

POS_SDK Development Manual

(V 1.1)

Content

Abstract.....	6
1.1 Nouns Explanation.....	6
Chapter 2. Function illustration.....	7
2.1 POS Constant definition list.....	7
2.2 TSPL Constant definition list.....	8
2.3 POS Function list.....	9
2.4 TSPL Function list.....	10
2.5 Function invoking example.....	11
2.6 POS Function illustrationLONG.....	14
POS_Port_OpenA/W(LPSTR szName, INT iPort, BOOL bFile, LPSTR szFilePath).....	14
LONG POS_Port_Close(LONG iPrinterID).....	15
LONG POS_Output_PrintFontStringA/W(LONG iPrinterID, INT iFont, INT iThick, INT iWidth, INT iHeight, INT iUnderLine, LPCSTR lpString).....	15
LONG POS_Control_SetRotaryPrint(LONG iPrinterID,INT n).....	16
LONG POS_Control_SetInvertedPrint(LONG iPrinterID,INT n).....	17
LONG POS_Control_OppositeColor(LONG iPrinterID, BOOL bOpposite).....	18
LONG POS_Control_AlignType(LONG iPrinterID, LONG iAlignType).....	19
LONG POS_Output_PrintOneDimensionalBarcodeA/W(LONG iPrinterID, INT iType, INT iWidth, INT iHeight, INT hri, LPCSTR lpString).....	19
LONG POS_Output_PrintTwoDimensionalBarcodeA/W(LONG iPrinterID, INT iType, INT parameter1, INT parameter2, INT parameter3, LPCSTR lpString).....	20
LONG POS_Output_SendLocalFileA/W(LONG iPrinterID, LPCSTR lpFilePath).....	21
LONG POS_Output_DownloadRamBmpA/W(LONG iPrinterID, LPCSTR lpFilePath).....	22
LONG POS_Output_PrintRamBmp(LONG iPrinterID,INT n).....	23
LONG POS_Output_PrintBmpDirectA/W(LONG iPrinterID, LPCSTR lpFilePath).....	23
LONG POS_Output_PrintStringA/W(LONG iPrinterID, LPSTR lpString).....	24
LONG POS_Output_PrintFlashBmp(LONG iPrinterID,INT n).....	24
LONG POS_Control_PrintTestpage(LONG iPrinterID).....	25

LONG POS_Output_PrintData(LONG iPrinterID, LPCSTR lpBuffer, INT iLength).....	26
LONG POS_Control_SetPrintPosition(LONG iPrinterID, LONG iLeftMargin, LONG iWidth).....	27
LONG POS_Control_SetLineSpace(LONG iPrinterID, INT iSpace).....	28
LONG POS_Control_SetPrintFontE(LONG iPrinterID, BOOL iFont, BOOL iBold, BOOL iDoubleWidth, BOOL iDoubleHeight, BOOL iUnderLine).....	28
LONG POS_Control_SetPrintFontC(LONG iPrinterID, BOOL iDoubleWidth, BOOL iDoubleHeight, BOOL iUnderLine).....	29
LONG POS_Control_CutPaper(LONG iPrinterID, LONG iType, LONG iLines).....	29
LONG POS_Control_FeedLines(LONG iPrinterID, LONG iLines).....	30
LONG POS_Control_CashDraw(LONG iPrinterID, LONG iPort, LONG iTime1, LONG iTime2).....	30
LONG POS_Control_BlackMark(LONG iPrinterID).....	31
LONG POS_Status_RTQueryStatus(LONG iPrinterID).....	31
LONG POS_Status_RTQueryTypeStatus(LONG iPrinterID, INT n).....	32
LONG POS_Status_QueryTaskStatus(LONG iPrinterID, INT second).....	34
LONG POS_Input_PrinterId(LONG iPrinterID, char *buf).....	35
LONG POS_Control_ReSet(LONG iPrinterID).....	36
LONG Color24_GrayBW(LPCTSTR szSourceFile, LPCTSTR szTargetFile).....	36
2.7 TSPL Function illustration	36
LONG SetIs21();.....	36
LONG PageSetupTSPL(LONG iPrintID, INT PageWidth, INT PageHeight);.....	37
LONG DrawLineTSPL(LONG iPrintID, INT StartX, INT StartY, INT LineWidth, INT LineHeight);.....	38
LONG PrintTSPL21(LONG iPrintID, INT Set);.....	39
LONG PrintTSPL51(LONG iPrintID, INT Set, INT Copy);.....	40
LONG DrawBorderTSPL(LONG iPrintID, INT LineWidth, INT top_left_x, INT top_left_y, INT bottom_right_x, INT bottom_right_y);.....	40
LONG DrawTextTSPL(LONG iPrintID, INT start_x, INT start_y, BOOL isSimplifiedChinese, INT xMultiplication, INT yMultiplication, INT rotate, CString content);.....	42
LONG DrawBarCodeTSPL(LONG iPrintID, INT start_x, INT start_y, CString type, INT height, BOOL isReadable, INT rotate, INT narrowWidth, INT wideWidth, CString content);.....	44
LONG ClearBuffTSPL(LONG iPrintID);.....	45
LONG GetPrinterStatusTSPL(LONG iPrintID).....	46
LONG DriveBeepTSPL(LONG iPrintID).....	47

LONG	SetPaperbackOrPaperFeedTSPL(LONG iPrintID,BOOL isFeedBack, INT mDot).....	47
LONG	ReverseAreaTSPL(LONG iPrintID,INT start_x, INT start_y, INT width, INT height).....	48
LONG	SetGAPTSP(LONG iPrintID,DOUBLE value).....	50
LONG	SetLabelReferenceTSPL(LONG iPrintID,INT x,INT y).....	51
LONG	DownLoadBitMapTSPL(LONG iPrintID,BOOL isMoveFlash,CString PathName).....	51
LONG	Draw2DBarCodeTSPL(LONG iPrintID,INT start_x, INT start_y, CString Max ,CString content).....	53
LONG	PutBitMapTSPL(LONG iPrintID,INT start_x, INT start_y,CString fileName).....	55
LONG	SetCharsetTSPL(LONG iPrintID,CString CharSetName).....	57
LONG	SelectCodePageTSPL(LONG iPrintID,INT value).....	58
Appendix		60
1. CODE128summarize		60
2. Character set		60

Version illustration

Version	SDK version	Amendment record	Write	Time
V1.0	V1007 and higher	This document is the assort document of V1007 version and higher, which don't compatible with previous versions		2016-6-30
V1.1	V1008 and higher	This document is based on development manual V1.0, adding TSPL command description.		2017-1-20

Abstract

POS_SDK is printer quadratic element development package provided for our POS printer and label printer model, which bases on Windows platform, including printer interface dynamic link library POS_SDK.dll、Demo routine source code、POS_SDK development manual. POS_SDK is interface calling function and printing solution provided for development of third-party software.

Printer interface dynamic link library POS_SDK.dll mainly package the on/off operate performance function of serial interface, parallel interface, USB interface and Ethernet interface. It also packages the ESC/POS instruction set performance function and performance function of sending and receiving data, uploading local file and printing .

Demo routine source code provide routine source code of POS_SDK.dll for users to refer in secondary development.

POS_SDK development manual is used in POS_SDK application illustration. Please read user's manual carefully for your convenience use.

Chapter 1. Printer Relevant Knowledge

1.1 Nouns Explanation

- **Printing Width:**The maximum horizontal print range that printer supports, decided by printer itself. For example, as for 80mm printer, the printing width is 72mm(576 dots), while as for 58mm printer, the printing width is 48(384 dots).
- **Printing Area:** Printing area can be set by commands, and it must be less than or equal to printing width.
- **Line Height:** The height of characters line. Line Height = Characters Height + Lines Space.
- **Black Mark Paper:** Black mark is black patch pre-printed on the paper, users can use it to locate printing position.
- **Dpi(dots per inch):** Printing dots on each inch paper
- **Vertical or Horizontal Moving Units:** A moving unit is a printing dot by default, the horizontal distance is 1/8mm and the vertical distance is 1/8mm.

1.2 Relevant Knowledge

- **Western Language Printing:** Western characters we called including ASCII characters and

Codepages by default. The ASCII characters range is 0x20~0x7F, the Codepage range is 0x80~0xFF. Western language (such as German and Spanish) have their own single byte codepage. Because Codepage code and Chinese code have overlapping part, please print Codepage content in Western language printing mode. Common Western language characters fonts are: Font A: 12x24 Font B: 9x17.

- **Chinese Printing:** The Chinese characters including simplified Chinese and traditional Chinese by default. The common simplified Chinese character set we use is GB2312 and GB18030, while the common traditional Chinese character set we use is BIG5. Common Chinese character font: 24x24 (dots).
- **Twice Height Printing:** In this printing mode, the height of character is twice of normal height of character.
- **Twice Width Printing:** In this printing mode, the width of character is twice of normal width of character.
- **TSPL command:** Feature of the models support TSPL command is that buffer similar with canvas will be built in the printer, the printed content could specify coordinate on the canvas, only call printer method will make the printer print content, otherwise does not print.

Chapter 2. Function illustration

2.1 POS Constant definition list

Classification	Statement	Value	Illustration
Function returned status	<code>#define POS_ES_PAPERENDING</code>	6L	Paper will run out
	<code>#define POS_ES_DRAWERHIGH</code>	5L	High level signal of cash draw
	<code>#define POS_ES_CUT</code>	4L	Cutter non-recover
	<code>#define POS_ES_DOOROPEN</code>	3L	Paper case open
	<code>#define POS_ES_HEAT</code>	2L	Print head overheat
	<code>#define POS_ES_PAPEROUT</code>	1L	Paper out
	<code>#define POS_ES_SUCCESS</code>	0L	successfully/send successfully/status normally/printing finished
	<code>#define POS_ES_INVALIDPARA</code>	-1L	Parameter error
	<code>#define POS_ES_WRITEFAIL</code>	-2L	Failed to write
	<code>#define POS_ES_READFAIL</code>	-3L	Failed to read

	<code>#define</code> POS_ES_NONMONOCHROME_BITMAP	-4L	not monochrome bitmap
	<code>#define</code> POS_ES_OVERTIME	-5L	overtime/write overtime/read overtime/printing unfinished
	<code>#define</code> POS_ES_FILEOPENERROR	-6L	Failed to open pic/file
	<code>#define</code> POS_ES_OTHERERRORS	-100L	Error caused by other reason
1D barcode	<code>#define</code> POS_BT_UPCA	4001L	UPC-A
	<code>#define</code> POS_BT_UPCE	4002L	UPC-E
	<code>#define</code> POS_BT_JAN13	4003L	JAN13 (EAN13)
	<code>#define</code> POS_BT_JAN8	4004L	JAN 8 (EAN8)
	<code>#define</code> POS_BT_CODE39	4005L	CODE39
	<code>#define</code> POS_BT_ITF	4006L	ITF
	<code>#define</code> POS_BT_CODABAR	4007L	CODABAR
	<code>#define</code> POS_BT_CODE93	4073L	CODE93
	<code>#define</code> POS_BT_CODE128	4074L	CODE128
2D barcode	<code>#define</code> POS_BT_PDF417	4100L	PDF417
	<code>#define</code> POS_BT_DATAMATRIX	4101L	Data Matrix
	<code>#define</code> POS_BT_QRCODE	4102L	QR Code
Barcode character position	<code>#define</code> POS_HT_NONE	4011L	Not print
	<code>#define</code> POS_HT_UP	4012L	Print above the barcode
	<code>#define</code> POS_HT_DOWN	4013L	Print under the barcode
	<code>#define</code> POS_HT_BOTH	4014L	Print both above and under the barcode

