

# APIHeartRate

---

sdk for heartrate band

pod v0.0.9

## Requirements

- iOS 10.0+
- Xcode 10.0+

## 安装

### CocoaPods

[CocoaPods](#) is a dependency manager for Cocoa projects.

如果你没有cocoapods

```
$ gem install cocoapods
```

在 [Podfile](#) 中添加:

```
source 'https://github.com/CocoaPods/Specs.git'
platform :ios, '10.0'
use_frameworks!

pod 'APIHeartRate'
```

接着运行:

```
$ pod install
```

### 手动安装

下载源文件APIHearateAPIV2.zip 解压后得到xcframework文件, 放入自己的项目

## 用法

### 引入头文件

在项目中引入

```
import APIHearateAPI
```

APIHeartRateManager

APIHeartRateManager是一个单例类，所有的连接和数据监听操作都通过它来实现  
持有一个引用

```
let apiManager = APIHeartRateManager.instance
```

注册监听

```
apiManager.addObserver(observer: self)
```

方法调用

```
//扫描设备
apiManager.startScan()
//获取电量
apiManager.getBatteryPower()
//获取厂商名
apiManager.getManufacturerName()
//获取Model名称
apiManager.getModelName()
....
```

接口列表

功能名称	方法	支持状态	固件支持版本
扫描设备	startScan(timeOut: Double)	✓	all
停止到扫描	stopScan()	✓	all
获取已连接设备	getConnectedDevice(uuid:[UUID])	✓	all
通过device对象连接设备	connectBluetooth(device: BleDiscoveryDevice)	✓	all

功能名称	方法	支持状态	固件支持版本
通过deviceID 连接设备	connectDeviceBy(deviceId: String)	✓	all
断开设备	disconnectDevice()	✓	all
注册事件监 听	registerBleListener(listener: APIHeartRateObserver)	✓	all
取消事件监 听	unregisterBleListener(listener: APIHeartRateObserver)	✓	all
获取设备电 量	getDeviceBattery()	✓	>= v1.1
获取厂商信 息	getDeviceManufacturer()	✓	>= v1.1
获取 ModelNum	getDeviceModelNum()	✓	>= v1.1
获取硬件版 本	getDeviceHardware()	✓	>= v1.1
获取软件版 本	getDeviceSoftware()	✓	>= v1.1
获取固件版 本	getDeviceFirmware()	✓	>= v1.1
获取系统ID	getDeviceSystemID()	✓	>= v1.1
设置心率最 大阈值	setDeviceThreshold(max: UInt8)	✓	>= v1.1
获取序列号	getDeviceSerial()	✗	---
获取步频	getDeviceStepFrequency()	✗	---
获取实时血 氧	getRealTimeOxygen()	✗	---
同步时间	syncTime()	✓	>= v1.1
主动读特征	readCharValue(charUUID: String)	✓	>= v1.1
主动设置特 征监听	setCharNotify(charUUID: String, enabled: Bool)	✓	>= v1.1

功能名称	方法	支持状态	固件支持版本
主动写特征	writeChar(charUUID: String, data:Data)	✔	>= v1.1
获取历史数据	-----	✘	---
Pod 0.0.7			
设置心率区间	setHeartRateThreshold(min: UInt8, max: UInt8)	✔	>= v1.1
ota接口	startSendOTAFile(data: Data) 状态监听接口 : bleOtaStauts(status: OtaStatus, progress: Float) ; bleOtaError(error: OtaError)	✔	>= v1.1
接收手环按钮切换值	armBandPlayStatusChange （被动）	✔	>= v1.1
长按5s接收解绑指令	armBandUnbind （被动）	✔	>= v1.2
恢复出厂设置	resetBand()	✘	---

APIHeartRateObserver

这个代理类返回所有错误和数据

```
// 成功更新阈值
func armBandMaxHeartRateUpdated()
// 成功同步时间
func armBandSystemTimeUpdated()
// 返回电量
func devicePower(power: String, device: BleDevice)
// 返回厂商信息
func deviceManufacturerName(manufacturerName: String, device: BleDevice)
// 返回Mac地址
func privateMacAddress(mac: String, device: BleDevice)
// 返回Model名称
func deviceModelString(modelString: String, device: BleDevice)
// 返回硬件版本
func deviceHardware(version: String, device: BleDevice)
// 返回固件版本
func deviceFirmware(version: String, device: BleDevice)
// 返回系统ID
func deviceSystemData(systemData: Data, device: BleDevice)
// 返回软件版本
```

```
func deviceSoftware(version: String, device: BleDevice)
//返回序列号
func deviceSerialNumber(serialNumber: String, device: BleDevice)
//返回心率
func armBandRealTimeHeartRate(hRInfo: HRInfo, device: BleDevice)
//返回当前设备连接状态
func bleConnectStatus(status: DeviceBleStatus, device: BleDevice?)
//返回实时步频
func armBandStepFrequency(frequencyDic: [String:String], device:
BleDevice)
//返回实时血氧
func armBandBloodOxygen(num: Int, device: BleDevice)
//扫描中
func didDiscoveryWith(discovery: [BleDiscoveryDevice])
//完成扫描
func didFinishDiscoveryWith(discovery: [BleDiscoveryDevice])
//连接断连相关错误
func bleConnectError(error: BleConnectError, device: BleDevice?)
//通用错误
func bleCommonError(error: BleCommonError)
//蓝牙适配器状态
func bleAvailability(status: BleAvailability)
```

## License

APIHeartRate is released under the MIT License.