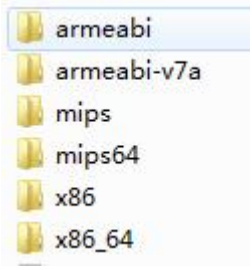
Android ESC SDK Manual

1. SDK Introduction

1) SDK jar :

In this jar, there are connectors which connect to the printer. Our SDK connectors include Bluetooth, USB, WIFI and signal interface. It also includes the connector of print commands, such as printing text, bar code, image, and so on.

2) SO Library :



2. Connecting Method

2.1 Bluetooth Connection

Connect Bluetooth :

```
int PortOpen(Context context,String portSetting)
```

Example:

Print.PortOpen(context,“Bluetooth,”+MAC)

MAC: Bluetooth address of printer

Return:

0: connection success

-1: connection failure

Disconnect Bluetooth:

```
public static boolean PortClose()
```

Example:

Print.PorClose()

Return:

True: disconnection success

False: disconnection failure

Whether Bluetooth is connected:

```
public static boolean IsOpened()
```

Example:

Print.IsOpened()

Return:

True: Bluetooth connected

False: Bluetooth unconnected

2.2 WIFI Connection

Connect WIFI:

```
int PortOpen(Context context,String portSetting)
```

Example:

Print.PortOpen(context,“WiFi,”+IP+”,”+PortNumber)

IP: IP address of printer

PortNumber: port Defaul: 9100

Return:

0: connection success

-1: connection failure

Disconnect WiFi:

```
public static boolean PortClose()
```

Example:

Print.PorClose()

Return:

True: disconnection success

False: disconnection failure

Whether WiFi is connected:

```
public static boolean IsOpened()
```

Example:

Print.IsOpened()

Return:

True: connected

False: unconnected

2.3 USB Connection

Connect USB:

```
int PortOpen(Context context,UsbDevice usbdevice)
```

Example:

Print.PortOpen(context,usbdevice)

usbdevice: UsbDevice object

Return:

0:connection success

-1:connection failure

Disconnect USB:

```
public static boolean PortClose()
```

Example:

Print.PorClose()

Return:

true:Disconnection success

false:disconnection failure

Whether USB is connected:

```
public static boolean IsOpened()
```

Example:

Print.IsOpened()

Return:

true:connected

false:unconnected

2.4 Signal Interface Connection

Connect signal interface:

```
int PortOpen(Context context,String portSetting)
```

Example:

Print.PortOpen("Serial,"+port+","+baudrate)

port: node of signal interface (differ from models) e.g./dev/ttyS1

baudrate: baud rate e.g.9600

Return:

0: connection success

-1: connection failure

Disconnect signal interface:

```
public static boolean PortClose()
```

Example:

Print.PorClose()

Return:

true: disconnection success

false: disconnection failure

Whether signal interface is connected:

```
public static boolean IsOpened()
```

Example:

Print.IsOpened()

Return:

true: connected

false: unconnected

3.Print Command

3.1 Paper feed

```
public static int PrintAndLineFeed()
```

Example:

Print.PrintAndLineFeed()

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.2 Paper feed after printing

```
public static int PrintAndFeed(int distance)
```

Example:

Print.PrintAndFeed(distance)

distance:paper feeding length(Unit: distance*y model unit mm)

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.3 Reverse feed after printing

```
public static int PrintAndReverseFeed(int distance)
```

Example:

Print.PrintAndReverseFeed(distance)

distance: reverse feed length (distance*y model unit mm)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.4 Print and feed n line

```
public static int PrintAndFeedNLine(byte lines)
```

Example:

Print.PrintAndFeedNLine(lines)

lines:N X (current line spacing)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.5 Print and reverse feed n line

```
public static int PrintAndReverseFeedNLine(int lines)
```

Example:Print.PrintAndReverseFeedNLine(lines)

lines:N X (current line spacing)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.6 Set text line space

```
public static int SetDefaultTextLineSpace()
```

Example:

Print.SetDefaultTextLineSpace()

