Advanced Card Systems Ltd.
Card & Reader Technologies

PTR89
Portable Thermal Printer

Application Programming Interface V1.00

Subject to change without prior notice

Table of Contents

List of Tables

1.0. Introduction

The ACR89 Handheld Smart Card Reader offers multiple features for the cost of single reliable device. It is equipped with 32-bit CPU running the embedded FreeRT Operating System (FreeRTOS) Kernel. ACS develops the additional feature of having the thermal printer to bring convenience in many types of smart card application. The PTR89 is a portable thermal printer especially designed for ACR89 series.

This document provides the PTR89 API (Application Programming Interface) commands to develop standalone application program with printing support for the ACR89 readers. Application software developers can make use of these APIs to develop their smart-card related application with printing function capabilities. The ACR89 terminal is equipped with 32-bit CPU running the embedded FreeRT Operating System (FreeRTOS) Kernel.

1.1. Scope and Limitation

This API document provides a detailed guide on implementing commands for the thermal printer using ACR89 Series.

1.2. Reference

Refer to this link for the details about the FreeRTOS software environment:

- http://www.freertos.org/

2.0. Printer API Functions

2.1. Data Structures

2.1.1. 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;
```

Used to set the print area under “Page Mode”

Data Member	Value (inclusive)	Description
HorizontalOrigin_X	0 to 383	Starting point in x-axis
VerticalOrigin_Y	0 to 882	Starting point in y-axis
PrintWidth_X	1 to 384	Width of the printing area
PrintHeight_Y	1 to 883	Height of the printing area
ucLineSpace	24 to 255	Line space

Table 1: Page Mode Data Member Description

Notes:

- *HorizontalOrigin_X + PrintMidth_X should be less than or equal to 384.*
- *VerticalOrigin_Y + PrintHeight_Y should be less than or equal to 883.*
- *Horizontal physical origin is equal to HorizontalOrigin_X*0.125 mm from the absolute origin.*
- *Vertical physical origin is equal to VerticalOrigin_Y*0.125 mm from the absolute origin.*
- *The actual width of printing = PrintWidth_X*0.125 mm.*
- *The actual height of printing = PrintHeight_Y*0.125 mm.*
- *The actual line space of printing = ucLineSpace*0.125 mm.*
- *The absolute origin is the upper left of the printable area, and both print width and height cannot be set to 0.*
- *The line spacing includes the height of the font.*

2.2. Functions

The printer can print 24 characters at most in a line. The character font size is 16x18 points, with 0.125 mm point space. The printer has two print modes, i.e. “Standard Mode” and “Page Mode.” Under the “Standard Mode,” the printer will print all the data sent to its buffer, regardless the printing area. On the other hand, under the “Page Mode,” the size of the printing area is needed to be set in advance, and the printing data is limited to 490 bytes. The exceeding data will be discarded.

2.2.1. Printer_Open

This function opens the printer port. Before using this function, the printer port should be in the state of closed, or otherwise, *Printer_Open* will return false.

```
[Printer.h]

BOOLEAN Printer_Open (
            void );
```

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully opening the printer port.

2.2.2. Printer_Close

This function closes the printer port.

```
[Printer.h]

BOOLEAN Printer_Close (
         void );
```

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully closing the printer port.

2.2.3. Printer_Reset

This function resets the printer and restores all user settings to default value.

```
[Printer.h]

void Printer_Reset (
      void );
```

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully resetting the printer.

2.2.4. Printer_FeedPaper

This function feeds paper from the printer.

```
[Printer.h]

BOOLEAN Printer_FeedPaper (
         UINT8 Num );
```

Parameters:

Num [in] Paper space to be fed [0-255, actual space is equal to Num*0.125 mm]

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully feeding paper.

2.2.5. Printer_SetLineSpaceSM

This function sets the line space.

```
[Printer.h]

BOOLEAN Printer_SetLineSpaceSM (
         UINT8 Num );
```

Parameters:

Num [in] Paper space to be fed [24-255, actual space is equal to Num*0.125 mm]

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully feeding paper.

2.2.6. Printer_PrinterStrSM

This function prints a string in the “Standard Mode.” The printing data size should be less than or equal to 65535 bytes and the control character ‘\n’ can be used.

```
[Printer.h]

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

Parameters:

Str [in] Null terminated string of characters to be printed

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully printing the string in the “Standard Mode.”

2.2.7. Printer_PrinterStrPM

This function prints a string in the “Page Mode.” The printing data size should be less than or equal to 490 bytes. If the data size is larger than 490 bytes, the exceeded data will be discarded. The control character ‘\n’ can be used.

```
[Printer.h]

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

Parameters:

Param [in] Printing area to be printed

Str [in] Null terminated string of characters to be printed

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully printing the string in the “Page Mode.”

2.2.8. Printer_PrinterDataSM

This function prints an array of characters in the “Standard Mode.” The control character ‘\n’ can be used.

```
[Printer.h]

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

Parameters:

Data [in] Pointer to the array of characters to be printed

Size [in] Size of the array of characters to be printed [byte]

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully printing the array of data in the “Standard Mode.”

2.2.9. Printer_PrinterDataPM

This function prints the array of data in the “Page Mode.” The printing data size should be less than or equal to 490 bytes. The control character ‘\n’ can be used.

```
[Printer.h]

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

Parameters:

Param [in] Printing area to be printed

Data [in] Pointer to the array of characters to be printed

Size [in] Size of the array of characters to be printed [byte, 1-490]

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully printing the array of data in the "Page Mode."

2.2.10. Printer_PrinterImage

This function prints an image. Each byte represents eight points printed in horizontal direction. The image data is printed one byte by one byte from left to right and from top to bottom in the paper.

```
[Printer.h]

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

Parameters:

pucBitMap [in] Image data to be printed

Width [in] Width of image

High [in] High of image

PrtImgMode [in] Image printing mode to be selected. Input "FALSE" if selecting single mode and the range of Width is between 1 and 192 (inclusive). Input "TRUE" if selecting double mode and the range of Width is between 1 and 384 (inclusive).

Returns:

BOOLEAN This function returns TRUE/FALSE of the state of successfully printing the image.