|  |
| --- |
|  |
| **Z400T Android SDK Description** |
| **V1.0** |
|  |
| **Author: WangYong** |
| **2022/06/09** |

# Change log

|  |  |  |
| --- | --- | --- |
| **Date** | **Log** | **Author** |
| 2022-06-09 | Create | WangYong |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |

# Catalog

[Change log 1](#_Toc105774104)

[Catalog 2](#_Toc105774105)

[Android SDK Intro 5](#_Toc105774106)

[1. Function and Purpose 5](#_Toc105774107)

[Integration 5](#_Toc105774108)

[1 .SDK framework 5](#_Toc105774109)

[API 6](#_Toc105774110)

[1.API initialization 6](#_Toc105774111)

[Description 6](#_Toc105774112)

[Parameters 6](#_Toc105774113)

[2. Connnect Device 6](#_Toc105774114)

[Description 6](#_Toc105774115)

[Parameters 6](#_Toc105774116)

[3. Add device Bluetooth connection status callback 7](#_Toc105774117)

[Description 7](#_Toc105774118)

[Parameters 7](#_Toc105774119)

[4. Remove device Bluetooth connection status callback 7](#_Toc105774120)

[Description 7](#_Toc105774121)

[Parameters 7](#_Toc105774122)

[5. Get Battery 8](#_Toc105774123)

[Description 8](#_Toc105774124)

[Parameters 8](#_Toc105774125)

[6. Get Device Version 8](#_Toc105774126)

[Description 8](#_Toc105774127)

[Parameters 8](#_Toc105774128)

[7. Get Device Collect Status 8](#_Toc105774129)

[Description 8](#_Toc105774130)

[Parameters 8](#_Toc105774131)

[8. Stop Monitoring/Collecting 9](#_Toc105774132)

[Description 9](#_Toc105774133)

[Parameters 9](#_Toc105774134)

[9. Get Sleep Data (Real-time) 9](#_Toc105774135)

[Description 9](#_Toc105774136)

[Parameters 9](#_Toc105774137)

[10. Add real-time data callback 9](#_Toc105774138)

[Parameters 10](#_Toc105774139)

[Parameters 10](#_Toc105774140)

[11. Remove real-time data callback 10](#_Toc105774141)

[Parameters 10](#_Toc105774142)

[Parameters 10](#_Toc105774143)

[12. Stop Getting Sleep Data 10](#_Toc105774144)

[Description 10](#_Toc105774145)

[Parameters 10](#_Toc105774146)

[13. Obtain temperature and humidity data 11](#_Toc105774147)

[Description 11](#_Toc105774148)

[Parameters 11](#_Toc105774149)

[14. Get Sleep Report 11](#_Toc105774150)

[Description 11](#_Toc105774151)

[Parameters 11](#_Toc105774152)

[15. Firmware Update 1 11](#_Toc105774153)

[Description 11](#_Toc105774154)

[Parameters 12](#_Toc105774155)

[16. Firmware Update 2 12](#_Toc105774156)

[Description 12](#_Toc105774157)

[Parameters 12](#_Toc105774158)

[Object Description 12](#_Toc105774159)

[StatusCode 12](#_Toc105774160)

[Description 12](#_Toc105774161)

[Fields 13](#_Toc105774162)

[IResultCallback<T> 13](#_Toc105774163)

[Description 13](#_Toc105774164)

[Function 13](#_Toc105774165)

[CallbackData<T> 13](#_Toc105774166)

[Description 13](#_Toc105774167)

[Fields 13](#_Toc105774168)

[LoginBean 14](#_Toc105774169)

[Description 14](#_Toc105774170)

[Fields 14](#_Toc105774171)

[BatteryBean 14](#_Toc105774172)

[Description 14](#_Toc105774173)

[Fields 14](#_Toc105774174)

[CollectStatus 14](#_Toc105774175)

[Description 14](#_Toc105774176)

[Fields 14](#_Toc105774177)

[RealTimeData 15](#_Toc105774178)

[Description 15](#_Toc105774179)

[Fields 15](#_Toc105774180)

[TempData 16](#_Toc105774181)

[Description 16](#_Toc105774182)

[Fields 16](#_Toc105774183)

[HistoryData 16](#_Toc105774184)

[Description 16](#_Toc105774185)

[Fields 16](#_Toc105774186)

[Summary 16](#_Toc105774187)

[Description 16](#_Toc105774188)

[Fields 16](#_Toc105774189)

[Detail 17](#_Toc105774190)

[Description 17](#_Toc105774191)

[Fields 17](#_Toc105774192)

[Analysis 18](#_Toc105774193)

[Description 18](#_Toc105774194)

[Fields 18](#_Toc105774195)

[SleepStatusType 22](#_Toc105774196)

[Description 22](#_Toc105774197)

[Fields 22](#_Toc105774198)

# Android SDK Intro

## Function and Purpose

Device 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 |
| z400tsdk.jar | Z400T sdk library |
| libz400t.so | Algorithm library |

# API

## 1.API initialization

Z400THelper.getInstance(Context mContext);

