**FR801xH 错误码参考**

Bluetooth Low Energy SOC

作者：董有才

www.freqchip.com



**目录**

[1 综述 3](#_Toc35709292)

[2 错误码解释 3](#_Toc35709293)

[2.1 BLE 5.0 协议栈 3](#_Toc35709294)

[2.1.1 OP\_ID 3](#_Toc35709295)

[2.1.2 ERR\_CODE 4](#_Toc35709296)

[2.1.3 链接断开错误码 7](#_Toc35709297)

[2.1.4 创建profile失败错误码 9](#_Toc35709298)

[2.2 操作系统组件 9](#_Toc35709299)

# 综述

本文档旨在介绍 801xH 软件SDK各组件涉及到的错误码。

801xH SDK包含以下组件：

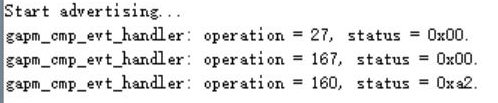
* BLE 5.0 协议栈和常见Profile
* BLE SIG Mesh协议栈
* 多个中间件组件
* 非抢占式操作系统
* 保持链接睡眠和关机睡眠调用接口
* 多种外设驱动
* 调试函数和错误处理
* 系统常用辅助函数

下面章节将逐一介绍各组件定义的错误码所代表的含义。

# 错误码解释

## **BLE 5.0 协议栈**

在调用头文件”gap\_api.h” 和”gatt\_api.h”定义的协议栈组件函数时，如果打开协议栈lib库底层的运行日志，日志运行时会按如下形式给出操作的执行结果。



*Fr801x H 协议栈lib库运行日志*

GAP操作执行结果日志采用这种格式: gapm\_cmp\_evt\_handler: operation = OP\_ID, status = ERR\_CODE.

其中OP\_ID是操作码，ERR\_CODE是执行结果代码。

### **OP\_ID**

常见的OP\_ID如下表所示

|  |  |  |
| --- | --- | --- |
| OP\_ID | 操作名称 | 含义 |
| 1 | GAPM\_RESET | 协议栈重启 |
| 3 | GAPM\_SET\_DEV\_CONFIG | 协议栈配置 |
| 23 | GAPM\_RESOLV\_ADDR | 执行Resolve addr解析动作 |
|  | … |  |
| 160 | GAPM\_CREATE\_ADV\_ACTIVITY | 底层创建一个广播动作。 |
| 161 | GAPM\_CREATE\_SCAN\_ACTIVITY | 底层创建一个扫描动作。 |
| 162 | GAPM\_CREATE\_INIT\_ACTIVITY | 底层创建一个主动连接动作。 |
| 163 | GAPM\_CREATE\_PERIOD\_SYNC\_ACTIVITY | 底层创建一个周期性同步动作。 |
| 164 | GAPM\_START\_ACTIVITY | 底层启动一个动作 |
| 165 | GAPM\_STOP\_ACTIVITY | 底层停止一个动作 |
| 166 | GAPM\_STOP\_ALL\_ACTIVITIES | 底层停止所有动作 |
| 167 | GAPM\_DELETE\_ACTIVITY | 底层删除一个动作 |
| 168 | GAPM\_SET\_ADV\_DATA | 底层设置广播动作的adv data |
| 169 | GAPM\_SET\_SCAN\_RSP\_DATA | 底层设置广播动作的adv rsp data |
| 170 | GAPM\_SET\_PERIOD\_ADV\_DATA | 底层设置周期性广播的adv data |

### **ERR\_CODE**

ERR\_CODE是OP\_ID执行的结果反馈。常见的ERR\_CODE被定义在components\ble\include\ble\_hl\_error.h头文件内，下面逐一解释如下。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x0 | GAP\_ERR\_NO\_ERROR | 执行成功，无错误 |

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x1 | ATT\_ERR\_INVALID\_HANDLE | GATT的操作的handler号无效 |

解决办法：

1 在client profile事件处理回调函数，动作完成分支，如果动作为GATT\_OP\_PEER\_SVC\_REGISTERED，打印扫描到的UUID对应的handler号是否为非0，如果为0，表示该UUID对应的handler没有扫描到，对该UUID对应的属性操作时，会产生上述错误。

