

## Revision Record

Version	Author	Date	Mark
V1.0	Null	2017.01.21	Initial Version
V1.1	Tony	2017.10.11	Increase serial port related [AP02]. ps:AP02 uses ESC command.
V1.1.1	Tony	2017.10.17	Update Features: Read the Print Result[ Success or Fail] Through handler to receive the state; reference: this document <a href="#">2.1.7 Status query ( For RP58A-MBU)</a> Ps: for RP58A-MBU, Bluetooth
v1.1.2	Tony	2018.02.06	Update Features: Add Printer Codepage Settings only support ESC command Reference: <a href="#">3.3.12 public void setHsCodepage</a> 、 <a href="#">3.3.13 public void setHsCodepageForCom</a>

# RongTa Print

--- for Android Development

# Catalog

Revision Record.....	1
RongTa Print.....	1
--- for Android Development.....	1
1、PrintDriver connect to APP methods.....	6
1.1PrintDriver Class .....	6
1.2Creat a Handler SubClass ConnStateHandler, to deal with the message from driver:.....	6
1.3Setting Handler: .....	7
1.4 public void setHandler(Handler handler) .....	7
1.5 Contants Class's state flag for connection: .....	7
1.6 Bluetooth connect Function ( There is no function in the SDK, It is recommended use.): .....	7
1.7 USBconnect Function ( There is no function in the SDK, It is recommended use.): .....	8
1.8 WIFIconnect Function ( There is no function in the SDK, It is recommended use.): .....	8
1.9 COM-AP02 Connect Funtion ( There is no function in the SDK, It is recommended use.): .....	8
2、PrintDriver Class's Function .....	8
2.1 HsBuletoothPrintDriver Class: .....	8
2.1.1 public static HsBluetoothPrintDriver getInstance() .....	8
2.1.2 public synchronized void connect(BluetoothDevice device).....	8
2.1.3 public synchronized void start().....	8
2.1.4 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTType,byte WifiType) .....	9
2.1.5 public void setDhcp(boolean enable) .....	9
2.1.6 public void setStaticIp(String ip, String subMask, String gateWay) .....	9
2.1.7 Status query ( For RP58A-MBU) .....	9
public void setHandler(Handler handler) .....	9
2.2 HsUsbPrintDriver Class: .....	9
2.2.1 public static HsUsbPrintDriver getInstance() .....	9
2.2.2 public void setUsbManager(UsbManager usbManager) .....	9
2.2.3 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTType,byte WifiType) .....	9
2.2.4 public void setDhcp(boolean enable) .....	10
2.2.5 public void setStaticIp(String ip, String subMask, String gateWay) .....	10
2.3 HsWifiPrintDriver Class: .....	10
2.3.1 public static HsWifiPrintDriver getInstance() .....	10
2.3.2 public boolean WIFISocket(String ip, int port).....	10
2.3.3public boolean isNoConnection(String ip).....	10
2.3.4 public void setWifiParam(String ssid,String password,byte type,byte	

WPAType,byte WPAEncryType,byte WEPTYPE,byte WifiType) .....	10
2.3.5 public void setDhcp(boolean enable) .....	11
2.3.6 public void setStaticIp(String ip, String subMask, String gateWay) .....	11
<b>2.4 HsComPrintDriver Class: .....</b>	<b>11</b>
2.4.1 public static HsComPrintDriver getInstance() .....	11
2.4.2 public void setHandler(Handler handler) .....	11
2.4.3 public void setCharsetName(String charsetName) .....	11
2.4.4 public void connect() .....	11
2.4.5 public void disconnect().....	11
2.4.6 public void WriteCmd(String dataString).....	11
2.4.7 public void WriteCmd(String dataString, boolean bGBK) .....	11
2.4.8 public void WriteCmd(byte[] out) .....	11
2.4.9 public synchronized int getState() .....	12
<b>3、Public RTPrinter Command .....</b>	<b>12</b>
<b>3.1 RTPrinter Basic Command .....</b>	<b>12</b>
3.1.1 public synchronized int getState() .....	12
3.1.2 public synchronized void stop() .....	12
3.1.3 public boolean IsNoConnection() .....	12
3.1.4 public void SetDefaultSetting().....	12
3.1.5 public void Begin() .....	12
3.1.6 public void LF() .....	12
3.1.7 public void CR() .....	12
3.1.8 public void SelftestPrint().....	12
3.1.9 public void Beep(byte times, byte time) .....	12
3.1.10 public void StatusInquiry() .....	12
3.1.11 public void CLS() .....	13
3.1.12 public void endPro ().....	13
<b>3.2 Print Position Functions .....</b>	<b>13</b>
3.2.1 public void SetRightSpacing(byte Distance) .....	13
3.2.2 public void SetAbsolutePrintPosition(byte nL, byte nH).....	13
3.2.3 public void SetRelativePrintPosition(byte nL, byte nH) .....	13
3.2.4 public void SetDefaultLineSpacing().....	13
3.2.5 public void SetLineSpacing(byte LineSpacing) .....	13
3.2.6 public void SetLeftStartSpacing(byte nL, byte nH) .....	13
3.2.7 public void SetAreaWidth(byte nL, byte nH) .....	14
<b>3.3 Set Character Functions.....</b>	<b>14</b>
3.3.1 public void SetCharacterPrintMode(byte CharacterPrintMode) .....	14
3.3.2 public void SetUnderline(byte UnderlineEn) .....	14
3.3.3 public void SetBold(byte BoldEn).....	15
3.3.4 public void SetCharacterFont(byte Font).....	15
3.3.5 public void SetPrintRotate(byte RotateEn) .....	15
3.3.6public void SetPrintRotate(byte Rotate) .....	15
3.3.7public void SetAlignMode(byte AlignMode) .....	15
3.3.8public void SetInvertPrint(byte InvertModeEn).....	16

