

# TSPL Printer SDK for Linux 说明文档



修订历史

修订日期	修订版本	修改描述	编辑	审核
2021-11-10	A0	建立文档		
2022-10-08	A1	v1.0.4		



## 目录

1. 系统需求 .....	1
1.1 操作系统和C/C++编译器 .....	1
1.2 依赖软件包 .....	1
1.3 权限要求 .....	1
2. SDK库安装 .....	2
3. Demo编译和运行 .....	2
4. 函数说明 .....	2



## 1. 系统需求

### 1.1 操作系统和C/C++编译器

系统：Ubuntu 16 / CentOS 8 x86-64 或以上版本

编译器：GCC 8.3.0 或以上版本

### 1.2 deb软件包安装

Ubuntu系统安装：

```
sudo dpkg -i gainscha-tspl-sdk-1.0.4.deb
```

CentOS暂无安装包提供。可以参照第二节手工安装库文件。

### 1.3 权限要求

USB设备连接需要特定权限才能正常访问。

(1) 如果是系统管理员

可以直接使用sudo获取root权限，并运行您的应用程序。

比如运行demo程序：

```
sudo ./testsdk
```

(2) 如果是普通用户（Ubuntu系统安装deb包则可跳过此步骤）

需要让管理员安装udev rules文件，让普通用户拥有访问USB打印机的权限。

执行以下命令进行安装：

```
sudo cp 91-usbpermission.rules /etc/udev/rules.d/  
sudo chmod 777 /etc/udev/rules.d/91-usbpermission.rules  
sudo udevadm control --reload-rules  
sudo systemctl restart systemd-udev  
sudo udevadm trigger
```

## 2. SDK库安装 (Ubuntu系统安装deb包则可跳过此步骤)

2.1 动态库: 将libtspl.so复制到/usr/lib64/下  
(注意这是64位库文件)

```
sudo cp libtspl.so /usr/lib64/  
ls -l /usr/lib | grep tspl
```

2.2 头文件: 将libtspl.h复制到/usr/include/下,  
或直接放到软件源码工程项目中.

```
sudo cp libtspl.h /usr/include/  
ls -l /usr/include | grep tspl
```

## 3. Demo编译和运行

3.1 在demo路径下, 直接执行g++编译指令后, 生成testsdk程序.  
在确保安装libtspl.so后可以直接运行testsdk查看输出效果。(或者设定LD\_LIBRARY\_PATH路径, 在此路径下能找到libtspl也可以正常运行)

```
cd demo  
g++ testsdk.cpp -o testsdk -ltspl  
./testsdk
```

## 4. 函数说明

### 4.1 openport

函数	int openport(const char * printrname);	
描述	连接USB打印机或串口打印机	
参数1	printrname	字符串
说明	(1) 如果是连接USB打印机, 那么参数为"USB" (2) 如果是连接串口打印机, 那么参数为"ttyS0"或"ttyS1"等, 具体要根据实际连接的串口号来决定。	
返回值	失败: 0	

	成功：非0
示例	<code>openport(“USB”);</code> 或 <code>openport(“ttyS0”);</code>

4.2 openethernet

函数	<code>int openethernet(const char * ipaddress, int portnumber);</code>	
描述	连接网络打印机	
参数1	<code>ipaddress</code>	字符串
参数2	<code>portnumber</code>	整型
说明	<code>ipaddress</code> 为打印机IP地址 <code>portnumber</code> 一般为9100	
返回值	失败： 0	
	成功：非0	
示例	<code>openport(“192.168.123.100”, 9100);</code>	

4.3 closeport

函数	<code>int closeport(void);</code>	
描述	关闭打印机连接.	
参数	无	
说明	如未连接打印机，则返回失败.	
返回值	失败： 0	
	成功：非0	
示例	<code>closeport();</code>	

4.4 sendcommand

函数	<code>int sendcommand(const char * printercommand);</code>	
描述	发送文本指令给打印机.	
参数1	<code>printercommand</code>	字符串
说明	发送指令，该函数在指令结尾自动添加回车换行“\r\n”	
返回值	失败： 0	
	成功：非0	



示例	sendcommand("DIRECTION 1");
----	-----------------------------

4.5 setup

函数	int setup(const char *width, const char *height, const char *speed, const char *density, const char *sensor, const char *vertical, const char *offset);		
描述	设置标签规格		
参数1	width	字符串	标签宽度，单位毫米
参数2	height	字符串	标签高度，单位毫米
参数3	speed	字符串	打印速度，单位寸/秒。值为"2"， "3"等(参考打印机手册)
参数4	density	字符串	打印浓度。值为"0"~"15"(参考打 印机手册)
参数5	sensor	字符串	传感器类型。值"1"为黑标，其他 为普通标签纸。可设为NULL。
参数6	vertical	字符串	传感器参数，黑标或间隙纵向宽 度，如间隙纸为"2"，连续纸为"0"
参数7	offset	字符串	传感器参数，偏移量，一般设为 "0"
返回值	失败： 0		
	成功：非0		
示例	setup("76", "80", "4", "8", "0", "0", "0");		

