

千寻车载惯导 iOS SDK

开发指南

V1.1.0

千寻位置网络有限公司

2016 年 12 月·上海

法律声明

版权所有© 2016，千寻位置网络有限公司。保留一切法律权利。本文档包含的所有内容除特别声明之外，版权均属于千寻位置网络有限公司所有，受《中华人民共和国著作权法》及相关法律法规和中国加入的所有知识产权方面的国际条约的保护。未经本公司书面许可，任何单位和个人不得以任何方式(电子或机械,包括影印)或理由对该文档或其包含的任何产品、服务、信息、材料的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售，否则将视为侵权，本公司必依法追究其法律责任。本文档并不代表供应商或其代理的承诺，千寻位置网络有限公司可在不作任何申明的情况下对本文档内容进行修改。本文档中提到的其它公司及其产品的商标所有权属于该商标的所有者。

千寻位置网络有限公司

联系邮箱：service@qxwz.com

官方网站：www.qxwz.com

目 录

第一章 概述	4
1.1 产品简介	4
第二章 开发指南	5
2.1 使用流程	5
2.2 申请 AppKey、AppSecret	5
2.3 配置工程	5
2.3.1 注意事项	5
第三章 SDK 接口说明	8
3.1 示例代码	8
3.2 API 说明	9
3.2.1 QXWZSDK 类	9
3.2.2 QXWZVDRManager 类	9
3.2.3 QXWZVDRManagerDelegate 代理	10
3.2.4 QXWZVDRLocation 坐标类	10
3.2.5 QXWZVDRLocation 坐标类	10

第一章 概述

1.1 产品简介

千寻车载惯导 iOS SDK 为基于 iOS 平台的移动端应用提供的开发包是根据智能终端内置或者汽车前装 GNSS 芯片定位结果与 MEMS 惯性传感器（陀螺仪，加速度计等）的原始数据，通过松耦合的方式实现 GNSS / INS 的数据融合，从而达到组合定位效果的定位包；是第一个实现基于云、端一体化的惯导产品，本文简称为车载惯导 SDK。该产品致力于解决在没有卫星信号和网络信号覆盖情况下的定位问题，提升千寻定位类产品的综合能力。

千寻位置网车载惯导 SDK（iOS 版）主要提供以下功能：

GPS 开启后，获取到当前的准确坐标并传给车载惯导引擎，根据当前传感器数据综合推算出运行中的当前坐标；当没有 GPS 数据时，根据当前持续的传感器数据和之前的位置信息推算出运行中的坐标。

第二章 开发指南

2.1 使用流程

SDK 使用流程大体如下：

需要先申请相关的 AppKey，获取相关的密钥。然后工程引入 SDK 包，调用对应的 API 接口完成操作。

2.2 申请 AppKey、AppSecret

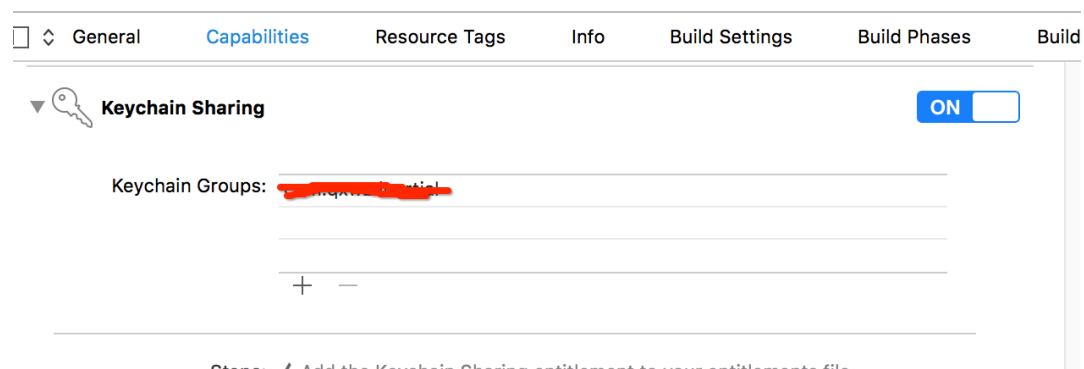
AppKey、AppSecret 由千寻位置网提供，获取办法请见[在线帮助文档](https://www.qxwz.com/help-document-settled-platform.html#link-0)

