

CITIZEN

Linux POS Print SDK

プログラムマニュアル

Ver. 1.10 用

シチズン・システムズ株式会社

更新履歴

年月日	バージョン	履歴
2016/09/07	1.00	新規
2016/10/19	1.01	バージョン番号のみ修正
2017/01/16	1.02	<ul style="list-style-type: none"> ・対応機種に CT-S253 を追加 ・「3.3 JIS 第3、第4水準漢字の印刷について」を追加
2017/06/08	1.03	<ul style="list-style-type: none"> ・対応機種に CT-S255 を追加
2018/12/19	1.04	<ul style="list-style-type: none"> ・対応機種に CT-S257 を追加 ・「3.2 UTF-8 エンコード文字列の印刷について」を追加
2019/01/25	1.05	<ul style="list-style-type: none"> ・対応機種に CT-S4500 を追加
2019/11/11	1.06	<ul style="list-style-type: none"> ・対応機種に PMU3300 を追加
2020/03/30	1.07	<ul style="list-style-type: none"> ・Status メソッドにニアエンプティ/ドロワー/用紙保持のステータスを追加 ・PrinterCheckEx メソッドを追加
2021/05/14	1.08	<ul style="list-style-type: none"> ・対応機種に CT-E301 と CT-E601 を追加 ・SetPrintCompletedTimeout メソッドを追加 ・PMU3300 の PageModeArea プロパティの値を修正 ・定数定義一覧を追加
2021/12/23	1.09	<ul style="list-style-type: none"> ・対応 OS に RaspberryPi OS (Bullseye) を追加 ・対応機種に CT-S4000 を追加
2022/07/29	1.10	<ul style="list-style-type: none"> ・対応 OS に RaspberryPi OS (Bullseye) 64bit を追加、Raspberry Pi OS Legacy より前を除外 (6 頁) ・対応機種に CT-S281II を追加 (7, 11 頁)
2023/11/21		<ul style="list-style-type: none"> ・対応機種に CT-S801III/CT-S851III を追加 (7, 13 頁)

ご注意

1. 本書の内容の一部、または全部を無断で転載することは、固くお断りいたします。
2. 本書の内容については、事前の予告なしに変更することがあります。
3. 本書の内容については万全を期して作成いたしましたが、万一誤り・お気付きの点がございましたら、ご連絡くださいますようお願いいたします。
4. 運用した結果の影響につきましては、3項にかかわらず責任を負いかねますのでご了承ください。
5. 上記に同意いただけない場合は、本SDKをご使用いただけません。

商標

記載されている会社名、製品名は、各社の商標または登録商標です。

CITIZEN は、シチズン時計株式会社の登録商標です。

目次

1. はじめに	6
1.1. ドキュメント対象範囲	6
1.2. 動作環境	6
1.3. プリンター設定	8
1.4. 定義方法	16
1.5. プログラム構造	17
1.6. 機能一覧	18
2. ライブラリインターフェイス	20
2.1. 戻り値	20
2.2. コンストラクタ	21
2.3. Connect メソッド	22
2.4. Disconnect メソッド	23
2.5. SetEncoding メソッド	24
2.6. PrinterCheck メソッド	25
2.7. Status メソッド	26
2.8. PrintText メソッド	28
2.9. PrintPaddingText メソッド	29
2.10. PrintBitmap メソッド	31
2.11. PrintMemoryBitmap メソッド	32
2.12. PrintNVBitmap メソッド	33
2.13. PrintBarCode メソッド	34
2.14. PrintPDF417 メソッド	35
2.15. PrintQRCode メソッド	36
2.16. CutPaper メソッド	37
2.17. UnitFeed メソッド	38
2.18. MarkFeed メソッド	39
2.19. OpenDrawer メソッド	40
2.20. TransactionPrint メソッド	41
2.21. RotatePrint メソッド	42
2.22. PageModePrint メソッド	43
2.23. ClearPrintArea メソッド	45
2.24. ClearOutput メソッド	46
2.25. PrintData メソッド	47
2.26. PrintNormal メソッド	48
2.27. SetPrintCompletedTimeout メソッド	49
2.28. GetVersionCode メソッド	50
2.29. GetVersionName メソッド	51
2.30. PrinterCheckEx メソッド	52
2.31. PageModeArea プロパティ	54
2.32. PageModePrintArea プロパティ	55
2.33. PageModePrintDirection プロパティ	56
2.34. PageModeHorizontalPosition プロパティ	57
2.35. PageModeVerticalPosition プロパティ	58
2.36. RecLineSpacing プロパティ	59
3. 補足	60
3.1. 印刷完了確認機能について	60
3.2. UTF-8 エンコード文字列の印刷について	60
3.3. JIS 第 3、第 4 水準漢字の印刷について	61

3.4. 定数定義一覧62

1. はじめに

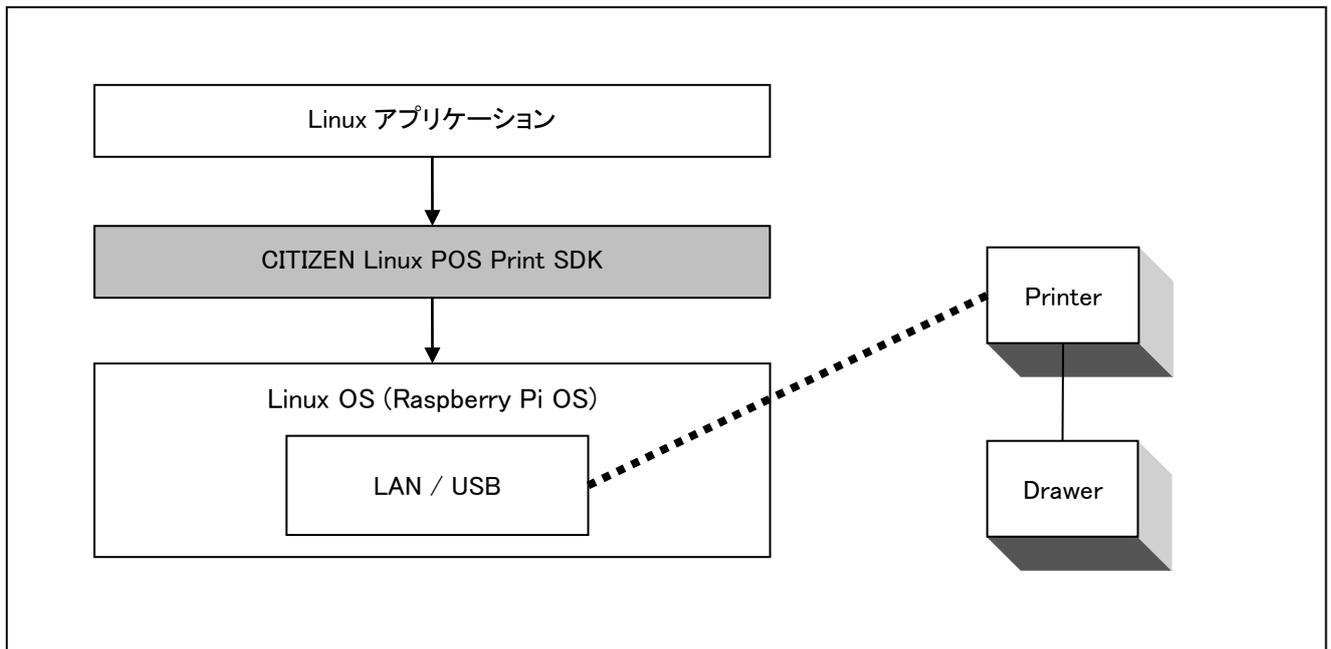
本ドキュメントは、CITIZEN Linux POS Print SDK のプログラムマニュアルです。

1.1. ドキュメント対象範囲

本ドキュメントは、CITIZEN POS プリンターを利用する Linux アプリケーションの開発者が参照することを目的としています。

1.2. 動作環境

本ライブラリは CITIZEN POS プリンターを利用する Linux アプリケーションから参照されることを想定しています。



SDK システム構成図

SDK 構成

本 SDK は、以下の2つのファイルで構成されています。

- libCSJPOSLib.so : 拡張ライブラリ
- CSJPOSLib.h : C++言語用ヘッダーファイル

対象オペレーティングシステム

本 SDK は、以下の Linux オペレーティングシステムに対応しています。

- Raspberry Pi OS (32bit, ARM Debian 11.0; Bullseye, Raspberry Pi 用 Linux)
- Raspberry Pi OS (64bit, ARM Debian 11.0; Bullseye, Raspberry Pi 用 Linux)
- Raspberry Pi OS (Legacy) (32bit, ARM Debian 10.0; Buster, Raspberry Pi 用 Linux)

対象モデル

本 SDK の対象モデルおよびそのモデルに対応するインターフェースは以下の通りです。
各モデルの機能詳細についてはプリンターの取扱説明書をご参照ください。

モデル系列	対象モデル	インターフェース	プリンター機能
CT-E301 系	CT-E301	有線 LAN, USB	標準
CT-E601 系	CT-E601	有線/無線 LAN, USB	標準
CT-S251 系	CT-S251	有線/無線 LAN, USB	標準
CT-S253 系	CT-S253	有線 LAN, USB	標準
CT-S255 系	CT-S255	有線/無線 LAN, USB	標準
CT-S257 系	CT-S257	有線/無線 LAN, USB	標準
CT-S281 系	CT-S281/281BT/281BD	USB	標準
	CT-S281-XL-M1		ブラックマーク紙対応
	CT-S281-XL		ラベル紙対応
CT-S281II 系	CT-S281II	USB	標準
	CT-S281II-L		ブラックマーク/ラベル紙対応
CT-S401 系	CT-S401	有線 LAN, USB	標準
CT-S601/651/ 801/851 系	CT-S601/651/801/851	有線/無線 LAN, USB	標準
	CT-S801/851-M		ブラックマーク紙対応
	CT-S801-L		ラベル紙対応
CT-S601II/651II/ 801II/851II 系	CT-S601II/651II/801II/ 851II	有線/無線 LAN, USB	標準
	CT-S801II/851II-M		ブラックマーク紙対応
	CT-S801II-L		ラベル紙対応
CT-S801III/851III 系	CT-S801III/851III	有線/無線 LAN, USB	標準
CT-S4000 系	CT-S4000	有線 LAN, USB	標準(表面ブラックマーク紙対応)
	CT-S4000-M		裏面ブラックマーク紙対応
	CT-S4000-L		ラベル紙対応
CT-S4500 系	CT-S4500	有線/無線 LAN, USB	標準(ブラックマーク/ラベル紙対応)
PMU3300 系	PMU3300	USB	標準(用紙出口センサー搭載)

