

**BIXOLON®**

API 레퍼런스 가이드

# Linux POS/Mobile SDK

---

Rev. 2.02

<http://www.bixolon.co.kr>

# 목차

<b>1. 매뉴얼 안내</b> .....	<b>3</b>
1-1 지원 Kernel & Platform, O/S.....	3
1-2 지원 인터페이스.....	3
1-3 지원 프린터.....	4
<b>2. Property</b> .....	<b>5</b>
2-1 CharacterSet (LONG R/W).....	5
2-2 International CharacterSet (LONG R/W).....	7
2-3 State (LONG R).....	8
<b>3. 메소드</b> .....	<b>9</b>
3-1 ConnectToPrinter.....	9
3-2 DisconnectPrinter.....	10
3-3 InitializePrinter.....	11
3-4 LineFeed.....	12
3-5 SetLeftMargin.....	13
3-6 SetUpsideDown.....	14
3-7 PartialCut.....	15
3-8 OpenCashDrawer.....	16
3-9 PrintText.....	17
3-10 PrintTextW.....	19
3-11 PrintBarcode.....	21
3-12 DirectIO.....	26
3-13 PrintImage.....	27
3-14 DownloadNVImage.....	28
3-15 PrintNVImage.....	29
3-16 RemoveAllNVImage.....	30
3-17 RemoveNVImage.....	31
3-18 GetNVImageKeyCode.....	32
3-19 SetAutoStatusCheck.....	33
3-20 GetStatus.....	34
3-21 SelectMode.....	35
3-22 NextPrintPos.....	36
3-23 AutoCalibration.....	37
3-24 SelectPageMode.....	38
3-25 PrintDataInPM.....	39
3-26 SetPrintAreaInPM.....	40
3-27 SetPrintDirectionInPM.....	41
3-28 SetVerticalPositionInPM.....	42
3-29 SetHorizontalPositionInPM.....	43
3-30 ReadStartMSR.....	44
3-31 ReadCancelMSR.....	45
3-32 ReadMSRData.....	46
3-33 ScrPowerUp.....	47
3-34 ScrPowerDown.....	48
3-35 ScrOperationMode.....	49
3-36 ScrExchangeAPDU.....	50
3-37 ScrCheckStatus.....	51
3-38 ScrSelectCard.....	52
3-39 getBatteryStatus.....	53

# 1. 매뉴얼 안내

이 SDK 매뉴얼에는 Linux OS용 응용프로그램 개발에 필요한 라이브러리에 대한 내용이 기술되어 있습니다.

## 1-1 지원 Kernel & Platform, O/S

- Kernel
  - Kernel 2.6.32 이후 버전
- 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 지원 인터페이스

- USB, Serial, Parallel, Bluetooth, Ethernet, WLAN

## 1-3 지원 프린터

모델	해상도(DPI)	최대 인쇄 너비
SRP-E300	180 dpi	512 dots
SRP-E302	203 dpi	576 dots
SRP-QE300	180 dpi	512 dots
SRP-QE302	203 dpi	576 dots
SRP-S300	203 dpi	576 dots
SRP-380	180 dpi	512 dots
SRP-382	203 dpi	576 dots
SRP-383	300 dpi	864 dots
SRP-330II	180 dpi	512 dots
SRP-332II	203 dpi	576 dots
SRP-F310II	180 dpi	512 dots
SRP-F312II	203 dpi	576 dots
SRP-F313II	203 dpi	640 dots
SRP-350III	180 dpi	512 dots
SRP-352III	203 dpi	576 dots
SRP-350plusIII	180 dpi	512 dots
SRP-352plusIII	203 dpi	576 dots
SRP-340II	180 dpi	512 dots
SRP-342II	203 dpi	576 dots
BK3-3	203 dpi	576 dots
STP-103III	203 dpi	384 dots
SPP-R200II	203 dpi	384 dots
SPP-R200III	203 dpi	384 dots
SPP-R210	203 dpi	384 dots
SPP-R300	203 dpi	576 dots
SPP-R310	203 dpi	576 dots
SPP-R400	203 dpi	832 dots
SPP-R410	203 dpi	832 dots

## 2. Property

제공하는 라이브러리에서 사용하는 상수 값은 `bxlConst.c` 파일에 선언되어 있습니다. 개발환경은 C를 기반으로 하였습니다.

### 2-1 CharacterSet (LONG R/W)

- 프린터의 코드페이지를 정의하는 속성으로 기본 값은 `CS_PC437`으로 설정되어 있습니다. `SetCharSet()`, `GetCharSet()` 함수를 사용하여 값을 설정하거나 설정된 값을 읽어 올 수 있습니다.

사용 가능한 코드페이지는 다음과 같습니다.


상수명	값	설명
<code>CS_PC437</code>	0	Code page PC437
<code>CS_KATAKANA</code>	1	Katakana
<code>CS_PC850</code>	2	Code page PC850
<code>CS_PC860</code>	3	Code page PC860
<code>CS_PC863</code>	4	Code page PC863
<code>CS_PC865</code>	5	Code page PC865
<code>CS_WPC1252</code>	16	Code page WPC1252
<code>CS_PC866</code>	17	Code page PC866
<code>CS_PC852</code>	18	Code page PC852
<code>CS_PC858</code>	19	Code page PC858
<code>CS_THAI42</code>	23	Code page THAI42
<code>CS_WPC1253</code>	24	Code page WPC1253
<code>CS_WPC1254</code>	25	Code page WPC1254
<code>CS_WPC1257</code>	26	Code page WPC1257
<code>CS_WPC1251</code>	28	Code page WPC1251
<code>CS_PC737</code>	29	Code page PC737
<code>CS_PC775</code>	30	Code page PC775
<code>CS_THAI14</code>	31	Code page THAI14
<code>CS_PC862</code>	33	Code page PC862
<code>CS_PC855</code>	36	Code page PC855
<code>CS_PC857</code>	37	Code page PC857
<code>CS_PC928</code>	38	Code page PC928
<code>CS_THAI16</code>	39	Code page THAI16
<code>CS_PC1258</code>	41	Code page PC1258
<code>CS_PC1250</code>	47	Code page PC1250
<code>CS_USER</code>	255	User set page

\* 예제

```
ConnectToPrinter(port);  
.....  
SetCharSet(CS_PC850);  
.....  
int32 nCharSet;  
nCharSet = GetCharSet();  
.....
```

**2-2 International CharacterSet (LONG R/W)**

- 프린터의 International character Set을 정의하는 속성으로 기본 값은 ICS\_USA 으로 설정되어 있습니다. SetInternationalChar(), GetInternationalChar() 함수를 사용하여 값을 설정하거나 설정된 값을 읽어 올 수 있습니다.

