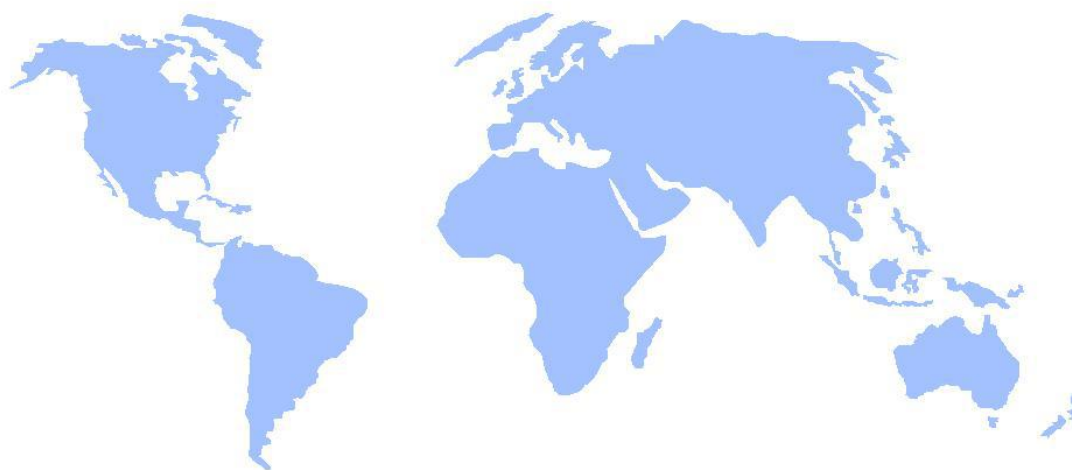




Android 扫描接口说明



www.seuic.com

修订记录

版本号	修订日期	修订内容	修订人员
1.0	2017/03/16	建立本文档	罗涛
1.1	2017/5/31	加入回调函数注意事项说明	罗涛
1.2	2017/11/13	修正关于图像读取接口的描述	罗涛
1.3	2017/11/20	添加新增参数介绍	罗涛

目录

概述.....	3
硬件平台.....	3
软件平台.....	3
适用对象.....	3
准备工作.....	3
Jar 包引用方法.....	3
所需的相关权限.....	6
扫描相关类介绍.....	6
扫描工厂类.....	6
扫描类.....	6
扫描数据回调接口类.....	7
读取图像回调接口类.....	7
扫描状态回调接口类.....	7
扫描数据类.....	8
扫描按键监听类.....	8
扫描接口介绍.....	9
获取扫描头上下文.....	9
打开扫描头.....	10
关闭扫描头.....	10
开始扫描.....	11
停止扫描.....	11
设置扫描条码回调.....	11
启用扫描头.....	12
禁用扫描头.....	12
获取条码参数.....	13
设置条码参数.....	13
获取扫描成功后图像.....	14
开始读取图像.....	14
停止读取图像.....	15
设置读取图像回调.....	15
设置扫描头状态回调.....	16
扫描参数介绍.....	17
条码设置参数.....	17
扫描状态参数.....	20

概述

我司提供的所有设备上已经内置了扫描服务及应用，方便客户定制扫描服务，但是由于有部分客户需要自己开发基于我司扫描SDK的应用，故本文档致力于介绍有关扫描SDK的相关内容，以便帮助用户更为灵活地使用我们的设备。

