# Industrial PC Requirements to Access GPIO/I2C/SPI/UART from Windows 10 UWP Applications

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

January 2018

Windows 10 Version 16299 Build 1709

GPIO, I2C, SPI, PWM, ADC, and UART have been standard IO on microcontrollers for decades. Many microcontrollers have these IO features multiplexed on pins, which gives the developer flexibility to configure the microcontroller for a specific application. SoC processors, like those based on ARM, integrated microcontroller IO to support mobile applications. For the PC platform, these IO were only available as add-ons in the form of USB or PC bus peripheral devices, which came with custom device drivers and API libraries. Integration of microcontroller IO into Intel Architecture processors started with the Intel Atom® processor 10 years ago and has been expanded in the Intel Atom family with each new generation. Intel calls this the Low Power Sub System (LPSS). Intel Atom developer boards like the MinnowBoard and Intel Joule platforms expose these IO to headers for users to directly access and write applications. Other Industrial PC boards expose these IO differently.

Historically, support for microcontroller IO in Windows has been made available in a variety of formats. Windows CE and the short lived .Net Micro Framework had API support for these IO built in, but support in Windows desktop was only available through custom drivers and API sets. As operating system development merged within Microsoft starting 10+ years ago, access to microcontroller IO was added in stages:

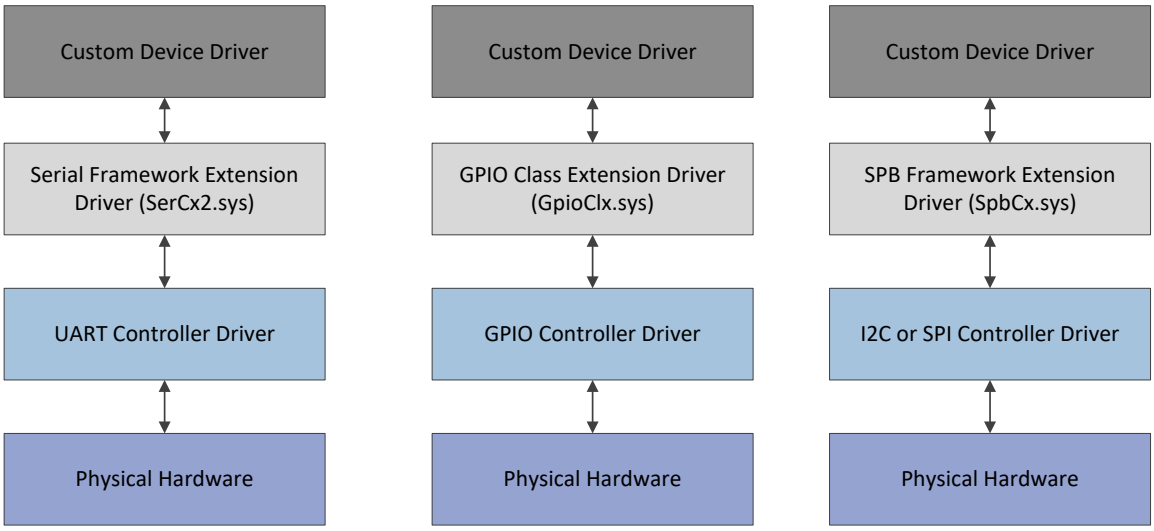
- First, Windows CE mobile was dropped as the mobile operating system in favor of Windows 8 running on ARM processors. Mobile hardware developers needed a solution to interface to peripherals attached to microcontroller IO. New driver frameworks were created so custom drivers could easily be written for dedicated devices connected to GPIO, I2C, SPI, and UART. OEMs could then access the data through standard C/C++ DLL API calls.
- Next, Windows 8.1 took the next step with the “Windows for Devices” side project, which had a cut down version of Windows 8.1 running on the Intel Galileo board. Only a single Universal Windows Platform (UWP) application could have control of the GUI interface. Since UWP applications run in a sandbox with limited access to outside resources, access through a DLL was not possible. Also many platforms exposed the IO to headers so applications could easily be programed to access different peripherals. A solution was needed to allow user mode applications to access the microcontroller IO directly. A resource hub and API set were created to bridge from UWP applications to GPIO, I2C, SPI, and UART ports.
- Finally, Windows 10 was launched as the single operating system solution that reaches a wide range of devices from simple IoT platforms to virtual / augmented reality platforms. There are different SKUs of Windows 10 from IoT Core, the smallest footprint, to Windows 10 Enterprise on the desktop. User mode application access to GPIO, I2C, SPI, and UART was only made available for Windows 10 IoT Core. The Raspberry Pi 2/3 and the MinnowBoard Max were the first platforms to support Windows 10 IoT Core and the API to access microcontroller IO. This allowed time to flush out any errors in the API and resource hub solution. For Windows desktop, user mode application support for GPIO, I2C, SPI, and UART was not made available until the release of Windows 10 version 16299, build 1709.

