# Scanning Application Development Documents (honeyWell)

File information

|  |  |
| --- | --- |
| Author |  |
| Application module | Company worker, customer |
| Application environment | android 4.4 /5.1  |
| Application type | Bay Trail /Cherry Trail |

Upgrade record

|  |  |  |  |
| --- | --- | --- | --- |
| version | **Change date** | **Change content** | Modifier |
| 1.0 | 2017-04-26 | Initial version |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Content

[Scanning Application Development Documents (honeyWell) 1](#_Toc25615)

[1、Summary 3](#_Toc20443)

[2、Description file and interface 3](#_Toc26612)

[3、Application startup and operation process 5](#_Toc1312)

[3.1 Startup and initialization 5](#_Toc21860)

[3.2 Scan data processing 5](#_Toc9394)

[3.3 Parameter setting 6](#_Toc4135)

[3.4 Code setting 8](#_Toc13465)

[3.5 Data output 9](#_Toc31234)

[4、 Other instructions 9](#_Toc7162)

**Disclaimer:**

**This document right emdoor all information, without permission, prohibit dissemination and use.**

# 1、Summary

This document is the extension of <Scanning component serial communication development document>, Honeywell scanning module as an example, the development of scanning app application (barcodescanner). The application of scanning showed that the function of setting and switching, code set, scan results excel derived, boot from start and other functions.

# 2、Description file and interface

According to the function of file, a brief description of each file and key interface:

SerialPortFinder.java

By reading the main equipment files, access to information of all serial devices.

SerialPort.java

By loading the underlying lib library, realize the open and close of serial port, initialization input and output stream.

BootBroadcastReceiver.java

Monitoring on completion of events, from the start application.

SerialPortManager.java

Serial operation management, package serial to open and close interface to encapsulate the write interface of serial port, open read data thread, and provide the data callback interface.

SerialPortService.java

The core of application service. initialization entrance, suffix data processing, HID file node serial write, serial port write instruction, scan data callback display, scan data written to database and so on.

DatabaseHelper.java

Database operation class, create the database, insert and delete data interface.

ExcelOperator.java

Excel data export operation, to achieve specified path to the excel file to export the data in the database.

FileNode.java

Data node operations, Can read and write to system data node. The current application only to write operations.

SoundPlayUtil.java

The tone auxiliary operations, Perform tone playback

BeanData.java

Code for data type. Including code name, code system commands, whether to open the three properties.

MListAdapter.java

Code to select list of list adapter.

MSpinnerAdapter.java

Code select Spinner drop-down box adapter.

MyApplication.java

Application class to implement the access function of the global data of the Application.The one-dimensional code data initialization, the two-dimensional code data initialization,the default configuration of reading code and other interfaces.

StartActivity.java

The main interface of application. real-time display scan results, option key to jump to other interface. (such as setting, code system settings, data export, on interface)

SettingsActivity.java

Set main interface. Including 1D and 2D scanning module, trigger button selection, parameter of code system restore factory settings, HID can be set up, tone can be set up, the vibration can be set up, scan results prefix settings, scan results suffix settings, table suffix enable settings, enter suffix enable settings.

PreferenceHead.java

The dynamic loading page of interactivity class.

SymbologiesQR.java

Two-dimensional module code display and settings interface.

SymbologiesSig.java

One-dimensional module code display and settings interface.

AboutActivity.java

About interface

# 3、Application startup and operation process

## 3.1 Startup and initialization

 The application of scanning, as a service running in the background. By monitoring on completion of event, can boot from the start (android.intent.action.BOOT\_COMPLETED) .Serialportservice class as the start of entrance, the first implementation of soundplayutil initialization, database creation, registration and other local broadcast operations; and then call serialportmanager class to start scanning components, complete device initialization.

 openUart(isSig ? 9600 : 115200);

serialportmanager class implements device initialization, for incoming two parameters to the underlying serial operation function: device node and baud rate.Q62 machine environment, equipment node is "/dev/ttyhsl1", the value of different baud rate module will have different modules. One-dimensional baud rate is 9600, the two-dimensional model group the baud rate is 115200. after the initialization, SerialPortManager class for serial read and write operations through the SerialPortManager instance of SerialPortManager. At the same time by setting the SerialPortListener callback function to the SerialPortManager instance, handle the returned data. So far. initialization is done, trigger scan data will be displayed.

