Developer Documentation of SUNMI's Ticket Printer

Revision Records

Version	Update date	Contents	Writer
V1.0.0	2020.9.14	The original version	Li Jiang
V1.1.4	2020.11.5	Added the full-parameter version of printText's interface; Optimized the coordinates after rotation.	Li Jiang

Introduction

Follow these instructions to use your self-service ticket printer through the API provided by SUNMI.

Connection Methods

This printer provides the AAR file, and you can use the standard steps to import.

You can refer to the demo source code to get the steps to call interface. The pseudocode is:

```
SdkManager.getInstance().initialization(getApplicationContext());
BasePrinter printer = SdkManager.getInstance().getPrinter();
if(printer.isOpened()){
    printer.init();
    printer.startTransBuffer(...);
    printer.printText(...);
    printer.printBitmap(...);
    printer.printBarcode(...);
    ...
    printer.endTransBuffer();
    printer.cutPaper(...);
}
```

When you exit the program, call onDestroy():

```
SdkManager.getInstance().destroy();
```

API

1. Initialize the printer

Method	int init()
Description	You can initialize the printer settings.
Parameter	None
Return	See Return Values

2. Turn off the printer

Method	int close()
Description	You can turn off the printer and disconnect the USB port.
Parameter	None
Return	See Return Values

3. Check whether the printer is connected

Method	boolen isOpened()
Description	You can check whether the printer has established connection and communication.
Parameter	None
Return	See Return Values

4. Get status of the printer

Method	int getStatus()
Description	You can get status of the printer.
Parameter	None
Return	See Table of the Printer's Status

5. Get real-time status of the printer

Method	int getStatus(StatusListener listener)
Description	You can get the printer's status asynchronously, with a default interval of 1s.
Parameter	See Callback of the Printer's Status
Return	See Return Values

6. Get Callback of the printer's status

Method	void onStatusChanged(int status)

Description	The callback is triggered when the printer's status changes.
Parameter	Status: See Table of the Printer's Status
Return	None

7. Start transaction printing

Method	int startTransBuffer(int w, int h)
Description	You can start printing things and set the print size of the data in the buffer.
Parameter	Print width (w): in dot; Data beyond the area will be discarded. Print height (h): The same as above.
Return	See Return Values

8. Finish transaction printing

Method	int endTransBuffer()
Description	The contents in the buffer are printed and the printing is over.
Parameter	None
Return	See Return Values

9. Print texts

Method	int printText(int x, int y, int rotation, String text)
Description	You can send texts to the buffer.
Parameter	Abscissa (x) Ordinate (y) Rotation: The effective parameters are 0, 90, 180, and 270. Text
Return	See Return Values

10. Print texts (full-parameter)

Method	int printText(int x, int y, String font, int rotation, int x_multiplication, int y_multiplication, int aligment, String text)
Description	You can send texts to the buffer.
Parameter	Abscissa (x) Ordinate (y) Font: The Chinese font is "simsun.TTF." Other fonts do not support Chinese. See the command set for details. Rotation: Rotate around (x, y); The effective parameters are 0, 90, 180, 270. X_multiplication: The effective parameters are above 1, and the common value is 10. Y_multiplication: The same as above. Alignment: Align with (x, y); 1 left-aligned, 2 centered, 3 right-aligned. Text
Return	See Return Values

11. Print pictures

Method	int printBitmap(int x, int y, Bitmap bitmap)
Description	You can send pictures to the buffer.
Parameter	Abscissa (x) Ordinate (y) Bitmap
Return	See Return Values

12. Print 1 D barcodes

Method	int printBarcode(int x, int y, String type, int height, int human_readable, int rotation, String data)
Description	You can send barcodes to the buffer.
Parameter	Abscissa (x) Ordinate (y) Type: See the instructions set for details. Height: in dot Human_readable: The effective parameters are: 0 Code is not displayed, 1 Code is displayed in left alignment, 2 Code is displayed in a centered manner, and 3 Code is displayed in right alignment. Rotation: The effective parameters are 0, 90, 180, 270. Data: See the command set for detailed limitations.
Return	See Return Values

13. Print 2D barcodes

Method	int printQrCode(int x, int y, String ecc, int cell, int rotation, String data)
Description	You can send QR codes to the buffer.
Parameter	Abscissa (x) Ordinate (y) ECC: The effective parameters of level of correctness are L: 7%, M: 15%, Q: 25%, H: 30% Cell: The effective parameters of are 1-10. Rotation: The effective parameters are 0, 90, 180, 270. Data: See the command set for detailed limitations.
Return	See Return Values

14. Feed paper by pixel

Method	int pixelWrap(int n)
Description	Paper is fed by pixel in dot.
Parameter	Line (n)
Return	See Return Values

15. Cut paper

Method	int cutPaper(int m, int n)

Description	You can cut paper.
Parameter	Cutting Method: (This module only supports the third method.) The effective parameters are 0 full-cut, 1 half-cut, 2 Cutting paper when the paper moves forward a certain distance Distance (n): in dot
Return	See Return Values

16. Send commands

Method	Int sendCommand(byte[] cmd)
Description	You can send commands.
Parameter	CMD see the command set.
Return	See Return Values

17. Send commands

Method	int sendCommand(String cmd)
Description	You can send commands.
Parameter	CMD see the command set.
Return	See Return Values

Appendix

Return Values

>=0 Implementation is successful. >0 indicates the length of data successfully sent.
-1 The printer is offline or not ready.
-2 The printer cannot receive print data due to full cache.
-3 Errors occur in sending data.
-4 Errors occur in sending the command or parameter.

Table of the Printer's Status

```
public class PrinterStatus {
   //Offline
    public static final int STATUS_OFFLINE = -1;
    //Ready
    public static final int STATUS_READY = 0;
    //Cover Open
    public static final int STATUS_HEAD_OPEN = -2;
    //Printer ribbon is stuck.
    public static final int STATUS_RIBBON_JAM = -3;
    //Out of printer ribbon
    public static final int STATUS_RIBBON_EMPTY = -4;
    //Print paper is stuck.
    public static final int STATUS_PAPER_JAM = -5;
    //Out of print paper
    public static final int STATUS_PAPER_EMPTY = -6;

      //Cutting
      public static final int STATUS_CUTTING = -7;
      //Waiting for pressing the PRINT button (Press to print mode)
      public static final int STATUS_WAITING_PRINT_KEY = -8;
      //Waiting for removing the label paper (Peel-off mode)
      public static final int STATUS_WAITING_TAKE_LABEL = -9;
      //Printing
      public static final int STATUS_PRINTING_BATCH = -10;
      //Paused
    public static final int STATUS_PAUSE = -11;
}
```