Research:

My research was done before the beginning the project as well as during in order to fix bugs in code and change the hardware design. I had originally wanted to make an automated feeder for animals or waterer for crops. However, due to the extra components that were not included in the starter kit I did not pursue those ideas. I then began thinking about what was included in the starter kit and what I could do. From working with the laser tag components and doing the blink and Morse code projects earlier in the semester, I had a pretty good sense of how to use LEDs. I then had the idea that I could make an automatic light with the components in the starter kit.

I began researching different aspects that I was unsure about. Those included the following:

Is it possible to use LEDs in an array with a photoresistor?

<http://forum.arduino.cc/index.php?topic=286509.0>

How should I set up the hardware for the photoresistor and LEDs?

<https://www.instructables.com/id/Arduino-Mega-2560-based-LDR-Light-Intensity-Contro/>

<https://www.allaboutcircuits.com/projects/an-arduino-controlled-light-sensor/>

<https://www.instructables.com/id/Arduino-LDR-With-LED/>

How to set up a 3-state button with the photoresistor?

<http://forum.arduino.cc/index.php?topic=270723.0>

<https://forum.arduino.cc/index.php?topic=16303.0>

How to set up a potentiometer?

<https://www.arduino.cc/en/tutorial/potentiometer>

I was stumped on the potentiometer and the button but, I asked professor Sabal to help and he helped get the code to be where it is now. The potentiometer and the button work with the photoresistors and the LEDs.

Algorithms:

As far as algorithms are concerned I believe that the three-state button is the main algorithm that I need to mention. Code Shown Below:

/\*Button has three states,(all off, Photoresistor & potentiometer controlled, and all on). When button is pushed there is a delay and goes to the next state.\*/

1. int buttonPush = digitalRead(buttonPin);
2. if (buttonPush == HIGH){

/\*Mod for 3 States\*/

3. state = ((state+1)%3);

/\*Delay for button in milliseconds\*/

4. delay(2000);

The button needs three states. One for all the way off, one for all the way on, and one for the LEDs to be controlled by the photoresistor read and is adjusted by the potentiometers. Line 1 translates to “the button being pushed is equal to the designated pin for the button to be read. Line 2 translates to when the button is moved to high (being pushed) then there are conditions and things it needs to do. Line 3 is giving a value/function for the state integer. The state is equal to the current state plus 1 and the %3 means that is has three states, so 0,1, and 2. Line 4 is a delay to make sure the button does not go to fast or is held longer (does not skip states).

Later in the code there are lines that determine what the LEDs do at each state. See below highlighted if statements. They translate to if the photoresistor is less than or equal to 500 and the state of the button is equal to 1 OR 2, then the LEDs are turned on. If this is not the case (else) then the LEDs are turned off.

/\*Reading Photoresistor for Green and White Combo. Photoresistors are analog reads.\*/

int ldrstateG = analogRead(ldrPinG);

/\*The number controls the level of light or darkness for the photoresistor to trip on the LEDs, it is also controlled by the potentiometer in this project.\*/

if ((ldrstateG <=500 && state==1) || state==2) {

/\*LEDS ON\*/

digitalWrite(ledPin, HIGH);

digitalWrite(ledPin2, HIGH);

digitalWrite(ledPin3, HIGH);

digitalWrite(ledPin4, HIGH);

digitalWrite(ledPin5, HIGH);

/\*For testing.\*/

Serial.println("LDR is DARK, LED is ON");

}

else {

/\*LEDS OFF\*/

digitalWrite(ledPin, LOW);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin3, LOW);

digitalWrite(ledPin4, LOW);

digitalWrite(ledPin5, LOW);

Serial.println("---------------");

}

/\*Reading Photoresistor for White. Photoresistors are analog reads.\*/

int ldrstateW = analogRead(ldrPinW);

/\*The number controls the level of light or darkness for the photoresistor to trip on the LEDs, it is also controlled by the potentiometer in this project.\*/

if ((ldrstateW <=500 && state==1) || state==2) {

/\*LEDS ON\*/

digitalWrite(ledPin6, HIGH);

digitalWrite(ledPin7, HIGH);

digitalWrite(ledPin8, HIGH);

digitalWrite(ledPin9, HIGH);

digitalWrite(ledPin10, HIGH);

}else{

/\*LEDS OFF\*/

digitalWrite(ledPin6, LOW);

digitalWrite(ledPin7, LOW);