Now that there is user mode access, how will microcontroller IO be supported on different Industrial PC platforms? There are many Industrial PC boards with the Intel Atom E3800 family (Bay Trail), and there are newer platforms featuring the next generation Intel Atom Processors

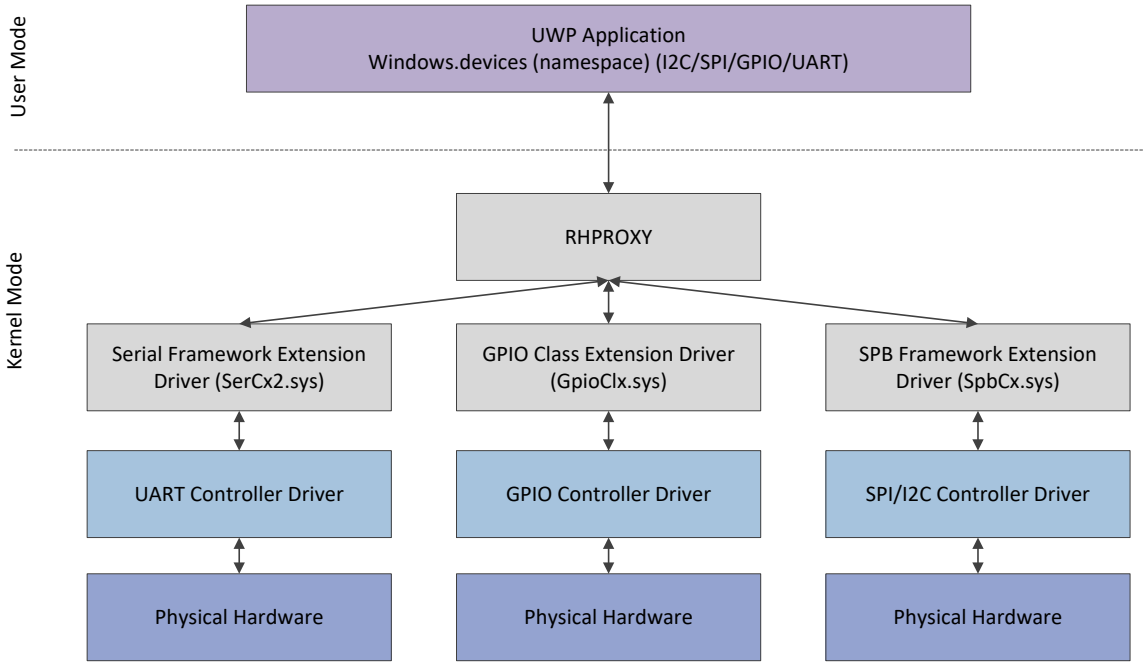
(Braswell / Cherry Trail: N3000, x5-E8000, x5-Z8350, and Apollo Lake: E3900) coming on the market. Each board exposes microcontroller IO differently. With the release of Windows 10 16299, we were able to obtain three Intel Atom platforms (x5-8350, N3350, and E3940) and the Minnowboard Max (E3800) to investigated IoT Core and IoT Enterprise to see if GPIO, I2C, SPI, and UART were accessible. We quickly found that even though the microcontroller IO is made available on the platform, it doesn't mean you can access the IO from a UWP application. There are two key pieces to gain access. The first piece comes from the Industrial PC board vendor with additional support being made in the firmware. The second piece comes with installing the correct device drivers.

