

# SunmiScanner 开发及使用文档

# 文档更新说明

编号	更新日期	组件版本	更新内容	撰写人
1.0.0	2018/04/24	v1.1.6	原始版本	Darren、徐贇庭
1.0.1	2018/06/07	v1.1.19	增加获取扫码头类型接口	Darren、Arthur
1.0.2	2018/11/23	v2.1.x	增加码制详细设置 增加前后缀设置 增加高级格式设置 增加虚拟按键设置 增加清除配置菜单 增加无直接输出 增加输出码类型 ID 修正部分描述	Darren、Arthur
1.0.3	2019/03/09	v2.3.x	添加前后缀支持“\xxx”转义 添加高级格式支持“\xxx”转义 添加广播输出支持原始 byte 数组 修改部分描述与图片说明	Darren、Arthur

# 1. 简介

商米 L2、P2Lite 等扫码专用设备，可用于商超，工业，医疗，农贸，执法等，L2 有两种扫码头的类型：



NewLand：新大陆扫码头，支持编码见附表 1；



Zebra：斑马扫码头，支持编码见附表 1；

Honeywell：霍尼韦尔，支持编码见附表 1；

扫码头是一个免开发的扫码设备，默认侧边按键触发扫码，有三种扫码结果的输出，默认为模拟键盘输出。

用户打开任意一个编辑框，在扫码成功后，会自动在编辑框输入扫码结果。

如果用户需要软件触发扫码或者需要自定义扫码按键，可以通过扫码服务提供的接口来设置。

下面介绍一下与扫码服务相关的接口文档说明（目前支持 aidl 的方式与服务连接）。

## 2. 连接扫码服务（AIDL）

### 2.1. AIDL 简介

AIDL 是 Android Interface Definition language 的缩写，它是一种 Android 内部进程通信接口的描述语言，通过它我们可以定义进程间的通信接口。

### 2.2. AIDL 使用

建立连接可分以下 5 步骤：

1. 在项目中添加资源文件中附带的 [AIDL 文件](#)。
2. 在控制扫码的代码类中实现 ServiceConnection。
3. 调用 ApplicationContext.bindService()，并在 ServiceConnection 实现中进行传递。注意：  
bindservice 是非阻塞调用，意味着调用完成后并没有立即绑定成功，必须以 serviceConnected 为准。
4. 在 ServiceConnection.onServiceConnected()实现中，你会接收一个 IBinder 实例(被调用的 Service)。调用 IScanInterface.Stub.asInterface(service)将参数转换为 IScanInterface 类型。
5. 现在就可以调用 IScanInterface 接口中定义的方法了。

绑定服务示例：

```
private static ServiceConnection conn = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service) {
        scanInterface = IScanInterface.Stub.asInterface(service);
        Log.i("setting", "Scanner Service Connected!");
    }

    @Override
    public void onServiceDisconnected(ComponentName name) {
        Log.e("setting", "Scanner Service Disconnected!");
        scanInterface = null;
    }
};

public void bindScannerService() {
    Intent intent = new Intent();
    intent.setPackage("com.sunmi.scanner");
```

```
intent.setAction("com.sunmi.scanner.IScanInterface");  
bindService(intent, conn, Service.BIND_AUTO_CREATE);  
}
```

## 2.3. AIDL 接口定义说明

编号	方法
1	void <b>sendKeyEvent</b> (KeyEvent key) 自定义按键触发扫码
2	void <b>scan</b> ( ) 触发开始扫码
3	void <b>stop</b> ( ) 触发停止扫码
4	int <b>getScannerModel</b> () 获取扫码头类型

### 1. 自定义按键触发扫码

**函数** : void **sendKeyEvent**(KeyEvent key)

**参数** :

key → KeyEvent 事件

对应 key 的 Action 为 KeyEvent.ACTION\_UP 时 : 触发开始扫码

对应 key 的 Action 为 KeyEvent.ACTION\_DOWN 时 : 触发停止扫码

**示例** :

```
@Override
public boolean dispatchKeyEvent(KeyEvent event) {
    // 示例 : 以 x 键值作为触发扫码按键
    if (event.getKeyCode() == x) {
        scanInterface.sendKeyEvent(event);
    }
    return super.dispatchKeyEvent(event);
}
```

