

BSP for Microsoft* Windows* Embedded Compact 7 (WEC7) Supporting Intel Platforms

*Based on the Intel® Atom™ Processor N2800/N2600/D2550
and Intel NM10 Express Chipset*

User Guide

August 2012

*Revision 1.0
Software Release version: 3.2 Gold*

Intel Confidential



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.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

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 go to: <http://www.intel.com/design/literature.htm> 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 Intel product 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.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2012, Intel Corporation. All rights reserved.



Contents

1	Introduction	5
1.1	What's New	5
1.1.1	Contents	5
1.1.2	Features supported	5
1.2	Prerequisites	6
2	Installing the Software	7
3	Creating a Project	8
4	Enabling the Intel® Embedded Media and Graphics Driver	9
5	Enabling the Intel® Ethernet Driver	11
6	Catalog Item Selection	12
7	Build Option Configuration	13
8	Building the Software Release	14
8.1	Loading the Image (nk.bin) on the BIOS	16
8.1.1	Hardware Prerequisites	16
8.1.2	BOOT BIOS Configuration	16
8.1.3	BOOT Device Setup (for USB Thumb Drive)	16
8.1.4	Loading the 'nk.bin' Image	17
8.1.5	Power Shutdown feature	17
8.2	Loading the Image on BIOS through the KITL connection	17
8.2.1	Loading nk.bin via eboot.bin over KITL connection	18
9	Loading the Image for Intel® BLDK	20
9.1.1	Hardware Prerequisites	20
9.1.2	Intel BLDK Firmware Configuration	20
9.1.3	BOOT Device	20
9.1.4	Booting the 'nk.bin' Image	20
9.1.5	Power Shutdown feature	21
10	Known Constraints	22
11	Booting-up WEC7 on Intel BLDK Firmware	23
11.1	Overview	23
11.1.1	OS Loader	23
11.1.2	BLDK-compatible WEC7 OS	23
11.2	Building OS Loader	24
11.3	Building BLDK Firmware	26
11.4	Building BLDK-compatible Intel Atom WEC7 OS	26
11.5	Requirements of AML Byte Code	26



Revision History

Document Number	Revision Number	Description	Revision Date
490123	0.5	Initial release for Alpha.	December 2011
490123	0.6	Updated from validation review.	December 2011
490123	0.6	Updated from BSP3.2 Alpha	March 2012
490123	0.7	Updated from validation review.	June 2012
490123	1.0	Updated for Gold Release	August 2012

§



1 Introduction

This document provides important information for installing Intel's Board Support Package (BSP) for the Windows* Embedded Compact 7 (INTEL_CS), Release version: 3.2 Gold.

1.1 What's New

1.1.1 Contents

This production release contains the platform specific software for the following boards (formerly Cedar Trail):

- Intel® Atom™ Processor N2800 with Intel NM10 chipset
- Intel® Atom™ Processor N2600 with Intel NM10 chipset
- Intel® Atom™ Processor D2550 with Intel NM10 chipset

1.1.2 Features supported

The major features supported by this BSP on the Intel® Atom™ Processor N2800/N2600/D2550 with Intel NM10 chipset are:

1. Intel® High Definition Audio driver:
 - a. 2x audio playback outputs at Port-A and Port-D
 - b. 1x audio recording input at Port-F
2. ATAPI driver for 2x SATA ports.
 - a. SATA AHCI mode is NOT supported.
3. USB1.0 UHCI & USB2.0 EHCI drivers for 4x on-board USB ports
4. KITL Ethernet Debug Driver for on-board Intel Ethernet PHY 82552v
5. KITL Ethernet Debug Driver for PCIe* Intel/Pro 1000 PT Desktop Ethernet Adapter
6. Legacy Serial COM port
7. Symmetric Multi-Processing (SMP)
8. Power Shutdown



1.2 Prerequisites

1. Microsoft Windows* Embedded Compact 7 Platform Builder* must be installed. This creates a WINCE700 base directory on the default hard drive (for example, the following path should exist on C: drive: "C:\WINCE700". If a WINCE700 base directory does not exist, the installation will fail.
2. Intel® Atom™ Processor N2800/N2600/D2550 chipset with NM10 platforms (formerly Cedar Trail) with AMI* BIOS Version: ACTCRB1.86C.076.A00.
3. Windows* Embedded Compact 7.0 Board Support Package (BSP) version: Intel_WEC7_BSP_3.2.0_Gold.zip.



