Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology for Linux*

Getting Started Guide

September 2009

Order Number: 320182-005US

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web Site.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2009, Intel Corporation. All rights reserved.

Contents

1.0 Introduction 10
1.1 About this Manual 10
1.2 Additional Information on Software 10
1.2.1 Where to Find Current Software and Documentation 10
1.2.2 Product Documentation 11
1.2.3 Pre-Boot Firmware 11
1.3 Related Software and Documentation 11
1.3.1 Open Source Software Required 11
1.4 Conventions 11
1.5 Software Overview 11
1.5.1 Features Implemented 11
1.5.2 List of Files in Release 12
1.5.3 Package Release Structure 12

2.0 Configuration Requirements 14
2.1 Development Board Configuration 14
2.1.1 Package Components 14
2.1.2 Development Kit Setup 14
2.1.3 Safety 15
2.1.4 Connecting the Serial ATA Hard Drive and Cable 15
2.1.5 Connecting the Keyboard and Mouse 15
2.1.6 Connecting the PCI Express* Video Card 15
2.1.7 Connecting the Serial ATA DVD-ROM Drive (Optional) 16
2.1.8 Connecting the Power Cables 16
2.1.9 Powering Up the System 16
2.2 Development Board Setup Requirements 21

3.0 Installing the OS on a Development Board 22
3.1 System Requirements 22
3.2 Acquiring CentOS 5.2 Linux 22
3.3 Installing CentOS 5.2 Linux 22
3.3.1 Recommended Installation Customizations 23
3.4 Installing CentOS 5.2 Kernel Source 24
3.5 Unpacking the EP80587 Security Software Linux Package 26
3.6 Updating PCI Device Names 26
3.7 Rebuilding the Kernel 26
3.7.1 Restore the Old Configuration 26
3.7.2 Modifying the Makefile 27
3.7.3 Building and Installing the Patched Kernel Source 27
3.7.4 Making the New Kernel Default 27
3.7.5 Rebooting and Verifying 28

4.0 Building EP80579 Security Software on a Target Development Board 29
4.1 Environment Setup 29
4.2 Build Options 29
4.3 Build Using Top Level Make 29
4.4 Installation of Build Output 31
4.4.1 Statistics Collection 31
4.5 Uninstalling Embedded and Security Kernel Modules 33

5.0 Runtime Configuration 34
5.1 Using the Intel® QuickAssist Technology Cryptographic API 34

6.0 Building, Installing and Loading Individual EP80579 Embedded Software Drivers ... 35

6.1 Global Configuration Unit and Gigabit Ethernet Drivers....35
6.1.1 Linux Compilation Instructions....35
6.1.2 Linux Module Load/Unload Instructions....35
6.1.3 Runtime Configuration of GCU and Gigabit Ethernet....36

7.0 Sample Applications....37

8.0 Pre-Boot Firmware (BIOS)....38
8.1 Pre-Boot Firmware Setup Menu....38
8.1.1 Serial Console Redirection....39
8.1.2 Changing the Boot Device....39
8.1.3 Maximum Memory Speed Setup....39
8.1.4 Coherent and Non-Coherent Memory Allocation....40
8.1.5 Legacy and AHCI SATA Mode....40
8.2 Pre-Boot Firmware Image Reflashing Instructions....41
8.2.1 Aptio Flash Update Utility (AFUEFI)....41

9.0 Uninstalling the Software....42
9.1 Linux Module/Driver Dependencies....42

10.0 Troubleshooting....43
10.1 Using a Graphics Card....43
10.2 Using the Serial Port....43

11.0 Glossary....45

Figures

1 Development Board - Top View of Component and Connector Locations....17
2 Development Board - Bottom View of Component and Connector Locations....18
3 Development Board - Side View of the Board Connectors....19
4 Development Board System Setup....21

Tables

1 Development Board - Key Components and Connectors Legend....19
2 Output Files Created in the $ICP_BUILD_OUTPUT Directory....31
3 Pre-Boot Firmware Setup Main Menu....38
4 Pre-Boot Firmware Setup Program Action Keys....38
5 Serial Console Redirection Default Settings....39
6 Linux Module/Driver Dependencies....42

Revision History

Date	Revision	Description
September 2009	005	Updates for 1.0.3 release including: • New Linux kernel version 2.6.28 supported • The EP80579 security software release package version 1.0.3 does not support OpenBSD/ FreeBSD Cryptographic Framework (OCF), OCF-Linux or any open source projects such as Openswan*, OpenSSL*, or Racoon*. Text describing this functionality has been removed and is not noted with changebars.
June 2009	004	• Provided support for CentOS 5.2 Linux* with major modifications in Chapter 3.0, "Installing the OS on a Development Board" and minor edits elsewhere in the document. • Removed Section 3.7, "Suggested Code Changes for Improving IPSec Performance" as the code changes are implemented in the Embedded software package. • Added Section 4.4.1, "Statistics Collection" on page 27. • Updated Section 6.0, "Build and Install Openswan" on page 35. • Added Linux sample codelet information in Section 9.1.3, Section 9.2.4, Section 9.3.4, and Section 9.5.4. • Removed "Using Debug Tools" section and added pointer to Programmer's Guide. • Added Section 7.0, "Sample Applications" on page 33. • Restructured Section 8.1.4, "Coherent and Non-Coherent Memory Allocation" on page 36 for clarity. • Other modifications noted with change bars.
December 2008	003	Updated Section 1.3.1, "Open Source Software Required" on page 7 Added Section 3.7, "Suggested Code Changes for Improving IPSec Performance" Updated Section 6.3, "Building and Installing OpenSSL" on page 36 Other changes identified by change bars.
September 2008	002	Updated Section 1.5.3, "Package Release Structure" on page 10. Updated Section 3.7.2, "Configuring OCF Parameters" on page 26. Updated Section 4.3, "Build Using Top Level Make" on page 30. Added information for alternate PHY support (see Section 9.6.4, "Alternative PHY Compilation" on page 49).
August 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 Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology release package. Additionally, this document describes some brief instructions on configuring the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board.

In this document, for convenience:

- "Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Board" is referred to as the "development board"
- "Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology" is referred to as the "EP80579 security software"

Note: The release package includes a directory with sample code.

1.2 Additional Information on Software

Note: The EP80579 security software release package version 1.0.3 does **not** support OpenBSD/FreeBSD Cryptographic Framework (OCF), OCF-Linux, or any open source projects such as Openswan*, OpenSSL*, or Racoon*. Text describing this functionality has been removed and is not noted with changebars. If your application requires OCF, you must use security software package version 1.0.2 which includes shim software to enable OCF support.

The EP80579 security software release package for Linux has been validated with CentOS 5.2 Linux using Kernel 2.6.28.10.

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.

Note: The EP80579 security software release package contains encryption software and is subject to export requirements defined by the U.S Department of Commerce. To satisfy these requirements, the End User Certification Form must be filled out and submitted for review/approval. Instructions on this process are included during the download process. Please note that this process may take up to two business days to complete.

