**National Textile University, Faisalabad**



**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Amna Khalid |
| **Class:** | BSCS\_A 5th Semester |
| **Registration No:** | 23-NTU-CS-1016 |
| **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/445872855716310017**](https://wokwi.com/projects/445872855716310017)

**SCREENSHOT:**

Both OFF:

A computer screen shot of a computer

AI-generated content may be incorrect.

Alternate blink:

A screenshot of a computer

AI-generated content may be incorrect.

Both ON:

A computer screen shot of a computer

AI-generated content may be incorrect.

PWM fade:

A computer screen shot of a computer

AI-generated content may be incorrect.

**Loom Link Video Link:**

<https://www.loom.com/share/408e4a6b82844adfab2b7d0995c2d7a7>

**Code Of The Program:**

//name:Amna khalid

//reg no:23-NTU-CS-1016

//TASK 1:1. 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>

#define LED1 25

#define LED2 27

#define LED3 26

#define BTN\_MODE 32

#define BTN\_RESET 33

Adafruit\_SSD1306 display(128, 64, &**Wire**, -1);

int mode = 1;

void showMode() {

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(WHITE);

  display.setCursor(0, 10);

  display.print("Mode: ");

  display.print(mode);

  if (mode == 1) display.print(" OFF");

  if (mode == 2) display.print(" ALT");

  if (mode == 3) display.print(" ON");

  if (mode == 4) display.print(" FADE");

  display.display();

}

void setup() {

  pinMode(LED1, OUTPUT);

  pinMode(LED2, OUTPUT);

  pinMode(LED3, OUTPUT);

  pinMode(BTN\_MODE, INPUT);

  pinMode(BTN\_RESET, INPUT);

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

  showMode();

}

void loop() {

  if (digitalRead(BTN\_RESET) == HIGH) {

    mode = 1;

    showMode();

  }

  if (digitalRead(BTN\_MODE) == HIGH) {

    mode++;

    if (mode > 4) mode = 1;

    showMode();

    delay(300);

  }

//

  if (mode == 1) {

    digitalWrite(LED1, LOW);

    digitalWrite(LED2, LOW);

    digitalWrite(LED3, LOW);

  }

  if (mode == 2) {

    digitalWrite(LED1, HIGH);

    digitalWrite(LED2, LOW);

    delay(300);

    digitalWrite(LED1, LOW);

    digitalWrite(LED2, HIGH);

    delay(300);

  }

  if (mode == 3) {

    digitalWrite(LED1, HIGH);

    digitalWrite(LED2, HIGH);

    digitalWrite(LED3, HIGH);

  }

  if (mode == 4) {

    for (int i = 0; i < 255; i++) {

      analogWrite(LED3, i);

      delay(5);

    }

    for (int i = 255; i > 0; i--) {

      analogWrite(LED3, i);

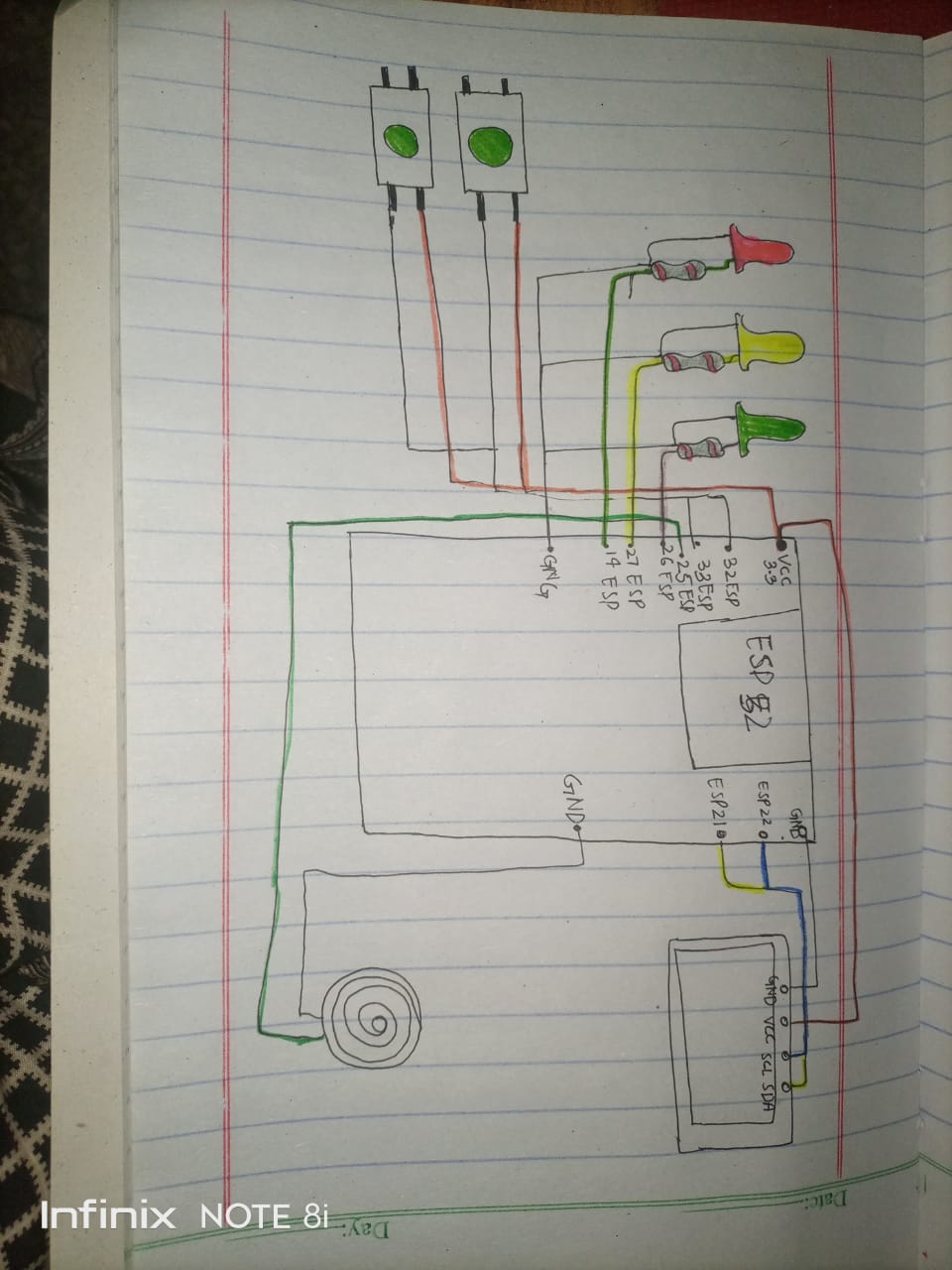
      delay(5);

