TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE AGUASCALIENTES

Tecnologías Inalámbricas

Unidad 1

Dávila Jasso Axhel Giovanni
23150315

Flores Dueñas Elvia Yuridia
23151302

Evaluativo 1 Control de potencia Bluetooth

Semestre 7

Ricardo Alejandro Rodríguez Jiménez
23 de febrero de 2026

Contenido

Tabla de contenido

Introducción

En esta práctica se llevará a cabo la integración entre una aplicación móvil desarrollada en MIT App Inventor y una placa de desarrollo ESP32-S3 utilizando comunicación Bluetooth de bajo consumo (BLE). La aplicación permitirá buscar dispositivos Bluetooth cercanos, mostrar el listado de los dispositivos encontrados y establecer la conexión con el ESP32 desde dispositivos Android e iOS.

Una vez establecida la conexión, la interfaz de la aplicación cambiará para mostrar controles que enviarán comandos al ESP32, los cuales serán interpretados para activar o desactivar un relevador conectado a la placa. De esta forma, será posible energizar o des energizar un electrodoméstico o cualquier sistema eléctrico conectado. Esta práctica permite aplicar de forma práctica los conceptos de comunicación inalámbrica e integración entre aplicaciones móviles y sistemas embebidos para el control remoto de dispositivos.

Video del Funcionamiento

https://drive.google.com/file/d/1MRtwsnhUhprv839ZaB1xGG2L4XknEoLM/view?usp=sharing

(Una disculpa por la musica de fondo profe, esque habia mucho ruido en casa y queria que se esuche el relevador en función**)**

Repositorio de GitHub

https://github.com/AXHEL1905/Evaluativo_1.git

Desarrollo

MATERIALES

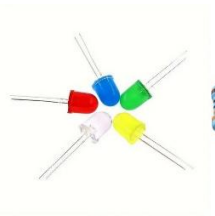
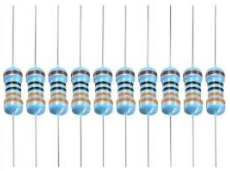
- 1 Protoboard
- 3 Led's (verde, rojo y azul)
- 8 jumpers
- 1 socket
- 20 cm de cable
- 1 clavija
- 1 relevador
- 3 resistencias de 330 oms
- 1 foco
- 1 Esp32 (Con cable de entrada micro USB)
- 1 dispositivo Android

HARDWARE

- App desarrollada en My App Inventor

Imágenes del material

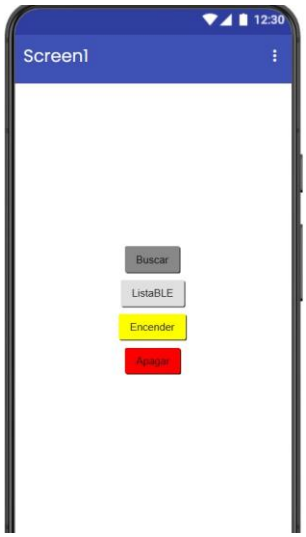






App de My App Inventor

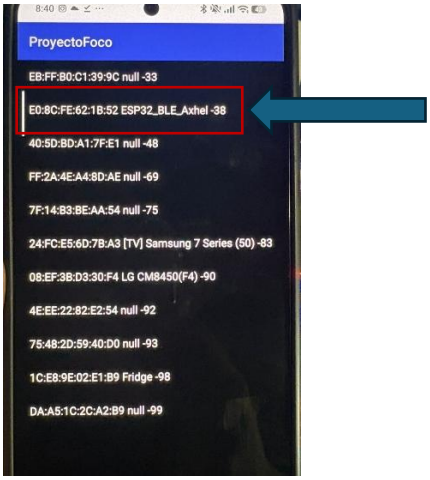


Haciendo uso de la placa de desarrollo ESP32 S3 o cualquiera disponible, crear una integración que comprenda lo siguiente:

App Móvil que pueda cumplir con las siguientes necesidades

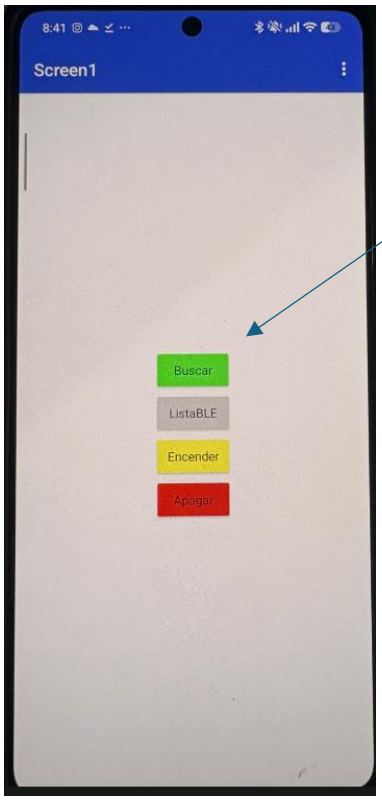
-Buscar los dispositivos Bluetooth cercanos

-Mostrar el listado de los dispositivos encontrados



-Realizar la conexión entre la app en IOS y Android con el dispositivo BLE Esp32

Al establecer correctamente la conexión, la interfaz de la App móvil deberá cambiar para mostrar una serie de botones que permitan el cambio de estado en una dase de potencia, con el fin de poder energizar o des energizar electrodomésticos o cualquier tipo de implemento, maquinaria o sistema que se energice con alta potencia.



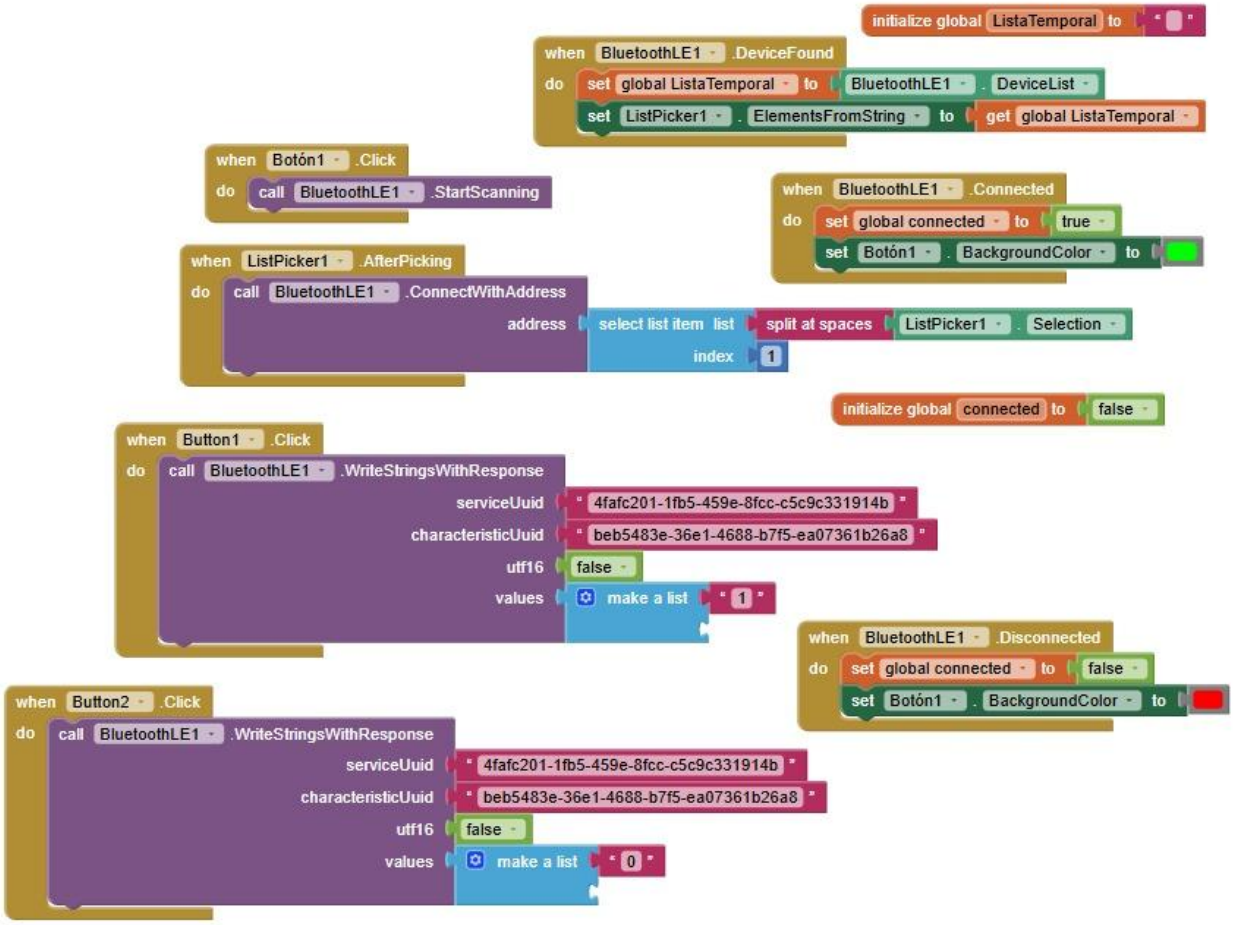
El color verde en el boton indica que la conexión se completo