示例代码如下

case GATTC\_MSG\_CMP\_EVT:

{

if(p\_msg->param.op.operation == GATT\_OP\_PEER\_SVC\_REGISTERED)

{

uint16\_t att\_handles[2]; //只定义了2个感兴趣的UUID，缓存它们的handler号

memcpy(att\_handles,p\_msg->param.op.arg,4); //拷贝上传的handler号

show\_reg((uint8\_t \*)att\_handles,4,1); //打印UUID对应的handler号

}

}

2 检查client\_att\_table定义的UUID，是否存在于对端的服务中。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x2 | ATT\_ERR\_READ\_NOT\_PERMITTED | GATT的读操作禁止 |

解决办法：

1 检查进行读操作的UUID，在对端服务中，有没有定义读操作的权限。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x3 | ATT\_ERR\_WRITE\_NOT\_PERMITTED | GATT的写操作禁止 |

解决办法：

1 检查进行读操作的UUID，在对端服务中，有没有定义写操作的权限。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x40 | GAP\_ERR\_INVALID\_PARAM | GAP操作输入参数非法 |

解决办法：

1 查找GAP操作对应的OP\_ID，找到执行该操作的API函数，检查输入的参数是否合法。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x42 | GAP\_ERR\_NOT\_SUPPORTED | GAP操作因为协议栈配置导致不支持 |

解决办法：

1 查找GAP操作对应的OP\_ID，询问FAE人员，协议栈底层lib库协议栈配置是否支持该GAP操作。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x43 | GAP\_ERR\_COMMAND\_DISALLOWED | GAP操作不被允许 |

解决办法：

1 通常该错误原因是，有同一个GAP操作正在执行，此时应用层再次调用该GAP操作，导致后面的GAP操作执行结果报这个错误。需要应用层检查是否存在前一个操作未完成的情况下，再次调用相同的操作。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x44 | GAP\_ERR\_CANCELED | GAP操作被取消 |

解决办法：

1 该ERR\_CODE不代表错误，代表某个正在执行的GAP操作被提前终止掉，比如调用gap\_scan\_stop()函数终止正在进行的扫描动作，就会上传该err\_code。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x45 | GAP\_ERR\_TIMEOUT | GAP操作超时导致被终止 |

解决办法：

1 找到该操作OP\_ID，查找调用该操作的gap API函数，是否有设置执行时间，比如启动广播的函数gap\_start\_advertising(uint8\_t)输入参数即为动作执行时间，如果非0，广播时间到后，广播动作会终止，然后上传该ERR\_CODE

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x46 | GAP\_ERR\_DISCONNECTED | GAP操作被终止因为链接已经断开 |

解决办法：

1 该ERR\_CODE不需要做进一步的纠错动作，这是底层通知某个GAP或GATT操作因为链接断开而被提前终止，比如调用gap\_conn\_param\_update(…) API进行链接参数更新时，如果在参数更新动作完成之前，链接断开，底层会打印该ERR\_CODE的日志。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x48 | GAP\_ERR\_REJECTED | GAP操作被对端拒绝 |

解决办法：

1 查找产生该ERR\_CODE的OP\_ID，找到调用该操作的API函数，某些操作被拒绝后会上传到GAP或GATT的回调函数内，应用层需要做进一步的处理。比如调用gap\_conn\_param\_update(…) API进行链接参数更新时，如果在参数被对端拒绝，底层会上传链接参数被拒绝的事件，应用层需要用新的链接参数进行更新或终止参数更新。

示例

case GAP\_EVT\_LINK\_PARAM\_REJECT:

gap\_evt\_link\_param\_reject\_t \*cs = &(event->param.link\_reject);

if(cs->status == GAP\_ERR\_COMMAND\_DISALLOWED)

break;

if(cs->status != GAP\_ERR\_REJECTED && cs->status != LL\_ERR\_UNACCEPTABLE\_CONN\_INT

&& cs->status != GAP\_ERR\_INVALID\_PARAM)

os\_timer\_start(&update\_param\_timer,3000,0);

else

os\_timer\_stop(&update\_param\_timer);

break;

