

## 开发指南

### 工程配置

### 开发注意事项

#### 1 数据模型

- 1.1 CRPDiscovery
- 1.2 CRPStepModel
- 1.3 CRPSleepModel
- 1.4 CRPUserInfoModel
- 1.5 CRPNewTrainingModel
- 1.6 CRPSportRecord
- 1.7 CRPTrainingGoalsModel
- 1.8 CRPTrainingGoalStateModel
- 1.9 CRPTrainingRecordModel
- 1.10 CRPSleepRecordModel
- 1.11 CRPTimingHRRecordModel
- 1.12 CRPTimingHRVRecordModel
- 1.13 CRPTimingO2RecordModel
- 1.14 CRPTimingStepsRecordModel
- 1.15 CRPContinueActivitysRecordModel
- 1.16 CRPActivityReminderModel
- 1.17 CRPHRRemind
- 1.18 CRPHeartRecordModel
- 1.19 CRPO2RecordModel
- 1.20 CRPHRVRecordModel
- 1.21 CRPGsensorInfoModel
- 1.22 CRPBatteryInfoModel
- 1.23 CRPActivityReminderStateModel
- 1.24 CRPSleepDetailModel
- 1.25 CRPSleepTemperatureDataModel
- 1.26 CRPRingInfoModel
- 1.27 CRPStressRecordModel

#### 2 代理方法

- 2.1 返回当前的戒指的连接状态

- 2.2 返回当前蓝牙状态
- 2.3 接收实时的运动计步数据
- 2.4接收心率测量结果(单次测量心率)
- 2.5 接收实时心率数据
- 2.6 接收心率变异性测量结果
- 2.7 接收血氧测量结果
- 2.8 接收固件的升级进度和状态
- 2.9 收到运动数据记录列表(可选)
- 2.10 收到运动模式状态(可选)
- 2.11 收到佩戴状态(可选)
- 2.12 收到日常目标提醒(可选)
- 2.13 收到久坐提醒(可选)
- 2.14 收到锻炼中的目标达成(可选)
- 2.15 接收压力测量结果

### 3 戒指扫描及连接

- 3.1 初始化
- 3.2 戒指扫描
- 3.3 取消扫描
- 3.4 连接
- 3.5 解除绑定

### 4 戒指交互

- 4.0 同步时间
- 4.1 固件升级
- 4.2 查询戒指电量
- 4.3 用户信息
- 4.4 活动步数
- 4.5 睡眠
- 4.6 锻炼目标
- 4.7 久坐提醒
- 4.8 心率预警
- 4.9 心率
- 4.10 血氧
- 4.11 HRV疲劳度

- 4.12 锻炼数据
- 4.13 获取活动量
- 4.14 佩戴状态
- 4.15 关机
- 4.16 恢复出厂设置
- 4.17 测试指令
- 4.18 体温
- 4.19 测量状态
- 4.20 重启
- 4.21 压力
- 4.22 戒指配置项
- 5 版本更新日志

## 开发指南

### 工程配置

- 导入framework包

导入文件夹下CRPSmartRing.framework至工程Embedded Binaries中。

### 开发注意事项

- 配置权限

在Info.plist中配置权限:

在Required background modes下添加App communicates using CoreBluetooth；（蓝牙权限）

在App Transport Security Settings下添加Allow Arbitrary Loads并设置YES(网络访问权限)；

添加Privacy - Bluetooth Always Usage Description的蓝牙权限字段

# 1 数据模型

## 1.1 CRPDiscovery

Modifier	Name	Explain
String	LocalName	设备名
Int	RSSI	Rssi
String	Mac	Mac地址
CRPRingInfoModel?	info	戒指设置项信息
Int?	hide	是否隐藏

## 1.2 CRPStepModel

Modifier	Name	Explain
Int	steps	步数
Int	distance	里程(米)
Int	calory	卡路里
Int	time	时间(值大于-1时为运动统计时间， 值为-1是表示该设备不支持运动时间统计)

