Printer interface specification

1 Overview

1 CsnPrinterLibs is a DLL written in C ++ on the Windows platform, and the DLL exports C-style functions.

2 When C # uses the SDK, you need to copy the contents of CsnPrinterLibs.cs in the C # print Demo directory. At the same time, you need to copy CsnPrinterLibs.dll to the bin \ Debug or bin \ Release directory, and then refer to Demo to write your own printing application

The program is written in 3 steps to enumerate the port-> open the port-> set the port-> print the required content-> close the port

3 The PrinterLibs function has the following categories

A Port_XXX

The functions beginning with Port mainly open the port, close the port, and enumerate the ports.

Support printing via serial port, parallel port, USB port, network port.

B POS_XXX

The functions beginning with POS are mainly encapsulated with ESC / POS commands, which can control the printer to print.

① Paper feed function can control printer paper feed

② Set series functions can set the printing format, etc.

③Print series functions can print text, barcodes, QR codes, pictures, etc.

④ Query series function can query printer status

⑤ Other functions can control cash drawers, cutters, buzzers, etc.

C PAGE_XXX

The function beginning with PAGE encapsulates the page mode command, which can control the printer in page mode print

① PAGE_PageEnter Enter page mode

② PAGE_SetPrintArea Set page mode print area

③ PAGE_DrawXXX Series functions print in the specified area

④ PAGE_PagePrint Print the entire page

⑤ PAGE_PageExit Exit page mode

Notes：

②③ Can be called repeatedly

Only models that support page mode can use these functions

4 Revision history

Time	Modification

2 Function declaration

Port Function

Port_OpenCOMIO

Open the serial port

Syntax

void * Port_OpenCOMIO(const char *name, unsigned int baudrate = 9600, const int flowcontrol = 0, const int parity = 0, const int databis = 8, const int stopbits = 0);

Parameters

name

Open com name,Can be obtained by EnumCOM

for example: COM1，COM2，COM3，...COM11...

baudrate

Baudrate

Normally choose: 9600,19200,38400,57600,115200.

default 9600

Need to keep the same as printer baudrate, suggest using high baudrate to get better printing speed.

flowcontrol

flowcontrol, The values are defined as follows:

value	define
0	NO
1	DsrDtr
2	CtsRts
3	Xon/Xoff

parity

Parity bit, The values are defined as follows:

value	define
0	no parity
1	odd parity
2	even parity

3 mark parity

4 space parity

databits

databits, value range is [4,8]

stopbits

Parity bit, The values are defined as follows:

value	define
0	1bit stopbits
1	1.5bit stopbits
2	2bit stopbits

Return value

Return handle, If open success, return non-zero value, else return zero.

Remarks

If serial port was occupied, open serial will fail

If baud rate don't match with printer baud rate, it won't print.

Port_OpenUSBIO

Open usb port

Syntax

void * Port_OpenUSBIO(const char *name);

Parameters

name

Open usb name,Can be obtained by EnumUSB

Return value

Return handle, If open success, return non-zero value, else return zero.

Remarks

USB printer connect to computer, if device manager appear "USB Printing Support", then can open USB this function.。

Port_OpenLPTIO

Open parallel port

Syntax

void * OpenLPTIO(const char *name);

Parameters

name

Open parallel name,Can be obtained by EnumUSB

for example：LPT1,LPT2,LPT3...

Return value

Return handle, If open success, return non-zero value, else return zero.

Remarks

Parallel port just support one way communication, just can write, but can't read

All the query status function, are invalid for parallel port.

Port_OpenPRNIO

Open printer port

Syntax

void * OpenPRNIO(const char *name);

Parameters

Name

Printer name

for example: POS58 Printer

Return value

Return handle, If open success, return non-zero value, else return zero.

Remarks

Port_OpenTCPIO

Open Tcp

Syntax

void * Port_OpenTCPIO(const char *ip, const unsigned short port);

Parameters

ip

IP Addres or printer name

For example: 192.168.1.87

port

Port Number

Fixed value: 9100

Return value

Return handle, If open success, return non-zero value, else return zero.

