

1. General

A set of functions dealing with initialization of the printer and the communication settings, various printer statuses, etc

1.1. DPSDKLibOpen

Opens the library, creates and initializes the globals. This function must be called prior to any other library functions. If **DPSDKLibOpen** fails, do not call any other library functions. If **DPSDKLibOpen** succeeds, call **DPSDKLibClose** when you are done using the library to release critical system resources.

Syntax

```
Err DPSDKLibOpen(UInt16 refNum, UInt32 *clientContextP)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

clientContextP

pointer to variable for returning client context. The client context is used to maintain client-specific data for multiple client support. The value returned here will be used as a parameter for other library functions, which require a client context

Return value

errNone

memErrNotEnoughSpace

Comments

1.2. DPSDKLibClose

Closes the library, frees client context and globals.

Syntax

```
Err DPSDKLibClose(UInt16 refNum, UInt32 clientContext)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

clientContext

client context

Return value

errNone

DPSDKLibErrStillOpen

Comments

1.3. DPSDK_InitPrinter

Initializes the printer library and opens the connection.

Syntax

```
Boolean DPSDK_InitPrinter(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, otherwise false

Comments

1.4. DPSDK_DonePrinter

Frees allocated memory chunks and closes the connection.

Syntax

```
void DPSDK_DonePrinter(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Comments

1.5. DPSDK_GetPrinterType

Returns the currently selected printer type.

Syntax

```
int DPSDK_GetPrinterType(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Printer type, one of the following:

Value	Meaning
PRINTER_PP50	Datecs PP-50
PRINTER_PP55	Datecs PP-55
PRINTER_CMP10	Citizen CMP-10
PRINTER_DPP350	Datecs DPP-350
PRINTER_DPP250	Datecs DPP-250

Comments

1.6. DPSDK_SetPrinterType

Sets the printer type, this can be done only if the connection is not open.

Syntax

```
BOOL DPSDK_SetPrinterType(UInt16 refNum, UInt8 Printer)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Printer

Printer type, one of the following:

Value	Meaning
PRINTER_PP50	Datecs PP-50
PRINTER_PP55	Datecs PP-55
PRINTER_CMP10	Citizen CMP-10
PRINTER_DPP350	Datecs DPP-350
PRINTER_DPP250	Datecs DPP-250

Return value

True if success, otherwise false

Comments

Useful for implementing your own Printer Setup screen or similar. Be sure you call this function before [DPSDK_InitPrinter](#) or call [DPSDK_DonePrinter](#) first. The SDK will remember this setting and the next time it's opened will try to use it.

1.7. DPSDK_GetConnectionType

Returns the currently selected connection type.

Syntax

```
int DPSDK_GetConnectionType(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Connection Type, one of the following:

Value	Meaning
CON_SERIAL	Serial Communication
CON_USB	USB Communication
CON_IRDA	Infrared Communication
CON_BLUETOOTH	Bluetooth Communication

Comments

1.8. DPSDK_SetConnectionType

Sets the connection type, this can be done only if the connection is not open.

Syntax

```
BOOL DPSDK_SetConnectionType(UInt16 refNum, UInt8 Connection)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Connection

Connection Type, one of the following:

Value	Meaning
CON_SERIAL	Serial Communication
CON_USB	USB Communication
CON_IRDA	Infrared Communication
CON_BLUETOOTH	Bluetooth Communication

Return value

True if success, otherwise false

Comments

Useful for implementing your own Printer Setup screen or similar. Be sure you call this function before [DPSDK_InitPrinter](#) or call [DPSDK_DonePrinter](#) first. The SDK will remember this setting and the next time it's opened will try to use it.

1.9. DPSDK_GetConnectionSpeed

Returns the currently selected connection speed.

Syntax

```
int DPSDK_GetConnectionSpeed(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Speed - communication speed, one of

Value	Meaning
Speed1200	1200Bps
Speed2400	2400Bps
Speed4800	4800Bps
Speed9600	9600Bps
Speed19200	19200Bps
Speed38400	38400Bps
Speed57600	57600Bps
Speed115200	115200Bps

Comments

1.10. DPSDK_SetConnectionSpeed

Sets the connection speed, this can be done only if the connection is not open.

Syntax

```
BOOL DPSDK_SetConnectionSpeed(UInt16 refNum, UInt8 Speed)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Speed

communication speed, one of

Value	Meaning
Speed1200	1200Bps
Speed2400	2400Bps
Speed4800	4800Bps
Speed9600	9600Bps
Speed19200	19200Bps
Speed38400	38400Bps
Speed57600	57600Bps
Speed115200	115200Bps

Return value

True if success, otherwise false

Comments

Useful for implementing your own Printer Setup screen or similar. Be sure you call this function before [DPSDK_InitPrinter](#) or call [DPSDK_DonePrinter](#) first. The SDK will remember this setting and the next time it's opened will try to use it.

1.11. DPSDK_ChangePrinterSpeed

Changes the communication speed the printer is working on.

Syntax

BOOL DPSDK_ChangePrinterSpeed(UINT16 refNum, UINT8 Speed)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Speed

communication speed, one of

Value	Meaning
Speed1200	1200Bps
Speed2400	2400Bps
Speed4800	4800Bps
Speed9600	9600Bps
Speed19200	19200Bps
Speed38400	38400Bps
Speed57600	57600Bps
Speed115200	115200Bps

Return value

True if success, otherwise false

Be sure that you do NOT mistake this function with [DPSDK_SetConnectionSpeed](#)!

Comments

This function changes the communication speed the printer is currently using, i.e. you can set it to work at 9600 if you want. The printer will remember this and will continue to use unless the battery is completely dead. So if you set it to 9600 you'll not be able to connect to it back on 57600, you have to use [DPSDK_SetConnectionSpeed](#) to set the sdk to 9600 so you can revert back to what you want. If the change is successful this function will reset the sdk settings so they matches the new connection speed so don't worry about that. Oh and I forgot - you can use this function only after the printer has been initialized. You'll see that a line of text will be printed to confirm the connection change. If you don't like that tell me and I'll remove it.

1.12. DPSDK_GetStatus

All-in-one function for various printer information

Syntax

```
Int32 DPSDK_GetStatus(UInt16 refNum, UInt8 statcmd)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

statcmd

which status do you want to get, one of the following:

Value	Meaning
STAT_ISCONNECTED	Returns 1 if the printer is connected and communication active
STAT_GETBATVOLT	Get battery level in voltage (v*10, i.e. 65 means 6.5v)
STAT_GETBATPERCENT	Get battery level in percents (see comments)
STAT_GETTEMPC	Get printer head temperature in Celsius
STAT_GETTEMPFR	Get printer head temperature in Fahrenheit
STAT_HASPAPER	Returns 1 if the printer has paper, 0 if not
STAT_GETVERSION	Returns the printer's firmware version, must be connected prior that
STAT_GETTYPE	Returns the printer model

Return value

The result of the specific function according to statcmd or one of these error codes:

Value	Meaning
StatErrNotImpl	The status you have requested is not implemented yet
StatErrNotConnected	This error can be returned when some of the status commands requires an active connection to the printer, like STAT_GETVERSION

Comments

The battery % remaining calculation can never be accurate, as the battery spends 70% of its charge on same voltage. This function tries to return most accurate value based on best guess but it is impossible to get exact value

1.13. DPSDK_ControlPanel

Fires a control panel form which lets the user select a printer model, connection type, connection speed, warnings, detect the printer, etc.

The control panel is not anymore built in the sdk, but rather a standalone program - DPSDKConfig.prc So it must be present for the function to work. The function returns immediatelly before configuration have finished.

Syntax

BOOL DPSDK_ControlPanel(UInt16 refNum, UInt16 flags)

Parameters

refNum

library reference number obtained via call to SysLibLoad

flags

Return value

Returns True if the config program can be found.

Comments

1.14. DPSDK_GetROMSerialNumber

Returns the printer unique serial number

Syntax

BOOL DPSDK_GetROMSerialNumber(UInt16 refNum, char *buffer)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buffer

an array that will hold the 9-byte serial number (8 digits+0 for terminating). An example serial number is "01228323"

Return value

True if success, otherwise false

Comments

1.15. DPSDK_GetLibraryInfo

Returns library information such as version, build data, etc.

Syntax

Int32 DPSDK_GetLibraryInfo(UInt16 refNum, UInt32 param)

Parameters

refNum

library reference number obtained via call to SysLibLoad

param

tells which library info you want to get, one of the following:

Value	Meaning
INFO_RELEASE	Returns release number
INFO_BUILD	Returns build date. The date is returned in integer in format DDMMYYYY, i.e. 29082004

INFO_STAGE	The library stage: STAGE_PREVIEW, STAGE_ALPHA, STAGE_BETA, STAGE_GAMMA, STAGE_RC, STAGE_FINAL
INFO_ISDEBUG	Returns 1 if debug code is enabled

Return value

The result returned is different depending on param or -1 if an error occurred.

Comments

1.16. DPSDK_DonePrinterEx

Frees allocated memory chunks and closes the connection.

Syntax