硬件平台

SDK 适用于我司所有终端设备，具体使用范围在函数接口说明中指明，若无指明表示都支持。

软件平台

SDK 基于 Android 4.3、Android 4.4、Android 5.1 版本，支持 Eclipse、Android Studio 开发工具

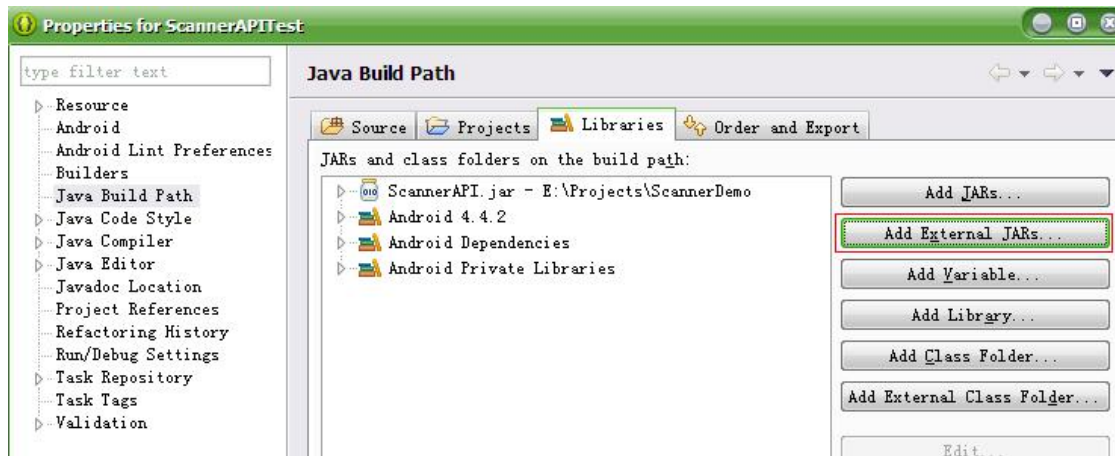
适用对象

希望在自己的应用中定制扫描的开发人员。

准备工作

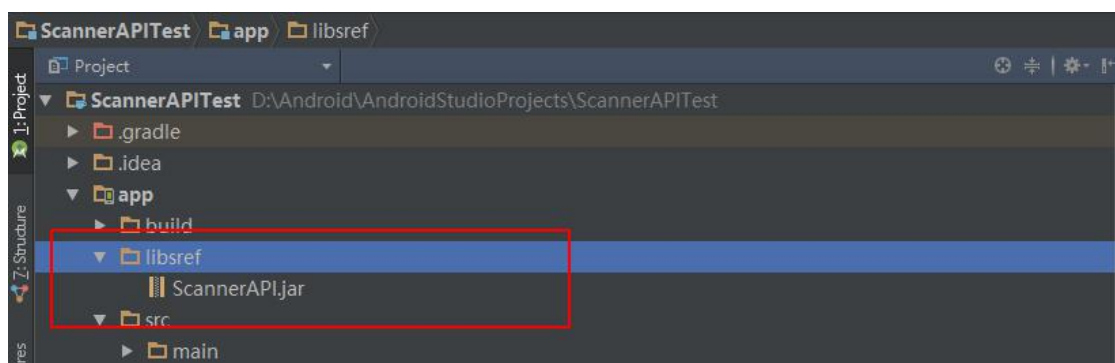
Jar 包引用方法

1、首先以 Eclipse 为例，右击项目 - Properties - Java Build Path - Libraries - Add External JARs，选择 ScannerAPI.jar 文件

