Circuit Design Assignment 3:

Embedded Programming

Assignment due Sunday at 11:59pm EST

# **1 Objectives**

Programming a hardware system is essential to any embedded designer and electrical engineer. Arduino is one of the most popular and accessible methods of programming a circuit, utilizing the microcontroller to generate signals and command hardware. In this assignment you’ll be using an Arduino to program a light, based on how you (the user) interface with various components like potentiometers and buttons. These skills will be useful as you jump into your final project next week.

# **2 Parts**

* Arduino Microcontroller
* Resistors
* Capacitors
* Potentiometer
* Push Button
* Buzzer
* LED

# **3 Lab**

For this series of exercises, we’ll use the code examples provided with the Arduino IDE under File>Examples. Each example provides a link to the official documentation and explanation of the example code.

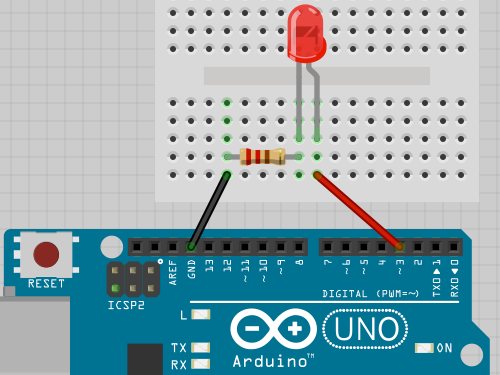
# 3.1: Blink

[Blink code documentation](http://arduino.cc/en/Tutorial/Blink)

1. Open File>Examples>01 Basics>Blink
2. Note the FOUR parts of the code
   1. Comment Header
   2. Global variable definition – define a var for the LED pin number
   3. setup() function – set the pin to be an OUT pin (can be IN or OUT)
   4. loop() function – blink the LED
3. Compile, Upload, and Test to make sure everything is working – the little LED on the board should blink
4. Make changes to the code – select all the code, copy, and create a new sketch with this code – then:
   1. Make an error in the code: In the setup() function, change pinMode to pinModer and verify/compile the code. **What happens?**

**States that there was a compilation error “ ‘pinModer’ “**

* 1. Hook up your own LED to pin 3 and change the “led” variable to reflect this. Compile, Upload, and Test to verify
  2. Vary the argument for the delay() function to change the feel of the blinking
  3. Add additional sets of digitalWrite()/delay() code to make more complex patterns of blinking



