|  |
| --- |
|  |
| **Binatone Android SDK Description** |
| **V1.0** |
|  |
| **Author: jaker** |
| **2018/08/31** |

# Change log

|  |  |  |
| --- | --- | --- |
| **Date** | **Log** | **Author** |
| 2018-08-31 | Create | jaker |
| 2019/05/08 | Adding MTU setup interface | smile |
| 2019/08/09 | Adding alarm monitoring interface | smile |
| 2021/11/16 | 1. Support multi device connection 2. Update historyDownload interface and update algorithm | smile |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |

# Catalog

[Change log 1](#_Toc17190937)

[Catalog 2](#_Toc17190938)

[Android SDK Intro 6](#_Toc17190939)

[1. Function and Purpose 6](#_Toc17190940)

[Integration 6](#_Toc17190941)

[1 .SDK framework 6](#_Toc17190942)

[2 .Integration 6](#_Toc17190943)

[Eclipse Config 6](#_Toc17190944)

[API 8](#_Toc17190945)

[1.API initialization 8](#_Toc17190946)

[Description 8](#_Toc17190947)

[Parameters 8](#_Toc17190948)

[2. SET MTU 8](#_Toc17190949)

[Description 8](#_Toc17190950)

[Parameters 8](#_Toc17190951)

[3. Connnect Device 9](#_Toc17190952)

[Description 9](#_Toc17190953)

[Parameters 9](#_Toc17190954)

[3. Get SDK Version 9](#_Toc17190955)

[Description 9](#_Toc17190956)

[4.Get Battery 10](#_Toc17190957)

[Description 10](#_Toc17190958)

[Parameters 10](#_Toc17190959)

[5.Get Device Information 10](#_Toc17190960)

[Description 10](#_Toc17190961)

[Parameters 10](#_Toc17190962)

[6.Get Device Version 10](#_Toc17190963)

[Description 11](#_Toc17190964)

[Parameters 11](#_Toc17190965)

[7.Get Birthday 11](#_Toc17190966)

[Description 11](#_Toc17190967)

[Parameters 11](#_Toc17190968)

[8.Set Birthday 11](#_Toc17190969)

[Description 11](#_Toc17190970)

[Parameters 11](#_Toc17190971)

[9.Get Apnea Alarm Parameters 12](#_Toc17190972)

[Description 12](#_Toc17190973)

[Parameters 12](#_Toc17190974)

[10.Set Apnea Alarm Parameters 12](#_Toc17190975)

[Description 12](#_Toc17190976)

[Parameters 12](#_Toc17190977)

[11.Get Out Of Bed Alarm Parameters 13](#_Toc17190978)

[Description 13](#_Toc17190979)

[Parameters 13](#_Toc17190980)

[12.Set Out Of Bed Alarm Parameters 13](#_Toc17190981)

[Description 13](#_Toc17190982)

[Parameters 13](#_Toc17190983)

[13.Get Sleep Data (Real-time) 13](#_Toc17190984)

[Description 14](#_Toc17190985)

[Parameters 14](#_Toc17190986)

[14.Stop Getting Data(Real-time) 14](#_Toc17190987)

[Description 14](#_Toc17190988)

[Parameters 14](#_Toc17190989)

[15.Get 24 hours Sleep Report 14](#_Toc17190990)

[Description 14](#_Toc17190991)

[Parameters 14](#_Toc17190992)

[16.Get Sleep Reports 15](#_Toc17190993)

[Description 15](#_Toc17190994)

[Parameters 15](#_Toc17190995)

[17.Firmware Update 1 15](#_Toc17190996)

[Description 15](#_Toc17190997)

[Parameters 15](#_Toc17190998)

[18.Firmware Update 2 16](#_Toc17190999)

[Description 16](#_Toc17191000)

[Parameters 16](#_Toc17191001)

[19.Restore factory settings 16](#_Toc17191002)

[Description 16](#_Toc17191003)

[Parameters 16](#_Toc17191004)

[20. Analysis of Sleep Data 16](#_Toc17191005)

[Description 17](#_Toc17191006)

[Sleep data analysis 17](#_Toc17191007)

[Parameters 17](#_Toc17191008)

[Return：Analysis 17](#_Toc17191009)

[21. Stop downloading historical data 17](#_Toc17191010)

[Description 17](#_Toc17191011)

[Stop downloading historical data 17](#_Toc17191012)

[22.Termination of access to the latest 24-hour historical data 17](#_Toc17191013)

