.ISHANI. SAI. RAKSHA.

BASKETBALL

PRO SHOOTERS

ODT

HELLO, WELCOME

Today we’re going to lead you through our ODT project :

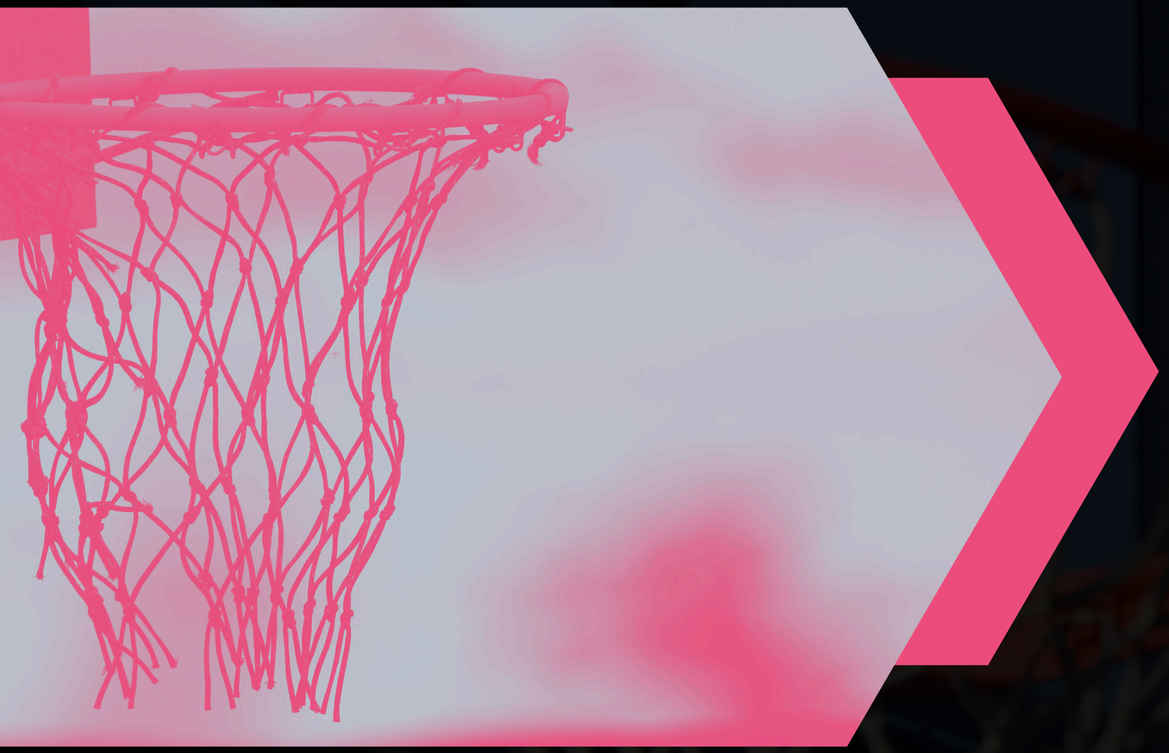
PRO BASKETBALL SHOOTER

Lets Go

WHAT LEAD TO THIS IDEA

This project is a simple interactive system built using MicroPython. We connected a sensor, a buzzer, and a stepper motor to the microcontroller. When the sensor detects an object, the system responds by playing a short victory tune and rotating the motor. So essentially, it demonstrates how an input from a sensor can trigger both sound and motion as outputs.

ODT



WHAT DID WE ACTUALLY MAKE?

Our project is inspired by arcade basketball games. When the ball goes through the hoop, the sensor detects the score, plays a victory sound, and rotates a decorative star using a stepper motor. It creates a small celebration effect every time someone scores

