**National Textile University, Faisalabad**



**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Ayesha Iftikhar |
| **Class:** | CS 5th (A) |
| **Registration No:** | 23-NTU-CS-1021 |
| **Assignment:** | 1 |
| **Course Name:** | Embedded IOT |
| **Submitted To:** | *Sir Nasir* |
| **Submission Date:** | 26/10/2025 |

***Title:***

“ESP32 Multi-Mode LED and Buzzer Control with OLED Display”

***Project Overview:***

This project shows how the multiple LEDs and Buzzer control with push buttons on ESP32 and their results shown on OLED display screen.

In **Task A** one button handles the different modes of LEDs e.g ON , OFF , Alternative Blink and PWM fade while the second button resets the whole system means all OFF.

In **Task B** one button detects the short and long press , the short press toggles the LED while the long press activates the buzzer tone.

***Tool Used:***

**Wokwi** used to test ESP32 virtually

***Question 3 — Implementation***

**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

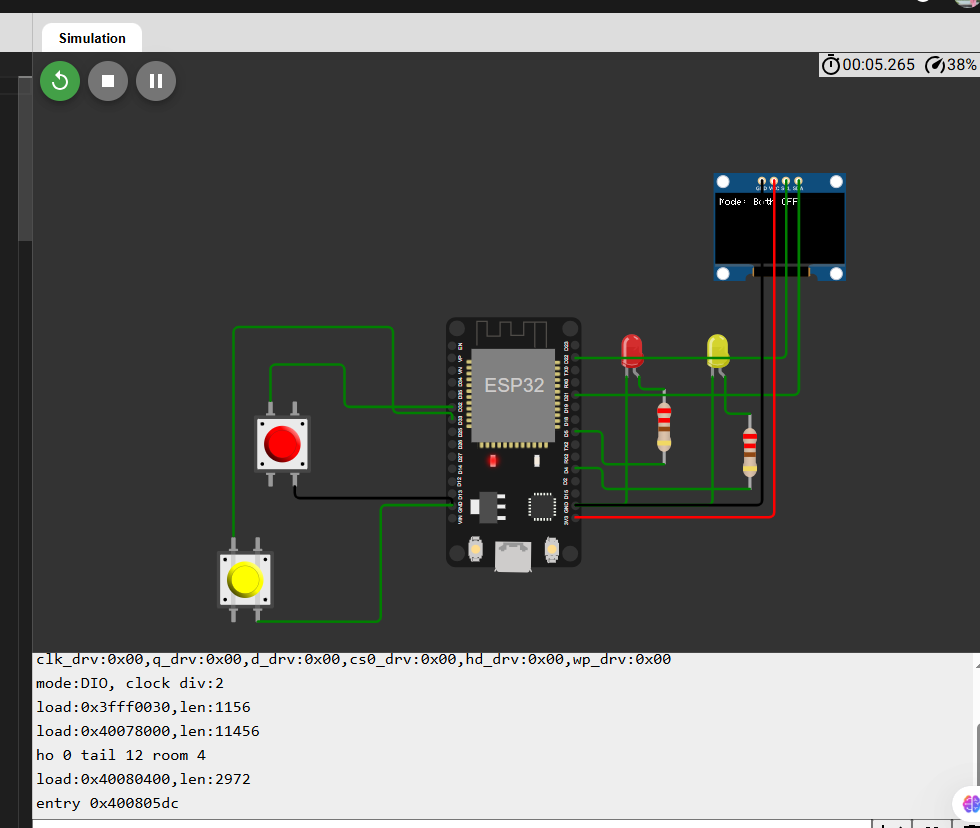
2. Alternate blink

3. Both ON

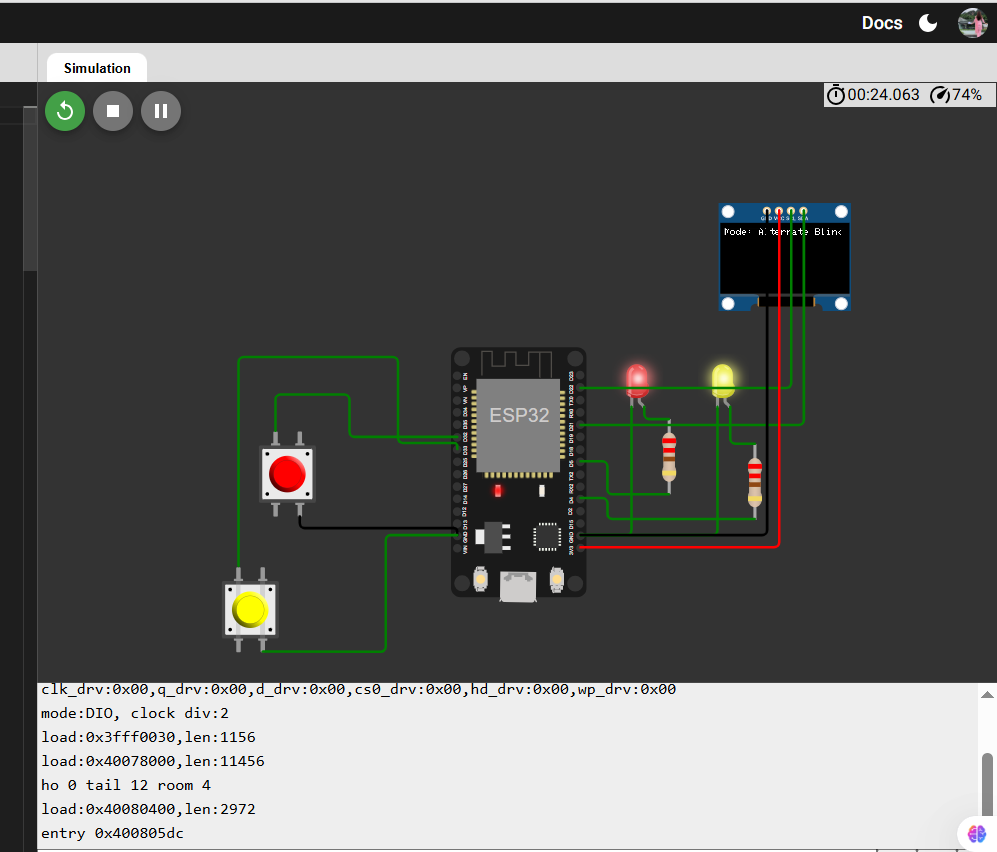
4. PWM fade

Use the second button to reset to OFF.