 

## 3.2 Scan data processing

Scanning the data returned by callback interface in mSerialPortListener, in callback functionthe public void onResult (string data) make corresponding treatment.

Automatically stored in database:

databaseHelper.insert(bardb,data);

At the same time display in the main interface:

 intent.putExtra(BARCODEPORT\_RECEIVEDDATA\_EXTRA\_DATA,edit\_data);

According to decide whether to write the data node set:

writeDataAsyncTask.execute(edit\_data);

## 3.3 Parameter setting

The application supports two-dimensional/two-dimensional switch module, scan trigger key switch, code system parameters to restore factory settings, HID can be set up, tone can be set up, the vibration can be set up, scan results prefix settings, scan results suffix settings, table suffix enable settings, enter suffix enable settings. (settingsactivity.java).

One and two dimensional switch module: the application of Honeywell (n4313) compatible one-dimensional and two-dimensional (n3680) two components. Due to the different baud rate need to switch, so in the process of key selection, through the bind mechanism to call the corresponding interface re-initialization of component equipment.

one-dimensional component： mBinder.openUartSig();

two-dimensional component： mBinder.openUartQR();

**Scan trigger button switch:** trigger button settings for selective setting. Select button, button response will trigger the scan.

**The factory settings of code system parameters :** perform a factory setting instruction, will be component parameters restored to the factory default parameters .

Intent intent = **new** Intent(SerialPortService.*BARCODEPORT\_WRITEDATA\_ACTION*);

intent.putExtra(SerialPortService.*BARCODEPORT\_WRITEDATA\_EXTRA\_DATA*, MyApplication.*FACTORY\_DEFAULT*);

 getContext().sendBroadcast(intent);

**HID** can be set up：The control is open, scanning the returned data will automatically write all the input text box.

 writeDataAsyncTask=**new** WriteDataAsyncTask();

 writeDataAsyncTask.execute(edit\_data);

tone can be set up：The control is open, return the data will have voice prompts.

 **if**(sound){

 SoundPlayUtil.*play*();

 }

vibration can be set up：The control is open, the data returned will be shaking.

 **if**(vibrate){

 Vibrator vibrator=(Vibrator)getSystemService(Context.*VIBRATOR\_SERVICE*);

 **long** [] pattern = {100,400};

 vibrator.vibrate(pattern,-1);

 }

**scan results prefix settings**：The front set prefix characters will be automatically added to the scan results.

**scan results suffix settings**：At the end of the suffix character will be automatically added to the scan results.

**table suffix enable settings**：At the end of the scan results increased table suffix.

 suffix = suffix+"\t";

Enter suffix enable settings：At the end of the scan results increased enter suffix.

 edit\_data+"\r\n"



## 3.4 Code setting

**Code logic:** scanning component has default parameter at the factory. When application is first to start the installation and recovery module parameters to factory default value. On this basis, user can set the component code parameter custom setting , results will be stored in local data, re switch records can be restored. Users can also take initiative click the "default settings" button code parameters restore factory default value.

**Code menu:** list each one code in listview list form. In the later in the checkbox display status is checked to indicate that the code has been opened, did not check shows the code system is closed.



## 3.5 Data output

The scanning data results from data export to excel file.

 ExcelOperator.*createFolder*(path\_result);

 ExcelOperator.*write*(**this**,path\_result+"/scan\_result.xls");

# Other instructions

1. The application is compatible with Honeywell component (n4313) one-dimensional and two-dimensional group (n3680), the default for one-dimensional components. Under condition of two-dimensional components, first launched in the settings menu to manually switch for two-dimensional module.
2. Confirm one/two-dimensional menu options are consistent with the module, and not normal communication, please restore the default parameters of engineering.
3. One-dimensional and two-dimensional assembly components, the code set menu has small differences.
4. The HID function is realized by writing data node to the main equipment data, data nodes corresponding to the main equipment need to exist, and can be written.