1.3. プリンター設定

本 SDK をご使用になる際は、プリンターのメモリースイッチ設定が以下の通り設定されていることが条件となります。

CT-E301 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-S601 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5

CT-S251 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360 ドット
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)

CT-S253 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360 ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(OP932) GB18030 EUC Hangul BIG5

CT-S255 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効

MSW No.	機能	設定
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(OP932) GB18030 EUC Hangul BIG5

CT-S257 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(OP932) GB18030 EUC Hangul BIG5-HKSCS

CT-S281 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効

3-7	CBM-270 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-3	USB モード	プリンタクラス
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)

CT-S281II 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	入力バッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM-270 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)

CT-S401 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	入力バッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360 ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)

CT-S601/651/801/851 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
10-3	ACK 出力タイミング	BUSY 前

CT-S601II/651II/801II/851II 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
10-3	ACK 出力タイミング	BUSY 前

CT-S801III/851III 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッタ動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッタ復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(CP932) GB18030 EUC Hangul BIG5
10-3	ACK 出力タイミング	BUSY 前

CT-S4000 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
1-7	DSR 信号選択 (シリアル I/F 使用時)	無効
1-8	INIT 信号選択 (シリアル I/F 使用時)	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカッター復帰	L/F 有効
3-3	パラレル 31Pin	リセット
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)
10-3	ACK 出力タイミング	BUSY 前

CT-S4500 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K byte
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカット動作	有効
2-4	フル桁印字	データ待ち
3-1	オートカット復帰	L/F 有効
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-2	縦基本計算ピッチ	360 ドット
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
7-6	DMA 制御	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-4	漢字コード (*2)	SJIS(GP932) GB18030 EUC Hangul BIG5-HKSCS

PMU3300 系 メモリースイッチ設定

MSW No.	機能	設定
1-1	電源 ON 通知設定	有効
1-2	インプットバッファ	4K bytes
1-3	Busy 条件	バッファフル
1-4	受信エラー文字	?文字
1-5	CR モード	無効
2-2	オートカッター動作	有効
2-4	フル桁印字	データ待ち
3-7	CBM1000 互換モード	有効
3-8	印字中カバーオープン	自動復帰
4-8	強制パーシャル	無効
5-3	USB モード	プリンタクラス
6-1	ドライバ用動作	有効
9-1	コードページ	Katakana (*1)
9-2	国際文字	日本 (*1)
9-3	漢字	有効 (*1)
9-4	JIS/シフト JIS	シフト JIS (*1)

*1 MSW No.9-1~4 は、日本語使用時の設定です。ご使用環境に合わせて変更してください。

*2 CT-E301/601,CT-S253/255/257/601II/651II/801II/851II/801III/851III/4500 系は、漢字コードを Shift_JIS、GB18030、EUC-KR、Big5 に切り替える事ができます。ご使用の環境に合わせて変更してください。

ファームウェア

CT-S601/651/801/851 系モデルにおいて本 SDK を正常に動作させるには、プリンターのファームウェアバージョンが下記の条件である必要があります。下記プリンターよりも古いプリンターをご使用の際は、ファームウェアをバージョンアップする必要があります。

機種	ファームウェアバージョン
CT-S601	DL00-2000 以降
CT-S651	DM00-2000 以降
CT-S801	DH00-2000 以降
CT-S851	DK00-2000 以降

1.4. 定義方法

ライブラリの追加

システムライブラリ(/usr/lib あるいは /usr/local/lib)に提供ライブラリファイルを追加してください。

必須パッケージのインストール

本 SDK を使用するには、以下のパッケージが必要です。

- ・g++
- ・libgdk-pixbuf2.0
- ・libsnp40 または libsnp30

パッケージのリストをサーバーから入手するため次のコマンドを実行してください。

```
sudo apt-get update
```

パッケージをインストールするため次のコマンド実行してください(使用環境により異なります)。

Raspberry Pi OS の場合:

```
sudo apt-get install g++ libgdk-pixbuf2.0 libsnp40
```

Raspbian、Raspberry Pi OS (Legacy)の場合:

```
sudo apt-get install g++ libgdk-pixbuf2.0 libsnp30
```

USB アクセス権の付加

USB ポートを使用して印刷する場合は、アプリケーションを実行するユーザーに USB ポートへのアクセス権の付加が必要です。次の様に「/etc/group」を修正してください(pi を追加した例)。

```
lp:x:7:pi
```

ヘッダの定義

本 SDK を使用する場合は、アプリケーションプログラムの先頭でヘッダーファイルを定義してください。

```
#include "CSJPOSLib.h"
```

ビルド方法

本 SDK を使用するには g++を使用してビルドする必要があります。

アプリケーションのビルド設定に本 SDK のヘッダーファイルとライブラリを組み込んでください。

ソースファイルとヘッダーファイルを配置したディレクトリでビルドする例は次の通りです。

```
g++ [ソースファイル] -o [実行ファイル] -l CSJPOSLib -I.
```

提供サンプルプログラムのビルドは、次のコマンドを実行してください。

```
g++ Citizen_POS_Sample1.cpp -o Citizen_POS_Sample1 -l CSJPOSLib -I.
```

実行ファイルの「Citizen_POS_Sample1」が生成されます。

1.5. プログラム構造

本 SDK を使用する場合のプログラム構造は、以下の通りです。

```

// インスタンス生成
CSJPOSLib::ESCPOSPrinter *printer = new CSJPOSLib::ESCPOSPrinter();

// プリンターへ接続
int result = printer->Connect(CMP_PORT_USB, "/dev/usb/lp0");
if (CMP_SUCCESS == result)
{
    // エンコード設定
    printer->SetEncoding(932);

    // 一括処理開始設定
    printer->TransactionPrint(CMP_TP_TRANSACTION);

    // テキスト印刷
    printer->PrintText("Citizen_POS_sample1¥n¥n",
        CMP_ALIGNMENT_CENTER, CMP_FNT_DEFAULT,
        CMP_TXT_1WIDTH | CMP_TXT_1HEIGHT);
    printer->PrintText("- Sample Print 1 -¥n",
        CMP_ALIGNMENT_CENTER, CMP_FNT_DEFAULT,
        CMP_TXT_1WIDTH | CMP_TXT_2HEIGHT);
    printer->PrintText("123456789012345678901234567890¥n",
        CMP_ALIGNMENT_RIGHT, CMP_FNT_DEFAULT,
        CMP_TXT_1WIDTH | CMP_TXT_1HEIGHT);

    // QRCode 印刷
    printer->PrintQRCode("http://www.citizen-systems.co.jp/", 6,
        CMP_QRCODE_EC_LEVEL_L, CMP_ALIGNMENT_RIGHT);

    // カット位置紙送り後、パーシャルカット
    printer->CutPaper(CMP_CUT_PARTIAL_PREFEED);

    // 一括処理印刷指定
    result = printer->TransactionPrint(CMP_TP_NORMAL);

    // 切断処理
    printer->Disconnect();

    if (CMP_SUCCESS != result)
    {
        // 印刷処理エラー
        printf("Transaction Error : %d¥n", result);
    }
}
else
{
    // 接続エラー
    printf("Connect Error : %d¥n", result);
}

```

クラス定義

接続処理

印刷処理

切断処理

1.6. 機能一覧

本 SDK は以下の機能を提供します。

メソッド一覧

No	機能	詳細
1	クラス生成 (コンストラクタ)	コンストラクタです。
2	プリンター接続処理 (Connect メソッド)	プリンターと接続します。
3	プリンター切断処理 (Disconnect メソッド)	プリンターとの接続を切断します。
4	エンコード設定処理 (SetEncoding メソッド)	エンコードを設定します。
5	プリンター状態確認処理 (PrinterCheck メソッド)	ステータスチェックコマンドを送信します。
6	プリンター状態取得処理 (Status メソッド)	プリンターのステータスを取得します。
7	印字処理 (PrintText メソッド)	テキストデータを印刷します。
8	空白埋め込み印字処理 (PrintPaddingText メソッド)	空白埋め込みテキストデータを印刷します。
9	ビットマップファイル印刷処理 (PrintBitmap メソッド)	ビットマップファイル[BMP/JPG/PNG/GIF]を印刷します。
10	ビットマップデータ印刷処理 (PrintMemoryBitmap メソッド)	ビットマップデータ[BMP/JPG/PNG/GIF]を印刷します。
11	NV 登録ビットマップ印刷処理 (PrintNVBitmap メソッド)	フラッシュメモリに保存されたビットマップ画像を印刷します。
12	バーコード印刷処理 (PrintBarCode メソッド)	一次元バーコードを印刷します。
13	PDF-417 印刷処理 (PrintPDF417 メソッド)	PDF417 バーコードを印刷します。
14	QR コード印刷処理 (PrintQRCode メソッド)	QRCode バーコードを印刷します。
15	用紙カット処理 (CutPaper メソッド)	用紙をカットします。
16	ドット単位紙送り処理 (UnitFeed メソッド)	ドット単位で紙送りします。
17	マーク紙送り処理 (MarkFeed メソッド)	ラベル/ブラックマーク紙用をサポートします。
18	ドロワー開処理 (OpenDrawer メソッド)	キャッシュドロワーを開けるコマンドを送信します。
19	一括処理開始/終了処理 (TransactionPrint メソッド)	一括処理モードを開始/終了します。
20	回転印刷処理 (RotatePrint メソッド)	回転方向モード(180度)を開始/終了します。
21	ページモード開始/終了処理 (PageModePrint メソッド)	ページモードを開始/終了します。
22	ページモード印刷領域消去処理 (ClearPrintArea メソッド)	ページモード印刷領域上の印刷データを消去します。
23	出力データクリア処理 (ClearOutput メソッド)	処理中のデータおよびプリンターのバッファをクリアします。
24	データ出力処理 (PrintData メソッド)	データをそのままプリンターに送信します。
25	OPOS 形式印刷処理 (PrintNormal メソッド)	OPOS エスケープシーケンスを使用してテキストを印刷します。
26	印刷完了確認タイムアウト設定 (SetPrintCompletedTimeout メソッド)	印刷完了通知を確認するタイムアウトを設定します。