2 Installing the Software

1. Obtain the Intel_WEC7_BSP_3.2.0_Gold.zip BSP. Extract the contents of this zipped archive into a temporary directory.
2. Double click the Intel_WEC7_BSP_3.2.0_Gold.msi to install the BSP. Press 'Next' button to proceed with the installation. This installs to the default path "c:\WINCE700\PLATFORM" and creates a new directory called "INTEL_CS" along with its subdirectories.
3. To confirm that the installation is complete, verify the existence of the "INTEL_CS" directory on the path: "c:\WINCE700\PLATFORM\INTEL_CS" and the items listed above in the "What's New" section are on the following paths:

- C:\WINCE700\PLATFORM\INTEL_CS\SRC\BOOTLOADER\EBOOT
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\BOOTLOADER\EBOOT\E100EBOOT\BIN
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\ETHDRV\E1000ETHDBG
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\ETHDRV\PCHEG20TGBDBG
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\INC
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\IO
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\IOCTL
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\KITL
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\MPSUPPORT
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\MEMORY
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\POWER
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\COMMON\PDQOAL\STARTUP
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\ICHHDA
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\BLOCK\ATAPI
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\PM
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\PCH_EG20T_DRIVERS
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\WECE100NDIS
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\KITL
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\OAL\
- C:\WINCE700\PLATFORM\INTEL_CS\SRC\UEFIBOOTLOADER\WEC7OSLoader



3 *Creating a Project*

1. Run Microsoft* Visual Studio* 2008
2. Go to File → New Project
3. Select project type 'Platform Builder'
4. Select 'OS Design', under 'Visual Studio Installed templates'. By default OSDesign1 will be assigned as the project name.
5. Click OK
6. After the design wizard pops up, click Next. This will take you to the 'Board Support Packages' selection window.
7. Select 'INTEL_CS: x86', then click Next
8. At 'Design Templates', click Next
9. At 'Applications and Media', click Next
10. At 'Networking and Communications', click Next
11. At 'OS Design Project Wizard Complete', click Finish. The platform builder will generate project files for your project. Acknowledge any warnings that may pop-up from Visual Studio. The default project name for your generated project will be 'OSDesign1' under `C:\WINCE700\OSDesigns\OSDesign1`.



4 Enabling the Intel® Embedded Media and Graphics Driver

This section lists the steps for enabling the Intel® Embedded Media and Graphics Driver (Intel® EMGD) for boards featuring the Intel® Atom™ Processor N2800/N2600/D2550:

1. The EMGD 1.12 release installation file used for Gold is named:
"EMGD_1_12_CDV_RC_3040_Extract.exe"
2. Double-click on "EMGD_1_12_CDV_RC_3040_Extract.exe" to install EMGD files. Click 'Continue' to proceed with the installation. This will install to the default path "C:\IEMGD\IEMGD_1_12".
3. From the IEMGD_1_12 folder, locate the 'IEMGD_HEAD_WEC7.zip' file at the following path:
"C:\IEMGD\IEMGD_1_12\plugins\Pre_Packaged.Drivers_1.12.0\WEC7"
4. Extract the contents of the IEMGD_HEAD_WEC7.zip. A folder "IEMGD_HEAD_WEC7" will be created in the directory after the zip file has been extracted. Locate a folder named "Driver" inside "IEMGD_HEAD_WEC7" and copy it inside "C:\WINCE700\platform\INTEL_CS\SRC\DRIVERS". Rename the folder from "Driver" to "EMGD".
5. Go to 'C:\WINCE700\platform\INTEL_CS', open "Intel_CS.bat" and append the flag "set WEC7_EMGD_DRIVER=1" at the end of the file. Also unset BSP_DISPLAY_FLAT by modifying it to look like: "set BSP_DISPLAY_FLAT=".
6. Go to the "C:\WINCE700\PLATFORM\INTEL_CS\FILES" directory. Open the "platform.reg" file and append the following registry entry to the end of the file:

```
IF WEC7_EMGD_DRIVER
[HKEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]
"Candidate3"="Drivers\\Display\\Intel"
; include the path to the emgd.reg file in the release package
#include $(PLATFORM_DRIVERS_DIR)\EMGD\emgd.reg
ENDIF WEC7_EMGD_DRIVER
```

7. Ensure that the file is saved.



8. Go to the "c:\WINCE700\PLATFORM\INTEL_CS\FILES" directory. Open the "platform.bib" file and append the following to TOP of the file before the copyright statement:

```
#define PLATFORM_DRIVERS_DIR $(_TARGETPLATROOT)\src\drivers
```

9. In the "platform.bib" file, append the following to the end of the file:

```
IF WEC7_EMGD_DRIVER
analog.dll      $(PLATFORM_DRIVERS_DIR)\EMGD\analog.dll  NK SHK
ddi_emgd.dll    $(PLATFORM_DRIVERS_DIR)\EMGD\ddi_emgd.dll  NK SHK
hdmi.dll        $(PLATFORM_DRIVERS_DIR)\EMGD\hdmi.dll      NK SHK
lvds.dll        $(PLATFORM_DRIVERS_DIR)\EMGD\lvds.dll    NK SHK
dp.dll          $(PLATFORM_DRIVERS_DIR)\EMGD\dp.dll      NK SHK
ENDIF WEC7_EMGD_DRIVER
```

10. Ensure that the file is saved.
11. For further details regarding EMGD Driver, see EMGD's User Guide located in: "c:\IEMGD\IEMGD_1_12\ documentation".



