HiTi

Card Printer CS-2xx series

Software Development Kit

Revision: 1.8.15
Date: 2016/11/17
HiTi Digital, Inc.

1. Overview

1.1. Introduction

The HiTi Card Printer CS2xx Software Development Kit (HiTiCS2xxSDK) is designed for independent software vendors who want to incorporate HiTi card printer driver functionality directly into their applications.

The HiTiCS2xxSDK allows you to control the card printing and card movement for encoding via the HiTiCS2xxSDK API (Application Programming Interface).

The HiTiCS2xxSDK connects printer by USB symbolic link, or by IP address when ethernet module is attached.

1.2. Contents

The HiTiCS2xxSDK contains the following:

HiTiCS2xxSDK.pdf	This document
PavoAPI.h	C/C++ header files
PavoAPI.bas	VB header files
PavoAPI.dll	redistributable DLL
PavoAPI.lib	import library for VC6
Samples	VC6 and VB6 sample applications which use the HiTiCS2xxSDK API

1.3. How To Use The HiTiCS2xxSDK API

To use the HiTiCS2xxSDK API , you should install the PavoAPI.DLL into your private application directory, where your EXE is located. You should not install the DLL into the Windows or Windows System directory, because of possible version conflicts.

Users of your application must have PavoAPI.DLL installed on their system in order for the HiTiCS2xxSDK API to work.

- For VC6, include the header file PavoAPI.H in your application.

  -- You can call LoadLibrary and GetProcAddress to get function pointers of export functions in PavoAPI.DLL.

  -- You can link with PavoAPI.lib to call PavoAPI functions directly.
-

1.4. Support Models

The HiTiCS2xxSDK currently supports following models:

- CS-2xx driver v2.0 or later
- CS-200e with firmware version v1.12 or later
- CS-220e with firmware version v0.97 or later

2. Design Guide

2.1. Encode Card

HiTi Card printer can support data encoding when the encoder modules are present. Please perform encoding by the module position of card path as following:

2.1.1. Contact IC Encoding

To encode a contact IC card, please follow the steps:

1. (Optional) Call PAVO_FindSCardReader() to get the smart card reader name attached in the printer. But this function will return nothing if exteral smart card reader is used.
2. Call PAVO_MoveCard() with position value MOVE_CARD_TO_IC_ENCODER to feed card to the contact encoder station.
3. Perform your encoding procedure. You can use Microsoft PCSC smart card API to encode smart card or use AU9520 Memory Card API at the folder ScardReaderSDK come with HiTiCS2xxSDK to encode memory card.
4. After encoding, if no other encoding or printing to do, you can move card out by calling PAVO_MoveCard() with position value MOVE_CARD_TO_HOPPER.

2.1.2. Contactless IC Encoding

To encode a contactless IC card, please follow the steps:

1. (Optional) Call PAVO_FindMatchedComPort() to get the COM port number of contactless encoder module.
2. Call PAVO_MoveCard() with position value MOVE_CARD_TO_RFID_ENCODER to feed card to the contactless encoder station.
3. Perform your encoding procedure. You can use EWTJ-680 MIFARE Reader API at the folder RFSDK come with HiTiCS2xxSDK to encode contactless card.
4. After encoding, if no other encoding or printing to do, you can move card out by calling PAVO_MoveCard() with position value MOVE_CARD_TO_HOPPER.

2.1.3. Magnetic Stripe Card Encoding

To encode a magnetic stripe card, please follow the steps:

1. Call PAVO_WriteMagTrackData() to write data to tracks.
2. Call PAVO_ReadMagTrackData() to read data from tracks.
3. After encoding, if no printing to do, you can move card out by calling PAVO_MoveCard() with position value MOVE_CARD_TO_HOPPER.

2.2. Print Card Through Driver

2.2.1. General Printing Procedure

To print a card is to create a print job to make HiTi card printer driver to print it out.

Following is the general printing procedure using Windows GDI functions.

1. Call Win32 API CreateDC() or similar method to create a printer DC of the destination printer.
2. Optional, call PAVO_ApplyJobSetting() to indicate driver setting changes. If you do not call this function, the job will use the current printer driver setting.
3. Indicate to begin a print job by calling Win32 API StartDoc()
4. Indicate to begin a page by calling Win32 API StartPage()
5. Draw anything to the printer DC
6. Indicate to end a page by calling Win32 API EndPage()
7. If need to print back side, do step 4, 5, 6.
8. Indicate to end a print job by calling Win32 API EndDoc()
9. Indicate to release the printer DC by calling Win32 API DeleteDC()

2.2.2. Encoding + Printing

You have to encode card first before printing card when you want to do encoding with printing. After encoding card finished, just begin printing procedure. The encoded card will be moved to print position automatically.

2.2.3. Print Resin K

If you want to print K of your own black data, please call PAVO_SetExtraDataToHDC() between StartPage() and your drawing code.

Please notice that the black data must be a 8-bits gray data with size 642x1014 for portrait or 1014x642 for landscape. And the start position must be at (0, 0).

And please notice that the pixel value 0xFF will print out blank, and 0x00 will print out black.

2.3. Receive Messages From Driver

The custom AP can receive some HiTi printer driver's message to get notifications.

To do this, needs some changes in PAVO_JOB_PROPERTY.

