

# Woosim Windows CE SDK Reference

---

Version 4.1

November 2020



## Contents

---

<b>1. OVERVIEW.....</b>	<b>3</b>
1.1. INTRODUCTION.....	3
1.2. PRINTING DATA BUFFER.....	3
1.3. PRINTER MODE.....	4
1.4. DEPRECATED APIS.....	4
1.5. DEFINITIONS AND ABBREVIATIONS.....	4
<b>2. API LIST.....</b>	<b>5</b>
2.1. DEVICE CONNECTION CONTROL.....	5
2.2. DIRECT TRANSFER TO DEVICE.....	5
2.3. PRINTING DATA BUFFER HANDLING.....	5
2.4. APPEND COMMAND TO PRINTING DATA BUFFER.....	5
2.4.1. <i>General</i> .....	5
2.4.2. <i>Page Mode</i> .....	6
2.4.3. <i>Image</i> .....	6
2.4.4. <i>Barcode</i> .....	6
2.5. DEPRECATED APIS.....	7
<b>3. API REFERENCE.....</b>	<b>8</b>
3.1. DEVICE CONNECTION CONTROL.....	8
3.2. DIRECT TRANSFER TO DEVICE.....	9
3.3. PRINTING DATA BUFFER HANDLING.....	11
3.4. APPEND COMMAND TO PRINTING DATA BUFFER.....	11
3.4.1. <i>General</i> .....	11
3.4.2. <i>Page Mode</i> .....	16
3.4.3. <i>Image</i> .....	19
3.4.4. <i>Barcode</i> .....	20
<b>4. SAMPLE PROJECTS.....</b>	<b>27</b>
4.1. C++ PROGRAM.....	27
4.2. C# PROGRAM.....	27

# 1. Overview

Copyright © 2020 Woosim System Inc.

## 1.1. Introduction

본 문서는 우심 프린터(이하 프린터)를 사용하는 Windows Mobile 응용 프로그램 개발을 위해 사용될 수 있는 Woosim Windows CE SDK(이하 SDK)에 대한 정보를 제공합니다.

SDK에 포함된 Library(이하 Library)는 MFC 기반으로 개발된 DLL 파일입니다. Library API는 사용자의 개발 언어에 따라 다른 방식으로 참조될 수 있으며, SDK에 포함된 예제 프로젝트를 통해 자세한 방법을 확인할 수 있습니다.

본 문서는 MFC 프로젝트 코드 기반으로 작성되었습니다.

Library와 예제 프로젝트는 MS Visual Studio 2008 환경에서 개발되었습니다.

우심 프린터는 모델에 따라 여러 종류의 MCU가 장착되어 있습니다. 프린터의 MCU는 self-test 기능을 통해 확인할 수 있습니다. 프린터 <FEED> 버튼을 누른 상태로 전원을 켜면 self-test 기능이 동작합니다.

SDK는 M16C, ARM, RX 버전의 MCU를 지원합니다. M37702 MCU는 제한적으로 지원됩니다. 모든 API는 RX MCU 기반으로 작성되었습니다. M16C, ARM MCU에서 동작하지 않는 API도 일부 존재합니다.

대부분의 Library API는 프린터 command 또는 프린터 command 모음을 의미합니다. 자주 사용되는 프린터 command를 Library API로 제공하고 있지만 모든 command를 포함하고 있지는 않습니다. Library API 형태로 제공되지 않는 command를 사용하려면 "Woosim Command Manual" 문서를 참고하시기 바랍니다.

## 1.2. Printing Data Buffer

Library에서 제공하는 API 동작은 크게 3가지로 분류할 수 있습니다.

- 1) 프린터와 연결 제어
- 2) Printing Data Buffer와 이를 이용하는 command 처리
- 3) Printing Data Buffer와 무관한 command 처리

Printing Data Buffer는 Library 내부에 프린터로 전송될 데이터를 저장해 두는 400KB 공간입니다. 프린터 동작에 관한 API가 호출되면 대응되는 프린터 command가 Printing Data Buffer에 저장됩니다. Printing Data Buffer에 저장된 내용은 특정 API가 호출되면 모두 한 번에 프린터로 전송됩니다.

하지만 일부 API는 Printing Data Buffer를 사용하지 않고 호출되는 즉시 프린터로 해당 command를 전송합니다. 프로그램 작성 시 Printing Data Buffer를 사용하는 API와 Printing Data Buffer를 사용하지 않는 API를 구별하는 것이 필요합니다.

### 1.3. Printer Mode

프린터는 Standard mode와 Page mode 2가지 모드를 제공합니다.

Standard mode에서는 일반적인 텍스트 경우처럼 프린터로 전달되는 데이터를 즉각적으로 인쇄합니다.

Page mode에서는 프린터로 전달되는 데이터를 지정된 영역 안에 인쇄할 내용을 표시만 해두고 있다가 특정 command가 들어오면 표시해 두었던 내용을 한 번에 인쇄합니다. Library API 중 Page mode에서만 유효한 API는 "Page\_"로 시작됩니다.

### 1.4. Deprecated APIs

Library 갱신 과정에서 신규 버전에서 더 이상 공식적으로 지원하지 않는 API가 생길 수 있습니다. 이 경우 가급적 신규 API를 적용하여 프로그램 작성하는 것을 권장합니다. 하지만 기존 프로그램 코드를 유지해야 하는 상황인 경우 기존 API를 Woosim\_Printer\_LIB.h 파일에 추가로 선언함으로써 기존 API를 유지할 수 있습니다.