5 Enabling the Intel® Ethernet Driver

This section lists the steps for enabling the Intel® Ethernet Driver for boards featuring the Intel® Atom™ Processor N2800/N2600/D2550:

1. Go to "C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\WECE100NDIS", and copy "E100WEC7.dll" and "E100WEC7.rel" to "C:\WINCE700\platform\INTEL_CS\FILES" directory. NOTE: These are the Ethernet driver binary files for Intel Ethernet 82552v on Intel Atom Processor N2800/N2600/D2550 boards
2. Go to "C:\WINCE700\PLATFORM\INTEL_CS" directory. Open the "INTEL_CS.bat" file and append the following flag at the end of the file:

```
set CEPB_INTELFE_PCI=1
set STATIC_IP=1
```

3. Ensure that the file is saved.
4. Go to "C:\WINCE700\PLATFORM\INTEL_CS\FILES" directory. Open the "platform.reg" file and append the registry entry (NOTE: Copy the registry entry from "C:\WINCE700\PLATFORM\INTEL_CS\SRC\DRIVERS\WECE100NDIS\E100WEC7.reg" file) at the end of the file.
5. Ensure that the file is saved.
6. Go to "C:\WINCE700\PLATFORM\INTEL_CS\FILES" directory. Open the "platform.bib" file and append the following at the end of the file:

```
IF CEPB_INTELFE_PCI
  e100wec7.dll $(_FLATRELEASEDIR)\e100wec7.dll NK SHK
ENDIF CEPB_INTELFE_PCI
```

7. Ensure that the file is saved.



6 Catalog Item Selection

Select the following catalog items in Visual Studio Catalog Items View before you build the project image.

Feature	Catalog Item Path
Active Sync	Core OS\Windows Embedded Compact\Applications-End User\ActiveSync
USB Mass Storage	Core OS\Windows Embedded Compact\Device Drivers\USB\USB Host\USB Class Drivers\USB Storage Class Driver
ATAPI (SATA & PATA)	Core OS\Windows Embedded Compact\Device Drivers\Storage Devices\ATAPI PCI Support Core OS\Windows Embedded Compact\Device Drivers\Storage Devices\ATAPI PCI Support\Basic ATAPI PCI CD/DVD-ROM Support
USB Keyboard & Mouse	Core OS\Windows Embedded Compact\Device Drivers\USB\USB Host\USB Class Drivers\USB Human Input Device (HID) Class Core OS\Windows Embedded Compact\Device Drivers\USB\USB Host\USB Class Drivers\USB Human Input Device (HID) Class\USB HID Keyboard and Mouse
Storage Manager Control Panel Applet	Core OS\Windows Embedded Compact\File Systems and Data Store\Storage Manager\Storage Manager Control Panel Applet
Mouse Cursor	Core OS\Windows Embedded Compact\Shell and User Interface\User Interface\Mouse
Audio Codec	Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media\Audio Codecs and Renderers\MP3 Codec Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media\Audio Codecs and Renderers\WMA Codec Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media\Audio Codecs and Renderers\Waveform Audio Renderer Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media\Audio Codecs and Renderers\Wave/AIFF/au/snd File Parser Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media\Audio Codecs and Renderers\MPEG-1 Layer 1 and 2 Audio Codec For more audio codec types, please refer to MSDN.
Windows Media Player	Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media\WMA and MP3 Local Playback Core OS\Windows Embedded Compact\Graphics and Multimedia Technologies\Media \Windows Media Player\Windows Media Player Applications\Windows Music Player For Windows Media Player options, refer to MSDN.



7 Build Option Configuration

From the Solution Explorer Tab, right click on the OSDesign project and click on 'Property'. Then, choose 'Configuration Properties\Build Options'.

Build Options	Description
Enable KITL (IMGNOKITL)	Please check if the system requires KITL transport connection to be established when image is downloaded through eboot.bin. Recommended to disable KITL if using local storage for boot process. Caution: If KITL is enabled and image is booted from local storage, the image may seem to hang or take much longer to boot.
Enable SMP Support in the kernel (IMGMPENABLE)	Please enable this build option for the Intel® Atom™ Processor N2800/N2600/D2550 because they support hyper-threading and multi-core.



8 Building the Software Release

1. Complete all the steps listed in 'Creating a Project'.
2. Under the Solution Explorer tab, right click on the project name (default name is OSDesign1) and select 'Properties'.
3. Change the 'Configuration' to 'INTEL_CS x86 Release'.
4. Click on Configuration Manager.
5. Make sure 'Intel_CS x86 Release' is selected in the 'Active solution configuration' field.
6. Make sure 'Intel_CS x86 Release' is the selected under 'Project contexts'. Ensure the check box under the 'Build' column is checked.
7. Close the project configuration windows.
8. To boot the image on the Intel Atom processor N2800/N2600/D2550 with Intel NM10 chipset CedarTrail platform, make sure the configuration in the Intel_CS.bat file in `C:\WINCE700\PLATFORM\INTEL_CS` is as follows:
"set BSP_NM10=1" is set and "set BSP_PCH_EG20T=" , "set BSP_ICH_82801HM=", "set BSP_ML7213=" are cleared under INTEL_CS.bat. The default configuration is "set BSP_NM10=1"
9. For the BLDK firmware, please make sure "set BSP_UEFI=1" within the BSP_NM10 batch option block.
10. The project is now ready to commence the build. From Microsoft Visual Studio 2008, go to the 'Build' Menu and click 'Build Solution'. The build process typically takes 1 hour to complete. The actual duration of the build completion depends on the specific hardware of the build machine, software and platform configurations chosen. Different stages of building and their progress can be viewed in the 'Output' window.



