

# PRACTICA 5 - Implementación de un servidor GATT basado en tablas

---

En la presente práctica vamos a llevar a cabo la implementación de un servidor GATT basado en tablas, para lo cual necesitaremos especificar dos partes:

- **El servidor BLE:** Será implementado en nuestra SoC STM32 mediante el ejemplo `examples/bluetooth/bluedroid/ble/gatt_server_service_table`.
- **El cliente BLE:** Será desarrollado mediante un equipo de sobremesa Linux a través de la herramienta `gatttool` en línea de comandos.

## Escaneado de dispositivos Bluetooth disponibles

### Tarea Básica

Edita el fichero `main/gatts_table_creat_demo.c` y modifica el nombre de tu dispositivo, que se enviará en cada anuncio emitido en la fase de advertising. Para ello, debes modificar el campo correspondiente de la estructura `raw_adv_data`. A continuación, compila y flashea el ejemplo, y comienza una sesión de escaneo de dispositivos BLE mediante la orden: `sudo hcitool lescan`. Deberás observar tu dispositivo en una de las líneas. Anota o recuerda su dirección MAC.

A continuación vamos a llevar a cabo el escaneo de los posibles servidores GATT disponibles mediante la interfaz Bluetooth de nuestro equipo Linux, para lo cual necesitaremos utilizar la herramienta de línea de comandos **hcitool**, la cual nos permitirá gestionar dicha interfaz y realizar un análisis de los dispositivos existentes. En el caso de que no se disponga de dicha herramienta, necesitaremos instalarla en nuestro equipo. En el siguiente [enlace](#) tendremos un breve resumen de las instrucciones para llevar a cabo la instalación en las principales distribuciones Linux.S

Antes de llevar a cabo el escaneo, necesitaremos ejecutar el ejemplo mencionado **gatt\_server\_service\_table** en nuestro SoC, para lo cual tenemos que modificar el nombre con el que dicho servidor se publicitará, de modo que podamos identificarlo sin problemas. En los siguientes cuadros podemos ver las modificaciones necesarias para cambiar el nombre del servidor GATT, las cuales se encuentran en la parte inicial del fichero **gatts\_table\_creat\_demo.c**.

```
#define SAMPLE_DEVICE_NAME      "ESP_GATTS_RPI1"    //"ESP_GATTS_DEMO"
.
.
.
static uint8_t raw_adv_data[] = {
    /* flags */
    0x02, 0x01, 0x06,
    /* tx power*/
    0x02, 0x0a, 0xeb,
    /* service uuid */
    0x03, 0x03, 0xFF, 0x00,
```

```

        /* device name */
        //0x0f, 0x09, 'E', 'S', 'P', '_', 'G', 'A', 'T', 'T', 'S', '_', 'D', 'E',
'M', 'O'
        0x0f, 0x09, 'E', 'S', 'P', '_', 'G', 'A', 'T', 'T', 'S', '_', 'R', 'P',
'I', 'I'
    };

```

Tras la configuración pasamos a ejecutar el ejemplo podemos obtener la siguiente salida, donde vemos como se ha comenzado el envío de los mensajes de anuncio **advertising**:

```

I (573) main_task: Calling app_main()
I (613) BTDM_INIT: BT controller compile version [946b762]
I (623) BTDM_INIT: Bluetooth MAC: 24:0a:c4:ea:36:b6
I (623) phy_init: phy_version 4670,719f9f6, Feb 18 2021,17:07:07
I (1043) main_task: Returned from app_main()
I (1043) GATTS_TABLE_DEMO: create attribute table successfully, the number handle
= 8

I (1053) GATTS_TABLE_DEMO: SERVICE_START_EVT, status 0, service_handle 40
I (1063) GATTS_TABLE_DEMO: advertising start successfully

```

Una vez hecho esto, ejecutaremos un escaneo de dispositivos Bluetooth con la herramienta **hcitool** con el objetivo de verificar que nuestro dispositivo se encuentra corriendo adecuadamente. Para lo cual ejecutaremos la la orden `sudo hcitool lescan | grep "RPI1"`, filtrando el resultado para obtener así un resultado más limpio.