### 1.5. Definitions and Abbreviations

API	Application Interface Unit
BPP	Bits Per Pixel
DBCS	Double Byte Character System
DLL	Dynamic Link Library
ECC	Error Correction Code
HRI	Human Readable Interpretation
MCU	Main Control Unit
MFC	Microsoft Foundation Class
MSR	Magnetic Stripe Reader
SCR	Smart Card Reader
SDK	Software Development Kit
TTF	True Type Font

---

## 2. API List

---

### 2.1. Device Connection Control

`BOOL` ClosePrinterConnection()  
`int` ConnectSerialPrinter(TCHAR \*portName, `int` baudRate, `int` timeoutMsec, `BOOL` bProtocol)  
`int` ConnectWirelessPrinter(TCHAR \*ipAddress, `int` port, `int` timeoutMsec, `BOOL` bProtocol)

### 2.2. Direct Transfer to Device

`void` CancelMSRMode()  
`void` CancelSCRMode()  
`void` EnterMSRMode(`int` n)  
`void` EnterSCRMode()  
`void` GetFirmwareVersion()  
`void` GetPrinterModelName()  
`int` GetPrinterStatus(`int` timeoutMsec)  
`int` GetPrinterStatusEx(`int` timeoutMsec)

### 2.3. Printing Data Buffer Handling

`void` ClearSpool()  
`int` PrintSpool(`BOOL` bDelete)

### 2.4. Append Command to Printing Data Buffer

#### 2.4.1. General

`int` ControlCommand(`BYTE` \*data, `int` length)  
`void` CutPaper(`int` mode)  
`void` FeedToMark()  
`void` InitLineSpace()  
`void` InitPrinterStatus()  
`void` PrintData()  
`void` PrintDotFeed(`int` dots)  
`void` PrintLineFeed(`int` lines)  
`int` PrintSpoolForTTF(TCHAR\* data, `BYTE` fontWidth, `BYTE` fontHeight)  
`void` SetAbsPosition(`int` distance)  
`void` SetCharCodeTable(`int` n, `int` mcu)  
`void` SetCharSpace(`int` n)  
`void` SetFontForTTF(TCHAR \*ttfFile)

```

void    SetFontSize(int n)
void    SetLineSpace(int n)
void    SetPositionFromMark(int distance)
void    SetTextAlignment(int n)
void    SetTextStyle(int underline, BOOL bold, int width, int height, BOOL reverse)
void    SetUpsideDown(BOOL set)
void    TextSaveSpool(TCHAR *text)

```

### 2.4.2. Page Mode

```

void    InitPageMode(int x, int y, int width, int height)
void    Page_ClearCurrentData()
void    Page_DrawBox(int x, int y, int width, int height, int thickness)
void    Page_DrawEllipse(int x, int y, int radiusW, int radiusH, int thickness)
void    Page_DrawLine(int x1, int y1, int x2, int y2, int thickness)
void    Page_DotFeed(int dots)
void    Page_LineFeed(int lines)
void    Page_Newline()
void    Page_Print()
void    Page_Print_StandardMode()
void    Page_SetArea(int x, int y, int width, int height)
void    Page_SetDirection(int n)
void    Page_SetPosition(int x, int y)
void    Page_SetStandardMode()
void    SetPageMode()

```

### 2.4.3. Image

```

void    CompressedBmpSaveSpool(TCHAR* bmpFilePath)
void    LoadLogoSaveSpool(int n)
void    NormalBmpSaveSpool(TCHAR* bmpFilePath)

```

### 2.4.4. Barcode

```

void    DataMatrixSaveSpool(int width, int height, int module, TCHAR *barcodeData)
void    GS1DatabarSaveSpool(int type, int n, TCHAR *barcodeData)
void    MaxicodeSaveSpool(int mode, TCHAR *barcodeData)
void    MicroPDF417SaveSpool(int width, int column, int row, int ratio,
    TCHAR *barcodeData, BOOL bHri)
void    OneDimensionBarcodeSaveSpool(BYTE barcodeType, int width, int height,
    BOOL bHri, TCHAR *barcodeData)
void    PDF417SaveSpool(int width, int column, int level, int ratio,
    TCHAR *barcodeData, BOOL bHri)
void    QRCodeSaveSpool(int version, TCHAR level, int module, TCHAR *barcodeData)

```

`void` TruncatedPDF417SaveSpool(`int` width, `int` column, `int` level, `int` ratio, TCHAR \*barcodeData, `BOOL` bHri)

## 2.5. Deprecated APIs

`int` BarcodeSaveSpool(`BYTE` barcodeType, `BYTE` width, `BYTE` height, `BOOL` bHri, TCHAR\* barcodeData, `int` prtWidth, `int` prtHeight, `int` xPos)

`int` BmpSaveSpool(TCHAR\* bmpFilePath, `int` x, `int` y)

`void` CardCancel()

`void` CardRead(`BOOL` bTrack2 = `TRUE`)

`void` CardRead\_M(`BOOL` bTrack2 = `TRUE`)

`BOOL` ClosePrinter()

`int` InitPrinter(TCHAR \*portName, `int` baudRate = 57600, `BOOL` bProtocol = `TRUE`)

`int` InitWlanPrinter(TCHAR \*ipAddress, `int` port, `int` timeoutMsec = 5000, `BOOL` bProtocol = `TRUE`)

`int` SaveSpool(TCHAR \*text, `int` extension = 0, `BOOL` bold = `FALSE`)

`void` SetCharCodeTableEX(`int` n)

## 3. API Reference

### 3.1. Device Connection Control

프린터와 연결하거나 연결을 종료할 때 사용합니다. Serial, Wi-Fi 연결을 지원합니다.

```
int ConnectSerialPrinter(TCHAR *portName, int baudRate = 57600, int timeoutMsec = 1000,
                        BOOL bProtocol = FALSE)
```

시리얼 통신 방식으로 프린터와 연결합니다.