11. The build process completes with a sequence of messages shown in "Output" window similar to the following:

```

MAKEIMG: BUILDMSG: Calling
C:\WINCE700\OSDesigns\OSDESIGN1\OSDESIGN1\RelDir\INTEL_CS_x86_Release\
PostRomImage.bat
MAKEIMG: BUILDMSG: Calling
C:\WINCE700\OSDesigns\OSDESIGN1\OSDESIGN1\RelDir\INTEL_CS_x86_Release\
PostMakeImg.bat
MAKEIMG: BUILDMSG: Calling
C:\WINCE700\public\common\oak\misc\pbpostmakeimg.bat
MAKEIMG: BUILDMSG: Make Image process completed successfully
BLDDemo: BUILDMSG: OS image exists at
C:\WINCE700\OSDesigns\OSDESIGN1\OSDESIGN1\RelDir\INTEL_CS_x86_Release\
nk.bin
BLDDemo: BUILDMSG: Volume in drive C is OSDisk
BLDDemo: BUILDMSG: Volume Serial Number is B820-910F
BLDDemo: BUILDMSG: Directory of
C:\WINCE700\OSDesigns\OSDESIGN1\OSDESIGN1\RelDir\INTEL_CS_x86_Release
BLDDemo: BUILDMSG: 02/14/2011 04:28 PM          37,686,963 NK.bin
BLDDemo: BUILDMSG:                1 File(s)      37,686,963 bytes
BLDDemo: BUILDMSG:                0 Dir(s)  52,106,936,320 bytes free
BLDDemo: BUILDMSG: cebase build complete.
BLDDemo: BUILDMSG: BldDemo ended at 16:29:01.69 on Mon 02/14/2011
(exit code 0)
BuildLogs: BUILDMSG: Exiting: BldDemo1.bat -c -qbsp (result code 0).
BuildLogs: BUILDMSG: C:\WINCE700\build.log
BuildLogs: BUILDMSG: C:\WINCE700\build.out
BuildLogs: BUILDMSG: C:\WINCE700\build.wrn

```

Note:

You will see some warnings during the build process and in the eventual output, but these can be considered harmless.

Once the build is successful, the kernel image nk.bin can be found in the following path:

```
C:\WINCE700\OSDesigns\OSDesign1\OSDesign1\RelDir\INTEL_CS_x86_Release
```



8.1 Loading the Image (nk.bin) on the BIOS

This section describes the process for loading the image (nk.bin) on the BIOS for the Intel® Atom™ Processor N2800/N2600/D2550 with Intel NM10 chipset (formerly Cedar Trail) platform.

8.1.1 Hardware Prerequisites

1. USB Keyboard
2. USB Mouse
3. USB mass storage device (e.g., USB thumb drive)
4. SATA Hard Disk. [optional]
5. 1x stereo audio speaker sets [Optional]
6. 1x mono microphone. [Optional]

8.1.2 BOOT BIOS Configuration

The validated AMI BIOS OEM Version: **ACTCRB1.86C.073.A00**

The BOOT settings also need to be configured appropriately for booting up the hardware platform through the relevant boot device. After entering the BIOS, follow the instructions below to configure:

1. Go to Boot Menu.
2. Go into Boot Option Priority.
3. Check to see if your boot device is in the list (make sure your boot device has been plugged in).

8.1.3 BOOT Device Setup (for USB Thumb Drive)

The following instructions need to be followed in order to ensure that your booting device is enabled properly to bootup the hardware platform. This example assumes you are using a USB thumb drive as your booting device.

1. Ensure your USB thumb drive is DOS bootable formatted
2. The USB thumb drive can be plugged into the any of the USB physical port on the hardware platform.
3. Files required to load WEC7 kernel image (nk.bin) include: himem.sys, config.sys (to load himem.sys), loadcepc.exe. The two DOS files can be generated by MakeImageDisk.exe located on the path:
C:\Program Files (x86)\Microsoft Platform
Builder\7.00\cepb\utilities.
4. Copy your compiled kernel image nk.bin into the USB thumb drive.



8.1.4 Loading the 'nk.bin' Image

1. Boot up platform, make sure your bootable device has been properly installed.
2. Once the DOS is loaded, type "loadcepc /L:800x600x16 nk.bin". This loads up the WEC7 nk.bin on the CedarTrail platform. Use "loadcepc nk.bin" for images that have been built with EMGD.

8.1.5 Power Shutdown feature

1. Boot up platform, make sure WEC7 OS is booted successfully.
2. Double click on the ShutdownApp.exe which launches the power shutdown application.
3. Click on the red shutdown button which eventually shuts down the whole system.
4. The ShutdownApp.exe will be provided in binary format on as-needed basis when requested by customer.

8.2 Loading the Image on BIOS through the KITL connection

This section describes the process for loading the image on the BIOS through the KITL connection on the Intel® Atom™ Processor N2800/N2600/D2550 with Intel NM10 chipset (formerly Cedar Trail) platform.