Recordar que para poder llevar a cabo el escaneo correctamente, nuestro equipo tienen que tener activada la tecnología Bluetooth. En la siguiente imagen podemos ver como se detecta correctamente los mensajes de anuncio de nuestro SoC y la dirección MAC con el que este se está anunciando.

```

debian12:~$ sudo hcitool lescan | grep "RPI1"
24:0A:C4:EA:36:B6 ESP_GATTS_RPI1

```

## Interaccionando con nuestro servidor GATT

En el presente apartado realizaremos diferentes interacciones con nuestro servidor GATT mediante el uso de la herramienta en línea de comandos **gatttool**. En el caso de no tenerla instalada previamente, en el siguiente (enlace)[<https://command-not-found.com/gatttool>] podemos ver un breve resumen de las instrucciones para llevar a cabo la instalación en las principales distribuciones Linux.

Para un mejor entendimiento de las acciones realizadas, dividiremos el presente apartado en pasos sucesivos.

## PASO 1: Establecimiento de conexión y consulta de características

### Tarea Básica

Mediante el comando correspondiente ( `characteristics` ), consulta y anota las características disponibles en tu servidor GATT.

Una vez instalada la herramienta, realizaremos la conexión con nuestro servidor GATT mediante la orden `gatttool -b "direccionMAC" -I`, el cual iniciará el intérprete de instrucciones en línea de comandos y cuya salida podremos ver en el siguiente cuadro, donde se ha llevado a cabo la conexión:

```
mario@debian12:~$ gatttool -b 24:0A:C4:EA:36:B6 -I
[24:0A:C4:EA:36:B6][LE]> connect
Attempting to connect to 24:0A:C4:EA:36:B6
Connection successful
[24:0A:C4:EA:36:B6][LE]>
```

De la misma manera, cuando llevamos acabo la conexión también obtenemos una notificación en nuestro servidor GATT, en el cual podemos ver la siguiente salida, la cual nos notifica que se hay un nuevo cliente conectado:

```
I (153232) GATTS_TABLE_DEMO: advertising start successfully
I (158682) GATTS_TABLE_DEMO: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (158682) GATTS_TABLE_DEMO: 7e 67 4a 19 24 4d
I (159082) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 16,
max_int = 32,conn_int = 24,latency = 0, timeout = 400
I (159362) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 0,
max_int = 0,conn_int = 6,latency = 0, timeout = 500
I (159562) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 0,
max_int = 0,conn_int = 24,latency = 0, timeout = 400
```

Una vez realizado esto, podemos determinar que la conexión ha sido establecida de exitosamente, por lo que ahora podremos examinar todas y cada una de las características disponibles en nuestro servidor GATT mediante la orden `characteristics`, enviada desde el interprete de comandos en nuestro sistema. En el siguiente cuadro podemos ver una salida de la ejecución de dicha orden:

```
[24:0A:C4:EA:36:B6][LE]> characteristics
handle: 0x0002, char properties: 0x20, char value handle: 0x0003, uuid: 00002a05-0000-1000-8000-00805f9b34fb
handle: 0x0005, char properties: 0x0a, char value handle: 0x0006, uuid: 00002b29-0000-1000-8000-00805f9b34fb
handle: 0x0007, char properties: 0x02, char value handle: 0x0008, uuid: 00002b2a-0000-1000-8000-00805f9b34fb
handle: 0x0015, char properties: 0x02, char value handle: 0x0016, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0017, char properties: 0x02, char value handle: 0x0018, uuid: 00002a01-0000-1000-8000-00805f9b34fb
```

```

handle: 0x0019, char properties: 0x02, char value handle: 0x001a, uuid: 00002aa6-
0000-1000-8000-00805f9b34fb
handle: 0x0029, char properties: 0x1a, char value handle: 0x002a, uuid: 0000ff01-
0000-1000-8000-00805f9b34fb
handle: 0x002c, char properties: 0x02, char value handle: 0x002d, uuid: 0000ff02-
0000-1000-8000-00805f9b34fb
handle: 0x002e, char properties: 0x08, char value handle: 0x002f, uuid: 0000ff03-
0000-1000-8000-00805f9b34fb
[24:0A:C4:EA:36:B6][LE]>

```

### Tarea Adicional

El manejador que permite leer desde la característica "Heart Rate Value" tiene un manejador de tipo char asociado. Anota su valor.

