Final Project: Balloon Laser Game

ELE 408/409
Professor: Brendan Smerbeck

Colin Davis
100650600
colin_davis@uri.edu
Sean Maglione
100666471

5/9/2022

Introduction:

The purpose of this project is to establish some sort of server/ client connection using raspberry pi. The rest of the project is fairly open ended. For this project we decided to use 2 servos to move a 1W laser in the up/down/left/right directions, in order to aim at an array of balloons to pop them. Then through the use of some hardware we are able to sense when the balloons are popped and upload an updated version of the balloon array to our server.

Balloon Array:

An array of black balloons are held in place by command hooks. Underneath each balloon lays an exposed photoresistor circuit. The point of this circuit is to sense the ambient light ie. sense if a balloon has been popped or not. The data from this circuit is collected by an arduino mega, which then sends that serial data to the Rpi4 via I2C connection.

Controls:

The main controls for this project are moving the laser up, down, left and right, and lastly firing the laser, and some other sub functions. However this is all done through a USB polling connection to the 3D Rumble joystick. Each button on the joystick is mapped to do some sort of function whether it is to move the servos, reset the website, or fire the laser.

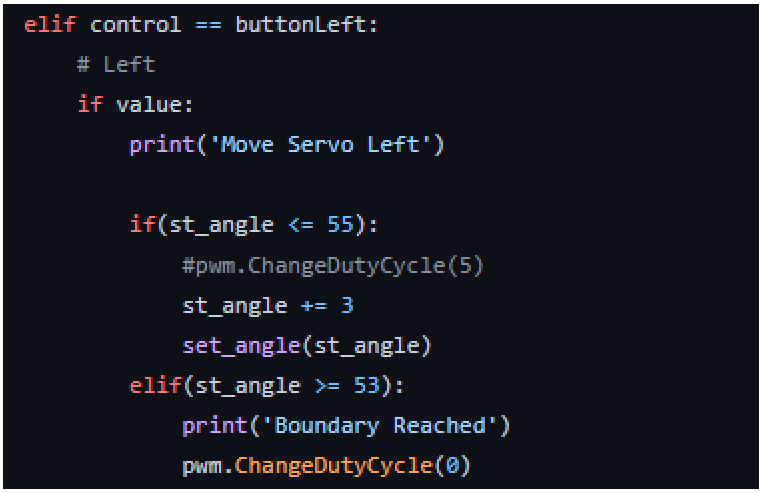
Server:

The server is a locally hosted apache server. This displays an HTML page with the updated balloon array. Every time a balloon is popped a JSON file is changed. The server host (laptop) is SSH-ed into the Rpi4, and then updates the HTML website with the new JSON data.

Externals:

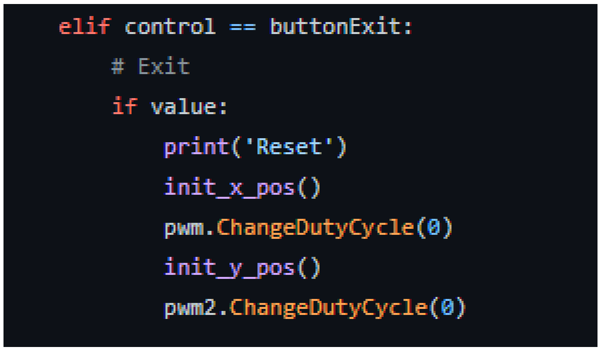
Since the laser is very powerful and has to be aimed before firing, we decided to mount a simple laser pointer onto the side of the main laser to allow the user to get a fairly accurate idea of where the laser will fire upon trigger pull. Also, since the laser itself draws a lot of power, we needed the help from a relay circuit. Since the relay circuit we used was a latching relay circuit we had to be cautious of the timing of firing the laser. (Spamming the trigger could cause issues or burn a chip). Knowing this we added a simple LED circuit that sits on the back of the laser that remains on while the laser is “ready” to fire, and shuts off when the laser is not ready to fire.

Code Snippets:



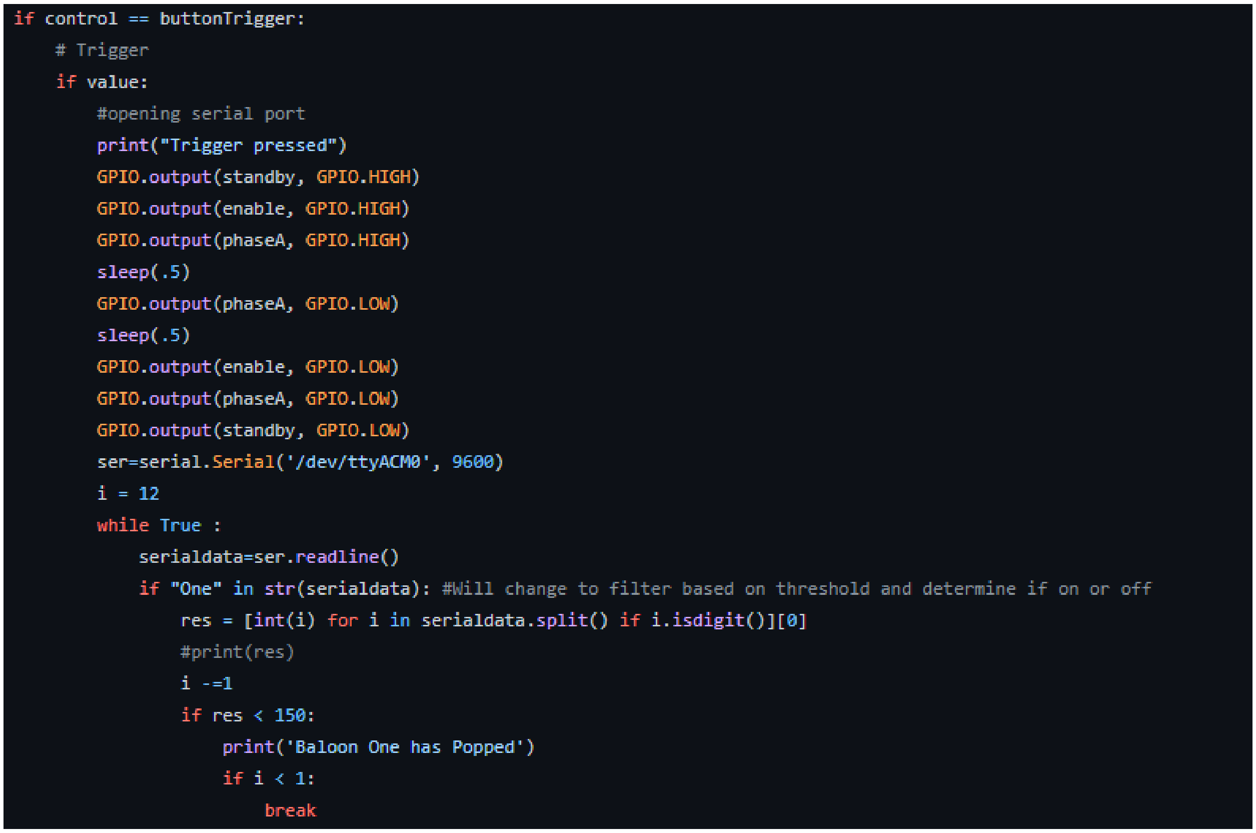
```
elif control == buttonLeft:
    # Left
    if value:
        print('Move Servo Left')

        if(st_angle <= 55):
            #pwm.ChangeDutyCycle(5)
            st_angle += 3
            set_angle(st_angle)
        elif(st_angle >= 53):
            print('Boundary Reached')
            pwm.ChangeDutyCycle(0)
```

**This is an Example of the Servo Left code. Every time the left button is pressed, the servo wil move the laser 3 degrees to the left until it has reached the boundary that we set. **



```
    elif control == buttonExit:
        # Exit
        if value:
            print('Reset')
            init_x_pos()
            pwm.ChangeDutyCycle(0)
            init_y_pos()
            pwm2.ChangeDutyCycle(0)
```

