

# iScan Pro 开发说明文档

当前版本： v1.3

版本状态： 已发布

## 文档更新说明

文档版本	更新时间	适配版本	更新内容	参与人
v1.0	2020/07/23	V1.0	初始内容	王航
v1.1	2020/11/12	V1.1	1、更新接口	林振宇/董文斌/ 鲁泉峰 /王航
V1.2	2020/12/29	V1.2	1、更新接口定义	鲁泉峰/王航
V1.3	2022/05/16	V1.3	1、更正部分接口参数 说明	林振宇

# 目录

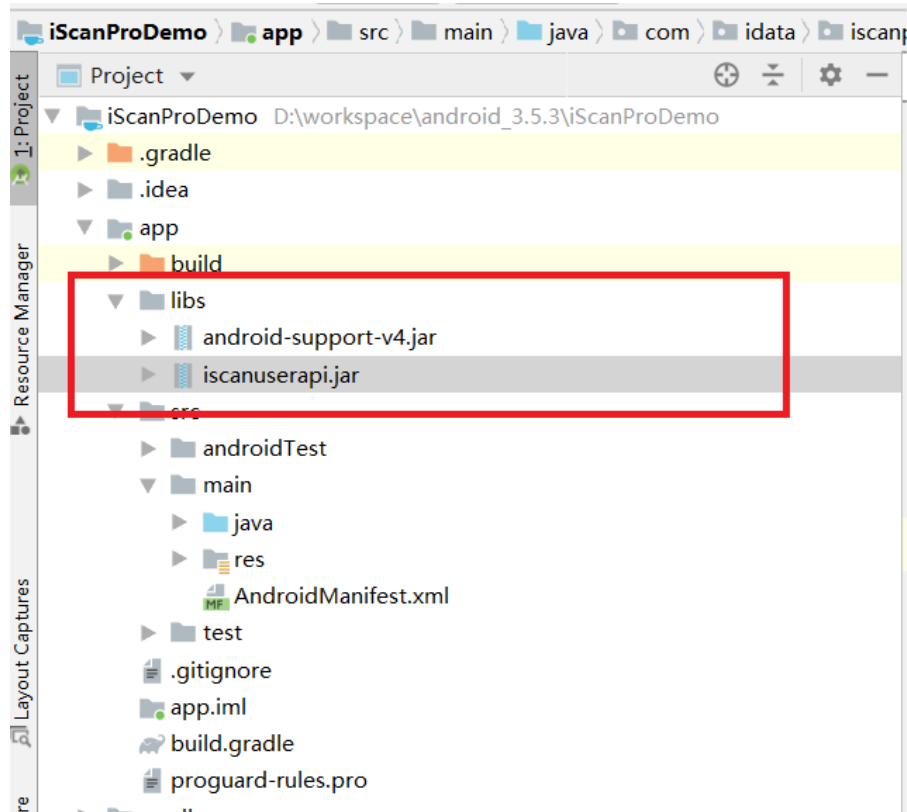
1、文档说明 .....	3
2、开发说明 .....	3
3、接口列表 .....	4
3.1 开启扫描 .....	4
3.2 关闭扫描 .....	4
3.3 开始扫码 .....	5
3.4 停止扫码 .....	5
3.5 禁用扫描按键 .....	5
3.6 配置扫描结果输出方式 .....	5
3.7 配置扫描提示方式 .....	6
3.8 配置附加按键 .....	6
3.9 配置条码数据处理规则 .....	6
3.10 配置触发方式 .....	7
3.11 配置扫描超时时间 .....	7
3.12 配置连续扫描间隔 .....	7
3.13 设置字符编码格式 .....	8
3.14 是否删除编辑框已有内容 .....	8
3.15 恢复默认配置参数 .....	8
3.16 配置解码区域模式 .....	8
3.17 设置激光补光模式 .....	8
3.18 配置条码控制开关 .....	9
3.19 图片输出模式 .....	9
3.20 注册扫描结果监听对象 .....	10
3.21 注销扫描结果监听对象 .....	10

## 1、文档说明

本文档主要说明 iData 终端设备扫描条码的开发说明，供相关行业以及集成方的相关开发人员参考。

## 2、开发说明

- 1) 请将 libs 目录的 “iscanuserapi.jar” 拷贝到工程的 libs 目录，如下图：



- 2) 将 “iscanuserapi.jar” 加入工程编译路径，添加后 “build.gradle” 文件如下：

```
dependencies {  
    implementation fileTree(include: ['*.jar'], dir: 'libs')  
    implementation 'androidx.appcompat:appcompat:1.0.2'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'  
    implementation files('libs/iscanuserapi.jar')  
}
```

