SII Print Class Library for Android™
アプリケーションプログラマーズガイド

Rev.09

[対象製品]
SLP720RTシリーズ
SLP721RTシリーズ

セイコーインスツル株式会社

Rev.01	2022年	3月
Rev.02	2022年	10月
Rev.03	2022年	12月
Rev.04	2023年	4月
Rev.05	2024年	3月
Rev.06	2024年	11月
Rev.07	2025年	1月
Rev.08	2026年	2月
Rev.09	2026年	6月

©セイコーインスツル株式会社　2022-2026
無断転載を禁じます。

Microsoft®、Windows®、および Microsoft Store は、米国およびその他の国における Microsoft Corporation の商標または登録商標です。
Android™は、Google LLC.の商標です。
Google Play および Google Play ロゴは、Google LLC の商標です。
App Store℠ は Apple Inc. のサービスマークです。
Bluetooth®ワードマークは、Bluetooth SIG, Inc. が所有する登録商標です。
Oracle および Java®は、Oracle Corporation およびその関連会社の米国およびその他の国における登録商標です。
その他の会社名および製品名は、各社の登録商標または商標です。

本書の内容は、断りなく変更することがあります。
本書及び本書に記載された製品の利用によって発生した
損害及びその回復に要する費用に対し、当社は一切の責任を負いかねます。
本書を第三者に無断で頒布することを禁じます。

はじめに

本書は、セイコーインスツル株式会社（以降：SII）が提供する「SII Print Class Library for Android™」（以降：SIIプリントクラスライブラリ）について説明します。

対象プリンター

SIIプリントクラスライブラリでサポートするプリンターを下記に記載します。

プリンター	インターフェイス
SLP720RTシリーズ	USB
	TCP/IP
SLP721RTシリーズ	Bluetooth
	USB
	TCP/IP

用語一覧

本書で使用する用語について下記の通りに定義します。

用語	内容
技術説明書	下記の技術説明書 ・SLP720RTシリーズ サーマルプリンタ 技術説明書 ・SLP721RTシリーズ サーマルプリンタ 技術説明書
取扱説明書	下記の取扱説明書 ・SLP720RTシリーズ サーマルプリンタ 取扱説明書 ・SLP721RTシリーズ サーマルプリンタ 取扱説明書
プリンターコマンド	「技術説明書」に記載されている、プリンターを制御するための命令

サポート用紙と本書での名称

SIIプリントクラスライブラリがサポートする用紙と本書での名称を下記に記載します。

全体	種類別	機能別	略称	サポート
用紙	レシート	マークなし用紙	レシート	✔
	ライナレスラベル紙	マークなしライナレスラベル紙	ラベル紙	✔
		マーク付きライナレスラベル紙	マーク紙	✔
	SLPラベル紙	SLPラベル紙	ラベル紙	✔

目次

1章　製品概要　1-1

1.1　SIIプリントクラスライブラリの提供機能 1-1
1.2　SIIプリントクラスライブラリの概要 1-1
1.2.1　SIIプリントクラスライブラリの構成 1-1
1.2.2　ライブラリの提供機能 1-2

2章　製品仕様　2-1

2.1　動作環境 2-1
2.2　プリンターの設定 2-2
2.3　注意事項 2-2

3章　ライブラリの利用方法　3-1

3.1　Androidアプリケーションの開発環境 3-1
3.2　提供ファイル 3-2
3.3　Android Studioのプロジェクトへの組み込み 3-3
3.4　作成したAndroidアプリケーションのAndroidデバイスでの利用 3-5
3.5　注意事項 3-5

4章　ライブラリの機能　4-1

4.1　ラベル印字機能 4-1
4.1.1　ラベルファイルの構造 4-2
(1)　オブジェクトの種類とライブラリでのサポート 4-2
(2)　ラベルファイルをライブラリで印字する際の注意点 4-3
①　全てのオブジェクト 4-3
②　テキストオブジェクト 4-3
③　イメージオブジェクト 4-4
④　バーコードオブジェクト 4-4
⑤　図オブジェクト 4-5
⑥　連絡帳オブジェクト 4-5
⑦　日付と時刻オブジェクト 4-5
⑧　グループオブジェクト 4-5
4.1.2　ラベルファイルの利用手順 4-6
(1)　ラベルファイルをそのままライブラリから印字する場合 4-6
(2)　ラベルファイルのオブジェクトデータを置換して印字する場合 4-6
4.2　ログファイル出力機能 4-7
4.2.1　ログ出力の設定方法 4-7
4.2.2　ログ出力の設定内容 4-7
4.2.3　ログファイル 4-7
4.3　APIリファレンス 4-8
4.3.1　PrinterManagerクラス 4-9

(1) メソッド一覧 4-9
(2) 定数一覧 4-10
① プリンターモデル 4-10
② 応答種別 4-11
③ 国際文字セット 4-11
④ コードページ 4-11
⑤ ポート種別 4-12
⑥ ラベルファイル印刷時マークあり・なし用紙選択 4-12
⑦ バーコード、PDF417専用 4-13
(3) 列挙型定数一覧 4-13
① 強調印字（CharacterBold） 4-13
② アンダーライン（CharacterUnderline） 4-13
③ 白黒反転印字（CharacterReverse） 4-13
④ 倒立印字（CharacterInversion） 4-14
⑤ 文字フォント（CharacterFont） 4-14
⑥ 文字倍率（CharacterScale） 4-14
⑦ 位置揃え（PrintAlignment） 4-15
⑧ 保留データの出力指定（OutputPendingData） 4-15
⑨ バーコードシンボル（BarcodeSymbol） 4-15
⑩ モジュールサイズ（ModuleSize） 4-16
⑪ HRI文字印字位置（HriPosition） 4-18
⑫ NW比（NwRatio） 4-18
⑬ エラー訂正レベル（ErrorCorrection） 4-19
⑭ PDF417シンボル（Pdf417Symbol） 4-19
⑮ QRコードモデル（QrModel） 4-19
⑯ Data Matrixモジュール（DataMatrixModule） 4-20
⑰ MaxiCodeモード（MaxiCodeMode） 4-21
⑱ カット方法（CuttingMethod） 4-21
⑲ 頭出し位置（FeedPosition） 4-21
⑳ ドロワ番号（DrawerNum） 4-21
㉑ パルス幅（PulseWidth） 4-22
㉒ ブザーパターン（BuzzerPattern） 4-22
㉓ ディザリング（Dithering） 4-22
㉔ イメージの回転方向（Rotate） 4-22
㉕ イメージの拡大縮小（ImageScale） 4-23
㉖ 一括処理選択（TransactionFunction） 4-23
(4) メソッド詳細 4-24
PrinterManager コンストラクタ 4-24
connect プリンターとの通信開始（Bluetooth） 4-24
connect プリンターとの通信開始（USB） 4-25
connect プリンターとの通信開始（TCP/IP） 4-25
disconnect プリンターとの通信切断 4-26
setBarcodeScannerListener バーコードスキャナーのコールバック開始・終了 4-26
sendText テキストデータの送信 4-26
sendTextEx 書式指定テキストデータの送信 4-27
printBarcode バーコードの印字 4-29

printPDF417　PDF417の印字.....4-33
printQRcode　QRコードの印字.....4-34
printDataMatrix　Data Matrixの印字.....4-35
printMaxiCode　MaxiCodeの印字.....4-36
printGS1DataBarStacked
　GS1 Databar Stackedの印字.....4-36
printGS1DataBarStackedOmnidirectional
　GS1 Databar Stacked Omni-directionalの印字.4-37
printGS1DataBarExpandedStacked
　GS1 Databar Expanded Stackedの印字.....4-38
printAztecCode　Aztec Codeの印字.....4-38
cutPaper　用紙のカット.....4-39
feedPosition　用紙の頭出し.....4-39
openDrawer　キャッシュドロワを開く.....4-39
buzzer　ブザーの鳴動.....4-40
externalBuzzer　外部ブザーの鳴動.....4-40
sendBinary　バイナリデータの送信.....4-40
sendDataFile　指定ファイルの送信.....4-41
sendImage　イメージの送信.....4-42
printPDF　PDFページの印字.....4-43
getStatus　プリンターステータスの取得.....4-45
setCallbackFunctionListener
　プリンターステータス変化のコールバック
　開始・終了.....4-46
abort　プリンターのデータ待ち状態解除.....4-46
registerLogo　ロゴの登録.....4-47
printLogo　ロゴの印字.....4-48
unregisterLogo　ロゴの登録削除.....4-48
registerStyleSheet　スタイルシートの登録.....4-48
unregisterStyleSheet
　スタイルシートの登録削除.....4-48
resetPrinter　プリンターのリセット.....4-49
getPrinterResponse　プリンターからの各種応答取得.....4-49
startDiscoveryPrinter
　プリンターの探索開始（Bluetooth）.....4-51
startDiscoveryPrinter
　プリンターの探索開始（USB）.....4-51
startDiscoveryPrinter
　プリンターの探索開始（TCP/IP）.....4-52
cancelDiscoveryPrinter
　プリンターの探索中断.....4-52
getFoundPrinter　発見されたプリンター情報リストの取得.....4-53
getSendTimeout　送信タイムアウト時間の取得.....4-53
setSendTimeout　送信タイムアウト時間の設定.....4-53
getReceiveTimeout　受信タイムアウト時間の取得.....4-53
setReceiveTimeout　受信タイムアウト時間の設定.....4-54
getInternationalCharacter
　国際文字セットの取得.....4-54

setInternationalCharacter
国際文字セットの設定.....4-54
getCodePage コードページの取得.....4-55
setCodePage コードページの設定.....4-55
getPrinterModel プリンターモデルの取得.....4-55
getPortType 接続中ポート種別の取得.....4-55
isConnect プリンターとの通信状態の確認.....4-56
getSocketKeepingTime
ソケット維持時間の取得.....4-56
setSocketKeepingTime
ソケット維持時間の設定.....4-56
getPrintSmartLabelMode
ラベルファイル印刷時の用紙の取得.....4-56
setPrintSmartLabelMode
ラベルファイル印刷時の用紙の設定.....4-57
getVersion SDKバージョンの取得.....4-57
printSmartLabelImageData
ラベルの印字.....4-57
controlTransaction 一括処理の開始・終了.....4-58
4.3.2 PrinterEventクラス.....4-60
(1) メソッド一覧.....4-60
(2) 終了イベント定数.....4-60
(3) メソッド詳細.....4-60
getEventType 終了イベントの取得.....4-60
4.3.3 PrinterListenerインターフェイス.....4-61
(1) メソッド一覧.....4-61
(2) メソッド詳細.....4-61
finishEvent プリンター探索の終了イベント.....4-59
4.3.4 PrinterInfoクラス.....4-62
(1) メソッド一覧.....4-62
(2) メソッド詳細.....4-62
getPrinterModelName プリンターモデル名の取得.....4-62
getBluetoothAddress Bluetoothアドレスの取得.....4-62
getMacAddress MACアドレスの取得.....4-62
getIpAddress IPアドレスの取得.....4-63
getIsBonded ペアリング状態の取得.....4-63
getDevicePath デバイスパスの取得.....4-63
4.3.5 PrinterIdentifierクラス.....4-64
(1) メソッド一覧.....4-64
(2) メソッド詳細.....4-64
getPrinterModelID 機種IDの取得.....4-64
getTypeID タイプIDの取得.....4-64
getRomVersionID ROMバージョンの取得.....4-64
getMainFirmwareVersion
ファームウェアバージョン(main)の取得.....4-65
getManufacturer メーカ名の取得.....4-65
getModelName モデル名の取得.....4-65
getSerialNumber シリアル番号の取得.....4-65

getBootFirmwareVersion
ファームウェアバージョン(boot)の取得.....4-65
getBootFirmwareChecksum
ファームウェアチェックサム(boot)の取得.....4-66
getMainFirmwareChecksum
ファームウェアチェックサム(main)の取得.....4-66
getFirmwareChecksum ファームウェアチェックサム(main + boot)の取得..
.....4-66
4.3.6 PrinterExceptionクラス.....4-67
(1) メソッド一覧.....4-67
(2) 定数一覧.....4-67
① エラーコード.....4-67
(3) メソッド詳細.....4-68
PrinterException コンストラクタ.....4-68
getErrorCode エラーコードの取得.....4-68
4.3.7 CallbackFunctionListenerインターフェイス.....4-69
(1) メソッド一覧.....4-69
(2) メソッド詳細.....4-69
onStatusChanged プリンターステータスの変化イベント.....4-69
4.3.8 BarcodeScannerListenerインターフェイス.....4-70
4.3.9 SmartLabelManagerクラス.....4-71
(1) メソッド一覧.....4-71
(2) メソッド詳細.....4-72
SmartLabelManager コンストラクタ.....4-72
selectSmartLabelFile
ラベルファイルの指定.....4-72
replaceSmartLabelTextData
ラベルのテキストデータの置換.....4-73
replaceSmartLabelImageData
ラベルのイメージデータの置換.....4-74
replaceSmartLabelBarcodeData
ラベルのバーコードデータの置換.....4-75

5章 サンプルプログラム 5-1

5.1 画面構成.....5-1
5.1.1 メイン画面.....5-1
5.1.2 [Settings]画面.....5-3
5.2 注意事項.....5-3

付録A 文字セット A-1

A.1 コードページ表(文字コードテーブル).....A-1
A.2 国際文字セット.....A-11

付属B　バーコードサイズ一覧　B-1

B.1　バーコードサイズ一覧 B-1
B.1.1 printBarcode B-1
B.1.2 printPDF417 ..... B-7
B.1.3 printQRCode B-8
B.1.4 printDataMatrix B-9
B.1.5 printMaxicode B-11
B.1.6 printGS1DataBarStacked B-12
B.1.7 printGS1DataBarStackedOmnidirectional B-13
B.1.8 printGS1DataBarExpandedStacked B-14

