

# Yaokan SDK3 iOS 说明文档

文件编号：YAOKANSDK3IOS-20190723 版本：v1.4

深圳遥看科技有限公司（版权所有，切勿拷贝）

版本	说明	备注	日期
v1	新建	Dong	20190723
v1.1	增加API	Huang	20190801
v1.2	格式化	Dong	20190802
v1.3	增加空调匹配	Huang	20190814
v1.4	射频API说明	Huang	20191130

## 1. 概述

YaokanSDK3 提供完整的设备配网，设备管理，遥控器管理功能，开箱即用，快速与已有App对接的目的。

## 2. 文档阅读对象

使用 YaokanSDK3 iOS 版的开发者

## 3. 接入

需要先向商务申请 APP\_ID

### 3.1 使用pod

- 在 Podfile 添加 `pod 'YaokanSDK'`，如果没有此文件则新建一个再添加。
- 然后执行 `$ pod install --verbose`。
- 如果找不到 YaokanSDK 请先更新 Pod 仓库：`pod repo update`

### 3.2 手动集成

- 打开 `[your_project].xcodproj`，选择 Target `[your_target_name]` 打开 General 标签项。
- 在 Embedded Binaries 中点击 `+` 号，点击 Add Other.. 打开 YaokanSDK 目录选择 `YaokanSDK.framework` 和 `CocoaAsyncSocket.framework` `MQTTClient.framework` `SocketRocket.framework`。直接将这4个文件拖进 Embedded Binaries 一样以。

## 4. API 列表

### 4.1 初始化接口

- 注册SDK

```
[YaokanSDK registApp:YOUR_APP_ID secret:YOUR_APP_SEC completion:^(NSError * _Nonnull error) {  
    }];
```

### 4.2 设备接口

1. 配置入网 在 [your\_project].xcodeproj , 选择 Target [your\_target\_name] 打开 Capabilities 标签项 打开 Access WiFi Information 。 从 iOS 13起获取SSID之前需要定位权限。

```
[YaokanSDK bindYKCWithSSID:@"2.4G-WIFI-SSID" password:@"wifipassword" completion:^(NSError * _Nullable error,
YKDevice * _Nullable device) {

}};
```

2. 获取设备列表

```
[YaokanSDK fetchBoundYKC:^(NSArray<YKDevice *> * _Nonnull devices, NSError *error) {

}};
```

3. 导入设备列表

```
YaokanSDK importYKDevice:@"[{"did":"A64D184C35EAB091","rf":"1","mac":"BCDDC289CCE2","type":"YKK_1011-","version":"1.1.6"}, {"did":"23C81FD441D21AB8","rf":"","mac":"DC4F22529F13","type":"YKK_1011","version":"1.1.6"}]";
```

4. 设备测试

```
[YaokanSDK toggleLEDWithYKId:@"targetMacAddr"];
```

5. 发码(不适用于空调类型)

```
[YaokanSDK sendRemoteWithYkId:[YKCenterCommon sharedInstance] currentYKId]
remoteId:remoteId remoteDeviceTypeId:typeId cmdkey:@"按键名"
completion:^(BOOL result, NSError * _Nonnull error) {

}};
```

6. 空调发码(不适用于上下左右 扫风功能)

```
[YaokanSDK sendACWithYKId:[YKCenterCommon sharedInstance] currentYKId]
remoteDevice:remote
withMode:mode
temp:temp
speed:speed
windU:windU
windL:windL
completion:^(BOOL result, NSError * _Nonnull error) {

}};
```

7. 空调上下扫风发码

```
[YaokanSDK sendAcWindUDWithYKId:[YKCenterCommon sharedInstance] currentYKId]
remoteDevice:remote
withMode:mode
temp:temp
speed:speed
windU:windU
windL:windL
completion:^(BOOL result, NSError * _Nonnull error) {

}};
```