Note: Set default text line (3.75mm)

```
public static int SetTextLineSpace(byte lineSpace)
```

Example:

Print.SetTextLineSpace(byte lineSpace)

lineSpace: line spacing (lineSpace*y model unit mm)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.7 Select character font

```
public static int SelectCharacterFont(byte characterFont)
```

Example:

Print.SelectCharacterFont(byte characterFont)

characterFont:

0:FontA big font

1:FontB small font

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.8 Set language

```
public static int SetCharacterSet(byte characterSet)
```

Example:

Print.SetCharacterSet(byte characterSet)

Set simple Chinese:

Print.LanguageEncode=”gb2312”

Print.SetCharacterSet(0)

Set English:

Print.LanguageEncode=”iso8859-1”

Print.SetCharacterSet(0)

Set traditional Chinese:

Print.LanguageEncode=”big5”

Print.SetCharacterSet(0)

To set other languages, please refer to Table 1-1 on the last two pages.

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.9 Set justification

```
public static int SetJustification(int justification)
```

Example:

Print.SetJustification(int justification)

justification: 0: left justifying

1: center

2:right justifying

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.10 Get printer status

```
public static int GetTransmitStatus(int transmitItem,byte[] statusData)
```

Function:

Print.GetTransmitStatus(int transmitItem,byte[] statusData)

transmitItem: 1:Get paper status

2:Get drawer status

statusData:return status(see as below), length is 1

Search paper:

Bit	OFF/ON	Hex	Decimal	Status
0,1	OFF	00	0	Paper near end sensor: enough paper
	ON	03	3	Paper near end sensor: paper near end
2,3	OFF	00	0	Paper out sensor: with paper
	ON	0c	12	Paper out sensor: without paper
4	OFF	00	0	Fixed
5,6	--	--	--	Reserved
7	OFF	00	0	Fixed

Search drawer:

Bit	OFF/ON	Hex	Decimal	Status
0	OFF	00	0	Signal of drawer pin 3 is low
	ON	01	1	Signal of drawer pin 3 is high
1-3	--	--	--	Reserved
4	OFF	00	0	Fixed
5,6	--	--	--	Reserved
7	OFF	00	0	Fixed

Example:

statusData=new byte[1];

Print.GetTransmitStatus(1, statusData);

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.11 Initializing printer

```
public static int Initialize()
```

Example:

Print.Initialize()

Restore the printer to the start-up status.

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.12 Set print density

```
public static int SetPrintDensity(byte density)
```

Function:

Print.SetPrintDensity(byte density)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.13 Set print speed

```
public static int SetPrintSpeed(byte speed)
```

Function:

Print.SetPrintSpeed(byte speed)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.14 Cut paper

```
public static int CutPaper(int cutMode)
```

Function:

Print.CutPaper(int cutMode)

cutMode: default is 1

Feed paper and then cut paper

```
public static int CutPaper(int cutMode,int distance)
```

Function:

Print.CutPaper(int cutMode,int distance)

cutMode: default is 1

distance: feeding distance (distance*y model unit mm)

note:

The paper travel distance is calculated from outside the print area.

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.15 Drawer

```
public static int OpenCashdrawer(int openMode)
```

Function:

Print.OpenCashdrawer(int openMode)

openMode: 0:Open No.1 drawer

1:Open No.2 drawer

2:Open two drawers

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.16 Beep buzzer

```
public static int BeepBuzzer(byte times,byte t1,byte t2)
```

Function:

Print.BeepBuzzer(byte times,byte t1,byte t2)

times: times of beep

t1: time of beep（ti× 100ms）。

t2: time of stop（t2× 100ms）。

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.17 Print text

1.Function:

Print.PrintText(String data)

data: text content

Example:

Print.PrintText(“TEXT\n”)

2.Function:

PrintText(String data,int alignment,int attribute,int textSize)

data: text content

alignment: alignment method 0:left alignment

1:center

2:right alignment

attribute: style.

0：Large font,no bold, no underline, no highlight.

1：small font,no bold, no underline, no highlight.