[Description 17](#_Toc17191014)

[23. Add alarm monitoring 17](#_Toc17191015)

[Description 17](#_Toc17191016)

[Parameters 18](#_Toc17191017)

[24. Remove alarm monitoring 18](#_Toc17191018)

[Description 18](#_Toc17191019)

[Parameters 18](#_Toc17191020)

[Object Description 18](#_Toc17191021)

[StatusCode 18](#_Toc17191022)

[Description 18](#_Toc17191023)

[Fields 18](#_Toc17191024)

[IResultCallback<T> 19](#_Toc17191025)

[Description 19](#_Toc17191026)

[Function 19](#_Toc17191027)

[CallbackData<T> 19](#_Toc17191028)

[Description 19](#_Toc17191029)

[Fields 19](#_Toc17191030)

[DeviceInfo 20](#_Toc17191031)

[Description 20](#_Toc17191032)

[Fields 20](#_Toc17191033)

[BatteryBean 20](#_Toc17191034)

[Description 20](#_Toc17191035)

[Fields 20](#_Toc17191036)

[RealTimeData 20](#_Toc17191037)

[Description 20](#_Toc17191038)

[Fields 20](#_Toc17191039)

[HistoryData 21](#_Toc17191040)

[Description 21](#_Toc17191041)

[Fields 21](#_Toc17191042)

[Summary 21](#_Toc17191043)

[Description 21](#_Toc17191044)

[Fields 21](#_Toc17191045)

[Detail 22](#_Toc17191046)

[Description 22](#_Toc17191047)

[Fields 22](#_Toc17191048)

[Analysis 22](#_Toc17191049)

[Description 22](#_Toc17191050)

[Fields 22](#_Toc17191051)

[AlarmConfig 23](#_Toc17191052)

[Description 23](#_Toc17191053)

[Fields 23](#_Toc17191054)

[Birthday 23](#_Toc17191055)

[Description 23](#_Toc17191056)

[Fields 23](#_Toc17191057)

[SleepStatusType 23](#_Toc17191058)

[Description 23](#_Toc17191059)

[Fields 24](#_Toc17191060)

[DeviceStatusListener 24](#_Toc17191061)

[Interface Description 24](#_Toc17191062)

[Constant Field Description 24](#_Toc17191063)

[Function 24](#_Toc17191064)

# Android SDK Intro

## Function and Purpose

Sleep\_sensor SDK,launched by Sleepace, is a software development kit for fast Internet APP development on android platform.

The SDK encapsulates the communication process between APP and hardware, and provides functions such as device configuration, device control and data query. Using the SDK, users do not need to care about complex communication protocols and the underlying implementation, only need to focus on the APP interaction and business level.

# Integration

## 1 .SDK framework

|  |  |
| --- | --- |
| **Framework** | **Description** |
| sdkcore.jar | SDK base core |
| binatonesdk.jar | Binatone SDK |
| sdkalgorithm.jar | Algorithm call library |
| libalgorithm.so | Algorithm library |

## 2 .Integration

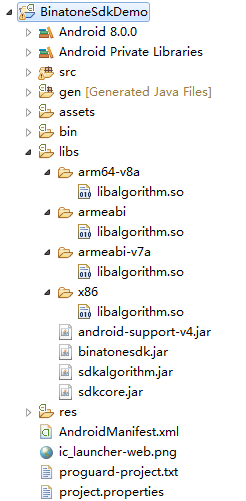
There are many Android development tools, and here we introduce the engineering configuration method of Sleepace SDK with Eclipse.

## Eclipse Config

**Step 1**：

In the project to create a "libs" folder, copy sdkcore.jar, binatonesdk.jar, sdkalgorithm.jar to "libs" folder, copy libalgorithm.so to "libs \ armeabi" folder.

