

Windows SDK

User Manual_(V1.5)



1. **openport(name)** 、 **openport_USB()** 、 **openports_USB(PrinterModel)** 、 **openport_Ethernet(IP, port)**

- Description: Specify and open the output port on the computer side

- Parameter:

➔ name: String type

(1) When printing stand-alone, specify the printer driver name

(2) If you are connected to a shared network printer, specify the path and the model name of the printer

e.g.: \\192.168.1.104\ Printer Model

➔ PrinterModel: String type · Printer model name

➔ IP: String type · IP address of the Printer

➔ port: int type · port number of the Printer

2. **closeport()** 、 **closeport_USB()** 、 **closeport_Ethernet()**

- Description: Close the output port

- Parameter: None

3. **detectUSB_USB()**

- Description: Detect the plugging and unplugging status of the printer's USB interface

- Parameter: None

- Return: String array type · All turned on and connected printers via USB.

4. **detectUSBStr_USB()**

- Description: Detect the plugging and unplugging status of the printer's USB interface
- Parameter: None
- Return: String type · All turned on and connected printers via USB.

5. **sendcommand(command) 、 sendcommand_USB(command) 、 sendcommand_Ethernet(command)**

- Description: Sent built-in commands to the printer. (At the end, \r\n will be automatically added to the end of the command)
- Parameter:
 - ➔ command: string type, Refer to TSPL programming manual commands for details.

6. **sendcommand_Array(command) 、 sendcommand_Array_USB(command) 、**

sendcommand_Array_Ethernet(command)

- Description: Sent built-in commands to the printer. (At the end, \r\n will be automatically added to the end of the command)
- Parameter :
 - ➔ command : byte array type, Refer to TSPL programming manual commands for details.

7. **sendcommand_noClRf(command) 、 sendcommand_noClRf_USB(command) 、**

sendcommand_noClRf_Ethernet(command)

- Description: Sent built-in commands to the printer. (At the end, \r\n will not be automatically added to the end of the command)
- Parameter :
 - ➔ command : string type, Refer to TSPL programming manual commands for details.

8. `clearbuffer()` 、 `clearbuffer_USB()` 、 `clearbuffer_Ethernet()`

- Description: Clear the image buffer.
- Parameter: None

9. `setup(width, height, speed, density, sensor, sensorDistance, sensorOffset)` 、

`setup_USB(width, height, speed, density, sensor, sensorDistance, sensorOffset)`

`setup_Ethernet(width, height, speed, density, sensor, sensorDistance, sensorOffset)`

- Description: Setup label width , label height, print speed, density, sensor type, gap/blackmark vertical distance and gap/blackmark offset distance.
- Parameter:

Parameter	Type	Description
width	string	Set up label width · unit : mm
height	string	Set up label height · unit: mm
speed	string	Defines the print speed (1~15: print speed at 1"~15"/sec) Maximum print speed depends on printer models, and selectalbe maximum speed is 15"/sec.
density	string	Set up print density (0~15); If the number is bigger, the printout will be darker.
sensor	string	Set up sensor type ; 0 : gap sensor 1 : black mark sensor
sensorDistance	string	Set up vertical gap height of the gap or black mark ° Unit: mm
sensorOffset	string	Set up offset distance of the gap or black mark, unit: mm, this parameter is set to 0 when the general label is used.

10. barcode(x, y, type, height, readable, rotation, narrow, wide, content) `

barcode_USB(x, y, type, height, readable, rotation, narrow, wide, content)

barcode_Ethernet(x, y, type, height, readable, rotation, narrow, wide, content)

■ Description: Use built-in barcode formats to print barcodes.

■ Parameter:

Parameter	Type	Description
x	string	Specify the x-coordinate bar code on the label, Unit: dot
y	string	Specify the y-coordinate bar code on the label, Unit: dot
type	string	Set up Code Type · refer to Appendix
height	string	Set up bar code height (in dots)
readable	string	Set up whether to print human recognizable interpretation (text) or not. 0: Do not print 1: Print barcode document left 2: Print barcode code document in the center 3: Print barcode document right
rotation	string	Set up barcode rotation 0 : No rotation 90 : Rotate 90 degrees clockwise 180 : Rotate 180 degrees clockwise 270 : Rotate 270 degrees clockwise
narrow	string	Set up narrow bar ratio (in dots), refer to Appendix
wide	string	Set up wide bar ratio (in dots), refer to Appendix
content	string	Content of barcode. Please note that the maximum number of digits of bar code content. refer to Appendix

11. `qrcode(x, y, ECCLevel, cellWidth, mode, rotation, content)` 、

`qrcode_USB(x, y, ECCLevel, cellWidth, mode, rotation, content)` 、

`qrcode_Ethernet(x, y, ECCLevel, cellWidth, mode, rotation, content)`

■ Description: Use built-in QR code formats to print QR code

■ Parameter:

Parameter	Type	Description
x	string	The upper left corner x-coordinate of the QR code, represented by a dot
y	string	The upper left corner y-coordinate of the QR code, represented by dot
ECCLevel	string	Error correction recovery level L : 7% M : 15% Q : 25% H : 30%
cellWidth	string	QR code shape width 1~10
mode	string	Set QR code mode : A : Auto M : Manual
rotation	string	Set the QRCode rotation 0 : 0 degrees 90 : 90 degrees 180 : 180 degrees 270 : 270 degrees
content	string	The encodable character set is described as below, *Encodable character set: 1) Numeric data: (digits 0~9) 2) Alphanumeric data Digits 0-9 Upper case letters A-Z Nine other characters: space, \$ % * + - . / :) 3) 8-bit byte data JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201 4) Kanji characters Shift JIS values 8140HEX –9FFCHEX and E040HEX –EAA4 HEX. These are values shifted from those of JIS X 0208. Refer to JIS X 0208 Annex 1 Shift Coded Representation for detail.

12. `printerfont(x, y, size, rotation, x_scale, y_scale, content)` 、

`printerfont_USB(x, y, size, rotation, x_scale, y_scale, content)`

`printerfont_Ethernet(x, y, size, rotation, x_scale, y_scale, content)`

■ Description: Use printer built-in fonts to print

■ Parameter:

Parameter	Type	Description
x	string	The x-coordinate of the text
y	string	The y-coordinate of the text
size	string	Built-in font type 1: 8*/12 dots 2: 12*20 dots 3: 16*24 dots 4: 24*32 dots 5: 32*48 dots TST24.BF2: Traditional Chinese 24*24 TST16.BF2: Traditional Chinese 16*16 TSS24.BF2: Simplified Chinese 24*24 TSS16.BF2: Simplified Chinese 16*16
rotation	string	The rotation angle of text 0 : No rotation 90: degrees, in clockwise direction 180 : degrees, in clockwise direction 270 : degrees, in clockwise direction
x_scale	string	Horizontal multiplication, Available factors: 1~10
y_scale	string	Vertical multiplication, Available factors: 1~10
content	string	Content of text string

13. `formfeed()` 、 `formfeed_USB()` 、 `formfeed_Ethernet()`

■ Description: Feed label to the top of next label.

This function should be used after setup function.

■ Parameter: None

14. nobackfeed() 、 nobackfeed_USB() 、 nobackfeed_Ethernet()

- Description: Set the paper not to back feed
- Parameter: None

15. printlabel (set, copy) 、 printlabel _USB(set, copy) 、 printlabel _Ethernet(set, copy)

- Description: Print the label format currently stored in the image buffer.
- Parameter:
 - ➔ set: string type, Specifies how many sets of labels will be printed.
 $1 \leq \text{set} \leq 999999999$.
 - ➔ copy: string type, Specifies how many copies should be printed for each particular label set.
 $1 \leq \text{copy} \leq 999999999$

16. downloadpcx (filename,memoryname) 、 downloadpcx_USB (filename, memoryname)

downloadpcx_Ethernet(filename,memoryname)

- Description: Download the monochrome PCX file to the main board flash memory.
- Parameter:
 - ➔ filename:String type, PCX file name.(File path could be included.)
 - ➔ memoryname: The file name in the printer's memory (Use capital letters in file name).

17. downloadbmp (filename, memoryname) 、 downloadbmp_USB (filename, memoryname)

downloadbmp_Ethernet (filename, memoryname)

- Description: Download the monochrome BMP file to the main board flash memory.
- Parameter:
 - ➔ filename:String type, BMP file name. (File path could be included.)
 - ➔ memoryname: The file name in the printer's memory (Use capital letters in file name).

18. `windowsfont (x,y,height,rotation,fontstyle,underline,fontname,content)`

`windowsfont_USB (x,y,height,rotation,fontstyle,underline,fontname,content)`

`windowsfont_Ethernet(x,y,height,rotation,fontstyle,underline,fontname,content)`

■ Description: Print text use the Windows TTF font.

■ Parameter:

Parameter	Type	Description
x	int	The x-coordinate of the text (dot).
y	int	The y-coordinate of the text (dot).
height	int	Font height (dot)
rotation	int	The rotation angle of text 0 : No rotation 90: degrees, in clockwise direction 180 : degrees, in clockwise direction 270 : degrees, in clockwise direction
fontstyle	int	Set the font style 0: Normal 1:Italic 2:Bold 3: Boldand Italic
underline	int	Set the underline 0: No udnerline 1: Add the underline
fontname	string	Windows font names, such as: Arial, Times new Roman, MingLiU, and DFKai-sb
content	string	Set the text content you want to print.

19. `getDLLVersion (returnWay)` 、 `getDLLVersion_USB(returnWay)` 、 `getDLLVersion_Ethernet(returnWay)`

■ Description: Return the SDK version.

