

FTP2176000N0
FCLPrintLib iOS
機能仕様書

第 3 版
2024 年 2 月 1 日

FCL コンポーネント株式会社

FTP2176000N0
FCLPrintLib iOS 機能仕様書

< 目 次 >

1. 適用
 2. 概要
 3. プリンタの接続方法
 4. ライブラリの使用方法
 5. プリンタ印刷用メソッド
 6. Notificationによる通知
 7. 各設定の初期値
 8. エラーコード
 9. 注意事項
 10. 既知の現象について
- 仕様書改版履歴表
ライブラリ改版履歴表

- ・本書の内容は予告なく変更する場合があります。
- ・本書の一部あるいは全部を無断で複写・複製・転載することは、固くお断りします。
- ・本書および本書に記載された製品の利用に起因するいかなる損害に対しても、当社は責任を負うものではありません。

- ・iPhone、iPad、iPad mini、iPad Air、iPad Pro、iPod touch、Xcode は、Apple, Inc. の米国およびその他の国における商標です。
- ・iOS は、米国およびその他の国における Cisco, Inc. の商標または登録商標であり、ライセンスに基づき使用されています。
- ・iPhone の商標は、アイホン株式会社のライセンスに基づき使用されています。
- ・Bluetooth® のワードマークおよびロゴは、Bluetooth SIG, Inc. が所有する登録商標であり、FCL コンポーネント株式会社はこれらのマークをライセンスに基づいて使用しています。その他の商標およびトレードネームは、それぞれの所有者に帰属します。
- ・QR コードは株式会社デンソーウェーブの登録商標です。
- ・その他、本書に記載されている商品・サービス名は、各社の商標または登録商標です。

1. 適用

本書は以下の FCL コンポーネント製 iOS 用ライブラリ FTP2176000N0 に適用します。

ライブラリセット名(提供ファイル名)	:FTP2176000N0_FCLPrintLib_iOS_Vxxx (FTP2176000N0_FCLPrintLib_iOS_Vxxx.zip) xxx はファイルバージョン
ファイル構成	:libFTP2176000N0_IOS.a FTP2176000N0_IOSLibrary.h FTPIOSCommonHeader.h
対応オペレーティングシステム	:iOS 11～17 iPadOS 13～17
対応アーキテクチャー	:arm64
対象プリンタ	:FTP-62HWSL000 シリーズ

2. 概要

本書では、ライブラリに実装されているプリンタ印刷用メソッドと使用例について記述します。

3. プリンタの接続方法

- (1) プリンタの電源を ON にします。
- (2) iOS の[設定]→[一般]→[Bluetooth]を開きます。
- (3) [Bluetooth]を[オン]にします。
- (4) デバイスリストにプリンタが表示されたら、タップします。
- (5) PIN コード入力画面が表示されたら PIN コードを入力します。
- (6) “接続されました”と表示されたら接続完了です。

4. ライブラリの使用方法

プロジェクトへの追加方法

- (1) FTP2176000N0_IOSLibrary.h、FTPIOSCommonHeader.h および libFTP2176000N0_IOS.a をプロジェクトフォルダ内または任意の場所にコピーします。以下の説明は Xcode7 の場合です。
- (2) Xcode 上からプロジェクトへ追加します。
右クリックメニューの "Add Files to (プロジェクト名) ..." から、FTP2176000N0_IOSLibrary.h、FTPIOSCommonHeader.h および libFTP2176000N0_IOS.a を追加します。
- (3) [Project Navigator]からプロジェクト名をクリックします。
[Targets]→[Build Phases]の[Link Binary With Libraries]にて
"ExternalAccessory.framework"
を追加します。
- (4) [Project Navigator]からプロジェクト名をクリックします。
[Targets]→[Info]の[Custom iOS Target Properties]にて、
"Supported external accessory protocols" Key
を追加します。
左側の矢印をクリックして[Item x](x は数字)に protocol を追加します。
protocol は[startPrinter:protocolName:series:メソッド](#)を参照して下さい。
- (5) ソースファイル上に"#import"を使って FTP2176000N0_IOSLibrary.h をインポートします。

基本的な印刷方法

- (1) iOS 設定画面にて、プリンタを iOS と接続します。
- (2) FTP2176000N0_IOSLibrary クラスを alloc メソッドおよび init メソッドを使用し、インスタンスを生成および初期化します。
例) FTP2176000N0_IOSLibrary* ftpLib = [[FTP2176000N0_IOSLibrary alloc] init];
- (3) startPrinter:メソッドを使用し、プリンタと通信可能な状態にします。
- (4) printCharacterString:メソッドで文字列を印刷します。
- (5) 印刷位置指定や回転印刷などを行う場合、startPage:メソッドを使用します。
- (6) setAbsolutePosition:メソッドで印刷位置絶対値指定を行います。
- (7) printBarcode:メソッドを使用してバーコードの展開を行います。
- (8) printPage:で展開したデータの印刷をします。
- (9) closePrinter:メソッドで通信終了します。
- (10) release メソッドでインスタンスを解放します。

5. プリンタ印刷用メソッド

ドライバセットに含まれる libFTP2176000N0_IOS.a は、アプリケーションからプリンタの制御を行うことを目的としたライブラリファイルです。

このライブラリファイル内のメソッドを呼び出すことで、アプリケーションはイメージ印刷やプリンタのステータス取得が可能です。

クラス名: FTP2176000N0_IOSLibrary

継承: NSObject

-メソッド一覧-

番号	メソッド名	機能
初期化系メソッド		
1	- startPrinter:protocolName:series:	プリンタとのセッションを開始します。
2	- closePrinter	プリンタとのセッションを終了します。
印刷系メソッド		
3	- detectMark	マーク検出をします。
4	- dotfeedPaper:	指定したドットライン数の用紙送りを行います。
5	- linefeedPaper:	指定した行数の用紙送りを行います。
6	- printBarcode:data:height:narrow:wide:hri:	バーコードを印刷します。
7	- printBitImage:width:height:compression:	ビットイメージを印刷します。
8	- printCharacterString:	文字列を印刷します。
9	- printCustomerBarcode:size:	カスタマバーコードを印刷します。
10	- printGS1DataBar:DataBarType: module:numberOfSegments:	GS1 DataBar を印刷します。
11	- printMaxiCode:mode: number:count:symbolCount:	マキシコードを印刷します。
12	- printPage:	印刷を開始します
13	- printPDF417Code:columnsOrSteps:width: height:errorLevel:	PDF417 を印刷します。
14	- printPicture:dithering:width:height:	イメージを印刷します
15	- printQRCode:modulemagnification:errorLevel: concatenation:code:division:parity:masking:	QR コードを印刷します。
16	- printRegisteredImage:mode:	登録イメージを印刷します。
17	- startPage:	ページモードを開始します。
印刷設定系メソッド		
18	- registBitImage:width:height:data:	ビットイメージデータからイメージ登録を行います。
19	- registPicture:image:dithering:width:height:	イメージ登録を行います。
20	- setAbsolutePosition:	展開位置を絶対値指定します。
21	- setCharacterStyle:kanjiSize:fontHSize: fontVSize:bold:bwReverse:underline:space: leftOfKanji:rightOfKanji:	文字修飾を指定します。
22	- setInternationalCharacterCode:country:	国際文字を指定します。
23	- setLineFeedAmount:page:	改行量を指定します。
24	- setOrientation:	印刷方向を指定します。

25	- setPaperType:feedBeforePrintPage: feedAfterMarkDetection:autoFeedLength:	用紙モード設定およびマーク検出頭出し量を指定します。
26	- setPrintSpeed:	印刷速度を指定します。
27	- setRelativePosition:	展開位置を相対値指定します。
28	- setRotation:	回転方向を指定します。
デバイス設定系メソッド		
29	- setAutoPowerOffTime:	自動パワーオフ時間を指定します。
30	- setBluetoothSecurityMode:encryption: PNCode:deviceName:	Bluetooth セキュリティモードを指定します。
31	- setLanguageOfCharacterString:	プリンタが搭載している言語を指定します。
32	- setPrintQuality:energy:halfPitch:autoDivesion:	印字品質、印加エネルギー補正およびハーフピッチモードを指定します。
プリンタ情報取得・待機系メソッド		
33	- getPrinterData:	プリンタ情報の取得を行います。
34	- waitPrintEnd:	印刷完了を待機します。

-プロパティ一覧 -

番号	プロパティ名	機能
1	@property(readonly) NSString* version	本ライブラリのバージョンが格納されています。 例)@"V3.00"

- インスタンスメソッド -

(1) startPrinter:protocolName:series:

- (int)startPrinter:(NSString *)*model*/ protocolName:(NSString *)*protocolName*
series:(int)*series*;
プリンタとのセッションを開始し、通信可能な状態にします。

パラメーター

model: modelNumber を指定します。(例: 62HWSL000_XXXXXX)

protocolName: protocol を指定します。(com.fujitsu.fcl.ftp-62hws1000-00)

series: 通信するプリンタシリーズを指定します。

<i>series</i>	プリンタシリーズ	印字幅(dot)
FTP_62hws1000	FTP-62HWSL000	384

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_AlreadyConnectedPrinter: 既にプリンタと接続中です。

Error_AlreadyConnectedPrinterOtherInstance: 他のインスタンスがプリンタを使用中です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

- 指定した *model* および *protocolName* に対応するプリンタを検索し、プリンタとの通信を確立します。また、本ライブラリのインスタンスの各パラメーターを初期化します。初期値は[各設定の初期値](#)を参照して下さい。
- プリンタが電源 OFF→ON されるなど、プリンタとの通信ができなくなった場合、iOS 設定画面よりデバイス再接続後、再度本メソッドを実行してください。その際、各パラメーターが初期化されます。
- プリンタ通信中に本メソッドを実行した場合、Error_AlreadyConnectedPrinter が返ります。
- 接続確立後、"[Notification](#)"により PrinterEvent_ConnectedPrinter が通知されます。

(2) closePrinter:

– (int)closePrinter;

プリンタとのセッションを終了します。

パラメーター

なし

戻り値

Error_Success: 成功しました。

解説

- ・プリンタとのセッションを終了します。
- ・ページモードの場合、ページモード領域のデータを破棄してからセッションをクローズします。
- ・データ送信中および印字/登録中に実行すると、データ異常が発生し誤動作の要因となる可能性があります。よって本メソッドは、Notification による各メソッドの結果通知後に実行することを推奨します。

(3) detectMark:

- (int)detectMark;
マーク検出を行います。

パラメーター

なし

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。

解説

- ・次のマーク位置まで用紙送りをを行います。
- ・行モードの場合、マーク検出動作が実行される位置から現在設定されているページ長以内にマークが検出されなかった場合には、マーク未検出となり用紙送りを停止します。ページ長は setLineFeedAmount:page:メソッドにて指定します。
- ・ページモードの場合、印刷後にマーク検出を行います。ページモード領域内にデータが展開されていない場合、マーク検出のみ行います。また、印刷後にページモード領域内のデータを破棄し、ページモードを継続します。展開位置はページモード領域の左上端に移動します。マーク検出動作が実行される位置から 204mm 以内にマークが検出されなかった場合には、マーク未検出となり用紙送りを停止します。

(4) dotfeedPaper:

- (int)dotfeedPaper:(int)*nDotLineFeed*,
指定したドットライン数の用紙送りを行います。

パラメーター

nDotLineFeed: 用紙送りするドットライン数を指定します。設定範囲 (0~255)

