BIXOLON®

Windows SDK API Reference Guide
Mobile Printer

Rev. 1.26

SPP-R200
SPP-R200II
SPP-R200III
SPP-R210
SPP-R220
SPP-R300
SPP-R310
SPP-R400
SPP-R410
SPP-R418

http://www.bixolon.com

■ Table of Contents

1. Manual Information

This SDK Manual provides information on the dll files required for Windows application program development.

We at BIXOLON maintain ongoing efforts to enhance and upgrade the functions and quality of all our products.   In following, product specifications and/or user manual content may be changed without prior notice.

1-1 Supporting O/S & Platform

- O/S
  - Microsoft Windows XP (32bit/64bit)
  - Microsoft Windows VISTA (32bit/64bit)
  - Microsoft Windows 7 (32bit/64bit)
  - Microsoft Windows 8 (32bit/64bit)
  - Microsoft Windows 8.1 (32bit/64bit)
  - Microsoft Windows 10 (32bit/64bit)
  - Microsoft Windows CE 5.0
  - Microsoft Windows CE 6.0 (supported after building with custom SDK)

- Platform
  - Windows Mobile 5
  - Windows Mobile 6

1-2 Supporting Devices

If your PDA cpu type is Armv4 or Armv4i, it is compatible with our module driver. (On Windows CE 6.0, your custom SDK has to be applied to Windows SDK to support your device.)

2. Properties

The constant values used in the dll files provided are declared in the bxlconst.h file.
The development environment was based on C++.

2-1 CharacterSet (LONG R/W)

As the attribute that defines the Code Page of the printer, the default setting is BXL_CS_437. Use the SetCharacterSet(), GetCharacterSet() functions to set values or recall previously set values.

The Code Pages that can be used are as follows.

Code	Value	Description
BXL_CS_PC437	0	Code page PC437
BXL_CS_KATAKANA	1	Katakana
BXL_CS_PC850	2	Code page PC850
BXL_CS_PC860	3	Code page PC860
BXL_CS_PC863	4	Code page PC863
BXL_CS_PC865	5	Code page PC865
BXL_CS_WPC1252	16	Code page WPC1252
BXL_CS_PC866	17	Code page PC866
BXL_CS_PC852	18	Code page PC852
BXL_CS_PC858	19	Code page PC858
BXL_CS_PC864	22	Code page PC864
BXL_CS_THAI42	23	Code page THAI42
BXL_CS_WPC1253	24	Code page WPC1253
BXL_CS_WPC1254	25	Code page WPC1254
BXL_CS_WPC1257	26	Code page WPC1257
BXL_CS_FARSI	27	Code page Farsi
BXL_CS_WPC1251	28	Code page WPC1251
BXL_CS_PC737	29	Code page PC737
BXL_CS_PC775	30	Code page PC775
BXL_CS_THAI14	31	Code page THAI14
BXL_CS_PC862	33	Code page PC862
BXL_CS_PC855	36	Code page PC855
BXL_CS_PC857	37	Code page PC857
BXL_CS_PC928	38	Code page PC928
BXL_CS_THAI16	39	Code page THAI16
BXL_CS_WPC1256	40	Code page PC1256
BXL_CS_PC1258	41	Code page PC1258
BXL_CS_KHMER	42	Code page KHMER
BXL_CS_PC1250	47	Code page PC1250
BXL_CS_USER	255	User set page

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

SetCharacterSet(BXL_CS_PC850);

…...

int nCharSet;

nCharSet = GetCharacterSet();

……
```

2-2 International CharacterSet (LONG R/W)

As the attribute that defines the International Character Set of the printer, the default setting is BXL_ICS_USA. Use the SetInterChrSet(), GetInterChrSet()functions to set values or recall previously set values.

The International Character Sets that can be used are as follows.

Code	Value	Description
BXL_ICS_USA	0	USA code setting
BXL_ICS_FRANCE	1	FRANCE code setting
BXL_ICS_GERMANY	2	GERMANY code setting
BXL_ICS_UK	3	UK code setting
BXL_ICS_DENMARK1	4	DENMARK1 code setting
BXL_ICS_SWEDEN	5	SWEDEN code setting
BXL_ICS_ITALY	6	ITALY code setting
BXL_ICS_SPAIN	7	SPAIN code setting
BXL_ICS_NORWAY	9	NORWAY code setting
BXL_ICS_DENMARK2	10	DENMARK 2 code setting
BXL_ICS_SPAIN2	11	SPAIN 2 code setting
BXL_ICS_LATIN	12	LATIN AMERICA code setting
BXL_ICS_KOREA	13	KOREA code setting

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

SetInterChrSet(BXL_ICS_SPAIN);

…...

int nCharSet;

nCharSet = GetInterChrSet();

……
```

2-3 State (LONG R)

This attribute sets the printer State. It can only be read, and is automatically set when examining the printer State by using the CheckPrinter function. As State values can be set in duplication, each value can be confirmed via bit calculation. The values of this attribute can be recalled by using the GetState function.

