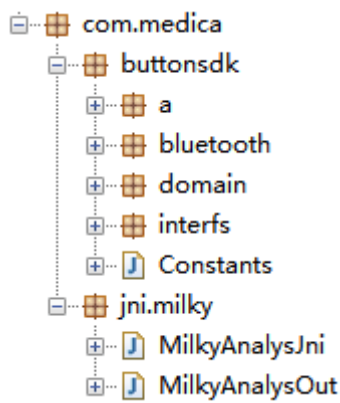


Button-SDK 接口详细说明

1.结构



com.medica.jni.milky 算法包，其中 MilkyAnalysJni 是算法核心类，MilkyAnalysOut 是算法分析结果输出类。后面章节会做详细使用说明。

com.medica.buttonsdk 主程序包，a 是混淆后的子包，因不重要，故不做说明。Bluetooth 是蓝牙接口通讯核心包。Domain 是存放领域对象的子包。Interfs 是接口包。Constants 是该 SDK 中定义的一些常亮。

2.相关类说明

```
package com.medica.jni.milky;
/**
 * 睡眠分析结果对象
 */
public class MilkyAnalysOut implements Serializable
```

字段说明

- analysVer

public java.lang.String analysVer
算法版本

- exceptioncode

public short exceptioncode
异常返回代码 0:分析成功 非 0: 分析失败

- flaginvalid

public short flaginvalid

分析的结果数据有效性标识 0: 有效; 非 0: 无效, 数据有异常, 不建议作为参考

- **offsetMS**

public short offsetMS

起始位置偏移: offset monitor start

- **offsetME**

public short offsetME

终止位置偏移: offset monitor end

- **monitormins**

public short monitormins

睡眠监测时长, 单位分钟

- **latencymins**

public short latencymins

入睡时长, 单位分钟

- **wakemins**

public short wakemins

清醒总时长, 单位分钟

- **wakeminsdurs**

public short wakeminsdurs

睡中清醒时长, 单位分钟

- **waketimesdurs**

public short waketimesdurs

睡中清醒次数

- **getupwakemins**

public short getupwakemins

起床前清醒时长, 单位分钟

- **lightmins**

public short lightmins

浅睡总时长, 单位分钟

- **transmins**

public short transmins

中睡总时长, 单位分钟

- **deepmins**

public short deepmins
深睡总时长，单位分钟

- **sleepmins**

public short sleepmins
有效睡眠时长（浅/中/深睡眠时间总和），单位分钟

- **sleepefficient**

public short sleepefficient
睡眠效率（有效睡眠百分比）

- **wakepercent**

public short wakepercent
清醒百分比

- **lightpercent**

public short lightpercent
浅睡百分比

- **transpercent**

public short transpercent
中睡百分比

- **deeppercent**

public short deeppercent
深睡百分比

- **motionfreq**

public short motionfreq
体动频率分布：-2:体动极少；-1:体动较少；0:体动正常；1 体动过多；2 体动极多

- **sleepscore**

public short sleepscore
睡眠得分（0-100 分）

- **markitem01**

public short markitem01
扣分 01:有效睡眠时长（过长:大于 0； 过短:小于 0）

- **markitem02**

public short markitem02
扣分 02:入睡时长

- **markitem03**

public short markitem03

扣分 03:清醒次数

- **markitem04**

public short markitem04

扣分 04:睡眠效率

- **markitem05**

public short markitem05

扣分 05:良性睡眠（中睡+深睡）占总监测时间百分比 附加:深睡占良性睡眠比率（Ratio = Deeppercent / (Transpercent + Deeppercent)）

- **markitem06**

public short markitem06

扣分 06:体动频率分布（过多:05/10）

- **markitem07**

public short markitem07

扣分 07:晚睡扣分，当天 23 点过后，第二天 12 点前，5 分/小时线性扣分，上限 15 分。

- **sleep_curve**

public float[] sleep_curve

睡眠状态曲线（输出范围:0-3）（0-清醒/1-浅睡/2-中睡/3-深睡）

- **sleep_event**

public short[] sleep_event

睡眠事件标志（0x01-入睡点，0x02-起床标志）

- **sleep_stage**

public short[] sleep_stage

睡眠状态分期（0-清醒/1-浅睡/2-中睡/3-深睡）

- **motion_intensity**

public short[] motion_intensity

体动强度分布（归一化体动强度，输出范围:0-250）

- **motion_density**

public short[] motion_density

体动密度分布(每分钟体动点个数，输出范围:0-60)

```
package com.medica.jni.milky;
```

```
/**
```

```
 * 睡眠结果分析算法
```

```
*/  
public class MilkyAnalysJni
```