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。
Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・指定したドットライン数の用紙送りを行います。ページモードでは展開位置の移動だけで実際の用紙送り動作は行われません。なお展開位置は縦方向のみ変化し、横方向は変化しません。
- ・*nDotLineFeed* == 0 を指定した場合、バッファー内にデータがあればバッファー内データの印字のみ実行し、バッファー内にデータがなければ用紙送り動作は行われません。

(5) linefeedPaper:

- (int)linefeedPaper:(int)*nLineFeed*,
指定した行数の改行を行います。

パラメーター

nLineFeed: 改行数を指定します。設定範囲 (0～255)

戻り値

Error_Success: 成功しました。

Error_IllegalParameter: パラメーターが範囲外です。

Error_NoDevice: プリンタ未検出です。

解説

- ・指定した行数の改行を行います。
- ・行の高さは setLineFeedAmount:メソッドによって設定されます。ページモードでは展開位置の移動だけで実際の印刷は行われません。
- ・*nLineFeed* == 0 を指定した場合、バッファー内にデータがあればバッファー内データの印刷のみ実行し、バッファー内にデータがなければ改行動作は行われません。

(6) printBarcode:data:height:narrow:wide:hri:

– (int)printBarcode:(int)*symbol*/ data:(NSData *)*data* height:(int)*height* narrow:(int)*narrowBar* wide:(int)*wideBar* hri:(int)*hri*;
バーコードを印刷します。

パラメーター

symbol: バーコードの種類を指定します。

<i>symbol</i>	バーコードのタイプ
Barcode_Upca	UPC-A
Barcode_Upce	UPC-E
Barcode_Ean8	EAN 8 (JAN 8)
Barcode_Ean13	EAN 13 (JAN 13)
Barcode_Itf	ITF (Interleaved 2 of 5)
Barcode_Codabar	CODABAR
Barcode_Code39	CODE39
Barcode_Code128	CODE128

data: バーコードのデータを指定します。

height: 高さをドット単位で指定します。設定範囲(1～255)

narrowBar: ナローバーの幅をドット単位で指定します。設定範囲(1～255)

wideBar: ワイドバーの幅をドット単位で指定します。設定範囲(1～255)

hri: HRI 文字の有無を選択します。

<i>hri</i>	HRI 文字の位置
HRI_None	HRI 文字なし
HRI_Below	バーコードの下

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・ページモードではバーコード展開だけで実際の印刷は行われません。
- ・UPC-A、UPC-E、EAN 8(JAN 8)、EAN 13(JAN 13)、CODE128 の場合、*wideBar* の値は無視されます。
- ・HRI 文字のサイズは 8x16、12x24、16x16、24x24 でバーコードの幅に合わせて自動設定します。
- ・setRotation:メソッドで回転を指定している場合、バーコード回転位置に合わせて HRI 文字も移動します。

- ・以下にパラメーター範囲を示します。パラメーター範囲外ではスキャナーで正常に読み取りできない可能性がありますので、使用しないで下さい。

バーコード	有効なパラメーター範囲	データ数
UPC-A	"0～9" の 10 種	11～12
UPC-E		11～12
EAN8 (JAN8)		7～8
EAN13(JAN13)		12～13
CODE39	"0～9" , "A～Z" , 空白, "\$ % + - * . /" の 43 種	1～
ITF	"0～9" の 10 種	2～
CODABAR	"0～9" , "A～D" , "\$ + - . / : " の 20 種	3～
CODE128	Code Set A: 0x00 ～ 0x5F Code Set B 0x20 ～ 0x7F Code Set C 0x00 ～ 0x63 但し、"I"を含むキャラクターは除く。	3～

- ・同一行内にデータが存在する場合、サイズによっては Error_Success が返る場合でも印刷できません。
- ・CODE39 を指定する場合、スタートキャラクターとストップキャラクターは自動的に付加されますので、キャラクターに設定する必要はありません。
- ・ITF を指定する場合、キャラクター数は偶数で指定する必要があります。奇数を指定した場合はパラメーター異常となります。
- ・CODABAR を指定する場合、キャラクターの先頭と末尾はかならず、'A' ～ 'D' のいずれかである必要があります。よってその間に任意のキャラクターを設定した合計 3 文字以上で指定する必要があります。これに従っていない場合は、パラメーター異常となります。
- ・UPC-E を指定する場合、展開は下記の表に基づいて行われます。UPC-A レフトコードは先頭 2 ～ 6 キャラクター、UPC-A ライトコードは先頭 7 ～ 11 キャラクター目を示し、短縮されるコードが実際に UPC-E として印字されます。UPC-A 先頭キャラクターが 0 以外か、下記の表に基づかないキャラクターが指定された場合、パラメーター異常となります。

例 05810000226 → 58226 に変換されます。
 09859363583 → パラメーター異常となります。

メーカーコード UPC-A のレフトコード					アイテムコード UPC-A のライトコード					短縮コード					
F1	F2	F3	F4	F5	A1	A2	A3	A4	A5	Z1	Z2	Z3	Z4	Z5	Z6
0-9	0-9	0	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	0
0-9	0-9	1	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	1
0-9	0-9	2	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	2
0-9	0-9	3-9	0	0	0	0	0	0-9	0-9	F1	F2	F3	A4	A5	3
0-9	0-9	0-9	1-9	0	0	0	0	0	0-9	F1	F2	F3	F4	A5	4
0-9	0-9	0-9	0-9	1-9	0	0	0	0	5-9	F1	F2	F3	F4	F5	A5

・CODE128 を印字する場合は以下のようにキャラクターを設定します。

1. バーコードの先頭は、“{A”, “{B”, “{C” のいずれかを指定してください。以下をそれぞれ、CODE A, CODE B, CODE C に設定します。
2. ファンクションコードを指定するには、“{1”, “{2”, “{3”, “{4”を指定します。それぞれ、FNC1, FNC2, FNC3 FNC4 の指定となります。
尚、CODE C では、FUNC1 のみ有効です。CODE C で、FUNC1 以外を指定するとパラメーター異常となります。
3. CODE B で、“{”を印字するには、“{[”を指定してください。
4. SHIFT を設定するには、“{S”を指定してください。以後の 1 キャラクターのコードセットが、CODE A ↔ CODE B のようにシフトします。CODE C で指定した場合、パラメーター異常となります。

・CODE A, CODE B, CODE C で印字可能なキャラクターを次に示します。

設定値 10 進	設定値 16 進	CODE A	CODE B	CODE C
0	0	NUL		0
1	1	SOH		1
2	2	STX		2
3	3	ETX		3
4	4	EOT		4
5	5	ENQ		5
6	6	ACK		6
7	7	BEL		7
8	8	BS		8
9	9	HT		9
10	0A	LF		10
11	0B	VT		11
12	0C	FF		12
13	0D	CR		13
14	0E	SO		14
15	0F	SI		15
16	10	DLE		16
17	11	DC1		17
18	12	DC2		18
19	13	DC3		19
20	14	DC4		20
21	15	NAK		21
22	16	SYN		22
23	17	ETB		23
24	18	CAN		24
25	19	EM		25
26	1A	SUB		26
27	1B	ESC		27
28	1C	FS		28
29	1D	GS		29
30	1E	RS		30
31	1F	US		31
32	20	SP	SP	32
33	21	!	!	33
34	22	”	”	34

設定値 10 進	設定値 16 進	CODE A	CODE B	CODE C
64	40	@	@	64
65	41	A	A	65
66	42	B	B	66
67	43	C	C	67
68	44	D	D	68
69	45	E	E	69
70	46	F	F	70
71	47	G	G	71
72	48	H	H	72
73	49	I	I	73
74	4A	J	J	74
75	4B	K	K	75
76	4C	L	L	76
77	4D	M	M	77
78	4E	N	N	78
79	4F	O	O	79
80	50	P	P	80
81	51	Q	Q	81
82	52	R	R	82
83	53	S	S	83
84	54	T	T	84
85	55	U	U	85
86	56	V	V	86
87	57	W	W	87
88	58	X	X	88
89	59	Y	Y	89
90	5A	Z	Z	90
91	5B	[[91
92	5C	\	\	92
93	5D]]	93
94	5E	^	^	94
95	5F	_	_	95
96	60		,	96
97	61		a	97
98	62		b	98

35	23	#	#	35
36	24	\$	\$	36
37	25	%	%	37
38	26	&	&	38
39	27	'	'	39
40	28	((40
41	29))	41
42	2A	★	★	42
43	2B	+	+	43
44	2C	,	,	44
45	2D	-	-	45
46	2E	.	.	46
47	2F	/	/	47
48	30	0	0	48
49	31	1	1	49
50	32	2	2	50
51	33	3	3	51
52	34	4	4	52
53	35	5	5	53
54	36	6	6	54
55	37	7	7	55
56	38	8	8	56
57	39	9	9	57
58	3A	:	:	58
59	3B	;	;	59
60	3C	<	<	60
61	3D	=	=	61
62	3E	>	>	62
63	3F	?	?	63

99	63		c	99
100	64		d	
101	65		e	
102	66		f	
103	67		G	
104	68		h	
105	69		i	
106	6A		j	
107	6B		k	
108	6C		l	
109	6D		m	
110	6E		n	
111	6F		o	
112	70		p	
113	71		q	
114	72		r	
115	73		s	
116	74		t	
117	75		u	
118	76		v	
119	77		w	
120	78		x	
121	79		y	
122	7A		z	
123	7B		{	
124	7C			
125	7D		}	
126	7E		~	
127	7F			

送信データ		特殊キャラクター	意味
16 進	ASCII		
7Bh+31h	{1	FNC1	ファンクションキャラクター
7Bh+32h	{2	FNC2	
7Bh+33h	{3	FNC3	
7Bh+34h	{4	FNC4	
7Bh+41h	{A	CODE A	コードセット選択キャラクター
7Bh+42h	{B	CODE B	
7Bh+43h	{C	CODE C	
7Bh+53h	{S	SHIFT	シフトキャラクター
7Bh+7Bh	{['{'	文字'{'を指定する

- ・展開されるバーコードサイズ(HRI 文字を含む)が印字領域を超える場合は Error_IllegalParameter を返します。

印字モード	角度	印刷方向	最大幅および高さ
行モード	0°	–	印字幅(dot)
ページモード	0°	左→右	幅: startPage メソッドの size.width
		右→左	
		上→下	幅: startPage メソッドの size.height
		下→上	
	180° 回転	左→右	幅: startPage メソッドの size.width - 1
		右→左	
		上→下	幅: startPage メソッドの size.height - 1
		下→上	
	右 90° 回転	左→右	幅: startPage メソッドの size.height
		右→左	
		上→下	幅: startPage メソッドの size.width
		下→上	
	左 90° 回転	左→右	幅: startPage メソッドの size.height - 1
		右→左	
		上→下	幅: startPage メソッドの size.width - 1
		下→上	

- ・ページモードの場合、展開されるバーコード幅および高さが範囲内であれば Error_Success を返しますが、印字開始位置 (setRelativePosition:メソッドまたは setAbsolutePosition:メソッドで指定) によっては印刷されません。
- ・角度および印刷方向に関わらず、バーコード幅が 864 ドットを超える場合、Error_IllegalParameter を返します。

- ・バーコードの展開には、各バーコードで規定されているクワイエットゾーンを確保して配置して下さい。他のバーコードやイメージの展開で重なり、クワイエットゾーンが確保されないとバーコードが正常に読み取れない可能性があります。