## *Resource Hub Proxy (RHPROXY)*

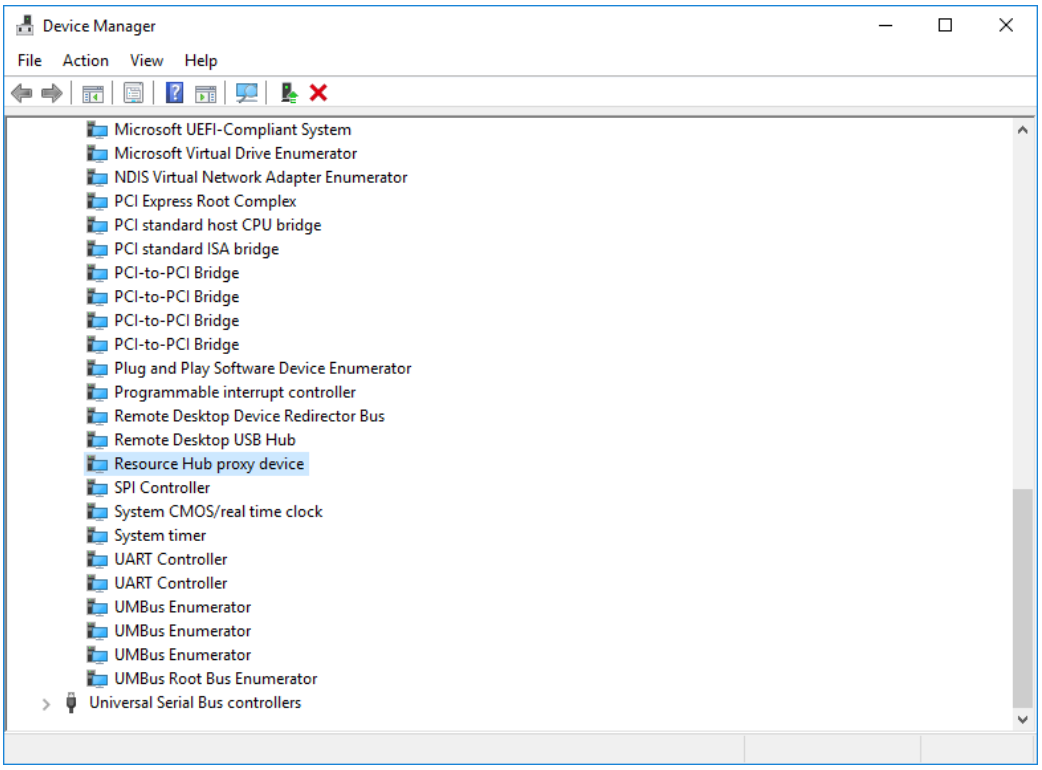
In Windows 8, three driver frameworks were created to support the different microcontroller IO. Mobile OEMs could write device drivers for dedicated devices attached to the IO.



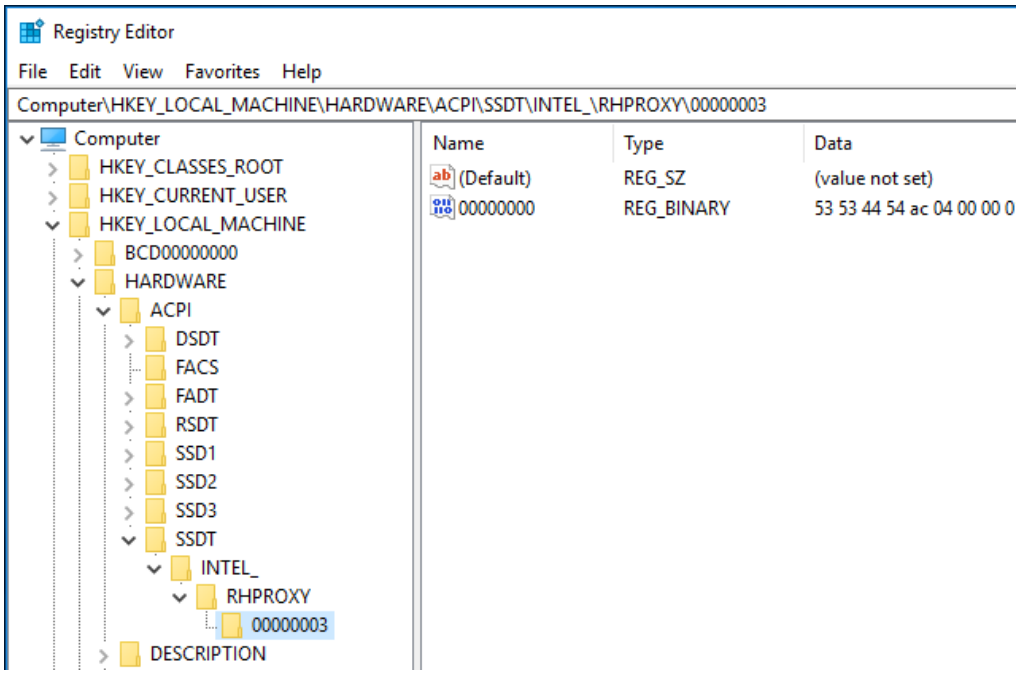

The extension drivers were designed to make device driver development easier, but device driver development is tricky. Writing applications is much easier, thus using a cut down Windows 8.1, the Windows for Devices program looked into a solution to access microcontroller IO from user mode applications. 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 used to program firmware chip rather than be exposed to IO headers for general access, RHPROXY lets the board manufacturers define in the 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.