ODT

THINGS WE USED

💡 IR Sensor
Detects light changes so detects our basketball

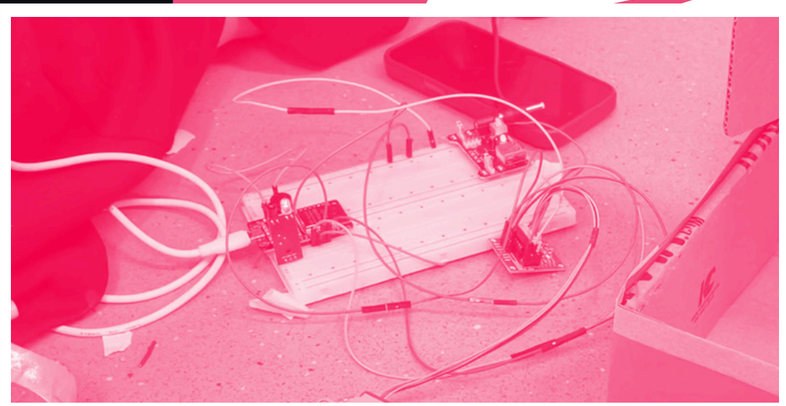
Perfect for laser hit detection
🔔 Buzzer
Gives sound when target is hit
Makes the game interactive

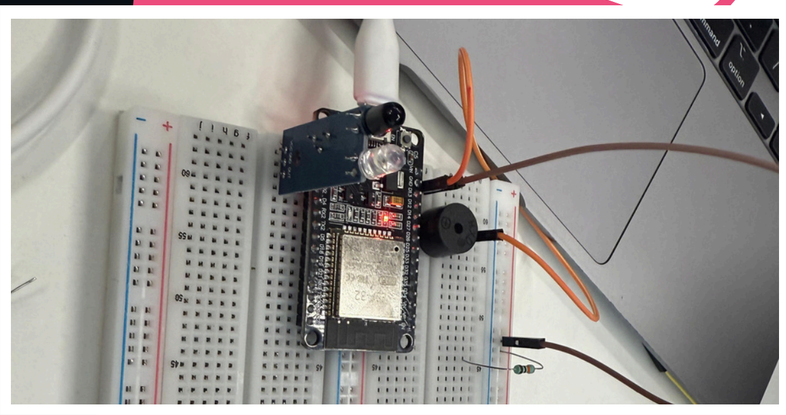
🔴 LED
Shows visual feedback
Indicates successful hit

Sport Presentation

PROGRAM FLOW

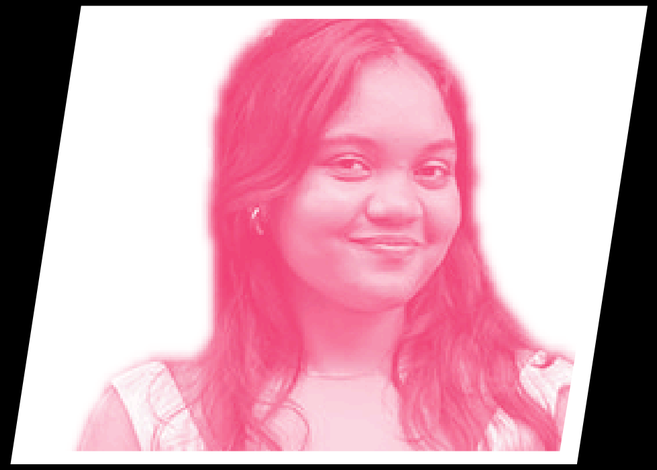
The logic of our program is based on a simple input–process output system.
First, the sensor continuously checks whether a ball has entered the hoop. This is the input.
When the sensor detects the ball, the program processes this condition and triggers two actions.
The first output is sound , the buzzer plays a short victory tune by changing frequencies to create different musical notes.
The second output is motion, the stepper motor rotates in a specific sequence by activating its coils one by one, which makes the attached star spin.
After completing both actions, the system pauses briefly and then goes back to checking the sensor again.
So overall, the program constantly waits for a score and responds with a celebratory sound and rotating star.