2.2 TSPL Constant definition list

Function returned status	<code>#define</code> TSPL_SUCCESS	0L	Normal
	<code>#define</code> TSPL_IDERROR	-1L	Printer handle error
	<code>#define</code> TSPL_PARAM_LESS_EQUAL_ZERO	-2L	Parameter under or equal to zero
	<code>#define</code> TSPL_PARAM_GREAT_RANGE	-3L	Parameter above specified range
	<code>#define</code> TSPL_PRINTER_STATUS_OUTPAPER	1L	Printer out of paper
	<code>#define</code>	2L	Printing

	TSPL_PRINTER_STATUS_WORK		
	#define TSPL_PRINTER_STATUS_ENCLOSURENOCLOSE	3L	Printer case open
	#define TSPL_PRINTER_STATUS_ERROR	4L	Printer internal error

2.3 POS Function list

Function classification	Functional function	Illustration
Port operation	1. POS Port OpenA/W	Open port
	2. POS Port Close	Close port
Print	1. POS Output PrintFontStringA/W	print formatting character
	2. POS Control SetRotaryPrint	Rotating printing
	3. POS Control SetInvertedPrint	Invert printing
	4. POS Control OppositeColor	Reverse printing
	5. POS Control AlignType	Align type
	6. POS Output PrintOneDimensionalBarcodeA/W	1D barcode printing
	7. POS Output PrintTwoDimensionalBarcodeA/W	2D barcode printing
	8. POS Output SendLocalFileA/W	Print local file
	9. POS Output DownloadRamBmpA/W	Bitmap download
	10. POS Output PrintRamBmp	Print downloaded bitmap
	11. POS Output PrintBmpDirectA/W	Drive mode printing
	12. POS Output PrintStringA/W	Print character string
	13. POS Output PrintFlashBmp	Print pre-downloaded bitmap
	14. POS Control PrintTestpage	Print test page
	15. POS Output PrintData	Print buffer data
	16. POS Control SetPrintPosition	Set print position
	17. POS Control SetLineSpace	Set line space
	18. POS Control SetPrintFontE	Western font print format
	19. POS Control SetPrintFontC	Chinese font print format
Machine control	1. POS Control CutPaper	Control printer cut paper
	2. POS Control FeedLines	Paper feeding
	3. POS Control CashDraw	Control printer cash

		draw
	4. POS_Control_BlackMark	Black mark position
Status query	1. POS_Status_RTQueryStatus	paper status checking
	2. POS_Status_RTQueryTypeStatus	printer status checking
	3. POS_Status_QueryTaskStatus	Printing finished demonstration
	4. POS_Input_PrinterId	Get printer ID
others	1. POS_Control_ReSet	Initialize printer
	2. Color24_GrayBW	24 bitmap convert to monochrome bitmap
illustration	The functions which have “A/W” in the rear of function name support Unicode character encoded mode. If using ANSI encode mode standard, please use function with “A”; if using Unicode encode mode, please use function with “W”	

2.4 TSPL Function list

Function classification	Functional function	Illustration
Print	1. LONG PageSetupTSPL	Set page width and page height
	2. LONG DrawLineTSPL	Draw line
	3. LONG PrintTSPL21	TL21 print buffer
	4. LONG PrintTSPL51	TL51 print buffer
	5. LONG DrawBorderTSPL	Draw rectangle
	6. LONG DrawTextTSPL	Print words
	7. LONG DrawBarCodeTSPL	1D barcode print
	8. LONG Draw2DBarCodeTSPL	2D barcode print(TL21 not supported now)
	9. LONG ClearBuffTSPL	Clear printer buffer
	10. LONG DownLoadBitMapTSPL	Bitmap download(TL21 not supported now)
	11. LONG PutBitMapTSPL	Put downloaded bitmap into printer buffer (TL21 not supported now)
	12. LONG ReverseAreaTSPL	Reverse printing on specified area
	13. LONG SetGAPTSPL	Set the vertical spacing between labels
	14. LONG SetLabelReferenceTSPL	Defines the origin position of the label

	15. LONG SetCharsetNameTSPL	Set the international character set (TL21 not supported now)
	16. LONG SelectCodePageTSPL	Choose characters codepage (TL51 not supported now)
Machine control	1. LONG DriveBeepTSPL	Drive buzzer beep for one sound
	2. LONG SetPaperbackOrPaperFeedTSPL	Feed and back paper
Status query	1. LONG GetPrinterStatusTSPL	Status Check
illustration	1. LONG SetIs21	Used to confirm it is TL21 model or not, for convenience to distinguish command and related condition

2.5 Function invoking example

➤ POS Function invoking example

```

//take USB interface printer as example
//invoking procedure:open port→send printing content→close port:
...
LONG  m_hPrinter;//efine port return handle
LONG  ret;//efine port print function return value variable
...
//invoke POS_Port_OpenA to open print port

m_hPrinter = POS_Port_OpenA("SP-USB1", 1002, false, null);
//identify opening port successfully or not
if(m_hPrinter < 0)
{
    POS_Port_Close(m_hPrinter);
    MessageBox(_T("open failed"));
    return;
}
else
{
    MessageBox(_T("open successfully"));
    return;
}
...
//send printing content
ret = POS_Output_PrintFontStringA(m_hPrinter, 0, 1, 1, 1, 0, "Hi, thank you for choosing our printer, We
will get your the best experience!\r\n");

```

```
//identify sending printing content successfully or not
```

```

switch (ret)
{
case POS_ES_SUCCESS:
    MessageBox(_T("sending successfully"));
    break;
case POS_ES_INVALIDPARA:
    MessageBox(_T("Parameter error"));
    break;
case POS_ES_WRITEFAIL:
    MessageBox(_T("sending failed"));
    break;
case POS_ES_OVERTIME:
    MessageBox(_T("overtime"));
    break;
case POS_ES_OTHERERRORS:
    MessageBox(_T("other error"));
    break;
}
...
POS_Port_Close(m_hPrinter); //close printing port

```

➤ TSPL function invoking example

Description: When use TSPL model printing, need to call `ageSetupTSPL` method to build a printing buffer in the printer, then call methods like `DrawTextTSPL` after this will write the data into printer buffer, but will not print immediately, only print after received the print command (TL21 model call `PrintTSPL21` method, TL51 model call `PrintTSPL51` method), or will not print.

```

//take USB interface printer as example
//invoking procedure:open port→send printing content→close port;
...
LONG m_hPrinter;//define port return handle
LONG ret;//define port print function return value variable
...
//invoke POS_Port_OpenA to open print port
m_hPrinter = POS_Port_OpenA("SP-USB1", 1002, false, null);
//identify opening port successfully or not
if(m_hPrinter < 0)
{
    POS_Port_Close(m_hPrinter);
    MessageBox(_T("open failed"));
    return;
}

```

```

    }
    else
    {
        MessageBox(_T("open successfully"));
        return;
    }
    ...
    CString info;
    SetIs21();//If TL21 model, need to invoke this method,to confirm the model is TL21, for convenience
    to distinguish command and related condition.
    ret = ClearBuffTSPL(m_hPrinter);//Clear the buffer in printer
    if(ret < 0)
    {
        info.Format("ClearBuffTSPL Parameter,m_hPrinter error:%d",ret);
        MessageBox(info);
    }
    //Set page width, page height, equal to set buffer in the internal printer
    ret = PageSetupTSPL(m_hPrinter,45,56);
    switch(ret)
    {
        case TSPL_IDERROR:
            info.Format("PageSetupTSPL Parameter m_hPrinter Error:%d",ret);
            MessageBox(info);
            break;
        case TSPL_PARAM_LESS_EQUAL_ZERO:
            info.Format("PageSetupTSPL Parameter PageWidth or PageHeight under or equal,error
code:%d",ret);
            MessageBox(info);
            break;
        case TSPL_PARAM_GREAT_RANGE:
            info.Format("PageSetupTSPL Parameter PageWidth or PageHeight above spefified range,
error code:%d",ret);
            MessageBox(info);
            break;
    }

    //Send text content to previous internal buffer in printer invoked by PageSetupTSPL,
    but will not print immediately
    ret = DrawTextTSPL(m_hPrinter,0,0,TRUE,1,1,0,"DrawTextTSPL");
    switch(ret)
    {
        case TSPL_IDERROR:
            info.Format("DrawTextTSPL Parameter m_hPrinter Error:%d",ret);
            MessageBox(info);

```

```

        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("DrawTextTSPL Parameter start_x < 0 or start_y < 0 or xMultiplication < 1 or
yMultiplication < 1 or content == NULL,error code:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_GREAT_RANGE:
        info.Format("DrawTextTSPL Parameter xMultiplication or yMultiplication above error
code:%d",ret);
        MessageBox(info);
        break;
    }
    ...
    PrintTSPL21(m_hPrinter,1);//If TL21 invoked this method, only invoked this method will the printer carry
out printing action.
    POS_Port_Close(m_hPrinter); //close printing window

```

2.6 POS Function illustration

LONG POS_Port_OpenA/W(LPSTR szName, INT iPort, BOOL bFile, LPSTR szFilePath)

Function

Open port

Parameter

- szName:Port Name, For example
 - Serial: "COM1:9600,N,8,1"
 - Parallel: "LPT1"
 - USB: "SP-USB001"
 - Ethernet: "192.168.1.114";
- iPort: Port Type, 1000 is serial port, 1001 is parallel port, 1002 is USB port, 1003is Ethernet port;
- bFile: choose Whether to save the printing content as the local document
- szFilePath: when bFile is the really numerical value, it is the path of local document saving, otherwise this parameter will be ignored.