3.3.9	public void SetFontEnlarge(byte FontEnlarge)	16
3.3.10	public void SetBlackReversePrint(byte BlackReverseEn)	16
3.3.11	public void setCharsetName(String charsetName)	16
3.3.12	public void setHsCodepage(byte selCodepage)	17
3.3.13	public void setHsCodepageForCom(byte selCodepage)	17
<b>3.4</b>	<b>Set Chinese character Functions</b>	<b>17</b>
3.4.1	public void SetChineseCharacterMode(byte ChineseCharacterMode)	17
3.4.2	public void SelChineseCodepage()	18
3.4.3	public void CancelChineseCodepage()	18
3.4.4	public void SetChineseUnderline(byte ChineseUnderlineEn)	18
<b>3.5</b>	<b>Control Drawer Function</b>	<b>18</b>
3.5.1	public void OpenDrawer(byte DrawerNumber, byte PulseStartTime, byte PulseEndTime)	18
<b>3.6</b>	<b>Control Cut Paper Functions</b>	<b>18</b>
3.6.1	public void CutPaper()	18
3.6.2	public void PartialCutPaper()	18
3.6.3	public void FeedAndCutPaper(byte CutMode)	19
3.6.4	public void FeedAndCutPaper(byte CutMode, byte FeedDistance)	19
<b>3.7</b>	<b>Special Print Functions</b>	<b>19</b>
3.7.1	public void AddCodePrint(BarcodeType CodeType, String data)	19
3.7.2	private void CODE_QR_CODE(String data)	19
3.7.3	public void printImage(Bitmap bitmap)	20
	public boolean printImage(Bitmap bitmap, int type, PrintImageCallback callback)	20
	public boolean printImageSync(Bitmap bitmap, int type)	20
3.7.4	public boolean defineNVLogo(String[] logolist, int type)	20
3.7.5	public void printNVLogo(byte nvLogoIndex, int type)	20
<b>3.8</b>	<b>status query</b>	<b>21</b>
3.8.1	public byte StatusInquiryFinish()	21
3.8.2	public byte StatusInquiryFinish(int handlerSign, Handler mHandler)	21
3.8.3	public byte StatusInquiry(byte type)	21
<b>4、</b>	<b>Label command section</b>	<b>21</b>
<b>4.1</b>	<b>Main Classes</b>	<b>21</b>
<b>4.2</b>	<b>The label commands which are identical with thermal printing commands</b>	<b>21</b>
<b>4.3</b>	<b>Label System Commands</b>	<b>22</b>
4.3.1	public void SetSize(String width, String length)	22
4.3.2	public void SetGAP (String width, String length)	22
4.3.3	public void SetOFFSET(String length)	22
4.3.4	public void SetSPEED(String speed)	22
4.3.5	public void SetDENSITY (String ensity)	22
4.3.6	public void SetDIRECTION (String dir)	22
4.3.7	public void SetREFERENCE(String posi_x, String posi_y)	22
4.3.8	public void SetSHIFT(String dots)	22
4.3.9	public void SetCODEPAGE(String num)	23
4.3.10	public void SetFEED (String dots)	23

4.3.11 public void SetBACKFEED (String dots).....	23
4.3.12 public void SetFORMFEED() .....	24
4.3.13 public void SetHOME().....	24
4.3.14 public void SetPRINT(String setM, String copyN) .....	24
4.3.15 public void SetSOUND(String level, String interval) .....	24
4.3.16 public void SetLIMITFEED (String n).....	24
<b>4.4 Label Formatting Commands.....</b>	<b>24</b>
4.4.1 public void SetBAR(String x, String y, String width, String height).....	24
4.4.2 public void drawBox(String x_start, String y_start, String x_end, String y_end, String lineThickness) .....	25
4.4.3 public void drawBitMap(String x, String y, String width, String height, String mode, byte[] data).....	25
4.4.4 public void PutBmp(String x, String y, String fileName).....	25
4.4.5 public void PutPcx(String x, String y, String fileName).....	26
4.4.6 public void ClearData(String X_start, String Y_start, String X_width, String Y_height).....	26
4.4.7 public void SetReverse(String X_start, String Y_start, String X_width, String Y_height).....	26
4.4.8 public void PrintText(String X, String Y, String font, String rotation, String x_multi, String y_multi, String content).....	26
4.4.9 public void loadProFile(String FileName) .....	27
4.4.10 public void loadDateFile(String FileName, String dataSize, String data) { .....	27
4.4.11 public void setCounter(String num, String step) { .....	28
4.4.12 public void CodePrint(String X, String Y, String codeType, String height, String readable, String rotation, String narrow, String wide, String code) .....	28
<b>5、 pdf Print .....</b>	<b>28</b>
<b>5.1 Steps to follow .....</b>	<b>28</b>
<b>Appendix 1 Codepage-Number Match List.....</b>	<b>30</b>

# RTPrinter Command for Android SDK Development Document

## Remind Developer:

The SDK include three interfaces that to connecting RTPrinter :WIFI、USB、Bluetooth.

## 1、 PrintDriver connect to APP methods

### 1.1 PrintDriver Class

- 1) HsBluetoothPrintDriverBluetooth
- 2) HsUsbPrintDriver USB
- 3) HsWifiPrintDriver WIFI

### 1.2 Creat a Handler SubClass ConnStateHandler, to deal with the message from driver:

```
private class ConnStateHandler extends Handler {
    @Override
    public void handleMessage(Message msg) {
        super.handleMessage(msg);
        Bundle data = msg.getData();
        switch (data.getInt("flag")) {
            /*The FLAG_STATE_CHANGE indicat that connection state will change
            when the Handler received message*/
            case Contants.FLAG_STATE_CHANGE:
                /*This Case indicate the current connection state that include four
                state and you can find a state from the Contants Class
                UNCONNECTED、
                CONNECTED_BY_BLUETOOTH、
                CONNECTED_BY_USB、
                CONNECTED_BY_WIFI) */
                int state = data.getInt("state");
                //As for receiving state , you can write corresponding Code in here

                break;
```

```

        /*The FLAG_FAIL_CONNECT indicat that connection is Failed when the
        Handler received message*/
        case Contants.FLAG_FAIL_CONNECT:
//As for receiving state , you can write corresponding Code in here

        break;

        /*The FLAG_FAIL_CONNECT indicat that connection is Successful when
        the Handler received message*/
        case Contants.FLAG_SUCCESS_CONNECT:
//As for receiving state , you can write corresponding Code in here

        break;
    }
}
}

```

### 1.3 Setting Handler:

```

ConnStateHandler connStateHandler = new ConnStateHandler();
HsBluetoothPrintDriver.getInstance().setHandler(connStateHandler);
HsUsbPrintDriver.getInstance().setHandler(connStateHandler);
HsWifiPrintDriver.getInstance().setHandler(connStateHandler);

```

### 1.4 public void setHandler(Handler handler)

Explain: This function is used to set up Handler when the Program has just started.

