

ROBOTPEN 鲁伯特

iOS RobotBoardSDK

参考手册

| | |
|-------------------------------|----|
| 1.RobotBoardSDK 下载 | 3 |
| 2.iOS RobotBoardSDK 目录结构..... | 4 |
| 3.配置 SDK 环境 | 4 |
| 4.应用..... | 5 |
| 1、初始化白板..... | 5 |
| 2、设置白板 | 5 |
| 3、自定义白板..... | 7 |
| 4、笔迹设置 | 8 |
| 5、插图、分页功能 | 10 |

简介：

iOS RobotBoardSDK 集成了笔服务功能，并将数据展示到应用中。

白板服务 SDK 功能：

1) 画板自定义

- 1、 自定义画板大小。
- 2、 自定义画板方向。
- 3、 自定义画布背景颜色和背景图。
- 4、 自定义画笔图标。

2) 笔迹功能

- 1、 实时展示电磁设备笔迹及手绘。
- 2、 自定义笔迹颜色、宽度。
- 3、 优化笔迹
- 4、 擦除、清除笔迹。

3) 插图、分页功能

- 1、 插入、调整、删除笔记插图。
- 2、 按顺序新建、插入、删除分页。
- 3、 新建笔记、删除笔记、笔记重命名。
- 4、 刷新白板数据。

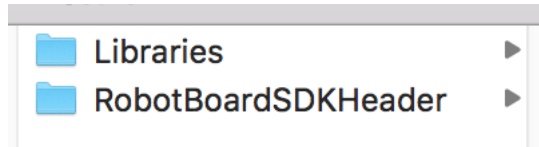
1. RobotBoardSDK 下载

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

RobotBoardSDK。

2. iOS RobotBoardSDK 目录结构

1. RobotBoardSDK.a 封装了 iOS RobotBoardSDK 功能实现。
2. RobotBoardSDK 文件包含了相关的头文件。



3. 配置 SDK 环境

一、手动导入

1. 将 iOS SDK 中的 RobotBoardSDKSDKHeader 和 Libraries 文件拷贝到应用开发的目录下。
2. 将 RobotBoardSDKSDKHeader 和 Libraries 文件添加到工程中。
3. 添加 SDK 依赖的系统库文件。本 SDK 的依赖库需添加 libz. tdb (libz. dylib)、UIKit.framework、Foundation.framework。

二、Cocoapods 导入

```
pod 'RobotPenSDK/RobotBoardSDK'
```

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

请先参考 iOS RobotPenSDK 参考手册。

4. 应用

1、初始化白板

```
//此处不要设置大小（在 setWB 方法里设置）  
  
self.WhiteBoardView = [[RobotWhiteBoardView alloc] init];  
  
//设置画板代理，必须实现  
  
self.WhiteBoardView.whiteBoardDelegate = self;
```

2、设置白板

```
//设置设备类型（必须设置）  
  
[self.WhiteBoardView setDeviceType: DeviceTypes];  
  
//设置画板的横竖方向 0：竖 1：横（必须设置）  
  
[self.WhiteBoardView setIsHorizontal:isHorizontal];  
  
//设置白板的 Frame（根据传入的 frame 自动设置, 必须设置）  
  
[self.WhiteBoardView setSceneSize:frame];
```

至此即可完成对白板的基础设置，即可用手绘制轨迹。

实现代理方法 GetWhiteBoardState 即可监听白板状态。

```
- (void)GetWhiteBoardState:(WBMessageType)State  
content:(NSString *)Info{ }
```

注：

1、 此时默认为手输入模式，只能用于手绘。

- 2、 setDeviceType 和 setIsHorizontal 必须在 setSceneSize 方法之前实现。
- 3、 setDeviceType 方法需要赋值一个非 0（UNKNOWN）的默认值。
- 4、 setIsHorizontal 方法设置白板的方向，已电磁板方向为准，电磁板短边在上下，宽边在左右为竖，横向为竖向顺时针旋转 90 度。
- 5、 竖向、横向坐标原点在左上角，横向坐标原点即为竖向的左下角。
- 6、 setSceneSize 方法设置白板大小，并非是设置的大小。是根据设置区域大小及设备类型（setDeviceType 的设置）的书写区域比例动态计算得出并居中。
- 7、 GetWhiteBoardState 中多张图片操作只会返回操作失败，不会返回具体第几张，失败的图片自动过滤。