During the development stage, the nk.bin can be transferred between Microsoft Platform Builder and the target platform via the Ethernet-based KITL connection. There are two options in the Intel® Atom™ Processor N2000 and D2000 Series Based Platform (formerly Cedar Trail) for Ethernet-based KITL:

- On-board Intel Ethernet PHY 82552V, **OR**
- Intel Pro/1000 PT Desktop Adapter

To build an eboot.bin image that can support the Intel Pro/1000 PT Desktop Adapter, make sure the following build setting is enabled in the INTEL_CS.bat file:

```
set BSP_KITL_POLL_MODE=1
set BSP_KITL_INTELGBE_E1000=1
```

A pre-build eboot.bin image and kitl.dll that support On-board Intel Ethernet PHY 82552v are included in the BSP. User can find the image under the location INTEL_CS/SRC/BOOTLOADER/EBOOT/E100EBOOT/BIN. Refer to the kitl.txt file for details.

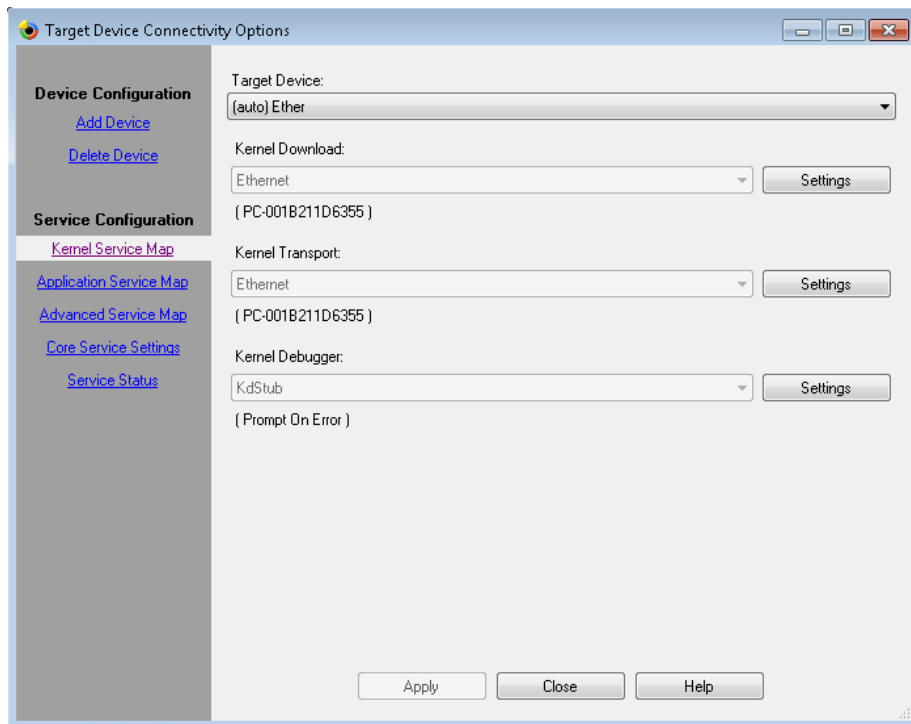
Note: The eboot.bin is available in c:\WINCE700\OSDesigns\OSDesign1\OSDesign1\RelDir\INTEL_CS_x86_Release\ when the BSP is built.

Note: Remember to enable KITL (IMGNOKITL=) on the BSP's project build option as explained above in order to build an nk.bin image that starts the KITL connection by default.

For details on loading image eboot, refer to: <http://msdn.microsoft.com/en-us/library/ms930104.aspx>

8.2.1 Loading nk.bin via eboot.bin over KITL connection

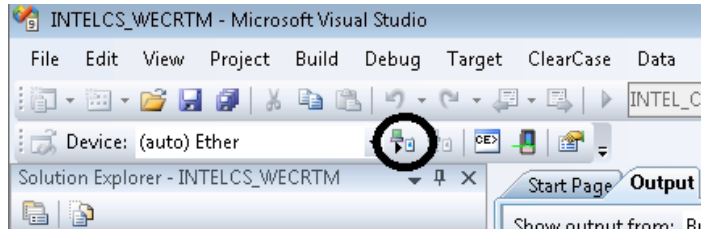
1. Copy eboot.bin onto the MS-DOS formatted USB mass storage with loadcepc.exe.
2. Connect the development PC to target platform via Ethernet cable over Gigabit switch.
3. Configure the development PC Ethernet card with local IP address & mask e.g. 172.30.179.1/255.255.255.0
4. On the command prompt of the target platform, type the following "C:\loadcepc /e:0:0:172.30.179.2 /L:800x600x16 eboot.bin" for the Intel® Atom™ Processor N2000 and D2000 Series Based Platform (formerly Cedar Trail).
5. Once the eboot.bin is loaded, the target platform will send a BOOTME message to Microsoft Platform Builder.
6. From Microsoft Platform Builder, click the menu "Target\Connectivity Options" and make sure the configurations are as follows:



Note: The target platform is chosen from the "Settings" button next to "Kernel Download: Ethernet option".