Parameters

<i>pPortName</i>	프린터와 연결된 시리얼 포트 (COM1, COM2, 등)
<i>baudRate</i>	프린터 시리얼 통신 속도. 프린터 self-test를 통해 확인할 수 있습니다.
<i>timeoutMsec</i>	연결을 시도하는 최대 시간 (msec 단위)
<i>bProtocol</i>	일반적인 경우 FALSE 유지합니다.

Returns

SUCCESS (1)	연결 성공
ALREADY_OPENED (-1)	프린터가 이미 연결된 상태
UNABLE_TO_OPEN_THE_PORT (-2)	포트 사용 불가
UNABLE_TO_CONFIGURE_THE_SERIAL_PORT (-3)	포트 초기화 실패
UNABLE_TO_SET_THE_TIMEOUT_PARAMETERS (-4)	포트 설정 실패
TIMEOUT (-7)	연결시도 시간 초과

```
int ConnectWirelessPrinter(TCHAR *ipAddress, int port, int timeoutMsec = 5000,
                          BOOL bProtocol = FALSE)
```

Wi-Fi 통신 방식으로 프린터와 연결합니다.

Parameters

<i>pIPAddress</i>	프린터에 할당된 IP address
<i>port</i>	프린터 port number
<i>timeoutMsec</i>	연결을 시도하는 최대 시간 (msec 단위)
<i>bProtocol</i>	일반적인 경우 FALSE 유지합니다.

Returns

SUCCESS (1)	연결 성공
ALREADY_OPENED (-1)	프린터가 이미 연결된 상태
SOCKET_ERROR (-2)	Socket 초기화 실패

CONNECT\_FAIL (-3)

연결 실패

TIMEOUT (-7)

연결시도 시간 초과

**BOOL ClosePrinterConnection()**

프린터와 연결을 종료합니다.

Returns

항상 TRUE

### 3.2. Direct Transfer to Device

Library에서 제공하는 대부분의 API는 Printing Data Buffer에 저장되었다가 한 번에 프린터로 전송됩니다. 하지만 여기서 소개하는 API들은 Printing Data Buffer를 사용하지 않고 호출되는 즉시 프린터에 command를 전송합니다.

**void EnterMSRMode(int n)**

MSR 모드로 들어갑니다.

MSR 카드는 3개의 트랙으로 구성되는데 MSR에 따라 읽을 수 있는 트랙에 차이가 있습니다. 읽을 수 있는 트랙에 따라 MSR은 12 Track, 23 Track, 123 Track으로 나뉘며 선택된 모드에 따라 읽는 트랙이 결정됩니다.

n	12 Track MSR	23 Track MSR	123 Track MSR
0	1 Track	2 Track	1 Track
1	2 Track	3 Track	2 Track
2	1,2 Track	2,3 Track	1,2 Track
3	N.A	N.A	1,2,3 Track
4	N.A	N.A	3 Track

MSR 카드 읽기에 성공하면 MSR 모드는 자동으로 종료됩니다.

MSR을 통해 읽은 각 트랙의 정보를 프린터에서 어떤 형태로 작성해서 보내는지는 Woosim Command Manual의 "Magnetic Card Data Output Format" 내용을 참조하시기 바랍니다.

Parameters

*n*                      읽을 대상이 되는 MSR 트랙 (0 ~ 4)

**void CancelMSRMode()**

MSR 카드 읽기 없이 MSR 모드를 종료합니다.

**void EnterSCRMode()**

SCR 모드로 들어갑니다.

**void CancelSCRMode()**

SCR 모드를 종료합니다.



NOT\_OPEN\_THE\_PORT (-11)

프린터 연결 오류

### 3.3. Printing Data Buffer Handling

**int PrintSpool(BOOL bDelete = TRUE)**

Printing Data Buffer 내용을 프린터로 전송합니다.

Parameters

*bDelete* TRUE : Printing Data Buffer 내용 삭제  
 FALSE : Printing Data Buffer 내용 유지

Returns

SUCCESS (1) 전송 성공  
 NOT\_OPEN\_THE\_PORT (-11) 프린터 연결 오류

**void ClearSpool()**

Printing Data Buffer 내용을 삭제합니다.

### 3.4. Append Command to Printing Data Buffer

Printing Data Buffer에 command와 데이터를 순차적으로 저장합니다. 저장된 내용은 PrintSpool() API에 의해 프린터로 전송되거나 ClearSpool() API에 의해 삭제될 수 있습니다.

#### 3.4.1. General

일반적으로 Standard mode에서 사용되지만 일부는 Page mode에서도 동작합니다. 다만 Standard mode 상황과 Page mode 상황에서 다르게 동작하는 경우도 있습니다.

**int ControlCommand(BYTE \*data, int length)**

byte stream 내용을 Printing Data Buffer에 추가합니다.

Parameters

*data* Printing Data Buffer에 추가할 byte stream  
*length* Printing Data Buffer에 추가할 byte stream 길이

Returns

전달된 byte stream 내용 추가 후 Printing Data Buffer에 저장된 데이터 크기

**void TextSaveSpool(TCHAR \*text)**

텍스트 내용을 Printing Data Buffer에 추가합니다.