Para poder determinar cual es el valor del UUID que corresponde a la característica **Heart Rate Value** necesitaremos examinar las variables variables definidas dentro del fichero **gatts\_table\_creat\_demo.c**, teniendo en cuenta que se corresponde con la definida como **GATTS\_CHAR\_UUID\_TEST\_A**. En el siguiente cuadro podemos ver la definición de la variable en cuestión y de aquellas adyacentes.

```

/* Service */
static const uint16_t GATTS_SERVICE_UUID_TEST = 0x00FF;
static const uint16_t GATTS_CHAR_UUID_TEST_A = 0xFF01;
static const uint16_t GATTS_CHAR_UUID_TEST_B = 0xFF02;

```

Teniendo esto en cuenta y en base a la tabla de características devueltas anteriormente por el servidor GATT, podemos ver como el UUID asociado a dicha variable es **0000ff01-0000-1000-8000-00805f9b34fb** y su handler, el cual utilizaremos más adelante y es necesario para interactuar con la misma, es **0x002a**

## PASO 2: Lectura y escritura de una característica

### Tarea Básica

Obtén los datos de lectura de la característica de medición del valor de monitorización de ritmo cardíaco. ¿Cuáles son? Deberías observar un valor de retorno de cuatro bytes con valor 0x00. Estos valores corresponden a los de la variable char\_value de tu código. Modifícalos, recompila y vuelve a flashear el código. ¿Han variado?

A continuación vamos a leer el valor asociado a la característica **Heart Rate Value** mediante el código de su manejador, para lo cual utilizaremos la instrucción **char-read-hnd "codigoManejador"**. Teniendo en cuenta que el valor de esta característica esta asociada a la variable **char\_value** de nuestro código, deberemos obtener el mismo resultado a la hora de llevar a cabo la consulta. En los siguientes cuadros podemos ver el valor de dicha variable en el código y el resultado obtenido al ejecutar la instrucción desde nuestro sistema, respectivamente:

```

static const uint8_t char_value[4] = {0x11, 0x22, 0x33, 0x44};

```

```
[24:0A:C4:EA:36:B6][LE]> char-read-hnd 0x002A
Characteristic value/descriptor: 11 22 33 44
```

Al llevar a cabo esto, nuestro SoC detecta la petición de lectura realizada y nos informa de ello mediante la salida del puerto serie. En el siguiente cuadro podemos ver dicha salida:

```
I (283472) GATTS_TABLE_DEMO: ESP_GATTS_READ_EVT
```

### Tarea Básica

Intenta ahora escribir en la anterior característica. Para ello, utiliza el comando `char-write-cmd` handler valor , siendo valor, por ejemplo, 11223344 .

Antes de pasar al escribir en la característica, vamos a modificar el valor asociada a la misma directamente desde el fichero **gatts\_table\_creat\_demo.c** y comprobar que ese se ha actualizado correctamente al llevar a cabo una nueva solicitud de lectura. Pera esto será necesario volver a establecer la conexión con nuestro servidor.

En los siguientes cuadros podemos el nuevo valor asociado a la variable y como este coincide con la respuesta obtenida al volver a llevar a cabo la petición de lectura desde nuestro sistema cliente:

```
static const uint8_t char_value[4] = {0x55, 0x66, 0x77, 0x88};
```

```
mario@debian12:~$ gatttool -b 24:0A:C4:EA:36:B6 -I
[24:0A:C4:EA:36:B6][LE]> connect
Attempting to connect to 24:0A:C4:EA:36:B6
Connection successful
[24:0A:C4:EA:36:B6][LE]> char-read-hnd 0x002A
Characteristic value/descriptor: 55 66 77 88
```