Printer State values are as follows.

Code	Value	Description
BXL_STS_NORMAL	0	Printer status is normal
BXL_STS_PAPEREMPTY	1	No paper in printer
BXL_STS_COVEROPEN	2	The printer paper cover is open
BXL_STS_MSR_READY	8	Currently in MSR read mode, printing not possible
BXL_STS_ERROR	32	An error has occurred
BXL_STS_NOT_OPEN	64	The printer is not open

* Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

CheckPrinter();

…...

int iState;

iState = GetState();

if ((iState & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY)
    ……
if ((iState & BXL_STS_COVEROPEN) == BXL_STS_COVEROPEN)
    ……
…….
```

2-4 PowerValue (LONG R)

This attribute sets the battery capacity value of the printer. It can only be read, and is set every time the status of the printer battery changes. The values of this attribute can be recalled by using the GetPowerValue function.

Battery Capacity values are as follows.

Code	Value	Description
BXL_PWR_FULL	0	Battery charge status: 75% or higher
BXL_PWR_HIGH	1	Battery charge status: 50% or higher
BXL_PWR_MIDDLE	2	Battery charge status: 10% or higher
BXL_PWR_LOW	3	Battery charge status: under 10%

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

CheckPrinter();

…...

int iPower;

iPower = GetPowerValue();

switch (iPower)
{
    case BXL_PWR_FULL:
        ……
        break;
    case BXL_PWR_HIGH:
        …….
        break;
    case BXL_PWR_MIDDLE:
        ……
        break;
    case BXL_PWR_LOW:
        ……
        break;
}

…….
```

3. Methods

The functions disclosed in the provided module are declared in the ExportFunc.h file.
The development environment was based on C++.

3-1 PrinterOpen

In order to use the module, this function is enabled. For PDA types with different In/Out ports, port settings for each are required. In general, two-way communication is possible with just the strOutPortName setting.

```
Long PrinterOpen(
   LPCTSTR strOutPortName,
   Long lTimeout
);
```

[Parameters]

* strOutPortName
[in] Unicode Data ending with null. The virtual Serial Port number and Baudrate of the currently connected device is conveyed for printing. If strOutPortName is NULL, strOutPortName sets to In/Out data processing.(Bluetooth & Serial ex : COM4:19200 or BSP4:19200, WLAN ex : NETxxx.xxx.xxx.xxx:9100 or ETHxxx.xxx.xxx.xxx:9100, USB ex(Desktop only) : USB:)

* lTimeout
[in] Specifies the timeout, in milliseconds, used to calculate the total timeout period for open operations.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_CREATE_ERROR	-102	Failure to create communication target
BXL_CONNECT_ERROR	-105	Failure to process of connection
BXL_BAD_ARGUMENT	-108	Incorrect argument specified
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure

* Example

```
long long lResult;

// serial or bluetooth
lResult = PrinterOpen(“COM4:19200”, 1000);
……

// WLAN
lResult = PrinterOpen(“ETHxxx.xxx.xxx.xxx:9100”, 1000);
……

// USB (Desktop only)
lResult = PrinterOpen(“USB:”, 1000);
```

3-2 PrinterClose

This function disables the SDK module. Functions related to printing and MSR are no longer available for use.

Long PrinterClose();

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success

* Example

```
long lResult;

lResult = PrinterClose();
……
```

3-3 LineFeed

This function dictates line feeding to the integer value conveyed for printing.

```
Long LineFeed (
   int nFeed,
);
```

[Parameters]

* nFeed
[in] The integer value conveys the number of lines for line feeding in the form of printing value.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

lResult = LineFeed(10);

……
```

3-4 PrintBarcode

This function supports the 1-dimension and 2-dimension barcode printing function.
GS1 and AZTEC barcode can be printed by SPP-R200III.

