

呼吸家 iOS 蓝牙 SDK 模块使用介绍

修订历史记录

| 版本 | 日期 | AMD | 修订者 | 说明 |
|------|------------|-----|-----|---------------|
| V1.1 | 2016/12/13 | A | 蒲小龙 | iOS 蓝牙 SDK 更新 |

1 iOS 蓝牙 SDK 介绍

1.1 BLEHelper 类，是一个单例管理整个蓝牙的连接更新操作，用类方法 shareBLEHelper 创建实例，蓝牙外设测试的数据会通过代理方法 - (void)BLEHelper:(BLEHelper *)helper detectedResults:(NSArray *)detectedResults currentResult:(SGHolderDetectedResult *)result;回调。

1.2 SGHolderDetectedResult 这是一个蓝牙测试数据的模型，将蓝牙的文本数据转换为模型数据，实例变量`pefGroups`是吹气波形图数据包含了 600 组 pef 值的数组。

2 iOS 蓝牙 SDK 使用介绍

2.1 可用直接将 SDK 文件拖到项目中

2.2 获取蓝牙单例对象调用类方法`self.helper = [BLEHelper shareBLEHelper];`在 demo 中我强引用了蓝牙单例，方便其他方法中访问。

2.3 设置代理`self.helper.delegate = self;`并遵守协议`BLEHelperDelegate`

2.4 连接蓝牙之前需要先传入检测人资料，调用单例方法`[self.helper requestMatchInfoWithHolderDict:matchInfo];`检测人资料需要以字典的形式当做参数格式如下：

```
NSMutableDictionary *matchInfo = @{
    @"PEF" : @"601",
    @"FEV1" : @"4.44",
    @"FVC" : @"5.22",
    @"gender" : @"0",
    @"height" : @"176",
    @"weight" : @"67",
    @"detectedNo" : @"6381",
    @"mobile" : @"13480016536",
    @"birthdate" : @"1989-07-07",
    @"saleChannel" : @"0",
    @"deviceNo" : @"A116080011"
};
```

2.5 上述方法完成后就可以连接蓝牙外设，连接蓝牙设备方法`[self.helper connectedPeripheral:nil imei:imeiNumber];`这里的Peripheral一般为空，`BLEHelper`类中会根据IMEI（设备号）扫描周边的蓝牙外设并自动连接，在 demo 中我在用户点击连接按钮

的时候触发此方法连接蓝牙外设，可以根据具体情况选择时机连接蓝牙。

2.6 接收蓝牙设备的测试数据调用如下代理方法，在 demo 接口中有注释说明，这里有两个参数，`detectedResults` 是多组检测数据，例如设备中保存了 20 组数据，每传输一组数据都会保存在这个数组中。`currentResult` 这个参数是当前传输的数据。

```
- (void)BLEHelper:(BLEHelper *)helper
detectedResults:(NSArray *)detectedResults
currentResult:(SGHolderDetectedResult *)result
{
    NSLog(@"results count - %zd \n current result
time %@",detectedResults.count,result.saveTime);
}
```

2.7 蓝牙状态监听代理方法 - (void)BLEHelper:(BLEHelper *)helper ConnectStatusType:(BLEConnectStatusType)type; 当蓝牙状态改变会调用此代理方法，一共 6 种枚举状态：

```
typedef NS_ENUM(NSInteger,BLEConnectStatusType) {
    BLEConnectOnStatus = 0,//连接
    BLEConnectingStatus,//连接中
    BLEConnectOffStatus,//断开
    BLEWillUpdateDeviceStatus,//将要更新
    BLEUpdatingDeviceStatus,//更新中
    BLEDidUpdateDeviceStatus//更新完成
};
```

2.8 单例中实例对象`peripherals`会保存当前手机扫描到的蓝牙外设，只有在调用`[self.helper connectedPeripheral:nil imei:imeiNumber];`此方法后`peripherals`此数组中才会扫描并保存。扫描以及保存规则根据宏`kBLEPeripheral`设置，SDK中以包含B的设备才被保存，后面的宏定义不需要更改。

2.9 蓝牙系统更新进度监听代理方法`-(void)BLEHelperWriteValueCount:(NSInteger)count atIndex:(NSInteger)index;`如果设备需要更新会从呼吸家服务器请求对应的更新包，参数`count`是总共分割了多少个包`index`是正在给蓝牙设备传输第几个包。

2.10 当不需要连接蓝牙需要断开可用调用方法:`-(void)cancelBLEConnect;`当调用此方法，会断开当前连接的蓝牙外设。

3 接口文档说明

3.1 获取蓝牙单例接口`[self.helper requestMatchInfoWithHolderDict:matchInfo];`,`matchInfo`传参说明如下图所示：

| 参数名 | 类型 | 长度 | 可空标志 | 说明 |
|-------------|----------|----|----------|--------------|
| PEF | NSString | | NULL | PEF 预计值 |
| FEV1 | NSString | | NULL | FEV1 预计值 |
| FVC | NSString | | NULL | FVC 预计值 |
| mobile | NSString | | NULL | 手机号 |
| detectedNo | NSString | | NOT NULL | 用户 id |
| saleChannel | NSString | | NOT NULL | 渠道号,从呼吸家申请得到 |
| gender | NSString | | NOT NULL | 检测人性别 |
| deviceNo | NSString | | NOT NULL | 设备 IMEI 号 |

| | | | | |
|-----------|----------|--|----------|------------------|
| height | NSString | | NOT NULL | 检测人身高 |
| weight | NSString | | NOT NULL | 检测人体重 |
| birthdate | NSString | | NOT NULL | 检测人生日 yyyy-MM-dd |

3.2 蓝牙连接 `[self.helper connectedPeripheral:nil imei:imeiNumber];`

| 参数名 | 类型 | 长度 | 可空标志 | 说明 |
|-----------------|------------------------|----|----------|----------|
| helper | BLEHelper | | NOT NULL | 蓝牙单例对象 |
| detectedResults | NSArray | | NULL | 所有检测数据数组 |
| result | SGHolderDetectedResult | | NULL | 当前测试数据 |

3.3 蓝牙状态监听代理方法 `(void)BLEHelper:(BLEHelper *)helper ConnectStatusType:(BLEConnectStatusType)type;`

| 参数名 | 类型 | 长度 | 可空标志 | 说明 |
|--------|----------------------|----|----------|----------|
| helper | NSString | | NOT NULL | 蓝牙单例对象 |
| type | BLEConnectStatusType | | NOT NULL | 所有检测数据数组 |

3.4 设备更新进度监听代理方法 - (void)BLEHelperWriteValueCount:(NSInteger)count atIndex:(NSInteger)index;

| 参数名 | 类型 | 长度 | 可空标志 | 说明 |
|-------|-----------|----|----------|--------|
| count | NSInteger | | NOT NULL | 更新总数 |
| index | NSInteger | | NOT NULL | 当前更新位置 |

4 注意事项。

4.1.iOS 版本要求: ≥ 8.0 设备支持 BT 4.0 使用 ARC 内存管理，使用 Xcode 8.1 打包

4.2.Framework 依赖 Corebluetooth.framework，开发前记得导入头文件。

4.3．发布请使用 release 版本，debug 使用 debug 版本的静态库