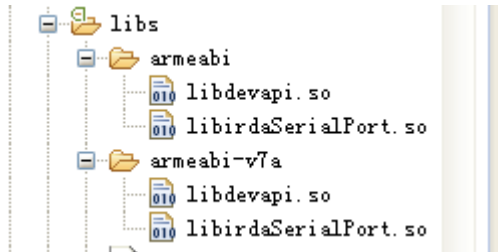


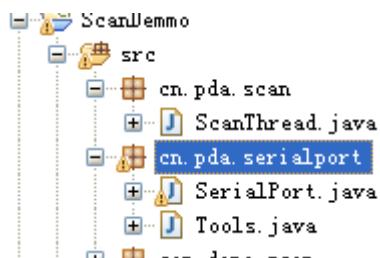
Honeywell 1D/Newland 2D Android API Instruction

1. Import the libraries to the project



Copy the folder of 'armeabi' and 'armeabi-v7a' to the catalogue of libs.

2. Copy JAVA Codes



3. API

3.1 SerialPort(int port, int baudrate, int flags)

Description: Open the serial port and init

Parameters: int port serial port

int baudrate baudrate

int flags status

Return: Serial port instance

3.2 getInputStream()

Description: Get Input Stream

Parameters: None

Return: InputStream

3.3 scanner_poweron()

Description: Power on the barcode scan engine

Parameters: None

Return: Null

3.4 scanner_poweroff()

Description: Power off the barcode scan engine

Parameters: None

Return: Null

3.5 scanner_trigon()

Description: Trigon scanner

Parameters: None

Return: Null

3.6 scanner_trigoff()

Description: Trigoff scanner

Parameters: None

Return: Null

Remark: Trigoff the scanner and use together with scanner_trigon()

4. Scan

Function of scan is packaged inside ScanThread.java in ScanDemmo via serial port.

4.1 Init the serial port

Parameters of port = 0;baudrate = 9600, flags = 0 by default; Clear the messy data after initing successfully.

4.2 Monitor the buffer data

```
@Override
    public void run() {
        try {
```

```

        int size = 0;
        byte[] buffer = new byte[2048];
        int available = 0;
        while(!isInterrupted()){
            available = is.available();
            if(available > 0){
                size = is.read(buffer);
                if(size > 0){
                    sendMessege(buffer, size, SCAN);
                }
            }
        }
    } catch (IOException e) {

        e.printStackTrace();
    }
    super.run();
}

private void sendMessege(byte[] data, int dataLen, int mode){
    String dataStr = new String(data, 0 , dataLen);
    Bundle bundle = new Bundle();
    bundle.putString("data", dataStr);
    Message msg = new Message();
    msg.what = mode;
    msg.setData(bundle);
    handler.sendMessage(msg);
}

```

4.3 Scan

```

public void scan(){
    if(mSerialPort.scanner_trig_stat() == true){
        mSerialPort.scanner_trigoff();
        try {
            Thread.sleep(50);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    mSerialPort.scanner_trigon();
}

```

```
}
```

4.4 Close the serial port and power when exiting

```
public void close() {  
    if (mSerialPort != null) {  
        mSerialPort.scanner_poweroff();  
        try {  
            if (is != null) {  
                is.close();  
            }  
            if (os != null) {  
                os.close();  
            }  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
        mSerialPort.close(port);  
    }  
}
```

4.5 ScanThread.java

```
package cn.pda.scan;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
  
import android.os.Bundle;  
import android.os.Handler;  
import android.os.Message;  
  
import cn.pda.serialport.SerialPort;  
  
public class ScanThread extends Thread {  
  
    private SerialPort mSerialPort ;  
    private InputStream is ;  
    private OutputStream os ;  
    /*serialport parameter*/
```

```

private int port = 0;
private int baudrate = 9600;
private int flags = 0;

private Handler handler ;

public static int SCAN = 1001;  //messege recv mode

/**
 * if throw exception, serialport initialize fail.
 * @throws SecurityException
 * @throws IOException
 */
public ScanThread(Handler handler) throws SecurityException, IOException{
    this.handler = handler;
    mSerialPort = new SerialPort(port, baudrate, flags);
    mSerialPort.scanner_poweron();
    is = mSerialPort.getInputStream();
    os = mSerialPort.getOutputStream();
    try {
        Thread.sleep(50);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    /** clear useless data */
    byte[] temp = new byte[128];
    is.read(temp);
}

@Override
public void run() {
    try {
        int size = 0;
        byte[] buffer = new byte[2048];
        int available = 0;
        while(!isInterrupted()){
            available = is.available();
            if(available > 0){
                size = is.read(buffer);
                if(size > 0){
                    sendMessege(buffer, size, SCAN);
                }
            }
        }
    }
}

```

```

        }
    }
} catch (IOException e) {

    e.printStackTrace();
}
super.run();
}

private void sendMessege(byte[] data, int dataLen, int mode){
    String dataStr = new String(data, 0 , dataLen);
    Bundle bundle = new Bundle();
    bundle.putString("data", dataStr);
    Message msg = new Message();
    msg.what = mode;
    msg.setData(bundle);
    handler.sendMessage(msg);
}

public void scan(){
    if(mSerialPort.scaner_trig_stat() == true){
        mSerialPort.scaner_trigoff();
        try {
            Thread.sleep(50);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    mSerialPort.scaner_trigon();
}

public void close(){
    if(mSerialPort != null){
        mSerialPort.scaner_poweroff();
        try {
            if(is != null){
                is.close();
            }
            if(os != null){
                os.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

```

        }
        mSerialPort.close(port);
    }
}
}

```

4.6 ScanThread.java

Init ScanThread

```

    try {
        scanThread = new ScanThread(mHandler);
    } catch (Exception e) {
        // exception
        Toast.makeText(getApplicationContext(), "serialport init
fail", 0)
            .show();
        return;
        // e.printStackTrace();
    }
    scanThread.start();

```

Call 'Scan'

```

buttonScan.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View v) {
        scanThread.scan();
    }

});

```

Get returned data

```

private Handler mHandler = new Handler() {
    public void handleMessage(android.os.Message msg) {
        if (msg.what == ScanThread.SCAN) {
            String data = msg.getData().getString("data");
            // Toast.makeText(getApplicationContext(), data, 0).show();
            sortAndadd(listBarcode, data);
            addListView();
            eidtBarCount.setText(listBarcode.size() + "");
        }
    };
};

```