27	バージョンコード取得 (GetVersionCode メソッド)	SDK のバージョン番号を数値で取得します。
28	バージョン文字列取得 (GetVersionName メソッド)	SDK のバージョン番号を文字列で取得します。
29	接続およびプリンター状態確認処理 (PrinterCheckEx メソッド)	未接続状態で、プリンターのステータスを取得します。

プロパティ一覧

No	機能	属性	詳細
1	ページモード領域取得 (PageModeArea プロパティ)	R	使用可能なページモード領域を示します。
2	ページモード印刷領域設定/取得 (PageModePrintArea プロパティ)	R/W	ページモード印刷領域を示します。
3	ページモード印刷方向設定/取得 (PageModePrintDirection プロパティ)	R/W	ページモード印刷領域内の印刷方向を示します。
4	印刷開始水平方向オフセット値設定/取得 (PageModeHorizontalPosition プロパティ)	R/W	ページモード印刷領域内の印刷開始位置の水平方向オフセット値を示します。
5	印刷開始垂直方向オフセット値設定/取得 (PageModeVerticalPosition プロパティ)	R/W	ページモード印刷領域内の印刷開始位置の垂直方向オフセット値を示します。
6	行間設定 (RecLineSpacing プロパティ)	R/W	通常文字の印刷行の高さを示します。

2. ライブラリインターフェイス

本 SDK のインターフェイスを以下に示します。

2.1. 戻り値

以降に示すメソッドは、下記の値を返します。

戻り値	説明
CMP_SUCCESS (0)	正常終了
CMP_E_CONNECTED (1001)	プリンターへ既に接続済みです。
CMP_E_DISCONNECT (1002)	プリンターへ接続していません。
CMP_E_NOTCONNECT (1003)	プリンターへ接続できませんでした。
CMP_E_CONNECT_NOTFOUND (1004)	プリンター接続後の対応機種確認に失敗しました。
CMP_E_CONNECT_OFFLINE (1005)	プリンター接続後のプリンター状態確認に失敗しました。
CMP_E_ILLEGAL (1101)	サポートされていない処理または無効なパラメータ値です。
CMP_E_OFFLINE (1102)	プリンターがオフラインです。
CMP_E_NOEXIST (1103)	指定のファイルが存在しません。
CMP_E_FAILURE (1104)	要求された処理が実行できません。
CMP_E_TIMEOUT (1105)	所定の時間が経過してもプリンターからの応答がありません。
CMP_EPTR_COVER_OPEN (1201)	プリンターのカバーが開いています。
CMP_EPTR_REC_EMPTY (1202)	用紙切れです。
CMP_EPTR_BADFORMAT (1203)	指定されたファイルの書式がサポートされていません。
CMP_EPTR_TOOBIG (1204)	指定されたビットマップのサイズが大きすぎます。

2.2. コンストラクタ

形式

ESCPOSPrinter ()

パラメータ

ありません。

説明

ライブラリのコンストラクタです。インスタンスを生成します。

戻り値

ありません。

使用例

```
CSJPOSLib::ESCPOSPrinter *printer = new CSJPOSLib::ESCPOSPrinter();
```

2.3. Connect メソッド

形式

- 1) int Connect (int connectType, std::string addr)
- 2) int Connect (int connectType, std::string addr, int port)
- 3) int Connect (int connectType, std::string addr, int port, int timeout)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
connectType	[IN]	接続タイプ	CMP_PORT_WiFi CMP_PORT_USB
addr	[IN]	接続先の IP アドレス、 または USB デバイスファイル	WiFi : 0.0.0.0~255.255.255.255 USB : /dev/usb/lp0~
Port	[IN]	接続先ポート番号	
Timeout	[IN]	タイムアウト (msec)	

説明

このメソッドは、プリンターと接続するために使用します。プリンターの接続タイプとアドレスを指定してください。接続先ポート番号は、接続タイプに CMP_PORT_WiFi を指定した場合のみ有効です。省略された場合は、9100 番で接続します。

タイムアウトは、プリンターへの接続の最大時間(ミリ秒単位)を指定します。省略された場合は、4000 ミリ秒で接続します。

プリンターと接続した際に、プリンターのステータスと対応機種を同時に確認します。

プリンターとの通信が不要になった場合は、必ず [Disconnect メソッド](#) を実行し、プリンターとの接続を切断してください。切断しなかった場合は、次の接続がエラーとなります。

戻り値

成功時は CMP_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。

それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

エラーコード	説明
CMP_E_NOTCONNECT (1003)	プリンターへ接続できませんでした。 ①プリンターが未接続 ②プリンターの電源が入っていない ③インターフェースポートのハンドルを取得できない
CMP_E_CONNECT_NOTFOUND (1004)	プリンター接続後の対応機種確認に失敗しました。 ①対応機種でない
CMP_E_CONNECT_OFFLINE (1005)	プリンター接続後のプリンター状態確認に失敗しました。 プリンターに次のエラーが発生している可能性があります。 ①カバーが開いている ②用紙が無い ③紙ジャム等によりオートカッターエラーが発生 ④回路故障等による復帰不可能エラーが発生

使用例

```
printer->Connect( CMP_PORT_WiFi, "192.168.182.100" );
```

```
printer->Connect( CMP_PORT_USB, "/dev/usb/lp0" );
```

2.4. Disconnect メソッド

形式

```
int Disconnect ()
```

パラメータ

ありません。

説明

このメソッドは、プリンターとの接続を切断するために使用します。
印刷の終了、あるいは、何らかのエラーが発生した場合は、本メソッドを実行して接続を切断してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->Disconnect();
```

2.5. SetEncoding メソッド

形式

int SetEncoding (int codepage)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
codepage	[IN]	コードページ	437:Code437 850:Code850 Multilingual 852:Code852 EasternEurope 857:Code857 Turkey 858:Code858 860:Code860 Portugal 863:Code863 Canada-French 864:Code864 Arabic 865:Code865 Norway 866:Code866 Russia 874:Code874 Tai Code 18 932:KANA 949:EUCKR 950:BIG5 1252:Windows Code 1258: Code1258 Vietnamese 54936:GB18030 65001:UTF-8

説明

このメソッドは、プリンター送信データのエンコードを設定するために使用します。

インスタンスを生成時に OS のデフォルト文字セットに初期化します。

コードページはプリンターのメモリスイッチの設定に合わせて設定してください(「[1.5 対応機種](#)」参照)。

本 SDK は、UTF-8 でエンコードされた文字列の印刷をサポートします。詳しくは、「[3.2 UTF-8 エンコード文字列の印刷について](#)」を参照してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->SetEncoding (932);
```

```
printer->SetEncoding (54936);
```

```
printer->SetEncoding (949);
```

```
printer->SetEncoding (950);
```

```
printer.SetEncoding ( 65001 );
```

2.6. PrinterCheck メソッド

形式

```
int PrinterCheck ()
```

パラメータ

ありません。

説明

このメソッドは、プリンターのステータス取得コマンドを送信するために使用します。

本メソッドの実行結果が成功の場合は、[Status メソッド](#)でプリンターのステータスを取得できます。

本メソッドの実行結果が失敗の場合は、通信異常やデバイスの異常が発生した可能性があります。この場合、[Disconnect メソッド](#)および [Connect メソッド](#)を使用して再接続してください。

接続後に時間を空けて印刷する場合は、必ず事前に本メソッドと [Status メソッド](#)を実行してプリンターのステータスを確認してください。

ネットワーク接続の場合、長時間放置すると自動的に切断されます。接続を保持する場合は、定期的に本メソッドを実行してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
if ( CMP_SUCCESS == printer->PrinterCheck() ) {  
    // Success  
} else {  
    // Fail  
}
```

2.7. Status メソッド

形式

- 1) int Status ()
- 2) int Status (int type)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
type	[IN]	ステータス種類	CMP_STS_PRINTEROFF CMP_STS_PAPER_EMPTY CMP_STS_COVER_OPEN CMP_STS_PAPER_NEAREMPTY CMP_STS_DRAWER_LEVEL_H CMP_STS_ONPRESENTER

説明

このメソッドは、PrinterCheck メソッドによって取得されたプリンターのステータスを確認するために使用します。メソッドの実行前に、[PrinterCheck メソッド](#)を実行する必要があります。引き数が指定されなかった場合は、プリンターのエラーを示すステータス(CMP_STS_COVER_OPEN、CMP_STS_PAPER_EMPTY、CMP_STS_PRINTEROFF)の論理和を返します。ステータス種類が指定された場合は、該当するステータスを返します。ステータス種類は組み合わせて指定できます。組み合わせる場合は論理和を指定してください。

戻り値

下記のステータスコードを返します。

ステータスコード	説明
CMP_STS_NORMAL (0)	プリンターは正常です。
CMP_STS_PRINTEROFF (128)	プリンターはオフラインです。
CMP_STS_PAPER_EMPTY (32)	用紙がありません。
CMP_STS_COVER_OPEN (16)	プリンタカバーが開いています。
CMP_STS_PAPER_NEAREMPTY (4)	ニアエンプティです。(type 指定時のみ)
CMP_STS_DRAWER_LEVEL_H (2)	ドロワーキックコネクタ3番ピンの状態が H です。(type 指定時のみ)
CMP_STS_ONPRESENTER (1)	プレゼンターまたは用紙出口センサーに用紙が保持されています。(PMU3300 の type 指定時のみ)

使用例

