

Woosim iOS SDK Programmer Reference

Version 2.1.1
September 2015



Contents

1. OVERVIEW.....	3
1.1. PURPOSE.....	3
1.2. GET STARTED.....	3
1.3. DEVELOPMENT ENVIRONMENT.....	3
1.4. DEFINITIONS AND ABBREVIATIONS.....	4
2. WSENCODER CLASS.....	5
2.1. TASKS.....	5
2.2. CONSTANTS.....	7
2.3. INSTANCE METHODS.....	10
2.3.1. <i>Barcode</i>	10
2.3.2. <i>Page mode</i>	18
2.3.3. <i>Standard mode</i>	24
2.3.4. <i>Text style</i>	27
2.3.5. <i>True type font</i>	32
2.3.6. <i>MSR</i>	32
2.3.7. <i>Smart card</i>	34
2.3.8. <i>Label paper</i>	34
2.3.9. <i>Miscellaneous</i>	35
2.4. DEPRECATED METHOD.....	36
3. WSPARSER CLASS.....	37
3.1. TASKS.....	37
3.2. CONSTANTS.....	37
3.3. INSTANCE VARIABLE.....	37
3.4. INSTANCE METHODS.....	37
4. SAMPLE CODES.....	39
4.1. BTPRINT.....	39
4.2. WiFiPRINT.....	39

1. Overview

1.1. Purpose

This document provides required information for iOS application developers using Woosim printers. The SDK is the API made based on the commands to be used frequently by users. If you want to use other functions that are not described in the document, please refer to the Woosim command manual.

1.2. Get Started

Woosim iOS Software Development Kit (SDK) package includes several sample projects, and you can use them to apply to your workspace.

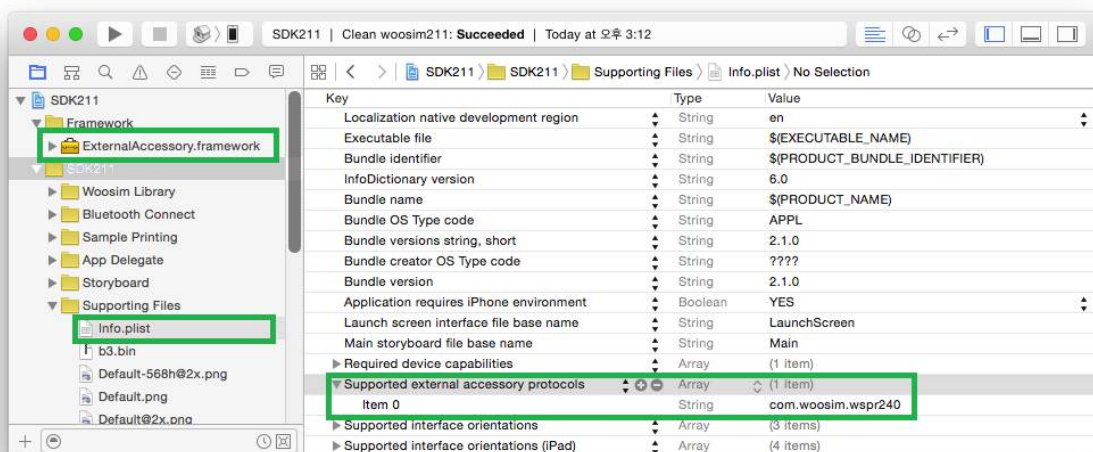
The file name of static library is libwoosim nn .a. It is included in each sample project.

You can easily create new projects by using provided samples.

1.3. Development Environment

Platform OS	OS X 10.10.5
Tool	Xcode 7.0
Base SDK	iOS 9.0
Deployment Target	iOS 6.1

In the file, Xcode Info.plist, ExternalAccessory.framework and com.woosim.wspr240 are added to Supported external accessory protocols as below.



1.4. Definitions and Abbreviations

DBCS	Double Bytes Character Sets
HRI	Human Readable Interpretation
MCU	Main Control Unit
MSR	Magnetic Stripe Reader
SDK	Software Development Kit

2. WSEncoder Class

The WSEncoder class is used for making data to send Woosim printer.

Each instance method makes printable data or control command for Woosim printer, then it would be sent to the printer by user application.

It inherits from NSObject.

2.1. Tasks

Barcode

- createBarcode: barcodeType:
- createBarcodeDataMatrix: symbolHeight: symbolWidth: moduleSize:
- createBarcodeDataMatrix: dataEncoding: symbolHeight: symbolWidth: moduleSize:
- createBarcodeMaxicode: mode:
- createBarcodeMicroPDF417: column: row: HVRatio:
- createBarcodeMicroPDF417: dataEncoding: column: row: HVRatio:
- createBarcodePDF417: column: securityLevel: HVRatio:
- createBarcodePDF417: dataEncoding: column: securityLevel: HVRatio:
- createBarcodeQRCode: symbolVersion: ECLevel: moduleSize:
- createBarcodeQRCode: dataEncoding: symbolVersion: ECLevel: moduleSize:
- createBarcodeTruncPDF417: column: securityLevel: HVRatio:
- createBarcodeTruncPDF417: dataEncoding: column: securityLevel: HVRatio:
- createGS1Databar: type: rowSegment:
- enableHRI:
- setBarcodeHeight:
- setBarcodeWidth:

Page mode

- clearDataInPageMode
- createDrawingAreaWithStartPositionX: withStartPositionY: withAreaWidth: withAreaHeight:
- createImageInPageMode: withStartPositionX: withStartPositionY:
- drawBoxWithWidth: withHeight: withLineThickness:
- drawDiagonalLineWithStartPositionX: withStartPositionY: withEndPositionX: withEndPositionY:
withThickness:
- drawEllipseWithCenterPositionX: withCenterPositionY: withRadiusX: withRadiusY: withThickness:
- drawHorizontalLineWithLength: withLineThickness:
- drawVerticalLineWithLength: withLineThickness:
- enterPageMode
- exitPageMode
- feedLineInPageMode

- feedNDotInPageMode:
- feedNLineInPageMode:
- movePositionToX: toY:
- printDataInPageMode
- setPrintingDirectionInPageMode:

