Sunmi-K1 Print Service Development Documentation

Introduction：

This documentation is mainly used to introduce how the Sunmi developers apply print service on Sunmi-K1 device to realize their own print task;

Sunmi-K1 invokes the printer via Aidl;

Connect print service via AIDL method:

AIDL is the abbreviation of Android Interface Definition language, which is a type of descriptive language of the communication interface of Android internal process. Through it we can define the communication interface between processes. Sunmi AIDL provides encapsulated common printing commands to make it easy for the developers to quickly access Sunmi printer. Meanwhile, Sunmi also supports all the common “ESC/POS” command set, capable of directly sending command set via the interface to control the printer.

The establishment of connection can be divided into the following 5 steps:

1. Add AIDL document (also including java document as for partial models) attached to the resource file in the project.
2. Realize ServiceConnection in the code class controlling the print.
3. Invoke ApplicationContext.bindService() and transfer in the realization of ServiceConnection.

Note: bindservice is non-blocking invoke, which means the immediate binding after the completion of invoke is not successful, and it must be subject to serviceConnected.

4. In the realization of ServiceConnection.onServiceConnected(), you will receive an IBinder example (Service to be invoked). Invoke ExtPrinterService.Stub.asInterface(service) and transform the parameters into the corresponding print service type in Aidl document.
5. Now it’s OK to invoke various methods defined in Aidl interface to carry out printing.

Pseudocode example：

Bound Service

```
Intent intent = new Intent();
intent.setPackage("com.sunmi.extprinterservice");
intent.setAction("com.sunmi.extprinterservice.PrinterService");
bindService(intent, serviceConnection, Context.BIND_AUTO_CREATE);
```

It is necessary to establish a new ServiceConnection service to bind the callback

```
ServiceConnection serviceConnection = new ServiceConnection()
   { @Override
   public void onServiceConnected(ComponentName name, IBinder service) {
      ExtPrinterService interface = ExtPrinterService.Stub.asInterface(service);
   }

   @Override
   public void onServiceDisconnected(ComponentName name)

   { }
};
```

Use interface object to realize one’s own printing task

```
interface.printText(“123456\n”);
```

Unbind the service after the completion of the usage

```
unbindService(serviceConnection);
```

Interface descriptions for print service：

1、Initialization of the printer

Method	int printerInit()
Descriptions	As for the initialization of the printer, the data in the print cache will be cleared, yet it will not affect the receive cache. And get the properties of all the previous settings back to default state.
Input	None
Return	See return value of the interface

2、Obtain the printer status

Method	int getPrinterStatus()
Descriptions	Obtain the operating status of the printer
Input	None
Return	-1 The printer is offline or the print service has not been connected to the printer 0 Normal operation of the printer 1 The printer is uncapped 2 Printer out of paper 3 The printer is going to be out of paper 4 Printer overheat

3、Paper feed of the printer according to row height

Method	Int lineWrap(int n) Paper feed of the printer for n rows
Descriptions	If there are data in the printer's print cache, the data will be outputted and the paper will be fed If the row height is set as 0, the paper feed distance will be 0
Input	Row number（0<n<256）
Return	See return value of the interface

4、Paper feed of the printer according to pixel

Method	Int pixelWrap(int n)
Descriptions	Paper feed of the printer for n rows of dots If there are data in the printer’ s print cache, data will be outputted and the paper will be fed for n rows of dots
Input	Pixels（0<n<256）
Return	See return value of the interface

5、Refresh the cache

Method	int flush()
Descriptions	Refresh the print cache. They will be outputted when there are data in the cache. The paper will be fed for one row when there is no data
Input	None
Return	See return value of the interface

6、 Horizontal positioning

Method	int tab()
Descriptions	Move the print position to the next horizontal positioning point If the horizontal positioning point exceeds the print area, move to the end of row If it’ s already at the end of row, line feed will be operated
Input	None
Return	See return value of the interface

7、Print text content

Method	int printText(String text)
Descriptions	This method will transform the input text into hexadecimal byte stream corresponding to character set coding The print service transforms the text content into gb18030 coding by default
Input	text Text content to be printed
Return	See return value of the interface

8、Print bar code

Method	int printBarCode(String code, int type, int width, int height, int hriPos)
Descriptions	Print the bar code of the custom content
Input	code Content of bar code （bar code content need to meet its format according to different bar code types） type Bar code type 0：UPC-A 1：UPC-E 2：EAN13 3：EAN8 4:CODE39 5:ITF 6:CODABAR 7:CODE93 8:CODE128 See remarks on bar code type width Bar code width 2-6 pixels （If the set bar code width is too large, the exceeded part of the whole barcode to the paper width will not output bar code content） height Bar code height 1-255 pixels hriPos HRI position 0：Not to print 1：Upward side of bar code 2：Downward side of bar code 3：Upward & downward side of bar code
Return	See return value of the interface

9、Print QR code

Method	int printQrCode(String code, int modeSize, int errorlevel)
Descriptions	Print QR code
Input	code QR code content to be printed, utf-8 character set by default modeSize Size of QR code block 1-16 pixels errorlevel Error correction level of QR code 0-3 four levels
Return	See return value of the interface