■ Parameter:

➔ RetrunWay:Int type, inputting parameter 0 will both jump out the SDK version message and return the SDK Version(stirng).

20. printerstatus_USB () 、 printerstatus_Ethernet()

- Description: Obatin the printer status.
- Parameter: None
- Returns string :

Return message	The status of the printer
00	Normal
01	Head opened
02	Paper Jam
03	Paper Jam and head opened
04	Out of paper
05	Out of paper and head opened
08	Out of ribbon
09	Out of ribbon and head opened
0A	Out of ribbon and paper jam
0B	Out of ribbon, paper jam and head opened
0C	Out of ribbon and out of paper
0D	Out of ribbon, out of paper and head opened
10	Pause
20	Printing
80	Other error

21. writeUHF (dataFormat,startBlockNo,byteSize,Gen2MemoryBank,datastring)

writeUHF_USB (dataFormat,startBlockNo,byteSize,Gen2MemoryBank,datastring)

writeUHF_Ethernet (dataFormat,startBlockNo,byteSize,Gen2MemoryBank,datastring)

- Description: Write data to UHF tag memory.
- Parameter:

Parameter	Type	Description
dataFormat	string	Define data format · default is“H” A : ASCII H : Hexadecimal
startBlockNo	int	Secify the 16-bit starting block number · Default : 2
byteSize	int	Values: 1 to n, where n is the maximum number of bytes for the tag. Default: 1

Gen2MemoryBank	string	Select Gen2 memory bank R : Reserved E : EPC (Default) T : TID(Tag ID) U : User
datastring	string	Data string.

22. EPCPWD_Action (action, password) 、 EPCPWD_Action_USB (action, password) 、

EPCPWD_Action_Ethernet (action, password)

- Description: Lock or unlock EPC memory with password for UHF GEN2 tag.
- Parameter:

Parameter	Type	Description
action	string	Action type U : unlock EPC memory bank L : lock EPC memory bank O : permanently unlock EPC memory bank P : permanently lock EPC memory bank
password	string	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

23. TIDPWD_Action(action, password) 、 TIDPWD_Action_USB (action, password) 、

TIDPWD_Action_Ethernet(action, password)

- Description: Lock or unlock TID memory with password for UHF GEN2 tag.
- Parameter:

Parameter	Type	Description
action	string	Action type U : unlock TID memory bank L : lock TID memory bank O : permanently unlock TID memory bank P : permanently lock TID memory bank
password	string	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

24. USERPWD_Action (action, password) 、 USERPWD_Action_USB (action, password) 、

USERPWD_Action_Ethernet (action, password)

- Description: Lock or unlock USER memory with password for UHF GEN2 tag.
- Parameter:

Parameter	Type	Description
action	string	Action type U : unlock USER memory bank L : lock USER memory bank O : permanently unlock USER memory bank P : permanently lock USER memory bank
password	string	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

25. AccessPWD_Action (action, password) 、 AccessPWD_Action_USB (action, password) 、

AccessPWD_Action_Ethernet (action, password)

- Description: Lock or unlock access password with password for UHF GEN2 tag.
- Parameter:

Parameter	Type	Description
action	string	Action type U : unlock the access password* L : lock the access password* O : permanently unlock the access password P : permanently lock the access password S : Set Password
password	string	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

26. KillPWD_Action (action, password) 、 KillPWD_Action_USB (action, password) 、

KillPWD_Action_Ethernet (action, password)

- Description: Lock or unlock kill password with password for UHF GEN2 tag.
- Parameter:

Parameter	Type	Description
action	string	Action type U : unlock the kill password* L : lock the kill password* O : permanently unlock the kill password P : permanently lock the kill password S : Set Password
password	string	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

27. Set_RFIDPorcedure (tagType, rw_position, void_printout, tryEncodie_times,error_handle, speed, retry_times)

Set_RFIDPorcedure_USB (tagType, rw_position, void_printout, tryEncodie_times,error_handle, speed, retry_times)

Set_RFIDPorcedure_Ethernet (tagType, rw_position, void_printout, tryEncodie_times,error_handle, speed, retry_times)

- Description: Set RFID procedure
- Parameter:

Parameter	Type	Description
tagType	int	Set Tag type · accepted value:1~10 · For UHF: 1 = ISO 18000 6C/Class 1 Gen2 (Q command) 8 = ISO 18000 6C/Class 1 Gen 2 (default) For HF 10 = UHF-J
rw_position	int	Move the media to the specified position on the label, measured in dot rows from the label top, before encoding Accept value: 0~9999(dot) · default is 0

void_printout	int	Set the length of the void printout in vertical (Y axis) dot rows. Accepted values: 0 to label length Default: label length
tryEncode_times	int	The number of labels that will be attempted in case of read/encode failure. Accepted values: 1 to 10 Default: 3
error_handle	string	If an error persists after the specified number of labels are tried, perform this error handling action. N : No action (Default) P : Pause mode E : Error mode
speed	int	If a label is voided, the speed at which "VOID" will be printed across the label. Accepted value: 2~10(IPS), Default is 2.
retry_times	int	The retry times of a tag that will be attempted in case of read/encode failure. Accepted value: 0~10 · Default is 6

28. Set_RFIDPorcedure_mm (tagType, rw_position, void_printout, tryEncode_times,error_handle, speed, retry_times,dpi)

Set_RFIDPorcedure_mm_USB (tagType, rw_position, void_printout, tryEncode_times,error_handle, speed, retry_times,dpi)

Set_RFIDPorcedure_mm_Ethernet (tagType, rw_position, void_printout, tryEncode_times,error_handle, speed, retry_times,dpi)

- Description: Set RFID procedure
- Parameter:

Parameter	Type	Description
tagType	int	Set Tag type · accepted value: 1~10 · For UHF: 1 = ISO 18000 6C/Class 1 Gen2 (Q command) 8 = ISO 18000 6C/Class 1 Gen 2 (default) For HF 10 = UHF-J
rw_position	double	Move the media to the specified position on the label, measured in dot rows from the label top, before encoding

		Accept value: 203dpi:0 ~ 1251 (mm) 300dpi:0 ~ 846 (mm) 600dpi:0 ~ 423 (mm) · default is 0
void_printout	double	Set the length of the void printout in vertical (Y axis) mm rows. Accepted values: 0 to label length Default: label length
tryEncodie_times	int	The number of labels that will be attempted in case of read/encode failure. Accepted values: 1 to 10 Default: 3
error_handle	String	If an error persists after the specified number of labels are tried, perform this error handling action. N : No action (Default) P : Pause mode E : Error mode
speed	int	If a label is voided, the speed at which "VOID" will be printed across the label. Accepted value: 2~10(IPS), Default is 2.
retry_times	int	The retry times of a tag that will be attempted in case of read/encode failure. Accepted value:0~10 · Default is 6
dpi	String	Set DPI of the printer 203: 203 dpi 300: 300 dpi 600: 600 dpi

29. writeHF (dataFormat,startBlockNo,byteSize,datastring)

writeHF_USB (dataFormat,startBlockNo,byteSize,datastring)

writeHF_Ethernet (dataFormat,startBlockNo,byteSize,datastring)

- Description: Write data to HF tag memory.
- Parameter:

Parameter	Type	Description
dataFormat	string	Define data format · default is "H" A : ASCII

		H : Hexadecimal
startBlockNo	int	Specify the 16-bit starting block number · Default : 2
byteSize	int	Values: 1 to n, where n is the maximum number of bytes for the tag. Default: 1
datastring	string	Data string

30. **printerfontblock** (x, y, width, height, fontname, rotation, x_scale, y_scale, space, align, content) 、

printerfontblock_USB(x, y, width, height, fontname, rotation, x_scale, y_scale, space, align,

content) 、

printerfontblock_Ethernet(x, y, width, height, fontname, rotation, x_scale, y_scale, space, align,

content)

■ Description: Use printer built-in fonts to print paragraph.

■ Parameter:

Parameter	Type	Description
x	string	The x-coordinate of the text (in dots)
y	string	The y-coordinate of the text (in dots)
width	string	The width of block for the paragraph in dots
height	string	The height of block for the paragraph in dots
fontname	string	Built-in font type 1: 8*12 dots 2: 12*20 dots 3: 16*24 dots 4: 24*32 dots 5: 32*48 dots TST24.BF2: Traditional Chinese 24*24 TST16.BF2: Traditional Chinese 16*16 TSS24.BF2: Simplified Chinese 24*24 TSS16.BF2: Simplified Chinese 16*16
rotation	string	The rotation angle of text 0 : No rotation 90 : degrees, in clockwise direction 180 : degrees, in clockwise direction 270 : degrees, in clockwise direction
x_scale	string	Horizontal multiplication, Available factors: 1~10

y_scale	string	Vertical multiplication, Available factors: 1~10
space	string	Add or delete the space between lines (in dots)
align	string	Text alignment 0 : default (Left) 1 : Left 2 : Center 3 : Right
content	string	Data in block. The maximum data length is 4092 bytes.