1.2.2 Product Documentation

The following documentation supporting this release can be accessed as described in Section 1.2.1:

- Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology for Linux* Getting Started Guide (this document)
- Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology Release Notes
- Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology Programmer's Guide
- Intel® EP80579 Software for Security Applications on Intel® QuickAssist Technology Cryptographic API Reference Manual
- Intel® EP80579 Software on Intel® QuickAssist Technology Debug Services API Reference Manual
- Intel® EP80579 Software Drivers for Embedded Applications Programmer's Guide and API Reference Manual

1.2.3 Pre-Boot Firmware

The latest release of the development board pre-boot firmware (BIOS) is also located on the Hardware Design resource center.

Please refer to Release Notes listed in Section 1.2.1 for the correct firmware version.

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.

1.3.1 Open Source Software Required

Depending on your applications, the following table shows required open source software.

Item	URL
Linux Kernel source	http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.28.10.tar.gz

1.4 Conventions

The following conventions are used in this manual:

- `Courier font` - commands and code examples

1.5 Software Overview

1.5.1 Features Implemented

The software provides the following features:

- Gigabit Ethernet (GbE) Controller Driver for Network Connectivity

- Support for Symmetric Cryptography (Synchronous and Asynchronous modes)
  - Ciphers:
    - NULL Cipher
    - DES (ECB, CBC),
    - 3DES (ECB, CBC, CTR)
    - AES (ECB, CBC, CTR, CCM, GCM)
    - ARC4
  - Hash Algorithms:
    - MD5
    - SHA-1
    - SHA-2 (224, 256, 384, 512)
    - Nested Hashing
  - Authentication Schemes:
    - AES-XCBC-MAC-96
    - HMAC-MD-5
    - HMAC-SHA-1
    - HMAC-SHA-2 (224, 256, 384, 512)
  - Chaining Cipher and Authentication algorithms
- Support for Public Key Cryptography (Synchronous and Asynchronous modes)
  - Large Number Modular Exponentiation and Inversion
  - RSA (Key Generation, Encrypt, Decrypt)
  - DSA (Parameter Generation and Signatures)
  - DH (Phase1 and Phase2 secret key generation)
  - SSL/TLS Key and Mask generation
  - True Random Number Generation (RNG)
  - Prime Number Tests

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 high-level package release directory structure is shown as follows:

Note: The sample code is contained in the ./Acceleration/library/icp_crypto/look_aside_crypto/src/sample_code/ directory.

```
./Acceleration
./Acceleration/drivers
./Acceleration/drivers/icp_asd
./Acceleration/firmware
./Acceleration/include
./Acceleration/include/dcc
./Acceleration/include/lac
./Acceleration/library
./Acceleration/library/common
./Acceleration/library/common/include
./Acceleration/library/icp_crypto
./Acceleration/library/icp_crypto/look_aside_crypto
./Acceleration/library/icp_crypto/look_aside_crypto/include
./Acceleration/library/icp_crypto/look_aside_crypto/src
./Acceleration/library/icp_crypto/look_aside_crypto/src/common
```

```
./Acceleration/library/icp_crypto/look_aside_crypto/src/linux
./Acceleration/library/icp_crypto/look_aside_crypto/src/sample_code
./Acceleration/library/icp_crypto/QATAL
./Acceleration/library/icp_crypto/QAT_FW
./Acceleration/library/icp_debug
./Acceleration/library/icp_debugmgmt
./Acceleration/library/icp_debugmgmt/MIL
./Acceleration/library/icp_services
./Acceleration/library/icp_services/ResourceManager
./Acceleration/library/icp_services/RuntimeTargetLibrary
./Acceleration/library/icp_utils
./Acceleration/library/icp_utils/OSAL
./build_system
./Embedded
./Embedded/src
./Embedded/src/GbE
```

2.0 Configuration Requirements

2.1 Development Board Configuration

The Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology Development Kit User's Guide contains complete details on the development board. The document describes the design, structure, and function of all development board features.

To facilitate quick start of the Software for Intel® EP80579 Integrated Processor product line, relevant sections from the development kit User's Guide have been included in this chapter. Please follow the directions in Section 1.2.1 to access the full User's Guide.

2.1.1 Package Components

The development kit includes the following:

- A 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)

The following items are required, but not supplied by Intel:

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

Note: Additional items may be required but are not supplied by Intel.

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 when 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 provides the main connection to 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 the step to set 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 1 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 USB connectors on the development board.

Note: The 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 1 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 (labeled on the board) unconnected.
3. Plug in the power connectors from each of the SATA 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 procedure 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 Kit User’s Guide.
2. Leaving the On/Off switch in the OFF position, connect 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 1 for the location of the power-on button (item I, lower-left).

Note: Table 1 is a legend for key items labeled in Figure 1, Figure 2 and Figure 3.

