

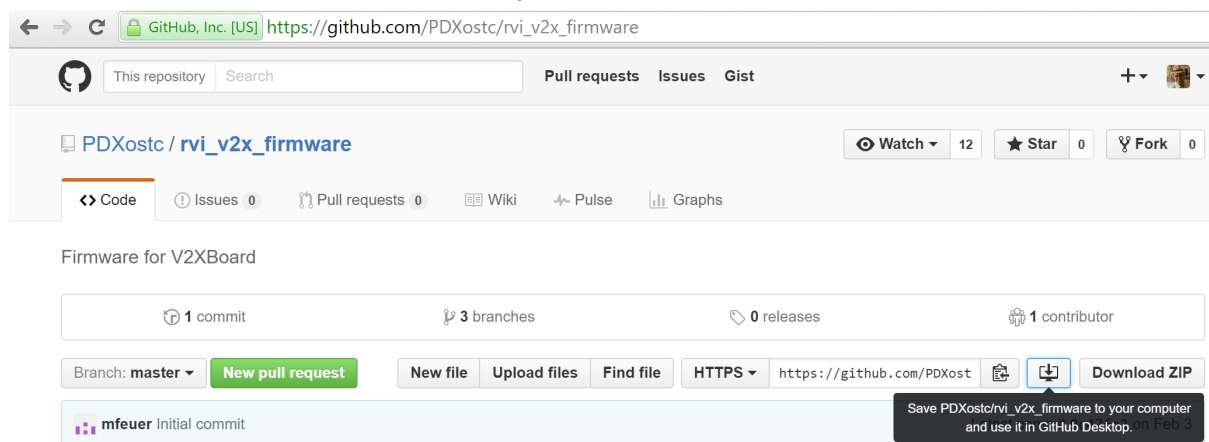
V2X Programmer's Guide

Brief: How to get the V2X project setup and ready to edit on your computer.

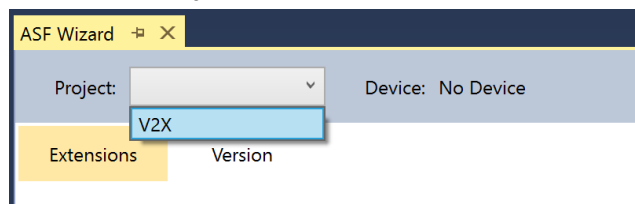
1. Go to <http://www.atmel.com/tools/ATMELSTUDIO.aspx>

Download the offline installer as a Guest if involvement is short. An account is not complicated if you will work on the project for some time. You will have to provide your email to get the installer (a link is sent).

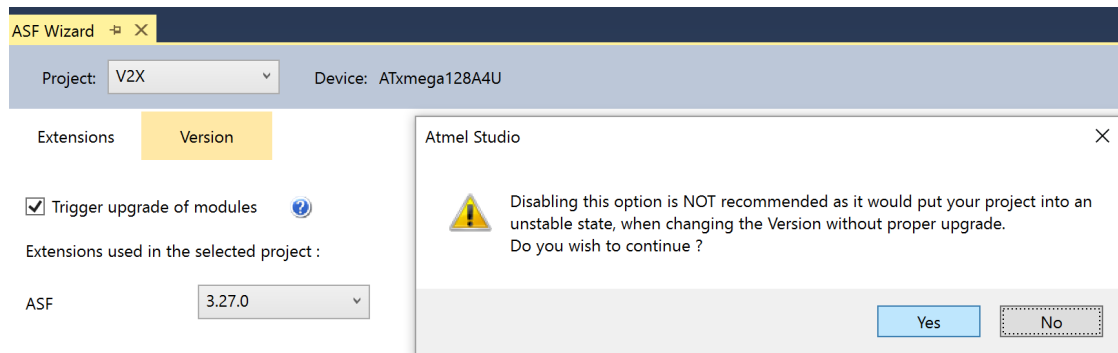
2. Install Atmel Studio 7 and the USB drivers. This will require running the installer with “run as administrator” privileges. Be sure to use “Advanced view” when the question is asked.
3. GIT Clone the repo to a local project folder.