1章　製品概要

本章では、SIIプリントクラスライブラリの製品概要について説明します。

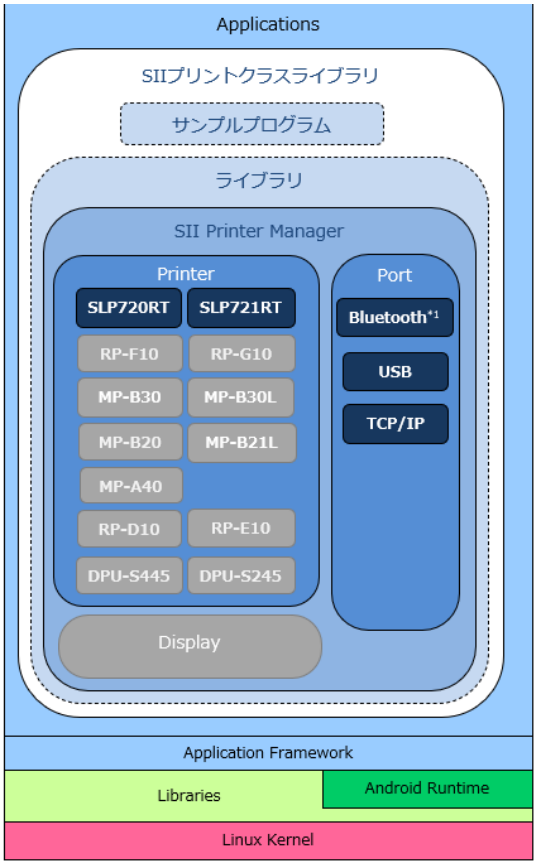
1.1　SIIプリントクラスライブラリの提供機能

ライブラリとサンプルプログラムを含むSIIプリントクラスライブラリは、AndroidアプリケーションにSII製プリンターSLP720RT/SLP721RTシリーズ（以降：プリンター）を利用するための機能を提供します。
また、SIIプリントクラスライブラリは、ライブラリのサンプルプログラムをAndroid Studioのプロジェクトで提供します。

1.2　SIIプリントクラスライブラリの概要

1.2.1　SIIプリントクラスライブラリの構成

SIIプリントクラスライブラリに含まれるライブラリとサンプルプログラムの範囲を、下図に破線で示します。



*1：SLP721RTのみサポート

1.2.2 ライブラリの提供機能

ライブラリを使用することにより、Androidアプリケーションは、Androidデバイスの通信ポート（Bluetooth、USB、またはTCP/IP）を通じて、プリンターに印字データやプリンターコマンドを容易に送信することができます。また、プリンターステータスを取得することができます。

ライブラリでは、下記の機能を提供します。

- プリンターとの接続及び切断
- プリンターへのデータ送信（印字データやプリンターコマンド*1）
- バーコードの印字、2次元コードの印字
- プリンターへのデータファイル送信（印字データやプリンターコマンド*1）
- プリンターステータスの取得
- プリンターのデータ待ち状態解除
- プリンターからの各種応答取得
- 印字命令の一括登録
- プリンターステータスのコールバック関数登録
- Bluetooth、またはTCP/IPによるプリンターの探索
- ドロワの動作制御
- ブザーの鳴動制御
- ラベルファイルの印字
- ラベルファイルのオブジェクトデータの置き換え
- ログファイルの出力

*1: プリンターからの応答を読み取るコマンドには対応しておりません。
プリンターから応答を読み取るには、**getStatus**または**getPrinterResponse**を使用してください。

（注意） SLP720RT/SLP721RTは、ページモード、ディスプレイ及びバーコードスキャナーのAPIをサポートしていません。

2章　製品仕様

本章では、ライブラリの製品仕様について説明します。

2.1　動作環境

ライブラリの動作環境を下記に示します。

<table>
<tr><td rowspan="2">プリンター</td><td colspan="2">モデル</td><td>SLP721RT</td><td colspan="2">SLP720RT/SLP721RT</td></tr>
<tr><td colspan="2">通信インターフェース</td><td>Bluetooth</td><td>USB</td><td>TCP/IP</td></tr>
<tr><td rowspan="15">Android
デバイス</td><td colspan="2">通信ポート</td><td>Bluetooth*1</td><td>USB*2</td><td>TCP/IP*3</td></tr>
<tr><td rowspan="14">OS</td><td>Android 7.0 (API 24)</td><td rowspan="14">対応</td><td rowspan="14">対応</td><td rowspan="14">対応</td></tr>
<tr><td>Android 7.1 (API 25)</td></tr>
<tr><td>Android 8.0 (API 26)</td></tr>
<tr><td>Android 8.1 (API 27)</td></tr>
<tr><td>Android 9.0 (API 28)</td></tr>
<tr><td>Android 10.0 (API 29)</td></tr>
<tr><td>Android 11.0 (API 30)</td></tr>
<tr><td>Android 12.0 (API 31)</td></tr>
<tr><td>Android 12.1 (API 32)</td></tr>
<tr><td>Android 13.0 (API 33)</td></tr>
<tr><td>Android 14.0 (API 34)</td></tr>
<tr><td>Android 15.0 (API 35)</td></tr>
<tr><td>Android 16.0 (API 36)</td></tr>
<tr><td colspan="3">サポート言語</td><td colspan="3">日本語/英語</td></tr>
</table>

*1: SPP（Serial Port Profile）によりBluetooth接続が確立されている必要があります。
*2: Androidデバイスが USBホスト機能をサポートしている必要があります。
*3: Androidデバイスが接続している無線LANアクセスポイントとプリンターが同じネットワークに接続されている必要があります。

2.2 プリンターの設定

ライブラリの利用時には、プリンターのメモリスイッチを下記の[値]に設定してください。

プリンターのメモリスイッチは、Google PlayのAndroidアプリ「SII Printer Utility」で設定できます。

メモリスイッチの詳細、及び工場出荷時の設定については、「取扱説明書」を参照してください。

MS	機能	値
1-2	テイクンモード選択 (Taken Mode)	0 : 有効*1 1 : 無効*2
1-3	マークモード選択 (Mark Mode)	0 : 有効*3 1 : 無効*4
4-6	用紙自動検出選択 (Paper Auto Detection)	0 : 有効*3 1 : 無効*3*4
5-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
5-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
5-3	エラー時データ破棄選択 (Error Through)	0 : 有効
5-4	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効
7	感熱紙選択 (Thermal Paper)	00B : レシート 01B : ライナレスラベル紙 10B : SLPラベル紙
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
17-3	用紙カット後の逆方向紙送り設定 (Backfeed After Cut)	0 : 有効 1 : 無効*5

*1: ライナレスラベル紙で連続印字する場合は、本設定を「有効」にしてください。
テイクンセンサのステータス応答は、本設定が「有効」の場合に応答します。

*2: レシート、SLPラベル紙で連続印字する場合は、本設定を「無効」にしてください。

*3: **feedPosition**を使用する場合は、下記のどちらかの設定が必要です。
・用紙を自動検出する場合は、プリンターのメモリスイッチMS4-6(用紙自動検出選択)を「有効」にしてください。
・用紙を指定する場合は、MS4-6(用紙自動検出選択)を「無効」、MS1-3(マークモード選択)を「有効」にしてください。
また、利用する用紙を下記の通りに選択してください。
・マーク付きライナレスラベル紙を使用する場合:
MS7(感熱紙選択)を「ライナレスラベル紙」にしてください。
・SLPラベル紙を使用する場合:
MS7(感熱紙選択)を「SLPラベル紙」にしてください。

*4: マークなし用紙またはマークなしライナレスラベル紙を利用する場合は、本設定を「無効」にし、プリンターのメモリスイッチMS7(感熱紙選択)で使用する用紙を選択してください。

*5: 下記の条件で**cutPaper**を実行し、直後に**printSmartLabelImageData**を使用する場合は、本設定を「無効」にしてください。
・MS1-3(マークモード選択)が「無効」
・**cutPaper**の*cuttingMethod*に**CUT_FULL**を指定、またはMS1-2(テイクンモード選択)が「有効」かつ*cuttingMethod*に**CUT_PARTIAL**を指定

2.3 注意事項

TCP/IP利用時はプリンタードライバーや他のライブラリと通信ポートの共用はできません。

3章　ライブラリの利用方法

本章では、Androidアプリケーションの開発環境とライブラリの利用方法について説明します。

3.1　Androidアプリケーションの開発環境

Androidアプリケーションを開発するためには、下記のツールが必要です。詳細は各URLを参照してください。

- Android Studio
  https://developer.android.com/studio/index.html
- Windows用USBドライバ（Windows環境で開発する場合）
  https://developer.android.com/studio/run/oem-usb.html

本書では本章以降、各ツールを利用できる環境が整っていることを前提に説明します。

3.2　提供ファイル

SIIプリントクラスライブラリのファイル構成は、下記の通りです。

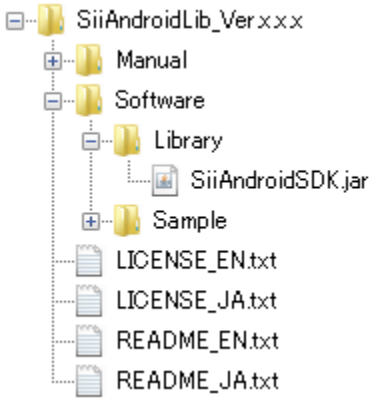


図 3-1

ライブラリは、JARファイル形式です。ライブラリのファイル名は、SiiAndroidSDK.jarです。

3.3 Android Studioのプロジェクトへの組み込み

SIIプリントクラスライブラリをAndroid Studioのプロジェクトに組込む手順を説明します。

SIIプリントクラスライブラリに含まれるサンプルプログラムについては、「5章 サンプルプログラム」を参照してください。

(1) Android Studioでプロジェクトを作成して、[libs]フォルダにライブラリファイル（SiiAndroidSDK.jar）をコピーしてください。[libs]フォルダが自動生成されない場合は、手動でフォルダを追加してください。
SIIプリントクラスライブラリに含まれるサンプルプログラムの場合は、フォルダが「¥Sample¥app¥libs」になります。

(2) ライブラリの追加が完了すると図 3-2の状態となります。

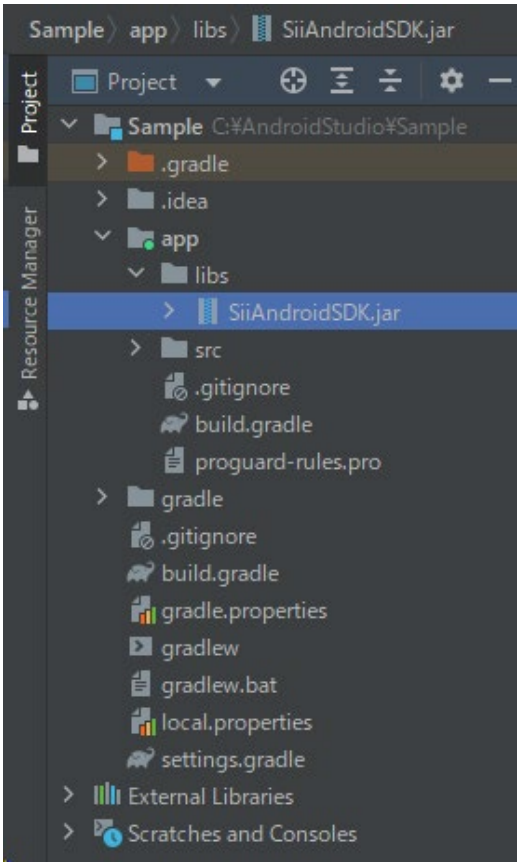


図 3-2

(3) アプリケーションのbuild.gradle(:app)のdependencies{}内に下記を追加してください。

```
implementation 'com.journeyapps:zxing-android-embedded:3.4.0@aar'
implementation 'com.google.zxing:core:3.4.1'
implementation files ('libs/SiiAndroidSDK.jar')
```

(4) メインとなるソースファイルの先頭に下記を追加してください。
（サンプルプログラムでは、MainActivity.javaの先頭に追加してください。）

```
import com.seikoinstruments.sdk.thermalprinter.PrinterManager;
import com.seikoinstruments.sdk.thermalprinter.xxxx;
```

(5) アプリケーションのマニフェスト(AndroidManifest.xml)に下記権限の宣言を追加してください。
また、アプリケーションに適切な権限をリクエストする処理(requestPermissions())を実装してください。

[Bluetoothを利用する場合]