31. readUHF_USB (dataFormat,startBlockNo,byteSize,Gen2MemoryBank)

readUHF_Ethernet (dataFormat,startBlockNo,byteSize,Gen2MemoryBank)

- Description: Read data from UHF tag memory (R command)

- Parameter:

Parameter	Type	Description
dataFormat	string	Set up callback returned data format · A : ASCII H : Hexadecimal (Default)
startBlockNo	int	Secify the 16-bit starting block number to read · Default is 0
byteSize	int	Secify the data lengths to read · default is 1
Gen2MemoryBank	string	Gen2 memory bank · R = Reserved E = EPC T = TID U = UESR Default : E

- Return Value(EPC Data):

DataFormat	Return string(example)
A	Data format is ASCII (ex: 24051324000103456400)
H	Data format is Hexadecimal (ex: 3234303531333234303030313033343536343030)

- Return Value (Error code) :

Code Type	Description
640000000000000000000000	Other errors
650000000000000000000000	Mor than mermory space

66000000000000000000000000000000	The memory is locked
67000000000000000000000000000000	Insufficient reading power
68000000000000000000000000000000	Unpecific error
69000000000000000000000000000000	CRC error
6A000000000000000000000000000000	If an error occurs during writing · how many words have been written in the reply.
6B000000000000000000000000000000	If the tag replies incorrectly during writing · besides the error code, it would also replies how many words have been written.
6C000000000000000000000000000000	no tag exists
6D000000000000000000000000000000	command format error
6E000000000000000000000000000000	Failed to set power strength
6F000000000000000000000000000000	Failed to set regulations

32. setPWD_Action(passwordArea, action, NewPassword, WritePassword)

setPWD_Action_USB(passwordArea, action, NewPassword, WritePassword)

setPWD_Action_Ethernet(passwordArea, action, NewPassword, WritePassword)

■ Description: Set Write/Read/Status/Kill Password to UHF GJB tag.

■ Parameter:

Parameter	Type	Description
passwordArea	string	Set password area K=Kill, W=Write (Default), R=Read, S=Status
action	string	S=Set Password
NewPassword	string	New password for password area setting above · 8 HEX characters. (0~9, A,B,C,D,E,F)
WritePassword	string	Writing password · 8 HEX characters. (0~9, A,B,C,D,E,F)

33. **writeGJB_UHF**(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, datastring, WritePassword)
writeGJB_UHF_USB(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, datastring, WritePassword)
writeGJB_UHF_Ethernet(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, datastring, WritePassword)

■ Description: Write data to UHF GJB tag memory

■ Parameter:

Parameter	Type	Description
dataFormat	string	Define data format · default is "H" A : ASCII H : Hexadecimal
startBlockNo	int	Specify the 16-bit starting block number · Default : 1
byteSize	int	Values: 1 to n, where n is the maximum number of bytes for the tag. Default: 1
Gen2MemoryBank	string	Select GJB memory bank to write R = Reserved E = EPC T = TID U = User 2 = User 2 3 = User 3 Default : E
datastring	string	Data string
WritePassword	string	Writing password · 8 HEX characters. (0~9, A,B,C,D,E,F)

34. **readGJB_UHF_USB**(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, ReadPassword)
readGJB_UHF_Ethernet(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, ReadPassword)

■ Description: Read data from UHF GJB tag.

■ Parameter:

Parameter	Type	Description
dataFormat	string	Set up callback returned data format · A : ASCII H : Hexadecimal (Default)
startBlockNo	int	Specify the 16-bit starting block number to read · Default is 0
byteSize	int	Specify the data lengths to read · default is 1
Gen2MemoryBank	string	Select GJB memory bank to read ·

		E = EPC T = TID U = User 2 = User 2 3 = User 3 Default : E
ReadPassword	string	Reading password · 8 HEX characters. (0~9, A,B,C,D,E,F)

■ Return Value(EPC Data):

DataFormat	Return string(example)
A	Data format is ASCII (ex: 24051324000103456400)
H	Data format is Hexadecimal (ex: 3234303531333234303030313033343536343030)

35. statusGJB_UHF(Gen2MemoryBank, action, StatusPassword)

statusGJB_UHF_USB(Gen2MemoryBank, action, StatusPassword)

statusGJB_UHF_Ethernet(Gen2MemoryBank, action, StatusPassword)

■ Description: Set memory status with password for UHF GJB tag.

■ Parameter:

Parameter	Type	Description
Gen2MemoryBank	string	Select GJB memory bank for set, E = EPC (default) T = TID U = User 2 = User 2 3 = User 3
action	string	Memory Status type A : Lock0(readable and writable) B : Lock1(read only) C : Lock2(write only) D : Lock3(non-readable and non-writable) *Each Memory support status : <ul style="list-style-type: none"> EPC area : A:read and write B: read only USER area(include User 、 User2 、 User3) : A:read and write B:read only C: write only D: not allow access

		<ul style="list-style-type: none"> TID area : B:read only D: not allow access SAFE area : C: write only D: not allow access
StatusPassword	string	Status password · 8 HEX characters. (0~9, A,B,C,D,E,F)

36. killGJB_UHF(KillPassword)

killGJB_UHF_USB(KillPassword)

killGJB_UHF_Ethernet(KillPassword)

■ Description: Kill UHF GJB tag.

■ Parameter:

Parameter	Type	Description
KillPassword	string	Killing password · 8 HEX characters. (0~9, A,B,C,D,E,F)

37. query_UHF_USB (dataFormat, PCReturnStatus, CRCReturnStatus)

query_UHF_Ethernet (dataFormat, PCReturnStatus, CRCReturnStatus)

■ Description: Read data from UHF tag memory (Q command)

■ Parameter:

Parameter	Type	Description
dataFormat	string	Set up callback return data format · A : ASCII H : Hexadecimal (Default)
PCReturnStatus	int	enable/disable PC value returned 0 : read epc data not include PC value 1 : read epc data include PC value
CRCReturnStatus	int	enable/disable CRC-16 value returned 0 : read epc data not include CRC-16 value 1 : read epc data include CRC-16 value

■ Return Value (Label information) :

Take query_UHF_USB("A", 1, 1) and query_UHF_USB("H", 1, 1) as an example :

(Both PC value and CRC-16 are returned)

DataFormat	Return string(example)
A	Data format is ASCII (ex: 24051324000103456400)
H	Data format is Hexadecimal (ex: 3234303531333234303030313033343536343030)

Take query_UHF_USB("A", 0, 0) and query_UHF_USB("H", 0, 0) as an example :

(Both PC value and CRC-16 are not returned)

DataFormat	Return string(example)
A	Data format is ASCII (ex: 0513240001034564)
H	Data format is Hexadecimal (ex: 30353133323430303031303334353634)

■ Return Value (Error code) :

Code Type	Description
64000000000000000000000000000000	Other errors
65000000000000000000000000000000	Mor than mermory space
66000000000000000000000000000000	The memory is locked
67000000000000000000000000000000	Insufficient reading power
68000000000000000000000000000000	Unpecific error
69000000000000000000000000000000	CRC error
6A000000000000000000000000000000	If an error occurs during writing · how many words have been written in the reply.
6B000000000000000000000000000000	If the tag replies incorrectly during writing · besides the error code, it would also replies how many words have been written.
6C000000000000000000000000000000	no tag exists
6D000000000000000000000000000000	command format error
6E000000000000000000000000000000	Failed to set power strength
6F000000000000000000000000000000	Failed to set regulations

38. RFIDAutoCalibration() 、 RFIDAutoCalibration_USB() 、 RFIDAutoCalibration_Ethernet()

- Description: Calibration for RFID label.
- Parameter:None

39. setDirectionAndMirror(direction,mirror) 、 setDirectionAndMirror_USB(direction,mirror) 、 setDirectionAndMirror_Ethernet (direction,mirror)

- Description: Set direction and mirror.
- Parameter:

Parameter	Type	Description
direction	int	Set direction, default:0 0 : Top out 1 : Bottom out
mirror	int	Set mirror 0 : No 1 : Yes

40. setShift (shiftY) 、 setShift_USB (shiftY) 、 setShift_Ethernet (shiftY)

- Description: Set the vertical displacement distance, when value is positive, it will shift in the printing direction, otherwise, it will shift in opposite direction.
- Parameter:
 - ➔ shiftY : int type, vertical displacement distance, the unit is dot.

41. printReverse(x_start, y_start, x_width, y_height) 、 printReverse_USB(x_start, y_start, x_width, y_height) 、 printReverse_Ethernet (x_start, y_start, x_width, y_height)

- Description: Reverse the designated area.
- Parameter:

Parameter	Type	Description
x_start	int	The x-coordinate of the area(in dots)
y_start	int	The y-coordinate of the area (in dots)
x_width	int	The width of area in dots
y_height	int	The height of area in dots

42. **setOffset(offset) 、 setOffset_USB(offset) 、 setOffset_Ethernet(offset)**

- Description: Set feed offset(Usually use with peel mode and cut mode)
- Parameter:
 - ➔ **offset** : double type, extra feed offset, the unit is mm

43. **setCutMode(mode, piece) 、 setCutMode_USB (mode, piece) 、 setCutMode_Ethernet (mode, piece)**

- Description: Set cut mode and cut number
- Parameter:

Parameter	Type	Description
mode	int	Set cut mode, default:1 0 : Backward 1 : Forward
piece	int	Set cut number

44. **setAfterPrintAction(mode) 、 setAfterPrintAction_USB(mode) 、 setAfterPrintAction_Ethernet(mode)**

- Description: Set action after print
- Parameter:

Parameter	Type	Description
mode	int	Set action after print, default:1 0 : Normal 1 : Tear Mode 2 : Peel Mode 3 : Cut Mode

45. **genericDefault () 、 genericDefault_USB () 、 genericDefault_Ethernet()**

- Description: Initialize the general setting value
- Parameter: None

46. **sensorDefault () 、 sensorDefault_USB () 、 sensorDefault_Ethernet ()**

- Description: Initialize the sensor setting value
- Parameter: None

47. **rfidSetupDefault () 、 rfidSetupDefault_USB () 、 rfidSetupDefault_Ethernet ()**

- Description: Initialize the RFID setting value
- Parameter: None

48. WifiFrequency(Frequency) 、WifiFrequency_USB(Frequency) 、WifiFrequency_Ethernet (Frequency)

- Description: When using a compatible 5G frequency band WiFi module, it can be used to switch between frequency bands.
- Parameter:

Parameter	Type	Description
Frequency	string	Module Frequency Setting: 2.4G: Use the 2.4GHz frequency band. 5G: Use the 5GHz frequency band. BOTH: Use dual-band frequency.