Figure 1. Development Board - Top View of Component and Connector Locations

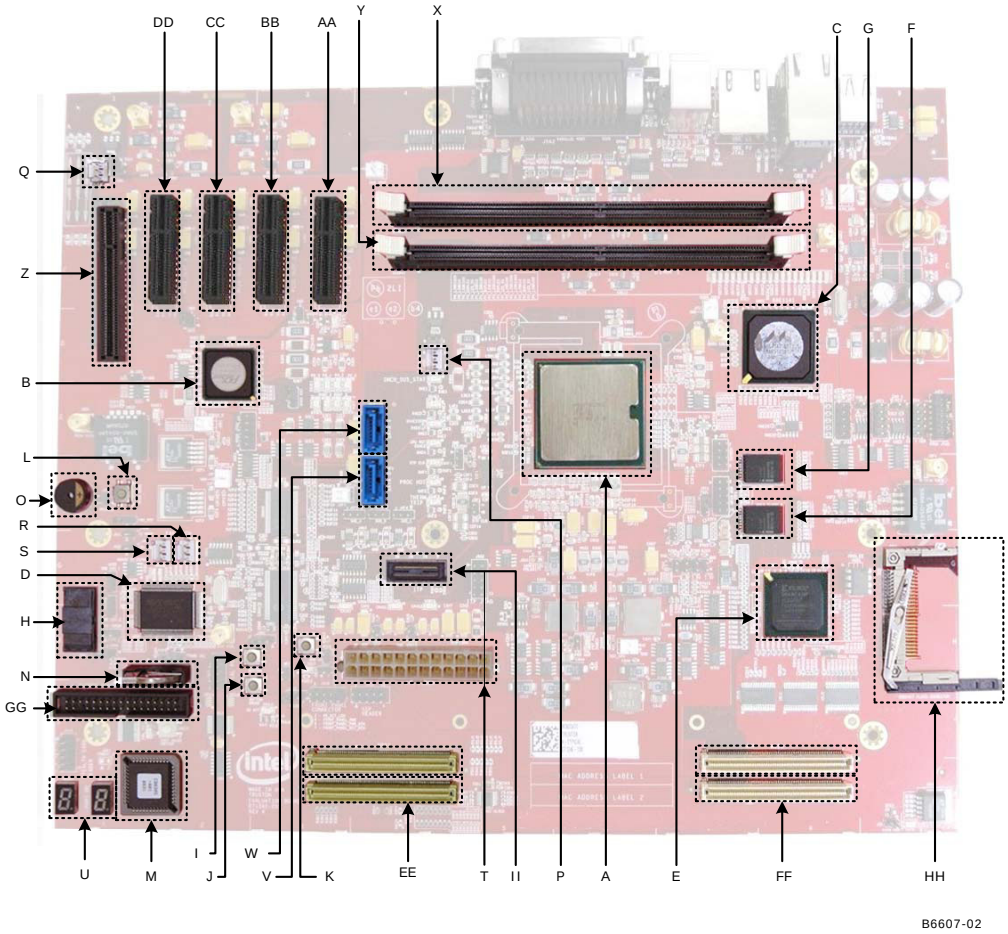


Figure 2. Development Board - Bottom View of Component and Connector Locations

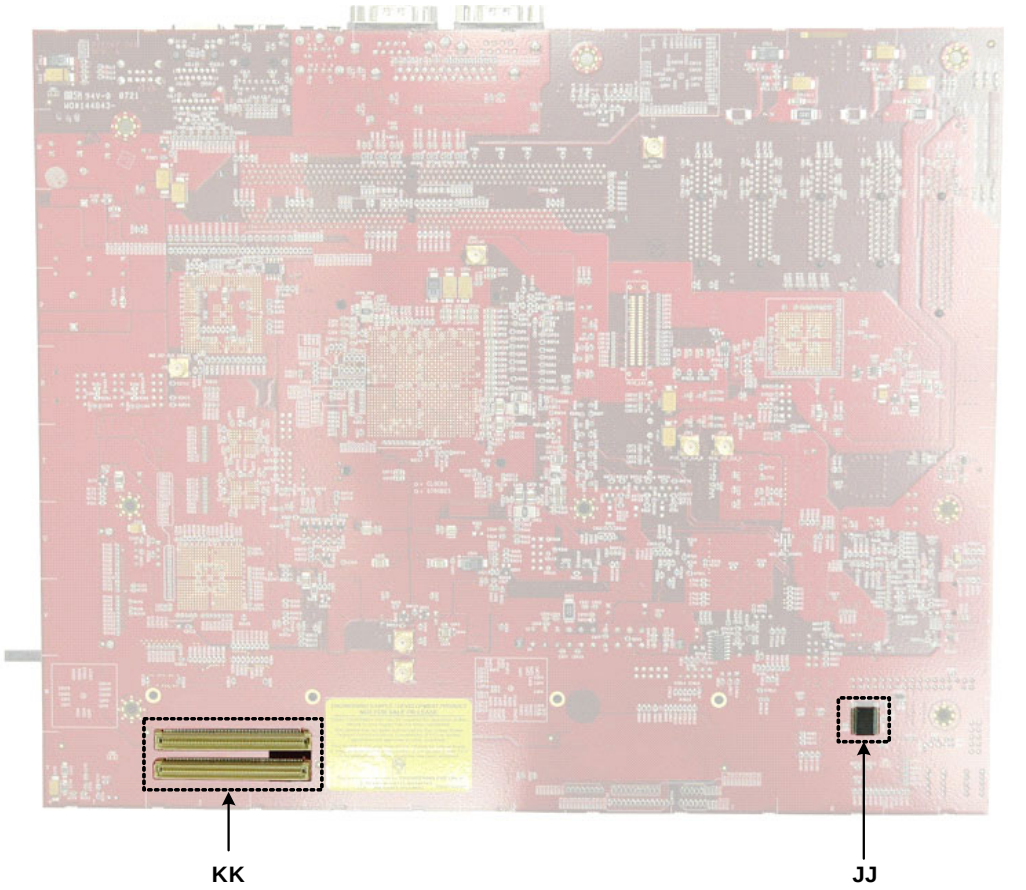
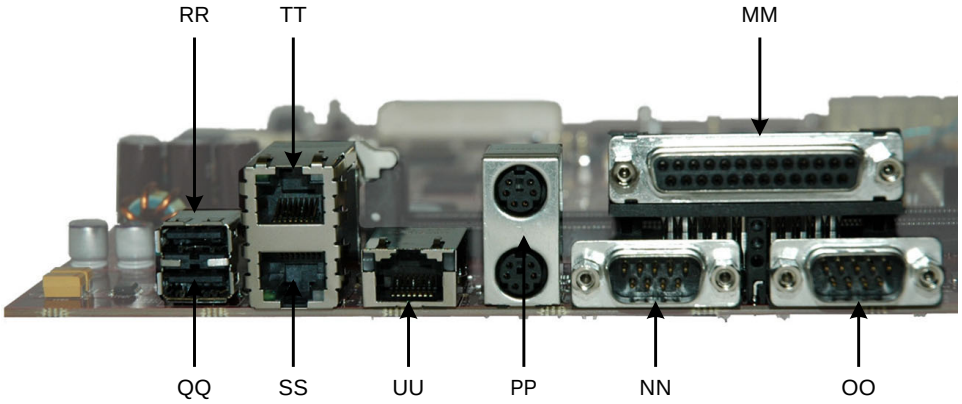


Figure 3. Development Board - Side View of the Board Connectors



B6605-01

Table 1. Development Board - Key Components and Connectors Legend (Sheet 1 of 2)

Callout	Component/Connector
A	Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology
B	PEX PCIe Switch Chip
C	Marvell* 88ME1141 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)