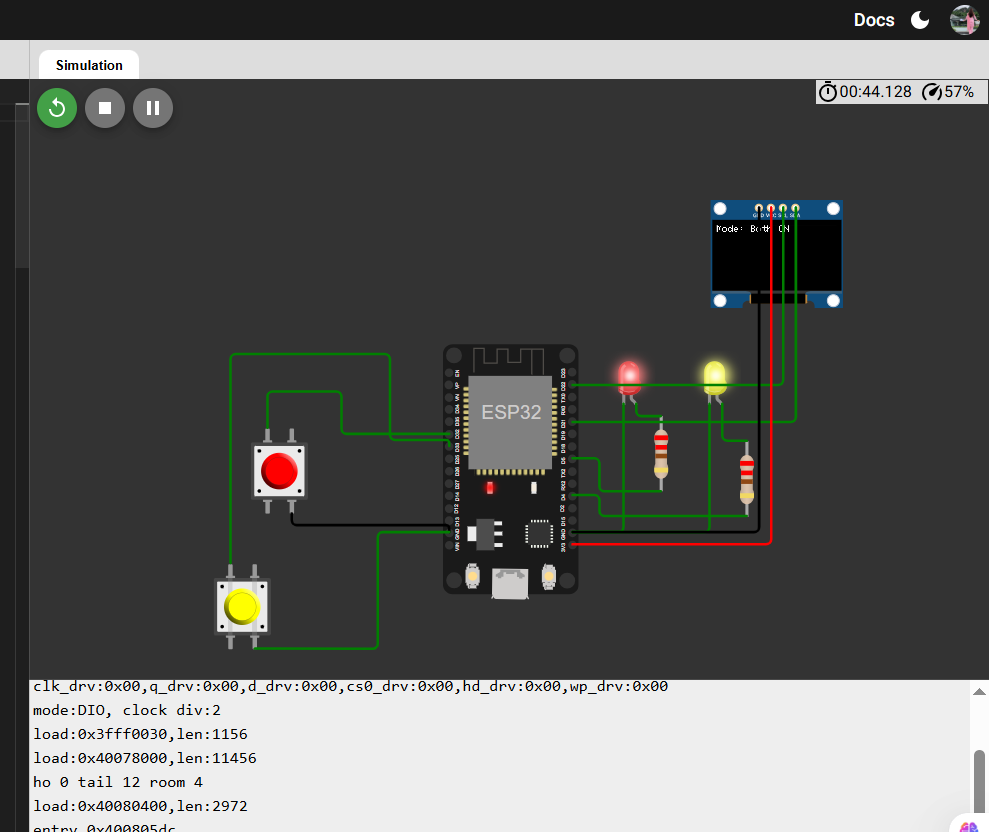
***Both OFF:***



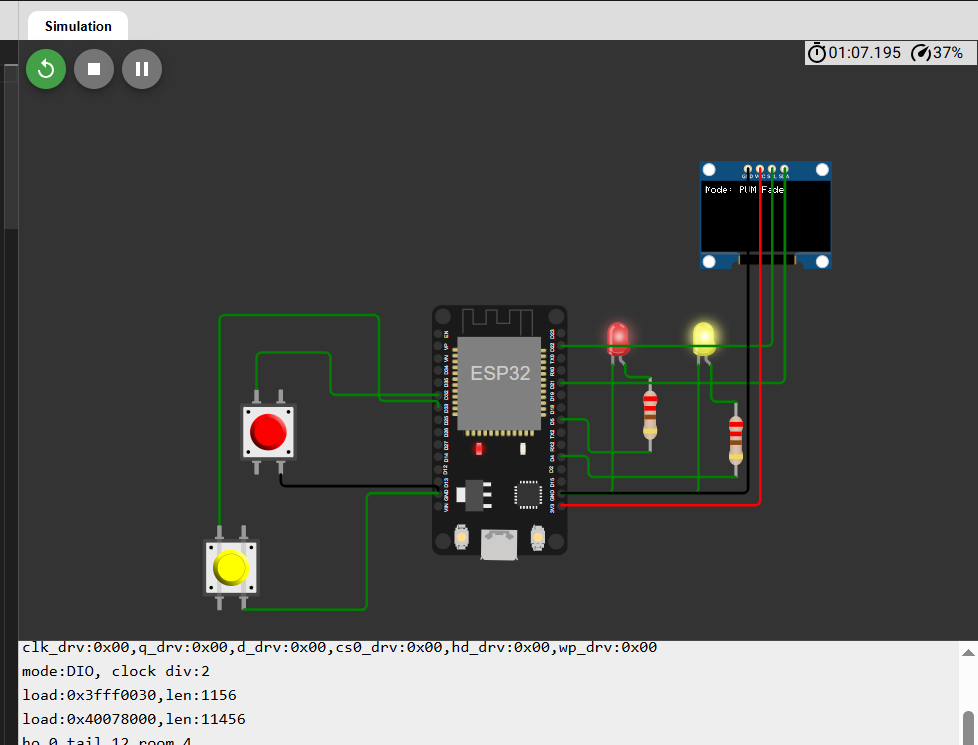
***2. Alternate Blink:***



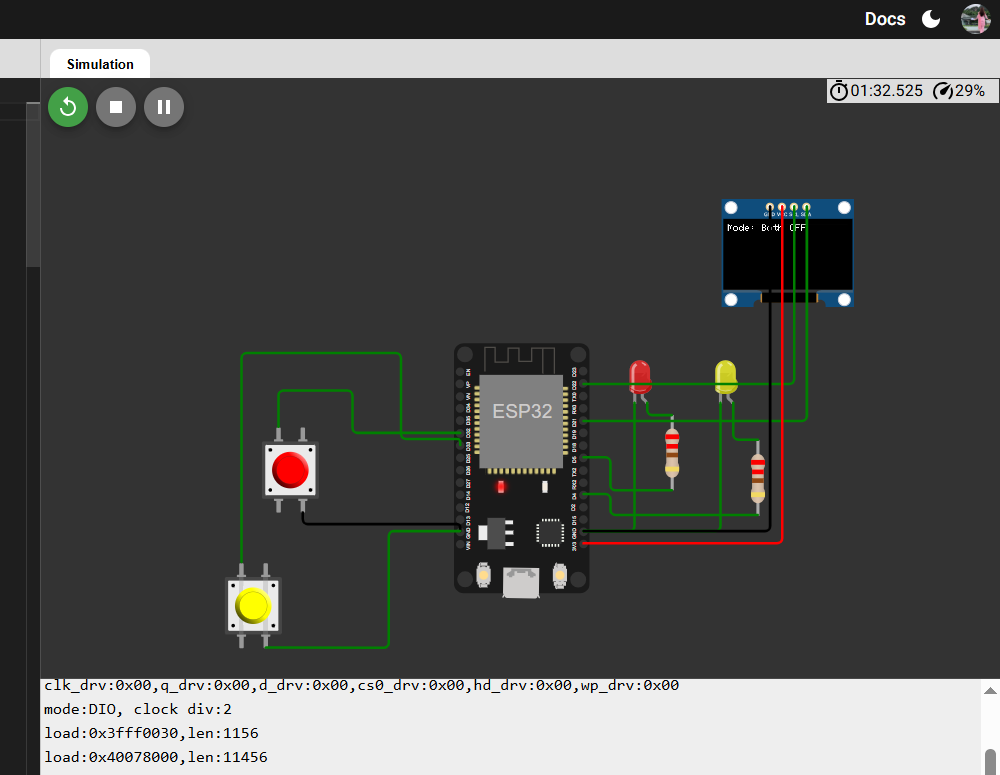
