

开发指南

工程配置

开发注意事项

1 数据模型

- 1.1 CRPDiscovery
- 1.2 StepModel
- 1.3 SleepModel
- 1.4 ProfileModel
- 1.5 AlarmModel
- 1.6 HeartModel
- 1.7 SportModel
- 1.8 weather
- 1.9 forecastWeather
- 1.10 newVersionInfo
- 1.11 ScreenContent
- 1.12 watchFaceInfo
- 1.13 watchFaceSupportModel
- 1.14 Physiological
- 1.15 SitRemindModel
- 1.16 eventRemind
- 1.17 drinkWaterRemind
- 1.18 hrRemind
- 1.19 newVersionTpInfo
- 1.20 contactProfileModel
- 1.21 CRPContact
- 1.22 CRPHRVDataModel
- 1.23 CRPMedicineReminderModel
- 1.24 CRPHeartRecordModel
- 1.25 CRPStressRecordModel
- 1.26 CRPExerciseGoalsModel
- 1.27 CRPExerciseGoalStateModel
- 1.28 CRPNewSportModel
- 1.29 CRPSportRecord
- 1.30 CRPEcardProfileModel
- 1.31 CRPECard
- 1.32 CRPScreenMarketVersionModel
- 1.33 CRPScreenTagModel
- 1.34 CRPScreenModel
- 1.35 CRPScreenModellInfo
- 1.36 CRPNewWatchFaceSupportModel
- 1.37 CRPNewSportingModel
- 1.38 CRPStockSelectionModel
- 1.39 CRPCalendarModel
- 1.40 CRPCalendarConfigModel
- 1.41 CRPCalendarRemindModel

2 代理方法

- 2.1 返回当前的手表的连接状态
- 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 接收动态心率数据

- 2.6 接收血压测量结果
- 2.7 接收实时心率数据
- 2.8 接收血氧测量结果
- 2.9 接收固件的升级进度和状态
- 2.10 收到拍照请求
- 2.11 收到天气获取请求(可选)
- 2.12 收到查找手机请求(可选)
- 2.13 收到实时的体温数值(可选)
- 2.14 收到体温测量状态(可选)
- 2.15 收到一键呼叫请求(可选)
- 2.16 收到HRV实时数据(可选)
- 2.17 收到当前来电号码(可选)
- 2.18 收到吃药提醒数据(可选)
- 2.19 收到运动模式状态(可选)
- 2.20 收到压力测量结果(可选)
- 2.21 收到新版本锻炼记录(可选)
- 2.22 收到GPS数据记录(可选)
- 2.23 收到GPS实时定点数据(可选)
- 2.24 收到GPS辅助点请求(可选)
- 2.25 收到GPS EPO数据请求(可选)
- 2.26 收到GPS EPO同步进度和状态(可选)
- 2.27 收到连接确认状态,使用connectConfirm或者connectConfirmByRandomCode发起连接确认, 手表选择后返回配对状态(可选)
- 2.28 收到压力测量结果(可选)
- 2.29 收到运动中的运动数据(可选)
- 2.30 收到手表退出相机界面(可选)
- 2.31 收到股票请求更新数据(可选)
- 2.32 收到GPT请求状态或者返回数据(可选)
- 2.33 收到GPT请求预览图(可选)
- 2.34 收到GPT请求背景图(可选)

3 手表扫描及连接

- 3.1 初始化
- 3.2 手表扫描
- 3.3 取消扫描
- 3.4 手表连接
- 3.5 解除绑定

4 手表交互

- 4.0 同步时间
- 4.1 固件升级
- 4.2 查询手表电量
- 4.3 用户信息
- 4.4 天气
- 4.5 活动步数
- 4.6 睡眠
- 4.7 设置时间制式
- 4.8 翻腕亮屏
- 4.9 目标步数
- 4.10 表盘
- 4.11 闹钟
- 4.12 语言
- 4.13 消息推送
- 4.14 久坐提醒

4.15 查找手表
4.16 心率
4.17 血压
4.18 血氧
4.19 校准GSensor
4.20 拍照
4.21 关机
4.22 勿扰时段
4.23 历史步数和睡眠数据
4.24 生理周期
4.25 心电测量
4.26 体温
4.27 亮屏时间
4.28 心率预警
4.29 喝水提醒
4.30 洗手提醒
4.31 快捷通讯
4.32 恢复出厂设置
4.33 HRV疲劳度
4.34 省电模式
4.35 吃药提醒
4.36 屏幕轻触唤醒
4.37 新版本锻炼数据
4.38 压力测量
4.39 锻炼目标
4.40 电子名片
4.41 GPS功能
4.42 音频媒体开关
4.43 震动强度
4.44 睡眠检测时间设置
4.45 股票
4.46 GPT
4.47 日历
4.48 扫码配对

5 版本更新日志

2.5
2.6
2.7
2.8
2.9
3.1
3.7
3.8
3.9
3.10
3.11
3.12
3.13
3.14
3.15
3.16
3.18.1

开发指南

工程配置

- 导入framework包

导入文件夹下CRPSmartBand.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

1.2 StepModel

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 SleepModel

Modifier	Name	Explain
Int	deep	深度睡眠
Int	light	浅度睡眠
[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
String	total	时长(分钟)
String	start	开始时间
String	end	结束时间

1.4 ProfileModel

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

1.5 AlarmModel

Modifier	Name	Explain
Int	id	Id(只有0, 1, 2)
Int	enable	0: 关 1: 开
AlarmType	type	闹钟类型 单次: 每天: 自定义
Int	hour	时
Int	minute	分
Int	year	年
Int	month	月
Int	day	日
[CRPWeekDay]	weekday	例: [.sun, .thu, .mon]

1.6 HeartModel

Modifier	Name	Explain
Double	starttime	开始测量时间
Double	endTime	结束时间
[Int]	detail	心率总数据

1.7 SportModel

Modifier	Name	Explain
String	date	日期
Int	starttime	开始时间
Int	endtime	结束时间
Int	vaildTime	有效时间
SportType	type	运动类型
Int	Step	步数
Int	Distance	距离
Int	kcal	卡路里

