|  |
| --- |
|  |
| **RestOn Android SDK Description** |
| **V1.0** |
|  |
| **Author: 闭周健** |
| **2017/07/10** |

# Change log

|  |  |  |
| --- | --- | --- |
| **Date** | **Log** | **Author** |
| 2017-09-13 | Create | 闭周健 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **内部文档 严禁外传** |

# Catalog

[Change log 1](#_Toc7853)

[Catalog 2](#_Toc5905)

[Android SDK Intro 5](#_Toc21934)

[1. Function and Purpose 5](#_Toc14378)

[Integration 5](#_Toc13486)

[1 .SDK framework 5](#_Toc5740)

[2 .Integration 5](#_Toc25025)

[Eclipse Config 5](#_Toc2756)

[API 7](#_Toc6310)

[1.API initialization 7](#_Toc5742)

[Description 7](#_Toc8247)

[Parameters 7](#_Toc19753)

[2. Connnect Device 7](#_Toc11358)

[Description 7](#_Toc6327)

[Parameters 7](#_Toc9464)

[3. Get Battery 8](#_Toc2377)

[Description 8](#_Toc4420)

[Parameters 8](#_Toc10748)

[4. Get Device Version 8](#_Toc8074)

[Description 8](#_Toc19119)

[Parameters 8](#_Toc1285)

[5. Set up automatic Collection 9](#_Toc17696)

[Description 9](#_Toc4132)

[Parameters 9](#_Toc17162)

[6. Start Monitoring/Collecting 9](#_Toc3250)

[Description 9](#_Toc28187)

[Parameters 9](#_Toc7793)

[7. Stop Monitoring/Collecting 10](#_Toc8369)

[Description 10](#_Toc27997)

[Parameters 10](#_Toc9607)

[8. Get Collection Status 10](#_Toc6447)

[Description 10](#_Toc17843)

[Parameters 10](#_Toc21982)

[9. Get Sleep Data (Real-time) 10](#_Toc8102)

[Description 11](#_Toc24872)

[Parameters 11](#_Toc22278)

[10. Stop Getting Sleep Data 11](#_Toc16749)

[Description 11](#_Toc27813)

[Parameters 11](#_Toc100)

[11. Get The Signal Strength (Real-time) 11](#_Toc18145)

[Description 11](#_Toc26736)

[Parameters 11](#_Toc30242)

[12. Stop Getting The Signal Strength 12](#_Toc16797)

[Description 12](#_Toc24166)

[Parameters 12](#_Toc13201)

[13. Get Sleep Report 12](#_Toc30189)

[Description 12](#_Toc23539)

[Parameters 12](#_Toc2692)

[14. Firmware Update 1 13](#_Toc25876)

[Description 13](#_Toc18955)

[Parameters 13](#_Toc1761)

[15. Firmware Update 2 13](#_Toc28700)

[Description 13](#_Toc27751)

[Parameters 13](#_Toc18945)

[四、 Object Description 14](#_Toc2672)

[StatusCode 14](#_Toc2914)

[Description 14](#_Toc6139)

[Fields 14](#_Toc335)

[DeviceCode 14](#_Toc8106)

[Description 14](#_Toc24199)

[Fields 14](#_Toc6483)

[IResultCallback<T> 15](#_Toc29498)

[Description 15](#_Toc13650)

[Function 15](#_Toc29589)

[CallbackData<T> 15](#_Toc3931)

[Description 15](#_Toc19406)

[Fields 15](#_Toc27858)

[LoginBean 15](#_Toc13722)

[Description 15](#_Toc2810)

[Fields 16](#_Toc8740)

[BatteryBean 16](#_Toc25461)

[Description 16](#_Toc25864)

[Fields 16](#_Toc16874)

[RealTimeData 16](#_Toc12257)

[Description 16](#_Toc7126)

[Fields 16](#_Toc11100)

[OriginalData 17](#_Toc25544)

[Description 17](#_Toc8703)

[Fields 17](#_Toc7065)

[HistoryData 17](#_Toc4606)

[Description 17](#_Toc17767)

[Fields 18](#_Toc26320)

[Summary 18](#_Toc4440)

[Description 18](#_Toc24680)

[Fields 18](#_Toc22122)

[Detail 18](#_Toc7487)

[Description 19](#_Toc5856)

[Fields 19](#_Toc9611)

[Analysis 19](#_Toc5394)

[Description 19](#_Toc2394)

[Fields 19](#_Toc25502)

[SleepStatusType 23](#_Toc28782)

[Description 23](#_Toc29780)

[Fields 23](#_Toc12813)

# Android SDK Intro

## Function and Purpose

RestOn 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 |
| HeartBreathDeviceCore.jar | Realtime data core |
| RestonSdk.jar | Reston SDK |
| SdkAlgorithm.jar | Algorithm call library |
| libalgorithm.so | Algorithm library |

## 2 .Integration

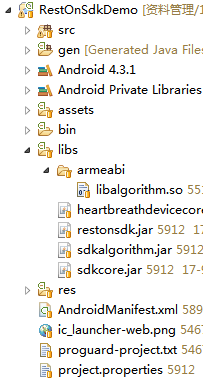
There are many Android development tools, and here we introduce the engineering configuration method of Sleepace SDK with Eclipse.

