

MPC8568E Development Kit Kit Configuration Guide

Revised: 21 Feb 2008



Freescale™, the Freescale logo, and CodeWarrior are trademarks or registered trademarks of Freescale Semiconductor, Inc. in the U.S. and/or other countries. The Power Architecture™ and Power.org word marks are trademarks and service marks licensed by Power.org. All other product or service names are the property of their respective owners. Copyright © 1999-2008 by Freescale Semiconductor, Inc. All rights reserved."

Copyright © 2008 by Freescale Semiconductor, Inc. All rights reserved.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

How to Contact Us

Corporate Headquarters	Freescale Semiconductor, Inc. 7700 West Parmer Lane Austin, TX 78729 U.S.A.
World Wide Web	http://www.freescale.com/codewarrior
Technical Support	http://www.freescale.com/support

Table of Contents

1	Introduction	5
	Development Kit Components	5
	MPC8568E-MDS-PB Board	6
	CodeWarrior™ Development Studio for Power Architecture™ Processors, Professional/Linux® Application Edition	7
	CodeWarrior™ Development Studio for Power Architecture™ Processors, Linux® Platform/Linux Application Edition	7
	NetComm Device Drivers	8
	CodeWarrior™ QUICC Engine™ Utility	8
	USB TAP Debug Probe	9
	MPC8568E-MDS-PB Linux® Board Support Package	9
	How this Manual is Organized	10
	Table of Acronyms	12
2	Preparing the MPC8568E-MDS-PB Board	15
	Setting Board Jumpers and Switches	15
	Preparing for Bare Board Development or for Linux® OS Customization	15
	Preparing for Linux® Application Development	17
3	Installing CodeWarrior™ Development Studio	21
	Installing the Bare Board Development Tools	21
	System Requirements	21
	Installation Procedure	22
	Installing the Linux® Development Tools	25
	System Requirements	25
	Installation Procedure	25
	Registering and Licensing Your CodeWarrior™ Product	28
4	Installing and Using the Netcomm Device Drivers	31
	Installing the Netcomm Device Driver Package	32
	Using the Netcomm Device Drivers	34

Table of Contents

5	Using the CodeWarrior™ QUICC Engine™ Utility	43
	Installing the QUICC Engine™ Utility	43
	System Requirements	43
	Installation Procedure	44
	Using the QUICC Engine™ Utility	45
6	Using the MPC8568E-MDS-PB BSP	55
	System Requirements	55
	Installing LTIB and the MPC8568E-MDS-PB Board Support Package.	56
	Index	59

Introduction

This document explains how to use each component of the MPC8568E-MDS-PB modular development kit. Specifically, the document explains how to:

- Set up the MPC8568E-MDS-PB processor board and connect it to your development PC.
- Install CodeWarrior™ Development Studio for Power Architecture™ Processors, Professional Edition.
- Install CodeWarrior Development Studio for Power Architecture Processors, Linux® Platform Edition.
- Use these CodeWarrior tools to create a simple project, build this project, and download and debug the resulting application on the processor board.
- Install the Netcomm Device Drivers package and use it to run protocol-based use cases on the processor board. This package includes the QUICC Engine utility.
- Use the USB TAP debug probe.
- Install the MPC8568E Linux board support package and the Linux Target Image Builder (LTIB) tools framework.

The sections of this chapter are:

- [“Development Kit Components”](#)
- [“How this Manual is Organized” on page 10](#)
- [“Table of Acronyms” on page 12](#)

Development Kit Components

The MPC8568E development kit consists of these items:

- An MPC8568E-MDS-PB processor board (PB) that is preflashed with an embedded Linux image.
- A Linux board support package (BSP) tailored to the MPC8568E-MDS-PB board, along with tools for generating custom Linux images from the BSP.
- Windows®- and Linux-hosted CodeWarrior software development tools along with debug hardware that lets these tools interact with the MPC8568E-MDS-PB board.
- The NetComm Device Drivers package.

Introduction

Development Kit Components

- The CodeWarrior QUICC Engine Utility (QEU), a GUI-based tool that generates C source code that initializes the QUICC Engine block of the MPC8568E processor.
- The CodeWarrior USB TAP debug probe, which lets you control the MPC8568E processor at the hardware level, as well as manipulate flash memory.

These sections describe the MPC8568E kit components:

- [“MPC8568E-MDS-PB Board”](#)
- [“CodeWarrior™ Development Studio for Power Architecture™ Processors, Professional/Linux® Application Edition” on page 7](#)
- [“CodeWarrior™ Development Studio for Power Architecture™ Processors, Linux® Platform/Linux Application Edition” on page 7](#)
- [“NetComm Device Drivers” on page 8](#)
- [“CodeWarrior™ QUICC Engine™ Utility” on page 8](#)
- [“USB TAP Debug Probe” on page 9](#)
- [“MPC8568E-MDS-PB Linux® Board Support Package” on page 9](#)

MPC8568E-MDS-PB Board

The MPC8568E-MDS-PB board helps you build applications based on the MPC8568E PowerQUICC III processor family. The board provides a development platform with which you can work before production hardware is available.

The processor board hosts an MPC8568E processor. This processor incorporates the QUICC Engine, a communications engine that supports a wide range of protocols. Additional enhancements include the e500 Power Architecture core (which scales up to 1.33 GHz), a DDR2 memory controller, and an integrated security engine.

Using these on-board resources in conjunction with the CodeWarrior debugger, you can download and run programs, set breakpoints, display memory and registers, and connect proprietary hardware with the expansion connectors. These programs can then be incorporated into any production application that uses an MPC8568E processor.

For embedded Linux development, the MPC8568E-MDS-PB board is pre-flashed with a Linux image derived from a board support package (BSP) developed by Freescale. Each time the board is powered up, it boots this Linux image.

Finally, for additional connectivity, you can attach the MPC8568E-MDS-PB board to the optional 85xx platform I/O board (PIB). For even more connectivity options, you can attach up to 4 PMC cards to the PIB.

CodeWarrior™ Development Studio for Power Architecture™ Processors, Professional/Linux® Application Edition

CodeWarrior Development Studio for Power Architecture Processors, Professional Edition combines the CodeWarrior Integrated Development Environment (IDE) with the latest CodeWarrior compiler, linker, and debugger technology.

This CodeWarrior tool chain lets you generate programs in executable and linkable (.elf) format. Because .elf format programs are self-sufficient, you can download and run such programs on the MPC8568E-MDS-PB processor board without using the Linux image in the board's flash memory.

The CodeWarrior debugger provides a rich set of features designed to help you quickly find and repair software defects, including:

- Support for breakpoints, watchpoints, and event points
- Display of stack crawls, memory, registers, register details, and cache windows
- Support for both the STABS and DWARF (versions 1 and 2) debugging information formats

CodeWarrior™ Development Studio for Power Architecture™ Processors, Linux® Platform/Linux Application Edition

CodeWarrior Development Studio for Power Architecture Processors, Linux Platform Edition combines the CodeWarrior Integrated Development Environment (IDE) with a GNU GCC toolchain and the latest CodeWarrior debugger technology.

This CodeWarrior product lets you develop and deploy embedded Linux applications quickly and easily. Further, the tool includes GCC v2.95.3, v3.3.2, and v3.4.3. Finally, if these GCC toolchains do not meet your need, you can configure the CodeWarrior IDE to use any GCC toolchain you want.

The GCC C/C++ compiler is optimized for the Freescale PowerQUICC III family of processors. You can generate PowerQUICC III applications and libraries that conform to the Power Architecture embedded application binary interface (EABI) by using this compiler along with the GCC linker for PowerQUICC III processors.

The CodeWarrior debugger provides a rich set of features designed to help you quickly find and repair software defects, including:

- Support for debugging multi-threaded and multi-process applications
- Support for breakpoints, watchpoints, and event points

- Display of stack crawl, memory, registers, register details, and cache windows
- Support for both the STABS and DWARF (versions 1 and 2) debugging information formats

NetComm Device Drivers

The purpose of the NetComm Device Drivers package is to help programmers develop software for the MPC8568E family of Power Architecture processors. The package consists of a set of powerful and flexible drivers for the peripherals and protocols supported by the MPC8568E processor's QUICC Engine coprocessor.

Each driver presents an API that abstracts the underlying hardware/firmware in a modular way. As a result, a client application can interface with the MPC8568E processor at the protocol level, thereby avoiding the complex task of directly manipulating the registers of the processor's on-board communications peripherals.

The drivers in the NetComm Device Drivers package can be used with or without the [CodeWarrior™ QUICC Engine™ Utility](#).

NOTE This kit supports both Linux and bare board development, but the NetComm Device Drivers package is just for bare board development.

CodeWarrior™ QUICC Engine™ Utility

The CodeWarrior QUICC Engine Utility (QEU) makes it easy to configure the MPC8568E processor to use any protocol or driver parameter supported by the device.

The QEU presents a graphical user interface with which you can:

- Select the QUICC Engine device blocks that your program will use.
- Assign any protocol and any interface to each device block selected.
- Modify any device, microcode, or peripheral driver API parameter.
- Configure a multi-protocol system.
- Generate C language source code that uses the device driver API (see [“NetComm Device Drivers” on page 8](#)) to configure the selected device blocks.
- Use dozens of existing use cases — simple, protocol-based applications to help you understand how to use the drivers API.

In addition, the QEU's GUI provides context-sensitive help and automatically detects configuration conflicts. Further, the GUI provides access to online assistance that helps you resolve these conflicts.

To use the QUICC Engine Utility with the CodeWarrior tools, you add the source code files the QEU generates to a MPC8568E CodeWarrior project. When you download the

binary file generated by this project to an MPC8568E-MDS-PB board, the selected QUICC Engine device blocks are configured according to the choices made in the QEU.

NOTE This kit supports both Linux and bare board development; however, the QUICC Engine Utility currently supports bare board development only.

USB TAP Debug Probe

The CodeWarrior USB TAP debug probe lets you control the MPC8568E processor's execution at the hardware level.

If you are writing software for execution on a bare board, you can use the USB TAP to download your programs to the MPC8568E-MDS-PB processor board and debug this software with minimal intrusion. If you are an embedded Linux developer, you can use the USB TAP and the CodeWarrior debugger to debug the Linux kernel and loadable kernel modules. Finally, you can use the USB TAP in conjunction with the CodeWarrior flash programmer to manipulate the flash memory of the processor board.

TIP Using Freescale's Ethernet TAP debug probe, your team can share a single MPC8568E-MDS-PB processor board over an IP network.

MPC8568E-MDS-PB Linux® Board Support Package

The MPC8568E-MDS-PB board support package (BSP) includes all elements required to build, deploy, and debug the Linux operating system, device drivers, kernel modules, and applications for the MPC8568E-MDS-PB processor board.