Parameters

*text* Printing Data Buffer에 추가할 문구

**void InitPrinterStatus()**

프린터 설정 일부를 초기화하고, 프린터 내부 저장 공간에 있는 데이터를 삭제하는 command를 Printing Data Buffer에 추가합니다.

**void PrintData()**

Standard mode에서 데이터를 인쇄하는 command를 Printing Data Buffer에 추가합니다. Page mode 상황에서는 줄 바꿈으로 동작합니다.

**void PrintDotFeed(int dots)**

Standard mode에서 데이터를 인쇄하고 용지를 배출하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*dots*                      배출하는 용지의 dot 단위 길이 (0 ~ 255)

**void PrintLineFeed(int lines)**

Standard mode에서 데이터를 인쇄하고 용지를 배출하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*lines*                      배출하는 용지의 줄 단위 길이 (0 ~ 255)

**void SetCharCodeTable(int n, int mcu)**

Code table 설정하는 command를 Printing Data Buffer에 추가합니다. MCU에 따라서 선택할 수 있는 code table에 차이가 있습니다.

Parameters

*n*                              Code table (하기 표 참조)

*mcu*                            프린터 MCU ID (M16C: 0, ARM: 1, RX: 2)

M16C, ARM MCU

<b>n</b>	<b>Character Code Table</b>
<b>0</b>	CP437 (USA, Standard Europe)
<b>1</b>	Katakana
<b>2</b>	Multilingual CP850
<b>3</b>	Portuguese CP860
<b>4</b>	ISO8859-15 (Latin9)
<b>5</b>	Polish
<b>255</b>	DBCS (Double Byte Character System) KSC5601, Shift-JIS, BIG5, GB2312 중 하나

RX MCU

<b>n</b>	<b>Character Code Table</b>
----------	-----------------------------

0	USA, Standard Europe [CP437]
1	Katakana
2	Multilingual(Latin-1) [CP850]
3	Portuguese [CP860]
4	Canadian-French [CP863]
5	Nordic [CP865]
6	Slavic(Latin-2) [CP852]
7	Turkish [CP857]
8	Greek [CP737]
9	Russian(Cyrillic) [CP866]
10	Hebrew [CP862]
11	Baltic [CP775]
12	Polish
13	Latin-9 [ISO8859-15]
14	Latin1[Win1252]
15	Multilingual Latin I + Euro[CP858]
16	Russian(Cyrillic)[CP855]
17	Russian(Cyrillic)[Win1251]
18	Central Europe[Win1250]
19	Greek[Win1253]
20	Turkish[Win1254]
21	Hebrew[Win1255]
22	Vietnam[Win1258]
23	Baltic[Win1257]
24	Azerbaijani
30	Thai[CP874]
40	Arabic [CP720]
41	Arabic [Win 1256]
42	Arabic (Farsi)
43	Arabic presentation forms B
50	Hindi Devanagari
255	DBCS (Double Byte Character System) KSC5601, Shift-JIS, BIG5, GB18030 중 하나

**void SetFontSize(int n)**

Font 크기 설정하는 command를 Printing Data Buffer에 추가합니다.

MCU와 선택된 code table에 따라서 선택할 수 있는 font 크기에 차이가 있습니다. 기본적인 크기는 아래 표와 같습니다.

n	RX	M16C, ARM
0	12x24	12x24
1	9x24	9x24
2	8x16	N.A

Thai font는 12x24 크기만 지원합니다.

Arabic과 Hindi font는 n 값이 0일 때 16x24 크기만 지원합니다.

DBCS font는 n 값이 0일 때 24x24 크기만 지원합니다.

Parameters

*n* Font 크기 (0 ~ 2)

**void SetTextAlignment(int n)**

문자 정렬방식을 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*n* 정렬 방식 (좌측: 0, 가운데: 1, 우측: 2)

**void SetTextStyle(int underline, BOOL bold, int width, int height, BOOL reverse)**

문자 속성을 설정하는 command를 Printing Data Buffer에 추가합니다.

설정할 수 있는 속성은 밑줄, 굵은체, 확대, 반전입니다.

Parameters

*underline* 밑줄 굽기 (0 ~ 2)  
*bold* 굵은체 설정 여부  
*width* 수평방향 문자 확대 (0 ~ 7)  
*height* 수직방향 문자 확대 (0 ~ 7)  
*reverse* 반전 설정 여부

**void InitLineSpace()**

문자열 줄간 간격을 초기화하는 command를 Printing Data Buffer에 추가합니다.

초기 값은 30 dot입니다.

**void SetLineSpace(int n)**

문자열 줄간 간격을 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*n* Dot 단위 문자열 줄간 간격 (0 ~ 255)

**void SetAbsPosition(int distance)**

인쇄 시작점을 좌측 기준점으로부터 우측으로 이동될 길이 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*distance*                    Dot 단위 이동 길이

**void SetCharSpace(int n)**

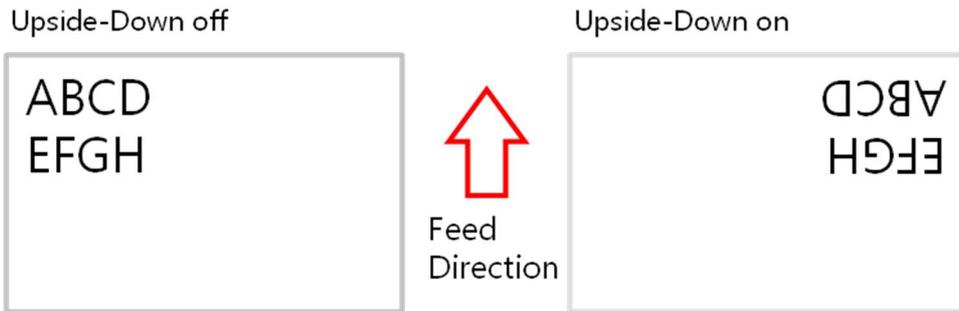
문자 자간 간격을 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*n*                                Dot 단위 문자 자간 간격 (0 ~ 255)

**void SetUpsideDown(BOOL set)**

Upside down 설정 또는 종료하는 command를 Printing Data Buffer에 추가합니다.



