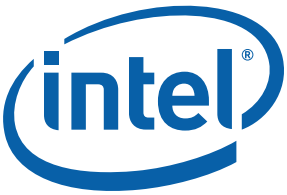


Intel® EP80579 Software Drivers for Embedded Applications on FreeBSD*

Getting Started Guide

September 2009

Order Number: 320152-005US

Contents

Figures

Tables

Revision History

Date	Revision	Description
September 2009	005	Added text (with changebars) in: • Section 5.1.3, “FreeBSD Sample Codelet” on page 22 • Section 5.6.1, “FreeBSD Compilation Instructions” on page 26
May 2009	004	Minor modifications to correct URLs for FreeBSD download and other text corrections. Change bars were not updated; change bars show edits from previous doc version (below).
March 2009	003	General updates to reflect support for FreeBSD* 7.1. Updated directory structure in Figure 1. Added Section 5.1.3, “FreeBSD Sample Codelet” on page 22 for CAN. Added Section 5.3.3, “FreeBSD Sample Codelet” on page 24 for WDT. Added Section 5.5.3, “FreeBSD Sample Codelet” on page 26 for IEEE 1588.
November 2008	002	General updates to reflect support for FreeBSD* 6.3. Updated directory structure in Figure 1. Updated mount command in Section 3.1.2, “Unpacking the EP80579 Integrated Processor FreeBSD Package” on page 17. Added Section 5.2.3, “FreeBSD Sample Codelet” on page 23.
July 2008	001	Initial release of this document.

§ §

1.0 Introduction

1.1 About this Manual

This Getting Started Guide documents the instructions to obtain, build (if necessary), install, and execute the software release package for the Intel® EP80579 Integrated Processor product line. Additionally, this document describes brief installation instructions for the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board.

Note: The "Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board" will be referred to as the "development board" in this document.

1.2 Additional Information on Software

The Intel® EP80579 Software Drivers for Embedded Applications on FreeBSD package has been validated with FreeBSD* 7.1.

1.2.1 Where to Find Current Software and Documentation

The software release and associated collateral can be found on the Hardware Design resource center.

1. In a web browser, go to http://www.intel.com/go/soc.
2. For Software and Pre-boot Firmware: Click on "Tools & Software" tab.
3. For Documentation: Click on "Technical Documents" tab.

1.2.2 Product Documentation

The following documentation is provided to support this software release:

- This Getting Started Guide
- Intel® EP80579 Software Drivers for Embedded Applications Release Notes
- Intel® EP80579 Software Drivers for Embedded Applications Programmer's Guide and API Reference Manual
- Software for Intel® EP80579 Integrated Processor Product Line PHY Porting Guide

Note: The Release Notes contains important information about each software release, such as the appropriate firmware version. Please review the Release Notes before proceeding with this document.

Please follow the directions in Section 1.2.1 to locate this collateral.

1.2.3 Pre-boot Firmware

The latest release of the development board pre-boot firmware (BIOS) is also located on Hardware Design resource center. Refer to the Intel® EP80579 Software Drivers for Embedded Applications Release Notes for latest version.

Please follow the directions in Section 1.2.1 to locate this firmware.

1.3 Related Software and Documentation

Refer to the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Kit User's Guide for information on the development board including board layout, components, connectors, jumpers, headers, power and environmental requirements, and pre-boot firmware.

Please follow the directions in Section 1.2.1 to locate this collateral.

1.4 Conventions

The following conventions are used in this manual:

- `Courier font` - commands and code examples
- Italics - directory names

1.5 Software Overview

1.5.1 Features Implemented

The software provides the following features:

- IEEE 1588 Hardware Assist Driver
- Controller Area Network (CAN) Hardware Access Driver
- Gigabit Ethernet (GbE) Controller Driver for Network Connectivity
- Advanced Host Controller Interface Software Support for SATA for Native Command Queuing and Hot Plug Capability
- SMBus Driver
- General Purpose I/O (GPIO) Hardware Access Driver
- Enhanced Direct Memory Access (EDMA) Hardware Assist Driver
- Watchdog Timer Hardware (WDT) Access Driver

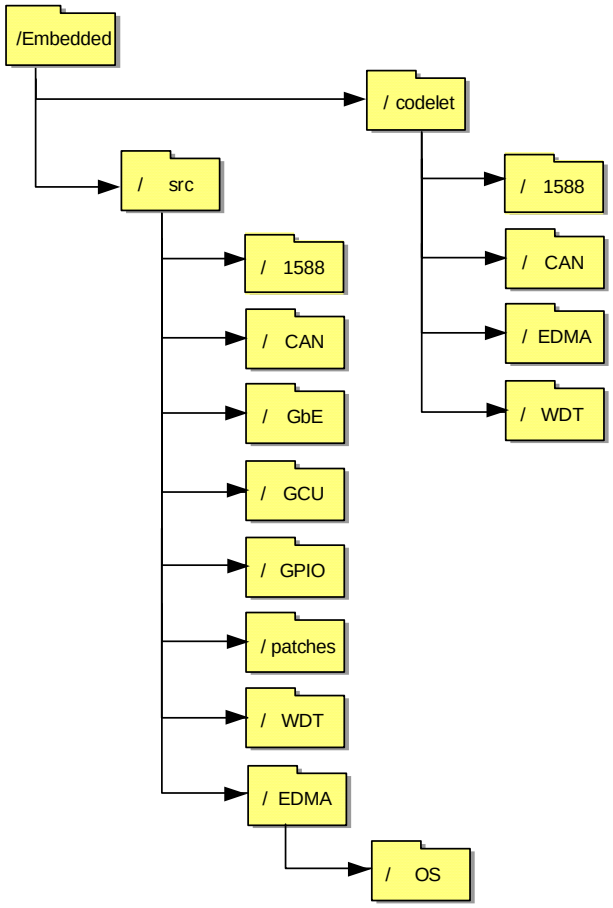
1.5.2 List of Files in Release

The Bill of Materials, sometimes referred to as the BOM, is included as a text file in the released software package. This text file is labeled "filelist" and is located at the top directory level for each release.

1.5.3 Package Release Structure

The package release structure is shown in Figure 1.

Figure 1. Software Package Release Structure - FreeBSD



2.0 Configuration Requirements

2.1 Development Board Configuration

Complete details about the development board can be found in the "Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Kit User's Guide". This document contains details on the design, structure, and function of all development board features.

To facilitate quick start of the EP80579 software drivers for embedded applications package, relevant sections from the Development Kit User's Guide have been included in this chapter. Please follow the directions in Section 1.2.1 for information on accessing the full User's Guide.

