Instituto Tecnológico®
de Aguascalientes

INSTITUTO TECNOLOGICO DE AGUASCALIENTES

Integrante:

Manuel Humberto Reyes Salas

María José Guerrero Rios

Jesús Eli Sánchez Ruvalcaba

Actividad: Practica 1

Materia: Tecnologías inalámbricas

Docente: RICARDO ALEJANDRO RODRIGUEZ JIMENEZ

Fecha: 19 de Febrero del 2026

Placa ESP-32

Es un microcontrolador creado por Espressif Systems. Es muy usado en proyectos de IoT porque integra WiFi y Bluetooth, además de ser potente y económico.

El ESP32 funciona como el "cerebro" de un sistema electrónico. Ejecuta un programa (firmware) que tú cargas desde tu computadora (usualmente con Arduino IDE o ESP-IDF).

¿Cómo funciona?

Se alimenta (3.3V o 5V según la placa).

Ejecuta un programa cargado desde tu computadora.

Lee entradas (sensores, botones).

Procesa la información.

Controla salidas (LEDs, motores, WiFi, Bluetooth).

Partes principales: 1. Procesador: Doble núcleo hasta 240 MHz, 2. Memoria: Flash (programa), SRAM (trabajo), ROM (arranque), 3. Conectividad: WiFi + Bluetooth (BLE), 3. Periféricos: ADC, DAC, PWM, SPI, I2C, UART, timers, 4. Modos de energía: Activo, Sleep y Deep Sleep (bajo consumo).

El ESP32 tiene aproximadamente 30-36 pines programables, llamados GPIO.

¿Para qué se usa?

- Domótica
- Monitoreo remoto
- Robots
- Control de motores
- Servidores web
- Proyectos IoT

En pocas palabras

El ESP32 es un microcontrolador potente, económico y con conexión inalámbrica integrada, ideal para proyectos inteligentes y conectados a internet.

Módulo Relé 5V 2 Canales

Un módulo relé 5V de 2 canales es un dispositivo electrónico que permite controlar circuitos de alto voltaje o corriente (como luces, motores o electrodomésticos) mediante señales de bajo voltaje provenientes de microcontroladores como Arduino o Raspberry Pi. Cuenta con dos relés independientes, cada uno capaz de conmutar cargas de hasta 10A a 250V AC o 30V DC.

Características Principales

- Voltaje de operación: 5V DC
- Tipo de relé: 2 canales SPDT (1 polo, 2 tiros)
- Señal de control: Lógica TTL (3.3V o 5V), activado por nivel bajo (LOW) en la mayoría de modelos
- Aislamiento: Optoacoplador para proteger el microcontrolador
- Indicadores LED: Uno por canal y otro de alimentación
- Compatibilidad: Arduino, ESP32, Raspberry Pi, PIC, entre otros

Practica

Explicación de la practica

En esta práctica desarrollamos un sistema de control inalámbrico mediante tecnología BLE, cuyo objetivo fue energizar y desenergizar una carga eléctrica (foco) utilizando un módulo relevador controlado por una placa ESP32 desde una aplicación móvil creada en MIT App Inventor.

El propósito principal fue integrar conocimientos de electrónica, programación y comunicación inalámbrica para diseñar un sistema funcional que permitiera controlar una carga de mayor potencia de manera remota y segura. Este tipo de aplicaciones es común en sistemas de automatización y domótica, donde se requiere encender o apagar dispositivos eléctricos sin necesidad de contacto directo.

Para el desarrollo del sistema se utilizó una placa ESP32, la cual funciona como el cerebro del proyecto. Este microcontrolador cuenta con conectividad Bluetooth integrada, lo que permitió configurarlo como servidor BLE. Al energizar el dispositivo, el programa cargado previamente desde el entorno Arduino IDE inicia la comunicación.

Posteriormente, se desarrolló una aplicación móvil en MIT App Inventor utilizando la extensión Bluetooth BLE. La aplicación fue diseñada con una interfaz sencilla que permite buscar dispositivos cercanos, mostrar una lista de los dispositivos encontrados y establecer la conexión con el ESP32. Una vez que la conexión se realiza correctamente, la interfaz cambia su estado y habilita los botones de control.

Cuando el usuario presiona el botón “ON”, la aplicación envía una cadena de texto al ESP32 mediante BLE. El microcontrolador interpreta el mensaje recibido y activa el pin configurado

