

星與智能穿戴定位 SDK

开发指南 V2.1

广东星與科技有限公司

2021 年 10 月

法律声明

Copyright© 2019 广东星與科技有限公司

版权所有，保留所有权利

未经本公司书面许可，任何单位和个人不得以任何方式（包括仿制、复制、誊写或转译）本文档部分或全部内容。不得以任何形式或任何方式（电子、机械、影印、录制或其他可能的方式）进行商品的传播或用于任何商业、盈利目的。对本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、传播或其它产品捆绑使用、销售，均视为侵权，本公司必依法追究其法律责任。

广东星興科技有限公司有权对本文档内容进行更新，恕不另行通知。除非另有特殊约定，本手册仅作为使用指导，本手册所有陈述、信息均不承担任何形式的担保。

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。

广东星興科技有限公司

联系邮箱：service@starcart.cn

官方网站：www.starcart.cn

文档更新记录

版本	更新日期	更新摘要
V1.0.0	2021-07	增加传感器数据融合
V1.1.0	2021-08	优化数据处理, 修复已知问题
V1.2.0	2021-09	优化融合策略
V2.0.0	2021-10	支持鉴权操作
V2.1.0	2021-10	支持 AGNSS、CORS 服务接入

星與科技

目录

文档更新记录.....	3
目录.....	4
第一章 概述.....	5
第二章 开发指南.....	6
2.1 开发准备.....	6
2.1.1 开发环境.....	6
2.1.2 资源清单.....	6
2.2 集成流程.....	7
2.3 详细步骤.....	7
2.3.1 SDK 包导入.....	7
2.3.2 build.gradle 配置.....	8
2.3.3 权限配置.....	9
2.3.4 SDK 初始化.....	9
2.3.5 SDK 启动.....	10
2.3.6 获取 A-GNSS 数据.....	11
2.3.7 CORS 启动与停止.....	11
2.3.8 融合定位.....	12
2.3.9 SDK 停止.....	12
第三章 接口说明.....	13
3.1 鉴权接口回调.....	13
3.3 AGNSS 接口回调.....	15
3.4 CORS 接口回调.....	15
3.5 其他状态信息回调.....	16

