

FlexiForce Circuit Experiment SOP
By: Krish Gupta
25 November 2025 (Updated 26 November 2025)

Overview

This document will explain how to set up the physical flexiforce circuit used for experimentation as well as how to install necessary software and perform setup procedures to ensure smooth operation.

Materials

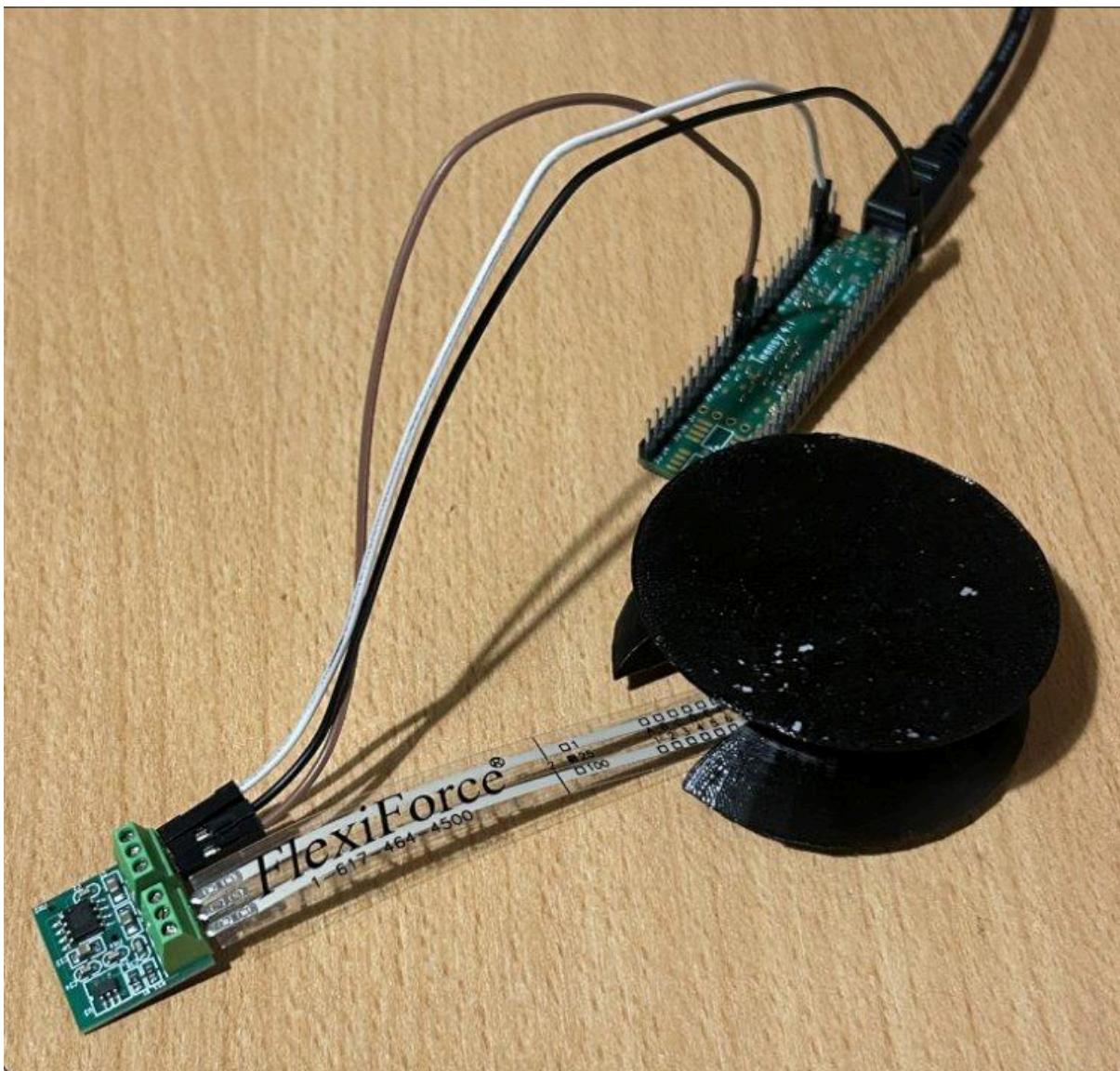
Physical:

1. [Teensy 4.1 board](#)
2. Amplifier module
3. [Flexiforce sensor](#)
4. 3 male-female jumper cables
5. Support fixture (from CAD files)
6. Scale
7. Screwdriver small enough to turn spring screws on amplifier
8. [USB 5-pin Micro-B cable](#)