```
int status = printer->Status();
if ( CMP_STS_NORMAL == status ) {
    // No Error
    int status2 = printer->Status(CMP_STS_PAPER_NEAREMPTY);
    if ( (CMP_STS_PAPER_NEAREMPTY & status2) > 0 ) {
        // Paper Near Empty
    }
} else {
    if ( (CMP_STS_COVER_OPEN & status) > 0 ) {
        // Cover Open
    }
    if ( (CMP_STS_PAPER_EMPTY & status) > 0 ) {
        // Paper Empty
    }
    if ( (CMP_STS_PRINTEROFF & status) > 0 ) {
```

```
        // Printer Offline
    }
}

int status3 = printer->Status(CMP_STS_DRAWER_LEVEL_H |
    CMP_STS_ONPRESENTER);
if ( (CMP_STS_DRAWER_LEVEL_H & status3) > 0 ) {
    // Status of pin 3 of drawer kick-out connector = H
}
if ( (CMP_STS_ONPRESENTER & status3) > 0 ) {
    // Paper is hold on presenter or paper exit sensor
}
}
```

2.8. PrintText メソッド

形式

int PrintText (std::string data, int alignment, int attribute, int textSize)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	テキストデータ	
alignment	[IN]	テキスト配置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え
attribute	[IN]	テキスト属性	CMP_FNT_DEFAULT: 標準フォント CMP_FNT_FONTB: フォント B CMP_FNT_FONTC: フォント C CMP_FNT_BOLD: 太字 CMP_FNT_REVERSE: 反転 CMP_FNT_UNDERLINE: 下線
textSize	[IN]	テキストサイズ	CMP_TXT_1WIDTH: 幅 1 倍 CMP_TXT_2WIDTH: 幅 2 倍 CMP_TXT_3WIDTH: 幅 3 倍 CMP_TXT_4WIDTH: 幅 4 倍 CMP_TXT_5WIDTH: 幅 5 倍 CMP_TXT_6WIDTH: 幅 6 倍 CMP_TXT_7WIDTH: 幅 7 倍 CMP_TXT_8WIDTH: 幅 8 倍 CMP_TXT_1HEIGHT: 高さ 1 倍 CMP_TXT_2HEIGHT: 高さ 2 倍 CMP_TXT_3HEIGHT: 高さ 3 倍 CMP_TXT_4HEIGHT: 高さ 4 倍 CMP_TXT_5HEIGHT: 高さ 5 倍 CMP_TXT_6HEIGHT: 高さ 6 倍 CMP_TXT_7HEIGHT: 高さ 7 倍 CMP_TXT_8HEIGHT: 高さ 8 倍

説明

このメソッドは、テキストの配置、属性、サイズを指定して、テキストを印刷するために使用します。

テキストデータの文字コードは、UTF-8 を使用してください。

テキスト属性は、フォント B、フォント C、太字、反転、下線を組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

テキストサイズは、幅と高さを組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintText( "Print text data.¥n",
    CMP_ALIGNMENT_CENTER,
    CMP_FNT_BOLD | CMP_FNT_UNDERLINE,
    CMP_TXT_2WIDTH | CMP_TXT_2HEIGHT );
```

2.9. PrintPaddingText メソッド

形式

int PrintPaddingText (std::string data, int attribute, int textSize, int length, int side)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	テキストデータ	
attribute	[IN]	テキスト属性	CMP_FNT_DEFAULT: 標準フォント CMP_FNT_FONTB: フォント B CMP_FNT_FONTC: フォント C CMP_FNT_BOLD: 太字 CMP_FNT_REVERSE: 反転 CMP_FNT_UNDERLINE: 下線
textSize	[IN]	テキストサイズ	CMP_TXT_1WIDTH: 幅 1 倍 CMP_TXT_2WIDTH: 幅 2 倍 CMP_TXT_3WIDTH: 幅 3 倍 CMP_TXT_4WIDTH: 幅 4 倍 CMP_TXT_5WIDTH: 幅 5 倍 CMP_TXT_6WIDTH: 幅 6 倍 CMP_TXT_7WIDTH: 幅 7 倍 CMP_TXT_8WIDTH: 幅 8 倍 CMP_TXT_1HEIGHT: 高さ 1 倍 CMP_TXT_2HEIGHT: 高さ 2 倍 CMP_TXT_3HEIGHT: 高さ 3 倍 CMP_TXT_4HEIGHT: 高さ 4 倍 CMP_TXT_5HEIGHT: 高さ 5 倍 CMP_TXT_6HEIGHT: 高さ 6 倍 CMP_TXT_7HEIGHT: 高さ 7 倍 CMP_TXT_8HEIGHT: 高さ 8 倍
length	[IN]	半角文字相当の長さ	1~
side	[IN]	空白埋め込みサイド	CMP_SIDE_RIGHT: テキストデータの右側 CMP_SIDE_LEFT: テキストデータの左側

説明

このメソッドは、テキストの属性、サイズ、半角文字相当の長さ、空白埋め込みサイドを指定して、空白が埋め込まれたテキストを印刷するために使用します。

テキストデータの文字コードは、UTF-8 を使用してください。テキストデータに結合文字を使用できません。

テキスト属性は、フォント B、フォント C、太字、反転、下線を組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

テキストサイズは、幅と高さを組み合わせて指定可能です。組み合わせる場合は論理和を指定してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
int nameSize = 24;      // 商品名サイズ
int priceSize = 7;     // 価格サイズ

// 1行目
printer->PrintPaddingText( "Sandwich",
    CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, nameSize, CMP_SIDE_RIGHT);
printer->PrintPaddingText( "5.00",
    CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, priceSize, CMP_SIDE_LEFT);
printer->PrintNormal("\n");

// 2行目
printer->PrintPaddingText( "Hamburg steak",
    CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, nameSize, CMP_SIDE_RIGHT );
printer->PrintPaddingText( "12.00",
    CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, priceSize, CMP_SIDE_LEFT );
printer->PrintNormal("\n");

// 3行目
printer->PrintPaddingText( "Coffee",
    CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, nameSize, CMP_SIDE_RIGHT );
printer->PrintPaddingText( "2.00",
    CMP_FNT_DEFAULT, CMP_TXT_1WIDTH, priceSize, CMP_SIDE_LEFT );
printer->PrintNormal("\n");
```

2.10. PrintBitmap メソッド

形式

- 1) int PrintBitmap (std::string fileName, int alignment)
- 2) int PrintBitmap (std::string fileName, int width, int alignment)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
fileName	[IN]	ビットマップファイル名	
width	[IN]	ビットマップ印刷幅	CMP_BM_ASIS: プリンターのドット当たり1ビットマップピクセルでビットマップを印刷します。 上記定数以外の1以上の値: ビットマップ幅(ドット単位)を指定します。
alignment	[IN]	ビットマップ配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の3以上の値: ビットマップ印刷を開始する左からの距離(ドット単位)を指定します。

説明

このメソッドは、ビットマップファイル名、印刷幅、配置位置、モードを指定して、ビットマップを印刷するために使用します。

印刷可能なビットマップ形式は、BMP/JPG/PNG/GIF です。

印刷幅が省略された場合は、CMP_BM_ASIS で印刷します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintBitmap( "samplebitmap.bmp",
    CMP_BM_ASIS,
    CMP_ALIGNMENT_CENTER);
```

2.11. PrintMemoryBitmap メソッド

形式

```
int PrintMemoryBitmap (char* data, int dataLength, int width, int alignment)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	ビットマップデータ	
dataLength	[IN]	ビットマップデータのサイズ	
width	[IN]	ビットマップ印刷幅	CMP_BM_ASIS: プリンターのドット当たり 1 ビットマップピクセルでビットマップを印刷します。 上記定数以外の 1 以上の値: ビットマップ幅(ドット単位)を指定します。
alignment	[IN]	ビットマップ配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: ビットマップ印刷を開始する左からの距離(ドット単位)を指定します。

説明

このメソッドは、ビットマップデータ、ビットマップデータのサイズ、印刷幅、配置位置、モードを指定して、ビットマップを印刷するために使用します。

印刷可能なビットマップ形式は、BMP/JPG/PNG/GIF です。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
char* data;
int dataLength;

// ビットマップデータ生成は省略

printer->PrintMemoryBitmap( data, dataLength,
    CMP_BM_ASIS,
    CMP_ALIGNMENT_CENTER);
```

2.12. PrintNVBitmap メソッド

形式

```
int PrintNVBitmap (int nvImageNumber)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
nvImageNumber	[IN]	プリンターのフラッシュメモリ内に格納されている画像番号	1-20

説明

このメソッドは、プリンターのフラッシュメモリに保存されているビットマップ画像(ロゴ)を印刷するために使用します。

このメソッドを使用するためには、事前にロゴの登録が必要です。プリンター用ユーティリティソフトウェアの「POS プリンターユーティリティ」を使用してください。

「POS プリンターユーティリティ」でロゴを登録する場合は、プリンターのモデルによって、登録モードが異なります。次の通り登録してください。

[CT-S281, PMU3300 系]

キーコード未使用モードでロゴ登録してください。

使用する画像番号に合わせて、順にロゴ登録する必要があります。

[CT-E301/601, CT-S251/253/255/257/281II/401/601/651/801/851/601II/651II/801II/851II/801III/851III/4000/4500 系]

キーコードモードでロゴ登録してください。

使用する画像番号に合わせて、キーコードを指定して登録する必要があります。

画像番号に対応するキーコードは次の通りです。

画像番号	キーコード (文字列)
1	"01"
2	"02"
3	"03"
⋮	⋮
19	"19"
20	"20"

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintNVBitmap( 1 );
```

