

美松科技打印机产品 SDK  
参考手册

深圳市美松科技有限公司  
软件研发部

修订历史				
版本	说明	作者	审核	修订日期
2.1.0.0	建立	易军干		2018-06-12
2.1.0.1	接口增加互斥锁	易军干		2018-07-24
2.1.0.2	增加流控设置, SetPrintportFlowCtrl	易军干		2018-08-01
2.2.0.0	增加对蓝牙打印的支持 SetPrintConn 增加打印 Data Matrix 码接口 PrintDataMatrix (目前只有 DLL 有此功能)	易军干		2018-12-11
	增加 PrintQRcode500II	易军干		201-03-01
	增加 PrintQRcodeII	易军干		201-03-01
	SetDevname 优化增加自动匹配 USB 接口功能, 能大幅提升打印速度	易军干		201-03-01

## 目 录

1. SDK 介绍.....	6
1.1 概述.....	6
1.2 本手册的预期读者.....	6
1.3 支持机型.....	6
1.4 支持连接方式.....	6
1.5 SDK 文件.....	6
2. 打印接口说明.....	7
2.1 打印机初始化和结束.....	7
2.1.1 设置设备名称(Windows).....	7
2.1.2 设置设备连接(Windows).....	8
2.1.3 设置设备名称(Linux).....	9
2.1.4 设置流控.....	10
2.1.5 自动配置 USB 端口.....	11
2.1.6 自动配置串口.....	11
2.1.7 打印机初始化.....	12
2.1.8 清理缓存.....	13
2.1.9 打印机关闭.....	13
2.1.10 设置汉字模式.....	14
2.1.11 设置区域国家和代码页.....	14
2.2 常用打印功能.....	16
2.2.1 打印字符串.....	16
2.2.2 打印切纸.....	17
2.2.3 打印并换行.....	17
2.2.4 打印细走纸.....	18
2.2.5 打印自检页.....	18
2.3 打印效果.....	19
2.3.1 设置行间距.....	19
2.3.2 设置字符间距.....	19
2.3.3 设置左边界.....	20
2.3.4 设置字符放大.....	20
2.3.5 设置文本放大.....	21
2.3.6 设置字符对齐.....	22
2.3.7 设置字体加粗.....	22
2.3.8 设置字体旋转.....	23
2.3.9 设置字体方向.....	24
2.3.10 设置反白.....	24
2.3.11 设置斜体.....	25
2.3.12 设置下划线.....	25
2.3.13 设置汉字放大.....	26
2.3.14 设置汉字间距.....	27
2.3.15 设置水平制表位置.....	27
2.3.16 执行到下一个水平制表位置.....	28
2.4 图片和条码打印.....	29

2.4.1	打印 QR 码.....	29
2.4.2	打印混排的 QR 码.....	30
2.4.3	打印一维条码 ().....	31
2.4.4	打印磁盘 BMP 文件 I.....	33
2.4.5	打印磁盘 BMP 文件 II.....	33
2.4.6	设置 NV 位图.....	34
2.4.7	打印 NV 位图.....	35
2.4.8	打印 Data Matrix 二维码.....	37
2.5	打印机信息.....	37
2.5.1	获取打印机状态.....	37
2.5.2	获取打印机特殊功能状态.....	38
2.5.3	获取打印机信息.....	39
2.5.4	获取开发包信息.....	40
2.6	黑标操作.....	41
2.6.1	设置黑标切纸偏移量.....	41
2.6.2	设置黑标打印进纸偏移量.....	41
2.6.3	检测黑标.....	42
2.6.4	检测黑标进纸到打印位置.....	42
2.6.5	检测黑标进纸到切纸位置 ().....	43
2.6.6	打印黑标切纸.....	43
2.7	其他接口.....	44
2.7.1	转发指令(发送).....	44
2.7.2	转发指令(发送和接收).....	45
2.8	定制类打印接口说明.....	45
2.8.1	设置指令模式.....	46
2.8.2	设置旋转打印模式.....	46
2.8.3	发送旋转模式数据.....	47
2.8.4	发送旋转模式条码.....	47
2.8.5	发送旋转模式换行.....	49
2.8.6	发送旋转模式左边距.....	49
2.8.7	打印旋转模式数据.....	50
2.8.8	设置打印机 ID 或名称.....	50
2.8.9	获取打印机 ID 或名称.....	51
2.8.10	扩展设置字符对齐.....	51
2.8.11	设置页模式.....	52
2.8.12	设置页模式数据打印起始位置.....	53
2.8.13	设置页模式打印方向.....	53
2.8.14	页模式数据打印.....	54
2.8.15	打印 QR 码 II.....	54
2.8.16	打印 QR 码 III.....	55
3.	调用示例.....	56
3.1	Msprintsdk.dll.....	56
3.2	Msprintsdk.ocx.....	57
3.3	Msprintsdk.so.....	57

---

4. 附录.....	58
4.1 配置文件.....	58
4.1.1 Windows 配置.....	58
4.1.2 Linux 配置.....	58

## 1. SDK 介绍

### 1.1 概述

软件开发工具包 Software Development Kit，我们在此简称为 SDK。

本 SDK 手册为基于美松科技打印机产品做应用软件开发提供技术参考。使用它可以更好、更快捷的构建你的应用软件。

- 不需要了解打印机的具体指令。
- 封装了和打印机的通信方式，无需处理和打印机的具体通信协议。
- 可以方便的打印图片。
- 对二维条码的支持。
- 获取打印机的状态。

### 1.2 本手册的预期读者

- 应用软件程序开发人员。

### 1.3 支持机型

- 无特别说明外，支持 MS-XXXX 全系列热敏打印机

### 1.4 支持连接方式

- USB 连接
- 串口连接

### 1.5 SDK 文件

- Msprintsdk.dll

动态链接库文件(Dynamic Link Library)，可被不同的应用软件程序使用，用于 Windows 操作系统平台，支持 VB、VC、C#、QT 等多种开发语言调用。

- Msprintsdk.ocx

对象类别扩充组件(Object Linking and Embedding (OLE) Control Extension)，用于 Windows 操作系统平台。支持 VB、VC、C#、QT、JAVA 等多种开发语言，主要应用于在网页开发中调用。