Like this:



**Setp 2:**

Config the “AndroidManifest.xml”

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

<uses-permission android:name=*"android.permission.BLUETOOTH\_ADMIN"*/>

<uses-feature android:name=*"android.hardware.bluetooth\_le"* android:required=*"true"*/>

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"*/>

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"*/>

<uses-permission android:name=*"android.permission.ACCESS\_COARSE\_LOCATION"* />

<uses-permission android:name=*"android.permission.ACCESS\_FINE\_LOCATION"* />

# API

## 1.API initialization

BinatoneHelper.getInstance(Context mContext);

### Description

BinatoneHelper Initialization

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| mContext | Context | Conetxt |

## SET MTU

**public** **void** setMtu(String address,**int** mtu, [IResultCallback](#_IDataCallback<T>)<Integer> cb)

### Description

Used to set the size of Bluetooth device mtu, call before connecting device, refer to Demo

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| mtu | int | MTU size, range: [20, 509] |
| cb | [IResultCallback](#_IDataCallback<T>)<Integer> | Callback interface, when set successfully, callback. getResult returns the MUT size actually supported by the device, which is the real writable data size, excluding one byte of ATT opcode and two bytes of ATT handle. |

## Connnect Device

**public** **void** login(String deviceName, String address, **int** userId, **int** timestamp**, int** timezone**, int** dstOffset**, int** timeout, [IResultCallback](#_IDataCallback<T>)<DeviceInfo> cb)

### Description

Connect Reston and setting userId

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceName | String | deviceName |
| address | String | Bluetooth address |
| userId | int | userId does not belong to Sleepace.  userId belong to partner  **Why need it：**  Reston separates the data according to userId.  It mean user A connects to device, generates and gets the data which only belong to user A. User A can’t get the data of user B |
| timestamp | int | Unix timestamp, Unit(second) |
| timezone | int | time zone, Unit(second) |
| dstOffset | int | Summer time offset, unit seconds |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<DeviceInfo> | Callback function, if success,return [DeviceInfo](#_LoginBean) Obj |

## 3. Get SDK Version

**public** String getSDKVersion()

### Description

get sdk version

## 4.Get Battery

**public** **void** getBattery(String address, **int** timeout, IResultCallback<BatteryBean> cb)

### Description

Get battery

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[BatteryBean](#_BatteryBean)> | Callback function, if success,return [BatteryBean](#_BatteryBean) Object |

## 5.Get Device Information

**public** **void** getDeviceInfo(String address, **int** timeout, IResultCallback<DeviceInfo> cb)

### Description

Get device information

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[DeviceInfo](#_DeviceInfo)> | Callback function, if success,return [DeviceInfo](#_DeviceInfo) Object |

## 6.Get Device Version

**public** **void** getDeviceVersion(String address, **int** timeout, IResultCallback<String> cb)

### Description

Get current version of device

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<String> | Callback function, if success,return the version of device |

## 7.Get Birthday

**public** **void** getBirthday(String address, **int** timeout, IResultCallback<[Birthday](#_Birthday)> cb)

### Description

Get birthday

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[Birthday](#_Birthday)> | Callback function, if success,return the birthday |

## 8.Set Birthday

**public** **void** setBirthday(String address, **int** year, **int** month, **int** day, **int** timeout, IResultCallback<Void> cb)

### Description

set birthday

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| year | int | year |
| month | int | month of year[1-12] |
| day | int | day of month |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<Void> | Callback function |

## 9.Get Apnea Alarm Parameters

**public** **void** getBreathPauseAlarm (String address, **int** timeout, IResultCallback<[AlarmConfig](#_AlarmConfig_1)> cb)

### Description

Get Apnea alarm parameters

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[AlarmConfig](#_AlarmConfig_1)> | Callback function, if success,return the AlarmConfig |

## 10.Set Apnea Alarm Parameters

**public** **void** setBreathPauseAlarm (String address, **boolean** enable, **int** hour, **int** minute, **int** duration, **int** timeout, IResultCallback<[Void](#_AlarmConfig)> cb)

### Description

set apnea alarm parameters

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| enable | boolean | alarm switch |
| hour | int | hour of day[0-23] |
| minute | int | minute |
| duration | int | Monitoring time, Unit((minute) |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[Void](#_AlarmConfig)> | Callback function |

## 11.Get Out Of Bed Alarm Parameters

**public** **void** getOutOfBedAlarm (String address, **int** timeout, IResultCallback<[AlarmConfig](#_AlarmConfig_1)> cb)

### Description

Get out of bed alarm parameters

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[AlarmConfig](#_AlarmConfig_1)> | Callback function, if success,return the AlarmConfig |

## 12.Set Out Of Bed Alarm Parameters

**public** **void** setOutOfBedAlarm (String address, **boolean** enable, **int** hour, **int** minute, **int** duration, **int** timeout, IResultCallback<[Void](#_AlarmConfig)> cb)

### Description

set out of bed alarm parameters

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| enable | boolean | alarm switch |
| hour | int | hour of day[0-23] |
| minute | int | minute |
| duration | int | Monitoring time, Unit((minute) |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[Void](#_AlarmConfig)> | Callback function |

