Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Práctica 3: Comunicación Bluetooth

Seguridad de las comunicaciones Inalambricas

Autor: Cheuk Kelly Ng Pante (alu0101364544@ull.edu.es)

Fecha: 8 de diciembre de 2025

Índice general

1. Comprobar si hay un adaptador de Bluetooth activo

Para comprobar si hay un adaptador de Bluetooth activo en el sistema, se puede utilizar el comando `hciconfig` en la terminal de Linux. Este comando muestra información sobre los dispositivos Bluetooth disponibles en el sistema. En la Figura 1.1 se muestra un ejemplo de la salida del comando `hciconfig`, donde se puede observar que hay un adaptador de Bluetooth activo (hci0) con su dirección MAC y estado.

```
❯ hciconfig
hci0:   Type: Primary  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 679:8  SCO MTU: 48:16
        DOWN
        RX bytes:690 acl:0 sco:0 events:49 errors:0
        TX bytes:3161 acl:0 sco:0 commands:49 errors:0
```

Figura 1.1: Salida del comando `hciconfig`

En este caso aparece el adaptador como `DOWN`, lo que indica que el adaptador está desactivado. Para activarlo, se puede utilizar el comando `sudo hciconfig hci0 up`, donde `hci0` es el nombre del adaptador de Bluetooth.

```
❯ sudo hciconfig hci0 up
❯ hciconfig
hci0:   Type: Primary  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 679:8  SCO MTU: 48:16
        UP RUNNING
        RX bytes:1282 acl:0 sco:0 events:83 errors:0
        TX bytes:3848 acl:0 sco:0 commands:83 errors:0
```

Figura 1.2: Activación del adaptador de Bluetooth

2. Herramientas de Bluez: bluetoothctl

La herramienta `bluetoothctl` es una utilidad de línea de comandos que forma parte del paquete BlueZ, el cual es la pila oficial de protocolos Bluetooth para Linux. Esta herramienta permite gestionar dispositivos Bluetooth, incluyendo la búsqueda, emparejamiento y conexión a dispositivos.

Para iniciar `bluetoothctl`, simplemente se debe abrir una terminal y escribir el comando `bluetoothctl`. Una vez dentro de la herramienta, se pueden utilizar varios comandos para interactuar con los dispositivos Bluetooth. Algunos de los comandos más comunes son:

- `power on/off`: Activa o desactiva el adaptador de Bluetooth.
- `scan on/off`: Inicia o detiene la búsqueda de dispositivos Bluetooth cercanos.
- `devices`: Muestra una lista de dispositivos Bluetooth conocidos.
- `connect <MAC>`: Conecta a un dispositivo Bluetooth emparejado.
- `disconnect <MAC>`: Desconecta de un dispositivo Bluetooth.

```
❯ bluetoothctl
Agent registered
[CHG] Controller 00:1A:7D:DA:71:13 Pairable: yes
[bluetooth]# list
Controller 00:1A:7D:DA:71:13 feichay-MS-7A38 [default]
[bluetooth]# 
```

Figura 2.1: Uso de la herramienta `bluetoothctl`

En la Figura 2.2 se muestra un ejemplo de cómo utilizar el comando `scan on` para buscar dispositivos Bluetooth cercanos. La salida muestra varios dispositivos encontrados, junto con sus direcciones MAC y nombres (si están disponibles).

```
[bluetooth]# scan on
Discovery started
[CHG] Controller 00:1A:7D:DA:71:13 Discovering: yes
[NEW] Device 79:29:01:B1:33:F4 JBL
[NEW] Device C0:28:8D:9A:1B:AA UE BOOM 2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Key: 0x0003
[CHG] Device C0:28:8D:9A:1B:AA ManufacturerData Value:
  00 a5 e3 00 20 05 00 00 00 00 00 00 c0 28 8d 9a  .... ........(..
  1b aa 32                                         ..2
```