- Msprintsdk.so

共享目标文件(shared object)，用于 Linux 操作系统平台。

本参考手册包括对 SDK 文件 Msprintsdk.dll、Msprintsdk.ocx 和 Msprintsdk.so 的说明，他们大部分接口通用。不同的部分在接口后面以下面的方式标识区分

Msprintsdk.dll 独有的接口标识 。

Msprintsdk.ocx 独有的接口标识 。

Msprintsdk.so 独有的接口标识 。

如果接口前面没有任何标识，表示适应于所有的 SDK 文件。

## 2. 打印接口说明

### 2.1 打印机初始化和结束

#### 2.1.1 设置设备名称(Windows)

[接口]

int SetPrintport(char \*strPort, int iBaudrate)  

[功能]

设置打印机设备连接方式、设备名称和串口通信波特率。

[参数]

参数名称	参数类型	参数说明
strPort	char *	打印机设备端口名称 一般为USB00X或COMX(其中X指数字1、2、3) 接口会根据端口名称前缀USB/COM自动识别连接方式，对应为USB方式连接和串口方式连接
iBaudrate	int	串口通信波特率 支持9600、38400、115200，需和打印机硬件设置保持一致 当打印机连接方式为串口时生效 当打印机连接方式为USB时，该参数无效

[返回值]

0 成功

1 失败

[示例]

```
int r = SetPrintport("COM1",115200);
```

```
int r = SetPrintport("USB001",0);
```

## 2.1.2 设置设备连接(Windows)

[接口]

```
int SetPrintConn(int iConnWay, char *strName, char *strValue) 
```

[功能]

设置打印机设备连接方式、设备名称和参数值。

[参数]

参数名称	参数类型	参数说明
iConnWay	int	打印机连接方式 1 串口 2 USB口 3 蓝牙
strName	char *	打印机设备端口名称 一般为USB00X或COMX(其中X指数字1、2、3) 或蓝牙名称
strValue	char *	连接方式为串口则为通信波特率 支持9600、38400、115200, 需和打印机硬件设置 保持一致 连接方式为USB时, 该参数无效 连接方式为蓝牙时, 为连接密码

[返回值]

0 成功

1 失败

**[示例]**

```
int r = SetPrintConn (1, "COM1", "115200");
```

```
int r = SetPrintConn (2, "USB001", "");
```

```
int r = SetPrintConn (3, " MS-BL58", "1234");
```

**[注意事项]**

此为升级到2.2版本增加对蓝牙打印机的支持添加，可以取代之前的SetPrintport的接口。

**2.1.3 设置设备名称(Linux)****[接口]**

```
int SetDevname(int iDevtype, char *cDevname, int iBaudrate)
```


**[功能]**

设置打印机设备连接方式、设备名称和串口通信波特率。

**[参数]**

参数名称	参数类型	参数说明
iDevtype	int	打印机设备连接方式 1 串口(包括直接串口和虚拟串口) 2 USB口 3 自动匹配USB
cDevname	char *	打印机设备名称 一般情况下当iDevtype=1时, 直接串口的值为/dev/ttySX(其中X指串口端口号 如0、1、2) 虚拟串口的值为/dev/ttyUSBX(其中X指虚拟串口 端口号如0、1、2)

		iDevtype=2时， USB口值为/dev/usb/lpX(其中X指USB端口号如0、1、2)
iDevValue	int	设备参数值 iDevtype=1时 为串口通信波特率 支持9600、38400、115200，需和打印机硬件设置保持一致 当iDevtype=2时，该参数无效 当iDevtype=3时 为USB PID值，为0时使用默认值0x2013

**[返回值]**

- 0 成功
- 1 失败

**[示例]**

```
int r = SetDevname(1, "/dev/ttyS0", 115200);
```

**[注意事项]**

SetDevname(3, "", 0)，自动匹配USB。它具有更快的执行速度，使用USB口的打印机建议使用此接口。默认的VID为0x519，PID为0x2013。

**2.1.4 设置流控**

**[接口]**

```
int SetPrintportFlowCtrl(int iFlowCtrlFlag) 
```

**[功能]**

设置串口和USB口的流控。

**[参数]**

参数名称	参数类型	参数说明
iFlowCtrlFlag	int	0 取消流控 1 设置流控

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r1 = SetPrintport("COM1", 115200);  
int r2 = SetPrintportFlowCtrl (1);
```

**[注意事项]**

需要在SetPrintport后调用。

当使用串口，设置流控，打印机异常时（如打印机未上电），将会一直阻塞，请谨慎使用。

### 2.1.5 自动配置 USB 端口

**[接口]**

```
SetUsbportauto()
```

**[功能]**

自动查找USB设备端口名称。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetUsbportauto();
```

**[注意事项]**

该接口仅适应于USB连接方式和打印机通信。

### 2.1.6 自动配置串口

**[接口]**

SetComportauto() 

**[功能]**

自动查找串口。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetComportauto();
```

**[注意事项]**

该接口仅适应于串口连接方式和打印机通信。

## 2.1.7 打印机初始化

**[接口]**

SetInit()

**[功能]**

打开设备或端口连接打印机，初始化打印机，完成缓存的清空等工作。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetInit();
```

**[注意事项]**

必须成功调用此接口后，才能使打印机工作。

勿频繁调用此接口。

调用此接口前需先设置好打印机连接方式、设备名称等参数。下面针对不同的SDK文件说明。

- Msprintsdk.dll

需先调用SetPrintport或SetUsbportauto设置参数，如果未调用，则会从dll所在的文件夹中查找Config.ini文件，存在Config.ini文件的话，那么会从该文件中读取。文件格式见附录[4.1.1 Windows配置](#)。

- Msprintsdk.ocx

需先调用SetPrintport或SetUsbportauto设置参数。

- Msprintsdk.so

需先调用SetDevname设置参数，如果未调用，则会从so所在的文件夹中查找Config.ini文件，存在Config.ini文件的话，那么会从该文件中读取。文件格式见附录[4.1.2 Linux配置](#)。

## 2.1.8 清理缓存

### [接口]

SetClean()

### [功能]

清理打印机缓存，清除之前设置的参数。

### [参数]

无

### [返回值]

0 成功

1 失败

### [示例]

```
int r = SetClean();
```

## 2.1.9 打印机关闭

### [接口]

SetClose()

