



蓝牙 + X

文档版本: 20231212

[查看在线版本](#)

目录

1	概述	2
2	数据结构	3
2.1	tuya_ble_ext_module_active_data_t	3
3	API 说明	4
3.1	宏配置	4
3.2	上报扩展模组信息接口	4
3.3	查询扩展模组设备信息回调事件	5
3.4	接收到扩展模组激活信息回调事件	8
4	使用方法	10
4.1	消息序列图	10
4.2	回调事件处理	10
4.3	代码文件	11
5	注意事项	12
6	联系涂鸦	13

本文介绍蓝牙 + X 能力。

1 概述

涂鸦的蓝牙设备一般为单蓝牙设备，即设备通过蓝牙配网完成后，仅可以通过蓝牙链路与云端进行数据交互。

但是当设备具有蓝牙 + X 能力后（其中的 X 为扩展模组，例如 Wi-Fi 模组、Cat.1 模组或 NB-IoT 模组等），在 X 模组未接入时，设备仍然是一个单蓝牙设备，通过蓝牙链路进行配网与控制。如果插入 X 模组，设备可通过蓝牙来激活 X 模组之后，设备就具有双链路通信能力，既可以通过蓝牙链路，又可以通过 X 链路与云端进行数据交互。

2 数据结构

2.1 tuyu_ble_ext_module_active_data_t

```
1 typedef struct{
2     uint8_t *p_data;    /**< ext module active data. */
3     uint16_t data_len;  /**< active data length. */
4 } tuyu_ble_ext_module_active_data_t;
```

- `p_data`: 扩展模组激活信息数据指针。
- `data_len`: 扩展模组激活信息数据长度。

3 API 说明

3.1 宏配置

宏定义	<code>TUYA_BLE_FEATURE_EXT_MODULE_ENABLE</code>
依赖项	无
描述	使能蓝牙 + X 功能： 禁用蓝牙 + X 功能：
	<pre>#define TUYA_BLE_FEATURE_EXT_MODULE_ENABLE (1)</pre> <pre>#define TUYA_BLE_FEATURE_EXT_MODULE_ENABLE (0)</pre>
备注	

3.2 上报扩展模组信息接口

函数原型	<pre>tuya_ble_status_t tuya_ble_ext_module_info_asynchronous_response (uint8_t *p_data, uint32_t len);</pre>
功能概述	异步回复或上报扩展模组设备信息。
参数说明	<code>p_data[in]</code> ：扩展模组设备信息数据指针。 <code>len[in]</code> ：扩展模组设备信息数据长度。
返回值	<code>TUYA_BLE_SUCCESS</code> ：发送成功。 <code>TUYA_BLE_ERR_INVALID_PARAM</code> ：参数无效。 <code>TUYA_BLE_ERR_NO_MEM</code> ：内存申请失败。 <code>TUYA_BLE_ERR_NO_EVENT</code> ：其他错误。
备注	当 App 查询扩展模组的设备信息时，蓝牙设备可以调用该 API 接口，向 App 回复扩展模组的设备信息。 或者当扩展模组的设备信息发生改变时，向 App 主动上报扩展模组的设备信息。

函数原型	<code>tuya_ble_status_t</code>
	<code>tuya_ble_ext_module_info_asynchronous_response</code> (<code>uint8_t *p_data</code> , <code>uint32_t len</code>);
扩展模组的设备信息格式详见下文描述。	

3.3 查询扩展模组设备信息回调事件

回调事件	<code>TUYA_BLE_CB_EVT_QUERY_EXT_MODULE_DEV_INFO</code>
数据结构	无。
描述	App 查询扩展模组设备信息。
备注	蓝牙设备需要按照扩展模组设备信息格式将数据进行拼装，然后通过 上报扩展模组信息接口 API 发送给 App。

扩展模组设备信息格式：

<code>inc_em_info</code>	<code>emt_type</code>	<code>N_TLD</code>	<code>TLD1</code>		<code>TLDn</code>
1 字节	1 字节	1 字节	1 + 2 + n 字节		1 + 2 + n 字节

