



## Bluetooth FAQ

Version: 1.0  
Release date: 2010

© 2009 - 2011 MediaTek Inc.

This document contains information that is proprietary to MediaTek Inc.

Unauthorized reproduction or disclosure of this information in whole or in part is strictly prohibited.

Specifications are subject to change without notice.

## Document Revision History

Revision	Date	Author	Description
1.0	2010-04-01	WG Wei	Bluetooth FAQ

## Table of Contents

Document Revision History.....	2
Table of Contents.....	2
Bluetooth .....	4
1. 蓝牙的 OPP 功能和 FTP 功能中，接收或者下载到的文件的存储路径如何修改？ .....	4
2. 蓝牙“认证需求”有何作用？ .....	6
3. 蓝牙远程控制服务(HID)功能及其使用方法。 .....	7
4. 蓝牙地址相关问题.....	7
5. 设置 FTP 共享文件夹后，接收到的文件还是不能保存到该目录？ .....	9
6. 如何修改本地蓝牙设备的默认名称.....	9
7. BT RF 测试 .....	10
8. 蓝牙中“搜寻免持装置”和“我的装置”→“搜寻新装置”有何区别？ .....	11
9. 如何解决无 SIM 文件系统中无蓝牙发送菜单.....	11
10. 电话本中通过蓝牙发送记录问题 .....	13
11. 蓝牙设置菜单里面关于 FTP 设置“Read only”和“full control”有何区别....	17
12. 什么是 SAP Profile .....	17
13. 什么是 HID .....	18
14. BIP 和 BPP 有何区别 .....	18
15. 用 DUN 在 PC 上创建拨号连接后还是无法成功拨号 .....	18
16. 如何透过 BT SPP 连接传输数据 .....	19
17. 蓝牙提示“硬件忙碌中”.....	25
18. 蓝牙待机电流大 .....	26
19. 蓝牙激活不了.....	26
20. 蓝牙版本升级需要替换哪些库 .....	27
21. HFP：蓝牙耳机不能接听/挂端电话 .....	27
22. 蓝牙耳机对第三方来电的处理异常 .....	27
23. 平台上蓝牙听歌断续 .....	28
24. 蓝牙来电无提示音.....	28

25.	蓝牙耳机通话发送声音小 .....	29
26.	修改串口名称.....	29
27.	AudioPlayer 中的蓝牙 Setting 的 Bluetooth Stereo Output 开关不起作用	29
28.	HSP 需要两次按键，才能执行 AT+BLDN .....	30
29.	蓝牙耳机播放按键在接听电话后不能播放音乐 .....	31
30.	AVRCP 会随 A2DP 的断开而断开.....	31
31.	A2DP.....	32
32.	如何利用 AT+EBTLB 进行测试 Loop back 2 testing.....	32
33.	如何在蓝牙搜索耳机并匹配完成后，默认 Audio player 中蓝牙立体声耳机为开? .....	32
34.	来电，蓝牙耳机听到的音乐是噪音.....	34
35.	某些版本蓝牙为什么不能 power on ? .....	35
36.	关闭 BT 单声道播放音乐，保留来电铃声的功能： .....	35
37.	不插入 T 卡时，去掉蓝牙 Storage 菜单中的 Memory card 选项 .....	36
38.	在发送方取消文件 OPP 发送时，接收方增加一个提示.....	36
39.	如何使 BT 在开机时默认打开或者关闭.....	36
40.	如何在在用户手动断开蓝牙耳机之后，将立体声也关掉? .....	38
41.	Add_H_free_right_softkey_in_bt_call_screen.....	39
42.	Auto_Connect_HFP_Change_Feature .....	40
43.	Auto_use_a2dp_after_BT_headset_plug_in.....	40
44.	Auto_use_a2dp_when_press_PLAY_key.....	40
45.	Redial_sim2_last_dial_number_using_hfp .....	40
46.	使用默认密码自动配对 Audio device .....	40
47.	09A 上实现文件传输时蓝牙图标闪烁的方法.....	40
48.	09B 上实现可以使用蓝牙耳机重拨 SIM2 电话的方案 .....	40
49.	PC 使用 FTP 推送文件到手机时，存储位置异常的解决方案 .....	40
50.	恢复出厂设置时，实现蓝牙的 Factory Restore 功能 .....	40
51.	若手机剩余存储空间不够，Popup 提示请插入 T 卡，并结束本次接收文件过程 (09A).....	43
52.	MTK 手机给 Android 手机发送文件，Android 手机确认接收前发送方进度不为 0%.....	44
53.	搜索过程无 pair 选项，必须先选项 cancel 才能进入下一步配对.....	45
54.	蓝牙接收文件存储位置设置，有 T 卡存 T 卡，无 T 卡存手机(09A).....	47
<b>Bluetooth Reference Design .....</b>		<b>48</b>
1.	蓝牙耳机单声道听歌，进搜索后，马上发送文件，导致声音从手机端播放 (09A) .....	48
2.	发送文件，选择新设备->搜索设备->配对成功后，需要继续在配对列表中选择新配对的设备才能发送 .....	51

## Bluetooth

---

1. 蓝牙的 OPP 功能和 FTP 功能中，接收或者下载到的文件的存储路径如何修改？
- 

**[AUTHOR]**

MSZ00137/MSZ00289

**[DESCRIPTION]**

**[SOLUTION]**

**CSR Solution**

OPP 接收文件的存储路径

```
#define MMI_OPP_MAKE_RECV_FILENAME(x, d) \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\%s\\%s%02d.%s", \
        MMI_PUBLIC_DRV, MMI_OPP_RECV_OBJ_FILEPATH, \
        MMI_OPP_RECV_OBJ_FILENAME, d, MMI_OPP_RECV_OBJ_FILEEXT );
```

```
#define MMI_OPP_MAKE_RECV_PATH(x) \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\%s\\", \
        MMI_PUBLIC_DRV, MMI_OPP_RECV_OBJ_FILEPATH );
```

```
#define MMI_OPP_MAKE_RECV_STORE_NAME(x, y) \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\%s\\%s", \
        MMI_PUBLIC_DRV, MMI_OPP_RECV_OBJ_FILEPATH, y );
```

FTP 下载文件的存储路径

```
#define MMI_FTP_MAKE_RECV_FILENAME(x, d) \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\%s\\%s%02d.%s", \
        MMI_PUBLIC_DRV, MMI_FTP_RECV_OBJ_FILEPATH, \
        MMI_FTP_RECV_OBJ_FILENAME, d, MMI_FTP_RECV_OBJ_FILEEXT );
```

```
#define MMI_FTP_MAKE_RECV_PATH(x) \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\%s\\", \
        MMI_PUBLIC_DRV, MMI_FTP_RECV_OBJ_FILEPATH );
```

#### MT6601 Solution

OPP 接收文件的存储路径

```
#define VOBJ_DRV      MMI_PUBLIC_DRV
```

```
#define MMI_OPP_MAKE_RECV_FILENAME(x, d) do{ \
    U8 temp_name[32]; \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\", MMI_PUBLIC_DRV); \
    UCS2Strcat((PS8) x, (PS8) MMI_OPP_RECV_OBJ_FILEPATH); \
    kal_wsprintf((kal_uint16 *)temp_name, "%s%02d.%s", \
        MMI_OPP_RECV_OBJ_FILENAME, d, MMI_OPP_RECV_OBJ_FILEEXT); \
    UCS2Strcat((PS8) x, (PS8)temp_name); \
}while(0)
```

```
#define MMI_OPP_MAKE_RECV_PATH(x) do{ \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\", MMI_PUBLIC_DRV); \
```

```
UCS2Strcat((PS8) x, (PS8) MMI_OPP_RECV_OBJ_FILEPATH); \
}while(0)
```

```
#define MMI_OPP_MAKE_RECV_STORE_NAME(x, y) do{ \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\", MMI_PUBLIC_DRV); \
    UCS2Strcat((PS8) x, (PS8) MMI_OPP_RECV_OBJ_FILEPATH); \
    UCS2Strcat((PS8) x, (PS8) y); \
}while(0)
```

FTP 下载文件的存储路径

```
#define FTPEC_MAKE_RECV_PATH(x) do{ \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\", MMI_PUBLIC_DRV); \
    UCS2Strcat((PS8)x, (PS8)FTPEC_RECV_OBJ_FILEPATH); \
}while(0)
```

```
#define MMI_FTP_MAKE_RECV_TMP_OBJNAME(x, d) do{ \
    U8 temp_name[32]; \
    kal_wsprintf( (kal_uint16 *)x, "%c:\\", MMI_PUBLIC_DRV); \
    UCS2Strcat((PS8)x, (PS8)FTPEC_RECV_OBJ_FILEPATH); \
    kal_wsprintf( (kal_uint16 *)temp_name, "%s%02d.%s", \
        MMI_FTP_RECV_TMP_OBJNAME, d, MMI_FTP_RECV_TMP_OBJEXT ); \
    UCS2Strcat((PS8)x, (PS8)temp_name); \
}while(0)
```

## 2. 蓝牙“认证需求”有何作用？

---

### [AUTHOR]

MSZ00409

### [DESCRIPTION]

### [SOLUTION]

蓝牙的安全机制如下：

PIN（用户输入）--> Link Key（用于蓝牙互连） --> Encryption Key（用于上层 Profile 服务加密的需要）。

PIN 码的主要作用体现在用于生成 Link Key 上：蓝牙设备间要能进行蓝牙服务，首先得互连，而 Link Key 则用来为保证蓝牙设备间相互允许建立连接。

认证的作用主要体现在 Encryption Key 上。认证关闭后，就不会生成 Encryption Key；认证开启后，就会有生成 Encryption Key 的功能。Encryption Key 在有些上层 Application 是需要的，如两台手机通过蓝牙互联后，需要传送电子卡的这类需要更高保密要求的信息，那么我

们就需要在 Link Key 的基础上再生成 Encryption Key。在采用 FTP 或 OPP 服务时，也可以采用开启认证来提高安全性。

补充：认证跟输入 PIN 码还不一样，PIN 码是蓝牙安全机制的基础，开/关闭“认证”并不影响 PIN 码，就是认证关闭了，PIN 同样还是需要输入。

### 3. 蓝牙远程控制服务(HID)功能及其使用方法。

---

#### [AUTHOR]

MSZ00489

#### [DESCRIPTION]

#### [SOLUTION]

手机上的蓝牙远程控制，就是指用蓝牙手机远程遥控其它蓝牙设备的的功能。要使用这个功能，需要做以下几个步骤：

1. 在你的电脑上装上PC远程控制服务软件(ex.BlueSoleil)和蓝牙适配器。
2. 通过BlueSoleil中的蓝牙人机输入设备服务连接上手机。这时手机上会出现三个菜单:桌面，多媒体播放器，幻灯片。
3. 进入三个菜单中的任意一个，通过手机的方向和数字键就可以控制电脑的动作了。
  - 一，进入桌面菜单能够实现桌面控制，像控制鼠标的移动，单击，双击。
  - 二，进入播放器菜单对媒体播放器进行控制(停止，播放，快进，后退等控制)，(前提是打开多媒体播放器ex,windowmedia)
  - 三，进入幻灯片，控制PPT页面(上一页，下一页等的控制)。

### 4. 蓝牙地址相关问题

---

#### [AUTHOR]

MSZ00409

#### [DESCRIPTION]

对蓝牙地址的作用及使用，我们常会有一些疑惑，下面是对其常见问题的汇总。

#### [SOLUTION]

蓝牙地址长度为6bytes， 每台蓝牙设备都应有一个唯一的蓝牙地址。目前，蓝牙地址的使用需要向IEEE申请授权使用。

#### 1. 蓝牙芯片的地址怎样获取？

蓝牙地址保存在  
NVRAM\_EF\_BTRADIO\_MT6601\_LID/NVRAM\_EF\_BTRADIO\_MT6611\_LID中。当蓝牙  
Poweron后，NVRAM中的蓝牙地址会读入到蓝牙芯片当中，这样蓝牙芯片就获取了蓝牙地址。

#### 2. xxxxxxxx6601这样的蓝牙地址有可能出现重复

首先需要声明的是xxxxxxxx6601这种默认生成的蓝牙地址是MTK内部测试使用。

蓝牙地址正确的获取途径是向IEEE申请授权使用, 在下载新的BIN档后, 可以通过相应的工具将申请蓝牙地址写入。

但是由于众多厂家没有申请蓝牙地址, 或者还申请了但还没有正式写入, 针对这种情况, 我们会给出一个默认的地址, 方便内部测试。默认地址具体生成的过程是这样: 在第一次蓝牙PowerOn的时候, 如果判断NVRAM中的蓝牙地址的前2个bytes为0x01, 0x66或0x11, 0x66, 系统会认为厂家并没有写入自己的蓝牙地址, 这时候系统会将蓝牙地址中剩下的4bytes用一组随机数来代替。

这样我们会在Bluetooth->about下看到这样的蓝牙地址xxxxxxxx6601或xxxxxxxx6611. 由于随机生成, 还是有可能出现地址重复。

### 3. 蓝牙地址反序

输入的蓝牙地址会以与输入相反的顺序显示, 常称反序, 这不是系统的BUG. 蓝牙地址的显示是遵循Bluetooth的规范 (GAP Profile Section 3.2.1)。蓝牙地址6bytes, 分为LAP, UAP, NAP三部分, 输入时, 是从LSB->MSB(LAP UAP NAP), 而显示时则为MSB->LSB(NAP UAP LAP). 如输入1234567890ab, 显示则为ab:90:78:56:34:12. 如果需要将蓝牙地址以与输入方式一致的方式显示, 可以参考mmi\_bt\_entry\_about函数中蓝牙地址的显示方式 (NAP, UAP, LAP), 修改为 (LAP, UAP, NAP)。

### 4. 蓝牙地址的作用?

蓝牙地址是整个蓝牙系统的基础, 它在很多地方会应用到。蓝牙设备间在连接时, 蓝牙地址会与输入的PIN组和一起生成Link key; Master的蓝牙地址会用来决定整个Piconet的Physical Channel和Hopping Sequence; 在每个Packet前面加入的Access Code, 也是来源于蓝牙地址LAP部分。

### 5. 为什么在Meta工具中, 使用Mediatek BT Tool修改蓝牙地址后, 重启开机, 进入Bluetooth->About查看, 蓝牙地址没有改变?

a). 在手机的Bluetooth->About 里的蓝牙地址是保存在NVRAM中。蓝牙PowerOn, 蓝牙芯片的地址会从NVRAM中对应的LID获取。

b). Meta的Mediatek BT Tool写蓝牙地址不是写到NVRAM, 而是直接通过HCI命令写到蓝牙芯片。这样做是临时改变蓝牙地址。目的是方便蓝牙芯片测试。因为如果该NVRAM, 要使蓝牙地址生效, 必须是改之后, 去Power OFF BT, 下次PowerOn的时候又从NVRAM中去读取, 才是改之后的地址, 这样太复杂。

因此通过Mediatek BT Tool不能永久修改蓝牙地址。

### 6. 怎样永久修改蓝牙地址?

目前有以下方法:

a) AT+EMBT: 具体使用方法请参考AT\_Command\_Full.pdf。



ex.