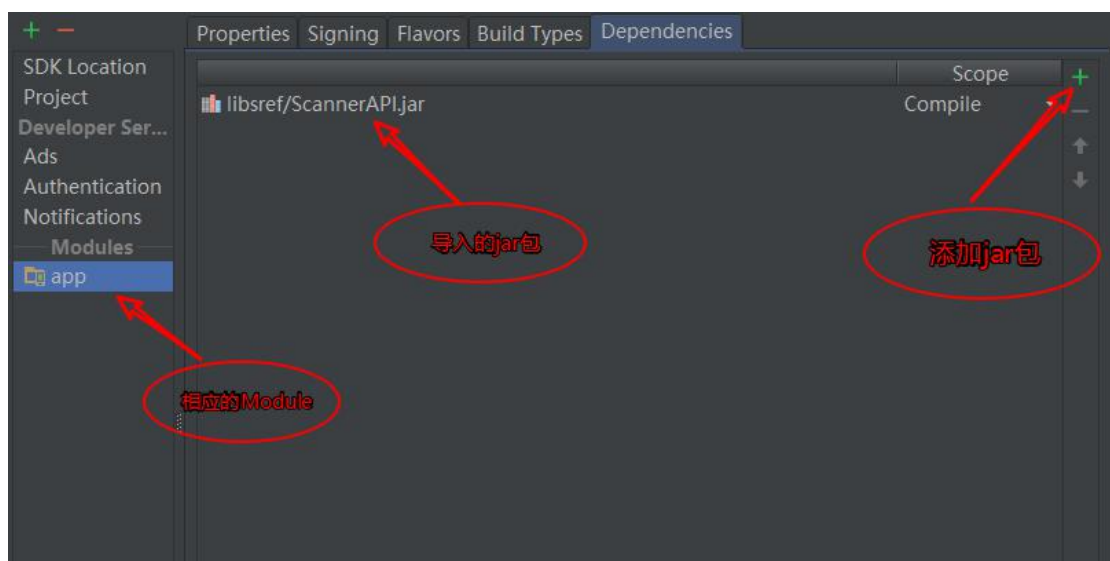
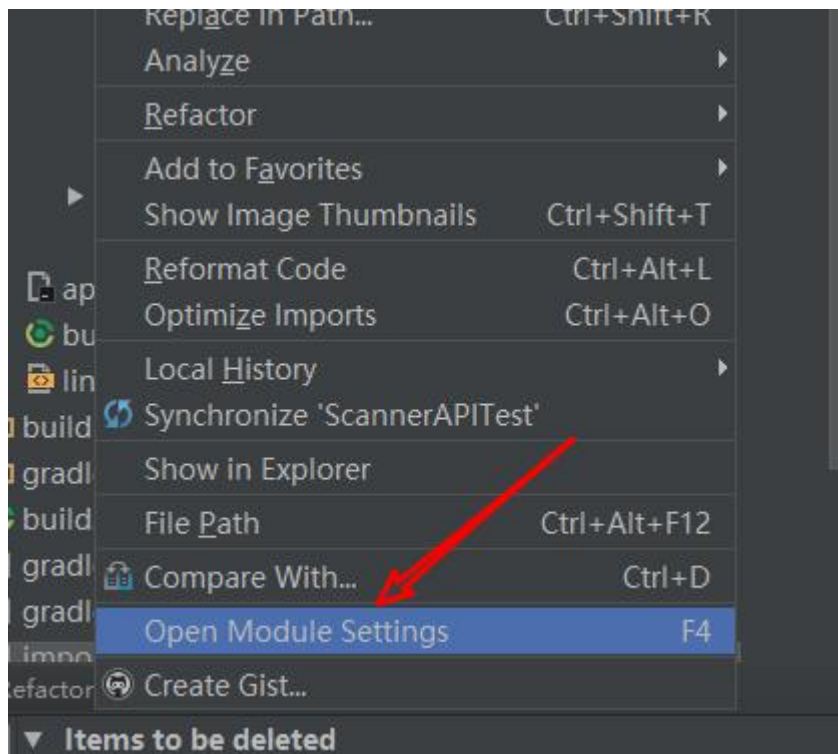


2、以 Android Studio Version: 2.2.2 版本为例，在 project 中，在 app 目录下新建一个 libsref 目录，将 ScannerAPI 放入该目录下，并右击工程选择 Open Module Settings 选中相应的 Module, 切换至最右方的 Dependencies，点击右方加号，选择 File dependency 添加 ScannerAPI.jar，并且修改 scope 为 provided，声明该 ScannerAPI.jar 只在编译时需要依赖，在运行时不需要。

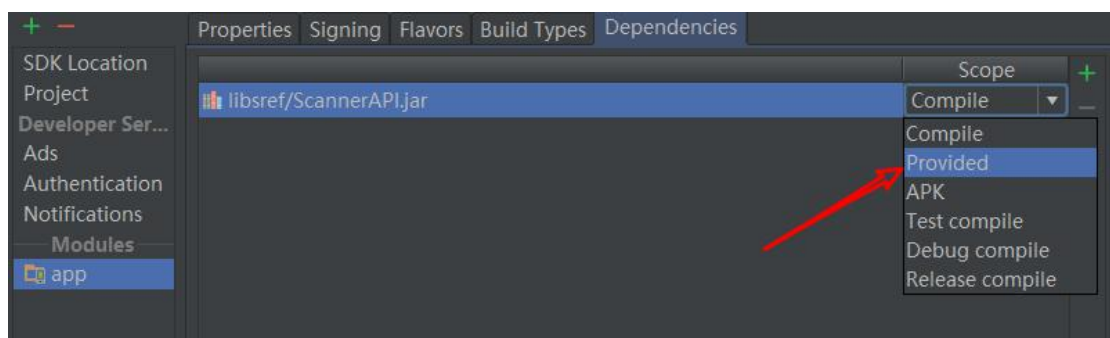
①将 ScannerAPI.jar 放入 app/libsref/下



②右击工程选择 Open Module Settings 选中相应的 Module, 切换至最后方的 Dependencies，点击右方加号，选择 File dependency 添加 ScannerAPI.jar