***3. Both ON***



***4. PWM fade:***



***Use the second button to reset to OFF***



***Short Clip:***



<https://www.loom.com/share/008458f6f49240bd966f4213a2bd1a88>

***Wokwi Link:***

<https://wokwi.com/projects/445722481831695361>

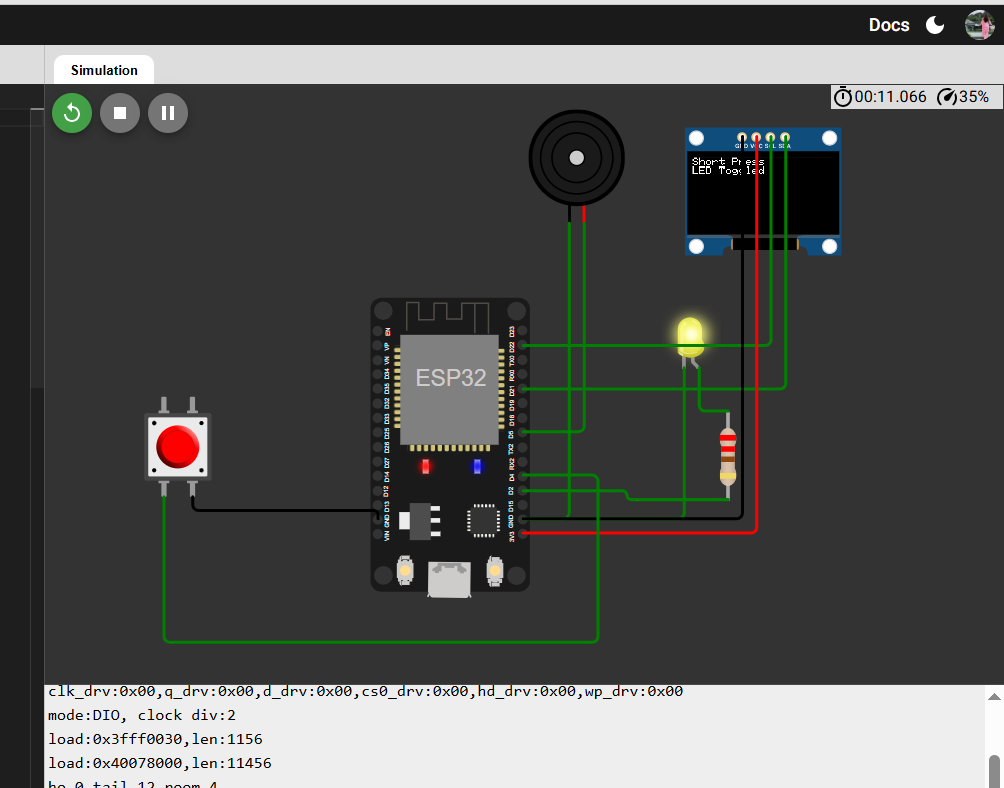
***Task B:***

Use a single button with press-type detection (display the event on the OLED)