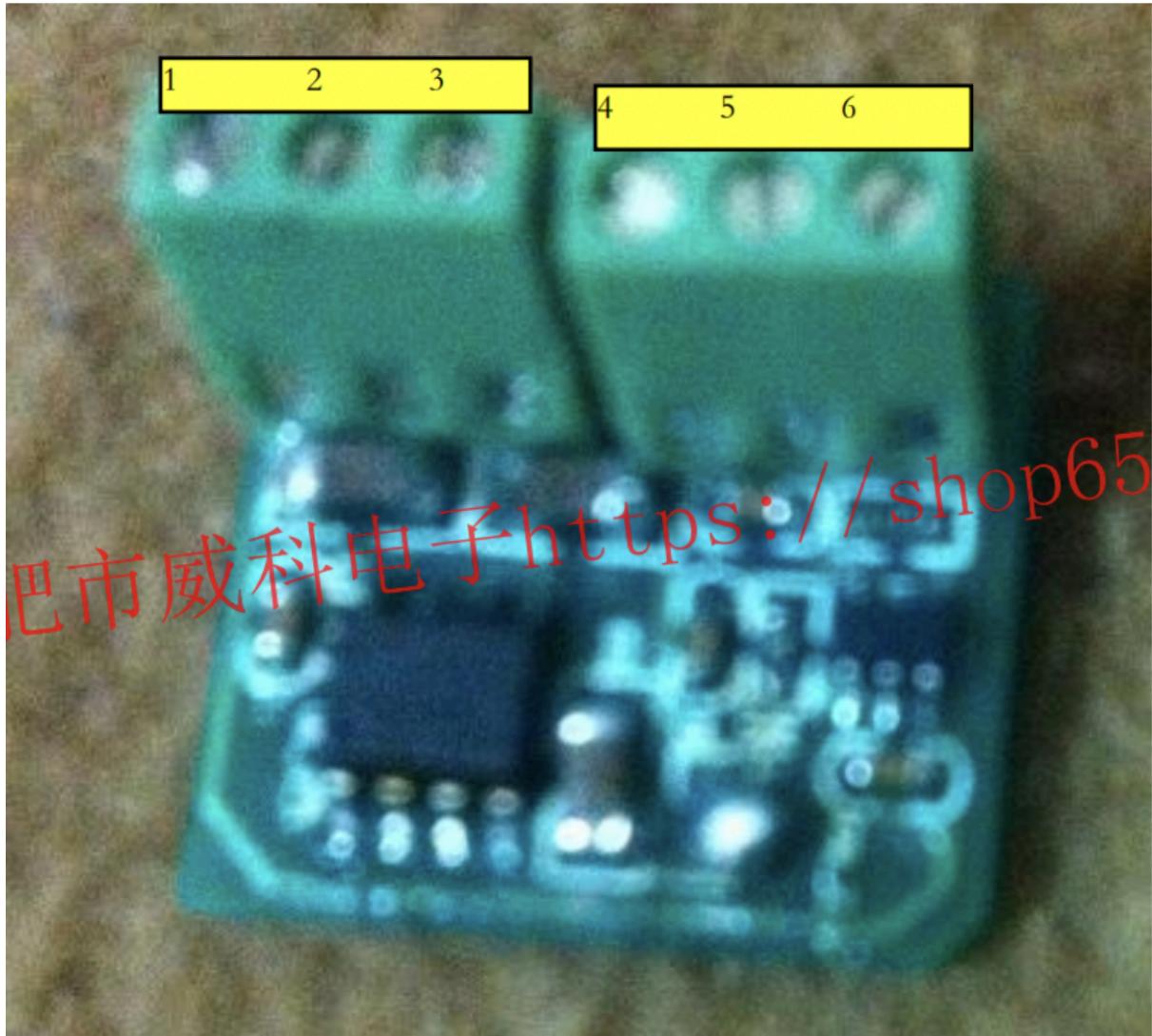
Software:

9. [MATLAB](#) and [Arduino IDE](#) software
 - a. Teensyduino add-on to copy and paste later →
https://www.pjrc.com/teensy/package_teensy_index.json
10. Code files (available on [GitHub](#)):
 - a. *TestCodeForceSensor.ino* (ArduinoIDE)
 - b. *TeensyCode.m* (MATLAB)

Circuit Setup



1. Loosen all 6 screws on the amplifier module enough to fit pins from jumper cables and flexiforce sensor
2. Orient the amplifier such that the terminal blocks are on the far side of the board from you. From left to right, the following connections should be made:
Pin 1: Power
Pin 2: Ground
Pin 3: Signal
Pins 4-6: Sensor pins
3. Once the pins are placed under the springs of the amplifier, tighten the screws to lock the pins in place.
4. The next step is to connect the female ends of the jumper cables to the teensy board. The connections should be as follows:
Power wire: 3.3V pin



Ground wire: ground pin

Signal wire: Analog pin (any of 14-23 or 34-41) (corresponds to A0-A9 and A10-A17)

5. Connect teensy board to computer using USB 5-pin Micro-B cable

Your circuit should now resemble the image shown at the beginning of the circuit setup protocol, minus the support fixture. For further information about the flexiforce sensor or teensy board, datasheets can be found hyperlinked in the materials section at the top.

Software Setup

Install MATLAB (no special packages or add-ons needed)

Installing Arduino IDE

Download [Arduino IDE](#) for your OS system. Link to the download page is hyperlinked here and in the materials section. The link can also be found on the Teensy board data sheet linked at the top. Follow steps for installation as prompted.

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger. For more details, check the [Arduino IDE 2.0 documentation](#).

