IOT DOMAIN ANALYST

LAB TASK-4

Name: SREENIVASAN S

Reg No: 21BEC0256

ESP32 Bluetooth Interfacing for led Control (Arduino IDE)

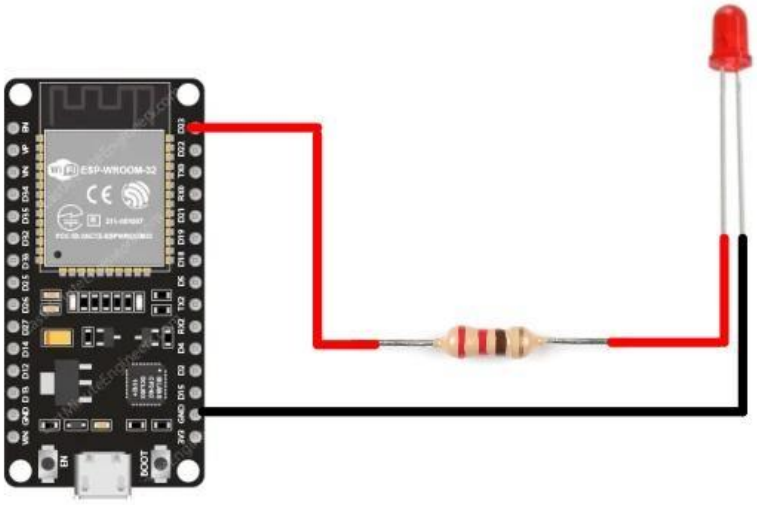
Aim:

Build a smart living room using ESP 32 and control the lights in your room using mobile phone enabled with Bluetooth protocol and use ESP32 Bluetooth Classic with Arduino IDE to exchange data between an ESP32 and an Android smartphone.

Apparatus Required:

- ESP32 development board
- Android Smartphone with Bluetooth
- 5mm LED
- 220 Ohm resistor
- Jumper wires
- Breadboard

Circuit Diagram



Procedure

- Connect an LED Anode to GPIO23, ESP32 GND pin to LED Cathode
- You need a Bluetooth Terminal application installed in your smartphone.
- Android app "Serial Bluetooth Terminal" available in the Play Store.
- Open your Arduino IDE, and go to **Sketch** **Include library** Manage **library** BluetoothSerial.h Install
- Upload the code to the ESP32. Make sure you have the right board and COM port selected.
- Go to your smartphone and open the "**Serial Bluetooth Terminal**" app. Make sure you've enable your smartphone's Bluetooth.
- To connect to the ESP32 for the first time, you need to pair a new device. Go to **Devices.**
- Click the settings icon, and select **Pair new device**. You should get a list with the available Bluetooth devices, including the **ESP32_IoTlab**,Pair with the **ESP32_IoTlab**.
- Then, go back to the Serial Bluetooth Terminal. Click the icon at the top to connect to the ESP32. You should get a "**Connected"** message.
- Then, you can write the **"a"** and **"b"** messages to control the LED.

- When the ESP32 receives the **“a”** message, we’ll turn the LED on, when it receives the **“b”** message, we’ll turn the LED off.

