BSP for Microsoft* Windows* 10 IoT Core 32-bit on Intel® Atom™ processor E3800 Product Family

User Guide

December 2015

Gold Release
Revision 001

Revision Number - 001US

Contents

Figures

Revision History

Date	Revision	Description
December 2015	001	BSP for Microsoft* Windows* 10 IoT Core 32-bit on Intel® Atom™ Processor E3800 Product Family – Gold Release

§

1.0 Introduction

1.1 Scope of Document

This User Guide describes how to install Microsoft* Windows* 10 developer tools to create an image for the Microsoft Windows 10 IoT Core 32-bit operating system, booting up the IoT Core image, upgrading I/O drivers (GPIO, I^2C*, SPI, and HSUART) for the IoT Core image, and best known methods for platforms and software drivers.

This User Guide is intended for OEMs and ODMs that are enabling IoT Core drivers for the Windows* 10 operating system with the Intel® Atom™ E3800 processor, Intel® Celeron® Processor N2XXX, and Intel® Celeron® Processor J1XXX.

1.2 System Requirements

The following operating system is supported:

- Microsoft Windows 10 IoT Core 32-bit operating system

1.3 Acronyms and Terminology

Term	Description
ADK	Assessment and Deployment Kit
BSP	Board Support Package
COM Port	Communication Port
CRB	Customer Reference Board
DMA	Direct Memory Access
EEAP	Ecosystem Engineering Access Program
EHCI	Enhanced Host Controller Interface
FFU	Full Flash Update
GPIO	General Purpose Input/Output
HSUART	High Speed Universal Asynchronous Receiver/Transmitter
I^2C*	Inter-Integrated Circuit*
NIC	Network Interface Card
ODM	Original Design Manufacturer
OEM	Original Equipment Manufacturer

Term	Description
PIO Mode	Programmed I/O Mode
RTM	Release to Manufacturing
SDK	Software Development Kit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
WDK	Windows Driver Kit

§

2.0 Best Known Configuration

The following is a list of software tools required to create and deploy the Windows* 10 IoT Core operating system with custom drivers on the hardware:

- Windows 10 RTM (Build 10586 or better) x86 or x64 (Windows 8.1 latest version also works but not recommended)
- Visual Studio* Professional 2015 : 14.0.23107.0 D14REL
- Visual Studio Tools for Universal Windows Apps : 14.0.23121.00 D14OOB
- Windows 10 IoT Core Packages : 10.0.10586
- Windows Software Development Kit - SDK Windows 10.0.10586
- Windows Driver Kit - WDK Windows 10.0.10586
- Windows Assessment and Deployment Kit - ADK Windows 10.0.10586
- Download Visual Studio and Windows development tool kits from https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx

§

3.0 Building Windows* 10 IoT Core Image

This section describes how to install Windows* 10 developer tools to create an image for the Windows 10 IoT Core 32-bit operating system, booting up the IoT Core image on a Bayley Bay CRB, and upgrading I/O drivers (GPIO, I2C*, SPI, and HSUART) for the IoT Core image.

3.1 Installing Windows 10 Developer Tools

The following are steps to install Windows 10 developer tools:

1. Install the Windows 10 RTM operating system on a development machine.
2. Install the Microsoft Visual Studio* Professional 2015 software on the development machine.

Note: When installing any editions of Microsoft Visual Studio 2015, do a Custom install and select the checkbox Universal Windows App Development Tools -> **Tools and Windows SDK**.

3. Download the "Windows 10 IoT Core for MinnowBoard Max" from following link http://ms-iot.github.io/content/en-US/Downloads.htm and install to the development machine.
4. Download and install the Windows 10 SDK from the following:

   https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx

   The SDK includes the header file and library to compile the driver.
5. Download and install the Windows 10 WDK from the following:

   https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx

   The WDK will be used to build driver packages (refer to Upgrading I/O drivers in).
6. Download and install the Windows 10 ADK from the following:

