National Textile University, Faisalabad

Department of Computer Science

Assignment # 1

Name	Rameen Fatima
Section	BSCS-B
Semester	5th
Registration no.	23-NTU-CS-1086
Course title	Embedded IOT systems
Submitted to	Sir Nasir
Submission date	23-10-2025

Documentation of Task A

Question 3

Implementation

ESP32 LED Mode Cycling with OLED Display

Circuit & Working Explanation

---

1. Introduction

This project demonstrates how an ESP32 microcontroller can control three LEDs using two push buttons, with real-time feedback displayed on an OLED screen.
The system implements PWM (Pulse Width Modulation) to control LED brightness and supports multiple operation modes that are selected using the buttons.

---

2. Circuit Overview

- The ESP32 controls three LEDs connected to GPIO 25, 26, and 33.
- Two push buttons are connected as:
  - **GPIO 14** → Mode Button (to switch between LED modes)
  - **GPIO 27** → Reset Button (to return to initial mode)
- One **OLED Display** (0.96", 128×64, I2C) is connected as:
  - **SDA → GPIO 21**
  - **SCL → GPIO 22**
- Each LED is connected in series with a 220Ω resistor for current limiting.
- All components share a common ground (GND) connection with the ESP32.

Pin Diagram		
Component	**ESP32 Pin**	**Description**
LED 1	25	First LED (PWM CH 0)
LED 2	26	Second LED (PWM CH 1)
LED 3	33	Third LED (PWM CH 2)
Mode Button	14	Switches between modes
Reset Button	27	Resets to Mode 0 (OFF)
OLED SDA	21	I²C Data Line
OLED SCL	22	I²C Clock Line
OLED VCC	3.3V	Power Supply
OLED GND	GND	Common Ground

---

4. Working Principle

Step 1: Initialization

- ESP32 initializes I/O pins for LEDs, buttons, and OLED.
- Configures PWM channels for LED brightness control.
- Displays “System Ready” message on OLED at startup.

Step 2: Button Input Handling

- **Mode Button (GPIO 14):** Cycles through the 4 LED modes.
- **Reset Button (GPIO 27):** Returns to the default OFF state.
- Buttons use internal pull-up resistors and software debouncing for stable input.

Step 3: Operating Modes

1. **Mode 0 – All LEDs OFF**
   Turns off all LEDs and shows “ALL OFF” on OLED.
2. **Mode 1 – Alternate Blinking**
   LEDs blink one after another in sequence (LED1 → LED2 → LED3).

3. **Mode 2 – All LEDs ON**
   All LEDs glow continuously.
4. **Mode 3 – PWM Fading Mode**
   LED brightness gradually increases and decreases using PWM.

Step 4: PWM Control

- PWM allows analog-like brightness control using digital signals.
- **Channels:** 0, 1, 2
- **Frequency:** 5 kHz
- **Resolution:** 8-bit (0–255 duty cycle)

Step 5: OLED Display Feedback

- OLED shows the active mode (e.g., “Mode 1: Alternate Blink”).
- In fade mode, a progress bar represents LED brightness.
- Updates dynamically with each button press.

---

5. Key Concepts Demonstrated

- **PWM (Pulse Width Modulation):** For LED brightness control.
- **Debouncing:** To avoid multiple false button presses.
- **I²C Communication:** Between ESP32 and OLED display.
- **State Machine Logic:** For smooth mode switching.
- **Real-Time Feedback:** User sees active mode on OLED.

CODE:

```
/*
  ------------------------------------------------------------
  Title       : ESP32 LED Control with OLED Display (3 LEDs)
  Author      : Rameen Fatima
  Reg. no.    : 23-NTU-CS-1086  (BSCS-5TH B)
  Description :
      This program uses two push buttons to control LED operation modes
      and display the current mode on an OLED screen.
      The circuit includes:
        • 3 LEDs (Pin 25, 26, 33)
        • 2 Push Buttons (Mode=14, Reset=27)
        • 1 OLED Display (I2C: SDA=21, SCL=22)
        • PWM-controlled LED brightness fade mode
  ------------------------------------------------------------
*/

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

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_ADDR 0x3C
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
// --- Pin Configuration ---
#define LED1 25          // First LED pin
#define LED2 26          // Second LED pin
#define LED3 33          // Third LED pin
#define BTN_MODE 14      // Button to change LED mode
#define BTN_RESET 27     // Button to reset to OFF mode

// --- PWM Configuration ---
#define PWM_CH1 0        // PWM Channel for LED1
#define PWM_CH2 1        // PWM Channel for LED2
#define PWM_CH3 2        // PWM Channel for LED3
#define PWM_FREQ 5000    // PWM frequency in Hz
#define PWM_RES 8        // PWM resolution (8-bit = 0–255)

int mode = 0;                // Current LED mode
unsigned long lastPress = 0; // For button debounce
int brightness = 0;          // Current LED brightness (for fade)
int fadeAmount = 10;         // Brightness change per loop in fade mode

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

  // --- Configure buttons as input with internal pull-up ---
  pinMode(BTN_MODE, INPUT_PULLUP);
  pinMode(BTN_RESET, INPUT_PULLUP);
```

```
  // --- Setup PWM for LEDs ---
  ledcSetup(PWM_CH1, PWM_FREQ, PWM_RES);
  ledcSetup(PWM_CH2, PWM_FREQ, PWM_RES);
  ledcSetup(PWM_CH3, PWM_FREQ, PWM_RES);
  ledcAttachPin(LED1, PWM_CH1);
  ledcAttachPin(LED2, PWM_CH2);
  ledcAttachPin(LED3, PWM_CH3);

  // --- Initialize OLED Display ---
  Wire.begin(21, 22);  // Set custom I2C pins
  if (!display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR)) {
    Serial.println("OLED not found!");
    while (true);  // Halt program if OLED fails
  }

  // --- Startup message ---
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(15, 25);
  display.println("System Ready...");
  display.display();
  delay(1000);
  display.clearDisplay();
}
```

```
/*
  drawUIFrame():
  Draws the OLED layout including header, border,
  and footer with pin information. Displays current mode text.
*/
void drawUIFrame(String modeText) {
  display.clearDisplay();

  // --- Header Bar ---
  display.fillRect(0, 0, 128, 12, SSD1306_WHITE);     // top header bar
  display.setTextColor(SSD1306_BLACK);
  display.setCursor(3, 2);
  display.print("ESP32 LED Controller");

  // --- Outer Border ---
  display.drawRect(0, 0, SCREEN_WIDTH, SCREEN_HEIGHT, SSD1306_WHITE);

  // --- Mode Title ---
  display.setTextColor(SSD1306_WHITE);
  display.setTextSize(1);
  display.setCursor(5, 18);
  display.print("Mode: ");
  display.println(modeText);

  // --- Footer Bar (Instructions) ---
  display.drawLine(0, 54, 128, 54, SSD1306_WHITE);
  display.setCursor(5, 56);
  display.print("MODE=14  RESET=27");
}
```

