National Textile University, Faisalabad

Name:	M. Saad Ehtsham
Class:	BS-CS(B)
Registration No:	23-NTU-CS-1072
Date:	26-oct-2025
Course Name:	Embedded systems and IOT
Submitted To:	Sir Nasir

LED MODE CONTROLLER WITH OLED

CODE :

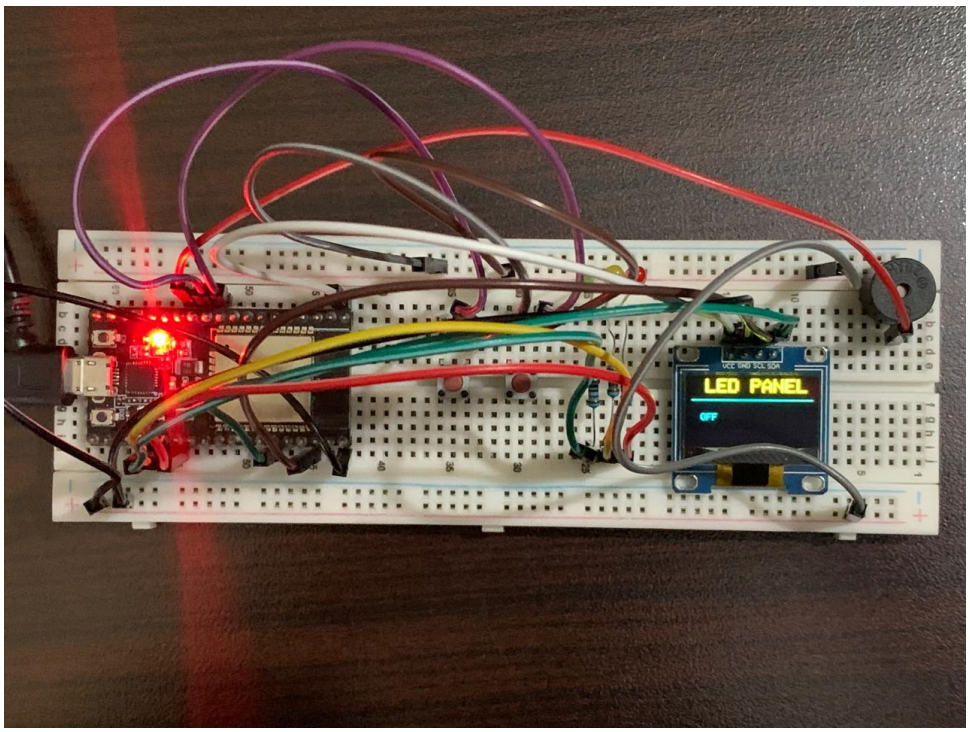
```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define W 128
#define H 64
Adafruit_SSD1306 scr(W, H, &Wire, -1);

#define LED_Y 4
#define LED_G 0
#define LED_R 2
#define BTN_MODE 26
#define BTN_RST 27

#define CH_Y 0
#define CH_G 1
#define CH_R 2
#define FREQ 4000
#define RES 8

hw_timer_t *blinkT = nullptr;

int modeSel = 0;
int blinkStep = 0;
bool oldMode = HIGH;
bool oldRst = HIGH;
unsigned long tPrev = 0;
const int tDelay = 600;
volatile unsigned long tick = 0;

void IRAM_ATTR timerTick() {
  tick++;
}

void showScreen() {
  scr.clearDisplay();
  scr.setTextSize(2);
  scr.setTextColor(SSD1306_WHITE);
```

