

Development Guide

Engineering configuration

Development considerations

1 Data model

- 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
- 1.9 CRPTrainingRecordModel
- 1.10 CRPSleepRecordModel
- 1.11 CRPTimingHRRecordModel
- 1.12 CRPTimingHRVRecordModel
- 1.13 CRPTimingO2RecordModel
- 1.14 CRPTimingStepsRecordModel
- 1.15 CRPContinueActivitysRecordModel
- 1.16 CRPActivityReminderModel
- 1.17 CRPHR Remind
- 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 CRPSressRecordModel

2 Agent method

- 2.1 Returns the current connection status of the Ring
- 2.2 Return current Bluetooth status
- 2.3 Receive real-time sports step data
- 2.4 Receive heart rate measurement (single heart rate measurement)
- 2.5 Receive real-time heart rate data
- 2.6 Receive heart rate variability measurement results
- 2.7 Receive blood oxygen measurement results
- 2.8 Receive firmware upgrade progress and status
- 2.9 Receive sports data record list (optional)
- 2.10 Receive sports mode status (optional)
- 2.11 Receive wearing status (optional)
- 2.12 Receive daily goal reminders (optional)
- 2.13 Receive sedentary reminder (optional)
- 2.14 Receive goal achievement in workout (optional)
- 2.15 Receive pressure measurement results

3 Ring scanning and connection

- 3.1 Initialization
- 3.2 Ring scan
- 3.3 Cancel scan
- 3.4 Connection
- 3.5 Unbind

4 Ring Interaction

- 4.0 Sync time
- 4.1 Firmware Upgrade
- 4.2 Query the power of the ring
- 4.3 User Information
- 4.4 Active steps
- 4.5 Sleep
- 4.6 Exercise Goals
- 4.7 Sedentary reminder
- 4.8 Heart rate warning
- 4.9 Heart Rate
- 4.10 Blood Oxygen

- 4.11 HRV Fatigue
- 4.12 Exercise data
- 4.13 Obtain activity volume
- 4.14 Wearing status
- 4.15 shutdown
- 4.16 Restore factory settings
- 4.17 Test command
- 4.18 body temperature
- 4.19 Measurement Status
- 4.20 restart
- 4.21 Pressure
- 4.22 Ring Configuration Items
- 5 Version update log

DEVELOPMENT GUIDE

ENGINEERING CONFIGURATION

- Import the framework package

Import CRPSmartRing.framework into the project embedded binary file.

DEVELOPMENT CONSIDERATIONS

- Configuration permission

Configuring permissions in Info.plist:

Add App streamings using CoreBluetooth under Required background modes (Bluetooth Permissions)

Add Allow Arbitrary Loads under App Transport Security Settings and set YES (network access)

Add Bluetooth Permissions for Privacy - Bluetooth Always Usage Description

1 DATA MODEL

1.1 CRPDiscovery

Modifier	Name	Explain
String	LocalName	Device name
Int	RSSI	Rssi
String	Mac	Mac address

1.2 CRPStepModel

Modifier	Name	Explain
Int	steps	steps
Int	distance	distance
Int	calory	calory
Int	time	time (When the value is greater than -1, it is the exercise statistics time, and the value -1 means that the device does not support exercise time statistics)

1.3 CRPSleepModel

Modifier	Name	Explain
Int	deep	Deep sleep
Int	light	Light sleep
(Dictionary<String,String>)	detail	Sleep data details

Sleep data details

Modifier	Name	Explain
String	type	state 0: Awake 1:Light 2: Deep 3: rapid eye movement
String	total	Length of sleep (Minute))
String	start	start time
String	end	end time

1.4 CRPUserInfoModel

Modifier	Name	Explain
Int	height	height
Int	weight	weight
Int	age	age
GenderOption	gender	gender
Int	stepLength	step Length

1.5 CRPNewTrainingModel

Modifier	Name	Explain
Int	id	serial number
Int	starttime	start time (timestamp)
Int	endtime	end time (timestamp)
Int	availldTime	valid time (seconds)
Int	hrAvg	Average heart rate
CRPTrainingType	type	exercise type
Int	Step	Number of steps
Int	Distance	Distance
Int	kcal	Calories
(Int)	heartRate	heart rate data
CRPTrainingGoalType	goalType	sports goal type
Int	goal	target value
Int	hrAddress	heart rate address
Int	hrLength	heart rate length

