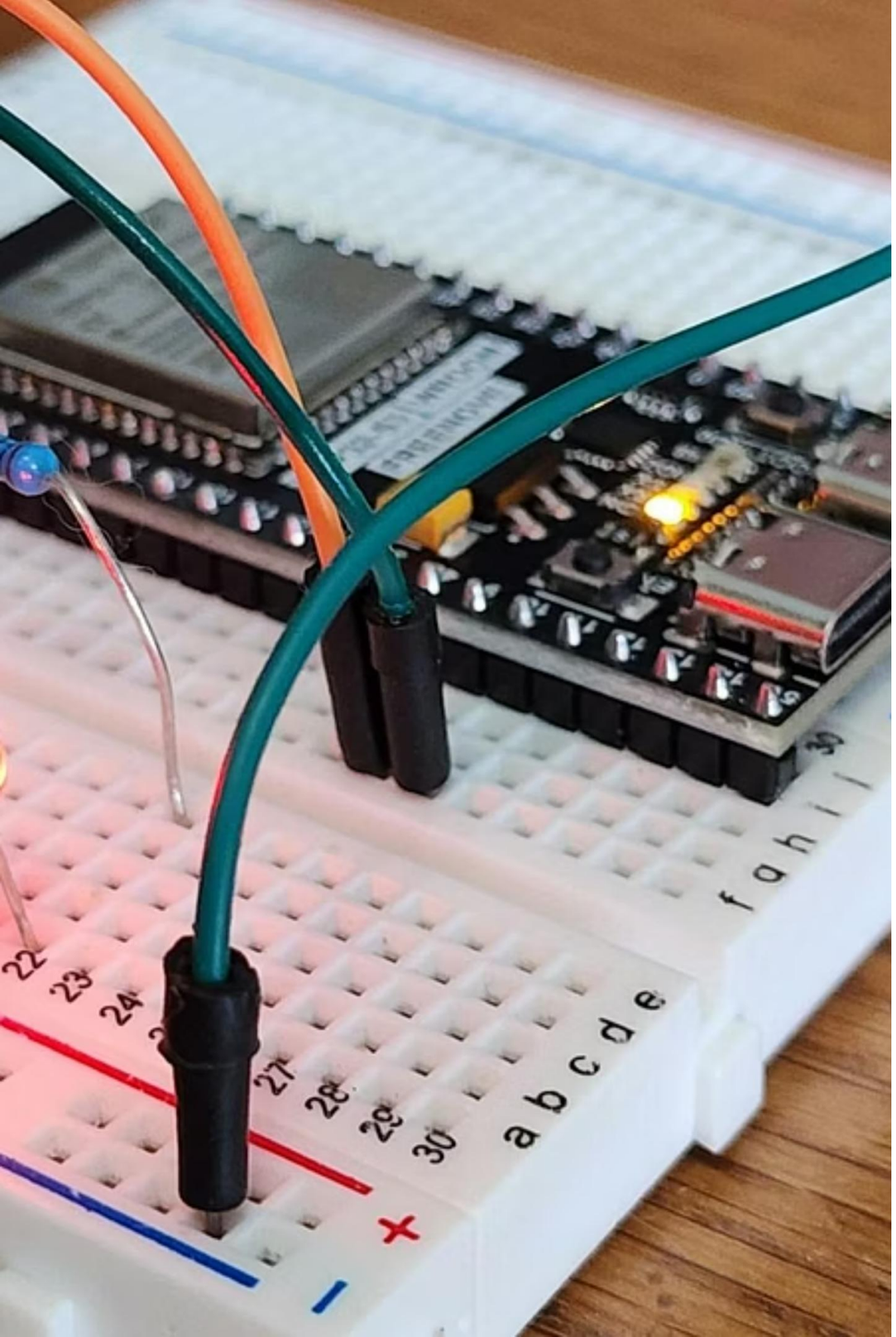


DIP Switch Based Input & Output Control System

A hardware-based control system demonstrating manual configuration of electronic components using DIP switches and Arduino microcontroller

Internet Of Things | Amrita Vishwa Vidyapeetham | Msc Int DataScience

CB.PS.I5DAS22112 - Dharmendra N K R

CB.PS.I5DAS2213 - Eshwar M

What is a DIP Switch?

