

StarPRNT iOS SDK ユーザーズマニュアル

2017 年 08 月 16 日

要約

このマニュアルは StarPRNT iOS SDK に関する情報を提供するものです。
ユーザー向けに StarPRNT アプリケーションの構築に関するガイドラインを示します。

ドキュメントの履歴

日付	変更内容
2016 年 04 月 01 日	発行
2016 年 08 月 11 日	誤記訂正
2016 年 09 月 12 日	SMCSAllReceipts クラスに uploadData メソッドを追加
2017 年 04 月 17 日	ブラックマーク、ページモードに対応
2017 年 08 月 16 日	StarIO フレームワークに含まれるクラスの API リファレンスを追加 SMBluetoothManagerFactory クラスを追加 外部機器（バーコードリーダー・スケール・カスタマーディスプレイ） の API リファレンスを追加

本書に関して

本マニュアルは、StarIO と Star POS プリンタが通信を行う、iOS アプリケーションの作成方法を解説しています。

また、このマニュアルは、アプリケーション・システム開発者を対象に作成しており、利用者は Swift もしくは Objective-C 言語の基礎を理解していることを前提としています。

スター精密グローバルサポートサイトの Developers セクションには、その他のオペレーティングシステムとプログラミング言語に利用可能な SDK が用意されています。最新の SDK、テクニカルドキュメント、FAQ 及び、その他の追加情報については、Developers セクションをご確認ください。

注意事項:

- iPad, iPhone, iPod, iPod touch, Retina は、米国および他の国々で登録された Apple Inc. の商標です。iPad Air, iPad mini は、Apple Inc. の商標です。“iPhone”の商標は、アイホン株式会社のライセンスにもとづき使用されています。IOS は、米国およびその他の国における Cisco 社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。
- Windows は、米国 Microsoft Corporation の、米国、日本およびその他の国における登録商標です。
- 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す。
- 本マニュアルの内容は、予告無く変更する場合があります。
- スター精密株式会社は、正確な情報を提供するためにあらゆる措置を取っていますが、誤りや不作為について責任を負うものではありません。
- スター精密株式会社は、このマニュアルに記載されている情報の使用に起因するいかなる損害に対しても責任を負うものではありません。
- 本マニュアルの一部、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。

目次

1	はじめに	9
1.1	対応プリンタ	9
1.2	プリンタを iOS デバイスに接続する	12
2	iOS StarPRNT プロジェクト構成方法	17
2.1	iOS StarPRNT SDK をプロジェクトに追加	17
2.2	必要なフレームワークをプロジェクトに追加	17
2.3	Information Property List へ項目を設定	17
3	SMPort クラス (StarIO.framework)	18
3.1	SMPort クラスを使った印刷処理フロー	19
3.2	モデル: SMPort クラス	20
3.3	getPort メソッド	21
3.4	releasePort メソッド	23
3.5	searchPrinter メソッド	24
3.6	writePort メソッド	26
3.7	readPort メソッド	27
3.8	getParsedStatus メソッド	28
3.9	beginCheckedBlock メソッド	29
3.10	endCheckedBlock メソッド	30
3.11	disconnect メソッド	31
3.12	getFirmwareInformation メソッド	31
3.13	StarIOVersion メソッド	32
3.14	portName プロパティ	32
3.15	portSettings プロパティ	32
3.16	timeoutMillis プロパティ	32
3.17	endCheckedBlockTimeoutMillis プロパティ	33
3.18	connected プロパティ	33
4	StarPrinterStatus 構造体 (StarIO.framework)	34
4.1	モデル: StarPrinterStatus 構造体	36
5	PortInfo クラス (StarIO.framework)	37
5.1	portName プロパティ	37
5.2	macAddress プロパティ	37
5.3	modelName プロパティ	37
6	SMBluetoothManager クラス (StarIO.framework)	38
6.1	SMBluetoothManager を使った Bluetooth 設定変更フロー	39
6.2	モデル: SMBluetoothManager クラス	40
6.3	initWithName メソッド	42
6.4	open メソッド	42
6.5	loadSetting メソッド	43
6.6	close メソッド	43
6.7	apply メソッド	44
6.8	portName プロパティ	44
6.9	deviceType プロパティ	44
6.10	opened プロパティ	44
6.11	deviceName プロパティ	45
6.12	iOSPortName プロパティ	45
6.13	autoConnect プロパティ	45

6.14	security プロパティ	46
6.15	pinCode プロパティ	46
6.16	deviceNameCapability プロパティ	46
6.17	iOSPortNameCapability プロパティ	46
6.18	autoConnectCapability プロパティ	46
6.19	securityTypeCapability プロパティ	47
6.20	pinCodeCapability プロパティ	47
6.21	SMDeviceType コンスタント	47
6.22	SMBluetoothSecurity コンスタント	47
6.23	SMBluetoothSettingCapability コンスタント	48
7	StarIoExt クラス (StarIO_Extension.framework)	49
7.1	createCommandBuilder メソッド	50
7.2	createDisplayCommandBuilder メソッド	51
7.3	createScaleCommandBuilder メソッド	52
7.4	createBcrConnectParser メソッド	53
7.5	createDisplayConnectParser メソッド	54
7.6	createScaleConnectParser メソッド	55
7.7	createScaleWeightParser メソッド	56
7.8	StarIoExtEmulation コンスタント	57
7.9	StarIoExtCharacterCode コンスタント	58
7.10	StarIoExtBcrModel コンスタント	58
7.11	StarIoExtScaleModel コンスタント	58
7.12	StarIoExtDisplayModel コンスタント	59
7.13	モデル : portSetting and StarIoExtEmulation	60
8	ISCBBuilder クラス (StarIO_Extension.framework)	61
8.1	モデル : ISCBBuilder interface メソッド	64
8.2	beginDocument メソッド	68
8.3	endDocument メソッド	69
8.4	appendInitialization メソッド	70
8.5	appendData メソッド	71
8.6	appendRawData メソッド	72
8.7	appendFontStyle メソッド	73
8.8	appendCodePage メソッド	74
8.9	appendInternational メソッド	77
8.10	appendLineFeed メソッド	79
8.11	appendUnitFeed メソッド	80
8.12	appendCharacterSpace メソッド	81
8.13	appendLineSpace メソッド	82
8.14	appendEmphasis メソッド	83
8.15	appendInvert メソッド	84
8.16	appendMultiple メソッド	85
8.17	appendUnderLine メソッド	87
8.18	appendLogo メソッド	88
8.19	appendAbsolutePosition メソッド	89
8.20	appendAlignment メソッド	90
8.21	appendCutPaper メソッド	91
8.22	appendPeripheral メソッド	92
8.23	appendSound メソッド	93

8.24	appendBarcodeData メソッド	94
8.25	appendBarcodeDataWithAbsolutePosition メソッド	96
8.26	appendBarcodeDataWithAlignment メソッド	98
8.27	appendPdf417Data メソッド	100
8.28	appendPdf417DataWithAbsolutePosition メソッド	102
8.29	appendPdf417DataWithAlignment メソッド	104
8.30	appendQrCodeData メソッド	106
8.31	appendQrCodeDataWithAbsolutePosition メソッド	107
8.32	appendQrCodeDataWithAlignment メソッド	108
8.33	appendBitmap メソッド	110
8.34	appendBitmapWithAbsolutePosition メソッド	112
8.35	appendBitmapWithAlignment メソッド	114
8.36	appendBlackMark メソッド	116
8.37	beginPageMode メソッド	117
8.38	endPageMode メソッド	118
8.39	appendPageModeVerticalAbsolutePosition メソッド	119
8.40	appendPageModeRotation メソッド	120
8.41	commands プロパティ	122
8.42	SCBInitializationType コンスタント	123
8.43	SCBFontStyleType コンスタント	123
8.44	SCBCodePageType コンスタント	124
8.45	SCBInternationalType コンスタント	126
8.46	SCBLogoSize コンスタント	127
8.47	SCBAlignmentPosition コンスタント	127
8.48	SCBCutPaperAction コンスタント	128
8.49	SCBPeripheralChannel コンスタント	128
8.50	SCBSoundChannel コンスタント	129
8.51	SCBBarcodeSymbology コンスタント	129
8.52	SCBBarcodeWidth コンスタント	130
8.53	SCBPdf417Level コンスタント	131
8.54	SCBQrCodeModel コンスタント	131
8.55	SCBQrCodeLevel コンスタント	132
8.56	SCBBitmapConverterRotation コンスタント	132
8.57	SCBBlackMarkType コンスタント	133
9	ISDCBBuilder インターフェイス (StarIO_Extension.framework)	134
9.1	モデル: ISDCBBuilder インターフェイス	135
9.2	appendData メソッド	135
9.3	appendBackSpace メソッド	136
9.4	appendHorizontalTab メソッド	136
9.5	appendLineFeed メソッド	136
9.6	appendCarriageReturn メソッド	137
9.7	appendBitmap メソッド	138
9.8	appendInternational メソッド	139
9.9	appendCodePage メソッド	140
9.10	appendDeleteToEndOfLine メソッド	141
9.11	appendClearScreen メソッド	141
9.12	appendHomePosition メソッド	141
9.13	appendTurnOn メソッド	142

9.14	appendSpecifiedPosition メソッド	142
9.15	appendCursorMode メソッド	143
9.16	appendContrastMode メソッド	143
9.17	appendUserDefinedCharacter メソッド	144
9.18	appendUserDefinedDbcsCharacter メソッド	145
9.19	commands プロパティ	147
9.20	passThroughCommands プロパティ	147
9.21	SDCBInternationalType コンスタント	148
9.22	SDCBCodePageType コンスタント	149
9.23	SDCBCursorMode コンスタント	150
9.24	SDCBContrastMode コンスタント	150
10	ISSCBBuilder インターフェイス (StarIO_Extension.framework)	151
10.1	モデル: ISSCBBuilder インターフェイス	151
10.2	appendData メソッド	152
10.3	appendZeroClear メソッド	152
10.4	appendUnitChange メソッド	153
10.5	commands プロパティ	154
10.6	passThroughCommands プロパティ	154
11	ISCPParser インターフェイス (StarIO_Extension.framework)	155
11.1	モデル: ISCPParser インターフェイス	155
11.2	createSendCommands メソッド	156
11.3	createReceiveCommands メソッド	157
11.4	completionHandler プロパティ	158
11.5	StarIoExtParserCompletionResult コンスタント	158
12	ISCPConnectParser インターフェイス (StarIO_Extension.framework)	159
12.1	モデル: ISCPConnectParser インターフェイス	159
12.2	connect メソッド	160
13	ISSCPWeightParser (StarIO_Extension.framework)	161
13.1	モデル: ISSCPWeightParser インターフェイス	161
13.2	weight メソッド	162
13.3	status メソッド	163
13.4	StarIoExtDisplayedWeightStatus コンスタント	163
14	StarPRNT iOS SDK Sample	164
14.1	Communication	164
15	StarIoExtManager クラス (StarIO_Extension.framework)	167
15.1	initWithType メソッド	168
15.2	connect メソッド	169
15.3	disconnect メソッド	170
15.4	port プロパティ	170
15.5	lock プロパティ	171
15.6	delegate プロパティ	171
15.7	printerOnlineStatus プロパティ	172
15.8	printerPaperStatus プロパティ	172
15.9	printerCoverStatus プロパティ	173
15.10	cashDrawerStatus プロパティ	173
15.11	barcodeReaderStatus プロパティ	174
15.12	cashDrawerOpenActiveHigh プロパティ	174
15.13	StarIoExtManagerType コンスタント	175

15.14	StarloExtManagerPrinterStatus コンスタント	175
15.15	StarloExtManagerPrinterPaperStatus コンスタント	176
15.16	StarloExtManagerPrinterCoverStatus コンスタント	176
15.17	StarloExtManagerCashDrawerStatus コンスタント	177
15.18	StarloExtManagerBarcodeReaderStatus コンスタント	177
16	StarloExtManagerDelegate プロトコル (StarIO_Extension.framework)	178
16.1	didPrinterImpossible メソッド	178
16.2	didPrinterOnline メソッド	179
16.3	didPrinterOffline メソッド	179
16.4	didPrinterPaperReady メソッド	180
16.5	didPrinterPaperNearEmpty メソッド	180
16.6	didPrinterPaperEmpty メソッド	181
16.7	didPrinterCoverOpen メソッド	181
16.8	didPrinterCoverClose メソッド	182
16.9	didCashDrawerOpen メソッド	182
16.10	didCashDrawerClose メソッド	183
16.11	didBarcodeReaderImpossible メソッド	183
16.12	didBarcodeReaderConnect メソッド	184
16.13	didBarcodeReaderDisconnect メソッド	184
16.14	didBarcodeDataReceive メソッド	185
16.15	didAccessoryConnectSuccess メソッド	186
16.16	didAccessoryConnectFailure メソッド	186
16.17	didAccessoryDisconnect メソッド	187
16.18	didStatusUpdate メソッド	188
17	SMBluetoothManagerFactory クラス (StarIO_Extension.framework)	189
17.1	getManager メソッド	189
18	API expanded for Swift support included in the StarIO.framework	190
19	SMCloudServices クラス (SMCloudServices.framework)	191
19.1	showRegistrationView メソッド	191
19.2	isRegistered メソッド	192
20	SMCSAllReceipts クラス (SMCloudServices.framework)	193
20.1	モデル : SMCSAllReceipts class メソッド	193
20.2	uploadBitmap メソッド	194
20.3	uploadData メソッド	195
20.4	updateStatus メソッド	197
20.5	generateAllReceipts メソッド	198
	Appendix A. AllReceipt™ 利用方法案内 (配布用)	200

1 はじめに

1.1 対応プリンタ

	対応プリンタ	F/W バージョン	対応インターフェイス	対応エミュレーション
POS プリンタ	mPOP	1.0 以降	Bluetooth	StarPRNT
	FVP10	1.2 以降	Bluetooth Ethernet	StarLine
	TSP100IIIW	1.0 以降	Wireless LAN	StarGraphic
	TSP100IIILAN	1.0 以降	Ethernet	StarGraphic
	TSP100IIIBI	1.0 以降	Bluetooth *2	StarGraphic
	TSP100IIIU	1.0 以降	USB	StarGraphic
	TSP100ECO	1.0 以降	USB *1	StarGraphic
	TSP100U	1.3 以降	USB *1	StarGraphic
	TSP100GT	1.0 以降	USB *1	StarGraphic
	TSP100LAN	2.0 以降	Ethernet	StarGraphic
	TSP650II	1.0 以降	Bluetooth *2 Ethernet	StarLine
	TSP700II	4.0 以降	Bluetooth *2	StarLine
		3.0 以降	Ethernet	
	TSP800II	2.0 以降	Bluetooth *2	StarLine
		1.2 以降	Ethernet	
モバイル プリンタ	BSC10 LAN	1.0 以降	Ethernet	ESC/POS
	SP700	4.0 以降	Bluetooth(欧米のみ)	StarDotImpact
		3.0 以降	Ethernet	
	SM-S210i	2.4 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-S220i	2.0 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-S230i	1.0 以降	Bluetooth	StarPRNT / EscPosMobile
	SM-T300i	2.4 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-T300	1.1 以降	Wireless LAN	EscPosMobile
	SM-T400i	2.4 以降	Bluetooth	StarPRNT *3 / EscPosMobile
	SM-L200	1.0 以降	Bluetooth Low Energy	StarPRNT / StarPRNTL *4
	SM-L300	1.1 以降	Bluetooth Low Energy	StarPRNT / StarPRNTL

*1 : Apple AirMac Express が必要です。

-
- *2 : データタイムアウト機能のタイムアウト値設定をサポート。TSP100IIIBI はファームウェアバージョン 1.0 以降、TSP650II, TSP800II はファームウェアバージョン 2.0 以降、TSP700II はファームウェアバージョン 5.0 以降が必要です。
 - *3 : StarPRNT エミュレーションのサポートはファームウェアバージョン 3.0 以降
 - *4 : StarPRNTL エミュレーションのサポートはファームウェアバージョン 2.0 以降

❖ モバイルプリンタ

StarPRNT エミュレーションを使用するには

StarPRNT エミュレーションを使用するには、プリンタ本体のエミュレーションを “StarPRNT” に設定する必要があります。（SM-L200、L300 は切り替え不要です。）

エミュレーションの切り替えは以下の手順で行ってください。

◆ ESC/POS ⇄ StarPRNT エミュレーション切り替え方法

1. プリンタの電源を入れ、プリンタカバーを開きます。
2. 電源ボタンと FEED ボタンを同時に長押しした後、ERROR ランプが 5 回点滅したことを確認しすぐに電源ボタンと FEED ボタンから指を放します。エミュレーションの切り替えが自動的に行われます。
3. 用紙をセット後、プリンタカバーを閉めると設定されたエミュレーションが印字されます。
ESC/POS の場合 : EMU = ESC/POS
StarPRNT の場合 : EMU = StarPRNT

エミュレーションが正しく切り替わっていない場合、再度 1～3 の手順を行ってください。

その際、2 の手順においては、点滅中に指を放さず、点滅が 5 回完了したことを確認してから指を放すように注意してください。

4. エミュレーションの切り替え後は、プリンタの電源を一度オフにしてから再投入してください。
選択したエミュレーションは、プリンタの電源を再投入することで有効になります。

1.2 プリンタを iOS デバイスに接続する

Bluetooth インターフェイス

Star POS デバイスは、工場出荷時の初期設定では機種ごとに"Star Micronics"等、共通の Bluetooth デバイス名が設定されています。同じ Bluetooth デバイス名の機種を複数台配置して運用される場合、Bluetooth デバイス名の変更を行うと Star POS デバイスの判別が付けやすく便利です。

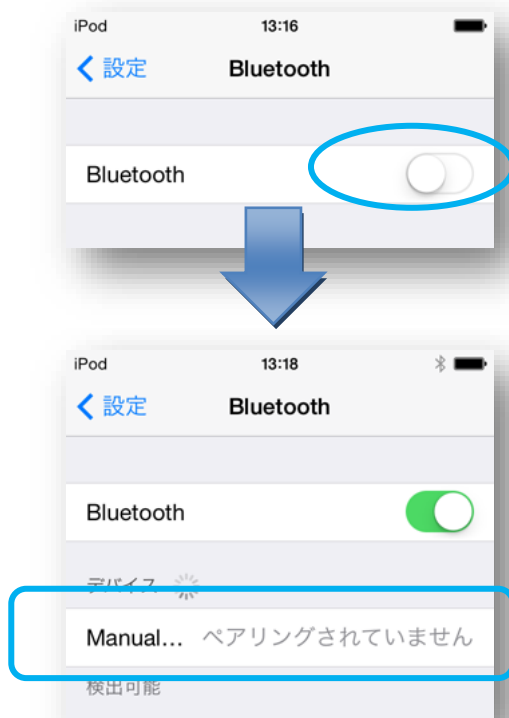
Bluetooth デバイス名の変更等、Star POS デバイスの LAN/Bluetooth 設定値は、mPOP の場合は mPOP Utility、TSP100IIIIBI の場合は TSP100III Utility といった専用ユーティリティを、それ以外のモデルでは Star Setting Utility を使用して変更することができます。それぞれ App Store よりダウンロードしてください。

◆ペアリング

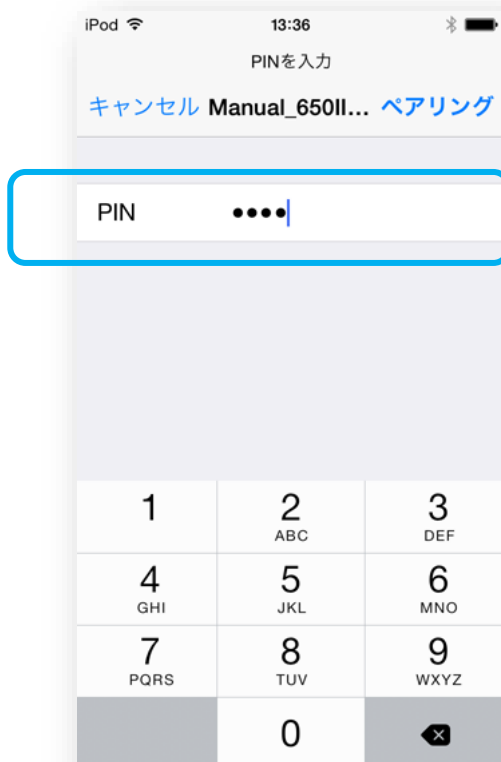
1. Star POS プリンタを、ペアリングを行う iOS デバイスと接続が可能な範囲に設置して電源を投入します。
2. iOS の[設定]より、[Bluetooth]をタップします。



3. [Bluetooth]を ” オン” に設定すると、iOS デバイスとペアリングが可能な Bluetooth デバイスの検索を行い、表示します。ペアリングを行う Star POS プリンタをタップします。



4. PIN を入力します。（Star POS プリンタの Bluetooth セキュリティが PIN Code の場合）



5. 以下の表示が確認できればペアリング完了です。



◆Bluetooth 名称の変更

App Store から mPOP Utility、TSP100III Utility、Star Setting Utility をダウンロードし、iOS ポート名を変更することができます。必要に応じてご使用ください。

iOS ポート名はペアリング実行後、以下の手順で確認できます。

「設定」 - 「一般」 - 「情報」 Bluetooth アドレスの下に表示

Ethernet / Wireless LAN インターフェイス

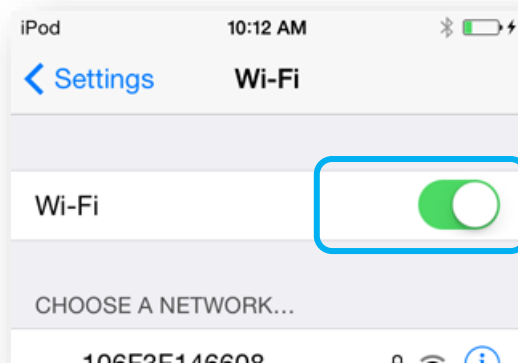
Star POS プリンタは、工場出荷時の初期設定は DHCP が有効になっています。使用するネットワークが DHCP を利用できる環境の場合、POS プリンタが自動的に IP アドレスを取得できるように、必要なネットワーク構成を構築してください。

また、プリンタ本体の #9100 Multi Session を無効に設定して使用してください。設定は、FEED ボタンを押しながら電源スイッチを ON することで実行される自己印字で確認できます。#9100 Multi Session の確認・変更、固定 IP アドレスの設定方法については、[こちらのリンク](#)の「対応 OS・環境一覧 > ユーティリティ一覧」より「イーサネットプリンタ利用手引き」をご参照ください。

POS プリンタの Ethernet 設定は、TSP100 シリーズ（TSP100III 含む）では TSP100III Utility*、それ以外のモデルでは Star Setting Utility*を使用して設定することができます。TSP100III Utility、Star Setting Utility は App Store よりダウンロードしてください。

* これらの Utility は、IP アドレスの割り当てられていない（IP アドレス：0.0.0.0）プリンタでは使用できません。

1. Star POS デバイスに IP アドレスを割り当て、ネットワークに接続します。
2. 「設定」をタップします。
3. 「Wi-Fi」を ON に設定します。



4. Star POS デバイスと同じネットワークに接続します。

USB インターフェイス<TSP100IIIU のみ>

プリンタと iOS デバイスを Lightning ケーブルで接続してください。

◆USB プリンタのデバッグ方法

※下記の表記、手順、URL は、Xcode のバージョンアップやサイト更新等により変更される可能性があります。本内容は 2017/4/4 時点で、Mac OS 10.12.4、Xcode 8.3 を使用して確認しています。

USB プリンタを使用する場合、iOS デバイスの Lightning ポートはプリンタとの接続に使用されるため、Xcode のデバッグ機能を使用することができません。
しかし、以下の方法でログの確認や Instruments の使用が可能です。

1. ログ出力

アプリ内で以下のコードを実行することで、NSLog() の出力をファイルに保存することができま
す。ここでは /tmp フォルダ内の xcode.log というファイルに保存しています。

```
NSString *tmpDirPath = NSTemporaryDirectory();
NSString *path = [NSString stringWithFormat:@"%s@xcode.log", tmpDirPath];
freopen([path cStringUsingEncoding:NSUTF8StringEncoding], "w+", stderr);
```

< ログの取り出し方 >

1. iOS デバイスを Lightning ケーブルで mac に接続します。
2. Xcode を起動し、メインメニューの” Window” - ”Devices” を選択して” Devices” ウィンドウを開きます。
3. 左ペイン内の ”Devices” 内から iOS デバイスを選択します。
4. 右ペインの ”Installed Apps” からログを取得するアプリをクリックして選択し、歯車アイコンをクリックして ”Download Container...” を選択後、アプリを mac に保存します。
5. 保存したファイルを右クリックし ”Show Package Contents” を選択します。
アプリの内容が表示されるので、tmp フォルダ内の xcode.log を取り出します。

2. Instruments

Instruments をネットワーク経由で使用方法については、Apple 社の Instruments User Guide（英語）をご参照ください。

<https://developer.apple.com/library/content/documentation/DeveloperTools/Conceptual/InstrumentsUserGuide/WorkingwithTargets.html>

2 iOS StarPRNT プロジェクト構成方法

2.1 iOS StarPRNT SDK をプロジェクトに追加

- “Build Phases”タブの”Link Binary With Libraries”に StarIO.framework を追加する
- “Build Phases”タブの”Link Binary With Libraries”に StarIO_Extension.framework を追加する
- “Build Phases”タブの”Link Binary With Libraries”に SMCloudServices.framework を追加する
- “Build Phases”タブの”Copy Bundle Resources”に SMCloudServices.framework フォルダ配下にある SMCloudServicesResources.bundle を追加する

–SMCloudServices.framework について–

スター精密クラウドサービス (AllReceipts 等) をアプリケーションで用いる際に追加する必要があります。

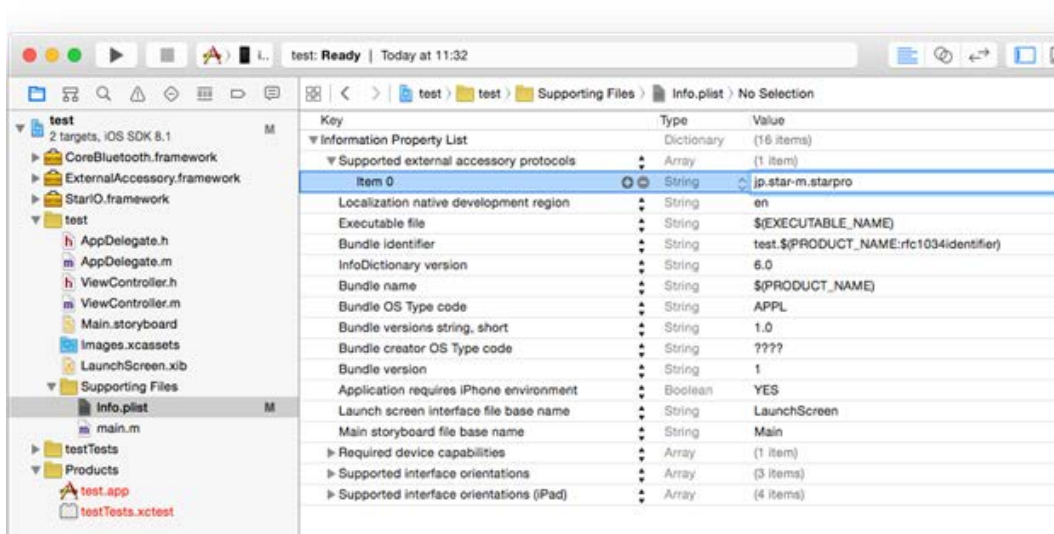
2.2 必要なフレームワークをプロジェクトに追加

- CoreBluetooth.framework
- ExternalAccessory.framework

2.3 Information Property List へ項目を設定

※Bluetooth プリントを使用しない場合は、この設定を行わないでください。

1. Information Property List (デフォルトでは”Info.plist”) を選択します。
2. Key ”Supported external accessory protocols”を追加
3. 項目名左側の▽をクリックして表示される”Item 0”の[Value]に ”jp.star-m.starpro” を設定