1.6 CRPSportRecord

Modifier	Name	Explain
Int	id	serial number
Int	startTime	start time (timestamp)
Int	type	type

1.7 CRPTrainingGoalsModel

Modifier	Name	Explain
Int	steps	number of steps
Int	distance	Mileage (meter)
Int	kcal	Calories
Int	exerciseTime	exercise time (minutes)

1.8 CRPTrainingGoalStateModel

Modifier	Name	Explain
Bool	open	switch
(Int)	weekday	Exercise day array (refer to CRPWeekday) example: (CRPWeekDay.sun.rawValue, CRPWeekDay.thu.rawValue, CRPWeekDay.mon.rawValue)

1.9 CRPTrainingRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
Int	step	number of steps
Int	kcal	Calories (small calories)
Int	distance	distance (cm)
Int	exerciseTime	exercise time (minutes)

1.9 CRPTrainingRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
Int	step	number of steps
Int	kcal	Calories (small calories)
Int	distance	distance (cm)
Int	exerciseTime	exercise time (minutes)

1.10 CRPSleepRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
Int	deep	deep sleep
Int	light	light sleep
Int	rem	REM
(Dictionary<String,String>)	detail	sleep data details

1.11 CRPTimingHRRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
(Int)	hearts	Heart rate data (288, one every 5 minutes)

1.12 CRPTimingHRVRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
(Int)	hrvs	Heart rate variability data (288, one every 5 minutes)

1.13 CRPTimingO2RecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
(Int)	o2s	blood oxygen data (288, one every 5 minutes)

1.14 CRPTimingStepsRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
(Int)	steps	Statistical data of half-hour steps (48, one half-hour)

1.15 CRPContinueActivitysRecordModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
(Int)	activities	activity statistics (1440, one per minute)

1.16 CRPActivityReminderModel

Modifier	Name	Explain
Bool	open	switch
Int	period	Sedentary reminder period (minutes)
Int	steps	maximum number of steps
Int	startHour	start time (24-hour format)
Int	endHour	end time (24-hour format)

1.17 CRPHR Remind

Modifier	Name	Explain
Bool	isRemind	switch (remind or not)
Int	max	heart rate reminder value

1.18 CRPHeartRecordModel

Modifier	Name	Explain
Int	value	measurement result
Int	time	timestamp

1.19 CRPO2RecordModel

Modifier	Name	Explain
Int	value	measurement result
Int	time	timestamp

1.20 CRPHRVRecordModel

Modifier	Name	Explain
Int	hrv	Heart Rate Variability
Int	time	timestamp

1.21 CRPGsensorInfoModel

Modifier	Name	Explain
Int	x	x-axis offset
Int	y	y-axis offset
Int	z	z-axis offset

1.22 CRPBatteryInfoModel

Modifier	Name	Explain
Int	value	power
Int	voltage	Voltage (in mv)

1.23 CRPActivityReminderStateModel

Modifier	Name	Explain
Bool	wearState	wear state (00: not worn, 01: worn)
Int	time	time (minutes)
Int	steps	Period steps

1.24 CRPSleepDetailModel

Modifier	Name	Explain
CRPSleepType	type	sleep state
Int	hour	hour
Int	minute	minute

1.25 CRPSleepTemperatureDataModel

Modifier	Name	Explain
Int	day	0: today; 1: yesterday; 2: the day before yesterday, and so on
(Int)	tempeartures	Body temperature data (288, one every 5 minutes)

1.26 CRPRingInfoModel

Modifier	Name	Explain
Int	color	color
Int	size	size
Int	type	type

1.27 CRPSressRecordModel

Modifier	Name	Explain
Int	stress	pressure value
Int	time	timestamp

2 AGENT METHOD

2.1 Returns the current connection status of the Ring

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

2.2 Return current Bluetooth status

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

2.3 Receive real-time sports step data

Use getStep or ring to generate new steps will be called back

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

2.4 Receive heart rate measurement (single heart rate measurement)

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

2.5 Receive real-time heart rate data

Generally only used for display and not for storage

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

2.6 Receive heart rate variability measurement results

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

2.7 Receive blood oxygen measurement results

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

2.8 Receive firmware upgrade progress and status

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

2.9 Receive sports data record list (optional)

(This method will be called back when using `getTrainingRecordList` or the ring ends the exercise)

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

2.10 Receive sports mode status (optional)

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

2.11 Receive wearing status (optional)

(00: not worn, 01: worn)

