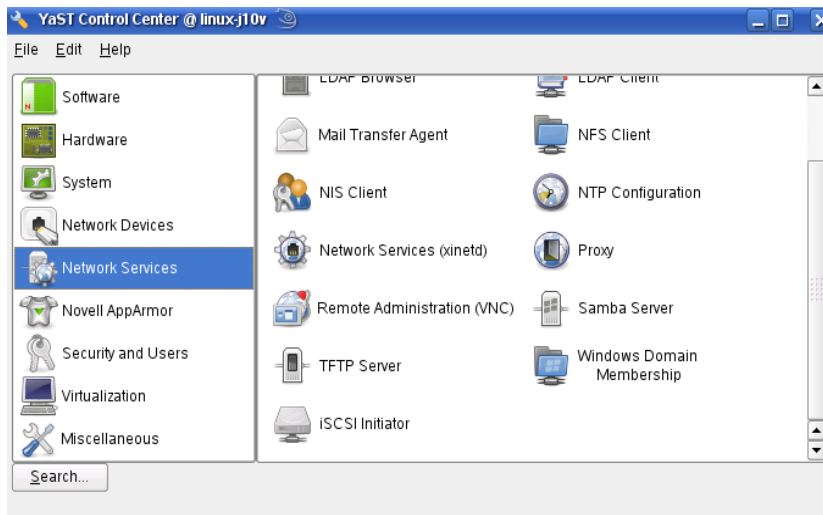
- Use the drop-down box in the *Firewall Zone* settings to set the current interface to *Internal Zone (Unprotected)*.
- Then press *Next*, and in the following window click *Finish* to complete the settings.

The firewall is now disabled for this network card.

Set Up TFTP Server

Later in this Quick Start you will learn how to write a new kernel image into the flash memory of the target. To download the kernel image from the target, you need have to have a TFTP server running. In this passage we show you how to configure a TFTP server.

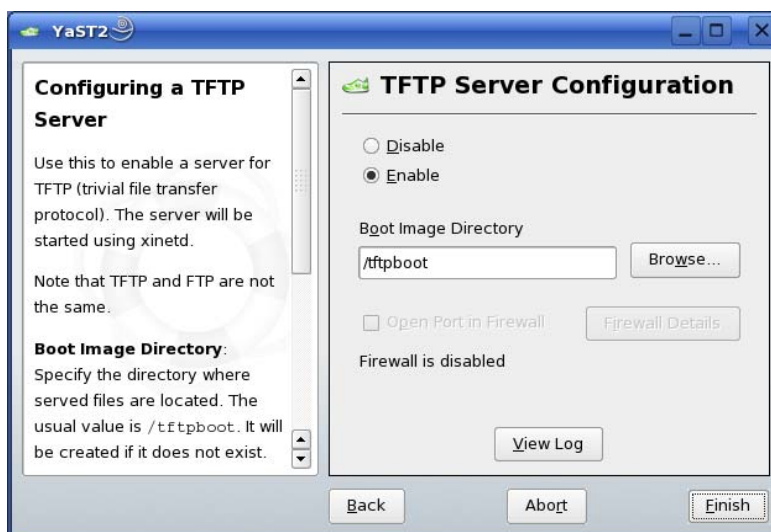
- Open the *YaST Control Center* if it is not already opened.



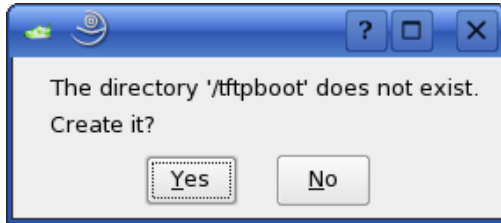
- Choose *TFTP Server* in *Network Services*.



If the *TFTP Server* icon does not exist, restart the YaST Control Center.



- Switch the selection to *Enable*.
- The path of the boot image directory should be */tftpboot*. If there is a different path, change it to */tftpboot*.
- Click *Finish*.



- Click *Yes* to create the */tftpboot* directory.

The TFTP server will be started.

- Close the YaST Control Center.



You have successfully finished the configuration of the host platform.

2.3 Linux-Z500PT-Kit Setup

In this section you will find a description of the Linux-Z500PT-Kit setup. The whole setup will be done by a graphical interface. At the end of the setup you will find all programs to develop applications for the target on your host PC.

The setup will install the following programs:

- *GNU C/C++ cross development tool chain* – you can use this tool chain to develop programs for the target on your host PC.
- *Eclipse SDK with CDT* – the Eclipse SDK is a platform and application framework for building software which can use the GNU C/C++ cross development tool chain.
- *Microcom* – a program for serial communication with the target.
- *HelloWorld* – this example program can be used to test how to download and execute a program on the target.
- *Linux Image archive* – this image archive contains different image source as well as need to boot the phyCORE-Z500PT.

- *OSELAS.BSP* and *PTXdist* – this program will be used to create the kernel image file for the target.

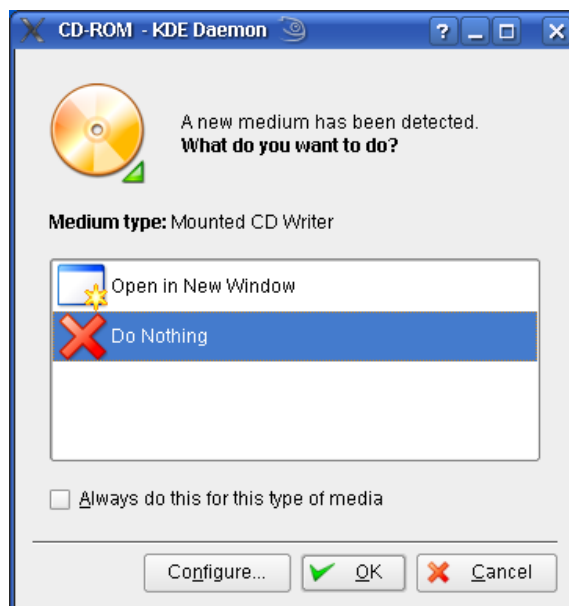
There will be some additional configuration steps performed on your PC:

- The setup program will create desktop links to the installed programs.
- The setup will also create desktop links to access the target via FTP, SSH, and Telnet.
- The path of the cross development tool chain will be added to the \$PATH environment variable.
- Read and write access to the serial interface will be added to your user account so you use the serial communication program Microcom.
- The setup will configure Microcom.

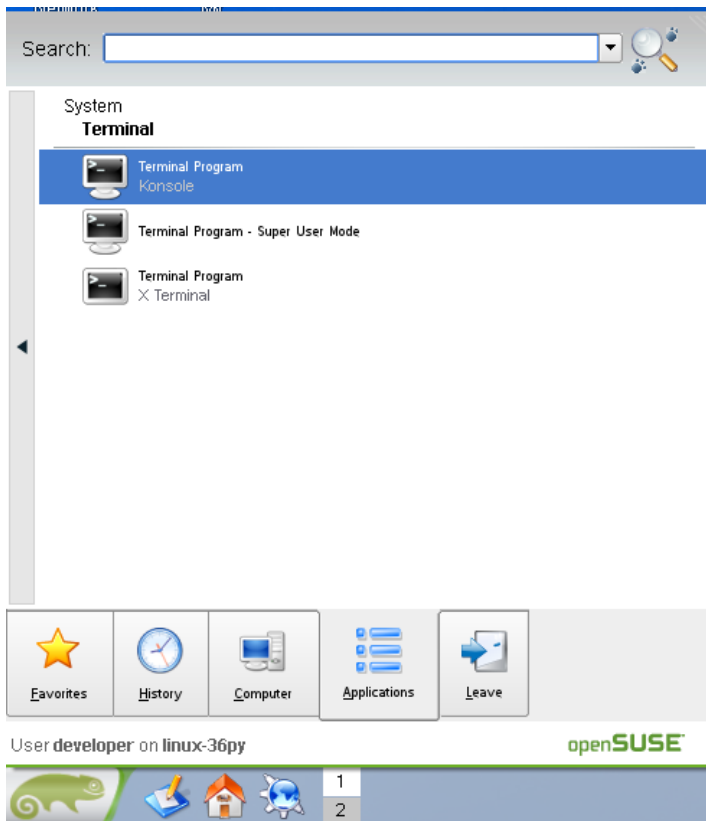
Starting the Setup

- To start with the Linux-Z500PT-Kit setup, put your PHYTEC Linux-Z500PT-Disc into your CD-ROM drive.

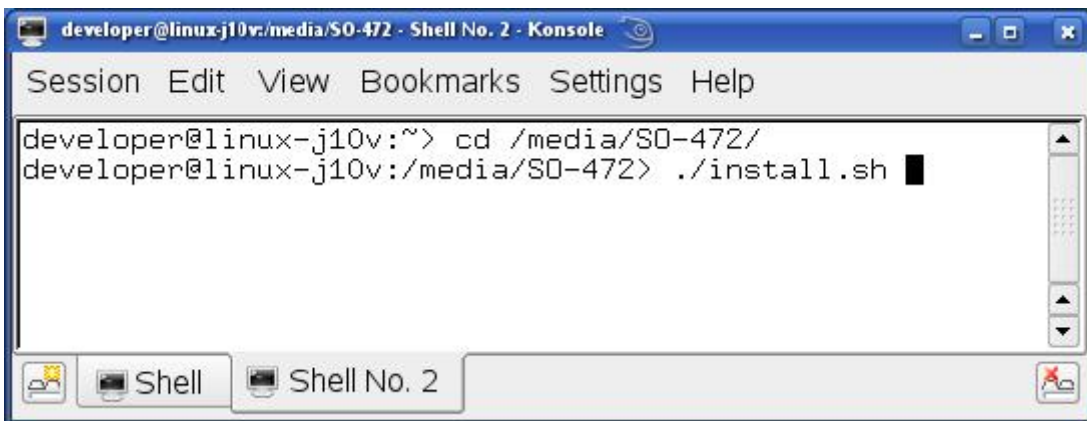
The following dialog may appear:



- Click *Cancel*.



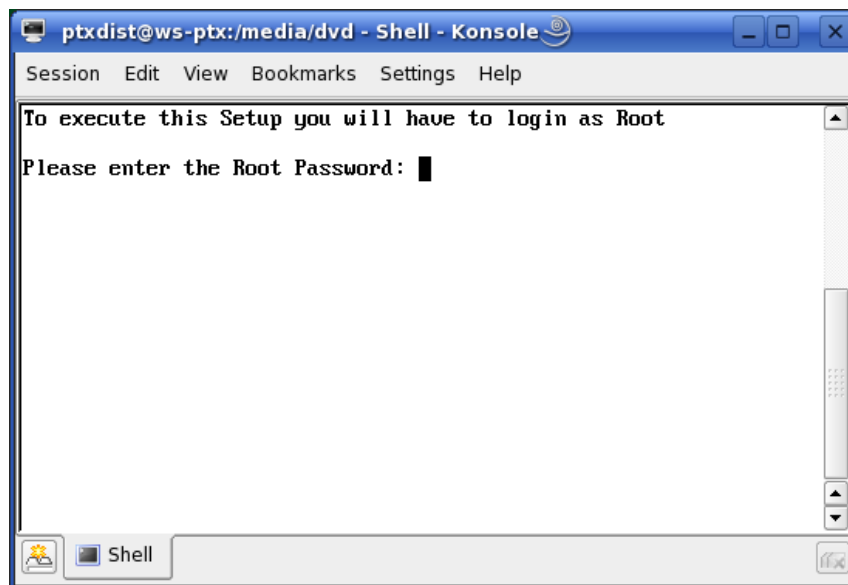
- From the *K menu*, select the *Applications* tab.
- Select *System* ► *Terminal* ► *Terminal Program / Konsole*.



- Type: **cd /media/SO-472**
- Enter **./install.sh** to launch the setup program.



The media may be mounted on a different mount point in the directory */media*. The mount points can be shown with the command **ls /media**. Change to the accordant directory if no directory *SO-472* should exist.

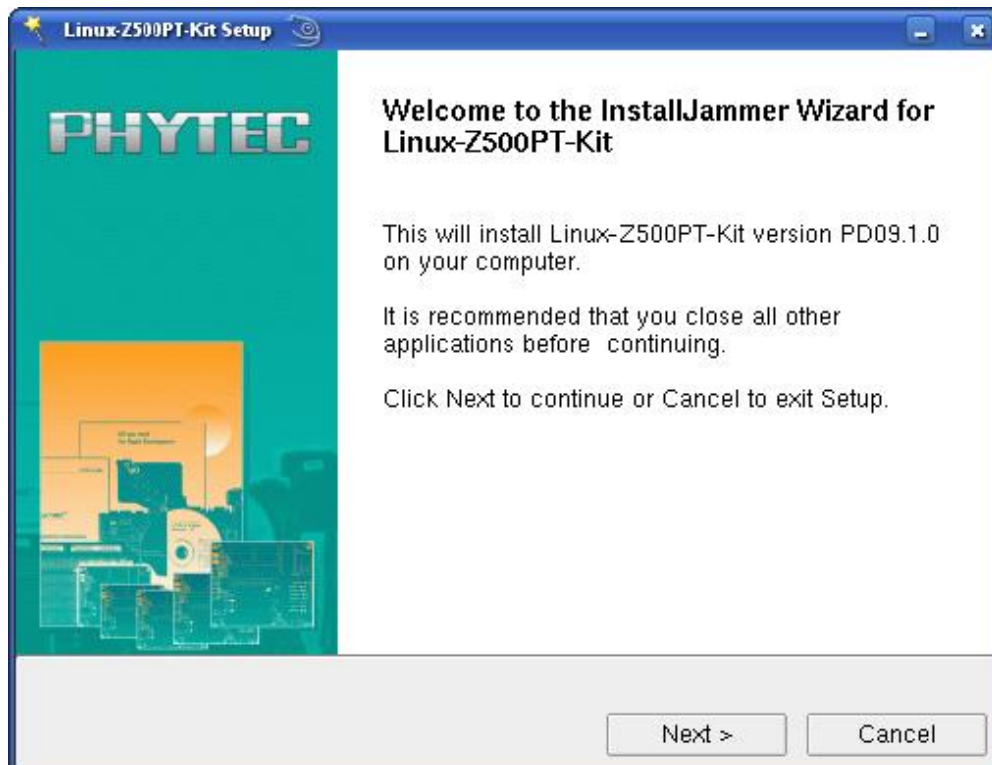


- Enter the root password.

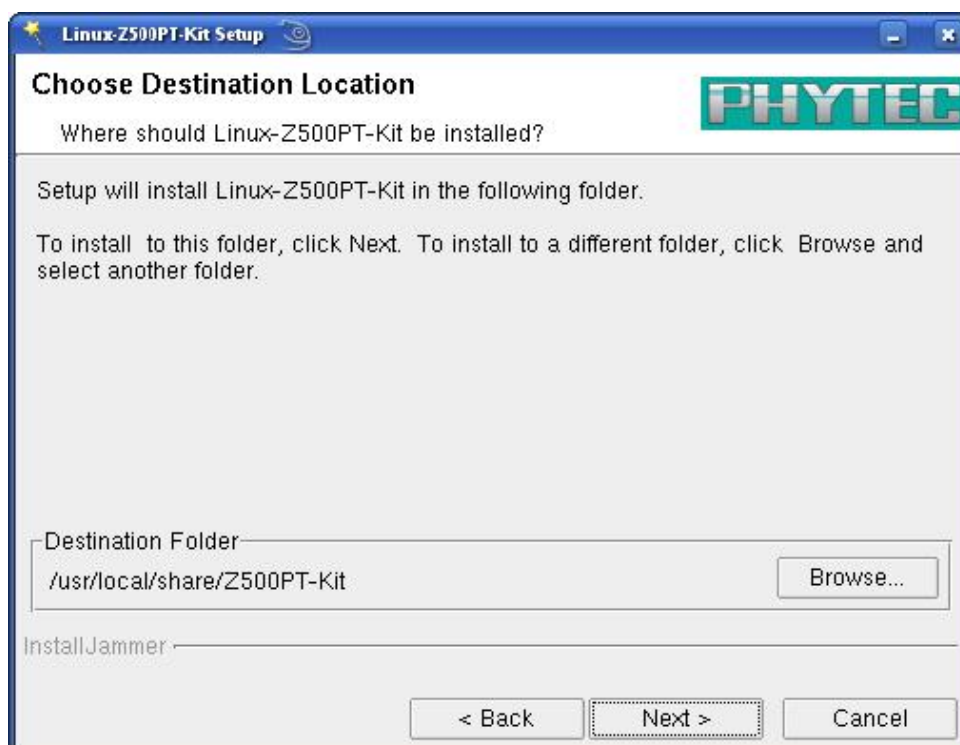


- Click *Yes* to proceed.

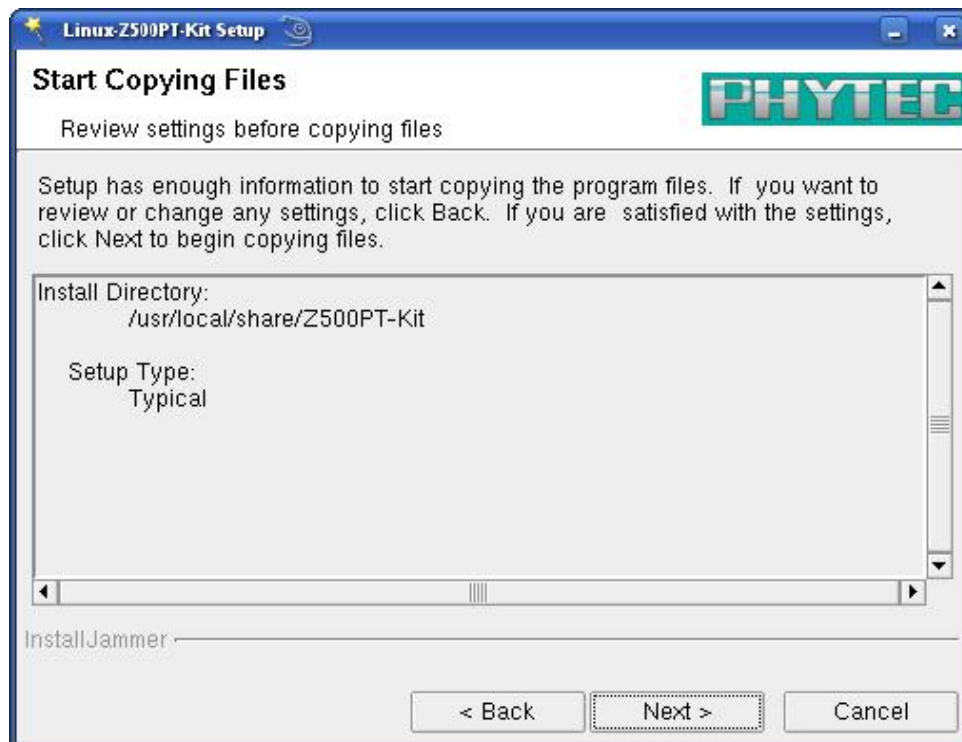
The welcome screen appears.



- Click *Next* to continue.



- Click *Next*.



- Click *Next* to copy all files to your hard disk.



OPTION

The default destination location is `/usr/local/share/Z500PT-Kit`. All path and file statements within this Quick Start manual are based on the assumption that you accept the default installation paths. If you decide to individually choose different paths, you must consider this for all further file and path statements when working with this Quick Start.

We strongly recommend accepting the default destination location.



The GCC C/C++ tool chain will be installed to the default directory `/opt/OSELAS.Toolchain-1.99.3/i586-unknown-linux-gnu`. The program `ptxdist` will be installed to `/usr/local/lib/ptxdist-1.99.12/` and the `OSELAS.BSP` to your home directory. All other programs and examples will be installed to the selected destination directory.

After the files have been copied, a dialog box for the Eclipse installation will appear.

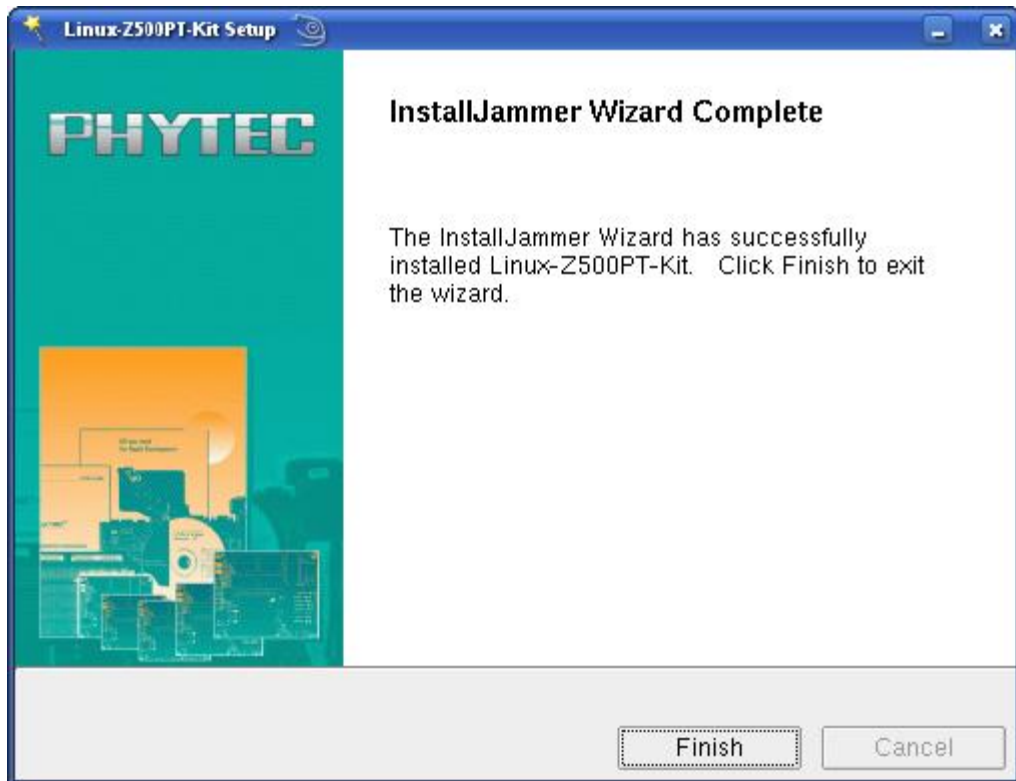
- Click *Yes* to install Eclipse. If you want to skip the installation of Eclipse, choose *No*.



CAUTION

We recommend installing Eclipse even if you already have installed Eclipse on your system. The version of Eclipse provided on the setup CD-ROM includes additional plug-ins.

- Click *Next*.



- Click *Finish* to exit the setup.
- Close the terminal window.

Now you will have to restart the KDE desktop.

- Open the *K Menu* from the lower-left corner of the desktop.
- Select the *Leave* tab and choose *Logout*.
- When the display manager appears, enter your login name and password to restart the KDE desktop.



You have successfully installed the software for the Linux-Z500PT-Kit. You can now use the programs you need to develop your own applications for the target on your host system. The setup program did all necessary configurations. In the following passage you can find some advanced configuration information.

2.4 Advanced Configuration Information

In this part you can find some information on how to change the configuration steps of the setup program by your own. The setup program performed all the following configuration steps. The information in this part is for users who want to use the Z500PT-Kit with a Linux distribution other than openSUSE. This is also interesting for users who want to see what configurations the setup program did.