3 SMPort クラス (StarIO.framework)

SMPort クラスはプリンタとの通信を行うためのクラスです。

メソッド

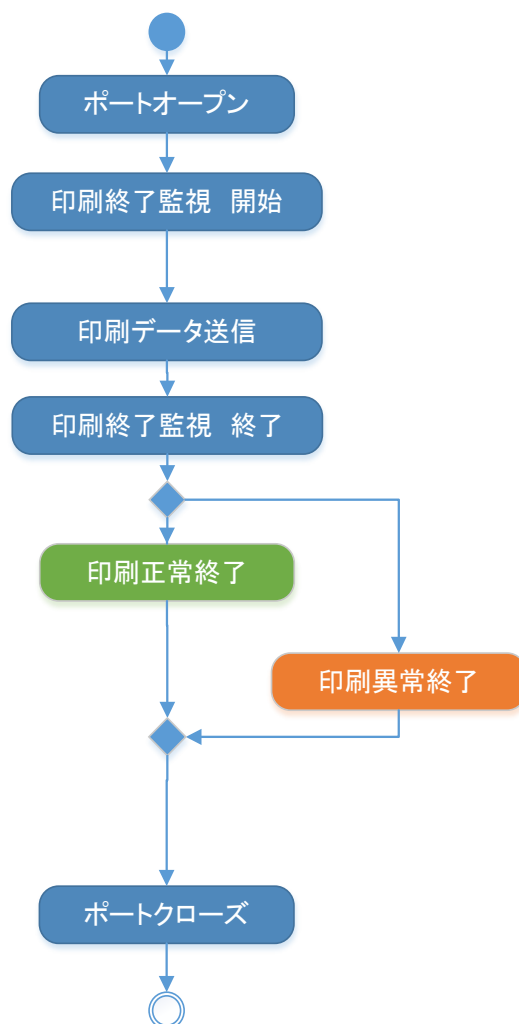
名称	説明
getPort	SMPort オブジェクトを生成し、指定されたポートへの接続をオープンします。
releasePort	指定されたポートへの接続をクローズし、SMPort オブジェクトを破棄します。
searchPrinter	iOS デバイスと接続できるプリンタを検索します。
writePort	プリンタにデータを書き込みます。
readPort	プリンタからデータを読み込みます。
getParsedStatus	プリンタのステータスを取得します。
beginCheckedBlock	印刷終了監視を開始します。
endCheckedBlock	印刷終了監視を終了します。
disconnect	指定された Bluetooth プリンタとのコネクションを切断します。
getFirmwareInformation	プリンタのモデル名とファームウェアバージョンを取得します。
StarIOVersion	StarIO ライブラリのバージョン番号を取得します。

プロパティ

名称	説明
portName	プリンタと通信するためのポートを特定する文字列
portSettings	プリンタとの接続設定情報
timeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）
endCheckedBlockTimeoutMillis	endCheckedBlock メソッドのタイムアウト値（ミリ秒指定）
connected	指定された Bluetooth プリンタの iOS デバイスとの接続状態

3.1 SMPort クラスを使った印刷処理フロー

SMPort クラスを利用し、以下の手順に沿って印刷を行います。



```

unsigned char command[] = {0x41, 0x42, 0x43, 0x44, 0x1B, 0x7A, 0x00, 0x1B, 0x64, 0x02};
uint bytesWritten = 0;
StarPrinterStatus_2 starPrinterStatus;
SMPort *port = nil;

@try
{
    port = [SMPort getPort:@"BT:Star Micronics" :@" :10000 ];

    // 印刷終了監視を開始
    [port beginCheckedBlock:&starPrinterStatus :2];

    if (starPrinterStatus.offline == SM_TRUE)
    {
        // プリンタオフライン
    }
    while (bytesWritten < sizeof (command)) {
        bytesWritten += [port writePort: command : bytesWritten : sizeof(command) - bytesWritten];
    }

    // 印刷終了監視を終了
    [port endCheckedBlock:&starPrinterStatus :2];

    if (starPrinterStatus.offline == SM_TRUE)
    {
        // プリンタオフライン
    }
}
@catch (PortException)
{
    // エラー発生
}
@finally
{
    [SMPort releasePort:port];
}
  
```

Communication.m を参照ください。

3.2 モデル: SMPort クラス

各モデルにおいて機能するメソッドは以下のとおりです。

Method	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i/T300	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
getPort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
releasePort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
searchPrinter	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
writePort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
readPort	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
getParsedStatus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
beginCheckedBlock	✓	✓	✓	✓	*1	✓	✓	*3	✓	*3	✓	✓	✓	✓	✓	✓	✓	✓	✓	*1
endCheckedBlock	✓	✓	✓	✓	*1	✓	✓	*3	✓	*3	✓	✓	✓	✓	✓	✓	✓	✓	✓	*1
disconnect *5	✓	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
getFirmwareInformation	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	*4
StarIOVersion	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

-: 無視

*1: ファームウェアバージョン 3.0 以降が必要です。

*2: TSP100IIIU, TSP100U, TSP100GT, TSP100ECO の場合、モデル名は TSP100 になります。

TSP100IIIW, TSP100IIILAN, TSP100LAN の場合、モデル名は TSP100LAN になります。

TSP100U, TSP100GT, TSP100ECO, TSP100IIIW, TSP100IIILAN, TSP100LAN はファームウェアバージョンを取得できません。

*3: SM-220i はファームウェアバージョン 2.1 未満、SM-T300 はファームウェアバージョン 2.4 未満では印刷データ送信完了の確認はできますが、印刷終了監視はできません。

*4: Apple AirMac シリーズを使用する場合、USB モデルは空文字を返します。

*5: Bluetooth インターフェイスでのみ動作します。

3.3 getPort メソッド

SMPort オブジェクトを生成し、指定されたポートへの接続をオープンします。

宣言

```
+ (SMPort *) getPort: (NSString *) portName : (NSString *) portSettings : (u_int32_t) timeoutMillis;
```

引数

名称	説明	型
portName	プリンタと通信するためのポートを特定する文字列	NSString *
portSettings	プリンタとの接続設定情報 <ul style="list-style-type: none"> エミュレーションの指定 データタイムアウト機能の設定 	NSString *
timeoutMillis	内部制御および API のタイムアウト値（ミリ秒指定）	u_int32_t

戻り値

説明	型
SMPort オブジェクト ポートオープンに失敗した場合、既にオープン済みであった場合は、nil が返されます。	SMPort *

印刷の手順については [SMPort クラスを使った印刷処理フロー](#) を参照ください。

1. portName 引数

インターフェイス	portName	説明	例
Bluetooth	BT:デバイス名	iOS ポート名で指定する場合	@ "BT:Star Micronics"
Bluetooth Low Energy	BLE:デバイス名	Bluetooth デバイス名で指定する場合	@ "BLE:STAR L200-00001"
	BLE:MAC アドレス	MAC アドレスで指定する場合	@ "BLE:00:11:62:1b:4d:f4"
Ethernet / Wireless LAN	TCP/IP アドレス	IP アドレスで指定してください。	@ "TCP:192.168.1.130"
USB	USB:デバイス名	iOS ポート名で指定してください。	@ "USB:TSP100"

2. portSettings 引数

portSettings 文字列は以下のフォーマットで指定してください。

- エミュレーション指定文字列
- エミュレーション指定文字列 + ";" で区切ったオプション指定文字列

エミュレーション	エミュレーション指定文字列
StarLine	""
StarPRNT	"Portable"
StarPRNTL	"Portable"
StarGraphic	""
StarDotImpct	""
EscPos	"escpos"
EscPosMobile	"Portable;escpos" もしくは "mini"

接続オプション(Bluetooth I/F のみ)	オプション指定文字列
データタイムアウト機能のタイムアウト値設定	"d[数値]" 例: "d10"

接続オプション(AirMac 使用時のみ)	オプション指定文字列
ポート番号	"[ポート番号]" 例: "9100"

データタイムアウト機能（Bluetooth I/F のみ、サポート状況は[対応プリンタ](#)を参照）

印刷中に指定した時間プリンタに対してデータが送られなかった場合にデータキャンセル機能を動作させます。指定時間は[数値]に 0～255（単位：秒）を設定します。デフォルトでは 3 秒となっています。範囲外の値を設定した場合、その値は無視されます。この機能により、データ送信中に Bluetooth 接続が切断された場合に、次の印刷が不正な内容になることを防ぐことができます。

portSettings 引数指定例

プリンタ接続環境	指定すべき portSettings
EscPosMobile エミュレーションのモバイルプリンタをデフォルト設定で使用する	"Portable;escpos"
StarLine エミュレーションの POS プリンタで、印刷中に 10 秒間データが送られなかった場合にデータキャンセル機能を動作させる	";d10"

3. timeoutMillis 引数

timeoutMillis は、ミリ秒のタイムアウト値で、内部制御と API での通信に使用されるものです。このパラメータは、API が制限された時間内で完了することを保証しますが、正確なタイムアウトの長さを保証するものではありません。

指定した時間が 10 秒未満の場合、タイムアウトは 10 秒になります。

portSettings パラメータで[データタイムアウト機能のタイムアウト値設定]を行う場合、timeoutMillis 引数の値をデータタイムアウト機能のタイムアウト値の指定時間より 3 秒以上長くするように設定してください。3 秒未満に設定した場合、内部制御により endCheckedBlock メソッドのタイムアウト時間は自動的に 3 秒長くするように設定されます。

4. Apple AirMac シリーズのプリンタ共有機能をご使用の場合

portName には、AirMac シリーズの IP アドレスを指定してください。

例): TCP:192.168.1.2

portSettings には、ポート番号を指定します。

"9100" ～ "9109"を順に指定し、接続に成功した値をご使用ください。

5. SM-L シリーズの注意事項

iOS 端末で Bluetooth Low Energy にて通信する場合、プリンタとの接続に時間がかかることがあります。接続に失敗した場合は、接続が成功するまでリトライを行ってください。

また、接続時間を短縮したい場合は、プリンタとの接続が常に維持されるようアプリケーションを設計してください。

* ただし、接続中のプリンタは他のアプリケーション、および他の端末から検出することができなくなります。

3.4 releasePort メソッド

指定されたポートへの接続をクローズし、SMPort オブジェクトを破棄します。

宣言

```
+ (void) releasePort: (SMPort *) port;
```

引数

名称	説明	型
port	以前に getPort メソッドにより生成された SMPort オブジェクト	SMPort *

戻り値

説明	型
-	-

印刷の手順については [SMPort クラスを使った印刷処理フロー](#) を参照ください。

1. 注意事項

getPort を実行した後は、必ず releasePort してから次の getPort を行ってください。releasePort をせずに次の getPort を行くと、nil が返されます。

3.5 searchPrinter メソッド

iOS デバイスと接続できるプリンタを検索します。

LAN 上のプリンタとペアリングされた Bluetooth プリンタを検索し、検索結果を NSArray 型で返します。

宣言

```
+ (NSArray *) searchPrinter;
```

```
+ (NSArray *) searchPrinter: (NSString *) target;
```

引数

名称	説明	型
target	検索したい Star プリンタのインターフェイスタイプを指定する文字列 target を指定しない場合、すべてのインターフェイスタイプを検索します。	NSString *
	Bluetooth I/F の場合	
	Bluetooth Low Energy I/F の場合	
	Ethernet / Wireless LAN I/F の場合	
	USB I/F の場合	

戻り値

説明	型
Star プリンタの検索結果 戻り値の NSArray には、PortInfo クラスのインスタンスが含まれます。 取得できる情報については PortInfo クラス を参照ください。	NSArray *

1. 検索実行時間

検索したいインターフェイスによって、searchPrinter メソッドの実行時間が異なります。

インターフェイス	実行時間
すべてのインターフェイス	7[s]
Bluetooth I/F	すぐに制御を返します。
Bluetooth Low Energy I/F	7[s]
Ethernet / Wireless LAN I/F	6[s]
USB I/F	すぐに制御を返します。

2. 制限事項

本 API はデバイスを確実に検出する事を保証するものではありません。

3. BLE インターフェイスでの注意事項

初めて searchPrinter メソッドでプリンタ情報を取得した際、portName が@"BLE:"となる場合があります。この場合、一旦 getPort メソッドでプリンタに接続を行ってください。これ以降、正しいデバイス名を取得、使用可能になります。

Example

//The following would be an actual usage of searchPrinter:

```
NSArray *portArray = [[SMPort searchPrinter] retain];
```

```
for (int i = 0; i < portArray.count; i++) {  
    PortInfo *port = [portArray objectAtIndex:i];  
    NSLog(@"Port Name: %@", port.portName);  
    NSLog(@"MAC Address : %@", port.macAddress);  
    NSLog(@"Model Name: %@", port.modelName);  
}  
  
[portArray release];
```

3.6 writePort メソッド

プリンタにデータを書き込みます。コマンドや印刷データの送信に使用します。
印字終了の確認を行うため、このメソッドの前後で beginCheckedBlock/endCheckedBlock を使用してください。

宣言

```
-(u_int32_t) writePort:(u_int8_t const *) writeBuffer :(u_int32_t) offset :(u_int32_t) size;
```

引数

名称	説明	型
writeBuffer	書き込みたいデータを格納する byte 配列	u_int8_t const *
offset	writeBuffer のどの位置から書き込み始めるかを指定するインデックス	u_int32_t
size	書き込む byte 数	u_int32_t

戻り値

説明	型
Bluetooth/Ethernet/Wireless LAN/USB インターフェイス 実際に書き込まれたバイト数。 データが全て書き込めなかった時でも、writePort メソッドは成功します。アプリケーションは期待されるデータが全て読み取れるまで、writePort メソッドを複数回呼び出す必要があります。または、しきい値に達するまで再試行をするようにします。	u_int32_t
Bluetooth Low Energy インターフェイス 成功時には、送信データのサイズが返ります。失敗時は 0 が返ります。	

例外

説明	型
通信エラーが発生したとき	PortException

印刷の手順については [SMPort クラスを使った印刷処理フロー](#) を参照ください。

3.7 readPort メソッド

プリンタからデータを読み込みます。プリンタから Raw byte を読み取る必要のある場合のみ、ご使用ください。

宣言

```
-(u_int32_t) readPort:(u_int8_t *) readBuffer :(u_int32_t) offSet :(u_int32_t) size;
```

引数

名称	説明	型
readBuffer	データが読み込まれる byte 配列	u_int8_t *
offset	readBuffer にデータを書き込み始めるインデックス	u_int32_t
size	読み込む byte 数	u_int32_t

戻り値

説明	型
実際に読み込まれたバイト数。 データが全て読み取れなかった時でも、readPort メソッドは成功します。アプリケーションは期待されるデータが全て読み取れるまで、readPort メソッドを複数回呼び出す必要があります。または、しきい値に達するまで再試行をするようにします。	u_int32_t

例外

説明	型
通信エラーが発生したとき	PortException

1. 制限事項

Raw Status の取得にこのメソッドを使用しないでください。ステータスの取得には getParsedStatus メソッドを使用してください。

3.8 getParsedStatus メソッド

プリンタのステータスを取得します。

宣言

- (void) getParsedStatus:(void *) starPrinterStatus :(u_int32_t) level;

引数

名称	説明	型
starPrinterStatus	現在のプリンタのステータスが格納されます。 取得できるステータスの種類は StarPrinterStatus 構造体 を参照ください。	void *
level	StarPrinterStatus 構造体のレベル (0,1,2 の指定が可能だが、通常は 2 を指定)	u_int32_t

戻り値

説明	型
-	-

例外

説明	型
通信エラーが発生したとき	PortException

Example

```
StarPrinterStatus_2 printerStatus;
```

```
[port getParsedStatus: &printerStatus : 2];
```

```
if (printerStatus.offline == SM_TRUE)
{
    if (printerStatus.coverOpen == SM_TRUE) {
        //There was a cover open error
    }
    else if (printerStatus.receiptPaperEmpty == SM_TRUE) {
        //There was a receipt paper empty error
    }
    else {
        //There was a offline error
    }
}
else {
    //If False, then the printer is online.
}
```

3.9 beginCheckedBlock メソッド

endCheckedBlock メソッドとセットで使用して印字終了の監視を行います。

最後まで正しく印刷されたか確認したい場合に、印刷データ送信の直前に本メソッドを、印刷データ送信の直後に endCheckedBlock メソッドを実行します。

宣言

```
- (void) beginCheckedBlock:(void *) starPrinterStatus :(u_int32_t) level;
```

引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ（StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定） このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	void *
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	u_int32_t

戻り値

説明	型
-	-

beginCheckedBlock/endCheckedBlock による印刷終了監視処理の手順は [SMPort クラスを使った印刷処理フロー](#) を参照ください。

3.10 endCheckedBlock メソッド

beginCheckedBlock メソッドとセットで使用して印字終了の監視を行います。
 プリンタの状態を監視し、送信した印刷データの印刷が完了すると制御を返します。
 印刷データ以外を送信した場合は、そのコマンドがプリンタに処理されると制御を返します。
 endCheckedBlockTimeoutMillis プロパティで指定したタイムアウト時間内に印刷が完了しなかった場合や、印刷中にプリンタエラーが発生した場合は、例外 PortException をスローします。

宣言

- (void) endCheckedBlock:(void *) starPrinterStatus :(u_int32_t) level;

引数

名称	説明	型
starPrinterStatus	StarPrinterStatus 構造体へのポインタ（StarPrinterStatus, StarPrinterStatus_1, StarPrinterStatus_2 の指定が可能だが、通常は 2 を指定） このメソッドが成功したとき、現在のプリンタのステータスが格納されます。	void *
level	StarPrinterStatus 構造体のレベル（0,1,2 の指定が可能だが、通常は 2 を指定）	u_int32_t

戻り値

説明	型
-	-

例外

説明	型
通信エラー（プリンタオフライン等でコマンド送信自体が失敗した場合）が発生したとき タイムアウト時間内にプリンタからの応答がないとき	PortException

1. タイムアウト時間

タイムアウト時間は、endCheckedBlockTimeoutMillis プロパティの値が使用されます。
 初期値は getPort で指定したタイムアウト時間となります。
 endCheckedBlockTimeoutMillis の値は、印刷時間より長くなるよう調整してください。
 また、10 秒未満の値が設定された場合にはタイムアウトは 10 秒になります。

beginCheckedBlock/endCheckedBlock による印刷終了監視処理の手順は [SMPort クラスを使った印刷処理フロー](#) を参照ください。

3.11 disconnect メソッド

指定された Bluetooth デバイスへのコネクションを切断します。
コネクションの切断後、Bluetooth デバイスは再び他の iOS 端末から接続することができるようになります。

宣言

- (BOOL) disconnect;

引数

名称	説明	型
-	-	-

戻り値

説明	型
切断成功・失敗 このメソッドは、以下の場合に失敗となります。 <ul style="list-style-type: none"> ● getPort で指定したタイムアウト時間内に切断が完了しなかった場合 ● プリンタが切断機能に対応していない場合（モバイルプリンタ等） Ethernet デバイスに対しては何も行いません。 Ethernet デバイスに対して実行した場合は常に YES を返します。	BOOL

3.12 getFirmwareInformation メソッド

プリンタのモデル名とファームウェアバージョンを取得します。

宣言

-(NSDictionary *) getFirmwareInformation;

引数

名称	説明	型
-	-	-

戻り値

説明	型
ファームウェア情報取得結果の NSDictionary 戻り値の NSDictionary のキーに @ModelName を設定することでモデル名を、 @FirmwareVersion を設定することでファームウェアバージョンを取得できます。	NSDictionary *

例外

説明	型
取得に失敗したとき	PortException

3.13 StarIOVersion メソッド

StarIO ライブラリのバージョンを取得します。

宣言

```
+ (NSString *) StarIOVersion;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
StarIO ライブラリのバージョン文字列	NSString *

3.14 portName プロパティ

プリンタと通信するためのポートを特定する文字列

宣言

```
- (NSString *) portName;
```

3.15 portSettings プロパティ

プリンタとの接続設定情報

宣言

```
- (NSString *) portSettings;
```

3.16 timeoutMillis プロパティ

内部制御および API のタイムアウト値（ミリ秒指定）

指定した時間が 10 秒未満の場合、タイムアウトは 10 秒になります。

宣言

```
- (u_int32_t) timeoutMillis;
```


3.17 endCheckedBlockTimeoutMillis プロパティ

endCheckedBlock メソッドのタイムアウト値（ミリ秒指定）

印刷に時間がかかる場合、この値を大きくする事で endCheckedBlock メソッドの印刷完了待ち時間を伸ばす事ができます。

初期値は、getPort メソッドの引数で指定したタイムアウト時間と同じになります。

指定した時間が 10 秒未満の場合、タイムアウトは 10 秒になります。

また、getPort メソッドの portSettings パラメータで[データタイムアウト機能の設定]を使用する場合、このメソッドで設定するタイムアウト時間には、データタイムアウト機能の指定時間より 3 秒以上長くなるように設定してください。3 秒未満に設定した場合、内部制御により自動的に 3 秒長くなるように設定されます。

宣言

@property (assign, readwrite, nonatomic) u_int32_t endCheckedBlockTimeoutMillis;

3.18 connected プロパティ

プリンタが iOS デバイスと接続されていれば "YES" を、そうでなければ "NO" を返します。

Bluetooth インターフェイスのみ対応しています。非対応のインターフェイスでは、常に "Yes" を返します。

Bluetooth 通信が切断されてからこのプロパティに反映されるまで、iOS デバイスの制限により 5 秒程度かかります。

宣言

- (BOOL) connected;

4 StarPrinterStatus 構造体 (StarIO.framework)

プリンタのステータスを保持する構造体です。

boolean 型とバイナリ(raw バイト配列)の両方の形式でプリンタステータスを保持します。取得できるステータスの種類はメンバー一覧をご覧ください。

メンバ

名称	説明	型	詳細
blackMarkError	ブラックマークエラー	SM_BOOLEAN	ブラックマークエラー(ブラックマーク設定時に非ブラックマーク用紙を使って印刷した場合等に発生)の時に SM_TRUE となる。通常時は SM_FALSE。
compulsionSwitch	コンパルジョンスイッチ	SM_BOOLEAN	ドロワのコンパルジョン SW が押されていると SM_TRUE となる。通常は SM_FALSE。
coverOpen	カバーオープン	SM_BOOLEAN	カバーが開いている場合に SM_TRUE となる。閉じている場合は SM_FALSE。
cutterError	オートカッターエラー	SM_BOOLEAN	カッターエラー発生時に SM_TRUE となる。
etbAvailable	ETB 使用可否	SM_BOOLEAN	ETB が使用可能な場合に SM_TRUE となる。使用できない場合は SM_FALSE。
etbCounter	ETB カウンタ	UCHAR	現在の ETB カウンタの値。
headThermistorError	ヘッドサーミスタエラー	SM_BOOLEAN	ヘッドサーミスタ異常値検出時に SM_TRUE となる。通常は SM_FALSE。
offline	オフライン	SM_BOOLEAN	オフラインの場合に SM_TRUE となる。オンライン時は SM_FALSE。
overTemp	印字ヘッド高温による停止中	SM_BOOLEAN	ヘッドが高温になり印刷停止している状態で SM_TRUE となる。通常は SM_FALSE。
raw	ステータスバイト列	UCHAR[63]	ステータスのバイト列 (例: HEX 23 86 00 00 00 00 00 00 00)
rawLength	raw の長さ	CHAR	raw の長さ
receiptPaperEmpty	用紙エンド	SM_BOOLEAN	用紙切れの場合は SM_TRUE となる。通常は SM_FALSE。
receiptPaperNearEmptyInner	用紙ニアエンド(内側)	SM_BOOLEAN	用紙ニアエンド状態の時に SM_TRUE となる。通常は SM_FALSE。
receiveBufferOverflow	受信バッファオーバーフロー	SM_BOOLEAN	受信バッファフルの時に SM_TRUE となる。通常は SM_FALSE。

名称	説明	型	詳細
unrecoverableError	復帰不可能エラー	SM_BOOLEAN	復帰不可能エラー(ヘッドサーミスタエラー、オートカッターエラー、電源電圧エラー等)が発生した場合に SM_TRUE となる。通常は SM_FALSE。
voltageError	電源電圧エラー	SM_BOOLEAN	電源電圧異常値検出時に SM_TRUE となる。通常は SM_FALSE。

4.1 モデル: StarPrinterStatus 構造体

各モデルにおいて機能するメンバは以下のとおりです。

Field	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
blackMarkError	-	✓	-	-	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
compulsionSwitch	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
coverOpen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
cutterError	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
etbAvailable	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
etbCounter	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
headThermistorError	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
offline	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
overTemp	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
raw	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
rawLength	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
receiptPaperEmpty	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
receiptPaperNearEmptyInner	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
receiveBufferOverflow	-	✓	-	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
unrecoverableError	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
voltageError	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

-: 機能しない

5 PortInfo クラス (StarIO.framework)

Star プリンタと接続するための情報やモデル情報を保持するクラスです。

プロパティ

名称	説明
portName	プリンタと通信するためのポートを特定する文字列。
macAddress	プリンタの MAC アドレスを表す文字列。
modelName	プリンタのモデル名を表す文字列。

5.1 portName プロパティ

プリンタと通信するためのポートを特定する文字列。

[getPort メソッド](#)で接続先プリンタを指定するのに使用できます。

宣言

```
@property (retain, readonly) NSString *portName;
```

5.2 macAddress プロパティ

プリンタの MAC アドレスを表す文字列。

宣言

```
@property (retain, readonly) NSString *portName;
```

5.3 modelName プロパティ

プリンタのモデル名を表す文字列。

宣言

```
@property (retain, readonly) NSString *portName;
```

6 SMBluetoothManager クラス (StarIO.framework)

SMBluetoothManager クラスは、Bluetooth インターフェイスの各種設定を行うためのクラスです。SMPort クラスと同時に使用しないでください。

メソッド

名称	説明
initWithName	SMBluetoothManager のインスタンスを作成します。
open	StarBluetooth デバイスとの接続を開きます。
loadSetting	接続した StarBluetooth デバイスから設定を取得します。
close	StarBluetooth デバイスとの接続を閉じます。
apply	StarBluetooth デバイスに指定した値を設定します。

プロパティ

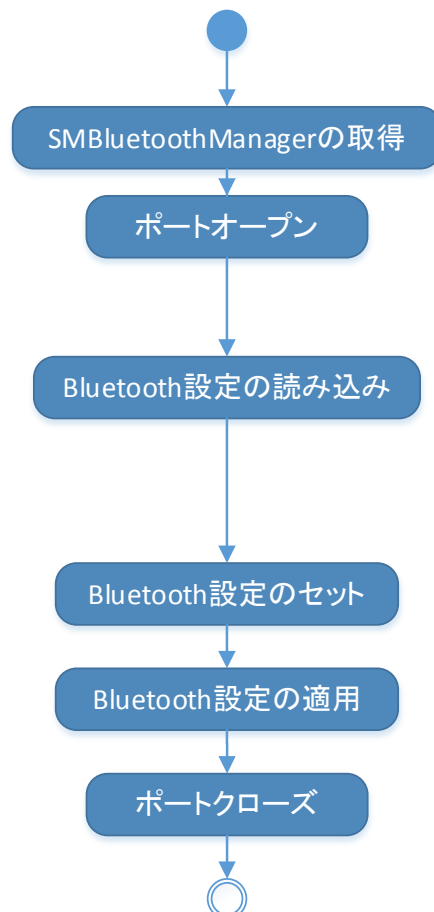
名称	説明
portName	プリンタと通信するためのポートを特定する文字列
deviceType	接続先プリンタの種類
opened	ポートがオープンされているかどうか
deviceName	Bluetooth デバイス名
iOSPortName	iOSPort 名
autoConnect	AutoConnection（自動接続機能）のオン/オフ
security	Bluetooth セキュリティタイプ
pinCode	Bluetooth ペアリング時に使用する PIN コード
deviceNameCapability	Bluetooth デバイス名の設定可否情報
iOSPortNameCapability	iOSPort 名の設定可否情報
autoConnectCapability	AutoConnection の設定可否情報
securityTypeCapability	Bluetooth セキュリティタイプの設定可否情報
pinCodeCapability	PIN コード設定可否情報

コンスタント

名称	説明
SMDeviceType	プリンタタイプ定数。
SMBluetoothSecurity	Bluetooth セキュリティタイプ定数。
SMBluetoothSettingCapability	Bluetooth 設定可否定数。

6.1 SMBluetoothManager を使った Bluetooth 設定変更フロー

SMBluetoothManager クラスを利用し、以下の手順に沿ってプリンタの Bluetooth 設定変更を行います。



