

iOS RobotPenManagerSDK

参考手册

简介 3

笔服务RobotPenManager 3

1.RobotPenManagerSDK 下载 3

2．iOS RobotPenManagerSDK 目录结构 4

3. 配置SDK环境 4

4.应用 5

扫描、连接设备 5

OTA 升级 6

同步笔记 6

5.相关API 7

6、笔服务状态相关 10

# 简介

iOS RobotPenManagerSDK 封装了智能笔和电磁板相关的API，应用只需准守协议，实现相应的代理方法即可获取到电磁本、电磁笔及电磁本笔记的相关数据。

注：iOS RobotPenManagerSDK需在iOS8.0及以上系统运行。

功能介绍：

### 笔服务RobotPenManager

笔服务功能：搜索、连接、配对、断开设备，获取设备基本信息，实现与设备通讯。

详情查看RobotPenManager.h文件

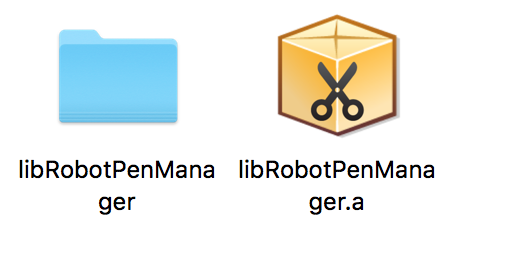
# 1.RobotPenManagerSDK 下载

请到<https://github.com/PPWrite/SDK>页面下载最新版本的iOS RobotPenManagerSDK。

# 2．iOS RobotPenManagerSDK 目录结构

1. libRobotPenManager.a封装了iOS RobotPenManagerSDK功能实现。

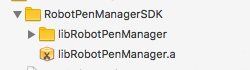
2. libRobotPenManager 文件包含了相关的头文件。



# 3. 配置SDK环境

1. 将iOS SDK中的 libRobotPenManager.a和libRobotPenManager文件拷贝到应用开发的目录下。

2.将libRobotPenManager.a和libRobotPenManager文件添加到工程中。



3. 添加SDK依赖的系统库文件。

本SDK的依赖库只需添加libsqlite3.tdb（libsqlite3. dylib）或libsqlite3.0.tdb（libsqlite3.0. dylib ）即可。

# 4.应用

1.遵守RobotPenDelegate代理

[[RobotPenManager sharePenManager]

setPenDelegate:self];

2.实现RobotPenDelegate中的代理方法即可获取相应的数据信息。

详见RobotPenManagerDemo。

## 扫描、连接设备

扫描设备scanDevice，发现设备则会通过getBufferDevice（）代理方法返回设备模型。

1. 连接设备，调用connectDevice方法连接设备，并通过代理方法getDeviceState（）获取设备连接的状态。

代码实现：

//搜索设备

[[RobotPenManager sharePenManager] scanDevice];

//发现设备 回调方法

- (void)getBufferDevice:(RobotPenDevice \*)device

//连接设备

[[RobotPenManager sharePenManager] connectDevice:RobotPenDevice];

//获取设备状态(设备连接状态):未链接、已链接、设备信息等详情查看DeviceState。

- (void)getDeviceState:(DeviceState)State

## OTA 升级­­­­

当在getDeviceState接收到DEVICE\_UPDATE\_CAN状态（设备可更新状态）时，即表示设备可升级，方式如下：

代码实现：

 //OTA升级 过程中不要进行其他操作。 升级前确保已设置代理

[[RobotPenManager sharePenManager] startOTA];

//OTA 升级状态 具体状态说明：请查看OTAState

- (void)OTAUpdateState:(OTAState)state

//OTA升级进度

- (void)OTAUpdateProgress:(float)progress

## 同步笔记

1. 通过协议方法getStorageNum（）获取笔记数量，num大于0表示有离线笔记。
2. 调用startSyncNote方法可以开始同步笔记。
3. 通过协议方法SyncState（）获取同步状态。通过协议方法getStorageNum（）获取笔记数量。
4. 调用stopSyncNote方法，可以停止同步。

代码实现：

//获取笔记数量 和 电量

- (void)getStorageNum:(int)num andBattery:(int)battery;

//开始同步笔记

[[RobotPenManager sharePenManager] startSyncNote];

// 同步状态。 详细同步状态查看 SYNCState

- (void)SyncState:(SYNCState)state

//停止同步笔记

[[RobotPenManager sharePenManager] stopSyncNote];