Standard mode

- createImageInStandardMode: withShiftPosition:
- printAndFeedNDotInStandardMode:
- printAndFeedNLineInStandardMode:
- printDataInStandardMode
- printImageInStandardMode
- setLeftMarginInStandardMode:
- setPrintableAreaWidthInStandardMode:
- setPrintingUpsideDownInStandardMode:
- setTextAlignInStandardMode:

Text style

- addString: encoding:
- addString: encoding: withTextWidth: withTextHeight: withTextBold
- resetLineSpacing
- reverseTextColor:
- selectTextCodeTable:
- setFontSize:
- setLetterSpacing:
- setLineSpacing:
- setTextBold:
- setTextSizeWithWidth: withHeight:
- setTextUnderline: withLineThickness:
- shiftAbsolutePosition:
- shiftRelativePosition::

True type font

- addStringWithTrueTypeFont: withStringWidth: withStringHeight:
- selectTrueTypeFontFile:

MSR

- enterMSR1stTrackMode
- enterMSR2ndTrackMode
- enterMSR3rdTrackMode

- enterMSRDoubleTrackMode
- enterMSRTripleTrackMode
- exitMSRMode

Smart Card

- enterSmartCardReaderMode
- exitSmartCardReaderMode

Label paper

- feedToBlackMarkPosition
- setBlackMarkPosition:

Miscellaneous

- changeDotFromCentimeter:
- clearPrinterBuffer
- cutPapaer:
- selectLogoImage:
- verifyPrinterVer

Deprecated

- createPrintableDataFromImage: coordinateX: coordinateY: width: height:

2.2. Constants

Barcode types used by *createBarcode:* method.

```
typedef enum {
    BARCODE_UPCA = 65,
    BARCODE_UPCE,
    BARCODE_EAN13,
    BARCODE_EAN8,
    BARCODE_CODE39,
    BARCODE_ITF,
    BARCODE_CODABAR,
    BARCODE_CODE93,
    BARCODE_CODE128
} BARCODE
```

Text width used by *setTextSizeWithWidth: withHeight:* and *addString: encoding: withTextWidth: withTextHeight: withTextBold:* methods.

```
typedef enum {
    TEXTWIDTH_1 = 0x00,
```

```

TEXTWIDTH_2,
TEXTWIDTH_3,
TEXTWIDTH_4,
TEXTWIDTH_5,
TEXTWIDTH_6,
TEXTWIDTH_7,
TEXTWIDTH_8
} TEXTWIDTH;

```

Text height types used by *setTextSizeWithWidth: withHeight:* and *addString: encoding: withTextWidth: withTextHeight: withTextBold:* methods.

```

typedef enum {
    TEXTHEIGHT_1 = 0x00,
    TEXTHEIGHT_2 = 0x10,
    TEXTHEIGHT_3 = 0x20,
    TEXTHEIGHT_4 = 0x30,
    TEXTHEIGHT_5 = 0x40,
    TEXTHEIGHT_6 = 0x50,
    TEXTHEIGHT_7 = 0x60,
    TEXTHEIGHT_8 = 0x70
} TEXTHEIGHT;

```

Font size used by *setFontSize:* method.

```

typedef enum {
    FONTSIZE_A = 0x00,
    FONTSIZE_B,
    FONTSIZE_C
} FONTSIZE;

```

Directions used by *setPrintingDirectionInPageMode:* method.

```

typedef enum{
    DIRECTION_LEFTTORIGHT = 0x00,
    DIRECTION_BOTTOMTOTOP,
    DIRECTION_RIGHTTOLEFT,
    DIRECTION_TOPTOBOTTOM
} DIRECTION;

```

Alignments used by *setTextAlignInStandardMode:* method.

```

typedef enum{
    ALIGN_LEFT = 0x00,

```

```

    ALIGN_CENTER,
    ALIGN_RIGHT
} ALIGN;

```

Cutting types used by *cutPaper*: method.

```

typedef enum {
    CUT_FULL = 0x00,
    CUT_PARTIAL
} CUT;

```

Languages used by *selectTextCodeTable*: method.

```

typedef enum {
    LANGUAGE_CP437 = 0,
    LANGUAGE_CP850 = 2,
    LANGUAGE_CP860 = 3,
    LANGUAGE_CP852 = 6,
    LANGUAGE_CP857 = 7,
    LANGUAGE_CP737 = 8,
    LANGUAGE_CP866 = 9,
    LANGUAGE_CP775 = 11,
    LANGUAGE_ISO8859_15 = 13,
    LANGUAGE_WIN1252 = 14,
    LANGUAGE_WIN1251 = 17,
    LANGUAGE_WIN1250 = 18,
    LANGUAGE_WIN1253 = 19,
    LANGUAGE_WIN1254 = 20,
    LANGUAGE_WIN1255 = 21,
    LANGUAGE_WIN1258 = 22,
    LANGUAGE_WIN1257 = 23,
    LANGUAGE_CP874 = 30,
    LANGUAGE_WIN1256 = 41,
    LANGUAGE_EUC_KR = 255,
    LANGUAGE_SHIFT_JIS = 255,
    LANGUAGE_GB18030 = 255,
    LANGUAGE_BIG5 = 255
} LANGUAGE

```

2.3. Instance Methods

All methods in WSEncoder class have return value that is a pointer of NSData stream command sent to the Woosim printer. If the return value is nil, it means that it uses a wrong parameter for the specific API or uses improper mode between the standard mode and the page mode. The default value is the standard mode.

2.3.1. Barcode

- (NSData *)**createBarcode**:(NSString *)data **barcodeType**:(BARCODE)type

Create printable 1D barcode data along with barcode type parameter.

Supported barcodes are defined as BARCODE type.

