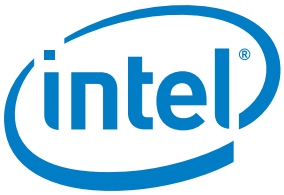


Intel® Atom™ Processor E3800 Product Family,

Intel® Celeron® Processor N2807 &

Intel® Celeron® Processor J1900

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

Programming Guide

October 2015

Version 001 for Software Maintenance Release 1

Document Number: 333231-001US

Contents

Figures

Tables

October 2015
Document Number: 333231-001US

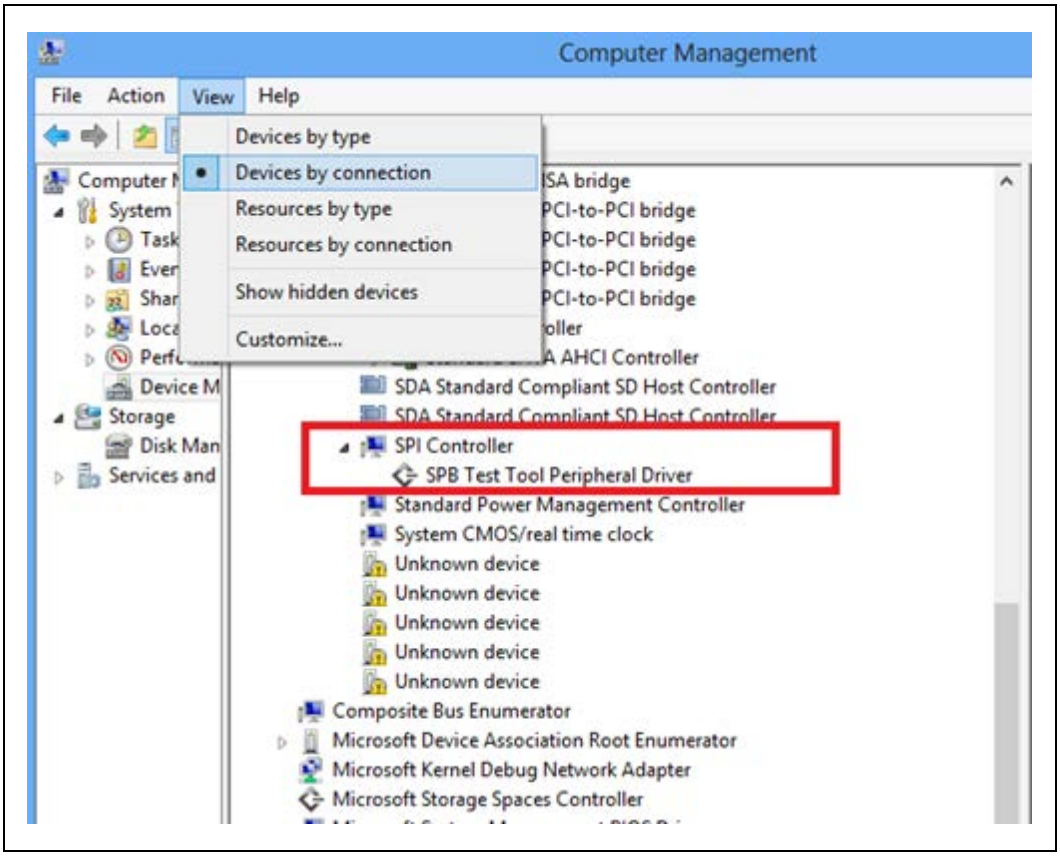
Revision History

Revision Number	Description	Revision Date
001	Software Maintenance Release 1.	October 2015

1.0 Introduction

The Microsoft* Windows* 8 operating system provides new frameworks for GPIO, I²C*, SPI and UART drivers. Therefore, user-mode applications cannot open the controller devices directly using traditional methods because the GPIO/I2C/SPI/UART controllers do not expose any symbolic links or GUID. The only way to open a controller device 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. The GPIO, I²C*, SPI and UART controller devices can be opened with a similar method. Figure 1 shows an example of a relationship between the controller (parent) and test device (sub-device).