该示例中，如果链接参数更新失败，原因是GAP\_ERR\_REJECTED 或 GAP\_ERR\_INVALID\_PARAM，则停止更新参数的定时器。否则启动更细参数定时器，继续更新链接参数。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x4A | GAP\_ERR\_ADV\_DATA\_INVALID | 设置广播Adv data或adv response data非法 |

解决办法：

1 检查调用void gap\_set\_advertising\_data(uint8\_t \*p\_adv\_data, uint8\_t adv\_data\_len); 和void gap\_set\_advertising\_rsp\_data(uint8\_t \*p\_rsp\_data, uint8\_t rsp\_data\_len);设置广播数据和广播回复数据时，数据格式是否有重复的字段，数据的总长度是否超过限制。

针对经典广播，adv 的数据长度最长为0x1F - 3。 Adv rsp data的数据长度最长为0x1F。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x75 | SMP\_ERROR\_REM\_PAIRING\_NOT\_SUPP | 对端不支持绑定操作。 |

解决办法：

1 该ERR\_CODE表明，对端设备不支持绑定操作

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x77 | SMP\_ERROR\_REM\_CMD\_NOT\_SUPPORTED | 本机不支持某个绑定的子动作。 |

解决办法：

1 该ERR\_CODE表明，本地设备不支持某个绑定子操作

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x78 | SMP\_ERROR\_REM\_UNSPECIFIED\_REASON | 因为某个原因绑定动作失败。 |

解决办法：

1 该ERR\_CODE表明，因为某个原因绑定动作失败，进一步的处理，请咨询FAE人员。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x9B | LL\_ERR\_ACL\_CON\_EXISTS | 针对同一个设备的链接已经存在了。 |

解决办法：

1 该ERR\_CODE表明，应用层发起主动连接动作去连一个已经建立链接的对端设备。应用层需要检查做主机时主动连接的函数是否在连接一个已经链接上的设备。设备由mac地址区分。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x9D | LL\_ERR\_CONN\_REJ\_LIMITED\_RESOURCES | 因为内存不足，导致主动连接动作失败。 |

解决办法：

1 该ERR\_CODE表明，内存不够了，导致主机去主动连接时，内部分配内存资源失败。应用层可以参考《FR8010X H内存泄漏调试》来查看是否有内存泄漏的问题。如果没有内存泄漏，应用层通过调用操作系统组件”os\_mem.h”定义的uint16\_t os\_get\_free\_heap\_size(void) 打印系统剩余的内存大小，查看是否存在剩余内存过小的问题。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0xA2 | LL\_ERR\_INVALID\_HCI\_PARAM | 发送到协议栈链路层的hci命令参数非法。 |

解决办法：

1 该ERR\_CODE表明，协议栈host层发送到链路层的hci命令的参数不合法。具体的原因请咨询FAE人员

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0xA8 | LL\_ERR\_PAIRING\_NOT\_ALLOWED | 绑定终止因为链路层不允许进行绑定操作。 |

解决办法：

1 该ERR\_CODE表明，绑定操作结束，因为本地设备链路层配置不支持绑定

### **链接断开错误码**

Fr801xH SDK框架下应用程序会遇到各种断开链接的原因，在设置了GAP事件回调函数后，链接断开的错误码信息会被上传。该错误码是单独编码，不属于上面介绍的GAP和GATT 的错误码集合。

示例代码

case GAP\_EVT\_DISCONNECT:

{

co\_printf("Link[%d] disconnect,reason:0x%02X\r\n",p\_event->param.disconnect.conidx

,p\_event->param.disconnect.reason);

}

break;

打印的log信息示例：Link[0],disconnect,reason:0x08. 其中Link[0]表示链接号为0的链接断开，断开的错误码是0x08.