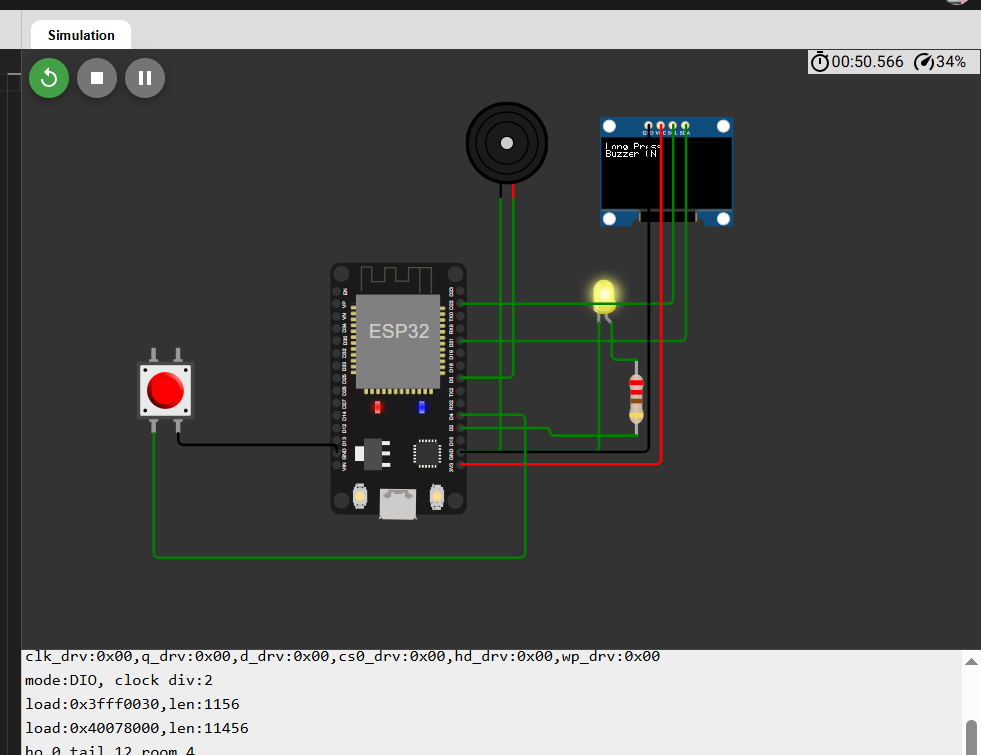
• Short press → toggle LED

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

***Short Press:***



***Long Press:***



***Short Clip:***

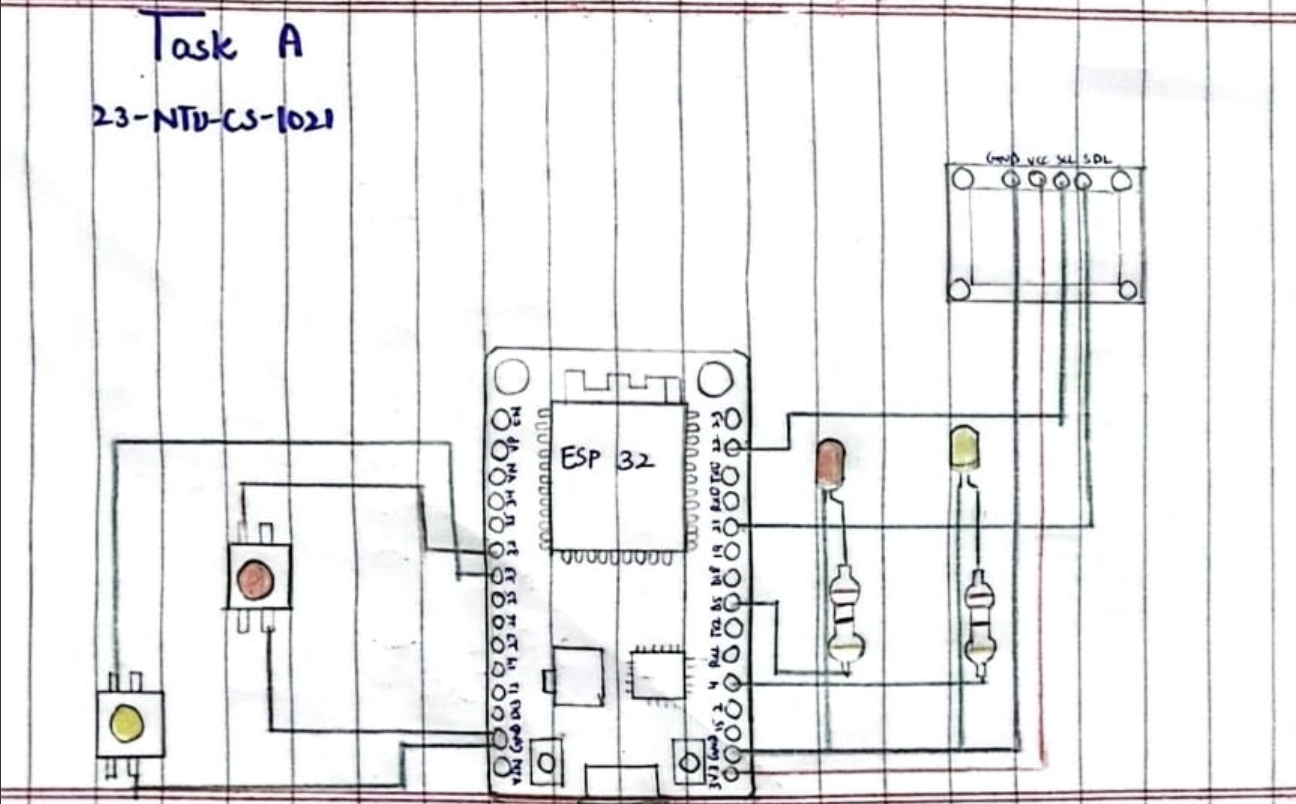