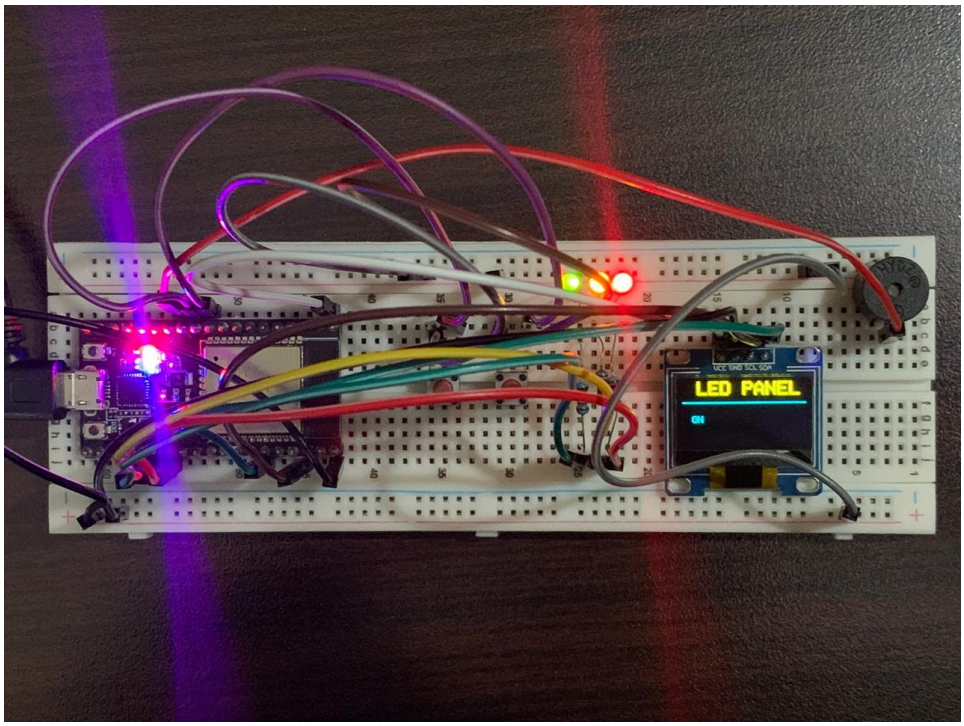
```
  scr.setCursor(15, 0);
  scr.println("LED PANEL");
  scr.drawLine(0, 20, 127, 20, SSD1306_WHITE);
  scr.setTextSize(1);
  scr.setCursor(10, 35);
  if (modeSel == 0) scr.print("OFF");
  else if (modeSel == 1) scr.print("Blink");
  else if (modeSel == 2) scr.print("ON");
  else if (modeSel == 3) scr.print("PWM");
  scr.display();
}

void setup() {
  Serial.begin(115200);
  pinMode(LED_Y, OUTPUT);
  pinMode(LED_G, OUTPUT);
  pinMode(LED_R, OUTPUT);
  pinMode(BTN_MODE, INPUT_PULLUP);
  pinMode(BTN_RST, INPUT_PULLUP);

  if (!scr.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    while (true);
  }

  ledcSetup(CH_Y, FREQ, RES);
  ledcSetup(CH_G, FREQ, RES);
  ledcSetup(CH_R, FREQ, RES);
  ledcAttachPin(LED_Y, CH_Y);
  ledcAttachPin(LED_G, CH_G);
  ledcAttachPin(LED_R, CH_R);

  blinkT = timerBegin(0, 80, true);
  timerAttachInterrupt(blinkT, &timerTick, true);
  timerAlarmWrite(blinkT, 1000000, true);
  timerAlarmEnable(blinkT);

  ledcWrite(CH_Y, 0);
  ledcWrite(CH_G, 0);
  ledcWrite(CH_R, 0);
  showScreen();
}

void loop() {
  bool nowMode = digitalRead(BTN_MODE);
  bool nowRst = digitalRead(BTN_RST);
```

