Intel® Atom™ Processor E3800 Product Family

Intel® Celeron® Processor N2807

Intel® Celeron® Processor J1900

Microsoft Windows* 8 (Win8, WES8) 32-bit & 64-bit I/O Drivers

Programming Guide

November 2014

Version 1.1 for Software Maintenance Release 1

Intel Confidential

By using this document, in addition to any agreements you have with Intel, you accept the terms set forth below.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

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.

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.

Code Names are only for use by Intel to identify products, platforms, programs, services, etc. ("products") in development by Intel that have not been made commercially available to the public, i.e., announced, launched or shipped. They are never to be used as "commercial" names for products. Also, they are not intended to function as trademarks.

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

Basis, Basis Peak, BlueMoon, BunnyPeople, Celeron, Centrino, Cilk, Flexpipe, Intel, the Intel logo, the Intel Anti-Theft technology logo, Intel AppUp, the Intel AppUp logo, Intel Atom, Intel CoFluent, Intel Core, Intel Inside, the Intel Inside logo, Intel Insider, Intel RealSense, Intel SingleDriver, Intel SpeedStep, Intel vPro, Intel Xeon Phi, Intel XScale, InTru, the InTru logo, the InTru Inside logo, InTru soundmark, Iris, Itanium, Kno, Look Inside., the Look Inside. logo, Mashery, MCS, MMX, Pentium, picoArray, Picochip, picoXcell, Puma, Quark, SMARTi, smartSignaling, Sound Mark, Stay With It, the Engineering Stay With It logo, The Creators Project, The Journey Inside, Thunderbolt, the Thunderbolt logo, Transcede, Transrf, Ultrabook, VTune, Xeon, X-GOLD, XMM, X-PMU and XPOSYS are trademarks of Intel Corporation in the U.S. and/or other countries.