## 1.3 CRPSleepModel

Modifier	Name	Explain
Int	deep	深度睡眠
Int	light	浅度睡眠
Int	rem	快速眼动睡眠
(Dictionary<String,String>)	detail	睡眠数据详情

睡眠数据详情

Modifier	Name	Explain
String	type	状态 0: 清醒 1:浅睡 2: 深睡 3: 快速眼动
String	total	时长(分钟)
String	start	开始时间
String	end	结束时间

## 1.4 CRPUserInfoModel

Modifier	Name	Explain
Int	height	身高
Int	weight	体重
Int	age	年龄
GenderOption	gender	性别
Int	stepLength	步长

## 1.5 CRPNewTrainingModel

Modifier	Name	Explain
Int	id	序号
Int	starttime	开始时间(时间戳)
Int	endtime	结束时间(时间戳)
Int	vaildTime	有效时间（秒）
Int	hrAvg	平均心率
CRPTrainingType	type	运动类型
Int	Step	步数
Int	Distance	距离
Int	kcal	卡路里
(Int)	heartRate	心率数据
CRPTrainingGoalType	goalType	运动目标类型
Int	goal	目标值
Int	hrAddress	心率地址
Int	hrLength	心率长度

## 1.6 CRPSportRecord

Modifier	Name	Explain
Int	id	序号
Int	startTime	开始时间(时间戳)
Int	type	类型

## 1.7 CRPTrainingGoalsModel

Modifier	Name	Explain
Int	steps	步数
Int	distance	里程(米)
Int	kcal	卡路里
Int	exerciseTime	锻炼时间(分钟)

## 1.8 CRPTrainingGoalStateModel

Modifier	Name	Explain
Bool	open	开关
(Int)	weekday	锻炼日数组（参考CRPWeekday）例： (CRPWeekDay.sun.rawValue, CRPWeekDay.thu.rawValue, CRPWeekDay.mon.rawValue)

## 1.9 CRPTrainingRecordModel

Modifier	Name	Explain
Int	day	0:今天； 1： 昨天； 2： 前天， 以此类推
Int	step	步数
Int	kcal	卡路里（小卡）
Int	distance	距离(厘米)
Int	exerciseTime	锻炼时间(分钟)

## 1.10 CRPSleepRecordModel

Modifier	Name	Explain
Int	day	0:今天; 1: 昨天; 2: 前天, 以此类推
Int	deep	深度睡眠
Int	light	浅度睡眠
Int	rem	快速眼动睡眠
(Dictionary<String,String>)	detail	睡眠数据详情

## 1.11 CRPTimingHRRecordModel

Modifier	Name	Explain
Int	day	0:今天; 1: 昨天; 2: 前天, 以此类推
(Int)	hearts	心率数据(288个, 5分钟一个)

## 1.12 CRPTimingHRVRecordModel

Modifier	Name	Explain
Int	day	0:今天; 1: 昨天; 2: 前天, 以此类推
(Int)	hrvs	心率变异性数据(288个, 5分钟一个)

## 1.13 CRPTimingO2RecordModel

Modifier	Name	Explain
Int	day	0:今天; 1: 昨天; 2: 前天, 以此类推
(Int)	o2s	血氧数据(288个, 5分钟一个)

## 1.14 CRPTimingStepsRecordModel



Modifier	Name	Explain
Int	day	0:今天; 1: 昨天; 2: 前天, 以此类推
(Int)	steps	半小时步数统计数据(48个, 半小时一个)

## 1.15 CRPContinueActivitysRecordModel

Modifier	Name	Explain
Int	day	0:今天; 1: 昨天; 2: 前天, 以此类推
(Int)	activitys	活动量统计数据(1440个, 一分钟一个)

## 1.16 CRPActivityReminderModel

Modifier	Name	Explain
Bool	open	开关
Int	period	久坐提醒周期(分钟)
Int	steps	最大步数
Int	startHour	开始时间(24小时制)
Int	endHour	结束时间(24小时制)