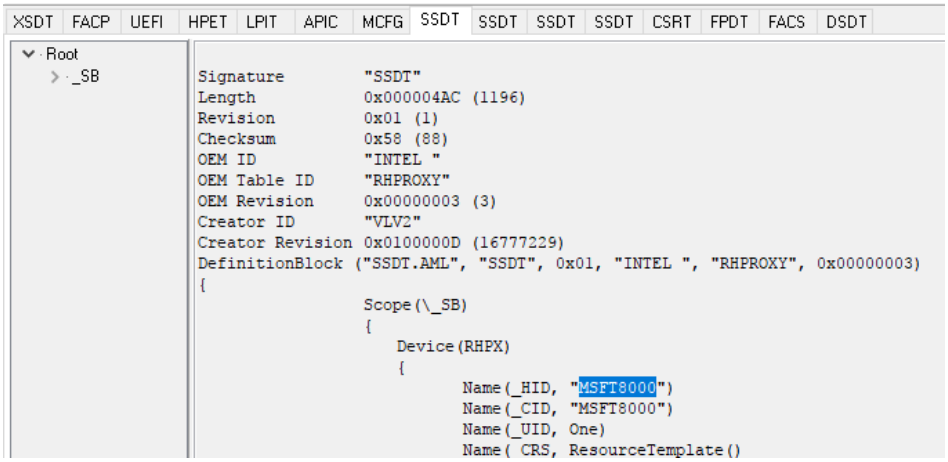

RHPROXY is a driver that shows up under the System Devices in Device Manger.




For the driver to load, the firmware must list “ACPI\MSFT8000” in the ACPI tables. In the Minnowboard Max, RHPROXY shows up in a Secondary System Description Table (SSDT). There are several options to check for RHPROXY support in your PC platform, the first is to install Windows 10 16299 and see if RHPROXY shows up in device manager or look in the registry under HKLM\HARDWARE\APCI\.



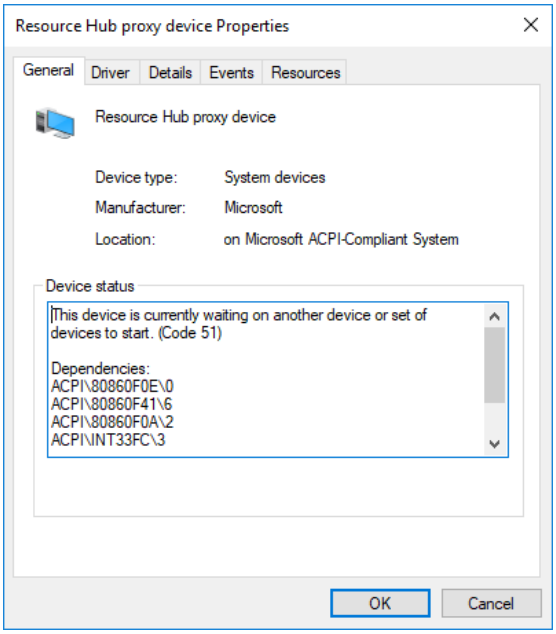

The other option is to use tools like the IASL tools (https://www.acpica.org/) or R/W Everything (http://rweverything.com/) to read the ACPI tables.




For the other Intel Atom platforms we tested, RHPROXY was not supported in the firmware, thus UWP application access to the microcontroller IO was not possible. Following the afore-mentioned guideline, one could add the support to the registry, but intimate knowledge of the hardware is required, which might not be possible in most cases. If you want to access GPIO, I2C, SPI, and UART from a UWP application, you will have to work with your hardware vendor to get support in the firmware.

### *Correct GPIO/I2C/SPI/UART Device Driver Support*

The second piece is installing the correct device drivers. After installing Windows Enterprise 16299 on the MinnowBoard Max, RHPROXY shows up with errors. Device drivers for GPIO, I2C, SPI, and UART are needed for RHPROXY to run correctly.