Parameters

data The barcode source data.
Data length and value is dependent on barcode type.

type The barcode type

parameter <i>type</i>	Barcode system	Data length	Data source
BARCODE_UPCA	UPCA	$11 \leq n \leq 12$	$48 \leq d \leq 57$
BARCODE_UPCE	UPCE	$11 \leq n \leq 12$	$48 \leq d \leq 57$
BARCODE_EAN13	EAN13	$11 \leq n \leq 13$	$48 \leq d \leq 57$
BARCODE_EAN8	EAN8	$7 \leq n \leq 8$	$48 \leq d \leq 57$
BARCODE_CODE39	CODE39	$1 \leq n \leq 255$	$48 \leq d \leq 57, 65 \leq d \leq 90,$ $d = 36, 43, 45, 46, 47, 58$
BARCODE_ITF	ITF	$1 \leq n \leq 255$ (even number)	$48 \leq d \leq 57$
BARCODE_CODABAR	CODABAR	$1 \leq n \leq 255$	$48 \leq d \leq 57, 65 \leq d \leq 68,$ $d = 36, 43, 45, 46, 47, 58$
BARCODE_CODE93	CODE93	$1 \leq n \leq 255$	$0 \leq d \leq 127$
BARCODE_CODE128	CODE128	$2 \leq n \leq 255$	$0 \leq d \leq 127$

Table 1 Barcode type

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**createBarcodePDF417**:(NSString *)data **column**:(uint8_t)column
securityLevel:(uint8_t)level **HVRatio**:(uint8_t)ratio

Create PDF417 2D barcode data. It is influenced by *enableHRI*: and *setBarcodeWidth*:. (Only ASCII code can be printed.)

Parameters

- data* The barcode source data
- column* The column number of 2D barcode (1 ~ 30)
- level* The security level to restore when barcode image is damaged (0 ~ 8)
- ratio* The horizontal and vertical ratio (2 ~ 5)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**createBarcodePDF417**:(NSString *)*data* **dataEncoding**:(NSStringEncoding)*encoding* **column**:(uint8_t)*column* **securityLevel**:(uint8_t)*level* **HVRatio**:(uint8_t)*ratio*

createBarcodePDF417: *column*: *securityLevel*: *HVRatio*: Same with API but other characters including ASCII can be printed .

For NSStringEncoding,TextVeiwcontroller/multiLanguageButton can be referred on the sample program.

- (NSData *)**createBarcodeDataMatrix**:(NSString *)*data* **symbolHeight**:(uint8_t)*height* **symbolWidth**:(uint8_t)*width* **moduleSize**:(uint8_t)*size*

Create Data Matrix (ECC200) 2D barcode data. The created barcode shape is square type if *height* and *width* are 0. (Only ASCII code can be printed.)

Parameters

- data* The barcode source data
- height* The height of the symbol (0 : auto size)
- width* The width of the symbol (0 : auto size)
- size* The module size (1 ~ 8)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

Symbol-size		Data capacity(bytes)			ECC(%) Error Correction code	Remark
Row (height)	Column (width)	Numeric	Alpha-numeric	Byte(8bit)		
10	10	6	3	3	62.5	1
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	42	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	

* Used only square type for auto-sized symbol.

Table 2 Data Matrix (ECC200 symbol) size

- (NSData *)**createBarcodeDataMatrix**:(NSString *)data **dataEncoding**:(NSStringEncoding)encoding **symbolHeight**:(uint8_t)height **symbolWidth**:(uint8_t)width **moduleSize**:(uint8_t)size

createBarcodeDataMatrix: symbolHeight: symbolWidth: moduleSize: Same with API but other characters including ASCII can be printed .

For NSStringEncoding,TextVeiwcontroller/multiLanguageButton can be referred on the sample program.

- (NSData *)**createBarcodeQRCode**:(NSString *)data **symbolVersion**:(uint8_t)version **ECLevel**:(char)level **moduleSize**:(uint8_t)size

Create QR-Code 2D barcode data. (Only ASCII code can be printed.)

Parameters

- data* The barcode source data
- version* The version of the symbol (0 ~ 40, 0 : auto size)
- level* The EC level ('L': 7%, 'M': 15%, 'Q': 25%, 'H': 30%)
- size* The module size (1 ~ 8)

EC Level(Error correction level)	description
L (Low)	7% of code words can be restored.
M (Medium)	15% of code words can be restored.
Q (Quartile)	25% of code words can be restored.
H (High)	30% of code words can be restored

Table 3 QR code EC Level

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

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

Table 4 QR Code capacity by EC Level

- (NSData *)**createBarcodeQRCode**:(NSString *)data **dataEncoding**:(NSStringEncoding)encoding **symbolVersion**:(uint8_t)version **ECLevel**:(char)level **moduleSize**:(uint8_t)size

createBarcodeQRCode: symbolVersion: ECLevel: moduleSize: Same with API but other characters including ASCII can be printed .

For NSStringEncoding,TextVeiwcontroller/multiLanguageButton can be referred on the sample program.

- (NSData *)**createBarcodeMicroPDF417**:(NSString *)data **column**:(uint8_t)column **row**:(uint8_t)row **HVRatio**:(uint8_t)ratio

Create micro PDF417 2D barcode data. It is influenced by *setBarcodeWidth*:. (Only ASCII code can be printed.)

Parameters

- data* The barcode source data
- column* The column number of 2D barcode (1 ~ 4)
- row* The row number of 2D barcode (4 ~ 44, 0 : auto size)
- ratio* The horizontal and vertical ratio (2 ~ 5)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

Number of Columns	Number of 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

Table 5 Micro PDF 417 Barcode

- (NSData *)**createBarcodeMicroPDF417**:(NSString *)data **dataEncoding**:(NSStringEncoding) encoding **column**:(uint8_t)column **row**:(uint8_t)row **HVRatio**:(uint8_t)ratio;

createBarcodeMicroPDF417: column: row: HVRatio: Same with API but other characters including ASCII can be printed .

For NSStringEncoding,TextVeiwcontroller/multiLanguageButton can be referred on the sample program.

- (NSData *)**createBarcodeTruncPDF417**:(NSString *)data **column**:(uint8_t)column **securityLevel**:(uint8_t)level **HVRatio**:(uint8_t)ratio

Create truncated PDF417 2D barcode data. It is influenced by *enableHRI*: and *setBarcodeWidth*:. (Only ASCII code can be printed.)

