National Textile University, Faisalabad

Department of Computer Science

Name:	Muhammad Abdullah
Class:	BSCS_B 5th Semester
Registration No:	23-NTU-CS-1056
Assignment :	Assignment 01
Course Name:	Embedded IoT Systems
Submitted To:	**Sir Nasir**
Submission Date:	23/10/2025

Task 1: LED Modes Control with Two Buttons and OLED Display

Objective:

The purpose of this project is to control multiple LEDs using two push buttons and to show the LED status on an OLED display. The buzzer is used for alerts, and all components are simulated in Wokwi.

Components Used:

- ESP32 Microcontroller
- 2 Push Buttons
- 3 LEDs (Red, Green, Blue)
- 1 Buzzer
- 1 OLED Display (0.96” I2C 128x64)
- Jumper Wires and Breadboard

Circuit Diagram:

The circuit was designed in **Wokwi**.
Connections are as follows:

- Push Button 1 → GPIO 4
- Push Button 2 → GPIO 5
- LEDs → GPIO 12, 13, 14
- Buzzer → GPIO 27
- OLED (SDA → 21, SCL → 22)

Working & Code Logic (Task A):

- **Button 1** cycles through four LED modes:

1. **Both OFF** – All LEDs remain off.
2. **Alternate Blink** – LEDs blink one after another.
3. **Both ON** – All LEDs turn on.
4. **PWM Fade** – LEDs fade in and out using analogWrite.

- **Button 2** resets all LEDs to OFF mode.
- The **OLED display** shows the current mode (e.g., “Mode: Alternate Blink”).

Example Output (on OLED):

Mode: Both OFF

Mode: Alternate Blink

Mode: Both ON

Mode: PWM Fade

Mode: Reset to OFF

Result:

The system successfully switches between LED states using the first button and resets them using the second. The OLED displays each mode clearly, and the circuit works as intended.

Task 2: Single Button Control with Short/Long Press Detection

Objective:

The aim of this project is to detect short and long presses of a single button and perform different actions. The OLED display shows the detected event, and a buzzer gives sound feedback.

Components Used:

- ESP32 Microcontroller
- 1 Push Button
- 1 LED
- 1 Buzzer
- 1 OLED Display (0.96” I2C 128x64)
- Breadboard and Jumper Wires

Circuit Diagram:

Connections:

- Button → GPIO 4
- LED → GPIO 12
- Buzzer → GPIO 27
- OLED (SDA → 21, SCL → 22)

Working & Code Logic (Task B):

- The program detects the **duration of button press**.
- If the button is pressed **short (< 1.5 seconds)** → **LED toggles ON/OFF**.
- If the button is pressed **long (> 1.5 seconds)** → **Buzzer plays a tone**.
- The **OLED display** shows the event type:
    - “Short Press: LED Toggled”
    - “Long Press: Buzzer Tone Played”

Example Output (on OLED):

Short Press → LED ON

Short Press → LED OFF

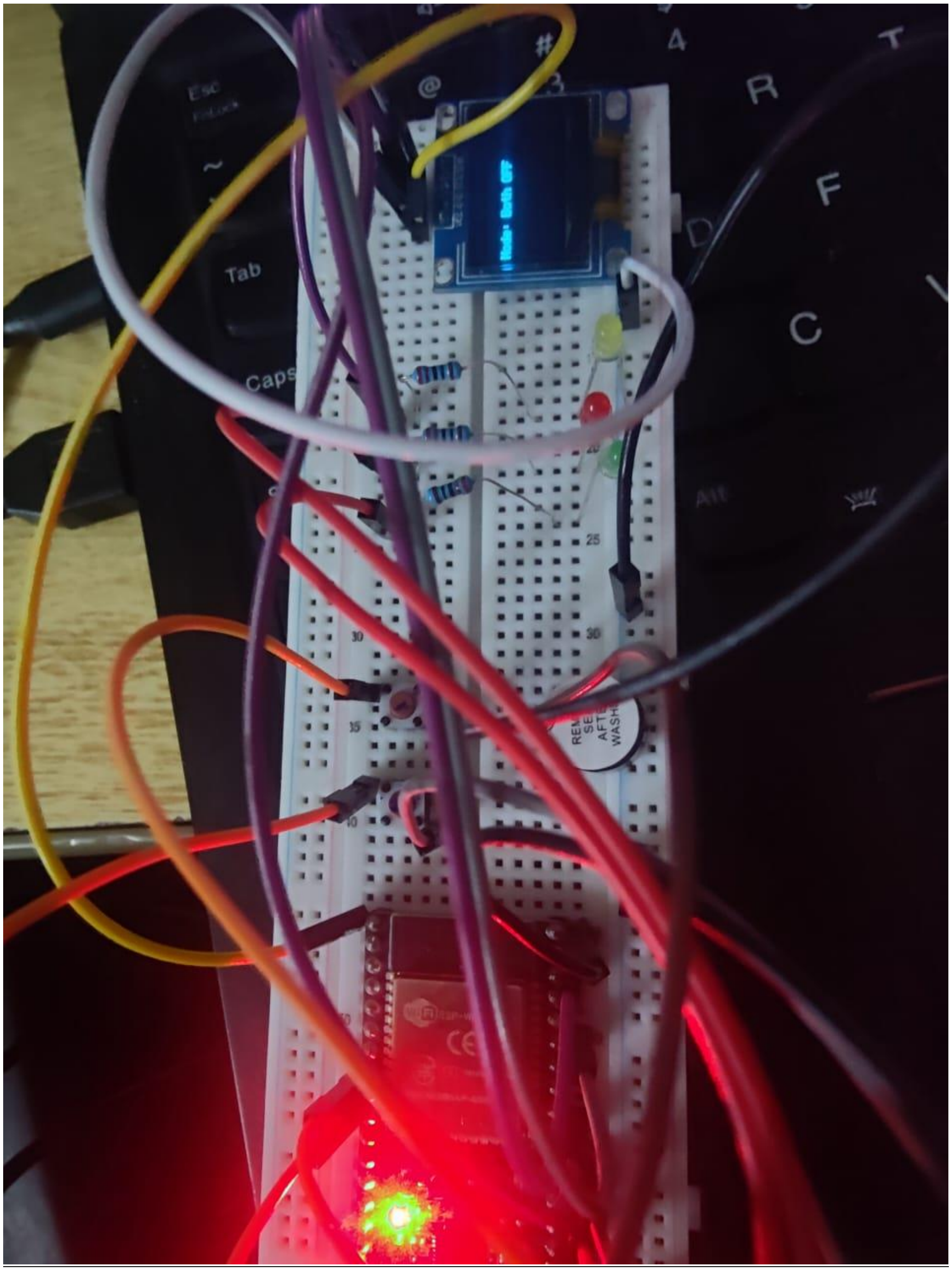
Long Press → Buzzer Tone

Result:

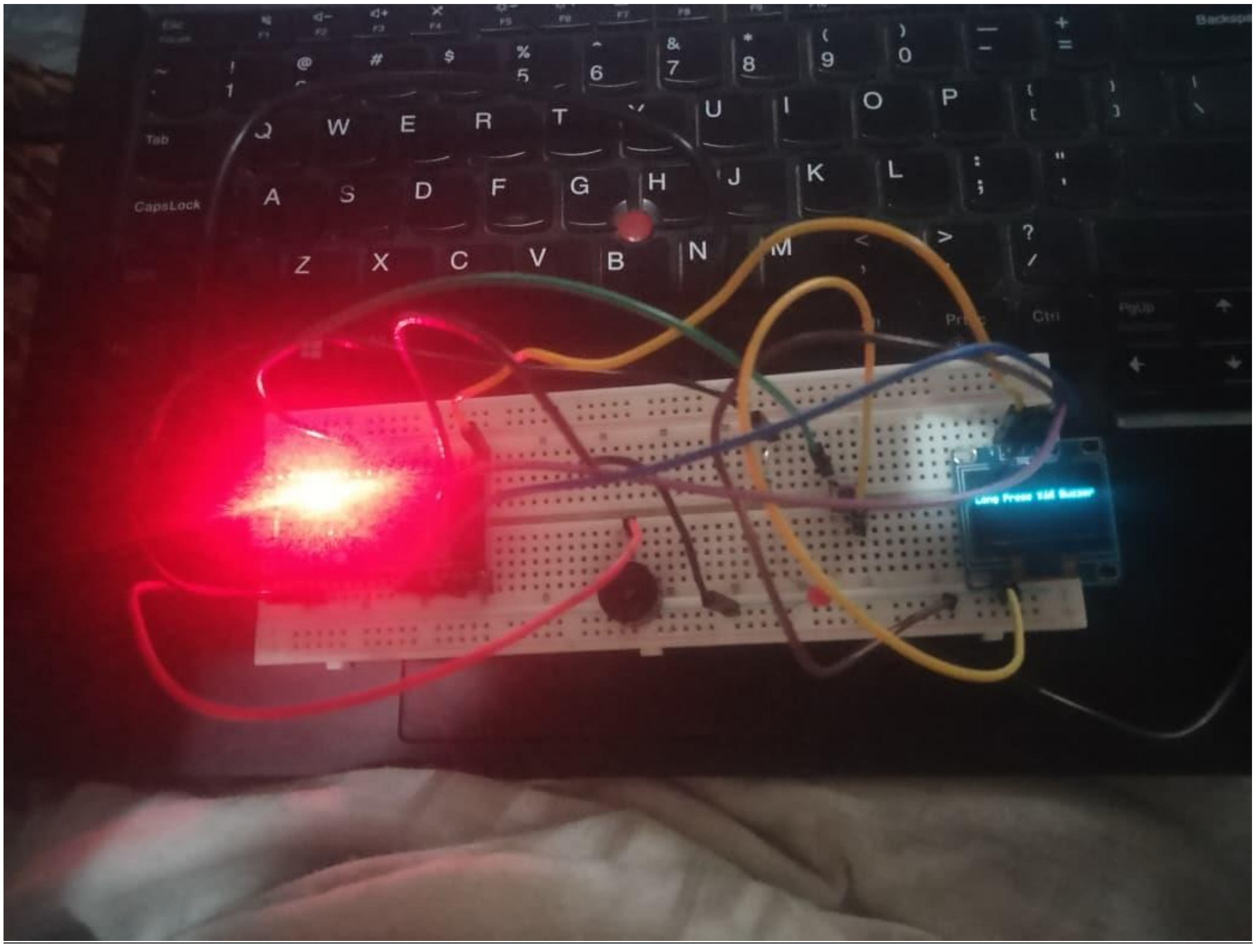
The circuit correctly detects short and long button presses. The OLED provides instant visual feedback, and the buzzer tone plays on a long press. The project demonstrates event-based button handling in embedded systems.

Pictures of Kit:

Task 01

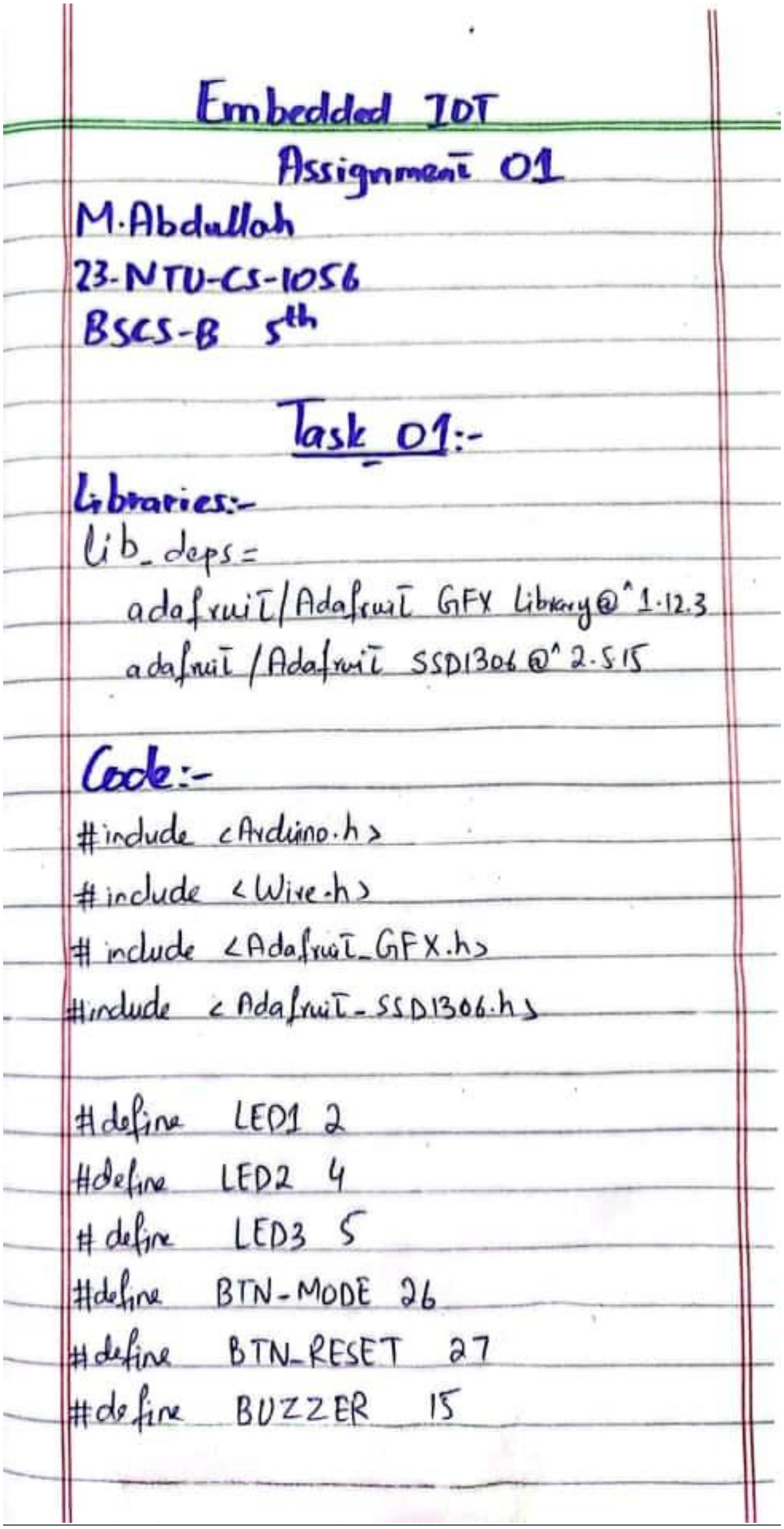


Task 02



Hand-Written Code:

Task 01



Embedded IOT

Assignment 01

M.Abdullah

23-NTU-CS-1056

BSCS-B 5th

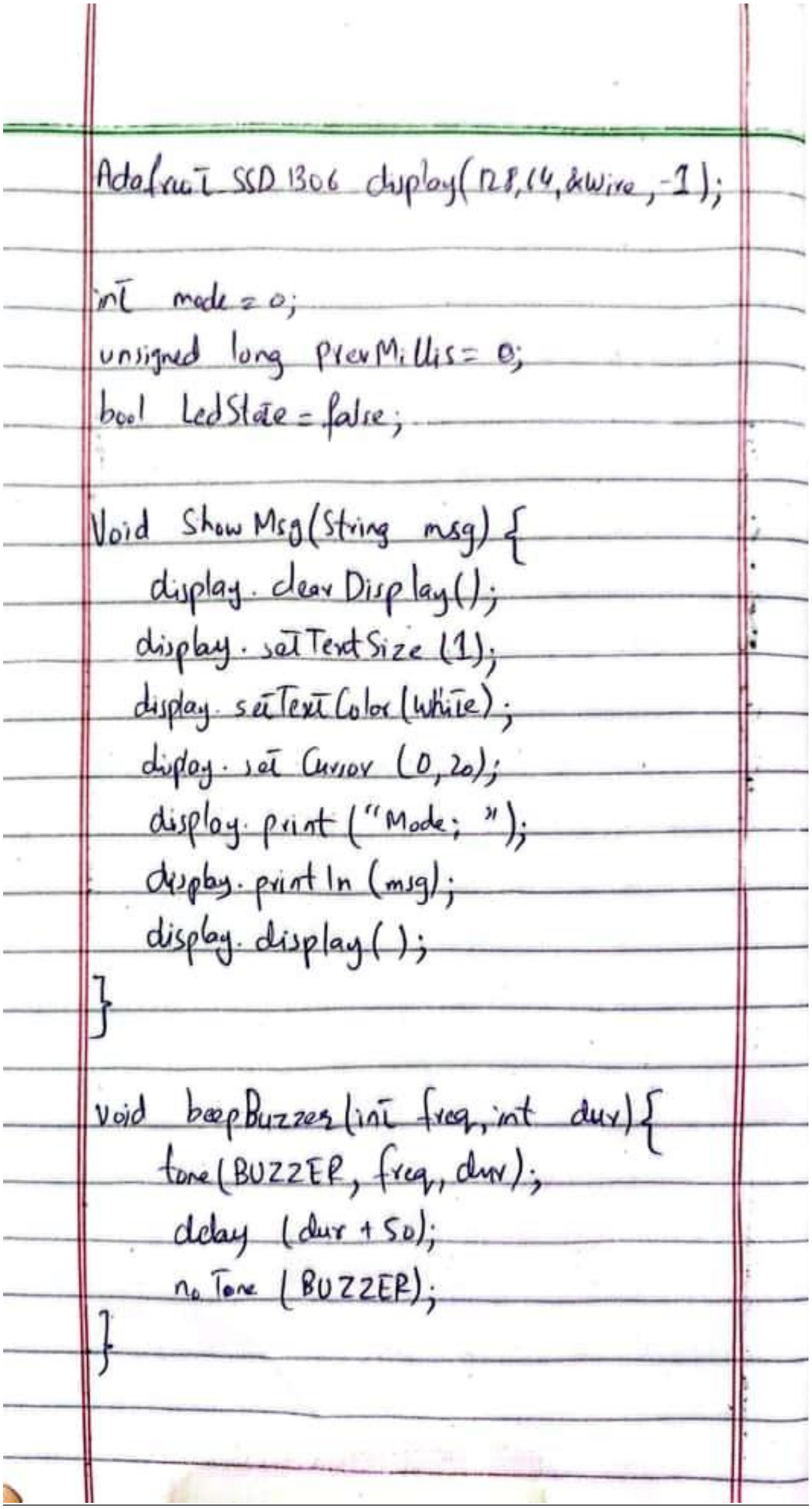
Task 01:-

Libraries:-

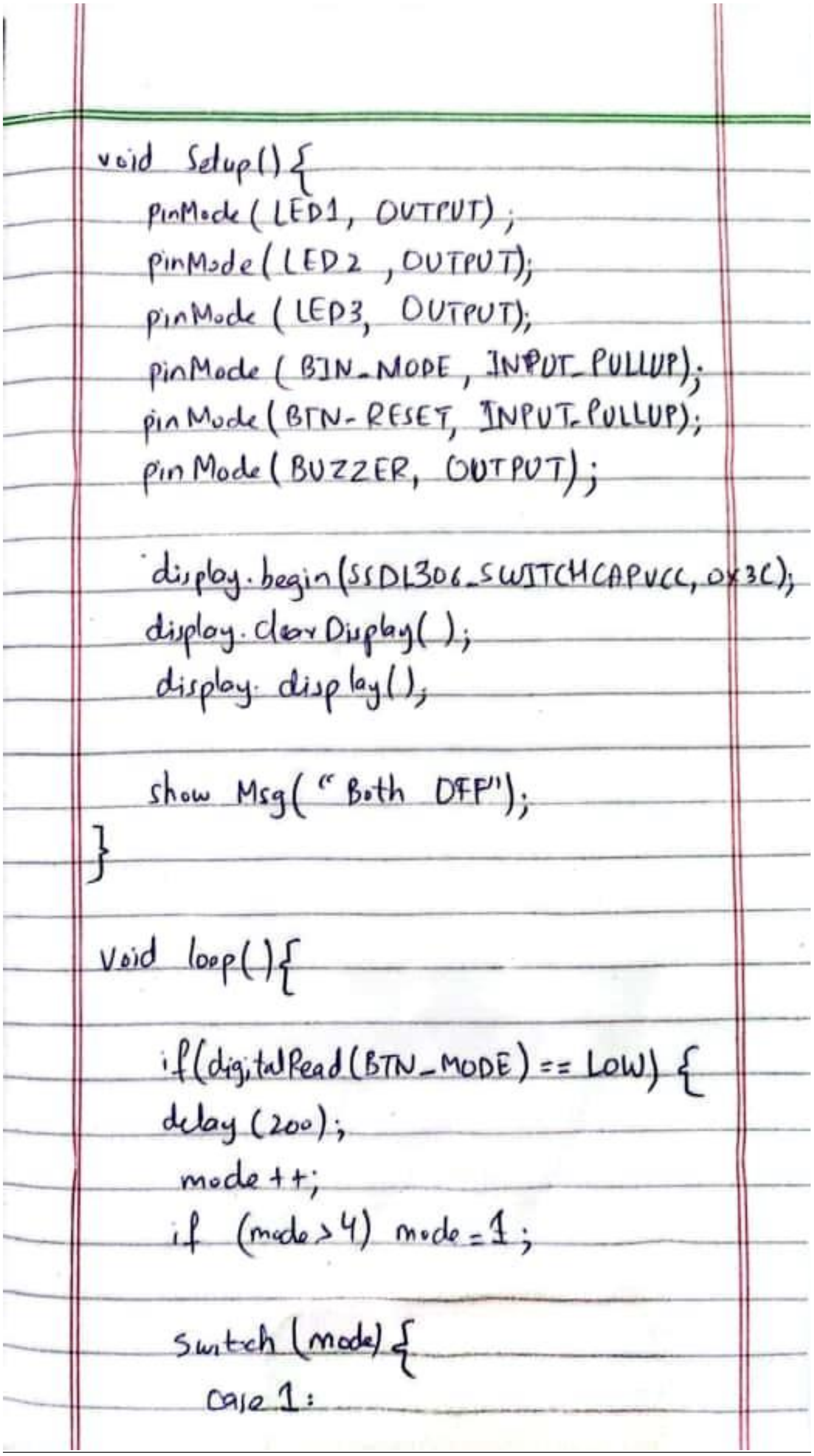
```
lib_deps=
   adafruit/Adafruit GFX Library@^1.12.3
   adafruit/Adafruit SSD1306@^2.5.15
```