(7) printBitImage:width:height:compression:

- (int)printBitImage:(NSData *)*data* width:(int)*widthDots* height:(int)*heightDots* compression:(BOOL)*dataCompression*;
ビットイメージを印刷します。

パラメーター

- data*: ビットイメージデータを指定します。
- widthDots*: 幅をドット単位で指定します。設定範囲(解説参照)
- heightDots*: 高さをドットライン単位で指定します。設定範囲(解説参照)
- dataCompression*: データを圧縮して送信するか選択します。
NO PackBits 圧縮なし
YES PackBits 圧縮あり

戻り値

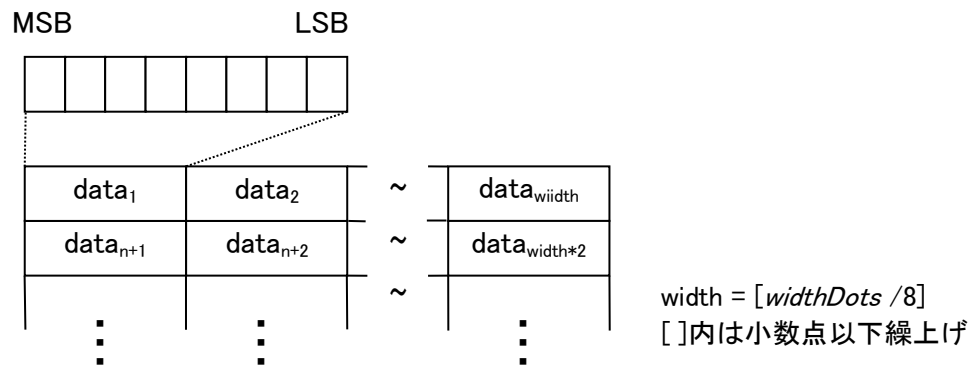
- Error_Success: 成功しました。
- Error_NoDevice: プリンタ未検出です。
- Error_IllegalParameter: パラメーターが範囲外です。
- Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・指定したビットイメージを印刷します。
- ・行モードの場合、印字開始位置は左端となります。ページモードの場合、startPage:メソッドで指定した領域内の左端となります。setOrientation:メソッドにより印刷方向が指定されている場合、印刷方向に対して左端となります。(例:印刷方向が下→上の場合、左→右時の下端が印字開始位置)
- ・*widthDots* および *heightDots* の設定範囲を以下に示します。

印刷モード	パラメーター	設定範囲
行モード	<i>widthDots</i>	印字幅(dot)
	<i>heightDots</i>	1～1023
ページモード 印刷方向が左→右または 右→左	<i>widthDots</i>	印字幅(dot)
	<i>heightDots</i>	1～1919
ページモード 印刷方向が下→上または 上→下	<i>widthDots</i>	FTP-62HWSL001:1～1920
	<i>heightDots</i>	印字幅(dot) - 1

- ・ページモードではビットイメージ展開だけで実際の印刷は行われません。
- ・startPage:メソッドで指定されたページモード領域をはみ出すデータはError_IllegalParameterを返し、印刷しません。
- ・ビットイメージのフォーマットを以下に示します。



- *data* のデータ数 == $\lceil \text{widthDots} / 8 \rceil \times \text{heightDots}$ となります。([]内は小数点以下繰上げ)
- PackBits 圧縮するとデータサイズが大きくなる場合、圧縮せずに送信します。
- 行モードの場合、setPrintSpeed:メソッドの設定によらず印刷速度は最大 10mm/秒固定となります。

(8) `printCharacterString:`

– `(int)printCharacterString:(NSString *)string;`
文字列を印刷します。

パラメーター

string: 印刷する文字列を指定します。(最大 4 × 1024 文字)

戻り値

`Error_Success`: 成功しました。

`Error_NoDevice`: プリンタ未検出です。

`Error_IllegalParameter`: パラメーターが範囲外です。

解説

- ・指定した文字列を印刷します。
- ・プリンタ印字幅を超える分のデータは次の行に折り返して印刷されます。ページモードの場合、指定された縦方向の領域をはみ出す文字列は印刷しません。
- ・`setLanguageOfCharacterString:` メソッドで指定した言語に対応するエンコードで *string* を変換してプリンタに送信します。変換できない文字やプリンタに搭載されていない文字は正常に印刷できません。
- ・*string* で指定された文字数が範囲外の場合、印刷せず `Error_IllegalParameter` を返します。

(9) printCustomerBarcode:size:

- (int)printCustomerBarcode:(NSData *)data size:(int)size;
カスタマバーコードを印刷します。

パラメーター

data: データを指定します。設定範囲 (7～20 バイト)

size: サイズを指定します。

size	バーコードのポイント数
CustomerBarcodeSize_8pt	8 pt
CustomerBarcodeSize_10pt	10 pt
CustomerBarcodeSize_11_5pt	11.5 pt

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・4 ステート 3 バー方式のカスタマバーコードを印刷します。
- ・ページモードではバーコード展開だけで実際の印刷は行われません。
- ・以下にパラメーター範囲を示します。

キャラクター	データ(16 進)	キャラクター	データ(16 進)	キャラクター	データ(16 進)
1	31	A	41	T	54
2	32	B	42	U	55
3	33	C	43	V	56
4	34	D	44	W	57
5	35	E	45	X	58
6	36	F	46	Y	59
7	37	G	47	Z	5A
8	38	H	48		
9	39	I	49		
0	30	J	4A		
–	2D	K	4B		
CC1	01	L	4C		
CC2	02	M	4D		
CC3	03	N	4E		
CC4	04	O	4F		
CC5	05	P	50		
CC6	06	Q	51		
CC7	07	R	52		
CC8	08	S	53		

- ・データフォーマットは、郵便番号 (7 桁の数字) に続く住所表示番号 (13 桁の英大文字、記号、制御コード) の最大 20 桁となります。英大文字は制御コード+数字に変換するため、1 文字あたり 2 桁でカウントします。データフォーマットに一致しない場合は Error_IllegalParameter を返します。

- ・データの桁数が 20 桁未満の場合は、自動で制御コード CC4 を付加します。データの桁数が 20 桁以上となった場合は、21 桁目以降は破棄します。データの 20 桁目が英大文字の場合、制御コードのみとなります。
- ・スタート/ストップコードおよびチェックディジットは、自動で付加します。
- ・展開されるバーコードサイズが印字領域を超える場合は Error_IllegalParameter を返します。

印字モード	角度	印刷方向	最大幅および高さ
行モード	0°	–	本プリンタでは印刷できません。
ページモード	0°	左→右	本プリンタでは印刷できません。
		右→左	
		上→下	幅: startPage メソッドの size.height
		下→上	高さ: startPage メソッドの size.width - 1
	180° 回転	左→右	本プリンタでは印刷できません。
		右→左	
		上→下	幅: startPage メソッドの size.height - 1
		下→上	高さ: startPage メソッドの size.width - 1
	右 90° 回転	左→右	幅: startPage メソッドの size.height
		右→左	
		上→下	本プリンタでは印刷できません。
		下→上	
	左 90° 回転	左→右	幅: startPage メソッドの size.height - 1
		右→左	
		上→下	本プリンタでは印刷できません。
		下→上	

- ・ページモードの場合、展開されるバーコード幅および高さが範囲内であれば Error_Success を返しますが、印字開始位置 (setRelativePosition:メソッドまたは setAbsolutePosition:メソッドで指定) によっては印刷されません。
- ・カスタムバーコードの展開には、上下左右に 2mm 以上のクワイエットゾーン (空白) を確保して配置して下さい。クワイエットゾーンが確保されないと正常に読み取れない可能性があります。

(10) `printGS1DataBar:dataBarType:module:numberOfSegments:`

– (int)`printGS1DataBar:(NSData *)data dataBarType:(int)gs1DataBarType module:(int)moduleSize
numberOfSegments:(int)segments;`
GS1 DataBar を印刷します。

パラメーター

data: データを指定します。

gs1DataBarType: GS1 DataBar のタイプを指定します。

<i>gs1DataBarType</i>	タイプ
GS1Databar_14	GS1 DataBar-14
GS1Databar_14_T	GS1 DataBar-14 Truncated
GS1Databar_14_S	GS1 DataBar-14 Stacked
GS1Databar_14_SO	GS1 DataBar-14 Stacked Omnidirectional
GS1Databar_L	GS1 DataBar Limited
GS1Databar_E	GS1 DataBar Expanded

moduleSize: ドット単位でモジュールサイズを指定します。設定範囲 (0～32)

2 または 3 の指定を推奨します。

segments: GS1 DataBar Expanded の一段あたりシンボルのセグメント数を指定します。
設定範囲 (2～22: 偶数のみ)

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・ページモードではバーコード展開だけで実際の印刷は行われません。

- ・以下にパラメーター範囲を示します。パラメーター範囲外ではスキャナーで正常に読み取りできない可能性がありますので、使用しないで下さい。

GS1 DataBar Expanded を除く GS1 DataBar								
	0	1	2	3	4	5	6	7
0				0				
1				1				
2				2				
3				3				
4				4				
5				5				
6				6				
7				7				
8				8				
9				9				
A								
B								
C								
D								
E								
F								

GS1 DataBar Expanded								
	0	1	2	3	4	5	6	7
0			SP	0		P		p
1			!	1	A	Q	a	q
2			“	2	B	R	b	r
3				3	C	S	c	s
4				4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			‘	7	G	W	g	w
8			(8	H	X	h	x
9)	9	I	Y	i	y
A			*	:	J	Z	j	z
B			+	;	K		k	
C			,	<	L		l	
D		FNC1	—	=	M		m	
E			.	>	N		n	
F			/	?	O	_	o	

- ・GS1 DataBar の個別仕様を以下に記載します。以下仕様のデータ長とは[*data length*]の戻り値です。dx とは *data* 内のバイトデータです。d1 が先頭バイトであり、以降 d2、d3、...、dx と続きます。

【GS1 DataBar-14】

- ・データ長 == 13 または 14 となります。
- ・データ長 == 13 となる場合、d1～d13 とチェックディジットをデータとする GS1 DataBar を印刷します。
- ・データ長 == 14 となる場合、14 番目のデータ d14 を除く d1～d13 とチェックディジットをデータとする GS1 DataBar を印刷します。
- ・チェックディジットは d1～d13 からモジュラス 10 で算出します。

【GS1 DataBar-14 Truncated】

- ・GS1 DataBar Truncated は GS1 DataBar-14 と高さのみ異なります。
- ・高さ以外の仕様は GS1 DataBar-14 と同様です。

【GS1 DataBar-14 Stacked】

- ・一行の分離パターンを挟んで上段と下段で構成されたシンボルです。
- ・シンボル構成以外の仕様は GS1 DataBar-14 と同様です。

【GS1 DataBar-14 Stacked Omnidirectional】

- ・三行の分離パターンを挟んで上段と下段で構成されたシンボルです。
- ・シンボル構成以外の仕様は GS1 DataBar-14 と同様です。

【GS1 DataBar Limited】

- ・データ長 == 13 または 14 となります。
- ・1 番目のデータ d1 が 0、1 以外である場合、該当 GS1 DataBar を印刷せず残りのデータ d は無視します。
- ・データ長 == 13 とした場合、d1～d13 とチェックディジットをデータとする GS1 DataBar を印刷します。