4. Switch to Develop branch of git repo.
5. Navigate into the repo \rvi_v2x_firmware\V2X_Firmware and open the V2X.cproj file.
6. Once the program launches, the ASF wizard should be displayed, select “V2X” from the Project pull down.



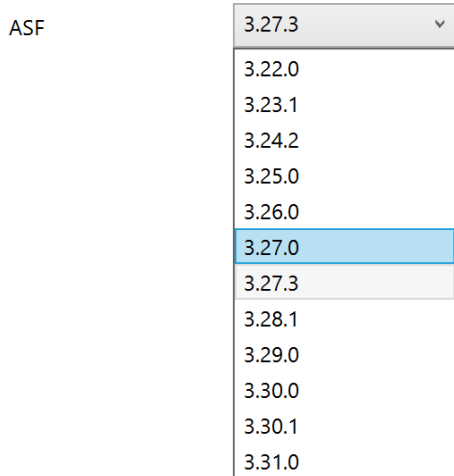
7. Click the version tab. Uncheck the Trigger upgrade box. A dialog will launch for confirmation.



8. Downgrade the project by one ASF version.

☐ Trigger upgrade of modules

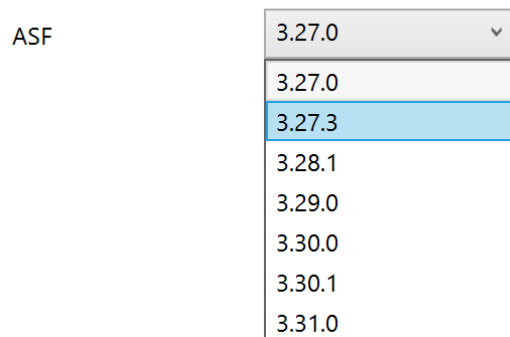
Extensions used in the selected project :