 <b>참고</b>	<p>다음의 경우에는 CharacterSet 설정을 확인이 필요할 수 있습니다.</p> <ol style="list-style-type: none"> <li>출력을 시도한 것과 다른 문자열이 출력 될 경우</li> <li>상형문자와 같은 형태로 깨진 문자열이 출력 될 경우</li> <li>문자가 '?'(물음표)와 같은 형태로 출력 될 경우</li> </ol>
---	---

사용 가능한 International character Set은 다음과 같습니다.

상수명	값	설명
ICS_USA	0	USA 코드 설정
ICS_FRANCE	1	FRANCE 코드 설정
ICS_GERMANY	2	GERMANY 코드 설정
ICS_UK	3	UK 코드 설정
ICS_DENMARK1	4	DENMARK1 코드 설정
ICS_SWEDEN	5	SWEDEN 코드 설정
ICS_ITALY	6	ITALY 코드 설정
ICS_SPAIN	7	SPAIN 코드 설정
ICS_JAPAN	8	JAPAN 코드 설정
ICS_NORWAY	9	NORWAY 코드 설정
ICS_DENMARK2	10	DENMARK 2 코드 설정
ICS_SPAIN2	11	SPAIN 2 코드 설정
ICS_LATIN	12	LATIN AMERICA 코드 설정
ICS_KOREA	13	KOREA 코드 설정

\* 예제

```

ConnectToPrinter(port)

.....

SetInternationalChar(ICS_SPAIN);

.....

int32 nCharSet;

nCharSet = GetInternationalChar();

.....
    
```

**2-3 State (LONG R)**

- 프린터 상태가 설정되는 속성입니다. 읽기만 가능하며 **GetStatus()** 함수를 호출하여 프린터 상태를 읽어올 수 있습니다. 상태 값은 중복되어 설정될 수 있으며 비트 연산을 통해 각각의 값을 확인 할 수 있습니다.

프린터 상태 값은 다음과 같습니다.

상수명	값	설명
STS_NORMAL	0	정상 상태
STS_PAPEREMPTY	1	용지 없음
STS_COVEROPEN	2	커버 열림
STS_NEAREND	4	용지 부족

\* 예제

```

ConnectToprinter(port)
.....

SetAutoStatusCheck(true);

int status = GetStatus();

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

.....
    
```



## 3. 메소드

Linux SDK에서 제공하는 함수들은 BxIPosAPI.h 파일에 선언되어 있습니다.  
개발환경은 C를 기반으로 하였습니다.

### 3-1 ConnectToPrinter

- 이 함수는 프린터와 통신을 위해 연결을 설정하는 함수입니다.

```
int ConnectToPrinter(const char *port)
```

#### [매개변수]

- \* const char \*port  
[in] 프린터와 연결할 인터페이스 정보

인터페이스	입력 정보	예시
USB	USB:	ConnectToPrinter("USB:")
Serial	serial:(baudrate) /dev/ttySX:(baudrate)	ConnectToPrinter("serial:115200") ConnectToPrinter("/dev/ttyS0:115200")
Parallel	parallel /dev/lpX	ConnectToPrinter("parallel") ConnectToPrinter("/dev/lp0")
Bluetooth	장치 MAC address /dev/ttySX:(baudrate)	ConnectToPrinter("7d:f0:7d:e4:e0:78") ConnectToPrinter("/dev/ttyS0:115200")
Ethernet, Wifi	IP주소, port번호	ConnectToPrinter("192.168.0.10:9100")

#### [반환 값]

상수명	값	설명
SUCCESS	0	함수 성공
PORT_OPEN_ERROR	-99	포트 열기 실패
NO_CONNECTED_PRINTER	-100	프린터가 연결되어 있지 않음
NO_BIXOLON_PRINTER	-101	미 지원 프린터

\* 예제

```
int ret;

// USB
ret = ConnectToPrinter("USB:");

// Serial
ret = ConnectToPrinter("serial:115200");

// Parallel
ret = ConnectToPrinter("parallel");

// bluetooth
ret = ConnectToPrinter("7d:f0:7d:e4:e0:78");

// Ethernet or WiFi
ret = ConnectToPrinter("192.168.0.10:9100");
```

**3-2 DisconnectPrinter**

- 이 함수는 프린터와 연결을 종료합니다.

void DisconnectPrinter();

**[매개변수]**

없음

**[반환 값]**

없음

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
DisconnectPrinter();
```

**3-3 InitializePrinter**

- 이 함수는 이전에 설정한 내용을 취소 하고, 전원을 켜올 때 상태로 초기화합니다.

int InitializePrinter();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```
.....  
InitializePrinter();  
.....
```

**3-4 LineFeed**

- 이 함수는 인자로 넘어오는 정수 값만큼 라인 피딩을 합니다.

int LineFeed (const unsigned int lineNumber);

**[매개변수]**

\* const unsigned int lineNumber

[in] 정수 값으로 라인 피딩 라인 수를 인자 값으로 전달합니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
MEM_ALLOC_ERROR	-120	내부 메모리 할당 실패

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
LineFeed(5);  
  
