倍泰血压计蓝牙SDK接口说明-iOS

|  |  |  |
| --- | --- | --- |
| 版本号 | 描述 | 日期 |
| V1.0.10 | 1. 新增获取设备电量接口与收到电量信息接口；  2. 新增下发设置单位的响应接口；  3. 新增心律不齐字段（isArrhythmia）；  4. 新增接口setDeviceBleName:用以指定设备蓝牙名；  5. 增加型号13B4-LT专用的熄屏与唤醒接口。 | 2022-05-11 |
| V1.0.9 | 蓝牙连接成功接口中添加了序列号字段（serialNumber） | 2022-01-13 |

本接口文档适用于倍泰Belter\_BT血压计SDK。

SDK通过BTBloodPBLEManager实例来管理血压计设备与手机的蓝牙连接与业务逻辑。本文提到的示例均可参考示例项目代码BTBloodPressureSDK-Demo。

# 一、环境配置说明

BTBloodPressureSDK支持在iOS 9.0及以上使用。

1. 将BTBloodPressureSDK.framework拖入项目中，注意勾选Copy items if needed.
2. 在target的Build phases中导入BTBloodPressureSDK.framework



1. 在需要使用的地方，引入头文件即可

示例：

#import "BTBloodPressureSDK/BTBloodPBLEManager.h"

# 二、接口说明

## 实例化

BTBloodPBLEManager为SDK管理类。要使用BTBloodPBLEManager还需成为它的delegate，并遵循BTBloodPBLEManagerDelegate协议。

1.1 接口：

/\*\*

实例化一个BTBloodPBLEManager

\*/

+ (instancetype)shareBTBloodPBLEManager;

@property (nonatomic,weak) id<BTBloodPBLEManagerDelegate> delegate;

1.2 示例：

self.bleManager = [BTBloodPBLEManager shareBTBloodPBLEManager];

self.bleManager.delegate = self;

## 开始扫描血压计设备

该接口用于开始搜索扫描周边的血压计设备，只有设备蓝牙是打开的，才可能被搜索到。

/\*\*

开始扫描周边设备

\*/

- (void)startScan;

## 停止扫描

该接口用于需要终止扫描时。

/\*\*

停止扫描

\*/

- (void)stopScan;

## 断开当前连接

该接口用于断开当前与血压计的蓝牙连接。

/\*\*

断开当前连接

\*/

- (void)disconnectDevice;

## 下发设置血压计单位

该接口用于，在APP与血压计连接后，设置血压计显示的单位。

/\*\*

下发设置血压计单位

@param unit 单位（mmHg/kPa）

\*/

- (void)sendToSetUnit:(BTBloodPressureUnit)unit;

其中，BTBloodPressureUnit为枚举类，代表血压计支持的两种单位：

typedef enum : NSUInteger {

BTBloodPressureUnitMmHg, // mmHg

BTBloodPressureUnitKPa, // kPa

} BTBloodPressureUnit;

## 下发获取设备电量

该接口用于查询血压计的电量。

/\*\*

下发获取设备电量

\*/

- (void)sendToGetDevicePower;

## 指定设备蓝牙名

此接口需要在BTBloodPBLEManager的初始化方法shareBTBloodPBLEManager后即刻调用，以便BTBloodPBLEManager可以扫描到设备。

此接口非必需，如果您不需要指定设备的蓝牙名称，可忽略此接口。SDK会以“Belter\_BT”为关键字去搜索连接。

/\*\*

指定设备蓝牙名

@param bleNameKeyword 设备蓝牙名关键字，当设备蓝牙名带有编码时，只传入关键字即可。

\*/

- (void)setDeviceBleName:(NSString \*)bleNameKeyword;

## 下发关闭设备显示屏（仅适用于指定型号13B4-LT）

该接口用于在APP与血压计连接后，下发指令关闭血压计上的屏幕显示。

仅用于型号为13B4-LT的血压计。

- (void)sendToTurnOffScreen\_13B4LT;

