



**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

```
//Practica 4 Bluetooth
//Team Debian
// Martinez Castellanos Santy Francisco
// Leal Lua Luis Roberto
// Hernandez Leon Julio Alejandro

#include <BTstackLib.h> // Biblioteca Bluetooth
#include <SPI.h> // Biblioteca SPI

static char characteristic_data = 'H';

void setup(void) {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT); // LED DE SALIDA

  // Funciones de callback
  BTstack.setBLEDeviceConnectedCallback(deviceConnectedCallback);
  BTstack.setBLEDeviceDisconnectedCallback(deviceDisconnectedCallback);
  BTstack.setGATTCharacteristicRead(gattReadCallback);
  BTstack.setGATTCharacteristicWrite(gattWriteCallback);

  // Base de datos GATT

  BTstack.addGATTService(new
  UUID("B8E06067-62AD-41BA-9231-206AE80AB551")); // Se añade un servicio
  GATT

  BTstack.addGATTCharacteristic(new
  UUID("f897177b-ae8-4767-8ecc-cc694fd5fcef"), ATT_PROPERTY_READ,
  "This is a String!"); // Se añade una característica GATT
```

```
BTstack.addGATTCharacteristicDynamic(new  
UUID("f897177b-ae8-4767-8ecc-cc694fd5fce0"), ATT_PROPERTY_READ |  
ATT_PROPERTY_WRITE | ATT_PROPERTY_NOTIFY, 0); // Se añade una  
característica GATT dinámica
```

```
// Inicia Bluetooth  
BTstack.setup();  
BTstack.startAdvertising();  
}
```

```
void loop(void) {  
    BTstack.loop();  
}
```

```
void deviceConnectedCallback(BLEStatus status, BLEDevice *device) {  
    (void) device;  
    switch (status) {  
        case BLE_STATUS_OK:  
            Serial.println("Device connected!"); // Indica si el dispositivo esta conectado  
            break;  
        default:  
            break;  
    }  
}
```

```
void deviceDisconnectedCallback(BLEDevice * device) {  
    (void) device;  
    Serial.println("Disconnected."); // Indica si el dispositivo se ha desconectado  
}
```

```

uint16_t gattReadCallback(uint16_t value_handle, uint8_t * buffer, uint16_t
buffer_size) {
    (void) value_handle;
    (void) buffer_size;
    if (buffer) {
        Serial.print("gattReadCallback, value: ");
        Serial.println(characteristic_data, HEX);    // Valor de la caractística
Hexadecimal
        buffer[0] = characteristic_data;

        // Control del LED
        if (characteristic_data == '1')
            digitalWrite(LED_BUILTIN, HIGH); // LED ENCENDIDO
        else if (characteristic_data == '0')
            digitalWrite(LED_BUILTIN, LOW); // LED APAGADO
    }
    return 1;
}

```

```

int gattWriteCallback(uint16_t value_handle, uint8_t *buffer, uint16_t size) {
    (void) value_handle;
    (void) size;
    characteristic_data = buffer[0];

    // Controla del LED
    if (characteristic_data == '1')
        digitalWrite(LED_BUILTIN, HIGH); // LED ENCENDIDO
    else if (characteristic_data == '0')
        digitalWrite(LED_BUILTIN, LOW); // LED APAGADO

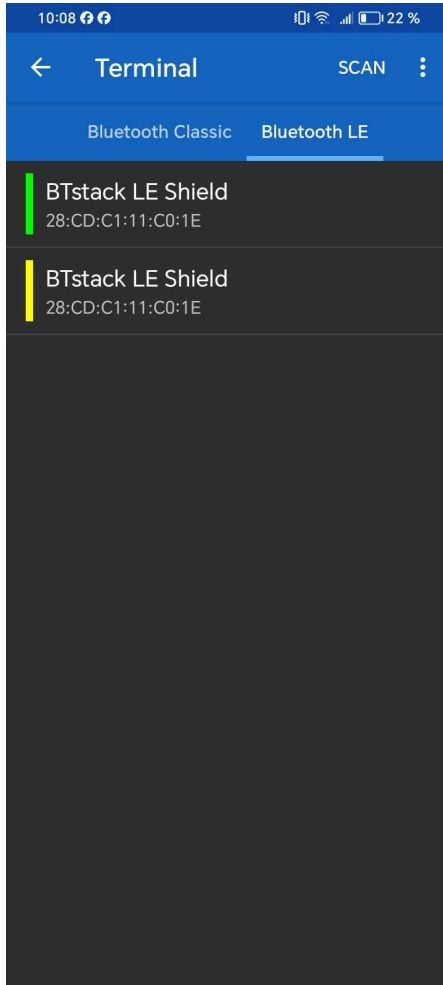
    Serial.print("gattWriteCallback , value ");
}