2.1.1 Package Components

The Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Kit includes the following:

- Development board containing the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology
- ATX12V power supply
- One DDR2-800 DIMM
- PCIe* graphics card
- SATA hard drive with cable
- SATA DVD-ROM with cable
- Two Controller Area Network cable connectors
- Power Cord (USA power cord supplied)

These items are not supplied from Intel. Please note this is not an exhaustive list of items not supplied.

- Mouse
- Keyboard
- Monitor
- Power supply cord (if country or region-specific power cord is required)

2.1.2 Development Kit Setup

Ensure that all components listed in Section 2.1.1 arrive together. Once all components have been identified and located, installation and setup can begin. This section describes how to set up the development board for operation.

Note: This document assumes that the user is familiar with the basic concepts required to install and configure hardware for a PC system.

2.1.3 Safety

The development board is shipped as an open system allowing for maximum flexibility in changing hardware configurations and peripherals in a lab environment. Since the board is not in a protective chassis, the user is required to take safety precautions in handling and operating the board. Some assembly is required before use.

Ensure a safe and static-free work environment before removing any components from their anti-static packaging. The development board is susceptible to electrostatic discharge that may cause failure or unpredictable operation. The development board must be operated on a flame-retardant surface because a chassis is not included with the board.

Caution: Connecting the wrong cable or reversing a cable may damage the board and may damage the device being connected. Since the board is not in a protective chassis, use caution when connecting cables to the board.

Caution: The power supply cord is the main disconnect device to main power (AC power). The socket outlet should be installed near the equipment and should be readily accessible. To avoid shock, ensure that the power cord is connected to a properly wired and grounded receptacle. Do not connect/disconnect any cables or perform installation/ maintenance of the boards in this product during an electrical storm. Ensure that any equipment to which this product will be attached is also connected to properly wired and grounded receptacles.

Note: Ensure that setting up the ATX power supply is the final step performed in the process of assembly.

2.1.4 Connecting the Serial ATA Hard Drive and Cable

The development board provides two Serial ATA (SATA) connectors. Connect cables to the appropriate drive sequentially, starting from Port 0 to Port 1. See Figure 2 and Table 1 for the location and identification of the SATA connectors.

Note: Intel recommends connecting the boot drive to SATA port 0.

2.1.5 Connecting the Keyboard and Mouse

Connect a PS/2 mouse and keyboard to the stacked PS/2 connector on the rear panel of the board. The bottom connector is the keyboard connector and the top connector is the mouse connector. Alternatively, a USB keyboard and a USB mouse can be connected to the development board's USB connectors.

Note: Mouse and keyboard are not supplied by Intel.

Note: The serial redirection feature can be enabled to remotely access the board through a serial cable without attaching a keyboard or mouse to the development board. Refer to the "Connecting the Serial Cable for Console Redirection" section of the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Kit User's Guide for more information.

2.1.6 Connecting the PCI Express Video Card

Populate the PCIe graphics card in any one of the PCIe slots.

2.1.7 Connecting the Serial ATA DVD-ROM Drive (Optional)

Connect the Serial ATA DVD-ROM drive to SATA Port 1 utilizing the cable that comes with the DVD-ROM drive. See Figure 2 and Table 1 for the location and identification of the SATA connectors.

2.1.8 Connecting the Power Cables

Use the following procedure to connect the power cables:

1. The board supports the use of ATX12V power supplies with either 2 x 10 or 2 x 12 main power cables.
2. Plug the main connector into the board. Ensure that the plug clip lines up with the clip lock and the connector pins easily fit into their appropriate slots. When using a power supply with a 2 x 10 main power cable, attach that cable to the right-most part of the main power connector, leaving pins 11, 12, 23 and 24 unconnected.
3. Plug in the power connectors from each of the SATA drives and disk drives.

2.1.9 Powering Up the System

Warning: Ensure the steps in the previous sections were strictly followed before powering up the system.

Use the following procedures to power up the development board:

1. Ensure that the processor heat sink and the fan are installed according to the procedure in the “Connecting the Processor Heatsink and Fan” section of the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board User’s Guide.
2. Leaving the On/Off switch in the OFF position, plug the power cable into the back of the power supply.
3. Once the board is set up, plug the cord into the power source.
4. Switch on the power supply.
5. Press the power-on button to start the system. Refer to Figure 2 for the location power-on button (item I, lower-left). Table 1 is a legend for key items labeled in Figure 2.

Figure 2. Development Board - Component and Connector Locations

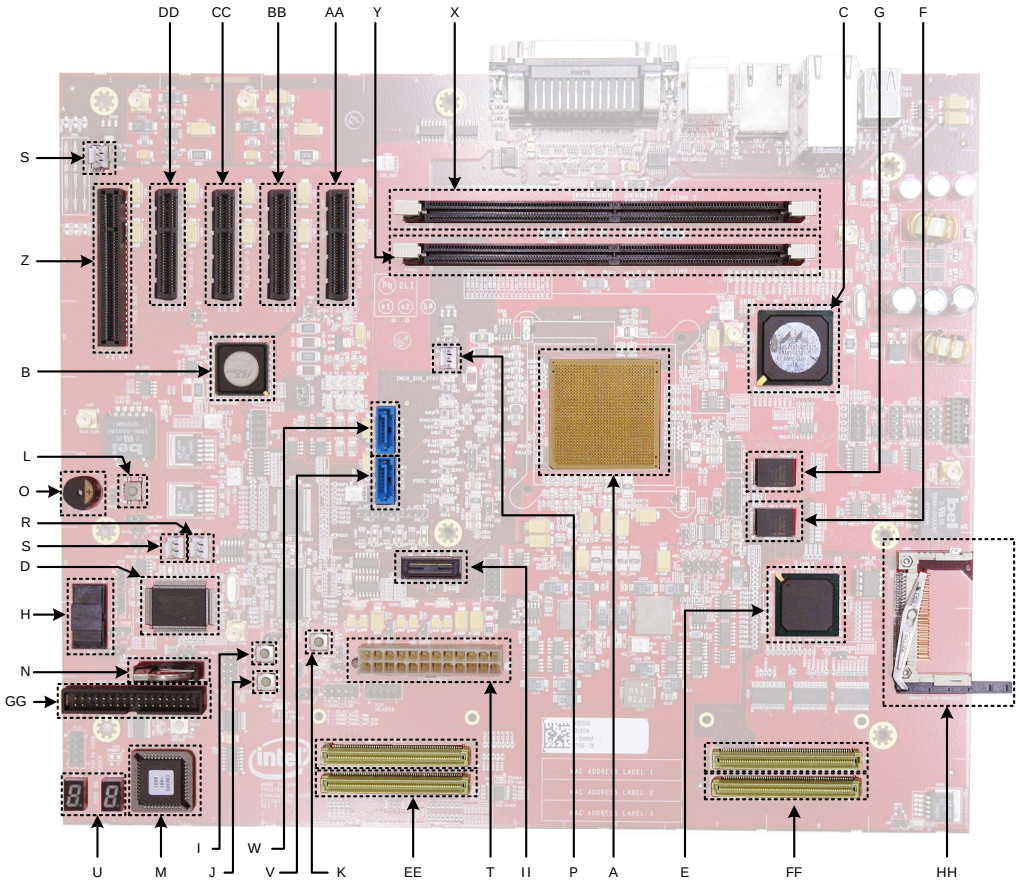
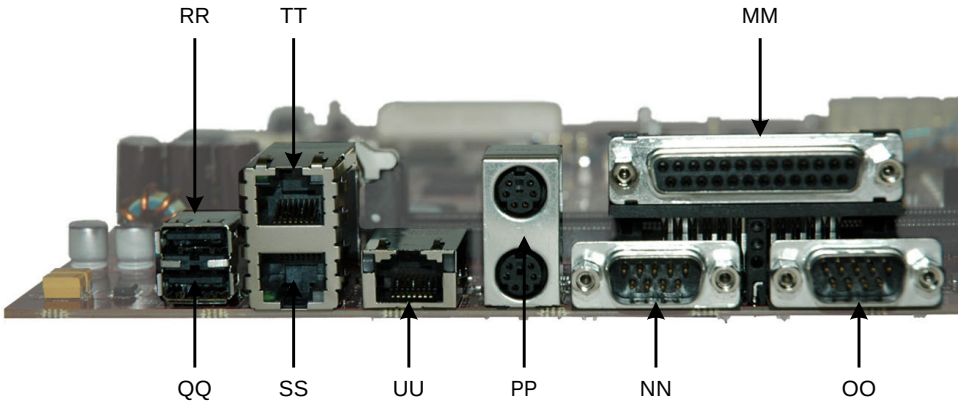


