

# 模块 透传协议 V1.30

## 目录

1 模组通讯设置 .....	3
2 模块通讯协议 .....	3
2.1 CMD_TEST = 0, /*设备通信测试*/ .....	7
2.2 CMD_VERSION = 1, /*获取软件版本*/ .....	7
2.3 CMD_MRAD = 10, /*获取最近连接的蓝牙设备地址*/ .....	7
2.4 CMD_STATE = 11, /*获取蓝牙工作状态*/ .....	8
2.5 CMD_DISCOVERABLE = 13, /*可发现状态控制*/ .....	8
2.6 CMD_DISC = 15, /*命令模组断开与手机的连接*/ .....	9
2.7 CMD_SPPDATA = 16, /*发送接收SPP数据*/ .....	9
2.8 CMD_BATMEAS = 18, /*查询电池电量*/ .....	9
2.9 CMD_POWERDOWN = 19, /*关机*/ .....	10
2.10 CMD_READADDR = 20, /*读取蓝牙模组的地址*/ .....	10
2.11 CMD_ATTDATA = 21, /*发送接收LE数据*/ .....	10
2.12 CMD_REENDISCOVERABLE = 22, /*控制断开后是否自动进入可发现/广播状态*/ .....	11
2.13 CMD_LEADVPARAMS = 23, /*设置LE广播间隔时间*/ .....	11
2.14 CMD_RSSI = 24, /*获取连接信号强度*/ .....	11
2.15 CMD_LECONPARAMS = 25, /*设置LE连接间隔时间*/ .....	12
2.16 CMD_UARTBAUD = 26, /*设置UART波特率*/ .....	12
2.17 CMD_RENAME = 27, /*修改设备名*/ .....	13
2.18 CMD_MODBTADDR = 28, /*修改模组蓝牙地址*/ .....	13
2.19 CMD_MODSVCNAME = 29, /*修改LE主服务名称*/ .....	13
2.20 CMD_ONLYBRIDGEMODE = 30, /*进入完全透明传输模式*/ .....	14
2.21 CMD_UART_RXFIFO_LEVEL = 31, /*串口接收fifo指示*/ .....	14
2.22 CMD_BT_RXFIFO_LEVEL = 32, /*蓝牙接收fifo指示*/ .....	14
2.23 CMD_POLL_TIME = 33, /*报告发送间隔时间(ms)*/ .....	15
2.24 CMD_CLASSIC_SCAN = 34, /*扫描周围Classic设备*/ .....	15
2.25 CMD_EXIT_CLASSIC_SCAN = 35, /*退出扫描周围Classic设备状态*/ .....	16
2.26 CMD_CREAT_RFCOMM = 36, /*向特定蓝牙设备发起SPP连接*/ .....	16

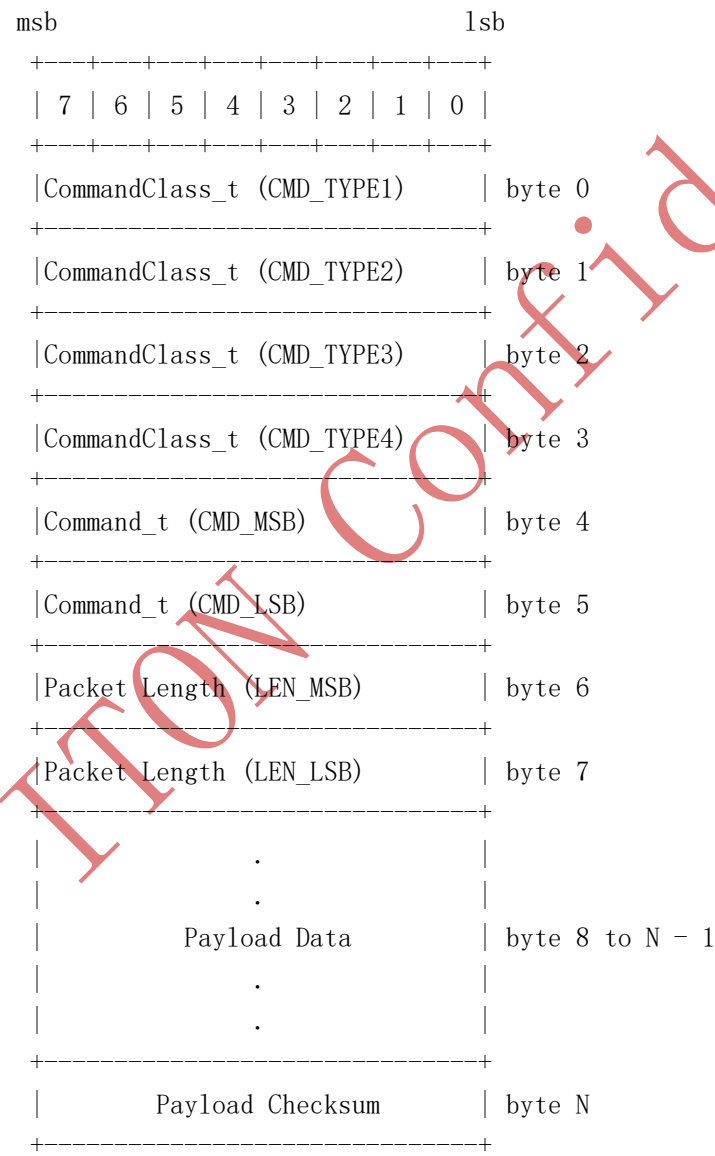
2.27 CMD_QUERY_RFCOMM = 37, /*查询特定的蓝牙设备SDP, 得到rfcomm_channel再发起SPP连接*/.....	17
2.28 CMD_SETUP_IBEACON = 38, /*设置iBeacon参数（只在uart上有效）*/.....	17
3.通讯功能.....	17
4.iBeacon功能.....	18
5.通过GPIO指示蓝牙连接.....	18
6.透传数据的方法.....	18
7. uart命令发送代码例程.....	18
8. 协议版本.....	20
8.1 模组硬件版本号.....	20
8.2 模组软件固件名称.....	20
8.3 版本说明.....	20
8.4 版本信息.....	20
8.5 版本记录.....	20
9.手机APP编写说明（LE）.....	21
10.操作流程图.....	22
10.1 模块做从机.....	22
10.2 模块做主机.....	23
11.FQA(常见问题解答).....	24