## 下发唤醒设备显示屏（仅适用于指定型号13B4-LT）

该接口用于在APP与血压计连接后，下发指令打开血压计上的屏幕显示。

仅用于型号为13B4-LT的血压计。

- (void)sendToTurnOnScreen\_13B4LT;

## 其他字段说明

###### 10.1 bleState

手机蓝牙状态，代表当前手机系统蓝牙的开/关等状态。

/\*\*

手机系统蓝牙状态

\*/

@property (nonatomic,assign) CBManagerState bleState;

###### 10.2 connectState

代表SDK与血压计设备的蓝牙连接状态。

/\*\*

设备连接状态

\*/

@property (nonatomic,assign) BTBloodPBLEManagerState connectState;

###### 10.3 selectDeviceIdt

当前需要连接的血压计设备的UUID。

在开始扫描周边设备后，APP可通过-BTBloodPBLEManager: didDiscoverDevice:;回调获取到当前扫描到的血压计设备的UUID，此时可以将想要连接的血压计的UUID设置为selectDeviceIdt，BTBloodPBLEManager便会去连接此设备。

详细调用流程见本文【SDK使用流程】中2.2部分。

# 三、SDK使用流程

## 3.1 初始化BTBloodPBLEManager

实例化一个BTBloodPBLEManager管理器，遵循BTBloodPBLEManagerDelegate协议，并成为其delegate。

示例：

self.bleManager = [BTBloodPBLEManager shareBTBloodPBLEManager];

self.bleManager.delegate = self;

## 3.2 扫描、连接血压计

SDK需要通过手机系统蓝牙，扫描周边的血压计设备，从而建立蓝牙连接。

#### 3.2.1检查手机系统蓝牙

在扫描周边设备之前，需要先通过bleState字段，检查手机系统蓝牙当前是否打开，如果手机蓝牙是开启状态，那么调用- (void)startScan;方法开始扫描。

示例：

// 如果手机蓝牙已开启

if (self.bleManager.bleState == CBManagerStatePoweredOn) {

// 开始扫描

[self.bleManager startScan];

}

#### 3.2.2 监听手机蓝牙是否开启

如果手机蓝牙未打开，APP需提醒用户打开，然后通过SDK的- BTBloodPBLEManagerDidUpdateState:回调，监听手机蓝牙是否开启。

如果用户打开了手机蓝牙，此时APP可调用- (void)startScan;方法开始扫描。

示例：

- (void)BTBloodPBLEManagerDidUpdateState:(CBManagerState)state

{

if (state == CBManagerStatePoweredOn) {

[self.bleManager startScan];

}else{

[SVProgressHUD showInfoWithStatus:@"Please turn on your iPhone's bluetooth"];

}

}

#### 3.2.3 连接设备

当SDK扫描到Belter\_BT血压计后，会通过- BTBloodPBLEManager: didDiscoverDevice:回调通知调用者。

其中**deviceUUID**为扫描到的血压计的UUID，如果APP想要连接此设备，则将deviceUUID赋值给BTBloodPBLEManager的selectDeviceIdt字段即可，BTBloodPBLEManager会自动去连接此设备。

示例：

/\*\*

扫描到设备

@param deviceUUID 设备UUID，连接用

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didDiscoverDeviceName:(NSString \*)name macAddress:(NSData \*)mac uuid:(NSUUID \*)deviceUUID

{

self.bleManager.selectDeviceIdt = deviceUUID;

}

## 3.3 监听连接状态

#### 3.3.1 连接成功回调：

参数说明：

serialNumber：当前已连接设备的序列号

- (void)BTBloodPBLEManagerDidConnectDevice:(BTBloodPBLEManager \*)manager serialNumber:(NSString \*)serialNumber

{

self.navigationItem.title = @"Connected";

NSLog(@"Connected! Device serialNumber = %@",serialNumber);

}

#### 3.3.2 连接失败回调：

- (void)BTBloodPBLEManagerDidFailToConnectDevice:(BTBloodPBLEManager \*)manager

{

self.navigationItem.title = @"Disconnected";

}

#### 3.3.3 血压计设备断开连接回调：

当血压计与手机蓝牙的连接断开时，APP如果想继续连接，可以重新开始扫描周边设备。

- (void)BTBloodPBLEManagerDidDisconnectDevice:(BTBloodPBLEManager \*)manager

{

self.navigationItem.title = @"Disconnected";

// If the device is disconnected, you'd better restart scanning.