During the setup program, the GCC C/C++ cross compiler was installed in the directory `/opt/OSELAS.Toolchain-1.99.3/i586-unknown-linux-gnu/gcc-4.3.2-glibc-2.8-binutils-2.18-kernel-2.6.27-sanitized/bin`. To start the cross compiler directly from every location of the system, the directory of the cross compiler was added to the `$PATH` environment variable. You can manually add the directory of the cross compiler to the `$PATH` by adding the following line in the file `/etc/profile`:

```
export PATH=/opt/OSELAS.Toolchain-1.99.3/i586-unknown-linux-  
gnu/gcc-4.3.2-glibc-2.8-binutils-2.18-kernel-2.6.27-  
sanitized/bin:“$PATH”
```

You can open a terminal program and use the cross compiler directly from the command line. For example, you can compile a C program with the following command:

```
i586-unknown-linux-gnu-gcc -o HelloWorld HelloWorld.c
```

In the standard configuration only the root user has write access to the serial interface. To use a serial communication tool like Microcom with normal user rights, you have to be a member of the group `uucp`. A user can be added to this group with the following command:

```
groupmod -A username uucp
```

The serial communication program was configured during the setup with the following configuration:

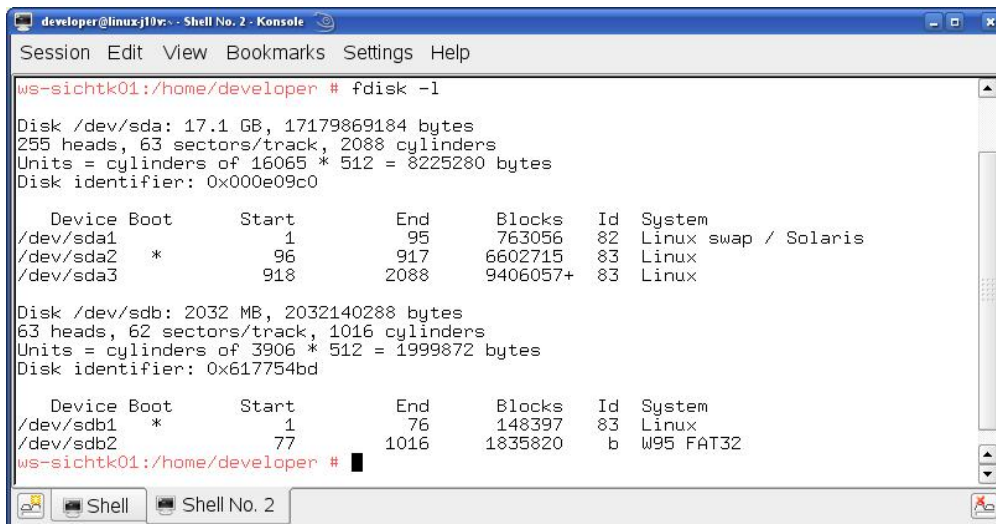
```
115200 baud, 1 start bit, 8 data bits, 1 stop bit, no parity, no flow control.
```

2.5 Creating a bootable USB flash disk

Before we can connect our host with the target we need to create a bootable USB flash disk.

In `/usr/local/share/Z500PT-Kit/` is a folder named *Images*. In this folder we will find the completed image named *hd.img*. This image file can be used to flash a bootable media device.

- First we connect the USB stick with the host and wait until the host has found the new device.
- Now we open a terminal window.
- Type: **su** in the terminal window and enter the root password to gain root access
- Enter: **fdisk -l** to see all connected devices



```

developer@linux-j10v: ~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
ws-sichtk01:/home/developer # fdisk -l

Disk /dev/sda: 17.1 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000e09c0

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            1           95       763056    82  Linux swap / Solaris
/dev/sda2            *          96          917       6602715    83  Linux
/dev/sda3            918        2088       9406057+    83  Linux

Disk /dev/sdb: 2032 MB, 2032140288 bytes
63 heads, 62 sectors/track, 1016 cylinders
Units = cylinders of 3906 * 512 = 1999872 bytes
Disk identifier: 0x617754bd

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            *          1           76       148397    83  Linux
/dev/sdb2            77        1016       1835820    b   W95 FAT32
ws-sichtk01:/home/developer # █

```

- From the output of **fdisk -l** we will have to determine the partition name of our USB stick. We can use the size of the USB Stick for identification. In our example the stick has a capacity of 2032MB and is therefore addressed by the device name `/dev/sdb`.
- Type: **exit**
- Finally we use the tool `dd` to write the image to the USB Stick. In our example we assume that the device is `/dev/sdb`.

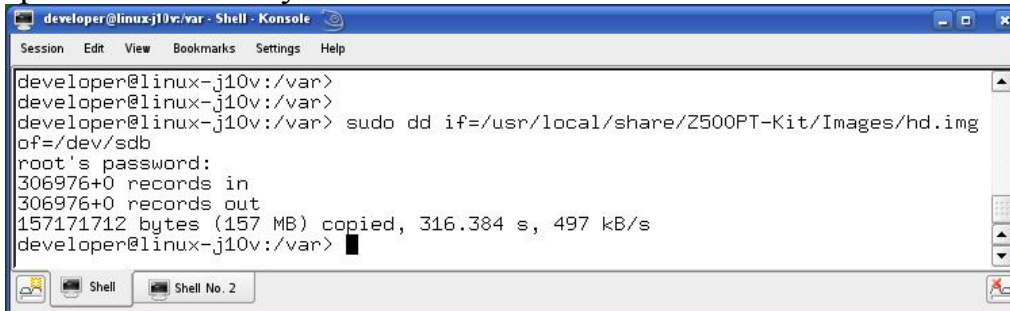


Make sure that the right device is chosen to write to, because otherwise the host system will get damaged!

Type:

sudo dd if=/usr/local/share/Z500PT-Kit/Images/hd.img of=/dev/sdb

The write process can take quite some time, depending on the usb access speed of the host system.



```
developer@linux-j10v:~$ sudo dd if=/usr/local/share/Z500PT-Kit/Images/hd.img of=/dev/sdb
root's password:
306976+0 records in
306976+0 records out
157171712 bytes (157 MB) copied, 316.384 s, 497 kB/s
developer@linux-j10v:~$
```



You have successfully created a bootable USB flash disk!

2.6 Connecting the Host with the Target

In this section you will learn how to connect your host PC with the target. The connection will be done using a cross-over Ethernet cable and a serial one-to-one cable. You will start Linux via the copied image on the USB flash disk (chapter 2.5), and you will be able to login with the serial communication program Microcom as well as via a Telnet session using a peer-to-peer network connection.

- Connect the USB flash disk with the USB port on the target and the Keyboard.
- Optional you can connect the monitor/display on the DVI-port.
- Connect the cross-over Ethernet cable with the connector X27 on the target and the appropriate network card of your host.
- Also connect the COM-Cable with the UART1 port and your PC



Ensure to use the one-to-one serial cable included in this Rapid Development Kit.

- Click the *Microcom* icon on your desktop.



If *Microcom* ends with an *Exitcode* be sure that the serial interface is not used by another application. If the problem still exists then check the folder `/var/lock/` for a file named `LCK...ttyS0` and remove it.

- Connect the AC adapter with the power supply connector X37 (5V) on your board.



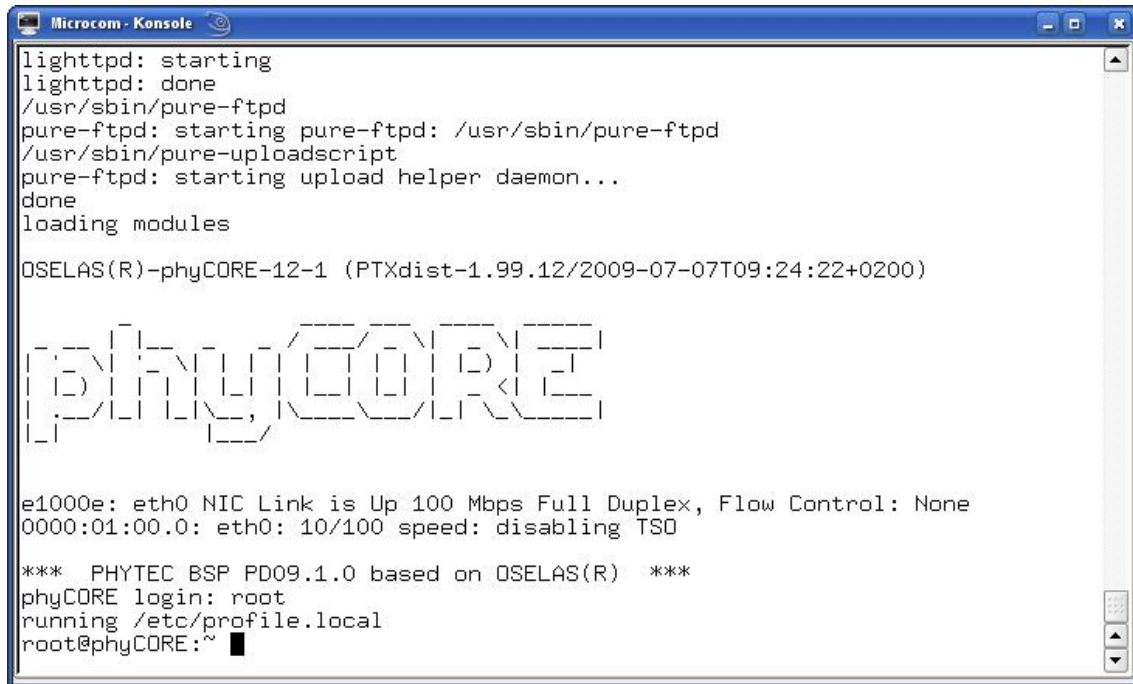
The power connector should have 5 VDC inside, and outside should be ground.

- After connecting the board with the power supply, the target starts booting and after a few seconds *Microcom* shows the output from the COM-Port.



During boot up, the BIOS boot up output can only be seen on a, via DVI connected, monitor. Boot loader and Kernel are by default set to output via COM-Port.

- Click enter and type **root** to login.



```

Microcom - Konsole
lighttpd: starting
lighttpd: done
/usr/sbin/pure-ftpd
pure-ftpd: starting pure-ftpd: /usr/sbin/pure-ftpd
/usr/sbin/pure-uploadsript
pure-ftpd: starting upload helper daemon...
done
loading modules
OSELAS(R)-phyCORE-12-1 (PTXdist-1.99.12/2009-07-07T09:24:22+0200)

PHYCORE

e1000e: eth0 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
0000:01:00.0: eth0: 10/100 speed: disabling TSO

*** PHYTEC BSP PD09.1.0 based on OSELAS(R) ***
phyCORE login: root
running /etc/profile.local
root@phyCORE:~ █

```

- After you have successfully logged in, you can close *Microcom*.



Troubleshooting:

If you don't see any output in the Microcom window there are two possibilities to troubleshoot. First check the serial connection between the target and your host.

At the end of the setup, you had to restart the KDE desktop. If you haven't done yet, restart the KDE desktop now and try again.

- The first potential problem could be missing read and write access to the serial interface:
- Open the *YaST Control Center*.
- Choose *Security and Users*.
- Choose *User Management*.

In the line of you user name should be the group *uucp*.

- If the group is missing, select your user name and click the *Edit* button.
- Select the tab *Details*.
- In *Groups*, check the group *uucp*.
- Click *Accept*.
- Click *Finish* and close YaST.
- You need to log out and log in again for the new group membership to take effect.

Another potential problem could be the BIOS boot priority.

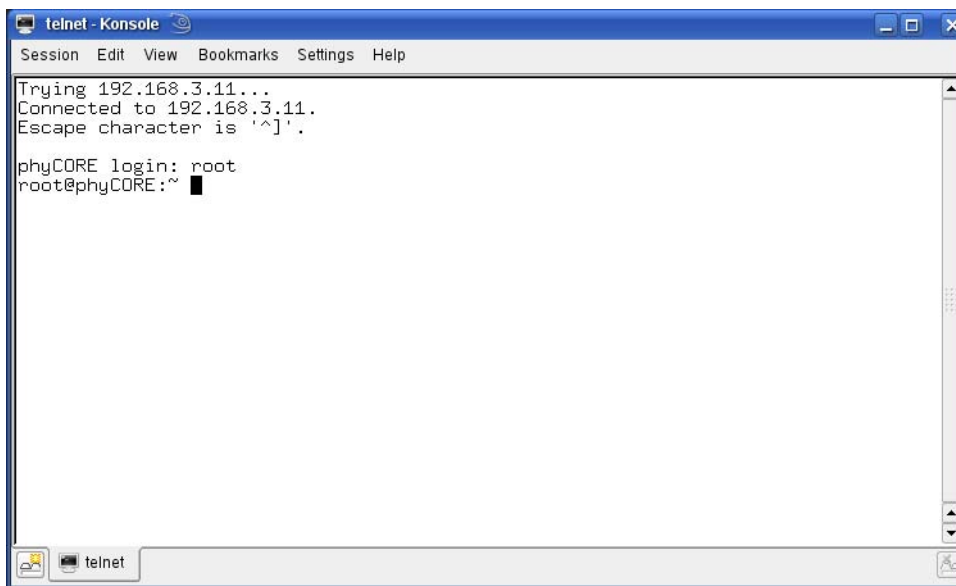
- You need a monitor that you connect with the DVI-port, to see the BIOS output.
- While the device is booting press **** to enter BIOS.
- BIOS will open and after that select *System Setup Utility*.
- Now step to the register *Boot* and change the *Boot Device Prioritization (BBS)*. So that the USB hard drive will checked first for an image. (a device can only be selected once)
- Save changes and exit BIOS. To do that step to register *Exit* and choose *Save Settings and Restart*.

Now you can test the network connection to the target.



- Click the *Telnet for Target* icon on your desktop.

A new window with a connection to the target opens.



If you can see the user login in the opened window, the network configurations were configured correctly.

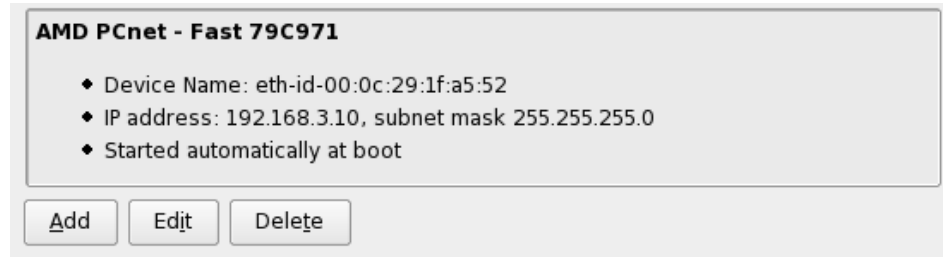
- Close the window.



Troubleshooting:

If you don't see the user login, check the connection between the target and the host. If you have installed more than one network card on your host, be sure to connect the cable with the network card you have configured with the IP address 192.168.3.10.

If you do not see the login, you may not have set up the right IP address of your host. You can check the settings of your network card by opening YaST. In the YaST Control Center you can select *Network Settings* in *Network Devices*. There should be the following configuration:



Information on how to configure your network device can be found in the section *Configuring the Host Platform*.

You can also check the IP address of your target.

- Open a Terminal and connect the COM-Port of your host with the target. Open Microcom and boot the Z500PT.
- After you have logged in type:

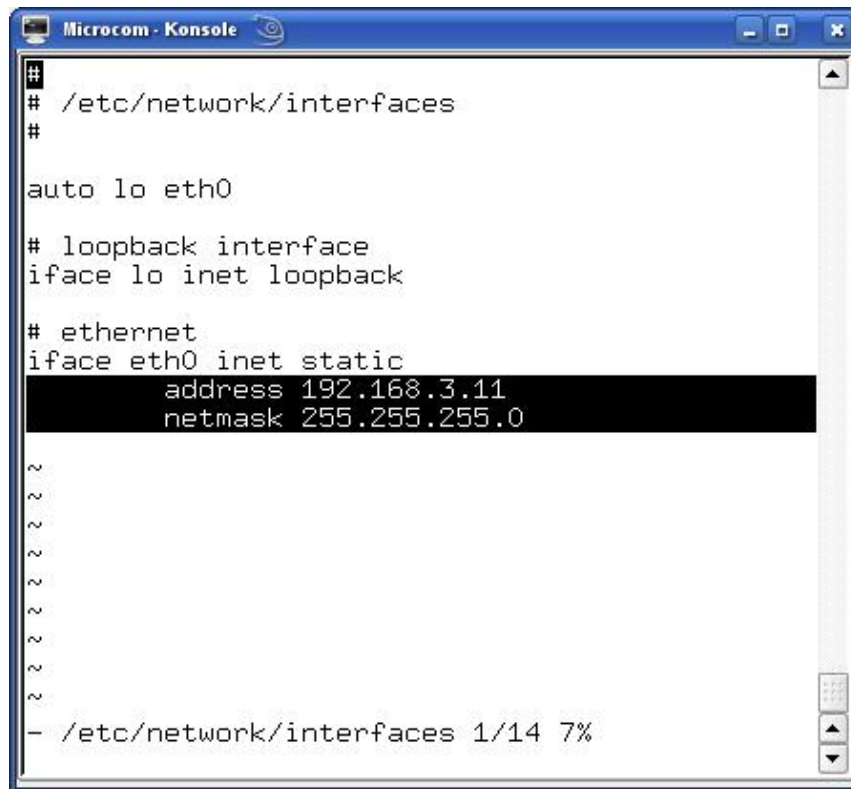
ifconfig

and compare the shown eth0 address with the following
IP 192.168.3.11 Subnet 255.255.255.0

If you have differing addresses you should change it with following command:

- **ifconfig eth0 192.168.3.11 netmask 255.255.255.0**

Try the new changes and when it works edit the */etc/network/interfaces* – file, in order to assure that during booting the right address is automatically set.



```
Microcom - Konsole
# /etc/network/interfaces
#
auto lo eth0
# loopback interface
iface lo inet loopback
# ethernet
iface eth0 inet static
    address 192.168.3.11
    netmask 255.255.255.0
~
~
~
~
~
~
~
- /etc/network/interfaces 1/14 7%
```



You have successfully set up all configurations to access your phyCORE-Z500PT from your host.

SUCCESS

2.7 Copying an Example to the Target

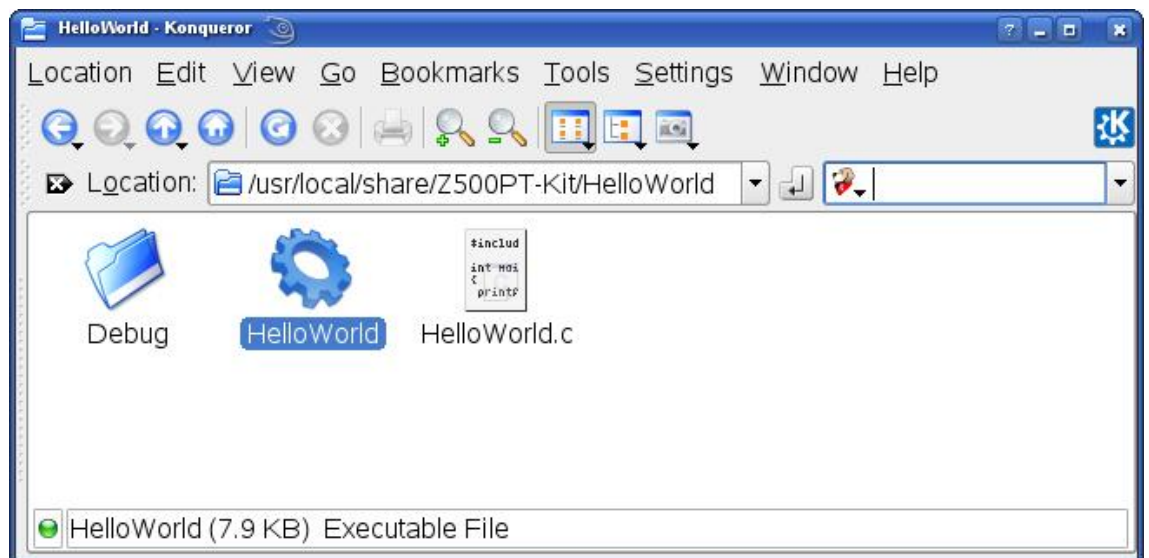
In this section you will learn how to copy an example program to the target using the FTP protocol with the *Konqueror* browser. After that you will execute the example on the target. At the end of this passage you can find some information on how to copy and execute a file on the target using the command line.

Copying a Program to the Target

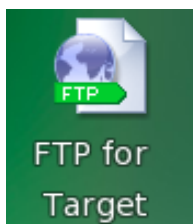


- First click the *Z500PT-Kit* icon on your KDE desktop.

A new window with the contents of the installation directory opens.

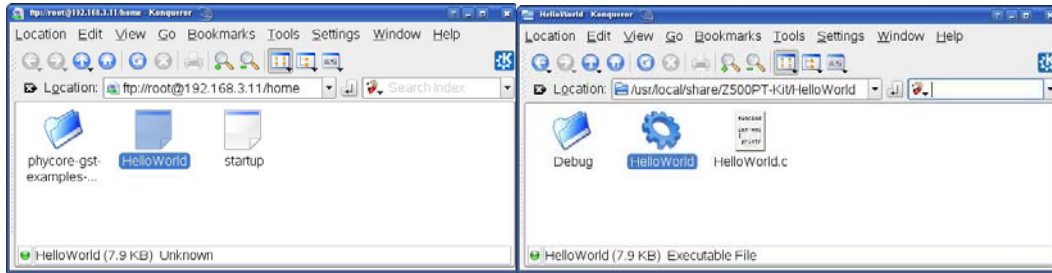


- Enter the directory *HelloWorld*.



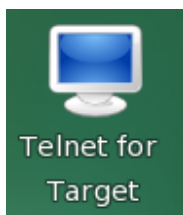
- Click the *FTP for Target* icon on your desktop.

A window with an FTP session to the target opens. Now you have two windows opened, one for the target and one for the host. You can use these two windows to copy files per “drag and drop” from the host to the target (and vice versa).

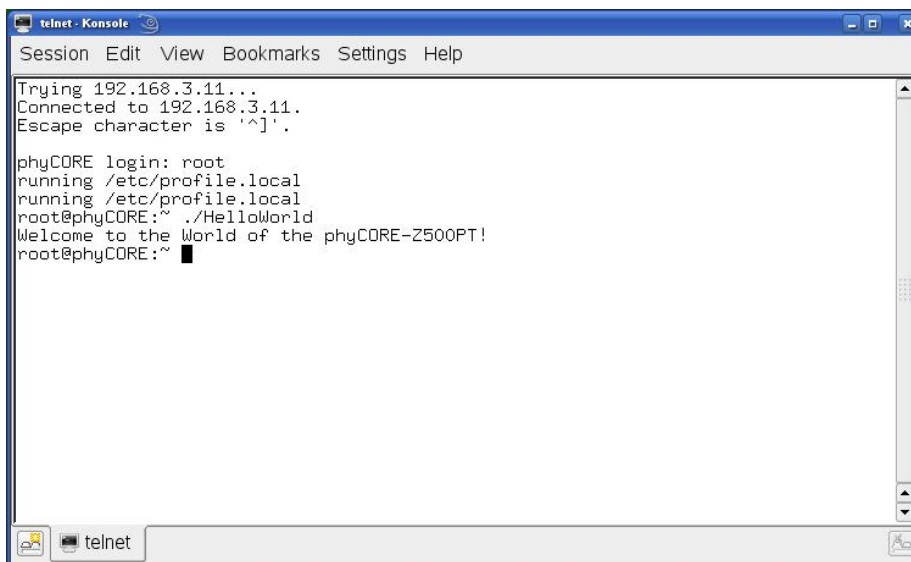


- Select the window that lists *HelloWorld* program on your hard disk.
- Click the *HelloWorld* program and hold the left mouse button pressed.
- Drag the program into the window with the FTP session to the target and release the mouse button.
- Choose *Copy here* in the appearing context menu.
- Close the two windows.

