

BÀI THỰC HÀNH 3: LẬP TRÌNH BLUETOOTH LOW ENERGY TRÊN KIT ESP32

I. Mục tiêu

- Hiểu được chuẩn giao tiếp Bluetooth Low Energy (BLE).
- Thiết lập mạng giao tiếp giữa các thiết bị thông qua chuẩn giao tiếp BLE.

II. Chuẩn bị trước

- Tìm hiểu về chuẩn giao tiếp BLE.

III. Nội dung thực hành

1. Lập trình BLE Observer

2. Lập trình BLE Broadcaster

3. Lập trình giao tiếp BLE Node và BLE Gateway

IV. Hướng dẫn thực hành

1. Giới thiệu về chuẩn giao tiếp BLE

Bluetooth Low Energy (viết tắt là BLE, còn được gọi là Bluetooth Smart) là được giới thiệu lần đầu ở phiên bản Bluetooth 4.0. BLE rất hấp dẫn với người dùng vì nó tiết kiệm năng lượng hơn, tối ưu hóa về chi phí, bằng thông so với phiên bản Bluetooth cổ điển, nhưng thực tế, BLE hoàn toàn là một kiểu giao tiếp khác biệt được thiết kế với các mục tiêu khác.

BLE là công nghệ để thiết lập các mạng giao tiếp không dây cá nhân (Wireless personal area networks – WPAN). Trong mạng BLE, các thiết bị được chia thành 2 nhóm theo chức năng: thiết bị trung tâm (central/gateway) và thiết bị ngoại vi (peripheral/end node). Một thiết bị trung tâm có thể kết nối được với nhiều thiết bị ngoại vi cùng lúc nhưng các thiết bị ngoại vi thì không thể.



Hình 1 Mạng BLE [1]

Mỗi thiết bị BLE ngoại vi sẽ báo cáo sự khả dụng/sự tồn tại của thiết bị bằng cách gửi các gói tin quảng cáo (quảng bá). Mỗi gói tin quảng cáo có thể chứa tới 31 bytes dữ liệu. Tần số gửi

gói tin quảng cáo có thể được thiết lập bởi từng thiết bị, khi giảm tần số gửi gói tin sẽ giúp tiết kiệm được năng lượng tiêu thụ.

Nếu một thiết bị BLE trung tâm nhận được một gói tin quảng cáo, muốn nhận thêm thông tin từ thiết bị gửi, nó có thể gửi một gói tin request đến thiết bị gửi. Sau đó, thiết bị gửi sẽ gửi lại một gói tin response với nhiều thông tin hơn.