- 请用如下AT命令实现蓝牙地址的修改:
- AT+EMBT=1 //poweron bt chip
- ok
- AT+EMBT=0,bt,1234567890ab //写地址
- ok
- AT+EMBT=2 //退出测试模式
- ok
- AT+EMBT=5 //读取地址 0xab9078563421

b) SN Station Tool(For Production Line). 怎样使用请参考User Manu。

## 5. 设置 FTP 共享文件夹后，接收到的文件还是不能保存到该目录？

---

### [AUTHOR]

MSZ00409

### [DESCRIPTION]

### [SOLUTION]

由于我们手机一般既有FTP Server又有FTP Client端的功能, 因此在进行相关设定的时候, 容易混淆.

蓝牙->设置->文件传输设置->分享目录, 是针对FTP Server端的设置, 并且这样的设置并不能改变接收文件的存储路径。

因为该设置是针对FTP Server端的设置, 改变该目录, 会使得用户在FTP Client端所浏览到的文件夹也就不同。

同时, 通过该设置, 可以使得Server针对不同的Client设置不同的浏览文件夹。

当我们的手机从别的蓝牙设备上PULL(下载)文件时, 手机这时候扮演的是Client端的角色. 此时, 若要修改接受文件的路径, 是针对FTP Client端的设置。

当FTP Client端从Server端PULL(下载)文件时, 会有一个默认的存储路径, 若需要改变文件存储的路径, 则需针对FTP Client端的设置做出修改。

如果要修改文件存储路径, 可以参考btmmiftpgprots.h中的FTPC\_MAKE\_RECV\_PATH和MMI\_FTP\_MAKE\_RECV\_TMP\_OBJNAME (07A) 或MMI\_FTP\_MAKE\_RECV\_TMP\_OBJNAME ((07B.08.16及之后的版本。))

注: 在07B.W08.16之后, Client端的文件存储路径可以选择存在Phone还是Memory Card(Bluetooth->Setting->Storage)。

## 6. 如何修改本地蓝牙设备的默认名称

---

#### [AUTHOR]

MSZ00489

#### [DESCRIPTION]

如何修改本地蓝牙设备的默认名称

#### [SOLUTION]

BTMMICM.c 中修改数组如下:

```
const U8 MMI_BT_DEF_HOST_BT_NAME[] = { 'M', 'T', 'K', 'B', 'T', 'D', 'E', 'V', 'I', 'C', 'E',  
'\0' };
```

## 7. BT RF 测试

---

### ATE操作

在连接好测试仪器和完成相关文件设置后，打开 ATE，按 Initialize Final Test 按钮初始化蓝牙测试仪器，初始化完毕后按 RF Final Test 按钮即可开始蓝牙手机测试

### 测试项目

在 Customer\_Setup.txt 中有列举出蓝牙手机 RF 测试的项目和指标的允许范围值:

BT DL POWER = -80.0

MAX OUTPUT POWER = 4.0

MIN OUTPUT POWER = -6.0

FREQUENCY ACCURACY = 75.0

DRIFT RATE DH1 = 25.0

DRIFT RATE DH3 = 40.0

DRIFT RATE DH5 = 40.0

DF2M = 115.0

DF1A MAX = 175.0

DF1A MIN = 140.0

MCH RATE = 0.8

BER = 0.1



```

#ifdef __MMI_OPP_SUPPORT__ //MTK add this line begin
    MENU_FMGR_FWD_BT,
#endif //MTK add this line end
    MENU_ITEM_END,
    SHOW, MOVEABLEWITHINPARENT|INSERTABLE, DISP_LIST,
STR_FMGR_FORWARD,0));

```

(4)第四处修改地方

```

    ADD_APPLICATION_MENUITEM((MENU_FMGR_FWD_OPTION_VCARD_NO_SI
M,MITEM_FMGR_GEN_TYPE_OPTIONS,
    2, //MTK modify this line from value 1 to 2
    MENU_FMGR_FWD_PHB,
#ifdef __MMI_OPP_SUPPORT__ //MTK add this line begin
    MENU_FMGR_FWD_BT,
#endif //MTK add this line end
    SHOW, MOVEABLEWITHINPARENT|INSERTABLE, DISP_LIST,
STR_FMGR_FORWARD,0));

```

(5)第五处修改地方

```

    ADD_APPLICATION_MENUITEM((MENU_FMGR_FWD_OPTION_VCAL_NO_SIM,
MITEM_FMGR_GEN_TYPE_OPTIONS,
    2, //MTK modify this line from value 1 to 2
    MENU_FMGR_FWD_TODOLIST,
#ifdef __MMI_OPP_SUPPORT__ //MTK add this line begin
    MENU_FMGR_FWD_BT,
#endif //MTK add this line end
    .....
));

```

(6) 第六处修改地方:

```

    ADD_APPLICATION_MENUITEM2((MITEM_FMGR_IMG_TYPE_NO_SIM_OPTION
S,0,
    MENU_FMGR_IMG_OPTION_VIEW,
    MENU_FMGR_GEN_OPTION_FORWARD, //MTK add this line

```

```
.....  
));
```

2.在 FileMgr.c 中的 fmgr\_get\_forward\_option\_menu 函数处，  
此函数共修改一处地方

```
/* no SIM or flight mode can't forward out */  
if (!mmi_bootup_is_network_service_available()) == MMI_FALSE //MTK change this  
line  
#ifdef __FLIGHT_MODE_SUPPORT__  
    mmi_bootup_get_active_flight_mode() != 0  
#endif  
    )  
    {  
        .....  
    }
```

## 10. 电话本中通过蓝牙发送记录问题

### [AUTHOR]

Mtk70489

### [DESCRIPTION]

1.在电话簿中选择某一条电话记录，选择通过蓝牙发送到 PC，会以同一个名字命名  
~vcard.vcf

2.在把日历或者待办事项通过蓝牙发送到 PC，被保存后，都以同样的一个~vcalendar.vcs  
名字命名。

期望效果：

如果是电话簿记录发送名片，则以电话记录中的姓名为文件名保存文件。

如果是备忘录，则以备注名保存。

### [SOLUTION]

(1)vcard 发送文件到 PC 文件名固定的问题，修改步骤如下：

vCard.c 中：

step1.

```
static void mmi_vcard_forward_bt(void)
```

```
{
```

```
.....
```

```
/* Create file path and folder. */
```

```
if (check_result == FS_NO_ERROR)
```

```

{
    path = OsIMalloc(VOBJ_MAX_PATH_LENGTH);
    //mmi_vcard_util_get_default_file(path); //MTK delete this line
    mmi_vcard_util_get_default_file_name(path); //MTK add this line
    .....
}

```

step2.

```

static void mmi_vcard_forward_bt_callback(void *buf_p, U32 result)
{
    .....
    //mmi_vcard_util_del_default_file; //MTK delete this line
    mmi_vcard_util_del_default_file_name(); //MTK add this line
    .....
}

```

step3.

在 vCard.c 文件中增加函数:

```

void mmi_vcard_util_del_default_file_name(void) //MTK add this function
{
    S8 *path;
    path = OsIMalloc(VOBJ_MAX_PATH_LENGTH);
    mmi_vcard_util_get_default_file_name(path);
    FS_Delete((U16*) path);
    OsIMfree(path);
}

```

step4.

vCard.h 中增加声明:

```

extern void mmi_vcard_util_get_default_file_name(S8 *path); //MTK add this line

```

(2).vcalendar 发送文件到 PC 文件名固定的问题, 修改步骤如下:

vCalendar.c 中, 添加函数如下:

step1.

```

void mmi_vclndr_util_get_default_filename(S8 *path) //MTK add this function
{

```

```

/*-----*/
/* Local Variables                                     */
/*-----*/

S8 *ascii_path;
S8 *file_name;
S8 filename[VOBJ_MAX_FILE_EXT_LEN];
S16 pError;
U16 name_length;
U16 path_length;
S8* vclndr_temp_filename;
S8* temp_ptr;
U16 vclndr_temp_filename_length = (MAX_TODO_NOTE_LEN + 1) *
ENCODING_LENGTH;

/*-----*/
/* Code Body                                           */
/*-----*/

ascii_path = OsiMalloc(VOBJ_MAX_PATH_LENGTH / ENCODING_LENGTH);
sprintf(ascii_path, "%c:\\%s\\", VOBJ_DRV, VOBJ_FILE_PATH);
AnsiiToUnicodeString(path, ascii_path);
path_length = UCS2Strlen(path);
OsiMfree(ascii_path);

file_name = OsiMalloc(VOBJ_MAX_FILENAME_LENGTH);
if (VOBJ_MAX_FILENAME_LENGTH > (MAX_TODO_NOTE_LEN + 1) *
ENCODING_LENGTH)
{
    name_length = (MAX_TODO_NOTE_LEN + 1) * ENCODING_LENGTH;
}
else
{
    name_length = VOBJ_MAX_FILENAME_LENGTH;
}

vclndr_temp_filename = OsiMalloc(vclndr_temp_filename_length);

memcpy(
    vclndr_temp_filename,

```

```

(&g_tdl_cntx.TaskList[g_tdl_cntx.CurrTaskIndex])->Note,
vclndr_temp_filename_length);

temp_ptr = mmi_fmgr_skip_space(vclndr_temp_filename);
UCS2Strcpy(file_name, temp_ptr);
OslMfree(vclndr_temp_filename);

if (!mmi_fmgr_is_filename_valid(file_name))
{
    AnsiiToUnicodeString(file_name, "~vcalendar.vcs");
    UCS2Strcat(path, file_name);
}
else
{
    if ((path_length + name_length/ENCODING_LENGTH +
VOBJ_MAX_FILE_EXT_LEN/ENCODING_LENGTH) >
VOBJ_MAX_PATH_LENGTH/ENCODING_LENGTH)
    {
        UCS2Strncat(path, file_name,
(VOBJ_MAX_PATH_LENGTH/ENCODING_LENGTH - path_length -
VOBJ_MAX_FILE_EXT_LEN/ENCODING_LENGTH));
    }
    else
    {
        UCS2Strcat(path, file_name);
    }
}
OslMfree(file_name);
AnsiiToUnicodeString(filename, VOBJ_VCLNDR_FILE_EXT);
UCS2Strcat(path, filename);
}

```

step2.

void mmi\_vclndr\_uti\_del\_default\_file(void) //MTK add this function

```

{
    /*-----*/
    /* Local Variables */
}

```



```

/*-----*/
S8 *path;

/*-----*/
/* Code Body */
/*-----*/

path = OsIMalloc(VOBJ_MAX_PATH_LENGTH);
mmi_vclndr_uti_get_default_filename(path);
FS_Delete((U16*) path);
OsIMfree(path);

}

```

step3.

vCalendar.h 中增加如下声明:

```
extern void mmi_vclndr_uti_get_default_filename(S8 *path); //MTK add this line
```

## 11. 蓝牙设置菜单里面关于 FTP 设置“Read only”和“full control”有何区别

---

### [AUTHOR]

MSZ00289

### [DESCRIPTION]

蓝牙设置菜单里面关于 FTP 设置“Read only”和“full control”有何区别

### [SOLUTION]

“Read Only”和“full control”的区别主要体现在其它设备透过 FTP 连接到 Server 后的权限。

- Read Only: 只能读取 FTP Server 上 share folder 里面的文件
- Full control: 不仅能够读取 FTP Server 上的文件, 还可以对 share folder 执行 写入, 删除, 创建 folder 等操作。