方法详细说明

- **analysis**

```
public static MilkyAnalysOut analysis(int startTime, Detail detail)
```

参数:

startTime - 开始监测时间，单位秒

detail - 监测详细数据

返回:

睡眠分析结果，仅供参考

```
package com.medica.buttonsdk.bluetooth;  
/**  
 * 蓝牙通讯，SDK接口核心类  
 */  
public class ButtonHelper
```

常量字段说明

- **STATE_DISCONNECTED**

```
public static final int STATE_DISCONNECTED
```

ble 设备连接状态：连接断开

- **STATE_CONNECTING**

```
public static final int STATE_CONNECTING
```

ble 设备连接状态：正在连接

- **STATE_CONNECTED**

```
public static final int STATE_CONNECTED
```

ble 设备连接状态：已连接

- **ACTION_LOW_POWER**

```
public static final String ACTION_LOW_POWER
```

设备低电量通知 Action

方法详细说明

- **getInstance**

`public static ButtonHelper getInstance(android.content.Context mContext)`

参数:

mContext - 上下文对象

返回: 该类的实例对象

- **isSupportBle**

`public boolean isSupportBle()`

判断手机是否支持 Ble 功能

返回: boolean 值

- **isBluetoothOpen**

`public boolean isBluetoothOpen()`

判断手机蓝牙开关是否打开

返回: boolean 值

- **openBluetooth**

`public void openBluetooth()`

无提示打开手机蓝牙开关

- **getConnectState**

`public int getConnectState()`

返回当前设备连接状态

返回: 设备状态 STATE_CONNECTING, STATE_CONNECTED, STATE_DISCONNECTED

- **addBleStateChangedListener**

`public void addBleStateChangedListener(BleStateChangedListener l)`

添加设备连接状态变化监听

参数:

l - 连接状态改变监听

- **addSleepStatusListener**

`public void addSleepStatusListener(SleepStatusListener l)`

添加睡眠状态监听接口

参数:

l - 睡眠状态监听接口

- **removeBleStateChangedListener**

`public void removeBleStateChangedListener(BleStateChangedListener l)`

移除设备连接状态变化监听

参数:

I - 连接状态改变监听

- **removeSleepStatusListener**

`public void removeSleepStatusListener(SleepStatusListener I)`

移除睡眠状态监听

参数:

I - 睡眠状态监听接口

- **scanBleDevice**

`public void scanBleDevice(BleScanListener bleScanListener)`

扫描睡眠监测设备, 扫描前请先确认打开蓝牙开关

参数:

bleScanListener - 扫描监听

- **stopScan**

`public void stopScan()`

停止扫描设备

- **connDevice**

`public void connDevice(ResultCallback callback)`

连接设备, 该方法用于一个设备的情况下, 方法内部会扫描设备并连接该设备, 多设备时, 需要用户自己扫描, 并选择需要的设备进行连接 请参考 `connDevice(BleDevice, ResultCallback)`

参数:

callback - 回调, 返回 Boolean, true: 连接成功, false: 连接失败

- **connDevice**

`public void connDevice(BleDevice bleDevice, ResultCallback callback)`

连接设备

参数:

bleDevice - 需要连接的设备

callback - 回调, 返回 Boolean, true: 连接成功, false: 连接失败

- **loginDevice**

`public void loginDevice(int userId, ResultCallback callback)`

登录设备

参数:

userId - 用户 id

callback - 回调, 返回 Boolean, true: 登录成功, false: 登录失败

- **disconnect**

`public void disconnect()`

断开设备连接

- **getDeviceStatus**

`public void getDeviceStatus(ResultCallback callback)`

查询设备当前采集状态

参数:

`callback` - 回调, 返回 `int`, `-1|0|1`; `-1`: 查询失败, `0`: 非采集状态, `1`: 正在采集

- **stopCollect**

`public void stopCollect(ResultCallback callback)`

停止采集, 设备会将前面的数据生成一段报告, 然后会重新开始采集, 主要用于同步数据; 设备真正停止采集是在设定的时间点停止采集。

参数:

`callback` - 回调, 返回 `Boolean`, `true`: 操作成功, `false`: 操作失败

- **queryHistorySummary**

`public void queryHistorySummary(int startTime, int endTime, ResultCallback callback)`

历史数据概要查询, 异步方法

参数:

`startTime` - 开始时间, 单位秒 `s`

`endTime` - 结束时间, 单位秒 `s`

`callback` - 回调, 返回: `ArrayList<Summary>`, 如果查询失败, 则返回 `null`

- **queryHistorySummary**

`public java.util.ArrayList<Summary> queryHistorySummary(int startTime, int endTime)`

历史数据概要查询, 同步方法

参数:

`startTime` - 开始时间, 单位秒 `s`

`endTime` - 结束时间, 单位秒 `s`

返回: `ArrayList<Summary>`, 如果查询失败, 则返回 `null`

- **queryHistoryDetail**

`public void queryHistoryDetail(Summary summ, ResultCallback callback)`

历史数据详细信息查询, 异步方法

参数:

`summ` - 概要信息

`callback` - 回调, 返回: `Detail`, 如果查询失败, 则返回 `null`

- **queryHistoryDetail**

`public Detail queryHistoryDetail(Summary summ)`

历史数据详细信息查询, 同步方法

参数:

`summ` - 概要信息

返回:

`Detail`, 如果查询失败, 则返回 `null`

- **getDeviceVersion**

```
public void getDeviceVersion(ResultCallback callback)
```

获取硬件的版本号

参数:

callback - 回调, 返回: String, 如: "3.30", 如果查询失败, 则返回 null

- **getDevicePower**

```
public void getDevicePower(ResultCallback callback)
```

获取纽扣电量百分比

参数:

callback - 回调, 返回 int, 如: 80, 则表示剩余电量 80%, 获取失败则返回-1

- **querySleepStatus**

```
public void querySleepStatus(ResultCallback callback)
```

查询睡眠状态

参数:

callback - 回调, 返回 SleepStatus, 查询失败时, SleepStatus 中返回无效值

- **setAutoStart**

```
public void setAutoStart(byte sHour,  
                          byte sMinute,  
                          byte eHour,  
                          byte eMinute,  
                          boolean[] repeat,  
                          ResultCallback callback)
```

设置自动开始监测

参数:

sHour - 开始小时 24 小时制

sMinute - 开始分钟

eHour - 结束小时 24 小时制

eMinute - 结束分钟

repeat - 重复, repeat[0]对应周一, repeat[1]对应周二, 以此类推

callback - 回调, 返回 Boolean, true:设置成功, false: 设置失败

- **setAlarmTime**

```
public void setAlarmTime(byte alartOFF,  
                          byte autoGet,  
                          byte autoMove,  
                          byte hourOfDay,  
                          byte minute,  
                          byte repeat,  
                          ResultCallback callback)
```

设置纽扣智能唤醒

参数:

alarmOFF - 闹钟开关, 1: 开, 0: 关

autoGet - 是否智能唤醒, 1: 是, 0: 否

autoMove - 唤醒时间范围, 单位分钟

hourOfDay - 小时, 24 小时制

minute - 分钟

repeat - 重复 (一个字节, 对应 8 位, 取后 7 位, 末位对应周一, 倒数第二位对应周二, 以此类推。1 表示重复, 0 不重复。如: 3, 对应二进制: 00000011, 表示: 周一周二重复)

callback - 回调, 返回 Boolean, true: 设置成功, false: 设置失败

- **getDeviceId**

```
public void getDeviceId(ResultCallback callback)
```

获取设备加密 id, 异步方法

参数:

callback - 回调, 返回 String, 获取失败返回: null

- **getDeviceId**

```
public java.lang.String getDeviceId()
```

获取设备加密 id 号 同步方法

返回: 加密 id, 获取失败, 返回: null

- **upgradeFirmwareByThread**

```
public void upgradeFirmwareByThread(float versionCode,  
                                     int crcBin,  
                                     int crcDes,  
                                     java.io.File file,  
                                     UpgradeCallback callback)
```