- データ長 == 14 とした場合、14 番目のデータ d14 を除く d1～d13 とチェックディジットをデータとする GS1 DataBar を印刷します。
 - チェックディジットは d1～d13 からモジュラス 10 で算出します。
- 【GS1 DataBar Expanded】
- データ長は可変であり、最大 74 数字または、41 英文字が入力可能です。ただし、データの配列により異なります。
 - GS1 DataBar Expanded はデータに応じて 4～22 個のセグメントシンボルで構成されています。
segment で指定した一段当たりセグメント数を越える GS1 DataBar Expanded を印刷する時、一段当たりセグメントを印刷した後、下段に分離パターンを印刷し、次のセグメントを印刷します。(GS1 DataBar Expanded Stacked)
 - GS1 DataBar Expanded Stacked は 2 段～11 段までシンボルを印刷することができます。
- *moduleSize* == 0 の場合、変更せず現在のモジュールサイズを使用します。
 - *segments* の値が奇数の場合、Error_IllegalParameter を返します。
 - GS1 DataBar Expanded を印刷する場合、*segments* == 22 なら GS1 DataBar Expanded(非 Stacked 型)、*segments* == 2 なら GS1 DataBar Expanded Stacked (Stacked 型)を常に印刷します。
 - GS1 DataBar Expanded 以外を指定した場合、*segments* は無視されます。

- ・展開される GS1 DataBar 幅が最大幅を超える場合は Error_IllegalParameter を返します。

印字モード	角度	印刷方向	最大幅および高さ
行モード	0°	–	印字幅(dot)
ページモード	0°	左→右	幅: startPage メソッドの size.width
		右→左	
		上→下	高さ: startPage メソッドの size.height – 1
		上→下	幅: startPage メソッドの size.height
		下→上	高さ: startPage メソッドの size.width – 1
		左→右	幅: startPage メソッドの size.width – 1
		右→左	高さ: startPage メソッドの size.height – 1
	180° 回転	左→右	幅: startPage メソッドの size.width – 1
		右→左	高さ: startPage メソッドの size.height – 1
		上→下	幅: startPage メソッドの size.height – 1
		下→上	高さ: startPage メソッドの size.width – 1
	右 90° 回転	左→右	幅: startPage メソッドの size.height
		右→左	高さ: startPage メソッドの size.width – 1
		上→下	幅: startPage メソッドの size.width
		下→上	高さ: startPage メソッドの size.height – 1
	左 90° 回転	左→右	幅: startPage メソッドの size.height – 1
		右→左	高さ: startPage メソッドの size.width – 1
		上→下	幅: startPage メソッドの size.width – 1
		下→上	高さ: startPage メソッドの size.height – 1

- ・ページモードの場合、展開されるバーコード幅および高さが範囲内であれば Error_Success を返しますが、印字開始位置 (setRelativePosition:メソッドまたは setAbsolutePosition:メソッドで指定) によっては印刷されません。
- ・角度および印刷方向に関わらず、バーコード幅が 864 ドットを超える場合、Error_IllegalParameter を返します。

(11) printMaxiCode:mode:number:count:symbolCount:

- (int)printMaxiCode:(NSData *)data mode:(int)symbolMode number:(int)symbolNumber
count:(int)symbolCount;
マキシコードを印刷します。

パラメーター

data: データを指定します。

symbolMode: シンボルモードを指定します。

symbolMode	モード
MaxiCodeMode_2	モード 2
MaxiCodeMode_3	モード 3
MaxiCodeMode_4	モード 4
MaxiCodeMode_5	モード 5
MaxiCodeMode_6	モード 6

symbolNumber: シンボルナンバーを指定します。設定範囲(1~8)

symbolCount: シンボルカウントを指定します。設定範囲(1~8)

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・マキシコードを印刷します。
- ・ページモードではバーコード展開だけで実際の印刷は行われません。
- ・以下にパラメーター範囲を示します。パラメーター範囲外ではスキャナーで正常に読み取りできない可能性がありますので、使用しないで下さい。

モード	有効なパラメーター範囲	データ数
モード 2	Primary Message 数字“0~9” Secondary Message 0x00 ~ 0xFF	18~85
モード 3	Primary Message 郵便コード “0~9”, “A~Z”, 空 白, “ “#%&'()*+,-./ ” 国名コード サービスコード 数字(0~9) Secondary Message 0x00 ~ 0xFF	18~85
モード 4、モード 5	0x00 ~ 0xFF	1~80
モード 6	0x00 ~ 0xFF	1~80

- ・ページモードの場合、展開されるバーコード幅および高さが範囲内であれば Error_Success を返しますが、印字開始位置(setRelativePosition:メソッドまたは setAbsolutePosition:メソッドで指定)によっては印刷されません。
- ・二次元コードの展開には、各二次元コードで規定されているクワイエットゾーン(マージン)を確

保して配置して下さい。他のバーコードやイメージの展開で重なり、クワイエットゾーンが確保されないと二次元コードが正常に読み取れない可能性があります。

(12) printPage:

– (int)printPage:(int)mode;
印刷を開始します

パラメーター

mode: 印刷後のページモードの動作を選択します。

mode	動作
EndPageMode	印刷後にページモード領域内のデータを破棄し、行モードに移行します。
PageDataKeep	印刷後もページモード領域のデータを保持したままページモードを継続します。
PageDataClear	印刷後にページモード領域内のデータを破棄し、ページモードを継続します。
CancelPageMode	印刷せずにページモードを終了し、行モードに移行します。

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。
Error_IllegalParameter: パラメーターが範囲外です。
Error_NotPageMode: ページモードではありません。

解説

- ・プリンタに展開したページモード領域のデータの印刷を開始します。
- ・行モードで実行した場合、Error_NotPageMode となります。
- ・setPaperType:feedBeforePrintPage:feedAfterMarkDetection:autoFeedLength:メソッドにて、ラベルが設定されている場合、mode == EndPageMode および mode == PageDataClear 設定時、印刷後にマーク検出を行います。
- ・印刷速度は印刷内容によって異なります。詳細は setPrintSpeed:メソッドの解説を確認して下さい。

- (13) `printPDF417Code:columnsOrSteps:width:height:errorLevel:`
- (int)printPDF417Code:(NSData *)*data* columnsOrSteps:(int)*columnsOrSteps*
width:(int)*moduleWidth* height:(int)*moduleHeight* errorLevel:(int)*errorCorrectionLevel*,
PDF417 を印刷します。

パラメーター

data: データを指定します。

columnsOrSteps: カラム数または段数を指定します。

カラム数 設定範囲(1~30)

段数 設定範囲($3 + N \sim 90 + N$: $N == 128$)

moduleWidth: ドット単位でモジュール幅を指定します。設定範囲(1~30)

moduleHeight: ドット単位でモジュール高さを指定します。設定範囲(1~30)

errorCorrectionLevel: エラー訂正レベルを指定します。設定範囲(0~8)

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・PDF417 を印刷します。
- ・ページモードではデータ展開だけで実際の印刷は行われません。
- ・印刷可能条件を以下に示します。印刷可能条件を満たさないデータの場合は本メソッド内で判定し、Error_IllegalParameter を返します。

有効なパラメーター範囲	データ数
0x00~0xFF	1~2710

- ・二次元コードの展開には、各二次元コードで規定されているクワイエットゾーン(マージン)を確保して配置して下さい。他のバーコードやイメージの展開で重なり、クワイエットゾーンが確保されないと二次元コードが正常に読み取れない可能性があります。

- ・展開される PDF417 幅が最大幅または高さを超える場合は `Error_IllegalParameter` を返します。

印字モード	角度	印刷方向	最大幅および高さ
行モード	0°	—	幅: 印字幅(dot) 高さ: 579
ページモード	0°	左→右	幅: startPage メソッドの size.width
		右→左	高さ: startPage メソッドの size.height - 1
		上→下	幅: startPage メソッドの size.height
		下→上	高さ: startPage メソッドの size.width - 1
	180° 回転	左→右	幅: startPage メソッドの size.width - 1
		右→左	高さ: startPage メソッドの size.height - 1
		上→下	幅: startPage メソッドの size.height - 1
		下→上	高さ: startPage メソッドの size.width - 1
	右 90° 回転	左→右	幅: startPage メソッドの size.height
		右→左	高さ: startPage メソッドの size.width - 1
		上→下	幅: startPage メソッドの size.width
		下→上	高さ: startPage メソッドの size.height - 1
	左 90° 回転	左→右	幅: startPage メソッドの size.height - 1
		右→左	高さ: startPage メソッドの size.width - 1
		上→下	幅: startPage メソッドの size.width - 1
		下→上	高さ: startPage メソッドの size.height - 1

- ・ページモードの場合、展開されるバーコード幅および高さが範囲内であれば `Error_Success` を返しますが、印字開始位置 (`setRelativePosition:メソッド` または `setAbsolutePosition:メソッド` で指定) によっては印刷されません。
- ・角度および印刷方向に関わらず、バーコード幅が 864 ドットを超える場合、`Error_IllegalParameter` を返します。

(14) printPicture:dithering:width:height:

- (int)printPicture:(UIImage *)*image* dithering:(int)*dithering* width:(int)*widthDots* height:(int)*heightDots*;
画像を印刷します。

パラメーター

image: 画像ファイル等の UIImage クラスオブジェクトを指定します。

dithering: ディザリングパターンを選択します。

<i>dithering</i>	ディザリングパターン
Dither_Bayer_2x2	Bayer 型 2x2
Dither_Bayer_4x4	Bayer 型 4x4
Dither_Bayer_8x8	Bayer 型 8x8
Dither_Bayer_16x16	Bayer 型 16x16
Dither_Clustared_2x2	渦巻き型 2x2
Dither_Clustared_4x4	渦巻き型 4x4
Dither_Clustared_8x8	渦巻き型 8x8
Dither_NoDithering	ディザリングなし

widthDots: 出力サイズの幅をドット単位で指定します。設定範囲(解説参照)

heightDots: 出力サイズの高さをドット単位で指定します。設定範囲(解説参照)

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_PageModeOutSide: ページモードにおいて領域外となりました。

Error_IllegalImage: 非対応のイメージです。

解説

- ・指定した画像を印刷します。

- ・*widthDots* および *heightDots* の設定範囲を以下に示します。

印刷モード	パラメーター	設定範囲
行モード	<i>widthDots</i>	0～印字幅(dot)
	<i>heightDots</i>	0～1920
ページモード 印刷方向が左→右または 左→右	<i>widthDots</i>	0～印字幅(dot)
	<i>heightDots</i>	0～1919
ページモード 印刷方向が下→上または 上→下	<i>widthDots</i>	FTP-62HWSL000:0～1919
	<i>heightDots</i>	0～印字幅(dot) - 1

- ・上記設定範囲内の場合でも、startPage:メソッドによるページモード領域をはみ出すデータの場合、Error_PageModeOutSide を返します。
- ・*widthDots* および *heightDots* に0を指定すると、アスペクト比を変更せず画像の1ピクセル = プリンタの1ドットとして印刷します。指定画像のサイズが印刷可能領域を超えている場合、アスペクト比を変更せずに縮小して印刷します。
- ・*widthDots* (*heightDots*)を0且つ *heightDots* (*widthDots*)を設定範囲内に指定した場合、幅(高さ)の最大値を超えない範囲で、アスペクト比変更が最小となるように幅(高さ)を変更します。
- ・*widthDots* または *heightDots* の一方でも設定範囲外の場合、Error_IllegalParameter を返します。
- ・印刷可能な画像形式は、JPEG、GIF、TIFF、BMP、PNG です。ただし、1ピクセルあたりのビット