Using Telnet to Execute a Program on the Target



- Click the *Telnet for Target* icon on your KDE desktop.



- Enter **root** as login and press **Enter**.
- Enter **./HelloWorld** and press **Enter**.

The program starts and you should see the following output:

Welcome to the World of the phyCORE-Z500PT!

Using SSH to Execute a Program on the Target

SSH can be used if you want to execute a program directly from the host on the target. Later, this will be used to execute programs out of Eclipse on the target. Before you can start programs out of Eclipse, you have to log in to the target via SSH from the command line for one time. This is necessary to add the RSA public key of the target to the list of known hosts.



TIP

There are several authentication methods when using SSH. The method used on the phyCORE-Z500PT is the *hosts.equiv* method combined with RSA-based host authentication.

If the machine the user logs in from is listed in */etc/hosts.equiv* on the remote machine, and the user name is the same on both sides, the user is allowed to log in.

On the target, the file */etc/hosts.equiv* has the following entry:

```
# file: /etc/hosts.equiv
#
# Allow access from everywhere.
#
+ +
```

The “+ +” means that every user can log in from every host.

When the host connects to the target, the file *~/.ssh/known_hosts* (on the host) is consulted when using *hosts.equiv* with RSA host authentication to check the public key of the target. The key must be listed in this file to be accepted. When the host connects to the target for the first

time, you will be asked to store the target's RSA public key to your `~/.ssh/known_hosts`. If you agree to do this, then the host will be able to connect to the target without entering a password.



This authentication method closes security holes due to IP spoofing, DNS spoofing, and routing spoofing. But note that `/etc/hosts.equiv` is, in general, inherently insecure and should be disabled if security is a concern.



- Click the *SSH for Target* icon on the desktop.

A new window opens.

```
ssh - Konsole
Session Edit View Bookmarks Settings Help
The authenticity of host '192.168.3.11 (192.168.3.11)' can't be established.
RSA key fingerprint is 23:05:70:cb:4f:2a:5a:de:03:13:7b:eb:8f:41:69:97.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.3.11' (RSA) to the list of known hosts.
root@phyCORE:~
```

In this window you can see that the authenticity of the phyCORE-Z500PT can't be established. This is normal if you want to create an SSH connection for the first time.

- Enter **yes** and press **Enter** to continue. The RSA public key of the target will be permanently added to the list of the known hosts.



Troubleshooting:

If an error occurs and you can't see the `root@phyCORE:~>` prompt, open a terminal window and enter the following command:

- **rm ~/.ssh/known_hosts**

Try to log in again by entering:

- **ssh root@192.168.3.11**
- Enter **yes** to add the target to the list of known hosts.

Now you should see the target's prompt.



We expect that you haven't changed the SSH configuration file on your host. If you change this file, the authentication may not work.

Now you are logged in, you can execute programs on the target.

- Type **./HelloWorld** to start the program you had copied to the phyCORE-Z500PT before.

The program starts and you should see the following output:

Welcome to the World of the phyCORE-Z500PT!

- Close the SSH window.



You have successfully copied and executed an example application on the target.

2.8 Advanced Information

Copying a Program to the Target with the Command Line

- Open a new terminal window.
- Change to */usr/local/share/Z500PT-Kit/HelloWorld*:

```
cd /usr/local/share/Z500PT-Kit/HelloWorld
```

- Copy the application to the target by typing:

```
ftp -u ftp://root:root@192.168.3.11/ HelloWorld
```

Be sure to enter a slash followed by a space after the IP address.

Executing a Program on the Target

- Open a Telnet session to the target:

```
telnet 192.168.3.11
```

- Type **root** and press **Enter**.
- Type **./HelloWorld** to start the application.

Executing a Program directly on the Target using SSH

- To start the program, type:

```
ssh root@192.168.3.11 ./HelloWorld
```

After the program has finished, SSH will logout automatically.

Chapter 3 Getting More Involved



70 min

In this chapter you will pass some continuative topics. First you will configure and compile your own kernel and root filesystem. With the configuration tools you can add additional features, or disable them if they are not needed. After the kernel and root filesystem configuration you will learn how to create your own image with the newly created changes.

Then you will start working with the Eclipse platform using the C/C++ Development Tools (CDT) in conjunction with the GCC C/C++ tool chain. You will learn how to configure the Eclipse platform and how to open an existing project. After that you will create your first own project and modify the example's source code.

At the end of this chapter you will execute the program as an external application out of Eclipse. Additionally, you will add your application to the startup configuration of the target so it is automatically started when the phyCORE-Z500PT boots.

3.1 Configuring and Compiling the Image

In this part you will learn how to configure and build a new Linux kernel and a root filesystem. First you will configure the kernel and the root filesystem with the help of PTXdist, a tool to build the Board Support Package and to create your own Image. After the configuration you will create your own image that you can write to the bootable USB flash disk.

The kernel used by PHYTEC is based a standard kernel available from www.kernel.org. Additionally, the kernel archive in your setup installation directory already includes all necessary patches for the phyCORE-Z500PT.

As the first step, open a new terminal.



- Click the terminal icon on your desktop.
- During the Linux-Z500PT-Kit-Setup a folder was created within your home directory in which you can find the *OSELAS.BSP*. Change to the directory:

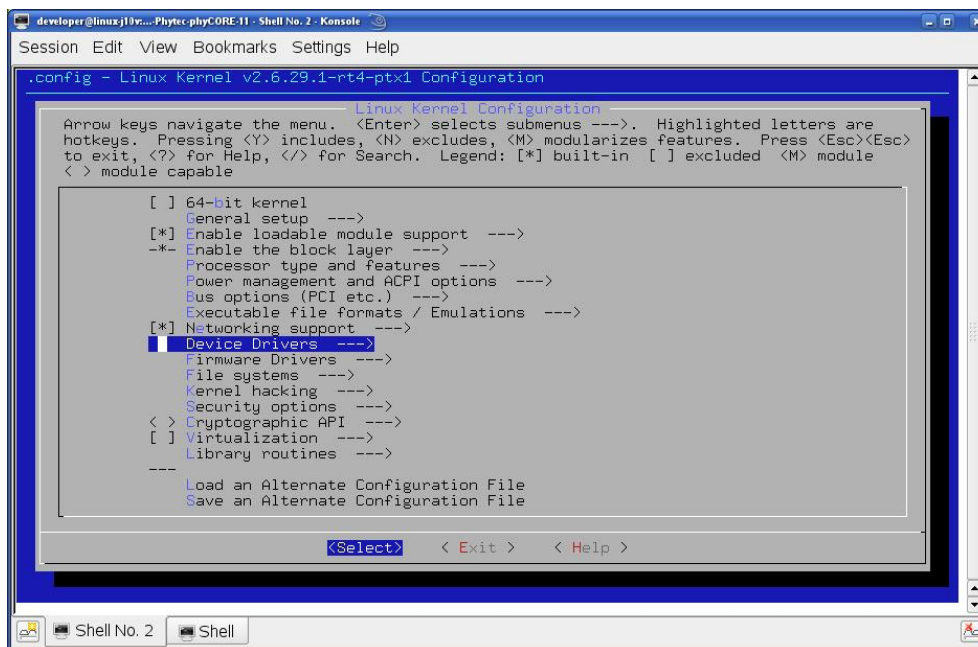
```
cd ~/OSELAS.BSP*
```

First we configure the kernel. In our example we show how to add *MMC/SD/SDIO card support*.

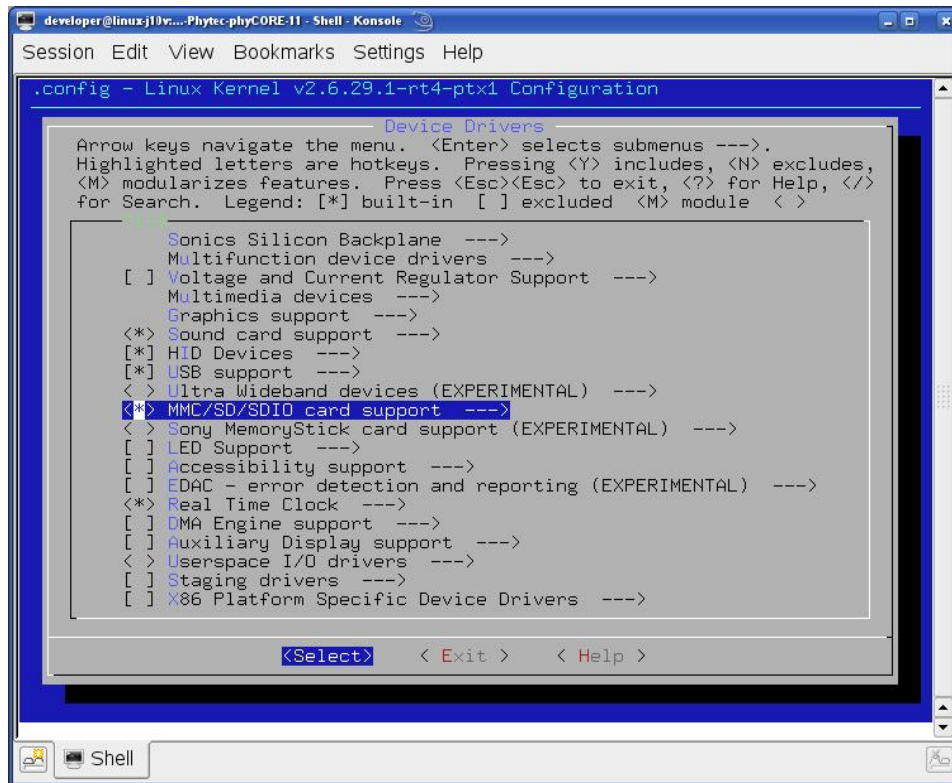
- Type:

```
ptxdist kernelconfig
```

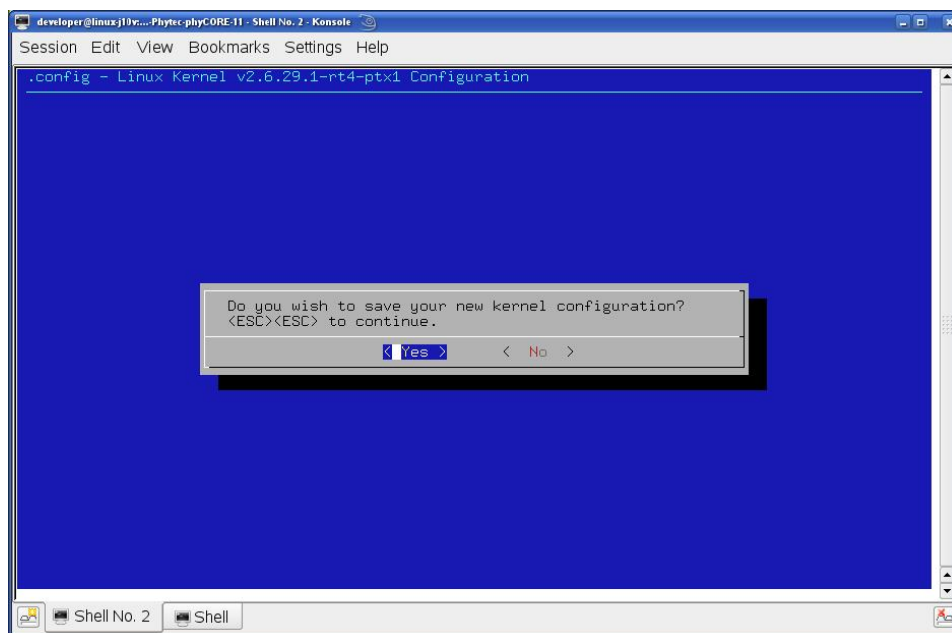
At first start PTXdist is checking dependencies. This may take some seconds.



- Select *Device Drivers*



- Highlight *MMC/SD/SDIO card support* and press **Y**



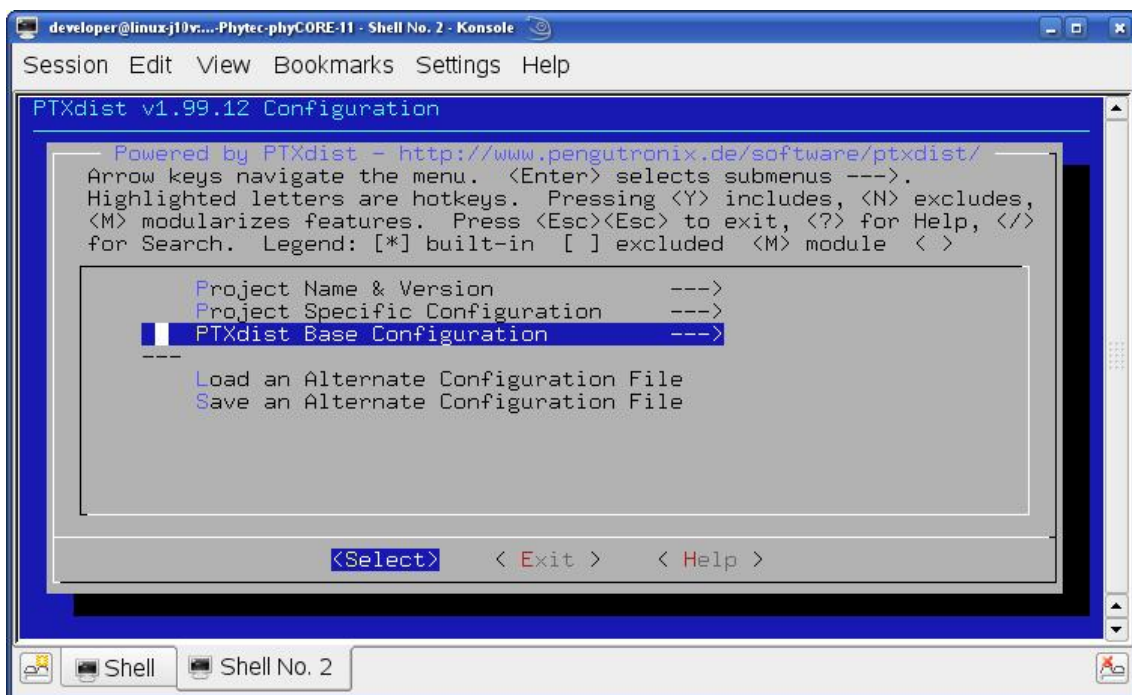
- Now save and exit the PTXdist kernelconfig tool.

The kernel configuration is finished. We can now configure our root filesystem. In our example we show how to add the Midnight Commander.

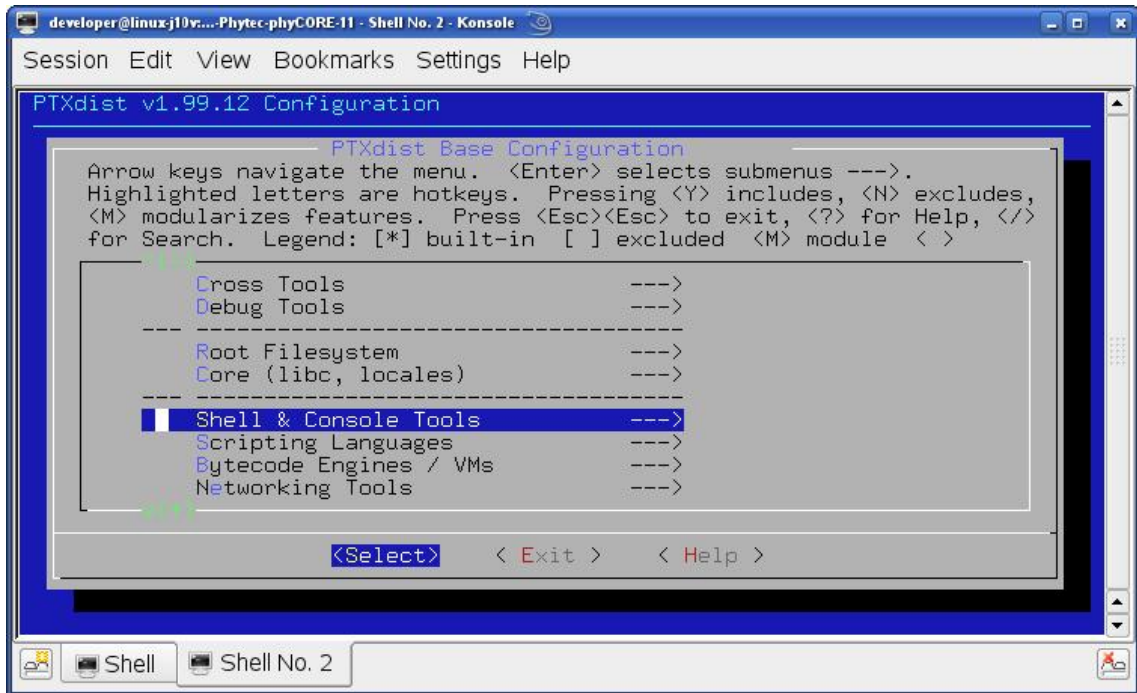
- Type:

ptxdist menuconfig

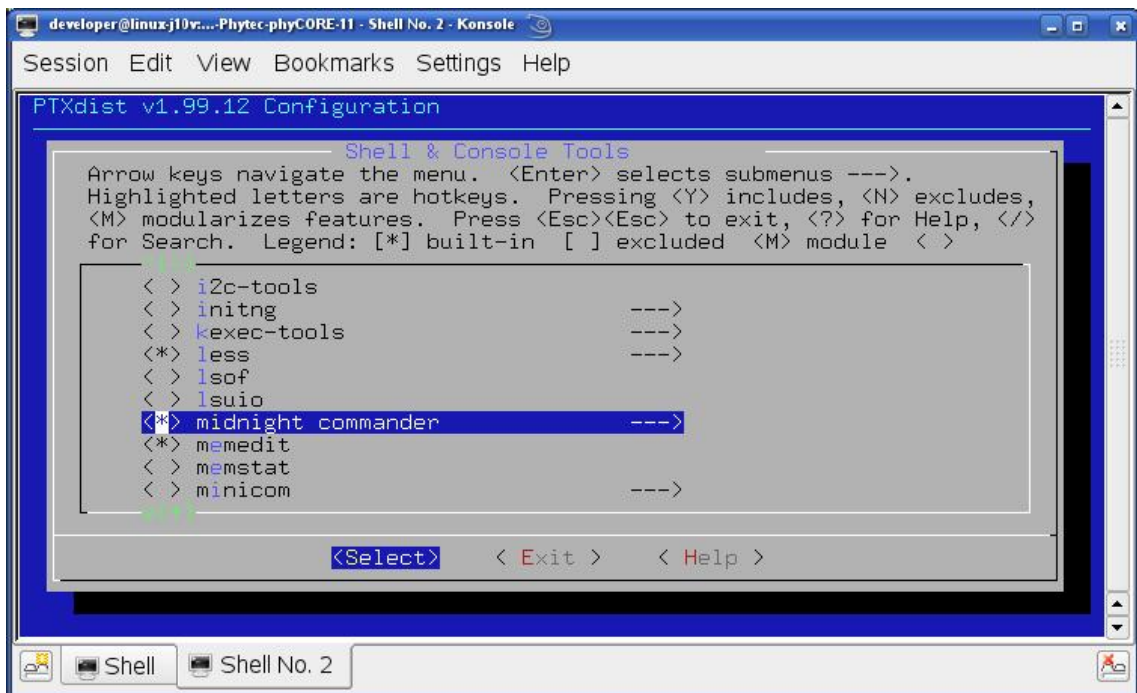
The configuration of PTXdist starts.



- Select *PTXdist Base Configuration*



- Select *Shell & Console Tools*



- Highlight *midnight commander* and press Y
- Now save and exit the PTXdist Configuration tool.

The kernel and the root filesystem configuration are now finished. Before we can create the image we have to build a new root filesystem.

- Type:

ptxdist go

PTXdist automatically determines which packages belong to the project and starts compiling their *targetinstall* stages (which put the compiled binaries into the root filesystem). While doing this, PTXdist determines all dependencies between the packets and brings them into the correct order. Even on reasonably fast machines the time to build the root filesystem is something around 30 minutes up to a few hours.

After we have built a root filesystem, we create an image, which can be flashed to the target device.

- Type:

ptxdist images

The sources will be compiled and the new image will be built. This will take a few minutes. The new image will be written to */OSELAS.BSP*/platform-phyCORE-Z500PT/images/*

- Close the terminal window.



In this section you learned how to configure and compile a new image. Now you know how to add or remove features to and from your kernel.

Writing the Image into the Target's solid state disk

In this passage you will find a description on how to write the newly created image into the phyCORE-Z500PT's solid state disk.

First we make a bootable USB flash disk with our new image. In chapter 2.5 you find an instruction how to do this. The difference between chapter

2.5 and now is the last command. Now you must select the newly created *hd.img* in the *if*-term. The complete *dd* command looks like this:

- **`dd if=~ / OSELAS.BSP* / platform-phyCORE-Z500PT / images / hd.img of=/dev/sdb`**

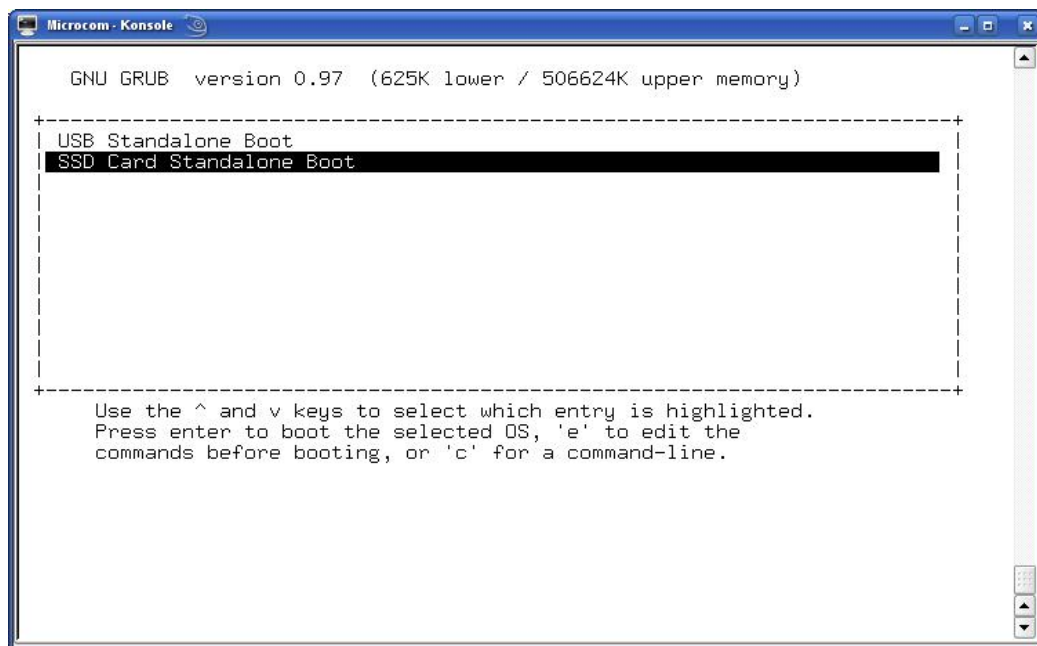


Make sure that the right device (“*of=/dev/sdb*”) is chosen to write to, because otherwise the host system will get damaged!