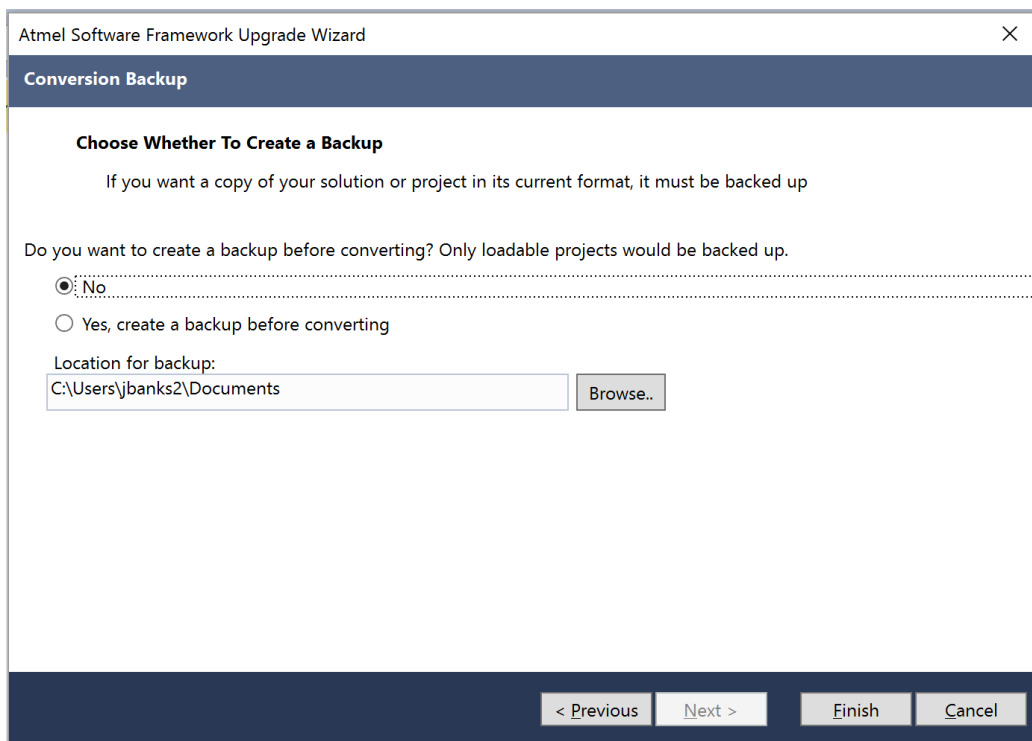
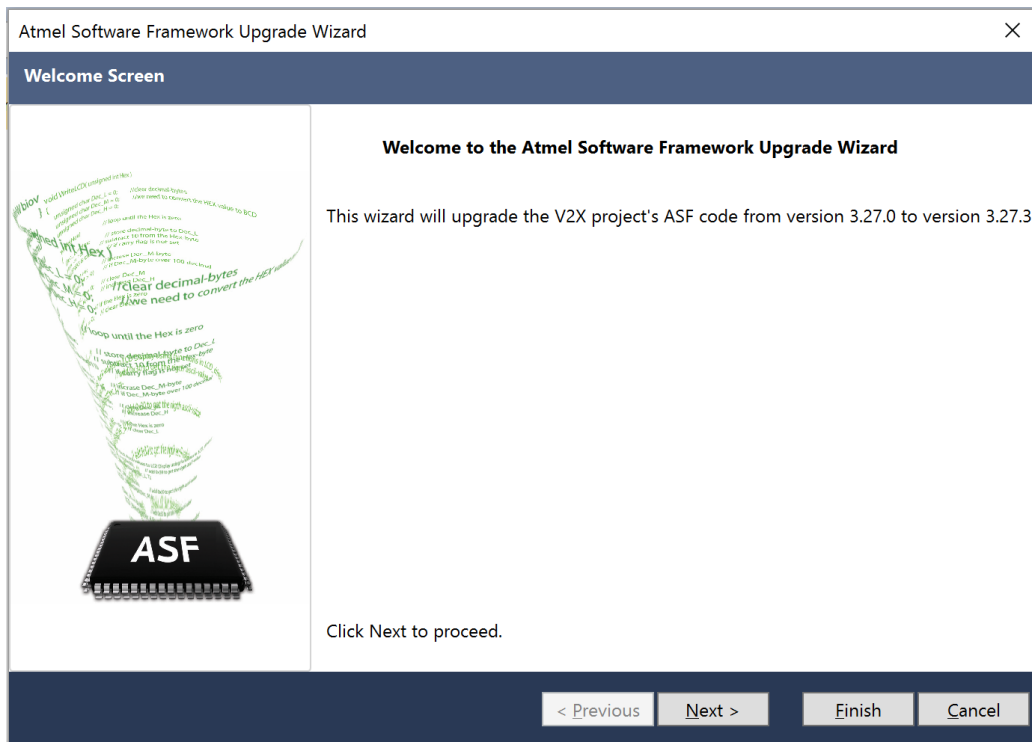
9. Check the Trigger Upgrade box then return the project to the current version. Click Upgrade.

☒ Trigger upgrade of modules

Extensions used in the selected project :



10. Follow the wizard, click “Next,” select “No,” click Finish. After the upgrade, there is no need to check the log. Click OK.