**Include a picture or video of your LED turning on and a snippet of your code.**

# 

# 

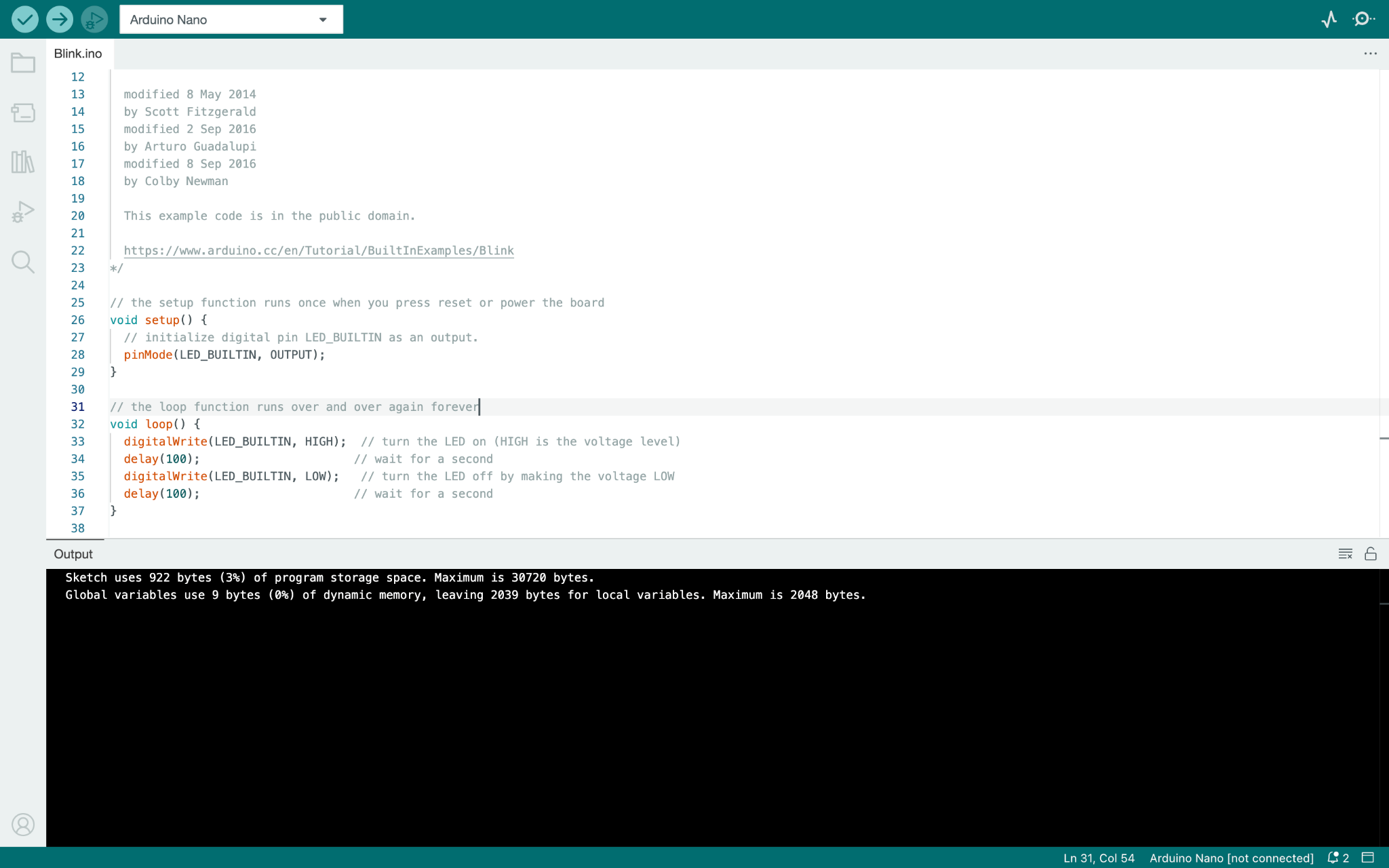