## 12. 什么是 SAP Profile

---

### [AUTHOR]

MSZ00289

### [DESCRIPTION]

### [SOLUTION]

SAP 全称 SIM Access Profile

可以通过连接 SAP Profile 获取 SAP Server SIM 卡上的一些信息，如 Phonebook，SMS 等。

SAP 目前主要应用在 Carkit 上面，可以通过 Carkit 连接手机的 SAP Profile，查找电话本，拨打电话

### 13. 什么是 HID

---

**[AUTHOR]**

MSZ00289

**[DESCRIPTION]**

**[SOLUTION]**

HID Profile 主要应用场景：

PC Side dongle 连接支持 HID 的手机，1) 可以将手机作为鼠标控制 PC；2) 将手机作为控制器控制 Windows Media Player（暂停，快进等）；3) 将手机作为 Presentor 控制 PPT 播放

以上 3 个功能是目前 MTK 已经实现了，实际上 HID 还定义了很多其它功能，如 游戏手柄等

### 14. BIP 和 BPP 有何区别

---

**[AUTHOR]**

MSZ00289

**[DESCRIPTION]**

**[SOLUTION]**

BIP 的操作对象主要是 Image，两台设备可以通过 BIP 连接共享图片。

BPP 是蓝牙打印服务，可以通过 BPP 打印 Vcard/Vcalendor/Image 等

一般蓝牙打印机可能会同时支持 BIP/BPP/OPP，因此，理论上通过上述 3 个 Profile 都可以实现打印功能

- OPP，仅仅是将 Object（如 Vcard）丢给打印机，由打印机进行打印，发送端无法控制打印属性

- BIP，只能传 Image 给打印机打印，同样不能控制打印属性

- BPP，蓝牙打印服务，可以对打印属性进行控制，如打印多少张，打印大小等。

### 15. 用 DUN 在 PC 上创建拨号连接后还是无法成功拨号

---

**[AUTHOR]**

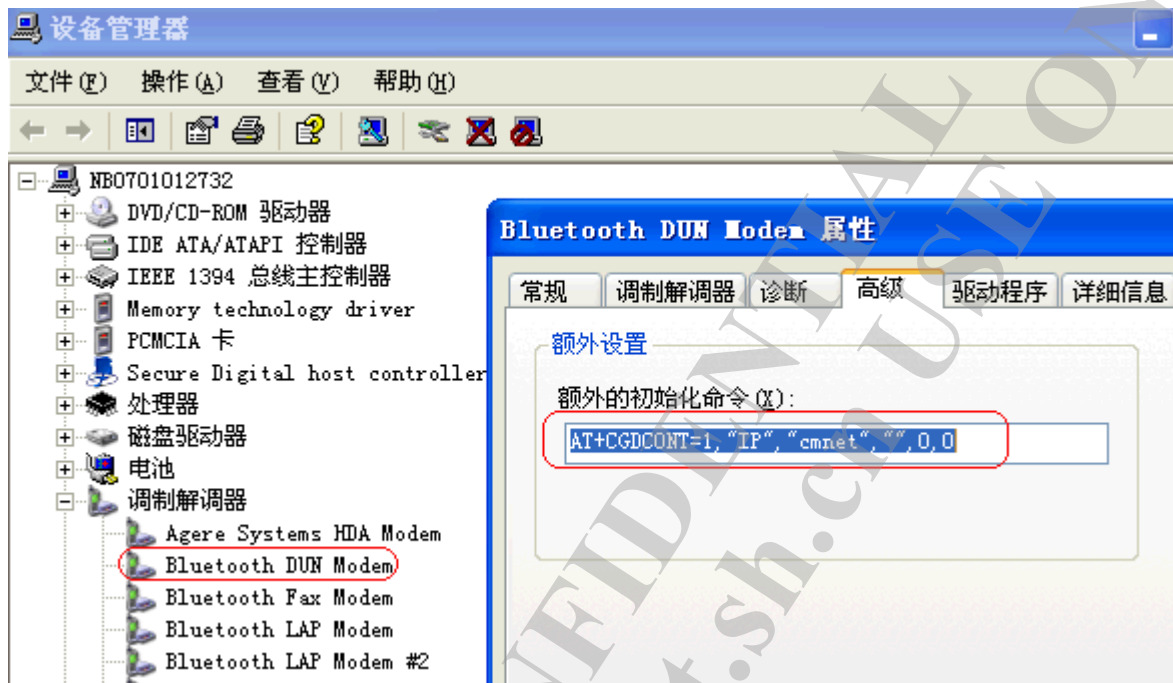
MSZ00289

**[DESCRIPTION]**

**[SOLUTION]**

1. 先确定 SIM 卡是否支持拨号上网(可以咨询运营商)

2. 若支持，则进入 Windows 设备管理器选择对应的 DUN Modem，进入“属性”--->“高级”，添加如下 AT Command



## 16. 如何透过 BT SPP 连接传输数据

**[AUTHOR]**

MSZ00289

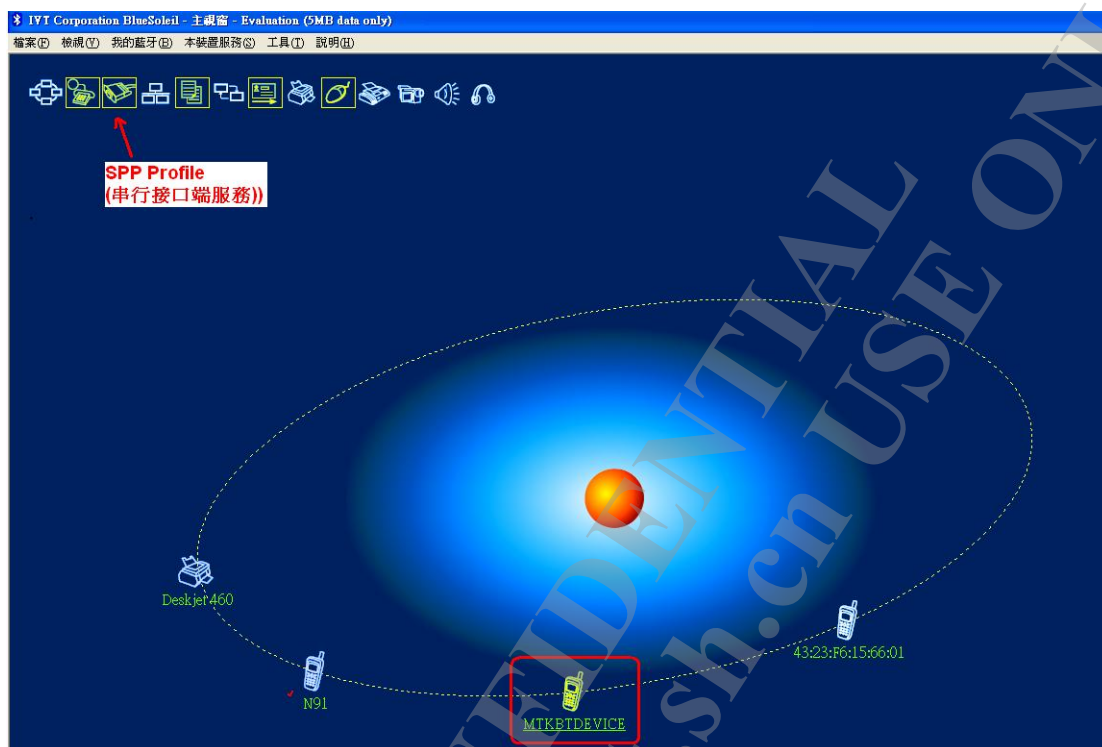
**[DESCRIPTION]**

**[SOLUTION]**

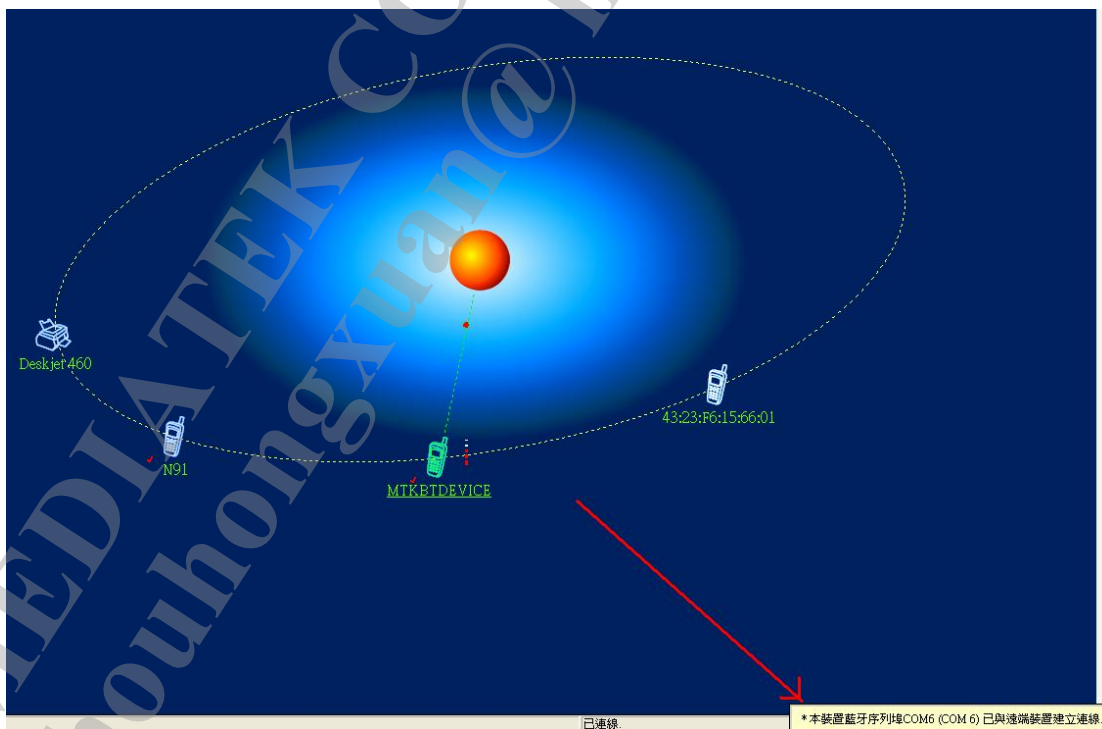
1. Use scenario : BT GPS, 蓝牙子母机, 多方蓝牙游戏
2. 参考文档:
  - a) SPP\_Client\_Feature\_for\_Customer\_MPW0732.doc
  - b) SPP\_Server\_Feature\_for\_Customer\_MPW0732.doc
3. SPP used for phonesuite:

[How to use the SPP Profile Service](#)

0. In your PC, install the Bluetooth Software and plug the Bluetooth dongle.
1. Search the Bluetooth Devices and its SPP service



2. Press the icon (SPP Profile) in IVT Window, then a COM Port will be generated when the connection is established



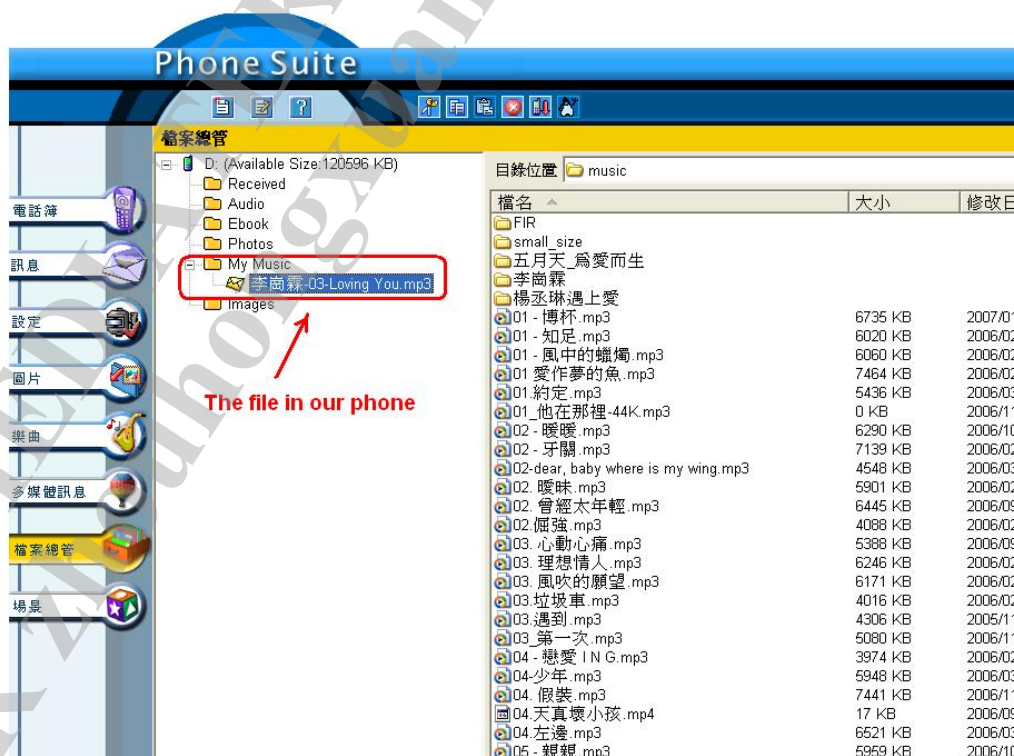
3. Open one application (e.g., PhoneSuite) to connect this Mobile Phone.  
From the Setting Panel:

Choose the correct platform and com port of the mobile phone which you want to connect.

If the “Already Connected” message is printed on window as the figure shown, it means that the connection is setup successfully.