### Description

Z400THelper Initialization

### Parameters

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

## Connnect Device

**public** **void** login(String deviceName, String address, DeviceType deviceType, **int** userId, **int** timeout, IResultCallback<LoginBean> cb)

### Description

Connect Pillow and setting userId

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceName | String | deviceName |
| address | String | Bluetooth address |
| deviceType | enum | ***DEVICE\_TYPE\_Z400T\_2*** |
| userId | int | userId does not belong to Sleepace.  userId belong to partner  **Why need it：**  Pillow 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 |
| timeout | int | Timeout, Unit(Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[LoginBean](#_LoginBean)> | Callback function, if success,return [LoginBean](#_LoginBean) Obj |

## Add device Bluetooth connection status callback

**public** **void** addConnectionStateCallback(IConnectionStateCallback stateCallback)

### Description

Used to add device Bluetooth connection status callback

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| stateCallback | IConnectionStateCallback | Callback function |

## Remove device Bluetooth connection status callback

**public** **void** removeConnectionStateCallback(IConnectionStateCallback stateCallback)

### Description

Used to delete device Bluetooth connection status callback

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| stateCallback | IConnectionStateCallback | Callback function |

## Get Battery

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

### Description

Get battery

### Parameters

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

## Get Device Version

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

### Description

Get current version of device

### Parameters

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

## Get Device Collect Status

**public** **void** getCollectStatus(**int** timeout, IResultCallback<[CollectStatus](#_CollectStatus)> cb)

### Description

get device collect status

### Parameters

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

## Stop Monitoring/Collecting

**public** **void** stopCollection(**int** timeout, IResultCallback<Void> cb)

### Description

Stop Monitoring/Collecting

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| timeout | int | Timeout, Unit(Millisecond) |
| cb | IResultCallback<Void> | Callback function |

## Get Sleep Data (Real-time)

**public** **void** startRealTimeData(**int** timeout, IResultCallback<Void> cb)

### Description

After successfully calling this method, the equipment will report the monitoring data in real time, such as heart rate, respiration, temperature, humidity, etc. This data is received by adding a real-time data callback interface.

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| timeout | int | Timeout, Unit(Millisecond) |
| cb | IResultCallback<Void> | Callback function |

## Add real-time data callback

**public** **void** addRealtimeDataCallback(IResultCallback<[RealTimeData](#_RealTimeData_1)> callback)

### Parameters

Used to receive monitoring data reported by device

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| cb | IResultCallback<[RealTimeData](#_RealTimeData_1)> | Callback function |

## Remove real-time data callback

**public** **void** removeRealtimeDataCallback(IResultCallback<[RealTimeData](#_RealTimeData_1)> callback)

### Parameters

Used to delete real-time data callback

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| cb | IResultCallback<[RealTimeData](#_RealTimeData_1)> | Callback function |

## Stop Getting Sleep Data

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

### Description

Stop getting real-time data

### Parameters

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

## Obtain temperature and humidity data

**public** **void** getTempData(**int** timeout, IResultCallback<[TempData](#_TempData)> cb)

### Description

Used to obtain temperature and humidity data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| timeout | int | Timeout, Unit(Millisecond) |
| cb | IResultCallback<[TempData](#_TempData)> | Callback function |

## Get Sleep Report

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

### Description

Get sleep report

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| 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 |

## Firmware Update 1

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

### Description

Firmware Update

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| 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 |

## Firmware Update 2

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

### Description

Firmware Update

### Parameters

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

# Object Description

## StatusCode

### Description

Status of execution

### Fields

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

## IResultCallback<T>

### Description

Callback interface

### Function

**void** onDataCallback(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 |

## LoginBean

### 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 |

## CollectStatus

### Description

device collect status

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| status | byte | Collect or not，  0：not  1: collecting |
| timestamp | int | Timestamp at start of collection |

## 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) |
| statusValue | int | The value of status. Unit(Millisecond) |
| sleepFlag | int | Asleep or not  1: asleep  0: not |
| wakeFlag | int | Awake or not  1: awake  0: not |
| eTemp | float | Ambient temperature (required equipment support) |
| eWet | int | Ambient humidity (required equipment support) |
| eLight | int | Ambient light intensity (required equipment support) |
| eCo2 | int | Environmental carbon dioxide content (required equipment support) |
| eNoise | short | Ambient noise (required equipment support) |

## TempData

### Description

Temperature and humidity data

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| temp | byte | Temperature in degrees Celsius |
| humidity | byte | humidity |

## HistoryData

### Description

The result of getting sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| summary | Summary | History data [Summary](#_Summary) |
| detail | Detail | History data [Detail](#_Detail) |
| analysis | 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) |
| stopMode | int | How to stop collecting：  0: Call the method “Stop Collecting”  1: stop automatically if you leave the bed for an hour  2: Error(a、Collect more than 24 hours，b、Pillow shutdown c、upgrade)  3: restart |
| timeStep | int | Record interval (default 60s, ie: 1 minute a time a point) |
| timezone | int | Timezone |

