



**TECNOLÓGICO
NACIONAL DE MÉXICO**
INSTITUTO TECNOLÓGICO DE TIJUANA



INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE ENERO JUNIO - 2024

INGENIERÍA EN SISTEMAS COMPUTACIONALES

LENGUAJES DE INTERFAZ

PRÁCTICA 4 - DOCUMENTACIÓN

JULIO ALEJANDRO HERNÁNDEZ LEÓN - 21211963
SANTY FRANCISCO MARTINEZ CASTELLANOS - 21211989
LUIS ROBERTO LEAL LUA - 21211970

RENÉ SOLIS REYES

Índice

Código Utilizado y Descripción.....	3
Fotografías.....	7

Código Utilizado y Descripción

//Práctica 4 - Control Bluetooth con Pico W

//Team Debian

//Integrantes:

//Santy Francisco Martinez Castellanos - 21211989

//Julio Alejandro Hernández León -21211963

//Luis Roberto Leal Lua - 21211970

#include <Arduino.h>

#include <WiFi.h>

#include <WiFiClientSecure.h>

// Nombre y clave de red

const char* ssid = "INFINITUM5A66_2.4";

const char* password = "duke1975";

// APY y Modelo

const char* api_key =

"sk-proj-SCrtDnfTA7psqYCbJUlnT3BlbkFJbiGIBcW6p4Av82yVF5KA";

const char* model_id = "gpt-3.5-turbo-0125";

// Configuración del host y URL para la API

const char* host = "api.openai.com";

const char* url = "/v1/chat/completions";

void setup() {

Serial.begin(115200);

delay(10);

// Conexión a Wi-Fi

Serial.println();

```

Serial.print("Conectando a ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi conectado");

// Configuración del cliente seguro
WiFiClientSecure client;
client.setInsecure();
client.setTimeout(10000);

// Conexión a la API de ChatGPT
Serial.print("Conectando a la API...");
if (!client.connect(host, 443)) {
    Serial.println("¡Error al conectar!");
    return;
}
Serial.println("¡Conexión exitosa!");

// Construcción de la carga útil de solicitud
String estructura = "{";
estructura += "\"model\": \"gpt-3.5-turbo-0125\",";
estructura += "\"messages\": [";
estructura += "{\"role\": \"user\",";
estructura += "\"content\": \"Dime un chiste del team windows\"}";
estructura += "],";
estructura += "\"temperature\": 0.7,\"";

```

```

estructura += "\"max_tokens\": 50,\";
estructura += "\"n\": 1,\";
estructura += "\"stop\": \"\\n\"}";

// Construcción de la solicitud HTTP
String resultado = "POST ";
resultado += url;
resultado += " HTTP/1.1\r\n";
resultado += "Host: ";
resultado += host;
resultado += "\r\n";
resultado += "Authorization: Bearer ";
resultado += api_key;
resultado += "\r\n";
resultado += "Content-Type: application/json\r\n";
resultado += "Content-Length: ";
resultado += estructura.length();
resultado += "\r\n\r\n";
resultado += estructura;

// Envío de la solicitud HTTP
Serial.println("Enviando solicitud...");
client.print(resultado);

// Espera la respuesta de la API de OpenAI
Serial.println("Esperando respuesta...");
while (client.connected()) {
    if (client.available()) {
        String response = client.readString();
        Serial.print(response);
    }
}

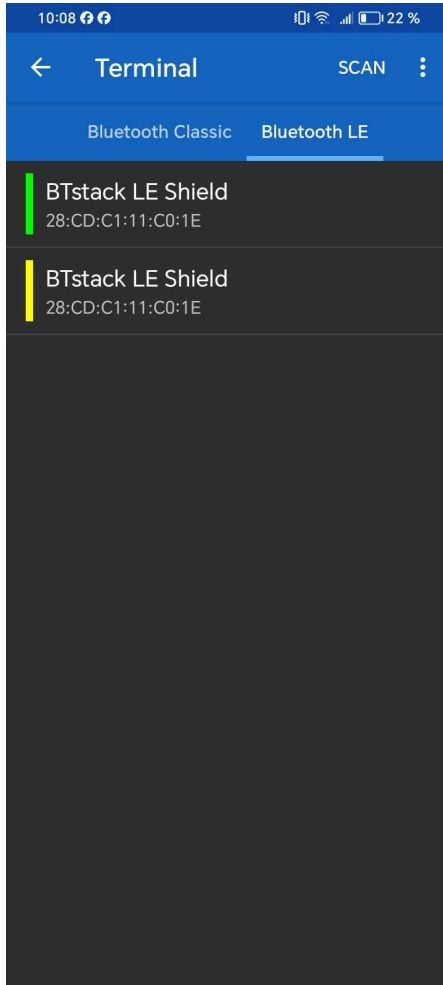
```

```
}  
}
```

