**National Textile University, Faisalabad**



**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Hiba Fatima |
| **Class:** | BSCS\_A 5th Semester |
| **Registration No:** | 23-NTU-CS-1037 |
| **Assignment :** | Project |
| **Course Name:** | Embedded IOT |
| **Submitted To:** | *Sir Nasir* |
| **Submission Date:** | 26-10-2025 |

**Project Documentation: Multi-Task ESP32 Control System**

**1. Project Overview**

This project implements two distinct tasks—LED mode cycling (Task A) and press-type detection (Task B)—on a single ESP32 Dev Module. The system is controlled by two pushbuttons and provides visual feedback via an OLED display and audible feedback via a buzzer.

| **Component** | **Quantity** | **Purpose** |
| --- | --- | --- |
| ESP32 Dev Module | 1 | Microcontroller running the logic. |
| OLED Display (SSD1306, I²C) | 1 | Displays the current mode and button event. |
| Push Button | 2 | Inputs: one for mode cycle/reset, one for press detection. |
| LED (Red, Green, Blue) | 3 | Visual outputs for mode sequences and toggling. |
| Buzzer | 1 | Audible output for the long-press event. |

**TASK A:**

Circuit Diagram: Design a Wokwi circuit and draw a neat hand-sketch including:

• 2 push buttons

• 3 LEDs

• 1 buzzer

• 1 OLED

Task A — Coding: Use one button to cycle through LED modes (display the current state on

the OLED):

1. Both OFF

2. Alternate blink

3. Both ON

4. PWM fade

Use the second button to reset to OFF.

