**Buzzer with an ESP8266 microcontroller**

**Components Needed**

1. **ESP8266 Module** (e.g., NodeMCU, Wemos D1 Mini)
2. **Buzzer** (active or passive)
3. **Breadboard and Jumper Wires**
4. **Power Supply** (USB or battery)

**Wiring**

1. **Connect the Buzzer:**
   * For an **active buzzer**, connect one pin to a digital output pin on the ESP8266 (e.g., D1) and the other pin to GND.
   * For a **passive buzzer**, do the same, but you'll need to generate a PWM signal to create sound.

**Code:**

const int buzzerPin = D1; // Change to your buzzer pin

void setup() {

pinMode(buzzerPin, OUTPUT);

}

void loop() {

// Turn the buzzer on

digitalWrite(buzzerPin, HIGH);

delay(500); // Beep for 500 milliseconds

// Turn the buzzer off

digitalWrite(buzzerPin, LOW);

delay(500); // Pause for 500 milliseconds

**}**

### Using PWM for Passive Buzzers

If you’re using a passive buzzer, you can generate different tones using PWM. Here's a quick example:

const int buzzerPin = D1;

void setup() {

pinMode(buzzerPin, OUTPUT);

}

void loop() {

tone(buzzerPin, 1000); // 1000 Hz

delay(500); // Sound for 500 ms

noTone(buzzerPin); // Stop sound

delay(500); // Pause

}

### Result:

This basic setup should get your buzzer working with the ESP8266. You can adjust the timing and frequency to create different sounds.