UP Board Firmware Settings to Enable SPI/I2C/GPIO/UART for Windows 10 IoT Enterprise

By Sean D. Liming and John R. Malin
Annabooks – www.annabooks.com

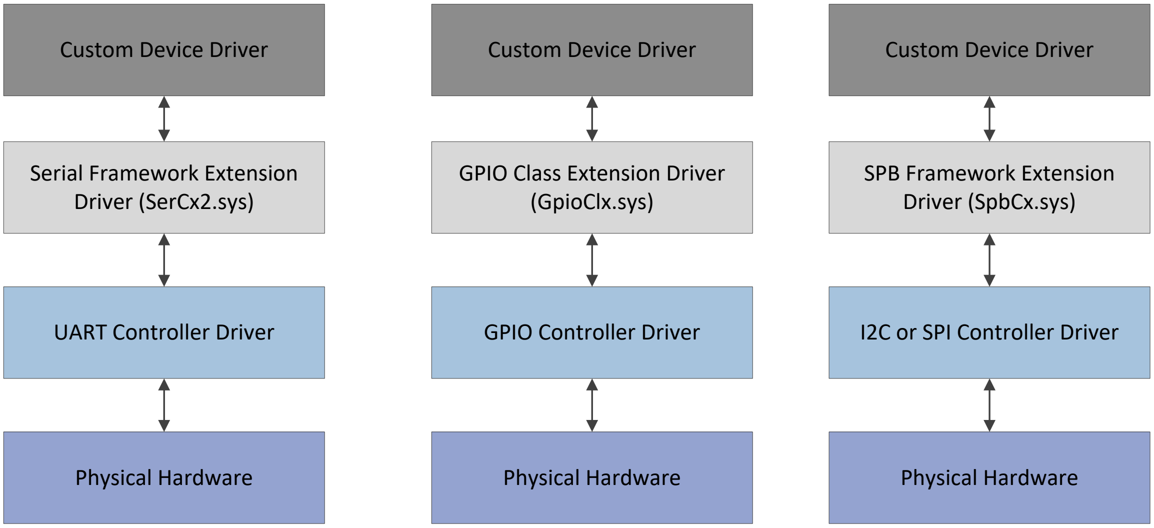
December 2020

Windows 10 Enterprise LTSC 2019
UP Board BIOS Version: UP-CHT01 R1.7 (UPC1DM17) 2/20/2019

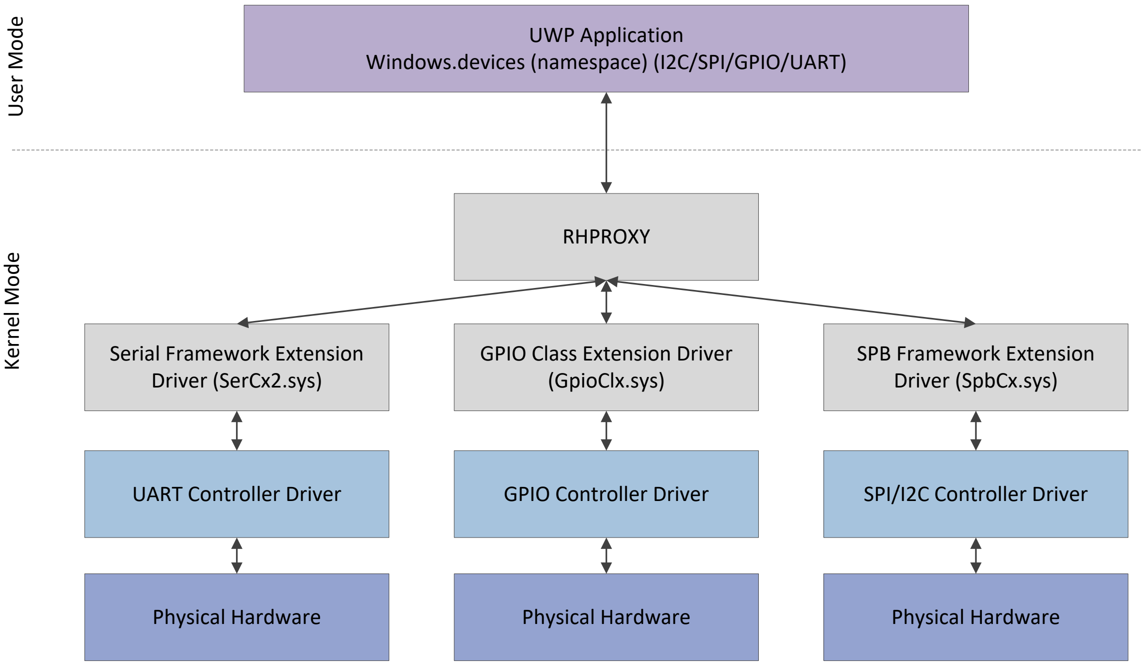
Back in January 2018, we released an article titled "Industrial PC Requirements to Access GPIO/I2C/SPI/UART from Windows 10 UWP Applications". The article discusses how Microsoft is requiring Resource HUB Proxy (RHPROXY) support in the firmware for UWP applications to interact with GPIO, I2C, GPIO, and UART. The UP Board was one of the platforms we investigated to see what was possible; but at the time, the firmware didn't have RHPROXY support. A few years later, the firmware has been updated to support RHPROXY and Windows IoT Core, so it is now possible to access the available GPIO, I2C, GPIO, and UART I/O on the 40-pin HAT connector. As there is not much documentation discussing the firmware setup, this article will cover the firmware settings that need to be set to access the I/O. Before we go further, it is a good idea to review RHPROXY.

