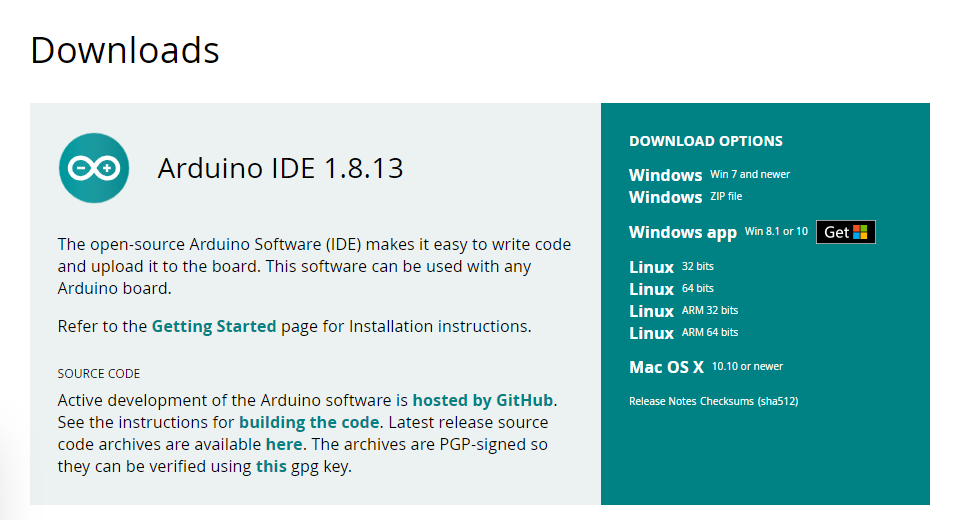
**Instructions for setting up Arduino Support for MATLAB:**

Prerequisites: Windows OS with MATLAB and Arduino\* installed

\* DO NOT install Arduino using Windows App Store, just download the regular Windows installer (or ZIP) and 

NOTE: This method does not allow you to directly modify the Arduino code, MATLAB handles that for you and you write code in MATLAB to control pins and interface with peripheral devices

1. Go to https://www.mathworks.com/hardware-support/arduino-matlab.html

2. Click 'Get support package' which will download the package\

3. Run the executable, which will open your Add-On Manager for MATLAB (it may prompt you to sign in)

4. Accept license agreement and click 'Next' to download and install the Arduino third-party package

5. Follow prompts for installer, which may require giving app permissions

6. When complete, click 'Setup Now' or type 'arduinosetup' in the command window

7. To setup the Arduino-MATLAB connection, you will have to select your board ('uno' for an Arduino Uno) and port (i.e. 'COM5')

The default communication libraries are I2C and SPI, and most sensors you work with will use one of those protocols

Click 'Program' to write the MATLAB Arduino Server to your board, which will allow you to use MATLAB to control

8. From the command window, create an instance of your board (i.e. myBoard = arduino('com5','uno'))

9. From the command window, TEST your connection by turning the built-in LED on and off (i.e. writeDigitalPin(myBoard,'D13',true))

Verify that the LED on the board turns on, then turn it back off (i.e. writeDigitalPin(myBoard,'D13',false))

10. Your good to go, check out the 'MATLAB Support Package for Arduino Hardware' help documentation for examples etc.

11. For reading an analog pin (such as with a temperature sensor), you can use readVoltage(myBoard,'A1') and then do the math on the MATLAB side