1.8 weather

Modifier	Name	Explain
Int	type	天气类型
Int	temp	实时温度
Int	pm25	PM2.5
String	festival	公立节日
String	city	城市名

天气类型

多云	雾	阴天	雨	雪	晴	沙尘暴	霾
0	1	2	3	4	5	6	7

1.9 forecastWeather

Modifier	Name	Explain
Int	type	天气类型
Int	MaxTemp	最高温度
Int	MinTemp	最低温度

1.10 newVersionInfo

Modifier	Name	Explain
upgradeType	Type	升级类型
String	version	新版本的版本号
String	log	新版本信息
String	logEn	新版本英文信息
Int	mcu	芯片类型 mcu = 4或8时为汉天下,mcu=7时为RealTek

1.11 ScreenContent

Modifier	Name	Explain
ContentPosition	position	屏幕信息的位置 0:上 1: 下
Content	upperContent	时间上方内容
Content	underContent	时间下方内容
UIColor	contentColor	字体颜色
String	MD	图片的MD5值

1.12 watchFaceInfo

Modifier	Name	Explain
Int	id	表盘的序号
Int	model	该表盘所属类型的序号
String	fileUrl	表盘的下载路径
String	imageUrl	表盘d预览图的下载路径

1.13 watchFaceSupportModel

Modifier	Name	Explain
[Int]	supportModel	支持类型的数据
Int	currentId	当前手表的下载表盘市场的序号(0或nil为当前手表没有下载过表盘市场的表盘)

1.14 Physiological

Modifier	Name	Explain
[reminderModel]	reminderModels	提醒模式
Int	cycleTime	生理周期的天数
Int	menstruationTime	经期的天数
Int	lastTimeMonth	上次经期的月份
Int	lastTimrDay	上次经期日
Int	remindTimeHour	小时
Int	remindTimeMintue	分钟

1.15 SitRemindModel

Modifier	Name	Explain
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)

1.16 eventRemind

Modifier	Name	Explain
Bool	isRemind	开关 (是否提醒)
Int	startHour	开始时间(24小时制)
Int	startMinute	开始时间(分钟)
Int	remindCount	提醒次数
Int	remindTimeInterval	提醒时间间隔 (分钟)

1.17 drinkWaterRemind

Modifier	Name	Explain
Bool	isRemind	开关 (是否提醒)
Int	startHour	开始时间(24小时制)
Int	startMinute	开始时间(分钟)
Int	remindCount	提醒次数
Int	remindTimeInterval	提醒时间间隔 (分钟)
Int	waterIntake	已经喝水的杯数

1.18 hrRemind

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

1.19 newVersionTpInfo

Modifier	Name	Explain
Stirng	tp_path	升级文件地址
String	tp_md5	升级文件MD5
String	tp_offset	文件升级地址

1.20 contactProfileModel

Modifier	Name	Explain
Int	contactMax	允许同步的联系人数量的最大值
Int	contactAvatarWidth	联系人头像的宽
Int	contactAvatarHeight	联系人头像的高

1.21 CRPContact

Modifier	Name	Explain
Int	contactID	序号
String	fullName	名称
UIImage	image	头像
String	selectPhoneNumber	电话号码

1.22 CRPHRVDataModel

Modifier	Name	Explain
Double	time	测量时间
Int	sportLevel	运动激烈等级
[Int]	detail	RRI数据

1.23 CRPMedicineReminderModel

Modifier	Name	Explain
Int	medincineID	序号
Int	pastDay	当前已经吃药的天数
String	medincineName	药品名名称
Int	cycleTime	吃药周期
[CRPMedicineTimeModel]	reminderTimes	提醒时间和吃药数量

1.24 CRPHeartRecordModel

Modifier	Name	Explain
Int	remindTimeHour	小时
Int	remindTimeMinute	分钟
Int	pillCount	吃药数量

1.25 CRPStressRecordModel

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

1.26 CRPExerciseGoalsModel

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)

1.27 CRPExerciseGoalStateModel

Modifier	Name	Explain
Int	state	开关状态
[Int]	weekday	锻炼日数组 (参考CRPWeekday)

1.28 CRPNewSportModel

Modifier	Name	Explain
Int	id	序号
Int	starttime	开始时间
Int	endtime	结束时间
Int	vaildTIme	有效时间
SportType	type	运动类型
Int	Step	步数
Int	Distance	距离
Int	kcal	卡路里
[Int]	heartRate	心率数据

1.29 CRPSportRecord

Modifier	Name	Explain
Int	id	序号
Int	startTime	开始时间
Int	type	锻炼类型

1.30 CRPEcardProfileModel

Modifier	Name	Explain
Int	eCardMax	ECard的最大数量
Int	eCardURLLimit	ECard URL的最大长度
[Int]	ids	当前id的排序数组

1.31 CRPECard

Modifier	Name	Explain
Int	id	序号
String	title	标题
String	url	链接

1.32 CRPScreenMarketVersionModel

Modifier	Name	Explain
Int	apiVersion	api接口版本
Int	funcVersion	功能版本

1.33 CRPScreenTagModel

Modifier	Name	Explain
Int	tag_name	标签名称
Int	id	标签序号
[CRPScreenModel?]	faces	表盘

1.34 CRPScreenModel

Modifier	Name	Explain
String?	uploader	设计师名称
String?	name	表盘名称
Int	id	表盘序号
Int	download	下载量
String?	file	文件下载链接
Int	size	表盘大小
String?	preview	预览图链接
String?	remark_lang	语言
String?	remark_cn	中文描述
String?	remark_en	英文描述

1.35 CRPScreenModelInfo

Modifier	Name	Explain
String?	uploader	设计师名称
String?	name	表盘名称
Int	id	表盘序号
Int	download	下载量
String?	file	文件下载链接
Int	size	表盘大小
String?	preview	预览图链接
String?	remark_lang	语言
String?	remark_cn	中文描述
String?	remark_en	英文描述
[CRPScreenModel]	face_list	相似的表盘列表

1.36 CRPNewWatchFaceSupportModel

Modifier	Name	Explain
[Int]	supportModel	支持类型的数据
Int	currentId	当前手表的下载表盘市场的序号(0或nil为当前手表没有下载过表盘市场的表盘)
Int	udtMaxSize	当前手表支持的表盘市场的大小

