

# IDO-SBC3566-V1B - Android 二次开发API说明

---

## 1 自定义API功能使用

### 1.1 关机和重启

#### 1.1.1 关机

#### 1.1.2 重启

### 1.2 定时开关机

### 1.3 以太网

#### 1.3.1 以太网口设置静态IP地址

#### 1.3.1 以太网口设置动态获取IP地址

### 1.4 WIFI

### 1.5 飞行模式

### 1.6 APP前台运行守护

### 1.7 截屏

### 1.8 设置系统时间

### 1.9 APK静默安装

### 1.10 屏幕旋转

### 1.11 RS485模式配置

### 1.12 状态栏显示和隐藏

#### 1.12.1 隐藏状态栏

#### 1.12.2 显示状态栏

### 1.13 导航栏显示和隐藏

#### 1.12.1 隐藏导航栏

#### 1.12.2 显示导航栏

## 2 Launcher配置方法

## 3 framework.jar导入Android Studio

## 4 Android APK 签名

# IDO-SBC3566-V1B

## Android系统二次开发API说明

深圳触觉智能科技有限公司

[www.industio.cn](http://www.industio.cn)

### 文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档，初始版本	FuYingzhe		2022/11/04

# 1 自定义API功能使用

## 1.1 关机和重启

通过广播的方式设置系统关机和重启

### 1.1.1 关机

功能说明：

发送广播的方式关闭设备。

参数说明：

名称	值	说明
confirm	true	发送广播后，会弹出是否关机的确认窗口
	false	发送广播后，无弹窗，直接关机

调用示例：

▼Java

```
1 Intent intent = new Intent("android.ido.intent.action.set.shutdown");
2 intent.putExtra("confirm", true);
3 sendBroadcast(intent);
```

### 1.1.2 重启

功能说明：

发送广播的方式重启设备。

参数说明：

名称	值	说明
confirm	true	发送广播后，会弹出是否重启的确认窗口
	false	发送广播后，无弹窗，直接重启

调用示例：

▼

Java

```
1 Intent intent = new Intent("android.intent.action.set.reboot");
2 intent.putExtra("confirm", true);
3 sendBroadcast(intent);
```

## 1.2 定时开关机

功能说明：

发送广播，设置定时开关机的时间，设置的时间需要是未来时间。

参数说明：

名称	值
timeon	整数类型时间数组 例：int[] mOnTime = {2022,9,26,9,35};
timeoff	整数类型时间数组 例：int[] mOffTime = {2022,9,26,9,34};

调用示例：

▼

Java

```
1 int[] mOffTime = {2022,9,26,9,34}; //{年,月,日,时,分}
2 int[] mOnTime = {2022,9,26,9,35}; //{年,月,日,时,分}
3 Intent mIntent = new Intent("android.intent.action.set.rtc.poweronoff");
4 mIntent.putExtra("timeon",mOnTime); //开机时间，见上面数组定义
5 mIntent.putExtra("timeoff",mOffTime); //关机时间，见上面数组定义
6 mIntent.putExtra("enable",true); //true开启,false关闭
7 sendBroadcast(mIntent);
```

## 1.3 以太网

### 1.3.1 以太网口设置静态IP地址

功能说明：

发送广播配置以太网口为静态IP地址。

参数说明：

名称	值	说明
mode	Static	设置为IP地址为静态模式
iface	eth0	以太网卡名称
ipAddr	IP地址	点分十进制IPV4地址
netmask	掩码	点分十进制IPV4掩码
gateway	网关地址	点分十进制IPV4地址
dns1	DNS	第一个点分十进制DNS地址
dns2	DNS	第二个点分十进制DNS地址，非必须

调用示例：

▼Java

```
1 Intent intent = new Intent("android.ido.intent.action.ethernet");
2 intent.putExtra("mode", "Static");
3 intent.putExtra("iface", "eth0");
4 intent.putExtra("ipAddr", "192.168.0.7");
5 intent.putExtra("netmask", "255.255.255.0");
6 intent.putExtra("gateway", "192.168.0.1");
7 intent.putExtra("dns1", "114.114.114.114");
8 intent.putExtra("dns2", "8.8.8.8"); //如果只有一个dns，可将dns2设置为和dns1一样
9 sendBroadcast(intent);
```