```
Long PrintBarcode(
   PCHAR Data,
   long symbology,
   long Height,
   long Width,
   long Alignment,
   long TextPosition
);
```

[Parameters]

* Data
[in] ANSI Code Data ending with null. Barcode data to be printed are conveyed.

* symbology
[in] Defines the barcode type. The barcode type is defined in bxlconst.h.

Code	Value	Data Number Limit	Data Value Range
BXL_BCS_UPCA	101	11 ≤ n ≤ 12	48 ≤ data ≤ 57
BXL_BCS_UPCE	102	11 ≤ n ≤ 12	48 ≤ data ≤ 57
BXL_BCS_EAN13 BXL_BCS_JAN13	104 106	12 ≤ n ≤ 13	48 ≤ data ≤ 57
BXL_BCS_EAN8 BXL_BCS_JAN8	103 105	7 ≤ n ≤ 8	48 ≤ data ≤ 57
BXL_BCS_ITF	107	1 ≤ n ≤ 255 (even number)	48 ≤ data ≤ 57
BXL_BCS_CODABAR	108	1 ≤ n ≤ 255	48 ≤ data ≤ 57, 65 ≤ data ≤ 68, data =36,43,45,46,47,58
BXL_BCS_CODE39	109	1 ≤ n ≤ 255	48 ≤ data ≤ 57, 65 ≤ data ≤ 90, data =32,36,37,43,45,46,47
BXL_BCS_CODE93	110	1 ≤ n ≤ 255	0 ≤ data ≤ 127
BXL_BCS_CODE128	111	2 ≤ n ≤ 255	0 ≤ data ≤ 127
BXL_BCS_PDF417	200	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_QRCODE	202~3	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_DATAMATRIX	204	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_MAXICODE	205~6	2 ≤ n ≤ 928	0 ≤ data ≤ 255
BXL_BCS_GS1_xxxxx	50~64	Refer to Command Manual	0 ≤ data ≤ 255
BXL_BCS_AZTEC_xxxxx	210~212		0 ≤ data ≤ 255

* Height
[in] This value sets the barcode height in Dot units. The value range is 1~255 at one dimensional barcodes and GS1 UCC/EAN-128 barcode or the range is 1~8 at GS1 DataBar barcode. Two dimensional barcodes are not affected.

* Width
[in] This value sets the barcode width according to a value range of 2~7 at one dimensional barcodes or the range is 1~8 at GS1 DataBar barcode and AZTEC barcode. If the print area of the barcode exceeds the printing paper, barcode printing may not be possible. The other two dimensional barcodes are not affected by this value.

* Alignment
[in] This value sets the barcode alignment.

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

* TextPosition
[in] This value sets the print position of the barcode data. Two dimensional barcodes have only the BXL_BC_TEXT_NONE value.

Code	Value	Description
BXL_BC_TEXT_NONE	0	Barcode data not printed
BXL_BC_TEXT_ABOVE	1	Barcode data printed above barcode
BXL_BC_TEXT_BELOW	2	Barcode data printed below barcode

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_BC_DATA_ERROR	-500	Error in barcode data
BXL_BC_NOT_SUPPORT	-501	Barcode type not supported

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……
PrintBarcode(“123456789012”, BXL_BCS_UPCA, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_UPCE, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_EAN13, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_JAN13, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“12345678”, BXL_BCS_EAN8, 50, 2, BXL_ALIGNMENT_LEFT, BXL_BC_TEXT_BELOW);

PrintBarcode(“12345678”, BXL_BCS_JAN8, 50, 2, BXL_ALIGNMENT_LEFT, BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_Code39, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_ITF, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);


PrintBarcode(“123456789012”, BXL_BCS_Codabar, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_Code93, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_Code128, 50, 2, BXL_ALIGNMENT_LEFT,
                    BXL_BC_TEXT_BELOW);

PrintBarcode(“123456789012”, BXL_BCS_PDF417, 0, 2, BXL_ALIGNMENT_LEFT, 0);

PrintBarcode(“123456789012”, BXL_BCS_QRCODE_MODEL1, 0, 2, BXL_ALIGNMENT_LEFT, 0);

PrintBarcode(“123456789012”, BXL_BCS_QRCODE_MODEL2, 0, 2, BXL_ALIGNMENT_LEFT, 0);

PrintBarcode(“123456789012”, BXL_BCS_DATAMATRIX, 0, 2, BXL_ALIGNMENT_LEFT, 0);

PrintBarcode(“123456789012”, BXL_BCS_MAXICODE_MODE4, 0, 2, BXL_ALIGNMENT_LEFT, 0);

PrintBarcode(“123456789012”, BXL_BCS_AZTEC_DATAMODE, 0, 2, BXL_ALIGNMENT_LEFT, 50);

PrintBarcode(“123456|abcde”, BXL_BCS_GS1_RSS14, 0, 2, BXL_ALIGNMENT_LEFT, 50);

……
```

3-5 PrintText

This function supports the text printing function.

```
Long PrintText(
   LPCTSTR Data,
   long Alignment,
   long Attribute,
   long TextSize
);
```

[Parameters]

* Data
[in] Unicode Data ending with null.   Conveys the barcode data to be printed.