2：Large font,bold, no underline, no highlight.

3：small font,bold, no underline, no highlight.

4：Large font,no bold, underline, no highlight.

5：small font,no bold, underline, no highlight.

6：Large font,bold, underline, no highlight.

7：small font,bold, underline, no highlight.

8：Large font,no bold, no underline, highlight.

9：small font,no bold, no underline, highlight.

10：Large font,bold, no underline, highlight.

11：small font,bold, no underline, highlight.

12：Large font,no bold, underline, highlight.

13：small font,no bold, underline, highlight.

14：Large font,bold, underline, highlight.

15：small font,bold, underline, highlight.

textSize: Font magnification.

[Range]:textSize=（0 To 7, 16 To 23, 32 To 39, 48 To 55, 64 To 71, 80 To 87, 96 To 103, 112 To 119；）

font hight multiple =textSize%8；

font width multiple=textSize/8;

Example:

Print.PrintText(“TEXT\n”,0,14,0)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.18 Print barcode

```
public static int PrintBarCode(int bcType,String bcData)
```

Function:

PrintBarCode(int bcType,String bcData)

bcType:type of bar code

m	Bar code system	Range of *n*	Range of *d*
65	UPC-A	$n = 11, 12$	$48 \le d \le 57$
66	UPC-E	$n = 11, 12$	$48 \le d \le 57$ [where $d1 = 48$]
67	JAN13 / EAN13	$n = 12, 13$	$48 \le d \le 57$
68	JAN8 / EAN8	$n = 7, 8$	$48 \le d \le 57$
69	CODE39	$1 \le n \le 255$	$48 \le d \le 57$, $65 \le d \le 90$, $d = 32, 36, 37, 42, 43, 45, 46, 47$
70	ITF	$2 \le n \le 254$ (even number)	$48 \le d \le 57$
71	CODABAR (NW-7)	$2 \le n \le 255$	$48 \le d \le 57$, $65 \le d \le 68$, $97 \le d \le 100$, $d = 36, 43, 45, 46, 47, 58$ [where $65 \le d1 \le 68$, $65 \le dn \le 68$, $97 \le d1 \le 100$, $97 \le dn \le 100$]
72	CODE93	$1 \le n \le 255$	$0 \le d \le 127$
73	CODE128	$2 \le n \le 255$	$0 \le d \le 127$ [where $d1 = 123$, $65 \le d2 \le 67$]

n indicates number of bytes of bar code data

d specifies bar code data

bcData: data of bar code

```
public static int PrintBarCode(int bcType,String bcData,int width,int height,int HRIPosition, int justification)
```

Function:

PrintBarCode(int bcType,String bcData,int width,int height,int

HRIPosition, int justification)

Parameter:

bcType: type of bar code

bcData: data of bar code

width:bar code width Range:(1-6)

	Width (mm)	Narrow Bar Code (mm)	Wide Bar Code (mm)
1	0.125	0.125	0.250
2	0.25	0.25	0.625
3	0.375	0.375	2.303
4	0.5	0.5	1.250
5	0.625	0.625	1.625
6	0.750	0.750	2

height:height of bar code range:1-255。

HRIPosition:

Select the HRI print position when printing bar code.

n	Print Position
0,48	No print
1,49	Above the bar code
2,50	Below the bar code
3,51	Both above and below the bar code

justification: justification method

0: left justifying

1: center

2: right justifying

Example:

Print.PrintBarCode(73,"{BS/N:{C\014\042\070\116{A3",1,50,2,0);//Print

code128:

S/N:123456783

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.19 Print 2D code

```
public static int PrintQRCode(String bcData)
```

Function:

PrintQRCode(String bcData)

Parameter:

bcData: data of 2D code

```
public static int PrintQRCode(String bcData,int sizeOfModule,int errorLevel,int justification)
```

Function:

PrintQRCode(String bcData,int sizeOfModule,int errorLevel,int

justification)

Parameter:

bcData: data of 2D code

sizeOfModule: size of 2D code range 1-16；

errorLevel: level of error correction

N	Function	Refer: recoverable character ratio
48	Select error correction L	7%
49	Select error correction M	15%
50	Select error correction Q	25%
51	Select error correction R	30%

justification: justification method

0: left justifying

1: center

2: right justifying

Example:

Print.PrintQRCode(“data of 2D code”,6,48,0)

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.20 Print bitmap

Function:

PrintBitmap(Bitmap bmp,int halftoneType,int luminance)

Parameter:

bmp: image object

halftoneType: algorithm type of image

0：Black-White.