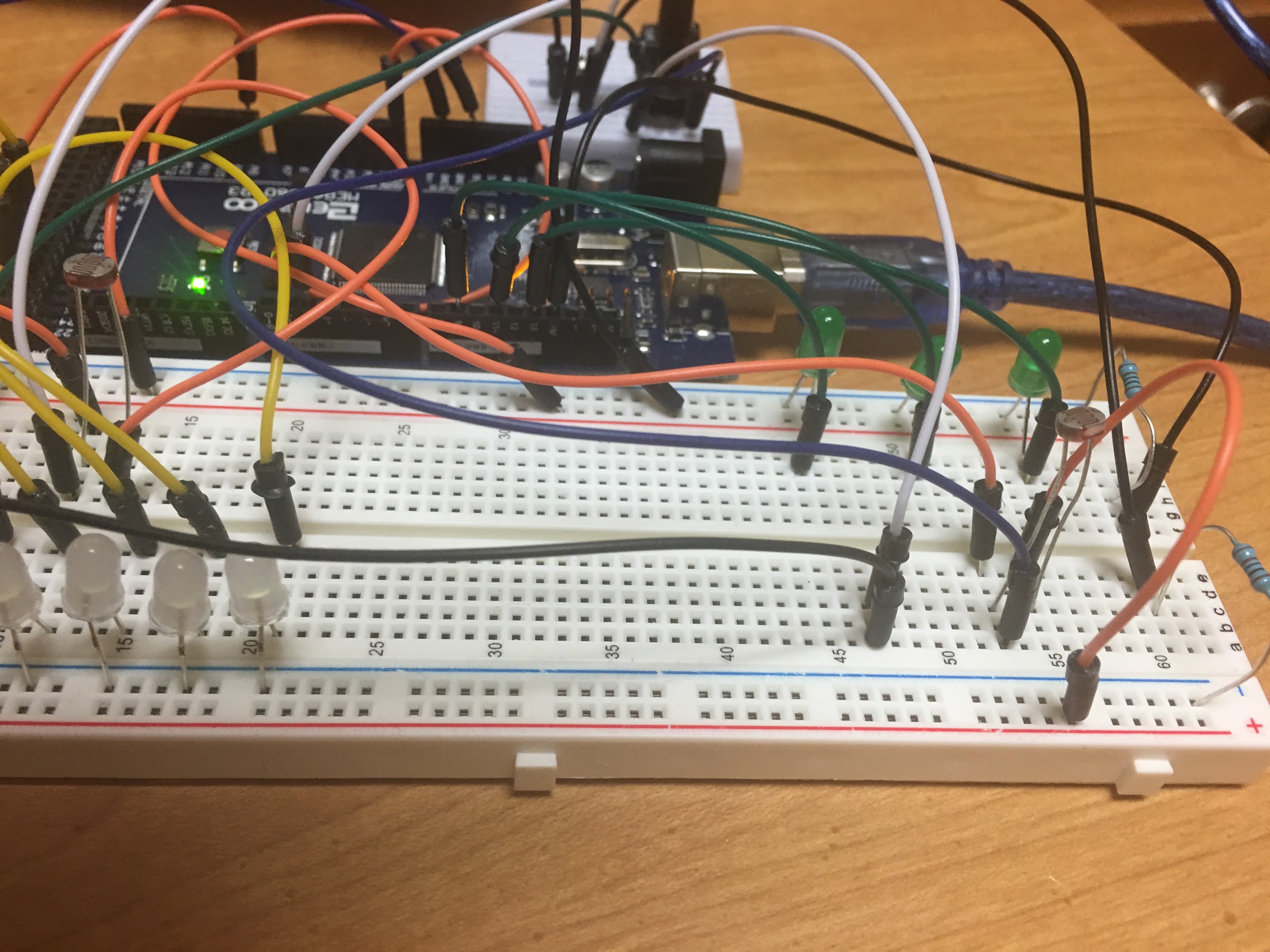
digitalWrite(ledPin8, LOW);