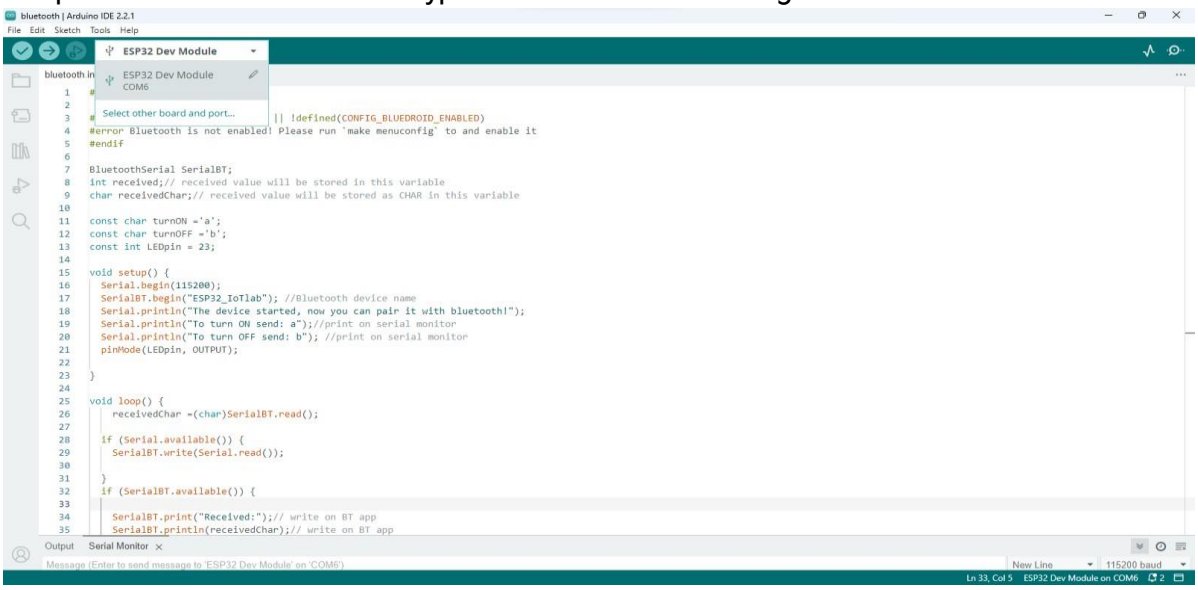
Code:

```
#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and
enable it #endif

BluetoothSerial SerialBT;
int received;// received value will be stored in this variable
char receivedChar;// received value will be stored as CHAR in this variable

const char turnON ='a';
const char turnOFF ='b';
const int LEDpin = 23;

void setup() {
Serial.begin(115200);
  SerialBT.begin("21BEC0256ESP32_IoTlab"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");
  Serial.println("To turn ON send: a");//print on serial monitor
Serial.println("To turn OFF send: b"); //print on serial monitor
  pinMode(LEDpin, OUTPUT);

}

void loop() {
    receivedChar =(char)SerialBT.read();

  if (Serial.available()) {
SerialBT.write(Serial.read());

}
  if (SerialBT.available()) {

    SerialBT.print("Received:");// write on BT app
    SerialBT.println(receivedChar);// write on BT app
    Serial.print ("Received:");//print on serial monitor
    Serial.println(receivedChar);//print on serial monitor
```

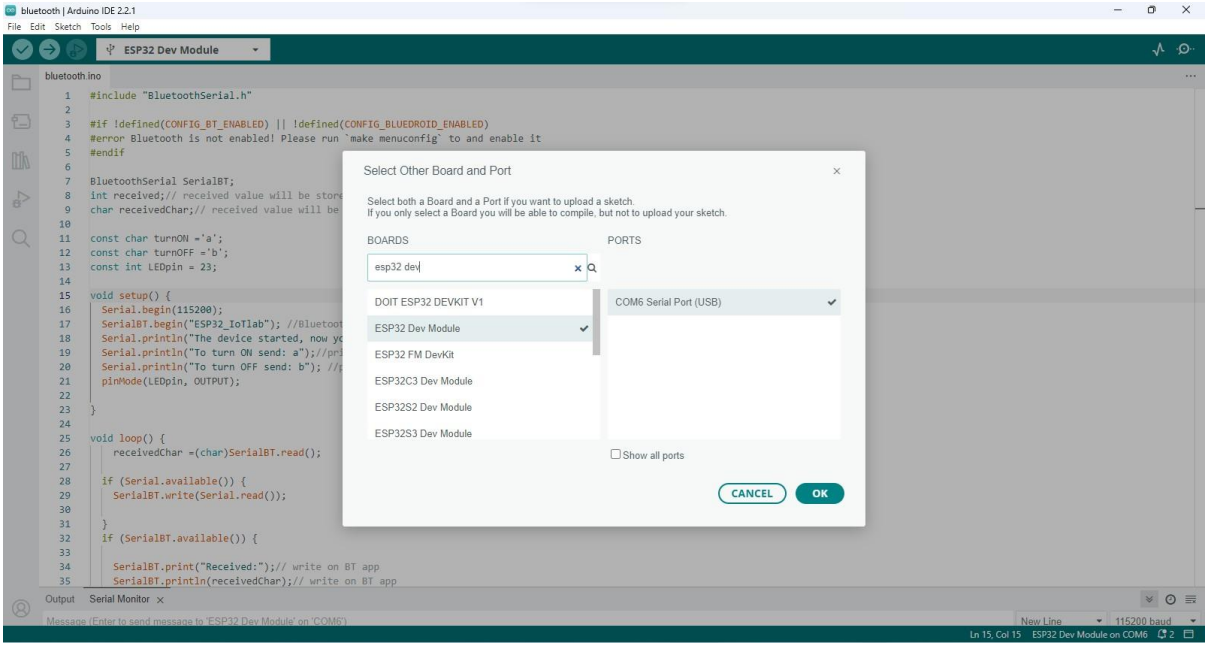
```
    //SerialBT.println(receivedChar);//print on the app
//SerialBT.write(receivedChar); //print on serial monitor
if(receivedChar == turnON)
    {
     SerialBT.println("LED ON:");// write on BT app
Serial.println("LED ON:");//write on serial monitor
     digitalWrite(LEDpin, HIGH);// turn the LED ON

    }
    if(receivedChar == turnOFF)
    {
     SerialBT.println("LED OFF:");// write on BT app
Serial.println("LED OFF:");//write on serial monitor
      digitalWrite(LEDpin, LOW);// turn the LED off
    }
       }
delay(10);
}
```