Parameters

*set*                                Upside down 설정 여부

**void SetFontForTTF(TCHAR \*ttfFile)**

사용할 TTF 파일을 설정하는 command를 Printing Data Buffer에 추가합니다.

TTF 파일은 프린터에 저장되어 있어야 합니다. 프린터에 TTF 파일을 추가로 저장하려면 별도의 프로그램이 필요하니 당사 지원부서에 문의해 주시기 바랍니다.

Parameters

*ttfFile*                        TTF 파일명

**int PrintSpoolForTTF(TCHAR\* data, BYTE fontWidth, BYTE fontHeight)**

TTF 파일을 이용해 문자열을 인쇄하는 command를 Printing Data Buffer에 추가합니다.

TTF 파일은 앞서 SetFontForTTF()로 선택되어야 합니다.

Parameters

*data*                                문자열 데이터

*fontWidth*                        TTF 문자의 point 단위 가로 크기 (4 ~ 255)

*fontHeight*                        TTF 문자의 point 단위 세로 크기 (4 ~ 255)

Returns

Printing Data Buffer 추가에 성공하면 1, 실패하면 0

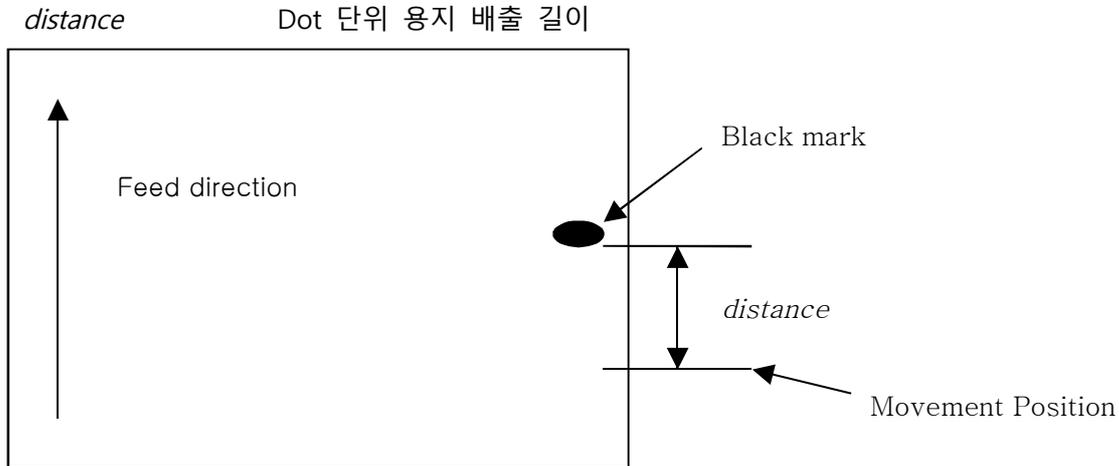
**void SetPositionFromMark(int distance)**

Black mark 감지한 지점으로부터 용지 배출 길이 설정하는 command를 Printing Data Buffer에 추가합니다.

용지 배출 길이는 dot 단위이며, 프린터 flash memory 공간에 저장됩니다. FeedToMark()에 의한 command가 동작할 때마다 센서에 감지된 위치에서 지정된 길이만큼 용지를 추가로 배출합니다.

이 command는 용지 배출 길이 수정이 필요할 때 1회만 사용합니다.

Parameters



**void FeedToMark()**

Black mark 감지한 지점까지 용지 배출하는 command를 Printing Data Buffer에 추가합니다. 실제로는 앞서 SetPositionFromMark()에 의해 지정된 길이만큼 추가로 용지가 배출됩니다. 프린터가 Mark 센서를 사용하도록 설정되어 있을 때만 동작합니다.

**void CutPaper(int mode)**

용지를 절단하는 command를 Printing Data Buffer에 추가합니다. 절단기가 장착된 프린터에서만 동작합니다. 전체 절단과 부분 절단을 선택할 수 있습니다.

Parameters

*mode*                      전체절단: 0, 부분절단: 1

**3.4.2. Page Mode**

**void SetPageMode()**

Standard mode에서 Page mode로 전환하는 command를 Printing Data Buffer에 추가합니다.

**void InitPageMode(int x, int y, int width, int height)**

Standard mode에서 Page mode로 전환하고 인쇄 영역을 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*x*                              Page mode 인쇄 영역 시작점의 x 좌표 (dot 단위)  
*y*                              Page mode 인쇄 영역 시작점의 y 좌표 (dot 단위)  
*width*                        Page mode 인쇄 영역 너비 (dot 단위)  
*height*                        Page mode 인쇄 영역 높이 (dot 단위, 최대값 2400)

**void Page\_DrawLine(int x1, int y1, int x2, int y2, int thickness)**

두 점을 연결하는 직선을 그리는 command를 Printing Data Buffer에 추가합니다.