Remarks

PC and printer need in the same network segment, so they can connect

Port_EnumCOM

Enumerate serial port

Syntax

size_t Port_EnumCOM(char *buffer, size_t length);

Parameters

buffer

The buffer to save enumerated port list.

length

The buffer size

Return value

Enumerated port count

Remarks

Port_EnumUSB

Enumerate usb port

Syntax

size_t Port_EnumUSB(char *buffer, size_t length);

Parameters

buffer

The buffer to save enumerated port list

length

The buffer size

Return value

Enumerated port count

Remarks

Port_EnumLPT

Enumerate parallel port

Syntax

size_t Port_EnumLPT(char *buffer, size_t length);

Parameters

buffer

The buffer to save enumerated port list

length

The buffer size

Return value

Enumerated port count

Remarks

Port_EnumPRN

Enumerate printer

Syntax

size_t Port_EnumCOM(char *buffer, size_t length);

Parameters

buffer

The buffer to save enumerated port list

length

The buffer size

Return value

Enumerated port count

Remarks

Port_SetPort

Set the printer communication port

Syntax
bool Port_SetPort(void *handle);

Parameters
handle
Port handle obtained through OpenXXX()

Return value
Returns true on success, false on failure.

Remarks

Port_ClosePort

Close the printer communication port

Syntax
void Port_ClosePort(void *handle);

Parameters
handle
Port handle obtained through OpenXXX()

Return value

Remarks

Pos Function

Pos_Reset

Reset printer

Syntax
bool Pos_Reset();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks

Pos_SelfTest

Print test page

Syntax
bool Pos_SelfTest();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks

Pos_FeedLine

printer feed one Lines

Syntax
bool Pos_FeedLine();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks

Pos_FeedHot

printer feed n dot(0.125mm)

Syntax
bool Pos_FeedHot(int n);

Parameters
n
n dot

Return value
Returns true on successful write, false on failed write.

Remarks

Pos_Feed_N_Line

printer feed n line

Syntax
bool Pos_Feed_N_Line(int n);

Parameters
n
n line.

Return value
Returns true on successful write, false on failed write.

Remarks

Pos_FeedNextLable

Feed the paper to the next label

Syntax
bool Pos_FeedNextLable();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks

The API is only for printers with non-standard label instructions, that is, the label machines that execute ESC/POS instructions.Model: lpm-261

Pos_BlackMark

Feed the paper to the next BlackMark

Syntax
bool Pos_BlackMark();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks
The API is only for printers with non-standard BlackMark instructions

Pos_Align

Set alignment

Syntax
bool Pos_Align(int value);

Parameters
value

print alignment, value are defined as follow:

value	define
0	align left
1	align center
2	align right

Return value
Returns true on successful write, false on failed write.

Remarks
This API is only used to align the instruction operation, and the API that needs to be aligned is used to supplement.

Pos_SetLineHeight

Set the line height

Syntax
bool Pos_SetLineHeight(int value);

Parameters
value
Set the line height(hot size：0.125mm)，value range is[0,255]

Return value
Returns true on successful write, false on failed write.

Remarks

Pos_Text

Print text

Syntax
bool Pos_Text(const wchar_t *prnText, int nLan, int nOrgx, int nWidthTimes, int nHeightTimes, int FontType, int nFontStyle);

Parameters
prnText
Text that needs to be printed
nLan
Type of text encoding for printing, each value is defined as follows:

value	define
0	GBK
1	UTF-8
3	BIG-5
4	SHIFT-JIS
5	EUC-KR

nOrgx
The value of the printed text position is defined as follows:

value	define
-1	align left
-2	align center
-3	align right
>=0	Starting from the n hot

nWidthTimes

Multiple of width magnification，value range is[0,7]

nHeightTimes

Higher magnification，value range is[0,7]

FontType

Type of font to be printed, each value is defined as follows:

value	define
0	12*24
1	9*17

nFontStyle

Type of font to be printed, each value is defined as follows:

value	define
0x00	normal
0x08	bold
0x80	1 hot underline
0x100	2 hot underline
0x200	convert, only efficient in line start
0x400	inverse, white in black
0x1000	every character rotates clockwise 90°

Return value

Returns true on successful write, false on failed write.

Remarks

NFontStyle values can be used with & to allow multiple styles to occur simultaneously.