Steps:

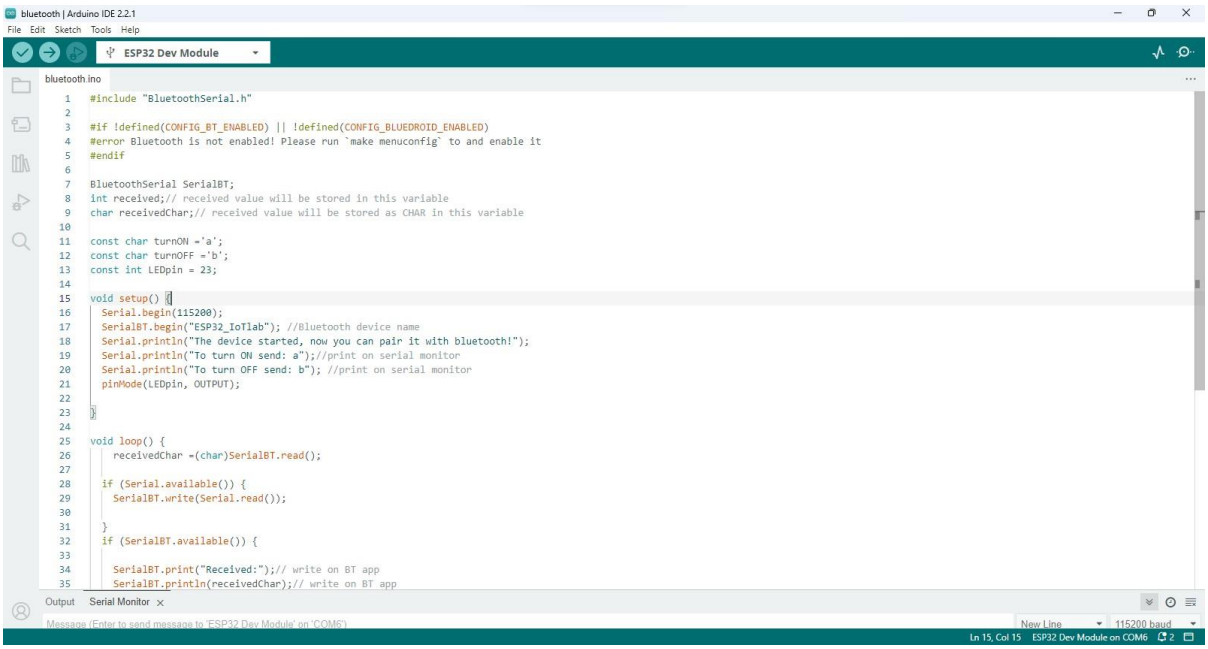
1. Open Arduino ide software ->Type the code ->Board Manager



Type ->Select ESP32 Dev Module ->Select Port

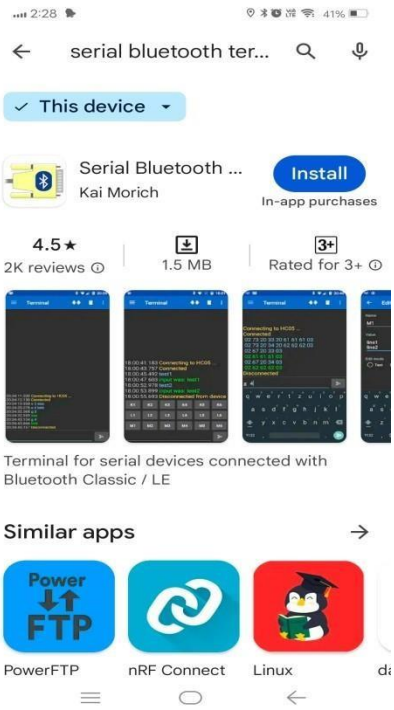


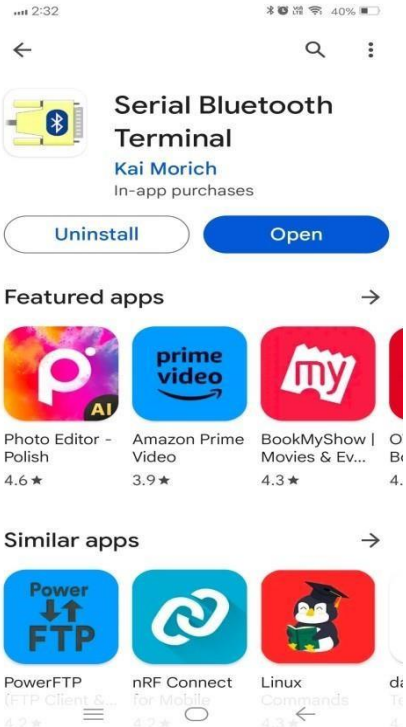
Change the Bluetooth Device name ->Verify and Upload