   https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx

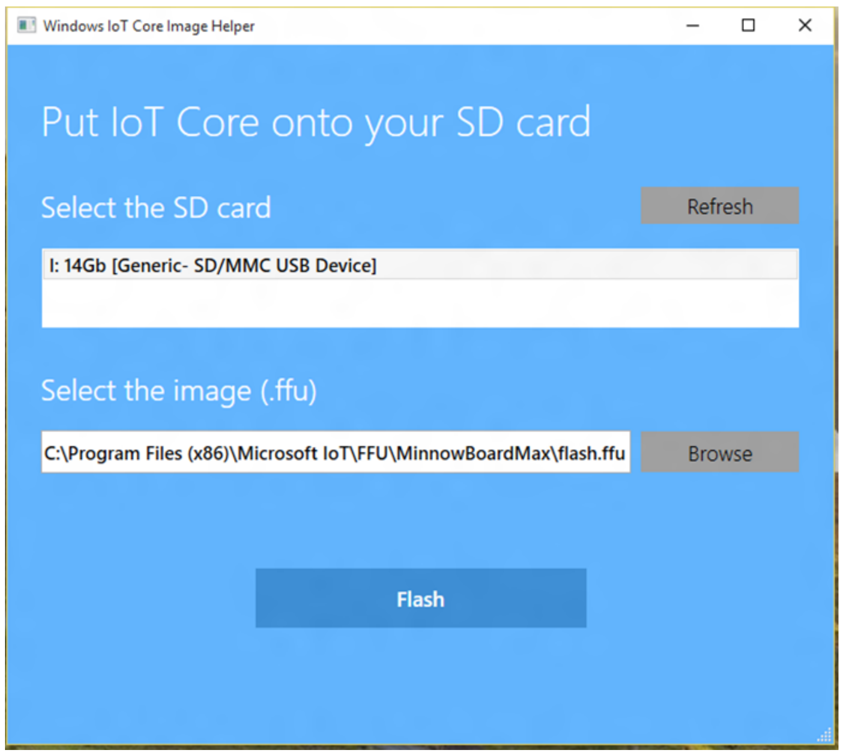
   In addition to new and improved deployment tools used to automate a large-scale Windows 10 deployment, the ADK includes the Windows Assessment Toolkit and Windows Performance Toolkit that access the quality and performance of systems or components.

3.2 Flashing Windows* 10 IoT Core FFU to Storage Device

3.2.1 Using SD or USB Storage Device

1. Insert an SD card, or a USB pen drive, into the development machine.
2. Run the **IoTCoreImageHelper.exe** tool to flash the storage device. This tool is located at C:\Program Files (x86)**Microsoft IoT**
3. This tool will enumerate the device as shown in Figure 1. Select the destination device, provide the FFU image location, and click **Flash** to flash the image.

Figure 1. Flash the IoT Core Image onto the SD* Card



4. Click on the Safely Remove Hardware icon in your task tray and select your device to safely remove it from the system. Failing to do this can cause corruption on the image.

3.2.2 Using SATA Disk

1. Connect the SATA disk to the development machine.
2. Open a Command Prompt and log in as the Administrator.
3. Run the "**diskpart**" command and then "**list disk**", and find the number **N** of this SATA disk.
4. Run the "**exit**" command to quit the DiskPart terminal.

5. Run the following command to flash the FFU image to SATA disk:

```
"C:\Program Files (x86)\Windows Kits\10\Assessment and
Deployment Kit\Deployment Tools\x86\DISM\dism.exe" /Apply-
Image /ImageFile:"C:\Program Files (x86)\Microsoft
IoT\FFU\MinnowBoardMax\Flash.ffu"
/ApplyDrive:\\.\PhysicalDriveN /SkipPlatformCheck
```