Parameters

<i>data</i>	The barcode source data
<i>column</i>	The column number of 2D barcode (1 ~ 4)
<i>level</i>	The security level to restore when bar code image is damaged (0 ~ 8)
<i>ratio</i>	The horizontal and vertical ratio (2 ~ 5)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**createBarcodeTruncPDF417**:(NSString *)data **dataEncoding**:(NSStringEncoding) encoding **column**:(uint8_t)column **securityLevel**:(uint8_t)level **HVRatio**:(uint8_t)ratio

createBarcodeTruncPDF417: column: securityLevel: HVRatio: Same with API but other characters including ASCII can be printed .

For NSStringEncoding,TextVeiwcontroller/multiLanguageButton can be referred on the sample program.

- (NSData *)**createBarcodeMaxicode**:(NSString *)data **mode**:(uint8_t)mode

Create Maxicode 2D barcode data. (Only ASCII code can be printed.)

Parameters

<i>data</i>	The barcode source data
	When <i>mode</i> is 2 or 3, first 15 bytes of the data is primary data. The primary data structure is as follow:
	- Post/Zip code (9 bytes)
	<i>mode</i> = 2: 5 digit zip code + 4 digit code extension. If code extension does not exist, then "0000" must be specified.

- mode* = 3: 6 alpha-numeric bytes + 3 bytes filler(ex. Spaces)
- Country code (3 digit): ISO 3166
- Class of service (3 digit)

mode The mode of Maxicode (2 ~ 6)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**createGS1Databar**:(NSString *)*data* **type**:(uint8_t)*type* **rowSegment**:(uint8_t)*segment*

Create GS1 databar data. It may be available since the firmware built on Oct. 11, 2012.

Parameters

- data* The barcode source data
- type* The GS1 databar type (0 ~ 6)
- segment* The segment per row that should be even number (2 ~ 20)
This parameter is valid only for type 6

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

Type	Data	Segment
0: GS1 Databar Omnidirectional	This field should be digits less than 14.	$2 \leq s \leq 20$
1: GS1 Databar Truncated		
2: GS1 Databar Stacked		
3: GS1 Databar Stacked Omnidirectional		
4: GS1 Databar Limited		
5: GS1 Databar Expanded	For AI, use '[' and ']' instead of '(' and ')'. Ex) "(01)90012345678908(3103)012233"	$2 \leq s \leq 20$ (even number)
6: GS1 Databar Expanded Stacked	->"[01]90012345678908[3103]012233"	

Table 6 GS1 Data bar barcode

- (NSData *)**enableHRI**:(BOOL)*HRI*

Turns HRI characters print mode on or off. It affects to 1 dimensional barcode, PDF417, and truncated PDF417.

Parameters

- HRI* Human Readable Interpretation (HRI) characters print mode indicator.
If true, HRI characters are printed at the bottom of barcode

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid

value.

- (NSData *)**setBarcodeWidth**:(uint8_t)width

Set barcode width. It affects to 1 dimensional barcode and PDF417 series.

Parameters

width The range of this parameter is (1 ~ 8)
If the width is out of printable area, barcode printing is ignored

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

- (NSData *)**setBarcodeHeight**:(uint8_t)height

Set barcode height in dot unit.

Parameters

height The range of this parameter is (1 ~ 255)

Return

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

2.3.2. Page mode

In the page mode, printable data can be assigned to a desired location and be printed all.

All methods described in this section except **enterPageMode** work properly in page mode.

Typical printing process in page mode

- 1) Enter page mode by using **enterPageMode**
- 2) Append printable data 1
- 3) Append printable data 2
- 4) Append printable data n
- 5) Print them by using **printDataInPageMode**
- 6) Exit page mode by using **exitPageMode**

- (NSData *)**clearDataInPageMode**

It can delete all data designated in the page mode. It is available only on the page mode.

Returns

Returns a pointer of NSData stream of control command.

References

enterPageMode, printDataInPageMode, and exitPageMode

- (NSData *)**createDrawingAreaWithStartPositionX:(uint16_t)positionX
withStartPositionY:(uint16_t) positionY withAreaWidth:(uint16_t)width
withAreaHeight:(uint16_t)height**

Set the page area. It is available only on the page mode.

Parameters

<i>positionX</i>	The start position in the X-axis in the page area
<i>positionY</i>	The start position in the Y-axis in the page area
<i>width</i>	Width of page area (unit : dot)
<i>height</i>	Height of page area (unit : dot, maximum : 2400 dot, 30 cm)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, exitPageMode, and changeCentimeterToDot:

- (NSData *)**createImageInPageMode:(CGImageRef)cgImage withStartPositionX:(uint16_t)positionX
withStartPositionY:(uint16_t)positionY**

Create image on the designated position in the page mode. In the case of large image, it may be printed slowly. It is available only on the page mode.

Parameter

<i>cgImage</i>	CGImage data
<i>positionX</i>	The position in the X-axis to print the image
<i>positionY</i>	The position in the Y-axis to print the image

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, exitPageMode;, and changeCentimeterToDot:

- (NSData *)**drawBoxWithWidth:(uint16_t)width withHeight:(uint16_t)height withLineThickness:
(uint8_t)thickness**

Draw a box in the page mode.

Parameters

<i>width</i>	Width of box
<i>height</i>	Height of box
<i>thickness</i>	Thickness of box line

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, exitPageMode, movePositionToX;, and changeCentimeterToDot:

```
- (NSData *)drawDiagonalLineWithStartPositionX:(uint16_t)startX withStartPositionY:(uint16_t)startY withEndPositionX:(uint16_t)endX withEndPositionY:(uint16_t)endY withThickness:(uint8_t)thickness
```

It draws a diagonal line in the Page Mode.

Parameters

startX X-coordinate of a starting point.
startY Y-coordinate of a starting point.
endX X-coordinate of an ending point.
endY Y-coordinate of an ending point.
thickness Thickness of diagonal line

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

```
- (NSData *)drawEllipseWithCenterPositionX:(uint16_t)x withCenterPositionY:(uint16_t)y withRadiusX:(uint16_t)radiusX withRadiusY:(uint16_t)radiusY withThickness:(uint8_t)thickness
```

It draws a circle and an ellipse in the Page Mode.