11. Navigate back to the Extension tab. Verify these modules are contained in the project:

11.1. If not present, select corresponding module from the “Available modules” field in the left pane

- 11.1.1. AC - Analog Comparator
- 11.1.2. ADC - Analog to Digital converter
- 11.1.3. Calendar Functionality (service)
- 11.1.4. Chip Reset Cause Access (Common API) (service)
- 11.1.5. CPU specific features (driver)
- 11.1.6. Delay routines (service)
- 11.1.7. DMA - Direct Memory Access (driver)
- 11.1.8. GPIO - General purpose Input/Output (service)
- 11.1.9. Interrupt management (Common API) (driver)
- 11.1.10. IOPORT - General purpose I/O Service (service)
- 11.1.11. IOPORT - Input/Output Port Controller (driver)
- 11.1.12. Memory Control Access Interface (service)
- 11.1.13. NVM - Non Volatile Memory (driver)
- 11.1.14. NVM - Non volatile memory access (CommonAPI) (Driver)
 - 11.1.14.1. No_extmem
- 11.1.15. PMIC - Programmable Multi-level Interrupt Controller (driver)
- 11.1.16. PWM service using timer/counter (service)
- 11.1.17. RTC - Real Time Counter (driver)
- 11.1.18. Serial I/O - Host (component)
 - 11.1.18.1. USB
- 11.1.19. Sleep Controller driver (driver)
- 11.1.20. Sleep manager (service)
- 11.1.21. SPI - Serial Peripheral Interface (driver)
- 11.1.22. SPI - Serial Peripheral Interface Master (Common API) (service)
 - 11.1.22.1. Standard_spi
- 11.1.23. Standard serial I/O (stdio) (driver)
- 11.1.24. System Clock Control (service)
- 11.1.25. TC - Timer Counter (driver)
- 11.1.26. USART - Serial interface (service)
- 11.1.27. USART - Universal Synchronous/Asynchronous Receiver/Transmitter (driver)
- 11.1.28. USB Device (service)
 - 11.1.28.1. Composite
- 11.1.29. WDT - Watchdog timer (service)

11.2. Click “ADD>>”

Note: Likely all necessary the modules will be in the “Selected Module” pane and only the sub-selection of “USB Device” and “Serial I/O - Host” will be incorrect.

Selected Modules

Generic board support (driver)

AC - Analog Comparator (driver)

ADC - Analog to Digital Converter (driver)

Calendar functionality (service)

Chip Reset Cause Access (Common API) (service)

CPU specific features (driver)

Delay routines (service)

DMA - Direct Memory Access (driver)

GPIO - General purpose Input/Output (service)

Interrupt management (Common API) (driver)

IOPORT - General purpose I/O service (service)

IOPORT - Input/Output Port Controller (driver)

Memory Control Access Interface (service)

NVM - Non Volatile Memory (driver)

NVM - Non Volatile Memory access (Common API) (driver) no_extmem

PMIC - Programmable Multi-level Interrupt Controller (driver)

Selected Modules

NVM - Non Volatile Memory access (Common API) (driver) no_extmem

PMIC - Programmable Multi-level Interrupt Controller (driver)

PWM service using timer/counter (service)

RTC - Real Time Counter (driver)

Serial I/O - Host (component) usb

Sleep Controller driver (driver)

Sleep manager (service)

SPI - Serial Peripheral Interface (driver)

SPI - Serial Peripheral Interface Master (Common API) (service) standard_spi

Standard serial I/O (stdio) (driver)

System Clock Control (service)

TC - Timer Counter (driver)

USART - Serial interface (service)

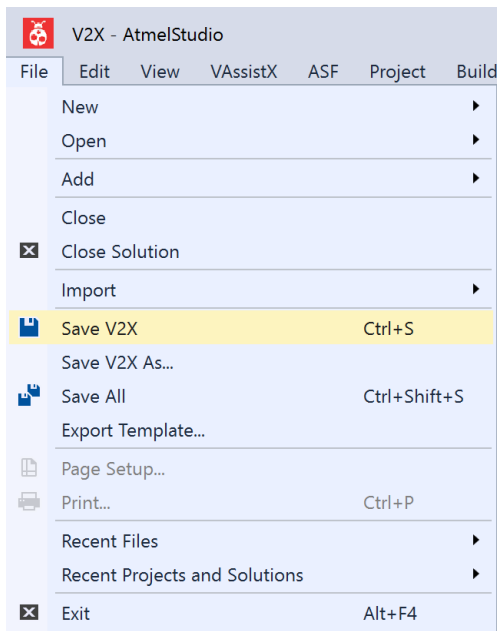
USART - Universal Synchronous/Asynchronous Receiver/Transmitter (driver)