.....
```

**3-5 SetLeftMargin**

- 이 함수는 출력 시 좌측 마진 크기를 설정합니다.

int SetLeftMargin (long margin);

**[매개변수]**

- \* long margin  
[in] 마진 크기. 0 ~ 용지 최대 너비 값.(dots)

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	잘못된 인자 값

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
SetLeftMargin(10);  
  
.....
```

**3-6 SetUpsideDown**

- 이 함수는 upside-down 기능을 설정합니다.

int SetUpsideDown (bool upsideDown);

**[매개변수]**

- \* bool upsideDown  
[in] upside-down 기능 설정 여부

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
SetUpsideDown(true);  
  
.....
```

**3-7 PartialCut**

- 이 함수는 partial cut 기능을 실행합니다.

int PartialCut();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	지원하지 않는 기능

\* 예제

```
ConnectToPrinter(.....);  
.....  
PartialCut();  
.....
```

**3-8 OpenCashDrawer**

- 이 함수는 금전통을 엽니다.

int OpenCashDrawer (unsigned int milliSec);

**[매개변수]**

\* unsigned int milliSec  
[in] 옽 신호 길어설정. 0 ~ 255 값을 가짐.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	지원하지 않는 기능

\* 예제

```
ConnectToPrinter(.....);  
.....  
OpenCashDrawer(100);  
.....
```



**3-9 PrintText**

- 이 함수는 텍스트를 인쇄하는 기능을 지원합니다.

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

**[매개변수]**

\* const char\* text  
 [in] null 종결자를 가진 문자열입니다. 인쇄 할 텍스트 데이터를 전달합니다.

\* const int alignment  
 [in] 텍스트의 정렬 설정을 하는 값입니다.

상수명	값	설명
ALIGNMENT_LEFT	0	왼쪽 정렬
ALIGNMENT_CENTER	1	가운데 정렬
ALIGNMENT_RIGHT	2	오른쪽 정렬

\* const unsigned int attribute  
 [in] 텍스트 속성을 설정합니다. 아래 값들은 중복 적용이 가능합니다.

상수명	값	설명
ATTR_FONTTYPE_A	0	Font A로 인쇄 (기본 값)
ATTR_FONTTYPE_B	1	Font B로 인쇄
ATTR_FONTTYPE_C	2	Font C로 인쇄
ATTR_BOLD	4	굵게 인쇄
ATTR_UNDERLINE_1	8	밑줄 추가 (1 도트 굵기)
ATTR_UNDERLINE_2	16	밑줄 추가 (2 도트 굵기)
ATTR_REVERSE	32	역상 인쇄

\* const unsigned int textSize  
 [in] 텍스트 크기를 설정하는 속성을 설정합니다. 가로 배율과 세로 배율 값을 중복으로 사용할 수 있습니다.

상수명	값	설명
TS_WIDTH_0	0x00	가로 너비 배율 x 1
TS_WIDTH_1	0x10	가로 너비 배율 x 2
TS_WIDTH_2	0x20	가로 너비 배율 x 3
TS_WIDTH_3	0x30	가로 너비 배율 x 4
TS_WIDTH_4	0x40	가로 너비 배율 x 5
TS_WIDTH_5	0x50	가로 너비 배율 x 6
TS_WIDTH_6	0x60	가로 너비 배율 x 7
TS_WIDTH_7	0x70	가로 너비 배율 x 8

상수명	값	설명
TS_HEIGHT_0	0x00	세로 높이 배율 x 1
TS_HEIGHT_1	0x01	세로 높이 배율 x 2
TS_HEIGHT_2	0x02	세로 높이 배율 x 3
TS_HEIGHT_3	0x03	세로 높이 배율 x 4
TS_HEIGHT_4	0x04	세로 높이 배율 x 5
TS_HEIGHT_5	0x05	세로 높이 배율 x 6
TS_HEIGHT_6	0x06	세로 높이 배율 x 7
TS_HEIGHT_7	0x07	세로 높이 배율 x 8

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
STATUS_ERROR	-103	인쇄할 수 없음
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```

ConnectToPrinter(.....);

.....

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

.....
    
```

**3-10 PrintTextW**

- 이 함수는 2Bytes 텍스트를 인쇄하는 기능을 지원합니다.

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

**[매개변수]**

\* const char\* text

[in] null 종결자를 가진 문자열입니다. 인쇄 할 텍스트 데이터를 전달합니다.

\* const int alignment

[in] 텍스트의 정렬 설정을 하는 값입니다.

상수명	값	설명
ALIGNMENT_LEFT	0	왼쪽 정렬
ALIGNMENT_CENTER	1	가운데 정렬
ALIGNMENT_RIGHT	2	오른쪽 정렬

\* const unsigned int attribute

[in] 텍스트 속성을 설정합니다. 아래 값들은 중복 적용이 가능합니다.

상수명	값	설명
ATTR_FONTTYPE_A	0	Font A로 인쇄 (기본 값)
ATTR_FONTTYPE_B	1	Font B로 인쇄
ATTR_FONTTYPE_C	2	Font C로 인쇄
ATTR_BOLD	4	굵게 인쇄
ATTR_UNDERLINE_1	8	밑줄 추가 (1 도트 굵기)
ATTR_UNDERLINE_2	16	밑줄 추가 (2 도트 굵기)
ATTR_REVERSE	32	역상 인쇄

\* const unsigned int textSize

[in] 텍스트 크기를 설정하는 속성을 설정합니다. 가로 배율과 세로 배율 값을 중복으로 사용할 수 있습니다.