Resource Hub Proxy (RHPROXY)

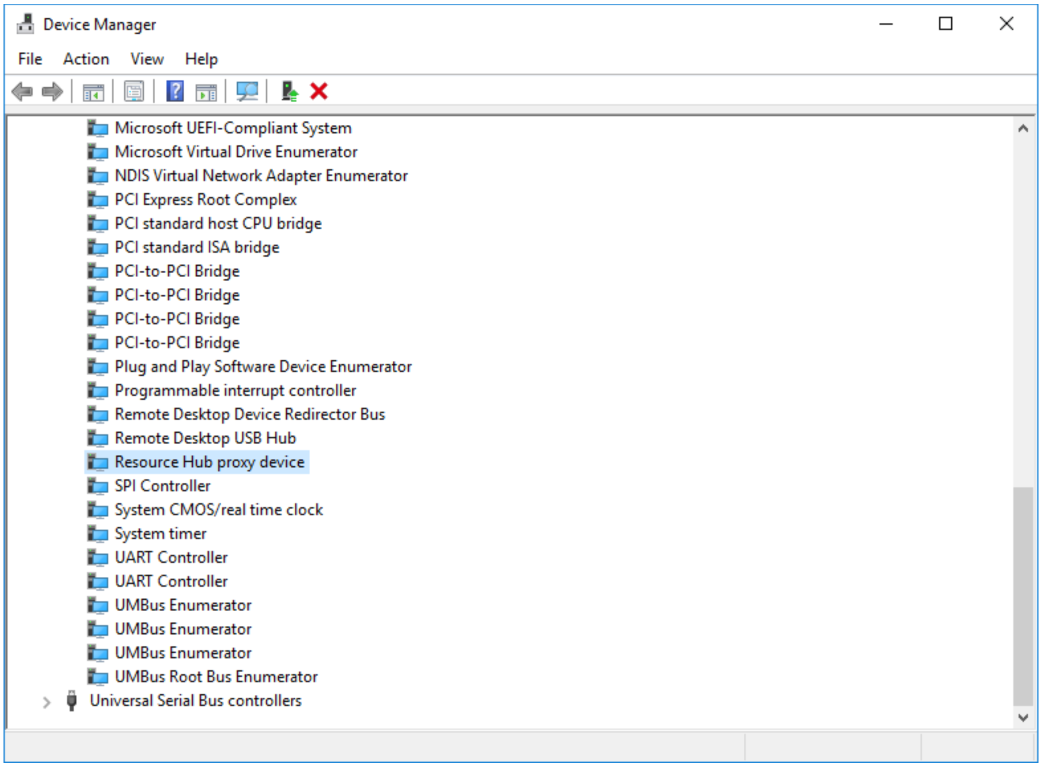
In Windows 8, three driver frameworks were created to support the different microcontroller IO. Mobile OEMs could write device drivers for dedicated devices attached to the IO.


Custom Device Driver
Custom Device Driver
Custom Device Driver
Serial Framework Extension Driver (SerCx2.sys)
GPIO Class Extension Driver (GpioClx.sys)
SPB Framework Extension Driver (SpbCx.sys)
UART Controller Driver
GPIO Controller Driver
I2C or SPI Controller Driver
Physical Hardware
Physical Hardware
Physical Hardware

The extension drivers were designed to make device driver development easier, but device driver development is tricky. Writing applications is much easier, thus using a cut-down Windows 8.1, the Windows for Devices program looked into a solution to access microcontroller IO from user-mode applications. The Resource Hub Proxy (RHPROXY) was developed to interact directly with the Framework Extension drivers to provide access to the microcontroller IO. Since the different IO can be wired to different devices, i.e. SPI can be used to program a firmware chip rather than be exposed to IO headers for general access. RHPROXY lets the board manufacturers define in their firmware what IO is available to the programmers. Setup is performed in the firmware as described in Guidelines to add RHPROXY support to the firmware is on the Windows Dev Center site.