### 1.5 Contants Class's state flag for connection:

```

Include: UNCONNECTER
        CONNECTED_BY_BLUETOOTH
        CONNECTED_BY_USB
        CONNECTED_BY_WIFI
        FLAG_STATE_CHANGE
        FALG_FAIL_CONNECT
        FLAG_SUCCESS_CONNECT
        FLAG_MSG_READ

```

### 1.6 Bluetooth connect Function (There is no function in the SDK, It is recommended use.):

```

private void connectBluetooth(BluetoothDevice bluetoothDevice) {
    HsBluetoothPrintDriver hsBluetoothPrintDriver = HsBluetoothPrintDriver.getInstance();
    hsBluetoothPrintDriver.start();
    hsBluetoothPrintDriver.connect(bluetoothDevice);
}

```

```
}
```

### **1.7 USBconnect Function (There is no function in the SDK, It is recommended use.):**

```
private void connectUsb(UsbDevice usbDevice){  
    HsUsbPrintDriver hsUsbPrintDriver = HsUsbPrintDriver.getInstance();  
    hsUsbPrintDriver.connect(usbDevice);  
}
```

### **1.8 WIFIconnect Function (There is no function in the SDK, It is recommended use.):**

```
private void connectWifi(String ip,int port) {  
    new Thread(new Runnable() {  
        @Override  
        public void run() {  
            HsWifiPrintDriver hsWifiPrintDriver = HsWifiPrintDriver.getInstance();  
            hsWifiPrintDriver.WIFISocket(ip,port);  
        }  
    }).start();  
}
```

### **1.9 COM-AP02 Connect Funtion (There is no function in the SDK, It is recommended use.):**

```
private void connectCOM () { // Serial connection for AP02  
    HsComPrintDriver.getInstance (). Connect ();  
}
```

## **2、 PrintDriver Class's Function**

### **2.1 HsBuletoothPrintDriver Class:**

#### **2.1.1 public static HsBluetoothPrintDriver getInstance()**

Explain: HsBuletoothPrintDriver Class can instantiate a Bluetooth object.

#### **2.1.2 public synchronized void connect(BluetoothDevice device)**

Explain: Start the ConnectThread to initiate a connection to a remote device.

Param: deviceThe BluetoothDevice to connect.

#### **2.1.3 public synchronized void start()**

Explain: Start the chat service. Specifically start AcceptThread to begin a session in I



listening (server) mode. Called by the Activity onResume().

#### **2.1.4 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTType,byte WifiType)**

Explain: Set WIFI info

ssid: wifi name

password: wifi password

type: (OPEN/WPA/WEP)

WPAType: (WPA-PSK/WPA2-PSK)

WPAEncryType: (AES/TPIK)

WEPTType: (OPEN/SHARE)

WifiType: wifiType (STA/AP)

#### **2.1.5 public void setDhcp(boolean enable)**

Explain: Set DHCP enable or disable

Enable: true is enable, false is disable

#### **2.1.6 public void setStaticIp(String ip, String subMask, String gateWay)**

Explain: Set ip

#### **2.1.7 Status query ( For RP58A-MBU)**

##### **public void setHandler(Handler handler)**

Explain: The program is called when it is started, used to set the Handler, can receive the return information, return the information type View Constants, as follows:

int FLAG\_BLUETOOTH\_RECEIVE\_DATA = 0x98;

// resultVal&0xff, can get the value like success:0x80 fail:0x00

The implementation can refer to the RTExample code

## **2.2 HsUsbPrintDriver Class:**

#### **2.2.1 public static HsUsbPrintDriver getInstance()**

Explain: HsUsbPrintDriver Class can instantiate a USB object.

#### **2.2.2 public void setUsbManager(UsbManager usbManager)**

Explain: This function is used to set up UsbManager when the Program has just started.

Param: usbManager

#### **2.2.3 public void setWifiParam(String ssid,String password,byte type,byte**

#### **WPAType,byte WPAEncryType,byte WEPTType,byte WifiType)**

Explain: Set WIFI info

ssid: wifi name

password: wifi password

type: (OPEN/WPA/WEP)  
WPAType: (WPA-PSK/WPA2-PSK)  
WPAEncryType: (AES/TPIK)  
WEPTType: (OPEN/SHARE)  
WifiType: wifiType (STA/AP)

#### **2.2.4 public void setDhcp(boolean enable)**

Explain: Set DHCP enable or disable  
Enable: true is enable, false is disable

#### **2.2.5 public void setStaticIp(String ip, String subMask, String gateWay)**

Explain: Set ip

### **2.3 HsWifiPrintDriver Class:**

#### **2.3.1 public static HsWifiPrintDriver getInstance()**

Explain: HsWifiPrintDriver can instantiate a WIFI object.

#### **2.3.2 public boolean WIFISocket(String ip, int port)**

Explain: Setting WIFI Socket  
Param: ip IP address  
port Device Port Number  
Return: TrueConnection is successful  
FalseConnection is fail

#### **2.3.3 public boolean isNoConnection(String ip)**

Explain: checkif WIFI is connected  
Param: ip IP address  
Return: TrueNoConnection  
FalseConnection

#### **2.3.4 public void setWifiParam(String ssid,String password,byte type,byte WPAType,byte WPAEncryType,byte WEPTType,byte WifiType)**

Explain: Set WIFI info  
ssid: wifi name  
password: wifi password  
type: (OPEN/WPA/WEP)  
WPAType: (WPA-PSK/WPA2-PSK)  
WPAEncryType: (AES/TPIK)  
WEPTType: (OPEN/SHARE)  
WifiType: wifiType (STA/AP)