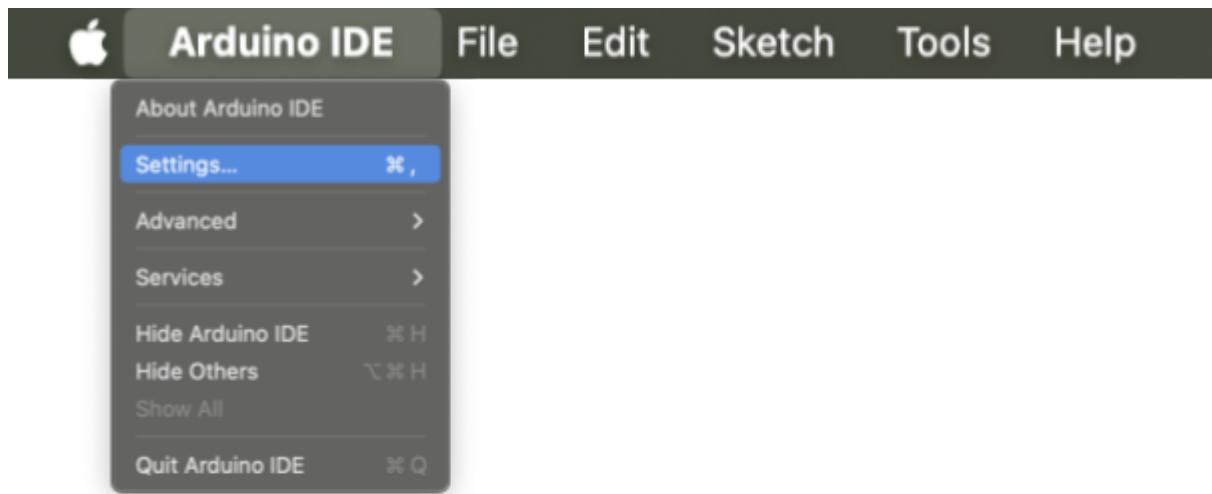
macOS Apple Silicon 11 Big Sur or newer (64-bit) [DOWNLOAD](#)

Nightly Builds

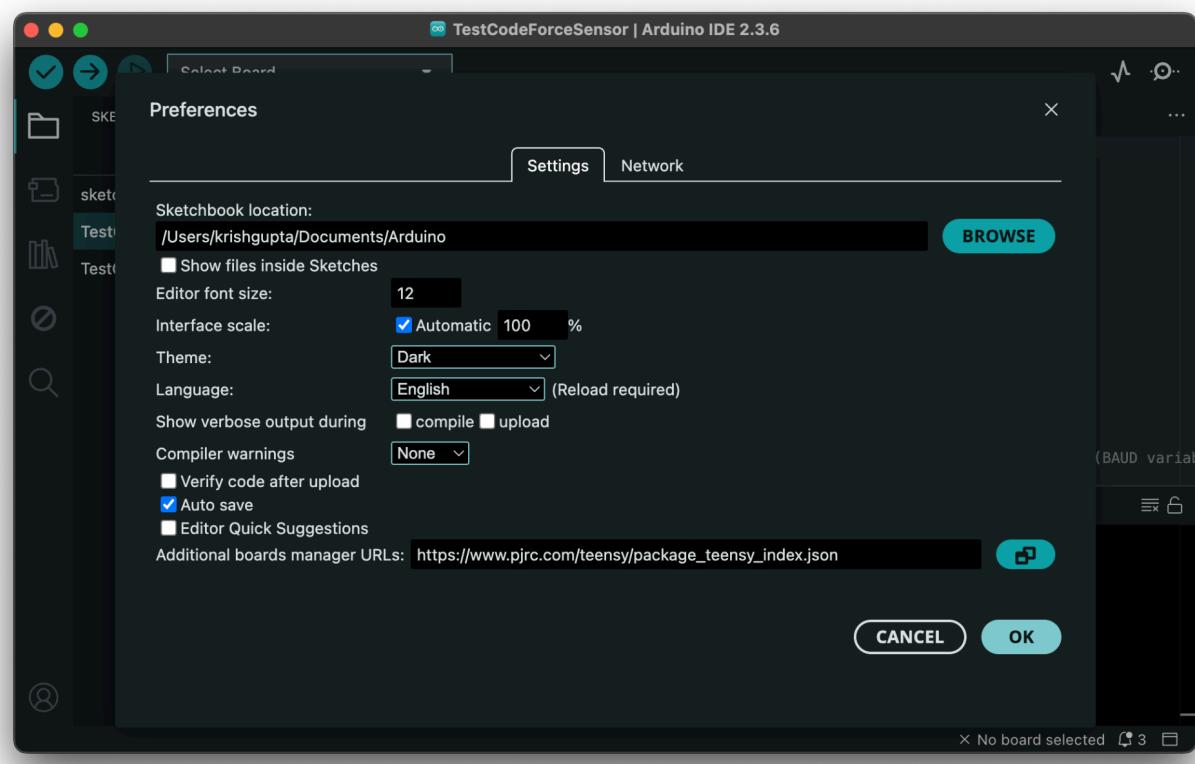
Download a preview of the incoming release with the most updated features and bugfixes.