User Mode
Kernel Mode
UWP Application
Windows.devices (namespace) (I2C/SPI/GPIO/UART)
RHPROXY
Serial Framework Extension Driver (SerCx2.sys)
GPIO Class Extension Driver (GpioClx.sys)
SPB Framework Extension Driver (SpbCx.sys)
UART Controller Driver
GPIO Controller Driver
SPI/I2C Controller Driver
Physical Hardware
Physical Hardware
Physical Hardware

RHPROXY is a driver that shows up under the System Devices in Device Manager.



Microsoft UEFI-Compliant System
Microsoft Virtual Drive Enumerator
NDIS Virtual Network Adapter Enumerator
PCI Express Root Complex
PCI standard host CPU bridge
PCI standard ISA bridge
PCI-to-PCI Bridge
PCI-to-PCI Bridge
PCI-to-PCI Bridge
PCI-to-PCI Bridge
Plug and Play Software Device Enumerator
Programmable interrupt controller
Remote Desktop Device Redirector Bus
Remote Desktop USB Hub
Resource Hub proxy device
SPI Controller
System CMOS/real time clock
System timer
UART Controller
UART Controller
UMBus Enumerator
UMBus Enumerator
UMBus Enumerator
UMBus Root Bus Enumerator
Universal Serial Bus controllers

UP Board BIOS Settings

When Windows 10 IoT Core support was added to the UP Board, the firmware was updated to support RHPROXY. The latest version, as of this writing, is posted on the top of this article. To enabled RHPROXY support, you have to set the operating system that is running. In the BIOS setup, set the following:

Boot -> OS Image ID: Windows 10 IoT Core

Since Windows 10 IoT Core and Windows 10 Enterprise LTSC 2019 (aka Windows 10 IoT Enterprise) are the same operating system, the setting applies to both. With the OS Image ID set, RHPROXY is enabled in the firmware. According to the UP Board specification, the I/O from the Intel Atom goes through a CPLD to the 40-pin HAT connector. The CPLD acts as a voltage level shifter so the 40-pin HAT connector has 3.3V tolerant pins. The HAT connector has two 3.3v out pins, two 5v out pins, and 8 ground (GND) pins. The rest of the 28 pins are for GPIO, I2C, UART, SPI, ADC, PWM, and SDIO.

There is UWP API support for I2C, UART, SPI, ADC, and PWM, but there is driver support only for I2C, UART, and SPI on the UP Board. What the firmware calls for pin numbers and IO names is different than what Windows will indicate. The differences can cause some confusion for the developers. Here is the explanation of the differences:

- The firmware lists the GPIO as 1-28, where Windows lists the GPIO as 0-27.
    - The GPIO must be configured for input or output, and one still needs to set the input or output drive mode for the corresponding pin in the Windows application. A mismatch between the firmware settings and the Windows application settings or the omission of either the firmware settings or the Windows application settings will result in an error.
- The firmware has I2C0 and I2C1, but Windows only supports the firmware's I2C0, which is called I2C1 in Windows.
- The UP Board specification calls pin 26 as SPI2_CS1N, but the firmware only shows this pin as GPIO23 rather than SPI2 chip select. This has been confirmed with Windows that pin 24 SPI2_CS0N is the only SPI chip select available.

In the **Advanced-> HAT Configuration** section of the firmware, there are two steps to set up the I/O. The first is to enable the serial I/O, so the shared GPIO on the pin is disabled. The table below lists the firmware options and the settings to enable the Serial I/O.

