天天手环 SDK 接口说明

版本号	修改内容描述	修改人	日期
V3.0	1.添加心率接口 2.添加久坐提醒接口 2.部分方法重命名	蔡露	2016-03-17
V4.0	修改设置关闭闹钟不成功的 bug	蔡露	2016-05-04
V6.0	固件升级，升级到设备另外特性。	黄健超	2016-06-06
V1.7.0	添加公英制单位切换接口	黄健超	2016-06-14
V1.8.0	添加交通卡通道接口	黄健超	2016-07-27
V1.9.0	添加 UV,动静态心率切换，增加智能小蜜蜂 SDK	黄健超	2016-10-19
V1.9.1	ICallbackStatus 类中静态整型变量改为 final 型；处理连续发送指令，延迟时间太久的问题。	黄健超	2016-11-01
V1.9.2	优化号码显示不全	黄健超	2016-11-03
V1.9.3	增加密码验证（只支持 dialog 平台且支持密码验证的固件）	黄健超	2016-11-17
V1.9.5	支持内容推送（WriteCommand.sendTextToBle(Content,TYPE)）	黄健超	2016-12-23
V1.9.6	增加游泳接口和血压接口	黄健超	2017-01-09
V1.9.7	优化各国电话号码推送（无新增接口）	黄健超	2017-01-10
V1.9.8	增加抬手亮屏，横竖屏切换，勿扰模式，兼容全字库推送	黄健超	2017-01-20
V1.9.9	增加天气 api，横竖屏控制 api	黄健超	2017-01-21
V2.0.0	优化固件升级（请参考 demo, MainActivity 中的 OnCreate 增加 mUpdates.registerBroadcastReceiver();），其他地方无变化	黄健超	2017-02-11
V2.0.1	1、 解决手环与应用上的路程和卡路里不一致的问题； 2、 增加写入操作状态的系统回调（请参考 demo, MainActivity） 3、 Demo 扫描界面增加信号值显示 4、 修改优化固件升级，添加了访问服务器获取固件版本的回调（请参考 demo, MainActivity）	黄健超	2017-02-21
V2.1.0	1、 优化睡眠策略，修改接口 2、 Public SleepTimeInfo querySleepInfo(String calendar) 去掉第一个参数。	黄健超	2017-03-02

V2.2.1	3.4. Updates固件升级 1、固件升级部分，有固件升级时，增加获取即将升级的固件版本号的方法。 2、修改手环闹钟接口，增加设置手环振动次数 3、优化睡眠策略 4、修改了一些文件的路径，导入新架包发现错误，请重新导包（注意 AndroidManifest.xml 中 BluetoothLeService 的路径）	黄健超	2017-03-21
V2.2.2	1、改名原公英制单位和小时制接口 sendUnitToBLE() 为 sendUnitAndHourFormatToBLE()；新增公英制单位和小时制接口，可传参数设置sendUnitAndHourFormatToBLE(int unitType, int hourFormat) 2、增加新的天气接口 syncWeatherToBLE，让开发者可以自己获取天气信息传入。	黄健超	2017-03-30
V2.3.0	1、 兼容数据库结构调整前的版本（解决覆盖安装，睡眠数据库不兼容的问题） 2、 增加Ibeacon 接口 3、 增加手环表盘切换和左右手切换 4、 增加同步最近七天天气接口	黄健超	2017-04-14
V2.4.0	1、 完善ibeacon的功能，添加TX power和advertising interval广播间隔； 2、 增加久坐提醒，带屏幕，三个闹钟的功能标志判断	黄健超	2017-04-25
V2.4.1	1、libs下的libAesJni.so更新和增加，请按照提供的demo修改	黄健超	2017-05-11

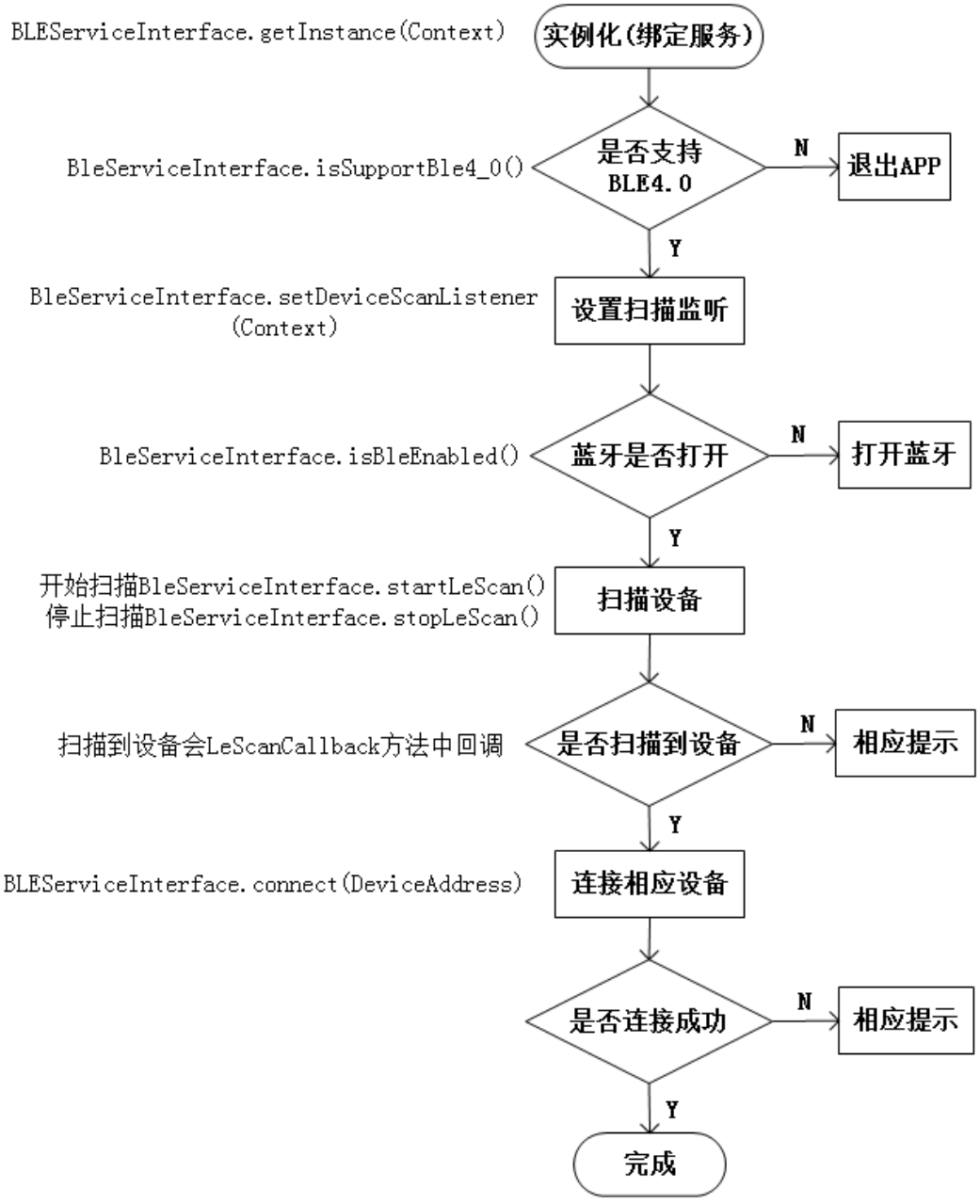
备注：

1、每更新一个版本，为了让开发者快速找到新增内容，我们会在新增内容处添加红色“（新）”的标识。

2、手环能保存 7 天的计步、睡眠、心率数据，而心率每天只保存 7 条最新的心率数据，超出 7 条，会用最新的数据替换最旧的数据。

目 录

1.启动流程图



2.具体类及其相应 API 方法接口

2.1.DeviceScanInterfacer 设备扫描回调接口

方法	`public void LeScanCallback(BluetoothDevice device, int rssi)`
返回值	无
参数说明	device：设备(包含设备所有信息，如设备名、设备地址等) rssi:线损值
说明	要获取扫描到的设备，需实现 DeviceScanInterfacer 接口，当扫描到设备时，在 LeScanCallback 中返回结果

2.2.BluetoothAdapter 蓝牙适配器

备注：适配器通过(BluetoothManager) getSystemService(Context.*BLUETOOTH_SERVICE*).getAdapter()获取	
方法	**boolean** isEnabled()
返回值	boolean
参数说明	无
说明	判断蓝牙是否打开
方法	**boolean** startLeScan(LeScanCallback callback)
返回值	boolean
参数说明	callback：扫描回调方法
说明	开始扫描设备
方法	**void** stopLeScan(LeScanCallback callback)
返回值	无
参数说明	callback：扫描回调
说明	停止扫描设备

