

# PRÁCTICA 6 - Implementación de un cliente GATT

---

## Tarea previa

```
static const char remote_device_name[] = "ESP_GATTS_CLIENT_NOCONNECT";
```

## Tarea 1

Configura los parámetros de escaneo para que éste se produzca con menos frecuencia (e.g. 1 segundo o un valor superior). Para ello, adapta el valor del campo `scan_interval` con el valor apropiado.

Para configurar el varlo de `scan_interval` a 4 segundos consultamos la documentación oficial en [este enlace](#).

En este caso 4 segundos = 4000 ms = 6400 \* 0.625. Convirtiendo este valor a hexadecimal obtenemos 0x1900, estableciendolo como valor por defecto en el parámetro `BLE_SCAN_INTERVAL` en sdkconfig.

```
static esp_ble_scan_params_t ble_scan_params = {
    .scan_type           = BLE_SCAN_TYPE_ACTIVE,
    .own_addr_type       = BLE_ADDR_TYPE_PUBLIC,
    .scan_filter_policy   = BLE_SCAN_FILTER_ALLOW_ALL,
    .scan_interval       = BLE_SCAN_LE_INTERVAL,
    .scan_window         = BLE_SCAN_LE_WINDOW,
    .scan_duplicate      = BLE_SCAN_DUPLICATE_DISABLE
};
```

## Tarea 2

El valor de duración del proceso de escaneo es configurable. En esta primera parte de la práctica, aumentalo para que el proceso sea más largo y así tengas más tiempo para analizar la salida producida. Comprueba que efectivamente el tiempo en el que el dispositivo se encuentra en fase de escaneo es el seleccionado.

Aumentamos el valor por defecto a 60 segundos para poder analizar la salida:

```
case ESP_GAP_BLE_SCAN_PARAM_SET_COMPLETE_EVT:
{
    // the unit of the duration is second
    uint32_t duration = 60;
    esp_ble_gap_start_scanning(duration);
    break;
}
```

## Tarea 3

Analiza el código y modifícalo para que el proceso de escaneo se produzca cíclicamente, con un parámetro de separación entre fases de escaneo definido a través de los menús de configuración. De la misma manera, añade a dichos menús un parámetro adicional que indique el intervalo entre eventos de escaneo.

Para conseguir que el proceso sea cíclico se han definido los siguientes parámetros en el fichero

`Kconfig.projbuild`:

```
config BLE_SCAN_LE_INTERVAL
    hex "Ble scan interval (Time = N * 0.625 msec Time Range: 2.5 msec to 10240 msec)"
    range 0x0004 0x4000
    default 0x1900

config BLE_SCAN_LE_WINDOW
    hex "The duration of the LE scan. LE_Scan_Window shall be less than or equal to LE_Scan_Interval (Time = N * 0.625 msec Time Range: 2.5 msec to 10240 msec)"
    range 0x0004 0x4000
    default 0x0010

config BLE_SCAN_DURATION
    int "Duration of a scan (s)"
    range 60 300
    default 60

config BLE_SCAN_DELAY
    int "Delay before starting new scan (s)"
    range 10 600
    default 30
```

Los parámetros `BLE_SCAN_LE_INTERVAL` y `BLE_SCAN_LE_WINDOW` se han definido como 4s y 1s respectivamente en la estructura `esp_ble_scan_params_t`.

```
static esp_ble_scan_params_t ble_scan_params = {
    .scan_type = BLE_SCAN_TYPE_ACTIVE,
    .own_addr_type = BLE_ADDR_TYPE_PUBLIC,
    .scan_filter_policy = BLE_SCAN_FILTER_ALLOW_ALL,
    .scan_interval = BLE_SCAN_LE_INTERVAL,
    .scan_window = BLE_SCAN_LE_WINDOW,
    .scan_duplicate = BLE_SCAN_DUPLICATE_DISABLE};
```

Por otro lado utilizamos `BLE_SCAN_DURATION` a la hora de iniciar un nuevo escaneo:

```
uint32_t duration = BLE_SCAN_DURATION; // the unit of the duration is second
esp_ble_gap_start_scanning(duration);
```