7. Once the "Target Device Connectivity Options" window is properly configured, click "Apply" or "Close".
8. Next, to attach the target platform, click the "Attach Device" button as highlighted below.



9. Finally, you will see the nk.bin image is downloaded over the Ethernet link and the platform is then booted up with KITL messages displayed at the "output" window of the Microsoft Visual Studio.



9 Loading the Image for Intel® BLDK

This section describes how to load the image in Intel® Atom™ Processor N2000 and D2000 Series Based Platform (formerly Cedar Trail) for Intel® BLDK.

9.1.1 Hardware Prerequisites

1. USB Keyboard
2. USB Mouse
3. USB mass storage device (e.g., USB thumb drive)
4. SATA Hard Disk. [optional]
5. 1x stereo audio speaker sets [Optional]
6. 1x mono microphone. [Optional]

9.1.2 Intel BLDK Firmware Configuration

The validated Intel BLDK version: To be added

9.1.3 BOOT Device

Intel BLDK for Cedar View can support booting WEC7 from USB mass storage, SATA hard disk and SATA solid state disk. These boot medium must be formatted under FAT32 file system. Please make sure the following files are located inside the boot medium in following directory path:

- efi\boot\BootIA32.efi [BLDK OS Loader]
- efi\boot\NK.bin [BLDK-built WEC7 image version]

BSP3.2 GOLD release is bundled with a pre-built UEFI OS Loader for WEC7 "BootIA32.efi" located in INTEL_CS\SRC\UEFIBOOTLOADER\BIN.

Please make sure NK.bin is created by setting BSP_UEFI=1 in Intel_CS.bat as described in section 8 of this document. This is the BLDK-built version.

9.1.4 Booting the 'nk.bin' Image

To boot-up the BLDK-built WEC7 image, simply insert the boot device on the target platform and power-on the target platform.

[Section 11](#) provides an overview of WEC7 boot-up path on BLDK firmware.



9.1.5 Power Shutdown feature

1. Boot up platform, make sure WEC7 OS is booted successfully.
2. Double click on the ShutdownApp.exe which launches the power shutdown application.
3. Click on red shutdown button which eventually shuts down the whole system.
4. The ShutdownApp.exe will be provided in binary format on as-needed basis when requested by customer.



10 Known Constraints

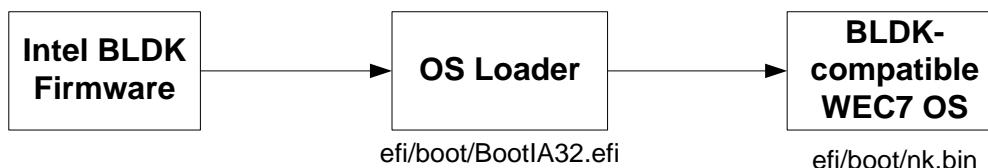
1. To enhance user experience in playing media files that contain mono-channel, the audio driver is designed to playback this type of media file in stereo speaker.
2. This BSP has been only tested on the specific Intel hardware boards below:
 - a. Intel® Atom™ Processor N2800 with Intel NM10 chipset based platform
 - b. Intel® Atom™ Processor N2600 with Intel NM10 chipset based platform
 - c. Intel® Atom™ Processor D2550 with Intel NM10 chipset based platform
3. Only polling mode KITL is supported.
4. For Intel® BLDK firmware for Intel® Atom™ processor N2800/N2600/D2550 + NM10 chipset based platform, the capability of ACPI parser implemented in Intel WEC7 BSP is limited to specific ASL descriptive language style. In current release, the ACPI parser does not support all possible ASL description as provisioned by ACPI specification. For details, refer to [Section 11](#).
5. On rare occasions, the UHCI#3 may fail to work. In order to prevent the issue, please make sure the following configuration is enabled:
 - a. From
C:\WINCE700\public\COMMON\oak\drivers\usb\hcd\uhc\uhci.reg,
- change the "order"=dword=1 to "order"=dword:2

And
 - b. From
C:\WINCE700\public\COMMON\oak\drivers\usb\hcd\usb20\ehcipdd\ehcd.reg
- change the "order"=dword:2 to "order"=dword:1



11 Booting-up WEC7 on Intel BLDK Firmware

11.1 Overview



The diagram above illustrates the WEC7 boot-up sequence on Intel BLDK firmware.

For the Intel® Atom™ Processor BSP3.2, the feature is currently available for Intel® Atom™ Processor N2000 and D2000 Series Based Platform (formerly Cedar Trail) only.

11.1.1 OS Loader

OS Loader is loaded by Intel BLDK firmware and is responsible to load WEC7 image from boot-medium to system memory. OS Loader also prepares the environment for booting up WEC7 image such as filling up boot parameters (RAM Top, ACPI RSDP address, display configuration and etc).

The source code for this OS Loader is located in Intel_CS/SRC/UEFIBOOTLOADER.

The pre-built binary of the BootIA32.efi is located in Intel_CS/SRC/UEFIBOOTLOADER/BIN.