Definition

A DIP switch (Dual In-Line Package switch) is a set of small manual ON/OFF switches mounted on a circuit board. Each switch provides a binary input (0 or 1) that allows users to configure electronic systems without software programming.

Key Features

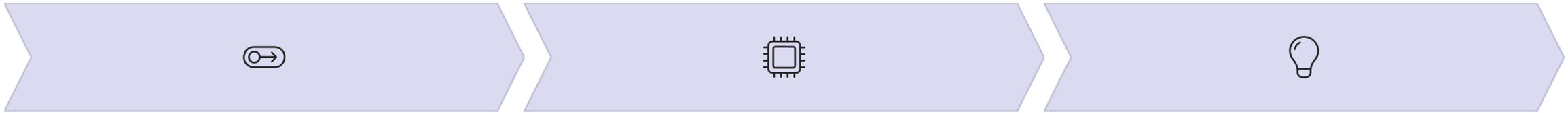
- Manual hardware configuration
- No software required for basic operation
- Binary input generation (0 and 1)
- Compact and reliable design

Binary Representation

ON = 1 (High)
OFF = 0 (Low)

Project Overview

This project demonstrates the use of 2-way and 4-way DIP switches as input devices to control an LED, a servo motor, and a buzzer using an Arduino microcontroller.



Input

2-way and 4-way DIP switches provide manual binary inputs

Processing

Arduino reads switch combinations and processes logic

Output

LED, servo motor, and buzzer respond to switch states

Different switch combinations generate different outputs, enabling manual hardware-based control without reprogramming.