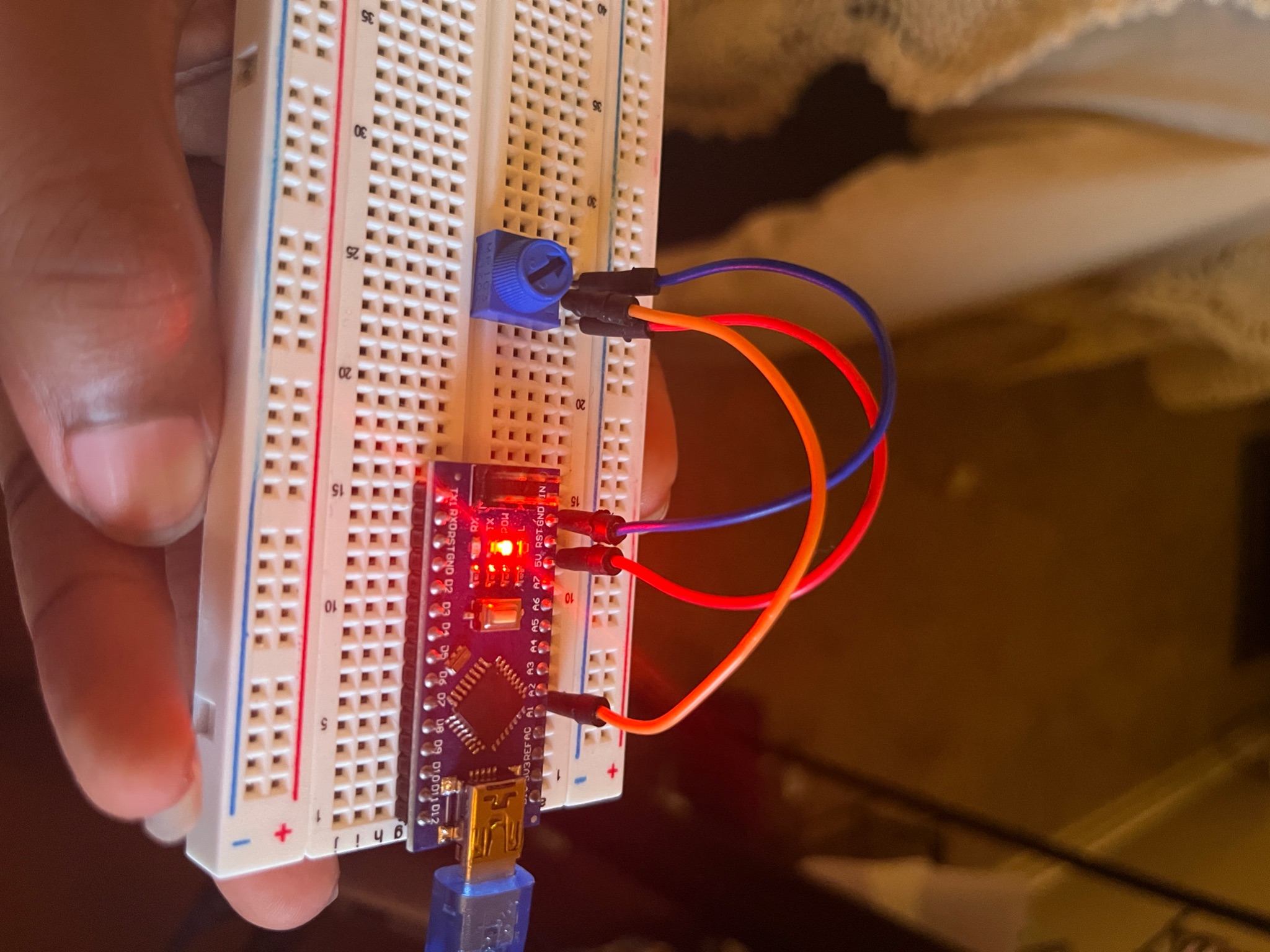
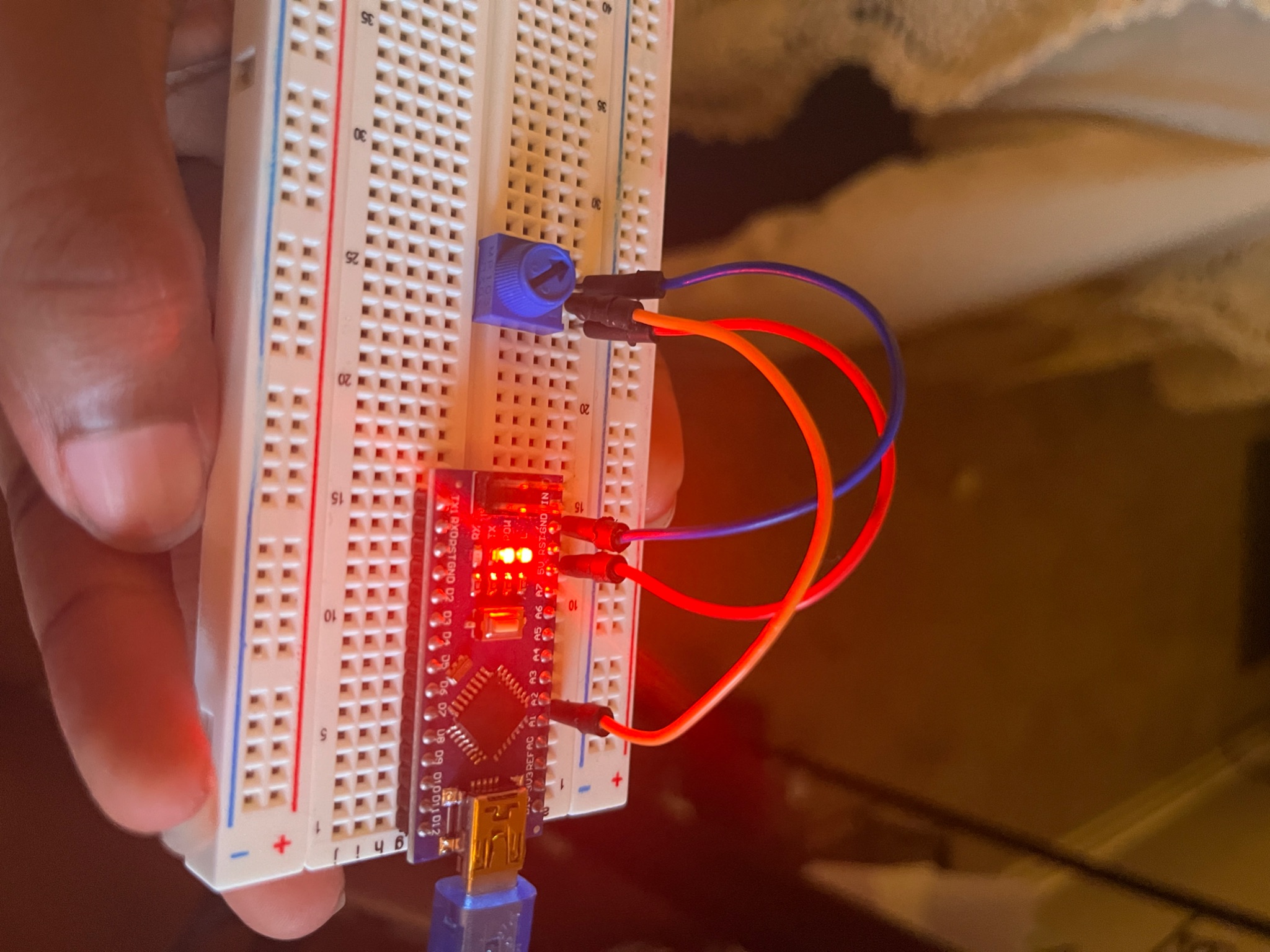
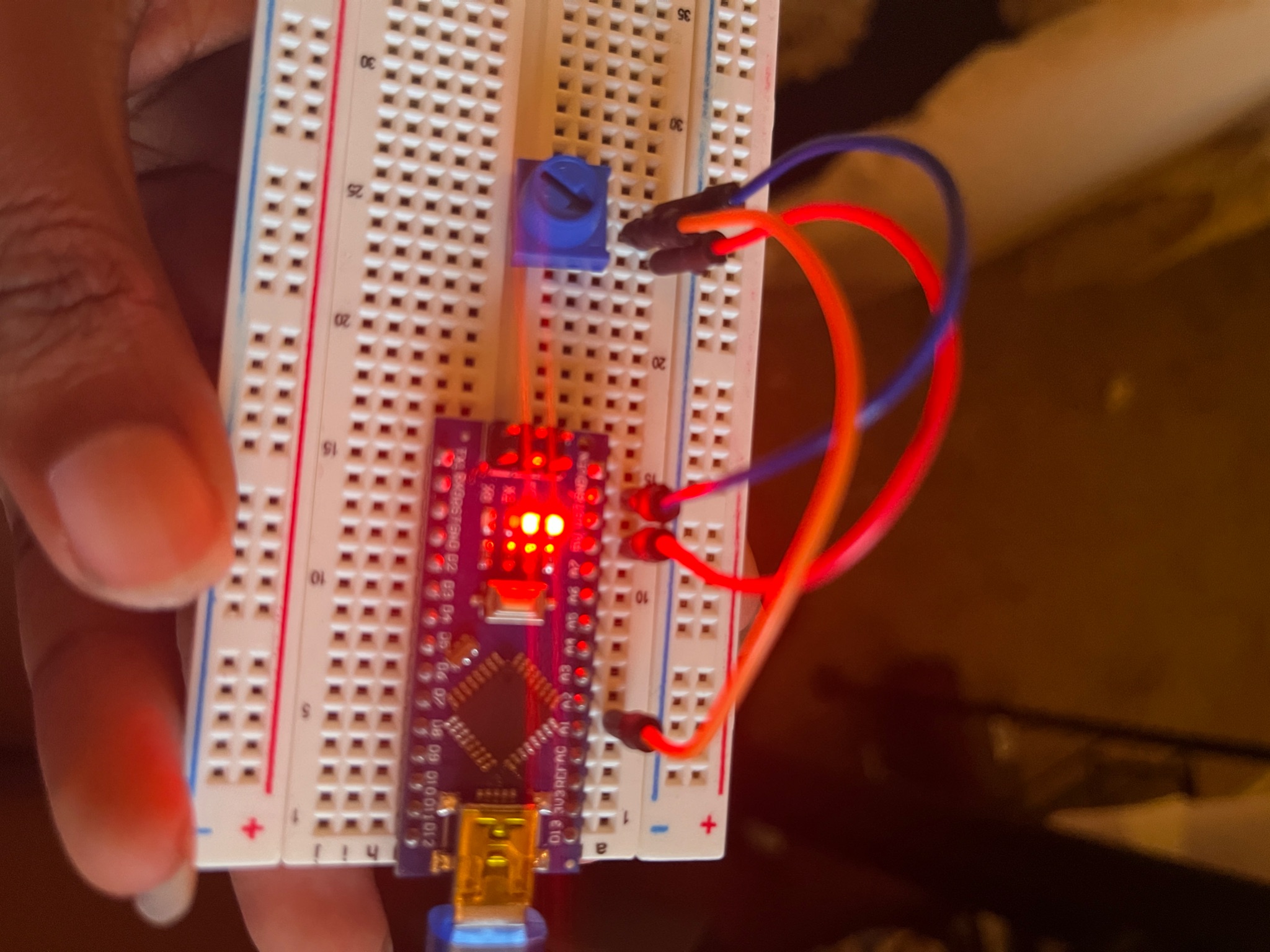