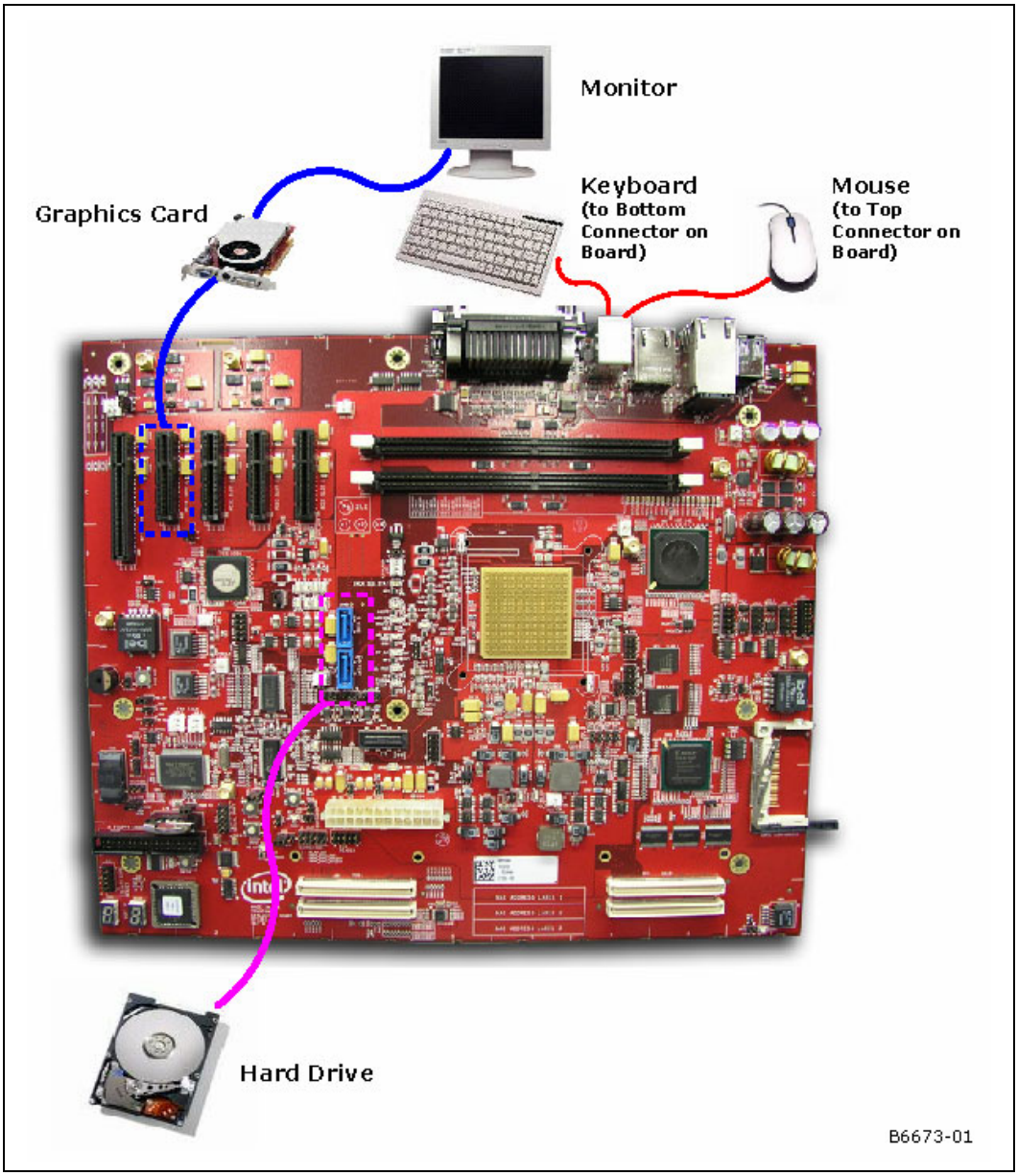
Table 1. Development Board - Key Components and Connectors Legend (Sheet 2 of 2)

Callout	Component/Connector
V	SATA port 0
W	SATA port 1
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 1
FF	Mezzanine connector 0
GG	Floppy Connector
HH	CF connector
II	ITP-XDP connector
JJ	Trusted Platform Module
KK	Mezzanine connector 2
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

2.2 Development Board Setup Requirements

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 Installing the OS on a Development Board

Installing the OS on a development board involves the installation of CentOS 5.2 from the CDs and the rebuilding of the kernel and module files after patching the kernel for the following:

- Built-in PCI device recognition

Note: System commands given in this chapter assume that the user is issuing commands from a bash shell. This is the default shell. Use the “echo $0” command to verify use of the bash shell or run “/bin/bash” to switch to the bash shell.

3.1 System Requirements

Please consult the CentOS minimum hardware requirements in the online Installation Guide at http://www.centos.org/docs/5/html/5.2/Installation_Guide. The development board meets these requirements.

Execute the following three steps:

- Get the latest BIOS image and install it as described in Section 8.2.1, “Aptio Flash Update Utility (AFUEFI)” on page 37
- Set the Legacy or AHCI SATA mode as described in Section 8.1.5, “Legacy and AHCI SATA Mode” on page 36
- Allocate the Coherent and Non-Coherent Memory as described in Section 8.1.4, “Coherent and Non-Coherent Memory Allocation” on page 36

Note: The SATA mode (Legacy or AHCI) cannot be changed after OS installation.

3.2 Acquiring CentOS 5.2 Linux

The software package has been validated with CentOS 5.2, and validated with GCC 4.1 and the Glibc 2.5 tool chain. The software package does not include a distribution of CentOS 5.2 or any Linux distribution. The package includes Linux device driver sources developed by Intel for EP80579 integrated processor features. The device driver source may be installed with a CentOS 5.2 distribution. Please acquire a source distribution of CentOS 5.2 Linux from http://centos.org.

3.3 Installing CentOS 5.2 Linux

For complete CentOS 5.2 installation instructions, please refer to the online Installation Guide at http://www.centos.org/docs/5/html/5.2/Installation_Guide. The following are some basic instructions. For the purposes of this Getting Started Guide, it is assumed that the installation is from CD images.

Warning: Decide to use the SATA drive in Legacy mode or AHCI (Advanced Host Controller Interface) mode prior to installing any Operating System. Once the SATA mode is set and the Operating System is installed, changing the SATA mode will prevent the Operating System from booting. Selecting to use SATA in AHCI mode will improve

performance of the system, provided the hard drive being used supports AHCI mode. Refer to Section 8.1.5, "Legacy and AHCI SATA Mode" on page 36 for instructions on selecting the desired SATA mode in the BIOS setup selection menu.

1. Prepare the system to boot from CDROM by using the boot selections in the BIOS Setup Menu. Details of the BIOS setup options are available in Chapter 8.0, "Pre-Boot Firmware (BIOS)."
2. Power on the system with disc #1 in the CDROM. The system should begin to boot from the CentOS installation disc #1.

   Note: It is recommended that all CentOS installation discs be verified at least once prior to an installation of the OS.
3. Upon booting from disc #1, the installation process prompts if the CD media should be tested.
4. Review the CentOS Release Notes during the installation if desired. Continue with the installation process whenever prompted with the "Next" button.
5. Select the language to use during the CentOS installation process.
6. Select the appropriate keyboard for the system.
7. Select the option to perform a fresh install of CentOS on the system.
8. The hard drive that comes with the development board is initially unformatted. Partition this hard drive.
9. Select the time zone in which the system is located.
10. Select the city nearest to the system's located.
11. Enter the desired root password for the system in the box labeled "Root Password:" and reaffirm the root password by entering the password in the box labelled "Confirm:".
12. Select "Customize Now" and see Section 3.3.1 for selection details. These are necessary to allow the compilation of the EP80579 security software drivers and the rebuilding of the Linux kernel to support the EP80579 integrated processor

3.3.1 Recommended Installation Customizations

1. Package customization is available through these selections:
   - Desktop Environments
   - Applications
   - Development
   - Servers
   - Base System
   - Virtualization
   - Clustering
   - Cluster Storage
   - Languages

   The recommended minimum package installation selections are given in the tips below. Additional detailed customization of each package can be accomplished by selecting a package (for example: Editor), and clicking the "Optional package" button to display the optional and default packages within the main package.