Pos_Cmd

Command sending

Syntax

```
bool    Pos_Cmd(unsigned char* cmd, int count);
```

Parameters

Cmd

Content of the sent hexadecimal command

count

Number of bytes to send the command

Return value

Returns true on successful write, false on failed write.

Remarks

Pos_Beep

buzzer beeps

Syntax

bool Pos_Beep(unsigned char nCount, unsigned char nMillis);

Parameters

nBeepCount

Beep times

nMillis

Beep time each time = 100 * nBeemMillis ms

Return value

Returns true on successful write, false on failed write.

Remarks

Please confirm whether the printer has a buzzer function.

Pos_KiskOutDrawer

Open cashbox

Syntax

bool Pos_KiskOutDrawer(int nId, int nHightTime = 20, int nLowTime = 60);

Parameters

nId

0 means: pulse was sent to cashbox to output pin 2

1 means: pulse was sent to cashbox to output pin 5

nHightTime

High electric level

nLowTime

Low electric level

Return value

Returns true on successful write, false on failed write.

Remarks

Please confirm whether the printer has the function of opening the money box.

Pos_FullCutPaper

Perform a full cut

Syntax
bool Pos_FullCutPaper();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks
Please confirm whether the printer has full cut function.

Pos_HalfCutPaper

Perform a half cut

Syntax
bool POS_HalfCutPaper();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks
Please confirm whether the printer has half cut function.

Pos_Barcode

Print the barcode

Syntax
bool Pos_Barcode(const char * BarcodeData, int nBarcodeType, int nOrgx, int nUnitWidth, int nUnitHeight, int nFontStyle, int FontPosition);

Parameters
BarcodeData
Printed bar code content

nBarcodeType

Type of barcode printed, each value is defined as follows:

值	类型
0x41	UPC-A
0x42	UPC-E
0x43	EAN13
0x44	EAN8
0x45	CODE39
0x46	ITF
0x47	CODABAR
0x48	CODE93
0x49	CODE128

nOrgx

The value of the printed bar code position is defined as follows:

value	define
-1	align left
-2	align center
-3	align right
>=0	Starting from the n hot

nUnitWidth

Bar code width，value range is[1,6]

nUnitHeight

Bar code height，value range is[1,255]

nFontStyle

The font type of readable character (HRI) is defined as follows:

value	define
0	12*24
1	9*17

FontPosition

The printing positions of readable characters (HRI) are defined as follows:

value	define
0	Doesn’t print
1	Only print in barcode upward
2	Only print in barcode downward
3	Print both barcode upward and downward

Return value

Returns true on successful write, false on failed write.

Remarks

nUnitWidth If the maximum width of the printer is exceeded, it is not printed.

Pos_Qrcode

Print QR code

Syntax

bool Pos_Qrcode(const wchar_t *QrcodeData, int nWidth = 2, int nVersion = 0, int nErrlevenl = 4);

Parameters

QrcodeData

QR code character string

nWidth

Unit width of each module for QR code,value range is[1,6]

Set module width properly can make QR code nicer

nVersion

QR code version size, this value is about QR code size. value range is[0,16]

Set as 0 to calculate QR code version size automatically

Pls set proper value for this value if you want the QR code size to be fixed.

nErrlevenl

Error correction level，value range is[1,4]

Return value

Returns true on successful write, false on failed write.

Remarks

If the printed qr code exceeds the print boundary, it is not printed.

Pos_EscQrcode

Print QR code （ESC/POS）

Syntax

bool Pos_EscQrcode(const wchar_t *QrcodeData, int nWidth = 4, int nErrlevenl = 4);

Parameters

QrcodeData

QR code character string.

nWidth

Unit width of each module for QR code,value range is[1,16]

Set module width properly can make QR code nicer

nErrlevenl

Error correction level，value range is[1,4]

Return value

Returns true on successful write, false on failed write.

Remarks

If the printed qr code exceeds the print boundary, it is not printed.

Pos_DoubleQrcode

Print Double QR code

Syntax

bool Pos_DoubleQrcode(const wchar_t *QrcodeData1,int QR1Position,int QR1Version, int QR1Ecc, const wchar_t *QrcodeData2, int QR2Position, int QR2Version, int QR2Ecc, int ModuleSize);

Parameters

QrcodeData1

QR code 1 character string.

