



**Advanced Card Systems Ltd.**  
Card & Reader Technologies

# PTR89

## 便携式热敏 打印机



应用程序编程接口 V1.00



## 目录

<b>1.0.</b>	<b>简介</b> .....	<b>3</b>
1.1.	文档范围和限制 .....	3
1.2.	参考 .....	3
<b>2.0.</b>	<b>数据结构</b> .....	<b>4</b>
2.1.	PRT_PAGE_MODE_PARAM .....	4
<b>3.0.</b>	<b>API 函数</b> .....	<b>5</b>
3.1.	Printer_Open .....	5
3.2.	Printer_Close .....	5
3.3.	Printer_Reset .....	5
3.4.	Printer_FeedPaper .....	6
3.5.	Printer_SetLineSpaceSM .....	6
3.6.	Printer_PrinterStrSM .....	6
3.7.	Printer_PrinterStrPM .....	7
3.8.	Printer_PrinterDataSM .....	7
3.9.	Printer_PrinterDataPM .....	7
3.10.	Printer_PrinterImage .....	8

## 表目录

<b>表 1</b>	<b>: 页面模式下的数据对象</b> .....	<b>4</b>
------------	---------------------------	----------



## 1.0. 简介

ACR89 手持式智能卡读写器在单个可靠设备的成本上提供多个功能部件。它配备一个 32 位 CPU，运行嵌入式实时操作系统（FreeRTOS）。另外 ACS 还为其配套开发出热敏打印机功能，以便为多种智能卡应用提供便利。PTR89 是专为 ACR89 系列读写器设计的便携式热敏打印机。

本文档将对 PTR89 的 API（应用程序编程接口）命令进行介绍，以帮助用户为 ACR89 读写器开发支持打印功能的脱机应用程序。应用软件开发人员可以通过这些 API 开发提供打印功能的智能卡应用。

### 1.1. 文档范围和限制

本 API 文档提供了使用 ACR89 系列读写器执行热敏打印机命令时的详细指南。

### 1.2. 参考

关于 FreeRTOS 软件环境的详细资料，请参考以下链接：

- <http://www.freertos.org/>



## 2.0. 数据结构

### 2.1. PRT\_PAGE\_MODE\_PARAM

```
[Printer.h]

typedef struct
{
    UINT16 HorizontalOrigin_X;
    UINT16 VerticalOrigin_Y;
    UINT16 PrintWidth_X;
    UINT16 PrintHeight_Y;
    UINT8 ucLineSpace;
} PRT_PAGE_MODE_PARAM;
```

用于在页面模式下设置打印区域。

数据成员	值 (包含)	说明
HorizontalOrigin_X	0 至 383	X 轴起点
VerticalOrigin_Y	0 至 882	Y 轴起点
PrintWidth_X	1 至 384	打印区域的宽度
PrintHeight_Y	1 至 883	打印区域的高度
ucLineSpace	24 至 255	行距

表1 : 页面模式下的数据对象

注:

- $HorizontalOrigin\_X + PrintWidth\_X$  应当小于等于 384。
- $VerticalOrigin\_Y + PrintHeight\_Y$  应当小于等于 883。
- 水平方向起点与绝对原点之间的距离为  $HorizontalOrigin\_X * 0.125$  毫米。
- 垂直方向起点与绝对原点之间的距离为  $VerticalOrigin\_Y * 0.125$  毫米。
- 实际打印宽度 =  $PrintWidth\_X * 0.125$  毫米。
- 实际打印高度 =  $PrintHeight\_Y * 0.125$  毫米。
- 实际打印行距 =  $ucLineSpace * 0.125$  毫米。
- 绝对原点指可打印区域的左上角，打印宽度和打印高度都不能设置为 0。
- 行距包含字体的高度。



## 3.0. API 函数

打印机一行最多能打印 24 个字符。打印字体为 16x18 点，点间距为 0.125 毫米。打印机提供两种打印模式，分别是“标准模式”和“页面模式”。在“标准模式”下，无论是否设置打印区域，打印机都会打印出所有发送至缓冲区的数据，而在“页面模式”下，需要提前设置打印区域的大小，并且打印的数据限定在 490 个字节以内。超出打印区域的数据将被弃用。