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"/>
```

[TCP/IPを利用する場合]

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

以上で、ライブラリの機能が利用可能になります。

3.4　作成したAndroidアプリケーションのAndroidデバイスでの利用

作成したAndroidアプリケーションをAndroidデバイス上で利用するためには、Androidデバイスで下記の設定を行ってください。

(1)　[設定] – [開発者向けオプション] – [USBデバッグ]をオンにしてください。(図 3-3)

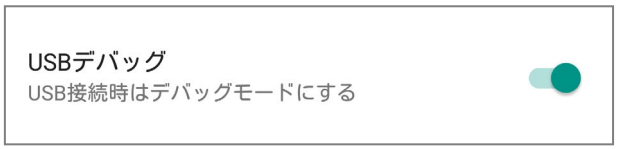


図 3-3

3.5　注意事項

- **対象範囲別ストレージについて**

  Android 10から導入された"対象範囲別ストレージ"ではアプリ固有のストレージとアプリ外部のストレージが区別されます。
  Android 10 (API 29)以降をアプリのターゲットとする場合、アプリ外部のストレージ内にあるメディアファイルに該当しないファイルを直接扱うことができません。メディアファイルに該当しないファイルは"ストレージ アクセス フレームワーク"を利用すると扱うことができます。
  対象範囲別ストレージの詳細については、下記を参照してください。

  - データ ストレージとファイル ストレージの概要
    https://developer.android.com/training/data-storage

4章　ライブラリの機能

本章では、ライブラリに実装されている各クラスのAPIについて説明します。

4.1　ラベル印字機能

ライブラリでは、Smart Label CreatorまたはSII Layout Editorで作成したラベルファイル（*.sl、*.slex）の印字ができます。
また、ラベルファイルを利用し、テキスト、イメージ及びバーコードデータの置き換えを行い、印字できる機能を提供します。

Smart Label CreatorまたはSII Layout Editorはラベル作成ができるソフトウェアです。

本書では、Smart Label Creator及びSII Layout Editorの共通の説明をする場合は、以降「アプリ」と称します。
個別に説明する必要がある場合は製品名を記載します。

アプリは下記のWebページからダウンロードができます。

- SLP720RT/SLP721RTシリーズ ダウンロードページ
  https://www.sii-ps.com/slp720rt/

また、アプリによっては直接ストアに移動し、インストールできます。

iOSまたはAndroid

QRコードをスマートフォンで読み取ってストアへ移動すると、アプリをインストールできます。

- iOS

SII Layout Editor

- Android

SII Layout Editor

Windows

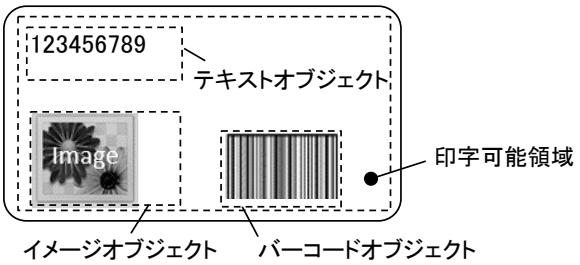
SII Layout Editor

下記をクリックまたはWebブラウザに入力し、ストアへ移動するとアプリをインストールできます。
https://www.microsoft.com/store/apps/9P5G2R1PS76W

参考　・App Store、Google PlayまたはMicrosoft Storeに移動して、「SII Layout Editor」を検索、インストールすることも可能です。

4.1.1　ラベルファイルの構造

ラベルファイルは、ラベルの印字可能領域内にオブジェクトを配置したファイルです。



ラベルファイル（*.sl、*.slex）の例

(1)　オブジェクトの種類とライブラリでのサポート

ライブラリでサポートできるオブジェクトを下記に示します。

オブジェクト	説明	ラベルファイル（*.sl、*.slex）作成元			ライブラリでのサポート
		SII Layout Editor		Smart Label Creator	
		iOS/Android	Windows	Windows	
テキストオブジェクト	テキストデータを扱うオブジェクト	✓	✓	✓	✓
イメージオブジェクト	イメージデータを扱うオブジェクト	✓	✓	✓	✓
バーコードオブジェクト	バーコードデータを扱うオブジェクト	✓	✓	✓	✓
図（四角形）オブジェクト	図（四角形）の描画データを扱うオブジェクト	✓	✓	✓	✓
図（円/楕円）オブジェクト	図（円/楕円）の描画データを扱うオブジェクト	✓	✓	✓	✓

オブジェクト	説明	ラベルファイル（*.sl、*.slex）作成元			ライブラリでのサポート
		SII Layout Editor		Smart Label Creator	
		iOS/Android	Windows	Windows	
図（線）オブジェクト	図（線）の描画データを扱うオブジェクト	✓	✓	✓	✓
飾り枠オブジェクト	飾り枠の描画データを扱うオブジェクト	―	―	✓	―
連絡帳オブジェクト	デバイスの連絡帳からデータを取り込むオブジェクト	✓	―	-	✓
日付と時刻オブジェクト	日付と時刻のデータを扱うオブジェクト	✓	✓	✓	✓
差出人オブジェクト	差出人データを扱うオブジェクト	―	―	✓	✓
グループオブジェクト	複数のオブジェクトをグループ化したオブジェクト	―	―	✓	✓

(2) ラベルファイルをライブラリで印字する際の注意点

アプリを使用したラベルファイルの印字とライブラリを使用したラベルファイルの印字では印字結果が異なる場合がありますので、実機にて事前に確認してください。
ライブラリを使用してラベルファイルを印字する際は、下記の点にご注意ください。

① 全てのオブジェクト

- 印字可能領域外に配置されたオブジェクトの描画はサポートしていません。

② テキストオブジェクト

アプリ

- 「連続番号」はサポートしていません。
- テキストの書式の[下線]に[二重下線]を設定している場合、ライブラリでは[下線]になります。
- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

Smart Label Creator

- 「Zip code」はサポートしていません。
- 「フィールドリンク」はサポートしていません。
- テキストの書式の[取り消し線]に[二重線]を設定している場合、ライブラリでは[線]になります。
- テキストの一部に対して装飾の解除や異なる装飾及びフォントサイズを付与する機能はサポートしていません。一文字目に付与されている装飾及びフォントサイズがテキスト全体の装飾となります。

③ イメージオブジェクト

アプリ

- ディザリングの設定が「バークス」または「ベイヤー」の場合、ライブラリでは「フロイド・スタンバーグ」となります。

Smart Label Creator

- イメージソースが「ファイルのリンク」の場合、ライブラリではイメージオブジェクトの印字ができません。
- イメージソースが「クリップアート」の場合、ライブラリではイメージオブジェクトの印字ができません。
- 「明るさ」及び「コントラスト」はサポートしていません。

④ バーコードオブジェクト

アプリ

- アプリでサポートするバーコードのうち、ライブラリでは下記のバーコードをサポートしています。
  - CODE39
  - ITF
  - CODE128
  - UPC-A
  - EAN13
  - CODABAR
  - UPC-E
  - EAN8
  - PDF417
  - Data Matrix
  - QRコード
- 「連続番号」はサポートしていません。
- 下記のバーコード設定は反映しません。
  - バー幅の比率
- アプリで生成されるバーコードイメージとライブラリで生成するバーコードイメージは同一バーコードイメージにはならない場合があります。
- アプリでバーコードオブジェクトの高さをバー高さより小さく指定した場合、ライブラリではバーコードはオブジェクト内に収まるように縮小して印字します。
- アプリでPDF417の「セキュリティ」を-1に設定した場合、ライブラリではセキュリティレベルを0固定で描画します。
- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

Smart Label Creator

- 「フィールドリンク」はサポートしていません。

- 下記のバーコード設定は反映しません。
  - PDF417のモード
  - Data Matrixのセキュリティ
  - QRコードのモード
- Data Matrixの「モード」を1（長方形）に設定した場合、ライブラリではバーコードをオブジェクトの中央に描画します。
- 書式の「横配置」の設定値によらず、ライブラリではHRI文字を常に左右中央揃えで描画します。

⑤ 図オブジェクト

アプリ

- 「線の太さ」が細い場合、破線・長破線・二重線の印字が潰れる可能性があります。
- アプリとライブラリでの図オブジェクトの描画位置が異なる場合があります。

⑥ 連絡帳オブジェクト

アプリ

- アプリで設定されているフォントがライブラリにない場合、テキストデータはシステム標準のフォントで印字します。

⑦ 日付と時刻オブジェクト

アプリ

- アプリで設定されているフォントがライブラリにない場合、テキストはシステム標準のフォントで印字します。

⑧ グループオブジェクト

Smart Label Creator

- グループ化されていたオブジェクトは、ライブラリではグループ解除されます。
- Smart Label Creatorとライブラリではグループ解除されたオブジェクトの描画位置が異なる場合があります。

4.1.2　ラベルファイルの利用手順

ラベルファイルを利用した印字手順を下記に示します。

(1)　ラベルファイルをそのままライブラリから印字する場合

印字命令例

①ラベルファイルの指定

②ラベルの印字

(2)　ラベルファイルのオブジェクトデータを置換して印字する場合

印字命令例

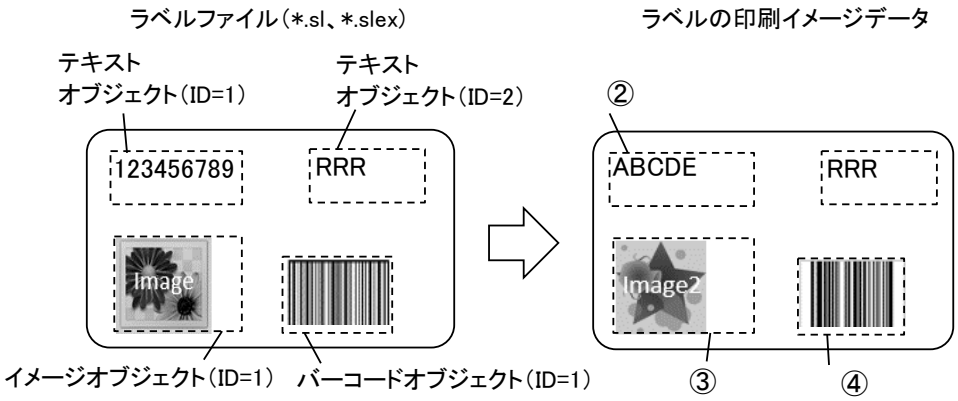
①ラベルファイルの指定

②ラベルのテキストデータの置換（テキストオブジェクト ID=1）

③ラベルのイメージデータの置換（イメージオブジェクト ID=1）

④ラベルのバーコードデータの置換（バーコードオブジェクト ID=1）

⑤ラベルの印字



4.2 ログファイル出力機能

ライブラリではログの取得とログファイルの出力ができます。

4.2.1 ログ出力の設定方法

ライブラリを組み込んだAndroidアプリケーションの固有のディレクトリ(例:内部共有ストレージ¥Android¥data¥<package name>¥files)に、下記内容のconfig.iniファイルを追加することで、ログの出力設定が可能になります。

config.ini