Code:-

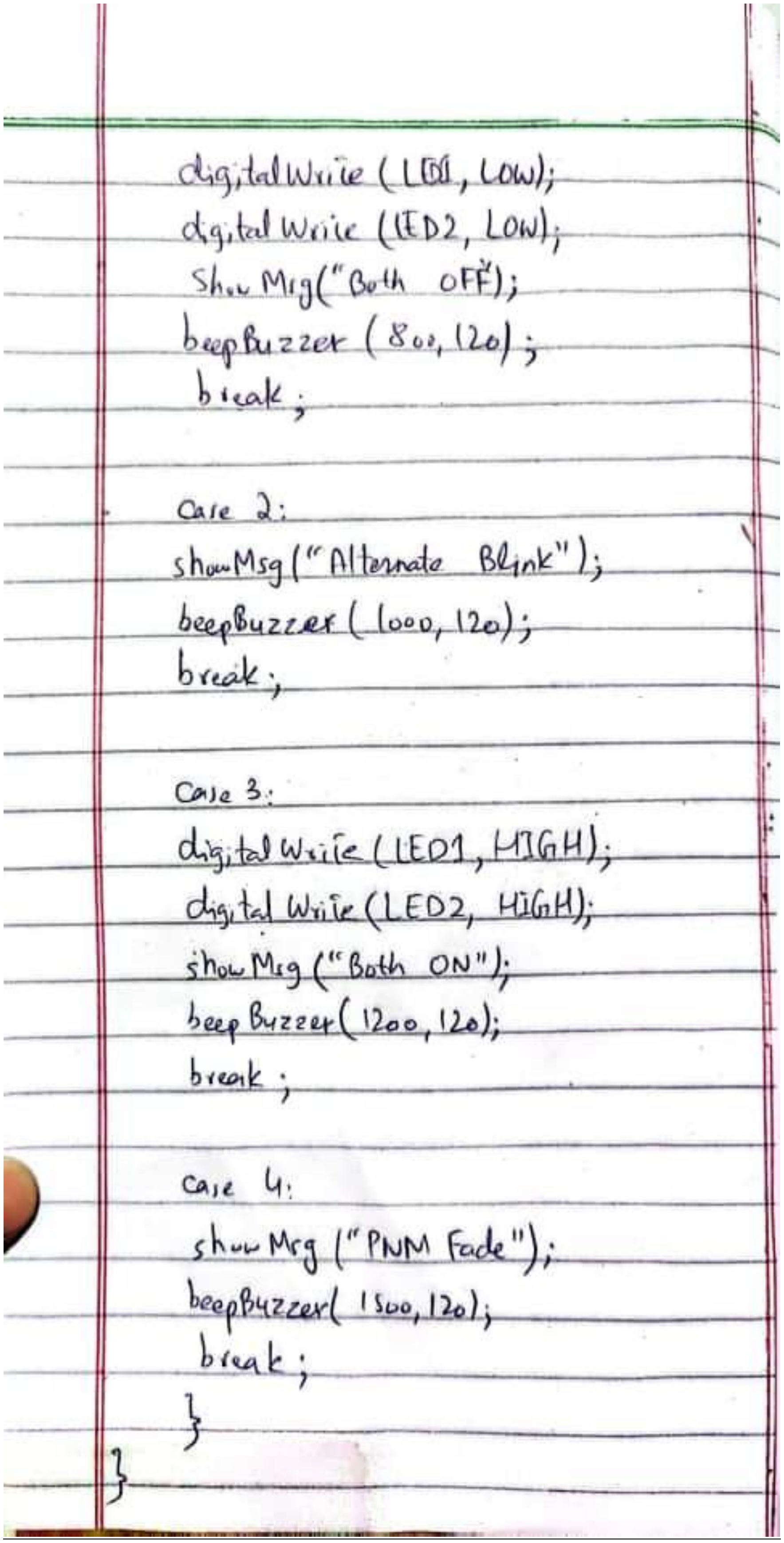
```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit-SSD1306.h>

#define LED1 2
#define LED2 4
#define LED3 5
#define BTN-MODE 26
#define BTN_RESET 27
#define BUZZER 15
```



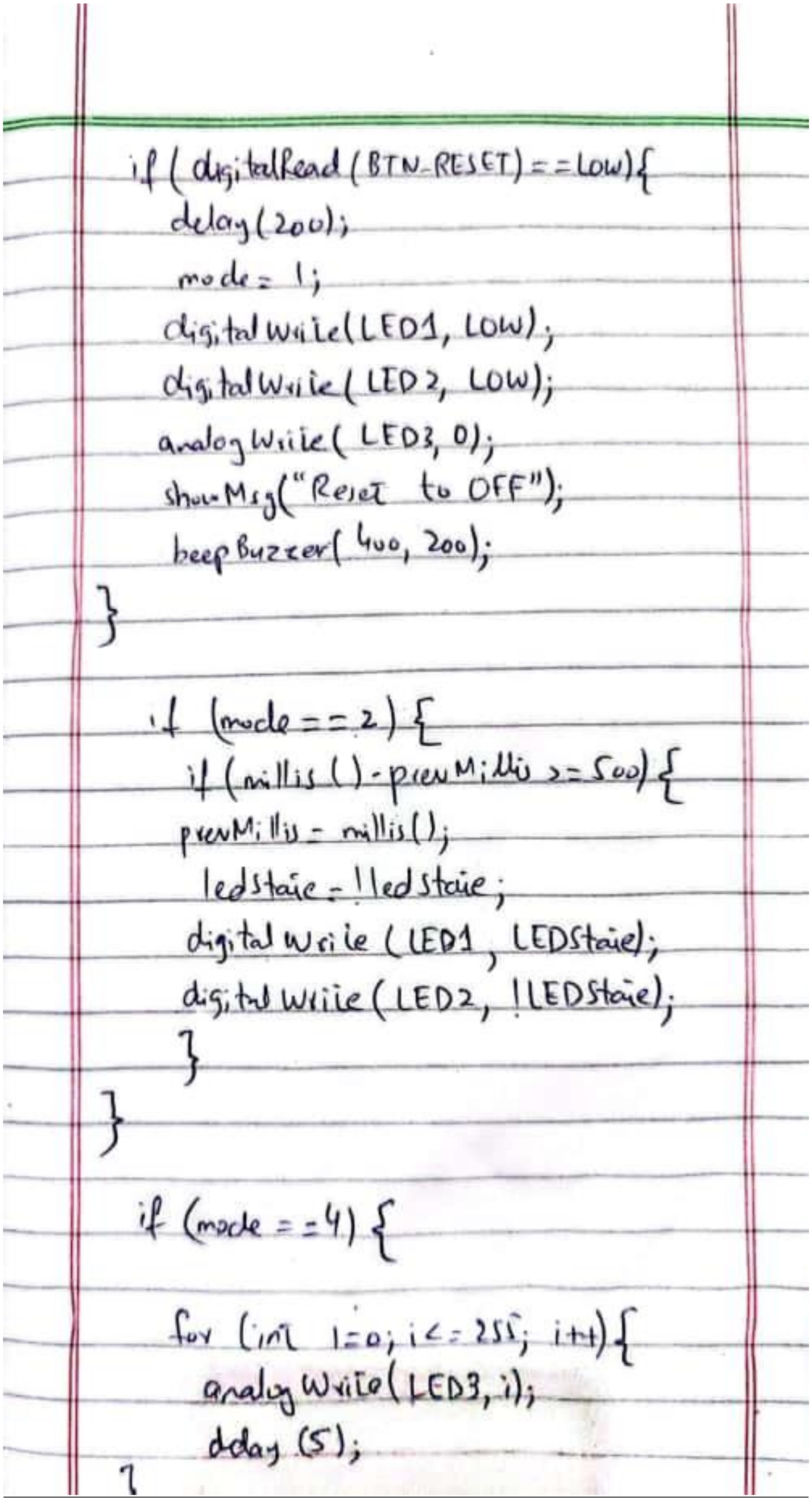
```
Adafruit_SSD1306 display(128,64,&Wire,-1);

int mode = 0;
unsigned long prevMillis = 0;
bool LedState = false;

Void ShowMsg(String msg) {
    display.clearDisplay();
    display.setTextSize (1);
    display.setTextColor(White);
    diplay.setCursor (0,20);
    display.print ("Mode; ");
    display.println (msg);
    display.display( );
}

void beepBuzzer(int freq, int dur){
    tone(BUZZER, freq, dur);
    delay (dur + 50);
    noTone (BUZZER);
}
```