```

// StarBluetoothManager の取得
SMBluetoothManager *bluetoothManager = [SMBluetoothManagerFactory getManager:portName emulation:emulation];

// ポートオープン
if ([bluetoothManager open] == NO) {
    // Fail to open port.
    return;
}

// Bluetooth 設定の読み込み
if ([bluetoothManager loadSetting] == NO) {
    // Fail to load settings.
    [bluetoothManager close];
    return;
}

// Bluetooth 設定のセット
// 例) デバイス名を"Star Micronics"に設定したい場合
bluetoothManager.deviceName = @"Star Micronics";

// Bluetooth 設定の適用
if ([bluetoothManager apply] == NO) {
    // Fail to apply settings.
    [bluetoothManager close];
    return;
}

// ポートクローズ
[bluetoothManager close];
  
```

BluetoothSettingViewController.m を参照ください。

6.2 モデル: SMBluetoothManager クラス

各モデルにおいて機能するメソッドは以下のとおりです。

Method	mPOP *1	FVP10	TSP100 (TSP100IIIのみ)	TSP650II	TSP700II	TSP800II	SM-S210i *2	SM-S220i *2	SM-S230i *2	SM-T300i *2	SM-T400i *2	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
initWithName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
open	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
loadSetting	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
close	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
apply	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓

- : 機能しない

*1 : ファームウェアバージョン 1.1 以降が必要です

*2 : ファームウェアバージョン 3.0 以降が必要です

各モデルにおいて設定できる（apply メソッド実行時に Bluetooth デバイスに反映される）プロパティは以下のとおりです。

Property	mPOP *1	FVP10	TSP100 (TSP100IIIのみ)	TSP650II	TSP700II	TSP800II	SM-S210i *2	SM-S220i *2	SM-S230i *2	SM-T300i *2	SM-T400i *2	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
deviceName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
iOSPortName	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
pinCode	-	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
autoConnect	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓
security	-	*3	-	*3	*3	*3	*4	*4	*4	*4	*4	-	*4	*4	*4	*4	*4	*4	*4	*3

- : 機能しない

*1 : ファームウェアバージョン 1.1 以降が必要です

*2 : ファームウェアバージョン 3.0 以降が必要です

*3 : 設定・取得できる security は PIN コードもしくは SSP

*4 : 設定・取得できる security は PIN コードもしくは Disable

6.3 initWithName メソッド

SMBluetoothManager のインスタンスを作成します。

宣言

```
-(id) initWithPortName: (NSString *) portName deviceType: (SMDeviceType) deviceType;
```

引数

名称	説明	型
portName	プリンタと通信するためのポートを特定する文字列 SMPort クラスの getPort メソッド の portName と同じです。	NSString *
deviceType	プリンタタイプ定数 mPOP, TSP100III BI を使用する場合 SMDeviceTypePortablePrinter を指定してください。	SMDeviceType

戻り値

説明	型
成功時は SMBluetoothManager のインスタンスを返します。 失敗時は nil を返します。	id

Bluetooth 設定変更の手順は [SMBluetoothManager を使った Bluetooth 設定変更フロー](#)を参照ください。

1. SMBluetoothManagerFactory クラス

[SMBluetoothManagerFactory クラス](#)は、プリンタのエミュレーションに対して適切な SMBluetoothManager オブジェクトを取得することができます。

6.4 open メソッド

StarBluetooth デバイスとの接続を開きます。

open メソッド実行後は、必ず loadSetting メソッドで現在の設定を取得してください。

宣言

```
-(BOOL) open;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
成功・失敗	BOOL

Bluetooth 設定変更の手順は [SMBluetoothManager を使った Bluetooth 設定変更フロー](#)を参照ください。

6.5 loadSetting メソッド

接続した StarBluetooth デバイスから設定を取得します。

宣言

- (BOOL) loadSetting;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Bluetooth 設定変更の手順は [SMBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

6.6 close メソッド

StarBluetooth デバイスとの接続を閉じます。

宣言

- (void) close;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Bluetooth 設定変更の手順は [SMBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

6.7 apply メソッド

deviceName、iOSSPortName、autoConnect、security、pinCode プロパティの値をデバイスに適用します。

宣言

- (BOOL) apply;

引数

名称	説明	型
-	-	-

戻り値

説明	型
成功・失敗	BOOL

1. apply メソッドで適用した値は、デバイスの電源再投入・再ペアリングを行った後に有効になります。

Bluetooth 設定変更の手順は [SMBluetoothManager を使った Bluetooth 設定変更フロー](#)を参照ください。

6.8 portName プロパティ

プリンタと通信するためのポートを特定する文字列

宣言

@property (nonatomic, readonly) NSString *portName;

6.9 deviceType プロパティ

接続先プリンタの種類

宣言

@property(nonatomic, readonly) SMDeviceType deviceType;

6.10 opened プロパティ

ポートがオープンされているかどうか

open メソッドが成功すると YES になります。その後 close メソッドを呼び出すと NO になります。

宣言

@property(nonatomic, readonly) BOOL opened;

6.11 deviceName プロパティ

Bluetooth デバイス名

この名前は、Bluetooth ではペアリング時に表示されます。

Bluetooth Low Energy では、通信時の接続ポート名として使用されます。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

宣言

```
@property(nonatomic, retain) NSString *deviceName;
```

1. 使用可能文字

0-9 a-z A-Z ; : ! ? # \$ % & , . @ _ - = Space / * + ~ ^ [{ () }] \

2. 文字数制限

1 文字以上かつ 16 文字以下

3. 注意事項

Bluetooth Low Energy の場合、変更した Bluetooth デバイス名は、デバイスの電源再投入・再接続を行った後に有効になります。

6.12 iOSPortName プロパティ

iOSPort 名

StarIO で Bluetooth 通信時に使用するポート名を取得、設定します。

Bluetooth Low Energy では使用されません。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

宣言

```
@property(nonatomic, retain) NSString *iOSPortName;
```

1. 使用可能文字

0-9 a-z A-Z ; : ! ? # \$ % & , . @ _ - = Space / * + ~ ^ [{ () }] \

2. 文字数制限

1 文字以上かつ 16 文字以下

6.13 autoConnect プロパティ

AutoConnection（自動接続機能）のオン／オフ

本機能は Bluetooth インターフェイスのみ対応しています。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

宣言

```
@property(nonatomic, assign) BOOL autoConnect;
```

1. 注意事項

security プロパティが PIN コード設定の場合は、この値に NO を設定してください。

6.14 security プロパティ

Bluetooth セキュリティタイプ

本機能は Bluetooth のみ対応しています。

loadSetting メソッドを呼び出した際に現在の設定値が読み込まれます。

設定するには、本プロパティを変更後 apply メソッドを実行します。

宣言

```
@property(n nonatomic, assign) SMBluetoothSecurity security;
```

6.15 pinCode プロパティ

Bluetooth ペアリング時に使用する PIN コード

本機能は Bluetooth のみ対応しています。

現在の設定値を取得することはできません。

現在の PIN コードから値を変更しない場合は nil を指定します。

宣言

```
@property(n nonatomic, retain) NSString *pinCode;
```

1. 使用可能文字

0-9 (SM-L200, SM-L300)

0-9 a-z A-Z (その他のモデル)

2. 文字数制限

4 文字 (SM-L200, SM-L300)

4 文字以上かつ 16 文字以下 (その他のモデル)

6.16 deviceNameCapability プロパティ

Bluetooth デバイス名の設定可否情報

宣言

```
@property (assign, readonly) SMBluetoothSettingCapability deviceNameCapability;
```

6.17 iOSPortNameCapability プロパティ

iOSPort 名の設定可否情報

宣言

```
@property (assign, readonly) SMBluetoothSettingCapability iOSPortNameCapability;
```

6.18 autoConnectCapability プロパティ

AutoConnection の設定可否情報

宣言

```
@property (assign, readonly) SMBluetoothSettingCapability autoConnectCapability;
```

6.19 securityTypeCapability プロパティ

Bluetooth セキュリティタイプの設定可否情報

宣言

@property (assign, readonly) SMBluetoothSettingCapability securityTypeCapability;

6.20 pinCodeCapability プロパティ

PIN コード設定可否情報

宣言

@property (assign, readonly) SMBluetoothSettingCapability pinCodeCapability;

6.21 SMDeviceType コンスタント

プリンタタイプ定数。

宣言

```
typedef enum _SMDeviceType {
    SMDeviceTypeUnknown = 0,
    SMDeviceTypeDesktopPrinter,
    SMDeviceTypePortablePrinter,
    SMDeviceTypeDKAirCash,
} SMDeviceType;
```

定数

名称	説明
SMDeviceTypeDesktopPrinter	デスクトッププリンタ
SMDeviceTypePortablePrinter	モバイルプリンタ mPOP, TSP100IIIIBI を使用する場合 SMDeviceTypePortablePrinter を指定してください。

6.22 SMBluetoothSecurity コンスタント

Bluetooth セキュリティタイプ定数。

宣言

```
typedef enum _SMBluetoothSecurity {
    SMBluetoothSecurityDisable,
    SMBluetoothSecuritySSP,
    SMBluetoothSecurityPINcode
} SMBluetoothSecurity;
```

定数

名称	説明
SMBluetoothSecurityPINcode	PIN コード入力によるセキュリティ
SMBluetoothSecuritySSP	インターフェイスカードのボタン押下によるセキュリティ
SMBluetoothSecurityDisable	セキュリティなし

6.23 SMBluetoothSettingCapability コンスタント

Bluetooth 設定可否定数。

宣言

```
typedef enum _SMBluetoothSettingCapability {
    SMBluetoothSettingCapabilitySupport,
    SMBluetoothSettingCapabilityNoSupport
} SMBluetoothSettingCapability;
```

定数

名称	説明
SMBluetoothSettingCapabilitySupport	現在接続中のプリンタではその項目・機能が設定できることを示す。
SMBluetoothSettingCapabilityNoSupport	現在接続中のプリンタではその項目・機能が設定できないことを示す。

7 StarloExt クラス (StarIO_Extension.framework)

メソッド

名称	説明
createCommandBuilder	コマンド生成オブジェクトを生成します。
createDisplayCommandBuilder	カスタマーディスプレイコマンド生成オブジェクトを生成します。
createScaleCommandBuilder	スケールコマンド生成オブジェクトを生成します。
createBcrConnectParser	バーコードリーダー接続確認コマンド応答解析オブジェクトを生成します。
createDisplayConnectParser	カスタマーディスプレイ接続確認コマンド応答解析オブジェクトを生成します。
createScaleConnectParser	スケール接続確認コマンド応答解析オブジェクトを生成します。
createScaleWeightParser	スケール表示重量取得コマンド応答解析オブジェクトを生成します。

コンスタント

名称	説明
StarloExtEmulation	エミュレーション指定定数。
StarloExtCharacterCode	文字コード指定定数。
StarloExtBcrModel	バーコードリーダーモデル指定定数。
StarloExtScaleModel	スケールモデル指定定数。
StarloExtDisplayModel	カスタマーディスプレイモデル指定定数。

7.1 createCommandBuilder メソッド

コマンド生成オブジェクトを生成します。

宣言

```
+ (ISCBBuilder *)createCommandBuilder:(StarloExtEmulation)emulation;
```

引数

名称	説明	型
emulation	エミュレーション指定 <ul style="list-style-type: none"> StarloExtEmulationStarPRNT ... StarPRNT エミュレーション StarloExtEmulationStarLine ... STAR Line Mode エミュレーション StarloExtEmulationStarGraphic ... STAR Graphic Mode エミュレーション StarloExtEmulationEscPos ... ESC/POS エミュレーション StarloExtEmulationEscPosMobile ... ESC/POS Mobile エミュレーション StarloExtEmulationStarDotImpact ... STAR ドットインパクトエミュレーション StarloExtEmulationStarPRNTL... StarPRNTL エミュレーション 	StarloExtEmulation

戻り値

説明	型
ISCBBuilder オブジェクト	ISCBBuilder *

Example

```
+ (NSData *)createCommandImage:(StarloExtEmulation)emulation image:(UIImage *)image {
    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

PrinterFunctions.m を参照ください。

7.2 createDisplayCommandBuilder メソッド

カスタマーディスプレイコマンド生成オブジェクトを生成します。

宣言

```
+ (ISDCBBuilder *)createDisplayCommandBuilder:(StarloExtDisplayModel)model;
```

引数

名称	説明	型
model	カスタマーディスプレイモデル指定定数 ✱ SCD222	StarloExtDisplayModel

戻り値

説明	型
ISDCBBuilder オブジェクト	ISDCBBuilder *

Example

```
- (void)alertView:(UIAlertView *)alertView didDismissWithButtonIndex:(NSInteger)buttonIndex {
    if (buttonIndex != alertView.cancelButtonIndex) {
        ISDCBBuilder *builder = [StarloExt createDisplayCommandBuilder:StarloExtDisplayModelSCD222];
        ...
    }
}
```

DisplayViewController.m / DisplayExtViewController.m を参照ください。

7.3 createScaleCommandBuilder メソッド

スケールコマンド生成オブジェクトを生成します。

宣言

```
+ (ISSCBBUILDER *)createScaleCommandBuilder:(StarIoExtScaleModel)model;
```

引数

名称	説明	型
model	スケールモデル指定定数 • APS10 • APS12 • APS20	StarIoExtScaleModel

戻り値

説明	型
ISSCBBUILDER オブジェクト	ISSCBBUILDER *

Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    ...
    else if (indexPath.row == 2) {
        ISSCBBUILDER *builder = [StarIoExt createScaleCommandBuilder:StarIoExtScaleModelAPS20];

        [ScaleFunctions appendZeroClear:builder];

//        NSData *commands = [builder.commands copy];
        NSData *commands = [builder.passThroughCommands copy];

        [Communication sendCommandsDoNotCheckCondition:commands port:port
        completionHandler:^(BOOL result, NSString *title, NSString *message) {
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:title message:message delegate:nil
            cancelButtonTitle:@"OK" otherButtonTitles:nil];

            [alertView show];
        }];
    }
    ...
}
```

ScaleViewController.m / ScaleExtViewController.m を参照ください。

7.4 createBcrConnectParser メソッド

バーコードリーダー接続確認コマンド応答解析オブジェクトを生成します。

宣言

```
+ (ISCPConnectParser *)createBcrConnectParser:(StarloExtBcrModel)model;
```

引数

名称	説明	型
model	バーコードリーダーモデル指定定数 ● POP1	StarloExtBcrModel

戻り値

説明	型
バーコードリーダー接続確認コマンド応答解析オブジェクト	ISCPConnectParser *

7.5 createDisplayConnectParser メソッド

カスタマーディスプレイ接続確認コマンド応答解析オブジェクトを生成します。

宣言

```
+ (ISCPConnectParser *)createDisplayConnectParser:(StarloExtDisplayModel)model;
```

引数

名称	説明	型
model	カスタマーディスプレイモデル指定定数 ● SCD222	StarloExtDisplayModel

戻り値

説明	型
カスタマーディスプレイ接続確認コマンド応答解析オブジェクト	ISCPConnectParser *

Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    ...
    @try {
        port = [SMPort getPort:[AppDelegate getPortName] :[AppDelegate getPortSettings] :10000]; // 10000mS!!!

        if (port != nil) {
            ISCPConnectParser *parser = [StarloExt createDisplayConnectParser:StarloExtDisplayModelSCD222];

            [Communication parseDoNotCheckCondition:parser port:port completionHandler:^(BOOL result, NSString
            *title, NSString *message) {
                if (result == YES) {
                    if (parser.connect == YES) {
                        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Check Status" message:@"Display
                        Connect." delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];

                        [alertView show];
                    }
                    else {
                        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Check Status" message:@"Display
                        Disconnect." delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];

                        [alertView show];
                    }
                }
            }];
            ...
        }
    }
    ...
}
```

DisplayViewController.m / DisplayExtViewController.m を参照ください。

7.6 createScaleConnectParser メソッド

スケール接続確認コマンド応答解析オブジェクトを生成します。

宣言

```
+ (ISCPConnectParser *)createScaleConnectParser:(StarIoExtScaleModel)model;
```

引数

名称	説明	型
model	スケールモデル指定定数 • APS10 • APS12 • APS20	StarIoExtScaleModel

戻り値

説明	型
スケール接続確認コマンド応答解析オブジェクト	ISCPConnectParser *

Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    [tableView deselectRowAtIndexPath:indexPath animated:YES];

    if (indexPath.section == 0) {
        self.blind = YES;

        SMPort *port = nil;

        @try {
            port = [SMPort getPort:[AppDelegate getPortName] :[AppDelegate getPortSettings] :10000];    // 10000mS!!!

            if (port != nil) {
                ISCPConnectParser *parser = [StarIoExt createScaleConnectParser:StarIoExtScaleModelAPS20];

                [Communication parseDoNotCheckCondition:parser port:port completionHandler:^(BOOL result, NSString *title,
                NSString *message) {
                    ...
                }
            }
        }
    }
}
```

ScaleViewController.m / ScaleExtViewController.m を参照ください。

7.7 createScaleWeightParser メソッド

スケール表示重量取得コマンド応答解析オブジェクトを生成します。

宣言

```
+ (ISSCPWeightParser *)createScaleWeightParser:(StarIoExtScaleModel)model;
```

引数

名称	説明	型
model	スケールモデル指定定数 <ul style="list-style-type: none"> • APS10 • APS12 • APS20 	StarIoExtScaleModel

戻り値

説明	型
スケール表示重量コマンド応答解析オブジェクト	ISSCPWeightParser *

Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    ...
    ISSCPWeightParser *weightParser = [StarIoExt createScaleWeightParser:StarIoExtScaleModelAPS20];

    [ScaleCommunication parseDoNotCheckCondition:weightParser port:port completionHandler:^(BOOL result, NSString
    *title, NSString *message) {
        ...
    }
}
```

ScaleViewController.m / ScaleExtViewController.m を参照ください。

7.8 StarloExtEmulation コンスタント

エミュレーション指定定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtEmulation) {
    StarloExtEmulationNone = 0,
    StarloExtEmulationStarPRNT,
    StarloExtEmulationStarLine,
    StarloExtEmulationStarGraphic,
    StarloExtEmulationEscPos,
    StarloExtEmulationEscPosMobile,
    StarloExtEmulationStarDotImpact,
    StarloExtEmulationStarPRNTL
};
```

定数

名称	説明
StarloExtEmulationStarPRNT	StarPRNT エミュレーション
StarloExtEmulationStarLine	STAR Line Mode エミュレーション
StarloExtEmulationStarGraphic	STAR Graphic Mode エミュレーション
StarloExtEmulationEscPos	ESC/POS エミュレーション
StarloExtEmulationEscPosMobile	ESC/POS Mobile エミュレーション
StarloExtEmulationStarDotImpact	STAR ドットインパクトエミュレーション
StarloExtEmulationStarPRNTL	StarPRNTL エミュレーション

1. StarPRNTL エミュレーションについて

ラスタイメージ印字コマンドを生成する [appendBitmap](#) 系メソッドでコマンドサイズを削減した StarPRNT エミュレーションです。ラスタイメージコマンド送信に掛かる時間が短縮されます。StarPRNT エミュレーションをサポートするモデルのうち、一部モデルのみ選択可能です。使用可能なモデルは[対応プリンタ](#)を参照ください。

7.9 StarloExtCharacterCode コンスタント

文字コード指定定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtCharacterCode) {
    StarloExtCharacterCodeNone = 0,
    StarloExtCharacterCodeStandard,
    StarloExtCharacterCodeJapanese,
    StarloExtCharacterCodeSimplifiedChinese,
    StarloExtCharacterCodeTraditionalChinese
};
```

定数

名称	説明
StarloExtCharacterCodeStandard	スタンダード
StarloExtCharacterCodeJapanese	日本語
StarloExtCharacterCodeSimplifiedChinese	簡体字中国語
StarloExtCharacterCodeTraditionalChinese	繁体字中国語

7.10 StarloExtBcrModel コンスタント

バーコードリーダーモデル指定定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtBcrModel) {
    StarloExtBcrModelNone = 0,
    StarloExtBcrModelPOP1
};
```

定数

名称	説明
StarloExtBcrModelPOP1	mPOP 指定バーコードリーダー BCR-POP1

7.11 StarloExtScaleModel コンスタント

スケールモデル指定定数。

宣言

```
c typedef NS_ENUM(NSInteger, StarloExtScaleModel) {
    StarloExtScaleModelNone = 0,
    StarloExtScaleModelAPS10,
    StarloExtScaleModelAPS12,
    StarloExtScaleModelAPS20
};
```

定数

名称	説明
StarloExtScaleModelAPS10	mPOP スケール（はかり）APS10 ※U.S.限定
StarloExtScaleModelAPS12	mPOP スケール（はかり）APS12 ※U.S.限定
StarloExtScaleModelAPS20	mPOP スケール（はかり）APS20 ※U.S.限定

7.12 StarloExtDisplayModel コンスタント

カスタマーディスプレイモデル指定定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtDisplayModel) {
    StarloExtDisplayModelNone = 0,
    StarloExtDisplayModelSCD222
};
```

定数

名称	説明
StarloExtDisplayModelSCD222	mPOP 指定カスタマーディスプレイ SCD222U

7.13 モデル : portSetting and StarloExtEmulation

各モデルに対して適用すべき portSettings 文字列と StarloExtEmulation 定数は以下のとおりです。

Model	portSettings	StarloExtEmulation
mPOP	""	StarloExtEmulationStarPRNT
FVP10	""	StarloExtEmulationStarLine
TSP100	""	StarloExtEmulationStarGraphic
TSP650II	""	StarloExtEmulationStarLine
TSP700II	""	StarloExtEmulationStarLine
TSP800II	""	StarloExtEmulationStarLine
SM-S210i	"mini"	StarloExtEmulationEscPosMobile
SM-S220i	"mini"	StarloExtEmulationEscPosMobile
SM-S230i	"mini"	StarloExtEmulationEscPosMobile
SM-T300i	"mini"	StarloExtEmulationEscPosMobile
SM-T400i	"mini"	StarloExtEmulationEscPosMobile
BSC10	"escpos"	StarloExtEmulationEscPos
SM-S210i StarPRNT	"Portable"	StarloExtEmulationStarPRNT
SM-S220i StarPRNT	"Portable"	StarloExtEmulationStarPRNT
SM-S230i StarPRNT	"Portable"	StarloExtEmulationStarPRNT
SM-T300i StarPRNT	"Portable"	StarloExtEmulationStarPRNT
SM-T400i StarPRNT	"Portable"	StarloExtEmulationStarPRNT
SM-L200	"Portable"	StarloExtEmulationStarPRNT
SM-L300	"Portable"	StarloExtEmulationStarPRNTL
SP700	""	StarloExtEmulationStarDotImpact

portSettings は StarIO.framework に含まれている SMPort クラスの getPort メソッドもしくは StarIO_Extension.framework に含まれている StarloExtManager クラスの initWithType メソッドに使用します。

StarloExtEmulation は StarIO_Extension.framework に含まれている StarloExt クラスの createCommandBuilder メソッドに使用します。

8 ISCBBuilder クラス (StarIO_Extension.framework)

メソッド

名称	説明
beginDocument	ドキュメント開始コマンドを生成し、commands プロパティに追加します。
endDocument	ドキュメント終了コマンドを生成し、commands プロパティに追加します。
appendInitialization	初期化コマンドを生成し、commands プロパティに追加します。
appendByte	データ（テキストやコマンド）を commands プロパティに追加します。
appendData	
appendBytes	
appendRawByte	生のデータ（テキストやコマンド）を commands プロパティに追加します。
appendRawData	
appendRawBytes	
appendFontStyle	フォントスタイル指定コマンドを生成し、commands プロパティに追加します。
appendCodePage	コードページ指定コマンドを生成し、commands プロパティに追加します。
appendInternational	国際文字指定コマンドを生成し、commands プロパティに追加します。
appendLineFeed	行単位の用紙送りコマンドを生成し、commands プロパティに追加します。
appendDataWithLineFeed	
appendBytesWithLineFeed	
appendUnitFeed	1 ドットライン単位の用紙送りコマンドを生成し、commands プロパティに追加します。
appendDataWithUnitFeed	
appendBytesWithUnitFeed	
appendCharacterSpace	文字間スペース設定コマンドを生成し、commands プロパティに追加します。
appendLineSpace	行間スペース設定コマンドを生成し、commands プロパティに追加します。
appendEmphasis	強調指定コマンドを生成し、commands プロパティに追加します。
appendDataWithEmphasis	
appendBytesWithEmphasis	
appendInvert	白黒反転指定コマンドを生成し、commands プロパティに追加します。
appendDataWithInvert	
appendBytesWithInvert	
appendMultiple	拡大指定コマンドを生成し、commands プロパティに追加します。
appendDataWithMultiple	
appendBytesWithMultiple	
appendMultipleHeight	
appendDataWithMultipleHeight	
appendBytesWithMultipleHeight	
appendMultipleWidth	
appendDataWithMultipleWidth	
appendBytesWithMultipleWidth	
appendUnderLine	アンダーライン指定コマンドを生成し、commands プロパティに追加します。
appendDataWithUnderLine	
appendBytesWithUnderLine	
appendLogo	ロゴ印刷コマンドを生成し、commands プロパティに追加します。

名称	説明
appendAbsolutePosition	絶対位置指定コマンドを生成し、commands プロパティに追加します。
appendDataWithAbsolutePosition	
appendBytesWithAbsolutePosition	
appendAlignment	位置揃え指定コマンドを生成し、commands プロパティに追加します。
appendDataWithAlignment	
appendBytesWithAlignment	
appendCutPaper	用紙カットコマンドを生成し、commands プロパティに追加します。
appendPeripheral	ペリフェラル駆動コマンドを生成し、commands プロパティに追加します。
appendSound	サウンド駆動コマンドを生成し、commands プロパティに追加します。
appendBarcodeData	バーコード印刷コマンドを生成し、commands プロパティに追加します。
appendBarcodeBytes	
appendBarcodeDataWithAbsolutePosition	絶対位置指定バーコード印刷コマンドを生成し、commands プロパティに追加します。
appendBarcodeBytesWithAbsolutePosition	
appendBarcodeDataWithAlignment	位置揃えバーコード印刷コマンドを生成し、commands プロパティに追加します。
appendBarcodeBytesWithAlignment	
appendPdf417Data	PDF417 印刷コマンドを生成し、commands プロパティに追加します。
appendPdf417Bytes	
appendPdf417DataWithAbsolutePosition	絶対位置指定 PDF417 印刷コマンドを生成し、commands プロパティに追加します。
appendPdf417BytesWithAbsolutePosition	
appendPdf417DataWithAlignment	位置揃え PDF417 印刷コマンドを生成し、commands プロパティに追加します。
appendPdf417BytesWithAlignment	
appendQrCodeData	QR コード印刷コマンドを生成し、commands プロパティに追加します。
appendQrCodeBytes	
appendQrCodeDataWithAbsolutePosition	絶対位置指定 QR コード印刷コマンドを生成し、commands プロパティに追加します。
appendQrCodeBytesWithAbsolutePosition	
appendQrCodeDataWithAlignment	位置揃え QR コード印刷コマンドを生成し、commands プロパティに追加します。
appendQrCodeBytesWithAlignment	
appendBitmap	ビットマップ印刷コマンドを生成し、commands プロパティに追加します。
appendBitmapWithAbsolutePosition	絶対位置指定ビットマップ印刷コマンドを生成し、commands プロパティに追加します。
appendBitmapWithAlignment	位置揃えビットマップ印刷コマンドを生成し、commands プロパティに追加します。
appendBlackMark	ブラックマークコマンドを生成し、commands プロパティに追加します。
beginPageMode	ページモード開始コマンドを生成し、commands プロパティに追加します。
endPageMode	ページモード終了コマンドを生成し、commands プロパティに追加します。
appendPageModeVerticalAbsolutePosition	ページモードの縦方向絶対位置指定コマンドを生成し、commands プロパティに追加します。
appendPageModeRotation	ページモード印刷方向コマンドを生成し、commands プロパティに追加します。

プロパティ

名称	説明
commands	生成、追加されたコマンド列です。

コンスタント

名称	説明
SCBInitializationType	初期化指定定数。
SCBFontStyleType	フォントスタイル指定定数。
SCBCodePageType	コードページ指定定数。
SCBInternationalType	国際文字指定定数。
SCBLogoSize	ロゴサイズ指定定数。
SCBAlignmentPosition	位置揃え指定定数。
SCBCutPaperAction	用紙カット指定定数。
SCBPeripheralChannel	ペリフェラルチャネル指定定数。
SCBSoundChannel	サウンドチャネル指定定数。
SCBBarcodeSymbology	バーコードシンボル指定定数。
SCBBarcodeWidth	バーコード幅指定定数。
SCBPdf417Level	PDF417 ECC（セキュリティレベル）指定定数。
SCBQRCodeModel	QRコードモデル指定定数
SCBQRCodeLevel	QRコード誤り訂正レベル指定定数。
SCBBitmapConverterRotation	ビットマップ回転指定定数。
SCBBlackMarkType	ブラックマーク指定定数。

8.1 モデル : ISCBBUILDER interface メソッド

各モデルにおいて機能するメソッドは以下のとおりです。

Function	Method	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Document control	beginDocument	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	endDocument	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Initialization	appendInitialization	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Data (Text and Command)	appendByte	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendData	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytes	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Raw data (Text and Command)	appendRawByte	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendRawData	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendRawBytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Font style	appendFontStyle	✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
Code page	appendCodePage	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
International	appendInternational	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Line feed	appendLineFeed	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithLineFeed	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithLineFeed	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Unit feed	appendUnitFeed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithUnitFeed	✓	✓	*1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithUnitFeed	✓	✓	*1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Character space	appendCharacterSpace	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Line space	appendLineSpace	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- : 無視

*1 : データ（テキストやコマンド）の追加は行わず、1 ドットライン単位用の紙送りコマンドの追加のみ行う

Function	Method	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Emphasis	appendEmphasis	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithEmphasis	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithEmphasis	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Invert	appendInvert	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithInvert	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithInvert	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multiple	appendMultiple	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithMultiple	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithMultiple	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultipleHeight	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithMultipleHeight	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithMultipleHeight	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendMultipleWidth	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithMultipleWidth	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithMultipleWidth	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Under line	appendUnderLine	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithUnderLine	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithUnderLine	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Logo	appendLogo	✓	✓	-	✓	✓	✓	*2	*2	*2	*2	*2	✓	✓	✓	✓	✓	✓	✓	✓	✓
Absolute position	appendAbsolutePosition	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithAbsolutePosition	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithAbsolutePosition	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Alignment	appendAlignment	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendDataWithAlignment	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBytesWithAlignment	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- : 無視

*2 : 常にノーマルサイズ

Function	Method	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Cut paper	appendCutPaper	✓	✓	✓	✓	✓	✓	*3	*3	*3	*3	*3	✓	*3	*3	*3	*3	*3	*3	*3	✓
Peripheral	appendPeripheral	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
Sound	appendSound	✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	-	-	-	-	-	-	-	✓
Barcode	appendBarcodeData	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeBytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeDataWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeBytesWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeDataWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendBarcodeBytesWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
PDF417	appendPdf417Data	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417Bytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417DataWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417BytesWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417DataWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPdf417BytesWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-

- : 無視

*3 : 用紙カットを行わない

*4 : 常に左揃え

Function	Method	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
QR code	appendQrCodeData	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeBytes	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeDataWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeBytesWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeDataWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendQrCodeBytesWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	-
Bitmap	appendBitmap	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBitmapWithAbsolutePosition	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	appendBitmapWithAlignment	✓	✓	*4	✓	✓	✓	*4	*4	*4	*4	*4	✓	✓	✓	✓	✓	✓	✓	✓	✓
Black mark	appendBlackMark	-	✓	-	-	✓	✓	*5	*5	*5	*5	*5	-	*5	*5	*5	*5	*5	*5	*5	✓
Page mode	beginPageMode	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	endPageMode	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPageModeVerticalAbsolutePosition	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-
	appendPageModeRotation	✓	✓	-	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-

- : 無視

*4 : 常に左揃え

*5 : プリンタ側で BM 有効設定にする必要有り。
設定方法はユーザーズマニュアル参照のこと。

8.2 beginDocument メソッド

ドキュメント開始コマンドを生成し、commands プロパティに追加します。

宣言

- (void)beginDocument;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.3 endDocument メソッド

ドキュメント終了コマンドを生成し、commands プロパティに追加します。

宣言

- (void)endDocument;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.4 appendInitialization メソッド

初期化コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendInitialization:(SCBInitializationType)type;

引数

名称	説明	型
type	初期化タイプ <ul style="list-style-type: none"> SCBInitializationTypeCommand ... コマンド初期化 SCBInitializationTypeReset ... プリンタリセット SCBInitializationTypeResetWithPrint ... プリンタリセット (自己印字実行) 	SCBInitializationType

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendUnderLine:YES];
    [builder appendMultiple:2 height:2];
    [builder appendData:otherData];

    [builder appendInitialization:SCBInitializationTypeCommand];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.5 appendData メソッド

データ（テキストやコマンド）を commands プロパティに追加します。

宣言

- (void)appendByte:(unsigned char)data;
- (void)appendData:(NSData *)otherData;
- (void)appendBytes:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
data	データ（テキストやコマンド）	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    unsigned char bytes[] = {0x48, 0x65, 0x6c, 0x6c, 0x6f, 0x20, 0x57, 0x6f, 0x72, 0x6c, 0x64, 0x2e};

    NSUInteger length = sizeof(bytes);

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendByte:0x0a];

    [builder appendBytes:bytes length:length];
    [builder appendByte:0x0a];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.6 appendRawData メソッド

生のデータ（テキストやコマンド）を commands プロパティに追加します。

宣言

- (void)appendRawByte:(unsigned char)data;
- (void)appendRawData:(NSData *)otherData;
- (void)appendRawBytes:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
data	生のデータ（テキストやコマンド）	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation image:(UIImage *)image {
    NSString *urlString = [SMCSAllReceipts uploadBitmap:image completion:nil];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    NSData *data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:YES qrCode:YES];
    [builder appendRawData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

AllReceiptsFunctions.m /各コマンド仕様書を参照ください。

8.7 appendFontStyle メソッド

フォントスタイル指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendFontStyle:(SCBFontStyleType)type;

引数

名称	説明	型
type	フォントスタイル <ul style="list-style-type: none"> SCBFontStyleTypeA ... フォント A (12 x 24 ドット) / 7 x 9 フォント (ハーフドット) SCBFontStyleTypeB ... フォント B (9 x 24 ドット) / 5 x 9 フォント (2P-1) 	SCBFontStyleType

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendFontStyle:SCBFontStyleTypeA];
    [builder appendData:otherData];
    [builder appendFontStyle:SCBFontStyleTypeB];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.8 appendCodePage メソッド

コードページ指定コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendCodePage:(SCBCodePageType)type;
```