```
LOGLEVEL=x
LOGSIZEMAX=xMB
LOGOUTPUT=x
```

参考　　xの設定内容については、「4.2.2 ログ出力の設定内容」を参照してください。

4.2.2 ログ出力の設定内容

項目	説明	設定内容
LOGLEVEL	ログレベル	0 : ログを記録しません。 1 : **PrinterException**スロー時のエラーログを記録します。 2 : APIの実行履歴を記録します。
LOGSIZEMAX	ログファイル最大サイズ	1MB : ログファイル最大サイズは1MB 5MB : ログファイル最大サイズは5MB 10MB : ログファイル最大サイズは10MB 50MB : ログファイル最大サイズは50MB
LOGOUTPUT	Logcat出力有効・無効	0 : Logcat出力無効 1 : Logcat出力有効

4.2.3 ログファイル

ログファイルは、ライブラリを組み込んだAndroidアプリケーションのローカルファイルとして保存されます。

ログファイル名 : PrinterManagerX.log (Xの範囲は、0～4)

最初のログファイルはPrinterManager0.logで作成されます。ログファイルの最大サイズを超えた場合、ファイル名をPrinterManager1.logに変更し、新たにPrinterManager0.logを作成します。

作成できるログファイルは最大5つです。

4.3 APIリファレンス

ライブラリのパッケージは、**com.seikoinstruments.sdk.thermalprinter**です。
com.seikoinstruments.sdk.thermalprinterには、下記のクラスが含まれます。

クラス名	説明	サポート*1
PrinterManager	プリンターとの通信、印字を行うためのAPIを提供します。 「**4.3.1 PrinterManagerクラス**」を参照してください。	✓
PrinterEvent	**startDiscoveryPrinter**が終了した際の終了イベントを取得するAPIを提供します。 「**4.3.2 PrinterEventクラス**」を参照してください。	✓
PrinterListener	**startDiscoveryPrinter**が終了した際の終了イベントを取得するためのインターフェイスです。 「**4.3.3 PrinterListenerインターフェイス**」を参照してください。	✓
PrinterInfo	**startDiscoveryPrinter**で発見されたプリンター情報を格納します。 「**4.3.4 PrinterInfoクラス**」を参照してください。	✓
PrinterIdentifier	**getPrinterResponse**で取得されたプリンターのプリンタIDを格納します。 「**4.3.5 PrinterIdentifierクラス**」を参照してください。	✓
PrinterException	API呼び出し時にスローされる例外クラスです。 「**4.3.6 PrinterExceptionクラス**」を参照してください。	✓
CallbackFunctionListener	プリンターステータスの変化イベントを取得するためのインターフェイスです。 「**4.3.7 CallbackFunctionListenerインターフェイス**」を参照してください。	✓
BarcodeScannerListener	バーコードスキャナーの接続、バーコードスキャナーの切断、または受信したバーコードデータを取得するためのインターフェイスです。	−
SmartLabelManager	ラベルファイルの指定、データの置き換えを行うためのAPIを提供します。 「**4.3.9 SmartLabelManager**クラス」を参照してください。	✓

*1: ✓:サポート、−:非サポート

（注意） SLP720RT/SLP721RTは、ページモード、ディスプレイ及びバーコードスキャナーのAPIをサポートしていません。

4.3.1 PrinterManagerクラス

(1) メソッド一覧

PrinterManagerクラスで提供されるメソッドを下記に示します。

名前	説明	サポート[*1]
PrinterManager	コンストラクタ	✓
connect	プリンターとの通信開始(Bluetooth)	✓[*2]
connect	プリンターとの通信開始(USB)	✓
connect	プリンターとの通信開始(TCP/IP)	✓
disconnect	プリンターとの通信切断	✓
setBarcodeScannerListener	バーコードスキャナーのコールバック開始・終了	−
sendText	テキストデータの送信	✓
sendTextEx	書式指定テキストデータの送信	✓
printBarcode	バーコードの印字	✓
printPDF417	PDF417の印字	✓
printQRcode	QRコードの印字	✓
printDataMatrix	Data Matrixの印字	✓
printMaxiCode	MaxiCodeの印字	✓
printGS1DataBarStacked	GS1 Databar Stackedの印字	✓
printGS1DataBarStackedOmnidirectional	GS1 Databar Stacked Omni-directionalの印字	✓
printGS1DataBarExpandedStacked	GS1 Databar Expanded Stackedの印字	✓
printAztecCode	Aztec Codeの印字	−
cutPaper	用紙のカット	✓
feedPosition	用紙の頭出し	✓
openDrawer	キャッシュドロワを開く	✓[*2]
buzzer	ブザーの鳴動	−
externalBuzzer	外部ブザーの鳴動	✓[*2]
sendBinary	バイナリデータの送信	✓
sendDataFile	指定ファイルの送信	✓
sendImage	イメージの送信	✓
printPDF	PDFページの印字	✓
getStatus	プリンターステータスの取得	✓
setCallbackFunctionListener	プリンターステータス変化のコールバック開始・終了	✓
abort	プリンターのデータ待ち状態解除	✓
registerLogo	ロゴの登録	✓
printLogo	ロゴの印字	✓
unregisterLogo	ロゴの登録削除	✓
registerStyleSheet	スタイルシートの登録	−
unregisterStyleSheet	スタイルシートの登録削除	−
resetPrinter	プリンターのリセット	✓

名前	説明	サポート*1
getPrinterResponse	プリンターからの各種応答取得	✓
startDiscoveryPrinter	プリンターの探索開始（Bluetooth）	✓*2
startDiscoveryPrinter	プリンターの探索開始（USB）	✓
startDiscoveryPrinter	プリンターの探索開始（TCP/IP）	✓
cancelDiscoveryPrinter	プリンターの探索中断	✓
getFoundPrinter	発見されたプリンター情報リストの取得	✓
getSendTimeout	送信タイムアウト時間の取得	✓
setSendTimeout	送信タイムアウト時間の設定	✓
getReceiveTimeout	受信タイムアウト時間の取得	✓
setReceiveTimeout	受信タイムアウト時間の設定	✓
getInternationalCharacter	国際文字セットの取得	✓
setInternationalCharacter	国際文字セットの設定	✓
getCodePage	コードページの取得	✓
setCodePage	コードページの設定	✓
getPrinterModel	プリンターモデルの取得	✓
getPortType	接続ポートの種別取得	✓
isConnect	プリンターとの通信状態の確認	✓
getSocketKeepingTime	ソケット維持時間の取得	✓
setSocketKeepingTime	ソケット維持時間の設定	✓
getPrintSmartLabelMode	ラベルファイル印刷時の用紙の取得	✓
setPrintSmartLabelMode	ラベルファイル印刷時の用紙の設定	✓
getVersion	SDKバージョンの取得	✓
printSmartLabelImageData	ラベルの印字	✓
controlTransaction	一括処理の開始・終了	✓

*1: ✓：サポート、－：非サポート
*2: SLP721RTのみサポートします。

(2) 定数一覧

① プリンターモデル

プリンターとの通信開始と、プリンターモデル取得で利用する定数を下記に示します。

定数名	説明	値
PRINTER_MODEL_SLP720RT	SLP720RT/SLP721RT	305
PRINTER_MODEL_DEFAULT	プリンターモデルの初期値	284

② 応答種別

プリンターからの各種応答取得で利用する定数を下記に示します。

定数名	説明	値
PRINTER_RESPONSE_REQUEST	実行応答リクエスト	0
PRINTER_RESPONSE_USER_AREA	ユーザ領域の残り容量の送信	1
PRINTER_RESPONSE_ARRANGE_USER_AREA	ユーザ領域の整理後の残り容量の送信	2
PRINTER_RESPONSE_NV_GRAPHICS	NVグラフィックスのメモリ容量の送信	3
PRINTER_RESPONSE_KEY_CODE	定義されているNVグラフィックスのキーコード一覧の送信	4
PRINTER_RESPONSE_MAINTENANCE_COUNTER	メンテンナンスカウンタの送信	7
PRINTER_RESPONSE_PRINTER_IDENTIFIER	プリンタIDの送信	8

③ 国際文字セット

国際文字セットの設定/取得で利用する定数を下記に示します。

定数名	説明	値
COUNTRY_USA	アメリカ(USA)	0
COUNTRY_FRANCE	フランス(France)	1
COUNTRY_GERMANY	ドイツ(Germany)	2
COUNTRY_ENGLAND	イギリス(United Kingdom)	3
COUNTRY_DENMARK_1	デンマークⅠ(Denmark I)	4
COUNTRY_SWEDEN	スウェーデン(Sweden)	5
COUNTRY_ITALY	イタリア(Italy)	6
COUNTRY_SPAIN	スペインⅠ(Spain I)	7
COUNTRY_JAPAN	日本(Japan)	8
COUNTRY_NORWAY	ノルウェー(Norway)	9
COUNTRY_DENMARK_2	デンマークⅡ(Denmark II)	10
COUNTRY_SPAIN_2	スペインⅡ(Spain II)	11
COUNTRY_LATIN_AMERICA	ラテンアメリカ(Latin America)	12
COUNTRY_ARABIA	アラビア(Arabia)	17

④ コードページ

コードページの設定/取得で利用する定数を下記に示します。

定数名	説明	値
CODE_PAGE_437	USA, Standard Europe(Code Page437)	0
CODE_PAGE_KATAKANA	Katakana	1

定数名	説明	値
CODE_PAGE_850	Multilingual(Code Page850)	2
CODE_PAGE_860	Portuguese(Code Page860)	3
CODE_PAGE_863	Canadian-French(Code Page863)	4
CODE_PAGE_865	Nordic(Code Page865)	5
CODE_PAGE_857	Turkish(Code Page857)	13
CODE_PAGE_737	Greek(Code Page737)	14
CODE_PAGE_1252	Latin(Code Page1252)	16
CODE_PAGE_866	Russian(Code Page866)	17
CODE_PAGE_852	Eastern Europe(Code Page852)	18
CODE_PAGE_858	Euro(Code Page858)	19
CODE_PAGE_855	Cyrillic(Code Page855)	34
CODE_PAGE_864[*1*2]	Arabic(Code Page864)	37
CODE_PAGE_1250	Central European(Code Page1250)	45
CODE_PAGE_1251	Cyrillic(Code Page1251)	46
CODE_PAGE_1253	Greek(Code Page1253)	47
CODE_PAGE_1254	Turkish(Code Page1254)	48

*1: Unicodeの20AChは印字できません。
*2: フォントBは印字できません。

⑤ ポート種別

プリンターとの通信開始と、接続ポート種別取得で利用する定数を下記に示します。

定数名	説明	値
PRINTER_TYPE_BLUETOOTH[*1]	Bluetooth	0
PRINTER_TYPE_USB	USB	1
PRINTER_TYPE_TCP	TCP/IP	2

*1: SLP721RTのみサポートします。

⑥ ラベルファイル印刷時マークあり・なし用紙選択

ラベルファイル印刷時の用紙の選択で利用する定数を下記に示します。

定数名	説明	値
PRINTSMARTLABEL_MODE_MARK	マーク紙	0
PRINTSMARTLABEL_MODE_NONEMARK	マークなし用紙	1

⑦ バーコード、PDF417専用

バーコードの印字と、PDF417の印字で利用する定数を下記に示します。

定数名	説明	値
BARCODE_HEIGHT_DEFAULT	バーコード高さの初期値	162
PDF417_MODULE_HEIGHT_DEFAULT	PDF417高さの初期値	10
PDF417_ROW_AUTO	行数自動選択	0
PDF417_COLUMN_AUTO	カラム数自動選択	0

(3) 列挙型定数一覧

① 強調印字（CharacterBold）

強調印字で利用する列挙型定数を下記に示します。

定数名	説明
BOLD_CANCEL	強調印字を解除
BOLD	強調印字を指定

② アンダーライン（CharacterUnderline）

アンダーラインで利用する列挙型定数を下記に示します。

定数名	説明
UNDERLINE_CANCEL	アンダーライン印字を解除
UNDERLINE_1	1ドット幅アンダーライン印字を指定
UNDERLINE_2	2ドット幅アンダーライン印字を指定

③ 白黒反転印字（CharacterReverse）

白黒反転印字で利用する列挙型定数を下記に示します。

定数名	説明
REVERSE_CANCEL	白黒反転印字を解除
REVERSE	白黒反転印字を指定

④ 倒立印字（CharacterInversion）

倒立印字で利用する列挙型定数を下記に示します。
改行前のテキストデータに倒立印字を追加することはできません。

定数名	説明
INVERSION_CANCEL	倒立印字を解除
INVERSION	倒立印字を指定

⑤ 文字フォント（CharacterFont）

文字フォントで利用する列挙型定数を下記に示します。

定数名	説明
FONT_A	フォントA（24×12）
FONT_B	フォントB（16×8）

⑥ 文字倍率（CharacterScale）

文字倍率で利用する列挙型定数を下記に示します。

定数名	説明
VERTICAL_1_HORIZONTAL_1	縦1倍・横1倍
VERTICAL_1_HORIZONTAL_2	縦1倍・横2倍
VERTICAL_1_HORIZONTAL_3	縦1倍・横3倍
VERTICAL_1_HORIZONTAL_4	縦1倍・横4倍
VERTICAL_2_HORIZONTAL_1	縦2倍・横1倍
VERTICAL_2_HORIZONTAL_2	縦2倍・横2倍
VERTICAL_2_HORIZONTAL_3	縦2倍・横3倍
VERTICAL_2_HORIZONTAL_4	縦2倍・横4倍
VERTICAL_2_HORIZONTAL_6	縦2倍・横6倍
VERTICAL_3_HORIZONTAL_1	縦3倍・横1倍
VERTICAL_3_HORIZONTAL_2	縦3倍・横2倍
VERTICAL_3_HORIZONTAL_3	縦3倍・横3倍
VERTICAL_3_HORIZONTAL_4	縦3倍・横4倍
VERTICAL_4_HORIZONTAL_1	縦4倍・横1倍
VERTICAL_4_HORIZONTAL_2	縦4倍・横2倍
VERTICAL_4_HORIZONTAL_3	縦4倍・横3倍
VERTICAL_4_HORIZONTAL_4	縦4倍・横4倍
VERTICAL_4_HORIZONTAL_6	縦4倍・横6倍
VERTICAL_4_HORIZONTAL_8	縦4倍・横8倍

定数名	説明
VERTICAL_6_HORIZONTAL_2	縦6倍・横2倍
VERTICAL_6_HORIZONTAL_4	縦6倍・横4倍
VERTICAL_6_HORIZONTAL_6	縦6倍・横6倍
VERTICAL_6_HORIZONTAL_8	縦6倍・横8倍
VERTICAL_8_HORIZONTAL_4	縦8倍・横4倍
VERTICAL_8_HORIZONTAL_6	縦8倍・横6倍
VERTICAL_8_HORIZONTAL_8	縦8倍・横8倍

⑦ 位置揃え（PrintAlignment）

位置揃えで利用する列挙型定数を下記に示します。
改行前のテキストデータに位置揃えを追加することはできません。

定数名	説明
ALIGNMENT_LEFT	左揃え
ALIGNMENT_CENTER	中央揃え
ALIGNMENT_RIGHT	右揃え

⑧ 保留データの出力指定（OutputPendingData）

保留データの出力指定で利用する列挙型定数を下記に示します。

定数名	説明
PENDING_DATA_OUTPUT_FIRST	保留データを先に出力してから処理を開始する
PENDING_DATA_OUTPUT_TOGETHER	保留データと同時に出力する

⑨ バーコードシンボル（BarcodeSymbol）

バーコードシンボルで利用する列挙型定数を下記に示します。

定数名	説明	構文[*1]
BARCODE_SYMBOL_UPC_A	UPC-A	(a)
BARCODE_SYMBOL_UPC_E	UPC-E	(a)
BARCODE_SYMBOL_EAN13	EAN13	(a)
BARCODE_SYMBOL_JAN13	JAN13	(a)
BARCODE_SYMBOL_EAN8	EAN8	(a)
BARCODE_SYMBOL_JAN8	JAN8	(a)
BARCODE_SYMBOL_CODE39	CODE39	(a), (b)
BARCODE_SYMBOL_CODE93	CODE93	(c)

定数名	説明	構文*1
BARCODE_SYMBOL_CODE128	CODE128	(c)
BARCODE_SYMBOL_ITF	ITF	(a), (b)
BARCODE_SYMBOL_CODABAR	CODABAR	(a), (b)
BARCODE_SYMBOL_EAN13_ADDON	EAN13 add-on	(a)
BARCODE_SYMBOL_JAN13_ADDON	JAN13 add-on	(a)
BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	(a)
BARCODE_SYMBOL_GS1_TRUNCATED	GS1 Databar Truncated	(a)
BARCODE_SYMBOL_GS1_LIMITED	GS1 Databar Limited	(a)
BARCODE_SYMBOL_GS1_EXPANDED	GS1 Databar Expanded	(a)

*1: 構文の内容は、**printBarcode**を参照してください。

⑩ モジュールサイズ(ModuleSize)

バーコードの幅、公称細エレメント幅、及びモジュールサイズで利用する列挙型定数を下記に示します。

定数名	説明	使用メソッド
BARCODE_MODULE_WIDTH_2	細エレメント 2ドット モジュール幅 0.250 mm	**printBarcode**
BARCODE_MODULE_WIDTH_3	細エレメント 3ドット モジュール幅 0.375 mm	
BARCODE_MODULE_WIDTH_4	細エレメント 4ドット モジュール幅 0.500 mm	
BARCODE_MODULE_WIDTH_5	細エレメント 5ドット モジュール幅 0.625 mm	
BARCODE_MODULE_WIDTH_6	細エレメント 6ドット モジュール幅 0.750 mm	
PDF417_MODULE_WIDTH_2	公称細エレメント幅 2ドット	**printPDF417**
PDF417_MODULE_WIDTH_3	公称細エレメント幅 3ドット	
PDF417_MODULE_WIDTH_4	公称細エレメント幅 4ドット	
PDF417_MODULE_WIDTH_5	公称細エレメント幅 5ドット	
PDF417_MODULE_WIDTH_6	公称細エレメント幅 6ドット	
PDF417_MODULE_WIDTH_7	公称細エレメント幅 7ドット	
PDF417_MODULE_WIDTH_8	公称細エレメント幅 8ドット	

定数名	説明	使用メソッド
QR_MODULE_SIZE_2	2ドット	**printQRcode**
QR_MODULE_SIZE_3	3ドット	
QR_MODULE_SIZE_4	4ドット	
QR_MODULE_SIZE_5	5ドット	