You can generate custom Linux images using this BSP in conjunction with the Linux Target Image Builder (LTIB) tools framework. Using the CodeWarrior flash programmer and the USB TAP probe, you can then write these images to the flash memory of your MPC8568E-MDS-PB processor board.

How this Manual is Organized

[Table 1.1](#) provides a brief description of each chapter in this document.

Table 1.1 Chapter Descriptions

Chapter Title	Description of Contents
Preparing the MPC8568E-MDS-PB Board	Explains how to connect the MPC8568E-MDS-PB board to your development PC for: • Bare board development • Linux OS customization • Linux application development
Installing CodeWarrior™ Development Studio	Explains how to install CodeWarrior Development Studio for Power Architecture Processors, Professional Edition. Explains how to install CodeWarrior Development Studio for Power Architecture Processors, Linux Platform Edition.
Creating and Debugging an MPC8568E-MDS-PB Project	Explains how to create a project for a bare board application using the CodeWarrior bare board toolchain, build this project, and download and debug the resulting <code>.elf</code> format binary on the MPC8568E-MDS-PB board. Explains how to create a project for a Linux application using the CodeWarrior embedded Linux toolchain, build this project, and download and debug the resulting application on the MPC8568E-MDS-PB board.
Installing and Using the Netcomm Device Drivers	Explains how to install and use the Netcomm Device Drivers package.
Using the CodeWarrior™ QUICC Engine™ Utility	Explains how to install and use the CodeWarrior QUICC Engine utility.
Using the MPC8568E-MDS-PB BSP	Explains how to install and use the MPC8568E-MDS-PB board support package and the Linux Target Image Builder (LTIB).

Depending upon the type of development you plan to do, you should read different sections of this document.

- If you are writing programs that will run on a bare board, read these sections:
 - [“Setting Board Jumpers and Switches” on page 15](#)
 - [“Preparing for Bare Board Development or for Linux® OS Customization” on page 15](#)
 - [“Installing the Bare Board Development Tools” on page 21](#)
 - [“Registering and Licensing Your CodeWarrior™ Product” on page 28](#)
 - [“Creating and Debugging a Bare Board Project” on page 43](#)
- If your bare board programs use the QUICC Engine coprocessor, you should also read these sections:
 - [“Installing and Using the Netcomm Device Drivers” on page 31](#)
 - [“Using the CodeWarrior™ QUICC Engine™ Utility” on page 43](#)
- If you are writing Linux applications that will run on a Linux image in flash memory, read these sections:
 - [“Setting Board Jumpers and Switches” on page 15](#)
 - [“Preparing for Linux® Application Development” on page 17](#)
 - [“Installing the Linux® Development Tools” on page 25](#)
 - [“Registering and Licensing Your CodeWarrior™ Product” on page 28](#)
 - [“Creating and Debugging a Linux® Application Project” on page 56](#)
- If you need to create a custom embedded Linux image and write it to the processor board’s flash memory, read these sections:
 - [“Setting Board Jumpers and Switches” on page 15](#)
 - [“Preparing for Bare Board Development or for Linux® OS Customization” on page 15](#)
 - [“Using the MPC8568E-MDS-PB BSP” on page 55](#)

Table of Acronyms

[Table 1.2](#) lists and defines acronyms used in this manual.

Table 1.2 Acronyms Used in this Manual

Acronym	Description
ADS	Application Development System A Freescale board that is a platform for developing custom embedded software for a particular Freescale microcontroller.
COP	Common On-chip Processor Name of connector to which you attach JTAG-type test equipment.
DWARF	Debug With Arbitrary Record Format. A portable debugging information format for ELF files. This information allows source-level debugging of ELF files.
ELF	Executable and Linkable Format The object, executable program, shared object, and core file format for Linux and many other UNIX operating systems.
IDE	Integrated Development Environment A system that supports the process of developing software.
JTAG	Joint Test Action Group A standard, on-chip, debug interface supported by many chips, including most members of the Power Architecture family. The standard specifies how to control and monitor the pins of compliant devices on a printed circuit board. In addition, the standard defines commands for reading and setting the values of pins and internal registers of devices. Test equipment connected to a JTAG port can identify the devices on a board and can control and monitor the output of each device.
LTIB	Linux Target Image Builder A software tool that lets you generate customized embedded Linux images from LTIB-format Linux board support packages.

Table 1.2 Acronyms Used in this Manual (*continued*)

Acronym	Description
MDS	Modular Development System A Freescale system of boards that provides a platform for developing custom embedded software for a particular Freescale microcontroller. An MDS system is more configurable and extensible than is an ADS board.
STABS	Symbol Table Directives Debugging information format for ELF and other UNIX object file formats. Supports fewer debugger features than DWARF.
TAP	Test Access Port 4-wire serial interface for accessing complex integrated circuits.

Preparing the MPC8568E-MDS-PB Board

This chapter explains how to prepare the MPC8568E-MDS-PB processor board for each type of development you can perform with the MPC8568E kit.

NOTE See the *MPC8568E-MDS Hardware Getting Started Guide* for instructions that explain how to assemble the board. See the *MPC8568E-MDS User's Guide* for detailed information about the board's operation.

The sections of this chapter are:

- [“Setting Board Jumpers and Switches”](#)
- [“Preparing for Bare Board Development or for Linux® OS Customization”](#)
- [“Preparing for Linux® Application Development” on page 17](#)

Setting Board Jumpers and Switches

Before you can use the MPC8568E-MDS-PB processor board, you must set the board's switches and jumpers correctly. To make these settings, refer to the printed *MPC8568E-MDS Hardware Getting Started Guide* (included in the kit).

Preparing for Bare Board Development or for Linux® OS Customization

Follow these steps to prepare the MPC8568E-MDS-PB processor board such that:

- You can download binary files to the board and debug them using the CodeWarrior debugger.
 - You can write Linux images to the board's flash memory.
1. Ensure that the board is powered off. Because the power switch, SW5, is a toggle, check that the power is off by looking at the board to see that no LEDs are on.

Preparing the MPC8568E-MDS-PB Board

Preparing for Bare Board Development or for Linux® OS Customization

2. Make power connections.
 - a. Connect the supplied 5 Volt power supply to board power connector P8.
 - b. Connect the supplied power cable between the power supply and a surge-protected power strip.
 - c. Make sure the power strip is turned OFF.
 - d. Connect the power strip to line power.
 - e. Turn on the power supply.
3. Perform initial board power up.
 - a. Press the board power switch (SW5).

The processor board powers up. LD2 and LD7 display a constant green light. LD1 flashes orange once at the end of the reset sequence.
 - b. Press the board power switch again to turn off power.

The board powers down.
4. Connect the USB TAP debug probe.
 - a. Connect the USB TAP debug probe's JTAG/COP connector to header P5 of the processor board.

NOTE The red wire of the USB TAP probe's JTAG/COP connector goes to pin 1 of board header P5.

- b. Connect the supplied USB cable to the USB TAP probe.
- c. Connect the USB cable to a USB 2.0 port of your computer.

NOTE Do not install the driver for the USB TAP now. If the Found New Hardware Wizard appears, click **Cancel**.

5. Make board-to-computer serial connection.
 - a. Connect the RJ45 plug of the supplied joined adapter cable to board serial connector RS232-1,2.
 - b. Connect one end of the supplied RS-232 straight-through serial cable to the Port 1 DB9 connector of the joined adapter cable.
 - c. Connect the other end of the RS-232 serial cable to a serial port of your computer.

The MPC8568E-MDS-PB processor board is now ready to execute binaries under control of the CodeWarrior debugger. See [“Creating and Debugging a Bare Board Project” on page 43](#) for instructions.

The processor board is also ready to be flashed with a custom Linux image. See [“Using the MPC8568E-MDS-PB BSP” on page 55](#) for instructions.

Preparing for Linux® Application Development

To prepare the MPC8568E-MDS-PB processor board so you can download Linux applications to the board and then debug them using the CodeWarrior debugger, follow these steps:

1. Ensure that power to the board is off.
Because the power switch, SW5, is a toggle, verify that the power is off by looking at the board to see that no LEDs are on.
2. Make board-to-computer serial connection.
 - a. Connect the RJ45 plug of the supplied joined adapter cable to board serial connector RS232-1,2.
 - b. Connect one end of the supplied RS-232 straight-through serial cable to the Port 1 DB9 connector of the joined adapter cable.
 - c. Connect the other end of the RS-232 serial cable to a serial port of your computer.
3. Connect the supplied Ethernet cable between board connector GETH1 and a local area network (LAN) hub, switch, or port.
4. Make power connections.
 - a. Connect the supplied power supply to board power connector P8.
 - b. Connect the supplied power cable between the power supply and a surge-protected power strip.
 - c. Make sure the power strip is turned OFF.
 - d. Connect the power strip to line power.
5. On your computer, start a terminal emulator program.
6. Configure the terminal emulator as shown in [Table 2.1](#).

Table 2.1 Terminal Emulator Configuration Settings

bits per second	115200
data bits	8
parity	none
stop bits	1
hardware flow control	none
software flow control	none

Preparing the MPC8568E-MDS-PB Board

Preparing for Linux® Application Development

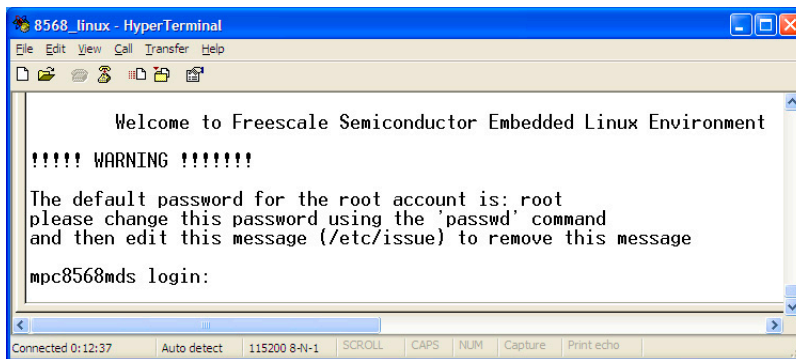
7. Press the power toggle switch (SW5) to turn on power to the board.

The processor board powers up. LD2 and LD7 display a constant green light. LD1 flashes orange once at the end of the reset sequence.

The emulator displays Linux boot status messages and then displays this shell prompt:
freescale login:

(See [Figure 2.1](#).)

Figure 2.1 Terminal Emulator Showing Linux® Shell Prompt



8. In the terminal emulator, type `root` at the login prompt and press **Enter**.

The system prompts you for a password.