Una vez finalizado el escaneo, dentro del tratamiento del evento `ESP_GAP_SEARCH_INQ_CMPL_EVT` esperamos 30s segundos establecidos por defecto en el parámetro `BLE_SCAN_DELAY`:

```
case ESP_GAP_SEARCH_INQ_CMPL_EVT:
    // Wait BLE_SCAN_DELAY seconds
    ESP_LOGI(GATTC_TAG, "Waiting %d seconds before starting a new scan",
BLE_SCAN_DELAY);
    vTaskDelay(BLE_SCAN_DELAY * 1000 / portTICK_PERIOD_MS);
    // Initiate new scan
    ESP_LOGI(GATTC_TAG, "Starting a new %d seconds duration scan",
BLE_SCAN_DURATION);
    uint32_t duration = BLE_SCAN_DURATION; // the unit of the duration is
second
    esp_ble_gap_start_scanning(duration);
    break;
```

Resumiendo, los valores establecidos para los parámetros son los siguientes:

- BLE\_SCAN\_LE\_INTERVAL: 4s
- BLE\_SCAN\_LE\_WINDOW: 10ms
- BLE\_SCAN\_DURATION: 60s
- BLE\_SCAN\_DELAY: 30s

## Tarea 4

Muestra por pantalla el valor de RSSI de cada dispositivo BLE encontrado. Para ello, busca en la estructura de parámetros recibidos el campo correspondiente.

Examinamos los parámetros de la estructura `esp_ble_gap_cb_param_t` recibida dentro del tratamiento del evento `ESP_GAP_BLE_SCAN_RESULT_EVT` en la [documentación oficial](#) mostrando su valor:

```
case ESP_GAP_SEARCH_INQ_RES_EVT:
    ESP_LOGI(GATTC_TAG, "Device found");
    esp_log_buffer_hex(GATTC_TAG, scan_result->scan_rst.bda, 6);
    ESP_LOGI(GATTC_TAG, "searched Adv Data Len %d, Scan Response Len %d",
scan_result->scan_rst.adv_data_len, scan_result->scan_rst.scan_rsp_len);
    adv_name = esp_ble_resolve_adv_data(scan_result->scan_rst.ble_adv,
ESP_BLE_AD_TYPE_NAME_CMPL,
&adv_name_len);
    ESP_LOGI(GATTC_TAG, "searched Device Name Len %d", adv_name_len);
    ESP_LOGI(GATTC_TAG, "searched Device RSSI %d", scan_result-
>scan_rst.rssi);
    esp_log_buffer_char(GATTC_TAG, adv_name, adv_name_len);
```

Verificamos como el valor es distinto para cada dispositivo detectado en el escaneo:

```
I (237932) GATTC_CLIENT: Device found
I (237932) GATTC_CLIENT: bc 14 85 ea 9c 18
I (237932) GATTC_CLIENT: searched Adv Data Len 28, Scan Response Len 0
I (237932) GATTC_CLIENT: searched Device Name Len 0
I (237942) GATTC_CLIENT: searched Device RSSI -74
I (237942) GATTC_CLIENT:

I (237952) GATTC_CLIENT: Device found
I (237952) GATTC_CLIENT: 88 0f 10 8d df 4e
I (237962) GATTC_CLIENT: searched Adv Data Len 31, Scan Response Len 0
I (237962) GATTC_CLIENT: searched Device Name Len 0
I (237972) GATTC_CLIENT: searched Device RSSI -77
I (237982) GATTC_CLIENT:

I (238052) GATTC_CLIENT: Device found
I (238052) GATTC_CLIENT: bc 14 85 ea 9c 18
I (238052) GATTC_CLIENT: searched Adv Data Len 28, Scan Response Len 0
I (238062) GATTC_CLIENT: searched Device Name Len 0
I (238062) GATTC_CLIENT: searched Device RSSI -75
I (238072) GATTC_CLIENT:

I (241052) GATTC_CLIENT: Waiting 30 seconds before starting a new scan
```

## Tarea 5

Añade a los menús de configuración un campo que indique la dirección MAC BLE de un dispositivo(conocido) que esté en fase de anuncio. Puedes utilizar los códigos de servidor de la práctica anterior.

