

IPDA030 Scanning development instructions

version	Changed	date	change content
V1.0	ct	2018-11-21	
V 1.1	Ct, Williams	2019-5-22	Android configuration

table of Contents

Document Overview.....	2
For people	2
Documentation purposes	2
Development environment and tools.....	2
The overall design.....	3
Interface functions	3
1.0 isScanOpened ()......	3
1.1 openScan ()......	3
1.2 closeScan ()......	3
1.3 startScan ()......	4
1.4 stopScan ()......	4
1.5 setScanLaserMode ()......	4
1.6 setOutScanMode ()	5
1.7 getScanLaserMode ()......	5
1. 8 getOutScanMode ()	5
1.9 resetScan ()......	6
2.0 getScanCodeValue ()	6
2.0.1 setScanUnBeep ()	6

2.0.2 setScanBeep ().....	7
2.0.3 getScanBeepState ()	7
2.0.4 setScanVibrate ().....	7
2.0.5 setScanUnVibrate ().....	7
2.0.6 getScanVibrateState ().....	8
2.2 Android development and deployment.....	8

Document Overview

Scan module interface description

For people

software developer

Software testers

Documentation purposes

Provide a reference for software developers

Development environment and tools

Development platform: win7 Ultimate 64

Development Tools: Android studio 3.1

Compiler environment: ndkr16B java 1.8

The overall design

Interface functions

1.0 isScanOpened ()

Function Interface	public boolean isScanOpened ()
Function Description	Scanning is open
Parameter Description	no
return value	boolean

1.1 openScan ()

Function Interface	public boolean openScan ()
Function Description	Open Scan
Parameter Description	no
return value	boolean

1.2 closeScan ()

Function Interface	public boolean closeScan ()
--------------------	-----------------------------

Function Description	Close scan
Parameter Description	no
return value	boolean

1.3 startScan ()

Function Interface	public boolean startScan ()
Function Description	Start Scan
Parameter Description	no
return value	boolean

1.4 stopScan ()

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

1.5 setScanLaserMode ()

Function Interface	public void setScanLaserMode (int mode)
Function Description	Continuous scan on or off
Parameter Description	mode: 4 open Kai continuous scan mode: 8 Off Closed continuous scan

return value	no
--------------	----

1.6 setOutScanMode ()

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

1.7 getScanLaserMode ()

Function Interface	public void setScanLaserMode (int mode)
Function Description	Get the current state of continuous scanning
Parameter Description	mode: 4 open Kai continuous scan mode: 8 Off Closed continuous scan
return value	Int

1. 8 getOutScanMode ()

Function Interface	public int getOutScanMode ()
--------------------	------------------------------

Function Description	Get the current scan mode
Parameter Description	mode: 0 broadcast mode mode: 1 mode edit box mode: 2 keyboard mode
return value	Int

1.9 resetScan ()

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

2.0 getScanCodeValue ()

Function Interface	public String getScanCodeValue ()
Function Description	Obtaining scan data
Parameter Description	no
return value	String

2.0.1 setScanUnBeep ()

Function Interface	public boolean setScanUnBeep ()
Function Description	Setting Sound off
Parameter Description	no
return value	boolean

2.0.2 setScanBeep ()

Function Interface	public boolean setScanBeep ()
Function Description	Open the sound settings
Parameter Description	no
return value	boolean

2.0.3 getScanBeepState ()

Function Interface	public boolean getScanBeepState ()
Function Description	Gets the current state of sound (true and false)
Parameter Description	no
return value	boolean

2.0.4 setScanVibrate ()

Function Interface	public boolean setScanVibrate ()
Function Description	Open shock
Parameter Description	no
return value	boolean

2.0.5 setScanUnVibrate ()

Function Interface	public boolean setScanUnVibrate
--------------------	---------------------------------

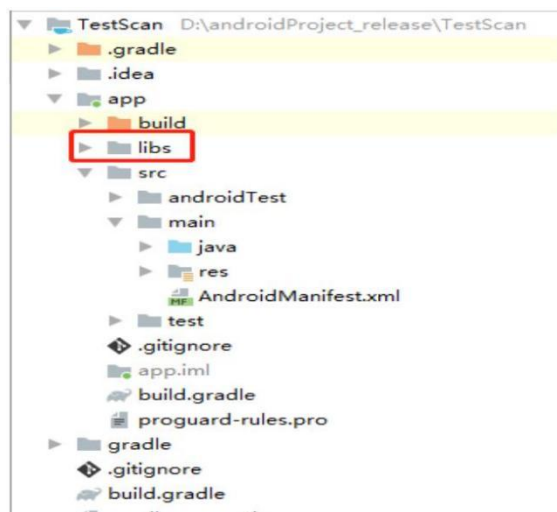
	()
Function Description	Off vibration
Parameter Description	no
return value	boolean

2.0.6 getScanVibrateState ()

Function Interface	public boolean getScanVibrateState ()
Function Description	Get the current state of shock (true and false)
Parameter Description	no
return value	boolean

2.2 Android development and deployment

sdk 2.2.1 will provide copies to the project under libs



2.2.2 Tool introduction

Class	Type
com.example.testscan.util.SoundPoolUtils	SoundPool

Raw audio files in the demo

2.2.3 Using the Operation

```
// init
```

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

```
// broadcast mode under registration required scan.rcv.message  
broadcast
```

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

```
byte temp = intent.getByteExtra("barcodeType", (byte)
0);
byte[] aimid = intent.getByteArrayExtra("aimid");
//broadCodeStr = new String(broadCode, 0,
broadCodeLen);
broadCodeStr = new String(broadCode);
if (broadCodeStr != "" && broadCode != null) {
    SoundPoolUtils.play(2);
    StringBuilder sb = new StringBuilder();
    sb.append(broadCodeStr);
    try {
        String utf8 = new String(
            sb.toString().getBytes("UTF-8"));
        String utf16 = new String(
            sb.toString().getBytes(), "UTF-16");
        String gbk = new String(
            sb.toString().getBytes("GBK"));
        showScanResult.append(utf8);
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    showScanResult.append("\n");
    showScanResult.setTextColor(Color.argb(255,
        random.nextInt(256),
        random.nextInt(256),
        random.nextInt(256)));
} else {
    SoundPoolUtils.play(2);
}
```

```
        sm.stopScan();  
    }  
};
```

Broadcast reception parameters

```
byte[] broadCode = intent.getByteArrayExtra("barcode"); //
```

Barcode data

```
int broadCodeLen = intent.getIntExtra("length", 0); // Data  
length
```

```
byte temp = intent.getByteExtra("barcodeType", (byte) 0); //  
Type
```

```
byte[] aimid = intent.getByteArrayExtra("aimid"); // id
```