- Set hParentWnd as custom AP main UI window handle, and apply flag FF_PARENT_HWND to dwFieldFlag.
- Set dwFlags with PAVO_FLAG_NOT_SHOW_ERROR_MSG_DLG and apply flag FF_FLAGS to dwFieldFlag if custom AP need to know the device error message.
- You can set dwCustomIndex as your own job index to perform your own job control. Set this value need to apply flag FF_CUSTOM_INDEX to dwFieldFlag.
- If your AP want to be notified when the card is printed out by printer, you can use the flag HITI_FLAG_WATCH_JOB_PRINTED. Then AP will receive MSG_JOB_PRINTED.
- The card printer driver will send messages as following:

Message	Value	Meaning
WM_PAVO_PRINTER	0x5555	Message sent by printer driver

WPARAM	Value	WPARAM Meaning	LPARAM meaning
MSG_JOB_BEGIN	1	Driver begin to process job	Spooler assigned job ID or dwCustomIndex
MSG_PRINT_ONE_PAGE	3	Driver process one page	Page number been processing
MSG_PRINT_ONE_COPY	4	Driver send one copy of one page data to printer	Copy number been sending to printer
MSG_JOB_END	6	Driver process job end	Spooler assigned job ID or dwCustomIndex
MSG_DEVICE_STATUS	7	Status of printer	Device Status Code
MSG_JOB_CANCELED	12	Driver cancel the job	Spooler assigned job ID or dwCustomIndex
MSG_JOB_PRINTED	24	Printer print out the job completely	Zero or dwCustomIndex

2.4. Print Card Directly

2.4.1. Normal Printing Procedure

Following is the normal printing procedure:

2. Prepare image to print and then assign to HITI_CARD_PRINT_PARAMETER.
3. Call PAVO_CheckPrinterStatus() to check if printer is in error state.
4. Call PAVO_PrintOneCard() to send image data to printer memory. If printer is in ready state, then it will get into printing state.
5. (Optional) Call PAVO_CheckPrinterStatus() to wait until printer leaving printing state. This step must be used when you want to do next card encoding.

2.4.2. Encoding + Printing

You have to encode card first before printing card when you want to do encoding with printing. After encoding card finished, just begin printing procedure. The encoded card will be moved to print position automatically.

2.5. Check Printer Status

When doing a print job, we sometimes need to check if printer is ready to work.

2.5.1. Check if any device error happened

- Call function: PAVO_CheckPrinterStatus()
- Check the returned status code, if non-zero, means that there are some errors happened or printer is at busy or printing state. See the definitions in Device Status Code.

2.5.2. To wait printer until not busy or printing

- Call function: PAVO_CheckPrinterStatus()
- When the printer is busy, means that the printer is doing some commands. At this time, we recommend programmer to wait the printer to be not busy. Otherwise, send command to printer may cause device in dangerous.
- When you have cards to encoding and printing, please wait printing job complete before encoding next card.
- Please notice that don't call this function too frequently. Because it will cause system loading heavy.
- Recommend waiting interval is half second or longer.

3. Reference

3.1. Structures

3.1.1. PAVO_JOB_PROPERTY

The PAVO_JOB_PROPERTY structure specifies general changeable print settings.

This structure is used by PAVO_ApplyJobSetting().

```
typedef struct tagPAVO_JOB_PROPERTY
{
        DWORD           dwSize;
        DWORD           dwCardType;
        DWORD           dwFlags;
        DWORD           dwDataFlag;

        HWND            hParentWnd;
        HWND            hReserved1;

        char*       pReserved1;
        char*       pReserved2;
        char*       pReserved3;
        char*       pReserved4;

        short           shOrientation;
        short           shCopies;
        short           shReserved1;
        short           shReserved2;

        DWORD           dwCustomIndex;
        DWORD           dwFieldFlag;
        DWORD           dwReserved3;
        DWORD           dwReserved4;

        WORD            wReserved1;
        WORD            wReserved2;
        BYTE            byTransparentCard;
        BYTE            byFlip;
        BYTE            byReserved1;
        BYTE            byCardThick;

        BYTE            byDuplex;
        BYTE            byRibbonType;
        BYTE            byPrintColor;
        BYTE            byDitherK;
        BYTE            byLamin;
        BYTE            byReserved6;
        BYTE            byLaminType;
        BYTE            byRotate180;
} PAVO_JOB_PROPERTY;
```

Member Descriptions:

- dwSize

    Specifies the byte size of this structure. Currently this is 80.

- dwCardType

    Specifies the card type to print out. So that driver will apply print area setting selected on driver UI of the card type.

    This field can be one of the following values:

Name	Value	Meaning
PAVO_CARD_TYPE_BLANK_CARD	0	Blank card
PAVO_CARD_TYPE_SMART_CHIP_6PIN	1	Smart chip card with 6 pins
PAVO_CARD_TYPE_SMART_CHIP_8PIN	2	Smart chip card with 8 pins
PAVO_CARD_TYPE_MAG_STRIP	3	Magnetic strip card
PAVO_CARD_TYPE_CHIP_MAG_STRIP	4	Smart chip card with 8 pins that has magnetic strip at back side
PAVO_CARD_TYPE_ADHESIVE_CARD	5	Blank card with adhesive at back side, only front side is printable.

- dwFlags

    Specifies the print out control settings.