A continuación vamos a realizar la escritura del valor asociado a dicha característica mediante la instrucción `char-write-cmd "manejador" "nuevoValor"`, siendo el valor utilizado como ejemplo **12345678**. En los siguientes cuadros podemos ver tanto la respuesta generado por nuestro sistema cliente como la notificación que nos muestra el servidor GATT al recibir la petición de escritura, respectivamente:

```
[24:0A:C4:EA:36:B6][LE]> char-write-cmd 0x002A 12345678
[24:0A:C4:EA:36:B6][LE]> char-read-hnd 0x002A
Characteristic value/descriptor: 12 34 56 78
```

```
I (409322) GATTS_TABLE_DEMO: GATT_WRITE_EVT, handle = 42, value len = 4, value :
I (409322) GATTS_TABLE_DEMO: 12 34 56 78
```

### PASO 3: Modificación de una característica de configuración

#### Tarea Básica

Intenta ahora escribir en la característica de configuración. Para ello, utiliza el comando `char-write-cmd` `handler valor`, siendo `valor`, por ejemplo, `0100`.

En el paso anterior hemos leído y modificado el valor asociado a la característica **Heart Rate Value**, cuyo manejador esta asociado al valor **0x002A**. A continuación vamos a modificar la configuración asociada a dicha característica.

Si miramos dentro del fichero **gatts\_table\_creat\_demo.c** podremos ver la sección de definición de características dentro de un array del tipo **esp\_gatts\_attr\_db\_t**, en el siguiente cuadro podemos ver tanto la definición e introducción en dicho array de la característica en cuestión, como de su valor de configuración. También podemos fijarnos como ambas tienen asociados los permisos de lectura y escritura.

```
/* Characteristic Value */
[IDX_CHAR_VAL_A] =
{{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&GATTS_CHAR_UUID_TEST_A,
ESP_GATT_PERM_READ | ESP_GATT_PERM_WRITE,
    GATTS_DEMO_CHAR_VAL_LEN_MAX, sizeof(char_value), (uint8_t *)char_value}},

/* Client Characteristic Configuration Descriptor */
[IDX_CHAR_CFG_A] =
{{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&character_client_config_uuid,
ESP_GATT_PERM_READ | ESP_GATT_PERM_WRITE,
    sizeof(uint16_t), sizeof(heart_measurement_ccc), (uint8_t
*)heart_measurement_ccc}},
```

Para poder visualizar todas las características disponibles en el servidor, incluyendo las de configuración, podemos utilizar la orden **char-desc**. A continuación podemos ver el resultado obtenido al llevar a cabo dicha ejecución:

```
[24:0A:C4:EA:36:B6][LE]> char-desc
handle: 0x0001, uuid: 00002800-0000-1000-8000-00805f9b34fb
handle: 0x0002, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0003, uuid: 00002a05-0000-1000-8000-00805f9b34fb
handle: 0x0004, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x0005, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0006, uuid: 00002b29-0000-1000-8000-00805f9b34fb
handle: 0x0007, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0008, uuid: 00002b2a-0000-1000-8000-00805f9b34fb
handle: 0x0014, uuid: 00002800-0000-1000-8000-00805f9b34fb
```

```

handle: 0x0015, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0016, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0017, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x0018, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x0019, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x001a, uuid: 00002aa6-0000-1000-8000-00805f9b34fb
handle: 0x0028, uuid: 00002800-0000-1000-8000-00805f9b34fb
handle: 0x0029, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x002a, uuid: 0000ff01-0000-1000-8000-00805f9b34fb
handle: 0x002b, uuid: 00002902-0000-1000-8000-00805f9b34fb
handle: 0x002c, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x002d, uuid: 0000ff02-0000-1000-8000-00805f9b34fb
handle: 0x002e, uuid: 00002803-0000-1000-8000-00805f9b34fb
handle: 0x002f, uuid: 0000ff03-0000-1000-8000-00805f9b34fb

```

Una vez visto esto, y teniendo en cuenta que la característica de configuración asociada a **Heart Rate Value** se corresponde con el siguiente UUID al de la misma, esto quiere decir que podremos interactuar con esta a través de la dirección **0x002b**.