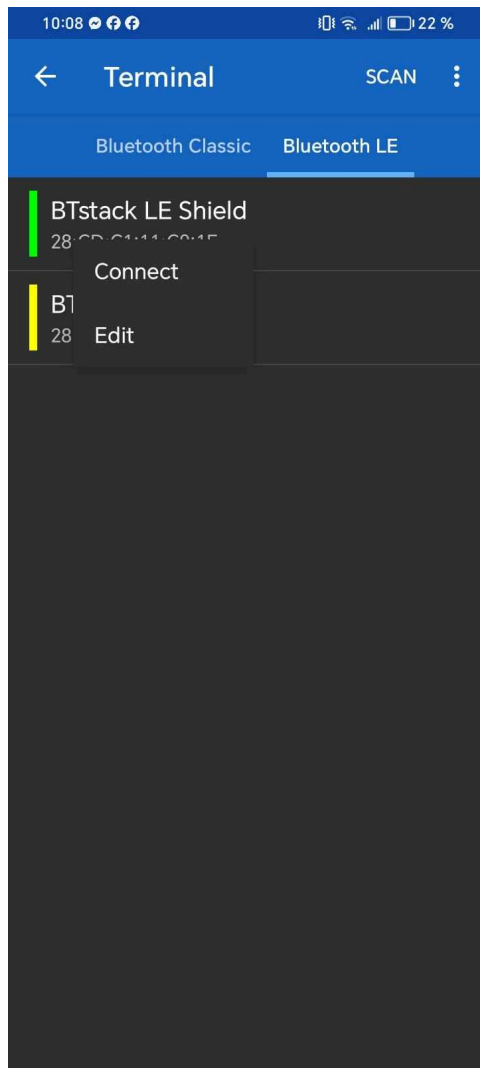
```
void loop() {  
  // No hay nada que hacer aquí, ya que todo el trabajo se realiza en setup()  
}
```

Fotografías

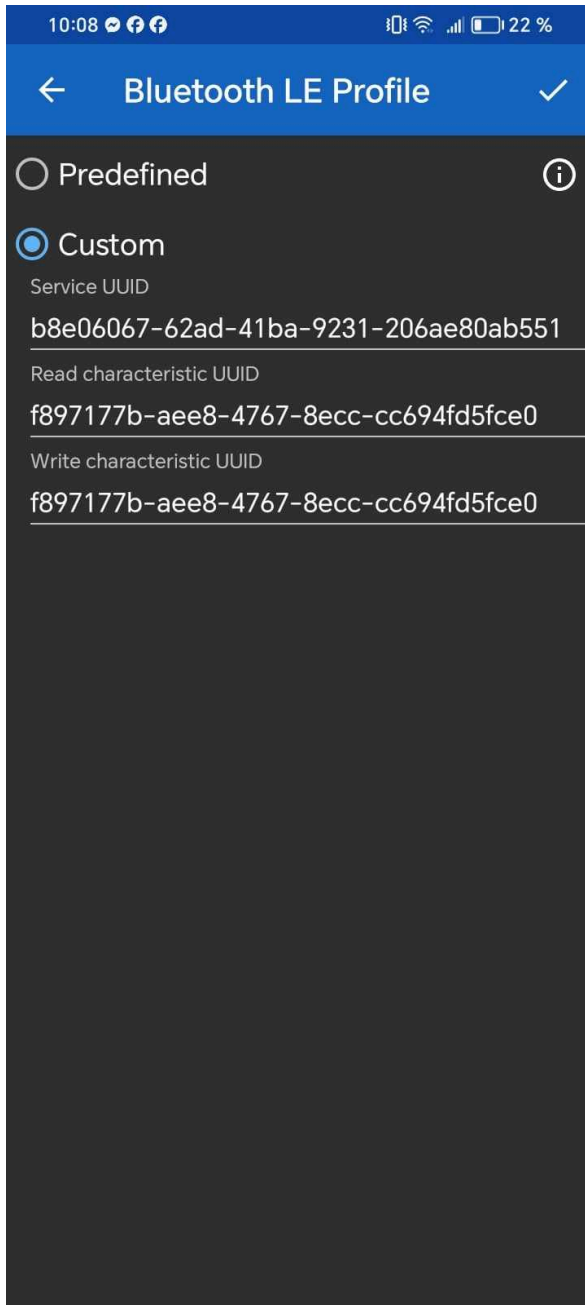
Se muestra la configuración para conectar dentro de la aplicación “Serial Bluetooth Terminal”.