Tip: **Desktop Environments:** Optional. Install Gnome or KDE if you wish to use a Graphical User Interface (GUI).

Tip: **Applications:** Editor. Others are optional.

Tip: **Development:** Development Libraries and Development Tools should be installed. Others are optional.

Tip: **Servers:** Depending on the end use connectivity, none to all server types may apply. Some common selections are DNS Name Server, FTP Server, Legacy Network Server, Network Servers or Web Servers.

Tip: **Base System:** Administration Tools, Base and System Tools are recommended to be installed.

Tip: **Virtualization**: Optional.

Tip: **Clustering**: Optional.

Tip: **Cluster Storage**: Optional.

Tip: **Languages:** Additional language selections vary depending on user location.

2. Following package selection, click the “Next” button when prompted. After a dependency check, the installation process is ready to begin and prompts the user to have the required CentOS CDs available. To begin the installation, click the “Continue” button, click “Back” to change a package selection, or click “Reboot” to abort this installation.
3. If continuing with this installation, when prompted, insert the requested installation CD and click “Next”. When the installation is complete, and a prompt is displayed to reboot the system, remove the final CD from the CDROM and choose to reboot the system.
4. When rebooting, some normal system setup is required, such as setting the time and date. When asked, follow the prompt by clicking the “Forward” button.
5. Disable “Firewall”.
6. SELinux should be changed from the default “Enforcing” to “disabled”.
7. Enabling kdump is optional, but may help to diagnose the cause of any future system crash. Minimum memory allocated to kdump is 128 MB. If kdump is selected, a reboot is required.
8. Set the time and date.

   Note: Although you are creating additional users in the next step, please note that the instructions in this document should be executed as the root user.
9. Create additional users.
10. Finally, when prompted, press the “Finish” button.

3.4 Installing CentOS 5.2 Kernel Source

Note: Always login as the root user to carry out the instructions in this document.

1. Follow the instructions in the man page for the “date” command to set the current date. For example:

```
date 072910302008
```

to set the date to July 29th, 2008, 10:30AM.

Note: If the Time, Date and Timezone are not set on the development board, the build command used later in this procedure will go into an infinite loop.

2. Create a directory for file download, build and install activities. For example, create a directory called “EP805XX_release” with the following commands:

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

3. Create the ICP_ROOT environment variable:

```
export ICP_ROOT=$PWD
```

4. Check to see if unifdef rpm is installed using the command:

```
rpm -q unifdef
```

5. If the rpm is not installed, copy unifdef rpm from CentOS Installation Disk 3 to the $ICP_ROOT directory and install the rpm using the following commands.

   a. Mount the CD-ROM drive as follows:

   Note: This step is not required if CD is automounted.

```
mkdir /mnt/cdrom
mount -t auto /dev/cdrom /mnt/cdrom
```

   b. With OS Disk 3 inserted in the CD-ROM drive, issue the following command:

```
cp /mnt/cdrom/CentOS5.2/unifdef-1.171-5.fc6.i386.rpm .
```

   c. Install the rpm as follows:

```
rpm -ivh unifdef-1.171-5.fc6.i386.rpm
```

6. Download linux-2.6.28.10.tar.gz kernel source from:
   http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.28.10.tar.gz
7. Move the kernel sources tarball to kernel sources directory:

```
mv linux-2.6.28.10.tar.gz /usr/src/kernels
```

8. Change to the sources directory:

```
cd /usr/src/kernels
```

9. Extract the kernel sources:

```
tar -xvzf  linux-2.6.28.10.tar.gz
```

10. Create a soft link to the kernel sources:

```
ln -s linux-2.6.28.10 linux
```

Note: If new kernel sources are to be used, the symbolic link must be updated to point to the new kernel source directory.

11. Set the environment variable KERNEL_SOURCE_ROOT to the new kernel source directory:

```
export KERNEL_SOURCE_ROOT=/usr/src/kernels/linux
```

3.5 Unpacking the EP80587 Security Software Linux Package

The EP80579 security software 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, a recommendation is provided.

Transfer the gzip file to the EP80579 target system using any preferred method, for example, a USB flash drive, CDROM or network transfer. Unpack the tarball in the $ICP_ROOT directory using the following commands:

> ***Note:*** Without the option 'm' in the following tar command, the build might get into an infinite loop if the date is set incorrectly.

```
cd $ICP_ROOT
tar -xmvzf Security.L.1.0.3-xxx.tar.gz
```

A new directory structure is created under the $ICP_ROOT directory that contains all EP80579 security software for Linux. See Section 1.5.3, "Package Release Structure" on page 8 for details of the directory structure of the software release.

You will also see a file "filelist" that contains the list of files included in the release.

3.6 Updating PCI Device Names

PCI device names must be updated prior to driver installation. This can be done by performing the following steps:

1. Download the most recent PCI ID list from:
   http://pciids.sourceforge.net/pci.ids.bz2
2. Backup current pci.ids file

   ```
   mv /usr/share/hwdata/pci.ids /usr/share/hwdata/pci.ids.old
   ```
3. Extract the contents of the archive

   ```
   bunzip2 pci.ids.bz2
   ```
4. Copy the updated file into the /usr/share/hwdata directory

   ```
   mv pci.ids /usr/share/hwdata
   ```

The utility "lspci" can be used to verify the changes. The strings returned by the command will contain "Intel Corporation EP80579".

3.7 Rebuilding the Kernel

3.7.1 Restore the Old Configuration

Select all default response in the following commands:

```
cd $KERNEL_SOURCE_ROOT
make menuconfig
```

Once the menu is displayed, simply save and exit to create a .config file in the directory pointed by $KERNEL_SOURCE_ROOT.

3.7.2 Modifying the Makefile

Modify the Makefile file in the current directory as follows:

Edit the line "EXTRAVERSION = -prep" by editing "-prep" to a name meaningful to the purpose of recompiling the kernel. A simple suggestion is to change "-prep" to "-EP805XX".

Once the kernel is recompiled, this helps to indicate which vmlinuz kernel files and entries in the /etc/grub.conf file are related to the recompiled kernel.

3.7.3 Building and Installing the Patched Kernel Source

Note: This step will take about an hour and a half to complete.

Change to the kernel source directory and execute the following commands to recompile the Linux kernel:

```
cd $KERNEL_SOURCE_ROOT
make && make modules && make modules_install && make install
```

Note: The make commands will be executed only if the preceding make commands were successful.