# 1 模组通讯设置

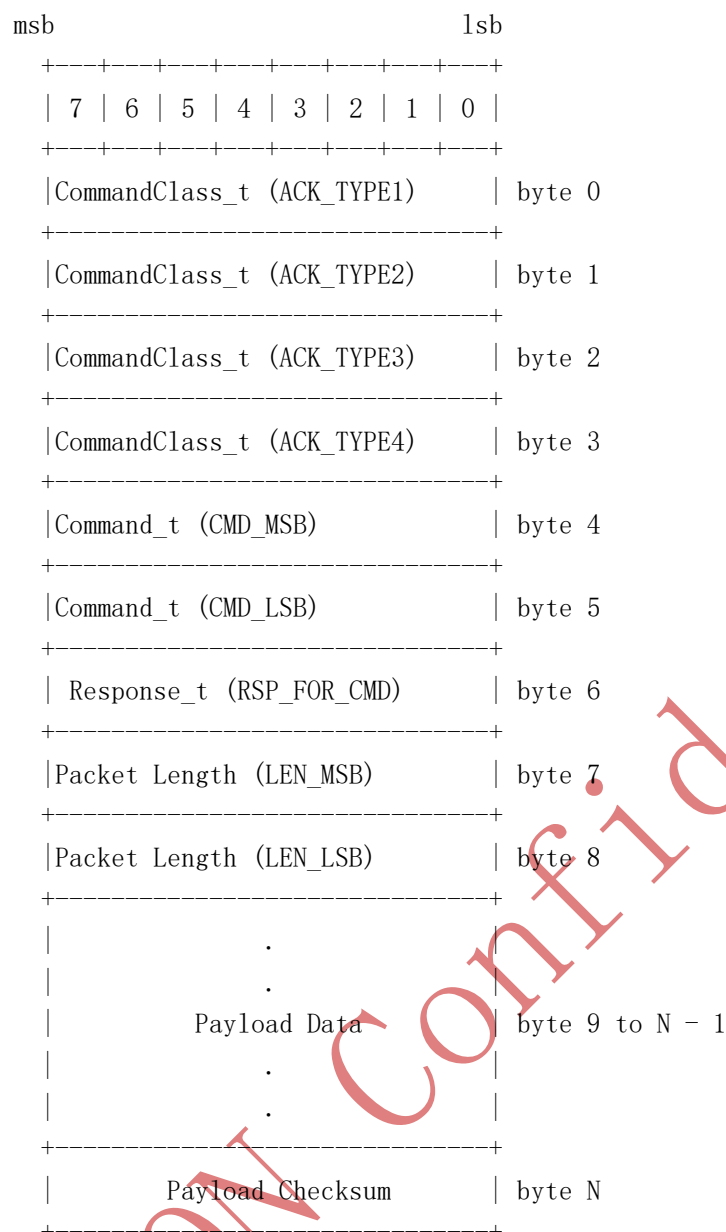
设备名称	ITON DM
接口	RS232
波特率	4800~1200000（缺省为 115200）
停止位	1
数据位	8
校验位	无

# 2 模块通讯协议

命令：



应答:



Payload Length 不包括 CommandClass\_t , command\_t 和 checksum。以上数据格式针对 HID 模式。

Checksum = (0xff - (Data 之和) ) + 0x01,其中 Checksum 和用于存放 data 之和的变量都用 uint\_8 来表示。

Checksum 只计算 Data 段, checksum 不对, 模组返回 fail, 设备收到之后可重发。

超时设置: 设备向模组发送命令, 建议使用超时时间 300ms, 300ms 之内如模组不回复, 可重发或者复位模组。

**/\*命令类型\*/**

```
typedef enum {  
    CMD_TYPE1 = 'i', /*命令类型*/ //0x69 74 63 7A  
    CMD_TYPE2 = 't', /*命令类型*/  
    CMD_TYPE3 = 'c', /*命令类型*/  
    CMD_TYPE4 = 'z', /*命令类型*/  
    ACK_TYPE1 = 'i', /*响应类型*/ //0x69 74 61 7A  
    ACK_TYPE2 = 't', /*响应类型*/  
    ACK_TYPE3 = 'a', /*响应类型*/  
    ACK_TYPE4 = 'z', /*响应类型*/  
    UNKNOW_TYPE = 0xff,  
} CommandClass_t;
```

**/\*设备和模组命令\*/**

部分命令为 GATT协议或者SPP协议专用命令（命令号后有备注专用）。

```
typedef enum {  
    CMD_TEST = 0, /*设备测试*/  
    CMD_VERSION = 1, /*获取软件版本*/  
    CMD_RESET = 2, /*模块复位*/  
    CMD_ORGL = 3, /*恢复默认状态（进入待机状态）*/  
    // CMD_HIDorSPP = 4, /*查询设备工作状态*/  
    // CMD_ADCN = 5, /*获取蓝牙配对列表中设备认证数量*/  
    // CMD_ADLS = 6, /*获取蓝牙配对列表中已认证设备地址清单*/  
    // CMD_RMSAD = 7, /*从配对列表清除指定配对设备*/  
    // CMD_RMAAD = 8, /*从配对列表删除所有配对设备*/  
    // CMD_FSAD = 9, /*从配对列表查找指定认证设备*/  
    CMD_MRAD = 10, /*获取最近连接的蓝牙认证设备地址*/  
    CMD_STATE = 11, /*获取蓝牙工作状态*/  
    // CMD_HIDCODE = 12, /*发送键码*/  
    CMD_DISCOVERABLE = 13, /*可发现状态控制*/  
    // CMD_RECONNECT = 14, /*命令模组开始回连列表手机*/  
    CMD_DISC = 15, /*命令模组断开与手机的连接*/  
    CMD_SPPDATA = 16, /*发送接收SPP数据*/ (SPP协议专用)  
    // CMD_EJECT = 17, /*发送弹IOS软键盘*/  
    CMD_BATMEAS = 18, /*查询电池电量*/  
    CMD_POWERDOWN = 19, /*关机*/  
    CMD_READADDR = 20, /*读取蓝牙模组的地址*/  
    CMD_ATTDATA = 21, /*发送接收LE数据*/ (GATT协议专用)  
    CMD_REENDISCOVERABLE = 22, /*控制断开后是否自动进入可发现状态*/  
    CMD_LEADVPARAMS = 23, /*设置LE广播间隔时间*/ (GATT协议专用)  
    CMD_RSSI = 24, /*获取连接信号强度*/  
    CMD_LECONPARAMS = 25, /*设置LE连接间隔时间*/ (GATT协议专用)  
    CMD_UARTBAUD = 26, /*设置UART波特率*/  
    CMD_RENAME = 27, /*修改设备名*/  
    CMD_MODBTADDR = 28, /*修改模组蓝牙地址*/  
}
```

```

CMD_MODSVNAME = 29, /*修改LE主服务名称*/(GATT协议专用)
CMD_ONLYBRIDGEMODE = 30, /*进入完全透明传输模式*/
CMD_UART_RXFIFO_LEVEL = 31, /*串口接收fifo指示*/
CMD_BT_RXFIFO_LEVEL = 32, /*蓝牙接收fifo指示*/
CMD_POLL_TIME = 33, /*报告发送间隔时间(ms)*/
CMD_CLASSIC_SCAN = 34, /*扫描周围Classic设备*/(SPP协议专用)
CMD_EXIT_CLASSIC_SCAN = 35, /*退出扫描周围Classic设备状态*/(SPP协议专用)
CMD_CREAT_RFCOMM = 36, /*向特定蓝牙设备发起SPP连接*/(SPP协议专用)
CMD_QUERY_RFCOMM = 37, /*查询特定的蓝牙设备SDP, 得到rfcomm_channel再发
起SPP连接*/(SPP协议专用)

CMD_UNKNOW = 0xffff, /*未知命令*/
} Command_t;