引数

名称	説明	型
type	<p>コードページ</p> <ul style="list-style-type: none"> • SCBCodePageTypeCP437 ... CodePage437 (USA, Std. Europe). • SCBCodePageTypeCP737 ... Codepage 737 (Greek). • SCBCodePageTypeCP772 ... Codepage 772 (Lithuanian). • SCBCodePageTypeCP774 ... Codepage 774 (Lithuanian). • SCBCodePageTypeCP851 ... Codepage 851 (Greek). • SCBCodePageTypeCP852 ... Codepage 852 (Latin-2). • SCBCodePageTypeCP855 ... Codepage 855 (Cyrillic Bulgarian). • SCBCodePageTypeCP857 ... Codepage 857 (Turkey). • SCBCodePageTypeCP858 ... Codepage 858 (Multilingual). • SCBCodePageTypeCP860 ... Codepage 860 (Portuguese). • SCBCodePageTypeCP861 ... Codepage 861 (Icelandic). • SCBCodePageTypeCP862 ... Codepage 862 (Israel (Hebrew)). • SCBCodePageTypeCP863 ... Codepage 863 (Canadian French). • SCBCodePageTypeCP864 ... Codepage 864 (Arabic). • SCBCodePageTypeCP865 ... Codepage 865 (Nordic). • SCBCodePageTypeCP866 ... Codepage 866 (Cyrillic Russian). • SCBCodePageTypeCP869 ... Codepage 869 (Greek). • SCBCodePageTypeCP874 ... Codepage 874 (Thai). • SCBCodePageTypeCP928 ... Codepage 928 (Greek). 	SCBCodePageType

名称	説明	型
type	<ul style="list-style-type: none"> • SCBCodePageTypeCP932 ... Katakana. • SCBCodePageTypeCP998 ... Normal. • SCBCodePageTypeCP999 ... Codepage 1252 (Windows Latin-1). • SCBCodePageTypeCP1001 ... Codepage 1001 (Arabic). • SCBCodePageTypeCP1250 ... Codepage 1250 (Windows Latin-2). • SCBCodePageTypeCP1251 ... Codepage 1251 (Windows Cyrillic). • SCBCodePageTypeCP1252 ... Codepage 1252 (Windows Latin-1). • SCBCodePageTypeCP2001 ... Codepage 2001 (Lithuanian-KBL). • SCBCodePageTypeCP3001 ... Codepage 3001 (Estonian-1). • SCBCodePageTypeCP3002 ... Codepage 3002 (Estonian-2). • SCBCodePageTypeCP3011 ... Codepage 3011 (Latvian-1). • SCBCodePageTypeCP3012 ... Codepage 3012 (Latvian-2). • SCBCodePageTypeCP3021 ... Codepage 3021 (Bulgarian). • SCBCodePageTypeCP3041 ... Codepage 3041 (Maltese). • SCBCodePageTypeCP3840 ... Codepage 3840 (IBM-Russian). • SCBCodePageTypeCP3841 ... Codepage 3841 (Gost). • SCBCodePageTypeCP3843 ... Codepage 3843 (Polish). • SCBCodePageTypeCP3844 ... Codepage 3844 (CS2). • SCBCodePageTypeCP3845 ... Codepage 3845 (Hungarian). • SCBCodePageTypeCP3846 ... Codepage 3846 (Turkish). • SCBCodePageTypeCP3847 ... Codepage 3847 (Brazil-ABNT). • SCBCodePageTypeCP3848 ... Codepage 3848 (Brazil-ABICOMP). • SCBCodePageTypeUTF8 ... UTF-8. • SCBCodePageTypeBlank ... User Setting Blank Code Page. 	SCBCodePageType

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    unsigned char bytes8[] = {0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e,
0x8f, 0x0a};
    unsigned char bytes9[] = {0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e,
0x9f, 0x0a};
    unsigned char bytesA[] = {0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae,
0xaf, 0x0a};
    unsigned char bytesB[] = {0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe,
0xbf, 0x0a};
    unsigned char bytesC[] = {0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce,
0xcf, 0x0a};
    unsigned char bytesD[] = {0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde,
0xdf, 0x0a};
    unsigned char bytesE[] = {0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee,
0xef, 0x0a};
    unsigned char bytesF[] = {0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff,
0x0a};

    NSUInteger length = sizeof(bytes8);

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendCodePage:SCBCodePageTypeCP932];

    [builder appendBytes:bytes8 length:length];
    [builder appendBytes:bytes9 length:length];
    [builder appendBytes:bytesA length:length];
    [builder appendBytes:bytesB length:length];
    [builder appendBytes:bytesC length:length];
    [builder appendBytes:bytesD length:length];
    [builder appendBytes:bytesE length:length];
    [builder appendBytes:bytesF length:length];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.9 appendInternational メソッド

国際文字指定コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendInternational:(SCBInternationalType)type;
```

引数

名称	説明	型
type	国際文字 <ul style="list-style-type: none"> • SCBInternationalTypeUSA ... USA • SCBInternationalTypeFrance ... フランス • SCBInternationalTypeGermany ... ドイツ • SCBInternationalTypeUK ... イギリス • SCBInternationalTypeDenmark ... デンマーク • SCBInternationalTypeSweden ... スウェーデン • SCBInternationalTypeItaly ... イタリア • SCBInternationalTypeSpain ... スペイン • SCBInternationalTypeJapan ... 日本 • SCBInternationalTypeNorway ... ノルウェー • SCBInternationalTypeDenmark2 ... デンマーク II • SCBInternationalTypeSpain2 ... スペイン II • SCBInternationalTypeLatinAmerica ... ラテンアメリカ • SCBInternationalTypeKorea ... 韓国 • SCBInternationalTypeIreland ... アイルランド • SCBInternationalTypeLegal ... Legal 	SCBInternationalType

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    unsigned char bytes[] = {0x23, 0x24, 0x40, 0x58, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x60, 0x7b, 0x7c, 0x7d, 0x7e, 0x0a};

    NSUInteger length = sizeof(bytes);

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendInternational:SCBInternationalTypeUSA];
    [builder appendBytes:bytes length:length];

    [builder appendInternational:SCBInternationalTypeJapan];
    [builder appendBytes:bytes length:length];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.10 appendLineFeed メソッド

行単位用の紙送りコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendLineFeed;
- (void)appendDataWithLineFeed:(NSData *)otherData;
- (void)appendBytesWithLineFeed:(const void *)bytes length:(NSUInteger)length;
- (void)appendLineFeed:(NSInteger)line;
- (void)appendDataWithLineFeed:(NSData *)otherData line:(NSInteger)line;
- (void)appendBytesWithLineFeed:(const void *)bytes length:(NSUInteger)length line:(NSInteger)line;

引数

名称	説明	型
line	紙送り量（行数単位）	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendLineFeed];

    [builder appendDataWithLineFeed:otherData];

    [builder appendData:otherData];
    [builder appendLineFeed:2];

    [builder appendDataWithLineFeed:otherData line:2];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.11 appendUnitFeed メソッド

1 ドットライン単位用の紙送りコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendUnitFeed:(NSInteger)unit;
- (void)appendDataWithUnitFeed:(NSData *)otherData unit:(NSInteger)unit;
- (void)appendBytesWithUnitFeed:(const void *)bytes length:(NSUInteger)length unit:(NSInteger)unit;

引数

名称	説明	型
unit	紙送り量（ドット数単位）	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendUnitFeed:64];

    [builder appendDataWithUnitFeed:otherData unit:64];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.12 appendCharacterSpace メソッド

文字間スペース設定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendCharacterSpace:(NSInteger)space;

引数

名称	説明	型
space	文字間スペース（ドット数単位）	NSInteger

戻り値

説明	型
-	-

1. 注意事項

日本語、中国語簡体字、繁体字、韓国語(DBCS)では文字間スペースが設定値の 2 倍になります。

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendCharacterSpace:0];
    [builder appendData:otherData];
    [builder appendCharacterSpace:4];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.13 appendLineSpace メソッド

行間スペース設定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendLineSpace:(NSInteger)lineSpace;

引数

名称	説明	型
lineSpace	行間スペース（ドット数単位）	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendLineSpace:32];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendLineSpace:24];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.14 appendEmphasis メソッド

強調指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendEmphasis:(BOOL)emphasis;
- (void)appendDataWithEmphasis:(NSData *)otherData;
- (void)appendBytesWithEmphasis:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
emphasis	強調 <ul style="list-style-type: none"> YES ... 強調設定 NO ... 強調解除 	BOOL
otherData	データ (テキストやコマンド)	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendEmphasis:YES];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendEmphasis:NO];
    [builder appendData:otherData];

    [builder appendDataWithEmphasis:otherData];
    [builder appendData:otherData];

    [builder appendDataWithEmphasis:otherDataHalf0];
    [builder appendData:otherDataHalf1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.15 appendInvert メソッド

白黒反転指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendInvert:(BOOL)invert;
- (void)appendDataWithInvert:(NSData *)otherData;
- (void)appendBytesWithInvert:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
invert	白黒反転 <ul style="list-style-type: none"> YES ... 白黒反転設定 NO ... 白黒反転解除 	BOOL
otherData	データ (テキストやコマンド)	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendInvert:YES];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendInvert:NO];
    [builder appendData:otherData];

    [builder appendDataWithInvert:otherData];
    [builder appendData:otherData];

    [builder appendDataWithInvert:otherDataHalf0];
    [builder appendData:otherDataHalf1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.16 appendMultiple メソッド

拡大指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendMultiple:(NSInteger)width height:(NSInteger)height;
- (void)appendDataWithMultiple:(NSData *)otherData width:(NSInteger)width height:(NSInteger)height;
- (void)appendBytesWithMultiple:(const void *)bytes length:(NSUInteger)length width:(NSInteger)width height:(NSInteger)height;
- (void)appendMultipleHeight:(NSInteger)height;
- (void)appendDataWithMultipleHeight:(NSData *)otherData height:(NSInteger)height;
- (void)appendBytesWithMultipleHeight:(const void *)bytes length:(NSUInteger)length height:(NSInteger)height;
- (void)appendMultipleWidth:(NSInteger)width;
- (void)appendDataWithMultipleWidth:(NSData *)otherData width:(NSInteger)width;
- (void)appendBytesWithMultipleWidth:(const void *)bytes length:(NSUInteger)length width:(NSInteger)width;

引数

名称	説明	型
width	横倍率	NSInteger
height	縦倍率	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendMultiple:2 height:2];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendMultiple:1 height:1];
    [builder appendData:otherData];

    [builder appendDataWithMultiple:otherData width:2 height:2];
    [builder appendData:        otherData];

    [builder appendDataWithMultiple:otherDataHalf0 width:2 height:2];
    [builder appendData:        otherDataHalf1];

    [builder appendData:otherDataHalf0];
    [builder appendDataWithMultiple:otherDataHalf1 width:2 height:2];

    [builder appendMultipleHeight:2];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendMultipleHeight:1];
    [builder appendData:otherData];

    [builder appendDataWithMultipleHeight:otherDataHalf0 height:2];
    [builder appendData:otherDataHalf1];

    [builder appendData:otherDataHalf0];
    [builder appendDataWithMultipleHeight:otherDataHalf1 height:2];

    [builder appendMultipleWidth:2];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendMultipleWidth:1];
    [builder appendData:otherData];

    [builder appendDataWithMultipleWidth:otherDataHalf0 width:2];
    [builder appendData:otherDataHalf1];

    [builder appendData:otherDataHalf0];
    [builder appendDataWithMultipleWidth:otherDataHalf1 width:2];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.17 appendUnderLine メソッド

アンダーライン指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendUnderLine:(BOOL)underLine;
- (void)appendDataWithUnderLine:(NSData *)otherData;
- (void)appendBytesWithUnderLine:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
underLine	アンダーライン <ul style="list-style-type: none"> YES ... アンダーライン設定 NO ... アンダーライン解除 	BOOL
otherData	データ (テキストやコマンド)	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf0 = [@"Hello " dataUsingEncoding:NSUTF8StringEncoding];
    NSData *otherDataHalf1 = [@"World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendUnderLine:YES];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendUnderLine:NO];
    [builder appendData:otherData];

    [builder appendDataWithUnderLine:otherData];
    [builder appendData:otherData];

    [builder appendDataWithUnderLine:otherDataHalf0];
    [builder appendData:otherDataHalf1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.18 appendLogo メソッド

ロゴ印刷コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendLogo:(SCBLogoSize)size number:(NSInteger)number;

引数

名称	説明	型
size	ロゴサイズ <ul style="list-style-type: none"> SCBLogoSizeNormal ... ノーマルモード SCBLogoSizeDoubleWidth ... 横 2 倍モード SCBLogoSizeDoubleHeight ... 縦 2 倍モード SCBLogoSizeDoubleWidthDoubleHeight ... 縦 2 倍、横 2 倍モード 	SCBLogoSize
number	ロゴ番号	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendLogo:SCBLogoSizeNormal number:1];
    [builder appendLogo:SCBLogoSizeDoubleWidth number:1];
    [builder appendLogo:SCBLogoSizeDoubleHeight number:1];
    [builder appendLogo:SCBLogoSizeDoubleWidthDoubleHeight number:1];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.19 appendAbsolutePosition メソッド

絶対位置指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendAbsolutePosition:(NSInteger)position;
- (void)appendDataWithAbsolutePosition:(NSData *)otherData position:(NSInteger)position;
- (void)appendBytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length position:(NSInteger)position;

引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	データ（テキストやコマンド）	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendAbsolutePosition:40];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendDataWithAbsolutePosition:otherData position:40];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.20 appendAlignment メソッド

位置揃え指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendAlignment:(SCBAlignmentPosition)position;
- (void)appendDataWithAlignment:(NSData *)otherData position:(SCBAlignmentPosition)position;
- (void)appendBytesWithAlignment:(const void *)bytes length:(NSUInteger)length position:(SCBAlignmentPosition)position;

引数

名称	説明	型
position	位置揃え <ul style="list-style-type: none"> SCBAlignmentPositionLeft ... 左揃え SCBAlignmentPositionCenter ... 中心揃え SCBAlignmentPositionRight ... 右揃え 	SCBAlignmentPosition
otherData	データ (テキストやコマンド)	NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendAlignment:SCBAlignmentPositionRight];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendAlignment:SCBAlignmentPositionLeft];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendDataWithAlignment:otherData position:SCBAlignmentPositionRight];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.21 appendCutPaper メソッド

用紙カットコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendCutPaper:(SCBCutPaperAction)action;

引数

名称	説明	型
action	用紙カット <ul style="list-style-type: none"> SCBCutPaperActionFullCut ... フルカット SCBCutPaperActionPartialCut ... パーシャルカット SCBCutPaperActionFullCutWithFeed ... 用紙送り付きフルカット SCBCutPaperActionPartialCutWithFeed ... 用紙送り付きパーシャルカット 	SCBCutPaperAction

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];
    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.22 appendPeripheral メソッド

ペリフェラル駆動コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendPeripheral:(SCBPeripheralChannel)channel;
- (void)appendPeripheral:(SCBPeripheralChannel)channel time:(NSInteger)time;

引数

名称	説明	型
channel	ペリフェラルチャネル <ul style="list-style-type: none"> • SCBPeripheralChannelNo1 ... チャネル 1 • SCBPeripheralChannelNo2 ... チャネル 2 	SCBPeripheralChannel
time	駆動時間 (1 ミリ秒単位) ※チャネル 1 のみ有効	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPeripheral:SCBPeripheralChannelNo1];
    [builder appendPeripheral:SCBPeripheralChannelNo2];
    [builder appendPeripheral:SCBPeripheralChannelNo1 time:2000];
    [builder appendPeripheral:SCBPeripheralChannelNo2 time:2000];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.23 appendSound メソッド

サウンド駆動コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendSound:(SCBSoundChannel)channel;
- (void)appendSound:(SCBSoundChannel)channel repeat:(NSInteger)repeat;

引数

名称	説明	型
channel	サウンドチャンネル <ul style="list-style-type: none"> SCBSoundChannelNo1 ... チャンネル 1 SCBSoundChannelNo2 ... チャンネル 2 	SCBSoundChannel
repeat	繰り返し回数	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendSound:SCBSoundChannelNo1];
    [builder appendSound:SCBSoundChannelNo2];
    [builder appendSound:SCBSoundChannelNo1 repeat:3];
    [builder appendSound:SCBSoundChannelNo2 repeat:3];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.24 appendBarcodeData メソッド

バーコード印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBarcodeData:(NSData *)otherData symbology:(SCBBarcodeSymbology)symbology
width:(SCBBarcodeWidth)width height:(NSInteger)height hri:(BOOL)hri;
```

```
- (void)appendBarcodeBytes:(const void *)bytes length:(NSUInteger)length
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width
height:(NSInteger)height hri:(BOOL)hri;
```

引数

名称	説明	型
otherData	バーコードデータ	NSData
bytes		const void *
length		NSUInteger
symbology	バーコードシンボル <ul style="list-style-type: none"> SCBBarcodeSymbologyUPCE ... UPC-E. SCBBarcodeSymbologyUPCA ... UPC-A. SCBBarcodeSymbologyJAN8 ... JAN/EAN8. SCBBarcodeSymbologyJAN13 ... JAN/EAN13. SCBBarcodeSymbologyCode39 ... Code39. SCBBarcodeSymbologyITF ... ITF. SCBBarcodeSymbologyCode128 ... Code128. SCBBarcodeSymbologyCode93 ... Code93. SCBBarcodeSymbologyCodabar ... Codabar. SCBBarcodeSymbologyNW7 ... NW7. 	SCBBarcodeSymbology
width	バーコード幅 <ul style="list-style-type: none"> SCBBarcodeWidthMode1 ... モード 1 SCBBarcodeWidthMode2 ... モード 2 SCBBarcodeWidthMode3 ... モード 3 SCBBarcodeWidthMode4 ... モード 4 SCBBarcodeWidthMode5 ... モード 5 SCBBarcodeWidthMode6 ... モード 6 SCBBarcodeWidthMode7 ... モード 7 SCBBarcodeWidthMode8 ... モード 8 SCBBarcodeWidthMode9 ... モード 9 	SCBBarcodeWidth
height	バーコードの高さ（ドット数単位）	NSInteger
hri	バー下文字列 <ul style="list-style-type: none"> YES ... 印刷あり NO ... 印刷なし 	BOOL

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherDataCode128 = [@"{B0123456789" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBarcodeData:otherDataCode128 symbology:SCBBarcodeSymbologyCode128
    width:SCBBarcodeWidthMode1 height:40 hri:YES];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.25 appendBarcodeDataWithAbsolutePosition メソッド

絶対位置指定バーコード印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBarcodeDataWithAbsolutePosition:(NSData *)otherData
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width
height:(NSInteger)height hri:(BOOL)hri position:(NSInteger)position;

- (void)appendBarcodeBytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width
height:(NSInteger)height hri:(BOOL)hri position:(NSInteger)position;
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	バーコードデータ	NSData
bytes		const void *
length		NSUInteger
symbology	バーコードシンボル <ul style="list-style-type: none"> SCBBarcodeSymbologyUPCE ... UPC-E. SCBBarcodeSymbologyUPCA ... UPC-A. SCBBarcodeSymbologyJAN8 ... JAN/EAN8. SCBBarcodeSymbologyJAN13 ... JAN/EAN13. SCBBarcodeSymbologyCode39 ... Code39. SCBBarcodeSymbologyITF ... ITF. SCBBarcodeSymbologyCode128 ... Code128. SCBBarcodeSymbologyCode93 ... Code93. SCBBarcodeSymbologyCodabar ... Codabar. SCBBarcodeSymbologyNW7 ... NW7. 	SCBBarcodeSymbology
width	バーコード幅 <ul style="list-style-type: none"> SCBBarcodeWidthMode1 ... モード 1 SCBBarcodeWidthMode2 ... モード 2 SCBBarcodeWidthMode3 ... モード 3 SCBBarcodeWidthMode4 ... モード 4 SCBBarcodeWidthMode5 ... モード 5 SCBBarcodeWidthMode6 ... モード 6 SCBBarcodeWidthMode7 ... モード 7 SCBBarcodeWidthMode8 ... モード 8 SCBBarcodeWidthMode9 ... モード 9 	SCBBarcodeWidth
height	バーコードの高さ（ドット数単位）	NSInteger
hri	バー下文字列 <ul style="list-style-type: none"> YES ... 印刷あり NO ... 印刷なし 	BOOL

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherDataCode128 = [@"{B0123456789" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBarcodeDataWithAbsolutePosition:otherDataCode128
     symbology:SCBBarcodeSymbologyCode128 width:SCBBarcodeWidthMode1 height:40 hri:YES position:40];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.26 appendBarcodeDataWithAlignment メソッド

位置揃えバーコード印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBarcodeDataWithAlignment:(NSData *)otherData
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width
height:(NSInteger)height hri:(BOOL)hri position:(SCBAlignmentPosition)position;