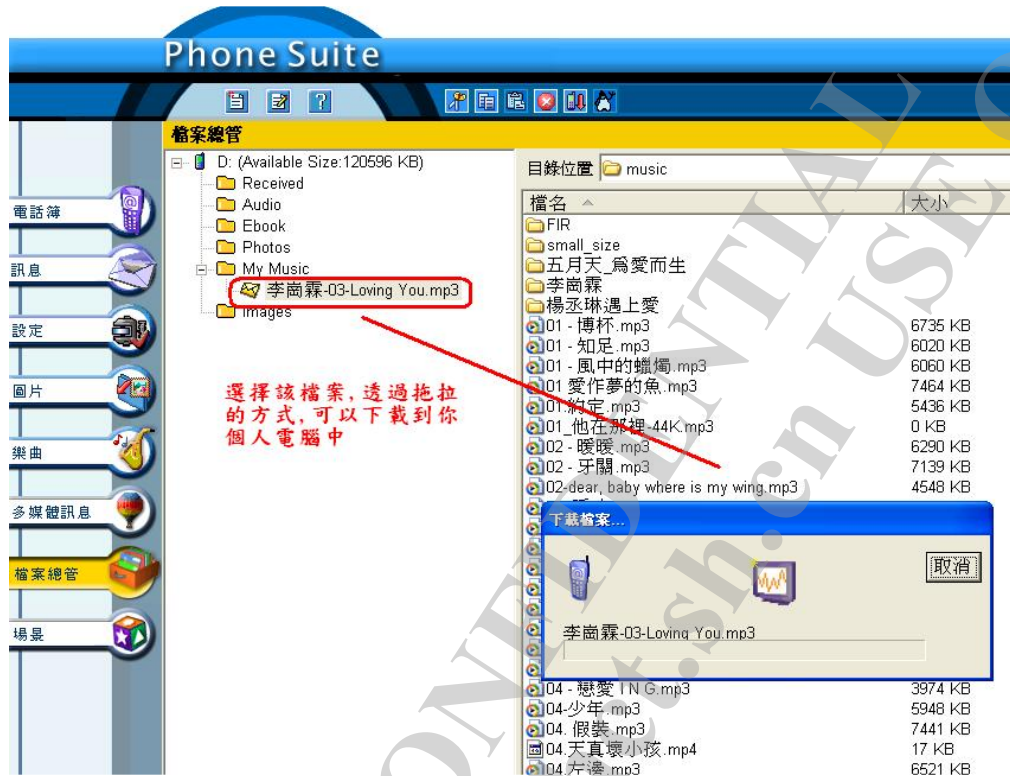


4. From File Manager, you can see files in our phone

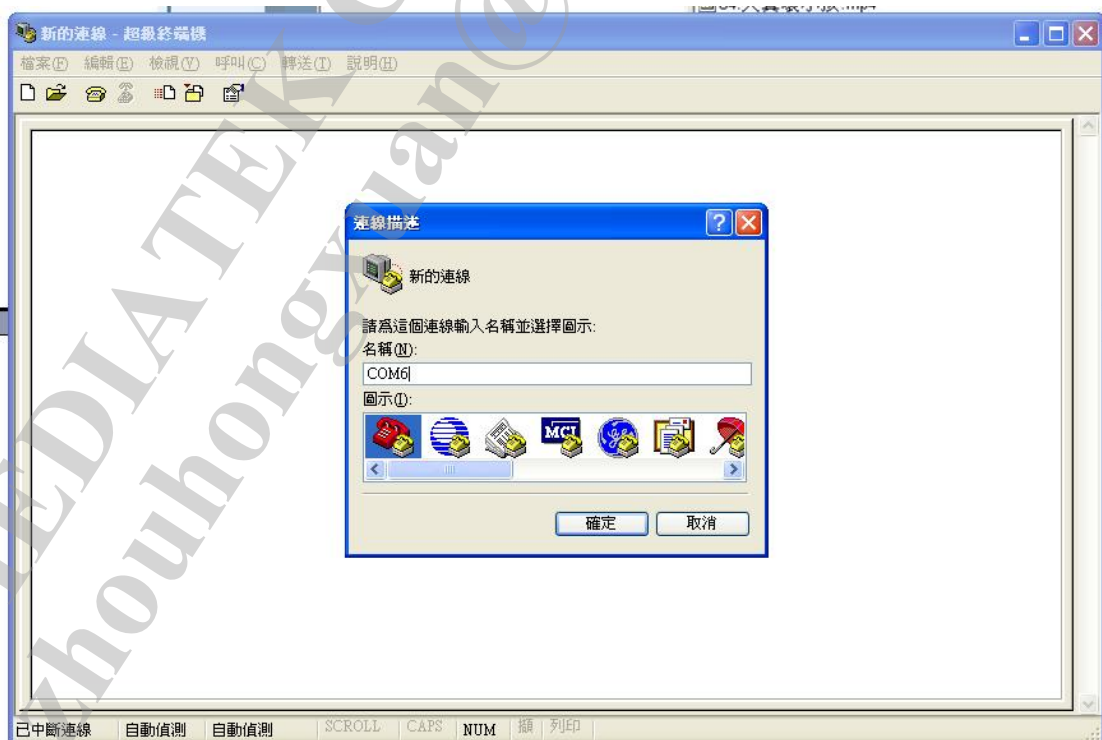




5. You can drag this file to your PC (i.e., Download this file)

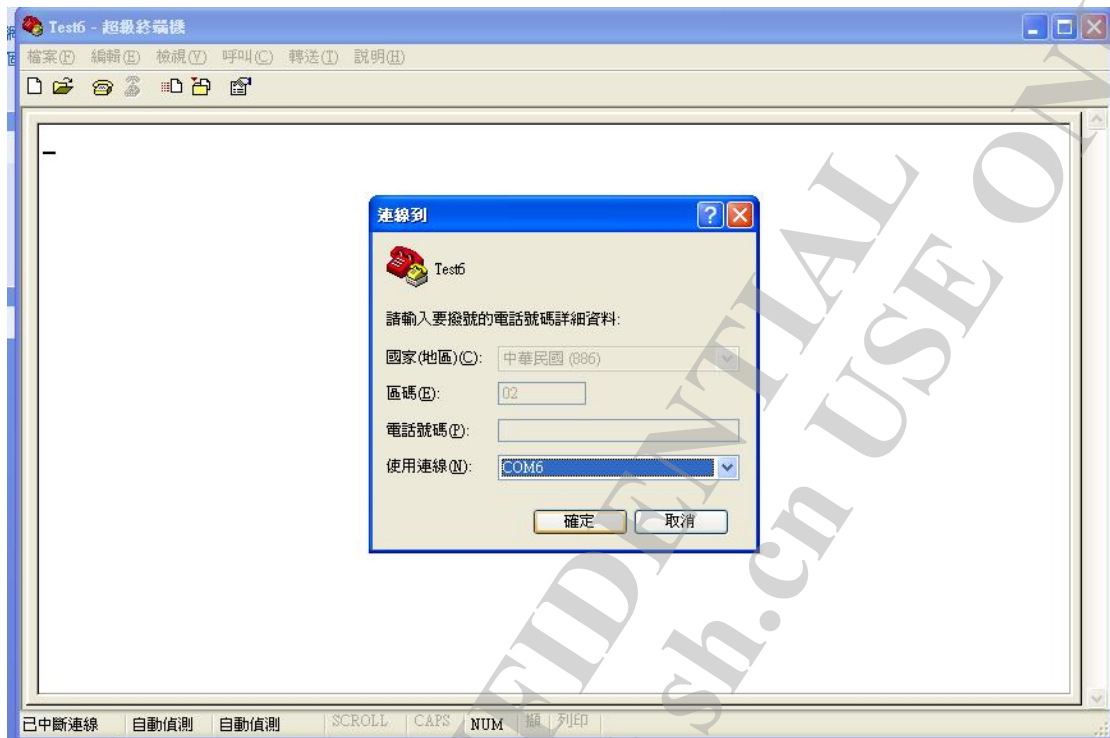


6. Open HyperTerminal (This is another application to use SPP Profile)



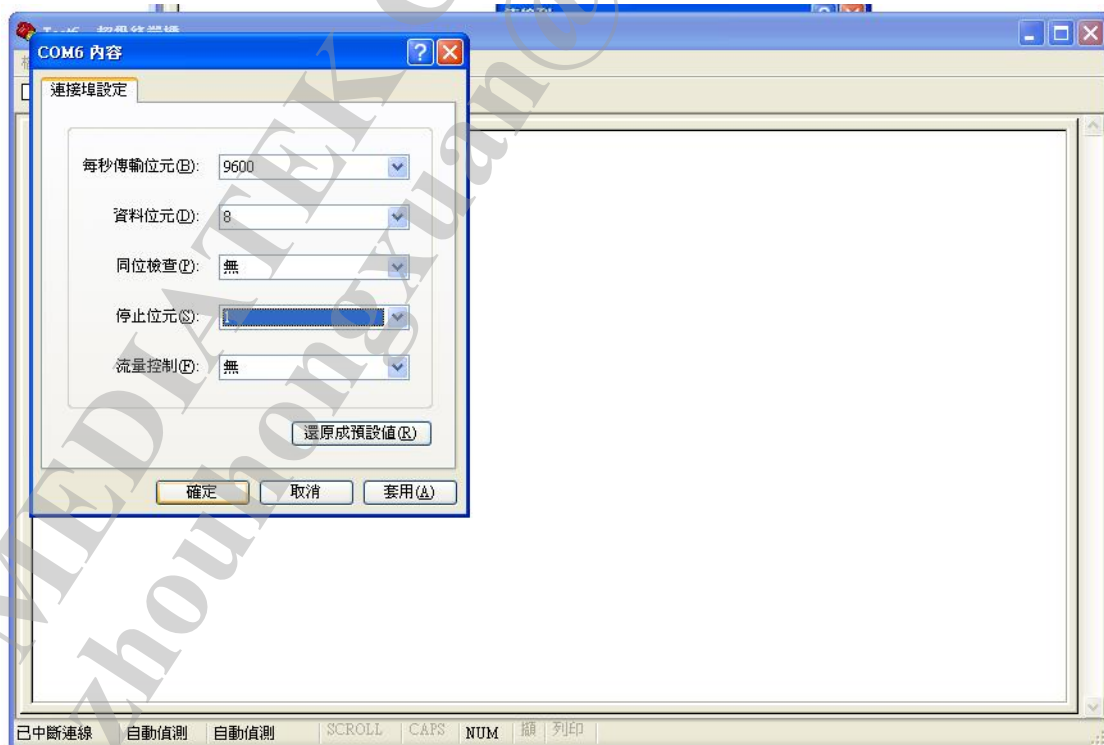
7. Setting the configuration of this com port

8-1. Choose the correct com port

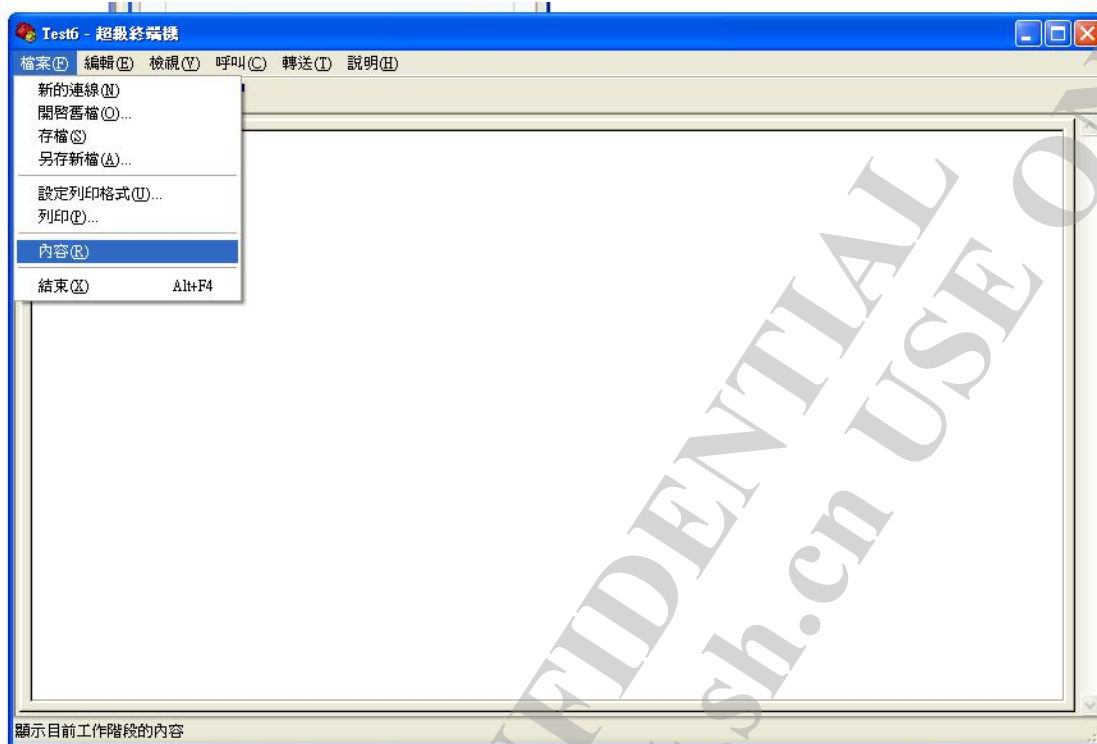


8-2.