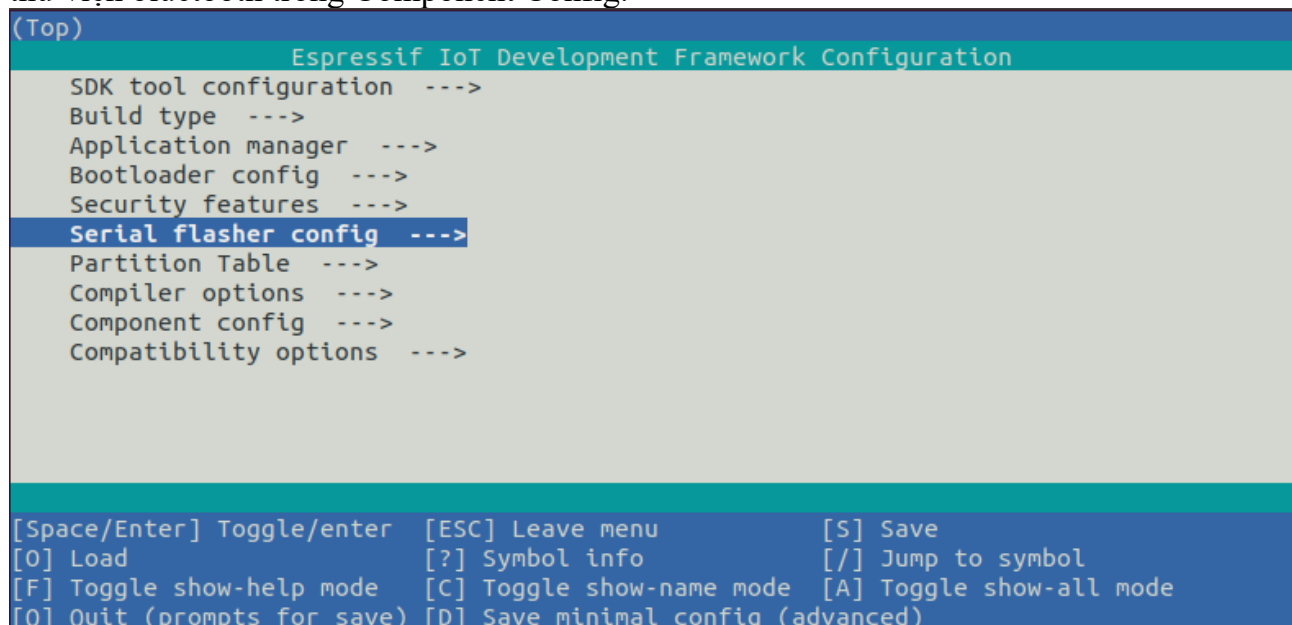


Hình 2 Quá trình gửi gói tin từ BLE peripheral tới BLE central [1]

Các gói tin trên được gửi theo đặc tả của BLE Generic Access Profile (GAP).

2. Lập trình BLE Scanner

☞ Để lập trình BLE cho ESP32, chạy chương trình menuconfig để thiết lập cho phép sử dụng thư viện bluetooth trong Component Config.



Hình 3 Thiết lập menuconfig [2]

☞ Các thư viện cần thiết để lập trình BLE Scanner:

```
#include "esp_bt.h"
#include "esp_bt_main.h"
#include "esp_gap_ble_api.h"
```

☞ Khởi tạo phân vùng NVS để sử dụng cho bluetooth driver.

```
ESP_ERROR_CHECK(nvs_flash_init());
```

☞ Để sử dụng BLE, chúng ta xóa bỏ vùng nhớ dùng để quản lý bluetooth cổ điển:

```
esp_bt_controller_mem_release(ESP_BT_MODE_CLASSIC_BT);
```

☞ Thiết lập chế độ cho BLE:

```
esp_bt_controller_config_t bt_cfg = BT_CONTROLLER_INIT_CONFIG_DEFAULT();
```

```
esp_bt_controller_init(&bt_cfg);
```

```
esp_bt_controller_enable(ESP_BT_MODE_BLE);
```

☞ Esp32 sử dụng framework esp-idf với Bluedroid để thiết lập bluetooth protocol stack, khởi tạo bluedroid và cho phép bluedroid hoạt động để sử dụng BLE:

```
esp_bluedroid_init();
```

```
esp_bluedroid_enable();
```