```
void DPSDK_DonePrinterEx(UInt16 refNum, UInt32 flags)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

flags

any combination of the following:

Value	Meaning
DONE_OFF	before closing the connection turns of the printer too

Return value

Comments

1.17. DPSDK_GetClock

Returns the value of the printer's internal real-time clock

Syntax

```
Boolean DPSDK_GetClock(UInt16 refNum, Int16 *year, Int16 *month, Int16 *day, Int16 *weekDay,
Int16 *hour, Int16 *minute, Int16 *second)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

year

pointer to an integer where the year will be stored

month

pointer to an integer where the month will be stored

day

pointer to an integer where the day will be stored

weekDay

pointer to an integer where the day of the week will be stored

hour

pointer to an integer where the hour will be stored

minute

pointer to an integer where the minute will be stored

second

pointer to an integer where the second will be stored

Return value

True if success, false if there was an error reading the RTC clock (or it was never set with

[DPSDK_SetClock](#))

Comments

1.18. DPSDK_SetClock

Sets the time of the printer's real-time clock

Syntax

Boolean DPSDK_SetClock(UInt16 refNum, Int16 year, Int16 month, Int16 day, Int16 weekDay, Int16 hour, Int16 minute, Int16 second)

Parameters

refNum

library reference number obtained via call to SysLibLoad

year

year number

month

month number

day

day number

weekDay

day of the week

hour

hour number

minute

minute number

second

seconds number

Return value

True if success, false if there was an error setting the RTC

Comments

1.19. DPSDKLib_OpenLibrary

User-level call to open the library. This inline function handles the messy task of finding or loading the library and calling its open function and handles cleanup if the library could not be opened.

Syntax

static Err DPSDKLib_OpenLibrary(UInt16 *refNumP, UInt32 *clientContextP)

Parameters

refNumP

Pointer to UInt16 variable that will hold the new library reference number to be used in later calls

clientContextP

pointer to variable for returning client context. The client context is used to maintain client-specific data for multiple client support. The value returned here will be used as a parameter for other library functions, which require a client context.

Return value

errNone
memErrNotEnoughSpace
sysErrLibNotFound
sysErrNoFreeRAM
sysErrNoFreeLibSlots

Comments

1.20. DPSDKLib_CloseLibrary

User-level call to close the shared library. This handles removal of the library from the system if there are no applications using the library.

Syntax

__inline Err DPSDKLib_CloseLibrary(UInt16 refNum, UInt32 clientContext)

Parameters

refNum

library reference number obtained via call to SysLibLoad

clientContext

client context (as returned by the open call)

Return value

errNone
sysErrParamErr

Comments

2. Graphics

A set of functions for drawing in virtual graphic page and print it

2.1. DPSDK_PrnStartPage

Opens a virtual page with the desired size.

Syntax

Boolean DPSDK_PrnStartPage(UInt16 refNum, UInt16 HSize, UInt16 VSize)

Parameters

refNum

library reference number obtained via call to SysLibLoad

HSize

Horizontal size in pixels (no more than 384)

VSize

Vertical size in pixels

Return value

True if success, otherwise false

Comments

Here's how all this is supposed to work:

1. Call [DPSDK_PrnStartPage](#) - it'll allocate resources and all the stuff needed.
2. Draw your imaginations on that page
3. Call [DPSDK_PrnEndPage](#) to print it and free some resources
4. Repeat 1-3 as many times you need
5. Call [DPSDK_PrnClose](#) to destroy all Prn stuff.

2.2. DPSDK_PrnClose

Frees all memory allocated and closes the last opened page.

Syntax

Boolean DPSDK_PrnClose(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, otherwise false

Comments

Here's how all this is supposed to work:

1. Call [DPSDK_PrnStartPage](#) - it'll allocate resources and all the stuff needed.
2. Draw your imaginations on that page
3. Call [DPSDK_PrnEndPage](#) to print it and free some resources
4. Repeat 1-3 as many times you need
5. Call [DPSDK_PrnClose](#) to destroy all Prn stuff.

2.3. DPSDK_PrnDrawBitmap

Draws a bitmap at the desired coordinates in the page.

Syntax

Boolean DPSDK_PrnDrawBitmap(UInt16 refNum, BitmapPtr bitmapP, UInt16 x, UInt16 y)

Parameters

refNum

library reference number obtained via call to SysLibLoad

bitmapP

Pointer to a BitmapType

x

Horizontal coordinate in pixels

y

Vertical coordinate in pixels

Return value

Comments

2.4. DPSDK_PrnDrawChars

Prints some text in the specified mode

Syntax

Boolean DPSDK_PrnDrawChars(UInt16 refNum, char *chars, UInt16 len, UInt16 x, UInt16 y)

Parameters

refNum

library reference number obtained via call to SysLibLoad

chars

The text to be printed

len

The length of the text

x

Horizontal coordinate in pixels

y

Vertical coordinate in pixels

Return value

True if success, otherwise false

Comments

2.5. DPSDK_PrnDrawTruncChars

Prints text truncated to MaxWidth

Syntax

Boolean DPSDK_PrnDrawTruncChars(UInt16 refNum, char *chars, UInt16 len, UInt16 x, UInt16 y, UInt16 MaxWidth)

Parameters

refNum

library reference number obtained via call to SysLibLoad

chars

The text to be printed

len

The length of the text

x

Horizontal coordinate in pixels

y

Vertical coordinate in pixels

MaxWidth

The maximum size in pixels

Return value

True if success, otherwise false

Comments

2.6. DPSDK_PrnDrawGrayLine

Draws a gray line

Syntax

Boolean DPSDK_PrnDrawGrayLine(UInt16 refNum, UInt16 x1, UInt16 y1, UInt16 x2, UInt16 y2)

Parameters

refNum

library reference number obtained via call to SysLibLoad

x1

Start X coordinate in pixels

y1

Start Y coordinate in pixels

x2

End X coordinate in pixels

y2

End Y coordinate in pixels

Return value

True if success, otherwise false

Comments

2.7. DPSDK_PrnDrawGrayRectangle

Draws a gray rectangle

Syntax

Boolean DPSDK_PrnDrawGrayRectangle(UInt16 refNum, RectanglePtr r, UInt16 cornerDiam)

Parameters

refNum

library reference number obtained via call to SysLibLoad

r

Pointer to RectangleType structure

cornerDiam

The corner diameter to round the rectangle

Return value

True if success, otherwise false

Comments

2.8. DPSDK_PrnFillRectangle

Fills a rectangle with the specified pattern

Syntax

Boolean DPSDK_PrnFillRectangle(UInt16 refNum, RectanglePtr r, UInt16 cornerDiam)

Parameters

refNum

library reference number obtained via call to SysLibLoad

r

Pointer to RectangleType structure

cornerDiam

The corner diameter to round the rectangle

Return value

True if success, otherwise false

Comments

2.9. DPSDK_PrnDrawGrayRectangleFrame

Draws a gray rectangle frame

Syntax

Boolean DPSDK_PrnDrawGrayRectangleFrame(UInt16 refNum, RectanglePtr r)

Parameters

refNum

library reference number obtained via call to SysLibLoad

r

Pointer to RectangleType structure

Return value

True if success, otherwise false

Comments

2.10. DPSDK_PrnDrawLine

Draws a black line

Syntax

Boolean DPSDK_PrnDrawLine(UInt16 refNum, UInt16 x1, UInt16 y1, UInt16 x2, UInt16 y2)

Parameters

refNum

library reference number obtained via call to SysLibLoad

x1

Start X coordinate in pixels

y1

Start Y coordinate in pixels

x2

End X coordinate in pixels

y2

End Y coordinate in pixels

Return value

True if success, otherwise false

Comments

2.11. DPSDK_PrnDrawRectangle

Draws a black rectangle

Syntax

Boolean DPSDK_PrnDrawRectangle(UInt16 refNum, RectanglePtr r, UInt16 cornerDiam)

Parameters

refNum

library reference number obtained via call to SysLibLoad

r

Pointer to RectangleType structure

cornerDiam

The corner diameter to round the rectangle

Return value

True if success, otherwise false

Comments

2.12. DPSDK_PrnDrawRectangleFrame

Draws a black rectangle frame

Syntax

Boolean DPSDK_PrnDrawRectangleFrame(UInt16 refNum, RectanglePtr r)

Parameters

refNum

library reference number obtained via call to SysLibLoad

r

Pointer to RectangleType structure

Return value

True if success, otherwise false

Comments

2.13. DPSDK_PrnDrawText

Draws a text aligned in a rectangle

Syntax

Boolean DPSDK_PrnDrawText(UInt16 refNum, char *text, UInt16 len, RectanglePtr r, UInt8 flags)

Parameters

refNum

library reference number obtained via call to SysLibLoad

text

The text to be printed

len

The length of the text

r

Pointer to RectangleType structure

flags

Text position flags as follows:

Value	Meaning
prnDrawTextLeft	Align in Left
prnDrawTextCenter	Align in Center
prnDrawTextRight	Align in Right
prnDrawTextBottom	Align vertical in Bottom
prnDrawTextVCenter	Align vertical in Center
prnDrawTextTop	Align vertical in Top
prnDrawTextSingleLine	Not used so far...
prnDrawTextMultiLine	Not used so far...
prnDrawTextWordBreak	Not used so far...
prnDrawTextClip	Not used so far...
prnDrawTextNoClip	Not used so far...

Return value

True if success, otherwise false

Comments

2.14. DPSDK_PrnGetFont

Returns the currently selected font

Syntax

UInt8 DPSDK_PrnGetFont(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

The currently selected font The default PalmOS fonts are listed below:

Value	Meaning
stdFont	Standard Font
boldFont	Bold Font
largeFont	Large Font
largeBoldFont	Large Bold Font
symbolFont	Various UI Images
symbol11Font	Larger Various UI Images
symbol7Font	Smaller Various UI Images
ledFont	Calculator-like font

Comments

2.15. DPSDK_PrnGetPageExtent

Return the page dimensions

Syntax

Boolean DPSDK_PrnGetPageExtent(UInt16 refNum, UInt16 *extentX, UInt16 *extentY)

Parameters

refNum

library reference number obtained via call to SysLibLoad

extentX

Pointer for returning Page's horizontal size in pixels

extentY

Pointer for returning Page's vertical size in pixels

Return value

True if success, otherwise false

Comments

2.16. DPSDK_PrnGetPattern

Gets the currently specified pattern

Syntax

Boolean DPSDK_PrnGetPattern(UInt16 refNum, CustomPatternType pattern)

Parameters

refNum

library reference number obtained via call to SysLibLoad

pattern

See the PalmOS documentation about CustomPattern

Return value

True if success, otherwise false

Comments

2.17. DPSDK_PrnGetUnderlineMode

Gets the currently specified underline mode

Syntax

Boolean DPSDK_PrnGetUnderlineMode(UInt16 refNum, UnderlineModeType *modeP)

Parameters

refNum

library reference number obtained via call to SysLibLoad

modeP

See the PalmOS documentation about UnderlineModeType

Return value

True if success, otherwise false

Comments

2.18. DPSDK_PrnEndPage

Prints the currently opened page and opens a blank one.

Syntax

Boolean DPSDK_PrnEndPage(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, otherwise false

Comments

Here's how all this is supposed to work:

1. Call [DPSDK_PrnStartPage](#) - it'll allocate resources and all the stuff needed.
2. Draw your imaginations on that page
3. Call [DPSDK_PrnEndPage](#) to print it and free some resources
4. Repeat 1-3 as many times you need
5. Call [DPSDK_PrnClose](#) to destroy all Prn stuff.

2.19. DPSDK_PrnSetFont

Sets the font used for text draw functions

Syntax

Boolean DPSDK_PrnSetFont(UInt16 refNum, UInt8 font)

Parameters

refNum

library reference number obtained via call to SysLibLoad

font

Font type, one of the following:

Value	Meaning
stdFont	Standard Font
boldFont	Bold Font
largeFont	Large Font
largeBoldFont	Large Bold Font
symbolFont	Various UI Images
symbol11Font	Larger Various UI Images
symbol7Font	Smaller Various UI Images
ledFont	Calculator-like font

Return value

True if success, otherwise false

Comments

2.20. DPSDK_PrnSetPattern

Sets a custom pattern

Syntax

Boolean DPSDK_PrnSetPattern(UInt16 refNum, CustomPatternType pattern)

Parameters

refNum

library reference number obtained via call to SysLibLoad

pattern

See the PalmOS documentation about CustomPattern

Return value

True if success, otherwise false

Comments

2.21. DPSDK_PrnSetUnderlineMode

Sets the underline mode

Syntax

Boolean DPSDK_PrnSetUnderlineMode(UInt16 refNum, UnderlineModeType mode)

Parameters

refNum

library reference number obtained via call to SysLibLoad

mode

See the PalmOS documentation about UnderlineModeType

Return value

True if success, otherwise false

Comments

2.22. DPSDK_PrnGetAverageCharWidth

Gets average character width according to currently selected font

Syntax