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

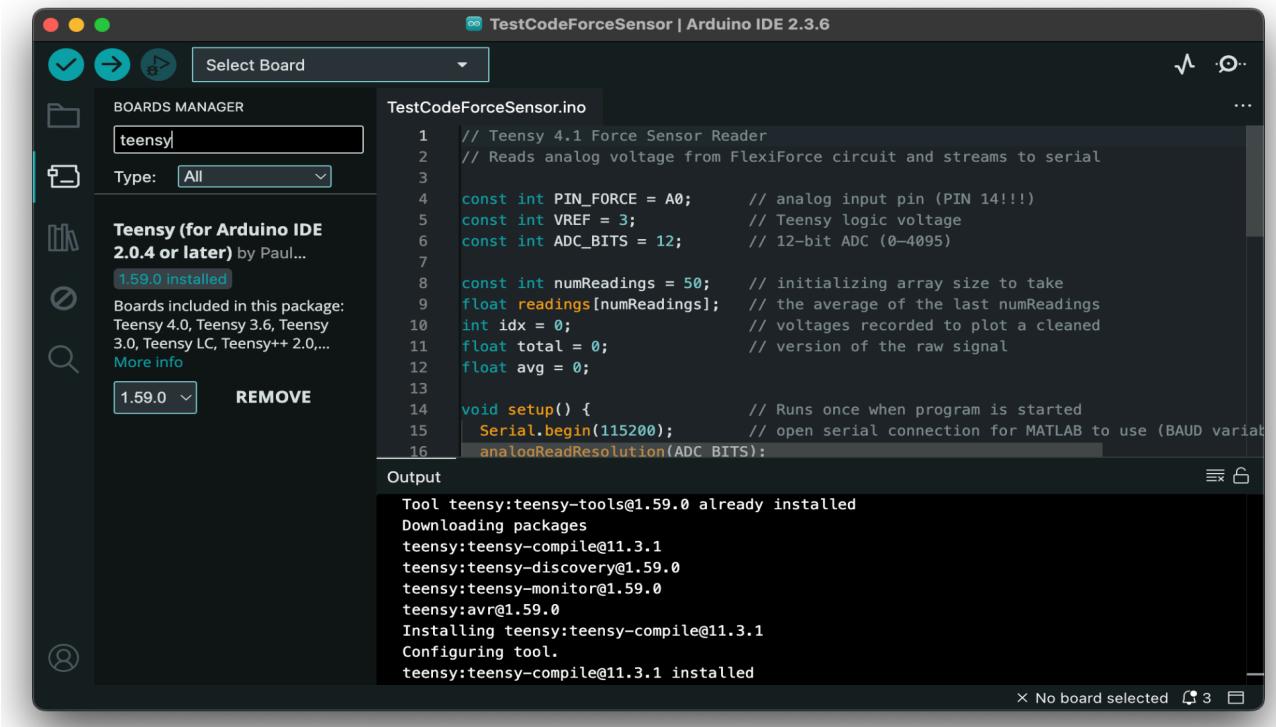
- Once installed, open the IDE and open the preferences page. For mac, go to the toolbar and select *Arduino IDE* → *Settings*. For windows, go to *File* → *Preferences*.



- A preference page should pop up. At the very bottom should be the option *Additional boards manager URLs*: with a text space next to it. Copy and paste the following link into the text box: https://www.pjrc.com/teensy/package_teensy_index.json. Press *OK*