Components Used

2-Way DIP Switch

Two-position manual switch for binary input selection

4-Way DIP Switch

Four-position manual switch providing multiple input combinations

Arduino Board

Microcontroller for processing inputs and controlling outputs

Jumper Wires

Connect components on breadboard and Arduino pins

LED

Visual indicator controlled by Arduino digital output

Servo Motor

Controlled rotational movement based on switch inputs

Flat Disc Buzzer

Audio alert device producing sound when activated

DIP Switch Connections

Power Connection

One side of each switch connected to 5V supply from Arduino

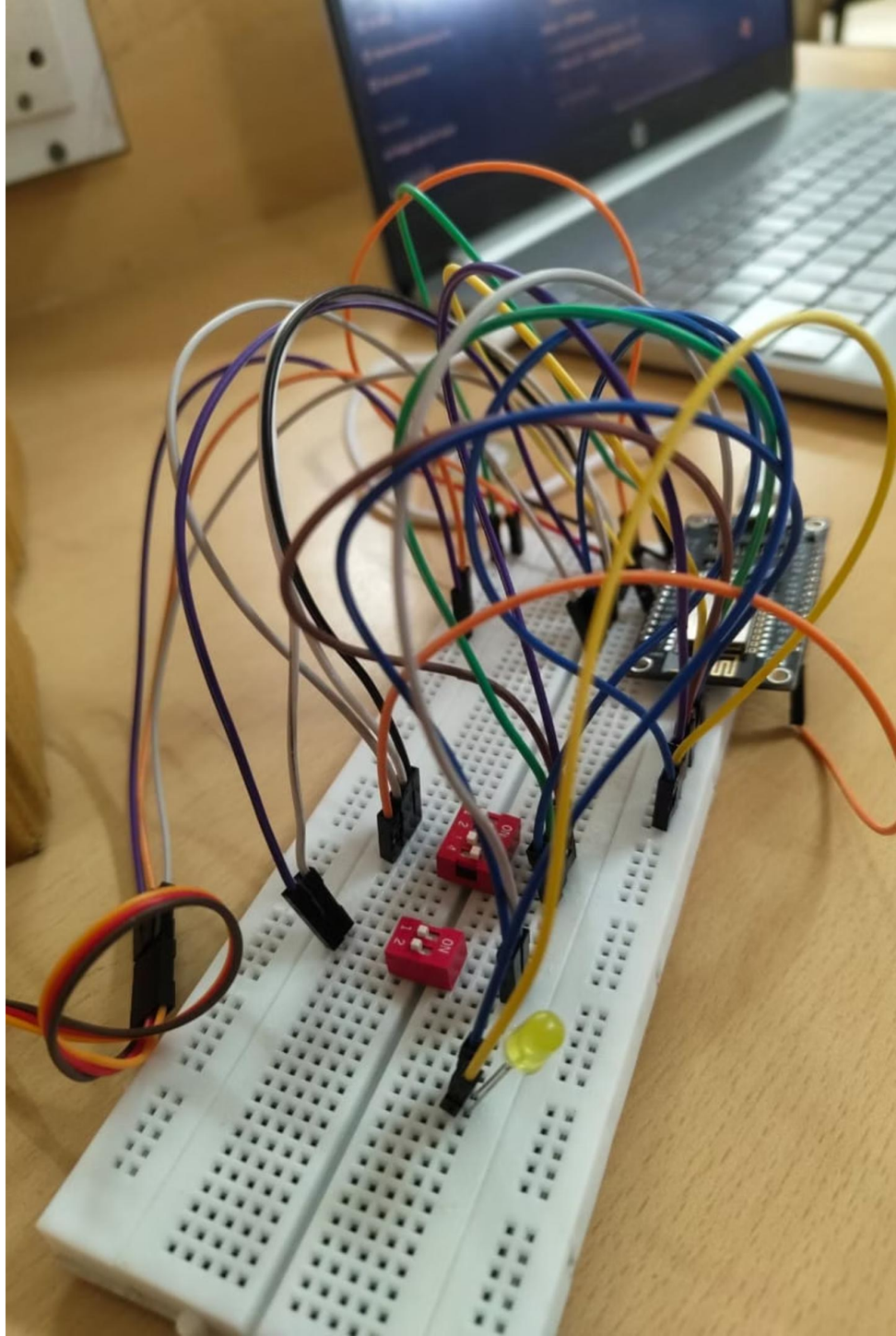
Input Pins

Other side connected to Arduino digital input pins with pull-down resistors

Ground

Pull-down resistors ensure stable LOW state when switch is OFF

This configuration ensures stable and accurate input readings by preventing floating inputs.



