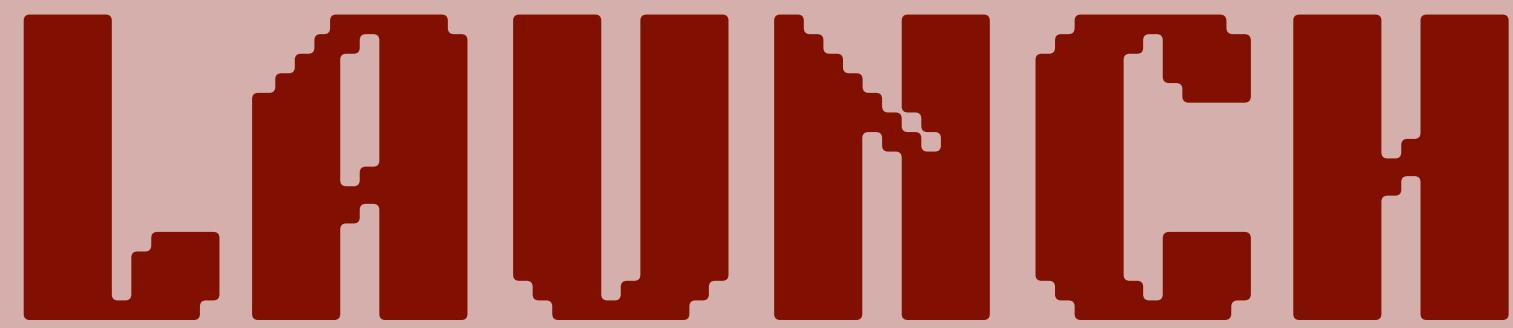


PRESENTING

The word "Launch" is displayed in a pixelated, blocky font. The letters are dark red with white outlines, set against a light gray background. The letters are slightly staggered, creating a sense of depth.