Para obtener la MAC podemos examinar la salida del servidor:

```
I (572) main_task: Calling app_main()
I (622) BTDM_INIT: BT controller compile version [946b762]
I (622) BTDM_INIT: Bluetooth MAC: 24:0a:c4:ea:28:c2
I (632) phy_init: phy_version 4670,719f9f6, Feb 18 2021,17:07:07
I (1042) main_task: Returned from app_main()
I (1042) GATTS_TABLE_DEMO: create attribute table successfully, the number handle
= 8

I (1052) GATTS_TABLE_DEMO: SERVICE_START_EVT, status 0, service_handle 40
I (1062) GATTS_TABLE_DEMO: advertising start successfully
```

Configuramos por tanto un nuevo parámetro en el fichero `Kconfig.projbuild` del cliente:

```

config MAC_KNOWN_DEVICE
    string "MAC for a known device on advertising state"
    default "24:0a:c4:ea:28:c2"
    help
        Specify a MAC address for the ethernet interface in the form of
        six hex 8-bit chars separated by colons (e.g.:
        aa:33:cc:22:e2:c0). The default is an empty string, which
        means the code will make 00:00:5E:00:53:XX, where XX will be
        random.

```

Añadiendo su definición en el fichero `gattc_demo.c` para poder ser utilizada más adelante:

```
#define MAC_KNOWN_DEVICE CONFIG_MAC_KNOWN_DEVICE
```

## Tarea 6

Modifica la fase de escaneo para que únicamente se muestre información (al menos nombre y RSSI) sobre el dispositivo cuya dirección MAC se haya configurado.

Para ello añadimos un nuevo filtro dentro del tratamiento del evento `ESP_GAP_SEARCH_INQ_RES_EVT`:

```

case ESP_GAP_SEARCH_INQ_RES_EVT:
    // MAC_KNOWN_DEVICE
    char macdetected[18];
    memset(&macdetected[0], 0, sizeof(macdetected)); // Clear array
    esp_bd_addr_to_string(scan_result->scan_rst.bda, macdetected);
    if (strcmp(macdetected, MAC_KNOWN_DEVICE) == 0)
    {
        adv_name = esp_ble_resolve_adv_data(scan_result->scan_rst.ble_adv,
                                            ESP_BLE_AD_TYPE_NAME_CMPL,
&adv_name_len);
        if (adv_name_len > 0)
        {
            ESP_LOGI(GATT_TAG, "MAC: %s | NAME: %s | RSSI: %d", macdetected,
adv_name, scan_result->scan_rst.rssi);
        }
    }
}

```

Podemos ver en la salida la información filtrada por la MAC introducida como parámetro:

```

I (6885072) GATT_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-50
I (6889072) GATT_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-48
I (6893092) GATT_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-45

```

```
I (6897082) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-50
I (6901062) GATTC_CLIENT: Waiting 30 seconds before starting a new scan
I (6931062) GATTC_CLIENT: Starting a new 60 seconds duration scan
I (6931062) GATTC_CLIENT: scan start success
I (6931092) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-49
I (6935082) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-45
I (6939082) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-50
I (6943072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-48
I (6947072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-46
I (6951072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-49
I (6955072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-46
I (6959092) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-46
I (6963092) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-49
I (6967072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-47
I (6971072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-45
I (6975072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-49
I (6983082) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-46
I (6987072) GATTC_CLIENT: MAC: 24:0a:c4:ea:28:c2 | NAME: ESP_GATTS_SERVER | RSSI:
-49
I (6991062) GATTC_CLIENT: Waiting 30 seconds before starting a new scan
```

## Tarea 7

Modifica el firmware básico para que, en función del valor de RSSI obtenido para el dispositivo de interés, el cliente reporte con más o menos frecuencia por pantalla un valor proporcional a la distancia supuesta para dicho dispositivo. Si dispones de un LED conectado a la placa, puedes hacer que la frecuencia de parpadeo del mismo sea proporcional a dicha distancia.