11.1.2 BLDK-compatible WEC7 OS

BLDK-compatible Intel Atom WEC7 kernel is a special-built WEC7 image that performs the following additional process during kernel initialization:

- Parse AML byte code stream (published by BLDK in ACPI table) to understand the system-wide interrupt route by retrieving PIRQ routing information for each PCI devices.
- Create mapping table between PIRQ and PIC IRQ lines.
- Configure PCI configuration header's IRQ line value for each PCI device during PCI enumeration.
- Configure PIRQ settings and its associated PIC IRQ lines to be level-triggered.

For this BSP3.2 GOLD release, the AML parser requires ACPI table to be set-up in certain fashion as documented in following section.



11.2 Building OS Loader

Source Code Package:

Intel BLDK Core for CedarView - Windows (CR-EDKII-Beta-2.3.9.4_Source)	To be added
UDK2010.UP3	http://sourceforge.net/projects/edk2/files/UDK2010%20Releases/UDK2010.UP3/UDK2010.UP3.Complete.zip/download
OpenSSL 0.9.8l	http://www.openssl.org/source/openssl-0.9.8l.tar.gz
UEFI WEC7 OS Loader	Intel_CS\SRC\UEFIBOOTLOADER\WEC7OSLoader

Tool Chain:

Microsoft Visual Studio 2008	Contact MSFT/MSDN
Intel BLDK Development Kit (Intel_BLDK_Application_v2.0.1-win.zip)	http://www.intel.com/p/en_US/embedded/hwsww/software/bldk#download
ACPI iASL Compiler (iasl-win-20070508.zip)	http://www.acpica.org/downloads/binary_tools.php
Windows Driver Kit version 3790.1830 (1830_usa_ddk.iso)	http://msdn.microsoft.com/en-us/windows/hardware/gg487463
Patch.exe (utility)	http://gnuwin32.sourceforge.net/packages/patch.htm

Preparing Build Environment

1. Install Microsoft Visual Studio 2008 on development PC.
2. Extract and install Intel BLDK Development Kit by double-click on Intel(R)_Boot_Loader_Development_Kit.exe. The tool is installed to C:\Program Files (x86)\Intel\BLDK.
3. Extract and install Intel BLDK Core Code Base (CR-EDKII-Beta-2.3.9.4.exe) for Cedar Trail to C:\CDVEDKII.
4. Extract iasl-win-20070508.zip to C:\ASL and install iasl.exe to C:\ASL.
5. Extract 1830_usa_ddk.iso and install Microsoft Window Driver Kit (1830_usa_ddk\setup.exe) into its default directory C:\WINDDK.
6. Extract UDK2010.UP3 to any folder, e.g. C:\UDK2010UP3.
7. Copy the entire C:\UDK2010UP3\UDK2010.UP3 folder to C:\CBEDKII.

Note: Choose to overwrite all conflicts detected by the copy process.

8. Extract C:\UDK2010UP3\P.UDK2010.UP3.Network\NetworkPkg.zip locally and copy the entire C:\UDK2010UP3\P.UDK2010.UP3.Network\NetworkPkg folder to C:\CDVEDKII.
9. Extract C:\UDK2010UP3\P.UDK2010.UP3.SourceDebugging\SourceLevelDebugPkg.zip locally and copy the entire C:\UDK2010UP3\P.UDK2010.UP3.SourceDebugging\SourceLevelDebugPkg folder to C:\CBEDKII.
10. Extract openssl-0.9.8l.tar.gz to any folder and copy the entire openssl-0.9.8l folder to C:\CBEDKII\CryptoPkg\Library\OpenSslLib\



11. Extract patch-2.5.9-7-bin.zip to any folder and copy bin\patch.exe utility executable to C:\CBEDKII\CryptoPkg\Library\OpensslLib\openssl-0.9.8l.
12. Open a command prompt and enter the following commands in the exact order to patch the OpenSSL:

```
>cd C:\CDVEDKII\CryptoPkg\Library\OpensslLib\openssl-0.9.8l
> patch -p0 -i ..\EDKII_openssl-0.9.8l.patch
> cd ..
> install.cmd
```

13. From Intel WEC7 BSP3.2 source code, copy the following files & folders from C:\INTEL_CS\SRC\UEFIBOOTLOADER\WEC7OSLoader to C:\CDVEDKII folder:
 - build_CE7OSLOADER.bat
 - cleanCE7OSLoader.bat
 - MdeModulePkg

Building WEC7 OS Loader for BLDK (BootIA32.efi)

1. Open a command prompt and enter the following commands to build BootIA32.efi

```
>cd C:\CDVEDKII
> edksetup.bat
[Ignore "!!! WARNING !!! No CYGWIN_HOME set, gcc build may not be used !!!"]
> build_CE7OSLOADER.bat
```

2. The output of the building process, i.e. BootIA32.efi is located in C:\CDVEDKII\Build\MdeModule\RELEASE_VS2008x86\IA32



11.3 Building BLDK Firmware

Please refer to documentation in http://www.intel.com/p/en_US/embedded/hsw/software/bldk#download for building BLDK firmware for the Intel Atom processor based platform.