[self.bleManager startScan];

}

## 3.4 监听测量数据

血压计测量数据分为三种类型：实时传送的动态收缩压、最终测量结果、测量出错结果。

#### 3.4.1 实时传送的动态收缩压

测量中，血压计会实时上传动态数据，因此SDK会一直调用该回调方法。

/\*\*

收到当前测量中的动态收缩压

@param systolicBP 收缩压

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedActiveSystolicBP:(NSInteger)systolicBP;

#### 3.4.2 最终测量结果

测量结束后，完整的测量结果包含收缩压、舒张压、脉搏与血压单位。

需要注意的是，无论当前单位是mmHg或kPa，血压数据均以mmHg传输，APP需要根据单位，对测量结果数据自行转换。

/\*\*

收到血压测量结果

@param systolicBP 收缩压

@param diastolicBP 舒张压

@param pulse 脉搏

@param unit 血压单位

@param isArrhythmia 是否心律不齐

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedMeasureResultSystolicBP:(NSInteger)systolicBP diastolicBP:(NSInteger)diastolicBP pulse:(NSInteger)pulse unit:(BTBloodPressureUnit)unit isArrhythmia:(BOOL)isArrhythmia;

#### 3.4.3 测量出错结果

当测量出现异常时，SDK通过该回调通知调用者。

示例：

/\*\*

收到测量出错信息

@param error 错误码，见 BTBloodPressureError 声明

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedMeasureFailed:(BTBloodPressureError)error;

其中BTBloodPressureError为测量错误枚举类，定义如下：

typedef enum : NSUInteger {

BTBloodPressureErrorDefault = 1, // 测量结果异常

BTBloodPressureError01, // 检测不到结果

BTBloodPressureError02, // 干扰太大

BTBloodPressureError03, // 充气异常，打不上气或者压力超过上限

BTBloodPressureErrorOther, // 其他错误

BTBloodPressureErrorLowPower, // 电池电量过低，提示用户更换电池

} BTBloodPressureError;

## 3.5 切换血压计单位

1. SDK提供接口下发切换血压计的显示单位，接口见本文接口说明第5点。

需要注意的是，血压计在测量状态时，不会响应切换单位，因此如果需要更改血压计的单位，推荐在测量结束后调用- sendToSetUnit:;接口设置。

示例：

/\*\*

收到血压测量结果

@param systolicBP 收缩压

@param diastolicBP 舒张压

@param pulse 脉搏

@param unit 血压单位

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedMeasureResultSystolicBP:(NSInteger)systolicBP diastolicBP:(NSInteger)diastolicBP pulse:(NSInteger)pulse unit:(BTBloodPressureUnit)unit

{

// 如果需要切换单位，请在测量结束后下发

[manager sendToSetUnit:BTBloodPressureUnitKPa];

}

1. 调用切换血压计单位接口后，SDK在收到血压计的响应指令（ACK）后，会通过代理方法 - (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedSetUnitACK:(BTBloodPressureUnit)unit;通知调用者。

示例：

/\*

收到下发设置单位ACK

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedSetUnitACK:(BTBloodPressureUnit)unit{

NSLog(@"Did received Set Unit ACK: %@",unit ? @"kpa":@"mmhg");

}

## 设备息屏

SDK提供接口下发息灭屏幕。

示例：

/\*\*

下发息灭屏幕

\*/

- (void)sendToTurnOffScreen;

## 同步时间

SDK提供接口下发同步时间。

注：需要在代理协议中 设备请求同步时间的方法内进行同步时间的接口调用。

示例：

/\*\*

设备请求同步时间

\*/

- (void)BTBloodPBLEManagerDidReceivedSyncTime:(BTBloodPBLEManager \*)manager

{

NSLog(@"%s ---",\_\_func\_\_);