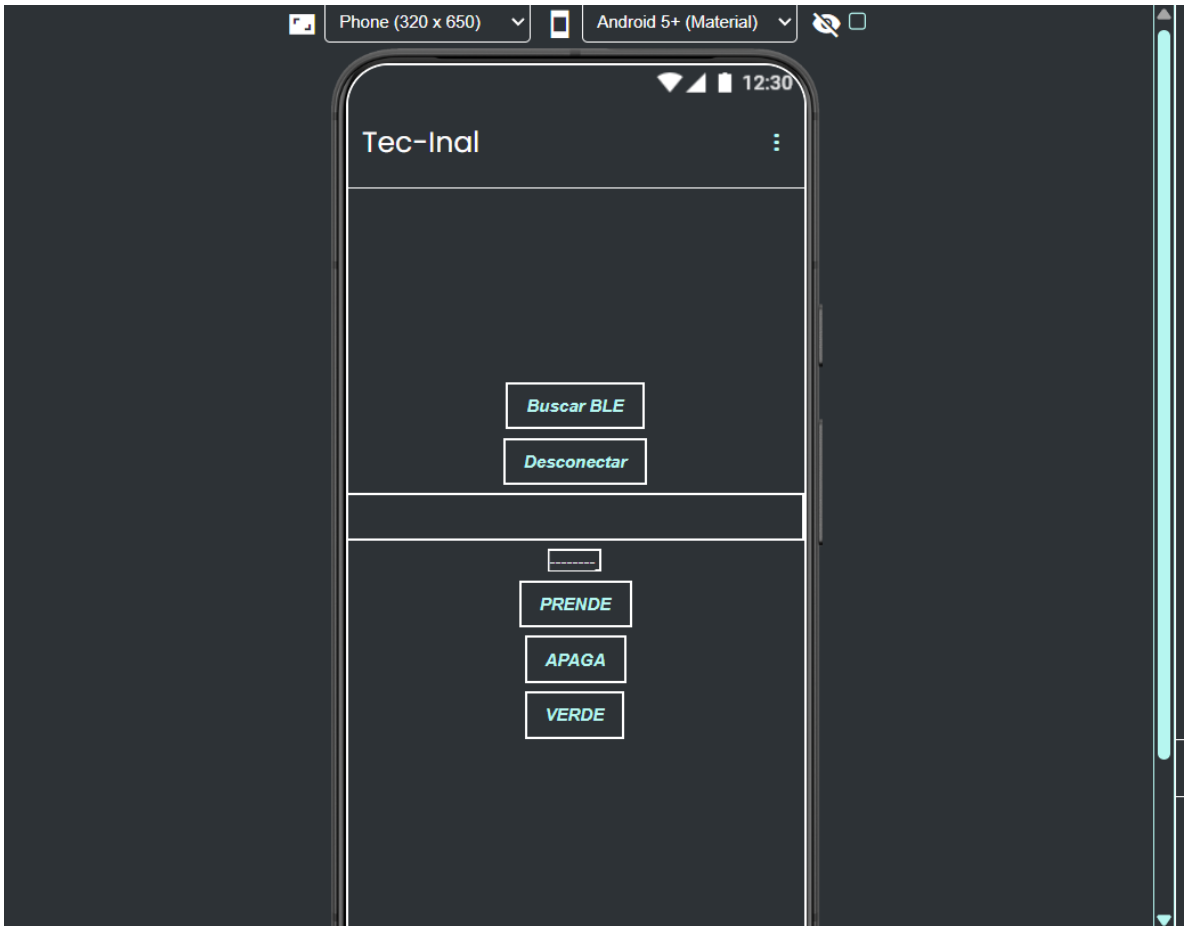
como salida digital, energizando el módulo relevador. Al cerrarse el contacto interno del relevador, se completa el circuito de corriente alterna y el foco enciende. De manera similar, al presionar el botón “OFF”, el ESP32 desactiva el pin, el relevador abre el circuito y el foco se apaga.

Durante las pruebas realizadas, se verificó que la comunicación entre la aplicación móvil y el ESP32 fuera estable, rápida y confiable. Se comprobó que cada comando enviado desde el teléfono era recibido correctamente por el microcontrolador, el cual respondía activando o desactivando la carga de manera inmediata.

Con esta práctica se logró demostrar el correcto funcionamiento de la comunicación inalámbrica BLE aplicada a un sistema real de control eléctrico, integrando hardware y software en un proyecto funcional que puede ser aplicado.

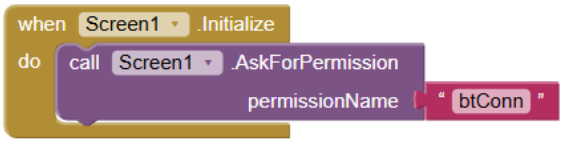
Código de bloques de app inventor

Diseño:

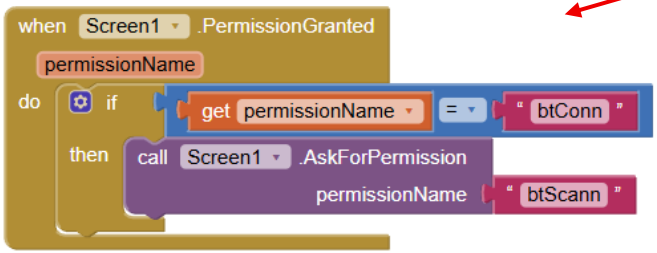


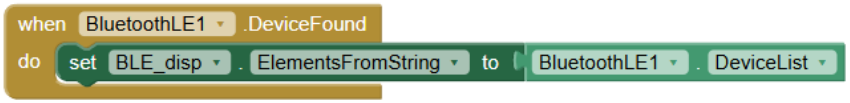
Código en bloques

En estos primeros bloques lo que estamos haciendo es que cuando se inicie la aplicación pida permiso para poder usar el bluetooth y después pedir permiso para conectarse a dispositivos bluettoth cercanos.