Assessment 1

OPEN DESIGN AND TECHNOLOGY

DIYA AND MAYUKA



**nostalgia lead us
to our idea...**

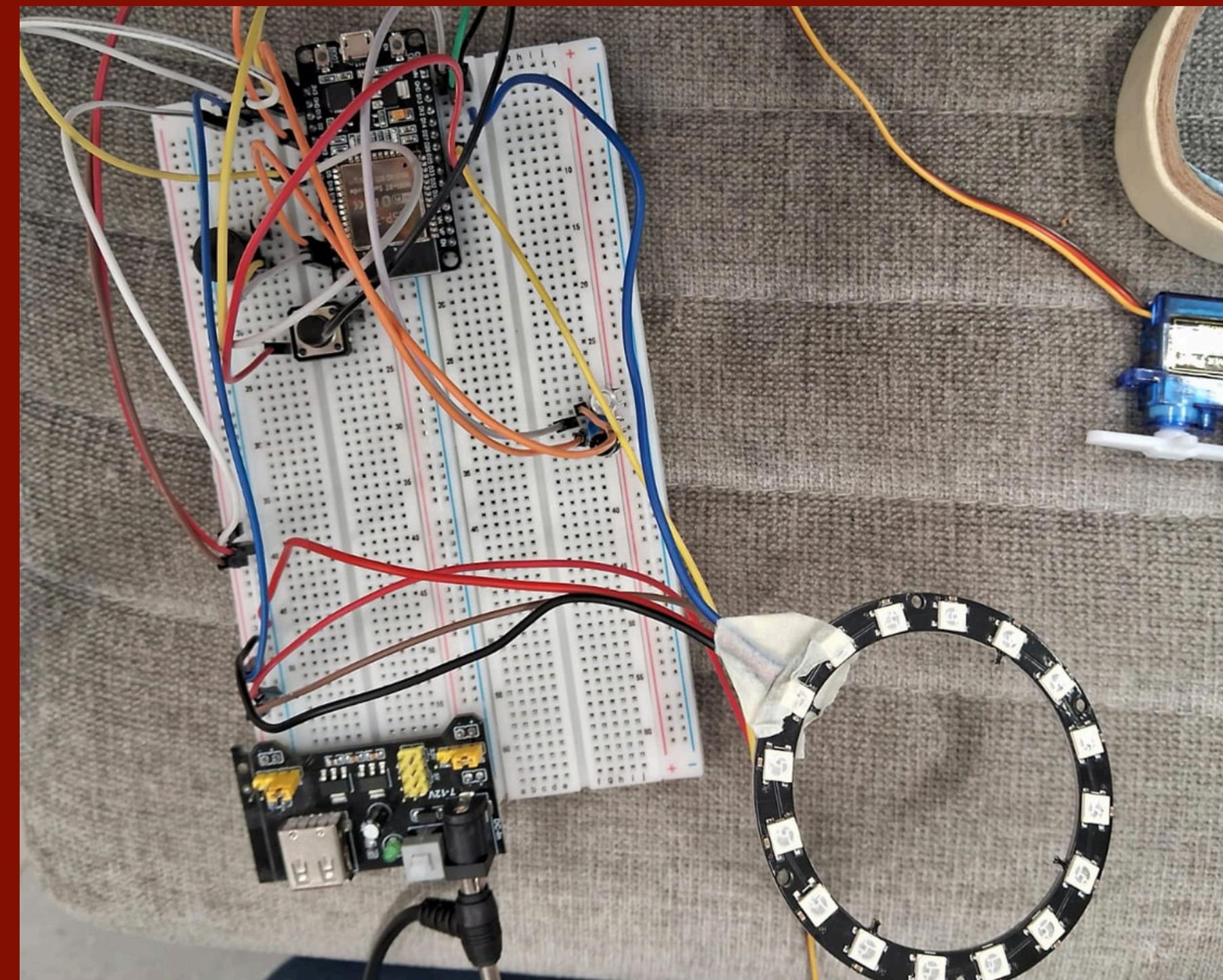
game

brief:

PRESSING THE PUSH BUTTON STARTS THE GAME WITH A BEEP.
TOUCHING THE WIRE SENSORS ACTIVATES THE SERVO MOTORS TO
LAUNCH THE BALL UPWARD. THE GOAL IS TO MAKE THE BALL TOUCH
THE IR SENSOR ON THE WALL. IF THE BALL DOES SUCCESSFULLY HIT
THE IR SENSOR, THE NEOPIXEL LIGHTS UP .

Electrical Components

- 2 Touch Pad (wire sensors)
- 1 PushButton
- 1 Buzzer
- 1 IR sensor
- 2 MicroServo Motors
- Power Module
- 1 Neopixel



Logic

- START BUTTON TRIGGERS GAME STATE AND INITIALIZES SYSTEM
- TOUCH SENSOR ACTS AS INPUT TO ACTIVATE SERVO LAUNCH
- SERVO MOTORS MOVE TO A DEFINED ANGLE TO CREATE CONTROLLED PROJECTILE MOTION
- SYSTEM WAITS FOR IR SENSOR INTERRUPTION WITHIN A SHORT DETECTION WINDOW
- IF IR SENSOR IS TRIGGERED → SUCCESS OUTPUT (NEOPIXEL ON)
- IF NOT TRIGGERED → FAIL STATE (NO LIGHT)
- HIDDEN IR PLACEMENT ADDS UNCERTAINTY AND TRIAL-BASED LEARNING

START TRIGGER



LAUNCH ACTIVATION



BALL PROPULSION



IR DETECTION



NEOPIXEL LIGHTS UP (SUCCESS)

The Pain Points :