### 3.1. Printer\_Open

此函数用于开启打印机端口。使用此函数之前，打印机的端口应处于关闭状态，否则 *Printer\_Open* 函数会返回错误信息。

```
[Printer.h]

BOOLEAN Printer_Open (
    void );
```

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE，指示打印机端口是否成功开启。

### 3.2. Printer\_Close

此函数用于关闭打印机端口。

```
[Printer.h]

BOOLEAN Printer_Close (
    void );
```

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE，指示打印机端口是否成功关闭。

### 3.3. Printer\_Reset

此函数用于复位打印机，并将所有的用户设置还原为默认值。

```
[Printer.h]

void Printer_Reset (
    void );
```

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE，指示打印机端口是否成功复位。



### 3.4. Printer\_FeedPaper

此函数用于从打印机进纸。

```
[Printer.h]

BOOLEAN Printer_FeedPaper (
    UINT8 Num );
```

参数:

**Num** [in] 待进纸的长度 [0-255, 实际长度等于 Num\*0.125 mm]

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE, 指示是否成功进纸。

### 3.5. Printer\_SetLineSpaceSM

此函数用于设置行距。

```
[Printer.h]

BOOLEAN Printer_SetLineSpaceSM (
    UINT8 Num );
```

参数:

**Num** [in] 待进纸的长度 [24-255, 实际长度等于 Num\*0.125 mm]

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE, 指示是否成功进纸。

### 3.6. Printer\_PrinterStrSM

此函数用于在“标准模式”下打印字符串。打印的数据大小不得超过 65535 字节, 可以使用控制字符 '\n'。

```
[Printer.h]

BOOLEAN Printer_PrinterStrSM (
    const UCHAR *Str );
```

参数:

**\*Str** [in] 待打印的以空字符结尾的字符串

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE, 指示是否在“标准模式”下成功打印字符串。



### 3.7. Printer\_PrinterStrPM

此函数用于在“页面模式”下打印字符串。打印数据的长度不得超过 490 个字节。如果数据长度超过 490 个字节，则超出的数据会被弃用。另外可以使用控制字符'\n'。

```
[Printer.h]

BOOLEAN Printer_PrintStrPM (
    const PRT_PAGE_MODE_PARAM *Param,
    const UCHAR *Str );
```

参数:

**\*Param** [in] 待打印的区域  
**\*Str** [in] 待打印的以空字符结尾的字符串

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE，指示是否在“页面模式”下成功打印字符串。

### 3.8. Printer\_PrinterDataSM

此函数用于在“标准模式”下打印字符数组，可以使用控制字符'\n'。

```
[Printer.h]

BOOLEAN Printer_PrintDataSM (
    const UINT8 *Data
    UINT16 Size );
```

参数:

**\*Data** [in] 指向待打印的字符数组的指针  
**Size** [in] 待打印的字符数组的大小[byte]

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE，指示是否在“标准模式”下成功打印数据数组。

### 3.9. Printer\_PrinterDataPM

此函数用于在“页面模式”下打印字符数组，打印数据的长度不得超过 490 个字节，可以使用控制字符'\n'。

```
[Printer.h]

BOOLEAN Printer_PrintDataPM (
    const PRT_PAGE_MODE_PARAM *Param,
    const UINT8 *Data
    UINT16 Size);
```

参数:



- \*Param** [in] 待打印的区域
- Data** [in] 指向待打印的字符数组的指针
- Size** [in] 待打印的字符数组的大小[byte, 1-490]

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE, 指示是否在“页面模式”下成功打印数据数组。

### 3.10. Printer\_PrinterImage

此函数用于打印图像。每个字节代表在水平方向上打印的八个点。图像数据按照从左至右和从上至下的方式按字节在纸上打印出来。

[Printer.h]

```
BOOLEAN Printer_PrintImage (  
    const UINT8 *pucBitMap,  
    UINT16 Width  
    UINT16 High  
    BOOLEAN PrtImgMode);
```

参数:

- \*pucBitMap** [in] 待打印的图像数据
- Width** [in] 图像的宽度
- High** [in] 图像的高度
- PrtImgMode** [in] 选定的图像打印模式。如果选择单一模式并且 **Width** 参数介于 1-192 (含) 之间, 则输入“FALSE”。如果选择双模式并且 **Width** 参数介于 1-384 (含) 之间, 则输入“TRUE”。

返回:

**BOOLEAN** 此函数返回 TRUE/FALSE, 指示是否成功打印图像