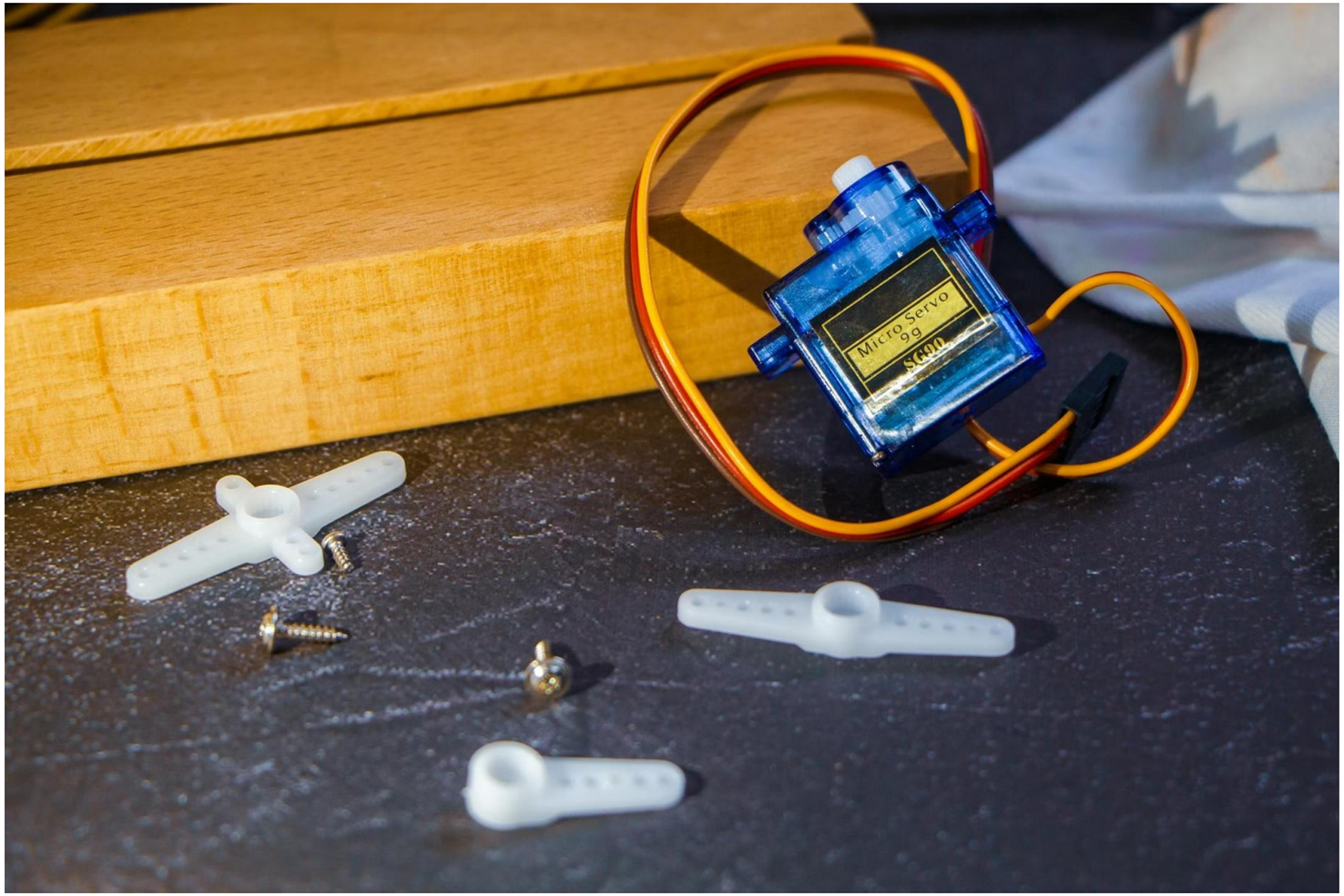
Servo Motor Connection

Wiring Details

- **Signal Pin:** Connected to Arduino PWM pin for angle control
- **VCC:** Connected to 5V power supply
- **GND:** Connected to ground for circuit completion

Operation

Based on DIP switch inputs, the servo rotates to predefined angles. The Arduino reads the binary combination and maps it to specific angular positions.