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

2.12 Receive daily goal reminders (optional)

(state: 0 (not reminded), 1 (reminded). After receiving the daily goal, you need to use the `setReceiveGoalReminder` method to notify the Ring App that the user has been notified)

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

2.13 Receive sedentary reminder (optional)

isReminder: (00: App does not need to be reminded, 01: App needs to be reminded)

The three data of wearState, time and step will only be answered when the sedentary reminder test command `setActivityReminderReach` is called

wearState: (00: not worn, 01: worn)

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

2.14 Receive goal achievement in workout (optional)

(state: 0 (not reminded), 1 (reminded) After receiving the exercise goal, you need to use the `setReceiveTrainingGoalReach` method to notify the Ring App that the user has been notified)

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

2.15 Receive pressure measurement results

```
receiveStress(_stress: Int)
```

3 RING SCANNING AND CONNECTION

3.1 Initialization

CRPSmartRingSDK is the entrance of the SDK, written in CRPSmartRingSDK singleton mode; setting the delegate will initialize CRPSmartRingSDK and create Bluetooth (reminder: creating Bluetooth is a time-consuming operation, which needs to be set in advance)

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

3.2 Ring scan

Normal scanning can only be started when permissions are allowed and Bluetooth is turned on. If the ring is found during the scanning process, use the `scan()` method to call back the ring found in the scan through the `progressHandler`, and the `completionHandler` will call back all the rings scanned in the entire scan. You can set the scan time. Because the Bluetooth scan operation is time-consuming, the recommended scan time is 10 seconds.

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

3.3 Cancel scan

Cancel ring scan.

```
interruptScan()
```

3.4 Connection

(CRPDiscovery) obtained by scanning, you can select the corresponding ring connection according to the localName and Mac address of CRPDiscovery

```
connet(_discovery: CRPDiscovery)
```

- Obtain the connected device in the system (the original CBPeripheral is obtained, if you need to connect this device, you need to use CBPeripheral to create CRPDiscovery and then call the connection method)

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

3.5 Unbind

Unbind the ring

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

- reconnect

```
reConnect()
```


4 RING INTERACTION

4.0 Sync time

Keep the ring consistent with the time on the App side

```
setTime()
```

4.1 Firmware Upgrade

- Query firmware version

Query the current ring firmware version.

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

Query the Mac address of the current ring

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

- Query new firmware

There is a new version of the firmware that can be upgraded by passing in the current version and the mac address to obtain it

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

- Goodix firmware upgrade

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

- interrupt upgrade

```
stopUpgrade()
```

4.2 Query the power of the ring

Query the current battery level of the ring (if the battery level is above 100, it is charging, if the battery level exceeds 101, it means it is charging, and the current battery level is the last two digits, for example: if the battery level is 159, it means it is charging, and the current battery level of the ring is 59)

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

4.3 User Information

- Get user information

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

- Set user information

```
setUserinfo(_ userinfo: CRPUserInfoModel)
```

4.4 Active steps

- Synchronize the current active steps and reply data through receiveSteps

```
getSteps()
```

- Obtain historical statistical step data

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

- Step statistics

Get step statistics (one value every half hour, 48 values in total)

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

4.5 Sleep

get sleep data

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

Set the sleep data of the day (instructions for testing only)

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

4.6 Exercise Goals

Exercise goals are divided into ordinary day exercise goals and exercise day exercise goals

- Get daily exercise goals

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

- Set daily exercise goals

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

- Get workout day goals

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

- Set workout day goals

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

- Set the exercise goal to achieve (only for test instructions, only for test instructions, after using this method, it will call back receiveDailyGoal(_ type: CRPTrainingGoalType, state: Int))

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

4.7 Sedentary reminder

- Query sedentary reminder settings

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

- Set sedentary reminder settings

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

- Test sedentary reminder (only for testing instructions, after using this method, the ring will return the relevant information of the current sedentary reminder through the receiveActivityReminder callback)

```
setActivityReminderReach()
```

4.8 Heart rate warning

- Obtain heart rate warning value

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

- Set heart rate warning value

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

4.9 Heart Rate

All heart rate-related real-time data will be called back through `receiveRealTimeHeartRate(_ heartRate: Int)`.