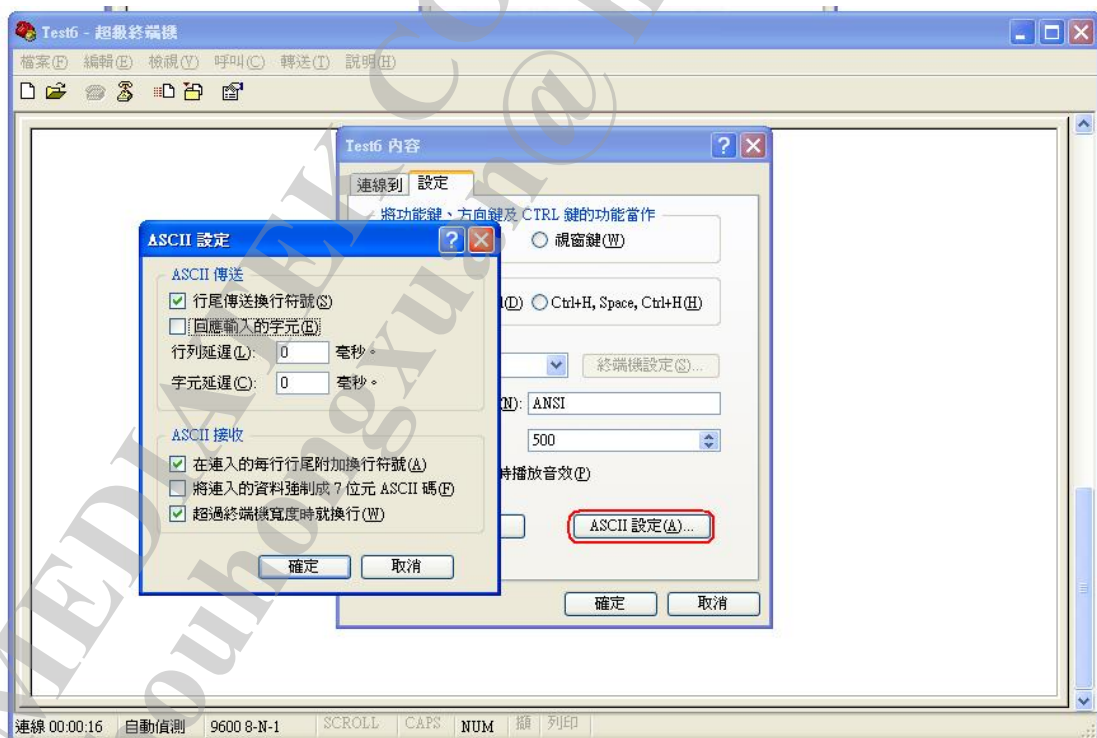
Configure the setting for this com port



8-3. Configure the setting about ASCII

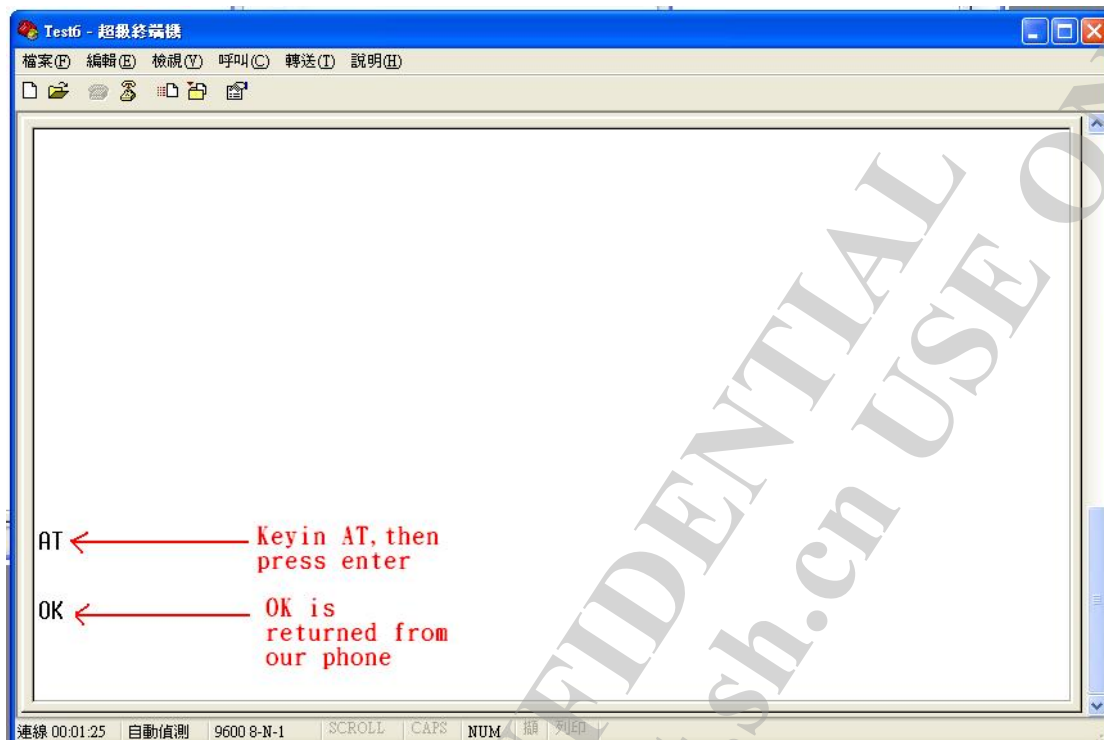


8-4.



8-5. Input AT command to test this SPP connection





## 17. 蓝牙提示“硬件忙碌中”

### [AUTHOR]

MSZ00489

### [Description]

打开蓝牙的情况下, 进行如下动作会 popup “硬件忙碌中”

1. 进入某些应用程序, ex. Entry camera Application, plug in earphone.
2. 蓝牙搜索设备或传输文件时.
3. 连接蓝牙耳机, 有电话进来.
4. 其它异常情况.

### [Solution]

出现“硬件忙碌中”, 是因为 UART3 RX 收到的数据包有错误, 导致 BB 蓝牙 Host 端复位产生的。

对于 1. 一般是应用程序运行的过程中操作到蓝牙的 gpio 导致的。如复用 power pin, reset pin, uart3 RX/TX pin. Ex. 进入 camera 时, 把电源的 PMIC/vmc 关闭, 导致蓝牙掉电, uart3 RX 收到的数据错误, 发生 BB bluetooth host 复位。或者应用程序把 gpio\_bt\_reset\_pin 拉低(意外), 均会出现。解决办法: 在 GPIO\_WriteIO, GPIO\_ModeSetup 加 trace, 把操作到的 gpio pin 打印出来, 看进入该应用程序后, 是否有操作到蓝牙的 gpio。

对于 2. 一般是低概率事件, 可能是 uart3 RX 被其它并行线干扰到, 需要检查 PCB 布线。Ex. LCD interface 与 uart3 平行布线。

对于 3.一般是低概率事件, GSM 干扰到 uart3 RX,需要检查 PCB 布线.可以把 GSM 功率调小或把 GSM 天线远离蓝牙芯片做测试, 看是否有改善来,确定是否是干扰的问题,如果是就需要改版。

对于 4,可能是其它情况导致的复位, 需要提供 catcher log 给我们分析, filter 设置如下  
MOD\_BT (All classes on)

## 18. 蓝牙待机电流大

---

### [Description]

蓝牙在如下情况下待机电流异常(平均电流, 无 SIM 测试结果).

1.连接上蓝牙耳机,不进行其它任何蓝牙操作(ex,听音乐, 打电话,传送文件),一直出现 20mA 左右待机电流.

2.开启蓝牙待机电流增大 2~6mA.

### [Solution]

对于 1.可能有两种情况:

BT chip 不能进入 Sleep:这可以用示波器测试 BT chip 26M 是否有波形输出来验证.有两种情况:

(1.)蓝牙连接(A2DP ,SCO link)没有断开,导致 Bt chip 不能进入休眠.(程序中为了省电, A2DP,Sco link 连接在一段时间内没有使用的话,手机会主动断开这些连接)。

(2.)可能该款耳机同手机存在兼容性问题, 导致 bt chip 不能进入省电模式, 需要提供耳机抓蓝牙底层数据包来分析原因。

BB 不能进入 Sleep: 可能是连接上耳机后把 BB sleep disable 了, 需要提供 catcher log 分析,filter 设置: (!)Field Trail sleepMode + MOD\_BT(All Classes On) .

对于 2.可能是蓝牙 power on 后, gpio 被拉高导致的.可以在函数 GPIO\_WriteIO 中加 trace 把各个管脚打印出来, 看是否有拉高

## 19. 蓝牙激活不了

---

### [AUTHOR]

MSZ00489

### [Description]

蓝牙 power on,但提示初始化失败

### [Solution]

1.检查蓝牙 gpio 配置,特别是 uart3 RX/TX 是否接反, 蓝牙供电是否正常

2.从 08.52.MP 的版本开始, 不支持 MT6601E3,MT6611E3 及其之前的蓝牙芯片,只支持 MT6601E5,MT6611E5 及其之后的蓝牙芯片。所以此类芯片在旧版本上可以激活, 在 08.52.MP 版本后激活不了。

如果是 MT6611E5,MT6601E5 的芯片在 08.52.MP 的版本也无法激活,则需要打上 patch MAUI\_01536702 , MAUI\_01629447

## 20. 蓝牙版本升级需要替换哪些库

---

### [AUTHOR]

MSZ00489

### [Description]

1.蓝牙做版本升级, 但因为工作量的关系, 又不想重新把应用程序移植到新版本上想在旧版本上做蓝牙的升级。需要替换哪些库文件?

### [Solution]

蓝牙升级总共需要替换三支库文件(btadp.lib,btstack.lib,btprofile.lib).

如果有 L4 代码, btadp 部分将是源代码, 没有 btadp.lib 替换。

另外蓝牙升级的时候, 最好对照两个版本 makefile 文件 BT 部分宏是否有改动, 如果有改动, 也要对应着同步更新过来,这样软件上下才能一致, Makefile 中 BT 部分的宏不能私自关闭, 如果出于节省空间或其它目的要关闭, 请务必申请 patch 来关闭相关 profile, 以更新蓝牙相关库, 否则会出现不易发现的异常情况。

## 21. HFP: 蓝牙耳机不能接听/挂断电话

---

### [AUTHOR]

MSZ00409

### [Description]

来电后, 按蓝牙耳机接听键, 不能将电话接起; 若从手机端接起, 按蓝牙耳机挂断键, 也不能将电话挂断。

### [Solution]

1. 请按照 HFP Debug SOP 检查是否为耳机自身的问题; (注: 大部分情况是耳机的问题), 并告知测试结果。

2. 针对双卡项目(G+C 或 G+G), 若蓝牙耳机不能接听/挂断副机的电话, 则需要自行开发针对副卡的 BT Handfree 处理流程。默认是没有的。

3. 若排除耳机和副卡的因素外, 还不能接听; 则需要提供 log。Filter:  
MOD\_BT,MOD\_L4C: all classes on; MOD\_MMI ( Trace\_INFO, Trace\_Group \_7)

PS.耳机要能接听和挂断电话, 是通过往手机发送 ATA/ATH 来实现的。

## 22. 蓝牙耳机对第三方来电的处理异常

---

### [AUTHOR]

MSZ00409

**[Description]**

三方来电后，按蓝牙耳机：

Case 1: 直接将第三方来电挂掉

Case 2: 提示指令不支持

**[Solution]**

Case 1: 这是在 07A 之前的设计，蓝牙耳机按键后，直接发 ATH 给手机，将电话挂掉。该设计，当时有参考 Nokia 的做法。

Case 2: 耳机端的问题，出现该现象的原因，耳机有发 ATA 给手机：由于手机此时有一通电话是出于 Active，此时在发 ATA 去将电话接起，这是网络所不支持的（因为一个单卡手机，不可能同时有两通电话处于 Active）。针对这种情况，请换其它耳机测试。

### 23. 平台上蓝牙听歌断续

---

**[AUTHOR]**

MSZ00409

**[Description]**

连上 bluetooth handfree/headset profile 之后，听音乐 3~5 分钟之后，声音出现断断续续。

**[Solution]**

23 平台不支持 A2DP，23 平台听歌，需要借助 HFP/HSP 的 SCO Link 来将音乐传送到蓝牙耳机

若出现听歌断续，可按照下面的方法修改：

1、首先找到 AVB.c 文件，然后在文件开头添加宏定义：

```
#define PROTECT_FROM_FULL 7168
```

2、在 AVB.c 找到 avb\_Process 函数（有两个同名函数，后面的函数才是我们需要修改的），修改 while 循环的条件如下：

```
while ( count < inLen && !AVB.isEOF && AVB.downsampler->GetOutputNum(AVB.downsampler)<PROTECT_FROM_FULL)
```

### 24. 蓝牙来电无提示音

---

**[AUTHOR]**

MSZ00409

**[Description]**

来电后，蓝牙耳机没有无提示音，所有耳机都一样

**[Solution]**

1. 此问题一般自行开关 BT\_AUDIO\_VIA\_SCO 导致。请先检查该 Compile option 的值是否有修改到。

2. 按照蓝牙 Debug SOP 检查；同时检查，耳机端是否有被 Mute 住。

3. 检查 HFP SCO Link 是否已经建立成功；铃声是透过 SCO Link 传送到蓝牙耳机（对应的消息为 MSG\_ID\_BT\_HFG\_SCO\_CONNECT\_REQ/CNF）。

PS.为什么自行关闭 BT\_AUDIO\_VIA\_SCO 会导致没有来电提示音？

1) 如果该宏一开始为 TRUE，则在连蓝牙耳机服务时，已经告诉耳机：手机会将彩铃的声音传过去，不用其自带的铃声（这部分是包含在 Lib 里）；而耳机知道后，则不响铃。

2) 若将其置为 FALSE，MMI 层则不会去送铃声。故彩铃没有送过去，从而导致耳机无来电提示音。

## 25. 蓝牙耳机通话发送声音小

---

### [AUTHOR]

MSZ00409

### [Description]

### [Solution]

请将 AM\_DSP\_BluetoothOn 函数中 DP\_VOL\_OUT\_PCM 的值改大一些（0x1000 : 0dB; maximum 0x7FFF）。

## 26. 修改串口名称

---

### [AUTHOR]

MSZ00409

### [Description]

Please change name from "Serial Port0" to "COM" when SPP Profile connected.

