Intel® Atom™ Processor E3800 Windows* 8 IO Drivers

Programming Guide

February 2014

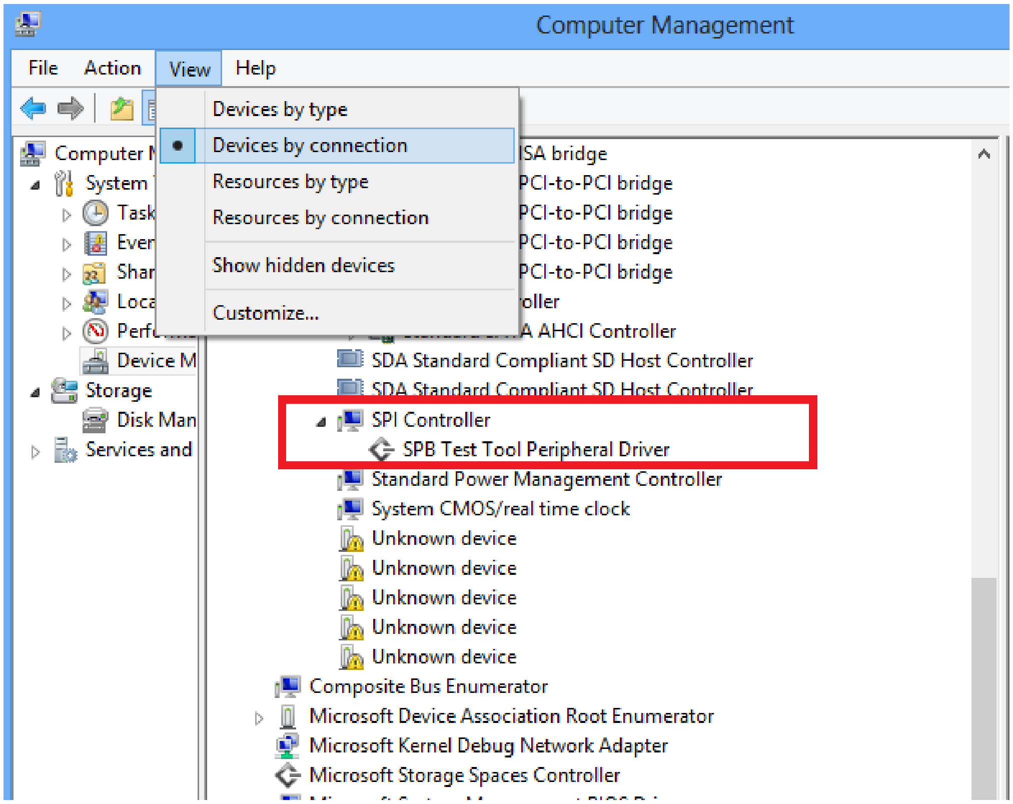
Revision 1.0
Software Release version: 2.0.0 Beta

Intel Confidential

Contents

1 Introduction

Microsoft Windows* 8 provided new frameworks for GPIO, I^2C, 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.