QR_MODULE_SIZE_6	6ドット	
QR_MODULE_SIZE_7	7ドット	
QR_MODULE_SIZE_8	8ドット	
QR_MODULE_SIZE_9	9ドット	
QR_MODULE_SIZE_10	10ドット	
QR_MODULE_SIZE_11	11ドット	
QR_MODULE_SIZE_12	12ドット	
QR_MODULE_SIZE_13	13ドット	
QR_MODULE_SIZE_14	14ドット	
QR_MODULE_SIZE_15	15ドット	
QR_MODULE_SIZE_16	16ドット	
DATAMATRIX_MODULE_SIZE_2	2ドット	**printDataMatrix**
DATAMATRIX_MODULE_SIZE_3	3ドット	
DATAMATRIX_MODULE_SIZE_4	4ドット	
DATAMATRIX_MODULE_SIZE_5	5ドット	
DATAMATRIX_MODULE_SIZE_6	6ドット	
DATAMATRIX_MODULE_SIZE_7	7ドット	
DATAMATRIX_MODULE_SIZE_8	8ドット	
DATAMATRIX_MODULE_SIZE_9	9ドット	
DATAMATRIX_MODULE_SIZE_10	10ドット	
DATAMATRIX_MODULE_SIZE_11	11ドット	
DATAMATRIX_MODULE_SIZE_12	12ドット	
DATAMATRIX_MODULE_SIZE_13	13ドット	
DATAMATRIX_MODULE_SIZE_14	14ドット	
DATAMATRIX_MODULE_SIZE_15	15ドット	
DATAMATRIX_MODULE_SIZE_16	16ドット	

定数名	説明	使用メソッド
GS1DATABAR_MODULE_SIZE_2	2ドット	● **printGS1DataBarStacked** ● **printGS1DataBarStackedOmnidirectional** ● **printGS1DataBarExpandedStacked**
GS1DATABAR_MODULE_SIZE_3	3ドット	
GS1DATABAR_MODULE_SIZE_4	4ドット	
GS1DATABAR_MODULE_SIZE_5	5ドット	
GS1DATABAR_MODULE_SIZE_6	6ドット	
GS1DATABAR_MODULE_SIZE_7	7ドット	
GS1DATABAR_MODULE_SIZE_8	8ドット	
GS1DATABAR_MODULE_SIZE_9	9ドット	
GS1DATABAR_MODULE_SIZE_10	10ドット	
GS1DATABAR_MODULE_SIZE_11	11ドット	
GS1DATABAR_MODULE_SIZE_12	12ドット	
GS1DATABAR_MODULE_SIZE_13	13ドット	
GS1DATABAR_MODULE_SIZE_14	14ドット	
GS1DATABAR_MODULE_SIZE_15	15ドット	
GS1DATABAR_MODULE_SIZE_16	16ドット	

⑪ HRI文字印字位置（HriPosition）

HRI文字印字位置で利用する列挙型定数を下記に示します。

定数名	説明
HRI_NONE	印字しない
HRI_POSITION_ABOVE	バーコードの上
HRI_POSITION_BELOW	バーコードの下
HRI_POSITION_ABOVE_BELOW	バーコードの上と下（両方）

⑫ NW比（NwRatio）

NW比で利用する列挙型定数を下記に示します。

定数名	説明
NWRATIO_1TO2	1:2
NWRATIO_1TO2_5	1:2.5
NWRATIO_1TO3	1:3

⑬ エラー訂正レベル（ErrorCorrection）

エラー訂正レベルで利用する列挙型定数を下記に示します。

定数名	説明	使用メソッド
PDF417_ERROR_CORRECTION_0	エラー訂正レベル 0	**printPDF417**
PDF417_ERROR_CORRECTION_1	エラー訂正レベル 1	
PDF417_ERROR_CORRECTION_2	エラー訂正レベル 2	
PDF417_ERROR_CORRECTION_3	エラー訂正レベル 3	
PDF417_ERROR_CORRECTION_4	エラー訂正レベル 4	
PDF417_ERROR_CORRECTION_5	エラー訂正レベル 5	
PDF417_ERROR_CORRECTION_6	エラー訂正レベル 6	
PDF417_ERROR_CORRECTION_7	エラー訂正レベル 7	
PDF417_ERROR_CORRECTION_8	エラー訂正レベル 8	
QR_ERROR_CORRECTION_L	エラー訂正レベル L	**printQRcode**
QR_ERROR_CORRECTION_M	エラー訂正レベル M	
QR_ERROR_CORRECTION_H	エラー訂正レベル H	
QR_ERROR_CORRECTION_Q	エラー訂正レベル Q	

⑭ PDF417シンボル（Pdf417Symbol）

PDF417シンボルで利用する列挙型定数を下記に示します。

定数名	説明
PDF417_STANDARD	PDF417
PDF417_COMPACT	コンパクトPDF417

⑮ QRコードモデル（QrModel）

QRコードモデルで利用する列挙型定数を下記に示します。

定数名	説明
QR_MODEL_2	QRコードモデル2

⑯ Data Matrixモジュール（DataMatrixModule）

Data Matrixモジュールで利用する列挙型定数を下記に示します。

定数名	説明
DATA_MATRIX_AUTO	モジュール数自動
DATA_MATRIX_10_10	モジュール数10×10
DATA_MATRIX_12_12	モジュール数12×12
DATA_MATRIX_14_14	モジュール数14×14
DATA_MATRIX_16_16	モジュール数16×16
DATA_MATRIX_18_18	モジュール数18×18
DATA_MATRIX_20_20	モジュール数20×20
DATA_MATRIX_22_22	モジュール数22×22
DATA_MATRIX_24_24	モジュール数24×24
DATA_MATRIX_26_26	モジュール数26×26
DATA_MATRIX_32_32	モジュール数32×32
DATA_MATRIX_36_36	モジュール数36×36
DATA_MATRIX_40_40	モジュール数40×40
DATA_MATRIX_44_44	モジュール数44×44
DATA_MATRIX_48_48	モジュール数48×48
DATA_MATRIX_52_52	モジュール数52×52
DATA_MATRIX_64_64	モジュール数64×64
DATA_MATRIX_72_72	モジュール数72×72
DATA_MATRIX_80_80	モジュール数80×80
DATA_MATRIX_88_88	モジュール数88×88
DATA_MATRIX_96_96	モジュール数96×96
DATA_MATRIX_104_104	モジュール数104×104
DATA_MATRIX_120_120	モジュール数120×120
DATA_MATRIX_132_132	モジュール数132×132
DATA_MATRIX_144_144	モジュール数144×144
DATA_MATRIX_8_18	モジュール数8×18
DATA_MATRIX_8_32	モジュール数8×32
DATA_MATRIX_12_26	モジュール数12×26
DATA_MATRIX_12_36	モジュール数12×36
DATA_MATRIX_16_36	モジュール数16×36
DATA_MATRIX_16_48	モジュール数16×48

⑰ MaxiCodeモード（MaxiCodeMode）

MaxiCodeモードで利用する列挙型定数を下記に示します。

定数名	説明
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

⑱ カット方法（CuttingMethod）

カット方法で利用する列挙型定数を下記に示します。

定数名	説明	
	用紙カット位置までの紙送り	カット方法
CUT_FULL	あり	フルカット
CUT_FULL_NO_FEED	なし	
CUT_PARTIAL	あり	パーシャルカット
CUT_PARTIAL_NO_FEED	なし	

⑲ 頭出し位置（FeedPosition）

マーク紙またはラベル紙の頭出し位置で利用する列挙型定数を下記に示します。

定数名	説明
FEED_CUTTER	マーク検出後、カット位置まで紙送りを行います。 紙送り長さはプリンターのメモリスイッチMS8～9（マーク位置補正）の長さになります。 紙送り長さの初期値は58ドット（7.25 mm）です。

⑳ ドロワ番号（DrawerNum）

ドロワ番号で利用する列挙型定数を下記に示します。

定数名	説明
DRAWER_1	ドロワ1
DRAWER_2	ドロワ2

㉑ パルス幅（PulseWidth）

パルス幅で利用する列挙型定数を下記に示します。

定数名	説明
ON_OFF_TIME_100	ON/OFF時間 100ミリ秒
ON_OFF_TIME_200	ON/OFF時間 200ミリ秒
ON_OFF_TIME_300	ON/OFF時間 300ミリ秒
ON_OFF_TIME_400	ON/OFF時間 400ミリ秒
ON_OFF_TIME_500	ON/OFF時間 500ミリ秒
ON_OFF_TIME_600	ON/OFF時間 600ミリ秒
ON_OFF_TIME_700	ON/OFF時間 700ミリ秒
ON_OFF_TIME_800	ON/OFF時間 800ミリ秒

㉒ ブザーパターン（BuzzerPattern）

ブザーパターンで利用する列挙型定数を下記に示します。

定数名	説明
BUZZER_PATTERN_1	パターン1
BUZZER_PATTERN_2	パターン2
BUZZER_PATTERN_3	パターン3
BUZZER_PATTERN_4	パターン4

㉓ ディザリング（Dithering）

ディザリングで利用する列挙型定数を下記に示します。

定数名	説明
DITHERING_DISABLE	ディザリング無効
DITHERING_ERRORDIFFUSION	ディザリング有効

㉔ イメージの回転方向（Rotate）

イメージの回転方向で利用する列挙型定数を下記に示します。

定数名	説明
ROTATE_NONE	回転なし
ROTATE_180	180度回転

㉕ イメージの拡大縮小（ImageScale）

イメージの拡大縮小で利用する列挙型定数を下記に示します。

定数名	説明
IMAGE_SCALE_NONE	拡大縮小なし
IMAGE_SCALE_WIDTH_FIT	印字幅に合わせて拡大縮小

㉖ 一括処理選択（TransactionFunction）

一括処理選択で利用する列挙型定数を下記に示します。

定数名	説明
TRANSACTION_CLEAR	一括処理の中止
TRANSACTION_START	一括処理の開始
TRANSACTION_PRINT	一括印字と一括処理の終了

(4) メソッド詳細

PrinterManager　コンストラクタ

com.seikoinstruments.sdk.thermalprinter.PrinterManagerクラスのコンストラクタです。

構文　public **PrinterManager**(Context *context*)

パラメータ　*context*　本メソッドを呼び出すアプリケーションのコンテキストを指定してください。
例： **MainActivity.this**

connect　プリンターとの通信開始（Bluetooth）

Bluetooth接続でプリンターとの通信を開始します。
SLP721RTのみサポートします。
構文（a）は、常にセキュアなモードでプリンターと通信を行います。
構文（b）は、セキュアなモード、またはインセキュアなモードを指定してプリンターとの通信を行います。

構文　(a) public void **connect**(int *printerModel*, String *address*) throws **PrinterException**

(b) public void **connect**(int *printerModel*, String *address*, boolean *secure*) throws **PrinterException**

パラメータ　*printerModel*　Bluetooth接続対象のプリンターモデル定数
利用可能な定数は、「4.3.1(2)① プリンターモデル」を参照してください。

address　Bluetoothアドレス
例： ″00:11:22:AA:BB:CC″

secure　true　セキュアなモードでプリンターと通信
false　インセキュアなモードでプリンターと通信
通常はセキュアなモードでの通信を推奨します。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーの内容は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは他の**PrinterManager**クラスのメソッドを使用する前に呼び出してください。

*address*で指定したBluetoothアドレスに対し、*printerModel*で指定したプリンターを接続します。
また、指定した*printerModel*を基に、接続時にプリンターの初期設定を行います。

本メソッドによりプリンターステータスのモニタリングを開始します。最新のプリンターステータスは**getStatus**で取得が可能です。
プリンターステータスの変化は、**onStatusChanged**、及び**setCallbackFunctionListener**により、イベントとして通知できます。

connect　プリンターとの通信開始（USB）

USB接続でプリンターとの通信を開始します。

構文　public void **connect**(int *printerModel*) throws **PrinterException**

パラメータ　*printerModel*　USB接続対象のプリンターモデル定数
利用可能な定数は、「4.3.1(2)① プリンターモデル」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーの内容は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは他の**PrinterManager**クラスのメソッドを使用する前に呼び出してください。

*printerModel*で指定したプリンターに接続します。
また、指定した*printerModel*を基に、接続時にプリンターの初期設定を行います。

本メソッドによりプリンターステータスのモニタリングを開始します。最新のプリンターステータスは**getStatus**で取得が可能です。
プリンターステータスの変化は、**onStatusChanged**、及び**setCallbackFunctionListener**により、イベントとして通知できます。

connect　プリンターとの通信開始（TCP/IP）

TCP/IP接続でプリンターとの通信を開始します。

構文　public void **connect**(int *printerModel*, String *address*) throws **PrinterException**

パラメータ　*printerModel*　イーサネット接続対象のプリンターモデル定数
利用可能な定数は、「4.3.1(2)① プリンターモデル」を参照してください。

address　IPアドレス
例：”192.168.0.190”

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーの内容は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは他の**PrinterManager**クラスのメソッドを使用する前に呼び出してください。

Androidデバイスと同じネットワークに接続されたプリンターと、TCP/IP接続で通信を開始します。
*address*で指定されたIPアドレスに対して接続を行います。通信にはTCPポート9100番、及び26100番を使用します。また、指定した*printerModel*を基に、接続時にプリンターの初期設定を行います。

- **ライブラリの TCP/IP 接続中のソケットの作成・破棄について**

  ライブラリは**connect**後、**disconnect**するまで作成したソケットを維持します。また、**disconnect**するまで別のアプリケーションから同じプリンターに接続することはできません。

プリンターへのデータ送信完了時を基準とし、**setSocketKeepingTime**で設定するソケット維持時間経過後、使用しているソケットをいったん破棄します。その後すぐに新規のソケットを作成し、次の接続に使用します。

本メソッドによりプリンターステータスのモニタリングを開始します。最新のプリンターステータスは**getStatus**で取得が可能です。
プリンターステータスの変化は、**onStatusChanged**、及び**setCallbackFunctionListener**により、イベントとして通知できます。

disconnect　プリンターとの通信切断

プリンターとの通信を終了し、プリンターステータスのモニタリングを終了します。

構文　public void **disconnect**() throws **PrinterException**

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーの内容は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは**controlTransaction**によって保持している印字データを破棄します。
setCallbackFunctionListenerによって保持されている**CallbackFunctionListener**インターフェイスのインスタンスは破棄され、コールバックも終了します。

注意　本メソッドの直前に、**getPrinterResponse**の**PRINTER_RESPONSE_REQUEST**（実行応答リクエスト）により実行応答を取得することを推奨します。取得しない場合は、Androidデバイスの印字データが全てプリンターに送信される前に通信が切断され、印字データの一部が消失する場合があります。
getPrinterResponseを実行しない場合は、お客様のプログラムにおいて、問題が無いことを十分評価してからご使用ください。

setBarcodeScannerListener　バーコードスキャナーのコールバック開始・終了

本メソッドはサポートしていません。実行した場合は、**PrinterException**をスローします。

構文　public void **setBarcodeScannerListener**(BarcodeScannerListener *listener*)
throws **PrinterException**

sendText　テキストデータの送信

テキストデータを送信します。

構文　public void **sendText**(String *text*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するテキストデータ
一度に指定可能なデータサイズは16KB（16384バイト）です。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　本メソッドは指定されたテキストデータを、**setInternationalCharacter**、及び**setCodePage**を基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。

テキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。

sendTextEx　書式指定テキストデータの送信

書式設定されたテキストデータをプリンターに送信します。
構文(a)及び(c)は、保留データを先に出力してから処理を開始します。
構文(b)は、保留データの出力指定の定数に従い処理を開始します。

構文　(a) public void **sendTextEx**(String *text*,
CharacterBold *bold*,
CharacterUnderline *underline*,
CharacterReverse *reverse*,
CharacterFont *font*,
CharacterScale *scale*,
PrintAlignment *alignment*) throws **PrinterException**

(b) public void **sendTextEx**(String *text*,
CharacterBold *bold*,
CharacterUnderline *underline*,
CharacterReverse *reverse*,
CharacterFont *font*,
CharacterScale *scale*,
PrintAlignment *alignment*,
OutputPendingData *output*) throws **PrinterException**