2.3.WriteCommandToBLE 与蓝牙端通信相关类

方法	`public static WriteCommandToBLE getInstance(Context context) {`
返回值	返回该类对象
参数说明	上下文
说明	单例实例化
方法	**public void** syncBLETime()
返回值	无
参数说明	无
说明	同步蓝牙端时间
方法	**public void** sendToReadBLEVersion()
返回值	无
参数说明	无
说明	请求蓝牙版本信息(通过广播 GlobalVariable.READ_BLE_VERSION_ACTION 返回,version=intent.getStringExtra(GlobalVariable.INTENT_BLE_VERSION_EXTRA))
方法	**public void** sendToReadBLEBattery()
返回值	无
参数说明	无
说明	请求蓝牙电池电量指令(通过广播 GlobalVariable.READ_BATTERY_ACTION

	返回，battery=intent.getIntExtra(GlobalVariable.INTENT_BLE_BATTERY_EXTRA))
方法	`public void sendStepLenAndWeightToBLE(int height, int weight,int offScreenTime,int stepTask, boolean isRraisHandbrightScreenSwitchOpen,boolean isHighestRateOpen,int highestRate) {`**（修改）**
返回值	无
参数说明	身高(cm)、体重(kg)、灭屏时间(秒)
说明	设置 BEL 端步长；体重；灭屏时间；目标步数；抬手亮屏开关 true 为开，false 为关；最高心率提醒，true 为开，false 为关；最后一个参数为最高心率提醒的值。**提醒：修改身高体重后，需要同步一次计步数据，应用上的距离和卡路里才会按照新修改的身高体重进行计算并更新。**
方法	**public void** syncAllStepData()
返回值	无
参数说明	无
说明	同步计步数据**(连上设备后，请同步一次步数(实际是在设置时间后，同步步数)；同步完成前，请不要进行其他任何的通信工作)**
方法	**public void** syncAllSleepData()
返回值	无
参数说明	无
说明	同步睡眠数据**(同步完成前，请不要进行其他任何的通信工作)**
方法	**public void** findBand(**int** vibrationCount)
返回值	无
参数说明	vibrationCount 手环震动次数
说明	查找手环
方法	**public void** sendNameToBLE(String name)
返回值	无
参数说明	name 电话联系人姓名
说明	发送名字指令
方法	**public void** sendNumberToBLE(String number, **int** type)
返回值	无
参数说明	number 电话号码。type 区分来电还是短信的号码; GlobalVariable. PhoneTyp 代表来电号码，GlobalVariable.SmsType 代码来短信号码，只能是这两种情况
说明	发送号码指令(number 是对应 name 的号码，所以该方法要结合 sendNameToBLE 使用)
方法	**public void** sendIncallCommand(**int** vibrationCount)

返回值	无
参数说明	vibrationCount 来电震动次数
说明	发送来电指令
方法	**public void** sendOffHookCommand()
返回值	无
参数说明	无
说明	发送接听/挂断电话指令
方法	**public void** sendStopVibrationCommand()
返回值	无
参数说明	无
说明	发送停止震动指令(例如收到接听/挂断电话返回指令后，可发送停止震动指令停止震动)
方法	**public void** sendSmsCommand(**int** vibrationCount)
返回值	无
参数说明	vibrationCount 震动次数
说明	发送收到短信指令,
方法	**public void** sendQQWeChatTypeCommand(**int** type)
返回值	无
参数说明	GlobalVariable.QQType 为 QQ; GlobalVariable.WeChatType 为微信
说明	发送收到 QQ\微信信息指令
方法	**public void** sendQQWeChatVibrationCommand(**int** vibrationCount)
返回值	无
参数说明	vibrationCount 震动次数
说明	发送 QQ\微信震动次数指令(收到返回 QQ\微信信息指令后，再发送震动次数指令)
方法	**public void** sendToGetStepOrSleepStatus()
返回值	无
参数说明	无
说明	发送指令读取当前是计步还是睡眠状态(在 ICallback.OnResult 中返回状态，true 表示计步状态，false 表示睡眠状态)
方法	`public void sendToSetAlarmCommand(int whichClock, byte weekPeroid, int hour, int minute, boolean isOpen,int shakePeriod) {`（修改）
返回值	无
参数说明	whichClock：第几个闹钟， **GlobalVariable.FIRST_CLOCK 第一个闹钟**

	GlobalVariable.SECOND_CLOCK 第二个闹钟 **GlobalVariable.THIRD_CLOCK 第三个闹钟** weekPeroid：闹钟周期， **GlobalVariable.SUNDAY 星期天** **GlobalVariable.MONDAY 星期一** **GlobalVariable.TUESDAY 星期二** **GlobalVariable.WEDNESDAY 星期三** **GlobalVariable.THURSDAY 星期四** **GlobalVariable.FRIDAY 星期五** **GlobalVariable.SATURDAY 星期六** **GlobalVariable.EVERYDAY 每天** **weekPeroid=GlobalVariable.SUNDAY|GlobalVariable.WEDNESDAY|GlobalVariable.FRIDAY 表示设置星期天、星期三、星期五** hour：闹铃小时**(24 小时制，如下午 1 点，hour=13)** minute：闹铃分钟 Open：true 打开闹钟；false 关闭闹钟 shakePeriod:手环振动次数，之前直接写死 5 次，现在改为可设置次数。
说明	发送设置闹钟指令
方法	**public void** openShakeMode()
返回值	无
参数说明	无
说明	打开摇摇功能(之后发现设备被摇一摇时，会在 ICallback 中返回状态，ICallbackStatus.DISCOVERY_DEVICE_SHAKE)，常用于摇摇拍照等功能的实现。
方法	**public void** closeShakeMode()
返回值	无
参数说明	无
说明	关闭摇摇功能
方法	**public void** sendCalibratTimeoutToBLE()
返回值	无
参数说明	无
说明	发送抬手亮屏校准超时指令(部分手环才有该方法)
方法	**public void** startCalibrate()
返回值	无
参数说明	无
说明	发送抬手亮屏校准开始指令(部分手环才有该方法)

方法	**public void** deleteDevicesAllData()
返回值	无
参数说明	无
说明	清除设备所有数据，即设备恢复出厂设置
方法	**public void** syncAllRateData()
返回值	无
参数说明	无
说明	发送同步心率数据指令
方法	**public void** sendRateTestCommand(**int** flag)
返回值	无
参数说明	flag：开启或关闭心率测试标志 GlobalVariable.RATE_TEST_START 开始测试 GlobalVariable.RATE_TEST_STOP 停止测试
说明	发送心率测试开启/关闭指令
方法	**public void** sendSedentaryRemindCommand(**int** flag, **int** period)
返回值	无
参数说明	**flag**：开启或关闭心率测试标志 GlobalVariable.OPEN_SEDENTARY_REMIND 开始测试 GlobalVariable.CLOSE_SEDENTARY_REMIND 停止测试 **period**：提醒周期(单位：分钟)
说明	发送久坐提醒功能开启/关闭指令以及提醒周期
方法	**public void** queryDeviceFearture ()
返回值	无
参数说明	无
说明	查询设备升级属性 （升级前必须调用查询）
方法	**public void** sendUnitToBLE ()（删除） **public void** sendUnitAndHourFormatToBLE()（新）
返回值	无
参数说明	无
说明	公英制单位切换 ，**12、24 小时制（跟随手机时间制）**
方法	public void sendUnitAndHourFormatToBLE(int unitType, int hourFormat) {（新）
返回值	无
参数说明	unitType：GlobalVariable.UNIT_TYPE_METRICE 公制单位 GlobalVariable.UNIT_TYPE_IMPERIAL 英制单位

	hourFormat：GlobalVariable.HOUR_FORMAT_12　12小时制 GlobalVariable.HOUR_FORMAT_24　24小时制
说明	
方法	public void openBLEchannel()
返回值	无
参数说明	无
说明	打开 BLE 通道
方法	public void closeBLEchannel()
返回值	无
参数说明	无
说明	关闭 BLE 通道
方法	public void sendAPDUToBLE(byte[] data)
返回值	无
参数说明	data :发送的 APDU 指令
说明	发送 APDU 指令到 BLE
方法	public static byte[] hexString2Bytes(String stringBytes)
返回值	byte[]数组
参数说明	stringBytes :字符串转 byte[]数组
说明	将 16 进制数的字符串转 byte[]数组
方法	public void sendTextToBle(String body, int messageType) {
返回值	无
参数说明	body 推送的内容;messageType 消息的类型：GlobalVariable. TYPE_PHONE GlobalVariable. TYPE_QQ， GlobalVariable. TYPE_WECHAT ，GlobalVariable. SMS
说明	推送的消息的内容到 BLE（参考 mianActivity 中调用）
方法	public void sendKeyOpenDynamicOrStaticRate(int status) {
返回值	无
参数说明	Status：静态心率或动态心率 GlobalVariable. RATE_STATIC GlobalVariable. RATE_DYNAMIC
说明	切换心率状态
方法	public void sendKeyToReadUV() {
返回值	无
参数说明	无
说明	发送指令到 ble 读取 UV 紫外线（支持 UV 紫外线的固件才有返回值）

