

FIZZO WATCH SDK

Android SDK 开发指南

版本：1.0

2017.5.4

配置工程

导入 Module

1. 将“FizzoWatchSDKSample”项目中的“fizzoWatchLib”库导入到工程中。
2. 在主工程的“build.gradle”中加 `compile project(':fizzoWatchLib')`,让主工程可引用 SDK。

修改你的 Application 类

1. 在 Application 类中增加 SDK 初始化代码：

```
@Override
public void onCreate() {
    super.onCreate();
    applicationContext = getApplicationContext();
    //初始化Fizzo SDK
    Fw.getManager().init(this);
    Fw.getManager().setDebug(true);
}
```

配置你的 AndroidManifest

1. 在 application 中增加配置 Application 的代码

```
<application
    android:name=".LocalApplication"
```

2. 增加 SDK 需要的权限

```
<!-- 蓝牙必备权限 -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

3. 增加 SDK 中的蓝牙变化监听接收器

```
<!-- 系统事件广播接收器 -->
<receiver android:name="cn.fizzo.watch.receiver.BluetoothReceiver">
    <intent-filter>
        <action android:name="android.bluetooth.adapter.action.STATE_CHANGED" />
    </intent-filter>
</receiver>
```

功能（主要参考 ConnectActivity 代码）

连接 FIZZO 设备

1. 连接 API :

```
Fw.getManager().addNewConnect(address)
```

参数 address: 连接设备的 mac 地址

2. 断开连接 API :

```
Fw.getManager().disConnectDevice();
```

3. 注册连接状态回调

```
Fw.getManager().registerConnectListener(this)
```

4. 注销连接状态回调

```
Fw.getManager().unRegisterConnectListener(this)
```

5. 实现监听状态的回调

```
public void connectStateChange(int state)
```

参数 state: 连接过程中的状态

State	含义
ConnectStates. <i>CONNECTING</i>	正在连接
ConnectStates. <i>DISCONNECT</i>	连接断开（有需要会自动重连）
ConnectStates. <i>CONNECT_FAIL</i>	连接发生错误（有需要会自动重连）

ConnectStates. <i>CONNECTED</i>	物理连接成功（不代表能工作）
ConnectStates. <i>CHECK_MANUFACTURER_OK</i>	验证制造商成功
ConnectStates. <i>CHECK_MANUFACTURER_FAIL</i>	验证制造商失败
ConnectStates. <i>WRITE_UTC_OK</i>	写入手表 UTC 时间成功
ConnectStates. <i>NOTIFY_HR_OK</i>	监听心率变化成功
ConnectStates. <i>READ_FIRMWARE_OK</i>	获取手环版本信息成功
ConnectStates. <i>NOTIFY_PRIVATE_SERVICE_OK</i>	监听私有服务成功(工作状态 OK)

获得心率等实时数据

1. 注册获取心率数据回调

```
Fw.getManager().register NotifyHrListener(this)
```

2. 注销获取心率数据回调

```
Fw.getManager().unRegister NotifyHrListener(this)
```

3. 实现心率等数据回调

```
public void notifyHr(HrEntity hrEntity)
```

参数 HrEntity：心率等实时数据类

字段（方法）	含义
getHr()	心率
getCadence()	步频
getStepCount()	今日累计步数（灯管开时可获取）

开关手环光管

1. 打开/关闭 光管

```
Fw.getManager().controlLight(boolean open);
```

参数 open: true,打开, false,关闭

激活反馈

1. 振动

```
Fw.getManager().vibrate();
```

2. 振动后短按反馈

a. 注册短按反馈回调

```
Fw.getManager().registerNotifyActiveListener(this);
```

b. 注销短按反馈回调

```
Fw.getManager().unRegisterNotifyActiveListener(this);
```

c. 实现短按反馈回调

```
public void notifyActive()
```