

# [TSPL Linux SDK]

[Printer TSPL Command Development Manual v1.2]

1. Information of the Manual.....	3
2.Operation System.....	3
3.Remark.....	3
4.Method.....	4
4.1 PrinterCreator.....	4
4.2 PrinterCreatorS.....	5
4.3 PrinterDestroy.....	6
4.4 PortOpen.....	7
4.5 PortClose.....	9
4.6 WriteData.....	10
4.7 ReadData.....	11
4.8 DirectIO.....	12
4.9 TSPL_SelfTest.....	14
4.10 TSPL_PrintImage.....	15
4.11 TSPL_Setup.....	17
4.12 TSPL_ClearBuffer.....	19
4.13 TSPL_Box.....	20
4.14 TSPL_BarCode.....	22
4.15 TSPL_QrCode.....	25
4.16 TSPL_Text.....	27
4.17 TSPL_Print.....	29
4.18 TSPL_FormFeed.....	30
4.19 TSPL_SetTear.....	31
4.20 TSPL_SetRibbon.....	32
4.21 TSPL_Offset.....	33
4.22 TSPL_Direction.....	34
4.23 TSPL_Feed.....	35
4.24 TSPL_Home.....	36
4.25 TSPL_Learn.....	37
4.26 TSPL_GetSN.....	38
4.27 TSPL_GetPrinterStatus.....	39
4.28 TSPL_SetCodePage.....	40
4.29 TSPL_PDF417.....	42
4.30 TSPL_Block.....	45
4.31 TSPL_Reverse.....	48
4.32 TSPL_GapDetect.....	49
4.33 TSPL_BitMap.....	50

# **1. Information of the Manual**

This SDK manual provides the so file information for Linux application development.

We continuously promote and update the function and quality of all our products. Any change to the product specification and the manual will be without any further notice.

## **2.Operation System**

- Kernel 2.6.32 or higher
- Linux 32 bit/64 bit

## **3.Remark**

- When error code Return Value is greater than 0, it is the internal error of Linux system, please refer to related help file.

## 4.Method

### 4.1 PrinterCreator

Set up the target printer of specified model (should create target printer before using any function).

```
int PrinterCreator(  
  
    void* handle,  
  
    const TCHAR* model  
  
);
```

#### Parameter:

*void\* handle*

[in,out] The created target printer object.

*const TCHAR\* model*

[in] Specify the model of target printer.

#### Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_INVALID_MODEL	-8	Invalid model name

## 4.2 PrinterCreatorS

Set up the target printer of specified model, the function is same to PrinterCreator (should create target printer before using any function).

**Void\* PrinterCreatorS(**

**const TCHAR\* *model***

**);**

### **Parameter:**

*const TCHAR\* model*

[in] Specify the model of target printer

### **Return:**

Success : return the handle of printer object.

Fail: return NULL, invalid handle.

## 4.3 PrinterDestroy

Release the resource of specified model printer that has set up (after operation completed and no more operation for printer, it should release the printer that has set up).

**int PrinterDestroy(**

**void\* *handle***

**);**

### Parameter:

*void\* handle*

[in] The handle of target printer object which needs to release.

### Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle

## 4.4 PortOpen

Open the communication port and connect with the printer. After successfully connected, other functions can be used. If failed connecting, please check the error information.

```
int PortOpen(  
  
    void* handle,  
  
    const TCHAR* ioSettings  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*const TCHAR\* ioSettings*

[in]Set up the parameter of communication port that connected to the target printer.Please see as below:

### Configuration List:

Type	Configuration	Description	Sample
USB	<b>USB</b> [,Position/Model/PortNum]	USB: connect any USB printer of our company USB[,Position]: When connecting to multi printers of our company, can specify connecting to one particular USB printer through USB position information (Position parameter)	USB USB,Port_#0004.Hub_#0003 USB,LPG4 USB,USB001
NET	<b>NET</b> , IP Add (IPV4)[,Port]	Specify the IP add and port of internet printer. If not specifying port, the default port is 9100.	NET,192.168.0.36 NET,192.168.0.36,9100
COM	<b>COM</b> <i>n</i> ,BAUDRATE_ <i>rate</i>	Specify the number and baud rate of connected serial port .	COM5,BAUDRATE_19200
LPT	<b>LPT</b> <i>n</i>	Specify the number of connected parallel port.	LPT1

Note: [ ] indicates selective parameter

**Return Value:**

Error Code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_OPEN_FAILED	-311	Port open failed



## 4.5 PortClose

This function is to close the communication port and disconnect with the printer.