下面分别介绍常见的断开链接错误码以及对应的处理措施。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x08 | LL\_ERR\_CON\_TIMEOUT | 链接因为握手超时导致断开。 |

解决办法：

1 该ERR\_CODE表明，链接因为握手连续失败，达到链接超时断开时间，底层主动断开链接。握手失败的可能原因是：a 链接参数设置的链接间隔时间太长，超过1秒，应用层软件通过链接参数更新API减少握手间隔。b 硬件天线的频偏与匹配参数没有调试过，需要咨询FAE人员如何调试这两个天线性能参数。c SDK版本太旧，射频参数没有更新。需要从git hub上下载最新Fr801x H SDK。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x13 | LL\_ERR\_REMOTE\_USER\_TERM\_CON | 链接被对端设备断开。 |

解决办法：

1 该ERR\_CODE表明，某个链接被对端设备主动断开。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x16 | LL\_ERR\_CON\_TERM\_BY\_LOCAL\_HOST | 链接被对本地设备自己主动断开。 |

解决办法：

1 该ERR\_CODE表明，某个链接被本地设备主动断开，在应用层调用了void gap\_disconnect\_req(uint8\_t conidx) API断开某个链接后，会打印该log信息。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x1F | LL\_ERR\_UNSPECIFIED\_ERROR | 握手接收失败，导致开窗过大，底层主动断开链接。 |

解决办法：

1 该ERR\_CODE表明，链接被底层主动断开，原因是握手失败导致开窗过大。解决办法参考ERR\_CODE:0x08的解决办法

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x22 | LL\_ERROR\_LMP\_RSP\_TIMEOUT | 发送的控制包未收到对端回复超时(默认是40秒)，主动断开。或发送断开链接包之后，链接超时时间内没有收到对端的ack。 |

解决办法：

1 该ERR\_CODE表明，对端在规定时间40s内没有回复控制包，或者链接超时时间没有ack 本地主动断开链接的包。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x3D | LL\_ERROR\_TERMINATED\_MIC\_FAILURE | 执行加密操作时，发现密码不对,主动断开 |

解决办法：

1 该ERR\_CODE表明，对链接进行加密操作时，对端的密码不对，底层主动断开链接。查看

a) 做为主机有没有针对未绑定设备直接进行加密链接操作，做为主机有没有调用绑定管理的绑定信息删除函数，删掉绑定信息。有没有更改绑定信息存储flash的起始地址。

b) 做为从机时，有没有调用绑定管理的绑定信息删除函数，删掉绑定信息。有没有更改绑定信息存储flash的起始地址。

|  |  |  |
| --- | --- | --- |
| ERR\_CODE | 名称 | 含义 |
| 0x3E | LL\_ERROR\_CONN\_FAILED\_TO\_BE\_EST | 建立链接时，没有收到第一包数据，建立链接失败，主动断开。 |

解决办法：

1 该ERR\_CODE表明，在链接建立时，未收到对端设备第一包的数据，导致链接没有建立起来。此种问题很可能跟天线射频性能有关系，参考ERR\_CODE:0x08的处理办法，调试天线频偏和匹配参数，同时更新SDK到最新版本。

### **创建profile失败错误码**

应用层调用proifile创建函数uint8\_t gatt\_add\_service(gatt\_service\_t \*p\_service) 和 uint8\_t gatt\_add\_client(gatt\_client\_t \*p\_client)后，返回值定义如下

|  |  |  |
| --- | --- | --- |
| 返回值 | 名称 | 含义 |
| 0xff | PRF\_ID\_INVALID | Profile创建失败。 |

解决办法：

1 检查应用层定义的Profile总数有没有超过8个，Profile创建个数超过8个后，再创建Profile就会返回该值。

|  |  |  |
| --- | --- | --- |
| 返回值 | 名称 | 含义 |
| 0~7 | PRF\_ID | Profile创建成功，返回被分配的profile ID号 |

## **操作系统组件**

该组件在调用任务创建函数uint16\_t os\_task\_create(os\_task\_func\_t task\_func)后，返回值定义如下

|  |  |  |
| --- | --- | --- |
| 返回值 | 名称 | 含义 |
| 0xff | TASK\_ID\_FAIL | 任务创建失败。 |

解决办法：

1 检查应用层定义的任务总数有没有超过50个，任务数超过50个后，再创建任务就会返回该值。

2 系统剩余内存不足，导致创建任务失败。调用uint16\_t os\_get\_free\_heap\_size(void) 查看系统剩余的内存大小。

|  |  |  |
| --- | --- | --- |
| 返回值 | 名称 | 含义 |
| 0~50 | TASK\_ID | 任务成功，返回被分配的任务ID号 |