Se requiere uso de la web MIT App Inventor y el uso de la extensión Bluetooth BLE, además de toda la integración para el uso de BLE

-Extension Bluetooth BLE

-Uso e integración de My App Inventor (MIT)

Diagrama de Bloque



Por la parte de la ESP deberá integrarse el código que permita funcionar como servidor BLE y cumpla con el siguiente flujo de trabajo:

Código de Arduino ID

```ino
#include <BLEDevice.h>
#include <BLEServer.h>
#include <BLEUtils.h>
#include <BLE2902.h>

// Pines LEDs
#define LED_ROJO 25
#define LED_AZUL 26
#define LED_VERDE 27

// Relevador
const int relayPin = 2;

// UUIDs
#define SERVICE_UUID        "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
#define CHARACTERISTIC_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"

// Callbacks conexión BLE
class MyServerCallbacks: public BLEServerCallbacks {
  void onConnect(BLEServer* pServer) {
    // Solo verde encendido
    digitalWrite(LED_ROJO, LOW);
    digitalWrite(LED_AZUL, LOW);
    digitalWrite(LED_VERDE, HIGH);
    Serial.println("Conectado");
  }

  void onDisconnect(BLEServer* pServer) {
    // Solo rojo encendido
    digitalWrite(LED_ROJO, HIGH);
    digitalWrite(LED_AZUL, LOW);
    digitalWrite(LED_VERDE, LOW);

    BLEDevice::startAdvertising();
    Serial.println("Desconectado");
  }
};

// Recibir datos
class MyCallbacks: public BLECharacteristicCallbacks {
  void onWrite(BLECharacteristic *pCharacteristic) {
    String value = pCharacteristic->getValue();
    if (value.length() > 0) {
      if (value[0] == '1') digitalWrite(relayPin, LOW);
      if (value[0] == '0') digitalWrite(relayPin, HIGH);
    }
  }
};

void setup() {
  Serial.begin(115200);

  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, HIGH);

  pinMode(LED_ROJO, OUTPUT);
  pinMode(LED_AZUL, OUTPUT);
  pinMode(LED_VERDE, OUTPUT);

  // Al encender: buscando / listo (solo azul)
  digitalWrite(LED_ROJO, LOW);
  digitalWrite(LED_AZUL, HIGH);
  digitalWrite(LED_VERDE, LOW);

  BLEDevice::init("ESP32_BLE_Axhel");
  BLEServer *pServer = BLEDevice::createServer();
  pServer->setCallbacks(new MyServerCallbacks());

  BLEService *pService = pServer->createService(SERVICE_UUID);

  BLECharacteristic *pCharacteristic = pService->createCharacteristic(
    CHARACTERISTIC_UUID,
    BLECharacteristic::PROPERTY_READ |
    BLECharacteristic::PROPERTY_WRITE
  );

  pCharacteristic->setCallbacks(new MyCallbacks());
  pService->start();

  BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
  pAdvertising->addServiceUUID(SERVICE_UUID);
  pAdvertising->setScanResponse(true);
  BLEDevice::startAdvertising();

  Serial.println("ESP32 buscando conexion BLE...");
}

void loop() {
  delay(10);
}
```

Al energizar el dispositivo la placa deberá mostrar en el monitor serial la leyenda que indique el estatus del dispositivo