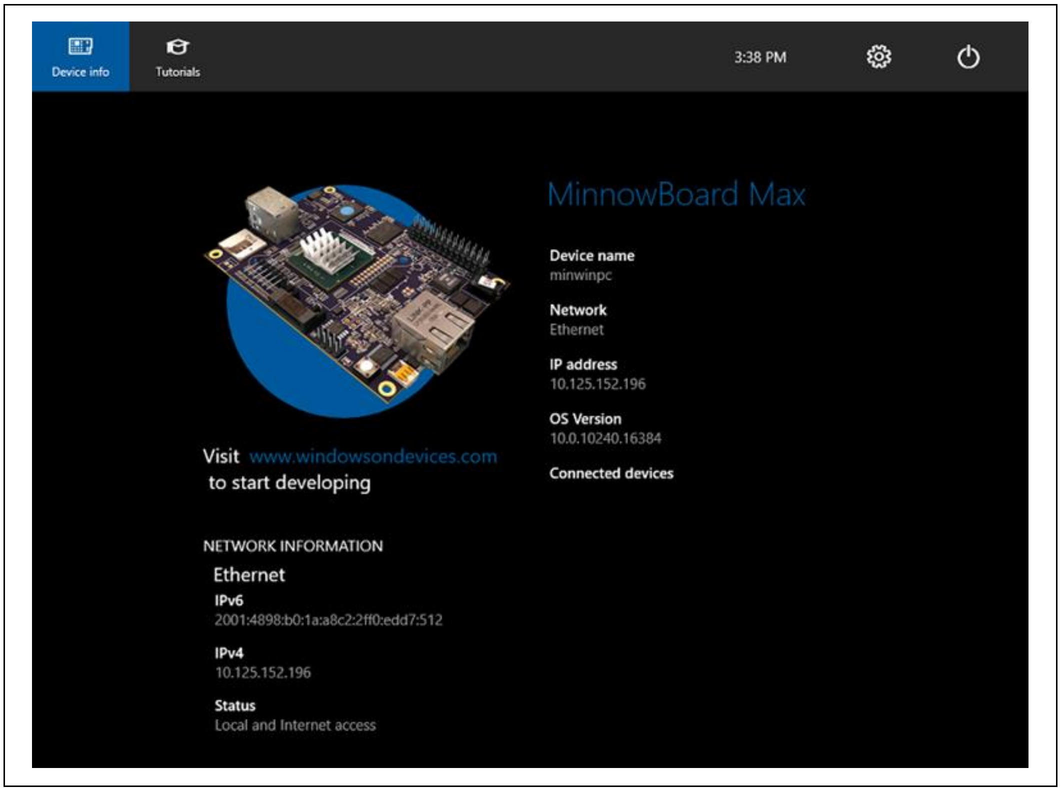
3.3 Booting up Windows* 10 IoT Core on Bayley Bay CRB

1. When the Bayley Bay CRB boots, press **F2** to get to the BIOS settings.
2. Navigate to Device Manager -> **System Setup**
   a. Boot -> OS Selection: Select "**Windows 8.X**".
   b. South Cluster Configuration -> LPSS & SCC Configuration -> LPSS & SCC Devices Mode: Select "**ACPI Mode**".
   c. South Cluster Configuration -> LPSS & SCC Configuration -> ACPI Reporting MMC/SD Media As: Select "**Non-Removable**".
3. Press **F4** to save these changes, or select "**Commit Changes and Exit**".
4. Navigate back to the first level and select Boot Maintenance Manager -> Boot Options -> **Add Boot Option**.
5. In File Explorer, select "**EFIESP**," item and press **Enter**, and navigate to "**`bootia32.efi`**" located at `EFI/Boot/`**`bootia32.efi`**. Press **Enter** to input the description, type any name, e.g. "IoTCoreImage".
6. Select "**Commit Changes and Exit**".
7. Navigate to "**Change Boot Order**". Highlight the boot order list and press **Enter.**
8. Highlight "**IoTCoreImage**" and press "**+**" to move it to the top of the list.
9. Select "**Commit Changes and Exit**".

   The Bayley Bay CRB should automatically boot to "**IoTCoreImage**".

Note: After selecting the Language, the device will boot up. If Language is not selected, the device may restart. After the device has booted, the default application from the USB thumb drive will launch and display the IP address of the Bayley Bay CRB.

Figure 2. Display of the Bayley Bay CRB IP Address



3.4 Upgrading I/O drivers in Windows 10 IoT Core

This section describes the steps to build a custom **.cab** and **.spkg** file using `pkggen.exe` for each of the following drivers:

- I^2C*
- GPIO
- SPI
- HSUART

For more `pkggen.exe` command options, refer to https://msdn.microsoft.com/en-us/library/dn756636(v=vs.85).aspx

This section also describes the steps to create a custom IoT Core FFU image.

As reference, refer to https://msdn.microsoft.com/en-us/library/windows/hardware/dn756642(v=vs.85).aspx

Use the following steps to build custom **.cab** and **.spkg** files for the drivers and create an FFU image:

1. Open an elevated **Deployment and Imaging Tools Environment** command prompt. The Deployment and Imaging Tools Environment shortcut can be found in the Windows start menu "**Windows Kits**" folder.
2. Set the following environmental variables:

```
set KITSROOT=C:\Program Files (x86)\Windows Kits\10\

set SIGN_OEM=1

set WPDKROOT=%KITSROOT%

set WPDKCONTENTROOT=C:\Program Files (x86)\Windows Kits\10

set path=%PATH%;%KITSROOT%\tools\bin\i386
```

3. To install the OEM certificates, go to the C:\Program Files (x86)\Windows Kits\10\Tools\bin**i386** directory in an elevated command console.