### **2.3.5 public void setDhcp(boolean enable)**

Explain: Set DHCP enable or disable

Enable: true is enable, false is disable

### **2.3.6 public void setStaticIp(String ip, String subMask, String gateWay)**

Explain: Set ip

## **2.4 HsComPrintDriver Class:**

### **2.4.1 public static HsComPrintDriver getInstance()**

Explain: HsComPrintDriver Instantiate an object

### **2.4.2 public void setHandler(Handler handler)**

Explain: The program is called when it is started, used to set the Handler, can receive the return information, return the information type View Contants, as follows:

int CONNECTED\_BY\_COM = 0x14;

int FLAG\_STATE\_CHANGE = 0x20;

int FLAG\_FAIL\_CONNECT = 0x21;

int FLAG\_SUCCESS\_CONNECT = 0x22;

int FLAG\_COM\_RECEIVE\_DATA = 0x99;

### **2.4.3 public void setCharsetName(String charsetName)**

Explain: Set the character encoding, can be set to "UTF-8", "GBK" and so on

### **2.4.4 public void connect()**

Explain: do connect

### **2.4.5 public void disconnect()**

Explain: do disconnect

### **2.4.6 public void WriteCmd(String dataString)**

Explain: Can directly print text, write instructions to the printer, the default character encoding

Param: dataString

### **2.4.7 public void WriteCmd(String dataString, boolean bGBK)**

Explain: Can directly print text, write instructions to the printer, specify the character encoding

Param:

dataString     string

bGBK            Encoding format

### **2.4.8 public void WriteCmd(byte[] out)**

Explain: Write instructions to the printer

Param:

out            byte [] form

#### **2.4.9    public synchronized int getState()**

Explain: Get Connected state

Reference: Contants class, refer to the description in 2.4.2

## **3、 Public RTPrinter Command**

### **3.1    RTPrinter Basic Command**

#### **3.1.1    public synchronized int getState()**

Explain: Return the current connection station.

#### **3.1.2    public synchronized void stop()**

Explain: Disconnect the Terminal Device and Printer Device.

#### **3.1.3    public boolean IsNoConnection()**

Explain: Judge Terminal Device and Printer Device connection state.

Return: Truedisconnected state

Falseconnected state

#### **3.1.4    public void SetDefaultSetting()**

Explain: Set default model of printer.

#### **3.1.5    public void Begin()**

Explain: Initiate, Reset, Clear Printer's buffer

#### **3.1.6    public void LF()**

Explain:The Printer's paper will go ahead one line by default setting line space when the print buffer was empty.

#### **3.1.7    public void CR()**

Explain: When printer was allowed to auto feed, this command the same as the LF().Reversed, this command was ignored.

#### **3.1.8    public void SelftestPrint()**

Explain: Print selftest page command.

#### **3.1.9    public void Beep(byte times, byte time)**

Explain: Buzzer's sound command.

Param: timesbeep times

time    a beep time

#### **3.1.10    public void StatusInquiry()**

Explain: Query state command.

### **3.1.11 public void CLS()**

Explain: clears the buffer

### **3.1.12 public void endPro ()**

Explain: End of program

## **3.2 Print Position Functions**

### **3.2.1 public void SetRightSpacing(byte Distance)**

Explain: Set character right space command.

Param: Distance

Range:  $0 \leq \text{Distance} \leq 255$

### **3.2.2 public void SetAbsolutePrintPosition(byte nL, byte nH)**

Explain: Sets the distance from the beginning of the line to the position at which subsequent characters are to be printed. The distance from the beginning of the line to the print position is :

$[(nL + nH \times 256) \times 0.125 \text{ mm}]$

Param:  $0 \leq nL \leq 255$  ,  $0 \leq nH \leq 255$

### **3.2.3 public void SetRelativePrintPosition(byte nL, byte nH)**

Explain: Sets the print starting position based on the current position using horizontal or vertical motion units. This command sets the distance from the current position to:

$[(nL + nH \times 256) \times 0.125 \text{ mm}]$

Param:  $0 \leq nL \leq 255$  ,  $0 \leq nH \leq 255$

### **3.2.4 public void SetDefaultLineSpacing()**

Explain: Selects 3.75 mm ( $30 \times 0.125 \text{ mm}$ ) line spacing.

### **3.2.5 public void SetLineSpacing(byte LineSpacing)**

Explain: Sets the line spacing to  $[n \times 0.125 \text{ mm}]$ .

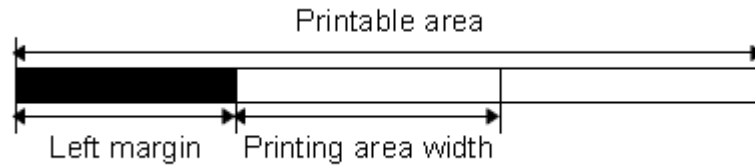
Param: LineSpacing

Range:  $0 \leq \text{LineSpacing} \leq 255$

### **3.2.6 public void SetLeftStartSpacing(byte nL, byte nH)**

Explain: Sets the left margin using nL and nH. The left margin is set to :

$[(nL + nH \times 256) \times 0.125 \text{ mm}]$



Param:  $0 \leq nL \leq 255$  ,  $0 \leq nH \leq 255$

### 3.2.7 public void SetAreaWidth(byte nL, byte nH)

Explain: Sets the printing area width to the area specified by nL and nH. The printing area width is set to :

$$[(nL + nH \times 256) \times 0.125\text{mm} (0.0049'')]$$

Param:  $0 \leq nL \leq 255$  ,  $0 \leq nH \leq 255$

## 3.3 Set Character Functions

### 3.3.1 public void SetCharacterPrintMode(byte CharacterPrintMode)

Explain: Selects print mode(s) using CharacterPrintMode as follows:

Bit	Off/On	Hex	Decimal	Function
0	Off	00	0	Character Font A (12×24).
	On	01	1	Character Font B (9×17).
1	-	-	-	Undefined.
2	-	-	-	Undefined.
3	Off	00	0	Emphasized mode not selected.
	On	08	8	Emphasized mode selected.
4	Off	00	0	Double-height mode not selected.
	On	10	16	Double-height mode selected.
5	Off	00	0	Double-width mode not selected.
	On	20	32	Double-width mode selected.
6	-	-	-	Undefined.
7	Off	00	0	Underline mode not selected.
	On	80	128	Underline mode selected.

Param:  $0 \leq \text{CharacterPrintMode} \leq 255$

### 3.3.2 public void SetUnderline(byte UnderlineEn)

Explain: Turns underline mode on or off, based on the following values

n	Function
0, 48	Turns off underline mode
1, 49	Turns on underline mode (1 dot thick)
2, 50	Turns on underline mode (2 dots thick)

Param: UnderlineEn

### 3.3.3 public void SetBold(byte BoldEn)

Explain: Turns emphasized mode on or off When the LSB of n is 0, emphasized mode is turned off. When the LSB of n is 1, emphasized mode is turned on.