    This field can be combination of the following values:

Name	Value	Meaning
PAVO_FLAG_NOT_SHOW_ERROR_MSG_DLG	0x00000001	Not show error message dialog, then error message will be sent to AP if hParentWnd is specified.
PAVO_FLAG_WAIT_MSG_DONE	0x00000002	Make driver to wait until AP process the message done. Otherwise driver will just cancel the job immediately after message is sent to AP.
PAVO_FLAG_NOT_SHOW_CLEAN_MSG	0x00000100	Not show clean message dialog.
PAVO_FLAG_WATCH_JOB_PRINTED	0x00000400	Indicate driver to notify AP when print card completely
PAVO_FLAG_MOVE_CARD_TO_STANDBY_AFTER_PRINTED	0x00010000	Move card to standby position after printing finished

- dwDataFlag

    Specifies the extra data type to pass to printer driver by calling PAVO_SetExtraDataToHDC().

    This field can be combination of the following values:

Name	Value	Meaning
PAVO_DATAFLAG_RESIN_FRONT	0x00000002	Resin K data of front side
PAVO_DATAFLAG_RESIN_BACK	0x00000010	Resin K data of back side

- hParentWnd

    Specifies the window handle to receive messages sent by printer driver.

- shOrientation

Specifies the print out orientation of the job.

1 for portrait and 2 for landscape.

- shCopies

    Specifies the print out copies of the job.

- dwCustomIndex

    Specify the job index generated by your program. This is used to help you to indicate the messages sent by driver is for which job.

- dwFieldFlag

    Specifies the field of this structure to be applied to printer DC.

    This field can be combination of the following values:

Name	Value	Meaning
FF_CARD_TYPE	0x00000001	dwCardType will be applied
FF_FLAGS	0x00000002	dwFlags will be applied
FF_DATA_FLAG	0x00000004	dwDataFlag will be applied
FF_PARENT_HWND	0x00000008	hParentWnd will be applied
FF_ORIENTATION	0x00000020	shOrientation will be applied
FF_COPIES	0x00000040	shCopies will be applied
FF_CUSTOM_INDEX	0x00000080	dwCustomIndex will be applied
FF_DUPLEX	0x00000100	byDuplex will be applied
FF_RIBBON_TYPE	0x00000200	byRibbonType will be applied
FF_PRINT_COLOR	0x00000400	byPrintColor will be applied
FF_DITHER_K	0x00000800	byDitherK will be applied
FF_LAMIN	0x00001000	byLamin will be applied
FF_LAMIN_TYPE	0x00002000	byLaminType will be applied
FF_ROTATE180	0x00004000	byRotate180 will be applied
FF_CARD_THICK	0x00010000	byCardThick will be applied
FF_ALL_FIELDS	0xFFFFFFFF	All fields will be applied

- byTransparentCard

    For CS-220e only.

    0 = use white card for printing

    1 = use transparent card for printing

- byFlip

    For CS-220e only.

    0 = not flip the output image for transparent card

    1 = flip the output image for transparent card

- byCardThick

    Specifies card thickness. For CS-2xx only.

    0 = 0.3mm

    1 = 0.5mm

    2 = 0.8mm

3 = 1.0mm

- byDuplex

  Specifies the sides to be printed of the job.

  This field can be combination of the following values:

Name	Value	Meaning
PAVO_DUPLEX_PRINT_FRONT_SIDE	0x01	Front side will be printed
PAVO_DUPLEX_PRINT_BACK_SIDE	0x02	Back side will be printed

- byRibbonType

  Specifies the sides to be printed of the job.

  This field can be one of the following values:

Name	Value	Meaning
PAVO_RIBBON_TYPE_YMCKO	0	YMCKO ribbon
PAVO_RIBBON_TYPE_K	1	K ribbon
PAVO_RIBBON_TYPE_KO	3	KO ribbon
PAVO_RIBBON_TYPE_YMCKOK	4	YMCKOK ribbon
PAVO_RIBBON_TYPE_HALF_YMCKO	5	1/2 YMCKO ribbon
PAVO_RIBBON_TYPE_YMCKFO	12	YMCKFO ribbon

- byPrintColor

  Specifies which side needs color printing of the job.

  This field is dependent on byDuplex.

  0x01 for print YMCO at front side. 0x02 for print YMCO at back side. 0x03 for print YMCO at both sides.

- byDitherK

  Specifies which side needs dither K printing of the job.

  This field is dependent on byDuplex.

  0x01 for print K with dither at front side. 0x02 for print K with dither at back side. 0x03 for print K with dither at both sides.

- byLamin

  Specifies which side needs to print lamination of the job.

  This field is independent on byDuplex.

  0x01 for print lamination at front side. 0x02 for print lamination at back side. 0x03 for print lamination at both sides.

- byLaminType

  Specifies lamination ribbon type.

  0x00=HiTi Standard Overlay.

  0x01=HiTi Standard Patch.

- byRotate180

  Specifies which side needs to do Rotate 180.

  0x00=Not rotate, 0x01=front side, 0x02=back side, 0x03=both sides.

3.1.2. HITI_CARD_PRINT_PARAMETER

The HITI_CARD_PRINT_PARAMETER structure specifies image data and heating energy for printing.