**[功能]**

关闭打印设备端口，释放打印机资源。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetClose();
```

**2.1.10 设置汉字模式****[接口]**

SetReadZKmode(int mode)

**[功能]**

设置汉字模式。

**[参数]**

参数名称	参数类型	参数说明
mode	int	汉字模式 0 进入汉字模式 1 退出汉字模式

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetReadZKmode(0);
```

**2.1.11 设置区域国家和代码页****[接口]**

SetCodepage(int country, int CPnumber)



**[功能]**

设置区域国家和代码页。

**[参数]**

参数名称	参数类型	参数说明
country	int	区域国家 详见下表区域国家参数
CPnumber	int	代码页 详见下表代码页参数

## 区域国家参数列表

区域国家参数	区域国家说明
0	美国
1	法国
2	德国
3	英国
4	丹麦 I
5	瑞典
6	意大利
7	西班牙
8	日本
9	挪威
10	丹麦 II

## 代码页参数列表

区域国家参数	区域国家说明
0	PC437[美国欧洲标准]
1	片假名
2	PC850[多语言]
3	PC860[葡萄牙语]

4	PC863[加拿大-法语]
5	PC865[北欧]
16	WPC1252
17	PC866[斯拉夫语2]
18	PC852[拉丁语2]
19	PC858[欧洲]

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetCodepage(0, 0);
```

**2.2 常用打印功能****2.2.1 打印字符串****[接口]**

```
int PrintString(char* strData, int iImme)
```

**[功能]**

打印字符串。

**[参数]**

参数名称	参数类型	参数说明
strData	char *	打印内容
iImme	int	是否立即打印 0 会换行并立即打印 1 不换行，不立即打印

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintString("PrintTest",0);
```

### 2.2.2 打印切纸

#### [接口]

```
int PrintCutpaper(int iMode)
```

#### [功能]

切纸。

#### [参数]

参数名称	参数类型	参数说明
iMode	int	切纸模式 0 全切 1 半切

#### [返回值]

0 成功

1 失败

#### [示例]

```
int r = PrintCutpaper(0);
```

### 2.2.3 打印并换行

#### [接口]

```
int PrintChargeRow()
```

#### [功能]

打印内容并换行，无打印内容的时候走一空白行。

#### [参数]

无

#### [返回值]

0 成功

1 失败

#### [示例]

```
int r = PrintString("PrintTest1",1);
    r = PrintString("PrintTest2",1);
    r = PrintChargeRow();
```

#### 2.2.4 打印细走纸

[接口]

int PrintFeedDot(int Lnumber)  

[功能]

走纸。

[参数]

参数名称	参数类型	参数说明
Lnumber	int	走纸单位 值范围 0-250 单位值 = 0.125mm

[返回值]

0 成功

1 失败

[示例]

```
int r = PrintFeedDot(10);
```

#### 2.2.5 打印自检页

[接口]

int PrintSelfcheck()  

[功能]

走纸。

[参数]

无

[返回值]

0 成功

1 失败

[示例]

```
int r = PrintSelfcheck();
```

## 2.3 打印效果

### 2.3.1 设置行间距

[接口]

```
int SetLinespace(int iLinespace)
```

[功能]

设置行间距。

[参数]

参数名称	参数类型	参数说明
iLinespace	int	行间距 值范围 0-127 单位值 = 0.125mm

[返回值]

0 成功

1 失败

[示例]

```
int r = iLinespace(10);
```

### 2.3.2 设置字符间距

[接口]

```
int SetSpacechar(int iSpace)
```

[功能]

设置字符间距。

[参数]

参数名称	参数类型	参数说明
iSpace	int	字符间距

		值范围 0-64 单位值 = 0.125mm
--	--	---------------------------

**[返回值]**

- 0 成功
- 1 失败

**[示例]**

```
int r = SetSpacechar(10);
```

### 2.3.3 设置左边界

**[接口]**

```
int SetLeftmargin(int iLeftspace)
```

**[功能]**

设置左边界。

**[参数]**

参数名称	参数类型	参数说明
iLeftspace	int	左边界 值范围 0-576 单位值 = 0.125mm

**[返回值]**

- 0 成功
- 1 失败

**[示例]**

```
int r = SetLeftmargin(10);
```

### 2.3.4 设置字符放大

**[接口]**

```
int SetSizechar(int iHeight, int iWidth, int iUnderline, int iAsciiptype)
```

**[功能]**

设置字符格式。

**[参数]**

参数名称	参数类型	参数说明
iHeight	int	倍高 0 无效 1 有效
iWidth	int	倍宽 0 无效 1 有效
iUnderline	int	下划线 0 无效 1 有效
iAsciiType	int	ASCII字形 0 12*24 1 9*17

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetSizechar(1, 1, 1, 1);
```

**2.3.5 设置文本放大****[接口]**

```
int SetSizetext(int iHeight, int iWidth)
```

**[功能]**

设置文本放大。

**[参数]**

参数名称	参数类型	参数说明
iHeight	int	放大高度 值范围 1-8

iWidth	int	放大宽度 值范围 1-8
--------	-----	-----------------

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetSizetext(2,2);
```

**2.3.6 设置字符对齐****[接口]**

```
int SetAlignment(int iAlignment)
```

**[功能]**

设置字符对齐。

**[参数]**

参数名称	参数类型	参数说明
iAlignment	int	字符对齐 0 左对齐 1 居中对齐 2 右对齐

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetAlignment(1);
```

**2.3.7 设置字体加粗****[接口]**

```
int SetBold(int iBold)
```

**[功能]**

设置字体加粗。

**[参数]**

参数名称	参数类型	参数说明
iBold	int	字体加粗 0 不加粗 1 加粗

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetBold(1);
```

### 2.3.8 设置字体旋转

**[接口]**

```
int SetRotate(int iRotate)
```

**[功能]**

设置字体旋转。

**[参数]**

参数名称	参数类型	参数说明
iRotate	int	字体旋转 0 解除旋转 1 顺时针度旋转90°

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetRotate(1);
```

### 2.3.9 设置字体方向

#### [接口]

