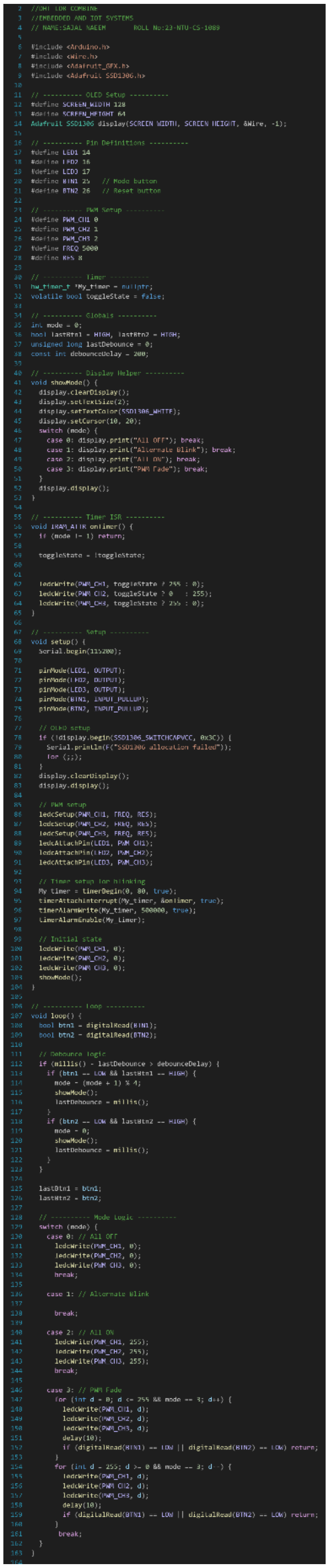
National Textile University, Faisalabad

Department of Computer Science

Name:	SAJAL NAEEM
Class:	BSCS 5TH B
Registration No:	23-NTU-CS-1089
Assignment:	1ST
Course Name:	IOT AND EMBEDDED SYSTEMS
Submitted To:	SIR NASIR MAHMAOOD
Submission Date:	26TH OCTOBER 2025

Code Screen Shot:

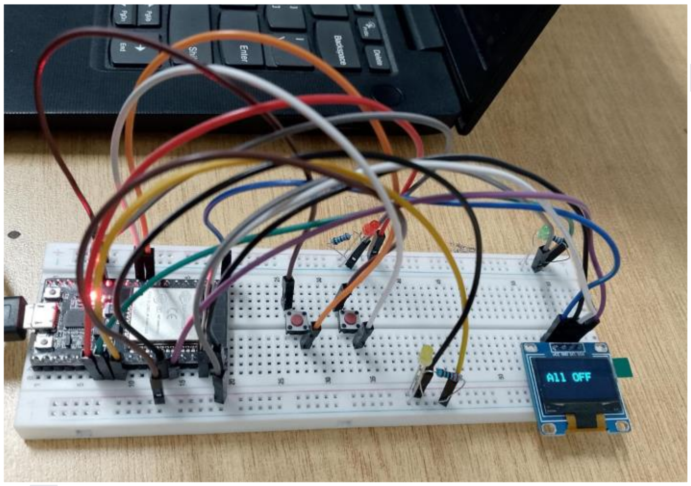


```
//DHT LDR COMBINE
//EMBEDDED AND IOT SYSTEMS
// NAME:SAJAL NAEEM       ROLL No:23-NTU-CS-1089

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

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

// ---------- Pin Definitions ----------
#define LED1 14
#define LED2 16
#define LED3 17
#define BTN1 25   // Mode button
#define BTN2 26   // Reset button

// ---------- PWM Setup ----------
#define PWM_CH1 0
#define PWM_CH2 1
#define PWM_CH3 2
#define FREQ 5000
#define RES 8

// ---------- Timer ----------
hw_timer_t *My_timer = nullptr;
volatile bool toggleState = false;

// ---------- Globals ----------
int mode = 0;
bool lastBtn1 = HIGH, lastBtn2 = HIGH;
unsigned long lastDebounce = 0;
const int debounceDelay = 200;

// ---------- Display Helper ----------
void showMode() {
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(10, 20);
  switch (mode) {
    case 0: display.print("All OFF"); break;
    case 1: display.print("Alternate Blink"); break;
    case 2: display.print("All ON"); break;
    case 3: display.print("PWM Fade"); break;
  }
  display.display();
}

// ---------- Timer ISR ----------
void IRAM_ATTR onTimer() {
  if (mode != 1) return;

  toggleState = !toggleState;


  ledcWrite(PWM_CH1, toggleState ? 255 : 0);
  ledcWrite(PWM_CH2, toggleState ? 0   : 255);
  ledcWrite(PWM_CH3, toggleState ? 255 : 0);
}

// ---------- Setup ----------
void setup() {
  Serial.begin(115200);

  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(BTN1, INPUT_PULLUP);
  pinMode(BTN2, INPUT_PULLUP);

  // OLED setup
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }
  display.clearDisplay();
  display.display();

  // PWM setup
  ledcSetup(PWM_CH1, FREQ, RES);
  ledcSetup(PWM_CH2, FREQ, RES);
  ledcSetup(PWM_CH3, FREQ, RES);
  ledcAttachPin(LED1, PWM_CH1);
  ledcAttachPin(LED2, PWM_CH2);
  ledcAttachPin(LED3, PWM_CH3);

  // Timer setup for blinking
  My_timer = timerBegin(0, 80, true);
  timerAttachInterrupt(My_timer, &onTimer, true);
  timerAlarmWrite(My_timer, 500000, true);
  timerAlarmEnable(My_timer);

  // Initial state
  ledcWrite(PWM_CH1, 0);
  ledcWrite(PWM_CH2, 0);
  ledcWrite(PWM_CH3, 0);
  showMode();
}

// ---------- Loop ----------
void loop() {
  bool btn1 = digitalRead(BTN1);
  bool btn2 = digitalRead(BTN2);

  // Debounce logic
  if (millis() - lastDebounce > debounceDelay) {
    if (btn1 == LOW && lastBtn1 == HIGH) {
      mode = (mode + 1) % 4;
      showMode();
      lastDebounce = millis();
    }
    if (btn2 == LOW && lastBtn2 == HIGH) {
      mode = 0;
      showMode();
      lastDebounce = millis();
    }
  }

  lastBtn1 = btn1;
  lastBtn2 = btn2;

  // ---------- Mode Logic ----------
  switch (mode) {
    case 0: // All OFF
      ledcWrite(PWM_CH1, 0);
      ledcWrite(PWM_CH2, 0);
      ledcWrite(PWM_CH3, 0);
      break;

    case 1: // Alternate Blink

      break;

    case 2: // All ON
      ledcWrite(PWM_CH1, 255);
      ledcWrite(PWM_CH2, 255);
      ledcWrite(PWM_CH3, 255);
      break;

    case 3: // PWM Fade
      for (int d = 0; d <= 255 && mode == 3; d++) {
        ledcWrite(PWM_CH1, d);
        ledcWrite(PWM_CH2, d);
        ledcWrite(PWM_CH3, d);
        delay(10);
        if (digitalRead(BTN1) == LOW || digitalRead(BTN2) == LOW) return;
      }
      for (int d = 255; d >= 0 && mode == 3; d--) {
        ledcWrite(PWM_CH1, d);
        ledcWrite(PWM_CH2, d);
        ledcWrite(PWM_CH3, d);
        delay(10);
        if (digitalRead(BTN1) == LOW || digitalRead(BTN2) == LOW) return;
      }
       break;
  }
}
```