USB Device (service) composite

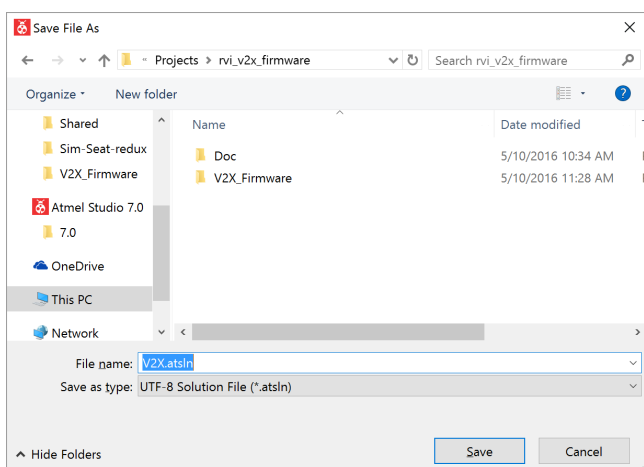
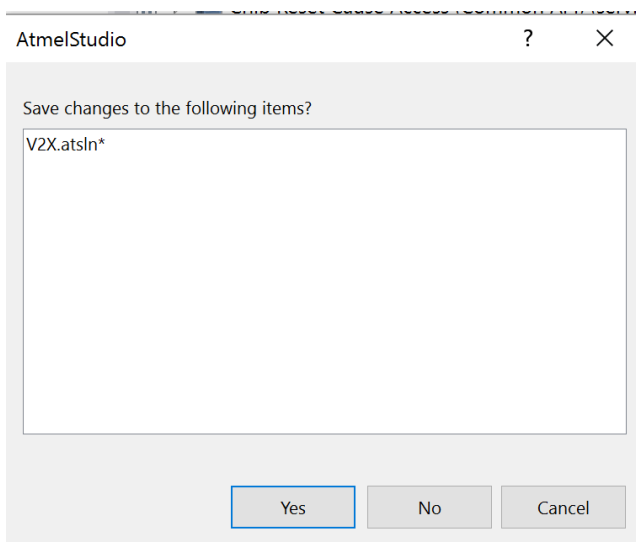
WDT - Watchdog Timer (driver)

12. Click “Apply”, then “Ok” when a confirmation dialog appears.




13. Save the project then Exit the tool.



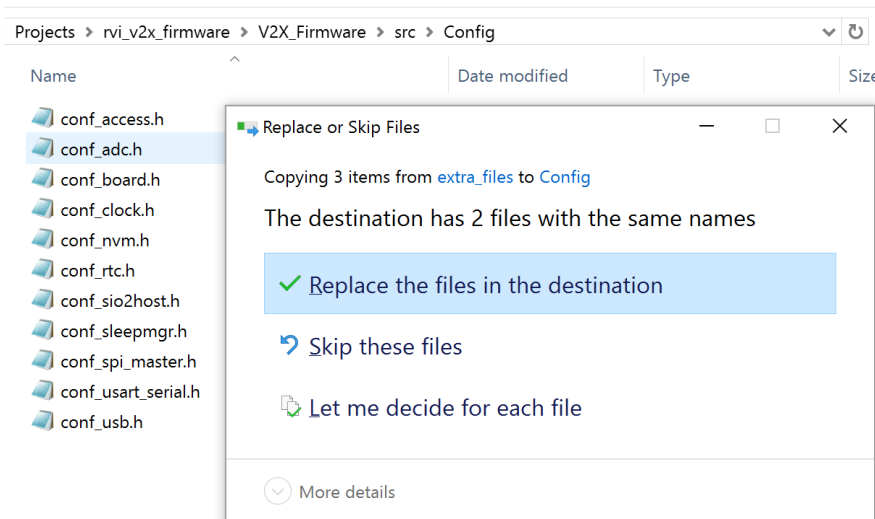
14. A popup dialog will ask to save the project. Click Yes. Click Save. the application will exit.