Figure 3. Side View of the Board Connectors



B6605-01

Table 1. Key Development Board Components and Connectors Legend

Callout	Component/Connector
A	Intel® EP80579 Integrated Processor
B	PEX PCIe Switch Chip
C	Marvell 8811E1 Quad PHY
D	Super IO Controller
E	FPGA
F	Flash memory 0
G	Flash memory 1
H	FWH
I	Power button
J	Reset button
K	Sleep button
L	PCIe Wake button
M	Port 80 IC
N	CMOS battery
O	On-board speaker
P	CPU FAN connector
Q	AUX FAN connector
R	AUX0 FAN connector
S	AUX1 FAN connector
T	ATX power connector
U	Two 7-segment display (Port 80)
V	SATA port 0
W	SATA port 1

Table 1. Key Development Board Components and Connectors Legend

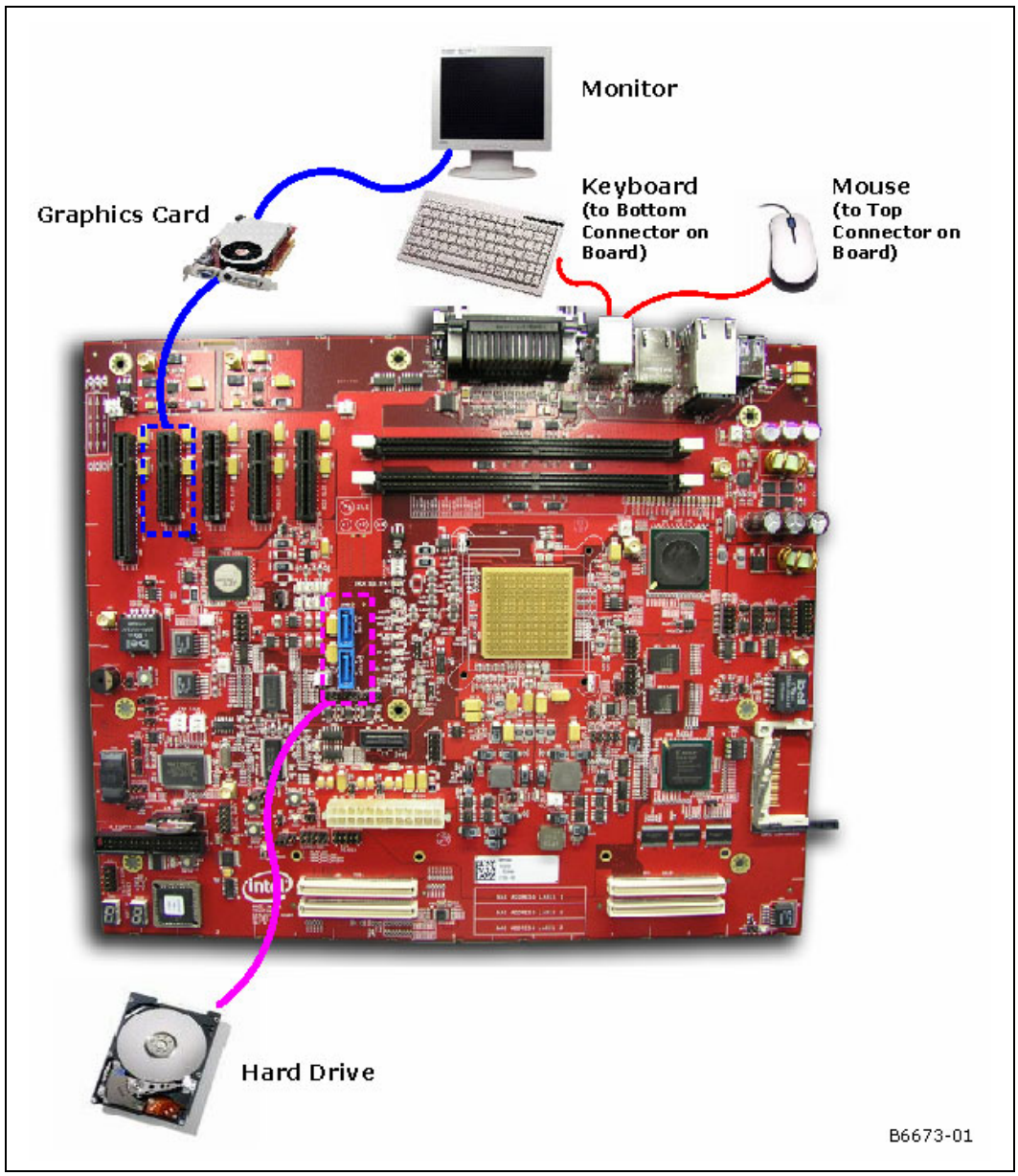
Callout	Component/Connector
X	DDR2 DIMM0
Y	DDR2 DIMM1
Z	Slot 0 x8 connector 4 lanes PCI Express
AA	Slot 1 x4 connector 1 lane PCI Express
BB	Slot 2 x4 connector 1 lane PCI Express
CC	Slot 3 x4 connector 1 lane PCI Express
DD	Slot 4 x4 connector 1 lane PCI Express
EE	Mezzanine connector 0
FF	Mezzanine connector 1
GG	Floppy Connector
HH	CF connector
II	ITP-XDP connector
LL	Mezzanine connector 3
MM	Parallel port
NN	COM1
OO	COM2
PP	PS/2 mouse (top)/keyboard (bottom)
QQ	USB port 0
RR	USB port 1
SS	RJ-45 Ethernet port 0
TT	RJ-45 Ethernet port 1
UU	RJ-45 Ethernet port 2
Note: Items JJ and KK are not shown in Figure 2 or Figure 3 and are intentionally omitted.	

