Instructions for libtsc.so library functions

❖ System requirement:

➢ C/C++ compiler requirement

- GCC Version newer than **5.4.0**
  - GCC version 5.4.0 is already installed in Ubuntu 16.04 LTS.

➢ Package requirement

- Make sure the following packages are installed on Linux system:

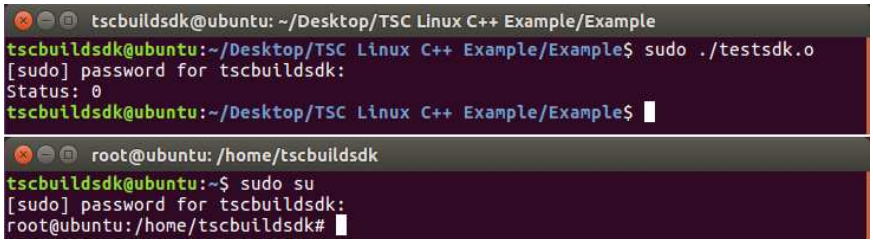
  libusb-1.0

  libsdl2

  libsdl2-ttf
  - In Ubuntu 16.04 LTS, you can install these packages via following command:
    - **sudo apt-get install libusb-1.0-0-dev**
    - **sudo apt-get install libsdl2-dev**
    - **sudo apt-get install libsdl2-ttf-dev**

➢ Permission requirement

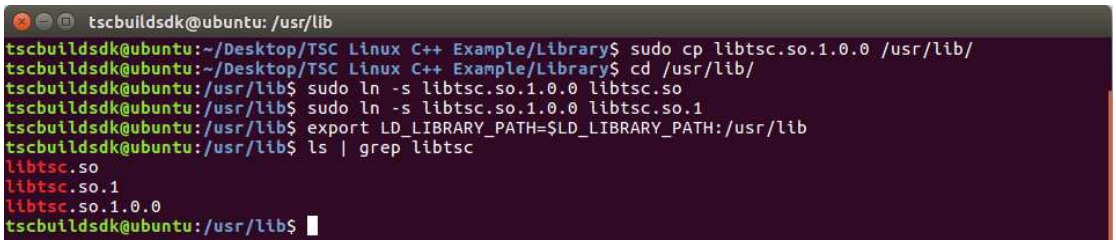
- For USB interface, software developed by this shared library must have **permission to write/read** TSC printers.
  - For System Administrator, use **sudo** command to execute the software as **Root Privileges**, or use **sudo su** command to switch to **Root Account**.



  - For User Accounts that cannot gain **Root Privileges**, System Administrator can add **udev rules** to enable USB permission of TSC printers by following step:
    1. Copy **90-usbpermission.rules** to **/etc/udev/rules.d** folder.
    2. Use following command to load the rule.
       - **sudo chmod 777 /etc/udev/rules.d/90-usbpermission.rules**
       - **sudo udevadm control --reload-rules**
       - **sudo service udev restart**
       - **sudo udevadm trigger**

❖ **Install TSC library:**

1. Copy **libtsc.so.1.0.0** to /**usr**/**lib**/ folder
2. Open terminal, use **cd** command to access /**usr**/**lib**/ folder
3. Use following command to link library:
   - **sudo ln -s libtsc.so.1.0.0 libtsc.so**
   - **sudo ln -s libtsc.so.1.0.0 libtsc.so.1**
   - **export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib**



*Note: Library folder include 3 files.

❖ **Functions:**

1. openport(a)

Description: Start the printer spool.

Parameter:

a: String

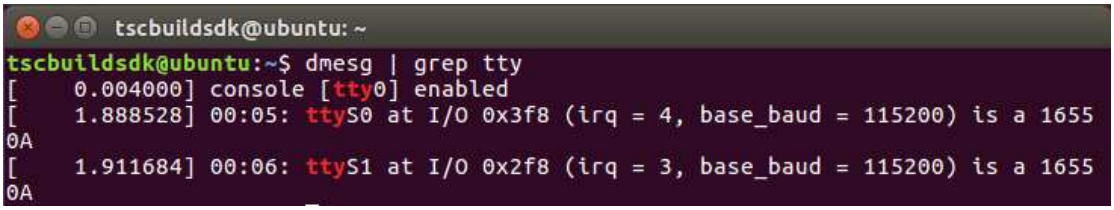
(1) For USB interface directly, please specify USB.