4.6 clearbuffer

函数	int clearbuffer(void);		
描述	清空标签内容		
参数	无		
返回值	失败： 0		
	成功：非0		
示例	clearbuffer();		

4.7 barcode

函数	int barcode(const char *x, const char *y, const char *type,
----	---

	<code>const char *height, const char *readable, const char *rotation, const char *narrow, const char *wide, const char *code);</code>		
描述	绘制条码		
参数1	x	字符串	X坐标, 单位像素点
参数2	y	字符串	Y坐标, 单位像素点
参数3	type	字符串	条码类型(参考打印机手册), 如 "128", "EAN13"等
参数4	height	字符串	条码高度, 单位像素点
参数5	readable	字符串	是否显示可读字符串, "1"可识, "0"不可
参数6	rotation	字符串	旋转, "0"不旋转, "90"顺时针旋转 90度, 其他"180", "270"等
参数7	narrow	字符串	窄bar宽度, 单位像素点, 如"1", "2"
参数8	wide	字符串	宽bar宽度, 单位像素点, 如"2", "4"
参数9	code	字符串	条码内容
返回值	失败: 0		
	成功: 非0		
示例	<code>barcode("100", "150", "128", "100", "1", "0", "2", "2", "Barcode Test");</code>		

4.8 printerfont

函数	<code>int printerfont(const char *x, const char *y, const char *fonttype, const char *rotation, const char *xmul, const char *ymul, const char *text);</code>		
描述	绘制条码		
参数1	x	字符串	X坐标, 单位像素点
参数2	y	字符串	Y坐标, 单位像素点
参数3	fonttype	字符串	字体类型(参考打印机手册), 如 "1": 8*12字体 "2": 12*20字体 ... "TSS24.BF2": 简体中文字体
参数4	rotation	字符串	旋转, "0"不旋转, "90"顺时针旋转 90度, 其他"180", "270"等
参数5	xmul	字符串	X方向放大倍数, 如"1", "2"等
参数6	ymul	字符串	Y方向放大倍数, 如"1", "2"等
参数7	text	字符串	要打印的文字内容



返回值	失败： 0		
	成功：非0		
示例	printerfont("100", "300", "3", "0", "1", "1", "Print Font Test");		

4.9 printlabel

函数	int printlabel(const char *set, const char *copy);		
描述	打印标签		
参数1	set	字符串	打印份数，一般为"1"
参数2	copy	字符串	每份重复打印次数，一般为"1"
说明	标签内容设定好后，必须调用printlabel()才能打印。		
返回值	失败： 0		
	成功：非0		
示例	printlabel("1", "1");		

4.10 downloadpcx

函数	int downloadpcx(const char *filename, const char *image_name);		
描述	保存文件至打印机		
参数1	filename	字符串	本地文件名
参数2	image_name	字符串	打印内存储文件名
说明	保存文件后，可以在后续指令中调用，用于打印。		
返回值	失败： 0		
	成功：非0		
示例	downloadpcx("UL.PCX", "UL.PCX");		

4.11 formfeed

函数	int formfeed(void);		
描述	控制打印机进一张纸		
参数	无		
返回值	失败： 0		



	成功：非0
示例	formfeed();

4.12 nobackfeed

函数	int nobackfeed(void);
描述	禁止走纸到撕纸处
参数	无
返回值	失败： 0
	成功：非0
示例	nobackfeed();

4.13 windowsfont

函数	int windowsfont(int x, int y, int fontheight, int rotation, const char *fontpath, const char *content);		
描述	绘制条码		
参数1	x	字符串	X坐标, 单位像素点
参数2	y	字符串	Y坐标, 单位像素点
参数3	fontheight	字符串	文字高度, TTF字体高度
参数4	rotation	字符串	旋转(暂不支持), 请设为"0"
参数5	fontpath	字符串	TTF文件路径和文件名
参数6	content	字符串	要打印的文字内容
返回值	失败： 0		
	成功：非0		
示例	printerfont("100", "300", "3", "0", "1", "1", "Print Font Test");		

4.14 about

函数	char* about(void);
描述	获取本SDK的版本信息
参数	无
返回值	SDK信息字符串。
示例	printf("SDK info: %s\n", about());



4.15 sendBinaryData

函数	int sendBinaryData(const void* binaryData, int dataLength);		
描述	发送二进制数据到打印机		
参数1	binaryData	指针	数据指针
参数2	dataLength	整型	数据长度
返回值	失败： 0		
	成功：非0		
说明	用于特殊指令发送，也可以用于发送普通指令。		
示例	sendBinaryData("CLS\r\n", 5);		

4.16 printerstatus

函数	int printerstatus(void);
描述	获取打印机状态
参数	无
返回值	打印机状态
示例	int status = printerstatus(); printf("printer status : 0x%x\n", status);

4.17 printername

函数	const char* printername(void);
描述	获取打印机型号名称
参数	无
返回值	打印机型号
示例	const char * model = printername(); printf("printer model : %s\n", model);