- (void)appendBarcodeBytesWithAlignment:(const void *)bytes length:(NSUInteger)length
symbology:(SCBBarcodeSymbology)symbology width:(SCBBarcodeWidth)width
height:(NSInteger)height hri:(BOOL)hri position:(SCBAlignmentPosition)position;
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> SCBAlignmentPositionLeft ... 左揃え SCBAlignmentPositionCenter ... 中心揃え SCBAlignmentPositionRight ... 右揃え 	SCBAlignmentPosition
otherData		NSData
bytes	バーコードデータ	const void *
length		NSUInteger
symbology	バーコードシンボル <ul style="list-style-type: none"> SCBBarcodeSymbologyUPCE ... UPC-E. SCBBarcodeSymbologyUPCA ... UPC-A. SCBBarcodeSymbologyJAN8 ... JAN/EAN8. SCBBarcodeSymbologyJAN13 ... JAN/EAN13. SCBBarcodeSymbologyCode39 ... Code39. SCBBarcodeSymbologyITF ... ITF. SCBBarcodeSymbologyCode128 ... Code128. SCBBarcodeSymbologyCode93 ... Code93. SCBBarcodeSymbologyCodabar ... Codabar. SCBBarcodeSymbologyNW7 ... NW7. 	SCBBarcodeSymbology
width	バーコード幅 <ul style="list-style-type: none"> SCBBarcodeWidthMode1 ... モード 1 SCBBarcodeWidthMode2 ... モード 2 SCBBarcodeWidthMode3 ... モード 3 SCBBarcodeWidthMode4 ... モード 4 SCBBarcodeWidthMode5 ... モード 5 SCBBarcodeWidthMode6 ... モード 6 SCBBarcodeWidthMode7 ... モード 7 SCBBarcodeWidthMode8 ... モード 8 SCBBarcodeWidthMode9 ... モード 9 	SCBBarcodeWidth
height	バーコードの高さ (ドット数単位)	NSInteger

名称	説明	型
hri	バー下文字列 <ul style="list-style-type: none"> YES ... 印刷あり NO ... 印刷なし 	BOOL

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherDataCode128 = [@"{B0123456789" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBarcodeDataWithAlignment:otherDataCode128 symbology:SCBBarcodeSymbologyCode128
     width:SCBBarcodeWidthMode1 height:40 hri:YES position:SCBAlignmentPositionCenter];
    [builder appendUnitFeed:32];

    [builder appendBarcodeDataWithAlignment:otherDataCode128 symbology:SCBBarcodeSymbologyCode128
     width:SCBBarcodeWidthMode1 height:40 hri:YES position:SCBAlignmentPositionRight];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.27 appendPdf417Data メソッド

PDF417 印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendPdf417Data:(NSData *)otherData line:(NSInteger)line column:(NSInteger)column
level:(SCBPdf417Level)level module:(NSInteger)module aspect:(NSInteger)aspect;
```

```
- (void)appendPdf417Bytes:(const void *)bytes length:(NSUInteger)length line:(NSInteger)line
column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module
aspect:(NSInteger)aspect;
```

引数

名称	説明	型
otherData	PDF417 データ	NSData
bytes		const void *
length		NSUInteger
line	ライン数	NSInteger
column	カラム数	NSInteger
level	PDF417 ECC (セキュリティレベル) <ul style="list-style-type: none"> SCBPdf417LevelECC0 ... セキュリティレベル 0 SCBPdf417LevelECC1 ... セキュリティレベル 1 SCBPdf417LevelECC2 ... セキュリティレベル 2 SCBPdf417LevelECC3 ... セキュリティレベル 3 SCBPdf417LevelECC4 ... セキュリティレベル 4 SCBPdf417LevelECC5 ... セキュリティレベル 5 SCBPdf417LevelECC6 ... セキュリティレベル 6 SCBPdf417LevelECC7 ... セキュリティレベル 7 SCBPdf417LevelECC8 ... セキュリティレベル 8 	SCBPdf417Level
module	モジュール X 方向サイズ	NSInteger
aspect	モジュールアスペクト比	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPdf417Data:otherData line:0 column:1 level:SCBPdf417LevelECC0 module:2 aspect:2];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.28 appendPdf417DataWithAbsolutePosition メソッド

絶対位置指定 PDF417 印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendPdf417DataWithAbsolutePosition:(NSData *)otherData line:(NSInteger)line
column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module
aspect:(NSInteger)aspect position:(NSInteger)position;
```

```
- (void)appendPdf417BytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length
line:(NSInteger)line column:(NSInteger)column level:(SCBPdf417Level)level
module:(NSInteger)module aspect:(NSInteger)aspect position:(NSInteger)position;
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	PDF417 データ	NSData
bytes		const void *
length		NSUInteger
line	ライン数	NSInteger
column	カラム数	NSInteger
level	PDF417 ECC（セキュリティレベル） <ul style="list-style-type: none"> SCBPdf417LevelECC0 ... セキュリティレベル 0 SCBPdf417LevelECC1 ... セキュリティレベル 1 SCBPdf417LevelECC2 ... セキュリティレベル 2 SCBPdf417LevelECC3 ... セキュリティレベル 3 SCBPdf417LevelECC4 ... セキュリティレベル 4 SCBPdf417LevelECC5 ... セキュリティレベル 5 SCBPdf417LevelECC6 ... セキュリティレベル 6 SCBPdf417LevelECC7 ... セキュリティレベル 7 SCBPdf417LevelECC8 ... セキュリティレベル 8 	SCBPdf417Level
module	モジュール X 方向サイズ	NSInteger
aspect	モジュールアスペクト比	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPdf417DataWithAbsolutePosition:otherData line:0 column:1 level:SCBPdf417LevelECC0
    module:2 aspect:2 position:40];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.29 appendPdf417DataWithAlignment メソッド

位置揃え PDF417 印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendPdf417DataWithAlignment:(NSData *)otherData line:(NSInteger)line
column:(NSInteger)column level:(SCBPdf417Level)level module:(NSInteger)module
aspect:(NSInteger)aspect position:(SCBAlignmentPosition)position;

- (void)appendPdf417BytesWithAlignment:(const void *)bytes length:(NSUInteger)length
line:(NSInteger)line column:(NSInteger)column level:(SCBPdf417Level)level
module:(NSInteger)module aspect:(NSInteger)aspect position:(SCBAlignmentPosition)position;
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> SCBAlignmentPositionLeft ... 左揃え SCBAlignmentPositionCenter ... 中心揃え SCBAlignmentPositionRight ... 右揃え 	SCBAlignmentPosition
otherData	PDF417 データ	NSData
bytes		const void *
length		NSUInteger
line	ライン数	NSInteger
column	カラム数	NSInteger
level	PDF417 ECC (セキュリティレベル) <ul style="list-style-type: none"> SCBPdf417LevelECC0 ... セキュリティレベル 0 SCBPdf417LevelECC1 ... セキュリティレベル 1 SCBPdf417LevelECC2 ... セキュリティレベル 2 SCBPdf417LevelECC3 ... セキュリティレベル 3 SCBPdf417LevelECC4 ... セキュリティレベル 4 SCBPdf417LevelECC5 ... セキュリティレベル 5 SCBPdf417LevelECC6 ... セキュリティレベル 6 SCBPdf417LevelECC7 ... セキュリティレベル 7 SCBPdf417LevelECC8 ... セキュリティレベル 8 	SCBPdf417Level
module	モジュール X 方向サイズ	NSInteger
aspect	モジュールアスペクト比	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendPdf417DataWithAlignment:otherData line:0 column:1 level:SCBPdf417LevelECC0 module:2
    aspect:2 position:SCBAlignmentPositionCenter];
    [builder appendUnitFeed:32];

    [builder appendPdf417DataWithAlignment:otherData line:0 column:1 level:SCBPdf417LevelECC0 module:2
    aspect:2 position:SCBAlignmentPositionRight];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.30 appendQrCodeData メソッド

QR コード印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendQrCodeData:(NSData *)otherData model:(SCBQrCodeModel)model
level:(SCBQrCodeLevel)level cell:(NSInteger)cell;
```

```
- (void)appendQrCodeBytes:(const void *)bytes length:(NSUInteger)length
model:(SCBQrCodeModel)model level:(SCBQrCodeLevel)level cell:(NSInteger)cell;
```

引数

名称	説明	型
otherData	QR コードデータ	NSData
bytes		const void *
length		NSUInteger
model	QR コードモデル <ul style="list-style-type: none"> SCBQrCodeModelNo1 ... モデル 1 SCBQrCodeModelNo2 ... モデル 2 	SCBQrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> SCBQrCodeLevelL ... 誤り訂正レベル L SCBQrCodeLevelM ... 誤り訂正レベル M SCBQrCodeLevelQ ... 誤り訂正レベル Q SCBQrCodeLevelH ... 誤り訂正レベル H 	SCBQrCodeLevel
cell	QR コードセルサイズ	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendQrCodeData:otherData model:SCBQrCodeModelNo2 level:SCBQrCodeLevelL cell:4];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.31 appendQrCodeDataWithAbsolutePosition メソッド

絶対位置指定 QR コード印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendQrCodeDataWithAbsolutePosition:(NSData *)otherData
model:(SCBQrCodeModel)model level:(SCBQrCodeLevel)level cell:(NSInteger)cell
position:(NSInteger)position;

- (void)appendQrCodeBytesWithAbsolutePosition:(const void *)bytes length:(NSUInteger)length
model:(SCBQrCodeModel)model level:(SCBQrCodeLevel)level cell:(NSInteger)cell
position:(NSInteger)position;
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
otherData	QR コードデータ	NSData
bytes		const void *
length		NSUInteger
model	QR コードモデル <ul style="list-style-type: none"> SCBQrCodeModelNo1 ... モデル 1 SCBQrCodeModelNo2 ... モデル 2 	SCBQrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> SCBQrCodeLevelL ... 誤り訂正レベル L SCBQrCodeLevelM ... 誤り訂正レベル M SCBQrCodeLevelQ ... 誤り訂正レベル Q SCBQrCodeLevelH ... 誤り訂正レベル H 	SCBQrCodeLevel
cell	QR コードセルサイズ	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendQrCodeDataWithAbsolutePosition:otherData model:SCBQrCodeModelNo2
level:SCBQrCodeLevelL cell:4 position:40];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.32 appendQrCodeDataWithAlignment メソッド

位置揃え QR コード印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendQrCodeDataWithAlignment:(NSData *)otherData model:(SCBQrCodeModel)model
level:(SCBQrCodeLevel)level cell:(NSInteger)cell position:(SCBAlignmentPosition)position;
```

```
- (void)appendQrCodeBytesWithAlignment:(const void *)bytes length:(NSUInteger)length
model:(SCBQrCodeModel)model level:(SCBQrCodeLevel)level cell:(NSInteger)cell
position:(SCBAlignmentPosition)position;
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> SCBAlignmentPositionLeft ... 左揃え SCBAlignmentPositionCenter ... 中心揃え SCBAlignmentPositionRight ... 右揃え 	SCBAlignmentPosition
otherData	QR コードデータ	NSData
bytes		const void *
length		NSUInteger
model	QR コードモデル <ul style="list-style-type: none"> SCBQrCodeModelNo1 ... モデル 1 SCBQrCodeModelNo2 ... モデル 2 	SCBQrCodeModel
level	QR コード誤り訂正レベル <ul style="list-style-type: none"> SCBQrCodeLevelL ... 誤り訂正レベル L SCBQrCodeLevelM ... 誤り訂正レベル M SCBQrCodeLevelQ ... 誤り訂正レベル Q SCBQrCodeLevelH ... 誤り訂正レベル H 	SCBQrCodeLevel
cell	QR コードセルサイズ	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation {
    NSData *otherData = [@"Hello World." dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendQrCodeDataWithAlignment:otherData model:SCBQrCodeModelNo2 level:SCBQrCodeLevelL
    cell:4 position:SCBAlignmentPositionCenter];
    [builder appendUnitFeed:32];

    [builder appendQrCodeDataWithAlignment:otherData model:SCBQrCodeModelNo2 level:SCBQrCodeLevelL
    cell:4 position:SCBAlignmentPositionRight];
    [builder appendUnitFeed:32];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m /各コマンド仕様書を参照ください。

8.33 appendBitmap メソッド

ビットマップ印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion width:(NSInteger)width bothScale:(BOOL)bothScale rotation:(SCBBitmapConverterRotation)rotation;
```

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion rotation:(SCBBitmapConverterRotation)rotation;
```

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion width:(NSInteger)width bothScale:(BOOL)bothScale;
```

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion;
```

引数

名称	説明	型
image	元となるビットマップオブジェクト	UIImage
diffusion	誤差拡散 <ul style="list-style-type: none"> YES ... 誤差拡散を行う NO ... 誤差拡散を行わない 	BOOL
width	変換後のビットマップ幅（ドット数単位）	NSInteger
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> YES ... 高さ変換を行う NO ... 高さ変換を行わない 	BOOL
rotation	変換後の回転 <ul style="list-style-type: none"> SCBBitmapConverterRotationNormal ... 回転なし SCBBitmapConverterRotationRight90 ... 右 90 度回転 SCBBitmapConverterRotationLeft90 ... 左 90 度回転 SCBBitmapConverterRotationRotate180 ... 180 度回転 	SCBBitmapConverterRotation

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    UIImage *starLogoImage = [UIImage imageNamed:@"StarLogoImage"];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:@"\nNormal\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES];

    [builder appendData:@"\nwidth:Full, bothScale:YES\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES width:width bothScale:YES];
    [builder appendData:@"\nwidth:Full, bothScale:NO\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES width:width bothScale:NO];

    [builder appendData:@"\nRotate180\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES rotation:SCBBitmapConverterRotationRotate180];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.34 appendBitmapWithAbsolutePosition メソッド

絶対位置指定ビットマップ印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
width:(NSInteger)width bothScale:(BOOL)bothScale rotation:(SCBBitmapConverterRotation)rotation
position:(NSInteger)position;
```

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
rotation:(SCBBitmapConverterRotation)rotation position:(NSInteger)position;
```

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
width:(NSInteger)width bothScale:(BOOL)bothScale position:(NSInteger)position;
```

```
- (void)appendBitmapWithAbsolutePosition:(UIImage *)image diffusion:(BOOL)diffusion
position:(NSInteger)position;
```

引数

名称	説明	型
position	絶対位置（ドット数単位）	NSInteger
image	元となるビットマップオブジェクト	UIImage
diffusion	誤差拡散 <ul style="list-style-type: none"> YES ... 誤差拡散を行う NO ... 誤差拡散を行わない 	BOOL
width	変換後のビットマップ幅（ドット数単位）	NSInteger
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> YES ... 高さ変換を行う NO ... 高さ変換を行わない 	BOOL
rotation	変換後の回転 <ul style="list-style-type: none"> SCBBitmapConverterRotationNormal ... 回転なし SCBBitmapConverterRotationRight90 ... 右 90 度回転 SCBBitmapConverterRotationLeft90 ... 左 90 度回転 SCBBitmapConverterRotationRotate180 ... 180 度回転 	SCBBitmapConverterRotation

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    UIImage *starLogoImage = [UIImage imageNamed:@"StarLogoImage"];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:@"\nNormal, AbsolutePosition:40*\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAbsolutePosition:starLogoImage diffusion:YES position:40];

    [builder appendData:@"\nwidth:Full, bothScale:YES, AbsolutePosition:40*\n"
dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES width:width bothScale:YES position:40];
    [builder appendData:@"\nwidth:Full, bothScale:NO, AbsolutePosition:40*\n"
dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmap:starLogoImage diffusion:YES width:width bothScale:NO position:40];

    [builder appendData:@"\nRotate180, AbsolutePosition:40*\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAbsolutePosition:starLogoImage diffusion:YES
rotation:SCBBitmapConverterRotationRotate180 position:40];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.35 appendBitmapWithAlignment メソッド

位置揃えビットマップ印刷コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion
width:(NSInteger)width bothScale:(BOOL)bothScale rotation:(SCBBitmapConverterRotation)rotation
position:(SCBAlignmentPosition)position;
```

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion
rotation:(SCBBitmapConverterRotation)rotation position:(SCBAlignmentPosition)position;
```

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion
width:(NSInteger)width bothScale:(BOOL)bothScale position:(SCBAlignmentPosition)position;
```

```
- (void)appendBitmapWithAlignment:(UIImage *)image diffusion:(BOOL)diffusion
position:(SCBAlignmentPosition)position;
```

引数

名称	説明	型
position	位置揃え指定 <ul style="list-style-type: none"> SCBAlignmentPositionLeft ... 左揃え SCBAlignmentPositionCenter ... 中心揃え SCBAlignmentPositionRight ... 右揃え 	SCBAlignmentPosition
image	元となるビットマップオブジェクト	UIImage
diffusion	誤差拡散 <ul style="list-style-type: none"> YES ... 誤差拡散を行う NO ... 誤差拡散を行わない 	BOOL
width	変換後のビットマップ幅（ドット数単位）	NSInteger
bothScale	width プロパティによる変換率に合わせた高さ変換 <ul style="list-style-type: none"> YES ... 高さ変換を行う NO ... 高さ変換を行わない 	BOOL
rotation	変換後の回転 <ul style="list-style-type: none"> SCBBitmapConverterRotationNormal ... 回転なし SCBBitmapConverterRotationRight90 ... 右 90 度回転 SCBBitmapConverterRotationLeft90 ... 左 90 度回転 SCBBitmapConverterRotationRotate180 ... 180 度回転 	SCBBitmapConverterRotation

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    UIImage *starLogoImage = [UIImage imageNamed:@"StarLogoImage"];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:@"\nNormal, Alignment:Center\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES position:SCBAlignmentPositionCenter];

    [builder appendData:@"\nwidth:Full, bothScale:YES, Alignment:Center\n"
    dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES width:width bothScale:YES
    position:SCBAlignmentPositionCenter];

    [builder appendData:@"\nwidth:Full, bothScale:NO, Alignment:Center\n"
    dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES width:width bothScale:NO
    position:SCBAlignmentPositionCenter];

    [builder appendData:@"\nRotate180, Alignment:Center\n" dataUsingEncoding:NSUTF8StringEncoding];
    [builder appendBitmapWithAlignment:starLogoImage diffusion:YES
    rotation:SCBBitmapConverterRotationRotate180 position:SCBAlignmentPositionCenter];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.36 appendBlackMark メソッド

ブラックマークコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendBlackMark:(SCBBlackMarkType)type;

引数

名称	説明	型
Type	ブラックマーク <ul style="list-style-type: none"> SCBBlackMarkTypeInvalid ... ブラックマーク無効 SCBBlackMarkTypeValid ... ブラックマーク有効 SCBBlackMarkTypeValidWithDetection ... ブラックマーク有効 + ブラックマーク検出動作 	SCBBlackMarkType

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBlackMark:SCBBlackMarkTypeValid];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    // [builder appendBlackMark:SCBBlackMarkTypeInvalid];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.37 beginPageMode メソッド

ページモード開始コマンドを生成し、commands プロパティに追加します。

宣言

- (void)beginPageMode:(CGRect)rect rotation:(SCBBitmapConverterRotation)rotation;

引数

名称	説明	型
rect	位置とサイズ（ドット数単位）	CGRect
rotation	印刷方向 <ul style="list-style-type: none"> SCBBitmapConverterRotationNormal ... 回転なし SCBBitmapConverterRotationRight90 ... 右 90 度回転 SCBBitmapConverterRotationLeft90 ... 左 90 度回転 SCBBitmapConverterRotationRotate180 ... 180 度回転 	SCBBitmapConverterRotation

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Rotate180 Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.38 endPageMode メソッド

ページモード終了コマンドを生成し、commands プロパティに追加します。

宣言

- (void)endPageMode;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Rotate180 Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.39 appendPageModeVerticalAbsolutePosition メソッド

ページモードの縦方向絶対位置指定コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendPageModeVerticalAbsolutePosition:(NSInteger)position;
```

引数

名称	説明	型
position	縦方向の絶対位置（ドット数単位）	NSInteger

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Rotate180 Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.40 appendPageModeRotation メソッド

ページモード印刷方向コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendPageModeRotation:(SCBBitmapConverterRotation)rotation;
```

引数

名称	説明	型
rotation	印刷方向 <ul style="list-style-type: none"> • SCBBitmapConverterRotationNormal ... 回転なし • SCBBitmapConverterRotationRight90 ... 右 90 度回転 • SCBBitmapConverterRotationLeft90 ... 左 90 度回転 • SCBBitmapConverterRotationRotate180 ... 180 度回転 	SCBBitmapConverterRotation

戻り値

説明	型
-	-

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation width:(NSInteger)width {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    int height = 30 * 8;    // 30mm!!!

    CGRect rect;

    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:[@"\n*Mixed Text*\n" dataUsingEncoding:NSUTF8StringEncoding]];

    rect = CGRectMake(0, 0, width, height);

    [builder beginPageMode:rect rotation:SCBBitmapConverterRotationNormal];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder appendPageModeRotation:SCBBitmapConverterRotationRotate180];

    [builder appendPageModeVerticalAbsolutePosition:height / 2];

    [builder appendDataWithAbsolutePosition:otherData position:width / 2];

    [builder endPageMode];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.41 commands プロパティ

生成、追加されたコマンド列です。

宣言

@property (nonatomic, readonly) NSMutableData *commands;

値

説明	型
生成、追加されたコマンド列	NSMutableData

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation {
    NSData *otherData = [@"Hello World.\n" dataUsingEncoding:NSUTF8StringEncoding];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:otherData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

ApiFunctions.m を参照ください。

8.42 SCBInitializationType コンスタント

初期化指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBInitializationType) {
    SCBInitializationTypeCommand
// SCBInitializationTypeReset,
// SCBInitializationTypeResetWithPrint
};
```

定数

名称	説明
SCBInitializationTypeCommand	コマンド初期化
SCBInitializationTypeReset	プリンタリセット
SCBInitializationTypeResetWithPrint	プリンタリセット（自己印字実行）

ApiFunctions.m を参照ください。

8.43 SCBFontStyleType コンスタント

フォントスタイル指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBFontStyleType) {
    SCBFontStyleTypeA,
    SCBFontStyleTypeB
};
```

定数

名称	説明
SCBFontStyleTypeA	フォント A（12 x 24 ドット） / 7 x 9 フォント（ハーフドット）
SCBFontStyleTypeB	フォント B（9 x 24 ドット） / 5 x 9 フォント（2P-1）

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.44 SCBCodePageType コンスタント

コードページ指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBCodePageType) {  
    SCBCodePageTypeCP437,  
    SCBCodePageTypeCP737,  
    SCBCodePageTypeCP772,  
    SCBCodePageTypeCP774,  
    SCBCodePageTypeCP851,  
    SCBCodePageTypeCP852,  
    SCBCodePageTypeCP855,  
    SCBCodePageTypeCP857,  
    SCBCodePageTypeCP858,  
    SCBCodePageTypeCP860,  
    SCBCodePageTypeCP861,  
    SCBCodePageTypeCP862,  
    SCBCodePageTypeCP863,  
    SCBCodePageTypeCP864,  
    SCBCodePageTypeCP865,  
    SCBCodePageTypeCP866,  
    SCBCodePageTypeCP869,  
    SCBCodePageTypeCP874,  
    SCBCodePageTypeCP928,  
    SCBCodePageTypeCP932,  
    SCBCodePageTypeCP998,  
    SCBCodePageTypeCP999,  
    SCBCodePageTypeCP1001,  
    SCBCodePageTypeCP1250,  
    SCBCodePageTypeCP1251,  
    SCBCodePageTypeCP1252,  
    SCBCodePageTypeCP2001,  
    SCBCodePageTypeCP3001,  
    SCBCodePageTypeCP3002,  
    SCBCodePageTypeCP3011,  
    SCBCodePageTypeCP3012,  
    SCBCodePageTypeCP3021,  
    SCBCodePageTypeCP3041,  
    SCBCodePageTypeCP3840,  
    SCBCodePageTypeCP3841,  
    SCBCodePageTypeCP3843,  
    SCBCodePageTypeCP3844,  
    SCBCodePageTypeCP3845,  
    SCBCodePageTypeCP3846,  
    SCBCodePageTypeCP3847,  
    SCBCodePageTypeCP3848,  
    SCBCodePageTypeUTF8,  
    SCBCodePageTypeBlank  
};
```

定数

名称	説明
SCBCodePageTypeCP437	CodePage437 (USA, Std. Europe).
SCBCodePageTypeCP737	Codepage 737 (Greek).
SCBCodePageTypeCP772	Codepage 772 (Lithuanian).
SCBCodePageTypeCP774	Codepage 774 (Lithuanian).
SCBCodePageTypeCP851	Codepage 851 (Greek).
SCBCodePageTypeCP852	Codepage 852 (Latin-2).
SCBCodePageTypeCP855	Codepage 855 (Cyrillic Bulgarian).
SCBCodePageTypeCP857	Codepage 857 (Turkey).
SCBCodePageTypeCP858	Codepage 858 (Multilingual).
SCBCodePageTypeCP860	Codepage 860 (Portuguese).
SCBCodePageTypeCP861	Codepage 861 (Icelandic).
SCBCodePageTypeCP862	Codepage 862 (Israel (Hebrew)).
SCBCodePageTypeCP863	Codepage 863 (Canadian French).
SCBCodePageTypeCP864	Codepage 864 (Arabic).
SCBCodePageTypeCP865	Codepage 865 (Nordic).
SCBCodePageTypeCP866	Codepage 866 (Cyrillic Russian).
SCBCodePageTypeCP869	Codepage 869 (Greek).
SCBCodePageTypeCP874	Codepage 874 (Thai).
SCBCodePageTypeCP928	Codepage 928 (Greek).
SCBCodePageTypeCP932	Katakana.
SCBCodePageTypeCP998	Normal.
SCBCodePageTypeCP999	Codepage 1252 (Windows Latin-1).
SCBCodePageTypeCP1001	Codepage 1001 (Arabic).
SCBCodePageTypeCP1250	Codepage 1250 (Windows Latin-2).
SCBCodePageTypeCP1251	Codepage 1251 (Windows Cyrillic).
SCBCodePageTypeCP1252	Codepage 1252 (Windows Latin-1).
SCBCodePageTypeCP2001	Codepage 2001 (Lithuanian-KBL).
SCBCodePageTypeCP3001	Codepage 3001 (Estonian-1).
SCBCodePageTypeCP3002	Codepage 3002 (Estonian-2).
SCBCodePageTypeCP3011	Codepage 3011 (Latvian-1).
SCBCodePageTypeCP3012	Codepage 3012 (Latvian-2).
SCBCodePageTypeCP3021	Codepage 3021 (Bulgarian).
SCBCodePageTypeCP3041	Codepage 3041 (Maltese).
SCBCodePageTypeCP3840	Codepage 3840 (IBM-Russian).
SCBCodePageTypeCP3841	Codepage 3841 (Gost).
SCBCodePageTypeCP3843	Codepage 3843 (Polish).
SCBCodePageTypeCP3844	Codepage 3844 (CS2).
SCBCodePageTypeCP3845	Codepage 3845 (Hungarian).
SCBCodePageTypeCP3846	Codepage 3846 (Turkish).
SCBCodePageTypeCP3847	Codepage 3847 (Brazil-ABNT).
SCBCodePageTypeCP3848	Codepage 3848 (Brazil-ABICOMP).
SCBCodePageTypeUTF8	UTF-8.
SCBCodePageTypeBlank	User Setting Blank Code Page.

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.45 SCBInternationalType コンスタント

国際文字指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBInternationalType) {
    SCBInternationalTypeUSA,
    SCBInternationalTypeFrance,
    SCBInternationalTypeGermany,
    SCBInternationalTypeUK,
    SCBInternationalTypeDenmark,
    SCBInternationalTypeSweden,
    SCBInternationalTypeItaly,
    SCBInternationalTypeSpain,
    SCBInternationalTypeJapan,
    SCBInternationalTypeNorway,
    SCBInternationalTypeDenmark2,
    SCBInternationalTypeSpain2,
    SCBInternationalTypeLatinAmerica,
    SCBInternationalTypeKorea,
    SCBInternationalTypeIreland,
    SCBInternationalTypeLegal
};
```

定数

名称	説明
SCBInternationalTypeUSA	USA
SCBInternationalTypeFrance	フランス
SCBInternationalTypeGermany	ドイツ
SCBInternationalTypeUK	イギリス
SCBInternationalTypeDenmark	デンマーク
SCBInternationalTypeSweden	スウェーデン
SCBInternationalTypeItaly	イタリア
SCBInternationalTypeSpain	スペイン
SCBInternationalTypeJapan	日本
SCBInternationalTypeNorway	ノルウェー
SCBInternationalTypeDenmark2	デンマーク II
SCBInternationalTypeSpain2	スペイン II
SCBInternationalTypeLatinAmerica	ラテンアメリカ
SCBInternationalTypeKorea	韓国
SCBInternationalTypeIreland	アイルランド
SCBInternationalTypeLegal	Legal

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.46 SCBLogoSize コンスタント

ロゴサイズ指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBLogoSize) {
    SCBLogoSizeNormal,
    SCBLogoSizeDoubleWidth,
    SCBLogoSizeDoubleHeight,
    SCBLogoSizeDoubleWidthDoubleHeight
};
```

定数

名称	説明
SCBLogoSizeNormal	ノーマルモード
SCBLogoSizeDoubleWidth	横 2 倍モード
SCBLogoSizeDoubleHeight	縦 2 倍モード
SCBLogoSizeDoubleWidthDoubleHeight	縦 2 倍、横 2 倍モード

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.47 SCBAlignmentPosition コンスタント

位置揃え指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBAlignmentPosition) {
    SCBAlignmentPositionLeft,
    SCBAlignmentPositionCenter,
    SCBAlignmentPositionRight
};
```

定数

名称	説明
SCBAlignmentPositionLeft	左揃え
SCBAlignmentPositionCenter	中心揃え
SCBAlignmentPositionRight	右揃え

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.48 SCBCutPaperAction コンスタント

用紙カット指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SCBCutPaperAction) {
    SCBCutPaperActionFullCut,
    SCBCutPaperActionPartialCut,
    SCBCutPaperActionFullCutWithFeed,
    SCBCutPaperActionPartialCutWithFeed
};
```

定数

名称	説明
SCBCutPaperActionFullCut	フルカット
SCBCutPaperActionPartialCut	パーシャルカット
SCBCutPaperActionFullCutWithFeed	用紙送り付きフルカット
SCBCutPaperActionPartialCutWithFeed	用紙送り付きパーシャルカット

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.49 SCBPeripheralChannel コンスタント

ペリフェラルチャネル指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SCBPeripheralChannel) {
    SCBPeripheralChannelNo1,
    SCBPeripheralChannelNo2
};
```

定数

名称	説明
SCBPeripheralChannelNo1	チャンネル 1
SCBPeripheralChannelNo2	チャンネル 2

ApiFunctions.m を参照ください。

8.50 SCBSoundChannel コンスタント

サウンドチャンネル指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SCBSoundChannel) {
    SCBSoundChannelNo1,
    SCBSoundChannelNo2
};
```

定数

名称	説明
SCBSoundChannelNo1	チャンネル 1
SCBSoundChannelNo2	チャンネル 2

ApiFunctions.m を参照ください。

8.51 SCBBarcodeSymbology コンスタント

バーコードシンボル指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBBarcodeSymbology) {
    SCBBarcodeSymbologyUPCE,
    SCBBarcodeSymbologyUPCA,
    SCBBarcodeSymbologyJAN8,
    SCBBarcodeSymbologyJAN13,
    SCBBarcodeSymbologyCode39,
    SCBBarcodeSymbologyITF,
    SCBBarcodeSymbologyCode128,
    SCBBarcodeSymbologyCode93,
    // SCBBarcodeSymbologyCodabar,
    SCBBarcodeSymbologyNW7
};
```

定数

名称	説明
SCBBarcodeSymbologyUPCE	UPC-E.
SCBBarcodeSymbologyUPCA	UPC-A.
SCBBarcodeSymbologyJAN8	JAN/EAN8.
SCBBarcodeSymbologyJAN13	JAN/EAN13.
SCBBarcodeSymbologyCode39	Code39.
SCBBarcodeSymbologyITF	ITF.
SCBBarcodeSymbologyCode128	Code128.
SCBBarcodeSymbologyCode93	Code93.
SCBBarcodeSymbologyCodabar	Codabar.
SCBBarcodeSymbologyNW7	NW7.