```
int PortClose(  
    void* handle  
);
```

### Parameter:

*void\* handle*  
[in,out] The created target printer object.

### Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle

## 4.6 WriteData

This function is to send data to the printer.

```
int WriteData(  
  
    void* handle,  
  
    unsigned char* writeData,  
  
    unsigned int writeNum  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*unsigned char\* writeData*

[in] The data sent to the printer (hex string).

*unsigned int writeNum*

[in] The length of the data sent.

### Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout
E_IO_READ_FAILED	-331	Read failed
E_IO_READ_TIMEOUT	-332	Read timeout

## 4.7 ReadData

This function is to read the printer data.

```
int ReadData(  
  
    void* handle,  
  
    unsigned char* readData,  
  
    unsigned int readNum,  
  
    unsigned int* preadedNum  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*unsigned char\* readData*

[in] Printer data that needs to be read.

*unsigned int readNum*

[in] The length of data that needs to be read.

*unsigned int\* preadedNum*

[in] The length of the data actually read.

### Return Value:

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_READ_FAILED	-331	Read failed
E_IO_READ_TIMEOUT	-332	Read timeout

## 4.8 DirectIO

This function is for the user to customize the data sent and read by the printer. If some functions do not provide a function interface, the user can send command data to the printer through this interface.

```
int DirectIO(  
  
    void* handle,  
  
    unsigned char* writedata,  
  
    unsigned int writeNum,  
  
    unsigned char* readdata,  
  
    unsigned int readNum,  
  
    unsigned int* preadedNum  
  
    );
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*unsigned char\* writedata*

[in] The data written to the printer.

*unsigned int writeNum*

[in] The length of the data written to the printer. When writeNum=0, the write data operation is not performed.

*unsigned char\* readdata*

[in,out] Get the data returned by the printer.

*unsigned int readNum*

[in] Preset the length of data that needs to be read. When readNum=0, the read data operation is not performed.

*unsigned int\* preadedNum*

[in,out] The length of the data actually read.

**Return Value:**

Error Code	Value	Description
E_SUCCESS	0	Normal
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout
E_IO_READ_FAILED	-331	Read failed
E_IO_READ_TIMEOUT	-332	Read timeout

## 4.9 TSPL\_SelfTest

This function is to print a self-test page for the printer, which contains basic configuration information for the printer. This function needs to be called after TSPL\_Print.

```
int TSPL_SelfTest(  
  
    void* handle  
  
);
```

### Parameter:

*void\* handle*  
[in,out] The created target printer object.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.10 TSPL\_PrintImage

This function is for printing the specified image (supports bmp, jpg, gif, etc), this function is suitable for printing LOGO.

Convert the image of the specified path to bitmap data and send it to the printer and print.

```
int TSPL_PrintImage(  
  
    void* handle,  
  
    int xPos,  
  
    int yPos,  
  
    int mode,  
  
    const TCHAR* filePath  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] Horizontal starting position(range:0-32000, unit:dot).

*int yPos*

[in] Vertical starting position(range:0-32000, unit:dot).

*int mode*

[in]Printer image mode.

Mode	Value
OVERWRITE	0
OR	1
XOR	2

*const TCHAR\* filePath*

[in]The correct path to the picture.

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	This model does not support this feature.
E_BAD_HANDLE	-6	Invalid handle
E_NOT_ENOUGH_MEMORY	-9	The computer does not have enough memory
E_IMAGE_BAD_SIZE	-25	Image size error
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout



## 4.11 TSPL\_Setup

This function is to set for the printer's basic parameters.