QR1Position

QR code 1 start print Position

QR1Version

QR code 1 version size, this value is about QR code size,范围[0，16]

QR1Ecc

QR code 1 Error correction level，value range is[0,3]

QrcodeData2

QR code 2 character string.

QR2Position

QR code 2 start print Position

QR2Version

QR code 2 version size, this value is about QR code size,范围[0，16]

QR2Ecc

QR code 2 Error correction level，value range is[0,3]

ModuleSize

QrCode size.[1，8]

Return value

Returns true on successful write, false on failed write.

Remarks

If the printing fails

please check whether the location of the second qr code printing overlaps with the first one or exceeds the printing boundary after printing.

Pos_ImagePrint

Print picture

Syntax

bool Pos_ImagePrint(const wchar_t *FileName, int nWidth = 384, int nBinaryAlgorithm = 0);

Parameters

FileName

Image path.

nWidth

Specifies the width (pixels) for the printer to print the image

The max. width doesn’t exceed 384 dots of 2 inches printer(58mm printer)

The max. width doesn’t exceed 576 dots of 3 inches printer(80mm printer)

nBinaryAlgorithm

Two-valued conversion method

value	define
0	uses shake method, which has good efficiency to colorful pictures.
1	uses average threshold value method, which has good efficiency to text pictures.

Return value

Returns true on successful write, false on failed write.

Remarks

Pos_PrintNVLogo

Print NV LOGO

Syntax

bool Pos_PrintNVLogo(unsigned short nLogo, unsigned short nWidth = 0);

Parameters

n

Print the N Logo,value range is[1,9]

nMode

Specify the mode of printing the Logo, and the value is defined as follows:

value	define
0	normal
1	double width
2	double height

3	double width | double height

Return value

Returns true on successful write, false on failed write.

Remarks

If there is no pre-loaded Logo in the printer, it will not print.Please use the printer tool to preload.

Pos_QueryPrinterErr

Select printer error

Syntax

int Pos_QueryPrinterErr(unsigned long nTimeout = 3000);

Parameters

nTimeout

Single time to query status exceeds time.

Return value

Return the error value, and the value is defined as follows:

value	define
1	pirnter good
-1	off-line
-2	not close cover
-3	paper shortage
-4	cut error
-5	hyperpyrexia
-6	failed

Remarks

The API cannot return more than one exception at a time.If you need to get more than one exception,
implement it yourself using the Pos_QueryStstus function.Do not insert query operations during printing.

Pos_QuerySтstus

Query status

Syntax

bool Pos_QuerySтstus(char *rBuffer, int type, unsigned long nTimeout);

Parameters

rBuffer

Stores the state returned by the printer

type

The data table for the query.value range is[1,4],See the documentation for the specific form.

type=1：Printer status

Bit	0/1	Hex.	Dec.	Function
0	0	00	0	Fixed is 0
1	1	02	2	Fixed is 1
2	0	00	0	One or two cashboxes are open(Fixed is 0 if this machine is without cashbox)
	1	04	4	Two cashboxes are closed
3	0	00	0	Online
	1	08	8	Offline
4	1	10	16	Fixed is 1
5, 6		--	--	Undefined
7	0	00	00	Paper has been tore off
	1	80	96	Paper hasn’t been tore off

type=2：Transmit offline status

Bit	0/1	Hex.	Dec.	Function
0	0	00	0	Fixed is 0
1	1	02	2	Fixed is 1
2	0	00	0	Upper cover is closed
	1	04	4	Upper cover is open
3	0	00	0	Paper feed key is un-pressed
	1	08	8	Paper feed key is pressed
4	1	10	16	Fixed is 1
5	0	00	0	Paper is not out
	1	20	32	Paper out
6	0	00	00	No error
	1	40	64	Error