方法	`public void sendKeyReadIsHasContentPush () {`
返回值	无
参数说明	无
说明	发送指令到 ble 查询是否支持内容推送（支持的固件才有返回值）
方法	`public void sendToQueryPasswardStatus() {`
返回值	无
参数说明	无
说明	发送指令查询密码状态
方法	`public void sendToSetOrInputPassward(String password,int type) {`
返回值	无
参数说明	Password ： 密码，4 位数字密码；type：类型， GlobalVariable. PASSWORD_TYPE_SET 设置密码 GlobalVariable.PASSWORD_TYPE_INPUT 输入密码 GlobalVariable.PASSWORD_TYPE_INPUT 重新输入密码
说明	
方法	`public void syncAllSwimData() {`
返回值	无
参数说明	无
说明	同步游泳数据
方法	`public void syncAllBloodPressureData() {`
返回值	无
参数说明	无
说明	同步血压数据
方法	`public void sendBloodPressureTestCommand(int flag){`
返回值	无
参数说明	flag：开启或关闭血压测试标志 GlobalVariable.BLOOD_PRESSURE_TEST_START 开启 GlobalVariable.BLOOD_PRESSURE_TEST_STOP 关闭
说明	开始或关闭测试血压
方法	`public void sendDisturbToBle(boolean isMessageOn, boolean isMotorOn, boolean isScreenOn, boolean disturbTimeSwitch, int from_time_hour, int from_time_minute, int from_time_minute, int to_time_minute) {`
返回值	无
参数说明	isMessageOn 消息勿扰，true 为开启，false 为关闭

	isMotorOn 马达勿扰，true 为开启，false 为关闭 isScreenOn 屏幕勿扰，true 为开启，false 为关闭 disturbTimeSwitch 勿扰时间，true 为开启，false 为关闭 from_time_hour 开始时间 小时 from_time_minute 开始时间 分钟 from_time_minute 结束时间 小时 to_time_minute 结束时间 分钟
说明	
方法	`public void sendToControlHVScreen(int type){`
返回值	无
参数说明	Type 横竖屏的状态， GlobalVariable.SCREEN_VERTICAL 竖屏 GlobalVariable.SCREEN_HORIZONTAL 横屏
说明	传入横屏或竖屏的参数，控制手环切换屏幕方向
方法	`public void syncWeatherToBLE(Context context ,String city){`
返回值	无
参数说明	context 上下文； City 城市名称，通过 amapLocation.getCity()获取得到的城市名称直接传进去。国内的，如深圳市，广州市，有“市”字不返回天气信息的话，可以尝试去掉“市”字
说明	支持国内国外。此操作有点耗时，sdk 内部访问天气网获取天气后，直接同步到手环
方法	`public void syncWeatherToBLE(String weatherCode,int tmpCurrent,int tmpTodayMax,int tmpTodayMin, int pm25,int aqi,String tomorrowWeatherCode, int tmpTomorrowMax,int tmpTomorrowMin){` **(新)**
返回值	无
参数说明	weatherCode 今天天气代码 ,今天天气状况代码 tmpCurrent 当前温度 tmpTodayMax 今天最高温度 tmpTodayMin 今天最低温度 pm25 今天pm25 aqi 今天aqi（空气质量指数） tomorrowWeatherCode 明天天气代码 ,明天天气状况代码

	tmpTomorrowMax 明天最高温 tmpTomorrowMin 明天最低温
说明	同步天气到 ble，传入的天气参数请开发者自己获取对应的天气信息传入
方法	public void sendKeySetTimingTestRate(boolean isOpen ,int timedTestTime) {(新)
返回值	无
参数说明	isOpen： true 为开，false 为关 timedTestTime 定时测试时间周期（单位：分钟）
说明	定时检测心率（定时测试心率）
方法	public void sendIbeaconSetCommand(String hexString, int commandType) {(新)// 调用后在接口Icallback onIbeaconWriteCallback回调，请参考MainActivity
返回值	无
参数说明	hexString:传入设置的数据 注意：UUID的数据长度固定为16byte的ASIIC； major和minor固定长度为2byte的数字。 Device name的长度范必须大于0小于14byte的ASIIC commandType： GlobalVariable. IBEACON_TYPE_UUID // Ibeacon 指令类型,设置UUID GlobalVariable. IBEACON_TYPE_MAJOR// Ibeacon 指令类型,设置major GlobalVariable. IBEACON_TYPE_MINOR// Ibeacon 指令类型,设置minor GlobalVariable. IBEACON_TYPE_DEVICE_NAME// Ibeacon 指令类型,设置蓝牙device name
说明	Ibeacon 功能设置
方法	public void sendIbeaconGetCommand(int commandType) {(新) // 调用后在接口Icallback onIbeaconWriteCallback回调，请参考MainActivity
返回值	无
参数说明	commandType： commandType： GlobalVariable. IBEACON_TYPE_UUID // Ibeacon 指令类型,获取UUID GlobalVariable. IBEACON_TYPE_MAJOR// Ibeacon 指令类型,获取major GlobalVariable. IBEACON_TYPE_MINOR// Ibeacon 指令类型,获取minor GlobalVariable. IBEACON_TYPE_DEVICE_NAME// Ibeacon 指令类型,获取蓝牙device name

说明	Ibeacon 功能查询
方法	`public void queryDialMode() {(新)`
返回值	无
参数说明	无
说明	查询表盘方式
方法	`public void controlDialSwitchAandLeftRightHand(int leftRightHand, int dialType) {(新)`
返回值	无

参数说明	`leftRightHand:`
	GlobalVariable. LEFT_HAND_WEAR 左手佩戴
	GlobalVariable. RIGHT_HAND_WEAR 右手佩戴
	GlobalVariable. NOT_SET_UP 不设置，保持上一次
	`dialType:`
	GlobalVariable. SHOW_HORIZONTAL_SCREEN 显示横屏
	GlobalVariable. SHOW_VERTICAL_ENGLISH_SCREEN 显示竖屏英文界面
	GlobalVariable. SHOW_VERTICAL_CHINESE_SCREEN 显示竖屏中文界面
	GlobalVariable. NOT_SET_UP 不设置，保持上一次

说明	控制表盘切换和左右手切换
方法	`public void syncWeatherToBLEForXiaoYang(SevenDayWeatherInfo info) {(新)`
返回值	无
参数说明	请参考 `SevenDayWeatherInfo`
说明	同步最近七天天气,

2.4.DataProcessing 数据处理类

方法	**public static DataProcessing getInstance(Context context)**
返回值	DataProcessing
参数说明	Context 上下文
说明	实例化，返回该类实例
方法	public static DataProcessing getInstance()
返回值	DataProcessing
参数说明	无
说明	获取该类实例（前提是已经实例化过）

方法	**public void setOnStepChangeListener(StepChangeListener listener)**
返回值	无
参数说明	StepChangeListener l 步数变化监听
说明	监听步数变化
方法	**public void setOnSleepChangeListener(SleepChangeListener listener)**
返回值	无
参数说明	listener 睡眠监听
说明	监听睡眠
方法	`Public void setOnBloodPressureListener(BloodPressureChangeListener listener) {`
返回值	无
参数说明	Listener 血压监听
说明	监听血压实时变化

2.5.UTESQLOperate 数据库操作类

方法	**public** UTESQLOperate(Context context)**（2.2.0 版本删除）**
返回值	无
参数说明	Context 上下文
说明	实例化
方法	`public static UTESQLOperate getInstance(Context context) {`**（新）**
返回值	无
参数说明	Context 上下文
说明	实例化(单例)
方法	**public void** updateStepSQL()
返回值	无
参数说明	无
说明	新一天初始化计步数据库
方法	**public int** queryStepDate(String queryDate)
返回值	int
参数说明	String queryDate 查询的日期 如 20150603
说明	查询一天的总步数
方法	**public** StepInfo queryStepInfo(String queryDate)
返回值	StepInfo 返回步数、距离、卡路里的集合

参数说明	queryDate 查询的日期
说明	查询一天的步数、距离、卡路里
方法	**public List<StepOneHourInfo> queryOneHourStepSQL(String calendar)**
返回值	List<StepOneHourInfo> ，返回各小时和步数集合
参数说明	Calendar 查询的日期
说明	查询某一天各小时步数(请参考 **StepOneHourInfo 类**)
方法	**public int** querySleepDate(String queryDate)
返回值	int 返回睡眠时间，单位为分钟
参数说明	queryDate 查询的日期
说明	查询一天的睡眠总时间
方法	**public SleepTimeInfo querySleepInfo(String calendar)** **(修改)**
返回值	SleepTimeInfo 返回某天睡眠时间、深睡、浅睡、清醒时间、睡眠状态、睡眠状态下持续时间、睡眠状态结束时间点集合
参数说明	calendar 某天日期（日期格式：如“**20150101**”）
说明	查询一天的睡眠详情(请参考 **SleepTimeInfo 类**)
方法	**public RateOneDayInfo queryRateOneDayMainInfo(String calendar)**
返回值	RateOneDayInfo 返回某天最后一次测试的心率值、最低心率、平均心率、最高心率值的集合 使用到 RateOneDayInfo 类中的方法有： **getCurrentRate()** 该天最后一次测试的心率值 **getLowestRate()** 该天最低心率值 **getVerageRate()** 该天平均心率值 **getHighestRate()** 该天最高心率值
参数说明	calendar 某天日期（日期格式：如“**20150101**”）
说明	查询一天主要几个心率值 (请参考 **RateOneDayInfo 类**)
方法	**public List<RateOneDayInfo> queryRateOneDayDetailInfo(String calendar)**
返回值	**List<RateOneDayInfo>**返回某天各个时间段测试的心率值的集合 使用到 RateOneDayInfo 类中的方法有： **getTime()** 该次心率的测试时间（分钟） **getRate()** 改时间测试得的心率值
参数说明	calendar 某天日期（日期格式：如“**20150101**”）
说明	查询一天的心率详情(请参考 **RateOneDayInfo 类**)
方法	**public void** updateRateSQL()
返回值	无
参数说明	无
说明	新一天初始化心率数据库

