

ZONERICH PRINTER Android SDK Help Doc

----- 2018.4.17

ZONERICH PRINTER Android SDK Help Doc	1
I:Introduction.....	4
II:Constants & functions	5
1. Constant definition.....	5
2. Common functions.....	8
1).SDK_Version.....	8
2).Prn_GetPortList	9
3).Prn_Connect.....	9
4).Prn_Disconnect	10
3.Thermal printer functions.....	11
1).Prn_PrinterInit	11
2).Prn_Status.....	11
3).Prn_PowerStatus	12
4).int Prn_PrintText	12
5).Prn_PrintEscText.....	14
6).Prn_PrintString.....	15
7).Prn_PrintBarcode.....	15
8).Prn_PrintQRCode.....	16
9).Prn_PrintBitmap.....	17
10).Prn_PrintBmp.....	18
11).Prn_CutPaper.....	19
12).Prn_CutPaper.....	20
13).Prn_OpenCashbox.....	20
14).Prn_LineFeed.....	21
15).Prn_MarkFeed.....	21
16).Prn_SetCharacterSet	22
17).Prn_SetInterCharacterSet	22
18).Prn_SetLineSpacing	23
19).Prn_SetFontStyle	23
20).Prn_SetFontSize.....	24
21).Prn_SetAlignment	25
22).Prn_SendData.....	26
23).Prn_ReadData.....	26



24).Prn_BeginTransaction.....	27
25).Prn_EndTransaction	27
26).Prn_GetMsrTrack.....	28
4.Label printer functions	29
1).prn_PageSetup.....	29
2).prn_PagePrint.....	29
3).prn_PageClear.....	29
4).prn_PrinterStop.....	30
5).prn_DrawBox.....	30
6).prn_DrawLine	30
7).prn_DrawInverseLine	31
8).prn_DrawText	31
9).prn_DrawTextAlign	34
10).prn_DrawBarcode.....	34
11).prn_BarcodeText.....	36
12).prn_PrintBitmap.....	36
13).prn_PrinterStatus.....	37
14).prn_AddLabelCommand.....	37
15).prn_DrawWatermark.....	37
5.Barcode printer functions.....	38
1).bar_SetSize.....	38
2).bar_SetGap.....	39
3).bar_SetBline	40
4).bar_SetOffset.....	40
5).bar_SetShift	41
6).bar_SetSpeed	41
7).bar_SetDensity	42
8).bar_SetDirection.....	42
9).bar_SetPeel.....	43
10).bar_SetCharacterSet	44
11).bar_CLS.....	44
12).bar_Print.....	44
12).bar_DrawBar	45
13).bar_Barcode	45
14).bar_QRCode.....	46
15).bar_Bitmap.....	47
16).bar_DrawBox	48
17).bar_PutPCX.....	48
18).bar_SetReverse	48
19).bar_Text	49
20).bar_DownloadBAS.....	50

21) .bar_DownloadFile	50
22) .bar_KillFile.....	51
23) .bar_RunFile.....	51
24) .bar_SetCounter.....	52
25) .bar_PrintCounter	52
III:Samples	53
IV:Others	54
1. Access port permission problem.	54
2. Batch mode.....	55

I:Introduction

1. Name: ZQPrinterSDK

2. Develop environment:

Eclipse

Android SDK 12-19

3. Lib files:

ZQPrinterSDK.jar

Armeabi\libzqcom.so (Note: this file could be ignored if the RS232 and USB ports are not used)

4.Lib files usage in Projects

1) Open a new Android project

2) Add file ZQPrinterSDK.jar and directory Armeabi in project *libs* directory (note: if there is no *libs* directory, make a new directory *libs*), then Refresh the project in Eclipse.

3) Terminal Printer, Define a variable *private* private ZQPrinterSDK prn = null, and add statement prn = new ZQPrinterSDK () in a proper position to invoke functions.

4) Label Printer, Define a variable *private* private ZQLabelSDK prn = null, and add statement prn = new ZQLabelSDK () in a proper position to invoke functions.

5) Barcode Printer, Define a variable *private* private ZQBarcodeSDK prn = null, and add statement prn = new ZQBarcodeSDK () in a proper position to invoke functions.



II:Constants & functions

1. Constant definition

Category	Name	Value	Description
Return value			
ErrorCode	SUCCESS	0	Successful
	PORTERROR	-1	Port error
	WRITEERROR	-2	Write error
	READERROR	-3	Read error
	NO PERMISSION	-4	No permission
	INVALIDPARAM	-5	Invalid parameter
	UKERROR	-99	Unknown error
Text alignment			
Alignment	LEFT	0	Left alignment
	CENTER	1	Center alignment
	RIGHT	2	Right alignment
Font Style			
Font	DEFAULT	0	Default
	FONTB	1	B Font
	BOLD	2	Bold
	UNDERLINE	4	Underline
	REVERSE	8	Black and white reversal
Font width			
WidthSize	SIZE0	0	Normal
	SIZE1	16	X 2
	SIZE2	32	X 3
	SIZE3	48	X 4
	SIZE4	64	X 5
	SIZE5	80	X 6
	SIZE6	96	X 7
	SIZE7	112	X 8
Font height			
HeightSize	SIZE0	0	Normal
	SIZE1	1	X 2
	SIZE2	2	X 3

	SIZE3	3	X 4
	SIZE4	4	X 5
	SIZE5	5	X 6
	SIZE6	6	X 7
	SIZE7	7	X 8
Code page			
PageCode	PC437	0	Code page PC437
	KATAKANA	1	Katakana
	PC850	2	Code page PC850
	PC860	3	Code page PC860
	PC863	4	Code page PC863
	PC865	5	Code page PC865
	WPC1253	6	Code page WPC1252
	TRAN	7	Code page TRAN
	WPC1256	8	Code page WPC1256
	PC737	9	Code page PC737
	WPC1250	10	Code page WPC1250
	WPC1252	16	Code page WPC1252
	PC866	17	Code page PC866
	PC852	18	Code page PC852
	PC858	19	Code page PC858
	USER	255	User set pages
Printer status			
PrinterStatus	ONLINE	0	Online
	COVEROPEN	-1	Cover is open
	EMPTY PAPER	-2	No paper
	OFFLINE	-3	Offline
	OTHER ERROR	-4	Other error
One dimensional bar code type			
Barcode	EAN13	103	
	JAN13	104	
	EAN8	105	
	JAN8	106	
	Code39	107	
	Code128	111	
	Code128_ZQ	112	
Barcode text position			
BarcodeText	TEXT_NONE	0	Not printed
	TEXT_ABOVE	1	Above the bar code
	TEXT_BELOW	2	Below the bar code

Power status			
PowerStatus	HIGH	700	Power high
	MIDDLE	701	Power middle
	LOW	702	Power low
	SMALL	703	Power small
	NO	704	Power no
Two dimensional barcode data type			
QRCode	NUM	49	Numeric
	ALPH	50	Alphanumeric
	CHN	51	Chinese
Bitmap size			
BitmapSize	ZQSIZE0	1	there might be some differences of different printers, Please take the actual effect as the standard
	ZQSZIE1	2	
	ZQSIZE2	3	
	ZQSIZE3	4	
	SIZE0	5	
	SIZE1	6	
	SIZE2	7	
	SIZE3	8	
MSR track			
MsrTrack	TRACK1	1	Manual-mode, Read track1 data
	TRACK2	2	Manual-mode, Read track2 data
	TRACK3	3	Manual-mode, Read track3 data
	TRACK12	4	Manual-mode, Read track1 & track2 data
	TRACK23	5	Manual-mode, Read track2 & track3 data
	TRACK123	6	Auto-mode, Read track1/track2/track3 data
Msr track data maximum length			
MsrTrackLen	LEN_TRACK1	84	
	LEN_TRACK2	45	
	LEN_TRACK3	112	
	LEN_TRACK12	122	
	LEN_TRACK23	150	
	LEN_TRACK123	242	

Connection status			
ConnectStatus	CONNECTED	0	Online
	DISCONNECTED	-1	Offline
Cashbox position			
Cashbox	PIN3	0	Default connection at PIN3.
	PIN5	1	Connection at PIN5.
Cutter option			
CutMode	FULLCUT	49	Full cut
	PARTIALCUT	50	Partial cut
Handshaking			
FlowControl	NONE	0	NONE
	XONXOFF	1	XON/XOFF
	HARDWARE	2	DTR/DSR
Port type			
PortType	WIFI	0	WIFI or NET
	BLUETOOTH	1	Bluetooth
	USB	2	USB
	COM	3	Serial or USB to serial
Units			
Unit	UNIT_INCH	0	English system (inch)
	UNIT_MM	1	Metric system (mm)
	UNIT_DOT	2	Dot measurement
Graphic modes			
ImgMode	MODE_OVERWRITE	0	OverWrite
	MODE_OR	1	Or
	MODE_XOR	2	Xor

2. Common functions

1).SDK_Version

String SDK_Version()

Description: Read SDK version info

Parameter: NULL

Return: SDK version, e.g 1.0



Copyright©2015 All Rights Reserved.