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

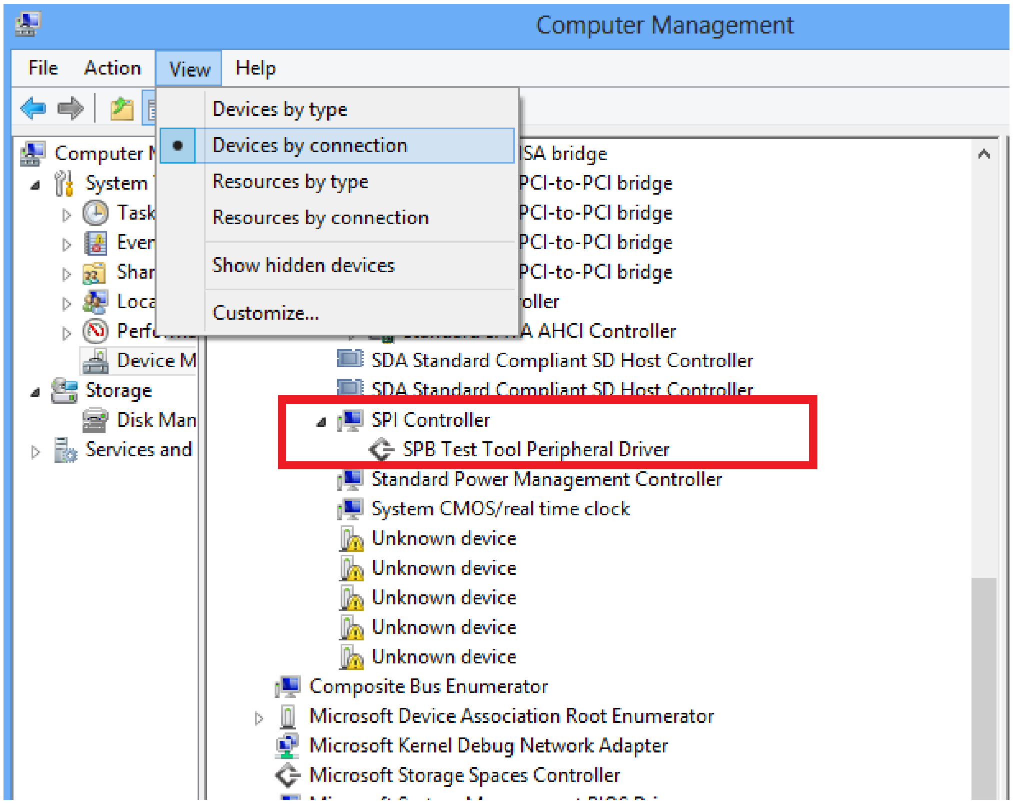
Contents

Revision History

Revision Number	Description	Revision Date
1.1	Maintenance Release 1	November 2014
1.0	Gold Release	April 2014

1 Introduction

Microsoft Windows* 8 provided new frameworks for GPIO, I^{2}C, SPI and UART driver. So user-mode applications cannot directly open the controller devices using traditional methods because the GPIO/I2C/SPI/UART controllers do not expose any symbolic links or GUID. So the only way is to mount one sub-device under the controller which is able to open the parent target device and use this sub-device to receive requests from user-mode applications. Overall, all these controllers can be used with a similar method. Below is a structure example of the relationship between the controller (parent) and test device (sub-device).



2 GPIO Driver

The GPIO driver in Windows 8 uses the Microsoft framework called GPIOClx. To use the GPIO controller, there must be a sub-device mounted under the specific GPIO controller. A user mode application can open this sub-device by using its symbolic name or GUID, and send IOCTLs or requests to it. Then only this sub-device can open the parent target (GPIO controller) and forward IOCTLs or requests to the GPIOClx framework, thus to GPIO controller driver.

Refer to the following for Microsoft framework:

http://msdn.microsoft.com/en-us/library/windows/hardware/hh439508%28v=vs.85%29.aspx

Here’s a **sample code** from Microsoft showing how to write the sub-device driver mounted under GPIO controller, to open its parent and forward requests:

http://code.msdn.microsoft.com/windowshardware/GPIO-Samples-d25ca63b

And the description of supported IOCTLs:

http://msdn.microsoft.com/en-us/library/windows/hardware/hh439470%28v=vs.85%29.aspx

3 I^{2}C and SPI Driver

I^{2}C and SPI drivers in Windows 8 used the Microsoft framework called the Simple Peripheral Bus (SPBClx). To use I^{2}C/SPI controller, there must be a sub-device mounted under the specific I^{2}C/SPI controller. A user mode application can open this sub-device by using its symbolic name or GUID, and send IOCTLs or requests to it. Then only this sub-device can open parent target (I^{2}C/SPI controller) and forward IOCTLs or requests to SPBClx framework, thus to I^{2}C/SPI controller driver.

Refer to the following for Microsoft framework:

http://msdn.microsoft.com/en-us/library/windows/hardware/hh450906%28v=vs.85%29.aspx

Here's a **sample code** from Microsoft showing how to write the sub-device driver mounted under the I^{2}C/SPI controller to open its parent and forward requests:

http://code.msdn.microsoft.com/windowshardware/SpbTestTool-adda6d71

And the description of supported IOCTLs is here:

http://msdn.microsoft.com/en-us/library/windows/hardware/hh450915%28v=vs.85%29.aspx

4 UART Driver

The UART driver in Win8 uses the framework of Microsoft called Serial Framework Extension (SerCx). To use UART controller, there must be a sub-device mounted under the specific UART controller. A user mode application can open this sub-device by using its symbolic name or GUID, and send IOCTLs or requests to it. Then only this sub-device can open parent target (UART controller) and forward IOCTLs or requests to SerCx framework, thus to UART controller driver.

Refer to the following for Microsoft framework:

http://msdn.microsoft.com/en-us/library/windows/hardware/dn265348%28v=vs.85%29.aspx

And the description of supported IOCTLs is here:

http://msdn.microsoft.com/en-us/library/windows/hardware/ff547466(v=vs.85).aspx

There are no sample codes of the sub-device provided by Microsoft, but it’s quite similar to the SPB or GPIO interface.