第一章 概述

星與智能穿戴定位 SDK 是星與科技开发的适用于安卓平台的智能智能穿戴类设备接入星與星與差分服务并融合多传感器数据以输出高精度定位 SDK 的开发包。

用户可以通过 SDK 轻松接入星與差分服务和 A-GNSS 服务，并且将其他传感器数据与 GNSS 数据融合得到高精度定位数据。

星與科技

第二章 开发指南

2.1 开发准备

2.1.1 开发环境

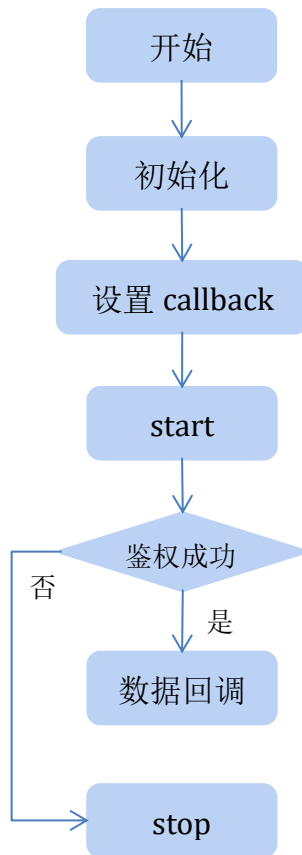
SDK 基于 JDK1.8、Android4.0+开发

2.1.2 资源清单

- `starcart_wearable_sdk_x.x-yyy.aar`

其中，`x.x-yyy` 版本号

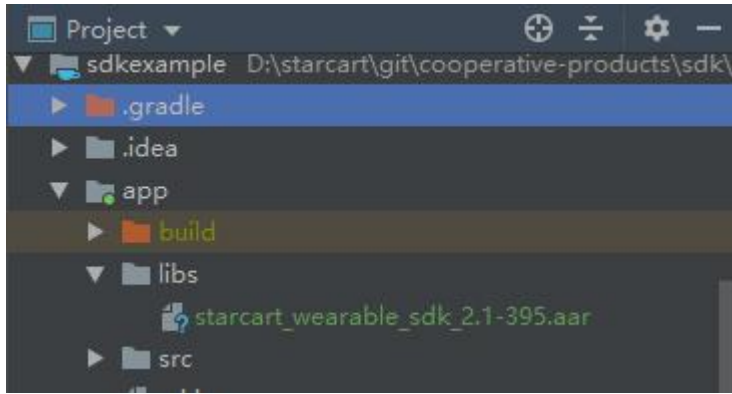
2.2 集成流程



2.3 详细步骤

2.3.1 SDK 包导入

在工程 `app/libs` 目录下放入 2.1.2 资源清单中的 `aar` 包，如下图所示：



2.3.2 build.gradle 配置

在 app/build.gradle 中 android 标签添加对 aar 包和相关库引用，如下：

```
android{
    .....
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
    .....
}

.....

dependencies {
    implementation fileTree(include: ['*.jar', '*.aar'], dir: 'libs')
    implementation 'com.squareup.okhttp3:okhttp:3.10.0'
}
```


2.3.3 权限配置

SDK 需要使用网络，所以 app 需要开通网络权限，在 app 的 AndroidManifest.xml 文件添加如下配置：

```
//网络权限
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

2.3.4 SDK 初始化

建议在 Application 或者 MainActivity 中初始化 SDK，调用示例如下：

```
//初始化 appkey 和设备号
StarDgnss.get().init(getApplicationContext(), "appkey", "deviceId");
//设置日志等级和回调实现
StarDgnss.get().setLogLevel(LogLevel.DEBUG).setDgnssCallback(new
OnDgnssCallback() {

    //初始化鉴权状态回调
    @Override
    public void onInit(int code, String msg) {

    }
    //AGNSS 数据回调，如果发起 AGNSS 请求的话
    @Override
    public void onAgnss(int code, byte[] data) {

    }
    //CORS 服务数据回调
    @Override
    public void onCorsData(long timestamp, int size, byte[] data) {

    }
    //CORS 服务状态回调
```

```
@Override
public void onCorsStatus(int code, String msg) {

}
//SDK 运行状态回调
@Override
public void onStatusChanged(int code, String msg) {

}
//SDK 信息日志等回调
@Override
public void onInfo(String msg) {

}
}
```

其中，appkey 为在星興开放平台注册创建应用时生成的密钥，deviceId 为设备 ID，建议使用设备唯一标识 IMEI 号，传入无效 appkey 和 deviceId 将抛出异常，初始化失败

2.3.5 SDK 启动

SDK 初始化后，需调用启动接口才会接入星興差分服务器进行鉴权操作，鉴权成功后才支持接入 CORS 和融合定位等操作，鉴权操作是异步的，鉴权结果回调参考 3.1 接口说明

```
//SDK 启动
StarDgnss.get().start();
```

2.3.6 获取 A-GNSS 数据

当前版本支持 A-GNSS 服务接入功能，该功能不需要等鉴权通过即可使用，在 SDK 初始化成功之后即可调用，发起 A-GNSS 请求是异步的，传入对应模块支持的系统获取对应系统星历，默认按 GPS 和北斗获取 A-GNSS 数据，支持标准 RTMC 数据和第三方 A-GNSS 数据，A-GNSS 结果回调参考 3.2 接口说明，示例如下：

```
//异步获取 A-GNSS 请求
ModuleType moduleType = ModuleType.HDS;
//moduleType.GPS = 1; //获取 GPS 星历，默认
//moduleType.BDS = 1; //获取北斗星历，默认
moduleType.GAL = 1; //获取伽利略星历
StarDgnss.get().getAgnssAsyn(moduleType);
```

2.3.7 CORS 启动与停止

当前版本支持 CORS 服务接入功能，该功能需要鉴权成功即 onInit 回调状态码为 1000 时才能生效，CORS 服务为独立线程接入，回调结果参考 3.4 接口说明，可通过如下接口启动和停止 CORS 服务：

