

THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY

BÁO CÁO LAB03

NHÓM 5	
Họ và tên	MSSV
Trần Lê Minh Đăng	21520684
Lê Hữu Đạt	21520697
Trần Văn Dương	21520763

Bài tập 1. Dựa trên project ble_compatibility_test, trả lời các câu hỏi sau:

- Trình bày cách cài đặt Advertise Data, Scan Response Data và quá trình quảng bá dữ liệu? Phân tích và cho biết ý nghĩa các phần tử của mảng raw_adv_data[]?
- Nêu cách cài đặt truyền/nhận dữ liệu giữa ESP32 và điện thoại (qua ứng dụng LightBlue® Explorer)?

Trả lời:

a.

- Phân tích và cho biết ý nghĩa các phần tử của mảng raw_adv_data[]

```
static uint8_t raw_adv_data[] = {
    /* flags */
    0x02, 0x01, 0x06,
    /* tx power*/
    0x02, 0x0a, 0xeb,
    /* service uuid */
    0x03, 0x03, 0xFF, 0x00,
    /* device name */
    0x0E, 0x09, 'B', 'L', 'E', '_', 'C', 'O', 'M', 'P', '_', 'T', 'E', 'S', 'T'
};
static uint8_t raw_scan_rsp_data[] = {
    /* flags */
    0x02, 0x01, 0x06,
    /* tx power */
    0x02, 0x0a, 0xeb,
    /* service uuid */
    0x03, 0x03, 0xFF, 0x00
};
```

- Trong mảng `raw_adv_data[]`, mỗi phần tử đại diện cho một trường thông điệp quảng bá trong giao thức BLE (Bluetooth Low Energy).

- Flags (0x02, 0x01, 0x06):
 - 0x02 là độ dài của phần dữ liệu trường flags
 - 0x01 là loại trường (flags).
 - 0x06 là giá trị của flags, đại diện cho General Discoverable Mode và BR/EDR Not Supported.
- Tx Power (0x02, 0x0a, 0xeb):
 - 0x02 là độ dài của phần dữ liệu trường tx power.
 - 0x0a là loại trường (tx power).
 - 0xeb là giá trị công suất truyền (tx power level). Trong trường hợp này, 0xEB thường được sử dụng để biểu thị mức công suất trung bình (-21 dBm).
- Service UUID (0x03, 0x03, 0xFF, 0x00):
 - 0x03 là độ dài của phần dữ liệu trường UUID.
 - 0x03 là loại trường (UUID).
 - 0xFF, 0x00 là UUID của dịch vụ.
- Device Name (0x0E, 0x09, 'B', 'L', 'E', ' ', 'C', 'O', 'M', 'P', ' ', 'T', 'E', 'S', 'T'):
 - 0x0E là độ dài của phần dữ liệu trường tên thiết bị.
 - 0x09 là loại trường (tên thiết bị).
 - Sau đó là chuỗi ký tự ASCII 'BLE_COMP_TEST', là tên của thiết bị. Đây là một tên tùy chỉnh được sử dụng để nhận dạng thiết bị trong quảng bá BLE.

2. Cách cài đặt Advertise Data, Scan Response Data và quá trình quảng bá dữ liệu

- Cấu hình Advertise Data và Scan Response

```
static uint8_t service_uuid[16] = {
    /* LSB <-----> MSB */
    //first uuid, 16bit, [12],[13] is the value
    0xfb, 0x34, 0x9b, 0x5f, 0x80, 0x00, 0x00, 0x80, 0x00, 0x10, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,
};
```

- Đầu tiên tạo mảng service_uuid[16] để định nghĩa một UUID dịch vụ cụ thể trong ứng dụng Bluetooth Low Energy (BLE). Trong đó [12],[13] là giá trị của UUID

```
/* The length of adv data must be less than 31 bytes */
static esp_ble_adv_data_t adv_data = {
    .set_scan_rsp      = false,
    .include_name       = true,
    .include_txpower    = true,
    .min_interval       = 0x20,
    .max_interval       = 0x40,
    .appearance         = 0x00,
    .manufacturer_len   = 0,    //TEST_MANUFACTURER_DATA_LEN,
    .p_manufacturer_data = NULL, //test_manufacturer,
    .service_data_len    = 0,
    .p_service_data     = NULL,
    .service_uuid_len    = sizeof(service_uuid),
    .p_service_uuid     = service_uuid,
    .flag = (ESP_BLE_ADV_FLAG_GEN_DISC | ESP_BLE_ADV_FLAG_BREDR_NOT_SPT),
};
```

- Cấu hình Advertising Data (adv_data):
 - set_scan_rsp: Xác định xem dữ liệu này có phải là dữ liệu phản hồi quét không (false => không).
 - include_name: Xác định xem tên của thiết bị có được bao gồm trong dữ liệu quảng bá không. (true => có)
 - include_txpower: Xác định xem công suất truyền của thiết bị có được bao gồm không. (true => có)
 - min_interval, max_interval: Khoảng thời gian giữa các lần quảng bá (s)
 - appearance: Xác định ngoại hình của thiết bị trong quảng bá (0x00)