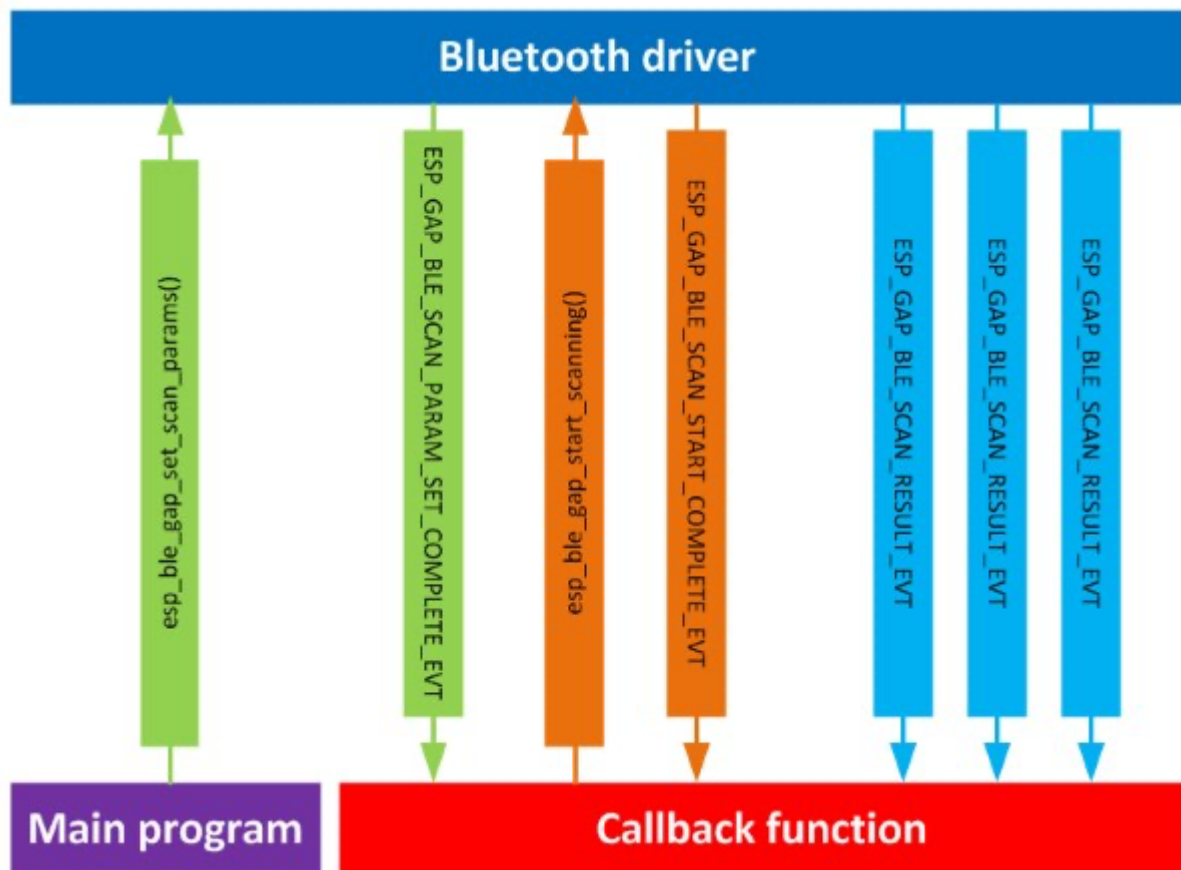
Xử lý GAP và các sự kiện:

☞ Bluetooth driver sử dụng callback function và các sự kiện để giao tiếp với hàm callback có prototype như sau:

```
static void esp_gap_cb(esp_gap_ble_cb_event_t event, esp_ble_gap_cb_param_t *param);
```

☞ Khai báo callback để sử dụng cho BLE GAP:

```
esp_ble_gap_register_callback(esp_gap_cb);
```



Hình 4 Quy trình các sự kiện khi scan BLE [1]

☞ Cài đặt thông số cho BLE:

```
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       = 0x50,  
    .scan_window         = 0x30  
};
```