## 1.17 CRPHRRRemind

Modifier	Name	Explain
Bool	isRemind	开关 (是否提醒)
Int	max	心率提醒值

## 1.18 CRPHeartRecordModel

Modifier	Name	Explain
Int	value	测量结果
Int	time	时间戳

## 1.19 CRPO2RecordModel

Modifier	Name	Explain
Int	value	测量结果
Int	time	时间戳

## 1.20 CRPHRVRecordModel

Modifier	Name	Explain
Int	hrv	心率变异性
Int	time	时间戳

## 1.21 CRPGsensorInfoModel

Modifier	Name	Explain
Int	x	x轴偏移量
Int	y	y轴偏移量
Int	z	z轴偏移量

## 1.22 CRPBatteryInfoModel

Modifier	Name	Explain
Int	value	电量
Int	voltage	电压(单位mv)

## 1.23 CRPActivityReminderStateModel

Modifier	Name	Explain
Bool	wearState	佩戴穿戴状态 （00:未佩戴， 01： 佩戴）
Int	time	时间(分钟)
Int	steps	期间步数

## 1.24 CRPSleepDetailModel

Modifier	Name	Explain
CRPSleepType	type	睡眠状态
Int	hour	小时
Int	minute	分钟

## 1.25 CRPSleepTemperatureDataModel

Modifier	Name	Explain
Int	day	0:今天； 1： 昨天； 2： 前天， 以此类推
(Int)	tempeartures	体温数据(288个， 5分钟一个)

## 1.26 CRPRingInfoModel

Modifier	Name	Explain
Int	color	颜色
Int	size	尺寸
Int	type	类型

## 1.27 CRPStressRecordModel

Modifier	Name	Explain
Int	stress	压力值
Int	time	时间戳

## 2 代理方法

### 2.1 返回当前的戒指的连接状态

```
didState(_ state: CRPState)
```

### 2.2 返回当前蓝牙状态

```
didBluetoothState(_ state: CRPBluetoothState)
```

### 2.3 接收实时的运动计步数据

使用getStep或者戒指产生新的步数都会回调

```
receiveSteps(_ model: CRPStepModel)
```

### 2.4接收心率测量结果(单次测量心率)

```
receiveHeartRate(_ heartRate: Int)
```

### 2.5 接收实时心率数据

一般只用于显示不用于存储

```
receiveRealTimeHeartRate(_ heartRate: Int)
```

## 2.6 接收心率变异性测量结果

```
receiveHRV(_ hrv: Int)
```

## 2.7 接收血氧测量结果

```
receiveSpO2(_ o2: Int)
```

## 2.8 接收固件的升级进度和状态

```
receiveOTA(_ state: CRPOTAState, _ progress: Int)
```

## 2.9 收到运动数据记录列表(可选)

(使用getTrainingRecordList或者戒指结束锻炼时都会回调这个方法)

```
receiveTrainingList(_ list: [CRPSportRecord])
```

## 2.10 收到运动模式状态(可选)

```
receiveTrainingState(_ state: CRPTrainingType)
```

## 2.11 收到佩戴状态(可选)

(00:未佩戴, 01: 佩戴)

```
receiveWearState(_ state: Int)
```

## 2.12 收到日常目标提醒(可选)

(state: 0(未提醒), 1(已提醒)收到日常目标达成后需要使用setReceiveGoalReminder方法通知戒指App已经通知用户)

```
receiveDailyGoal(_ type: CRPTrainingGoalType, state: Int)
```

## 2.13 收到久坐提醒(可选)

isReminder:(00: App不需要提醒, 01: App需要提醒)

wearState,time和step这三个数据只会在久坐提醒测试指令setActivityReminderReach调用的情况下才会有回复

wearState: (00:未佩戴, 01: 佩戴)

```
receiveActivityReminder(isReminder: Int, wearState: Int, time: Int, step: Int)
```

## 2.14 收到锻炼中的目标达成(可选)

(state: 0(未提醒), 1(已提醒)收到锻炼中目标达成后需要使用 setReceiveTrainingGoalReach方法通知戒指App已经通知用户)

```
receiveTrainingGoal(_ type: CRPTrainingGoalType, state: Int)
```