Note: NULL

2) .Prn_GetPortList

```
String[] Prn_GetPortList(int nPortType, Context context)
```

Description:Get the port list.

Parameter:

Int nPortType:Port type, as below:

Value		Description
PortType.WIFI	0	WIFI or NET
PortType.BLUETOOTH	1	Bluetooth
PortType.USB	2	USB
PortType.COM	3	Serial or USB to serial

Context context: Activity of context, usb port type can not be null, the other port types may be null.

Return: Device name.

Note:NULL.

3) .Prn_Connect

```
int Prn_Connect(String strAddr, Context context)
```

Description: connects to printer

Parameter:

String strAddr: printer port parameter, format is as follows:

WIFI or NET	IP:Port number (Generally is 9100)	e.g: 192.168.1.10:9100	Note: for some printer models with Ethernet/WIFI function, if the host machine didn't communicate with printer for a certain time, the printer will disconnect
-------------------	---	---------------------------	---

			automatically. Then the host will need to re-connect with printer.
Bluetooth/BLE	Bluetooth address	e.g: 00:12:6F:33:00:A5	
USB	The node name under directory /dev. Directory dev does not included.	e.g: USB0、USB1...	Note: some of the USB printers does not support data return function. To read data from these interfaces won't work, e.g. Prn_Status, Prn_ReadData.
Serial or USB to serial	Serial name:baudrate	e.g: ttyS0:9600 ttyUSB0:9600	Note: some of the USB printers does not support data return function. To read data from these interfaces won't work, e.g. Prn_Status, Prn_ReadData.

Context context: Activity of context, usb port type can not be null, the other port types may be null.

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.INVALIDPARAM	Printer parameter invalid
ErrorCode.PORTERROR	Port error
ErrorCode.NO PERMISSION	No permission

Note: Please refer to part IV (Others • Access port permission problem)

4) .Prn_Disconnect

```
int Prn_Disconnect()
```

Description: Disconnect printer

Parameter: NULL

Return: **ErrorCode.SUCCESS:** successful.

Note:

3.Thermal printer functions

1).Prn_PrinterInit

```
int Prn_PrinterInit()
```

Description: to initialize the printer. Configurations to the printer will be reset to default values, e.g. printing height, width, line space etc. It corresponds to command 1B 40.

Parameter: NULL

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Prn_Connect failure
ErrorCode.WRITEERROR	Send data to printer failed

Note:

2).Prn_Status

```
int Prn_Status()
```

Description: return printer status.

Parameter: NULL

Return:

PrinterStatus.ONLINE	OK
PrinterStatus.COVEROPEN	Cover is open
PrinterStatus.EMPTY PAPER	No paper
PrinterStatus.OFFLINE	Offline

PrinterStatus.OTHERERROR

Other error

Note:

1. When the Bluetooth is in sleep mode, to check the status will get "paper out" feedback and awake Bluetooth. The correct status will be available only at the second check.

2. This function is just to check general status of the printer. As most printer models only have one sensor. There is no difference of "Cover Open" and "Paper Out". "Paper Out" will be feedback in both situations. To get more details of the printer status, command "0x10 0x04 n (n could be 0x01, 0x02, 0x03, 0x04)" need to be sent to printer, then by judging the returns to get the printer status.

3) .Prn_PowerStatus

```
int Prn_PowerStatus()
```

Description: return the power status.

Parameter: NULL

Return:

PowerStatus.HIGH	Power high
PowerStatus.MIDDLE	Power middle
PowerStatus.LOW	Power low
PowerStatus.SMALL	Power small
PowerStatus.NO	No power
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: only AB-3x0M series printers support this function.

4) .int Prn_PrintText

```
int Prn_PrintText(String Data, int Alignment, int Attribute,
int TextSize)
```

Description: text printing function.

Parameter:

String Data: Pointer to the character string to print, generally the ASCII code of "Enter" should be included at the end of the content line. Otherwise the printer will perform printing only after the fulfill of the characters lines.

int Alignment: Text alignment:

Value		Description
Alignment.LEFT	0	Left alignment
Alignment.CENTER	1	Center alignment
Alignment.RIGHT	2	Right alignment

int Attribute: Text properties. The following values can be duplicated:

Value		Description
Font.DEFAULT	0	FontA, 12x24
Font.FONTB	1	FontB, 9x17, Valid only for ASCII characters
Font.BOLD	2	Adds Bold lettering attribute
Font.UNDERLINE	4	Adds Underlining attribute
Font.REVERSE	8	Adds Reverse lettering attribute

Note: there might be some differences of different printers. for example, the underline will be no difference in inverse printing

int TextSize: Text size properties. The width and height values can be duplicated:

Value		Description
WidthSize.SIZE0	0	Width magnification set to x1
WidthSize.SIZE1	16	Width magnification set to x2
WidthSize.SIZE2	32	Width magnification set to x3
WidthSize.SIZE3	48	Width magnification set to x4
WidthSize.SIZE4	64	Width magnification set to x5
WidthSize.SIZE5	80	Width magnification set to x6
WidthSize.SIZE6	96	Width magnification set to x7
WidthSize.SIZE7	112	Width magnification set to x8
HeightSize.SIZE0	0	Height magnification set to x1
HeightSize.SIZE1	1	Height magnification set to x2
HeightSize.SIZE2	2	Height magnification set to x3
HeightSize.SIZE3	3	Height magnification set to x4
HeightSize.SIZE4	4	Height magnification set to x5

HeightSize.SIZE5	5	Height magnification set to x6
HeightSize.SIZE6	6	Height magnification set to x7
HeightSize.SIZE7	7	Height magnification set to x8

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: n/a

5).Prn_PrintEscText

```
int Prn_PrintEscText(String strPrint)
```

Description: Print ESC Sequence.**Parameter:**

String strPrint: String data including the ESC Sequence:

Command	Description
ESC [#]P	Cut paper, The placeholder '#' is replaced by "100" is full-cut, replaced by other ASCII Decimal string is Partial-cut.
ESC [#]fP	Feed the paper forward by 6 lines and cut paper. The placeholder '#' is replaced by "100" is full-cut, replaced by other ASCII Decimal string is Partial-cut.
ESC #B	Prints the prestored image, The placeholder '#' is replaced by the bitmap.
ESC [#]lF	Feed the paper forward by lines. The placeholder '#' is replaced by an ASCII Decimal string denotes the number of lines to be feed. If '#' is omitted, then Feeding is set to be one line.
ESC bC	Prints in bold or double strike
ESC uC	Prints with underline
ESC 1C	Prints normal size
ESC 2C	Prints double wide characters
ESC 3C	Prints double high characters
ESC 4C	Prints double high/double wide characters
ESC #hC	Prints with the width scaled '#' times the normal size, where '#' is replaced By an ASCII decimal string. Range is 0-8
ESC #vC	Prints with the height scaled '#' times the normal size, where '#' is replaced By an ASCII decimal string. Range is 0-8

ESC cA	Aligns following text in the center
ESC rA	Aligns following text in the right
ESC lA	Aligns following text in the left.
ESC N	Restores printer characteristics to normal condition

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: n/a

6) .Prn_PrintString

```
int Prn_PrintString(String strPrint)
```

Description: Print text string.**Parameter:**

String strPrint: Pointer to the character string to print.

