A blue text on a white background

Description automatically generatedA black and grey logo with a graduation cap and book

Description automatically generated

**GOVERNMENT OF TAMILNADU**

DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

**IoT AND ITS APPLICATIONS**

**HOST INSTITUTION**

xxxx

COIMBATORE – 04

**TRAINING PARTNER**

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

|  |  |
| --- | --- |
| **NAME** | **ROLL NO** |
| Name 1 | Roll no 1 |
| Name 2 | Roll no 2 |
| Name 3 | Roll no 3 |
| Name 4 | Roll no 4 |
| Name 5 | Roll no 5 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1 | ABSTRACT | 1 |
| 2 | INTRODUCTION | 2 |
| 3 | HARDWARE AND SOFTWARE REQUIREMENTS | 3 |
| 4 | BLOCK DIAGRAM | 8 |
| 5 | CODE | 9 |
| 6 | OUTPUT RESULTS | 11 |
| 7 | CONCLUSION | 13 |

**ABSTRACT**

Device-to-device (D2D) wireless communication using Bluetooth technology enables direct data exchange between devices without the need for intermediary infrastructure like routers or base stations. Bluetooth, a widely adopted short-range wireless communication protocol, supports a variety of applications, including file transfer, audio streaming, and device synchronization. This technology is energy-efficient, operates in the 2.4 GHz ISM band, and can connect multiple devices simultaneously in a piconet. D2D communication via Bluetooth enhances user experience by offering a convenient, low-power solution for connecting a wide range of devices in proximity. This abstract summarizes the fundamental aspects and advantages of Bluetooth-based D2D communication.

1

**INTRODUCTION**

Bluetooth is a short-range wireless communication technology that allows devices to exchange data over short distances without the need for cables. It operates in the 2.4 GHz ISM band and uses frequency-hopping spread spectrum to minimize interference. Bluetooth is commonly used for connecting peripherals like headphones, keyboards, and mice to devices such as smartphones, laptops, and tablets. It supports a range of up to 100 meters, depending on the device class, and offers low power consumption, making it ideal for portable devices.

2

**HARDWARE AND SOFTWARE REQUIREMENTS**

**HARDWARE REQUIREMENT**

**1.ESP32 Microcontroller**

**2 .LED**

**3.Jumper Wires**

**4.USB Cable**

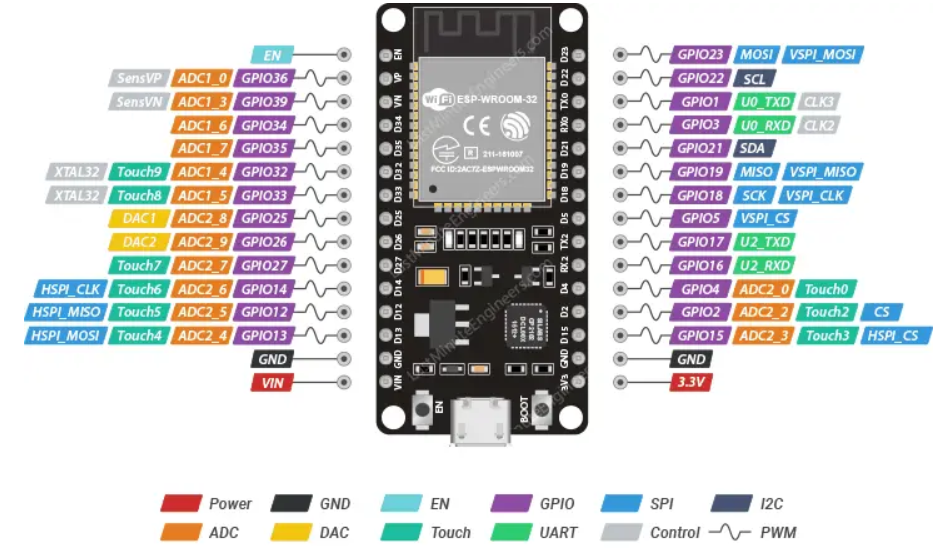
**SOFTWARE REQUIREMENT**

**1.Arduino IDE**

3

**ESP32 Microcontroller**

The ESP32 is a highly versatile microcontroller developed by Espressif Systems, designed for a wide range of applications, particularly in the Internet of Things (IoT) space. It is renowned for its combination of high performance, integrated wireless connectivity, and a rich set of features, all at a low cost. The ESP32 is commonly used in projects that require both Wi-Fi and Bluetooth capabilities, making it suitable for smart home devices, sensor networks, and wearable technology.