Return Numerical Value

- True: Printer handle;
- False: POS_ES_INVALIDPARA(-1)
 - POS_ES_OVERTIME(-5)
 - POS_ES_FILEOPENERROR(-6)
 - POS_ES_OTHERERRORS(-100)

LONG POS_Port_Close(LONG iPrinterID)**Function**

Close Port

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

LONG POS_Output_PrintFontStringA/W(LONG iPrinterID, INT iFont, INT iThick, INT iWidth, INT iHeight, INT iUnderLine, LPCSTR lpString)**Function**

Printing formatted characters string

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port
- iFont:when it is 0, printer choose standard ASCII Font A (12 × 24); when it is 1, printer choose compressed ASCII Font B (9 × 17);
- iThick:when it is 0, printer cancel over-striking mode; when it is 1, printer is over-striking mode
- iWidth:when it is 0, printer cancel the double width mode; when it is 1, printer is double width mode
- iHeight:when it is 0, printer cancel the double height mode; when it is 1, printer is double height mode
- iUnderLine:when it is 0, printer cancel the underline mode; when it is 1, printer is underline mode
- lpString:Character string ended with empty byte

Return Value:

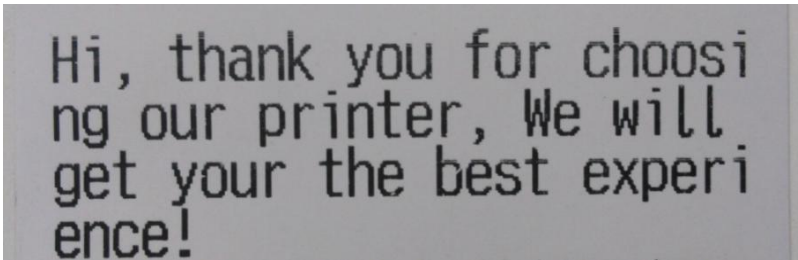
- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- Over striking mode is available for character and Chinese character, other mode are only available for character except over striking mode
- when choose twice width and twice height mode at the same time, the character will be magnified twice horizontally and vertically
- the content less than one row should end with “\r\n” to be printed

Printing Effect

POS_Output_PrintFontStringA(m_hPrinter,0,1,1,1,0, "Hi, thank you for choosing our printer, We will get your the best experience!\r\n");// Printing character string in iThick, iHeight, iWidth mode

**LONG POS_Control_SetRotaryPrint(LONG iPrinterID,INT n)****Function**

Choose/Cancel 90° rotation printing mode

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- n: when it is 0, printer cancel rotate 90 degree clockwise mode; when it is 1, printer choose rotate 90 degree clockwise mode

Return Numerical Value

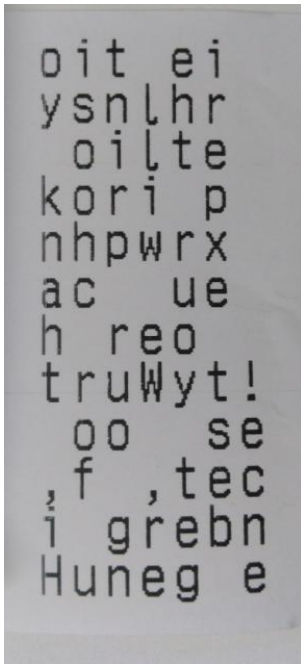
- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- 90° rotation printing mode is effective for character and Chinese character, while is noneffective for 1D barcode, 2D barcode and bitmap.
- This function doesn't support POS76 series printer

Printing Effect

POS_Control_SetRotaryPrint(m_hPrinter,1);// choose 90° rotation printing mode



LONG POS_Control_SetInvertedPrint(LONG iPrinterID,INT n)

Function

Choose/Cancel inversion printing mode

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port
- n:when it is 0, printer cancel inversion printing mode; when it is 1, printer choose inversion printing mode

Return Numerical Value

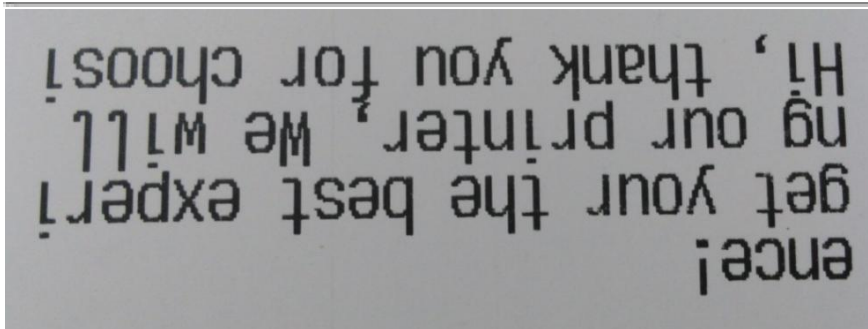
- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARAM(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- inversion printing mode is effective for character, Chinese character, 1D code, 2D code and bitmap.

Printing Effect

- *POS_Control_SetInvertedPrint(m_hPrinter,1);//choose inversion printing mode*



LONG POS_Control_OppositeColor(LONG iPrinterID, BOOL bOppsite)

Function

Choose/cancel reverse printing mode

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- bOppsite: 0 means cancel reverse printing, 1 means choose reverse printing

Return Numerical Value

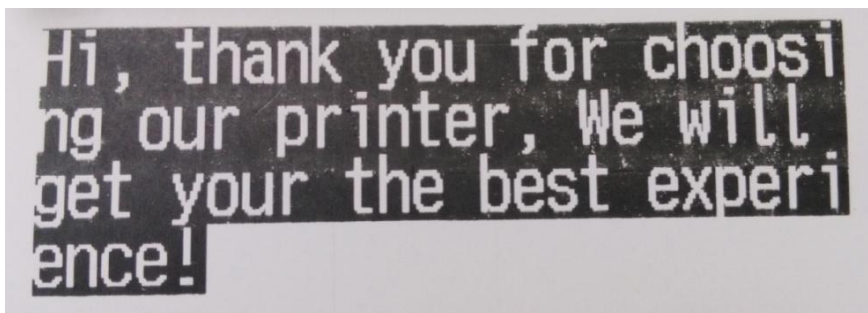
- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- Reverse printing mode is available for all character(except barcode character)
- Reverse printing mode in precedence to underline mode. Underline mode is ineffective in the reverse printing mode, the set underline will be effective after canceling the reverse printing mode
- Reverse printing mode hasn't influence on the space between the lines.
- The function don't support POS76 series printers

Printing Effect

POS_Control_OppositeColor(m_hPrinter, TRUE); //set reverse white printing mode



LONG POS_Control_AlignType(LONG iPrinterID, LONG iAlignType)**Function**

Setting character alignment method(left alignment/middle alignment/right alignment)

Parameter

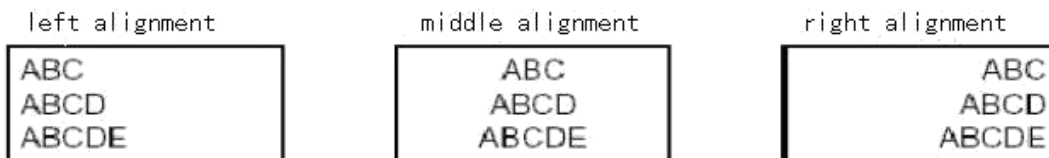
- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port
- iAlignType:Method of alignment, "0" Left alignment "1" Middle alignment "2" Right alignment

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- If have set printing area, then align in the printing area

Printing Effect**LONG POS_Output_PrintOneDimensionalBarcodeA/W(LONG iPrinterID, INT iType, INT iWidth, INT iHeight, INT hri, LPCSTR lpString)****Function**

Printing 1D barcode

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port
- iType:choose one dimensional barcode type. The types which can be chosen refer to constant definition list
- iWidth:choose barcode width, $2 \leq iWidth \leq 6$;
- iHeight:choose barcode height, $1 \leq iHeight \leq 255$;
- hri:choose the printing position of barcode character, which can be chosen are below
 - POS_HT_NONE:not print;
 - POS_HT_DOWN:print under the barcode
 - POS_HT_UP:print above the barcode
 - POS_HT_BOTH:print both under the barcode and above the barcode
- lpString:barcode data, please refer to the illustration of requirement for barcode data

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)

POS_ES_WRITEFAIL(-2)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

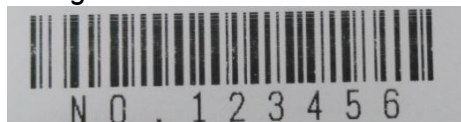
Illustration

- When choose POS_BT_UPCA、POS_BT_UPCE、POS_BT_JAN13、POS_BT_JAN8、POS_BT_CODE39、POS_BT_ITF、POS_BT_CODABAR, barcode data should end with "0"
- This function doesn't support POS76 series printers
- Barcode data requirement

Barcode type	Number of amount(n)	Data (d)
POS_BT_UPCA	$11 \leq n \leq 12$	$48 \leq d \leq 57$
POS_BT_UPCE	$11 \leq n \leq 12$	$48 \leq d \leq 57$
POS_BT_JAN13	$12 \leq n \leq 13$	$48 \leq d \leq 57$
POS_BT_JAN8	$7 \leq n \leq 8$	$48 \leq d \leq 57$
POS_BT_CODE39	$1 \leq n \leq 255$	$45 \leq d \leq 57, 65 \leq d \leq 90, 32, 36, 37, 43$
POS_BT_ITF	$1 \leq n \leq 255$	$48 \leq d \leq 57$
POS_BT_CODABA R	$1 \leq n \leq 255$	$48 \leq d \leq 57, 65 \leq d \leq 68, 36, 43, 45, 46, 47, 58$
POS_BT_CODE93	$1 \leq n \leq 255$	$0 \leq d \leq 127$
POS_BT_CODE12 8	$2 \leq n \leq 255$	$0 \leq d \leq 127$

Printing Effect

POS_Output_PrintOneDimensionalBarcodeA(m_hPrinter, POS_BT_CODE128, 2, 50, 4013, "NO.123456"); // Printing CODE128 barcode



LONG POS_Output_PrintTwoDimensionalBarcodeA(W(LONG iPrinterID, INT iType, INT parameter1, INT parameter2, INT parameter3, LPCSTR lpString)

Function

Printing 2D barcode

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port

- iType:choose two dimensional barcode type, two dimensional barcode types which can be chosen refer to constant definition list
- parameter1:
 - POS_BT_PDF417:means the character number of each row. The max value of v should less than the max value of model permit because different modes have different paper width
 - POS_BT_DATAMATRIX:means the height of figure
 - POS_BT_QRCODE:means the version number of figure
- parameter2:
 - POS_BT_PDF417:means the class of error correction
 - POS_BT_DATAMATRIX:means the width of figure
 - POS_BT_QRCODE:means the class of error correction (L:7%, M:15%,Q:25%,H:30%);
- parameter3:means the magnification times in lengthways, $1 \leq \text{parameter3} \leq 6$;
- lpString:barcode data, please refer to the illustration of requirement for barcode data