10、Print pictures

Method	int printBitmap(Bitmap bitmap, int mode)
Descriptions	Transform bitmap picture into raster bitmap picture for printing This method is applicable for pictures with a print width within the print paper The serial port flow control mode will be blocked when the data are too large, and the data will be lost when there is no flow control
Input	bitmap Bitmap picture to be printed，the width & height of the picture are both integral multiples of 8 mode0：Common 1：Double width 2：Double height 3：Double height & double width
Return	See return value of the interface

11、Set lateral horizontal positioning point

Method	int setHorizontalTab(int[] k)
Description	The position of the horizontal positioning will be marked. The position of each mark point will be specified by the character width of k[n] ascii. The default horizontal positioning point takes the character width of 8 ascii as a positioning point
Input k	lateral positioning point array k[n]， the maximum length of the array (n) can reach 32， 0<k[n]<256 Within the array, there must be an ascending order, otherwise exception will be thrown out When k is set as null, default positioning point will be recovered, the default position is an interval of 8 characters
Return	See return value of the interface

12、Set alignment

Method	int setAlignMode(int type)
Descriptions	Set the alignment of the print content
Input	0 Left aligned (default)、1 Centered、2 Right aligned
Return	See return value of the interface

13、Cutter's paper cutting

Method	int cutPaper(int m, int n)
Descriptions	Paper cutting
Input	m paper cutting mode 0：full cutting 1：half cutting 2：cutting paper feed n paper feed This parameter is valid only when the setting m=2. Due to different printer models, the distances from the cutter to the printing head are different. When n=0, the paper will be automatically fed for this distance. When n>0, it will be fed for additionally set distance
Return	See return value of the interface

14、Send data command

Method	int sendRawData(byte[] cmd)
Descriptions	Send epson control command
Input	cmd epson command to be sent
Return	See return value of the interface

15、Set font size

Method	int setFontZoom(int hori, int veri)
Descriptions	Due to limitation by the printer hard font library, the font size can only be amplified by multiples. This method can be applied to control the amplification of font in lateral & longitudinal direction
Input	The range of hori 、veri is 1-8, indicating the amplification of font in lateral & longitudinal direction. Error parameter will be returned if the setting is out of the range
Return	See return value of the interface

16、Table printing

Method	int printColumnsText(String[] colsTextArr, int[] colsWidthArr, int[] colsAlign)
Descriptions	Print content is outputted via table. Each array indicates the data & format on this column, and you need to invoke this method several times to reach the style effect of table output Note：the previous set style will be initialized after invoking this method !
Input	colsTextArr: content of each column to be printed, supporting Chinese & ascii code colsWidthArr：maximum character amount to be contained in each column.Take ascii code number as unit as for character amount（ one Chinese character equals 2 ascii code amount） When the text content exceeds the max amount to be contained, it will move to the next row of this column. The print will not be carried out if max character amount of all the columns exceeds that to be contained by one row colsAlign：alignment of content of each column. It will be effective only when the character amount of the content is less than max character amount
Return	see return value of the interface

Descriptions for the interface return value:

Since the print task carried out by the printer is asynchronous with sending data, the interface return does not represent whether the actual printing is successful or not. Except special instructions, the interface return values all indicate the reception conditions of the print service in terms of this command. Thus, it is possible that the printer can still receive the print data even in abnormal state such as lack of paper, etc. When the exception is recovered, the cache will be continuously carried out to print data;

When the return value >=0, it indicates that this command has been sent successfully, the printer will process it; As for the successfully returned concrete value, refer to the printer status return value ;
When the return value <0, it indicates that sending of this command has failed, the print task will not be carried out. The concrete errors are as follows:

-1 The printer is off line or it is not ready
-2 The cache is full, unable to receive print data
-3 Exception of the data sending
-4 Errors of sending command or parameters

Descriptions of the bar code type:

code39 maximum 13 numbers in length to be printed

code93 maximum 17 numbers in length to be printed

code128 code128 are divided into three types：{A,{B,{C；A type：including capital letters, numbers, punctuations, etc. B type: capital & lower case letters, numbers; C type: pure numbers; As for default B type code, if you are going to use this code, it is necessary to add “{A”、“{B” 、“{C” before the content to declare the type to be used, and you can use a mixture of it, for example: “{A2344A”，”{C123123”，”{A1A{B13Bxc{C12”.

ean8 requires 8 digit number (the last one is check digit)

with an effective length of 8 numbers

ean13 with an effective length of 13 numbers, among which

the last one is check digit

ITF requires the input of numbers, with an effective length less than 14 digits, which must be of even bit

Status feedback:

The printer status can be actively obtained by invoking the interface

getPrinterStatus（）; it can also be obtained asynchronously via

registering broadcast:

Function	action
Printable	com.sunmi.extprinterservice.NORMAL_ACTION
The printer is off line	com.sunmi.extprinterservice.OFLLINE_ACTION
The printer compartment cover is not closed	com.sunmi.extprinterservice.COVER_OPEN_ACTION
Lack of paper as for the printer	com.sunmi.extprinterservice.OUT_OF_PAPER_ACTION
The printer paper will be used up	com.sunmi.extprinterservice.LESS_OF_PAPER_ACTION
Printer overheat	com.sunmi.extprinterservice.HOT_ACTION