- manufacturer_len, p_manufacturer_data: Thông tin nhà sản xuất (trong trường hợp này, không xác định).
- service_data_len, p_service_data: Dữ liệu dịch vụ (trong trường hợp này, không xác định).
- service_uuid_len, p_service_uuid: UUID của dịch vụ (trong trường hợp này, sử dụng service_uuid đã được định nghĩa trước đó).
- flag: Các cờ quảng bá.
 - ESP_BLE_ADV_FLAG_GEN_DISC (quảng bá chung)
 - ESP_BLE_ADV_FLAG_BREDR_NOT_SPT (không hỗ trợ BR/EDR).

```
// scan response data
static esp_ble_adv_data_t scan_rsp_data = {
    .set_scan_rsp      = true,
    .include_name      = true,
    .include_txpower   = true,
    .min_interval      = 0x20,
    .max_interval      = 0x40,
    .appearance        = 0x00,
    .manufacturer_len  = 0, //TEST_MANUFACTURER_DATA_LEN,
    .p_manufacturer_data = NULL, //&test_manufacturer[0],
    .service_data_len   = 0,
    .p_service_data     = NULL,
    .service_uuid_len   = 16,
    .p_service_uuid     = service_uuid,
    .flag = (ESP_BLE_ADV_FLAG_GEN_DISC | ESP_BLE_ADV_FLAG_BREDR_NOT_SPT),
};
```

- Cấu hình Advertising Data (adv_data):
 - set_scan_rsp: Xác định xem dữ liệu này có phải là dữ liệu phản hồi quét không. Ở đây đặt true để xác định rằng dữ liệu này là dữ liệu phản hồi
 - Các trường hợp còn lại tương tự

- Quá trình quảng bá dữ liệu

```

case ESP_GAP_BLE_ADV_DATA_SET_COMPLETE_EVT:
    adv_config_done &= (~ADV_CONFIG_FLAG);
    if (adv_config_done == 0){
        esp_ble_gap_start_advertising(&adv_params);
    }
    break;
case ESP_GAP_BLE_SCAN_RSP_DATA_SET_COMPLETE_EVT:
    adv_config_done &= (~SCAN_RSP_CONFIG_FLAG);
    if (adv_config_done == 0){
        esp_ble_gap_start_advertising(&adv_params);
    }
    break;

```

- Sau khi advertising data và scan response data đã được cài đặt xong, bắt đầu quảng bá với hàm `esp_ble_gap_start_advertising(&adv_params)`. Với `adv_params` là:

```

static esp_ble_adv_params_t adv_params = {
    .adv_int_min      = 0x40,
    .adv_int_max      = 0x40,
    .adv_type          = ADV_TYPE_IND,
    .own_addr_type     = BLE_ADDR_TYPE_PUBLIC,
    .channel_map       = ADV_CHNL_ALL,
    .adv_filter_policy = ADV_FILTER_ALLOW_SCAN_ANY_CON_ANY,
};

```

- `adv_type`:
 - Xác định loại quảng bá được sử dụng. Trong trường hợp này, `ADV_TYPE_IND` được sử dụng, đại diện cho quảng bá không được đồng bộ hóa.
 - Các thiết bị khác trong phạm vi có thể nhận được dữ liệu quảng bá này.
- `own_addr_type`:
 - Xác định loại địa chỉ của thiết bị (own address type) được sử dụng trong quá trình quảng bá.

- Trong đoạn mã, BLE_ADDR_TYPE_PUBLIC được sử dụng, đại diện cho địa chỉ công cộng của thiết bị.
- channel_map:
 - Xác định bản đồ kênh (channel map) được sử dụng trong quá trình quảng bá.
 - Trong đoạn mã, ADV_CHNL_ALL được sử dụng để chỉ định rằng tất cả các kênh BLE sẽ được sử dụng cho quảng bá.
- adv_filter_policy:
 - Xác định chính sách bộ lọc quảng bá (advertising filter policy) được sử dụng.
 - Trong đoạn mã, ADV_FILTER_ALLOW_SCAN_ANY_CON_ANY được sử dụng, cho phép quảng bá được quét bởi bất kỳ thiết bị nào và kết nối bởi bất kỳ thiết bị nào.
- Kiểm tra quảng bá thành công hay thất bại

```
case ESP_GAP_BLE_ADV_START_COMPLETE_EVT:
    /* advertising start complete event to indicate advertising start successfully or failed */
    if (param->adv_start_cmpl.status != ESP_BT_STATUS_SUCCESS) {
        ESP_LOGE(EXAMPLE_TAG, "advertising start failed");
    }else{
        ESP_LOGI(EXAMPLE_TAG, "(0) ***** advertising start successfully ***** \n");
    }
    break;
```

- Khi có thiết bị kết nối thì ứng dụng kết nối tới BLE sẽ nhận được thông báo về một mã Passkey từ thiết bị Bluetooth Low Energy (BLE). Mã Passkey này sẽ được hiển thị cho người dùng để nhập vào.

```
break;
case ESP_GAP_BLE_PASSKEY_NOTIF_EVT: //the app will receive this evt when the IO has Output capability and the peer device has Input capability.
    //show the passkey number to the user to input it in the peer device.
    ESP_LOGI(EXAMPLE_TAG, "The passkey notify number:%d", param->ble_security.key_notif.passkey);
    break;
```

- Quá trình xác thực (authentication) giữa thiết bị BLE và thiết bị đối tác

```

case ESP_GAP_BLE_AUTH_CMPL_EVT: {
    esp_bd_addr_t bd_addr;
    memcpy(bd_addr, param->ble_security.auth_cmpl.bd_addr, sizeof(esp_bd_addr_t));
    EXAMPLE_DEBUG(EXAMPLE_TAG, "remote BD_ADDR: %08x%04x", \
        (bd_addr[0] << 24) + (bd_addr[1] << 16) + (bd_addr[2] << 8) + bd_addr[3], \
        (bd_addr[4] << 8) + bd_addr[5]);
    EXAMPLE_DEBUG(EXAMPLE_TAG, "address type = %d", param->ble_security.auth_cmpl.addr_type);
    if (param->ble_security.auth_cmpl.success){
        ESP_LOGI(EXAMPLE_TAG, "(1) ***** pair status = success ***** \n");
    }
    else {
        ESP_LOGI(EXAMPLE_TAG, "***** pair status = fail, reason = 0x%x *****\n", param->ble_security.auth_cmpl.fail_reason);
    }
    show_bonded_devices();
    break;
}