ODT

MEET OUR PLAYERS



ishani mitra

ir sensor coding

Circuit Diagram

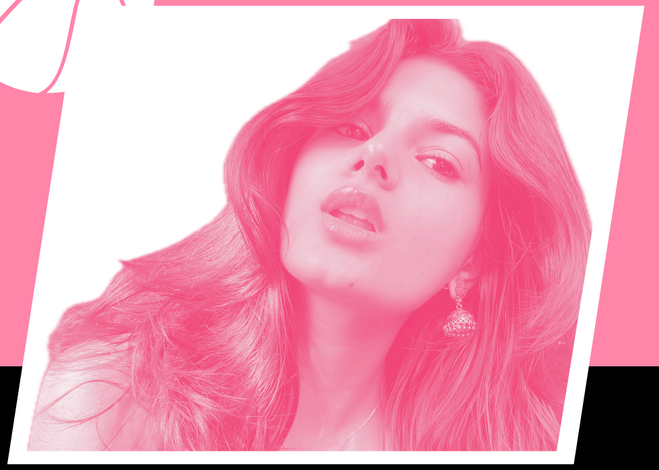
PPT

sai chavre

Stepper motor coding

PPT & Documentation

Debugging



raksha pujari

buzzer coding

PPT & Video

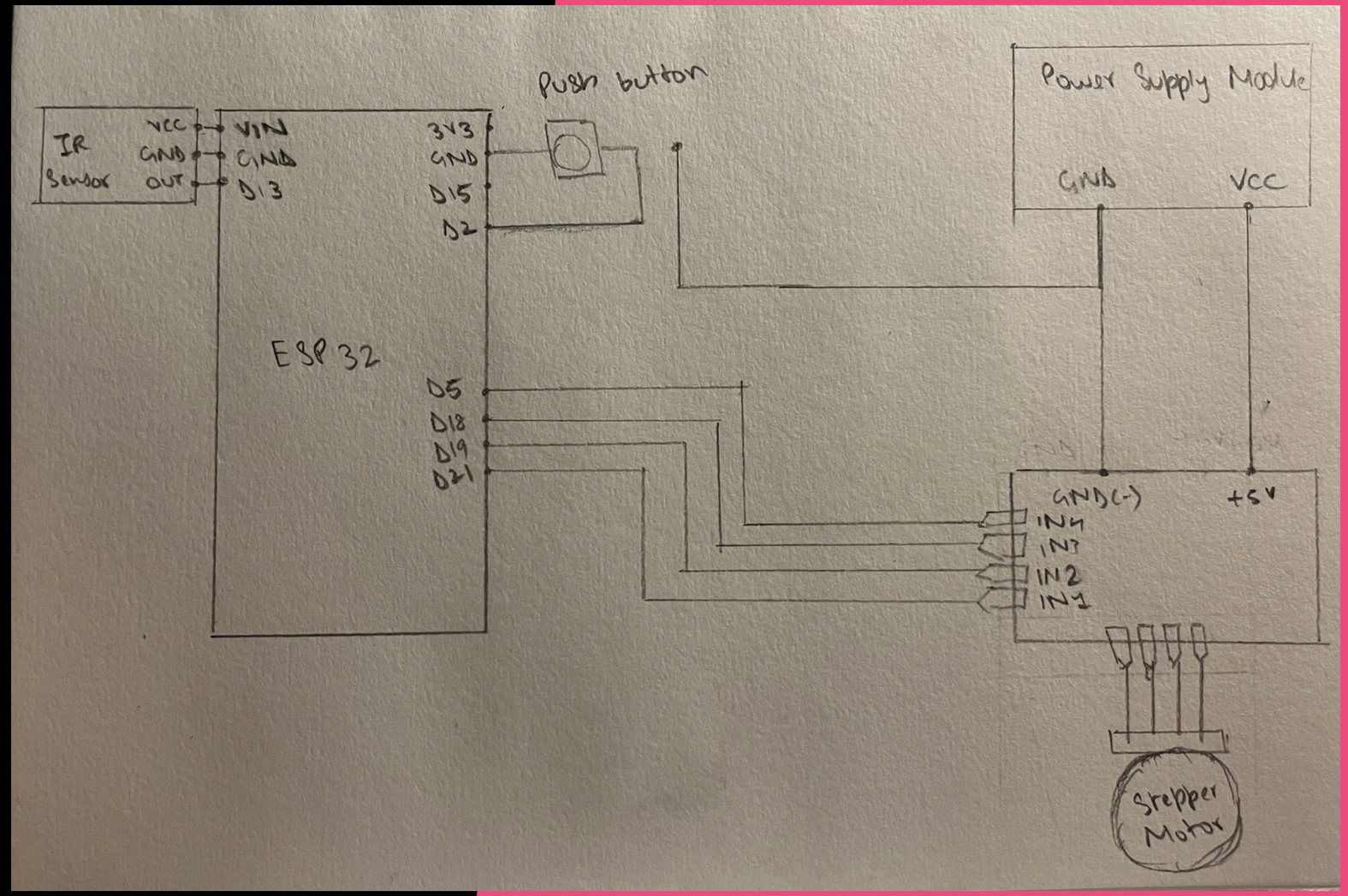
(All three helped in model making and circuit making)