5 BKMs

5.1 How to Enable COM Port to Support Serial Debugging

The common serial port on Bayley Bay & Bakersport boards doesn't work. The actual serial port is the MicroUSB port near the COM port on CRB board. A USB cable is needed to connect the micro-USB port in CRB board to the USB port on the host machine.

1. Open the Windows Command Prompt and type “`bcdedit /debug on`” followed by “`bcdedit /dbgsettings serial debugport:1 baudrate:115200`” to enable the debug mode.
2. Insert the micro-USB cable from the platform to the host machine.
3. Install a “winDbg” on the host machine from this link:

   http://msdn.microsoft.com/en-US/windows/hardware/hh852363
4. Open “windbg” in the host machine → “File” menu → "Kernel Debug".
5. Enter the following settings:
   - Baud rate: 115200
   - Port: COM5 (depending on the host machine's USB port name)
6. Reboot the system.

5.2 Another Way to Install the I/O Driver

By default, the IO driver will be installed via ”Intel Atom E3800 Win8 IO Drivers 32Bit.msi” or “Intel Atom E3800 Win8 IO Drivers 64Bit.msi”. Alternatively, you can also install with a Windows image. To do this, you need to extract the INF and SYS file from a system that has installed the Intel IO drivers. The Intel IO driver INF and sys files can be found at the following folders after installation:

For 64 bit driver: [Program Files]\Intel\Intel Atom E3800 Win8 IO Drivers 64bit.

For 32 bit driver: [Program Files]\Intel\Intel Atom E3800 Win8 IO Drivers 32bit.

To perform the alternative Windows 8 driver installation, please consult the following link: http://technet.microsoft.com/en-us/library/hh825070.aspx

5.3 How to Create OS Boot from USB Device for Win8

1. Prepare the setup environment:

    - Connect the recommended USB flash device (from which you wish to deploy the Win8 image) to the USB port.
    - Connect the storage device that contains the Win8 image
    - Connect to the hard disk which has the Win8 operating system

2. Power up the system and install the Win8 OS.
3. Open Control Panel and select Windows To Go.
4. Choose the drive you want to use. Select the desired drive and click Next.
5. Choose the desired Win8 image and click Next.
6. Use BitLocker Password if you want or click Skip.
7. When you are ready to create your Windows To Go workspace, click Create to start the installation process.

Note: Once you click Create, the data inside the Windows To Go USB flash drive will be deleted permanently.

8. In Choose a boot option, recommend to select No. Follow by either select either Save and restart or Save and close.
9. Make sure the Windows To Go USB flash drive is set to first boot in the BIOS.
10. Restart the system and boot into the Windows To Go USB flash drive.
11. Windows 8 in Windows To Go USB flash drive can be loaded without error.

