

[toc]

扫描设备

1) 接口方法

```
startScan(timeOut: Double)
```

2) 接口描述：

- 扫描蓝牙设备

3) 参数:

字段名称	字段说明	类型	必填	备注
timeOut	扫描超时时间	Double	N	默认为空

5) 结果回调:

```
func didDiscoveryWith(devices: [BleDicoverDevice]) {}
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
devices	扫描结果数组	[BleDicoverDevice]	-	-

断开连接

1) 接口方法

```
func disconnectDevice()
```

2) 接口描述：

- 主动断开蓝牙设备

3) 参数: -

5) 结果回调:

```
func bleConnectStatus(status: DeviceBleStatus, device: BleDevice?) {}
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
status	蓝牙设备状态	DeviceBleStatus	-	connecting connected disconnecting disconnected 四个状态
device	连接的蓝牙设备	BleDevice	-	断开后为空

注册事件监听

1) 接口方法

```
func registerBleListener(listener: APIHeartRateObserver)
```

2) 接口描述 :

- 注册回调，所有的返回值都从以上回调中获取

3) 参数:

字段名称	字段说明	类型	必填	备注
listener	传入实现该协议的对象	APIHeartRateObserver	Y	

5) 结果回调: -

6) 结果参数说明: -

取消事件监听

1) 接口方法

```
func unregisterBleListener(listener: APIHeartRateObserver)
```

2) 接口描述 :

- 取消回调事件的监听

3) 参数:

字段名称	字段说明	类型	必填	备注
------	------	----	----	----

字段名称	字段说明	类型	必填	备注
listener	传入实现该协议的对象	APIHeartRateObserver	Y	

- 5) 结果回调: -
- 6) 结果参数说明: -

读取设备电量

- 1) 接口方法

```
func getDeviceBattery()
```

- 2) 接口描述 :
- 读取设备电量

- 3) 参数: -

- 5) 结果回调:

```
func devicePower(power: String, device: BleDevice) {}
```

- 6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
power	电量值	String	-	-

制造商信息

- 1) 接口方法

```
func getDeviceManufacturer()
```

- 2) 接口描述 :
- 读取制造商信息

- 3) 参数: -

- 5) 结果回调:

```
func deviceManufacturerName(manufacturerName: String, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
manufacturerName	厂商名称	String	-	-
device	当前连接的设备	BleDevice	-	-

Model信息

1) 接口方法

```
func getDeviceModelNum()
```

2) 接口描述 :

- 读取型号信息

3) 参数: -

5) 结果回调:

```
func deviceModelString(modelString: String, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
modelString	型号信息	String	-	-
device	当前连接的设备	BleDevice	-	-

硬件版本

1) 接口方法

```
func getDeviceHardware()
```

2) 接口描述 :

- 读取制硬件版本信息

3) 参数: -

5) 结果回调:

```
func deviceHardware(version: String, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
version	硬件版本	String	-	-
device	当前连接的设备	BleDevice	-	-

软件版本

1) 接口方法

```
func getDeviceSoftware()
```

2) 接口描述 :

- 读取设备软件版本信息

3) 参数: -

5) 结果回调:

```
func deviceSoftware(version: String, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
version	软件版本	String	-	-
device	当前连接设备	BleDevice	-	-

固件信息

1) 接口方法

```
func getDeviceFirmware()
```

2) 接口描述：

- 读取固件信息

3) 参数: -

5) 结果回调:

```
func deviceFirmware(version: String, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
version	固件版本	String	-	-
device	当前连接的设备	BleDevice	-	-

系统ID

1) 接口方法

```
func getDeviceSystemID()
```

2) 接口描述：

- 读取设备系统ID信息

3) 参数: -

5) 结果回调:

```
func deviceSystemData(systemData: Data, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
systemData	系统ID	Data	-	7字节的值
device	当前连接的设备	BleDevice	-	-