Parameters

x X-coordinate of a center point
y Y-coordinate of a center point
radiusX A radius on X-coordinate
radiusY A radius on Y-coordinate
thickness Thickness of ellipse line

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**drawHorizontalLineWithLength:(uint16_t)length withLineThickness:(uint8_t)thickness**

Draw a horizontal line in the page mode.

Parameters

length Length of line (unit : dot)
thickness Thickness of line (range: 0 ~ 255)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, exitPageMode, movePositionToX;, and
 changeCentimeterToDot:

- (NSData *)**drawVerticalLineWithLength:(uint16_t)length withLineThickness:(uint8_t)thickness**

Draw a vertical line in the page mode.

Parameters

length Length of line (dot unit)
thickness Thickness of line (range: 0 ~ 255)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, exitPageMode, movePositionToX;, and
 changeCentimeterToDot:

- (NSData *)**enterPageMode**

Start page mode

Returns

Returns a pointer of NSData stream of control command.

References

printDataInPageMode and exitPageMode

- (NSData *)**exitPageMode**

Exit the page mode and move to the standard mode. It would be called after the **printDataInPageMode** method.

Returns

Returns a pointer of NSData stream of control command.

References

enterPageMode and printDataInPageMode

- (NSData *)**feedLineInPageMode**

Feed a line as the an interval in the page mode.

Returns

Returns a pointer of NSData stream of control command.

References

enterPageMode, printDataInPageMode, exitPageMode, setLineSpacing;, and resetLineSpacing

- (NSData *)**feedNDotInPageMode:(uint8_t)n**

Feed the designated dots in the page mode.

Parameter

n dot unit value to be feed (0 ~ 255)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

Reference

enterPageMode, printDataInPageMode, and exitPageMode

- (NSData *)**feedNLineInPageMode:(uint8_t)n**

Feed the designated lines as an interval in the page mode.

Parameter

n line unit value to be feed (0 ~ 255)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, exitPageMode, setLineSpacing;, and resetLineSpacing

- (NSData *)**movePositionToX:(uint16_t)pointX toY:(uint16_t)pointY**

Assign the position to be printed in the page mode.

Parameters

pointX X-coordinate of start position

pointY Y-coordinate of start position

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid

value.

References

drawBoxWithWidth:, drawVerticalLineWithLength:, addString:, addStringWithTrueTypeFont:, selectLogoImage:, and so on

- (NSData *)printDataInPageMode

It prints data on the designated area in the page mode. Returns

Returns a pointer of NSData stream of control command.

References

enterPageMode and exitPageMode

- (NSData *)setPrintingDirectionInPageMode:(DIRECTION)direction

It sets a printing direction in the page mode.

Parameters

direction printing direction (0 ~ 3)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

enterPageMode, printDataInPageMode, and exitPageMode

Direction parameter	Print direction	Starting position
0	Left to right	Upper left (A in the figure)
1	Bottom to top	Lower left (B in the figure)
2	Right to left	Lower right (C in the figure)
3	Top to bottom	Upper right (D in the figure)

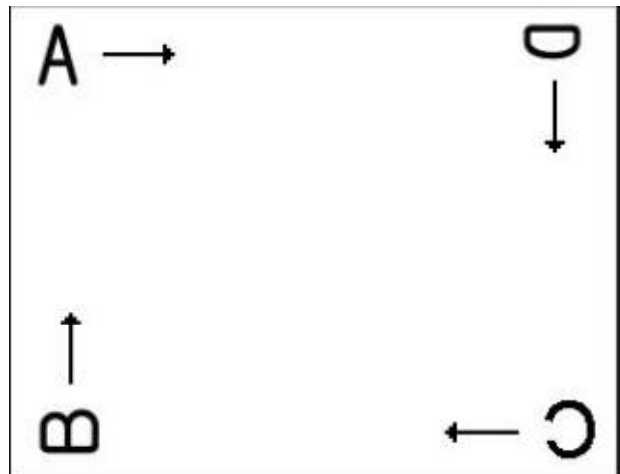


Table 7 Print Direction

2.3.3. Standard mode

In the standard mode, printable data would be printed line by line except for an image and a barcode. All methods described in this section work properly in the standard mode.

The standard mode is the default mode of Woosim printer.

- (NSData *)**createImageInStandardMode:(CGImageRef)cgImage
withShiftPosition:(uint16_t)position**

It creates an image on the designated position in the standard mode. To print the created image, you may use the **printImageInStandardMode** method. In the case of a large image, it may be printed slowly.

Parameters

cgImage CGImage data
position The position of the X-axis to print a image

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

References

printImageInStandardMode

- (NSData *)**printAndFeedNDotInStandardMode:(uint8_t)dot**

It prints data after feeding the designated dots in the standard mode.

Parameters

dot dot value to be feed (0 ~ 255)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

printDataInStandardMode and printAndFeedNLineInStandardMode

- (NSData *)**printAndFeedNLineInStandardMode:(uint8_t)line**

It prints data after feeding the designated lines in the standard mode. Parameters

line line value to be feed (0 ~ 255)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setLineSpacing;, resetLineSpacing, and printDataInStandardMode

- (NSData *)**printDataInStandardMode**

Print data in standard mode.

Returns

Returns a pointer of NSData stream of control command.

References

printAndFeedNLineInStandardMode and printAndFeedNLineInStandardMode

- (NSData *)**printImageInStandardMode**

It prints an image in the standard mode. It may be called after **createImageInStandardMode** method. In the case of a large image, it may be printed slowly.

Returns

Returns a pointer of NSData stream of control command.

References

createImageInStandardMode

- (NSData *)**setLeftMarginInStandardMode:(uint16_t)margin**

It sets left margin for the current line in the standard mode. If the margin value exceeds printable area, this method may be ignored.