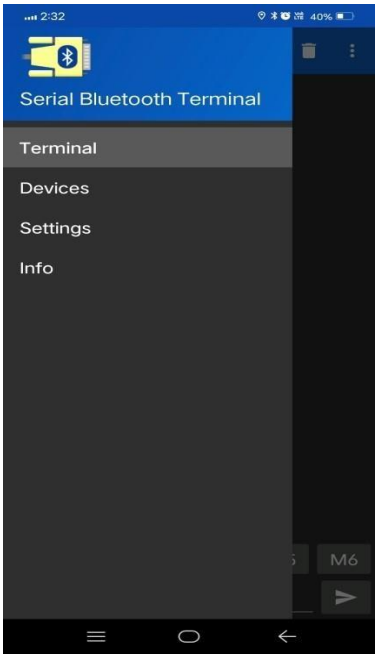
Step 1: Android Smartphone ->Bluetooth ->Pair the ESP32_IoTlab

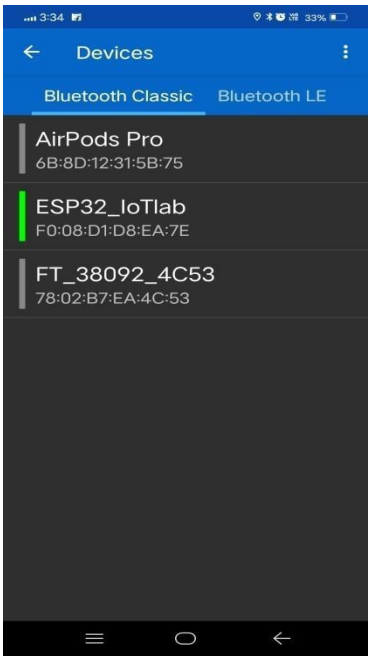
Step 2: Play store ->Serial Bluetooth terminal->Install the App ->open

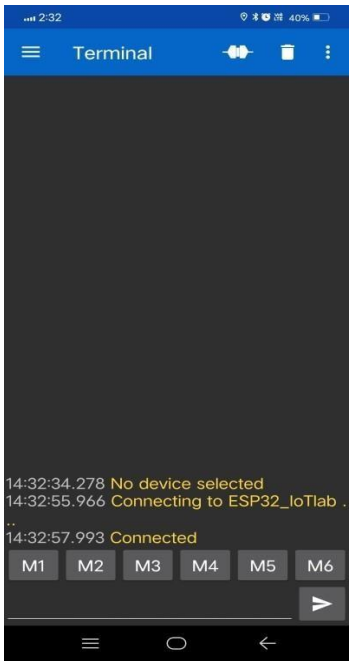




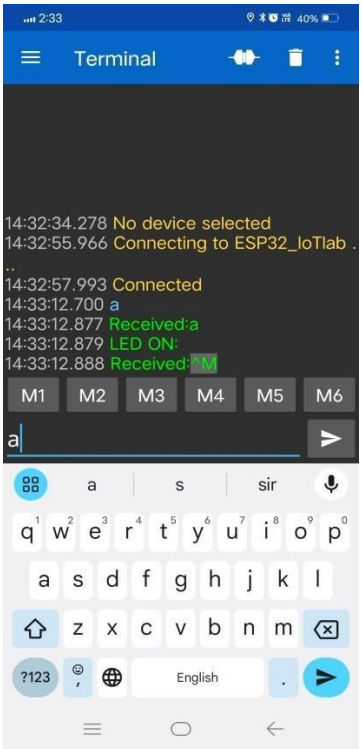
Terminal ->Devcies ->Select Bluetooth Device_>Connected