Conexión.



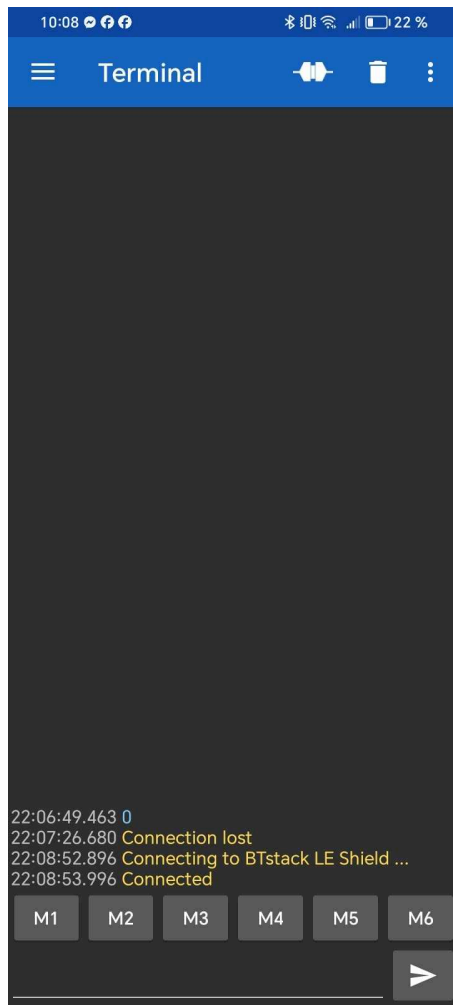
El perfil de bluetooth que se utiliza es customizado y se insertaron los siguientes UUID.