笔输入接入流程：

- 1、 调用 SetDrawType 方法设置输入方式。
- 2、 实现协议方法 getPointInfo 获取实时数据。
- 3、 调用划线方法 drawLine 即可画线。

// 输入的方式（0:手、1:笔）

```
[self.WhiteBoardView SetDrawType:PenDraw];
```

```
-(void)getPointInfo:(RobotPenPoint *)point{
```

```
    //画板画线
```

```
    [self.WhiteBoardView drawLine:point]; }
```

3、自定义白板

1、设置白板背景颜色和背景图

1、调用设置画板背景颜色方法 `setBgColor` 即可设置背景颜色。

// 设置 背景颜色

```
[self.WhiteBoardView setBackgroundColor:BG_NoteColor];
```

2、调用设置白板背景图方法 `setBgPhoto` 即可设置背景图片。

该设置方法为平铺设置。

2、设置笔图标

1、调用设置笔图标方法 `setPenIcon` 即可对比图标进行设置。

2、调用设置不可书写图标方法 `setPenModleBanIcon` 即可对比不可书写状态图标进行设置。

4、笔迹设置

1、自定义笔迹颜色、宽度

1、实现协议方法 `getPenColor` 即可设置笔迹颜色。

//获取笔颜色，需要设置笔迹颜色功能则需要实现

```
- (UIColor *)getPenColor{  
    return _PenColor;}  

```

2、实现协议方法 `getPenWeight` 即可设置笔迹宽度。

//获取笔的粗细，需要设置笔迹宽度功能则需要实现

```
- (CGFloat)getPenWeight{  
    return 1;}  

```

注：`getPenWeight` 设置在不优化下为笔迹宽度，在开启笔迹优化下为笔迹基础宽度，笔记的宽度会以此进行计算。

如果不设置，默认宽度为 1。

2、优化笔迹

实现协议方法 `getIsPressure` 即可设置优化笔迹。

//大于 0 表示是有压感，需要用到压感功能则需要实现

```
- (int)getIsPressure{  
    return 1;}  

```

注：优化笔迹只对笔输入轨迹有效，对手输入无效。笔输入建议打开。

3、擦除、清除笔迹。

1、实现协议方法 `getIsRubber` 即可设置为橡皮擦模式，可擦除笔记。

//大于 0 表示是橡皮擦，需要用到橡皮擦功能则需要实现

```
- (int) getIsRubber {  
  
    return 0;  
  
}
```