4

**LED**

An LED (Light Emitting Diode) is a semiconductor light source that emits light when an electric current passes through it. It is one of the most energy-efficient and long-lasting lighting technologies available. LEDs are widely used in various applications, from simple indicators to complex displays, due to their small size, low power consumption, and durability. In the context of an anti-theft security system, the LED serves as a visual indicator to signal the detection of motion. When the PIR sensor detects movement, the ESP32 microcontroller sends a signal to the LED, causing it to light up. This visual alert can help in quickly identifying the presence of an intruder or unauthorized access in the monitored area.The LED’s immediate response and visibility make it an effective component for such security systems, providing an instant indication of a security breach.

****

5

**Jumper Wires**

Jumper wires are essential in this project, used to connect the ESP32 microcontroller to the components on the breadboard. These wires provide a flexible and reliable way to link the microcontroller’s GPIO pins to the LEDs, resistors, and other circuit elements, enabling proper signal and power flow. Their ease of use allows for quick modifications and testing during the prototyping stage.

****

**USB Cable**

The USB cable is a critical tool in this project, used to connect the ESP32 microcontroller to a computer for power supply, programming, and debugging. It enables the transfer of code and data between the development environment and the microcontroller, facilitating the upload of firmware and real-time communication during the development process. The USB connection also allows for serial monitoring, providing valuable insights into the system's performance and behavior.



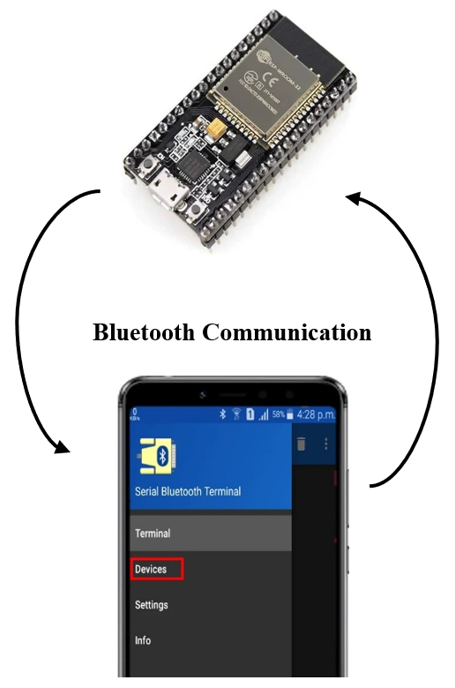
6

**Arduino IDE**

The Arduino IDE is a user-friendly platform that allows for easy programming and uploading of code to Arduino boards. In a device-to-device communication project, it enables seamless coding and management of serial communication between devices, particularly when using Bluetooth modules. Its integrated development environment simplifies debugging and testing, making it ideal for wireless communication projects.

7

**BLOCK DIAGRAM**



8

**CODE**

#include "BluetoothSerial.h" // Include the Bluetooth Serial library

BluetoothSerial SerialBT; // Create a Bluetooth Serial object

const int ledPin = 2; // Pin where the LED is connected

void setup() {

pinMode(ledPin, OUTPUT); // Set the LED pin as an output

Serial.begin(115200); // Start the serial communication

SerialBT.begin("ESP32\_LED"); // Start Bluetooth with the name "ESP32\_LED"

Serial.println("Bluetooth device is ready to pair");

}

void loop() {

if (SerialBT.available()) { // Check if data is available to read

char incomingChar = SerialBT.read(); // Read the incoming data

Serial.print("Received: ");

Serial.println(incomingChar);

if (incomingChar == '1') { // If '1' is received

digitalWrite(ledPin, HIGH); // Turn the LED on

Serial.println("LED ON");

} else if (incomingChar == '0') { // If '0' is received

digitalWrite(ledPin, LOW); // Turn the LED off

9

Serial.println("LED OFF");