상수명	값	설명
TS_WIDTH_0	0x00	가로 너비 배율 x 1
TS_WIDTH_1	0x10	가로 너비 배율 x 2
TS_WIDTH_2	0x20	가로 너비 배율 x 3
TS_WIDTH_3	0x30	가로 너비 배율 x 4
TS_WIDTH_4	0x40	가로 너비 배율 x 5
TS_WIDTH_5	0x50	가로 너비 배율 x 6
TS_WIDTH_6	0x60	가로 너비 배율 x 7
TS_WIDTH_7	0x70	가로 너비 배율 x 8

상수명	값	설명
TS_HEIGHT_0	0x00	세로 높이 배율 x 1
TS_HEIGHT_1	0x01	세로 높이 배율 x 2
TS_HEIGHT_2	0x02	세로 높이 배율 x 3
TS_HEIGHT_3	0x03	세로 높이 배율 x 4
TS_HEIGHT_4	0x04	세로 높이 배율 x 5
TS_HEIGHT_5	0x05	세로 높이 배율 x 6
TS_HEIGHT_6	0x06	세로 높이 배율 x 7
TS_HEIGHT_7	0x07	세로 높이 배율 x 8

\* const unsigned int codePage  
 [in] 문자열을 인코딩 방식을 설정합니다.

상수명	값	설명
ENCODING_ASCII	0	ASCII
ENCODING_EUCKR	1	한국어 (EUC-KR)
ENCODING_CP949	2	한국어 (CP949)
ENCODING_EUCCN	3	중국어 (EUC-CN)
ENCODING_GB18030	4	중국어 (GB18030)
ENCODING_BIG5	5	중국어 (BIG5)
ENCODING_CP950	6	중국어 (CP950)
ENCODING_EUCJP	7	일본어 (EUC-JP)
ENCODING_CP932	8	일본어 (CP932)
ENCODING_CP874	9	태국어 (CP874)

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
STATUS_ERROR	-103	인쇄할 수 없음
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```
ConnectToPrinter(.....);

.....

PrintTextW("한국어 출력 테스트.\n", ALIGNMENT_LEFT,
ATTR_FONTTYPE_A, TS_HEIGHT_0 | TS_WIDTH_0, ENCODING_CP949);

.....
```

**3-11 PrintBarcode**

- 이 함수는 1차원 및 2차원 바코드를 인쇄하는 기능을 지원합니다.

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

**[매개변수]**

\* const int barcodeType

[in] 바코드 타입을 정의합니다. 바코드 타입의 bxlConst.c 에 정의 되어 있습니다.

상수명	값	설명
BARCODE_UPCA	0	UPC-A 바코드
BARCODE_UPCE	1	UPC-E 바코드
BARCODE_EAN13	3	JAN-13(EAN-13) 바코드
BARCODE_JAN13	5	
BARCODE_EAN8	2	JAN-8(EAN-8) 바코드
BARCODE_JAN8	4	
BARCODE_ITF	6	ITF 바코드
BARCODE_CODABAR	7	CODABAR 바코드
BARCODE_CODE39	8	CODE39 바코드
BARCODE_CODE93	9	CODE93 바코드
BARCODE_CODE128	10	CODE128 바코드
BARCODE_PDF417	11	PDF417 바코드
BARCODE_QRCODE	12	QR CODE 바코드
BARCODE_DATAMATRIX	13	DATAMATRIX 바코드
BARCODE_MAXICODE	14	MAXICODE 바코드
BARCODE_AZTEC	15	AZTEC 바코드
BARCODE_GS1	16	GS1 바코드

\* const char\* barcodeData

[in] 인쇄 할 바코드 데이터를 전달 합니다.

\* const barcodeInfo\_s\* barcodeInfo

[in] 바코드 관련 속성 정보 저장하는 구조체

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

unsigned int mode

[in] QR Code를 인쇄하는 경우 인쇄 모델값을 전달 합니다.

상수명	값	설명
BARCODE_QR_MODEL1	1	Model 1
BARCODE_QR_MODEL2	2	Model 2

Maxi Code를 인쇄하는 경우 인쇄 모드값을 전달 합니다.

상수명	값	설명
BARCODE_MAXI_MODE2	1	Mode 2
BARCODE_MAXI_MODE3	2	Mode 3
BARCODE_MAXI_MODE4	3	Mode 4

AZTEC을 인쇄하는 경우 인쇄 모드값을 전달 합니다.

상수명	값	설명
BARCODE_AZTEC_DATAMODE	1	Data mode
BARCODE_AZTEC_GS1MODE	2	gs1 mode
BARCODE_AZTEC_UNICODE	3	Unicode mode

GS1을 인쇄하는 경우 인쇄 모드값을 전달 합니다.

상수명	값	설명
BARCODE_GS1_RSS14	1	GS1 DataBar Omnidirectional
BARCODE_GS1_RSS14TRUNCATED	2	GS1 DataBar Truncated
BARCODE_GS1_RSS14STACKED	3	GS1 DataBar Stacked
BARCODE_GS1_RSS14STACKEDOMNI	4	GS1 DataBar Stacked Omnidirectional
BARCODE_GS1_UPCA	5	UPC-A
BARCODE_GS1_UPCE	6	UPC-E
BARCODE_GS1_EAN13	7	EAN-13
BARCODE_GS1_EAN8	8	EAN-8
BARCODE_GS1_EAN128AB	9	UCC/EAN-128&CC-A/B
BARCODE_GS1_EAN128C	10	UCC/EAN-128&CC-C

unsigned int height

[in] 바코드의 높이를 설정하는 값으로 1~255 의 높이 단계를 가지고 있습니다. 바코드의 인쇄 크기가 프린터 용지를 넘어갈 때에 바코드 인쇄가 되지 않을 수 있습니다. 2차원 바코드는 이 값의 영향을 받지 않습니다.

unsigned int width

[in] 바코드의 너비를 설정하는 값으로 2~7 의 넓이 단계를 가지고 있습니다. 바코드의 인쇄 크기가 프린터 용지를 넘어갈 때에 바코드 인쇄가 되지 않을 수 있습니다. 2차원 바코드는 이 값의 영향을 받지 않습니다.

**unsigned int alignment**

[in] 바코드의 정렬 설정을 하는 값입니다.

상수명	값	설명
BXL_ALIGNMENT_LEFT	0	왼쪽 정렬
BXL_ALIGNMENT_CENTER	1	가운데 정렬
BXL_ALIGNMENT_RIGHT	2	오른쪽 정렬

**unsigned int textPosition**

[in] 바코드 데이터를 출력시키는 위치를 설정합니다.

2차원 바코드는 BXL\_BC\_TEXT\_NONE 값만을 가지고 있습니다.

상수명	값	설명
BXL_BC_TEXT_NONE	0	바코드 데이터 미 인쇄
BXL_BC_TEXT_ABOVE	1	바코드 데이터 상단 인쇄
BXL_BC_TEXT_BELOW	2	바코드 데이터 하단 인쇄

**unsigned int attribute**

[in] GS1을 인쇄하는 경우 2D바코드와 1D바코드 구분자의 높이를 설정합니다.

1 또는 2의 값을 가집니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
WRONG_BARCODE_TYPE	-115	지원되지 않는 바코드 유형
WRONG_BC_DATA_ERROR	-116	잘못된 바코드 데이터

**\* 예제**