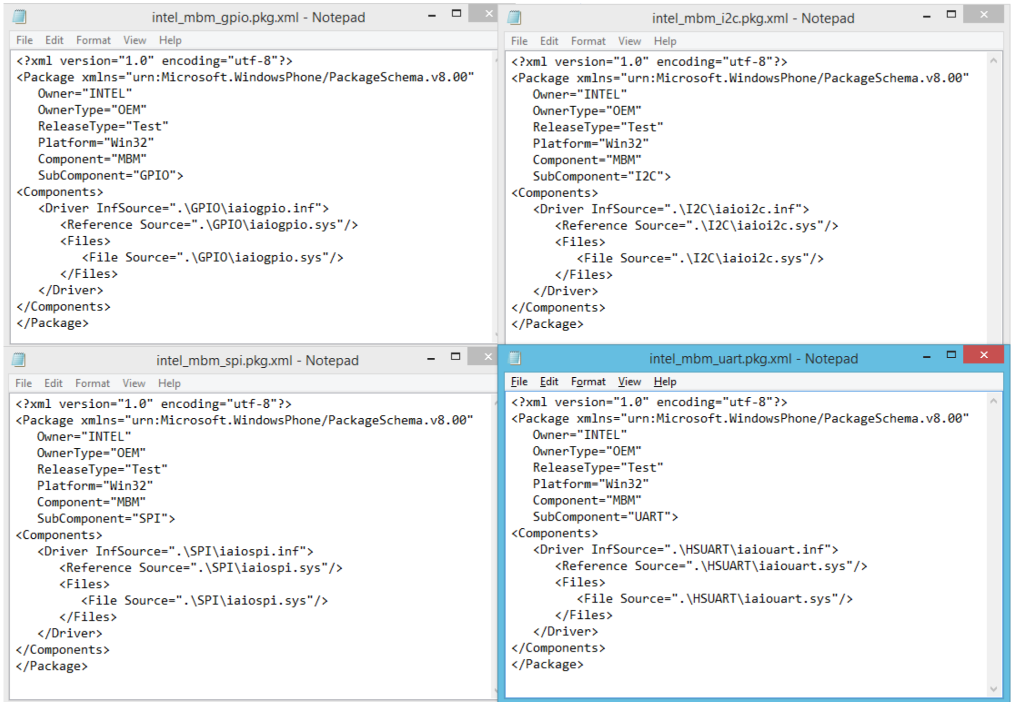
Note: The directory is applicable to the Windows* 10 x86 development machine. Change the directory accordingly if running on a Windows 10 x64 machine.

4. Run the following command: `installoemcerts.cmd`

Note: This command is only needed when building the package for the first time.

5. Create the **`.pkg.xml`** file for each driver. The file names and contents are shown in Figure 3.

Figure 3. Contents of the .pkg.xml Files



intel_mbm_gpio.pkg.xml - Notepad

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="urn:Microsoft.WindowsPhone/PackageSchema.v8.00"
   Owner="INTEL"
   OwnerType="OEM"
   ReleaseType="Test"
   Platform="Win32"
   Component="MBM"
   SubComponent="GPIO">
<Components>
   <Driver InfSource=".\GPIO\iaiogpio.inf">
      <Reference Source=".\GPIO\iaiogpio.sys"/>
      <Files>
         <File Source=".\GPIO\iaiogpio.sys"/>
      </Files>
   </Driver>
</Components>
</Package>
```

intel_mbm_i2c.pkg.xml - Notepad

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="urn:Microsoft.WindowsPhone/PackageSchema.v8.00"
   Owner="INTEL"
   OwnerType="OEM"
   ReleaseType="Test"
   Platform="Win32"
   Component="MBM"
   SubComponent="I2C">
<Components>
   <Driver InfSource=".\I2C\iaioi2c.inf">
      <Reference Source=".\I2C\iaioi2c.sys"/>
      <Files>
         <File Source=".\I2C\iaioi2c.sys"/>
      </Files>
   </Driver>
</Components>
</Package>
```