### 1.3.1 以太网口设置动态获取IP地址

功能说明：

发送广播配置以太网口IP地址获取方式为DHCP。

参数说明：

名称	值	说明
mode	DHCP	设置为IP地址为动态获取
iface	eth0	以太网卡名称

调用示例：

▼ Java

```
1 Intent intent = new Intent("android.ido.intent.action.ethernet");
2 intent.putExtra("mode", "DHCP");
3 intent.putExtra("iface", "eth0");
4 sendBroadcast(intent);
```

## 1.4 WIFI

功能说明：

发送广播开启和关闭WIFI。

参数说明：

名称	值	说明
enable	true	开启WIFI
	false	关闭WIFI

调用示例：

▼ Java

```
1 Intent intent = new Intent( "android.ido.intent.action.wifionoff");
2 intent.putExtra("enable", true); //true, 开启; false, 关闭
3 sendBroadcast(intent);
```

## 1.5 飞行模式

功能说明：

此功能用于配置系统的飞行模式开启和关闭。

参数说明：

名称	值	说明
enable	true	开启飞行模式

	false	关闭飞行模式
--	-------	--------

调用示例：

▼ Java

```
1 Intent intent = new Intent("android.intent.action.set.airPlaneMode");
2 intent.putExtra("enable", true);
3 sendBroadcast(intent);
```

## 1.6 APP前台运行守护

功能说明：

此功能是设置循环检测APP是否在最前端运行（在界面上显示）。

参数说明：

名称	值	说明
enable	true	开启检测功能
	false	关闭检测功能
packageName	包名	需要保持前台运行的APK的包名
className	类名	需要保持前台运行的APK的类名
checkCnt	检测时间	单位2秒，例：设置为3，则6秒未检测到APP在最前端，则重启拉起

调用示例：

```

1  private void AppTopCheck(String packageName, String className, boolean enable){
2      Intent intent = new Intent( "android.ido.intent.action.set.appcheck");
3
4      //开启、关闭app check 功能，开启后，一直检测当前最前端显示
5      //是否为设置的app，此设置断电会保存，开机自动运行
6      intent.putExtra("enable", enable);
7
8      //检测最前端运行的APP包名
9      intent.putExtra("packageName", packageName);
10     //如果检测到最前端运行的APP不为参数里面设置的包名，则重启APP
11     intent.putExtra("className", className);
12
13     //此参数为设置多少次未检测到APP 在前端运行则重启app，默认循环检测APP 的时间为2
    秒，
14     //这里设置为3 次，如果APP 未在前面运行，则6 秒后则重启app
15     intent.putExtra("checkCnt", 3);
16     sendBroadcast(intent);
17 }

```

## 1.7 截屏

功能说明：

此功能用于截取当前屏幕内容。

参数说明：

名称	值	说明
path	保存路径	例： /sdcard/screenshot.png
id	屏幕ID	此参数缺省为0,当有多个屏幕可选择0、1

调用示例：

```

1  Intent intent = new Intent( "android.ido.intent.action.screenshot");
2  intent.putExtra("path","/sdcard/screenshot.png");//保存png文件的绝对路径
3  intent.putExtra("id","0");//屏幕ID,此参数缺省为0,当有多个屏幕可选择0、1、2...
4  sendBroadcast(intent);

```



## 1.8 设置系统时间

功能说明：

此功能用于设置系统RTC时间。

参数说明：

名称	值	说明
time	整数类型时间数组	整数类型时间数组 {年，月，日，时，分，秒} 例：int[] time = {2022,9,26,9,35,0};

调用示例：

Java

```
1  int[] time = {2022, 9, 30, 18, 0, 0};
2  Intent intent = new Intent("android.ido.intent.action.settime");
3  intent.putExtra("time", time);
4  sendBroadcast(intent);
```

## 1.9 APK静默安装

功能说明：

此功能用于静默安装指定路径的apk。

参数说明：

名称	值	说明
apkFilePath	apk文件路径	绝对路径
autostart	true/false	安装完后是否自动运行

调用示例：

```

1 Intent intent = new Intent("android.intent.action.SILENT_INSTALL_PACKAGE");
2 intent.putExtra("apkFilePath", fileName);//安装apk 绝对路径
3 intent.putExtra("autostart", true);//true:安装完成后自动运行
4 sendBroadcast(intent);

```

## 1.10 屏幕旋转

功能说明：

此功能用于旋转屏幕方向。

参数说明：

名称	值	说明
rotation	旋转角度	旋转角度，0/90/180/270

调用示例：

```

1 Intent intent = new Intent("android.intent.action.lcdrotation");
2 intent.putExtra("rotation", angle);//旋转角度，0/90/180/270
3 sendBroadcast(intent);