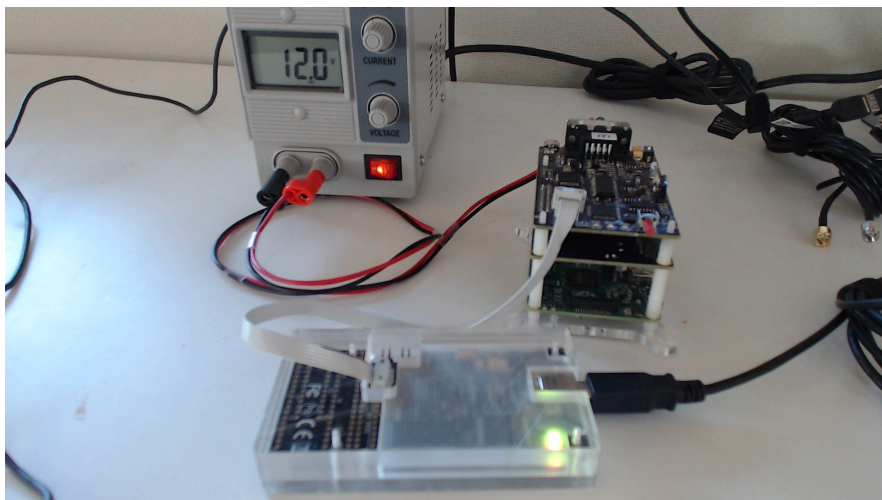
15. Navigate to `\rvi_v2x_firmware\V2X_Firmware\src\V2X\extra_files` copy the 3 files.

| Projects > rvi_v2x_firmware > V2X_Firmware > src > V2X > extra_files | |
|--|-------------------|
| Name | Date modified |
|  conf_board.h | 5/10/2016 2:29 PM |
|  conf_clock.h | 5/10/2016 2:29 PM |
|  conf_usb.h | 5/10/2016 2:29 PM |

16. Navigate to `\rvi_v2x_firmware\V2X_Firmware\src\V2X\extra_files` paste the 3 files overwriting any conflicts.

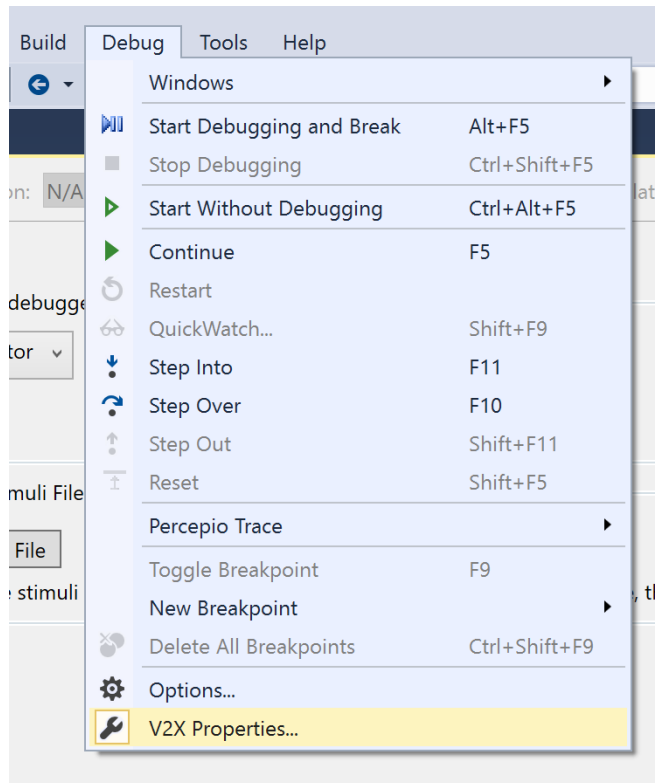


17. Launch the the V2X.atsln project file in `\rvi_v2x_firmware\V2X_Firmware`. Atmel Studio will start. The ASF wizard tab can be closed. The project should successfully build (F7).
18. Connect the V2X hardware to dragon, power and USB