2.2 Development Board Setup Requirements

2.2.1 Stand-alone Target System

Figure 4 shows the system setup when the target development board is also used for build and install.

Figure 4. Development Board System Setup



3.0 System Requirements for Installing an OS on a Development Board

3.1 Installing FreeBSD

FreeBSD* download software can be acquired from http://www.freebsd.org.

For complete installation instructions and the FreeBSD* 7.1 release Readme document, please refer to FreeBSD's web site at the following locations:

- http://www.freebsd.org/releases/7.1R
- http://www.freebsd.org/releases/7.1R/readme.html

The following are the instructions that Intel has used to install FreeBSD on the development board.

1. The user is shown a list of Boot options. Select: 1 - Boot FreeBSD [default]
2. Select the appropriate Country from the next list.
3. The user is shown a list of options for software installation. Select: Standard - "Begin a Standard installation (recommended)".
4. In the next few screens, the user will be asked to create a DOS-Style partition on the disk. There may be warnings about geometry being incorrect; however, these error messages can be ignored. Select: A = Use Entire Disk; and then confirm by selecting: Q = Finish.
5. Next install a Boot Manager. Select "Standard - Install a standard MBR (no boot manager)".
6. Create default sized BSD partitions inside the fdisk partition, created in a previous step. Select: A = Auto Defaults; and then confirm by selecting: Q = Finish.
7. Choose the distribution to use. Select: "4 Developer -- Full sources, binaries and doc but no games".
8. Answer [Yes] to the question: Would you like to install the FreeBSD ports collection?
9. Note the user is taken to the previous screen showing the various distributions. Distribution was selected in previous step so select: "X Exit - Exit this menu (returning to previous menu)" to continue the installation.
10. Identify the installation media. Select: "1 CD/DVD - Install from a FreeBSD CD/DVD".
11. The system warns you of loss of data in the disk. To continue with the installation, select "Yes". File systems are created and files copied. This step will take a few minutes to complete and display a Congratulations message at the end. After this message is displayed additional setup/configuration can be performed. The following steps provide guidance for the remaining configuration.
12. Do not configure any Ethernet or SLIP/PPP network devices.
13. Configure the machine to function as a Network Gateway.

14. In the next few screens inetd and related services can be enabled. Do not edit the default configuration file. The questions can be deceptive. Select Yes and No carefully.
15. Enable the SSH login.
16. Do not ask for Anonymous FTP access to your machine.
17. Do not configure the machine as a NFS server.
18. Do not configure the machine as a NFS client.
19. Do not customize the system console settings.
20. In the next few screens, the machines time zone is set. Carefully select Yes and No to these questions. Some are counter intuitive and install does not give the ability go back and correct.
21. Do not select Linux binary compatibility.
22. In the next few screens the Mouse can be configured. When finished, exit and go back to previous screen to continue.
23. The system informs you of the availability of ready-to-run applications. Select YES to select applications from the following:
    a. Select all items under: devel - Software development utilities and libraries
    b. Select preferred editors under: editors - Common text editors

Note: Numerous additional entries are selected as dependents of above choices.

24. Add a user account if required.
25. Set the root password.
26. Install will give an opportunity to set any last options. Select and set any options may have missed.
27. Select “Exit Install” to complete the installation.
28. At this point system will reboot. If the BIOS Boot Priority Option is set to select the Hard Disk in the absence of CD ROM, then the installed FreeBSD image will start and display the login prompt. Login using the root or user password.

3.1.1 Build Environment Requirements

FreeBSD 7.1 includes GCC-4.2.1. Refer to the FreeBSD 7.1 Release Notes for a list of contributed software. The FreeBSD 7.1 Release Notes for the i386 release can be found at the following location:

http://www.freebsd.org/releases/7.1R/relnotes.html

3.1.2 Unpacking the EP80579 Integrated Processor FreeBSD Package

The EP80579 software drivers for embedded applications package comes in the form of a tarball. See Section 1.2.1, “Where to Find Current Software and Documentation” on page 6 for the software location. The package can be unpacked at any location on the system, but for the purposes of this Getting Started Guide document, a recommendation is proposed.

Create a directory in the root directory called “*EP805XX_release*” with the following commands:

```
cd /
mkdir EP805XX_release
cd EP805XX_release
```

Transfer the tarball to the development board using any preferred method, for example, USB memory stick, CDROM or network transfer. Place and unpack the tarball in the */EP805XX_release* directory using the following command:

```
tar -xvzf <tarball name>
```

Tip: To mount a USB memory stick, issue the following commands:

```
mkdir /mnt/usb (Skip this command if directory exists)
mount -t msdosfs /dev/da0s1 /mnt/usb
```

A new directory structure is created under the */EP805XX_release* directory that contains all EP80579 embedded software drivers for FreeBSD. See Figure 1, “Software Package Release Structure - FreeBSD” on page 8 for a detailed view of the directory structure.

3.1.3 Patching the Kernel for PCI Device Recognition

Execute the following commands to patch the FreeBSD kernel:

```
cd /EP8057XX/Embedded/src/patches
patch -p0 < Intel_EP80579_FreeBSD_71.patch
```

Upon patch application, review the output. The output should contain text similar to:

```
Hunk #1 succeeded at 64...
```

This indicates that this part of the patch was applied successfully.