intel_mbm_spi.pkg.xml - Notepad

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="urn:Microsoft.WindowsPhone/PackageSchema.v8.00"
   Owner="INTEL"
   OwnerType="OEM"
   ReleaseType="Test"
   Platform="Win32"
   Component="MBM"
   SubComponent="SPI">
<Components>
   <Driver InfSource=".\SPI\iaiospi.inf">
      <Reference Source=".\SPI\iaiospi.sys"/>
      <Files>
         <File Source=".\SPI\iaiospi.sys"/>
      </Files>
   </Driver>
</Components>
</Package>
```

intel_mbm_uart.pkg.xml - Notepad

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns="urn:Microsoft.WindowsPhone/PackageSchema.v8.00"
   Owner="INTEL"
   OwnerType="OEM"
   ReleaseType="Test"
   Platform="Win32"
   Component="MBM"
   SubComponent="UART">
<Components>
   <Driver InfSource=".\HSUART\iaiouart.inf">
      <Reference Source=".\HSUART\iaiouart.sys"/>
      <Files>
         <File Source=".\HSUART\iaiouart.sys"/>
      </Files>
   </Driver>
</Components>
</Package>
```

6. Copy the drivers' **`.pkg.xml`** files, and the INF and SYS files to `C:\`**`Intel_IO`**
7. Run the following command:

```
cd c:\intel_io
```

8. Run the following command to generate the package for the I2C driver:

```
pkggen intel_mbm_i2c.pkg.xml
/config:"%KITSROOT%\Tools\bin\i386\pkggen.cfg.xml" /cpu:x86
/variables:"HIVE_ROOT=%WPDKCONTENTROOT%\CoreSystem\10.0.10586.
0\x86" /version:"10.0.1004.0"
```

9. After the package generation for I^2C* driver is complete, the following files can be found at the C:**\intel_io folder**:
   - INTEL.MBM.I2C.cab
   - INTEL.MBM.I2C.spkg
10. Run the following command to generate the package for the GPIO driver:

```
pkggen intel_mbm_gpio.pkg.xml
/config:"%KITSROOT%\Tools\bin\i386\pkggen.cfg.xml" /cpu:x86
/variables:"HIVE_ROOT=%WPDKCONTENTROOT%\CoreSystem\10.0.10240.
0\x86" /version:"10.0.1004.0"
```

11. After the package generation for GPIO driver is complete, the following files can be found at the C:\intel_io folder:
    - INTEL.MBM.GPIO.cab
    - INTEL.MBM.GPIO.spkg
12. Run the following command to generate the package for the SPI driver:

```
pkggen intel_mbm_spi.pkg.xml
/config:"%KITSROOT%\Tools\bin\i386\pkggen.cfg.xml" /cpu:x86
/variables:"HIVE_ROOT=%WPDKCONTENTROOT%\CoreSystem\10.0.10240.
0\x86" /version:"10.0.1004.0"
```

13. Modify the SPI driver version to “1.1.1.1008”.
14. After the package generation for SPI driver is complete, the following files can be found at the C:\intel_io folder:
    - INTEL.MBM.SPI.cab
    - INTEL.MBM.SPI.spkg
15. Run the following command to generate the package for the HSUART driver:

```
pkggen intel_mbm_uart.pkg.xml
/config:"%KITSROOT%\Tools\bin\i386\pkggen.cfg.xml" /cpu:x86
/variables:"HIVE_ROOT=%WPDKCONTENTROOT%\CoreSystem\10.0.10240.
0\x86" /version:"10.0.1004.0"
```

16. Modify the HSUART driver version to “**10.0.1008**”.
17. After the package generation for HSUART driver is complete, the following files can be found at the C:**intel_io folder**:
    - INTEL.MBM.UART.cab
    - INTEL.MBM.UART.spkg
18. Replace the existing **.cab** and **.spkg** files in C:\Program Files (x86)\Windows Kits\10\MSPackages\Retail\X86\fre with the new **.cab** and **.spkg** files.
19. Execute the following commands to create a custom IoT Core FFU image.

```
cd c:\ffu

set AKROOT=%KITSROOT%

imggen.cmd IoTCore.FFU
"%KITSROOT%OEMInputSamples\MBM\ProductionOEMInput.xml"
"%KITSROOT%MSPackages" x86
```

The build process takes about 20 to 30 minutes. The **`IoTCore.FFU`** image can be found at the `C:\`**`FFU`** **folder**.