(<https://www.qxwz.com/help-document-settled-platform.html#link-0>)。

2.3 配置工程

2.3.1 注意事项

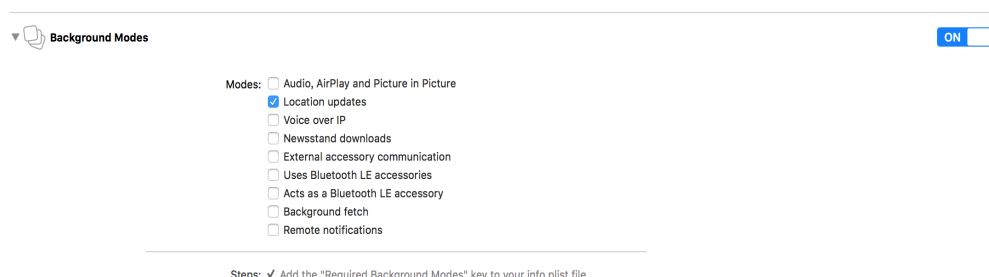
1、iOS 车载惯导 SDK，为了满足一台设备能够拥有一个唯一的 ID，采用 keychain 的方式，在 SDK 首次启动的时候会生成一个 UUID。为了确保 SDK 能够顺利接入，请进入项目的配置，选择 Capabilities，将 Keychain Sharing 打开，如下图：



2、您的项目需要配置允许用户位置授权项，查看 Info.plist 文件源码，添加如下配置。

```
<key>NSLocationAlwaysUsageDescription</key>
<string>yours app 要使用您的位置</string>
```

3、如果您的 app 需要在后台使用 VDR SDK，请在 Capabilities->Background Modes 下勾选 Location updates。



2.3.2 配置步骤

(1) 下载开发包并解压。解压后得到三个文件，分别是

- QXWZVDRSDK.framework
- QXWZSDK_CoreData.bundle
- libQXWZVDRInertia.a

(2) 引入 VDR 静态包

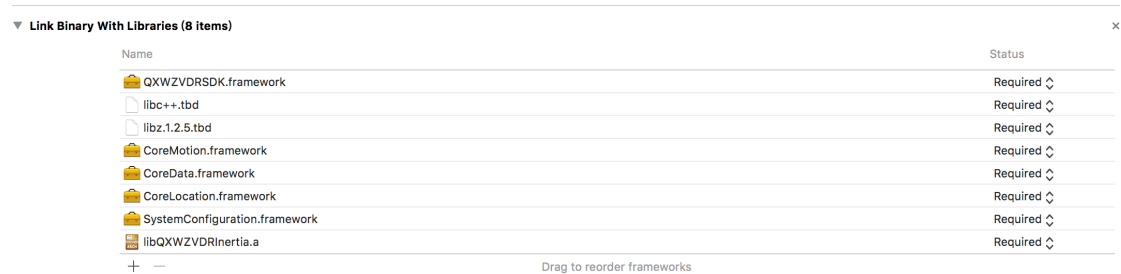
在 Targets->Build Phases->Link Binary With Libraries 配置项点击 “+”，弹出添加列表后，点击 “Add Other...”，添加 QXWZVDRSDK.framework、libQXWZVDRInertia.a 到工程中。

(3) 引入系统库文件

在 Targets->General->Linked Frameworks and Libraries 中点击 “+”，弹出添加列表后添加如下系统库：

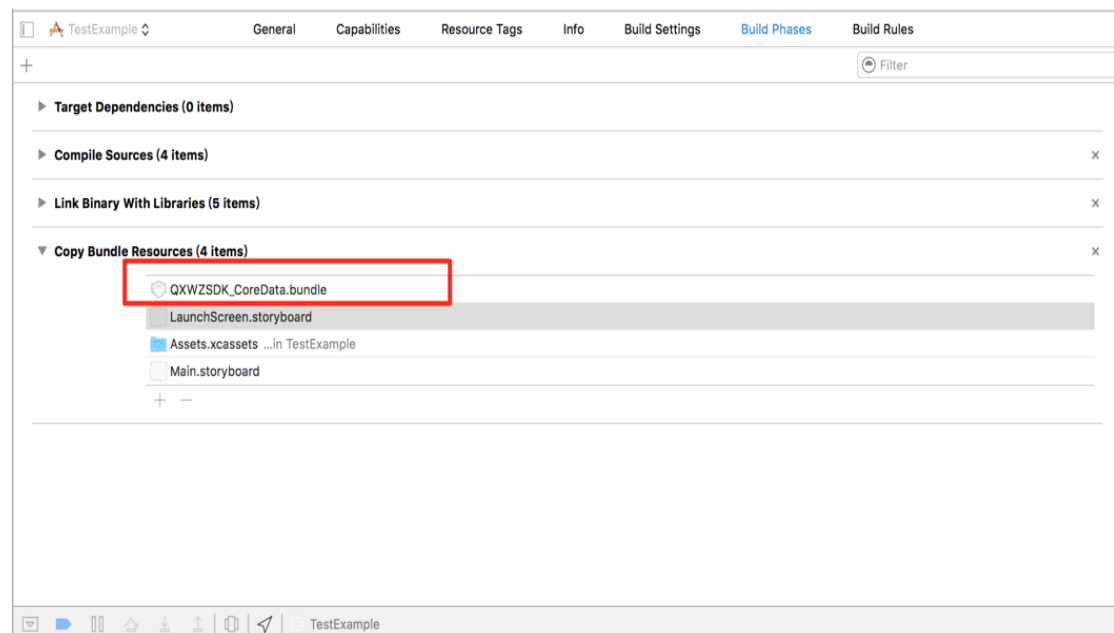
- CoreLocation.framework
- CoreMotion.framework
- CoreData.framework
- SystemConfiguration.framework
- libc++.tdb
- libz.1.2.5.tdb

