

Scanning Development Guide

Document revision history

version number	*Change status	brief introduction	date	Change person	Approved date	Approver
V1.0	c	initial version	2021/3/15	Ct		
V1.1	m	Iterative version	2022/10/10	King		

*Change status: C = created, A = added, M = modified, D = deleted

Document approval record

Serial number	Approver	Character	Approval date	signature	Remarks

table of Contents

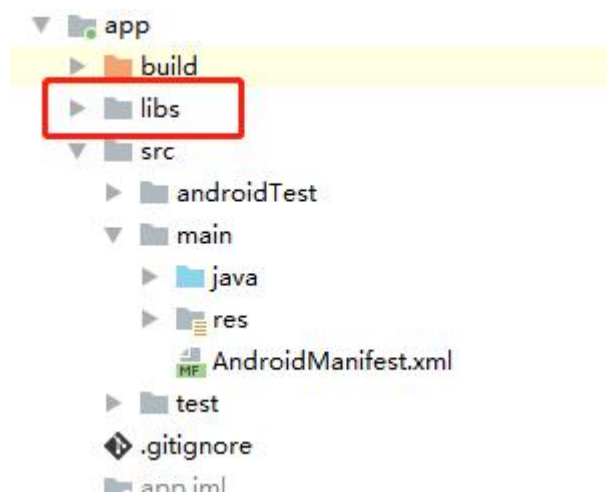
Scanning Development Guide	1
1. Introduction	2
2. Precautions	3
2.1 Configuration android Development environment	3
2.2 Scan call flow	3
3. interface	4
3.1.0 Check whether the scan is turned on	4
3.1.1 Open scan	4
3.1.2 Turn off scanning	5
3.1.3 Start scan	5
3.1.4 Stop scanning	5
3.1.5 Set continuous scan	5
3.1.6 Set scan mode	5
3.1.7 Get the current continuous scan status	6
3.1.8 Get the current scan mode	6
3.1.9 Reset scan	6
4 appendix1	6
4.1. Call example	6

1. Introduction

In order to facilitate secondary development, we provide a function library that can be run on the Java platform. The library is written in Java language.

2. Precautions

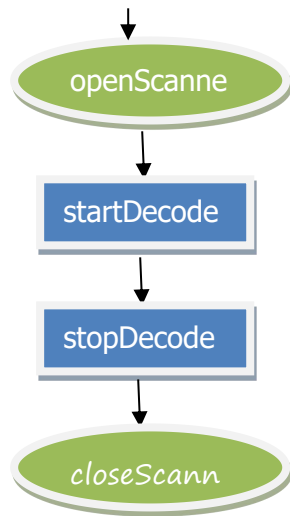
2.1 configure android development environment



- 1、 Import the jar package in the libs folder in the demo into the project
- 2、 In app/build.gradle Add a reference to the jar package

2.2 Scan call process

Register to
broadcast



3.interface

3.1.0 Check whether the scan is turned on

Function interface	public boolean isScanOpened()
Function Description	Whether scanning is on
Parameter Description	no
return value	boolean

3.1.1 Open scan

Function interface	public boolean openScan()
Function Description	Open scan
Parameter Description	no
return value	boolean

3.1.2 Turn off scanning

Function interface	public boolean closeScan()
Function Description	Turn off scan
Parameter Description	no
return value	boolean

3.1.3 Start scan

Function interface	public boolean startScan()
Function Description	Start scanning
Parameter Description	no
return value	boolean

3.1.4 Stop scanning

Function interface	public boolean stopScan()
Function Description	Stop scanning
Parameter Description	no
return value	boolean

3.1.5 Set continuous scan

Function interface	public void setScanLaserMode(int mode)
Function Description	Turn continuous scanning on or off
Parameter Description	mode: 4 on Start continuous scan mode: 8 off Closed continuous scan
return value	no

3.1.6 Set scan mode

Function interface	public boolean setOutScanMode(int mode)
Function Description	Set scan mode
Parameter Description	mode:0 broadcast mode mode:1 edit box mode mode: 2 keyboard mode
return value	no

3.1.7 Get the current continuous scan status

Function interface	public void setScanLaserMode(int mode)
Function Description	Get the current continuous scan status
Parameter Description	mode: 4 on Start continuous scan mode: 8 off Closed continuous scan
return value	Int

3.1.8 Get the current scan mode

Function interface	public int getOutScanMode()
Function Description	Get the current scan mode
Parameter Description	mode:0 broadcast mode mode:1 edit box mode mode: 2 keyboard mode
return value	Int

3.1.9 Reset scan

Function interface	public boolean resetScan()
Function Description	Reset scan
Parameter Description	no
return value	boolean

4 Appendix 1

4.1. Call example

```
// instantiate
```

```
ScanDevice sd = new ScanDevice();
sm.setOutScanMode(0); // Mode-value: 0 broadcast mode, 1 edit box mode,
2 keyboard modes
```

```
// Need to register in broadcast mode scan.rcv.messagebroadcast
```

```
IntentFilter filter = new IntentFilter();
filter.addAction(SCAN_ACTION);
registerReceiver(mScanReceiver, filter);
```

Broadcast example:

```
private BroadcastReceiver mScanReceiver=new BroadcastReceiver () {
    @Override
    public void onReceive(Context context, Intent intent) {
        Log.e ("TAG", "onReceive: "+intent.getAction ());
        String action = intent.getAction ();
        if (action.equals (SCAN_ACTION)){
            byte[] barcode=intent.getByteArrayExtra ("barcode");
            int barodelen=intent.getIntExtra ("length", 0);
            byte temp=intent.getByteExtra ("barcodeType", (byte) 0);
            byte[] aimid=intent.getByteArrayExtra ("aimid");
            barcodeStr=new String (barcode, 0, barodelen);
            showScanResult.append (barcodeStr);
            showScanResult.append ("\n");
            UtilSound.play ();
            sm.stopScan ();
        }
    }
};
```

Broadcast receiving parameters

```
byte[] broadCode = intent.getByteArrayExtra("barcode"); // Barcode
data
int broadCodeLen = intent.getIntExtra("length", 0); // Data length
```