§

4.0 Platform BKMs

The following are recommended platform reworks to enable the supported I/O drivers (GPIO, I^{2}C*, SPI, and HSUART) on Intel CRBs. This is not an exhaustive list of platform reworks.

4.1 How to Rework Bakersport Fab B USB3.0 Port

By default, the Bakersport Fab B CRB has an issue with the USB3.0 port. This port fails to read several types of USB3.0 thumb drives and couldn't achieve USB3.0 performance.

Note: Patriot Memory* 64 GB and EDGE Memory* DiskGO* 32 GB thumb drives are not recommended to be used in EHCI mode.

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

4.2 How to rework Bakersport Fab B I^{2}C* Port 6

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

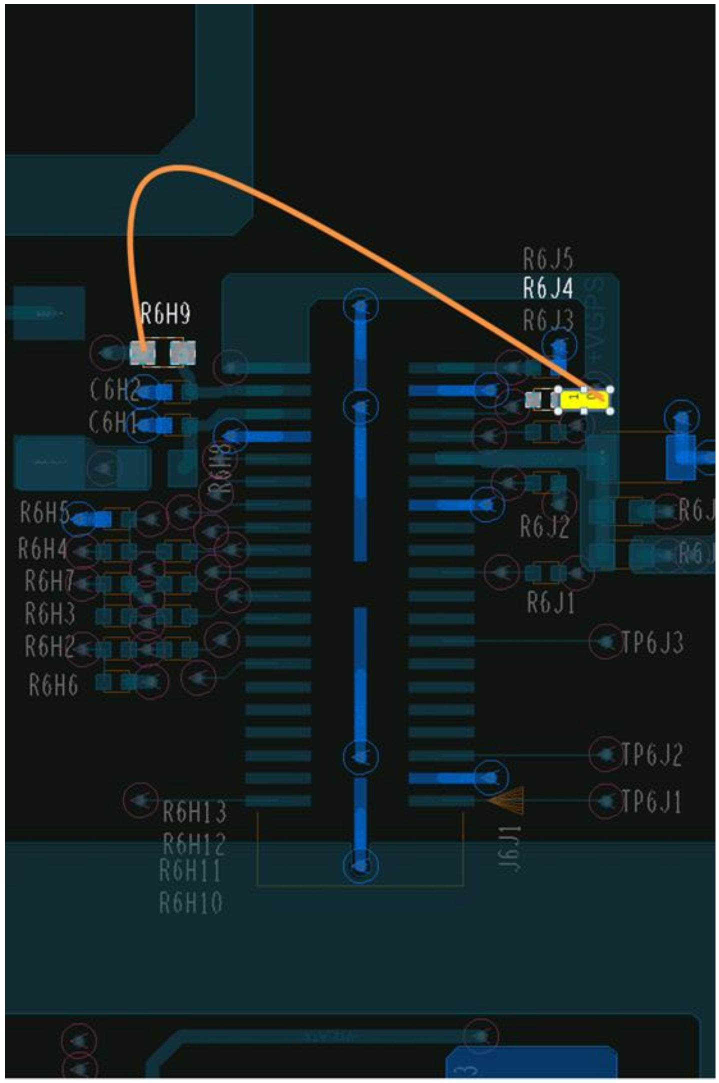
Affected Platform	Bakersport boards (PBA# G72250-200 Rev 02) (Fab B)
Rework Steps	1) UnStuff R5H9, R5H12, R5H8, R5H10 2) Stuff R5H4 (22 ohms) 3) Stuff R5H3 (22 ohms)

4.3 How to rework UART in Bakersport and Bayley Bay

By default, Bakersport Fab B CRB and Bayley Bay Fab 03 CRB have an issue with the UART2 port. This port triggers an unwanted interrupt. Add a 10 K resistor to mitigate this issue.