* Alignment
[in] This value sets the barcode alignment.

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

* Attribute
[in] This value sets the text properties.   The following values can be duplicated.

Code	Value	Description
BXL_FT_DEFAULT	0	Basic setting value Font A, print using the basic device font
BXL_FT_FONTB	1	Set to Font B
BXL_FT_FONTC	2	Set to Font C
BXL_FT_BOLD	2	Adds Bold lettering attribute
BXL_FT_UNDERLINE	4	Adds Underlining attribute, set at 1-dot thick
BXL_FT_UNDERTHICK	6	Adds Underlining attribute, set at 2-dot thick
BXL_FT_REVERSE	8	Adds Reverse lettering attribute

* TextSize
[in] This value sets the text size properties. The width and height values can be uplicated.

Code	Value	Description
BXL_TS_0WIDTH	0	Width magnification set to x1
BXL_TS_1WIDTH	16	Width magnification set to x2
BXL_TS_2WIDTH	32	Width magnification set to x3
BXL_TS_3WIDTH	48	Width magnification set to x4
BXL_TS_4WIDTH	64	Width magnification set to x5
BXL_TS_5WIDTH	80	Width magnification set to x6
BXL_TS_6WIDTH	96	Width magnification set to x7
BXL_TS_7WIDTH	112	Width magnification set to x8

Code	Value	Description
BXL_TS_0HEIGHT	0	Height magnification set to x1
BXL_TS_1HEIGHT	1	Height magnification set to x2
BXL_TS_2HEIGHT	2	Height magnification set to x3
BXL_TS_3HEIGHT	3	Height magnification set to x4
BXL_TS_4HEIGHT	4	Height magnification set to x5
BXL_TS_5HEIGHT	5	Height magnification set to x6
BXL_TS_6HEIGHT	6	Height magnification set to x7
BXL_TS_7HEIGHT	7	Height magnification set to x8

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

PrintText("Bixolon Mobile Printer.\n", BXL_ALIGNMENT_LEFT, BXL_FT_DEFAULT,
            BXL_TS_0WIDTH | BXL_TS_0HEIGHT);
……
```

3-6 PrintBitmap

This function prints image files. (Bmp, jpg, Gif) but standard ce o/s Only support bitmap format.

```
Long PrintBitmap (
  LPCTSTR FileName,
  long Width,
  long Alignment,
  long Level
);
```

[Parameters]

* FileName
[in] Expressed as a row of Unicode characters, the full path of the image file is set.

* Width
[in] This value is the width of the image file to be converted.   The value range is 0~384.
When setting the value as below, the image size is modified to fit conditions.

Code	Value	Description
BXL_WIDTH_FULL	-1	Set to a value of 384, image is fit to full size of paper
BXL_WIDTH_NONE	-2	No variation given to image size

* Alignment
[in] This value sets the barcode alignment.

Code	Value	Description
BXL_ALIGNMENT_LEFT	0	Left-aligned
BXL_ALIGNMENT_CENTER	1	Centered
BXL_ALIGNMENT_RIGHT	2	Right-aligned

* Level
[in] This value sets the color level of the image.   The value range is 0 ~ 100.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_BITMAPLOAD_ERROR	-400	Failure to read image file

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

PrintBitmap(strDirBmp, 300, BXL_ALIGNMENT_CENTER, 50);

……
```

3-7 CheckPrinter

This function checks printer status(cover open, paper empty) and battery status.
You can get the checked status info using GetStat() and GetPowerValue().

Long CheckPrinter ();

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_READ_ERROR	-301	Failure to receive data
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

CheckPrinter();

…...

int iState;

iState = GetState();

if ((iState & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY)
     ……

int iPower;

iPower = GetPowerValue();

…….
```

3-8 InitializePrinter

This method cancels conditions previously set and initializes the printer to the conditions having existed at power on.

Long InitializePrinter();

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

InitializePrinter(strDirBmp, 300, BXL_ALIGNMENT_CENTER, 50);

……
```

3-9 DirectIO

This function can send and read data defined by the user.

```
Long DirectIO (
   PCHAR Data,
   UINT uiWrite,
   PCHAR pRequet,
   UINT uiRead
);
```

[Parameters]

* PCHAR Data
  [in] Data to be sent to the printer, data is ANSI code

* UINT uiWrite
  [in] Size of data to be sent to the printer

* PCHAR pRequet
  [in, out] Sets the values read from the buffered defined by the caller

* UINT uiRead
  [in, out] Converts the size of date to be read by the caller to the actual size of read data

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

char cmd[3] = {0x10, 0x04, 0x02};
char* readData;
UINT readLen = 0;

lResult = DirectIO(cmd, sizeof(cmd), readData, readLen);

……
```

3-10 MsrReadReady

This function switches the printer to MSR Ready status. Printing is not possible when in eady status. After returning via BXL_SUCCESS and if the State property is BXL_STS_MSR_READY, the standby status is deemed normal.

Long CheckPrinter ();

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
     return;

……
```

3-11 MsrReadCancel

This function deactivates the MSR Ready status of the printer.

