National Textile University, Faisalabad

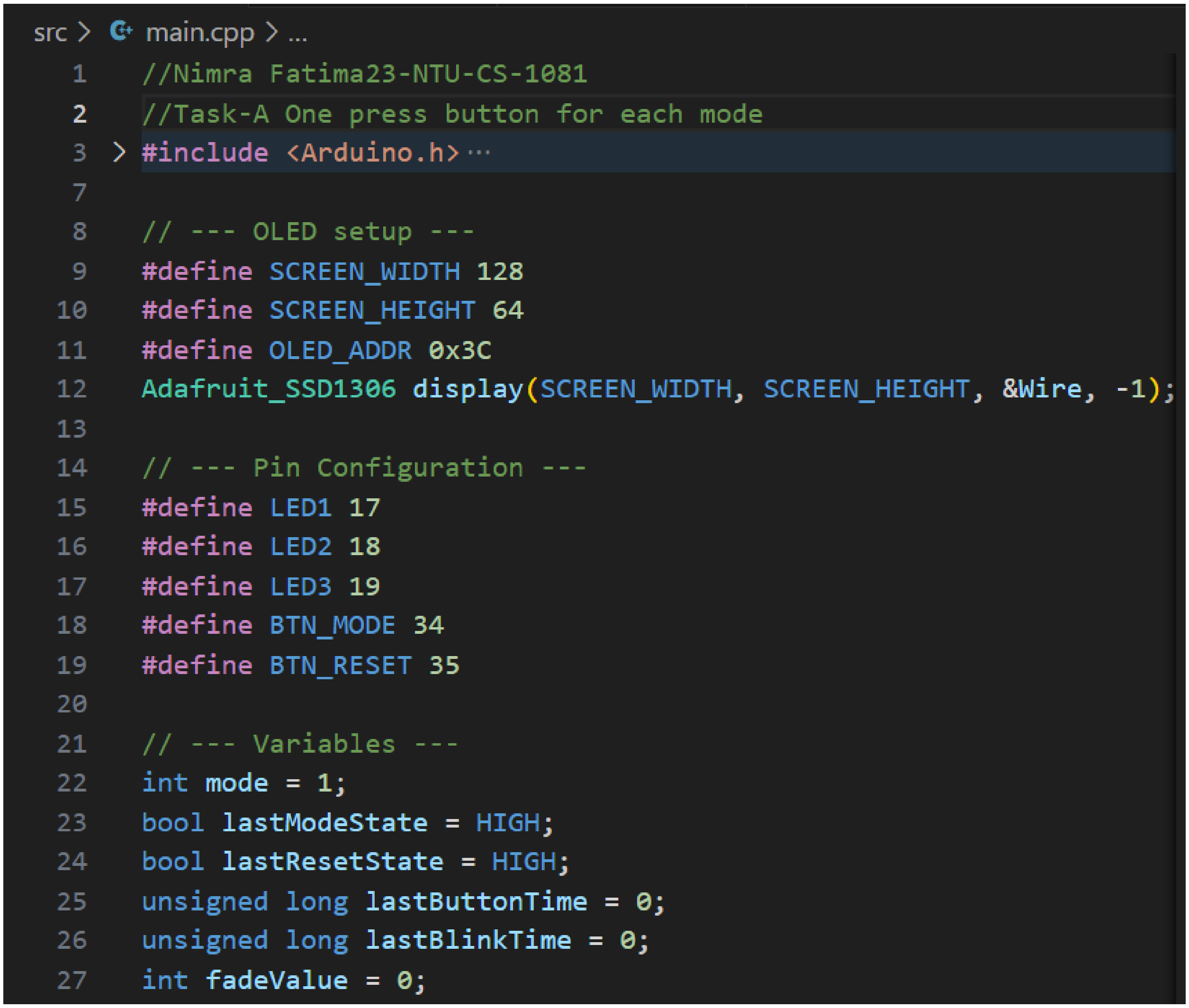
Department of Computer Science

Name	Nimra Fatima
Reg no	23-NTU-CS-1081
Section	BSCS 5th -B
Subject	Embedded & IoT Systems
Assignment	1-Task(A)
Submitted to	Sir Nasir