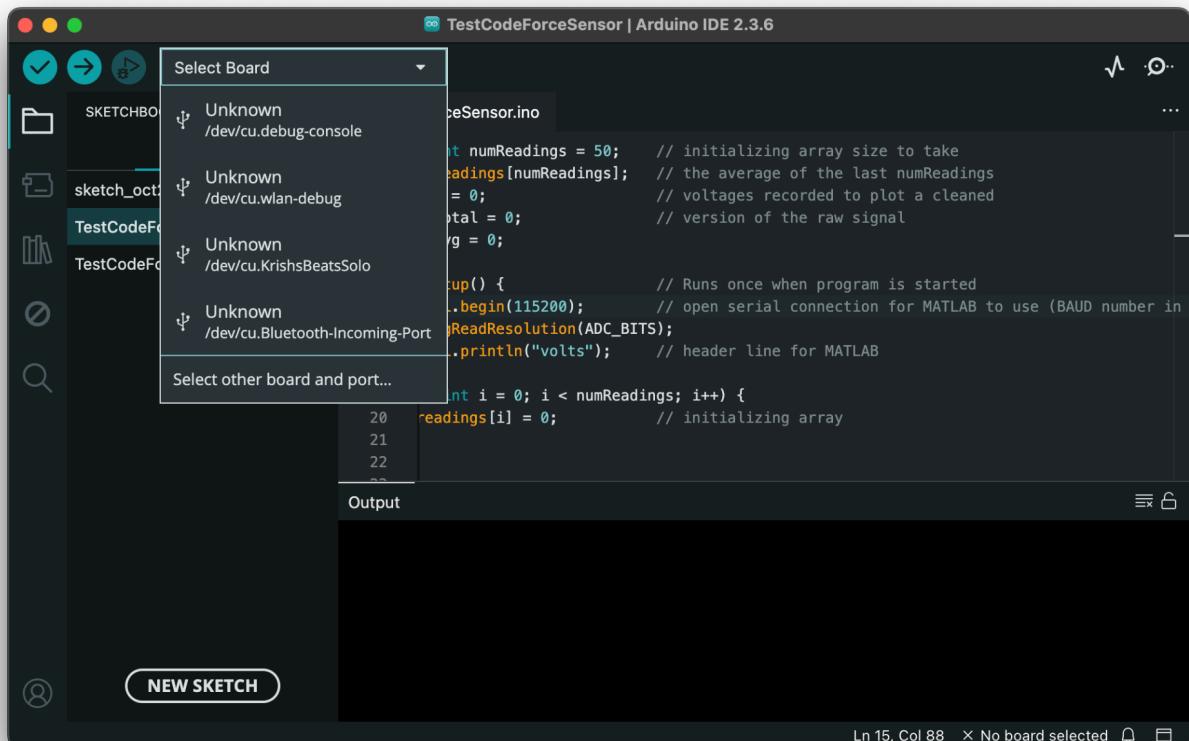


3. Close the preferences tab. Press on the *Boards Manager* option on the vertical tool bar on the left. In the search bar, type *teensy* and install the following IDE add-on.

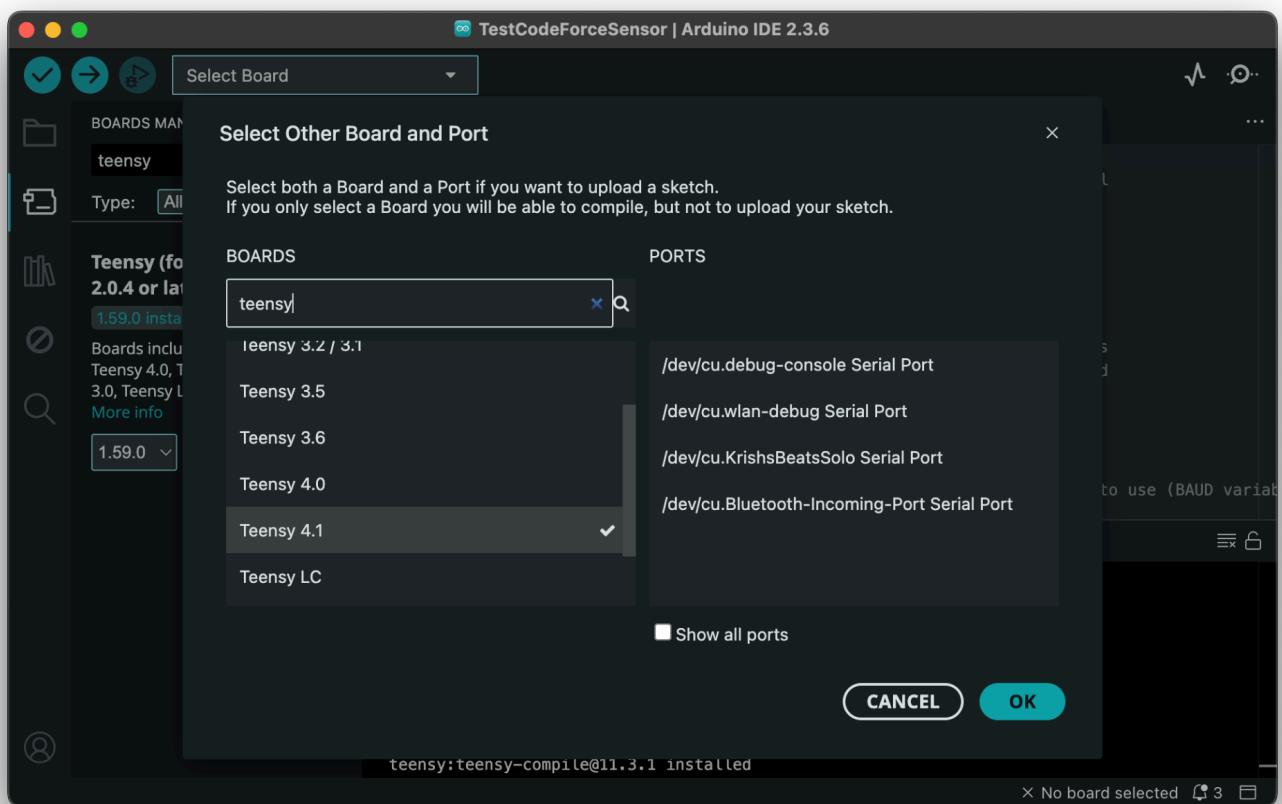


4. Once installed, press the physical button on the teensy board to activate it. Then on the IDE, click the *Select Board* dropdown menu and select your teensy board connection and port. For mac, use the /cu port connection instead of the /tty

connection. It should look like `/dev/cu.usbmodemXXXXXXXX` where the X represents a unique number for your port. For windows, it will appear as `COMX`.