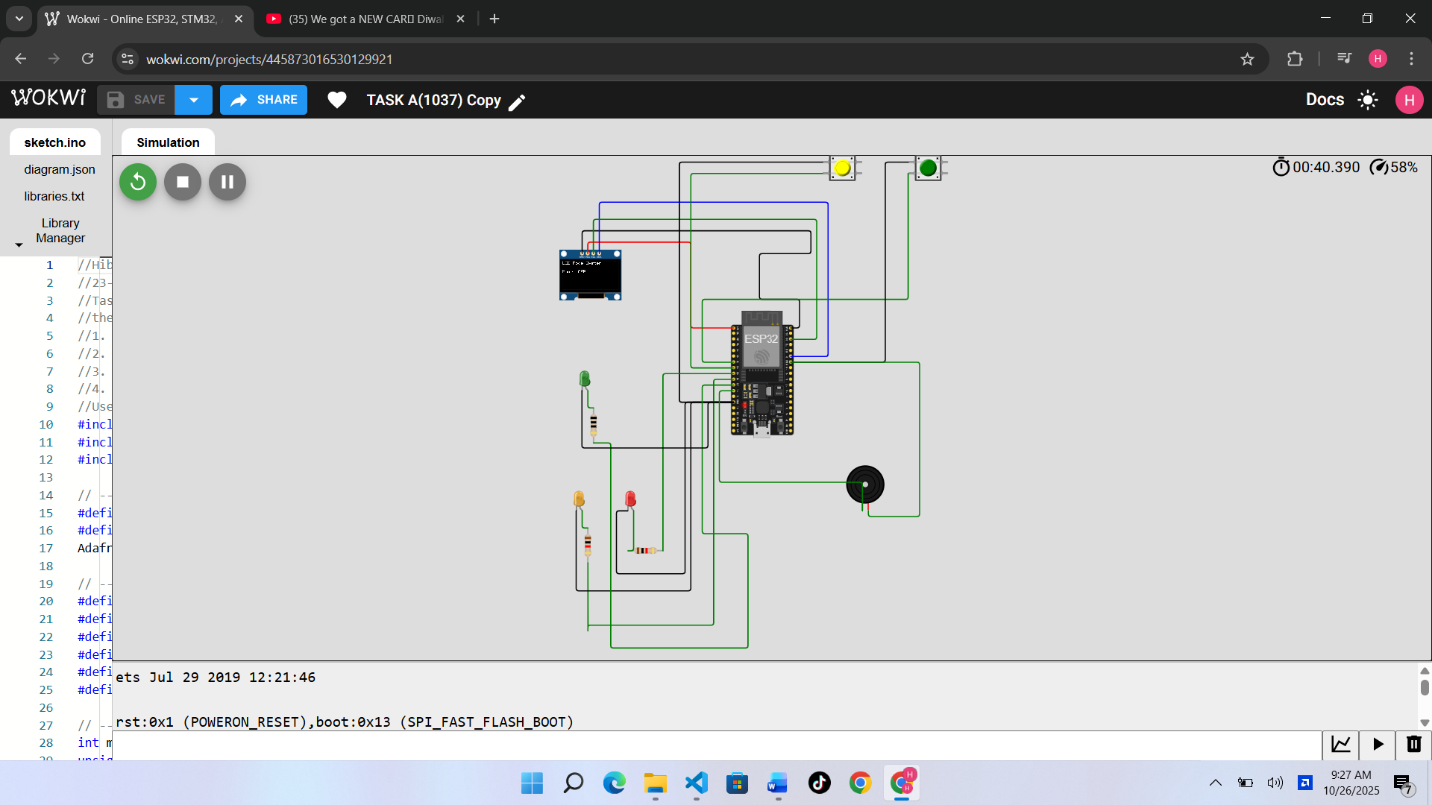
**TASK A :FIXED BUTTON ISSUE**

**WOKWI LINK :**

[**https://wokwi.com/projects/445873016530129921**](https://wokwi.com/projects/445873016530129921)

**SCREENSHOT:**

Both OFF:

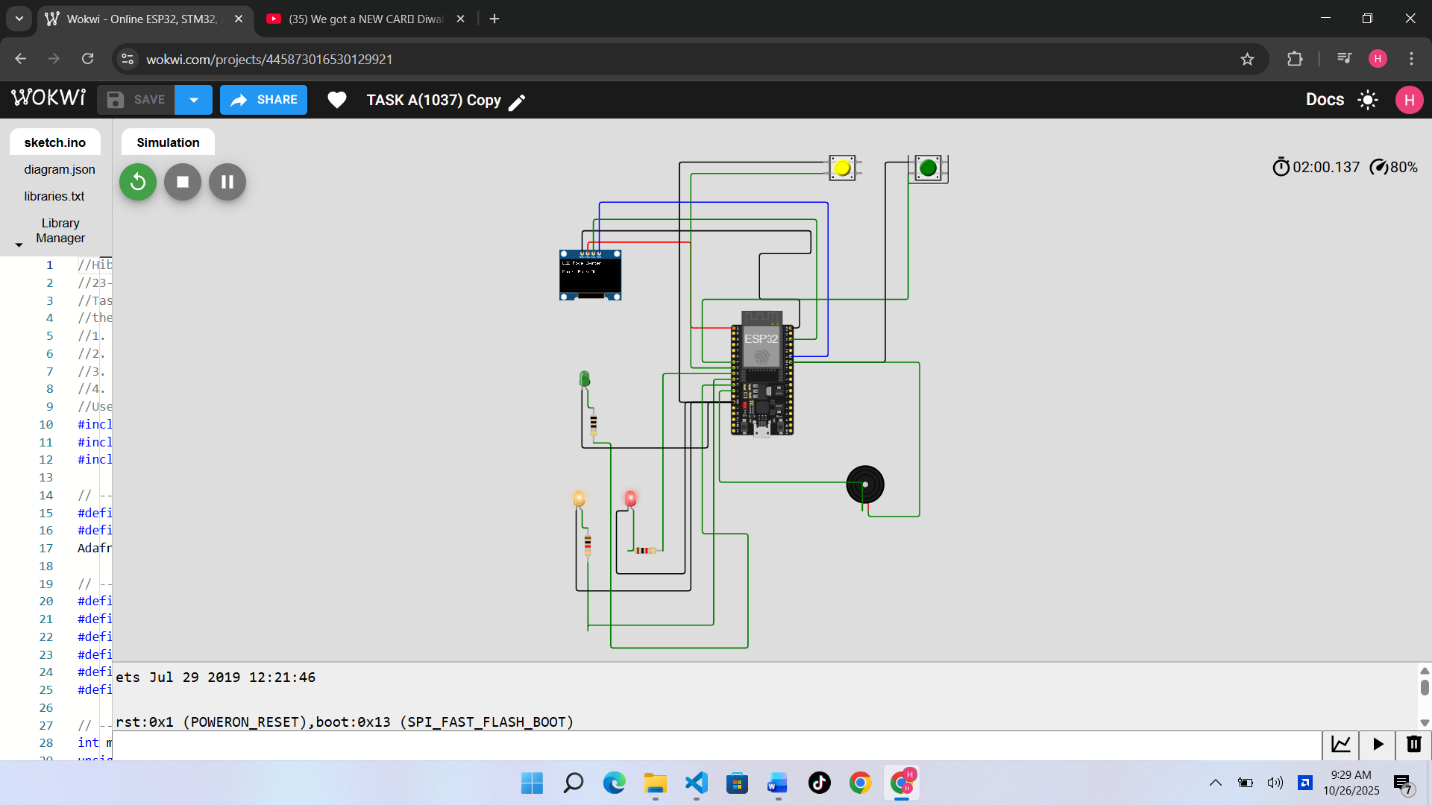


Alternate blink:

A computer screen shot of a computer

AI-generated content may be incorrect.

Both ON:

PWM fade:

A computer screen shot of a computer

AI-generated content may be incorrect.

**Loom Link Video Link:**

<https://www.loom.com/share/f5a0f906af794c0ab559880db1026b48>

**Code Of The Program:**

//Hiba Fatima

//23-Ntu-cs-1037

//Task A — Coding: Use one button to cycle through LED modes (display the current state on

//the OLED):

//1. Both OFF

//2. Alternate blink

//3. Both ON

//4. PWM fade

//Use the second button to reset to OFF.

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

// -------- OLED CONFIG ----------

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// -------- PIN DEFINITIONS ----------

#define LED1 25        // Red

#define LED2 26        // Yellow

#define LED3 27        // Optional Green

#define BUTTON\_MODE 32 // Mode button

#define BUTTON\_RESET 33 // Reset button

#define BUZZER 14

// -------- GLOBAL VARIABLES ----------

int mode = 0; // 0: OFF, 1: Alternate Blink, 2: Both ON, 3: PWM Fade

unsigned long lastPress = 0;

int fadeValue = 0;

int fadeDir = 1;

// -------- FUNCTION DECLARATIONS ----------

void showMode();

void setup() {

  pinMode(LED1, OUTPUT);

  pinMode(LED2, OUTPUT);

  pinMode(LED3, OUTPUT);

  pinMode(BUZZER, OUTPUT);

  pinMode(BUTTON\_MODE, INPUT\_PULLUP);

  pinMode(BUTTON\_RESET, INPUT\_PULLUP);

  Wire.begin();

  display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);

  display.clearDisplay();

  display.setTextColor(SSD1306\_WHITE);

  display.setTextSize(1);

  display.setCursor(0, 0);

  display.println("System Ready...");

  display.display();

  delay(1000);

  showMode();

}

void loop() {

  // ----- MODE BUTTON -----

  if (digitalRead(BUTTON\_MODE) == LOW) {

    delay(200); // debounce

    mode = (mode + 1) % 4;

    showMode();

    while (digitalRead(BUTTON\_MODE) == LOW); // wait for release

  }

  // ----- RESET BUTTON -----

  if (digitalRead(BUTTON\_RESET) == LOW) {

    delay(200);

    mode = 0;

    showMode();

    while (digitalRead(BUTTON\_RESET) == LOW);