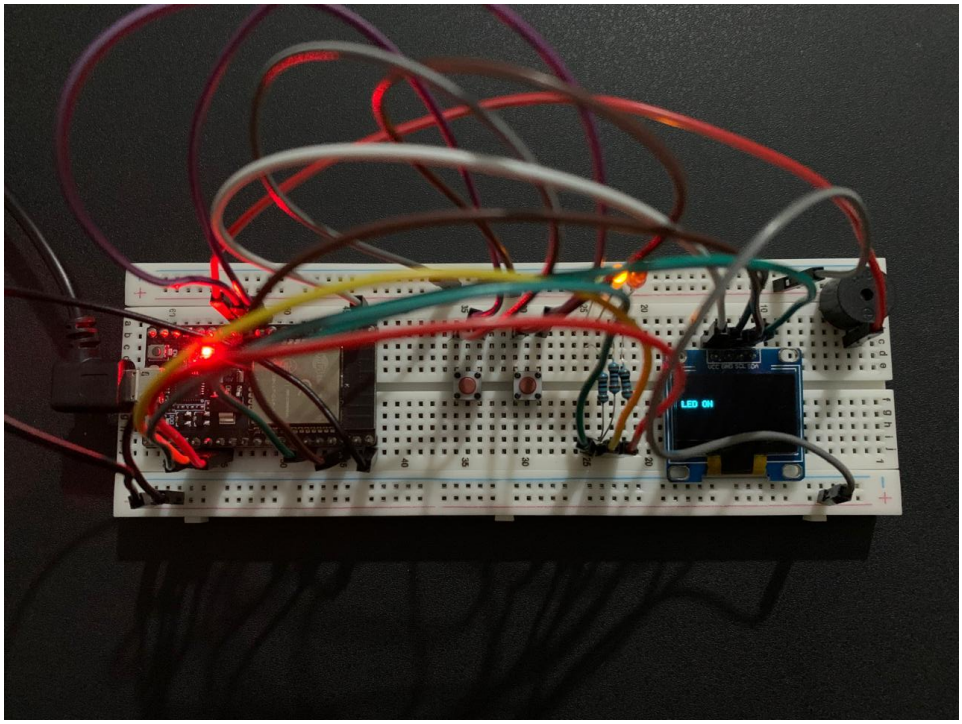
```
if (millis() - tPrev > tDelay) {
  if (nowMode == LOW && oldMode == HIGH) {
    modeSel = (modeSel + 1) % 4;
    blinkStep = 0;
    showScreen();
    tPrev = millis();
  }
  if (nowRst == LOW && oldRst == HIGH) {
    modeSel = 0;
    blinkStep = 0;
    showScreen();
    tPrev = millis();
  }
}

oldMode = nowMode;
oldRst = nowRst;

if (modeSel == 0) {
  ledcWrite(CH_Y, 0);
  ledcWrite(CH_G, 0);
  ledcWrite(CH_R, 0);
}
else if (modeSel == 1) {
  static unsigned long lastTick = 0;
  if (tick != lastTick) {
    lastTick = tick;
    blinkStep = (blinkStep + 1) % 3;
    if (blinkStep == 0) {
      ledcWrite(CH_Y, 255);
      ledcWrite(CH_G, 0);
      ledcWrite(CH_R, 0);
    } else if (blinkStep == 1) {
      ledcWrite(CH_Y, 0);
      ledcWrite(CH_G, 255);
      ledcWrite(CH_R, 0);
    } else {
      ledcWrite(CH_Y, 0);
      ledcWrite(CH_G, 0);
      ledcWrite(CH_R, 255);
    }
  }
}
else if (modeSel == 2) {
```

```
    ledcWrite(CH_Y, 255);
    ledcWrite(CH_G, 255);
    ledcWrite(CH_R, 255);
  }
  else if (modeSel) {
    for (int i = 0; i <= 255 && modeSel == 3; i++) {
      ledcWrite(CH_Y, i);
      ledcWrite(CH_G, i);
      ledcWrite(CH_R, i);
      delay(5);
      if (digitalRead(BTN_MODE) == LOW || digitalRead(BTN_RST) == LOW) return;
    }
    for (int i = 255; i >= 0 && modeSel == 3; i--) {
      ledcWrite(CH_Y, i);
      ledcWrite(CH_G, i);
      ledcWrite(CH_R, i);
      delay(5);
      if (digitalRead(BTN_MODE) == LOW || digitalRead(BTN_RST) == LOW) return;
    }
  }
}
```

OUTPUT :