数が 24/32 ビット以外の UIImage クラスオブジェクトは印刷できません。また、画像の圧縮方法等によっては印刷できない場合があります。本メソッドで印刷できない形式の場合、`Error_IllegalImage` を返します。

- ・行モードの場合、`setPrintSpeed:`メソッドの設定によらず印刷速度は最大 10mm/秒固定となります。
- ・`dither == Dither_NoDithering` を指定した場合、ディザリング処理を行わずに印刷します。*image* の各ピクセルが `RGB=(0, 0, 0)` の場合黒、それ以外は白で印刷します。拡大、縮小を行うと色情報に変化する場合があるため、ドットバイドットとなるサイズを指定して実行することを推奨します。
- ・ページモードではイメージ展開だけで実際の印刷は行われません。
- ・行モードの場合、印字開始位置は左端となります。
- ・ページモードの場合、`startPage:`メソッドで指定した領域内の左端となります。`setOrientation:`メソッドにより印刷方向が指定されている場合、印刷方向に対して左端となります。(例: 左 90° 回転の場合、回転なし時の下端が印字開始位置)
- ・ページモードの場合、展開される幅および高さが範囲内であれば `Error_Success` を返しますが、印字開始位置 (`setRelativePosition:`メソッドまたは `setAbsolutePosition:`メソッドで指定) によっては印刷されません。

(15) `printQRCode:magnification:errorLevel:concatenation:code:division:parity:masking:`
- (int)`printQRCode:(NSData *)data magnification:(int)moduleMagnificaton errorLevel:`
 (int)`errorCorrectionLevel concatenation:(BOOL)qrConcatenation code:(int)codeNumber`
 division:(int)`divisionNumber parity:(unsigned char)dataParity masking:(int)masking;`
QRコードを印刷します。

パラメーター

`data`: データを指定します。

`moduleMagnificaton`: モジュール倍率を指定します。

<code>moduleMagnificaton</code>	モジュール倍率指定
<code>QRModule_3x3</code>	3 倍 (1 モジュール = 3 × 3 ドット)
<code>QRModule_4x4</code>	4 倍 (1 モジュール = 4 × 4 ドット)

`errorCorrectionLevel`: 誤り訂正レベルを指定します。

<code>errorCorrectionLevel</code>	誤り訂正レベル
<code>QRLevel_H</code>	超高信頼度レベル
<code>QRLevel_Q</code>	高信頼度レベル
<code>QRLevel_M</code>	標準レベル
<code>QRLevel_L</code>	高密度レベル

`qrConcatenation`: 連結モードの指定をします。

NO 通常モード

YES 連結モード

`codeNumber`: コード番号を指定します。設定範囲 (1 ~ 16)

`divisionNumber`: 分割数を指定します。設定範囲 (1 ~ 16)

`dataParity`: 全データ (分割前のデータ) をバイト単位で xor を取った値を、0x00 ~ 0xFF の値で指定してください。連結モードが指定されていない場合は無視されます。

`masking`: マスキングを指定します。

<code>masking</code>	マスキング
0 ~ 7	マスク番号
<code>Masking_None</code>	マスクなし
<code>Masking_Auto</code>	マスク自動選択

戻り値

`Error_Success`: 成功しました。

`Error_NoDevice`: プリンタ未検出です。

`Error_IllegalParameter`: パラメーターが範囲外です。

`Error_PageModeOutSide`: ページモードにおいて領域外となりました。

解説

- ・QRコードを印刷します。
- ・ページモードではデータ展開だけで実際の印刷は行われません。
- ・連結モードが指定されていない場合、`codeNumber`、`divisionNumber`、`dataParity` の値は無視されます。

- ・以下にパラメーター範囲を示します。パラメーター範囲外ではスキャナーで正常に読み取りできない可能性がありますので、使用しないで下さい。

キャラクタモード	モード設定	有効なデータ範囲
数字モード	0x10	数字 “0～9”
英数字モード	0x11	“0～9”, “A～Z”, 空白, “\$, %, *, +, -, ., /, :”
8ビットバイトモード	0x12	0x00 ～ 0xFF
シフト JIS 漢字モード	0x13	第 1 バイト: 0x81 ～ 0x9F, 0xE0 ～ 0xEA 第 2 バイト: 0x40 ～ 0x7E, 0x80 ～ 0xFC

- ・ *data* の第 1 バイト目にモード設定(0x10～0x13)、以降にデータを指定します。
- ・ 8ビットバイトモードの場合は、モード設定 0x12 の後に 2 バイトの BCD コードでデータバイト数を付加して下さい。
- ・ 二次元コードの展開には、各二次元コードで規定されているクワイエットゾーン(マージン)を確保して配置して下さい。他のバーコードやイメージの展開で重なり、クワイエットゾーンが確保されないと二次元コードが正常に読み取れない可能性があります。
- ・ 角度および印刷方向に関わらず、バーコード幅が 864ドットを超える場合、Error_IllegalParameter を返します。

・展開される QR コード幅が最大幅または高さを超える場合は Error_IllegalParameter を返します。

印字モード	角度	印刷方向	最大幅および高さ
行モード	0°	－	幅、高さ: 印字幅(dot)
ページモード	0°	左→右	幅: startPage メソッドの size.width
		右→左	
		上→下	高さ: startPage メソッドの size.height - 1
		下→上	
	180° 回転	左→右	幅: startPage メソッドの size.width - 1
		右→左	
		上→下	高さ: startPage メソッドの size.height - 1
		下→上	
	右 90° 回転	左→右	幅: startPage メソッドの size.height
		右→左	
		上→下	高さ: startPage メソッドの size.width - 1
		下→上	
	左 90° 回転	左→右	幅: startPage メソッドの size.height - 1
		右→左	
		上→下	高さ: startPage メソッドの size.width - 1
		下→上	

・ページモードの場合、展開されるバーコード幅および高さが範囲内であれば Error_Success を返しますが、印字開始位置 (setRelativePosition:メソッドまたは setAbsolutePosition:メソッドで指定) によっては印刷されません。

(16) printRegisteredImage:mode:

- (int)printRegisteredImage:(int)*Number* mode:(int)*printMode*;
登録イメージを印刷します。

パラメーター

Number: 印刷するイメージの ID 番号を指定します。設定範囲(1～255)

printMode: 印刷モードを指定します。

<i>printMode</i>	印刷モード
PrintSize_Normal	指定した登録イメージデータを登録したサイズで印字します。
PrintSize_Wx2	指定した登録イメージデータを横 2 倍に拡大して印字します。
PrintSize_Hx2	指定した登録イメージデータを縦 2 倍に拡大して印字します。
PrintSize_X4	指定した登録イメージデータを縦横 2 倍に拡大して印字します。

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・未登録の *Number* を指定した場合、Error_Success を返しますが印刷またはデータ展開されません。
- ・ページモードではデータ展開だけで実際の印刷は行われません。
- ・イメージデータが印字領域をはみ出した場合、はみ出した部分は無視されます。
- ・印刷されるイメージデータの高さが 256 ドットライン以上となる場合には、255 ドットライン分のデータを印刷します。

(17) startPage:

– (int)startPage:(CGRect)rect;
ページモードを開始します。

パラメーター

rect: ページモード領域をドット単位で以下のように指定します。

origin.x 横方向始点を指定します。設定範囲FTP-62HWSL000: (0～383)
origin.y 縦方向始点を指定します。設定範囲FTP-62HWSL000: (0～1919)
size.width 横方向長さを指定します。設定範囲 FTP-62HWSL000: (1～384)
size.height 縦方向長さを指定します。設定範囲 FTP-62HWSL000: (1～1920)

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。
Error_IllegalParameter: パラメーターが範囲外です。
Error_AlreadyPageMode: 既にページモードです。

解説

- ・領域を指定し、ページモードを開始します。
- ・*rect* の各パラメーターにおいて、小数点以下の値は切り捨てられます。
- ・(origin.x + size.width) が印字幅(dot)を超える場合、または (origin.y + size.height) が最大高さを超える場合、Error_IllegalParameter を返します。
- ・ページモード開始時の展開基準位置はページの左上 (origin.x および origin.y の位置) に設定されています。ただし、ページモード上端の 1 ドットラインは印字不可能となるため、dotFeedPaper: メソッド等を用いて次のドットラインから印字を開始してください。文字は文字の左下を展開基準位置に合わせて展開されますので、そのまま展開を開始すると展開領域からはみだします。開始時は 1 行改行等を行って、文字フォントが展開領域内に納まるようにしてください。
- ・ページモード中はデータ展開のみで印刷動作が行われません。printPage: メソッドを実行することにより印刷が行われます。
- ・ページモード中に本メソッドを実行した場合、Error_AlreadyPageMode が返ります。

(18) registBitimage:width:height:data:

– (int)registBitimage:(int)*imageID* width:(int)*widthBytes* height:(int)*heightDots* data:(NSData *)*data*;
ビットイメージデータからイメージ登録を行います。

パラメーター

imageID: 登録するイメージの ID 番号を指定します。設定範囲(1～255)

widthBytes: 幅をバイト単位で指定します。設定範囲(1～印字幅(mm))

heightDots: 高さをドットライン単位で指定します。設定範囲(1～512)

data: データを指定します。

戻り値

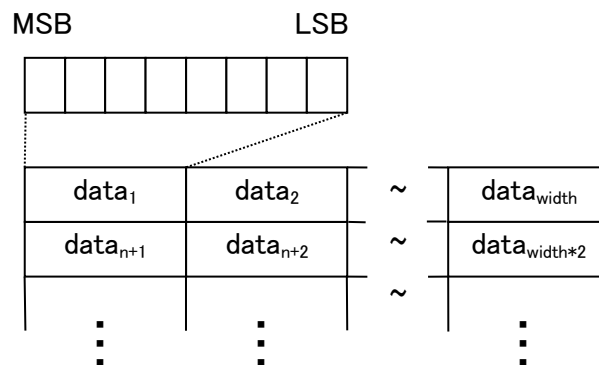
Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・ビットイメージデータからイメージ登録を行います。
- ・*data* のサイズと (*widthBytes* × *heightDots*) の値が異なる場合、Error_IllegalParameter を返します。
- ・既に登録されている *imageID* を指定した場合、上書きされます。
- ・*data* == nil または、[*data* length] の戻り値が 0 の場合、指定された *imageID* が既に登録されているなら、該当データを削除します。登録されていない *imageID* の場合、何もしません。
- ・本プリンタの全メモリ容量は 192 × 1024 バイトです。この容量を超えるデータを登録することはできませんのでご注意ください。
- ・イメージ登録のフォーマットを以下に示します。



データ数([*data* length]の戻り値) == *widthBytes* × *heightDots* となります。

(19) registPicture:image:dithering:width:height:

- (int)registPicture:(int)*imageID* image:(UIImage *)*image* dithering:(int)*dithering* width:(int)*widthDots* height:(int)*heightDots*;
画像ファイルを登録します。

パラメーター

imageID: 登録するイメージの ID 番号を指定します。設定範囲(1~255)

image: 画像ファイル等の UIImage クラスオブジェクトを指定します。

dithering: ディザリングパターンを選択します。

<i>dithering</i>	ディザリングパターン
Dither_Bayer_2x2	Bayer 型 2x2
Dither_Bayer_4x4	Bayer 型 4x4
Dither_Bayer_8x8	Bayer 型 8x8
Dither_Bayer_16x16	Bayer 型 16x16
Dither_Clustared_2x2	渦巻き型 2x2
Dither_Clustared_4x4	渦巻き型 4x4
Dither_Clustared_8x8	渦巻き型 8x8
Dither_NoDithering	ディザリングなし

widthDots: 幅をドット単位で指定します。設定範囲(0~印字幅(dot))

heightDots: 高さをドット単位で指定します。設定範囲(0~512)