The Hardware IDs are defined in the firmware in the Differentiated System Description Table (DSDT). Every platform will have different Hardware IDs for the microcontroller IO. For the Minnowboard Max, the Hardware IDs for the IO are as follows:

| Hardware IDs | Driver Name |
|---|---|
| ACPI\80860F0E\0 | SPI Controller |
| ACPI\80860F41\6 | I2C Controller |
| ACPI\80860F0A\2 | UART Controller |
| ACPI\80860F0A\1 | UART Controller |
| ACPI\INT33FC\3 | GPIO Controller |
| ACPI\INT33FC\1 | GPIO Controller |

The GPIO, I2C, SPI, and UART drivers that loaded with Windows 10 Enterprise 16299 desktop had the correct Hardware IDs, but they were not the correct versions for use with RHPROXY. For example, the SPI controller showed up as the “Intel(R) Atom/Celeron/Pentium Processor SPI Controller”, and there wasn’t a driver file associated with the listing. Checking the INF files for other IO drivers showed they were built with the framework extension, but they still didn’t work.

```
:
:
; -------------- Driver install section
[i2c.AddService]
DisplayName    = %Driver_Service.Desc%
ServiceType    = 1                ; SERVICE_KERNEL_DRIVER
StartType      = 3                ; SERVICE_DEMAND_START
ErrorControl   = 1                ; SERVICE_ERROR_NORMAL
ServiceBinary  = %12%\iaioi2c.sys
Dependencies   = SpbCx
AddReg         = i2c_Service_addReg
:
:
```

Intel released a BSP for the E3800 family called the Intel® Embedded Drivers for Microsoft Windows* 10 IoT Core. Once these drivers were loaded, RHPROXY ran as expected. The table below lists the processor and Intel Atom BSPs for Windows 10 IoT Core.

| Intel Atom® Processor | Windows 10 IoT Core BSP Website |
|---|---|
| E3900 (Apollo Lake) | https://www.intel.com/content/www/us/en/embedded/products/apollo-lake/technical-library.html |
| N3000, x5-E8000 and x5-Z8350 (Braswell / Cherry Trail) | https://www.intel.com/content/www/us/en/embedded/products/braswell/software-and-drivers.html |
| Intel Atom® E3800 (Bay Trail) | https://downloadcenter.intel.com/download/25618/Intel-Embedded-Drivers-for-Microsoft-Windows-10-IoT-Core-32-bit-and-64-bit-MR1 |

Although RHPROXY wasn't supported on the other Intel Atom platforms, the Intel BSP for Windows 10 IoT Core drivers for their respective processors installed without issue. Even though application access through RHPROXY is not support, OEMs could still write device drivers that interact with the framework extensions to access peripherals connected to the IO. Of course, this means close working knowledge of the hardware to make sure you are accessing the correct port. More information can be found online:

| IO | Driver Framework Information |
|---|---|
| GPIO | https://docs.microsoft.com/en-us/windows-hardware/drivers/gpio/ |
| SPI/I2C | https://docs.microsoft.com/en-us/windows-hardware/drivers/spb/ |
| UART | https://msdn.microsoft.com/en-us/library/windows/hardware/dn265309(v=vs.85).aspx |

## *Some Notes*

We tested a few UWP IO applications on Windows Desktop to see if they could access GPIO, I2C, SPI, and UART. The applications were successfully tested in Windows 10 IoT Core. GPIO, SPI, and I2C worked as expected. The only device we were not able to access in Windows Desktop was the UART. A USB-to-serial device was successfully accessed.

Industrial PC platforms have come with multiple COM ports that use the serial.sys driver. UWP applications cannot directly access standard COM ports that use serial.sys. Windows Form applications could access COM ports just fine. The differences between UART and COM port is discussed in the Differences Between SerCx2.sys and Serial.sys MSDN article.

Most OEMs use Windows 10 IoT Enterprise (aka Windows 10 Enterprise LTSB 2016 version 14393). User mode access to microcontroller IO is not available in this release, but most likely will be made available in the Windows 10 LTSC (LTSB to be renamed to LTSC) in 2019. If you choose to use version 16299 now in your device, you will be on the CBB (to be renamed to SAC) license and would have to do an OS version update once a year, which is not ideal.

## *Summary*

With access to microcontroller IO now available to UWP applications on Windows desktop version 16299, OEMs have a choice between Windows IoT Core and Windows Enterprise to run on Intel Atom processors. Choosing a board that supports user mode access (RHPROXY) will take some close interaction with the hardware vendor. The techniques and tools provided in this paper to check out if RHPOXY has been added to the ACPI tables will help with the hardware search. Industrial PC board manufacturers should follow the online guidelines to add RHPROXY if they want to support Windows 10 IoT Core and UWP application access. RHPROXY is not the only piece to the puzzle. The correct device drivers are also needed to complete the solution.