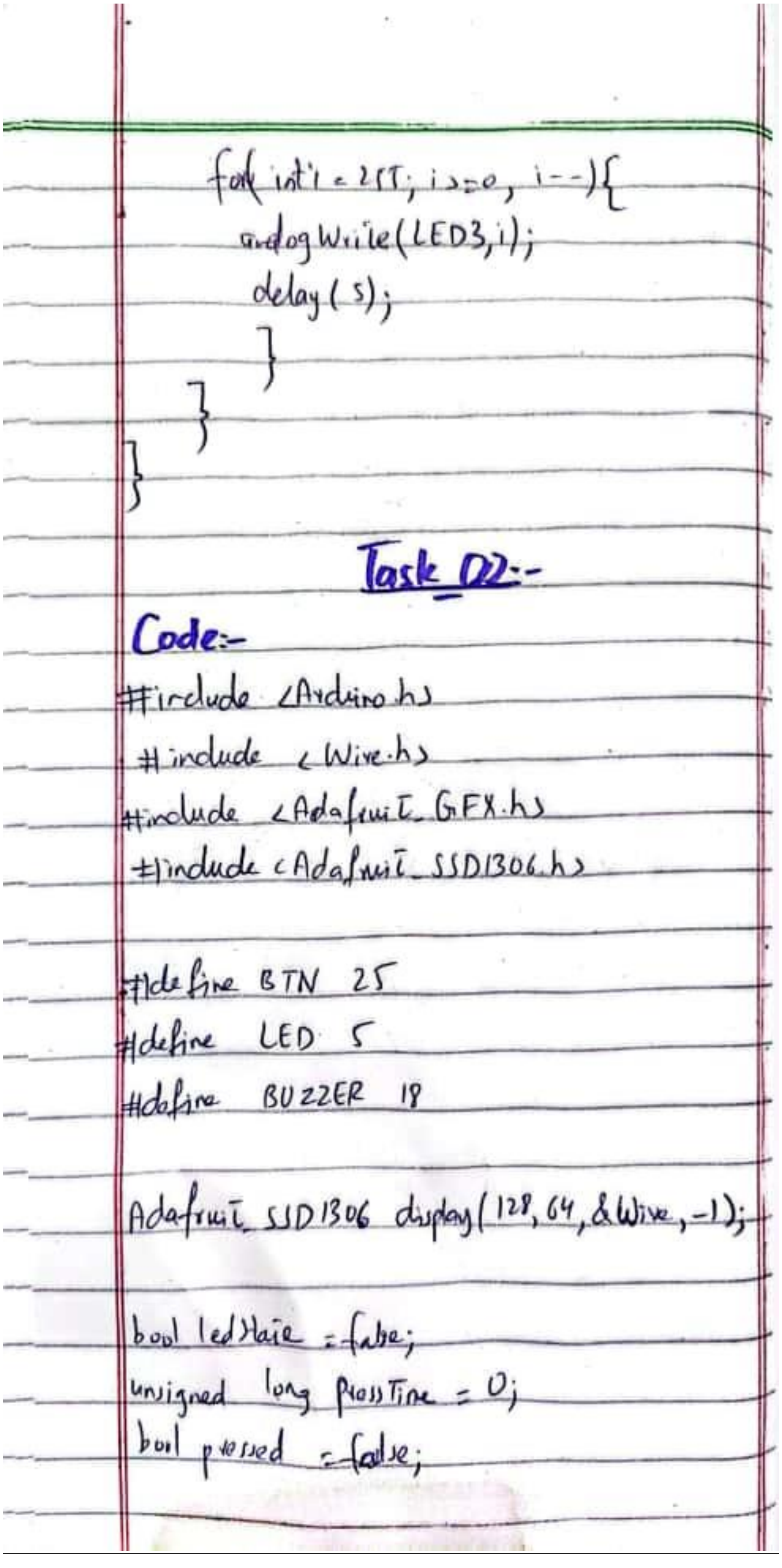
```
void Setup(){
    pinMode ( LED1, OUTPUT) ;
    pinMode(LED2 ,OUTPUT);
    pinMode ( LED3, OUTPUT);
    pinMode ( BTN_MODE , INPUT_PULLUP);
    pin Mode(BTN-RESET, INPUT_PULLUP);
    pin Mode(BUZZER, OUTPUT);

    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay( );
    display. display();

    show Msg(“Both OFF”);
}

Void loop(){

    if(digitalRead(BTN_MODE) == LOW) {
    delay (200);
     mode ++;
     if (mode > 4) mode = 1;

     Switch (mode) {
      case 1:
```



```
digitalWrite (LED1, LOW);
digital Write (LED2, LOW);
Show Msg("Both OFF");
beepBuzzer (800, 120);
break;

case 2:
showMsg("Alternate Blink");
beepBuzzer(1000, 120);
break;

Case 3:
digital Write(LED1, HIGH);
digital Write(LED2, HIGH);
show Msg("Both ON");
beep Buzzer(1200, 120);
break;

case 4:
show Msg("PNM Fade");
beepBuzzer( 1500,120);
break;
}
}
```



```
if ( digitalRead (BTN-RESET) = = LOW){
   delay(200);
   mode = 1;
   digitalWrite(LED1, LOW);
   digitalWrite( LED2, LOW);
   analogWrite( LED3, 0);
   showMsg("Reset  to OFF");
   beepBuzzer( 400, 200);
}

  if  (mode = = 2) {
    if (millis () - prevMillis >= 500) {
   prevMillis = millis();
    ledState = !ledState;
   digitalWrite (LED1, LEDState);
   digitalWrite(LED2, !LEDState);
   }
}

 if (mode = = 4) {

   for (int  i=0; i<=255; i++){
     analogWrite(LED3, i);
     delay (5);
}
```