```
int TSPL_Setup(  
    void* handle,  
    int labelWidth,  
    int labelHeight,  
    int speed,  
    int density,  
    int type,  
    int gap,  
    int offset  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int labelWidth*

[in] Set the preprinted label width (unit: mm).

*int labelHeight*

[in] Set the preprinted label length (unit: mm).

*int speed*

[in] Set the printer printing speed (range: 2-4 for normal models and 2-7 for high-speed models, please refer to the corresponding product specifications for specific specifications).

*int density*

[in] Set the printer print density (range: 0-15).

*int type*

[in] Set the label paper type (0: black mark / continuous paper 1: stitch mark / continuous paper).

*int gap*

[in] The spacing between the two labels, if set to 0, indicates that the current is continuous paper.

*int offset*

[in] The position where each label stops when the peel mode is set.

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.12 TSPL\_ClearBuffer

This function is to clear the printer memory cache. Clear the printer cache before executing print data.

**int TSPL\_ClearBuffer(**

**void\* handle**

**);**

### Parameter:

*void\* handle*

[in,out] The created target printer object.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.13 TSPL\_Box

This function is to print a line or rectangle.

```
int TSPL_Box(  
    void* handle,  
    int x_start,  
    int y_start,  
    int x_end,  
    int y_end,  
    int thickness,  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int x\_start*

[in] Horizontal starting position (range:0-32000, unit:dot).

*int y\_start*

[in] Vertical starting position (range:0-32000, unit:dot).

*int x\_end*

[in] Horizontal end position (range:0-32000, unit:dot).

*int y\_end*

[in] Vertical end position (range:0-32000, unit:dot).

*int thickness*

[in] Line width(unit:dot).

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.14 TSPL\_BarCode

This function is to print one-dimensional barcodes.

```
int TSPL_BarCode(  
    void* handle,  
    int xPos,  
    int yPos,  
    int type,  
    int height,  
    int readable,  
    int rotation,  
    int narrow,  
    int wide,  
    const TCHAR* data  
);
```

### Parameter:

*void\* handle*

[in,out]The created target printer object.

*const int xPos*

[in] Horizontal starting position(range:0-32000, unit:dot).

*const int yPos*

[in] Vertical starting position(range:0-32000, unit:dot).

*const int type*

[in] Set the barcode type.

Barcode type	Value
Code 128	0
Code 128M	1
EAN 128	2
Interleaved 2 of 5	3
Interleaved 2 of 5 with check digits	4
Code 39	5
Code 39C	6
Code 39S	7
Code 93	8
EAN 13	9
EAN 13 with 2 digits add-on	10
EAN 13 with 5 digits add-on	11
EAN 8	12
EAN 8 with 2 digits add-on	13
EAN 8 with 5 digits add-on	14
Codabar	15
Postnet	16
UPC-A	17
UPC-A with 2 digits add-on	18
UPC-A with 5 digits add-on	19
UPC-E	20
UPC-E with 2 digits add-on	21
UPC-E with 5 digits add-on	22
China post code	23
MSI code	24
MSI with check digit	25
PLESSEY code	26
ITF 14 code	27
EAN 14 code	28
Code 11	29

*int height*

[in] Set the bar code print height(expressed in dots).

*int readable*

[in] The settings are readable (0: unreadable, 1: readable).

*int rotation*

[in] Set the barcode direction.

Barcode direction	Value
Not rotating	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int narrow*

[in] Set the narrow Bar width, represented by dots.

*int wide*

[in] Set the width of the wide Bar, represented by dots.

*const TCHAR\* data*

[in] Bar code data.

**Return value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout



## 4.15 TSPL\_QrCode

This function is to print a QR code.