A continuación vamos a llevar a cabo tanto una lectura como una escritura en dicha característica, para lo cual utilizaremos las ordenes **char-read-hnd "codigoManejador"** y **char-write-cmd "codigoManejador"** respectivamente. En el siguiente cuadro podemos ver como el valor inicial es **0000**, sin embargo, nosotros realizaremos un par de escrituras para ver algunas de las diferentes modificaciones que podemos hacer sobre dicha configuración, siendo los valores **0100** y **0200**

```

[24:0A:C4:EA:36:B6][LE]> char-read-hnd 0x002B
Characteristic value/descriptor: 00 00
[24:0A:C4:EA:36:B6][LE]> char-write-cmd 0x002B 0100
Notification handle = 0x002a value: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
[24:0A:C4:EA:36:B6][LE]> char-write-cmd 0x002B 0200
Indication handle = 0x002a value: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
[24:0A:C4:EA:36:B6][LE]> char-write-cmd 0x002B 0100
Notification handle = 0x002a value: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
[24:0A:C4:EA:36:B6][LE]> char-read-hnd 0x002B
Characteristic value/descriptor: 01 00

```

Además de ver como la respuesta del cliente nos indica que la escritura se ha realizado exitosamente, en el siguiente cuadro también podemos visualizar el mensaje de notificación enviado por el servidor GATT:

```

I (42802) GATTS_TABLE_DEMO: ESP_GATTS_READ_EVT
I (66642) GATTS_TABLE_DEMO: GATT_WRITE_EVT, handle = 43, value len = 2, value :
I (66642) GATTS_TABLE_DEMO: 01 00
I (66642) GATTS_TABLE_DEMO: notify enable
I (66652) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (328642) GATTS_TABLE_DEMO: GATT_WRITE_EVT, handle = 43, value len = 2, value :
I (328652) GATTS_TABLE_DEMO: 02 00
I (328652) GATTS_TABLE_DEMO: indicate enable
I (328722) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42

```

```
I (346642) GATTS_TABLE_DEMO: GATT_WRITE_EVT, handle = 43, value len = 2, value :  
I (346652) GATTS_TABLE_DEMO: 01 00  
I (346652) GATTS_TABLE_DEMO: notify enable  
I (346652) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
```

## PASO 4: Configurar una notificación

Para poder activar las notificaciones una vez conectado al dispositivo BLE se han modificado distintos apartados:

### PASO 1: Guardar el valor de `IDX_CHAR_CFG_A`

Una vez conectado, el usuario puede activar las notificaciones escribiendo sobre la característica de configuración con el comando

```
char-write-cmd 0x002b 0100
```

Para guardar el valor dentro de la característica se ha añadido el siguiente código dentro de la función `gatts_profile_event_handler` para tratar el evento `ESP_GATTS_WRITE_EVT`:

```
esp_gatt_status_t status =  
esp_ble_gatts_set_attr_value(heart_rate_handle_table[IDX_CHAR_CFG_A], param-  
>write.len, param->write.value);  
if (status == ESP_GATT_OK)  
{  
    ESP_LOGI("APP", "IDX_CHAR_CFG_A written value: %02X %02X", param-  
>write.value[0], param->write.value[1]);  
}  
else  
{  
    ESP_LOGE("APP", "Error writen value for IDX_CHAR_CFG_A, error code: 0x%x",  
status);  
}
```

### PASO 2: Registrar la función `publish_data_task` dentro del tratamiento del evento

`ESP_GATTS_CONNECT_EVT` ante la conexión de un nuevo usuario

```
ESP_LOGI(GATTS_TABLE_TAG, "Creating publish_data_task...");  
xTaskCreate(&publish_data_task, "publish_data_task", 4096, NULL, 5,  
&heartBeatHandle);
```

### PASO 3: Crear la función `publish_data_task`



Haciendo uso de las funciones `esp_ble_gatts_get_attr_value` y `esp_ble_gatts_set_attr_value` se realizan las siguientes acciones:

- Obtener el valor en curso de `IDX_CHAR_VAL_A`
- Actualizar el byte 1 con un valor aleatorio
- Guardar el varlo `IDX_CHAR_VAL_A`
- Obtener el valor de configuración `IDX_CHAR_CFG_A`
- Si tiene las notificaciones activadas, obtener el valor actualizado de `IDX_CHAR_VAL_A` y enviarlo al usuario