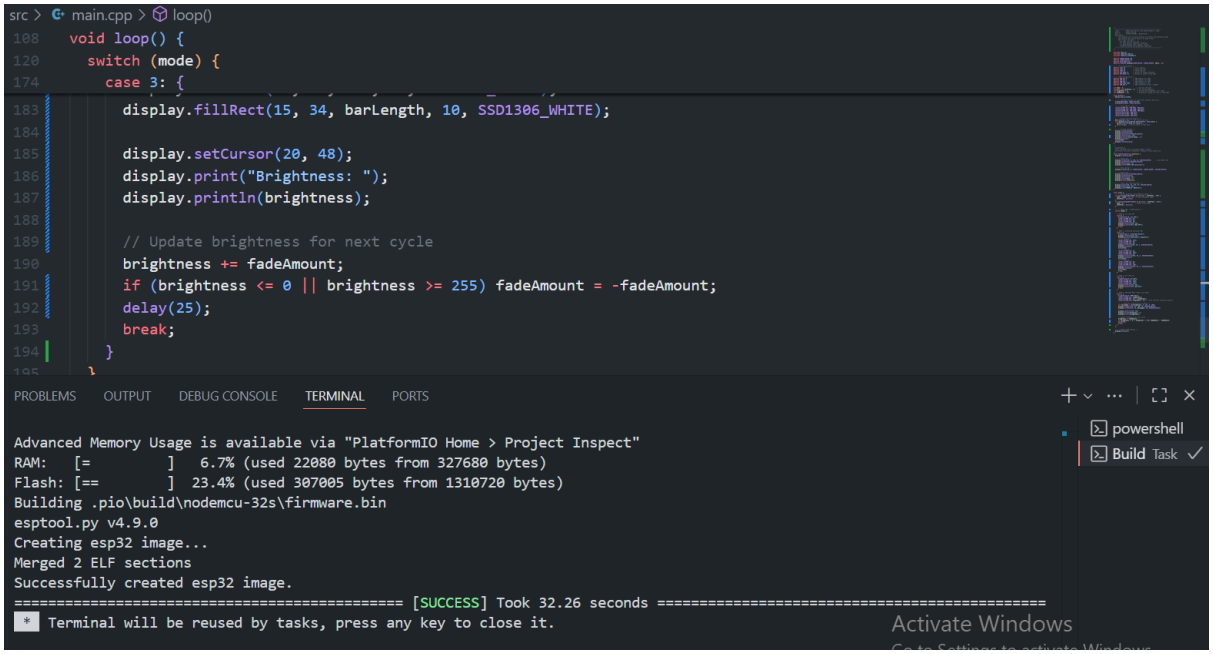
```
void loop() {
  // --- Button Handling (with debounce delay) ---
  if (!digitalRead(BTN_MODE) && millis() - lastPress > 250) {
    mode = (mode + 1) % 4;  // Cycle through 4 modes
    lastPress = millis();
  }
  if (!digitalRead(BTN_RESET) && millis() - lastPress > 250) {
    mode = 0;               // Reset to OFF mode
    lastPress = millis();
  }

  // --- Mode Logic + OLED Display ---
  switch (mode) {

    // MODE 0: All LEDs OFF
    case 0: {
      drawUIFrame("0: ALL OFF");
      ledcWrite(PWM_CH1, 0);
      ledcWrite(PWM_CH2, 0);
      ledcWrite(PWM_CH3, 0);
      display.setCursor(20, 35);
      display.println("All LEDs OFF");
      break;
    }
```

```
    // MODE 1: Alternate Blinking LEDs
    case 1: {
      drawUIFrame("1: Alternate Blink");
      display.setCursor(20, 35);
      display.println("Blinking in Sequence");
      // --- LED1 ON ---
      ledcWrite(PWM_CH1, 255);
      ledcWrite(PWM_CH2, 0);
      ledcWrite(PWM_CH3, 0);
      display.fillCircle(95, 38, 3, SSD1306_WHITE);
      display.display();
      delay(300);
      // --- LED2 ON ---
      ledcWrite(PWM_CH1, 0);
      ledcWrite(PWM_CH2, 255);
      ledcWrite(PWM_CH3, 0);
      display.fillCircle(105, 38, 3, SSD1306_WHITE);
      display.display();
      delay(300);
      // --- LED3 ON ---
      ledcWrite(PWM_CH1, 0);
      ledcWrite(PWM_CH2, 0);
      ledcWrite(PWM_CH3, 255);
      display.fillCircle(115, 38, 3, SSD1306_WHITE);
      display.display();
      delay(300);
      break;
```