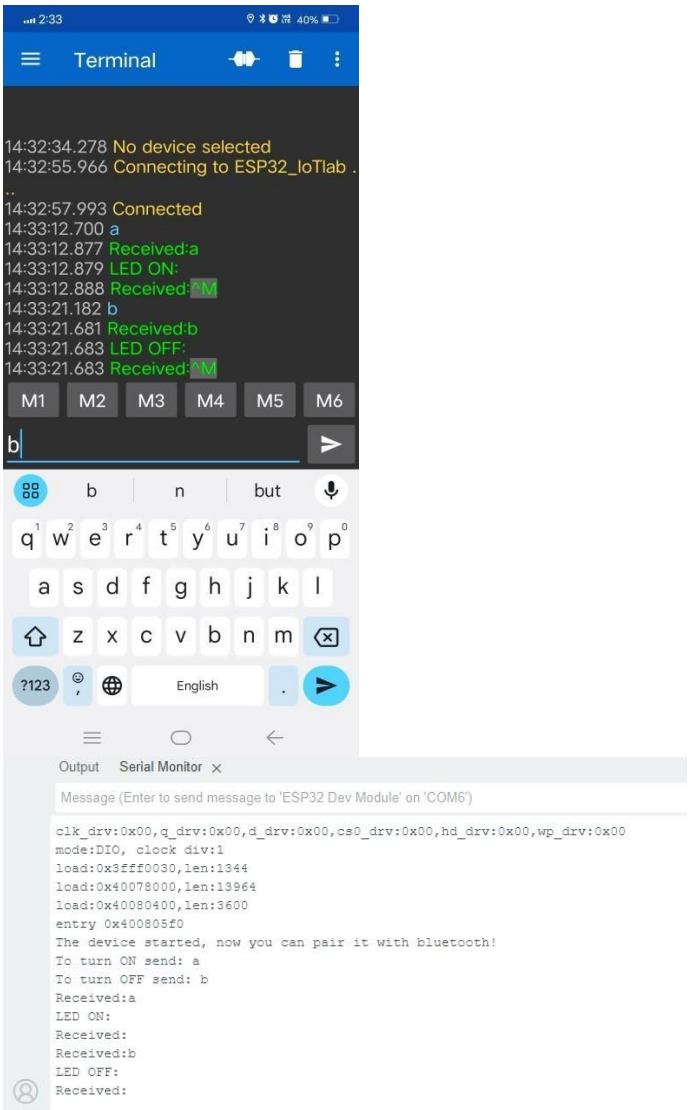




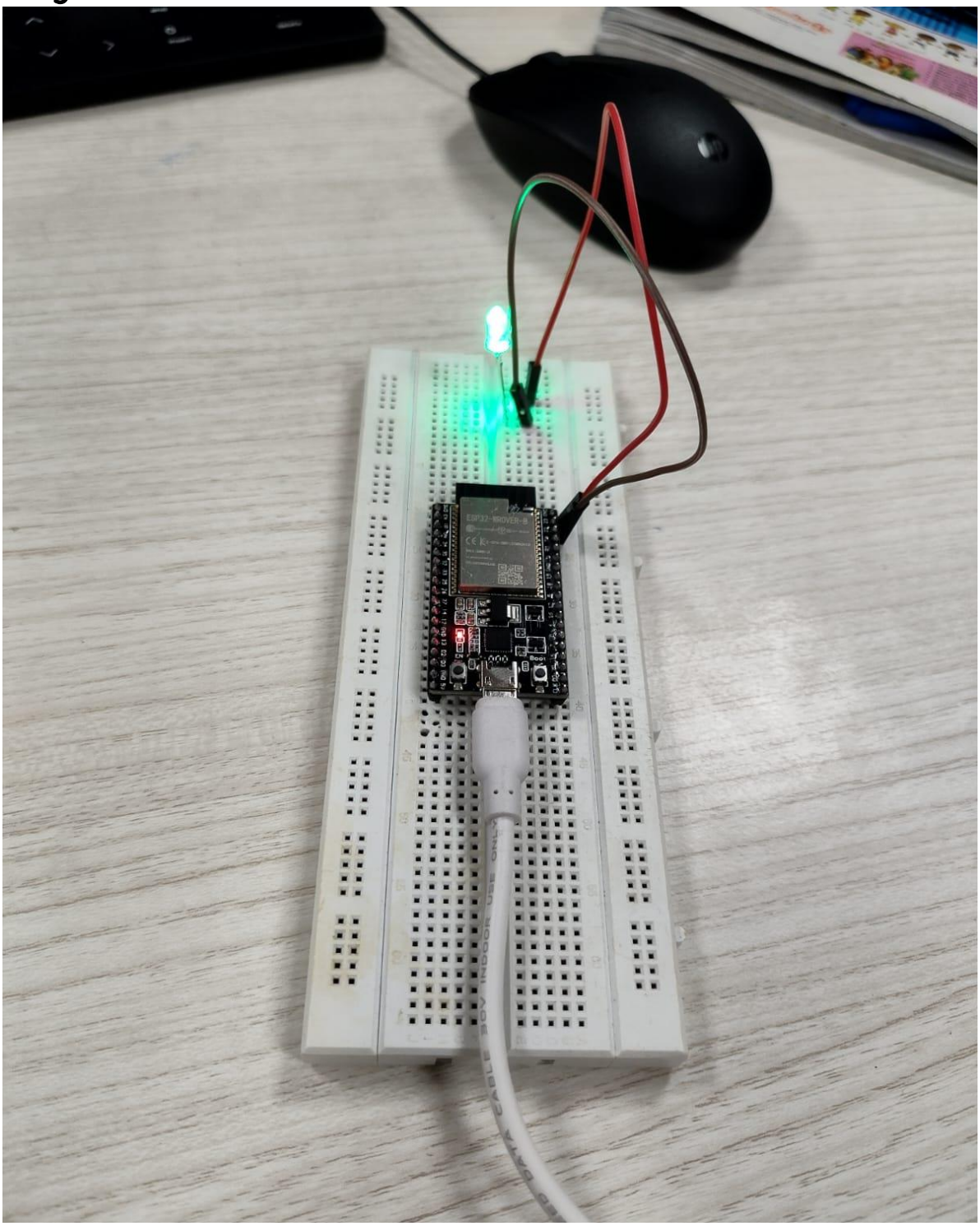
Then, you can write the **“a”** Send ->ESP32 Received “a” ->Serial Monitor LED ON