1.37 CRPNewSportingModel

Modifier	Name	Explain
Int	startTime	开始时间
CRPSportState	type	运动状态
Int	Step	步数
Int	sportTime	运动时间
Int	Distance	距离
Int	kcal	卡路里
Int	heartRate	实时心率数据

1.38 CRPStockSelectionModel

Modifier	Name	Explain
Int	id	序号
Double	regularMarketOpen	开盘价
Double	regularMarketDayHigh	最高价
Double	regularMarketDayLow	最低价
Int	regularMarketVolume	成交量
Double	peRatio	市盈率
Int	marketCap	市值
Double	fiftyTwoWeekHigh	52周最高价
Double	fiftyTwoWeekLow	52周最低价
Int	averageDailyVolume3Month	平均成交量
Double	regularMarketPrice	收盘价
String	currency	币种
String	shortName	简称
String	symbol	关键字
String	exchange	交易所简称
Double	regularMarketPreviousClose	昨日收盘价
Bool	isOpen	是否开盘

1.39 CRPCalendarModel

Modifier	Name	Explain
Int	id	序号
String	title	标题
Date	startTime	开始时间
Date	endTime	结束时间

1.40 CRPCalendarConfigModel

Modifier	Name	Explain
Int	limit	支持日历同步个数
[CRPCalendarRecordModel]	records	当前同步的日历记录

1.41 CRPCalendarRemindModel

Modifier	Name	Explain
Bool	isRemind	是否提醒
Int	remindTime	提前多少分钟提醒

2 代理方法

2.1 返回当前的手表的连接状态

```
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

```
receiveSteps(_ model: StepModel)
```

2.4接收心率测量结果(单次测量心率)

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

2.5 接收动态心率数据

```
receiveHeartRateAll(_ model: HeartModel)
```

2.6 接收血压测量结果

```
receiveBloodPressure(_ heartRate: Int, _ sbp: Int, _ dbp: Int)
```

2.7 接收实时心率数据

一般只用于显示不用于存储，部分手表支持返回实时RRI数据

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

2.8 接收血氧测量结果

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

2.9 接收固件的升级进度和状态

```
receiveUpgrde(_ state: UpgradeState, _ progress: Int)
```

2.10 收到拍照请求

```
recevieTakePhoto()
```

2.11 收到天气获取请求(可选)

手表可以保存实时天气2个小时，2小时以后会清空天气信息。当手表无今日天气信息时，手表切换到天气界面时会通过receiveWeather()请求获取天气信息

```
recevieWeather()
```

2.12 收到查找手机请求(可选)

```
recevieFindPhone()
```

2.13 收到实时的体温数值(可选)

手表收到实时体温数值后，保留最后一次收到的值，当收到测量结束状态时，最后的值就是测量结果

```
receiveRealTimeTemperature(_ temp: Double)
```

2.14 收到体温测量状态(可选)

用于判断单次体温测量的状态 收到state为1时为正在测量，收到state为0时为测量结束

```
receiveTemperature(_ state: Int)
```

2.15 收到一键呼叫请求(可选)

```
receiveCalling()
```

2.16 收到HRV实时数据(可选)

```
receviceHRVRealTime(_ model: CRPHRVDataModel)
```

2.17 收到当前来电号码(可选)

```
receivePhoneNumber(number: String)
```

2.18 收到吃药提醒数据(可选)

```
receiveMedicineInfo(_ max: Int, _ model: CRPMedicineReminderModel?)
```

2.19 收到运动模式状态(可选)

err为1时，表示启动失败，充电中或者低电时state回复end的状态；手表已经正在锻炼，返回正在锻炼的类型

```
receiveSportState(_ state: SportType, _ err: Int)
```

2.20 收到压力测量结果(可选)

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

2.21 收到新版本锻炼记录(可选)

锻炼结束时或调用getSportRecordList方法时回调数据

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

2.22 收到GPS数据记录(可选)

调用getGPSDataRecordList方法时回调数据，获取到的是记录的开始时间列表，可通过getGPSRecordData方法获取相应记录的详细的数据

```
receiveGPSDataRecordList(time: [Int])
```

2.23 收到GPS实时定点数据(可选)

```
receiveGPSRealTime(location: CLLocationCoordinate2D)
```

2.24 收到GPS辅助点请求(可选)

收到GPS辅助点请求，需调用syncGPSAuxiliary方法发送

```
receiveGPSAuxiliaryRequest()
```

2.25 收到GPS EPO数据请求(可选)

收到GPS EPO数据请求，需调用startSyncEPO方法传输相应的EPO数据

```
receiveEPORequest(type: Int)
```

2.26 收到GPS EPO同步进度和状态(可选)

```
receiveEPOSync(_ state: CRPUgradeState, _ progress: Int)
```

2.27 收到连接确认状态,使用connectConfirm或者connectConfirmByRandomCode发起连接确认，手表选择后返回配对状态(可选)

```
receiveConnectConfirm(_ state: CRPPairState)
```

2.28 收到压力测量结果(可选)

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

2.29 收到运动中的运动数据(可选)

```
receiveRealTimeSportData(data: CRPNewSportingModel)
```

2.30 收到手表退出相机界面(可选)

部分手表支持退出相机界面通知

```
receiveExitCameraView()
```

2.31 收到股票请求更新数据(可选)

```
receiveStockRequestUpdateInfo()
```

2.32 收到GPT请求状态或者返回数据(可选)

```
receiveGPTState(type: CRPGPTType, state: CRPGPRequestState, result: NSData)
```

2.33 收到GPT请求预览图(可选)

```
receiveRequestGPTPreviewImage()
```

2.34 收到GPT请求背景图(可选)

```
receiveRequestGPTImage()
```

3 手表扫描及连接

3.1 初始化

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

```
CRPSmartBandSDK.sharedInstance.delegate = self
```

3.2 手表扫描

在权限允许和蓝牙开启状态下，才能开始正常扫描。在扫描过程中发现手表，通过scan()方法，通过progressHandler回调扫描中发现的手表，completionHandler回调整个扫描中扫描到的所有手表，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?);
```