```

## 1.11 RS485模式配置

功能说明：

此功能针对ttyS0，ttyS7和ttyS9三路串口；当硬件上配置使用RS485功能时，可通过此接口来软件设置驱动启用GPIO引脚来控制RS485收发器。当硬件上配置为非RS485功能时，可通过此接口来软件设置驱动禁用GPIO自动切换收发。

默认情况下，只有ttyS0设置为RS485模式。

参数说明：

名称	值	说明
port	串口节点号	如ttyS0，填0；ttyS7填7；ttyS9填9

enable	true/false	true: 设置为RS485模式, 由GPIO自动控制RS485收发; false: 关闭驱动RS485控制;
--------	------------	--

调用示例:

▼

Java

```
1 private void setRs485ModeEnable(int port, boolean enable){
2     Intent intent = new Intent("android.ido.intent.action.rs485");
3     intent.putExtra("port", port);
4     intent.putExtra("enable", enable);
5     sendBroadcast(intent);
6 }
```

调用方法

▼

Java

```
1 setRs485ModeEnable(0,true);    //设置 /dev/ttyS0 为RS485模式
2 setRs485ModeEnable(7,false);   //禁用 /dev/ttyS7 的RS485模式
3 setRs485ModeEnable(9,false);   //禁用 /dev/ttyS9 的RS485模式
```

# 1.12 状态栏显示和隐藏

功能说明: 此功能用于设置状态栏的隐藏和显示。

## 1.12.1 隐藏状态栏

参数说明:

名称	值	说明
save	true/false	true,断电保存; false, 断电不保存。

调用示例:

▼

Java

```
1 Intent intent = new Intent("android.ido.intent.action.statusbar.HIDE");
2 intent.putExtra("save", true); //true,断电保存; false, 断电不保存
3 sendBroadcast(intent);
```

## 1.12.2 显示状态栏

参数说明：

名称	值	说明
save	true/false	true,断电保存； false， 断电不保存。

调用示例：

▼ Java

```
1 Intent intent = new Intent("android.ido.intent.action.statusbar.SHOW");
2 intent.putExtra("save", true); //true,断电保存； false， 断电不保存
3 sendBroadcast(intent);
```

## 1.13 导航栏显示和隐藏

功能说明：此功能用于设置导航栏的隐藏和显示。

### 1.12.1 隐藏导航栏

参数说明：

名称	值	说明
save	true/false	true,断电保存； false， 断电不保存。

调用示例：

▼ Java

```
1 Intent intent = new Intent("android.ido.intent.action.navigation.HIDE");
2 intent.putExtra("save", true); //true,断电保存； false， 断电不保存
3 sendBroadcast(intent);
```

### 1.12.2 显示导航栏

参数说明：

名称	值	说明
----	---	----

save	true/false	true,断电保存; false, 断电不保存。
------	------------	--------------------------

调用示例:

```

1  Intent intent = new Intent("android.ido.intent.action.navigation.SHOW");
2  intent.putExtra("save", true); //true,断电保存; false, 断电不保存
3  sendBroadcast(intent);