```

//注释的内容是配合 HID 协议的, 暂时不提供。

**/\*设备和模组应答\*/**

```

typedef enum {
    ACK_FAIL,          /*失败*/
    ACK_TRUE,          /*成功*/
    ACK_PENDING,       /*收到命令, 准备执行命令*/
    ACK_UNKNOWCMD,     /*收到未知命令*/
    NONE,
} Response_t;

```

注意: Response\_t 仅代表收到设备端命令(数据, checksum)的正确与否, 如为 ACK\_FAIL, 则命令需要重发。而返回的运行结果都设置在 payload 中。

如何看懂表格:

第一行表示一帧中的数据在数据结构中的位置。

第二行表示该数据实际值, 如有用[括号]标出, 表示该数据不止一个字节。例如:

CommandClass_t	Command_t	Packet Length	Data1	Data2	Data3	Data4	CHECKSUM
[CMD_TYPE]	[CMD_TEST]	[4]	'T'	'E'	'S'	'T'	0xC0

[CMD\_TYPE]相当于上文数据包格式所提到的 CMD\_TYPE1, CMD\_TYPE2, CMD\_TYPE3, CMD\_TYPE4 按顺序组成的数组。所以[CMD\_TYPE] = CMD\_TYPE1, CMD\_TYPE2, CMD\_TYPE3, CMD\_TYPE4 = 0x69, 0x74, 0x63, 0x7A。

同理[CMD\_TEST]相当于上文数据包格式所提到的 CMD\_MSB, CMD\_LSB 按顺序组成的数组, 而 CMD\_TEST 命令又可以在上文中查到是 0, 所以[CMD\_TEST] = CMD\_MSB, CMD\_LSB = 0x00, 0x00。

不括号的为纯数据。标字符标识符号''的为字符类型数据, 取 ascii 码发送即可。

以下为通信的一些基本命令和应答, 其中应答部分只列出典型的数据, 并不代表蓝牙模组只会回这些数据。

## 2.1 CMD\_TEST = 0, /\*设备通信测试\*/

CommandClass_t	Command_t	Packet Length	Data1	Data2	Data3	Data4	CHECKSUM
[CMD_TYPE]	[CMD_TEST]	[4]	'T'	'E'	'S'	'T'	0xC0

应答

CommandClass_t	Command_t	Response_t	Packet Length	TestResult	CHECKSUM
[ACK_TYPE]	[CMD_TEST]	ACK_TRUE/ ACK_FAIL	[1]	1/0	

TestResult: 1 测试数据接收正确。

0 测试数据接收不正确。

## 2.2 CMD\_VERSION = 1, /\*获取软件版本\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_VERSION]	[0]	0

应答

CommandClass_t	Command_t	Response_t	Packet Length	Data1	Data2	CHECKSUM
[ACK_TYPE]	[CMD_VERSION]	ACK_TRUE/ACK_FAIL	[2]	0x00	0x01	

注：例子中版本号为 0.1，大端格式

## 2.3 CMD\_MRAD = 10, /\*获取最近连接的蓝牙设备地址\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_MRAD]	[0]	0

应答

CommandClass_t	Command_t	Response_t	Packet Length	Addr[0]	Addr[1]	Addr[2]
[ACK_TYPE]	[CMD_MRAD]	ACK_TRUE	[6]	0xDC	0x2C	0x26

接上表：

Addr[3]	Addr[4]	Addr[5]	CHECKSUM
0x12	0x34	0x56	

注:蓝牙地址为大端格式。

## 2.4 CMD\_STATE = 11, /\*获取蓝牙工作状态\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_STATE]	[0]	0

应答

```
typedef enum {  
    usr_INITIALIZED      = 0, /*初始化状态*/  
    usr_READY            = 1, /*准备状态*/  
    usr_CONNECTING       = 2, /*连接中*/  
    usr_CONNECTED        = 3, /*已连接*/  
    usr_RECONNECTING     = 4, /*正在回连*/  
    usr_DISCONNECTED     = 5, /*断开状态*/  
    usr_INQUIRY          = 6, /*扫描周围设备*/  
    usr_DISCOVERABLE     = 1 << 4, /*可发现状态*/ //(Classic)  
    usr_ADVERTISEMENTS   = 1 << 5, /*广播状态*/ //(LE)  
    usr_CONNECTABLE      = 1 << 6, /*可连接状态*/  
} BTState_t;
```

CommandClass_t	Command_t	Response_t	Packet Length	ClassicBTState	LEBTState	CHECKSUM
[ACK_TYPE]	[CMD_STATE]	ACK_TRUE	[2]	usr_INITIALIZED	usr_INITIALIZED	

注: ClassicBTState 和 LEBTState 分别指示不同的蓝牙连接。

## 2.5 CMD\_DISCOVERABLE = 13, /\*可发现状态控制\*/

CommandClass_t	Command_t	Packet Length	Discover Type	CHECKSUM
[CMD_TYPE]	[CMD_DISCOVERABLE]	[1]	0	0

Discover Type:

0 -> 退出可发现状态

1 -> 命令 Classic 进入可发现状态

2 -> 命令 LE 进入可发现状态

3 -> 命令 Classic 和 LE 同时进入可发现状态

4 -> 命令 LE 进入不可连接状态下的广播(用于有蓝牙链接的时候广播 iBeacon 数据), Classic 进入可发现状态。

5 ->命令 LE 进入不可连接状态下的广播(用于有蓝牙链接的时候广播 iBeacon 数据), Classic 进入不可发现状态。

注: Discover Type 上电默认值为 3。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_DISCOVERABLE]	ACK_TRUE	[0]	

## 2.6 CMD\_DISC = 15, /\*命令模组断开与手机的连接\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_DISC]	[0]	0

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_DISC]	ACK_TRUE	[0]	

## 2.7 CMD\_SPPDATA = 16, /\*发送接收 SPP 数据\*/

CommandClass_t	Command_t	Packet Length	(payload)	CHECKSUM
[CMD_TYPE]	[CMD_SPPDATA]	[N]		

注:N 最大值建议为 255。

应答

CommandClass_t	Command_t	Response_t	Packet Length	Res	CHECKSUM
[ACK_TYPE]	[CMD_SPPDATA]	ACK_TRUE	[1]		

Res: 0 -> 数据发送正确, 不为 0 表示数据发送有异常。

设备通过 uart 向模组发送 cmd 并附带数据在 payload 中, 模组根据数据校验的结果和向手机发送数据的结果返回对应的 ack。手机向模组发送的数据, 模组直接向设备转发纯数据。

## 2.8 CMD\_BATMEAS = 18, /\*查询电池电量\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_BATMEAS]	[0]	0

应答

CommandClass_t	Command_t	Response_t	Packet Length	Data1	Data2	CHECKSUM
[ACK_TYPE]	[CMD_BATMEAS]	ACK_TRUE	[2]	0x0c	0x1e	

注: 数据表示方法为: 电源电压\* (采集返回值/4096), 数据为大端格式。

如电源电压 3.0V 数据为 0xc1e 时, 采集到的电压值应是  $3.0 * (0xc1e/4096) \approx 2.27V$ 。

默认 AD 输入脚为 PA.0。

## 2.9 CMD\_POWERDOWN = 19, /\*关机\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_POWERDOWN]	[0]	0

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_POWERDOWN]	ACK_TRUE	[0]	

注：应答关机命令后过 500ms 模组会进入 1uA 休眠模式。由于 bcm20710 目前还不能进入关机状态。

## 2.10 CMD\_READADDR = 20, /\*读取蓝牙模组的地址\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_READADDR]	[0]	

应答

1) 先回一个让外部 MCU 等待的数据帧。

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_READADDR]	ACK_PENDING	[0]	0

2) 再回复模组蓝牙地址

CommandClass_t	Command_t	Response_t	Packet Length	Addr[0]	Addr[1]	Addr[2]
[ACK_TYPE]	[CMD_READADDR]	ACK_TRUE	[6]	0xDC	0x2C	0x26

接上表：

Addr[3]	Addr[4]	Addr[5]	CHECKSUM
0x12	0x34	0x56	

注：蓝牙地址为大端格式。

## 2.11 CMD\_ATTDATA = 21, /\*发送接收 LE 数据\*/

CommandClass_t	Command_t	Packet Length	(payload)	CHECKSUM
[CMD_TYPE]	[CMD_ATTDATA]	[N]		

注:N 最大值建议为 255。

应答

CommandClass_t	Command_t	Response_t	Packet Length	Res	CHECKSUM
[ACK_TYPE]	[CMD_ATTDATA]	ACK_TRUE	[1]	0/1	

Res: 0 -> 数据发送正确, 不为 0 表示数据发送有异常。

客户设备通过 uart 向模组发送 cmd 并附带数据在 payload 中, 模组根据数据校验的结果和向手机发送数据的结果返回对应的 ack。手机向模组发送的数据, 模组直接向设备转发纯数据。

## 2.12 CMD\_REENDISCOVERABLE = 22, /\*控制断开后是否自动进入可发现/广播状态\*/

CommandClass_t	Command_t	Packet Length	Reen	CHECKSUM
[CMD_TYPE]	[CMD_REENDISCOVERABLE]	[1]		

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_REENDISCOVERABLE]	ACK_TRUE	[0]	

Reen:

- 0 -> 蓝牙连接断开后不进入可发现状态
  - 1 -> 命令 Classic 在断开后进入可发现状态
  - 2 -> 命令 LE 在断开后进入可发现状态
  - 3 -> 命令 Classic 和 LE 在断开后都进入可发现状态
- 注: Reen 上电默认值为 3

## 2.13 CMD\_LEADVPARAMS = 23, /\*设置 LE 广播间隔时间\*/

CommandClass_t	Command_t	PacketLength	le_adv_int_msb	le_adv_int_lsb	CHECKSUM
[CMD_TYPE]	[CMD_LEADVPARAMS]	[2]	0x40	0x00	

注:数据为大端格式, le\_adv\_int 取值范围为 0x20~0x4000, 1unit = 0.625 msec。上电默认 2048。

应答

CommandClass_t	Command_t	Response_t	Packet Length	Res	CHECKSUM
[ACK_TYPE]	[CMD_LEADVPARAMS]	ACK_TRUE	[1]		

Res: 0 -> 输入正确

1 -> 输入值超过取值范围, 在这个情况下 le\_adv\_int 会自动设置为 0x800。

## 2.14 CMD\_RSSI = 24, /\*获取连接信号强度\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_RSSI]	[0]	0

应答

1) 先回一个让外部 MCU 等待的数据帧。

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_RSSI]	ACK_PENDING	[0]	0

2) 再回复模组测到的 rssi

CommandClass_t	Command_t	Response_t	Packet Length	Rssi	CHECKSUM
[ACK_TYPE]	[CMD_RSSI]	ACK_TRUE/ACK_FAIL	[1]		

注: 如不存在连接, Response\_t 将返回 ACK\_FAIL

其中 RSSI 是有符号数据范围是-127~127。

## 2.15 CMD\_LECONPARAMS = 25, /\*设置 LE 连接间隔时间\*/

CommandClass_t	Command_t	Packet Length	conn_interval_min_msb	conn_interval_min_lsb
[CMD_TYPE]	[CMD_LECONPARAMS]	[6]		