方法	public void saveDynamicRate(String calendar,String curTime, int curRate, int testTime, int average, int max, int min) {
返回值	无
参数说明	Calendar 测试心率的日期，格式 yyyy/MM/dd；curTime 测试心率时的时间（格式如：08:13:59）；curRate 当前心率值；average 此次测试心率的多个心率值的平均值；max 此次测试心率的多个心率值中的最高心率值；min 此次测试心率的多个心率值中的最低心率值；
说明	保存某次测试心率的详细参数
方法	public ArrayList<RateDynamicHistoryInfo> queryDynamicRate(String calendar) {
返回值	ArrayList < RateDynamicHistoryInfo >返回某天各个时间段测试的心率值的集合 使用到 RateDynamicHistoryInfo 类中的方法有： **getCalendar()** 该次心率的测试日期 **getTheTime()**该次心率的测试时间 getTestTime()该次心率的测试时长，单位秒 getCurrentRate()当前心率值 getAverageRate()该次心率的平均值 getHighestRate() getLowestRate()
参数说明	calendar 某天日期（日期格式：如“**2015/01/01**”）
说明	通过日期查询某天所有心率的测试情况
方法	`public List<BPVOneDayInfo> queryBloodPressureOneDayInfo(String calendar) {`
返回值	List<BPVOneDayInfo> 使用到 BPVOneDayInfo 类中的方法有： getBloodPressureTime()该次血压的测试时间 getHightBloodPressure()当前血压的高压值 getLowBloodPressure()当前血压的低压值
参数说明	calendar 某天日期（日期格式：如“**20170101**”）
说明	通过日期查询某天所有血压的测试情况

2.6.PedometerUtils 工具类

方法	public PedometerUtils(Context context)
返回值	无
参数说明	Context 上下文
说明	实例化。

2.7.StepChangeListener 步数变化监听接口

方法	**public void** onStepChange(**int** steps,**float** distance,**int** calories)
返回值	无
参数说明	int steps,float distance,int calories
说明	Interface 计步数据变化接口,返回步数、距离、卡路里。 使用时，需要提前调用 DataProcessing 中的方法 setOnStepChangeListener(StepChangeListener listener)设置监听

2.8.SleepChangeListener 睡眠变化监听接口

方法	**public void** onSleepChange()
返回值	无
参数说明	无
说明	Interface 睡眠数据变化接口。使用时，需要提前调用 DataProcessing 中的方法 setOnSleepChangeListener(SleepChangeListener listener)设置监听，再调用 querySleepInfo 设置更新睡眠信息

2.9.RateChangeListener 心率变化监听接口

方法	**public void** onRateChange(**int** rateValue, **int** status)
返回值	无
参数说明	rateValue：心率值 status：当前心率值的状态，仅两种 GlobalVariable.RATE_TESTING 测试中 GlobalVariable.RATE_TEST_FINISH 测试完成，说明当前值为测试最终结果
说明	Interface 心率数据变化接口，返回当前测试的实时心率值和心率值的状态。 使用时，需要提前调用 DataProcessing 中的方法 setOnRateListener(RateChangeListener listener)设置监听

2.10.ICallback 与设备通信回调类

方法	**public void** OnResult(**boolean** result, **int** status);
返回值	无
参数说明	result：true 通信完成 status：如 ICallbackStatus 类中描述
说明	ICallback 回调结果
方法	`public void OnDataResult(boolean result, int status,byte[] data);`
返回值	无
参数说明	result：true 通信完成 status：如 ICallbackStatus 类中描述 data ：BLE 返回的数据
说明	`OnDataResult` 交通卡接口回调结果

方法	public void onCharacteristicWriteCallback(int status); (新)
返回值	无
参数说明	status = 0 为写入成功，其他或无回调表示失败
说明	写入操作状态的系统回调，status = 0 为写入成功，其他或无回调表示失败
方法	public void onIbeaconWriteCallback(boolean result, int ibeaconSetOrGet,int ibeaconType,String data); (新) 请参考MainActivity使用
返回值	无
参数说明	Result 状态 ibeaconSetOrGet：设置或获取 GlobalVariable. IBEACON_SET //设置 GlobalVariable. IBEACON_GET//获取 ibeaconType：设置或获取的类型 GlobalVariable. IBEACON_TYPE_UUID // Ibeacon 指令类型，UUID GlobalVariable. IBEACON_TYPE_MAJOR// Ibeacon 指令类型，major GlobalVariable. IBEACON_TYPE_MINOR// Ibeacon 指令类型，minor GlobalVariable. IBEACON_TYPE_DEVICE_NAME// Ibeacon 指令类型,蓝牙 device name data 设置或返回的数据
说明	Ibeacon 功能设置和读取回调
方法	public void onQueryDialModeCallback(boolean result, int screenWith,int screenWith,int screenCount); (新)
返回值	无
参数说明	Result 状态 screenWith 宽 screenWith 高 screenCount 表盘个数
说明	查询表盘方式的回调，请参考 MainActivity 使用
方法	public void onControlDialCallback(boolean result, int leftRightHand,int dialType); (新)
返回值	无
参数说明	Result 状态

	leftRightHand： GlobalVariable. LEFT_HAND_WEAR 左手佩戴 GlobalVariable. RIGHT_HAND_WEAR　右手佩戴 GlobalVariable. NOT_SET_UP　不设置，保持上一次 dialType： GlobalVariable. SHOW_HORIZONTAL_SCREEN　显示横屏 GlobalVariable. SHOW_VERTICAL_ENGLISH_SCREEN 显示竖屏英文界面 GlobalVariable. SHOW_VERTICAL_CHINESE_SCREEN　显示竖屏中文界面 GlobalVariable. NOT_SET_UP　不设置，保持上一次
说明	控制表盘切换和左右手切换回调，请参考 MainActivity 使用