③更改 compile 为 provide, 声明该 ScannerAPI.jar 只在编译时需要依赖, 在运行时不需要,最后保存设置即可



④，如果设置成功，会在相应的 build.gradle 中有如下 dependencies

```
dependencies {
    provided files('libsref/ScannerAPI.jar')
}
```

所需的相关权限

在进行基于我们的扫描 SDK 开发的时候，需在 AndroidManifest.xml 的 application 节点下添加以下代码

```
<uses-library android:name="android.scanner.library"/>
```

D500/D510/D330S 设备中二维扫描需添加以下权限

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

注意点：

- 1. 在一个应用中只需要打开一次扫描头即可，不需要打开多次，否则会耗时太大，产生卡屏现象。
- 2. 退出程序的时候可以选择关闭扫描头，其余场合不可关闭。
- 3. 可以参照测试 demo 中的方式，在 ScannerService 中定义 action，在使用扫描的 UI 中直接使用动态注册广播接收器接收到扫描解析的结果。

扫描相关类介绍

扫描工厂类

包名	com.seuic.scanner
类名	ScannerFactory
说明	扫描工厂类，主要用于获取扫描头资源

扫描类

包名	com.seuic.scanner
类名	Scanner
说明	扫描主要实现类，定义了一系列扫描操作方法

扫描数据回调接口类

包名	com.seuic.scanner
类名	DecodeInfoCallBack
回调函数	onDecodeComplete(DecodeInfo barcodeInfo)
回调函数参数	param1: 条码信息, DecodeInfo
回调函数返回值	void
说明	扫描回调函数, 用户需引用该接口, 并调用 Scanner 类中 setDecodeInfoCallBack 后可获得条码信息。 注意: 该回调方法中请勿进行 UI 操作!

读取图像回调接口类

包名	com.seuic.scanner
类名	VideoCallBack
回调函数	onVideoCallBack(int width, int height, byte[] img)
回调函数参数	param1: 返回图像宽度 param2: 返回图像高度 param3: 返回的图像数据 (Bitmap 格式)
回调函数返回值	true: 操作成功, 停止出光并终止对图像的读取 false: 操作失败, 继续读取图像
说明	扫描回调函数, 用户需引用该接口, 并调用 Scanner 类中的 setVideoCallBack 后可获得图像数据, 同时每取得一帧图像都会调用该函数, 第三方的相关操作在此处处理。 注意: 该回调方法中请勿进行 UI 操作!

扫描状态回调接口类

包名	com.seuic.scanner
类名	StatusCallBack
回调函数	onStatusCallBack(int status, int info)
回调函数参数	param1: 状态信息, 0: 扫描成功 1: 扫描超时 2: 扫描被取消 3: 扫描出现错误 (暂未使用) 4: 其他事件 (暂未使用)

	param2: 其他额外信息（暂未使用） 扫描状态参数参见： 扫描状态参数
回调函数返回值	void
说明	状态回调函数，用户可引用该接口，并且调用 Scanner 类中的 setStatusCallBack 后可以当扫描头状态更新时获得当前扫描状态。 注意：该回调方法中请勿进行 UI 操作！

扫描数据类

包名	com.seuic.scanner
类名	DecodeInfo
类成员	String barcode: 条码数据 String codetype: 条码类型 int length: 条码长度

扫描按键监听类

包名	com.seuic.scanner
类名	ScannerKey
类成员变量	KEY_DOWN: 扫描按键按下 KEY_UP: 扫描按键抬起
类方法	int open() 打开扫描键监听，返回 1: 打开成功 -1: 打开失败 int getKeyEvent() 获取按键事件, 返回: 1: 扫描键按下, 0: 扫描键抬起, -1: 获取失败 注意: 该方法将阻塞进程，推荐异步调用。 void close() 关闭扫描键监听
说明	扫描按键监听类

示例代码：

```
import com.seuic.scanner.Scanner;
import com.seuic.scanner.ScannerKey;
import com.seuic.scanner.ScannerFactory;
public class ScannerApiTest {
    .....
    Runnable runnable = new Runnable() {
        @Override
        public void run() {
            int ret1 = ScannerKey.open();
            if (ret1 > -1) {
```



```

        while (true) {
            int ret = ScannerKey.getKeyEvent();
            if (ret > -1) {
                switch (ret) {
                    case ScannerKey.KEY_DOWN:
                        scanner.startScan();
                        break;
                    case ScannerKey.KEY_UP:
                        scanner.stopScan();
                        break;
                }
            }
        }
    }
};
.....
ScannerKey.close();
}

