National Textile University, Faisalabad

Department of Computer Science

Name:	Fareena Shahbaz
Class:	BSCS 5th _A
Registration No:	23-NTU-CS-1024
Course Name:	Embedded IoT Systems
Submitted To:	Sir Nasir Mahmood
Submission Date:	26/10/2025

Assignment#1

Contents

TASK-A

Simple LED Toggle with OLED Display

Wokwi Project Link:

https://wokwi.com/projects/445497187511394305

Objective

To implement a **basic LED control** using a **push button** with visual feedback on an **OLED display**.

Hardware Components

Component	Description
ESP32	Microcontroller
Push Button	Used to toggle LED
LED	Indicates state (ON/OFF)
OLED (SSD1306)	Displays current LED state

Working Principle

1. Button is connected using **internal pull-up resistor**.
2. When pressed (LOW), it toggles the LED state.
3. OLED updates message to show **"LED ON"** or **"LED OFF"**.

System Behavior

Action	LED State	OLED Message
Initial	OFF	"LED OFF"
1st Press	ON	"LED ON"

2nd Press	OFF	“LED OFF”

Features

- Debounced button press detection.
- Real-time OLED message updates.
- Simple LED toggling logic for beginners.

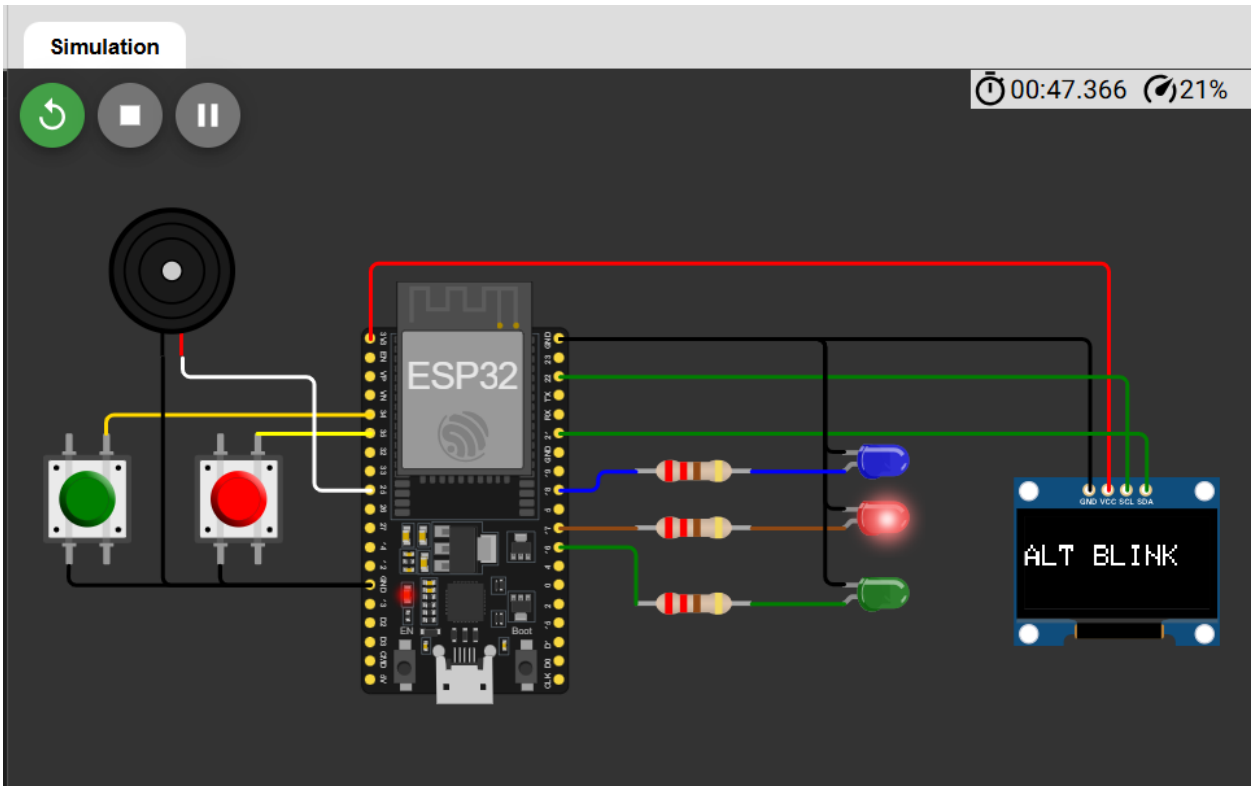
Expected Output

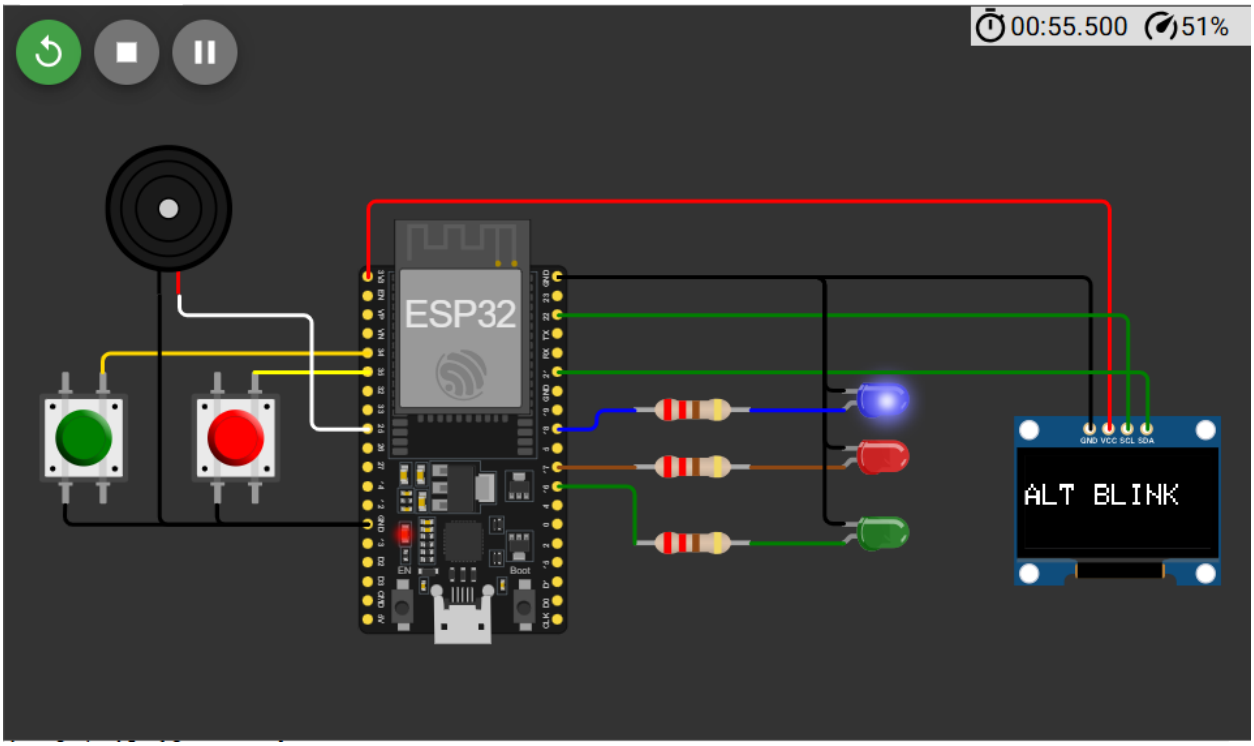
- Pressing the button toggles LED.
- OLED correctly shows **LED ON / LED OFF**.
- Serial monitor logs events like “LED ON” or “LED OFF”.