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.52 SCBBarcodeWidth コンスタント

バーコード幅指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBBarcodeWidth) {
    SCBBarcodeWidthMode1,
    SCBBarcodeWidthMode2,
    SCBBarcodeWidthMode3,
    SCBBarcodeWidthMode4,
    SCBBarcodeWidthMode5,
    SCBBarcodeWidthMode6,
    SCBBarcodeWidthMode7,
    SCBBarcodeWidthMode8,
    SCBBarcodeWidthMode9
};
```

定数

名称	説明
SCBBarcodeWidthMode1	モード 1
SCBBarcodeWidthMode2	モード 2
SCBBarcodeWidthMode3	モード 3
SCBBarcodeWidthMode4	モード 4
SCBBarcodeWidthMode5	モード 5
SCBBarcodeWidthMode6	モード 6
SCBBarcodeWidthMode7	モード 7
SCBBarcodeWidthMode8	モード 8
SCBBarcodeWidthMode9	モード 9

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.53 SCBPdf417Level コンスタント

PDF417 ECC（セキュリティレベル）指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBPdf417Level) {
    SCBPdf417LevelECC0,
    SCBPdf417LevelECC1,
    SCBPdf417LevelECC2,
    SCBPdf417LevelECC3,
    SCBPdf417LevelECC4,
    SCBPdf417LevelECC5,
    SCBPdf417LevelECC6,
    SCBPdf417LevelECC7,
    SCBPdf417LevelECC8
};
```

定数

名称	説明
SCBPdf417LevelECC0	セキュリティレベル 0
SCBPdf417LevelECC1	セキュリティレベル 1
SCBPdf417LevelECC2	セキュリティレベル 2
SCBPdf417LevelECC3	セキュリティレベル 3
SCBPdf417LevelECC4	セキュリティレベル 4
SCBPdf417LevelECC5	セキュリティレベル 5
SCBPdf417LevelECC6	セキュリティレベル 6
SCBPdf417LevelECC7	セキュリティレベル 7
SCBPdf417LevelECC8	セキュリティレベル 8

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.54 SCBQRCodeModel コンスタント

QR コードモデル指定定数

宣言

```
typedef NS_ENUM(NSUInteger, SCBQRCodeModel) {
    SCBQRCodeModelNo1,
    SCBQRCodeModelNo2
};
```

定数

名称	説明
SCBQRCodeModelNo1	モデル 1
SCBQRCodeModelNo2	モデル 2

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.55 SCBQRCodeLevel コンスタント

QR コード誤り訂正レベル指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBQRCodeLevel) {
    SCBQRCodeLevelL,
    SCBQRCodeLevelM,
    SCBQRCodeLevelQ,
    SCBQRCodeLevelH
};
```

定数

名称	説明
SCBQRCodeLevelL	誤り訂正レベル L
SCBQRCodeLevelM	誤り訂正レベル M
SCBQRCodeLevelQ	誤り訂正レベル Q
SCBQRCodeLevelH	誤り訂正レベル H

ApiFunctions.m / 各コマンド仕様書を参照ください。

8.56 SCBBitmapConverterRotation コンスタント

ビットマップ回転指定定数。

宣言

```
typedef NS_ENUM(NSUInteger, SCBBitmapConverterRotation) {
    SCBBitmapConverterRotationNormal,
    SCBBitmapConverterRotationRight90,
    SCBBitmapConverterRotationLeft90,
    SCBBitmapConverterRotationRotate180
};
```

定数

名称	説明
SCBBitmapConverterRotationNormal	回転なし
SCBBitmapConverterRotationRight90	右 90 度回転
SCBBitmapConverterRotationLeft90	左 90 度回転
SCBBitmapConverterRotationRotate180	180 度回転

ApiFunctions.m を参照ください。

8.57 SCBBlackMarkType コンスタント

ブラックマーク指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SCBBlackMarkType) {
    SCBBlackMarkTypeInvalid,
    SCBBlackMarkTypeValid,
    SCBBlackMarkTypeValidWithDetection
};
```

定数

名称	説明
SCBBlackMarkTypeInvalid	ブラックマーク無効
SCBBlackMarkTypeValid	ブラックマーク有効
SCBBlackMarkTypeValidWithDetection	ブラックマーク有効 + ブラックマーク検出動作

ApiFunctions.m / 各コマンド仕様書を参照ください。

9 ISDCBBuilder インターフェイス (StarIO_Extension.framework)

カスタマーディスプレイ制御用コマンドを生成する機能を提供するインターフェイスです。

メソッド

名称	説明
appendByte appendData appendBytes	データ（テキストやコマンド）を commands プロパティに追加します。
appendBackSpace	バックスペースコマンドを生成し、commands プロパティに追加します。
appendHorizontalTab	ホリゾンタルタブコマンドを生成し、commands プロパティに追加します。
appendLineFeed	ラインフィードコマンドを生成し、commands プロパティに追加します。
appendCarriageReturn	キャリッジリターンコマンドを生成し、commands プロパティに追加します。
appendBitmap	グラフィック表示コマンドを生成し、commands プロパティに追加します。
appendInternational	国際文字指定コマンドを生成し、commands プロパティに追加します。
appendCodePage	コードページ指定コマンドを生成し、commands プロパティに追加します。
appendDeleteToEndOfLine	画面を行末までクリアするコマンドを生成し、commands プロパティに追加します。
appendClearScreen	画面をクリアするコマンドを生成し、commands プロパティに追加します。
appendHomePosition	カーソルをホームポジションに移動するコマンドを生成し、commands プロパティに追加します。
appendTurnOn	バックライトをオン・オフするコマンドを生成し、commands プロパティに追加します。
appendSpecifiedPosition	カーソルを指定された位置に移動するコマンドを生成し、commands プロパティに追加します。
appendCursorMode	カーソルモード変更コマンドを生成し、commands プロパティに追加します。
appendContrastMode	コントラストモード変更コマンドを生成し、commands プロパティに追加します。
appendUserDefinedCharacter	ユーザー定義文字(SBCS)登録コマンドを生成し、commands プロパティに追加します。
appendUserDefinedDbcsCharacter	ユーザー定義文字(DBCS)登録コマンドを生成し、commands プロパティに追加します。

プロパティ

名称	説明
commands	生成、追加されたコマンド列です。* <i>readonly</i>
passThroughCommands	生成、追加されたコマンド列の先頭にプリンタへのパススルーコマンドが追加されたコマンド列です。* <i>readonly</i>

コンスタント

名称	説明
SDCBIternationalType	国際文字指定定数。
SDCBCodePageType	コードページ指定定数。
SDCBCursorMode	カーソルモード指定定数。
SDCBCContrastMode	コントラストモード指定定数。

9.1 モデル : ISDCBBUILDER インターフェイス

ISDCBBUILDER インターフェイスは mPOP/TSP100IIIU 専用です。

9.2 appendData メソッド

データ（テキストやコマンド）を commands プロパティに追加します。

宣言

- (void)appendByte:(unsigned char)data;
- (void)appendData:(NSData *)otherData;
- (void)appendBytes:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
data	データ（テキストやコマンド）	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

Example

```
+ (void)appendTextPattern:(ISDCBBUILDER *)builder number:(int)number {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

unsigned char pattern1[] =
"\x020\x021\x022\x023\x024\x025\x026\x027\x028\x029\x02a\x02b\x02c\x02d\x02e\x02f\x030\x031\x032\x033"
"\x034\x035\x036\x037\x038\x039\x03a\x03b\x03c\x03d\x03e\x03f\x040\x041\x042\x043\x044\x045\x046\x047";
...

switch (number) {
default : [builder appendBytes:pattern1 length:sizeof(pattern1)]; break; // 0
case 1 : [builder appendBytes:pattern2 length:sizeof(pattern2)]; break;
case 2 : [builder appendBytes:pattern3 length:sizeof(pattern3)]; break;
case 3 : [builder appendBytes:pattern4 length:sizeof(pattern4)]; break;
case 4 : [builder appendBytes:pattern5 length:sizeof(pattern5)]; break;
case 5 : [builder appendBytes:pattern6 length:sizeof(pattern6)]; break;
}
}
```

DisplayFunctions.m を参照ください。

9.3 appendBackSpace メソッド

バックスペースコマンドを生成し、commands プロパティに追加します。

宣言

```
void appendBackSpace();
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.4 appendHorizontalTab メソッド

ホリゾンタルタブコマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendHorizontalTab;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.5 appendLineFeed メソッド

ラインフィードコマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendLineFeed;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.6 appendCarriageReturn メソッド

キャリッジリターンコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendCarriageReturn;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.7 appendBitmap メソッド

グラフィック表示コマンドを生成し、commands プロパティに追加します。

宣言

```
- (void)appendBitmap:(UIImage *)image diffusion:(BOOL)diffusion;
```

引数

名称	説明	型
image	表示したいイメージ 横 160dot、縦 40dot のモノクロ Bitmap クラスのインスタンスを指定してください。 異なるサイズの Bitmap が指定された場合、横 160dot、縦 40dot にリサイズされます。 Bitmap の各ピクセルの色は自動でモノクロ 2 値に変換されます。	UIImage *
diffusion	誤差拡散 <ul style="list-style-type: none"> true ... 誤差拡散を行う false ... 誤差拡散を行わない 	BOOL

戻り値

説明	型
-	-

Example

```
+ (void)appendGraphicPattern:(ISDCBBuilder *)builder number:(int)number {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
// [builder appendSpecifiedPosition:1 y:1];

UIImage *image;

switch (number) {
default : image = [UIImage imageNamed:@"DisplayImage1.png"]; break; // 0
case 1 : image = [UIImage imageNamed:@"DisplayImage2.png"]; break;
case 2 : image = [UIImage imageNamed:@"DisplayImage3.png"]; break;
case 3 : image = [UIImage imageNamed:@"DisplayImage4.png"]; break;
}

[builder appendBitmap:image diffusion:YES];
}
```

DisplayFunctions.m を参照ください。

9.8 appendInternational メソッド

国際文字指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendInternational:(SDCBInternationalType)type;

引数

名称	説明	型
type	国際文字指定定数	SDCBInternationalType

戻り値

説明	型
-	-

Example

```
+ (void)appendCharacterSet:(ISDCBBuilder *)builder internationalType:(SDCBInternationalType)internationalType
codePageType:(SDCBCodePageType)codePageType {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:internationalType];
[builder appendCodePage :codePageType];

    unsigned char pattern1[] =
    "\x02d\x020\x020\x020\x023\x024\x040\x05b\x05c\x05d\x05e\x060\x07b\x07c\x07d\x07e\x020\x020\x020\x02d"

    "\x0a0\x0a1\x0a2\x0a3\x0a4\x0a5\x0a6\x0a7\x0a8\x0a9\x0aa\x0ab\x0ac\x0ad\x0ae\x0af\x0b0\x0b1\x0b2\x0b3";

    ...

    switch (codePageType) {
        default : [builder appendBytes:pattern1 length:sizeof(pattern1)]; break; //
CP437,Katakana,CP850,CP860,CP863,CP865,CP1252,CP866,CP852,CP858
        case SDCBCodePageTypeJapanese : [builder appendBytes:pattern2 length:sizeof(pattern2)]; break;
        case SDCBCodePageTypeSimplifiedChinese : [builder appendBytes:pattern3 length:sizeof(pattern3)]; break;
        case SDCBCodePageTypeTraditionalChinese : [builder appendBytes:pattern4 length:sizeof(pattern4)]; break;
        case SDCBCodePageTypeHangul : [builder appendBytes:pattern5 length:sizeof(pattern5)]; break;
    }
}
```

DisplayFunctions.m を参照ください。

9.9 appendCodePage メソッド

コードページ指定コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendCodePage:(SDCBCCodePageType)type;

引数

名称	説明	型
type	コードページ指定定数	SDCBCCodePageType

戻り値

説明	型
-	-

Example

```
+ (void)appendCharacterSet:(ISDCBBuilder *)builder internationalType:(SDCBInternationalType)internationalType
codePageType:(SDCBCCodePageType)codePageType {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:internationalType];
[builder appendCodePage :codePageType];

unsigned char pattern1[] =
"\x02d\x020\x020\x020\x023\x024\x040\x05b\x05c\x05d\x05e\x060\x07b\x07c\x07d\x07e\x020\x020\x020\x02d"
"\x0a0\x0a1\x0a2\x0a3\x0a4\x0a5\x0a6\x0a7\x0a8\x0a9\x0aa\x0ab\x0ac\x0ad\x0ae\x0af\x0b0\x0b1\x0b2\x0b3";

...

switch (codePageType) {
default : [builder appendBytes:pattern1 length:sizeof(pattern1)]; break; //
CP437,Katakana,CP850,CP860,CP863,CP865,CP1252,CP866,CP852,CP858
case SDCBCCodePageTypeJapanese : [builder appendBytes:pattern2 length:sizeof(pattern2)]; break;
case SDCBCCodePageTypeSimplifiedChinese : [builder appendBytes:pattern3 length:sizeof(pattern3)]; break;
case SDCBCCodePageTypeTraditionalChinese : [builder appendBytes:pattern4 length:sizeof(pattern4)]; break;
case SDCBCCodePageTypeHangul : [builder appendBytes:pattern5 length:sizeof(pattern5)]; break;
}
}
```

DisplayFunctions.m を参照ください。

9.10 appendDeleteToEndOfLine メソッド

画面を行末までクリアするコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendDeleteToEndOfLine;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.11 appendClearScreen メソッド

画面をクリアするコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendClearScreen;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
+ (void)appendClearScreen:(ISDCBBuilder *)builder {
    [builder appendClearScreen];
}
```

DisplayFunctions.m を参照ください。

9.12 appendHomePosition メソッド

カーソルをホームポジションに移動するコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendHomePosition;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

9.13 appendTurnOn メソッド

バックライトをオン・オフするコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendTurnOn:(BOOL)turnOn;

引数

名称	説明	型
turnOn	バックライトのオン・オフ	BOOL

戻り値

説明	型
-	-

Example

```
+ (void)appendTurnOn:(ISDCBBuilder *)builder turnOn:(BOOL)turnOn {
    ...
    [builder appendTurnOn:turnOn];
}
```

DisplayFunctions.m を参照ください。

9.14 appendSpecifiedPosition メソッド

カーソルを指定された位置に移動するコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendSpecifiedPosition:(int)x y:(int)y;

引数

名称	説明	型
x	桁位置（一番左の桁が 1）	int
y	行位置（一番上の行が 1）	int

戻り値

説明	型
-	-

Example

```
+ (void)appendCursorMode:(ISDCBBuilder *)builder cursorMode:(SDCBCursorMode)cursorMode {
// [builder appendClearScreen];
    [builder appendCursorMode:SDCBCursorModeOff];
    [builder appendSpecifiedPosition:1 y:1];

    unsigned char pattern[] = "Star Micronics    "
        "Total :    12345";

    [builder appendBytes:pattern length:sizeof(pattern)];

    [builder appendSpecifiedPosition:20 y:2];

    [builder appendCursorMode:cursorMode];
}
```

DisplayFunctions.m を参照ください。

9.15 appendCursorMode メソッド

カーソルモード変更コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendCursorMode:(SDCBCursorMode)cursorMode;

引数

名称	説明	型
cursorMode	カーソルモード指定定数	SDCBCursorMode

戻り値

説明	型
-	-

Example

```
+ (void)appendCursorMode:(ISDCBBuilder *)builder cursorMode:(SDCBCursorMode)cursorMode {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

unsigned char pattern[] = "Star Micronics    "
                        "Total :    12345";

[builder appendBytes:pattern length:sizeof(pattern)];

[builder appendSpecifiedPosition:20 y:2];

[builder appendCursorMode:cursorMode];
}
```

DisplayFunctions.m を参照ください。

9.16 appendContrastMode メソッド

コントラストモード変更コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendContrastMode:(SDCBCContrastMode)contrastMode;

引数

名称	説明	型
contrastMode	コントラストモード指定定数	SDCBCContrastMode

戻り値

説明	型
-	-

Example

```
+ (void)appendContrastMode:(ISDCBBuilder *)builder contrastMode:(SDCBCContrastMode)contrastMode {
...
[builder appendContrastMode:contrastMode];
}
```

DisplayFunctions.m を参照ください。

9.17 appendUserDefinedCharacter メソッド

ユーザー定義文字(SBCS)登録コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendUserDefinedCharacter:(int)index code:(int)code font:(unsigned char *)font;

引数

名称	説明	型
index	フォント番号 $00h \leq \text{index} \leq 1Fh$	int
code	ユーザー定義文字の文字コード $20h \leq \text{code} \leq 7Fh$ appendCodePage メソッドで SDCBCodePageTypeJapanese を指定した場合、以下の範囲も使用可能です。 $A0h \leq \text{code} \leq DFh$	int
font	16byte のフォントデータ ユーザー定義文字のフォントデータのフォーマットを参照してください。 null を指定すると、指定したフォント番号のユーザー定義文字を解除します。	unsigned char *

戻り値

説明	型
-	-

Example

```
+ (void)appendUserDefinedCharacter:(ISDCBBuilder *)builder set:(BOOL)set {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:SDCBInternationalTypeUSA];
[builder appendCodePage :SDCBCCodePageTypeJapanese];

if (set) {
    [builder appendUserDefinedCharacter:0 code:0x20 font:(unsigned char *)
"\x000\x000\x032\x000\x049\x000\x049\x07f\x026\x048\x000\x048\x000\x030\x000\x000"];

    [builder appendUserDefinedDbcsCharacter:0 code:0x8140 font:(unsigned char *)
"\x000\x000\x000\x000\x000\x000\x000\x000\x003\x020\x004\x090\x004\x090\x002\x060"
"\x000\x000\x007\x0f0\x004\x080\x004\x080\x003\x000\x000\x000\x000\x000\x000"];
}
else {
    [builder appendUserDefinedCharacter:0 code:0x00 font:nil];

    [builder appendUserDefinedDbcsCharacter:0 code:0x0000 font:nil];
}

unsigned char pattern[] =
"\x05b\x020\x020\x053\x074\x061\x072\x020\x04d\x069\x063\x072\x06f\x06e\x069\x063\x073\x020\x020\x05d"
"\x05b\x081\x040\x081\x040\x083\x058\x083\x05e\x081\x05b\x090\x0b8\x096\x0a7\x081\x040\x081\x040\x05d";

[builder appendBytes:pattern length:sizeof(pattern)];
}
```

DisplayFunctions.m を参照ください。

9.18 appendUserDefinedDbcsCharacter メソッド

ユーザー定義文字(DBCS)登録コマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendUserDefinedDbcsCharacter:(int)index code:(int)code font:(unsigned char *)font;

引数

名称	説明	型
index	フォント番号 $00h \leq \text{index} \leq 0Fh$	int
code	ユーザー定義文字の文字コード $20h \leq \text{code} \leq 7Fh$ appendCodePage メソッドで SDCBCodePageTypeJapanese を指定した場合、以下の範囲も使用可能です。 $80\ 00h \leq \text{code} \leq FF\ FFh$	int
font	32byte のフォントデータ ユーザー定義文字のフォントデータのフォーマットを参照してください。 null を指定すると、指定したフォント番号のユーザー定義文字を解除します。	unsigned char *

戻り値

説明	型
-	-

Example

```
+ (void)appendUserDefinedCharacter:(ISDCBBuilder *)builder set:(BOOL)set {
// [builder appendClearScreen];
[builder appendCursorMode:SDCBCursorModeOff];
[builder appendSpecifiedPosition:1 y:1];

[builder appendInternational:SDCBInternationalTypeUSA];
[builder appendCodePage :SDCBCCodePageTypeJapanese];

if (set) {
[builder appendUserDefinedCharacter:0 code:0x20 font:(unsigned char *)
"\x000\x000\x032\x000\x049\x000\x049\x07f\x026\x048\x000\x048\x000\x030\x000\x000"];

[builder appendUserDefinedDbcsCharacter:0 code:0x8140 font:(unsigned char *)
"\x000\x000\x000\x000\x000\x000\x000\x000\x003\x020\x004\x090\x004\x090\x002\x060"

"\x000\x000\x007\x0f0\x004\x080\x004\x080\x003\x000\x000\x000\x000\x000\x000\x000"];
}
else {
[builder appendUserDefinedCharacter:0 code:0x00 font:nil];

[builder appendUserDefinedDbcsCharacter:0 code:0x0000 font:nil];
}

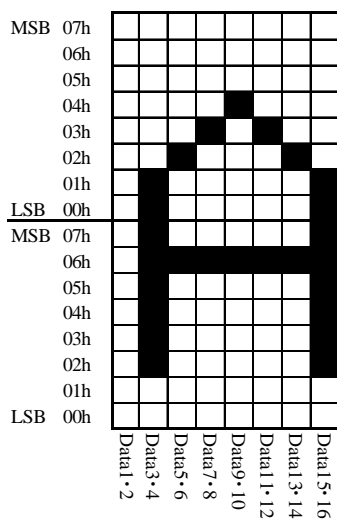
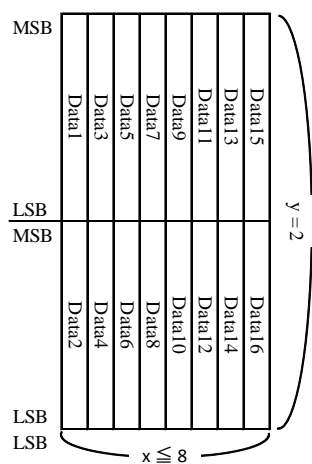
unsigned char pattern[] =
"\x05b\x020\x020\x053\x074\x061\x072\x020\x04d\x069\x063\x072\x06f\x06e\x069\x063\x073\x020\x020\x05d"
"\x05b\x081\x040\x081\x040\x083\x058\x083\x05e\x081\x05b\x090\x0b8\x096\x0a7\x081\x040\x081\x040\x05d";

[builder appendBytes:pattern length:sizeof(pattern)];
}
```

DisplayFunctions.m を参照ください。

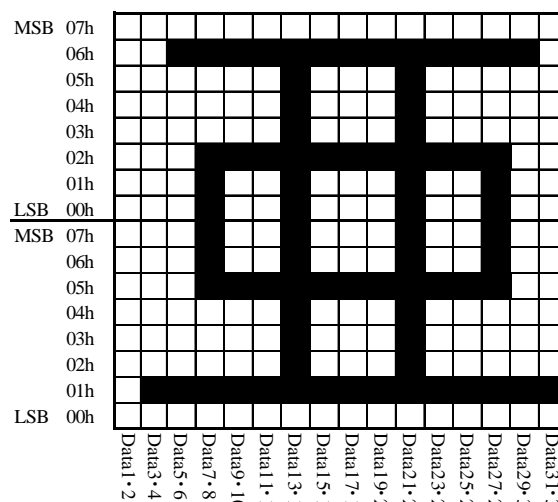
ユーザ定義文字のフォントデータのフォーマット

ユーザ一定義文字 (SBCD)



Byte	1	3	5	7	9	11	13	15
Data	00h	03h	04h	08h	10h	08h	04h	03h
Byte	2	4	6	8	10	12	14	16
Data	00h	FCh	40h	40h	40h	40h	40h	FCh

ユーザ一定義文字 (DBCS)



Byte	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
Data	00h	00h	40h	47h	44h	44h	7Fh	44h	44h	44h	7Fh	44h	44h	47h	44h	00h
Byte	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
Data	00h	02h	02h	E2h	22h	22h	FEh	22h	22h	22h	FEh	22h	22h	E2h	22h	02h

9.19 commands プロパティ

生成、追加されたコマンド列です。

宣言

@property (nonatomic, readonly) NSMutableData *commands;

値

説明	型
生成、追加されたコマンド列	NSMutableData

9.20 passThroughCommands プロパティ

生成、追加されたコマンド列の先頭にプリンタへのパススルーコマンドが追加されたコマンド列です。

宣言

@property (nonatomic, readonly) NSMutableData *passThroughCommands;

値

説明	型
生成、追加されたコマンド列	NSMutableData

Example

```
ISDCBBuilder *builder = [StarloExt createDisplayCommandBuilder:StarloExtDisplayModelISCD222];
```

```
...
```

```
NSData *commands = [builder.passThroughCommands copy];
```