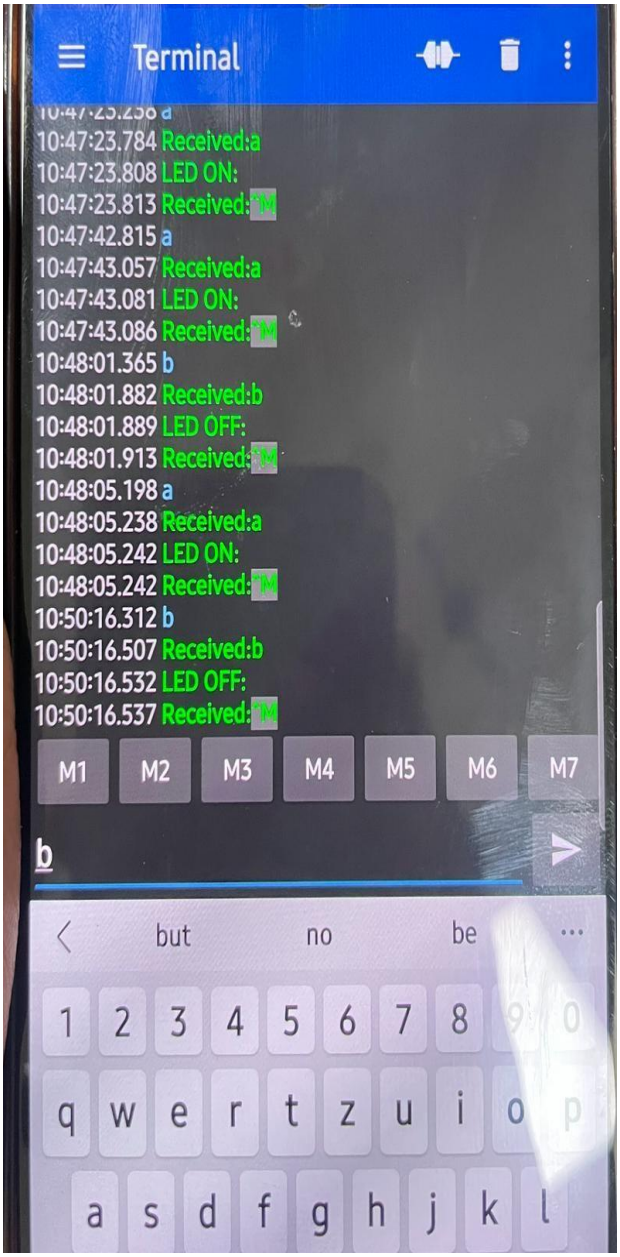


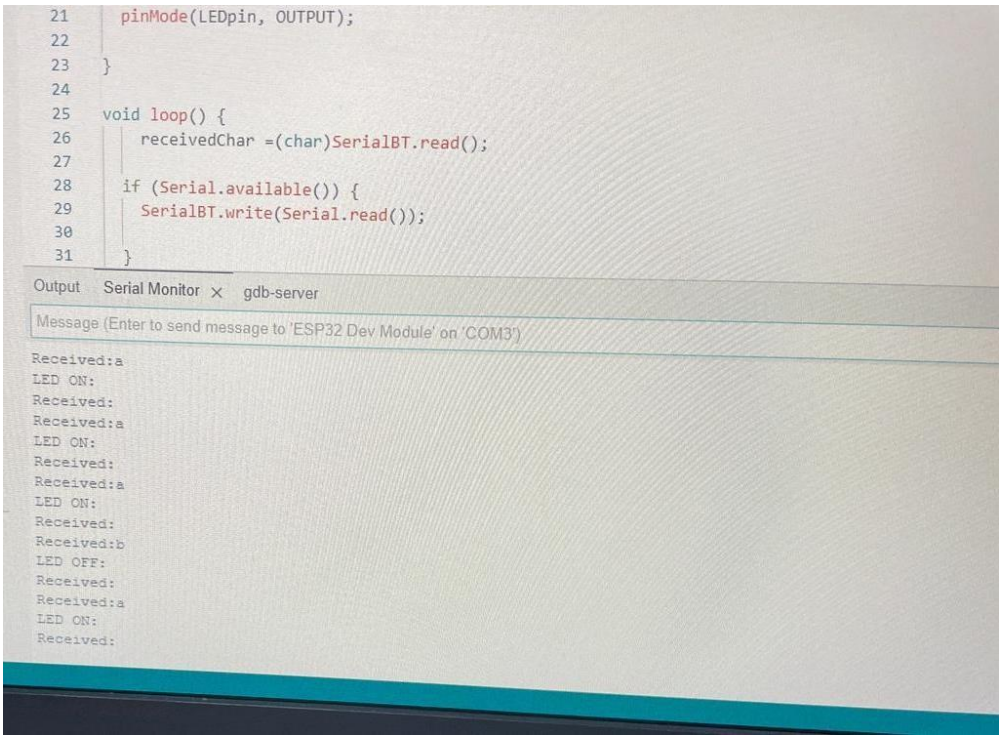
Then, you can write the **“b”** Send ->ESP32 Received “a” -> Serial Monitor LED OFF



Images:




Terminal
10:47:23.784 Received:a
10:47:23.808 LED ON:
10:47:23.813 Received:M
10:47:42.815 a
10:47:43.057 Received:a
10:47:43.081 LED ON:
10:47:43.086 Received:M
10:48:01.365 b
10:48:01.882 Received:b
10:48:01.889 LED OFF:
10:48:01.913 Received:M
10:48:05.198 a
10:48:05.238 Received:a