2.11.ICallbackStatus 与设备通信交互结果的回调状态

```
public class ICallbackStatus {
    public static int REAL_TIME_STEP = 0;//当前属于实时步数操作
    public static int OFFLINE_STEP_SYNCING = 1;//离线步数同步中
    public static int OFFLINE_STEP_SYNC_OK = 2;//离线步数同步完成
    public static int REAL_TIME_SLEEP = 3;//当前属于实时睡眠操作
    public static int OFFLINE_SLEEP_SYNCING = 4;//离线睡眠同步中
    public static int OFFLINE_SLEEP_SYNC_OK = 5;//离线睡眠同步完成
    public static int SYNC_TIME_OK = 6;//设置时间操作完成
    public static int GET_BLE_VERSION_OK = 7;//获取设备版本号操作完成
    public static int SET_STEPLEN_WEIGHT_OK = 8;//设置身高体重操作完成
    public static int GET_SPORT_STATUS_OK = 9;//获取运动状态(计步或睡眠)操作完成
    public static int GET_BLE_BATTERY_OK = 10;//获取电量操作完成
    public static int PRESS_DEVICE_BUTTON = 11;//按下设备按钮操作

    public static int SEND_INCALL_OR_SMS_NAME_OK = 12;//发送来电或短信用户名字操作完成
    public static int SEND_INCALL_NUMBER_OK = 13;//发送来电号码操作完成
    public static int SEND_SMS_NUMBER_OK = 14;//发送短信号码操作完成
    public static final int SEND_OFFHOOK_OK = 15;//挂断/接听电话操作完成
    public static final int SEND_QQ_COMMAND_OK = 16;//发送QQ指令操作完成
    public static final int SEND_WECHAT_COMMAND_OK = 17;//发送微信指令操作完成

    public static final int OPERATION_FAILE = 18;//操作失败

    public static final int DISCONNECT_STATUS = 19;//未连接/连接失败/断开连接
    public static final int CONNECTED_STATUS = 20;//已连接

    public static final int DISCOVERY_DEVICE_SHAKE = 21;//发现设备摇一摇(常用于摇摇拍照)
```

```
    public static final int OFFLINE_RATE_SYNCING = 22;//离线心率同步中
    public static final int OFFLINE_RATE_SYNC_OK = 23;//离线心率同步完成
    public static final RATE_TEST_FAILE = 24;//心率测试失败

    public static final int SEDENTARY_REMIND_OPEN = 25;//久坐提醒已打开
    public static final int SEDENTARY_REMIND_CLOSE = 26;//久坐提醒已关闭

    public static final int SET_METRICE_OK = 27;//设置公制单位成功
    public static final int SET_INCH_OK = 28;//设置英制单位成功
    public static final int SET_FIRST_ALARM_CLOCK_OK = 29;//设置第1个闹钟OK
    public static final int SET_SECOND_ALARM_CLOCK_OK = 30;//设置第2个闹钟OK
    public static final int SET_THIRD_ALARM_CLOCK_OK = 31;//设置第 3 个闹钟 OK

//
    public static final int OPEN_CHANNEL_OK = 32;//打开通道OK
    public static final int CLOSE_CHANNEL_OK = 33;//关闭通道OK
    public static final int BLE_DATA_BACK_OK = 34; //测试通道 OK，通道正常

    public static final int GET_UV_BACK_OK = 35; //读取UV紫外线OK
    public static final int SEND_QQ_WHAT_SMS_CONTENT_OK = 36; //发送QQ、微信、短信内容OK
    public static final int SEND_PHONE_NAME_NUMBER_OK = 37; //发送来电名字和号码 OK
    public static final int IS_SUPPOERT_PUSH_CONTENT = 38; //调用
sendKeyReadIsHasContentPush查询是否支持内容推送后，如果有这个返回值，则表示支持内容推送

    public static final int BLE_SERVICE_START_OK = 39; //BluetoothLeService 服务开启完
成后回调.前提是：需在 BluetoothLeService 服务开启完成前设置监听，否则无回调
    public static final int PASSWORD_SET = 40; //没设置过密码，请设置 4 位数字密码
    public static final int PASSWORD_INPUT = 41; //已设置过密码，请输入已设置的 4 位数字
密码
    public static final int PASSWORD_AUTHENTICATION_OK = 42; //验证成功或者设置密码成
功
    public static final int PASSWORD_INPUT_AGAIN = 43; //验证失败或者设置密码失败，请重
新输入 4 位数字密码，如果已设置过密码，请输入已设置的密码
    public static final int OFFLINE_SWIM_SYNCING = 44;//游泳数据同步中
    public static final int OFFLINE_SWIM_SYNC_OK = 45;//游泳数据同步完成
    public static final int OFFLINE_BLOOD_PRESSURE_SYNCING = 46;//血压数据同步中
    public static final int OFFLINE_BLOOD_PRESSURE_SYNC_OK = 47;//血压数据同步完成
    public static final int BLOOD_PRESSURE_TEST_TIME_OUT = 48;//血压检测超时
    public static final int BLOOD_PRESSURE_TEST_ERROR = 49;//血压检测错误
    public static final int BLOOD_PRESSURE_TEST_STAT = 50;//血压检测开始
    public static final int SEVEN_DAY_WEATHER_SYNC_SUCCESS = 51;//7天天气同步完成

}
```


2.12.BLEServiceOperate 蓝牙通信服务操作类

方法	public static BLEServiceOperate.getInstance(Context context)
返回值	BLEServiceOperate 接口对象
参数说明	context 内容
说明	实例化 BLEServiceOperate 接口对象
方法	**public** BluetoothLeService getBleService()
返回值	BluetoothLeService 对象
参数说明	无
说明	获取蓝牙通信服务
方法	**public boolean** isSupportBle4_0()
返回值	boolean
参数说明	无
说明	是否支持蓝牙 4.0
方法	**public boolean** isBleEnabled()
返回值	boolean 对象
参数说明	无
说明	蓝牙是否已经开启
方法	public void setDeviceScanListener(DeviceScanInterfacer l)
返回值	无
参数说明	DeviceScanInterfacer 扫描监听类
说明	设置扫描监听(只有设置监听后，才有扫描回调结果)
方法	**public void** startLeScan()
返回值	无
参数说明	无
说明	开始扫描设备
方法	**public void** stopLeScan()
返回值	无
参数说明	无
说明	停止扫描设备
方法	**public boolean** connect(String address)
返回值	boolean
参数说明	设备地址
说明	连接设备

方法	**public void** disConnect()
返回值	无
参数说明	无
说明	断开设备
方法	**public void** unBindService()
返回值	无
参数说明	无
说明	解绑服务

2.13.BluetoothLeService 蓝牙通信服务类

方法	**public void** setRssiHandler(Handler handler)
返回值	无
参数说明	handler：线损值 handler 对象
说明	设置线损值 handler，以便线损值的监听（线损值的 message 为 GlobalVariable.GET_RSSI_MSG）
方法	**public void** readRssi()
返回值	无
参数说明	无
说明	读取线损值，每读一次，即监听一次

2.14.StepOneHourInfo 各小时和步数集合类

备注：结合 **List<StepOneHourInfo> queryOneHourStepSQL(String calendar)**使用	
方法	**public int** getTime()
返回值	时间(小时)
参数说明	无
说明	获取某小时
方法	**public int** getStep()
返回值	步数
参数说明	无
说明	获取某小时的步数

2.15.StepInfo 步数信息集合类

备注：结合 StepInfo queryStepInfo(String queryDate)使用	
方法	**public int** getStep()
返回值	步数
参数说明	无
说明	获取某天步数
方法	**public int** getCalories()
返回值	卡路里

参数说明	无
说明	获取某天卡路里
方法	**public float** getDistance()
返回值	距离
参数说明	无
说明	获取某天运动距离

2.16.SleepTimeInfo 睡眠信息集合类 **(修改)**

备注：结合 **SleepTimeInfo querySleepInfo(String calendar)**使用	
方法	**public int** getSleepTotalTime()
返回值	睡眠总时间（分钟，24 小时制）
参数说明	无
说明	获取某天的睡眠总时间
方法	**public int** getDeepTime()
返回值	深睡眠总时间（分钟，24 小时制）
参数说明	无
说明	获取某天深睡眠的总时间
方法	**public int** getLightTime()
返回值	浅睡眠总时间（分钟，24 小时制）
参数说明	无
说明	获取某天浅睡眠的总时间
方法	**public int** getAwakeTime () **(新)**
返回值	清醒时间
参数说明	无
说明	获取某天睡眠的清醒时间
方法	**public int** getAwakeCount ()
返回值	清醒次数
参数说明	无
说明	获取某天睡眠的清醒次数
方法	**public int** getBeginTime () **(新)**
返回值	入睡时间
参数说明	无
说明	获取某天睡眠的入睡时间
方法	**public int** getEndTime() **(新)**
返回值	起床时间
参数说明	无

说明	获取某天睡眠的起床时间
方法	**public int** getSleepTotalTime **(新)**
返回值	当天睡眠总时间
参数说明	无
说明	获取某天睡眠总时间
方法	**public int**[] getSleepStatueArray()
返回值	整形数组
参数说明	无
说明	获取某天各时段的睡眠状态(深睡、浅睡、清醒三种状态) 深睡状态：0 浅睡状态：1 清醒状态：2
方法	**public int**[] getDurationTimeArray()
返回值	整形数组
参数说明	无
说明	获取某天各睡眠状态的持续时间（分钟，24 小时制）
方法	**public int**[] getTimePointArray()
返回值	整形数组
参数说明	无
说明	获取某天个睡眠状态的结束时间点（分钟，24 小时制）
备注：以上三个组数是一一对应的，即浅睡眠，持续了 30 分钟，在 252 分钟(4 时 12 分)结束浅睡眠(浅睡眠的起始时间=252-30)，转而切换到另一种睡眠状态	

2.17.RateOneDayInfo 心率信息集合类

备注：结合 **RateOneDayInfo queryRateOneDayMainInfo(String calendar)或 List<RateOneDayInfo> queryRateOneDayDetailInfo(String calendar)** 使用	
方法	**public int** getTime()
返回值	时间（分钟）
参数说明	无
说明	获取某天某次心率的测试时间
方法	**public int** getRate()
返回值	某次心率值
参数说明	无
说明	获取某天某时间测试的心率值
方法	**public int** getVerageRate()
返回值	平均心率值
参数说明	无

说明	获取某天平均心率值
方法	**public int** getLowestRate()
返回值	最低心率值
参数说明	无
说明	获取某天最低心率值
方法	**public int** getHighestRate()
返回值	最高心率值
参数说明	无
说明	获取某天最高心率值
方法	**public int** getCurrentRate()
返回值	当前心率值
参数说明	无
说明	获取某天最后一次测试的心率值
备注：以上几个方法虽同在 RateOneDayInfo 类中，但应用场景不同： getTime()和 getRate()常配合用在了解某天各时间段心率测试详情； getVerageRate()、getLowestRate()、getHighestRate()、getCurrentRate()常配合用在了解某天心率从测试的总体情况。	

2.18.CalendarUtils 日期类(格式：yyyymmdd)