9. In the terminal emulator, type `root` at the password prompt and press **Enter**.

The system logs you in as user `root` and displays this prompt:

`~ #`

10. At the prompt, enter this command:

`~ # ifconfig eth0 IPAddress netmask Mask`

(where *IPAddress* is an available, static IP address on your network, and *Mask* is the mask appropriate for your subnet).

The `ifconfig` utility assigns the specified IP address and netmask to Ethernet port 0.

NOTE If you do not have an unused, static IP address, obtain one from your network administrator.

11. Test the processor board's IP network connection.

- a. At any computer on the network, enter this command:

```
ping IPAddress
```

(where *IPAddress* is the IP address you assigned to Ethernet port J10 of the processor board).

The ping utility displays messages like these:

```
PING 10.82.191.3 (10.82.191.3): 56 data bytes
64 bytes from 10.82.191.3: icmp_seq=0 ttl=64 time=0.1 ms
64 bytes from 10.82.191.3: icmp_seq=1 ttl=64 time=0.0 ms
64 bytes from 10.82.191.3: icmp_seq=2 ttl=64 time=0.0 ms
64 bytes from 10.82.191.3: icmp_seq=3 ttl=64 time=0.0 ms
64 bytes from 10.82.191.3: icmp_seq=4 ttl=64 time=0.0 ms

--- 10.82.191.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

(where 10.82.191.3 is the IP address assigned to the board's Ethernet port).

- b. Press **Ctrl-C**.

The ping utility exits.

12. At the prompt, enter this command:

```
~ # apptrk :1000 &
```

The CodeWarrior Target-Resident Kernel (AppTRK) runs in the background on the processor board and listens on port 1000 for CodeWarrior debugger connections.

The terminal emulator redisplay the # prompt.

The MPC8568E-MDS-PB board is now ready to run Linux applications under control of the CodeWarrior debugger. See [“Creating and Debugging an MPC8568E-MDS-PB Project” on page 43](#) for instructions that explain how to create and debug such an application.

Installing CodeWarrior™ Development Studio

This chapter explains how to install the CodeWarrior Development Studio product required to perform the type of development you plan to perform. Next, the chapter explains how to register this product.

The sections are:

- [“Installing the Bare Board Development Tools”](#)
- [“Installing the Linux® Development Tools” on page 25](#)
- [“Registering and Licensing Your CodeWarrior™ Product” on page 28](#)

Installing the Bare Board Development Tools

This section explains how to install CodeWarrior Development Studio for Power Architecture Processors, Professional Edition. This product is for bare board development and runs on the Windows® operating system.

System Requirements

[Table 3.1](#) lists the minimum requirements a PC running Windows® must meet to run CodeWarrior Development Studio for Power Architecture Processors, Professional Edition.

Table 3.1 Professional Edition — System Requirements

Hardware	• PC with 1 GHz Intel® Pentium®-compatible processor (2 GHz recommended) • 512 MB RAM (1 GB RAM recommended) • CD-ROM drive
----------	---

Installing CodeWarrior™ Development Studio

Installing the Bare Board Development Tools

Table 3.1 Professional Edition — System Requirements (*continued*)

Operating system	• Microsoft® Windows 2000, Windows XP
Free disk space	• May require as much as 2 GB, depending on the choices selected. Additional space required for project files

Installation Procedure

NOTE If you are using a USB TAP debug probe, disconnect it from your PC before beginning the CodeWarrior software installation. If the USB TAP is connected during installation, the driver for this device may not be installed properly.

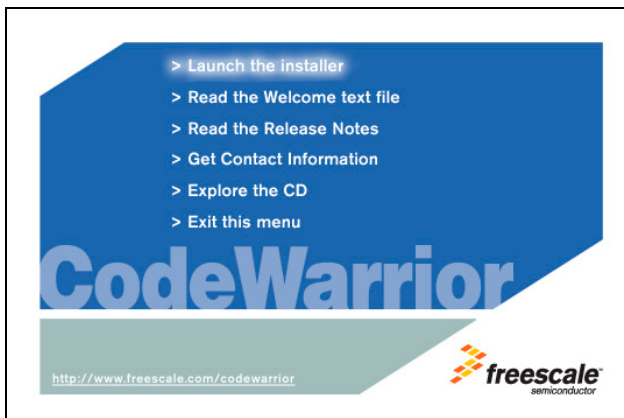
To install CodeWarrior Development Studio for Power Architecture Processors, Professional Edition, follow these steps:

1. Put the selected product installation CD, either CodeWarrior for Power Architecture v8.7 or v8.8, in the CD drive of your PC.

The CodeWarrior Setup menu appears ([Figure 3.1](#)).

NOTE If auto-install is disabled, display the setup menu by running `Launch.exe` manually. This program is in the `Windows` directory of the installation CD.

Figure 3.1 CodeWarrior™ Setup Menu

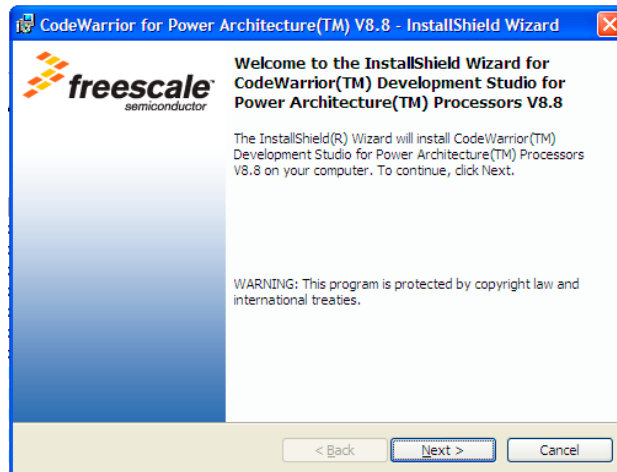


2. Click **Launch the installer**.

The CodeWarrior install wizard starts and displays its welcome page ([Figure 3.2](#)).

NOTE If you are installing CodeWarrior for Power Architecture Processors, v8.7 rather than v8.8, the wizard welcome page will have v8.7 rather than v8.8 in it.

Figure 3.2 CodeWarrior™ Install Wizard — Welcome Page



3. Follow the wizard's on-screen instructions to install the CodeWarrior software and the desired GCC toolchains.

NOTE As you progress through the wizard, make sure you choose the Professional Edition, not the Linux Application Edition.

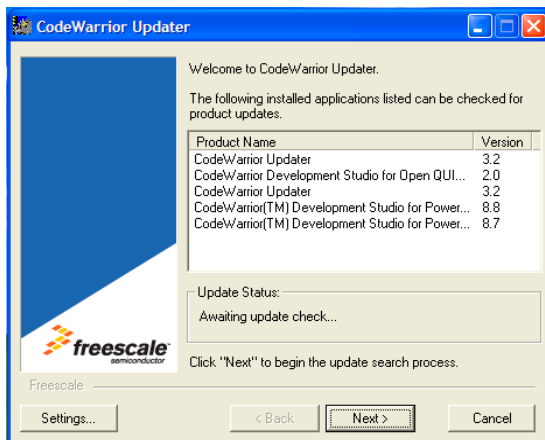
When installation is finished, the wizard displays the installation complete page.

4. In the installation complete page:
 - Check the Yes, check for program updates box.
 - Click **Finish**.The **CodeWarrior Updater** window ([Figure 3.3](#)) appears.

Installing CodeWarrior™ Development Studio

Installing the Bare Board Development Tools

Figure 3.3 CodeWarrior™ Updater Window



NOTE If the CodeWarrior Updater already has the correct configuration, skip to step 8.

5. Click **Settings**.

The **CodeWarrior Updater Settings** dialog box appears.

6. Use this dialog box to make settings the updater requires to connect to the Internet and to configure the behavior of the updater program itself.

7. Click **OK** to save your configuration choices or click **Cancel** to discard your choices.

The **CodeWarrior Updater Settings** dialog box closes.

8. In the **CodeWarrior Updater** window, click **Next**.

The CodeWarrior Updater checks for newer versions of the CodeWarrior products installed on your PC.

9. Follow the on-screen instructions to download CodeWarrior product updates.

10. When the CodeWarrior Updater displays the Update Check Complete! message, click **Finish**.

The updater program exits. CodeWarrior software installation is complete.

For instructions that explain how to register and license your CodeWarrior product, see [“Registering and Licensing Your CodeWarrior™ Product” on page 28](#).

Installing the Linux® Development Tools

This section explains how to install CodeWarrior Development Studio for Power Architecture Processors, Linux Platform/Application Edition. This product lets you perform embedded Linux application and kernel development and runs on the Linux® operating system.

System Requirements

[Table 3.1](#) lists the minimum requirements a PC running Linux must meet to run CodeWarrior Development Studio for Power Architecture Processors, Platform Edition.

Table 3.2 Linux™ Platform Edition — System Requirements

Hardware	• PC with 800 MHz Intel® Pentium®-compatible processor • 512 MB RAM • CD-ROM drive • USB 2.0 port (for use with USB TAP) • Ethernet port • Serial port
Operating system	• Red Hat™ Enterprise Linux (RHEL) 3.0 or 4.0
Free disk space	• May require as much as 1GB, depending on options selected. Additional space required for project files

Installation Procedure

NOTE If you are using a USB TAP probe, disconnect it from your Linux PC before beginning the CodeWarrior software installation. If the USB TAP is connected during installation, the driver for this device may not be installed properly.

To install CodeWarrior Development Studio for Power Architecture Processors, Linux Platform Edition, follow these steps:

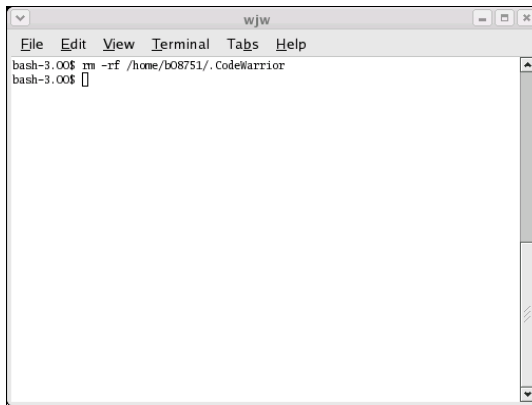
1. Put the installation CD for the embedded Linux tools in the CD drive of your Linux PC.
2. Mount the CD on the host filesystem, if necessary.
3. Open a terminal window.

A shell session ([Figure 3.4](#)) starts.

Installing CodeWarrior™ Development Studio

Installing the Linux® Development Tools

Figure 3.4 Linux® Terminal Window



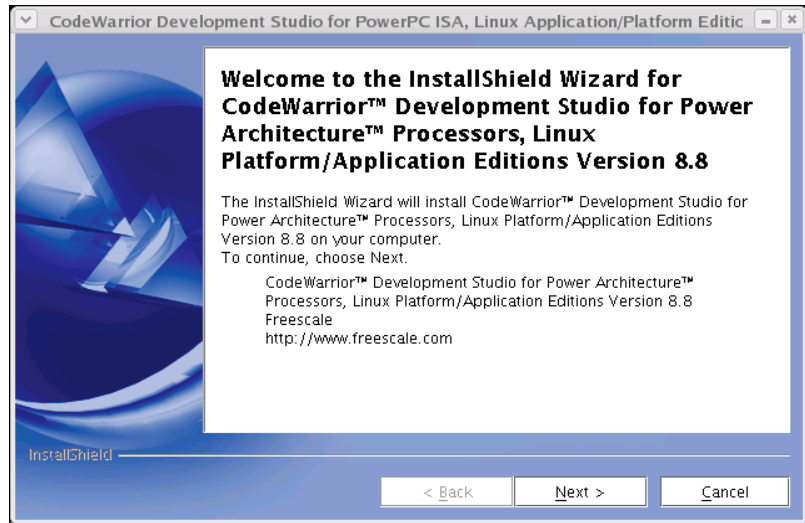
4. Enter this command:
`rm -rf userHome/.CodeWarrior`
(where *userHome* is a placeholder for the full path of your home directory).
5. So the installer can run in graphical mode, enter this command:
`xhost +`
6. Use the `su` command to obtain superuser privileges.

NOTE You must have superuser privileges to install the CodeWarrior software.

7. Change directory to the directory on which you mounted the CD media.
8. Enter this command:
`./setu linux.bin`
The CodeWarrior install wizard starts and displays its welcome page ([Figure 3.5](#)).

NOTE If your kit includes CodeWarrior for Power Architecture Processors, v8.7 rather than v8.8, the wizard welcome page will have v8.7 rather than v8.8 in it.

Figure 3.5 Install Wizard — Welcome Page



9. Follow the wizard's on-screen instructions until the installation directory page appears.
10. If the default path displayed on this page (`/usr/local/Freescale`) already includes the name CodeWarrior, follow these steps:
 - a. Click **Browse**.
The **Select a directory** dialog box appears.
 - b. Use this dialog box to select a different destination path.
 - c. Click **Open**.
The dialog box closes. The install wizard displays the chosen destination path.

NOTE Do not select a destination directory name that contains embedded spaces. If you do, the CodeWarrior software will not run.

11. Click **Next** repeatedly until the GCC toolchain page appears.
12. On this page, check the box for each GCC toolchain you want to install.
13. Click **Next**.
If you selected just one GCC toolchain, the install wizard displays the summary information page.
If you selected more than one GCC toolchain, the install wizard displays the select default toolchain page. In this case, follow these steps:

Installing CodeWarrior™ Development Studio

Registering and Licensing Your CodeWarrior™ Product

- a. Select the GCC toolchain to make the CodeWarrior default toolchain.
- b. Click **Next**.

The install wizard displays the summary information page.

14. Click **Next**.

The install wizard installs the CodeWarrior software, creates an uninstaller, and then displays the install selected GCC toolchains page.

15. Click **Next**.

The install wizard installs each GCC toolchain chosen previously and then displays the registration instructions page.

16. Read this page and click **Next**.

The install wizard displays its summary information page.

17. Click **Finish**.

The install wizard exits.

18. In the terminal window, type `exit` at the `#` prompt.

The system restores your normal privileges.

19. Close the terminal window.

CodeWarrior installation is complete.

For instructions that explain how to register and license your CodeWarrior product, see [“Registering and Licensing Your CodeWarrior™ Product”](#).

Registering and Licensing Your CodeWarrior™ Product