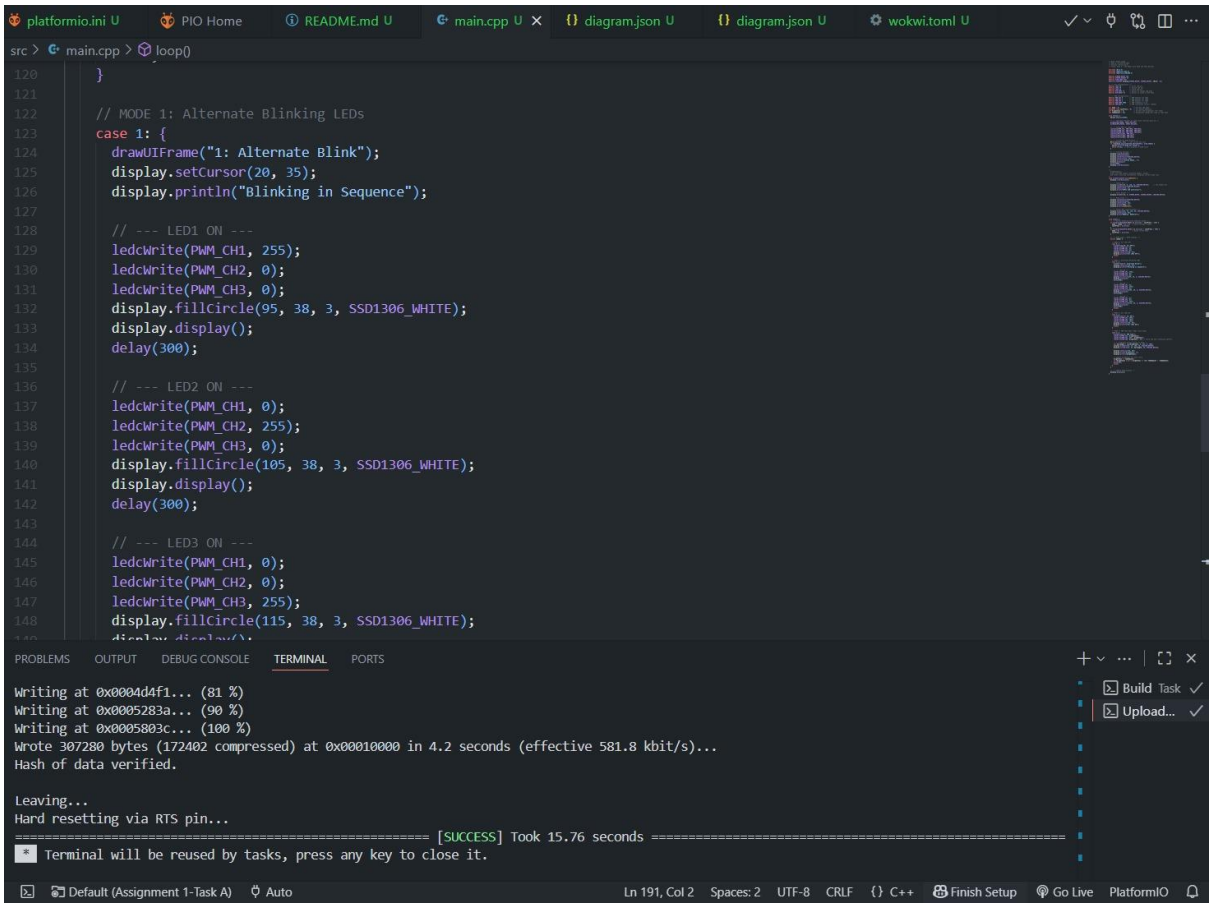
```
  // MODE 2: All LEDs ON
  case 2: {
    drawUIFrame("2: All ON");
    ledcWrite(PWM_CH1, 255);
    ledcWrite(PWM_CH2, 255);
    ledcWrite(PWM_CH3, 255);
    display.setCursor(20, 35);
    display.println("All LEDs ON");
    break;
  }

  // MODE 3: PWM Fade Mode (LEDs cross-fade)
  case 3: {
    drawUIFrame("3: PWM Fade");
    ledcWrite(PWM_CH1, brightness);
    ledcWrite(PWM_CH2, 255 - brightness);
    ledcWrite(PWM_CH3, brightness / 2); // third LED half intensity pattern

    // --- Fade Bar Visualization on OLED ---
    int barLength = map(brightness, 0, 255, 0, 100);
    display.drawRect(15, 34, 100, 10, SSD1306_WHITE);
    display.fillRect(15, 34, barLength, 10, SSD1306_WHITE);

    display.setCursor(20, 48);
    display.print("Brightness: ");
    display.println(brightness);
```

```
      // Update brightness for next cycle
      brightness += fadeAmount;
      if (brightness <= 0 || brightness >= 255) fadeAmount = -fadeAmount;
      delay(25);
      break;
    }
  }

  // --- Update OLED Display ---
  display.display();
}
```

Build success:

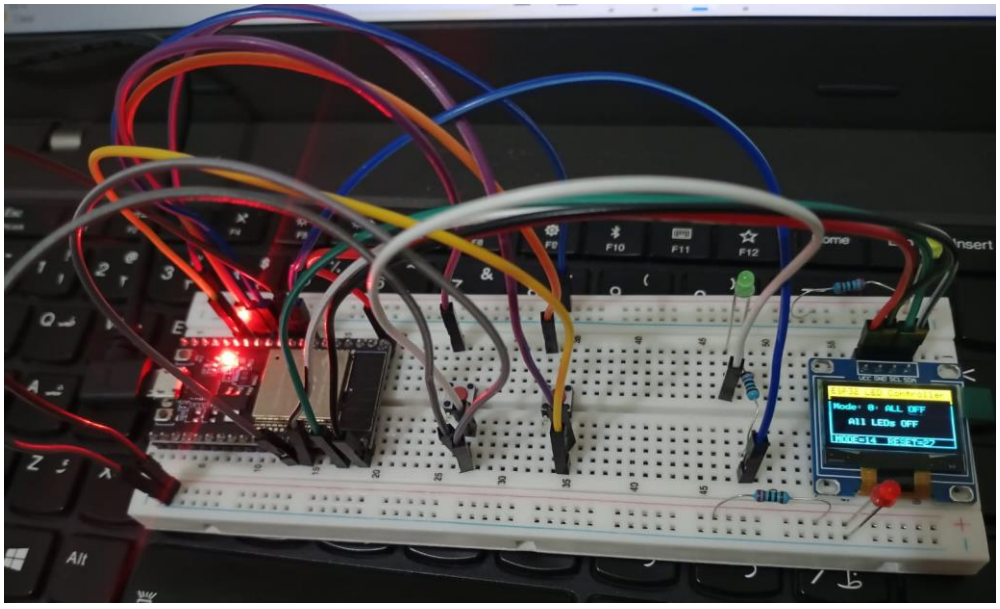


Upload success:

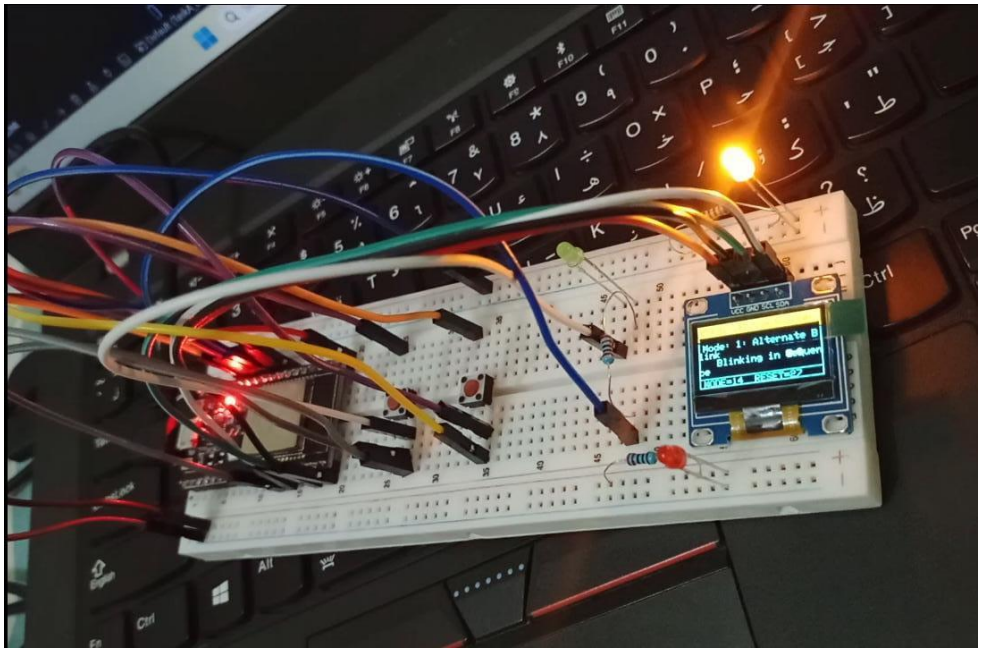


Hardware output:

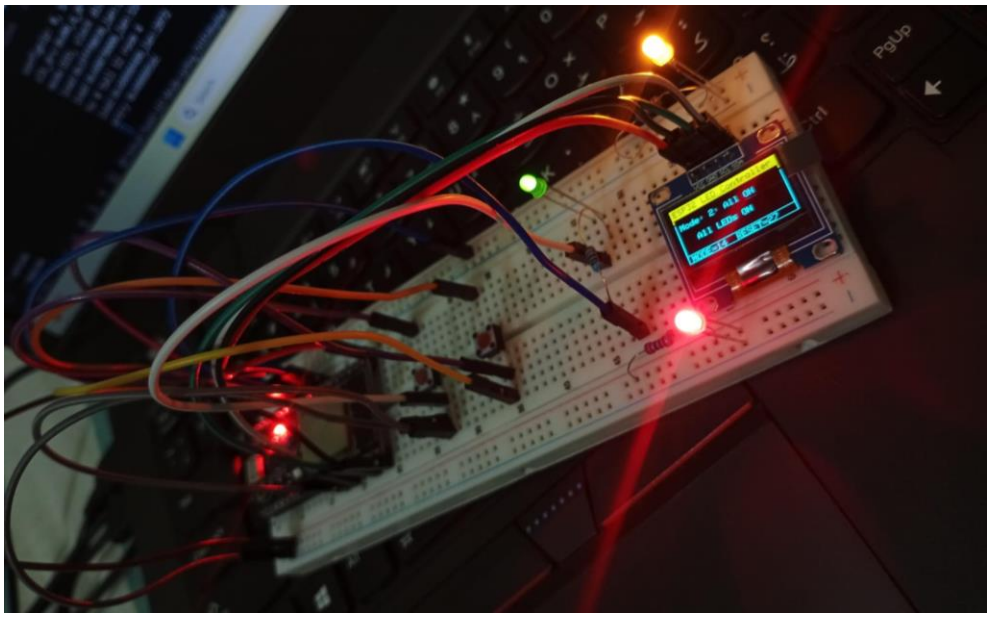
Mode 0 (ALL OFF):