The following patch must be applied to ensure the correct device strings are returned by the pciconf utility. The patch file can be found in the */EP805XX_release/Embedded/src/patches* directory. Apply the patch to the pci_vendors.patch file using the following commands:

```
cd /usr/share/misc
patch -p0 < /EP805XX_release/Embedded/src/patches/pci_vendors.patch
```

Upon patch application, review the output. The output should contain text similar to:

```
Hunk #1 succeeded at 9268...
```

This indicates that the patch was applied successfully.

3.1.4 Recompiling the FreeBSD Kernel

Complete instructions on building and installing the FreeBSD kernel are available at the following location:

http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig-building.html

Recompile the kernel by executing the following commands:

```
cd /usr/src/sys/i386/conf
cp GENERIC MYKERNEL
cd /usr/src
make buildkernel KERNCONF=MYKERNEL
```

Note: Kernel compilation takes about 30 minutes to complete.

Install the recompiled kernel by executing the following commands:

```
cd /usr/src
make installkernel KERNCONF=MYKERNEL
```

Check if a new kernel was built by executing the following commands and examine the date and time that the kernel was produced:

```
cd /boot/kernel
ls -l kern*
```

Rebooting the system brings the system up with the newly compiled kernel.

§ §

4.0 Building and Installing EP80579 Software on a Target Development Board

This chapter provides instructions for setting up the environment and building/compilation instructions for the Intel® EP80579 Software Drivers for Embedded Applications package.

4.1 Environment Setup

There are no environment variables to set for the Intel® EP80579 Software Drivers for Embedded Applications package for FreeBSD*.

4.2 Build All Embedded Software Drivers

Change directory to the */EP805XX_release/Embedded* directory using the following command:

```
cd /EP805XX_release/Embedded
```

Build all software drivers in the package using the following commands:

```
make clean
make
make install
```

Under the */EP805XX_release/Embedded/src* directory are directories for each feature component. In each feature component directory, the compiled components will be placed for that component. The */EP805XX_release/Embedded/build* directory is created and copies of all .ko executables are placed there. Executables are also copied to /boot/kernel directory and load entries are added to /boot/loader.conf files. In this fashion, the EP80579 embedded software drivers are loaded at system boot.

After “make install”, a reboot is not required to use the drivers. At this point, installation of the drivers is complete.

Note: Instructions for building and installing drivers individually are also provided in this guide. See Chapter 5.0, “Runtime Configuration”.

4.3 Uninstalling Embedded Software Drivers

The make uninstall command is available to uninstall all drivers from the top level */EP805XX_release/Embedded* directory. A reboot is required to complete the uninstall process. To uninstall all the EP80579 embedded software drivers, execute the following command from the */EP805XX_release/Embedded* directory:

```
make uninstall
```

5.0 Runtime Configuration

This chapter describes how to compile and install EP80579 drivers individually.

5.1 Controller Area Network Driver

5.1.1 FreeBSD Compilation Instructions

All source files for the FreeBSD* release of the Controller Area Network (CAN) driver are located in the following directory within the FreeBSD compatible EP80579 embedded software drivers release:

/EP805XX_release/Embedded/src/CAN

Compilation of the FreeBSD CAN driver separately from the rest of the software package is possible. Change to the */EP805XX_release/Embedded/src/CAN* directory and execute the following commands:

```
make clean
make
make install (this will install the driver for persistency)
```

The CAN driver compiles and the resulting can.ko file is placed in the */EP805XX_release/Embedded/build* directory.

5.1.2 FreeBSD Module Load / Unload Instructions

Note: This step is not necessary if the make install command above succeeded. This is only necessary to understand how to load and unload this driver individually.

To load the FreeBSD CAN driver, execute the following command from the directory where the compiled executable resides:

```
kldload ./can.ko
```

To unload the FreeBSD CAN driver execute the following command:

```
kldunload can.ko
```

The kldstat command may be used to confirm if a module has been loaded or unloaded.

```
kldstat | grep can
```

The output of the kldstat command lists all modules loaded in the system. Look for an entry titled “can”.

5.1.3 FreeBSD Sample Codelet

A codelet is provided to demonstrate how a sample application interfaces with the Controller Area Network driver. This codelet is intended to run on an Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board with FreeBSD installed. The EP80579 Embedded Software kernel should also be installed, but the codelet works even if it is not. Run the codelet as follows:

Note: In order to run this application, make sure a loopback CAN cable is connected to CAN 0 and CAN 1 port through CAN header. For additional details, refer to the Readme document location in */EP805XX_release/Embedded/codelet/CAN*.

1. Login as root. Root permissions are required for all operations.
2. Open a terminal window, change directory to */EP805XX_release/Embedded/codelet/CAN* and execute the bash script. Do not move the script to another location.

   ```
   [root@localhost ~]# ./install.csh
   ```

   The script checks for the Controller Area Network driver and client drivers, runs all of the makefiles in the correct order, loads the client driver, and runs the application.

   The application output can be viewed in the terminal window.
3. Client driver output can be viewed in a second terminal window by printing /var/log/ messages as follows:

   ```
   [root@localhost ~]# tail -f /var/log/messages
   ```

4. For information about the CAN codelet, refer to the Readme document located at */EP805XX_release/Embedded/codelet/CAN.*

5.2 Enhanced Direct Memory Access Driver

5.2.1 FreeBSD Compilation Instructions

All source files for the FreeBSD release of the Enhanced Direct Memory Access (EDMA) driver are located in the following directory within the FreeBSD compatible EP80579 embedded software drivers release:

/EP805XX_release/Embedded/src/EDMA

Compilation of the FreeBSD EDMA driver separately from the rest of the software package is possible. Change to the */EP805XX_release/Embedded/src/EDMA* directory and execute the following commands:

```
make clean
make
make install (this will install the driver for persistency)
```

The EDMA driver compiles, is installed, and the resulting dma.ko file is placed in the */EP805XX_release/Embedded/build* directory.

5.2.2 FreeBSD Module Load/Unload Instructions

Note: This step is not necessary if the make install command above succeeded. This is only necessary to understand how to load and unload this driver individually.

To load the FreeBSD EDMA driver, execute the following command from the directory where the compiled executable resides:

```
kldload ./dma.ko
```

To unload the FreeBSD EDMA driver execute the following command:

```
kldunload dma.ko
```

The kldstat command may be used to confirm if a module has been loaded or unloaded:

```
kldstat | grep dma
```

The output of the kldstat command lists all modules loaded in the system. Look for an entry titled “dma”.