Code Explanation

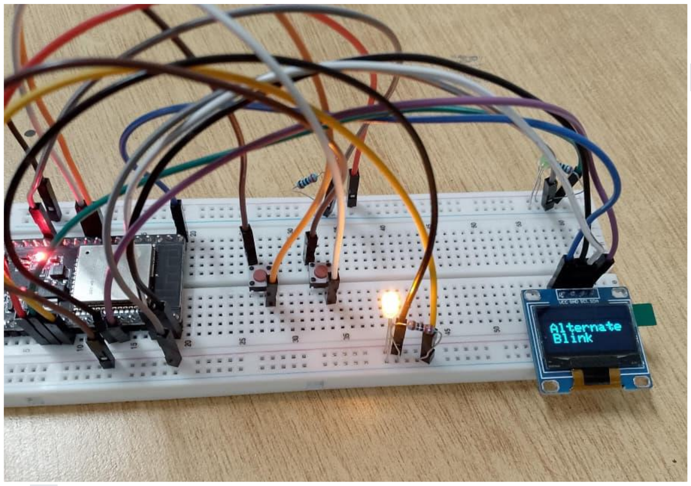
In this code we have integrated three LED and two push buttons, and we have applied logic to switch between 4 different modes. The first button is used to switch between modes while the second button is used to reset the mode. in 1st mode all the led will be off then in second mode the LED will blink in 2 vs 1 pattern. in third mode the led will be on for still time and in 4th mode the LED will have an PWM fade.

Output Pictures:



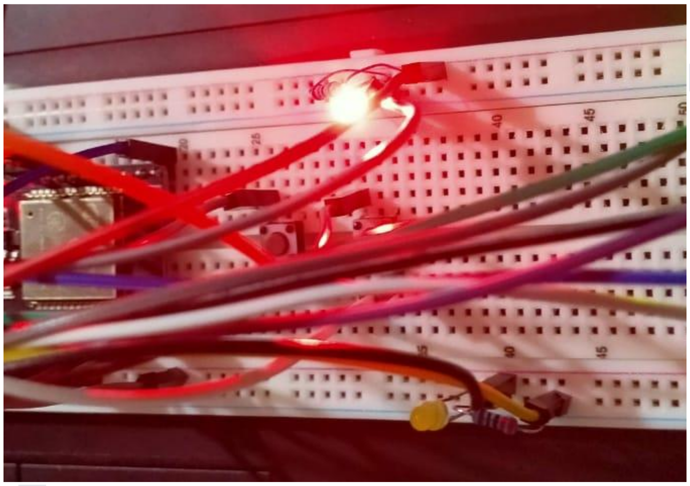
All LED are OFF

Mode 1



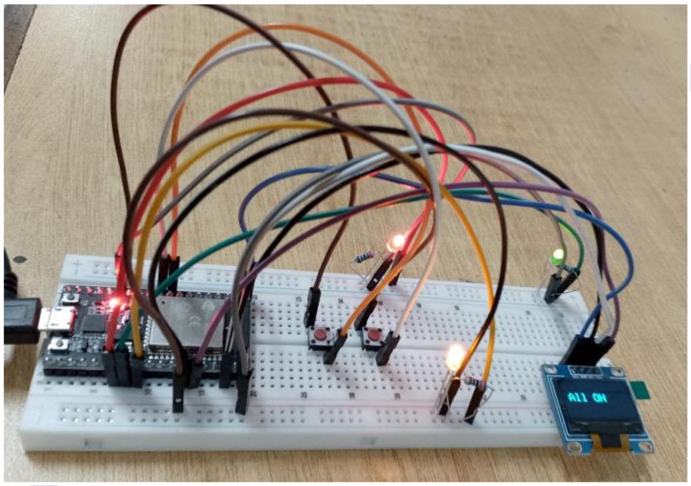
ALternate Blink

Mode 2



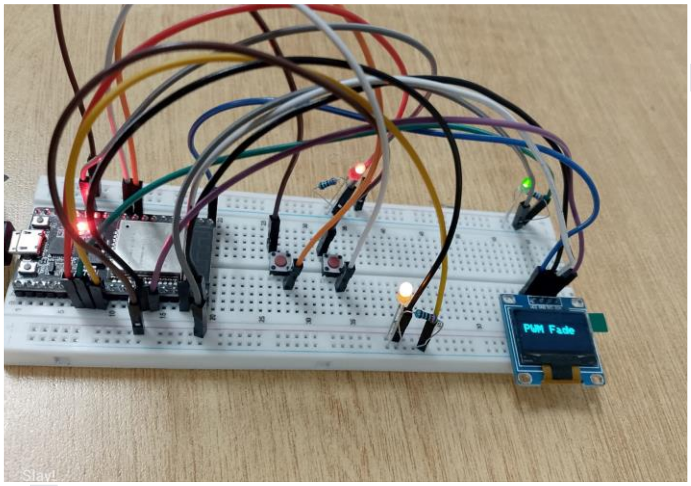
Mode 2

Alternate Blink



Mode 3

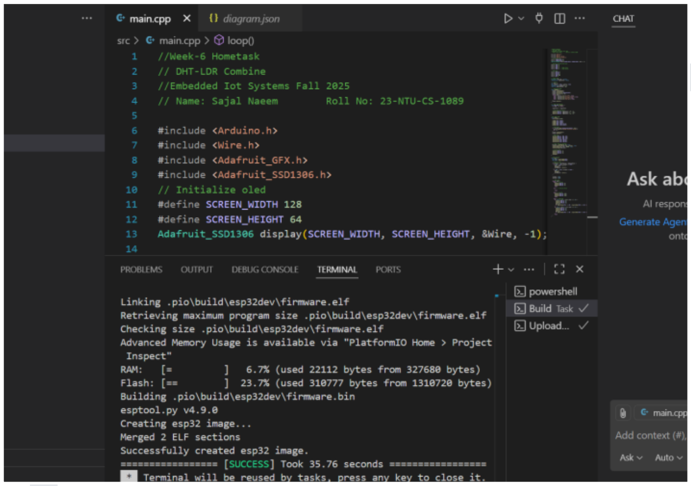
All are ON



Mode 4

PWM fade

Code Build Screenshot:



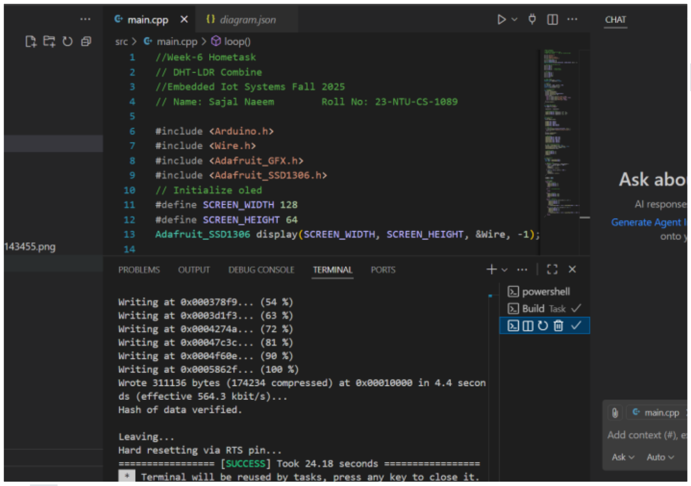
```
//Week-6 Hometask
// DHT-LDR Combine
//Embedded Iot Systems Fall 2025
// Name: Sajal Naeem        Roll No: 23-NTU-CS-1089

#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
// Initialize oled
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
Linking .pio\build\esp32dev\firmware.elf
Retrieving maximum program size .pio\build\esp32dev\firmware.elf
Checking size .pio\build\esp32dev\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project
 Inspect"
RAM:   [=         ]   6.7% (used 22112 bytes from 327680 bytes)
Flash: [==        ]  23.7% (used 310777 bytes from 1310720 bytes)
Building .pio\build\esp32dev\firmware.bin
esptool.py v4.9.0
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
================= [SUCCESS] Took 35.76 seconds =================
 *  Terminal will be reused by tasks, press any key to close it.
```

Build Screenshot

Code is successfully build

Code Upload Screenshot:



```
//Week-6 Hometask
// DHT-LDR Combine
//Embedded Iot Systems Fall 2025
// Name: Sajal Naeem        Roll No: 23-NTU-CS-1089

#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
// Initialize oled
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

```

```
Writing at 0x000378f9... (54 %)
Writing at 0x0003d1f3... (63 %)
Writing at 0x0004274a... (72 %)
Writing at 0x00047c3c... (81 %)
Writing at 0x0004f60e... (90 %)
Writing at 0x0005862f... (100 %)
Wrote 311136 bytes (174234 compressed) at 0x00010000 in 4.4 secon
ds (effective 564.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
================= [SUCCESS] Took 24.18 seconds =================
 * Terminal will be reused by tasks, press any key to close it.
```