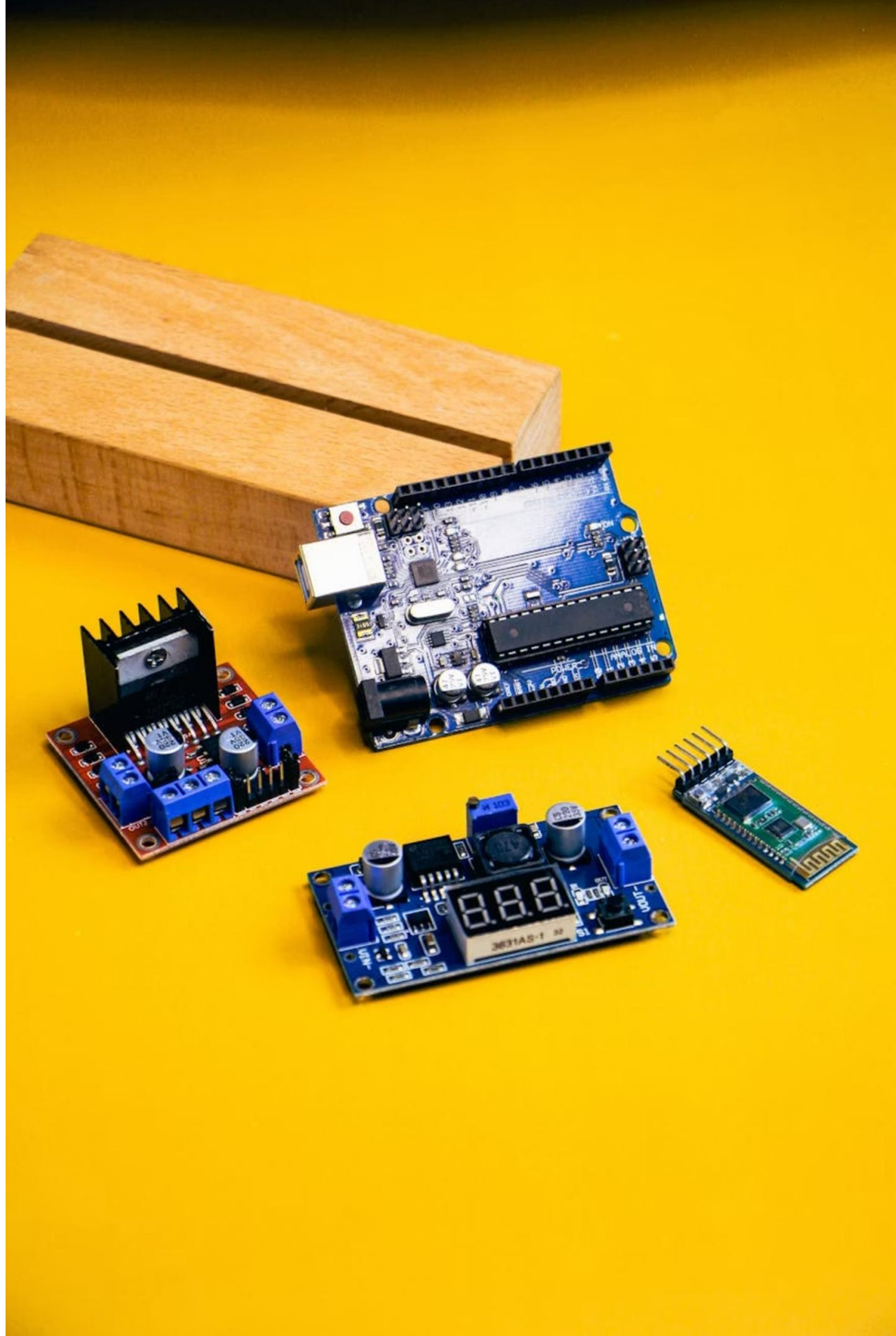
LED and Buzzer Connections

LED Circuit

LED connected through current-limiting resistor to Arduino digital output pin. Resistor prevents excessive current flow and protects the LED from damage.

Buzzer Circuit

Flat disc buzzer connected directly to Arduino digital output pin. Generates audible tone when output pin is set HIGH based on switch inputs.



How the System Works

01

Switch Configuration

User manually sets DIP switches to desired ON/OFF combinations

02

Input Reading

Arduino continuously reads the binary state of all switch positions

03

Logic Processing

Programmed logic interprets switch combinations and determines output actions

04

Output Activation

LED illuminates, servo rotates to specified angle, and buzzer sounds based on inputs