5.2.3 FreeBSD Sample Codelet

A codelet is provided to demonstrate how a sample application interfaces with the EDMA driver. This codelet is intended to run on an Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board with FreeBSD installed. The EP80579 Embedded Software kernel should also be installed, but the codelet works even if it is not. Run the codelet as follows:

1. Log in as root. Root privileges are required for some of the operations.
2. Open a terminal window, change directory to */EP805XX_release/Embedded/codelet/EDMA* and execute the following C-shell script. Do not move the script to another location.

   ```
   [root@localhost ~]# ./install.csh
   ```

   The script checks for the EDMA and client drivers, runs all of the makefiles in the correct order, loads the client driver, and runs the application.

   The application and client driver output can be viewed in the terminal window.
3. For information about the EDMA codelet, refer to the Readme document located at */EP805XX_release/Embedded/codelet/EDMA.*

5.3 WDT - Watchdog Timer

5.3.1 FreeBSD Compilation Instructions

All source files for the FreeBSD release of the Watchdog Timer (WDT) driver are located in the following directory within the FreeBSD compatible EP80579 embedded software drivers release:

/EP805XX_release/Embedded/src/WDT

Compilation of the FreeBSD WDT driver separately from the rest of the software package is possible. Enter the */EP805XX_release/Embedded/src/WDT* directory and execute the following commands:

```
make clean
make
make install (this will install the driver for persistency)
```

The WDT driver compiles, is installed, and the resulting iwdt.ko file is placed in the */EP805XX_release/Embedded/build* directory.

5.3.2 FreeBSD Module Load/Unload Instructions

Note: This step is not necessary if the make install command above succeeded. This is only necessary to understand how to load and unload this driver individually.

To load the FreeBSD Watchdog Timer driver, execute the following command from the directory where the compiled executable resides:

```
kldload ./iwdt.ko
```

To unload the FreeBSD Watchdog Timer driver execute the following command:

```
kldunload iwdt.ko
```

The kldstat command may be used to confirm if a module has been loaded or unloaded:

```
kldstat | grep iwdt
```

The output of the kldstat command lists all modules loaded in the system. Look for an entry titled “iwdt”.

5.3.3 FreeBSD Sample Codelet

A codelet is provided to demonstrate how a sample application interfaces with the Watchdog Timer driver. This codelet is intended to run on an Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board with FreeBSD installed. The EP80579 Embedded Software kernel should also be installed, but the codelet works even if it is not. Run the codelet as follows:

1. Login as root. Root permissions are required for all operations.
2. Open a terminal window, change directory to */EP805XX_release/Embedded/codelet/WDT* and execute the bash script. Do not move the script to another location.

   ```
   [root@localhost ~]# ./install.csh
   ```

   The script checks that the Watchdog Timer driver is loaded, runs all of the makefiles in the correct order, and runs the application.

   The application output can be viewed in the terminal window.
3. For information about the WDT codelet, refer to the Readme document located at */EP805XX_release/Embedded/codelet/WDT.*

5.4 GPIO

5.4.1 FreeBSD Compilation Instructions

All source files for the FreeBSD release of General Purpose I/O (GPIO) driver are located in the following directory within the FreeBSD compatible EP80579 embedded software drivers release:

/EP805XX_release/Embedded/src/GPIO

Compilation of the FreeBSD GPIO driver separately from the rest of the software package is possible. Enter the */EP805XX_release/Embedded/src/GPIO* directory and execute the following commands:

```
make clean
make
```

```
make install (this will install the driver for persistency)
```

The GPIO driver compiles, is installed, and the resulting GPIO.ko file is placed in the */EP805XX_release/Embedded/build* directory.

5.4.2 FreeBSD Module Load / Unload Instructions

Note: This step is not necessary if the make install command above succeeded. This is only necessary to understand how to load and unload this driver individually.

To load the FreeBSD General Purpose I/O driver, execute the following command from the directory where the compiled executable resides:

```
kldload ./gpio.ko
```

To unload the FreeBSD General Purpose I/O driver, execute the following command:

```
kldunload gpio.ko
```

The kldstat command may be used to confirm if a module has been loaded or unloaded:

```
kldstat | grep gpio
```

The output of the kldstat command lists all modules loaded in the system. Look for an entry titled “gpio”.

5.5 IEEE 1588 Hardware Assist

5.5.1 FreeBSD Compilation Instructions

All source files for the FreeBSD release of IEEE 1588 Hardware Assist driver are located in the following directory within the FreeBSD compatible EP80579 embedded software drivers release:

```
/EP805XX_release/Embedded/src/1588
```

Compilation of the FreeBSD IEEE 1588 driver separately from the rest of the software package is possible. Enter the */EP805XX_release/Embedded/src/1588* directory and execute the following commands:

```
make clean
make
make install (this will install the driver for persistency)
```

The IEEE 1588 Hardware Assist driver compiles, is installed, and the resulting timesync.ko file is placed in the */EP805XX_release/Embedded/build* directory.

5.5.2 FreeBSD Module Load / Unload Instructions

Note: This step is not necessary if the make install command above succeeded. This is only necessary to understand how to load and unload this driver individually.

To load the FreeBSD IEEE 1588 Hardware Assist driver, execute the following command from the directory where the compiled executable resides:

```
kldload ./timesync.ko
```

To unload the FreeBSD IEEE 1588 Hardware Assist driver, execute the following command:

```
kldunload timesync.ko
```

The kldstat command may be used to confirm if a module has been loaded or unloaded:

```
kldstat | grep timesync
```

The output of the kldstat command lists all modules loaded in the system. Look for an entry titled “timesync”.

5.5.3 FreeBSD Sample Codelet

A codelet is provided to demonstrate how a sample application interfaces with the IEEE 1588 Hardware Assist driver. This codelet is intended to run on an Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board with FreeBSD installed. The EP80579 Embedded Software kernel should also be installed, but the codelet works even if it is not. Run the codelet as follows:

1. Login as root. Root permissions are required for all operations.
2. Open a terminal window, change directory to */EP805XX_release/Embedded/codelet/1588* and execute the bash script. Do not move the script to another location.

   ```
   [root@localhost ~]# ./install.csh
   ```

   The script checks for the IEEE 1588 Hardware Assist driver and client drivers, runs all of the makefiles in the correct order, loads the client driver, and runs the application.

   The application output can be viewed in the terminal window.
3. Client driver output can be viewed in a second terminal window by printing /var/log/ messages as follows:

   ```
   [root@localhost ~]# tail -f /var/log/messages
   ```

4. For information about the 1588 codelet, refer to the Readme document located at */EP805XX_release/Embedded/codelet/1588.*

5.6 Global Configuration Unit and Gigabit Ethernet Driver

Two drivers complete the software features set for network connectivity on the EP80579 integrated Gigabit Ethernet controllers; the Global Configuration Unit (GCU) driver and the Gigabit Ethernet (GbE) driver. The GCU driver controls the MAC and administrative activities. The GbE driver controls the network connectivity. The GbE driver is dependent on the GCU driver.