## Eclipse Config

**Step 1**：

In the project to create a "libs" folder, copy SdkCore.jar, HeartBreathDeviceCore.jar, RestonSdk.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"*/>

# API

## 1.API initialization

RestOnHelper.getInstance(Context mContext);

### Description

RestOnHelper Initialization

### Parameters

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

## Connnect Device

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

### Description

Connect Reston and setting userId

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceName | String | deviceName |
| address | String | Bluetooth address |
| deviceCode | String | [DeviceCode](#_StatusCode) |
| 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 |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[LoginBean](#_LoginBean)> | Callback function, if success,return [LoginBean](#_LoginBean) Obj |

## 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 |

## Set up automatic Collection

**public** **void** setAutoCollection(**boolean** enable, **int** hour, **int** minute, **int** repeat, **int** timeout, IResultCallback<Void> cb)

### Description

Set up automatic collection

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| enable | boolean | The automatic monitoring work or not |
| hour | int | The hour of work time (0 - 23) |
| minute | int | The minute of work time (0 - 59) |
| repeat | int | repeat mode，For example: 00000111, from right to left, represents Monday, Tuesday, Wednesday respectively, if the bit is 1, that means it will repeat at the same day, otherwise, it will not repeat. 127(decimalism) = 1111111(Binary), it means repeating from Monday to Sunday. 16 = 0010000, it means just repeating on Friday |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IDataCallbac](#_IDataCallback<T>)k<Void> | Callback function |

## Start Monitoring/Collecting

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

### Description

Start monitoring/collecting

### Parameters

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

**public** **void** getCollectionStatus(**int** timeout, IResultCallback<Byte> cb)

### Description

Get collection status, monitoring or not

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<Byte> | Callback function,  Collection status 1:Collecting,0：not |

## Get Sleep Data (Real-time)

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

### Description

Get Real-time Data

### Parameters

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

## Get The Signal Strength (Real-time)

**public** **void** startOriginalData(**int** timeout, IResultCallback<OriginalData> cb)

### Description

Get the signal strength

### Parameters

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

## Stop Getting The Signal Strength

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

### Description

Stop Getting The Signal Strength

### Parameters

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

## Environment data

**Public void** getEnvironmentalData(**int** timeout, IResultCallback<EnvironmentData> cb)

### Description

get environment data

### Parameters

|  |  |  |
| --- | --- | --- |
| **字段** | **类型** | **说明** |
| timeout | int | Timeout, Unit((Millisecond) |
| cb | [IResultCallback](#_IDataCallback<T>)<[EnvironmentData](#_EnvironmentData)> | Callback function |

# Object Description

## StatusCode

### Description

Status of execution

### Fields

|  |  |
| --- | --- |
| **Field** | **Description** |
| STATUS\_SUCCESS | success |
| STATUS\_FAILED | failed |
| STATUS\_TIMEOUT | timeout |
| STATUS\_DISCONNECT | Bluetooth is disconnected |
| STATUS\_BLUETOOTH\_NOT\_OPEN | Bluetooth is not open |
| STATUS\_PARAMETER\_ERROR | Parameter error |

## DeviceCode

### Description

Device code

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **value** | **Description** |
| Z2\_9\_0 | 9-0 | RestOn(Z200) |
| Z4\_22\_3 | 22-3 | RestOn(Z400T)\_with Temperature & Humidity |
| Z4\_22\_4 | 22-4 | RestOn(Z400T)\_without Temperature & Humidity |

## IResultCallback<T>

### Description

Callback interface

### Function

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

callback function

## CallbackData<T>

### Description

Callback object

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| status | short | [Status](#_StatusCode) of execution |
| callbackType | short | 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 |

## 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(second) |
| sleepFlag | int | Asleep or not  1: asleep  0: not |
| wakeFlag | int | Awake critical logo :  1: yes  0: no |

## OriginalData

### Description

The result of getting signal strength

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| heartRate | float[] | Heart rate |
| breathRate | Float[] | Breath rate |

## 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) |
| 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、reston 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 |
| md\_perc\_effective\_sleep\_decrease\_scale | short | Score Deduction:Score Deduction due to good sleeping (ratio of middle sleep/deep sleep) |

## 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 |

## EnvironmentData

### Description

The result of getting environment data

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| temperature | int | Temperature,Unit(Celsius) |
| humidity | int | humidity |