

## SYD8811 HID 例程从机控制何时配对的方法

对于普通的 HID 例程，蓝牙连接上的时候是会自动配对的，但是有些奇怪的应用，在连接上的时候不能够进行配对，而是由用户调用某个 API 后再配对，而且功能还要能够正常使用！

这里就需要使用到 Service\_Change 的功能，这里首先要保证 Profile 中有 GENERIC\_ATTRIBUTE 服务：

Handle: 0x0001 (1)

Service Name: **GENERIC\_ATTRIBUTE**

UUID: 1801

Handle: 0x0008 (8)

Characteristic

Name: GATT.SERVICE\_CHANGED

UUID: 2A05

Handle: 0x0009 (9)

ValHdl: 0x000A (10)

Encrypt: ☐

Property:

- ☐ Broadcast
- ☐ Write
- ☐ SignedWrite
- ☒ Read
- ☐ Notify
- ☐ WriteWithoutResponse
- ☒ Indicate

Field	Value
Start of Affected Attribute Handle Range	0x0000
End of Affected Attribute Handle Range	0x0000

Service Name: **UART**

然后去掉 HID 的 Encrypt 勾选，这样的 Profile，iOS 手机在连接上后就不会弹出配对绑定接口了！也就说没有进入加密流程：

Debug Log	
Time	Log
17:13:01:493	SYD RTT Init
17:13:01:493	Syd8811_HID Apr 8 2020:16:46:59 StartAdv
17:13:04:497	GAP_EVT_CONNECTED addr:[0000] F9 ED B4 C7 FC 6F
17:13:05:008	GENERIC_ATTRIBUTE enable
17:13:06:874	hid_key_tx enable

这里所有去掉的勾选如下（不同的 profile 下面是不一样的）：

Characteristic

Name:

UUID:

Handle: 0x001C (28)

ValHdl: 0x001D (29)

Encrypt: ☒

Property: ☐ Broadcast ☒ Read ☐ WriteWithoutResponse  
☐ Write ☒ Notify ☐ Indicate  
☐ SignedWrite

Field	Value	Value R
Report Value(uint8)	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00	<input checked="" type="checkbox"/>

Characteristic

Name:

UUID:

Handle: 0x0020 (32)

ValHdl: 0x0021 (33)

Encrypt: ☒

Property: ☐ Broadcast ☒ Read ☐ WriteWithoutResponse  
☐ Write ☒ Notify ☐ Indicate  
☐ SignedWrite

Field	Value	Value R
Report Value(uint8)	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00	<input checked="" type="checkbox"/>

Characteristic

Name:

UUID:

Handle: 0x0024 (36)

ValHdl: 0x0025 (37)

Encrypt: ☒

Property: ☐ Broadcast ☒ Read ☐ WriteWithoutResponse  
☐ Write ☒ Notify ☐ Indicate  
☐ SignedWrite

Field	Value	Value R
Report Value(uint8)	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00	<input checked="" type="checkbox"/>

Characteristic

Name:

UUID:

Handle: 0x0028 (40)

ValHdl: 0x0029 (41)

Encrypt: ☒

Property: ☐ Broadcast ☒ Read ☒ WriteWithoutResponse  
☒ Write ☐ Notify ☐ Indicate  
☐ SignedWrite

Field	Value	Value R
Report Value(uint8)	0x00	<input checked="" type="checkbox"/>

Characteristic

Name:

UUID:

Handle: 0x002B (43)

ValHdl: 0x002C (44)

Encrypt: ☒

Property: ☐ Broadcast ☒ Read ☐ WriteWithoutResponse  
☐ Write ☐ Notify ☐ Indicate  
☐ SignedWrite

从上面的 log 可以看出 IOS 手机连接上后同时打开了 GENERIC\_ATTRIBUTE 的 Notify,所以这时候数据是可以发送出去的!