8. 空调左右扫风发码

```
[YaokanSDK sendAcWindUDWithYKId:[[YKCenterCommon sharedInstance] currentYKId]
    remoteDevice:remote
    withMode:mode
    temp:temp
    speed:speed
    windU:windU
    windL:windL
    completion:^(BOOL result, NSError * _Nonnull error) {

    }
    }];
```

#### 9. 学习红外

```
[YaokanSDK learnIRWithYKId:[[YKCenterCommon sharedInstance] currentYKId]
    remote:_remote key:key.key originRid:_remote.remoteId
    completion:^(NSString * _Nonnull ridNew, NSError * _Nonnull error) {
    }
    }];
```

#### 10. 学习射频(已创建遥控器 传 YKRemoteDevice 类型对象)

```
[YaokanSDK learnRFWithRemote:YKRemoteDevice key:@"yourLearnKey"
    completion:^(NSString * _Nonnull ridNew, NSError * _Nonnull error) {
    }
    }];
```

#### 11. 学习射频(对码阶段 传 YKRemoteMatchDevice 类型)

```
[YaokanSDK learnRFWithMatchRemote:YKRemoteMatchDevice key:@"yourLearnKey"
    completion:^(NSString * _Nonnull ridNew, NSError * _Nonnull error) {
    }
    }];
```

#### 12. 停止学习(已创建的遥控器 传 YKRemoteDevice 类型)

```
[YaokanSDK learnStopWithRemote:(YKRemoteDevice *)remote];
```

#### 13. 停止学习(匹配对码阶段 传 YKRemoteMatchDevice 类型)

```
[YaokanSDK learnStopWithMatchRemote:(YKRemoteMatchDevice *)matchRemote];
```

#### 14. 设备开灯、关灯

```
[YaokanSDK turnOnLEDWithYKId:@"targetMacAddr"];
[YaokanSDK turnOffLEDWithYKId:@"targetMacAddr"];
```

#### 15. 硬件升级OTA

```
[YaokanSDK firmwareUpdateWithYKId:@"targetMacAddr"];
```

#### 16. 获取硬件版本

```
[YaokanSDK checkDeviceVersion:(NSString *)ykcId
 completion:(void (^__nullable)(NSString *version,NSString *otaVersion,NSError *error)){
}];
```

#### 17. 设备复位

```
[YaokanSDK restoreWithYKId:@"targetMacAddr"];
```

## 4.2 遥控器接口

### 1. 获取被遥控设备类型列表

```
[YaokanSDK fetchRemoteDeviceTypeWithYKId:[YKCenterCommon sharedInstance] currentYKId]
 completion:^(NSArray<YKRemoteDeviceType *> * _Nonnull types, NSError * _Nonnull error) {
}];
```

be\_rc\_type 的值请参考文档《遥看SDK3技术说明》3.2.4 获取被遥控设备类型列表。

### 2. 获取设备品牌

```
[YaokanSDK fetchRemoteDeviceBrandWithYKId:[YKCenterCommon sharedInstance] currentYKId]
 remoteDeviceTypeId:self.deviceType.tid.integerValue
 completion:^(NSArray<YKRemoteDeviceBrand *> * _Nonnull brands,
             NSError * _Nonnull error)
{
}];
```

### 3. 进入一级匹配(非空调)

```
[YaokanSDK requestFirstMatchRemoteDeviceWithYKId:[YKCenterCommon sharedInstance] currentYKId]
 remoteDeviceTypeId:_tid
 remoteDeviceBrandId:self.deviceBrand.bid
 completion:^(NSArray<YKRemoteMatchDevice *> * _Nonnull mathes, NSError * _Nonnull error) {
}];
```

### 4. 进入二级匹配(非空调)

```
[YaokanSDK requestSecondMatchRemoteDeviceWithYKId:[YKCenterCommon sharedInstance] currentYKId]
 remoteDeviceTypeId:_tid
 remoteDeviceBrandId:_bid group:_gid
 completion:^(NSArray<YKRemoteMatchDevice *> * _Nonnull mathes, NSError * _Nonnull error) {
    if (mathes.count > 0) {
    }
}];
```