Printable area:

1inch printer: 192 dots	2inch printer: 384 dots
3inch printer: 576 dots	4inch printer: 832 dots

Parameters

margin left margin (dot unit)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setPrintableAreaWidthInStandardMode

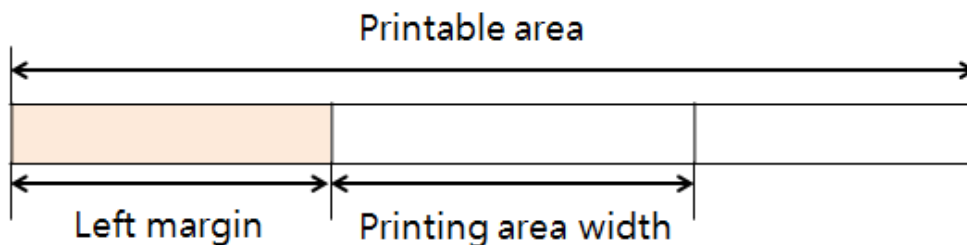


Figure 1 Left margin

- (NSData *)**setPrintableAreaWidthInStandardMode:(uint16_t)width**

It sets the width of a printing area for the current line in the standard mode. If the left margin plus the width of the printing area exceeds printable area, this method may be ignored.

Printable area:

1inch printer: 192 dots	2inch printer: 384 dots
3inch printer: 576 dots	4inch printer: 832 dots

Parameters

width printing area width (dot unit)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setLeftMarginInStandardMode

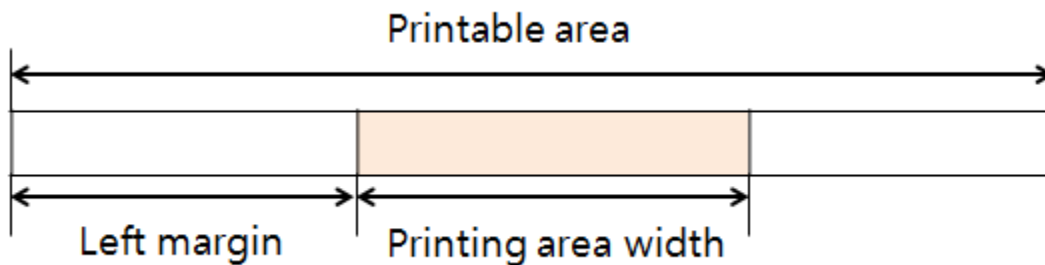


Figure 2 Printing area width

- (NSData *)**setPrintingUpsideDownInStandardMode:(Boolean)upsideDown**

It turns on or off upside-down printing in the standard mode.

Parameters

upsideDown YES: turn on upside down mode
 NO: turn off upside down mode

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

- (NSData *)**setTextAlignInStandardMode:(ALIGN)align**

It sets alignment in the standard mode.

Parameters

align set alignment (ALIGN_LEFT, ALIGN_CENTER, ALIGN_RIGHT)
 selected alignment value (ALIGN_LEFT, ALIGN_CENTER, or ALIGN_RIGHT)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

2.3.4. Text style

All methods described in this section work properly in the standard mode and the page mode. True type font configuration is not applied to imaging and barcode HRI.

- (NSData *)**addString**:(NSString *)string **encoding**:(NSStringEncoding) encoding

It makes string data with specific encoding form.

Parameters

<i>string</i>	string data
<i>encoding</i>	string encoding type

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**addString**:(NSString *)string **encoding**:(NSStringEncoding)encoding **withTextWidth:** (TEXTWIDTH)width **withTextHeight:**(TEXTHEIGHT)height **withTextBold:**(Boolean)bold

It makes string data with specific encoding form and text style effects.

Parameters

<i>string</i>	string data
<i>encoding</i>	string encoding type
<i>width</i>	character width
<i>Height</i>	character height
<i>bold</i>	turn on or off bold style

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**resetLineSpacing**

It sets one line feeding value to default value 30 dots.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**reverseTextColor**:(Boolean)reverseMode

It reverses text black and white color. It is not applied to barcode HRI, image and true type font.

Parameters

reverseMode YES: turn on reverse effect
 NO: turn off reverse effect

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setTextBold, setTextUnderline: withLineThickness:, and setTextSizeWithWidth: withHeight:

- (NSData *)**selectTextCodeTable:(LANGUAGE)codetable**

It selects language code to print. This method has no meaning for alpha-numeric and symbol characters in ASCII range (0x00 ~ 0x7F). The default language code is LANGUAGE_CP437, but some printers have other values. Woosim printer has one DBCS language among Korean, Japanese, and Chinese.

Parameters

codetable language code (Refer to the <Table 8>)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

- (NSData *)**setFontSize:(FONTSIZE)type**

It sets font size. Some font types cannot be selected by language restriction (refer to the Table 8). All of the existing text style effects to be set will be reset such as Underline, bold and etc. The **setFontSize()** method can be called before calling the APIs related to the text style.

Parameters

type font size (FONTSIZE_A, FONTSIZE_B, or FONTSIZE_C)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setTextSizeWithWidth: withHeight: , setMacroForTextCodeTable: size:, and selectTextCodeTable

FONTSIZE	LANGUAGE
	LANGUAGE_CP437 : USA, Standard Europe
FONTSIZE_A: 12 x 24	LANGUAGE_CP850 : Multilingual(Latin-1)
FONTSIZE_B: 9 x 24	LANGUAGE_CP860 : Portuguese
FONTSIZE_C: 8 x 16	LANGUAGE_CP852 : Slavic(Latin-2)
	LANGUAGE_CP857 : Turkish

	LANGUAGE_CP737 : Greek LANGUAGE_CP866 : Russian(Cyrillic) LANGUAGE_CP775 : Baltic LANGUAGE_ISO8859_15 : Latin-9 LANGUAGE_WIN1252 : Latin1 LANGUAGE_CP858 : Multilingual Latin I + Euro LANGUAGE_WIN1251 : Russian(Cyrillic) LANGUAGE_WIN1250 : Central Europe LANGUAGE_WIN1253 : Greek LANGUAGE_WIN1254 : Turkish LANGUAGE_WIN1255 : Hebrew LANGUAGE_WIN1258 : Vietnam LANGUAGE_WIN1257 : Baltic
FONTSIZE_A: 12 x 24 FONTSIZE_B: None FONTSIZE_C: None	LANGUAGE_CP874 : Thai
FONTSIZE_A: 16 x 24 FONTSIZE_B: None FONTSIZE_C: None	LANGUAGE_WIN1256 : Arabic
FONTSIZE_A: 24 x 24 FONTSIZE_B: 16 x 24 FONTSIZE_C: None	LANGUAGE_EUC_KR : Korean
FONTSIZE_A: 24 x 24 FONTSIZE_B: None FONTSIZE_C: None	LANGUAGE_GB18030 : Simplified Chinese LANGUAGE_Big5 : Traditional Chinese LANGUAGE_SHIFT_JIS : Japanese

Table 8 Font language

- (NSData *)**setLetterSpacing:** (uint8_t)spacing

It sets space between letters. The space is made on right side of letter, and the default value is 0.