```

ConnectToPrinter(.....);

char* barcodeData = "123456789012";
barcodeInfo_s barcodeInfo;

.....

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

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

barcodeInfo.height = 50;
    
```

```
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_EAN13, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_JAN13, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_EAN8, "12345678", &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_JAN8, "12345678", &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_CODE39, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_CODE93, barcodeData, &barcodeInfo);

barcodeInfo.height = 50;
barcodeInfo.width = 2;
barcodeInfo.textPosition = BARCODE_TEXT_BELOW;
PrintBarcode(BARCODE_CODE128, barcodeData, &barcodeInfo);

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

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

// ***** 2D barcode

barcodeInfo.height = 0;
barcodeInfo.width = 2;
barcodeInfo.attribute = 0;
PrintBarcode(BARCODE_PDF417, barcodeData, &barcodeInfo);

barcodeInfo.mode = BARCODE_QR_MODEL1;
barcodeInfo.height = 0;
barcodeInfo.width = 2;
barcodeInfo.attribute = 0;
PrintBarcode(BARCODE_QRCODE, barcodeData, &barcodeInfo);
barcodeInfo.mode = BARCODE_QR_MODEL2;
```



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

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

```
barcodeInfo.mode = BARCODE_MAXI_MODE4;  
barcodeInfo.height = 0;  
barcodeInfo.width = 2;  
barcodeInfo.attribute = 0;  
PrintBarcode(BARCODE_MAXICODE, barcodeData, &barcodeInfo);
```

**3-12 DirectIO**

- 이 함수는 사용자 정의 데이터를 보내고 읽을 수 있습니다.

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

**[매개변수]**

- \* char\* writeData,  
[in] 프린터로 전송할 데이터입니다.
- \* int writeLen  
[in] 전송할 데이터 크기입니다.  
writeData에 NULL, writeLen을 0으로 입력하면 write 동작은 하지 않습니다.
- \* char\* readData,  
[in] 프린터로 전송할 데이터 입니다.
- \* int\* readLen  
[in] 호출자가 읽어 들일 데이터 크기를 받습니다.  
readData에 NULL, readLen을 0으로 입력하면 read 동작은 하지 않습니다.
- \* unsigned int mTimeout  
[in] 데이터를 읽기 위해 대기하는 시간입니다. 읽은 데이터가 없어도 해당 시간이 지나면 반환합니다. 0으로 설정하면 데이터가 들어올 때까지 대기합니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
READ_TIMEOUT	-127	데이터를 읽기 전에 타임아웃

\* 예제

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

ConnectToPrinter(.....);

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

.....
```

**3-13 PrintImage**

- 이 함수는 이미지 파일을 인쇄합니다.

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

**[매개변수]**

- \* const char \*imagePath  
[in] 이미지 파일의 전체 경로에 대한 문자열입니다. 지원하는 이미지 타입은 JPG, BMP, GIF 입니다.
- \* const int width  
[in] 이미지 출력 너비를 설정합니다.
- \* const bool compress  
[in] RLE 이미지 압축 여부를 설정합니다.
- \* const unsigned int alignment  
[in] 이미지 정렬을 설정하는 값입니다

상수명	값	설명
ALIGNMENT_LEFT	0	왼쪽 정렬
ALIGNMENT_CENTER	1	가운데 정렬
ALIGNMENT_RIGHT	2	오른쪽 정렬

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
IMAGE_OPEN_ERROR	-118	이미지 파일 열수 없음
MEM_ALLOC_ERROR	-120	내부 메모리 할당 실패

- \* 예제

```
ConnectToPrinter(.....);

.....

PrintImage(filePath, 150, true, ALIGNMENT_CENTER);

.....
```

**3-14 DownloadNVImage**

- 이 함수는 프린터의 비 휘발성 메모리 영역에 이미지를 저장합니다.

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

**[매개변수]**

\* const char \*imagePath

[in] 이미지 파일의 전체 경로에 대한 문자열입니다. 지원하는 이미지 타입은 JPG, BMP, GIF 입니다.

\* const unsigned int keyCode

[in] 이미지를 저장할 메모리 영역 주소. 0 ~ 255 사이의 값으로 설정.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
IMAGE_OPEN_ERROR	-118	이미지 파일 열수 없음
MEM_ALLOC_ERROR	-120	내부 메모리 할당 실패

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
DownloadNVImage(filePath, 0x01);  
  
.....
```

**3-15 PrintNVImage**

- 이 함수는 프린터의 비 휘발성 메모리 영역에 저장되어 있는 이미지를 출력합니다.

int PrintNVImage (const unsigned int keyCode);

**[매개변수]**

\* const unsigned int keyCode

[in] 출력할 이미지 주소 코드. 0 ~ 255 사이의 값으로 설정.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
DownloadNVImage(filePath, 0x01);  
  
.....  
  
PrintNVImage(0x01);
```

**3-16 RemoveAllNVImage**

- 이 함수는 비휘발성 메모리 영역에 저장되어 있는 모든 이미지를 삭제합니다.

int RemoveAllNVImage ();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
.....  
RemoveAllNVImage();
```

**3-17 RemoveNVImage**

- 이 함수는 비휘발성 메모리 영역에 저장되어 있는 해당 주소의 이미지를 삭제합니다.

int RemoveNVImage (const unsigned int keyCode);

**[매개변수]**

\* const unsigned int keyCode

[in] 출력할 이미지 주소 코드. 0 ~ 255 사이의 값으로 설정.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
DownloadNVImage(filePath, 0x01);  
.....  
RemoveNVImage(0x01);
```

**3-18 GetNVImageKeyCode**

- 이 함수는 비휘발성 메모리 영역에 저장되어 이미지의 주소 목록을 읽어옵니다.