```

## 2 Launcher配置方法

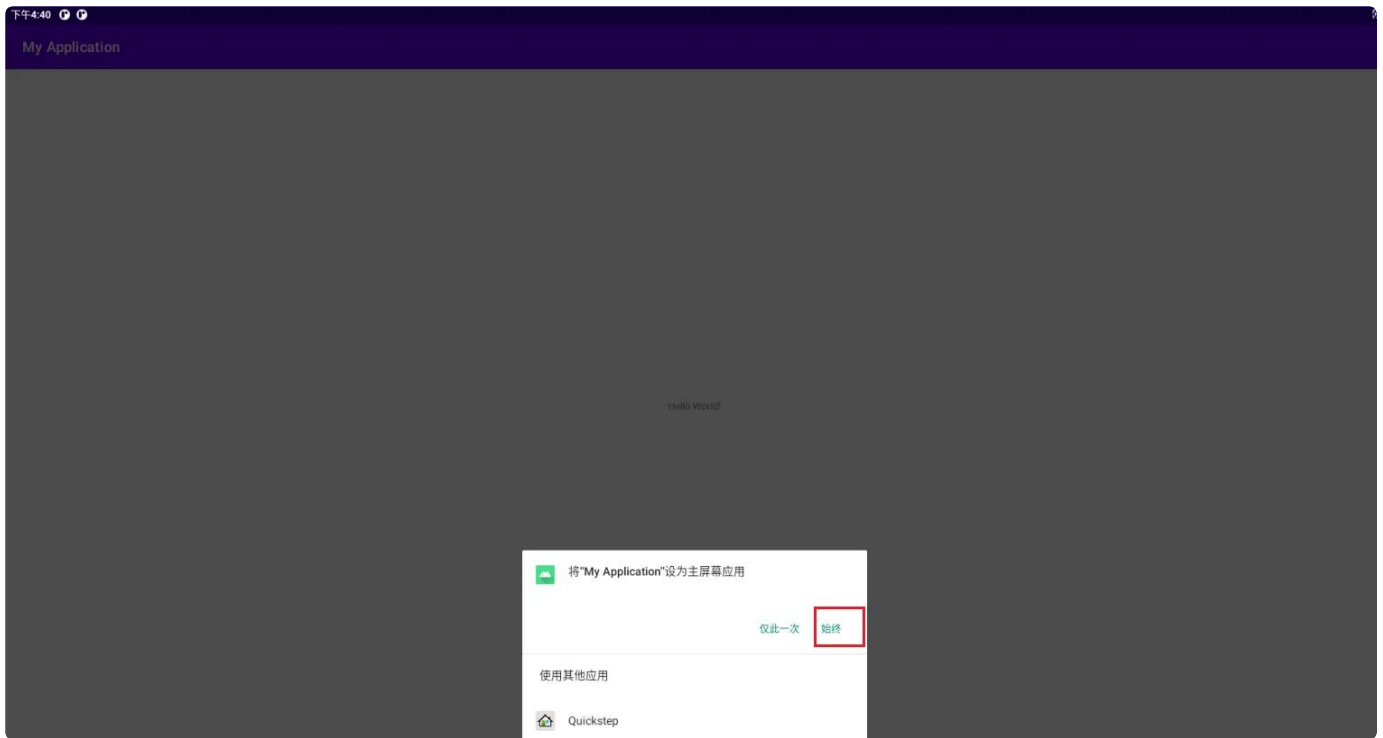
修改APP源码的AndroidManifest.xml, 添加以下内容

```

1  <application
2      ...
3      <activity android:name=".MainActivity">
4          <intent-filter>
5              <action android:name="android.intent.action.MAIN" />
6              <category android:name="android.intent.category.HOME" />
7              <category android:name="android.intent.category.DEFAULT" />
8              <category android:name="android.intent.category.LAUNCHER" />
9          </intent-filter>
10     </activity>
11     ...
12 </application>

```

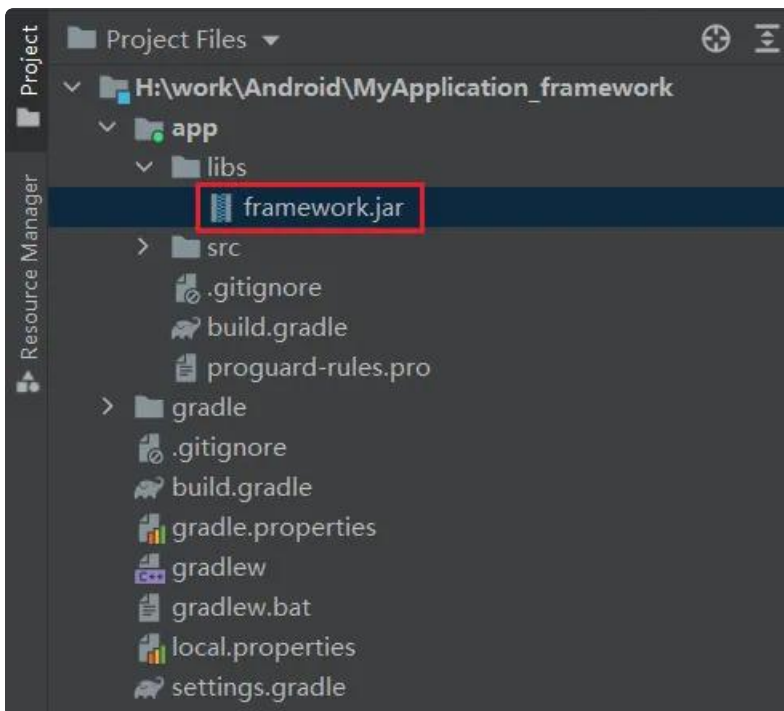
将修改后的apk文件安装至主板中, 主板界面上打开安装的应用程序, 之后退出到主页时, 会弹出提示主屏幕应用选择框, 选择安装的应用并设置为“始终”, 之后此应用程序将会作为默认的Launcher。