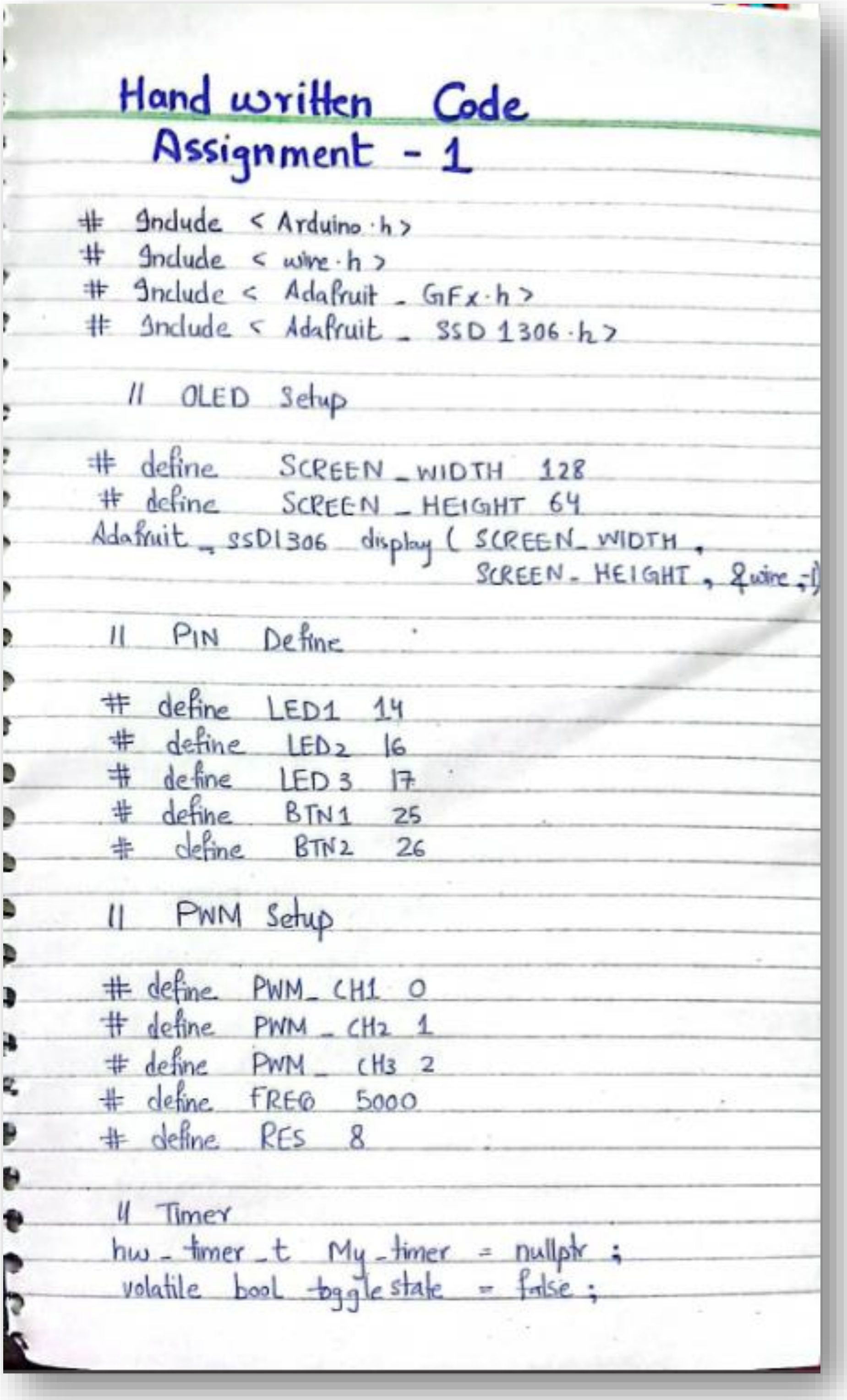
Upload Screenshot

Code is Successfully uploaded

WOKWI Project Link:

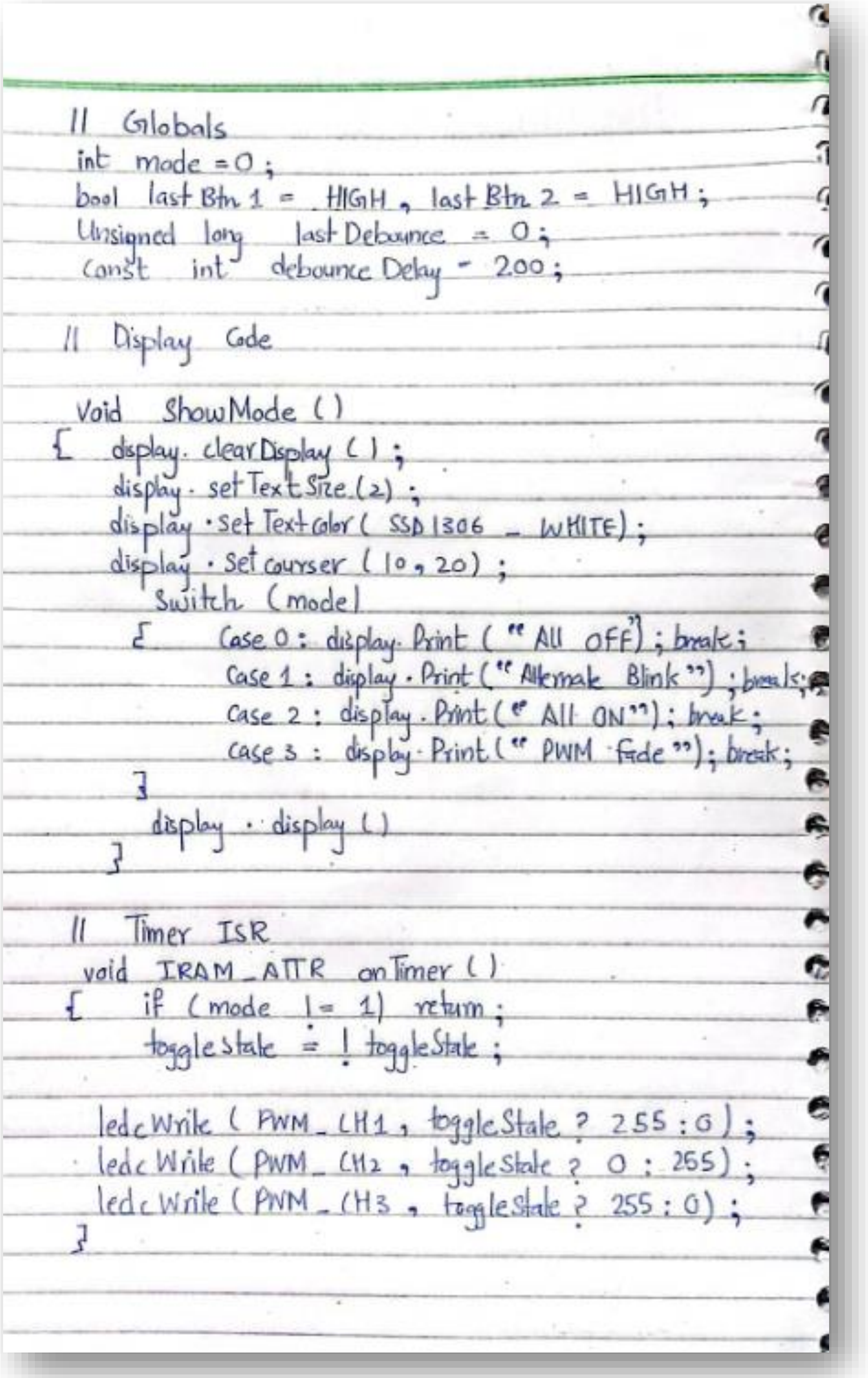
https://wokwi.com/projects/445158302195113985

Handwritten Code:

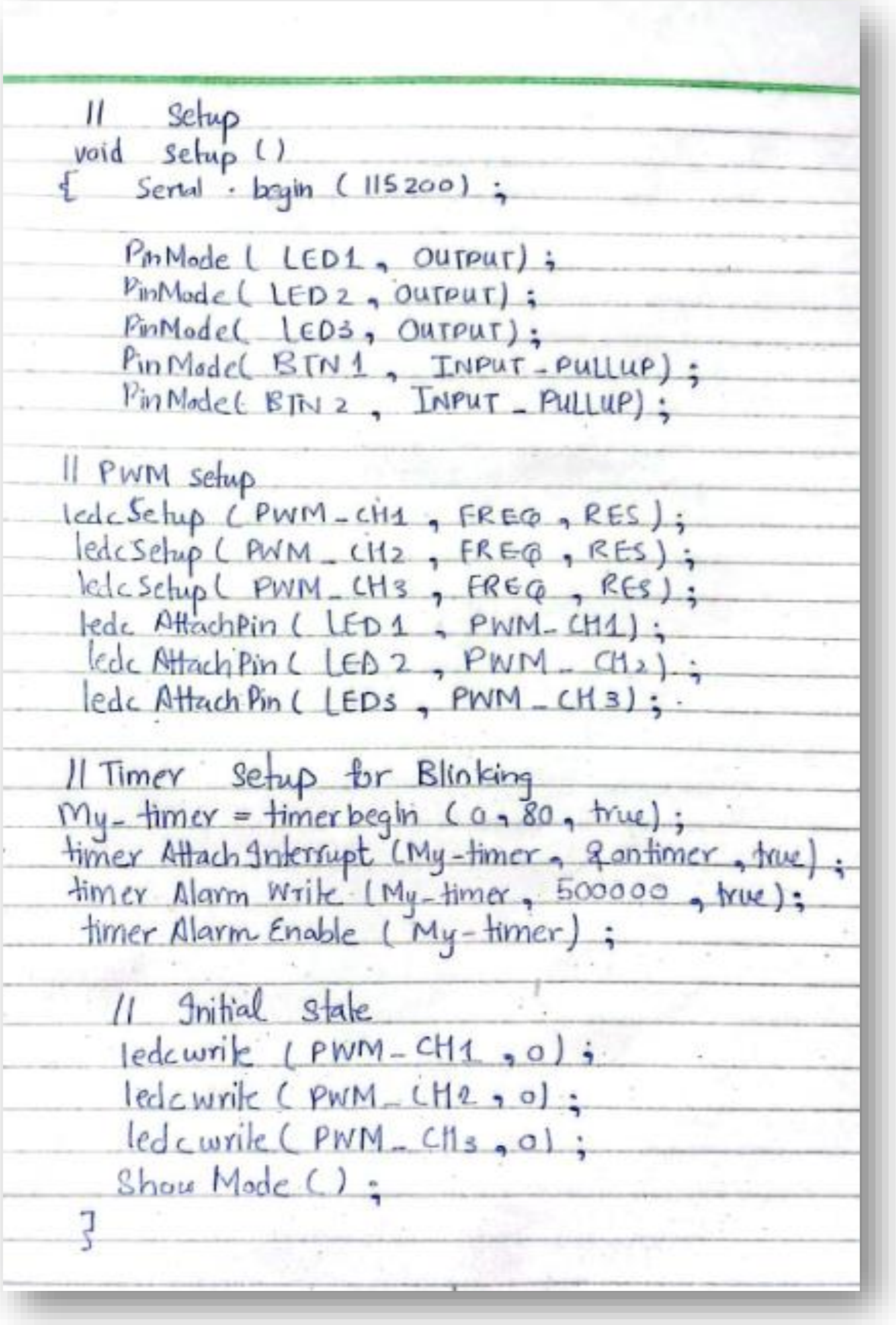


Hand written Code
Assignment - 1

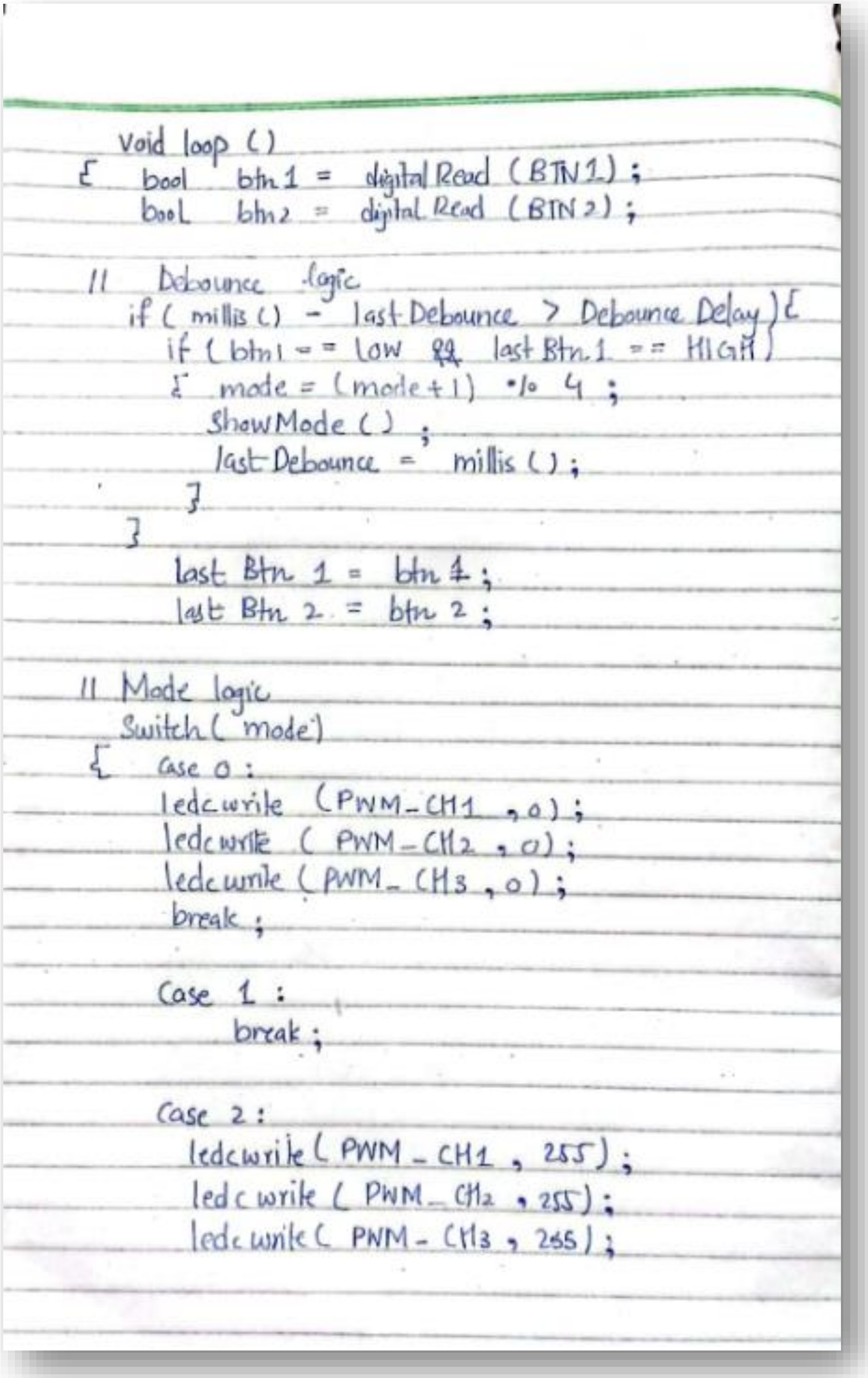
```
# Include < Arduino.h >
# Include < wire.h >
# Include < Adafruit_GFx.h >
# Include < Adafruit_SSD1306.h >

  // OLED Setup

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

  // PIN Define

# define  LED1  14
# define  LED2  16
# define  LED3  17
# define  BTN1  25
# define  BTN2  26

  // PWM Setup

# define  PWM_CH1  0
# define  PWM_CH2  1
# define  PWM_CH3  2
# define  FREQ   5000
# define  RES    8

  // Timer
hw_timer_t  My_timer = nullptr ;
volatile bool toggle state = false ;
```