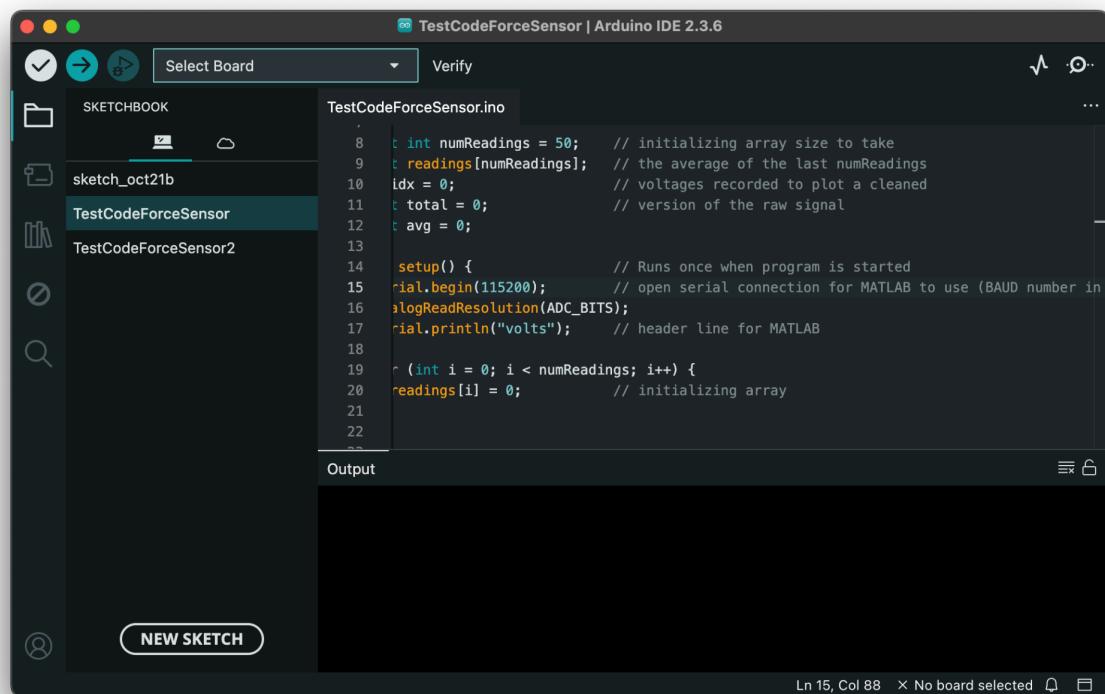
If it doesn't show up, press the *Select other board and port...* option and search for your board and connection.