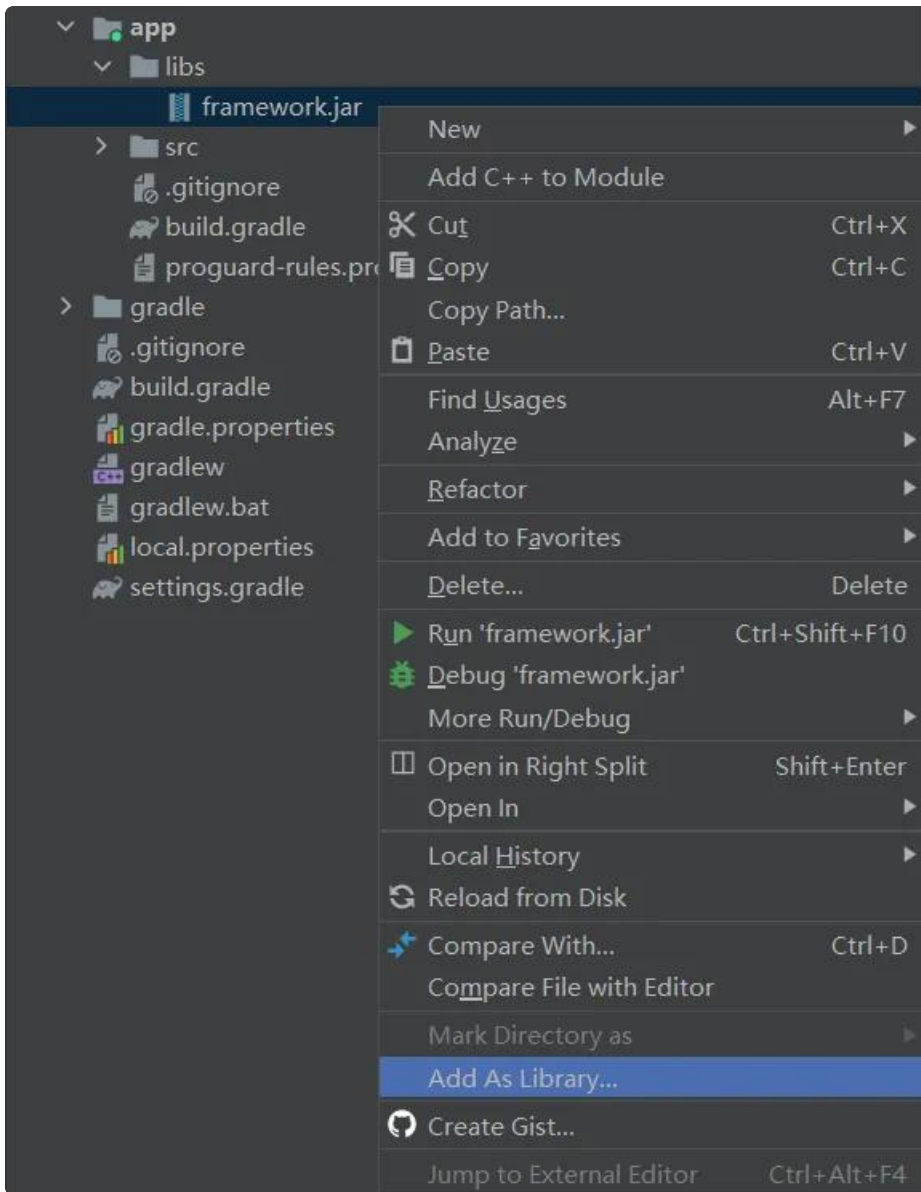
### 3 framework.jar导入Android Studio

framework.jar下载地址：

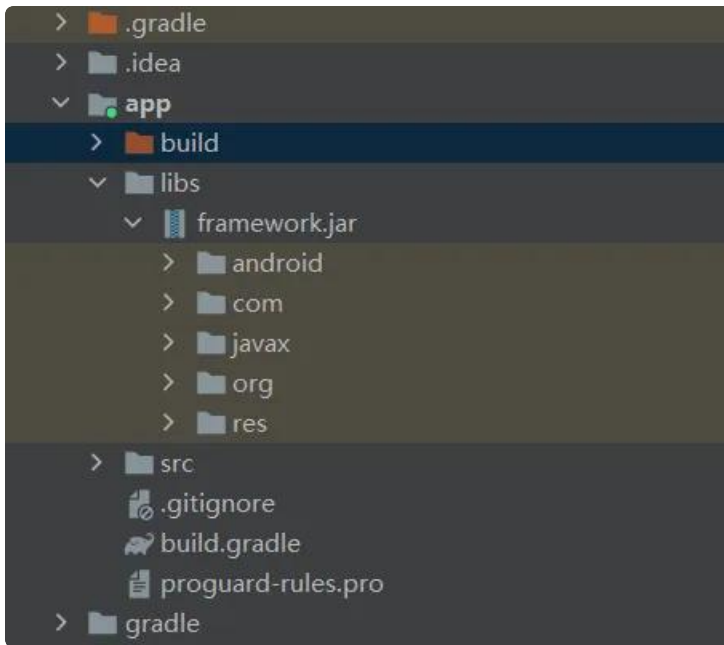
1. 将framework.jar文件添加至app\libs\ 目录



2. 右键“framework.jar”，选择“Add As Library”



导入成功后，framework.jar包下可以看到隐藏的接口列表



3. 导入包后，需要提高jar包的优先级，修改示例如下

修改build.gradle(项目名称下的文件)

```
1  buildscript {
2      repositories {
3          google()
4          mavenCentral()
5      }
6
7      dependencies {
8          classpath "com.android.tools.build:gradle:7.0.4"
9
10         // NOTE: Do not place your application dependencies here; they belong
11         // in the individual module build.gradle files
12     }
13
14     gradle.projectsEvaluated {
15         tasks.withType(JavaCompile){
16             options.compilerArgs.add('-Xbootclasspath/p:app\\libs\\framework.jar')
17         }
18     }
19 }
```

修改build.gradle(:app/build.gradle)

在dependencies下添加 compileOnly files('libs\\framework.jar')



```

1 dependencies {
2     compileOnly files('libs\\framework.jar')
3     implementation 'androidx.appcompat:appcompat:1.2.0'
4     implementation 'com.google.android.material:material:1.3.0'
5     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
6     testImplementation 'junit:junit:4.+'
7     androidTestImplementation 'androidx.test.ext:junit:1.1.2'
8     androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
9 }

```

## 4 Android APK 签名

### 1. 签名文件获取

链接: <https://pan.baidu.com/s/1vOXLd9E7y6G5-YEm6FFPhg?pwd=1234>

提取码: 1234

参数	说明
sbc3566.jks	签名文件
123456	密码
sbc3566	签名文件别名