LED PANEL


LED ON

WOKWI LINK :

https://wokwi.com/projects/445728441175956481

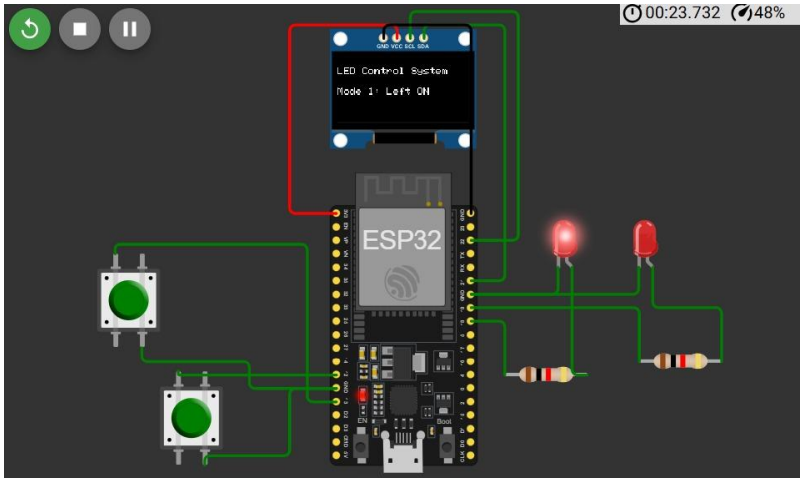
WOKWI CODE :

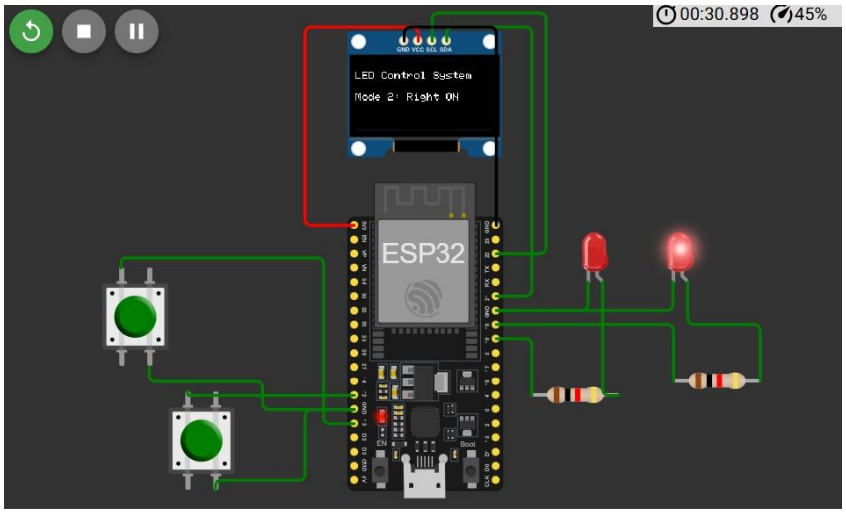
```
/* NAME : Saad Ehtsham */
/* Reg no : 23-NTU-CS-1072 */
/* TITLE : LED MODE CONTROLLER WITH OLED + PWM FADE (ESP32 analogWrite) */

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

// OLED configuration
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// Pin definitions
#define LED_LEFT 18
#define LED_RIGHT 19
#define BTN_RESET 12
#define BTN_MODE 13

// Fade parameters
#define FADE_INTERVAL 5  // ms per brightness step

// Variables
int mode = 0; // 0=Both OFF, 1=Left ON, 2=Right ON, 3=Both ON
unsigned long lastDebounce = 0;
const unsigned long debounceDelay = 200;
int leftBrightness = 0;
int rightBrightness = 0;
unsigned long lastFadeTime = 0;

void setup() {
  Serial.begin(115200);

  // Initialize OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found!");
```