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: n/a

7) .Prn_PrintBarcode

```
int Prn_PrintBarcode(String strBarcode, int Symbology, int
Height, int Width, int TextPosition)
```

Description: Print one dimensional bar code.**Parameter:**

String strBarcode: Data of barcode, the characters supported differ from the selected barcode types.

int Symbology: Defines the barcode type:



Type	Number of Data	Data Value Range
Barcode.EAN8 Barcode.JAN8	$7 \leq n \leq 8$	$48 \leq \text{data} \leq 57$
Barcode.EAN13 Barcode.JAN13	$12 \leq n \leq 13$	$48 \leq \text{data} \leq 57$
Barcode.CODE39	$1 \leq n \leq 255$	$48 \leq \text{data} \leq 57$ $65 \leq \text{data} \leq 90$ Data=32 36 37 43 45 46 47
Barcode.CODE128	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$
Barcode.CODE128_ZQ	$2 \leq n \leq 255$	$0 \leq \text{data} \leq 127$

Note: some printers support only standard CODE128 or ZONERICH CODE128, or both of the types.

int Height: Barcode height in Dot units, The value range is 1-255.

int Width: Barcode width according to a value range of 2~7.

Note: If the print Area of the barcode exceeds the printing paper, barcode printing may not be possible.

int TextPosition: Print position of the barcode data:

Value	Description
BarcodeText.NONE 0	Not Printed
BarcodeText.ABOVE 1	Above the bar code
BarcodeText.BELOW 2	Below the bar code

Note: Some printer is not supported, e.g AB-321M V1.0B2, JAN8 and JAN13 is fixed to print under barcode. For CODE39 and CODE128, they are fixed to not print text.

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.INVALIDPARAM	Invalid parameter
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: if the printer supports barcode alignments, Prn_SetAlignment could be used to configure the barcodes. Pay attention that some printers don't support barcode alignment, right alignment or center alignment may result in exceeding of the printing ranges.

8) .Prn_PrintQRCode


```
int Prn_PrintQRCode(int nMode, String strCode)
```

Description: Print two dimensional bar code

Parameter:

int nMode: two dimensional bar code type:

Value		Description
QRCode.NUM	0	Numeric
QRCode.ALPH	1	Alphanumeric
QRCode.CHN	2	Chinese

String strCode: Barcode data

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.INVALIDPARAM	Invalid parameter
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note: this function is not applicable for printers which don't support 2D barcode printing.

9) .Prn_PrintBitmap

```
int Prn_PrintBitmap(Bitmap image, int Type)
```

Description: Print image (Using 1D 76 Command)

Parameter:

Bitmap image: Android bitmap data.

int Type: print format: (Note: there might be different printing effects for different printers.)

Value		Description
BitmapSize.ZQSIZE0	0	Normal
BitmapSize.ZQSIZE1	1	Double-width
BitmapSize.ZQSIZE2	2	Double-height
BitmapSize.ZQSIZE3	3	Quadruple

Return:



<code>ErrorCode.SUCCESS</code>	Successful
<code>ErrorCode.INVALIDPARAM</code>	Invalid parameter
<code>ErrorCode.PORTERROR</code>	Port error
<code>ErrorCode.WRITEERROR</code>	Send data to priner failed

Note:

1. Printer support only monochrome bitmap print.
2. If the printer support Bitmap image alignment, please invoke `Prn_SetAlignment` to configure the alignment. If the printer don't support Bitmap image alignment, to align the images, please add some spaces at the left of the image.
3. Because of the compatibility of different printers, if the printer prints messy code, please try to invoke `Prn_PrintBMP1B2A` instead.

10).`Prn_PrintBmp`

```
int Prn_PrintBmp(String strFileName, int Type)
```

Description: Print image (Using 1B 2A Command)

Parameter:

String strFileName:Windows BMPmonochrome bitmap file full path

int Type: print format: (Note: different printer models may got different printing results.)

Value		Description
<code>BitmapSize.ZQSIZE0</code>	0	8-dot single-density, Width X 2,Height X 3
<code>BitmapSize.ZQSIZE1</code>	1	8-dot double-density, Width X 1,Height X 3
<code>BitmapSize.ZQSIZE2</code>	2	24-dot single-density, Width X 2,Height X 1
<code>BitmapSize.ZQSIZE3</code>	3	24-dot double-density, Normal size
<code>BitmapSize.SIZE0</code>	4	8-dot single-density, Width X 2,Height X 3
<code>BitmapSize.SIZE1</code>	5	8-dot double-density, Width X 1,Height X 3



广州市中崎商业机器有限公司
Zonerich Business Machine CO.,LTD

BitmapSize.SIZE2	6	24-dot single-density, Width X 2, Height X 1
BitmapSize.SIZE3	7	24-dot double-density, Normal size

Note: most of ZONERICH printers don't need to add 0A at the end of command 1B 2A to make transferring. But AB-88H do. And SIZE0~SIZE3 compare to ZQSIZE0~ZQSIZE3 is that 0A is added at the end of the command. So SIZE0~SIZE3 is applicable for AB-88H, and ZQSIZE0~ZQSIZE3 is applicable for other general printer models. For AB-88H, not only 0A should be added, it should be configured with proper line space. Otherwise the Bitmap printing will get overlap or unexpected space. And the line space should be reconfigured for text printing after the image printing.

int Alignment: Image alignment:

Value		Description
Alignment.LEFT	0	Left alignment
Alignment.CENTER	1	Center alignment
Alignment.RIGHT	2	Right alignment

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.INVALIDPARAM	Invalid parameter
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note:

11) .Prn_CutPaper

```
int Prn_CutPaper()
```

Description: Cut paper

Parameter: NULL

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note: this function is not valid for printers without auto cutter.

12) .Prn_CutPaper

```
int Prn_CutPaper(int nCutMode)
```

Description: Cut paper

Parameter: int nCutMode: Cutter option:

Value		Description
CutMode.FULLCUT	0x30	Full-cut
CutMode.PARTIALCUT	0x31	Partial-cut

Note: some printers though come with auto cutter, the cutter only performs partial cut or full cut, whatever parameter been transferred, the printer will only perform partial cut or full cut.

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: this function is not applicable for printers without auto cutter. For some old models, please invoke Prn_CutPaper() to perform cutting. If the printer supports partial or full cut, please use Prn_SendData to perform cutting.

13) .Prn_OpenCashbox

```
int Prn_OpenCashbox(int iWitch)
```

Description: Open cashbox (Default is Cashbox.PIN3)

Parameter:

int iWitch: to open the specified cash drawer.

Value		Description
Cashbox.PIN3	0	The cash drawer plugged in PIN3



Cashbox.PIN5

1

The cash drawer plugged in PIN5

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note: NULL

14) .Prn_LineFeed

```
int Prn_LineFeed(int nLine)
```

Description: Line feeding**Parameter:**

int nLine: The number of lines for line feeding.

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note: NULL

15) .Prn_MarkFeed

```
int Prn_MarkFeed()
```

Description: Mark feeding.**Parameter:** NULL**Return:**

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note:

1. Some printers don't support black mark function, then this function is not applicable.

2. Some printers may not be compatible with this command. Users need to invoke Prn_SendData to accomplish this function.

16) .Prn_SetCharacterSet

```
int Prn_SetCharacterSet(String strCode)
```

Description: Set the code page of the data, Default is gb2312.