## 13.Get Sleep Data (Real-time)

**public** **void** startRealTimeData(String address, **int** timeout, IResultCallback<[RealTimeData](#_RealTimeData_1)> cb)

### Description

Get Real-time Data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[RealTimeData](#_RealTimeData)> | Callback function |

## 14.Stop Getting Data(Real-time)

**public** **void** stopRealTimeData(String address, **int** timeout, IResultCallback<Void> cb)

### Description

Stop getting real-time data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<Void> | Callback function |

## 

## 15.Get 24 hours Sleep Report

**public** **void** getLast24HourData(String address, **int** endTime, **int** sex, IResultCallback<HistoryData> cb)

### Description

Get 24 hours of sleep data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| endTime | int | The time of end point,Unit(Second) |
| sex | int | Gender,1:male 2:female |
| cb | [IResultCallback<HistoryData>](#_HistoryData) | Callback function |

## 16.Get Sleep Reports

**public** **void** historyDownload(String address, **int** startTime, **int** endTime, **int** sex, IResultCallback<List<HistoryData>> cb)

### Description

Get historical sleep data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| starTime | int | Start time(timestamp), Unit(second) |
| endTime | int | End time(timestamp), Unit(second) |
| sex | int | Gender,1:male 0:female |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<List<[HistoryData](#_HistoryData)>> | Callback function |

## 17.Firmware Update 1

**public** **void** upgradeDevice(String address, **final** **long** crcDes, **final** **long** crcBin, **final** File file, **final** IResultCallback<Integer> cb)

### Description

Firmware Update

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| crcDes | long | Get it from Sleepace |
| crcBin | long | Get it from Sleepace |
| file | File | Firmware object |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<Integer> | Callback function, Return upgrade progress |

## 18.Firmware Update 2

**public** **void** upgradeDevice(String address, **final** **long** crcDes, **final** **long** crcBin, **final** InputStream is, **final** IResultCallback<Integer> cb)

### Description

Firmware Update

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| crcDes | long | Get it from Sleepace |
| crcBin | long | Get it from Sleepace |
| is | InputStream | file inputstream |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<Integer> | Callback function, Return upgrade progress |

## 19.Restore factory settings

**public** **void** restoreFactorySettings(String address, **int** timeout, IDataCallback<Void> cb)

### Description

Restore factory settings

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<Void> | Callback function |

## 20. Analysis of Sleep Data

com.sleepace.sdk.binatone.util.AnalysisUtil

**public** **static** Analysis analysData(Summary summ, Detail detail,

**int** sex)

### Description

### Sleep data analysis

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| summ | Summary | Summary structure of sleep data |
| detail | Detail | Detail structure of sleep data |
| sex | int | 1: Male, 0: Female |

### Return：Analysis

## 21. Stop downloading historical data

**public** **void** abortHistoryDownload(String address)

### Description

### Stop downloading historical data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |

## 22.Termination of access to the latest 24-hour historical data

**public** **void** abortHistoryDownload(String address)

### Description

Termination of access to the latest 24-hour historical data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| address | String | Bluetooth address |

## 23. Add alarm monitoring

**public** **boolean** addDeviceStatusListener ([DeviceStatusListener](#_DeviceStatusListener) alarmListener)

### Description

Alarm for monitoring equipment

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| alarmListener | [DeviceStatusListener](#_DeviceStatusListener) | Alarm monitoring interface |

## 24. Remove alarm monitoring

**public** **boolean** removeDeviceStatusListener ([DeviceStatusListener](#_DeviceStatusListener) alarmListener)

### Description

Used to remove alarm monitor

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| alarmListener | [DeviceStatusListener](#_DeviceStatusListener) | Alarm monitoring interface |

# Object Description

## StatusCode

### Description

Status of execution

### Fields

|  |  |
| --- | --- |
| **Field** | **Description** |
| SUCCESS | success |
| FAIL | failed |
| TIMEOUT | timeout |
| DISCONNECT | Bluetooth is disconnected |
| NOT\_ENABLE | Bluetooth is not open |
| PARAMETER\_ERROR | Parameter error |

## IResultCallback<T>

### Description

Callback interface

### Function

**void** onResultCallback(CallbackData<T> cd)

callback function

## CallbackData<T>

### Description

Callback object

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| status | Int | [Status](#_StatusCode) of execution |
| type | int | Interface Type, used to distinguish between operating interface |
| result | T | The result of execution |

## DeviceInfo

### Description

The result of Connnect Device.