49. Bitmap(x,y, width,height,mode,filename)

Bitmap_USB(x,y, width,height,mode,filename)

Bitmap_Ethernet(x,y, width,height,mode,filename)

- Description: Convert the image into a monochrome bitmap and directly print using the printer.
- Parameter:

Parameter	Type	Description
x	string	The x-coordinate of the text (in dots)
y	string	The y-coordinate of the text (in dots)
width	int	The width of the image is represented in bytes.
height	int	The height of the image is represented in dots (pixels).
mode	int	Image format 0: OVERWRITE 1: OR 2: XOR
filename	string	The file name (including the path). The supported image formats are as follows: 1. BMP (Bitmap): A bitmap format. 2. JPG (JPEG): Compressed image format. 3. PNG (Portable Network Graphics): Lossless compressed image format. 4. GIF (Graphics Interchange Format): Format that supports multiple images, commonly used for animations. 5. TIFF (Tagged Image File Format): High-quality lossless compressed image format. 6. ICO (Icon): Icon format used for displaying icons of files, programs, or folders.

		7. WMF (Windows Metafile): Windows graphics file format. EMF (Enhanced Metafile): Enhanced Windows graphics file format.
--	--	---

50. compressBitmap(x,y, width,height,filename)

compressBitmap_USB(x,y, width,height,filename)

compressBitmap_Ethernet(x,y, width,height,filename)

- Description: Convert the image into a monochrome bitmap, compress it, and then print using the printer.

- Parameter:

Parameter	Type	Description
x	string	The x-coordinate of the text (in dots)
y	string	The y-coordinate of the text (in dots)
width	int	The width of the image is represented in bytes.
height	int	The height of the image is represented in dots (pixels).
filename	string	<p>The file name (including the path).</p> <p>The supported image formats are as follows:</p> <ol style="list-style-type: none"> 1. BMP (Bitmap): A bitmap format. 2. JPG (JPEG): Compressed image format. 3. PNG (Portable Network Graphics): Lossless compressed image format. 4. GIF (Graphics Interchange Format): Format that supports multiple images, commonly used for animations. 5. TIFF (Tagged Image File Format): High-quality lossless compressed image format. 6. ICO (Icon): Icon format used for displaying icons of files, programs, or folders. 7. WMF (Windows Metafile): Windows graphics file format. <p>EMF (Enhanced Metafile): Enhanced Windows graphics file format.</p>

1. C#(.Netframework)

1.Add GTSPL_SDK.dll as reference.

2.Import GTSPL_SDK:

```
using GTSPL_SDK;
```

3.Sample program:

```
//Single-machine printing
```

```
Driver driver = new Driver();
```

```
driver.openport("Printer Model");
```

```
driver.setup("54", "30", "2", "3", "0", "3", "0");
```

```
driver.sendcommand("DIRECTION 1");
```

```
driver.setDirectionAndMirror(1, 1);
```

```
driver.setShift(50);
```

```
driver.setAfterPrintAction(2);
```

```
driver.setOffset(20);
```

```
driver.setCutMode(0,2);
```

```
driver.clearbuffer();
```

```
driver.sendcommand("TEXT 100,100,\"3\",0,1,1,\"REVERSE\");
```

```
driver.printReverse(90, 90, 128, 40);
```

```
driver.barcode("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567");
```

```
driver.qrcode("220", "260", "M", "3", "A", "0", "AABCB03abcN123");
```

```
driver.printerfont("50", "10", "3", "0", "1", "1", "Print Font 123456");
```

```
driver.printerfontblock("35", "15", "790", "90", "1", "0", "8", "8", "0", "1", "We stand behind our products  
with one of the most comprehensive support programs in the Auto-ID industry.");
```

```
string file = Environment.CurrentDirectory;
```

```

driver.downloadbmp(file + "\\CIRCLE.BMP", "CIRCLE.BMP");

driver.sendcommand("PUTBMP 150,30,\"CIRCLE.BMP\"");

driver.windowfont(100, 20, 48, 0, 0, 0, "arial", "C# Driver test");

int BitmapResult = driver.Bitmap("-500", "70", 400, 350, 1, "Thunder.png ");

BitmapResult = driver.compressBitmap("-600", "70", 666, 357, "Cat1.jpg");    //BitmapResult is the
result of processing the Bitmap. A value of 1 means success, and a value of 0 means failure.

driver.printlabel("1", "1");

driver.genericDefault();

driver.sensorDefault();

driver.WifiFrequency("5G");


//RFID
driver.RFIDAutoCalibration();

driver.rfidSetupDefault();

//UHF GEN2    //startBlockNo: Gen2 default 2

driver.writeUHF("H", 2, 12, "E", "414142424343444445454646");

driver.printlabel("1", "1");

driver.EPCPWD_Action("U", "12345678");

driver.TIDPWD_Action("L", "12345678");

driver.USERPWD_Action("L", "12345678");

driver.AccessPWD_Action("S", "12345678");

driver.KillPWD_Action("L", "12345678");

driver.Set_RFIDPorcedure(8, 8, 32, 3, "N", 2, 2);

driver.Set_RFIDPorcedure_mm(5, 40, 32, 5, "N", 5, 5, "203");

driver.writeHF("H", 0, 12, "414142424343444445454646");

driver.printlabel("1", "1");

```

```
//UHF GJB Write Data
```

```
driver.writeGJB_UHF("H", 1, 12, "E", "414142424343444445454646", "12345678");
```

***Use write password to write data(startBlockNo: GJB default 1)**

```
//UHF GJB set password
```

```
// Use write password to set a new status password
```

```
driver.setPWD_Action("S", "S", "11112222", "12345678");
```

```
// Use write password to set a new read password
```

```
driver.setPWD_Action("R", "S", "33334444", "12345678");
```

```
// Use write password to set a new kill password
```

```
driver.setPWD_Action("K", "S", "55556666", "12345678");
```

```
// Use write password to set a new write password
```

```
driver.setPWD_Action("W", "S", "87654321", "12345678");
```

```
//UHF GJB Set status of data block
```

```
driver.statusGJB_UHF("E", "B", "11112222"); // Use status password to set a new status
```

```
driver.printlabel("1", "1");
```

```
//UHF GJB kill tag
```

```
driver.killGJB_UHF ("55556666"); // Use kill password to delete the tag
```

```
driver.printlabel("1", "1");
```

```
driver.closeport();
```

```
//Specify the USB interface
```

```
USB usb = new USB();
```

```
usb.openport_USB();
```

```
string[] PrinterName = usb.detectUSB_USB(); // There are two or more printers.
```

```
string[] PrinterName = usb.detectUSBStr_USB() == null ? null : usb.detectUSBStr_USB().Split('\n');
```

***If usb.detectUSBStr_USB() is not null, use '\n' to separate each model.**

```
usb.openports_USB(PrinterName[0]);           //Connect to the first detected printer

usb.setup_USB("54", "30", "2", "3", "0", "3", "0");

usb.sendcommand_USB("DIRECTION 1");

usb.setDirectionAndMirror_USB(1,1);

usb.setShift_USB(50);

usb.setAfterPrintAction_USB(2);

usb.setOffset_USB(20);

usb.setCutMode_USB(0,2);

usb.clearbuffer_USB();

usb.sendcommand_USB("TEXT 100,100,\"3\",0,1,1,\"REVERSE\"");

usb.printReverse_USB(90,90,128,40);

usb.barcode_USB("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567");

usb.qrcode_USB("220", "260", "M", "3", "A", "0", "AABCB03abcN123");

usb.printerfont_USB("50", "10", "3", "0", "1", "1", "Print Font 123456");

string file = Environment.CurrentDirectory;

usb.downloadpcx_USB(file + "\\UL.PCX", "UL.PCX"

usb.windowfont_USB(10, 100, 48, 0, 0, 0, "arial", "C# WIN TEST");

int BitmapResult = usb.Bitmap_USB("-500", "70", 400, 350, 1, "Thunder.png");

BitmapResult=usb.compressBitmap_USB("-600", "70", 666, 357, "Cat1.jpg"); //BitmapResult is the
result of processing the Bitmap. A value of 1 means success, and a value of 0 means failure.

usb.printlabel_USB("1", "1");

var status = usb.printerstatus_USB();

usb.genericDefault_USB();

usb.sensorDefault_USB();
```

```

usb.WifiFrequency_USB("5G");

// Print Simplified Chinese text

string ststring "默认简体中文测试";

usb.clearbuffer_USB();

usb.printerfont_USB ("100", "10", "TSS24.BF2", "0", "1", "1", stString);

usb.printlabel_USB (1, 1, this);

// Print traditional Chinese text

string ttstring "默認繁體中文測試";

usb.clearbuffer_USB();

usb.printerfont_USB("100", "10", " TST24.BF2", "0", "1", "1", ttString);

usb.printlabel_USB(1, 1, this);

//Print BLOCK

string ttString="We stand behind our products with one of the most comprehensive support programs
in the Auto-ID industry.";

usb.clearbuffer_USB();

usb.printerfontblock_USB("35","15","790","90","1","0","8","8","0","1", ttString);

usb.printlabel_USB(1, 1, this);

//RFID

usb.RFIDAutoCalibration_USB();

usb.rfidSetupDefault_USB();

//UHF GEN2

usb.writeUHF_USB("H", 2, 12, "E", "414142424343444445454646");

// startBlockNo: Gen2 default:2

usb.printlabel_USB("1", "1");