}

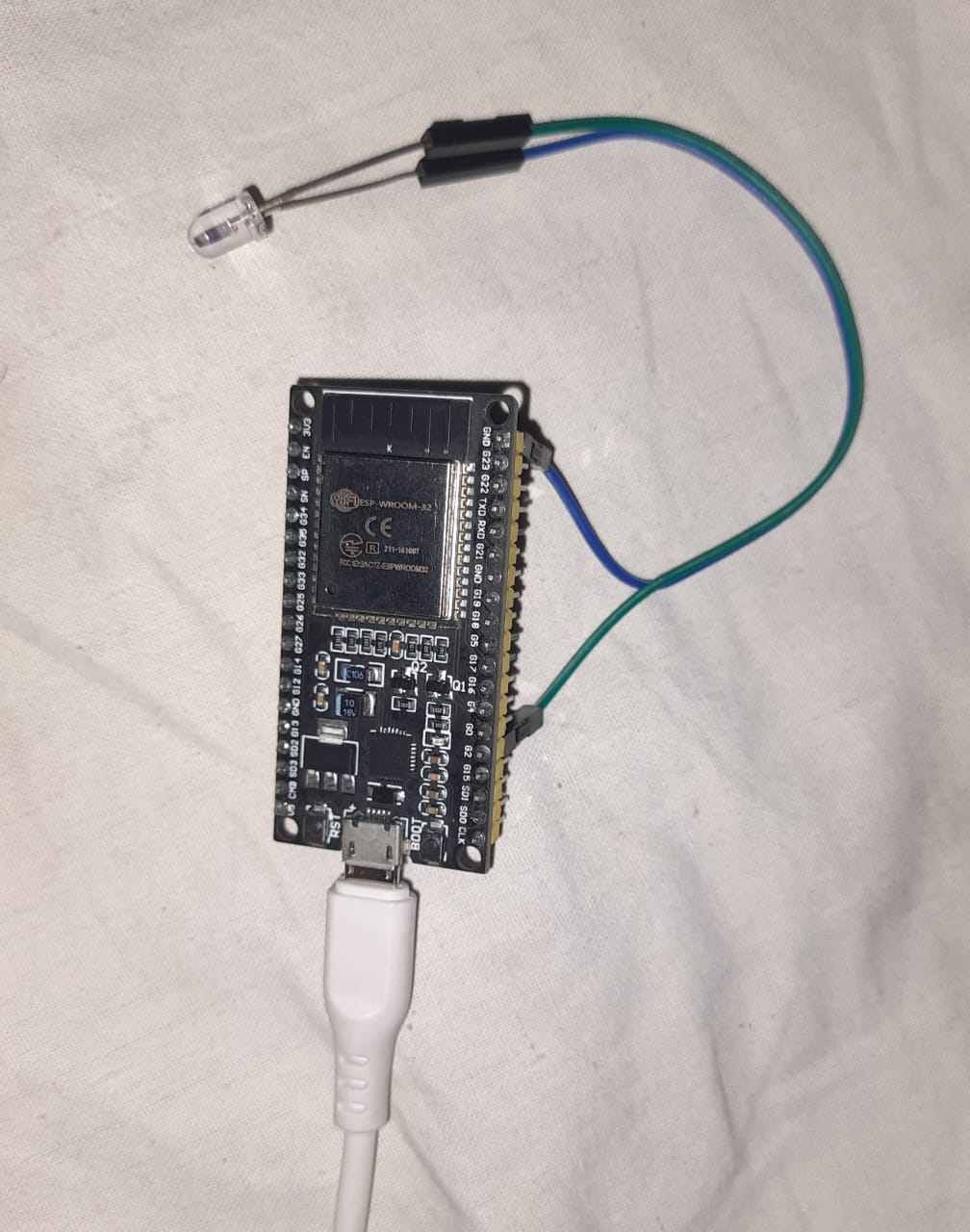
}

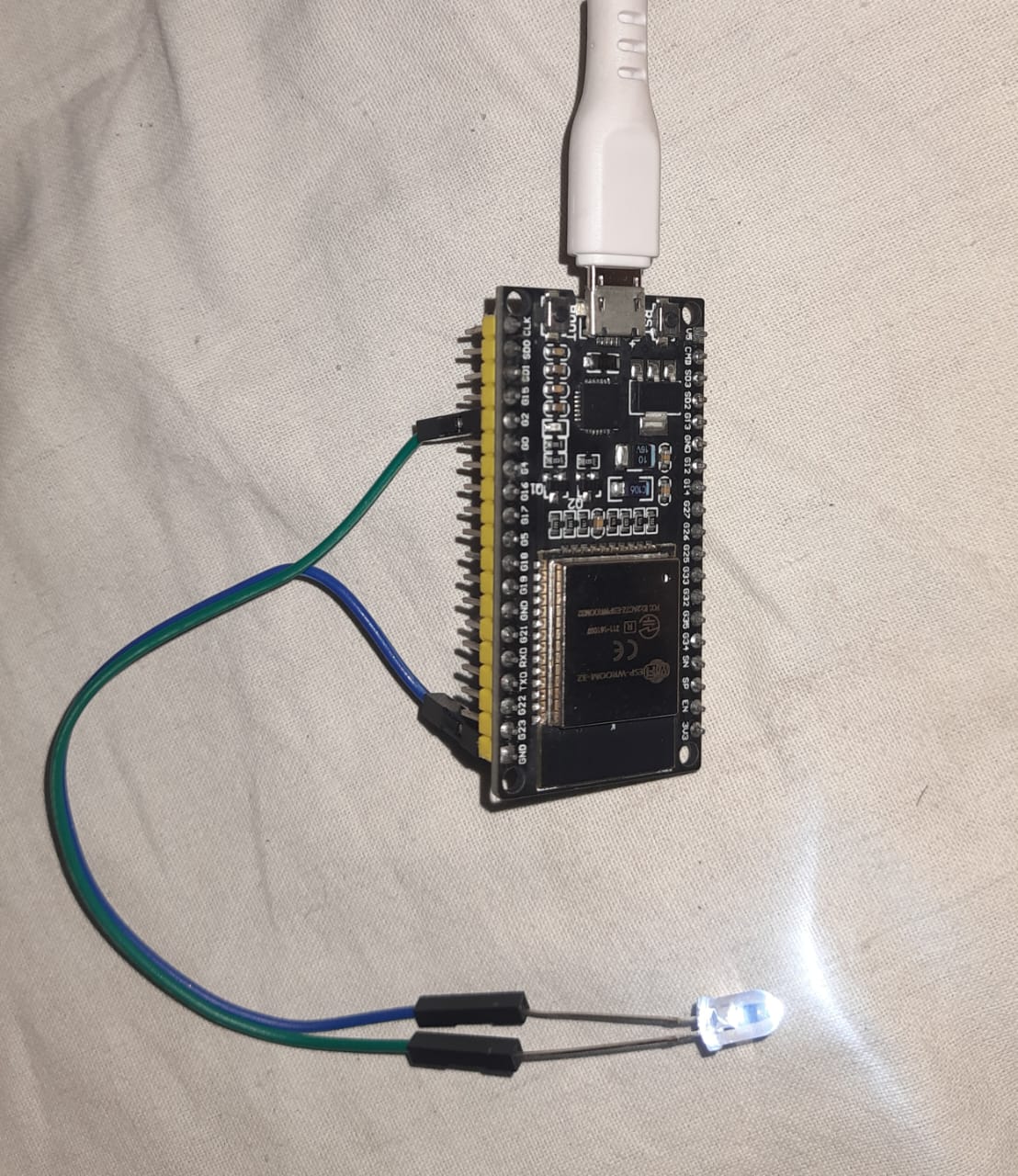
delay(1000); // Small delay to debounce

}

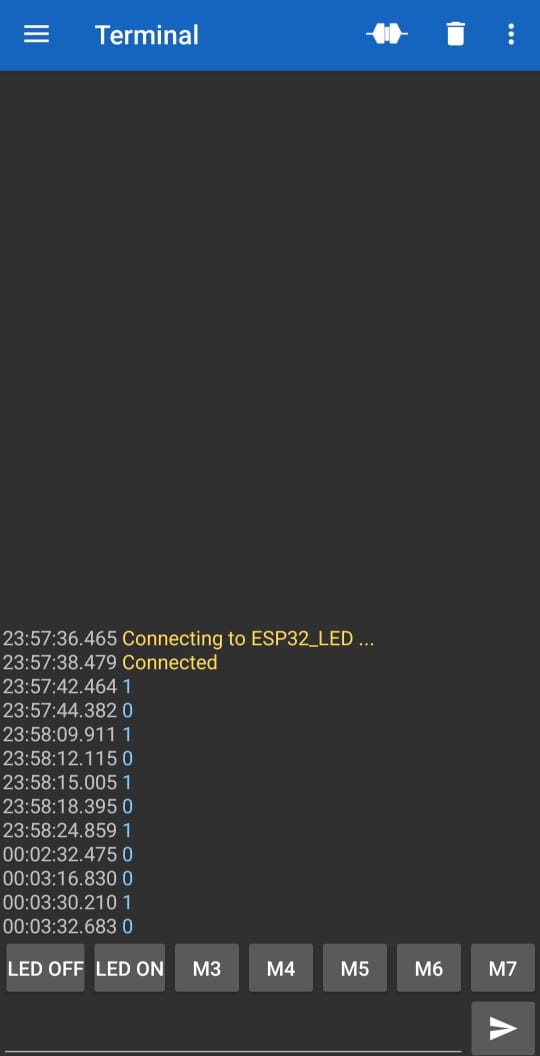
10

**OUTPUT RESULTS**





11

****

12

**CONCLUSION**

The Bluetooth-controlled LED project using the ESP32 was successfully completed, demonstrating effective wireless control capabilities. The ESP32 was configured as a Bluetooth server to receive on/off commands, reliably toggling the LED's state in response. This project showcased the ESP32's versatility and efficient Bluetooth communication, providing a solid foundation for more advanced applications and further exploration into IoT solutions.

13