## 2.15 接收压力测量结果

```
receiveStress(_ stress: Int)
```

# 3 戒指扫描及连接

## 3.1 初始化

CRPSmartRingSDK是SDK的入口, CRPSmartRingSDK单例模式编写的; 设置delegate将初始化CRPSmartRingSDK并创建蓝牙(提示: 创建蓝牙是耗时操作,需提前设置)

```
CRPSmartRingSDK.sharedInstance.delegate = self
```

## 3.2 戒指扫描

在权限允许和蓝牙开启状态下，才能开始正常扫描。在扫描过程中发现戒指，通过scan()方法，通过progressHandler回调扫描中发现的戒指，completionHandler回调整个扫描中扫描到的所有戒指，可设置扫描时长,因蓝牙扫描操作是费时操作，建议扫描时长为10秒。

```
scan(_ duration: TimeInterval = 10, progressHandler:
scanProgressHandler?, completionHandler:
scanCompletionHandler?);
```

## 3.3 取消扫描

取消戒指扫描。

```
interruptScan()
```

## 3.4 连接

通过扫描获取的(CRPDiscovery),可根据CRPDiscovery的localName和Mac地址选择选择相应的戒指连接

```
connect(_ discovery: CRPDiscovery)
```

- 获取系统中已连接设备(获取到的是原始CBPeripheral，如需连接这个设备需要使用CBPeripheral去创建CRPDiscovery然后调用连接方法)

```
connectedPeripheral() -> [CBPeripheral]
```

## 3.5 解除绑定

解除戒指绑定

```
remove(_ handler: @escaping removeHandler)
```

- 重新连接

```
reConnect()
```

## 4 戒指交互

### 4.0 同步时间

保持戒指与App端时间一致

```
setTime()
```

### 4.1 固件升级

- 查询固件版本

查询当前戒指固件版本。

```
getSoftver(_ handler: @escaping stringHandler)
```

查询当前戒指Mac地址

```
getMac(_ handler: @escaping stringHandler)
```

- 查询新固件

通过传入当前版本的和mac地址获取时候有新版本固件可升级

```
checkLatest(_ version: String, _ mac: String, handler:  
@escaping versionHandler)
```

- Goodix固件升级

```
startGoodixUpgradeFromFile(zipPath: String)
```

- 中断升级

```
stopUpgrade()
```



## 4.2 查询戒指电量

查询戒指当前电量(电量100以上为充电中，入出现超过101的电量，表示正在充电，当前电量为后两位数，例：收到电量159，表示正在充电，戒指当前电量为59)

```
getBattery(_ handler: @escaping intHandler)
```

## 4.3 用户信息

- 获取用户信息

```
getUserinfo(_ handler: @escaping profileHandler)
```

- 设置用户信息

```
setUserinfo(_ Userinfo: CRPUserInfoModel)
```

## 4.4 活动步数

- 同步当前活动步数，通过receiveSteps回复数据

```
getSteps()
```

- 获取历史的统步数数据

```
getTrainingData(_ day: Int, _ handler: @escaping  
exerciseRecordHandler)
```

- 步数统计数据

获取步数统计(每半个小时一个值，总共48个值)

```
getStepArchiveData(_ day: Int, _ handler: @escaping  
fullDayStepsRecordHandler)
```

## 4.5 睡眠

获取睡眠数据

```
getSleepData(_ day: Int, _ handler: @escaping  
sleepRecordHandler)
```

设置当天睡眠数据(仅用于测试的指令)

```
setSleepData(data: [CRPSleepDetailModel])
```

## 4.6 锻炼目标

锻炼目标分平常日锻炼目标和锻炼日锻炼目标

- 获取平常日锻炼目标

```
getNormalTrainingGoal(_ handler: @escaping  
normalexerciseGoalDateHandler)
```

- 设置平常日锻炼目标

```
setNormalTrainingGoal(_ goals: CRPTrainingGoalsModel)
```

- 获取锻炼日目标

```
getTrainingDayGoal(_ handler: @escaping  
exerciseGoalDateHandler)
```