```

```

string data = usb.readUHF_USB("H", 0, 12, "E");

string data_Q = usb.query_UHF_USB( "A", 0, 0);           //Read EPC data(Q command)

usb.EPCPWD_Action_USB("L","12345678");

usb.TIDPWD_Action_USB("L", "12345678");

usb.USERPWD_Action_USB("L", "12345678");

usb.AccessPWD_Action_USB("U", "12345678");

usb.KillPWD_Action_USB("U", "12345678");

usb.Set_RFIDPorcedure_USB(8,8,32,3,"N",2,2);

usb.Set_RFIDPorcedure_mm_USB(5, 40, 32, 5, "N", 5, 5, "203");

usb.writeHF_USB("H", 0, 12, "414142424343444445454646");

usb.printlabel_USB("1", "1");


//UHF GJB write data

usb.writeGJB_UHF_USB("H", 1, 12, "E", "414142424343444445454646", "12345678");

*Use write password to write data(startBlockNo: GJB default 1)

//UHF GJB set password

// Use write password to set a new status password

usb.setPWD_Action_USB ("S", "S", "11112222", "12345678");

// Use write password to set a new read password

usb.setPWD_Action_USB ("R", "S", "33334444", "12345678");

// Use write password to set a new kill password

usb.setPWD_Action_USB ("K", "S", "55556666", "12345678");

//Use write password to set a new write password

usb.setPWD_Action_USB ("W", "S", "87654321", "12345678");

//UHF GJB Set status

usb.statusGJB_UHF_USB ("E", "B", "11112222"); // Use status password to set status

```



```
usb.printlabel_USB ("1", "1");
```

```
//UHF GJB read data
```

```
//Use read password, read data
```

```
string data =usb. readGJB_UHF_USB("H", 0, 12, "E", "33334444");
```

```
//UHF GJB kill rfid tag
```

```
usb.killGJB_UHF_USB ("55556666"); //Use kill password to kill rfid tag
```

```
usb.printlabel_USB ("1", "1");
```

```
usb.closePort_USB(1000);
```

```
usb.getDLLVersion_USB(1);
```

```
// Specify the Ethernet interface
```

```
SocketConnect socketconnect = new SocketConnect();
```

```
socketconnect.openport_Ethernet("192.168.66.177", 9100);
```

```
socketconnect.setup_Ethernet ("54", "30", "2", "3", "0", "3", "0");
```

```
socketconnect.sendcommand_Ethernet ("DIRECTION 1");
```

```
socketconnect.setDirectionAndMirror_Ethernet(1,1);
```

```
socketconnect.setShift_Ethernet(50);
```

```
socketconnect.setAfterPrintAction_Ethernet(2);
```

```
socketconnect.setOffset_Ethernet(20);
```

```
socketconnect.setCutMode_Ethernet(0,2);
```

```
socketconnect.clearbuffer_Ethernet ();
```

```
socketconnect.sendcommand_Ethernet("TEXT 100,100,\"3\",0,1,1,\"REVERSE\");
```

```
socketconnect.printReverse_Ethernet(90, 90, 128, 40);
```

```
socketconnect.barcode_Ethernet ("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567");
```

```
socketconnect.qrcode_Ethernet ("220", "260", "M", "3", "A", "0", "AABCB03abcN123");
```

```
socketconnect.printerfont_Ethernet ("50", "10", "3", "0", "1", "1", "Print Font 123456");
```

```
string file = Environment.CurrentDirectory;

socketconnect.downloadpcx_Ethernet (file + "\\UL.PCX", "UL.PCX");

socketconnect.windowfont_Ethernet (10, 100, 48, 0, 0, 0, "arial", "C# WIN TEST");

int BitmapResult = socketconnect.Bitmap_Ethernet("-500", "70", 400, 350, 1, "Thunder.png");

BitmapResult = socketconnect.compressBitmap_Ethernet("-600", "70", 666, 357, "Cat1.jpg"); //
BitmapResult is the result of processing the Bitmap. A value of 1 means success, and a value of 0
means failure.
```

```
socketconnect.printlabel_Ethernet ("1", "1");

var status = socketconnect.printerstatus_Ethernet ();

socketconnect.genericDefault_Ethernet();

socketconnect.sensorDefault_Ethernet();

socketconnect.WifiFrequency_Ethernet("5G");

// Print Simplified Chinese text

string stString="默认简体中文测试";

socketconnect.clearbuffer_Ethernet ();

socketconnect.printerfont_Ethernet ("100", "10", "TSS24.BF2", "0", "1", "1", stString);

socketconnect.printlabel_Ethernet (1, 1, this);

// Print traditional Chinese text

string ttString="默認繁體中文測試";

socketconnect.clearbuffer_Ethernet ();

socketconnect.printerfont_Ethernet ("100", "10", " TST24.BF2", "0", "1", "1", ttString);

socketconnect.printlabel_Ethernet (1, 1, this);

//Print BLOCK

string ttString="We stand behind our products with one of the most comprehensive support programs
in the Auto-ID industry.";
```

```

socketconnect.clearbuffer_Ethernet ();

socketconnect.printerfontblock_Ethernet ("35","15","790","90","1","0","8","8","0","1", ttString);

socketconnect.printlabel_Ethernet (1, 1, this);

//RFID

socketconnect.RFIDAutoCalibration_Ethernet();

socketconnect.rfidSetupDefault_Ethernet();

//UHF GEN2

socketconnect.writeUHF_Ethernet("H", 2, 12,"E","414142424343444445454646");

//startBlockNo : Gen2 default:2

socketconnect.printlabel_Ethernet ("1", "1");

string data = socketconnect.readUHF_Ethernet ("H", 0, 12, "E");

string data_Q = socketconnect.query_UHF_Ethernet ( "A", 0, 0);

//Read EPC data(Q command)

socketconnect.EPCPWD_Action_Ethernet ("L","12345678");

socketconnect.TIDPWD_Action_Ethernet ("L", "12345678");

socketconnect.USERPWD_Action_Ethernet ("L", "12345678");

socketconnect.AccessPWD_Action_Ethernet ("U", "12345678");

socketconnect.KillPWD_Action_Ethernet ("U", "12345678");

socketconnect.Set_RFIDPorcedure_Ethernet (8,8,32,3,"N",2,2);

socketconnect.Set_RFIDPorcedure_mm_Ethernet(5, 40, 32, 5, "N", 5, 5, "203");

socketconnect.writeHF_Ethernet ("H", 0, 12, "414142424343444445454646");

socketconnect.printlabel_Ethernet ("1", "1");

//UHF GJB write data

socketconnect.writeGJB_UHF_Ethernet ("H", 1, 12, "E", "414142424343444445454646", "12345678");

*Use write password to write data(startBlockNo: GJB default 1)

//UHF GJB set password

```

```
// Use write password to set a new status password
socketconnect.setPWD_Action_Ethernet("S", "S", "11112222", "12345678");

// Use write password to set a new read password
socketconnect.setPWD_Action_Ethernet ("R", "S", "33334444", "12345678");

// Use write password to set a new kill password
socketconnect.setPWD_Action_Ethernet ("K", "S", "55556666", "12345678");

// Use write password to set a new write password
socketconnect.setPWD_Action_Ethernet("W", "S", "87654321", "12345678");

//UHF GJB set status

// Use status password to set status
socketconnect.statusGJB_UHF_Ethernet ("E", "B", "11112222");

socketconnect.printlabel_Ethernet ("1", "1");

//UHF GJB read data

// Use read password to read data
string data = socketconnect. readGJB_UHF_Ethernet ("H", 0, 12, "E", "33334444");

//UHF GJB kill tag

socketconnect.killGJB_UHF_Ethernet ("55556666"); //Use kill password to kill tag

socketconnect.printlabel_Ethernet ("1", "1");

socketconnect.closePort_Ethernet (1000);
socketconnect.getDLLVersion_Ethernet (1);
```

2. Java

1.import jan-5.5.0.jar :

```
import com.sun.jna.Library;
```

```
import com.sun.jna.Native;
```

2.Create the class of the dll

```

class GTSPL {

    interface GTSPL_SDK extends Library {

        GTSPL_SDK INSTANCE = (GTSPL_SDK) Native.load("GTSPL_SDK_C", GTSPL_SDK.class);

        int openport_USB();

        int openport(String PrinterName);

        ....

    }
}