Parameter:

String strCode: Code page, e.g: gb2312 is Simplified Chinese, big5 is Traditional Chinese.

Return:

ErrorCode.SUCCESS	Successful
-------------------	------------

Note: n/a.

17) .Prn_SetInterCharacterSet

```
int Prn_SetInterCharacterSet(int Value)
```

Description: Set the code page of the printer

Parameter:

Int Value: The Code Pages that can be used are as follows:

Value		Description
PC437	0	Code page PC437
KATAKANA	1	Katakana
PC850	2	Code page PC850
PC860	3	Code page PC860
PC863	4	Code page PC863
PC865	5	Code page PC865
WPC1253	6	Code page WPC1252
TRAN	7	Code page TRAN
WPC1256	8	Code page WPC1256
PC737	9	Code page PC737
WPC1250	10	Code page WPC1250

WPC1252	16	Code page WPC1252
PC866	17	Code page PC866
PC852	18	Code page PC852
PC858	19	Code page PC858
USER	255	User set pages

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note:

1. This function takes effect only after reboot for some printer models. The code pages take effect forever after configuration.
2. Some printers support code page configuration via DIP switches. This function is not applicable.

18) .Prn_SetLineSpacing

```
int Prn_SetLineSpacing(int nSpace)
```

Description: Sets the line spacing**Parameter:**

int nSpace: The handle return by Prn_Connect function

int nSpace: the value of line space. As the difference of printer mechanism and default parameters, the space unit and range might be different from printer models. Please refer to command explanation of 1B 33 for different printers. When nSpace=0, the configuration is reset.

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note:

19) .Prn_SetFontStyle

```
int Prn_SetFontStyle(int nStyle)
```

Description: Text properties.

Parameter:

int nFontStyle: Sets the text properties. The following values can be duplicated:

Value		Description
Font.DEFAULT	0	FontA
Font.FONTB	1	FontB
Font.BOLD	2	Adds Bold lettering attribute
Font.UNDERLINE	4	Adds Underlining attribute
Font.REVERSE	8	Adds Reverse lettering attribute

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note: If to invoke this function with Prn_SetFontSize, please use it before Prn_SetFontSize, otherwise the font size could not be changed.

20).Prn_SetFontSize

```
int Prn_SetFontSize(int nSize)
```

Description: Text size properties.

Parameter:

int nSize: The text size properties. The width and height values can be duplicated:

Value		Description
WidthSize.SIZE0	0	Normal
WidthSize.SIZE1	16	X 2
WidthSize.SIZE2	32	X 3
WidthSize.SIZE3	48	X 4
WidthSize.SIZE4	64	X 5
WidthSize.SIZE5	80	X 6
WidthSize.SIZE6	96	X 7

WidthSize.SIZE7	112	X 8
HeightSize.SIZE0	0	Normal
HeightSize.SIZE1	1	X 2
HeightSize.SIZE2	2	X 3
HeightSize.SIZE3	3	X 4
HeightSize.SIZE4	4	X 5
HeightSize.SIZE5	5	X 6
HeightSize.SIZE6	6	X 7
HeightSize.SIZE7	7	X 8

Note: If to invoke this function with `Prn_SetFontStyle`, please use it before `Prn_SetFontStyle`, otherwise the font will be reset to default size after `Prn_SetFontStyle` invoked.

Return:

<code>ErrorCode.SUCCESS</code>	Successful
<code>ErrorCode.PORTERROR</code>	Port error
<code>ErrorCode.WRITEERROR</code>	Send data to priner failed

Note: If to invoke this function with `Prn_SetFontStyle`, please use it before `Prn_SetFontStyle`, otherwise the font will be reset to default size after `Prn_SetFontStyle` invoked.

21) .Prn_SetAlignment

```
int Prn_SetAlignment(int nAlignment)
```

Description: Set text and image alignment.

Note: some printers may not (fully) support this function. This function generally is applicable for text printing. This function may be applicable for Bitmap image or barcode for Some printers

Parameter:

`int Alignment`: Text alignment:

Value		Description
<code>Alignment.LEFT</code>	0	Left alignment
<code>Alignment.CENTER</code>	1	Center alignment
<code>Alignment.RIGHT</code>	2	Right alignment

Return:

<code>ErrorCode.SUCCESS</code>	Successful
--------------------------------	------------

ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note:

22) .Prn_SendData

```
int Prn_SendData(byte[] byData, int nLen)
```

Description: Send data to printer**Parameter:**

Byte[] byData: Pointer to the character string to send, Can be a printer control command, can also be printed content.

Int nLen: The length of byData

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to priner failed

Note:

23) .Prn_ReadData

```
int Prn_ReadData(byte []by, int nLen, int nTimeout)
```

Description: Reveives the data returned by the printer.**Parameter:**

Byte []by: Pointer to a buffer that receives the data.

int nLen: The length of the data to be received, Be careful not to be greater than the length of the by.

int nTimeOut: time out, units is ms

Return:

>0	The length of the received data
ErrorCode.PORTERROR	Port error

Note:

1、 This command is to control the printer further. It is used to read data from printer. For example, the status returns might be different from printer models, if the command Prn_Status cannot return accurate status, users could send command 10 01 n to printer to fetch specific status.

24) .Prn_BeginTransaction

```
int Prn_BeginTransaction()
```

Description: Enable batch mode.

Parameter: NULL

Return:

ErrorCode.SUCCESS	Successful
-------------------	------------

Note:

1. In batch mode, all data (except data sent by Prn_Status, Prn_PowerStatus, Prn_GetMsrTrack) won't be sent out instantly but stored in buffer locally. The data will be sent out only after quit of batch mode by EndTransaction.

2. Please refer to part IV (Others • Batch mode).

25) .Prn_EndTransaction

```
int Prn_EndTransaction()
```

Description: Exit batch mode.

Parameter: NULL

Return:

ErrorCode.SUCCESS	Successful
ErrorCode.PORTERROR	Port error
ErrorCode.WRITEERROR	Send data to printer failed

Note: for the printers with Ethernet port support batch mode, it's better to disconnect communication after quit of batch mode. It is good

for the step host to print data immediately. Please refer to Part IV (Others • Batch mode).

26) .Prn_GetMsrTrack

```
byte[] Prn_GetMsrTrack(int nTrack, int nTimeout)
```

Description: Retrieve and check a printer's MSR data.

Parameter:

int nTrack: MSR Track:

Value		Description
MsrTrack.TRACK1	1	Manual-mode, Read track1 data
MsrTrack.TRACK2	2	Manual-mode, Read track2 data
MsrTrack.TRACK3	3	Manual-mode, Read track3 data
MsrTrack.TRACK12	4	Manual-mode, Read track1 & track2 data
MsrTrack.TRACK23	5	Manual-mode, Read track2 & track3 data
MsrTrack.TRACK123	6	Auto-mode, Read track1/track2/track3 data

Manual mode: to invoke this function to get track data after swiping card.

Automatic mode: to invoke this function to get all three tracks' data after swiping card.

To transit operation mode of card reader, please refer to corresponding printer manuals.

int nTimeout: Time out, units is ms.

Return: The length of the received data.

Note:

4.Label printer functions

1) .prn_PageSetup

```
void prn_PageSetup (int pageHeight, int pageWidth)
```

Description: to set the size of label page

Parameter:

int pageWidth: to define the width of label page

int pageHeight: to define the height of label page

Returns: NULL

Note: the maximum supported printing page is 576 x 2000dots. With 90 degrees rotated, it is 2000 x 576 dots.