Note: The Global Configuration Unit driver must be installed prior to installation of the Gigabit Ethernet driver.

5.6.1 FreeBSD Compilation Instructions

All source files for the FreeBSD release of the EP80579 Global Configuration Unit driver and Gigabit Ethernet driver are located in the following directories within the FreeBSD compatible EP80579 embedded software drivers release:

```
/EP805XX_release/Embedded/src/GbE
/EP805XX_release/Embedded/src/GCU
```

Compilation of the FreeBSD GCU and GbE drivers separately from the rest of the software package is possible. Ensure that GCU drivers are built and loaded before the GbE drivers. Enter the */EP805XX_release/Embedded/src/GbE* and */EP805XX_release/Embedded/src/GCU* directory and execute the following commands:

```
make clean
make
make install (this will install the driver for persistency)
```

The Gigabit Ethernet driver compiles, is installed, and the resulting iegbe.ko file is placed in the */EP805XX_release/Embedded/build* directory.

5.6.2 FreeBSD Module Load / Unload Instructions

Note: This step is not necessary if the make install command above succeeded. This is only necessary to understand how to load and unload this driver individually.

To load the FreeBSD Gigabit Ethernet driver, execute the following command from the directory where the compiled executable resides:

```
kldload ./gcu.ko
kldload ./iegbe.ko
```

To unload the FreeBSD Gigabit Ethernet driver, execute the following command:

```
kldunload iegbe.ko
kldload gcu.ko
```

The kldstat command may be used to confirm if a module has been loaded or unloaded:

```
kldstat | grep iegbe
kldstat | grep gcu
```

The output of the kldstat command lists all modules loaded in the system. Look for an entry titled “iegbe” and “gcu”.

Configuration of the Gigabit Ethernet ports provided by the EP80579 integrated processor is through the traditional FreeBSD network command, ifconfig. Please consult the man pages for ifconfig for details.

5.7 SMBus

5.7.1 FreeBSD Compilation Instructions

All source files for the SMBus FreeBSD support are available in the FreeBSD kernel’s open source.

5.7.2 FreeBSD Module Load / Unload Instructions

Three drivers must be loaded to support SMBus:

- smb.ko
- smbus.ko
- ichsmb.ko

The smb.ko and smbus.ko drivers are installed using one command:

```
kldload smb
```

The ichsmb.ko driver is installed using the following command:

```
kldload ichsmb
```

These drivers can be unloaded using the following commands:

```
kldunload smb.ko
kldunload smbus.ko
kldunload ichsmb.ko
```

To confirm these modules have been loaded or unloaded, execute the following command:

```
kldstat
```

The output of the kldstat command lists all modules loaded in the system. Look for entries entitled “smb”, “smbus”, and “ichsmb”.

6.0 Pre-boot (BIOS) Firmware

The pre-boot firmware is executed when the system is powered up or reset. It initializes and configures system memory, devices and buses/interfaces.

The pre-boot firmware is based on the AMI Aptio* 4.5 core and compliant to EFI v1.1. The firmware is stored in the Firmware Hub (FWH) or SPI (Serial Peripheral Interface) Flash; the FWH or SPI Flash can be updated using a flash utility tool that is provided by Intel or by using a floppy drive connected to the floppy header.

The pre-boot firmware setup menu can be used to view and modify the system settings for the development board. The setup menu is accessed by pressing the <Del> key during pre-boot firmware boot up (before the operating system boot begins). The setup menu bar is shown in Table 2.

6.1 Pre-boot Firmware Setup Menu

Table 2 shows the pre-boot firmware setup main menu and provides a brief description of each menu option. Table 3 shows the function keys that can be used when navigating and selecting options from pre-boot firmware menus.

Table 2. Pre-boot Firmware Setup Main Menu

Main	Advanced	Chipset	Security	Boot	Exit
Displays processor and memory configuration Setup for CMOS system date and time	Configures advanced features and settings	Configures different major components	Setup passwords and security features	Selects boot options and configurations	Saves or discards changes to setup program options

Table 3. Pre-boot Firmware Setup Program Function Keys

Function Key	Description
< or >	Moves cursor left or right in the main menu
^ or v	Moves cursor up or down to select sub-menu items
Enter	Executes command or selects the submenu
F7	Discard changes
F8	Load the fail-safe default
F9	Load the optimal default configuration for the current menu
F10	Save the current configuration and exit the setup menu
ESC	Exit the setup menu

6.1.1 Serial Console Redirection

The pre-boot firmware supports redirection of both video and keyboard via a serial port. When console redirection is enabled, the remote console terminal sends keystrokes to the Intel® EP80579 Development Board pre-boot firmware and the pre-boot firmware redirects the video to the console terminal.

As an option, the Intel® EP80579 Development Board can be operated without keyboard or video and can run entirely via the remote serial console. This includes accessing the pre-boot firmware setup menu.

Console redirection ends when operating system boot up begins. After boot up begins, the operating system is responsible for continuing the redirection.

Note: pre-boot firmware console redirection is text only. Graphical data, such as logos, are not redirected.

Table 4 shows the default settings of the serial console redirection.

Table 4. Serial Console Redirection Default Settings

Parameter	Default
Port Number	COM 1
Baud Rate	115200
Data Bits	8
Parity	None
Stop bits	1
Flow Control	None

6.1.2 Changing the Boot Device

Use the following procedure to change the boot device:

1. Press the <Del> key during POST to enter the pre-boot firmware setup menu.
2. Use the arrow keys to navigate to the <BOOT> menu.
3. Move the cursor to <Boot Device Priority>.
4. Select the desired booting sequence list.

Note: Follow the instructions on the right side of the pre-boot firmware screen to navigate and change pre-boot firmware settings.

6.1.3 Maximum Memory Speed Setup

The maximum memory speed supported on the development board can be selected using the Maximum Memory Speed Setup option available in the BIOS Setup Menu on the Chipset tab.

Enter the BIOS Setup Menu and select the Chipset tab. Select North bridge, and navigate down to the bottom option, titled Max Memory Speed Support. Select this option using the Enter button. A selection box appears providing the following options:

- 400 MHz
- 533 MHz
- 667 MHz
- 800 MHz

The default setting in the BIOS is 400 MHz. If a higher speed memory DIMM is inserted into the development board, the corresponding memory speed must be selected in the BIOS Setup Menu to support the intended speed. Otherwise, the memory is reduced to the default of 400 MHz.