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

Error_IllegalImage: 非対応のイメージです。

解説

- 指定した UIImage クラスオブジェクトを登録します。
- widthDots* および *heightDots* に 0 を指定すると、アスペクト比を変更せず画像の 1 ピクセル＝プリンタの 1 ドットとして印刷します。指定画像のサイズが設定範囲を超えている場合、アスペクト比を変更せずに縮小して印刷します。
- widthDots*(*heightDots*)を 0 且つ *heightDots*(*widthDots*)を設定範囲内に指定した場合、幅(高さ)の最大値を超えない範囲で、アスペクト比変更が最小となるように幅(高さ)を変更します。
- widthDots* または *heightDots* の一方でも設定範囲外の場合、Error_IllegalParameter を返します。
- 登録可能な画像形式は、JPEG、GIF、TIFF、BMP、PNG です。ただし、1 ピクセルあたりの bit 数が 24/32bit 以外の画像ファイルは登録できません。ファイルの圧縮方法等によっては登録できない場合があります。本メソッドで印刷できないファイルの場合、Error_IllegalFile を返します。
- 幅はドット単位で指定可能ですが、プリンタに登録されるのはバイト(8ドット)単位です。そのため、右端のバイトが 1 バイト(8ドット)に満たない場合、足りないドットを白で満たします。
- dither* == Dither_NoDithering を指定した場合、ディザリング処理を行いません。*image* の各ピクセルが RGB=(0, 0, 0)の場合黒、それ以外は白で登録します。拡大、縮小を行うと色情報が変化する場合がありますため、ドットバイドットとなるサイズを指定して実行することを推奨します。
- 既に登録されている *imageID* を指定した場合、上書きされます。
- 本プリンタの全メモリ容量は 192 × 1024 バイトです。この容量を超えるデータを登録することはできませんのでご注意ください。

(20) setAbsolutePosition:

- (int)setAbsolutePosition:(CGPoint)*point*;
展開位置を絶対値指定します。

パラメーター

point: 展開位置をドット単位で以下のように指定します。

- x 横方向データ展開開始の絶対位置を指定します。設定範囲(解説参照)
- y 縦方向データ展開開始の絶対位置を指定します。設定範囲(解説参照)

戻り値

- Error_Success: 成功しました。
- Error_NoDevice: プリンタ未検出です。
- Error_NotPageMode: ページモードではありません。
- Error_IllegalParameter: パラメーターが範囲外です。
- Error_PageModeOutSide: ページモードにおいて領域外となりました。

解説

- ・展開位置を絶対値指定します。

印刷モード	パラメーター	設定範囲
印刷方向が左→右または 左→右	x	0～印字幅(dot) – 1
	y	FTP-62HWSL000:0～1919
印刷方向が下→上または 上→下	x	FTP-62HWSL000:0～1919
	y	0～印字幅(dot) – 1

- ・基準となる始点は、setOrientation:メソッドにより決定します。印刷方向が左→右の場合ページモードの左上端です。
- ・設定範囲内であってもページモード領域を超える位置指定をした場合、Error_PageModeOutSide が返ります。
- ・*point* の各パラメーターにおいて、小数点以下の値は切り捨てられます。
- ・本メソッドはページモード中のみ有効となり、行モードでは Error_NotPageMode を返します。

- (21) `setCharacterStyle:kanjiSize:fontHSize:fontVSize:bold:bwReverse:underline:space:leftOfKanji:rightOfKanji:`
`leftOfKanji:rightOfKanji:`
 – (int)`setCharacterStyle:(int)size kanjiSize:(int)kanjiSize fontHSize:(int)fontHSize fontVSize:(int)fontVSize bold:(bool)bold bwReverse:(BOOL)blackWhiteReverse underline:(int)underline space:(int)space leftOfKanji:(int)leftOfKanji rightOfKanji:(int)rightOfKanji;`
 文字修飾を設定します。

パラメーター

size: ANK 文字のフォントサイズを指定します。

size	ANK フォントサイズ指定
Font_8x16	8 × 16 ドット
Font_12x24	12 × 24 ドット
Font_16x16	16 × 16 ドット
Font_24x24	24 × 24 ドット

kanjiSize: 漢字のフォントサイズを指定します。

kanjiSize	漢字フォントサイズ指定
Font_16x16	16 × 16 ドット
Font_24x24	24 × 24 ドット

fontHSize: 文字サイズの横方向の倍率指定をします。

fontVSize: 文字サイズの縦方向の倍率を指定します。

<i>fontHSize</i> <i>fontVSize</i>	倍率指定
FontDSize_05	0.5 倍
FontDSize_10	1 倍
FontDSize_15	1.5 倍
FontDSize_20	2 倍
FontDSize_25	2.5 倍
FontDSize_30	3 倍
FontDSize_35	3.5 倍
FontDSize_40	4 倍

bold: 強調印字の指定を行います。

本プリンタは強調印字に対応していません。NO を指定してください。

blackWhiteReverse: 白黒反転印字の指定を行います。

YES 白黒反転印字を行います。

NO 白黒反転印字を解除します。

underline: アンダーラインの指定を行います。

underline	アンダーライン指定
UnderLine_None	アンダーラインなし
UnderLine_1Dot	1 ドットのアンダーライン
UnderLine_2Dot	2 ドットのアンダーライン

space: ANK 文字の文字間スペースの指定を行います。漢字には影響しません。

設定範囲 (0~32)

leftOfKanji: 漢字の左スペースの指定を行います。

本プリンタは漢字の左スペース指定に対応していません。0 を指定して下さい。

rightOfKanji: 漢字の右スペースの指定を行います。4ドット単位で指定して下さい。
設定範囲(0~32)

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。
Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・文字修飾を設定します。
- ・*space* による文字間スペース設定は文字の右側に設定されます。
- ・文字間スペースおよび漢字の左右スペースは行モード、ページモードでそれぞれ別に設定されます
- ・1文字の占有ドット数は横方向の実印字ドット数に文字間スペースを加算した値が設定されます。よって、現在の文字位置と右マージンの間に前記占有値が確保できない場合には、印字改行後次行左端にデータ展開位置が設定されます。
- ・文字拡大指定時には文字間スペース量も拡大されます。
- ・*fontHSize* および *fontVSize* で FontDSize_05 を指定した場合フォントの一部を間引くため、文字が変形する場合があります。
- ・*rightOfKanji* は 4 ドット単位で指定してください。端数は切り捨てられます。
- ・*setRotation*: メソッドで Rotation_None 以外を指定時、アンダーラインなし固定となります。再度アンダーラインありを指定する場合、*setRotation*: メソッドで Rotation_None を指定して下さい。

(22) setInternationalCharCode:country:

- (int)setInternationalCharCode:(int)*characterCodeTable* country:(int)*country*;
文字コードテーブルおよび国際文字を指定します。

パラメーター

characterCodeTable: 文字コードテーブルを指定します。

<i>characterCodeTable</i>	文字コードテーブル指定
CharacterCode_Katakana	カタカナ
CharacterCode_ExpandedGraphics	拡張グラフィックス

country: 国際文字を指定します。設定範囲 (0～13)

<i>country</i>	国際文字指定
CharacterSet_USA	アメリカ
CharacterSet_French	フランス
CharacterSet_German	ドイツ
CharacterSet_UK	イギリス
CharacterSet_Denmark	デンマーク
CharacterSet_Sweden	スウェーデン
CharacterSet_Italy	イタリア
CharacterSet_Spain	スペイン
CharacterSet_Japan	日本
CharacterSet_Norway	ノルウェー
CharacterSet_Denmark_2	デンマーク 2
CharacterSet_Spain_2	スペイン 2
CharacterSet_Ratin_America	ラテンアメリカ
CharacterSet_Japan_2	日本 2

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・文字コードテーブルおよび国際文字を指定します。

(23) `setLineFeedAmount:page:`

- `(int)setLineFeedAmount:(int)dotLines page:(int)page;`
改行量および 1 ページの行数を指定します。

パラメーター

`dotLines`: 改行量をドットライン単位で指定します。設定範囲 (0~255)

`page`: 1 ページの行数を指定します。設定範囲 (0~127)

戻り値

`Error_Success`: 成功しました。

`Error_NoDevice`: プリンタ未検出です。

`Error_IllegalParameter`: パラメーターが範囲外です。

解説

- ・改行量および行モードでの 1 ページの行数を指定します。
- ・ $(dotLines \times page)$ が 1 ページの長さとなります。ページモードでの 1 ページの長さは `startPage:` メソッドで指定します。
- ・行モードの場合、現在指定している文字種の高さ分の改行は最低でも行います。印刷なし改行の場合には指定された改行量だけ改行します。例えば文字高さが 24 ドットラインの文字種を指定しているときに改行量を 10 ドットラインに設定した場合、印刷あり改行では 24 ドットライン、印刷なし改行では 10 ドットラインの改行を行います。ページモードの場合、常に指定された改行量だけ改行を行います。

(24) setOrientation:

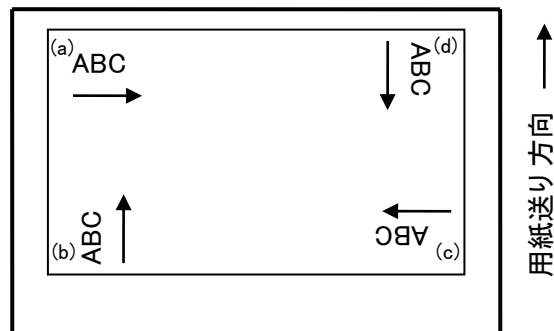
- (int)setOrientation:(int)orientation,
印刷方向を指定します。

パラメーター

orientation: 印刷方向を指定します。

本メソッド実行後のデータ展開始点は startPage:メソッドで設定した印字領域内で、以下のよう
に設定されます。

orientation	印刷方向	展開始点
Orientation_None	左 → 右	左上[下図(a)]
Orientation_L90	下 → 上	左下[下図(b)]
Orientation_180	右 → 左	右下[下図(c)]
Orientation_R90	上 → 下	右上[下図(d)]



戻り値

- Error_Success: 成功しました。
- Error_NoDevice: プリンタ未検出です。
- Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・印刷方向を指定します。
- ・印刷方向が下→上、または上→下の場合、startPage:メソッドの size.width の値が高さ、size.height の値が幅となります。
- ・印刷方向の指定はページモードでのみ有効です。行モードでは内部的な設定値のみが変更されます。
- ・ページモード領域にデータが展開されている場合、領域のデータを印刷後に本メソッドが実行されます。その際、ページモード領域のデータは破棄されます。データが展開されていない場合、直ちに本メソッドが実行されます。

- (25) `setPaperType:feedBeforePrintPage:feedAfterMarkDetection:autoFeedLength:`
- (int)setPaperType:(int)*paperType* feedBeforePrintPage:(int)*feedBeforePrintPage*
feedAfterMarkDetection:(int)*feedAfterMarkDetection* autoFeedLength:(int)*autoFeedLength*;
用紙モードおよびマーク検出頭出し量を指定します。

