Now that you’ve finished section 2 of the board and have finished soldering the LEDs, you’ll begin testing your LEDs.

**Testing**

Testing is always an important part of the electronics manufacturing process, as it helps to reduce the amount of error that can happen during any part of the manufacturing process. Within the scope of this project, testing can reduce the difficulty of the debugging for later steps. Inserting testing into any project can set a benchmark, when done correctly, that can show all of the tested components to be functioning, which can help to reduce the amount of troubleshooting that is needed when a problem does appear.

Testing the functionality of the LEDs is critical before finishing area 3 of your board because it can ensure the functionality of the LED array before moving on to another potentially buggy area, the audio response area.

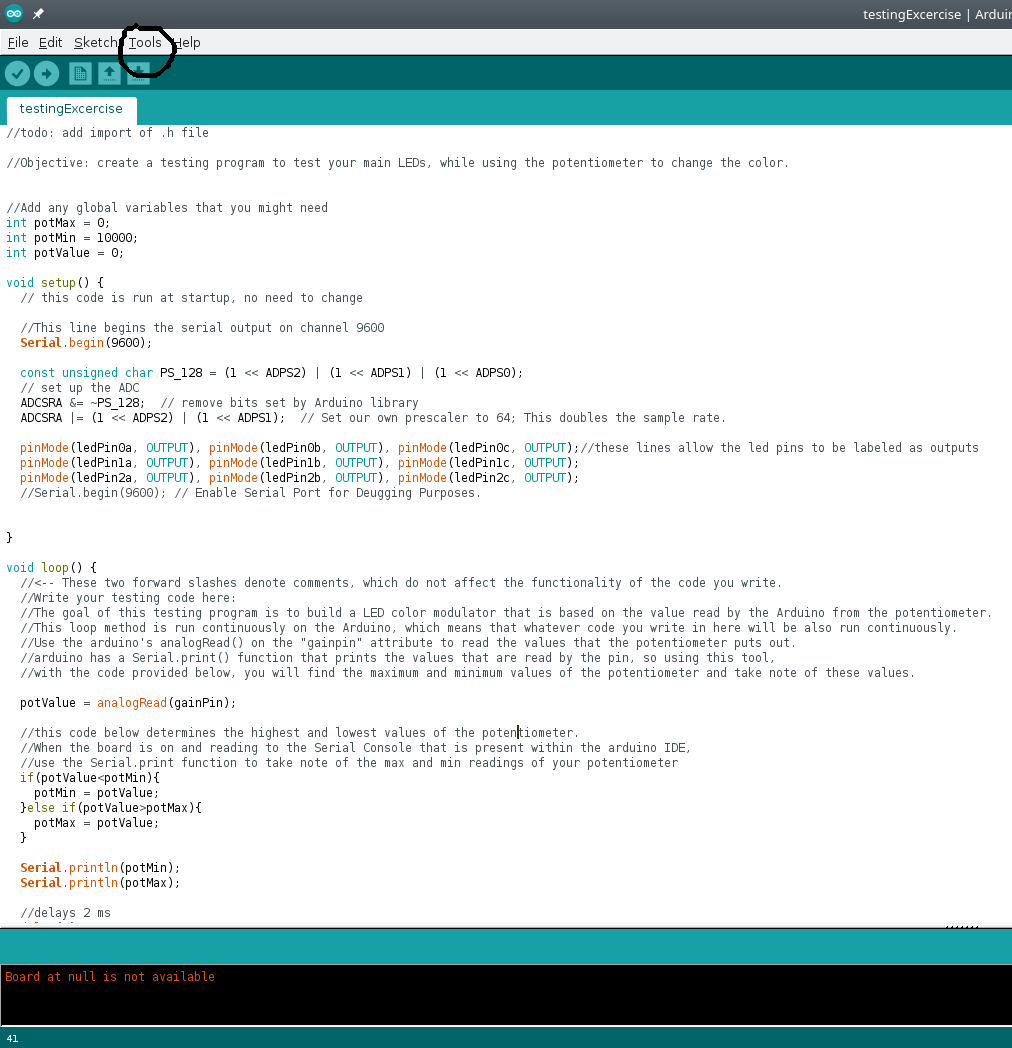
**How to Test**

Each LED that is soldered to wires and attached to the board connections has three built-in LEDs, a red, blue, and green LED. To ensure complete functionality, testing all three of these LEDs and all of the other components that you have installed to the board is necessary. For this assignment, write a simple Arduino program to use the input from the potentiometer you installed to control the colors of the LEDs in order to cycle through all of the LED colors. To accomplish this, you will need:

* PRISM kit assembled to part 2
* Computer with Arduino software
* Starter file on PolyLearn

To begin, attach your Arduino to your computer using the USB OTG cable provided in you kit. Then, open the Arduino software and open the starter file from PolyLearn. This file will have instructions inside of it to continue this testing phase.

To find the Serial Monitor, which is necessary to find the maximum and minimum values for the potentiometer reading, navigate to the Tools Menu > Serial Monitor in the Arduino Editor. Ensure that the channel setting is at 9600 Baud.



After you have finished programming the board, show your instructor the full functionality through the given Arduino file and, once given the go-ahead, continue with part 3.

// to be added to:

//create write-up to show understanding of the code (highlight the variables that can be changed)

After part 3:

**Hardware and Software**

The connection between hardware and software can sometimes be confusing and convoluted, but through this tutorial, you’ll learn how the software programmed on the Arduino connects to the hardware that you just finished soldering.

**Arduino**

**after finishing the project,**

**can play with some things in the code?**

**More colors? (smaller ranges?)**

**customization of the project (make them show what they have done on the project)**

Ideas for what we want to do:

Arduino:

polarity of LEDS

brightness dependent on voltage

difference between digital and analog output

explain how analog output works on the arduino: is discrete steps (analog looking signal with small steps in the signal.)

pulse width modulation – related to the DAC

test only 13-10 pins

use graphs over time to display how the input/output is changing between digital and analog

talk through the code – make a video that leads the students to find the code – discourages copy paste

PLACEMENT

placed before the code is uploaded

“Before uploading the final code, we will test a few of the input and output pins, so that we can ensure that your Arduino board is functional.

VIDEO:

talk about all of the syntax that is used in the code – curly braces, void keyword, what setup() and loop() mean. Parentheses are originally used to pass through variables into the **function** describe what each non-word means – details are important

Separate videos from the digital output and the analog input/output

digital:

show changes of the pins after all of the code is written

analog:

explain why potvalue is instansiated outside of the main loop

Serial monitor – explain why the begin needs to be written before any data can be sent to the computer (on what baud/ what is baud?)

explain the importance of the variable potvalue