6.1.4 Coherent and Non-Coherent Memory Allocation

The development board supports allocation of memory regions for coherent and non-coherent use. Coherent and non-coherent memory use features are for development boards that use the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology. Intel® EP80579 Integrated Processors (without Intel® QuickAssist Technology) do not make use of the memory set aside for these features. Therefore, it is recommended that the settings for coherent and non-coherent usage be set to zero so that no memory is allocated to these features.

Enter the BIOS Setup Menu and select the Chipset tab. Select North bridge, and navigate down towards the bottom to the Coherent Mem Size option and press Enter to select this option. A dialog box is displayed prompting the user to enter a value. Type the numerical value zero, “0”, and press Enter. Navigate to the next option, Non-Coherent Mem Size, and press Enter to select this option. A dialog box is displayed prompting the user to enter a value. Again, type the numerical value zero, “0”, and press enter. These selections override the BIOS default settings and allocate no memory regions for these two features.

Note: This software package requires the pre-boot firmware (BIOS) for your hardware to allocate the values for each region called out in Table 5. For more information on these regions, refer to the Intel® EP80579 Integrated Processor Product Line Datasheet, Section 3.0.

Table 5. Memory Allocation Settings

Datasheet name	Software name	Region Size
IA/ASU Shared (Coherent)	CDRAM	0
IA/ASU Shared (AIOC-Direct)	NCDRAM	0

6.1.5 Legacy and AHCI SATA Mode

The development board supports hard drives in legacy SATA mode and in Advanced Host Controller Interface (AHCI) mode. AHCI mode provides advanced capabilities and improved performance, provided the hard drive supports the following features:

- Hot Plug
- Native Command Queuing
- Speeds up to 3 Gb/s

Refer to the Serial ATA Organization web site for more information:

http://www.serialata.org/

The development board pre-boot firmware (BIOS) can be configured in either Legacy or AHCI mode as desired. The BIOS defaults to Legacy mode because not all hard drives support AHCI. To toggle the BIOS to either Legacy or AHCI mode, proceed as follows:

1. Press the <Del> key during POST to enter the pre-boot firmware setup menu.
2. Use the arrow keys to navigate to the Advanced menu.

3. Use the arrow keys to navigate to the IDE Configuration option.
4. Select the IDE Configuration option.
5. Use the arrow keys to navigate to the SATA Mode option.
6. Press the Enter key. A SATA Mode popup window appears.
7. Select either Legacy or AHCI as desired. Do **not** use Native as a selection.
8. Press F4 to save.
9. Choose Yes. The system continues the boot process.

6.2 Pre-boot Firmware Image Reflashing Instructions

One method is available for updating the pre-boot firmware flash images located on the development board firmware hub:

- AFUEFI Flash Recovery

It is possible that updated pre-boot firmware images will become available from Intel for the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board. The latest pre-boot firmware image is available from Intel's public web site, http://www.intel.com/go/soc located with all other collateral related to the EP80579 integrated processor.

If the pre-boot firmware image should become corrupted on the board, also utilize these instructions to reflash the image to the firmware hub.

6.2.1 Aptio Flash Update Utility (AFUEFI)

Use the following instructions to update the development board pre-boot firmware image using a USB memory stick and the Aptio Flash Update Utility from AMI.

Necessary Hardware:

- development board
- Socketed Firmware Hub
- USB Memory Stick

Necessary Software:

- development board pre-boot firmware Image
- Aptio Flash Update Utility - AFUEFI

Steps to Reflash Image:

1. Load the AFUEFI utility onto the USB memory stick.
2. Load the pre-boot firmware image onto the USB memory stick.
3. Boot the development board to the EFI shell. Change the boot setting in the BIOS Setup Selection to boot from the EFI shell if needed.
4. Insert the USB memory stick into the USB port.
5. Once the USB memory stick is recognized on the system (activity seen on the USB memory stick), several commands are available as follows:
   — Type "map -r" to list all devices available.
   — Type "fs0:" to enter USB device.
   — Type "ls" to list all files.

6. Once the “fs0:” command has been initiated, execute the AFUEFI utility.
   — Type “AFUEFI <pre-boot firmware image name> /X /P /B /N”
   (the <pre-boot firmware image name> will be similar to TRXTG055.ROM)
7. Reboot the development board once reflashing has completed.
8. Confirm the image has been updated to the reflashed image by looking in BIOS Setup.

7.0 Uninstalling the Software

Please refer to the instructions for individual loading and unloading modules in Chapter 5.0, “Runtime Configuration”. To uninstall all drivers, see the last paragraph in Section 4.3, “Uninstalling Embedded Software Drivers” on page 20.

7.1 FreeBSD Modules / Driver Dependencies

Table 6 lists the dependencies for the driver modules or patch within the Intel® EP80579 Software Drivers for Embedded Applications package. OS installation is assumed.

Table 6. FreeBSD Module / Driver Dependencies

Module / Driver / Patch	Dependency 1	Dependency 2
EP80579 integrated processor FreeBSD Patch	OS Source	None
Controller Area Network (CAN)	None	-
Enhanced DMA (EDMA)	None	-
IEEE 1588 Hardware Assist (1588)	None	-
General Purpose IO (GPIO)	None	-
Gigabit Ethernet (GbE)	GCU	None
Global Configuration Unit (GCU)	None	-
SMBus (ichsmb)	smb	smbus
Watchdog Timer (WDT)	None	-

8.0 Troubleshooting

- It is advised to plug the Matrox Millennium G550 PCIe* graphics card into the PCI Express* slot closest to memory. Lack of video has been exhibited in the PCI Express x4 slot.
- When using the Matrox Millennium G550 PCIe graphics card, utilize the bottom video port on this dual port video card. The top port will not output any video to the display.
- Some LCD monitors will not function properly with the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board. In one case, an Acer monitor, model AL1916 C reported “Input Not Supported” upon boot complete. Try an alternative LCD monitor if no video support is displayed.
- Upon installation of FreeBSD* 7.1, a warning message will be displayed about the geometry of the disk drive. It is ok to accept the disk geometry that FreeBSD is displaying.

9.0 Glossary

AHCI	Advanced Host Controller Interface
CAN	Controller Area Network
EDMA	Enhanced Direct Memory Access
GbE	Gigabit Ethernet
GCU	Global Configuration Unit
GPIO	General Purpose Input Output
IEEE	Institute of Electrical and Electronics Engineers
IHS	Integrated Heat Spreader
SMbus	System Management Bus
TIM	Thermal Interface Material
WDT	Watchdog Timer

§ §