接上表

conn_interval_max_msb	conn_interval_max_lsb	supervision_timeout_msb

接上表:

supervision_timeout_lsb	CHECKSUM

应答

CommandClass_t	Command_t	Response_t	Packet Length	Res	CHECKSUM
[ACK_TYPE]	[CMD_LECONPARAMS]	ACK_TRUE	[1]		

Res: 0 -> 输入正确

1 -> 输入值超过取值范围。

各参数取值规则:

conn\_interval\_min 取值  $\geq 20\text{ms}$ , 1unit = 1.25 ms。

conn\_interval\_max 取值  $\leq 2\text{s}$ , 1unit = 1.25 ms。

supervision\_timeout 取值  $\leq 6\text{s}$ , 1unit = 10ms。

另外  $\text{conn\_interval\_max} * 3 \leq \text{supervision\_timeout}$ 。

注意: Slave Latency 恒定设置为 0;

## 2.16 CMD\_UARTBAUD = 26, /\*设置 UART 波特率\*/

CommandClass_t	Command_t	Packet Length	Baud[0]	Baud[1]	Baud[2]	CHECKSUM
[CMD_TYPE]	[CMD_UARTBAUD]	[3]				

波特率计算方式为:  $\text{baud} = (\text{Baud}[0] \ll 16) + (\text{Baud}[1] \ll 8) + \text{Baud}[2]$

例如要设置为 921600 波特率, 则  $\text{Baud}[0] = 0x0e$ ,  $\text{Baud}[1] = 0x10$ ,  $\text{Baud}[2] = 0x00$ 。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_UARTBAUD]	ACK_TRUE	[0]	0

## 2.17 CMD\_RENAME = 27, /\*修改设备名\*/

CommandClass_t	Command_t	Packet Length	Name[0]	.....	Name[19]	CHECKSUM
[CMD_TYPE]	[CMD_RENAME]	[Packet Length <20]		.....		

注:因为设备名涉及到 classic 和 ble 的设备名,所以在当前的协议里将最大设备名称长度限制为 20 字节。

修改设备名后,模组每次上电都会使用最近修改的设备名。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_RENAME]	ACK_TRUE	[0]	0

## 2.18 CMD\_MODBTADDR = 28, /\*修改模组蓝牙地址\*/

CommandClass_t	Command_t	Packet Length	Addr[0]	.....	Addr[5]	CHECKSUM
[CMD_TYPE]	[CMD_MODBTADDR]	[6]		.....		

蓝牙地址为 6 个字节,使用大端格式。

修改蓝牙地址后,模组每次上电都会使用最近修改的蓝牙地址。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_RENAME]	ACK_TRUE	[0]	0

## 2.19 CMD\_MODSVCNAME = 29, /\*修改 LE 主服务名称\*/

CommandClass_t	Command_t	PacketLength	Name[0]	.....	Name[19]	CHECKSUM
[CMD_TYPE]	[CMD_MODSVCNAME]	[len]		.....		

注:需要使用字符类型数据写服务名称,修改服务名后,模组每次上电都会使用最近修改的服务名。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_MODSVCNAME]	ACK_TRUE	[0]	0

## 2.20 CMD\_ONLYBRIDGEMODE = 30, /\*进入完全透明传输模式\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_ONLYBRIDGEMODE]	[0]	0

进入该模式后，蓝牙模组将不会再检测各种 CMD\_TYPE 命令，直到蓝牙连接断开。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_ONLYBRIDGEMODE]	ACK_TRUE	[0]	0

## 2.21 CMD\_UART\_RXFIFO\_LEVEL = 31, /\*串口接收 fifo 指示\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_UART_RXFIFO_LEVEL]	[0]	0

应答

CommandClass_t	Command_t	Response_t	PacketLength	Fifo_level	CHECKSUM
[ACK_TYPE]	[CMD_UART_RXFIFO_LEVEL]	ACK_TRUE	[1]		

Fifo\_level: 数据范围 0~255byte。如该值一直较高，说明串口往蓝牙的转发率较低，而串口的数据写入过快，需要适当控制串口流量。

## 2.22 CMD\_BT\_RXFIFO\_LEVEL = 32, /\*蓝牙接收 fifo 指示\*/

CommandClass_t	Command_t	Packet Length	CHECKSUM
[CMD_TYPE]	[CMD_BT_RXFIFO_LEVEL]	[0]	0

应答

CommandClass_t	Command_t	Response_t	Packet Length	Fifo_level	CHECKSUM
[ACK_TYPE]	[CMD_BT_RXFIFO_LEVEL]	ACK_TRUE	[1]		

Fifo\_level: 数据范围 0~255byte。如该值一直较高，说明蓝牙往串口的转发率较低，而蓝牙发过来的数据过快，需要适当控制蓝牙流量。

## 2.23 CMD\_POLL\_TIME = 33, /\*报告发送间隔时间(ms)\*/

CommandClass_t	Command_t	Packet Length	Poll[0]	Poll[1]	CHECKSUM
[CMD_TYPE]	[CMD_POLL_TIME]	[2]			

报告发送间隔时间(ms) = (Poll[0] << 8) + Poll[1]。

例如要设置为间隔 1s 发送报告，则 Poll[0] = 0x03, Poll[1] = 0xe8。

程序会间隔这个时间检查一次发送 buff 是否有数据，并发送。报告发送间隔默认为 10ms。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_POLL_TIME]	ACK_TRUE	[0]	0

## 2.24 CMD\_CLASSIC\_SCAN = 34, /\*扫描周围 Classic 设备\*/

CommandClass_t	Command_t	Packet Length	Inquiry Interval	Num Responses	CHECKSUM
[CMD_TYPE]	[CMD_CLASSIC_SCAN]	[2]	N	P	0

注:Inquiry Interval:扫描时间 = N \* 1.28 sec, 填 0 表示一直扫描周围设备。建议值 5。

Num Responses:返回扫描设备的个数, 填 0 表示不限制个数。建议值 0。

应答

CommandClass_t	Command_t	Response_t	Packet Length	index	Addr[0]	Addr[1]	Addr[2]
[ACK_TYPE]	[CMD_CLASSIC_SCAN]	ACK_TRUE	[14]	0	0xdc	0x2c	0x26

接上表:

Addr[3]	Addr[4]	Addr[5]	Cod[0]	Cod[1]	Cod[2]	PageScanRepetitionMode
0x01	0x02	0x03	0x7a	0x02	0x0c	0x01

接上表:

ClkOffset[0]	ClkOffset[1]	RSSI	CHECKSUM
0x21	0xec	0xc9	

注:index:索引。扫描到周围的设备后就加入索引, 扫描到重复设备不会再次新建索引。索引号从 0 开始表示第一个扫描到的设备。

其中 RSSI 是有符号数据范围是-127~127.