# 5.相关API

RobotPenDelegate相关代理 具体API接口信息 查看RobotPenManager.h

|  |  |  |
| --- | --- | --- |
| 方法名 | 功能 | 备注 |
| -(void)getDeviceState:  (DeviceState)State; | 获取设备状态 |  |
| - (void)getPointInfo:  (PenPoint \*)point; | 获取点数据 |  |
| - (void)getDeviceVersion:  (NSDictionary \*)infos; | 获取设备信息 |  |
| - (void)AutoConnectingDevice:  (PenDevice \*)device | 自动连接的中的设备 |  |
| - (void)OTAUpdateState:(OTAState)state; | OTA升级状态 |  |
| - (void)OTAUpdateProgress:  (float)progess; | OTA升级进度 |  |
| - (void)SyncState:(SYNCState)state; | 同步笔记的状态 |  |
| - (void)getSyncData:(Trails \*)trails; | 获取离线笔记轨迹 |  |
| - (void)getSyncNote:(Note \*)note; | 获取离线笔记 |  |
| - (void)getStorageNum:(int)num; | 获取离线笔记数量 |  |
| - (void)getBufferDevice:(PenDevice \*)device; | 获取蓝牙发现的设备 |  |
| - (void)getDeviceEvent:(DeviceEventType)Type; | 获取电磁本点击事件 |  |

RobotPenManager相关方法

|  |  |  |
| --- | --- | --- |
| 方法名 | 功能 | 备注 |
| + (RobotPenManager \*)sharePenManager; | 单例初始化 |  |
| - (BOOL)checkDeviceConnect; | 检查设备是否连接 |  |
| - (PenDevice \*)getConnectDevice; | 获取当前连接的设备 |  |
| - (void)connectDevice:(PenDevice \*)penDevice :(id<RobotPenDelegate>)delegate; | 连接设备 |  |
| - (void)scanDevice:  (id<RobotPenDelegate>)delegate; | 扫描设备 |  |
| - (void)disconnectDevice; | 断开连接设备 |  |
| - (NSDictionary \*)getDeviceVersion; | 获取版本号 |  |
| - (DeviceType)getConnectDeviceType; | 获取设备类型 |  |
| - (void)setPenDelegate:  (id<RobotPenDelegate>)delegate; | 设置回调代理 |  |
| -(void)stopScanDevice; | 停止扫描 |  |
| - (BOOL)checkDeviceConnect; | 检查设备是否连接 |  |
| - (BOOL)checkIsHaveMatch; | //检查是否有配对过的设备 |  |
| - (void)AutoCheckDeviceConnect; | //设备自动链接 |  |
| - (void)deleteConnect; | 取消配对 |  |
| -(void)startOTA:  (id<RobotPenDelegate>)delegate; | 开始OTA升级 |  |
| - (void)startSyncNote:  (id<RobotPenDelegate>)delegate; | 开始同步笔记 |  |
| -(void)stopSyncNote; | 停止同步笔记 |  |
| - (void)SendPage:  (int)Current :(int)Totla; | 发送页码信息 |  |

# 6、笔服务状态相关

//连接状态

typedef enum {

    /\*\*没有找到设备\*\*/

    NOTHING,

    /\*\*正在连接\*\*/

    CONNECTING,

    /\*\*连接成功\*\*/

    CONNECTED,

    /\*\*连接错误\*\*/

    CONNECT\_FAIL,

    /\*\*正在断开\*\*/

    DISCONNECTING,

    /\*\*已断开\*\*/

    DISCONNECTED,

    /\*\*开始发现服务\*\*/

    SERVICES\_START,

    /\*\*服务准备完成\*\*/

    SERVICES\_READY,

    /\*\*发现服务失败\*\*/

    SERVICES\_FAIL,

    /\*\*笔准备完成\*\*/

    PEN\_READY,

    /\*\*笔初始化完成\*\*/

    PEN\_INIT\_COMPLETE,

    /\*\*设备信息获取\*\*/

    DEVICE\_INFO\_END,

    /\*\*检查设备更新\*\*/

    DEVICE\_UPDATE

}DeviceState;

//同步笔记状态

typedef enum {

    SYNC\_ERROR,

    SYNC\_NOTE,

    SYNC\_NO\_NOTE,

    SYNC\_SUCCESS,

    SYNC\_START,

    SYNC\_STOP,

    SYNC\_COMPLETE,

}SYNCState;

//OTA状态

typedef enum {

OTA\_ERROR,

OTA\_DATA,

OTA\_UPDATE,

OTA\_SUCCESS,

OTA\_RESET,

OTA\_ONE\_SUCCESS,//一个升级成功

}OTAState;

//SENSOR状态

typedef enum {

/\*\* 模组升级错误\*/

SENSOR\_ERROR,

/\*\* 进入模组升级模式\*/

SENSOR\_REDAY,

/\*\* 进入模组升级数据\*/

SENSOR\_DATA,

/\*\* 模组正在升级\*/

SENSOR\_UPDATE,

/\*\* 模组升级成功\*/

SENSOR\_SUCCESS,

/\*\* 模组升级失败\*/

SENSOR\_FAIL,

}SensorState;