```
int SetDirection(int iDirection)
```

#### [功能]

设置字体方向。

#### [参数]

参数名称	参数类型	参数说明
iDirection	int	字体方向 0 解除旋转 1 旋转180度

#### [返回值]

0 成功

1 失败

#### [示例]

```
int r = SetDirection(0);
```

### 2.3.10 设置反白

#### [接口]

```
int SetWhitemodel(int iWhite)
```

#### [功能]

设置反白。

#### [参数]

参数名称	参数类型	参数说明
iWhite	int	反白 0 取消反白 1 设置反白

#### [返回值]

0 成功

1 失败

**[示例]**

```
int r = SetWhitemodel(1);
```

**2.3.11 设置斜体****[接口]**

```
int SetItalic(int iItalic)
```

**[功能]**

设置斜体。

**[参数]**

参数名称	参数类型	参数说明
iItalic	int	斜体 0 取消斜体 1 设置斜体

**[返回值]**

- 0 成功
- 1 失败

**[示例]**

```
int r = SetItalic (1);
```

**2.3.12 设置下划线****[接口]**

```
int SetUnderline(int underline)  
```

**[功能]**

设置下划线（字符，ASCII 都有效）。

**[参数]**

参数名称	参数类型	参数说明
underline	int	下划线 0 无 1 一个点下划线

		2 两个点下划线
--	--	----------

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetUnderline (1);
```

**2.3.13 设置汉字放大****[接口]**

```
int SetSizechinese(int iHeight, int iWidth, int iUnderline, int iChinesetype)
```

**[功能]**

设置汉字格式。

**[参数]**

参数名称	参数类型	参数说明
iHeight	int	倍高 0 无效 1 有效
iWidth	int	倍宽 0 无效 1 有效
iUnderline	int	下划线 0 无效 1 有效
iChinesetype	int	汉字字形 0 24*24 1 16*16

**[返回值]**

0 成功

1 失败

## [示例]

```
int r = SetSizechinese(1, 1, 1, 1);
```

## 2.3.14 设置汉字间距

## [接口]

```
int SetSpacechinese(int iChsleftspace, int iChsrightspace)
```

## [功能]

设置汉字间距。

## [参数]

参数名称	参数类型	参数说明
iChsleftspace	int	汉字左空 值范围 0-64 单位值 = 0.125mm
iChsrightspace	int	汉字右空 值范围 0-64 单位值 = 0.125mm

## [返回值]

0 成功

1 失败

## [示例]

```
int r = SetSpacechinese(10, 10);
```

## 2.3.15 设置水平制表位置

## [接口]

```
int SetHTseat(const char* bHTseat, int iLength)
```

## [功能]

设置水平制表位置。

## [参数]

参数名称	参数类型	参数说明
------	------	------

bHTseat	char*	水平制表的位置 从小到大 单位一个ASCII字符 不能为0
iLength	int	水平制表的位置数据的个数 值范围 1-32

**[返回值]**

0 成功

1 失败

**[示例]**参见[执行到下一个水平制表位置](#)**2.3.16 执行到下一个水平制表位置****[接口]**

int PrintNextHT()

**[功能]**

执行到下一个水平制表位置。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
char cSeat[3]={10, 18, 25};
SetHTseat (cSeat, 3);
PrintString("1", 1);
PrintNextHT();
PrintString("2", 1);
PrintNextHT();
PrintString("3", 1);
PrintNextHT();
```

```
PrintString("4", 0);  
PrintString("1a", 1);  
PrintNextHT();  
PrintString("2a", 1);  
PrintNextHT();  
PrintString("3a", 1);  
PrintNextHT();  
PrintString("4a", 0);  
PrintString("1b", 1);  
PrintNextHT();  
PrintString("2b", 1);  
PrintNextHT();  
PrintString("3b", 1);  
PrintNextHT();  
PrintString("4b", 0);
```

打印效果

1	2	3	4
1a	2a	3a	4a
1b	2b	3b	4b

**[注意事项]**

在OCX中参数格式使用SetHTseat("10, 18, 25, ", 3);

## 2.4 图片和条码打印

### 2.4.1 打印 QR 码

**[接口]**

```
int PrintQrcode(const char* strData, int iLmargin, int iMside, int iRound)
```



**[功能]**

打印QR码。

### [参数]

参数名称	参数类型	参数说明
strData	char*	QR码内容
ilmargin	int	左边距 值范围 0-27 单位值 = 1mm
iMside	int	QR码大小 值范围 1-8
iRound	int	环绕模式 0 立即打印（不混排） 1 环绕（混排，有些机型不支持）

### [返回值]

- 0 成功
- 1 失败

### [示例]

```
int r = PrintQrcode("QR Code", 2, 2, 0);
```

### [注意事项]

打印QR码有多个接口，参看定制类接口说明。跟打印机型号有关系，如有问题，具体可咨询技服人员。

## 2.4.2 打印混排的 QR 码

### [接口]

PrintRemainQR() 

### [功能]

QR混排打印时候，打印混排的QR码。

### [参数]

无

### [返回值]

- 0 成功
- 1 失败

**[示例]**

```
PrintQrcode("QR Code:123456", 2, 4, 1);
SetLeftmargin(120);
PrintString("QR Code:", 0);
PrintString("123456", 0);
int r = PrintRemainQR();
```

打印效果



### 2.4.3 打印一维条码()

**[接口]**

```
int Print1Dbar(int iWidth, int iHeight, int iHrsize, int iHriseat, int iCodetype, const char* strData)
```

**[功能]**

打印一维条码。

**[参数]**

参数名称	参数类型	参数说明
iWidth	int	条码宽度 值范围 2-6 单位值 = 0.125mm
iHeight	int	条码高度 值范围 1-255 单位值 = 0.125mm
iHrsize	int	条码显示字符字型

		0 12*24 1 9*17
iHriseat	int	条码显示字符位置 0 无 1 上 2 下 3 上下
iCodetype	int	条码类型 详见下表条码类型参数
strData	char*	条码内容

条码类型参数列表

条码类型参数	条码类型说明
0	* UPC-A
1	* UPC-E
2	* EAN13
3	* EAN8
4	* CODE39
5	* ITF
6	* CODABAR
7	* Standard EAN13
8	* Standard EAN8
9	* CODE93
10	* CODE128

[返回值]

0 成功

1 失败

[示例]

