

ANDROID_BLE&SPP_SDK

深圳市中易腾达科技股份有限公司

编号：ANDROID_BLE&SPP_SDK

版本：ANDROID_BLE&SPP_SDK_V1.0.0

修订记录

时间	版本	修订人	备注
2015-01-27	V1.0.0	Jack-Liang	初稿

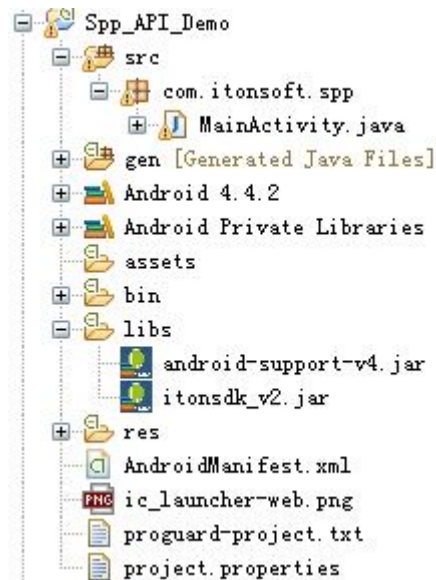
SPP 集成步骤及示例分析

1. 新建工程，导入 SDK JAR 包

添加 JAR 包：

1. 将官方的提供 SDK JAR 包 (itonsdk_v2.jar) 放到工程的 libs 目录下
2. 添加 JAR 包：工程 右键 properties java build path libraries add external Jar;

DEMO 工程如下图：



同时，我们需要在 AndroidManifest.xml 文件中，添加应用程序权限：

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

2. 创建 SPP API 接口类对象

在 MainActivity 中，创建微博 API 接口类对象 itonSpp。

```
itonSpp = new ItonSpp(getApplicationContext(), new ItonUartCallBack() {});
```

itonSpp.itonInit(); 打开蓝牙适配器（必有调用）

3. ItonUartCallBack 接口

SPP 有用接口：

```
new ItonUartCallBack() {
    @Override
    public void itonDeviceStatue(final int isConnect) {
        // 当前状态 Auto-generated method stub
        runOnUiThread(new Runnable() {
            public void run() {
                String deviceState = null;
                if (isConnect == 0) {
                    deviceState = "断开";
                } else if (isConnect == 1) {
                    deviceState = "正在连接";
                } else {

```

```
        deviceState = "连接";
    }
    display.append(deviceState + "\n");
}

});

}

@Override
public void itonRead(final byte[] readBuffer, final int
length) {
    // 数据接口Auto-generated method stub
    runOnUiThread(new Runnable() {
        public void run() {

        }
    });
}

}
```

4. 学习 ItonSpp 对象方法

1. **public void** itonInit(); 打开蓝牙适配器（必须调用）；
2. **public void** itonConnectDevice(BluetoothDevice device) 连接蓝牙设备
3. **public void** itonDisconnectDevice() 断开蓝牙设备
4. **public int** itonGetCurrentState() 蓝牙当前状态 0 断开, 1 正连接, 2 连接
5. **public void** itonWirte(**byte**[] wirteBuffer) 写命令

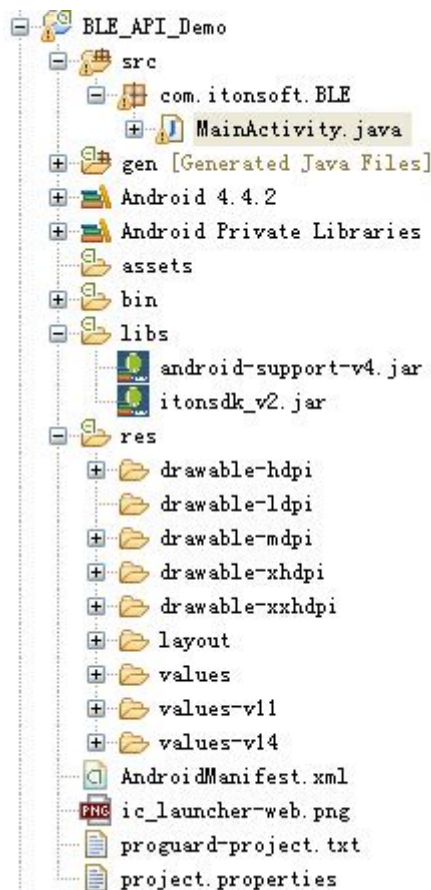
BLE 集成步骤及示例分析

3. 新建工程，导入 SDK JAR 包

添加 JAR 包：

1. 将官方的提供 SDK JAR 包（itonsdk_v2.jar）放到工程的 libs 目录下
2. 添加 JAR 包：工程 右键 properties java build path libraries add external Jar;

DEMO 工程如下图：



同时，我们需要在 AndroidManifest.xml 文件中，添加应用程序权限：

```
<uses-feature
    android:name="android.hardware.bluetooth_le"
    android:required="true" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

4. 创建 BLE API 接口类对象

在 MainActivity 中，创建微博 API 接口类对象 itonBLE 。

```
itonBLE = new ItonBLE(getApplicationContext(), new ItonUartCallBack() {});
```

itonBLE .itonInit();打开蓝牙适配器（必有调用）

4. ItonUartCallBack 接口

BLE 有用接口：

```
new ItonUartCallBack() {
```

```
@Override
```

```
public void itonDeviceStatue(final int isConnect) {
```

```
// 当前状态 Auto-generated method stub
```

```
runOnUiThread(new Runnable() {
```

```
public void run() {
```

```
String deviceState = null;
```

```
if (isConnect == 0) {
```

```
deviceState = "断开";
```

```
        } else {  
            deviceState = "连接";  
        }  
    }  
  
    });  
  
}  
  
@Override  
public void itonRead(final byte[] readBuffer, final int  
length) {  
    // 数据接口Auto-generated method stub  
    runOnUiThread(new Runnable() {  
        public void run() {  
  
        }  
    });  
  
}
```

5. 学习 ItonBLE 对象方法

3. **public void** itonInit(); 打开蓝牙适配器（必须调用）；
4. **public void** itonConnectDevice(BluetoothDevice device) 连接蓝牙设备
6. **public void** itonDisconnectDevice() 断开蓝牙设备
7. **public int** itonGetCurrentState() 蓝牙当前状态 0 断开, 2 连接
8. **public void** itonWirte(**byte**[] wirteBuffer) 写命令