注：橡皮擦模式只能用手绘的方式擦除。

2、调用清除笔迹方法 `cleanTrail` 即可清除当前画布上的所有笔迹。

//清除所有笔迹

```
[self.WhiteBoardView cleanTrail];
```

注： `cleanTrail` 方法只会清除当前画布上的笔迹，如果是多页笔记，则只会清除当前笔记当前页上的笔迹数据，其他分页上的数据不受影响。

5、插图、分页功能

1、插入、调整、删除笔记插图。

1、调用插入图片方法 `insterPhotoWithPath` 可以将图片插入到白板中。

2、调用插入图片组方法 `insterPhotoWithUrls` 可以插入多张图片。

3、调用编辑图片方法 `setPhotoEdit` 可以进入图片编辑模式。

4、调用旋转图片方法 `currPhotoRotate90` 可以顺时针旋转图片，每次旋转 90 度。

5、调用设置图片模式方法 `setPhotoScaleType` 可以设置图片为填充或居中显示。

6、调用清除图片方法 `cleanPhoto` 可以删除当前页上的图片。

7、调用清空白板内容方法 `cleanScreen` 可以删除当前分页上的所有笔迹和图片。

注：1、iOS 白板服务每个分页的白板只支持一张图片。

2、`insterPhotoWithUrls` 插入 n 张图片时会自动再创建 $n-1$ 个分页，并将图片按顺序分别添加到相应分页中。

3、图片编辑的方法必须先进入图片编辑模式。

4、`cleanPhoto` 方法只会清除当前画布上的图片，如果是多页笔记，则只会清除当前笔记当前页上的笔迹图片，其他分页上的图片不受影响。

2、按顺序新建、插入、删除分页。

1、调用向下翻页方法 `nextPage` 可以向下翻页，如果有下一个分页则翻页，否则创建一个新的分页再翻页。

2、调用向上翻页方法 `frontPage` 可以向上翻页，如果有上一个分页则翻页，否则创建一个新的分页再翻页。

3、调用出入分页方法 `NewNextPage` 可以在当前页与下一页之间插入新的分页。

4、调用删除分页方法 `delCurrPage` 可以删除当前分页。

5、调用跳转指定页方法 `turnToPageWithPage` 可以跳转到目标分页。

注：`turnToPageWithPage` 方法的参数为显示的页码值。例如当前有1-10页，入参为6则跳到第六页。

分页的操作必须基于笔记上。

3、新建笔记、删除笔记、笔记重命名。

1、调用新建笔记方法 `BuildNote` 即可创建一本笔记。

2、调用删除笔记方法 `DeleteNoteWithNoteKey` 可删除笔记。

3、调用重命名方法 `ChangeNameWithNoteKey` 可对笔记重命名。

新建笔记代码实现：

```
//打开数据库

[RobotSqlManager checkRobotSqlManager];

//创建笔记的Model

RobotNote *notemodel = [[RobotNote alloc] init];

//笔记的NoteKey（唯一标识），必填

notemodel.NoteKey = @"WB";

//笔记的设备类型，必填

notemodel.DeviceType = DeviceTypes;

//笔记方向，必填

notemodel.IsHorizontal = 0;

//笔记的标题

notemodel.Title = @"白板";

// 创建一本笔记

[RobotSqlManager BuildNote:notemodel Success:^(id
responseObject) {

    } Failure:^(NSError *error) { }];
```

注：以上为数据库操作。

NoteKey 为笔记唯一标识，笔记相关操作的唯一标识，建议采用时间戳等确保唯一性。

NoteKey、DeviceType、IsHorizontal 为必填项，其他参数可视情况添加。

笔记创建完成可进行分页操作。

4、刷新白板数据

1、实现协议方法 `getNoteKey` 即可设置需要打开的笔记。

2、调用刷新笔记方法 `RefreshAll` 即可刷新笔记。

3、实现协议方法 `getNoteTitle` 即可显示笔记名称。

//获取当前笔记本Key，需要用到画板功能则需要实现

```
- (NSString *)getNoteKey{  
  
    return _NoteKey;  
  
}
```

//打开数据库

```
[RobotSqlManager checkRobotSqlManager];
```

//刷新白板内容（本地数据库操作,用不到数据库的可以不写）

```
[self.WhiteBoardView RefreshAll];
```

//设置笔记标题，需要设置笔迹标题功能则需要实现

```
- (NSString *)getNoteTitle {  
  
    return _NoteTitle;  
  
}
```

注：只要设置正确的 `NoteKey`（笔记唯一标识）即可执行刷新操作。刷新操作为数据库操作，同一页面数据库操作只需打开一次即可。

5、页码检测设备专用

关于页码检测信息获取请参考 RobotPenSDK 参考手册。

页码检测设备的笔记管理通过笔记唯一标识 NoteKey 管理。

分页的管理还可以通过页码 Page 或页码标识 Blockkey 来管理，建议用 Page 比较直观。

在笔记存在的情况下用管理页码方法 setManagePageWithPage 直接管理笔记分页即可。

若笔记不存在，则需先创建笔记在调用管理页码方法管理笔记。

示例代码：

```
- (void)getDevicePage:(int)page andNoteId:(int)NoteId
{ //管理页码信息
    [self.WhiteBoardView setManagePageWithPage:page];
}
```

页码识别的离线笔记流程：

- 1、 实现协议方法 getTASyncNote 获取离线笔记信息
- 2、 调用页码识别专用保存笔记方法 SaveTANote 保存笔记信息。
- 3、 调用页码识别专用设置轨迹 BlockKey 方法
SetBlockWithBlock 即可

详情可参考 RobotBoardDemo，页码识别效果可以参考 Demo 中的“录课白板”页面。