2) .prn_PagePrint

```
int prn_PagePrint(int rotate)
```

Description: to print label pages

Parameter:

int rotate: to set the rotate angle, values as below:

Value	Description
0	No rotate
1	90 degrees of clockwise rotating

Returns:

PrinterConst.ErrorCode.SUCCESS	0	Succeed
PrinterConst.ErrorCode.WRITEERROR	-2	Failed to send data to printer. The possibility of disconnection to printer

Note: null

3) .prn_PageClear

```
void prn_PageClear()
```

Description: to clear page, all configurations to be reset

Parameter: NULL

Returns: NULL

Note: n/a

4) .prn_PrinterStop

```
int prn_PrinterStop()
```

Description: to stop printing and clear buffer

Parameter: NULL

Returns:

PrinterConst.ErrorCode.SUCCESS	0	Succeed
PrinterConst.ErrorCode.WRITEERROR	-2	Failed to send data to printer. The possibility of disconnection to printer

Note: n/a

5) .prn_DrawBox

```
void prn_DrawBox(int lineWidth, int x0, int y0, int x1, int y1)
```

Description: to draw rectangles (boxes)

Parameter:

int lineWidth: to define the width of the drawing lines (unit: dots)

int x0: to define the x-coordinate of the start point

int y0: to define the y-coordinate of the start point

int x1: to define the x-coordinate of the end point

int y1: to define the y-coordinate of the end point

Return: NULL

Note: n/a

6) .prn_DrawLine

```
void prn_DrawLine(int lineWidth, int x0, int y0, int x1, int y1)
```

Description: to draw lines

Parameter:

int lineWidth: to define the width of the drawing line (unit: dots)

int x0: to define the x-coordinate of the start point

int y0: to define the y-coordinate of the start point

int x1: to define the x-coordinate of the end point

int y1: to define the y-coordinate of the end point

Returns: NULL

Note: n/a

7) .prn_DrawInverseLine

```
void prn_DrawInverseLine(int x0, int y0, int x1, int y1)
```

Description: to draw blocks (generally for inverse printing)

Parameter:

int x0: to define the x-coordinate of the start point

int y0: to define the y-coordinate of the start point

int x1: to define the x-coordinate of the end point

int y1: to define the y-coordinate of the end point

Returns: NULL

Note: the inverse printing should be like this: **SAVE**. To print like this, the first step should be to invoke function *prn_DrawInverseLine* to print out a whole block, then invoke *prn_DrawText* to print content (the inverse function should not be enabled in this invoking).

8) .prn_DrawText

```
void prn_DrawText (int x, int y, String text, String fontName, int  
fontsize, int rotate, int bold, int underline, int reverse)
```

Description: to print text**Parameter:**

int x: to define the x-coordinate of the start point

int y: to define the y-coordinate of the start point

String text: the text content

String fontName: font names to refer below table.

int fontsize: font sizes to refer below table.

Character font sizes table reference:

fontName	fontsize	Width x Height	Description
0	0	8x16	fixed pitch dot font -- Dark
0	1	16x16	fixed pitch dot font -- Dark
0	2	8x32	fixed pitch dot font -- Dark
0	3	16x32	fixed pitch dot font -- Dark
0	4	24x32	fixed pitch dot font -- Dark
0	5	16x32	fixed pitch dot font -- Dark
0	6	32x48	fixed pitch dot font -- Dark
1	0	16x48	fixed pitch dot font -- <i>Italic</i>
2	0	16x16	fixed pitch dot font -- Light
4	0	24x48	fixed pitch dot font -- Light
4	1	24x72	fixed pitch dot font -- Light
4	2	48x48	fixed pitch dot font -- Light
4	3	24x72	fixed pitch dot font -- Light
4	4	24x72	fixed pitch dot font -- Light
4	5	24x72	fixed pitch dot font -- Light
4	6	24x72	fixed pitch dot font -- Light
4	7	24x72	fixed pitch dot font -- Light
5	0	12x24	fixed pitch dot font -- Dark
5	1	12x24	fixed pitch dot font -- Dark
5	2	24x48	fixed pitch dot font -- Dark
5	3	24x48	fixed pitch dot font -- Dark
7	0	12x48	fixed pitch dot font -- Light
7	1	12x48	fixed pitch dot font -- Light
8	0	12x24	fixed pitch dot font --Light
55	0	8x16	fixed pitch dot font --Light

Chinese character font sizes

fontName	fontsize	Width x Height	Description
55	The higher half of the byte represents for the times of width to	Chinese: 16x16 English: 8x8	fixed pitch dot font

	zoom in The lower half of the byte represents for the times of height to zoom in		
24	The higher half of the byte represents for the times of width to zoom in The higher half of the byte represents for the times of height to zoom in	Chinese: 24×24 English: 12×12	fixed pitch dot font
32	The higher half of the byte represents for the times of width to zoom in The higher half of the byte represents for the times of height to zoom in	Chinese: 32×32 English: 16×32	
128	0	Chinese:128×128 English:64×128	<i>Note:Some printer is not supported</i>

For example: fontsize=0x11, both width and height to be enlarged by 1 time. To double the width and height.

fontsize=0x10, to double the width. Height to be remained at original size.

fontsize=0x23, to enlarge the width to 3 times, to enlarge the height to 4 times.

int rotate: to set the rotating, values are as below:

Value	Description
0	No rotate
1	90 degrees at clockwise rotate
2	180 degrees at clockwise rotate
3	270 degrees at clockwise rotate

int bold: to set bold font, values from 0-5. 0 is for normal font.

int underline: to set underline, values are as below table,

Value	Description
0	No underline
1	Underlined by 1 dot width

int reverse: to set inverse printing, values are as below table,

Value	Description
0	No inverse (white background and black content)
1	Inverse printing (black background and white content)

Returns: NULL

Note: n/a

9) .prn_DrawTextAlign

```
void prn_DrawTextAlign (int x0, int y0, int x1, int y1, int
nAlign, String text, String fontName, int fontsize, int rotate,
int bold, int underline, int reverse)
```

Description: to print text contents in a designated block space with certain alignment

Parameter:

int x0: to define the x-coordinate of the start point

int y0: to define the y-coordinate of the start point

int x1: to define the x-coordinate of the end point

int y1: to define the y-coordinate of the end point

int nAlign: text alignment, values are as below:

Value	Description
0	Left alignment
1	Center alignment
2	Right alignment

String text: text content

String fontName: font name, refer to *prn_DrawText*

int fontsize: font size, refer to *prn_DrawText*

int rotate: to set rotate printing, refer to *prn_DrawText*

int bold: to set bold font, values are from 0-5, 0 means normal font.

int underline: to set underline, refer to *prn_DrawText*

int reverse: to set inverse printing, refer to *prn_DrawText*

Returns: NULL

Note: the text alignment is only for horizontal directions. The vertical alignment could be fixed with y-coordinate pre-calculation.

10) .prn_DrawBarcode

```
int prn_DrawBarcode(int x, int y, String text, int barcodetype,
int rotate, int linewidth, int height)
```



Description: to print barcode**Parameter:**

int x: to define the x-coordinate of the start point

int y: to define the y-coordinate of the start point

String text: the content of barcode

int barcodetype: to select barcode type, selections are as below table.

Value	Description
0	JAN13(EAN13)
1	JAN8(EAN8)
2	CODE39
3	CODE93
4	CODE128
5	CODABAR
6	ITF
7	UPC-A
8	UPC-E
9	EAN13+2
10	EAN13+5
11	EAN8+2
12	EAN8+5
13	UPCA+2
14	UPCA+5
15	UPCE+2
16	UPCE+5
17	Postnet
18	MSI
19	QR Barcode
128	CODE128
39	CODE39
93	CODE93

int rotate: to set rotate printing. Refer to below table:

Value	Description
0	No rotate
1	90 degrees at clockwise rotate
2	180 degrees at clockwise rotate
3	270 degrees at clockwise rotate

int linewidth: Narrow barcode scale factor for 1D barcode (barcodetype!=19); QR code type for 2D barcode (barcodetype=19), values are 1 or 2.

int height: to define the barcode height (unit: dots) for 1D barcode (barcodetype!=19); to define the level of Width/Height for 2D barcode (barcodetype=19), values are from 1 to 32.