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

**[매개변수]**

- \* char \*keyCodeList  
[in, out] 이미지 주소 목록을 저장할 버퍼
- \* unsigned int \*listLen  
[in, out] keyCodeList의 길이

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NV_NO_KEY	-121	정의된 NV key 없음
WRONG_RESPONSE	-122	잘못된 NV data 응답
NOT_SUPPORT	-124	미 지원 함수

\* 예제

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

ConnectToPrinter(.....);

DownloadNVImage(filePath, 0x01);

.....

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



**3-19 SetAutoStatusCheck**

- 이 함수는 프린터 상태(커버오픈, 페이퍼 없음)를 체크를 위해 ASB모드를 설정, 해제 합니다.

int SetAutoStatusCheck(bool enable);

**[매개변수]**

\* bool enable  
[in] ASB 모드를 설정, 해제 합니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```
int status = 0x00;  
ConnectToPrinter(.....);  
SetAutoStatusCheck(true);  
.....  
status = GetStatus();  
.....  
if ((status & BXL_STS_PAPEREMPTY) == BXL_STS_PAPEREMPTY)  
    .....
```

**3-20 GetStatus**

- 이 함수는 프린터 상태(커버오픈, 페이퍼 없음) 정보를 읽어 옵니다.

int GetStatus()

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
STS_NORMAL	0	정상 상태
STS_PAPEREMPTY	1	용지 없음
STS_COVEROPEN	2	커버 열림
STS_NEAREND	4	용지 부족

\* 예제

```
int status = 0x00;

ConnectToPrinter(.....);

SetAutoStatusCheck(true);

.....

status = GetStatus();

.....

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

**3-21 SelectMode**

- 이 함수는 Label/Receipt Mode 를 선택 합니다. 모바일 프린터만 지원합니다.

int SelectMode(bool labelMode);

**[매개변수]**

- \* bool labelMode  
[in] Label Mode의 사용 여부를 전달 합니다.  
이 값이 TRUE 일 경우, 라벨모드가 선택됩니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
// Select Label Mode  
if (SelectMode(true) != SUCCESS)  
    return;  
  
NextPrintPos();  
  
// Select Receipt Mode  
if (SelectMode(false) != SUCCESS)  
    return;  
  
.....
```

**3-22 NextPrintPos**

- 이 함수는 다음 라벨 용지의 시작 지점으로 용지를 피딩합니다. 이 함수는 모바일 프린터가 라벨 모드일 때에만 동작을 합니다.

int NextPrintPos();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
if (SelectMode(true) != SUCCESS)  
    return;  
  
NextPrintPos();  
  
.....
```

**3-23 AutoCalibration**

- 이 함수는 Label Mode 일 때 Auto Calibration 을 수행 합니다.  
이 함수는 모바일 프린터가 라벨 모드일 때에만 동작을 합니다.

int AutoCalibration();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
if (SelectMode(true) != SUCCESS)  
    return;  
  
AutoCalibration();  
  
.....
```

**3-24 SelectPageMode**

- 이 함수는 페이지 모드 사용여부를 선택 합니다.

int SelectPageMode(bool pageMode);

**[매개변수]**

\* bool pageMode

[in] 페이지 모드의 사용 여부

이 값이 TRUE 일 경우, 페이지 모드가 선택됩니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
// Select Page Mode  
if (SelectPageMode(true) != SUCCESS)  
    return;  
  
.....  
  
// Select Standard Mode  
if (SelectPageMode(false) != SUCCESS)  
    return;  
  
.....
```

**3-25 PrintDataInPM**

- 이 함수는 페이지 모드 일 때 프린터 버퍼에 있는 모든 데이터를 출력하고, 프린터는 인쇄가 완료 된 후 표준 모드로 변경 됩니다.

int PrintDataInPM();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```
ConnectToPrinter(.....);  
  
.....  
  
// Select Page Mode  
if (SelectPageMode(true) != SUCCESS)  
    return;  
  
.....  
  
PrintDataInPM();
```

**3-26 SetPrintAreaInPM**

- 이 함수는 페이지 모드 일 때 인쇄 영역의 크기 및 위치를 지정 합니다.

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

**[매개변수]**

- \* long x  
[in] 인쇄 영역의 x 좌표
- \* long y  
[in] 인쇄 영역의 y 좌표
- \* long width  
[in] 인쇄 영역의 가로 크기
- \* long height  
[in] 인쇄 영역의 세로 크기

용지의 너비가 58mm 일 때 설정 : x = 0, y = 0, width = 384, height = 840

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	인자 값이 유효하지 않음
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```

ConnectToPrinter(.....);

.....

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

SetPrintAreaInPM(0, 0, 416, 416);

.....

PrintDataInPM();
    
```



**3-27 SetPrintDirectionInPM**

- 이 함수는 페이지 모드 일 때 출력 방향을 지정 합니다.

int SetPrintDirectionInPM (int printDirection);

**[매개변수]**

\* int printDirection

상수명	값	출력 방향	인쇄 시작 위치	회전
PAGEMODE_ROTATE_0	48	왼쪽 -> 오른쪽	왼쪽 위	0도
PAGEMODE_ROTATE_90	51	위 -> 아래	오른쪽 위	90도
PAGEMODE_ROTATE_180	50	오른쪽 -> 왼쪽	오른쪽 아래	180도
PAGEMODE_ROTATE_270	49	아래 -> 위	왼쪽 아래	270도

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	인자 값이 유효하지 않음
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```

ConnectToPrinter(.....);

.....