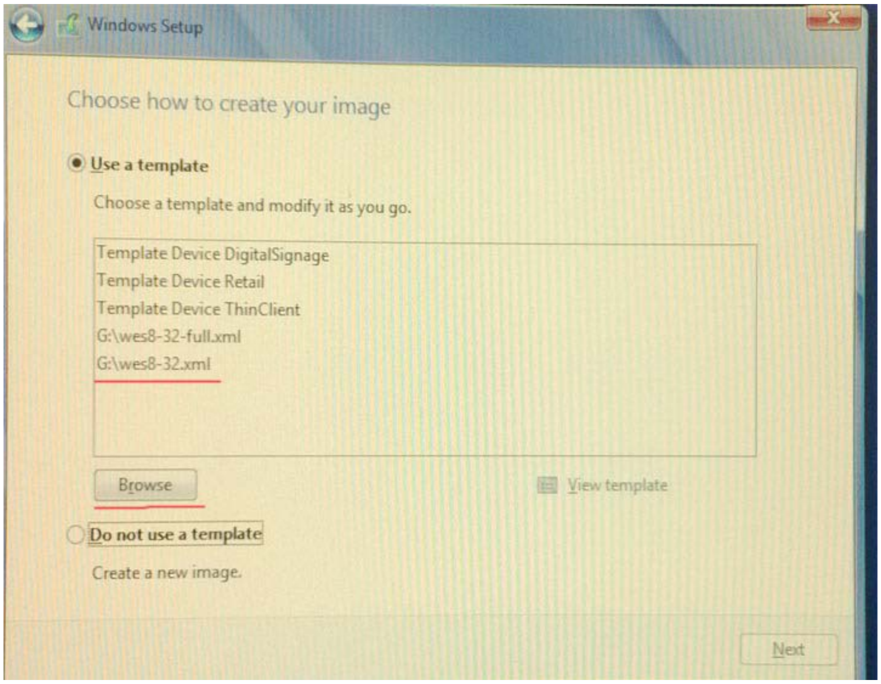
5.4 How to Install USB 3.0 and eMMC Driver in WES8

WES8 doesn’t install in-box drivers for USB 3.0 and eMMC card by default. USB 3.0 and eMMC card will not work when the default installation template is used to install WES8.

Custom installation templates for WES8 are provided to include the drivers during OS installation. They are located in the Intel IO Driver Installation zip package’s Custom Templates folder.

“**WES8_64bit_Installation_Template**” for WES8 64-bit
“**WES8_32bit_Installation_Template**” for WES8 32-bit

The file is an XML file that you can save into a flash thumb drive and use during your Windows setup.



Windows Setup

Choose how to create your image

● Use a template

Choose a template and modify it as you go.

Template Device DigitalSignage
Template Device Retail
Template Device ThinClient
G:\wes8-32-full.xml
G:\wes8-32.xml

○ Do not use a template

Create a new image.

6 Platform BKMs

6.1 How to Rework Bakersport Fab B I^{2}C Port 6

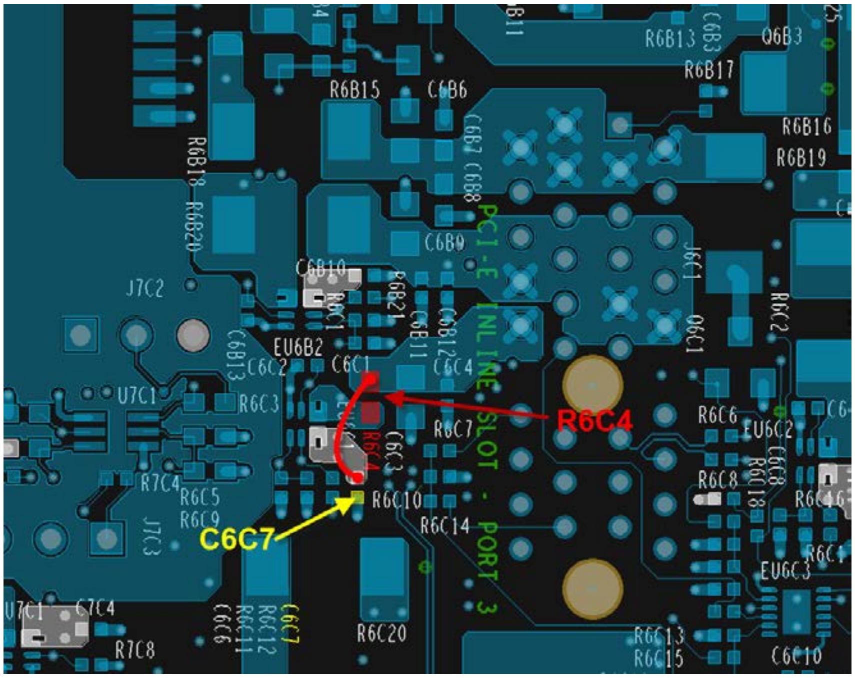
By default, Bakersport Fab B has an issue with I^{2}C port 6. This port fails to read and write due to an incorrect resistor connection.