Param:  $0 \leq \text{BoldEn} \leq 255$

### 3.3.4 public void SetCharacterFont(byte Font)

Explain: Selects the character font.

n	Function
0, 48	Character Font A (12×24) selected.
1, 49	Character Font B (9×17) selected.

Param: Font

### 3.3.5 public void SetPrintRotate(byte RotateEn)

Explain: Turns 90° clockwise rotation mode on/off

RotateEn is used as follows:

RotateEn	Function
1	Turns off 90° clockwise rotation mode
2	Turns on 90° clockwise rotation mode

Param: RotateEn

### 3.3.6 public void SetPrintRotate(byte Rotate)

Explain: Turns 90° clockwise rotation mode on/off

Rotate is used as follows:

RotateEn	Function
1	Turns off 90° clockwise rotation mode
2	Turns on 90° clockwise rotation mode
0	normal

Param: Rotate

### 3.3.7 public void SetAlignMode(byte AlignMode)

Explain: Aligns all the data in one line to the specified position.

Align

AlignMode selects the justification as follows:

n	Justification
0, 48	Left justification
1, 49	Centering
2, 50	Right justification

Param: AlignMode

### 3.3.8 public void SetInvertPrint(byte InvertModeEn)

Explain: Turns upside-down printing mode on or off.

When the LSB of n is 0, upside-down printing mode is turned off.

When the LSB of n is 1, upside-down printing mode is turned on.

Param:  $0 \leq \text{InvertModeEn} \leq 255$

### 3.3.9 public void SetFontEnlarge(byte FontEnlarge)

Explain: Selects the character height using bits 0 to 2 and selects the character width using bits 4 to 7, as follows:

Bit	Off/On	Hex	Decimal	Function
0				Character height selection. See Table 2.
1				
2				
3				
4				Character width selection. See Table 1.
5				
6				
7				

**Table 1**  
**Character Width Selection**

Hex	Decimal	Width
00	0	1(normal)
10	16	2(double-width)
20	32	3
30	48	4
40	64	5
50	80	6
60	96	7
70	112	8

Param:  $0 \leq \text{FontEnlarge} \leq 255$

**Table 2**  
**Character Height Selection**

Hex	Decimal	Width
00	0	1(normal)
01	1	2(double-height)
02	2	3
03	3	4
04	4	5
05	5	6
06	6	7
07	7	8

### 3.3.10 public void SetBlackReversePrint(byte BlackReverseEn)

Explain: Turns on or off white/black reverse printing mode.

When the LSB of n is 0, white/black reverse mode is turned off.

When the LSB of n is 1, white/black reverse mode is turned on.

Param:  $0 \leq \text{BlackReverseEn} \leq 255$

### 3.3.11 public void setCharsetName(String charsetName)

Explain: **charsetName**, if no setting, is GBK charset default.

Param: charsetName print english or Chinese set "GBK", else set "UTF-8".



### 3.3.12 public void setHsCodepage(byte selCodepage)

Explain: Printer codepage settings, this function only support **portable pos**, Like RPP200,02N and so on, Only support **ESC commands**.

Param: selCodepage, type: byte ,

accept value: Reference: [Appendix 1 Codepage-Number Match List](#)

### 3.3.13 public void setHsCodepageForCom(byte selCodepage)

Explain: Printer codepage settings, this function only support **AP02**, Only support **ESC commands**.

Param: selCodepage, type: byte ,

accept value: Reference: [Appendix 1 Codepage-Number Match List](#)

**How to use:**

```
HsComPrintDriver.getInstance().setHsCodepageForCom(byte N);
```

**For Example** WCP1251 Codepage settings:

```
int codePageWCP1251 = 6;
```

```
HsComPrintDriver.getInstance().setHsCodepageForCom(codePageWCP1251);
```

```
4 = "CP865 [Nordic]"  
5 = "CP865 [Nordic]"  
6 = "WCP1251 [Cyrillic]"  
7 = "CP866 Cyrillic #2"  
8 = "MIK[Cyrillic /Bulgarian]"  
9 = "CP755 [East Europe, Latvian 2]"  
10 = "Iran"
```

## 3.4 Set Chinese character Functions

### 3.4.1 public void SetChineseCharacterMode(byte ChineseCharacterMode)

Explain: Sets the print mode for Kanji characters, using ChineseCharacterMode as follows:

Bit	Off/On	Hex	Decimal	Function
0	—	—	—	Undefined.
1	—	—	—	Undefined.
2	Off	00	0	Double-width mode is OFF.
	On	04	4	Double-width mode is ON.
3	Off	00	0	Double-height mode is OFF.
	On	08		Double-height mode is ON.
4	—	—	—	Undefined.
5	—	—	—	Undefined.

6	—	—	—	Undefined.
7	Off	00	0	Underline mode is OFF.
	On	80	128	Underline mode is ON.

Param:  $0 \leq \text{ChineseCharacterMode} \leq 255$

### 3.4.2 public void SelChineseCodepage()

Explain: Selects Kanji character mode.

### 3.4.3 public void CancelChineseCodepage()

Explain: Cancels Kanji character mode.

### 3.4.4 public void SetChineseUnderline(byte ChineseUnderlineEn)

Explain: Turns underline mode for Kanji characters on or off, based on the following values of ChineseUnderlineEn.

ChineseUnderlineEn	Function
0, 48	Turns off underline mode for Kanji characters
1, 49	Turns on underline mode for Kanji characters (1-dot thick)
2, 50	Turns on underline mode for Kanji characters (2-dot thick)

Param: ChineseUnderlineEn

## 3.5 Control Drawer Function

### 3.5.1 public void OpenDrawer(byte DrawerNumber, byte PulseStartTime, byte PulseEndTime)

Explain: Outputs the pulse specified by PulseStartTime and PulseEndTime to connector pin DrawerNumber as follow :

On time= PulseStartTime x 2 millisecond

Off time= PulseEndTime x 2 millisecond

DrawerNumber =0/48 Drawer kick –out connector pin 2;

DrawerNumber =1/49 Drawer kick –out connector pin 5.