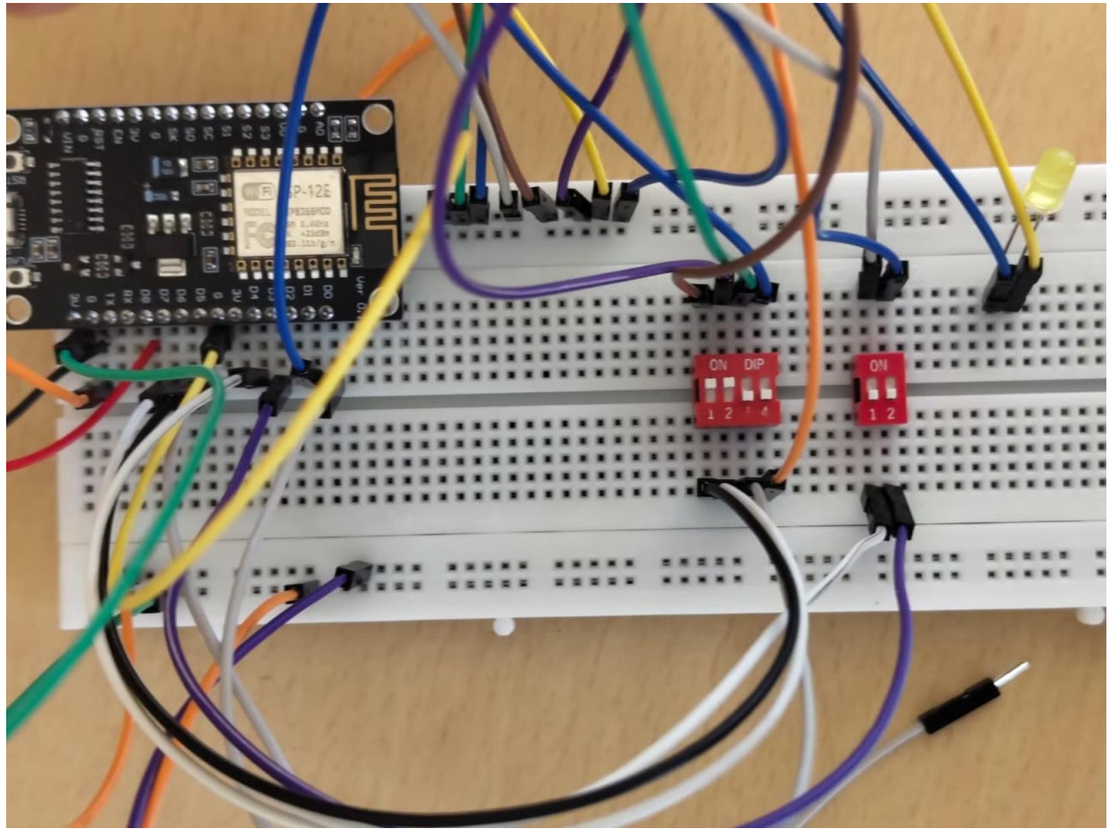
05

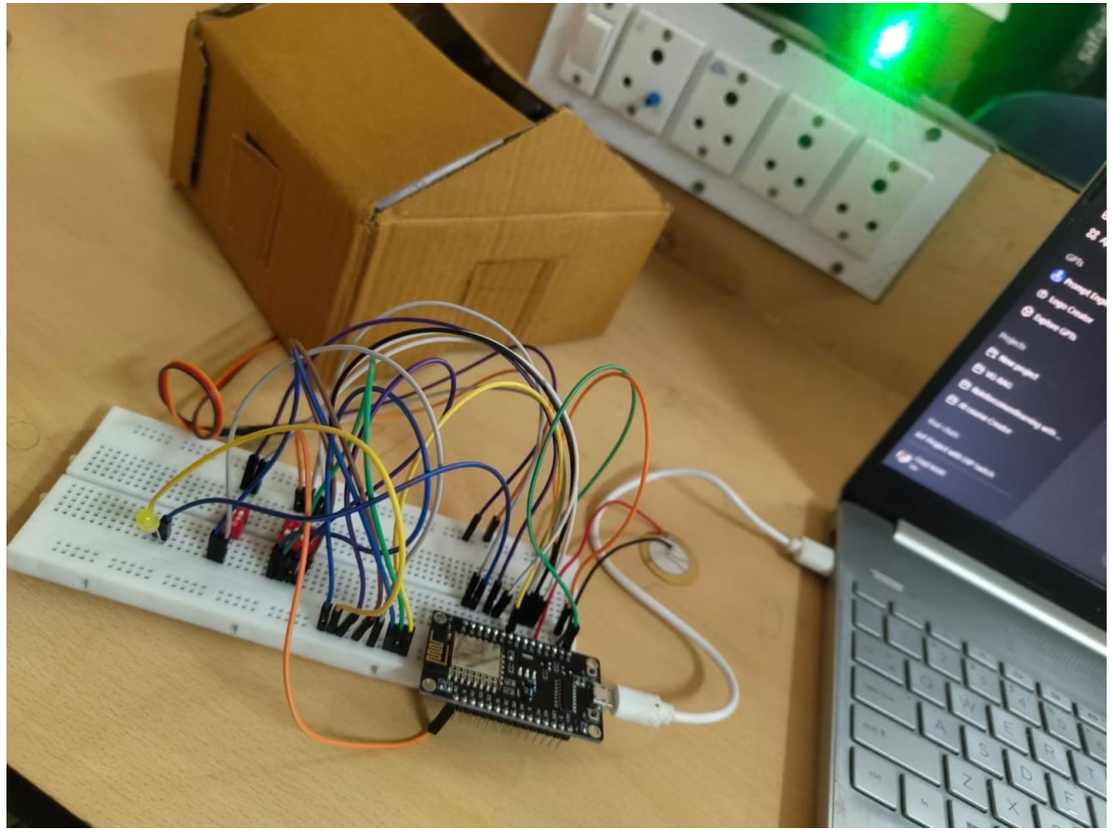
Continuous Monitoring

System continuously monitors switch states for real-time response to changes

C O D E

```
#include <Servo.h>
#define LEDPIN D1
#define PIEZOPIN D8
#define SERVOPIN D5
Servo myservo;
void setup() {
  Serial.begin(115200);
  // -------- Servo --------
  myservo.attach(SERVOPIN);
  myservo.write(0); // locked start
  // -------- 4-way DIP (password) --------
  pinMode(D6, INPUT_PULLUP);
  pinMode(D7, INPUT_PULLUP);
  pinMode(D0, INPUT_PULLUP);
  pinMode(A0, INPUT_PULLUP);
  // -------- 2-way DIP --------
  pinMode(D3, INPUT_PULLUP);
  pinMode(D4, INPUT_PULLUP);
  // -------- outputs --------
  pinMode(LEDPIN, OUTPUT);
  digitalWrite(LEDPIN, LOW);
  noTone(PIEZOPIN);
  Serial.println("System Ready");
}
void loop() {
  // ===== Read Password Switches =====
  int b1 = !digitalRead(D6);
  int b2 = !digitalRead(D7);
  int b3 = !digitalRead(D0);
  int b4 = !digitalRead(A0);
  Serial.print("PASS = ");
  Serial.print(b1);
  Serial.print(b2);
  Serial.print(b3);
  Serial.println(b4);
  // ===== Password Check = 1101 =====
  if (b1==1 && b2==1 && b3==0 && b4==1) {
    myservo.write(180);
    Serial.println("UNLOCKED");
  } else {
    myservo.write(0);
    Serial.println("LOCKED");
  }
  // ===== 2-way DIP Controls =====
  bool light_on = !digitalRead(D3);
  bool buzzer_on = !digitalRead(D4);
  digitalWrite(LEDPIN, light_on ? HIGH : LOW);
  if (buzzer_on) tone(PIEZOPIN, 2000);
  else noTone(PIEZOPIN);
  Serial.print("LED=");
  Serial.print(light_on);
  Serial.print(" BUZZ=");
  Serial.println(buzzer_on);
  Serial.println("------");
  delay(700);
}
```