    }

  }

}

**HANDMADE SKETCH:**



**TASK B:**

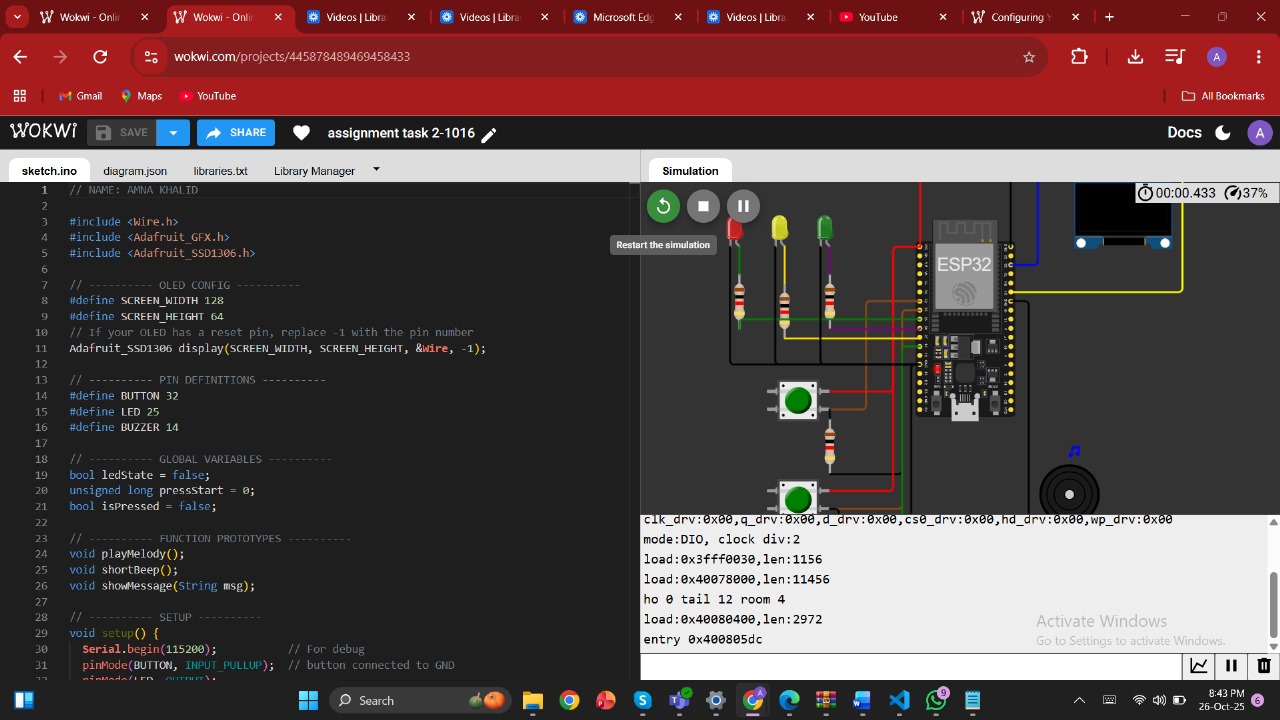
**— Coding: Use a single button with press-type detection (display the event on the OLED): • Short press → toggle LED • Long press (> 1.5 s) → play a buzzer tone.**

**WOKWI LINK:**

<https://wokwi.com/projects/445878489469458433>

**WORKING SCREENSHOT:**

**Buzzer running:**



**Loom video link:**

[**https://www.loom.com/share/69575515d16d42e2ae0780c8dac4d495**](https://www.loom.com/share/69575515d16d42e2ae0780c8dac4d495)

**Code of the program:**

// NAME: AMNA KHALID

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

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

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

// If your OLED has a reset pin, replace -1 with the pin number

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;

// ---------- FUNCTION PROTOTYPES ----------

void playMelody();

void shortBeep();

void showMessage(String msg);

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

void setup() {

**Serial**.begin(115200);           // For debug

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

  pinMode(LED, OUTPUT);

  pinMode(BUZZER, OUTPUT);

**Wire**.begin();

  // Initialize OLED

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

**Serial**.println("SSD1306 allocation failed");

    for (;;); // stop here if OLED fails

  }

  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 button press

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

    isPressed = true;

    pressStart = millis(); // time press began

  }

  // Detect button release

  if (buttonState == HIGH && isPressed) {

    isPressed = false;

    unsigned long pressDuration = millis() - pressStart;

    if (pressDuration > 1500) {

      // Long press -> play 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/Amnaworkspace/embedded\_iOT\_1016\_Amna.git**](https://github.com/Amnaworkspace/embedded_iOT_1016_Amna.git)