Figura 2.2: Escaneo de dispositivos Bluetooth con `bluetoothctl`

Una vez escaneado se para el escaneo con el comando `scan off`. Luego, se lista los dispositivos encontrados con el comando `devices`, como se muestra en la Figura 2.3.

```
[bluetooth]# scan off
Discovery stopped
[CHG] Device C0:28:8D:9A:1B:AA TxPower is nil
[CHG] Device C0:28:8D:9A:1B:AA RSSI is nil
[CHG] Device 79:29:01:B1:33:F4 TxPower is nil
[CHG] Device 79:29:01:B1:33:F4 RSSI is nil
[CHG] Controller 00:1A:7D:DA:71:13 Discovering: no
[bluetooth]# devices
Device 5C:AD:BA:CD:4C:94 iPhone cknp
Device 79:29:01:B1:33:F4 JBL
Device C0:28:8D:9A:1B:AA UE BOOM 2
```

Figura 2.3: Listado de dispositivos Bluetooth con `bluetoothctl`

Una vez que se tiene la dirección MAC del dispositivo al que se desea conectar, se puede utilizar el comando `connect <MAC>` para establecer la conexión. En la Figura 2.4 se muestra un ejemplo de cómo conectar a un dispositivo Bluetooth utilizando su dirección MAC.

```
[bluetooth]# devices
Device 5C:AD:BA:CD:4C:94 iPhone cknp
Device C0:28:8D:9A:1B:AA UE BOOM 2
[bluetooth]# connect C0:28:8D:9A:1B:AA
Attempting to connect to C0:28:8D:9A:1B:AA
[CHG] Device C0:28:8D:9A:1B:AA Connected: yes
[CHG] Device C0:28:8D:9A:1B:AA Modalias: bluetooth:v000ApFFFFdFFFF
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 00001101-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 00001108-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 0000110b-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 0000111e-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device C0:28:8D:9A:1B:AA ServicesResolved: yes
[CHG] Device C0:28:8D:9A:1B:AA Paired: yes
Connection successful
[UE BOOM 2]#
```

Figura 2.4: Conexión a un dispositivo Bluetooth con `bluetoothctl`

Ya conectado, se puede entrar al menú de servicios del dispositivo con el comando `menu gatt`, como se muestra en la Figura 2.5.