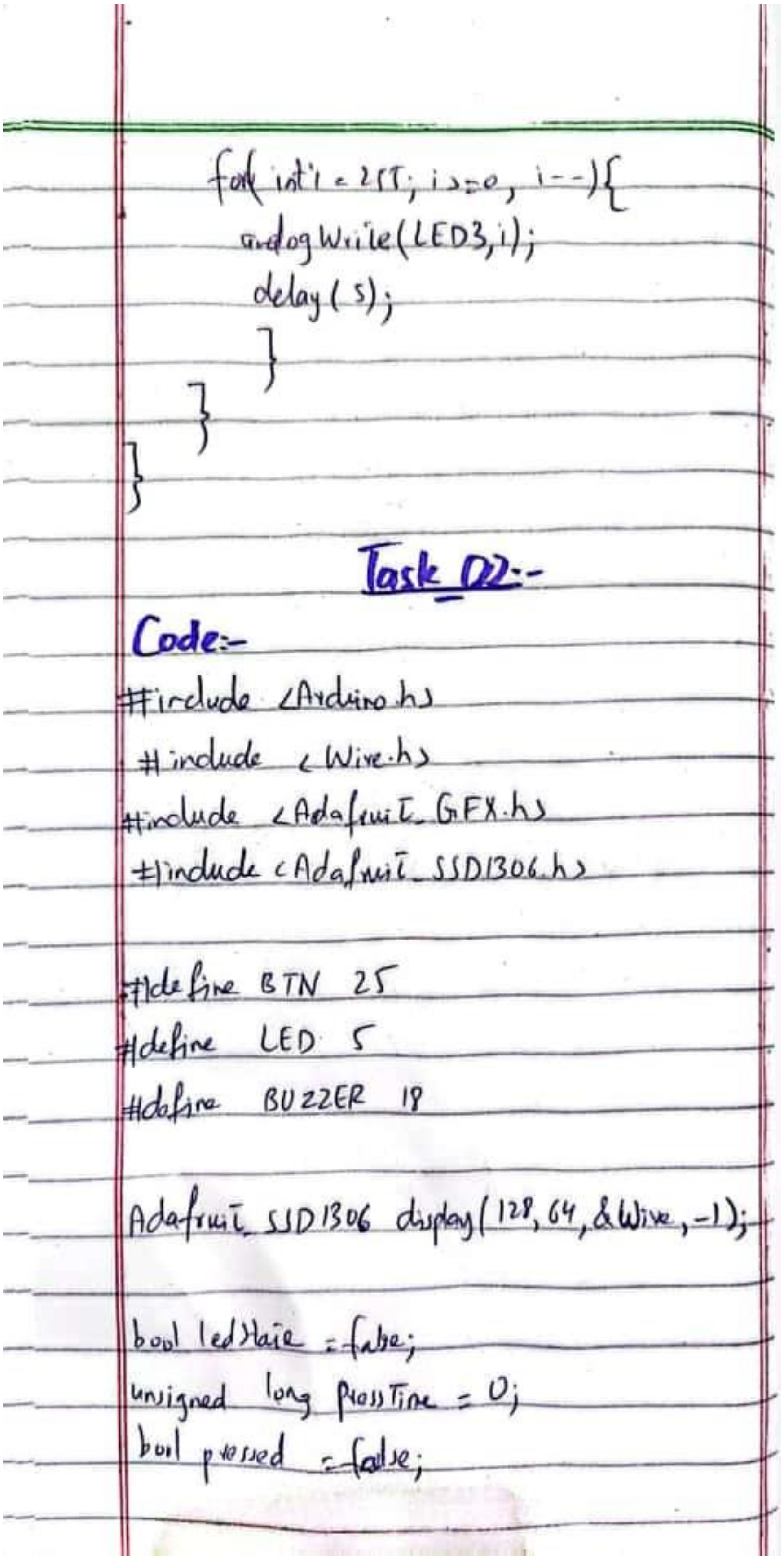
```
     for( int i = 255; i>=0, i--){
       analogWrite(LED3,i);
       delay( 5);
       }
   }
}
```

Task 02:-

Code:-

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

#define BTN 25
#define LED 5
#define BUZZER 18

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

bool ledState = false;
unsigned long pressTime = 0;
bool pressed = false;
```

Task 02

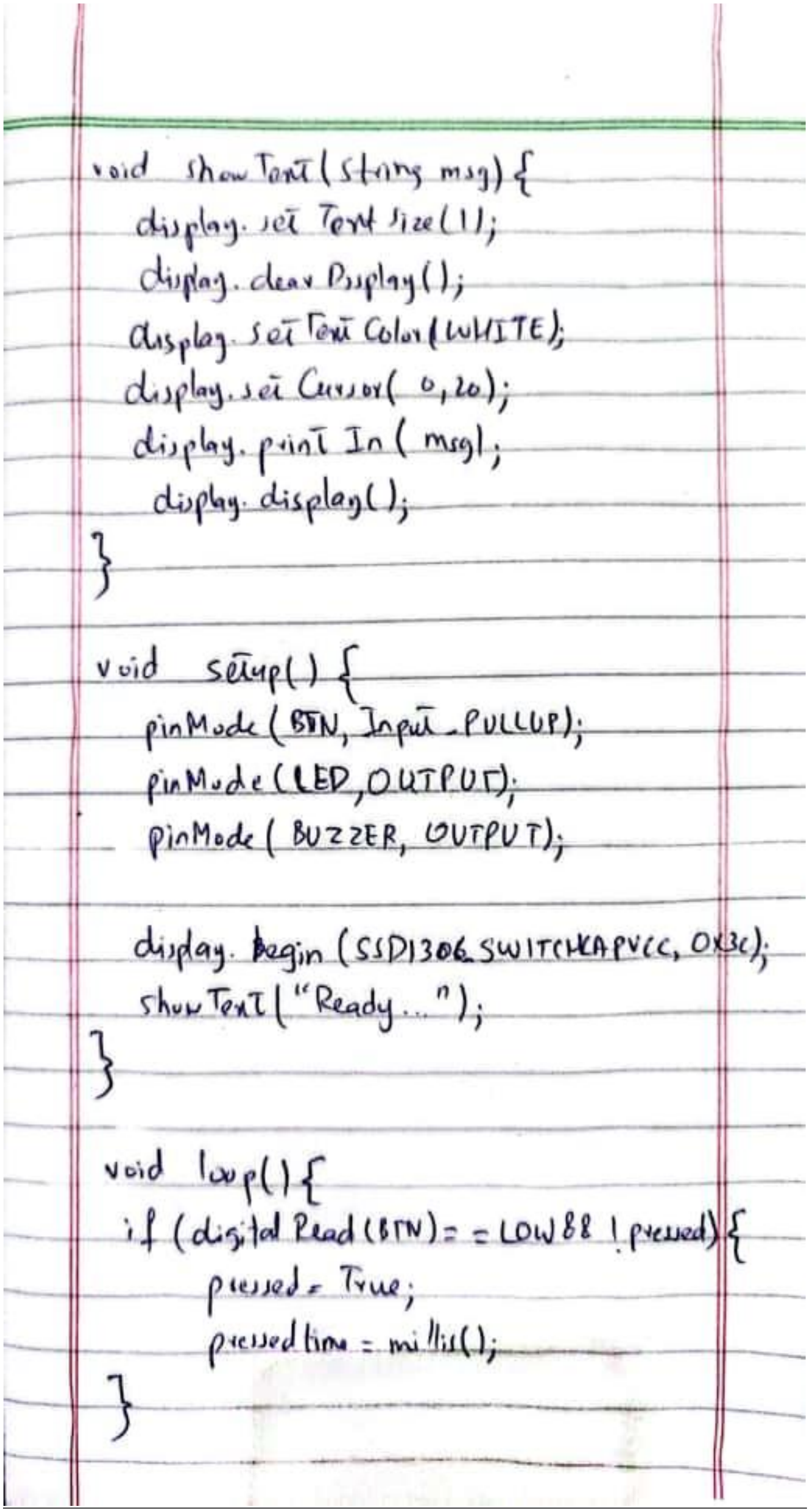