- `inc_em_info`：是否包含扩展模组信息，`0x01`：包含，`0x00`：不包含。
- `emt_type`：扩展模组类型。
 - `0x01`：NB-IoT 模组
 - `0x02`：Wi-Fi 模组
 - `0x03`：Cat.1 模组
 - `0x04`：Zigbee 模组
- `N_TLD`：后续 TLD 数据单元组的个数。
- `TLD`：数据单元组，其中：
 - Type 表示类型，占 1 字节
 - Length 为数据长度，占 2 字节

- Data 为实际数据。具体的信息见下表：

- TLD - 0x01 扩展模组信息

Type	Length	Data
------	--------	------

0x01 ~		当 emt_type 为 NB-IoT 模组时，设备信息 JSON 字符串格式：
--------	--	--

```
{  
  "pv" : "3.0" ,  
  "bv" : "3.0" ,  
  "sv" : "1.0.0" ,  
  "i" : "866242050005256" ,  
  "opt" :1,  
  "h" :28800  
}
```

其中：

pv 为 string 类型，表示协议版本。

bv 为 string 类型，表示基线版本。

sv 为 string 类型，表示固件版本。

i 为 string 类型，表示设备 IMEI。

h 为 int 类型，表示心跳时间，单位秒。

opt 为 int 类型，由于 APN 名称变化多样，因此用 **opt** 来映射表示。0：中国电信，0：中国联通，1：中国移动，2：直连涂鸦 IoT 云。

当 **emt_type** 为 **Wi-Fi** 模组时，设备信息 JSON 字符串格式：

可不包含，模组激活时自己会上报。

当 **emt_type** 为 **Cat.1** 模组时，设备信息 JSON 字符串格式：

可不包含，模组激活时自己会上报。

- TLD - 0x02 扩展模组插拔状态

Type	Length	Data
0x02	0x01	0x00: 扩展模组 弹出 状态。 0x01: 扩展模组 插入 状态。

- TLD - 0x03 通信优先级

Type LengthData

0x0: ~	一个优先级用 1 个字节表示，其顺序表示优先级从高到低。例如某设备通信优先级定义为 Bluetooth Low Energy > MQTT，则 Data 数据为 0x03, 0x01。
0x00:	LAN 局域网
0x01:	MQTT
0x02:	HTTP
0x03:	Bluetooth Low Energy
0x04:	Bluetooth Mesh
0x05:	涂鸦 Mesh
0x06:	Beacon

- TLD - 0x04 OTA 优先级，**暂未使用**
- TLD - 0x05 扩展模组绑定状态

Type	Length	Data
0x05	0x01	0x00: 扩展模组 未绑定 状态。 0x01: 扩展模组 已绑定 状态。

- TLD - 0x06 OTA 通道与版本，**暂不使用**
- TLD - 0x07 被动查询/主动上报标识

Type	Length	Data
0x07	0x01	用于标识该信息是 App 查询的还是设备主动上报的，定义如下： 0x00：被动查询。 0x01：设备主动上报。App 无需回复应答主动上报的信息。

当扩展模组为 Wi-Fi 或 Cat.1 时，模组的设备信息在激活时会上报，蓝牙设备与 App 间同步的设备信息更多的为物理上的接入状态、通信优先级和模组绑定状态。

扩展模组相关信息的回复，SDK 提供了参考 Demo。详见 `tal_feature_ext_module` 和 `tal_ble_x_demo.c` 文件。

3.4 接收到扩展模组激活信息回调事件

TUVA_BLE_CB_EVT_EXT_MODULE_ACTIVE_INFO_RECEIVED	
回调事件	
数据结构	<code>tuya_ble_ext_module_active_data_t</code>
描述	App 发送扩展模组激活信息

激活信息数据格式为：

<code>emt_type</code>	TLD
1 字节	1 + 2 + n 字节

- `emt_type`：扩展模组类型
 - 0x01：NB-IoT 模组
 - 0x02：Wi-Fi 模组
 - 0x03：Cat.1 模组
 - 0x04：Zigbee 模组
- TLD：数据单元组，其中：
 - Type 表示类型，占 1 字节
 - Length 为数据长度，占 2 字节