- 获取系统蓝牙已连接设备，获取到的是原始的CBPeripheral的列表，可使用CBPeripheral初始化CRPDiscovery后进行连接。

```
getConnectedBand(_ handler: connectedPeripheralHandler)
```

3.3 取消扫描

取消手表扫描。

```
interrupScan()
```

3.4 手表连接

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手表连接

```
connet(_ discovery: CRPDiscovery)
```

3.5 解除绑定

- 解除手表绑定

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

- 重新连接

```
reConnect()
```

- 双模蓝牙解除手表连接通知

此方法需要在解除手表绑定前发起，用于解决之后连接时蓝牙配对弹窗有可能显示慢的问题

```
removeBond()
```

4 手表交互

4.0 同步时间

保持手表与手机端时间一致

```
setTime()
```

4.1 固件升级

- 查询固件版本

查询手表当前固件版本。

```
getSoftver(_ 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)
```

- 查询手表的DFU状态(DFU不为0时既手表处于升级状态)

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

- Nordic固件升级(汉天下和Nordic的固件升级方式不同，需要根据查询到的muc不同使用相应的升级方式，mcu为4时使用汉天下升级方式升级)

Nordic在收到新版本固件回调后，可调用此接口对手表进行升级

```
startUpgreded(info: newVersionInfo?)
```

- Nordic停止固件升级

```
stopUpgrade()
```

- 汉天下获取OTA模式时的Mac地址(汉天下固件升级需先获取固件升级时的Mac地址)

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

- 汉天下固件升级

在收到新版本固件回调后，当mcu为4或8时使用此接口对手表进行升级

```
startHTXOTA(info: newVersionInfo, upgradeMac: String, isUser: Bool, _ isPatch:Bool = true)
```

- 汉天下停止固件升级

```
stopHTXOTA()
```

- RealTek固件升级

在收到新版本固件回调后，当mcu为7时使用此接口对手表进行升级

```
startRealTekUpgrade(info: newVersionInfo?)
```

- RealTek TP升级

在收到新版本Tp回调后，使用此接口对手表进行升级

```
startTpUpgrade(tpInfo: newVersionTpInfo?)
```

- Goodix固件升级

mcu为10时，手表为Goodix，使用checkLatest方法获取到最新版本后，下载fileUrl的文件，然后把文件地址传入此方法中

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

4.2 查询手表电量

查询手表当前电量

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

4.3 用户信息

- 设置用户信息

```
setProfile(_ profile: ProfileModel)
```

- 设置步长

手表固件1.6.6及以上可向手表设置步长，更加精准计算活动数据

```
setStepLength(length: Int)
```

4.4 天气

- 设置今日天气

向手表推送今日天气信息。

```
setWeather(_ weather: weather)
```

- 设置未来7日天气

向手表推送未来7天天气信息。

```
setForecastWeather(_ weathers:[forecastWeather])
```

- 设置当前城市名

部分手表支持设置当前城市名

```
sendCityName(_ cityName: String)
```

- 新版本天气设置(支持设置日出日落时间)

```
setNewVerWeather(weather: CRPNewWeather)
```

4.5 活动步数

- 同步今日活动步数

```
getSteps(_ handler:@escaping stepHandler)
```

- 获取步数统计数据

部分手表支持最近两天步数半小时分类统计

获取今天步数统计

```
get24HourSteps(_ handler: @escaping intsHandler)
```

获取昨天步数统计

```
getAgo24HourSteps(_ handler: @escaping intsHandler)
```

- 获取卡路里统计数据

部分手表支持最近两天卡路里半小时分类统计(数据单位: 小卡)

获取今天卡路里统计

```
get24HourCals(_ handler: @escaping intsHandler)
```

获取昨天卡路里统计

```
getAgo24HourCals(_ handler: @escaping intsHandler)
```

- 获取距离统计数据

部分手表支持最近两天距离半小时分类统计(数据单位: 厘米)

获取今天距离统计

```
get24HourDistances(_ handler: @escaping intsHandler)
```

获取昨天距离统计

```
getAgo24HourDistances(_ handler: @escaping intsHandler)
```

- 部分手手表支持获取多天历史活动步数(day:0-6; 0为今天, 1为昨天, 以此类推)

```
getHistoryStepData(day: Int, handler: @escaping historyStpeHandler)
```

4.6 睡眠

获取今日睡眠数据

```
getSleepData(_ handler: @escaping sleepHandler)
```

- 部分手手表支持获取多天历史睡眠数据(day:0-6; 0为今天, 1为昨天, 以此类推)

```
getHistorySleepData(day: Int, handler: @escaping historySleepHandler)
```

4.7 设置时间制式

手表时间制式支持12小时制和24小时制。

- 设置时间制

```
setTimeFormat(_ type:Int)
```

- 查询时间制式

查询手表正在使用的时间制式。

```
getTimeformat(_ handler: @escaping IntHandler)
```

type: 0: 12-timeformat 1: 24-timeformat

4.8 翻腕亮屏

- 设置翻腕亮屏状态

开启或者关闭手表翻腕亮屏功能。

```
setQuickView(_ isOn: Bool)
```

- 查询手表翻腕亮屏开启状态.

```
getQuickView(_ handler: @escaping IntHandler)
```

0: 关闭 1: 开启

- 设置翻腕亮屏时间

```
setQuickViewTime(_ periodTime: periodTimeModel)
```

- 获取翻腕亮屏时间

```
getQuickViewTime(_ handler: @escaping periodTimeHandler)
```

4.9 目标步数

- 设置目标步数

向手表推送用户设置的目标步数，当天活动步数达到目标步数时，手表会有目标达成提示。

```
setGoal(_ goal: Int)
```

- 查询目标步数

查询手表被设置的目标步数

```
getGoal(_ handler: @escaping IntHandler)
```

4.10 表盘

- 切换手表表盘

手表支持三个不同的表盘，可以自由切换。

```
setDial(_ type:Int)
```

type:

第一个表盘	第二个表盘	第三个表盘
0	1	2

- 查询手表表盘

查询手表正在使用的表盘序号

```
getDial(_ handler: @escaping IntHandler)
```

- 更换自定义表盘背景图片

部分1.3寸寸彩屏的表盘支持更换背景图片片，图片片尺寸寸为240 * 240 px,升级过程会通过receiveUpgradeScreen回调进度和状态

imageSize和compressionType需要通过getScreenContent方法获取

```
startChangeScreen(_ image: UIImage, _ imageSize :ScreenImageSize, _ isCricle: Bool = false, _ compressionType: Int)
```