This structure is used by PAVO_PrintOneCard().

```
typedef struct tagHITI_CARD_PRINT_PARAMETER
{
        unsigned long       dwSize;
        unsigned long       dwReserve1;
        unsigned long       dwReserve2;
        unsigned long       dwReserve3;

        unsigned char       byOrientation;
        unsigned char       byCardThickness;
        unsigned char       byTransparentCard;
        unsigned char       byReserve4;
        unsigned char       byReserve5;
        unsigned char       byReserve6;
        unsigned char       byReserve7;
        unsigned char       byReserve8;

        BITMAP              *lpFrontBGR;
        BITMAP              *lpFrontK;
        BITMAP              *lpFrontO;

        BITMAP              *lpBackBGR;
        BITMAP              *lpBackK;
        BITMAP              *lpBackO;

        unsigned char       *lpReserve1;
        unsigned char       *lpReserve2;

} HITI_CARD_PRINT_PARAMETER;
```

Member Descriptions:

- dwSize

  Specifies the byte size of this structure.
- byOrientation

  Specifies the image data orientation. 1=portrait. 2=landscape.
- byCardThickness

  Specifies the card thickness. 0=0.3mm. 1=0.5mm. 2=0.8mm. 3=1.0mm.
- byTransparentCard

  Specifies to print by transparent card. 0 = No. 1 = Yes.
- lpFrontBGR

  Specifies the color data to be printed on front side. In fact, it is a BITMAP structure.

It is 24-bits per pixel. Byte order is BGRBGRBGR.

For portrait printing, its width is 642, height is 1014.

For landscape printing, its width is 1014, height is 642.

- lpFrontK

  Specifies the K data to be printed on front side. In fact, it is a BITMAP structure.

  It is 8-bits per pixel. Each pixel with value 0x00 will be transfer to card by K panel.

  For portrait printing, its width is 642, height is 1014.

  For landscape printing, its width is 1014, height is 642.

- lpFrontO

  Specifies the O data to be printed on front side. In fact, it is a BITMAP structure.

  It is 8-bits per pixel. Each pixel with value 0xFF will be transfer to card by O panel.

  For portrait printing, its width is 642, height is 1014.

  For landscape printing, its width is 1014, height is 642.

- lpBackBGR

  Specifies the color data to be printed on back side. In fact, it is a BITMAP structure.

  It is 24-bits per pixel. Byte order is BGRBGRBGR.

  For portrait printing, its width is 642, height is 1014.

  For landscape printing, its width is 1014, height is 642.

- lpBackK

  Specifies the K data to be printed on back side. In fact, it is a BITMAP structure.

  It is 8-bits per pixel. Each pixel with value 0x00 will be transfer to card by K panel.

  For portrait printing, its width is 642, height is 1014.

  For landscape printing, its width is 1014, height is 642.

- lpBackO

  Specifies the O data to be printed on back side. In fact, it is a BITMAP structure.

  It is 8-bits per pixel. Each pixel with value 0xFF will be transfer to card by O panel.

  For portrait printing, its width is 642, height is 1014.

  For landscape printing, its width is 1014, height is 642.

  Specifies the print out copies of the job.

3.1.3. HITI_HEATING_ENERGY

The HITI_HEATING_ENERGY structure specifies heating energy for printing.

This structure is used by PAVO_PrintOneCard().

```
typedef struct tagHITI_HEATING_ENERGY
{
    //Heating Energy, value range of following
    //fields are -127 ~ 127
    //Density Adjustment ------------------------
    //Front side
    char                    chFrontDenYMC;
    char                    chFrontDenK;
    char                    chFrontDenO;
    char                    chFrontDenResinK;

    //Back side
    char                    chBackDenYMC;
    char                    chBackDenK;
    char                    chBackDenO;
    char                    chBackDenResinK;

    //Sensitivity Adjustment --------------------
    //Front side
    char                    chFrontSenYMC;
    char                    chFrontSenK;
    char                    chFrontSenO;
    char                    chFrontSenResinK;

    //Back side
    char                    chBackSenYMC;
    char                    chBackSenK;
    char                    chBackSenO;
    char                    chBackSenResinK;
}HITI_HEATING_ENERGY;
```

- Heating Energy fields

  Specify heating energy adjustment same as driver UI Heating Energy tab.

  The value range of these fields are -127 ~ 127.

-

3.1.4. MAG_TRACK_DATA2

The MAG_TRACK_DATA2 structure specifies the track data for magnetic stripe card encoding or reading.

This structure can be used by PAVO_ReadMagTrackData() or PAVO_WriteMagTrackData().

```
typedef struct tagMAG_TRACK_DATA2
{
        char                    szTrack1[256];
        char                    szTrack2[256];
        char                    szTrack3[256];

        unsigned char           byTrackFlag;
        unsigned char           byEncodeMode;
        unsigned char           byCoercivity;
        unsigned char           byT2BPI;

        unsigned char           byRawLenT1;
        unsigned char           byRawLenT2;
        unsigned char           byRawLenT3;
        unsigned char           byReserve[25];

}MAG_TRACK_DATA2;
```

Member Descriptions:

- szTrack1

  This is the buffer of track 1 data to be encoded to or read from magnetic stripe card.

  The track 1 allow 78 characters maximum. The allowed characters are: between {},

  { !"#$&'()*+,-./0123456789:;<=>@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_}.

  Start Sentinel is '%', End Sentinel is '?', both will be auto added by encoder.

- szTrack2

  This is the buffer of track 2 data to be encoded to or read from magnetic stripe card.

  The track 2 allow 39 characters maximum. The allowed characters are between {},

  {0123456789:<=>}.

  Start Sentinel is ';', End Sentinel is '?', both will be auto added by encoder.

- szTrack3

  This is the buffer of track 3 data to be encoded to or read from magnetic stripe card.

  The track 2 allow 39 characters maximum. The allowed characters are between {},

  {0123456789:<=>}.