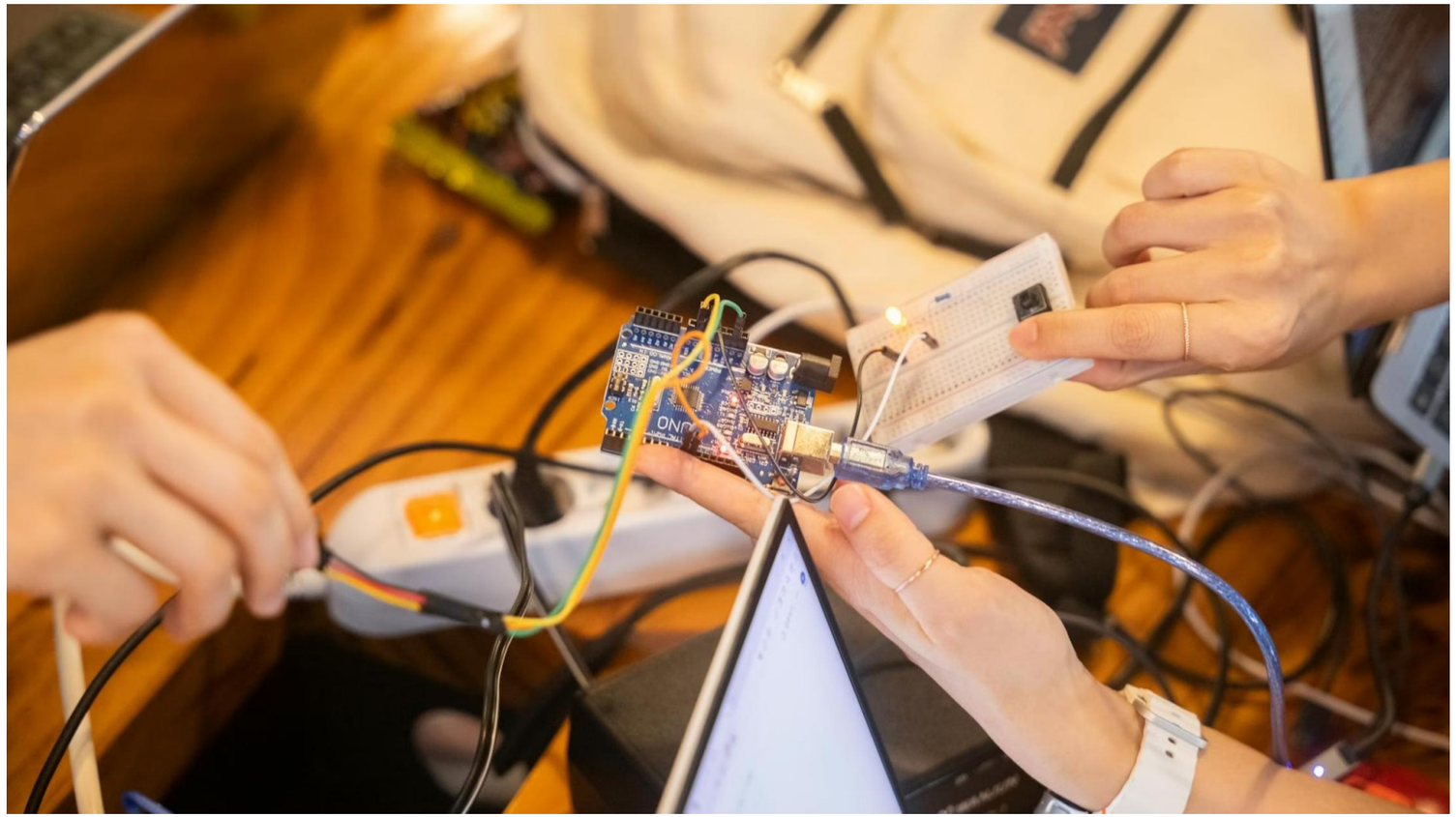

ON DIP
1 2 3 4
ON
1 2



Conclusion

Key Takeaways

This project successfully demonstrated the implementation of a DIP Switch Based Input & Output Control System. We gained practical experience with digital input/output interfacing, the crucial role of pull-up/pull-down resistors, and hardware-based logic using Arduino. The modular and adaptable design showcases the versatility of DIP switches for various embedded applications.



Project Summary: This DIP switch based system effectively demonstrates fundamental concepts of digital electronics and microcontroller interfacing through practical, hands-on implementation, providing a solid foundation for future embedded projects.