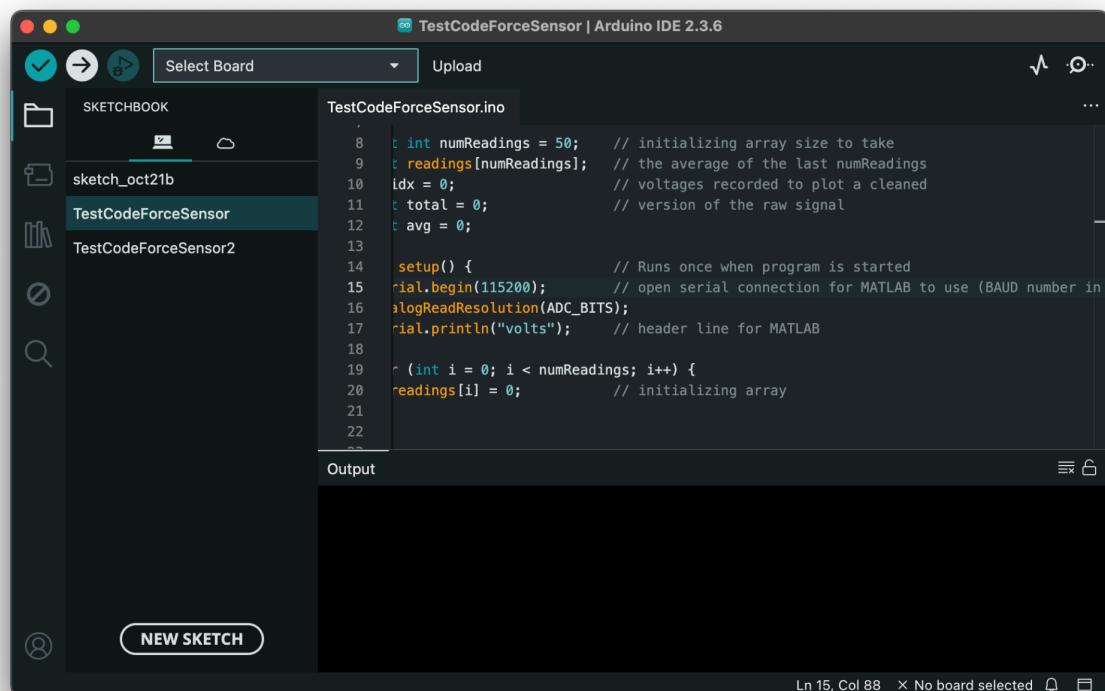
Running Experiment

1. TEENSYDUINO:

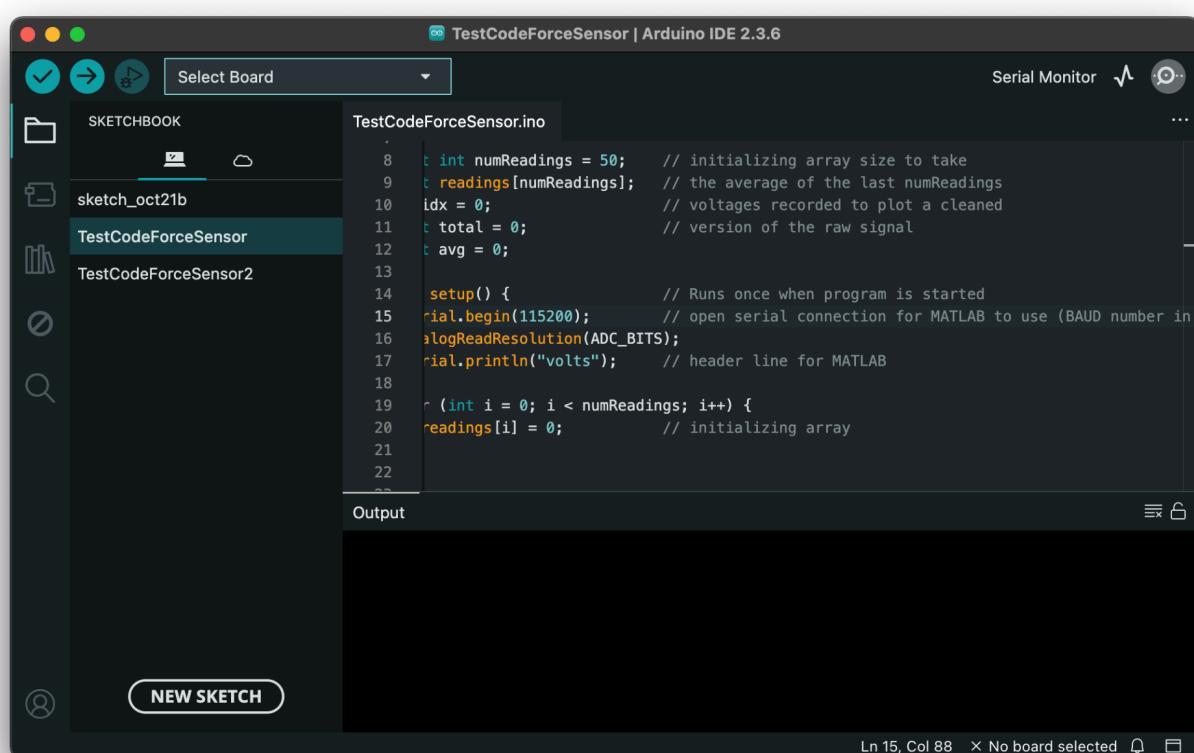
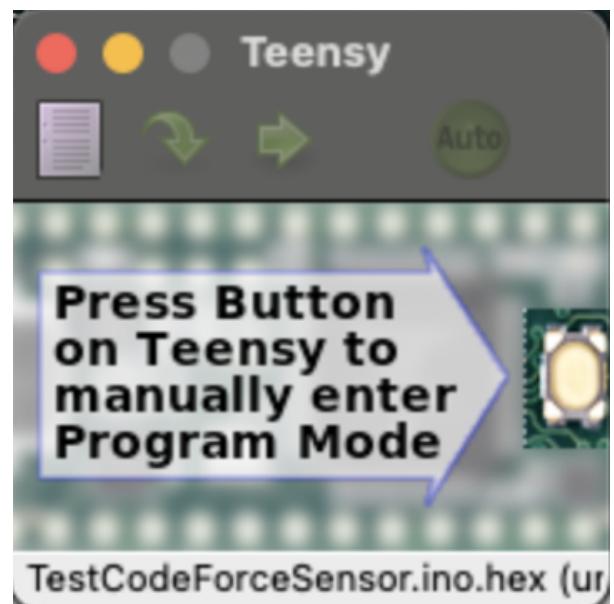
- Open *TestCodeForceSensor.ino* file in Arduino IDE
- Verify code before running by pressing check mark button in the top left corner



- When validated, press the button on teensy board to initialise the board setup. Red LED light should flash.
- Upload code to teensy board by pressing the arrow button next to the compiler. If no errors come up, code will be sent to teensy and automatically started