Firmware Option	**Setting**	**Notes**
LPSS I2C #1 Support	**ACPI**	I2C0
I2C #1 Speed	**400K**	I2C0 max speed
LPSS I2C #2 Support	**Disabled**	I2C1 – Not supported in Windows
LPSS PWM #1 Support	**Disabled**	Pin 32 No driver available
LPSS PWM #2 Support	**Disabled**	Pin 33 No driver available
LPSS SPI Support	**ACPI**	Enables SP2
LPSS HSUART Support	**ACPI**	Enables UART1
SCC SDIO Support	**Disabled**	Not supported
I2C0/GPIO Selection	**ISC0**	Need to set this so I2C is enabled
I2S/GPIO Selection	**GPIO**	There is no I2S support in Windows
HAT Output Enabled	**Enabled**	This is the master switch for the CPLD to enable the HAT I/O
LED2	**Off**	Unknown
LED3	**Off**	Unknown
LED4	**Off**	Unknown
ADC	**Disabled**	Pin 7 No driver available

If SPI, UART, I2C along with ADC, SDIO, I2S, and PWM were disabled, all 28 of the GPIO pins will be GPIO pins in Windows. Once SPI, UART, and I2C are enabled, the pins for these I/O are dedicated to the serial devices and cannot be used for GPIO.

The second step is to configure the GPIO. In the **Advanced-> HAT, Configuration-> GPIO Configuration In Pin Order**, you can set any enabled GPIO for input or output. If set to output the initial state of Low (Logic 0 / 0V) or High (Logic 1 / 3.3v) can also be set.

If the above table has the settings to enable I2C0, SPI, and UART1, the following table for the 40-pin HAT header shows the I/O in **bold** that is available for Windows UWP applications.

OS-Serial I/O	OS-GPIO	BIOS	UP Board Documentation	HAT Header		UP Board Documentation	BIOS	OS-GPIO	OS-Serial I/O
				Odd Pin	Even Pin				
			3.3V	1	2	5V			
	GPIO0	GPIO1	GPIO1/~~I2C1_SDA~~	3	4	5V			
	GPIO1	GPIO2	GPIO2/~~I2C1_SCL~~	5	6	GND			
	GPIO2	GPIO3	GPIO3/ ~~ADC~~	7	8	GPIO16/UART1-TX	GPIO16	GPIO15	**UART1-TX**
			GND	9	10	GPIO17/UART1-RX	GPIO17	GPIO16	**UART1-RX**
UART1-RTS	GPIO3	GPIO4	GPIO4/UART1_RTS	11	12	GPIO18	GPIO18	**GPIO17**	
	GPIO4	GPIO5	GPIO5/~~SDIO~~	13	14	GND			
	GPIO5	GPIO6	GPIO6/~~SDIO~~	15	16	GPIO19 /~~SDIO~~	GPIO19	**GPIO18**	
			3.3V	17	18	GPIO20/~~SDIO~~	GPIO20	**GPIO19**	
SPI2-MOSI	GPIO6	GPIO7	GPIO7/SPI2-MOSI	19	20	GND			
SPI2-MISO	GPIO7	GPIO8	GPIO8/SPI2-MISO	21	22	GPIO21 /~~SDIO~~	GPIO21	**GPIO20**	
SPI2-CLK	GPIO8	GPIO9	GPIO9/SP2-CLK	23	24	GPIO22/SPI2-CS0N	GPIO22	GPIO21	**SPI2-CS0**
			GND	25	26	GPIO23	GPIO23	**GPIO22**	
I2C1_SDA	GPIO9	GPIO10	GPIO10/I2C0_SDA	27	28	GPIO24/I2C0_SCL	GPIO24	GPIO23	**I2C1_SCL**
	GPIO10	GPIO11	GPIO11	29	30	GND			
	GPIO11	GPIO12	GPIO12	31	32	GPIO25/~~PWM0~~	GPIO25	**GPIO24**	
	GPIO12	GPIO13	GPIO13/~~PWM1~~	33	34	GND			
	GPIO13	GPIO14	GPIO14	35	36	GPIO26/UART1_CTS	GPIO26	GPIO25	**UART1-CTS**
	GPIO14	GPIO15	GPIO15/~~SDIO~~	37	38	GPIO27 / ~~I2S-DI~~	GPIO27	**GPIO26**	
			GND	39	40	GPIO28 / ~~I2S-DO~~	GPIO28	**GPIO27**	

Note: The I/O that is crossed out is not available for Windows at the time of this writing.

Once you save the changes and boot into Windows, Device Manager will show RHPROXY support, and UWP applications can be developed to access the I/O.

Summary

GPIO, I2C, SPI, PWM, ADC, and UART have been standard IO on microcontrollers for decades. Many microcontrollers have these IO features multiplexed on pins, so care should be taken by the developer to properly configure these multiplexed pins. Intel added support for these microcontroller I/Os on their latest Atom™ processors. It is up to the board manufacturer to make these I/Os available to the developer. For Windows support, the firmware has to support RHPROXY.