- 3) 在 Context 中的 onCreate 方法中初始化 miScanInterface。
- 4) 初始化完成调用 miScanInterface 方法进行扫描操作。

```

//定义接口操作对象
private iScanInterface miScanInterface ;

//定义数据回调接口
private IScanListener scanResltListener = new IScanListener() {
    /**
     * @param data 条码数据
     * @param type 条码类型
     * @param decodeTime 解码耗时
     * @param keyDownTime 按键时间
     * @param imagePath 图片路径，注意：默认图片输出为关闭状态；
     *                                     如果需要条码图片需要打开图片输出选项。
     */
    @Override
    public void onScanResults(String data, int type, long decodeTime, long keyDownTime, String imagePath) {
        Log.d(TAG, msg: "onScanResults: data="+data);
        Log.d(TAG, msg: "onScanResults: type="+type);
        Log.d(TAG, msg: "onScanResults: decodeTime="+decodeTime);
        Log.d(TAG, msg: "onScanResults: keyDownTime="+keyDownTime);
        Log.d(TAG, msg: "onScanResults: imagePath="+imagePath);
    }
};

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //初始化接口对象
    miScanInterface = new iScanInterface(context: this);
    //注册数据回调接口
    miScanInterface.registerScan(scanResltListener);
}

```

## 5) 应用退出销毁

```

@Override
protected void onDestroy() {
    super.onDestroy();
    //应用关闭后注销回调
    miScanInterface.unregisterScan(scanResltListener);
}

```

## 3、接口列表

### 3.1 开启扫描

**void open()**

说明：给扫描头上电，上电后才能进行扫描，扫描前先执行此操作再启动扫描。

输入参数：空

返回值：空

注意：该接口不可频繁调用，为省电需要一般在程序开启时调用一次，或者不调用交给 iScanPro 自动控制，频繁开关串口容易导致扫描引擎卡死。

### 3.2 关闭扫描

**void close()**

说明：关闭扫描头，关闭后扫描头将停止工作，不再响应解码请求。

输入参数：空

返回值：空

注意：默认情况下系统会自动进行扫描头管理，若无特殊情况，建议应用开发方不要调用该接口。

### 3.3 开始扫码

`void scan_start()`

说明：开始扫码，调用后扫描头会出光并识别条码。

输入参数：空

返回值：空

### 3.4 停止扫码

`void scan_stop()`

说明：停止扫描，此操作执行后，扫描头光会灭掉

输入参数：空

返回值：空

### 3.5 禁用扫描按键

`void lockScanKey(boolean enable)`

说明：禁用扫描按键，禁用后按扫描键将不再响应扫描请求。

输入参数：enable，扫描按键禁用状态。

false：禁用扫描按键，禁用后用户可以自行开发定义扫描按键功能。

true：解除禁用，解除后按扫描按键可以自动触发扫码功能。

返回值：空

注意：应用禁用后，在应用退出时请解除禁用，否则会导致无法扫描的问题。

### 3.6 配置扫描结果输出方式

`void setOutputMode(int mode)`

说明：配置扫描结果输出模式，输出方式说明见“输出方式”章节。

输入参数：mode，输出方式，默认：0，支持如下值：

0：扫描结果直接发送到焦点编辑框内

1：扫描结果以广播模式发送，当选择广播模式时，可以通过Android 广播机制监听扫描结果，扫描结果默认广播定义如下：

action: **android.intent.action.SCANRESULT**

extras: **value**，识别结果，识别失败时 value 值为空。

length, 条码长度。

2：扫描结果以模拟按键模式发送

返回值：空

### 3.7 配置扫描提示方式

**void enablePlayBeep(boolean enable)**

说明：配置扫描成功后是否播放声音

输入参数：enable，扫描成功后声音播放状态

true：扫描成功后，播放声音

false：扫描成功后，不播放声音

返回值：空

**void enableFailurePlayBeep(boolean enable)**

说明：扫描失败后是否播放声音

输入参数：enable，扫描失败后提示音状态

true：扫描失败后，允许播放声音

false：扫描失败后，不播放声音

返回值：空

**void enablePlayVibrate(boolean enable)**

说明：扫描成功后是否振动

输入参数：enable，扫描成功后振动提示状态

true：扫描成功后，振动提示

false：扫描成功后，不振动

返回值：空

**void lightSet(boolean enable)**

说明：配置扫描指示灯状态

输入参数：enable，扫描指示灯状态

true：扫描过程开启指示灯

false：扫描过程关闭指示灯

返回值：空

### 3.8 配置附加按键

**void enableAddKeyValue(int value)**

说明：将指定按键的键值附加到扫描结果后。

输入参数：value，参数值，附加值类型。支持的值如下：

0：不附加内容

1：附加回车键

2：附加 TAB 键

3：附加换行符(\n)

返回值：空

### 3.9 配置条码数据处理规则

**void addPrefix(String text)**

描述：扫描成功后，为扫描结果字符串添加前缀

输入参数: text, 扫描结果前添加的前缀字符

返回值: 空

**void addSuffix(String text)**

描述: 扫描成功后, 为扫描结果字符串添加后缀

输入参数: text, 扫描结果添加的后缀

返回值: 空

**void filterCharacter(String text)**

描述: 过滤特定字符

输入参数: text, 需要过滤的字符

返回值: 空

### 3.10 配置触发方式

**void continuousScan(boolean enable)**

描述: 配置连续扫描状态

输入参数: enable, 连扫状态

true: 开启连续扫描,

说明: 设置打开连扫之后, 需要点击扫描按键才能正常连扫。

false: 关闭连续扫描。

返回值: 空

**void effortScan(boolean enable)**

描述: 省力模式, 按下按键直到解到条码或者扫描超时才会停止扫描

输入参数: enable, 省力模式状态

true: 开启省力模式

false: 关闭省力模式

返回值: 空

### 3.11 配置扫描超时时间

**void setTimeout(int value)**

描述: 设置超时时间, 超过指定时间后自动停止扫描

输入参数: value, 连续扫描间隔时间, 时间单位: ms (毫秒)