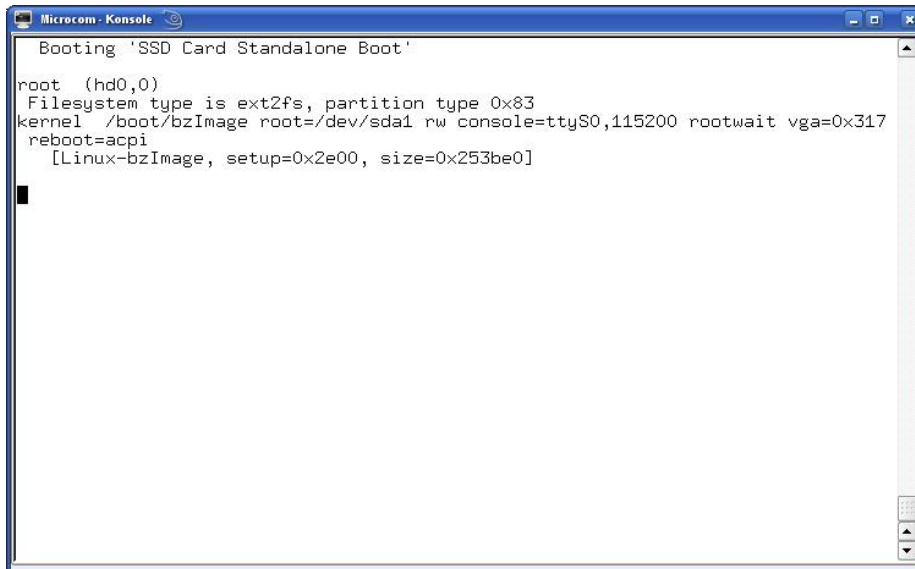
When it is finish we boot the device as explained in chapter 2.6 . After booting and a successful login it is time to write the image from the USB stick into the solid state disk.

Again we need the *dd* tool. Now the *if*-term is the USB stick and the *of*-term is the solid state disk. Write following command in Microcom:

- **`dd if=/dev/sdb of=/dev/sda`**
- Reboot the phyCORE-Z500PT

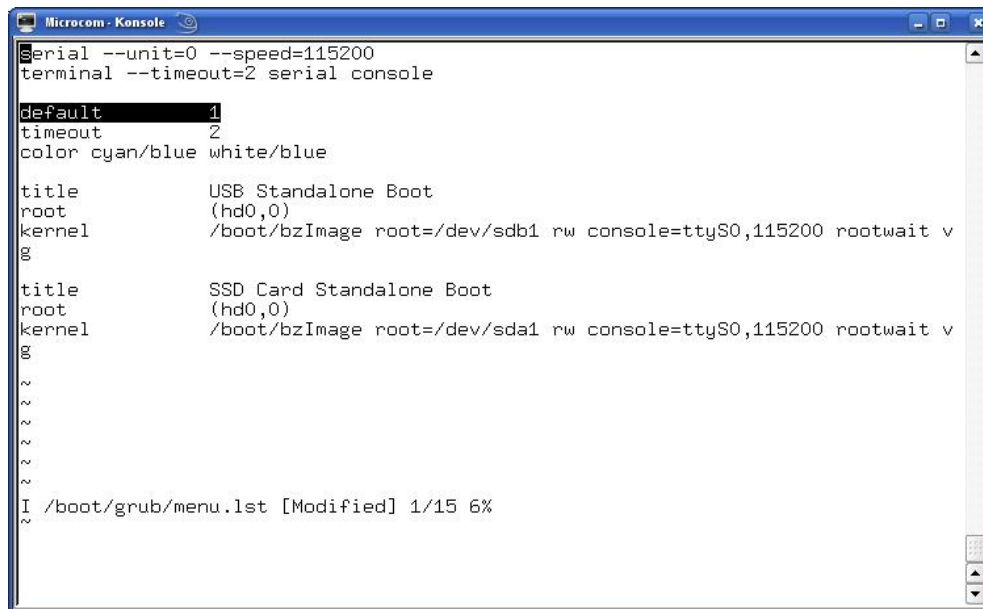


- When the *GRUB bootloader* screen appears press any key and after that select *SSD Card Standalone Boot*.



```
Microcom - Konsole
Booting 'SSD Card Standalone Boot'
root (hd0,0)
Filesystem type is ext2fs, partition type 0x83
kernel /boot/bzImage root=/dev/sda1 rw console=ttyS0,115200 rootwait vga=0x317
reboot=acpi
 [Linux-bzImage, setup=0x2e00, size=0x253be0]
```

- The board is booting the image.
- When the kernel has completely booted, you can login
- Then we must edit */boot/grub/menu.lst*, for example with *vim*, to ensure that previously made changes are permanent and will be used as default settings on the next reboot. In this file we must change the variable *default* to 1 to ensure that the device is booting from the solid state disk (0 for booting from USB).



```
Microcom - Konsole
Serial --unit=0 --speed=115200
terminal --timeout=2 serial console
default 1
timeout 2
color cyan/blue white/blue

title          USB Standalone Boot
root           (hd0,0)
kernel        /boot/bzImage root=/dev/sdb1 rw console=ttyS0,115200 rootwait v
g

title          SSD Card Standalone Boot
root           (hd0,0)
kernel        /boot/bzImage root=/dev/sda1 rw console=ttyS0,115200 rootwait v
g
~
~
~
~
~
I /boot/grub/menu.lst [Modified] 1/15 6%
```



We recommend to change the menu.lst also on the host, because if you regenerate or flash a image to your target it will get lost.

A template of this file can be found in

```
/OSELAS.BSP*/projectroot/boot/grub/menu.lst.phyCORE-Z500PT
```



Troubleshooting:

If any problem occurs after writing the image into solid state disk, you can restore the original image from your setup CD-ROM. You can find the description in chapter 2.5



In this section you learned how to write the image from the USB stick to the solid state disk, and finally the target was started with the new image.

3.2 Opening an Existing Project

In this section you will import an existing Eclipse project into your workspace. The imported example project will be compiled with the cross compiler. After compiling the project, you will copy and execute the newly created program on the target.

Copying the *HelloWorld* Project



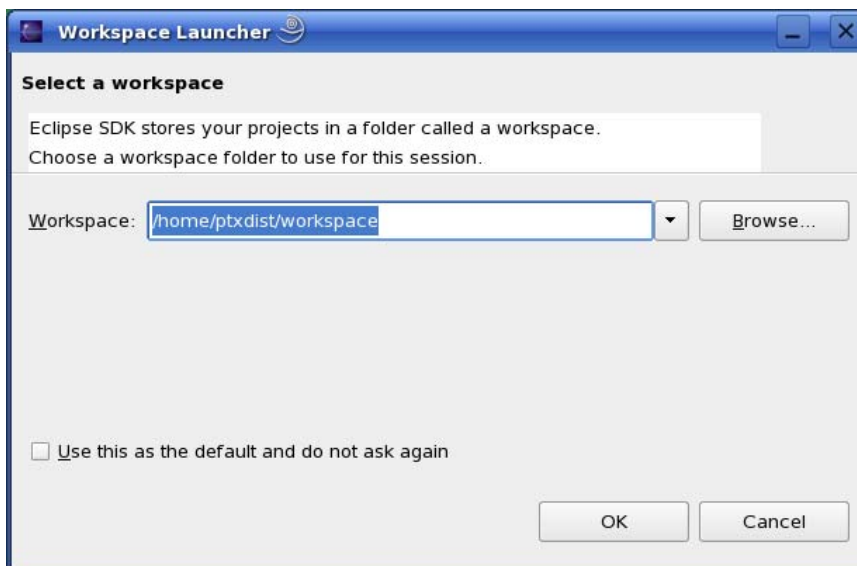
- Click the *Z500PT-Kit* icon on your KDE desktop.
- Right-click the *HelloWorld* directory and select *Copy*.
- Browse to your home directory.

- If the *workspace* directory doesn't exist, create a directory *workspace* in your home directory.
- Enter the *workspace* directory.
- Right-click in the *workspace* directory and select *Paste*.

Starting Eclipse and Importing the Example Project

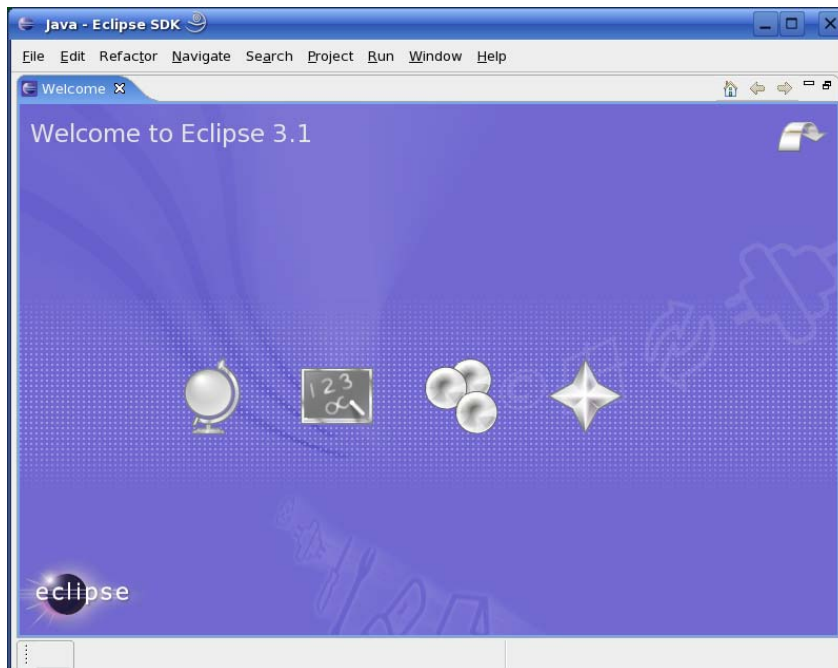


- Click the *Eclipse* icon to start the application. You can find this icon on your desktop.

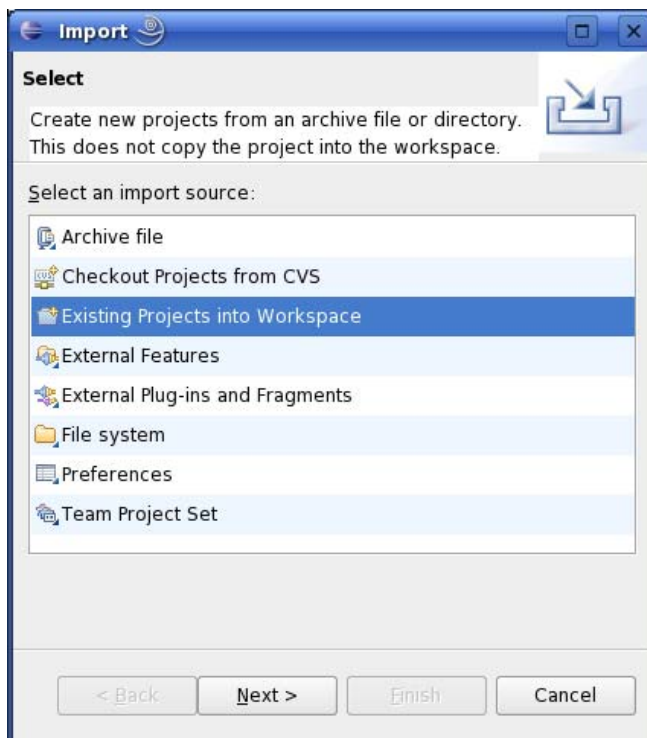


- Confirm the Workspace directory with *OK*.

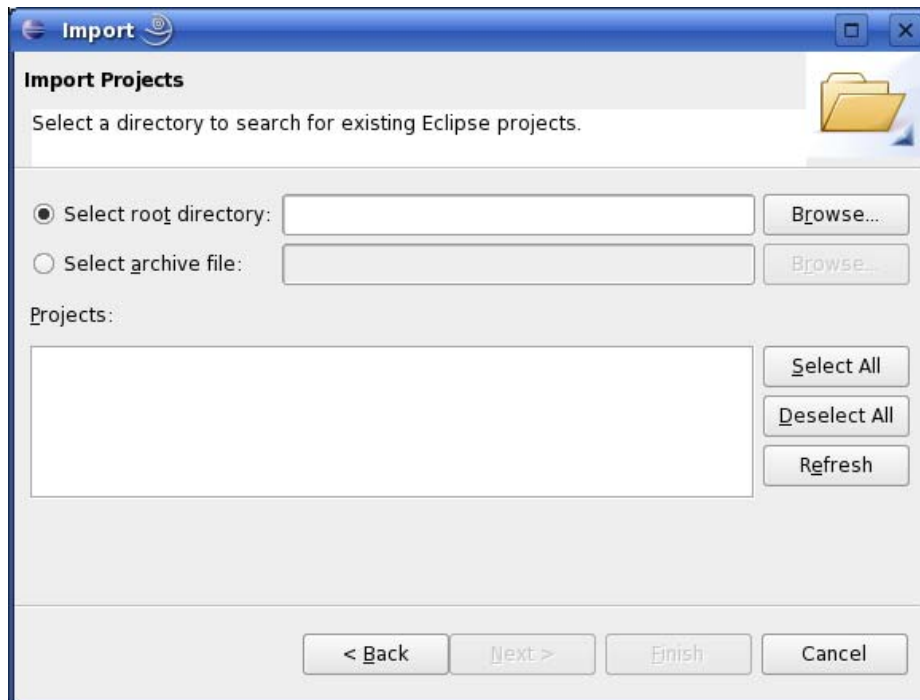
The *Welcome* screen will appear.



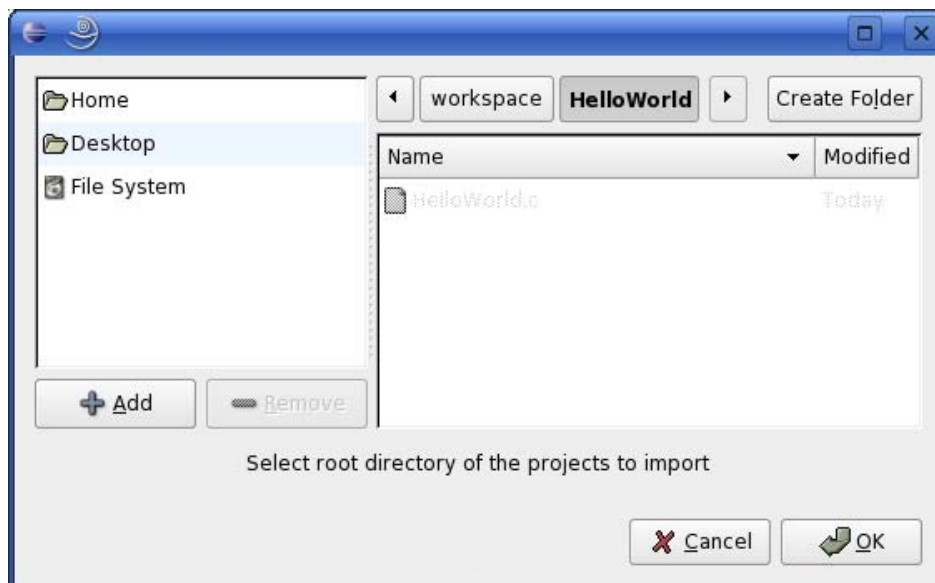
- Select *File* ► *Import* from the menu bar.



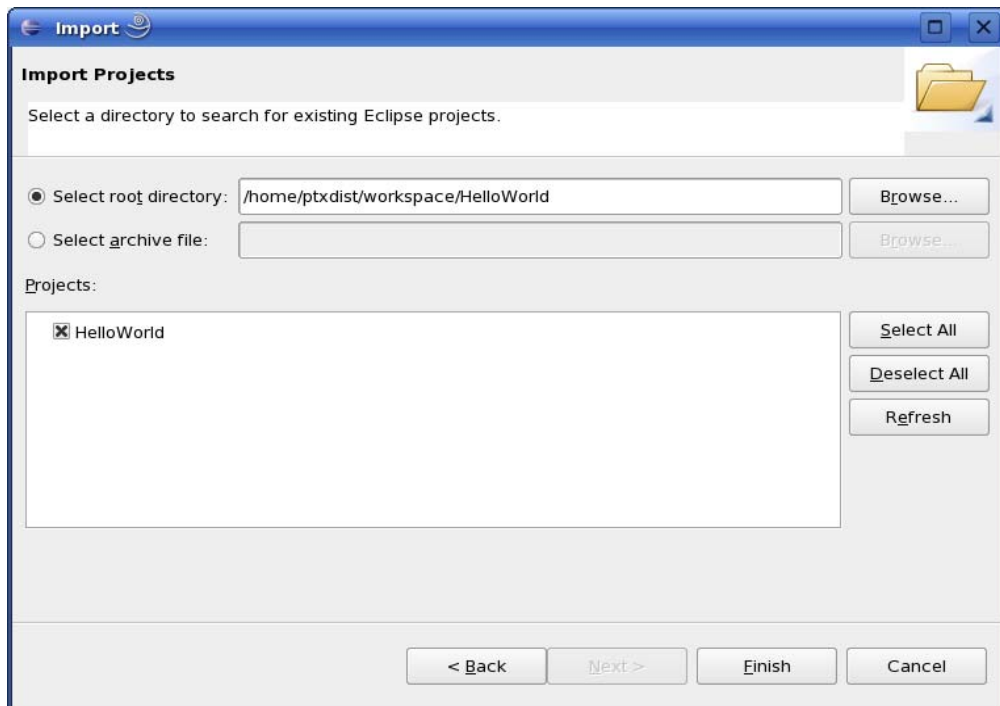
- Select *Existing Projects into Workspace*.
- Click *Next*.



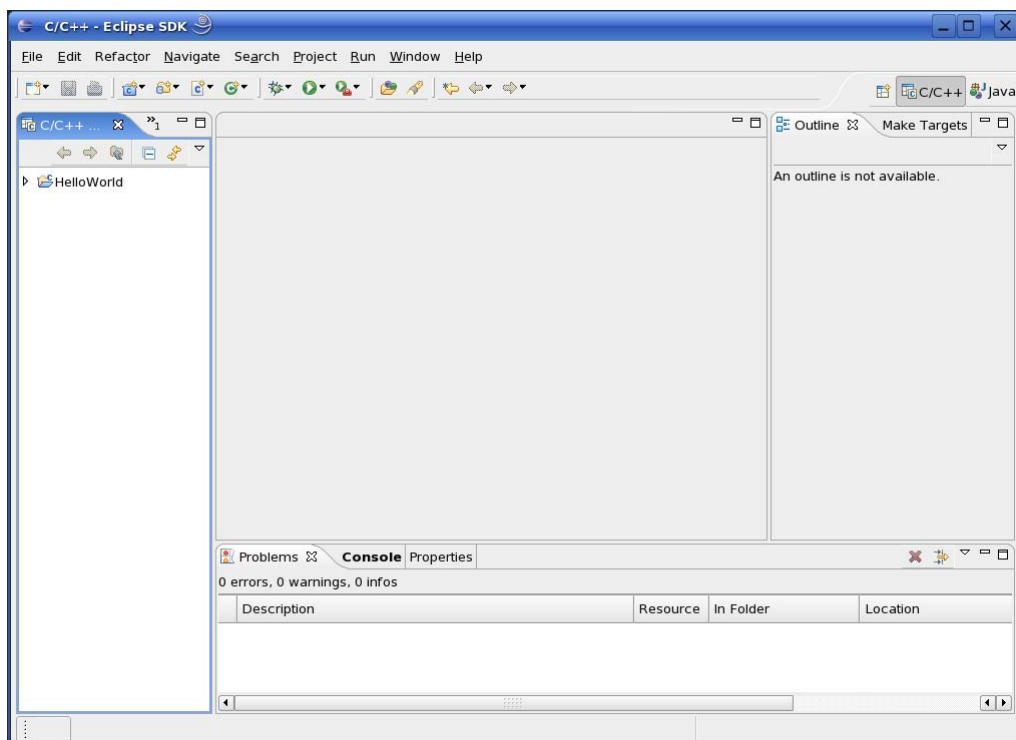
- Select *Browse*.



- Double-click the *HelloWorld* directory in your home directory.
- Click *OK*.



- Select *Finish* to import the project.



- Close the *Welcome* screen.

The *HelloWorld* program will be compiled and the *HelloWorld* executable is built for the target. Then the *HelloWorld* file is copied to the target using FTP. After the file has been copied to the target, the program is executed on the target using SSH. You should now see the “*Welcome to the World of the phyCORE-Z500PT!*” message in the *Console* window.

- Select the *Console* tab.

You will see the following content in the *Console* window:



```

C-Build [my>HelloWorld]
building file: ../helloworld.c
Invoking: GCC C Compiler
gcc -O0 -g3 -Wall -c -fmessage-length=0 -oHelloWorld.o ../HelloWorld.c
Finished building: ../HelloWorld.c

Building target: HelloWorld
Invoking: GCC C Linker
gcc -o HelloWorld ../HelloWorld.o
Finished building target: HelloWorld
|
Build complete for project HelloWorld
  
```



If the project is not built automatically, you will have to check *Project* ► *Build automatically* from the menu bar.



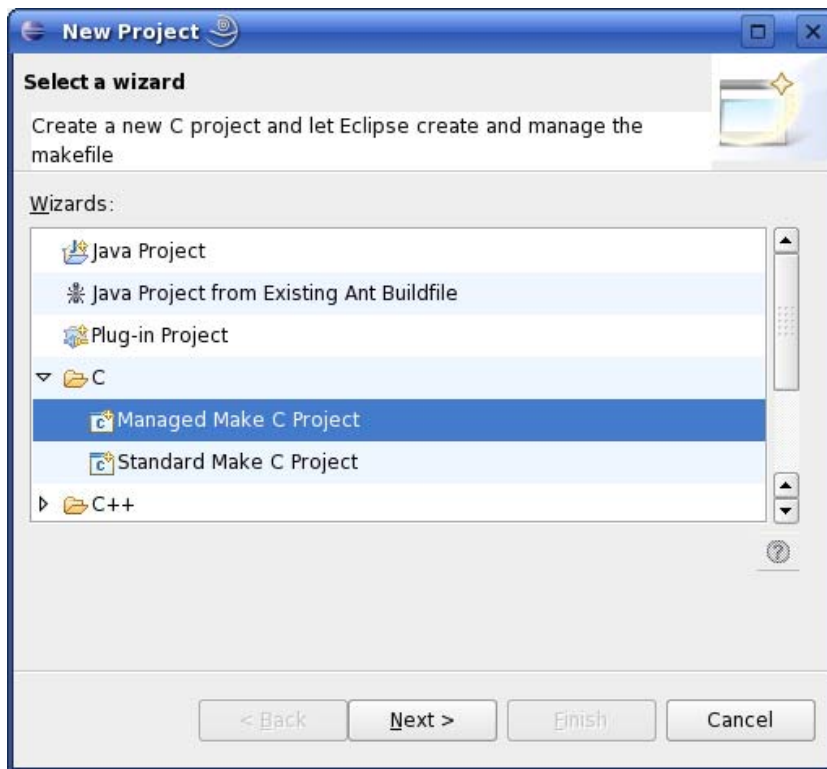
You have successfully passed the first steps with the Eclipse IDE. You are now able to import existing projects into the Eclipse Workspace. You can compile an existing project and execute the program on the target.