2.13. PrintBarcode メソッド

形式

int PrintBarcode (std::string data, int symbology, int height, int width, int alignment, int textPosition)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷するバーコードデータ	
symbology	[IN]	バーコードタイプ	CMP_BCS_UPCA: UPC-A CMP_BCS_UPCE: UPC-E CMP_BCS_EAN8: EAN8(=JAN8) CMP_BCS_JAN8: JAN8(=EAN8) CMP_BCS_EAN13: EAN13(=JAN13) CMP_BCS_JAN13: JAN13(=EAN13) CMP_BCS_ITF: Interleaved 2 of 5 CMP_BCS_Codabar: Codabar CMP_BCS_Code39: コード 39 CMP_BCS_Code93: コード 93 CMP_BCS_Code128: コード 128
height	[IN]	バーコード高さ	1~255 バーコードの高さ(ドット単位)を指定します。
width	[IN]	バーコード横サイズ(倍率)	2~6 バーコードのモジュール幅(ドット単位)を指定します。
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離(ドット単位)を指定します。
textPosition	[IN]	可視コード印字位置	CMP_HRI_TEXT_NONE: 印刷しない CMP_HRI_TEXT_ABOVE: バーコードの上 CMP_HRI_TEXT_BELOW: バーコードの下

説明

このメソッドは、一次元バーコードを印刷するために使用します。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintBarcode( "123456789012",
    CMP_BCS_UPCA,
    50,
    2,
    CMP_ALIGNMENT_LEFT,
    CMP_HRI_TEXT_ABOVE );
```

2.14. PrintPDF417 メソッド

形式

int PrintPDF417 (std::string data, int digits, int steps, int moduleWidth, int stepHeight, int ECLevel, int alignment)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷データ	
digits	[IN]	桁数	0: 自動 1~30
steps	[IN]	段数	0: 自動 3~90
moduleWidth	[IN]	モジュール幅	2~8 バーコードのモジュール幅(ドット単位)を指定します。
stepHeight	[IN]	段の高さ	2~8
ECLevel	[IN]	エラー訂正レベル	CMP_PDF417_EC_LEVEL_0: レベル 0 CMP_PDF417_EC_LEVEL_1: レベル 1 CMP_PDF417_EC_LEVEL_2: レベル 2 CMP_PDF417_EC_LEVEL_3: レベル 3 CMP_PDF417_EC_LEVEL_4: レベル 4 CMP_PDF417_EC_LEVEL_5: レベル 5 CMP_PDF417_EC_LEVEL_6: レベル 6 CMP_PDF417_EC_LEVEL_7: レベル 7 CMP_PDF417_EC_LEVEL_8: レベル 8
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離 (ドット単位)を指定します。

説明

このメソッドは、PDF417 バーコードを印刷するために使用します。

各パラメータの詳細は、各プリンターのコマンドリファレンスを参照してください。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintPDF417(
    "http://www.citizen-systems.co.jp/printer/index.html",
    0, 0, 3, 3,
    CMP_PDF417_EC_LEVEL_0,
    CMP_ALIGNMENT_LEFT );
```

2.15. PrintQRCode メソッド

形式

```
int PrintQRCode (std::string data, int moduleSize, int ECLevel, int alignment)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印刷データ	
moduleSize	[IN]	モジュールサイズ	1~16(ドット単位時) バーコードのモジュールサイズ(ドット単位)を指定します。
ECLevel	[IN]	エラー修正レベル	CMP_QRCODE_EC_LEVEL_L: レベル L(7%) CMP_QRCODE_EC_LEVEL_M: レベル M(15%) CMP_QRCODE_EC_LEVEL_Q: レベル Q(25%) CMP_QRCODE_EC_LEVEL_H: レベル H(30%)
alignment	[IN]	バーコード配置位置	CMP_ALIGNMENT_LEFT: 左揃え CMP_ALIGNMENT_CENTER: 中央揃え CMP_ALIGNMENT_RIGHT: 右揃え 上記定数以外の 3 以上の値: バーコード印刷を開始する左からの距離(ドット単位)を指定します。

説明

このメソッドは、QR コードを印刷するために使用します。

各パラメータの詳細は、各プリンターのコマンドリファレンスを参照してください。

ページモード中のバーコード配置位置の CMP_ALIGNMENT_CENTER と CMP_ALIGNMENT_RIGHT の指定は無視されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintQRCode (
    "http://www.citizen-systems.co.jp/printer/index.html",
    4,
    CMP_QRCODE_EC_LEVEL_L,
    CMP_ALIGNMENT_LEFT );
```

2.16. CutPaper メソッド

形式

int CutPaper (int type)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
type	[IN]	カット種類	CMP_CUT_FULL: フルカット CMP_CUT_PARTIAL: パーシャルカット CMP_CUT_FULL_PREFEED: カット位置紙送り後、フルカット CMP_CUT_PARTIAL_PREFEED: カット位置紙送り後、パーシャルカット

説明

このメソッドは、用紙をカットするために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->CutPaper( CMP_CUT_PARTIAL_PREFEED );
```

2.17. UnitFeed メソッド

形式

```
int UnitFeed (int ufCount)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
ufCount	[IN]	紙送り数	紙送り数(ドット単位)を指定します。

説明

このメソッドは、ドット単位で紙送りするために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->UnitFeed( 200 );
```

2.18. MarkFeed メソッド

形式

int MarkFeed (int type)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
type	[IN]	ラベル用紙／ブラックマーク用紙のハンドリングの種類	CMP_MF_TO_CUTTER: 自動カッターのカット位置の上までフィードし、さらにカット CMP_MF_TO_NEXT_TOF: 次の用紙先頭印字位置までフィード

説明

このメソッドは、ラベル用紙／ブラックマーク用紙を利用するために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->MarkFeed( CMP_MF_TO_CUTTER );
```

2.19. OpenDrawer メソッド

形式

int OpenDrawer (int drawer, int pulseLen)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
drawer	[IN]	キャッシュドローワー番号	CMP_DRAWER_1: ドローワー1 CMP_DRAWER_2: ドローワー2
pulseLen	[IN]	シグナルの長さ	1~8 (× 100) msec

説明

このメソッドは、プリンターに接続されたキャッシュドローワーをオープンするために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->OpenDrawer( CMP_DRAWER_1, 1 );
```

2.20. TransactionPrint メソッド

形式

int TransactionPrint (int control)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
control	[IN]	一括処理を開始／終了を指定	CMP_TP_TRANSACTION: 一括処理の開始 CMP_TP_NORMAL: バッファデータを印刷して一括処理を終了

説明

このメソッドは、一括処理モードを開始または終了するために使用します。

control が CMP_TP_TRANSACTION ならば、一括処理モードに入ります。これ以降のメソッド呼び出しは、印刷データをバッファリングします。一括処理モードに該当するメソッドは以下の通りです。

PrintText, PrintPaddingText, PrintBitmap, PrintMemoryBitmap, PrintBarcode, PrintPDF417, PrintQRCode, CutPaper, UnitFeed, MarkFeed, OpenDrawer, RotatePrint, PageModePrint, ClearPrintArea, PrintData, PrintNormal

control が CMP_TP_NORMAL ならば、一括処理モードを抜けます。データがバッファリングされていたならば、そのデータは印刷されます。一括処理全体は、一つのメッセージとして処理されます。

[ClearOutput メソッド](#)を呼び出すことによって、一括処理モードは取り消されます。バッファされた印刷行も削除されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->TransactionPrint( CMP_TP_TRANSACTION );
printer->PrintNVBitmap( 1 );
printer->PrintBarcode( "123456789012", CMP_BCS_UPCA, 50, 2,
    CMP_ALIGNMENT_LEFT, CMP_HRI_TEXT_ABOVE );
printer->PrintText( "Line 1\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
    CMP_TXT_1WIDTH );
printer->PrintText( "Line 2\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
    CMP_TXT_1WIDTH );
printer->PrintText( "Line 3\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
    CMP_TXT_1WIDTH );
printer->PrintBarcode( "123456789012", CMP_BCS_UPCA, 50, 2,
    CMP_ALIGNMENT_LEFT, CMP_HRI_TEXT_ABOVE );
printer->PrintNVBitmap( 1 );
printer->CutPaper( CMP_CUT_PARTIAL_PREFEED );
printer->TransactionPrint( CMP_TP_NORMAL );
```

2.21. RotatePrint メソッド

形式

int RotatePrint (int rotation)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
rotation	[IN]	回転方向を指定	CMP_RP_ROTATE180: 180度回転印刷、つまり倒立印字の開始 CMP_RP_BARCODE: 回転バーコード印刷の開始、この値は上記の回転印刷開始の値との論理和 CMP_RP_BITMAP: 回転ビットマップ印刷の開始、この値は上記の回転印刷開始の値との論理和 CMP_RP_NORMAL: 回転印刷の終了