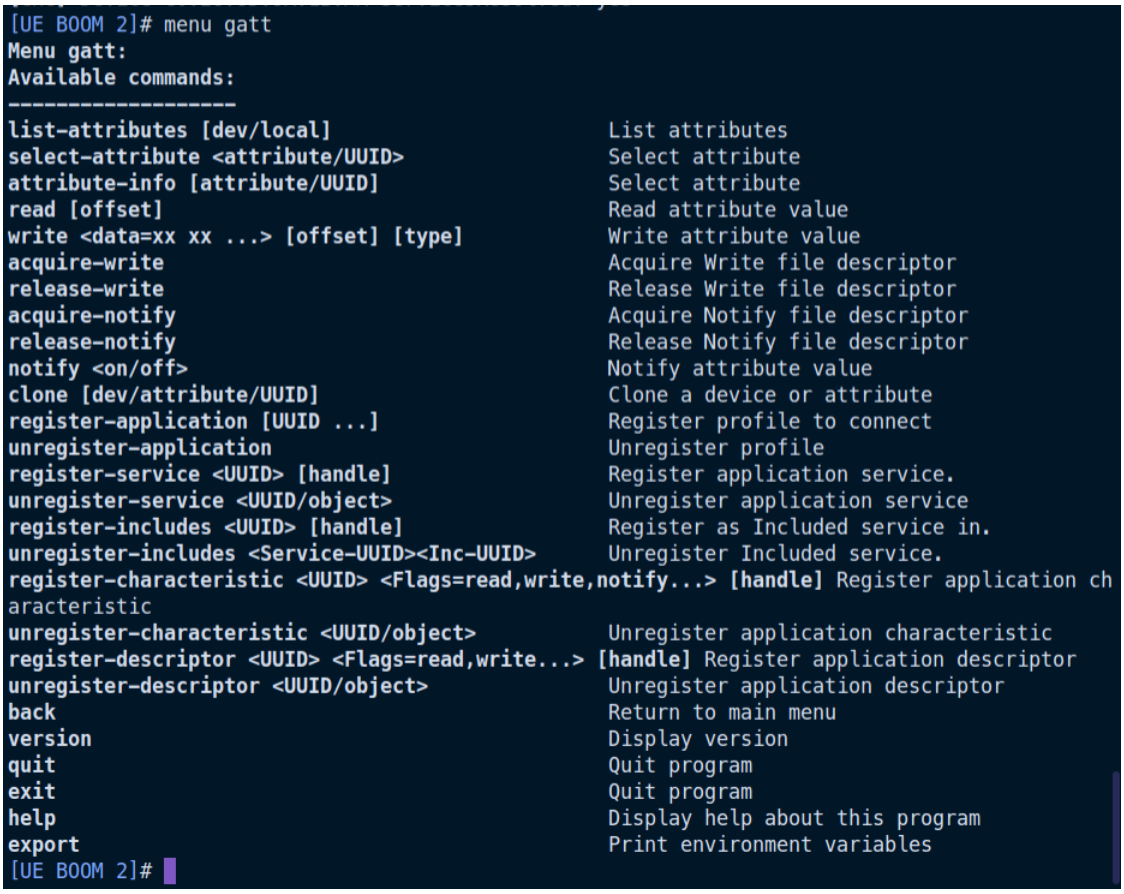

Figura 2.5: Menú GATT en `bluetoothctl`

Y se puede listar los servicios disponibles con el comando `list-attributes`, como se muestra en la Figura 2.6 aunque en este caso no hay servicios disponibles.

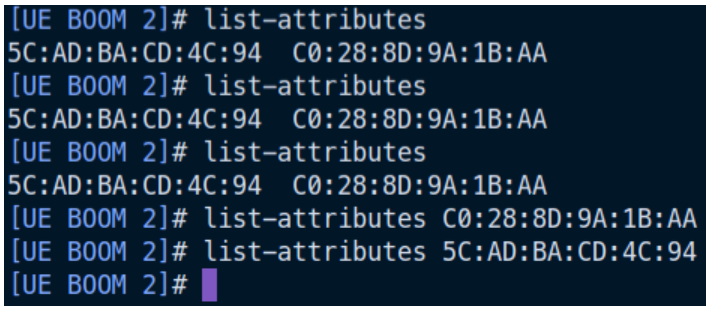

Figura 2.6: Listado de atributos en `bluetoothctl`

Se ha probado con varios dispositivos Bluetooth, pero no se han encontrado servicios disponibles para explorar. En la Figura 2.7 se muestra otro intento con un dispositivo diferente, pero nuevamente no se encontraron servicios disponibles.

```
[LinkBuds S]# list-attributes
5C:AD:BA:CD:4C:94  AC:80:0A:C0:7F:F6  C0:28:8D:9A:1B:AA
[LinkBuds S]# list-attributes
5C:AD:BA:CD:4C:94  AC:80:0A:C0:7F:F6  C0:28:8D:9A:1B:AA
[LinkBuds S]# list-attributes AC:80:0A:C0:7F:F6
[LinkBuds S]#
```

Figura 2.7: Otro intento de listado de atributos en `bluetoothctl`

3. Trabajando con Apps (nRF Connect)

Durante esta práctica se ha utilizado la aplicación nRF Connect para analizar el comportamiento de distintos dispositivos Bluetooth Low Energy (BLE) cercanos, centrándose finalmente en los auriculares `LE_LinkBuds S`. En la vista de escaneo se identificaron varios dispositivos (teléfonos, periféricos de Logitech y otros anunciados como `N/A`), observándose sus niveles de señal (RSSI) y comprobando cuáles eran conectables. En el caso concreto de `LE_LinkBuds S`, el RSSI se mantuvo alrededor de -55 dBm, lo que indica una proximidad física relativamente cercana y una conexión estable, como se aprecia en la gráfica temporal de la señal.