固件升级, 异步方法

参数:

versionCode - 设备版本号

crcBin - 校验码

crcDes - 校验码

file - 固件文件

callback - 回调, onResult 方法中返回 Boolean, true: 升级成功, false: 升级失败;
onUpgrade 方法中回调进度

- **upgradeFirmware**

```
public boolean upgradeFirmware(float versionCode,  
                               int crcBin,  
                               int crcDes,  
                               java.io.File file,  
                               UpgradeCallback callback)
```

固件升级, 同步方法, 回调中仅返回升级进度

参数:

versionCode - 设备版本号

crcBin - 校验码

crcDes - 校验码

file - 固件文件

callback - 升级进度回调

返回: 升级结果, true: 升级成功, false: 升级失败

- **syncTimeByThread**

public void syncTimeByThread(ResultCallback callback)

对时, 异步方法。 使用说明参见: syncTime()

参数:

callback - 回调, 返回 Boolean, true: 对时成功, false: 对时失败

- **syncTime**

public boolean syncTime()

对时, 同步方法。 用于同步手机和固件上的时间, 同步成功之后, 设置自动开始监测才会准确有效, 建议每次连上设备就做一次对时操作

返回: Boolean, true: 对时成功, false: 对时失败

```
package com.medica.buttonsdk.domain;
```

```
/**
```

```
 * 纽扣对象
```

```
 */
```

```
public class BleDevice implements Serializable
```

字段详细说明

- **deviceId**

public String deviceId

设备密文 id

- **deviceName**

public String deviceName

设备名称, 如: B1-xxxxxxxxxx

- **modelName**

public String modelName

设备型号名称, 如: Sleepace B

- **address**

public String address

设备地址

```
package com.medica.buttonsdk.domain;
/**
 * 睡眠数据详细信息
 */
public class Detail implements Serializable
```

字段详细说明

- feature1**

public int[] feature1
原始数据 Feature1
- feature2**

public int[] feature2
原始数据 Feature2
- statusFlag**

public int[] statusFlag
原始数据 statusFlag
- sleepStage**

public byte[] sleepStage
睡眠结果数组，如：深睡眠，浅睡眠，清醒等
- Sleep_Curve**

public float[] Sleep_Curve
波形图曲线
- Sleep_Event**

public short[] Sleep_Event
波形图的特殊点

```
package com.medica.buttonsdk.domain;
/**
 * 实时数据对象
 */
public class SleepStatus implements Serializable
```

字段详细说明

- **sleepFlag**

public int sleepFlag

入睡标识，1 标识入睡，0 未入睡，-1 无效值 说明：此参数用于判断用户是否入睡，在睡眠辅助时使用，如果监测到用户入睡，则停止睡眠辅助

- **wakeFlag**

public int wakeFlag

清醒标识，1 标识清醒，0 未清醒，-1 无效值 说明：此参数用于判断用户是否清醒，在智能唤醒时使用，如果监测到用户进入清醒状态，则响起闹钟，唤醒用户

```
package com.medica.buttonsdk.domain;
/**
 * 睡眠数据概要信息
 */
public class Summary implements Serializable
```

字段详细说明

- **userId**

public int userId

用户 id

- **deviceId**

public String deviceId

设备加密 id

- **source**

public int source

数据来源，即设备类型，多种设备的情况下，管理分析数据时用到
Constants.DEVICE_TYPE_BUTTON

取值参考：

- **startTime**

public int startTime

开始睡觉时间，单位秒（s）

- **timezone**

public float timezone

时区

- **timeStep**

public int timeStep

时间节点，如：60 秒为一个分析的时间节点

- **recordCount**

public int recordCount

记录条数

- **stopMode**

public int stopMode

设备停止模式，自动模式，手动模式

```
package com.medica.buttonsdk.interfs;
```

```
/**
```

```
 * 扫描监听接口
```

```
 */
```

```
public interface BleScanListener
```

方法详细说明

- **onBleScanStart**

void onBleScanStart()

扫描开始

- **onBleScan**

void onBleScan(BleDevice device)

扫描到设备时回调该方法 注意：异步方法，非线程安全

参数：

device - 扫描到的设备

- **onBleScanFinish**

void onBleScanFinish()