- SERVO TIMING MISMATCH CAN AFFECT LAUNCH ACCURACY
- BALL TRAJECTORY MAY BE INCONSISTENT DUE TO FORCE VARIATION
- IR SENSOR MAY FALSELY TRIGGER DUE TO AMBIENT LIGHT
- IR SENSOR ALIGNMENT IS CRITICAL
- BALL MIGHT BOUNCE AND MISS SENSOR
- POWER SUPPLY INSTABILITY CAN CAUSE SERVO JITTER
- TOUCH SENSOR SENSITIVITY MAY CAUSE ACCIDENTAL LAUNCHES

```
1 from machine import Pin , TouchPad, PWM
2 import neopixel
3 import time
4
5 neo = neopixel.NeoPixel(Pin(12, Pin.OUT), 16)
6 buzz = PWM(Pin(23))
7 buzz.duty(0)
8 pb = Pin(32, Pin.IN, Pin.PULL_UP)
9 sm = PWM(Pin(25), freq=50)
10 sm2 = PWM(Pin(5), freq=50)
11 tp = TouchPad(Pin(4))
12 tp2 = TouchPad(Pin(13))
13 ir = Pin(22, Pin.IN, Pin.PULL_UP)
14 freq = [400, 800, 1200, 2000]
15 a = freq[0]
16 b = freq[1]
17 c = freq[2]
18 d = freq[3]
19 count = 0
```

```
21 while True :
22     pbv = pb.value()
23
24     if pbv == 0 :
25         buzz.freq(a)
26         buzz.duty(768)
27         time.sleep(0.6)
28         buzz.duty(0)
29         time.sleep(0.15)
30
31         buzz.freq(b)
32         buzz.duty(768)
33         time.sleep(0.6)
34         buzz.duty(0)
35         time.sleep(0.15)
36
37         buzz.freq(c)
38         buzz.duty(768)
39         time.sleep(0.6)
40         buzz.duty(0)
41         time.sleep(0.15)
42
43         buzz.freq(d)
44         buzz.duty(768)
45         time.sleep(0.6)
46         buzz.duty(0)
47         time.sleep(0.3)
48         buzz.duty(0)
49         time.sleep(1)
50
51     tpv = tp.read()
52     tp2v = tp2.read()
53     irv = ir.value()
54
55
56     if tpv < 200 :
57         count = count + 1
58
59     print(count)
60     for i in range(120,22,60) :
```

CODE

```
61         sm.duty(i)
62         print(i)
63         time.sleep(0.15)
64
65     for x in range(22,120,60) :
66         sm.duty(x)
67         print(x)
68         time.sleep(0.15)
69         time.sleep(1)
70
71     if tp2v < 200 :
72         count = count + 1
73
74     print(count)
75
76     for r in range(120,22,60) :
77         sm2.duty(r)
78         print(r)
79         time.sleep(0.15)
80
```

```
81         for y in range(22,120,60) :
82             sm2.duty(y)
83             print(y)
84             time.sleep(0.15)
85             time.sleep(1)
86
87
88     if irv == 0 :
89         print("goal scored")
90         for n in range(0, 16, 1) :
91             print(n)
92             neo[n] = (0, 0, 255)
93             neo.write()
94             time.sleep(0.1)
95
96     else :
97         for y in range(16) :
98             neo[y] = (0,0,0)
99             neo.write()
100
```