Start Sentinel is ';', End Sentinel is '?', both will be auto added by encoder.

- byTrackFlag

  Specify the track to be used. 0x01=track1, 0x02=track2, 0x04=track3. Can be combined of 2 or 3 tracks.

- byEncodeMode

  Not been used currently.

- byCoercivity

  Specify the coercivity to be used for encoding. 0=Lo-Co, 1=Hi-Co.

- byT2BPI

  Specify the BPI of track 2 to be used. Can be 75 or 210, if not set it, will use 75.

3.2. Functions

3.2.1. PAVO_ApplyJobSetting

Apply settings to the printer DC for print job.

DWORD __stdcall **PAVO_ApplyJobSetting**(char* szPrinter, HDC hDC, BYTE* lpInDevMode, BYTE* lpInJobProp);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinterName

[in] Printer name shown in the printer folder of the printer.

hDC

[in] Handle of the printer DC created by CreateDC.

For VB, this can be the member hdc of printer object. Such as Printer.hdc.

For Delphi, this can be Printer.Canvas.Handle.

lpInDevMode

[in] Currently this must be zero.

lpInJobProp

[in] Pointer of the PAVO_JOB_PROPERTY structure that specify settings to be changed.

Remarks

This function will combine the printer's current setting shown in the printer folder with the PAVO_JOB_PROPERTY structure. Then will call ResetDC() to apply changes to the printer.

If you want to use some setting not defined in PAVO_JOB_PROPERTY, please change the setting through printer UI and then save it.

3.2.2. PAVO_CheckPrinterStatus

Get the device status of the specified printer.

DWORD __stdcall **PAVO_CheckPrinterStatus**(char* szPrinter, DWORD *lpdwStatus);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

lpdwStatus

[out] Pointer to the buffer to receive the status of the printer. Possible values are defined in Device Status Code.

Remarks

3.2.3. PAVO_DoCommand

Perform HiTi defined commands.

DWORD __stdcall **PAVO_DoCommand**(char* szPrinter, DWORD dwCommand);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

dwCommand

[in] Specify to command to do by printer.

Name of dwCommand	Value	Meaning
PAVO_COMMAND_RESET_PRINTER	100	Reset Printer to initial state
PAVO_RIBBON_PAVO_COMMAND_CLEAN_CARD_PATH	105	Clean card path
PAVO_COMMAND_RESET_PRINT_COUNT	106	Reset print count
PAVO_COMMAND_FLIP_CARD	202	Flip card by flipper

Remarks

When dwCommand=PAVO_RIBBON_PAVO_COMMAND_CLEAN_CARD_PATH, this function will return immediately. And printer will perform card feeding to do clean card path.

After this action, call PAVO_CheckDeviceStatus() to wait until busy state is cleared.

3.2.4. PAVO_FindCOMPort

Find the COM port number of magnetic strip module and RF module attached to the specified card printer.

DWORD __stdcall **PAVO_FindComPort**(char* szPrinter, DWORD* lpdwMagPort, DWORD* lpdwRFPort);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

lpdwMagPort

[out] Not used. Because CS-2xx has no magnetic COM port.

lpdwRFPort

[out] Pointer to the buffer to receive COM port number of RFID module.

Remarks

3.2.5. PAVO_FindSCardReader

Find the smart card reader name attached to the specified card printer.

DWORD __stdcall **PAVO_FindSCardReader**(char* szPrinter, char* szReaderName);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

szReaderName

[out] Pointer to the buffer to receive smart card reader name.

Remarks

3.2.6. PAVO_GetDeviceInfo

Get hardware information of printer.

DWORD __stdcall **PAVO_GetDeviceInfo**(char* szPrinter, DWORD dwInfoType, BYTE *lpInfoData, DWORD *lpdwDataLen);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

dwInfoType

[in] Indicate which kind information to get from printer.

Name	Va lue	Device info meaning	Data type in lpInfoData	Data length
PAVO_DEVINFO_MFG_SERIAL	1	Manufacture serial number	char array	Number of char
PAVO_DEVINFO_MODEL_NAME	2	Printer model name	char array	Number of char
PAVO_DEVINFO_FIRMWARE_VERSION	3	Printer firmware version	char array	Number of char
PAVO_DEVINFO_RIBBON_INFO	4	Current ribbon information	DWORD array	2 DWORDs
PAVO_DEVINFO_PRINT_COUNT	5	Number of printed cards	DWORD	1 DWORD
PAVO_DEVINFO_CARD_POSITION	6	Current card position	DWORD	1 DWORD

lpInfoData

[in, out] Pointer to the buffer to receive printer information.

lpdwDataLen

[in, out] Indicate the buffer size. And return the information data size.

Remarks

For dwInfoType is PAVO_DEVINFO_CARD_POSITION, the position value is as following:

Card position value	Card position
0	Out of printer
1	Start printing position
2	Mag out position
3	Mag in position
4	Contact encoder position
5	Contactless encoder position
6	Flipper position
7	Card jam

3.2.7. PAVO_MoveCard

Move card to the specified position in card printer.

DWORD __stdcall **PAVO_MoveCard**(char* szPrinter, DWORD dwPosition);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

dwPosition

[in] Card position to move in card printer.

