1.  集成

## 1.1  下载

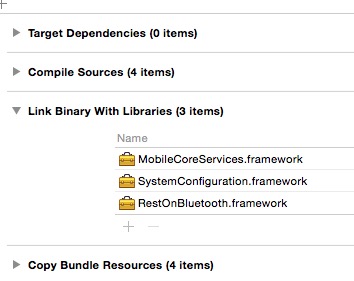
下载MilkyBluetooth.zip并解压缩

## 1.2 导入SDK

1）解压缩后得到三个目录，分别为doc，sdk，demo。

2）将目录sdk里的MilkyBluetooth.framework拖拽到你的工程

3）在Link Binary With Libraries添加 MobileCoreService.framework, SystemConfiguration.framework。



# **2.  实现基本功能**

## 2.1  头文件说明

**MilkyHistoryDataModel.h ： 历史数据类,枕扣生成的报告.**

#import <Foundation/Foundation.h>

@interface MilkyHistoryDataModel : NSObject

//起始时间日期"yyyy-MM-dd"

@property(nonatomic,strong)NSString \*date;

//时区

@property(nonatomic,strong)NSNumber \*timezone;

//睡眠总时长,实际睡觉总时长,除掉入睡时间和清醒时间。

@property(nonatomic,strong)NSNumber \*duration;

//上床时间的Unix时间戳(timeIntervalSince1970)

@property(nonatomic,strong)NSNumber \*startTime;

//睡眠得分

@property(nonatomic,strong)NSNumber \*score;

//浅睡眠时长(分钟)

@property(nonatomic,strong)NSNumber \*light;

//中睡眠时长(分钟)

@property(nonatomic,strong)NSNumber \*rem;

//深睡眠时长(分钟)

@property(nonatomic,strong)NSNumber \*deep;

//清醒时长(分钟)

@property(nonatomic,strong)NSNumber \*wake;

//深睡眠百分比

@property(nonatomic,strong)NSNumber \*MdDeepSleepPerc;

//中睡百分比

@property(nonatomic,strong)NSNumber \*MdRemSleepPerc;

//浅睡百分比

@property(nonatomic,strong)NSNumber \*MdLightSleepPerc;

//清醒百分比

@property(nonatomic,strong)NSNumber \*MdWakeSleepPerc;

//起床前的那段清醒时间(分钟)

@property (nonatomic,strong)NSNumber \*MdWakeUpTime;

//清醒次数

@property(nonatomic,strong)NSNumber \*wake\_times;

//时间步(60)保留,暂无用

@property(nonatomic,strong)NSNumber \*timeStep;

//总记录数(监测了多少分钟,一分钟一个点)

@property(nonatomic,strong)NSNumber \*recordCount;

/\*结束方式

\* 0: 正常结束

\* 1: 自动结束

\* 2: 强制结束(如:关机)

\* 3: 错误结束(如:供电不足、系统异常)

\*/

@property(nonatomic,strong)NSNumber \*stopMode;

//各个状态持续时间,(秒)

@property(nonatomic,strong)NSArray \*statusValueArray;

/\*sleepStateArray 睡眠等级数据，说明：

0 清醒

1 浅睡

2 中睡

3 深睡

\*/

@property(nonatomic,strong)NSArray \*sleepStateArray;

//体动强度（float类型,归一化体动强度，输出范围:0-255）

@property(nonatomic,strong)NSArray \*motionIntensityArray;

//入睡所花时间

@property(nonatomic,strong)NSNumber \*asleepTime;

//睡眠效率（有效睡眠时长所占的百分比）

@property (nonatomic,strong)NSNumber \*SleepEfficient;

//体动等级(-2:极少:监测失败；-1:过少:信息丢失；+0:正常；+1:过多:躁动不安；+2:极多:严重躁动不安)

@property (nonatomic,strong)NSNumber \*bodyMoveLevel;

//算法版本号

@property(nonatomic,strong)NSString \*arithmeticVer;

//睡眠状态曲线（float类型,输出范围:0.0-3.0）

@property (nonatomic,strong)NSArray \*SleepCurveArray;