#### PASO 4: Tratamiento del evento de desconexión `ESP_GATTS_DISCONNECT_EVT`

En caso de deconexión del cliente realizamos las siguientes acciones:

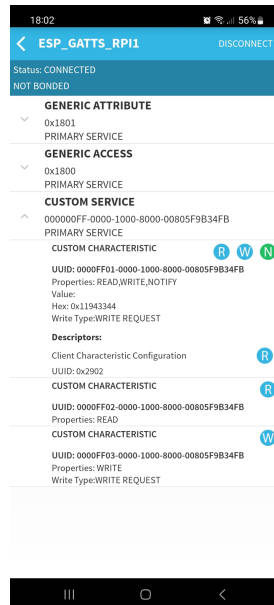
- Borramos la tarea periodica `publish_data_task`
- Volvemos a dejar el valor de `IDX_CHAR_CFG_A` a `0x0000`

#### PASO 5: Ejemplo de salida en el cliente linux usando `gatttool`

A continuación tenemos un cuadro donde se encuentra la conexión con el dispositivo y la ejecución de la orden de escritura en la característica de configuración del ritmo cardíaco. Podemos ver como la recepción de mensajes se lleva a cabo de manera periódica.

```
gatttool -b 94:3C:C6:CD:BB:4E -I
[94:3C:C6:CD:BB:4E][LE]> connect
Attempting to connect to 94:3C:C6:CD:BB:4E
Connection successful
[94:3C:C6:CD:BB:4E][LE]> char-write-cmd 0x002b 0100
Notification handle = 0x002a value: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
Notification handle = 0x002a value: 22 41 22 22
Notification handle = 0x002a value: 22 99 22 22
Notification handle = 0x002a value: 22 8c 22 22
```

#### PASO 6: Ejemplo de salide del dispositivo conectando con el movil y la aplicación BLE Scanner



```

I (573) main_task: Calling app_main()
I (593) BTDM_INIT: BT controller compile version [946b762]
I (593) BTDM_INIT: Bluetooth MAC: 94:3c:c6:0d:bb:4e
I (603) phy_init: phy_version 4670,719f9f6, Feb 18 2021,17:07:07
I (1033) main_task: Returned from app_main()
I (1043) GATTS_TABLE_DEMO: create attribute table successfully, the number handle
= 8
I (1043) GATTS_TABLE_DEMO: SERVICE_START_EVT, status 0, service_handle 40
I (1053) GATTS_TABLE_DEMO: advertising start successfully
I (4753) GATTS_TABLE_DEMO: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (4753) GATTS_TABLE_DEMO: 5f b2 d5 67 42 17
I (4753) GATTS_TABLE_DEMO: Creating publish_data_task...
I (4753) APP: Value retrieved for IDX_CHAR_VAL_A: 11 22 33 44
I (4763) APP: Value retrieved 44332211...
I (4763) APP: Value generated 44333411...
I (4773) APP: Update value for IDX_CHAR_VAL_A: 11 34 33 44
I (4783) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (4783) APP: Notifications disabled, not sending data...
I (5143) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 16,
max_int = 32, conn_int = 6, latency = 0, timeout = 500
I (5423) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 0,
max_int = 0, conn_int = 24, latency = 0, timeout = 400
I (5793) APP: Value retrieved for IDX_CHAR_VAL_A: 11 34 33 44
I (5793) APP: Value retrieved 44333411...
I (5793) APP: Value generated 4433F011...
I (5793) APP: Update value for IDX_CHAR_VAL_A: 11 F0 33 44
I (5803) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (5803) APP: Notifications disabled, not sending data...
I (6813) APP: Value retrieved for IDX_CHAR_VAL_A: 11 F0 33 44
I (6813) APP: Value retrieved 4433F011...
I (6813) APP: Value generated 4433EE11...
I (6813) APP: Update value for IDX_CHAR_VAL_A: 11 EE 33 44
I (6823) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (6823) APP: Notifications disabled, not sending data...
I (7833) APP: Value retrieved for IDX_CHAR_VAL_A: 11 EE 33 44
I (7833) APP: Value retrieved 4433EE11...