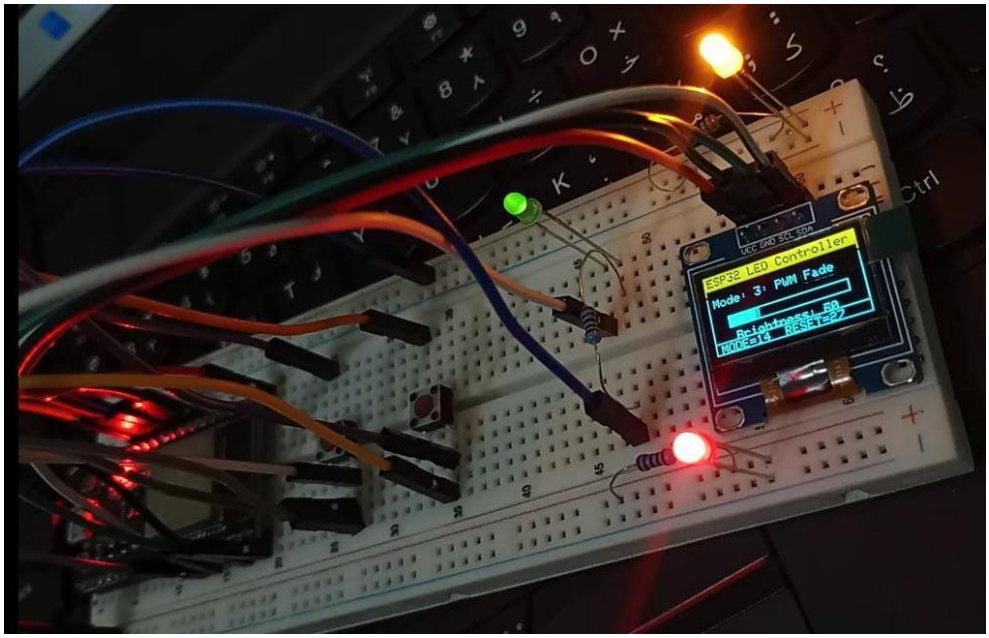
Mode 1 (Alternate blink):



Mode 2 (ALL ON):



Mode 3 (PWM fade):



Handwritten code images:

Rameen Fatima 23-NTU-CS-1086 BSCS-5th B

ASSIGNMENT 1

Task (a)

LED mode cycling with OLED display:

Code :-

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

# define SCREEN_WIDTH 128
# define SCREEN_HEIGHT 64
# define OLED_ADDR 0x3C

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

// --- Pin Configuration ---

# define LED1 25
# define LED2 26
# define LED3 33
# define BTN_Mode 14
# define BTN_RESET 27

// --- PWM Configuration ---

# define PWM_CH1 0
# define PWM_CH2 1
# define PWM_CH3 2
# define PWM_FREQ 5000
```

```
# define PWM_RES 8

int mode = 0;
unsigned long lastPress = 0;
int brightness = 0;
int fadeAmount = 10;

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

pinMode(BTN_Mode, INPUT_PULLUP);
pinMode(BTN_RESET, INPUT_PULLUP);

ledcSetup(PWM_CH1, PWM_FREQ, PWM_RES);
ledcSetup(PWM_CH2, PWM_FREQ, PWM_RES);
ledcSetup(PWM_CH3, PWM_FREQ, PWM_RES);
ledcAttachPin(LED1, PWM_CH1);
ledcAttachPin(LED2, PWM_CH2);
ledcAttachPin(LED3, PWM_CH3);

Wire.begin(21, 22);
if (!display.begin(SSD1306_SWITCHAPVCC,
                   OLED_ADDR)) {
  Serial.println("OLED not found!");
  while (true);
}

// --- Startup Message ---

display.clearDisplay();
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(15, 25);
```

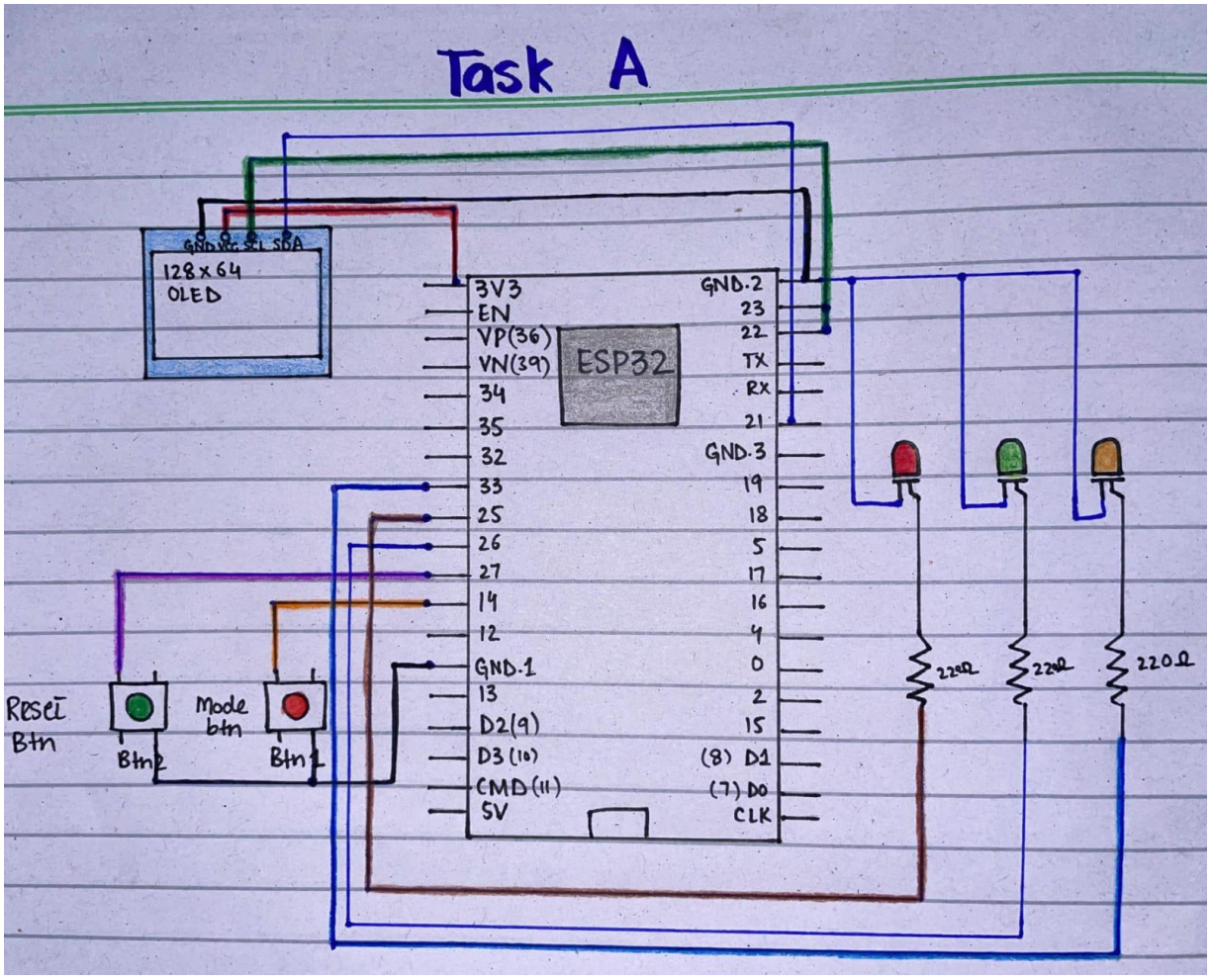
```
  display.println("System Ready...");
  display.display();
  delay(1000);
  display.clearDisplay();
}

void drawUIFrame(String modeText) {
  display.clearDisplay();

  display.fillRect(0,0,128,12,SSD1306_WHITE);
  display.setTextColor(SSD1306_BLACK);
  display.setCursor(3,2);
  display.print("ESP32 LED Controller");

  display.drawRect(0, 0, SCREEN_WIDTH,
  SCREEN_HEIGHT, SSD1306_WHITE);

  display.setTextColor(SSD1306_WHITE);
  display.setTextSize(1);
  display.setCursor(5,18);
  display.print("Mode: ");
  display.println(modeText);

  display.drawLine(0,54,128,54,SSD1306_WHITE);
  display.setCursor(5,56);
  display.print("Mode = 14    RESET = 27");

void loop(){

  if(!digitalRead(BTN_Mode) && millis()-
      lastPress > 250){
    mode = (mode+1) % 4;
    lastPress = millis();
  }
```

```
if (!digitalRead(BTN_RESET) && millis() -
    lastPress > 250) {

    mode = 0;
    lastPress = millis();
}

// --- Mode logic + OLED display ---

switch (mode) {

// MODE 0 : ALL LEDs OFF
case 0: {
  drawUIFrame ("0: ALL OFF");
  ledcWrite (PWM_CH1, 0);
  ledcWrite (PWM_CH2, 0);
  ledcWrite (PWM_CH3, 0);
  display.setCursor (20, 35);
  display.println ("ALL LEDs OFF");
  break;
}

// MODE 1: Alternate Blinking LEDs
case 1: {
  drawUIFrame ("1: Alternate Blink");
  display.setCursor (20, 35);
  display.println ("Blinking in sequence");

  ledcWrite (PWM_CH1, 255);
  ledcWrite (PWM_CH2, 0);
  ledcWrite (PWM_CH3, 0);
  display.fillCircle (95, 38, 3, SSD1306_WHITE);
  display.display();
```

```
    delay(300);

    ledcWrite(PWM_CH1, 0);
    ledcWrite(PWM_CH2, 255);
    ledcWrite(PWM_CH3, 0);
    display.fillCircle(105, 38, 3, SSD1306_WHITE);
    display.display();
    delay(300);

    ledcWrite(PWM_CH1, 0);
    ledcWrite(PWM_CH2, 0);
    ledcWrite(PWM_CH3, 255);
    display.fillCircle(115, 38, 3, SSD1306_WHITE);
    display.display();
    delay(300);
    break;
  }

  // MODE 2 : All LEDs ON
  case 2: {
  drawUIFrame("2: ALL ON");
  ledcWrite(PWM_CH1, 255);
  ledcWrite(PWM_CH2, 255);
  ledcWrite(PWM_CH3, 255);
  display.setCursor(20, 35);
  display.println("ALL LEDs ON");
  break;
  }

  // MODE 3 : PWM Fade Mode
  case 3: {
  drawUIFrame("3: PWM Fade");
  ledcWrite(PWM_CH1, brightness);
```

```
ledcWrite (PWM_CH2, 255-brightness);
ledc Write (PWM_CH3, brightness / 2);

int barlength = map (brightness, 0, 255,
                     0, 100);
display.drawRect (15, 34, 100, 10, SSD1306_WHITE);
display.fillRect (15,34, barlength, 10, SSD1306_WHITE);

display.setCursor (20, 48);
display.print ("Brightness : ");
display.println (brightness);

brightness += fadeAmount;
if (brightness <=0 || brightness >= 255)
  fadeAmount = -fadeAmount;
delay (25);
break;
}
}

display.display ();
}
```

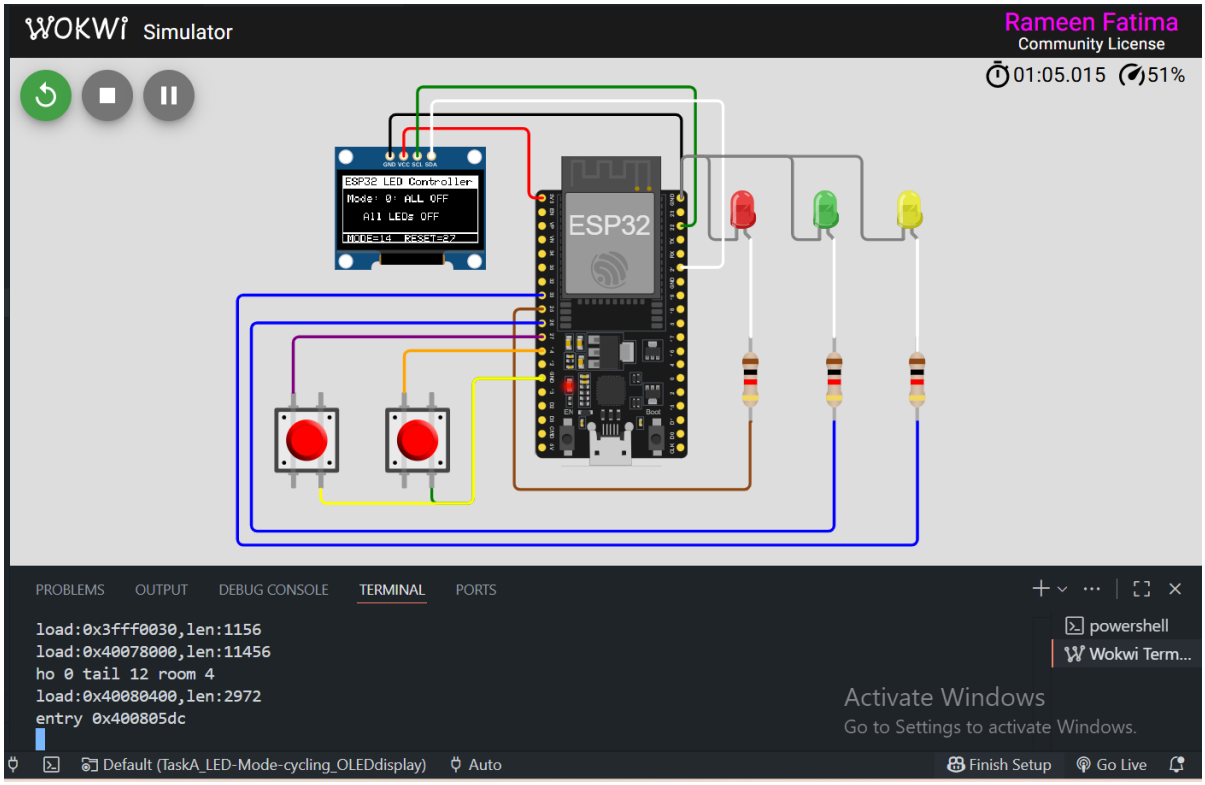
Hand sketch Diagram:



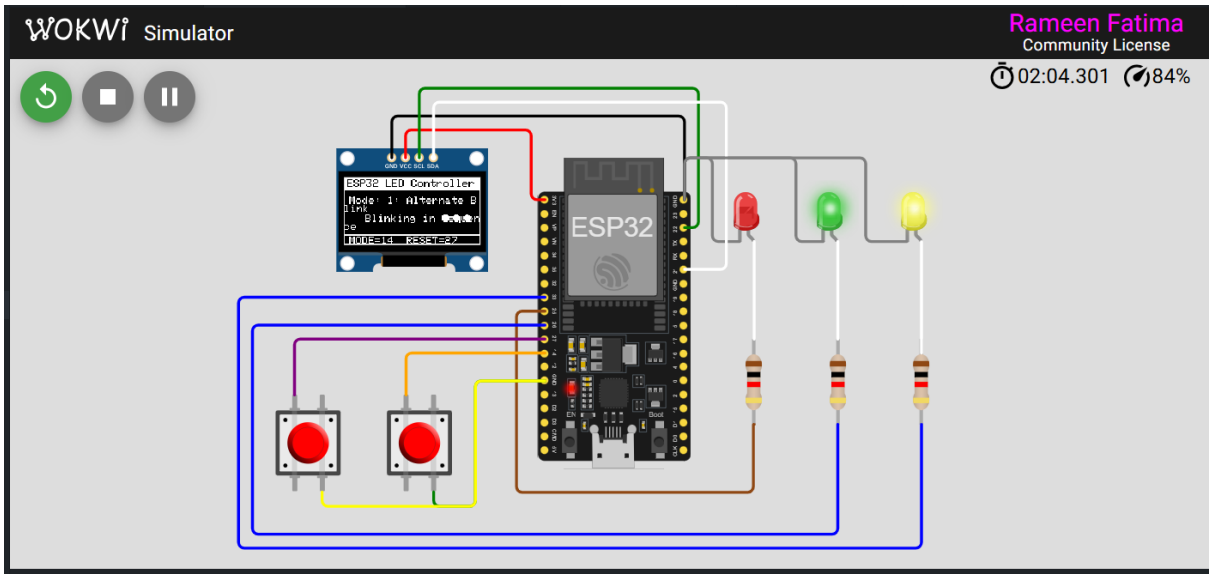
Wokwi circuit diagrams:

Button1:

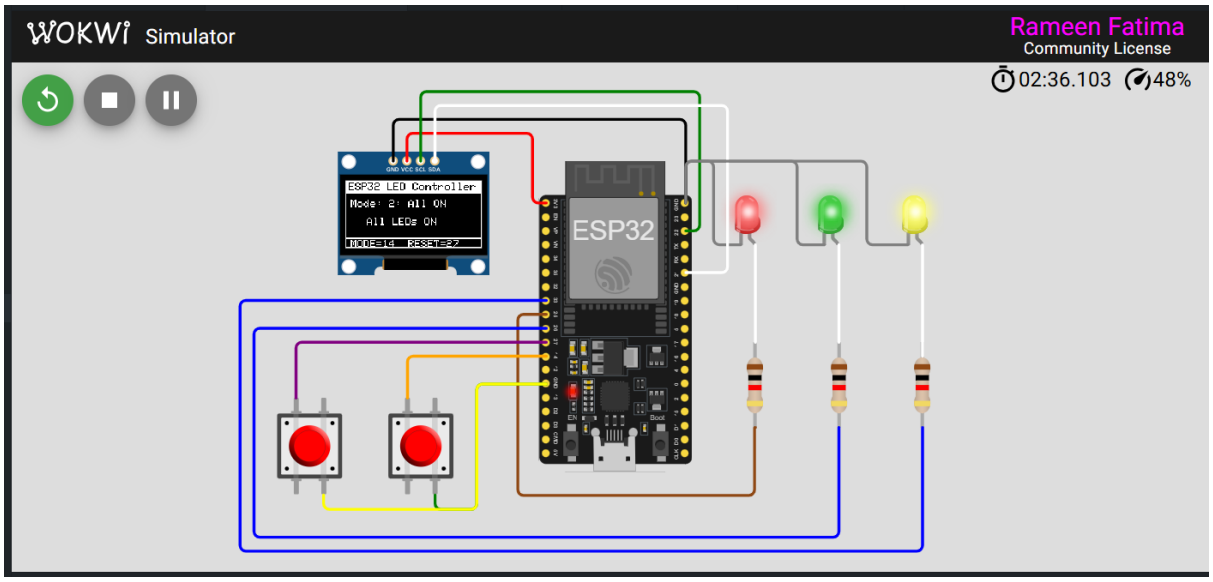
Mode 0 (ALL OFF):