Param: DrawerNumberSet Value = 1 OR 2

PulseStartTimeOpen time= PulseStartTime x 2 millisecond

PulseEndTimeClose time= PulseEndTime x 2 millisecond

Range:  $0 \leq \text{PulseStartTime} \leq 2550 \leq \text{PulseEndTime} \leq 255$

## 3.6 Control Cut Paper Functions

### 3.6.1 public void CutPaper()

Explain: full cut paper.

### 3.6.2 public void PartialCutPaper()

Explain: partial cut paper

### 3.6.3 public void FeedAndCutPaper(byte CutMode)

Explain: Select cut mode and cut paper

Param: CutModeSet Value = 1 OR 49, Print Mode only has partial cut paper.

CutMode	Print mode
1, 49	Partial cut (one point left uncut)

### 3.6.4 public void FeedAndCutPaper(byte CutMode, byte FeedDistance)

Explain: Select cut mode and cut paper

Param: Selects a mode for cutting paper and executes paper cutting. The value of Cut Modeselects the mode as follows:

CutMode	Print mode
66	Feeds paper (cutting position + [FeedDistance $\times$ 0.125 mm]), and cuts the paper partially (onepoint left uncut).

## 3.7 Special Print Functions

### 3.7.1 public void AddCodePrint(BarcodeType CodeType, String data)

Explain: BarCode Command.

Param: CodeType

data

BarCode enum type:

```
public enum BarcodeType{
    UPC_A,
    UPC_E,
    EAN13,
    EAN8,
    CODE39,
    ITF,
    CODABAR,
    CODE93,
    CODE128,
    QR_CODE
}
```

### 3.7.2 private void CODE\_QR\_CODE(String data)

Explain: QRCode Command。

Param: data

### 3.7.3 public void printImage(Bitmap bitmap)

**public boolean printImage(Bitmap bitmap, int type, PrintImageCallback callback)**

Explain: Print Image

Param: bitmap           Bitmap Object

type           TYPE\_58: 58 printer serials (384dots)

TYPE\_80: 80printer serials (576dots)

//TYPE\_300: RPP300printer serials (640dots)

callback       Print the end of the picture, callback, to prevent print out of order, can be new PrintImageCallback (), in the realization of the method, the operation after printing pictures

### **public boolean printImageSync(Bitmap bitmap, int type)**

Explain: Print Image Sync

Param: bitmap           Bitmap Object

type           TYPE\_58: 58 printer serials (384dots)

TYPE\_80: 80printer serials (576dots)

ps:

Print the picture for the time-consuming operation, it is recommended in the sub-thread call the method;

### 3.7.4 public boolean defineNVLogo(String[] logolist, int type)

Explain: define NV Logo to printer

Param: **logolist**       the full path string array of nv logo. For example:  
**/storage/emulated/0/DCIM/Camera/1.bmp"**

type   TYPE\_58: 58 printer serials (384dots)

TYPE\_80: 80printer serials (576dots)

TYPE\_300: RPP300printer serials (640dots)

### 3.7.5 public void printNVLogo(byte nvLogoIndex, int type)

Explain: print nv logo.

Param: nvLogoIndex the index of nv logo,1,2,3 etc

type   TYPE\_58: 58 printer serials (384dots)

TYPE\_80: 80printer serials (576dots)

TYPE\_300: RPP300printer serials (640dots)

## 3.8 status query

### 3.8.1 public byte StatusInquiryFinish()

Explain: call this api after printing, query if print is success.

Param: none

Return value: return 0x80 if print success,else return error value

### 3.8.2 public byte StatusInquiryFinish(int handlerSign, Handler mHandler)

Explain: call this api after printing, query if print is success.

Param: handlerSign:the FLAG of Handler,you define it  
mhandler:the object of Handler

Return value:

Elample:Bundle data = msg.getData();

“data.getInt("flag")” is handlerSign

int inquiry\_status =data.getInt("state", 0) & 0xFF;

the “inquiry\_status” is statue,it is success if return 0x80

### 3.8.3 public byte StatusInquiry(byte type)

Explain: query status of printer

Param: type value can be 1 2 3 4

Return value: status value of printer。 You can reference 80 instruction manual DLE EOT n instructions for detail.

## 4、 Label command section

### 4.1 Main Classes

- 1) LabelBluetoothPrintDriver Bluetooth
- 2) LabelUsbPrintDriver USB
- 3) LabelWifiPrintDriver WIFI

### 4.2 The label commands which are identical with thermal printing commands

Begin(), SetCLS(), stop(), WIFISocket(), WIFI\_Write(), BT\_Write(), SetPrintRotate(),  
SetCLS(), LF(), CR(), Beep(), StatusInquiry(), SelftestPrint(), IsNoConnection(), endPro()  
.....

## 4.3 Label System Commands

### 4.3.1 public void SetSize(String width, String length)

Explain: This command defines the label width and length

Param: width: Label width(mm)    length: Label Length(mm)

### 4.3.2 public void SetGAP (String width, String length)

Explain: Define the gap distance between two labels

Param: width: The gap distance between two labels    ( $0 \leq m \leq 25.4$  (mm))

Length: The offset distance of the gap

### 4.3.3 public void SetOFFSET(String length)

Explain: This command defines the selective, extra label feeding length each form feed takes, which, especially in peel-off mode, is used to adjust label stop position, so as for label to register at proper places for the intended purposes. The printer backtracks the extra feeding length before the next run of printing

param: length : The offset distance (mm)

### 4.3.4 public void SetSPEED(String speed)

Explain: This command defines the print speed

param: speed = 2/3/4/5

### 4.3.5 public void SetDENSITY (String ensity)

Explain: This command designates the level of darkness of printing

param: ensity = 0~15

### 4.3.6 public void SetDIRECTION (String dir)

Explain: This command defines the printout direction and mirror image. And this will be memorized in EEPROM.

param: dir    0: Print normal image. 1: Print mirror image

### 4.3.7 public void SetREFERENCE(String posi\_x, String posi\_y)

Explain: This command defines the reference point of the label. The reference (origin), point varies with the print direction

param: posi\_x: Horizontal coordinate, with "dot" as the unit.