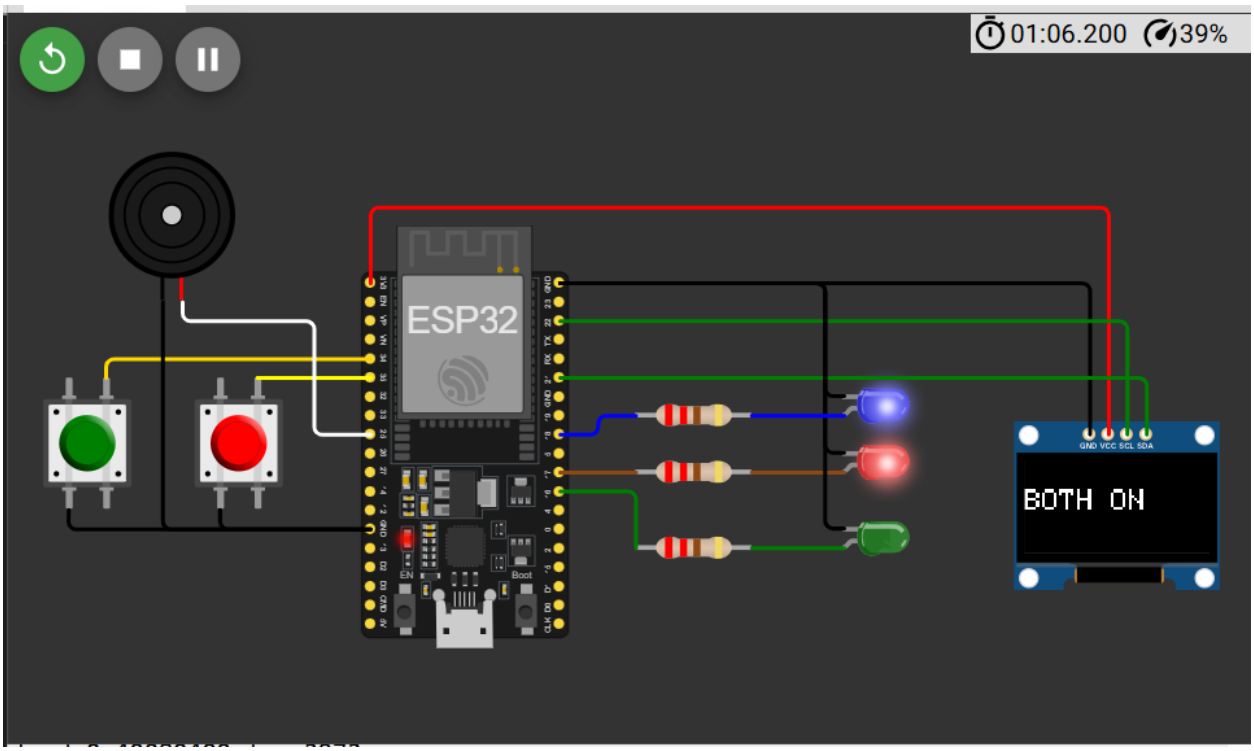
Summary

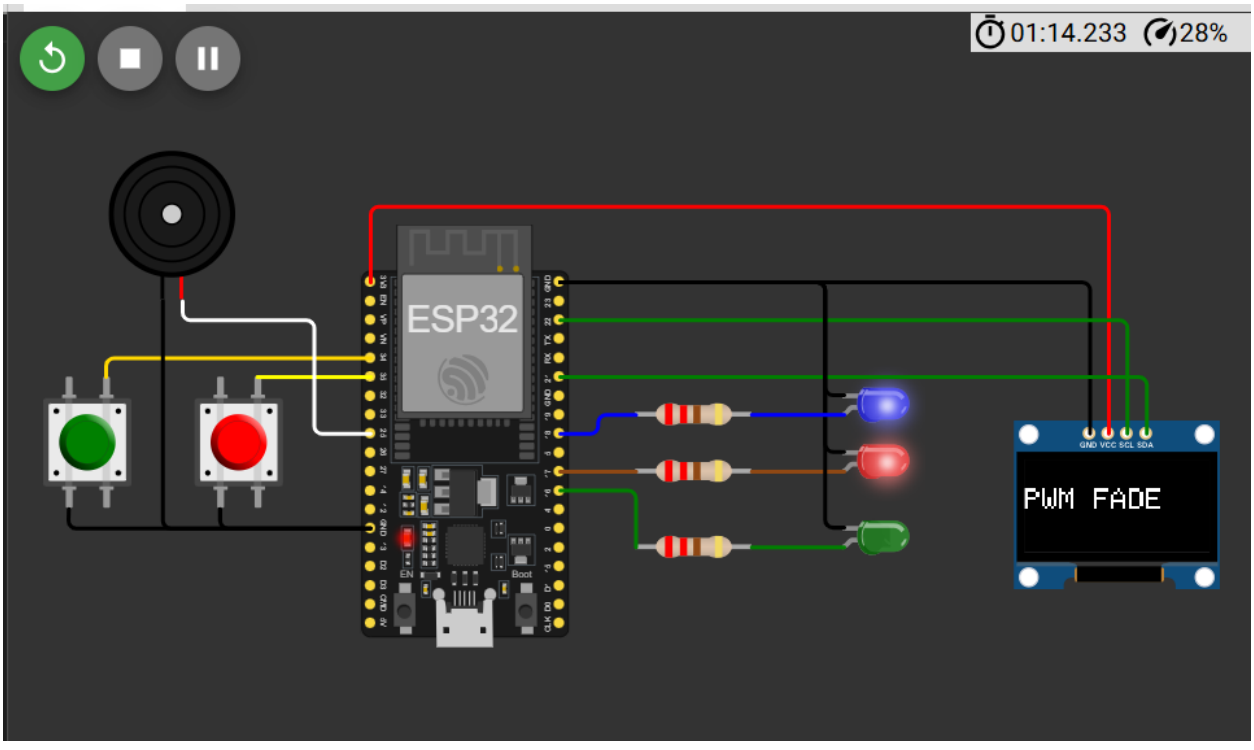
Action	**Output**
Button Press	LED toggles
OLED	Displays LED state
Serial Monitor	Shows debug messages