```
    for( int i = 255; i>=0, i--){
      analogWrite(LED3,i);
      delay(5);
      }
  }
}
```

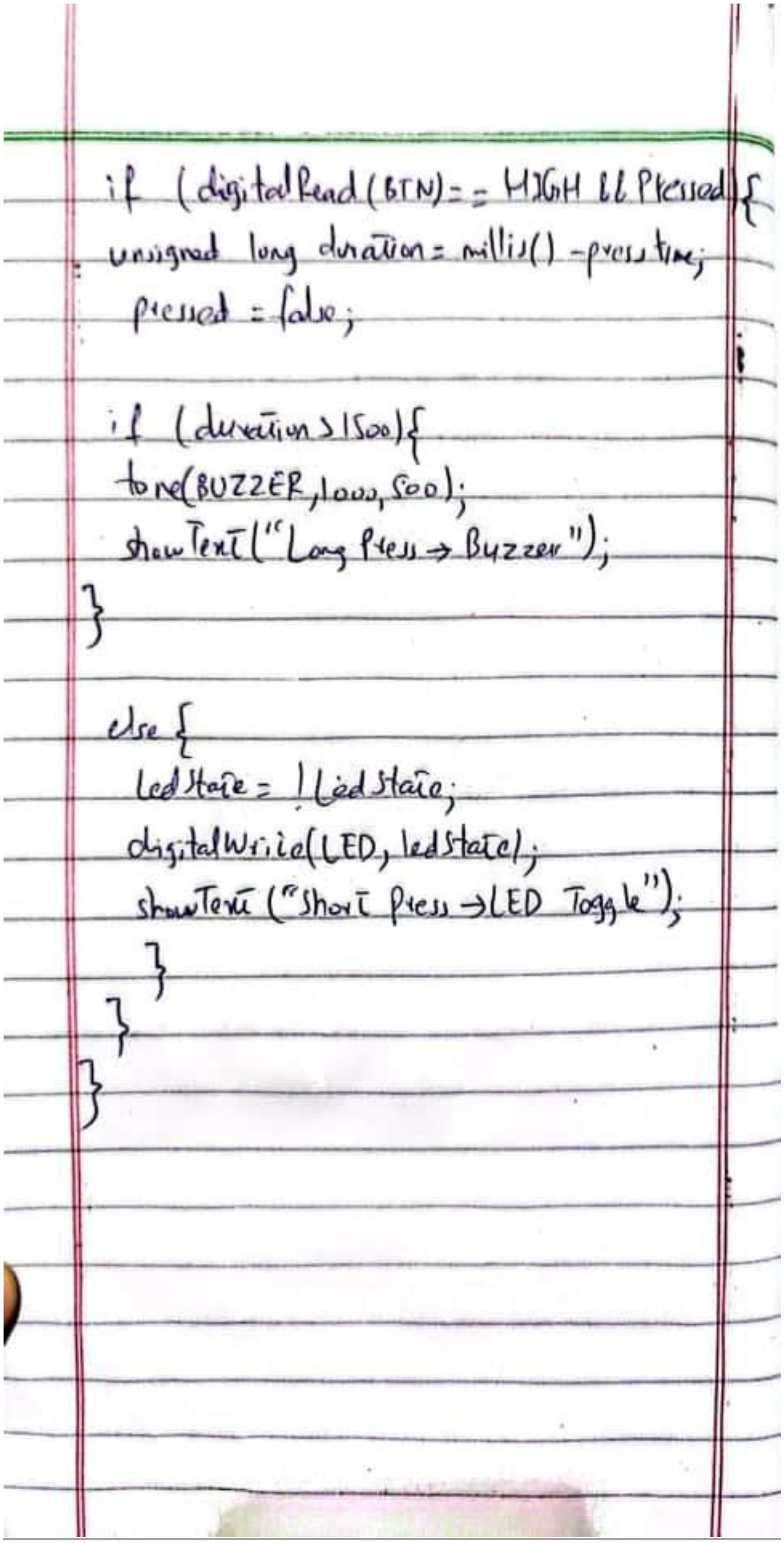
Task 02:-

Code:-

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

#define BTN 25
#define LED 5
#define BUZZER 18

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

bool ledState = false;
unsigned long pressTime = 0;
bool pressed = false;
```