```
UInt16 DPSDK_PrnGetAverageCharWidth(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Returns the average character width of the selected font

Comments

2.23. DPSDK_PrnGetCharHeight

Gets character height according to currently selected font

Syntax

```
UInt16 DPSDK_PrnGetCharHeight(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Returns the character height of the selected font

Comments

2.24. DPSDK_PrnGetCharWidth

Gets single character width according to currently selected font

Syntax

```
UInt16 DPSDK_PrnGetCharWidth(UInt16 refNum, Char ch)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

ch

The character which width will be returned

Return value

Character width

Comments

2.25. DPSDK_PrnGetCharsInWidth

Finds the length in bytes of the characters from a specified string that fit within a passed width.

Syntax

```
Boolean DPSDK_PrnGetCharsInWidth(UInt16 refNum, char *string, UInt16 *stringWidthP, UInt16 *stringLengthP, Boolean *fitWithinWidth)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

string

A pointer to the character string

stringWidthP

The maximum width to allow (in pixels). Upon return, contains the actual width allowed. Note that this value does not include any trailing spaces or tabs, which are stripped by this function.

stringLengthP

The maximum length of text to allow, in bytes (assumes current font). Upon return, contains the number of bytes of text that can appear within the width. Note that this value does not include any trailing space or tabs, which are stripped by this function.

fitWithinWidth

Upon return, false if the string is considered truncated, true if it isn't.

Return value

True if success, otherwise false

Comments

2.26. DPSDK_PrnGetCharsWidth

Gets characters width according to currently selected font

Syntax

UInt16 DPSDK_PrnGetCharsWidth(UInt16 refNum, char *chars, UInt16 len)

Parameters

refNum

library reference number obtained via call to SysLibLoad

chars

Pointer to string

len

Number of characters to process

Return value

Characters width

Comments

2.27. DPSDK_PrnSetTextMode

Sets the way the text will cooperate with the rest of the image

Syntax

Boolean DPSDK_PrnSetTextMode(UInt16 refNum, UInt16 TextMode)

Parameters

refNum

library reference number obtained via call to SysLibLoad

TextMode

Text mode, one of the following:

Value	Meaning
winPaint	Write color-matched source pixels to the destination.
winErase	Write backColor if the source pixel is transparent.
winMask	Write backColor if the source pixel is not transparent.
winInvert	Bitwise XOR the color-matched source pixel onto the destination.
winOverlay	Write color-matched source pixel to the destination if the source pixel is not transparent. Transparent pixels are skipped. For a 1-bit display, the "off" bits are considered to be the transparent color. Note that this definition of winOverlay is new for Palm OS 5: in Palm OS 4.x, the destination is set (only) where the source pixels are "on."
winPaintInverse	Invert the source pixel color and then proceed as with winPaint.
winSwap	Swap the backColor and foreColor destination colors if the source is a pattern (the type of pattern is disregarded). If the source is a bitmap, then the bitmap is transferred using winPaint mode instead

Return value

True if success, otherwise false

Comments

2.28. DPSDK_PrnGetTextMode

Returns the currently selected text mode

Syntax

UInt16 DPSDK_PrnGetTextMode(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Text mode, one of the following:

Value	Meaning
winPaint	Write color-matched source pixels to the destination.
winErase	Write backColor if the source pixel is transparent.
winMask	Write backColor if the source pixel is not transparent.
winInvert	Bitwise XOR the color-matched source pixel onto the destination.
winOverlay	Write color-matched source pixel to the destination if the source pixel is not transparent. Transparent pixels are skipped. For a 1-bit display, the "off" bits are considered to be the transparent color. Note that this definition of winOverlay is new for Palm OS 5: in Palm OS 4.x, the destination is set (only) where the source pixels are "on."
winPaintInverse	Invert the source pixel color and then proceed as with winPaint.
winSwap	Swap the backColor and foreColor destination colors if the source is a pattern (the type of pattern is disregarded). If the source is a bitmap, then the bitmap is transferred using winPaint mode instead

Comments

3. Printing

A set of functions for direct printing of graphics or text, controlling print settings, etc

3.1. DPSDK_LineFeed

Feeds the paper one line.

Syntax