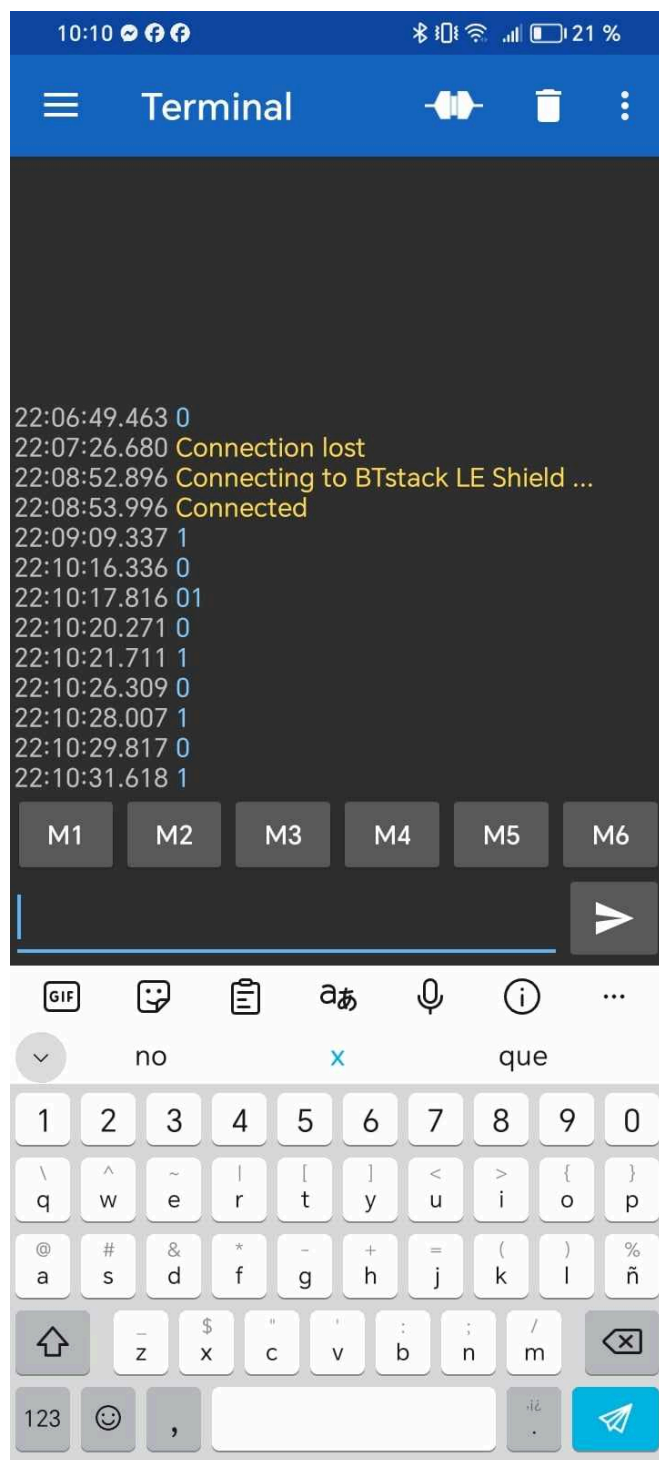


Cuando el dispositivo es conectado:

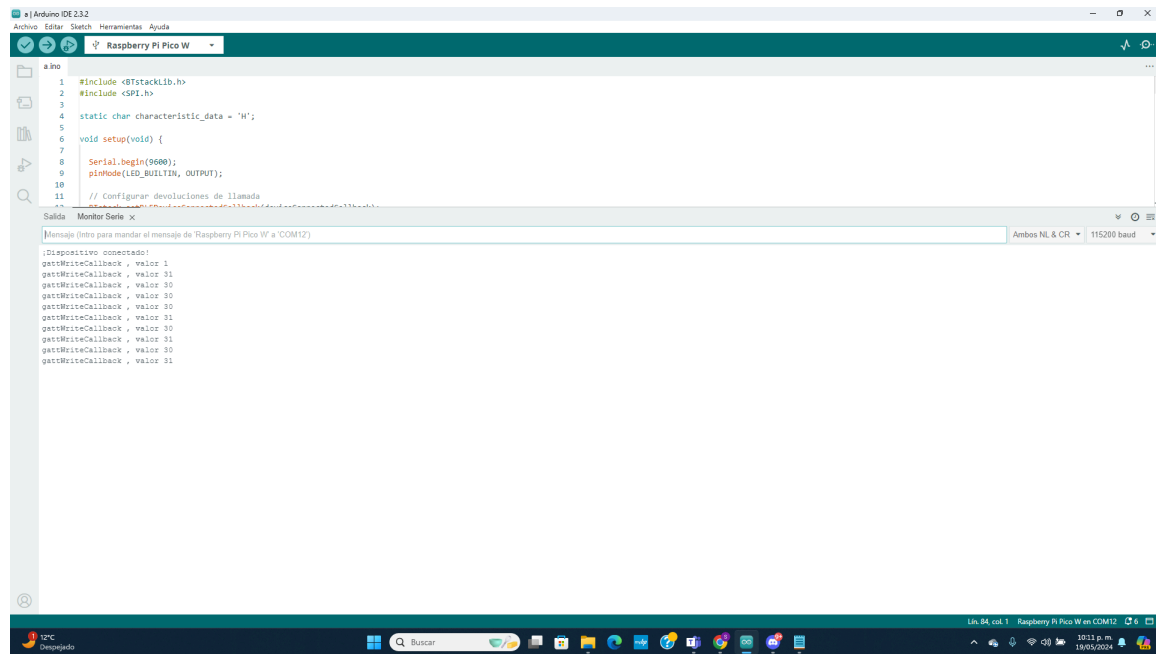


En la aplicación se confirma el resultado.





Se muestran los resultados en la aplicación de Arduino.



```
1 #include <BTstackLib.h>
2 #include <SPI.h>
3
4 static char characteristic_data = 'H';
5
6 void setup(void) {
7   Serial.begin(9600);
8   pinMode(LED_BUILTIN, OUTPUT);
9
10  // Configurar devoluciones de llamada
11
```

Salida Monitor Serie x

Dispositivo conectado:

```
getWriteCallback, valor 1
getWriteCallback, valor 01
getWriteCallback, valor 00
getWriteCallback, valor 00
getWriteCallback, valor 00
getWriteCallback, valor 01
getWriteCallback, valor 00
getWriteCallback, valor 00
getWriteCallback, valor 01
getWriteCallback, valor 00
getWriteCallback, valor 00
getWriteCallback, valor 01
```

Linux 64 col. 1 Raspberry Pi Pico W en COM12 10:18 19/05/2024