### 5. 逐个匹配

```
[YaokanSDK fetchMatchRemoteDeviceWithYKId:[YKCenterCommon sharedInstance] currentYKId]
 remoteDeviceTypeId:_deviceType.tid.integerValue
 remoteDeviceBrandId:_deviceBrand.bid
 completion:^(NSArray<YKRemoteMatchDevice *> * _Nonnull mathes, NSError * _Nonnull error)
{
}];
```

### 6. 匹配阶段空调发码

```
[YaokanSDK sendMatchACWithYKcId:ykcId
    remoteDevice:remoteDevice
    withMode:mode
    temp:temp
    speed:speed
    windU:windU
    windL:windL
    completion:(BOOL result, NSError *error))completion{
    {
}};
```

#### 7. 匹配阶段空调上下风

```
[YaokanSDK sendMatchAcWindUDWithYKcId:ykcId
    remoteDevice:matchDevice
    withMode:mode
    temp:temp
    speed:speed
    windU:windU
    windL:windL
    completion:^(BOOL result, NSError * _Nonnull error) {
}};
```

#### 8. 匹配阶段空调左右风

```
[YaokanSDK sendMatchAcWindLRWithYKcId:ykcId
    remoteDevice:matchDevice
    withMode:mode
    temp:temp
    speed:speed
    windU:windU
    windL:windL
    completion:^(BOOL result, NSError * _Nonnull error) {
}};
```

#### 9. 保存遥控器

```
[YaokanSDK saveRemoteDeivceWithYKcId:[YKCenterCommon sharedInstance] currentYKcId]
remoteDeviceTypeId:typeId
remoteDeviceId:matchDevice.rid completion:^(YKRemoteDevice * _Nonnull remote, NSError * _Nonnull error) {
}};
```

#### 10. 删除遥控器

```
+ (void)removeRemoteDeivceWithYKcId:(NSString *)ykcId
remote:(YKRemoteDevice *)remote
completion:(void (^__nullable)(NSError *error))completion;
```

#### 11. 获取遥控器详情

```
+ (void)requestRemoteDeivceWithYKcId:(NSString *)ykcId
remoteDeviceTypeId:(NSUInteger)typeId
remoteDeviceId:(NSString *)remoteDeviceId
completion:(void (^__nullable)(NSArray *matchKeys, NSError *error))completion;
```

#### 12. 获取遥控器列表

```
[YKRemoteDevice modelsWithYkcId:[[YKCenterCommon sharedInstance] currentYKId]];
```

#### 13. 导出遥控器JSON数据

```
[YaokanSDK restoreWithYKId:@"targetMacAddr"];
```

#### 14. 导入遥控器列表

```
[YKRemoteDevice exportRemotesWithYkcId:[[YKCenterCommon sharedInstance] currentYKId]];
```

### 4.3 SDK错误码表

主要列出了调用SDK的时候，SDK回调返回的错误码信息

值	代码	说明
0	YKSDK_SUCC	操作成功
1001	YKSDK_APPID_INVALID	APPID无效
1002	YKSDK_SMARTCONFIG_TIMEOUT	设备配置超时
1003	YKSDK_DEVICE_DID_INVALID	设备 did 无效
1004	YKSDK_DEVICE_DID_INVALID	设备 did 无效
1005	YKSDK_CONNECTION_TIMEOUT	连接超时
1006	YKSDK_CONNECTION_REFUSED	连接被拒绝
1007	YKSDK_CONNECTION_ERROR	连接错误
1008	YKSDK_CONNECTION_CLOSED	连接被关闭
1009	YKSDK_SSL_HANDSHAKE_FAILED	ssl 握手失败
1010	YKSDK_INTERNET_NOT_REACHABLE	当前外网不可达
1011	YKSDK_NOT_INITIALIZED	SDK 未初始化
1012	YKSDK_PERMISSION_NOT_SET	app 权限不足