```

3. Sample program :

```

System.load(System.getProperty("user.dir") + "\\GTSPL_SDK_C.dll");

//Single-machine printing

GTSPL.GTSPL_SDK.INSTANCE.openport("Printer Model");

GTSPL.GTSPL_SDK.INSTANCE.setup("54", "30", "2", "3", "0", "3", "0");

GTSPL.GTSPL_SDK.INSTANCE.sendcommand("DIRECTION 1");

GTSPL.GTSPL_SDK.INSTANCE.setDirectionAndMirror(1, 1);

GTSPL.GTSPL_SDK.INSTANCE.setShift(50);

GTSPL.GTSPL_SDK.INSTANCE.setAfterPrintAction(2);

GTSPL.GTSPL_SDK.INSTANCE.setOffset(20);

GTSPL.GTSPL_SDK.INSTANCE.setCutMode(0,2);

GTSPL.GTSPL_SDK.INSTANCE.clearbuffer();

GTSPL.GTSPL_SDK.INSTANCE.sendcommand("TEXT 100,100,\"3\",0,1,1,\"REVERSE\"");

GTSPL.GTSPL_SDK.INSTANCE.printReverse(90, 90, 128, 40);

GTSPL.GTSPL_SDK.INSTANCE.barcode("30", "30", "128", "100", "1", "0", "2", "2",
"barcode1234567");

GTSPL.GTSPL_SDK.INSTANCE.qrcode("50", "100", "H", "4", "A", "0", "QRcode1234567");

GTSPL.GTSPL_SDK.INSTANCE.printerfont("50", "10", "2", "0", "1", "1", "Print Font 123456");

GTSPL.GTSPL_SDK.INSTANCE.downloadbmp(System.getProperty("user.dir") + "\\CIRCLE.BMP",

```

```
"CIRCLE.BMP");

GTSPL.GTSPL_SDK.INSTANCE.windowfont(5, 150, 48, 0, 0, 0, "arial", "JAVA WIN DRIVER");

GTSPL.GTSPL_SDK.INSTANCE.sendcommand("PUTBMP 150,30,\"CIRCLE.BMP\"");

GTSPL.GTSPL_SDK.INSTANCE.printerfontblock("15", "15", "790", "90", "0", "0", "8", "8", "20",
"2", "We stand behind our products with one of the most comprehensive support programs in the
Auto-ID industry.");
```

```
// Initialization
```

```
GTSPL.GTSPL_SDK.INSTANCE.genericDefault();

GTSPL.GTSPL_SDK.INSTANCE.sensorDefault();

GTSPL.GTSPL_SDK.INSTANCE.rfidSetupDefault();
```

```
// Modify Wifi frequency band
```

```
GTSPL.GTSPL_SDK.INSTANCE.WifiFrequency("5G");
```

```
//Print Block
```

```
WString str = new WString("-- 默认简体中文测试：海天米醋白米醋 1 瓶 450ml --");

GTSPL.GTSPL_SDK.INSTANCE.printerfontblockUnicode("15", "15", "790", "90", "TSS24.BF2", "0", "1",
"1", "0", "1", str);

GTSPL.GTSPL_SDK.INSTANCE.printlabel("1", "1");

GTSPL.GTSPL_SDK.INSTANCE.closeport();

GTSPL.GTSPL_SDK.INSTANCE.getDLLVersion(0);
```

```
//UHF GEN2
```

```
GTSPL.GTSPL_SDK.INSTANCE.writeUHF("H", 0, 12, "E", "11223344556677889900AABB");

GTSPL.GTSPL_SDK.INSTANCE.printlabel("1", "1");

GTSPL.GTSPL_SDK.INSTANCE.EPCPWD_Action("L", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.TIDPWD_Action("L", "12345678");
```

```

GTSPL.GTSPL_SDK.INSTANCE.USERPWD_Action("L", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.AccessPWD_Action("U", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.KillPWD_Action("U", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.Set_RFIDPorcedure(8,8,32,"3","N",2,2);

GTSPL.GTSPL_SDK.INSTANCE.Set_RFIDPorcedure_mm(5, 40, 32, 5, "N", 5, 5, "203");

GTSPL.GTSPL_SDK.INSTANCE.writeHF("H", 0, 12, "414142424343444445454646")

GTSPL.GTSPL_SDK.INSTANCE.printlabel("1", "1");

//UHF GJB write data

GTSPL.GTSPL_SDK.INSTANCE.writeGJB_UHF("H", 1, 12, "E", "414142424343444445454646",
"12345678");

*Use write password to write data(startBlockNo: GJB default 1)

//UHF GJB set password

// Use write password to set a new status password
GTSPL.GTSPL_SDK.INSTANCE.setPWD_Action("S", "S", "11112222", "12345678");

// Use write password to set a new read password
GTSPL.GTSPL_SDK.INSTANCE.setPWD_Action("R", "S", "33334444", "12345678");

// Use write password to set a new kill password GTSPL.GTSPL_SDK.INSTANCE.setPWD_Action("K",
"S", "55556666", "12345678");

// Use write password to set a new write password
GTSPL.GTSPL_SDK.INSTANCE.setPWD_Action("W", "S", "87654321", "12345678");


//UHF GJB set status

// Use status password to set status
GTSPL.GTSPL_SDK.INSTANCE.statusGJB_UHF("E", "B", "11112222");

GTSPL.GTSPL_SDK.INSTANCE.printlabel("1", "1");

```

```

//UHF GJB kill tag

//Use kill password to kill rfid tag

GTSP.LGTSP.L_SDK.INSTANCE.killGJB_UHF ("55556666");

GTSP.LGTSP.L_SDK.INSTANCE.printlabel("1", "1");

GTSP.LGTSP.L_SDK.INSTANCE.closeport();

// Print Simplified Chinese text

WString str=new WString("默认简体中文测试"); //Need WString of jni

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer();

GTSP.LGTSP.L_SDK.INSTANCE.printerfontUnicode("100", "10", "TSS24.BF2", "0", "1", "1", str);

GTSP.LGTSP.L_SDK.INSTANCE.printlabel(1, 1);

// Print traditional Chinese text

WString str=new Wstring("默认繁体中文测试"); ///Need WString of jni

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer();

GTSP.LGTSP.L_SDK.INSTANCE. printerfontUnicode("100", "10", " TST24.BF2", "0", "1", "1", str);

GTSP.LGTSP.L_SDK.INSTANCE.printlabel(1, 1);

GTSP.LGTSP.L_SDK.INSTANCE.closeport();

GTSP.LGTSP.L_SDK.INSTANCE.getDLLVersion(0);

//RFID Calibration

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer();

GTSP.LGTSP.L_SDK.INSTANCE.RFIDAutoCalibration();

GTSP.LGTSP.L_SDK.INSTANCE.closeport();


// Specify the USB interface

// There are two or more printers

String PrinterName = GTSP.LGTSP.L_SDK.INSTANCE.detectUSBStr_USB();

String[] PrinterNameStringArray = PrinterName.split("\\n+");

```



```

GTSP.LGTSP.L_SDK.INSTANCE.openport_USB("Printer Model");

// Single-machine printing

GTSP.LGTSP.L_SDK.INSTANCE.openport_USB();

GTSP.LGTSP.L_SDK.INSTANCE.setup_USB("54", "30", "2", "3", "0", "3", "0");

GTSP.LGTSP.L_SDK.INSTANCE.sendcommand_USB("DIRECTION 1");

GTSP.LGTSP.L_SDK.INSTANCE.setDirectionAndMirror_USB(1, 1);

GTSP.LGTSP.L_SDK.INSTANCE.setShift_USB(50);

GTSP.LGTSP.L_SDK.INSTANCE.setAfterPrintAction_USB(2);

GTSP.LGTSP.L_SDK.INSTANCE.setOffset_USB(20);

GTSP.LGTSP.L_SDK.INSTANCE.setCutMode_USB(0,2);

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer_USB();

GTSP.LGTSP.L_SDK.INSTANCE.sendcommand_USB("TEXT 100,100,\"3\",0,1,1,\"REVERSE\"");

GTSP.LGTSP.L_SDK.INSTANCE.printReverse_USB(90, 90, 128, 40);

GTSP.LGTSP.L_SDK.INSTANCE.barcode_USB("30", "30", "128", "100", "1", "0", "2", "2",
"barcode1234567");

GTSP.LGTSP.L_SDK.INSTANCE.qrcode_USB("50", "100", "H", "4", "A", "0", "QRcode1234567");

GTSP.LGTSP.L_SDK.INSTANCE.printerfont_USB("50", "10", "2", "0", "1", "1", "Print Font 123456");

GTSP.LGTSP.L_SDK.INSTANCE.windowfont_USB(5, 150, 48, 0, 0, 0, "arial", "JAVA WIN DRIVER");

GTSP.LGTSP.L_SDK.INSTANCE.downloadpcx_USB(System.getProperty("user.dir")+ "\\UL.PCX",
"UL.PCX");

    GTSP.LGTSP.L_SDK.INSTANCE.sendcommand_USB("PUTPCX 50,10,\"UL.PCX\"");

GTSP.LGTSP.L_SDK.INSTANCE.printerfontblock_USB("15", "15", "790", "90", "0", "0", "8", "8", "20",
"2", "We stand behind our products with one of the most comprehensive support programs in the
Auto-ID industry.");

// Initialization