OUR CODE
AND WHAT EVERY LINE MEANS

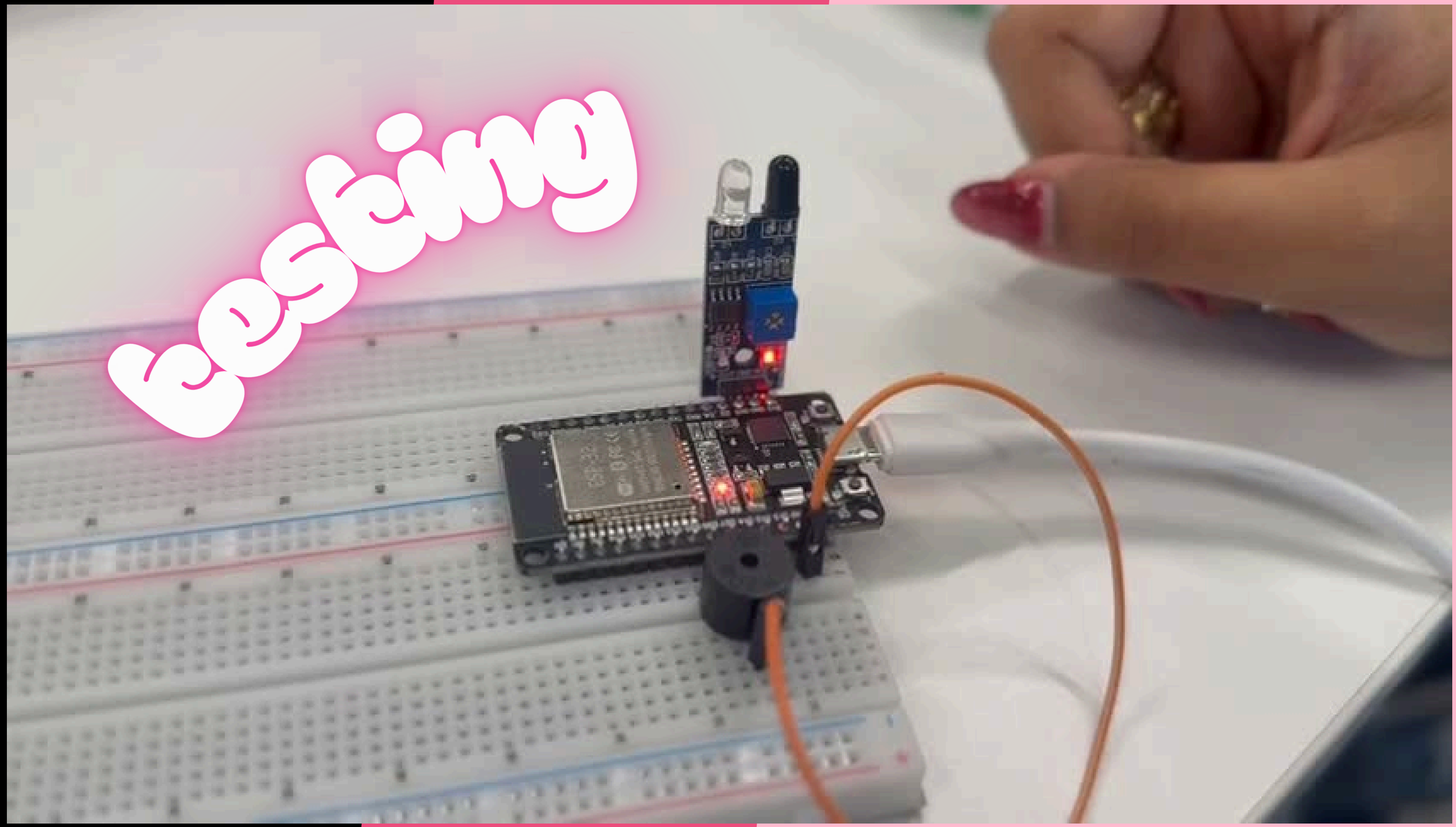
```
from machine import Pin, PWM
import time                     # importing time so we can add delays

buzzer = PWM(Pin(27))
buzzer.duty(0)                  # keep buzzer off at the start

sensor = Pin(15, Pin.IN)        # sensor connected to pin 15 (input to detect ball)

IN1 = Pin(5, Pin.OUT)           # stepper motor wire 1
IN2 = Pin(18, Pin.OUT)          # stepper motor wire 2
IN3 = Pin(19, Pin.OUT)          # stepper motor wire 3
IN4 = Pin(21, Pin.OUT)          # stepper motor wire 4

victory_tune = [523, 659, 784, 1046]   # frequencies of the victory sound (C, E, G, high C)

while True:                     # keep running forever

    if sensor.value() == 0:     # if ball goes inside and sensor detects it

        buzzer.duty(512)        # turn buzzer on (medium volume)

        for note in victory_tune:   # go through each note in the tune
            buzzer.freq(note)       # change pitch to that note
            time.sleep(0.2)         # play each note for 0.2 seconds

        buzzer.freq(1046)       # play the last high note again
        time.sleep(0.5)         # hold it longer for dramatic effect
        buzzer.duty(0)          # turn buzzer off
```

```
    for i in range(200):   # rotate motor 200 times (controls how much it spins)

        IN1.value(1)       # turn on coil 1
        IN2.value(0)
        IN3.value(0)
        IN4.value(0)
        time.sleep(0.005)  # tiny delay so motor can move

        IN1.value(0)
        IN2.value(1)       # turn on coil 2
        IN3.value(0)
        IN4.value(0)
        time.sleep(0.005)

        IN1.value(0)
        IN2.value(0)
        IN3.value(1)       # turn on coil 3
        IN4.value(0)
        time.sleep(0.005)

        IN1.value(0)
        IN2.value(0)
        IN3.value(0)
        IN4.value(1)       # turn on coil 4
        time.sleep(0.005)

    IN1.value(0)           # turn everything off after spinning
    IN2.value(0)
    IN3.value(0)
    IN4.value(0)
```