- 修改自定义表盘布局

```
setupScreenContent(content: ScreenContent)
```

content的MD5值需设置为背景图的MD5值，当MD5值为""时,背景图片恢复默认背景

- 获取指定表盘布局

```
getScreenContent(_ handler: @escaping screenContentHandler)
```

- 获取支持摆盘类型

部分手表新增了一个表盘为，查询支持的表盘类型以及当前的表表盘在表盘市场的ID

```
getWatchFaceSupportModel(_ handler: @escaping watchFaceSupportHandler)
```

- 获取表盘市场

根据表盘的支持类型获取能可以更换的表盘列表(分页获取)

currentPage: 当前第几页(从1开始); perPage: 每页返回多少个表盘(建议设置为18)

```
getWatchFaceInfo(_ supportModel: [Int], currentPage: Int, perPage: Int ,_ handler:  
@escaping watchFaceHandler)
```

- 发送表盘文件

向手表发送表盘文件，期间手表会重启，升级过程会通过receiveUpgradeScreen回调进度和状态
watchFaceID：杰里平台的手表需要发送表盘文件的ID才能完成表盘传输

```
startChangeWatchFace(_ Info: watchFaceInfo, _ watchFaceID: Int = 0)
```

- 获取表盘市场空间大小

部分手表对表盘大小有限制，获取表盘需传入大小限制获取的表盘大小

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

- 杰里表盘市场功能版本

杰里平台的手表表盘需要根据功能版本去区分支持的表盘市场表盘

```
getJLScreenMarketVersionInfo(_ handler: @escaping screenMarketVersionInfoHandler)
```

- 获取新版本版本表盘市场标签列表

supportModel:固件支持的表盘模版(getWatchFaceSupportModel或者getNewWatchFaceSupportModel方法获取);

ver:手表固件版本号;

currentPage: 当前第几页(从1开始);

perPage: 每页返回多少个表盘(建议设置为18);

memorySize: 使用getUDTMaxSize或getNewWatchFaceSupportModel获取，如无回调，传0即可;

platform:蓝牙扫描时可获取;

model:杰里平台需要使用getJLScreenMarketVersionInfo获取，其它平台apiVersion和funcVersion设置为0即可;

```
getNewWatchFaceTagList(supportModel: [Int], ver: String, currentPage: Int, perPage: Int,  
memorySize: Int,platform: CRPPlatform, model: CRPScreenMarketVersionModel, _ handler:  
@escaping watchFaceTagListHandler)
```

- 获取新版本版本表盘市场单标签下表盘列表

tag_id:标签id;

ver:手表固件版本号;

supportModel:固件支持的表盘模版(getWatchFaceSupportModel或者getNewWatchFaceSupportModel方法获取);

currentPage: 当前第几页(从1开始);

perPage: 每页返回多少个表盘(建议设置为18);

memorySize: 使用getUDTMaxSize或getNewWatchFaceSupportModel获取，如无回调，传0即可;

platform:蓝牙扫描时可获取;

model:杰里平台需要使用getJLScreenMarketVersionInfo获取，其它平台apiVersion和funcVersion设置为0即可;

```
getNewWatchFaceList(tag_id: Int, ver: String, supportModel: [Int], currentPage: Int, perPage: Int, memorySize: Int, platform: CRPPlatform, model: CRPScreenMarketVersionModel, _ handler: @escaping watchFaceListHandler)
```

- 获取新版本版本表盘市场表盘详情

screenId:需要查询的表盘id;

ver:手表固件版本号;

supportModel:固件支持的表盘模版(getWatchFaceSupportModel方法获取);

memorySize: 使用getUDTMaxSize或getNewWatchFaceSupportModel获取, 如无回调, 传0即可;

platform:蓝牙扫描时可获取;

model:杰里平台需要使用getJLScreenMarketVersionInfo获取, 其它平台apiVersion和funcVersion设置为0即可;

```
getNewWatchFaceInfo(screenId: Int, ver: String, supportModel: [Int], memorySize: Int, platform: CRPPlatform, model: CRPScreenMarketVersionModel, _ handler: @escaping watchFaceInfoHandler)
```

4.11 闹钟

- 设置手表闹钟

手表可支持三个闹钟, 可根据闹钟序号分别设置闹钟信息。单次闹钟支持设置日期, 其它重复类型的闹钟不支持。

```
setAlarm(_ alarm: AlarmModel)
```

- 查询所有闹钟

查询手表保存的所有闹钟信息

```
getAlarms(_ handler: @escaping alarmHandler)
```

- 新版本闹钟

获取所有闹钟信息

```
getNewVerAlarm(_ handler: @escaping alarmHandler)
```

设置闹钟

```
setNewVerAlarm(_ alarm: AlarmModel)
```

新增闹钟

```
setNewVerAddAlarm(_ alarm: AlarmModel)
```

删除闹钟

```
setNewVerDeleteAlarm(_ id: Int)
```

删除所有闹钟

```
setNewVerDeleteAllAlarm()
```

4.12 语言

- 设置手表语言

手表支持多国语言。

```
setLanguage(_ lang: Int)
```

lang:

英 语	中 文 简 体	日 语	韩 语	德 语	法 语	西 班 牙 语	阿 拉 伯 语	俄 语	中 文 繁 体	乌 克 兰 语	意 大 利 语	葡 萄 牙 语	荷 兰 语	波 兰 语	瑞 典 语	芬 兰 语	丹 麦 语	挪 威 语	匈 牙 利 语	捷 克 语	保 加 利 亚	罗 马 尼 亚	斯 洛 伐 克 语	拉 脱 维 亚 语	印 度 尼 西 亚 语	泰 语	土 耳 其 语	越 南 语	印 地 语
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