方法	**public static** String getCalendar(**int** padding)
返回值	String
参数说明	padding 与今天相比的相对值 padding=0，代表今天 padding=-1，代表昨天 padding=-2，代表前天 以此类推
说明	获取日期，如：今天=getCalendar(0)，也可写成”20150101”

2.19.GetFunctionList 查询支持功能

方法	`public static boolean isSupportFunction(Context context, int functionType) {`
返回值	**boolean**
参数说明	Context 上下文 functionType： GlobalVariable.IS_SUPPORT_PASS_WORD_PAIR 返回 true 表示支持密码配对，false 表示不支持 GlobalVariable.IS_SUPPORT_MULTIPLE_LANGUAGE 返回 true 表示支持全字库，false 表示不支持 GlobalVariable.IS_SUPPORT_Do_NOT_DISTURB 返回 true 表示支持勿扰模式，false 表示不支持 GlobalVariable.IS_SUPPORT_SWIMMING 返回 true 表示支持游泳功能，false 表示不

	支持 GlobalVariable.IS_SUPPORT_HOR_VER 返回 true 表示支持横竖屏切换，false 表示不支持 GlobalVariable.IS_SUPPORT_RAISE_HAND_BRIGHT 返回 true 表示支持抬手亮屏，false 表示不支持 GlobalVariable. IS_SUPPORT_DIAL_SWITCH// 返回true表示支持表盘切换和左右手切换功能 GlobalVariable. IS_SUPPORT_IBEACON 返回 true 支持 ibeacon 这个功能，false 表示不支持 GlobalVariable. IS_SUPPORT_SEVEN_DAYS_WEATHER 返回 true 支持 七天天气，false 表示不支持
说明	获取日期，如：今天=getCalendar(0)，也可写成”20150101”

2.20. SevenDayWeatherInfo 7 天天气信息集合类**（新）,天气气候** WeatherCode

请参考文档最后的图 1.

方法	public SevenDayWeatherInfo(String cityName, String todayWeatherCode,int todayTmpCurrent, int todayTmpMax, int todayTmpMin,int todayPm25, int todayAqi, String secondDayWeatherCode,int secondDayTmpMax, int secondDayTmpMin,String thirdDayWeatherCode, int thirdDayTmpMax, int thirdDayTmpMin,String fourthDayWeatherCode, int fourthDayTmpMax,int fourthDayTmpMin, String fifthDayWeatherCode,int fifthDayTmpMax, int fifthDayTmpMin, String sixthDayWeatherCode,int sixthDayTmpMax, int sixthDayTmpMin,String seventhDayWeatherCode, int seventhDayTmpMax,int seventhDayTmpMin) {
	setCityName(cityName);
	setTodayWeatherCode(todayWeatherCode);
	setTodayTmpCurrent(todayTmpCurrent);
	setTodayTmpMax(todayTmpMax);
	setTodayTmpMin(todayTmpMin);
	setTodayPm25(todayPm25);
	setTodayAqi(todayAqi);
	
	setSecondDayWeatherCode(secondDayWeatherCode);
	setSecondDayTmpMax(secondDayTmpMax);
	setSecondDayTmpMin(secondDayTmpMin);
	
	setThirdDayWeatherCode(thirdDayWeatherCode);
	setThirdDayTmpMax(thirdDayTmpMax);

	setThirdDayTmpMin(thirdDayTmpMin);
	
	setFourthDayWeatherCode(fourthDayWeatherCode);
	setFourthDayTmpMax(fourthDayTmpMax);
	setFourthDayTmpMin(fourthDayTmpMin);
	
	setFifthDayWeatherCode(fifthDayWeatherCode);
	setFifthDayTmpMax(fifthDayTmpMax);
	setFifthDayTmpMin(fifthDayTmpMin);
	
	setSixthDayWeatherCode(sixthDayWeatherCode);
	setSixthDayTmpMax(sixthDayTmpMax);
	setSixthDayTmpMin(sixthDayTmpMin);
	
	setSeventhDayWeatherCode(seventhDayWeatherCode);
	setSeventhDayTmpMax(seventhDayTmpMax);
	setSeventhDayTmpMin(seventhDayTmpMin);
	
	}

返回值	SevenDayWeatherInfo　7天天气集合
参数说明	7天天气信息
说明	构造方法
方法	public void setCityName(String cityName) {
返回值	无
参数说明	城市名称，城市只支持中文，最多 4 个字
说明	设置城市名称
方法	public void setTodayWeatherCode(String todayWeatherCode) {
返回值	无
参数说明	气候
说明	设置今天天气气候
方法	public void setTodayTmpCurrent(int todayTmpCurrent) {
返回值	无
参数说明	当前温度
说明	设置当前温度

方法	`public void setTodayTmpMax(int todayTmpMax) {`
返回值	无
参数说明	今天最高温度
说明	设置今天最高温度
方法	`public void setTodayTmpMin(int todayTmpMin) {`
返回值	无
参数说明	今天最低温度
说明	设置今天最低温度
方法	`public void setTodayPm25(int todayPm25) {`
返回值	无
参数说明	今天 pm2.5
说明	设置今天 pm2.5
方法	`public void setTodayAqi(int todayAqi) {`
返回值	无
参数说明	今天 aqi
说明	设置今天 aqi
方法	`public void setSecondDayWeatherCode(String secondDayWeatherCode) {//设置明天天气`
返回值	无
参数说明	明天气候
说明	设置明天气候
方法	`public void setSecondDayTmpMax(int secondDayTmpMax) {//设置明天最高温度`
返回值	无
参数说明	明天最高温度
说明	设置明天最高温度
方法	`public void setSecondDayTmpMin(int secondDayTmpMin) {//设置明天最低温`
返回值	无
参数说明	明天最低温度

说明	设置明天最低温度
	public void setThirdDayWeatherCode(String thirdDayWeatherCode) {//设置后天气候
	public void setThirdDayTmpMax(int thirdDayTmpMax) {//设置后天最高温
	public void setThirdDayTmpMin(int thirdDayTmpMin) {//设置后天最高温
	public void setFourthDayWeatherCode(String fourthDayWeatherCode) {////设置大后天最高温（第四天）
	public void setFourthDayTmpMax(int fourthDayTmpMax) {（第四天）
	public void setFourthDayTmpMin(int fourthDayTmpMin) {//第四天
	public void setFifthDayWeatherCode(String fifthDayWeatherCode) {//第5天
	public void setFifthDayTmpMax(int fifthDayTmpMax) {

	//第 5 天
	public void setFifthDayTmpMin(int fifthDayTmpMin) { //第 5 天
	public void setSixthDayWeatherCode(String sixthDayWeatherCode) {//第6天
	public void setSixthDayTmpMax(int sixthDayTmpMax) { //第 6 天
	public void setSixthDayTmpMin(int sixthDayTmpMin) { //第 6 天
	public void setSeventhDayWeatherCode(String seventhDayWeatherCode) {//第7天
	public void setSeventhDayTmpMax(int seventhDayTmpMax) {//第7天
	public void setSeventhDayTmpMin(int seventhDayTmpMin) {//第7天

3.变量说明

3.1.UUIDUtils

```
public class UUIDUtils {
    /*
     * 用于手机 apk 端写命令到蓝牙端的 UUID
     */
    public static final String ONLY_WRITE_UUID = "xxxxxx";
    /*
     * 用于蓝牙端返回数据到手机 apk 端  的 UUID
     */
    public static final String ONLY_READ_UUID = "xxxxxx";
    /*
     * 服务 UUID
     */
    public static final String SIMPLE_SERVICE_UUID = "xxxxxx";
    public static final String BATTERY_LEVELE = "xxxxxx";
}
```