En este caso hemos bajado el valor del parámetro `BLE_SCAN_LE_INTERVAL` a 1s es decir 0x640 en hexadecimal.

Al final de la función `app_main` hemos inicializado el led:

```
// Configure led for blinking
ESP_LOGI(GATTC_TAG, "Configure blinking led!");
gpio_reset_pin(BLINK_GPIO);
```

```
/* Set the GPIO as a push/pull output */
gpio_set_direction(BLINK_GPIO, GPIO_MODE_OUTPUT);
```

Se ha establecido un parámetro estático `static int delay_blinking = 0;` y una función que realiza la operación de parpadeo y espera el tiempo establecido:

```
static void taskFunction_blinking(void *parameters)
{
    while (1)
    {
        if (delay_blinking > 0)
        {
            /* Set the GPIO level according to the state (LOW or HIGH)*/
            gpio_set_level(BLINK_GPIO, s_led_state);
            /* Toggle the LED state */
            s_led_state = !s_led_state;
            vTaskDelay(delay_blinking / portTICK_PERIOD_MS);
        }
    }
    vTaskDelete(NULL);
}
```

Este parámetro se calcula automáticamente en función del RSSI multiplicándolo por 4 para ver las diferencias en milisegundos:

```
delay_blinking = abs(scan_result->scan_rst.rssi) * 4;
esp_log_buffer_char(GATTC_TAG, adv_name, adv_name_len);
ESP_LOGI(GATTC_TAG, "MAC: %s", macdetected);
ESP_LOGI(GATTC_TAG, "RSSI: %d | Blinking delay %d ms", scan_result->scan_rst.rssi,
delay_blinking);
ESP_LOGI(GATTC_TAG, "-----");
```

