

National Textile University, Faisalabad



Department of Computer Science

| | |
|-------------------------|---------------------------------|
| Name: | Aleeha Aiman |
| Class: | Bs Computer Science (Section A) |
| Registration No: | 23-NTU-CS-1243 |
| Course Name: | Embedded IOT Systems |
| Submitted To: | <i>Sir Nasir Mahmood</i> |
| Submission Date: | 26 October ,2025 |

Assignment 1 -Question 3

IMPLEMENTATION

Contents

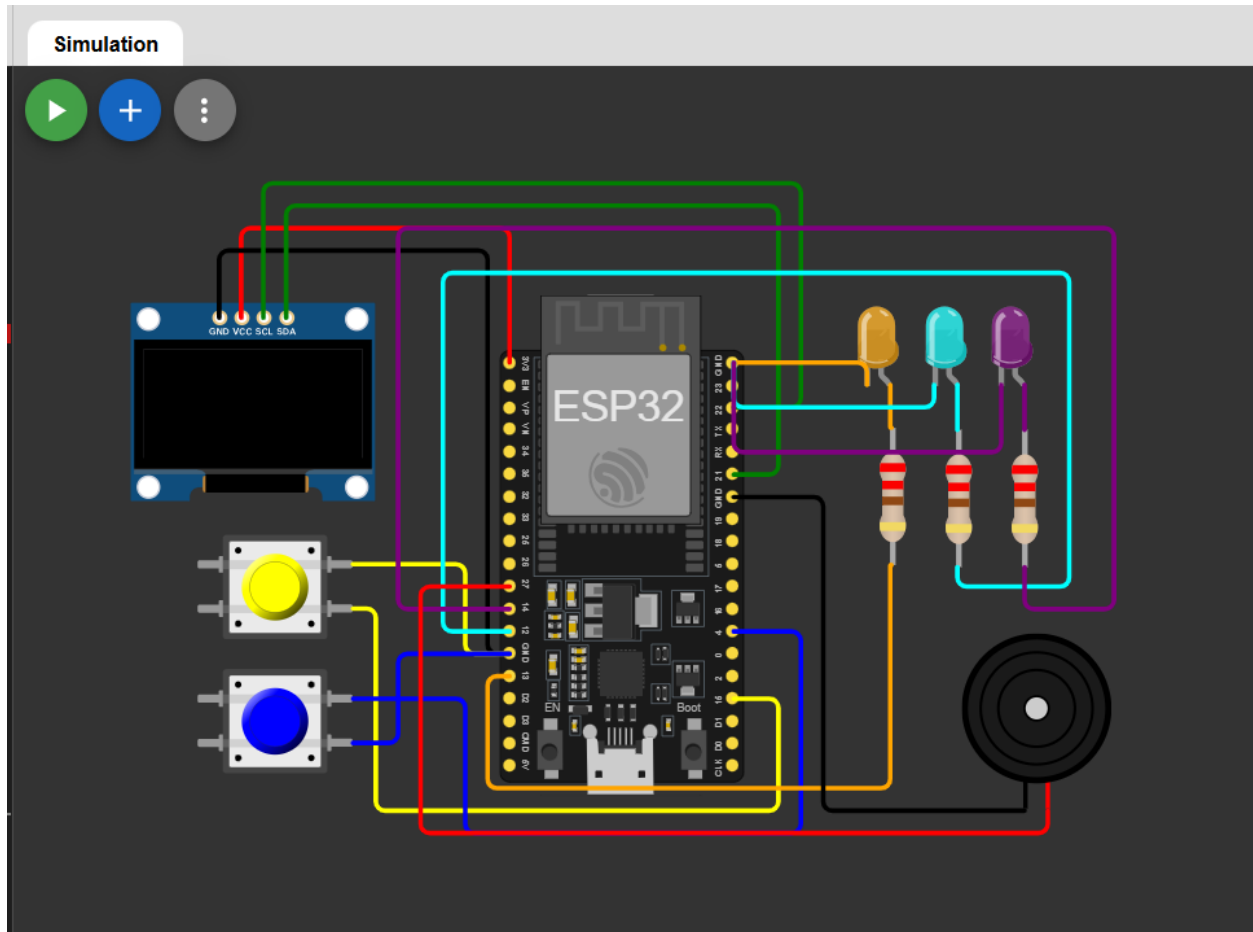
| | |
|--------------------------------|-----------|
| Circuit Diagram | 2 |
| Wokwi circuit | 3 |
| Hand Drawn Sketch | 4 |
| PIN MAP | 5 |
| Task A..... | 5 |
| CODE : | 6 |
| Output Screenshots..... | 8 |
| Task B..... | 11 |
| CODE : | 11 |
| Output Screenshots..... | 13 |