/\*sleepCurveStatusArray 睡眠曲线中的各类状态，值说明：

0 一切正常

1 初始化状态

2 呼吸暂停(枕扣无该项)

3 心跳暂停(枕扣无该项)

4 体动

5 离床(枕扣无该项)

6 翻身(枕扣无该项)

\*/

@property (nonatomic,strong)NSArray \*sleepCurveStatusArray;

//数据异常标记(数据有效性标识，short类型,0-有效；bit1-体动过弱；bit2-体动过少)

@property (nonatomic,strong)NSNumber \*flaginvalid;

//睡眠时间过长扣分

@property (nonatomic,strong)NSString \*md\_sleep\_time\_decrease\_scale\_long;

//睡眠时间过短扣分

@property (nonatomic,strong)NSString \*md\_sleep\_time\_decrease\_scale\_short;

//清醒次数扣分

@property (nonatomic,strong)NSString \*md\_wake\_cnt\_decrease\_scale;

//体动扣分,也叫躁动不安扣分

@property (nonatomic,strong)NSString \*md\_body\_move\_decrease\_scale;

//深睡眠扣分

@property (nonatomic,strong)NSString \*md\_perc\_deep\_decrease\_scale;

//难以入睡扣分

@property (nonatomic,strong)NSString \*md\_fall\_asleep\_time\_decrease\_scale;

//睡觉时间扣分(太晚睡)

@property (nonatomic,strong)NSString \*md\_start\_time\_decrease\_scale;

//良性睡眠扣分,(中睡/深睡 占入睡后时间百分比)

@property (nonatomic,strong)NSString \*benignSleepLow;

//睡眠效率扣分

@property (nonatomic,strong)NSString \*sleepaceEfficientLow;

//体动过少扣分,与躁动不安对应

@property (nonatomic,strong)NSString \*bodyMoveLow;

//原始数据,状态0;

@property(nonatomic,strong)NSArray \*originFeatureValue0Array;

//原始数据,状态1;

@property(nonatomic,strong)NSArray \*originFeatureValue1Array;

//原始数据-tag

@property(nonatomic,strong)NSArray \*originTagArray;

@end

**BLEHelper.h ：提供设备的蓝牙通讯的接口,**

@property(nonatomic,readonly)CBPeripheral \*peripheral;//蓝牙服务

@property(nonatomic,readonly)CBCharacteristic \*readCharactertic;//特征

/\*

share： 第一次调用时务必要调initWithPeripheral:readCharactertic:接口，用以初始化蓝牙的参数

参数：

peripheral 蓝牙服务

readCharactertic 蓝牙读特征

\*/

+(BLEHelper \*)share;

//与share相对，清空成员变量

+(void)deshare;

/\*

initWithPeripheral：初始化， 第一次调用share接口时务必要调用本接口，用以初始化蓝牙的参数

参数：

peripheral 蓝牙服务

readCharactertic 蓝牙读特征

\*/

- (void) initWithPeripheral:(CBPeripheral \*)peripheral readCharactertic:(CBCharacteristic \*)readCharactertic;

**BleManager.h: 扫描连接蓝牙,接收蓝牙通信数据**

//是否已连接设备

- (BOOL)isConnected;

//扫描设备

-(void)scanfPeripheralWithWithSuccess:(void(^)(NSString \* ID, NSString \*name, CBPeripheral \*peripheral))success;

//连接设备

-(void)bleConnectWithPeripheral:(CBPeripheral \*)peripheral

success:(void (^)(void))success

failure:(void (^)(NSString \*error))failure;

//清除连接请求的block

- (void)clearConnectRequestHandle;

/\*\*

\* 主动断开蓝牙连接,不触发重连操作

\*/

-(void)bleDisconnectPeripheralWithsuccess:(void (^)(void))success;

**NewBleServiceInterface.h: 枕扣所有功能接口**

#import <Foundation/Foundation.h>

#import <UIKit/UIKit.h>

#import "BLEHelper.h"

@class BTHNetworkAssistHandle;

@class SLPDownloadHistoryDataAssist;

@class SLPBHTRequestHelper;

@class NewBlePackage;

@class NewBleRequest;

@class BLEHelper;