Rework Steps	1) UnStuff R5H9, R5H12, R5H8, R5H10 2) Stuff R5H4 (22 ohms) 3) Stuff R5H3 (22 ohms)
Affected Platform	Bakersport boards **(PBA# G72250-200 Rev 02) (Fab B)**

6.2 How to Rework Bayley Bay Fab 3 PCI-E INLI Slot Port 3

By default, Bayley Bay Fab 03 has an issue with PCI-E Slot 3. This PCIe slot fails to detect network card after shutdown followed by power up (without switch off the main power)

Rework Steps	1. Remove R6C4 2. Add jumper wire from C6C7 to R6C4 as shown below.
Reasons for the rework:	NIC cards don't get recognized in Windows while the jumper block (J7C2) is configured to Desktop mode, pins [1–2]. Failure mode occurs in PCI-E Slot 3
Affected Platform	Bayley Bay boards Fab 3 (IOTG configured) platforms only



6.3 How to Rework Bakersport Fab B USB 3.0 Port

By default, Bakersport Fab B has an issue with USB 3.0 port. This port fails to read several USB 3.0 thumb drives and couldn’t achieve USB 3.0 performance.

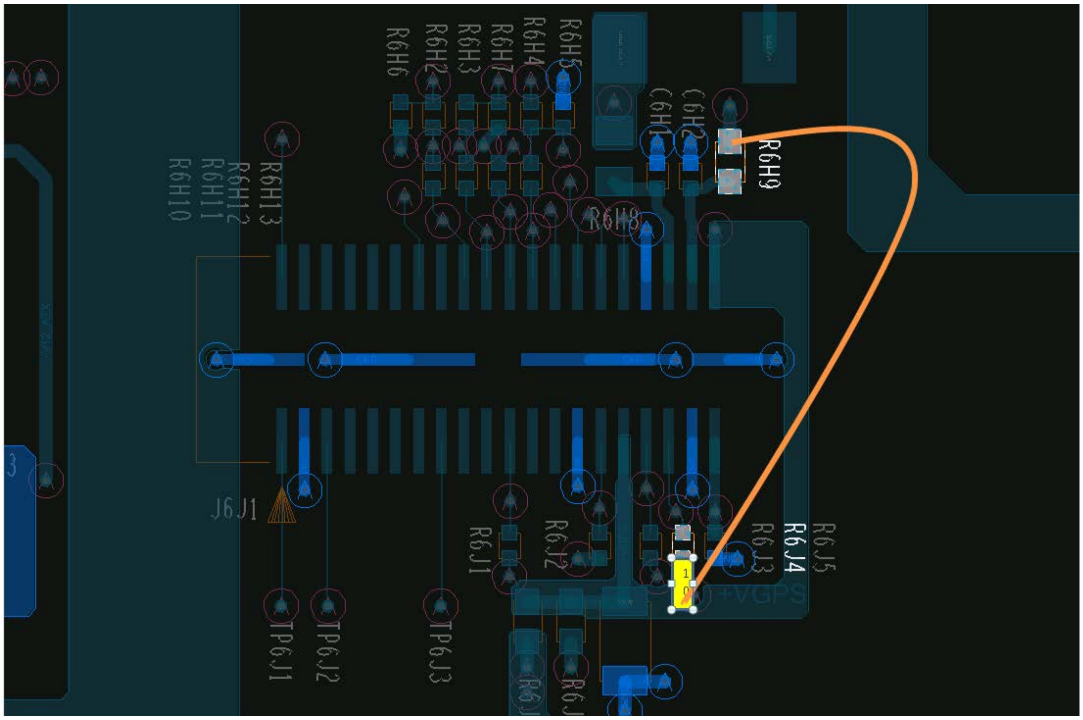
Rework Steps	1) UnStuff choke on L8A2 2) Stuff R8A4 and R8A3 (0 ohms)
Affected Platform	Bakersport boards **(PBA# G72250-200 Rev 02) (Fab B)**

Note: Patriot Memory 64GB and EDGE DiskGo* 32GB thumb drive are not recommended for use in EHCI mode.