3.3 Creating a New Project

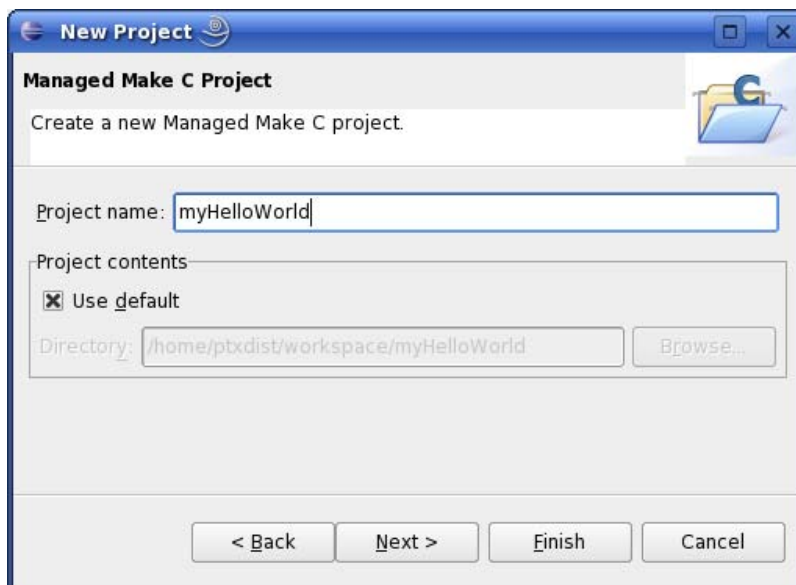
In this section you will learn how to create a new project with Eclipse and how to configure the project for use with the GNU C/C++ cross development tool chain.

- Open Eclipse if it isn't already opened.
- Select *File* ► *New* ► *Project* from the menu bar.

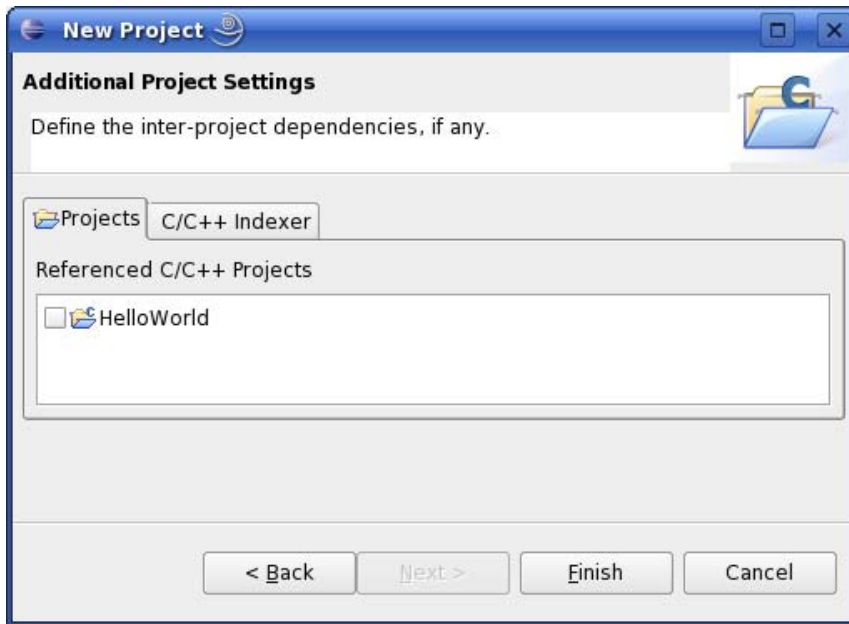
A new dialog opens.



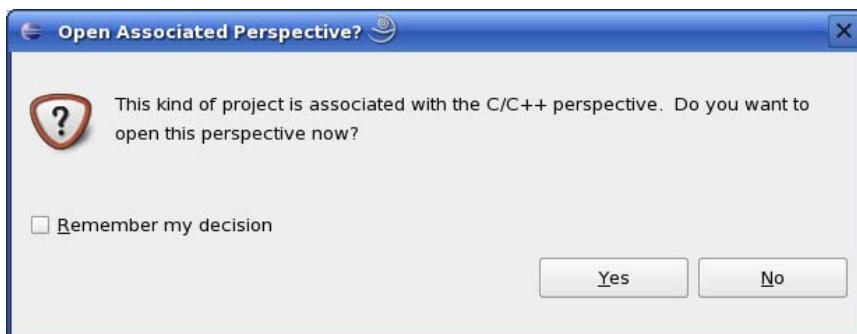
- Select *Managed Make C Project* and click *Next*.



- Enter the project name *myHelloWorld* and click *Next*.
- Click *Next*.

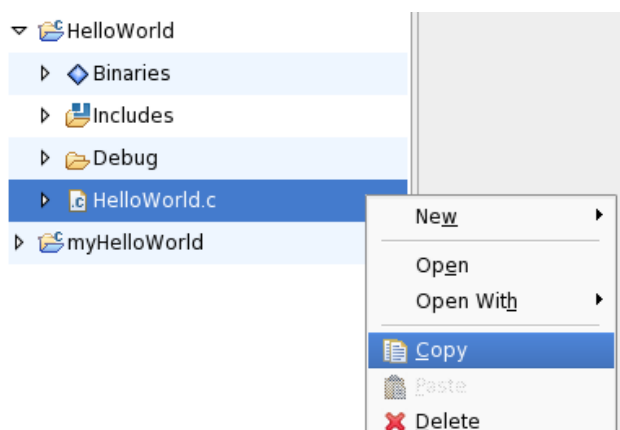
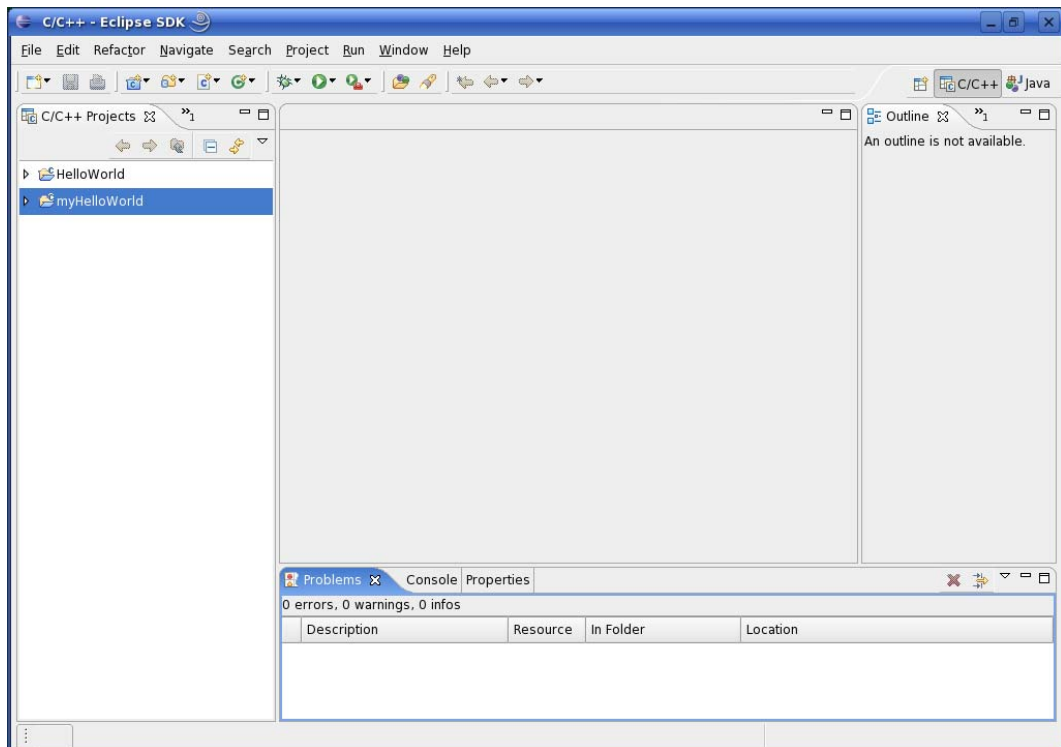


- Click *Finish*.

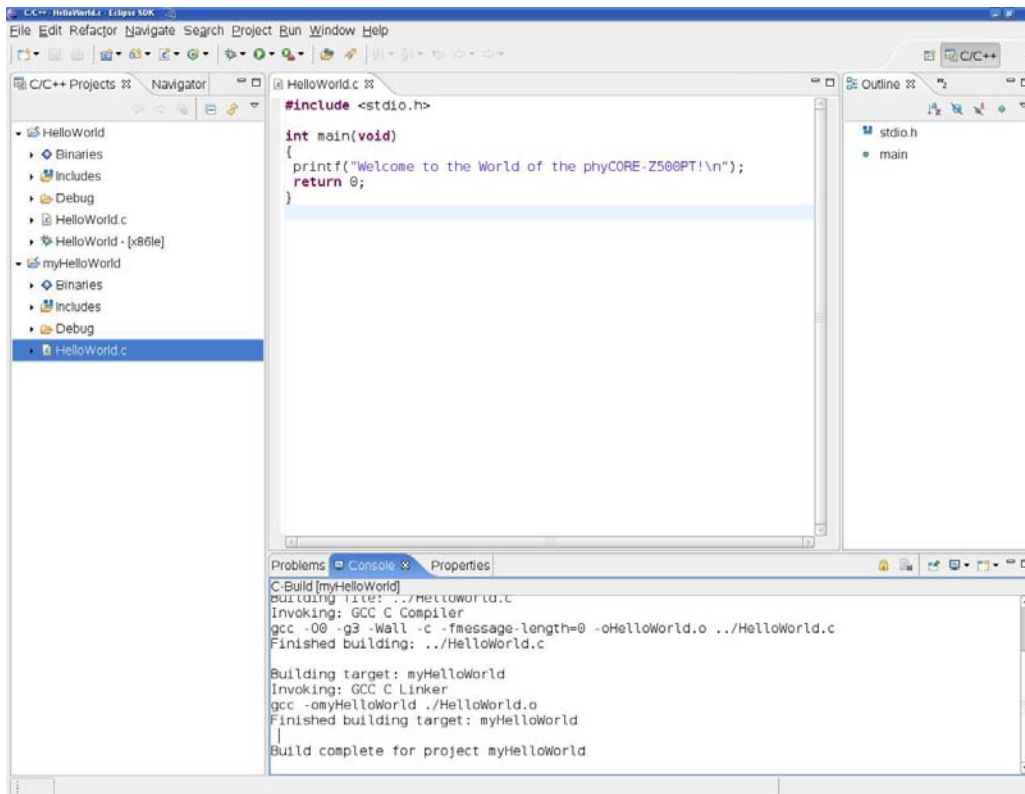


- Select *Yes* to open the C/C++ perspective.

You will see the C/C++ IDE with the *myHelloWorld* project.



- Right-click on *HelloWorld.c* in the *HelloWorld* project which we have worked with previously.
- Select *Copy*.
- Select the *myHelloWorld* project.
- Right-click and select *Paste*.



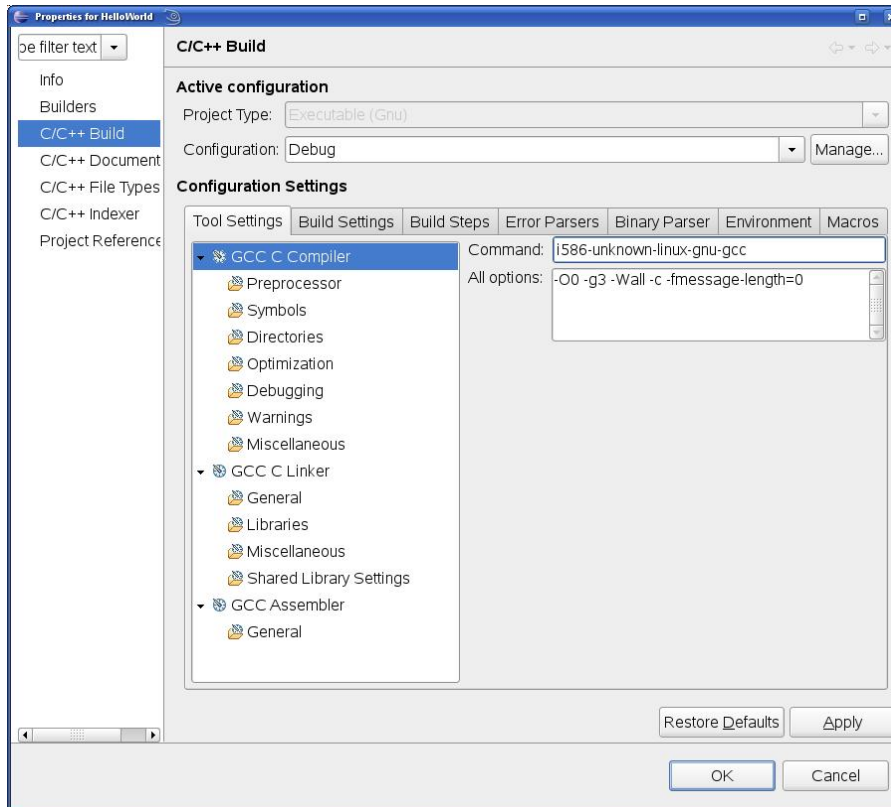
- Double-click on *HelloWorld.c* in the *myHelloWorld* project.

If *Build Automatically* from the *Project* menu is selected, the *HelloWorld* application will now be compiled and created with the standard GCC C/C++ compiler suitable for your host machine. You will find the executable file, which can only be run on your host system, in the *workspace/myHelloWorld/Debug* directory.

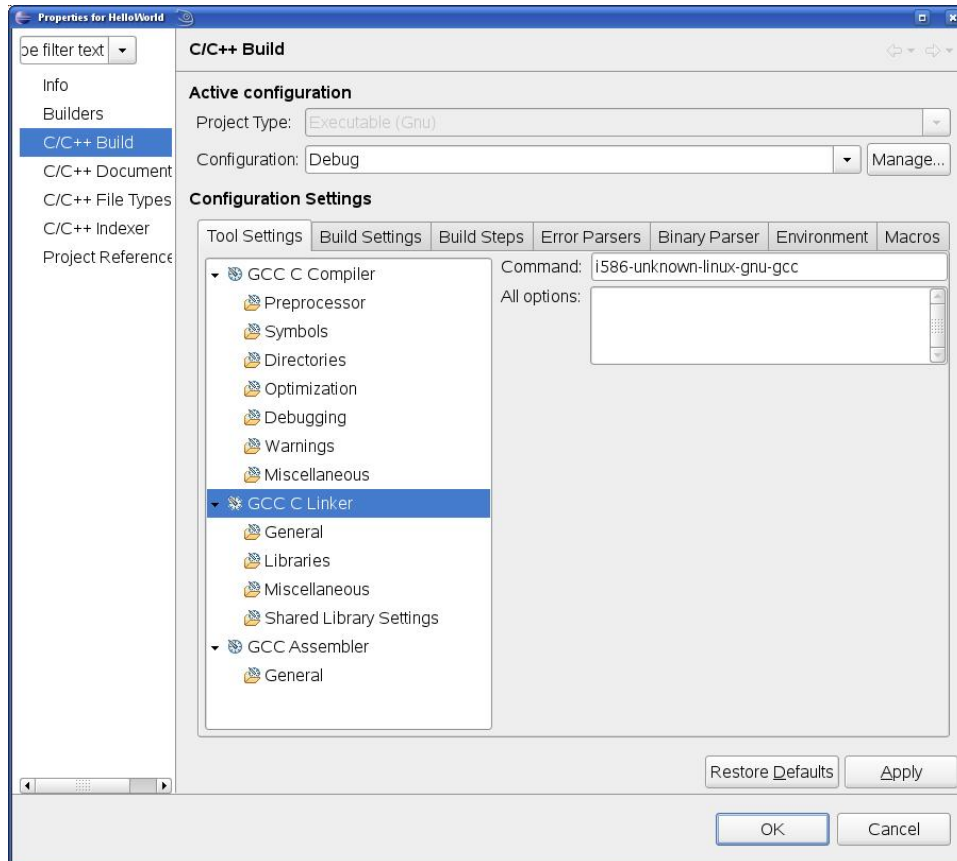
To compile your project for the phyCORE-Z500PT instead, you will have to use the GNU C/C++ cross compiler.

- Right-click the *myHelloWorld* project and choose *Properties*.

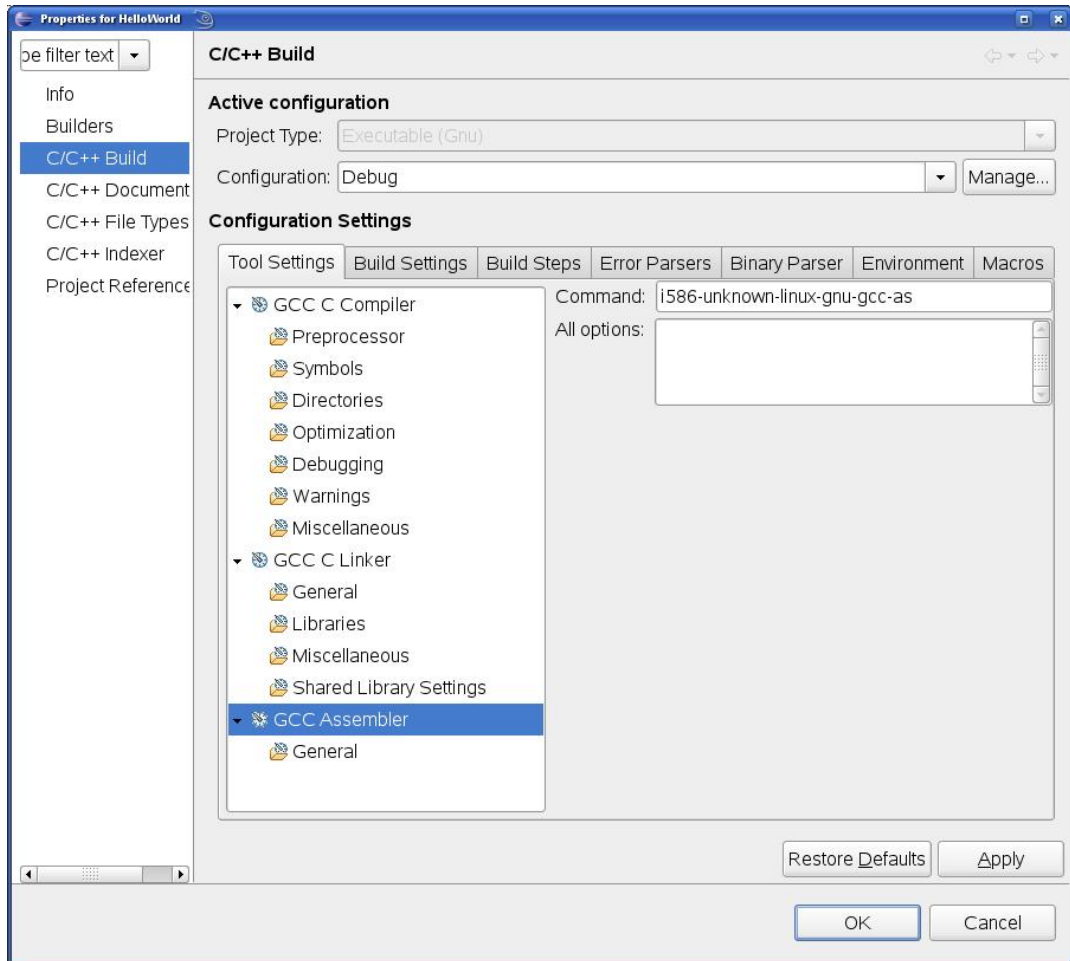
The *Properties* dialog appears.



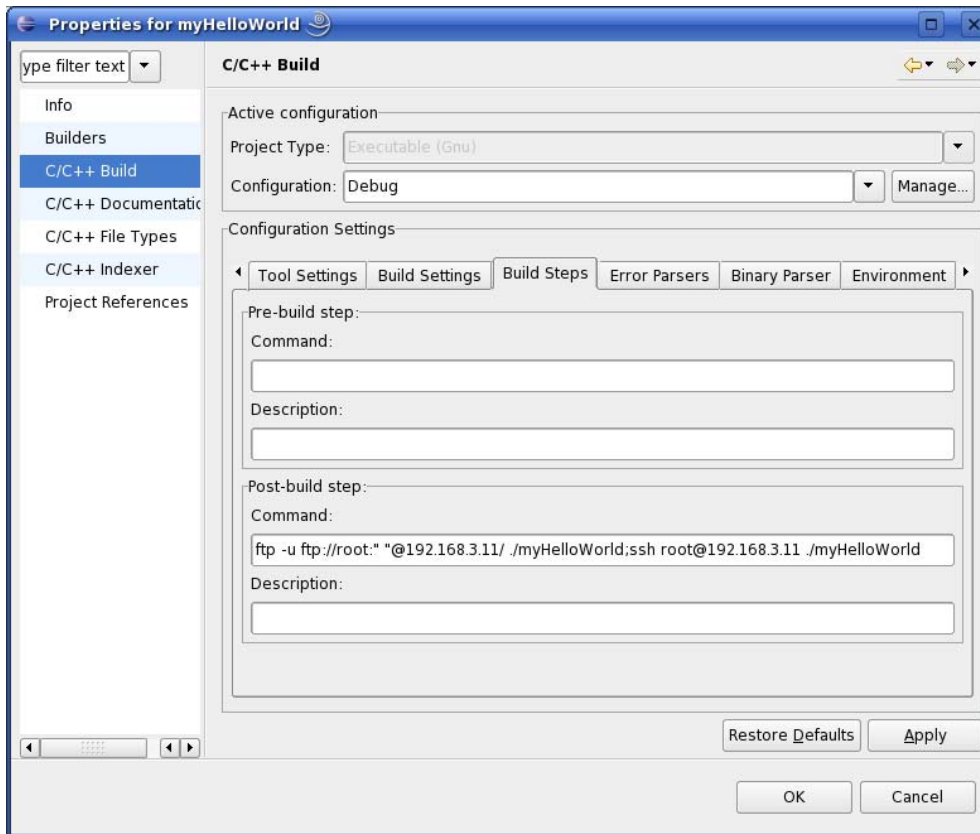
- Select *C/C++ Build*.
- Select *GCC C Compiler*.
- Enter **i586-unknown-linux-gnu-gcc** into the *Command* input field.



- Select *GCC C Linker*.
- Enter **i586-unknown-linux-gnu-gcc** into the *Command* input field.