Circuit Diagram:

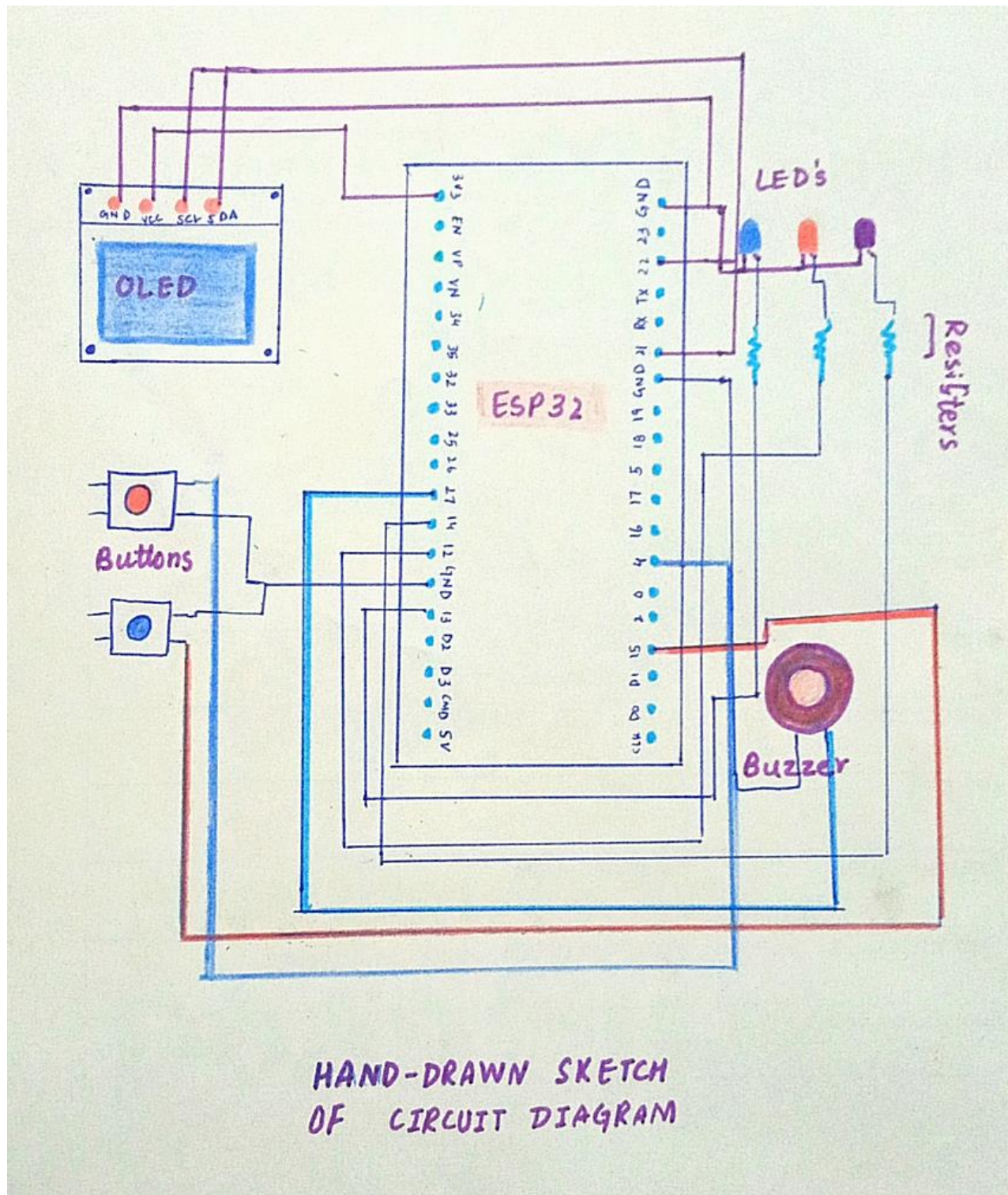
Design a Wokwi circuit and draw a neat hand-sketch including:

- **2 push buttons**
- **3 LEDs**
- **1 buzzer**
- **1 OLED**

Wokwi circuit



Hand Drawn Sketch



PIN MAP

| Component | Pin (GPIO) | GND |
|-----------------|------------|-----|
| Button (Yellow) | GPIO 15 | GND |
| LED 1 (Orange) | GPIO 13 | GND |
| LED 2 (Cyan) | GPIO 12 | GND |
| LED 3 (Purple) | GPIO 14 | GND |
| Buzzer | GPIO 27 | GND |
| OLED SDA | GPIO 21 | — |
| OLED SCL | GPIO 22 | — |

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.