(c) public void **sendTextEx**(String *text*,
CharacterBold *bold*,
CharacterUnderline *underline*,
CharacterReverse *reverse*,
CharacterInversion *inversion*,
CharacterFont *font*,
CharacterScale *scale*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するテキストデータ
一度に指定可能なデータサイズは16KB(16384バイト)です。

bold　強調印字
利用可能な定数は、「4.3.1(3)① 強調印字(CharacterBold)」を参照してください。

underline　アンダーライン
利用可能な定数は、「4.3.1(3)② アンダーライン（CharacterUnderline）」を参照してください。

reverse　白黒反転印字
利用可能な定数は、「4.3.1(3)③ 白黒反転印字（CharacterReverse）」を参照してください。

inversion　倒立印字
利用可能な定数は、「4.3.1(3)④ 倒立印字（CharacterInversion）」を参照してください。

font　文字フォント
利用可能な定数は、「4.3.1(3)⑤ 文字フォント（CharacterFont）」を参照してください。

scale　文字倍率
利用可能な定数は、「4.3.1(3)⑥ 文字倍率（CharacterScale）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

output　保留データの出力指定
利用可能な定数は、「4.3.1(3)⑧ 保留データの出力指定（OutputPendingData）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　本メソッドは指定されたテキストデータを、**setInternationalCharacter**、及び**setCodePage**を基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。

sendBinary、及び**sendDataFile**により下記のプリンターコマンドを送信してテキストデータのレイアウトを行う場合は、構文（b）のメソッドの*output*に**PENDING_DATA_OUTPUT_TOGETHER**を指定してください。
- 「水平タブ」
- 「絶対位置の指定」
- 「相対位置の指定」

構文（a）及び(c)のメソッドを実行、または構文（b）のメソッドの*output*に**PENDING_DATA_OUTPUT_FIRST**を指定した場合、上記で設定した印字位置は無効になります。

構文（b）のメソッドの*output*に**PENDING_DATA_OUTPUT_TOGETHER**を指定した場合、テキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。

printBarcode　　バーコードの印字

バーコードを印字します。
構文(a)は、バーコードデータを文字列で指定できます。
構文(b)は、バーコードデータを文字列で指定でき、NW比を指定できます。
構文(c)は、バーコードデータをバイト配列で指定できます。
構文(d)は、サポートしていません。

構文　(a) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
String *text*,
ModuleSize *moduleSize*,
int *moduleHeight*,
HriPosition *hriPosition*,
CharacterFont *hriFont*,
PrintAlignment *alignment*) throws **PrinterException**

(b) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
String *text*,
ModuleSize *moduleSize*,
int *moduleHeight*,
HriPosition *hriPosition*,
CharacterFont *hriFont*,
PrintAlignment *alignment*,
NwRatio *nwRatio*) throws **PrinterException**

(c) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
byte[] *data*,
ModuleSize *moduleSize*,
int *moduleHeight*,
HriPosition *hriPosition*,
CharacterFont *hriFont*,
PrintAlignment *alignment*) throws **PrinterException**

(d) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
String *text*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*barcodeSymbol*　　バーコードシンボル
利用可能な定数、及び対応する構文は、「4.3.1(3)⑨ バーコードシンボル（BarcodeSymbol）」を参照してください。

text(data)　　プリンターに送信するバーコードデータ
バーコードの入力条件は下記の通りです。

バーコード	データ数	入力可能データ 文字列（データ）	備考
UPC-A	11～12文字	'0' ～ '9'	
UPC-E	11～12文字	'0' ～ '9'	

バーコード	データ数	入力可能データ 文字列（データ）	備考
EAN13 JAN13	12～13文字	'0' ～ '9'	
EAN8 JAN8	7～8文字	'0' ～ '9'	
CODE39	1～150文字	'0' ～ '9' 'A' ～ 'Z' ' '、'$'、'%'、'+'、'−'、'.'、'/'	スタートコード及びストップコード('*')は自動付加されます。
CODE93	1～150バイト	(0x00 ～ 0x2E)	末尾に0x2F以上のデータを入力してください。
CODE128	2～150バイト	(0x00 ～ 0x66)	CODE128コードセットのスタートコード(0x67～0x69)で入力する場合。 末尾に0x67以上のデータを入力してください。
		(0x00 ～ 0x7F)	CODE128特殊コードのスタートコード(″{A″, ″{B″, ″{C″)で開始する場合。
ITF	2～150文字 (ただし偶数個)	'0' ～ '9'	
CODABAR	1～150文字	'0' ～ '9' '$'、'+'、'−'、'.'、'/'、':'	先頭と末尾に'A'～'D'のいずれかを指定する必要があります。
EAN13 add-on JAN13 add-on	Add-on 2: 14～15文字 Add-on 5: 17～18文字	'0' ～ '9'	
カスタマバーコード	−	−	サポートしていません。
GS1 Databar Omni-directional	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Truncated	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Limited	13文字	'0' ～ '9'	チェックデジットは自動計算されます。
GS1 Databar Expanded	2～255文字	' ' ～ '‴' '%' ～ '?' 'A' ～ 'Z' ' ' 'a' ～ 'z' '{'	

moduleSize　　バーコードの幅

利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ(ModuleSize)」を参照してください。

moduleHeight　　バーコードの高さ(ドット)

- *barcodeSymbol*が下記の場合の有効範囲は、1～255です。

  BARCODE_SYMBOL_UPC_A
  BARCODE_SYMBOL_UPC_E
  BARCODE_SYMBOL_EAN13
  BARCODE_SYMBOL_JAN13

BARCODE_SYMBOL_EAN8
BARCODE_SYMBOL_JAN8
BARCODE_SYMBOL_CODE39
BARCODE_SYMBOL_CODE93
BARCODE_SYMBOL_CODE128
BARCODE_SYMBOL_ITF
BARCODE_SYMBOL_CODABAR
BARCODE_SYMBOL_EAN13_ADDON
BARCODE_SYMBOL_JAN13_ADDON

- *barcodeSymbol*が下記の場合は、*barcodeSymbol*と*moduleSize*により有効範囲が異なります。

barcodeSymbol	*moduleSize*	有効範囲
BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL		
	BARCODE_MODULE_WIDTH_2	66 ～ 255
	BARCODE_MODULE_WIDTH_3	99 ～ 255
	BARCODE_MODULE_WIDTH_4	132 ～ 255
	BARCODE_MODULE_WIDTH_5	165 ～ 255
	BARCODE_MODULE_WIDTH_6	198 ～ 255
BARCODE_SYMBOL_GS1_TRUNCATED		
	BARCODE_MODULE_WIDTH_2	26 ～ 255
	BARCODE_MODULE_WIDTH_3	39 ～ 255
	BARCODE_MODULE_WIDTH_4	52 ～ 255
	BARCODE_MODULE_WIDTH_5	65 ～ 255
	BARCODE_MODULE_WIDTH_6	78 ～ 255
BARCODE_SYMBOL_GS1_LIMITED		
	BARCODE_MODULE_WIDTH_2	20 ～ 255
	BARCODE_MODULE_WIDTH_3	30 ～ 255
	BARCODE_MODULE_WIDTH_4	40 ～ 255
	BARCODE_MODULE_WIDTH_5	50 ～ 255
	BARCODE_MODULE_WIDTH_6	60 ～ 255
BARCODE_SYMBOL_GS1_EXPANDED		
	BARCODE_MODULE_WIDTH_2	68 ～ 255
	BARCODE_MODULE_WIDTH_3	102 ～ 255
	BARCODE_MODULE_WIDTH_4	136 ～ 255
	BARCODE_MODULE_WIDTH_5	170 ～ 255
	BARCODE_MODULE_WIDTH_6	204 ～ 255

hriPosition　　HRI文字印字位置
利用可能な定数は、「4.3.1(3)⑪ HRI文字印字位置（HriPosition）」を参照してください。

hriFont　　HRI文字フォント
利用可能な定数は、「4.3.1(3)⑤ 文字フォント（CharacterFont）」を参照してください。

alignment　　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

nwRatio　　NW比
利用可能な定数は、「4.3.1(3)⑫ NW比（NwRatio）」を参照してください。
指定した*moduleSize*と*nwRatio*により、太エレメントの幅が下表のように設定されます。

moduleSize	*nwRatio*		
	NWRATIO_1TO2	**NWRATIO_1TO2_5**	**NWRATIO_1TO3**
BARCODE_MODULE_WIDTH_2	0.500 mm（4ドット）	0.625 mm（5ドット）	0.750 mm（6ドット）
BARCODE_MODULE_WIDTH_3	0.750 mm（6ドット）	1.000 mm（8ドット）	1.125 mm（9ドット）
BARCODE_MODULE_WIDTH_4	1.000 mm（8ドット）	1.250 mm（10ドット）	1.500 mm（12ドット）
BARCODE_MODULE_WIDTH_5	1.250 mm（10ドット）	1.625 mm（13ドット）	1.875 mm（15ドット）
BARCODE_MODULE_WIDTH_6	1.500 mm（12ドット）	1.875 mm（15ドット）	2.250 mm（18ドット）

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

注意　　クワイエットゾーンの確保は行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printPDF417　PDF417の印字

PDF417を印字します。
構文(a)は、PDF417シンボルを指定できます。
構文(b)は、PDF417シンボルは通常のPDF417に固定されます。

構文　(a) public void **printPDF417**(String *text*,
ErrorCorrection *errorCorrection*,
int *row*,
int *column*,
ModuleSize *moduleSize*,
int *moduleHeight*,
PrintAlignment *alignment*,
Pdf417Symbol *pdf417Symbol*) throws **PrinterException**

(b) public void **printPDF417**(String *text*,
ErrorCorrection *errorCorrection*,
int *row*,
int *column*,
ModuleSize *moduleSize*,
int *moduleHeight*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するバーコードデータ

errorCorrection　エラー訂正レベル
利用可能な定数は、「4.3.1(3)⑬ エラー訂正レベル(ErrorCorrection)」を参照してください。

row　段数(段)
有効範囲は、0、3～90です。
0を指定した場合は段数が自動設定されます。

column　データ領域のカラム数
有効範囲は、0～30です。
0を指定した場合はデータ領域のカラム数が自動設定されます。

moduleSize　公称細エレメント幅
利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ(ModuleSize)」を参照してください。

moduleHeight　段高さ(ドット)
有効範囲は、2～127です。
段高さの設定を小さくすると、バーコードスキャナーによっては読み取れない場合があります。通常の使用では、3以上を設定してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え(PrintAlignment)」を参照してください。

pdf417Symbol　PDF417のシンボル
利用可能な定数は、「4.3.1(3)⑭ PDF417シンボル（Pdf417Symbol）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

注意　クワイエットゾーンの確保は行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printQRcode　QRコードの印字

QRコードを印字します。
構文（a）は、QRコードモデルはQRコードモデル2に固定されます。
構文（b）は、QRコードモデルを指定できます。

構文　(a) public void **printQRcode**(String *text*,
ErrorCorrection *errorCorrection*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

(b) public void **printQRcode**(String *text*,
ErrorCorrection *errorCorrection*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*,
QrModel *model*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するバーコードデータ
バージョンは構文（a）、（b）のいずれでも*text*で設定したデータ数に応じて自動設定されます。

errorCorrection　エラー訂正レベル
利用可能な定数は、「4.3.1(3)⑬ エラー訂正レベル（ErrorCorrection）」を参照してください。

moduleSize　モジュールサイズ
利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ（ModuleSize）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

model　QRコードモデル
利用可能な定数は、「4.3.1(3)⑮ QRコードモデル（QrModel）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

注意　クワイエットゾーンの確保は行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printDataMatrix　Data Matrixの印字

Data Matrixを印字します。

構文　public void **printDataMatrix**(String *text*,
DataMatrixModule *dataMatrixModule*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するバーコードデータ

dataMatrixModule　Data Matrixモジュール数
利用可能な定数は、「4.3.1(3)⑯ Data Matrixモジュール（DataMatrixModule）」を参照してください。

moduleSize　モジュールサイズ
利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ（ModuleSize)」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

注意　クワイエットゾーンの確保は行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printMaxiCode　MaxiCodeの印字

MaxiCodeを印字します。

構文　public void **printMaxiCode**(String *text*,
MaxiCodeMode *maxiCodeMode*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するバーコードデータ

- *maxiCodeMode*が**MAXI_CODE_2**の場合
  データの先頭にサービスクラス（3桁）、国コード（3桁）、郵便番号（9桁）を付加してください。
- *maxiCodeMode*が**MAXI_CODE_3**の場合
  データの先頭にサービスクラス（3桁）、国コード（3桁）、郵便番号（6桁）を付加してください。

maxiCodeMode　MaxiCodeモード
利用可能な定数は、「4.3.1(3)⑰ MaxiCodeモード（MaxiCodeMode）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

注意　クワイエットゾーンの確保は行いません。バーコードシンボルの規格に応じて、クワイエットゾーンの設定を行ってください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printGS1DataBarStacked　GS1 Databar Stackedの印字

GS1 Databar Stackedを印字します。

構文　public void **printGS1DataBarStacked**(String *text*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するバーコードデータ
'0'～'9'を13文字入力してください。先頭の'01'はプリンターにより自動付加されます。チェックデジットはプリンターにより自動計算されます。

moduleSize　　モジュールサイズ
利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ（ModuleSize)」を参照してください。

alignment　　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printGS1DataBarStackedOmnidirectional　GS1 Databar Stacked Omni-directionalの印字

GS1 Databar Stacked Omni-directionalを印字します。

構文　public void **printGS1DataBarStackedOmnidirectional**(String *text*,