## 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 |
| eHumidity | int[] | Humidity(required equipment support) |
| eTemp | int[] | Temperature(required equipment support) |

## 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) |
| fallAlseepAllTime | int | Fall asleep time(Unit:min) |
| wakeAndLeaveBedBeforeAllTime | int | Duration of awake before getting up(Unit:min) |
| leaveBedTimes | int | Counts of leaving bed |
| trunOverTimes | int | Counts of turning over |
| bodyMovementTimes | int | Counts of body movement |
| heartBeatPauseTimes | int | Counts of Heat beat Pause |
| breathPauseTimes | int | Counts of apnea |
| deepSleepPerc | int | Deep sleep percentage |
| inSleepPerc | int | Mid sleep percentage |
| lightSleepPerc | int | Light sleep percentage |
| wakeSleepPerc | int | Awake percentage |
| duration | int | Sleep duration(Unit:min) |
| wakeTimes | int | Awake times |
| lightSleepAllTime | int | Duration of Light sleep(Unit:min) |
| inSleepAllTime | int | Duration of Mid sleep(Unit:min) |
| deepSleepAllTime | int | Duration of Deep sleep)(Unit:min) |
| wake | int | Duration of Awake)(Unit:min) |
| breathPauseAllTime | int | Duration of Apnea(Unit:seconds) |
| heartBeatPauseAllTime | int | Duration of heart beat pause)(Unit:seconds) |
| outOfBedDuration | int | Duration of leaving bed(Unit:min) |
| maxHeartBeatRate | int | Maximum heart rate(n counts per min) |
| maxBreathRate | int | Maximum breath rate(n counts per min) |
| minHeartBeatRate | int | Minimum heart rate)(n counts per min) |
| minBreathRate | int | Minimum breath rate(n counts per min) |
| heartBeatRateFastAllTime | int | Duration of tachycardia(Unit:seconds) |
| heartBeatRateSlowAllTime | int | Duration of bradycardia(Unit:seconds) |
| breathRateFastAllTime | int | Duration of tachypnea(Unit:seconds |
| breathRateSlowAllTime | int | Duration of bradypnea(Unit:seconds) |
| sleepScore | int | Score  90>=score<=100 Bravo!  80>=score<90 Good!  60>=score<80, average!  score <60 Bad |
| sleepCurveArray | float[] | Example:  [0.212,1.231,2.111,0.212,1.231,2.111,....]  0: awake 0 ~ 1: light sleep 1 ~ 2: moderate sleep 2 ~ 3: deep sleep  Drawing sleep curve (Unit:min) |
| sleepCurveStatusArray | ~~short[]~~ | ~~Sleep Event Flag (Unit:min)~~ |
| breathRateStatusAry | int[] | Apnea, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: Duration of Apnea in this minute(Unit:seconds) |
| heartRateStatusAry | int[] | Heart beat pause, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: Duration of Heat beat Pause in this minute(Unit:seconds) |
| leftBedStatusAry | int[] | Leave bed, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: Duration of leaving bed in this minute(Unit:seconds) |
| turnOverStatusAry | int[] | Turn over, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: the times of turning over |
| scaArray | byte[] | Analytical Sleep Status  0:awake, 1:light Sleep, 2:middle sleep, 3:deep sleep |
| algorithmVer | String | Algorithm version |
| fallsleepTimeStamp | int | The time you fall asleep(timestamp) |
| wakeupTimeStamp | int | The time you wake up(timestamp) |
| reportFlag | int | 1. Long report(>3h) 2. Short report(>10m && <3h) |
| md\_body\_move\_decrease\_scale | short | Score Deduction:Score Deduction due to body movement |
| md\_leave\_bed\_decrease\_scale |  | Score Deduction:Score Deduction due to the times of leaving bed |
| md\_wake\_cnt\_decrease\_scale | short | Score Deduction:Score Deduction due to the wake count |
| md\_start\_time\_decrease\_scale | short | Score Deduction:Score Deduction due to sleeping time (too late) |
| md\_fall\_asleep\_time\_decrease\_scale | short | Score Deduction:Score Deduction due to long falling sleep time |
| md\_perc\_deep\_decrease\_scale | short | Score Deduction:Score Deduction due to the deep sleep |
| md\_sleep\_time\_decrease\_scale | short | Score Deduction due to sleeping time too short |
| md\_sleep\_time\_increase\_scale | short | Score Deduction due to sleeping time too long |
| md\_breath\_stop\_decrease\_scale | short | Score Deduction:Score Deduction due to breathing stop |
| md\_heart\_stop\_decrease\_scale | short | Score Deduction:Score Deduction due to Heart beat stop |
| md\_heart\_low\_decrease\_scale | short | Score Deduction:Score Deduction due to slow heart beat |
| md\_heart\_high\_decrease\_scale | short | Score Deduction:Score Deduction due to Rapid heart beat |
| md\_breath\_low\_decrease\_scale | short | Score Deduction:Score Deduction due to slow breathing |
| md\_breath\_high\_decrease\_scale | short | Score Deduction:Score Deduction due to rapid breathing |

## 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 |