- 设置锻炼日目标

```
setTrainingDayGoal(_ model: CRPTrainingGoalStateModel, _ goals:  
CRPTrainingGoalsModel)
```

- 设置锻炼目标达成(仅用于测试指令，仅用于测试的指令，使用此方法后，会回调  
receiveDailyGoal(\_ type: CRPTrainingGoalType, state: Int))

```
setGoalReminder(type: CRPTrainingGoalType)
```

## 4.7 久坐提醒

- 查询久坐提醒设置

```
getActivityReminderInfo(_ handler: @escaping sitRemindHandler)
```

- 设置久坐提醒设置

```
setActivityReminder(_ ActivityReminder:  
CRPActivityReminderModel)
```

- 测试久坐提醒(仅用于测试指令, 使用此方法后, 戒指会通过receiveActivityReminder回调返回当前久坐提醒的相关信息)

```
setActivityReminderReach()
```

## 4.8 心率预警

- 获取心率预警值

```
getHeartRateRemind(_ handler: @escaping hrRemindHandler)
```

- 设置心率预警值

```
setHeartRateRemind(_ remind: CRPHRRemind)
```

## 4.9 心率

所有心率相关的实时数据都会通过receiveRealTimeHeartRate(\_ heartRate: Int)回调。

- 开始动态心率测量

启动单次心率测量

```
setStartSingleHR()
```

- 结束单次心率测量

结束心率测量。测量结果通过receiveHeartRate(\_ heartRate: Int)回调。(收到结果为255或者0表示心率测量中断)

```
setStopSingleHR()
```

- 获取单次心率测量历史记录

```
getHeartRecordData(_ handler: @escaping heartRecordDataHandler)
```

- 获取定时心率测量状态

```
getTimingHeartRateInterval(_ handler: @escaping intHandler)
```

- 开启定时测量心率  
戒指支持24小时定时测量心率，从0点0分开始测量，可以设置测量时间间隔（0为关闭，正数为开启,并设置为测量量时间间隔为：正数 \* 5分钟）

```
setTimingHeartRate(_ interval: Int)
```

- 获取定时测量心率数据

```
getTimingHeartRate(_ day: Int, _ handler: @escaping fullDayHRRecordHandler)
```

## 4.10 血氧

- 启动单次血氧测量

开始测量血氧。

```
setStartSpO2()
```

- 结束单次测量血氧

结束测量血氧，测量时间过短会导致无测量结果。结果通过receiveSpO2(\_ o2: Int)回调(收到结果为255或者0表示血氧测量中断)

```
setStopSpO2()
```

- 获取单次血氧测量历史记录

```
getO2RecordData(_ handler: @escaping o2RecordDataHandler)
```

- 获取定时血氧测量状态

```
getTimingO2Interval(_ handler: @escaping intHandler)
```

- 开启定时测量血氧  
戒指支持24小时定时测量血氧，从0点0分开始测量，可以设置测量时间间隔（0为关闭，正数为开启,并设置为测量量时间间隔为：正数 \* 5分钟）

```
setTimingO2(_ interval: Int)
```

- 获取定时血氧数据

```
getTimingO2(_ day: Int, _ hanler: @escaping  
fullDayO2RecordHandler)
```

## 4.11 HRV疲劳度

- 启动HRV单次测量

```
setStartHRV()
```

- 结束单次HRV测量，测量时间过短会导致无测量结果。结果通过receiveHRV(\_ hrv: Int)回调(收到结果为255或者0表示HRV测量中断)

```
setStopHRV()
```

- 获取单次HRV测量历史记录

```
getHRVRecord(_ handler: @escaping hrvRecordDataHandler)
```

- 获取定时HRV测量状态

```
getTimingHRVInterval(_ handler: @escaping intHandler)
```

- 开启定时测量HRV  
戒指支持24小时定时测量HRV，从0点0分开始测量，可以设置测量时间间隔（0为关闭，

正数为开启,并设置为测量量时间间隔为：正数 \* 5分钟)

```
setTimingHRV(_ interval: Int)
```

- 获取定时HRV测量数据

```
getTimingHRV(_ day: Int, _ handler: @escaping  
fullDayHRVRecordHandler)
```