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

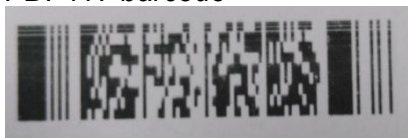
Illustration

- This function doesn't support POS76 series printers
- Barcode data requirement

Barcode type	Data length(n)	parameter1(v)	parameter2(r)	parameter3(k)	Data content(d)
QR Code	$1 \leq n \leq 65535$	$1 \leq v \leq 16$	$r=76,77,81,72$	$1 \leq k \leq 6$	$0 \leq d \leq 255$
Data Matrix	$1 \leq n \leq 65535$	$0 \leq v \leq 144$	$8 \leq r \leq 144$	$1 \leq k \leq 6$	$0 \leq d \leq 255$
PDF417	$1 \leq n \leq 65535$	$1 \leq v \leq 30$	$0 \leq r \leq 8$	$1 \leq k \leq 6$	$0 \leq d \leq 255$

Printing Effect

POS_Output_PrintTwoDimensionalBarcodeA(m_hPrinter,POS_BT_PDF417,2,6,6,"www.test.com");//Print PDF417 barcode



LONG POS_Output_SendLocalFileA(W(LONG iPrinterID, LPCSTR lpFilePath)

Function

Print local TXT document

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- lpFilePath: local TXT document store path

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_OVERTIME(-5)
 POS_ES_FILEOPENERROR(-6)
 POS_ES_OTHERERRORS(-100)

LONG POS_Output_DownloadRamBmpA/W(LONG iPrinterID, LPCSTR lpFilePath)**Function**

Download local monochrome bitmap to printer RAM

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- lpFilePath: Local single color bitmap storage path

Returned value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_NONMONOCHROMEBITMAP(-4)
 POS_ES_OVERTIME(-5)
 POS_ES_FILEOPENERROR(-6)
 POS_ES_OTHERERRORS(-100)

Illustration

- The size of downloaded pic should not beyond the printer RAM area, printer will exit directly if oversize and the data will be regarded as ordinary data.
- The horizontal width of pic should less than the max printable width, the width of 80mm paper is 576dots, the width of 58mm paper is 384dots
- This function can download pic (such as LOGO, digital signature) into the printer RAM before printing, Invoking POS_Output_PrintRamBmp function for printing, it can reduce the waiting time of data transmission when printing pics.
- This function doesn't support POS76 series printers and POS58 series printers

LONG POS_Output_PrintRamBmp(LONG iPrinterID,INT n)**Function**

Print local monochrome bitmap downloaded to printer RAM according to the chosen mode

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- n:choose bitmap printing mode, $0 \leq n \leq 3$;

n	mode	vertical resolution (DPI)	lateral resolution (DPI)
0	normal	203	203
1	Twice width	203	101
2	Twice height	101	203
3	Twice width&Twice height	101	101

Illustration

- This function doesn't support POS76 series printers and POS58 series printers

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

LONG POS_Output_PrintBmpDirectA/W(LONG iPrinterID, LPCSTR lpFilePath)**Function**

Print local single color bitmap

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- lpFilePath: local monochrome bitmap store path

Illustration

This function doesn't support POS76 series printers

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_NONMONOCHROMEBITMAP(-4)
POS_ES_OVERTIME(-5)

POS_ES_FILEOPENERROR(-6)

POS_ES_OTHERERRORS(-100)

LONG POS_Output_PrintStringA/W(LONG iPrinterID, LPSTR lpString)

Function

Print characters string

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- lpString: characters string which end with the empty byte

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

Illustration

- the content less than one row should end with “\r\n” to be printed

LONG POS_Output_PrintFlashBmp(LONG iPrinterID, INT n)

Function

Print pre-downloaded bitmap

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- n: bitmap store number, $1 \leq n \leq 8$;

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

Illustration

- This function should be used with printer setting tool, firstly download bitmap in specific position by setting tool, then use the function to print

LONG POS_Control_PrintTestpage(LONG iPrinterID)**Function**

Print test page

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port;

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

LONG POS_Output_PrintData(LONG iPrinterID, LPCSTR lpBuffer, INT iLength)**Function**

Send hexadecimal data to printer

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- lpBuffer:The data will be sent in the buffer;
- iLength:the size of buffer

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Invoking example

```

POS_Output_PrintStringA(m_hPrinter, "Hello world!\r\n");Print"Hello world!"
byte[] cmd1 = { 0x48,0x65,0x6C,0x6C,0x6F,0x20,0x77,0x6F,0x72,0x6C,0x64,0x21, 0x0A};
POS_Output_PrintData(m_hPrinter, cmd1, 13);Print"Hello world!"
...
byte[] cmd2 = { 0x1d, 0x56, 0x42, 0x00 };//cutting paper instruction
LONG ret = POS_Output_PrintData(m_hPrinter, cmd2, 4);//Sent hex data
//identify sending printing content successfully or not
switch (ret)
{
case POS_ES_SUCCESS:
    MessageBox(_T("sending successfully"));
    break;
case POS_ES_INVALIDPARA:
    MessageBox(_T("Parameter error"));
    break;
case POS_ES_WRITEFAIL:
    MessageBox(_T("sending failed"));
    break;
case POS_ES_OVERTIME:
    MessageBox(_T("overtime"));
    break;
case POS_ES_OTHERERRORS:
    MessageBox(_T("other error"));
    break;
}
...

```

LONG POS_Control_SetPrintPosition(LONG iPrinterID, LONG iLeftMargin, LONG iWidth)**Function**

Set the printing left margin and printing width

Parameter

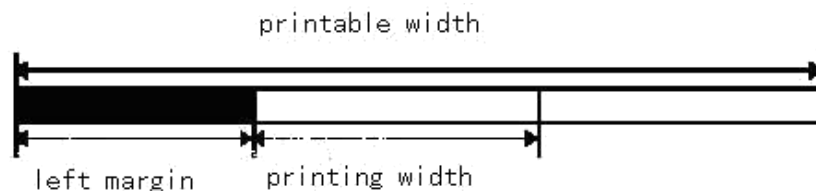
- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- iLeftMargin:left margin, default is "0"
- iWidth:set printing area, default is "0"

Return Numerical Value

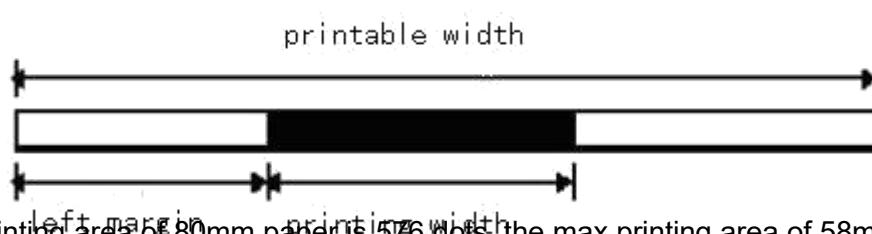
- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- Use iLeftMargin to set left margin, iLeftMargin is the left margin printing dots which need to be set



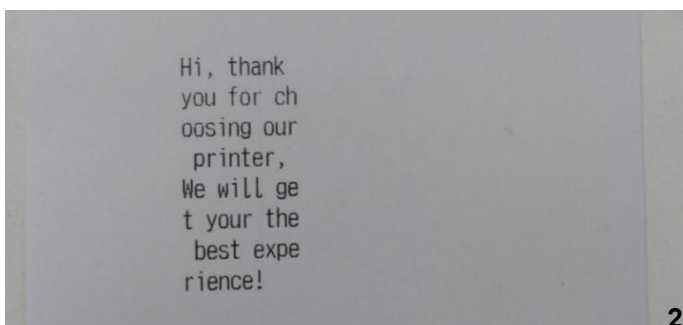
- Use iWidth to set printing area, iWidth is the printing dots which need to be set in printing area. If [left margin+printing width] is over the printable area, the printing area = the printable area -left margin



- the max printing area of 80mm paper is 576 dots, the max printing area of 58mm paper is 384 dots, the max printing area of 76mm paper is 420dots,
- This function doesn't support POS76 series printers and POS58 series printers

Print Effect

POS_Control_SetPrintPosition(m_hPrinter, 80, 80);//set print left margin to 80dots, then printable width is 80dots



LONG POS_Control_SetLineSpace(LONG iPrinterID, INT iSpace)**Function**

Setting printing height of line(the height of character+line space)

Parameter

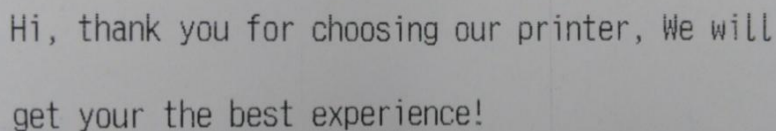
- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- iSpace:set height of line, $0 \leq n \leq 255$;

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Print Effect

POS_Control_SetLineSpace(m_hPrinter, 80);//set the height of line to 80 dots



Hi, thank you for choosing our printer, We will
get your the best experience!

LONG POS_Control_SetPrintFontE(LONG iPrinterID,BOOL iFont,BOOL iBold,BOOL iDoubleWidth,BOOL iDoubleHeight, BOOL iUnderLine)**Function**

Set the print format of western characters font

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- iFont: Font: true=Normal font(12x24), false=Compressed font(9x17);
- iBold: Over striking; true=over striking, false=normal;
- iDoubleWidth: Twice width: true=twice width, false=normal;
- iDoubleHeigh: Twice height: true=twice height, false=normal;
- iUnderLine: Underline: true=underline, false=non-underline;

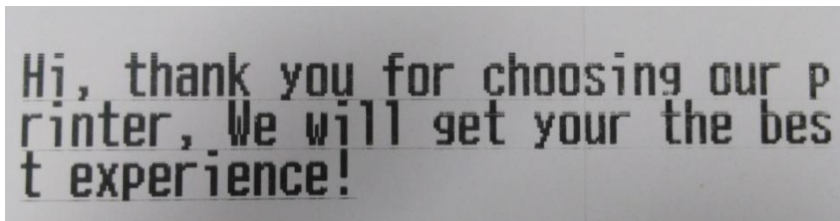
Return Numerical Value

- Ture:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)

POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

Print effect

POS_Control_SetPrintFontE(m_hPrinter,true,true,true,true);//Set the print format of western character font



**LONG POS_Control_SetPrintFontC(LONG iPrinterID,BOOL iDoubleWidth,BOOL
 iDoubleHeight,BOOL iUnderLine)**

Function

Set the print format of Chinese character font

Parameter

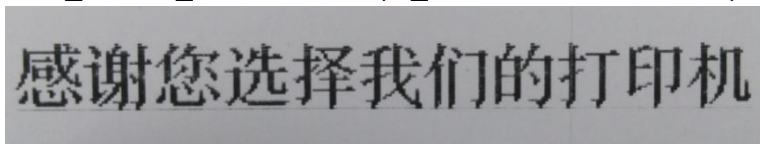
- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port;
- iFont:Font:true=Normal font(12x24), false=Compressed font(9x17);
- iDoubleWidth:Twice width:true=twice width, false=normal;
- iDoubleHeight:Twice height:true=twice height, false=normal;
- iUnderLine:Underline:true=underline, false=non-underline;