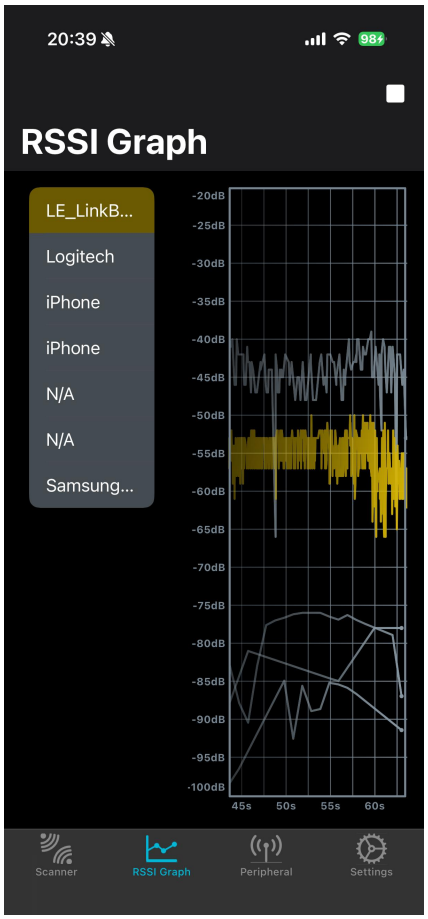

Figura 3.1: Gráfica de RSSI en nRF Connect para varios dispositivos BLE, destacando `LE_LinkBuds S`.

Una vez establecida la conexión con `LE_LinkBuds S`, se procedió a la exploración de la información que ofrece la aplicación sobre el dispositivo. En la ficha del dispositivo se muestra que es conectable, se identifica como tipo *Google* y anuncia, entre otros, un servicio con UUID corto `FE03`, además del nivel de señal y la latencia de los paquetes.

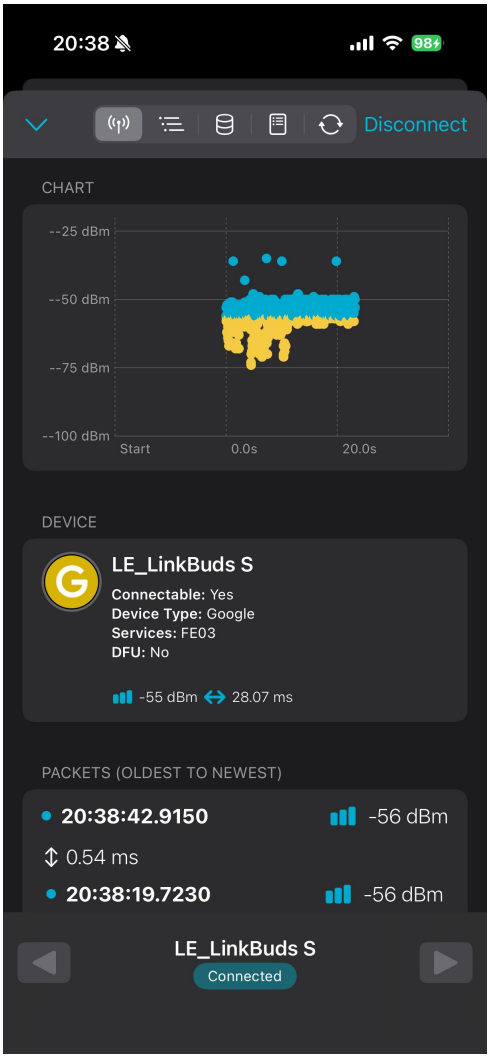

Figura 3.2: Detalle del dispositivo `LE_LinkBuds S`: estado de conexión, tipo de dispositivo, servicios anunciados y RSSI.

En la pestaña de *Attribute Table* del servidor GATT de nRF Connect se observa que, en esta sesión, el dispositivo móvil no está suscrito a ninguna característica del servidor, por lo que no se reciben notificaciones ni indicaciones automáticas desde la app actuando como periférico. Esta vista resulta útil para comprobar de un vistazo si existen suscripciones activas o si todas las operaciones se realizan únicamente mediante lecturas y escrituras explícitas.

Figura 3.3: Vista del servidor GATT en nRF Connect iOS indicando que no hay características suscritas.

Por otro lado, al inspeccionar la información del dispositivo remoto en nRF Connect se observa, en primer lugar, la sección *Advertised Services*, donde el auricular anuncia un servicio con UUID corto `FE03` junto con los servicios genéricos estándar `Generic Access` (UUID 0x1800) y `Generic Attribute` (UUID 0x1801). A continuación, en la tabla de atributos GATT que se obtiene tras la conexión, aparecen varios servicios propietarios con UUIDs de 128 bits etiquetados como *Unknown Service*, cada uno de ellos con características asociadas (*Unknown Characteristic*) sobre las que es posible consultar propiedades como `Read`, `Write` o `Notify` para inferir su posible función, aun sin documentación pública específica.