CODE :

```
// Name : Aleeha Aiman
//Reg No: 23-NTU-CS-1243
//Class: Bs Cs 5th (A)
//Title : 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.

#include <Arduino.h>
#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);

#define BTN_MODE 15
#define BTN_RESET 4
#define LED1 13
#define LED2 12
#define LED3 14
#define BUZZER 27

const int PWMChannel = 0;
const int PWMFrequency = 5000; //I have given a higher frequency for smooth
                                //observation as it Turns ON/OFF 5000 times in 1
second
const int PWMResolution = 8; //it means duty cycle can be set from 0-255 i.e
                              //2^8, so at 100% brightness
                              //we need to set it to 255

int mode = 0;
unsigned long lastPress = 0; //I used unsigned as time is always positive
int fadeValue = 0;
int fadeDir = 1; //it is basically fade direction (1 for increasing, -1 for
decreasing)

unsigned long blinkTimer = 0; // Timer for blinking LEDs
bool blinkState = false;
```

```

void showMode(const char* msg) { //Print our current mode on OLED
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 20);
    display.print("Mode: ");
    display.println(msg);
    display.display();
}

void setup() {
    pinMode(BTN_MODE, INPUT_PULLUP);
    pinMode(BTN_RESET, INPUT_PULLUP);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(BUZZER, OUTPUT);

    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    showMode("OFF");

    ledcSetup(PWMChannel, PWMFrequency, PWMResolution);
    ledcAttachPin(LED3, PWMChannel); //i connected LED3 to PWM for FADE mode
}

void loop() {
    if (digitalRead(BTN_MODE) == LOW && millis() - lastPress > 300) { //used millis
for debouncing
        lastPress = millis();
        mode = (mode + 1) % 4; //4 modes: 0,1,2,3
        switch (mode) {
            case 0: showMode("OFF"); break;
            case 1: showMode("ALTERNATE BLINK"); break;
            case 2: showMode("ALL ON"); break;
            case 3: showMode("FADE"); break;
        }
    }

    if (digitalRead(BTN_RESET) == LOW) {
        mode = 0;
        showMode("OFF");
    }

    switch (mode) {
        case 0:

```

```

digitalWrite(LED1, LOW); //Turn OFF LEDs
digitalWrite(LED2, LOW);
ledcWrite(PWMChannel, 0);
break;

case 1:
    if (millis() - blinkTimer >= 300) { //After every 300 ms toggle the LEDs
        blinkTimer = millis();
        blinkState = !blinkState;
        digitalWrite(LED1, blinkState);
        digitalWrite(LED2, !blinkState);
    }
    break;

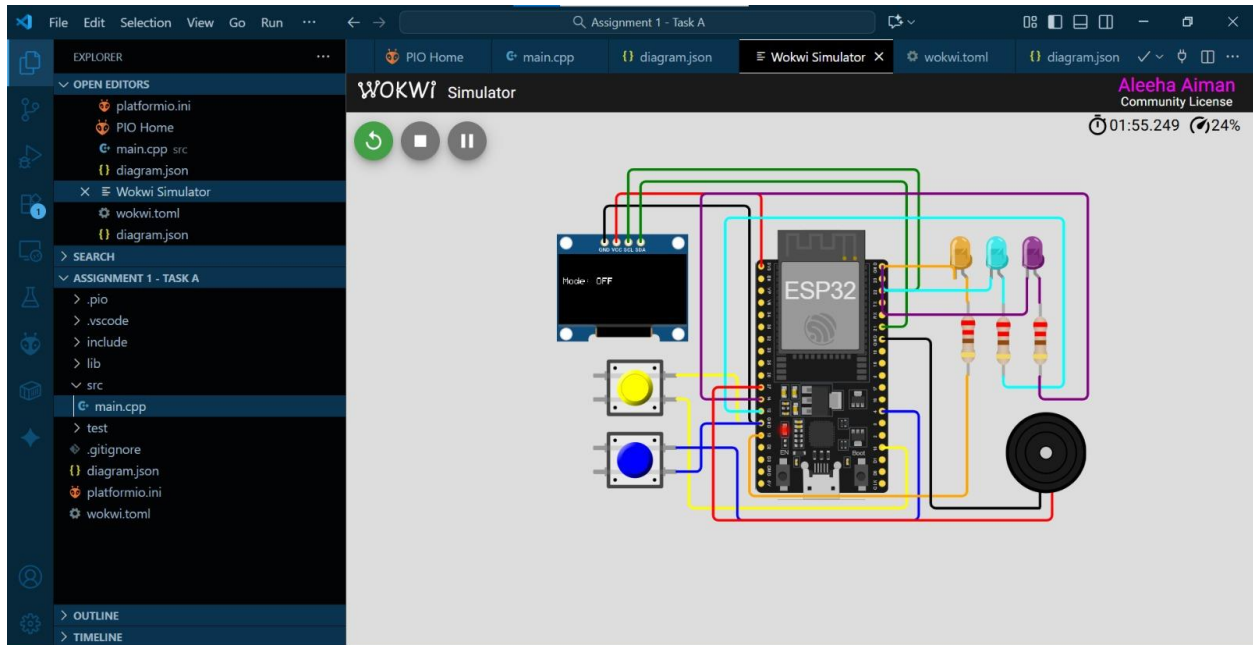
case 2:
    digitalWrite(LED1, HIGH); //Turn ON LEDs
    digitalWrite(LED2, HIGH);
    ledcWrite(PWMChannel, 255); //Full brightness
    break;

case 3:
    ledcWrite(PWMChannel, fadeValue); //Set brightness according to fadeValue
    fadeValue += fadeDir * 5; //Change fadeValue by 5 in each iteration
    if (fadeValue <= 0 || fadeValue >= 255) fadeDir = -fadeDir; //Reverse
direction like when we
    //will reach at zero brightness reverse the direction to increase
brightness
    delay(20);
    break;
}
}

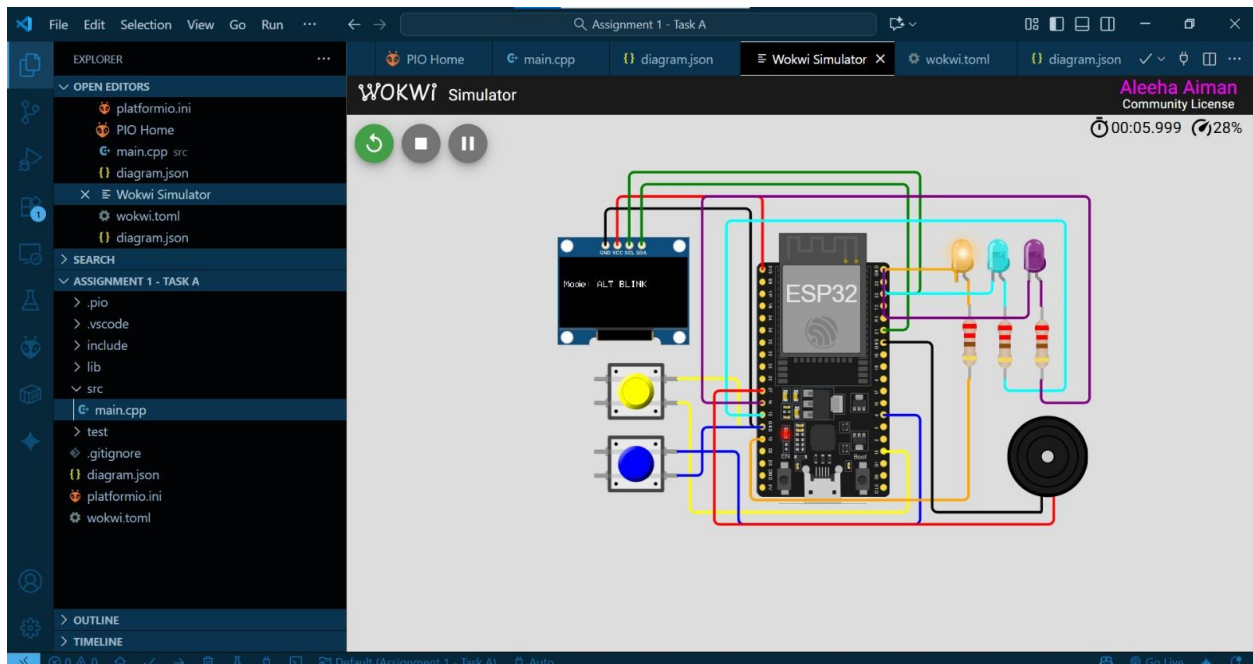
```