- Start dynamic heart rate measurement

Start a single heart rate measurement

```
setStartSingleHR()
```

- End a single heart rate measurement

End heart rate measurement. The measurement result is called back through `receiveHeartRate(_ heartRate: Int)`. (receiving a result of 255 or 0 means that the heart rate measurement is interrupted)

```
setStopSingleHR()
```

- Get a single heart rate measurement history

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

- Get timing heart rate measurement status

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

- Turn on timing heart rate measurement
The ring supports 24-hour regular heart rate measurement, starting from 0:00, and the measurement time interval can be set (0 is off, positive number is on, and the time interval is set to the measurement amount: positive number * 5 minutes)

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

- Obtain regularly measured heart rate data

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

4.10 Blood Oxygen

- Start a single blood oxygen measurement

Start measuring blood oxygen.

```
setStartSpO2()
```

- End single blood oxygen measurement

End the measurement of blood oxygen. If the measurement time is too short, there will be no measurement results. The result is called back through receiveSpO2(_ o2: Int) (receiving a result of 255 or 0 means the blood oxygen measurement is interrupted)

```
setStopSpO2()
```

- Obtain a single blood oxygen measurement history

```
getO2RecordData(_ handler: @escaping O2RecordDataHandler)
```

- Obtain the status of timed blood oxygen measurement

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

- Turn on timing blood oxygen measurement
The ring supports 24-hour timing measurement of blood oxygen, starting from 0:00, and the measurement interval can be set (0 is off, a positive number is on, and the time interval is set to be: positive number * 5 minutes)

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

- Get timed blood oxygen data

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

4.11 HRV Fatigue

- Start HRV single measurement

```
setStartHRV()
```

- End a single HRV measurement, if the measurement time is too short, there will be no measurement results. The result is called back through receiveHRV(_ hrv: Int) (the received result is 255 or 0, indicating that the HRV measurement is interrupted)

```
setStopHRV()
```

- Get a single HRV measurement history

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

- Get timed HRV measurement status

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

- Turn on timing measurement HRV

The ring supports 24-hour timing measurement of HRV, starting from 0:00, and the measurement time interval can be set (0 is off, a positive number is on, and the time interval is set to a positive number * 5 minutes)

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

- Acquire timed HRV measurement data

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

4.12 Exercise data

- Get the types of workouts your ring supports

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

- Set workout status

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

- Get current workout status

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

- Set a single exercise goal to achieve (only for testing instructions, after using this method, it will call back receiveTrainingGoal(_ type: CRPTrainingGoalType, state: Int))

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

- Get historical exercise records, return exercise list through receiveTrainingList(list: (CRPTrainingRecord)) callback

```
getTrainingRecordList()
```

- Get the detailed data of the corresponding exercise (use the id in the list returned in receiveTrainingList to query)

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

- Set workout data (for testing only)

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

4.13 Obtain activity volume

Get activity records (one value per minute, a total of 1440)

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

4.14 Wearing status

Get the wearing state and return the result through receiveWearState(_ state: Int)

```
getWearState()
```

4.15 shutdown

The ring shuts down.

```
shutDown()
```


4.16 Restore factory settings

- reset

```
reset()
```

4.17 Test command

- Get Ring GitHash

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

- Get ring GSensor data

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

- Obtain the power information data of the ring

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

4.18 body temperature

- Set body temperature measurement status

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

- Get body temperature measurement status

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

- Obtain body temperature measurement data

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

4.19 Measurement Status

- Get current body temperature measurement status

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

4.20 restart

- reboot

```
reboot()
```

- Get the number of restarts (only for testing)

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

4.21 Pressure

- Start pressure single measurement

```
setStartStress()
```

- End a single HRV measurement, if the measurement time is too short, there will be no measurement results. The result is called back through `receiveStress(_stress: Int)` (receiving a result of 255 or 0 means that the pressure measurement is interrupted)

```
setStopStress()
```

- Get a single pressure measurement history

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

4.22 Ring Configuration Items

- Get ring configuration items

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

- Set ring configuration items

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

5 VERSION UPDATE LOG

1.0.4

Increase body temperature function

Increase the acquisition of measurement status acquisition

Increase exercise state acquisition

Initiate motion and set motion target methods combined

1.0.5

Added broadcast to obtain ring configuration item information

Add the method of getting and setting ring configuration item information

Increase pressure single measurement method

1.0.6

1. Increase the method of obtaining historical single pressure measurement
2. Modify the method of obtaining historical single hrv measurement records
3. Modify the method of obtaining all-day hrv measurement records