### [Solution]

Please change the default value of variable "SerialPortServiceNamePattern" in file Spsdpdb.c from "Serial Port0" to "COM".

If there is no source code, please ask PM to release btprofiles.lib.

## 27. AudioPlayer 中的蓝牙 Setting 的 Bluetooth Stereo Output 开关不起作用

---

### [AUTHOR]

MSZ00409

### [Description]

MP3 中的蓝牙耳机开关不起作用

### [Solution]

该开关的作用，并非关闭音乐，而是开关蓝牙立体声（Stereo）和非立体声。当选择 On 是，表示是选择蓝牙立体声服务 A2DP 来播放音乐，而当选择 OFF 时，则表示选择非立体声（透过 SCO Link 传输，语音音质）来播放音乐。

有 2 种方式来获取播放音乐的蓝牙设备：

1.MP3 播放过程中，打开蓝牙，查找到蓝牙耳机后连接，，输入蓝牙配对密码后再联机  
--> 这种查找方式是只查找支持特定服务的装置：A2DP

2.或者在我的设备中联机后 MP3 即被转移到蓝牙耳机上播放 ==> 该方式是搜索任意蓝牙设备，并选择一个支持 HFP/HSP（蓝牙免持装置服务/蓝牙耳机装置服务）来听音乐。利用 HFP/HSP 支持的 SCO Link，将音乐 Forward 到蓝牙耳机，就像打电话时将话音传送到蓝牙耳机一样。

总之，通过该开关可以在立体声/非立体声之间切换。那为什么两者都要支持呢？因为有些蓝牙耳机，特别是低端蓝牙耳机，有可能不支持 A2DP 服务来听音乐，而为了使得耳机即使不支持 A2DP 也能听音乐，我们采用蓝牙 SCO Link 来将音乐传送到蓝牙耳机，这样就可以听音乐了。只是这种效果比立体声的音质要差一些。

如果要将 Mp3 的声音输出到蓝牙耳机的路径关闭，一，针对非立体声，可以按照您提到的操作去执行.二，针对立体声音乐，您可以进到 Bluetooth-> My active device,选择当前所使用的蓝牙耳机，断开 A2DP 服务，或者直接将蓝牙耳机删除掉。

注：是否支持通过 SCO Link（非立体声）听 MP3，是由 Compile option: BT\_AUDIO\_VIA\_SCO 来控制的，如果为 TRUE，则表示支持。需要注意的是，该 Compile Option 不可自行开关。

## 28. HSP 需要两次按键，才能执行 AT+BLDN

### [AUTHOR]

MSZ00409

### [Description]

蓝牙耳机连接免持设备服务后，在播放音乐中短按一次蓝牙耳机键就可以呼出电话；蓝牙耳机连接耳机设备服务后，在播放音乐中第一次短按蓝牙耳机键切换到手机播放音乐，第二次短按蓝牙耳机键才可以呼出电话，请问设计如此吗？

### [Solution]

设计既是如此。

连上 HSP 时，在 Audioplayer 界面下，的确需要按两次。第一次按键不会去拨号的原因是因此此时存在 SCO Link，此时按键会先断开 SCO Link，第二次按键，检测到 SCO Link 不存在，就会去拨最后拨打的号码。

为什么需要将之前的 SCO Link 断开，才可以去执行 AT+BLDN(拨打手机上最后一次拨打的电话呢)？

原因有三：

其一，可能其它的应用此时正在通过 SCO Link 传送数据，需要断开 SCO Link 来使其它应用不再将数据往蓝牙耳机上发；

其二，打电话的优先级别比其它的应用都要高，毕竟蓝牙耳机 H S P 服务最主要的功能还是打电话。



其三，什么时候去执行 AT+BLDN 是由手机端的逻辑决定的。因为连上 HSP 时，按蓝牙耳机按键（接听/挂断）只会发出 AT+CKPD=200,而不会发出 AT+BLDN，需要手机根据自身的设计，将 AT+CKPD 转换成 AT+BLDN。

另，而 HFP 则可以明确地发出 AT+BLDN，所以 HFP 只需要按一次键。

PS. SCO Link 是蓝牙的一种同步连接，主要用于传送语音数据，如通话音。

## 29. 蓝牙耳机播放按键在接听电话后不能播放音乐

---

### [AUTHOR]

MSZ00409

### [Description]

- 1、菜单—附加功能--蓝牙：与蓝牙耳机连接
  - 2、用蓝牙耳机听立体声音乐，在音乐播放器界面按蓝牙耳机暂停键，音乐暂停
  - 3、此时来电话，用蓝牙耳机接听然后挂断，然后在按蓝牙耳机的播放键播放音乐
- 结果：音乐播放器不能播放音乐，蓝牙耳机播放按键也没有起作用

### [Solution]

这类问题主要是 SCO Link 有影响到耳机端对 AVRCP 的响应，具体怎样影响到，需要了解耳机端的做法。比较可能的原因是耳机端认为 SCO Link 还存在，不应该有 A2DP 的操作。

为了解决此问题，可以将 BT\_SCO\_CONNECT\_TIMEOUT\_DURATION 的值改短，建议为 3s。Nokia 的做法是，如果 SCO Link 超过 1 s 不用，则 Disconnect。

PS. BT\_SCO\_CONNECT\_TIMEOUT\_DURATION 值表示允许 SCO Link 空闲的时间间隔，超过这个时间，将会断开。

## 30. AVRCP 会随 A2DP 的断开而断开

---

### [AUTHOR]

MSZ00409

### [Description]

请问开启蓝牙立体声输出后，在连接中设备中选择进阶声音播放服务断开时会同时断开影音远程控制服务，选择影音远程控制服务断开时不会同时断开进阶声音播放服务。请问是否是设计问题？还是 A2DP 和 AVRCP 这两个 PROFILE 这样约定的。

### [Solution]

的确，设计就是这样。

A2DP 和 AVRCP 一般会一起出现。A2DP 主要用来传送声音数据；AVRCP 主要控制歌曲的播放，上一首/下一首等。

当手机主动与耳机连接上 A2DP 之后，耳机会主动与手机连上 AVRCP；反之，手机主动与耳机断开 A2DP 之后，耳机这时候检测到 A2DP 不存在，也会将 AVRCP 断开。因此，耳机决定是否与手机连上 AVRCP，主要是根据 A2DP 是否存在决定的。

至于强行从手机段，断开 AVRCP，是可以的，但 A2DP 是不会断，这主要是因为 AVRCP 是依附 A2DP，而非 A2DP 依附于 AVRCP。

### 31. A2DP

---

#### [AUTHOR]

MSZ00409

#### [Solution]

A2DP(Advanced Audio Distribute Profile), 俗称蓝牙立体声，采样频率为 16/32/44.1/48 KHz, 相对只有 8KHz 的非立体声（通过 Bluetooth Handfree profile + SCO Link 传送 PCM 数据，主要应用在通话），采样频率要高很多，音质效果也要好很多。

说到 A2DP, 不得不提到 AVRCP(Audio/Video Remote Control Profile). AVRCP 主要用来控制 Audio/Video 的播放，常见的功能有：上一首/下一首，播放/暂停。目前业界的流行得做法是，AVRCP 的存在与否依赖于 A2DP，即耳机根据是否有 A2DP 连接是否存在，决定连接/断开 AVRCP 连接，且 AVRCP 的连接都是由耳机端发起，而不是手机端。

### 32. 如何利用 AT+EBTLB 进行测试 Loop back 2 testing.

---

#### [AUTHOR]

MSZ00489

#### [DESCRIPTION]

工厂利用 AT+EBTLB 进行蓝牙 Loop back 2 testing 操作步骤.

#### [SOLUTION]

For LP2 testing, you have to

- (1) know remote device's bluetooth device address and passcode.
- (2) rework the remote bluetooth headset to route the headset's speaker output to mic input.
- (3) let remote bluetooth headset to be at inquiry scan mode
- (4) send correct AT command to our MS phone.
- (5) then MS phone will setup HFP connection and SCO link with remote bluetooth headset
- (6) send one voice patten via the sco link to remote bluetooth device.
- (7) and we expect that remote headset shall loopback the voice pattern to MS phone and MS phone reply "OK" AT command. If not, MS phone will reply "ERROR".

对于(4),command sequence :

AT+EMBT =1 //power on Bluetooth

AT+EBTLB=1,remote\_bt\_address,"pin code"

### 33. 如何在蓝牙搜索耳机并匹配完成后，默认 Audio player 中蓝牙立体声耳机为开？

---



## [AUTHOR]

MSZ00489

## [DESCRIPTION]

选择一款立体声耳机并与之配对连接，希望 Audio player\Bluetooth setting 中，把该款耳机默认设置开，这样进入 audio player 界面，直接可以用蓝牙耳机播放音乐，不用再去设置与选择立体声耳机。

## [SOLUTION]

该功能实现的代码如下：

1.Av\_bt.c 文件开始处，加入如下两行代码：

```
#include "bt_dm_struct.h" /*MTK70489 add this line*/  
extern MMI_BOOL mmi_bt_is_service_support(mmi_bt_dev_struct *dev_p, U32  
service_id);/*MTK70489 add this line */
```

2.Av\_bt.c 文件中,加入如下函数：

```
void mmi_a2dp_setting_after_device_paired(U32 lap,U8 uap,U16 nap) /*Yunhai add this  
line */  
{  
    mmi_bt_dev_struct * dev_p;  
    U32 j;  
    for (j = 0; j < g_mmi_bth_ctx.paired_dev_num; j++)  
    {  
        if (g_mmi_bth_ctx.paired_list[j].bd_addr.lap == lap  
            && g_mmi_bth_ctx.paired_list[j].bd_addr.uap == uap &&  
g_mmi_bth_ctx.paired_list[j].bd_addr.nap == nap)  
        {  
            break;  
        }  
    }  
    if(mmi_bt_is_service_support((mmi_bt_dev_struct  
*)(&g_mmi_bth_ctx.paired_list[j]),BT_ADVANCED_AUDIO_PROFILE_UUID))  
    {  
        dev_p = (mmi_bt_dev_struct *)&g_mmi_bth_ctx.paired_list[j];  
        a2dp_settings.bt_a2dp_output = MMI_TRUE;  
        a2dp_settings.bt_a2dp_headset_valid = MMI_TRUE;  
        memcpy( &a2dp_settings.bt_a2dp_headset,  
dev_p,sizeof(mmi_bt_dev_struct) );  
        mmi_a2dp_write_bt_settings();
```

```
}
```

```
}
```

3.BTMMICm.c 中，修改代码:

```
void mmi_bth_cm_sdc_search_cnf_hdlr(void *msg)
{
.....

    default:
        break;

}
/*MTK70489 add this line start*/
    mmi_a2dp_setting_after_device_paired(msg_p->bd_addr.lap,msg_p-
>bd_addr.uap,msg_p->bd_addr.nap);
/*MTK70489 add this line end*/

    /* write to NVRAM---store the service list */
    mmi_bth_nvram_write_info();

}
```

如果软件版本中没有:

```
mmi_a2dp_write_bt_settings();
```

```
a2dp_settings
```

请用如下代替:

```
mmi_audply_write_bt_settings
```

```
bt_settings
```

#### 34. 来电，蓝牙耳机听到的音乐是噪音.

---

**[AUTHOR]**

MSZ00489

**[DESCRIPTION]**

某些软件版本，来电的时候，蓝牙耳机听到的来电音是噪音或者噪音很大，接通电话后，声音正常。可能是 EDI 声音导入到蓝牙 DSP 引起.