Position Name	Value	Description
MOVE_CARD_TO_IC_ENCODER	1	Move Card to Contact Encoder Station
MOVE_CARD_TO_RFID_ENCODER	3	Move Card to Contactless Encoder Station
MOVE_CARD_TO_REJECT_BOX	4	Move Card to Reject Box (at bottom of new flipper)
MOVE_CARD_TO_HOPPER	5	Move Card to Output Hopper
MOVE_CARD_TO_FLIPPER	6	Move Card to Flipper
MOVE_CARD_TO_PRINT_FROM_FLIPPER	7	Move Card to Print Position from Flipper
MOVE_CARD_TO_STANDBY_POSITION	10	Move Card to Standby Position

Remarks

3.2.8. PAVO_PrintOneCard

Send one card image data to printer memory. If printer is at ready state, it will perform printing immediately.

DWORD __stdcall **PAVO_PrintOneCard**(char* szPrinter, HITI_CARD_PRINT_PARAMETER *lpJobPara, HITI_HEATING_ENERGY *lpHeatEnergy, unsigned char* lpReserved);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

lpJobPara

[in] Pointer to the buffer of HITI_CARD_PRINT_PARAMETER that specify image data to print out.

lpHeatEnergy

[in] Pointer to the buffer of HITI_HEATING_ENERGY that specify to adjust heating energy of each plane when print out.

lpReserved

Not used.

Remarks

3.2.9. PAVO_ReadMagTrackData

Perform magnetic strip card reading function.

DWORD __stdcall **PAVO_ReadMagTrackData**(char* szPrinter, DWORD dwCOM, BYTE* lpTrackData);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

dwCOM

[in] Not used.

lpTrackData

[out] Pointer to the buffer of MAG_TRACK_DATA2 structure to receive the tracks data.

Remarks

3.2.10. PAVO_SetExtraDataToHDC

This function allows you to set non-RGB data to print.

DWORD __stdcall **PAVO_SetExtraDataToHDC**(HDC hDC, DWORD dwType, DWORD x, DWORD y, BITMAP* lpBmp);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

hDC

[in] Handle of the printer DC.

For VB, this can be the member hdc of printer object. Such as Printer.hdc.

For Delphi, this can be Printer.Canvas.Handle.

dwType

[in] Specify the data type.

x, y

[in] Must be zero.

lpBmp

[in] Pointer to the buffer of BITMAP structure that has the image information to set to printer. For resin K or O, the image data with bmBitsPixel=8 is MUST.

Remarks

3.2.11. PAVO_WriteMagTrackData

Perform magnetic strip card writing function.

DWORD __stdcall **PAVO_WriteMagTrackData**(char* szPrinter, DWORD dwCOM, BYTE* lpTrackData);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

dwCOM

[in] Not used.

lpTrackData

[in] Pointer to the buffer of MAG_TRACK_DATA2 structure with tracks data to be encoded.

Remarks

This function does NOT verify the encoded data. You can verify data by calling PAVO_ReadMagTrackData().

3.2.12. PAVO_SetPassword

Set password that will be stored in printer used for authentication before lock or unlock printer.

DWORD __stdcall **PAVO_SetPassword**(char* szPrinter, char* szCurrentPasswd, char* szNewPasswd);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

szCurrentPasswd

[in] The password currently saved in printer. Leave it empty for the first time setting password.

szNewPasswd

[in] The new password to be saved in printer.

Remarks

The password can be 4 characters of A-Z, a-z and 0-9.

3.2.13. PAVO_SetSecurityMode

Set to lock or unlock printer.

DWORD __stdcall **PAVO_SetSecurityMode**(char* szPrinter, char* szCurrentPasswd, long nSecurityMode);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

szCurrentPasswd

[in] The password currently saved in printer.

nSecurityMode

[in] 0 = unlock printer. 1= lock printer.

Remarks

3.2.14. PAVO_SetStandbyParameters

Set standby time and position after card is printed.

DWORD __stdcall **PAVO_SetStandbyParameters**(char* szPrinter, BYTE byStandbyPos, BYTE byStandbyTime);

Return Value

Return zero if no error happened.

Otherwise return WIN32 system error code.

Parameters

szPrinter

[in] Printer queue name or USB symbolic link or IP address.

byStandbyPos

[in]

byStandbyTime

[in]