1：Shake.

2：gater.

luminance：brightness （range: -100 To100）

Example:

Print.PrintBitmap(bmp,1,0)

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.21 Send data to the printer

Function:

int WriteData(**byte**[] bData)

Parameter:

bData：the data sent to the printer

ExamplePrint.WriteData(“123abc\n”.getBytes("GB2312"))//Send the data of ‘123abc’ to the printer.

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.22 Read data from the printer

Function:

byte[] ReadData(**int** time)

Parameter:

Time: time of timeout (Unit: millisecond)

Example:

Print.ReadData(2000)

//The data read from the printer.

Return:

The data returned from the printer, length = 0 no data returned from the printer.

3.23 Print PDF417

Function:

```
int PrintPDF417(String bcData,
                byte dataColumns,
                byte dataRows,
                byte moduleWidth,
                byte rowHeight,
                byte errorMode,
                byte errorLevel,
                byte options)
```

Parameter:

bcData: the content of data

dataColumns: sets the number of columns in data print area (range: 0-30).

0: automatic setting. The number of print columns is set according to the print range.

dataRows: sets the umber of rows of PDF417 (range: 0, 3-90).

0: automatic setting. The number of print rows is set according to the print range.

moduleWidth: sets the width of the module (range: 2-8).

rowHeight: sets the height of the module =n*width (range: 2-n-8).

errorMode: error correction mode

48: level mode

49: ratio mode

errorLevel: two modes (n)

Level mode:

n	Function	Error Correction Number of PDF417 Code
48	Select error correction level 0	2
49	Select error correction level 1	4
50	Select error correction level 2	8
51	Select error correction level 3	16
52	Select error correction level 4	32
53	Select error correction level 5	64
54	Select error correction level 6	128
55	Select error correction level 7	256
56	Select error correction level 8	512

Ratio mode: ［Data code ×n ×0.1＝(A)］(decimal part round-off)

A	Function	Error Correction Number of PDF417 Code
0~3	Select error correction level 1	4
4~10	Select error correction level 2	8
11~20	Select error correction level 3	16
21~45	Select error correction level 4	32
46~100	Select error correction level 5	64
101~200	Select error correction level 6	128
201~400	Select error correction level 7	256
Above 400	Select error correction level 8	512

Options: select the options

0: select standard PDF417

1: select compacted PDF417

Return:

≠-1: sending (to printer) success

＝-1: sending (to printer) failure

Example:

Print.PrintPDF417(“123456”,(byte)0,(byte)0,(byte)3,(byte)3,(byte)49,(byte)1,(byte)0)

3.24 Label location

Function:

int GotoNextLabel()

Note:

This command is only for label location, not applicable for continuous paper.

Example:

Print.GotoNextLabel()

//Locate to the gap of label paper.

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.25 Select page mode

```
public static int SelectPageMode()
```

Example:

Print.SelectPageMode()

Note: The printer should support page mode function.

Under page mode you can set the position which you want to print.

//Enter page mode

Print.SelectPageMode()

```
//Set print area
Print.SetPageModePrintArea(0,0,200,200)
//Set print direction
Print.SetPageModePrintDirection(0)
//Set position of x, y
Print.SetPageModeAbsolutePosition(0,0)
//Print 2D code (can also print text and bar code)
Print.PrintQRCode("abcdef",4,48,1)
//Print
Print.PrintDataInPageMode()
```

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.26 Set print area in page mode

int SetPageModePrintArea(**int** horizontal,**int** vertical,**int** width,**int** height)

Note: The printer should support page mode function. And it can take effect only when entering the page mode.