Output Screenshots

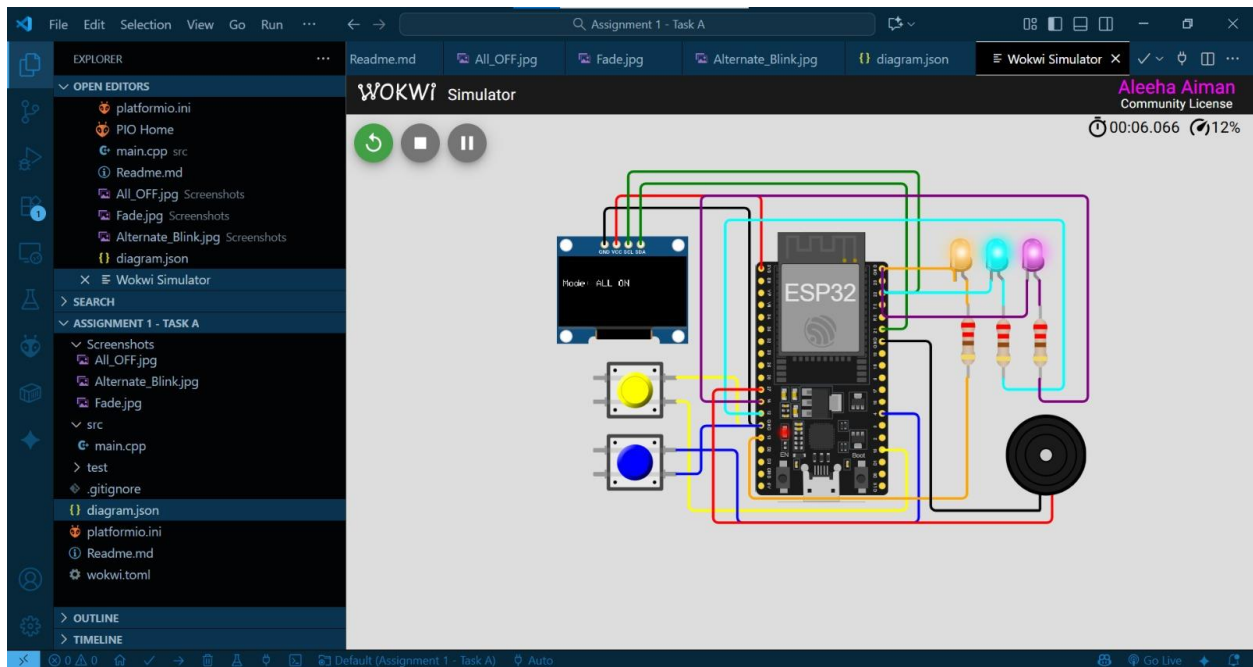
1. Both OFF



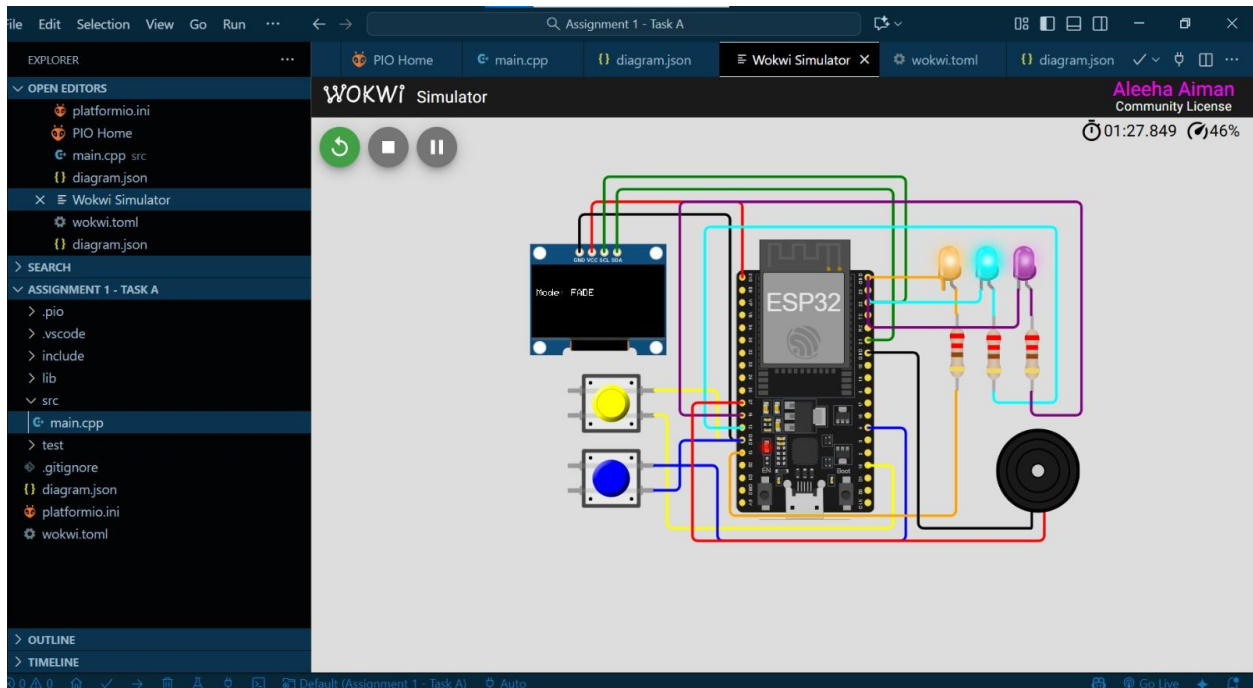
2. Alternate blink



3. ALL ON



4. PWM fade



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**

CODE :

```
//Title : 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

//Aleeha Aiman
//Regno: 23-NTU-CS-1243
//BsCs5th A

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

#define BTN1 15 // i have connected yellow button to pin 15
#define LED1 13
#define LED2 12
#define LED3 14
#define BUZZ 27

Adafruit_SSD1306 oled(128, 64, &Wire, -1);

unsigned long pressStart = 0; //it is for storing time in milis also i used
unsigned as time will be positive always
bool isPressed = false;
bool ledsOn = false;

void setup() {
  pinMode(BTN1, INPUT_PULLUP); //pin15 ie button 1 gets the input
```

```

pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(BUZZ, OUTPUT);

digitalWrite(LED1, LOW); //initially all led's and buzzer is low
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
digitalWrite(BUZZ, LOW);

oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);
oled.clearDisplay();
oled.setTextSize(1);
oled.setTextColor(SSD1306_WHITE);
oled.setCursor(0, 0);
oled.println("Ready...");
oled.display(); // just when then the code executes OLED displays "ready"
}

void loop() {
    int buttonState = digitalRead(BTN1);

    if (buttonState == LOW && !isPressed) {
        isPressed = true;
        pressStart = millis(); //when the button is pressed, this variables starts
stroing time in milis
    }

    if (buttonState == HIGH && isPressed) {
        unsigned long duration = millis() - pressStart; //it records the total time
when button is in pressed state
        isPressed = false;
        oled.clearDisplay();

        if (duration < 1500) {
            ledsOn = !ledsOn; //if button was ON for 1.5 sec then LEDs are turned ON
            digitalWrite(LED1, ledsOn);
            digitalWrite(LED2, ledsOn);
            digitalWrite(LED3, ledsOn);

            oled.setCursor(0, 0);
            oled.println("Short Press Detected");
            oled.setCursor(0, 16);
            oled.println(ledsOn ? "LEDs: ON" : "LEDs: OFF");
            oled.display();