DisplayViewController.m / DisplayExtViewController.m を参照ください。

9.21 SDCBInternationalType コンスタント

国際文字指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SDCBInternationalType) {
    SDCBInternationalTypeUSA      = 0x00,
    SDCBInternationalTypeFrance   = 0x01,
    SDCBInternationalTypeGermany  = 0x02,
    SDCBInternationalTypeUK       = 0x03,
    SDCBInternationalTypeDenmark  = 0x04,
    SDCBInternationalTypeSweden   = 0x05,
    SDCBInternationalTypeItaly    = 0x06,
    SDCBInternationalTypeSpain    = 0x07,
    SDCBInternationalTypeJapan    = 0x08,
    SDCBInternationalTypeNorway   = 0x09,
    SDCBInternationalTypeDenmark2 = 0x0a,
    SDCBInternationalTypeSpain2   = 0x0b,
    SDCBInternationalTypeLatinAmerica = 0x0c,
    SDCBInternationalTypeKorea    = 0x0d
};
```

定数

名称	説明
SDCBInternationalTypeUSA	USA
SDCBInternationalTypeFrance	フランス
SDCBInternationalTypeGermany	ドイツ
SDCBInternationalTypeUK	イギリス
SDCBInternationalTypeDenmark	デンマーク
SDCBInternationalTypeSweden	スウェーデン
SDCBInternationalTypeItaly	イタリア
SDCBInternationalTypeSpain	スペイン
SDCBInternationalTypeJapan	日本
SDCBInternationalTypeNorway	ノルウェー
SDCBInternationalTypeDenmark2	デンマーク II
SDCBInternationalTypeSpain2	スペイン II
SDCBInternationalTypeLatinAmerica	ラテンアメリカ
SDCBInternationalTypeKorea	韓国

9.22 SDCBCodePageType コンスタント

コードページ指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SDCBCodePageType) {
    SDCBCodePageTypeCP437      = 0x00,
    SDCBCodePageTypeKatakana   = 0x01,
    SDCBCodePageTypeCP850      = 0x02,
    SDCBCodePageTypeCP860      = 0x03,
    SDCBCodePageTypeCP863      = 0x04,
    SDCBCodePageTypeCP865      = 0x05,
    SDCBCodePageTypeCP1252     = 0x06,
    SDCBCodePageTypeCP866      = 0x07,
    SDCBCodePageTypeCP852      = 0x08,
    SDCBCodePageTypeCP858      = 0x09,
    SDCBCodePageTypeJapanese   = 0x0a,
    SDCBCodePageTypeSimplifiedChinese = 0x0b,
    SDCBCodePageTypeTraditionalChinese = 0x0c,
    SDCBCodePageTypeHangul     = 0x0d
};
```

定数

名称	説明
SDCBCodePageTypeCP437	CodePage437 (USA, Std. Europe).
SDCBCodePageTypeKatakana	Katakana.
SDCBCodePageTypeCP850	PC850 (Multilingual)
SDCBCodePageTypeCP860	PC860 (Portuguese)
SDCBCodePageTypeCP863	PC863 (Canadian-French)
SDCBCodePageTypeCP865	PC865 (Norwegian)
SDCBCodePageTypeCP1252	WPC1252
SDCBCodePageTypeCP866	PC866 [Cyrillic #2]
SDCBCodePageTypeCP852	PC852 [Latin 2]
SDCBCodePageTypeCP858	Page 19 [PC858]
SDCBCodePageTypeJapanese	Japanese font (shift JIS)
SDCBCodePageTypeSimplifiedChinese	Simplified Chinese (GB2312)
SDCBCodePageTypeTraditionalChinese	Traditional Chinese (Big5)
SDCBCodePageTypeHangul	Hangul (KSC5601)

9.23 SDCBCursorMode コンスタント

カーソルモード指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SDCBCursorMode) {
    SDCBCursorModeOff = 0x00,
    SDCBCursorModeBlink = 0x01,
    SDCBCursorModeOn = 0x02
};
```

定数

名称	説明
SDCBCursorModeOff	カーソル消灯
SDCBCursorModeBlink	カーソル点滅
SDCBCursorModeOn	カーソル点灯

9.24 SDCBContrastMode コンスタント

コントラストモード指定定数。

宣言

```
typedef NS_ENUM(NSInteger, SDCBContrastMode) {
    SDCBContrastModeMinus3 = 0x00,
    SDCBContrastModeMinus2 = 0x01,
    SDCBContrastModeMinus1 = 0x02,
    SDCBContrastModeDefault = 0x03,
    SDCBContrastModePlus1 = 0x04,
    SDCBContrastModePlus2 = 0x05,
    SDCBContrastModePlus3 = 0x06
};
```

定数

名称	説明
SDCBContrastModeMinus3	コントラストマイナス 3
SDCBContrastModeMinus2	コントラストマイナス 2
SDCBContrastModeMinus1	コントラストマイナス 1
SDCBContrastModeDefault	デフォルト
SDCBContrastModePlus1	コントラストプラス 1
SDCBContrastModePlus2	コントラストプラス 2
SDCBContrastModePlus3	コントラストプラス 3

10 ISSCBBUILDER インターフェイス (StarIO_Extension.framework)

スケール（はかり）制御用コマンドを生成する機能を提供するインターフェイスです。

メソッド

Name	Description
appendByte	データを commands プロパティに追加します。
appendData	
appendBytes	
appendZeroClear	スケールの重量をゼロクリアするコマンドを生成し、コマンドバッファに追加します。
appendUnitChange	スケールの単位を切り替えるコマンドを生成し、コマンドバッファに追加します。

プロパティ

名称	説明
commands	生成、追加されたコマンド列です。* <i>readonly</i>
passThroughCommands	生成、追加されたコマンド列の先頭にプリンタへのパススルーコマンドが追加されたコマンド列です。* <i>readonly</i>

10.1 モデル : ISSCBBUILDER インターフェイス

ISSCBBUILDER インターフェイスは mPOP 専用です。

10.2 appendData メソッド

データを commands プロパティに追加します。

宣言

- (void)appendByte:(unsigned char)data;
- (void)appendData:(NSData *)otherData;
- (void)appendBytes:(const void *)bytes length:(NSUInteger)length;

引数

名称	説明	型
data	データ	unsigned char
otherData		NSData
bytes		const void *
length		NSUInteger

戻り値

説明	型
-	-

10.3 appendZeroClear メソッド

スケールの重量をゼロクリアするコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendZeroClear;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
+ (void)appendZeroClear:(ISSCBuilder *)builder {
    [builder appendZeroClear];
}
```

ScaleFunctions.m を参照ください。

10.4 appendUnitChange メソッド

スケールの単位を切り替えるコマンドを生成し、commands プロパティに追加します。

宣言

- (void)appendUnitChange;

引数

名称	説明	型
-	-	-

戻り値

説明	型
-	-

Example

```
+ (void)appendUnitChange:(ISSCBuilder *)builder {
    [builder appendUnitChange];
}
```

ScaleFunctions.m を参照ください。

10.5 commands プロパティ

生成、追加されたコマンド列です。

宣言

@property (nonatomic, readonly) NSMutableData *commands;

値

説明	型
生成、追加されたコマンド列	NSMutableData

10.6 passThroughCommands プロパティ

生成、追加されたコマンド列の先頭にプリンタへのパススルーコマンドが追加されたコマンド列です。

宣言

@property (nonatomic, readonly) NSMutableData *passThroughCommands;

値

説明	型
生成、追加されたコマンド列	NSMutableData

Example

```
ISSCBBuilder *builder = [StarIoExt createScaleCommandBuilder:StarIoExtScaleModelAPS20];

[ScaleFunctions appendZeroClear:builder];

NSData *commands = [builder.passThroughCommands copy];

[Communication sendCommandsDoNotCheckCondition:commands port:port completionHandler:^(BOOL result, NSString
*title, NSString *message) {
    UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:title message:message delegate:nil cancelButtonTitle:@"OK"
otherButtonTitles:nil];

    [alertView show];
}];
```

ScaleViewController.m / ScaleExtViewController.m を参照ください。

11 ISCPParser インターフェイス (StarIO_Extension.framework)

外部機器（バーコードリーダー・カスタマディスプレイ・スケール）制御用コマンドの応答を解析する機能を提供するインターフェイスです。

メソッド

名称	説明
createSendCommands	外部機器（スケール・カスタマディスプレイ・バーコードリーダー）からの応答を受け取るコマンドを生成します。
createReceiveCommands	外部機器（スケール）からの応答を受け取るコマンドを生成します。

プロパティ

名称	説明
completionHandler	createSendCommands・createReceiveCommands で生成したコマンドを送信後の、外部機器（スケール・カスタマディスプレイ・バーコードリーダー）からのコマンド応答を解析します。

コンスタント

名称	説明
StarIoExtParserCompletionResult	解析結果定数。

11.1 モデル : ISCPParser インターフェイス

ISCPParser インターフェイスは mPOP/TSP100IIIU 専用です。

11.2 createSendCommands メソッド

外部機器(スケール・カスタマディスプレイ・バーコードリーダー)からの応答を受け取るコマンドを生成します。

宣言

- (NSData *)createSendCommands;

引数

名称	説明	型
-	-	-

戻り値

説明	型
生成されたコマンド列	NSData *

Example

```
+ (BOOL)parseDoNotCheckCondition:(ISCPParser *)parser
    port:(SMPort *)port
    completionHandler:(SendCompletionHandler)completionHandler {
    BOOL result = NO;

    NSString *title = @"";
    NSString *message = @"";

    NSData *sendCommands = [parser createSendCommands];
    NSData *receiveCommands = [parser createReceiveCommands];

    @try {
        while (YES) {
            ...

            while (result == NO) {
                ...

                while (total < (uint32_t) sendCommands.length) {
                    uint32_t written = [port writePort:sendCommands.bytes :total :(uint32_t) sendCommands.length - total];

                    total += written;

                    if ([NSDate date] timeIntervalSinceDate:startDate] >= 30.0) { // 30000mS!!!
                        title = @"Printer Error";
                        message = @"Write port timed out";
                        break;
                    }
                }

                if (total < (uint32_t) sendCommands.length) {
                    break;
                }

                NSDate *innerStartDate = [NSDate date];

                while (result == NO) {
                    ...

                    while (total < (uint32_t) receiveCommands.length) {
                        uint32_t written = [port writePort:receiveCommands.bytes :total :(uint32_t) receiveCommands.length -
total];

                        total += written;
```

```

        if ([NSDate date] timeIntervalSinceDate:innerStartDate] >= 30.0) {    // 30000mS!!!
            title = @"Printer Error";
            message = @"Write port timed out";
            break;
        }
    }
    ...

    while (completionResult == StarloExtParserCompletionResultInvalid) {
        ...

        int readLength = [port readPort:buffer :amount :1024 - amount];

        ...

        completionResult = parser.completionHandler(buffer, &amount);

        if (completionResult == StarloExtParserCompletionResultSuccess) {
            title = @"Send Commands";
            message = @"Success";

            result = YES;
        }
    }
}
break;
}
...
}

```

Communication.m / ScaleCommunication.m を参照ください。

11.3 createReceiveCommands メソッド

外部機器(スケール)からの応答を受け取るコマンドを生成します。

宣言

- (NSData *)createReceiveCommands;

引数

名称	説明	型
-	-	-

戻り値

説明	型
生成されたコマンド列	NSData *

Example

createSendCommands メソッドの Example を参照ください。

11.4 completionHandler プロパティ

createSendCommands・createReceiveCommands で生成したコマンドを送信後の、外部機器(スケール・カスタマディスプレイ・バーコードリーダー)からのコマンド応答を解析します。

宣言

```
typedef StarloExtParserCompletionResult (^StarloExtParserCompletionHandler)(uint8_t *buffer, int *length);
```

```
@property (nonatomic, copy) StarloExtParserCompletionHandler completionHandler;
```

値

説明	型
解析ハンドラ	StarloExtParserCompletionHandler

Example

createSendCommands メソッドの Example を参照ください。

11.5 StarloExtParserCompletionResult コンスタント

解析結果定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtParserCompletionResult) {
    StarloExtParserCompletionResultInvalid = 0,
    StarloExtParserCompletionResultSuccess,
    StarloExtParserCompletionResultFailure
};
```

定数

名称	説明
StarloExtParserCompletionResultInvalid	不定 (コマンド応答未完了)
StarloExtParserCompletionResultSuccess	成功
StarloExtParserCompletionResultFailure	失敗

12 ISCPConnectParser インターフェイス (StarIO_Extension.framework)

外部機器(スケール・カスタマーディスプレイ・バーコードリーダー)の接続・切断ステータスを取得する機能を提供するインターフェイスです。

宣言

@interface ISCPConnectParser : ISCPParser

メソッド

名称	説明
connect	外部機器(スケール・カスタマーディスプレイ・バーコードリーダー)の接続・切断ステータスを取得します。

12.1 モデル : ISCPConnectParser インターフェイス

ISCPConnectParser インターフェイスは mPOP/TSP100IIIU 専用です。

12.2 connect メソッド

外部機器(スケール・カスタマーディスプレイ・バーコードリーダー)の接続・切断ステータスを取得します。

completionHandler が StarloExtParserCompletionResultSuccess を返した後に実行してください。

宣言

- (BOOL)connect;

引数

名称	説明	型
-	-	-

戻り値

説明	型
接続・切断ステータス	BOOL

Example

```
ISCPConnectParser *parser = [StarloExt createDisplayConnectParser:StarloExtDisplayModelSCD222];

[Communication parseDoNotCheckCondition:parser port:port completionHandler:^(BOOL result, NSString *title, NSString *message) {
    if (result == YES) {
        if (parser.connect == YES) {
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Check Status" message:@"Display Connect."
            delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];

            [alertView show];
        }
        else {
            UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Check Status" message:@"Display Disconnect."
            delegate:nil cancelButtonTitle:@"OK" otherButtonTitles:nil];

            [alertView show];
        }
    }
    else {
        // UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Failure" message:@"Display Impossible." delegate:nil
        cancelButtonTitle:@"OK" otherButtonTitles:nil];
        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"Failure" message:@"Printer Impossible." delegate:nil
        cancelButtonTitle:@"OK" otherButtonTitles:nil];

        [alertView show];
    }
}];
```

DisplayViewController.m / DisplayExtViewController.m / ScaleViewController.m / ScaleExtViewController.m を参照ください。

13 ISSCPWeightParser (StarIO_Extension.framework)

スケール制御用コマンド応答を解析する機能とスケール表示重量を取得する機能を提供するインターフェイスです。

宣言

@interface ISSCPWeightParser : ISCPParser

メソッド

名称	説明
weight	スケール表示重量を取得します。
status	スケールの計測ステータスを取得します。

コンスタント

名称	説明
StarIoExtDisplayedWeightStatus	重量の計測状態定数。

13.1 モデル : ISSCPWeightParser インターフェイス

ISSCPWeightParser インターフェイスは mPOP 専用です。

13.2 weight メソッド

スケール表示重量を取得します。

completionHandler が StarloExtParserCompletionResultSuccess を返した後に実行してください。

宣言

-(NSString *)weight;

引数

名称	説明	型
-	-	-

戻り値

説明	型
スケールの表示重量	NSString *

Example

```
[ScaleCommunication parseDoNotCheckCondition:weightParser port:port completionHandler:^(BOOL result, NSString
 *title, NSString *message) {
    if (result == YES) {
        UIAlertView *alertView;

        switch (weightParser.status) {
            default :
                // case StarloExtDisplayedWeightStatusZero :
                alertView = [[UIAlertView alloc] initWithTitle:@"Success [Zero]"
                    message:weightParser.weight
                    delegate:nil
                    cancelButtonTitle:@"OK"
                    otherButtonTitles:nil];

                break;
            case StarloExtDisplayedWeightStatusNotInMotion :
                alertView = [[UIAlertView alloc] initWithTitle:@"Success [Not in motion]"
                    message:weightParser.weight
                    delegate:nil
                    cancelButtonTitle:@"OK"
                    otherButtonTitles:nil];

                break;
            case StarloExtDisplayedWeightStatusMotion :
                alertView = [[UIAlertView alloc] initWithTitle:@"Success [Motion]"
                    message:weightParser.weight
                    delegate:nil
                    cancelButtonTitle:@"OK"
                    otherButtonTitles:nil];

                break;
        }

        [alertView show];
    }
    else { // Because the scale doesn't sometimes react.
        UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:title message:message delegate:nil
            cancelButtonTitle:@"OK" otherButtonTitles:nil];

        [alertView show];
    }
}];
```

ScaleViewController.m / ScaleExtViewController.m を参照ください。

13.3 status メソッド

スケール表示重量を取得します。

completionHandler が StarloExtParserCompletionResultSuccess を返した後に実行してください。

宣言

```
- (StarloExtDisplayedWeightStatus)status;
```

引数

名称	説明	型
-	-	-

戻り値

説明	型
スケールの計測ステータス	StarloExtDisplayedWeightStatus

Example

weight メソッドの Example を参照ください。

13.4 StarloExtDisplayedWeightStatus コンスタント

重量の計測状態定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtDisplayedWeightStatus) {
    StarloExtDisplayedWeightStatusInvalid = 0,
    StarloExtDisplayedWeightStatusZero,
    StarloExtDisplayedWeightStatusNotInMotion,
    StarloExtDisplayedWeightStatusMotion
};
```

定数

名称	説明
StarloExtDisplayedWeightStatusInvalid	不定
StarloExtDisplayedWeightStatusZero	ゼロクリアされた状態
StarloExtDisplayedWeightStatusNotInMotion	計測終了状態
StarloExtDisplayedWeightStatusMotion	計測中

14 StarPRNT iOS SDK Sample

14.1 Communication

StarIO iOS SDK 相当の印刷データ送信の一例と StarIoExtManager オブジェクトを使用した印刷データ送信の一例が用意されています。

-StarIoExtManager について-

StarIoExtManager を用いた場合、デバイス間が常時接続となります。

そのため、他のアプリケーションや他の端末とデバイスを共有する場合、また他のアプリケーションへの遷移やスリープ状態への遷移が想定される場合、デバイスとの接続状態に関する実装上の配慮が必要となります。

Example (StarIO iOS SDK 相当)

```
+ (BOOL)sendCommands:(NSData *)commands portName:(NSString *)portName portSettings:(NSString *)portSettings timeout:(NSInteger)timeout {
    BOOL result = NO;

    ...

    SMPort *port = nil;

    @try {
        while (YES) {
            port = [SMPort getPort:portName :portSettings :(uint32_t) timeout];

            if (port == nil) {

                ...

                break;
            }

            StarPrinterStatus_2 printerStatus;

            [port beginCheckedBlock:&printerStatus :2];

            if (printerStatus.offline == SM_TRUE) {

                ...

                break;
            }

            NSDate *startDate = [NSDate date];

            uint32_t total = 0;

            while (total < commandLength) {
                uint32_t written = [port writePort:commandsBytes :total :commandLength - total];

                total += written;

                if ([NSDate date] timeIntervalSinceDate:startDate] >= 30.0) { // 30000mS!!!
                    break;
                }
            }
        }
    }
```

```

    if (total < commandLength) {
        ...
        break;
    }

    port.endCheckedBlockTimeoutMillis = 30000; // 30000mS!!!
    [port endCheckedBlock:&printerStatus :2];

    if (printerStatus.offline == SM_TRUE) {
        ...
        break;
    }

    result = YES;
    break;
}
}
@catch (PortException *exc) {
    ...
}
@finally {
    if (port != nil) {
        [SMPort releasePort:port];
    }
}

...

return result;
}

```

Example (StarIoExtManager オブジェクトを使用)

```

+ (BOOL)sendCommands:(NSData *)commands port:(SMPort *)port {
    BOOL result = NO;

    ...

    @try {
        while (YES) {

            ...

            StarPrinterStatus_2 printerStatus;

            [port beginCheckedBlock:&printerStatus :2];

            if (printerStatus.offline == SM_TRUE) {

                ...

                break;
            }

            NSDate *startDate = [NSDate date];

            uint32_t total = 0;

```

```

while (total < commandLength) {
    uint32_t written = [port writePort:commandsBytes :total :commandLength - total];

    total += written;

    if ([[NSDate date] timeIntervalSinceDate:startDate] >= 30.0) { // 30000mS!!!
        break;
    }
}

if (total < commandLength) {

    ...

    break;
}

port.endCheckedBlockTimeoutMillis = 30000; // 30000mS!!!

[port endCheckedBlock:&printerStatus :2];

if (printerStatus.offline == SM_TRUE) {

    ...

    break;
}

result = YES;
break;
}
}
}catch (PortException *exc) {

    ...

}

...

return result;
}

```

15 StarIoExtManager クラス (StarIO_Extension.framework)

メソッド

名称	説明
initWithType	StarIoExtManager を初期化します。
connect	マネージメントを開始します。
disconnect	マネージメントを停止します。

プロパティ

名称	説明
port	SMPort オブジェクト。 * <i>readonly</i>
lock	port プロパティによる通信権利の排他制御オブジェクト。 * <i>readonly</i>
delegate	StarIoExtManager のデリゲートオブジェクト。
printerStatus	プリンタステータス。 * <i>readonly</i>
printerPaperStatus	プリンタ用紙ステータス。 * <i>readonly</i>
printerCoverStatus	プリンタカバーステータス。 * <i>readonly</i>
cashDrawerStatus	キャッシュドロアーステータス。 * <i>readonly</i>
barcodeReaderStatus	バーコードリーダーステータス。 * <i>readonly</i>
cashDrawerOpenActiveHigh	キャッシュドロア開閉センサのモード。

コンスタント

名称	説明
StarIoExtManagerType	マネージャタイプ定数。
StarIoExtManagerPrinterStatus	プリンタステータス定数。
StarIoExtManagerPrinterPaperStatus	プリンタ用紙ステータス定数。
StarIoExtManagerPrinterCoverStatus	プリンタカバーステータス定数。
StarIoExtManagerCashDrawerStatus	キャッシュドロアーステータス定数。
StarIoExtManagerBarcodeReaderStatus	バーコードリーダーステータス定数。

-Bluetooth 接続時の自動パワーダウンモードについて-

SM-S シリーズ、SM-T シリーズにおいては自動パワーダウンモードを“NO USE”（工場出荷時設定）に設定してください。

15.1 initWithType メソッド

StarloExtManager を初期化します。

宣言

```
-(id)initWithType:(StarloExtManagerType)type portName:(NSString *)portName
portSettings:(NSString *)portSettings ioTimeoutMillis:(NSUInteger)ioTimeoutMillis;
```

引数

名称	説明	型
type	マネージャタイプ <ul style="list-style-type: none"> StarloExtManagerTypeStandard ... プリンタ、キャッシュドロワのマネージメント StarloExtManagerTypeWithBarcodeReader ... プリンタ、キャッシュドロワ、バーコードリーダーのマネージメント StarloExtManagerTypeOnlyBarcodeReader ... バーコードリーダーのマネージメント 	StarloExtManagerType
portName	プリンタポート名	NSString *
portSettings	ポートセッティング	NSString *
ioTimeoutMillis	内部制御および API のタイムアウト値	NSUInteger

戻り値

説明	型
StarloExtManager オブジェクト	StarloExtManager

Example

```
-(void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    _starloExtManager = [[StarloExtManager alloc] initWithType:StarloExtManagerTypeStandard
                                                             portName:[AppDelegate getPortName]
                                                             portSettings:[AppDelegate getPortSettings]
                                                             ioTimeoutMillis:10000]; // 10000mS!!!

    _starloExtManager.delegate = self;
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

15.2 connect メソッド

マネージメントを開始します。

宣言

- (BOOL)connect;

引数

名称	説明	型
-	-	-

戻り値

説明	型
リザルト <ul style="list-style-type: none"> YES ... 成功 NO ... 失敗 	BOOL

Example

```
- (void)viewWillAppear:(BOOL)animated {
    [super viewWillAppear:animated];

    if (_starloExtManager.port != nil) {
        [_starloExtManager disconnect];
    }

    if ([_starloExtManager connect] == NO) {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Fail to Open Port."
                                                             message:@""
                                                             delegate:self
                                                             cancelButtonTitle:@"OK"
                                                             otherButtonTitles:nil];

        [alert show];
    }
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

15.3 disconnect メソッド

マネージメントを停止します。

宣言

- (BOOL)disconnect;

引数

名称	説明	型
-	-	-

戻り値

説明	型
リザルト	BOOL
• YES ... 成功	
• NO ... 失敗	

Example

```
- (void)viewWillDisappear:(BOOL)animated {
    [super viewWillDisappear:animated];

    [_starloExtManager disconnect];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

15.4 port プロパティ

SMPort オブジェクト。* *readonly*

宣言

@property (readonly, nonatomic) SMPort *port;

15.5 lock プロパティ

port プロパティによる通信権利の排他制御オブジェクト。* *readonly*

宣言

@property (readonly, nonatomic) NSRecursiveLock *lock;

Example

```
- (IBAction)touchUpInsidePrintButton:(id)sender {
    ...

    [_starloExtManager.lock lock];

    [Communication sendCommands:commands port:[_starloExtManager port]];

    [_starloExtManager.lock unlock];

    ...
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。

15.6 delegate プロパティ

StarloExtManager のデリゲートオブジェクト。

宣言

@property (weak, nonatomic) id<StarloExtManagerDelegate> delegate;

Example

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    _starloExtManager = [[StarloExtManager alloc] initWithType:StarloExtManagerTypeOnlyBarcodeReader
        portName:[AppDelegate getPortName]
        portSettings:[AppDelegate getPortSettings]
        ioTimeoutMillis:10000]; // 10000mS!!!