Long MsrReadCancel();

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_MSR_NOTREADY	-602	Not in READY status

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

MsrReadReady();

……

MsrReadCancel();

……
```

3-12 MsrReadTrack

This function recalls MSR data. If the MSR is in read mode and BXL_MSR_DATAEMPTY is returned, the card has not yet been read in the MSR. Scanning the card in the MSR again can be tried, or the MSRReadCancel function can be used to cancel read mode.

```
Long MsrReadTrack (
   PCHAR Data1,
   PCHAR Data2,
   PCHAR Data3
);
```

[Parameters]

* PCHAR Data1
[in, out] MSR Data Track 1 read from the buffer defined by the caller

* PCHAR Data2
[in, out] MSR Data Track 2 read from the buffer defined by the caller

* PCHAR Data3
[in, out] MSR Data Track 3 read from the buffer defined by the caller

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_MSR_DATAEMPTY	-603	Not in READY status

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
    return;

char Track1[200];
char Track2[200];
char Track3[200];

memset(Track1, 0, sizeof(Track1));
memset(Track2, 0, sizeof(Track2));
memset(Track3, 0, sizeof(Track3));

lResult = MsrReadTrack(Track1, Track2, Track3);

if (lResult != BXL_SUCCESS)
    ……
else
    ……
……
```

3-13 MsrRead1Track

This function returns MSR 1 Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long MsrRead1Track (PCHAR Track, UINT uiLength);

[Parameters]

* PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller

* UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR 1 track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	If sentinel mode set 2, it isn’t support
BXL_MSR_DATAEMPTY	-603	Not in READY status

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
    return;

char Track [200];

memset(Track, 0, sizeof(Track));

lResult = MsrRead1Track(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
    ……
else
    ……
……
```

3-14 MsrRead2Track

This function returns MSR 2 Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long MsrRead2Track (PCHAR Track, UINT uiLength);

[Parameters]

* PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller

* UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR 2 track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	If sentinel mode set 2, it isn't support
BXL_MSR_DATAEMPTY	-603	Not in READY status

*Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
     return;

char Track [200];

memset(Track, 0, sizeof(Track));

lResult = MsrRead2Track(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
     ……
else
     ……
……
```

3-15 MsrRead3Track

This function returns MSR 3 Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long MsrRead3Track (PCHAR Track, UINT uiLength);

[Parameters]

* PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller

* UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR 3 track data will be returned. If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	If sentinel mode set 2, it isn’t support
BXL_MSR_DATAEMPTY	-603	Not in READY status

*Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
    return;

char Track [200];

memset(Track, 0, sizeof(Track));

lResult = MsrRead3Track(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
    ……
else
    ……
……
```

3-16 MsrReadFullTrack

This function returns MSR Full Track data. If the MSR is in read mode and ERROR Code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long MsrReadFullTrack (PCHAR Track, UINT uiLength);

[Parameters]

* PCHAR Track
[in, out] MSR Track data read from the buffer defined by the caller

* UINT uiLength
[in] Size of Track buffer to be sent to the printer

[Return Values]

If the function is successful, MSR Full track data will be returned. Each track data is separated by 0x1c (format : [track1 data]0x1c[track2 data]0x1c[track3 data]0x1c)
If it has failed, the function will return error code and NULL pointer.

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	If sentinel mode set 2, it isn’t support
BXL_MSR_DATAEMPTY	-603	Not in READY status

*Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
    return;

char Track [600];

memset(Track, 0, sizeof(Track));

lResult = MsrReadFullTrack(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
    ……
else
    ……
……
```

3-17 GetTrack1

This function returns MSR Track1 data. If the MSR is in read mode and ERROR code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long GetTrack1 (TCHAR* tcData, UINT dwLen);

[Parameters]

* TCHAR tcData
[in, out] MSR Track1 data will be returned

* UINT dwLen
[in, out] MSR Track1 data length will be returned

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_MSR_DATAEMPTY	-603	Not in READY status

*Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
     return;

TCHAR Track [200];

memset(Track, 0, sizeof(Track));

lResult = GetTrack1(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
     ……
else
     ……
……
```

3-18 GetTrack2

This function returns MSR Track2 data. If the MSR is in read mode and ERROR code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long GetTrack2 (TCHAR* tcData, UINT dwLen);

[Parameters]

* TCHAR tcData
[in, out] MSR Track2 data will be returned

* UINT dwLen
[in, out] MSR Track2 data length will be returned

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_MSR_DATAEMPTY	-603	Not in READY status

*Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
    return;

TCHAR Track [200];

memset(Track, 0, sizeof(Track));

lResult = GetTrack2(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
    ……
else
    ……
……
```

3-19 GetTrack3

This function returns MSR Track3 data. If the MSR is in read mode and ERROR code is returned, the card has not yet been read by the MSR. Strip the card with the MSR again. MSRReadCancel function can be used to cancel read mode.