### 2. 触发开始扫码

**函数** : void **scan**( )

**备注** : 需要与 **stop()**方法配合使用 , 开始识别扫码。

**示例** :

```
scanInterface.scan();
```

### 3. 触发停止扫码

**函数：**void **stop()**

**备注：**需要与 **scan()**方法配合使用，停止识别扫码。

**示例：**

```
scanInterface.stop();
```

### 4. 获取扫码头类型

**函数：**int **getScannerModel()**

**返回值：**类型

100 → NONE

101 → P2Lite/V2Pro/P2Pro(em1365/BSM1825)

102 → L2-newland(EM2096)

103 → L2-zabra(SE4710)

104 → L2-HoneyWell(N3601)

105 → L2-HoneyWell(N6603)

106 → L2-Zabra(SE4750)

107 → L2-Zabra(EM1350)

**示例：**

```
scanInterface.getScannerModel();
```

### 3. 扫码头设置

可以从“系统设置”中，找到“扫码头设置”进入扫码头详细设置页面



#### 3.1. 清除配置

点击右上角的菜单图标，然后点击“清除配置”，如下图：



点击“清除配置”可以将配置还原成扫码头的默认配置，如设置云端配置，将生效云端配置。

#### 3.2. 输出编码设置

默认 UTF-8

可选 UTF-8, GBK, ISO-8859-1, SHITF-JIS

输出编码设置

UTF-8	<input checked="" type="radio"/>
GBK	<input type="radio"/>
ISO-8859-1	<input type="radio"/>
SHITF-JIS	<input type="radio"/>

取消
确定

### 3.3. 提示方式设置

默认打开声音提示和震动提示

提示方式设置

声音提示	<input checked="" type="checkbox"/>
震动提示	<input type="checkbox"/>

取消
确定

### 3.4. 输出方式设置

#### 1. 设置

默认选中模拟键盘方式输出，默认自动补回车

默认开启广播输出

反回码类型默认不输出，相关请参考 [CodeID](#)



直接填充输出，可选择字符转按键（将结束符以模拟按键的方式输出）



## 2. 广播输出说明

监听广播：“com.sunmi.scanner.ACTION\_DATA\_CODE\_RECEIVED”

字段说明：

data：字符数据；

source\_byte：byte 数组原始数据（不含结束符、CodeID、前后缀与高级设置等基础设置，需要 2.3.1 以上版本）；

示例：

```
private static final String ACTION_DATA_CODE_RECEIVED =
"com.sunmi.scanner.ACTION_DATA_CODE_RECEIVED";
private static final String DATA = "data";
private static final String SOURCE = "source_byte";

private BroadcastReceiver receiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String code = intent.getStringExtra(DATA);
        String arr = intent.getByteArrayExtra(SOURCE);
        if (code != null && !code.isEmpty()) {
            mCode.setText(code);
        }
    }
};

private void registerReceiver() {
    IntentFilter filter = new IntentFilter();
    filter.addAction(ACTION_DATA_CODE_RECEIVED);
    registerReceiver(receiver, filter);
}
```

### 3.5. 虚拟按键设置

默认悬浮按键是关的



### 3.6. 触发模式设置

默认短按触发扫码，松开按键停止扫码。



### 3.7. 前后缀设置

前后缀功能默认是关闭的，支持“\n”、“\t”、“\xxx”(三位 ASCII 的十进制值 0-255，需要 2.2.2 以上版本)特殊字符，如需添加“\”请输入“\\”：