```
int r = PrintIDbar(3, 60, 0, 1, 10, "IDbar");
```

#### 2.4.4 打印磁盘 BMP 文件 I

##### [接口]

```
int PrintDiskbmpfile(const char* strPath)
```

##### [功能]

打印磁盘BMP文件。

##### [参数]

参数名称	参数类型	参数说明
strPath	char*	图像文件路径 若只有文件名则使用当前路径，若指定全路径则使用指定的路径 单色BMP格式文件

##### [返回值]

- 0 成功
- 1 失败

##### [示例]

```
int r = PrintDiskbmpfile("D:\\test.bmp");
```

打印效果



##### [注意事项]

仅支持单色BMP格式文件，即图像深度为1bit的Bitmap文件。

#### 2.4.5 打印磁盘 BMP 文件 II

##### [接口]

```
int PrintDiskimgfile(const char* strPath)
```

##### [功能]

打印磁盘BMP文件。

### [参数]

参数名称	参数类型	参数说明
strPath	char*	图像文件路径 若只有文件名则使用当前路径，若指定全路径则使用指定的路径 单色和24位色深的BMP格式文件

### [返回值]

0 成功

1 失败

### [示例]

```
int r = PrintDiskimgfile("D:\\test.bmp");
```

打印效果



### [注意事项]

支持单色和24位色深的BMP格式文件，即图像深度为1bit和24bit的Bitmap文件。

## 2.4.6 设置NV位图

### [接口]

```
int SetNvbmp(int iNums, const char* strPath)
```



### [功能]

设置NV位图。

### [参数]

参数名称	参数类型	参数说明
iNums	int	位图数量 单个位图文件最大64KB

		位图文件的数量限制为，所有文件合计最大192KB
strPath	char*	图像文件路径 若只有文件名则使用当前路径，若指定全路径则使用指定的路径，以” ;” 分隔 个数需和iNums参数一致

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetNvbmp(3, "D:\\test1.bmp;D:\\BMP\\test2. bmp;test3. bmp");
```

**[注意事项]**

仅支持单色BMP格式文件，即图像深度为1bit的Bitmap文件。

该操作会替换之前设置的位图。

图片内容存储于打印机NV Falsh中，该接口适应于固定的图片内容，如LOGO信息。

### 2.4.7 打印 NV 位图

**[接口]**

```
int PrintNvbmp(int iNvindex,int iMode)
```

**[功能]**

打印NV BMP文件。

**[参数]**

参数名称	参数类型	参数说明
iNvindex	int	NV位图索引，从1开始 值范围 1-N，N取决于SetNvbmp设置的量，当N大于SetNvbmp设置的量时，打印机不会打印。
iMode	int	位图大小 48 普通 49 倍宽 50 倍高

## 51 倍宽倍高(4倍大小)

## [返回值]

0 成功

1 失败

## [示例]

```
int r = PrintNvbmp(1, 48);
```

打印效果



```
int r = PrintNvbmp(1, 49);
```

打印效果



```
int r = PrintNvbmp(1, 50);
```

打印效果



```
int r = PrintNvbmp(1, 51);
```

打印效果



### 2.4.8 打印 Data Matrix 二维码

#### [接口]

```
int PrintDataMatrix(const char* strData, int iSize)
```

#### [功能]

打印Data Matrix二维码。

#### [参数]

参数名称	参数类型	参数说明
strData	char*	二维码内容
iSize	int	矩阵大小 最小为2 最大和内容长度有关

#### [返回值]

0 成功

1 失败

#### [示例]

```
int r = PrintDataMatrix("DataMatrix",6);
```

## 2.5 打印机信息

### 2.5.1 获取打印机状态

#### [接口]

```
int GetStatus()
```

**[功能]**

获取打印机状态。

**[参数]**

无

**[返回值]**

返回值	返回值说明
0	打印机正常
1	打印机未连接或未上电
2	打印机和调用库不匹配
3	打印头打开
4	切刀未复位
5	打印头温度异常
6	黑标错误，可能的情况为 黑标传感器坏 使用错误白纸 黑标不标准，浓度偏低
7	纸尽
8	纸将尽

**[示例]**

```
int r = GetStatus();
```

## 2.5.2 获取打印机特殊功能状态

**[接口]**

```
int GetStatusspecial ()
```

**[功能]**

获取打印机特殊功能状态。

**[参数]**

无

### [返回值]

返回值	返回值说明
0	打印机正常
1	打印机未连接或未上电
2	打印机和调用库不匹配
3	当前使用打印机无特殊功能
4	容纸器没有可靠上纸
5	纸张堵在出票口，持续堆叠
6	卡纸，出纸口未堵的情况下，胶辊无法驱动纸持续前进了。 比如纸卷在胶辊上了、切刀堵住了纸的前进、纸张被拉住无法前进等
7	拽纸，打印机感受到凭条被外力拖拽
8	出纸传感器有纸

### [示例]

```
int r = GetStatusspecial();
```

### [注意事项]

仅适用于D347等部分机型和部分状态。

EP802-TM 不支持 4、7

EP802-TMP 不支持 5、7

## 2.5.3 获取打印机信息

### [接口]

```
int GetProductinformation(int iFstype, char* bFiddata, int iFidlen)
```



### [功能]

获取产品的基本信息。

### [参数]

参数名称	参数类型	参数说明
iFstype	int	信息类型 1 打印头型号ID 2 PCB板版本 3 固件版本 4 生产厂商信息 5 打印机型号 6 支持的中文编码格式 7 校验码信息(CheckSum)
bFiddata	char*	返回信息内容 输出值 iFstype=7时, 返回值为两字节十六进制数值, 高位在前, 低位在后。
iFidlen	int	返回信息长度(此值无效)

**[返回值]**

0 成功

1 失败

**[示例]**

```
int iLen = 0;
char bFiddata[30]={0};
int r = GetProductinformation(3,bFiddata,iLen);
```

**2.5.4 获取开发包信息****[接口]**

GetSDKinformation(char\* bInfodata)  