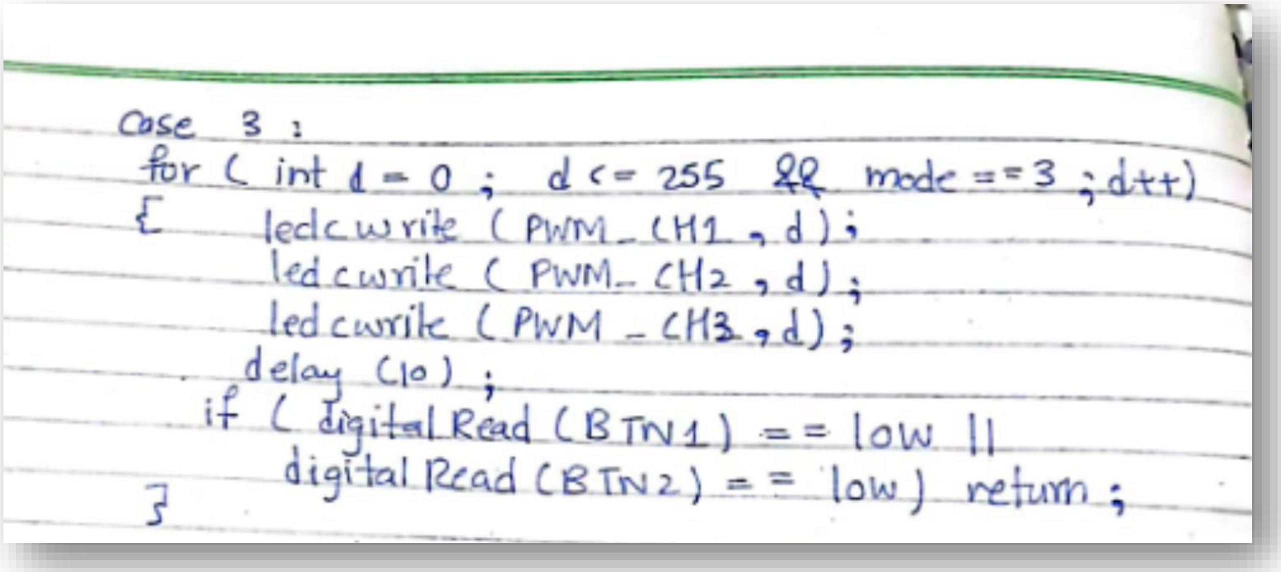
```
// Globals
int mode =0 ;
bool last Btn 1 = HIGH , last Btn 2 = HIGH ;
Unsigned long   last Debounce = 0 ;
Const int debounce Delay - 200 ;

// Display Code

Void ShowMode ( )
{ display. clearDisplay ( ) ;
  display . setTextSize (2) ;
  display . set Textcolor ( SSD1306 _ WHITE) ;
  display . Set courser ( 10 , 20) ;
    Switch (mode)
   {     Case 0 : display. Print ( " All OFF") ; break ;
         Case 1 : display . Print (" Alternate Blink ") ; break;
         Case 2 : display . Print (" All ON") ; break ;
         case 3 : display . Print (" PWM fade ") ; break ;
   }
    display . display ( )
  }

// Timer ISR
void IRAM_ATTR onTimer ( )
{    if (mode != 1) return ;
     togglestate = ! toggleState ;

  ledcWrite ( PWM_CH1 , toggleState ? 255 : 0 ) ;
  ledcWrite ( PWM_CH2 , toggleState ? 0 : 255) ;
  ledcWrite ( PWM_CH3 , toggleState ? 255 : 0) ;
}
```



```
//  Setup
void  setup ()
{   Serial . begin ( 115200) ;

     PinMode ( LED1 , OUTPUT) ;
     PinMode ( LED2 , OUTPUT) ;
     PinMode(  LED3 , OUTPUT) ;
     PinMode( BTN1 ,  INPUT_PULLUP) ;
     PinMode( BTN2 ,  INPUT_PULLUP) ;

// PWM setup
ledcSetup (PWM-CH1 , FREQ , RES ) ;
 ledcSetup ( PWM_CH2 , FREQ , RES) ;
 ledcSetup ( PWM_CH3 , FREQ , RES) ;
 ledc AttachPin ( LED1 , PWM-CH1) ;
  ledc AttachPin ( LED 2 , PWM_CH2) ;
 ledc AttachPin ( LED3 , PWM_CH3) ;

// Timer  Setup for Blinking
My_timer = timerbegin ( 0 , 80 , true) ;
timer AttachInterrupt (My-timer , &ontimer , true) ;
timer Alarm Write (My_timer , 500000 , true) ;
 timer Alarm Enable ( My-timer) ;

   //  Initial  state
   ledcwrite ( PWM-CH1 , 0) ;
   ledcwrite ( PWM_CH2 , 0) ;
   ledcwrite ( PWM_CH3 , 0) ;
   Show Mode ( ) ;
 }
```