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(PAGEMODE_ROTATE_90);

.....

PrintDataInPM();
    
```

**3-28 SetVerticalPositionInPM**

- 이 함수는 페이지 모드 일 때 인쇄 시작위치(가로)를 지정합니다.

int SetVerticalPositionInPM (long position, bool relative);

**[매개변수]**

\* long position

[in] 설정 할 인쇄 시작 위치

\* bool relative

[in] 현재 위치에서의 상대적 위치 인지 절대적 위치인지 설정  
이 값이 TRUE 일 경우, 상대적 위치로 설정됩니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	인자 값이 유효하지 않음
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```

ConnectToPrinter(.....);

.....

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(PAGEMODE_ROTATE_90);

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

.....

PrintDataInPM();
    
```

**3-29 SetHorizontalPositionInPM**

- 이 함수는 인쇄 시작 위치를 지정합니다.

int SetHorizontalPositionInPM (long position, bool relative);

**[매개변수]**

\* long position

[in] 설정 할 인쇄 시작 위치

\* bool relative

[in] 현재 위치에서의 상대적 위치 인지 절대적 위치인지 설정  
이 값이 TRUE 일 경우, 상대적 위치로 설정됩니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	인자 값이 유효하지 않음
NOT_SUPPORT	-124	미 지원 함수

\* 예제

```

ConnectToPrinter(.....);

.....

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

SetPrintAreaInPM(0, 0, 416, 416);
SetPrintDirectionInPM(PAGEMODE_ROTATE_90);

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

.....

PrintDataInPM();
    
```

**3-30 ReadStartMSR**

- 이 함수는 프린터를 MSR Ready 상태로 전환합니다. SUCCESS가 반환되면 정상적인 대기 상태인 것으로 판단합니다. 모바일 프린터만 지원합니다.

long ReadStartMSR();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
NOT_SUPPORT	-124	MSR 기능 미 지원
WRITE_ERROR	-105	데이터 전송 실패

\* 예제

```
int ret;  
  
ConnectToPrinter(.....);  
  
ret = ReadStartMSR();  
  
if (SUCCESS != ret)  
    return;  
  
.....
```

**3-31 ReadCancelMSR**

- 이 함수는 프린터 MSR Ready 상태를 해제 합니다. 모바일 프린터만 지원합니다.

long ReadCancelMSR();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	MSR 기능 미 지원

\* 예제

```
ConnectToPrinter(.....);  
  
ret = ReadStartMSR();  
  
if (SUCCESS != ret)  
    return;  
  
.....  
  
ReadCancelMSR();  
  
.....
```

**3-32 ReadMSRData**

- 이 함수는 MSR 데이터를 읽어옵니다. 데이터를 읽으면 데이터 값을 반환하고 그렇지 않으면 ReadCancelMSR 함수를 사용하여 읽기 모드를 취소 할 수 있습니다. 모바일 프린터만 지원합니다.

```
long ReadMSRData(char *pMSRData1, char *pMSRData2, char *pMSRData3,
                const unsigned int bufLen);
```

**[매개변수]**

- \* char \*pMSRData1  
[out] 호출자가 정의한 버퍼에 MSR Data Track 1 번을 읽어 옵니다.
- \* char \*pMSRData2  
[out] 호출자가 정의한 버퍼에 MSR Data Track 2 번을 읽어 옵니다.
- \* char \*pMSRData3  
[out] 호출자가 정의한 버퍼에 MSR Data Track 3 번을 읽어 옵니다.
- \* const unsigned int bufLen  
[in] pMSRData1, pMSRData2, pMSRData3 buf의 크기입니다.

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
NOT_SUPPORT	-124	MSR 기능 미 지원

\* 예제

```
ConnectToPrinter(.....);
.....

ret = ReadStartMSR();

if (SUCCESS != ret)
    return;

char track1[120] = {0x00, };
char track2[120] = {0x00, };
char track3[120] = {0x00, };

ret = ReadMSRData(track1, track2, track3, sizeof(track1));

if (SUCCESS == ret)
    .....
else
    .....
```

**3-33 ScrPowerUp**

- 이 함수는 SCR의 전원을 On합니다. 이 함수는 SPP-R210 SCR 모델에서만 사용가능 합니다.

long ScrPowerUp(char \*pATR, unsigned int \*ATRLen, char \*pResponse);

**[매개변수]**

- \* char \*pATR  
[in, out] ATR(Answer To Reset) data
- \* unsigned int \*ATRLen  
[in, out] ATR data 길이
- \* char \*pResponse  
[in, out] power up 명령에 대한 응답

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	인자 값이 유효하지 않음
NOT_SUPPORT	-124	미 지원 함수
SCR_RESPONSE_ERROR	-126	잘못된 응답 데이터

\* 예제

```
int ret;
char response = 0xff;
char pATR[512] = {0x00,};
unsigned int atrLen = 512;

ConnectToPrinter(.....);

.....

ret = ScrPowerUp(pATR, &atrLen, &response);

if (0x00 != scrData.ResponseS)
    return;

.....
```

**3-34 ScrPowerDown**

- 이 함수는 SCR의 전원을 off합니다. 이 함수는 SPP-R210 SCR 모델에서만 사용가능 합니다.

long ScrPowerDown(char \*pResponse);

**[매개변수]**

\* char \*pResponse  
[in, out] power up 명령에 대한 응답

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
BAD_ARGUMENT	-117	인자 값이 유효하지 않음
NOT_SUPPORT	-124	미 지원 함수
SCR_RESPONSE_ERROR	-126	잘못된 응답 데이터

\* 예제

```
int ret;  
char response = 0xff;  
  
ConnectToPrinter(.....);  
  
.....  
  
ret = ScrPowerDown(&response);  
  
if (0x00 != response)  
    return;  
  
.....
```



**3-35 ScrOperationMode**

- 이 함수는 동작 모드를 설정합니다. 이 함수는 SPP-R210 SCR 모델에서만 사용가능 합니다.

long ScrOperationMode(uint mode, char \*pResponse);

**[매개변수]**

- \* UINT mode  
[in] Operating mode.