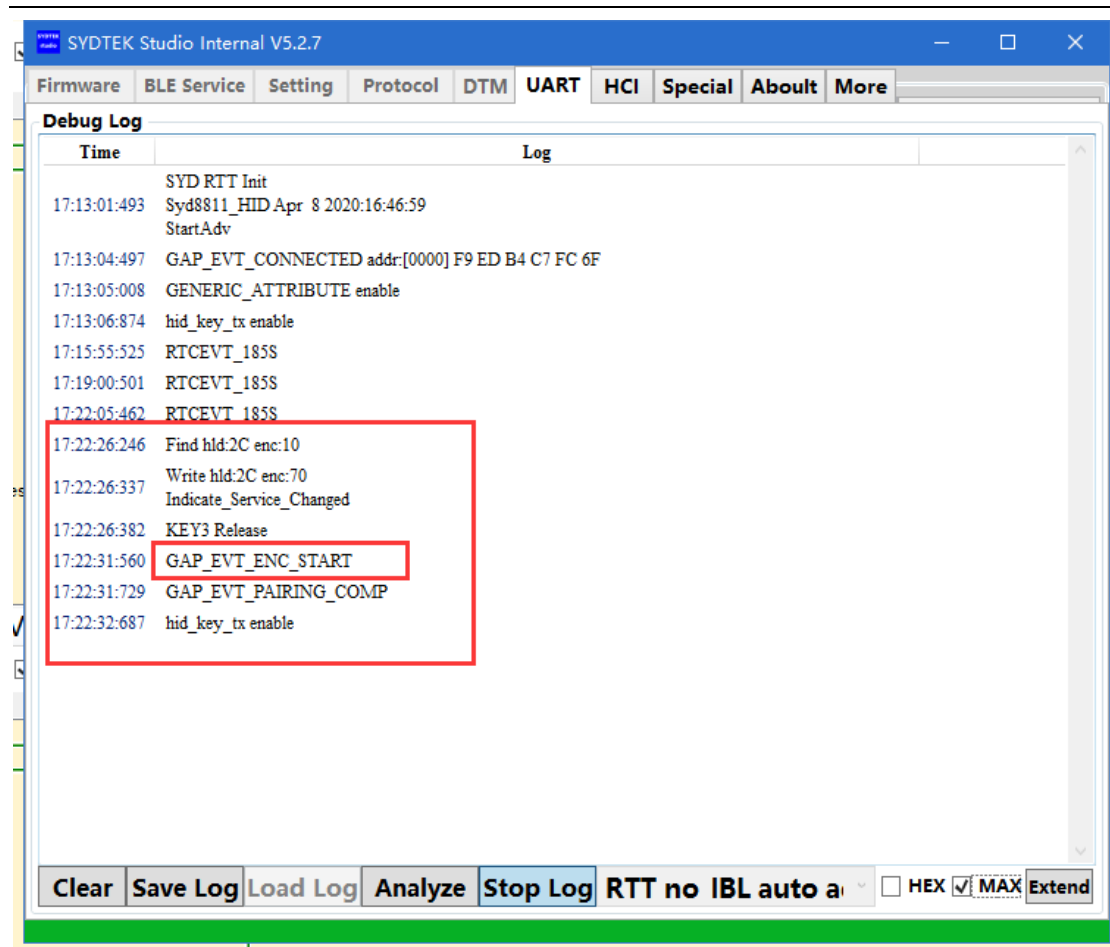
这里按下 KEY3 将会用代码把上面去掉的 Encrypt 勾选再次勾选上, 然后告诉主机, 让主机来进行加密配对:

```

858
859
860 if(GPIO_Pin_Read(U32BIT(KEY3)))
861 {
862     delay_ms(5);
863     if(GPIO_Pin_Read(U32BIT(KEY3)))
864     {
865         KEY3_Status = 1;
866         if(service_change_tx==1)
867         {
868             if((service_change_state & BIT0)==0)
869             {
870                 {
871                     uint8_t enc=0,hld=0x2c;
872                     if(APP_Read_Attribute_Encrypt(hld,&enc))
873                     {
874                         #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
875                         DBGPRINTF("Find hld:%x enc:%x\r\n",hld,enc);
876                         #endif
877                         if(enc==0x10)
878                         {
879                             enc=0x70;
880                             APP_Write_Attribute_Encrypt(hld,&enc);
881                             #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
882                             DBGPRINTF("Write hld:%x enc:%x\r\n",hld,enc);
883                             #endif
884                             if(BLE_Indicate_Service_Changed(0x19, 0xFFFF))
885                             {
886                                 service_change_state |=BIT0;
887                             }
888                             #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
889                             DBGPRINTF("Indicate_Service_Changed\r\n");
890                             #endif
891                         }
892                     }
893                     else
894                     {
895                         #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
896                         DBGPRINTF("Can not Find enc hld:%x\r\n",hld);
897                         #endif
898                     }
899                     hld=0x2b;
900                     if(APP_Read_Character_Permission(hld,&enc))
901                     {
902                         #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
903                         DBGPRINTF("Find hld:%x Permission:%x\r\n",hld,enc);
904                         #endif
905                     }
906                     else
907                     {
908                         #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
909                         DBGPRINTF("Can not Find Permission hld:%x\r\n",hld);
910                         #endif
911                     }
912                 }
913             }
914         }
915     }
916 }
917 else
918 {
919     if(KEY3_Status == 1)
920     {
921         KEY3_Status = 0;
922         #if defined(_DEBUG_) || defined(_SYD_RTT_DEBUG_)
923         DBGPRINTF("KEY3 Release\r\n");
924         #endif
925     }
926 }
927
928 if(TIMER_EVT)
929 {
930

```

可以看到 IOS 确实能够弹出配对窗口, 并且能够重新加密配对, 整个过程 log 显示如下:





注意：本文中的某些函数只有最新的 Lib 才有，这里请使用下面的代码：



Source

Code20200408 17

盛芯微 Confidential