Save the configuration of the new 2.6.28-EP805XX kernel in the /boot directory using the following command:

```
cp $KERNEL_SOURCE_ROOT/.config /boot/config-2.6.28-EP805XX
```

The new kernel is compiled and placed in the /boot directory. The new kernel is named vmlinuz-2.6.28-EP805XX. A symbolic link is created from vmlinuz -> vmlinuz-2.6.28-EP805XX.

3.7.4 Making the New Kernel Default

Also, the /boot/grub/grub.conf file is modified to include the possibility of booting to this newly compiled kernel. An entry is created for the new kernel:

```
title CentOS (2.6.28-EP805XX)
    root (hd0,0)
    kernel /vmlinuz-2.6.28-EP805XX ro root=/dev/VolGroup00/LogVol00 rhgb quiet
    initrd /initrd-2.6.28-EP805XX.img
```

1. The title may be changed to be more reflective of this kernel's purpose or name.
2. To boot to this kernel image by default, modify the /boot/grub/grub.conf file's "default" entry. Entries begin from zero (0) and are counted top-down. Edit "default=<kernel entry number>", where <kernel entry number> is the number entry for the new vmlinuz-2.6.28-EP805XX kernel.
3. Skip to step (5) if your platform uses less than 1GB of memory.
4. Add the " mem=832M" (without the quotes, use leading space to demarcate) to the end of the kernel line. A sample entry might look like the following, assuming 1 GB physical memory:

```
kernel /vmlinuz-2.6.28-EP805XX ro root=/dev/VolGroup00/LogVol00 rhgb mem=832M
```

Note: 832M = 1G - 32M(NCDRAM size) - 32M(CDRAM size) -128M).
32M = 0x2000 (BIOS setting for CDRAM and NCDRAM size) * 4K (page size)

Although we do not require 32M of CDRAM and NCDRAM for this release, all validations have been done using the above values. All features have been validated using these BIOS settings.

5. Save the changes.

3.7.5 Rebooting and Verifying

Rebooting the system reboots to the newly compiled and installed kernel.

Execute the following command to ensure that the correct kernel has been loaded:

```
uname -r
```

The output should look like:

```
2.6.28-EP805XX
```

4.0 Building EP80579 Security Software on a Target Development Board

This chapter provides instructions for unpacking the EP80579 security software package, setting up the environment and building/compilation instructions.

Note: System commands given in this chapter assume that the user is issuing commands from a bash shell. This is the default shell. Use the “echo $0” command to verify use of the bash shell or run “/bin/bash” to switch to the bash shell.

4.1 Environment Setup

After unpacking the release package (as described in Section 3.5), issue the following commands to restore the environment variables:

Note: All environment variables set for kernel rebuild are lost when the system was rebooted.

Note: As per the instruction in the previous chapters, the kernel source are untarred at KERNEL_SOURCE_ROOT and other sources are untarred at ICP_ROOT.

```
export ICP_ROOT=/EP805XX_release
export ICP_BUILDSYSTEM_PATH=$ICP_ROOT/build_system
export KERNEL_SOURCE_ROOT=/usr/src/kernels/linux
```

Follow the instructions in the man page for the “date” command to set the current date. For example,

```
date 102910302007
```

sets the date to October 29th, 2007, 10:30AM.

4.2 Build Options

Section 4.3, “Build Using Top Level Make” on page 25 describes the instructions for building Security and Embedded components together.

4.3 Build Using Top Level Make

It is required to set an ICP_BUILD_OUTPUT environment variable that defines where built objects are copied and where the sample load scripts can find the built objects.

For example, to store the resulting files in a directory called $ICP_ROOT/StagingArea, use the following commands:

```
mkdir $ICP_ROOT/StagingArea
export ICP_BUILD_OUTPUT=$ICP_ROOT/StagingArea
```

Caution: Please make sure that the current date has been set, otherwise the make process goes into an endless loop.

The following command builds the EP80579 security software and EP80579 embedded software drivers:

```
cd $ICP_ROOT
make clean
make
```

Note: The following warnings are displayed when icp_asd.ko module is built. These warnings do not limit/impact functionality in any way, as the error messages are related to symbols that are defined in other modules that are present in the system when icp_asd.ko is loaded and are resolved at load time.

```
WARNING: "icp_AsdCfgLacStop" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgHalStart" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "halAe_Init" [/EP805XX_release/Acceleration/drivers/icp_asd/src/kernel/
linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgHalInit" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgHalStop" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "UcLo_MapObjAddr" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgLacShutdown" [/EP805XX_release/Acceleration/drivers/icp_asd/
src/kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgLacStart" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "QatComms_intr" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "halAe_DelLib" [/EP805XX_release/Acceleration/drivers/icp_asd/src/kernel/
linux/icp_asd.ko] undefined!
WARNING: "UcLo_InitLib" [/EP805XX_release/Acceleration/drivers/icp_asd/src/kernel/
linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgQatalStop" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgQatalShutdown" [/EP805XX_release/Acceleration/drivers/icp_asd/
src/kernel/linux/icp_asd.ko] undefined!
WARNING: "QatComms_bh_schedule_register" [/EP805XX_release/Acceleration/drivers/
icp_asd/src/kernel/linux/icp_asd.ko] undefined!
WARNING: "UcLo_DeleObj" [/EP805XX_release/Acceleration/drivers/icp_asd/src/kernel/
linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgQatalInit" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "QatComms_bh_handler" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgHalShutdown" [/EP805XX_release/Acceleration/drivers/icp_asd/
src/kernel/linux/icp_asd.ko] undefined!
WARNING: "UcLo_WriteUimageAll" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgLacInit" [/EP805XX_release/Acceleration/drivers/icp_asd/src/
kernel/linux/icp_asd.ko] undefined!
WARNING: "icp_AsdCfgQatalStart" [/EP805XX_release/Acceleration/drivers/icp_asd/
src/kernel/linux/icp_asd.ko] undefined!
```

Table 2. Output Files Created in the $ICP_BUILD_OUTPUT Directory

File Name	Description
*.ko	Kernel modules
*.bin	ASU microcode files
icp_asd.conf	Installed in /etc and is the configuration file used by ASD
asd_ctl	User space utility that downloads the /etc/icp_asd.conf file to ASD during initialization
debugmgr	Proprietary debug command to get software version information and data dump
install_security.sh	Installation scripts
qat_service	Shell script used to Start and Stop kernel modules

4.4 Installation of Build Output

Execute the following command to install the Embedded and Accelerated kernel modules:

```
make install
```

Besides copying the kernel modules to a sub-directory under /lib, a startup script is added in the rcx.d directory to make the accelerated kernel modules persistent (that is, they are available for use after a system reset). The kernel modules are loaded to enable EP80579 embedded software drivers and EP80579 security software.

Follow the instructions in Chapter 5.0, "Runtime Configuration."

4.4.1 Statistics Collection

The Cryptographic API supports statistics retrieval and display for the individual symmetric and asymmetric components. Statistic collection requires the use of atomic operations which are expensive, therefore disabling this functionality results in significant performance gains.