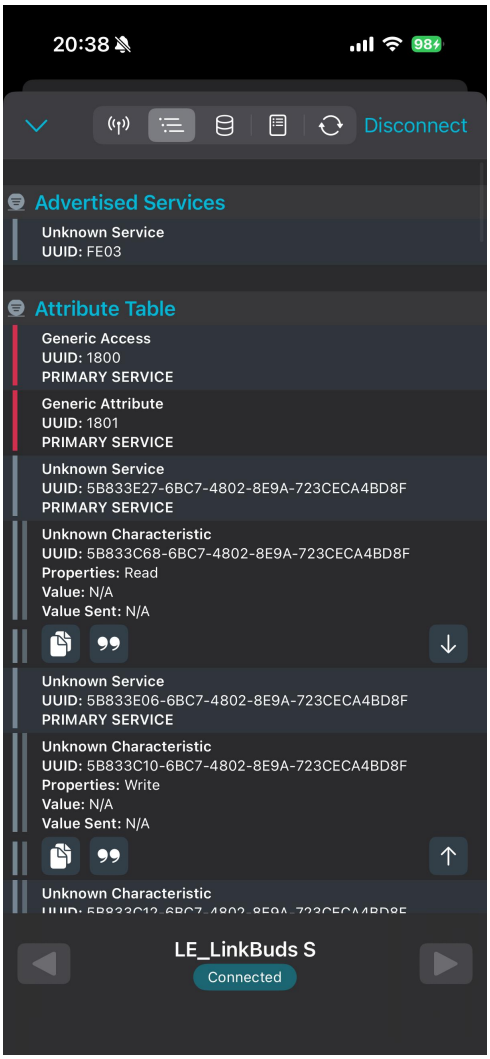

Figura 3.4: Sección *Advertised Services* y tabla de servicios GATT del dispositivo `LE_LinkBuds S` en nRF Connect.

En varias de estas características se probaron operaciones de lectura y escritura, utilizando los parsers de datos de nRF Connect (hexadecimal, enteros con signo y sin signo, booleanos y texto UTF-8) para interpretar los valores intercambiados. Algunas características se mapearon a plantillas internas de la aplicación, como *Heart Rate Sensor Location* o *LED and Button State*, lo que permite representar el mismo dato de distintas formas sin modificar el valor almacenado y facilita la comprensión práctica del modelo GATT basado en servicios y características.

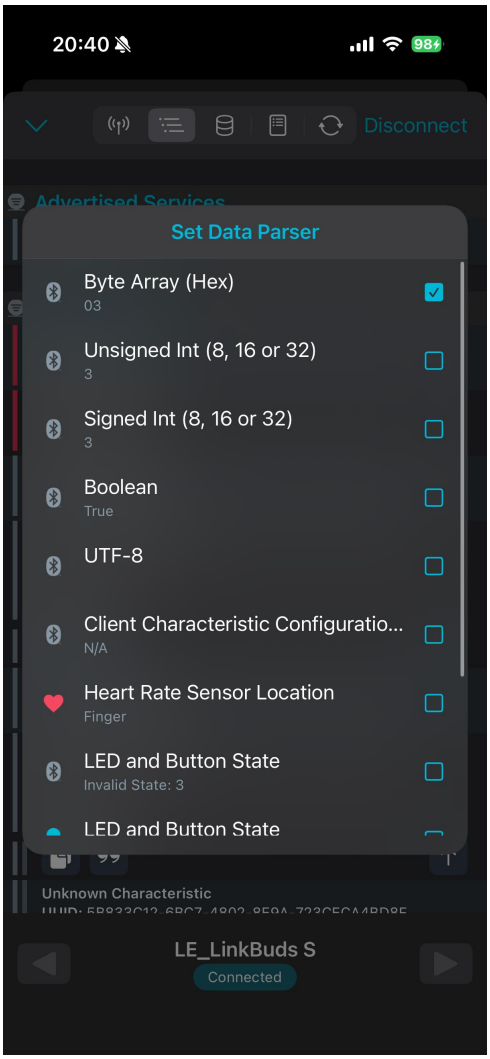

Figura 3.5: Selección de distintos parsers de datos para interpretar el valor de una característica GATT en nRF Connect.