```

- param->ble_security.auth_cmpl.bd_addr:
 - Đây là địa chỉ Bluetooth (BD_ADDR) của thiết bị BLE đối tác đã hoàn tất quá trình xác thực.
 - Địa chỉ này được sao chép từ thông tin trong sự kiện và được sử dụng để xác định thiết bị đã xác thực.
- param->ble_security.auth_cmpl.addr_type:
 - Loại địa chỉ của thiết bị đối tác (BD_ADDR).
 - Thường được sử dụng để xác định liệu địa chỉ này là địa chỉ công cộng, địa chỉ ngẫu nhiên hay địa chỉ khác.
- param->ble_security.auth_cmpl.success:
 - Biểu thị kết quả của quá trình xác thực.
 - Nếu giá trị này là true, quá trình xác thực đã thành công.
 - Nếu giá trị này là false, quá trình xác thực đã thất bại và lý do thất bại được cung cấp trong trường fail_reason.
- param->ble_security.auth_cmpl.fail_reason:
 - Lý do thất bại của quá trình xác thực, chỉ được cung cấp nếu success là false.
 - Lý do này có thể bao gồm các mã lỗi được định nghĩa trước, giúp xác định nguyên nhân cụ thể của việc xác thực không thành công.

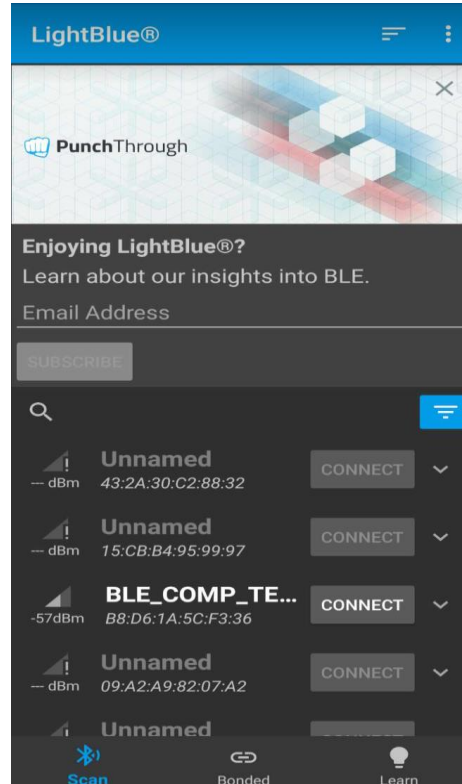
b. Nêu cách cài đặt truyền/nhận dữ liệu giữa ESP32 và điện thoại (qua ứng dụng LightBlue® Explorer)?

- Bắt đầu quảng bá

```
I (515) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (521) heap_init: At 40097F78 len 00008088 (32 KiB): IRAM
I (529) spi_flash: detected chip: generic
I (532) spi_flash: flash io: dio
I (538) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (571) BTDM_INIT: BT controller compile version [0f0c5a2]
I (581) system_api: Base MAC address is not set
I (581) system_api: read default base MAC address from EFUSE
I (581) BTDM_INIT: Bluetooth MAC: b8:d6:1a:5c:f3:36
I (591) phy_init: phy_version 4791,2c4672b,Dec 20 2023,16:06:06
I (1021) BLE_COMP: create attribute table successfully, the number handle = 11

I (1041) BLE_COMP: (0) ***** advertising start successfully *****
```

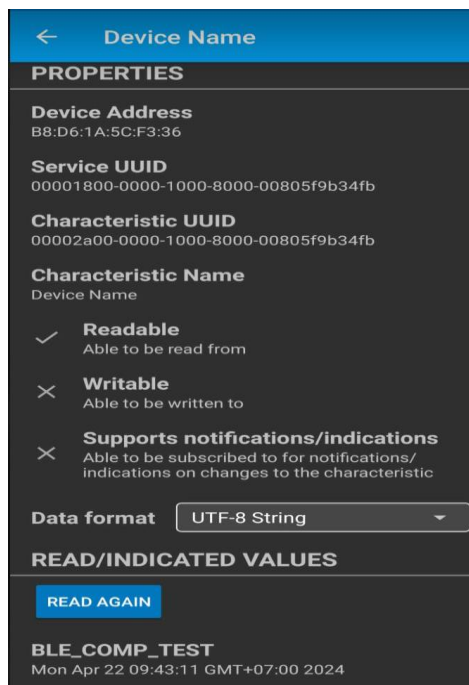
-Sau khi quảng bá thành công, mở ứng dụng LightBlue để kết nối với thiết bị BLE_COMP_TEST



-Giao diện sau khi kết nối. Tại đây ta có thể thấy các mục như advertisement data, device information, ...