Parameters

<i>x1</i>	Page mode 인쇄 영역 내 한 지점의 x 좌표 (dot 단위)
<i>y1</i>	Page mode 인쇄 영역 내 한 지점의 y 좌표 (dot 단위)
<i>x2</i>	Page mode 인쇄 영역 내 다른 지점의 x 좌표 (dot 단위)
<i>y2</i>	Page mode 인쇄 영역 내 다른 지점의 y 좌표 (dot 단위)
<i>thickness</i>	직선의 두께 (dot 단위)

**void Page\_DrawBox(int x, int y, int width, int height, int thickness)**

사각형을 그리는 command를 Printing Data Buffer에 추가합니다.

Parameters

<i>x</i>	사각형 시작점의 x 좌표 (dot 단위)
<i>y</i>	사각형 시작점의 y 좌표 (dot 단위)
<i>width</i>	사각형 너비 (dot 단위)
<i>height</i>	사각형 높이 (dot 단위)
<i>thickness</i>	사각형 테두리의 두께 (dot 단위)

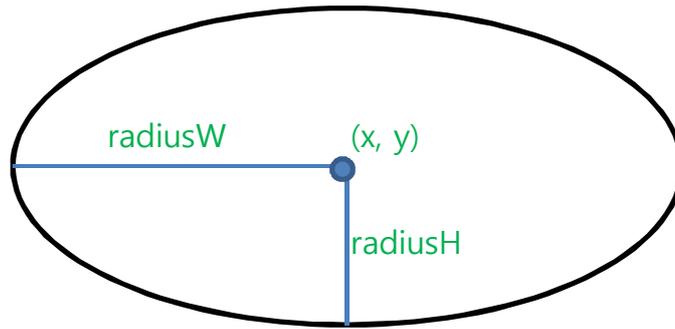


**void Page\_DrawEllipse(int x, int y, int radiusW, int radiusH, int thickness)**

타원형을 그리는 command를 Printing Data Buffer에 추가합니다.

Parameters

<i>x</i>	타원형 중심점의 x 좌표 (dot 단위)
<i>y</i>	타원형 중심점의 y 좌표 (dot 단위)
<i>radiusW</i>	타원형 가로방향 반지름 (dot 단위)
<i>radiusH</i>	타원형 세로방향 반지름 (dot 단위)
<i>thickness</i>	타원형 테두리의 두께 (dot 단위)



**void Page\_DotFeed(int dots)**

Page mode 인쇄영역 내에서 위치 이동하는 command를 Printing Data Buffer에 추가합니다. Dot 단위로 이동하며, 아래 방향으로 이동된 줄의 시작점이 새로운 인자 위치가 됩니다.

Parameters

*dots* 이동하는 dot 단위 길이 (0 ~ 255)

**void Page\_LineFeed(int lines)**

Page mode 인쇄영역 내에서 위치 이동하는 command를 Printing Data Buffer에 추가합니다. 줄 단위로 이동하며, 아래 방향으로 이동된 줄의 시작점이 새로운 인자 위치가 됩니다.

Parameters

*lines* 이동하는 줄 단위 길이 (0 ~ 255)

**void Page\_Newline()**

Page mode 인쇄영역 내에서 위치 이동하는 command를 Printing Data Buffer에 추가합니다. 한 줄 아래 방향으로 이동된 줄의 시작점이 새로운 인자 위치가 됩니다.

**void Page\_ClearCurrentData()**

Page mode에서 작성된 내용을 삭제하는 command를 Printing Data Buffer에 추가합니다.

**void Page\_SetArea(int x, int y, int width, int height)**

Page mode 인쇄 영역을 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*x* Page mode 인쇄 영역 시작점의 x 좌표 (dot 단위)

*y* Page mode 인쇄 영역 시작점의 y 좌표 (dot 단위)

*width* Page mode 인쇄 영역 너비 (dot 단위)

*height* Page mode 인쇄 영역 높이 (dot 단위, 최대값 2400)

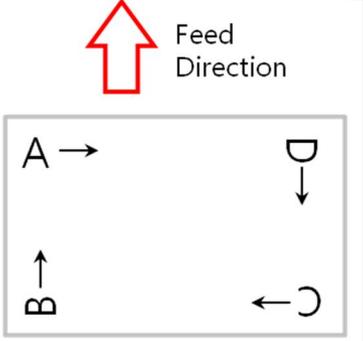
**void Page\_SetDirection(int n)**

Page mode 인쇄 영역 내에서 출력 위치와 방향을 설정하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*n* 인쇄 방향 설정 (0 ~ 3)

<i>n</i>	시작점	인쇄 방향
0	좌상단 (A)	좌측 → 우측
1	좌하단 (B)	아래 → 위
2	우하단 (C)	우측 → 좌측
3	우상단 (D)	위 → 아래



**void Page\_SetPosition(int x, int y)**

Page mode 인쇄 영역 내에서 인자 위치를 이동하는 command를 Printing Data Buffer에 추가합니다.

Parameters

- x*                      Page mode 인쇄 영역 내의 x 좌표 (dot 단위)
- y*                      Page mode 인쇄 영역 내의 y 좌표 (dot 단위)

**void Page\_Print()**

Page mode에서 작성된 내용을 인쇄하는 command를 Printing Data Buffer에 추가합니다. 출력 후에도 Page mode 내에 작성된 내용은 삭제되지 않고 남아 있습니다. 출력 후 작성한 내용을 지우려면 Page\_ClearCurrentData()을 이용합니다.

**void Page\_SetStandardMode()**

Page mode에서 Standard mode로 전환하는 command를 Printing Data Buffer에 추가합니다.