```
void DPSDK_LineFeed(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Comments

3.2. DPSDK_SelfTest

Prints the Printer Self Test

Syntax

```
Boolean DPSDK_SelfTest(UInt16 refNum, Boolean longest)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

longtest

if True, prints the Long Printer Test, otherwise prints the Short Printer Test

Return value

True if success, otherwise false

Comments

3.3. DPSDK_PrintText

Prints text in the current font and style. If the text is longer than the current characters per line, the characters that exceed the limit are printed on the next line.

Syntax

```
Boolean DPSDK_PrintText(UInt16 refNum, char *text)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

text

a string to be printed

Return value

True if success, otherwise false

Comments

3.4. DPSDK_PrintTextWrapped

Prints text word wrapped in the current font and style

Syntax

Boolean DPSDK_PrintTextWrapped(UInt16 refNum, char *text)

Parameters

refNum

library reference number obtained via call to SysLibLoad

text

a string to be printed

Return value

True if success, otherwise false

Comments

3.5. DPSDK_PrintBarcode

Prints Barcode in the current style

Syntax

Boolean DPSDK_PrintBarcode(UInt16 refNum, int type, char *barcode)

Parameters

refNum

library reference number obtained via call to SysLibLoad

type

One of the following values:

Value	Meaning
Bar_UPCA	UPC-A
Bar_UPCE	UPC-E
Bar_EAN13	EAN13 (JAN13)
Bar_EAN8	EAN 8 (JAN8)
Bar_CODE39	CODE 39
Bar_ITF	ITF
Bar_CODABAR	CODABAR (NW-7)
Bar_CODE93	CODE 93
Bar_CODE128	CODE 128
Bar_PDF417	PDF-417

barcode

null-terminated barcode string

Return value

True if success, otherwise false

Comments

Regardless of the specified feed pitch, this command feeds the paper that is required to print a bar code. If the character code cannot be printed in the respective bar code system, the bar code so far will be printed, processing the subsequent data as normal data. When a bar code whose number of characters to

be printed is fixed has been selected, the number of characters k have to be always made equal to the number of characters to be printed. (The bar code is not printed when not matching.) When the horizontal direction exceeds one line length, the bar code is not printed.

[CODE 128 additional information]

Code 128 covers all ASCII codes from 0 to 127, and it has three tables A, B and C that can be used in the same bar code.

Table A: contains ASCII symbols with codes 0 to 95 and control symbols FNC1, FNC2, FNC3, FNC4, SHIFT, CODEB, CODEC.

Table B: contains symbols with ASCII codes from 32 to 127 and control characters FNC1, FNC2, FNC3, FNC4, SHIFT, CODEA, CODEC.

Table C: It is used for coding bar code areas that consist only of numerals. Each symbol gives two digits that are coded with ASCII codes from 0 to 99. Control characters FNC1, CODEA, CODEB are also available.

The bar code must start with one of the symbols CODEA, CODEB or CODEC, that determines which table will be used. If it is necessary the current table could be changed later by inserting one of these symbols in to the bar code. The symbol following SHIFT character is considered to be from table B, if table A is current, or from table A, if table B is current. If a symbol that is not valid for the current table is used, the whole bar code is not printed.

Control symbols are assigned two bytes as shown:

Value	Meaning
FNC1	{ 1
FNC2	{ 2
FNC3	{ 3
FNC4	{ 4
CODEA	{ A
CODEB	{ B
CODEC	{ C
SHIFT	{ S
{	{{

3.6. DPSDK_PrintBuffer

Sends a byte sequence directly to the printer.

Syntax

Boolean DPSDK_PrintBuffer(UInt16 refNum, char *buf, int size)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

the buffer that should be sent

size

the size of the buffer

Return value

True if success, otherwise false

Comments

3.7. DPSDK_PrintLogo

Prints stored logo. The logo can be loaded with @DPSDK_UploadLogo or LogoManager program

Syntax

Boolean DPSDK_PrintLogo(UInt16 refNum, UInt8 m)

Parameters

refNum

library reference number obtained via call to SysLibLoad

m

mode, the valid values are:

Value	Meaning
0	Normal mode 203x203 DPI
1	Double wide mode 203x101 DPI
2	Double high mode 101x203 DPI
3	DW / DH mode 101x101 DPI

Return value

Comments

3.8. DPSDK_PrintScreen

Gets the screen content within the rectangle with top-left corner (x,y) and bottom-right corner (x+w,y+h) and scales it in percents before print. The screen coordinates begins from zero

Syntax

Boolean DPSDK_PrintScreen(UInt16 refNum, int X, int Y, int W, int H, int percents)

Parameters

refNum

library reference number obtained via call to SysLibLoad

X

toleft X coordinate of the rectangle within the screen. Pass -1 to make it auto (beginning)

Y

toleft Y coordinate of the rectangle within the screen. Pass -1 to make it auto (beginning)

W

width of the rectangle within the screen. Pass -1 to make it auto (full screen width)

H

height of the rectangle within the screen. Pass -1 to make it auto (full screen height)

percents

scale of the screen image in %. If you specify -1, it will be scaled to fit the printer paper width

Return value

Comments

3.9. DPSDK_SetBarcodeScale

Sets the scale factor for barcodes.

Syntax

```
void DPSDK_SetBarcodeScale(UINT16 refNum, int scale)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

scale

valid values are 2, 3, 4

Return value

Comments

3.10. DPSDK_SetBarcodeHeight

Sets the barcode's height

Syntax

```
void DPSDK_SetBarcodeHeight(UINT16 refNum, int height)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

height

the valid value is between 1 and 255

Return value

Comments

3.11. DPSDK_SetBarcodeText

Sets the presence and position of barcode's text. This text is the string passed to [DPSDK_PrintBarcode](#).

Syntax

```
void DPSDK_SetBarcodeText(UINT16 refNum, int text)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

text

one of the following:

Value	Meaning
BARTEXT_NONE	No text
BARTEXT_ABOVE	Text above the barcode
BARTEXT_BELOW	Text below the barcode
BARTEXT_BOTH	Text bot above and below

Return value

Comments

3.12. DPSDK_SetIntensity

Sets the darkness of the printout.

Syntax

```
void DPSDK_SetIntensity(UInt16 refNum, unsigned int intensity)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

intensity

one of the following values:

Value	Meaning
Intensity70	Intensity set to 70%
Intensity80	Intensity set to 80%
Intensity90	Intensity set to 90%
Intensity100	Intensity set to 100% (default)
Intensity120	Intensity set to 120%
Intensity150	Intensity set to 150%

Return value

Comments

3.13. DPSDK_PrintBarcodeN

Prints Barcode in the current style

Syntax

```
Boolean DPSDK_PrintBarcodeN(UInt16 refNum, int type, UInt8 *barcode, UInt8 N)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

type

barcode type, see [DPSDK_PrintBarcode](#)

barcode

barcode data

N

number of symbols in the barcode data

Return value

True if success, otherwise false

Comments

See [DPSDK_PrintBarcode](#) function

3.14. DPSDK_SetBarcodeAlign

Sets the horizontal and vertical aligning of the barcode and the text. This text is the string passed to [DPSDK_PrintBarcode](#).

Syntax

```
void DPSDK_SetBarcodeAlign(UINT16 refNum, int align)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

align

one of the following:

Value	Meaning
BARALIGN_HLEFT	Left end alignment
BARALIGN_HCENTER	Horizontal centered
BARALIGN_HRIGHT	Right end alignment
BARALIGN_VLEFT	Left end aligned vertical bar code
BARALIGN_VCENTER	Centred vertical bar code
BARALIGN_VRIGHT	Right end aligned vertical bar code

Return value

Comments

3.15. DPSDK_SetImageSettings

Sets the brightness/contrast values used when you print bitmap/screen with 265 colors or more