Por último la salida sería como la siguiente:

```
I (3075) GATTC_CLIENT: -----
I (4055) GATTC_CLIENT: ESP_GATTS_SERV
I (4055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (4055) GATTC_CLIENT: RSSI: -50 | Blinking delay 200 ms
I (4055) GATTC_CLIENT: -----
I (6075) GATTC_CLIENT: ESP_GATTS_SERV
I (6075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (6075) GATTC_CLIENT: RSSI: -43 | Blinking delay 172 ms
I (6085) GATTC_CLIENT: -----
I (7065) GATTC_CLIENT: ESP_GATTS_SERV
I (7065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (7065) GATTC_CLIENT: RSSI: -49 | Blinking delay 196 ms
```

```
I (7065) GATTC_CLIENT: -----
I (8065) GATTC_CLIENT: ESP_GATTS_SERV
I (8065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (8065) GATTC_CLIENT: RSSI: -45 | Blinking delay 180 ms
I (8065) GATTC_CLIENT: -----
I (12065) GATTC_CLIENT: ESP_GATTS_SERV
I (12065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (12065) GATTC_CLIENT: RSSI: -44 | Blinking delay 176 ms
I (12065) GATTC_CLIENT: -----
I (14065) GATTC_CLIENT: ESP_GATTS_SERV
I (14065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (14065) GATTC_CLIENT: RSSI: -47 | Blinking delay 188 ms
I (14075) GATTC_CLIENT: -----
I (16075) GATTC_CLIENT: ESP_GATTS_SERV
I (16075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (16075) GATTC_CLIENT: RSSI: -48 | Blinking delay 192 ms
I (16075) GATTC_CLIENT: -----
I (17055) GATTC_CLIENT: ESP_GATTS_SERV
I (17065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (17065) GATTC_CLIENT: RSSI: -49 | Blinking delay 196 ms
I (17065) GATTC_CLIENT: -----
I (18055) GATTC_CLIENT: ESP_GATTS_SERV
I (18055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (18055) GATTC_CLIENT: RSSI: -51 | Blinking delay 204 ms
I (18055) GATTC_CLIENT: -----
I (20065) GATTC_CLIENT: ESP_GATTS_SERV
I (20065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (20065) GATTC_CLIENT: RSSI: -50 | Blinking delay 200 ms
I (20065) GATTC_CLIENT: -----
I (21065) GATTC_CLIENT: ESP_GATTS_SERV
I (21065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (21065) GATTC_CLIENT: RSSI: -50 | Blinking delay 200 ms
I (21065) GATTC_CLIENT: -----
I (24075) GATTC_CLIENT: ESP_GATTS_SERV
I (24075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (24075) GATTC_CLIENT: RSSI: -51 | Blinking delay 204 ms
I (24085) GATTC_CLIENT: -----
I (25065) GATTC_CLIENT: ESP_GATTS_SERV
I (25065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (25065) GATTC_CLIENT: RSSI: -47 | Blinking delay 188 ms
I (25065) GATTC_CLIENT: -----
I (26045) GATTC_CLIENT: ESP_GATTS_SERV
I (26055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (26055) GATTC_CLIENT: RSSI: -43 | Blinking delay 172 ms
I (26055) GATTC_CLIENT: -----
I (29065) GATTC_CLIENT: ESP_GATTS_SERV
I (29065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (29065) GATTC_CLIENT: RSSI: -42 | Blinking delay 168 ms
I (29075) GATTC_CLIENT: -----
I (31055) GATTC_CLIENT: ESP_GATTS_SERV
I (31055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (31055) GATTC_CLIENT: RSSI: -27 | Blinking delay 108 ms
I (31055) GATTC_CLIENT: -----
I (32055) GATTC_CLIENT: ESP_GATTS_SERV
```



```
I (32065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (32065) GATTC_CLIENT: RSSI: -23 | Blinking delay 92 ms
I (32065) GATTC_CLIENT: -----
I (33065) GATTC_CLIENT: ESP_GATTS_SERV
I (33065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (33065) GATTC_CLIENT: RSSI: -18 | Blinking delay 72 ms
I (33065) GATTC_CLIENT: -----
I (35075) GATTC_CLIENT: ESP_GATTS_SERV
I (35075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (35075) GATTC_CLIENT: RSSI: -23 | Blinking delay 92 ms
I (35085) GATTC_CLIENT: -----
I (36065) GATTC_CLIENT: ESP_GATTS_SERV
I (36065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (36065) GATTC_CLIENT: RSSI: -18 | Blinking delay 72 ms
I (36075) GATTC_CLIENT: -----
I (38065) GATTC_CLIENT: ESP_GATTS_SERV
I (38065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (38065) GATTC_CLIENT: RSSI: -24 | Blinking delay 96 ms
I (38065) GATTC_CLIENT: -----
I (40055) GATTC_CLIENT: ESP_GATTS_SERV
I (40055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (40055) GATTC_CLIENT: RSSI: -28 | Blinking delay 112 ms
I (40055) GATTC_CLIENT: -----
I (41065) GATTC_CLIENT: ESP_GATTS_SERV
I (41065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (41065) GATTC_CLIENT: RSSI: -34 | Blinking delay 136 ms
I (41065) GATTC_CLIENT: -----
I (42055) GATTC_CLIENT: ESP_GATTS_SERV
I (42055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (42055) GATTC_CLIENT: RSSI: -61 | Blinking delay 244 ms
I (42065) GATTC_CLIENT: -----
I (47075) GATTC_CLIENT: ESP_GATTS_SERV
I (47075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (47075) GATTC_CLIENT: RSSI: -81 | Blinking delay 324 ms
I (47075) GATTC_CLIENT: -----
I (50055) GATTC_CLIENT: ESP_GATTS_SERV
I (50055) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (50055) GATTC_CLIENT: RSSI: -80 | Blinking delay 320 ms
I (50065) GATTC_CLIENT: -----
I (52075) GATTC_CLIENT: ESP_GATTS_SERV
I (52075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (52075) GATTC_CLIENT: RSSI: -79 | Blinking delay 316 ms
I (52075) GATTC_CLIENT: -----
I (55065) GATTC_CLIENT: ESP_GATTS_SERV
I (55065) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (55065) GATTC_CLIENT: RSSI: -80 | Blinking delay 320 ms
I (55065) GATTC_CLIENT: -----
I (59075) GATTC_CLIENT: ESP_GATTS_SERV
I (59075) GATTC_CLIENT: MAC: 8c:aa:b5:b8:bf:f6
I (59075) GATTC_CLIENT: RSSI: -28 | Blinking delay 112 ms
I (59085) GATTC_CLIENT: -----
I (61045) GATTC_CLIENT: Waiting 30 seconds before starting a new scan
```