~~序列号信息(未实现)~~

1) 接口方法

```
func getDeviceSerial()
```

2) 接口描述：

- 读取设备序列号信息

3) 参数: -

5) 结果回调:

```
func deviceSerialNumber(serialNumer: String, device: BleDevice)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
serialNumer	序列号	String	-	-
device	当前连接的设备	BleDevice	-	-

步频(未实现)

1) 接口方法

```
func getDeviceStepFrequency()
```

2) 接口描述：

- 获取步频信息

3) 参数: -

5) 结果回调:

```
func armBandStepFrequency(frequencyDic: [String: String], device: BleDevice) {}
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
------	------	----	----	----

字段名称	字段说明	类型	必填	备注
frequencyDic	步频字典	[String:String]	-	-
device	当前连接的设备	BleDevice	-	-

~~实时血氧(未实现)~~

1) 接口方法

```
func getRealTimeOxygen()
```

2) 接口描述：

- 获取实时血氧

3) 参数: -

5) 结果回调:

```
func armBandBloodOxygen(num: Int, device: BleDevice) {}
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
num	血氧值	Int	-	-
device	当前连接的设备	BleDevice	-	-

同步时间

1) 接口方法

```
func syncTime()
```

2) 接口描述：

- 同步当前设备时间

3) 参数: -

5) 结果回调:


```
func armBandSystemTimeUpdated() {}
```

6) 结果说明： -

心跳阈值

1) 接口方法

```
func setHeartRateThreshold(min: UInt8,max: UInt8)
```

2) 接口描述：

- 设置心跳阈值，心跳小于左边界为绿灯，大于右边界为红灯

3) 参数:

字段名称	字段说明	类型	必填	备注
min	左边界	UInt8	Y	-
max	右边界	UInt8	N	默认为0

5) 结果回调:

```
func armBandMaxHeartRateUpdated() {}
```

切换播放状态

1) 接口方法： -

2) 接口描述：

- 点击设备按钮，切换状态，没有主动调用接口

3) 参数 -

5) 结果回调:

```
func armBandPlayStatusChange() {} //每次点按钮均回调一次
```

长按配对

1) 接口方法： -

2) 接口描述：

- 长按关机键关机并保持不放5s，设备进入配对模式，并传回结果回调
- 固件版本>=v1.2

3) 参数 -

5) 结果回调:

```
func armBandUnbind() {} //进入配对模式后回调一次
```

设备充电状态

1) 接口方法： -

2) 接口描述：

- 此接口也是被动回调，一共有充电中、充电完成、未充电三个状态

3) 参数 -

5) 结果回调:

```
func batteryStatus(state: BatteryStatus) {}
```

6) 返回结果说明：

字段名称	字段说明	类型	必填	备注
state	充电状态	BatteryStatus	-	normal,charging,full,unknow

心率测量中状态回调

1) 接口方法： -

2) 接口描述：

- 心率测量中状态回调，此时不出心率值

3) 参数 -

5) 结果回调:

```
func heartRateInMeasuring()
```

恢复出厂设置(未实现)

1) 接口方法

```
func resetBand()
```

2) 接口描述：

- 恢复出厂设置

3) 参数: -

5) 结果回调: -

6) 结果说明： -

OTA

1) 接口方法

```
func startSendOTAFile(data: Data)
```

2) 接口描述：

- 进行OTA操作

3) 参数:

字段名称	字段说明	类型	必填	备注
data	ota文件数据	Data	Y	-

5) 结果回调:

```
func bleOtaStauts(status: OtaStatus, progress: Float);  
func bleOtaError(error: OtaError)
```

6) 结果参数说明:

字段名称	字段说明	类型	必填	备注
status	ota状态	OtaStatus	-	start,erase,inprogress,finished,failed
progress	更新进度	Float	-	-
error	异常信息	OtaError	-	invalidFile,interrupted,invalidResponse,reSend