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

Print Effect

POS_Control_SetPrintFontC(m_hPrinter, true, true, true);//Set the print format of Chinese character font



LONG POS_Control_CutPaper(LONG iPrinterID, LONG iType, LONG iLines)

Function

Content of printing buffer, feeding paper line which set by ilines and cutting paper

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- iType: Cutter type, 1 "Full Cut" 0 "Partial Cut"
- iLines: The line of paper feeding;

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- full cut/partial cut need the support of cutter
- This function doesn't support POS58 series printer

LONG POS_Control_FeedLines(LONG iPrinterID, LONG iLines)**Function**

feeding paper line which is set by parameter ilines

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- iLines: The line of feeding paper forward, $0 \leq iLines \leq 255$;

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

LONG POS_Control_CashDraw(LONG iPrinterID, LONG iPort, LONG iTime1, LONG iTime2)**Function**

Open cash drawer

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- iPort: Cash drawer number: 1 means to open 1# cash drawer, 2 means to open 2# cash drawer, 3 means to open cash drawer 1# 2# cash drawers.
- iTime1: Time of open cash drawer [iTime1 × 2 ms], $0 \leq iTime1 \leq 255$;
- iTime2: Time of close cash drawer [iTime2 × 2 ms], $0 \leq iTime2 \leq 255$;

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
 - POS_ES_WRITEFAIL(-2)
 - POS_ES_OVERTIME(-5)
 - POS_ES_OTHERERRORS(-100)

Illustration

- the time of opening cash drawer should be determined by the used cash draw parameter

LONG POS_Control_BlackMark(LONG iPrinterID)**Function**

Feeding paper to black mark position

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 - POS_ES_WRITEFAIL(-2)
 - POS_ES_OVERTIME(-5)
 - POS_ES_OTHERERRORS(-100)

LONG POS_Status_RTQueryStatus(LONG iPrinterID)**Function**

Checking status of printer paper

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port

Return Numerical Value

- True: POS_ES_SUCCESS(0)
 - POS_ES_PAPEROUT(1)
- False: POS_ES_INVALIDPARA(-1)
 - POS_ES_WRITEFAIL(-2)
 - POS_ES_READFAIL(-3)
 - POS_ES_OVERTIME(-5)
 - POS_ES_OTHERERRORS(-100)

Illustration

- This function is to check status of printer paper, paper in return POS_ES_SUCCESS (0), paper out

```
return POS_ES_PAPEROUT(1)
```

Invoking example

```
...
LONG ret = POS_Status_RTQueryStatus(m_hPrinter);
switch (ret)
{
case POS_ES_SUCCESS:
    MessageBox(_T("paper in"));
    break;
case POS_ES_PAPEROUT:
    MessageBox(_T("paper out"));
    break;
case POS_ES_INVALIDPARA:
    MessageBox(_T("Parameter error"));
    break;
case POS_ES_WRITEFAIL:
    MessageBox(_T("writing failed"));
    break;
case POS_ES_READFAIL:
    MessageBox(_T("reading failed"));
    break;
case POS_ES_OVERTIME:
    MessageBox(_T("overtime"));
    break;
case POS_ES_OTHERERRORS:
    MessageBox(_T("other error"));
    break;
}
...
```

LONG POS_Status_RTQueryTypeStatus(LONG iPrinterID,INT n)

Function

Check status of printer status

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- n: check type number, 1、2、3、4;

Return Numerical Value

- True:POS_ES_SUCCESS(0)
 POS_ES_PAPEROUT(1)
 POS_ES_DOOROPEN(3)
 POS_ES_CUT(4)

- POS_ES_DRAWERHIGH(5)
- POS_ES_PAPERENDING(6)
- False:POS_ES_INVALIDPARA(-1)
- POS_ES_WRITEFAIL(-2)
- POS_ES_READFAIL(-3)
- POS_ES_OVERTIME(-5)
- POS_ES_OTHERERRORS(-100)

Illustration

- Parameter n means: n=1, printer status; n=2, off-line status; n=3, error status; n=4, paper sensor status;

Invoking example

```

...
//take parameter n=2, check printer off-line status as example

int nDoorOpen = 0;    //paper case open

LONG ret = POS_Status_RTQueryTypeStatus(m_hPrinter,2);
switch (ret)
{
case POS_ES_INVALIDPARA:
    MessageBox(_T("Parameter error"));
    return;
    break;
case POS_ES_WRITEFAIL:
    MessageBox(_T("writing failed"));
    return;
    break;
case POS_ES_READFAIL:
    MessageBox(_T("Reading failed"));
    return;
    break;
case POS_ES_OVERTIME:
    MessageBox(_T("over time"));
    return;
    break;
case POS_ES_OTHERERRORS:
    MessageBox(_T("other error"));
    return;
    break;
}
if (ret == POS_ES_DOOROPEN)
{
    nDoorOpen = 1;
}

```

```

CString strMessage;
strMessage = _T("#");

if (nDoorOpen == 1)
{
    strMessage = strMessage + _T("paper case open") + _T("#");
}
else
{
    strMessage = strMessage + _T("paper case close") + _T("#");
}
MessageBox(strMessage);
...

```

LONG POS_Status_QueryTaskStatus(LONG iPrinterID,INT second)

Function

Check the printing task finished or not

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port
- Second: it is unit second, that is over time value set by users, which should be more than the time of printing task; if the time of printing task is more than the time function set by user, which lead to printing task unfinished, at this time, if the setting time value is less than 5seconds, printer will regard it as 5 seconds.

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_READFAIL(-3)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

Illustration

- The function is to check the printer execute this command or not, that is to check the print task before this command is finished or not.
- This function doesn't support POS76 series printers

Invoking example

```

...
Send printing task
...
//overtime value second 为 5s
LONG ret = POS_Status_QueryTaskStatus(m_hPrinter,5);
switch (ret)

```

```

{
case POS_ES_SUCCESS:
    MessageBox(_T("printing finished"));
    break;
case POS_ES_READFAIL:
    MessageBox(_T("reading data error"));
    break;
case POS_ES_INVALIDPARA:
    MessageBox(_T("parameter error"));
    break;
case POS_ES_WRITEFAIL:
    MessageBox(_T("sending failed"));
    break;
case POS_ES_OVERTIME:
    MessageBox(_T("task printing..."));
    break;
case POS_ES_OTHERERRORS:
    MessageBox(_T("other error"));
    break;
}

```

...

LONG POS_Input_PrinterId(LONG iPrinterID, char *buf)

Function

Get printer factory information

Parameter

- iPrinterID: Printer handle, it will be confirmed by the return Numerical Value from the opened port
- char *buf: receiving data buf defined by user

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
 POS_ES_WRITEFAIL(-2)
 POS_ES_READFAIL(-3)
 POS_ES_OVERTIME(-5)
 POS_ES_OTHERERRORS(-100)

Illustration

- The function returned factory information is “_” + “factory name”

LONG POS_Control_ReSet(LONG iPrinterID)**Function**

Initialize the printer

Parameter

- iPrinterID:Printer handle, it will be confirmed by the return Numerical Value from the opened port

Return Numerical Value

- True: POS_ES_SUCCESS (0)
- False: POS_ES_INVALIDPARA(-1)
POS_ES_WRITEFAIL(-2)
POS_ES_OVERTIME(-5)
POS_ES_OTHERERRORS(-100)

Illustration

- The function is to initialize the printer, and to clear the data of printing buffer. The printer mode will be set in the default mode when power on.

LONG Color24_GrayBW(LPCTSTR szSourceFile,LPCTSTR szTargetFile)**Function**

To convert the 24bit bitmap into monochrome bitmap

Parameter

- szSourceFile: The store path of bitmap needed to be converted
- szTargetFile: The store path of converted bitmap

Return Numerical Value

- True:POS_ES_SUCCESS (0)
- False:POS_ES_INVALIDPARA(-1)
POS_ES_READFAIL(-3)
POS_ES_FILEOPENERROR (-6)
POS_ES_OTHERERRORS(-100)

2.7 TSPL Function illustration**LONG SetIs21();****Function**

To confirm the model is TL21, to distinguish internal command and related condition

Parameter

- None

Returned value

- True: TSPL_SUCCESS(0)
- False: None(no error returned value for the method)

Invoked example

SetIs21();//If it is TL21, printer will need this method

LONG PageSetupTSPL(LONG iPrintID, INT PageWidth, INT PageHeight);**Function**

Set page width and page height, build buffer in internal printer

Parameter

- iPrintID: printer handle is decided by returned value which open port
- PageWidth: Page width
- PageHeight: Page height

Returned value

- True: TSPL_SUCCESS(0)
- False: TSPL_IDERROR(-1): iPrintID Return this error when got error

TSPL_PARAM_LESS_EQUAL_ZERO(-2): When using TL21, return this error when ParameterPageWidth < 1 or PageHeight < 1. When using TL51, return this error when ParameterPageWidth < 1 or PageHeight < 1.

TSPL_PARAM_GREAT_RANGE (-3): When using TL21, return this error when ParameterPageWidth > 45 or PageHeight > 56. When using TL51, return this error when ParameterPageWidth > 100 or PageHeight > 225.

- Others: TL51printer max label square is length*width= 225X100mm
TL21printer max label square is length*width= 56X45mm

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//If TL21, the printer will need this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameter m_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,45,56);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPL error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:

```

```

        info.Format("PageSetupTSPLParameterPageWidth or PageHeight under or equal 0, error
code:%d",ret);
        MessageBox(info);
        break;

        case TSPL_PARAM_GREAT_RANGE:
            info.Format("PageSetupTSPLParameterPageWidth or PageHeight above specified range,
error code:%d",ret);
            MessageBox(info);
            break;
    }

```

LONG DrawLineTSPL(LONG iPrintID,INT StartX, INT StartY, INT LineWidth, INT LineHeight);

Function

Draw line

Parameter

- iPrintID: Printer handle, is decided by returned value which open port
- startX: Start at left above level direction, in dots(cannot exceed page width)
- startY: Start at left top corner in vertical direction, in dots(cannot exceed page height)
- lineHeight: Line height, in dots(cannot exceed page height)
- lineLength: Line width, in dots(cannot exceed page width)

Returned value

- True: TSPL_SUCCESS(0)
- False: TSPL_IDERROR(-1): iPrintID Return this error when error occur
TSPL_PARAM_LESS_EQUAL_ZERO(-2): Parameter StartX under 0 or StartY under 0 or LineWidth under 0 or LineHeight under 0 return this error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 need this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameterm_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,45,56);
switch(ret)
{
    case TSPL_IDERROR:

```

```

        info.Format("PageSetupTSPLParameter_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth 或 PageHeight under or equal to 0 ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight above or equal to zero ,
error code:%d",ret);
        MessageBox(info);
        break;
}

ret = DrawLineTSPL(m_hPrinter,10,10,100,100);
switch(ret)
{
    case TSPL_IDERROR:::
        info.Format("DrawLineTSPLParameter_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("DrawLineTSPLParameter StartX under 0 or StartY under 0 or LineWidth
under 0 or LineHeight under 0 return this error, error code:%d",ret);
        MessageBox(info);
        break;
}
PrintTSPL21(m_hPrinter,1);
//PrintTSPL(m_hPrinter,1,1);//TL51 need this method

```

LONG PrintTSPL21(LONG iPrintID,INT Set);

Function

Only received PRINT order will the printer carry out printing action, otherwise do not print

Parameter

- iPrintID: Printer handle, is decided by returned value which opened the port
- Set: Specify the printing pieces(set)

Returned value

- True: TSPL_SUCCESS(0)
- Error: TSPL_IDERROR(-1): Parameter iPrintID returned this error

TSPL_PARAM_LESS_EQUAL_ZERO(-2): Parameter Set returned this value when the value is under 1

TSPL_PARAM_GREAT_RANGE(-3): Parameter Set above 65535

说明

- TL21 model printing method

LONG PrintTSPL51(LONG iPrintID,INT Set,INT Copy);

Function

Only received PRINT order will the printer carry out printing action, otherwise do not print

Parameter

- iPrintID: Printer handle, is decided by returned value which opened the port
- Set: Specify the printing pieces(set)
- Copy: Print how many pieces for every barcode page on every piece

Returned value

- True: TSPL_SUCCESS(0)
- Error: TSPL_IDERROR(-1): Parameter iPrintID return this error
 TSPL_PARAM_LESS_EQUAL_ZERO(-2): Parameter Set or Copy under 1 return this value
 TSPL_PARAM_GREAT_RANGE(-3): Parameter Set or Copy above 65535 return this value

Description

- TL51 model printing method

LONG DrawBorderTSPL(LONG iPrintID,INT LineWidth, INT top_left_x, INT top_left_y, INT bottom_right_x, INT bottom_right_y);

Function

Draw frame

Parameter

- iPrintID: printer handle, is decided by Returned value of opened port
- lineWidth: rectangle line width, in dots (cannot exceed label width)
- top_left_x: the start position x on up left corner on horizontal direction of rectangle, in dots(cannot exceed page width)
- top_left_y: start position y at up left corner in vertical direction of rectangle, in dots(cannot exceed page height)
- bottom_right_x: the end position x on right bottom corner on horizontal direction of rectangle, in dots(cannot exceed page width)
- bottom_right_y: the end position y on right bottom corner on vertical direction of rectangle, in

dots(cannot exceed page height)

Returned value

- True: TSPL_SUCCESS(0)
- False: TSPL_IDERROR(-1): ParameteriPrintID return this error
 TSPL_PARAM_LESS_EQUAL_ZERO(-2): 当 ParameterLineWidth < 0 or top_left_x < 0 or top_left_y < 0 or bottom_right_x < 0 or bottom_right_y < 0, return this error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 need this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameterm_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,45,56);
switch(ret)
{
    case TSPL_IDERROR::
        info.Format("PageSetupTSPL error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth 或 PageHeight under or equal to 小于
或 0, error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight above specified range,
error code:%d",ret);
        MessageBox(info);
        break;
}

ret = DrawBorderTSPL(m_hPrinter,5,20,20,200,200);
switch(ret)
{
    case TSPL_IDERROR::
        info.Format("DrawBorderTSPL error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("DrawBorderTSPLParameterLineWidth < 1 or top_left_x < 1 or top_left_y < 1

```

```

or bottom_right_x < 1 or bottom_right_y < 1, error code:%d",ret);
    MessageBox(info);
    break;
}
PrintTSPL21(m_hPrinter,1);
//PrintTSPL(m_hPrinter,1,1);//TL51 printer need this method

```

LONG DrawTextTSPL(LONG iPrintID,INT start_x, INT start_y, BOOL isSimplifiedChinese, INT xMultiplication,INT yMultiplication, INT rotate, CString content);

Function

Print text

Parameter

- iPrintID: printer handle, is decided by returned value of opened port.
- start_x: Starting position coordinate of words on x direction, in dots(cannot exceed page width)
- start_y: Starting position coordinate of words on y direction,(cannot exceed page height)
- isSimplifiedChinese: true simplified Chinese24×Font(GB Code);false traditional Chinese 24×Font(BIG 5)
- xMultiplication: Magnification times range on X direction 1—4
- yMultiplication: Magnification times range on Y direction 1—4
- rotate: rotate angle of text, clockwise is 0, not rotate 90, clockwise 180, clockwise 270, value is 0,90,180,270
- content: The content which is going to be printed, please use \" to print double quotes mark in the program.

Please use \[R] to print CR in program if print D(hex) characters
 use \[A] to print LF in program if print D(hex) characters

Please

Returned value

- True: TSPL_SUCCESS(0)
- Error: TSPL_IDERROR(-1): Parameter iPrintID, return this error
 TSPL_PARAM_LESS_EQUAL_ZERO(-2): when Parameter start_x < 0 or start_y < 0 or xMultiplication < 1 or yMultiplication < 1 or content == NULL, return this error
 TSPL_PARAM_GREAT_RANGE(-3): Parameter xMultiplication or yMultiplication is above 4, or rotate Parameter value is among 0,90,180,270, return this error

Invoke example:

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)

```

```

{
    info.Format("ClearBuffTSPLParametterm_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,45,56);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPLParametterm_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth 或 PageHeight under or equal to 0 ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is above specified
range, error code, error code:%d",ret);
        MessageBox(info);
        break;
}

ret = DrawTextTSPL(m_hPrinter,0,0,TRUE,1,1,0,"DrawTextTSPL");
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("DrawTextTSPLParametterm_hPrinter error code:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("DrawTextTSPL Parameterstart_x < 0 or start_y < 0 or xMultiplication < 1 or yMultiplication
< 1 or content == NULL,error code:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_GREAT_RANGE:
        info.Format("DrawTextTSPL ParameterxMultiplication or yMultiplication above 4
Error code:%d",ret);
        MessageBox(info);
        break;
}
PrintTSPL21(m_hPrinter,1);
//PrintTSPL(m_hPrinter,1,1);//TL51 printer use this method

```

LONG DrawBarCodeTSPL(LONG iPrintID,INT start_x, INT start_y, CString type, INT height, BOOL isReadable,INT rotate, INT narrowWidth, INT wideWidth, CString content);

Function

Print 1D barcode

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- start_x: Upper left corner position on horizontal direction of barcode, in dots(cannot exceed page width)
- start_y: Upper left corner position on vertical direction of barcode, in dots(cannot exceed page height)
- type: barcode type, including "UPCA", "UPCE", "EAN13", "EAN8", "39", "CODABAR", "93", "128"
- height: barcode height, in dots, (cannot exceed page height)
- isReadable: FALSE, cannot be seen by human eyes, TRUE, can be seen by human eyes
- rotate: text rotate angle, clockwise is 0, not rotate 90, clockwise 180, clockwise 270, value is 0,90,180,270
- narrowWidth: width of narrow strip of barcode, in dots
- wideWidth: width of wide strip of barcode, in dots
- content: Barcode content, can only be number numeric string, otherwise will print error.

Returned value

- True: TSPL_SUCCESS(0)
- False: TSPL_IDERROR(-1): Parameter iPrintID, returned this error
TSPL_PARAM_LESS_EQUAL_ZERO(-2): start_x < 0 or start_y < 0 or height < 0 or rotate < 0 or narrowWidth < 0 or wideWidth < 0 or content == NULL, return this error
TSPL_PARAM_GREAT_RANGE(-3): rotate Parameter value is not among 0,90,180,270 or type is not "UPCA", "UPCE", "EAN13", "EAN8", "39", "CODABAR", "93", "128", return this error.

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameter m_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,45,56);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);

```

```

        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight under or equal to 0 ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth 或 PageHeight above specified range,
error code:%d",ret);
        MessageBox(info);
        break;
}

ret = DrawBarCodeTSPL(m_hPrinter,10,10,"128",30,TRUE,0,2,4,"123456");
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("DrawBarCodeTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("DrawBarCodeTSPL Parameter start_x < 0 or start_y < 0 or height < 0 or rotate < 0 or
narrowWidth < 0 or wideWidth < 0 or content == NULL

        error code:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_GREAT_RANGE:
        info.Format("DrawBarCodeTSPL Parameter rotateParameter value is not among
0,90,180,270, or type is not \"UPCA\", \"UPCE\", \"EAN13\", \"EAN8\", \"39\", \"CODABAR\", \"93\", \"128\" error
code:%d",ret);
        MessageBox(info);
        break;
}
PrintTSPL21(m_hPrinter,1);
//PrintTSPL(m_hPrinter,1,1); //TL51 printer need to use this method

```

LONG ClearBuffTSPL(LONG iPrintID);

Function

Clear the printer buffer

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port

Returned value

- True: TSPL_SUCCESS(0)
- Error: TSPL_IDERROR(-1): Parameter iPrintID, return this error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameter m_hPrinter error:%d",ret);
    MessageBox(info);
}

```

LONG GetPrinterStatusTSPL(LONG iPrintID)

Function

Gain printer status

Parameter

iPrintID: Printer handle, is confirmed by returned value of opened port

Returned value

- True: TSPL_SUCCESS(0)
- False: TSPL_IDERROR(-1): Parameter iPrintID, return this value
 - TSPL_PRINTER_STATUS_OUTPAPER(1): printer is paper out
 - TSPL_PRINTER_STATUS_WORK(2): printing
 - TSPL_PRINTER_STATUS_ENCLOSURENOCLOSE(3): paper house opened
 - TSPL_PRINTER_STATUS_ERROR(4): printer internal error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need this method
ret = GetPrinterStatusTSPL(m_hPrinter);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("GetPrinterStatusTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PRINTER_STATUS_OUTPAPER:

```

```

        info.Format("GetPrinterStatusTSPL, printer is paper out:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PRINTER_STATUS_WORK:
        info.Format("GetPrinterStatusTSPL,printing:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PRINTER_STATUS_ENCLOSURENOCLOSE:
        info.Format("GetPrinterStatusTSPL, paper house opened:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PRINTER_STATUS_ERROR:
        info.Format("GetPrinterStatusTSPL,printer internal error:%d",ret);
        MessageBox(info);
        break;
}

```

LONG DriveBeepTSPL(LONG iPrintID)

Function

Drive the buzzer beep one sound

Parameter

iPrintID: Printer handle, is confirmed by returned value of opened port

Returned value

- True:TSPL_SUCCESS(0)
- False:TSPL_IDERROR(-1): ParameteriPrintID , return this value

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need this method
ret = DriveBeepTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("DriveBeepTSPLParameter m_hPrinter error:%d",ret);
    MessageBox(info);
}

```

LONG SetPaperbackOrPaperFeedTSPL(LONG iPrintID,BOOL isFeedBack, INT mDot)