6.4 How to Rework UART in Bakersport and Bayley Bay

By default, Bakersport Fab B has an issue with I^{2}C port 6. This port fails to read and write due to incorrect resistor connection.

Rework Steps	1. Place a 10K resistor followed by a wire from R6J4 to R6H9 See below rework layout, yellow box is the 10K PU resistor followed by orange wire to R6H9
Affected Platform	Bakersport boards **(PBA# G72250-200 Rev 02) (Fab B)** Bayley Bay boards Fab 3 (IOTG configured) platforms only


R6H6
R6H2
R6H3
R6H7
R6H4
R6H5
C6H1
C6H2
R6H9
R6H13
R6H12
R6H11
R6H10
R6H8
J6J1
R6J1
R6J2
R6J3
R6J4
R6J5
+VGPS
TP6J1
TP6J2
TP6J3

6.5 Setting up the BIOS

BIOS setup is required for both Win8 and WES8 64-bit OS.

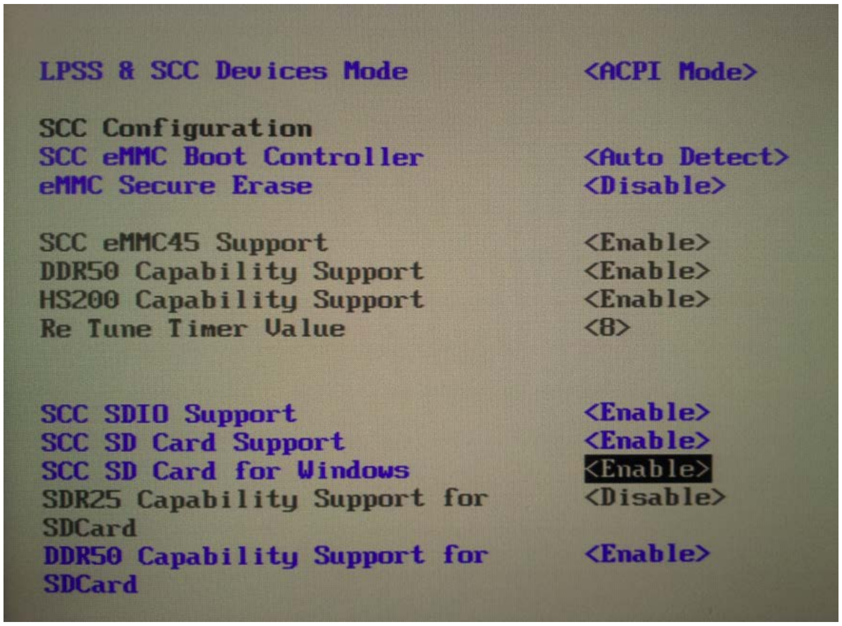
1. In BIOS setting, enter in “Device Manager -> System Setup > Boot > OS Selection: select Windows8.X and press F4 to save.

2. In BIOS setting, enter in “Device Manager -> System Setup > Boot > disable the UEFI Security Boot then commit changes and Exit.

3. After reboot, make sure setting below has been changed:

   “Device Manager -> *South Cluster Configuration—LPSS & SCC Configuration*
   - LPSS & SCC Device Mode = “ACPI Mode”
   - SCC SD Card for Windows = “Enable”