The space between letters alters as the letter size

Parameters

spacing space between letters in dot unit (0~255)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setTextSizeWithWidth: withHeight and changeCentimeterToDot

- (NSData *)**setLineSpacing**:(uint8_t)spacing

It sets space between lines. The default value is 30 dots.

Parameters

spacing space between lines in dot unit (0~255)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

resetLineSpacing, printDataInStandardMode, feedNLineInPageMode, and changeCentimeterToDot

- (NSData *)**setTextBold**:(Boolean)bold

It turns on or off bold text effect. It is not applied to barcode HRI, image, and true type font.

Parameter

bold YES: turn on bold text effect
 NO: turn off bold text effect (default value)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

Reference

setTextUnderline: withLineThickness:, setTextSizeWithWidth: withHeight:, and reverseTextColor:

- (NSData *)**setTextSizeWithWidth**:(TEXTWIDTH)width **withHeight**:(TEXTHEIGHT)height

It sets the width and/or the height of characters to 1~8 times larger than the basic font size. It is not applied to barcode HRI, image, and true type font.

Parameters

width number of times width from basic font width (1~8)
height number of times height from basic font height (1~8)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setFontSize:

Parameter <i>width</i>	Parameter <i>height</i>
TEXTWIDTH_1: basic width	TEXTHEIGHT_1: basic height
TEXTWIDTH_2: double width	TEXTHEIGHT_2: double height
TEXTWIDTH_3: 3 times width	TEXTHEIGHT_3: 3 times height

TEXTWIDTH_4: 4 times width	TEXTHEIGHT_4: 4 times height
TEXTWIDTH_5: 5 times width	TEXTHEIGHT_5: 5 times height
TEXTWIDTH_6: 6 times width	TEXTHEIGHT_6: 6 times height
TEXTWIDTH_7: 7 times width	TEXTHEIGHT_7: 7 times height
TEXTWIDTH_8: 8 times width	TEXTHEIGHT_8: 8 times height

Table 9 Text size

- (NSData *)**setTextUnderline:**(Boolean)underline **withLineThickness:**(int)thickness

It turns on or off the text underline effect. It is not applied to barcode HRI, image, and true type font.

Parameters

- underline* YES: turn on underline
NO: turn off underline
- thickness* thickness of underline in dot unit (1 ~ 2)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

setTextBold;, setTextSizeWithWidth: withHeight;, and reverseTextColor:

- (NSData *)**shiftAbsolutePosition:**(uint16_t)position

It sets the printing start position based on the beginning of the line. If the moved position pass the printable area, the next printable data cannot be printed.

Parameters

- position* dot length to be moved from the beginning of the line

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

shiftRelativePosition and changeCentimeterToDot

- (NSData *)**shiftRelativePosition:**(uint16_t)position

It sets the printing start position based on the current position. If the moved position pass the printable area, the next printable data cannot be printed.

Parameters

- position* dot length to be moved from the current position

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value

References

shiftAbsolutePosition and changeCentimeterToDot

2.3.5. True type font

- (NSData *)**addStringWithTrueTypeFont:(NSString*)data withStringWidth:(uint8_t)width
withStringHeight:(uint8_t)height**

It makes string data with true type font. Text style effects are not applied to string data.

Parameters

<i>data</i>	string data
<i>width</i>	character width (4 ~ 255)
<i>height</i>	character height (4 ~ 255)

Returns

Returns a pointer of printable NSData stream with control command or nil if any parameter has invalid value.

- (NSData *)**selectTrueTypeFontFile:(NSString*)fileName**

It selects true type font file stored in the printer. The extension of file should be "tff".

True type font files, WSTTFB.tff and WSTTFL.tff, including alpha-numeric and European symbol characters are saved in Woosim printers manufactured since May 2015. Additional true type font file can be downloaded by utility program of Woosim Systems Inc Please contact the sales department of Woosim Systems Inc. if you need the program.

Parameters

<i>fileName</i>	true type font file name including extension (English only)
-----------------	---

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

2.3.6. MSR

These methods below are used to get the information of magnetic stripe reader card tracks. Payment solution is not provided from Woosim Systems.

- (NSData *)**enterMSR1stTrackMode**

It enters into the magnetic card 1st track reading mode. When the control command is sent to the Woosim printers, the power lamp and the error lamp of the printer are blinking alternately. The

magnetic card reading mode will be terminated by swiping magnetic card, or sending control command to cancel magnetic card reading mode.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**enterMSR2ndTrackMode**

It enters into the magnetic card 2nd track reading mode. When this control command is sent to the Woosim printer, the power lamp and the error lamp of the printer are blinking alternately. The magnetic card reading mode will be terminated by swiping magnetic card, or sending control command to cancel magnetic card reading mode.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**enterMSR3rdTrackMode**

It enters into the magnetic card 3rd track reading mode only for 123 tracks MSR. When the control command is sent to the Woosim printer, the power lamp and the error lamp of the printer are blinking alternately. The magnetic card reading mode will be terminated by swiping a magnetic card or sending control command to cancel the magnetic card reading mode.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**enterMSRDoubleTrackMode**

It enters into the magnetic card dual track (track 1 and 2, or track 2 and 3 dependant on MSR type) reading mode. When the control command is sent to the Woosim printer, the power lamp and the error lamp of the printer are blinking alternately. The magnetic card reading mode will be terminated by swiping a magnetic card or sending control command to cancel the magnetic card reading mode.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**enterMSRTripleTrackMode**

It enters into the magnetic card triple track reading mode only for 123 tracks MSR. When the control command is sent to the Woosim printer, the power lamp and the error lamp of the printer are blinking alternately. The magnetic card reading mode will be terminated by swiping a magnetic card or sending control command to cancel the magnetic card reading mode.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**exitMSRMode**

It exits from the magnetic card reading mode.