```

扫描接口介绍

接口具体使用范围若无指明表示所有扫描头都支持。

获取扫描头上下文

引用包名	com.seuic.scanner
引用类	ScannerFactory
接口	getScanner
接口参数	Context context
返回值	Scanner 实例
说明	获取扫描类实例

示例代码：

```

import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest {
    .....
    Scanner scanner = ScannerFactory.getScanner(context);
    .....
}

```

```
}
```

打开扫描头

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	open
接口参数	void
接口返回值	boolean
说明	返回 true ，打开扫描头成功，返回 false ，打开扫描头失败。 注意：仅在该方法返回为 true 时，才可进行扫描头其他操作

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest {
    .....
    Scanner scanner = ScannerFactory.getScanner(context);
    Boolean status = scanner.open();
    .....
}
```

关闭扫描头

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	close
接口参数	void
接口返回值	void
说明	释放扫描头资源

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;

public class ScannerApiTest {
    .....
    Scanner scanner = ScannerFactory.getScanner(context);
    Boolean status = scanner.close ();
    .....
}
```

开始扫描

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	startScan
接口参数	void
接口返回值	void
说明	开始扫描： 扫描头出光进入条码读取模式

停止扫描

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	stopScan
接口参数	void
接口返回值	void
说明	停止扫描，可使扫描头灭光并终止条码读取

设置扫描条码回调

引用包名	com.seuic.scanner
引用类名	Scanner DecodeInfoCallBack DecodeInfo
接口名称	setDecodeInfoCallBack
接口参数	param1: DecodeInfoCallBack
接口返回值	void
说明	设置用户扫描结束后的回调函数，该函数是获取包括条码数据、长度、类型的信息。 在设置之前确保实现了 DecodeInfoCallBack 中的 onDecodeComplete 方法

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
import com.seuic.scanner.DecodeInfoCallBack;
import com.seuic.scanner.DecodeInfo;
```

```
public class ScannerApiTest implements DecodeInfoCallBack {
    .....
    Scanner scanner = ScannerFactory.getScanner(context);
    scanner.setDecodeInfoCallBack(this);
    scanner.startScan();

    .....
    scanner.stopScan();

    @Override
    public void onDecodeComplete(DecodeInfo info) {
        .....
    }
    .....
}
```

启用扫描头

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	enable
接口参数	void
接口返回值	void
说明	启用扫描头

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest {
    .....
    Scanner scanner = ScannerFactory.getScanner(context);
    scanner.enable();
    .....
}
```

禁用扫描头

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	disable
接口参数	void
接口返回值	void

说明	禁用扫描头，该操作完成后，将无法进行扫描操作
----	------------------------

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest {
    .....
    Scanner scanner = ScannerFactory.getScanner(context);
    scanner.disable();
    .....
}
```

获取条码参数

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	getParams
接口参数	param1：参数代码 例如：获取 Scan Angle 的值，条码参数便是 0x02，具体参数代码参见： 条码设置参数
接口返回值	int 类型，条码参数当前值
说明	根据参数代码获取该参数当前的设定值

设置条码参数

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	setParams
接口参数	param1：参数代码 例如：设置 Scan Angle 的值，条码参数便是 0x02 param2：需要设置的参数值 参数代码参见： 条码设置参数
接口返回值	boolean： true 设置成功 false 设置失败
说明	根据参数代码设置扫描参数值

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest {
    .....
```

```

Scanner scanner = ScannerFactory.getScanner(context);
int value = scanner.getParams(0x02);

.....

scanner.setParams(0x02, 1);
}

```

获取扫描成功后图像

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	getLastImage
接口参数	void
接口返回值	byte[] :图像数据, width:832 height:640,图像格式为 bitmap
说明	当完成一次成功的扫描后, 调用该接口, 获取该次用于解码的扫描图像 注意: 该接口仅适用于A9/A-PAD/A6L-P/CRUISE中的S8

示例代码:

```

import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest {

    .....

    Scanner scanner = ScannerFactory.getScanner(context);
    scanner.startScan ();

    .....

    byte[] image = scanner.getLastImage();
}