Like: **"USB"**

(2) For serial port devices, please specify tty name.

Like: **"ttyS0"**

You can use "**dmesg** | **grep tty**" command to list the tty devices.



2. openethernet(a, b)

Description: Start the printer spool.

Parameter:

a: String, IP Address, for example **"192.168.0.100"**

b: Integer, printer port number, for example **9100**

3. closeport()

Description: Close Windows printer spool.

Parameter: None

4. setup(a,b,c,d,e,f,g)

Description: Set up label width, label height, print speed, print density, sensor type, gap/black mark vertical distance、gap/black mark shift distance

Parameter:

a: string, sets up label width; unit: mm

b: string, sets up label height; unit: mm

c: string, sets up print speed, (selectable print speeds vary on different printer models)

1.0: sets print speed at 1.0"/sec

1.5: sets print speed at 1.5"/sec

2.0: sets print speed at 2.0"/sec

3.0: sets print speed at 3.0"/sec

4.0: sets print speed at 4.0"/sec

6.0: sets print speed at 6.0"/sec

8.0: sets print speed at 8.0"/sec

10.0: sets print speed at 10.0"/sec

12.0: sets print speed at 12.0"/sec

d: string, sets up print density

0~15，the greater the number, the darker the printing

e: string, sets up the sensor type to be used

0: signifies that vertical gap sensor is to be used

1: signifies that black mark sensor is to be used

f: string, sets up vertical gap height of the gap/black mark; unit: mm

g: string, sets up shift distance of the gap/black mark; unit:: mm; in the case of the average label, set this parameter to be 0.

5. clearbuffer()

Description: Clear

Parameter: None

6. barcode(a,b,c,d,e,f,g,h,I)

Description: Use built-in bar code formats to print

Parameter:

a: string; the starting point of the bar code along the X direction, given in points (of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1point=1/12 mm)

b: string; the starting point of the bar code along the Y direction, given in points (of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1 point=1/12 mm)

c: string

128	Code 128, switching code subset A, B, C automatically
128M	Code 128, switching code subset A, B, C manually.
EAN128	Code 128, switching code subset A, B, C automatically
25	Interleaved 2 of 5
25C	Interleaved 2 of 5 with check digits
39	Code 39
39C	Code 39 with check digits
93	Code 93
EAN13	EAN 13
EAN13+2	EAN 13 with 2 digits add-on
EAN13+5	EAN 13 with 5 digits add-on
EAN8	EAN 8
EAN8+2	EAN 8 with 2 digits add-on

EAN8+5	EAN 8 with 5 digits add-on
CODA	Codabar
POST	Postnet
UPCA	UPC-A
UPCA+2	UPC-A with 2 digits add-on
UPCA+5	UPC-A with 5 digits add-on
UPCE	UPC-E
UPCE+2	UPC-E with 2 digits add-on
UPCE+5	UPC-E with 5 digits add-on

d: string; sets up bar code height, given in points
e: string, sets up whether to print human recognizable interpretation (text) or not.
0: prints no interpretation
1: prints interpretation
f: string; sets up rotation degrees
0: rotates 0 degree
90: rotates 90 degrees
180: rotates180 degrees
270: rotates 270 degrees
g: string; sets up narrow bar ratio, refer to TSPL user's manual
h: string; sets up wide bar ratio, refer to TSPL user's manual
I: string; bar code content

7. printerfont(a,b,c,d,e,f,g)

Description: Use printer built-in fonts to print
Parameter:
a: string; the starting point of text (character string) along the X direction, given in points
(of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1 point=1/12 mm)
b: string; the starting point of text (character string) along the Y direction, given in points
(of 200 DPI, 1 point=1/8 mm; of 300 DPI, 1 point=1/12 mm)
c: string; built-in font type name, 12 kinds in sum
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

