BLE应用

前言

针对WCH BLE协议栈一些问题的通用的解决方案。\ 注：对于CH579/CH573/CH582若存在兼容问题应特别注明。

1.从机多绑定切换问题

说明

当设备作为从机，通过条件触发（如按键）实现连接不同的主机的功能。\ 应用举例：蓝牙键盘通过按键连接不同的主机。

实现方法

方法1

- 通过切换不同的MAC地址与不同的设备绑定。此方法最为简单，但是弊端也很明显，由于一颗芯片只有一个独一无二的MAC地址，切换的MAC若是人为的存在与其他蓝牙MAC地址重复的概率。 若生产过程中给同一芯片烧录多个MAC地址可完美解决此问题。

方法2

在应用层对绑定进行管理,简而言之，应用层获取并记录所有绑定设备的身份地址（注1）并编好号，当想要和编号1的主机连接时，拒绝与其他编号的主机连接（注2）。 详解与代码实现：

首先，创建对绑定设备管理的结构体

```
typedef struct
{
    UINT8 addr_type;
    UINT8 McaAddr[6];
    UINT8 Direct_flag;
}Device_ID_;

typedef struct
{
    UINT8 CHECKFLAG;
    UINT8 ID_Num;
    Device_ID_ ID[4]; //here
    UINT8 mod;
    Led_Info_  Led_Info;
    RF_Pair_Info_  RF_Pair_Info;
}Device_Info_;
```

然后，写好HID应用，使能绑定功能

需要注意的是，这边需要关闭GAPBOND_ERASE_AUTO功能，因为在多绑定的应用中，很容易出现绑定的设备超过绑定列表大小的问题，若开启自动擦除，先前绑定的设备回连回产生问题===

```
uint8 erase = DISABLE;
GAPBondMgr_SetParameter( GAPBOND_ERASE_AUTO, sizeof( uint8 ), &erase );
```

接着就到了关键的地方，获取设备的身份地址，只有在与设备建立绑定后，才会获取设备的身份地址，所以在绑定状态回调中再获取身份地址

```
gapBondRec_t bondRecn;
tmos_snv_read(mainRecordNvID(n), sizeof(gapBondRec_t), &bondRecn); //n绑定设备的索引（bondIdx），如第一个绑定的设备n=0，第二个n=1...
memcpy( MacAddrn,bondRecn.publicAddr, 6 );
```

这边由于gapBondRec_t中stateFlags 的获取方式协议栈没给出（其实有相关代码只是没有放出来，获取后期会放出），所以只能上层管理，如绑定了第一个设备查询bondIdx0 查询到主机的省份地址后，将地址储存起来自己管理

```
memcpy( Device_Info.ID[n].McaAddr ,MacAddrn, 6);
```

然后需要计算地址的地址类型

```
uint8 gapDetermineAddrTypee( uint8 *addr)
{
  uint8 tmp;
  if( addr != NULL){
    tmp = addr[5] & 0xC0;
    if ( tmp == 0xC0 ){
      return ADDRTYPE_STATIC;
    }
        else if ( tmp == 0x40 ){
      return ADDRTYPE_PRIVATE_RESOLVE;
    }
      else if(tmp == 0x80)
      {
            return ADDRTYPE_PUBLIC;
      }

      else{
      return ADDRTYPE_PRIVATE_NONRESOLVE;
    }
  }
}
Device_Info.ID[n].addr_type = gapDetermineAddrTypee(MacAddrn);
```

最后，地址类型和地址都获取到了，只需要禁止指定地址连接即可

```
GAPBondMgr_SetParameter( GAPBOND_DISABLE_SINGLEBOND ,7, &Device_Info.ID[0].addr_type );
```

注1：身份地址为唯一的地址，手机设置里看到的蓝牙地址就是身份地址，然而手机与设备连接时使用的地址是随机的，所以在连接状态回调中获取的地址（连接地址）是随机的。 注2：拒绝与主机连接并不是不连接，而是连接上之后立马主动断开，断开的时间空隙让其他主机连接。个人认为此方法并不是很完善，但是了解的业内似乎都是这么做的，可能暂时没有更好的办法。

2.HID设备添加报表

说明

在蓝牙HID设备，如蓝牙键盘中增加多媒体报表的示例，本示例为新增report ID，如原来键盘是Reprot ID 1，新增多媒体报表 Report ID 2 。不适用在Report ID 1重复合其他功能。

1.在添加报表

在hidkbdservice.c中修改报表，将整理好的consumer报表加到现有报表hidReportMap。具体协议参考USB报表。

2.新建consumer配置文件中变量

```
// HID Report characteristic, Consumer input
static uint8 hidReportConsumerInProps = GATT_PROP_READ | GATT_PROP_NOTIFY;
static uint8 hidReportConsumerIn;
static gattCharCfg_t hidReportConsumerInClientCharCfg[GATT_MAX_NUM_CONN];
// HID Report Reference characteristic descriptor, Consumer input
static uint8 hidReportRefConsumerIn[HID_REPORT_REF_LEN] =
            { HID_RPT_ID_CONSUMER_IN, HID_REPORT_TYPE_INPUT };
```

3.在属性表中添加Consumer属性

```
  {
  { ATT_BT_UUID_SIZE, characterUUID },
  GATT_PERMIT_READ,
  0,
  &hidReportConsumerInProps
},

  // HID Report characteristic, Consumer input
  {
    { ATT_BT_UUID_SIZE, hidReportUUID },
    GATT_PERMIT_ENCRYPT_READ,
    0,
    &hidReportConsumerIn
  },

  // HID Report characteristic client characteristic configuration
  {
    { ATT_BT_UUID_SIZE, clientCharCfgUUID },
    GATT_PERMIT_READ | GATT_PERMIT_ENCRYPT_WRITE,
    0,
    (uint8 *) &hidReportConsumerInClientCharCfg
  },

  // HID Report Reference characteristic descriptor, Consumer input
  {
    { ATT_BT_UUID_SIZE, reportRefUUID },
    GATT_PERMIT_READ,
    0,
    hidReportRefConsumerIn
  },
```

4.根据属性表修改属性表index（注意顺序对应）

```
HID_REPORT_CONSUMER_IN_DECL_IDX, // HID Report characteristic, Consumer input declaration
HID_REPORT_CONSUMER_IN_IDX,      // HID Report characteristic, Consumer input
HID_REPORT_CONSUMER_IN_CCCD_IDX, // HID Report characteristic client characteristic configuration
HID_REPORT_REF_Consumer_IN_IDX,  // HID Report Reference characteristic descriptor, Consumer input
```

5.将Consumer报表注册进蓝牙协议栈

```
GATTServApp_InitCharCfg( INVALID_CONNHANDLE, hidReportConsumerInClientCharCfg );

hidRptMap[n].id = hidReportRefConsumerIn[0];
hidRptMap[n].type = hidReportRefConsumerIn[1];
hidRptMap[n].handle = hidAttrTbl[HID_REPORT_CONSUMER_IN_IDX].handle;
hidRptMap[n].cccdHandle = hidAttrTbl[HID_REPORT_CONSUMER_IN_CCCD_IDX].handle;
hidRptMap[n].mode = HID_PROTOCOL_MODE_REPORT;
```

至此，多媒体报表已经添加完成，如需上传多媒体数据，直接调用HidDev_Report(uint8 id, uint8 type, uint8 len, uint8*pData)函数即可，注意report ID与数据长度需要与报表中对应。

评论

WCH技术wiki