Figure 1. Example of a Controller (Parent) and Test Device (Sub-Device) Relationship



October 2015
Document Number: 333231-001US

1.1 Acronyms and Terminology

Table 1. Acronyms and Terminology

Term	Description
BKM	Best Known Method
EFI	Extensible Firmware Interface
EHCI	Enhanced Host Controller Interface
GPIO	General Purpose Input/Output
GUID	Globally Unique Identifier
IOCTL	Input/Output Control
WES8	Windows* Embedded 8 Standard
Win8	Windows 8 Enterprise

2.0 GPIO Driver

The GPIO driver in the Microsoft* Windows* 8 operating system uses the Microsoft framework called GPIOClx. To use a 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 the 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 code the sub-device driver mounted under GPIO controller, to open its parent and forward requests:

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

Refer to the following for the description of supported IOCTLs:

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

3.0 I²C* and SPI Driver

I²C* and SPI drivers in the Microsoft* Windows* 8 operating system use the Microsoft framework called the Simple Peripheral Bus (SPBClx). To use an I²C/SPI controller, there must be a sub-device mounted under the specific I²C/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 the parent target (I²C/SPI controller) and forward IOCTLs or requests to the SPBClx framework, thus to the I²C/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 code the sub-device driver mounted under the I²C/SPI controller to open its parent and forward requests:

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

Refer to the following for the description of supported IOCTLs:

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

4.0 UART Driver

The UART driver in the Microsoft* Windows* 8 operating system uses the Microsoft framework called Serial Framework Extension (SerCx). To use a 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 the parent target (UART controller) and forward IOCTLs or requests to the SerCx framework, thus to the 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

Refer to the following for the description of supported IOCTLs:

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 the codes are quite similar to the SPB or GPIO interface codes.

5.0 BKMs

5.1 How to Enable COM Port to Support Serial Debugging

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

1. Open the Microsoft* 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 the "winDbg" tool 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 I/O 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 I/O drivers. The Intel I/O 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, refer to 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 operating system.
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 desired or click Skip.
7. When you are ready to create your Windows To Go workspace, click Create to start the installation process.

Note: After clicking Create, the data in the Windows To Go USB flash drive will be deleted permanently.

8. In Choose a Boot option, it’s recommended to select No, followed by either Save and Restart or Save and Close.
9. Ensure that 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 then be loaded without error.

5.4 How to Install USB 3.0 and eMMC* Drivers in WES8

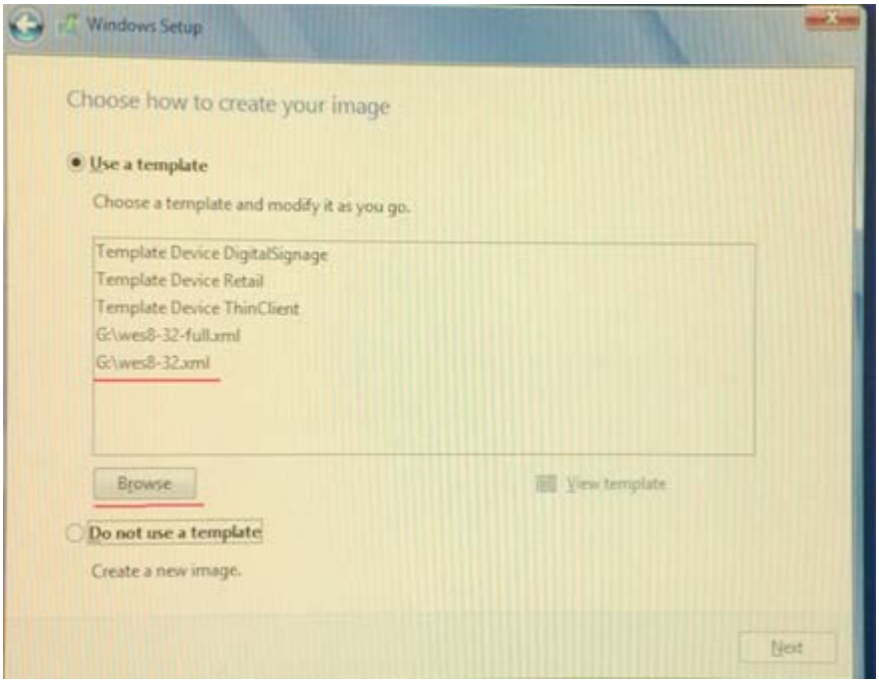
The WES8 operating system does not include inbox drivers for USB 3.0 and eMMC* card by default. Hence, USB 3.0 and eMMC card will not work when the default installation template is used to install the WES8 operating system.

Custom installation templates for the WES8 operating system are provided to include the drivers during operating system installation. They are located in the Intel I/O Driver Installation zip package's Custom Templates folder:

- For WES8 64-bit operating system: "**WES8_64bit_Installation_Template**"
- For WES8 32-bit operating system: "**WES8_32bit_Installation_Template**"

The installation template is an XML file that can be saved into a flash thumb drive and used during Windows setup, as shown in Figure 2.