- Select *GCC Assembler*.
- In the *Command* input field, change the default **as** to **i586-unknown-linux-gnu-as**.
- Click *Apply*.



- Select the *Build Steps* tab.
- Enter following command in the *Command* input field:

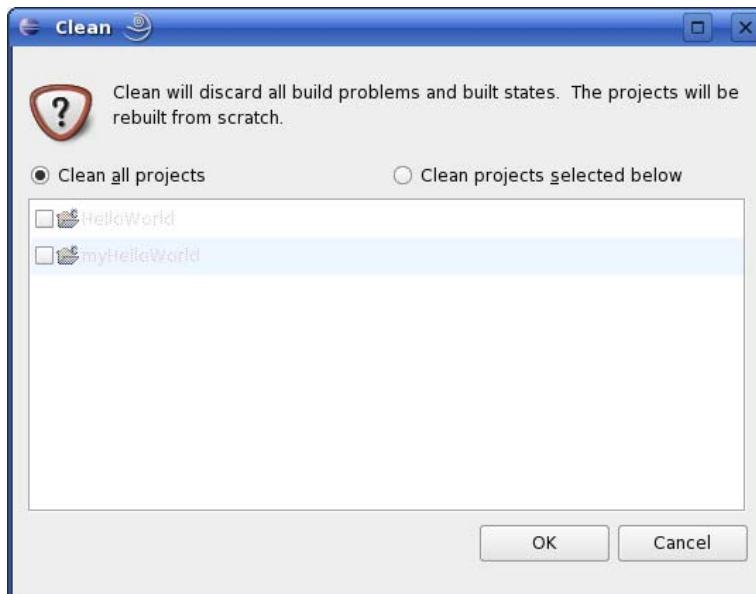
```
ftp -u ftp://root:root@192.168.3.11/ ./myHelloWorld; ssh  
root@192.168.3.11 ./myHelloWorld
```



CAUTION

Be sure to enter the semicolon between **./myHelloWorld** and **ssh**.

- Click *Apply*.
- Click *OK*.



- Select *Project* ► *Clean* from the menu bar.
- Confirm with *OK*.

The project will be rebuilt.

- Select the *Console* tab.

If no errors occur while building the project, you will see the following output:

```

C-Build [myHelloWorld]
HelloWorld.c
Finished building: ../HelloWorld.c

Building target: myHelloWorld
Invoking: GCC C Linker
i586-unknown-linux-gnu-gcc -omyHelloWorld ../HelloWorld.o
Finished building target: myHelloWorld

make --no-print-directory post-build
ftp -u ftp://root:root@192.168.3.11/ ./myHelloWorld;ssh root@192.168.3.11 ./
myHelloWorld
Welcome to the World of the phyCORE-Z500PT!

Build complete for project myHelloWorld

```



You have successfully created your first own project with the Eclipse IDE. You have configured the project to create an application for your target platform.

3.4 Changing the Demo Application

Now we will extend the *myHelloWorld* application. The extended *myHelloWorld* application will write an output to the first serial interface as well as the standard output.

- Open Eclipse if it is not opened yet.
- Double-click *HelloWorld.c* in the *myHelloWorld* project.
- First include the following two additional header files:

```
#include <unistd.h>
#include <fcntl.h>
```

- Then add the function *write_tty()*, which writes *n* bytes to the first serial interface (which, on the phyCORE-Z500PT, is connected to the system console */dev/console*):

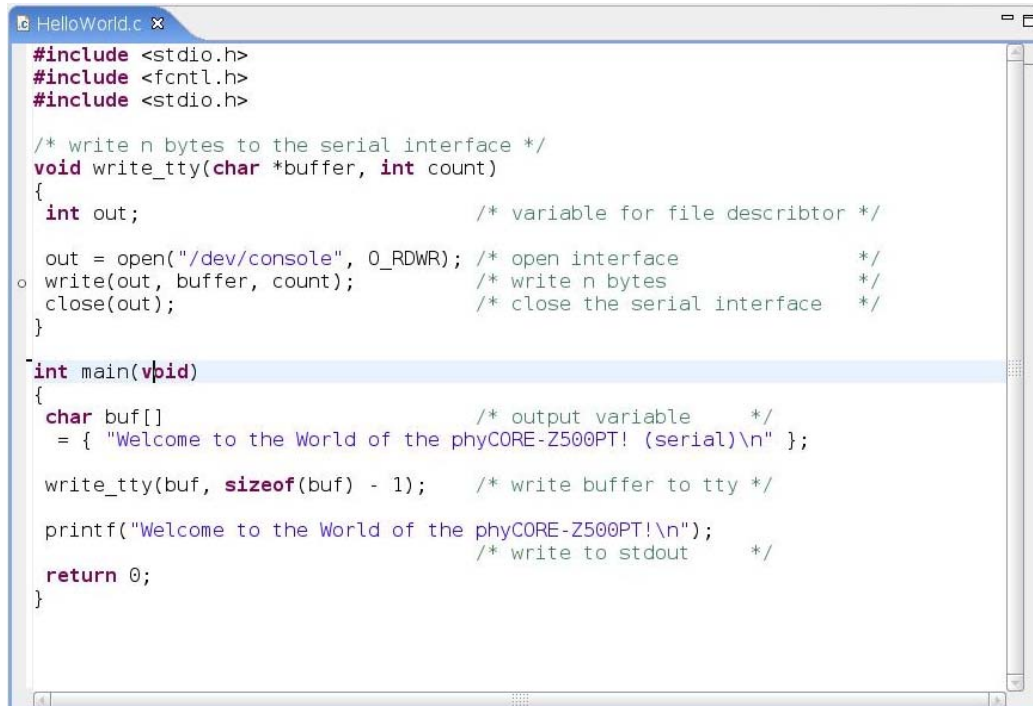
```
void write_tty(char *buffer, int count)
{
    int out;

    out = open("/dev/console", O_RDWR);
    write(out, buffer, count);
    close(out);
}
```

- Enter the following two lines in the *main()* function to declare the buffer and call the *write_tty()* function.

```
char buf[] = { "Welcome to the World of the
                phyCORE-Z500PT! (serial)\n" };
write_tty(buf, sizeof(buf) - 1);
```

In the next screenshot you can see the complete program.



```

HelloWorld.c
#include <stdio.h>
#include <fcntl.h>
#include <stdio.h>

/* write n bytes to the serial interface */
void write_tty(char *buffer, int count)
{
    int out;                /* variable for file descriptor */

    out = open("/dev/console", O_RDWR); /* open interface */
    write(out, buffer, count); /* write n bytes */
    close(out);              /* close the serial interface */
}

int main(void)
{
    char buf[]              /* output variable */
    = { "Welcome to the World of the phyCORE-Z500PT! (serial)\n" };

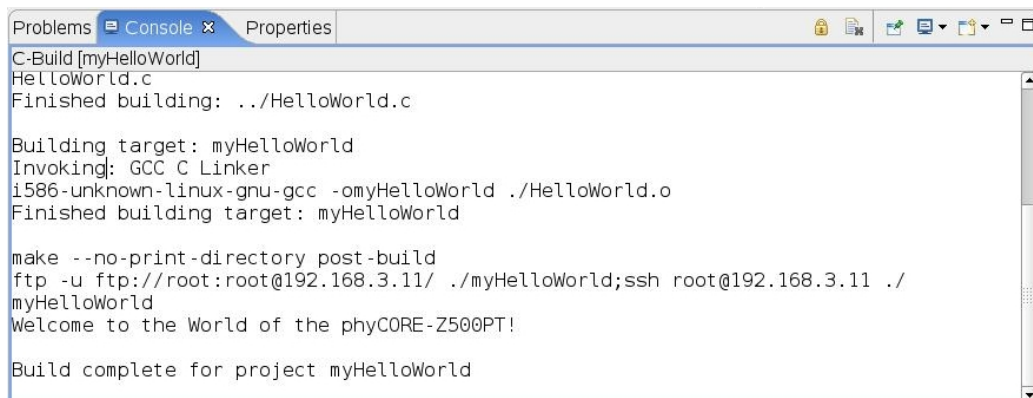
    write_tty(buf, sizeof(buf) - 1); /* write buffer to tty */

    printf("Welcome to the World of the phyCORE-Z500PT!\n");
    /* write to stdout */
    return 0;
}

```

- Save your program after changing the code.

The application will be compiled, built, copied to the target, and executed.



```

Problems Console Properties
C-Build [myHelloWorld]
HelloWorld.c
Finished building: ../HelloWorld.c

Building target: myHelloWorld
Invoking: GCC C Linker
i586-unknown-linux-gnu-gcc -omyHelloWorld ../HelloWorld.o
Finished building target: myHelloWorld

make --no-print-directory post-build
ftp -u ftp://root:root@192.168.3.11/ ./myHelloWorld;ssh root@192.168.3.11 ./
myHelloWorld
Welcome to the World of the phyCORE-Z500PT!

Build complete for project myHelloWorld

```

Executing the Program on the Target using Microcom

- Click the *Microcom* icon on the desktop.
- If you are not logged in, enter **root** and press **Enter**.
- Type *./myHelloWorld* to start the application.

- You will see the following output:

Welcome to the World of the phyCORE-Z500PT! (serial)
Welcome to the World of the phyCORE-Z500PT!

- Close Microcom.

When you start the application over an SSH session, you only see one output line. When you execute the program with Microcom, you see two output lines.



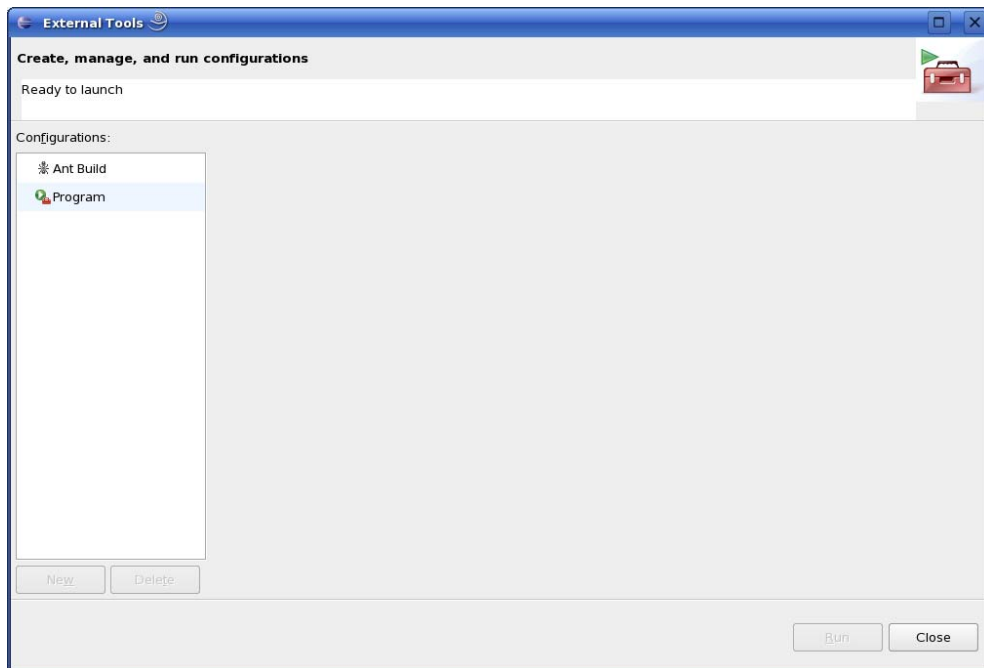
The first line is a direct output on the serial interface. You can't see this line in an SSH session, because you are connected over a TCP/IP connection to the target. With Microcom, however, you have direct access to serial interface, so you can also see the line that it written to the serial console.

In this passage you have changed an existing application. You also learned how to access the serial interface. First you called the function *open()* on the device */dev/console*. The return value of this function was a file descriptor. With the file descriptor you called the function *write()* to send *n* bytes to the device */dev/console*. After that, the file descriptor was closed with the function *close()*.

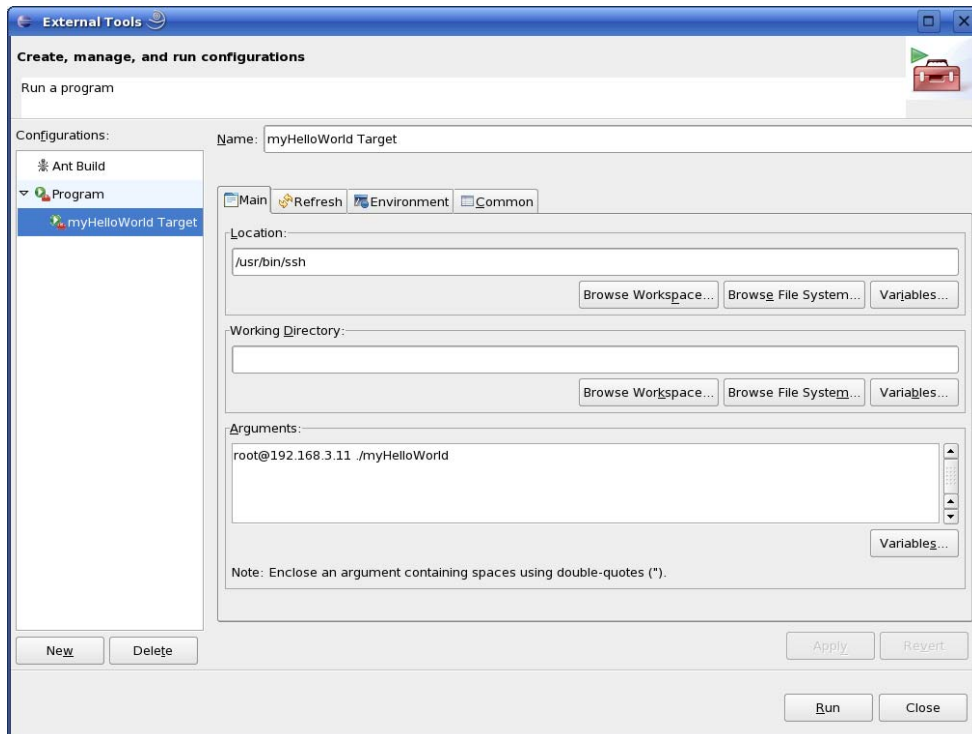
This procedure is in principle quite typical for Linux because Linux treats everything like a file.

3.5 Starting a Program out of Eclipse on the Target

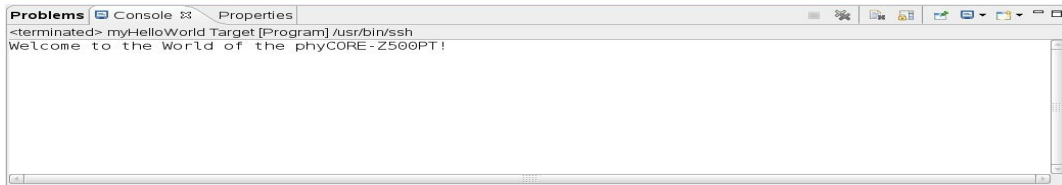
After compiling a project in Eclipse, the program is copied to the target and directly executed. A program can also be executed on the target without compiling a project. In the following section you will learn how to start a program on the target as an external tool.



- Select *Run* ► *External Tools* ► *External Tools* from the menu bar.
- Select *Program*.
- Select *New*.



- In the *Name* input field, enter: **myHelloWorld Target**
- Enter **/usr/bin/ssh** in the *Location* input field.
- Enter **root@192.168.3.11 ./myHelloWorld** into the *Arguments* field.
- Select *Apply*.



- Select *Run*.

If you want to execute the program the next time, you can use the *Run External Programs* button from the menu bar.



3.6 Automatically Starting the Program when Booting the Target

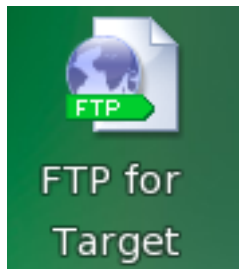
In this passage you will integrate the *myHelloWorld* program into the startup process of the target. When you have finished this part, the *myHelloWorld* application will be started automatically each time you are starting the target.



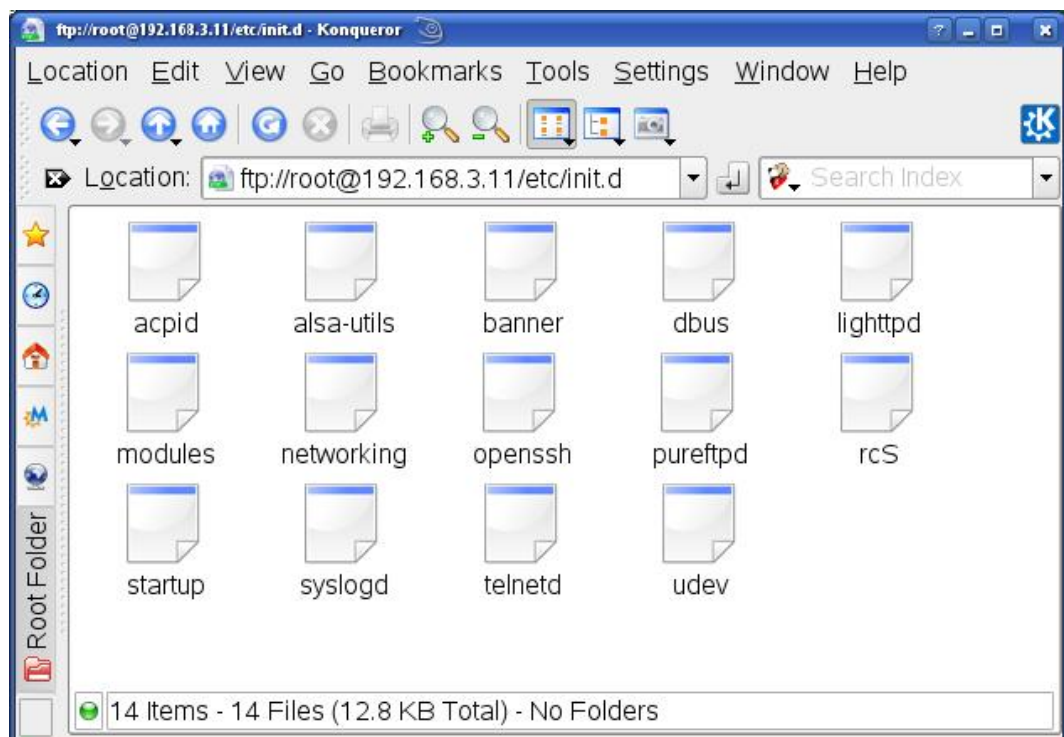
The scripts for controlling the system startup live in */etc/init.d*. These are executed directly or indirectly by */sbin/init*, the father of all processes. The configuration of */sbin/init* is placed in */etc/inittab*.

After system startup, */sbin/init* will switch to the default run level, as configured in */etc/inittab*. It calls the run level master script */etc/init.d/rcS* to start or stop services provided by the other scripts in */etc/init.d*. This is done by the help of symbolic links in the directory */etc/rc.d*. These links point to the actual startup scripts in */etc/init.d*.

First you will have to create a startup script in */etc/init.d*.



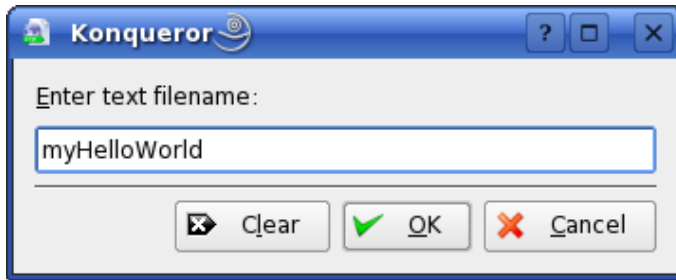
- Click the *FTP for Target* icon on your KDE desktop.



- Browse to the target's */etc/init.d*. If an authorization dialog should appear, just click on *OK* (no password is required for FTP access).

In the directory */etc/init.d* you can see the existing scripts.

- Right-click in the opened window and select *Create New* ► *Text File*.



- Enter *myHelloWorld*.
- Click *OK*.
- Right-click on *myHelloWorld* and select *Open with*.
- Enter **kwrite** and click *OK*.

The text editor *KWrite* starts with an empty document.

- Enter the following two lines:

```
#!/bin/sh
```

```
/home/myHelloWorld
```

- Select *File* ► *Save*.
- Close the *KWrite* window.
- Close the *FTP* window.



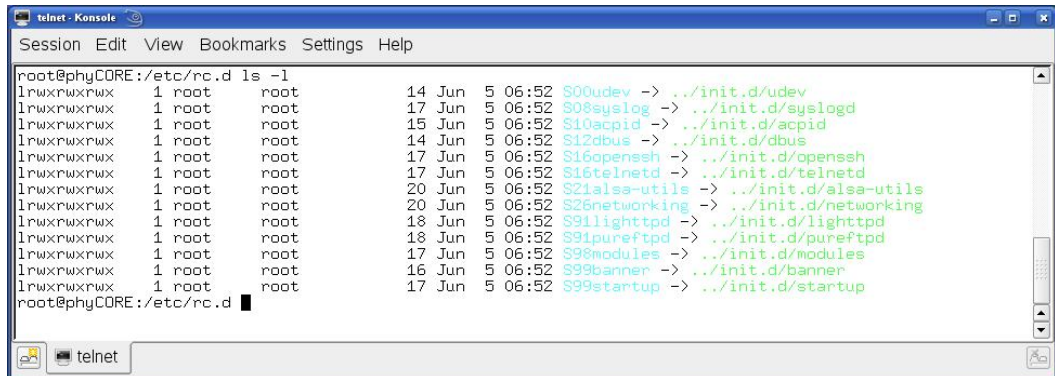
- Click the *Telnet for Target* icon on your desktop.
- Enter **root** and press **Enter** to login.

- The startup script we have just created with KWrite must be made executable in order to run it later:

chmod a+x /etc/init.d/myHelloWorld

- Change to the directory `/etc/rc.d`. Type the following command:

cd /etc/rc.d



```

telnet - Konsole
Session Edit View Bookmarks Settings Help

root@phyCORE:/etc/rc.d ls -l
lrwxrwxrwx 1 root root 14 Jun 5 06:52 S00udev -> ../init.d/udev
lrwxrwxrwx 1 root root 17 Jun 5 06:52 S08syslog -> ../init.d/syslogd
lrwxrwxrwx 1 root root 15 Jun 5 06:52 S10acpid -> ../init.d/acpid
lrwxrwxrwx 1 root root 14 Jun 5 06:52 S12dbus -> ../init.d/dbus
lrwxrwxrwx 1 root root 17 Jun 5 06:52 S16openssh -> ../init.d/openssh
lrwxrwxrwx 1 root root 17 Jun 5 06:52 S16telnetd -> ../init.d/telnetd
lrwxrwxrwx 1 root root 20 Jun 5 06:52 S21alsa-utils -> ../init.d/alsa-utils
lrwxrwxrwx 1 root root 20 Jun 5 06:52 S26networking -> ../init.d/networking
lrwxrwxrwx 1 root root 18 Jun 5 06:52 S91lighttpd -> ../init.d/lighttpd
lrwxrwxrwx 1 root root 18 Jun 5 06:52 S91pureftpd -> ../init.d/pureftpd
lrwxrwxrwx 1 root root 17 Jun 5 06:52 S98modules -> ../init.d/modules
lrwxrwxrwx 1 root root 16 Jun 5 06:52 S98banner -> ../init.d/banner
lrwxrwxrwx 1 root root 17 Jun 5 06:52 S99startup -> ../init.d/startup
root@phyCORE:/etc/rc.d █

```

- Enter **ls -l** to list the directory contents.

