

# 集成指南

| **类别** | **兼容范围** |
| --- | --- |
| minSdkVersion系统 | 最低21 |
| 开发环境 | 建议使用最新版本Android Studio 进行开发 |
| 跟新时间 | 2020-11-24 |

DEMO 中已经集成了 SDK。您可以参考DEMO，集成SDK。

集成前，请先测通DEMO，了解调用原理。

## AndroidManifest.xml 文件

设置权限：

<!-- 必要的权限 -->

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

## 引入aar包

将****remote\_sdk.aar** 复制到您的项目**app/libs目录中。确认在build.gradle文件中引入。

<!-- 引入aar文件 -->

implementation files('libs/sdk.aar')

## 接口调用过程

DEMO在SDK的基础上，实现了调用逻辑。SDK方式调用请参考MainActivity，本文仅描述SDK的调用方式。

### 获取 AirController实例

AirController mAirController = new AirController(this, new AirControlListener() {

  @Override  
    public void onCameraReset() {

     }

}); // this 是Context的之类，如Activity

mAirController.start(); // 启动服务连接

注意 必须调用start(); 该类是其他类正常工作的基类,在不需要使用该服务时记得释放资源调用stop();

### 获取DataStream实例

DataStream mDataStream = new DataStream (mAirController, 1, new DataStreamListener() {

  @Override  
  public void onDataReceived(byte[] data) {

// 接收数据时调用此方法  
  }

@Override  
  public void onUartReady(boolean b) {

// 准备就绪或不获取/设置Uart波特率时将调用此方法  
  }

  @Override  
  public void onSetUartBaudrateDone(Types.Result result) {

// 设置Uart波特率的请求完成后，将调用此方法  
  }

});

mDataStream .start();

注意 必须调用start(); 在不需要使用该服务时记得释放资源调用stop();

### 获取VideoStream实例

VideoStream mVideoStream = new VideoStream(mAirController, new VideoStreamListener() {

  @Override  
  public void onVideoStreamIdChanged(int id) {

// 更改此VideoStream的视频流ID时，将调用此方法

}

@Override  
  public void onPlayReady(boolean ready) {

// 准备好视频流播放时调用此方法  
  }

@Override  
  public void onPlayStateChanged(boolean playing) {

// 播放状态改变时调用此方法，true播放中  
  }

  @Override  
  public void onVideoStreamResolutionChanged(Types.Resolution resolution) {

// 更改视频流的分辨率时调用此方法  
  }

});

mVideoStream.start();

// 设置显示视频视图，参考demo中的使用情况

mVideoStream.setSurface(surface);

// 播放视频

mVideoStream.play();

注意 必须调用start(); 在不需要使用该服务时记得释放资源调用stop();

其它方法请参见 接口文档或demo