注意：请确认手表支持的语言，设置手表不支持的语言将无效

- 获取当前手表语言

```
getLanguage(_ handler: @escaping IntHandler)
```

- 设置手表当前语言版本

手表语言版本分中文版和国际版，中文版只显示中文，国际版才能切换外语

```
setLanguageVersion(_ isChinese: Bool)
```

4.13 消息推送

- 设置其它消息推送状态

开启或者关闭其它消息推送。

```
setNotification(_ switchs:[NotificationType])
```

- 查询其它消息推送状态

查询其它消息推送状态

```
getNotifications(_ handler: @escaping NotificationsHandler)
```

- 新版本的消息推送设置

查询手表支持的消息推送类型，根据返回的列表对照NotificationType知道手表支持的消息推送类型

```
getNotificationSupportType(_ handler: @escaping notificationsHandler)
```

查询新版本消息推送类型状态

```
getNotificationState(_ handler: @escaping notificationsHandler)
```

设置新版本消息推送类型状态

```
setNotificationState(_ states: [Int])
```

4.14 久坐提醒

- 设置久坐提醒状态

开启或者关闭久坐提醒。

```
setRemindersToMove(_ isOn: Bool)
```

- 查询久坐提醒状态

查询久坐提醒状态

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

- 设置久坐提醒时间

设置久坐提醒有效时间段

```
setSitRemind(_ sitRemind: SitRemindModel )
```

- 获取久坐提醒时间

获取久坐提醒时间段

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

4.15 查找手表

查找手表，手表收到此指令以后会震动几秒。

```
setFindDevice()
```

4.16 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

开始动态心率测量

```
setStartHeart()
```

- 停止动态心率测量

停止动态心率测量。测量结果通过receiveHeartRateAll(_ model: HeartModel)回调。

```
setStopHeart()
```

- 查询上次动态心率测量结果

在未连接状态下测量动态心率，手表可以保存最后一次的测量结果。结果通过receiveHeartRateAll回调。

```
getHeartData()
```

- 开启定时测量心率

手表支持24小时定时测量心率，从0点0分开始测量量，可以设置测量量时间间隔（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）

```
set24HourHeartRate(_ interval:Int)
```

- 获取24小时定时测量状态

```
get24HourHeartRateInterval(_ handler: @escaping IntHandler)
```

- 查询今天定时测量心率数据

```
get24HourHeartRate(_ handler: @escaping IntsHandler)
```

- 查询昨天定时测量心率数据

```
getAgo24HourHeartRate(_ handler: @escaping IntsHandler)
```

- 获取运动数据

部分手表支持多种运动模式的心率测量量，测量量结果包括心率和卡路里等其它运动相关数据，此接口用于获取卡路里等数据。保存最近三次的运动数据。

```
getSportData(_ handler: @escaping SportHandler)
```

- 开启静态心率单次测量

```
setStartSingleHR()
```

- 结束静态心率单次测量

结束单次测量量。测量量时间过短，会导致无无测量量数据。测量结果通过receiveHeartRate(_ heartRate: Int)回调。

```
setStopSingleHR()
```

- 获取单次心率测量历史记录(部分手表支持)

```
getHeartRecordData()
```

- 部分手表支支持获取多天历史定时测量心率数据(day:0-6; 0为今天, 1为昨天 , 以此类推)

```
getHistoryHrData(day: Int, handler: @escaping historyHrHandler)
```

4.17 血压

- 开始测量血压

开始测量血压。

```
setStartBlood()
```

- 停止测量血压

停止测量血压，测量时间过短会导致无测量结果。测量结果通过receiveBloodPressure(_ heartRate: Int, _ sbp: Int, _ dbp: Int)回调。

```
setStopBlood()
```

部分手表支持全天血压功能

- 设置全天血压

```
setFullDayBPStatus(open: Bool)
```

- 获取全天血压状态

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

- 获取全天血压数据

```
getFullDayBPData(_ handler: @escaping intsHandler)
```

- 获取单次血压测量历史记录(部分手表支持)

```
getBPRRecordData()
```

4.18 血氧

- 开始测量血氧

开始测量血氧。

```
setStartSpO2()
```

- 停止测量血氧

停止测量血氧，测量时间过短会导致无测量结果。结果通过receiveSpO2(_ o2: Int)回调。

```
setStopSpO2()
```

部分手表支持全天血氧功能

- 设置全天血氧

```
setFullDayO2Status(open: Bool)
```

- 获取全天血氧状态

```
getFullDayO2Status(_ handler: @escaping intHandler)
```

- 获取全天血氧数据

```
getFullDayO2Data(_ handler: @escaping intsHandler)
```

- 获取单次血氧测量历史记录(部分手表支持)

```
getO2RecordData()
```

4.19 校准GSensor

- 校准GSensor

手表出现翻腕亮屏不灵敏或者计步不准，可校准GSensor以修复，校准过程中，手表水平置于桌面。

```
sendCalibration()
```

4.20 拍照

- 切换至拍照界面

切换手表至拍照界面。

```
switchCameraView()
```

- 设置拍照监听

长按手表拍照界面，可以触发手表的拍照指令，通过receiveTakePhoto()回调。

4.21 关机

手表关机。

```
shutdown()
```

4.22 勿扰时段

- 设置勿扰时段

手表支持勿扰时段，勿扰时段内不显示消息推送。

```
setDisturbTime(_ periodTime: periodTimeModel)
```

- 查询勿扰时段

查询手表设置的勿扰时段。

```
getDisturbTime(_ handler: @escaping periodTimeHandler)
```

4.23 历史步数和睡眠数据

手表可保存前两天的计步数据和睡眠数据(allDataHandler中包含stepModels和sleepModels,数组的第一个为昨天数据，第二个为前天数据)

- 获取历史数据

```
getAllData(_ handler: @escaping allDataHandler)
```

4.24 生理周期

部分手表支持设置生理周期提醒

- 设置生理周期提醒

```
setPhysiological(physiological: Physiological)
```

- 查询生理周期提醒

```
getPhysiological(_ handler: @escaping physiologicalHandler)
```

4.25 心电测量

部分手表支持心电测量

- 开始心电测量

```
startECGMeasure()
```

- 停止心电测量

```
stopECGMeasure()
```

- 检测是否是新的心电测量方式

新的测量量方方式，手手表可以保存最后一次未发送的测量量结果;老版本则不能。

```
isNewECGMeasurementVersion()
```

- 查询上次心心电数据

查询手手表保存的心心电数据

```
getLastMeasureECGData()
```

- 设置心心电测量量期间心心率

用测量量所得到的数据，通过心心电算法库计算出瞬时心率，发送至手手表。

```
sendECGHeartRate(heartRate: Int)
```

4.26 体温

部分手表支持体温测量

- 开始单次体温测量

receiveRealTimeTemperature回调方法返回测量值，receiveTemperature回调方法返回测量状态

```
sendSingleTemperatureStart()
```