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| hardwareVersion | String | Device version |
| deviceId | String | Device id |

## BatteryBean

### Description

The result of getting battery

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| chargingState | int | Charging or not，  0：not  1: charging |
| quantity | int | Percentage of battery, It’s disable when charging |

## RealTimeData

### Description

The result of getting sleep data (Real-time)

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| heartRate | short | Heart rate |
| breathRate | short | Breath rate |
| status | byte | [SleepStatusType](#_SleepStatusType_1) |
| statusValue | int | The value of status. Unit(second) |

## HistoryData

### Description

The result of getting sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| summary | Summary | History data [Summary](#_Summary) |
| detail | Detail | History data [Detail](#_Detail) |
| analy | Analysis | [Analysis](#_Analysis) |

## Summary

### Description

Summary of sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| recordCount | int | The length of Collecting.  Unit(minute)  Eg:  300. It means you collect for 300 minutes |
| startTime | int | Start time(timestamp). Unit(second) |

## Detail

### Description

Detail of sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| breathRate | int[] | Breath rate |
| heartRate | int[] | Heart rate |
| status | int[] | Status |
| statusValue | int[] | The value of status |

## Analysis

### Description

Analysis of sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| avgBreathRate | int | Average breath rate(n counts per min) |
| avgHeartRate | int | Average heart rate(n counts per min) |
| duration | int | Sleep duration(Unit:min) |
| wake | int | Duration of Awake)(Unit:min) |
| outOfBedDuration | int | Counts of leaving bed |
| discTime | int |  |
| algorithmVer | String | Algorithm version |
| sca\_array | byte[] | Sleep state array  0: device is not connected;  1: not in bed;  2: asleep;  3: awake; |
| breathPauseEvent | int[][] | Apnea event data. The array length indicates how many times apnea has occurred, where breathpauseevent [i] [0] indicates apnea time, in seconds.  Breathpauseevent [i] [1] indicates the duration of apnea, in seconds |
| outOfBedEvent | int[][] | Out of bed event data. The array length indicates the number of out of bed events, where outofbedevent [i] [0] indicates the out of bed time, in seconds.  outofbedevent [i] [1] indicates the duration of leaving the bed, in seconds |

## AlarmConfig

### Description

Alarm info

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| enable | boolean | Alarm switch |
| hour | int | alarm time hour[0-23] |
| minute | int | alarm time minute[0-59] |
| duration | int | Monitoring time(Unit:min) |

## Birthday

### Description

birthday

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| year | int | year |
| month | int | month of year[1-12] |
| day | int | day of month |

## SleepStatusType

### Description

Status value of monitoring

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value** | **Description** |
| SLEEP\_OK | byte | 0x00 | normal |
| SLEEP\_INIT | byte | 0x01 | init |
| SLEEP\_B\_STOP | byte | 0x02 | apnea |
| SLEEP\_H\_STOP | byte | 0x03 | Heartbeat pause |
| SLEEP\_BODYMOVE | byte | 0x04 | Body movement |
| SLEEP\_LEAVE | byte | 0x05 | Leaving bed |
| SLEEP\_TURN\_OVER | byte | 0x06 | Turning over |
| SLEEP\_BODYMOVE\_TEMP | byte | 0x07 | Amplitude of body motion |
| SLEEP\_INVALID | byte | -1 | invalid |

## DeviceStatusListener

### Interface Description

Device Status Interface

### Constant Field Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value** | **Description** |
| STATUS\_LOW\_POWER | byte | 0x01 | Low power |
| STATUS\_START\_COLLECT | byte | 0x02 | Collection Start (Normal Start will not be notified) |
| STATUS\_STOP\_COLLECT | byte | 0x03 | Acquisition End (Normal End will not be notified) |
| STATUS\_BREATH\_PAUSE | byte | 0x04 | Apnea alarm |
| STATUS\_OUT\_OF\_BED | byte | 0x05 | Out of bed alarm |
| STATUS\_STOP\_ALARM | byte | 0x06 | End Alarm (End All) |

### Function

**void** onStatusTriggered(String address, **byte** status, **byte** statusValue)

In the device status callback method, status is the field constant in DeviceStatusListener, statusValue is the status value, and when the status is STATUS\_LOW\_POWER, the status value represents the percentage of electricity (0-100) in other status, which is meaningless.