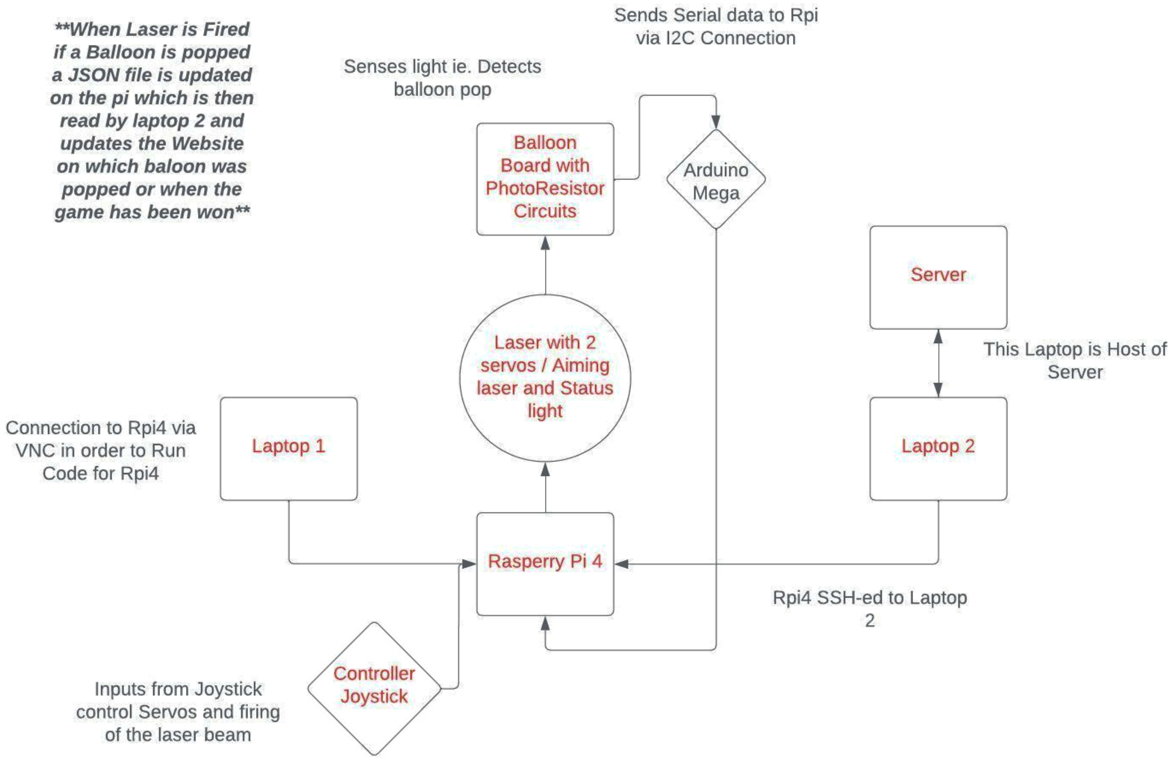
** This is a snippet of code that relates to a reset button. Upon this button being pressed, no matter the location of the horizontal and vertical servo, it will zero out the laser to get back to ground zero. **



```
if control == buttonTrigger:
    # Trigger
    if value:
        #opening serial port
        print("Trigger pressed")
        GPIO.output(standby, GPIO.HIGH)
        GPIO.output(enable, GPIO.HIGH)
        GPIO.output(phaseA, GPIO.HIGH)
        sleep(.5)
        GPIO.output(phaseA, GPIO.LOW)
        sleep(.5)
        GPIO.output(enable, GPIO.LOW)
        GPIO.output(phaseA, GPIO.LOW)
        GPIO.output(standby, GPIO.LOW)
        ser=serial.Serial('/dev/ttyACM0', 9600)
        i = 12
        while True :
            serialdata=ser.readline()
            if "One" in str(serialdata): #Will change to filter based on threshold and determine if on or off
                res = [int(i) for i in serialdata.split() if i.isdigit()][0]
                #print(res)
                i -=1
                if res < 150:
                    print('Baloon One has Popped')
                    if i < 1:
                        break
```

**Code for trigger pull: The first part is for the relay, to turn it on and off fast enough to send a laser shot to pop the balloon it is currently aimed at. Then the Serial Data from the arduino which is sampling extremely fast is read and compared to see if the balloon had been popped or not.

Block Diagram:


When Laser is Fired if a Balloon is popped a JSON file is updated on the pi which is then read by laptop 2 and updates the Website on which baloon was popped or when the game has been won
Senses light ie. Detects balloon pop
Sends Serial data to Rpi via I2C Connection
Balloon Board with PhotoResistor Circuits
Arduino Mega
Server
Laser with 2 servos / Aiming laser and Status light
This Laptop is Host of Server
Connection to Rpi4 via VNC in order to Run Code for Rpi4
Laptop 1
Laptop 2
Rasperry Pi 4
Rpi4 SSH-ed to Laptop 2
Controller Joystick
Inputs from Joystick control Servos and firing of the laser beam

Circuits :