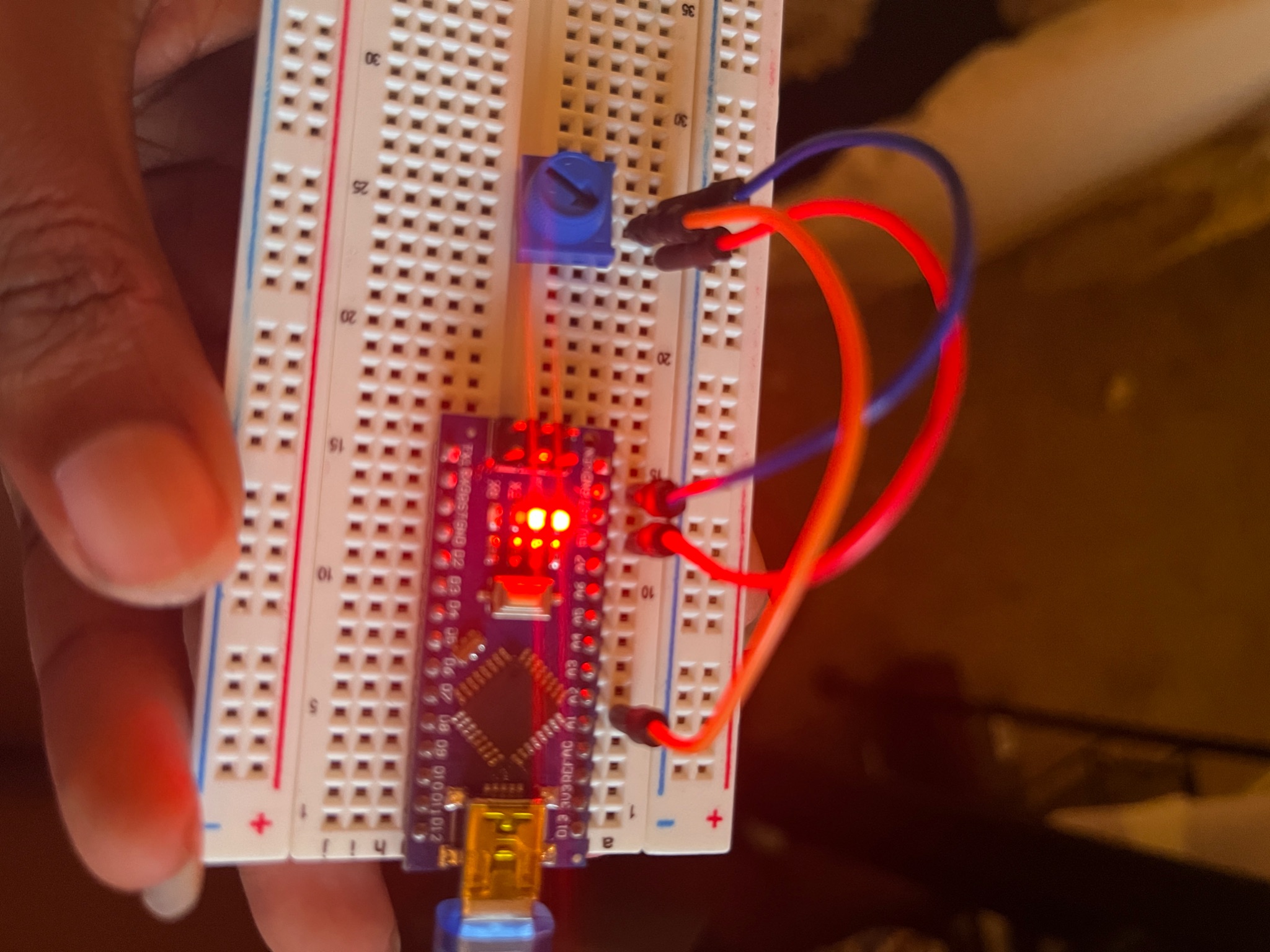
# 3.2: AnalogInOutSerial – Read an analog input and set the rate of an LED blink