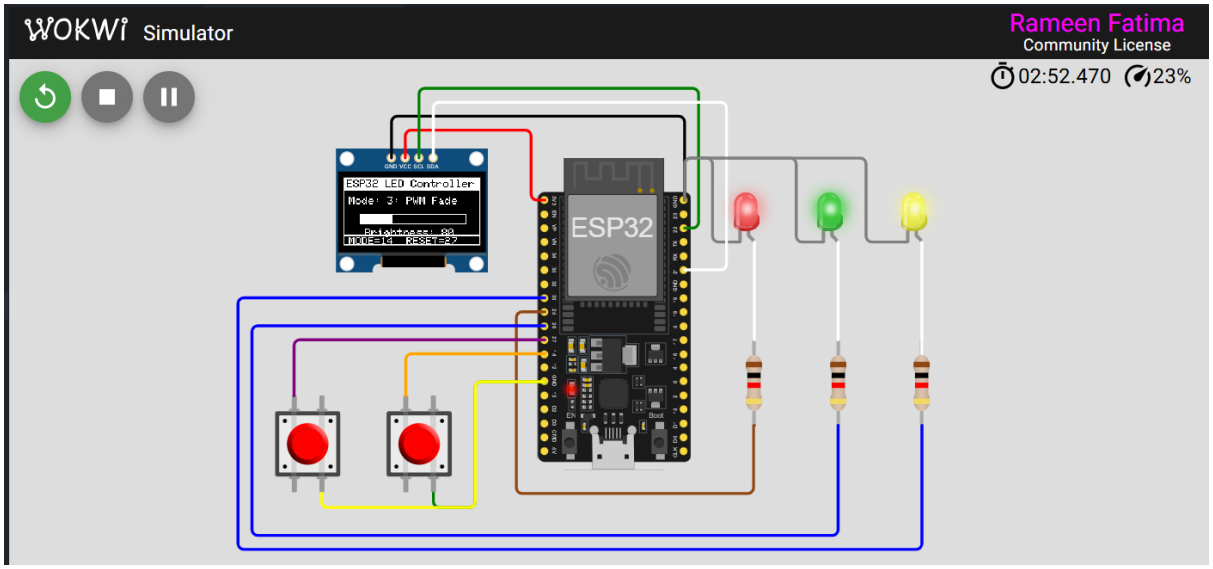
Mode 1 (Alternate Blink):



Mode 2 (ALL ON):

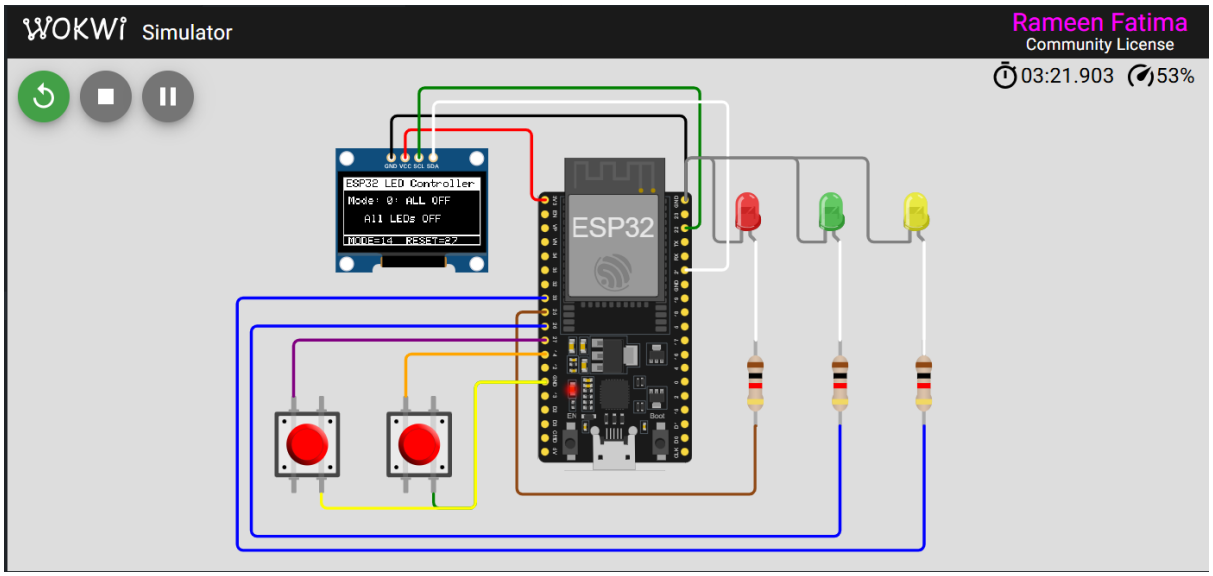


Mode 3 (PWM fade):



Button 2:

Reset to Mode 0:



Wokwi Project Link:

https://wokwi.com/projects/445351775582333953