パラメーター

paperType: 用紙種類を指定します。

<i>paperType</i>	用紙種類
PaperType_ContinuousPaper	連続用紙
PaperType_Label	ラベル

feedBeforePrintPage: `printPage`: メソッドによる印刷開始前の用紙送り量をドットライン単位で指定します。設定範囲(-24~255)

feedAfterMarkDetection: マーク検出頭出し量をドットライン単位で指定します。
設定範囲(0~255)

autoFeedLength: 本プリンタでは使用されません。0を指定して下さい。

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・用紙モードおよび各種用紙送り量を指定します。
- ・ラベルを指定した場合、用紙セット時にマークを検出するまで最大で約 203mm の用紙送りを行います。
- ・*feedBeforePrintPage* > 0 の場合、順方向に用紙送りします。*feedBeforePrintPage* < 0 の場合、逆方向に用紙送りします。

(26) setPrintSpeed:

- (int)setPrintSpeed:(int)*speed*;
印刷速度を指定します。

パラメーター

speed: 印刷速度を mm/sec 単位で指定します。設定範囲(5～120)

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。
Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・印刷速度を指定します。
- ・本メソッドは最高速度の指定であるため、setPrintQuality:energy:halfPitch:メソッドや印刷内容等によっては速度が低下します。
- ・ページモードで二次元コードやバーコードを含む印刷を行う場合、回転指定や設定内容によっては *speed* 設定を超えない範囲で印刷速度の制限を行います。

印刷速度が 30mm/sec に制限される場合

- ・ GS1DataBar(スタック型)
- ・ カスタマバーコード
- ・ マキシコード
- ・ PDF417
- ・ モジュール倍率 3 倍の QR コード
- ・ setRotation:メソッドや setOrientation:メソッド指定時、用紙送り方向に対しバーが垂直となるバーコード

印刷速度が 80mm/sec に制限される場合

- ・ モジュール倍率 4 倍の QR コード

- ・ 同じページ内に印刷速度が制限される二次元コードやバーコードが複数含まれる場合は、最も低い速度に制限されます。

(27) setRelativePosition:

- (int)setRelativePosition:(CGPoint)point;
展開位置を相対値指定します。

パラメーター

point: 展開位置をドット単位で以下のように指定します。

- x 横方向データ展開開始の相対位置を指定します。設定範囲(解説参照)
- y 縦方向データ展開開始の相対位置を指定します。設定範囲(解説参照)

戻り値

- Error_Success: 成功しました。
- Error_NoDevice: プリンタ未検出です。
- Error_IllegalParameter: パラメーターが範囲外です。
- Error_PageModeOutSide: ページモードにおいて領域外となりました。
- Error_NotPageMode: ページモードではありません。

解説

- ・展開位置を現在の位置から相対値指定します。

印刷モード	パラメーター	設定範囲
印刷方向が左→右または 左→右	x	FTP-62HWSL000: -383～383
	y	FTP-62HWSL000: -1919～1919
印刷方向が下→上または 上→下	x	FTP-62HWSL000: -1919～1919
	y	FTP-62HWSL000: -383～383

- ・設定範囲内であってもページモード領域を超える指定をした場合、Error_Success が返りますが位置指定は無視されます。
- ・point の各パラメーターにおいて、小数点以下の値は切り捨てられます。
- ・本メソッドはページモード中のみ有効となり、行モードでは Error_NotPageMode を返します。

(28) setRotation:

- (int)setRotation:(int)rotation;
文字列およびバーコードの回転角度を指定します。

パラメーター

rotation: 回転角度を指定します。

rotation	回転角度
Rotation_None	回転なし
Rotation_L90	左 90° 回転
Rotation_180	180° 回転
Rotation_R90	右 90° 回転

戻り値

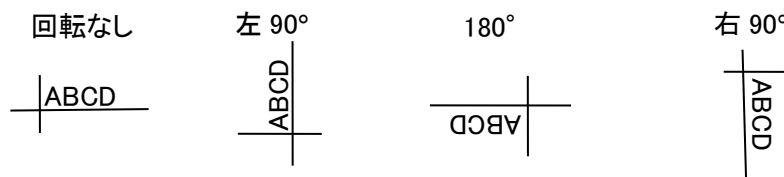
- Error_Success: 成功しました。
- Error_NoDevice: プリンタ未検出です。
- Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・回転角度の指定はページモードでのみ有効です。行モードでは内部的な設定値のみが変更されます。
- ・本機能が有効となるメソッドは下記のとおりです。

printBarcode:data:height:narrow:wide:hri:
printCharacterString:
printCustomerBarcode:size:
printGS1DataBar:dataBarType:module:numberOfSegments:
printMaxiCode:mode:number:count:symbolCount:
printPDF417Code:columnsOrSteps:width:height:errorLevel:rotate:
printQRCode:magnification:errorLevel:concatenation:code:division:parity:masking:
printRegisteredImage:mode:

- ・基点は文字列やバーコードの左下端で、回転角度指定時は以下の位置になります



- ・setOrientation:メソッドによる印刷方向指定に対し、相対的に回転されます。
- ・setCharacterStyle:...メソッドにて指定した文字間スペース、漢字の左右スペースは回転されません。アンダーラインは印刷されません。

(29) `setAutoPowerOffTime:`

– `(int)setAutoPowerOffTime:(int)minutes;`

オートパワーオフ時間を指定します。

パラメーター

minutes: オートパワーオフ時間を分単位で指定します。設定範囲 (0～120)

戻り値

`Error_Success`: 成功しました。

`Error_NoDevice`: プリンタ未検出です。

`Error_IllegalParameter`: パラメーターが範囲外です。

解説

- ・オートパワーオフ時間を設定します。
- ・*minutes* == 0 に設定した場合、オートパワーオフを行いません。

- (30) `setBluetoothSecurityMode:encryption:PINCode:deviceName:`
 - (int)`setBluetoothSecurityMode:(int)security encryption:(BOOL)encryptionOn`
 `PINCode:(NSString *)PINCode deviceName:(NSString *)deviceName;`
 Bluetooth セキュリティモードを指定します。

パラメーター

security: Bluetooth のセキュリティモードを指定します。

<i>security</i>	セキュリティモード
<code>BTSecurityLevel_1</code>	セキュリティモード 1
<code>BTSecurityLevel_3</code>	セキュリティモード 3
<code>BTSecurityLevel_4</code>	セキュリティモード 4

encryptionOn: 暗号化を指定します。

NO 暗号化なし

YES 暗号化あり

PINCode: Bluetooth の PIN Code を設定します。

<i>PINCode</i>	設定範囲
文字列の長さ	4～16
指定できる文字	空白 ! # \$ % & ' () * + , - . / 0～9 : ; < = > ? @ A～Z [\] ^ _ ` a～z { } ~

deviceName: Bluetooth の Device Name を設定します。

<i>deviceName</i>	設定範囲
文字列の長さ	2～40
指定できる文字	空白 ! # \$ % & ' () * + , - . / 0～9 : ; < = > ? @ A～Z [\] ^ _ ` a～z { } ~

戻り値

`Error_Success`: 成功しました。

`Error_NoDevice`: プリンタ未検出です。

`Error_IllegalParameter`: パラメーターが範囲外です。

解説

- Bluetooth セキュリティモードを指定します。
- 変更はプリンタの電源 OFF→ON 後に有効となります。
- *security* == `BT_SECURITY_LEVEL1` の場合、*encryptionOn* の値に関わらず暗号化なしとなります。

BT_SECURITY_LEVEL4 の場合、*encryptionOn* の値に関わらず暗号化ありとなります。

(31) setLanguageOfCharacterString:

– (int)setLanguageOfCharacterString:(int)*language*;
プリンタが搭載している言語を指定します。

パラメーター

language: プリンタが搭載している言語を指定します。

<i>language</i>	プリンタの対応言語
Language_Japanese	日本語
Language_TraditionalChinese	繁体字

戻り値

Error_Success: 成功しました。

Error_IllegalParameter: パラメーターが範囲外です。

解説

• printCharacterString: メソッドは本メソッドで指定した言語で文字列を印刷します。

(32) setPrintQuality:energy:halfPitch:thermalHeadDivision:

– (int)setPrintQuality:(int)*paperType* energy:(int)*energy* halfPitch:(BOOL)*halfPitch*
thermalHeadDivision:(int)*division*;

印字品質、基準エネルギー、ハーフピッチ、サーマルヘッドのブロック分割駆動を指定します。

パラメーター

paperType: 用紙種類ごとの印字品質を指定します。設定範囲(解説参照)

energy: 基準エネルギー値を指定します。

<i>energy</i>	基準エネルギー
Default_Energy	デフォルト
0 ~ 127	カスタム

halfPitch: ハーフピッチを指定します。

YES ハーフピッチモード

NO ノーマルピッチモード

division: サーマルヘッドのブロック分割駆動を指定します。

<i>division</i>	分割駆動数
Division_Auto	自動分割
Division_Fixed_6	固定 6 分割

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

・印字品質、基準エネルギーを指定します。使用する用紙および環境により、各パラメーターを調整してください。

・*paperType* には下記の値以外指定しないで下さい。

・用紙種類ごとの印字品質および基準エネルギーは下記のとおりです。

energy == Default_Energy とした場合、*paperType* で指定した印字品質に対応するデフォルトの基準エネルギーが自動的に設定されます。

基準エネルギーを変更する場合、下記の値をベースに調整して下さい。

<i>paperType</i>	<i>energy</i>	用紙種類
2	31	
3	36	PD150R
4	38	TP50KJ-R
5	35	
7	35	HW54S

・ハーフピッチモードでは、1つの発熱抵抗体に対し、用紙送り方向に1/2ドットライン分ずらして2回通電を行います。本モードを使用すると発熱抵抗体のドット隙間が無くなり、印字品質が向上します。

・*division* == Division_Auto を指定した場合、サーマルヘッドの分割駆動を自動的に実行します。

(33) `getPrinterData:`

- `(int)getPrinterData:(int)dataType;`
プリンタ情報の取得を行います。

パラメーター

dataType: 取得する情報を指定してください。

<i>dataType</i>	取得する情報
PrinterData_Status	プリンタステータス
PrinterData_FirmwareVersion	ファームウェアバージョン

戻り値

Error_Success: 成功しました。

Error_NoDevice: プリンタ未検出です。

Error_IllegalParameter: パラメーターが範囲外です。

解説

- ・プリンタ情報の取得を行います。
- ・取得した情報は[Notification](#)にて通知されます。

(34) waitPrintEnd:

– (int)waitPrintEnd:(int)*timeOut*;
印刷完了を待機します。

パラメーター

timeOut: 印字完了待機のタイムアウト時間を秒単位で指定します。(1～1800)
Default_TimeOut: デフォルトは 180 秒となっています。

戻り値

Error_Success: 成功しました。
Error_NoDevice: プリンタ未検出です。
Error_IllegalParameter: パラメーターが範囲外です。
Error_Timeout: タイムアウトしました。

解説

- ・印刷完了を待機します。印刷完了またはタイムアウトとなるまでメソッドは処理を返しません。
- ・印刷中でない場合、即処理を返します。
- ・メソッド実行後、指定したタイムアウト時間経過後に印刷完了しない場合、Error_Timeout が返ります。
- ・実行するメソッドおよび送信データによって印刷完了までの時間は異なります。また、本メソッドをメインスレッド上で実行すると、処理を返すまで一時的にメインスレッドが停止した状態となります。実運用条件にて十分確認の上ご使用下さい。