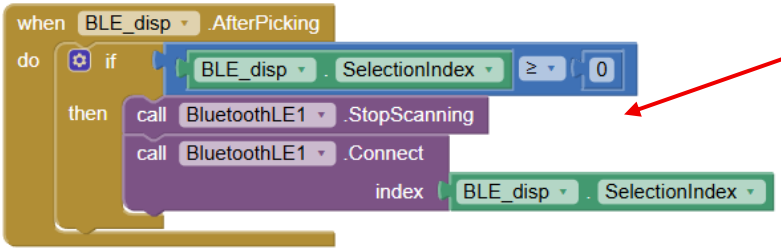


En esta parte lo que realiza es que todos los dispositivos bluetooth que encuentre los va agregando a una lista

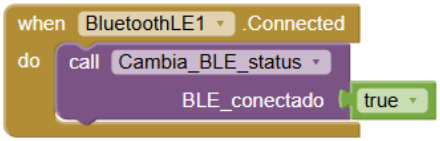




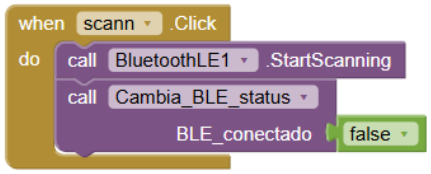
Lo que realiza es que verifica si se selecciono algo, si esto es verdadero deja de escanear y se conecta al dispositivo seleccionado



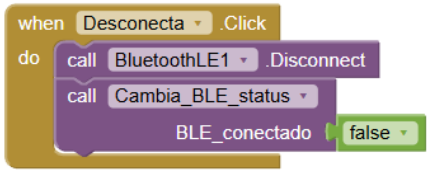
Cuando la conexión fue exitosa el estado cambia para indicar que esta conectado y manda llamar un método

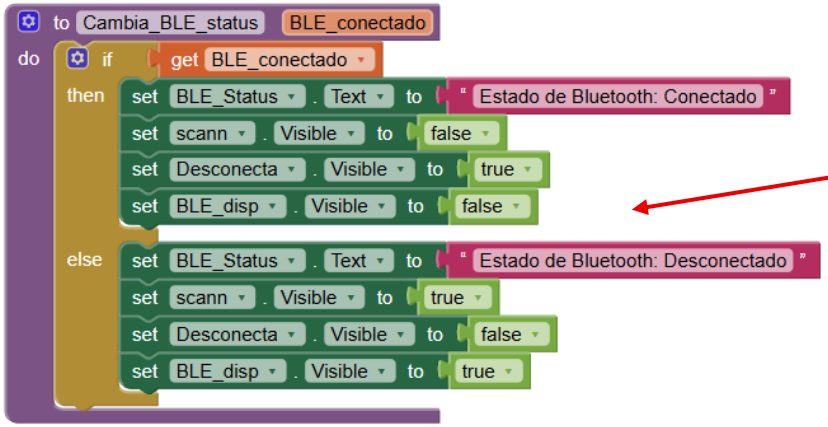


Cuando se presiona el botón de desconecta, este cierra la conexión bluetooth y lo pone en estado de desconectado



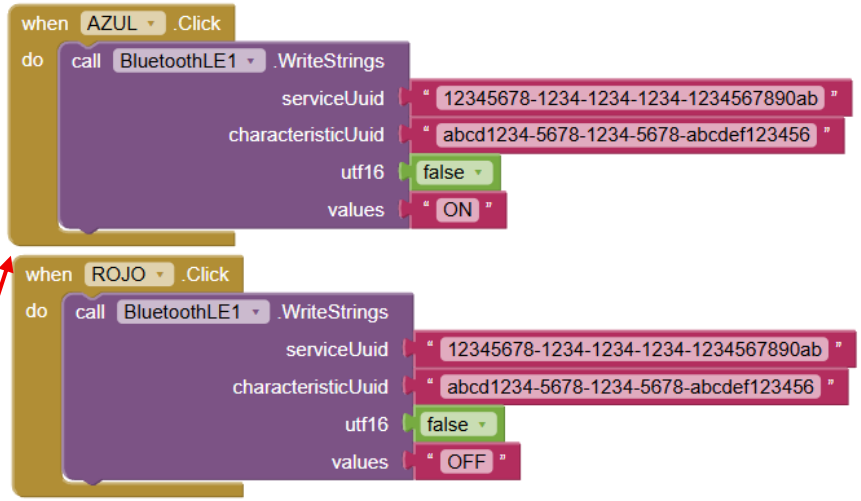
En estos bloques lo que realiza es que al momento de presionar el botón de scann, empieza a buscar dispositivos bluetooth cercanos y pone el estado en desconectado mientras busca





Este es el bloque mas importante, lo primero que realiza es que recibe el parámetro si esta conectado o no si está conectado lo que realiza es que muestra un mensaje de conectado y oculta el botón de scann y la lista de dispositivos cercanos, además de que muestra el boton para desconectar.

Si recibe el parámetro que esta desconectado lo que realiza es que muestra el mensaje de desconectado y permite volver a escanear dispositivos y oculta el botón de desconectar.



En estos bloques lo que estamos realizando es que estamos enviando comandos al dispositivo bluetooth, cuando presionamos el botón Azul la aplicación envia el texto ON a el dispositivo y este comando funciona para prender un led, mientras que por el otro lado si presionamos el botón Rojo realiza lo contrario y la aplicación manda el texto OFF a el dispositivo y este comando funciona para apagar el led.