系统默认: 30000

返回值: 空

### 3.12 配置连续扫描间隔

**void setIntervalTime(int value)**

描述: 配置连续扫描间隔时间, 连续扫描模式下, 每次扫描成功后间隔时间

输入参数: value, 连扫间隔时间, 时间单位: ms (毫秒)

系统默认: 5000

返回值: 空

### 3.13 设置字符编码格式

**void setEncodeFormart(int mode)**

描述：配置解码字符编码格式

输入参数：mode，编码类型值，可选编码如下：

- 0: Auto
- 1: GB2312
- 2: GBK
- 3: GB18030
- 4: UTF-8
- 5: ISO-8859-1
- 6: BIG5
- 7: SJIS
- 8: EUC-JP

返回值：空

### 3.14 是否删除编辑框已有内容

**void setDelete(boolean enable)**

描述：扫到条码之后是否删除编辑框内容

输入参数：enable，自动清空已有内容状态

true：扫到条码之后删除编辑框内容

false：扫到条码之后不删除编辑框内容，默认值。

返回值：空

### 3.15 恢复默认配置参数

**void resetScan()**

描述：恢复扫描参数默认设置，恢复后扫描配置参数恢复到出厂状态。

输入参数：空

返回值：空

### 3.16 配置解码区域模式

**void setCenterMode(int mode)**

描述：配置解码区域模式。

输入参数：mode，解码区域模式，支持如下值：

- 0：区域解码，默认值。
- 1：中心解码，开启后系统只识别瞄准光中心区域的条码。

返回值：空

### 3.17 设置激光补光模式

**void setAimLightMode(int mode)**



描述：设置激光补光模式

输入参数：mode，灯光模式。

0：补光和激光都亮

1：只有激光亮

2：只有补光亮

3：都不亮

返回值：空

### 3.18 配置条码控制开关

```
void setBarcodeEnable(int barcodeId,boolean enable)
```

描述：配置条码控制开关

输入参数：barcodeId，条码类型，支持如下值：

0：Aztec

1：Codabar

2：Code11

3：Code128

4：Code39

6：Code93

8：DataMatrix

9：EAN8

10：EAN13

11：Interleaved 2 of 5

12：Maxicode

13：Micropdf

15：Pdf417

17：QR

19：UPCA

20：UPCE0

48：HANXIN

enable，条码状态，支持如下值：

true：支持指定条码解码

false：关闭指定条码解码

返回值：空

### 3.19 图片输出模式

```
void saveImageMode(int mode)
```

描述：配置图片输出模式

输入参数：mode，图片输出模式，支持如下值：

0：关闭，默认值

- 1: 一直输出
- 2: 解码成功输出
- 3: 解码失败输出

返回值: 空

备注: 1) 此接口仅限 DS7000/7000Pro 扫描头有效

2) 回调图片需要设备获取读取权限, 安卓10.0版本需要在 AndroidManifest.xml 文件 Application 中添加  
`android:requestLegacyExternalStorage="true"`

3) APP 须在主线程运行

### 3.20 注册扫描结果监听对象

**void registerScan(IScanListener mIScanListener)**

描述: 注册监听器, 实现监听器回调方法获取保存图片路径

输入参数: mIScanListener: 监听器对象, 使用示例如下:

实现回调方法如下:

```
public void onScanResults(String data, int type, long decodeTime,
    long keyDownTime, String imagePath) {
    /**
```

不能直接进行UI操作, 建议调用 **runOnUiThread** 切换主线程进行操作

参数说明: data: 条码结果, type: 条码类型

decodeTime: 解码时间, keyDownTime: 扫描按键按下时间

imagePath: 扫描头获取图片的路径

```
*/
    }
}
```

返回值: 无

备注: 1) 此接口仅限 DS7000/7000Pro 扫描头有效, 如果需要获取图像路径, 需要先调用 `saveImageMode` 方法开启图片输出功能。

2) 接口回调会返回条码和图片, 另外这个接口的条码返回不受输出模式的设置参数影响, 建议配合设置输出模式为广播, 避免默认输入框输出模式扫描条码后导致误触UI控件。

3) 开启图片输出后, 请在图片使用结束后删除, 节省设备的存储空间。

### 3.21 注销扫描结果监听对象

**void unregisterScan(IScanListener mIScanListener)**

描述: 注销扫描结果监听器, 在应用销毁时调用。

输入参数: mIScanListener: 监听器对象, 初始化时的 listener 对象

返回值: 无