Configuration is performed by including parameters when the icp_crypto kernel module is loaded. As such, the module must be unloaded and then re-loaded for these changes to take effect. Before unloading the icp_crypto module, other security modules must unloaded. The modules can be unloaded using the following instructions:

```
rmmod icp_asd
rmmod icp_crypto
```

See Section 4.4.1.4 for instructions on loading the modules again.

4.4.1.1 Enable / Disable All Statistics Collection

By default, all statistics collection is enabled when icp_crypto module is loaded. The parameter icp_crypto_statistics_master can be used to enable or disable all statistics collection. These commands must be executed in the directory where the icp_crypto kernel object is located.

To enable statistic collection:

```
# modprobe icp_crypto icp_crypto_statistics_master=all
```

When this command is issued, the module loads with all statistics collection enabled. This is the default setting, so it is not necessary to pass in icp_crypto_statistics_master as shown in the example above. The real value of the icp_crypto_statistics_master parameter is observed when used in conjunction with disabling specific statistics gathering as discussed in Section 4.4.1.2.

To disable all statistic collection:

```
# modprobe icp_crypto icp_crypto_statistics_master=none
```

When this command is issued, the module loads with all statistics collection disabled. All individual statistics settings discussed in Section 4.4.1.2 are ignored when this parameter is set to none.

4.4.1.2 Disabling Individual Statistics Gathering

As mentioned previously, it is possible to control which statistics are collected. When icp_crypto_statistics_master is set to all, all statistics are collected unless they are individually disabled. The following statistics can be individually disabled:

- icp_crypto_statistics_dsa
- icp_crypto_statistics_dh
- icp_crypto_statistics_ln
- icp_crypto_statistics_prime
- icp_crypto_statistics_rsa
- icp_crypto_statistics_key
- icp_crypto_statistics_cb
- icp_crypto_statistics_alg_chain
- icp_crypto_statistics_random
- icp_crypto_statistics_msgs

To disable specific statistics, set the parameter value to off when loading the icp_crypto module. For example, the following command will enable all statistics gathering except the dsa module:

```
# modprobe icp_crypto icp_crypto_statistics_master=all
icp_crypto_statistics_dsa=off
```

4.4.1.3 Verbose Output

The parameter icp_crypto_verbose can be turned on to inform the user of the statistics gathering options selected at load time. Remember if icp_crypto_statistics_master is set to all, all statistics collecting is enabled, unless otherwise noted under the individual statistics settings. Use the command “dmesg” to view the statistics gathering settings after loading icp_crypto kernel module.

```
# modprobe icp_crypto icp_crypto_verbose=1 icp_crypto_statistics_master=all
icp_crypto_statistics_dsa=off
# dmesg
    ...
    Loading LAC Module
    statistics_master = all --> All statistics ON
      Collecting statistics:
        statistics_master    = all
        statistics_dsa       = off
        statistics_dh        = on
        statistics_ln        = on
        statistics_prime     = on
```

```
statistics_rsa       = on
statistics_key       = on
statistics_cb        = on
statistics_alg_chain = on
statistics_random    = on
statistics_msgs      = on
```

4.4.1.4 Reloading Security Modules

After updating the statistics gathering options of the icp_crypto kernel module, it is necessary to re-load the other security modules. This can be accomplished by performing the following steps:

```
# modprobe icp_asd
# /etc/init.d/asd_ctl
```

4.5 Uninstalling Embedded and Security Kernel Modules

The following uninstall option is available to unload the kernel modules and uninstall the kernel modules from the relevant /lib sub-directory.

```
make uninstall
```

The user can load and unload the kernel modules without deleting them from the /lib sub-directory using the command:

```
/etc/init.d/qat_service stop
```

5.0 Runtime Configuration

A user working in an X-Windows environment may want to execute the following command to disable X-Windows and work in a command line environment:

```
init 3
```

Note: The user can switch amongst six screens by entering the key sequence <ALT><F1| F2| F3| F4| F5| F6>. The “init 5” command takes the user back into X-Windows mode.

5.1 Using the Intel® QuickAssist Technology Cryptographic API

At this point, the user can configure and use a Network crypto application such as VPN.

6.0 Building, Installing and Loading Individual EP80579 Embedded Software Drivers

Before building and loading EP80579 embedded software drivers, define the following environment variable:

```
export ICP_ROOT=/EP805XX
```

Note: As per the instruction in the previous chapters, the kernel source are untarred at KERNEL_SOURCE_ROOT and other sources are untarred at ICP_ROOT.

6.1 Global Configuration Unit and Gigabit Ethernet Drivers

Two drivers are required for enabling network connectivity on the Gigabit Ethernet controllers in the EP80579: 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.

6.1.1 Linux Compilation Instructions

All source files for the Linux release of the Global Configuration Unit (GCU) driver and the Gigabit Ethernet (GbE) driver are located in the following directory within the Linux compatible EP80579 security software release:

```
$ICP_ROOT/Embedded/src/GbE
```

Compilation of both the GCU and GbE drivers separately from the rest of the software package is possible. Enter the *$ICP_ROOT/Embedded/src/GbE* directory and execute the following commands:

```
make clean
make
```

The GCU and GbE drivers compile and the resulting gcu.ko and iegbe.ko files are placed in the *$ICP_ROOT/Embedded/src/GbE/build/linux_2.6/kernel_space* directory.

6.1.2 Linux Module Load/Unload Instructions

To load the Linux Global Configuration Unit driver, execute the following command from the directory where the compiled executable resides (*$ICP_ROOT/Embedded/src/GbE/build/linux_2.6/kernel_space*):

```
insmod gcu.ko
insmod iegbe.ko
```

To unload the Linux Global Configuration Unit driver, execute the following commands:

```
rmmod gcu.ko
rmmod iegbe.ko
```

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

```
lsmod | grep gcu
lsmod | grep iegbe
```

The output of the above commands list modules loaded in the system containing “gcu” and “iegbe” as part of their name.

6.1.3 Runtime Configuration of GCU and Gigabit Ethernet

Runtime configuration of network support is through traditional ifconfig commands, detailed in man pages installed on the system.

Note: In Linux, the Ethernet ports TT, SS and UU in Figure 3 on page 15 appears as eth0, eth1 and eth2 respectively.

7.0 Sample Applications

Sample applications are included with this software release that present examples of how to use to cryptographic APIs. They can be found in the directory: ./Acceleration/library/icp_crypto/look_aside_crypto/src/sample_code

The directory structure is as follows:

```
./Acceleration/library/icp_crypto/look_aside_crypto/src/sample_code

.../functional

.../functional/asym

.../functional/asym/diffie_hellman_sample

.../functional/asym/prime_sample

.../functional/common (This folder is FreeBSD-specific. Others occur in both OSes.)

.../functional/include

.../functional/sym

.../functional/sym/alg_chaining_sample

.../functional/sym/cipher_sample

.../functional/sym/hash_sample

.../performance
```

For instructions on using the sample applications, refer to README.txt located in: ./Acceleration/library/icp_crypto/look_aside_crypto/src/sample_code

8.0 Pre-Boot Firmware (BIOS)

The pre-boot firmware is executed when the system is powered up or reset. It initializes and configures system memory, devices and 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 Serial Peripheral Interface (SPI) Flash; the FWH or SPI Flash can be updated using a flash utility tool that is provided.