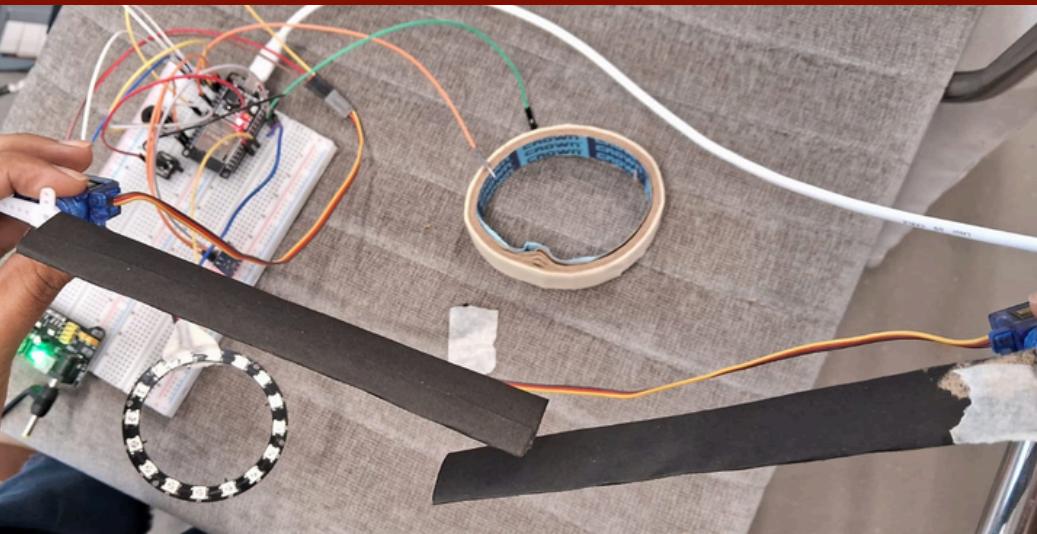
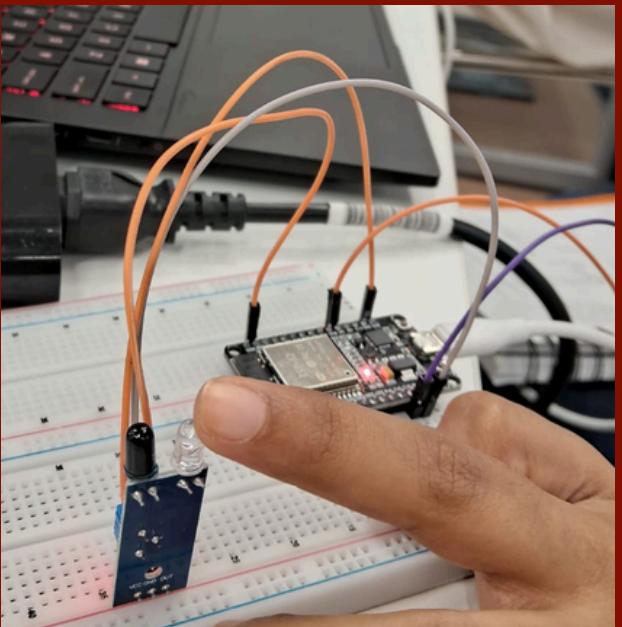
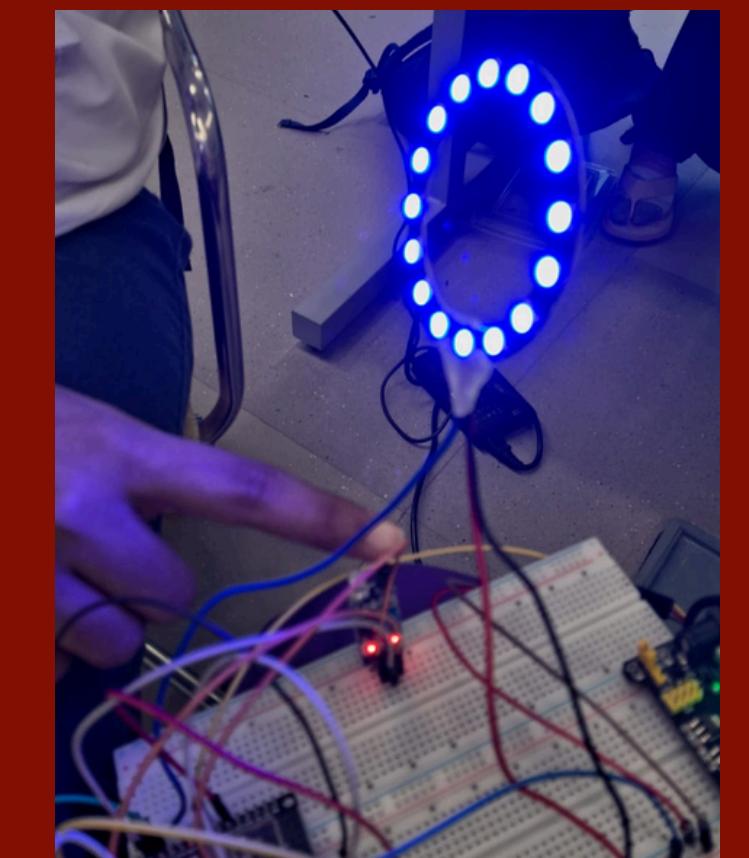