[AnalogInOutSerial code documentation](https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogInOutSerial)

1. Open File>Examples>Analog>AnalogInOutSerial
2. Connect a knob to analog pin 0
3. Compile, Upload, and Test to make sure everything is working – the little LED on the board should blink at different speeds depending on the position of the knob
4. See how changing the knob varies the blink rate
5. Look at the following parts of code:
   1. outputValue = [map](https://www.arduino.cc/reference/en/language/functions/math/map/) (sensorValue, 0, 1023, 0, 255);
   2. [analogWrite](http://arduino.cc/en/Reference/AnalogWrite) (analogOutPin, outputValue); // analogWrite() sets the PWM brightness of the output LED
   3. [Serial.print](http://arduino.cc/en/Serial/Print) (“sensor = ” ); // prints out the values onto the user interface
   4. [Serial.println](http://arduino.cc/en/Serial/Println) (outputValue); // prints with a carriage return
   5. Make a new sketch (clone the function given) and change the settings in the map() function to see how it works

**Include a picture or video of your LED blinking based off of moving the potentiometer and a snippet of your code.**



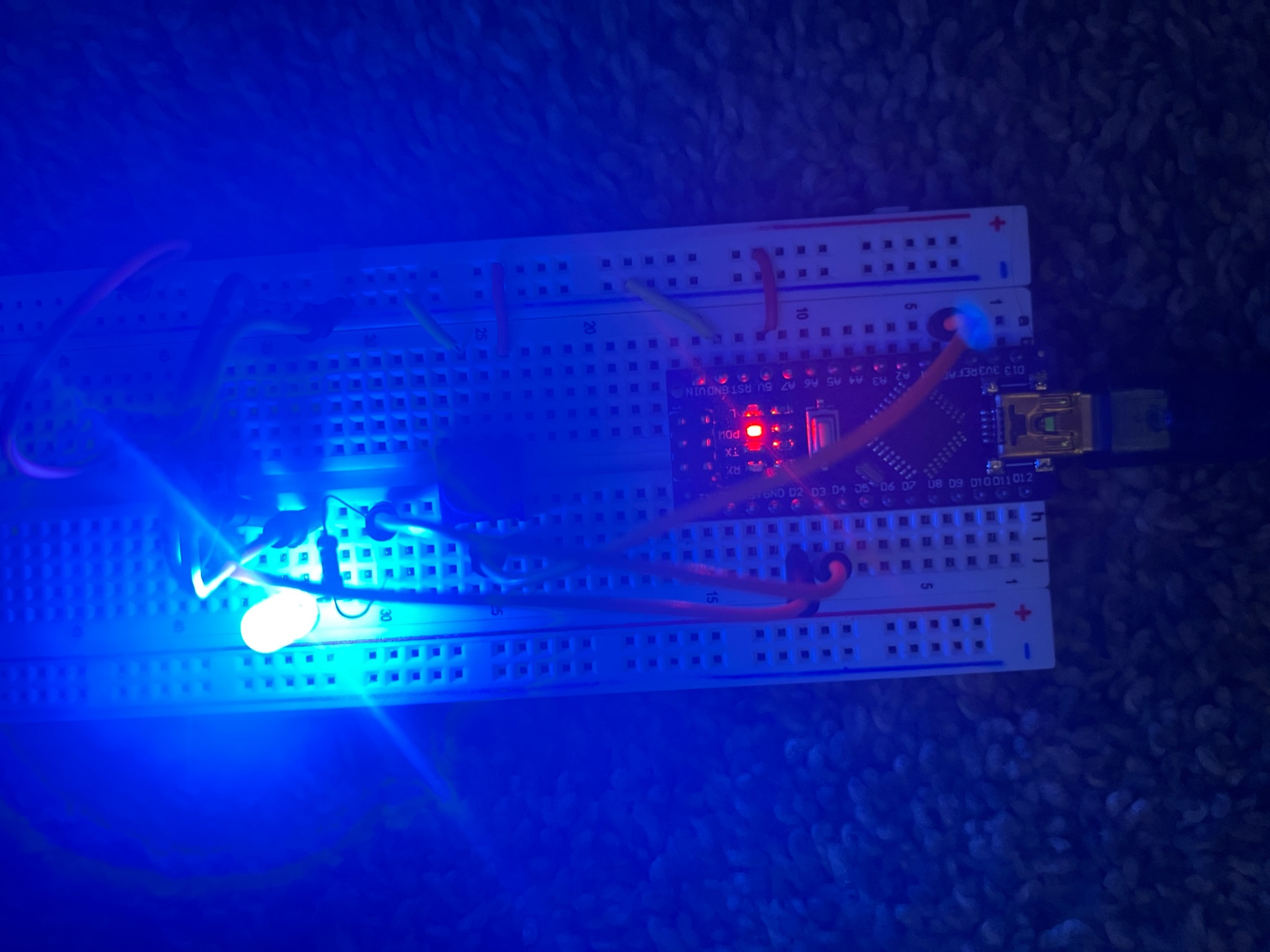
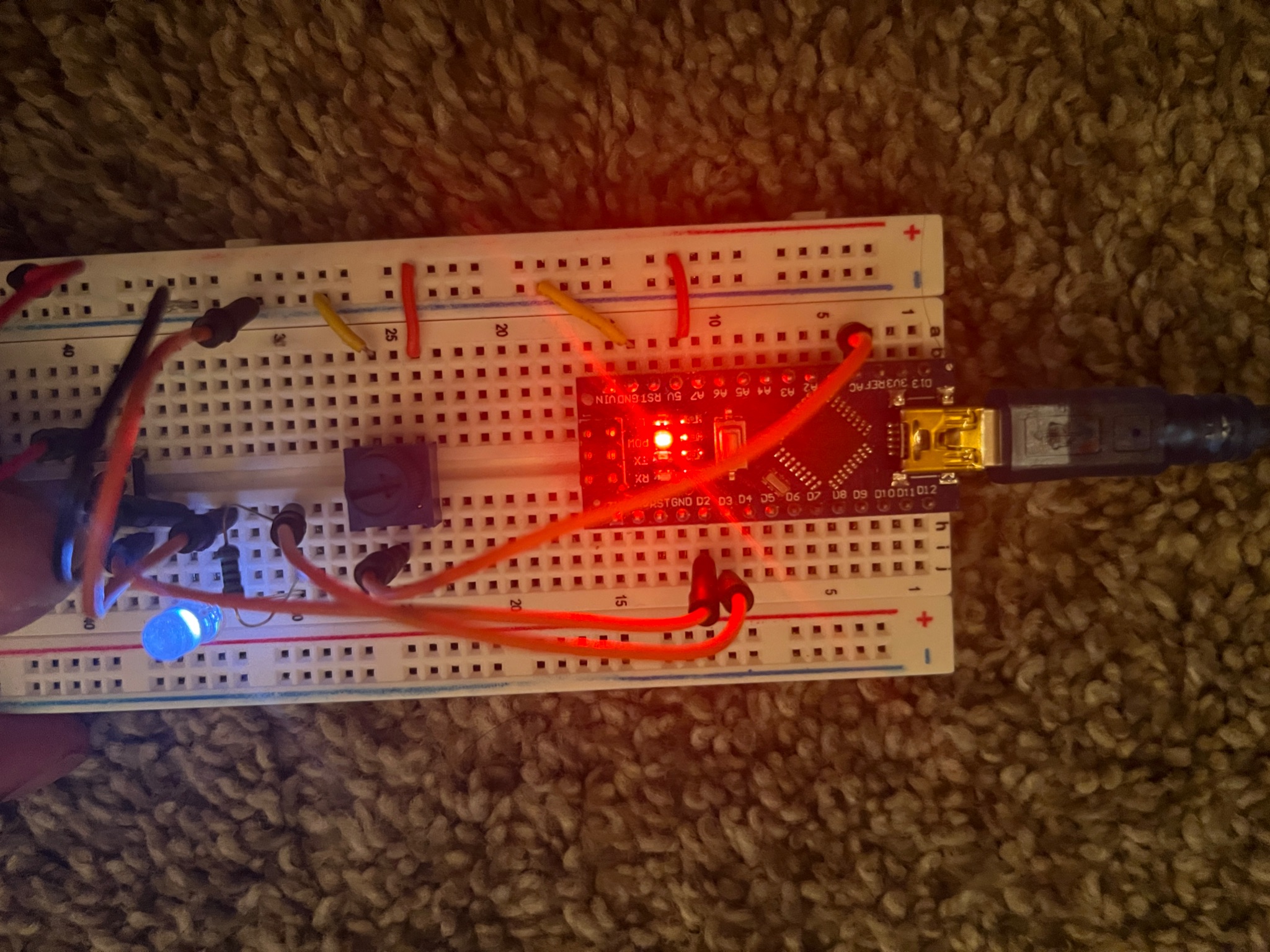
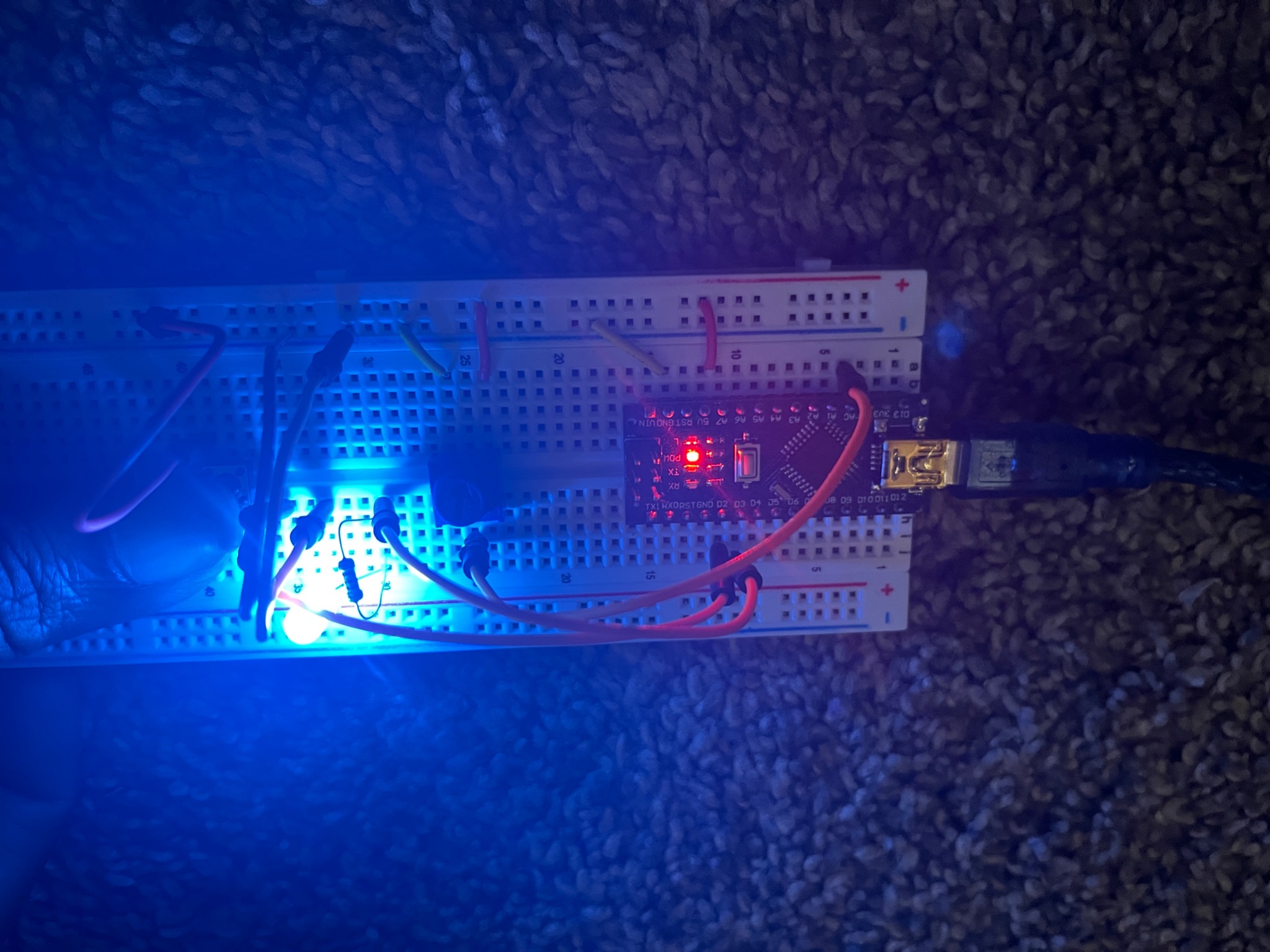
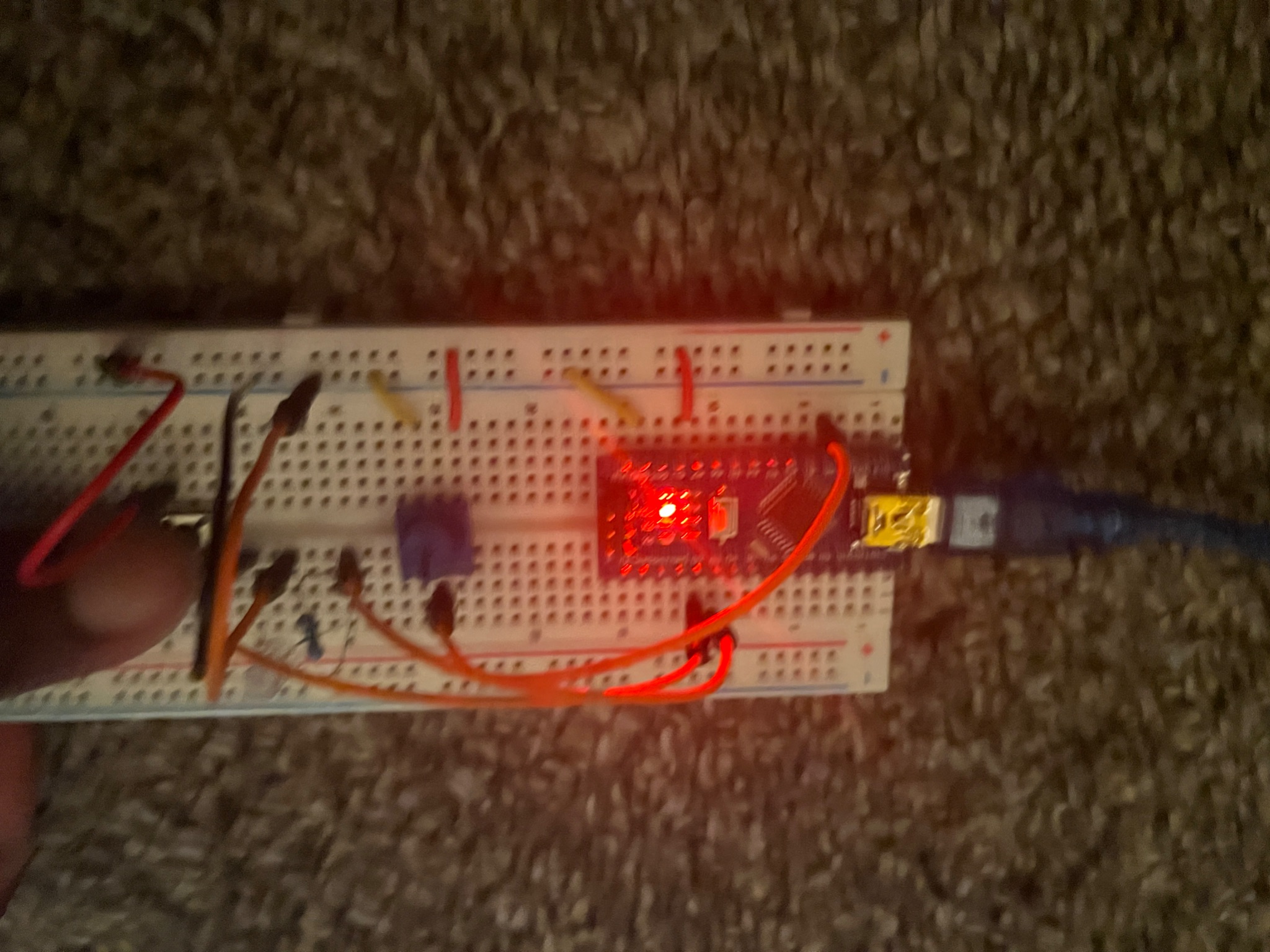
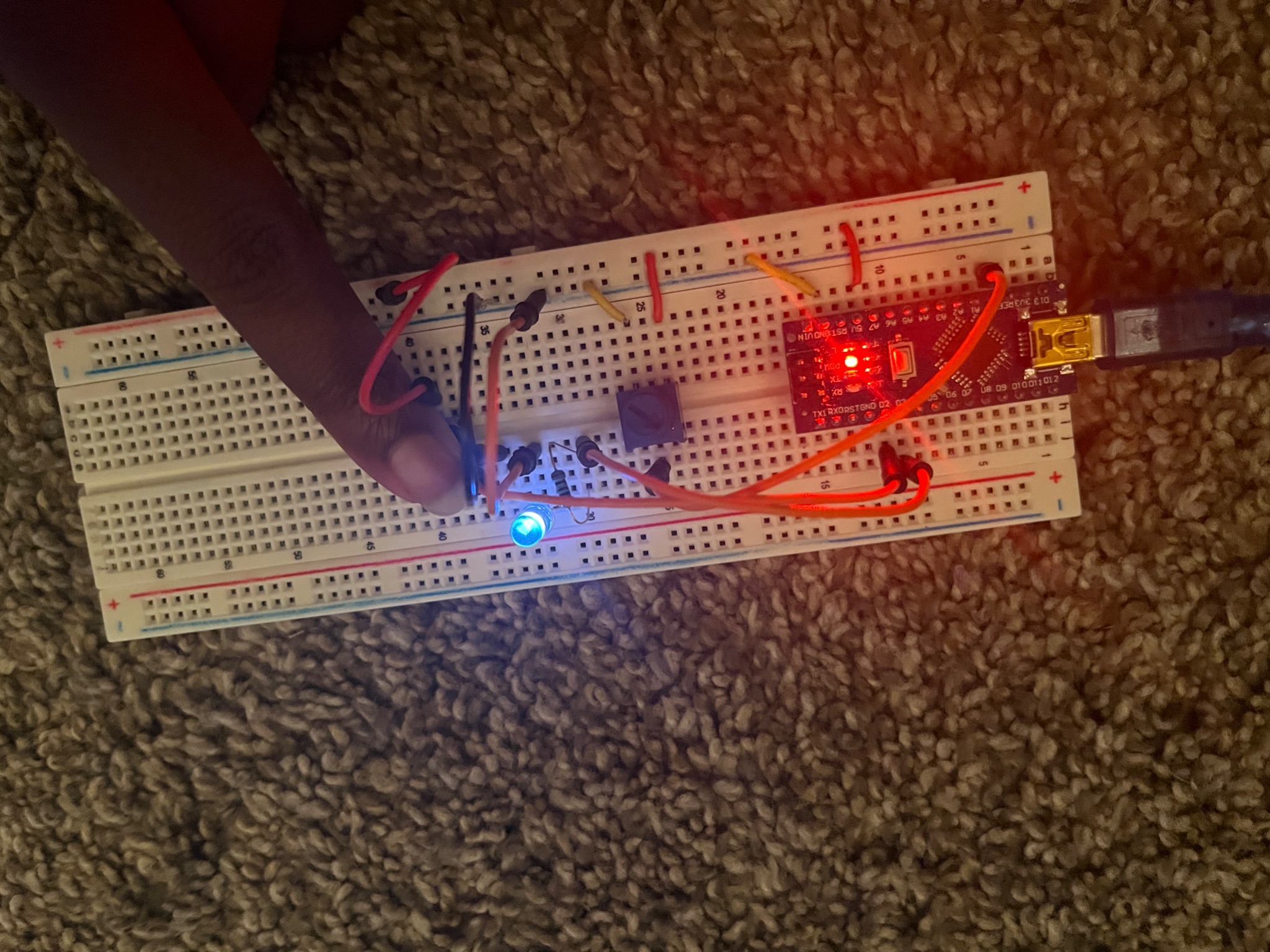
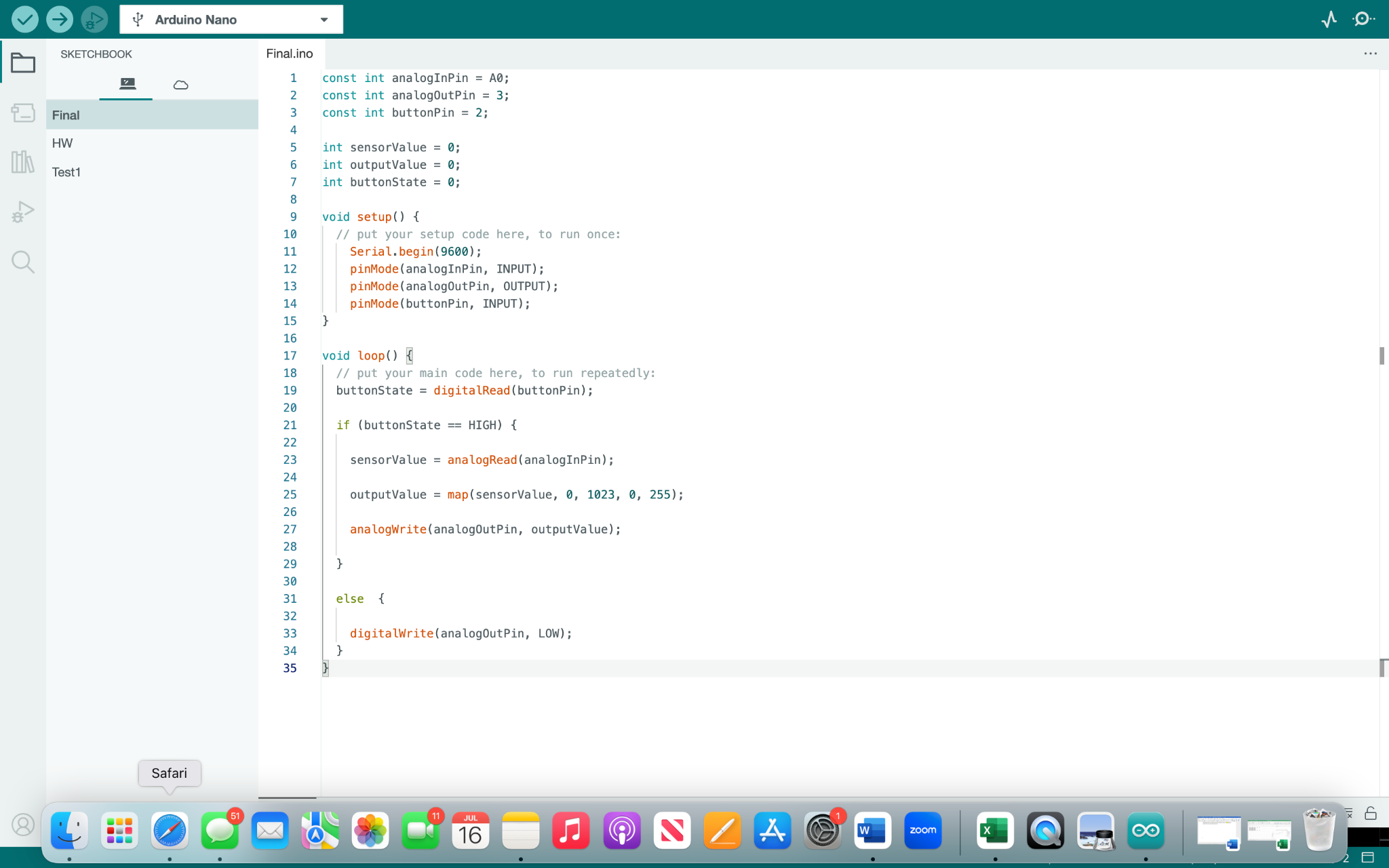
****