-Trong phần GENERIC ATTRIBUTE ta có thể subscribe trong mục Service Changed, đọc các dữ liệu của hệ thống như tên thiết bị,.. ở Device Name, Appearance, Central Address Resolution. Ví dụ ta có thể đọc tên thiết bị trong phần Device Name



- Cài đặt để truyền và nhận dữ liệu giữa ESP32 và điện thoại

```
/* Full Database Description - Used to add attributes into the database */
static const esp_gatts_attr_db_t gatt_db[HRS_IDX_NB] =
{
    // Service Declaration
    [IDX_SVC] =
    {{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&primary_service_uuid, ESP_GATT_PERM_READ,
        sizeof(uint16_t), sizeof(GATTS_SERVICE_UUID_TEST), (uint8_t *)&GATTS_SERVICE_UUID_TEST}},

    /* Characteristic Declaration */
    [IDX_CHAR_A] =
    {{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&character_declaration_uuid, ESP_GATT_PERM_READ,
        CHAR_DECLARATION_SIZE, CHAR_DECLARATION_SIZE, (uint8_t *)&char_prop_read_write}},

    /* Characteristic Value */
    [IDX_CHAR_VAL_A] =
    {{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&CHAR_1_SHORT_WR, ESP_GATT_PERM_READ | ESP_GATT_PERM_WRITE | ESP_GATT_PERM_READ,
        SHORT_CHAR_VAL_LEN, sizeof(char_c), (uint8_t *)&char_c}},

    /* Characteristic User Descriptor */
    [IDX_CHAR_CFG_A] =
    {{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&character_user_description, ESP_GATT_PERM_READ,
        sizeof(char1_name), sizeof(char1_name), (uint8_t *)&char1_name}},
}
```

- Service Declaration (Khái báo Dịch vụ):
 - Attribute này đại diện cho việc khai báo một dịch vụ.
 - IDX_SVC chỉ định vị trí của attribute trong bảng.
 - UUID của dịch vụ được khai báo là GATTS_SERVICE_UUID_TEST.
 - Thuộc tính của attribute này là ESP_GATT_PERM_READ, nghĩa là nó có thể được đọc.
 - Kích thước của dữ liệu được khai báo là sizeof(GATTS_SERVICE_UUID_TEST), tức là kích thước của UUID dịch vụ.
- Characteristic Declaration (Khái báo Đặc điểm):
 - Attribute này đại diện cho việc khai báo một đặc điểm (characteristic).
 - IDX_CHAR_A chỉ định vị trí của attribute trong bảng.
 - UUID của đặc điểm được khai báo là character_declaration_uuid.
 - Thuộc tính của attribute này là ESP_GATT_PERM_READ, nghĩa là nó có thể được đọc.
 - Kích thước của dữ liệu được khai báo là CHAR_DECLARATION_SIZE.

- Characteristic Value (Giá trị Đặc điểm):
 - Attribute này chứa giá trị của đặc điểm.
 - `IDX_CHAR_VAL_A` chỉ định vị trí của attribute trong bảng.
 - UUID của đặc điểm được khai báo là `CHAR_1_SHORT_WR`.
 - Thuộc tính của attribute này là `ESP_GATT_PERM_READ | ESP_GATT_PERM_WRITE | ESP_GATT_PERM_READ_ENC_MITM`, nghĩa là nó có thể được đọc, ghi, và được đọc bằng mã hóa với MITM.
 - Kích thước của dữ liệu được khai báo là `sizeof(char_c)`.
- Characteristic User Descriptor (Miêu tả Người dùng Đặc điểm):
 - Attribute này chứa miêu tả của đặc điểm dành cho người dùng.
 - `IDX_CHAR_CFG_A` chỉ định vị trí của attribute trong bảng.
 - UUID của đặc điểm được khai báo là `character_user_description`.
 - Thuộc tính của attribute này là `ESP_GATT_PERM_READ`, nghĩa là nó có thể được đọc.
 - Kích thước của dữ liệu được khai báo là `sizeof(char1_name)`.

- Ví dụ ta muốn đọc giá trị là “Hello”

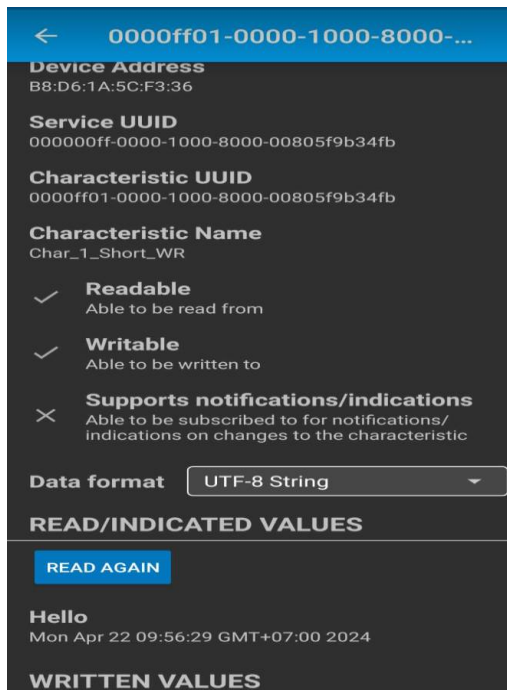
Đầu tiên khai báo mảng `char_c[] = “Hello”`

```
static const uint8_t char_c[] = "Hello";
```