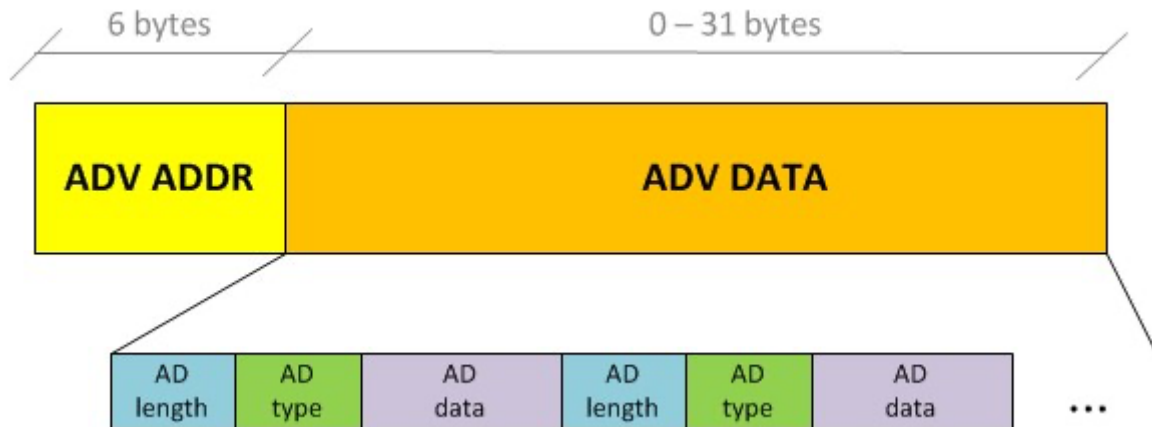
Ngoài ra, ta có thể sử dụng hàm `esp_ble_gap_set_scan_params(&ble_scan_params)` để cập nhật các thông số mới.

☞ **Hiện thị thông tin của các thiết bị đã scan được:**

```
case ESP_GAP_BLE_SCAN_RESULT_EVT:
    if(param->scan_rst.search_evt == ESP_GAP_SEARCH_INQ_RES_EVT) {
        printf("Device found: ADDR=");
        for(int i = 0; i < ESP_BD_ADDR_LEN; i++) {
            printf("%02X", param->scan_rst.bda[i]);
            if(i != ESP_BD_ADDR_LEN -1) printf(":");
        }
    }
```

3. Lập trình BLE Broadcaster

Phần trên chúng ta đã tìm hiểu về BLE scanner và đã nhận được các gói tin quảng cáo từ các thiết bị ngoại vi peripheral. Định dạng của gói tin quảng cáo như thể hiện trong *Hình 5*.



Hình 5 Định dạng gói tin quảng cáo [1]

Trong đó, ADV ADDR là địa chỉ MAC của thiết bị. ADV DATA là trường dữ liệu với tối đa 31 bytes. Một gói tin quảng cáo có thể có từ 1 cấu trúc với 3 thành phần sau:

- AD length: kích thước (số byte) của mỗi cấu trúc.
- AD type: loại dữ liệu chứa trong cấu trúc.
- AD data: dữ liệu thực tế.

Ví dụ: Nếu chúng ta truyền một cái tên local, ta sẽ sử dụng AD type 0x09 theo đặc tả của bluetooth.

