

Coreaiot 信标 SDK iOS

开发环境

- Xcode 12.3 以上
- iOS 10.0 以上

安装

Cocoapods（推荐）

```
pod 'RTLSSensor', :git => 'https://github.com/coreaiot/RTLSSensor.git',  
:tag => '1.0.1'
```

下载地址

<https://github.com/coreaiot/RTLSSensor/releases>

用户授权

- NSBluetoothPeripheralUsageDescription

使用方法

```
import RTLSSensor  
  
let ble = Ble()  
ble.onStateChange { state in  
    if state == .poweredOn {  
        // 设置 ID  
        ble.id = 0x1234  
    }  
}
```

```
// 设置报警状态
ble.alarm = true
// 设置电池电量
ble.battery = 10

// 开始广播
try! ble.start()

// 停止广播
ble.stop()
}
}
```

Ble 类

构造方法

```
public override init()
```

属性

```
/**
 * 蓝牙状态改变事件
 * 只会触发一次
 * 请调用 ble.stateChange.then 或 ble.onStateChange 后状态是 .poweredOn 后再
使用 start 或 stop 方法
 */
public let stateChange: Promise<CBManagerState>

/**
 * ID
```

```

* MAC 地址的后两个字节
* 值域: [0, 0xffff]
* 不可以是 [56E6, 9730, 0BAA, 8E5C, C403,DB34, 67F7, 2745, ACCC, 2F61]
*/
public var id: UInt = 0

/**
* 报警状态
*/
public var alarm: Bool = false

/**
* 电池电量
* 值域: [0, 10]
* 实际电量为 值 * 10 * 100%
* 例: 值为 5 表示电池电量为 50%
*/
public var battery: UInt = 0

/**
* peripheral 是否处于广播状态
*/
public private(set) var isAdvertising = false

```

方法

```

/**
* 蓝牙状态改变事件
* 每次改变都会调用回调方法
* 请调用 ble.onStateChange 或 ble.stateChange.then 后状态是 .poweredOn 后再
使用 start 或 stop 方法
*/
public func onStateChange(_ callback: @escaping (CBManagerState) ->
Void)

```

```
/**
 * 开始广播
 */
public func start() throws

/**
 * 停止广播
 */
public func stop()
```

Changelog

1.0.1 iOS SDK – Apr 15, 2021

- 添加方法 `onStateChange` 用于多次监听蓝牙状态改变事件。
- 添加属性 `isAdvertising` 用于获取 `peripheral` 的广播状态。