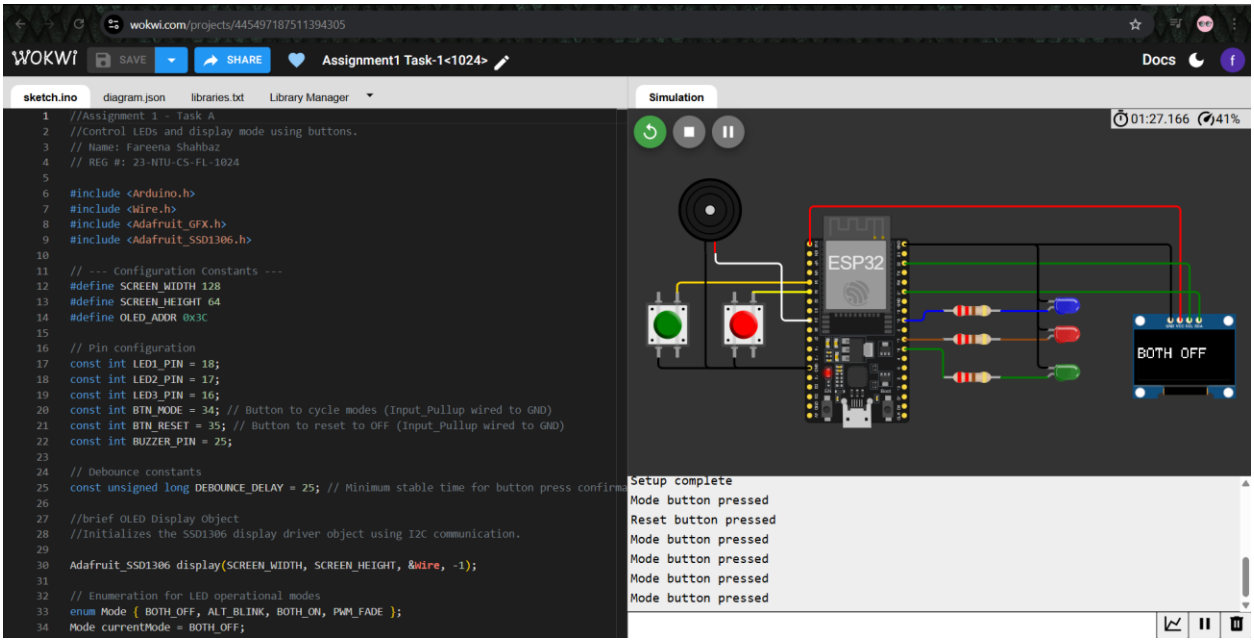
ScreenShots:


Simulation
00:47.366
21%
ESP32
ALT BLINK


00:55.500
51%
ESP32
ALT BLINK


01:06.200
39%
ESP32
BOTH ON


01:14.233
28%
ESP32
PWM FADE




```
//Assignment 1 - Task A
//Control LEDs and display mode using buttons.
// Name: Fareena Shahbaz
// REG #: 23-NTU-CS-FL-1024

#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// --- Configuration Constants ---
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDR 0x3C

// Pin configuration
const int LED1_PIN = 18;
const int LED2_PIN = 17;
const int LED3_PIN = 16;
const int BTN_MODE = 34; // Button to cycle modes (Input_Pullup wired to GND)
const int BTN_RESET = 35; // Button to reset to OFF (Input_Pullup wired to GND)
const int BUZZER_PIN = 25;

// Debounce constants
const unsigned long DEBOUNCE_DELAY = 25; // Minimum stable time for button press confirm

//brief OLED Display Object
//Initializes the SSD1306 display driver object using I2C communication.

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Enumeration for LED operational modes
enum Mode { BOTH_OFF, ALT_BLINK, BOTH_ON, PWM_FADE };
Mode currentMode = BOTH_OFF;
```