上述(2)(3)两步完成之后，Link Binary With Libraries 应该如截图：



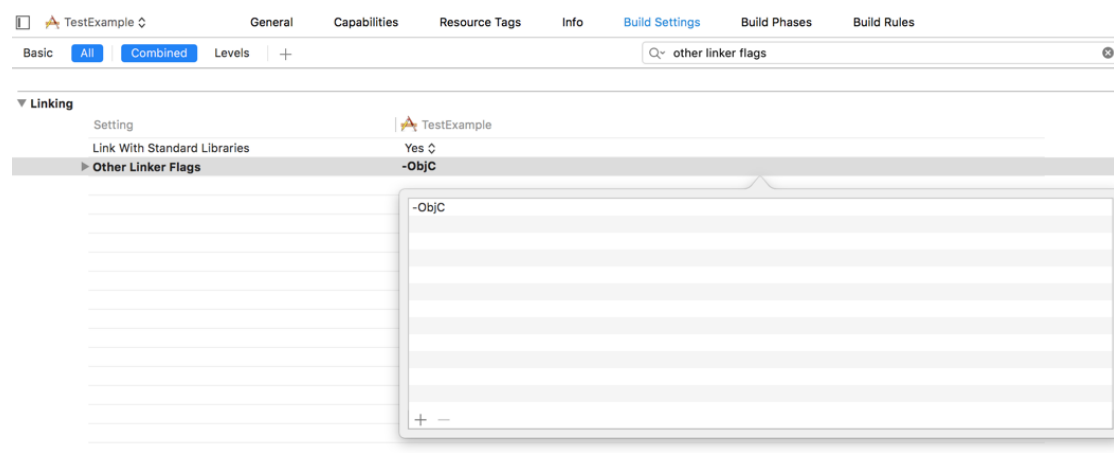
(4) 引入资源文件

在 TARGETS->Build Phases->Copy Bundle Resources 中点击 “+”，弹出添加列表后添加如下资源文件：QXWZSDK_CoreData.bundle



(5) 添加 Other Linker Flags

左侧目录中选中工程名，在 Targets->Building Settings->Linking->Other Linker Flags 中加上-ObjC



第三章 SDK 接口说明

3.1 示例代码

```
#import <QXWZVDRSDK/QXWZLocation.h>
#import <QXWZVDRSDK/QXWZVDRManager.h>
#import <QXWZVDRSDK/QXWZVDREnum.h>
#import <QXWZVDRSDK/QXWZSDK.h>

@interface TestViewController()<QXWZVDRManagerDelegate>
@property (nonatomic, strong) QXWZVDRManager *vdrManager;
@end

@implementation TestViewController

- (instancetype)init
{
    self = [super init];
    if (self) {
        [QXWZSDK setAppKey:@"从千寻位置官网申请的 appKey" appSecret:@"从千寻位置官网申请的 appSecret"];
        _vdrManager = [[QXWZVDRManager alloc] init];
        _vdrManager.delegate = self;
    }
    return self;
}

// 启动 VDR SDK
- (void)start{
    [_vdrManager startUpdatingLocation];
}

// 关闭 VDR SDK
- (void)stop{
    [_vdrManager stopUpdatingLocation];
}

#pragma mark - QXWZVDRManagerDelegate

- (void)qxwz_vdrManager:(QXWZVDRManager *)vdrManager location:(QXWZLocation *)location posFlag:(QXWZPositionFlag)posFlag{
    // Todo: 坐标更新时回调
}
```

```

}

-(void)qxwz_vdrManager:(QXWZVDRManager *)vdrManager
statusUpdate:(QXWZVDRStatus)status message:(NSString *)message{
    // Todo: 惯导状态变化时回调
}

@end