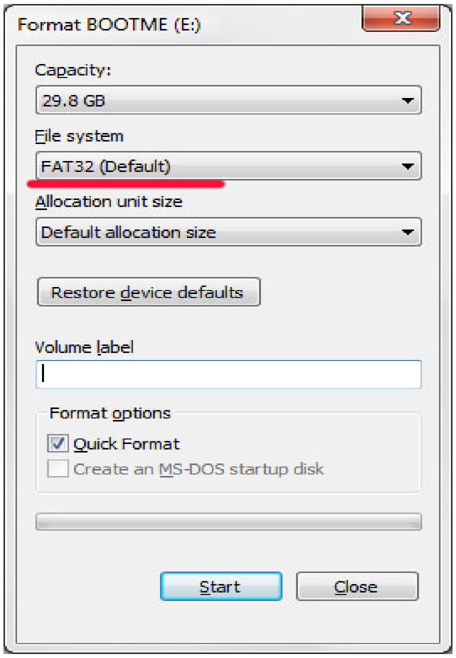
   South Cluster Configuration—Audio Configuration
   - *Audio Controller* = “Enable”



```
LPSS & SCC Devices Mode                  <ACPI Mode>

SCC Configuration
SCC eMMC Boot Controller                 <Auto Detect>
eMMC Secure Erase                        <Disable>

SCC eMMC45 Support                       <Enable>
DDR50 Capability Support                 <Enable>
HS200 Capability Support                 <Enable>
Re Tune Timer Value                      <8>


SCC SDIO Support                         <Enable>
SCC SD Card Support                      <Enable>
SCC SD Card for Windows                  <Enable>
SDR25 Capability Support for             <Disable>
SDCard
DDR50 Capability Support for             <Enable>
SDCard
```

6.6 OS Installation Environment Settings

1. Get a thumb drive with capacity larger than 8GB – 32GB, and format it with FAT32.



Format BOOTME (E:)

Capacity:
29.8 GB

File system
FAT32 (Default)

Allocation unit size
Default allocation size

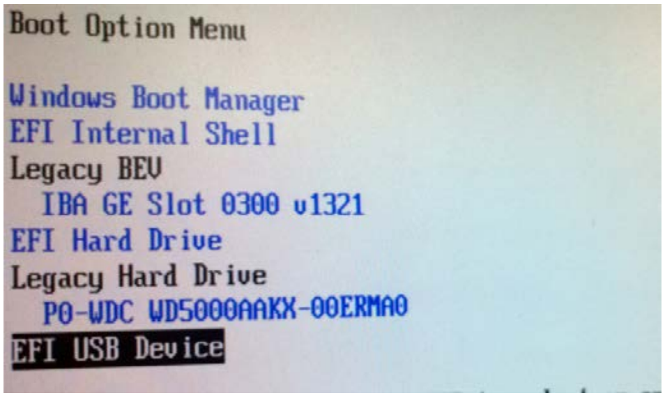
Volume label

Format options
☑ Quick Format
□ Create an MS-DOS startup disk

2. Extract all files from the ISO image of Win8/WES8 64 bit to your thumb drive.

   NOTE: For WES8, to include additional Windows Embedded Catalog drivers and application, you may copy Intel's custom template file created with WES8 Toolkit Please refer to Section 5.4 for steps for generating the XML for WES 32 and WES 64 bit.

3. Reboot and access the BIOS Settings again. Choose "Boot Manager" then the "EFI USB Device" to boot from.



```
Boot Option Menu

Windows Boot Manager
EFI Internal Shell
Legacy BEV
   IBA GE Slot 0300 v1321
EFI Hard Drive
Legacy Hard Drive
   P0-WDC WD5000AAKX-00ERMA0
EFI USB Device
```

4. After booting into the USB thumb drive, for Win8, install with the Windows OS default installation steps. Click next, choose a partition to install, and then start the installation.

For WES8, during setup for installation, when asked to choose a template, click "Browse" and choose to use the custom template that you have generate. Attached is the **Baytrail_Template_WES8_64.xml** for reference. Click next, and choose a partition to install, and then start the installation.

Also refer to BKM Section 5.4, "**How to install USB 3.0 and eMMC driver in WES8**" about why we need this template file.

5. The system will reboot after OS installation. Ensure the system is able to boot into the OS.

6. Reboot the system and boot into OS.