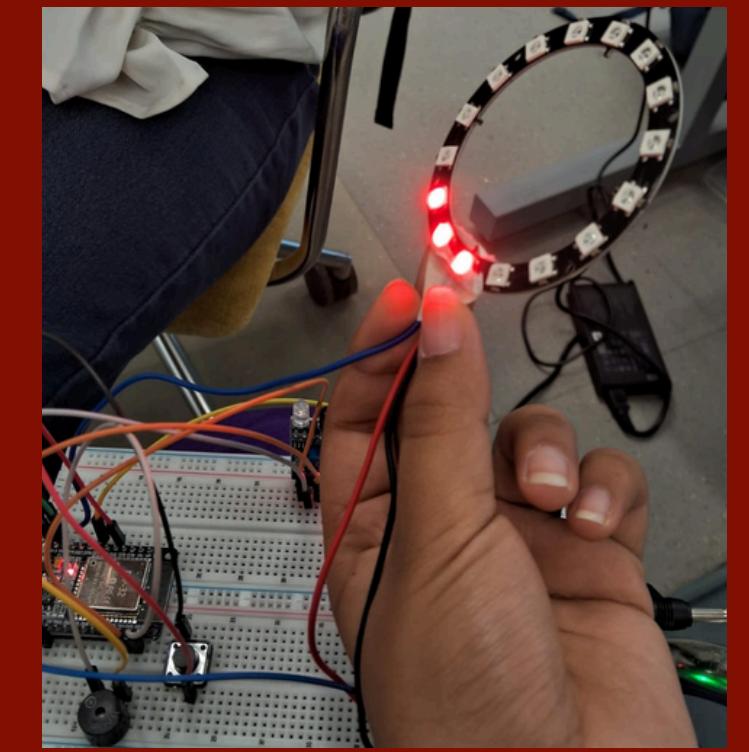
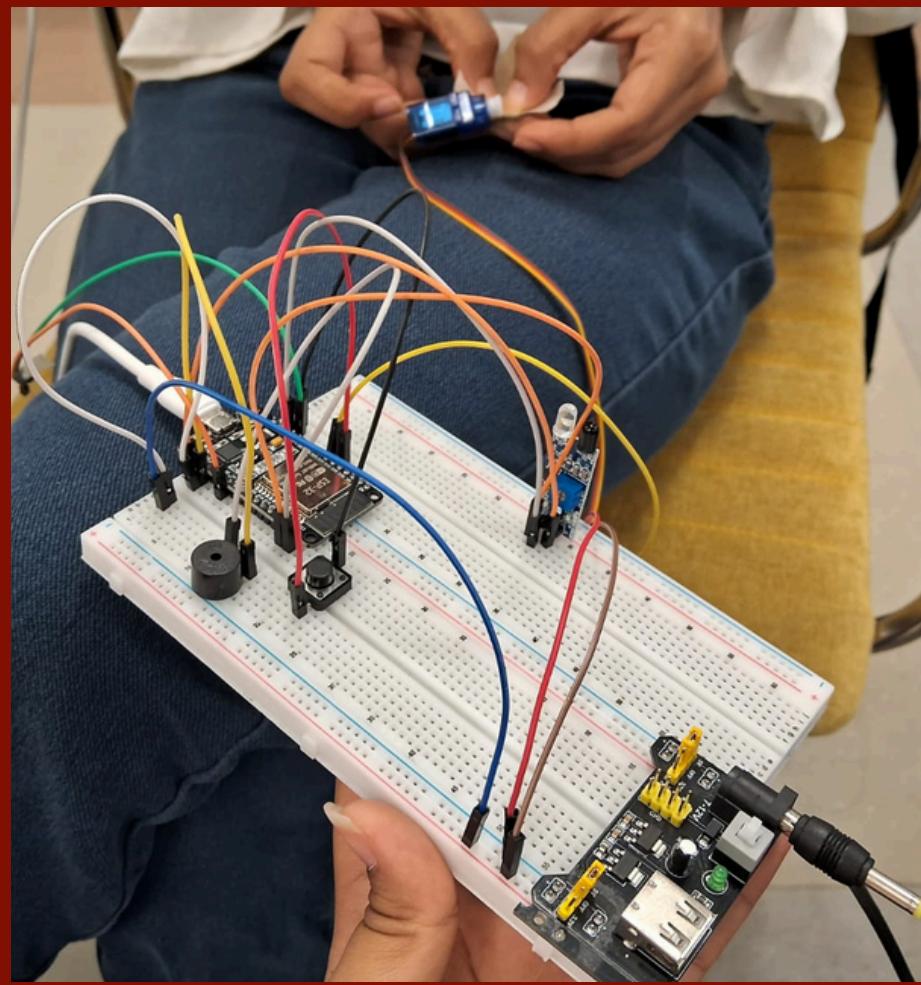
The Working (testing)