**[功能]**

获取开发包版本等信息。

**[参数]**

参数名称	参数类型	参数说明
------	------	------

bInfodata	char*	返回信息内容 输出值
-----------	-------	---------------

**[返回值]**

0 成功

1 失败

**[示例]**

```
char bInfodata[20]={0};
int r = GetSDKinformation(bInfodata);
```

**2.6 黑标操作****2.6.1 设置黑标切纸偏移量****[接口]**

```
int SetMarkoffsetcut(int iOffset)
```

**[功能]**

设置黑标定位偏移量。

**[参数]**

参数名称	参数类型	参数说明
iOffset	int	偏移量 值范围 0-1600 单位值 = 0.125mm

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetMarkoffsetcut(10);
```

**2.6.2 设置黑标打印进纸偏移量****[接口]**

```
int SetMarkoffsetprint(int iOffset)
```

**[功能]**

设置黑标定位偏移量。

**[参数]**

参数名称	参数类型	参数说明
iOffset	int	偏移量 值范围 0-1600 单位值 = 0.125mm

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetMarkoffsetprint(10);
```

**2.6.3 检测黑标****[接口]**

```
int PrintMarkposition()
```

**[功能]**

黑标模式下检测黑标，停止在黑标位置。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintMarkposition();
```

**2.6.4 检测黑标进纸到打印位置****[接口]**

```
int PrintMarkpositionPrint()
```

**[功能]**

黑标模式下检测黑标并进纸到打印位置。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintMarkpositionPrint();
```

**[注意事项]**

进纸偏移量影响走纸距离。

### 2.6.5 检测黑标进纸到切纸位置()

**[接口]**

```
int PrintMarkpositioncut()
```

**[功能]**

黑标模式下检测黑标并进纸到切纸位置。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintMarkpositioncut();
```

**[注意事项]**

切纸偏移量影响走纸距离。

### 2.6.6 打印黑标切纸

**[接口]**

```
int PrintMarkcutpaper(int iMode)
```

### [功能]

黑标切纸。

### [参数]

参数名称	参数类型	参数说明
iMode	int	切纸模式 0 检测黑标全切 1 不检测黑标半切

### [返回值]

0 成功

1 失败

### [示例]

```
int r = PrintMarkcutpaper(0);
```

## 2.7 其他接口

### 2.7.1 转发指令(发送)

#### [接口]

```
int PrintTransmit(const char* bCmd, int iLength)
```

#### [功能]

将指令原样转给打印机。

#### [参数]

参数名称	参数类型	参数说明
bCmd	char*	指令
iLength	int	指令长度

#### [返回值]

0 成功

1 失败

#### [示例]

```
char bCmd[2]={0x1B, 0x69}; //切纸指令
```

```
int r = PrintTransmit(bCmd, 2);
```

### 2.7.2 转发指令(发送和接收)

#### [接口]

```
int GetTransmit(const char* bCmd, int iLength, char* bRecv, int iReLen)
```



#### [功能]

将指令原样转给打印机，并接收返回值。

#### [参数]

参数名称	参数类型	参数说明
bCmd	char*	指令
iLength	int	指令长度
bRecv	char*	返回数据
iReLen	int	返回数据的长度 (此值无效)

#### [返回值]

>0 成功，表示返回数据的长度

<=0 失败

#### [示例]

```
char bCmd[3]={0x10, 0x04, 0x01}; //查询打印机状态
char bRecv[3]={0};
int iReLen = 0;
int r = GetTransmit(bCmd, 3, bRecv, iReLen);
```

## 2.8 定制类打印接口说明

下列接口只适用于特定型号的打印机型，如EP800、EP802等。

具体可咨询技服人员。

## 2.8.1 设置指令模式

### [接口]

```
SetCommandmode(int iMode)
```

### [功能]

切换打印机指令模式。以支持实现定制类打印接口功能。

### [参数]

参数名称	参数类型	参数说明
iMode	int	指令模式 2 EPIC模式 3 EPOS模式

### [返回值]

0 成功

1 失败

### [示例]

```
int r = SetCommandmode(2);
```

### [注意事项]

使用定制类打印接口，需先将指令模式切换为EPIC模式。使用完成后，需将指令模式再切换为EPOS模式，才能使用常用的标准打印接口。

## 2.8.2 设置旋转打印模式

### [接口]

```
int SetRotation_Intomode()
```



### [功能]

设置进入旋转打印模式。

### [参数]

无

### [返回值]

0 成功

1 失败

**[示例]**

```
int r = SetRotation_Intomode();
```

**2.8.3 发送旋转模式数据****[接口]**

```
int PrintRotation_Sendtext(char* strData, int iImme)
```

**[功能]**

旋转模式下传文本数据。

**[参数]**

参数名称	参数类型	参数说明
strData	char*	文本数据
iImme	int	是否换行 0 换行 1 不换行

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintRotation_Sendtext("Rotation", 1);
```

**2.8.4 发送旋转模式条码****[接口]**

```
int PrintRotation_Sendcode(int leftspace, int iWidth, int iHeight, int
```

```
iCodetype, const char* strData)
```

**[功能]**

旋转模式下传条码数据。

**[参数]**

参数名称	参数类型	参数说明
------	------	------

leftspace	int	条码左边距 值范围 0-36 单位mm
iWidth	int	条码宽度 值范围 2-6 单位值 = 0.125mm
iHeight	int	条码高度 值范围 1-10 单位值 = 3mm
iCodetype	int	条码类型 详见下表条码类型参数
strData	char*	条码内容

条码类型参数列表

条码类型参数	条码类型说明
0	* UPC-A
1	* UPC-E
2	* EAN13
3	* EAN8
4	* CODE39
5	* ITF
6	* CODABAR
7	* <del>Standard EAN13</del>
8	* <del>Standard EAN8</del>
9	* CODE93
10	* CODE128

## [返回值]

- 0 成功
- 1 失败

**[示例]**

```
int r = PrintRotation_Sendcode(10, 3, 60, 10, "1Dbar");
```

**2.8.5 发送旋转模式换行****[接口]**

```
int PrintRotation_Changeline()
```

**[功能]**

旋转模式下换行。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintRotation_Changeline();
```

**2.8.6 发送旋转模式左边距****[接口]**

```
int SetRotation_Leftspace(int iLeftspace)
```

**[功能]**

设置旋转模式下左边距。

**[参数]**

参数名称	参数类型	参数说明
iLeftspace	int	左边距 值范围 0-36 单位mm

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetRotation_Leftspace(10);
```