10:48:05.242 LED ON:
10:48:05.242 Received:M
10:50:16.312 b
10:50:16.507 Received:b
10:50:16.532 LED OFF:
10:50:16.537 Received:M
M1
M2
M3
M4
M5
M6
M7
b
but
no
be
1 2 3 4 5 6 7 8 9 0
q w e r t z u i o p
a s d f g h j k l

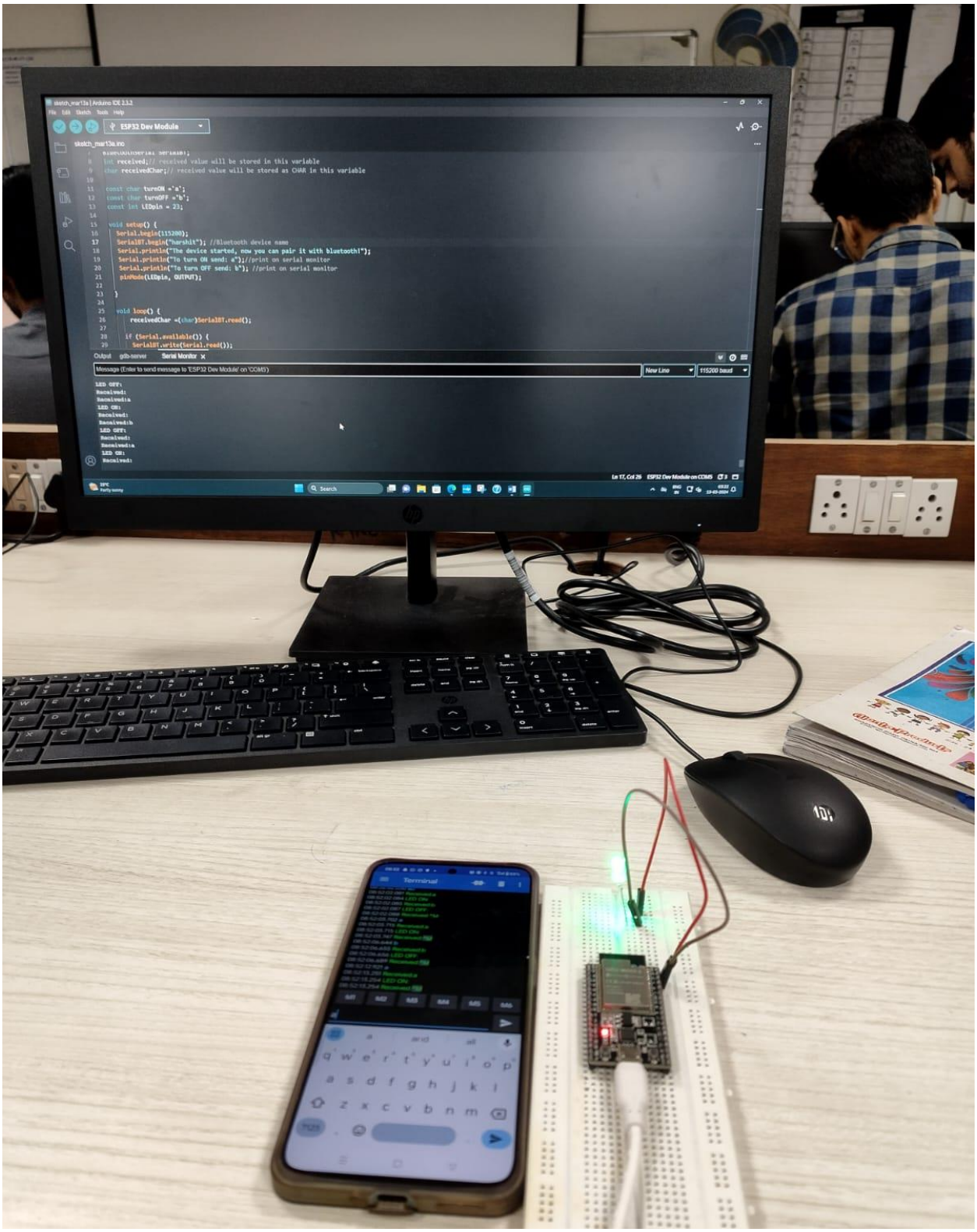


```
21    pinMode(LEDpin, OUTPUT);
22
23  }
24
25  void loop() {
26      receivedChar =(char)SerialBT.read();
27
28    if (Serial.available()) {
29      SerialBT.write(Serial.read());
30
31    }
```