**void Page\_Print\_StandardMode()**

Page mode에서 작성된 내용을 인쇄하고 Page mode에서 Standard mode로 전환하는 command를 Printing Data Buffer에 추가합니다.

**3.4.3. Image**

**void NormalBmpSaveSpool(TCHAR\* bmpFilePath)**

이미지 파일을 인쇄하는 command를 Printing Data Buffer에 추가합니다. 1 bit per pixel 또는 24 bits per pixel BMP 파일을 지원합니다.

Parameters

- bmpFilePath*            인쇄할 파일 경로

**void CompressedBmpSaveSpool(TCHAR\* bmpFilePath)**

이미지 파일을 인쇄하는 command를 Printing Data Buffer에 추가합니다. BMP, JPG, PNG 파일을 지원합니다. 이미지 데이터를 압축해서 전송되는 크기를 줄입니다.

Parameters

- bmpFilePath*            인쇄할 파일 경로



73	CODE128	$2 \leq n \leq 255$	$0 \leq d \leq 127$ d=C1H (FNC1) d=C2H (FNC2) d=C3H (FNC3) d=C4H (FNC4)
----	---------	---------------------	---

```
void PDF417SaveSpool(int width, int column, int level, int ratio, TCHAR *barcodeData,
                    BOOL bHri)
```

PDF417 2차원 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

- width*                    바코드 데이터 너비 (1 ~ 8)
- column*                    바코드 열의 수 (1 ~ 30)
- level*                    바코드 이미지 손상 시 복구 수준 (0 ~ 8)
- ratio*                    가로 세로 비율 (2 ~ 5)
- barcodeData*            바코드 데이터.
- bHri*                    HRI 문자 인쇄 여부

```
void DataMatrixSaveSpool(int height, int width, int module, TCHAR *barcodeData)
```

Data Matrix 2차원 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

- height*                    바코드 높이 (0 : 자동 크기 조절)
- width*                    바코드 너비 (0 : 자동 크기 조절)
- module*                    바코드 모듈 크기 (1 ~ 8)
- barcodeData*            바코드 데이터.

Symbol size		Capacity (bytes)			*ECC(%)	Remark
Row	Column	Numeric	Alpha-numeric	Byte (8bit)		
10	10	6	3	3	62.5	
12	12	10	6	5	58.3	
8	18	10	6	5	58.3	Rectangular
14	14	16	9	8	55.6	
8	32	20	12	10	52.4	Rectangular
16	16	24	15	12	50.0	
12	26	32	21	16	46.7	Rectangular
18	18	36	24	18	43.8	
20	20	44	30	22	45.0	
12	36	44	30	22	45.0	Rectangular

22	22	60	24	30	40.0	
16	36	34	45	32	42.9	Rectangular
24	24	72	51	36	40.0	
26	26	88	63	44	38.9	
16	48	98	72	49	36.4	Rectangular
32	32	124	90	62	36.7	
36	36	172	126	86	32.8	
40	40	228	168	114	29.6	
44	44	288	213	144	28.0	
48	48	348	258	174	28.1	
52	52	408	303	204	29.2	
64	64	560	417	280	28.6	
72	72	736	549	368	28.1	
80	80	912	681	456	29.6	
88	88	1152	861	576	28.0	
96	96	1392	1041	696	28.1	
104	104	1632	1221	816	29.2	
120	120	2100	1572	1050	28.0	
132	132	2608	1953	1304	27.6	
144	144	3116	2334	1558	28.5	

\* ECC: Error Correction Code rate

`void QRCodeSaveSpool(int version, char level, int module, TCHAR *barcodeData)`

QR-Code 2차원 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

*version*                      바코드 크기 (1 ~ 40, 0 : 자동 크기 조절)

*level*                         EC 수준 (L: 7%, M: 15%, Q: 25%, H: 30%)

*module*                      바코드 모듈 크기 (1 ~ 8)

*barcodeData*                바코드 데이터.

Version	Capacity (Code words) by EC level			
	L ( 7% )	M ( 15% )	Q ( 25% )	H ( 30% )
1	19	16	13	9
2	34	28	22	16
3	55	44	34	26
4	80	64	48	36

5	108	86	62	46
6	136	108	76	60
7	156	124	88	66
8	194	154	110	86
9	232	182	132	100
10	274	216	154	122
11	324	254	180	140
12	370	290	206	158
13	428	334	244	180
14	461	365	261	197
15	523	415	295	223
16	589	453	325	253
17	647	507	367	283
18	721	563	397	313
19	795	627	445	341
20	861	669	485	385
21	932	714	512	406
22	1006	782	568	442
23	1094	860	614	464
24	1174	914	664	514
25	1276	1000	718	538
26	1370	1062	754	596
27	1468	1128	808	628
28	1531	1193	871	661
29	1631	1267	911	701
30	1735	1373	985	745
31	1843	1455	1033	793
32	1955	1541	1115	845
33	2071	1631	1171	901
34	2191	1725	1231	961
35	2306	1812	1286	986
36	2434	1914	1354	1054
37	2566	1992	1426	1096
38	2702	2102	1502	1142
39	2812	2216	1582	1222

40	2956	2334	1666	1276
----	------	------	------	------

```
void MicroPDF417SaveSpool(int width, int column, int row, int ratio,
                          TCHAR *barcodeData, BOOL bHri)
```

Micro PDF417 2차원 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

- width*                바코드 데이터 너비 (1 ~ 8)
- column*             바코드 열의 수 (1 ~ 4)
- row*                 바코드 행의 수 (4 ~ 44, 0 : 자동 크기 조절)
- ratio*                가로 세로 비율 (2 ~ 5)
- barcodeData*        바코드 데이터.
- bHri*                 HRI 문자 인쇄 여부