- 结束单次体温测量

```
sendSingleTemperatureEnd()
```

- 获取自动体温测量开关

1: 开启；0: 关闭

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

- 设置自动体温测量开关

```
sendAutoTemperature(_ isOpen: Bool)
```

- 获取自动体温测量数据
每半小时一个数据

```
getAutoTemperatureData(_ todayHandler: @escaping doublesHandler, _ yesterdayHandler:  
@escaping doublesHandler)
```

- 获取体温单位
1:华氏度； 0:摄氏度

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

- 设置体温单位
1:华氏度； 0:摄氏度

```
setTemperatureUnit(_ unit:Int)
```

部分手表支持全天新版本全天体温功能(一分钟测一次)

- 设置全天体温

```
setFullDayTemperatureStatus(open: Bool)
```

- 获取全天体温状态

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

- 获取全天体温数据

```
getFullDayTemperatureData(_ handler: @escaping intsHandler)
```

4.27 亮屏时间

部分手表支持亮屏时间选择

- 设置亮屏时间

```
sendAutoLockTime(_ time: Int)
```

- 获取亮屏时间

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

4.28 心率预警

部分手表支持心率预警

- 设置心率预警值

```
sendHeartRateRemind(_ remind: hrRemind)
```

- 获取心率预警值

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

4.29 喝水提醒

部分手表支持喝水提醒

- 设置喝水提醒数据

```
sendDrinkWaterRemind(_ remind: drinkWaterRemind)
```

- 获取喝水提醒数据

```
getDrinkWaterRemind(_ handler: @escaping drinkWaterHandler)
```

4.30 洗手提醒

部分手表支持洗手提醒

- 设置洗手提醒数据

```
sendHandWashRemind(_ remind: eventRemind)
```

- 获取洗手提醒数据

```
getHandWashRemind(_ handler: @escaping reminderHandler)
```

4.31 快捷通讯

部分手表支持同步联系人功能

- 获取联系人配置项

```
getContactProfile(_ handler: @escaping contactProfileHandler)
```

- 设置联系人

```
setContact(profile: contactProfileModel, contacts: [CRPContact])
```

- 清空联系人

```
cleanAllContact()
```

- 删除指定联系人

```
deleteContact(contactID: Int)
```

- 获取当前联系人个数

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

- 获取当前联系人支持版本(0:联系人号码不支持任何符号, 1:联系人号码支持传输"+","*","#"这三个符号。如无回复则表示联系人号码不支持任何符号)

```
getContactSupportVer()
```

4.32 恢复出厂设置

部分手表支持恢复出厂设置

- 恢复出厂设置

```
reset()
```

4.33 HRV疲劳度

部分手表支持HRV

- 设置测量间隔

```
setHRV(_ interval:Int)
```

- 获取测量间隔

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

- 获取历史数据中某天的疲劳度数据量(dayIndex: 0~6,可获取7天的数据, 0:今天, 1:昨天, 以此类推)

```
getHRVCount(_ dayIndex: Int, _ handler: @escaping intHandler)
```

- 获取历史数据中某天指定条的的疲劳度数据

```
getHRVData(_ dayIndex: Int, _ index: Int, _ handler: @escaping hrvDataHandler)
```

- 主动测量获取RRI数据

```
startHRVMeasure()
```

- 停止测量RRI数据

```
stopHRVMeasure()
```

4.34 省电模式

部分手表支持省电模式

- 设置省电模式

```
setPowerSaveState(open: Bool)
```

- 获取省电模式状态

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

4.35 吃药提醒

部分手表支持吃药提醒

- 获取当前吃药提醒信息,通过receiveMedicineInfo方法返回数据

```
getMedicineInfo()
```

- 设置吃药提醒

```
setMedicine(medicine: CRPMedicineReminderModel)
```

- 删除指定id的吃药提醒

```
deleteMedicine(id: Int)
```

- 删除所有吃药提醒设置

```
deleteAllMedicine()
```

4.36 屏幕轻触唤醒

部分手表支持设置屏幕轻触唤醒

- 设置屏幕轻触唤醒

```
setTapToWake(open: Bool)
```

- 获取屏幕轻触唤醒设置(0: 关闭, 1: 开启)

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

4.37 新版本锻炼数据

部分手表支持开启运动模式

- 设置运动模式

```
setSportState(state: SportType)
```

部分手表支持新版本的锻炼数据

- 获取锻炼列表

通过receiveSportList回调方法返回数据

```
getSportRecordList()
```

- 获取锻炼详细数据

根据receiveSportList返回的列表，判断startTime>0时调用以下方法获取相应id的详细数据

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

4.38 压力测量

部分手表支持压力测量

- 查询是否支持压力测量

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

- 开始压力测量，测量结果通过receiveStress(_ stress: Int)回调返回

```
setStartStress()
```

- 结束压力测量

```
setStopStress()
```

- 获取压力测量记录

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

- 获取全天压力测量开关(0:关闭, 1: 开启)

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

- 设置全天压力测量开关(0:关闭, 1: 开启)

```
setAutoStress(_ interval:Int)
```

- 获取今天的全天压力数据

```
getTodayAutoStressData(_ handler: @escaping intsHandler)
```

- 获取昨天的全天压力数据

```
getYesterDayAutoStressData(_ handler: @escaping intsHandler)
```