## 2.25 CMD\_EXIT\_CLASSIC\_SCAN = 35, /\*退出扫描周围 Classic 设备状态\*/

CommandClass_t	Command_t	Packet Length	Clear Flag	CHECKSUM
[CMD_TYPE]	[CMD_EXIT_CLASSIC_SCAN]	[1]	C	0

Clear Flag: 0 -> 退出扫描状态, 不清除已加入索引的设备信息。

1 -> 退出扫描状态, 并且清除已加入索引的设备信息。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_EXIT_CLASSIC_SCAN]	ACK_TRUE	[0]	

## 2.26 CMD\_CREAT\_RFCOMM = 36, /\*向特定蓝牙设备发起 SPP 连接\*/

CommandClass_t	Command_t	Packet Length	Addr[0]	Addr[1]	Addr[2]
[CMD_TYPE]	[CMD_CREAT_RFCOMM]	[6]	0xdc	0x2c	0x26

接上表

Addr[3]	Addr[4]	Addr[5]	CHECKSUM
0x12	0x34	0x56	0

注: 蓝牙地址使用大端格式。该命令发起的 SPP 连接是不查询移动设备的 rfcomm\_channel, 直接默认 rfcomm\_channel 为 1。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_CREAT_RFCOMM]	ACK_TRUE	[0]	

## 2.27 CMD\_QUERY\_RFCOMM = 37, /\*查询特定的蓝牙设备SDP, 得到 rfcomm\_channel 再发起 SPP 连接\*/

CommandClass_t	Command_t	Packet Length	Addr[0]	Addr[1]	Addr[2]
[CMD_TYPE]	[CMD_QUERY_RFCOMM]	[6]	0xdc	0x2c	0x26

接上表

Addr[3]	Addr[4]	Addr[5]	CHECKSUM
0x12	0x34	0x56	0

注：蓝牙地址使用大端格式。该命令会发起查询移动设备的 rfcomm\_channel 号，得到对应的通道号后再连接，兼容性比较高。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_QUERY_RFCOMM]	ACK_TRUE	[0]	

## 2.28 CMD\_SETUP\_IBEACON = 38, /\*设置 iBeacon 参数（只在 uart 上有效）\*/

CommandClass_t	Command_t	Packet Length	Proximity UUID[0]	.....	Proximity UUID[15]
[CMD_TYPE]	[CMD_SETUP_IBEACON]	[21]	0x00		0x00

接上表

Major[0]	Major[1]	Minor[0]	Minor[1]	TXpower	CHECKSUM
0x00	0x00	0x00	0x00	0x00	0

注：参数都使用大端格式。

应答

CommandClass_t	Command_t	Response_t	Packet Length	CHECKSUM
[ACK_TYPE]	[CMD_SETUP_IBEACON]	ACK_TRUE	[0]	

例子以下都为 8Bit hex:

6974637a002600150102030405060708090a0b0c0d0e0f10000100040073

## 3.通讯功能

- 1、当前支持 spp、gatt 协议。
- 2、客户设备端和手机端都可以通过发送各种命令配置蓝牙模块、通过蓝牙模块收发数据。
- 3、uart 一帧数据个数最大为 255 字节（包括 CommandClass\_t, command\_t）。
- 4、spp 和 gatt 模式下不记录主机信息，也不回连主机，设备通过配置蓝牙模组使模组进入可发现和广播状态。
- 5、客户设备端和手机端不发送特征码的情况下，数据作为透传数据发送和接收。

## 4.iBeacon 功能

iBeacon 的广播数据和 ble 基本的广播数据同时广播，ble 或者 classic 有蓝牙连接后，iBeacon 会持续广播。用户可以根据需要关掉 iBeacon 的广播。

## 5.通过 GPIO 指示蓝牙连接

蓝牙模组除了可以通过 uart 命令查询蓝牙状态外还可以通过蓝牙模组的 P24 (PA.4) 管脚指示蓝牙连接，有蓝牙连接为高电平，无蓝牙连接为低电平。

## 6.透传数据的方法

使用透传功能，可以不需要记住各种命令和应答就可以使用蓝牙模组。在默认情况下，蓝牙模组上电就会使 classic 和 ble 可发现，被移动终端连上其中一种后，就会关闭另外一种的可见状态。而且在移动终端连接上蓝牙模组之后，就可以随意发送数据，蓝牙模组会进行透明传输，将数据转发到外部 mcu。使用外部 mcu 控制蓝牙模组也可以做同样的操作，蓝牙模组会把数据转发给移动终端。

## 7. uart 命令发送代码例程

```
#define MSB(x) ((x)>>8)
#define LSB(x) ((x)&0xff)
typedef enum {
    CMD_TYPE1 = 'i', /*命令类型*/ //69 74 63 7A
    CMD_TYPE2 = 't', /*命令类型*/
    CMD_TYPE3 = 'c', /*命令类型*/
    CMD_TYPE4 = 'z', /*命令类型*/
    UNKNOW_TYPE = 0xff,
} CommandClass_t;
enum CMD_PacketEnums {
    CMD_CMD_TYPE1_INDEX = 0,
```

```

    CMD_CMD_TYPE2_INDEX = 1,
    CMD_CMD_TYPE3_INDEX = 2,
    CMD_CMD_TYPE4_INDEX = 3,
    CMD_CMD_MSB_INDEX    = 4,
    CMD_CMD_LSB_INDEX    = 5,
    CMD_PKT_LEN_MSB_INDEX = 6,
    CMD_PKT_LEN_LSB_INDEX = 7,
    CMD_PAYLOAD_INDEX     = 8
};

//! Writes data to a ring buffer.
//!
//! \param ptRingBuf points to the ring buffer to be written to.
//! \param pucData points to the data to be written.
//! \param ulLength is the number of bytes to be written.
//!
//! This function write a sequence of bytes into a ring buffer.
uint8_t RingBufWrite(tRingBufObject *ptRingBuf, uint8_t *pucData,
uint16_t ulLength)
{
}

uint8_t cmd_calchecksum(uint8_t seqindex, uint8_t checkbuff[])
{
    uint8_t selfchecksum = 0, cnt = 0;
    for (cnt = CMD_PAYLOAD_INDEX; cnt < seqindex; cnt++)
    {
        selfchecksum += checkbuff[cnt];
    }
    selfchecksum = (0xff - selfchecksum) + 1;
    return selfchecksum;
}

uint8_t txpacketbuf[300] = { 0 };
uint8_t seqnum = 0;
txpacketbuf[seqnum++] = CMD_TYPE1;
txpacketbuf[seqnum++] = CMD_TYPE2;
txpacketbuf[seqnum++] = CMD_TYPE3;
txpacketbuf[seqnum++] = CMD_TYPE4;
txpacketbuf[seqnum++] = MSB(CMD_TEST);
txpacketbuf[seqnum++] = LSB(CMD_TEST);

txpacketbuf[seqnum++] = 0; //data len msb
txpacketbuf[seqnum++] = 4; //data len lsb