```

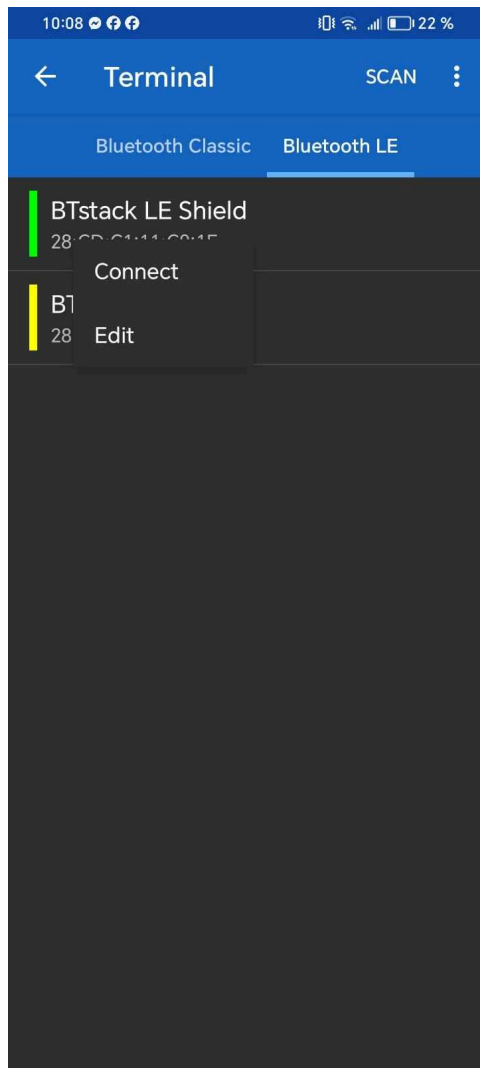
```
Serial.println(characteristic_data, HEX); // Valor de la característica Hexadecimal  
return 0;  
}
```