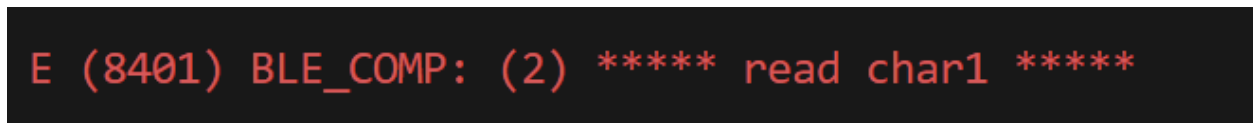
Tiếp đến để mảng này `[IDX_CHAR_VAL_A]`. Kích thước tối đa của dữ liệu là 10 byte được khai báo ở biến `SHORT_CHAR_VAL_LEN`

```
/* Characteristic Value */
[IDX_CHAR_VAL_A] =
{{ESP_GATT_AUTO_RSP}, {ESP_UUID_LEN_16, (uint8_t *)&CHAR_1_SHORT_WR, ESP_GATT_PERM_READ | ESP_GATT_PERM_WRITE | ESP_GATT_PERM_READ_ENC_MITM,
SHORT_CHAR_VAL_LEN, sizeof(char_c), (uint8_t *)char_c}},
```

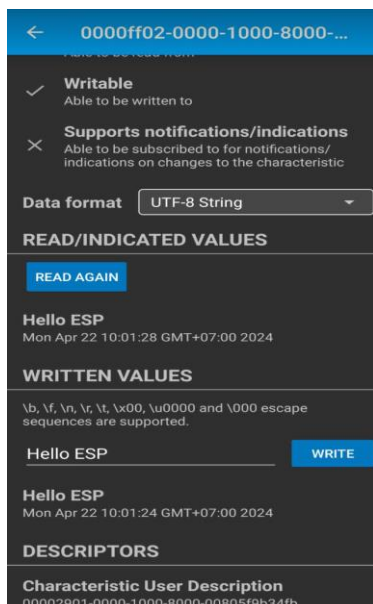
Vào app LightBlue để đọc dữ liệu



Trên monitor sẽ in ra thông báo đọc dữ liệu



Để ghi dữ liệu



Bài tập 2. Dựa trên test case được cung cấp trong file

ble_compatibility_test_case.md:

Hãy viết báo cáo kết quả test (lưu ý thực hiện đủ số test lần yêu cầu).

Trả lời:

- Test for ADV Performance (*)

- Nội dung test

Search Device

Refresh the scanning in LightBlue® Explorer to check if the device to be tested can be found quickly. Please repeat this action 10 times.

Test Results

The test is passed, if you get the following results:

- The device starts advertising and outputs the log `(0) ***** advertising start successfully *****`.
- LightBlue® Explorer scans and successfully discovers `BLE_COMP_TEST` each time.

Kết quả:

```
I (1041) BLE_COMP: (0) ***** advertising start successfully *****
```

The screenshot shows the LightBlue Explorer app interface. On the left, a list of discovered BLE devices is displayed. The selected device, BLE_COMP_TEST, is highlighted. On the right, the properties of the selected device are shown in a table format.

Properties	Values
Tx Power	9
Local name	BLE_COMP_TEST
Adv. packet	0x0201060E09424C455F434F4D505F54455354020A090303FF000512200040000201060E09424C455F434F4D505F54455354020A090303FF00051220004000
Adv. flags	• LE General Discoverable Mode • BR/EDR not supported
Adv. service UUIDs	• 000000ff-0000-1000-8000-00805f9b34fb • 000000ff-0000-1000-8000-00805f9b34fb

- Test for Pairing Performance

Connect Device

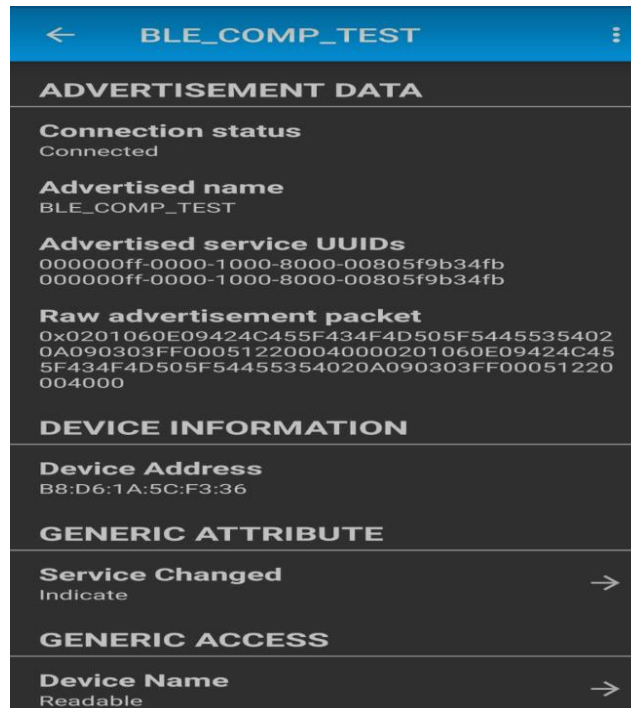
- Open the LightBlue® Explorer scan list and tap on the device name `BLE_COMP_TEST` to establish connection.
- ESP device prints a passkey: `The passkey notify number: 123456`.
- A prompt on the smartphone appears asking if you want to pair. Tap on *Pair*, and then enter the passkey "123456".

Test Results

The test is passed, if you get the following results:

- If the connection is successful:
 - Smartphone shows DATA beginning with `ADVERTISEMENT DATA`
 - ESP device outputs the log: `ESP_GATTS_CONNECT_EVT`
- When the pairing is established, the device shows the following log in green: `(1) ***** pair status = success *****`

Kết quả



```
I (35399) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (35679) BLE_COMP: The passkey notify number:123456
W (44489) BT_SMP: FOR LE SC LTK IS USED INSTEAD OF STK
I (44929) BLE_COMP: (1) ***** pair status = success *****
```

-Test for Service Discovery Performance

Test for Service Discovery Performance

Test Procedures

In LightBlue® Explorer, check the contents of **GATT SERVICES & CHARACTERISTICS**.