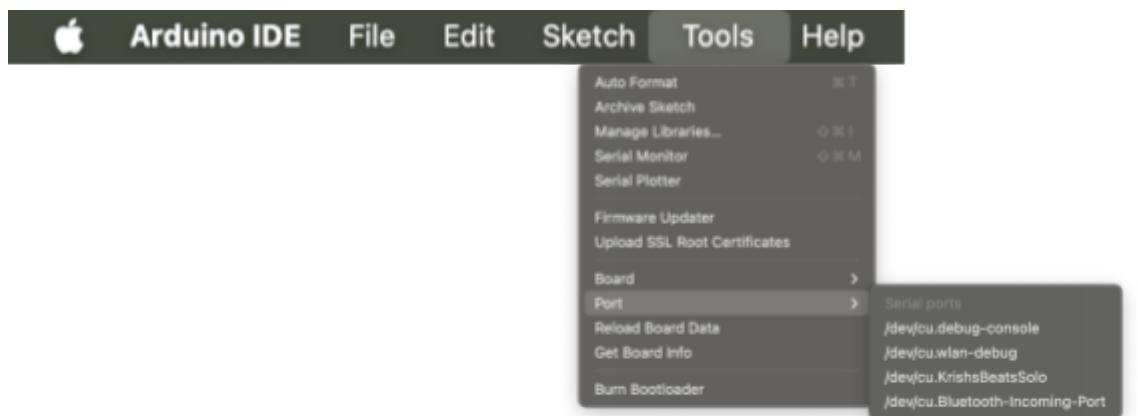


- e. Either this step or step 2 should have opened the teensy loader window that allows you to control when to reboot the teensy or reprogram it. If it is set to auto, then uploading new code from teensyduino will automatically pause and reboot the program with new code. If auto is deselected, then it is in manual mode and you will need to manually press the button on the teensy to put it into reprogram mode. Then uploading the new code from teensy should reboot the board with the new program. If it doesn't automatically do so, press the straight horizontal arrow (reboot button) to restart the program. If the physical button is pressed while in auto mode, the program will reboot and auto-restart code, so essentially nothing will happen since the same code will be restarted.
 - f. Once this is done, verify code and data recording in real time is working by pressing the serial monitor button on the right side of the window or by going to Teensyduino → Tools → Serial Monitor. This is how the voltage data is sent to MATLAB to read.



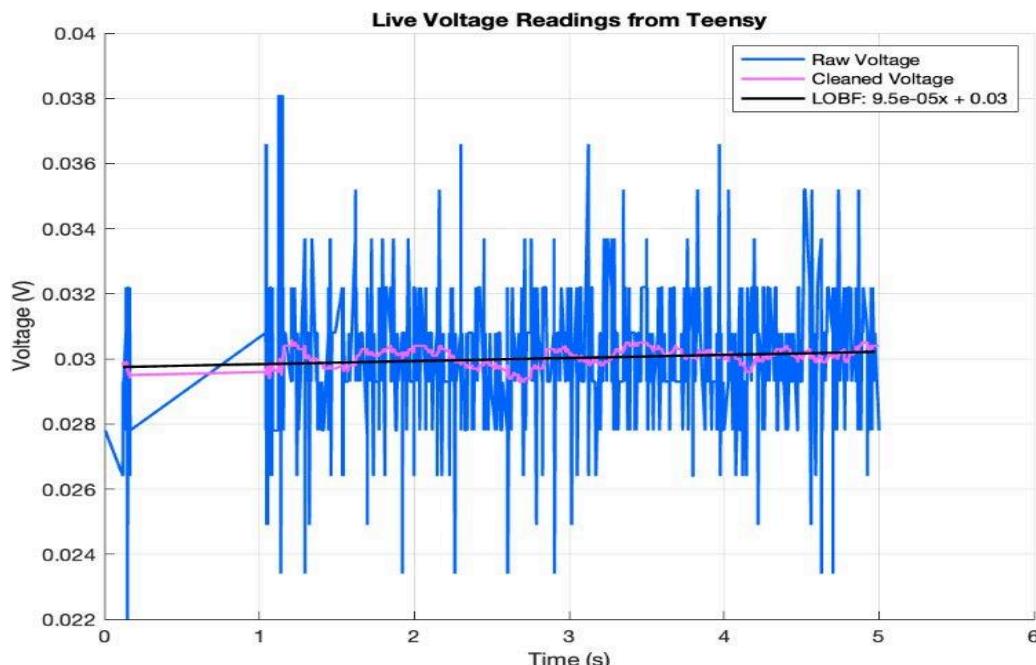
2. MATLAB:

- a. Open *TeensyCode.m* matlab file
- b. Ensure hard coded connections for serial port and baud number are correct.
Serial number is the last 9 digits of the port connection used to connect teensy board to the ArduinoIDE. You can also recheck by going clicking on the ArduinoIDE window and using the toolbar going to Tools → Port and seeing which port of yours is connected. Baud number is set in the *TestCodeForceSensor.ino* file in the *setup()* block using the function *Serial.begin()*.



```
void setup() {                                // Runs once when program is started
    Serial.begin(115200);                      // open serial connection for MATLAB to use (BAUD
                                                // number in MATLAB)
    analogReadResolution(ADC_BITS);
    Serial.println("volts");                  // header line for MATLAB
```

- c. After placing an object on the sensing pad, run the first section of code to read and plot the live voltage data sent by the force sensor through teensyduino. Cleaned data will be auto saved in a .csv file but be sure to rename the file every time or else previous data will be overwritten. There is no code to manually save graphs as png.



- d. Sections below are in reference to plotting Voltage-Mass and Force-Voltage data collected from initial circuit testing experiments and can be ignored or removed. However, if needed, there is code on how linear fit equations and R^2 values were calculated and plotted on scatter plots to be used in new collected data.