int *moduleHeight*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　　プリンターに送信するバーコードデータ
'0'～'9'を13文字入力してください。先頭の'01'はプリンターにより自動付加されます。チェックデジットはプリンターにより自動計算されます。

moduleHeight　　段高さ(モジュール数)
有効範囲は、33～255です。

moduleSize　　モジュールサイズ
利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ（ModuleSize)」を参照してください。

alignment　　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printGS1DataBarExpandedStacked　GS1 Databar Expanded Stackedの印字

GS1 Databar Expanded Stackedを印字します。

構文　public void **printGS1DataBarExpandedStacked**(String *text*,
int *column*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*text*　プリンターに送信するバーコードデータ
' '、'!'、'"'、'%'、'&'、'''、'('、')'、'*'、'+'、','、'-'、'.'、'/'、':'、';'、'<'、'='、'>'、'?'、'_'、'0'～'9'、'A'～'Z'、'a'～'z'を任意の文字数分入力してください。
FNC1は、'{1'を入力してください。チェックデジットはプリンターにより自動計算されませんので必ず入力してください。

column　カラム数
1行のカラム数を指定してください。
有効範囲は、2～20の偶数です。

moduleSize　モジュールサイズ
利用可能な定数は、「4.3.1(3)⑩ モジュールサイズ（ModuleSize）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

参考　バーコードサイズについての詳細は、「付録B バーコードサイズ一覧」を参照してください。

printAztecCode　Aztec Codeの印字

本メソッドはサポートしていません。実行した場合は、**PrinterException**をスローします。

構文　public void **printAztecCode**(String *text*,
int *layer*,
int *errorCorrection*,
ModuleSize *moduleSize*,
AztecSymbol *aztecSymbol*,
PrintAlignment *alignment*) throws **PrinterException**

cutPaper　用紙のカット

用紙カット位置までの紙送りあり／なしを選択し、用紙のカットを行います。

構文　public void **cutPaper**(CuttingMethod *cuttingMethod*) throws **PrinterException**

パラメータ　*cuttingMethod*　カット方法
利用可能な定数は、「4.3.1(3)⑱ カット方法（CuttingMethod）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

feedPosition　用紙の頭出し

マーク紙またはラベル紙をカット位置まで頭出しを行います。

構文　public void **feedPosition**(FeedPosition *feedPosition*) throws **PrinterException**

パラメータ　*feedPosition*　頭出し位置
利用可能な定数は、「4.3.1(3)⑲ 頭出し位置（FeedPosition）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

注意　マーク紙またはラベル紙の頭出し位置で本メソッドを実行しても、頭出しは行いません。

openDrawer　キャッシュドロワを開く

指定されたキャッシュドロワをオープンします。
SLP721RTのみサポートします。

構文　public void **openDrawer**(DrawerNum *drawerNum*, PulseWidth *onOffTime*) throws **PrinterException**

パラメータ　*drawerNum*　ドロワ番号
利用可能な定数は、「4.3.1(3)⑳ ドロワ番号（DrawerNum）」を参照してください。

onOffTime　パルス幅
利用可能な定数は、「4.3.1(3)㉑ パルス幅（PulseWidth）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

buzzer　ブザーの鳴動

本メソッドはサポートしていません。実行した場合は、**PrinterException**をスローします。

構文　public void **buzzer**(int *onTime*, int *offTime*) throws **PrinterException**

externalBuzzer　外部ブザーの鳴動

外部ブザーを鳴動します。
SLP721RTのみサポートします。

構文　public void **externalBuzzer**(BuzzerPattern *buzzerPattern*, int *buzzerCount*)
throws **PrinterException**

パラメータ　*buzzerPattern*　ブザーパターン
利用可能な定数は、「4.3.1(3)㉒ ブザーパターン（BuzzerPattern）」を参照してください。
外部ブザーの鳴動は下記のいずれかの条件で停止します。
・*buzzerCount*で設定したブザーの回数分の鳴動
・カバーオープン
・プリンターコマンド「外付けブザーの停止」の実行

buzzerCount　鳴動回数（回）
*buzzerCount*で設定した回数だけ外部ブザーを鳴動します。
有効範囲は、1～255です。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

sendBinary　バイナリデータの送信

バイナリデータをプリンターに送信します。

構文　public void **sendBinary**(byte [] *binary*) throws **PrinterException**

パラメータ　*binary*　プリンターに送信するバイナリデータ
一度に指定可能なデータサイズは3MB(3145728バイト)です。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　本メソッドは指定されたバイナリデータを変換せずにプリンターに送信します。

本メソッドでプリンターコマンドをバイナリデータとして送信することにより、ライブラリではサポートされていないプリンター機能を利用できます。ただし、本メソッドはプリンターから応答を取得するコマンドには対応しておりません。

sendDataFile　指定ファイルの送信

ファイルデータを送信します。
構文(a)は、ディザリングは無効に固定されます。
構文(b)は、ディザリングを指定できます。

構文　(a) public void **sendDataFile**(String *fileName*,
PrintAlignment *alignment*) throws **PrinterException**

(b) public void **sendDataFile**(String *fileName*,
PrintAlignment *alignment*,
Dithering *dithering*) throws **PrinterException**

パラメータ　*fileName*　プリンターに送信するデータファイルパス
入力可能な形式について下記に説明します。

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.5 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

指定可能なファイルサイズは、最大16MB(16777216バイト)です。
送信可能なファイル拡張子とファイルの送信について下記に説明します。

- .bmp、.jpg、.jpeg、.png
  イメージファイルとしてプリンターに送信します。カラーイメージの場合は、2値化処理によりモノクロイメージに変換して登録します。イメージファイルをプリンターのメモリ上に展開してから一括で印字を行ないます。

- .txt
  テキストデータとしてプリンターに送信します。テキストデータのフォーマットはUTF-8をサポートしています。**setInternationalCharacter**、及び**setCodePage**の設定を基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。
  本メソッドではテキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。

- .bin、.dat
  バイナリデータとして変換せずにプリンターに送信します。

alignment　位置揃え
*fileName*で指定されたファイルの拡張子が.bmp、.jpg、.jpeg、.png、.txtの場合に有効です。
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

dithering　ディザリング
*fileName*で指定されたファイルの拡張子が.bmp、.jpg、.jpeg、.pngの場合に有効です。
利用可能な定数は、「4.3.1(3)㉓ ディザリング（Dithering）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

sendImage　イメージの送信

指定された「android.graphics.Bitmap」型のイメージを印字します。
構文（a）は、イメージの位置揃えとディザリングを指定できます。
構文（b）は、イメージの位置揃えが指定でき、ディザリングは無効に固定されます。
構文（c）は、イメージの位置揃えは左揃え、ディザリングは無効に固定されます。

構文　(a) public void **sendImage**(Bitmap *bitmap*,
PrintAlignment *alignment*,
Dithering *dithering*) throws **PrinterException**

(b) public void **sendImage**(Bitmap *bitmap*,
PrintAlignment *alignment*) throws **PrinterException**

(c) public void **sendImage**(Bitmap *bitmap*) throws **PrinterException**

パラメータ　*bitmap*　プリンターに送信する「android.graphics.Bitmap」型オブジェクト
イメージファイルとしてプリンターに送信します。カラーイメージの場合は、2値化処理によりモノクロイメージに変換して登録します。イメージファイルをプリンターのメモリ上に展開してから一括で印字を行ないます。

alignment 位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

dithering ディザリング
利用可能な定数は、「4.3.1(3)㉓ ディザリング（Dithering）」を参照してください。

例外 **PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

printPDF PDFページの印字

選択したPDFファイルの指定したページを印字します。

構文 public void **printPDF**(String *fileName*,
int *startIndex*,
int *endIndex*,
Rotate *rotate*,
ImageScale *imageScale*,
int *bottomMargin*,
Dithering *dithering*,
PrintAlignment *alignment*) throws **PrinterException**

パラメータ *fileName* PDFファイルパス
サポートするPDFのファイル拡張子は、.pdfです。
指定可能なファイルサイズは、最大16MB(16777216バイト)です。
入力可能な形式について下記に説明します。

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.5 注意事項 - 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

startIndex　印字するページの開始番号
有効範囲は、-1、及び1～PDFファイルのページ数です。
*startIndex*に-1を指定した場合、全てのページを印字します。
*startIndex*に-1を指定した場合、*endIndex*の値は無視されます。
*startIndex*に1以上を指定した場合、*startIndex*に指定したページ番号から*endIndex*に指定したページ番号のページまでを印字します。
*startIndex*に*endIndex*で指定した値より大きい値を指定した場合はエラーとなります。
*startIndex*及び*endIndex*に範囲外の値を指定した場合、エラーとなります。

endIndex　印字するページの終了番号
有効範囲は、1～2147483647です。
*endIndex*にPDFファイルページ数より大きい値を指定した場合、*startIndex*に指定したページ番号からPDFファイルの最後のページまでを印字します。

rotate　イメージの回転方向
利用可能な定数は、「4.3.1(3)㉔ イメージの回転方向（Rotate）」を参照してください。

imageScale　イメージの拡大縮小
利用可能な定数は、「4.3.1(3)㉕ イメージの拡大縮小（ImageScale）」を参照してください。
*imageScale*に**IMAGE_SCALE_WIDTH_FIT**を指定した場合、アスペクト比を維持しながら画像の幅をプリンターの印字幅に変換します。

bottomMargin　下余白（ドット）
有効範囲は、-1、及び0～2400です。
*bottomMargin*に-1を指定した場合、下余白を維持してイメージを作成し印字します。
*bottomMargin*に0～2400の値を指定した場合、下余白を指定したサイズに変更します。
*bottomMargin*に0～2400の値を指定した場合、空白ページの印字は行いません。

dithering　ディザリング
利用可能な定数は、「4.3.1(3)㉓ ディザリング（Dithering）」を参照してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　PDFデータを、プリンターで印字可能な形式に変換してプリンターに送信します。

カラーのイメージデータは、二値化処理によりモノクロイメージに変換します。

プリンターのメモリスイッチMS1-3(マークモード選択)が「有効」の場合、PDFページの印字後マーク紙またはラベル紙の頭出しを行います。

注意　一度に100枚より多く印字した場合の印字動作保証は行っておりません。

getStatus　プリンターステータスの取得

最新のプリンターステータスを取得します。
構文(a)は、プリンターステータスが戻り値で返されます。
構文(b)は、プリンターステータスがint型の配列に格納されます。

構文　(a) public int **getStatus**() throws **PrinterException**

(b) public void **getStatus**(int [] *buf*) throws **PrinterException**

戻り値　プリンターから取得したステータス

パラメータ　*buf*　プリンターから取得したステータス

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　プリンターから取得したステータスは、NSInteger型の配列に格納されます。

プリンターステータスを下記に示します。
通信失敗時は、0x80000000を示します。

ビット	機能	値	
		0	1
0	電圧エラー	エラーなし	エラーあり
1	ハードウェアエラー	エラーなし	エラーあり
2	ヘッド温度エラー	エラーなし	エラーあり
3	カッタエラー	エラーなし	エラーあり
4	紙無しエラー	エラーなし	エラーあり
5	予約済み	固定	-
6	マーク検出時紙ジャムエラー	エラーなし	エラーあり
7	カバーオープンエラー	エラーなし	エラーあり
8	フィードスイッチの状態	OFF	ON
9	予約済み	固定	-
10	紙送り状態	停止	動作中
11	復帰待ち状態	復帰待ち状態でない	復帰待ち状態である
12	予約済み	固定	-
13	テイクンセンサの状態	紙取り出し済み	紙取り出し待ち
14	予約済み	-	固定
15	ドロワスイッチ入力の状態	Low[*1]	High

ビット	機能	値	
		0	1
16	フラッシュメモリ書き換え中	書き換え中でない	書き換え中である
17～18	予約済み	固定	-
19～31	予約済み	-	固定

*1: SLP720RTはLowに固定です。

setCallbackFunctionListener　プリンターステータス変化のコールバック開始・終了

プリンターステータスの変化に応じて実行するコールバックを開始、または終了します。

構文　public void **setCallbackFunctionListener**(CallbackFunctionListener *listener*) throws **PrinterException**

パラメータ　*listener*　**CallbackFunctionListener**インターフェイスのインスタンス

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　コールバックで実行する処理は**onStatusChanged**で登録してください。
*listener*に**CallbackFunctionListener**インターフェイスのインスタンスを指定して実行すると、コールバックを開始します。
*listener*にnullを指定して実行すると、コールバックを終了します。

保持されている**CallbackFunctionListener**インターフェイスのインスタンスは下記のどちらの条件で破棄されます。

- *listener*にnullを指定して本メソッドを実行
- **disconnect**の実行

本メソッドの呼び出しは、**connect**を実行して、**isConnect**がtrueの時に使用してください。

abort　プリンターのデータ待ち状態解除

プリンターのデータ待ち状態を解除します。

構文　public void **abort**() throws **PrinterException**

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　**sendDataFile**によるイメージファイルの送信が中断した場合、プリンターは指定したイメージファイルの受信完了まで、他の処理を受け付けません（メソッドや送信データが誤解釈され、イメージファイルの続きと認識されます）。この状況を解消するために、本メソッドを利用し、プリンターのデータ待ち状態を解除します。
本メソッドを実行した場合、未印字のイメージファイルの一部が印字されることがあります。

registerLogo　ロゴの登録

イメージファイルをロゴとしてプリンターのNVグラフィックスのメモリに登録します。
構文(a)は、ディザリングはディザリング無効に固定されます。
構文(b)は、ディザリングを指定できます。

構文　(a) public void **registerLogo**(String *fileName*, String *id*) throws **PrinterException**

(b) public void **registerLogo**(String *fileName*,
String *id*,
Dithering *dithering*) throws **PrinterException**

パラメータ　*fileName*　ロゴとして登録するイメージファイルのファイルパス
入力可能な形式について下記に説明します。

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.5 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

サポートするイメージファイルのファイル拡張子は、.bmp、.jpg、.jpeg、.pngです。カラーイメージの場合は、2値化処理によりモノクロイメージに変換して登録します。

id　登録するロゴのID(キーコード)
登録するロゴのIDを2文字の文字列で指定してください。
有効な文字は、英数字('0'～'9'、'A'～'Z'、'a'～'z')などのASCII文字コード20h(スペース)～7Eh(チルダ)の文字です。