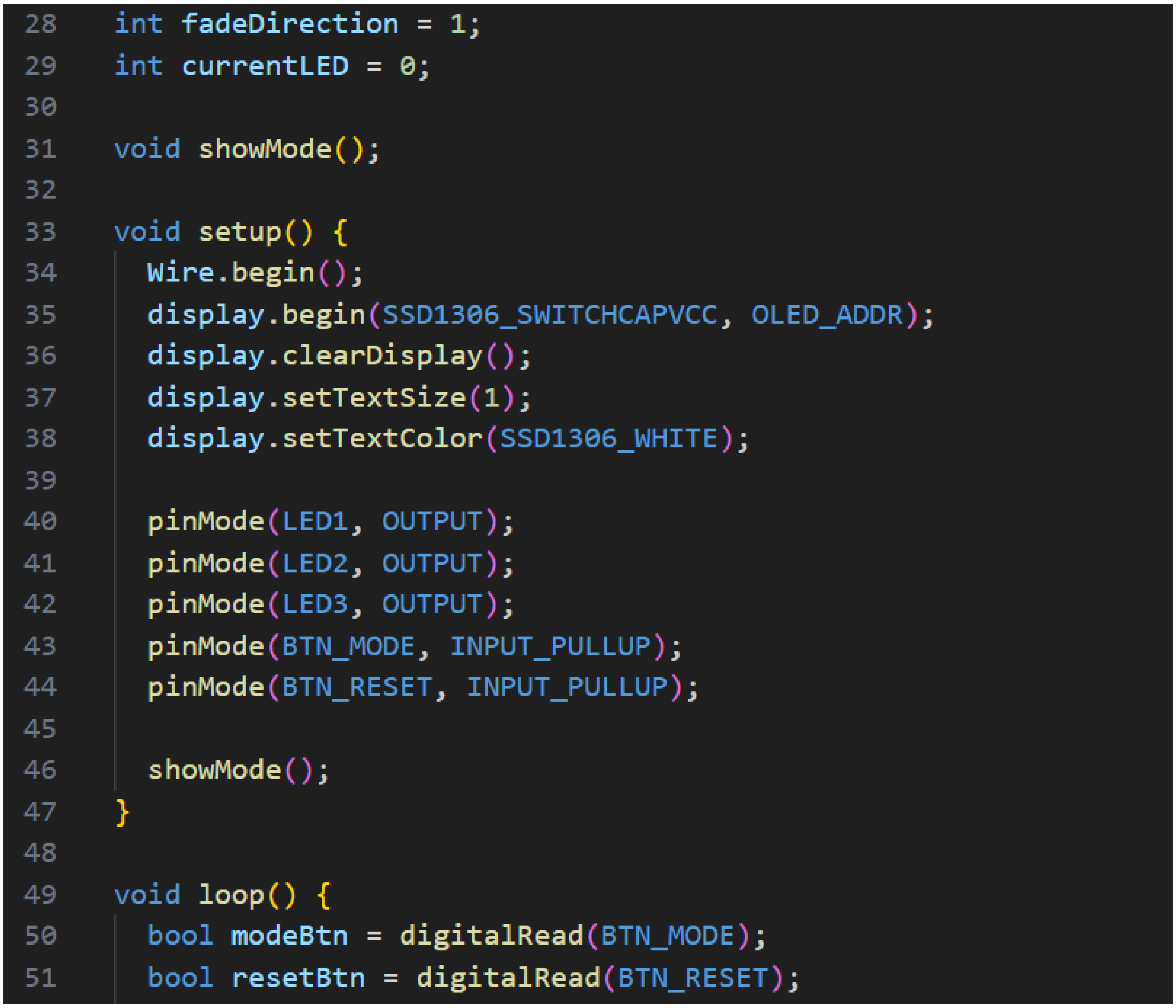
Task A

Circuit Diagram: Design a Wokwi circuit and draw a neat hand-sketch including: • 2 push buttons • 3 LEDs • 1 buzzer • 1 OLED

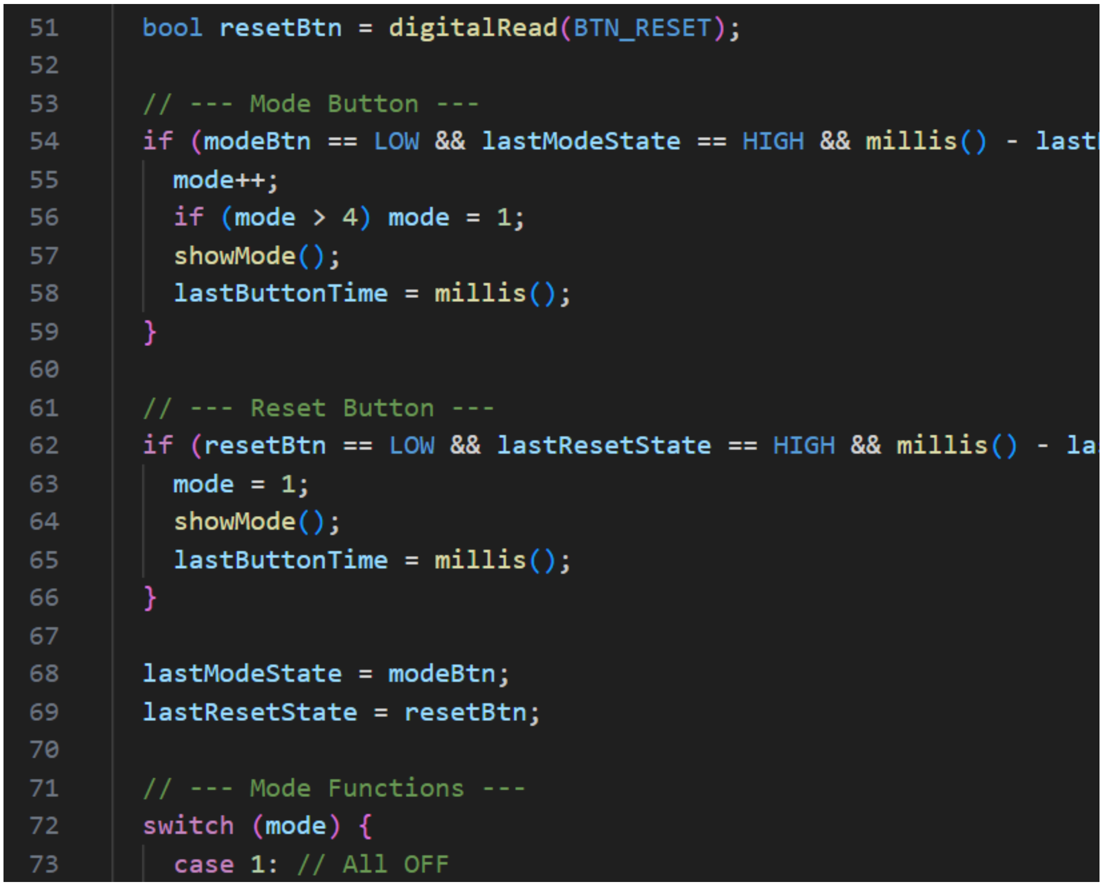
Task A — Coding: Use one button to cycle through LED modes (display the current state on the OLED): 1. Both OFF 2. Alternate blink 3. Both ON 4. PWM fade Use the second button to reset to OFF.