Function

Control paper feeding or back

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- isFeedBack: whether back paper, true back paper
- mDot: $1 \leq mDot \leq 1000$

Returned value

- True: TSPL_SUCCESS(0)
- False: TSPL_IDERROR(-1): Parameter iPrintID, return this error
TSPL_PARAM_GREAT_RANGE(-3): when Parameter mDot less than 1 or more than 1000, return this error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need to use this method
ret = SetPaperbackOrPaperFeedTSPL(m_hPrinter,TRUE,50);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("SetPaperbackOrPaperFeedTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_GREAT_RANGE:

        info.Format("SetPaperbackOrPaperFeedTSPL when Parameter mDot is under 1 or above1000,
return this value:%d",ret);
        MessageBox(info);
        break;
}

```

LONG ReverseAreaTSPL(LONG iPrintID,INT start_x, INT start_y, INT width, INT height)**Function**

Reverse printing in specified area

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- start_x: Top left corner X coordinate in the area, in dots,(can't exceed the page width)
- start_y : Top left corner Y coordinate in the area, in dots,(can't exceed the page height)
- width:area width ,in dots(can't exceed page width)

- height: area height ,in dots(can't exceed page height)

Returned value

- True: TSPL_SUCCESS(0)
- Error: TSPL_IDERROR(-1): ParameteriPrintID , return this error
TSPL_PARAM_LESS_EQUAL_ZERO(-2): start_x < 0 or start_y < 0 or height < 0 or width < 0

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need to use this method
ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameter_m_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,45,56);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPLParameter_m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is under or equal to 0, error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is above specified scope, error code:%d",ret);
        MessageBox(info);
        break;
}

ret = ReverseAreaTSPL(m_hPrinter,10,10,45,56);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("ReverseAreaTSPLParameter_m_hPrinter error:%d",ret);
        MessageBox(info);
        break;

```

```

        case TSPL_PARAM_LESS_EQUAL_ZERO:
            info.Format("ReverseAreaTSPLParameterstart_x < 0 or start_y < 0 or height
< 0 or width < 0, error code:%d",ret);
            MessageBox(info);
            break;
    }
    PrintTSPL21(m_hPrinter,1);//TL21 printer need to use this method
    //PrintTSPL(m_hPrinter,1,1);//TL51 printer need to use this method

```

LONG SetGAPTSPL(LONG iPrintID,DOUBLE value)

Function

Set the vertical distance between labels

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- value: $0 \leq \text{value} \leq 25.4$ (mm)

Returned value

- True:TSPL_SUCCESS(0)
- Fault:TSPL_IDERROR(-1): ParameteriPrintID , return this error
TSPL_PARAM_GREAT_RANGE(-3): when Parametervalue is under or equal to 1, or
above or equal to 25.4, return this error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 need to use this method
ret = SetGAPTSPL(m_hPrinter,10);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("SetGAPTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_GREAT_RANGE:
        info.Format("SetGAPTSPL when Parametervalue is under or equal to 1, or above or equal
to 25.4, return this error error:%d",ret);
        MessageBox(info);
        break;
}

```

LONG SetLabelReferenceTSPL(LONG iPrintID,INT x,INT y)**Function**

Define original reference coordinate of label

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- X: Coordinate position on horizontal direction, in dots(cannot exceed page width)
- y: Coordinate position on vertical direction, in dots(cannot exceed page height)

Returned value

- True:TSPL_SUCCESS(0)
- Fault:TSPL_IDERROR(-1): ParameteriPrintID, return this error
TSPL_PARAM_LESS_EQUAL_ZERO(-2): $x < 0 \parallel y < 0$, return this error

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 printer need to use this method
ret = SetLabelReferenceTSPL(m_hPrinter,10,10);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("SetLabelReferenceTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("SetLabelReferenceTSPLParameter x < 0 || y < 0 返回此 error, error code:%d",ret);
        MessageBox(info);
        break;
}

```

LONG DownloadBitMapTSPL(LONG iPrintID,BOOL isMoveFlash,CString PathName)**Function**

Download bitmap to printer

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- isMoveFlash: false: download bitmap to printer memory(data in printer memory will lost if loose power); true: download to flash
- PathName: Download to memory or FLASH, path or name, name could contain 8 characters at most.

Returned value

- True:TSPL_SUCCESS(0)
- Fault:TSPL_IDERROR(-1): ParameteriPrintID got error or failed to open file, return this error
TSPL_PARAM_LESS_EQUAL_ZERO(-2): ParameterPathName==NULL return this error

Description

- TL21 model does not support this method at present, TL51 support.

Invoke example

```

LONG ret =0;
CString info;

ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameterm_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,100,225);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPLParameterm_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is under or equal to 0 ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is above specified
range, error code:%d",ret);
        MessageBox(info);
        break;
}

ret = DownloadBitMapTSPL(m_hPrinter,FALSE,"D://LOG.bmp");
switch(ret)
{

    case TSPL_IDERROR:
        info.Format("DownloadBitMapTSPLParameterm_hPrinter error or failed to open

```

```

file:%d",ret);
    MessageBox(info);
    break;
case TSPL_PARAM_LESS_EQUAL_ZERO:
    info.Format("DownLoadBitMapTSPLParameterPathName==NULL:%d",ret);
    MessageBox(info);
    break;

}

ret = PutBitMapTSPL(m_hPrinter,10,10,"D://LOG.BMP");
switch(ret)
{

case TSPL_IDERROR:
    info.Format("PutBitMapTSPLParameter_hPrinter error:%d",ret);
    MessageBox(info);
    break;
case TSPL_PARAM_LESS_EQUAL_ZERO:
    info.Format("PutBitMapTSPLParameterfileName==NULL:%d",ret);
    MessageBox(info);
    break;

}

PrintTSPL(m_hPrinter,1,1);

```

LONG Draw2DBarcodeTSPL(LONG iPrintID,INT start_x, INT start_y, CString Max ,CString content)

Function

Print 2D barcode

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- start_x: coordinate position on horizontal direction, in dots(cannot exceed page width)
- start_y: coordinate position on vertical direction, in dots(cannot exceed page height)
- Max: module size, range("x1" ~ "x6") string
- Content: 2D Barcode content

Returned value

- True:TSPL_SUCCESS(0)
- False:TSPL_IDERROR(-1): ParameteriPrintID got error, return this error

TSPL_PARAM_LESS_EQUAL_ZERO(-2): Parameterstart_x < 0 or start_y < 0 or content == NULL

TSPL_PARAM_GREAT_RANGE(-3): when max Parameter value is not among string"x1" ~ "x6" range, then return this error.

Description

- TL21 model does not support this method, TL51 supported.

Invoke example

```

LONG ret = 0;
CString info;

ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)
{
    info.Format("ClearBuffTSPLParameter m_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,100,225);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameter PageWidth or PageHeight is under or equal to 0 ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameter PageWidth or PageHeight is above specified
range, error code:%d",ret);
        MessageBox(info);
        break;
}

ret = Draw2DBarCodeTSPL(m_hPrinter,10,10,"x1","12345");
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("Draw2DBarCodeTSPLParameter m_hPrinter error:%d",ret);

```

```

        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("Draw2DBarcodeTSPLParameterstart_x < 0 or start_y < 0 or content==NULL ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("Draw2DBarcodeTSPL when max Parameter value is not among
string x1 ~ x6 range, then return this error, error code:%d",ret);
        MessageBox(info);
        break;

}
PrintTSPL(m_hPrinter,1,1);

```

LONG PutBitmapTSPL(LONG iPrintID,INT start_x, INT start_y,CString fileName)

Function

Put downloaded bitmap to print buffer area

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- start_x: coordinate position in horizontal position, in dots(cannot exceed page width)
- start_y: coordinate position in vertical position, in dots(cannot exceed page height)
- fileName: Downloaded bitmap name in printer memory, must be same with bitmap downloaded in printer memory or flash, need to add suffix name and in capital format, otherwise probably cannot print out. DownloadBitmap method downloaded bitmap in printer, then PutBitmap method put bitmap into printer buffer.

Returned value

- True:TSPL_SUCCESS(0)
- Fault:TSPL_IDERROR(-1): ParameteriPrintID got error, return this error
- TSPL_PARAM_LESS_EQUAL_ZERO(-2): Parameterstart_x < 0 or start_y < 0 or

fileName== NULL

Description

- TL21 model does not support this method for now, TL51 supported.

Invoke example

```

LONG ret =0;
CString info;

```

```

ret = ClearBuffTSPL(m_hPrinter);
if(ret < 0)

```

```

{
    info.Format("ClearBuffTSPLParameter_m_hPrinter error:%d",ret);
    MessageBox(info);
}
ret = PageSetupTSPL(m_hPrinter,100,225);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("PageSetupTSPLParameter_m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is less or equal to 0 ,
error code:%d",ret);
        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("PageSetupTSPLParameterPageWidth or PageHeight is more than or equal to
specified scope, error code:%d",ret);
        MessageBox(info);
        break;
}

ret = DownLoadBitMapTSPL(m_hPrinter,FALSE,"D://LOG.bmp");
switch(ret)
{

    case TSPL_IDERROR:
        info.Format("DownLoadBitMapTSPLParameter_m_hPrinter got error or failed to open
file:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("DownLoadBitMapTSPLParameterPathName==NULL:%d",ret);
        MessageBox(info);
        break;
}

ret = PutBitMapTSPL(m_hPrinter,10,10,"D://LOG.BMP");
switch(ret)
{

    case TSPL_IDERROR:

```



```

        info.Format("PutBitMapTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("PutBitMapTSPLParameter fileName==NULL:%d",ret);
        MessageBox(info);
        break;
}

PrintTSPL(m_hPrinter,1,1);

```

LONG SetCharsetNameTSPL(LONG iPrintID,CString CharSetName)

Function

Set international Characters Set

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- CharSetName: Optional character range is among below string:
"U.S.A","France","Germany","U.K","Denmark","Sweden","Italy","Spain","Japan","Norway","Denmark","Spain","Latin","Korea","Slovenia","China"

Returned value

- True: TSPL_SUCCESS(0)
- Fault: TSPL_IDERROR(-1): when Parameter iPrintID got error, return this error.
TSPL_PARAM_LESS_EQUAL_ZERO(-2): when CharSetNameParameter is NULL, return this error
TSPL_PARAM_GREAT_RANGE(-3): when CharSetNameParameter is not at above range, return this error

Description

- TL21 model does not support this method, TL51 supported.

Invoke example

```

LONG ret =0;
CString info;

```

```

ret = SetCharsetNameTSPL(m_hPrinter,"U.S.A");
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("SetCharsetNameTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;
    case TSPL_PARAM_LESS_EQUAL_ZERO:
        info.Format("SetCharsetNameTSPL when CharSetNameParameter is NULL, return this error, error code:%d",ret);

```

```

        MessageBox(info);
        break;

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("SetCharsetNameTSPL when CharsetNameParameter is not at
above range, return this error, error code:%d",ret);
        MessageBox(info);
        break;

}

```

LONG SelectCodePageTSPL(LONG iPrintID,INT value)

Function

Choose characters codepage

Parameter

- iPrintID: Printer handle, is confirmed by returned value of opened port
- value: Choose the NO.n page from below character codes pages.