Output | Serial Monitor × | gdb-server

Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')

```
Received:a
LED ON:
Received:
Received:a
LED ON:
Received:
Received:a
LED ON:
Received:
Received:b
LED OFF:
Received:
Received:a
LED ON:
Received:
```


sketch_mar13a | Arduino IDE 2.1.1
ESP32 Dev Module
sketch_mar13a.ino
int received;// received value will be stored in this variable
char receivedChar;// received value will be stored as CHAR in this variable
const char turnON ='a';
const char turnOFF ='b';
const int LEDpin = 23;
void setup() {
Serial.begin(115200);
SerialBT.begin("harshit"); //Bluetooth device name
Serial.println("The device started, now you can pair it with bluetooth!");
Serial.println("To turn ON send: a");//print on serial monitor
Serial.println("To turn OFF send: b"); //print on serial monitor
pinMode(LEDpin, OUTPUT);
}
void loop() {
receivedChar =(char)SerialBT.read();
if (Serial.available()) {
SerialBT.write(Serial.read());
Output
gdb-server
Serial Monitor
Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')
New Line
115200 baud
LED OFF:
Received:
Received:a
LED ON:
Received:
Received:b
LED OFF:
Received:
Received:a
LED ON:
Received:
Ln 17, Col 26
ESP32 Dev Module on COM5
Search
Terminal

SREENIVASA05
21BEC0256

18/03/24

IoT Lab

ESP32 Bluetooth Interfacing for led Control (Arduino IDE)

Recalls

* while typing 'a' in mobile, led is turned ON.
* while typing 'b' in mobile, led is turned OFF.

Code:

```
#include "BluetoothSerial.h"
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run 'make menuconfig' to and enable it
#endif

BluetoothSerial SerialBT;
int received; // received value will be stored in this variable
char receivedChar; //received value will be stored as CHAR in this variable

const char turnON = 'a';
const char turnOFF = 'b';
const int LEDpin = 23;

void setup() {
Serial.begin(115200);
SerialBT.begin("RegnoESP32_IoTLab");
Serial.println("The device started, now you can pair it with bluetooth!");
Serial.println("To turn ON send: a");
Serial.println("To turn OFF send: b");
pinMode(LEDpin, OUTPUT);
}
```

```
void loop() {
receivedChar = (char) SerialBT.read();
if (Serial.available()) {
   SerialBT.write(Serial.read());
}
if (SerialBT.available()) {
  SerialBT.print("Received:");
  SerialBT.print(receivedChar);
  Serial.print("Received:");
  Serial.println(receivedChar);
// SerialBT.println(receivedChar);
// SerialBT.write(receivedChar);
if (receivedChar == turnON) {
   SerialBT.println("LED ON:");
   Serial.println("LED ON:");
   digitalWrite(LEDpin, HIGH);
}
if (receivedChar == turnOFF) {
     SerialBT.println("LED OFF:");
     Serial.println("LED OFF:");
     digitalWrite(LEDpin, LOW);
   }
}
  delay(100);
}
```