Catalog

Printer Command for Android SDK Development Document

Remind Developer:

The SDK include three interfaces that to connecting Printer : WIFI、

USB、Bluetooth.

1、PrintDriver connect to APP methods

1.1 PrintDriver Class

1）HsBluetoothPrintDriver　　　Bluetooth
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 USB connect 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 WIFI connect 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();
}
```

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：device The BluetoothDevice to connect.

2.1.3 public synchronized void start()

Explain：Start the chat service. Specifically start AcceptThread to begin a
session in listening (server) mode. Called by the Activity onResume().

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

2.3 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: True Connection is successful

False Connection is fail

3、 Public Printer Command

3.1 Printer 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: True disconnected state

False connected 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： times   beep times
           time   a beep time

3.1.10   public void StatusInquiry()

Explain：Query state command.

3.2 Print Position Functions

3.2.1 public void SetRightSpacing(byte Distance)

Explain：Set character right space command.
Param：Distance
Range：0 <= Distance <= 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 <= nL <= 255 , 0 <= nH <= 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 <= nL <= 255 , 0 <= nH <= 255

3.2.4 public void SetDefaultLineSpacing()

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

3.2.5 public void SetLineSpacing(byte LineSpacing)

Explain：Sets the line spacing to [n$\times$0.125 mm].
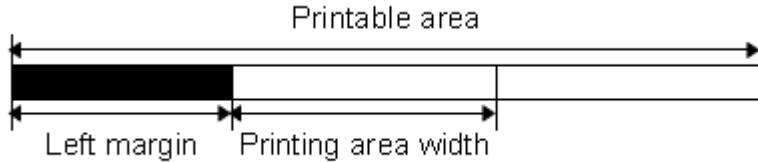Param： LineSpacing
Range： 0 <= LineSpacing <= 255

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 <= nL <= 255 , 0 <= nH <= 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 :

$$[(nL + nH \times 256) \times 0.125mm (0.0049")]$$

Param： 0 <= nL <= 255 , 0 <= nH <= 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 <= CharacterPrintMode <= 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 <= BoldEn <= 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 SetRotate(byte RotateEn)

Explain： Turns 90   clockwise rotation mode on/off

RotateEn is used as follows:

RotateEn	Function
0,48	Turns off 90° clockwise rotation mode
1,49	Turns on 90° clockwise rotation mode

Param： RotateEn

3.3.6 public void SetAlignMode(byte AlignMode)

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

AlignMode selects the justification as follows:

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

Param： AlignMode

3.3.7 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 LSBof n is 1, upside-down printing mode is turned on.

Param： 0 <= InvertModeEn <= 255

3.3.8 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

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

Param： 0 <= FontEnlarge <= 255

3.3.9 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 <= BlackReverseEn <= 255

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 <= ChineseCharacterMode <= 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： DrawerNumber   Set Value = 1 OR 2

         PulseStartTime   Open time= PulseStartTime x 2 millisecond

         PulseEndTime   Close time= PulseEndTime x 2 millisecond

Range：   0 <= PulseStartTime <= 255   0 <= PulseEndTime <= 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： CutMode   Set 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 CutMode selects the mode as follows:

CutMode	Print mode
66	Feeds paper (cutting position + [FeedDistance×0.125 mm]), and cuts the paper partially (one point 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)

Explain： Print Image

Param： bitmap