6. Notification による通知

NSNotificationCenter クラスにより、各メソッドの完了および getPrinterData:メソッドによる情報等を通知します。

(1) Notification による通知受信の登録

- (void)addObserver:(id)notificationObserver selector:(SEL)notificationSelector
name:(NSString *)notificationName object:(id)notificationSender;

パラメーター

notificationObserver: 通知を受け取るオブジェクトを指定してください。

selector: 通知を受信するメソッドを指定してください。

name: @"FCLReceivedNotificationForMultiplePrinters"を指定してください。

object: nil を指定してください。

(使用例)

```
[[NSNotificationCenter defaultCenter] addObserver:self  
selector:@selector(_sessionDataReceived:)  
name:@"FCLReceivedNotificationForMultiplePrinters" object:nil];
```

(2)通知の受信方法 (*selector*を"_sessionDataReceived"とした場合)

- (void)_sessionDataReceived:(NSNotification *)notification{
 NSDictionary* notificationDictionary = nil;
 NSNumber* notificationType = nil;
 NSString* notificationPrinterModel = nil;
 NSData* notificationData = nil;
 notificationDictionary = [notification userInfo];

 notificationType = [notificationDictionary objectForKey:@"notificationType"];
 notificationPrinterModel = [notificationDictionary objectForKey:@"printerModelNumber"];
 notificationData = [notificationDictionary objectForKey:@"data"];
}

解説

- ・プリンタ情報が通知されます。

notificationType: 通知の種類

notificationType	通知の種類
PrinterData_Status	各メソッドの印刷結果およびプリンタステータス
PrinterData_FirmwareVersion	ファームウェアバージョン
PrinterEvent_ConnectedPrinter	プリンタが接続されました。
PrinterEvent_DisconnectedPrinter	プリンタが切断されました。

printerModelNumber: 通知対象のプリンタの modelNumber

data: データ

notificationType == PrinterData_Status

各メソッドの印刷結果およびプリンタステータス

第 1 バイト (印刷結果)

第 1 バイト	印刷結果
Error_Success	成功しました。
Error_PrinterError	用紙なし等のプリンタエラーが発生しました。

第 2 バイト (プリンタ情報)

bit	ステータス	内 容
0	未使用	0 固定
1	未使用	0 固定
2	未使用	1 固定
3	オンライン/オフラインの状態	0:オンライン 1:オフライン
4	未使用	0 固定
5	未使用	0 固定
6	フィードスイッチによる用紙送り	0:用紙送り中でない 1:用紙送り中
7	未使用	0 固定

第 3 バイト (エラー情報)

bit	ステータス	内 容
0	ローバッテリー	0:未発生 1:発生
1	受信データ異常	0:正常 1:異常
2	カバーオープン	0:未発生 1:発生
3	未使用	0 固定
4	マーク検出異常	0:未発生 1:発生
5	ハードウェア異常	0:未発生 1:発生
6	ヘッド温度異常	0:未発生 1:発生
7	電源電圧異常	0:未発生 1:発生

* ハードウェア異常とは、MCU 動作異常、ヘッド熱暴走、サーマルヘッドケーブル外れ、ヒューズ断、登録異常です。

第 4 バイト (用紙検出情報)

bit	ステータス	内 容
0	未使用	0 固定
1	未使用	0 固定
2	用紙なし	0:未発生 1:発生
3~7	未使用	0 固定

第 5 バイト (ID) 範囲(128～255)
各メソッドを実行する毎にインクリメントされる値です。
255 の次は 128 となります。

notificationType == PrinterData_FirmwareVersion

ファームウェアバージョン

ファームウェアバージョンの *data* の構成は下記のとおりです。

(例: ブートプログラムバージョン V1.00, メインプログラムバージョン V1.01)

"62HB0 1.00 1.01"

([36]₁₆ [32]₁₆ [48]₁₆ [42]₁₆ [30]₁₆ [20]₁₆ [31]₁₆ [2E]₁₆ [30]₁₆ [30]₁₆ [20]₁₆ [31]₁₆ [2E]₁₆ [30]₁₆ [31]₁₆)

notificationType == PrinterEvent_CennectedPrinter

startPrinter:メソッド実行後、プリンタとの接続に成功すると通知されます。

notificationType == PrinterEvent_DisconnecetedPrinter

プリンタとの通信が切断されると通知されます。

closePrinter:メソッド実行時は通知されません。

7. 各設定の初期値

startPrinter:メソッド実行時に各設定が以下に示すように初期化されます。

設定可能なメソッド	パラメーターまたは設定	初期値
startPage:	印刷モード	行モード
setAbsolutePosition:	展開位置の絶対値指定	横方向 0 縦方向 0
setCharacterStyle:kanjiSize:doubleSize:bold:bwReverse:underline:space:leftOfKanji:rightOfKanji:	ANK フォントサイズ	12 × 24 ドット
	漢字フォントサイズ	24 × 24 ドット
	横方向/縦方向倍率	1 倍
	強調印字	なし
	白黒反転印字	なし
	アンダーライン	なし
	ANK 文字の文字間スペース	0
	漢字の左スペース	0
	漢字の右スペース	0
setInternationalCharacterCode:country:	文字コードテーブル	カタカナ
	国際文字	(日本語搭載プリンタ)日本 (繁体字搭載プリンタ)アメリカ
setLineFeedAmount:page:	改行量	26
	1 ページの行数	44
setOrientation:	印刷方向	左→右
setPaperType:feedBefore	用紙種類	ラベル
PrintPage:feedAfterMarkDetection:autoFeedLength:	印刷前用紙逆送り量	0
	マーク検出頭出し量	0
	自動給紙量	80 (160 ドットライン)
setPrintQuality:energy:halfPitch:thermalHeadDivision:	印字品質	7
	基準エネルギー	35
	ピッチモード	ハーフピッチ
	サーマルヘッドのブロック分割駆動	自動分割
setRelativePosition:	展開位置の相対位置指定	横方向 0 縦方向 0
setPrintSpeed:	印刷速度	120
setRotation:	回転角度	回転なし
setAutoPowerOffTime:	オートパワーオフ時間	保持
setBluetoothSecurityMode:encryption:	Bluetooth セキュリティモード	保持
	暗号化	保持
	Bluetooth PIN コード	保持
	Bluetooth デバイス名	保持
setLanguageOfCharacterString	言語	日本語

8. エラーコード

以下の表に本ライブラリが返すエラーコードについて記述します。

定数	エラー内容
Error_Success	成功しました。
Error_Failed	プリンタがオフラインです。
Error_NoDevice	プリンタ未検出です。
Error_NoExistFile	ファイルが存在しません。
Error_IllegalFileFormat	ファイルフォーマットが不正です。
Error_IllegalImage	非対応のイメージです。
Error_IllegalParameter	パラメーターが範囲外です。
Error_PageModeOutSide	ページモードにおいて領域外となりました。
Error_NotPageMode	ページモードではありません。
Error_IllegalFontDataSize	フォントデータのサイズが間違っています。
Error_MarkNoDetect	マーク未検出となりました。
Error_AlreadyPageMode	既にページモードです。
Error_AlreadyConnectedPrinter	既にプリンタと接続中です。
Error_PrinterError	用紙なし等のプリンタエラーが発生しました。
Error_MemoryAllocation	ライブラリ内でメモリ確保に失敗しました。
Error_TimeOut	タイムアウトしました。
Error_AlreadyConnectedPrinterOtherInstance	既に他のインスタンスが使用中です。

9. 注意事項

- ・オペレーティングシステムおよびデバイスの組み合わせによっては正常に動作しない場合があります。
- ・本ライブラリは非同期動作 (waitPrintEnd:メソッドを除く) となります。各メソッドの戻り値が Error_Success の場合でも、実際の印刷が成功するとは限りません。
- ・本ライブラリは実機動作用にビルドされております。よって iOS シミュレータ上では動作しません。
- ・行モードで各メソッドを実行した場合、データが 1 行分に満たないと印刷されずプリンタにバッファリングされた状態となります。即座に印刷する場合、メソッド実行後に linefeedPaper:メソッドを実行するか、printCharacterString:メソッドにより改行コード(“\n”)を送信し、改行して下さい。
- ・本ライブラリはメインスレッド上でのみ動作します。インスタンスをメインスレッド上で生成して下さい。
- ・複数のアプリから同時に接続することはできません。
- ・iOS が同時に接続可能な Bluetooth デバイス数は環境によって異なります。
- ・iOS 仕様により、大量のデータを連続で送信する場合に通信が停止する場合があります。複数ページにわたるデータを連続で送信する場合は十分に評価を行って下さい。通信が停止した場合、closePrinter:メソッドおよび startPrinter:メソッドで再接続して下さい。
- ・ホームボタンをクリックする等の動作でアプリをバックグラウンド状態にした場合、通信が切断されます。データ送信中に通信が切断された場合、再度 iOS デバイスとプリンタのペアリングが必要な場合があります。バックグラウンド状態での通信の継続が必要な場合、アプリプロジェクトの info.plist に下記を追加することでバックグラウンドでの通信を継続することが可能です。ただし、アプリのバッテリー消費量が高くなる場合があるため、実運用条件にて充分確認の上でご使用下さい。

```
<key>UIBackgroundModes</key>
<array>
  <string>external-accessory</string>
</array>
```

10. 既知の現象について

特定の iOS バージョンもしくはデバイスによっては、下記の現象を確認しています。
なお、本項目に記載した項目以外には問題がないことを保証するものではありません。

iOS バージョン	iOS 9.3.x
デバイス	iPhone SE
現象	・ startPrinter:メソッドに失敗する場合があります。
回避方法	・ iOS 10 で使用する。

iOS バージョン	iOS 9, 10
デバイス	iPad mini 2 iPad mini 3
現象	・ iOS 設定アプリの Bluetooth 画面において、接続および接続解除を繰り返すと、再接続しても一定時間後に未接続状態になる。 Bluetooth デバイスを複数接続していると、現象発生頻度が上がる。
回避方法	・ 接続および接続解除を繰り返し行わない。 ・ 現象発生した場合、ペアリング解除後に再接続する。

iOS バージョン	iOS 8
デバイス	iPad mini 2 iPad mini 3
現象	・ iOS 設定アプリの Bluetooth 画面において、ペアリングおよびペアリング解除を連続で実行すると iOS がハングアップおよび再起動する。
回避方法	・ ペアリングおよびペアリング解除を連続で実行しない。

仕様書改版履歴表

版数	改版日	改版内容	備考
01	20160129	初版	
02	20170131	<ul style="list-style-type: none">・ setLanguageOfCharacterStringメソッドを追加。 繁体字搭載プリンタに対応。・ versionプロパティを追加。・ setBluetoothSecurityMode:encryption:PINCode:deviceName: メソッドのパラメーター説明を変更。・ 既知の現象について項目を追加。	
03	20240201	<ul style="list-style-type: none">・ 社名変更・ 対応OS、アーキテクチャーを変更・ Bitcode削除	

ライブラリ改版履歴表

バージョン	改版日	改版内容	備考
V3.00	20151228	-	
V3.01	20170130	<ul style="list-style-type: none">・ iOS 10のサポート・ iOS 7のサポート終了・ setLanguageOfCharacterStringメソッドを追加。 繁体字搭載プリンタに対応。・ printQRCodeのデータビット数算出処理を修正。・ versionプロパティを追加	
V4.00	20240201	<ul style="list-style-type: none">・ 社名変更・ 対応OS、アーキテクチャーを変更・ Bitcode削除	