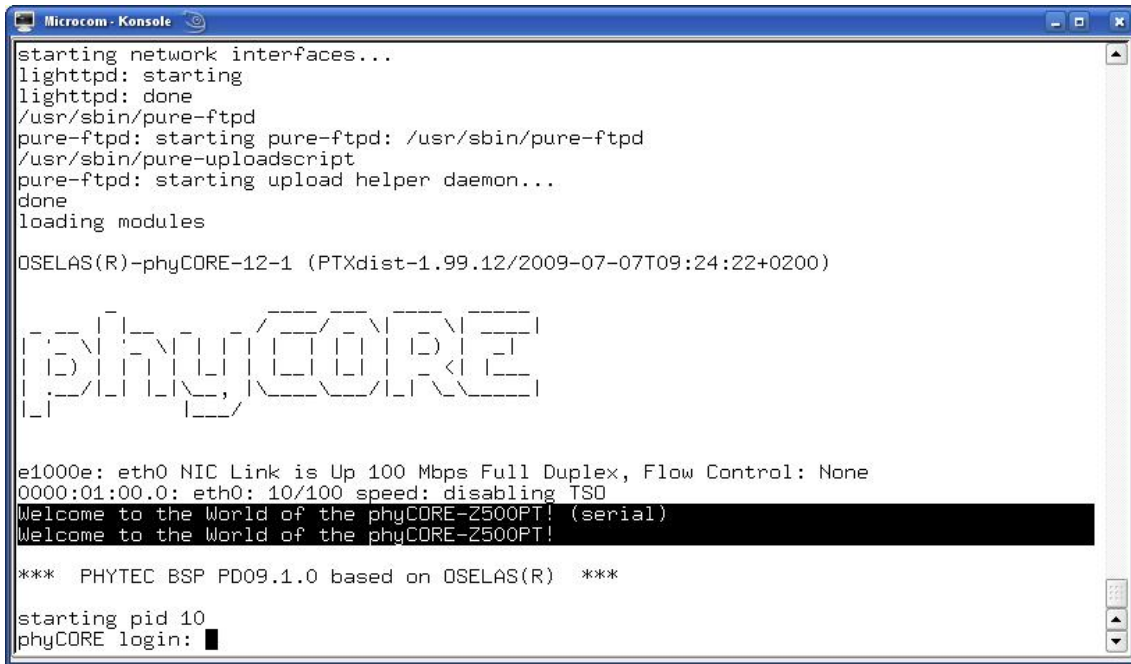
You can see the different links to the scripts in the directory `/etc/init.d`. The scripts are started in alphabetic order: The script `udev` is the first script started because the link starts with `S00...`, whereas `S99startup` will be started last.

To start your `myHelloWorld` application automatically when the system boots, you have to create a new link to the start script `/etc/init.d/myHelloWorld`.

- In `/etc/rc.d`, create a symbolic link which points to `/etc/init.d/myHelloWorld`. Enter the following command:

ln -s /etc/init.d/myHelloWorld /etc/rc.d/S99myHelloWorld

- Type **ls -l** again to check the newly created link.
- Close the window.
- Open Microcom.
- Push the RESET button on the target to restart your system.



```
Microcom - Konsole
starting network interfaces...
lighttpd: starting
lighttpd: done
/usr/sbin/pure-ftpd
pure-ftpd: starting pure-ftpd: /usr/sbin/pure-ftpd
/usr/sbin/pure-uploadscript
pure-ftpd: starting upload helper daemon...
done
loading modules
OSELAS(R)-phyCORE-12-1 (PTXdist-1.99.12/2009-07-07T09:24:22+0200)

phyCORE

e1000e: eth0 NIC Link is Up 100 Mbps Full Duplex, Flow Control: None
0000:01:00.0: eth0: 10/100 speed: disabling TSO
Welcome to the World of the phyCORE-Z500PT! (serial)
Welcome to the World of the phyCORE-Z500PT!

*** PHYTEC BSP PD09.1.0 based on OSELAS(R) ***

starting pid 10
phyCORE login: █
```

The program *myHelloWorld* now starts automatically on startup. Because its link in */etc/rc.d* starts with *S99...*, you should see *myHelloWorld*'s output near the output of the two other scripts that start with *S99...* (which print all sorts of version information).

- Close Microcom.

Now you can add your own programs to the root file system and start these programs automatically when your phyCORE-Z500PT boots.



You have successfully passed the “Getting More Involved” part of this Quick Start.

Chapter 4 Debugging an Example Project

**20 min**

TIME

In this chapter you will learn using the GNU debugger, GDB, on the host for remote debugging in conjunction with the GDB server on the target. GDB is the symbolic debugger of the GNU project and is arguably the most important debugging tool for any Linux system.

First you will start the GDB server on the target. Then you will configure the Eclipse platform and start the GNU debugger out of Eclipse using the Debug view.

The CDT extends the standard Eclipse Debug view with functions for debugging C/C++ code. The Debug view allows you to manage the debugging and running of a program in the Workbench. Using the Debug view you will be able to set breakpoints/watchpoints in the code and trace variables and registers. The Debug view displays the stack frame for the threads of each target you are debugging. Each thread in your program appears as a node in the tree, and the Debug view displays the process for each target you are running.

The GDB client is running on the host and is used to control the GDB server on the target, which in turn controls the application running on the target. GDB client and GDB server can communicate over a TCP/IP network connection as well as via a serial interface. In this Quick Start we will only describe debugging via TCP/IP.

4.1 Starting the GDB Server on the Target

In this passage you will learn how to start the GDB server on the target. The GDB server will be used to start and control the *myHelloWorld* program.

To debug a program with GDB, the program needs extended debugging symbols. With the default Eclipse settings, these debugging symbols have already been added when building the program.



- Open Microcom.
- Reboot the phyCORE-Z500PT
- Type **root** and press **Enter**.
- Start the GDB server:

gdbserver 192.168.3.11:10000 myHelloWorld

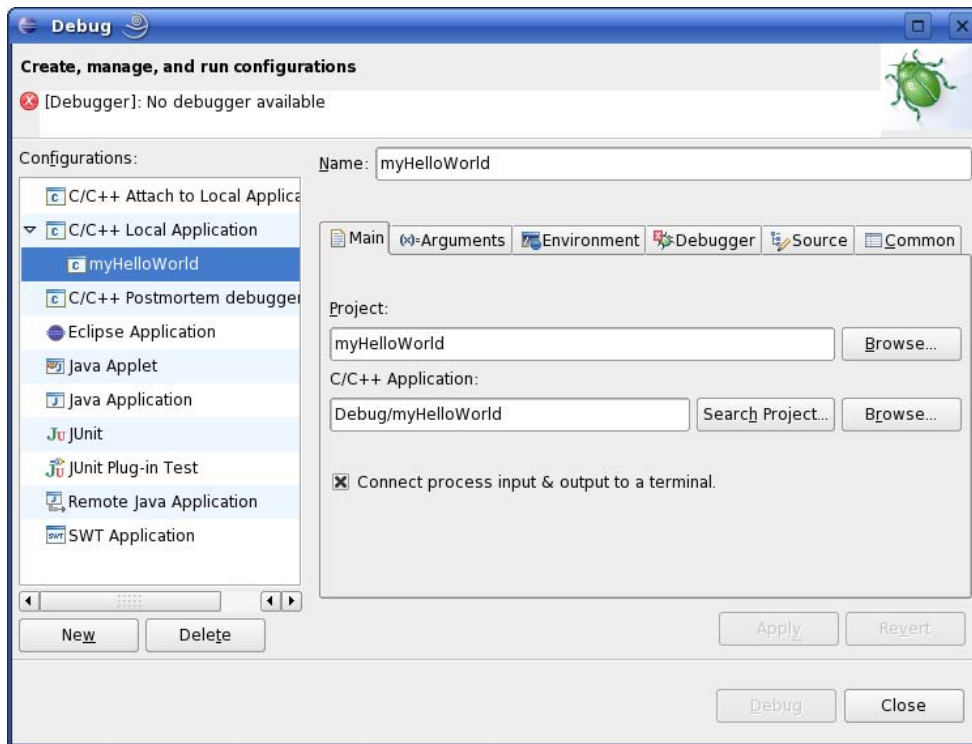
You have started the GDB server on the target. The GDB server is now waiting for connections on TCP port 10000.

4.2 Configuring and Starting the Debugger in Eclipse

In this passage you will learn how to configure your project settings to use Eclipse with the GNU debugger. After the configuration of your project settings, the GNU debugger will start and connect to the GDB server on the target.

- Start Eclipse if the application is not started yet.
- Select *myHelloWorld* in the *Navigator* window.
- Select *Run* ► *Debug* from the menu bar.

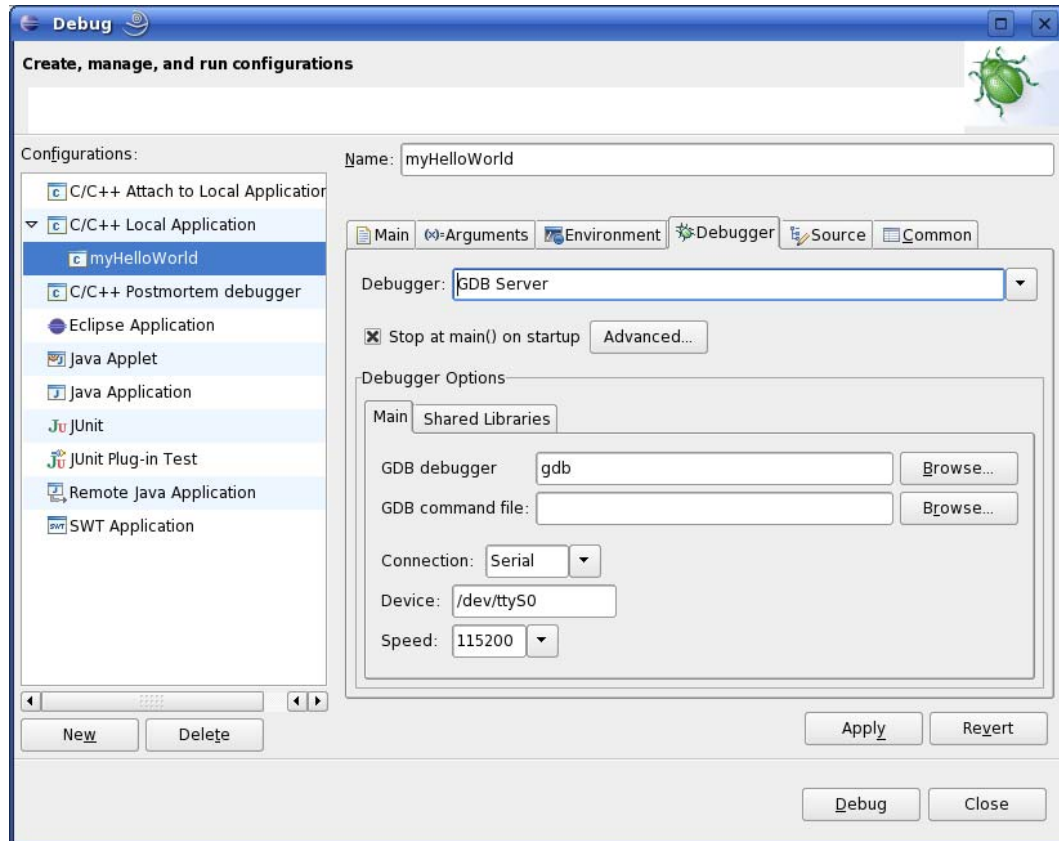
A dialog to create, manage, and run applications will appear.



- Select *C/C++ Local Application*.
- Click *New*.



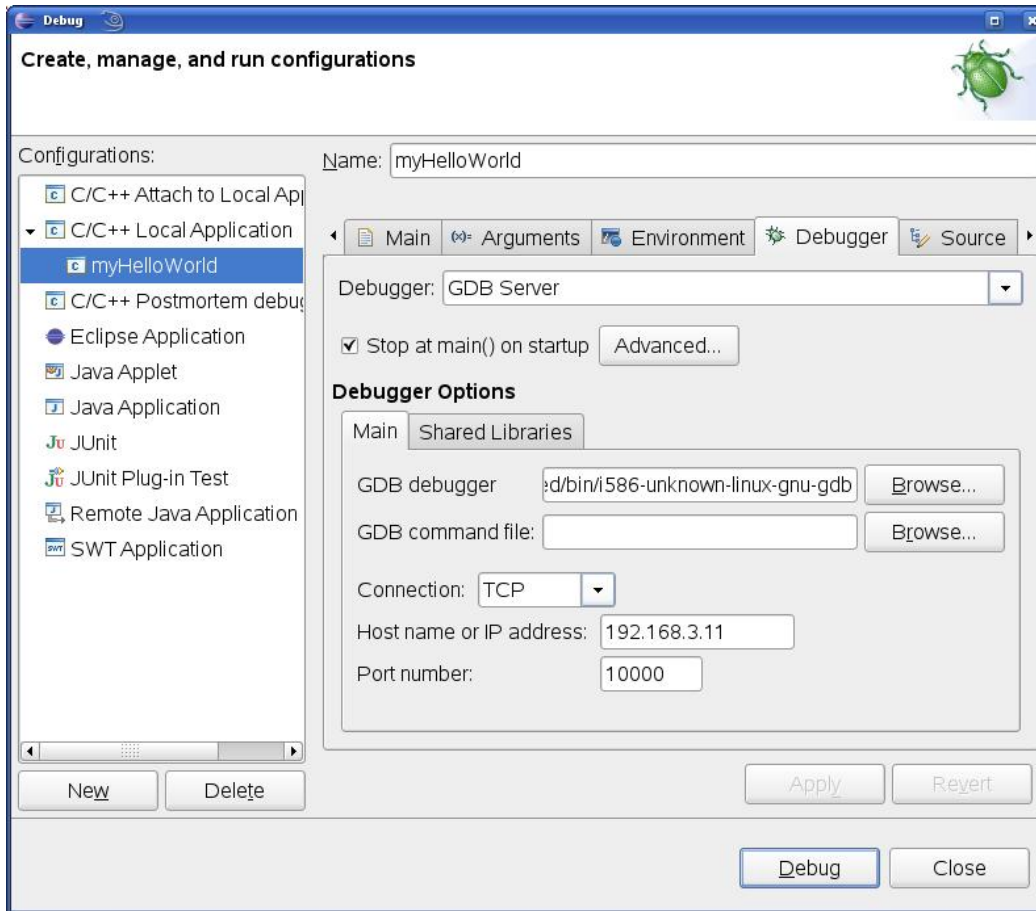
- Select the *Search Project* button.
- Click *OK* .



- Select the *Debugger* tab.
- Select *GDB Server* from the *Debugger* drop-down box.
- Click the *Browse* button right beside the *GDB debugger* input field.

A new dialog opens to choose the GDB executable.

- Click on *File System*.
- Navigate to the directory */opt/OSELAS.Toolchain-1.99.3/i586-unknown-linux-gnu/gcc-4.3.2-glibc-2.8-binutils-2.18-kernel-2.6.27-sanitized/bin*.
- Select the file *i586-unknown-linux-gnu-gdb*.
- Click *OK*.



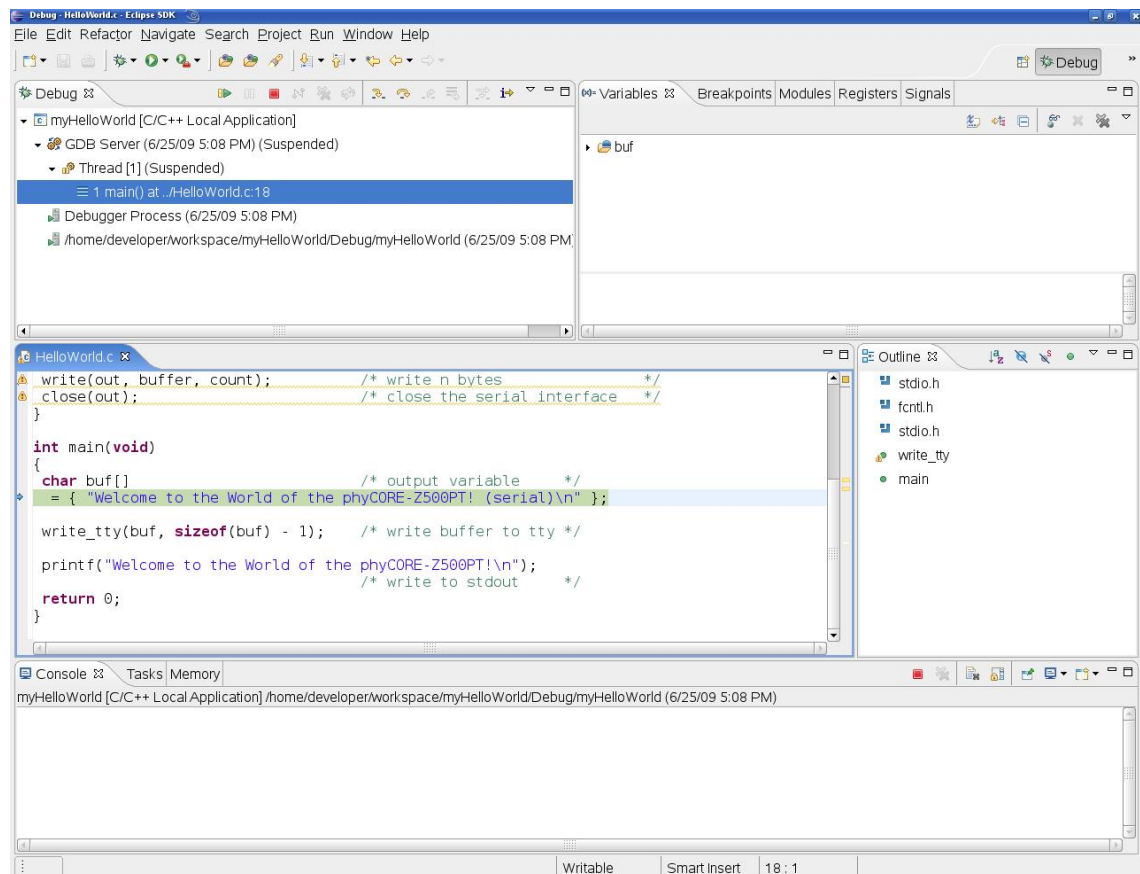
- From the *Connection* drop-down box, select *TCP*.
- Enter **192.168.3.11** (the target's IP address) in the *Host name* input field. The host's GDB will connect to this IP address to communicate with the target's GDB server.
- Click *Apply*.
- Click *Debug*.

A new dialog appears.



- Select *Yes* to switch to the *Debug* perspective.

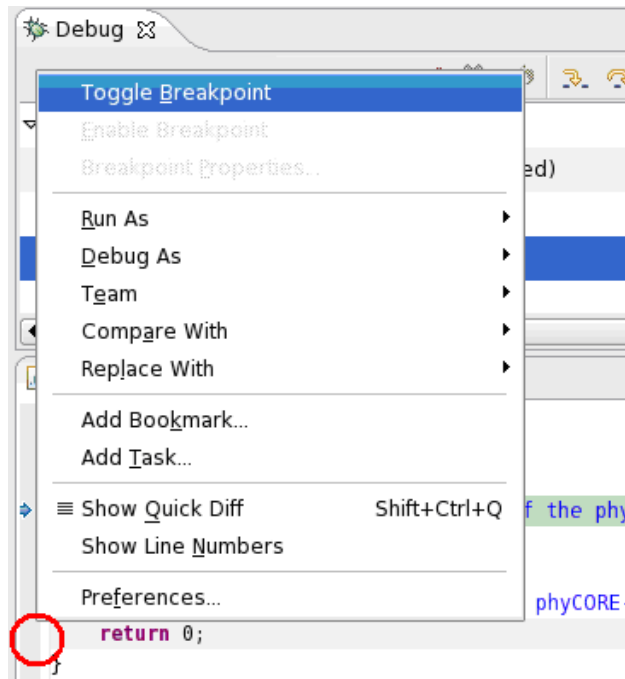
The *Debug* perspective opens and the debugger stops at the first line automatically. The host's GDB is now connected to the GDB server on the target.



You have configured your project for remote debugging. You have started the GNU debugger in Eclipse and connected the host's GDB with the target's GDB server. You can now start to debug the project.

4.3 Setting a Breakpoint

Now you will set a breakpoint in your program. The breakpoint will be set on the last line. If you resume the application, the debugger will stop on this line.

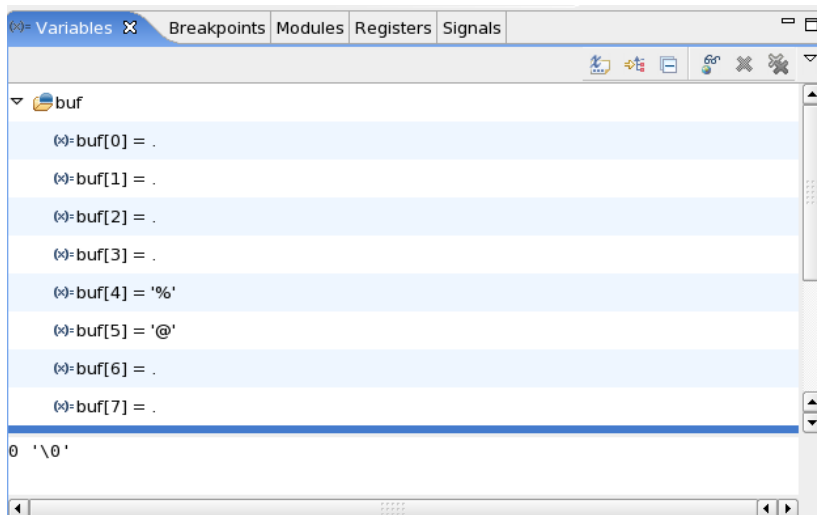



- Select the last line in *main()*.
- Right-click into the small grey border on the left-hand side and select *Toggle Breakpoint* to set a new breakpoint.

4.4 Stepping and Watching Variable Contents

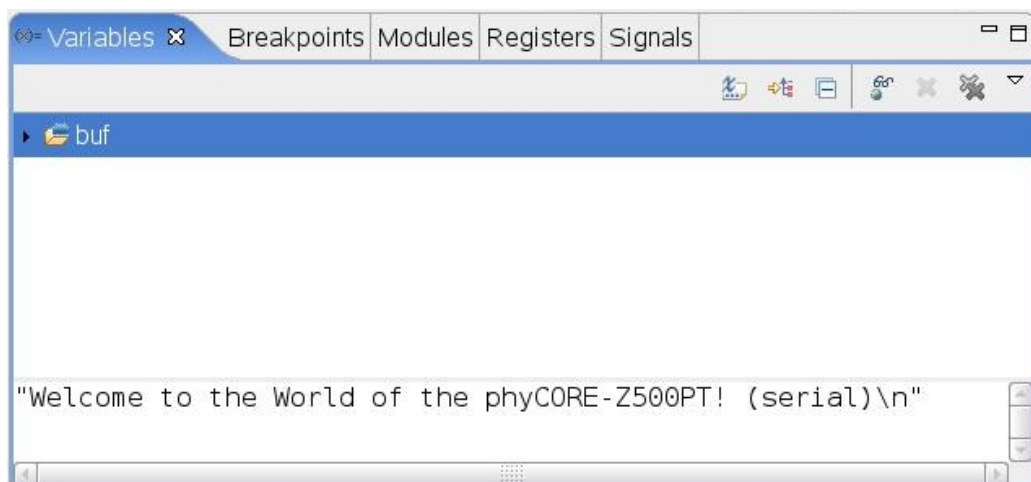
In this part you will step through the example project with the debugger. You will also learn how to watch the content of a variable.