7	0	00	0	Fixed is 0

type=3：Transmit error status

Bit	0/1	Hex.	Dec.	Function
0	0	00	0	Fixed is 0
1	1	02	2	Fixed is 1
2		--	--	Undefined
3	0	00	0	Cutter has no error
	1	08	8	Cutter has error
4	1	10	16	Fixed is 1
5	0	00	0	No recoverable error
	1	20	32	Has recoverable error
6	0	00	00	Printer head temp. and voltage are normal
	1	40	64	Printer head temp. or voltage is out of range
7	0	00	0	Fixed is 0

type=4：Transmit paper sensor status

Bit	0/1	Hex.	Dec	Function
0	0	00	0	Fixed is 0
1	1	02	2	Fixed is 1
2，3	0	00	0	With paper
	1	0C	12	Paper will be out
4	1	10	16	Fixed is 1
5，6	0	00	0	With paper
	1	60	96	Paper out
7	0	00	0	Fixed is 0

nTimeout

Single time to query status exceeds time.

Return value

Returns true on successful write, false on failed write.

Remarks

Pos_SetPrinterBaudrate

Set the printer baud rate

Syntax

bool Pos_SetPrinterBaudrate(int nBaudrate);

Parameters

nBaudrate

Set the printer baud rate.

for example：9600 19200 38400 57600 115200

Return value

Returns true on successful write, false on failed write.

Remarks

If you connect with a serial port, restart OpenCOM after setting the baud rate.

Pos_SetPrinterBasic

Set basic printer parameters

Syntax

bool Pos_SetPrinterBasic(int nFontStyle, int nDensity, int nLine, int nBeep, int nCut);

Parameters

nFontStyle

Set the font specification, each value is defined as follows:

value	fout
0	9*17
1	12*24
2	9*24
3	16*18

nDensity

Set the concentration, each value is defined as follows:

value	fout
0	light
1	normal
2	little Dark
3	Dark

nLine

Set the feeding mode, each value is defined as follows:

value	mode
0	0x0A
1	0x0D

nBeep

Whether the buzzer is enabled, each value is defined as follows:

value	on/off

0	off
1	on

nCut

Whether the cutter is enabled, each value is defined as follows:

value	on/off
0	off
1	on

Return value

Returns true on successful write, false on failed write.

Remarks

Font specification only sets non-double-byte text, while Chinese, Japanese, Korean and other fonts are 24*24 and cannot be modified.

Buzzer switch please confirm whether the model used has buzzer function

Please confirm whether the machine is equipped with cutting function

Page Function

Page_SelectPageMode

Select page mode

Syntax

bool Page_SelectPageMode();

Parameters

Return value

Returns true on successful write, false on failed write.

Remarks

Make sure the printer has page-mode functionality, which does not print directly. You need to call Page_PrintPage after filling in the data.

Page_PrintPage

Print the content in page mode

Syntax
bool Page_PrintPage();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks
Make sure the printer has page-mode functionality, which does not print directly. You need to call Page_PrintPage after filling in the data.

Page_ExitPageMode

Exit page mode

Syntax
bool Page_ExitPageMode();

Parameters

Return value
Returns true on successful write, false on failed write.

Remarks

Page_SetVerticalAbsolutePrintPosition

Set the vertical absolute print position

Syntax
bool Page_SetVerticalAbsolutePrintPosition(unsigned short nPosition);

Parameters
nPosition
Print position

Return value
Returns true on successful write, false on failed write.

Remarks

Page_SetHorizontalAbsolutePrintPosition

Set the horizontal absolute print position

Syntax
bool Page_SetHorizontalAbsolutePrintPosition(unsigned short nPosition);

Parameters
nPosition
Print position

Return value
Returns true on successful write, false on failed write.

Remarks

Page_SetVerticalRelativePrintPosition

Set the vertical relative to the print position

Syntax
bool Page_SetVerticalRelativePrintPosition(unsigned short nPosition);

Parameters
nPosition
Print position

Return value
Returns true on successful write, false on failed write.

Remarks

Page_SetHorizontalRelativePrintPosition

Set the horizontal relative print position

Syntax

bool Page_SetHorizontalRelativePrintPosition(unsigned short nPosition);

Parameters

nPosition

Print position

Return value

Returns true on successful write, false on failed write.

Remarks

Page_SetPageModeDrawDirection

Set the printing direction in page mode

Syntax

bool Page_SetPageModeDrawDirection(unsigned short nPosition);

Parameters

nDirection

Print the direction of the region, and each value is defined as follows:

0 from left to right

1 from the bottom up

2 from right to left

3 from top to bottom

Return value

Returns true on successful write, false on failed write.