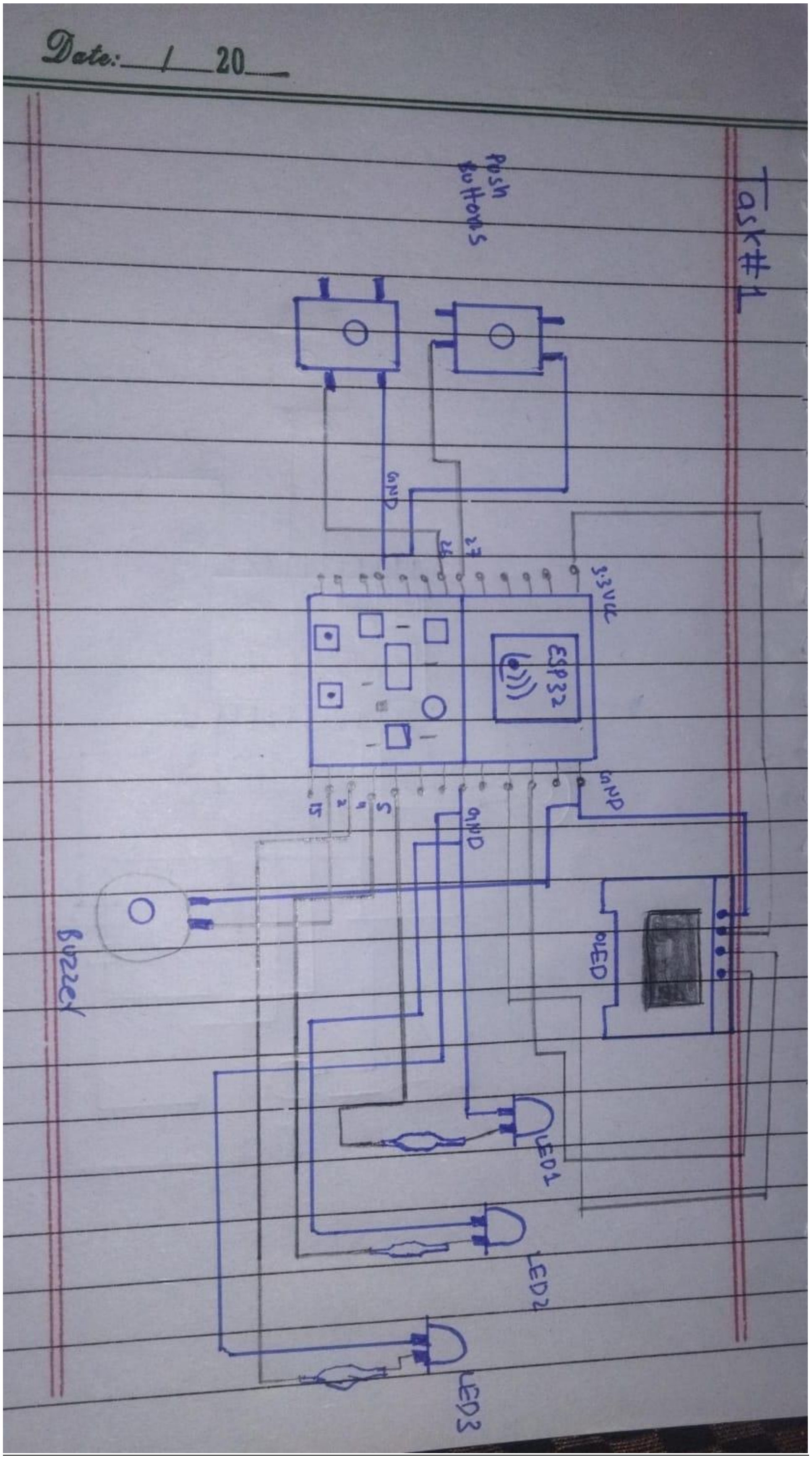
```
void showText(string msg){
  display.setTextSize(1);
  display.clearDisplay();
  display.setTextColor(WHITE);
  display.setCursor( 0,20);
  display.printIn( msg);
  display.display();
}

void setup() {
  pinMode(BTN, Input_PULLUP);
  pinMode(LED,OUTPUT);
  pinMode( BUZZER, OUTPUT);

  display.begin(SSD1306_SWITCHCAPVCC, 0x3c);
  showText("Ready...");
}

void loop(){
 if (digitalRead(BTN)= = LOW && !pressed){
      pressed = True;
      pressedtime = millis();
 }
```



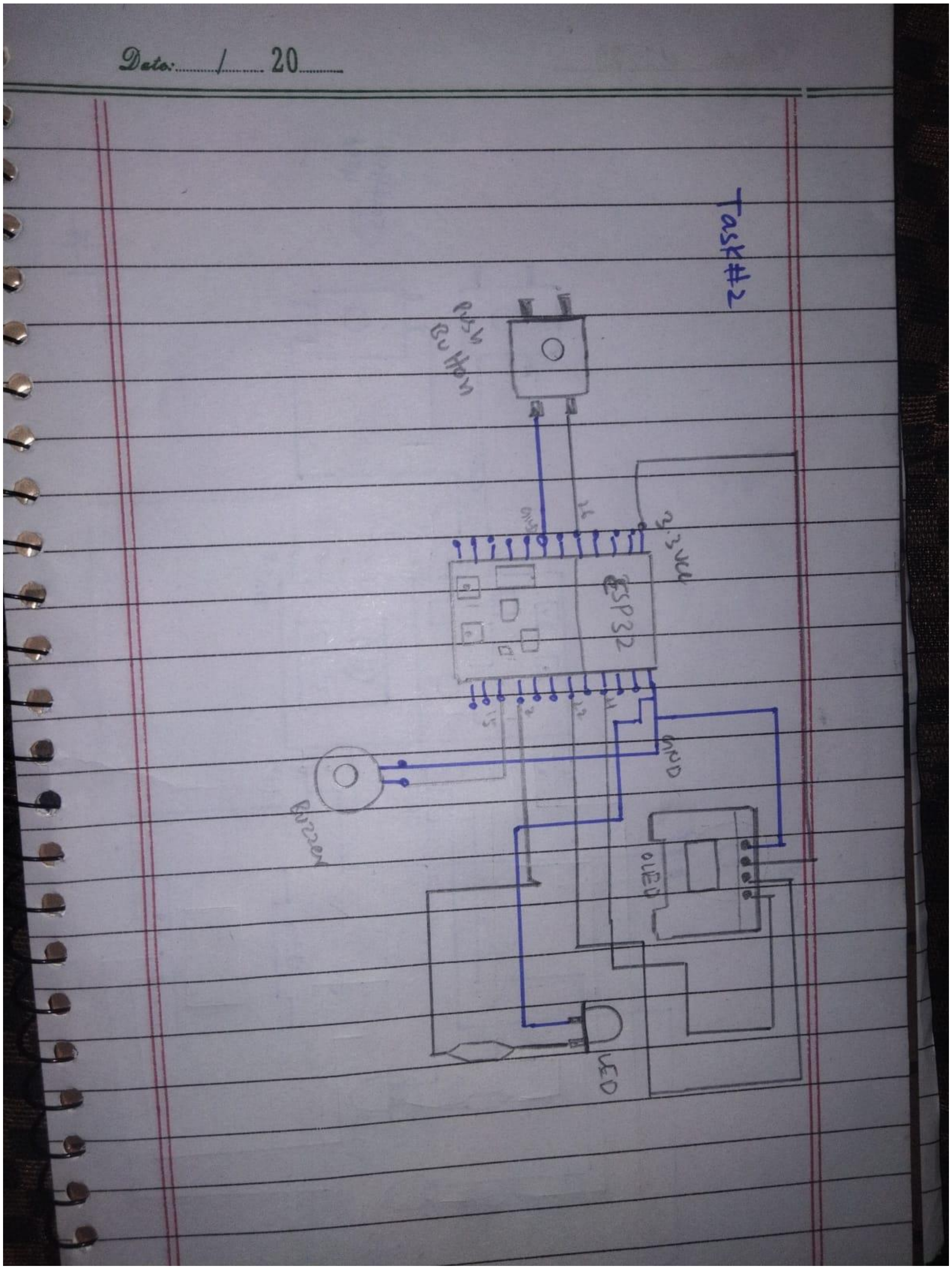
```
if (digitalRead(BTN)== HIGH && Pressed){
unsigned long duration= millis() -presstime;
 pressed = false;

if (duration >1500){
tone(BUZZER,1000,500);
showText("Long Press → Buzzer");
}

else {
 ledState = !LedState;
 digitalWrite(LED, ledState);
 showText ("Short Press →LED Toggle");
   }
 }
}
```

Handmade Hardware:

Task 01


Date: / 20
Task#1
Push buttons
GND
26
27
3.3Vcc
ESP32
15
2
4
5
GND
GND
OLED
Buzzer
LED1
LED2
LED3

Task 02


Date: ..../....... 20.......
Task#2
Push Button
GND
26
3.3Vcc
ESP32
15
2
22
21
GND
Buzzer
OLED
LED

Wokwi Link:

Task 01: https://wokwi.com/projects/445735049640047617

Task 02: https://wokwi.com/projects/445792788940889089

GitHub Repo:

Link: https://github.com/abdullah-061317/CS-B-5th-23-NTU-CS-1056-Embedded-IoT.git