### 2. Android Studio工程配置签名

a. 修改AndroidManifest.xml, 添加android.uid.system 如下

```

1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     package="com.example.myapplication"
4     android:sharedUserId="android.uid.system">

```

b. 在工程根目录新建一个signature文件夹, 并将sbc3566.jks 文件放入该文件夹下;

c. 修改工程根目录的 app/build.gradle 文件, 添加signingConfigs和配置buildTypes如下:

```
1  android {
2      compileSdk 32
3
4      defaultConfig {
5          applicationId "com.example.myapplication"
6          minSdk 21
7          targetSdk 32
8          versionCode 1
9          versionName "1.0"
10
11         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRu
12         nner"
13     }
14     signingConfigs {
15         release {
16             storeFile file("../signature/sbc3566.jks")
17             storePassword '123456'
18             keyAlias 'sbc3566'
19             keyPassword '123456'
20         }
21
22         debug {
23             storeFile file("../signature/sbc3566.jks")
24             storePassword '123456'
25             keyAlias 'sbc3566'
26             keyPassword '123456'
27         }
28     }
29
30     buildTypes {
31         release {
32             minifyEnabled false
33             proguardFiles getDefaultProguardFile('proguard-android-opt
34             imize.txt'), 'proguard-rules.pro'
35             signingConfig signingConfigs.release
36         }
37         compileOptions {
38             sourceCompatibility JavaVersion.VERSION_1_8
39             targetCompatibility JavaVersion.VERSION_1_8
40         }
41     }
```