TASK-B

Short & Long Press with OLED Feedback

Wokwi Project Link:

https://wokwi.com/projects/445804337034818561

Objective

To implement **button press duration detection** on ESP32.

- A **short press** toggles the LED state (ON/OFF).
- A **long press (more than 1.5 seconds)** plays a **buzzer tone**.
- The **OLED display** updates messages according to the current system state.

Hardware Components

Component	Description
ESP32	Main microcontroller

LED	Visual indicator for short press action
Buzzer	Emits tone for long press
Push Button	Used to trigger actions
OLED (SSD1306)	Displays system messages

Pin Configuration

Component	ESP32 Pin
LED	GPIO 18
Button	GPIO 34
Buzzer	GPIO 25
OLED SDA	GPIO 21
OLED SCL	GPIO 22

Logic Explanation

1. **Button Press Detection:**
    - The button is connected with an **internal pull-up resistor**, meaning it reads **HIGH** when not pressed and **LOW** when pressed.
2. **Debouncing:**
    - To avoid noise in mechanical button presses, a **50 ms delay** (DEBOUNCE_DELAY) is used.
3. **Press Duration Calculation:**
    - When the button is pressed, pressStartTime is recorded using millis().
    - On release, the total duration is checked.
4. **Short Press Action:**
    - If the duration is **less than 1.5 seconds**, the LED toggles its state (ON ↔ OFF).
    - The OLED displays **"LED ON"** or **"LED OFF"** accordingly.

5. **Long Press Action:**
    - If the duration is **more than 1.5 seconds**, the **buzzer plays a tone** at 1000 Hz for 500 ms.
    - The OLED displays **"BUZZER ON"** temporarily.

Functions Description

Function	Purpose
updateOLED(const char *msg)	Clears OLED and displays a new message (like "LED ON" or "BUZZER ON")
playBuzzerTone()	Plays a 1000 Hz tone for 500 ms using tone()
loop()	Detects button press, determines press duration, and executes actions accordingly
setup()	Initializes all hardware components (OLED, button, LED, buzzer)

OLED Display Messages

Event	Display Message
System Start	LED OFF
Short Press (LED ON)	LED ON
Short Press (LED OFF)	LED OFF
Long Press	BUZZER ON

Flow Summary

1. Button Pressed → Time Recorded
2. If released quickly → Toggle LED
3. If held long → Play Buzzer
4. OLED updates messages accordingly

Key Features

- Uses **non-blocking** timing via millis() instead of delay() for press detection.
- Displays **real-time feedback** on OLED.
- Implements **software debouncing**.
- Demonstrates **multi-peripheral coordination** on ESP32.