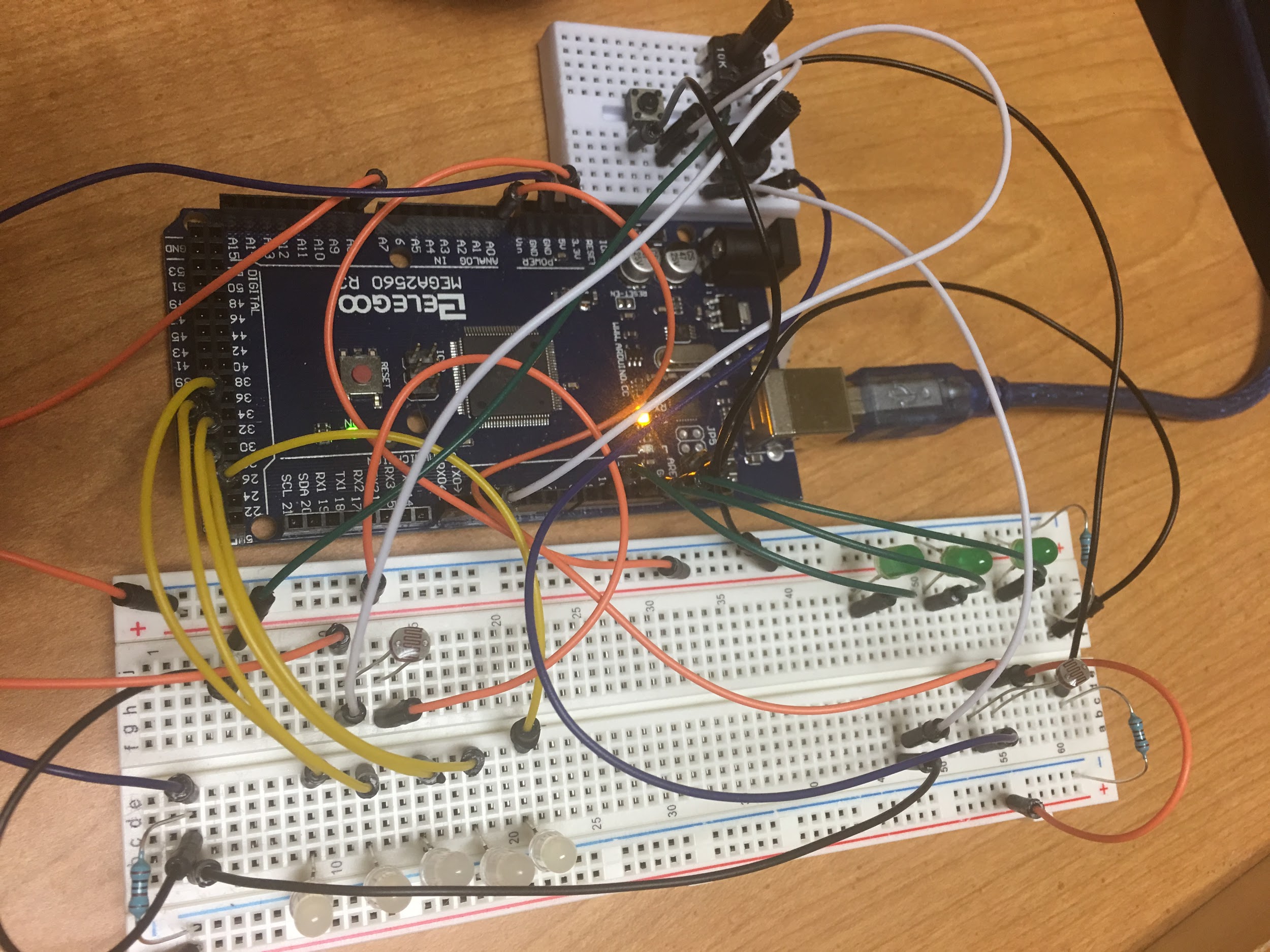
digitalWrite(ledPin9, LOW);

digitalWrite(ledPin10, LOW);

Serial.println("---------------");

}

Final Hardware Design Choices:



The images above show all of my final design hardware. I decided to use a button and two potentiometers. I used two photoresistors, each controlled by a potentiometer and the button. I removed the white LEDs from the Green side, simply due to the fact that they were not working properly, however, I have left the code in the program for the white LEDs to be added back into the setup as some point in time. I ended up using the positive line for the button instead of turning pin 50 into a 5-volt pin. I used resistors on both negative lines to directly connect the LEDs and use less jumper wires. I also decided to have two photoresistors for my demonstration due to the design stated in Paper #4. Green LEDs were used to show the customization factor and how they could be used not just as normal lights, but they could be used in urban farming or greenhouses.