Long GetTrack3 (TCHAR* tcData, UINT dwLen);

[Parameters]

* TCHAR tcData
[in, out] MSR Track3 data will be returned

* UINT dwLen
[in, out] MSR Track3 data length will be returned

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_MSR_DATAEMPTY	-603	Not in READY status

*Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

MsrReadReady();

if (GetStat() != BXL_STS_MSR_READY)
     return;

TCHAR Track [200];

memset(Track, 0, sizeof(Track));

lResult = GetTrack3(Track, sizeof(Track));

if (lResult != BXL_SUCCESS)
     ……
else
     ……
……
```

3-20 SelectMode

This method selects label or receipt mode.

Long SelectMode(BOOL bLabelMode)

[Parameters]

* BOOL bLabelMode
[in] Use or not use Label Mode
if this value is true, label mode is selected. Otherwise receipt mode is selected.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

// Select Label Mode
if (SelectMode(true) != BXL_SUCCESS)
    return;

NextPrintPos();

// Select Receipt Mode
if (SelectMode(false) != BXL_SUCCESS)
    return;

……
```

3-21 NextPrintPos

If the label function is set, the paper is fed up to the next printing position.

Long NextPrintPos ();

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	Printer is not label mode
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

if (SelectMode(true) != BXL_SUCCESS)
      return;

NextPrintPos();

……
```

3-22 AutoCalibration

This method executes auto calibration in label mode.

Long AutoCalibration()

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

if (SelectMode(true) != BXL_SUCCESS)
    return;

AutoCalibration();

……
```

3-23 SelectPageMode

This command switches standard mode and page mode.

Long SelectPageMode(BOOL bPageMode)

[Parameters]

* BOOL bLabelMode
[in] Use or not use page mode
if this value is ture, page mode is selected. Otherwise standard mode is selected.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

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

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

……
```

3-24 FormFeed

This method prints all data collected in the printer buffer In page mode. After completion of printing, the printer is returned to standard mode.

Long FormFeed(long nFeed)

[Parameters]

* long nFeed
[in] the number of form feed to execute.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

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

……

FormFeed(2);
```

3-25 SetPrintAreaInPM

This function sets the position and the size of the printing area in page mode as following.

Long SetPrintAreaInPM(long x, long y, long width, long height)

[Parameters]

* long x
[in] Horizontal starting position

* long y
[in] Vertical starting position

* long width
[in] Horizontal printing area width

* long height
[in] Vertical printing area width

When paper width of 58mm is selected: x = 0, y = 0, width = 384, height = 840

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_BAD_ARGUMENT	-108	Incorrect argument specified
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

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

SetPrintAreaInPM(0, 0, 416, 416);

……

FormFeed(2);
```

3-26 SetPrintDirectionInPM

This method selects the print direction and starting position in page mode.

Long SetPrintDirectionInPM(long printDirection)

[Parameters]

* long printDirection
[in] This value sets the print position.

printDirection	Value	Print Direction	Starting Position	Rotaion
BXL_PD_LEFT_TO_RIGHT	48	Left -> Right	Upper left	0-degree
BXL_PD_BOTTOM_TO_TOP	49	Bottom -> Top	Lower left	270-degree
BXL_PD_RIGHT_TO_LEFT	50	Right -> Left	Lower right	180-degree
BXL_PD_TOP_TO_BOTTOM	51	Top -> Bottom	Upper right	90-degree

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_BAD_ARGUMENT	-108	Incorrect argument specified
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

……

FormFeed(2);
```

3-27 SetVerticalPositionInPM

This method sets the absolute vertical print starting position to vertical or horizontal motion unit.