```

```

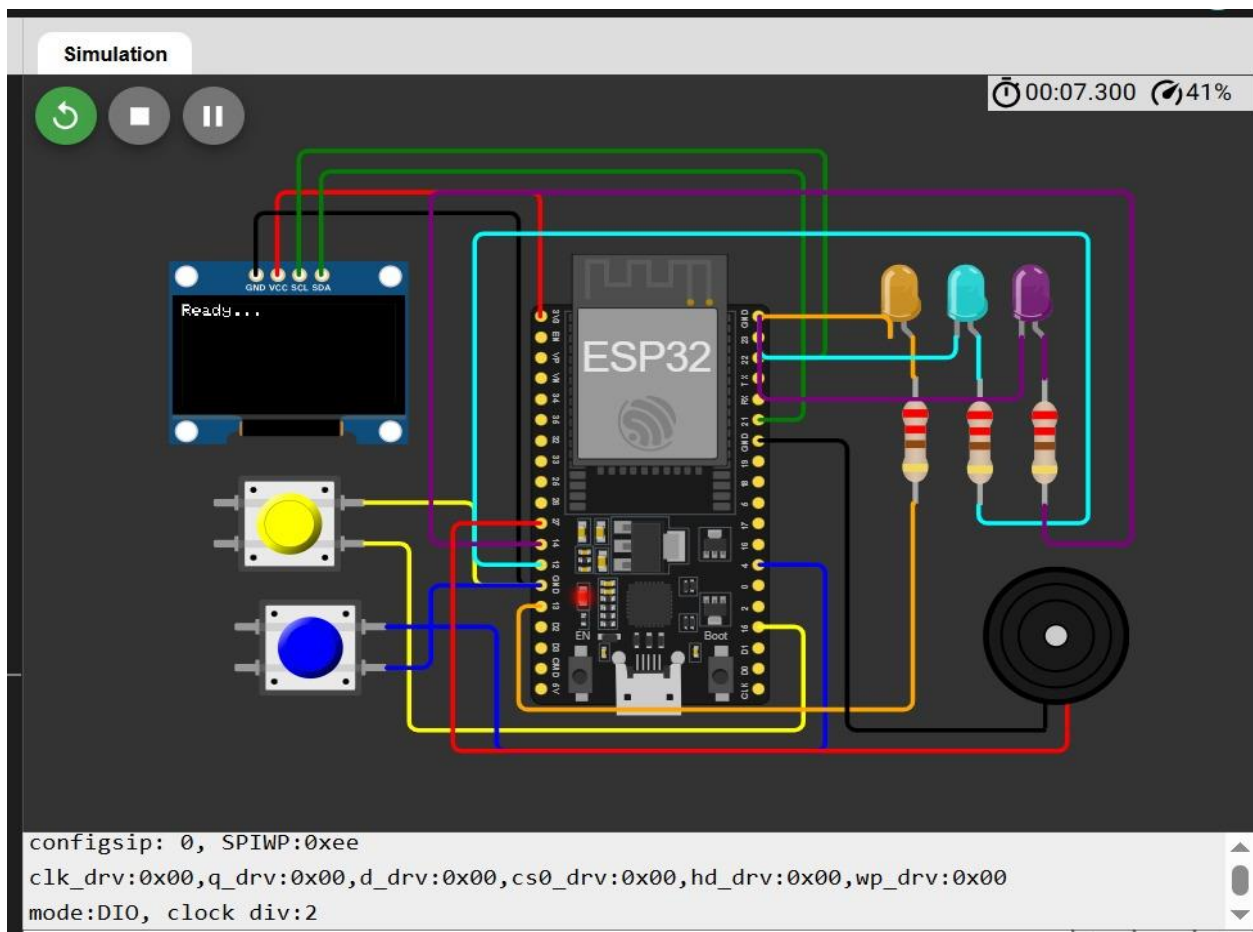
    } else {
        oled.setCursor(0, 0); //if we held it for more than 1.5 sec, buzzer plays
the sound
        oled.println("Long Press Detected");
        oled.setCursor(0, 16);
        oled.println("Playing buzzer...");
        oled.display();

        tone(BUZZ, 1000, 500); // sound is of 1000 Hz and it plays for half a sec
        delay(500);
        noTone(BUZZ); //Buzzer stops producing sound
    }
}
}

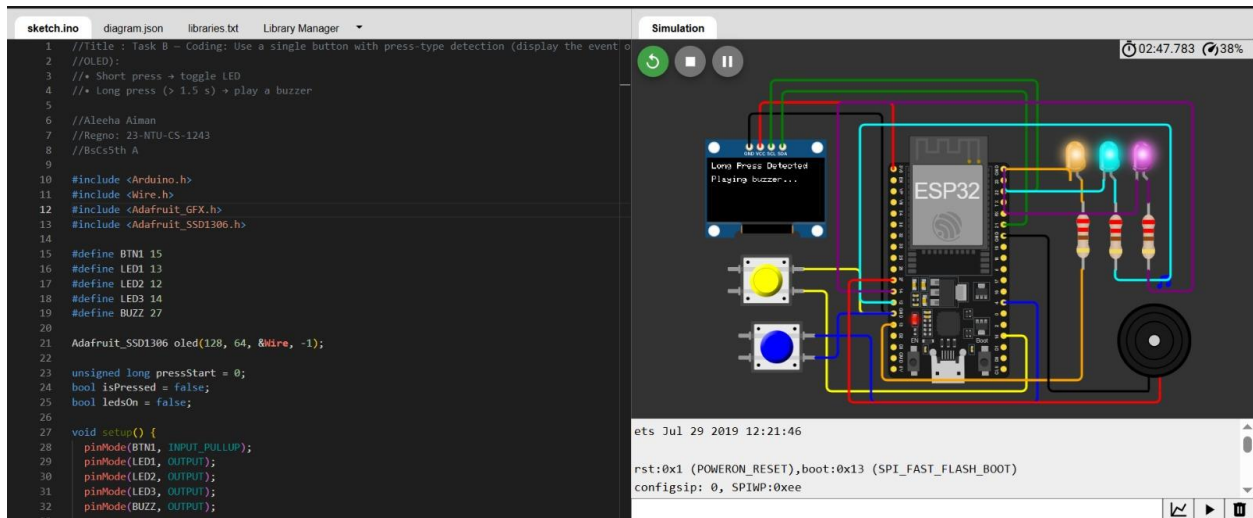
```