```
int TSPL_QrCode(  
    void* handle,  
    int xPos,  
    int yPos,  
    int eccLevel,  
    int width,  
    int mode,  
    int rotation,  
    int model,  
    int mask,  
    const TCHAR* data  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] horizontal starting position(range:0-32000, unit:dot).

*int yPos*

[in] vertical starting position(range:0-32000, unit:dot).

*int eccLevel*

[in] Error correction level.

Error tolerant Rate	Value
L(7%)	0
M(15%)	1
Q(25%)	2
H(30%)	3

*int width*

[in] Set the QR code print width (range: 1-10).

*int mode*

[in] Automatic / manual coding (0: automatic, 1: manual).

*int rotation*

[in] Set the barcode direction.

Barcode direction	Value
Not rotating	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int model*

[in] Set the QR code version (0: Basic, 1: Enhanced).

*int mask*

[in] Mask mode(range: 0-8).

*const TCHAR\* data*

[in] QR code data

#### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.16 TSPL\_Text

This function is to print text for the printer.

```
int TSPL_Text(  
    void* handle,  
    int xPos,  
    int yPos,  
    int font,  
    int rotation,  
    int xMultiplication,  
    int yMultiplication ,  
    const TCHAR* data  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] Horizontal starting position (range: 0-32000, unit: dot).

*int yPos*

[in] Vertical starting position (range: 0-32000, unit: dot).

*int font*

[in] font.

Font lattice	Value
normal	0
8x12	1
12x20	2
16x24	3
24x32	4
32x48	5
14x19	6
21x27	7
14x25	8
Chinese	9

*int rotation*

[in] Set the print direction.

Print direction	Value
Not rotating	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int xMultiplication*

[in] Font width magnification.

*int yMultiplication*

[in] Font height magnification.

*const TCHAR\* data*

[in] Text data.

#### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.17 TSPL\_Print

This function is to perform printing operations.

```
int TSPL_Print(  
  
    void* handle,  
  
    int num,  
  
    int copies  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int num*

[in] The number of different labels.

*int copies*

[in] Print the number of copies of each label.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.18 TSPL\_FormFeed

This function is to perform a paper feed operation for the printer and navigate to the next label.

**int TSPL\_FormFeed(**

**void\* *handle***

**);**

### Parameter:

*void\* handle*

[in,out] The created target printer object.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.19 TSPL\_SetTear

This function is to enable/disable the tear-off function for setting the printer.

```
int TSPL_SetTear(
```

```
    void* handle,
```

```
    int on_off
```

```
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object .

*int on\_off*

[in] Enable/disable the tear-off function (0: OFF, 1: ON).

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.20 TSPL\_SetRibbon

This function is to enable/disable the sensor detection.

```
int TSPL_SetRibbon(
```

```
    void* handle,
```

```
    int on_off
```

```
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int on\_off*

[in] Enable/disable sensor detection: (0: OFF, 1: ON).

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout



## 4.21 TSPL\_Offset

This function is to define an extra length for each label of the printer.

```
int TSPL_Offset(  
  
    void* handle,  
  
    int distance  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int distance*

[in] Specify the paper feed length (unit: mm).

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.22 TSPL\_Direction

This function is to set the printer printing direction.

```
int TSPL_Direction(  
  
    void* handle,  
  
    int direction  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int direction*

[in] Print direction (0: normal direction, non-zero: 180 degrees).

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.23 TSPL\_Feed

This function is to specify the length of the printer.

```
int TSPL_Feed(  
  
    void* handle,  
  
    int len  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int len*

[in] Specify the paper length (unit: dot).

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.24 TSPL\_Home

This function is to locate labels for the printer.

```
int TSPL_Home(  
  
    void* handle  
  
);
```

### Parameter:

*void\* handle*  
[in,out] The created target printer object.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.25 TSPL\_Learn

This function is to learn labels for printers.