Figure 2. Use an Installation Template during Windows* Setup



6.0 Platform BKMs

6.1 How to Rework Bakersport Fab B I²C* Port 6

By default, Bakersport Fab B has an issue with I²C* port 6. This port fails to read and write because of an incorrect resistor connection.

Table 2. How to Rework Bakersport Fab B I2C* Port 6

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)**

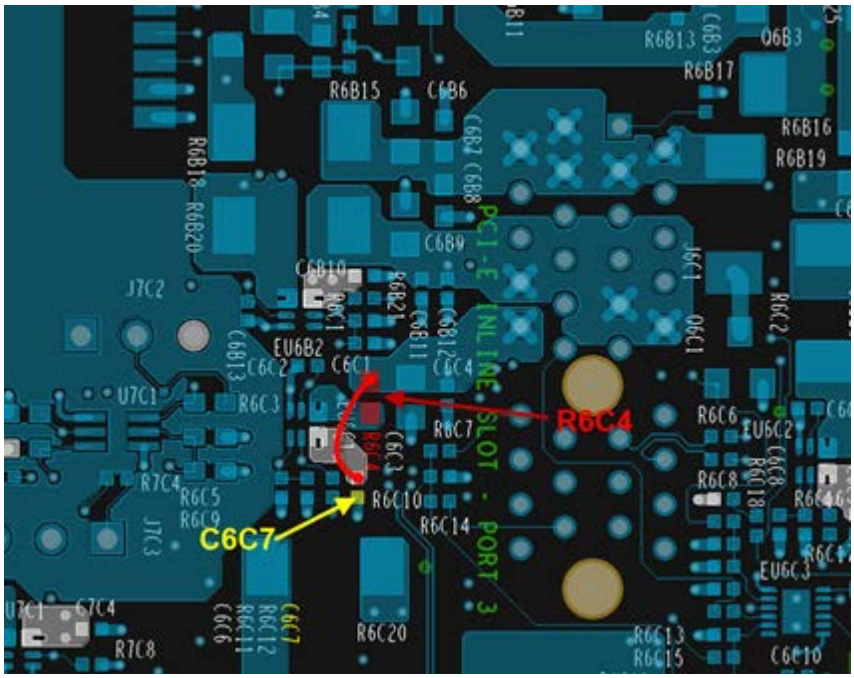
6.2 How to Rework Bayley Bay Fab 3 PCIe* INLI Slot Port 3

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

Table 3. How to Rework Bayley Bay Fab 3 PCIe* INLI Slot Port 3

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

Figure 3. How to Rework Bayley Bay Fab 3 PCIe* INLI Slot Port 3



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 could not achieve USB 3.0 performance.

Table 4. How to Rework Bakersport Fab B USB 3.0 Port

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 Memory* DiskGO* 32GB thumb drives are not recommended for use in EHCI mode.

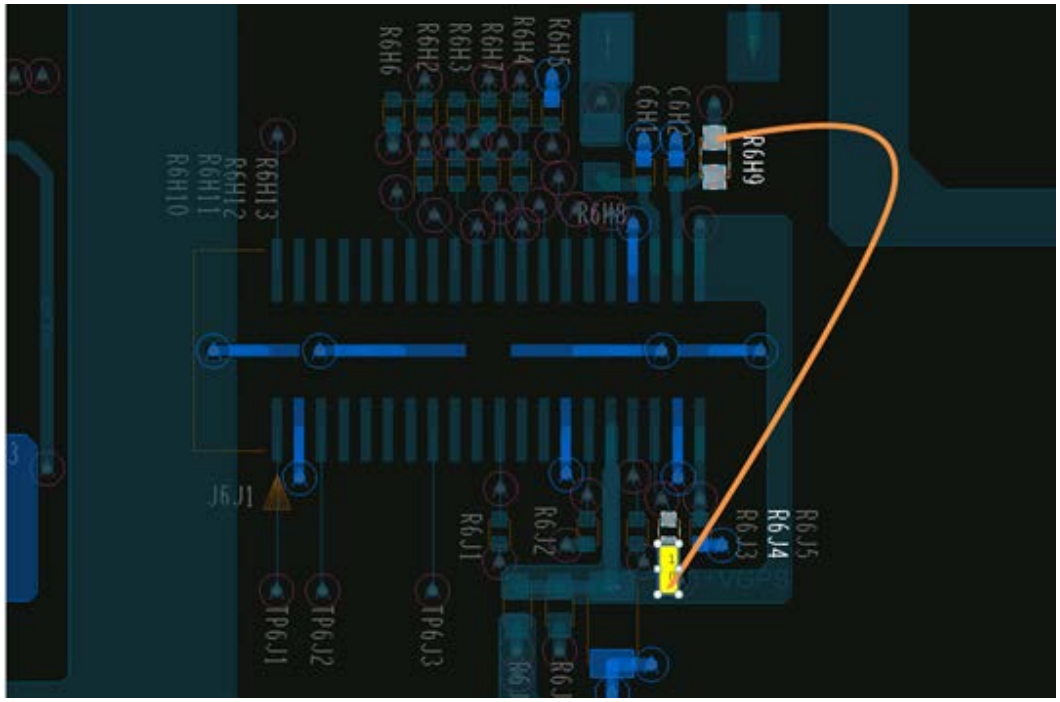
6.4 How to Rework UART in Bakersport and Bayley Bay

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

Table 5. How to Rework UART in Bakersport and Bayley Bay

Rework Steps	Place a 10K resistor followed by a wire from R6J4 to R6H9. Figure 4 shows that the 10K PU resistor (denoted by a yellow box) is wired (denoted by an orange curved line) to R6H9.
Affected Platform	Bakersport boards **(PBA# G72250-200 Rev 02) (Fab B)** Bayley Bay boards Fab 3 (IOTG-configured) platforms only

Figure 4. How to Rework UART in Bakersport and Bayley Bay



6.5 Setting up the BIOS

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

1. In the BIOS setting, go to “Device Manager -> System Setup > Boot > OS Selection”. Select Windows8.X and press F4 to save.

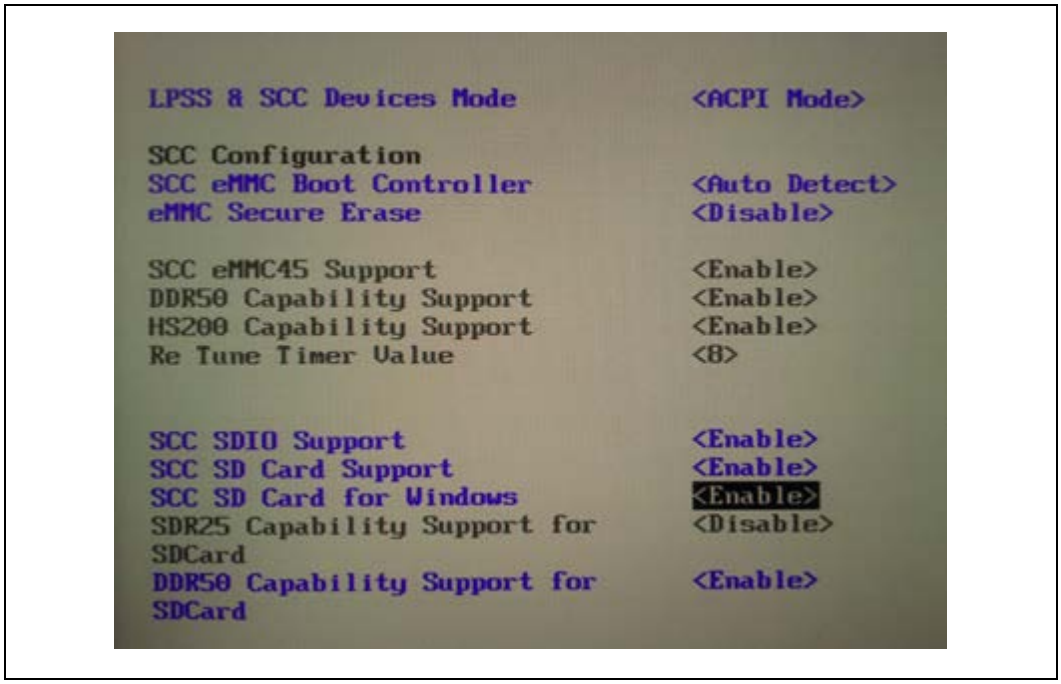
2. In the BIOS setting, go to “Device Manager -> System Setup > Boot”. Disable the UEFI Security Boot and then commit changes and exit.