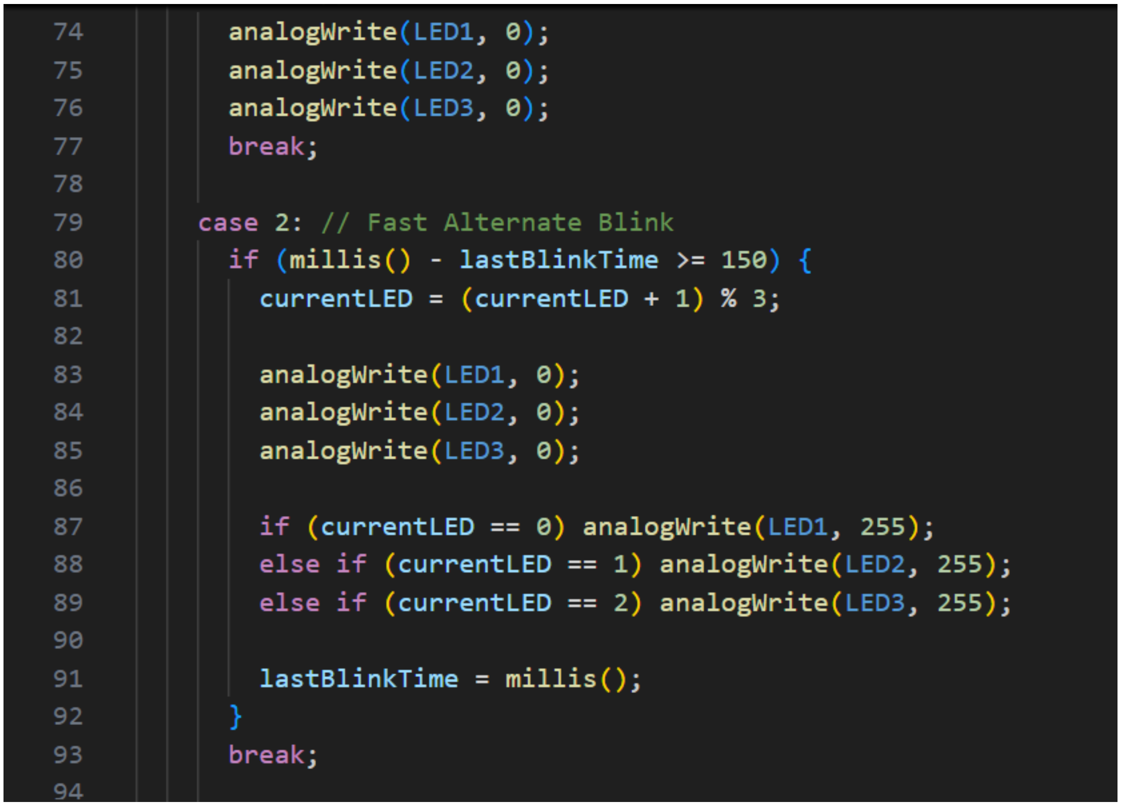
```
//Nimra Fatima23-NTU-CS-1081
//Task-A One press button for each mode
#include <Arduino.h>···

// --- OLED setup ---
#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 17
#define LED2 18
#define LED3 19
#define BTN_MODE 34
#define BTN_RESET 35

// --- Variables ---
int mode = 1;
bool lastModeState = HIGH;
bool lastResetState = HIGH;
unsigned long lastButtonTime = 0;
unsigned long lastBlinkTime = 0;
int fadeValue = 0;
```