```

```
I (7833) APP: Value generated 4433B011...
I (7833) APP: Update value for IDX_CHAR_VAL_A: 11 B0 33 44
I (7843) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (7843) APP: Notifications disabled, not sending data...
I (8853) APP: Value retrieved for IDX_CHAR_VAL_A: 11 B0 33 44
I (8853) APP: Value retrieved 4433B011...
I (8853) APP: Value generated 4433F311...
I (8853) APP: Update value for IDX_CHAR_VAL_A: 11 F3 33 44
I (8863) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (8863) APP: Notifications disabled, not sending data...
I (9873) APP: Value retrieved for IDX_CHAR_VAL_A: 11 F3 33 44
I (9873) APP: Value retrieved 4433F311...
I (9873) APP: Value generated 44334211...
I (9873) APP: Update value for IDX_CHAR_VAL_A: 11 42 33 44
I (9883) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (9883) APP: Notifications disabled, not sending data...
I (10893) APP: Value retrieved for IDX_CHAR_VAL_A: 11 42 33 44
I (10893) APP: Value retrieved 44334211...
I (10893) APP: Value generated 4433A211...
I (10893) APP: Update value for IDX_CHAR_VAL_A: 11 A2 33 44
I (10903) APP: Retrieved value for IDX_CHAR_CFG_A: 00 00
I (10903) APP: Notifications disabled, not sending data...
I (11853) GATTS_TABLE_DEMO: GATT_WRITE_EVT, handle = 43, value len = 2, value :
I (11853) GATTS_TABLE_DEMO: 01 00
I (11853) GATTS_TABLE_DEMO: notify enable
I (11863) APP: IDX_CHAR_CFG_A written value: 01 00
I (11873) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (11913) APP: Value retrieved for IDX_CHAR_VAL_A: 11 A2 33 44
I (11913) APP: Value retrieved 4433A211...
I (11913) APP: Value generated 4433DC11...
I (11913) APP: Update value for IDX_CHAR_VAL_A: 11 DC 33 44
I (11923) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (11923) APP: Notifications enabled, sending data...
I (11933) APP: Retrieving value for IDX_CHAR_VAL_A: 11 DC 33 44
I (11943) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (11943) APP: Data sent successfully: 11 DC 33 44
I (12953) APP: Value retrieved for IDX_CHAR_VAL_A: 11 DC 33 44
I (12953) APP: Value retrieved 4433DC11...
I (12953) APP: Value generated 4433E711...
I (12953) APP: Update value for IDX_CHAR_VAL_A: 11 E7 33 44
I (12963) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (12963) APP: Notifications enabled, sending data...
I (12973) APP: Retrieving value for IDX_CHAR_VAL_A: 11 E7 33 44
I (12983) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (12983) APP: Data sent successfully: 11 E7 33 44
I (13993) APP: Value retrieved for IDX_CHAR_VAL_A: 11 E7 33 44
I (13993) APP: Value retrieved 4433E711...
I (13993) APP: Value generated 44336311...
I (13993) APP: Update value for IDX_CHAR_VAL_A: 11 63 33 44
I (14003) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (14003) APP: Notifications enabled, sending data...
I (14013) APP: Retrieving value for IDX_CHAR_VAL_A: 11 63 33 44
I (14023) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (14023) APP: Data sent successfully: 11 63 33 44
```

```
I (15033) APP: Value retrieved for IDX_CHAR_VAL_A: 11 63 33 44
I (15033) APP: Value retrieved 44336311...
I (15033) APP: Value generated 44334811...
I (15033) APP: Update value for IDX_CHAR_VAL_A: 11 48 33 44
I (15043) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (15043) APP: Notifications enabled, sending data...
I (15053) APP: Retrieving value for IDX_CHAR_VAL_A: 11 48 33 44
I (15063) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (15063) APP: Data sent successfully: 11 48 33 44
I (16073) APP: Value retrieved for IDX_CHAR_VAL_A: 11 48 33 44
I (16073) APP: Value retrieved 44334811...
I (16073) APP: Value generated 44339411...
I (16073) APP: Update value for IDX_CHAR_VAL_A: 11 94 33 44