    _starloExtManager.delegate = self;
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

15.7 printerOnlineStatus プロパティ

プリンタオンラインステータス。* *readonly*

宣言

@property (readonly, nonatomic) StarloExtManagerPrinterStatus printerOnlineStatus;

値

説明	型
プリンタステータス <ul style="list-style-type: none"> StarloExtManagerPrinterStatusInvalid ... 不定 StarloExtManagerPrinterStatusImpossible ... プリンタ使用不能 StarloExtManagerPrinterStatusOnline ... プリンタオンライン検出 StarloExtManagerPrinterStatusOffline ... プリンタオフライン検出 	StarloExtManagerPrinterStatus

15.8 printerPaperStatus プロパティ

プリンタ用紙ステータス。* *readonly*

宣言

@property (readonly, nonatomic) StarloExtManagerPrinterPaperStatus printerPaperStatus;

値

説明	型
プリンタ用紙ステータス <ul style="list-style-type: none"> StarloExtManagerPrinterPaperStatusInvalid ... 不定 StarloExtManagerPrinterPaperStatusImpossible ... 機器使用不能 StarloExtManagerPrinterPaperStatusReady ... プリンタ用紙レディ検出 StarloExtManagerPrinterPaperStatusNearEmpty ... プリンタ用紙ニアエンド検出 StarloExtManagerPrinterPaperStatusEmpty ... プリンタ用紙エンpty検出 	StarloExtManagerPrinterPaperStatus

15.9 printerCoverStatus プロパティ

プリンタカバーステータス。* *readonly*

宣言

@property (readonly, nonatomic) StarloExtManagerPrinterCoverStatus printerCoverStatus;

値

説明	型
プリンタカバーステータス <ul style="list-style-type: none"> StarloExtManagerPrinterCoverStatusInvalid ... 不定 StarloExtManagerPrinterCoverStatusImpossible ... 機器使用不能 StarloExtManagerPrinterCoverStatusOpen ... プリンタカバーオープン検出 StarloExtManagerPrinterCoverStatusClose ... プリンタカバークローズ検出 	StarloExtManagerPrinterCoverStatus

15.10 cashDrawerStatus プロパティ

キャッシュドローステータス。* *readonly*

宣言

@property (readonly, nonatomic) StarloExtManagerCashDrawerStatus cashDrawerStatus;

値

説明	型
キャッシュドローステータス <ul style="list-style-type: none"> StarloExtManagerCashDrawerStatusInvalid ... 不定 StarloExtManagerCashDrawerStatusImpossible ... 機器使用不能 StarloExtManagerCashDrawerStatusOpen ... キャッシュドロアーオープン検出 StarloExtManagerCashDrawerStatusClose ... キャッシュドロアークローズ検出 	StarloExtManagerCashDrawerStatus

15.11 barcodeReaderStatus プロパティ

バーコードリーダーステータス。* *readonly*

宣言

@property (readonly, nonatomic) StarloExtManagerBarcodeReaderStatus barcodeReaderStatus;

値

説明	型
バーコードリーダーステータス <ul style="list-style-type: none"> StarloExtManagerBarcodeReaderStatusInvalid ... 不定 StarloExtManagerBarcodeReaderStatusImpossible ... 機器使用不能 StarloExtManagerBarcodeReaderStatusConnect ... バーコードリーダー接続検出 StarloExtManagerBarcodeReaderStatusDisconnect ... バーコードリーダー切断検出 	StarloExtManagerBarcodeReaderStatus

15.12 cashDrawerOpenActiveHigh プロパティ

キャッシュドロアー開閉センサのモード。

宣言

@property (nonatomic) BOOL cashDrawerOpenActiveHigh;

値

説明	型
キャッシュドロアー開閉センサモード <ul style="list-style-type: none"> YES ... High アクティブ NO ... Low アクティブ 	BOOL

15.13 StarloExtManagerType コンスタント

マネージャタイプ定数。

宣言

```
typedef NS_ENUM(NSUInteger, StarloExtManagerType) {
    StarloExtManagerTypeStandard = 0,
    StarloExtManagerTypeWithBarcodeReader,
    StarloExtManagerTypeOnlyBarcodeReader,
};
```

定数

名称	説明
StarloExtManagerTypeStandard	プリンタ、キャッシュドロワのマネージメント
StarloExtManagerTypeWithBarcodeReader	プリンタ、キャッシュドロワ、バーコードリーダーのマネージメント
StarloExtManagerTypeOnlyBarcodeReader	バーコードリーダーのマネージメント

15.14 StarloExtManagerPrinterStatus コンスタント

プリンタステータス定数。

宣言

```
typedef NS_ENUM(NSUInteger, StarloExtManagerPrinterStatus) {
    StarloExtManagerPrinterStatusInvalid = 0,
    StarloExtManagerPrinterStatusImpossible,
    StarloExtManagerPrinterStatusOnline,
    StarloExtManagerPrinterStatusOffline
};
```

定数

名称	説明
StarloExtManagerPrinterStatusInvalid	不定
StarloExtManagerPrinterStatusImpossible	プリンタ使用不能
StarloExtManagerPrinterStatusPrinterOnline	プリンタオンライン検出
StarloExtManagerPrinterStatusPrinterOffline	プリンタオフライン検出

15.15 StarloExtManagerPrinterPaperStatus コンスタント

プリンタ用紙ステータス定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerPrinterPaperStatus) {
    StarloExtManagerPrinterPaperStatusInvalid = 0,
    StarloExtManagerPrinterPaperStatusImpossible,
    StarloExtManagerPrinterPaperStatusReady,
    StarloExtManagerPrinterPaperStatusNearEmpty,
    StarloExtManagerPrinterPaperStatusEmpty
};
```

定数

名称	説明
StarloExtManagerPrinterPaperStatusInvalid	不定
StarloExtManagerPrinterPaperStatusImpossible	プリンタ使用不能
StarloExtManagerPrinterPaperStatusReady	プリンタ用紙レディ検出
StarloExtManagerPrinterPaperStatusNearEmpty	プリンタ用紙ニアエンド検出
StarloExtManagerPrinterPaperStatusEmpty	プリンタ用紙エンプティ検出

15.16 StarloExtManagerPrinterCoverStatus コンスタント

プリンタカバーステータス定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerPrinterCoverStatus) {
    StarloExtManagerPrinterCoverStatusInvalid = 0,
    StarloExtManagerPrinterCoverStatusImpossible,
    StarloExtManagerPrinterCoverStatusOpen,
    StarloExtManagerPrinterCoverStatusClose
};
```

定数

名称	説明
StarloExtManagerPrinterCoverStatusInvalid	不定
StarloExtManagerPrinterCoverStatusImpossible	プリンタ使用不能
StarloExtManagerPrinterCoverStatusOpen	プリンタカバーオープン検出
StarloExtManagerPrinterCoverStatusClose	プリンタカバークローズ検出

15.17 StarloExtManagerCashDrawerStatus コンスタント

キャッシュドロアステータス定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerCashDrawerStatus) {
    StarloExtManagerCashDrawerStatusInvalid = 0,
    StarloExtManagerCashDrawerStatusImpossible,
    StarloExtManagerCashDrawerStatusOpen,
    StarloExtManagerCashDrawerStatusClose
};
```

定数

名称	説明
StarloExtManagerCashDrawerStatusInvalid	不定
StarloExtManagerCashDrawerStatusImpossible	キャッシュドロア使用不能
StarloExtManagerCashDrawerStatusOpen	キャッシュドロアオープン検出
StarloExtManagerCashDrawerStatusClose	キャッシュドロアクローズ検出

15.18 StarloExtManagerBarcodeReaderStatus コンスタント

バーコードリーダーステータス定数。

宣言

```
typedef NS_ENUM(NSInteger, StarloExtManagerBarcodeReaderStatus) {
    StarloExtManagerBarcodeReaderStatusInvalid = 0,
    StarloExtManagerBarcodeReaderStatusImpossible,
    StarloExtManagerBarcodeReaderStatusConnect,
    StarloExtManagerBarcodeReaderStatusDisconnect
};
```

定数

名称	説明
StarloExtManagerBarcodeReaderStatusInvalid	不定
StarloExtManagerBarcodeReaderStatusImpossible	バーコードリーダー使用不能
StarloExtManagerBarcodeReaderStatusConnect	バーコードリーダー接続検出
StarloExtManagerBarcodeReaderStatusDisconnect	バーコードリーダー切断検出

16 StarloExtManagerDelegate プロトコル (StarIO_Extension.framework)

メソッド

名称	説明
didPrinterImpossible	プリンタ使用不能を通知。 * <i>optional</i>
didPrinterOnline	プリンタオンラインを通知。 * <i>optional</i>
didPrinterOffline	プリンタオフラインを通知。 * <i>optional</i>
didPrinterPaperReady	プリンタ用紙レディを通知。 * <i>optional</i>
didPrinterPaperNearEmpty	プリンタ用紙ニアエンプティを通知。 * <i>optional</i>
didPrinterPaperEmpty	プリンタ用紙エンプティを通知。 * <i>optional</i>
didPrinterCoverOpen	プリンタカバーオープンを通知。 * <i>optional</i>
didPrinterCoverClose	プリンタカバークローズを通知。 * <i>optional</i>
didCashDrawerOpen	キャッシュドロアーオープンを通知。 * <i>optional</i>
didCashDrawerClose	キャッシュドロアークローズを通知。 * <i>optional</i>
didBarcodeReaderImpossible	バーコードリーダー使用不能を通知。 * <i>optional</i>
didBarcodeReaderConnect	バーコードリーダー接続を通知。 * <i>optional</i>
didBarcodeReaderDisconnect	バーコードリーダー切断を通知。 * <i>optional</i>
didBarcodeDataReceive	バーコードデータ受信を通知。 * <i>optional</i>
didAccessoryConnectSuccess	Bluetooth アクセサリー接続成功を通知。 * <i>optional</i>
didAccessoryConnectFailure	Bluetooth アクセサリー接続失敗を通知。 * <i>optional</i>
didAccessoryDisconnect	Bluetooth アクセサリー切断を通知。 * <i>optional</i>
didStatusUpdate	ステータスアップデートを通知。 * <i>optional</i>

16.1 didPrinterImpossible メソッド

プリンタ使用不能を通知。 * *optional*

宣言

- (void)didPrinterImpossible;
- (void)didPrinterImpossible:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didPrinterImpossible:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Impossible.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。

16.2 didPrinterOnline メソッド

プリンタオンラインを通知。* optional

宣言

- (void)didPrinterOnline;
- (void)didPrinterOnline:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didPrinterOnline:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Online.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.3 didPrinterOffline メソッド

プリンタオフラインを通知。* optional

宣言

- (void)didPrinterOffline;
- (void)didPrinterOffline:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didPrinterOffline:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Offline.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.4 didPrinterPaperReady メソッド

プリンタ用紙レディを通知。* optional

宣言

- (void)didPrinterPaperReady;
- (void)didPrinterPaperReady:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didPrinterPaperReady:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Paper Ready.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.5 didPrinterPaperNearEmpty メソッド

プリンタ用紙ニアエンプティを通知。* optional

宣言

- (void)didPrinterPaperNearEmpty;
- (void)didPrinterPaperNearEmpty:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didPrinterPaperNearEmpty:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Paper Near Empty.";
    _commentLabel.textColor = [UIColor orangeColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.6 didPrinterPaperEmpty メソッド

プリンタ用紙エンプティを通知。* *optional*

宣言

- (void)didPrinterPaperEmpty;
- (void)didPrinterPaperEmpty:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didPrinterPaperEmpty:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Paper Empty.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.7 didPrinterCoverOpen メソッド

プリンタカバーオープンを通知。* *optional*

宣言

- (void)didPrinterCoverOpen;
- (void)didPrinterCoverOpen:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didPrinterCoverOpen:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Cover Open.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.8 didPrinterCoverClose メソッド

プリンタカバークローズを通知。* *optional*

宣言

- (void)didPrinterCoverClose;
- (void)didPrinterCoverClose:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didPrinterCoverClose:(StarloExtManager *)manager {
    _commentLabel.text = @"Printer Cover Close.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CombinationExtViewController.m を参照ください。

16.9 didCashDrawerOpen メソッド

キャッシュドロアーオープンを通知。* *optional*

宣言

- (void)didCashDrawerOpen;
- (void)didCashDrawerOpen:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didCashDrawerOpen:(StarloExtManager *)manager {
    _commentLabel.text = @"Cash Drawer Open.";
    // _commentLabel.textColor = [UIColor redColor];
    _commentLabel.textColor = [UIColor magentaColor];
}
```

CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。

16.10 didCashDrawerClose メソッド

キャッシュドロアークローズを通知。* optional

宣言

- (void)didCashDrawerClose;
- (void)didCashDrawerClose:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didCashDrawerClose:(StarloExtManager *)manager {
    _commentLabel.text = @"Cash Drawer Close.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

CashDrawerExtViewController.m / CombinationExtViewController.m を参照ください。

16.11 didBarcodeReaderImpossible メソッド

バーコードリーダー使用不能を通知。* optional

宣言

- (void)didBarcodeReaderImpossible;
- (void)didBarcodeReaderImpossible:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didBarcodeReaderImpossible:(StarloExtManager *)manager {
    _commentLabel.text = @"Barcode Reader Impossible.";
    _commentLabel.textColor = [UIColor redColor];
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

16.12 didBarcodeReaderConnect メソッド

バーコードリーダー接続を通知。* *optional*

宣言

- (void)didBarcodeReaderConnect;
- (void)didBarcodeReaderConnect:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didBarcodeReaderConnect:(StarloExtManager *)manager {
    _commentLabel.text = @"Barcode Reader Connect.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

16.13 didBarcodeReaderDisconnect メソッド

バーコードリーダー切断を通知。* *optional*

宣言

- (void)didBarcodeReaderDisconnect;
- (void)didBarcodeReaderDisconnect:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didBarcodeReaderDisconnect:(StarloExtManager *)manager {
    _commentLabel.text = @"Barcode Reader Disconnect.";
    _commentLabel.textColor = [UIColor redColor];
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

16.14 didBarcodeDataReceive メソッド

バーコードデータ受信を通知。* *optional*

宣言

- (void)didBarcodeDataReceive:(NSData *)data;
- (void)didBarcodeDataReceive:(StarloExtManager *)manager data:(NSData *)data;

引数

名称	説明	型
data	受信したバーコードデータ	NSData
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didBarcodeDataReceive:(StarloExtManager *)manager data:(NSData *)data {
    NSMutableString *text = [NSMutableString stringWithString:@""];

    const uint8_t *p = [data bytes];

    for (int i = 0; i < data.length; i++) {
        uint8_t ch = *(p + i);

        if (ch >= 0x20 && ch <= 0x7f) {
            [text appendFormat:@"%c", (char) ch];
        }
        else if (ch == 0x0d) {
            ...

            text = [NSMutableString stringWithString:@""];
        }
    }
}
```

BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

16.15 didAccessoryConnectSuccess メソッド

Bluetooth アクセサリー接続成功を通知。* optional

宣言

- (void)didAccessoryConnectSuccess;
- (void)didAccessoryConnectSuccess:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didAccessoryConnectSuccess:(StarloExtManager *)manager {
    _commentLabel.text = @"Accessory Connect Success.";
    _commentLabel.textColor = [UIColor blueColor];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m /
CombinationExtViewController.m を参照ください。

16.16 didAccessoryConnectFailure メソッド

Bluetooth アクセサリー接続失敗を通知。* optional

宣言

- (void)didAccessoryConnectFailure;
- (void)didAccessoryConnectFailure:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didAccessoryConnectFailure:(StarloExtManager *)manager {
    _commentLabel.text = @"Accessory Connect Failure.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m /
CombinationExtViewController.m を参照ください。

16.17 didAccessoryDisconnect メソッド

Bluetooth アクセサリー切断を通知。* *optional*

宣言

- (void)didAccessoryDisconnect;
- (void)didAccessoryDisconnect:(StarloExtManager *)manager;

引数

名称	説明	型
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
-(void)didAccessoryDisconnect:(StarloExtManager *)manager {
    _commentLabel.text = @"Accessory Disconnect.";
    _commentLabel.textColor = [UIColor redColor];
}
```

PrinterExtViewController.m / CashDrawerExtViewController.m / BarcodeReaderExtViewController.m / CombinationExtViewController.m を参照ください。

16.18 didStatusUpdate メソッド

ステータスアップデートを通知。* *optional*

宣言

- (void)didStatusUpdate:(NSString *)status;

- (void)didStatusUpdate:(StarloExtManager *)manager status:(NSString *)status;

引数

名称	説明	型
status	<p>アップデートステータス</p> <p>自動ステータス (※) の 3 バイト目～6 バイト目を 16 進数文字列化したもの (例 : "28000000")</p> <p>-ステータス通知タイミング-</p> <ul style="list-style-type: none"> ステータス変化時 ステータス未変化時、5 分毎 <p>※STAR Line Mode コマンド仕様書もしくは StarPRNT コマンド仕様書参照ください</p>	NSString *
manager	呼び出し元の StarloExtManager オブジェクト	StarloExtManager

戻り値

説明	型
-	-

Example

```
- (void)didStatusUpdate:(StarloExtManager *)manager status:(NSString *)status {
    _commentLabel.text = status;
    _commentLabel.textColor = [UIColor greenColor];
}
```

AllReceiptsExtViewController.m を参照ください。

17 SMBluetoothManagerFactory クラス (StarIO_Extension.framework)

プリンタのエミュレーションに対して適切な SMBluetoothManager オブジェクトを取得するためのクラスです。

メソッド

名称	説明
getManager	SMBluetoothManager クラスのオブジェクトを取得します。

17.1 getManager メソッド

引数で渡すエミュレーション用の SMBluetoothManager クラスのオブジェクトを取得します。

宣言

```
- (SMBluetoothManager *)getManager:(NSString *)portName
emulation:(StarloExtEmulation)emulation;
```

引数

名称	説明	型
portName	プリンタポート名	NSString *
emulation	エミュレーション	StarloExtEmulation

戻り値

説明	型
SMBluetoothManager オブジェクト	SMBluetoothManager *

Bluetooth 設定変更の手順は [SMBluetoothManager を使った Bluetooth 設定変更フロー](#) を参照ください。

18 API expanded for Swift support included in the StarIO.framework

NSError によってメソッドの終了結果を知ることのできる API が用意されました。

Swift 宣言

```
public func writePort(writeBuffer: UnsafePointer<UInt8>, _ offSet: UInt32, _ size: UInt32, _ error:
NSErrorPointer) -> UInt32
```

```
public func readPort(readBuffer: UnsafeMutablePointer<UInt8>, _ offSet: UInt32, _ size: UInt32, _ error:
NSErrorPointer) -> UInt32
```

```
public func getParsedStatus(starPrinterStatus: UnsafeMutablePointer<Void>, _ level: UInt32, _ error:
NSErrorPointer) -> UInt32
```

```
public func getFirmwareInformation(error: NSErrorPointer) -> [NSObject : AnyObject]!
```

```
public func getDipSwitchInformation(error: NSErrorPointer) -> [NSObject : AnyObject]!
```

```
public func getOnlineStatusWithError(error: NSErrorPointer) -> UInt32
```

```
public func beginCheckedBlock(starPrinterStatus: UnsafeMutablePointer<Void>, _ level: UInt32, _ error:
NSErrorPointer) -> UInt32
```

```
public func endCheckedBlock(starPrinterStatus: UnsafeMutablePointer<Void>, _ level: UInt32, _ error:
NSErrorPointer) -> UInt32
```

Objective-c 宣言

```
-(u_int32_t)writePort:(u_int8_t const *)writeBuffer :(u_int32_t)offSet :(u_int32_t)size :(NSError **)error;
```

```
-(u_int32_t)readPort:(u_int8_t *)readBuffer :(u_int32_t)offSet :(u_int32_t)size :(NSError **)error;
```

```
-(NSInteger)getParsedStatus:(void *)starPrinterStatus :(u_int32_t)level :(NSError **)error;
```

```
-(NSDictionary *)getFirmwareInformation:(NSError **)error;
```

```
-(NSDictionary *)getDipSwitchInformation:(NSError **)error;
```

```
-(NSInteger)getOnlineStatus:(NSError **)error;
```

```
-(NSInteger)beginCheckedBlock:(void *)starPrinterStatus :(u_int32_t)level :(NSError **)error;
```

```
-(NSInteger)endCheckedBlock:(void *)starPrinterStatus :(u_int32_t)level :(NSError **)error;
```

19 SMCloudServices クラス (SMCloudServices.framework)

メソッド

名称	説明
showRegistrationView	スター精密クラウドサービス登録のビューを表示します。
isRegistered	アプリケーションがスター精密クラウドサービスに登録されているか確認します。

19.1 showRegistrationView メソッド

スター精密クラウドサービス登録のビューを表示します。



宣言

```
+ (void)showRegistrationView:(void (^)(BOOL isRegistered))completion;
```

引数

名称	説明	型
completion	ビューが閉じられた際、アプリケーションの登録状況を以下の引数によって捉えることができます。 -isRegistered- • YES ... 登録されている • NO ... 登録されていない	void (^)(BOOL isRegistered)

戻り値

説明	型
-	-

Example

```
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath {
    ...

    [SMCloudServices showRegistrationView:^(BOOL isRegistration) {
        [_tableView reloadData];
    }];

    ...
}
```

AllReceiptsViewController.m を参照ください。

19.2 isRegistered メソッド

アプリケーションがスター精密クラウドサービスに登録されているか確認します。

宣言

+ (BOOL)isRegistered;

引数

名称	説明	型
-	-	-

戻り値

説明	型
リザルト <ul style="list-style-type: none"> YES ... 登録されている NO ... 登録されていない 	BOOL

Example

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    ...

    BOOL userInteractionEnabled = YES;

    if ([SMCloudServices isRegistered] == NO) {
        userInteractionEnabled = NO;
    }

    ...

    return cell;
}
```

AllReceiptsViewController.m を参照ください。

20 SMCSAllReceipts クラス (SMCloudServices.framework)

メソッド

名称	説明
uploadBitmap	ビットマップをスター精密クラウドサービスへアップロードします。
uploadData	データをスター精密クラウドサービスへアップロードします。
updateStatus	スター精密クラウドサービス上のデバイスステータス情報をアップデートします。
generateAllReceipts	AllReceipts™利用のための印刷データを生成します。

20.1 モデル : SMCSAllReceipts class メソッド

各モデルにおいて機能するメソッドは以下のとおりです。

Function		Method	mPOP	FVP10	TSP100	TSP650II	TSP700II	TSP800II	SM-S210i	SM-S220i	SM-S230i	SM-T300i	SM-T400i	BSC10	SM-S210i StarPRNT	SM-S220i StarPRNT	SM-S230i StarPRNT	SM-T300i StarPRNT	SM-T400i StarPRNT	SM-L200	SM-L300	SP700
Upload	uploadBitmap		✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
	uploadData		✓	✓	-	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
Update	updateStatus		✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-
Generate	generateAllReceipts		✓	✓	✓	✓	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-

- : 対象外

20.2 uploadBitmap メソッド

ビットマップ(UImage オブジェクト)をスター精密クラウドサービスへアップロードします。

宣言

```
+ (NSString *)uploadBitmap:(UImage *)image completion:(void (^)(NSInteger statusCode, NSError *error))completion;
```

引数

名称	説明	型
image	アップロードするビットマップ	UImage
completion	<p>アップロードが完了した際、アップロード結果を以下の引数によって捉えることができます。</p> <p>-statusCode-</p> <ul style="list-style-type: none"> HTTP ステータスコード <p>-error-</p> <ul style="list-style-type: none"> アップロードに失敗した際のエラー情報 	void (^)(NSInteger statusCode, NSError *error)

戻り値

説明	型
アップロード URL	NSString *

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation image:(UImage *)image {
    NSString *urlString = [SMCSAllReceipts uploadBitmap:image completion:nil];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    NSData *data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:YES qrCode:YES];

    [builder appendRawData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

AllReceiptsFunctions.m を参照ください。

20.3 uploadData メソッド

データ(NSData オブジェクト)をスター精密クラウドサービスへアップロードします。

宣言

```
+ (NSString *)uploadData:(NSData *)data emulation:(StarloExtEmulation)emulation
characterCode:(StarloExtCharacterCode)characterCode width:(NSInteger)width completion:(void
(^)(NSInteger statusCode, NSError *error))completion;
```

引数

名称	説明	型
data	アップロードするデータ	NSData
emulation	エミュレーション指定 <ul style="list-style-type: none"> StarloExtEmulationStarPRNT ... StarPRNT エミュレーション StarloExtEmulationStarLine ... STAR Line Mode エミュレーション StarloExtEmulationStarGraphic ... STAR Graphic Mode エミュレーション StarloExtEmulationEscPos ... ESC/POS エミュレーション StarloExtEmulationEscPosMobile ... ESC/POS Mobile エミュレーション StarloExtEmulationStarDotImpact ... STAR ドットインパクトエミュレーション 	StarloExtEmulation
characterCode	文字コード指定 <ul style="list-style-type: none"> StarloExtCharacterCodeStandard ... スタンダード StarloExtCharacterCodeJapanese ... 日本語 StarloExtCharacterCodeSimplifiedChinese ... 簡体字中国語 StarloExtCharacterCodeTraditionalChinese ... 繁体字中国語 	StarloExtCharacterCode
width	印刷領域幅 (ドット数単位)	NSInteger
completion	アップロードが完了した際、アップロード結果を以下の引数によって捉えることができます。 -statusCode- <ul style="list-style-type: none"> HTTP ステータスコード -error- <ul style="list-style-type: none"> アップロードに失敗した際のエラー情報 	void (^)(NSInteger statusCode, NSError *error)

戻り値

説明	型
アップロード URL	NSString *

Example

```
+ (NSData *)createData:(StarIoExtEmulation)emulation characterCode:(StarIoExtCharacterCode)characterCode
data:(NSData *)data width:(NSInteger)width {
    ISCBBuilder *builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:data];

    [builder endDocument];

    NSData *receiptData = [builder.commands copy];

    NSString *urlString = [SMCSAllReceipts uploadData:receiptData emulation:emulation
characterCode:characterCode width:width completion:nil];

    builder = [StarIoExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendData:data];

    NSData *allReceiptsData = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:YES
qrCode:YES];

    [builder appendRawData:allReceiptsData];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

AllReceiptsFunctions.m を参照ください。

20.4 updateStatus メソッド

スター精密クラウドサービス上のデバイスステータス情報をアップデートします。

宣言

```
+ (void)updateStatus:(NSString *)status completion:(void (^)(NSInteger statusCode, NSError *error))completion;
```

引数

名称	説明	型
status	<p>アップデートステータス</p> <p>自動ステータス (※) の3バイト目～6バイト目を16進数文字列化したもの (例: "28000000")</p> <p>※STAR Line Mode コマンド仕様書もしくはStarPRNT コマンド仕様書参照ください</p>	NSString *
completion	<p>アップデートが完了した際、アップデート結果を以下の引数によって捉えることができます。</p> <p>-statusCode-</p> <ul style="list-style-type: none"> HTTP ステータスコード <p>-error-</p> <ul style="list-style-type: none"> アップデートに失敗した際のエラー情報 	void (^)(NSInteger statusCode, NSError *error)

戻り値

説明	型
-	-

Example

```
- (void)didStatusUpdate:(StarIoExtManager *)manager status:(NSString *)status {
    [SMCSAllReceipts updateStatus:status completion:^(NSInteger statusCode, NSError *error) {
        ...
    }];
}
```

AllReceiptsExtViewController.m を参照ください。

20.5 generateAllReceipts メソッド

AllReceipts™ 利用のための印刷データを生成します。

宣言

```
+ (NSData *)generateAllReceipts:(NSString *)urlString emulation:(StarloExtEmulation)emulation
info:(BOOL)info qrCode:(BOOL)qrCode;
+ (NSData *)generateAllReceipts:(NSString *)urlString emulation:(StarloExtEmulation)emulation
info:(BOOL)info qrCode:(BOOL)qrCode width:(NSInteger)width;
```

引数

名称	説明	型
urlString	アップロード URL	NSString *
emulation	エミュレーション <ul style="list-style-type: none"> StarloExtEmulationStarPRNT ... StarPRNT エミュレーション StarloExtEmulationStarLine ... STAR Line Mode エミュレーション StarloExtEmulationStarGraphic ... STAR Graphic Mode エミュレーション StarloExtEmulationEscPos ... ESC/POS エミュレーション StarloExtEmulationEscPosMobile ... ESC/POS Mobile エミュレーション StarloExtEmulationStarDotImpact ... STAR ドットインパクトエミュレーション 	StarloExtEmulation
info	インフォメーションロゴ <ul style="list-style-type: none"> YES ... 印刷データに含める NO ... 印刷データに含めない 	BOOL
qrCode	QR コード <ul style="list-style-type: none"> YES ... 印刷データに含める NO ... 印刷データに含めない 	BOOL
width	印刷領域幅 (ドット数単位)	NSInteger

戻り値

説明	型
生成された印刷データ	NSData

Example

```
+ (NSData *)createData:(StarloExtEmulation)emulation image:(UIImage *)image {
    NSString *urlString = [SMCSAllReceipts uploadBitmap:image completion:nil];

    ISCBBuilder *builder = [StarloExt createCommandBuilder:emulation];

    [builder beginDocument];

    [builder appendBitmap:image diffusion:NO];

    NSData *data;

    if (emulation == StarloExtEmulationStarGraphic) {
        data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:info qrCode:qrCode width:
width]; // Support to centering in Star Graphic.
    }
    else {
        data = [SMCSAllReceipts generateAllReceipts:urlString emulation:emulation info:info qrCode:qrCode]; //
Non support to centering in Star Graphic.
    }

    [builder appendRawData:data];

    [builder appendCutPaper:SCBCutPaperActionPartialCutWithFeed];

    [builder endDocument];

    return [builder.commands copy];
}
```

AllReceiptsFunctions.m を参照ください。

Appendix A. AllReceipt™ 利用方法案内 (配布用)

無料の電子レシートサービス『AllReceipts™』について、詳しくは下記 URL をご参照ください。

www.allreceipts.com

下記サイトにて店舗アカウントを登録いただくだけで、その日からサービスの利用が可能です。

＜＜スター精密クラウドサービス 店舗アカウント登録サイト＞＞

www.starcloudservices.com

1) デバイスの登録

ご登録いただいたスター精密クラウドサービスのアカウントでデバイスを登録します。

デバイスの登録をすると、スター精密クラウドサービスのダッシュボードから登録したデバイスの管理が可能となります。

2) あらかじめご登録いただいた、スター精密クラウドサービスアカウントのユーザー名（Eメール）とパスワードを入力して [登録] をタップします。



3) デバイスの登録が成功すると、接続したスター精密クラウドサービスのアカウント情報が表示されます。

