Task 1 Documentation

Title: LED Control using Two Push Buttons and Display OLED

Name: Maryam Hamid

Registration No: 23-NTU-CS-1046

Course: Embedded IoT Systems (CSE-3079)

Objecve

To design an ESP32-based system that controls three LEDs using **two push butons** and displays the current operang mode on a **128×64 OLED**.
The project demonstrates concepts of **non-blocking programming**, **buton debouncing**, **soware PWM**, and **OLED interfacing** via I²C.

Hardware Components

Component	Quanty	Descripon
ESP32 (NodeMCU-32S)	1	Main microcontroller
Push Butons	2	For mode change and reset
LEDs (Red, Green, Blue)	3	Visual indicators for LED modes
OLED Display (SSD1306 128×64)	1	Displays the current LED mode
Resistors (220Ω)	3	Limit current for LEDs
Jumper Wires	—	For connecons
Breadboard	1	For circuit assembly

Pin Configuraon

Component Funcon	ESP32 GPIO Pin

Red LED	Output	16
Green LED	Output	17
Blue LED	Output	18

Mode Buton Input (with internal pull-up) 32

Reset Buton Input (with internal pull-up) 33

OLED SDA	I²C Data	21
OLED SCL	I²C Clock	22
OLED VCC	Power	3.3V
OLED GND	Ground	GND

Working Principle

The ESP32 reads buton inputs and controls three LEDs based on the selected **mode**. The OLED displays the current mode name for user feedback.

Buton Funcons

- **Mode Buton (GPIO 32):** Cycles through 4 modes:
    1. All OFF
    2. Alternate Blink (LEDs blink one by one)
    3. All ON
    4. PWM Fade (LED brightness smoothly increases and decreases)
- **Reset Buton (GPIO 33):**
  Immediately resets to Mode 0 (All LEDs OFF) and updates the OLED.

Key Features

- Uses **non-blocking millis()-based ming** (no delay() in LED control loops).
- Implements **debouncing** for both butons to prevent mulple unwanted triggers.
- Implements **soware-based PWM** (brightness control) for fade effect — works in Wokwi even without analogWrite.

- OLED uses the **Adafruit_GFX** and **Adafruit_SSD1306** libraries over the **I²C bus**.

Mode Descripon

Mode	Name	Behavior	OLED Display
0	All OFF	All LEDs off	“Mode: All OFF”
1	Alternate Blink	LEDs blink sequenally (R→G→B)	“Mode: Alternate Blink”
2	All ON	All three LEDs ON	“Mode: All ON”
3	PWM Fade	All LEDs fade in and out smoothly	“Mode: PWM Fade”

Soware Overview

Libraries Used:

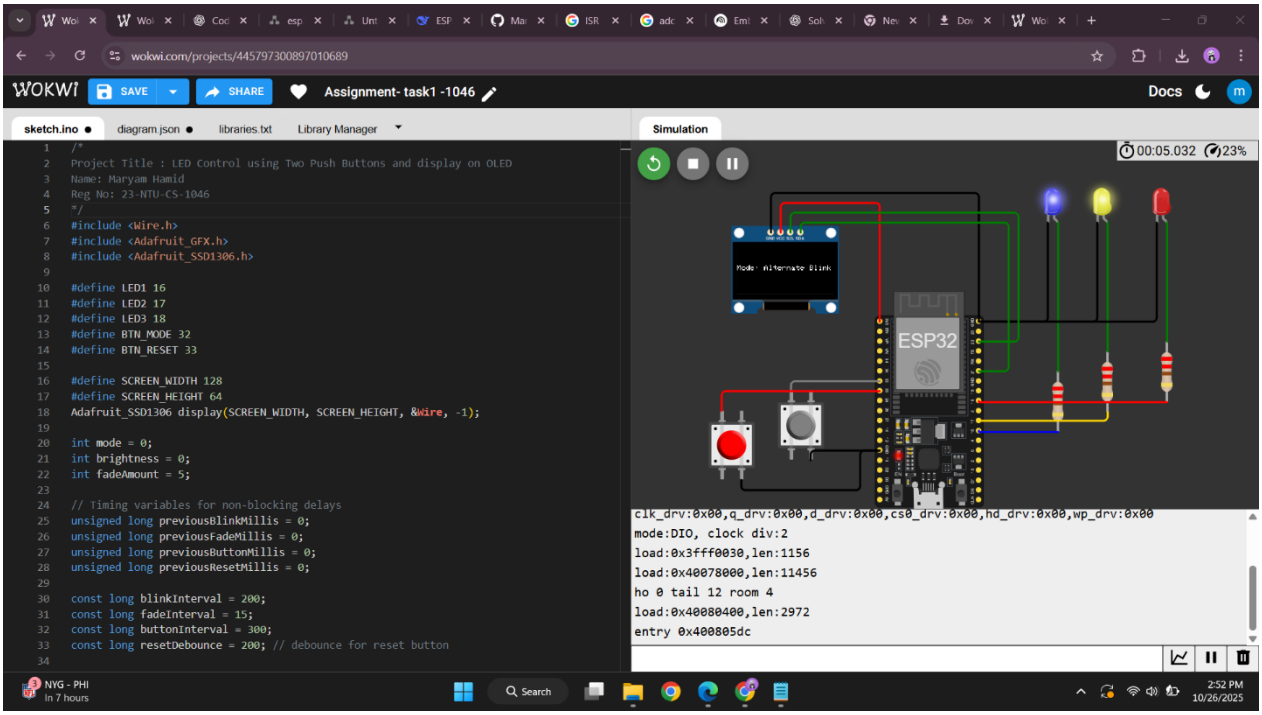
```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Core Concepts Implemented:

- **Non-blocking delays:** Using millis() for smooth multasking.
- **Debouncing:** Time-based filtering of buton input noise.
- **Soware PWM:** Manual generaon of duty cycle using ming logic.
- **OLED Communicaon:** Display updates via I²C (SDA=21, SCL=22).

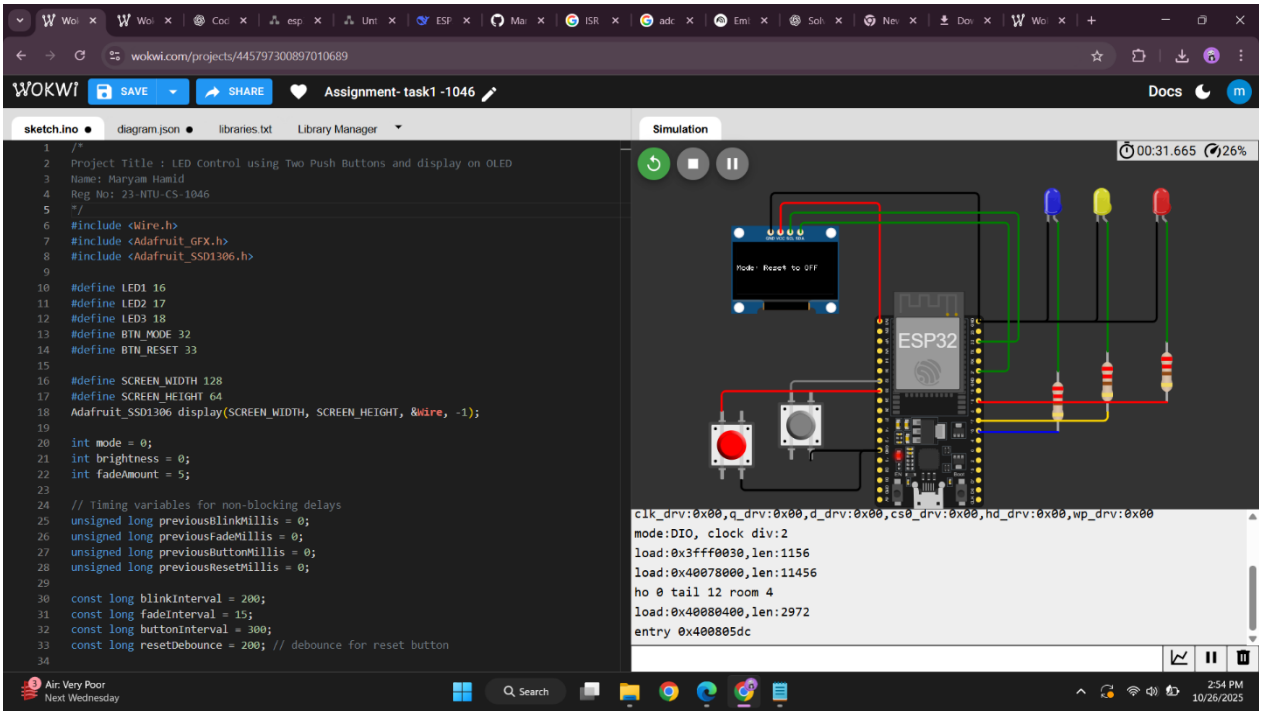
Screenshots of Different Modes of LEDS

ALTERNATE BLINK



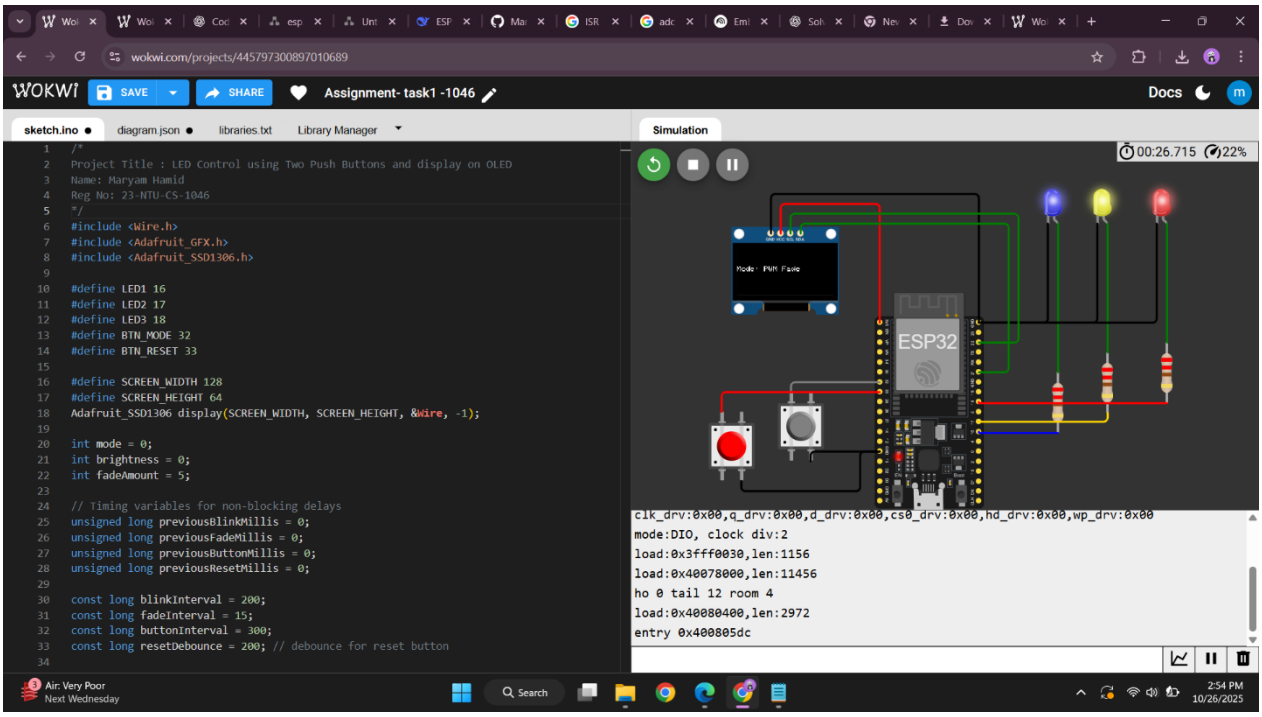