**2.8.7 打印旋转模式数据****[接口]**

```
int PrintRotation_Data()  
```

**[功能]**

打印进入旋转模式后保存的数据，并退出旋转模式默认EPOS指令模式。

**[参数]**

无

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintRotation_Data();
```

**2.8.8 设置打印机 ID 或名称****[接口]**

```
int SetPrintIDorName(char* strIDorNAME)  
```

**[功能]**

设置打印机ID或名称。

**[参数]**

参数名称	参数类型	参数说明
strIDorNAME	char*	打印机ID或名称 值长度范围 1-14

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = SetPrintIDorName("EP800-000001");
```

### 2.8.9 获取打印机 ID 或名称

#### [接口]

```
int GetPrintIDorName(char* strIDorNAME) 
```

#### [功能]

获取打印机ID或名称。

#### [参数]

参数名称	参数类型	参数说明
strIDorNAME	char*	打印机ID或名称 返回值

#### [返回值]

0 成功

1 失败

#### [示例]

```
char bRecv[30]={0};
```

```
int r = GetPrintIDorName(bRecv);
```

### 2.8.10 扩展设置字符对齐

#### [接口]

```
int SetAlignmentLeftRight(int iAlignment) 
```

#### [功能]

在同一行上左对齐和右对齐。

#### [参数]

参数名称	参数类型	参数说明
iAlignment	int	字符对齐 0 左对齐 2 右对齐

#### [返回值]

0 成功

1 失败

#### [示例]

```
SetClean();
SetAlignmentLeftRight(0);
PrintString("Left",1);
SetAlignmentLeftRight(2);
PrintString("Right",0);
```

#### [注意事项]

需要按照示例使用，否则可能会导致数据错乱。

### 2.8.11 设置页模式

#### [接口]

```
int SetPagemode(int iMode,int Xrange,int Yrange)
```



#### [功能]

将数据按整页的模式打印。

#### [参数]

参数名称	参数类型	参数说明
iMode	int	0 退出页模式 1 进入页模式
Xrange	int	页宽度，最大576
Yrange	int	页高度，最大640

#### [返回值]

0 成功

1 失败

#### [示例]

```
int r = SetPagemode(1, 576, 640);
```

## 2.8.12 设置页模式数据打印起始位置

[接口]

int SetPagestartposition(int Xdot, int Ydot) 

[功能]

设置页模式数据打印起始位置。

[参数]

参数名称	参数类型	参数说明
Xdot	int	X位置, 最大576
Ydot	int	Y位置, 最大640

[返回值]

0 成功

1 失败

[示例]

```
int r = SetPagestartposition(40, 0);
```

## 2.8.13 设置页模式打印方向

[接口]

int SetPagedirection(int iDirection) 

[功能]

页模式打印方向。

[参数]

参数名称	参数类型	参数说明
iDirection	int	打印方向 0 正常 1 旋转90度 2 旋转180度 3 旋转270度

[返回值]

0 成功

1 失败

[示例]

```
int r = SetPagedirection(2);
```

#### 2.8.14 页模式数据打印

[接口]

```
int PrintPagedata()  
```

[功能]

页模式打印整页数据。

[参数]

无

[返回值]

0 成功

1 失败

[示例]

```
int r = PrintPagedata ();
```

#### 2.8.15 打印 QR 码 II

[接口]

```
int PrintQrcodeII(const char* strData, int iLen, int iMside) 
```

[功能]

打印QR码。

[参数]