Remarks

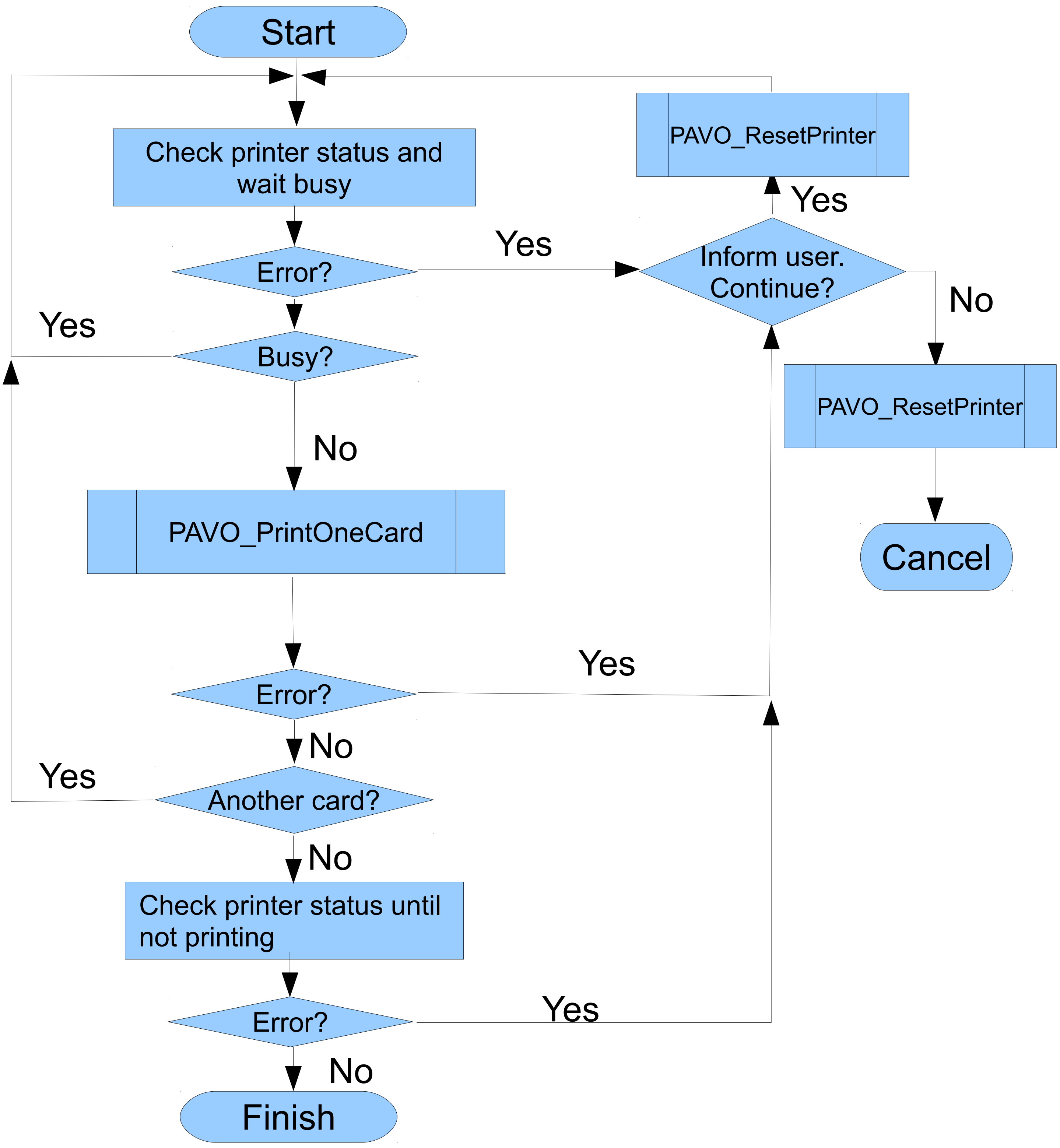
3.3. Device Status Code

The meanings of HiTi defined device status code are as following:

Name	Value	Description
PAVO_DS_BUSY	0x00080000	Device is at busy state
PAVO_DS_OFFLINE	0x00000080	Device is disconnected or power off
PAVO_DS_PRINTING	0x00000002	Printer is printing
PAVO_DS_PROCESSING_DATA	0x00000005	Driver is processing print data
PAVO_DS_SENDING_DATA	0x00000006	Driver is sending data to printer
PAVO_DS_CARD_MISMATCH	0x000100FE	Card mismatch
PAVO_DS_CMD_SEQ_ERROR	0x000301FE	Command sequence error
PAVO_DS_SRAM_ERROR	0x00030001	SRAM error
PAVO_DS_SDRAM_ERROR	0x00030101	SDRAM error
PAVO_DS_ADC_ERROR	0x00030201	ADC error
PAVO_DS_NVRAM_ERROR	0x00030301	NVRAM R/W error
PAVO_DS_SDRAM_CHECKSUM_ERROR	0x00030302	Check sum error - SDRAM
PAVO_DS_FW_WRITE_ERROR	0x00030701	Firmware write error
PAVO_DS_COVER_OPEN	0x00050001	Cover or door open or Ajar
PAVO_DS_COVER_OPEN_SLAVE	0x00050201	Cover or door open or Ajar in slave printer
PAVO_DS_REJECT_BOX_MISSING	0x00050301	Rejected box missing
PAVO_DS_REJECT_BOX_FULL	0x00050401	Rejected box full
PAVO_DS_CARD_OUT	0x00008000	Card out or feeding error
PAVO_DS_CARD_LOW	0x00008001	Card low
PAVO_DS_RIBBON_MISSING	0x00080004	Ribbon missing
PAVO_DS_OUT_OF_RIBBON	0x00080103	Out of ribbon
PAVO_DS_RIBBON_IC_RW_ERROR	0x000804FE	Ribbon IC R/W error
PAVO_DS_UNSUPPORT_RIBBON	0x000806FE	Unsupported ribbon
PAVO_DS_UNKNOWN_RIBBON	0x000808FE	Unknown ribbon
PAVO_DS_RIBBON_MISSING_SLAVE	0x00080204	Ribbon missing in slave printer
PAVO_DS_OUT_OF_RIBBON_SLAVE	0x00080303	Out of ribbon in slave printer
PAVO_DS_RIBBON_IC_RW_ERROR_SLAVE	0x000805FE	Ribbon IC R/W error in slave printer
PAVO_DS_UNSUPPORT_RIBBON_SLAVE	0x000807FE	Unsupported ribbon in slave printer
PAVO_DS_UNKNOWN_RIBBON_SLAVE	0x000809FE	Unknown ribbon in slave printer
PAVO_DS_CARD_JAM1	0x00030000	Card jam in card path
PAVO_DS_CARD_JAM2	0x00040000	Card jam in flipper
PAVO_DS_CARD_JAM3	0x00050000	Card jam in eject box
PAVO_DS_CARD_JAM4	0x00030100	Card jam in card path of slave printer
PAVO_DS_CARD_JAM5	0x00040100	Card jam in flipper of slave printer
PAVO_DS_CARD_JAM6	0x00050100	Card jam in eject box of slave printer
PAVO_DS_CARD_JAM7	0x00060100	Card jam between master and slave
PAVO_DS_CARD_JAM8	0x00070100	Card jam in end of master
PAVO_DS_WRITE_FAIL	0x0000001F	Send command to printer fail
PAVO_DS_READ_FAIL	0x0000002F	Get response from printer fail
PAVO_DS_CARD_THICK_WRONG	0x00008010	Card thickness selector is not placed in the right position
PAVO_DS_NO_FLIPPER_MODULE	0x1100000D	The flipper module is not attached
SCARD_E_NO_SMARTCARD	0x8010000C	The operation requires a Smart Card, but no Smart Card is currently in the device.
PAVO_DS_LOCKED	0x00007540	Printer is locked.