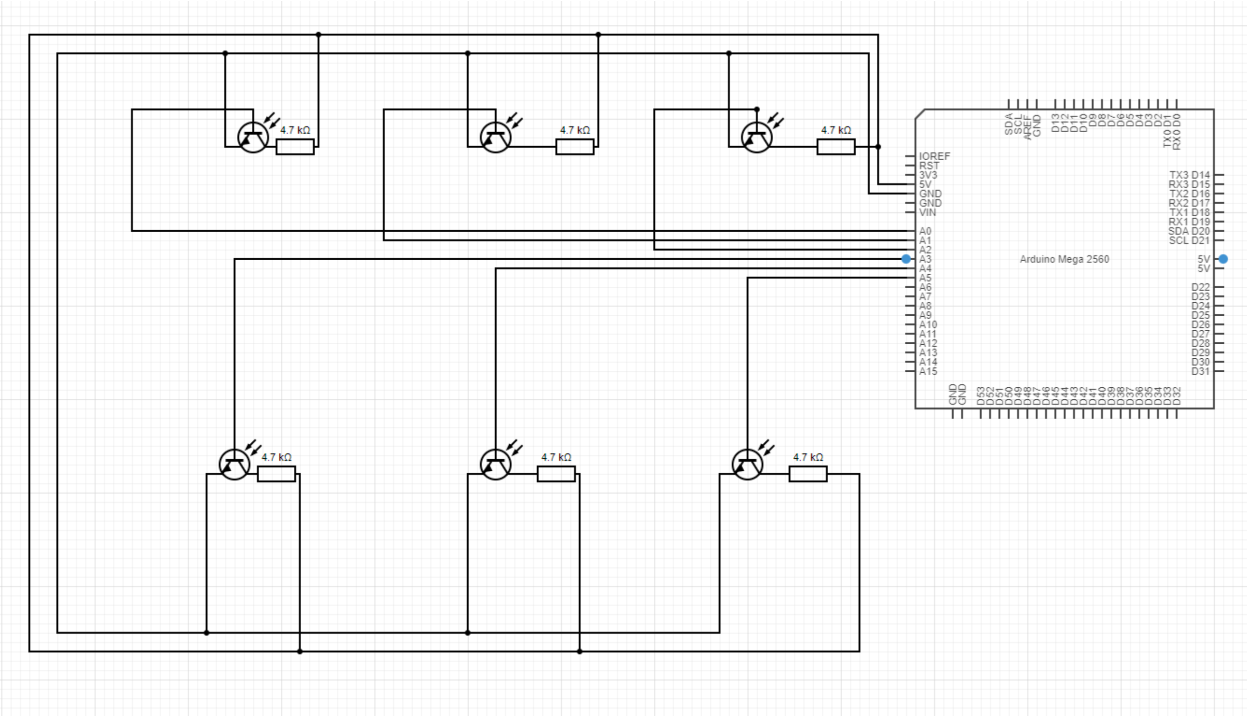


Photo Resistor Circuit Array

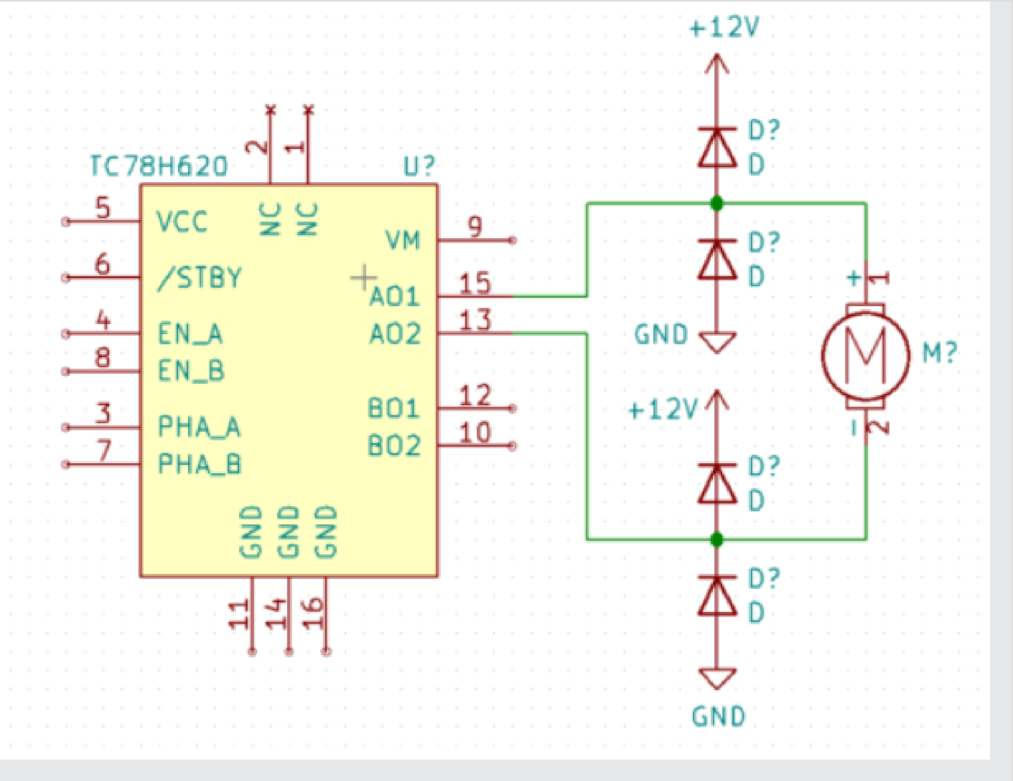


Figure 5: TC78H620 H-Bridge circuit with flyback diodes

Relay Circuit Diagram

Conclusion:

Overall the project was successful. We were able to fire the laser and move the laser in all directions and accurately enough to pop all six balloons on the array. After each trigger pull the current status of the balloon array was then updated to the website and visually displayed for the user to see. In the future, filtering would need to be dont to make the servos move smoother, and a better mounting apparatus would be