Returns: NULL

Note: n/a

11) .prn_BarcodeText

```
void prn_BarcodeText (String fontName, int fontsize, int  
offset)
```

Description: to start barcode text printing

Parameter:

String fontName: font name, refer to *prn_DrawText*

int fontsize: font size, refer to *prn_DrawText*

int offset: to define the space (offset) between barcode and text

Return: NULL

Note: 1) The function should be invoked before *prn_Barcode*

2) The barcode text is aligned centrally comparing to barcode by default

3) With invoking of this function, the text printing will be validated for all the current printing. To not print barcode text for a next barcode, *prn_AddLabelCommand("BARCODE-TEXT OFF\r\n")* should be invoked before next printing.

12) .prn_PrintBitmap

```
void prn_PrintBitmap(int startx, int starty, Bitmap bmp)
```

Description: to print Bitmap image

Parameter:

int startx: to define the x-coordinate of the start point

int starty: to define the y-coordinate of the start point

Bitmap bmp: image content of Bitmap

Returns: NULL

Note: n/a

13) .prn_PrinterStatus

```
int prn_PrinterStatus()
```

Description: to fetch the status of printer

Parameter: NULL

Returns: -1: no response from printer

0: correct status from printer

>0: incorrect status from printer (Bit 0 represents for cover open; Bit 1 represents for no paper; Bit2 represents for temperature anomaly; Bit 3 represents for printer busy.Bit 4 represents for low power; Bit 5 represents for print now)

Note: n/a

14) .prn_AddLabelCommand

```
int prn_AddLabelCommand(String strCmd)
```

Description: to send label printing commands

Parameter:

String strCmd: Label printing commands, to refer to corresponding command list manual.

Returns: NULL

Note: to accomplish special printing effects

15) .prn_DrawWatermark

```
void prn_DrawWatermark (int x, int y, String text, String  
fontName, int fontsize, int rotate, int pattern, int gray)
```

Description: to print watermark

Parameter:

int x: to define the x-coordinate of the start point

int y: to define the y-coordinate of the start point

String text: the text content

String fontName: font names, reference prn_Drawtext.



int fontsize: font sizes, reference prn_Drawtext.

int rotate: to set the rotating,reference prn_Drawtext:

int pattern: pattern, Refer to below table :

Value	Description
100	solid black/default pattern
101	Horizontal lines
102	Vertical lines
103	Right rising diagonal lines
104	Left rising diagonal lines
105	Square pattern
106	Cross hatch pattern
107	Dot

int gray: Gray level,range:0-4,0-black,4-gray.

Note: Some printer is not supported.

5.Barcode printer functions

1).bar_SetSize

```
int bar_SetSize (double dWidth, double dHeight, int nUnit)
```

Description: Defines the label width and length

Parameter :

double dWidth: Label width

double dHeight: Label length

int nUnit: Unit, Refer to below table :

Value	Description
Unit.UNIT_INCH	English system (inch)
Unit.UNIT_MM	Metric system (mm)
Unit.UNIT_DOT	Dot measurement

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

2) .bar_SetGap

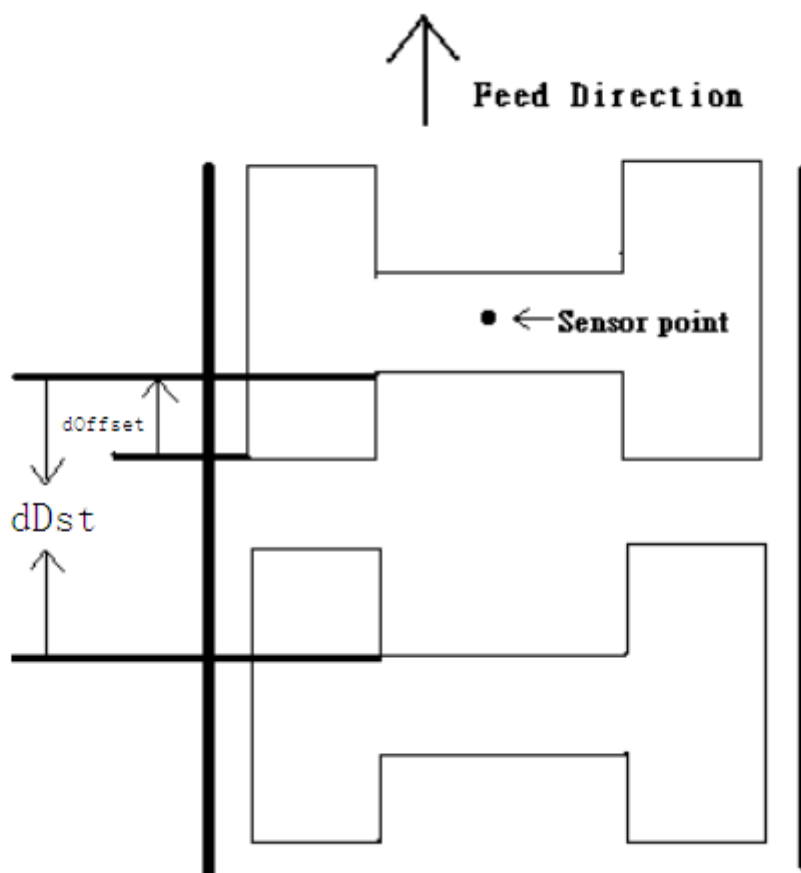
```
int bar_SetGap (double dDst, double dOffset, int nUnit)
```

Description: Set the distance between two labels

Parameter:

double dDst: The gap distance between two labels.

double dOffset: The offset distance of the gap. Special gap as below:



int nUnit: Unit, Refer to below table:

Value	Description
-------	-------------

Unit.UNIT_INCH	0	English system (inch)
Unit.UNIT_MM	1	Metric system (mm)
Unit.UNIT_DOT	2	Dot measurement

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

3) .bar_SetBline

```
int bar_SetBline (double dHeight, double dExtra, int nUnit)
```

Description: Sets the height of the black line and the user-defined extra label feeding length each form feed takes.

Parameter :

double dHeight: The height of black line either in inch or mm

double dExtra: The extra label feeding length. Continuous label is 0.

int nUnit: int nUnit: Unit, Refer to below table:

Value		Description
Unit.UNIT_INCH	0	English system (inch)
Unit.UNIT_MM	1	Metric system (mm)

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

4) .bar_SetOffset

```
int bar_SetOffset (double dOffset, int nUnit)
```


Description : Defines the selective, extra label feeding length each form feed takes, which especially in peel-off mode and cutter mode, is used to adjust label stop position, so as for label to register at proper places for the intended purposes. The printer back tracks the extra feeding length before the next run of printing.

Parameter :

double dOffset: The offset distance

int nUnit: int nUnit: Unit, Refer to below table:

Value		Description
Unit.UNIT_INCH	0	English system (inch)
Unit.UNIT_MM	1	Metric system (mm)

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

5) .bar_SetShift

```
int bar_SetShift (int nShift)
```

Description: Moves the label's vertical position. A positive value moves the label further from the printing direction; a negative value moves the label towards the printing direction.

Parameter :

int nShift: the unit is dot.

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

6) .bar_SetSpeed

```
int bar_SetSpeed (int nSpeed)
```

Description: Defines the print speed

Parameter:

int nSpeed: Printing speed in inch per second, range is $1 \leq nSpeed \leq 6$

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

7) .bar_SetDensity

```
int bar_SetDensity (int nDensity)
```

Description: Sets the printing darkness.