[manager sendToSyncTime:[NSDate date]];

}

## 获取离线记录

SDK提供接口下发获取离线记录。

注：需要在测量结束后下发。

示例：

[self.bleManager sendToGetHistoryList];

## 收到离线记录并解析

SDK提供代理方法获取收到设备上发的离线记录。

示例：

/\*\*

收到历史记录

@param systolicBP 收缩压

@param diastolicBP 舒张压

@param pulse 脉搏

@param isArrhythmia 是否心律不齐

@param unit 血压单位

@param timeString 测量时间

@param index 当前第几条历史记录 note:start from 1

@param totalCount 历史记录总条数

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedHistoryMeasureResultSystolicBP:(NSInteger)systolicBP diastolicBP:(NSInteger)diastolicBP pulse:(NSInteger)pulse isArrhythmia:(BOOL)isArrhythmia unit:(BTBloodPressureUnit)unit time:(NSString \*)timeString index:(NSInteger)index totalCount:(NSInteger)totalCount

{

NSLog(@"did received history: --- index=%ld, count=%ld, sys=%ld, dia=%ld, pul=%ld, isArrhythmia=%d, time=%@,",index, totalCount,systolicBP,diastolicBP,pulse,isArrhythmia,timeString);

}

## 3.10 获取设备电量

1. SDK提供接口获取血压计当前的电池电量。

示例：

- (IBAction)getDevicePowerClick:(id)sender {

[self.bleManager sendToGetDevicePower];

}

1. SDK在收到血压计回发的电量后，会通过代理方法- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedDevicePower:(int)power;通知监听者。

示例：

/\*\*

收到设备电量

@param power 电量

\*/

- (void)BTBloodPBLEManager:(BTBloodPBLEManager \*)manager didReceivedDevicePower:(int)power{

NSLog(@"Did received Device Power: %d%%",power);

}

## 指定要连接的设备蓝牙名

SDK提供接口指定需要扫描、连接的设备蓝牙名。详细介绍见本文2.7.

示例：

self.bleManager = [BTBloodPBLEManager shareBTBloodPBLEManager];

self.bleManager.delegate = self;

[self.bleManager setDeviceBleName:@"Belter"];

## 设备熄屏（仅适用于指定型号13B4-LT）

1. SDK提供接口下发关闭设备的显示屏。

此接口仅用于型号13B4-LT设备，在血压计测量过程中调用此接口亦有效。

示例：

- (IBAction)turnOffScreenClick:(id)sender {

[self.bleManager sendToTurnOffScreen\_13B4LT];

}

1. SDK在收到血压计的熄屏响应指令（ACK）后，会通过代理方法- (void) BTBloodPBLEManagerDidReceivedTurnOffScreenACK\_13B4LT:(BTBloodPBLEManager \*)manager; 通知调用者。

示例：

/\*\*

收到熄灭屏幕指令ACK（13B4-LT专用）

\*/

- (void)BTBloodPBLEManagerDidReceivedTurnOffScreenACK\_13B4LT:(BTBloodPBLEManager \*)manager{

NSLog(@"Did received Turn Off Screen ACK（13B4-LT专用）");

}

## 设备亮屏（仅适用于指定型号13B4-LT）

1. SDK提供接口下发唤醒设备的显示屏。

此接口仅用于型号13B4-LT设备，在血压计测量过程中调用此接口亦有效。

示例：

- (IBAction)turnOnScreenClick:(id)sender {

[self.bleManager sendToTurnOnScreen\_13B4LT];

}

1. SDK在收到血压计的亮屏响应指令（ACK）后，会通过代理方法- (void) BTBloodPBLEManagerDidReceivedTurnOnScreenACK\_13B4LT:(BTBloodPBLEManager \*)manager; 通知调用者。

示例：

/\*\*

收到唤醒屏幕指令ACK（13B4-LT专用）

\*/

- (void)BTBloodPBLEManagerDidReceivedTurnOnScreenACK\_13B4LT:(BTBloodPBLEManager \*)manager{

NSLog(@"Did received Turn On Screen ACK（13B4-LT专用）");

}