******

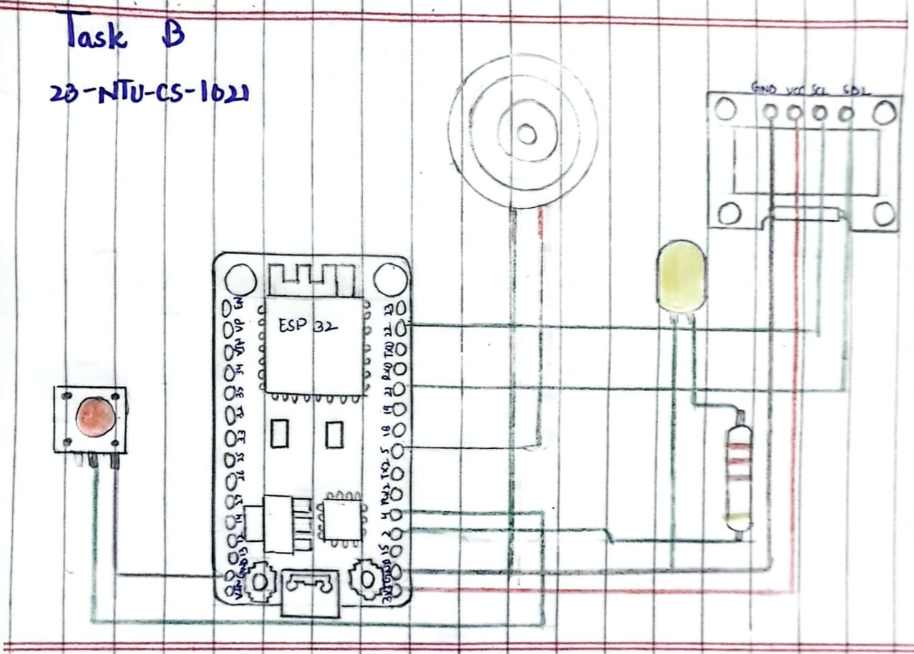
[**https://www.loom.com/share/4e66fa34267048079ec12a8751aa7606**](https://www.loom.com/share/4e66fa34267048079ec12a8751aa7606)

