**BYBlueTooth使用文档**

**开发环境**

BYBlueTooth使用Objective-C进行开发，开发环境为Xcode8.3，使用iOS10.3进行真机调试。

**导入SDK**

SDK使用.frameWork方式进行打包。在使用SDK时，只需要将 BYBlueTooth.framework 拉进项目中即可使用。



拉进项目时注意勾选“Copy items if needed”。



导入完添加下面语句即可使用：

//OC

#import <BYBlueTooth/BYBlueTooth.h>

//Swift

import BYBlueTooth

# 基本使用

### 获取BYBlueTooth单例:

在使用BYBlueTooth时，都是对这个单例进行操作。

//OC

BYBlueTooth \*bb = [BYBlueTooth shareBYBlueTooth];

//Swift

let bb = BYBlueTooth.share()

#### 搜索蓝牙设备：

在链式语句的最后需要调用begin闭包。

//OC

bb.scanForPeripherals().begin();

//Swift

bb?.scanForPeripherals()()?.begin()()

#### 连接蓝牙设备（需要持有CBPeripheral对象）：

enjoy:完成一整套蓝牙流程（包括连接，搜索服务，发现特征值，读取订阅特征值），调用此方法前前面必须要有having方法

begin:开始执行链式语句

stop(sec:):设置链式语句超时时间，sec秒后没有连接成功则断开

//OC

bb.having(\_currPeripheral).enjoy().begin().stop(10);

//Swift

bb?.having()(currPeripheral)?.enjoy()()?.begin()()?.stop()(10)

#### 更多自定义操作：

执行完搜索与连接设备后就已经完成蓝牙操作相关流程了，如果想要对设备进行更多操作，可以对具体环节进行自定义操作，具体看**Blocks设置说明** 章节。

#### ****注意：Blocks设置需要在begin前执行！****

# Blocks设置说明

在想要对蓝牙扫描，连接，获取服务，发现特征值，订阅特征值的过程中介入更多自定义操作，可以实现以下Blocks。

#### ****注意：Blocks设置需要在begin前执行！****

Blocks设置可以参与蓝牙设备流程的各个环节，以及一些自定义操作。

#### ****Blocks设置时需要传入标识符Key参数，推荐使用类名 。****

NSStringFromClass([self class])

#### 注意：dealloc函数中需要显式删除已经设置的block，只需要传入标识符Key即可。

- (void)dealloc {

[bb removeBlockWithKey:NSStringFromClass([self class])];

}

### 蓝牙流程相关Blocks

#### 设备状态改变的block

- (void)setBlockOnCentralManagerDidUpdateStateWithKey:(NSString \*)key block:(void (^)(CBCentralManager \*central))block

#### 找到符合setFilterOnDiscoverPeripherals规则的Peripherals的block

如果没有实现setFilterOnDiscoverPeripherals，则会返回搜索到的CBPeripherals

- (void)setBlockOnDiscoverToPeripheralsWithKey:(NSString \*)key block:(void (^)(CBCentralManager \*central,CBPeripheral \*peripheral,NSDictionary \*advertisementData, NSNumber \*RSSI))block;

#### 连接Peripherals成功的block

- (void)setBlockOnConnectedWithKey:(NSString \*)key block:(void (^)(CBCentralManager \*central,CBPeripheral \*peripheral))block;

#### 连接Peripherals失败的block

- (void)setBlockOnFailToConnectWithKey:(NSString \*)key block:(void (^)(CBCentralManager \*central,CBPeripheral \*peripheral,NSError \*error))block;

#### 断开Peripherals的连接的block

在断开连接Block中，如果error参数不为nil，说明此处是异常断开，可以在此进行重连操作。如果error为nil，则说明此Block在调用cancelPeripheralConnection后触发

- (void)setBlockOnDisconnectWithKey:(NSString \*)key block:(void (^)(CBCentralManager \*central,CBPeripheral \*peripheral,NSError \*error))block;