```

开始读取图像

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	startVideo
接口参数	param1: int 类型, 设置超时时间, 单位为: ms
接口返回值	0: 回调函数 onVideoCallback 返回成功自动退出无需调用 stopVide: 9: 参数无效退出 17: 调用stopVideo手动退出 6: 取图像失败退出

	25: 超时退出 2: 未知错误退出
说明	开始读取图像：扫描头开始出光并进入图像读取模式，为阻塞函数，且与startScan不可同时使用。 注意：该接口仅适用于A9/A-PAD/A6L-P/CRUISE中的S8

停止读取图像

引用包名	com.seuic.scanner
引用类	Scanner
接口名称	stopVideo
接口参数	void
接口返回值	void
说明	扫描头停止出光并终止对图像地读取。 注意：该接口仅适用于A9/A-PAD/A6L-P/CRUISE中的S8

设置读取图像回调

引用包名	com.seuic.scanner
引用类名	Scanner VideoCallBack
接口名称	setVideoCallBack
接口参数	param1:VideoCallBack
接口返回值	void
说明	设置用户扫描进入读取图像模式后的回调函数，该接口仅适用于A9/A-PAD/A6L-P/CRUISE 中的 S8

示例代码：

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
import com.seuic.scanner.VideoCallBack;

public class ScannerApiTest implements VideoCallBack {
    .....

    Scanner scanner = ScannerFactory.getScanner(context);
    scanner.setVideoCallBack(this);
    scanner.startVideo(3000);

    .....

    scanner.stopVideo();
}
```

```
@Override
public boolean onVideoCallBack(int width, int height, byte[] img)
{
    .....
}
.....
}
```

设置扫描头状态回调

引用包名	com.seuic.scanner
引用类名	Scanner StatusCallBack
接口名称	setStatusCallBack
接口参数	param1: StatusCallBack 接口类
接口返回值	void
说明	设置用户获取当前扫描状态的回调函数，在设置之前务必实现 StatusCallBack 中的 onStatusCallBack 方法, 设置之后有关扫描的状态会自动回调 onStatusCallBack 将当前状态反馈给用户

示例代码:

```
import com.seuic.scanner.ScannerFactory;
import com.seuic.scanner.Scanner;
public class ScannerApiTest implements StatusCallBack {
    .....

    Scanner scanner = ScannerFactory.getScanner(context);
    scanner.setStatusCallBack(this);