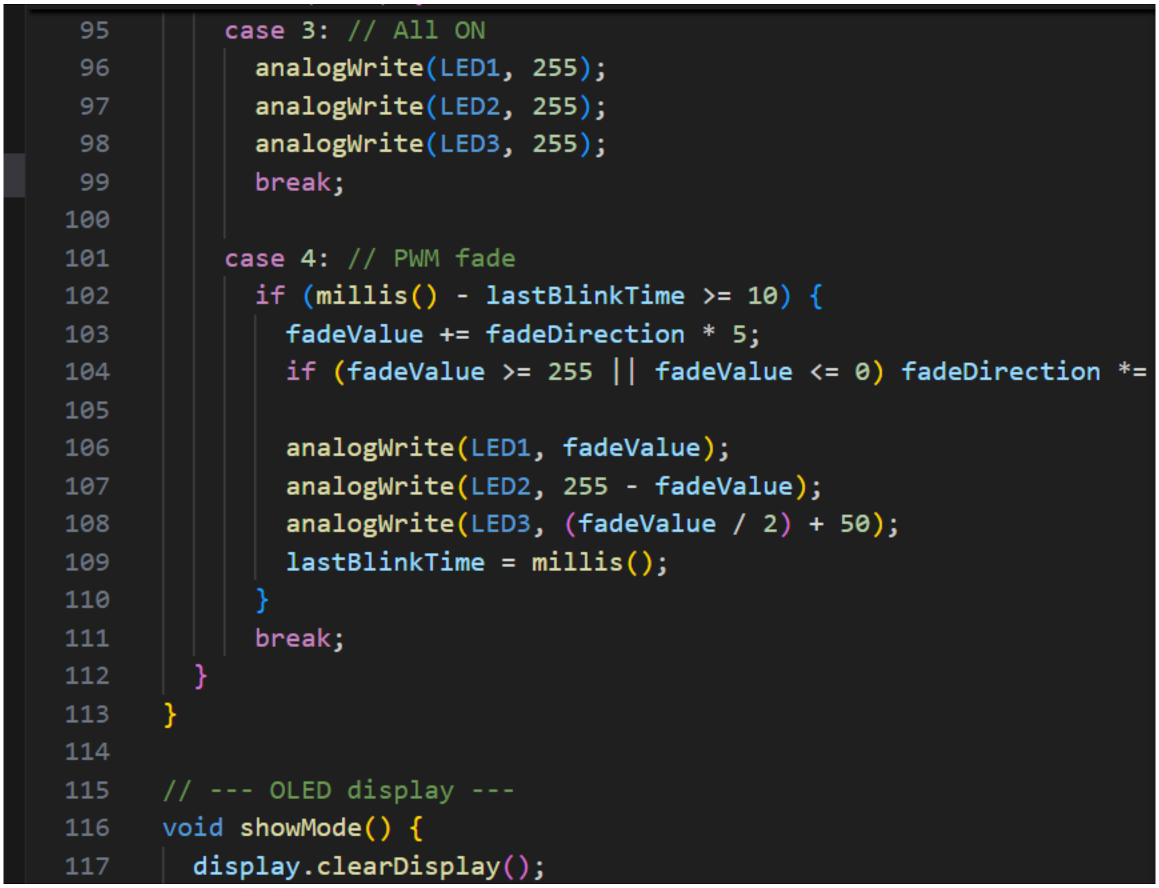
```
int fadeDirection = 1;
int currentLED = 0;

void showMode();

void setup() {
  Wire.begin();
  display.begin(SSD1306_SWITCHCAPVCC, OLED_ADDR);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(BTN_MODE, INPUT_PULLUP);
  pinMode(BTN_RESET, INPUT_PULLUP);

  showMode();
}

void loop() {
  bool modeBtn = digitalRead(BTN_MODE);
  bool resetBtn = digitalRead(BTN_RESET);
```