11.4 Building BLDK-compatible Intel Atom WEC7 OS

A BLDK-compatible Intel Atom processor WEC7 image is specific to platform. Please "set BSP_UEFI=1" for the desired Intel Atom processor based platform.

For BSP3.2 GOLD, ONLY Intel Atom processor N2600/N2800/D2550 + NM10 chipset based platform (formerly Cedar Trail) is supported.

11.5 Requirements of AML Byte Code

This parser retrieves the PIRQ routing information for each PCI devices under following assumptions in the ACPI specification.

- Each device has, if any, *only one* _PRT information which has the route information.
- And, a _PRT can be found in the package of a DEVICE() op.
- Child PCI bus can be found in the Root PCI bus scope.

The implementation of simple AML parser design includes three subtasks, address them below in order.

1. Retrieve the Root PCI bus routing information.
 - Search the root PCI bus device in the AML byte codes by identifying a DEVICE() OP which has the root device PNPID "PNP0A08"
 - Find _PRT Method OP under this root DEVICE() OP package. Then, get the route information.
2. Retrieve other child PCI buses' routing information.
 - Search any other PCI DEVICE() OP with _PRT Method to get the route information.
3. Retrieve the system-wide available interrupt PIRQ numbers.
 - Search the PCI interrupt link device in the AML byte codes by identifying a DEVICE() OP which has the PCI interrupt link device PNPID "PNP0C0F"

The AML parser is not a full-featured AML parser/executor, therefore, it is not able to support all valid ASL usages. Please use the following template for the PCI routing report in ASL code.



```

Device (XXXX) (
    ...
    Method (_PRT, 0) {
        If (LEqual(YYYY, Zero)) { // 8259 Interrupt Routing
            return (Package() {
                ...
                Package() { 0x0000FFFF, 0, ZZZZ, 0 }
                ...
            })
        } else { // IOAPIC Routing
            return (Package() {
                ...
                Package() {0x0000ffff, 0, 0, 16},
                ...
            })
        }
    }
}

XXXX: The PCI device name

YYYY: The variable used to indicate which interrupt controller the
      OS will use. 0: 8259 interrupt controller 1: I/O APIC
      interrupt controller

ZZZZ: The device path for the link device. The last node must be
      LNKA-LNKH

```

The general rule:

The legacy 8259 PIRQ routing information returned by _PRT method should be able to retrieve through static AML code parsing.



Detailed rules and examples:

- I. The PCI interrupt routing information should be reported through the _PRT method within proper device scope.

The recommended way:

```
Device (DEV0) (  
    ...  
    Method (_PRT, 0) {  
        return (Package() {  
            ...  
            Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.LNKA, 0 }  
            ...  
        })  
    }  
}
```

- II. Report _PRT through "Name" keyword is **NOT** supported.

For example (**NOT supported**):

```
Device (DEV0) (  
    ...  
    Name (_PRT, Package() {  
        ...  
        Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.LNKA, 0 }  
        ...  
    })  
}
```



III. _PRT method should return a direct constant package.

Returning a PCI routing information package through a package variable object **cannot** be supported.

For example **(NOT supported)**:

```

Device (DEV0) (
    ...
    Name (PR00, Package() {
        ...
        Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.LNKA, 0 },
        ...
    }
    Method (_PRT, 0) {
        Return (PR00)
    }
}
    
```

IV. The legacy 8259 link object name should be LNKA, LNKB, LNKC, LNKD, LNKE, LNKF, LNKG and LNKH.

Other names **CANNOT** be recognized by the simplified ASL parser.

For example **(NOT supported)**:

```

Device (DEV0) (
    ...
    Method (_PRT, 0) {
        return (Package() {
            ...
            Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.IRQA, 0 }
            ...
        })
    }
}
    
```



- V. Only one set of routing information should be reported in _PRT method for legacy 8259 controller.

Returning different PCI routing information package on different conditions cannot be supported.

For example **(NOT supported)**:

```
Device (DEV0) (  
    ...  
    Method (_PRT, 0) {  
        If (LEqual(TST1, Zero)) {  
            return (Package() {  
                ...  
                Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.LNKA, 0 }  
                ...  
            })  
        } else {  
            return (Package() {  
                ...  
                Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.IRQB, 0 }  
                ...  
            })  
        }  
    }  
}
```



VI. However, it is allowed to use condition to determine whether to return 8259 interrupt information or APIC interrupt information.

```

Device (DEV0) (
    ...
    Method (_PRT, 0) {
        If (LEqual(/GPIC, Zero)) { // 8259 Interrupt Routing
            return (Package() {
                ...
                Package() { 0x0000FFFF, 0, \_SB.ICH0.LPC0.LNKA, 0 }
                ...
            })
        } else { // IOAPIC Routing
            return (Package() {
                ...
                Package() {0x0000ffff, 0, 0, 16},
                ...
            })
        }
    }
}

```



VII. The legacy 8259 interrupt link device should use the following format to describe the possible interrupt numbers.

```
Device (LNKA) (  
    ...  
    Name(_PRS,  
        ResourceTemplate()  
        {IRQ(Level,ActiveLow,Shared){12,11,10,9,7,5,4,3}})  
    ...  
}
```