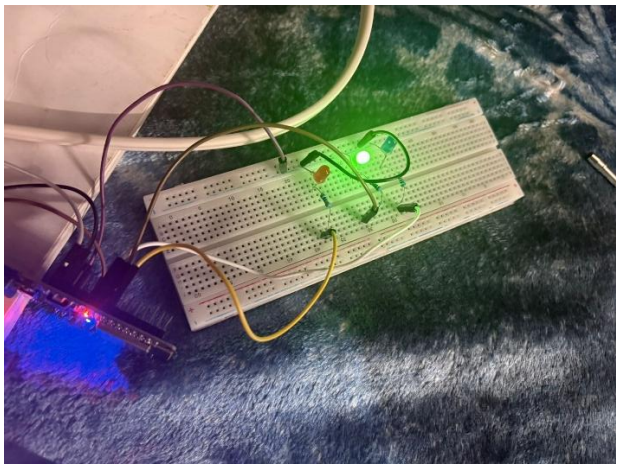
- Código cargado a la placa Esp32

```
Salida
Writing at 0x0010e2d5 [==========================>   ]  95.7% 655360/684852 bytes...

Writing at 0x00113d31 [==========================> ]  98.1% 671744/684852 bytes...

Writing at 0x00118930 [==============================] 100.0% 684852/684852 bytes...
Wrote 1083696 bytes (684852 compressed) at 0x00010000 in 11.2 seconds (776.7 kbit/s).
Hash of data verified.

Hard resetting via RTS pin...
```

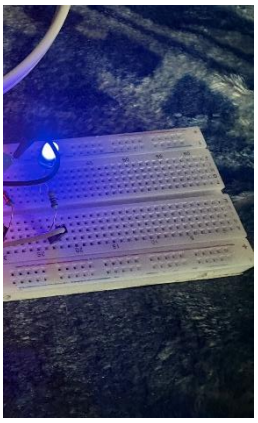
- Leyenda del Monitor Serial

```
Salida  Monitor Serie ×
Mensaje (Intro para mandar el mensaje de 'ESP32 Dev Module' a 'COM6')
mode:DIO, clock div:1
load:0x3fff0030,len:4876
ho 0 tail 12 room 4
load:0x40078000,len:16560
load:0x40080400,len:3500
entry 0x400805b4
ESP32 buscando conexion BLE...
Conectado
```

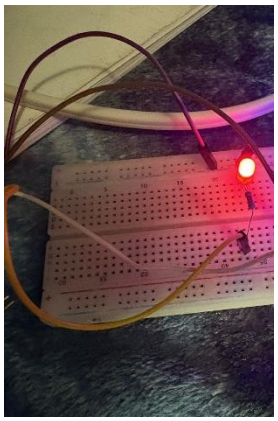
-Listo para Conectar (LED Verde)



-Bluetooth Conectado (LED Azul)