    @Override
    public boolean onStatusCallBack(int status, int info) {
        .....
    }
    .....
}
```


扫描参数介绍

条码设置参数

Parameter	Parameter Number (Hex)
Laser On Time	0x01
Scan Angle	0x02
Time Out Between Same Symbol	0x03
Linear Code Type Security Levels	0x04
Bi-directional Redundancy	0x06
Aimer/Illumination lighting combinations	0x07
Mobile Phone/Display Mode	0x08
Illumination Power Level	0x09
Decode Time Out	0x0a
Adaptive Scanning	0x0b
Continue Scan Mode	0x0c
Exposure Mode	0x0d
Picklist Mode	0x0e
Aim Accuracy Level	0x0f
GS1 AI Separator	0x10
Transfer Aim ID	0x11
Inverse Mode	0x12
Low Power Delay Time	0x13
Hold Scanning Time Out	0x14
UPC-A	0x101
UPC-E	0x102
UPC-E1	0x103
EAN-8	0x104
EAN-13	0x105
Bookland EAN	0x106
Transmit UPC-A Check Digit	0x107
Transmit UPC-E Check Digit	0x108
Transmit UPC-E1 Check Digit	0x109
Transmit EAN-13 Check Digit	0x10a
Transmit EAN-8 Check Digit	0x10b
UPC-A Preamble	0x10c
UPC-E Preamble	0x10d
UPC-E1 Preamble	0x10e
Convert UPC-E to A	0x10f
Convert UPC-E1 to A	0x110
Convert EAN-8 to 13	0x111

UCC Coupon Extended Code	0x112
UPC-E Expand	0x113
UPC-A 2 Char Addenda	0x114
UPC-A 5 Char Addenda	0x115
UPC-E 2 Char Addenda	0x116
UPC-E 5 Char Addenda	0x117
EAN13 2 Char Addenda	0x118
EAN13 5 Char Addenda	0x119
EAN8 2 Char Addenda	0x11a
EAN8 5 Char Addenda	0x11b
Code 128	0x121
Code 128 Minimum Length	0x122
Code 128 Maximum Length	0x123
ISBT 128	0x124
GS1-128	0x125
ISBT 128 Predefined Concatenation Sequences On/Off	0x126
Code 39	0x131
Trioptic Code 39	0x132
Code 39 Full ASCII	0x133
Convert Code 39 to Code 32	0x134
Code 32 Prefix	0x135
Code 39 Minimum Length	0x136
Code 39 Maximum Length	0x137
Transmit Code 39 Check Digit	0x138
Code 39 Check Digit Verification	0x139
Transmit Code 39 Start/Stop Characters	0x13a
Code 39 Quiet Zone	0x13b
Code 93	0x141
Code 93 Minimum Length	0x142
Code 93 Maximum Length	0x143
Code 11	0x151
Code 11 Minimum Length	0x152
Code 11 Maximum Length	0x153
Code 11 Check Digit Verification	0x154
Transmit Code 11 Check Digit	0x155
Interleaved 2 of 5	0x161
Interleaved 2 of 5 Minimum Length	0x162
Interleaved 2 of 5 Maximum Length	0x163
Discrete 2 of 5	0x171
Discrete 2 of 5 Minimum Length	0x172
Discrete 2 of 5 Maximum Length	0x173
Chinese 2 of 5	0x181

Chinese 2 of 5 Minimum Length	0x182
Chinese 2 of 5 Maximum Length	0x183
Codabar	0x191
Codabar Minimum Length	0x192
Codabar Maximum Length	0x193
Transmit Codabar Start/Stop Characters	0x194
MSI	0x201
MSI Minimum Length	0x202
MSI Maximum Length	0x203
RSS 14	0x211
RSS limit	0x212
RSS expanded	0x213
Convert RSS to UPC/EAN	0x214
QR Code	0x221
QR Code Minimum Length	0x222
QR Code Maximum Length	0x223
QR Inverse	0x224
MicroQR	0x225
Data Matrix	0x231
Data Matrix Minimum Length	0x232
Data Matrix Maximum Length	0x233
Data Matrix Inverse	0x234
Data Matrix Decode Mirror Images	0x235
China Post	0x241
China Post Minimum Length	0x242
China Post Maximum Length	0x243
Matrix 2 of 5	0x251
Matrix 2 of 5 Minimum Length	0x252
Matrix 2 of 5 Maximum Length	0x253
US Postnet	0x261
US Planet	0x262
British Post	0x263
Japanese Post	0x264
Australian Post	0x265
UPU FICS Postal	0x266
USPS 4CB/One Code/Intelligent Mail	0x267
KIX(Netherlands) Post	0x268
Transmit US Postal Check Digit	0x269
Transmit Planet Check Digit	0x26a
Korean 2 of 5	0x271
EAN UCC Composite	0x2b1
EAN UCC Composite Minimum Length	0x2b2
EAN UCC Composite Maximum Length	0x2b3

PDF417	0x2c1
PDF417 Minimum Length	0x2c2
PDF417 Maximum Length	0x2c3
Micro PDF417	0x2d1
Micro PDF417 Minimum Length	0x2d2
Micro PDF417 Maximum Length	0x2d3
MaxiCode	0x2e1
MaxiCode Minimum Length	0x2e2
MaxiCode Maximum Length	0x2e3
Aztec Code	0x2f1
Aztec Code Minimum Length	0x2f2
Aztec Code Maximum Length	0x2f3
Aztec Inverse	0x2f4
Codablock	0x301
Codablock Minimum Length	0x302
Codablock Maximum Length	0x303
Telepen	0x311
Telepen Minimum Length	0x312
Telepen Maximum Length	0x313
Telepen Old Syle Mode	0x314
Set Default Parameter	0x321
Set Code Charset	0x322
OCR	0x331
Set OCR Template	0x332
Composite CC-C	0x341
Composite CC-A/B	0x342
Composite TLC-39	0x343
UPC Composite	0x344
Transmit Code ID Character	0x351
Grid Matrix	0x361

各扫描头参数详细信息见：[扫描头条码参数汇总\(仅供内部使用\)](#)

扫描状态参数

Status	Status Number(Hex)
Decode Complete	0x00
Decode Time Out	0x01
Decode has been Canceled	0x02
Some Errors Happened	0x03
Decode Event	0x04