Parameter:

horizontal: x-coordinate of start point

vertical: y-coordinate of start point

width: width of the area

height: height of the area

Example:

```
//Enter page mode
Print.SelectPageMode()
//Set print area
Print.SetPageModePrintArea(0,0,200,200)
//Set print direction
Print.SetPageModePrintDirection(0)
//Set position of x, y
Print.SetPageModeAbsolutePosition(0,0)
//Print 2D code (can also print text and bar code)
Print.PrintQRCode("abcdef",4,48,1)
//Print
Print.PrintDataInPageMode()
```

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.27 Set print direction in page mode

int SetPageModePrintDirection(**int** direction)

Note: The printer should support page mode function. And it can take effect only when entering the page mode.

Parameter:

direction: print direction

0：0°

1：90°

2：180°

3：270°

Example:

```
//Enter page mode
Print.SelectPageMode()
//Set print area
Print.SetPageModePrintArea(0,0,200,200)
//Set print direction
Print.SetPageModePrintDirection(0)
//Set position of x, y
Print.SetPageModeAbsolutePosition(0,0)
//Print 2D code (can also print text and bar code)
Print.PrintQRCode("abcdef",4,48,1)
//Print
```

Print.PrintDataInPageMode()

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.28 Set print position of x, y in page mode

int SetPageModeAbsolutePosition(**int** xPosition, **int** yPosition)

Note: The printer should support page mode function. And it can take effect only when entering the page mode.

Parameter:

xPosition：X-coordinate

yPosition：Y-coordinate

Example:

//Enter page mode

Print.SelectPageMode()

//Set print area

Print.SetPageModePrintArea(0,0,200,200)

//Set print direction

Print.SetPageModePrintDirection(0)

//Set position of x, y

Print.SetPageModeAbsolutePosition(0,0)

//Print 2D code (can also print text and bar code)

Print.PrintQRCode("abcdef",4,48,1)

//Print

Print.PrintDataInPageMode()

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.29 Print in page mode

int PrintDataInPageMode()

Note: The printer should support page mode function. And it can take effect only when entering the page mode.

Example:

//Enter page mode

Print.SelectPageMode()

//Set print area

Print.SetPageModePrintArea(0,0,200,200)

//Set print direction

Print.SetPageModePrintDirection(0)

//Set position of x, y

Print.SetPageModeAbsolutePosition(0,0)

//Print 2D code (can also print text and bar code)

Print.PrintQRCode("abcdef",4,48,1)

//Print

Print.PrintDataInPageMode()

Return:

≠-1:sending (to printer) success

-1:sending (to printer) failure

3.30 Get NV bitmap list

int RefreshImageList(List<**byte**[]> lbImageIndex)

Note: Only when printer supports NV bitmap function can it take effect.

Parameter:

lbImageIndex: serial number of image list

Example:

Print.RefreshImageList(lbImageIndex)

Return:

-1: Printer does not support NV bitmap function.

1: Successfully get NV bitmap list.

3.31 Get NV bitmap memory capacity

int QueryNVStoreCapacity(**int**[] iSpace)

Note: Only when printer supports NV bitmap function can it take effect.

Parameter:

iSpace: memory capacity

Example:

iSpace=new int[1];

Print.QueryNVStoreCapacity(iSpace);

Return:

-1: Printer does not support NV bitmap function.

1: Successfully get NV bitmap list.

3.32 Get NV bitmap remaining memory capacity

int QueryNVStoreRemainingCapacity(**int**[] storeRemainingCapacity)

Note: Only when printer supports NV bitmap function can it take effect.

Parameter:

storeRemainingCapacity: remaining memory capacity

Example:

storeRemainingCapacity=new int[1];

Print.QueryNVStoreRemainingCapacity(storeRemainingCapacity);

Return:

-1: Printer does not support NV bitmap function.

1: Successfully get NV bitmap list.

3.33 Print NV bitmap

int PrintNVImage(String imageNo,**int** scaleMode)

Note: Only when printer supports NV bitmap function can it take effect.