posi\_y: Vertical coordinate, with "dot" as the unit

### 4.3.8 public void SetSHIFT(String dots)

Explain: This command can be used to fine-tune the entire label up or down from its current position. The position relates to the top edge of the label. A negative value moves the entire label away from the top of the label; a positive value moves the entire label to the top of label

param: dots: The maximum value is 1 inch. For 203 DPI printers, the range is -203 to 203; The unit is dot

#### 4.3.9 public void SetCODEPAGE(String num)

Explain: 选择对应的国际字符集

<u>Parameter</u>	<u>Description</u>
num	name or number of code page, which can be divided into 8-bit code page further

##### 7-bit code page Character set name

USA: USA

BRI: British

GER: German

FRE: French

DAN: Danish

ITA: Italian

SPA: Spanish

SWE: Swedish

SWI: Swiss

##### 8-bit code page number

437: United States

850: Multilingual

852: Slavic

860: Portuguese

863: Canadian/French

865: Nordic

##### Windows code page

1250: Central Europe

1252: Latin I

1253: Greek

1254: Turkish

#### 4.3.10 public void SetFEED (String dots)

Explain: This command feeds label with the specified length.  
The length is specified by dot.

param: dots : unit: dot  $1 \leq n \leq 9999$

**Note:** 203 DPI: 1 mm = 8 dots

#### 4.3.11 public void SetBACKFEED (String dots)

Explain: To back feed label with the specified length. The length is specified by dot.

param: dots : unit: dot  $1 \leq n \leq 9999$

**Note:** 203 DPI: 1 mm = 8 dots

#### 4.3.12 public void SetFORMFEED()

Explain: This command feeds label to the beginning of next label

#### 4.3.13 public void SetHOME()

Explain: It is not expected the first label will be printed on the right position when the printer power is turned on. This command will feed label to the beginning of next label .The size and gap of the label should be setup in advance.

#### 4.3.14 public void SetPRINT(String setM, String copyN)

Explain: This command prints the label format stored in the image buffer.

param: setM: Specifies how many sets of labels will be printed.

$1 \leq \text{setM} \leq 999999999$

copyN: Specifies how many copies should be printed for each set of label.

$1 \leq n \leq 999999999$

#### 4.3.15 public void SetSOUND(String level, String interval)

Explain: This command is used to control the sound frequency of the beeper. There are 10 levels of sounds. The timing control the sound can be set by the “interval” parameter.

param: level: Sound level: 0~9

Interval: Sound interval: 1~4095

#### 4.3.16 public void SetLIMITFEED (String n)

Explain: When feeding labels, if the gap sensor is not set to a suitable sensitivity, the printer will not be able to locate the correct position of the gap. This command is used stop label feeding and make the red LED flash if the printer does not locate gap after feeding the length of one label plus one preset value.

param: n:mm

### 4.4 Label Formatting Commands

#### 4.4.1 public void SetBAR(String x, String y, String width, String height)

Explain: This command is used to draw a line or a bar on the label format

<u>Parameter</u>	<u>Description</u>
x	The upper left corner x-coordinate in dot
y	The upper left corner y-coordinate in dot
width	The width of bar in dot
height	The height of bar in dot

**Note:** 203 DPI: 1 mm = 8 dots



Recommended max. bar height is 12mm at 3" width. Bar height over than 12 mm may damage the power supply and affect the print quality.

Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.

#### 4.4.2 public void drawBox(String x\_start, String y\_start, String x\_end, String y\_end, String lineThickness)

Explain: This command is used to draw rectangles on the label.

<u>Parameter</u>	<u>Description</u>
X_start	Specify x-coordinate of upper left corner in dot
Y_start	Specify y-coordinate of upper left corner in dot
X_end	Specify x-coordinate of lower right corner in dot
Y_end	Specify y-coordinate of lower right corner in dot
Line	thickness Line thickness of the box in dot

**Note:** 203 DPI: 1 mm = 8 dots

Recommended max. thickness of box is 12mm at 3" width.

Thickness of box that is larger than 12 mm may damage the power supply and affect the print quality.

Max. print ratio is different for each printer model. Desktop and industrial printer print ratio is limited to 20% and 30% respectively.

#### 4.4.3 public void drawBitMap(String x, String y, String width, String height, String mode, byte[] data)

Explain: This command is used to draw bitmap images (Not BMP graphic file).

<u>Parameter</u>	<u>Description</u>
x	Specify the x-coordinate of the bitmap image
y	Specify the y-coordinate of the bitmap image
width	The width of the image in bytes
height	The height of the image in dot
mode	Graphic mode is listed below:
0	OVERWRITE
1	OR
2	XOR
data	The bitmap data

#### 4.4.4 public void PutBmp(String x, String y, String fileName)

Explain: This command is used to print BMP format image.

Parameter

X: The x-coordinate of the BMP format image

Y: The y-coordinate of the BMP format image

Filename: The downloaded BMP filename.

#### 4.4.5 public void PutPcx(String x, String y, String fileName)

Explain: This command is used to print PCX format image

Parameter

X: The x-coordinate of the PCX image

Y: The y-coordinate of the PCX image

Filename: The downloaded PCX filename. Case sensitiveFilename

#### 4.4.6 public void ClearData(String X\_start, String Y\_start, String X\_width, String Y\_height)

Explain: This command is used to clear a specified region in image buffer

<u>Parameter</u>	<u>Description</u>
X_start	The x-coordinate of the starting point in dot
Y_start	The y-coordinate of the starting point in dot
X_width	The region width in x-axis direction in dot
Y_height	The region height in y-axis direction in dot

#### 4.4.7 public void SetReverse(String X\_start, String Y\_start, String X\_width, String Y\_height)

Explain: This command is used to reverse a region in image buffer.