- Expand *buf* in the *Variables* window.

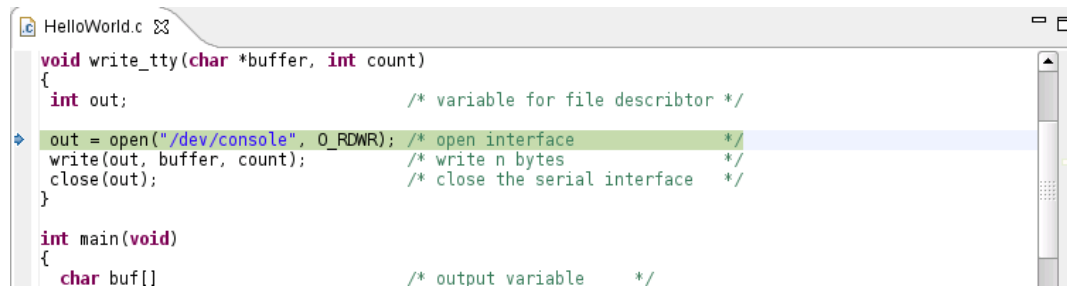


- Click on the *Step Over*  button in the *Debug* window to step to the next line.

You will see the content of the *buf* variable in the *Variables* window.



- Click on the variable *buf*.



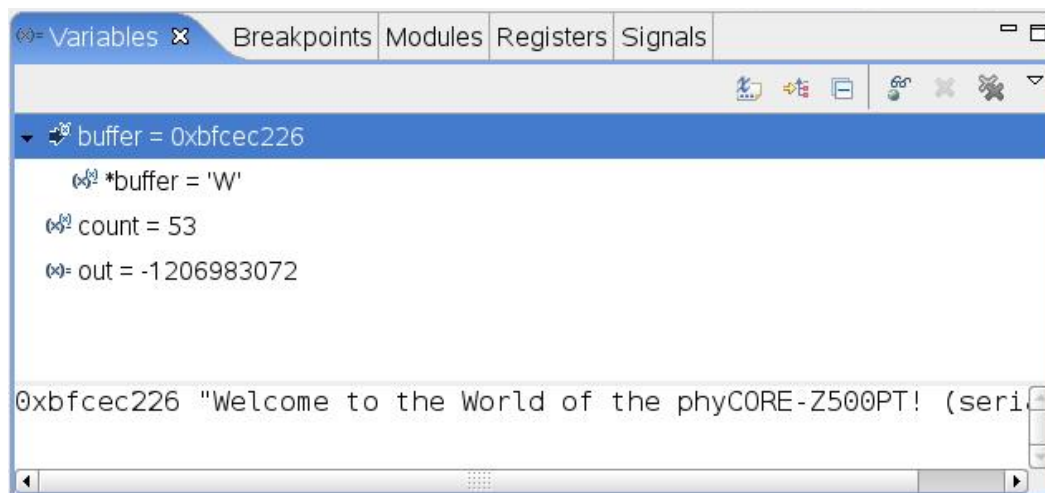
```
void write_tty(char *buffer, int count)
{
    int out; /* variable for file descriptor */
    out = open("/dev/console", O_RDWR); /* open interface */
    write(out, buffer, count); /* write n bytes */
    close(out); /* close the serial interface */
}

int main(void)
{
    char buf[] /* output variable */
```

- Then click on the button *Step into*  to enter the function `write_tty()`.

The debugger stops in `write_tty()`.

You will see the following variable window:



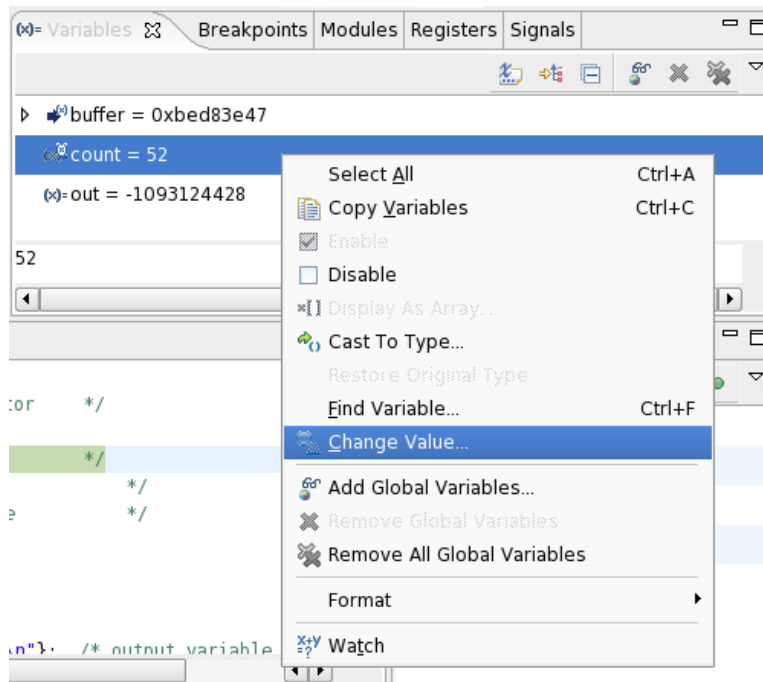
```
Variables Breakpoints Modules Registers Signals
buffer = 0xbfcec226
  *buffer = 'W'
  count = 53
  out = -1206983072
0xbfcec226 "Welcome to the World of the phyCORE-Z500PT! (seri
```

- Click on the variable `buffer`.

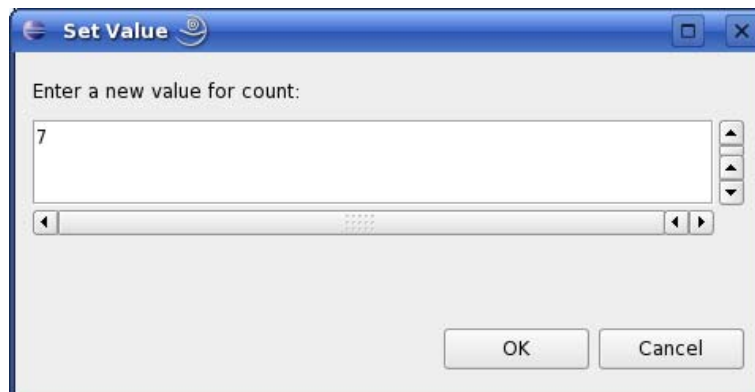
You will probably see a different address at the `buffer` pointer. Remember what address is shown in your case; you will need this address later.


4.5 Changing Variable Values

In this section you will change the value of a variable. At the end of this part you will see the effect of this change.



- Select the *count* variable in the *Variables* window.
- Right-click on *count* and select *Change Value*.



- Change the value of *count* to **7** and click *OK*.
- Open Microcom if the application is not already opened.
- Go back to Eclipse.
- Click the *Step Over*  button **two times**.
- Change to Microcom.

```

root@phyCORE:~$ gdbserver 192.168.3.10:10000 myHelloWorld
Process myHelloWorld created; pid = 434
Listening on port 10000

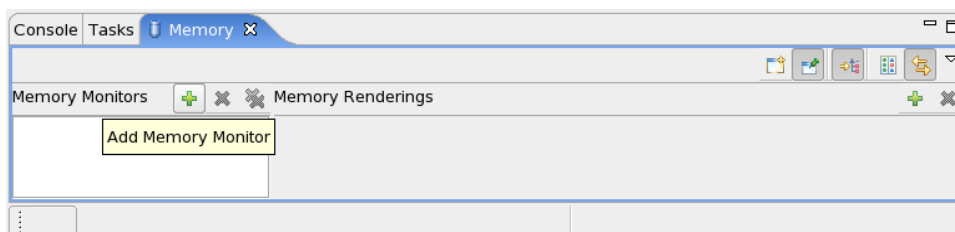
Is
Quit
Remote debugging from host 192.168.3.10
Welcome

```

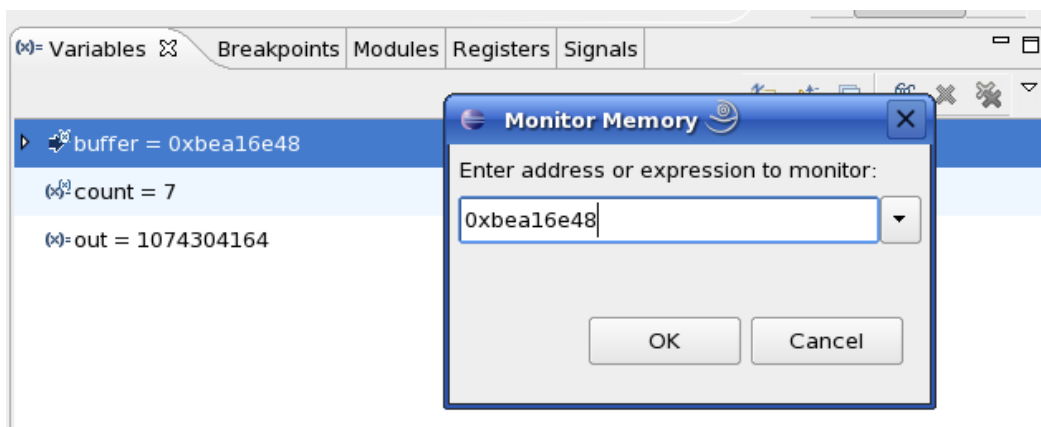
You will see the output *Welcome* in the Microcom window. This means that due to changing the *counter* variable's value, instead of printing the full "Welcome to the World of the phyCORE-Z500PT" string, only the first seven characters of the buffer were written to the screen.

4.6 Using the Memory Monitor

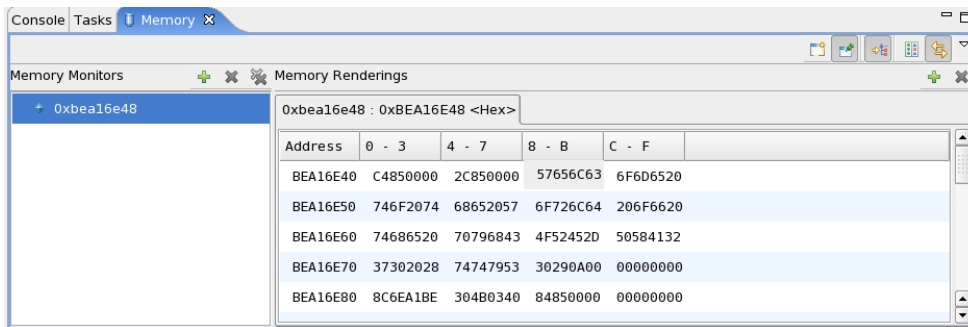
In the last section of this chapter you will use the memory monitor to watch the content at a memory address.



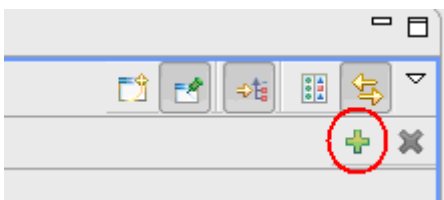
- Select the *Memory* tab.
- Click *Add Memory Monitor*.



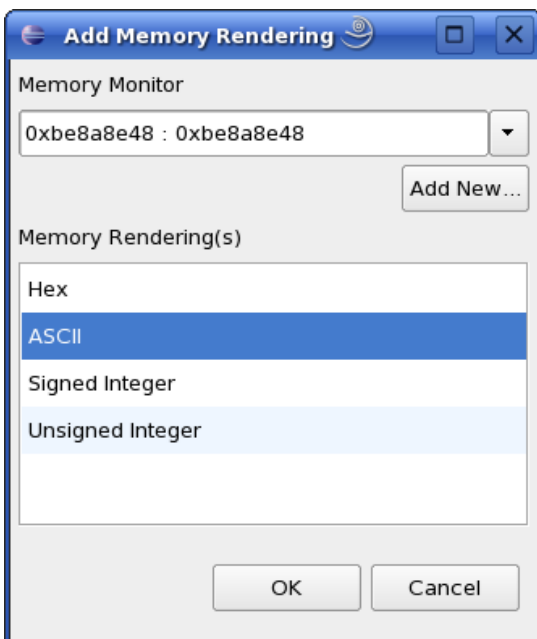
- Enter the address of *buffer* and click on *OK*. Remember that the variable's address might differ on your system.



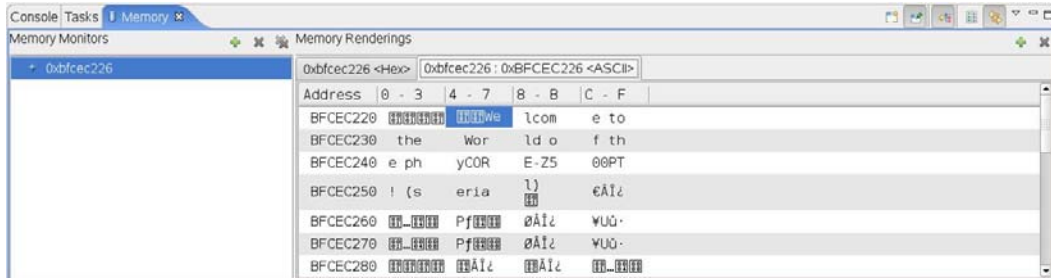
- Change the window size.



- Click *Add Rendering*.

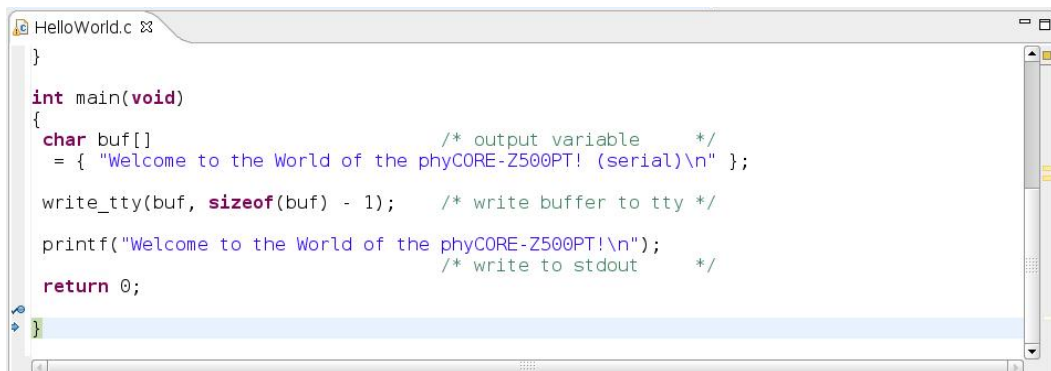


- Select *ASCII* and click *OK*.




You can see the contents of the variable *buffer* at the address *0xbeda6e48* (or whatever address is used on your system).

- Now click the *Resume*  button from the menu bar.



The debugger stops at the breakpoint in the last line of *main()*.

- Click the *Resume*  button to end the application.



You have successfully passed the debugging chapter. You are now able to configure and use Eclipse for remote debugging. You can step through a project, watch and change the content of variables, and you can use the memory monitor to view the content at a memory address.

Chapter 5 Further Information

In the PTXdist User Manual you can find further information. The manual is located on your setup CD-ROM in the directory:

PHYTEC/PCM041 phyCORE-Z500PT/Linux-Kit/BSP/OSELAS

The PTXdist User Manual includes information on the following topics:

- Installation and Configuration of PTXdist.
- Building and using a tool chain.
- Create and activate a PTXdist project.
- Running the phyCORE-Z500PT from network only.
- Running the phyCORE-Z500PT stand-alone.
- The phyCORE-Z500PT's Board Support Package (BSP).
- Accessing Peripherals on the phyCORE-Z500PT.

Chapter 6 **Summary**

This Quick Start manual provided a general “Rapid Development Kit” description, as well as software installation advice and an example program enabling quick out-of-the-box start-up of the phyCORE-Z500PT in conjunction with the Eclipse IDE and GNU C/C++ software tools.

In the *Getting Started* section you learned how to configure your host to provide a basis for working with your target platform. You installed the Rapid Development Kit software and learned how to copy and run a program on the target.

In the *Getting More Involved* section you got step-by-step instructions on how to configure and build a new kernel, modify the example application, create and build new projects, and copy programs to your phyCORE-Z500PT using Eclipse.

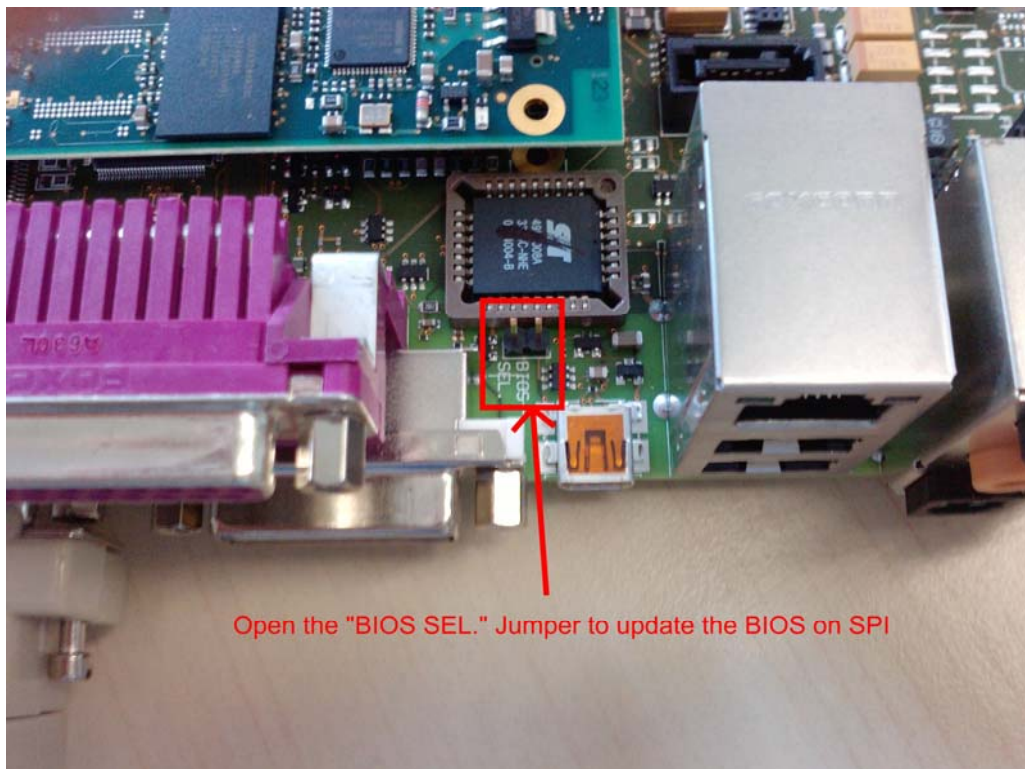
The *Debugging* part of this Quick Start gave you information on setting up and using the GNU debugger with the Eclipse IDE. You learned how to set breakpoints, watching and changing variable contents, and using the memory monitor.

Appendix A Updating BIOS on SPI

In this chapter you will learn how to update the BIOS with the help of the SPIprog – Tool on the SPI. In our example we show how to update to the delivery status BIOS.

First the phyCORE-Z500PT has to be prepared.

- We take a look on the “*BIOS SEL.*” Jumper on the base board. This jumper must be open. Otherwise not the BIOS on the SPI will be updated, but the BIOS on the base board.



- Start *Microcom*.
- Connect the board with the power supply, boot it and enter the root password.

After you have successfully logged in it is time to copy the *SPIprog – Tool* and the BIOS via FTP into the target. In the destination folder of the Linux-

Z500PT-Setup you find a subfolder named *SPIprog* in which you find the delivery status BIOS and the SPIprog – Tool.

Now we open the terminal window and change to the directory:

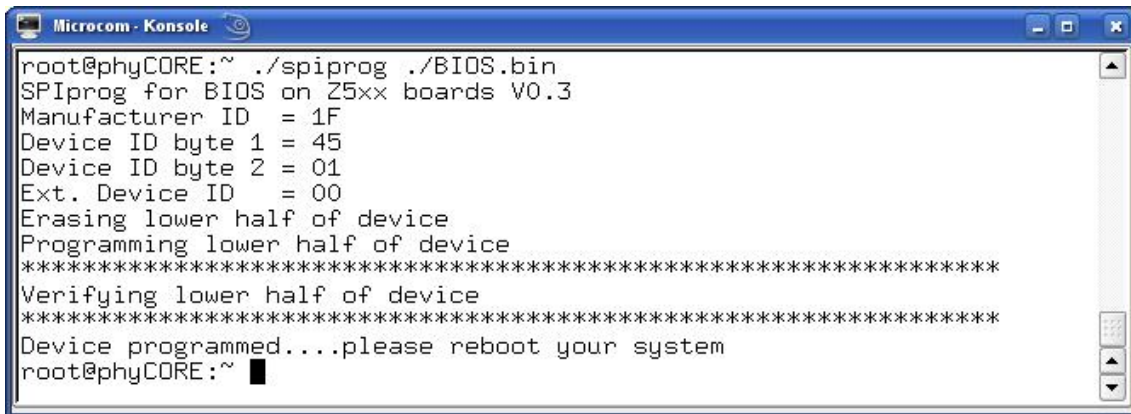
- `cd /usr/local/share/Z500PT-Kit/SPIprog/`
- Copy the files via ftp into the target:

`ftp -u ftp://root:root@192.168.3.11/ spiprogram`

`ftp -u ftp://root:root@192.168.3.11/ BIOS_Z500PT.bin`

- Finally we can start the *SPIprog* – tool. Switch to *Microcom* and type:

`./spiprogram ./BIOS_Z500PT.bin`



```

Microcom - Konsole
root@phyCORE:~ ./spiprogram ./BIOS.bin
SPIprog for BIOS on Z5xx boards V0.3
Manufacturer ID = 1F
Device ID byte 1 = 45
Device ID byte 2 = 01
Ext. Device ID = 00
Erasing lower half of device
Programming lower half of device
*****
Verifying lower half of device
*****
Device programmed....please reboot your system
root@phyCORE:~ █

```

- Restart your system with the updated BIOS.



TIP

If something goes wrong with your new BIOS, you could find the old in:

`/usr/local/share/Z500PT-Kit/SPIprog/BIOS_Z500PT.bin.`



SUCCESS

We have successfully updated the BIOS on the SPI.

Document: phyCORE-Z500PT with Linux
Quick Start Instructions
Document Number: L-738e_1 August 2009

How would you improve this manual?

Did you find any mistakes in this manual? page

Submitted by:

Customer number: _____

Name: _____

Company: _____

Address: _____

Return to:

PHYTEC Messtechnik GmbH

Robert-Koch-Str. 39

D-55129 Mainz

Fax: +49 (6131) 9221-26