Rework Steps	1. Place a 10 K resistor followed by a 28 AWG wire from R6J4 to R6H9 Figure 4 shows the rework layout. The 10 K PU resistor (denoted by a yellow box) is wired (denoted by an orange curved line) to R6H9.
Affected Platform	Bakersport CRB (PBA# G72250-200 Rev 02) (Fab B) Bayley Bay Fab 3 CRB (IOTG configured) only

Figure 4. Bakersport CRB and Bayley Bay CRB UART Rework Layout

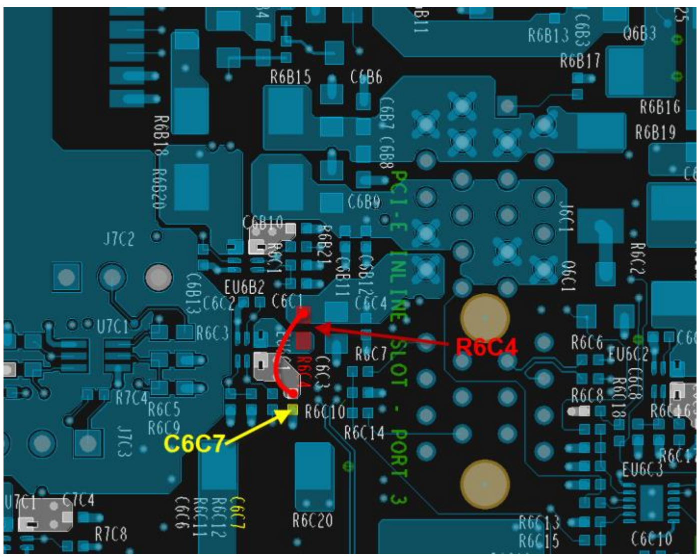


4.4 How to Rework Bayley Bay Fab 3 CRB PCIe* INLI Slot-Port 3

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

Affected Platform	Bayley Bay Fab 3 CRB (IOTG configured) only
Rework Steps	1. Remove R6C4 2. Add a jumper wire from C6C7 to R6C4 as shown in Figure 5.
Reasons for the Rework:	NICs are not recognized in the Windows* operating system if the jumper block (J7C2) is configured to Desktop mode, pins [1–2]. Failure mode occurs in PCIe Slot3.

Figure 5. Bayley Bay Fab 3 CRB PCIe* Slot 3 Rework



4.5 How to use Serial Port in Bayley Bay

The common serial port on the Bayley Bay CRB does not work. The actual serial port is the Micro USB port near the COM port on CRB board. Use the USB cable to connect the Micro USB port on the CRB board to the USB port on the host machine (laptop or desktop).

Install a driver from http://www.ftdichip.com/FTDrivers.htm to the host machine to have the virtual COM port on the host machine communicate with the Bayley Bay CRB.

§

5.0 Software Driver BKMs

5.1 How to Disable the DMA Feature for I^2C*

The seven I^2C* controllers in the Intel® Atom™ E3800 processor use the Windows* registry to control the DMA feature.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iaioi2c\
Parameters]

"ForceDma"="0,0,0,0,0,0,0"
```

ForceDma (string type) consists of seven values mapped to the seven I^2C controllers, which are device IDs from **0F41** to **0F47h**.

Value 0 disables the DMA, and I^2C data will be read and written in PIO mode.

For values other than 0, if data length is more than the specified value, I^2C data will be read and written in DMA mode; if data length is less than the specified value, I^2C data will be read and written in PIO mode.

By default, without any registry settings, I^2C will use the PIO mode.

5.2 How to Set the Baud Rates of HSUART

1. The baud rate is calculated based on the following method:

   Baud rate = (SourceClockFrequency) / (16 * divisor)

   Source Clock Frequency = 50000000 * PrescalerMValue /PrescalerNValue * 2

   For example, to set baud rate to 1M:

   Set PrescalerMValue = 64

   Set PrescalerNValue = 100

   SourceClockFrequency = 64,000,000

   The values of SourceClockFrequency, PrescalerMValue, and PrescalerNValue can be customized from the Windows registry. Reboot the system after setting these values.

2. To support baud rate between 230,400 and 3,686,400, create and change the following registry setting:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iaiouart
\Parameters]

   ;High speed source clock, M and N prescalers

   "HSUartSourceClockFrequency"=dword:01c1f8f8

   "HSUartPrescalerMValue"=dword:00003fff

   "HSUartPrescalerNValue"=dword:00006c80
```

3. To support baud rate between 300 and 115,200, change the following registry setting for low speed source clock, M and N prescalers:

```
"UartSourceClockFrequency"=dword:001c2000

"UartPrescalerMValue"=dword:0000025a

"UartPrescalerNValue"=dword:00007fff
```

Refer to Section 27.2.3, Baud Rate Generator in the *Intel® Atom™ processor E3800 Product Family Datasheet,* Doc# 538136, for details.

§