Test Results

The test is passed, if you get the following results:

- Service that starts with **000000ff** appears at the bottom of your smartphone.
- This service contains 3 characteristics
 - **Char_1_Short_WR**
 - **Char_2_Long_WR**
 - **Char_3_Short_Notify**

Kết quả:

**000000ff-0000-1000-8000-00805f9
b34fb**

Char_1_Short_WR

Readable, Writable



Char_2_Long_WR

Readable, Writable



Char_3_Short_Notify

Notify



-Test for Read and Encrypt

Test Procedures

Read the value of `Char_1` in LightBlue, and tap on `READ AGAIN`.

Test Results

- Encryption is successful, if your smartphone shows the value "11 22 33 44", and the ESP device prints the log: `(2) ***** read char_1 *****`.
- Encryption fails, if your smartphone shows a blank screen, and the ESP device outputs the error log in red: `GATT_INSUF_AUTHENTICATION: MITM Required`.

Kết quả:

READ/INDICATED VALUES

READ AGAIN

11 22 33 44

Mon Apr 22 10:48:32 GMT+07:00 2024

E (21799) BLE_COMP: (2) ***** read char1 *****

-Test for Short Read and Write

Test for Short Read and Write

Test Procedures

- Navigate to the WRITE interface in LightBlue® Explorer, and write the value "88 99" to `Char_1`.
- Read `Char_1` and check if its value is consistent with the data you have written to it.

Test Results

The test is passed, if you get the following results:

- ESP device prints the log: `(3)***** short write success *****`.
- LightBlue® Explorer shows "88 99" below `READ AGAIN`.

Kết quả:

READ/INDICATED VALUES

READ AGAIN

88 99

Mon Apr 22 10:57:41 GMT+07:00 2024

WRITTEN VALUES

8899

WRITE

88 99

Mon Apr 22 10:57:15 GMT+07:00 2024

I (544659) BLE_COMP: (3)***** short write success *****

E (571119) BLE_COMP: (2) ***** read char1 *****

-Test for Long Read and Write

Test for Long Read and Write

Test Procedures

- Navigate to the WRITE interface in LightBlue® Explorer, and write the string `0x001122...FF001122...FF` of 256 bytes to `Char_2`. The data takes up 16 lines and looks as follows:

```
00112233445566778899AABBCCDDEEFF
00112233445566778899AABBCCDDEEFF
...
00112233445566778899AABBCCDDEEFF
00112233445566778899AABBCCDDEEFF
```

- Read `Char_2` and check if its value is consistent with the data you have written to it.

Test Results

The test is passed, if you get the following results:

- The device prints the log: `ESP_GATTS_EXEC_WRITE_EVT, Length=256` and `(4) ***** long write success *****`.
- LightBlue® Explorer shows `(5) ***** read char_2 *****` below `READ AGAIN`.

Note:

The data to be written can be copied from a text file and pasted into LightBlue® Explorer.

Kết quả:

READ/INDICATED VALUES

READ AGAIN

```
00 11 22 33 44 55 66 77 88 99 AA BB
CC DD EE FF 00 11 22 33 44 55 66 77 88
99 AA BB CC DD EE FF 00 11 22 33 44
55 66 77 88 99 AA BB CC DD EE FF 00
11 22 33 44 55 66 77 88 99 AA BB CC
DD EE FF 00 11 22 33 44 55 66 77 88 99
AA BB CC DD EE FF 00 11 22 33 44 55
66 77 88 99 AA BB CC DD EE FF 00 11
22 33 44 55 66 77 88 99 AA BB CC DD
EE FF 00 11 22 33 44 55 66 77 88 99 AA
BB CC DD EE FF 00 11 22 33 44 55 66
77 88 99 AA BB CC DD EE FF 00 11 22
33 44 55 66 77 88 99 AA BB CC DD EE
FF 00 11 22 33 44 55 66 77 88 99 AA BB
CC DD EE FF 00 11 22 33 44 55 66 77 88
99 AA BB CC DD EE FF 00 11 22 33 44
55 66 77 88 99 AA BB CC DD EE FF 00
11 22 33 44 55 66 77 88 99 AA BB CC
DD EE FF 00 11 22 33 44 55 66 77 88 99
AA BB CC DD EE FF 00 11 22 33 44 55
66 77 88 99 AA BB CC DD EE FF
Mon Apr 22 11:05:43 GMT+07:00 2024
```

WRITTEN VALUES

45566778899AABBCCDDEEFF

WRITE

```
I (1109809) BLE_COMP: ESP_GATTS_EXEC_WRITE_EVT, Length=256
I (1109809) BLE_COMP: (4) ***** long write success *****

E (1111969) BLE_COMP: (5) ***** read char2 *****
E (1112059) BLE_COMP: (5) ***** read char2 *****
E (1112149) BLE_COMP: (5) ***** read char2 *****
E (1112239) BLE_COMP: (5) ***** read char2 *****
E (1112329) BLE_COMP: (5) ***** read char2 *****
E (1112419) BLE_COMP: (5) ***** read char2 *****
E (1112509) BLE_COMP: (5) ***** read char2 *****
E (1112599) BLE_COMP: (5) ***** read char2 *****
E (1112689) BLE_COMP: (5) ***** read char2 *****
E (1112779) BLE_COMP: (5) ***** read char2 *****
E (1112869) BLE_COMP: (5) ***** read char2 *****
E (1112959) BLE_COMP: (5) ***** read char2 *****
```

-Test for Short Notify

Test Procedures

- Enter `Char_3` and tap on `SUBSCRIBE` to enable its Notify function.
- Your phone automatically receives Notify data from the device.

