

# MV 蓝牙协议栈中 SPP 功能使用说明

V1.0 2014-9-16

## 1. 引言

本文档描述了 MV 蓝牙协议栈中蓝牙串口（SPP）功能的使用方法。协议栈仅支持 SPP 的 Server 角色，即仅支持手机、电脑、平板等 Master 设备主动连接的情形，不支持 SPP 的开机回连，也不支持两个设备 SPP 的互联。SPP 功能需要在协议栈初始化中配置为 enable 后才能使用，否则为未开启状态。启用 SPP 的配置主要有如下几种：

BT\_FEATURE\_SUPPORTED\_ALL

BT\_FEATURE\_SUPPORTED\_HIFI\_SPP

BT\_FEATURE\_SUPPORTED\_SPP

BT\_FEATURE\_SUPPORTED\_HF\_SPP

蓝牙 SPP 的本身仅仅是一个数据通路，并不涉及更多的数据交互协议。如果用户希望将 SPP 应用到某个特定情景下，用户需要自己预先定义好相关的数据交互协议并在主机端及音箱端（MV BT 运行的设备）做对应的功能实现。本文档介绍的为音箱端 SPP 对数据的处理方法及注意事项。主机端的实现方法不属于本文档描述的范畴。山景标准发布的 SDK 中通常会包含一个能运行在 android 系统下并且与标准 SDK 相匹配的一个功能示例 APK。用户可在此基础上做任意合适的裁剪和扩充。

## 2. 使用方法

SPP 代码的入口定义在 SDK 发布标准文件【spp\_app\_proc.c】中的函数

**DWORD UserProcSppData(VOID);**

该函数为 SPP 交互的**唯一**入口，SPP 的各相关接口函数一定需要在该入口之下进行调用，**不得在其他任意地方进行调用**。该函数由协议栈进行回调处理，因此原则上该函数内的所有处理均不得采用明显的长时间**阻塞**的方式进行实现。函

数的返回值**务必**遵守下面的两个规则：

- 1) 当 SPP 内有数据被消耗（或取出）返回 1（非零）
- 2) 当 SPP 内无数据消耗（或取出）返回 0 值。

SPP 的所有接口函数都定义在文件【spp\_app\_proc.h】中，主要包括如下几个函数

```
void      SppFlushRecBuf(int16_t Size);
uint16_t  SppSendData(uint8_t* Buf, uint16_t Size);
int16_t   SppGetSendBufFreeSize(void);
uint16_t  SppReadData(uint8_t* Buf, uint16_t Len);
uint16_t  SppGetRecBufDataSize(void);
```

- 执行函数 SppFlushRecBuf（int16\_t Size）将会清除接收缓冲区中指定数据长度的数据，如果缓冲区中的数据少于期望的值，则情况缓冲区后返回。如果输入参数-1 表示清空所有缓冲区数据。
- 执行函数 SppSendData（uint8\_t\* Buf, uint16\_t Size）将会发送数据到发送缓冲区。如果发送缓冲区剩余空间小于给定的数据量，那么缓冲填满后函数返回，但不会返回实际发送数据大小。本函数仅返回值表示填充缓冲区的行为是否得到执行（不论实际填充数据的多少）。因此在发送前建议调用函数 SppGetSendBufFreeSize（）获取当前发送缓冲区大小。  
**注意：在原地等待当前发送缓冲区可用量的变化是没有意义的。**
- 执行函数 SppGetSendBufFreeSize（void）返回当前可用发送缓冲区大小。
- 执行函数 SppReadData（uint8\_t\* Buf, uint16\_t Len）从接收缓冲区中读取指定数据量的内容。返回值为实际读取的数据量。建议在读取前可以通过调用函数 SppGetRecBufDataSize（）获取当前缓冲区数据的量。

**注意：在原地等待以期读取更多的数据是没有意义的。**

- 执行函数 SppGetRecBufDataSize（void）可获得当前接收缓冲区中的数据量。

### 3. 使用示例

为了 SPP 功能的正常交互，下面为一个最小系统的使用示例：

```
uint32_t UserProcSppData(void)
{
    uint16_t Size = SppGetRecBufDataSize(); //获取当前接收缓冲区数据量
    if(0 == Size) //如果没有数据
    {
        //如果需要发送数据可在这里增加代码
        // to do ...

        return 0; //一定返回零值
    }

    //如果需要处理接收到的数据可在这里增加代码
    // to do ...

    //如果需要发送数据可在这里增加代码
    // to do ...

    SppFlushRecBuf (-1); //清空所有接收缓冲区
    return 1; //一定返回非零值
}
```

### 4. 注意事项

使用过程中需要注意如下几点：

A) 所有接口函数的调用**务必**在函数 UserProcSppData（）内才是安全的。

B) 协议栈内 SPP 的数据有**收、发双向软件流控**，如果接收缓冲区中的数据不被取走，那么有可能导致发送端数据发送失败。

C) 协议栈内 SPP 的接收、发送缓冲区均大小超过 512 字节，建议每次数据的发送不要超过 512 字节。如果音箱端有写盘操作，建议每次以 512 个字节进行收发效率会高一些。

D) 协议栈不支持 SPP 的开机回连。

E) 如果**仅连接 SPP**，注意到某些设备（如电脑）的配对操作仅支持某种类型的 IOP，如果发现配对异常可在协议栈初始化前先调用下述函数

BTSetIoCapSyncFlag (void)

如果先配对连接其他的 profile 后再连接 SPP，一般不会出现异常。

修订记录

V1.0, yzhou initial version, 2014-9-16