I (16083) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (16083) APP: Notifications enabled, sending data...
I (16093) APP: Retrieving value for IDX_CHAR_VAL_A: 11 94 33 44
I (16103) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (16103) APP: Data sent successfully: 11 94 33 44
I (17113) APP: Value retrieved for IDX_CHAR_VAL_A: 11 94 33 44
I (17113) APP: Value retrieved 44339411...
I (17113) APP: Value generated 4433E211...
I (17113) APP: Update value for IDX_CHAR_VAL_A: 11 E2 33 44
I (17123) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (17123) APP: Notifications enabled, sending data...
I (17133) APP: Retrieving value for IDX_CHAR_VAL_A: 11 E2 33 44
I (17143) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (17143) APP: Data sent successfully: 11 E2 33 44
I (18153) APP: Value retrieved for IDX_CHAR_VAL_A: 11 E2 33 44
I (18153) APP: Value retrieved 4433E211...
I (18153) APP: Value generated 4433F911...
I (18153) APP: Update value for IDX_CHAR_VAL_A: 11 F9 33 44
I (18163) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (18163) APP: Notifications enabled, sending data...
I (18173) APP: Retrieving value for IDX_CHAR_VAL_A: 11 F9 33 44
I (18183) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (18183) APP: Data sent successfully: 11 F9 33 44
I (19193) APP: Value retrieved for IDX_CHAR_VAL_A: 11 F9 33 44
I (19193) APP: Value retrieved 4433F911...
I (19193) APP: Value generated 44339111...
I (19193) APP: Update value for IDX_CHAR_VAL_A: 11 91 33 44
I (19203) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (19203) APP: Notifications enabled, sending data...
I (19213) APP: Retrieving value for IDX_CHAR_VAL_A: 11 91 33 44
I (19223) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (19223) APP: Data sent successfully: 11 91 33 44
I (20233) APP: Value retrieved for IDX_CHAR_VAL_A: 11 91 33 44
I (20233) APP: Value retrieved 44339111...
I (20233) APP: Value generated 4433DF11...
I (20233) APP: Update value for IDX_CHAR_VAL_A: 11 DF 33 44
I (20243) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (20243) APP: Notifications enabled, sending data...
I (20253) APP: Retrieving value for IDX_CHAR_VAL_A: 11 DF 33 44
I (20263) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (20263) APP: Data sent successfully: 11 DF 33 44
```

```
I (21273) APP: Value retrieved for IDX_CHAR_VAL_A: 11 DF 33 44
I (21273) APP: Value retrieved 4433DF11...
I (21273) APP: Value generated 44335311...
I (21273) APP: Update value for IDX_CHAR_VAL_A: 11 53 33 44
I (21283) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (21283) APP: Notifications enabled, sending data...
I (21293) APP: Retrieving value for IDX_CHAR_VAL_A: 11 53 33 44
I (21303) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (21303) APP: Data sent successfully: 11 53 33 44
I (22313) APP: Value retrieved for IDX_CHAR_VAL_A: 11 53 33 44
I (22313) APP: Value retrieved 44335311...
I (22313) APP: Value generated 44333911...
I (22313) APP: Update value for IDX_CHAR_VAL_A: 11 39 33 44
I (22323) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 0,
max_int = 0,conn_int = 6,latency = 0, timeout = 500
I (22323) APP: Retrieved value for IDX_CHAR_CFG_A: 01 00
I (22343) APP: Notifications enabled, sending data...
I (22343) APP: Retrieving value for IDX_CHAR_VAL_A: 11 39 33 44
I (22353) APP: Data sent successfully: 11 39 33 44
I (22353) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (22433) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 0,
max_int = 0,conn_int = 24,latency = 0, timeout = 400
W (23163) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (23163) GATTS_TABLE_DEMO: ESP_GATTS_DISCONNECT_EVT, reason = 0x13
I (23163) APP: IDX_CHAR_CFG_A set to 0x0000
I (23183) GATTS_TABLE_DEMO: advertising start successfully``
```