参数名称	参数类型	参数说明
strData	char*	QR码内容 支持不可见字符
iLen	int	strData内容长度

iMside	int	QR码大小 值范围 1-16
--------	-----	-------------------

**[返回值]**

0 成功

1 失败

**[示例]**

```
char cQRData[128]={0x31, 0x32, 0x33, 0x34, 0x35, 0x36};
int r = PrintQrcodeII(cQRData, 6, 8);
```

**2.8.16 打印 QR 码 III****[接口]**

```
int PrintQrcode500II(int iwidth, const char* strData)
```

**[功能]**

打印QR码。

**[参数]**

参数名称	参数类型	参数说明
iwidth	int	QR码大小 值范围 1-8
strData	char*	QR码内容

**[返回值]**

0 成功

1 失败

**[示例]**

```
int r = PrintQrcode500II (3, "QR Code");
```

**[注意事项]**

兼容以前SDK1.0版本保留。

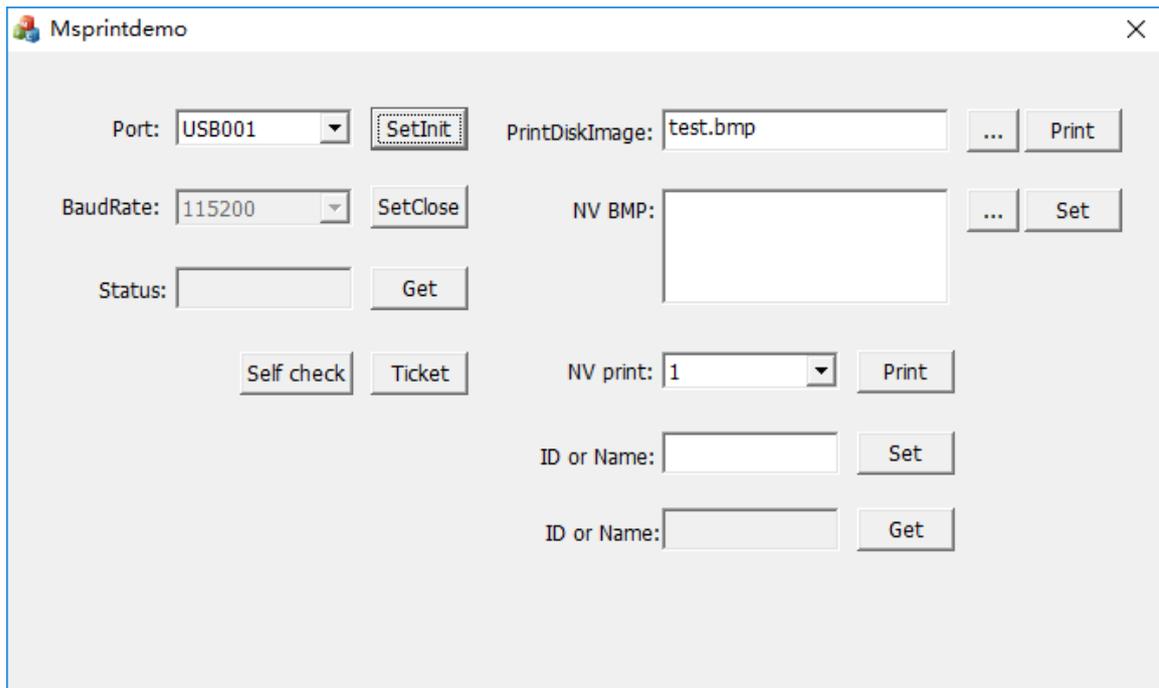
### 3. 调用示例

#### 3.1 Msprintsdk.dll

##### VC示例

Msprintdemo

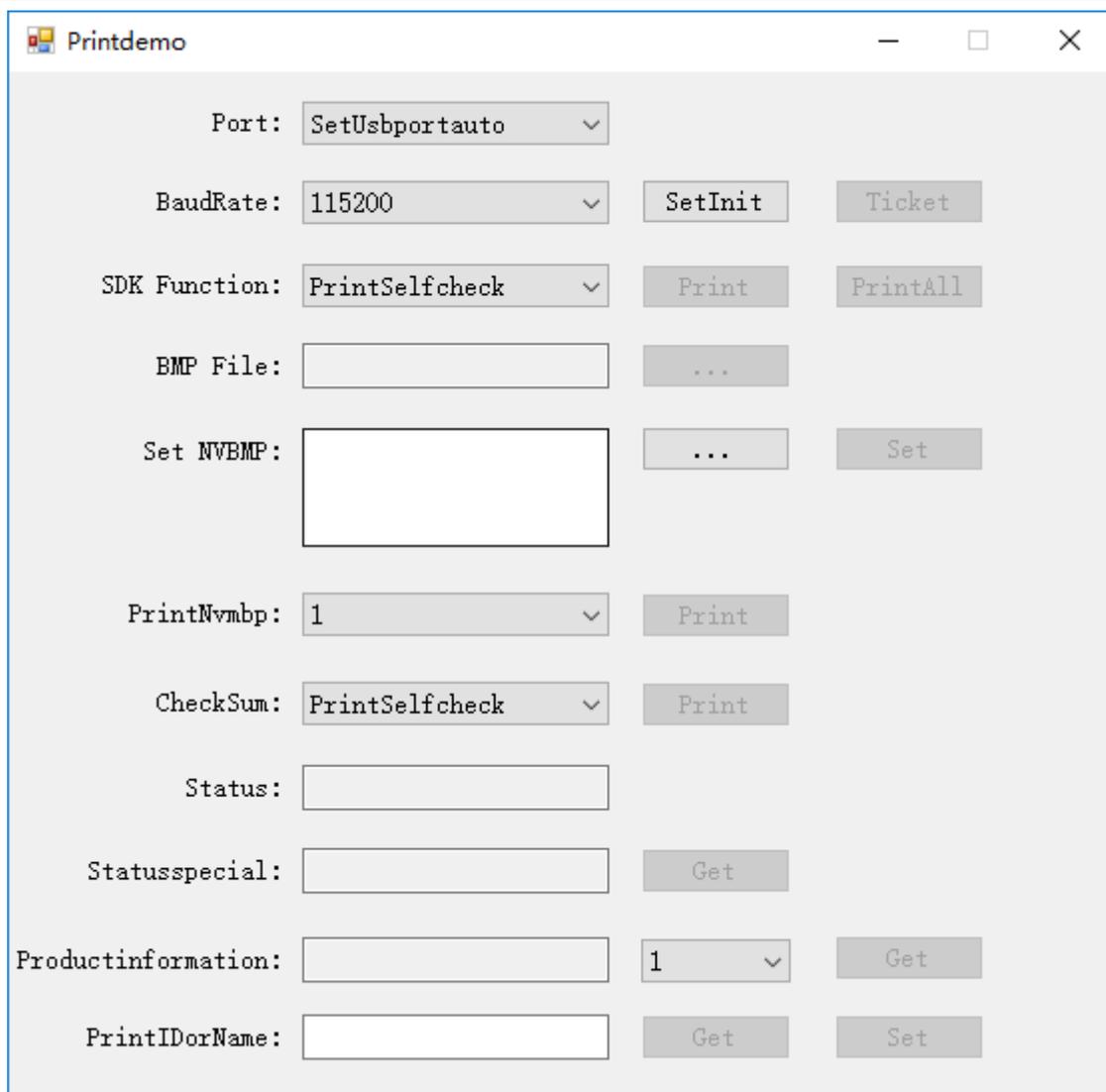
Visual Studio 2008 VC++下编译通过



##### C#示例

Msprintcsharp

Visual Studio 2008 C#下编译通过



注意项目属性中，生成->目标平台，设置为X86；勾选允许不安全代码。

### 3.2 Msprintsdk.ocx

使用前需要注册，XP系统可运行RegMsprintsdk1.bat注册；win7/win8/win10系统时RegMsprintsdk2.bat上右键，选”以管理员身份运行”注册

使用示例参见 Msprintsdk.htm

### 3.3 Msprintsdk.so

示例 msprintdemo 编译

```
gcc -o msprintdemo msprintdemo.c /lib/Msprintsdk.so -lstdc++
```

## 4. 附录

### 4.1 配置文件

#### 4.1.1 Windows 配置

**[文件名]**

Config.ini

**[示例]**

[PortConfig]

Portname=COM3

Baudrate=115200

**[配置说明]**

同本手册[2.1.1 设置设备名称 \(Windows\)](#)中对应参数。

#### 4.1.2 Linux 配置

**[文件名]**

Config.ini

**[示例]**

[DevConfig]

DevType=1

DevName=/dev/ttyUSB0

Baudrate=115200

或

[DevConfig]

DevType=3

PID=8211

**[配置说明]**

同本手册[2.1.3 设置设备名称\(Linux\)](#)中对应参数。