説明

このメソッドは、回転印刷モードを開始または終了するために使用します。

rotation に CMP_RP_ROTATE180 が含まれている場合は、倒立印刷モードに入ります。回転印刷モードに該当するメソッドは以下の通りです。

PrintText、PrintPaddingText、PrintNormal

更に CMP_RP_BARCODE あるいは CMP_RP_BITMAP が含まれている場合は、次のメソッドも回転して印刷されます。

PrintBarcod、PrintPDF417、PrintQRCode あるいは PrintBitmap、PrintMemoryBitmap

rotation が CMP_RP_NORMAL ならば、回転印刷モードを抜けます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->RotatePrint( CMP_RP_ROTATE180 | CMP_RP_BARCODE | CMP_RP_BITMAP );
printer->PrintBitmap( "samplebitmap.bmp", CMP_BM_ASIS,
    CMP_ALIGNMENT_CENTER );
printer->PrintBarCode( "123456789012", CMP_BCS_UPCA, 50, 2,
    CMP_ALIGNMENT_LEFT, CMP_HRI_TEXT_ABOVE );
printer->PrintText( "Line 3\r\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
    CMP_TXT_1WIDTH );
printer->PrintText( "Line 2\r\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
    CMP_TXT_1WIDTH );
printer->PrintText( "Line 1\r\n", CMP_ALIGNMENT_LEFT, CMP_FNT_DEFAULT,
    CMP_TXT_1WIDTH );
printer->CutPaper( CMP_CUT_PARTIAL_PREFEED );
printer->RotatePrint( CMP_RP_NORMAL );
```

2.22. PageModePrint メソッド

形式

int PageModePrint (int control)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
control	[IN]	ページモードコントロール	CMP_PM_PAGE_MODE: ページモードの開始 CMP_PM_PRINT_SAVE: ページモード印刷領域の印刷データを印刷し、その印刷データを保存 CMP_PM_NORMAL: ページモード印刷領域の印刷データを印刷し、その印刷データを消去し、ページモードを終了 CMP_PM_CANCEL: ページモード印刷領域の印刷データを消去し、何も印刷せずにページモードを終了

説明

このメソッドは、ページモードを開始または終了するために使用します。

control に CMP_PM_PAGE_MODE を指定した場合、ページモードを開始します。これ以降のメソッド呼び出しは、印刷データをバッファリングします。ページモードで利用可能なメソッドは以下の通りです。

PrintText, PrintPaddingText, PrintBitmap, PrintMemoryBitmap, PrintBarCode, PrintPDF417, PrintQRCode, PrintNormal

control に CMP_PM_PRINT_SAVE を指定した場合、ページモードが継続します。バッファリングした印刷データがあれば、それを保存して、印刷します。ページモード領域内に印刷データを追加して、同じレイアウトで印刷したい場合に用います。

control に CMP_PM_NORMAL を指定した場合、ページモードを終了し、通常の状態へ遷移します。バッファリングした印刷データがあれば、それを印刷します。バッファリングした印刷データは保存されません。

control に CMP_PM_CANCEL を指定した場合、ページモードを終了し、通常の状態へ遷移します。バッファリングした印刷データがあっても、それらは印刷されず、保存もされません。

PageModePrint メソッドが呼び出された時、[PageModePrintArea プロパティ](#)で定義されたページモード印刷領域上のすべての印刷データが印刷され、ページモード印刷領域の下端まで紙送りされます。

複数のページモード印刷領域が定義されていた場合は、それぞれのページモード印刷領域上のすべての印刷データが印刷され、最も下方に配置されているページモード印刷領域の下端まで紙送りされます。

ページモード領域全体を1つのデータとして送信します。

[ClearOutput メソッド](#)を呼び出すことによって、ページモードは取り消されます。バッファされた印刷行も削除されます。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

2.23. ClearPrintArea メソッド

形式

```
int ClearPrintArea ()
```

パラメータ

ありません。

説明

このメソッドは、[PageModePrintArea プロパティ](#)で定義したページモード印刷領域上の印刷データを消去するために使用します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->ClearPrintArea ();
```

2.24. ClearOutput メソッド

形式

```
int ClearOutput ()
```

パラメータ

ありません。

説明

このメソッドは、[TransactionPrint メソッド](#)や [PageModePrint メソッド](#)でバッファリングされている全ての送信データをクリアするために使用します。同時にプリンター上の印刷中データをクリアするコマンドを送信します。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->ClearOutput ();
```

2.25. PrintData メソッド

形式

```
int PrintData (char* data, int dataLength)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	送信データ	
dataLength	[IN]	送信データのサイズ	

説明

このメソッドは、データをそのままプリンターに送信するために使用します。

通常は必要ありませんが、プリンターの ESC コマンドを直接送信したい場合に使用してください。

ご使用の際は、他のメソッドに影響を与えない様に注意する必要があります。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
// ブザーを鳴らす (ブザー対応プリンターが必要です)
char data[] = { 0x1b, 0x1e };
printer->PrintData( data, sizeof(data) );
```

2.26. PrintNormal メソッド

形式

```
int PrintNormal (std::string data)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
data	[IN]	印字データ (OPOS エスケープシーケンス対応)	

説明

このメソッドは、OPOS で定義されているエスケープシーケンスを使用して印字する場合に使用します。OPOS に精通している場合に、ご使用ください。印字データの文字コードは、UTF-8 を使用してください。本 SDK で対応するエスケープシーケンスは、下記の通りです。詳細は OPOS の仕様書を参照ください。

エスケープシーケンス		注意事項
用紙カット	ESC #P	パーシャルカット(1-99)、フルカット(0,100)
フィードと用紙カット	ESC #fP	パーシャルカット(1-99)、フルカット(0,100)
ビットマップ印刷	ESC #B	1-20(プリンターに登録された Bitmap 番号を指定) Bitmap 印字後の印字位置は初期状態(左寄せ)にもどります
複数行フィード	ESC #fF	
単位フィード	ESC #uF	
バーコード印刷	ESC #R	
フォントタイプ指定	ESC #fT	
ボールド	ESC bC	
アンダーライン	ESC #uC	
カスタムカラー	ESC #rC	専用の 2 色紙を使用時のみ有効
赤色	ESC rC	専用の 2 色紙を使用時のみ有効
反転文字	ESC rVc	
縦横 1 倍角	ESC 1C	
横倍角	ESC 2C	
縦倍角	ESC 3C	
縦横倍角	ESC 4C	
横倍率	ESC #hC	1-8
縦倍率	ESC #vC	1-8
中央揃え	ESC cA	
右寄せ	ESC rA	
ノーマル	ESC N	

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
printer->PrintNormal( "%n\u001b|4C- Receipt -%n" );
```

2.27. SetPrintCompletedTimeout メソッド

形式

```
int SetPrintCompletedTimeout(int timeout)
```

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
timeout	[IN]	印刷完了通知タイムアウト (msec)	0: タイムアウトを自動調整します。 上記定数以外の 1 以上の値: タイムアウトを msec 単位で指定します。

説明

このメソッドは、印刷完了通知を確認するタイムアウトを設定するために使用します。タイムアウトは、インスタンス生成時に 0 に初期化します。0 を指定した場合は、印刷データに合わせてタイムアウトを自動的に調整されます。1 以上の値を指定した場合は、指定されたタイムアウトに固定されます。印刷完了確認処理の詳細は、「[3.1 印刷完了確認機能について](#)」を参照してください。

戻り値

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

使用例

```
// 自動調整
Printer->SetPrintCompletedTimeout( 0 );

// 90sec 固定
Printer->SetPrintCompletedTimeout( 90000 );
```

2.28. GetVersionCode メソッド

形式

```
int GetVersionCode ()
```

パラメータ

ありません。

説明

このメソッドは、SDK のバージョン番号を数値で取得するために使用します。

戻り値

SDK のバージョン番号を数値 (Ver1.00 の場合:100) で返します。

使用例

```
printer->GetVersionCode ();
```

2.29. GetVersionName メソッド

形式

`std::string GetVersionName ()`

パラメータ

ありません。

説明

このメソッドは、SDK のバージョン番号を文字列で取得するために使用します。

戻り値

SDK のバージョン番号を文字列 (Ver1.00 の場合: "1.00") で返します。

使用例

```
printer->GetVersionCode ();
```

2.30. PrinterCheckEx メソッド

形式

- 1) int PrinterCheckEx (int *status, int connectType, string addr)
- 2) int PrinterCheckEx (int *status, int connectType, string addr, int port)
- 3) int PrinterCheckEx (int *status, int connectType, string addr, int port, int timeout)

パラメータ

パラメータの意味と設定可能な値は以下の通りです。

値	[IN/OUT]	意味	設定可能範囲
status	[OUT]	ステータスコード	
connectType	[IN]	接続タイプ	CMP_PORT_SNMP
addr	[IN]	接続先の IP アドレス、	SNMP : 0.0.0.0~255.255.255.255
Port	[IN]	接続先ポート番号	
Timeout	[IN]	タイムアウト (msec)	

説明

このメソッドは、未接続の状態からプリンターのステータスを取得するために使用します。プリンターの接続タイプとアドレスを指定してください。本バージョンでは接続タイプの CMP_PORT_SNMP のみ対応しています。接続先ポート番号は、接続タイプに CMP_PORT_WiFi を指定した場合のみ有効です。本バージョンでは無視されますので、必要な場合は任意の値を設定してください。

タイムアウトは、プリンターへの接続の最大時間(ミリ秒単位)を指定します。省略された場合は、4000 ミリ秒で接続します。

接続タイプの CMP_PORT_SNMP は、Network 接続のプリンターのみ使用できます。本接続タイプを利用する事により、他の接続に関係なくステータスを取得することができます。本接続タイプを使用するには、本機能に対応したプリンターが必要です。

ステータスの取得が成功の場合は、下記のステータスコードを論理和で status パラメータにセットします。

ステータスコード	説明
CMP_STS_NORMAL (0)	プリンターは正常です。
CMP_STS_PRINTEROFF (128)	プリンターはオフラインです。
CMP_STS_PAPER_EMPTY (32)	用紙がありません。
CMP_STS_COVER_OPEN (16)	プリンタカバーが開いています。
CMP_STS_PAPER_NEAREMPTY (4)	ニアエンプティです。
CMP_STS_DRAWER_LEVEL_H (2)	ドロワーキックコネクタ3番ピンの状態が H です。

戻り値

成功時は CMP_SUCCESS(0) を返します。失敗時は下記のエラーコードの説明を確認してください。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

エラーコード	説明
CMP_E_NOTCONNECT (1003)	プリンターへ接続できませんでした。 ①プリンターが未接続 ②プリンターの電源が入っていない ③インターフェースポートのハンドルを取得できない
CMP_E_CONNECT_NOTFOUND (1004)	プリンター接続後の対応機種確認に失敗しました。 ① 応機種でない

使用例