```
int TSPL_Learn(  
  
    void* handle  
  
);
```

### Parameter:

*void\* handle*  
[in,out] The created target printer object.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.26 TSPL\_GetSN

This function is to get the printer SN serial number.

```
int TSPL_GetSN(  
  
    void* handle,  
  
    char* sn  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*char\* sn*

[in] Get the SN serial number.

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.27 TSPL\_GetPrinterStatus

This function is to get the printer status.

```
int TSPL_GetPrinterStatus(
```

```
    void* handle,
```

```
    int* status
```

```
);
```

**parameter:**

*void\* handle*

[in,out] The created target printer object.

*int\* status*

[in,out] The current status of the printer. Accumulated value when there are multiple states

Status	Value
Normal	0
The print head is opened	1
Paper jam	2
Out of paper	4
Out of ribbon	8
Print pause	16
Printing	32
Cover opened	64
Other error	128

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.28 TSPL\_SetCodePage

This function is to set the code page for the printer.

```
int TSPL_SetCodePage(
```

```
    void* handle,
```

```
    char* codepage
```

```
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*Char\* codepage*

[in] International character set.

7-bit code page		8-bit code page		Windows code page		ISO code page	
US A	USA	437	USA	1250	Central Europe	8859-1	Latin 1
BRI	British	737	Greek	1251	Cyrillic	8859-2	Latin 2
GE R	German	850	Multilingual	1252	Latin I	8859-3	Latin 3
FRE	French	851	Greek1	1253	Greek	8859-4	Baltic
DA N	Danish	852	Slavic	1254	Turkish	8859-5	Cyrillic
ITA	Italian	855	Cyrillic	1255	Hebrew	8859-6	Arabic
SPA	Spanish	857	Turkish	1256	Arabic	8859-7	Greek
SW E	Swedish	860	Portuguese	1257	Baltic	8859-8	Hebrew
SWI	Swiss	861	Icelandic	1258	Vietnam	8859-9	Turkish
		862	Hebrew	932	Japanese shift-JIS	8859-10	Latin 6
		863	Canadian/French	936	Simplified	8859-15	Latin 9
		864	Arabic	949	Korean		
		865	Nordic	950	Traditional Chinese BIG5		
		866	Russian	UTF-8	UTF 8		
		869	Greek2				



**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.29 TSPL\_PDF417

This function is to print the PDF417 QR code.

```
int TSPL_PDF417(  
  
    void* handle,  
  
    int xPos,  
  
    int yPos,  
  
    int width,  
  
    int height,  
  
    int rotate,  
  
    const TCHAR* option,  
  
    const TCHAR* data  
  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] Horizontal starting position (range: 0-32000, unit: dot).

*int yPos*

[in] Vertical starting position (range: 0-32000, unit: dot).

*int width*

[in] Bar code width (unit: dot).

*int height*

[in] Bar code height (unit: dot).

*int rotate*

[in] Set the barcode direction.

Barcode direction	Value
Not rotating	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*const TCHAR\* option*

[in] Optional.example: P1, M1, U1,1,1, W2, H4, R1, C1, T1, L1

P	Data compression method: 0: automatic 1: binary mode
E	Error check level (0~8)
M	Bar code center print mode 0: This mode will print in the upper left corner alignment area 1: will print in the middle area
Ux,y,z	Code readable x: the x coordinate specified by the readable character y: the y coordinate specified by the readable character c: maximum number of readable characters per line
W	Module width (2~9: dot)
H	Height of small bar code (4~99: dot)
R	Maximum number of rows
C	Maximum number of columns
T	Whether to cut off 0: No 1: Yes
Lm	Indicates the length (1~2048: mm)

*const TCHAR\* data*

[in] QR code data.