### 3.3: LightDimmerSwitch – Use a switch sensor to turn a dimmable LED on or off

Create a new sketch that works like this:

* A switch sensor connected to digital pin 2
* An LED connected to Digital Pin 3
* A knob connected to Analog Pin 1
* If the switch is on (pressed down), the LED should be on
* If the switch is off (not pressed), the LED should be off
* When on, the brightness of the LED should be controlled by the knob

**Please include a picture of your code, and a video or series of photos showing the results expected above**

****

Suggested steps:

1. Use AnalogInOutSerial example as your starting point. Make a new sketch and manually type in the AnalogInOutSerial code, with modifications for the different pins used for the knob and LED (you can skip all the comments).
2. Once that is running okay, review this [Button (Links to an external site.)](http://arduino.cc/en/Tutorial/Button) sketch on the use of [digitalRead (Links to an external site.)](http://arduino.cc/en/Reference/DigitalRead) () – File>Examples>Digital>Button
3. Now add code to your sketch that checks the HIGH or LOW value of the switch using the digitalRead() function
4. If the switch is on, run your code the checks the analogRead() and sets the brightness with that mapped value using analogWrite()
5. If the switch is off, simply set the LED to be off using analogWrite()

### 3.4: Optional Extra Credit (+10 points)

This is an optional extension project for those who want an opportunity for extra credit and further build their Arduino skills. There’s no need to submit anything for this section if you do not attempt it. If you’re interested in this part please choose an item from the following list:

1. Create a lie detector using your Arduino Nano. Follow this example to create a lie detector <https://projecthub.arduino.cc/BuildItDR/arduino-lie-detector-41f703> . Please note rather than soldering wires directly to your Arduino, please use your breadboards and just hold the ends of your wires (strip your own for more contact coverage!). A rough approximation is more than adequate to get full points on this part of the assignment, no need to make a formal enclosure or have any permanent connections.
2. Make an arduino project using the Wokwi Online Simulator software (<https://wokwi.com/> → Start from Scratch ). As a bare minimum add an LED screen that says your name when the user interacts with a sensor of your choosing. Feel free to expand upon this and turn it into a game with multiple sensor or user interactions options if you have time! For more context into what makes Wokwi so interesting feel free to check out the following software summary: <https://maker.pro/arduino/tutorial/free-arduino-simulator>
3. Come up with your own project! Use the arduino nano plus a combination of at least one new arduino library (<https://github.com/Lembed/Awesome-arduino>) and at least one other electronic component supplied in your kit to achieve an objective of your choosing (that is not identical to anything in either this homework assignment or lecture). Feel free to get creative!

# **4 Reflection**

Individually, answer the questions below. Two to four sentences for each question is sufficient. Answers that demonstrate little thought or effort will not receive credit!

# 4.1: What was the most valuable thing you learned, and why?

The most valuable lesson I learned was how to apply code to circuit boards, because often my written code just has an output on a screen, not an actual board.

4.2: What skills or concepts are you still struggling with? What will you do to learn or practice these concepts?

One concept I struggled with was the actual wiring of my board just because it is a new skill that I am learning. In order to improve on my skills I continue to watch Youtube videos for help.