## Tarea previa

```
static const char remote_device_name[] = "ESP_GATTS_SERV";
```

## Tarea 8

Modifica el cliente para que las notificaciones enviadas por tu servidor GATT modificado en la práctica anterior sean recibidas y monitorizadas por parte del cliente GATT.

Tras cambiar el nombre de `remote_device_name` para que coincida con el nombre del servidor observamos los siguientes pasos en los logs del servidor:

- Conexión establecida
- Creación de la tarea de generación de valores aleatorios
- Confirmación de escritura del valor 0x10 en `IDX_CHAR_CFG_A`
- Envío de valores aleatorios al cliente suscrito a notificaciones de la característica `IDX_CHAR_VAL_A`

Salida Servidor:

```
I (624902) GATTS_TABLE_DEMO: advertising start successfully
I (625082) GATTS_TABLE_DEMO: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (625082) GATTS_TABLE_DEMO: 24 0a c4 ea 28 c2
I (625082) GATTS_TABLE_DEMO: Creating publish_data_task...
I (625092) APP: Update value for IDX_CHAR_VAL_A: 11 E1 33 44
I (625102) APP: Notifications disabled, not sending data...
I (625262) GATTS_TABLE_DEMO: update connection params status = 0, min_int = 16,
max_int = 32,conn_int = 32,latency = 0, timeout = 400
I (626062) GATTS_TABLE_DEMO: ESP_GATTS_MTU_EVT, MTU 500
I (626102) APP: Update value for IDX_CHAR_VAL_A: 11 F6 33 44
I (626102) APP: Notifications disabled, not sending data...
I (626222) GATTS_TABLE_DEMO: GATT_WRITE_EVT, handle = 43, value len = 2, value :
I (626222) GATTS_TABLE_DEMO: 01 00
I (626232) GATTS_TABLE_DEMO: notify enable
I (626232) APP: IDX_CHAR_CFG_A written value: 01 00
I (626242) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (627102) APP: Update value for IDX_CHAR_VAL_A: 11 F4 33 44
I (627102) APP: Notifications enabled, sending data...
I (627102) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (627102) APP: Data sent successfully: 11 F4 33 44
I (628112) APP: Update value for IDX_CHAR_VAL_A: 11 8A 33 44
I (628112) APP: Notifications enabled, sending data...
I (628112) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (628112) APP: Data sent successfully: 11 8A 33 44
I (629122) APP: Update value for IDX_CHAR_VAL_A: 11 C9 33 44
I (629122) APP: Notifications enabled, sending data...
I (629122) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (629122) APP: Data sent successfully: 11 C9 33 44
I (630132) APP: Update value for IDX_CHAR_VAL_A: 11 A4 33 44
I (630132) APP: Notifications enabled, sending data...
```

```
I (630132) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (630132) APP: Data sent successfully: 11 A4 33 44
I (631142) APP: Update value for IDX_CHAR_VAL_A: 11 1E 33 44
I (631142) APP: Notifications enabled, sending data...
I (631142) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (631142) APP: Data sent successfully: 11 1E 33 44
I (632152) APP: Update value for IDX_CHAR_VAL_A: 11 E9 33 44
I (632152) APP: Notifications enabled, sending data...
I (632152) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (632152) APP: Data sent successfully: 11 E9 33 44
I (633162) APP: Update value for IDX_CHAR_VAL_A: 11 96 33 44
I (633162) APP: Notifications enabled, sending data...
I (633162) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (633162) APP: Data sent successfully: 11 96 33 44
I (634172) APP: Update value for IDX_CHAR_VAL_A: 11 29 33 44
I (634172) APP: Notifications enabled, sending data...
I (634172) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (634172) APP: Data sent successfully: 11 29 33 44
I (635182) APP: Update value for IDX_CHAR_VAL_A: 11 6E 33 44
I (635182) APP: Notifications enabled, sending data...
I (635182) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (635182) APP: Data sent successfully: 11 6E 33 44
I (636192) APP: Update value for IDX_CHAR_VAL_A: 11 6F 33 44
I (636192) APP: Notifications enabled, sending data...
I (636192) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (636192) APP: Data sent successfully: 11 6F 33 44
I (637202) APP: Update value for IDX_CHAR_VAL_A: 11 0F 33 44
I (637202) APP: Notifications enabled, sending data...
I (637202) GATTS_TABLE_DEMO: ESP_GATTS_CONF_EVT, status = 0, attr_handle 42
I (637202) APP: Data sent successfully: 11 0F 33 44
```