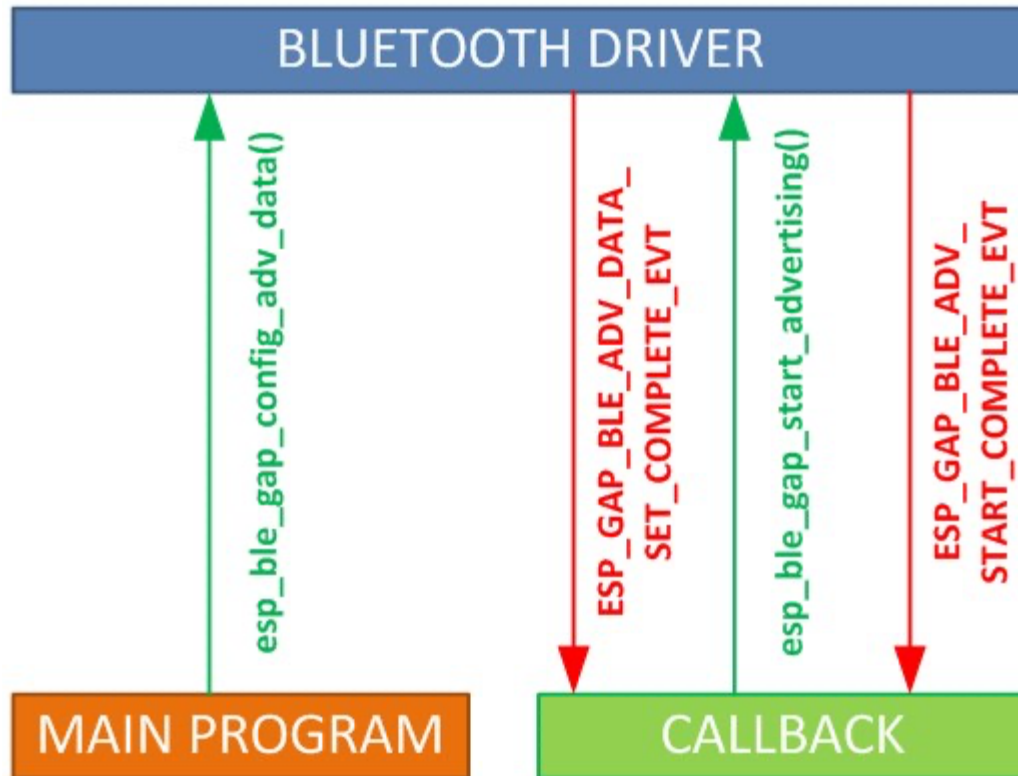
Data Type Value ^	Data Type Name ^	Reference for Definition ^
0x06	«Incomplete List of 128-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS and 4.0)Vol. 3, Part C, sections 11.1.1 and 18.2 (v4.0)Core Specification Supplement, Part A, section 1.1
0x07	«Complete List of 128-bit Service Class UUIDs»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.1 (v2.1 + EDR, 3.0 + HS and 4.0)Vol. 3, Part C, sections 11.1.1 and 18.2 (v4.0)Core Specification Supplement, Part A, section 1.1
0x08	«Shortened Local Name»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.2 (v2.1 + EDR, 3.0 + HS and 4.0)Vol. 3, Part C, sections 11.1.2 and 18.4 (v4.0)Core Specification Supplement, Part A, section 1.2
0x09	«Complete Local Name»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.2 (v2.1 + EDR, 3.0 + HS and 4.0)Vol. 3, Part C, sections 11.1.2 and 18.4 (v4.0)Core Specification Supplement, Part A, section 1.2
0x0A	«Tx Power Level»	Bluetooth Core Specification:Vol. 3, Part C, section 8.1.5 (v2.1 + EDR, 3.0 + HS and 4.0)Vol. 3, Part C, sections 11.1.5 and 18.3 (v4.0)Core Specification Supplement, Part A, section 1.5

Hình 6 Một số mẫu AD type [3]

Quy trình gửi gói tin quảng cáo:

- Chương trình cài đặt dữ liệu để gửi bằng hàm **esp_ble_gap_config_adv_data()**

- Khi cài đặt được hoàn thiện, driver sẽ gửi một sự kiện **ESP_GAP_BLE_ADV_DATA_SET_COMPLETE_EVT**
- Chương trình bắt đầu quá trình gửi gói tin quảng cáo bằng hàm **esp_ble_gap_start_advertising()**
- Khi quá trình gửi gói tin quảng cáo đã bắt đầu, driver sẽ gửi một sự kiện **ESP_GAP_BLE_ADV_START_COMPLETE_EVT**



Hình 7 Quy trình gửi gói tin quảng cáo [1]

☞ Cài đặt thư viện để xử lý các thông số của gói tin quảng cáo:

```
#include "esp_gap_ble_api.h"
```

☞ Dữ liệu quảng cáo có thể được cài đặt thông qua hàm sau:

```
esp_err_t esp_ble_gap_config_adv_data(esp_ble_adv_data_t *adv_data);
```