Remarks

Page_SetPageArea

Set the page area in page mode

Syntax

bool Page_SetPageArea(unsigned short x, unsigned short y, unsigned short w, unsigned short h);

Parameters

x

Transverse starting position

y

Longitudinal starting position

w

Print width

h

Print height

Return value

Returns true on successful write, false on failed write.

Remarks

Page_Text

Print text in page mode

Syntax

bool Page_Text(const wchar_t *prnText, int nLan, int nOrgx, int nWidthTimes, int nHeightTimes, int FontType, int nFontStyle);

Parameters

prnText

Text that needs to be printed

nLan

Type of text encoding for printing, each value is defined as f

value	define
0	GBK
1	UTF-8
3	BIG-5
4	SHIFT-JIS
5	EUC-KR

nOrgx

The value of the printed text position is defined as follows:

value	define

-1	align left
-2	align center
-3	align right
>=0	Starting from the n hot

nWidthTimes

Multiple of width magnification，value range is[0,7]

nHeightTimes

Higher magnification，value range is[0,7]

FontType

Type of font to be printed, each value is defined as follows:

value	define
0	12*24
1	9*17

nFontStyle

Type of font to be printed, each value is defined as follows:

value	define
0x00	normal
0x08	bold
0x80	1 hot underline
0x100	2 hot underline
0x200	convert, only efficient in line start
0x400	inverse, white in black
0x1000	every character rotates clockwise 90°

Return value

Returns true on successful write, false on failed write.

Remark

nFontStyle values can be used with & to allow multiple styles to occur simultaneously.

Page_Barcode

Print the barcode in page mode

Syntax

bool Page_Barcode(const char * BarcodeData, int nBarcodeType, int nOrgx, int nUnitWidth, int nUnitHeight, int nFontStyle, int FontPosition);

Parameters

BarcodeData

Printed bar code content

nBarcodeType

Type of barcode printed, each value is defined as follows:

value	define
0x41	UPC-A
0x42	UPC-E
0x43	EAN13
0x44	EAN8
0x45	CODE39
0x46	ITF
0x47	CODABAR
0x48	CODE93

nOrgx

The value of the printed bar code position is defined as follows:

value	define
-1	align left
-2	align center
-3	align right
>=0	Starting from the n hot

nUnitWidth

Bar code width，value range is[1,6]

nUnitHeight

Bar code height，value range is[1,255]

nFontStyle

The font type of readable character (HRI) is defined as follows:

value	define
0	12*24
1	9*17

FontPosition

The printing positions of readable characters (HRI) are defined as follows:

value	define
0	Doesn’t print
1	Only print in barcode upward
2	Only print in barcode downward
3	Print both barcode upward and downward

Return value

Returns true on successful write, false on failed write.

Remarks

nUnitWidth If the maximum width of the printer is exceeded, it is not printed.

Page_Qrcode

Print QR code in page mode

Syntax

bool Page_Qrcode(const wchar_t *QrcodeData, int nWidth = 4, int nErrlevenl = 4);

Parameters

QrcodeData

QR code character string.

nWidth

Unit width of each module for QR code,value range is[1,16]

Set module width properly can make QR code nicer

nErrlevenl

Error correction level，value range is[1,4]

Return value

Returns true on successful write, false on failed write.

Remarks

If the printed qr code exceeds the print boundary, it is not printed.

Page_ImagePrint

Print picture in page mode

Syntax

bool Page_ImagePrint(const wchar_t *FileName, int nWidth = 384, int nBinaryAlgorithm = 0);

Parameters

FileName

Image path.

nWidth

Specifies the width (pixels) for the printer to print the image

The max. width doesn’t exceed 384 dots of 2 inches printer(58mm printer)

The max. width doesn’t exceed 576 dots of 3 inches printer(80mm printer)

nBinaryAlgorithm

Two-valued conversion method

value	define
0	uses shake method, which has good efficiency to colorful pictures.
1	uses average threshold value method, which has good efficiency to text pictures.

Return value

Returns true on successful write, false on failed write.