Bluetooth Pairing

Version: 20231108

Online Version

Contents

1 Overview

1.1 Description

For Wi-Fi and Bluetooth combo devices, the pairing data can be sent to the device over Bluetooth.

After entering pairing mode, the device sends Bluetooth advertising packets and waits to be discovered by the mobile app. The mobile app can initiate a Bluetooth connection to the device and send pairing data to it.

1.2 Pairing process

```
sequenceDiagram
    note over Device: Device enters pairing mode
    Device ->> Device: Device regularly <br>sends advertising packets
    note over Client: User opens mobile app
    Client ->> Device: Scan for advertising packets
    Device -->> Client: Respond to scanning
    Client ->> Client: Parse advertising packet
    note over Client: User taps Connect on mobile app
    Client ->> Device: Initiate Bluetooth connection
    Device-->> Client: Connected
    Client ->> Device: Exchange key
    Device-->> Client: Return key
    Client ->> Device: Send encrypted SSID/Password/Token
    Device-->> Client: Return decryption success
    Device ->> Router: Connect to router
    Router-->> Device: Connected
    note over  Device: Connect to cloud for <br>activation and binding
```

1.3 Initialization process

```
  graph LR
    A[Bluetooth connection <br>initialization] --> B[Bluetooth service
<br>initialization] --> C[Bluetooth stack <br>initialization] --> D[
Turn on Bluetooth <br>advertising] --> E[Bluetooth data <br>channel
initialization]
```

2 Development guide

2.1 Runtime environment

Be sure `tuya_iot_config.h` is configured in the header file and enable the Bluetooth service and Bluetooth pairing. The code below shows the macros that should be enabled.

```c
/* Enable/disable Bluetooth service */
#define ENABLE_BT_SERVICE 1
/* Enable/disable Bluetooth pairing */
#define ENABLE_BT_NETCFG 1
```

2.2 Associated components

- svc_bt

2.3 How to

During initialization, Bluetooth pairing is initialized automatically.

3 API description

3.1 Set Bluetooth device name

This API is used to change the Bluetooth device name. In Tuya’ s Bluetooth protocol v3.x, the default Bluetooth device name is `TY`. In v4.x, the default name is `TUYA__`. The macro `TUYA_BLE_VER` in `tuya_iot_config.h` specifies the protocol version you use.

The limit on the length of the Bluetooth device name is two characters in v3.x and five characters in v4.x.
This API should be called before device initialization.

```
/**
 * @brief Set name for Bluetooth device.
 *
 * @param[in] dev_name: device name
 *
 * @note This API is used for setting the name of Bluetooth device.
This API should be called before Bluetooth is initialized.
 *
 * @return OPRT_OK on success. For others on error, please refer to
tuya_error_code.h
 */
OPERATE_RET tuya_set_bt_device_name(CHAR_T* dev_name);
```

3.2 Enable/disable the Bluetooth service

This API is used to enable or disable the Bluetooth service. If you disable this service, the Bluetooth stack is turned off and Bluetooth is unavailable. By default, the Bluetooth service is enabled.

> This API is designed with reliability in mind. Repeated enable or disable actions are invalid.

```
/**
 * @brief Set enable switch for Bluetooth service
 *
 * @param[in] switch: enable switch: TRUE-open, FALSE-close
 *
 * @note This API is used for enabling Bluetooth service. It should
be called before SDK initialization.
 *
 * @return VOID
 */
VOID_T tuya_ble_set_serv_switch(BOOL_T swith);
```

3.3 Get the status of Bluetooth service

This API is used to check if the Bluetooth service is enabled.

```
/**
 * @brief Get status of Bluetooth service
 *
 * @param VOID
 *
 * @note This API is used to get status of the Bluetooth service.
 *
 * @return status of the Bluetooth service
 */
TUYA_BT_SERV_STAT tuya_ble_get_serv_stat(VOID_T);
```

3.4 Restart Bluetooth advertising

This API is used to restart Bluetooth advertising. When advertising parameters are updated, this API is called to update the advertising packets. For common applications, there is no need to take care of this API.

```
/**
 * @brief Resend the Bluetooth adv data to hal and start Bluetooth a
dv
 *
 * @param
 *
 * @note
 *
 * @return VOID
 */
VOID_T tuya_ble_restart_adv(VOID_T);
```

3.5 Set Bluetooth startup property

You need to call this API before you call the system service initialization API.

This API is used to set the Bluetooth startup property. If you do not call this API, all Bluetooth capabilities are started by default when the device boots.

- If you set TUYA_BLE_ABILITY_NONE, the Bluetooth service is not started when the device boots.
- If you set TUYA_BLE_ABILITY_NETCFG, only Bluetooth pairing is enabled when the device boots. The Bluetooth service will be disabled after the device is paired.

```
typedef BYTE_T TUYA_BLE_ABILITY_T;
#define TUYA_BLE_ABILITY_NONE       0x0
#define TUYA_BLE_ABILITY_NETCFG     0x1
#define TUYA_BLE_ABILITY_ALL        0xFF
/**
  * @brief Set abilitiy attribute of Bluetooth
  *
  * @param attr
  *
  * @note  This API is used to set abilitiy attribute of Bluetooth.
  *
  * @return VOID
*/
VOID_T tuya_ble_set_startup_attr(TUYA_BLE_ABILITY_T attr);
```

4 FAQs

4.1 To reduce power usage, can I disable the Bluetooth service after the device is paired over Bluetooth?

Yes, you can.

There are two options to achieve this:

- Option 1: Before the framework is initialized, set `tuya_ble_set_startup_attr` to only enable Bluetooth pairing.
- Option 2: After the device is paired, call `tuya_ble_set_serv_switch` and set it to `FALSE` to disable the Bluetooth service.

4.2 Why can' t devices be discovered during pairing?

Check if the device sends advertising packets using the nRF Sniffer. If the device fails to send advertising packets, it cannot be discovered by the **Smart Life** app.

If the device can send advertising packets, check if it has been paired. If so, reset the device and pair it again.