Syntax

```
void DPSDK_SetImageSettings(UINT16 refNum, Boolean BrightEnabled, int Brightness, Boolean ContrastEnabled, int Contrast)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

BrightEnabled

Enable Brightness processing

Brightness

Brightness value [0..200], default is 100

ContrastEnabled

Enable contrast processing

Contrast

Contrast value [0..200], default is 100

Return value

True if success, otherwise false

Comments

This settings will be used in [DPSDK_PrintScreen](#), [@DPSDK_ScreenCapture](#), [DPSDK_PrintBitmapPtr](#) and [DPSDK_PrintBitmapRes](#)

3.16. DPSDK_ScreenInfo

Returns information about current device's screen.

Syntax

Boolean DPSDK_ScreenInfo(UInt16 refNum, UInt16 *width, UInt16 *height, UInt16 *colordepth)

Parameters

refNum

library reference number obtained via call to SysLibLoad

width

where the screen width is returned

height

where the screen height is returned

colordepth

where the screen color depth is returned. I.e. 1 for B/W, 2 for 4 grays, 4 for 16 grays, 8 for 256 colors, 16 for 65536 colors....

Return value

True if success, otherwise false

Comments

This function returns the "real" coordinates, which you can use to pass to [DPSDK_PrintScreen](#), etc. By "real" I mean that it'll return 320x320 on device with high-res, will calculate the sizes correct if you have opened the input area on Tungsten|T3, will catch landscape mode, etc.

3.17. DPSDK_PrintBitmapRes

Prints bitmap from resource ID that goes with the application PRC. Bitmap resource must be up to 8 bit colors.

Syntax

Boolean DPSDK_PrintBitmapRes(UInt16 refNum, DmResID ResID, int X, int Y, int W, int H, int percents)

Parameters

refNum

library reference number obtained via call to SysLibLoad

ResID

Resource Identifier from already opened Database. Use DmGetResource to get it

X

toleft X coordinate of the rectangle within bitmap. Pass -1 to make it auto (beginning)

Y

toleft Y coordinate of the rectangle within bitmap. Pass -1 to make it auto (beginning)

W

width of the rectangle within bitmap. Pass -1 to make it auto (full bitmap width)

H

height of the rectangle within bitmap. Pass -1 to make it auto (full bitmap height)

percents

scale of the bitmap in %. If you specify -1, bitmap will be scaled to fit the printer paper width

Return value

True if success, otherwise false

Comments**3.18. DPSDK_PrintBitmapPtr**

Prints bitmap from a given pointer. Bitmap must be up to 8 bit colors.

Syntax

Boolean DPSDK_PrintBitmapPtr(UInt16 refNum, BitmapPtr Bmp, int X, int Y, int W, int H, int percents)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Bmp

pointer to BitmapType structure

X

topleft X coordinate of the rectangle within bitmap. Pass -1 to make it auto (beginning)

Y

topleft Y coordinate of the rectangle within bitmap. Pass -1 to make it auto (beginning)

W

width of the rectangle within bitmap. Pass -1 to make it auto (full bitmap width)

H

height of the rectangle within bitmap. Pass -1 to make it auto (full bitmap height)

percents

scale of the bitmap in %. If you specify -1, bitmap will be scaled to fit the printer paper width

Return value

True if success, otherwise false

Comments**3.19. DPSDK_LineFeedX**

Move the paper X lines (pixels). Each pixel is 1/203 of the inch

Syntax

void DPSDK_LineFeedX(UInt16 refNum, UInt16 lines)

Parameters

refNum

library reference number obtained via call to SysLibLoad

lines

the number of lines to move the paper

Return value**Comments****3.20. DPSDK_SetLeftMargin**

Sets the left margin (beginning of the text, graphics, everything) in pixels (1/203 of the inch)

Syntax

BOOL DPSDK_SetLeftMargin(UInt16 refNum, UInt16 margin)

Parameters

refNum

library reference number obtained via call to SysLibLoad

margin

the value in pixels (1/203 of the inch)

Return value

Comments

3.21. DPSDK_ResetDefaults

Resets various printer settings to their default values

Syntax

BOOL DPSDK_ResetDefaults(UInt16 refNum, UInt16 flags)

Parameters

refNum

library reference number obtained via call to SysLibLoad

flags

any combination of the following

Reset_PrintSettings

Resets printer printing settings - margin settings, printing intensity level...

Reset_FontSettings

Sets all font settings to their defaults - i.e. bold, underline..

Reset_BarcodeSettings

Sets all barcode settings to their defaults - i.e. height, text position, scale...

Return value

True if success, otherwise false

Comments

3.22. DPSDK_SetLineSpace

Sets the vertical line spacing. See comments

Syntax

BOOL DPSDK_SetLineSpace(UInt16 refNum, UInt16 linespace)

Parameters

refNum

library reference number obtained via call to SysLibLoad

linespace

the line spacing in pixels (1/203 inch)

Return value

Comments

Every line of text has some height, for example if you print with big font, the height of the line will be 24 + some spacing so it does not overlap with the next one. This function sets the total spacing - the font height + additional. So in that case, setting it to 30 for example will leave some space between the lines. If you enter too little value the printer will correct it so no lines overlapping - for example you can set it to 1 and the lines will still have a little space between them.

3.23. DPSDK_PrintTextEx

This is the ultimate all-in-one printing function that replaces nearly all style & text printing functions and provides a lot more functionality

Syntax

BOOL DPSDK_PrintTextEx(UINT16 refNum, char *text, UINT32 flags)

Parameters

refNum

library reference number obtained via call to SysLibLoad

text

text - the text to be printed. This can be either "plain" text, or text with formatting inside. Formatting sequences can be used in any combination to achieve the effect needed. Please check comments section about text formatting

flags

any combination of the following:

Value	Meaning
Text_Italic	Turns on italic font
Text_Bold	Turns on bold font
Text_Underlined	Turns on underlined font
Text_DoubleWidth	Turns on doublewidth font
Text_DoubleHeight	Turns on doubleheight font
Text_Rotated	Turns on rotated (90) font
Text_Reversed	Turns on reversed (180) font
Text_AlignLeft	Select left alignment
Text_AlignCenter	Select center alignment
Text_AlignRight	Select right alignment
Text_BigFont	Sets big font
Text_SmallFont	Sets small font
Text_DontLF	Don't pass linefeed at the end of the text
Text_WordWrap	Turns on text wordwrapping
Text_UseOldSettings	Ignore these flags and use the last text/font settings
Text_SaveSettings	After function compete saves these settings for later use

Return value

True if success, otherwise false

Comments

Besides the flags, you can insert special "sequences" inside the text to be printed that control the fonts, styles, text format, etc. Below is a list of acceptable commands:

Value	Meaning
-------	---------

[==]	Reset all formatting to defaults - 100% intensity, normal font, left aligning
[=B]	Sets big font
[=S]	Sets small font
[=L]	Select left alignment
[=C]	Select center alignment
[=R]	Select right alignment
[=Ix]	Change intensity (0-70%, 1-80%, 2-90%, 3-100%, 4-120%, 5-150%)
[+B]/[-B]	B] - Turns on/off bold font
[+I]/[-I]	I] - Turns on/off italic font
[+U]/[-U]	U] - Turns on/off underlined font
[+DW]/[-DW]	DW] - Turns on/off doublewidth font
[+DH]/[-DH]	DH] - Turns on/off doubleheight font
[+V]/[-V]	V] - Turns on/off reversed (180) font
[+R]/[-R]	R] - Turns on/off rotated (90) font
[+W]/[-W]	W] - Turns on/off text wordwrapping

A sample demonstration:

```
"[=C][+B][+DW][+DH]YO!\r\n[-DW][-DH]This Demonstrates the use of some of the capabilities of
PrintTextEx. Nice function.\r\n[=L]You wanna left?\r\n[=R]...or right?\r\n[=C]well then
center?\r\n[+B][+I]Or italic? [+DW][+DH]and BIG?[=S]\r\nnono - small?\r\n:)\r\n\r\n"
```

3.24. DPSDK_SetFontPack

Allows you to select between the printer's fixed internal fonts (A&B) and downloadable ones

Syntax

Boolean DPSDK_SetFontPack(UInt16 refNum, UInt16 pack)

Parameters

refNum

library reference number obtained via call to SysLibLoad

pack

the fonts to use, one of the following:

Value	Meaning
0	Use printer's internal fonts
1	Use custom fonts (if loaded)

Return value

True if success, otherwise false

Comments

3.25. DPSDK_LoadCustomFont

Loads custom font pack to the printer and activates it

Syntax

Boolean DPSDK_LoadCustomFont(UInt16 refNum, UInt16 cardId, char *filename)

Parameters

refNum

library reference number obtained via call to SysLibLoad

cardId

the card id where the font file resides. Usually 0

filename

the name of the file containing the font pack. Those files are being created/modified with FontManager program

Return value

Comments

3.26. DPSDK_PaperFeed

Feeds the paper so it can be easily tear off without cutting important info or wasting paper.

Syntax

```
void DPSDK_PaperFeed(UInt16 refNum)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Comments

3.27. DPSDK_PTSendData

Sends raw data directly to the printer. This function handles the protocol mode.