Columns	Rows	Max Data Bytes	Max Alpha Characters	Max Digits
1	11	3	6	8
1	14	7	12	17
1	17	10	18	26
1	20	13	22	32
1	24	18	30	44
1	28	22	38	55
2	8	8	14	20
2	11	14	24	35
2	14	21	36	52
2	17	27	46	67
2	40	33	56	82
2	46	38	64	93
2	52	43	72	105
3	6	6	10	14
3	8	10	18	26
3	10	15	26	38
3	12	20	34	49
3	15	27	46	67
3	20	39	66	96
3	26	54	90	132
3	32	68	114	167

3	38	82	138	202
3	44	97	162	237
4	4	8	14	20
4	6	13	22	32
4	8	20	34	49
4	10	27	46	67
4	12	34	58	85
4	15	45	76	111
4	20	63	106	155
4	26	85	142	208
4	32	106	178	261
4	38	128	214	313
4	44	150	250	366

**void TruncatedPDF417SaveSpool(int width, int column, int level, int ratio, TCHAR \*barcodeData, BOOL bHri)**

Truncated PDF417 2차원 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

- width*                    바코드 데이터 너비 (1 ~ 8)
- column*                    바코드 열의 수 (1 ~ 4)
- level*                    바코드 이미지 손상 시 복구 수준 (0 ~ 8)
- ratio*                    가로 세로 비율 (2 ~ 5)
- barcodeData*            바코드 데이터.
- bHri*                    HRI 문자 인쇄 여부

**void MaxicodeSaveSpool(int mode, TCHAR \*barcodeData)**

Maxicode 2차원 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

- mode*                    바코드 모드 (2 ~ 6)
- barcodeData*            바코드 데이터.

**void GS1DatabarSaveSpool(int type, int n, TCHAR \*barcodeData)**

GS1 Databar 바코드 인쇄하는 command를 Printing Data Buffer에 추가합니다.

Parameters

<i>type</i>	<p>바코드 타입 (0 ~ 6)</p> <p>0: GS1 Databar Omnidirectional</p> <p>1: GS1 Databar Truncated</p> <p>2: GS1 Databar Stacked</p> <p>3: GS1 Databar Stacked Omnidirectional</p> <p>4: GS1 Databar Limited</p> <p>5: GS1 Databar Expanded</p> <p>6: GS1 Databar Expanded Stacked</p>
<i>n</i>	<p>각 행의 segment 수 (2 ~ 20, 짝수)</p> <p>바코드 타입 6인 경우에만 적용</p>
<i>barcodeData</i>	<p>바코드 데이터.</p> <p>타입 0~4: 13개 이내의 숫자</p> <p>타입 5~6: GS1 General Specifications 데이터 표준 준수. ( ) 대신 [ ] 사용.</p>

## 4. Sample Projects

SDK에는 Library 사용에 참고할 수 있는 예제 프로젝트가 포함되어 있습니다. 모든 예제 프로젝트는 Windows Mobile 6 Professional platform SDK를 사용했습니다.

예제 프로젝트는 각각 Visual C++, Visual C# 언어로 작성되었으며, Visual C++ 프로젝트는 MFC 기반으로, Visual C# 프로젝트는 .NET Compact Framework 3.5 기반으로 개발되었습니다.

모든 예제 프로젝트는 동일한 UI 구성으로 작성되었으며, 다음 기능을 포함합니다.

- Serial, Wi-Fi 연결
- 텍스트, 이미지, 바코드 인쇄
- Page mode 인쇄
- MSR 모듈로 카드 읽기

예제 프로젝트 폴더에는 실행 파일도 포함되어 있습니다. 실행 파일을 통해 프린터를 동작시키려면 Self-Test를 통해 프린터 정보를 확인해야 합니다.

프린터로부터 전송되는 데이터는 DLL이 수신하여 다른 프로그램에게 데이터 수신을 알립니다. 프로그램은 프린터가 전송하는 데이터를 전달받기 위해 `RegisterWindowMessage()`를 사용합니다. 이 때 파라미터로 `WOOSIM_PRINTER_LIB.h` 파일에 정의된 문구를 사용합니다.

```
UINT UWM_RECEIVE_DATA = RegisterWindowMessage("WOOSIM_PRT_OK");
```

이후 `UWM_RECEIVE_DATA`를 프로그램 내의 windows message loop에서 식별하여 프린터가 전송한 데이터를 전달 받습니다. 구체적인 내용은 예제 프로젝트를 참고하시기 바랍니다.

### 4.1. C++ Program

Library를 이용해서 C/C++ 프로그램을 개발하는 경우 SDK에 포함된 아래 두 파일을 프로젝트 소스 폴더에 복사합니다.

- `WOOSIM_PRINTER_LIB.h`
- `WoosimPrinter.lib`

생성된 실행 파일과 같은 폴더 또는 참조 가능한 폴더에 SDK에 포함된 DLL 파일을 복사해서 사용합니다.

### 4.2. C# Program

`WOOSIM_PRINTER_LIB.h` 파일을 직접 참조할 수 없기 때문에 `DllImport` 구문을 통해 Library API를 참조합니다.

```
[DllImport("WoosimPrinter.DLL")]
```

```
public static extern bool ClosePrinterConnection();
```

예제 프로젝트 Form1.cs 파일에 작성된 내용을 참고하시기 바랍니다.