## Fotografías

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

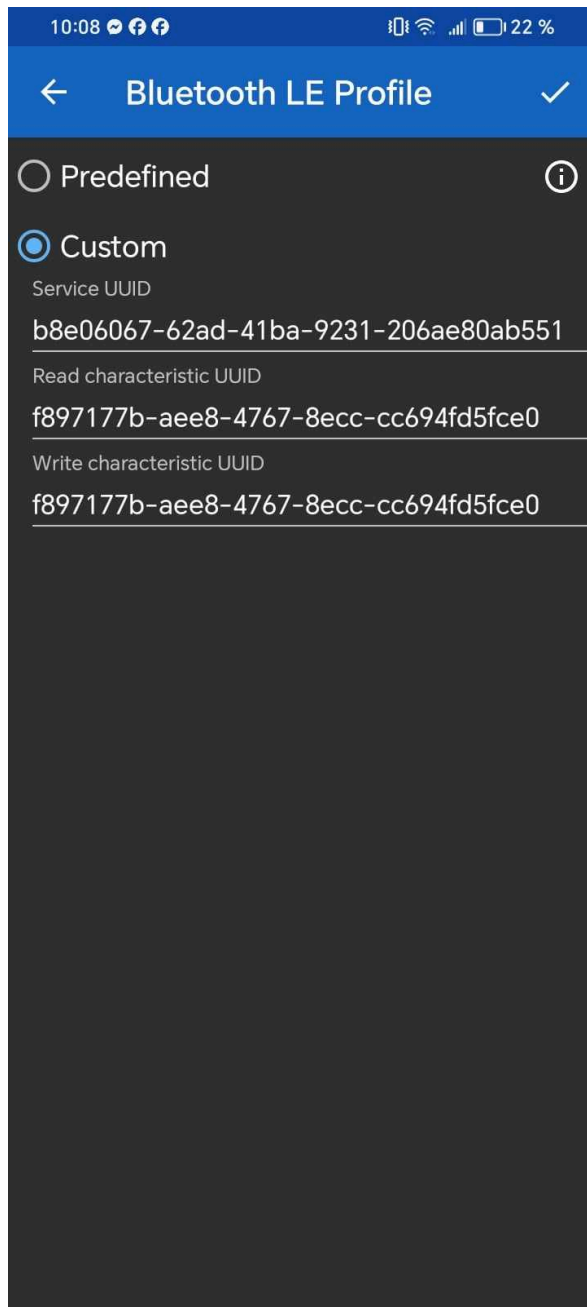


Conexión.





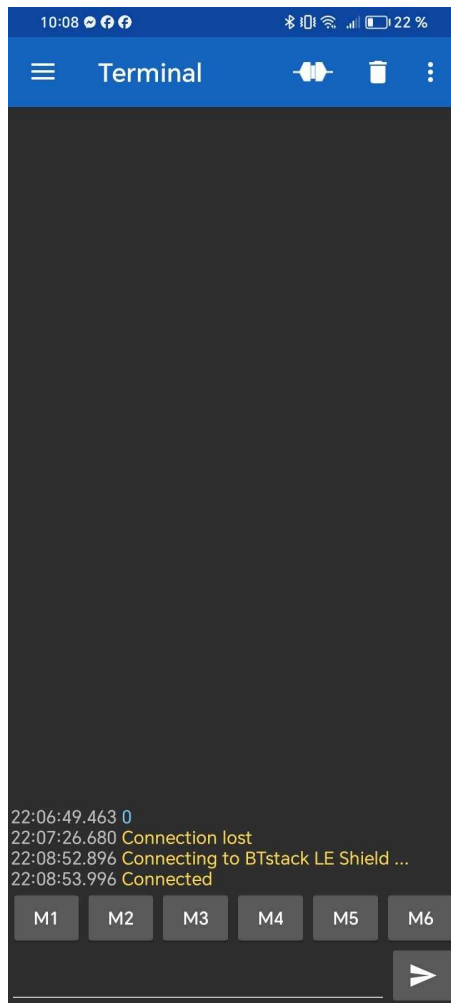
El perfil de bluetooth que se utiliza es customizado y se insertaron los siguientes UUID.