3. After reboot, ensure that the following settings have been changed:

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

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

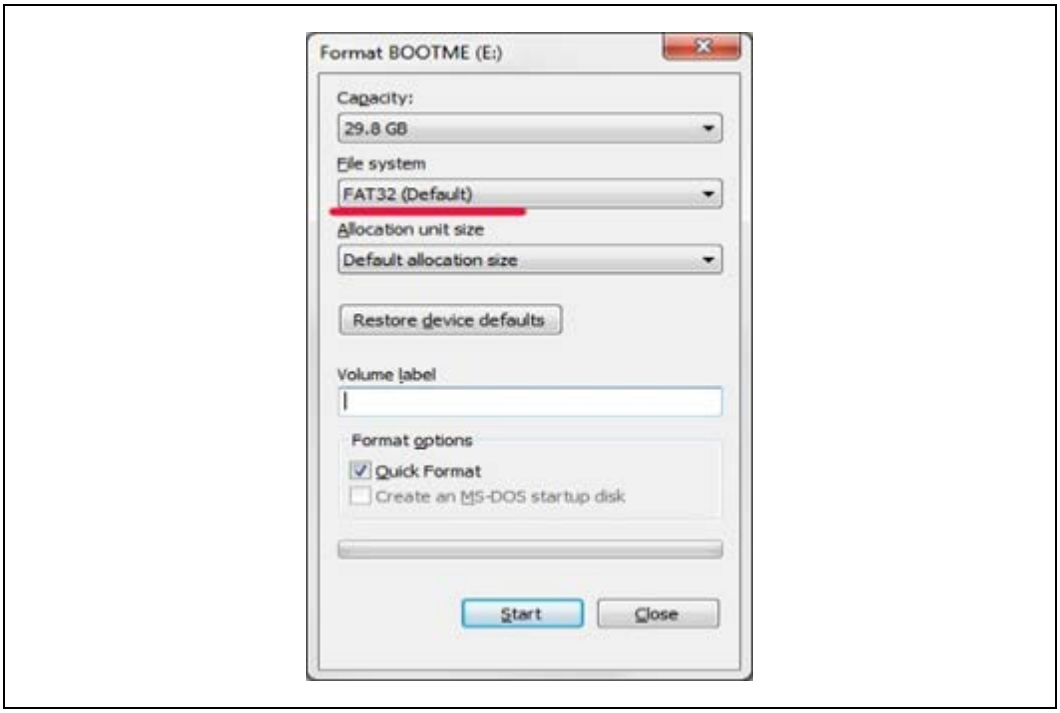
Figure 5. LPSS & SCC Configuration



6.6 OS Installation Environment Settings

1. Get a thumb drive with a capacity between 8GB – 32GB, and format it with FAT32 file system, as shown in Figure 6.

Figure 6. Format a Thumb Drive with FAT32 File System

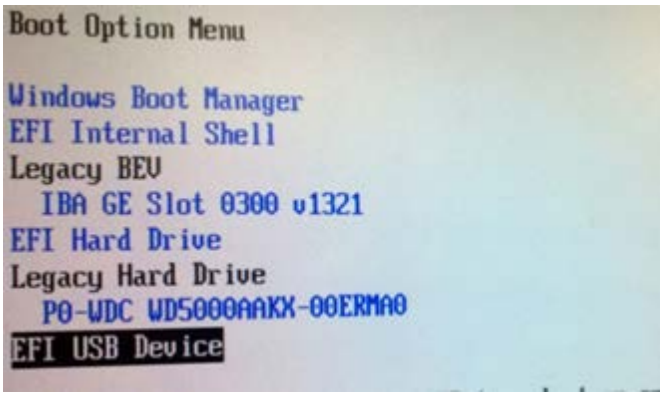


2. Extract all files from the ISO image of Win8 or WES8 64-bit operating system to your thumb drive.

   NOTE: For the WES8 operating system, use Intel's custom template file created with the WES8 Toolkit to include additional Windows Embedded Catalog drivers and application. Refer to Section 5.4 for steps on generating the XML file for WES8 32- and WES8 64-bit operating systems.

3. Reboot and access the BIOS Settings. Choose "Boot Manager", and then the "EFI USB Device" to boot from, as shown in Figure 7.

Figure 7. Select EFI USB Device from the Boot Option Menu



4. After booting into the USB thumb drive, follow these instructions according to the operating system:
   - For the Win8 operating system, install using the Windows operating system default installation steps. Click Next, choose a partition to install, and then start the installation.
   - For the WES8 operating system, during setup for installation, when asked to choose a template, click "Browse" and choose to use the custom template that you have generated. Use the **WES8_64bit_Installation_Template.xml** (or the **WES8_32bit_Installation_Template.xml**) file provided for reference. Click Next, choose a partition to install, and then start the installation.

     Refer to BKM in Section 5.4 on why this template file is needed.

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

6. Reboot the system and boot into the operating system.