Syntax

```
Boolean DPSDK_PTSendData(UInt16 refNum, void *buf, UInt32 length, UInt32 timeout)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

address of the buffer with the data

length

length of the data

timeout

timeout in milliseconds

Return value

True if all the data have been successfully sent

Comments

3.28. DPSDK_PTReceiveData

Receives raw data directly from the printer. This function handles the protocol mode.

Syntax

Int32 DPSDK_PTReceiveData(UInt16 refNum, void *buf, UInt32 length, UInt32 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

address of the buffer to store the received data

length

maximum length of the received data

timeout

timeout in milliseconds

Return value

Number of bytes received or -1 if error occurred

Comments**3.29. DPSDK_PTReceiveDataStop**

Receives raw data up to stop byte directly from the printer. Useful for reading null-terminating string information for example. This function handles the protocol mode.

Syntax

Int32 DPSDK_PTReceiveDataStop(UInt16 refNum, void *buf, UInt32 length, UInt8 stop, UInt32 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

address of the buffer to store the received data

length

maximum length of the received data

stop

byte value which stops the reading

timeout

timeout in milliseconds

Return value

Number of bytes being read or -1 if error occurred

Comments

4. Barcode reader

A set of functions for managing the optional barcode reader that can be attached to PP-55

4.1. DPSDK_PPBarcodeRead

Scan a barcode using the built-in barcode reader into PP-55 (optional)

Syntax

```
UInt16 DPSDK_PPBarcodeRead(UInt16 refNum, UInt16 *type, char *barcode)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

type

a pointer where to store the read barcode type, one of the following:

Value	Meaning
0x16	Bookland
0x02	Codabar
0x0C	Code11
0x20	Code32
0x03	Code128
0x01	Code39
0x13	Code39_FullASCII
0x07	Code93
0x1D	Composite
0x17	Coupon
0x04	D25
0x1B	DataMatrix
0x0F	EAN_128
0x0B	EAN_13
0x4B	EAN_13_plus_2
0x8B	EAN_13_plus_5
0x0A	EAN_8
0x4A	EAN_8_plus_2
0x8A	EAN_8_plus_5
0x05	IATA
0x19	ISBT_128
0x21	ISBT_128_concateneted
0x06	ITF
0x28	MacroPDF
0x0E	MSI
0x11	PDF_417
0x26	Postbar_Canada
0x1E	Postnet_US
0x23	Postal_Australia
0x22	Postal_Japan
0x27	Postal_UK

0x1C	QR_code
0x31	RSS_limited
0x30	RSS_14
0x32	RSS_Expanded
0x24	Signature
0x15	TriopticCode39
0x08	UPCA
0x48	UPCA_plus_2
0x88	UPCA_plus_5
0x09	UPCE
0x49	UPCE_plus_2
0x89	UPCE_plus_5
0x10	UPCE1
0x50	UPCE1_plus_2
0x90	UPCE1_plus_5

barcode

string that holds the read barcode. Be sure it is long enough

Return value

Some of the following:

Value	Meaning
PPBarErrNone	No error
PPBarErrNoBar	No barcode to read
PPBarErrFailed	Operation failed. This means reader error.

Comments

4.2. DPSDK_BarcodeType2Text

Helper function that gives the string representation of any barcode type, i.e. for 0x01 will return "Code 39"

Syntax

```
void DPSDK_BarcodeType2Text(UInt16 refNum, UInt8 type, char *text)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

type

barcode type number

text

a pointer to a character buffer of at least 22 symbols

Return value

Comments

5. Magnetic Stripe reader

A set of functions for managing the optional magnetic stripe reader that can be attached to PP-50, PP-55, DPP-250, DPP-350, CMP-10

5.1. DPSDK_ReadPaymentCard

Reads magnetic card and returns information retrieved from it. If there is a field that you do not want to take from the card, send NULL. The Procedure will wait up to timeout seconds.

Syntax

Boolean DPSDK_ReadPaymentCard(UInt16 refNum, char *Name, UInt16 sizeN, char *Number, UInt16 sizeD, UInt16 *year, UInt16 *month, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Name

buffer to store cardholder name

sizeN

size of buffer Name

Number

buffer to store Card Number

sizeD

size of buffer Number

year

pointer to UInt16 for Card expiration year

month

pointer to UInt16 for Card expiration month

timeout

seconds to wait for the card data

Return value

True if success, otherwise false

Comments

5.2. DPSDK_ReadCardTracks

Reads magnetic cards returns information retrieved from it. If there is a field that you do not want to take from the card, send NULL. The Procedure will wait up to timeout seconds

Syntax

Boolean DPSDK_ReadCardTracks(UInt16 refNum, char *Track1, char *Track2, char *Track3, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Track1

buffer to store track1 info, null-terminated string

Track2

buffer to store track2 info, null-terminated string

Track3

buffer to store track3 info, null-terminated string

timeout

timeout in seconds to read the card

Return value

True if success, otherwise false

Comments

5.3. DPSDK_ReadCardTracksStart

Initiates magnetic card reading process and returns immediately. The reading continues in the background and every second ReadCardTick event is posted in the current application's queue, on read failing or timeout elapsed the ReadCardTracksFailed event is posted, on success a ReadCardTracksSuccess event is posted and the information is ready to use. If you set timeout value to 0 no background processing is initiated, in this case you must call ReadCardResult multiply times until you get the result or use it to cancel the reading! If there is a field that you do not want to take from a card, send NULL. The Procedure waits up to timeout seconds for response from the printer.

Syntax

Boolean DPSDK_ReadCardTracksStart(UInt16 refNum, char *Track1, char *Track2, char *Track3, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Track1

buffer to store track1 info, null-terminated string

Track2

buffer to store track2 info, null-terminated string

Track3

buffer to store track3 info, null-terminated string

timeout

timeout in seconds to read the card

Return value

Comments

Since the function works in the background the variables you send MUST be global or they must live long enough until the reading is stopped!

Note: Do NOT unload the library while reading in progress! This will result in Fatal Exception!

5.4. DPSDK_ReadCardResult

Gets the result from the reading initiated by **DPSDK_ReadCardTracksStart** or **DPSDK_ReadPaymentCardStart**. Returns true if the reading is successful and the information is stored in the values you specified before. If you set cancel to TRUE, the reading process will end immediately and the reader will be turned off.

Syntax

Boolean DPSDK_ReadCardResult(UInt16 refNum, Boolean Stop)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Stop

if TRUE stops reading

Return value

True reading is successful and the information is stored in the values you specified before.

Comments

Use this function only when you set the timeout value of the corresponding Start functions to 0. Otherwise use the event management.

5.5. DPSDK_ReadPaymentCardStart

Initiates magnetic card reading process and returns immediately. The reading continues in the background and every second ReadCardTick event is posted in the current application's queue. Upon read failing or timeout elapsed the ReadPaymentCardFailed event is posted, upon success - a ReadPaymentCardSuccess event is posted and the information is ready to use. If you set timeout value to 0 no background processing is initiated, in this case you must call ReadCardResult multiple times until you get the result or use it to cancel the reading! If there is a field that you do not want to take from a card, send NULL. The Procedure waits up to timeout seconds for response from the printer.

Syntax

Boolean DPSDK_ReadPaymentCardStart(UInt16 refNum, char *Name, UInt16 sizeN, char *Number, UInt16 sizeD, UInt16 *year, UInt16 *month, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Name

buffer to store cardholder name

sizeN

size of buffer Name

Number

buffer to store Card Number

sizeD

size of buffer Number

year

pointer to UInt16 for Card expiration year

month

pointer to UInt16 for Card expiration month

timeout

seconds to wait for the card data

Return value

True if success, otherwise false

Comments

Since the function works in the background the variables you send **MUST** be global or they must live long enough until the reading is stopped! Do NOT unload the library while reading in progress! This will result in Fatal Exception!

5.6. DPSDK_ProcessPaymentCardData

Given the magnetic card raw data, this function tries to extract the valuable information from it

Syntax

```
BOOL DPSDK_ProcessPaymentCardData(UInt16 refNum, char *data, char *Name, char *Number,  
    UInt16 *Year, UInt16 *Month)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

data

raw magnetic card data, can be get from [DPSDK_ReadCardTracks](#) for example combining the tracks

Name

where to return the extracted card holder name

Number

where to return the extracted card number

Year

where the extracted expiration year will be written

Month

where the extracted expiration month will be written

Return value

True if success, otherwise false

Comments

You can pass 0(NULL, nil, null or whatever) to any of the parameters except data if you do not want to retrieve this element.

5.7. DPSDK_ProcessPaymentCardTracks

Given the magnetic card raw data, this function tries to extract the valuable information from it

Syntax

```
BOOL DPSDK_ProcessPaymentCardTracks(UInt16 refNum, char *Track1, char *Track2, char *Track3,  
    char *Name, char *Number, UInt16 *Year, UInt16 *Month)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

Track1

raw magnetic card track 1 data, can be get from [DPSDK_ReadCardTracks](#)

Track2

raw magnetic card track 2 data, can be get from [DPSDK_ReadCardTracks](#)

Track3

raw magnetic card track 3 data, can be get from [DPSDK_ReadCardTracks](#)

Name

where to return the extracted card holder name

Number

where to return the extracted card number

Year

where the extracted expiration year will be written

Month

where the extracted expiration month will be written

Return value

True if success, otherwise false

Comments

You can pass 0(NULL, nil, null or whatever) to any of the parameters except data if you do not want to retrieve this element.

6. Mifare reader

A set of functions for managing the optional mifare reader that can be attached to PP-55, DPP-250, DPP-350

6.1. DPSDK_MIFARE_Ident

Returns mifare reader identification string

Syntax

BOOL DPSDK_MIFARE_Ident(UInt16 refNum, char *ident, Int32 length)

Parameters

refNum

library reference number obtained via call to SysLibLoad

ident

pointer to a buffer to receive mifare identification

length

maximum size of the ident buffer

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

This function can be used to query about mifare reader being present. The returned identification string can not be more than 64 characters.

6.2. DPSDK_MIFARE_PowerOn

Powers on mifare reader module. Call this function before any other mifare card functions

Syntax

BOOL DPSDK_MIFARE_PowerOn(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.3. DPSDK_MIFARE_PowerOff

Powers off mifare reader module. Call this function after you are done with the mifare module

Syntax

BOOL DPSDK_MIFARE_PowerOff(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, false if there was a problem accessing mifare reader module

Comments**6.4. DPSDK_MIFARE_Config**

Configure mifare reader module.

Syntax

```
BOOL DPSDK_MIFARE_Config(UInt16 refNum, UInt8 *status)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments**6.5. DPSDK_MIFARE_Request**

Performs a search for available cards nearby

Syntax

```
BOOL DPSDK_MIFARE_Request(UInt16 refNum, UInt8 *status, UInt8 *rq1, UInt8 *rq2)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

status

one of the MF_STAT constants

rq1

RQ1 and RQ2 return the card type

rq2

RQ1 and RQ2 return the card type

Return value

True if success, false if there was a problem accessing mifare reader module

Comments**6.6. DPSDK_MIFARE_Anticollision**

Gets mifare card serial number

Syntax

```
BOOL DPSDK_MIFARE_Anticollision(UInt16 refNum, UInt8 *status, UInt32 *serial)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

status

one of the MF_STAT constants

serial

card serial number

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.7. DPSDK_MIFARE_SelectCard

Selects the card found with [DPSDK_MIFARE_Anticollision](#)

Syntax

BOOL DPSDK_MIFARE_SelectCard(UInt16 refNum, UInt32 serial, UInt8 *status, UInt8 *sak)

Parameters

refNum

library reference number obtained via call to SysLibLoad

serial

serial number returned by [DPSDK_MIFARE_Anticollision](#)

status

one of the MF_STAT constants

sak

SAK

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.8. DPSDK_MIFARE_AuthByKey

Authenticate some block with direct key.

Syntax

BOOL DPSDK_MIFARE_AuthByKey(UInt16 refNum, UInt32 serial, UInt8 type, UInt8 block, UInt8 key[6], UInt8 *status)

Parameters

refNum

library reference number obtained via call to SysLibLoad

serial

card serial number returned by [DPSDK_MIFARE_Anticollision](#)

type

key type - 'A' or 'B'

block

block number

key

direct key value to authenticate block with. Default key for most cards is FF FF FF FF FF FF

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.9. DPSDK_MIFARE_Read

Reads a single 16 bytes block from the mifare card

Syntax

BOOL DPSDK_MIFARE_Read(UInt16 refNum, UInt8 address, UInt8 *status, UInt8 data[16])

Parameters

refNum

library reference number obtained via call to SysLibLoad

address

block number to read

status

one of the MF_STAT constants

data

a buffer where the data read is stored

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.10. DPSDK_MIFARE_Write

Writes a single 16 bytes block to the mifare card

Syntax

BOOL DPSDK_MIFARE_Write(UInt16 refNum, UInt8 address, UInt8 data[16], UInt8 *status)

Parameters

refNum

library reference number obtained via call to SysLibLoad

address

block number

data

the data to write

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.11. DPSDK_MIFARE_ValueIncrement

Increments the value stored in mifare value card

Syntax

```
BOOL DPSDK_MIFARE_ValueIncrement(UInt16 refNum, UInt8 addr_old, UInt8 addr_new, UInt32 value, UInt8 *status)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

addr_old

source block number

addr_new

destination block number

value

the value to increment with

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.12. DPSDK_MIFARE_ValueDecrement

Decrements the value stored in mifare value card

Syntax

```
BOOL DPSDK_MIFARE_ValueDecrement(UInt16 refNum, UInt8 addr_old, UInt8 addr_new, UInt32 value, UInt8 *status)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

addr_old

source block number

addr_new

destination block number

value

the value to decrement with

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.13. DPSDK_MIFARE_ValueRestore

Restores the value stored in mifare value card

Syntax

```
BOOL DPSDK_MIFARE_ValueRestore(UInt16 refNum, UInt8 addr_old, UInt8 addr_new, UInt32 value,
    UInt8 *status)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

addr_old

source block number

addr_new

destination block number

value

not used

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.14. DPSDK_MIFARE_GetSerial

Return selected's card serial number

Syntax

```
BOOL DPSDK_MIFARE_GetSerial(UInt16 refNum, UInt8 *status, UInt32 *serial)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

status

one of the MF_STAT constants

serial

serial number

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

6.15. DPSDK_MIFARE_WriteValue

Writes value stored in mifare value card

Syntax

```
BOOL DPSDK_MIFARE_WriteValue(UInt16 refNum, UInt8 address, UInt32 value, UInt8 *status)
```

Parameters

refNum

library reference number obtained via call to SysLibLoad

address

the block number to store value to

value

value to store

status

one of the MF_STAT constants

Return value

True if success, false if there was a problem accessing mifare reader module

Comments

7. Smartcard reader

A set of functions for managing the optional smartcard reader that can be attached to PP-55, DPP-250, DPP-350

7.1. DPSDK_SCInit

Initializes the smartcard reader module. Call this function before any other communication with the card

Syntax

Boolean DPSDK_SCInit(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, otherwise false

Comments

7.2. DPSDK_SCDone

Frees system resources allocated by [DPSDK_SCInit](#)

Syntax

Boolean DPSDK_SCDone(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success, otherwise false

Comments

7.3. DPSDK_SCCardPower

Turns on or off smartcard reader power. Use this function to turn the power on just after the [DPSDK_SCInit](#) and power it off before calling [DPSDK_SCDone](#)

Syntax

Boolean DPSDK_SCCardPower(UInt16 refNum, BOOL on)

Parameters

refNum

library reference number obtained via call to SysLibLoad

on

true if to power on the reader, false to power it down

Return value

True if success, otherwise false

Comments

7.4. DPSDK_SCIIsCardPresent

Checks if smartcard is present in the reader

Syntax

Boolean DPSDK_SCIIsCardPresent(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if smartcard is present

Comments

7.5. DPSDK_SCSetCardType

Sets smartcard type. Only MCU_CARD(1) cards are supported currently

Syntax

Boolean DPSDK_SCSetCardType(UInt16 refNum, UInt8 cardtype)

Parameters

refNum

library reference number obtained via call to SysLibLoad

cardtype

MCU_CARD(1)

Return value

True if success, otherwise false

Comments

7.6. DPSDK_SCResetCard

Resets smartcard and returns the Answer To Reset

Syntax

Boolean DPSDK_SCResetCard(UInt16 refNum, UInt8 *reset, UInt16 *resetlen, UInt8 *protocol)

Parameters

refNum

library reference number obtained via call to SysLibLoad

reset

buffer where to return the ATR

resetlen

size of the buffer. Upon return stores the size of the actual ATR

protocol

card protocol, currently only T0(0) is supported

Return value

True if success, otherwise false

Comments

7.7. DPSDK_SCAPDU

Executes an APDU command on the smartcard

Syntax

Boolean DPSDK_SCAPDU(UInt16 refNum, UInt8 CLA, UInt8 INS, UInt8 P1, UInt8 P2, void *data, Int16 datalen, void *receive, Int16 *rcvlen, UInt8 *ST1, UInt8 *ST2)

Parameters

refNum

library reference number obtained via call to SysLibLoad

CLA

the CLA byte

INS

the INS byte

P1

the P1 byte

P2

the P2 byte

data

data to send to the card, pass NULL if you don't want to send any

datalen

length of the data in the data buffer

receive

buffer where to store the received data or NULL if you don't want any

rcvlen

size of the receive buffer, upon return the size of the data being read is returned

ST1

status byte 1

ST2

status byte 2

Return value

True if success, otherwise false

Comments

7.8. DPSDK_SCSelectFile

Selects a file in the smartcard. The file needs to be selected before performing any read/write operations to it. Not all smartcards support files

Syntax

Int16 DPSDK_SCSelectFile(UInt16 refNum, UInt16 fileid)

Parameters

refNum

library reference number obtained via call to SysLibLoad

fileid

file identifier

Return value

Error code, one of the following:

Value	Meaning
SCErrNone	Success
SCErrFailed	Generic communication error
SCErrFileNotFound	File not found
SCErrRecordNotFound	Record not found
SCErrInvalidLength	Invalid data length
SCErrNoFileSelected	You have to select file first

Comments

7.9. DPSDK_SCReadRecord

Reads a record from the previously selected file with [DPSDK_SCSelectFile](#)

Syntax

Int16 DPSDK_SCReadRecord(UInt16 refNum, UInt8 recnum, UInt8 *databuf, UInt8 length)

Parameters

refNum

library reference number obtained via call to SysLibLoad

recnum

record number to read

databuf

data buffer where to store the data

length

length of the data buffer, upon return stores the actual number of bytes being read

Return value

Error code, one of the following:

Value	Meaning
SCErrNone	Success
SCErrFailed	Generic communication error
SCErrFileNotFound	File not found
SCErrRecordNotFound	Record not found
SCErrInvalidLength	Invalid data length
SCErrNoFileSelected	You have to select file first

Comments

7.10. DPSDK_SCWriteRecord

Writes a record to the previously selected file with [DPSDK_SCSelectFile](#)

Syntax

Int16 DPSDK_SCWriteRecord(UInt16 refNum, UInt8 recnum, UInt8 *databuf, UInt8 length)

Parameters

refNum

library reference number obtained via call to SysLibLoad

recnum

record number to write to

databuf

pointer to data buffer with the data to write

length

length of the data in databuf

Return value

Error code, one of the following:

Value	Meaning
SCErrNone	Success
SCErrFailed	Generic communication error
SCErrFileNotFound	File not found
SCErrRecordNotFound	Record not found
SCErrInvalidLength	Invalid data length
SCErrNoFileSelected	You have to select file first

Comments

7.11. DPSDK_SCSetCardSlot

Selects the slot where the smartcard is inserted. Only PP-55 supports 2 slots

Syntax

Boolean DPSDK_SCSetCardSlot(UInt16 refNum, UInt8 slot)

Parameters

refNum

library reference number obtained via call to SysLibLoad

slot

one of the following:

Value	Meaning
SC_MAINSLOT(0)	main slot (SIM)
SC_SAMSLOT(1)	additional slot (SAM)

Return value

True if success, otherwise false

Comments

8. External Communication

A set of functions allowing communication with devices attached to the printer. Only PP-50 and PP-55 support external devices

8.1. DPSDK_ExtOpenConnection

Initializes external connection with device attached to PP-50's port. The printer must be initialized successfully before. The function sleeps the printer, terminates current connection and initializes a new one with the specified parameters.

Syntax

Boolean DPSDK_ExtOpenConnection(UInt16 refNum, UInt32 baudrate, UInt8 databits, UInt8 stopbits, UInt8 parity)

Parameters

refNum

library reference number obtained via call to SysLibLoad

baudrate

databits

stopbits

parity

Return value

True if success, otherwise false

Comments

8.2. DPSDK_ExtSendData

Sends data to external device.

Syntax

Boolean DPSDK_ExtSendData(UInt16 refNum, void *buf, UInt32 length, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

pointer to buffer holding the data

length

the number of bytes to send

timeout

timeout in seconds

Return value

True if success, otherwise false

Comments

8.3. DPSDK_ExtReceiveData

Receives data from external device.

Syntax

Int32 DPSDK_ExtReceiveData(UInt16 refNum, void *buf, UInt32 length, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

pointer to buffer to store the data

length

the maximum number of bytes to receive

timeout

timeout in seconds

Return value

Number of bytes received or error (-1)

Comments

8.4. DPSDK_ExtCloseConnection

Closes the external connection and re-init back the connection with the printer

Syntax

Boolean DPSDK_ExtCloseConnection(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if success. If this function fails that means that the re-initialization of the normal connection failed.
So the printer is not available at that time

Comments

8.5. DPSDK_ExtGetDTR

Syntax

Boolean DPSDK_ExtGetDTR(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Comments

8.6. DPSDK_ExtSetDTR

OBSOLETE!

Set the DTR signal status

Syntax

void DPSDK_ExtSetDTR(UInt16 refNum, Boolean DTR)

Parameters

refNum

library reference number obtained via call to SysLibLoad

DTR

True or False

Return value

Comments

8.7. DPSDK_ExtGetRTS

Syntax

Boolean DPSDK_ExtGetRTS(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Comments

8.8. DPSDK_ExtGetCTS

OBSOLETE!

Get the CTS signal status

Syntax

Boolean DPSDK_ExtGetCTS(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if ON

Comments

8.9. DPSDK_ExtGetDSR

OBSOLETE!

Get the DSR signal status

Syntax

Boolean DPSDK_ExtGetDSR(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if ON

Comments

8.10. DPSDK_ExtGetBreak

OBSOLETE!

Get the Break signal status

Syntax

Boolean DPSDK_ExtGetBreak(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

True if ON

Comments

8.11. DPSDK_ExtSetBreak

Syntax

void DPSDK_ExtSetBreak(UInt16 refNum, Boolean Break)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Break

True or False

Return value

Comments

8.12. DPSDK_ExtReceiveDataStop

Receives data from external device until a timeout or Stop-value found.

Syntax

Int32 DPSDK_ExtReceiveDataStop(UInt16 refNum, void *buf, UInt32 length, UInt16 timeout, UInt8 stop)

Parameters

refNum

library reference number obtained via call to SysLibLoad

buf

pointer to buffer to store the data

length

the maximum number of bytes to receive

timeout

timeout in seconds

stop

the stop value

Return value

Number of bytes received or error (-1)

Comments

8.13. DPSDK_ExtSetChannel

Syntax

Boolean DPSDK_ExtSetChannel(UInt16 refNum, UInt8 channel)

Parameters

refNum

library reference number obtained via call to SysLibLoad

channel

Return value

Comments

8.14. DPSDK_ExtGetLibRefNum

Syntax

UInt16 DPSDK_ExtGetLibRefNum(UInt16 refNum)

Parameters

refNum

library reference number obtained via call to SysLibLoad

Return value

Comments

8.15. DPSDK_ExtExecCommand

Syntax

Int16 DPSDK_ExtExecCommand(UInt16 refNum, UInt8 command, void *buf, UInt32 length, void *responsedata, UInt32 *responselength, UInt16 timeout)

Parameters

refNum

library reference number obtained via call to SysLibLoad

command

command number

buf

data buffer or null

length

length of data buffer

responsedata

response data bufffer

responselength

response length

timeout

timeout in seconds

Return value

error code or 0 if success

Comments

9. Unfinished

Unfinished documentation part