```

3.2 API 说明

3.2.1 QXWZSDK 类

用于设置 appKey、appSecret，首先要调用 setAppKey:appSecret 方法。

方法/变量	描述
setAppKey:appSecret	设置从千寻位置官网申请的 appKey、appSecret，该方法要在所有 API 之前调用。
sdkVersion	查看当前 VDR SDK 的版本号
appKey	查看设置的 appKey
deviceId	查看设备的唯一标识

3.2.2 QXWZVDRManager 类

iOS VDR SDK 主入口，主要负责采集 GPS 坐标和传感器数据并计算出推算位置。

方法/变量	描述
delegate	设置需要处理回调的类的对象。
isStarted	VDR SDK 是否已经启动。
startUpdatingLocation	启动惯导。
stopUpdatingLocation	关闭惯导，会关闭 GPS 定位、传感器，释放部分资源。
setDebugModel	调用该方法，SDK 会收集运行日志，帮助开发者分析问题，默认情况下不要调用。

3.2.3 QXWZVDRManagerDelegate 代理

该代理里面包含两个回调方法。

方法/变量	描述
<code>-(void)qxwz_vdrManager:(QXWZVDRManager *)vdrManager location:(QXWZLocation *)location posFlag:(QXWZPositionFlag)posFlag;</code>	当坐标发生变化时回调。 location 中主要关注 latitude、longitude 和 timestamp； posFlag 是定位状态,表示当前坐标是自主定位、自主惯导组合定位还是纯惯导定位。
<code>-(void)qxwz_vdrManager:(QXWZVDRManager *)vdrManager statusUpdate:(QXWZVDRStatus)status message:(NSString *)message;</code>	当惯导状态发生变化时回调。 status 为枚举类型,具体的取值范围请参照 QXWZVDREnum； message 为返回的状态描述。

3.2.4 QXWZVDRLocation 坐标类

继承自 CLLocation, 可使用 CLLocation 的经纬度、高程、方向、速度、时间戳等。

方法/变量	描述
posFlag	位置类型, 参考 QXWZPositionFlag
numOfAcc	急加速次数
numOfDecel	急减速次数
numOfSharpTurn	急转弯次数
numOfSharpSwitch	急变道次数
numOfOverSpeed	超速次数

3.2.5 QXWZVDRLocation 坐标类

VDR 运行状态枚举：

QXWZVDRStatus 枚举 Code	message 描述
QXWZ_STATUS_APPKEY_IDENTIFY_FAILURE	AppKey 认证失败, 可能是 appKey 和 appSecret 错误
QXWZ_STATUS_APPKEY_IDENTIFY_SUCCESS	AppKey 认证成功

QXWZ_STATUS_NETWORK_UNAVAILABLE	网络异常
QXWZ_STATUS_VDR_MAX_USER	APP KEY 用户已达上限
QXWZ_STATUS_ILLEGAL_APP_KEY	非法 APP KEY
QXWZ_STATUS_ILLEGAL_APP_SECRET	非法 APP SECRET
QXWZ_STATUS_SDK_INTERNAL_ERROR	DK 内部错误
QXWZ_STATUS_OPENAPI_SYSTEM_ERROR	服务端内部错误
QXWZ_STATUS_VDR_STARTING	VDR 开始启动
QXWZ_STATUS_NO_GPS_PERMISSION	用户没有授权 GPS 权限
QXWZ_STATUS_VDR_STARTED_SUCCESS	VDR 完成启动
QXWZ_STATUS_VDR_STARTED_FAIL	VDR 启动失败
QXWZ_STATUS_VDR_CLOSE_SUCCESS	VDR 关闭成功
QXWZ_STATUS_VDR_CLOSE_FAIL	VDR 关闭失败

VDR 回调位置枚举

QXWZPositionFlag 枚举 Code	message 描述
QXWZPositionFlag_Invalid	出现这种状态码，一般 VDR 初始化的过程中是没有获取到 GPS 坐标。
QXWZPositionFlag_Single	单点定位，表示是纯 GPS 定位。
QXWZPositionFlag_Group	GPS、惯导组合定位。惯导初始化成功，并且这个时候获取到的 GPS 坐标精度不够，会返回组合定位。
QXWZPositionFlag_VDR	纯惯导定位。惯导初始化成功，并且这个时候获取不到 GPS 坐标，会返回纯惯导定位。