```
/*
Project Title : LED Control using Two Push Buttons and display on OLED
Name: Maryam Hamid
Reg No: 23-NTU-CS-1046
*/
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define LED1 16
#define LED2 17
#define LED3 18
#define BTN_MODE 32
#define BTN_RESET 33

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

int mode = 0;
int brightness = 0;
int fadeAmount = 5;

// Timing variables for non-blocking delays
unsigned long previousBlinkMillis = 0;
unsigned long previousFadeMillis = 0;
unsigned long previousButtonMillis = 0;
unsigned long previousResetMillis = 0;

const long blinkInterval = 200;
const long fadeInterval = 15;
const long buttonInterval = 300;
const long resetDebounce = 200; // debounce for reset button
```

RESET TO OFF



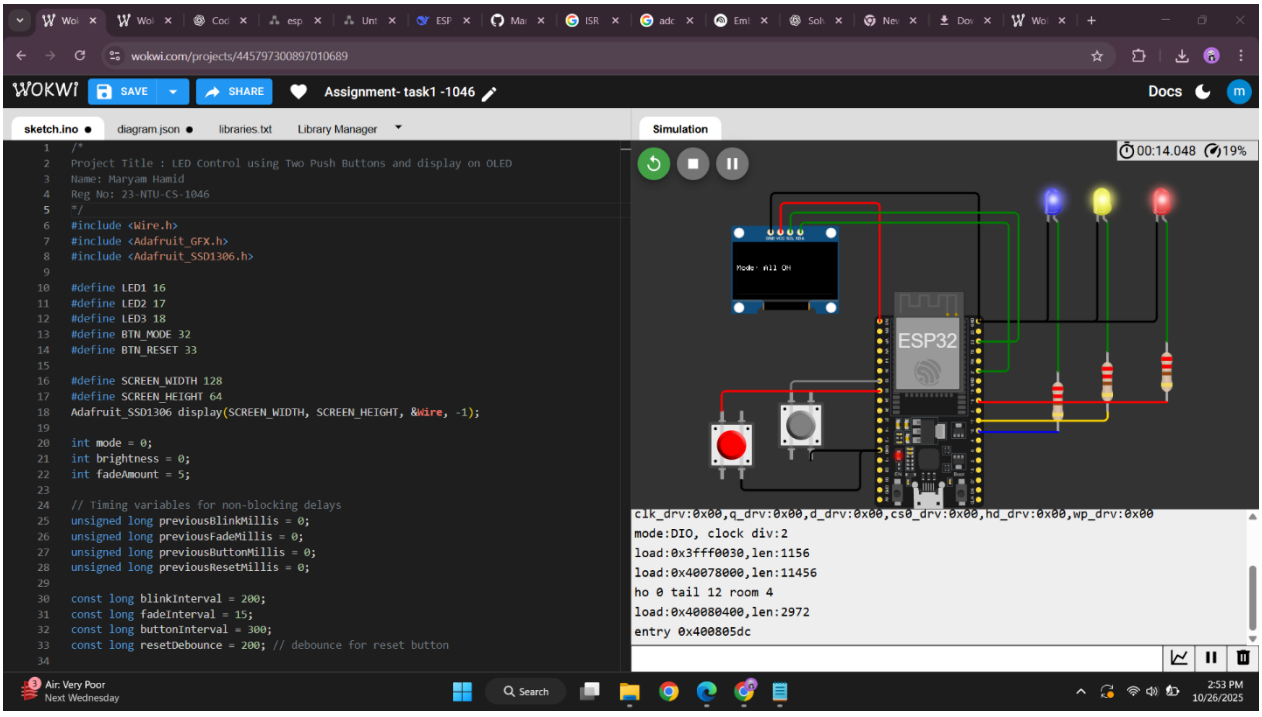

```
/*
Project Title : LED Control using Two Push Buttons and display on OLED
Name: Maryam Hamid
Reg No: 23-NTU-CS-1046
*/
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define LED1 16
#define LED2 17
#define LED3 18
#define BTN_MODE 32
#define BTN_RESET 33

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

int mode = 0;
int brightness = 0;
int fadeAmount = 5;

// Timing variables for non-blocking delays
unsigned long previousBlinkMillis = 0;
unsigned long previousFadeMillis = 0;
unsigned long previousButtonMillis = 0;
unsigned long previousResetMillis = 0;

const long blinkInterval = 200;
const long fadeInterval = 15;
const long buttonInterval = 300;
const long resetDebounce = 200; // debounce for reset button
```

PWM FADE




```
/*
Project Title : LED Control using Two Push Buttons and display on OLED
Name: Maryam Hamid
Reg No: 23-NTU-CS-1046
*/
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define LED1 16
#define LED2 17
#define LED3 18
#define BTN_MODE 32
#define BTN_RESET 33

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

int mode = 0;
int brightness = 0;
int fadeAmount = 5;

// Timing variables for non-blocking delays
unsigned long previousBlinkMillis = 0;
unsigned long previousFadeMillis = 0;
unsigned long previousButtonMillis = 0;
unsigned long previousResetMillis = 0;

const long blinkInterval = 200;
const long fadeInterval = 15;
const long buttonInterval = 300;
const long resetDebounce = 200; // debounce for reset button
```

ALL ON




```
/*
Project Title : LED Control using Two Push Buttons and display on OLED
Name: Maryam Hamid
Reg No: 23-NTU-CS-1046
*/
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define LED1 16
#define LED2 17
#define LED3 18
#define BTN_MODE 32
#define BTN_RESET 33

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

int mode = 0;
int brightness = 0;
int fadeAmount = 5;

// Timing variables for non-blocking delays
unsigned long previousBlinkMillis = 0;
unsigned long previousFadeMillis = 0;
unsigned long previousButtonMillis = 0;
unsigned long previousResetMillis = 0;

const long blinkInterval = 200;
const long fadeInterval = 15;
const long buttonInterval = 300;
const long resetDebounce = 200; // debounce for reset button
```