Código de Arduino

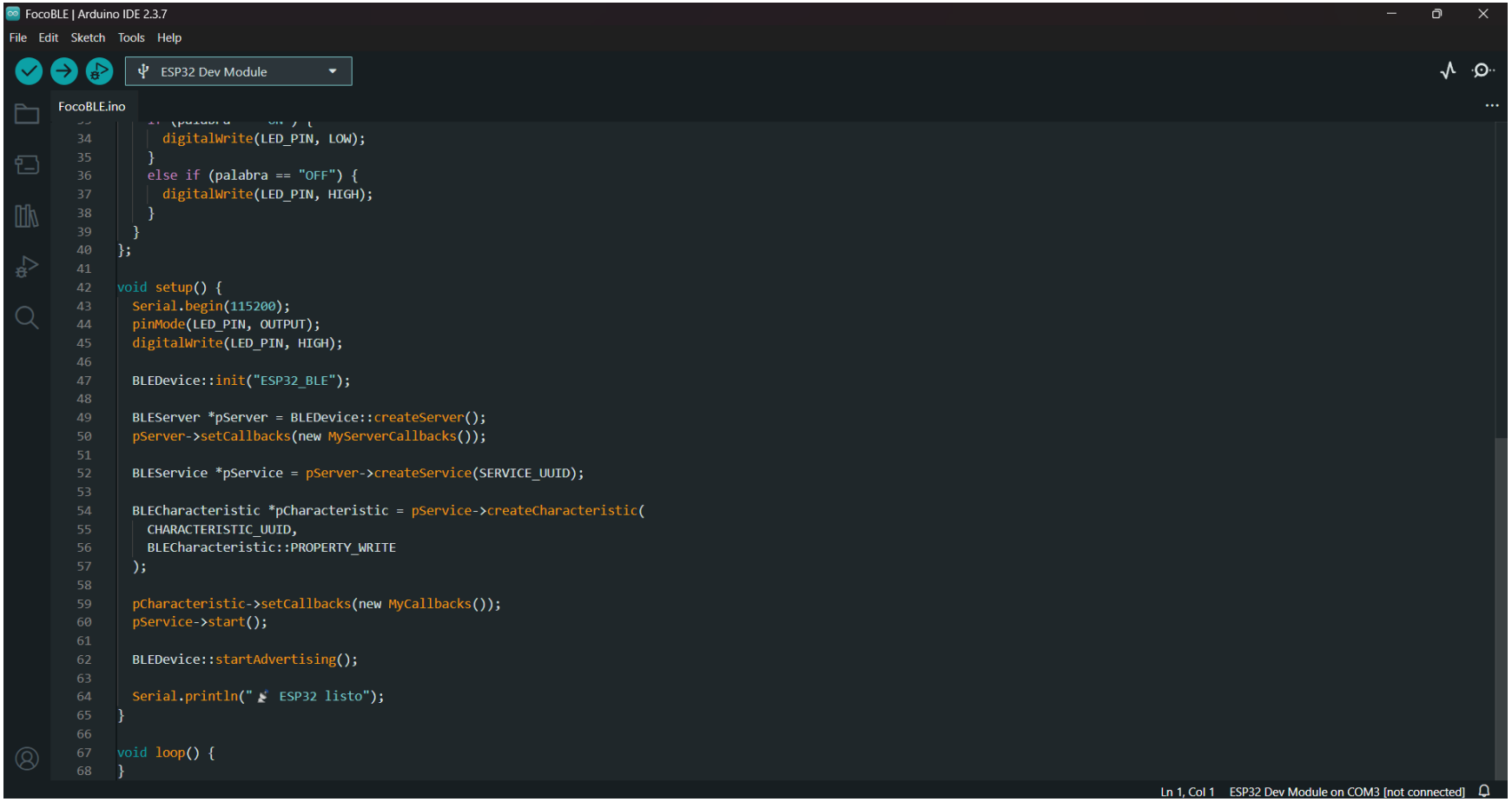
FocoBLE | Arduino IDE 2.3.7

File Edit Sketch Tools Help

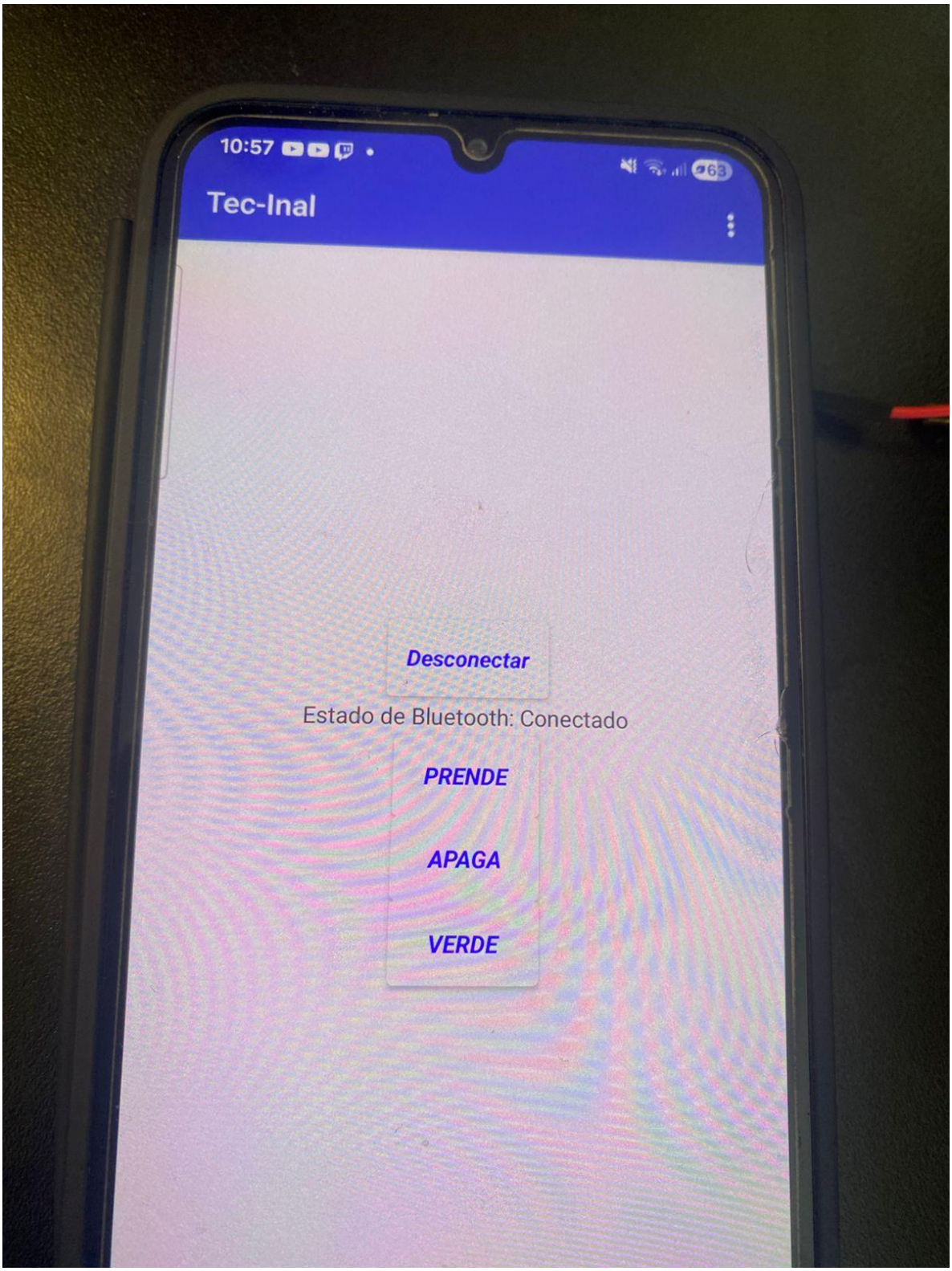
ESP32 Dev Module

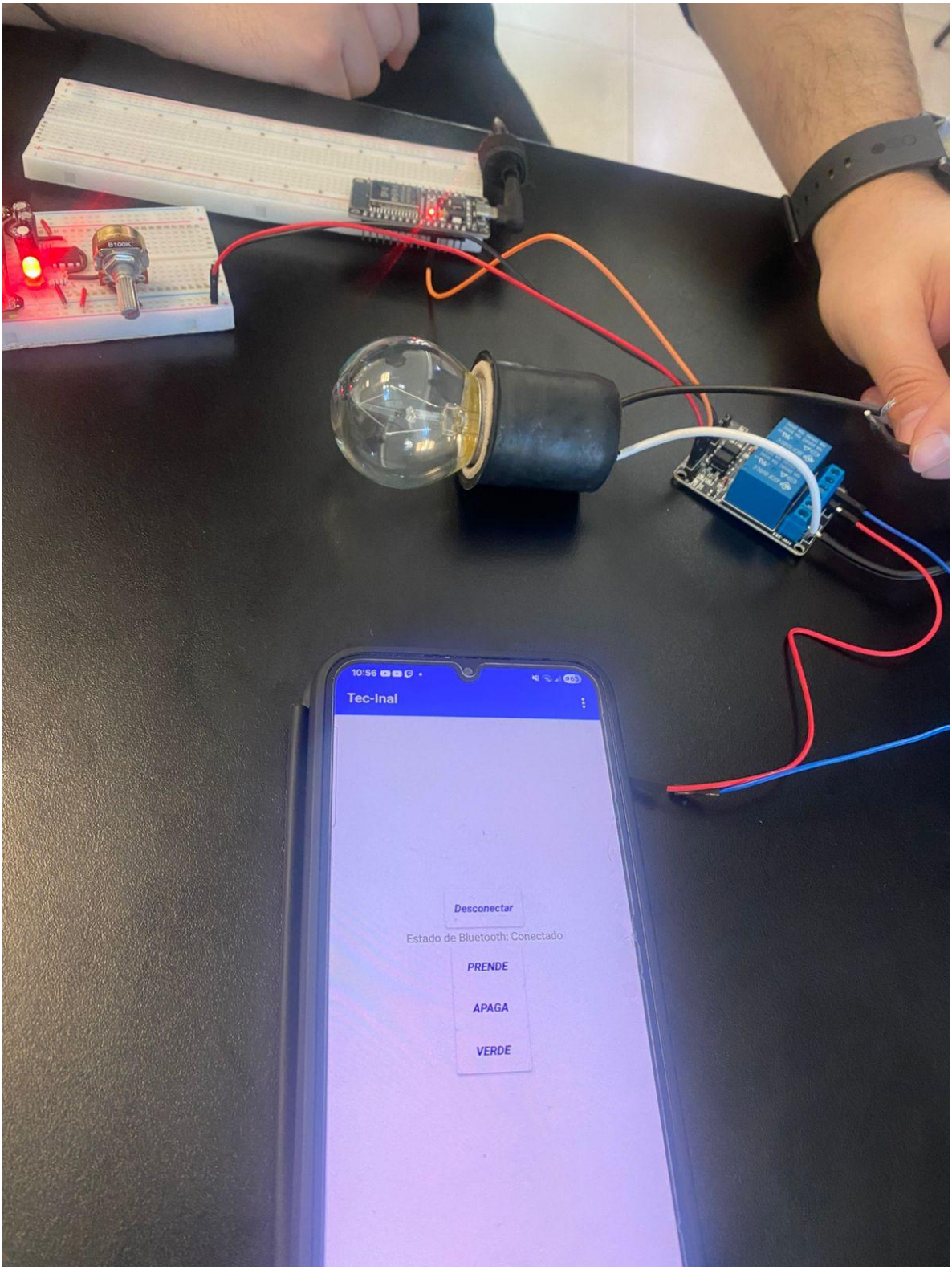
FocoBLE.ino

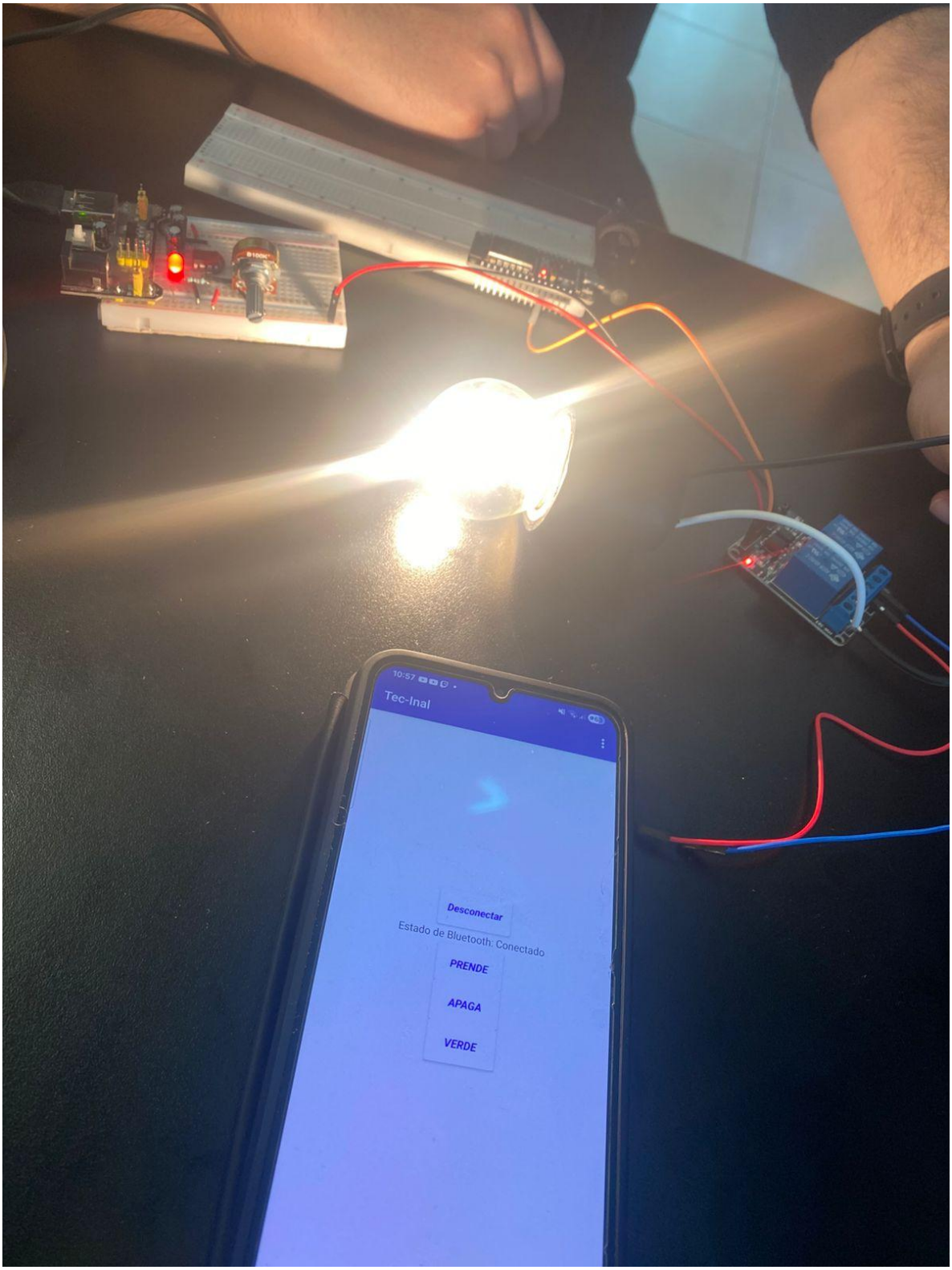
```
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>

#define SERVICE_UUID        "12345678-1234-1234-1234-1234567890ab"
#define CHARACTERISTIC_UUID "abcd1234-5678-1234-5678-abcdef123456"
#define LED_PIN 14   // normalmente es el correcto

bool dispositivoConectado = false;

class MyServerCallbacks : public BLEServerCallbacks {
  void onConnect(BLEServer* pServer) {
    dispositivoConectado = true;
    Serial.println("📲 Conectado");
  }

  void onDisconnect(BLEServer* pServer) {
    dispositivoConectado = false;
    Serial.println("❌ Desconectado");
    BLEDevice::startAdvertising();
  }
};

class MyCallbacks : public BLECharacteristicCallbacks {
  void onWrite(BLECharacteristic* pCharacteristic) {

    String palabra = pCharacteristic->getValue();
    palabra.trim();

    Serial.print("📩 Palabra recibida: ");
    Serial.println(palabra);

    if (palabra == "ON") {
      digitalWrite(LED_PIN, LOW);
    }
```