```
//启动 cors，需传入当前位置 GGA，也可以传入经纬度初始化
StarDgnss.get().startCors("$GPGGA,082230.000,2307.3293732,N,11320.7778254,E,4,22,0.755,3.700,M,-6.713,M,01,0032*49");

//停止 cors
StarDgnss.get().stopCors();
```

2.3.8 融合定位

当前版本支持融合定位功能，该功能需要鉴权成功即 `onInit` 回调状态码为 1000 时才能生效，使用时需其他传感器数据的数据，以及传入定位模块输出的 GGA 数据，两个调用中间无先后顺序，实时更新数据即可，调用示例如下：

```
//传入传感器数据
SensorData sensorData = new SensorData();
sensorData.setRelative_height(0.9f); //相对高度
sensorData.setTemperature(27f);     //当前温度
sensorData.setPressure(1031f);      //大气压强
StarDgnss.get().setSensorData(sensorData);

//融合 GGA
result = StarDgnss.get().fusion(gga);
```

2.3.9 SDK 停止

SDK 初始化后，可调用如下接口释放相关资源、停止 SDK：

```
//SDK 停止
StarDgnss.get().stop();
```

第三章 接口说明

3.1 鉴权接口回调

SDK 鉴权信息通过初始化时设置的回调实现中 onInit 接口回调：

```
//鉴权信息回调  
void onInit(int code,String msg);
```

其中回调参数 code 可能如下：

code	message	说明
1000	Success	成功
1001	Failure	执行失败
1002	Missing param	参数缺失
1003	Invalid param	无效参数
1004	Invalid request	无效请求
1005	Unknown error	未知错误
2000	App does not exist	应用不存在
2001	App reach limit count	应用用量已到达限额
3000	User does not exist	用户不存在
3001	User name is already used	用户账号已被占用
3002	User is disabled	用户账号被禁用

3003	User activation time exceeded	用户账号激活时间已超过有效期
3004	User has been expired	用户账号的使用时间已到期
3005	User has not been activated	用户账号不存在激活时间
3006	User activation way is error	用户账号激活方式不符合
4000	Device does not exist	设备不存在
4001	Device code is already used	设备编码已被占用
5000	Service does not exist	服务不存在
5001	Service is expired	服务已过期
5002	Service config missing	服务配置缺失
6000	Vrs invalid service type	VRS 服务类型不正确
6001	No valid gga	无效的 gga
6002	No obs message	没有观测电文
6003	No nav message	没有导航电文
6004	No message	没有观测电文和导航电文
6005	No vrs info	不存在 VRS 的信息
7000	Token is expired	Token 已过期
7001	Token is invalid	Token 无效
7002	Token does not exist	Token 不存在
8000	Version exist	SDK 版本已存在

3.3 AGNSS 接口回调

SDK 接入 AGNSS 服务的数据和状态通过初始化时设置的回调实现中 onAgnss 接口回调：

```
//AGNSS 数据回调  
void onAgnss(int code,String msg);
```

其中回调参数中状态码参考 3.1

3.4 CORS 接口回调

SDK 接入 CORS 服务数据和状态通过初始化时设置的回调实现中 onAgnss 接口回调：

```
//CORS 数据回调数据  
void onCorsData(long timestamp,int size,byte[] data);  
  
//CORS 服务状态回调  
void onCorsStatus(int code,String msg);
```

其中回调参数中状态码参考 3.1 以及如下

code	message	说明
4005	Service Connecting	TCP 服务连接中
4006	Service Connected	TCP 服务已连接
4008	Service User Identity Success	用户认证成功
4013	GGA send success	GGA 发送成功

3.5 其他状态信息回调

SDK 其他数据和状态通过初始化时设置的回调实现中

```
//其他状态回调  
void onStatusChanged(int code, String msg);  
  
//日志信息等回调  
void onInfo(String msg);
```

星與科技