**[SOLUTION]**

AFE\_TurnOnEDI 裡面 操作 AFE\_VDB\_CON 的那一行拿掉

```
void AFE_TurnOnEDI( kal_uint16 fmt, kal_uint16 cycle )
{
    ASSERT( fmt < 2 );
    ASSERT( cycle == 16 || cycle == 32 );
    /*AFE_VDB_CON |= 0x0040;          /*Delete this line*/
    *AFE_EDT_CON = (((cycle-1)<<2) | (fmt<<1) | 1);
}
```

### 35. 某些版本蓝牙为什么不能 power on ?

---

#### [AUTHOR]

MSZ00489

#### [DESCRIPTION]

同样的硬件，某些版本可以 power on 蓝牙，某些版本 power on 不了。

#### [SOLUTION]

我们软件从 08.52.MP 版本开始不支持 MT6601 E3/MT6611 E3 版本的蓝牙芯片,所以如果使用的是 E3 的芯片,在 08.52.MP 版本之前蓝牙是可以激活的，08.52.MP 之后就激活不了,请不要在 08.52.MP 版本后使用 E3 芯片(E3 芯片已经停止量产很久，可能市场上有存货). 可以使用 E5 及其后续版本蓝牙芯片。

查看蓝牙芯片版本有两种方法:

1. 手机上查看，操作步骤:

Power on 蓝牙/进入工程模式/Bluetooth/Get chip version

2. 查看蓝牙芯片的丝印。

### 36. 关闭 BT 单声道播放音乐，保留来电铃声的功能：

---

#### [Description]

实现功能：关闭蓝牙耳机单声道（SCO Link）播放音乐功能，但 来电铃声(in-band ring)可以传到蓝牙耳机

#### [Solution]

- 1、在 Makefile 中保持 BT\_AUDIO\_VIA\_SCO 为 TRUE
- 2、在 Option.mak 中关闭\_\_ BT\_AUDIO\_VIA\_SCO\_\_

### 37. 不插入 T 卡时，去掉蓝牙 Storage 菜单中的 Memory card 选项

---

#### [Description]

实现功能：在插入 T 卡时，从 Storage 菜单中去掉 Memory card 选项。(08B.08.52)

#### [Solution]

在 mmi\_bt\_entry\_set\_rcv\_drv() 中，将 mmi\_fmgr\_get\_full\_storage\_inline\_list(...) 改成 mmi\_fmgr\_get\_exist\_storage\_inline\_list(...), 参数不变。

### 38. 在发送方取消文件 OPP 发送时，接收方增加一个提示

---

#### [Description]

As title

#### [Solution]

在 mmi\_bt\_opps\_disconnect\_ind\_handler() 判断接收状态，并增加提示：

```
if (GOPP(ops_state) > MMI_OPS_ACTIVE)
{
    //Added by Yuebing begin
        if(GOPP(ops_state) == MMI_OPS_RECEIVING)
        {
            mmi_bt_opp_notify_err_request_hdlr(
                (UI_string_type)GetString(STR_GLOBAL_CANCEL),
                MMI_EVENT_FAILURE);
        }
    //Added by Yuebing end
```

### 39. 如何使 BT 在开机时默认打开或者关闭

---

#### [Description]

目前 base 版本的做法是，在前一次关机时，NVRAM 记录 BT 当前状态 (on or off)，在下次开机时根据这个状态决定 BT 是否打开。

若需要使 BT 不受前一次关机前状态影响，设为默认打开或者关闭，可以通过下面方法进行修改。

## [Solution]

### 1) 默认打开的方法:

在 09B 版本上, 如 09B.W10.12

```
SRV_BT_CM_SET_FLAG(SRV_BT_CM_MASK_ACTIVATED); // 将
SRV_BT_CM_MASK_ACTIVATED 置位

/* update the supported service list of host phone */
while (support_profile_tbl[i].uuid != SRV_BT_CM_MAX_PROFILE_UUID)
{
    g_srv_bt_cm_ctx.host_dev.service_list[i] =
support_profile_tbl[i].uuid;
    i++;
}
g_srv_bt_cm_ctx.host_dev.service_list_num = i;
srv_bt_cm_nvram_write_info();
```

在 09A 及之前版本上, 如 09A.W09.52

```
MMI_BTH_SET_FLAG(MMI_BTH_MASK_ACTIVATED); // 将
MMI_BTH_MASK_ACTIVATED 置位

/* update the supported service list of host phone */
while (support_profile_tbl[i].uuid != MMI_BTH_MAX_PROFILE_UUID)
{
    g_mmi_bth_ctx.host_dev.service_list[i] =
support_profile_tbl[i].uuid;
    i++;
}
g_mmi_bth_ctx.host_dev.service_list_num = i;
mmi_bth_nvram_write_info();
```

### 2) 默认关闭的方法:

在 09B 版本上, 如 09B.W10.12

将 `SRV_BT_CM_SET_FLAG()` 换成 `SRV_BT_CM_RESET_FLAG()`

在 09A 及之前版本上，如 09A.W09.52

将 `MMI_BTH_SET_FLAG()` 换成 `MMI_BTH_RESET_FLAG()`

#### 40. 如何在用户手动断开蓝牙耳机之后，将立体声也关掉？

##### [Description]

用户在手动断开蓝牙耳机的连接后，如果不关闭蓝牙耳机，在切换下一首、暂停后重新播放等情况下，`media player` 会重新连接蓝牙耳机。

由于是用户手动断开的连接，此时自动连上，可能在体验上不会很好。解决方法是在用户手动断开蓝牙耳机后，`media player` 中的立体声设置也关掉。

##### [Version]

09B.10.12

##### [Solution]

1、在 `BTMMIA2DP.c` 中

```
void mmi_bt_a2dp_disconnect_req_by_cm(srv_bt_cm_bt_addr dev_addr, U32 connect_id)
{
    /*-----*/
    /* Local Variables */
    /*-----*/

    /*-----*/
    /* Code Body */
    /*-----*/

    g_mmi_a2dp_ctx.disc_req_flag |= MMI_A2DP_REQ_FLAG_CM;

    switch (g_mmi_a2dp_ctx.state)
    {
```

```
case MMI_A2DP_STATE_CONNECTED:
```

```
    //Added by Yuebing begin
```

```
    mmi_a2dp_bt_settings_turn_off();
```

```
    //Added by Yuebing end
```

```
    bt_a2dp_send_signal_disconnect_req(MOD_MMI, connect_id);
```

```
    MMI_A2DP_STATE_TRANS(MMI_A2DP_STATE_DISCONNECTING);
```

```
    ...
```

```
    ...
```

2、在 av\_bt.c 中添加一个函数：

```
extern void mmi_a2dp_bt_settings_turn_off(void); //Added by Yuebing
```

```
//Added by Yuebing begin
```

```
void mmi_a2dp_bt_settings_turn_off(void)
```

```
{
```

```
    a2dp_settings.bt_a2dp_output = MMI_FALSE;
```

```
    a2dp_settings.bt_a2dp_headset_valid = MMI_FALSE;
```

```
    mmi_a2dp_write_bt_settings();
```

```
}
```

```
//Added by Yuebing end
```

#### 41. Add\_H\_free\_right\_softkey\_in\_bt\_call\_screen

---

Refer to **Add\_H\_free\_right\_softkey\_in\_bt\_call\_screen\_CN.pdf** on

DMS

#### 42. Auto\_Connect\_HFP\_Change\_Feature

---

Refer to **Auto\_Connect\_HFP\_Change\_Feature\_CN.pdf** on DMS

#### 43. Auto\_use\_a2dp\_after\_BT\_headset\_plug\_in

---

Refer to **Auto\_use\_a2dp\_after\_BT\_headset\_plug\_in\_CN.pdf** on DMS

#### 44. Auto\_use\_a2dp\_when\_press\_PLAY\_key

---

Refer to **Auto\_use\_a2dp\_when\_press\_PLAY\_key\_CN.pdf** on DMS

#### 45. Redial\_sim2\_last\_dial\_number\_using\_hfp

---

Refer to **Redial\_sim2\_last\_dial\_number\_using\_hfp\_CN.pdf** on DMS

#### 46. 使用默认密码自动配对 Audio device

---

Refer to **Audio\_device\_auto\_pair.pdf** on DMS

#### 47. 09A 上实现文件传输时蓝牙图标闪烁的方法

---

Refer to **BT\_icon\_blinking\_while\_transfer\_file\_09A.pdf** on DMS

#### 48. 09B 上实现可以使用蓝牙耳机重拨 SIM2 电话的方案

---

Refer to **BT\_SIM2\_redial\_09B.pdf** on DMS

#### 49. PC 使用 FTP 推送文件到手机时，存储位置异常的解决方案

---

Refer to **FTPS\_storage.pdf** on DMS

#### 50. 恢复出厂设置时，实现蓝牙的 Factory Restore 功能

---

##### [Description]

##### 客户需求：

在进行恢复出厂设置操作时，默认设置是不会对蓝牙相关的设置进行恢复的。

若需要在 Menu→Settings→Restore Factory Settings 进行手机恢复出厂设置时，希望将蓝牙相关设置一并进行 Factory Restore，包括：默认关闭蓝牙、恢复蓝牙名称、清空已配对列表信息等（注：不会对蓝牙地址进行更新），则可参考此文档实现。



## [Version]

09B.10.12

## [Solution]

### 1、在 BT NVRAM LID 中，加入 Factory Restore 属性：

文件：nvr\_common\_config.c 中数组：logical\_data\_item\_table\_comm\_app[] 中找到 NVRAM\_EF\_BT\_INFO\_LID 及 NVRAM\_EF\_BTHF\_CONFIG\_LID，将其 Attr 属性更改为：NVRAM\_ATTR\_AVERAGE | NVRAM\_ATTR\_FACTORY\_RESET（原来是 NVRAM\_ATTR\_AVERAGE）。

添加该 attr 后，当进行恢复出厂设置操作时，会对此 NVRAM LID 进行 Restore，如下是对 NVRAM\_EF\_BT\_INFO\_LID 的更改（NVRAM\_EF\_BTHF\_CONFIG\_LID 同理）：

```
{
    NVRAM_EF_BT_INFO_LID,
    NVRAM_EF_BT_INFO_SIZE,
    NVRAM_EF_BT_INFO_TOTAL,
    NVRAM_NORMAL(NVRAM_EF_ZERO_DEFAULT),
    NVRAM_ATTR_AVERAGE | NVRAM_ATTR_FACTORY_RESET,
    NVRAM_CATEGORY_USER,
    "MP25",
    VER(NVRAM_EF_BT_INFO_LID),
    "Bluetooth info",
    NVRAM_APP_RESERVED
}
```

### 2、对系统盘中保存的 BT 相关的 System File 进行 Restore:

添加 Restore BT 相关的 System File 函数：

```
void mmi_bluetooth_restore_factory_default(void)
{
    FS_XDelete(L"z:\\@bip", FS_FILE_TYPE | FS_DIR_TYPE | FS_RECURSIVE_TYPE,
    NULL, 0);
    FS_XDelete(L"z:\\@bt", FS_FILE_TYPE | FS_DIR_TYPE | FS_RECURSIVE_TYPE,
    NULL, 0);
    FS_XDelete(L"z:\\@ftp", FS_FILE_TYPE | FS_DIR_TYPE | FS_RECURSIVE_TYPE,
    NULL, 0);
    return;
}
```

### 3、在 Factory Restore 过程中，添加对蓝牙的 Restore 处理

在 Factory Restore 过程中，添加对蓝牙的 Restore 功能，主要包括两个方面，一是在添加对处理蓝牙 Factory Restore 的 Callback 函数，二是在 Factory Reset 前添加对蓝牙的 Check

。

首先添加如下两个 Check 函数，若判断到当前有蓝牙链接存在，则禁止掉 Factory Restore:

```
U8 mmi_bluetooth_restore_factory_check(void)
{
    if (srv_bt_cm_is_activated () == MMI_TRUE)
    {
        if (srv_bt_cm_get_existed_conn_num())
            return 0;
    }
    return 1;
}
```

```
U8 RstStartRestoreResetAppCheck(void)
{
    U8 result=mmi_bluetooth_restore_factory_check();
    return result;
}
```

另外，在 void RstStartRestoreResetAppStatus(void)中添加对如下函数的调用：

`mmi_bluetooth_restore_factory_default();` //删除相关的系统文件

同时，将 static mmi\_restore\_callback\_struct mmi\_restore\_app\_callback[] =

```
{
    {NVRAM_APP_PHNSET, 0, RstStartRestoreResetAppStatus,NULL},
    . . .
}
```

更改为

```
static mmi_restore_callback_struct mmi_restore_app_callback[] =
{
    {NVRAM_APP_PHNSET, 0, RstStartRestoreResetAppStatus,
    RstStartRestoreResetAppCheck},
```

```
    . . . . .  
}
```

## 51. 若手机剩余存储空间不够，Popup 提示请插入 T 卡，并结束本次接收文件过程 (09A)

---

### [AUTHOR]

71024

### [Description]

正常流程是在接收过程中，写文件由于空间不够而失败才提示 memory full。按此方法修改是在接收开始判断手机剩余空间是否足够接收该文件，若否才提示用户插入 T 卡，当然本次接收也会失败。

### [Solution]

```
mmi_bt_opps_proc_first_push_data  
{S32 freespace;  
...  
retval = mmi_opp_gen_temp_obj(  
    GOPP(recv_path),  
    GOPP(recv_drv),  
    ind->obj_mime_type,  
    GOPP(ops_obj_name));  
MMI_OPP_LOG1(MMI_BT_OPP_OPPTS_GEN_TEMP_OBJ, retval);  
if (retval < FS_NO_ERROR)  
{  
    mmi_opp_set_fs_flag(retval);  
    mmi_bt_opps_push_ind_write_error();  
    return;  
}  
//add begin  
if(GOPP(recv_drv) == MMI_PUBLIC_DRV)    //若存储位置为收集  
{freespace=fmgr_main_check_free_space(MMI_PUBLIC_DRV,ind->total_obj_len); //剩余空  
间不足够接收本次文件  
  
if(freespace)    //freespace=0 mean there is enough space  
{
```

```

//在这里 popup 需要提示的内容
MMI_OPS_LOG_STATE_ERR();
mmi_bt_opps_push_res(GOPP(ops_goep_conn_id), GOEP_INTERNAL_SERVER_ERR);
return;
}
//add end
...}