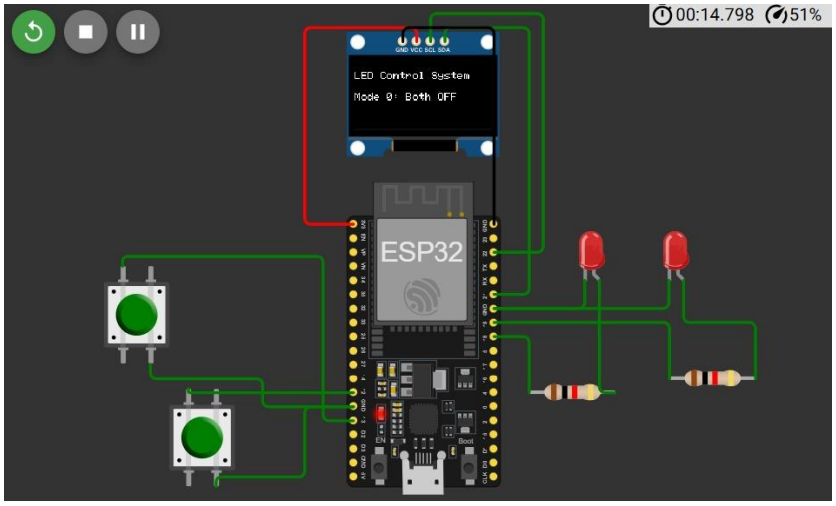
```
    for (;;);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 10);
  display.println("System Starting...");
  display.display();
  delay(1000);

  // Initialize LEDs
  pinMode(LED_LEFT, OUTPUT);
  pinMode(LED_RIGHT, OUTPUT);

  // Initialize buttons with internal pull-ups
  pinMode(BTN_MODE, INPUT_PULLUP);
  pinMode(BTN_RESET, INPUT_PULLUP);

  updateDisplay();
}

void loop() {
  unsigned long currentMillis = millis();

  // Button handling
  if (digitalRead(BTN_MODE) == LOW && currentMillis - lastDebounce >
debounceDelay) {
    mode = (mode + 1) % 4;
    lastDebounce = currentMillis;
    updateDisplay();
  }

  if (digitalRead(BTN_RESET) == LOW && currentMillis - lastDebounce >
debounceDelay) {
    mode = 0;
    lastDebounce = currentMillis;
    updateDisplay();
  }

  // Fade LEDs smoothly using analogWrite
  if (currentMillis - lastFadeTime >= FADE_INTERVAL) {
    lastFadeTime = currentMillis;

    int targetLeft = (mode == 1 || mode == 3) ? 255 : 0;
    int targetRight = (mode == 2 || mode == 3) ? 255 : 0;
```

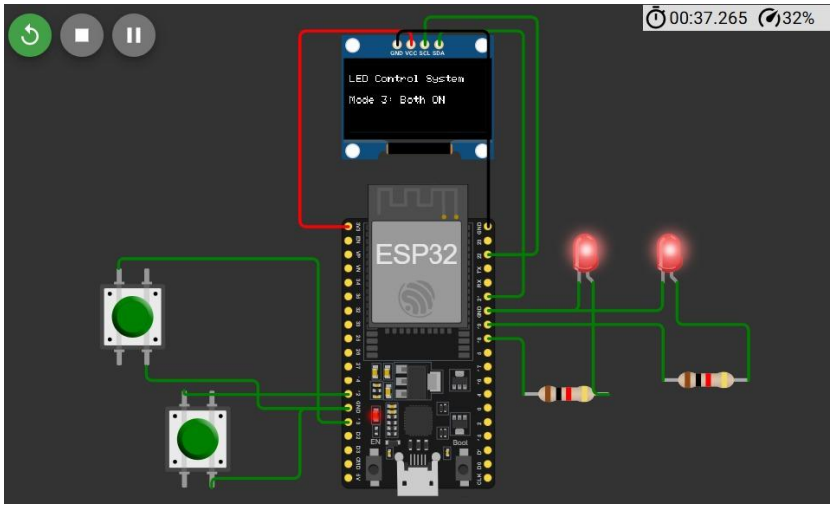
```
    if (leftBrightness < targetLeft) leftBrightness++;
    else if (leftBrightness > targetLeft) leftBrightness--;

    if (rightBrightness < targetRight) rightBrightness++;
    else if (rightBrightness > targetRight) rightBrightness--;

    analogWrite(LED_LEFT, leftBrightness);
    analogWrite(LED_RIGHT, rightBrightness);
  }
}

// OLED display function
void updateDisplay() {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 10);
  display.println("LED Control System");
  display.setCursor(0, 30);

  switch (mode) {
    case 0: display.println("Mode 0: Both OFF"); break;
    case 1: display.println("Mode 1: Left ON"); break;
    case 2: display.println("Mode 2: Right ON"); break;
    case 3: display.println("Mode 3: Both ON"); break;
  }

  display.display();
}
```

WOKWI OUTPUT:




00:30.898
45%
LED Control System
Mode 2: Right ON
ESP32


00:14.798
51%
LED Control System
Mode 0: Both OFF
ESP32


00:37.265
32%
LED Control System
Mode 3: Both ON
ESP32