3.2.GlobalVariable

```
public class GlobalVariable {
    // 轻型数据库名字
    public static String SettingSP = "SettingSP";
    //蓝牙连接状态 ，SharedPreferences KEY值
    public static final String BLE_CONNECTED_SP = "ble_connected";
    // 上一次连接的蓝牙地址 ，SharedPreferences KEY值
    public static final String LAST_CLICK_DEVICE_ADDRESS_SP = "last_click_device_address";
    //当天已保存的步数，SharedPreferences KEY值
    public static final String CURRENT_DAY_SAVED_STEP_COUNT = "current_day_saved_step_count";
    // 今天的步数，临时存储而已方便更新今天界面而已，步数一更新就变了，SharedPreferences KEY值
    public static final String YC_PED_CURRENT_HOUR_STEPS_SP = "current_hour_steps";
    //今天的步数，临时存储而已方便更新今天界面而已，步数一更新就变了，SharedPreferences KEY值
    public static final String YC_PED_STEPS_SP = "steps";
    //没用到，SharedPreferences KEY值
    public static final String YC_PED_PACE_SP = "pace";
    // 今天的路程，临时存储而已方便更新今天界面而已，步数一更新就变了，SharedPreferences KEY值
    public static final String YC_PED_DISTANCE_SP = "distance";
    //没用到，SharedPreferences KEY值
    public static final String YC_PED_SPEED_SP = "speed";
    // 今天的卡路里，临时存储而已方便更新今天界面而已，步数一更新就变了，SharedPreferences KEY
值
    public static final String YC_PED_CALORIES_SP = "calories";
    //当前小时的步数，如当前时间是10：42分，则表示10：00~10：42分的步数，SharedPreferences KEY
值
    public static final String YC_PED_UNFINISH_HOUR_STEP_SP = "unfinish_hour_step";
    // 当前小时值，SharedPreferences KEY值
    public static final String YC_PED_UNFINISH_HOUR_VALUE_SP = "unfinish_hour_value";
    // 实时计步时，上一条计步数据来时的步数，SharedPreferences KEY值
    public static final String YC_PED_LAST_HOUR_STEP_SP = "last_hour_step";
    //实时计步时，上一条计步数据来时的小时数，SharedPreferences KEY值
    public static final String YC_PED_LAST_HOUR_VALUE_SP = "last_hour_value";
    //设置BLE步长和体重的广播ACTION
    public static final String SET_BLE_STEP_LENGTH_WEIGHT = "set_ble_step_length_weight";
    public static final String BLE_DISCONNECT_BINDDEVICE_ACTION =
"ble_disconnect_binddevice_action";


    public static final String CIRCLE_BTN_TYPE = "circle_btn_type";
    // 个人信息      年龄，SharedPreferences KEY值
    public static final String PERSONAGE_AGE = "personage_age_sp";
    //个人信息      步长，SharedPreferences KEY值
    public static final String PERSONAGE_STEP_LENGTH = "step_length";
    // 个人信息      身高，SharedPreferences KEY值
```

```
    public static final String PERSONAGE_HEIGHT = "personage_height";
    //个人信息      体重，SharedPreferences KEY值
    public static final String PERSONAGE_WEIGHT = "body_weight";
    //上一次同步计步数据结束帧的日期，SharedPreferences KEY值
    public static final String B2FD_CALENDAR_SP = "b2fd_calendar_sp";
    //上一次同步睡眠数据结束帧的日期，SharedPreferences KEY值
    public static final String B3FD_CALENDAR_SP = "b3fd_calendar_sp";
    public static final String RSSI_SP = "rssi_sp";

    public static final String BEL_BATTERY_VALUE_SP = "ble_battery_value";
    public static final String BLE_STEP_STATUS_SP = "ble_step_status";
    public static final String BLE_SLEEP_STATUS_SP = "ble_sleep_status";
    public static final String STEP_MODE_SP = "step_mode";

    public static final String IMG_LOCAL_VERSION_NAME_SP = "img_local_version_name";
    public static final String BLE_CALENDAR_SP = "ble_step_calendar";
    // 同步睡眠数据广播
    public static final String SLEEP_ALL_DATA_ACTION = "sleep_all_data_action";
    // 同步计步数据广播
    public static final String STEP_ALL_DATA_ACTION = "step_all_data_action";
    public static final String ONLY_SET_OFF_SCREEN_TIME = "only_set_off_screen_time";
    public static final String OFF_SCREEN_TIME = "off_screen_time";
    public static final String LAST_DAY_NUMBER_SP = "last_day_number";
    public static final String LAST_DAY_CALLENDAR_SP = "last_day_calendar";
    public static final String FIRST_OPEN_APK = "first_open_apk";

    public static final String LAST_REFRESH_TIME = "last_refresh_time"; //运动详情   上一次刷新时间
    public static final String LAST_REFRESH_PHONE_MINUTE = "last_refresh_phone_minute";//上一次刷
新时的手机分钟数
    public static final String LAST_REFRESH_STEP = "last_refresh_step";//上一次刷新时的步数

    public static String READ_BATTERY_ACTION = "read_battery_action";
    public static String READ_BLE_VERSION_ACTION = "read_ble_version_action";

    public final static String SEND_DEVICE_INFO_ACTION = "send_device_info_action";

    public static int QQType = 1;
    public static int WeChatType = 2;
    public static int PhoneType = 2;
    public static int SmsType = 3;

    public static final String EXTRA_RSSI = "RSSI";
    public static final String EXTRA_RSSI_STATUS = "RSSI_STATUS";
    public static final String INTENT_BLE_VERSION_EXTRA = "get_ble_version";
```

```
    public static final String INTENT_BLE_BATTERY_EXTRA = "get_ble_battery";

    /**
     * Handler Message
     * */
    public static final int GET_RSSI_MSG = 10;
    public static final int REFRESH_UI_MSG = 11;
    public static final int NEEDLESS_REFRESH_UI_MSG = 12;

    /**
     * Alarm clock
     */
    //peroid
    public static final byte ALARM_INVALID = 0x00;
     public static final byte SUNDAY = 0x01;
    public static final byte MONDAY = 0x02;
    public static final byte TUESDAY = 0x04;
    public static final byte WEDNESDAY = 0x08;
    public static final byte THURSDAY = 0x10;
    public static final byte FRIDAY = 0x20;
    public static final byte SATURDAY = 0x40;
    public static final byte EVERYDAY = 0x7F;
    //which clock
    public static final int FIRST_CLOCK = 1;
    public static final int SECOND_CLOCK = 2;
    public static final int THIRD_CLOCK = 3;

   /
    //update status
    public static final int OLD_VERSION_STATUS = 1;//当前是旧版本状态，允许升级
    public static final int FREQUENT_ACCESS_STATUS = 2;//频繁访问服务器状态
    public static final int NEWEST_VERSION_STATUS = 3;//当前已是最新版本状态
    public static final int ACCESS_VERSER_STATUS = 4;//正在访问服务器

    //rate
    public static final int RATE_TESTING = 0;//心率测试中
    public static final int RATE_TEST_FINISH = 1;//心率测试完成
    public static final int RATE_TEST_START = 2;//开始心率测试
    public static final int RATE_TEST_STOP = 3;//停止心率测试
    //sedentary remind
    public static final int OPEN_SEDENTARY_REMIND = 1;//打开久坐提醒
    public static final int CLOSE_SEDENTARY_REMIND = 0;//关闭久坐提醒

   //
```

```
    public static final String DEVICE_FEATURE_MSEEAGE = "device_feature_message";//设备升级属性是推送功能
    public static final String DEVICE_FEATURE_UPDATE = "device_feature_update";//设备升级属性是升级功能
    public static final String DEVICE_FEATURE_WECHAT = "device_feature_wechat";//设备升级属性是微信排行
    public static final String DEVICE_FEATURE_KEY = "device_feature_key";
    public static final String CHANGE_DEVICE_FEATURE_SUCCESS_ACTION = "change_device_feature_success_action";
    public static final String BLUETOOTH_REBOOT_SUCCESS_KEY = "bluetooth_reboot_success_key";
    public static final String BLUETOOTH_REBOOT_SUCCESS_ACTION = "bluetooth_reboot_success_action";
    public static final String DEVICE_FEATURE_IS_INVALID = "device_feature_is_invalid";
    public static final int FIRST_FEATURE = 1;
    public static final int SECOND_FEATURE = 2;
    public static final int UPDATE_BLE_PROGRESS_MSG = 103;//固件升级进度百分比
    public final static String ACTION_GATT_CONNECT_FAILURE = "bluetooth.le.ACTION_GATT_FAILURE";

     public static final String IS_METRIC_UNIT_SP = "is_metric_unit_sp";//公英制单位KEY


    public static final String UV_VALUE_SP = "uv_value_sp";//UV紫外线KEY
    public static final int RATE_STATIC = 1; //静态心率
    public static final int RATE_DYNAMIC = 2;//动态心率
    public static final int TYPE_PHONE = 0;//消息内容类型 推送来电名字和号码
    public static final int TYPE_QQ = 1;//消息内容类型  昵称和内容
    public static final int TYPE_WECHAT = 2;//消息内容类型
    public static final int TYPE_SMS = 3;//消息内容类型   来短信名字、号码、内容
        // RK 固件升级
    public static String patchDownLaodAddresses;
    public static String patchServiceVersionCode = "";
    public static boolean BleAndPatchAllHasNews = false;
    public static boolean BleHasNews = false;
    public static boolean PatchHasNews = false;
    public static int updateCount = 0;
    public static String BLE_UPDATE_AGAIN = "ble_update_again";
    public static final String BLE_PATCH_DOWNLOAD_ADDR_SP = "ble_patch_download_addr";
    public static final String PATH_LOCAL_VERSION_NAME_SP = "path_local_version_name";
    public static final int PASSWORD_TYPE_SET = 1;//设置密码
    public static final int PASSWORD_TYPE_INPUT = 2;//输入密码
    public static final int PASSWORD_TYPE_INPUT_AGAIN = 3;//重新输入密码
    //public static boolean IS_SUPPORT_FULL_FONT = false;//是否是全字库（删除，用GetFunctionList.isSupportFunction 判断）
```

```
    public static final String SMS_RECEIVED_NUMBER = "sms_received_number";
    /*
     * 个人信息    性别，SharedPreferences KEY 值，boolean   true 为男，false 为女
     */
    public static final String PERSONAGE_GENDER = "personage_gender_sp";//个人信息  性
别
    public static final String LAST_BLOOD_PRESSURE_CALENDAR_SP =
"last_blood_pressure_calendar_sp";
    public static final int BLOOD_PRESSURE_TEST_STOP = 0;//停止血压测试
    public static final int BLOOD_PRESSURE_TEST_START = 1;//开始血压测试
    public static final int BLOOD_PRESSURE_TESTING = 3;//血压测试中
    public static final int BLOOD_PRESSURE_TEST_FINISH = 4;//血压测试完成

    public static final String CHARACTERISTIC_FUNCTION_LIST_SP =
"characteristic_function_list_sp";
    public static final int IS_SUPPORT_PASS_WORD_PAIR = 0x01; // 密码配对
    public static final int IS_SUPPORT_WEATHER_FORECAST = 0x02; // 天气预报
    public static final int IS_SUPPORT_MULTIPLE_LANGUAGE = 0x04; // 多国语言
    public static final int IS_SUPPORT_Do_NOT_DISTURB = 0x08; // 勿扰
    public static final int IS_SUPPORT_SWIMMING = 0x10; // 游泳
    public static final int IS_SUPPORT_HOR_VER = 0x20; // 横竖屏
    public static final int IS_SUPPORT_RAISE_HAND_BRIGHT = 0x40; // 抬手亮屏开关
    public static final int SCREEN_VERTICAL = 1;//竖屏
    public static final int SCREEN_HORIZONTAL = 2;//横屏

    public static final String WEATHER_FORECAST_SWITCH_SP =
"weather_forecast_switch_sp";
    public static final String WEATHER_FORECAST_ACTION = "weather_forecast_action";
    public static final String WEATHER_FORECAST_TODAY_WEATHER = "todayWeather";
    public static final String WEATHER_FORECAST_TODAY_CURRENT_TMP = "todayCurrentTmp";
    public static final String WEATHER_FORECAST_TODAY_TMP_MAX = "todayTmpMax";
    public static final String WEATHER_FORECAST_TODAY_TMP_MIN = "todayTmpMin";
    public static final String WEATHER_FORECAST_TODAY_PM25 = "todayPM25";
    public static final String WEATHER_FORECAST_TODAY_AQI = "todayAqi";
    public static final String WEATHER_FORECAST_TOMORROW_WEATHER = "tommorrowWeather";
    public static final String WEATHER_FORECAST_TOMORROW_TMP_MAX = "tommorrowTmpMax";
    public static final String WEATHER_FORECAST_TOMORROW_TMP_MIN = "tommorrowTmpMin";

    （新）
    public static final int SERVER_IS_BUSY = 201;//访问服务器回调状态,服务器忙
   public static final int SERVER_CALL_BACK_SUCCESSFULL = 202;//访问服务器回调状态,服
务连接正常,已获取到固件版本
    public static final int UNIT_TYPE_METRICE = 1;// 公制单位
    public static final int UNIT_TYPE_IMPERIAL = 2;// 英制单位
```

```
    public static final int HOUR_FORMAT_24 = 1;// 24小时制
    public static final int HOUR_FORMAT_12 = 2;// 12 小时制
    （新）
    public static final int IS_SUPPORT_DIAL_SWITCH = 0x800; // 判断是否支持表盘切换和左右手切换功能
    public static final int IS_SUPPORT_IBEACON = 0x1000; // 判断是否支持ibeacon这个功能
    public static final int IS_SUPPORT_SEVEN_DAYS_WEATHER = 0x2000; // 判断是否支持 七天天气
（新）
    public static final int IS_SUPPORT_SEDENTARY_REMINDER = 0x10000; // 支持久坐提醒功能Support sedentary reminder
    public static final int IS_SUPPORT_THREE_ALARM_CLOCK = 0x20000; // 支持多个闹钟（三个）
    public static final int IS_SUPPORT_TRANSACTION_REMINDER = 0x40000; // 支持事务提醒 Support for transaction reminders

    public static final int IBEACON_TYPE_UUID = 1;// Ibeacon  指令类型,设置UUID/获取UUID
    public static final int IBEACON_TYPE_MAJOR = 2;// Ibeacon  指令类型,设置major/获取major
    public static final int IBEACON_TYPE_MINOR = 4;// Ibeacon  指令类型,设置minor/获取minor
    public static final int IBEACON_TYPE_DEVICE_NAME = 8;// Ibeacon  指令类型,设置蓝牙device name/获取蓝牙device name
（新）

    public static final int IBEACON_TYPE_TX_POWER = 16;// Ibeacon  指令类型,设置蓝牙TX_POWER/获取蓝牙TX_POWER
    public static final int IBEACON_TYPE_ADVERTISING_INTERVAL= 32;// Ibeacon  指令类型,设置蓝牙 advertising interval/获取蓝牙 advertising interval
    public static final int IBEACON_SET = 0;// Ibeacon  设置(设置UUID/设置major,设置minor,设置蓝牙device name)
    public static final int IBEACON_GET = 1;// Ibeacon  获取(设置UUID/设置major,设置minor,设置蓝牙device name)
    public static final int LEFT_HAND_WEAR = 0;// 左手佩戴 Left hand wear
    public static final int RIGHT_HAND_WEAR = 1;// 右手佩戴  Right hand wear
    public static final int NOT_SET_UP = 0xff;// 不设置
    public static final int SHOW_HORIZONTAL_SCREEN = 0;// 显示横屏 Show horizontal screen
    public static final int SHOW_VERTICAL_ENGLISH_SCREEN = 1;// 显示竖屏英文界面
    public static final int SHOW_VERTICAL_CHINESE_SCREEN = 2;// 显示竖屏中文界面

}
```