***Wokwi Link:***

<https://wokwi.com/projects/445722393261615105>

***Hand Sketch:***





***Task A***

***Code:***

// --------------------------------------------------------------

// Title      : LED Mode Control with OLED Display (Task A)

// Author     : Ayesha Iftikhar

// Reg. No.   : 23-NTU-CS-1021

// Section    : BSCS 5th (A)

// Course     : Internet of Things (IoT)

// Instructor : [Your Teacher’s Name]

// Date       : [Date of Submission]

// Description:

// This program uses two push buttons and an OLED display to control LEDs

// and a buzzer connected to an ESP32 microcontroller.

// - Button 1 cycles through four LED modes:

//   1. Both OFF

//   2. Alternate Blink

//   3. Both ON

//   4. PWM Fade

// - Button 2 resets the system back to the OFF state.

// The current mode is displayed on the OLED screen in real time.

// Tools Used:

// - Wokwi (for circuit simulation)

// - Arduino IDE (for programming)

// --------------------------------------------------------------

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

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

const int button1Pin = 32;

const int button2Pin = 33;

const int led1Pin = 5;

const int led2Pin = 4;

const int buzzerPin = 15;

volatile bool button1Pressed = false;

volatile bool button2Pressed = false;

unsigned long lastDebounceTime1 = 0;

unsigned long lastDebounceTime2 = 0;

const unsigned long debounceDelay = 150;

bool led1State = LOW;

bool led2State = LOW;

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

unsigned long lastBlinkTime = 0;

int fadeValue = 0;

int fadeStep = 5;

unsigned long lastFadeTime = 0;

void IRAM\_ATTR handleButton1() {

  button1Pressed = true;

}

void IRAM\_ATTR handleButton2() {

  button2Pressed = true;

}

void setup() {

  pinMode(button1Pin, INPUT\_PULLUP);

  pinMode(button2Pin, INPUT\_PULLUP);

  pinMode(led1Pin, OUTPUT);

  pinMode(led2Pin, OUTPUT);

  pinMode(buzzerPin, OUTPUT);

  attachInterrupt(digitalPinToInterrupt(button1Pin), handleButton1, FALLING);

  attachInterrupt(digitalPinToInterrupt(button2Pin), handleButton2, FALLING);

  // --- OLED setup ---

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

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(WHITE);

  display.setCursor(0, 0);

  display.println("Mode: Both OFF");

  display.display();

}

void loop() {

  unsigned long currentTime = millis();

  // --- Handle Button 1 (Mode Change) ---

  if (button1Pressed && (currentTime - lastDebounceTime1 > debounceDelay)) {

    button1Pressed = false;

    lastDebounceTime1 = currentTime;

    if (digitalRead(button1Pin) == LOW) {

      mode = (mode + 1) % 4;  // Cycle through modes

      updateDisplay();

    }

  }

  // --- Handle Button 2 (Reset to OFF) ---

  if (button2Pressed && (currentTime - lastDebounceTime2 > debounceDelay)) {

    button2Pressed = false;

    lastDebounceTime2 = currentTime;

    if (digitalRead(button2Pin) == LOW) {

      mode = 0;

      updateDisplay();