ERROR_MAGCARD_CONNECT_FAIL	1850	Cannot connect COM port of magnetic module
ERROR_MAGCARD_READ_FAIL	1851	Read track data fail
ERROR_MAGCARD_WRITE_FAIL	1852	Write track data fail
ERROR_MAGCARD_NO_TRACK_SELECTED	1853	No track is specified to be read/write
ERROR_MAGCARD_EMPTY_TRACK_DATA	1855	One of the track data is empty
ERROR_MAGCARD_NO_MODULE	1856	The magnetic stripe encoder module is not attached
PAVO_DS_0100_COVER_OPEN	0x00000100	0100 Cover open
PAVO_DS_0101_FLIPPER_COVER_OPEN	0x00000101	0101 Flipper cover open
PAVO_DS_0200_IC_MISSING	0x00000200	0200 IC chip missing
PAVO_DS_0201_RIBBON_MISSING	0x00000201	0201 Ribbon missing
PAVO_DS_0202_RIBON_MISMATCH	0x00000202	0202 Ribbon mismatch
PAVO_DS_0203_RIBBON_TYPE_ERROR	0x00000203	0203 Ribbon type error
PAVO_DS_0300_RIBBON_SEARCH_FAIL	0x00000300	0300 Ribbon search fail
PAVO_DS_0301_RIBBON_OUT	0x00000301	0301 Ribbon out
PAVO_DS_0302_PRINT_FAIL	0x00000302	0302 Print fail
PAVO_DS_0303_PRINT_FAIL	0x00000303	0303 Print fail
PAVO_DS_0304_RIBBON_OUT	0x00000304	0304 Ribbon out
PAVO_DS_0400_CARD_OUT	0x00000400	0400 Card out
PAVO_DS_0500_CARD_JAM	0x00000500	0500 Card jam
PAVO_DS_0501_CARD_JAM	0x00000501	0501 Card jam
PAVO_DS_0502_CARD_JAM	0x00000502	0502 Card jam
PAVO_DS_0503_CARD_JAM	0x00000503	0503 Card jam
PAVO_DS_0504_CARD_JAM	0x00000504	0504 Card jam
PAVO_DS_0505_CARD_JAM	0x00000505	0505 Card jam
PAVO_DS_0506_CARD_JAM	0x00000506	0506 Card jam
PAVO_DS_0507_CARD_JAM	0x00000507	0507 Card jam
PAVO_DS_0508_CARD_JAM	0x00000508	0508 Card jam
PAVO_DS_0600_CARD_MISMATCH	0x00000600	0600 Card mismatch
PAVO_DS_0700_CAM_ERROR	0x00000700	0700 Cam error
PAVO_DS_0800_FLIPPER_ERROR	0x00000800	0800 Flipper error
PAVO_DS_0801_FLIPPER_ERROR	0x00000801	0801 Flipper error
PAVO_DS_0802_FLIPPER_ERROR	0x00000802	0802 Flipper error
PAVO_DS_0803_FLIPPER_ERROR	0x00000803	0803 Flipper error
PAVO_DS_0900_NVRAM_ERROR	0x00000900	0900 NVRAM error
PAVO_DS_1000_RIBBON_ERROR	0x00001000	1000 Ribbon error
PAVO_DS_1100_RBN_TAKE_CALIB_FAIL	0x00001100	1100 RBN Take Calibration Failed
PAVO_DS_1101_RBN_SUPPLY_CALIB_FAIL	0x00001101	1101 RBN Supply Calibration Failed
PAVO_DS_1200_ADC_ERROR	0x00001200	1200 ADC error
PAVO_DS_1300_FW_ERROR	0x00001300	1300 FW error
PAVO_DS_1301_FW_ERROR	0x00001301	1301 FW error
PAVO_DS_1400_POWER_SUPPLY_ERROR	0x00001400	1400 Power supply error

3.4. Diagram of printing flow


Start
Check printer status and wait busy
Error?
Yes
Inform user. Continue?
Yes
PAVO_ResetPrinter
No
PAVO_ResetPrinter
Cancel
Busy?
Yes
No
PAVO_PrintOneCard
Error?
Yes
No
Another card?
Yes
No
Check printer status until not printing
Error?
Yes
No
Finish