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 begins). The setup menu bar is shown in Table 3.

8.1 Pre-Boot Firmware Setup Menu

Table 3 shows the pre-boot firmware setup main menu and provides a brief description of each menu option. Table 4 shows the action keys that can be used when navigating and selecting options from pre-boot firmware menus.

Table 3. 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 4. Pre-Boot Firmware Setup Program Action 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 value for the current menu
F10	Save the current configuration and exit the setup menu
ESC	Exit the setup menu

8.1.1 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 development board pre-boot firmware, and the pre-boot firmware redirects the video to the console terminal.

As an option, the 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: The pre-boot firmware console redirection is text only. Graphical data, such as logos, are not redirected.

Table 5 shows the default settings of the serial console redirection.

Table 5. 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

8.1.2 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.

8.1.3 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 the North Bridge sub tab. Navigate down to the bottom option, titled Max Memory Speed Support, and 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.

8.1.4 Coherent and Non-Coherent Memory Allocation

The development board supports allocation of memory regions for coherent and non-coherent use. For more information on these regions, refer to the Intel® EP80579 Integrated Processor Product Line Datasheet, Section 3.0.

Coherent and non-coherent memory use features are for development boards that use the Intel® EP80579 Integrated Processor with Intel® QuickAssist Technology. EP80579 integrated processors **without** Intel® QuickAssist Technology do not make use of the memory set aside for these features.

This software package requires the pre-boot firmware (BIOS) for your hardware to allocate the values for each region.

To allocate memory regions, perform the following steps:

1. Enter the BIOS Setup Menu and select the Chipset tab.
2. 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 “2000” and press Enter.
3. 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.
   Type the numerical value “2000” and press Enter.

8.1.5 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.

8.2 Pre-Boot Firmware Image Reflashing Instructions

One method is available for updating the pre-boot firmware flash images located on the development board FWH or SPI Flash:

- 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.

8.2.1 Aptio Flash Update Utility (AFUEFI)

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

Necessary hardware:

- development board
- Socketed Firmware Hub or SPI Flash
- USB flash drive

Necessary software:

- development board pre-boot firmware image
- Aptio Flash Update Utility from AMI - AFUEFI

Steps to reflash the image:

1. Execute the AFUEFI utility onto the USB flash drive.
2. Load the pre-boot firmware image onto the USB flash drive.
3. Change the boot setting in the BIOS Setup Selection to boot from the EFI shell. Boot the development board to the EFI shell.
4. Insert the USB flash drive into the USB port.
5. Once the USB flash drive is recognized on the system (activity can be seen on the USB flash drive LED if present), 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. Enter:

   ```
   AFUEFI <pre-boot firmware image name> /X /P /B /N
   ```

   where the <pre-boot firmware image name> will be TRXTG063.ROM or similar
7. Reboot the development board when reflashing has completed.
8. Confirm that the image has been updated to the reflashed image by checking the Flash Image identity in the BIOS Setup.

9.0 Uninstalling the Software

Please refer to instructions on loading and unloading individual modules in Chapter 6.0.

9.1 Linux Module / Driver Dependencies

Table 6 lists the dependencies for the driver modules or patch within the EP80579 security software package. OS installation is assumed.

Table 6. Linux Module / Driver Dependencies

Module / Driver / Patch	Dependency 1	Dependency 2
EP80579 integrated processor Linux Patch	OS Source	None
icp_asd	icp_crypto	icp_debug
icp_crypto	icp_services	icp_debug
icp_debugmgmt	icp_debug	
Gigabit Ethernet (GbE)	GCU	None
Global Configuration Unit (GCU)	None	-

10.0 Troubleshooting

Refer to the Release Notes for your software package for a list of known errata, implications, and workarounds.

10.1 Using a Graphics Card

Some LCD monitors do not function properly with the development board. In one case, an “Input Not Supported” message was reported upon boot completion. Try an alternative LCD monitor if video is not displayed.

10.2 Using the Serial Port

The user can bring up the system without using the Graphics card. In this case, the IO takes place over the serial port. The serial port is configured as follows:

1. Enable communication over all serial ports in the BIOS setup. Configure ports to 115200, No Parity, 8 bit, 1 stop bit and No Flow Control.
2. Use a female-female 9-pin RS232 cable to connect the development board to a terminal emulator application such as HyperTerminal* in Windows. Configure the terminal emulator to 115200, No Parity, 8 bit, 1 stop bit and No Flow Control.

   Note: Changing from the default sio0 COM device (the one on the left when you face the platform) is complex and requires editing of /etc/make.conf file and rebuilding of bootloader and kernel.
3. Edit grub.conf:

   The bootloader enables Grub output on the serial port and console logging via the serial port.

   a. Make a backup of /etc/grub.conf before doing any changes.

   b. Edit that file and just underneath "timeout=5" add:

   ```
   serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
   terminal --timeout=0 serial console
   ```

   At the end of each “kernel” line append:

   ```
   console=tty0 console=ttyS0,115200
   ```

   c. If the user does not want a graphic card, they can boot without one, provided they modify the grub.conf file and comment out the line:

   ```
   splashimage=(hd0,0)/grub/splash.xpm.gz
   ```

   Otherwise, if no graphics card is available, the system will hang at boot.
4. Edit /etc/inittab (enables shell login from the serial port):

   a. Make a backup of /etc/inittab before making any changes and edit the file.

b. Just below the line “Run gettys in standard runlevels”, add the following line:

```
so:2345:respawn:/sbin/agetty -L -f /etc/issueserial 115200 ttyS0 v 115200
vt100-nav
```

5. You may have to edit /etc/securetty and add the lines:

```
ttyS0
ttyS1
```

6. Remove the graphics card and restart the system.

11.0 Glossary

AHCI	Advanced Host Controller Interface
ASD	Acceleration System Driver
ASU	Acceleration Services Unit
CDRAM	Coherent DRAM
DIMM	Dual Inline Memory Module
EFI	Extensible Firmware Interface
FWH	Firmware Hub
GbE	Gigabit Ethernet
GCU	Global Configuration Unit
IEEE	Institute of Electrical and Electronics Engineers
IHS	Integrated Heat Spreader
NCDRAM	Non-Coherent DRAM
POST	Power On Self Test
RNG	Random Number Generator
SATA	Serial Advanced Technology Attachment
SKU	Stock Keeping Unit
TIM	Thermal Interface Material

§ §