

## 一、需要支持建立 IAP 的连接

## 1、在截图的文件定义代码段。

```
me/lihaihua/br22 master/SDK/apps/common/config/bt profile config.c
25/2020 05:16:49 PM 5,483 bytes C,C++,C#,ObjC Source UTF-8:
83 FILTERED LINES
84 u8 hfp_profile_support = 1;
85 SDP_RECORD_HANDLER_REGISTER(hfp_sdp_record_item) = {
86     .service_record = (u8 *)sdp_hfp_service_data,
87     .service_record_handle = 0x00010003,
88 };
89 #endif
90 #define USER_SUPPORT_PROFILE_IAP 1
91 #if (USER_SUPPORT_PROFILE_IAP==1)
92 u8 iap_profile_support = 1;
93 extern const u8 sdp_iap_service_data[];
94 SDP_RECORD_HANDLER_REGISTER(iap_sdp_record_item) = {
95     .service_record = (u8 *)sdp_iap_service_data,
96     .service_record_handle = 0x00010008,
97 };
98 #endif
```

```
#define USER_SUPPORT_PROFILE_IAP 1
#if (USER_SUPPORT_PROFILE_IAP==1)
u8 iap_profile_support = 1;
extern const u8 sdp_iap_service_data[];
SDP_RECORD_HANDLER_REGISTER(iap_sdp_record_item) = {
    .service_record = (u8 *)sdp_iap_service_data,
    .service_record_handle = 0x00010008,
};
#endif
```

## 2、自己定义一个同名函数用于处理数据

```
void __iap2_data_deal(u8 packet_type, u16 channel, u8 *packet, u16 size)
{
    switch (packet_type) {
    case 1:
        printf("iap connect#####handler\n");
        break;
    case 2:
        printf("iap disconnect#####handler\n");
        break;
    case 7:
        break;
    }
}
```

## 3、IAP 的发送函数。

```
extern int user_iap_send_data(u8 rfcomm_cid, u8 *buf, u16 len);
```

如果是支持多个手机连接的应用，参数 rfcomm\_cid 的值需要用回调函数 \_\_iap2\_data\_deal 的 channel 值。如果只是支持一个手机连接，那么可以直接传参数 0，会找当前连接的地址去连接。没有就会直接丢包了。

## 二、用户自定义支持多个 RFCOMM 应用。

1、注意，每个 RFCOMM 的 UUID 是要唯一的。库里面默认已经支持了一个 UUID 为 0x1101 的 spp 服务，对应的 RFCOMM cid 是 0xA。一般用户不需要修改定义了。

如果用户想自己添加其它 UUID 的服务，比如 AMA，baidu 之类的，可以自己修改一个 SDP 的服务定义。类似

```
SDP_RECORD_HANDLER_REGISTER(spp_sdp_record_item) = {
    .service_record = (u8 *)sdp_dueros_spp_service_data,
    .service_record_handle = 0x0001000c,
};
```

PS: 不懂 sdp 服务格式的可提供 UUID，联系杰理提供需要的数组

2、自定义的用的本地 RFCOMM CID 可使用的有 0xB, 0xC, 0xD 三个

### 3、自己定义一个同名函数用于处理数据

```
int mutil_handle_data_deal(u8 local_id, u8 packet_type, u16 channel, u8 *packet, u16 size)
{
    printf("local_id %d\n", local_id);
    switch (packet_type) {
    case 1:
        printf("mutil connect#####weak handler\n");
        break;
    case 2:
        printf("nutil disconnect#####weak_handler\n");
        break;
    case 7:
        put_buf(packet, size);
        break;
    }
    return 1;    //return 1 就不会从原来默认的回调函数返回了。
}
```

### 4、同时有多个 RFCOMM 服务连接的发数接口。

```
extern int multi_spp_send_data(u8 local_cid, u8 remote_cid, u8 *buf, u16 len);
```

如果只支持一个手机的连接, 那么第一个参数 local\_cid, 是跟 SDP 信息的 RFCOMM CID 对应, 目前提供的有 0xA, 0xB, 0xC, 0xD。可以根据自己定义的对对应关系给不同的服务发数据。这种情况 remote\_cid 直接填 0。

例如: multi\_spp\_send\_data(0xa, 0, "3333", 4);

如果支持两个手机的连接, 那么就要自己记录 mutil\_handle\_data\_deal 的 channel 值传到第二个参数 remote\_cid, local\_cid 传 0。

### 5、RFCOMM 的回连有一个搜索 sdp 服务的流程。需要 rfcomm 回连的细节再联系杰理沟通, 需要定制添加。