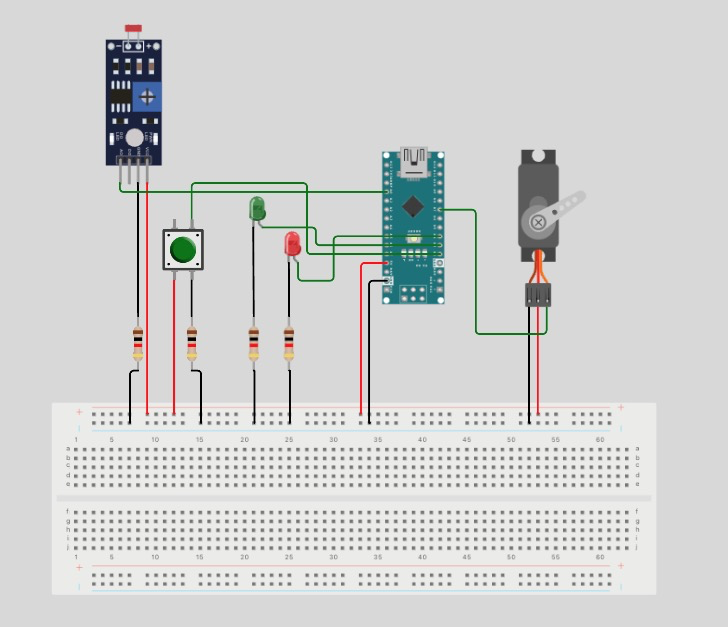
Setup



Writeup

I had issues with pins on both the simulator (wokwi.com) and my physical implementation with an Arduino Uno R3. Due to this issue I had to change the button pin to 2, green LED pin to 3, and red LED pin to 4.

This code is designed to control a servo motor, a photoresistor, and a set of LEDs using an Arduino microcontroller. The code includes two modes of operation: a light meter mode and a timer mode.

The code begins by importing the Servo and AbleButtons libraries. It then defines several variables, including the pins used to connect the photoresistor, servo motor, pushbutton, and LEDs to the Arduino. It also defines several other variables, including the current mode of operation, the light threshold for the timer mode, and a counter for the timer mode.

The code then defines a button object and sets up some additional variables to debounce the button input. Several functions are also defined to control the state of the LEDs, move the servo motor, and handle button presses.

In the setup() function, the code initializes the serial monitor, sets the pin modes for the button and LEDs, attaches the servo motor to its pin, and initializes the button and LED states.

The loop() function is then executed repeatedly. Within the loop, the code reads the state of the button and the current value of the photoresistor. Depending on the current mode, the code then either moves the servo motor to a corresponding angle based on the photoresistor value or decrements the counter and adjusts the state of the LEDs based on the current value of the photoresistor and the light threshold.

When the button is pressed, the handle\_button\_pressed() function checks to see if the button has been pressed to toggle between the two modes of operation.

Overall, this code allows for the precise control of a servo motor based on the input from a photoresistor or the use of a timer mode that relies on the input from the photoresistor and a pushbutton to decrease a counter. The LEDs provide visual feedback on the state of the system.

# Outcomes

This project was a success according to the example tests as this system can pass all of them. I decided to split up my code into smaller more digestible functions, this allows my code to be readable and modular. Some design decisions that I had to make were deciding the minimum and maximum light values for the photoresistor. I decided to go with a value of 300 as the maximum when a phone’s flashlight was held right next to the light sensor, and a value of 10 when the sensor was covered with a finger. I decided to clamp the light value between 0 and 300 as the rare circumstances where that value was greater than 300 would cause issues with the servo motor. Overall, this was a great project to dive into the world of Arduinos, sensors, and motors.