## 4.12 锻炼数据

- 获取戒指支持的锻炼类型

```
getTrainingSupportList(_ handler: @escaping  
exerciseSupportListHandler)
```

- 设置锻炼状态

```
setTraining(state: CRPTrainingType, type: CRPTrainingGoalType,  
goal: Int)
```

- 获取当前锻炼状态

```
getTraingingState(_ handler: @escaping trainingStateHandler)
```

- 设置单次锻炼目标达成(仅用于测试的指令, 使用此方法后, 会回调  
receiveTrainingGoal(\_ type: CRPTrainingGoalType, state: Int))

```
setTrainingGoalReach(type: CRPTrainingGoalType)
```

- 获取历史锻炼记录,通过receiveTrainingList(\_ list: (CRPTrainingRecord))回调返回运动  
列表

```
getTrainingtRecordList()
```

- 获取相应锻炼详细数据(使用receiveTrainingList中返回的列表中的id去查询)

```
getTrainingRecordData(id: Int, _ handler: @escaping  
sportDetailHandler)
```

- 设置锻炼数据(仅用于测试)

```
setTrainingData(step: Int, cal: Int, exerciseTime: Int,  
distance: Int)
```

## 4.13 获取活动量

获取活动量记录(一分钟一个值, 总共1440个)

```
getActivityArchiveData(_ day: Int, _ handler: @escaping  
fullDayActivitysRecordHandler)
```

## 4.14 佩戴状态

获取佩戴状态, 通过receiveWearState(\_ state: Int)返回结果

```
getWearState()
```

## 4.15 关机

戒指关机。

```
shutDown()
```

## 4.16 恢复出厂设置

- 恢复出厂设置

```
reset()
```

## 4.17 测试指令

- 获取戒指GitHash

```
getGitHashInfo(handler: @escaping stringHandler)
```

- 获取戒指GSensor数据

```
getGSensorInfo(handler: @escaping gsensorHandler)
```

- 获取戒指电量信息数据

```
getBatteryInfo(handler: @escaping batteryInfoHandler)
```

## 4.18 体温

- 设置体温测量状态

```
setSleepTemperatureState(open: Bool)
```

- 获取体温测量状态

```
getSleepTemperatureState(handler: @escaping boolHandler)
```

- 获取体温测量数据

```
getSleepTemperatureData(day: Int, handler: @escaping  
sleepTemperatureDataHandler)
```

## 4.19 测量状态

- 获取当前体温测量状态

```
getMeasurementState(handler: @escaping measurementStateHandler)
```

## 4.20 重启

- 重启

```
reboot()
```

- 获取重启次数(仅用于测试)

```
getRebootTimes(handler: @escaping intHandler)
```



## 4.21压力

- 启动压力单次测量

```
setStartStress()
```

- 结束单次HRV测量，测量时间过短会导致无测量结果。结果通过receiveStress(\_ stress: Int)回调(收到结果为255或者0表示压力测量中断)

```
setStopStress()
```

- 获取单次压力测量历史记录

```
getStressRecord(_ handler: @escaping stressRecordDataHandler)
```

## 4.22戒指配置项

- 获取戒指配置项

```
getRingInfo(handler: @escaping ringInfoHandler)
```

- 设置戒指配置项

```
setRingInfo(info: CRPRingInfoModel)
```

# 5 版本更新日志

1.0.4

增加体温功能

增加获取测量状态获取

增加运动状态获取

启动运动和设置运动目标方法合并

## 1.0.5

增加广播获取戒指配置项信息

增加获取和设置戒指配置项信息方法

增加压力单次测量方法

## 1.0.6

1.增加获取历史单次压力测量方法

2.修改获取历史单次hrv测量记录方法

3.修改获取全天hrv测量记录的方法