NOTE Free versions of the CodeWarrior tools include a permanent, but feature-limited license. You do not need to register these tools to obtain a license. You can obtain an evaluation license (full featured, but time-limited) from the CodeWarrior Licensing and Registration web page at the Freescale web site. To obtain a permanent license for your product, you need an entitlement ID (registration code), as explained in this section.

Installing CodeWarrior™ Development Studio

Registering and Licensing Your CodeWarrior™ Product

To register and license your CodeWarrior product, follow these steps:

1. Start a web browser.
2. Go to this web address:

<http://www.freescale.com/cwregister>

The **Licensing and Registration** page of the CodeWarrior licensing and registration system appears.

3. Click the Registration and Activation link.

The **Registration and Activation** page appears.

4. Follow the on-screen instructions on this page.

A license file (`license.dat`) is copied to your development machine.

5. If you installed the Windows-hosted tools, copy `license.dat` to the directory in which you installed these tools.

NOTE The default installation directory for the Windows-hosted CodeWarrior tools is: `C:\Program Files\Freescale\CodeWarrior PA V8.x`

6. If you installed the Linux-hosted tools, copy `license.dat` to the CodeWarrior IDE/CodeWarrior subdirectory of the directory in which you installed these tools.

NOTE The default installation directory for the Linux-hosted CodeWarrior tools is: `/usr/local/Freescale`.

Your CodeWarrior product is ready to use.

NOTE To register your product, you need an entitlement ID. If you do not have this ID and believe you are entitled to a permanent license, please go to <http://www.freescale.com/support>. On this page, open a service request, using Category “Technical Request” and Topic “License Issues.”

Installing and Using the Netcomm Device Drivers

This chapter explains how to install and use the Netcomm Device Drivers package.

The Netcomm Device Drivers package is a complete, fully modular set of system drivers peripheral drivers, and protocol drivers supporting the MPC8568E device and the board support package (BSP) for MDS boards.

The most important features of the Netcomm Device Drivers package are listed below.

- Complete set of device driver API routines that supports the device's functionality
- Simple, minimal configuration, so you can get started quickly
- Preset, default configuration values for easy set and run
- Detailed, low-level configuration allows for parametric changes to peripheral functionality
- Interrupt and polling modes for communication peripherals
- Statistics gathering capability
- Complete memory management for MPC8568E parameters, tables, and BDs
- Multiple type memory management supporting MURAM and external memories
- Object-oriented design and implementation in the C language
- Operating system agnostic
- Built-in hooks that ease integration with an operating system
- Portability
- External hardware independence

The system layer driver is integrated with the device driver available for the use of MDS boards and operating system hooks support. The system layer driver:

- Enables an optional initialization layer, which can be unique for each processor-board-operating system combination
- Creates an additional abstraction layer

Within the Netcomm Device Drivers package, there are *use cases*. Each use case is a simple, protocol-based application that helps you start designing your application.

Installing and Using the Netcomm Device Drivers

Installing the Netcomm Device Driver Package

Each use case is based on at least one protocol. Each use case:

- Configures the device
- Configures the MDS board
- Generates and transmits data on a peripheral (for example, a UCC or an MCC)
- Receives data from the peripheral
- Compares the transmitted and received data
- Presents the results of the use case

NOTE You must use CodeWarrior for Power Architecture Processors, **v8.7** with the Netcomm Device Drivers, v2.0 package. Using CodeWarrior for Power Architecture v8.8 with the Netcomm Device Drivers, v2.0 package is not supported.

NOTE The QUICC Engine™ utility (QEU) includes most of the same use cases as the Netcomm Device Drivers package. Using the QEU, you can view all parameters that can be configured in a device, microcode, and driver using a specific protocol. The QEU is included in the device drivers package.

The sections of this chapter are:

- [“Installing the Netcomm Device Driver Package”](#)
- [“Using the Netcomm Device Drivers” on page 34](#)

Installing the Netcomm Device Driver Package

The Netcomm Device Drivers package includes:

- The QUICC Engine utility
- The Netcomm device drivers

When you install this package, both these components are installed, so this installation procedure applies to the QUICC Engine Utility as well as to the Netcomm device drivers.

Installing and Using the Netcomm Device Drivers

Installing the Netcomm Device Driver Package

To install the Netcomm Device Drivers package, follow these steps:

1. Put the installation Development Kit Disk in the CD drive of your Windows PC.

Two subfolders are displayed:

- Proprietary Netcomm Device Driver — contains a full, proprietary bare board BSP
- Proprietary and GPL Netcomm Device Driver — includes everything listed above plus GPL USB stack

2. Choose the Proprietary Netcomm Device Driver folder.

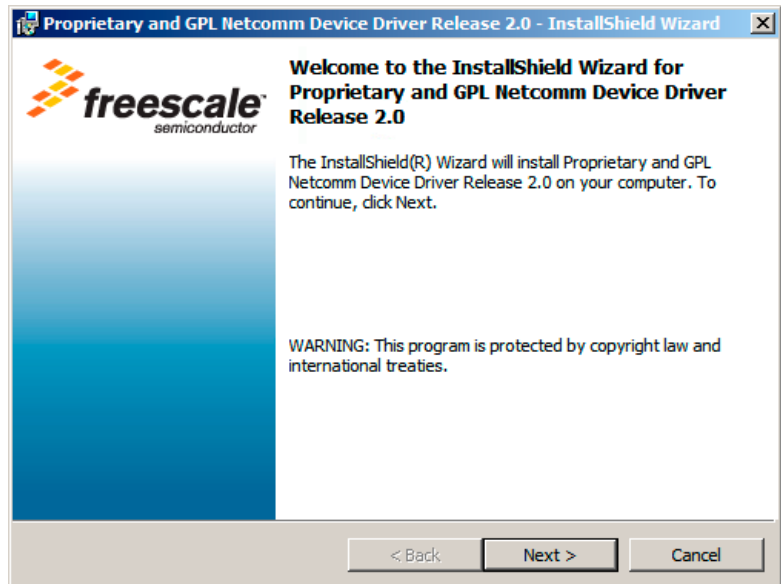
A list of files appears.

NOTE The following procedure explains how to install and use the proprietary device drivers; however, you can use the same procedure to install both the proprietary and the GPL package.

3. Double-click on the exe file you want to download.

The InstallShield wizard initializes and displays its welcome page ([Figure 4.1](#)).

Figure 4.1 InstallShield Wizard Welcome Page



Installing and Using the Netcomm Device Drivers

Using the Netcomm Device Drivers

4. Follow the wizard's on-screen instructions to install the Netcomm Device Drivers package.

During installation, you can specify an alternate installation directory and must accept the displayed license agreements.

5. When the InstallShield wizard displays its installation complete page, click **Finish**.

The wizard exits. Installation of the Netcomm Device Drivers package is complete.

Using the Netcomm Device Drivers

To demonstrate what you can do with the Netcomm Device Drivers package, this section walks you through the Ethernet use case, one of the use cases included in this package.

NOTE You must use CodeWarrior for Power Architecture Processors, **v8.7** with the Netcomm Device Drivers, v2.0 package.
Using CodeWarrior for Power Architecture v8.8 with the Netcomm Device Drivers, v2.0 package is not supported.

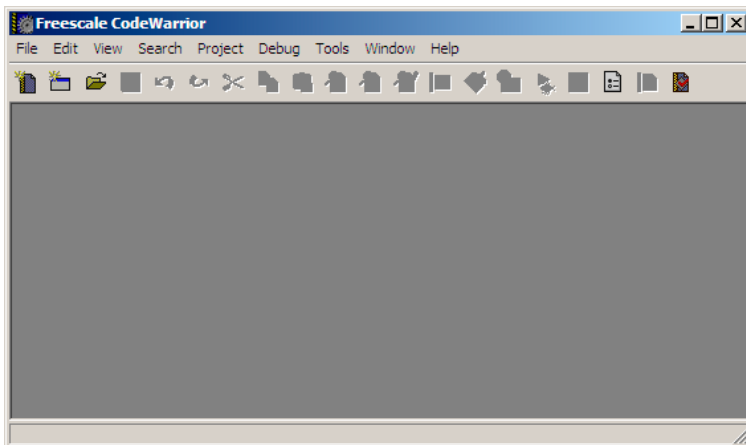
To inspect the Ethernet use case, follow these steps:

1. Start the CodeWarrior IDE.

To do this, select **Start > Programs > Freescale CodeWarrior > CW for Power Architecture V8.7 > CodeWarrior IDE** from the Windows taskbar.

The CodeWarrior IDE starts and displays its main window ([Figure 4.2](#)).

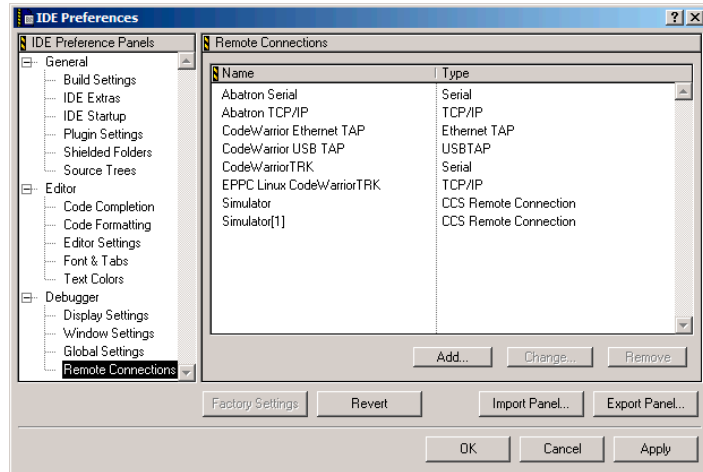
Figure 4.2 CodeWarrior™ IDE — Main Window



2. Create a USB TAP remote connection for the MPC8568E-MDS-PB processor board.
 - a. From the IDE menu bar, select **Edit > Preferences**.

The **IDE Preferences** window ([Figure 4.3](#)) appears.

Figure 4.3 IDE Preferences Window Showing Remote Connections Preference Panel



- b. From pane on the left side of the **IDE Preferences** window, select Remote Connections.

The **Remote Connections** preference panel appears on the right side of the **IDE Preferences** window.

- c. In the **Remote Connections** panel, click **Add**.

The **New Connection** dialog box ([Figure 4.4](#)) appears.

Installing and Using the Netcomm Device Drivers

Using the Netcomm Device Drivers

Figure 4.4 New Connection Dialog Box

d. Fill in the items of the **New Connection** dialog box as shown in [Table 4.1](#).

Table 4.1 USB TAP 8568 Remote Connection — Settings

Dialog Box Item	Setting
Name text box	USB TAP 8568
Debugger dropdown menu	CCS EPPC Protocol Plugin
Connection Type dropdown menu	USB TAP
Use default Serial Number checkbox	Checked
CCS timeout text box	10
Interface Clock Frequency dropdown menu	4.03 MHz
Mem Read Delay text box	0

Table 4.1 USB TAP 8568 Remote Connection — Settings

Dialog Box Item	Setting
Mem Write Delay text box	0
TAP Memory Buffer (hex) checkbox	Unchecked
Reset Target on Launch checkbox	Checked
Do not use fast download checkbox	Unchecked
Force shell download checkbox	Unchecked
Enable Logging checkbox	Unchecked
Asynchronous Multi-Core Control checkbox	Checked
Use JTAG Configuration File checkbox	Checked
Use JTAG Configuration File text box	Enter this path and filename: <i>installDir</i>\NetCommSw\ build\bare_8568_cw_build\8568_jtag.txt. where <i>installDir</i> is the directory in which you installed the NetComm Device Drivers package. Alternatively, click Choose to display a dialog box with which to select this file.

- e. Click **OK**.

The **New Connection** dialog box closes.

The USB TAP 8568 remote connection appears in the **Remote Connections** preference panel.

- f. In the **IDE Preferences** window, click **OK**.

The **IDE Preferences** window closes.

Installing and Using the Netcomm Device Drivers

Using the Netcomm Device Drivers

3. Open the Netcomm Device Drivers Ethernet use case project.
 - a. Select **File > Open**.

The **Open** dialog box appears.
 - b. Use this dialog box to navigate to the Netcomm Device Driver project file named `GethUseCases.mcp`.

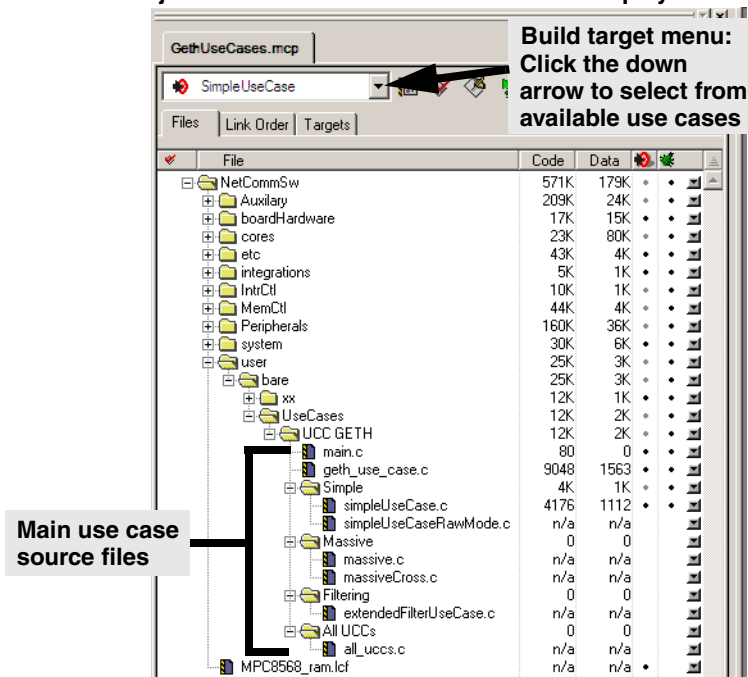
The project in this directory:

```
installDir\NetCommSW\build\bare_8568_cw_build\user\bare\UseCases\UCC\GETH\GethUseCases.mcp
```

where *installDir* is the directory in which you installed the Netcomm Device Drivers package.
 - c. Click **Open**.

The CodeWarrior IDE loads the GETHernet use case project. A project window (Figure 4.5) appears, listing the files of the use case.
4. From the project window's build target dropdown menu, select the `SimpleUseCase` build target.

Figure 4.5 CodeWarrior™ Project Window — GETH Use Case Files Displayed



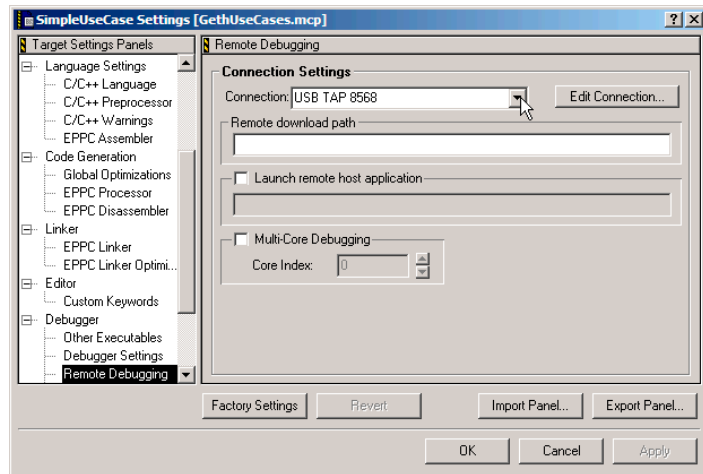
NOTE At this point, you can use the project window's build target dropdown menu to select other use cases available for the Geth protocol.

You may want to try those use cases that match your needs. However, we recommend that you do this *after* you complete this section.

5. Modify the build target's settings, so it uses the USB TAP 8568 remote connection.
 - a. Press **Alt-F7**.

The **Target Settings** window (Figure 4.6) appears.

Figure 4.6 Target Settings Window Showing Remote Debugging Settings Panel



- - b. From pane on the left side of the **Target Settings** window, select Remote Debugging.

The **Remote Debugging** target settings panel appears on the right side of the **Target Settings** window.

- -
 - c. From the Connection dropdown menu of the **Remote Debugging** target settings panel, select USB TAP 8568.
 - d. Click **Apply**.

The IDE assigns the USB TAP 8568 remote connection to the SimpleUseCase build target.

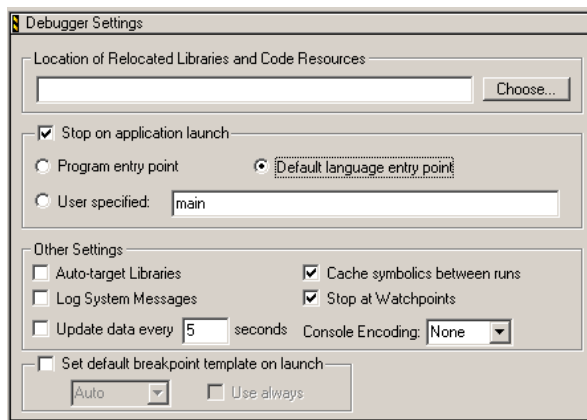
Installing and Using the Netcomm Device Drivers

Using the Netcomm Device Drivers

6. Modify the build target's settings, so the executable stops at main() on program launch.
 - a. From pane on the left side of the **Target Settings** window, select **Debugger Settings**.

The **Debugger Settings** target settings panel ([Figure 4.7](#)) appears on the right side of the **Target Settings** window.

Figure 4.7 Debugger Settings Target Settings Panel



- b. In this target settings panel, check the **Stop on application launch** checkbox.

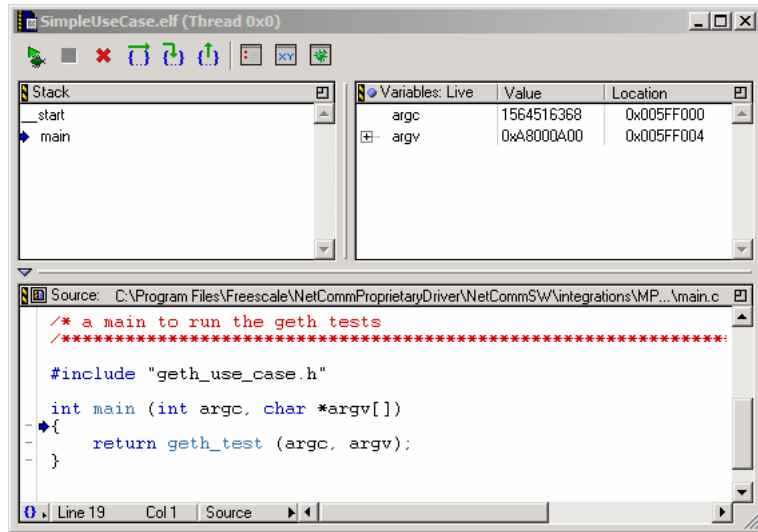
The items in the **Stop on application launch** group activate.
 - c. Select the **Default language entry point** option.
 - d. Click **OK**.

The IDE assigns the new settings to the SimpleUseCase build target and closes the **Target Settings** window.
7. Select **Project > Make**.

The IDE compiles the project's source code files and generates an executable file.
8. Select **Project > Debug**.

The IDE opens a console window, downloads the executable file to the MPC8568E-MDS-PB board, halts execution at the first statement in `main()`, and displays the debugger window ([Figure 4.8](#)).

Figure 4.8 CodeWarrior™ Debugger Window



9. Connect a Gethernet loopback cable (provided with the processor board) to the processor board port labeled GETH1.
10. Connect a second Gethernet loopback cable (provided with the processor board) to the processor board port labeled GETH2.
11. In the debugger window, click the run  button.

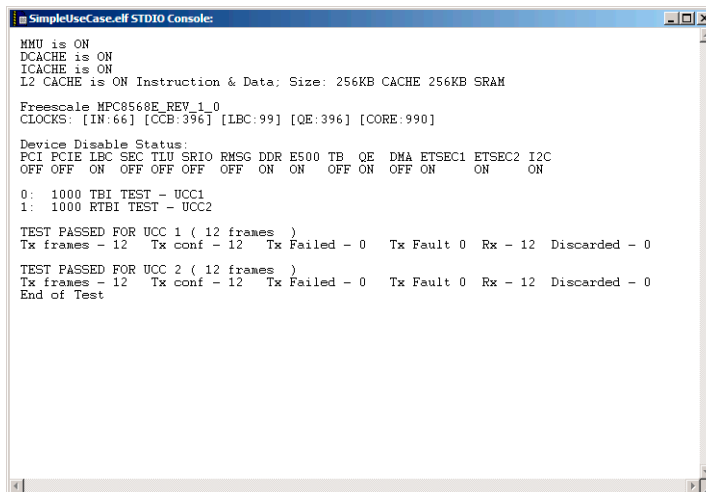
The program resumes execution, transmits/receives 200 Ethernet frames, writes status messages to the console window, and terminates.

[Figure 4.9](#) shows the console window after program termination.

Installing and Using the Netcomm Device Drivers

Using the Netcomm Device Drivers

Figure 4.9 Console Window after Program Termination