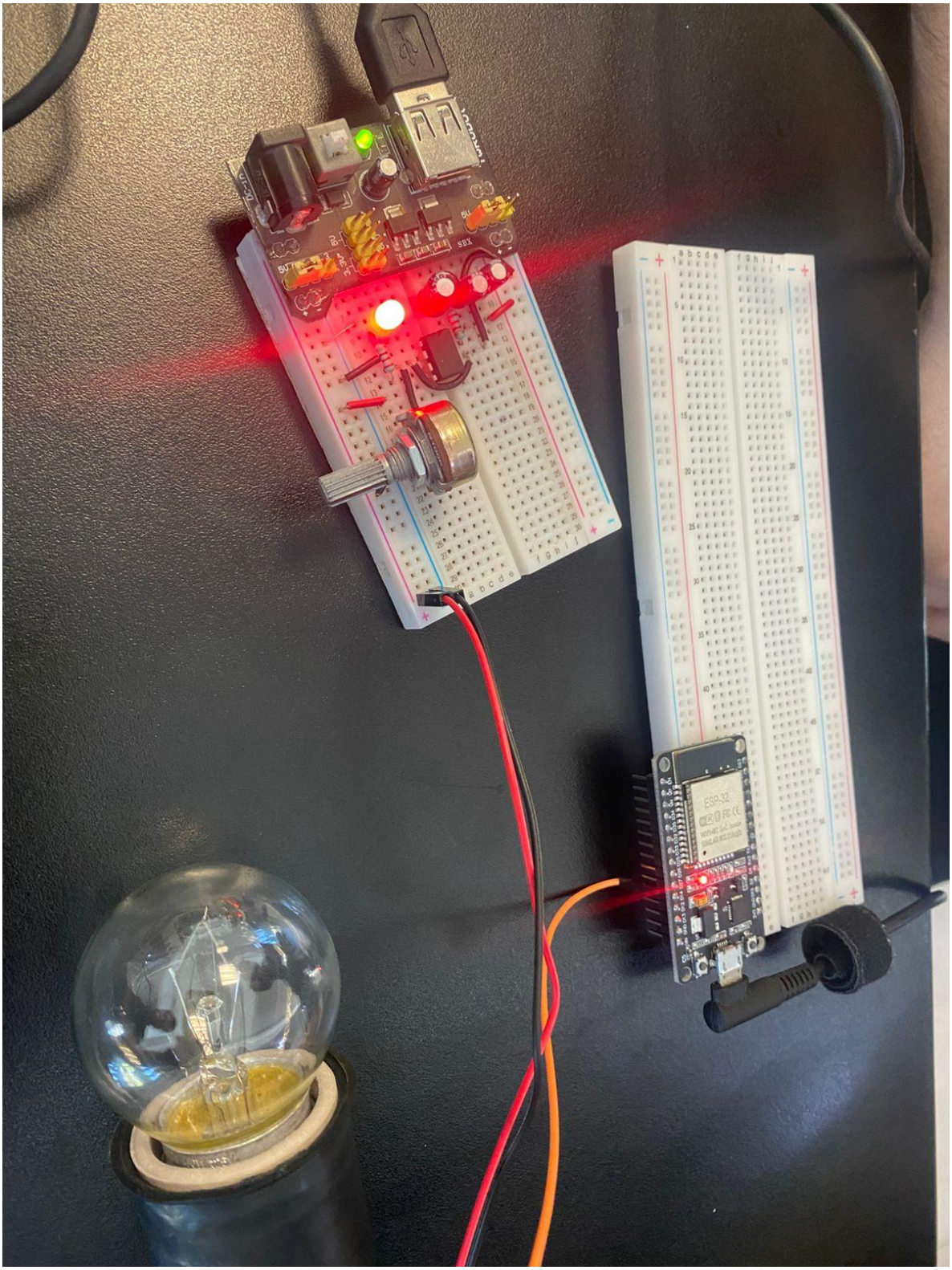

FocoBLE | Arduino IDE 2.3.7
File Edit Sketch Tools Help
ESP32 Dev Module
FocoBLE.ino
digitalWrite(LED_PIN, LOW);
}
else if (palabra == "OFF") {
digitalWrite(LED_PIN, HIGH);
}
}
};
void setup() {
Serial.begin(115200);
pinMode(LED_PIN, OUTPUT);
digitalWrite(LED_PIN, HIGH);
BLEDevice::init("ESP32_BLE");
BLEServer *pServer = BLEDevice::createServer();
pServer->setCallbacks(new MyServerCallbacks());
BLEService *pService = pServer->createService(SERVICE_UUID);
BLECharacteristic *pCharacteristic = pService->createCharacteristic(
CHARACTERISTIC_UUID,
BLECharacteristic::PROPERTY_WRITE
);
pCharacteristic->setCallbacks(new MyCallbacks());
pService->start();
BLEDevice::startAdvertising();
Serial.println(" ESP32 listo");
}
void loop() {
}
Ln 1, Col 1 ESP32 Dev Module on COM3 [not connected]