```

```

GTSP.LGTSP.L_SDK.INSTANCE.genericDefault_USB();

GTSP.LGTSP.L_SDK.INSTANCE.sensorDefault_USB();

GTSP.LGTSP.L_SDK.INSTANCE.rfidSetupDefault_USB();


// Modify Wifi frequency band
GTSP.LGTSP.L_SDK.INSTANCE.WifiFrequency_USB("5G");


//Print Block
WString str = new WString("-- 默认简体中文测试：海天米醋白米醋 1 瓶 450ml --");
GTSP.LGTSP.L_SDK.INSTANCE.printerfontblockUnicode_USB("15", "15", "790", "90", "TSS24.BF2", "0",
"1", "1", "0", "1", str);
GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB("1", "1");
String status = GTSP.LGTSP.L_SDK.INSTANCE.printerstatus_USB();


//UHF GEN2
GTSP.LGTSP.L_SDK.INSTANCE.writeUHF_USB("H", 0, 12, "E", "11223344556677889900AABB");
GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB("1", "1");
String status = GTSP.LGTSP.L_SDK.INSTANCE.readUHF_USB("A", 0, 12, "E");
GTSP.LGTSP.L_SDK.INSTANCE.EPCPWD_Action_USB("L", "12345678");
GTSP.LGTSP.L_SDK.INSTANCE.TIDPWD_Action_USB("L", "12345678");
GTSP.LGTSP.L_SDK.INSTANCE.USERPWD_Action_USB("L", "12345678");
GTSP.LGTSP.L_SDK.INSTANCE.AccessPWD_Action_USB("U", "12345678");
GTSP.LGTSP.L_SDK.INSTANCE.KillPWD_Action_USB("U", "12345678");
GTSP.LGTSP.L_SDK.INSTANCE.Set_RFIDPorcedure_USB(8, 8, 32, "3", "N", 2, 2);
GTSP.LGTSP.L_SDK.INSTANCE.Set_RFIDPorcedure_mm_USB(5, 40, 32, 5, "N", 5, 5, "203");
GTSP.LGTSP.L_SDK.INSTANCE.writeHF_USB("H", 0, 12, "4141424243434444445454646")

```

```
GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB("1", "1");
```

```
//UHF GJB write data
```

```
GTSP.LGTSP.L_SDK.INSTANCE.writeGJB_UHF_USB("H", 1, 12, "E", "4141424243434444445454646",  
"12345678");
```

***Use write password to write data(startBlockNo: GJB default 1)**

```
//UHF GJB set password
```

```
// Use write password to set a new status password
```

```
GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_USB ("S", "S", "11112222", "12345678");
```

```
// Use write password to set a new read password
```

```
GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_USB ("R", "S", "33334444", "12345678");
```

```
// Use write password to set a new kill password
```

```
GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_USB ("K", "S", "55556666", "12345678");
```

```
// Use write password to set a new write password
```

```
GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_USB ("W", "S", "87654321", "12345678");
```

```
//UHF GJB set status
```

```
// Use status password to set status
```

```
GTSP.LGTSP.L_SDK.INSTANCE.statusGJB_UHF_USB ("E", "B", "11112222");
```

```
GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB( ("1", "1");
```

```
//UHF GJB read data
```

```
//Use read password, read data
```

```
String data = GTSP.LGTSP.L_SDK.INSTANCE. readGJB_UHF_USB("H", 0, 12, "E", "33334444");
```

```
//UHF GJB kill tag
```

```
//Use kill password to kill rfid tag
```

```
GTSP.LGTSP.L_SDK.INSTANCE.killGJB_UHF_USB ("55556666");
```

```

GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB( "1", "1");

GTSP.LGTSP.L_SDK.INSTANCE.closePort_USB(1000);

GTSP.LGTSP.L_SDK.INSTANCE.getDLLVersion_USB(1);

// Print Simplified Chinese text

WString str=new WString("默认简体中文测试"); // Need WString of jni

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer_USB();

GTSP.LGTSP.L_SDK.INSTANCE.printerfontUnicode_USB ("100", "10", "TSS24.BF2", "0", "1", "1", str);

    GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB (1, 1);

// Print traditional Chinese text

WString str=new Wstring("默认繁体中文测试");// Need WString of jni

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer_USB();

GTSP.LGTSP.L_SDK.INSTANCE. printerfontUnicode_USB ("100", "10", " TST24.BF2", "0", "1", "1", str);

GTSP.LGTSP.L_SDK.INSTANCE.printlabel_USB(1, 1);

GTSP.LGTSP.L_SDK.INSTANCE.closeport_USB();

GTSP.LGTSP.L_SDK.INSTANCE.getDLLVersion_USB (0);

//RFID Calibration

GTSP.LGTSP.L_SDK.INSTANCE.clearbufferr_USB ();

GTSP.LGTSP.L_SDK.INSTANCE.RFIDAutoCalibrationr_USB ();

GTSP.LGTSP.L_SDK.INSTANCE.closeportr_USB ();


// Specify the Ethernet interface

GTSP.LGTSP.L_SDK.INSTANCE.openport_Ethernet(IP,Port);

GTSP.LGTSP.L_SDK.INSTANCE.setup_Ethernet("54", "30", "2", "3", "0", "3", "0");

GTSP.LGTSP.L_SDK.INSTANCE.sendcommand_Ethernet("DIRECTION 1");

GTSP.LGTSP.L_SDK.INSTANCE.setOffset_Ethernet(20);

GTSP.LGTSP.L_SDK.INSTANCE.setCutMode_Ethernet(0,2);

```

```

GTSP.LGTSP.L_SDK.INSTANCE.setDirectionAndMirror_Ethernet(1, 1);

GTSP.LGTSP.L_SDK.INSTANCE.setShift_Ethernet(50);

GTSP.LGTSP.L_SDK.INSTANCE.setAfterPrintAction_Ethernet(2);

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer_Ethernet();

GTSP.LGTSP.L_SDK.INSTANCE.sendcommand_Ethernet("TEXT 100,100,\"3\",0,1,1,\"REVERSE\"");

GTSP.LGTSP.L_SDK.INSTANCE.printReverse_Ethernet(90, 90, 128, 40);

GTSP.LGTSP.L_SDK.INSTANCE.barcode_Ethernet("30", "30", "128", "100", "1", "0", "2", "2",
"barcode1234567");

GTSP.LGTSP.L_SDK.INSTANCE.qrcode_Ethernet("50", "100", "H", "4", "A", "0", "QRcode1234567");

GTSP.LGTSP.L_SDK.INSTANCE.printerfont_Ethernet("50", "10", "2", "0", "1", "1", "Print Font
123456");

GTSP.LGTSP.L_SDK.INSTANCE.windowfont_Ethernet(5, 150, 48, 0, 0, 0, "arial", "JAVA WIN DRIVER");

GTSP.LGTSP.L_SDK.INSTANCE.downloadpcx_Ethernet(System.getProperty("user.dir")+ "\\UL.PCX",
"UL.PCX");

    GTSP.LGTSP.L_SDK.INSTANCE.sendcommand_Ethernet("PUTPCX 50,10,\"UL.PCX\"");

GTSP.LGTSP.L_SDK.INSTANCE.printerfontblock_Ethernet("15", "15", "790", "90", "0", "0", "8", "8",
"20", "2", "We stand behind our products with one of the most comprehensive support programs in
the Auto-ID industry.");

// Initialization

GTSP.LGTSP.L_SDK.INSTANCE.genericDefault_Ethernet();

GTSP.LGTSP.L_SDK.INSTANCE.sensorDefault_Ethernet();

GTSP.LGTSP.L_SDK.INSTANCE.rfidSetupDefault_Ethernet();

// Modify Wifi frequency band

GTSP.LGTSP.L_SDK.INSTANCE.WifiFrequency_Ethernet("5G");

```

```
// Print Block
```

```
WString str = new WString("-- 默认简体中文测试：海天米醋白米醋 1 瓶 450ml --");

GTSPL.GTSPL_SDK.INSTANCE.printerfontblockUnicode_Ethernet("15", "15", "790", "90", "TSS24.BF2",
"0", "1", "1", "0", "1", str);

GTSPL.GTSPL_SDK.INSTANCE.printlabel_Ethernet("1", "1");

String status = GTSPL.GTSPL_SDK.INSTANCE.printerstatus_Ethernet();
```

```
//UHF GEN2
```

```
GTSPL.GTSPL_SDK.INSTANCE.writeUHF_Ethernet("H", 0, 12, "E", "11223344556677889900AABB");

GTSPL.GTSPL_SDK.INSTANCE.printlabel_Ethernet("1", "1");

String status = GTSPL.GTSPL_SDK.INSTANCE.readUHF_Ethernet("A", 0, 12, "E");

GTSPL.GTSPL_SDK.INSTANCE.EPCPWD_Action_Ethernet("L", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.TIDPWD_Action_Ethernet("L", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.USERPWD_Action_Ethernet("L", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.AccessPWD_Action_Ethernet("U", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.KillPWD_Action_Ethernet("U", "12345678");

GTSPL.GTSPL_SDK.INSTANCE.Set_RFIDPorcedure_Ethernet(8, 8, 32, "3", "N", 2, 2);

GTSPL.GTSPL_SDK.INSTANCE.Set_RFIDPorcedure_mm_Ethernet(5, 40, 32, 5, "N", 5, 5, "203");

GTSPL.GTSPL_SDK.INSTANCE.writeHF_Ethernet("H", 0, 12, "4141424243434444445454646")

GTSPL.GTSPL_SDK.INSTANCE.printlabel_Ethernet("1", "1");
```

```
//UHF GJB write data
```

```
GTSPL.GTSPL_SDK.INSTANCE.writeGJB_UHF_Ethernet("H", 1, 12, "E",
"4141424243434444445454646", "12345678");
```

***Use write password to write data(startBlockNo: GJB default 1)**

//UHF GJB set password

// Use write password to set a new status password

GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_Ethernet("S", "S", "11112222", "12345678");

// Use write password to set a new read password

GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_Ethernet("R", "S", "33334444", "12345678");

// Use write password to set a new kill password

GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_Ethernet("K", "S", "55556666", "12345678");

// Use write password to set a new write password

GTSP.LGTSP.L_SDK.INSTANCE.setPWD_Action_Ethernet("W", "S", "87654321", "12345678");

//UHF GJB set status

// Use status password to set status

GTSP.LGTSP.L_SDK.INSTANCE.statusGJB_UHF_Ethernet("E", "B", "11112222");