4.39 锻炼目标

部分手表支持锻炼目标

- 获取平常日锻炼目标

```
getNormalExerciseGoal()
```

- 设置平常日锻炼目标

```
setNormalExerciseGoal(_ goals: CRPExerciseGoalsModel)
```

- 获取锻炼日目标

```
getExerciseDayGoal()
```

- 设置锻炼日目标

```
setExerciseDayGoal(_ goals: CRPExerciseGoalsModel)
```

- 获取锻炼日设置状态

```
getExerciseState()
```

- 设置锻炼日设置状态

```
setExerciseState(_ state: CRPExerciseGoalStateManager)
```

4.40 电子名片

部分手表支持电子名片

- 获取电子名片配置信息

```
getECardProfile(handler: @escaping eCardProfileHandler)
```

- 设置电子名片

根据获取的电子名片配置信息设置电子名片

```
setECard(profile: CRPEcardProfileModel, eCard: CRPECard)
```

- 获取电子名片

```
getECardInfo(id: Int, _ handler: @escaping eCardInfoHandler)
```

- 删除电子名片

```
deleteECard(id: Int)
```

- 电子名片排序

```
setECardSequenceArray(ids:[Int])
```

4.41 GPS功能

部分手表支持GPS功能

- 获取GPS数据记录列表，通过receiveGPSDataRecordList(time: [Int])方法返回记录的开始时间列表

```
getGPSDataRecordList()
```

- 获取GPS记录的详细数据

使用receiveGPSDataRecordList(time: [Int])返回的开始时间去获取对应，每两秒定位一个点

```
getGPSRecordData(time: Int, _ handler: @escaping gpsDataHandler)
```

- 启动同步EPO数据

```
startSyncEPO(path: String, type: Int)
```

- 同步GPS辅助点

手表开启GPS锻炼时会通过receiveGPSAuxiliaryRequest()向App请求辅助定位点

```
syncGPSAuxiliary(local: CLLocationCoordinate2D)
```

4.42 音频媒体开关

部分手表音频媒体开关(0: 未连接通话蓝牙，1: A2DP连接，2: A2DP断开，0xFF: 不支持A2DP控制)

- 设置音频媒体状态

```
setA2DPControl(state: Int)
```

- 获取音频媒体状态

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

- 获取音频蓝牙名称

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

4.43 震动强度

部分手表支持设置震动强度(1: 弱, 2: 中等, 3: 强)

- 设置震动强度

```
setVibrationStrength(type: Int)
```

- 获取震动强度

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

4.44 睡眠检测时间设置

部分手表支持睡眠检测时间设置（限制时间段：开始睡眠时间20~3，结束6~12点）

- 设置睡眠检测时间

```
setSleepTime(info: CRPSleepDetectionInfo)
```

- 获取睡眠检测时间

```
getSleepTime(handler: @escaping sleepDetectionInfoHandler)
```

4.45 股票

部分手表支持股票功能

- 获取股票支持数量

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

- 设置股票为无信息

```
setNullStockData()
```

- 设置或更新股票信息

```
setStockData(data: CRPStockSelectionModel)
```

- 删除指定id股票

```
deleteStock(id: Int)
```

- 设置显示股票排序

```
setStockSequence(ids: [Int])
```

4.46 GPT

部分手表支持GPT功能

- 获取GPT支持类型

```
getGPTSupportType(_ handler: @escaping gptSupportTypeHandler)
```

- 设置GPT状态

```
setGPTState(type: CRPGPTType, state: CRPGPTReplyState)
```

- 通过解析receiveGPTState的结果音频数据后设置GPT问题

```
setGPTQuestion(type: CRPGPTType, content:String)
```

- 回复问题答案

```
setGPTAnswer(content:String)
```

- 传输AI表盘图片预览图

只能在receiveRequestGPTPreviewImage方法回调后发起

```
setGPTPreImage(image: UIImage)
```

- 传输AI表盘图片

可主动发起或者receiveRequestGPTImage回调后发起

```
startGPTScreen(_ image: UIImage, _ imageSize :ScreenImageSize, _ isCricle: Bool = false, _ compressionType: Int)
```

- 修改AI表盘布局

```
setupGPTScreenContent(content: ScreenContent)
```

content的MD5值需设置为背景图的MD5值，当MD5值为""时,背景图片恢复默认背景

- 获取AI表盘布局

```
getGPTScreenContent(_ handler: @escaping screenContentHandler)
```

4.47 日历

部分手表支持同步日历功能

- 获取日历配置

```
getCalendarConfig(handler: @escaping calendarConfigHandler)
```

- 设置日历

```
setCalendar(calendar:CRPCalendarModel)
```

- 获取日历信息

日历功能主要以App为准

```
getCalendar(id: Int, handler: @escaping calendarHandler)
```

- 删除指定id日历

```
deleteCalendar(id: Int)
```

- 删除所有日历

```
deleteAllCalendar()
```

- 设置日历提醒设置

```
setCalendarReminder(model: CRPCalendarRemindModel)
```

- 获取日历提醒设置

```
getCalendarReminder(handler: @escaping calendarRemindHandler)
```

4.48 扫码配对

部分手表支持扫码配对,当CRPDiscovery的scan_code_bind属性为true时表示该手表支持扫码绑定功能

- 设置配对随机码，配对状态将通过receiveConnectConfirm返回

```
connectConfirmByRandomCode(randomCode:Int)
```

- App主动触发配对结果

```
setPairReply(allow: Bool)
```

5 版本更新日志

2.5

添加步数步数统计接口

2.6

添加生理周期相关接口

2.7

移除非必要的第三方库依赖(Alamofire)

2.8

修改打包版本为Xcode11-swift5.1

2.9

添加心电测量方法

3.1

添加久坐提醒设置有效时间

3.7

添加体温功能

3.8

步数数据添加时间统计

3.9

增加联系人同步功能

3.10

增加疲劳度测量功能

3.11

增加新版本的全天血氧，血压，体温测量功能

3.12

增加吃药提醒功能

3.13

增加获取单次测量记录和屏幕轻触唤醒开关

3.14

增加开启运动模式

3.15

增加压力测量

增加新版本锻炼数据

增加电子名片功能

3.16

增加GPS功能

兼容RealTek8763固件升级

增加震动强度

睡眠检测时间设置

新版本天气设置，新版本闹钟设置和新版本消息推送设置

增加全天压力测量功能

适配杰里平台

增加获取运动中的运动数据

3.16.26

增加股票

增加GPT功能

3.18.1

增加扫码配对相关方法

增加日历相关接口