[Video link-](#)

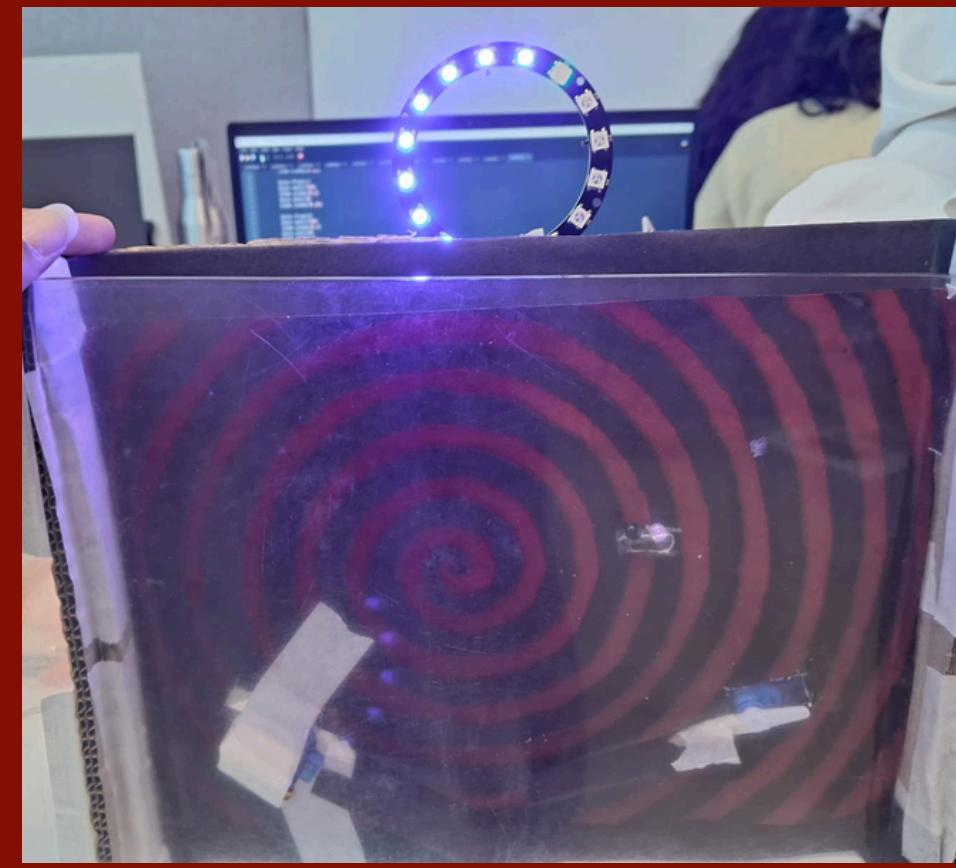
[https://youtube.com/shorts/
ECbYcy1cSqw?feature=share](https://youtube.com/shorts/ECbYcy1cSqw?feature=share)

```
1  from machine import Pin, TouchPad
2
3  import time
4
5  my_touch_pin = TouchPad(Pin(4))
6  tp = TouchPad(Pin(13))
7
8  while True:
9      tpv = tp.read()
10     touch_pin_values = my_touch_pin.read()
11     print(touch_pin_values)
12     if touch_pin_values < 200 :
13
14         print("touch detected")
15     if tpv < 200 :
16         print("touch 2")
17
18     time.sleep(0.2)|
```