☞ Cấu trúc dữ liệu của gói tin quảng cáo:


```

/// Advertising data content, according to "Supplement to the Bluetooth Core Specification"
typedef struct {
    bool        set_scan_rsp;           /*!< Set this advertising data as scan re
    bool        include_name;           /*!< Advertising data include device nam
    bool        include_txpower;        /*!< Advertising data include TX power *,
    int         min_interval;           /*!< Advertising data show advertising m:
    int         max_interval;           /*!< Advertising data show advertising m:
    int         appearance;            /*!< External appearance of device */
    uint16_t    manufacturer_len;       /*!< Manufacturer data length */
    uint8_t     *p_manufacturer_data;   /*!< Manufacturer data point */
    uint16_t    service_data_len;       /*!< Service data length */
    uint8_t     *p_service_data;        /*!< Service data point */
    uint16_t    service_uuid_len;       /*!< Service uuid length */
    uint8_t     *p_service_uuid;        /*!< Service uuid array point */
    uint8_t     flag;                  /*!< Advertising flag of discovery mode,
} esp_ble_adv_data_t;

```

Hình 8 Cấu trúc dữ liệu gói tin quảng cáo [2]

☞ Gửi tên thiết bị bằng hàm `esp_ble_gap_set_device_name()`:

```

static esp_ble_adv_data_t adv_data = {
    .include_name = true,
};
[...]
ESP_ERROR_CHECK(esp_ble_gap_set_device_name("ESP32_BLE"));
ESP_ERROR_CHECK(esp_ble_gap_config_adv_data(&adv_data));

```

☞ Thiết lập các thông số để quảng cáo và bắt đầu gửi gói tin quảng cáo bằng hàm:

`esp_err_t esp_ble_gap_start_advertising(esp_ble_adv_params_t *adv_params);`

```

/// Advertising parameters
typedef struct {
    uint16_t    adv_int_min;           /*!< Minimum advertising interval for
                                         undirected and low duty cycle directed adve
                                         Range: 0x0020 to 0x4000 Default: N = 0x0800
                                         Time = N * 0.625 msec Time Range: 20 ms to

    uint16_t    adv_int_max;           /*!< Maximum advertising interval for
                                         undirected and low duty cycle directed adve
                                         Range: 0x0020 to 0x4000 Default: N = 0x0800
                                         Time = N * 0.625 msec Time Range: 20 ms to

    esp_ble_adv_type_t    adv_type;     /*!< Advertising type */
    esp_ble_addr_type_t    own_addr_type; /*!< Owner bluetooth device address type */
    esp_bd_addr_t    peer_addr;         /*!< Peer device bluetooth device address */
    esp_ble_addr_type_t    peer_addr_type; /*!< Peer device bluetooth device address typ
    esp_ble_adv_channel_t    channel_map; /*!< Advertising channel map */
    esp_ble_adv_filter_t    adv_filter_policy; /*!< Advertising filter policy */
} esp_ble_adv_params_t;

```

Hình 9 Các thông số trong gói tin quảng cáo [2]

V. Tài liệu tham khảo

- [1] L. Dentella, 2020. [Online]. Available: <http://www.lucadentella.it/en/category/esp32/>. [Accessed 2021].
- [2] "ESP-IDF Programming Guide," [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/bluetooth/bt_le.html. [Accessed 1 2021].
- [3] "Bluetooth," [Online]. Available: <https://www.bluetooth.com/>. [Accessed 1 2021].
- [4] M. S. SWARTZ, "Learn ESP32," [Online]. Available: https://www.learnesp32.com/20_bluetooth---intro. [Accessed 1 2021].
- [5] "ESP32.VN," [Online]. Available: <https://esp32.vn/>. [Accessed 1 2021].
- [6] C. C. A. & R. D. Kevin Townsend, "Getting Started with Bluetooth Low Energy," O'Reilly Media, 2014, p. 164.