    }

  }

  // --- Handle each mode ---

  if (mode == 0) {  // Both OFF

    digitalWrite(led1Pin, LOW);

    digitalWrite(led2Pin, LOW);

    digitalWrite(buzzerPin, LOW);

  }

  else if (mode == 1) {  // Alternate blink

    if (currentTime - lastBlinkTime >= 500) {

      lastBlinkTime = currentTime;

      led1State = !led1State;

      led2State = !led2State;

      digitalWrite(led1Pin, led1State);

      digitalWrite(led2Pin, led2State);

      digitalWrite(buzzerPin, led1State);

    }

  }

  else if (mode == 2) {  // Both ON

    digitalWrite(led1Pin, HIGH);

    digitalWrite(led2Pin, HIGH);

    digitalWrite(buzzerPin, HIGH);

  }

  else if (mode == 3) {  // PWM Fade (non-blocking)

    if (currentTime - lastFadeTime >= 15) {  // Adjust fade speed

      lastFadeTime = currentTime;

      fadeValue += fadeStep;

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

      // Simulate PWM brightness

      int brightnessDelay = map(fadeValue, 0, 255, 1, 15);

      digitalWrite(led1Pin, HIGH);

      digitalWrite(led2Pin, LOW);

      delayMicroseconds(brightnessDelay \* 100);

      digitalWrite(led1Pin, LOW);

      digitalWrite(led2Pin, HIGH);

      delayMicroseconds(brightnessDelay \* 100);

      digitalWrite(buzzerPin, fadeValue % 2);

    }

  }

}

// --- OLED update function ---

void updateDisplay() {

  display.clearDisplay();

  display.setCursor(0, 0);

  display.setTextSize(1);

  display.setTextColor(WHITE);

  display.print("Mode: ");

  if (mode == 0) display.println("Both OFF");

  else if (mode == 1) display.println("Alternate Blink");

  else if (mode == 2) display.println("Both ON");

  else if (mode == 3) display.println("PWM Fade");

  display.display();

}

***Task B :***

***Code:***

// --------------------------------------------------------------

// Title      : Button Press Type Detection with OLED Display (Task B)

// Author     : Ayesha Iftikhar

// Reg. No.   : 23-NTU-CS-1021

// Section    : BSCS 5th (A)

// Course     : Internet of Things (IoT)

// Instructor : [Your Teacher’s Name]

// Date       : [Date of Submission]

// Description:

// This program detects short and long presses of a single push button

// using an ESP32 and displays the event on an OLED display.

// Functions:

// - Short press (< 1.5 seconds): Toggles the LED ON or OFF.

// - Long press (> 1.5 seconds): Plays a buzzer tone for 0.5 seconds.

// The system uses the Adafruit SSD1306 OLED library for display

// and the built-in tone() function for generating buzzer sound.

// Tools Used:

// - Wokwi (for simulation)

// - Arduino IDE (for programming)

// --------------------------------------------------------------

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define BUTTON 4

#define LED 2

#define BUZZER 5

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

unsigned long pressTime = 0;

bool buttonState = false;

bool lastButtonState = false;

bool ledState = false;

void setup() {

  pinMode(BUTTON, INPUT\_PULLUP);

  pinMode(LED, OUTPUT);

  pinMode(BUZZER, OUTPUT);

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

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(WHITE);

  display.setCursor(0, 0);

  display.println("Ready...");

  display.display();

}

void loop() {

  buttonState = digitalRead(BUTTON) == LOW;

  if (buttonState && !lastButtonState) {

    pressTime = millis(); // button just pressed

  }

  // when button released

  if (!buttonState && lastButtonState) {

    unsigned long pressDuration = millis() - pressTime;

    display.clearDisplay();

    display.setCursor(0, 0);

    if (pressDuration < 1500) {

      // short press → toggle LED

      ledState = !ledState;

      digitalWrite(LED, ledState);

      display.println("Short Press");

      display.println("LED Toggled");

    }

    else {

      // long press → buzzer tone

      display.println("Long Press");

      display.println("Buzzer ON");

      tone(BUZZER, 1000, 500); // 1kHz tone for 0.5s

    }

    display.display();

  }

  lastButtonState = buttonState;

}