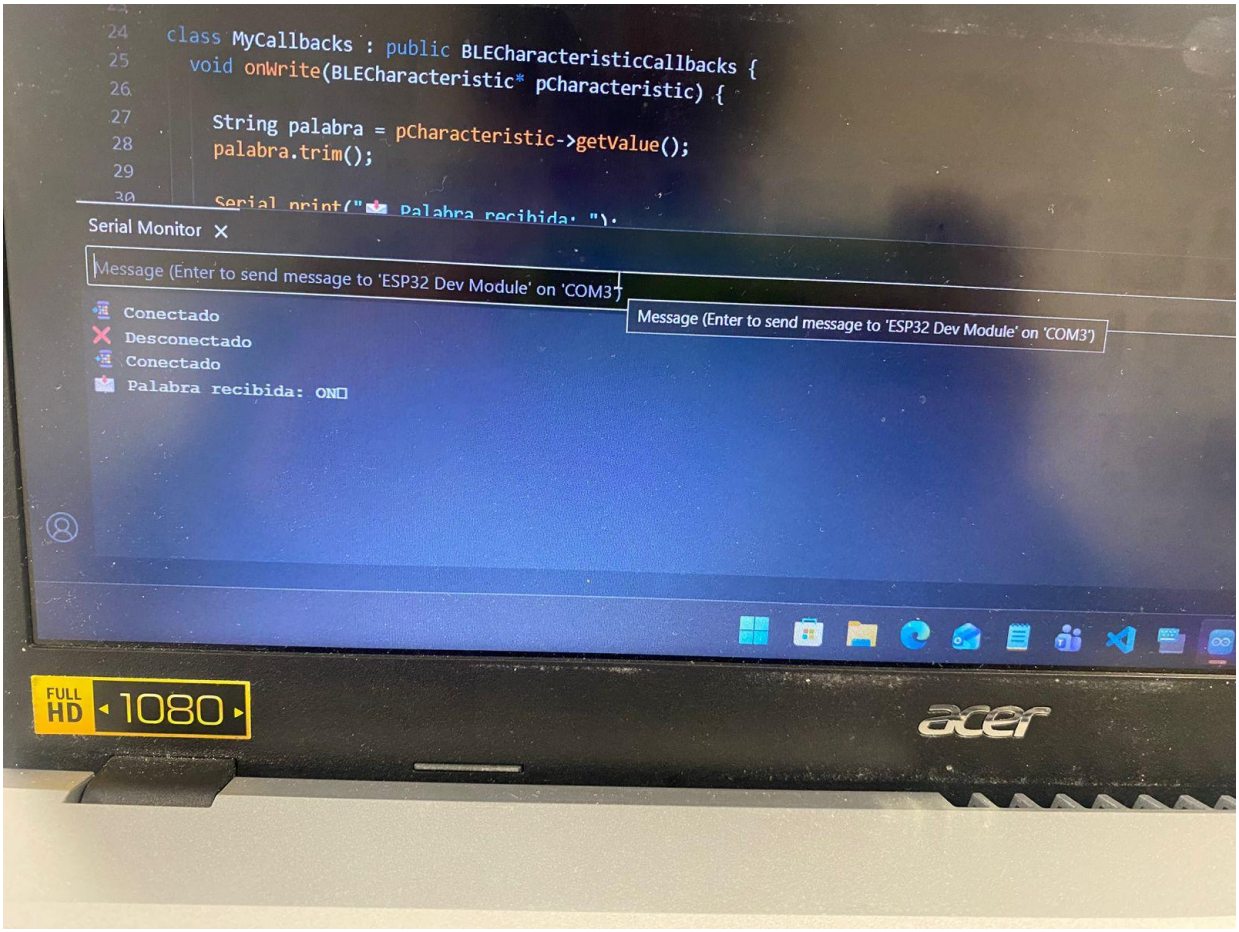
Fotos de la practica




10:56
Tec-Inal
Desconectar
Estado de Bluetooth: Conectado
PRENDE
APAGA
VERDE


10:57
Tec-Inal
Desconectar
Estado de Bluetooth: Conectado
PRENDE
APAGA
VERDE


ESP-32


class MyCallbacks : public BLECharacteristicCallbacks {
void onWrite(BLECharacteristic* pCharacteristic) {
String palabra = pCharacteristic->getValue();
palabra.trim();
Serial Monitor
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')
Conectado
Desconectado
Conectado
Palabra recibida: ON
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')
FULL HD 1080
acer

Conclusiones

Manuel Humberto Reyes Salas

En esta práctica pude comprender mejor cómo funciona la tecnología Bluetooth Low Energy aplicada a un sistema real. No solo vimos la teoría, sino que logramos llevarla a la práctica al controlar un foco mediante un relevador usando la placa ESP32 y una aplicación creada en MIT App Inventor. Esto me ayudó a entender cómo se comunican los dispositivos de forma inalámbrica y cómo un microcontrolador puede interpretar comandos enviados desde un celular.

También reforcé mis conocimientos en programación y electrónica, ya que fue necesario configurar correctamente el ESP32, establecer la comunicación BLE y diseñar una aplicación funcional. Me pareció interesante ver cómo al enviar un simple comando como “ON” u “OFF” se puede controlar una carga eléctrica de manera remota.

María José Guerrero Rios

La realización de esta práctica permitió entender y poner en práctica los fundamentos de la comunicación inalámbrica a través de Bluetooth Low Energy en un sistema de control aplicado a una situación real. Se consiguió integrar conocimientos de programación, electrónica y desarrollo de aplicaciones móviles en un único proyecto completamente funcional.

El ESP32 se destacó como una herramienta potente y flexible dentro del ámbito del Internet de las Cosas (IoT), ya que facilitó el control remoto de cargas eléctricas mediante una interfaz intuitiva diseñada en MIT App Inventor.

Este tipo de implementaciones tiene aplicación directa en áreas como la domótica, la automatización industrial, el control a distancia de maquinaria y los sistemas inteligentes, constituyendo una alternativa actual, accesible y eficiente para la gestión inalámbrica de dispositivos eléctricos.

Jesús Eli Sánchez Ruvalcaba

Con esta practica pudimos poner a prueba nuestra habilidad para utilizar componentes que necesitan de voltajes más altos y energía externa, me gusto bastante ya que fue relativamente sencilla y entretenida de hacer, los únicos inconvenientes que tuvimos fue que el componente para encender el foco no servia con el voltaje directo del esp32, por lo que tuvimos que utilizar el voltaje y la tierra de otro componente, además de que a la hora de recibir el output, el componente necesitaba tierra por lo que tuvimos que invertir la entrega de energía del esp32.

Este tipo de practicas son de mucha ayuda, ya que mas allá de solo ver la teoría, también es necesario ver la practica, no importa si nos equivocamos esto nos da muy buenas bases para proyectos futuros.