#### 设置查找到服务的block

- (void)setBlockOnDiscoverServicesWithKey:(NSString \*)key block:(void (^)(CBPeripheral \*peripheral,NSError \*error))block;

#### 设置查找到Characteristics的block

- (void)setBlockOnDiscoverCharacteristicsWithKey:(NSString \*)key block:(void (^)(CBPeripheral \*peripheral,CBService \*service,NSError \*error))block;

#### 设置获取到最新Characteristics值的block

此处读取到的值为原始值NSData，使用characteristic.value获取

- (void)setBlockOnReadValueForCharacteristicWithKey:(NSString \*)key block:(void (^)(CBPeripheral \*peripheral,CBCharacteristic \*characteristic,NSError \*error))block;

#### 设置获取到完整数据的Block，完整数据不包含$，\*，校验和以及结束符<CR><LF>

此处完整数据是指SDK根据北云数据协议，对读取到的所有特征值的value进行处理获得的一条完整格式、且经过校验的数据。在返回这条数据时，去除协议开头的$符号以及校验和、结束符<CR><LF>，方便开发者进行数据处理。

- (void)setBlockOnReadCompleteValueWithKey:(NSString \*)key block:(void (^)(NSString \*value))block;

#### 设置获取到GGA定位数据的Block

在所有获取到的完整数据中，根据协议头进行判断。在获取到GGA协议数据后进行解析，并返回一个BYLocationModel，该model的包括了从协议中提取出来的所有相关信息。

- (void)setBlockOnReadGGAValueWithKey:(NSString \*)key block:(void (^)(BYLocationModel \*value))block;

### 过滤器Blocks

#### 设置查找Peripherals的规则

这个Block返回一个Bool值，用于过滤蓝牙设备特殊的命名。搜索符合规定的Peripherals将会通过setBlockOnDiscoverToPeripheralsWithKey返回。

- (void)setFilterOnDiscoverPeripherals:(BOOL (^)(NSString \*peripheralName, NSDictionary \*advertisementData, NSNumber \*RSSI))filter;

#### 设置指定的Service名称

指定特征值所属的Service名称。设置后SDK不会去搜索其他Service名称的特征值。

- (void)setFilterForServiceName:(BOOL (^)(NSString \*serviceName))filter;

#### 指定用于读取数据的Characteristic名称

只从指定的特征值读取数据。如果没有搜索到指定的特征值名称，将不能读取数据。

- (void)setFilterForReadCharacteristicName:(BOOL (^)(NSString \*characteristicName))filter;

#### 指定用于写入数据的Characteristic名称

只从指定的特征值写入数据。如果没有搜索到指定的特征值名称，将不能写入数据。

- (void)setFilterForWriteCharacteristicName:(BOOL (^)(NSString \*characteristicName))filter;

# 其他接口说明

#### 设置是否显示SDK的打印信息：

YES为打印，NO为不打印

//OC

[BYBlueTooth isShowLog:YES];

//Swift

BYBlueTooth.isShowLog(true)

#### 断开与CBPeripheral的连接（需要在连接到CBPeripheral后持有该对象）：

- (void)cancelPeripheralConnection:(CBPeripheral \*)peripheral;

#### 停止扫描

- (void)cancelScan;

#### 向Characteristic写入数据，格式为ASCII

由于北云协议使用的是ASCII格式来制定协议，所以此处提供直接用ASCII格式来写入的接口

- (void)writeToPeripheralWithASCIIString:(NSString \*)string;

#### 向Characteristic写入数据，格式为HEX

写入16进制格式组成的数据

- (void)writeToPeripheralWithHEXString:(NSString \*)string;

### 使用地址，端口号，用户名，密码，挂接点进行cors登录

- (void)corsLoginWithip:(NSString \*)ip port:(NSString \*)port user:(NSString \*)user password:(NSString \*)password mount:(NSString \*)mount;