Long SetVerticalPositionInPM(long motionUnit)

[Parameters]

* long motionUnit
[in] This value sets the vertical print position.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_BAD_ARGUMENT	-108	Incorrect argument specified
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

SetVerticalPositionInPM(160);
SetHorizontalPositionInPM(40);
PrintText(“Bixolon Mobile Printer.”, 0, BXL_FT_DEFAULT, BXL_TS_0WIDTH |
BXL_TS_0HEIGHT);

……

PrintDataInPageMode();

FormFeed(2);
```

3-28 SetHorizontalPositionInPM

This method specifies the next print starting position in reference to the left edge of the print area.
The printing start position is calculated using vertical or horizontal motion units

Long SetHorizontalPositionInPM(long motionUnit)

[Parameters]

* long motionUnit
[in] This value sets the print position.

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_BAD_ARGUMENT	-108	Incorrect argument specified
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

SetVerticalPositionInPM(160);
SetHorizontalPositionInPM(40);
PrintText(“Bixolon Mobile Printer.”, 0, BXL_FT_DEFAULT, BXL_TS_0WIDTH |
BXL_TS_0HEIGHT);

……

PrintDataInPageMode();

FormFeed(2);
```

3-29 PrintDataInPageMode

This method prints the data in the print buffer collectively in page mode.

Long PrintDataInPageMode()

[Parameters]

None

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_STS_ERROR	32	An error has occurred
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_WRITE_ERROR	-300	Failure to transmit data

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(BXL_PD_LEFT_TO_RIGHT);

SetVerticalPositionInPM(160);
SetHorizontalPositionInPM(40);
PrintText(“Bixolon Mobile Printer.”, 0, BXL_FT_DEFAULT, BXL_TS_0WIDTH |
BXL_TS_0HEIGHT);

……

PrintDataInPageMode();

FormFeed(2);
```

3-30 ScrPowerUp

This function powers up SCR and returns ATR(Answer To Reset) data. This function can only use on the SPP-R210 SCR model.

```
Long ScrPowerUp (
   PCHAR pATR,
   UINT& ATRLen,
   PCHAR ResponseS
);
```

[Parameters]

* PCHAR pATR
  [in, out] ATR(Answer To Reset) data

* UINT& ATRLen
  [in, out] ATR data length

* PCHAR ResponseS
  [in, out] Response of Power UP command

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	Printer does not support SCR
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure
BXL_SCR_RESPONSE_ERR	-700	Return wrong response data

* Example

```
long lResult;
CHAR cAtr[512] = {0,};
UINT lAtrLen = 512;
CHAR cResponseS = 0xff;

lResult = PrinterOpen("portinfo…", 1000);

……

lResult = ScrPowerUp(cAtr, lAtrLen, &cResponseS);

if (lResult != BXL_SUCCESS || cResponseS != 0x00)
    return;

……
```

3-31 ScrPowerDown

This function powers down SCR. This function can only use on the SPP-R210 SCR model.

```
Long ScrPowerDown (
   PCHAR ResponseS
);
```

[Parameters]

* PCHAR ResponseS
  [in, out] Response of Power Down command

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	Printer does not support SCR
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure
BXL_SCR_RESPONSE_ERR	-700	Return wrong response data

* Example

```
long lResult;
CHAR cResponseS = 0xff;

lResult = PrinterOpen(“portinfo…”, 1000);

……

lResult = ScrPowerDown(&cResponseS);

if (lResult != BXL_SUCCESS || cResponseS != 0x00)
     return;

……
```

3-32 ScrOperationMode

This function changes the operating mode. This function can only use on the SPP-R210 SCR model.

```
Long ScrOperationMode (
   UINT mode,
   PCHAR ResponseS
);
```

[Parameters]

* UINT mode
  [in] Operating mode

Code	Value	Description
BXL_SCR_MODE_APDU	0	APDU mode
BXL_SCR_MODE_TPDU	1	TPDU mode

* PCHAR ResponseS
  [in, out] Response of select operating mode command

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	Printer does not support SCR
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure
BXL_SCR_RESPONSE_ERR	-700	Return wrong response data

* Example

```
long lResult;
CHAR cResponseS = 0xff;

lResult = PrinterOpen(“portinfo…”, 1000);

……

lResult = ScrOperationMode(BXL_SCR_MODE_APDU, &cResopnseS);

if (lResult != BXL_SUCCESS || cResponseS != 0x00)
    return;

……
```

3-33 ScrExchangeAPDU

This function exchanges APDU/TPDU. This function can only use on the SPP-R210 SCR model.