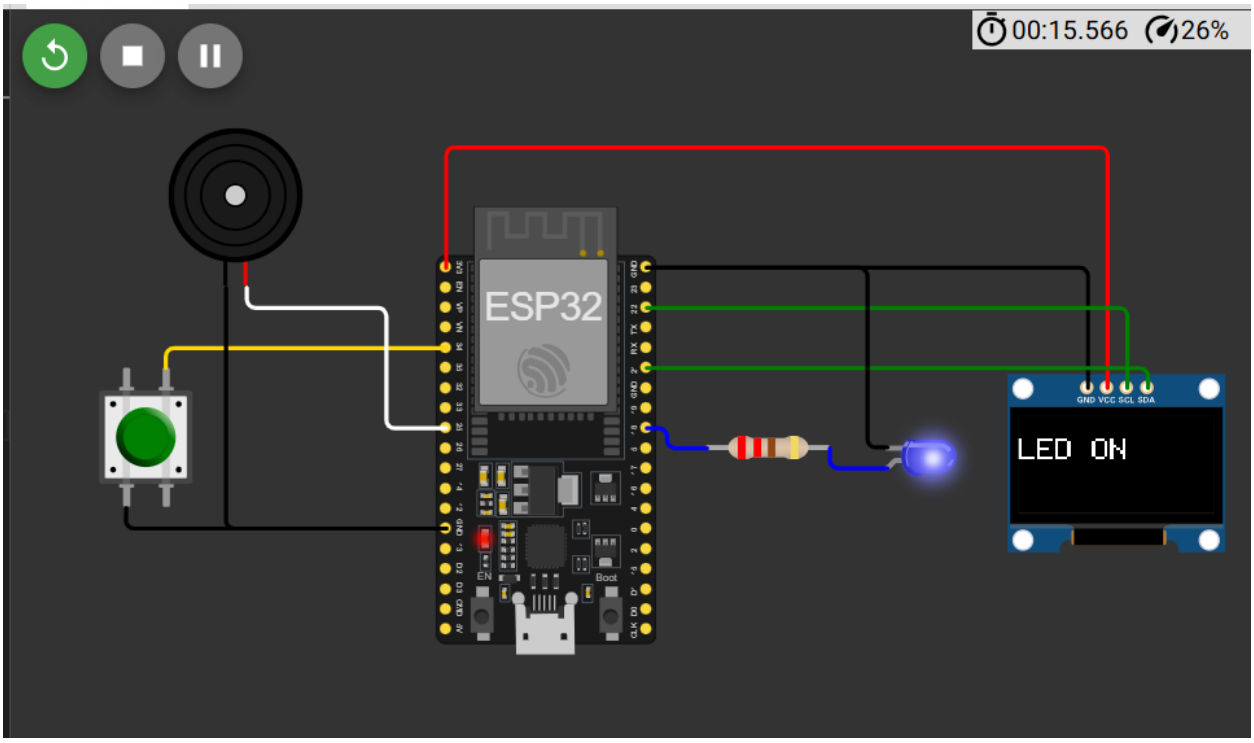
Expected Output

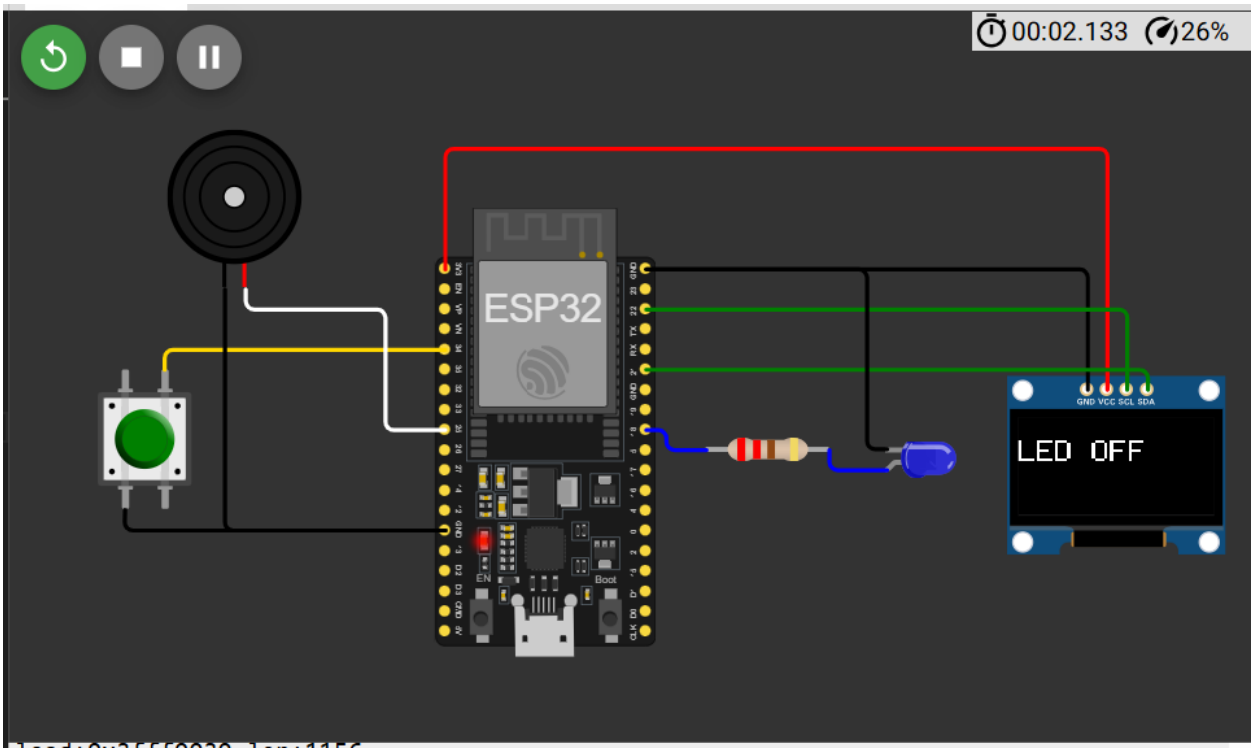
- **Short Press:** LED toggles ON/OFF, OLED shows LED state.
- **Long Press (>1.5s):** Buzzer plays tone, OLED shows “BUZZER ON”.