Output Screenshots

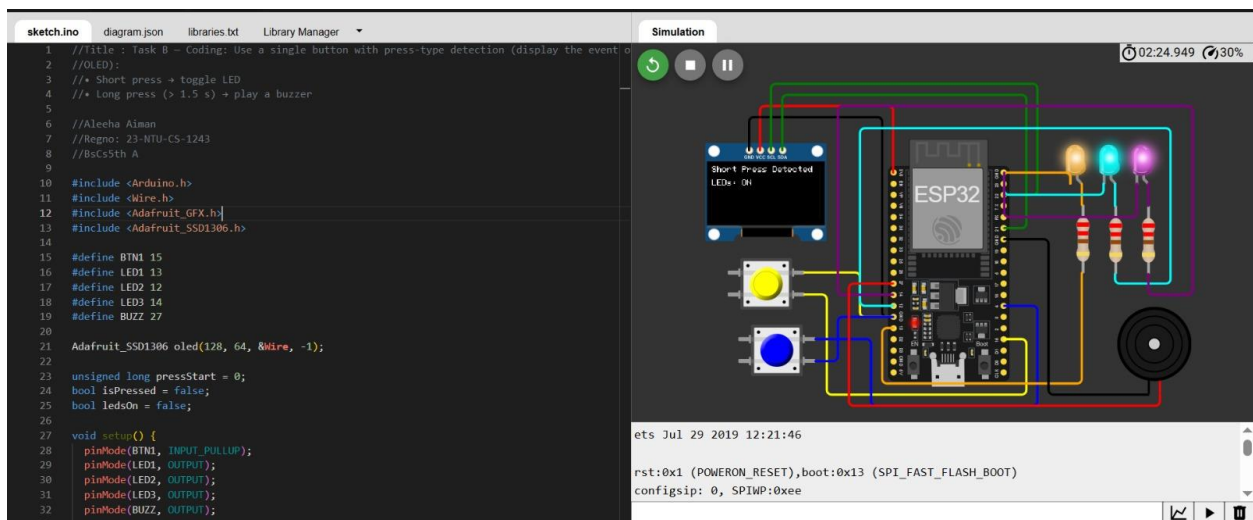
1. Ready:



2. Playing Buzzer -Long Press Detected



3. All LED ON -Short Press detected



4.LED OFF

WOKWI SAVE SHARE Assignemnt 1-TaskB Docs

sketch.ino diagram.json libraries.txt Library Manager

```
1 //Title : Task 8 - Coding: Use a single button with press-type detection (display the event o
2 //OLED);
3 //+ Short press + toggle LED
4 //+ Long press (> 1.5 s) + play a buzzer
5
6 //Aleeha Aiman
7 //Regno: 23-NTU-CS-1243
8 //BSc5th A
9
10 #include <Arduino.h>
11 #include <Wire.h>
12 #include <Adafruit_GFX.h>
13 #include <Adafruit_SSD1306.h>
14
15 #define BTN1 15
16 #define LED1 13
17 #define LED2 12
18 #define LED3 14
19 #define BUZZ 27
20
21 Adafruit_SSD1306 oled(128, 64, &Wire, -1);
22
23 unsigned long pressStart = 0;
24 bool isPressed = false;
25 bool ledsOn = false;
26
27 void setup() {
28   pinMode(BTN1, INPUT_PULLUP);
29   pinMode(LED1, OUTPUT);
30   pinMode(LED2, OUTPUT);
31   pinMode(LED3, OUTPUT);
32   pinMode(BUZZ, OUTPUT);
33 }
```

Simulation

03:10.249 40%

ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)

configip: 0, SPIWP:0xee