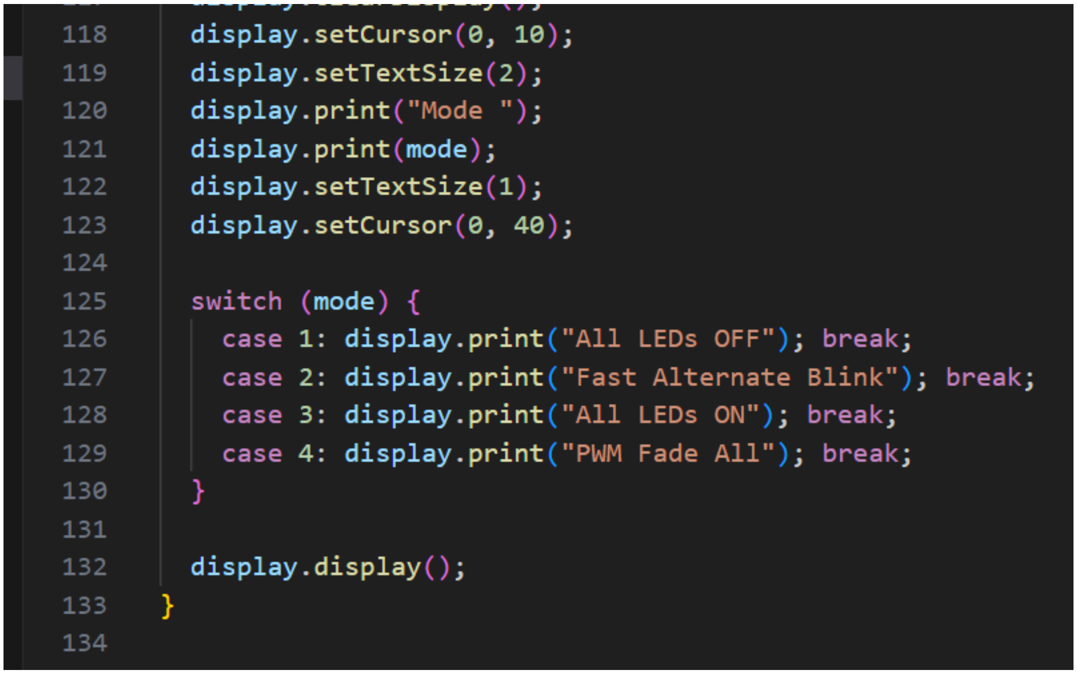
```
  bool resetBtn = digitalRead(BTN_RESET);

  // --- Mode Button ---
  if (modeBtn == LOW && lastModeState == HIGH && millis() - last
    mode++;
    if (mode > 4) mode = 1;
    showMode();
    lastButtonTime = millis();
  }

  // --- Reset Button ---
  if (resetBtn == LOW && lastResetState == HIGH && millis() - la
    mode = 1;
    showMode();
    lastButtonTime = millis();
  }

  lastModeState = modeBtn;
  lastResetState = resetBtn;

  // --- Mode Functions ---
  switch (mode) {
    case 1: // All OFF
```



```
      analogWrite(LED1, 0);
      analogWrite(LED2, 0);
      analogWrite(LED3, 0);
      break;

    case 2: // Fast Alternate Blink
      if (millis() - lastBlinkTime >= 150) {
        currentLED = (currentLED + 1) % 3;

        analogWrite(LED1, 0);
        analogWrite(LED2, 0);
        analogWrite(LED3, 0);

        if (currentLED == 0) analogWrite(LED1, 255);
        else if (currentLED == 1) analogWrite(LED2, 255);
        else if (currentLED == 2) analogWrite(LED3, 255);

        lastBlinkTime = millis();
      }
      break;

```



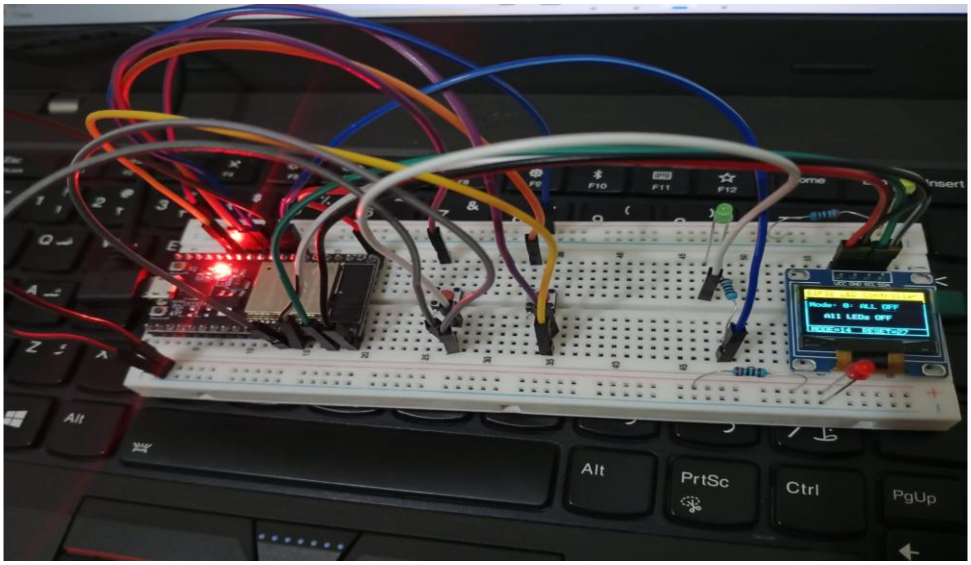
```
    case 3: // All ON
      analogWrite(LED1, 255);
      analogWrite(LED2, 255);
      analogWrite(LED3, 255);
      break;

    case 4: // PWM fade
      if (millis() - lastBlinkTime >= 10) {
        fadeValue += fadeDirection * 5;
        if (fadeValue >= 255 || fadeValue <= 0) fadeDirection *=

        analogWrite(LED1, fadeValue);
        analogWrite(LED2, 255 - fadeValue);
        analogWrite(LED3, (fadeValue / 2) + 50);
        lastBlinkTime = millis();
      }
      break;
  }
}

// --- OLED display ---
void showMode() {
  display.clearDisplay();
```