@interface NewBleServiceInterface : NSObject

/\*

\*时间校准

@timstamp 时间戳

@timezone 时区偏移

@specialTime 特殊时令

@specialTimeOffSet 特殊时令偏移

\*/

+(void)checkDeviceTimeWithTimestamp:(UInt32)timstamp timeZoneOffSet:(int32\_t)timezone specialTime:(uint)specialTime specialTimeOffSet:(int32\_t)specialTimeOffSet;

/\*\*

\* 枕扣时间校准快捷入口

\*/

+(void)milkyTimeCalibration;

/\*\*

\* 获取枕扣采集状态

\*

\* @param success 成功返回 status: 1：正在采集状态,0：未处于采集状态

\* @param failure 请求失败

\*/

+ (void)queryMilkyCollectStatusWithSuccess:(void (^)(NSInteger status))success

failure:(void (^)(void))failure;

/\*\*

\* 获取枕扣ID

\*

\* @param success 成功

\* @param fail 失败

\*/

+(void)getMilkyIdWithSuccess:(void(^)(NSString \*milkyID))success fail:(void (^)())fail;

/\*

\*系统版本信息

\*/

+(void)newBleGetDeviceVersionWithSuccess:(void (^)(NSString \*version))success

failure:(void (^)(void))failure;

/\*

\*电池信息

\*/

+(void)newBleGetDevicePowerStatusSuccess:(iBlock)success

failure:(void (^)(void))failure;

/\*

\*睡眠监测配置

@hour 小时

@minutes 分

@times 时长

\*/

+(void)setMilkySleepMonitorWithHour:(uint)hour mintues:(uint)minutes duration:(uint)times WithSuccess:(void (^)(id responseObject))success fail:(void (^)())fail;

/\*\*

\* 设置闹铃信息到枕扣

\*

\* @param valid 闹铃是否可用

\* @param offset 智能闹钟偏移(分钟)

\* @param hour 时

\* @param min 分

\* @param weekArr 重复日

\*/

+ (void)setAlarmInfoToMilkyWithValid:(int)valid andOffset:(int)offset andHour:(int)hour andMinutes:(int)min andWeekArr:(NSArray \*)weekArr andSuccess:(void (^)(id responseObject))success fail:(void (^)())fail;

/\*\*

\* 将当前用户id设置到枕扣,用于标记枕扣数据归属哪个用户

\*/

+ (void)setUserIdToMilky:(int)userId WithSuccess:(void (^)(id responseObject))success fail:(void (^)())fail;

/\*\*

\* 睡眠状态信息

\*

\* @param success 获取成功

\* ^(int isAsleep, int isAwake) (isAsleep 0:未入睡 1:入睡 其他:无效) (isAwake 0:未清醒 1:清醒 其他:无效)

\* @param fail 失败

\*/

+(void)getSleepingStatusInfoWithSuccess:(milkySleepingStatusInfoBlock)success fail:(void (^)())fail;

/\*

\*采集状态查询

\*/

+(void)checkStateCollection;

/\*\*

\* 更改采集状态操作

\*

\* @param startOrStop 0x00:结束采集(已结束，则忽略该命令) 0x01:开始采集(已开始, 则忽略该命令) 其他：无效

\* @param successBlock 成功回调

\*/

+(void)changedSleepStatus:(int)startOrStop withSuccess:(changedMilkySleepStatusSuccessBlock)successBlock;

/\*

\*历史数据下载

@startTime 起始时间

@startIndex 起始位置

@count 下载数目

\*/

+ (void)downloadHistoryDataFromTime:(UInt32)startTime

endTime:(UInt32)endTime

failed:(SLPBHTRequestFailed)failed

querySucceed:(SLPBHTHistoryDataQuerySucceed)querySucceed

dataRecieved:(SLPBHTHistoryDataRecieved)dataRecieved

completion:(SLPBHTRequestCompletion)completion;

## 3.  枕扣连接的流程

1）敲击枕扣,灯亮起时表示枕扣蓝牙已开启

2）手机端搜索并连接枕扣.

3）长时间未操作,枕扣蓝牙会断开,需重复上述步骤才能再次调用功能接口