Parameter:

imageNo: serial number of image

scaleMode: mode (default: 0)

Example:

Print.PrintNVImage(imageNo,0);

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.34 Delete specified NV bitmap

int DeleteSpecifiedNVImage(String sImageIndex)

Note: Only when printer supports NV bitmap function can it take effect.

Parameter:

sImageIndex: serial number of image

Example:

Print.DeleteSpecifiedNVImage(sImageIndex);

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.35 Delete all NV bitmap

int DeleteAllNVImage()

Note: Only when printer supports NV bitmap function can it take effect.

Example:

Print.DeleteAllNVImage();

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.36 Download NV bitmap to the printer

int DefineNVImage(String[] sArrFile, Handler handler)

Note: Only when printer supports NV bitmap function can it take effect.

Parameter:

sArrFile: image path

Handler: Handler object

message.what: maximum number of data package

mesage.arg1: download progress

Example:

Print.DefineNVImage(sArrFile,handler);

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.37 Print Binary file

boolean PrintBinaryFile(String strPRNFile)

Parameter:

strPRNFile: bin file path

Example:

Print.PrintBinaryFile(strPRNFile);

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.38 Get printer function list

int CapturePrinterFunction(**int** ModelPropertyKeyBeep,

int[] propType, **byte**[] value, **int**[] dataLen)

Parameter:

ModelPropertyKeyBeep：function code

MODEL_PROPERTY_KEY_BEEP：beeper

MODEL_PROPERTY_KEY_CUT：cut paper

MODEL_PROPERTY_KEY_DRAWER：drawer

MODEL_PROPERTY_KEY_BARCODE：bar code

MODEL_PROPERTY_KEY_PAGEMODE：page mode

MODEL_PROPERTY_KEY_GET_REMAINING_POWE：power

MODEL_PROPERTY_CONNECT_TYPE：connection type

MODEL_PROPERTY_KEY_PRINT_RECEIPT：receipt

propType: type number

Value: whether to support

(beeper, cut paper, drawer, page mode, power, receipt)

Value[0]==0 support.

Otherwise nonsupport

(Bar code)

String barcode =new String(Value);

barcode contains QRCODE, support 2D code

barcode contains PDF417, support PDF417

dataLen: length of return data

Example:

```
int[] propType=new int[1];
byte[] Value=new byte[500];
int[] DataLen=new int[1];
Print.CapturePrinterFunction(ModelPropertyKeyBeep,propType,Value,Da
taLen);
```

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.39 Clear page mode print area data

int ClearPageModePrintAreaData()

Example:

Print.ClearPageModePrintAreaData();

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.40 Print and return standard mode

int PrintAndReturnStandardMode()

Example:

Print.PrintAndReturnStandardMode();

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.41 Set the left margin

```
int SetLeftMargin(int iLeftMargin)
```

Parameter:

iLeftMargin： Left margin (unit px)

Example:

Print.PrintAndReturnStandardMode();

Return:

≠-1: sending (to printer) success

-1: sending (to printer) failure

3.42 Read magnetic card information

Note: This feature is only supported on printers that have a magnetic card feature.

```
void setTrackCardReaderMode(int track,CardReader
cardReader,int outTime)
```

Parameter:

track： Track (range: 1-5).

CardReader： The data is returned to the interface.

Succeed(byte[] data);//The data returned.

Failure(int error);

//error--> 1:Connection disconnected, 2: timeout, 3: other errors.

outTime： Timeout (in milliseconds).