<u>Parameter</u>	<u>Description</u>
X_start	The x-coordinate of the starting point in dot
Y_start	The y-coordinate of the starting point in dot
X_width	The region width in x-axis direction in dot
Y_height	The region height in y-axis direction in dot

注意事项: 200 DPI: 1 mm = 8 dots

300 DPI: 1 mm = 12 dots

#### 4.4.8 public void PrintText(String X, String Y, String font, String rotation, String x\_multi, String y\_multi, String content)

Explain: This command is used to print text on label

<u>Parameter</u>	<u>Description</u>
X	The x-coordinate of the text
Y	The y-coordinate of the text
Font	Font name
0	Monotype CG Triumvirate Bold Condensed, font width and height is stretchable

1	8 x 12 fixed pitch dot font
2	12 x 20 fixed pitch dot font
3	16 x 24 fixed pitch dot font
4	24 x 32 fixed pitch dot font
5	32 x 48 dot fixed pitch font
6	14 x 19 dot fixed pitch font OCR-B
7	21 x 27 dot fixed pitch font OCR-B
8	14 x25 dot fixed pitch font OCR-A
TST24.BF2	Traditional Chinese 24 x 24 font
TSS24.BF2	Simplified Chinese 24 x 24 font (GB)
K	Korean 24 x 24 font (KS)
<b>Rotation</b>	<b>The rotation angle of text</b>
0	0 degree
90	90 degrees, in clockwise direction
180	180 degrees, in clockwise direction
270	270 degrees, in clockwise direction
X-multiplication:	Horizontal multiplication, up to 10x.
Available factors:	1~10 width (point) of true type font. 1 point=1/72 inch.
Y-multiplication:	Vertical multiplication, up to 10x.
	Available factors: 1~10 For true type font, this parameter is used to specify the height (point) of true type font. 1 point=1/72 inch.

**Note:**

1. If there is any double quote (") within the text, please change it to \["].
2. If font "0" is used, the font width and font height is stretchable  
by x-multiplication and y-multiplication parameter. It is  
expressed by pt (point). 1 point=1/72inch.

#### 4.4.9 public void loadProFile(String FileName)

Explain: download the program file

param: FileName: The filename resident in printer memory

Note:

- (1). The filename is case sensitive.
- (2). The extension of the program file must be ".BAS"

#### 4.4.10 public void loadDateFile(String FileName, String dataSize, String data) {

Explain: download the data file

Parameter:

FileName: The name of data file that will remain resident in the printer memory. It is case sensitive.

DATA SIZE: The actual size (numbers of byte) of the data file without header.

**Note:** For text data file, CR (carriage return) 0x0D and LF (Line Feed) 0x0A is the separator of data.

#### 4.4.11 public void setCounter(String num, String step) {

Explain: Counter can be a real counter or a variable.

This setting sets the counter number in program and their increments. There are three different kind of counters: digit (0~9~0), lower case letter(a~z~a) and upper case letter (A~Z~A).

Parameter:

Num : counter number. There are 51 counters available (@0~@50) in the printer.

Step: The increment of the counter, can be positive or negative. -999999999<= step <=999999999

If the counter is used as a fixed variable, please set the increment to 0.

#### 4.4.12 public void CodePrint(String X, String Y, String codeType, String height, String readable, String rotation, String narrow, String wide, String code)

Explain: This command is used to print 1D barcodes on label form

<u>Parameter</u>	<u>Description</u>
X	Specify the x-coordinate of the bar code on label
Y	Specify the y-coordinate of the bar code on label
codeType:	UPCA\EAN8\EAN13\39\39C\ITF\CODEBAR\128M\25\25C.....
height	bar code height expressed by dot
Readable	0: human not readable 1: human readable
rotation	Rotate bar code clockwise in degrees (0, 90, 180, 270)
narrow	width of narrow element in dot
wide	width of wide element in dot
code	bar code content

Example:

```
CodePrint ("100","100","39","96",1,0,2,4,"1000")
```

## 5、 pdf Print

### 5.1 Steps to follow

- 1.Add all the program package under the directory of "pdf-import\libs\" to your own project;
- 2.Add the corresponding resources under the folder of "pdf-import" to the project(As for the style interface of the resource file, it can be modified according to your needs);

3.Refer to the AndroidManifest.xml and build.gradle files under folder of pdf-import, you can change your own corresponding project file;

4.Call the following statement

```
Intent intent = new Intent ();  
Intent.setClass (this, ChoosePDFActivity.class);  
StartActivity (intent)
```

Note:Please compile with reference to RExample, if you find any resource files missing, please go and search under project of RExample

# Appendix 1 Codepage-Number Match List

255 = "CHINESE"  
254 = "BIG5"  
253 = "UTF-8"  
252 = "SHIFT-JIS"  
251 = "EUC-KR"  
250 = "UNICODE"  
249 = "Kazakhstan"  
0 = "CP437 [U.S.A. = Standard Europe]"  
1 = "Katakana"  
2 = "CP850 [Multilingual]"  
3 = "CP860 [Portuguese]"  
4 = "CP863 [Canadian-French]"  
5 = "CP865 [Nordic]"  
6 = "WCP1251 [Cyrillic]"  
7 = "CP866 Cyrilliec #2"  
8 = "MIK[Cyrillic /Bulgarian]"  
9 = "CP755 [East Europe, Latvian 2]"  
10 = "Iran"  
15 = "CP862 [Hebrew]"  
16 = "WCP1252 Latin I"  
17 = "WCP1253 [Greek]"  
18 = "CP852 [Latina 2]"  
19 = "CP858 Multilingual Latin I +Euro"  
20 = "Iran II"  
21 = "Latvian"  
22 = "CP864 [Arabic]"  
23 = "ISO-8859-1 [West Europe]"  
24 = "CP737 [Greek]"  
25 = "WCP1257 [Baltic]"  
26 = "Thai"  
27 = "CP720[Arabic]"  
28 = "CP855"  
29 = "CP857[Turkish]"  
30 = "WCP1250[Central Eurpoe]"  
31 = "CP775"  
32 = "WCP1254[Turkish]"  
33 = "WCP1255[Hebrew]"  
34 = "WCP1256[Arabic]"  
35 = "WCP1258[Vietnam]"  
36 = "ISO-8859-2[Latin 2]"  
37 = "ISO-8859-3[Latin 3]"

38 = "ISO-8859-4[Baltic]"  
39 = "ISO-8859-5[Cyrillic]"  
40 = "ISO-8859-6[Arabic]"  
41 = "ISO-8859-7[Greek]"  
42 = "ISO-8859-8[Hebrew]"  
43 = "ISO-8859-9[Turkish]"  
44 = "ISO-8859-15 [Latin 3]"  
45 = "Thai2"  
46 = "CP856"  
47 = "Cp874"  
48 = "TCVN3"  
49 = "Armenian"