TTT24.BF2: Traditional Chinese 24*24 (Telecommunication Code)
TSS24.BF2: Simplified Chinese 24*24
TSS16.BF2: Simplified Chinese 16*16
K: Japan, Korean font 24*24,
L: Japan Korean font 16*16

d: string; sets up the rotation degree of the text (character string)
0: rotates 0 degree
90: rotate 90 degrees
180: rotate 180 degrees
270: rotate 270 degrees

e: string; sets up the magnification rate of text (character string) along the X direction, range: 1~8

f: string; sets up the magnification rate of text (character string) along the Y direction, range: 1~8

g: string; prints the content of text (character string)

8. sendcommand(command)

Description: Sends built-in commands to the bar code printer

Parameter: Refer to TSPL for details

9. printlabel(a,b)

Description: Print label content

Parameter:

a: string; sets up the number of label sets

b: string, sets up the number of print copies

10. downloadpcx(a,b)

Description: Download mono PCX graphic files to the printer

Parameter:

a: string; file name (including file path)

Like: **"/home/tsc/UL.PCX"**

b: string, names of files that are to be downloaded in the printer memory (Please use capital letters)

11. formfeed()

Description: Skip to next page (of label); this function is to be used after setup

Parameter: None

12. nobackfeed()

Description: disable the backfeed function

Parameter: None

13. windowsfont(a,b,c,d,e,f)

Description: Use Windows font to print text.

Parameter:

a: Integer, the starting point of the text along the X direction, given in points

b: Integer, the starting point of the text along the Y direction, given in points

c: Integer, the font height, given in points.

d: Integer, rotation in counter clockwise direction

0 -> 0 degree

90-> 90 degree

180-> 180 degree

270-> 270 degree

e: String, font name (including file path).

Like: **"/home/tsc/Arial.ttf"**

f: String, text to be printed.

14. about()

Description: Return DLL version.

Parameter: None

15. sendBinaryData(a,b)

Description: Send binary data to printer.

a: Hex, Hex data

b: Integer, Hex data length

16. printerstatus()

Description: Return printer status. Please refer to TSPL manual <ESC>!? command.

Parameter: None

17. printername()

Description: Return printer model name

Parameter: None

18. printerserial()

Description: Return printer serial number.

Parameter: None

Examples for C++

```
#include <stdio.h>
#include "libtsc.h"

int main(int argc, char *argv[]) {

	openport("USB");
	//openport("ttyS1");
	//openethernet("192.168.0.100", 9100);

	printf("Status: %d\r\n",printerstatus());

	sendcommand("SIZE 100 mm, 63 mm");
	sendcommand("GAP 2 mm, 0 mm");
	sendcommand("SPEED 4");
	sendcommand("DENSITY 8");
	sendcommand("DIRECTION 1");
	clearbuffer();
	barcode("100", "150", "128", "100", "1", "0", "2", "2", "Barcode Test");
	printerfont("100", "300", "3", "0", "1", "1", "Print Font Test");
	windowsfont(100, 50, 50, 0, "ARIALUNI.TTF", "にっぽんご English 中文한글");
	downloadpcx("UL.PCX", "UL.PCX");
	sendcommand("PUTPCX 100,400,\"UL.PCX\"");
	printlabel("1", "1");

	closeport();

	return 0;
}
```

- Use following command to build and test the example by terminal:
  - **sudo g++ -w testsdk.cpp -ltsc -o testsdk.o**
  - **sudo chmod +x testsdk.o**
  - **./testsdk.o**

```
tscbuildsdk@ubuntu: ~/Desktop/TSC Linux C++ Example/Example
tscbuildsdk@ubuntu:~/Desktop/TSC Linux C++ Example/Example$ sudo g++ -w testsdk.cpp -ltsc -o testsdk.o
tscbuildsdk@ubuntu:~/Desktop/TSC Linux C++ Example/Example$ sudo chmod +x testsdk.o
tscbuildsdk@ubuntu:~/Desktop/TSC Linux C++ Example/Example$ ./testsdk.o
Status: 0
tscbuildsdk@ubuntu:~/Desktop/TSC Linux C++ Example/Example$
```

- Result: