API Reference Guide

Linux SDK

Ver. 1.01

AP3

Table of Contents

1. Manual guide

This SDK manual describes the contents of the library required to develop Linux OS application programs.

1-1 Supported Kernel & Platform

• Kernel
  - Kernel 2.6.32 or higher

• Platform
  - Linux 32bit / 64bit
  - Raspberry PI

• O/S
  - openSUSE 11.3 32bit / 64bit
  - Red Hat Enterprise Linux 7.3 64bit
  - CentOS 6.6 32bit / 64bit
  - Ubuntu 10.04 LTS 32bit / 64bit

1-2 Supported Interfaces

• USB, Serial, Ethernet

1-3 Supported Printers

Model	DPI	Max Printable Width
AP3	180 dpi	512 dots
AP3-3	203 dpi	576 dots

2. Property

The constants used by the library are declared in Ap3Const.h. The development environment is based on C.

2-1 CharacterSet (LONG R/W)

• This is the property for defining the printer's code page and set to CS_437 by default. The values can be set and read using SetCharSet() and GetCharSet().

The following code pages can be used:

Constant	Value	Description
CS_PC437	0	Code page PC437
CS_KATAKANA	1	Katakana
CS_PC850	2	Code page PC850
CS_PC860	3	Code page PC860
CS_PC863	4	Code page PC863
CS_PC865	5	Code page PC865
CS_WPC1252	16	Code page WPC1252
CS_PC866	17	Code page PC866
CS_PC852	18	Code page PC852
CS_PC858	19	Code page PC858
CS_PC864	22	Code page PC864
CS_THAI42	23	Code page THAI42
CS_WPC1253	24	Code page WPC1253
CS_WPC1254	25	Code page WPC1254
CS_WPC1257	26	Code page WPC1257
CS_FARSI	27	Code page Farsi
CS_WPC1251	28	Code page WPC1251
CS_PC737	29	Code page PC737
CS_PC775	30	Code page PC775
CS_THAI14	31	Code page THAI14
CS_PC862	33	Code page PC862
CS_PC855	36	Code page PC855
CS_PC857	37	Code page PC857
CS_PC928	38	Code page PC928
CS_THAI16	39	Code page THAI16
CS_WPC1256	40	Code page PC1256
CS_PC1258	41	Code page PC1258
CS_KHMER	42	Code page KHMER
CS_PC1250	47	Code page PC1250
CS_USER	255	User set page

* Example

```
ConnectToPrinter(port);

……

SetCharSet(CS_PC850);

…...

int32 nCharSet;

nCharSet = GetCharSet();

……
```

2-2 International CharacterSet (LONG R/W)

• This is the property for defining International character Set and set to ICS_USA by default. The values can be set or read using SetInternationalChar() and GetInternationalChar().

⚠ **Note**	CharacterSet settings may need to be verified in the following cases 1. When character strings other than the one you tried to print are printed 2. When a broken string is printed in the same form as hieroglyphic characters 3. When characters are printed in the form of '?' (question mark)

The following International character Set can be used:

Constant	Value	Description
ICS_USA	0	USA code
ICS_FRANCE	1	FRANCE code
ICS_GERMANY	2	GERMANY code
ICS_UK	3	UK code
ICS_DENMARK1	4	DENMARK1 code
ICS_SWEDEN	5	SWEDEN code
ICS_ITALY	6	ITALY code
ICS_SPAIN	7	SPAIN code
ICS_NORWAY	9	NORWAY code
ICS_DENMARK2	10	DENMARK 2 code
ICS_SPAIN2	11	SPAIN 2 code
ICS_LATIN	12	LATIN AMERICA code
ICS_KOREA	13	KOREA code

* Example

```
ConnectToPrinter(port)

……

SetInternationalChar(ICS_SPAIN);

……

int32 nCharSet;

nCharSet = GetInternationalChar();

……
```

2-3 State (LONG R)

• This is the property that sets the printer status. It is read only and calls GetStatus() to read the printer status. The status value can be set in duplicate and each value can be checked using bitwise operation.

These are the printer status values:

Constant	Value	Description
STS_NORMAL	0	The printer status is normal.
STS_PAPEREMPTY	1	There is no paper.
STS_COVEROPEN	2	The paper cover is open.

* Example

```
ConnectToprinter(port)
……

SetAutoStatusCheck(true);

int status = GetStatus();


if ((status & STS_PAPEREMPTY) == STS_PAPEREMPTY)
    ……
if ((status & STS_COVEROPEN) == STS_COVEROPEN)
    ……

…….
```

3. Method

The functions provided by Linux SDK are declared in Ap3PosAPI.h.
The development environment is based on C.

3-1 ConnectToPrinter

• Set the connection for communication with the printer.

int ConnectToPrinter(const char *port)

[Parameters]

* const char *port
  [in] Interface to be connected to the printer

Interface	Input Data	Example
USB	USB:	ConnectToPrinter(“USB:”)
Serial	serial:(baudrate) /dev/ttyX:(baudrate)	ConnectToPrinter(“serial:115200”) ConnectToPrinter(“/dev/tty0:115200”)
Ethernet	IP address, port no.	ConnectToPrinter(“192.168.0.10:9100”)

[Return Values]

Constant	Value	Description
SUCCESS	0	The operation is successful.
PORT_OPEN_ERROR	-99	The communication port cannot be opened.
NO_CONNECTED_PRINTER	-100	The printer is not connected.
NO_AP3_PRINTER	-101	It is not a AP3 printer.

* Example

```
int ret;

// USB
ret = ConnectToPrinter(“USB:”);

// Serial
ret = ConnectToPrinter(“serial:115200”);

// Ethernet
ret = ConnectToPrinter(“192.168.0.10:9100”);
……
```

3-2 DisconnectPrinter

• Disconnect the printer.

void DisconnectPrinter();

[Parameters]

None

[Return Values]

None

* Example

```
ConnectToPrinter(……..);

……

DisconnectPrinter();
```

3-3 InitializePrinter

• Cancel the existing settings and initialize.

int InitializePrinter();

[Parameters]

None

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.

* Example

```
……

InitializePrinter();

……
```

3-4 LineFeed

• Line feed to the amount of the integer value set as a factor.

int LineFeed (const unsigned int lineNum);

[Parameters]

* const unsigned int lineNum
  [in] Send the number of line feeding lines in an integer value as a factor.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
MEM_ALLOC_ERROR	-120	The allocation of internal memory failed.

* Example

```
ConnectToPrinter(……..);

……

LineFeed(5);

……
```

3-5 SetLeftMargin

• Set the left margin.

int SetLeftMargin (long margin);

[Parameters]

* long margin
  [in] Margin size (0 ~ 65535)

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
BAD_ARGUMENT	-117	Incorrect values have been entered.

* Example

```
ConnectToPrinter(……..);

……

SetLeftMargin(10);

……
```

3-6 SetUpsideDown

• Set the upside-down function.

int SetUpsizeDown (bool upsideDown);

[Parameters]

* bool upsideDown
  [in] Enable/disable the upside-down function.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.

* Example

```
ConnectToPrinter(……..);

……

SetUpsideDown(true);

……
```

3-7 PartialCut

• Enable the partial cut function.

int PartialCut();

[Parameters]

None

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

PartialCut();

……
```

3-8 OpenCashDrawer

• Open the cash drawer.

int OpenCashDrawer (unsigned int milliSec);

[Parameters]

* unsigned int milliSec
  [in] Set the open signal length (0 ~ 255).

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

OpenCashDrawer(100);

……
```

3-9 PrintText

• Print texts.

int PrintText(const char* text, const int alignment, const unsigned int attribute,
const unsigned int textSize);

[Parameters]

* const char* text
[in] String with null as a terminator. Send the text data to print.

* const int alignment
[in] Set the text alignment.

Constant	Value	Description
ALIGNMENT_LEFT	0	Align to the left
ALIGNMENT_CENTER	1	Align to the center
ALIGNMENT_RIGHT	2	Align to the right

* const unsigned int attribute
[in] Set the text attributes. The following values can be applied in duplicate.

Constant	Value	Description
ATTR_FONTTYPE_A	0	Set to Font A. (default)
ATTR_FONTTYPE_B	1	Set to Font B
ATTR_FONTTYPE_C	2	Set to Font C
ATTR_BOLD	4	Add Bold
ATTR_UNDERLINE_1	8	Add 1-dot underline
ATTR_UNDERLINE_2	16	Add 2-dot underline
ATTR_REVERSE	32	Add the reverse text attribute.

* const unsigned int textSize
[in] Set the text size. The width and height scale can be used in duplicate.

Constant	Value	Description
TS_WIDTH_0	0x00	Set the width scale to x1
TS_WIDTH_1	0x10	Set the width scale to x2
TS_WIDTH_2	0x20	Set the width scale to x3
TS_WIDTH_3	0x30	Set the width scale to x4
TS_WIDTH_4	0x40	Set the width scale to x5
TS_WIDTH_5	0x50	Set the width scale to x6
TS_WIDTH_6	0x60	Set the width scale to x7
TS_WIDTH_7	0x70	Set the width scale to x8

Constant	Value	Description
TS_HEIGHT_0	0x00	Set the height scale to x1
TS_HEIGHT_1	0x01	Set the height scale to x2
TS_HEIGHT_2	0x02	Set the height scale to x3
TS_HEIGHT_3	0x03	Set the height scale to x4
TS_HEIGHT_4	0x04	Set the height scale to x5
TS_HEIGHT_5	0x05	Set the height scale to x6
TS_HEIGHT_6	0x06	Set the height scale to x7
TS_HEIGHT_7	0x07	Set the height scale to x8

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
STATUS_ERROR	-103	Not ready to print.
WRITE_ERROR	-105	Data transmission failed.

* Example

```
ConnectToPrinter(……..);

……

PrintText("AP3 Linux SDK Text.\n", ALIGNMENT_LEFT,
            ATTR_FONTTYPE_A, TS_HEIGHT_0 | TS_WIDTH_0);
……
```

3-10 PrintTextW

• Print 2Bytes texts.

int PrintTextW(const char* text, const int alignment, const unsigned int attribute,
const unsigned int textSize, const unsigned int codePage);

[Parameters]

* const char* text
[in] String with null as a terminator. Send the text data to print.

* const int alignment
[in] Set the text alignment.

Constant	Value	Description
ALIGNMENT_LEFT	0	Align to the left
ALIGNMENT_CENTER	1	Align to the center
ALIGNMENT_RIGHT	2	Align to the right

* const unsigned int attribute
[in] Set the text attributes. The following values can be applied in duplicate.

Constant	Value	Description
ATTR_FONTTYPE_A	0	Set to Font A. Print with the default device font.
ATTR_FONTTYPE_B	1	Set to Font B
ATTR_FONTTYPE_C	2	Set to Font C
ATTR_BOLD	4	Add Bold
ATTR_UNDERLINE_1	8	Add 1-dot underline
ATTR_UNDERLINE_2	16	Add 2-dot underline
ATTR_REVERSE	32	Add the reverse text attribute.

* const unsigned int textSize
[in] Set the text size. The width and height scale can be used in duplicate.

Constant	Value	Description
TS_WIDTH_0	0x00	Set the width scale to x1
TS_WIDTH_1	0x10	Set the width scale to x2
TS_WIDTH_2	0x20	Set the width scale to x3
TS_WIDTH_3	0x30	Set the width scale to x4
TS_WIDTH_4	0x40	Set the width scale to x5
TS_WIDTH_5	0x50	Set the width scale to x6
TS_WIDTH_6	0x60	Set the width scale to x7
TS_WIDTH_7	0x70	Set the width scale to x8

Constant	Value	Description
TS_HEIGHT_0	0x00	Set the height scale to x1
TS_HEIGHT_1	0x01	Set the height scale to x2
TS_HEIGHT_2	0x02	Set the height scale to x3
TS_HEIGHT_3	0x03	Set the height scale to x4
TS_HEIGHT_4	0x04	Set the height scale to x5
TS_HEIGHT_5	0x05	Set the height scale to x6
TS_HEIGHT_6	0x06	Set the height scale to x7
TS_HEIGHT_7	0x07	Set the height scale to x8

* const unsigned int codePage
[in] Set the encoding type for character strings.

Constant	Value	Description
ENCODING_ASCII	0	ASCII
ENCODING_EUCKR	1	Korean (EUC-KR)
ENCODING_CP949	2	Korean (CP949)
ENCODING_EUCCN	3	Chinese (EUC-CN)
ENCODING_GB18030	4	Chinese (GB18030)
ENCODING_BIG5	5	Chinese (BIG5)
ENCODING_CP950	6	Chinese (CP950)
ENCODING_EUCJP	7	Japanese (EUC-JP)
ENCODING_CP932	8	Japanese (CP932)
ENCODING_CP874	9	Thai (CP874)

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
STATUS_ERROR	-103	Not ready to print.
WRITE_ERROR	-105	Data transmission failed.

* Example

```
ConnectToPrinter(……..);

……

PrintTextW("Korean Printing Test.\n", ALIGNMENT_LEFT,
ATTR_FONTTYPE_A, TS_HEIGHT_0 | TS_WIDTH_0, ENCODING _CP949);
……
```

3-11 PrintBarcode

• Print 1- and 2-dimensional barcodes.

```
int PrintBarcode(const int barcodeType, const char* barcodeData,
                 const barcodeInfo_s* barcodeInfo);
```

[Parameters]

* const int barcodeType
  [in] Set the barcode type. Defined in Ap3Const.h.

Constant	Value	Description
BARCODE_UPCA	0	Print UPC-A barcode.
BARCODE_UPCE	1	Print UPC-E barcode.
BARCODE_EAN13 BARCODE_JAN13	3 5	Print JAN-13(EAN-13) barcode.
BARCODE_EAN8 BARCODE_JAN8	2 4	Print JAN-8(EAN-8) barcode.
BARCODE_ITF	6	Print ITF barcode.
BARCODE_CODABAR	7	Print CODABAR barcode.
BARCODE_CODE39	8	Print CODE39 barcode.
BARCODE_CODE93	9	Print CODE93 barcode.
BARCODE_CODE128	10	Print CODE128 barcode.
BARCODE_PDF417	11	Print PDF417 barcode.
BARCODE_QRCODE	12	Print QR CODE barcode.
BARCODE_DATAMATRIX	13	Print DATAMATRIX barcode.

* const char* barcodeData
  [in] Send the barcode data to print.

* const barcodeInfo_s* barcodeInfo
  [in] Structure for storing barcode attributes

```
Struct _barcodeInfo
{
    unsigned int mode;
    unsigned int height;
    unsigned int width;
    unsigned int alignment;
    unsigned int textPosition;
    unsigned int attribute;
};
```

unsigned int mode
[in] Send the model value when printing QR Code.

Constant	Value	Description
BARCODE_QR_MODEL1	1	Model 1
BARCODE_QR_MODEL2	2	Model 2

unsigned int height
[in] Set the barcode height (1~255). If the barcode is larger than the paper size, the barcode may not be printed. 2-dimensional barcodes are not subject to this value.

unsigned int width
[in] Set the barcode width (2~7). If the barcode is larger than the paper size, the barcode may not be printed. 2-dimensional barcodes are not subject to this value.

unsigned int alignment
[in] Set the barcode alignment.

Constant	Value	Description
ALIGNMENT_LEFT	0	Align to the left
ALIGNMENT_CENTER	1	Align to the center
ALIGNMENT_RIGHT	2	Align to the right

unsigned int textPosition
[in] Set the barcode data printing position. 2-dimensional barcode has only BARCODE_TEXT_NONE .

Constant	Value	Description
BARCODE_TEXT_NONE	0	No barcode data is printed.
BARCODE_TEXT_ABOVE	1	The barcode data is printed above the barcode.
BARCODE_TEXT_BELOW	2	The barcode data is printed below the barcode.

unsigned int attribute
[in] Set the separator height of 2D and 1D barcodes when printing GS1 (1 or 2).

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
WRONG_BARCODE_TYPE	-115	The barcode type is not supported.
WRONG_BC_DATA_ERROR	-116	The barcode data is incorrect.

* Example

```
ConnectToPrinter(……..);

char* barcodeData = “123456789012”;
barcodeInfo_s barcodeInfo;

……

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_UPCA, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_UPCE, barcodeData, &barcodeInfo);



barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_EAN13, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_JAN13, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_EAN8, “12345678”, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_JAN8, “12345678”, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_CODE39, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_CODE93, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_CODE128, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
```

```
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_ITF, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCDE_CODABAR, barcodeData, &barcodeInfo);

// *************** 2D barcode

barcodeInfo.height = 0;
barcodeInfo.width = 2;
barcodeInfo.attribute = 0;
PrintBarcode(BARCDE_PDF417, barcodeData, &barcodeInfo);

barcodeInfo.mode = BARCODE_QR_MODEL1;
barcodeInfo.height = 0;
barcodeInfo.width = 2;
barcodeInfo.attribute = 0;
PrintBarcode(BARCDE_QRCODE, barcodeData, &barcodeInfo);

barcodeInfo.mode = BARCODE_QR_MODEL2;
barcodeInfo.height = 0;
barcodeInfo.width = 2;
barcodeInfo.attribute = 0;
PrintBarcode(BARCDE_QRCODE, barcodeData, &barcodeInfo);

barcodeInfo.height = 0;
barcodeInfo.width = 2;
barcodeInfo.attribute = 0;
PrintBarcode(BARCDE_DATAMATRIX, barcodeData, &barcodeInfo);
```

3-12 DirectIO

• Send and read the user-defined data.

int DirectIO(char* writeData, int writeLen, char* readData, int* readLen,
unsigned int mTimeout);

[Parameters]

* char* writeData,
[in] Data to be sent to the printer

* int writeLen
[in] Size of data to be sent to the printer
write does not function if 0 is entered to writeData for NULL, writeLen.

* char* readData,
[in] Data to be sent to the printer

* int* readLen
[in] Read the size of the data to be read by the caller.
read does not function if 0 is entered to readData for NULL, readLen.

* unsigned int mTimeout
[in]] Waiting time before reading the data. Even if no data has been read, it returns after the set amount of time. If set to 0, it waits until there is incoming data.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
READ_TIMEOUT	-127	It timed out before reading the data.

* Example

```
char cmd[3] = { 0x10, 0x04, 0x01};
char readBuf[20] = {0x00,};
int readLen;

ConnectToPrinter(……..);

DirectIO(cmd, sizeof(cmd), readBuf, &readLen, 0);

……
```

3-13 PrintImage

• Print image files.

int PrintImage (const char *imagePath, const bool compress,
const unsigned int alignment);

[Parameters]

* const char *imagePath
[in] String for the image file path. JPG, BMP and GIF are supported.

* const bool compress
[in] Set whether to compress RLE image.

* const unsigned int alignment
[in] Set the image alignment.

Constant	Value	Description
ALIGNMENT_LEFT	0	Align to the left
ALIGNMENT_CENTER	1	Align to the center
ALIGNMENT_RIGHT	2	Align to the right

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
IMAGE_OPEN_ERROR	-118	The image file cannot be opened.
MEM_ALLOC_ERROR	-120	The allocation of internal memory failed.

* Example

```
ConnectToPrinter(……..);

……

PrintBitmap(filePath, true, ALIGNMENT_CENTER);

……
```

3-14 DownloadNVImage

• Save images to the non-volatile memory area of the printer.

int DownloadNVImage (const char *imagePath, const unsigned int keyCode);

[Parameters]

* const char *imagePath
  [in] String for the image file path. JPG, BMP and GIF are supported.

* const unsigned int keyCode
  [in] Address of the memory area where image is stored (0 ~ 255).

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
IMAGE_OPEN_ERROR	-118	The image file cannot be opened.
MEM_ALLOC_ERROR	-120	The allocation of internal memory failed.

* Example

```
ConnectToPrinter(……..);

……

DownloadNVImage(filePath, 0x01);

……
```

3-15 PrintNVImage

• Print the images stored in the non-volatile memory area of the printer.

int PrintNVImage (const unsigned int keyCode);

[Parameters]

* const unsigned int keyCode
  [in] Address code of the image to be printed (0 ~ 255)

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.

* Example

```
ConnectToPrinter(……..);

……

DownloadNVImage(filePath, 0x01);

……

PrintNVImage(0x01);
```

3-16 RemoveAllNVImage

• Remove all the images stored in the non-volatile memory area of the printer.

int RemoveAllNVImage ();

[Parameters]

None

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);
……

RemoveAllNVImage();
```

3-17 RemoveNVImage

• Remove the images with the specified address stored in the non-volatile memory area of the printer.

int RemoveNVImage (const unsigned int keyCode);

[Parameters]

* const unsigned int keyCode
  [in] Address code of the image to be printed (0 ~ 255)

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

DownloadNVImage(filePath, 0x01);
…….

RemoveNVImage(0x01);
```

3-18 GetNVImageKeyCode

• Read the address list of the images stored in the non-volatile memory area of the printer.

int GetNVImageKeyCode (char *keyCodeList, unsigned int *listLen);

[Parameters]

* char *keyCodeList
  [in, out] Buffer to save the list of image address

* unsinged int *listLen
  [in, out] Length of keyCodeList

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NV_NO_KEY	-121	No NV key is defined.
WRONG_RESPONSE	-122	Incorrect NV data response
NOT_SUPPORT	-124	The function is not supported.

* Example

```
char keyList[128] = {0x00, };
unsigned int listLen = 0;
int ret;

ConnectToPrinter(……..);

DownloadNVImage(filePath, 0x01);

…….

ret = GetNVImageKeyCode(keyList, &listLen);
```

3-19 SetAutoStatusCheck

• Enable/disable ASB mode to check the printer status (cover open, no paper).

int SetAutoStatusCheck(bool enable);

[Parameters]

* bool enable
[in] Enable/disable ASB mode.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.

* Example

```
int status = 0x00;

ConnectToPrinter(……..);

SetAutoStatusCheck(true);

…...

status = GetStatus();
……

if ((status & STS_PAPEREMPTY) == STS_PAPEREMPTY)
    ……
```

3-20 GetStatus

• Read the printer status (cover open, no paper).

int GetStatus();

[Parameters]

None

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
READ_TIMEOUT	-127	No status data

* Example

```
int status = 0x00;

ConnectToPrinter(……..);

SetAutoStatusCheck(true);

…...

status = GetStatus();
……

if ((status & STS_PAPEREMPTY) == STS_PAPEREMPTY)
    ……
```

3-21 SelectPageMode

• Enable/disable Page Mode.

int SelectPageMode(bool pageMode);

[Parameters]

* bool pageMode
  [in] Set whether to use Page Mode.
  Page Mode is selected if set to TRUE.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

// Select Page Mode
if (SelectPageMode(true) != SUCCESS)
    return;

…....

// Select Standard Mode
if (SelectPageMode(false) != SUCCESS)
    return;

……
```

3-22 PrintDataInPM

• Prints all the data in the printer buffer if set to Page Mode and the printer is switched to Standard Mode after printing.

int PrintDataInPM();

[Parameters]

None

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

// Select Page Mode
if (SelectPageMode(true) != SUCCESS)
    return;

……

PrintDataInPM();
```

3-23 SetPrintAreaInPM

• Sets the size and position of the printing area when set to Page Mode.

int SetPrintAreaInPM (long x, long y, long width, long height);

[Parameters]

* long x
  [in] x-coordinates of the printing area

* long y
  [in] y-coordinates of the printing area

* long width
  [in] width of the printing area

* long height
  [in] height of the printing area

Width of 78mm: x = 0, y = 0, width = 512, height = 840

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
BAD_ARGUMENT	-117	The specified argument is not correct.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

// Select Page Mode
if (SelectPageMode(true) != SUCCESS)
    return;

SetPrintAreaInPM(0, 0, 512, 512);

……

PrintDataInPM();
```

3-24 SetPrintDirectionInPM

• Set the printing direction in the Page Mode.

int SetPrintDirectionInPM (int printDirection);

[Parameters]

* int printDirection

Constant	Value	Direction	Starting Position	Rotation
PAGEMODE_ROTATE_0	48	Left -> Right	Top left	0 degree
PAGEMODE_ROTATE_90	51	Top -> Bottom	Top right	90 degrees
PAGEMODE_ROTATE_180	50	Right -> Left	Bottom right	180 degrees
PAGEMODE_ROTATE_270	49	Bottom -> Top	Bottom left	270 degrees

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
BAD_ARGUMENT	-117	The specified argument is not correct.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

// Select Page Mode
if (SelectPageMode(true) != SUCCESS)
    return;

SetPrintAreaInPM(0, 0, 512, 512);
SetPrintDirectionInPM(PAGEMODE_ROTATE_90);

……

PrintDataInPM();
```

3-25 SetVerticalPositionInPM

• Set the vertical position for printing in the Page Mode.

int SetVerticalPositionInPM (long position, bool relative);

[Parameters]

* long position
  [in] Starting position to be set

* bool relative
  [in] Set whether it is relative or absolute position from the current position.
  If TRUE, it is set to relative position.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
BAD_ARGUMENT	-117	The specified argument is not correct.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

// Select Page Mode
if (SelectPageMode(true) != SUCCESS)
    return;

SetPrintAreaInPM(0, 0, 512, 512);
SetPrintDirectionInPM(PAGEMODE_ROTATE_90);

SetVerticalPositionInPM(160, false);
SetHorizontalPositionInPM(40, false);
PrintText("AP3 Printer.", ALIGNMENT_CENTER, ATTR_FONTTYPE_A,
            TS_WIDTH_0 | TS_HEIGHT_0);

……

PrintDataInPM();
```

3-26 SetHorizontalPositionInPM

• Set the horizontal position for printing.

int SetHorizontalPositionInPM (long position, bool relative);

[Parameters]

* long position
  [in] Starting position to be set

* bool relative
  [in] Set whether it is relative or absolute position from the current position.
  If TRUE, it is set to relative position.

[Return Values]

Constant	Value	Description
SUCCESS	0	The function is successful.
WRITE_ERROR	-105	Data transmission failed.
BAD_ARGUMENT	-117	The specified argument is not correct.
NOT_SUPPORT	-124	The function is not supported.

* Example

```
ConnectToPrinter(……..);

……

// Select Page Mode
if (SelectPageMode(true) != SUCCESS)
    return;

SetPrintAreaInPM(0, 0, 512, 512);
SetPrintDirectionInPM(PAGEMODE_ROTATE_90);

SetVerticalPositionInPM(160, false);
SetHorizontalPositionInPM(40, false);
PrintText("AP3 Printer.", ALIGNMENT_CENTER, ATTR_FONTTYPE_A,
            TS_WIDTH_0 | TS_HEIGHT_0);

……

PrintDataInPM();
```

Copyright

Everint Co., Ltd. maintains ongoing efforts to enhance and upgrade the functions and quality of all our products.
In the following, product specifications and/or user manual content may be changed without prior notice.

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer “OFF”, before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer “OFF”.

Revision history

Rev.	Date	Page	Description
1.00	19.07.18	-	New
1.01	25.04.19	3	1.Updated supported OS 2.Add model AP3-3