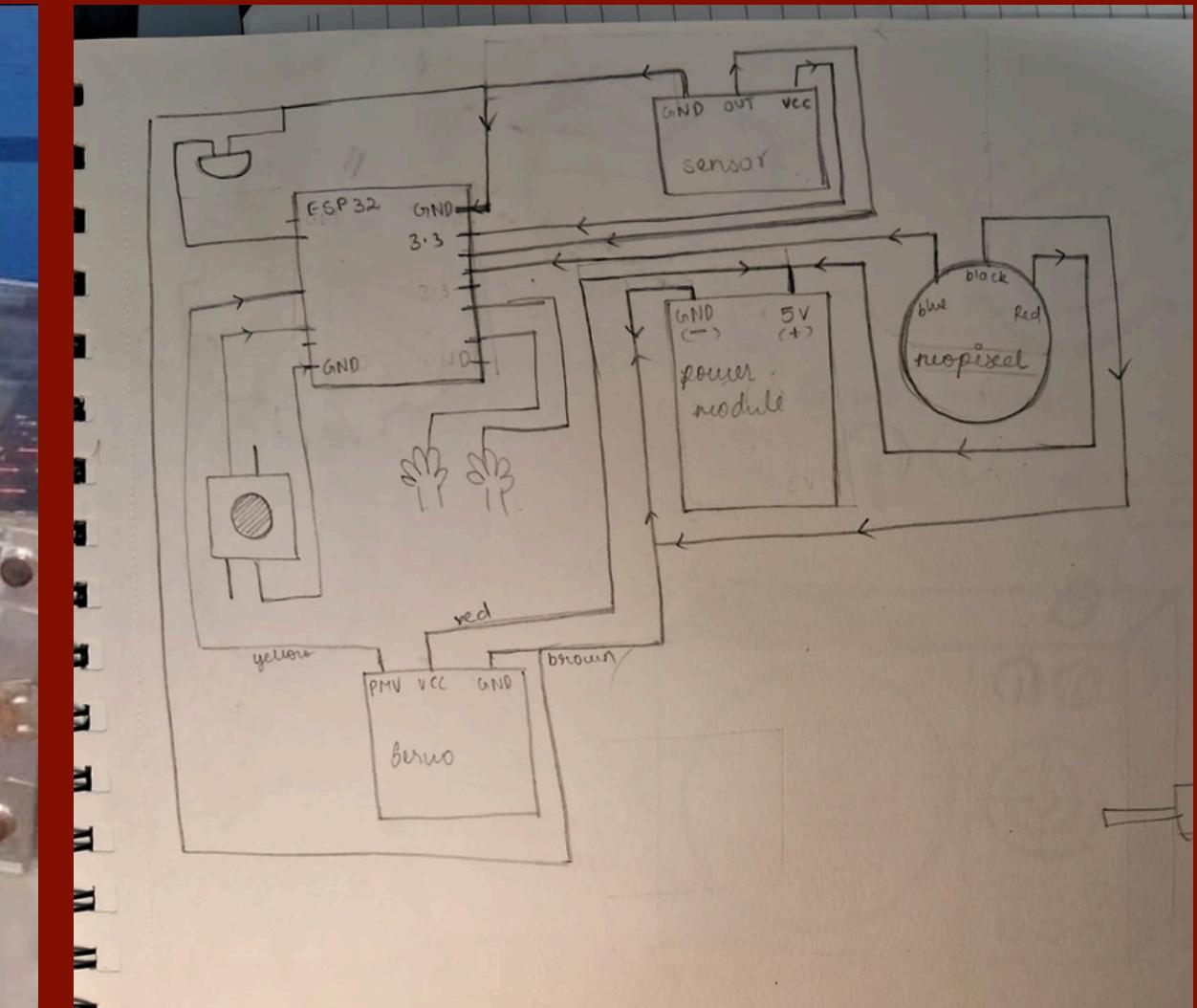
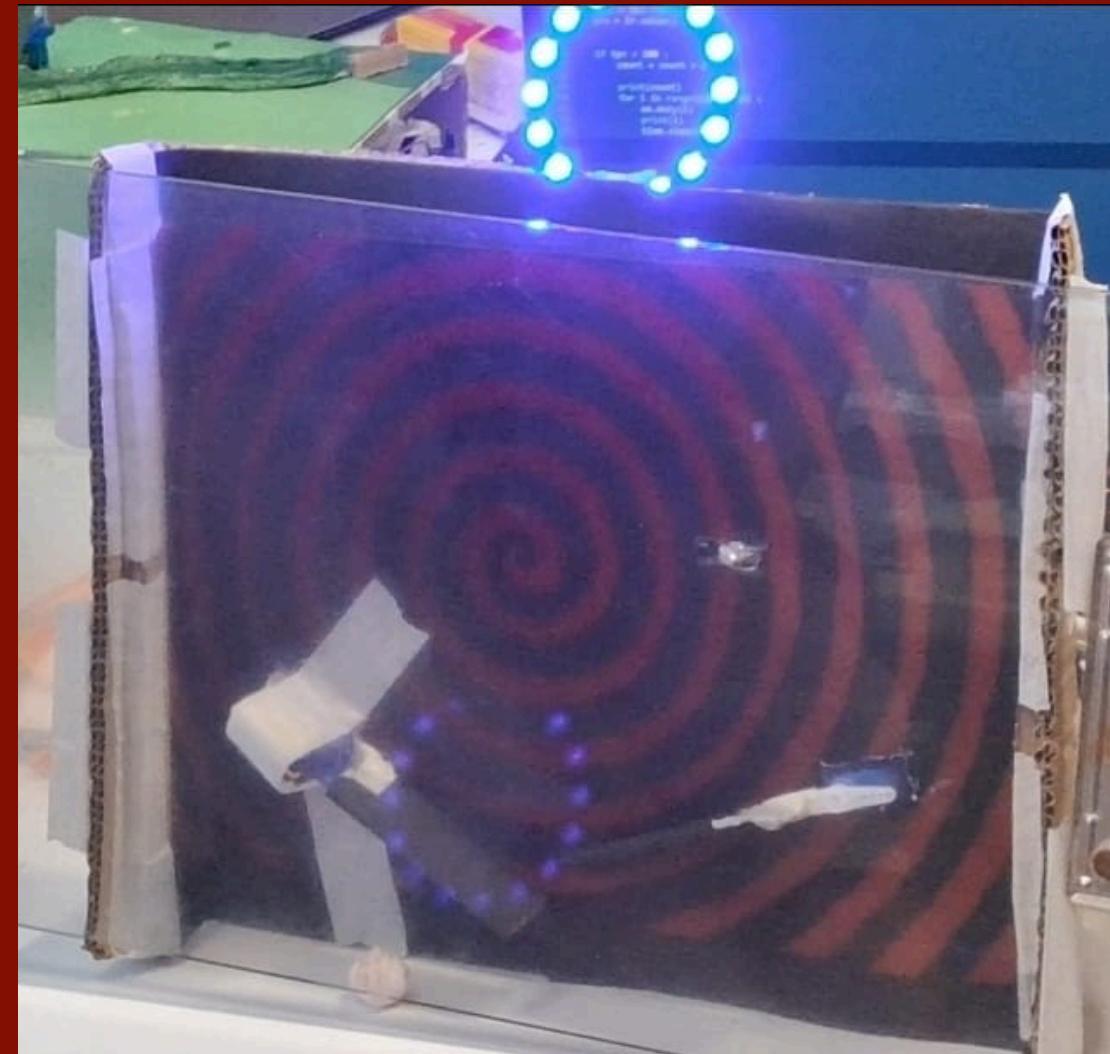
```
Shell x
touch detected
120
touch detected
158
touch detected
437
511
510
510
touch 2
511
```



INITIAL STAGES: TESTING



NEOPIXEL LIGHTING



FINAL PRODUCT

SHAFTS OF THE MOTOR

Final Working

**Video link-
<https://youtu.be/2C2Q7gljvgE>**

Contributions

DIYA- CODES LINE 1-50, CIRCUIT
BUILDING, PPT

MAYUKA- CODES LINE 51-98, CIRCUIT
BUILDING, PPT

THANK YOU