```
Long ScrExchangeAPDU (
   PCHAR pCmdAPDU,
   UINT cmdAPDULen,
   PCHAR pRspAPDU,
   UINT& rspAPDULen,
   PCHAR ResponseS
);
```

[Parameters]

* PCHAR pCmdAPDU
  [in] Command APDU data

* UINT cmdAPDULen
  [in] Command APDU length

* PCHAR pRspAPDU
  [in, out] Response APDU data

* UINT& rspAPDULen
  [in, out] Response APDU length

* PCHAR ResponseS
  [in, out] Response of APDU command

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	Printer does not support SCR
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure
BXL_SCR_RESPONSE_ERR	-700	Return wrong response data

* Example

```
long lResult;
CHAR CmdAPDU[512] = {0,};
CHAR RspAPDU[512] = {0,};
UINT cmdLen, rspLen;
CHAR cResponseS = 0xff;

lResult = PrinterOpen(“portinfo…”, 1000);

……

lResult = ScrExchangeAPDU(CmdAPDU, rspLen,
                                   RspAPDU, &cmdLen, &cResopnseS);

if (lResult != BXL_SUCCESS || cResponseS != 0x00)
    return;
……
```

3-34 ScrChkCardStatus

This function checks smart card status. This function can only use on the SPP-R210 SCR model.

```
Long ScrChkCardStatus (
   PCHAR pStatus,
   PCHAR ResponseS
);
```

[Parameters]

* PCHAR pStatus
  [in, out] smart card status data

* PCHAR ResponseS
  [in, out] Response of Power UP command

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_OPENED	-101	Unable to open communication port
BXL_NOT_SUPPORT	-107	Printer does not support SCR
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure
BXL_SCR_RESPONSE_ERR	-700	Return wrong response data

* Example

```
long lResult;
CHAR cStatus = 0x00;
CHAR cResponseS = 0xff;

lResult = PrinterOpen(“portinfo…”, 1000);

……

lResult = ScrChkCardStatus(BXL_SCR_MODE_APDU, &cResopnseS);

if (lResult != BXL_SUCCESS || cResponseS != 0x00)
     return;
……
```

3-35 ScrSelectCard

This function selects smart card, SAM1, SAM2. This function can only use on the SPP-R210 SCR model.

```
Long ScrSelectCard (
   UINT card,
   PCHAR ResponseS
);
```

[Parameters]

* UINT card
  [in] smart card for communication

Code	Value	Description
BXL_SCR_SMARTCARD	48	Select Smart card for communication
BXL_SCR_SAM1	49	Select SAM1 for communication
BXL_SCR_SAM2	50	Select SAM2 for communication

* PCHAR ResponseS
  [in, out] Response of Power UP command

[Return Values]

Code	Value	Description
BXL_SUCCESS	0	Returns after function success
BXL_NOT_SUPPORT	-107	Printer does not support SCR
BXL_WRITE_ERROR	-300	Failure to transmit data
BXL_READ_ERROR	-301	Error in received value or reception failure
BXL_SCR_RESPONSE_ERR	-700	Return wrong response data

* Example

```
long lResult;
CHAR cResponseS = 0xff;

lResult = PrinterOpen(“portinfo…”, 1000);

……

lResult = ScrSelectCard(BXL_SCR_SMARTCARD, &cResopnseS);

if (lResult != BXL_SUCCESS || cResponseS != 0x00)
    return;
……
```

4. Message

If some events have occurred in printer, the SDK will send message through the "RegisterWindowMessage(Win32 API)".

4-1 SetMsrMsgMode (BOOL W)

This function sets MSR Message capacity value of the printer. The default setting is FALSE. Use the SetMsrMsgMode() functions to set values. Defaut value is false.

Capacity values are as follows.

Code	Value	Description
TRUE	1	Enable MSR Message functions
FALSE	0	Disable MSR Message functions

* Example

```
long lResult;

lResult = PrinterOpen("portinfo…", 1000);

……

SetMsrMsgMode(TRUE);

……
```

4-2 BXL_LPARAM _MSR

This message will be posted when the printer reads MSR Track data from MSR sensor.

lParam is set by the following data.

Code	Description
BXL_MSG_TRACK1	0x01
BXL_MSG_TRACK2	0x02
BXL_MSG_TRACK3	0x04

Example) BXL_MSG_TRACK1 + BXL_MSG_TRACK2 = 0x03

wParam = 10

* Example

```
long lResult;

lResult = PrinterOpen(“portinfo…”, 1000);

……

SetMsrMsgMode(TRUE);

……

long lMsrMessage = RegisterWindowsMessage(“_MSR_NOTIFY”);
MSG msg;
int nRetry = 10;

memset(&message, 0, sizeof(message));

while(PeekMessage(&message, NULL, 0, 0, PM_REMOVE) && nRetry--)
{
    If(msg.message == lMsrMessage)
    {
      if (msg.lParam & BXL_MSG_TRACK1)
         // read track1 data

      if (msg.lParam & BXL_MSG_TRACK2)
         // read track2 data

      if (msg.lParam & BXL_MSG_TRACK3)
         // read track3 data

      break;
    }

    Sleep(100);
}
```