```

```
txpacketbuf[seqnum++] = 'T';
txpacketbuf[seqnum++] = 'E';
txpacketbuf[seqnum++] = 'S';
txpacketbuf[seqnum++] = 'T';
txpacketbuf[seqnum] = cmd_calchecksum(seqnum, txpacketbuf);
RingBufWrite(&UartTxRingBuff, txpacketbuf, (uint16_t)(txpacketbuf[CMD_PKT_LEN_MSB_INDEX]<<8)
+ txpacketbuf[CMD_PKT_LEN_LSB_INDEX] + 10);
```

## 8. 协议版本

### 8.1 模组硬件版本号

---

硬件版本号: BB2710-29 Ver1.2

### 8.2 模组软件固件名称

---

软件固件名称: BB2710-29\_A130

### 8.3 版本说明

---

例:v1.23

1.2 表示主版本号为 1.2, 该版本可以在协议中用查询版本号查询到。该版本有修改表示协议有修改和增加了功能。

0.03 表示文档编号为 3, 该位只表示文档有更新或者初始化配置有修改(例如波特率和设备名), 协议本身是不变的。

### 8.4 版本信息

---

BLE 下查询版本信息请参考

[BB2710-29 开发套件操作手册.pdf](#) 文件中的 3.3 PC 端 与 iOS 手机端 (2) 项

### 8.5 版本记录

---

2015-01-30: 1.11 初始版本。

2015-02-05: 1.12 添加版本信息查询以及更新下发 rssi 命令的部分注释, 注明 rssi 是有符号数,

数据范围-127~127

2015-02-05: 1.2 增加命令 CMD\_QUERY\_RFCOMM = 37, /\*查询特定的蓝牙设备 SDP, 得到 rfcmm\_channel 再发起 SPP 连接\*/  
修正一些术语。增加 ACK\_PENDING 应答帧的说明。

2015-02-06: 1.21 添加手机 APP 编写说明 (LE)

2015-02-28:1.22 添加操作流程图

2015-03-02:1.3 增加 iBeacon 功能, 增加命令 CMD\_SETUP\_IBEACON = 38, /\*设置 iBeacon 参数 (只在 uart 上有效) \*/  
修改命令 CMD\_DISCOVERABLE = 13, /\*可发现状态控制\*/, 增加参数。

## 9.手机 APP 编写说明 (LE)

GATT 服务列表如下, 手机端可以参考使用:

Type:	UUID(0x):	Permissions:
PRIMARY_SERVICE,	FF00	
CHARACTERISTIC,	FF01,	WRITE   DYNAMIC,
CHARACTERISTIC,	FF02,	READ   NOTIFY   DYNAMIC,
CHARACTERISTIC,	0000FF11-0000-1000-8000-00805F9B34FB	READ   NOTIFY   DYNAMIC,

其中:

0xff00 服务号。

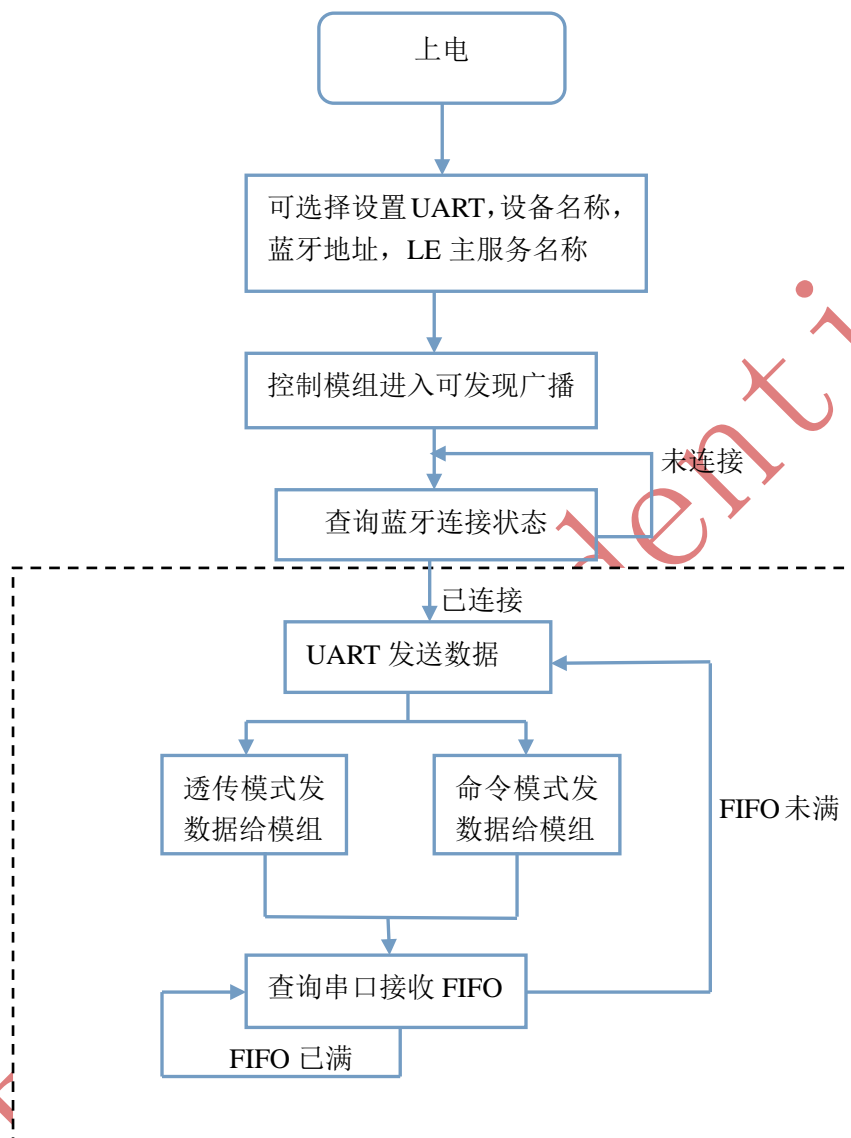
0xff01 app 发送数据给模组。

0xff02 模组发送数据给 app。

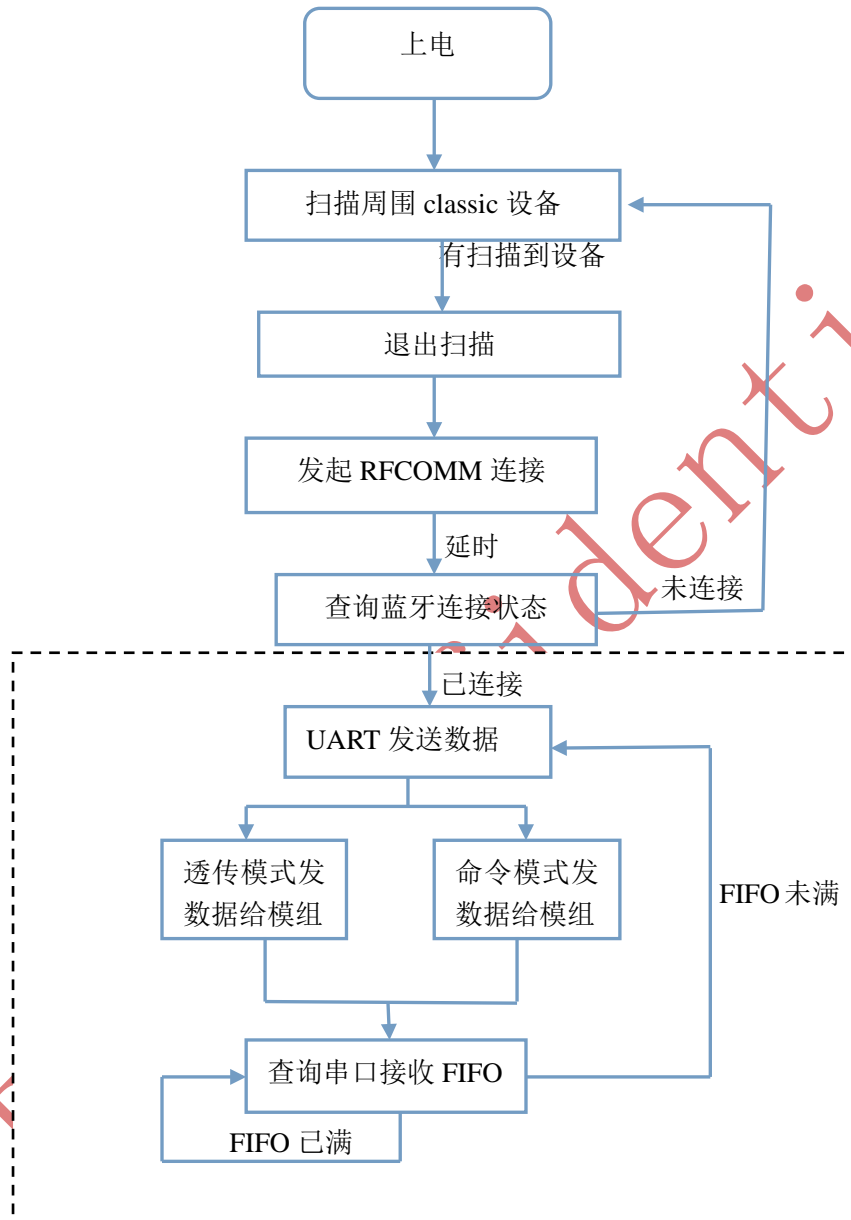
0000FF11-0000-1000-8000-00805F9B34FB 用于测试, 打开 NOTIFY 之后模组会间隔 1s 发送 count xx 给 app。

## 10. 操作流程图

### 10.1 模块做从机



## 10.2 模块做主机



## 11.FQA(常见问题解答)

1. 客户反映模组功耗大，接上电源就有 40mA。

**解析：**发现是客户的 uart 端口有问题，造成模组拉电流现象。

2. 透传数据的最大长度？

**解析：**透传数据，那就没有什么数据长度可言，无限制。只是客户在调试的时候要根据实际的 uart 波特率和 fifo 深度情况进行数据流量的控制。如果客户不要求大数据流量，都不会出现丢包的问题。

3. 测试命令 24(获取连接信号强度)，模组一直回复信号强度为 0

**解析：**获取连接信号强度。如文档所说不存在蓝牙连接的时候，Response\_t 这个位置将返回 0，如果这个字节是 0，那 rssi 是不是 0 都没有意义，因为没有蓝牙连接。

```
MCU-> 蓝牙模组: 69 74 63 7a 00 18 00 00 00
MCU <- 蓝牙模组: 69 74 61 7a 00 18 02 00 00 00
MCU <- 蓝牙模组: 69 74 61 7a 00 18 01 00 01 f2 0e
```

4. 测试命令 35(退出扫描周围的 CLASSIC 状态)，无论命令中的标志位是为 1 或 0, 测试结果都一样

**解析：**发 35 号命令的先决条件是已经发了 34 号命令。

5. 测试命令 36(向特定蓝牙发起 Rfcomm 连接)，不知道是什么意思

**解析：**这个和 4) 中所说的一样，34 号命令扫描到周围的设备，外部 mcu 选择对应设备，再通过 36 号命令对该设备发起蓝牙连接。当然如果 mcu 操作得比较快，需要先发 35 号命令退出扫描状态。

6. 模块在不同模式下，有效数据传输速率能达到多少字节？

**解析：**classic: 可以做到 255Byte/2ms。ble: 可以做到 23Byte/2ms。如需要提高空中数据率，需要发送 CMD\_POLL\_TIME = 33, /\*报告发送间隔时间(ms)\*/，默认为每间隔 10ms 发送一次数据。

7. 模块 iBeacon 下，app 搜到的广播数据断断续续

**解析：**适当设置减小广播间隔时间。