Test Results

The test is passed, if you get the following results:

- ESP device prints the log: `(6) ***** send notify AA BB *****`.
- "AA BB" appears on your smartphone.

Kết quả:

Data format Hex

READ/INDICATED VALUES

READ AGAIN

UNSUBSCRIBE

AA BB
Mon Apr 22 11:09:25 GMT+07:00 2024

DESCRIPTORS

Characteristic User Description
00002901-0000-1000-8000-00805f9b34fb

Successfully subscribed to characteristic changes

I (1274549) BLE_COMP: (6) ***** send notify AA BB *****

-Test for Connection Success Rate (*)

Test for Connection Success Rate (*)

Test procedures

- Break the connection
- Re-establish the connection
- Repeat 10 times

Test Results

The test is passed, if you get the following results:

- Your phone establishes the connection successfully, and the ESP device outputs the log: `(1) ***** pair status = success *****`.
- Your phone breaks the connection, and the device outputs the log: `ESP_GATTS_DISCONNECT_EVT`.
- Connection can be set up each time with no issues.

Kết quả:

```
W (1399289) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1399289) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1399309) BLE_COMP: (0) ***** advertising start successfully *****

I (1402849) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1403209) BLE_COMP: (1) ***** pair status = success *****

W (1408019) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1408019) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1408039) BLE_COMP: (0) ***** advertising start successfully *****

I (1408249) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1408609) BLE_COMP: (1) ***** pair status = success *****

W (1413329) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1413329) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1413349) BLE_COMP: (0) ***** advertising start successfully *****

I (1417479) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1417879) BLE_COMP: (1) ***** pair status = success *****

W (1422509) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1422509) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1422529) BLE_COMP: (0) ***** advertising start successfully *****

I (1424719) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1425119) BLE_COMP: (1) ***** pair status = success *****

W (1428579) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1428589) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1428599) BLE_COMP: (0) ***** advertising start successfully *****

I (1428999) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1429399) BLE_COMP: (1) ***** pair status = success *****

W (1433709) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1433719) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1433729) BLE_COMP: (0) ***** advertising start successfully *****

I (1437679) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1438089) BLE_COMP: (1) ***** pair status = success *****

W (1442579) BT_HCI: hcif disc complete: hdl 0x0, rsn 0x13
I (1442579) BLE_COMP: ESP_GATTS_DISCONNECT_EVT, reason = 19
I (1442599) BLE_COMP: (0) ***** advertising start successfully *****

I (1444389) BLE_COMP: ESP_GATTS_CONNECT_EVT, conn_id = 0
I (1444789) BLE_COMP: (1) ***** pair status = success *****
```

-Test for Long Connection Stability

The connection must be stable throughout the tests.

Note:

If the existing connection breaks:

- LightBlue® Explorer prints `Disconnected`.
- ESP device outputs the log: `ESP_GATTS_DISCONNECT_EVT, reason = (0) ***** advertising start successfully *****`.

Kết quả: Kết nối ổn định

Bài tập 3. Một nhà hàng muốn gửi khuyến mãi giảm giá 50% đến tất cả các khách hàng đứng trước cửa nhà hàng:

Sử dụng ESP32 để hiện thực mô hình BLE Broadcasting gửi khuyến mãi cho tất cả các điện thoại ở gần. Test bằng cách dùng nhiều điện thoại và quan sát trường dữ liệu tương ứng trên ứng dụng LightBlue.