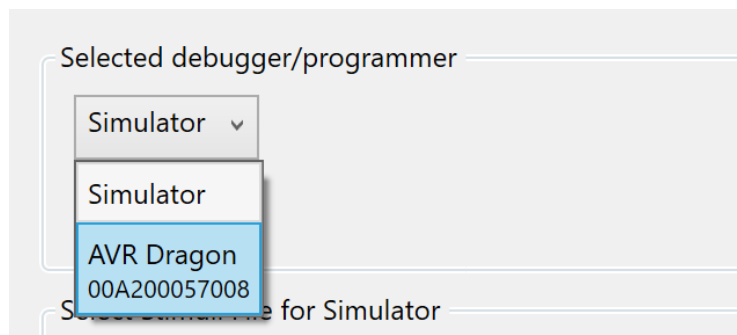


19. Plug in the AVR Dragon programmer by USB (drivers should auto install)

20. Use the menu Debug -> V2X Properties...

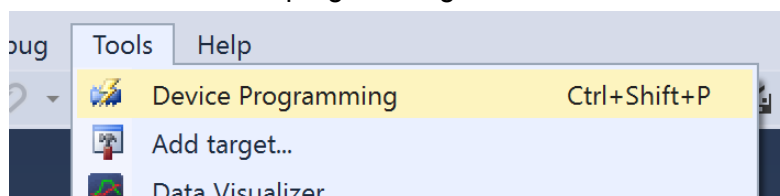


21. If the dragon installed properly it should appear in the list. Select it, click save from the buttons along the top. Close the tab.

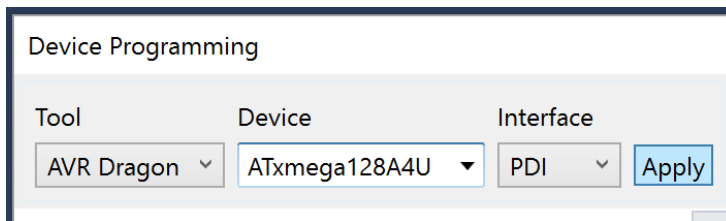


22. Verify the V2X hardware is powered on. If the V2X has never been installed before you will need to hold the button down while any programming operations are performed.

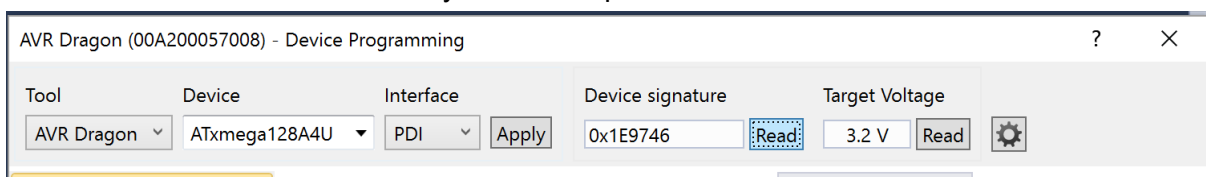
23. Select Device programming from the Tools menu.



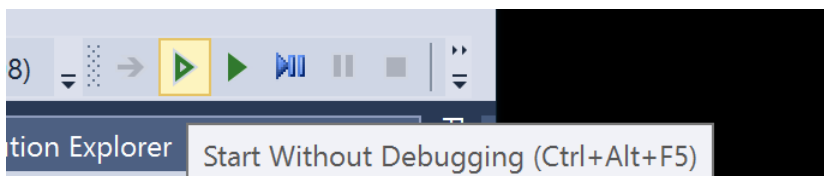
24. Select the Dragon as the device, select the ATxmega128A4U from the drop down, PDI interface, then click Apply”



25. Hold the V2X button down and click device signature. If the chip is connected properly and operating the ID value 0x1E9746 will populate the Device Signature field and the Target Voltage field will also populate. If this test passes, close the window and move on. Failure discussion is beyond the scope of this document.



26. Install the firmware to the V2X hardware. Again, you may need to hold the V2X button down to install the firmware



27. The V2X should now mount a USB2 hub, 3 ports labeled V2X and 5 ports labeled SIM5320.