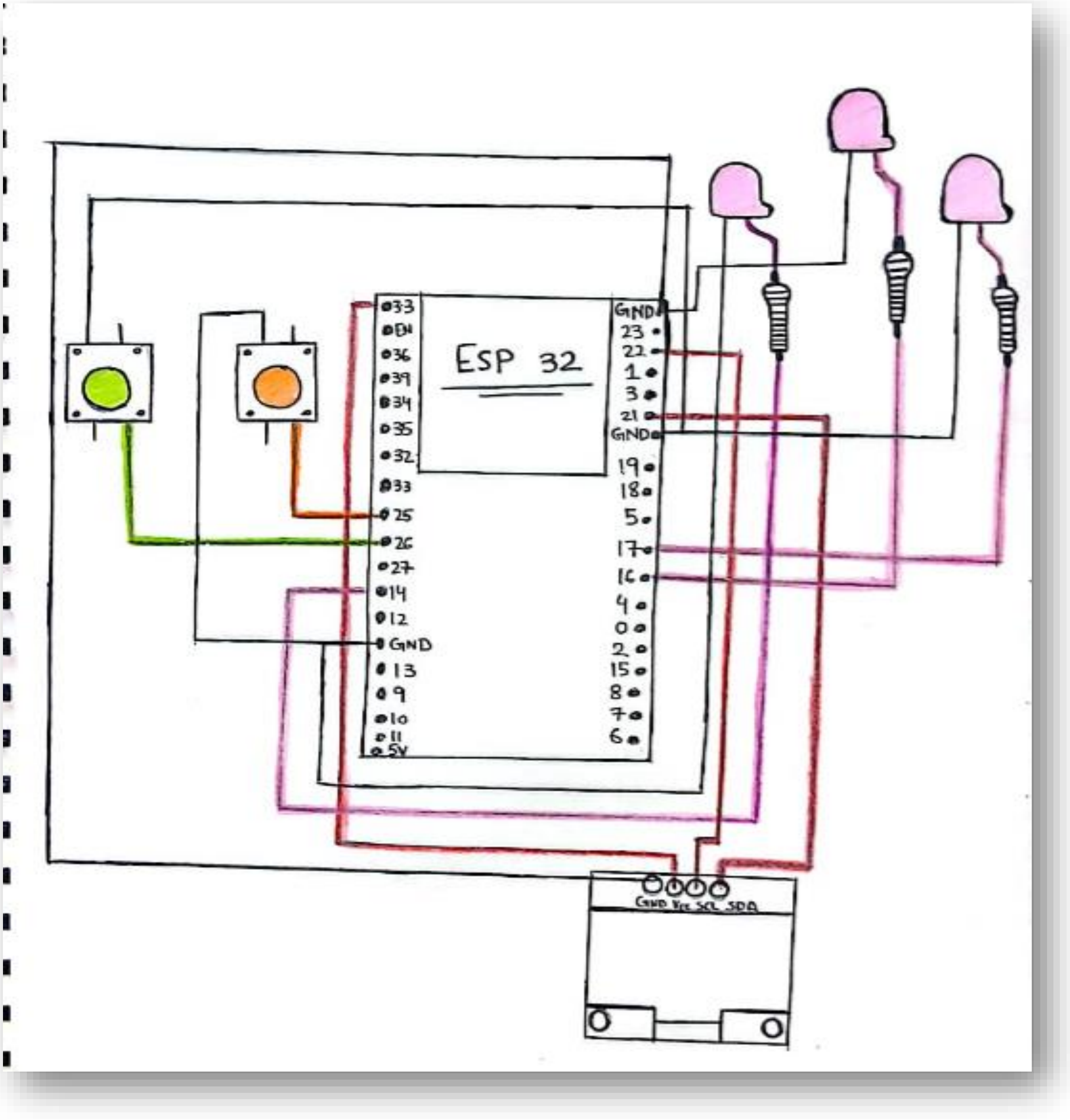
```
Void loop ()
{  bool   btn1 =  digital Read (BTN1) ;
   bool   btn2 =  digital Read (BTN2) ;

// Debounce logic
  if ( millis () - last Debounce > Debounce Delay ){
     if ( btn1 == Low && last Btn 1 == HIGH )
     { mode = (mode+1) % 4 ;
       ShowMode () ;
       last Debounce = millis () ;
     }
  }
     last Btn 1 = btn 1 ;
     last Btn 2 = btn 2 ;

// Mode logic
  Switch ( mode )
{   case 0 :
     ledcwrite (PWM_CH1 , 0) ;
     ledcwrite ( PWM_CH2 , 0) ;
     ledcwrite ( PWM_CH3 , 0) ;
     break ;

     Case 1 :
          break ;

     Case 2 :
       ledcwrite ( PWM_CH1 , 255) ;
       ledcwrite ( PWM_CH2 , 255) ;
       ledcwrite ( PWM_CH3 , 265) ;
```



```
case  3 :
 for ( int d = 0 ;  d <= 255  && mode == 3 ;d++)
 {       ledcwrite ( PWM_CH1 , d );
         ledcwrite ( PWM_ CH2 , d );
         ledcwrite ( PWM _ CH3 , d );
        delay (10) ;
     if ( digitalRead (BTN1) == low ||
          digital Read (BTN2) == low) return ;
 }
```

Hand Drawn Sketch:

Component Wire	Pin Color
Ground Pins	Black
Button 1 pin	orange
Button 2 pin	green
Led pins	pink
OLED pins	red


ESP 32
3.3
EN
36
39
34
35
32
33
25
26
27
14
12
GND
13
9
10
11
5V
GND
23
22
1
3
21
GND
19
18
5
17
16
4
0
2
15
8
7
6
GND
Vcc
SCL
SDA

Pin Map:

Component	Pin name	Pin number
Led 1	Resistor pin	GPIO 14
Led 1	Ground pin	GND 1
Led 2	Resistor pin	GPIO 16
Led 2	Ground pin	GND 2
Led 3	Resistor pin	GPIO 17
Led 3	Ground pin	GND 3
Mode Button	One side	GPIO 25
Mode Button	Second side	GND 1
Reset Button	One side	GPIO 26
Reset Button	Second side	GND 3
OLED	Ground Pin	GND 2
OLED	Voltage pin	GPIO 3.3
OLED	Serial clock pin	GPIO 22
OLED	Serial data pin	GPIO 21