扫描结束

```
package com.medica.buttonsdk.interfs;
```

```
/**
```

```
 * 设备连接状态改变监听接口
```

```
 */
```

```
public interface BleStateChangedListener
```

方法详细说明

- **onStateChanged**

`void onStateChanged(int state)`

参数:

state - 取值参考

ButtonHelper.STATE_DISCONNECTED,

ButtonHelper.STATE_CONNECTING,

ButtonHelper.STATE_CONNECTED

```
package com.medica.buttonsdk.interfs;
```

```
/**
```

```
 * 睡眠状态改变监听接口
```

```
 */
```

```
public interface SleepStatusListener
```

方法详细资料

- **onSleepStatusChanged**

`void onSleepStatusChanged(SleepStatus status)`

参数:

status - 取值参考 SleepStatus

```
package com.medica.buttonsdk.interfs;
```

```
/**
```

```
 * 异步方法回调接口
```

```
 */
```

```
public interface ResultCallback
```

方法详细说明

- **onResult**

`void onResult(Method method, Object result)`

回调方法

参数:

method - 方法

result - 返回结果对象，可能是 Boolean，String，自定义对象，具体返回类型，请参考调用的方法的注释说明

```
package com.medica.buttonsdk.interfs;
/**
 * 升级固件回调接口
 */
public interface UpgradeCallback extends ResultCallback
```

方法详细说明

- **onUpgrade**

void onUpgrade(int progress)
升级固件包进度接口，最大进度 100
参数:
progress - 升级进度

```
package com.medica.buttonsdk.interfs;
/**
 * sdk中所有方法的枚举类型
 */
public enum Method
```

枚举常量详细说明

- **CONNECT_DEVICE**

public static final Method CONNECT_DEVICE
连接设备

- **LOGIN**

public static final Method LOGIN
登录

- **SYNC_TIME**

public static final Method SYNC_TIME
对时

- **GET_DEVICE_VERSION**

public static final Method GET_DEVICE_VERSION
获取设备版本

- **GET_DEVICE_STATUS**

public static final Method GET_DEVICE_STATUS
获取设备状态

- **STOP_COLLECT**

public static final Method STOP_COLLECT
停止采集数据

- **QUERY_SLEEP_STATUS**

public static final Method QUERY_SLEEP_STATUS
查询睡眠状态

- **QUERY_HISTORY_SUMMARY**

public static final Method QUERY_HISTORY_SUMMARY
查询历史数据概要信息

- **QUERY_HISTORY_DETAIL**

public static final Method QUERY_HISTORY_DETAIL
查询历史数据详细信息

- **GET_DEVICE_POWER**

public static final Method GET_DEVICE_POWER
获取 RestOn 电量百分比

- **SET_AUTO_START**

public static final Method SET_AUTO_START
设置自动开始监测

- **SET_ALARM_TIME**

public static final Method SET_ALARM_TIME
设置闹钟时间

- **GET_DEVICE_ID**

public static final Method GET_DEVICE_ID
获取设备 id

- **UPGRADE_FIRMWARE**

public static final Method UPGRADE_FIRMWARE
升级固件

```
package com.medica.buttonsdk;  
/**  
 * 纽扣 SDK相关常量  
 */
```

`public class` Constants

字段详细说明

• SDK_VERSION

public static final String SDK_VERSION
SDK 版本

• CONFIG_FILENAME

public static final String CONFIG_FILENAME
配置文件名称

• DEVICE_TYPE_BUTTON

public static final short DEVICE_TYPE_BUTTON
纽扣设备

常量字段值

com.medica.buttonsdk.Constants		
限定符和类型	常量字段	值
public static final String	CONFIG_FILENAME	"btnSdkConfig"
public static final short	DEVICE_TYPE_BUTTON	10
public static final String	SDK_VERSION	"1.0.0"

com.medica.buttonsdk.bluetooth.ButtonHelper		
限定符和类型	常量字段	值
public static final String	ACTION_LOW_POWER	"com.medica.buttonsdk.Device_LowPower"
public static final int	STATE_CONNECTED	2
public static final int	STATE_CONNECTING	1
public static final int	STATE_DISCONNECTED	0