3.3. OnServerCallbackListener 固件升级访问服务器回调类 （新）

方法	`public void OnServerCallback(int status);`
返回值	无
参数说明	`status`：GlobalVariable.*SERVER_IS_BUSY* //访问服务器回调状态,服务器忙 GlobalVariable. *SERVER_CALL_BACK_SUCCESSFULL* = 202;//访问服务器回调状态,服务连接正常,已获取到固件版本
说明	OnServerCallbackListener 回调结果

3.4. Updates 固件升级 （新）

方法	`public String getServerBtImgVersion() {`（新）
返回值	返回“no new version”或者返回从服务器获取到的固件版本号（升级的版本）
参数说明	无
说明	获取新版本的版本号
方法	`public String getServerPatchVersion() {`（新）
返回值	返回“no new version”或者返回从服务器获取到的 patch 版本号（升级的版本）
参数说明	无
说明	获取新版本的版本号。提示：RK 平台才有 patch 版本号。
方法	`public boolean isRKPlatform() {`（新）
返回值	`true :RK平台;false:dialog平台`
参数说明	无
说明	判断平台

天气状况代码 Code（图 1）

图 1：

代码	中文	英文
100	晴	Sunny/Clear
101	多云	Cloudy

102	少云	Few Clouds
103	晴间多云	Partly Cloudy
104	阴	Overcast
200	有风	Windy
201	平静	Calm
202	微风	Light Breeze
203	和风	Moderate/Gentle Breeze
204	清风	Fresh Breeze
205	强风/劲风	Strong Breeze
206	疾风	High Wind, Near Gale
207	大风	Gale
208	烈风	Strong Gale
209	风暴	Storm
210	狂爆风	Violent Storm
211	飓风	Hurricane
212	龙卷风	Tornado

213	热带风暴	Tropical Storm
300	阵雨	Shower Rain
301	强阵雨	Heavy Shower Rain
302	雷阵雨	Thundershower
303	强雷阵雨	Heavy Thunderstorm
304	雷阵雨伴有冰雹	Hail
305	小雨	Light Rain
306	中雨	Moderate Rain
307	大雨	Heavy Rain
308	极端降雨	Extreme Rain
309	毛毛雨/细雨	Drizzle Rain
310	暴雨	Storm
311	大暴雨	Heavy Storm
312	特大暴雨	Severe Storm
313	冻雨	Freezing Rain
400	小雪	Light Snow

401	中雪	Moderate Snow
402	大雪	Heavy Snow
403	暴雪	Snowstorm
404	雨夹雪	Sleet
405	雨雪天气	Rain And Snow
406	阵雨夹雪	Shower Snow
407	阵雪	Snow Flurry
500	薄雾	Mist
501	雾	Foggy
502	霾	Haze
503	扬沙	Sand
504	浮尘	Dust
507	沙尘暴	Duststorm
508	强沙尘暴	Sandstorm
900	热	Hot
901	冷	Cold

999	未知	Unknown

4.免责声明

深圳市优创亿科技有限公司所提供的所有服务内容旨在协助客户加速产品的研发进度，在服务过程中所提供的任何程序、文档、测试结果、方案、支持等资料和信息，都仅供参考，客户有权不使用，本公司不提供任何的完整性、可靠性等保证，若在客户使用过程中因任何原因造成特别的、偶然的或间接的损失，本公司不承担任何责任。