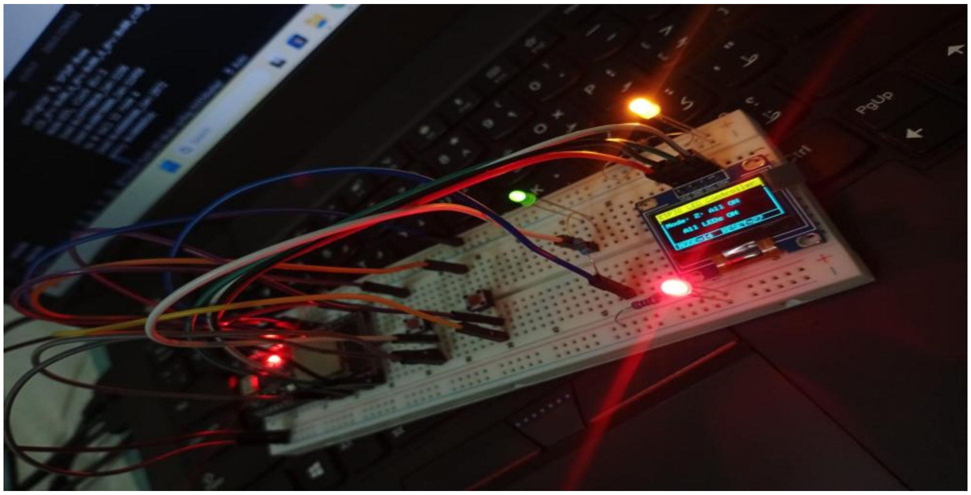
```
  display.setCursor(0, 10);
  display.setTextSize(2);
  display.print("Mode ");
  display.print(mode);
  display.setTextSize(1);
  display.setCursor(0, 40);

  switch (mode) {
    case 1: display.print("All LEDs OFF"); break;
    case 2: display.print("Fast Alternate Blink"); break;
    case 3: display.print("All LEDs ON"); break;
    case 4: display.print("PWM Fade All"); break;
  }

  display.display();
}
```

Output:

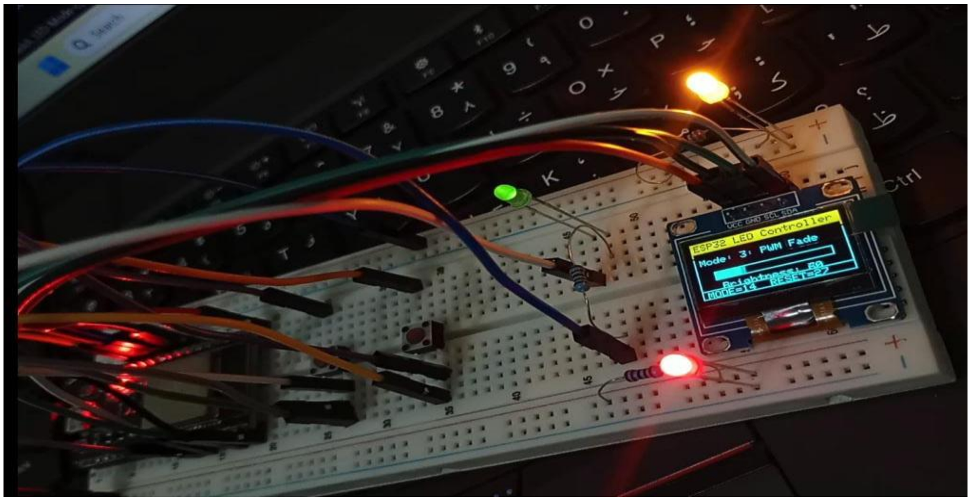
Both are OFF:



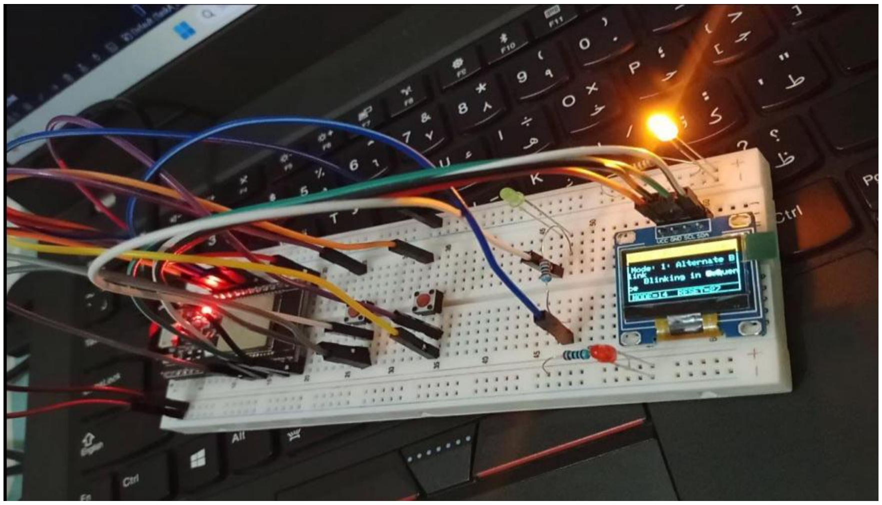
Both are ON:



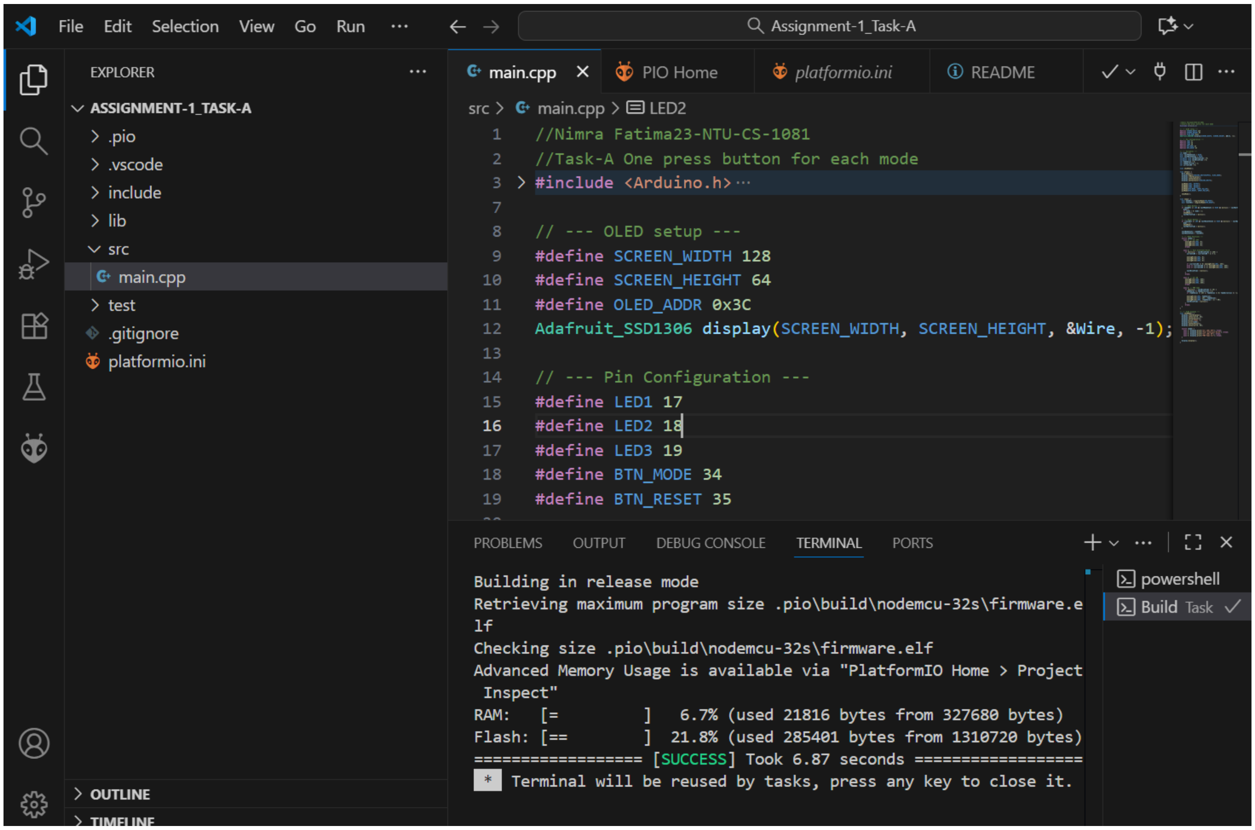
PWM Fade:



Alternate blink:



Success output:



```cpp
//Nimra Fatima23-NTU-CS-1081
//Task-A One press button for each mode
#include <Arduino.h>

// --- OLED setup ---
#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 17
#define LED2 18
#define LED3 19
#define BTN_MODE 34
#define BTN_RESET 35
```

```
Building in release mode
Retrieving maximum program size .pio\build\nodemcu-32s\firmware.e
lf
Checking size .pio\build\nodemcu-32s\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project
 Inspect"
RAM:   [=         ]   6.7% (used 21816 bytes from 327680 bytes)
Flash: [==        ]  21.8% (used 285401 bytes from 1310720 bytes)
================== [SUCCESS] Took 6.87 seconds ==================
 *  Terminal will be reused by tasks, press any key to close it.
```

Hand Written Code:

Task A: Assignment # 1

"One button to cycle through LED nodes"

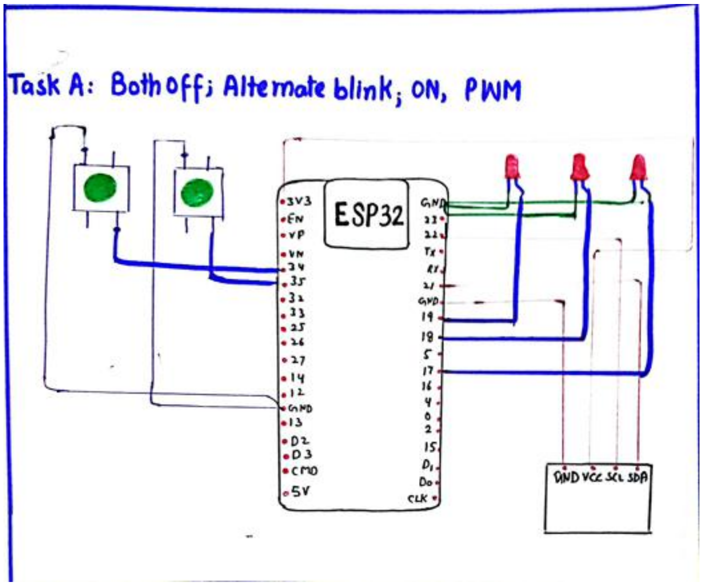
```
#include <Arduino.h>
#include <wire.h>
#include <Adafruit_GFX.h>
#include < Adafruit_SSD1306.h>

   //----OLED setup ----
#define SCREEN_Width 128
# define SCREE_Height 64
# define OLED_ADDR 0X3C
Adafruit_SSD1306 display (SCREEN_Width ,
        SCREEN_Height, & Wire, -1);
      // Pin configuration
#define Button1 34
#define Button2 64
# define OLED_ADDR 0X3C
# define Red-LED 14
# define Blue_LED 18
# define purple_LED 17
       // Variables
int mode. = 1 ;
bool lastButton1 = HIGH;
bool lastButton2 = HIGH;
unsigned long last ButtonTime =0;

Unsigned long lastBlink Time -0;
int blinkstate =0 ;
```

```
int fadevalu = 0;
int fadedirection = 1;

Void showMode ();

Void setap () {

 Serial . begin (115200);
 Wire.begin ( ) ;
 if (! display. begin (F("OLED failed!"));
  While (true);
 }
 display. clearDisplay();
        // OLED display
PinMode (RED_LED, output) ;
Pin Mode (BLUE_LED, output) ;
Pin Mode (Purple_LED, output);
PinMode (Button1 , input_pullup);
pinMode (Button2, input_pullup);

Show Mode ();

}
} Void loop () {
 bool buttonstate = digitalRead (Button1);
 bool buttonstate = digitalRead (Button2);
```

```
if (buttonstate == low && button == High &&
      mills () - lastbuttonTime >300 ) {
 mode = 1 ;
 Show Mode () ;
 lastButton Time = millis ();
 }
          // Mode functions.
Switch(Mode) {
 Case 1 :            // ALL OFF
 Analoge write (Red_LED, 0);
 Analoge Write (Blue-LED ,0);
 Analoge Write (PURPLE_LED ,0 ),
 break;
             // All ON with fast Alternate blink
Case 2:
if (millis () - lastBlinkTime >=500){
 blinkstate = ! blinkstate;
 anadogue write ( )
 lastblinkTime = millis ();
 }
 break;
           // All ON
Case 3:
if (millis () -last Blink Time >=10)
 Analogwrite (LED , 255 ),

 break ;
```

```
case 4:          //PWM display
 if (millis () - lastblink >= 10)}

 fadevalue += fadedirection *5
  if (fadevalue >=255 || fadeValue <=0)
   fadeDirection *= -1 ;

 AnalogWrite (RED-LED, Fadevalue)
 AnalogWrite (BLUE_LED, 255 - fadevalue);
 AnalogWrite (Purple_LED, Fadevalue/2 +50);

 lastBlink = millis () ;
 } Break ; }
 }
Void showMode (){
          //OLED display
display.clearDisplay () ;
display.setcursor (0,10);
display. TextSize (2 );
display . display () ;
switch(Mode) {
 case 1 : display.print (" All LED'S OFF);break;
 case 2 : display.print (" All blink Alternate); break;
 case 3 : display.print (" All LED'S ON "); break;
 case 4 : display.print ( All PWM Fade") ; break; }
 display.display () ;
 }
```

Hand Written Diagram:


Task A: Both off; Alternate blink; ON, PWM
ESP32
3V3
EN
VP
VN
34
35
32
33
25
26
27
14
12
GND
13
D2
D3
CMD
5V
GND
23
22
TX
RX
21
GND
19
18
5
17
16
4
0
2
15
D1
D0
CLK
GND
VCC
SCL
SDA