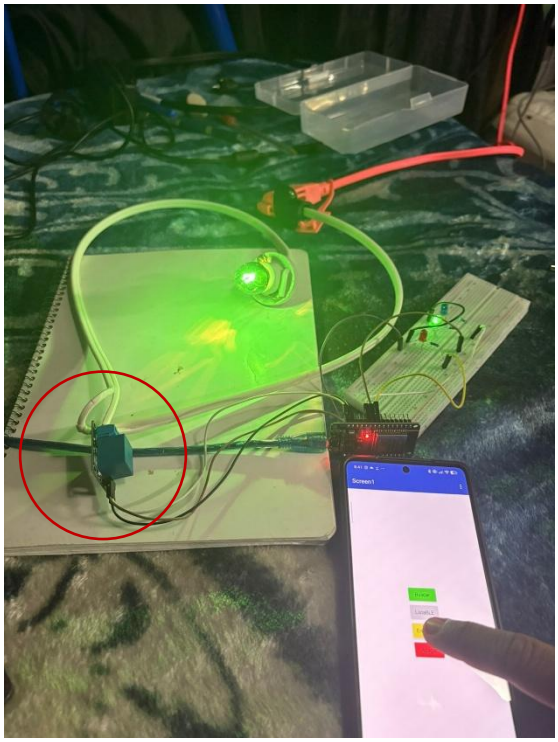


-Bluetooth desconectado (LED Rojo

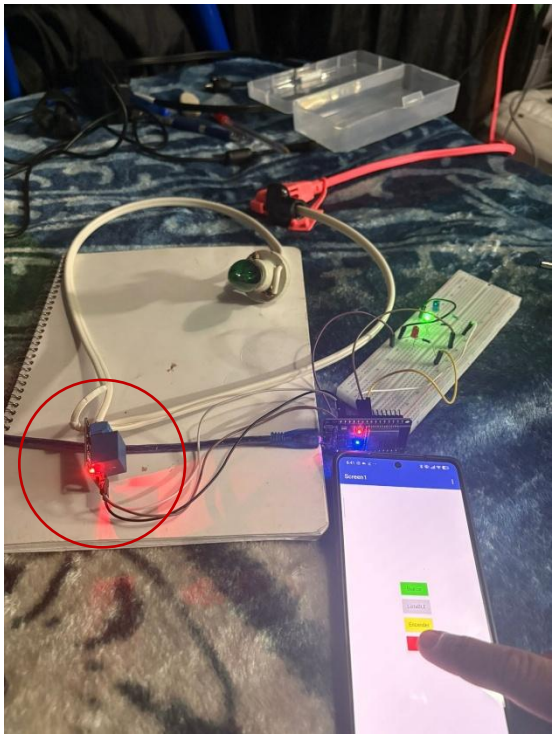


Con la cadena de datos enviada por el dispositivo móvil y registrara por el ESP32, deberá interpretar el significado y activar o desactivar un relevador por medio de alguno de los pines disponibles en la placa. así mismo, cuando la señal enviada sea la correcta deberá desactivar el relevador, con lo cual se des energizará la línea eléctrica apagando el electrodoméstico.

ENCENDIDO



APAGADO



En estas fotos el led lo teníamos implementado que el verde fuera que logro la conexión pero se termino cambiando

Conclusión

Dávila Jasso Axhel Giovanni

Con esta práctica se logró integrar una aplicación móvil con la placa ESP32-S3 mediante Bluetooth BLE, permitiendo el envío de comandos para controlar un relevador y, con ello, encender o apagar un dispositivo eléctrico. Se comprendió el funcionamiento del ESP32 como servidor BLE y la comunicación con una app desarrollada en MIT App Inventor.

Además, se reforzó la relación entre software y hardware, identificando la importancia de una correcta configuración y pruebas para que la comunicación funcione de manera estable. Esta práctica permitió aplicar los conceptos vistos en clase en un escenario real de control inalámbrico de dispositivos.

Flores Dueñas Elvia Yuridia

En este proyecto se desarrolló una aplicación móvil que se comunica con la placa ESP32 mediante Bluetooth Low Energy (BLE) para controlar un relevador. Para mí, este proyecto fue muy útil porque me permitió aprender cómo funciona la comunicación BLE entre una app móvil y un microcontrolador, además de comprender cómo integrar hardware y software en un mismo sistema. También aprendí a interpretar los datos enviados desde la aplicación y a controlar dispositivos de alta potencia mediante un relevador. En general, este trabajo ayudó a reforzar los conocimientos sobre programación, electrónica y automatización

Referencias Bibliográficas

Referencia sobre Bluetooth Low Energy (BLE)

Akhayad, Y. (2016). *BLE: Análisis de las prestaciones y aplicaciones del protocolo Bluetooth Low Energy (BLE)* (Trabajo de Fin de Grado). Universidad Politécnica de Cataluña. https://upcommons.upc.edu/bitstream/handle/2099.1/19610/TFG_YAAkhayad.pdf

---

Referencia sobre ESP32 y BLE

[Castañeda, J.] (2021). *Control de dispositivos mediante Bluetooth Low Energy con ESP32* (Tesis de Licenciatura). Repositorio Institucional, Universidad Tecnológica de Pereira. https://repositorio.utp.edu.co/handle/11059/12345

Referencia sobre MIT App Inventor y BLE

Universidad de Málaga. (2023). *Desarrollo de aplicaciones móviles con MIT App Inventor para comunicación Bluetooth*. Repositorio Académico Universidad de Málaga. https://hdl.handle.net/10630/12345