  }

  // ----- LED BEHAVIOR -----

  switch (mode) {

    case 0: // OFF

      digitalWrite(LED1, LOW);

      digitalWrite(LED2, LOW);

      break;

    case 1: // Alternate blink

      digitalWrite(LED1, HIGH);

      digitalWrite(LED2, LOW);

      delay(300);

      digitalWrite(LED1, LOW);

      digitalWrite(LED2, HIGH);

      delay(300);

      break;

    case 2: // Both ON

      digitalWrite(LED1, HIGH);

      digitalWrite(LED2, HIGH);

      break;

    case 3: // PWM fade on LED1

      analogWrite(LED1, fadeValue);

      analogWrite(LED2, fadeValue);

      fadeValue += fadeDir \* 10;

      if (fadeValue >= 255 || fadeValue <= 0) fadeDir = -fadeDir;

      delay(30);

      break;

  }

}

// -------- DISPLAY FUNCTION ----------

void showMode() {

  display.clearDisplay();

  display.setCursor(0, 0);

  display.println("LED Mode System");

  display.setCursor(0, 20);

  switch (mode) {

    case 0: display.println("Mode: OFF"); break;

    case 1: display.println("Mode: Alternate Blink"); break;

    case 2: display.println("Mode: Both ON"); break;

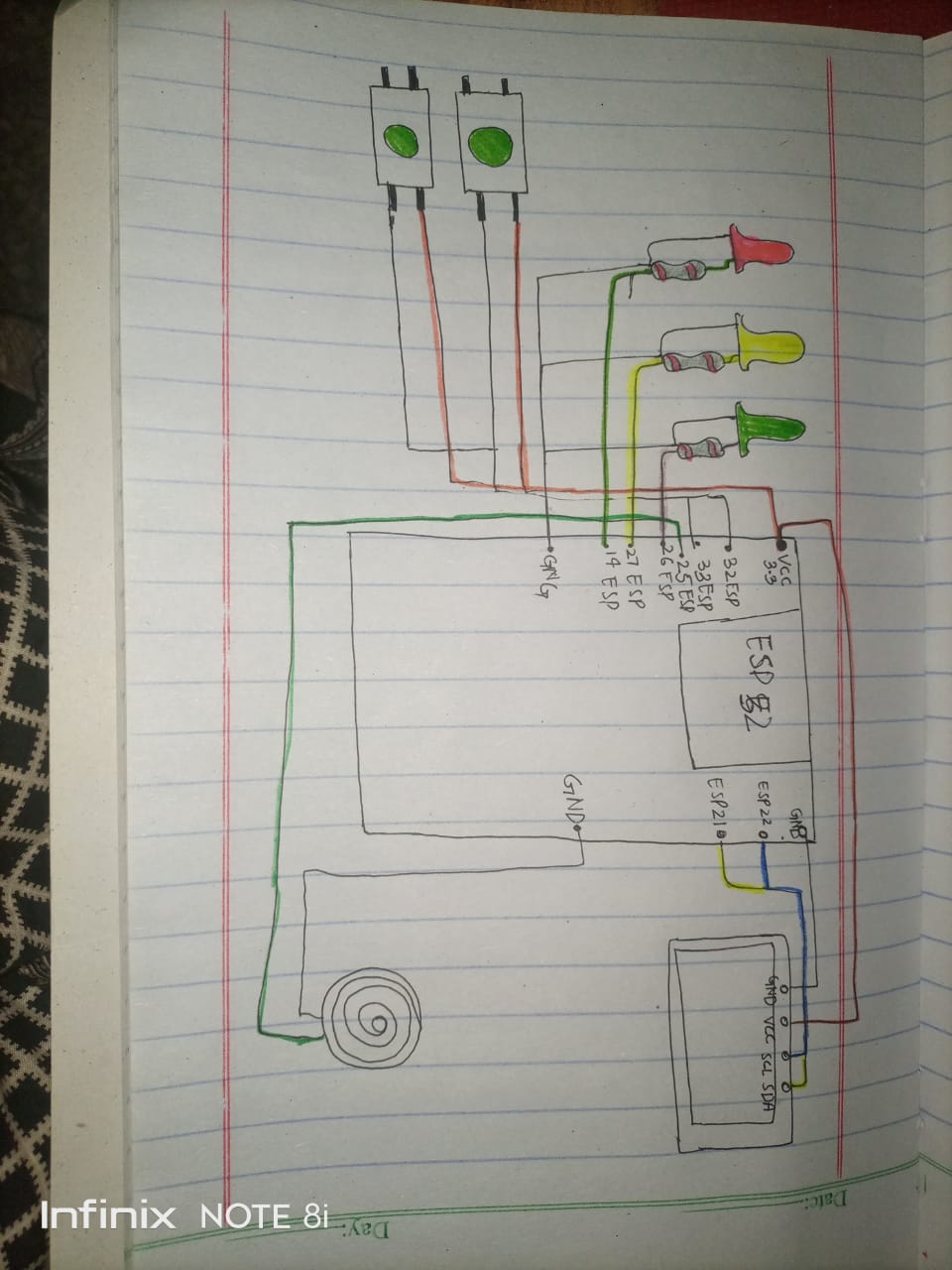
    case 3: display.println("Mode: PWM Fade"); break;