Por otro lado, en el cliente se ejecutan las siguientes acciones:

- Conexión al servidor
- Configuración de MTU de la conexión
- Descubrimiento de servicios
- Suscripción a notificaciones

Salida Cliente:

```
I (4095) GATTC_CLIENT: searched device ESP_GATTS_SERV

I (4105) GATTC_CLIENT: connect to the remote device.
I (4115) GATTC_CLIENT: stop scan successfully
I (4175) GATTC_CLIENT: ESP_GATTC_CONNECT_EVT conn_id 0, if 3
I (4175) GATTC_CLIENT: REMOTE BDA:
I (4175) GATTC_CLIENT: 8c aa b5 b8 bf f6
I (4185) GATTC_CLIENT: open success
I (4385) GATTC_CLIENT: update connection params status = 0, min_int = 16, max_int
= 32,conn_int = 32,latency = 0, timeout = 400
I (5105) GATTC_CLIENT: discover service complete conn_id 0
```

```
I (5105) GATTC_CLIENT: SEARCH RES: conn_id = 0 is primary service 1
I (5105) GATTC_CLIENT: start handle 40 end handle 65535 current handle value 40
I (5115) GATTC_CLIENT: service found
I (5125) GATTC_CLIENT: UUID16: ff
I (5125) GATTC_CLIENT: Get service information from remote device
I (5135) GATTC_CLIENT: ESP_GATTC_SEARCH_CMPL_EVT
I (5135) GATTC_CLIENT: ESP_GATTC_REG_FOR_NOTIFY_EVT
I (5185) GATTC_CLIENT: ESP_GATTC_CFG_MTU_EVT, Status 0, MTU 500, conn_id 0
I (5345) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (5345) GATTC_CLIENT: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
I (5345) GATTC_CLIENT: write descr success
I (6225) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (6225) GATTC_CLIENT: 11 f4 33 44
I (7225) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (7225) GATTC_CLIENT: 11 8a 33 44
I (8225) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (8225) GATTC_CLIENT: 11 c9 33 44
I (9225) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (9225) GATTC_CLIENT: 11 a4 33 44
I (10265) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (10265) GATTC_CLIENT: 11 1e 33 44
I (11265) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (11265) GATTC_CLIENT: 11 e9 33 44
I (12265) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (12265) GATTC_CLIENT: 11 96 33 44
I (13265) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (13265) GATTC_CLIENT: 11 29 33 44
I (14305) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (14305) GATTC_CLIENT: 11 6e 33 44
I (15305) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (15305) GATTC_CLIENT: 11 6f 33 44
I (16305) GATTC_CLIENT: ESP_GATTC_NOTIFY_EVT, receive notify value:
I (16305) GATTC_CLIENT: 11 0f 33 44
```