- Data 为实际数据。具体的信息见下表：

- 当 `emt_type` 为 **NB-IoT 模组** 时，TLD 定义为：

Type	Length	Data
0x01	~	JSON 字符串，内容主要包括 devID 和 secKey 等： { "s" : "nYqC0z6yuqJzrggALzatsB42K0PCz****" }

- 当 `emt_type` 为 **Wi-Fi/Cat.1 模组** 时，TLD 定义为：

Type	Length	Data
0x01	~	JSON 字符串，内容主要包括 devID、secKey 和 localKey 等，云端与设备端对齐，中间链路如 App、蓝牙进行透传： { "devId" : "6cba0dda87e9ae35f8c****" , "region" : "AY" , "env" : "pr_0" , "localKey" : "a427f948b8451***" }

- 当 `emt_type` 为 **Zigbee 模组** 时，TLD 定义为：

Type	Length	Data
0x01	~	PAN：2 字节。 exPAN：8 字节。 Channel：1 字节。 net key：16 字节。

4 使用方法

4.1 消息序列图

```
1 sequenceDiagram
2     participant A as 应用程序
3     participant X as X 模组
4     participant T as 涂鸦蓝牙 LE SDK
5     participant M as 手机 App
6     participant C as 云端
7
8     Note over A, C: 已连接涂鸦蓝牙 LE
9
10    M ->> T: 请求设备信息
11    T ->> A: TUYA_BLE_CB_EVT_QUERY_EXT_MODULE_DEV_INFO
12    Note over A, X: 从 X 模组获取设备信息
13    A ->> T: tuya_ble_ext_module_info_asynchronous_response()
14    T ->> M: 响应设备信息
15
16    opt 存在 X 模组
17        M ->> C: 上报设备信息
18        C ->> M: 返回激活信息
19        M ->> T: 下发激活信息
20        T ->> A: TUYA_BLE_CB_EVT_EXT_MODULE<br>_ACTIVE_INFO_RECEIVED
21        T ->> M: 应答激活信息
22        Note over A, X: 向 X 模组发送激活信息
23
24        X ->> M: 激活成功
25    end
```

4.2 回调事件处理

```
1  STATIC VOID_T tuya_ble_protocol_callback(tuya_ble_cb_evt_param_t* ev
2  ent)
3  {
4      switch(event->evt)
5      {
6          //...
7  #if defined(TUYA_BLE_FEATURE_EXT_MODULE_ENABLE) && (TUYA_BLE_FEATURE
8  _EXT_MODULE_ENABLE != 0)
9      case TUYA_BLE_CB_EVT_QUERY_EXT_MODULE_DEV_INFO: {
10         //...
11     } break;
12     case TUYA_BLE_CB_EVT_EXT_MODULE_ACTIVE_INFO_RECEIVED: {
13         //...
14     } break;
15 #endif
16     //...
17 }
18 }
```

4.3 代码文件

```
1  tal_feature_ext_module
2      include
3          tal_ble_x_demo.h
4          tal_feature_ext_module.h
5      src
6          tal_ble_x_demo.c
7          tal_feature_ext_module.c
```

5 注意事项

- 要完整支持蓝牙 + X 功能，除了需要完成该文档所描述的蓝牙端的能力配置，同时在 X 端也需要实现相关的能力支持。在使用涂鸦蓝牙 + X 完整功能时，需要首先和 X 模组端的产品或相关研发人员确定，X 模组是否支持该能力。
- 该文档中所描述的蓝牙 + X 能力，仅仅代表的是设备具有通过蓝牙链路来激活 X 扩展模组的能力，并不涉及到蓝牙与 X 模组之间的具体激活流程，以及通过 X 链路进行 DP 或 OTA 等其他涂鸦数据的交互流程。若要实现这些功能，则工程代码中还需要增加蓝牙与 X 模块之间的协议交互代码。

6 联系涂鸦

开发过程中遇到任何问题，都可以在 [涂鸦开发者论坛-蓝牙板块](#) 进行提问。

提问前建议首先查阅 [官方资料](#)或参考已有帖子，提问前请认真阅读 [发帖规范](#)。