  }

  display.display();

}

**HANDMADE SKETCH:**



**TASK B:**

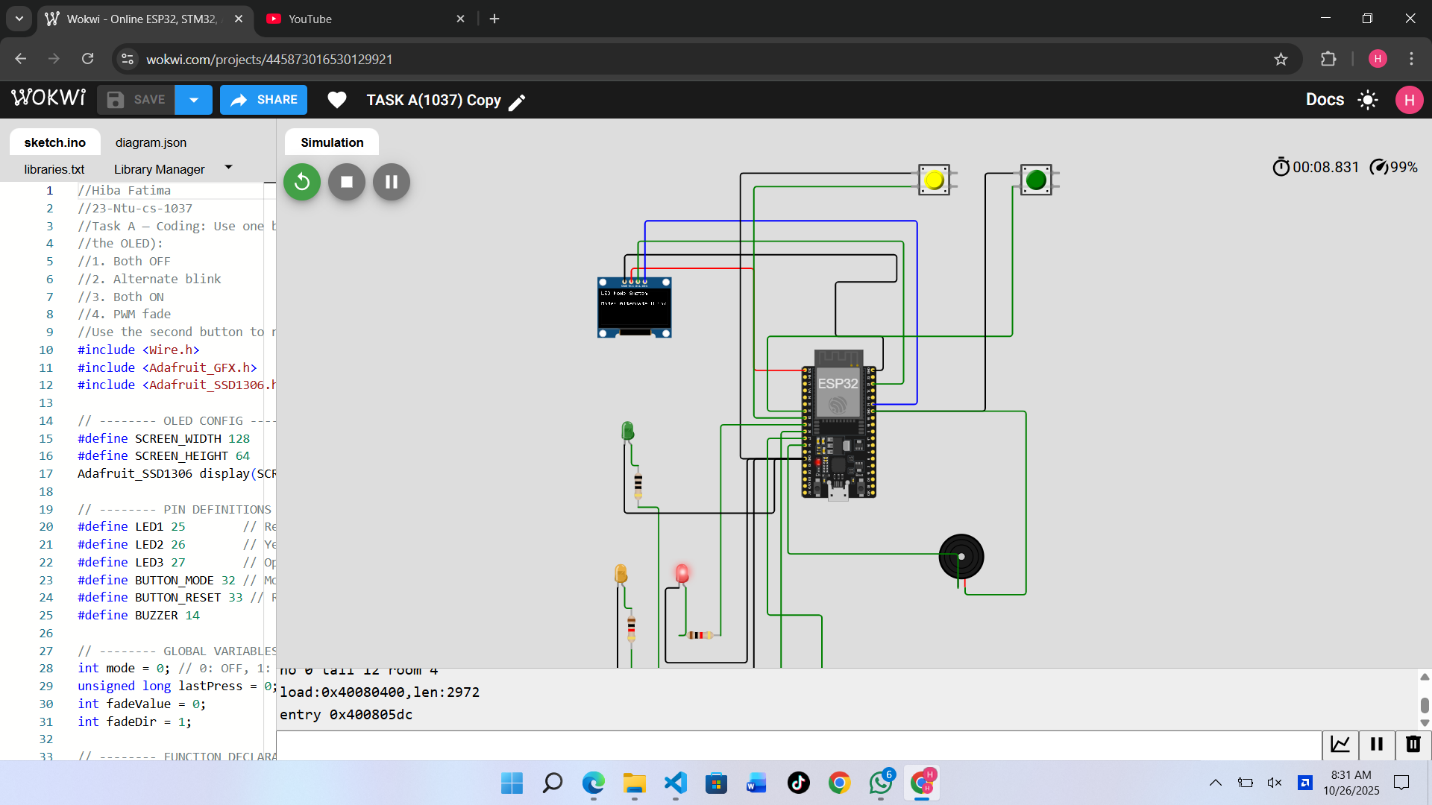
— Coding: Use a single button with press-type detection (display the event on the OLED): • Short press → toggle LED • Long press (> 1.5 s) → play a buzzer tone.