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.30 TSPL\_Block

This function is to print the contents of the segment into the label.

```
int TSPL_Block(  
    void* handle,  
    int xPos,  
    int yPos,  
    int width,  
    int height,  
    int font,  
    int rotate,  
    int x_multiplication,  
    int y_multiplication,  
    int space,  
    int alginment,  
    const char* data  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] Horizontal starting position (range: 0-32000, unit: dot).

*int yPos*

[in] Vertical starting position (range: 0-32000, unit: dot).

*int width*

[in] Bar code width (unit: dot).

*int height*

[in] Bar code height (unit: dot).

*int font*

[in] font.

Font lattice	Value
normal	0
8x12	1
12x20	2
16x24	3
24x32	4
32x48	5
14x19	6
21x27	7
14x25	8
Chinese	9

*int rotate*

[in] Set the barcode direction.

Barcode direction	Value
Not rotating	0
Rotate 90 degrees	1
Rotate 180 degrees	2
Rotate 270 degrees	3

*int x\_multiplication*

[in] Horizontal font magnification (1~10).

*int y\_multiplication*

[in] Vertical font magnification (1~10).

*int space*

[in] Set the distance between the lines (unit: dot).

*int alginment*

[in] Set the text alignment.

Alignment	Value
DEFAULT	0
ALIGNMENT_LEFT	1
ALIGNMENT_CENTER	2
ALIGNMENT_RIGHT	3

*const char\* data*

[in] Content data. The maximum is 4092 bytes.

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.31 TSPL\_Reverse

This function is to set the print content in the specified area as reversed, i.e. the blank portion is black, and the pre-printed portion is white.

**int TSPL\_Reverse (**

**void\*** *handle*,

**int** *xPos*,

**int** *yPos*,

**int** *width*,

**int** *height*

**);**

**Parameter:**

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] Horizontal starting position (range: 0-32000, unit: dot).

*int yPos*

[in] Vertical starting position (range: 0-32000, unit: dot).

*int width*

[in] Horizontal width

*int height*

[in] Vertical width

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout



## 4.32 TSPL\_GapDetect

This function is to determine the size of the label and the size of the spacing between the two labels.

```
int TSPL_GapDetect(  
  
    void* handle,  
  
    int paper_length,  
  
    int gap_length  
);
```

### Parameter:

*void\* handle*

[in,out] The created target printer object.

*int paper\_length*

[in] Paper length (range: 0-32000, unit: dot).

*int gap\_length*

[in] Gap length (range: 0-32000, unit: dot).

### Return Value:

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	Invalid model type
E_BAD_HANDLE	-6	Invalid handle
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout

## 4.33 TSPL\_BitMap

This function functions to print a bitmap image。

```
int TSPL_BitMap(  
    void* handle,  
    int xPos,  
    int yPos,  
    int mode,  
    int width,  
    int height  
    const TCHAR* data  
);
```

参数:

*void\* handle*

[in,out] The created target printer object.

*int xPos*

[in] Horizontal starting position (range: 0-32000, unit: dot).

*int yPos*

[in] Vertical starting position (range: 0-32000, unit: dot).

*int mode*

[in]Printer image mode.

Mode	Value
OVERWRITE	0
OR	1
XOR	2

*int width*

[in] Horizontal width (unit: byte).

*int height*

[in] Vertical width(unit: dot).

*const TCHAR\* data*  
[in]Image data.

**Return Value:**

Error code	Value	Description
E_SUCCESS	0	Normal
E_INVALID_PARAMETER	-1	Invalid parameter
E_NOT_ENOUGH_BUFFER	-2	Not enough buffer
E_INVALID_MODEL_TYPE	-3	This model does not support this feature.
E_BAD_HANDLE	-6	Invalid handle
E_NOT_ENOUGH_MEMORY	-9	The computer does not have enough memory
E_IMAGE_BAD_SIZE	-25	Image size error
E_IO_PORT_NOT_OPEN	-309	Communication port not open
E_IO_WRITE_FAILED	-321	Write failed
E_IO_WRITE_TIMEOUT	-322	Write timeout