Returns

Returns a pointer of NSData stream of control command.

2.3.7. Smart card

Smart card payment solutions are not provided from Woosim Systems.

- (NSData *)**enterSmartCardReaderMode**

It enters into the smart card reading mode. It is available for Woosim printers mounting the smart card reader module.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**exitSmartCardReaderMode**

It exits from the smart card reading mode. It is available for Woosim printers mounting the smart card reader module.

Returns

Returns a pointer of NSData stream of control command.

2.3.8. Label paper

- (NSData *)**feedToBlackMarkPosition**

It feeds paper by the pre-set length as the black mark position. The command will be sent to the last. The printer should be set as the Mark Use mode (MARK = USE). Please refer to the Woosim printer operator manual to configure the printer device. The document can be downloaded from Woosim Systems Inc.'s website. (www.woosim.com).

Returns

Returns a pointer of NSData stream of control command.

References

setBlackMarkPosition

- (NSData *)**setBlackMarkPosition:(uint16_t)position**

It sets the moving distance to the position of the cutter at the paper sensor of the printer. When the value is set, the power lamp and the error lamp are blinking in turn. The API cannot be used with other APIs together. Please use separately. The value to be set can be checked by pushing the feed button. If feeding more than the position to be set, the value of the position can be reduced. If feeding less, please increase the value. It is recommended that the position value should be from 0x80. The printer mode should be "Mark=Use" in order to the command. If more information on

the setting, please refer to the operator manual on the Woosim Systems Inc.'s website(<http://www.woosim.com>)..

Parameters

position dot length to be moved from the black mark

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

References

feedToBlackMarkPosition

2.3.9. Miscellaneous

- (uint16_t)**changeDotFromCentimeter:(float)cm**

It changes the parameter value from centimeter unit to dot unit (1 dot is 0.0125cm in 203dpi printer).

Parameters

cm centimeter unit value to change to dot unit

Returns

Returns dot unit value converted from centimeter unit

- (NSData *)**clearPrinterBuffer**

It initializes data in printer buffer.

Returns

Returns a pointer of NSData stream of control command.

- (NSData *)**cutPapaer:(CUT)type**

It cuts the paper. It is available for Woosim printers mounting the auto cutter module.

Parameters

type CUT_FULL: full cut
CUT_PARTIAL: partial cut

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

- (NSData *)**selectLogoImage:(uint8_t)index**

It selects a logo image stored in printer to print.

The image data should be downloaded before using the control command. It can be downloaded by utility program of Woosim Systems. Please contact sales department of Woosim Systems Inc. if you need the program.

Parameters

index index of a logo image (start from 0)

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

- (NSData *)**verifyPrinterVer**

The function can make the firmware version of the printer check with the return value. For the details, please refer to WSMiscellaneuousViewController/checkFirmwareButton.

Returns

Returns a pointer of NSData stream of control command or nil if any parameter has invalid value.

2.4. Deprecated Method

- (void)**createPrintableDataFromImage**:(CGImageRef)image coordinateX:(int)x coordinateY:(int)y width:(int)width height:(int)height

Convert CGImage data to user defined bit-image formats.

It should be called in page mode and can cover large image within the maximum page length limit. This converted data does not have printing command, so the application programmer should use additional printing command if it is necessary.

Parameters

image The CGImage data

x The horizontal starting position to print in dot unit

y The vertical starting position to print in dot unit

width The printing area width in dot unit. If width is less or equal to 0, it is used image width. Maximum value is according to each product width:
1 inch: 192, 2 inch: 384, 3 inch: 576, 4 inch: 832

height The printing area height in dot unit. If height is less or equal to 0, it is used image height. Maximum value is 2400

[Note] To avoid problems, the method should not be used with methods related to the page mode section.

3. WSParser Class

The WSParser class provides method for parsing incoming data from printer.

It Inherits from NSObject.

3.1. Tasks

Decoding income data

- decode:

3.2. Constants

Incoming data types used as the return value of *decode:* method.

```
typedef enum {
    WSDATA_MSR,
    WSDATA_MSRFAIL,
    WSDATA_SPECIAL,
    WSDATA_UNKNOWN
} WSDATATYPE
```

3.3. Instance Variable

@property (nonatomic, readonly) NSData ***track1**

The parsed MSR track 1 data coming from Woosim printer (read only).

@property (nonatomic, readonly) NSData ***track2**

The parsed MSR track 2 data coming from Woosim printer (read only).

@property (nonatomic, readonly) NSData ***track3**

The parsed MSR track 3 data coming from Woosim printer (read only).

@property (nonatomic, readonly) int specialDataType

In case incoming data type is WSDATA_SPECIAL as decoding result, this variable has detailed type information of the data.

3.4. Instance Methods

- (WSDATATYPE)**decode:**(NSData *)srcData

Decode data coming from Woosim printer, and process it according to the content type. If the data is from MSR, it will be decoded in the library and stored in each property.

Parameters

srcData The incoming data from printer

Returns

Returns data type of *srcData*.

4. Sample Codes

Woosim iOS SDK includes several sample applications to provide useful help.

4.1. BTPrint

It is a traditional sample application including following functions:

- Text file printing
- 1bpp bitmap printing
- Various 1D and 2D bar code printing
- Magnetic card reading mode setting and show the magnetic card data by swiping
- Various printing samples

Before run this application, user should connect iDevice to Woosim printer via Bluetooth. Bluetooth connection can be established by Bluetooth setting menu in the iDevice. When user runs the application, he/she can see the list of devices that can be connected through Bluetooth.

4.2. WiFiPrint

It has the same functions as BTPrint, but Wi-Fi connection is used instead of Bluetooth.

To setup Wi-Fi connection, user should know IP address and port number of target printer.