**WOKWI LINK:**

https://wokwi.com/projects/445873593457762305

**WORKING SCREENSHOT:**

**Buzzer running:**



**Loom video link:**

https://www.loom.com/share/28a3c60e49ba4aefb3ebd774c85f3a5d

**Code of the program:**

//Hiba Fatima

//23-Ntu-cs-1037

//Task B — Coding: Use a single button with press-type detection (display the event on the

//OLED):

// Short press → toggle LED

//Long press (> 1.5 s) → play a buzzer tone

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

// ---------- OLED CONFIG ----------

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

// ---------- PIN DEFINITIONS ----------

#define BUTTON 32

#define LED 25

#define BUZZER 14

// ---------- GLOBAL VARIABLES ----------

bool ledState = false;

unsigned long pressStart = 0;

bool isPressed = false;

// ---------- SETUP ----------

void setup() {

  pinMode(BUTTON, INPUT\_PULLUP);   // button connected to GND

  pinMode(LED, OUTPUT);

  pinMode(BUZZER, OUTPUT);

  Wire.begin();

  display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);

  display.clearDisplay();

  display.setTextColor(SSD1306\_WHITE);

  display.setTextSize(1);

  display.setCursor(0, 0);

  display.println("System Ready...");

  display.display();

  delay(1000);

  showMessage("Waiting for button...");

}

// ---------- MAIN LOOP ----------

void loop() {

  int buttonState = digitalRead(BUTTON);

  // detect press

  if (buttonState == LOW && !isPressed) {

    isPressed = true;

    pressStart = millis(); // time press began

  }

  // detect release

  if (buttonState == HIGH && isPressed) {

    isPressed = false;

    unsigned long pressDuration = millis() - pressStart;

    if (pressDuration > 1500) {

      // Long press -> play buzzer melody

      playMelody();

      showMessage("Long Press -> Melody");

    } else {

      // Short press -> toggle LED + short beep

      ledState = !ledState;

      digitalWrite(LED, ledState);

      shortBeep();

      showMessage("Short Press -> LED " + String(ledState ? "ON" : "OFF"));

    }

  }

}

// ---------- FUNCTIONS ----------

void playMelody() {

  // simple 3-tone melody for long press

  tone(BUZZER, 500, 200);   // low tone

  delay(250);

  tone(BUZZER, 800, 200);   // medium tone

  delay(250);

  tone(BUZZER, 1200, 300);  // high tone

  delay(350);

  noTone(BUZZER);

}

void shortBeep() {

  // quick beep for short press

  tone(BUZZER, 1000, 100);

  delay(150);

  noTone(BUZZER);

}

void showMessage(String msg) {

  display.clearDisplay();

  display.setCursor(0, 0);

  display.println("Button Event:");

  display.setCursor(0, 20);

  display.println(msg);

  display.display();

}

**Handmade sketch:**

A drawing of a wiring diagram

AI-generated content may be incorrect.

**GITHUB LINK:**

[**https://github.com/HIBA-616/Embedded-iot-1037-.git**](https://github.com/HIBA-616/Embedded-iot-1037-.git)