CIRCUIT DIAGRAM



Testing



```
from machine import Pin, PWM
import time
buzzer = PWM(Pin(27))
buzzer.duty(0)
victory_tune = [523, 659, 784, 1046]
sensor = Pin(15, Pin.IN)
in1 = Pin(18, Pin.OUT)
in2 = Pin(19, Pin.OUT)
in3 = Pin(5, Pin.OUT)
in4 = Pin(21, Pin.OUT)
seq = [
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, 1]
]

my_delay = 0.005

while True:

    if sensor.value() == 0:

        buzzer.duty(512)


        for note in victory_tune:

            buzzer.freq(note)


            for step in seq:
                in1.value(step[0])
                in2.value(step[1])
                in3.value(step[2])
                in4.value(step[3])
                time.sleep(my_delay)

            time.sleep(0.2)

        buzzer.freq(1046)
        time.sleep(0.5)

        buzzer.duty(0)


        in1.value(0)
        in2.value(0)
        in3.value(0)
        in4.value(0)

        time.sleep(1)
```

WHAT WE MADE

LEARNINGS AND PAIN POINTS

💻 Programming Skills

- Reading analog values from ESP32
- using while true anf if statements
- Controlling motors and buzzers

🤝 Teamwork & Project Management

- Dividing tasks among team members
- Managing time effectively
- Testing and improving the system together

🔌 Hardware Knowledge

- How to connect and power electronic components safely
- How to create a voltage divider using an IR
- Importance of stable power supply

🧠 Problem-Solving

- Fixing incorrect sensor readings
- Adjusting sensitivity for accurate hit detection
- Debugging errors in code