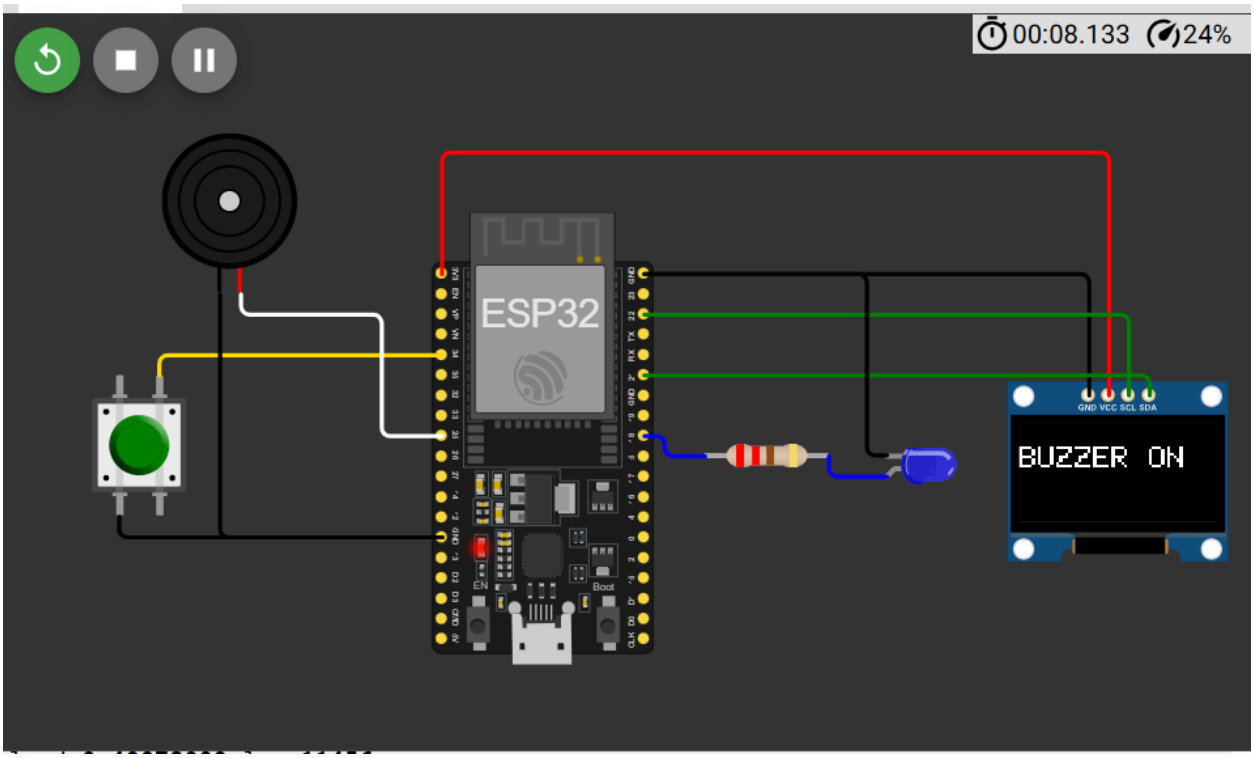
Summary

Action	**Result**
Short Press	LED toggles
Long Press	Buzzer plays
OLED	Shows feedback for every action
Serial Monitor	Logs button events and actions

ScreenShots And Working:






00:08.133
24%
ESP32
GND VCC SCL SDA
BUZZER ON