상수명	값	설명
SCR_MODE_APDU	0	APDU mode
SCR_MODE_TPDU	1	TPDU mode

- \* char \*pResponse  
[in, out] power up 명령에 대한 응답

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수
SCR_RESPONSE_ERROR	-126	잘못된 응답 데이터

- \* 예제

```

int ret;
char response = 0xff;

ConnectToPrinter(.....);

.....

ret = ScrOperationMode(SCR_MODE_APDU, &response);

if (0x00 != response)
    return;

.....
    
```

**3-36 ScrExchangeAPDU**

- 이 함수는 APDU/TPDU 데이터 통신을 합니다.  
이 함수는 SPP-R210 SCR 모델에서만 사용가능 합니다.

```
long ScrExchangeAPDU (const char *APDUCmd, unsigned int APDULen,
                    char *APDURsp, unsigned int *APDURspLen, char *pResponse);
```

**[매개변수]**

- \* const char \*APDUCmd  
[in] APDU data 명령
- \* unsigned int APDULen  
[in] APDU 명령 길이
- \* char \*APDURsp  
[in, out] APDU data 응답
- \* unsigned int APDURspLen  
[in, out] APDU 응답 길이
- \* char \*pResponse  
[in, out] power up 명령에 대한 응답

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수
SCR_RESPONSE_ERROR	-126	잘못된 응답 데이터

\* 예제

```
int ret;
char response = 0xff;
char cmdAPDU[512] = {0x00, };
char rspAPDU[512] = {0x00, };
unsigned int cmdLen, rspLen;

ConnectToPrinter(.....);

.....

scrData = ScrExchangeAPDU(cmdAPDU, cmdLen, rspAPDU, rspLen, &response);

if (0x00 != response)
    return;

.....
```

**3-37 ScrCheckStatus**

- 이 함수는 스마트카드 상태를 체크합니다.  
이 함수는 SPP-R210 SCR 모델에서만 사용가능 합니다.

long ScrCheckStatus (char \*status, char \*pResponse);

**[매개변수]**

\* char \*status  
[in, out] 스마트카드 상태 데이터

\* char \*pResponse  
[in, out] 명령에 대한 응답

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수
SCR_RESPONSE_ERROR	-126	잘못된 응답 데이터

\* 예제

```
int ret;
char response = 0xff;
char status = 0x00;

ConnectToPrinter(.....);

.....

ret = ScrCheckStatus(&status, &response);

if (0x00 != response)
    return;

.....
```

**3-38 ScrSelectCard**

- 이 함수는 Smart card, SAM1, SAM2 통신 선택합니다.  
이 함수는 SPP-R210 SCR 모델에서만 사용가능 합니다.

long ScrSelectCard (uint card, char \*pResponse);

**[매개변수]**

- \* UINT card  
[in] 통신할 스마트 카드

상수명	값	설명
SCR_SMARTCARD	48	스마트 카드(Smart card)와 통신 설정
SCR_SAM1	49	SAM1와 통신 설정
SCR_SAM2	50	SAM2와 통신 설정

- \* char \*pResponse  
[in, out] 명령에 대한 응답

**[반환 값]**

상수명	값	설명
SUCCESS	0	함수 성공
WRITE_ERROR	-105	데이터 전송 실패
NOT_SUPPORT	-124	미 지원 함수
SCR_RESPONSE_ERROR	-126	잘못된 응답 데이터

- \* 예제

```
int ret;
char response = 0xff;

ConnectToPrinter(.....);

.....

ret = ScrSelectCard(SCR_SMARTCARD, &response);

if (0x00 != response)
    return;

.....
```

**3-39 getBatteryStatus**

- 이 함수는 SRP-Q300/SRP-Q302 배터리 상태(FULL, HIGH, MIDDLE, LOW)를 체크합니다.

int getBatteryStatus();

**[매개변수]**

없음

**[반환 값]**

상수명	값	설명
SUCCESS	0 to 3	함수 성공 0 : Battery Full 1 : Battery High 2 : Battery Middle 3 : Battery Low
READ_TIMEOUT	-1	응답 없음

\* 예제

```
int status = 0x00;

ConnectToPrinter(.....);

.....

status = getBatteryStatus();

.....
```

## 저작권

© BIXOLON Co., Ltd. 모든 권한을 소유합니다.

이 사용설명서와 제품에 사용된 저작물은 저작권법에 의해 보호되어 있습니다.

(주)빅솔론의 사전 서면 동의 없이 사용 설명서 및 제품에 사용된 저작물에 대한 일부 또는 전체를 무단으로 복제, 저장, 전송하는 것을 금합니다.

제공된 정보는 본 제품에만 해당되며 다른 제품에 대해서는 적용되지 않습니다.

또한 본 정보 사용으로 인해 발생하는 직/간접적 손해에 대해 책임지지 않습니다.

- 빅솔론 로고는 (주)빅솔론의 등록상표입니다.
- 모든 다른 상표 또는 제품 이름은 해당하는 회사 또는 조직의 상표입니다.

(주)빅솔론은 제품의 기능과 품질 향상을 위하여 지속적인 개선을 하고 있습니다.

이로 인하여 제품의 사양과 매뉴얼의 내용은 사전 통보 없이 변경될 수 있습니다.

## 사용시 주의사항

프린터와 같은 전자 제품은 정전기에 의해 쉽게 훼손될 수 있습니다.

정전기로부터 프린터를 보호하기 위해서는 프린터 후면 부에 케이블을 연결하거나 제거하기 전에 반드시 프린터 전원을 끄십시오.

만약 프린터가 정전기로부터 손상을 입었을 경우에는 가까운 구입처에 문의하십시오.