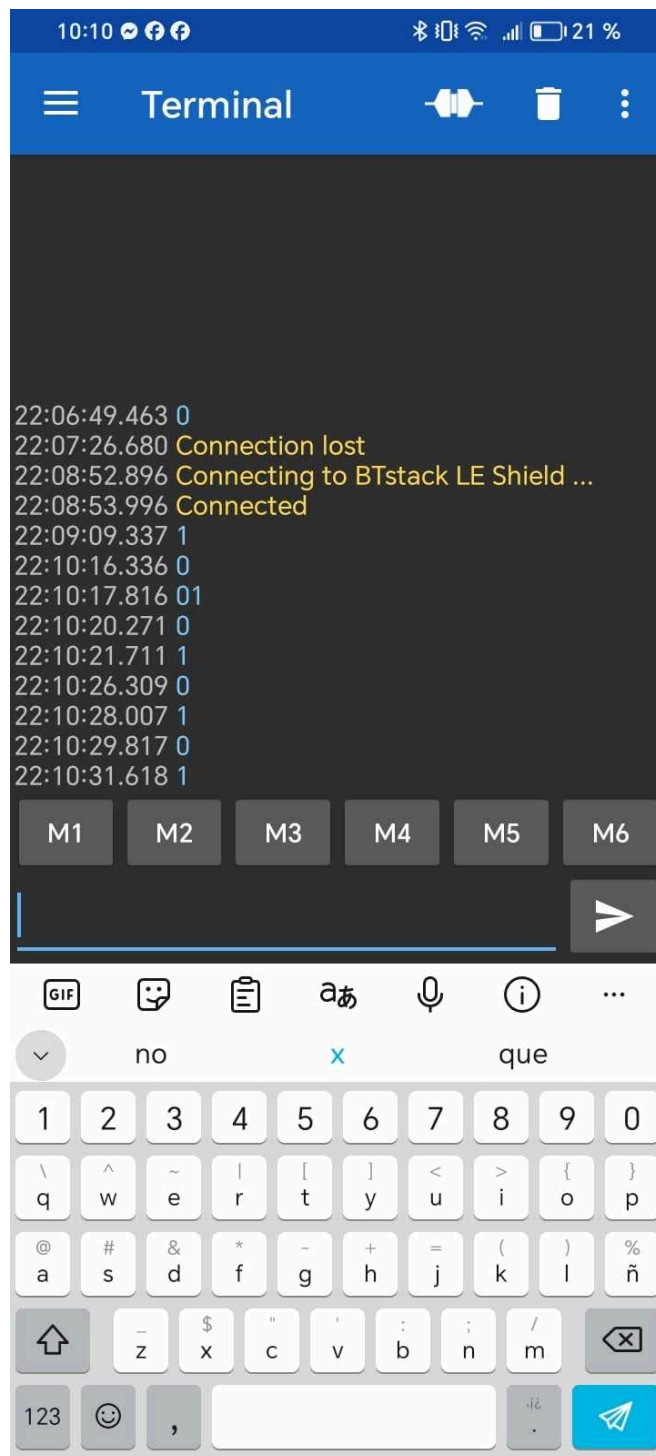


Cuando el dispositivo es conectado:

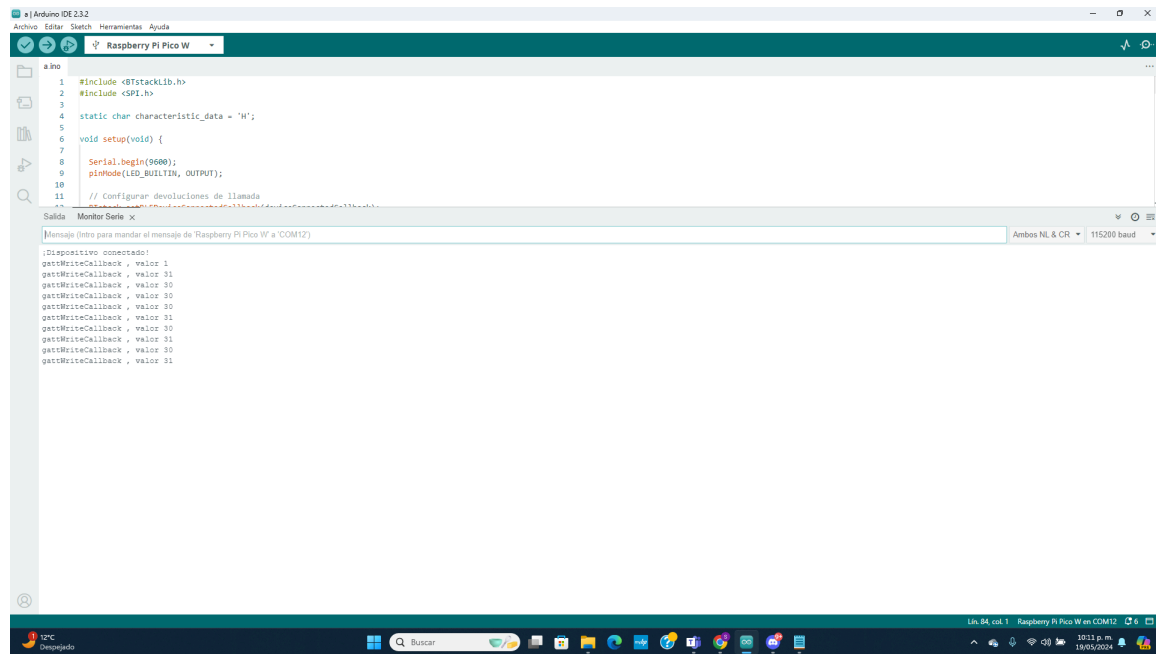


En la aplicación se confirma el resultado.





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
getWriteCallback, valor 00
getWriteCallback, valor 01

```

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