Parameter:

int nDensity: $0 \leq nDensity \leq 15$, 0 specifies the lightest level, 15 specifies the darkest level

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

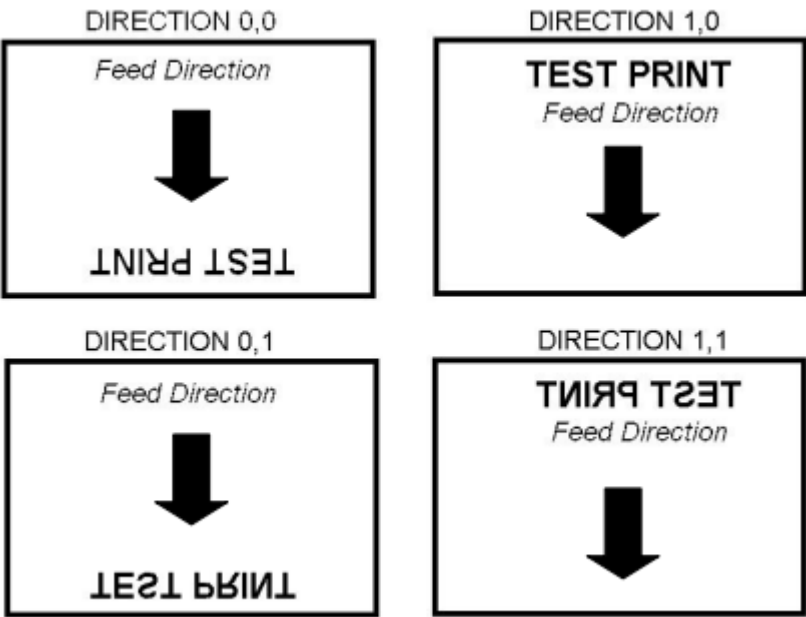
8) .bar_SetDirection

```
int bar_SetDirection (int nDirection, int nMirror)
```

Description: Defines the printout direction and mirror image.

Parameter :

int nDirection: 0 or 1. Please refer to the illustrations below:
int nDirection: 0:Print normal image. 1:Print mirror image.



Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

9) .bar_SetPeel

```
int bar_SetPeel (boolean bFlag)
```

Description: Enable/disable the self-peeling function. When this function is set on, the printer stops after each label printing, and does not print the next label until the peeled label is taken away.

Parameter :

boolean bFlag: true:Enable the self-peeling function. False: Disable the self-peeling function.

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

10) .bar_SetCharacterSet

```
int bar_SetCharacterSet(String strCode)
```

Description: Set the code page of the data, Default is gb2312.

Parameter :

String strCode: Code page, e.g: gb2312 is Simplified Chinese, big5 is Traditional Chinese.

Returns :

ErrorCode.SUCCESS	SUCCESS
-------------------	---------

Note: n/a

11) .bar_CLS

```
void bar_CLS ()
```

Description: Clears the image buffer

Parameter: NULL

Returns: NULL

Note: This function must be placed after bar_SetSize function.

12) .bar_Print

```
int bar_Print (int nSets, int nCopies)
```

Description: Prints the label format currently stored in the image buffer.

Parameter:

int nSets: Specifies how many sets of labels will be printed.

1<=nSets<=999999999

int nCopies: Specifies how many copies should be printed for each particular label set. 1<=nCopies<=999999999

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

12) .bar_DrawBar

```
int bar_DrawBar (int nX, int nY, int nWidth, int nHeight)
```

Description: Draw a bar on the label format

Parameter:

int nX: The upper left corner x-coordinate (in dots)

int nY: The upper left corner y-coordinate (in dots)

int nWidth: Bar width (in dots)

int nHeight: Bar height (in dots)

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

13) .bar_Barcode

```
int bar_Barcode (int nX, int nY, String strCodeType, int
nHeight, int nReadable, int nRotation, int nNarrow, int nWide,
String strCode)
```

Description: print 1D barcodes.

Parameter:

int nX: Specify the x-coordinate of the bar code on the label

int nY: Specify the y-coordinate of the bar code on the label

String strCodeType: Code type, only support "128".

int nHeight: Bar code height (in dots)

int nReadable: 0 not readable, 1 human readable

int nRotation: Rotate, values as below:

Value	Description
0	No rotation
90	Rotate 90 degrees clockwise
180	Rotate 180 degrees clockwise
270	Rotate 270 degrees clockwise

int nNarrow: Width of narrow element (in dots)

int nWide: Width of wide element (in dots)

String strCode: bar code content

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

14) .bar_QRCode

```
int bar_QRCode (int nX, int nY, int nWidth, int nHeight, String
strCode)
```

```
int bar_QRCode (int nX, int nY, int nWidth, int nHeight, String
strCode, Bitmap markBMP)
```



Description: Print 2D barcodes.

Parameter:

int nX: Specify the x-coordinate of the bar code on the label

int nY: Specify the y-coordinate of the bar code on the label

int nWidth: Bar code width (in dots)

int nHeight: Bar code height (in dots)

String strCode: bar code content

Bitmap markBMP: This parameter is optional, for the 2D barcode Center print watermark Logo.

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

15).bar_Bitmap

```
int bar_Bitmap (int nX, int nY, int nMode, String strFileName)
```

```
int bar_Bitmap (int nX, int nY, int nMode, Bitmap image)
```

Description: draw bitmap images

Parameter:

int nX: Specify the x-coordinate

int nY: Specify the y-coordinate

int nMode: Graphic modes listed below:

PrinterConst.ImgMode.MODE_OVERWRITE	0	OverWrite
PrinterConst.ImgMode.MODE_OR	1	Or
PrinterConst.ImgMode.MODE_XOR	2	Xor

String strFileName:Windows BMP monochrome bitmap file full path.

OR

Bitmap image:Android bitmap data.

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:



16) .bar_DrawBox

```
int bar_DrawBox (int nXStart, int nYStart, int nXEnd, int
nYEnd, int nThickness)
```

Description: draw rectangles on the label

Parameter:

int nXStart: Specify x-coordinate of upper left corner (in dots)

int nYStart: Specify y-coordinate of upper left corner (in dots)

int nXEnd: Specify x-coordinate of lower right corner (in dots)

int nYEnd: Specify y-coordinate of lower right corner (in dots)

int nThickness: Line thickness (in dots)

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

17) .bar_PutPCX

```
int bar_PutPCX (int nX, int nY, String strFileName)
```

Description: Prints PCX format images.

Parameter:

int nX: The x-coordinate of the PCX image.

int nY: The y-coordinate of the PCX image

String strFileName: PCX filename, Case sensitive.

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

18) .bar_SetReverse


```
int bar_SetReverse(int nX, int nY, int nWidth, int nHeight)
```

Description: Reverse a region in image buffer.

Parameter:

int nX: The x-coordinate of the starting point (in dots)

int nY: The y-coordinate of the starting point (in dots)

int nWidth: X-axis region width (in dots)

int nHeight: Y-axis region height (in dots)

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

19).bar_Text

```
int bar_Text(int nX, int nY, String strFontName, int
nRotation, int nXMulti, int nYMulti, String strContent)
```

Description: Print text on label

Parameter:

int nX: The x-coordinate of the text.

int nY: The y-coordinate of the text.

String strFontName: Font name.

Font name	Width X Height
1	8 x 12
2	12 x 20
3	16 x 24
4	24 x 32
5	32 x 48
7	14 x 25
8	21 x 27
中文	24 x 24 (English 12 x 24)

int nRotation: Rotate, value as below:

Value	Description
-------	-------------

0	No rotation
90	90 degrees, in clockwise direction
180	180 degrees, in clockwise direction
270	270 degrees, in clockwise direction

`int nXMulti`: Horizontal multiplication, up to 10x. Available factors: 1 ~ 10

`int nYMulti`: Vertical multiplication, up to 10x. Available factors: 1 ~ 10

`String strContent`: Text content, If there is any double quote(") within the text, please change it to \["].

Returns:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	SUCCESS
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	Invalid parameter

Note:

20) `.bar_DownloadBAS`

```
int bar_DownloadBAS(String strFileName, String strContent)
```

Description: Download program files.

Parameter:

`String strFileName`: The filename resident in printer memory. Filenames are case sensitive. File extensions must be ".BAS". Filenames must be in 8.3 format.

`String strContent`: Program file content, Does not need to include DOWNLOAD and EOP, the function will automatically add.