```
int status;
int result = printer->PrinterCheckEx(&status, CMP_PORT_SNMP,
                                     "192.168.182.100");

if ( CMP_SUCCESS == result ) {
    if ( CMP_STS_NORMAL == status ) {
        // Status Normal
    } else {
        if ( (CMP_STS_COVER_OPEN & status) > 0 ) {
            // Cover Open
        }
        if ( (CMP_STS_PAPER_EMPTY & status) > 0 ) {
            // Paper Empty
        }
        if ( (CMP_STS_PRINTEROFF & status) > 0 ) {
            // Paper Offline
        }
        if ( (CMP_STS_PAPER_NEAREMPTY & status) > 0 ) {
            // Paper Near Empty
        }
        if ( (CMP_STS_DRAWER_LEVEL_H & status) > 0 ) {
            // Status of pin 3 of drawer kick-out connector = H
        }
    }
} else {
    // PrinterCheckEx Error
}
```

2.31. PageModeArea プロパティ

形式

std::string PageModeArea

属性

Read only

説明

このプロパティは、ページモード印刷領域で設定可能な最大値である、ページ領域をドット単位で保持します。このプロパティの構成は、カンマ区切りの 2 つの ASCII 数字で構成され、幅と高さの順に列挙します。ページ領域は、プリンターのハードウェア的な能力で決定されます。

[CT-S251 系]: "432,1662"

[CT-S281/281II 系]: "384,938"

[CT-E301/601 系]: "576,1662"

[CT-S253/255/257/401/601/651/801/851/601II/651II/801II/851II/801III/851III 系]:
"576,1662"

[CT-S4000/4500 系]: "832,1662"

[PMU3300 系]: "576,1662"

例えば、文字列が "384,938" であれば、ページモード領域は幅 384 単位、高さ 938 単位です。このページモード領域は、左上隅 (0,0) と右下隅 (383,937) で囲まれる長方形を示します。

このプロパティにアクセスする前に Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで設定されます。

設定方法

なし。

取得方法

std::string GetPageModeArea()

戻り値として、ページ領域を返します。

2.32. PageModePrintArea プロパティ

形式

std::string PageModePrintArea

属性

Read/Write

説明

このプロパティは、ページモード印刷領域をドット単位で保持します。ページモード印刷領域のサイズは、ページ領域より大きく取ることはできません。

このプロパティは、カンマ区切りの 4 つの ASCII 数字のみで構成され、空白文字を含めることはできません。構成は、始点の水平方向座標、始点の垂直方向座標、水平方向の幅、垂直方向の高さ、の順に列挙します。ページモード印刷領域の右端を超える文字は、次の行に印刷されます。ページモード印刷領域の下端を超える文字と画像は、印刷されません。

例えば文字列が、“50,100,200,400” である場合、ページモード印刷領域の座標は、左上隅 (50,100) と右下隅 (249,499) で囲まれる長方形を示します。

このプロパティにアクセスする前に Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで “0,0,0,0” に初期化されます。

設定方法

int SetPageModePrintArea (std::string area)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

取得方法

std::string GetPageModePrintArea ()

戻り値として、設定されているページモード印刷領域を返します。

2.33. PageModePrintDirection プロパティ

形式

int PageModePrintDirection

属性

Read/Write

説明

このプロパティは、現在編集集中のページモード印刷領域内の印刷方向を保持します。値は次の通りです。

値	意味
CMP_PD_LEFT_TO_RIGHT	ページモード印刷領域の左上隅を始点に、左から右方向へ印刷します。通常印刷方向です。
CMP_PD_BOTTOM_TO_TOP	ページモード印刷領域の左下隅を始点に、下から上方向へ印刷します。左 90 度回転印刷です。
CMP_PD_RIGHT_TO_LEFT	ページモード印刷領域の右下隅を始点に、右から左方向へ印刷します。180 度回転印刷です。
CMP_PD_TOP_TO_BOTTOM	ページモード印刷領域の右上隅を始点に、上から下方向へ印刷します。右 90 度回転印刷です。

このプロパティを変更すると、PageModeHorizontalPosition プロパティと PageModeVerticalPosition プロパティで示される印刷開始点の補正方向も変化します。

更にページモード印刷領域を切り替えることで、文字の回転方向を組み合わせたレシートや単票を印刷することができます。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで CMP_PD_LEFT_TO_RIGHT に初期化されます。

設定方法

int SetPageModePrintDirection (int direction)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

取得方法

int GetPageModePrintDirection ()

戻り値として、設定されているページモード印刷領域の印刷方向を返します。

2.34. PageModeHorizontalPosition プロパティ

形式

int PageModeHorizontalPosition

属性

Read/Write

説明

このプロパティは、ページモード印刷領域内の印刷開始位置を水平方向に補正するためのオフセット値をドット単位で保持します。

水平方向とは、PageModePrintDirection プロパティで設定した印刷方向と同じ方向を指します。

このプロパティは、現在位置ではなくて、最後に指定した水平方向のオフセット値の設定となります。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドでゼロ(0)に初期化されます。

設定方法

int SetPageModeHorizontalPosition (int position)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

取得方法

int GetPageModeHorizontalPosition ()

戻り値として、設定されているページモード印刷領域の水平方向オフセットを返します。

2.35. PageModeVerticalPosition プロパティ

形式

int PageModeVerticalPosition

属性

Read/Write

説明

このプロパティは、ページモード印刷領域内の印刷開始位置を垂直方向に補正するためのオフセット値をドット単位で保持します。

垂直方向とは、PageModePrintDirection プロパティで設定した印刷方向に対して垂直な方向です。このプロパティは、現在位置ではなくて、最後に指定した垂直方向のオフセット値の設定となります。

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドでゼロ(0)に初期化されます。

設定方法

int SetPageModeVerticalPosition (int position)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

取得方法

int GetPageModeVerticalPosition ()

戻り値として、設定されているページモード印刷領域の垂直方向オフセットを返します。

2.36. RecLineSpacing プロパティ

形式

int RecLineSpacing

属性

Read/Write

説明

このプロパティは、通常文字の印刷行の高さをドット単位で保持します。すなわち、印字行の高さと行間スペースの高さの両方を加えたものです。

現在の行間の値によっては、縦倍角文字はこの値を超える場合があります。この場合の行間スペースは無しになります。

RecLineSpacing プロパティの影響を受けるメソッドは以下の通りです。

[PrintText メソッド](#)

[PrintNormal メソッド](#)

このプロパティにアクセスする前に、Connect メソッドを完了して下さい。このプロパティは、Connect メソッドで 34 に初期化されます。

設定方法

int SetRecLineSpacing (int spacing)

パラメータに、設定したいプロパティ値を指定してください。

成功時は CMP_SUCCESS(0) を返します。それ以外のエラーコードは「[2.1 戻り値](#)」を参照してください。

取得方法

int GetRecLineSpacing ()

戻り値として、設定されている通常文字の印刷行の高さを返します。

3. 補足

本 SDK の補足事項を以下に示します。

3.1. 印刷完了確認機能について

本 SDK は、プリンターへのデータ送信後、印刷完了通知を待ち、メソッドの成功／失敗を判断しています。印刷完了確認機能は次のタイミングで処理されます。

- (1) 一括処理 (TransactionPrint メソッド) の完了時
- (2) ページモード (PageModePrint メソッド) の完了時
- (3) 一括処理またはページモードのバッファリング中以外のメソッドのデータ送信時

印刷完了確認機能は、プリンターの応答を待つため時間がかかります。複数のメソッドを連続して印刷する場合は、一括処理 (TransactionPrint メソッド) を使用する事によりスムーズな印刷が可能です。

印刷完了通知を確認するタイムアウトは、印刷データに合わせて自動的に調整されます。印刷データによっては、毎回タイムアウトエラーが発生する場合があります。その場合は、実際の印刷時間に合わせて [SetPrintCompletedTimeout メソッド](#) でタイムアウトを設定してください。

3.2. UTF-8 エンコード文字列の印刷について

本 SDK は、UTF-8 でエンコードされた文字列の印刷をサポートします。
プリンター送信データのエンコードを、[SetEncoding メソッド](#) を使用して、65001 に設定してください。

使用例

```
printer.SetEncoding( 65001 );
```

対応機種

機種	ファームウェアバージョン	制限事項
CT-S251	EM01-0304 以降	※1
CT-S401	DT00-1000 以降 DT10-1100 以降	
CT-S601II	EE00-0200 以降	※2
CT-S651II	EA00-0200 以降	
CT-S801II	ED00-0200 以降	
CT-S851II	DY00-0200 以降	※3
CT-E301	全バージョン	
CT-E601		
CT-S253		
CT-S255		
CT-S257		
CT-S4500		

注意

- ※1 プリンターは日本語、韓国語、簡体字中国語、繁体字中国語の同時印刷に未対応です。印刷可能な言語はプリンターの各仕向けに決められた1つの言語のみとなります。
- ※2 プリンターは日本語、韓国語、簡体字中国語、繁体字中国語の同時印刷に未対応です。印刷可能な言語はプリンターの MSW9-4 で指定される1つの言語のみとなります。
- ※3 プリンターは日本語、韓国語、簡体字中国語、繁体字中国語の同時印刷に対応します。複数の言語を印刷する場合、プリンターは MSW9-4 の設定に割り当てられた言語に基づいて、使用する文字を順次検索します。その際、字体、字形、書体が一様とならない場合があります。ご注意ください。

言語と書体 (CT-S253/255 系)

言語	書体
日本語フォント 韓国語フォント	ゴシック体
簡体中国語フォント 繁体中国語フォント	明朝体

言語と書体 (CT-E301/601, CT-S257/4500 系)

言語	書体
日本語フォント 韓国語フォント 簡体中国語フォント 繁体中国語フォント	ゴシック体

3.3. JIS 第3、第4水準漢字の印刷について

本 SDK は、JIS 第3、第4水準漢字の印刷をサポートしています。

JIS 第3、第4水準漢字を印刷には、プリンター送信データのエンコードを、“UTF-8”に設定する必要があります。詳しくは、「[3.2 UTF-8 エンコード文字列の印刷について](#)」を参照してください。

使用例