Example：

```
Print.setTrackCardReaderMode(track,new
Print.CardReader() {
  @Override
  public void Succeed(final byte[] data) {
```

```
            }
        @Override
        public void Failure(final int error) {
            }
    },30*1000);
```

3.43 Exit magnetic card mode

```
boolean CancelTrackCardReaderMode()
```

Example:

```
Print.CancelTrackCardReaderMode();
```

Return:

true: success。

false: failure。

3.44 Set mobile unit

```
int setPrintResolution(int x,int y)
```

Note:

his interface is a mobile unit that sets the x-axis and y-axis.

unit: 25.4/ x mm,

25.4/ y　mm。

range：（0-255）。

Return：

> 0: sending (to printer) success

-1: sending (to printer) failure

-2:Parameter error

Example：

Print.setPrintResolution(203,203);

3.45 Print rectangle

```
int PrintPageRectangle(int x,int y,int width,
int height,int lineWidth)
```

Note：

This interface is only supported by some printers.

unit：PX。

Parameter ：

x： The top left x coordinate.

y: The top left y coordinate.

width： The width of the rectangle.

height： The height of the rectangle.

lineWidth： Line width.

Return：

≠-1: sending (to printer) success

-1: sending (to printer) failure

Example：

```
Print.PrintPageRectangle(0,0,100,100,2);
```

3.46 Print Line

```
int PrintPageLine(int x1,int y1,int x2,int y2,
int lineWidth)
```

Note：

This interface is only supported by some printers.

unit： PX。

Parameter ：

x1: Starting x coordinate.

y1: Starting y coordinate.

x2: End x coordinate.

y2: End y coordinate.

lineWidth： Line width.

Return：

≠-1: sending (to printer) success

-1: sending (to printer) failure

Example：

Print.PrintPageLine(0,0,100,100,2);

3.47 Image data compression print

`int` PrintBitmapLZO(`Bitmap` bitmap,`int` halftoneType)

Note：

This interface is only supported by some printers, and the printer prints the original size of the image.

8 px=1 mm。

Parameter ：

bitmap： image object.

halftoneType: Image algorithm type.

0： Binary algorithm.

1： Halftone algorithm.

Return：

≠-1: sending (to printer) success

-1: sending (to printer) failure

Example：

Print.PrintBitmapLZO(bitmap,0);

Table 1-1

Name	Character Set	Code page
Default	0	gb2312
Chinese Simplified	0	gb2312
Chinese Traditional	0	big5
PC437(USA)	0	iso8859-1
KataKana	1	Shift_JIS
PC850(Multilingual)	2	iso8859-3
PC860(Portuguese)	3	iso8859-6
PC863(Canadian-French)	4	iso8859-1
PC865(Nordic)	5	iso8859-1
PC857(Turkish)	13	IBM857
PC737(Greek)	14	iso8859-7
ISO8859-7(Greek)	15	iso8859-7
WCP1252	16	iso8859-1
PC866(Cyrillic #2)	17	iso8859-5
PC852(Latin 2)	18	iso8859-2
PC858(Euro)	19	iso8859-15
KU42	20	ISO8859-11
TIS11(Thai)	21	ISO8859-11
TIS18(Thai)	26	ISO8859-11
PC720	32	iso8859-6
WPC775	33	iso8859-1
PC855(Cyrillic)	33	iso8859-5
PC862(Hebrew)	36	iso8859-8
PC864(Arabic)	37	iso8859-6
ISO8859-2(Latin2)	39	iso8859-2
ISO8859-15(Latin9)	40	iso8859-15
WPC1250	45	iso8859-2
WPC1251(Cyrillic)	46	iso8859-5
WPC1253	47	iso8859-7
WPC1254	48	iso8859-3
WPC1255	49	iso8859-8
WPC1256	50	Windows-1256
WPC1257	51	iso8859-1
WPC1258	52	bg2312
MIK(Cyrillic/Bulgarian)	54	iso8859-15
CP755	55	iso8859-5
Iran	56	iso8859-6
Iran II	57	iso8859-6
Latvian	58	iso8859-4

ISO-8859-1(West Europe)	59	iso8859-1
ISO-8859-3(Latin 3)	60	iso8859-3
ISO-8859-4(Baltic)	61	iso8859-4
ISO-8859-5(Cyrillic)	62	iso8859-5
ISO-8859-6(Arabic)	63	iso8859-6
ISO-8859-8(Hebrew)	64	iso8859-8
ISO-8859-9(Turkish)	65	iso8859-9
PC856	66	iso8859-8
ABICOIM	67	iso8859-15