GTSP.LGTSP.L_SDK.INSTANCE.printlabel_Ethernet(("1", "1");

//UHF GJB read data

//Use read password, read data

String data = GTSP.LGTSP.L_SDK.INSTANCE. readGJB_UHF_Ethernet("H", 0, 12, "E", "33334444");

//UHF GJB kill tag

//Use kill password to kill rfid tag

GTSP.LGTSP.L_SDK.INSTANCE.killGJB_UHF_Ethernet("5555666");

GTSP.LGTSP.L_SDK.INSTANCE.printlabel_Ethernet(("1", "1");

GTSP.LGTSP.L_SDK.INSTANCE.closePort_Ethernet(1000);

GTSP.LGTSP.L_SDK.INSTANCE.getDLLVersion_Ethernet(1);

//Print Simplified Chinese text

WString str=new WString("默认简体中文测试"); // Need WString of jni

GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer_Ethernet();

```
GTSP.LGTSP.L_SDK.INSTANCE.printerfontUnicode_Ethernet("100", "10", "TSS24.BF2", "0", "1", "1",
str);
```

```
GTSP.LGTSP.L_SDK.INSTANCE.printlabel_Ethernet(1, 1);
```

```
// Print traditional Chinese text
```

```
WString str=new Wstring("默認繁體中文測試");// Need WString of jni
```

```
GTSP.LGTSP.L_SDK.INSTANCE.clearbuffer_Ethernet();
```

```
GTSP.LGTSP.L_SDK.INSTANCE. printerfontUnicode_Ethernet("100", "10", " TST24.BF2", "0", "1", "1",
str);
```

```
GTSP.LGTSP.L_SDK.INSTANCE.printlabel_Ethernet(1, 1);
```

```
GTSP.LGTSP.L_SDK.INSTANCE.closeport_Ethernet();
```

```
GTSP.LGTSP.L_SDK.INSTANCE.getDLLVersion_Ethernet(0);
```

```
//RFID Calibration
```

```
GTSP.LGTSP.L_SDK.INSTANCE.clearbufferr_Ethernet();
```

```
GTSP.LGTSP.L_SDK.INSTANCE.RFIDAutoCalibrationr_Ethernet();
```

```
GTSP.LGTSP.L_SDK.INSTANCE.closeport_Ethernet();
```

4.DLL Location :

[Using IDE]

GTSP.L_SDK.dll placed under the bin folder of jdk

Example : C:\Program Files\Java\jdk1.8.0_221\bin

GTSP.L_SDK_C.dll placed under the java project folder

[Directly execute .jar file when the jdk is not installed]

GTSP.L_SDK.dll placed under the bin folder of jre

Example : C:\Program Files\Java\jre1.8.0_251\bin

GTSP_SDK_C.dll placed in the same path as the .jar file

Under the same path as the .jar file, there also needs a lib folder, and there must be jna-5.5.0.jar in it

5.Driver Mode, execute need to install printer driver

3. Asp.Net

1.Add x86 to the dll in project

2.Sample program:

Single-machine printing

```
GTSP_SDK.Driver driver = new GTSP_SDK.Driver();
driver.openport("Printer Model");
driver.setup("54", "30", "2", "3", "0", "3", "0");
driver.sendcommand("DIRECTION 1");
driver.clearbuffer();
driver.barcode("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567");
string path = HttpContext.Current.Server.MapPath("~/");
driver.downloadpcx(path + "\\CIRCLE.BMP", "CIRCLE.BMP");
driver.sendcommand("PUTBMP 150,30,\"CIRCLE.BMP\"");
driver.windowfont(50, 20, 48, 0, 0, 0, "impact", "Windows Font ASP.NET");
driver.getDLLVersion(1);
driver.printlabel("1", "1");
driver.closeport();
```

```
//Specify the USB transport interface
```

```

GTSPL_SDK.USB usb = new GTSPL_SDK.USB();

usb.openport_USB();

usb.setup_USB("54", "30", "2", "3", "0", "3", "0");

usb.sendcommand_USB("DIRECTION 1");

usb.clearbuffer_USB();

usb.barcode_USB("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567usb");

usb.printerfont_USB("50", "10", "3", "0", "1", "1", "Print Font usb Asp");

usb.downloadpcx_USB(path + "\\UL.PCX", "UL.PCX");

usb.sendcommand_USB("PUTPCX 50,10,\"UL.PCX\"");

usb.windowfont_USB(10, 100, 48, 0, 0, 0, "impact", "Windows Font ASP USB");

usb.printlabel_USB("1", "1");

var status = usb.printerstatus_USB();

usb.getDLLVersion_USB(1);

// Print Simplified Chinese text

string ststring "默认简体中文测试";

usb.clearbuffer_USB();

usb.printerfont_USB("100", "10", "TSS24.BF2", "0", "1", "1", stString);

usb.printlabel_USB(1, 1);

// Print traditional Chinese text

string ttstring "默認繁體中文測試";

usb.clearbuffer_USB();

usb.printerfont_USB ("100", "10", " TST24.BF2", "0", "1", "1", ttString);

usb.printlabel_USB (1, 1);

usb.closeport_USB();

```

4. VB.Net

1.Add dll into the project

2.Sample program:

```
Dim path = Environment.CurrentDirectory

//Single-machine printing

Dim driver As New GTSPL_SDK.Driver

driver.openport("Gainscha GS-2208D")

driver.setup("54", "30", "2", "3", "0", "3", "0")

driver.sendcommand("DIRECTION 1")

driver.clearbuffer()

driver.barcode("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567")

driver.printerfont("50", "10", "3", "0", "1", "1", "PrintFont in vb.net")

driver.downloadbmp(path + "\\CIRCLE.BMP", "CIRCLE.BMP")

driver.sendcommand("PUTBMP 50,10, ""CIRCLE.BMP""")

driver.windowfont(50, 20, 48, 0, 0, 0, "impact", "VB Imapct Driver")

driver.printlabel("1", "1")

driver.closeport()


// Specify the USB interface

Dim usb As New GTSPL_SDK.USB

usb.openport_USB()

usb.setup_USB("54", "30", "2", "3", "0", "3", "0")

usb.sendcommand_USB("DIRECTION 1")
```

```

usb.clearbuffer_USB()

usb.barcode_USB("30", "30", "128", "100", "1", "0", "2", "2", "barcode1234567")

usb.downloadbmp_USB(path + "\\CIRCLE.BMP", "CIRCLE.BMP")

usb.sendcommand_USB("PUTBMP 50,10,""CIRCLE.BMP"")

usb.windowfont_USB(50, 20, 48, 0, 0, 0, "impact", "VB Imapct Driver")

Dim status = usb.printerstatus_USB()

Dim vcode = usb.getDLLVersion_USB("1")

usb.printlabel_USB("1", "1")

usb.closeport_USB()

// Print Simplified Chinese text

usb.clearbuffer_USB();

usb.printerfont_USB("100", "10", "TSS24.BF2", "0", "1", "1", "默认简体中文测试");

usb.printlabel_USB(1, 1);

// Print traditional Chinese text

usb.clearbuffer_USB();

usb.printerfont_USB ("100", "10", " TST24.BF2", "0", "1", "1", "默認繁體中文測試");

usb.printlabel_USB (1, 1);

usb.closeport_USB();

```

Appendix

Code Type	Description	Narrow : Width					Max.data length
		1:1	1:2	1:3	2:5	3:7	
128	Code 128, switching code subset automatically.	V					
128M	Code 128, switching code subset manually.	V					
EAN128	EAN128, switching code subset automatically.	V					
EAN128M	EAN128M, switching code subset manually.	V					
25	Interleaved 2 of 5.		V	V	V		Length is even
25C	Interleaved 2 of 5 with check digit.		V	V	V		Length is odd
25S	Standard 2 of 5.		V	V	V		
25I	Industrial 2 of 5.		V	V	V		
39	Code 39, switching standard and full ASCII mode automatically.		V	V	V		
39C	Code 39 with check digit.		V	V	V		
93	Code 93.			V			
EAN13	EAN 13.	V					12
EAN13+2	EAN 13 with 2 digits add-on.	V					14
EAN13+5	EAN 13 with 5 digits add-on.	V					17
EANB	EAN 8.	V					7
EANB+2	EAN 8 with 2 digits add-on.	V					96
EANB+5	EAN 8 with 5 digits add-on.	V					12
CODA	Codabar.		V	V	V		
POST	Postnet.	V					5,9,11
UPCA	UPC-A.	V					11
UPCA+2	UPC-A with 2 digits add-on.	V					13
UPA+5	UPC-A with 5 digits add-on.	V					16
UPCE	UPC-E.	V					6
UPCE+2	UPC-E with 2 digits add-on.	V					8
UPE+5	UPC-E with 5 digits add-on.	V					11
MSI	MSI.		V	V	V		
MSIC	MSI with check digit.		V	V	V		
PLESSEY	PLESSEY.		V	V	V		

CPOST	China post.					V	
ITF14	ITF14.		V	V	V		13
EAN14	EAN14.	V					13
11	Code 11.		V	V	V		
TELEPEN	Telepen. *Since V6.89EZ.		V	V	V		
TELEPENN	Telepen number. *Since V6.89EZ.		V	V	V		
PLANET	Planet. *Since V6.89EZ.	V					
CODE49	Code 49. *Since V6.89EZ.	V					
DPI	Deutsche Post Identcode. *Since V6.91EZ.		V	V	V		11
DPL	Deutsche Post Leitcode. *Since V6.91EZ.		V	V	V		13
LOGMARS	A special use of Code 39. *Since V6.88EZ.		V	V	V		