N	代码页	Code Page
0	CP437 [美国, 欧洲标准]	CP437 [U.S.A., Standard Europe]
1	KataKana [片假名]	Katakana
2	PC850 [多语言]	PC850 [Multilingual]
3	PC860 [葡萄牙]	PC860 [Portuguese]
4	PC863 [加拿大-法语]	PC863 [Canadian-French]
5	PC865 [北欧]	PC865 [Nordic]
6	WCP1251 [斯拉夫语]	WCP1251 [Cyrillic]
7	CP866 斯拉夫2	CP866 Cyrillic #2
8	MIK[斯拉夫/保加利亚]	MIK[Cyrillic /Bulgarian]
9	CP755 [东欧, 拉脱维亚 2]	CP755 [East Europe, Latvian 2]
10	[伊朗, 波斯]	Iran
11	保留	reserve
12	保留	reserve
13	保留	reserve
14	保留	reserve
15	CP862 [希伯来]	CP862 [Hebrew]
16	WCP1252 [拉丁语 1]	WCP1252 Latin I
17	WCP1253 [希腊]	WCP1253 [Greek]
18	CP852 [拉丁语 2]	CP852 [Latina 2]
19	CP858 [多种语言拉丁语 1+欧元符]	CP858 Multilingual Latin I +Euro)
20	伊朗 II[波斯语]	Iran II
21	拉脱维亚	Latvian
22	CP864 [阿拉伯语]	CP864 [Arabic]

23	ISO-8859-1 [西欧]	ISO-8859-1 [West Europe]
24	CP737 [希腊]	CP737 [Greek]
25	WCP1257 [波罗的海]	WCP1257 [Baltic]
26	[泰文1]	Thai 1
27	CP720[阿拉伯语]	CP720[Arabic]
28	CP855	CP855
29	CP857[土耳其语]	CP857[Turkish]
30	WCP1250[中欧]	WCP1250[Central Eurpoe]
31	CP775	CP775
32	WCP1254[土耳其语]	WCP1254[Turkish]
33	WCP1255[希伯来语]	WCP1255[Hebrew]
34	WCP1256[阿拉伯语]	WCP1256[Arabic]
35	WCP1258[越南语]	WCP1258[Vietnam]
36	ISO-8859-2[拉丁语2]	ISO-8859-2[Latin 2]
37	ISO-8859-3[拉丁语3]	ISO-8859-3[Latin 3]
38	ISO-8859-4[波罗的语]	ISO-8859-4[Baltic]
39	ISO-8859-5[斯拉夫语]	ISO-8859-5[Cyrillic]
40	ISO-8859-6[阿拉伯语]	ISO-8859-6[Arabic]
41	ISO-8859-7[希腊语]	ISO-8859-7[Greek]
42	ISO-8859-8[希伯来语]	ISO-8859-8[Hebrew]
43	ISO-8859-9[土耳其语]	ISO-8859-9[Turkish]
44	ISO-8859-15[拉丁语9]	ISO-8859-15 [Latin 3]
45	[泰文2]	Thai2
46	CP856	CP856

Returned value

- True: TSPL_SUCCESS(0)
- Fault: TSPL_IDERROR(-1): when ParameteriPrintID got error, return this error.
TSPL_PARAM_GREAT_RANGE(-3): when valueParameter is not at above range, return this error

Description

- TL51 model does not support this method now, TL21 supported

Invoke example

```

LONG ret =0;
CString info;
SetIs21();//TL21 need to use this method
ret = SelectCodePageTSPL(m_hPrinter,21);
switch(ret)
{
    case TSPL_IDERROR:
        info.Format("SelectCodePageTSPLParameter m_hPrinter error:%d",ret);
        MessageBox(info);
        break;

```



```

    case TSPL_PARAM_GREAT_RANGE:
        info.Format("SelectCodePageTSPL when CharSetNameParameter is not at
above range, return this error, error code:%d",ret);
        MessageBox(info);
        break;
}

```

Appendix

1. CODE128 summarize

CODE128 can use character set A、character set B and character set C interchangeably, and can encode 128 characters of ASCII, 100 numbers 00~99 and some special characters. The code character of each character set are below:

- Character set A: ASCII characters 00H~5FH
- Character set B: ASCII characters 20H~7FH
- Character set C: 100 numbers 00~99

CODE128 can encode below special character

- SHIFT Character

“SHIFT” can convert the first character after SHIFT character from character set A to character set B or from character set B to character set A. From the second character, it will return to the character set which is used by SHIFT previously. “SHIFT” characters can only be used alternatively between character set A and character set B. It can't make the encoding character enter or exit the status of character set C.

- Character set chosen character (CODE A、CODE B、CODE C)

These characters can convert the rear code characters to character A、B or C

- Function character (FNC1、FNC2、FNC3、FNC4)

The use of functional characters depends on the application software. In character set C, only FNC1 is available.

2. Character set

Character set A

chara cter	send data		charac ter	send data		charac ter	send data	
	Hex	Decimal		Hex	Decimal		Hex	Decimal
NULL	00	0	&	26	38	L	4C	76
SOH	01	1	'	27	39	M	4D	77
STX	02	2	(28	40	N	4E	78
ETX	03	3)	29	41	O	4F	49
EOT	04	4	*	2A	42	P	50	80
ENQ	05	5	+	2B	43	Q	51	81
ACK	06	6	,	2C	44	R	52	82
BEL	07	7	-	2D	45	S	53	83
BS	08	8	.	2E	46	T	54	84
HT	09	9	/	2F	47	U	55	85
LF	0A	10	0	30	48	V	56	86
VT	0B	11	1	31	49	W	57	87
FF	0C	12	2	32	50	X	58	88
CR	0D	13	3	33	51	Y	59	89
SO	0E	14	4	34	52	Z	5A	90
SI	0F	15	5	35	53	[5B	91
DLE	10	16	6	36	54	\	5C	92
DC1	11	17	7	37	55]	5D	93
DC2	12	18	8	38	56	^	5E	94
DC3	13	19	9	39	57	_	5F	95
DC4	14	20	:	3A	58	FNC1	7B,31	123,49
NAK	15	21	;	3B	59	FNC2	7B,32	123,50
SYN	16	22	<	3C	60	FNC3	7B,33	123,51
ETB	17	23	=	3D	61	FNC4	7B,34	123,52
CAN	18	24	>	3E	62	SHIFT	7B,53	123,83
EM	19	25	?	3F	63	CODEB	7B,42	123,66
SUB	1A	26	@	40	64	CODEC	7B,43	123,67
ESC	1B	27	A	41	65			
FS	1C	28	B	42	66			
GS	1D	29	C	43	67			
RS	1E	30	D	44	68			
US	1F	31	E	45	69			
SP	20	32	F	46	70			
!	21	33	G	47	71			
"	22	34	H	48	72			
#	23	35	I	49	73			
\$	24	36	J	4A	74			
%	25	37	K	4B	75			

Character set B

chara cter	send data		charac ter	send data		charac ter	send data	
	Hex	Decimal		Hex	Decimal		Hex	Decimal

SP	20	32	F	46	70	l	6C	108
!	21	33	G	47	71	m	6D	109
"	22	34	H	48	72	n	6E	110
#	23	35	I	49	73	o	6F	111
\$	24	36	J	4A	74	p	70	112
%	25	37	K	4B	75	q	71	113
&	26	38	L	4C	76	r	72	114
'	27	39	M	4D	77	s	73	115
(28	40	N	4E	78	t	74	116
)	29	41	O	4F	79	u	75	117
*	2A	42	P	50	80	v	76	118
+	2B	43	Q	51	81	w	77	119
,	2C	44	R	52	82	x	78	120
-	2D	45	S	53	83	y	79	121
.	2E	46	T	54	84	z	7A	122
/	2F	47	U	55	85	{	7B,7B	123,123
0	30	48	V	56	86		7C	124
1	31	49	W	57	87	}	7D	125
2	32	50	X	58	88	—	7E	126
3	33	51	Y	59	89	DEL	7F	127
4	34	52	Z	5A	90	FNC1	7B,31	123,49
5	35	53	[5B	91	FNC2	7B,32	123,50
6	36	54	\	5C	92	FNC3	7B,33	123,51
7	37	55]	5D	93	FNC4	7B,34	123,52
8	38	56	^	5E	94	SHIFT	7B,53	123,83
9	39	57	_	5F	95	CODE	7B,41	123,65
:	3A	58	`	60	96	A	7B,43	123,67
;	3B	59	a	61	97	CODE		
<	3C	60	b	62	98	C		
=	3D	61	c	63	99			
>	3E	62	d	64	100			
?	3F	63	e	65	101			
@	40	64	f	66	102			
A	41	65	g	67	103			
B	42	66	h	68	104			
C	43	67	i	69	105			
D	44	68	j	6A	106			
E	45	69	k	6B	107			

Character set C

character	send data		Character	send data		character	send data	
	Hex	Decimal		Hex	Decimal		Hex	Decimal

0	00	0	38	26	38	76	4C	76
1	01	1	39	27	39	77	4D	77
2	02	2	40	28	40	78	4E	78
3	03	3	41	29	41	79	4F	79
4	04	4	42	2A	42	80	50	80
5	05	5	43	2B	43	81	51	81
6	06	6	44	2C	44	82	52	82
7	07	7	45	2D	45	83	53	83
8	08	8	46	2E	46	84	54	84
9	09	9	47	2F	47	85	55	85
10	0A	10	48	30	48	86	56	86
11	0B	11	49	31	49	87	57	87
12	0C	12	50	32	50	88	58	88
13	0D	13	51	33	51	89	59	89
14	0E	14	52	34	52	90	5A	90
15	0F	15	53	35	53	91	5B	91
16	10	16	54	36	54	92	5C	92
17	11	17	55	37	55	93	5D	93
18	12	18	56	38	56	94	5E	94
19	13	19	57	39	57	95	5F	95
20	14	20	58	3A	58	96	60	96
21	15	21	59	3B	59	97	61	97
22	16	22	60	3C	60	98	62	98
23	17	23	61	3D	61	99	63	99
24	18	24	62	3E	62	FNC1	7B,31	123,49
25	19	25	63	3F	63	CODEA	7B,41	123,65
26	1A	26	64	40	64	CODEB	7B,42	123,66
27	1B	27	65	41	65			
28	1C	28	66	42	66			
29	1D	29	67	43	67			
30	1E	30	68	44	68			
31	1F	31	69	45	69			
32	20	32	70	46	70			
33	21	33	71	47	71			
34	22	34	72	48	72			
35	23	35	73	49	73			
36	24	36	74	4A	74			
37	25	37	75	4B	75			