```
// エンコード設定
```

```
printer.SetEncoding( 65001 );
```

```
// テキスト印刷
```

```
printer.PrintText("啞焰鷗摑麴嘘俠頰軀俱繫妍鹵嚙攢繡蔣醬蟬搔瘦Yn",
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_FNT_DEFAULT,
    ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1WIDTH );
printer.PrintText("驪筭填顛鄧禱瀆吞囊剝潑酸屏并麵萊屢藹蠟汙Yn",
    ESCPOSConst.CMP_ALIGNMENT_LEFT, ESCPOSConst.CMP_FNT_DEFAULT,
    ESCPOSConst.CMP_TXT_1WIDTH | ESCPOSConst.CMP_TXT_1WIDTH );
```

3.4. 定数定義一覧

No	項目	定数名	型	値	説明
1	処理結果	CMP_SUCCESS	int	0	正常終了
		CMP_E_CONNECTED	int	1001	接続済み
		CMP_E_DISCONNECT	int	1002	未接続
		CMP_E_NOTCONNECT	int	1003	接続失敗
		CMP_E_CONNECT_NOTFOUND	int	1004	未対応機種
		CMP_E_CONNECT_OFFLINE	int	1005	プリンター状態異常
		CMP_E_ILLEGAL	int	1101	未対応処理または無効パラメータ
		CMP_E_OFFLINE	int	1102	オフライン
		CMP_E_NOEXIST	int	1103	ファイルが存在しない
		CMP_E_FAILURE	int	1104	処理異常
		CMP_E_TIMEOUT	int	1105	書き込みタイムアウト
		CMP_EPTR_COVER_OPEN	int	1201	プリンターのカバーオープン
		CMP_EPTR_REC_EMPTY	int	1202	用紙切れ
		CMP_EPTR_BADFORMAT	int	1203	画像フォーマット異常
CMP_EPTR_CMP_EPTR_TOOBIG	int	1204	画像サイズが大きすぎる		
2	接続インターフェース	CMP_PORT_WIFI	int	0	ネットワーク接続
		CMP_PORT_USB	int	3	USB 接続
		CMP_PORT_SNMP	int	6	SNMP 接続
3	ステータス	CMP_STS_NORMAL	int	0	ステータス正常
		CMP_STS_DRAWER_LEVEL_H	int	2	ドロワーキックコネクタ3番ピン状態 H
		CMP_STS_PAPER_NEAREMPTY	int	4	ニアエンプティ
		CMP_STS_COVER_OPEN	int	16	カバーオープン
		CMP_STS_PAPER_EMPTY	int	32	用紙切れ
		CMP_STS_PRINTEROFF	int	128	オフライン
4	配置	CMP_ALIGNMENT_LEFT	int	0	左揃え
		CMP_ALIGNMENT_CENTER	Int	1	中揃え
		CMP_ALIGNMENT_RIGHT	int	2	右揃え
5	文字属性	CMP_FNT_DEFAULT	int	0	標準フォント
		CMP_FNT_FONTB	int	1	フォント B
		CMP_FNT_FONTC	int	2	フォント C
		CMP_FNT_BOLD	int	8	太字
		CMP_FNT_REVERSE	int	64	反転
		CMP_FNT_UNDERLINE	int	128	下線
6	文字サイズ	CMP_TXT_1WIDTH	int	0	幅 1 倍
		CMP_TXT_2WIDTH	int	16	幅 2 倍
		CMP_TXT_3WIDTH	int	32	幅 3 倍
		CMP_TXT_4WIDTH	int	48	幅 4 倍
		CMP_TXT_5WIDTH	int	64	幅 5 倍
		CMP_TXT_6WIDTH	int	80	幅 6 倍
		CMP_TXT_7WIDTH	int	96	幅 7 倍
		CMP_TXT_8WIDTH	int	112	幅 8 倍
		CMP_TXT_1HEIGHT	int	0	高さ 1 倍
		CMP_TXT_2HEIGHT	int	1	高さ 2 倍
		CMP_TXT_3HEIGHT	int	2	高さ 3 倍
		CMP_TXT_4HEIGHT	int	3	高さ 4 倍
		CMP_TXT_5HEIGHT	int	4	高さ 5 倍

		CMP_TXT_6HEIGHT	int	5	高さ 6 倍
		CMP_TXT_7HEIGHT	int	6	高さ 7 倍
		CMP_TXT_8HEIGHT	int	7	高さ 8 倍
7	サイド	CMP_SIDE_RIGHT	int	0	右側
		CMP_SIDE_LEFT	int	1	左側
8	画像幅	CMP_BM_ASIS	int	-11	1ドット当たり1ピクセル
9	バーコード種類	CMP_BCS_UPCA	int	101	UPC-A
		CMP_BCS_UPCE	int	102	UPC-E
		CMP_BCS_EAN8	int	103	EAN8
		CMP_BCS_EAN13	int	104	EAN13
		CMP_BCS_JAN8	int	105	JAN8
		CMP_BCS_JAN13	int	106	JAN13
		CMP_BCS_ITF	int	107	Interleaved 2 of 5
		CMP_BCS_Codabar	int	108	Codabar
		CMP_BCS_Code39	int	109	Code39
		CMP_BCS_Code93	int	110	Code93
		CMP_BCS_Code128	int	111	Code128
		CMP_BCS_GS1DATABAR	int	131	GS1 DataBar Omnidirectional
		CMP_BCS_GS1DATABAR_E	int	132	GS1 DataBar Expanded
		CMP_BCS_GS1DATABAR_S	int	133	GS1 DataBar Stacked
		CMP_BCS_GS1DATABAR_E_S	int	134	GS1 DataBar Expanded Stacked
10	可視コード	CMP_HRI_TEXT_NONE	int	0	なし
		CMP_HRI_TEXT_ABOVE	int	1	バーコードの上
		CMP_HRI_TEXT_BELOW	int	2	バーコードの下
11	エラー修正レベル(PDF417)	CMP_PDF417_EC_LEVEL_0	int	48	レベル 0
		CMP_PDF417_EC_LEVEL_1	int	49	レベル 1
		CMP_PDF417_EC_LEVEL_2	int	50	レベル 2
		CMP_PDF417_EC_LEVEL_3	int	51	レベル 3
		CMP_PDF417_EC_LEVEL_4	int	52	レベル 4
		CMP_PDF417_EC_LEVEL_5	int	53	レベル 5
		CMP_PDF417_EC_LEVEL_6	int	54	レベル 6
		CMP_PDF417_EC_LEVEL_7	int	55	レベル 7
		CMP_PDF417_EC_LEVEL_8	int	56	レベル 8
12	エラー修正レベル(QR Code)	CMP_QRCODE_EC_LEVEL_L	int	48	レベル L(7%)
		CMP_QRCODE_EC_LEVEL_M	int	49	レベル M(15%)
		CMP_QRCODE_EC_LEVEL_Q	int	50	レベル Q(25%)
		CMP_QRCODE_EC_LEVEL_H	int	51	レベル H(30%)
13	カット種類	CMP_CUT_FULL	int	-1	フルカット
		CMP_CUT_PARTIAL	int	-2	パーシャルカット
		CMP_CUT_FULL_PREFEED	int	-3	カット位置送り後フルカット
		CMP_CUT_PARTIAL_PREFEED	int	-4	カット位置送り後パーシャルカット
14	マークフィード種類	CMP_MF_TO_CUTTER	int	2	フィードしてカット
		CMP_MF_TO_NEXT_TOF	int	8	次の用紙までフィード
15	ドロワー番号	CMP_DRAWER_1	int	1	ドロワー1
		CMP_DRAWER_2	int	2	ドロワー2
16	一括処理コントロール	CMP_TP_TRANSACTION	int	11	一括処理の開始
		CMP_TP_NORMAL	int	12	一括処理の印刷
17	回転方向コントロール	CMP_RT_NORMAL	int	0x0001	回転印刷の終了
		CMP_RT_ROTATE180	int	0x0103	倒立印刷の開始

		CMP_RP_BARCODE	int	0x1000	バーコード回転の開始
		CMP_RP_BITMAP	int	0x2000	画像回転の開始
18	ページモード コントロール	CMP_PM_PAGE_MODE	int	1	ページモードの開始
		CMP_PM_PRINT_SAVE	int	2	印刷とデータ保持
		CMP_PM_NORMAL	int	3	印刷とページモード終了
		CMP_PM_CANCEL	int	4	ページモードキャンセル
19	ページモード 印刷方向	CMP_PD_LEFT_TO_RIGHT	int	1	通常印刷方向
		CMP_PD_BOTTOM_TO_TOP	int	2	左 90 度回転印刷
		CMP_PD_RIGHT_TO_LEFT	int	3	180 度回転印刷
		CMP_PD_TOP_TO_BOTTOM	int	4	右 90 度回転印刷

CITIZEN Linux POS Print SDK プログラムマニュアル

2023/11/21 Ver.1.10 用

シチズン・システムズ株式会社

<https://csj.citizen.co.jp/>