```
SimpleUseCase.elf STDIO Console:
MMU is ON
DCACHE is ON
ICACHE is ON
L2 CACHE is ON Instruction & Data; Size: 256KB CACHE 256KB SRAM
Freescale MPC8568E_REV_1.0
CLOCKS: [IN:66] [CCB:396] [LBC:99] [QE:396] [CORE:990]

Device Disable Status:
PCI PCIE LBC SEC TLU SRIO RMSG DDR ES00 TB QE DMA ETSEC1 ETSEC2 I2C
OFF OFF ON OFF OFF OFF OFF ON ON OFF ON OFF ON ON ON

0: 1000 TBI TEST - UCC1
1: 1000 RTBI TEST - UCC2

TEST PASSED FOR UCC 1 ( 12 frames )
Tx frames - 12 Tx conf - 12 Tx Failed - 0 Tx Fault 0 Rx - 12 Discarded - 0

TEST PASSED FOR UCC 2 ( 12 frames )
Tx frames - 12 Tx conf - 12 Tx Failed - 0 Tx Fault 0 Rx - 12 Discarded - 0
End of Test
```

12. In the debugger window, click the kill  button.

NOTE You may want to review the source code and modify some parameters or configuration settings. To do this, double-click the file `simpleUseCase.c` in the project window. The IDE opens this file in an editor window, where you can modify the file.

However, we recommend that you do this *after* completing this section's steps.

At the point, the debug session ends and the debugger windows close.

You have seen how to use the device driver Geth simple use case to configure the MPC8568E processor, generate packets, and compare the received and transmitted packets.

In addition, you have seen where the main use case source code files reside in a CodeWarrior MPC8568E project, built this project, and downloaded and run the resulting executable file on the MPC8568E-MDS-PB board.

The output generated by this program shows that the device successfully transmitted and received Ethernet frames using the board's two RGMII Ethernet interfaces.

Using the CodeWarrior™ QUICC Engine™ Utility

This chapter explains how to install and use the CodeWarrior QUICC Engine Utility.

The QUICC Engine Utility (QEU) is a GUI-based tool that speeds and simplifies the initialization and configuration of the drivers, device, microcode, and communications protocols managed by the MPC8568E device. The QEU provides an easy-to-use environment for handling common QUICC Engine initialization tasks.

In addition, the QEU lets you configure use cases included in the Netcomm Device Driver package. These use cases let you adapt working code to suit your needs. Further, the use cases demonstrate how to use the device driver API.

The sections of this chapter are:

- [“Installing the QUICC Engine™ Utility”](#)
- [“Using the QUICC Engine™ Utility” on page 45](#)

NOTE You must use CodeWarrior for Power Architecture Processors, **v8.7** with the QUICC Engine Utility, v2.0.
Using CodeWarrior for Power Architecture v8.8 with the QUICC Engine Utility, v2.0 is not supported.

Installing the QUICC Engine™ Utility

System Requirements

[Table 5.1](#) lists the minimum requirements a PC running Windows® must meet to run the CodeWarrior QUICC Engine utility.

Using the CodeWarrior™ QUICC Engine™ Utility

Installing the QUICC Engine™ Utility

Table 5.1 CodeWarrior™ QUICC Engine™ Utility — System Requirements

Hardware	• PC with 1.8 GHz Intel® Pentium III®-compatible processor • 128 MB RAM • CD-ROM drive • USB 2.0 port
Operating system	• Microsoft® Windows 2000/XP SP2
Free disk space	• 500 MB for the QUICC Engine utility - Additional space required for project files

Installation Procedure

NOTE The installer for the Netcomm Device Driver package installs:

- The QUICC Engine utility
- The Netcomm device drivers

As a result, if you have already installed the Netcomm Device Drivers package, you can skip the procedure below.

To install the CodeWarrior QUICC Engine Utility, follow these steps:

1. Put the installation Development Kit Disk in the CD drive of your Windows PC.
2. Choose the Netcomm Device Driver folder.

Two subfolders are displayed:

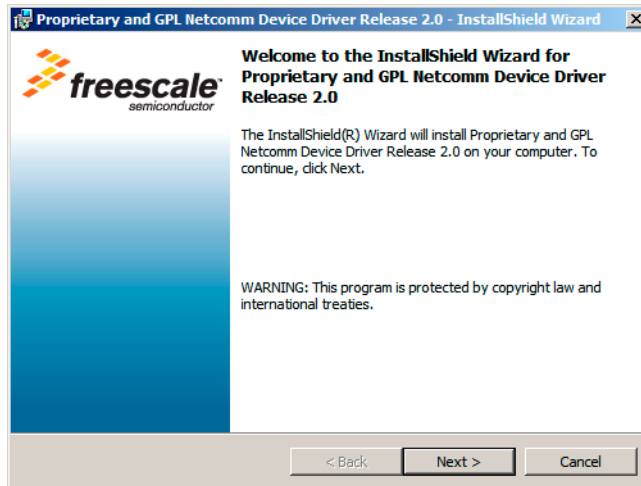
 - Proprietary Netcomm Device Driver — contains full, proprietary bare board BSP
 - Proprietary and GPL Netcomm Device Driver — contains the above release plus GPL PCI and USB stacks

NOTE The following steps explain how to install the proprietary version; however, you can use the same procedure to install the proprietary and GPL version.

3. Double click the exe file you want to download.

The Install Wizard initializes itself (this may take a while) and then displays its welcome page ([Figure 5.1](#)).

Figure 5.1 Installation Wizard — Welcome Page



4. Follow the wizard's on-screen instructions to install the QUICC Engine Utility and the Netcomm Device Drivers software.
During installation, you may want to specify an alternate installation directory and must accept the displayed license agreements.
5. When the Install Wizard displays its installation complete page, click **Finish**.
The Install Wizard exits. Installation of the QUICC Engine utility and of the Netcomm Device Driver package is complete.

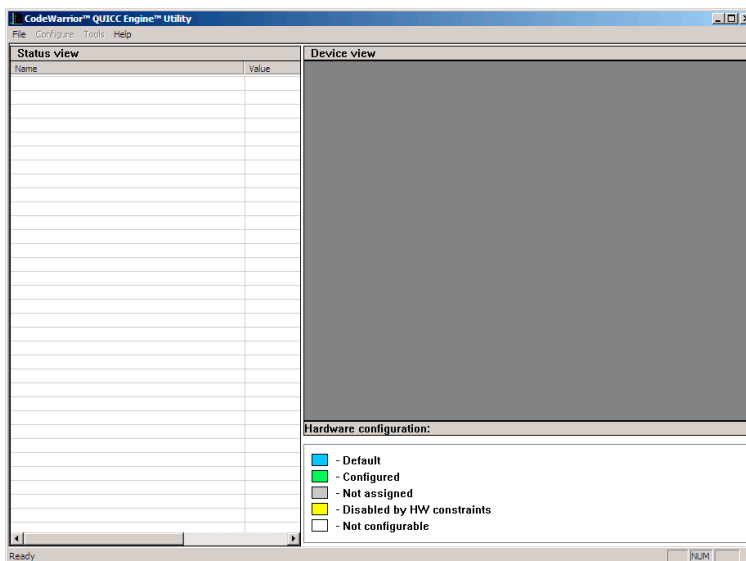
Using the QUICC Engine™ Utility

To demonstrate what the QUICC Engine Utility can do, this section walks you through the Ethernet use case, one of the use cases included in the QEU product.

To enact the Ethernet use case, follow these steps:

1. Start the QUICC Engine utility.
To do this, select **Start > Programs > Freescale > Proprietary Netcomm Device Driver Release 2.0 > CodeWarrior QUICC Engine Utility** from the Windows taskbar.
The QEU starts and displays its main window ([Figure 5.2](#)).

Figure 5.2 QUICC Engine™ Utility — Main Window



2. Open the QEU Ethernet use case project.

- a. Select **File > Open**.

The **Open** dialog box appears.

- b. Use this dialog box to navigate to the QEU project file named `eth_simple_UCC_1_2.cwp`.

This project file is in this directory:

`installDir\CWQEU\UseCases_8568\QEU_UseCase\Projects`

where `installDir` is the directory in which the QEU is installed.

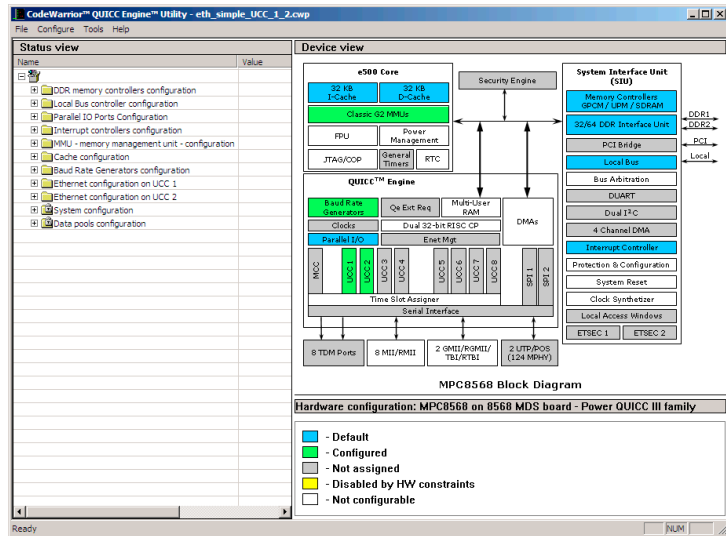
- c. Click **Open**.

The QEU loads the Ethernet use case project. Once the load completes, the QEU's main window ([Figure 5.3](#)) shows the Power Architecture core blocks, QUICC Engine blocks, and System Interface Unit blocks the project uses.

The **Device view** pane of the main window shows a block diagram of the MPC8568E processor. The green blocks are chip elements that the Ethernet use case project both manipulates and has configured. The blue blocks are chip elements that the use case manipulates and for which the default configuration is sufficient. The gray blocks are chip elements that the use case does not use, and the white blocks are chip elements for which the QEU cannot generate code.

When expanded, the nodes of the tree displayed in the **Status view** pane show the current configuration of each green and blue block.

Figure 5.3 QEU Main Window with the Ethernet Use Case Project Open



3. In the **Device view** pane, click on the green block labeled UCC1 (Unified Communications Controller 1).

The QEU displays a tabbed window titled **Ethernet configuration on UCC1**.

The tabs of this window show the UCC1 block's current configuration block. These tabs show that UCC1 is configured:

- As an Ethernet device with a MAC address of 10-34-56-78-9A-BC.
- To transmit and receive at 1 Gbps over a GMII (Gigabyte Media Independent Interface) interface.

4. Click **Cancel**.

The **Ethernet configuration on UCC1** window closes.

5. In the **Device view** pane, click on the green block labeled UCC2 (Unified Communications Controller 2).

The QEU displays a tabbed window titled **Ethernet configuration on UCC2**.

The tabs of this window show the UCC2 block's current configuration. These tabs show that UCC2 is configured the same as UCC1, except that the MAC address assigned to UCC2 is 18-34-56-78-9A-BC.

Using the CodeWarrior™ QUICC Engine™ Utility

Using the QUICC Engine™ Utility

6. Click **Cancel**.

The **Ethernet configuration on UCC2** window closes.

Based on previous steps, you know that the QEU Ethernet use case project is configured to generate code that sets up the MPC8568E chip's QUICC Engine block. It is set up to perform 1 Gbps Ethernet communications over two GMII Ethernet devices.

NOTE At this point, you could use the QEU to modify the project's QUICC Engine configuration to fit the requirements of your application. See the *CodeWarrior™ QUICC Engine™ Utility User's Guide* for instructions that explain how to accomplish this task. For the purposes of this demonstration, however, the project does not need to be modified.

7. Optionally, backup the existing Ethernet use case source code files.
 - a. In Windows Explorer, navigate to this directory:
`installDir\CWQEU\UseCases_8568\QEU_UseCase\Projects`
 - b. Create a subdirectory of the `Projects` directory, and name it `backup`.
 - c. Copy each header (*.h) and implementation (*.c) file from the
`installDir\CWQEU\UseCases_8568\QEU_UseCase\Init\` directory
to the `backup\` directory.
8. Use the QEU to generate source code files.
 - a. Select **Tools > Generate Code**.
The **Save as** dialog box appears.
 - b. In the **Save as** dialog box, navigate to:
`installDir\CWQEU\UseCases_8568\QEU_UseCase\Init\`
 - c. In the File name text box, type `qeu_init`.
 - d. Click **Save**.

The QEU generates six source code files. The contents of these files are determined by the configuration of each of the chip's blocks.

The names of four of the generated files begin with the prefix you entered in the **Save as** dialog box. The fifth file starts with the prefix `cwque_`. The sixth file is named `system_config.h`.

9. Exit the QUICC Engine utility.
10. Set up the MPC8568E-MDS-PB board for debugging by following the instructions in the [Preparing for Bare Board Development or for Linux® OS Customization](#) topic.
11. Start the CodeWarrior IDE.

To do this, select **Start > Programs > Freescale CodeWarrior > CW for Power Architecture V8.7 > CodeWarrior IDE** from the taskbar.

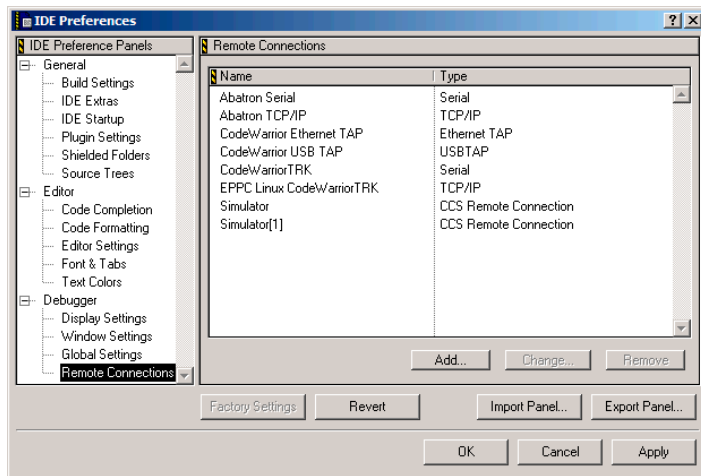
NOTE In this step, you must start CodeWarrior Professional Edition, not CodeWarrior Linux Platform Edition.

The CodeWarrior IDE starts and displays its main window.

12. Create a USB TAP remote connection for the MPC8568E-MDS-PB processor board.
 - a. From the IDE menu bar, select **Edit > Preferences**.

The **IDE Preferences** window appears ([Figure 5.4](#)).

Figure 5.4 IDE Preferences Window Showing Remote Connections Preference Panel



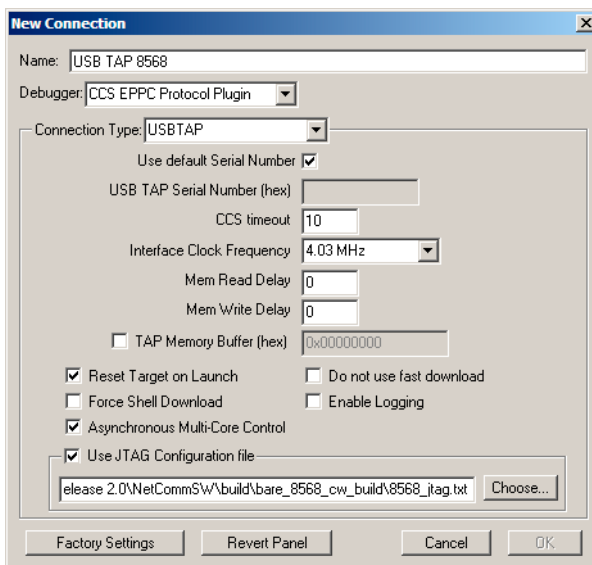
- b. From pane on the left side of the **IDE Preferences** window, select Remote Connections.

The **Remote Connections** preference panel appears on the right side of the **IDE Preferences** window.

- c. In the **Remote Connections** panel, click **Create**.

The **New Connection** dialog box ([Figure 5.5](#)) appears.

Figure 5.5 New Connection Dialog Box



- d. Fill in the items of the **New Connection** dialog box as shown in [Table 5.2](#).

Table 5.2 USB TAP 8568 Remote Connection — Settings

Dialog Box Item	Setting
Name text box	USB TAP 8568
Debugger dropdown menu	CCS EPPC Protocol Plugin
Connection Type dropdown menu	USB TAP
Use default Serial Number checkbox	Checked
CCS timeout text box	10
Interface Clock Frequency dropdown menu	4.03 MHz
Mem Read Delay text box	0

Table 5.2 USB TAP 8568 Remote Connection — Settings

Dialog Box Item	Setting
Mem Write Delay text box	0
TAP Memory Buffer (hex) checkbox	Unchecked
Reset Target on Launch checkbox	Checked
Do not use fast download checkbox	Unchecked
Force shell download checkbox	Unchecked
Enable Logging checkbox	Unchecked
Asynchronous Multi-Core Control checkbox	Checked
Use JTAG Configuration File checkbox	Checked
Use JTAG Configuration File text box	Enter this path and filename: <i>installDir</i> \NetCommSw\ build\bare_8568_cw_build\8568_jtag.txt. where <i>installDir</i> is the directory in which you installed the NetComm Device Drivers package. Alternatively, click Choose to display a dialog box with which to select this file.

- e. Click **OK**.

The **New Connection** dialog box closes.

The USB TAP 8568 remote connection appears in the **Remote Connections** preference panel.

- f. In the **IDE Preferences** window, click **OK**.

The **IDE Preferences** window closes.

Using the CodeWarrior™ QUICC Engine™ Utility

Using the QUICC Engine™ Utility

13. Open the CodeWarrior Ethernet use case project.

- a. Select **File > Open**.

The **Open** dialog box appears.

- b. Use this dialog box to navigate to the CodeWarrior project file named `QEU_UseCase.mcp`.

This project file is in this directory:

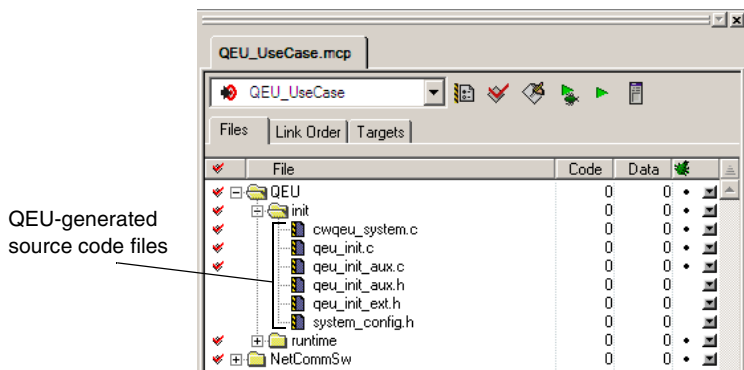
`installDir\CWQEU\UseCases_8568\QEU_UseCase`

where `installDir` is the directory in which the QEU is installed.

- c. Click **Open**.

The CodeWarrior IDE opens the CodeWarrior Ethernet use case project and displays it in a project window ([Figure 5.6](#)). This project already contains the files that the QEU just generated.

Figure 5.6 CodeWarrior™ Project Window Showing the Ethernet Use Case Project

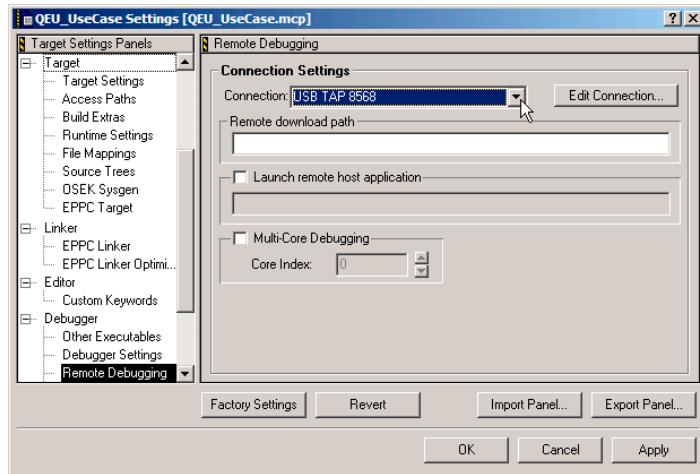


14. Modify the project's settings, so it uses the USB TAP 8568 remote connection.

- a. Press **Alt-F7**.

The **Target Settings** window ([Figure 5.7](#)) appears.

Figure 5.7 Target Settings Window Showing Remote Debugging Settings Panel



- b. From pane on the left side of the **Target Settings** window, select Remote Debugging.

The **Remote Debugging** target settings panel appears on the right side of the **Target Settings** window.

- c. From the Connection dropdown menu of the **Remote Debugging** target settings panel, select USB TAP 8568.
- d. From pane on the left side of the **Target Settings** window, select Debugger Settings.

The **Debugger Settings** target settings panel appears on the right side of the **Target Settings** window.

- e. Check the Stop on application launch checkbox.

This setting halts the program execution at the first statement in `main()`.

- f. Click **OK**.

The USB TAP 8568 remote connection is assigned to the project. Also, the project is configured to stop execution at first statement in `main()`. The **Target Settings** window closes.

15. Select **Project > Make**.

The IDE compiles the project's source code files and generates a binary file (in executable and linkable (ELF) format).

Using the CodeWarrior™ QUICC Engine™ Utility

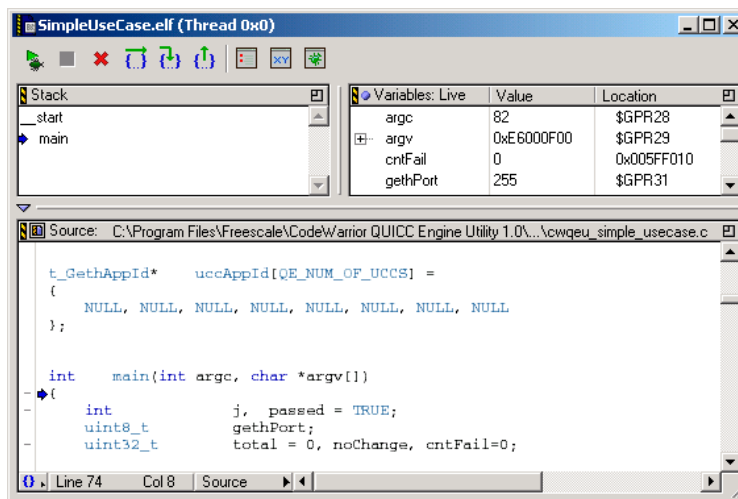
Using the QUICC Engine™ Utility

16. Select **Project > Debug**.

The IDE opens a console window, downloads the binary to the MPC8568E-MDS-PB board, halts execution at the first statement in `main()`, and displays the debugger window.

[Figure 5.8](#) shows the debugger window.

Figure 5.8 Debugger Window



17. In the debugger window, click the run button.

The program resumes execution, uses the QUICC Engine's UCC1 and UCC2 blocks to transmit/receive 200 Ethernet frames, writes status messages to the console window, and terminates.

18. In the debugger window, click the kill button.

The debug session ends and the debugger window closes.

That's it. You have seen how to use the QUICC Engine Utility to configure the MPC8568E processor's QUICC Engine block and to generate source code that sets up this block as configured.

In addition, you have seen where the generated source code files fit into a CodeWarrior MPC8568E project, built this project, and downloaded and run the resulting binary on the MPC8568E board. Finally, the output generated by this program shows that the chip's QUICC Engine block successfully transmitted and received Ethernet frames using the board's two GMII Ethernet interfaces.

Using the MPC8568E-MDS-PB BSP

This chapter explains how to install the Linux® Target Image Builder (LTIB) tools framework and the MPC8568E-MDS-PB Linux board support package (BSP).

Use LTIB in conjunction with the board support package to create custom Linux images for your MPC8568E-MDS-PB processor board. You can write these images to the flash memory of the processor board using the CodeWarrior flash programmer and the USB TAP probe.

The sections of the chapter are:

- [“System Requirements”](#)
- [“Installing LTIB and the MPC8568E-MDS-PB Board Support Package” on page 56](#)

System Requirements

[Table 6.1](#) lists the minimum requirements a PC running Linux must meet to run the LTIB tools framework.

Table 6.1 Linux® Target Image Builder Tools Framework — System Requirements

Hardware	• PC with 800 MHz Intel® Pentium®-compatible processor • 256 MB RAM • CD-ROM drive • USB 2.0 port • Ethernet card and port • Serial port
Operating system	• Debian 3.1r0 (stable, unstable), Red Hat® Enterprise 3.0 or 4.0, SuSE® 8.2, 9.1, 9.2, or 10.0, Fedora Core 1, 2, 3, or 4
Free disk space	• 1.5 GB for LTIB installation • An additional 2 GB minimum for project storage.

Using the MPC8568E-MDS-PB BSP

Installing LTIB and the MPC8568E-MDS-PB Board Support Package

Table 6.1 Linux® Target Image Builder Tools Framework — System Requirements

Additional	• NFS server • TFTP server • rsync • perl
------------	---

Installing LTIB and the MPC8568E-MDS-PB Board Support Package

To install the LTIB tools framework and the MPC8568E-MDS-PB board support package, follow these steps:

1. Login to your Linux PC *without* root privileges.
2. Open a terminal window.
3. Put the BSP installation CD in the CD-ROM drive of your PC running Linux.

The system automatically mounts the CD.

If the system does not mount the CD automatically, enter this command:

```
mount /media/cdrom -o exec
```

NOTE Your Linux distribution may require that you mount a CD on a directory other than `/media/cdrom`.

For example, some distributions require that you use `/mnt/cdrom`. If you cannot mount the CD using the command above, refer to your distribution's documentation for instructions.

4. Enter this command:
`/media/cdrom/install`

The LTIB install script starts.

5. Follow the on-screen instructions to install LTIB and the MPC8568E-MDS-PB board support package.
6. To complete installation:
 - a. Change directories to `installDir/ltib`
where `installDir` is the path in which you installed the LTIB tools framework.
 - b. Enter this command:
`./ltib`
Installation of LTIB and the MPC8568E-MDS-PB board support package completes.

7. Enter this command:
`umount /media/cdrom`

The system unmounts the BSP installation CD.

The LTIB tools framework and the MPC8568E-MDS-PB board support package are installed. You can now use LTIB to create customized Linux images from this BSP. For instructions, see `START_HERE.html` in the root directory of the BSP installation CD.

Using the MPC8568E-MDS-PB BSP

Installing LTIB and the MPC8568E-MDS-PB Board Support Package

Index

A

acronyms, table of 12
ADS, defined 12
application development system, defined 12

C

CodeWarrior Development Studio
 installing
 bare board tools 21
 Linux tools 25
 registering 28
common on-chip processor 12
COP, defined 12

D

debug with arbitrary record format, defined 12
development kit overview 5
DWARF, defined 12

E

ELF, defined 12
executable and linkable format, defined 12

H

how to
 install LTIB 56
 install MPC8568E-MDS-PB BSP 56
 install the bare board tools 21
 install the Linux tools 25
 install the QUICC Engine utility 43
 prepare the MPC8568E-MDS-PB
 for bare board development 15
 for Linux development 17
 for Linux OS customization 15
 register CodeWarrior Development
 Studio 28
 use the QUICC Engine Utility 45

I

IDE, defined 12
installing the bare board tools 21

installing the Linux tools 25
installing the QUICC Engine Utility 43
integrated development environment, defined 12

J

joint test access group, defined 12
JTAG, defined 12
jumpers 15
jumpers, setting board 15

K

kit components 5

L

Linux target image builder, defined 12
Linux target image builder, system
 requirements 55
LTIB
 defined 12
 installing 56

M

MDS, defined 13
modular development system, defined 13
MPC8568E-MDS-PB BSP, installing 56

O

overview, development kit 5

P

preparing the MPC8568E-MDS-PB
 for bare board development 15
 for Linux development 17
 for Linux OS customization 15

Q

QUICC Engine Utility
 defined 43
 installing 43
 using 45

R

registering CodeWarrior Development Studio 28

requirements, system

- bare board development tools 21

- Linux development tools 25

- QUICC Engine Utility 43

S

setting board jumpers and switches 15

STABS, defined 13

switches, setting board 15

symbol table directives, defined 13

system requirements

- bare board development tools 21

- Linux development tools 25

- Linux target image builder (LTIB) 55

- QUICC Engine Utility 43

T

table of acronyms 12

TAP, defined 13

test access port, defined 13

U

using the QUICC Engine Utility 45