Returns:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	SUCCESS
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	Invalid parameter

Note:

21) `.bar_DownloadFile`

```
int bar_DownloadFile(String strFileName, String  
strLocalFileName)
```

Description: Download data files.

Parameter:

String strFileName: The name of data file that will remain resident in the printer memory (case sensitive).

String strLocalFileName: Full path of the file.

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

22) .bar_KillFile

```
int bar_KillFile(String strFileName)
```

Description: Delete a file in the printer memory.

Parameter:

String strFileName: The name of program files or data file

Returns:

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note:

23) .bar_RunFile

```
int bar_RunFile(String strFileName)
```

Description: Execute a program resident in the printer memory.

Parameter:

String strFileName: program file name.



Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

24) .bar_SetCounter

```
int bar_SetCounter(int n, int nStep, String strExp)
```

Description : Set the counter number in the program and its increments. There are three different types of counters: digit (0~9~0), lower case letter (a~z~a) or upper case letter (A~Z~A).

Parameter :

int n: Counter number. There are 51 counters available 0~50 in the printer.

int nStep: The increment of the counter, can be positive or negative.

Range is -999999999 <= nStep <= 999999999.

String strExp: Initial strng.

Returns :

PrinterConst.ErrorCode.SUCCESS	0	SUCCESS
PrinterConst.ErrorCode.INVALIDPARAM	-2	Invalid parameter

Note :

25) .bar_PrintCounter

```
int bar_PrintCounter(int nX, int nY, String strFontName, int
nRotation, int nXMulti, int nYMulti, int nCounter)
```

Description : Print the Counter.

Parameter :

int nX: The x-coordinate of the text.

`int nY`: The y-coordinate of the text.
`String strFontName`: Font name, please see `bar_Text` function.
`int nRotation`: Rotate, please see `bar_Text` function.
`int nXMulti`: Horizontal multiplication, up to 10x, Available factors: 1 ~ 10
`int nYMulti`: Vertical multiplication, up to 10x, Available factors: 1 ~ 10
`int nCounter`: Counter number. There are 51 counters available 0~50 in the printer.

Returns:

<code>PrinterConst.ErrorCode.SUCCESS</code>	0	SUCCESS
<code>PrinterConst.ErrorCode.INVALIDPARAM</code>	-2	Invalid parameter

Note:

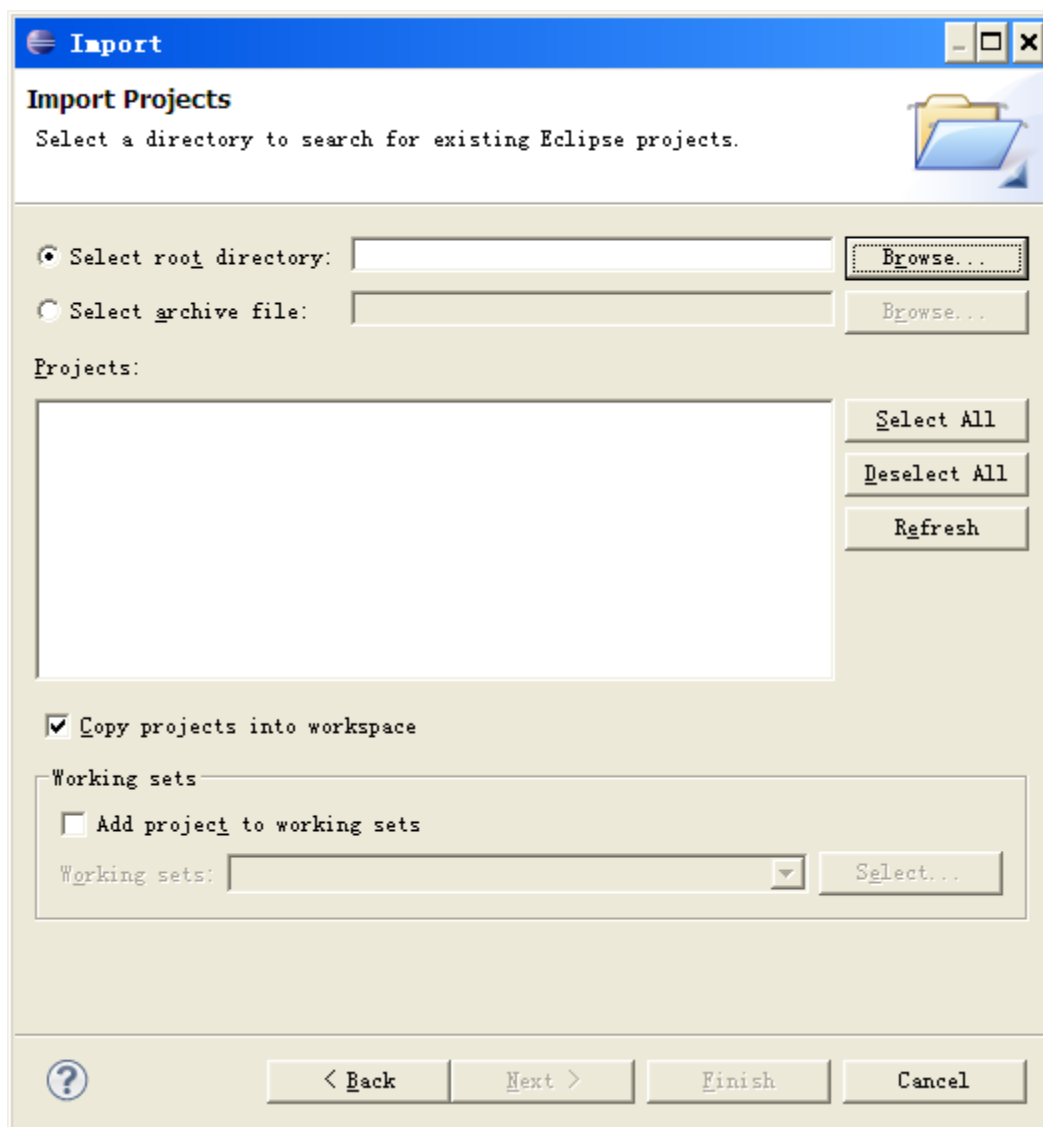
III:Samples

ZQPrinterDemo: Thermal printer

LabelDemo: Label printer

ZQPrinterSDKDemo and ZQLabelSDKDemo is for Eclipse environment.

In Eclipse, click File->Import->General->Existing Projects into Workspace



Click Browse and select ZQLabelSample folder, then click Finish button.

IV:Others

1. Access port permission problem.

Various types of ports are required to have corresponding access rights, such as:

WIFI: Need to be added in AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Bluetooth: Need to be added in AndroidManifest.xml:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Serial port: need to ensure that there can be read and write permissions.

USB:

Please check as below:

In the /system/etc/permissionsdirectory, android.hardware.usb.host.xml and handheld_core_hardware.xml (or tablet_core_hardware.xml), <permissions> which should contain <feature name="android.hardware.usb.host"/>.

2. Batch mode

Printers are mono-channel devices. That means the printer can only communicate with one host machine at the same time. If there are multi-hosts sending data to the printer, the printer may get confused. In some cases there are several hosts communicating with one printer (kitchen printer in restaurants), one of the host machines need to be configured as server host. All data sent to printer need to be concentrated and sent to the printer via the server host. ZONERICH AB-88H Ethernet/WIFI printer supports batch mode, it can communicate with several hosts at the same time. Please pay attention to below tips:

1. The printer supports maximum 5 hosts. If there are more than 5 hosts, server host is needed.
2. The judge rules of printing is as below:

- a. The printer will print data from the most first connected host machine;
- b. Only after the disconnection from the previous host machine, the next printing job from another host machine could be performed.

Then, to accomplish Ethernet printing jobs, functions related to batch mode should be invoked. At the same situation, after all printing data sent (Prn_EndTransaction invoked), the communication should be disconnected in time (Note: if there is no active disconnection from program, there is still a timeout to disconnection the communication), for the better of next printing job from other hosts.