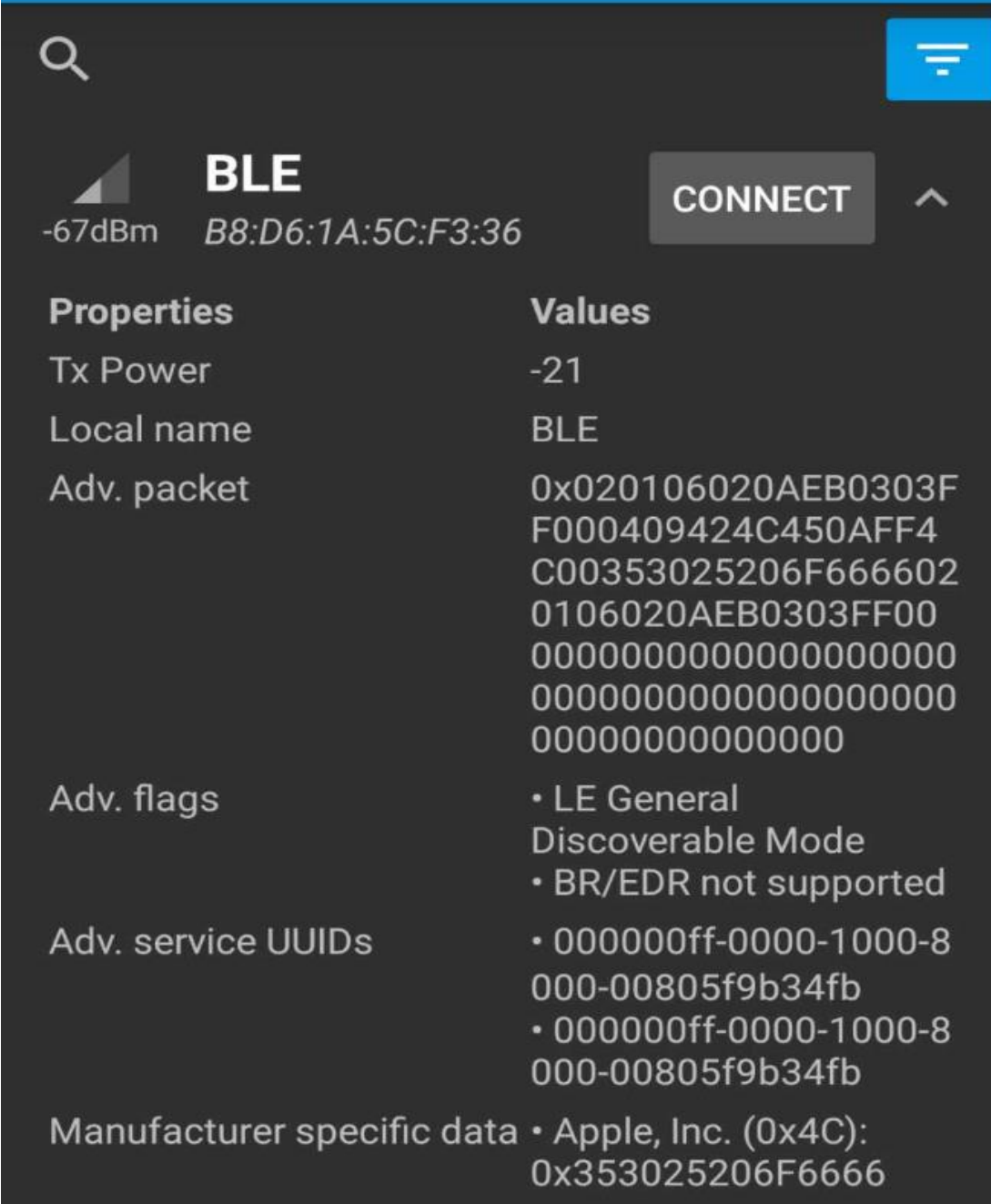
Trả lời:

```
#define CONFIG_SET_RAW_ADV_DATA
#ifdef CONFIG_SET_RAW_ADV_DATA
static uint8_t raw_adv_data[] = {
    /* flags */
    0x02, 0x01, 0x06,
    /* tx power*/
    0x02, 0x0a, 0xeb,
    /* service uuid */
    0x03, 0x03, 0xFF, 0x00,
    /* device name */
    0x04, 0x09, 'B', 'L', 'E',
    /* manufacturer */
    0x0A, 0xFF, 0x4C, 0x00, '5', '0', '%', ' ', 'o', 'f', 'f'
};
```

Thêm trường manufacturer với

- 0x0A là độ dài

- 0xFF là ad type: Manufacturer Specific Data
- 0x4c, 0x00 (0x004c) là giá trị Company Identifiers ,ở đây là Apple Inc
- Phần sau là dữ liệu muốn hiển thị



BLE **CONNECT**

-67dBm B8:D6:1A:5C:F3:36

Properties	Values
Tx Power	-21
Local name	BLE
Adv. packet	0x020106020AEB0303F F000409424C450AFF4 C00353025206F666602 0106020AEB0303FF00 00000000000000000000 00000000000000000000 0000000000000000
Adv. flags	• LE General Discoverable Mode • BR/EDR not supported
Adv. service UUIDs	• 000000ff-0000-1000-8 000-00805f9b34fb • 000000ff-0000-1000-8 000-00805f9b34fb
Manufacturer specific data	• Apple, Inc. (0x4C): 0x353025206F6666

Dữ liệu được hiển thị dưới dạng mã hex trong trường manufacturer specific data.

Bài tập 4. Viết một chương trình thực hiện các công việc sau:

- Quảng bá ESP32 với tên thiết bị là tên nhóm.
- Thiết lập kết nối ESP32 và điện thoại.
- Dùng ứng dụng LightBlue để đọc và gửi dữ liệu đến ESP32 theo thứ tự:

```
> ESP32: Please input ID of each team member:  
> Mobile: <ID1>  
> Mobile: <ID2>  
> Mobile: <ID3>  
...
```

- Mỗi khi điện thoại gửi dữ liệu ID (mã số sinh viên) thành công, ESP32 hiển thị ID đó lên LCD.

Trả lời:

- Sửa tên thiết bị thành tên nhóm

```
#define CONFIG_SET_RAW_ADV_DATA  
#ifdef CONFIG_SET_RAW_ADV_DATA  
static uint8_t raw_adv_data[] = {  
    /* flags */  
    0x02, 0x01, 0x06,  
    /* tx power*/  
    0x02, 0x0a, 0xeb,  
    /* service uuid */  
    0x03, 0x03, 0xFF, 0x00,  
    /* device name */  
    0x04, 0x09, 'N', 'H', 'O', 'M', ' ', '5',  
    // /* manufacturer */  
    // 0x0A, 0xFF, 0x4C, 0x00, '5', '0', '%', ' ', 'o', 'f', 'f'  
};
```

-

```

}
if(gatt_db_handle_table[IDX_CHAR_VAL_B] == param->write.handle) {
    uint8_t write_data[] = "21520684";
    uint8_t write_data2[] = "21520697";
    uint8_t write_data3[] = "21520763";
    if(memcmp(write_data, param->write.value, param->write.len) == 0){
        task_ssd1306_display_text("21520684");
    }
    else if(memcmp(write_data2, param->write.value, param->write.len) == 0){
        task_ssd1306_display_text("\n21520697");
    }
    else if(memcmp(write_data3, param->write.value, param->write.len) == 0){
        task_ssd1306_display_text("\n\n21520763\n");
    }
    ESP_LOGI(EXAMPLE_TAG, "(3)***** ID write success ***** \n");
}
}

```

Video Demo:[Demo](#)