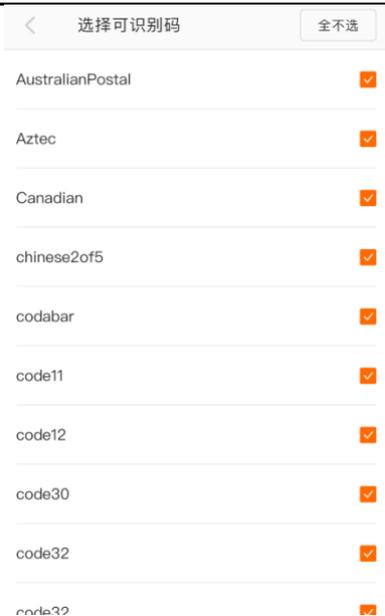


可以手动开启，最长支持 10 位



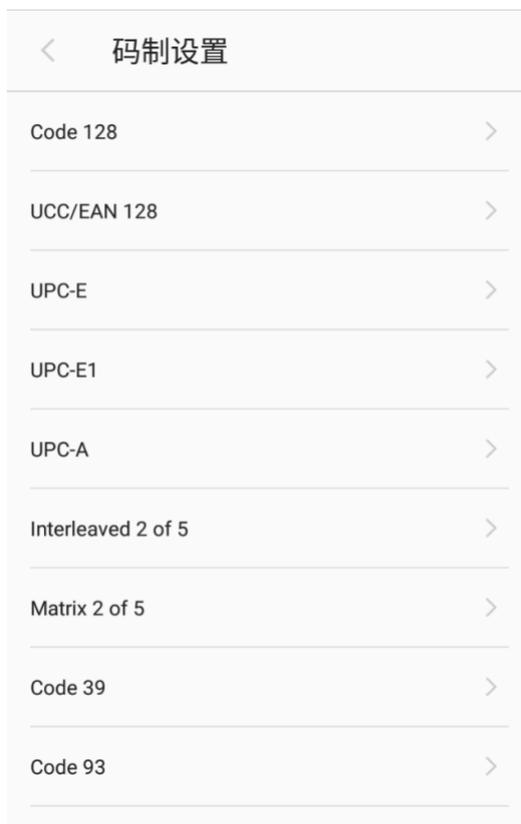
### 3.8. 识别码制设置

用户可在设置界面中选择开启或关闭制定识别码，默认开启全部识别码。



### 3.9. 码制详细配置

从扫码头设置中找到“码制设置-设置解码器参数”进入码制设置页面，如下：



点击其中的一个码制，如“Code 39”，如下图：

Code 39

最小长度

2

最大长度

55

使能校验

---

支持Full ASCII

---

可识读TRIOPTIC CODE 39

---

结果扩展为Code 32

取消
确定

### 3. 10. 高级格式设置

高级格式设置-字符替换规则，如下图：

<
高级格式设置

高级格式设置

已禁用

规则 (1/10)

新增

规则1

✎ 编辑
🗑 删除

1

高级格式设置，默认关闭，可以手动开启，最多可支持 10 条规则，可对规则进行“编辑”与“删除”操作，点击

“新增”可添加，如下图：

&lt; 新增

保存

将扫码结果中的目标字符替换成设定的字符，如：扫码结果为“1234356”，希望把结果中的“3”替换为“ab”，以达到最终输出的结果为“12ab4ab56”，则需要目标字符中输入3，替换字符中输入ab；目标字符支持“\”+ASCII码(三位十进制)的形式，如需把“组群分隔符”替换为“ab”，可以在目标字符中输入“\029”，在替换字符中输入“ab”

目标字符

 0/20

替换字符

 0/20

最长可支持 20 个字符，支持“\n”、“\t”、“\xxx”(三位 ASCII 的十进制值 0-255，需要 2.2.5 以上版本)特殊字符，如需添加“\”请输入“\\”。

## 4. 附录

### 4.1. 附表 1

码制	Newland (2096)	Zebra (1350)	Zebra (4710/4750)	Fp(1825) NL(1365)	Honeywell ( 3601/6603 )
Code128	√	√	√	√	√
UCC•EAN12 ( GS1 )	√	√	√	√	√
ISBT 128		√	√		√
EAN8	√	√	√	√	√
EAN13	√	√	√	√	√
UPC-E	√	√	√	√	√
UPC-E1		√	√		√
UPC-A	√	√	√	√	√
Interleaved 2 of 5 ( ITF )	√	√	√	√	√
Matrix 2 of 5	√	√	√	√	√
Code39	√	√	√	√	√
Codabar	√	√	√	√	√
Code93	√	√	√	√	√
GS1 DataBar(RSS)	√	√	√		√
Composite-UCC	√				√
Composite-UPC	√				√
Code11	√	√	√	√	√
ISBN	√	√	√	√	
Industrial 2 of 5	√			√	√
Standard 2 of 5(IATA)	√			√	√
Discrete 2 of 5 ( DTF )		√	√		

Chinese 2 of 5		√	√		
Korea 3 of 5		√	√		
Plessey	√			√	
MIS-Plessey(MSI)	√	√	√	√	√
Composite A/B			√		
Composite C			√		
ISSN EAN		√	√	√	
PDF417	√		√		√
QR Code	√		√		√
Aztec			√		√
DataMatrix	√		√		√
汉信码	√		√		√
MaxiCode			√		√
AustralinPostal			√		√
US Postnet			√		√
US Planet			√		√
Uk Postal			√		√
Japan Postal			√		√
Canadian Post					√
KIX(Netherlands) Post					√
Korea Post					√
China Post(HongKong 2of5)					√
Telepen					√
Grid Matrix					√
Codablock					√
Code 32					√
Coupon Code					√

TLC 39					√
--------	--	--	--	--	---