```

## 52. MTK 手机给 Android 手机发送文件，Android 手机确认接收前发送方进度不为 0%

### [AUTHOR]

71024

### [Description]

文件发送过程，MMI 会划分多个 Packet 给 BT 再发送（第一个包不带数据，只有相关文件信息），我们平台在接收过程是在接收第一个包时提示接收，确认之后再接收包含文件数据的包，所以不会有百分比显示错误的问题。

而 android 平台在接收过程是接收第二个包时才提示确认接收（发送数据已更新），由于接收确认方式不同所以会造成百分比显示不正常。

### [Solution]

发送时对方响应后才更新文件进度（原来为在发送请求后，对方响应前就更新）。(09B)

#### 1、oppsrv.c

起始处定义 U32 my\_remain\_put\_len=0;

#### 2、srv\_oppc\_send\_first\_packet

```

{...
FS_GetFileSize(srv_opp_p->oppc_ctx.h_snd, &len);
srv_opp_p->oppc_ctx.total_obj_len = len;
srv_opp_p->oppc_ctx.remain_put_len = len;
my_remain_put_len=len; //add this line
...}

```

#### 3、srv\_oppc\_push\_rsp\_handler

```

{...
my_remain_put_len = srv_opp_p->oppc_ctx.remain_put_len; //add this line
if (srv_opp_p->oppc_ctx.remain_put_len > 0)
{
...
}
}

```

```

4、btmmiopp.c
extern U32 my_remain_put_len;
5、mmi_opp_calculate_persent
{...
//return (U8) (((double)(obj_info->total_obj_len - obj_info->remain_put_len) /
return (U8) (((double)(obj_info->total_obj_len - my_remain_put_len) /
(double)obj_info->total_obj_len) * 100.0);
}

```

### 53. 搜索过程无 pair 选项，必须先选项 cancel 才能进入下一步配对

#### [AUTHOR]

71024

#### [Description]

10A11.01 后 6432 项目上出于节省更多的 Memory，目前是在搜索过程没有 cancel 直接配对的 feature 去掉了，而且这样与部分对比机的行为也一致。如果需要与之前的流程一致，可按以下步骤完成。

#### [Solution]

1、BTMMISCR.C 文件起始处增加定义  
kal\_uint8 search\_abort;

定义以下函数

```

void my_entry_abort_discovering(void)
{
    srv_bt_cm_search_abort ();
    search_abort=TRUE;

    mmi_bt_show_progress_scr(g_mmi_bt_scr_cntx.inq_gid, SCR_BT_PAIR_PRO,
    STR_BT_START_PAIRING, STR_BT_START_PAIRING, NULL);
}

```

## 2、mmi\_bt\_entry\_inq\_result

起始处赋值

```
search_abort=FALSE;
```

...

并作以下修改

```
ShowCategory184Screen(  
    STR_BT_SCH_RESULT,  
    IMG_BT_SCH_TITLE,  
    STR_BT_PAIR_REQ, //软按键字符串  
    IMG_GLOBAL_OK,   //软按键图标  
    //0,  
    //0,  
    rsk_str,  
    IMG_GLOBAL_OK,  
    discov_dev_counter,  
    mmi_bt_get_discovered_item,  
    mmi_bt_get_discovered_item_hint,  
    (S32) g_mmi_bt_scr_ctx.hilight_item,  
    guiBuffer);
```

## 3、mmi\_bt\_inq\_result\_hilite\_hdlr 做以下修改

```
{...
```

```
if (g_mmi_bt_scr_ctx.flag & MMI_BT_INQUIRY)
```

```
{
```

```
    SetRightSoftkeyFunction(mmi_bt_inq_result_cancel,KEY_EVENT_UP);
```

```
    //add begin
```

```
    SetLeftSoftkeyFunction(my_entry_abort_discovering, KEY_EVENT_UP);
```

```
    ChangeCenterSoftkey(0, IMG_GLOBAL_COMMON_CSK);
```

```
    SetCenterSoftkeyFunction(my_entry_abort_discovering, KEY_EVENT_UP);
```

```
    //add end
```

```
}
```

## 4、mmi\_bt\_inq\_complete 做以下修改

```
{...
```

```

if (discov_dev_number == 0)
{
...
}

//add begin
else
if(search_abort==TRUE)
{
    srv_bt_cm_bond ((U32) (g_mmi_bt_scr_ctxt.highlight_item));

    /* to notify bt profile that inquiry is stopped */
    #if defined(__MMI_BT_AUDIO_VIA_SCO__) || defined(__MMI_BT_FM_VIA_SCO__)
        srv_bt_sco_inquiry_stop_callback();
    #endif
    return;
}
//add end

else
{ ...

```

#### 54. 蓝牙接收文件存储位置设置，有 T 卡存 T 卡，无 T 卡存手机(09A)

---

##### **[AUTHOR]**

71024

##### **[Description]**

可在开机读取 nvram 时增加判断 memory card 的状态，该修改可能会与蓝牙存储路径设置菜单冲突，建议修改这里需要去掉蓝牙存储路径设置菜单。

##### **[Solution]**

开机读取 nvram 设置时，增加判断是否当前有存储卡，设置相关全局变量（需保存 nvram，如果没有去掉该菜单）。

//不同版本可能在其他函数，只需搜索读取 NVRAM\_BT\_RECEIVE\_DRIVE 的地方，对应的地方进行赋值就可以了

```
mimi_btcomn_nvram_read{...结尾处添加
if(FS_GetDevStatus(MMI_CARD_DRV, FS_MOUNT_STATE_ENUM) == FS_NO_ERROR)
{g_mmi_btcomn_cntx.recv_drv = MMI_CARD_DRV;
g_mmi_btcomn_cntx.always_ask = 0;
}

else if(FS_GetDevStatus(MMI_CARD2_DRV, FS_MOUNT_STATE_ENUM) ==
FS_NO_ERROR)
{g_mmi_btcomn_cntx.recv_drv = MMI_CARD2_DRV;
g_mmi_btcomn_cntx.always_ask = 0;
}

else
{g_mmi_btcomn_cntx.recv_drv = MMI_PUBLIC_DRV;
g_mmi_btcomn_cntx.always_ask = 0;
}

mimi_btcomn_nvram_write(g_mmi_btcomn_cntx.recv_drv,
g_mmi_btcomn_cntx.always_ask);
}
```

## Bluetooth Reference Design

---

1. 蓝牙耳机单声道听歌，进搜索后，马上发送文件，导致声音从手机端播放 (09A)

---

**[AUTHOR]**

71024

**[Description]**



由于 chip limitation 进搜索后，会断开 sco link，停止搜索后会开启 3s timer 再去连接 sco link，此时可造成 goep connect 与 sco link connect 重叠在一起——send goep connect req then send sco link connect(not receive goep connect cnf)，导致建立 sco link 失败。

### [Solution]

可按一下两个步骤解决，主要是保证先建立 sco link 然后再建立 goep connect

- 1、缩短停止搜索后建立 sco link 的 timer
- 2、延后发送 goep connect req，直到建立 sco link 完成。

#### 第一步

设置 timer 的函数是：mdi\_audio\_start\_background\_timer，为不影响 audio 其他地方的设置，可自定义一个和该函数结构类似的函数

```
void mdi_audio_resume_background_play_by_app(MDI_AUD_SUSPEND_BG_APP
app_id)
{
    /*-----*/
    /* Local Variables */
    /*-----*/

    /*-----*/
    /* Code Body */
    /*-----*/

    MDI_AUDIO_LOCK;
    MMI_TRACE(MMI_MEDIA_TRC_G4_MDI,
MDI_AUDIO_TRC_RESUME_BG_PLAY_APP, app_id);

    g_mdi_aud_suspend_app[app_id/8] &= ~(1 << (app_id%8));

    // mdi_audio_start_background_timer();
    my_mdi_audio_start_background_timer(); //change for short timer
    MDI_AUDIO_UNLOCK;
}
```

void my\_mdi\_audio\_start\_background\_timer(void) //自定义函数

```
{
```

```

/*-----*/
/* Local Variables */
/*-----*/

/*-----*/
/* Code Body */
/*-----*/
MDI_AUDIO_LOCK;
if (!mdi_audio_is_background_play_suspended() && !mdi_audio_is_in_call())
{
    StartTimer(TIMER_PROFILES_ALERT_PLAY, 700,
mdi_audio_background_timeout_handler);
}
MDI_AUDIO_UNLOCK;
}

```

## 第二步 红色字体为自定义全局变量或自定义函数

```

1、 mmi_profiles_bt_inquiry_start_callback
{....
    disconn_sco_inq=FALSE;
    case MDI_AUDIO_BT_STATE_OPEN:
    case MDI_AUDIO_BT_STATE_WAIT_OPEN:
        if(g_mmi_profiles_bt_in_call)
        {
            g_mmi_profiles_bt_inquering = MMI_FALSE;
            /* May let caller always call stop inquiry function to resume background play*/
            mdi_audio_resume_background_play_by_app(MDI_AUD_SUSPEND_BG_SCO_IN
Q);
            return MMI_PROFILES_BT_INQ_SCO_CONNECTED_IN_CALL;
        }
        StopTimer(BT_SCO_CONNECT_TIMER);
        g_mmi_profiles_bt_inquiry_callback = callback;
// add flag
    disconn_sco_inq=TRUE; //disconnect sco link while bt in inquiring ( sco link exist)
// add end
    mmi_profiles_bt_disconnect_sco();
    break;

```

```

2、mmi_opc_conn_server
{..
//mmi_bt_oppc_connect_req((U16)OPPC_MAX_OBEX_PACKET_LENGTH,
GOPP(opc_dev_addr));
if(disconn_sco_inq==TRUE)
{goep_timer=0;
gui_start_timer(500,send_goep_connect_req); // 500ms can be adjust
}
else
mmi_bt_oppc_connect_req((U16)OPPC_MAX_OBEX_PACKET_LENGTH,
GOPP(opc_dev_addr));
}

```

```

3、send_goep_connect_req
{
if (aud_bt_hfp_ctx.state == AUD_BT_HFP_STATE_OPEN || goep_timer>5) //wait for sco
connect finsh, or wait too long
{mmi_bt_oppc_connect_req((U16)OPPC_MAX_OBEX_PACKET_LENGTH,
GOPP(opc_dev_addr));
}
else
gui_start_timer(500,send_goep_connect_req);
goep_timer++;
}

```

2. 发送文件，选择新设备->搜索设备->配对成功后，需要在配对列表中选择新配对的设备才能发送

---

**[AUTHOR]**

71024

**[Description]**

可在 MMI 修改使现有流程与对比机一致，即在该界面配对成功后直接发送给该设备。

**[Solution]**

1、在 BTMMISCR.C 起始处定义

```
kal_uint8 search_dev_andsend;
```

2、mmi\_bt\_my\_specific\_dev\_group\_proc

```
{...
```

```
    case EVT_ID_CUI_MENU_ITEM_SELECT:
```

```
    {
```

```
        MMI_BT_SET_HILIGHT_ITEM(menu_evt->highlighted_menu_id);
```

```
        if(menu_evt->highlighted_menu_id == 0)
```

```
        { if (g_mmi_bt_scr_cntx.specific_cod_mask &
```

```
MMI_BT_OBJECT_TRANSFER_MAJOR_SERVICE_MASK)
```

```
            search_dev_andsend=TRUE; //如果是在发送文件的设备列表界面
```

```
mask flag
```

```
            mmi_bt_inquiry_specific_cod_dev();
```

```
        }
```

```
        else
```

```
        {
```

```
            mmi_bt_select_my_specific_dev();
```

```
        }
```

```
    }
```

```
    break;
```

```
...
```

```
    case EVT_ID_CUI_MENU_CLOSE_REQUEST:
```

```
    { search_dev_andsend=FALSE; // 退出这个 screen group，置为 false
```

```
        mmi_bt_entry_my_specific_dev_rsk_hdlr();
```

```
        break;
```

```
    }
```

配对之后即发送

3、在 btcmsrvprmt.c 文件起始处增加

```
extern kal_uint8 search_dev_andsend;
```

4、srv\_bt\_cm\_sdc\_search\_cnf\_hdlr 结尾处 //配对成功会进行 service search

```
...
```

```

    srv_bt_cm_nvram_write_info();
    //add begin
    if(search_dev_andsend==TRUE)
    {

        g_mmi_bt_scr_ctx.hilight_item=g_srv_bt_cm_ctx.paired_dev_num+1;
        mmi_bt_select_my_specific_dev();

    }
    //add end
}

```

配对、并设置鉴权后再发送

3、mmi\_bt\_cm\_auth\_group\_proc

```

{...case EVT_ID_GROUP_DELETE_REQ_IN_END_KEY:
    //add begin
    if(search_dev_andsend==TRUE)
    {kal_prompt_trace(MOD_MMI,"@@search_dev_andsend=TRUE,send
file");

        g_mmi_bt_scr_ctx.hilight_item=g_srv_bt_cm_ctx.paired_dev_num+1;
//g_srv_bt_cm_ctx 需要进行声明
        mmi_bt_select_my_specific_dev();
    }
    //add end
    mmi_frm_end_scenario(MMI_SCENARIO_ID_DEFAULT);
    break;
default:
    break;
}

```