dithering　ディザリング
利用可能な定数は、「4.3.1(3)㉓ ディザリング(Dithering)」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

printLogo　ロゴの印字

登録済みのロゴを印字します。

構文　public void **printLogo**(String *id*, PrintAlignment *alignment*) throws **PrinterException**

パラメータ　*id*　印刷するロゴのID（キーコード）
登録済みのロゴのIDを文字列で指定してください。

alignment　位置揃え
利用可能な定数は、「4.3.1(3)⑦ 位置揃え（PrintAlignment）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

unregisterLogo　ロゴの登録削除

登録済みのロゴを削除します。

構文　public void **unregisterLogo**(String *id*) throws **PrinterException**

パラメータ　*id*　削除するロゴのID（キーコード）
登録済みのロゴのIDを文字列で指定してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

registerStyleSheet　スタイルシートの登録

本メソッドはサポートしていません。実行した場合は、**PrinterException**をスローします。

構文　public void **registerStyleSheet**(String *fileName*, int *num*) throws **PrinterException**

unregisterStyleSheet　スタイルシートの登録削除

本メソッドはサポートしていません。実行した場合は、**PrinterException**をスローします。

構文　public void **unregisterStyleSheet**(int *num*) throws **PrinterException**

resetPrinter　プリンターのリセット

プリンターをハードウェアリセットします。

構文　public void **resetPrinter**() throws **PrinterException**

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　Bluetooth接続の場合：
プリンターコマンド「プリンタリセット」によりプリンターをハードウェアリセットします。

USB接続の場合：
USBプリンタークラスのSOFT_RESET機能によりプリンターをリセットします。

TCP/IP接続の場合：
接続中のプリンターに対して、TCPポート26100番への弊社独自コマンド（リセット要求）によりプリンターをリセットします。

本メソッド実行後も、プリンターとの接続は保持されます。

getPrinterResponse　プリンターからの各種応答取得

プリンターの応答データを取得します。

構文　public void **getPrinterResponse**(int *id,* Object *buf*) throws **PrinterException**

パラメータ　*id*　応答種別定数
利用可能な定数は、「4.3.1(2)② 応答種別」を参照してください。

buf　取得した応答データを格納するバッファー
*id*で指定した応答データを*buf*で指定されたオブジェクトに格納します。
応答種別定数によりバッファーの型が異なります。
バッファーの型は、下表を参照してください。

応答種別定数		
	パラメータ	説明
PRINTER_RESPONSE_REQUEST（実行応答リクエスト）		
	buf	長さ1のint型の配列を指定してください。 *buf[0]*には0～15（00h～0Fh）を指定してください。 応答取得成功時には、*buf[0]*に実行応答リクエストの応答コードが128～143（80h～8Fh）で格納されます。

応答種別定数		
	パラメータ	説明
PRINTER_RESPONSE_USER_AREA（ユーザ領域の残り容量の送信）		
	buf	長さ1のint型の配列を指定してください。 応答取得成功時には、ユーザ領域の残り容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_ARRANGE_USER_AREA（ユーザ領域の整理後の残り容量の送信）		
	buf	長さ1のint型の配列を指定してください。 応答取得成功時には、ユーザ領域の整理後の残り容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_NV_GRAPHICS（NVグラフィックスのメモリ容量の送信）		
	buf	長さ1のint型の配列を指定してください。 応答取得成功時には、NVグラフィックスのメモリ容量（単位：バイト）が数値で格納されます。
PRINTER_RESPONSE_KEY_CODE（定義されているNVグラフィックスのキーコード一覧の送信）		
	buf	ArrayList<String>型の配列を指定してください。 応答取得成功時には、NVグラフィックスのキーコードが文字列の配列で格納されます。 例：*buf.size()* = 3、 *buf[0]* = ”22”、 *buf[1]* = ”23”、*buf[2]* = ”24”など
PRINTER_RESPONSE_MAINTENANCE_COUNTER（メンテンナンスカウンタの送信）		
	buf	長さ8のint型の配列を指定してください。 応答取得成功時は、下記のように格納されます。 *buf[0]* ：紙送り行数（単位：100ドットライン） *buf[1]* ：サーマルヘッド通電回数（単位：100ドットライン） *buf[2]* ：オートカッタ駆動回数 *buf[3]* ：製品の駆動時間（単位：分） *buf[4]* ：紙送り行数（積算）（単位：100ドットライン） *buf[5]* ：サーマルヘッド通電回数（積算）（単位：100ドットライン） *buf[6]* ：オートカッタ駆動回数（積算） *buf[7]* ：製品の駆動時間（積算）（単位：分）
PRINTER_RESPONSE_PRINTER_IDENTIFIER（プリンタIDの送信）		
	buf	ArrayList<PrinterIdentifier>型のリストを指定してください。 応答取得成功時には、*buf*の先頭要素に**PrinterIdentifier**クラスのオブジェクトが格納されます。 例：*buf.size()* = 1、 *buf.get(0)* = ”（**PrinterIdentifier**クラスのオブジェクト）” **PrinterIdentifier**クラスのオブジェクトについては、 「**4.3.5 PrinterIdentifierクラス**」を参照してください。

例外　　**PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

startDiscoveryPrinter　プリンターの探索開始（Bluetooth）

Bluetooth接続のプリンターを探索します。発見されたプリンターのプリンター情報は、**PrinterInfo**クラスに格納されます。
SLP721RTのみサポートします。

構文　public void **startDiscoveryPrinter**(PrinterListener *listener*) throws **PrinterException**

パラメータ　*listener*　**PrinterListener**のインスタンス
本メソッドの終了や**cancelDiscoveryPrinter**による中断は*listener*に設定されたインスタンスを経由して、**finishEvent**により、終了イベントとしてユーザアプリケーションに通知されます。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドではSII製プリンター以外も発見される場合があります。
また、すでにライブラリや他のアプリケーションでBluetooth接続が確立されているプリンターは発見されません。

本メソッドはアプリケーションのメインスレッドからは呼び出さないでください。

startDiscoveryPrinter　プリンターの探索開始（USB）

USB接続のプリンターを探索します。発見されたプリンターのプリンター情報は、**PrinterInfo**クラスに格納されます。

構文　public void **startDiscoveryPrinter**(PrinterListener *listener*, int *deviceType*)
throws **PrinterException**

パラメータ　*listener*　**PrinterListener**のインスタンス
本メソッドの終了や**cancelDiscoveryPrinter**による中断は*listener*に設定されたインスタンスを経由して、**finishEvent**により、終了イベントとしてユーザアプリケーションに通知されます。

deviceType　ポート種別
PRINTER_TYPE_USBを指定してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドはSII製プリンターを探索します。発見したプリンターのプリンター情報は、後述する**PrinterInfo**クラスに格納されます。

startDiscoveryPrinter　プリンターの探索開始（TCP/IP）

同じネットワークに接続されているSII製プリンターを探索します。

構文　public void **startDiscoveryPrinter**(PrinterListener *listener*, int *retry*, int *timeout*)
throws **PrinterException**

パラメータ　*listener*　**PrinterListener**のインスタンス
本メソッドの終了や**cancelDiscoveryPrinter**による中断は*listener*に設定されたインスタンスを経由して、**finishEvent**により、終了イベントとしてユーザアプリケーションに通知されます。

retry　リトライ回数（回）
*retry*で設定した回数だけローカルブロードキャストパケットを送信します。
有効範囲は、1～5です。
1未満の値を指定した場合は1回に設定されます。
5を超える値を指定した場合は5回に設定されます。

timeout　探索タイムアウト時間（ミリ秒：ms）
探索1回あたりのタイムアウト時間を設定します。ローカルブロードキャストパケットの送信毎に、*timeout*で指定した時間が経過するまでプリンターからの応答を待ちます。
有効範囲は、3000～60000です。
3000未満の値を指定した場合は3000msに設定されます。
60000を超える値を指定した場合は60000msに設定されます。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　本メソッドは、SII製プリンターを探索します。発見したプリンターのプリンター情報は、後述する**PrinterInfo**クラスに格納されます。

cancelDiscoveryPrinter　プリンターの探索中断

実行中の**startDiscoveryPrinter**を中断します。

構文　public void **cancelDiscoveryPrinter**()

説明　本メソッドによる中断は、**startDiscoveryPrinter**の*listener*に設定されたインスタンスを経由して、**finishEvent**により、終了イベントとしてユーザアプリケーションに通知されます。

getFoundPrinter　発見されたプリンター情報リストの取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報を、格納先の**PrinterInfo**クラスからArrayListで取得します。

構文　public ArrayList<**PrinterInfo**> **getFoundPrinter**()

戻り値　**PrinterInfo**クラスのArrayList

getSendTimeout　送信タイムアウト時間の取得

データ送信時のタイムアウト時間を取得します。

構文　public int **getSendTimeout**()

戻り値　送信タイムアウト時間（ミリ秒：ms）

説明　本メソッドは**isConnect**がtrue/falseに関わらず取得が可能です。

setSendTimeout　送信タイムアウト時間の設定

データ送信時のタイムアウト時間を設定します。

構文　public void **setSendTimeout**(int *sendTimeout*)

パラメータ　*sendTimeout*　送信タイムアウト時間（ミリ秒：ms）
有効範囲は、100～90000です。
有効範囲外の値を指定した場合は10000msに設定されます。

説明　本メソッドで送信タイムアウト時間の設定を行っていない場合の値は10000となります。

本メソッドは**isConnect**がtrue/falseに関わらず設定が可能です。

設定したタイムアウト時間が有効になるのは、次回データ送信時となります。

getReceiveTimeout　受信タイムアウト時間の取得

データ受信時のタイムアウト時間を取得します。

構文　public int **getReceiveTimeout**()

戻り値　受信タイムアウト時間（ミリ秒：ms）

説明　本メソッドは**isConnect**がtrue/falseに関わらず取得が可能です。

setReceiveTimeout　受信タイムアウト時間の設定

データ受信時のタイムアウト時間を設定します。

構文　public void **setReceiveTimeout**(int *receiveTimeout*)

パラメータ　*receiveTimeout*　受信タイムアウト時間（ミリ秒：ms）
有効範囲は、100～90000です。
有効範囲外の値を指定した場合は10000msに設定されます。

説明　本メソッドで受信タイムアウト時間の設定を行っていない場合の値は10000となります。

本メソッドは**isConnect**がtrue/falseに関わらず設定が可能です。

設定したタイムアウト時間が有効になるのは、次回データ受信時となります。

getInternationalCharacter　国際文字セットの取得

国際文字セットの値を取得します。

構文　public int **getInternationalCharacter**()

戻り値　値については、「4.3.1(2)③ 国際文字セット」を参照してください。

説明　**sendText**、**sendTextEx**、及び**sendDataFile**にてテキストデータを送信した場合は、下記に示す文字コードの印字結果が異なります。印刷される文字に関しては、「付録A 文字セット」を参照してください。

国際文字セット設定により印字結果が異なる文字コード
0x23、0x24、0x40、0x5B、0x5C、0x5D、0x5E、0x60、0x7B、0x7C、0x7D、0x7E

setInternationalCharacter　国際文字セットの設定

国際文字セットの値を設定します。

構文　public void **setInternationalCharacter**(int *internationalCharacter*)

パラメータ　*internationalCharacter*　国際文字セット定数
設定可能な値は、「4.3.1(2)③ 国際文字セット」を参照してください。
無効な値を指定した場合は無視されます。

説明　本メソッドで国際文字セットの設定を行っていない場合は、Androidデバイスの言語設定により下記の国際文字セットとなります。

Androidデバイスの言語設定が日本の場合：　**COUNTRY_JAPAN**
Androidデバイスの言語設定が日本以外の場合：**COUNTRY_USA**

getCodePage　コードページの取得

コードページの値を取得します。

構文　public int **getCodePage**()

戻り値　値については、「4.3.1(2)④ コードページ」を参照してください。

説明　**sendText**、**sendTextEx**、及び**sendDataFile**にてテキストデータを送信する場合に利用するエンコーダが変更されます。印刷される文字に関しては、「付録A 文字セット」を参照してください。

setCodePage　コードページの設定

コードページの値を設定します。

構文　public void **setCodePage**(int *codePage*)

パラメータ　*codePage*　コードページ定数
設定可能な値は、「4.3.1(2)④ コードページ」を参照してください。
無効な値を指定した場合は無視されます。

説明　本メソッドでコードページの設定を行っていない場合は、Androidデバイスの言語設定により下記のコードページとなります。
Androidデバイスの言語設定が日本の場合：　**CODE_PAGE_KATAKANA**
Androidデバイスの言語設定が日本以外の場合：**CODE_PAGE_1252**

getPrinterModel　プリンターモデルの取得

通信中のプリンターモデルの値を取得します。

構文　public int **getPrinterModel**()

戻り値　値については、「4.3.1(2)① プリンターモデル」を参照してください。
isConnectがfalseの場合は、**PRINTER_MODEL_DEFAULT**が返ります。

説明　プリンターが未接続でも、一度**connect**が成功している場合は、前回接続に成功したプリンターモデル値が返ります。

getPortType　接続中ポート種別の取得

プリンターとの接続に利用しているポート種別の値を取得します。

構文　public int **getPortType**()

戻り値　値については、「4.3.1(2)⑤ ポート種別」を参照してください。
isConnectがfalseの場合は、**PRINTER_TYPE_BLUETOOTH**が返ります。

説明　プリンターが未接続でも、一度**connect**が成功している場合は、前回接続に成功したポート種別の値が返ります。

isConnect　プリンターとの通信状態の確認

プリンターとの通信状態を確認します。

構文　public boolean **isConnect**()

戻り値　true　プリンターと通信中
false　プリンターと未接続

説明　データ送信に失敗すると、プリンターとの通信を終了し、本メソッドはfalseになります。falseの場合は、**connect**で再度通信を開始してください。

getSocketKeepingTime　ソケット維持時間の取得

ソケット維持時間を取得します。

構文　public int **getSocketKeepingTime**()

戻り値　ソケット維持時間（ミリ秒：ms）

説明　本メソッドは**isConnect**がtrue/falseに関わらず取得が可能です。

setSocketKeepingTime　ソケット維持時間の設定

ソケット維持時間を設定します。

構文　public void **setSocketKeepingTime**(int *socketKeepingTime*)

有効範囲　60000～300000（ミリ秒：ms）
60000未満の値を指定した場合は60000msに設定されます。
300000を超える値を指定した場合は300000msに設定されます。

初期値　300000

説明　本メソッドは**isConnect**がtrue/falseに関わらず設定が可能です。

ソケット維持時間には、接続するプリンターの受信タイムアウト時間設定と等しい時間を指定してください。受信タイムアウト時間の設定は、Google PlayのAndroidアプリ「SII Printer Utility」で変更できます。

設定したソケット維持時間が有効になるのは、次回**connect**実行時となります。

getPrintSmartLabelMode　ラベルファイル印刷時の用紙の取得

ラベルファイル印刷時の用紙を取得します。

構文　public int **getPrintSmartLabelMode**()

戻り値　値については、「4.3.1(2)⑥ ラベルファイル印刷時マークあり・なし用紙選択」を参照してください。

説明　　本メソッドは**isConnect**がtrue/falseに関わらず取得が可能です。

setPrintSmartLabelMode　ラベルファイル印刷時の用紙の設定

ラベルファイル印刷時の用紙を設定します。

構文　　public void **setPrintSmartLabelMode**(int *paperMode*)

パラメータ　*paperMode*　　ラベルファイル印刷時マークあり・なし用紙選択
設定可能な値は、「4.3.1(2)⑥ ラベルファイル印刷時マークあり・なし用紙選択」を参照してください。
無効な値を指定した場合は無視されます。

説明　　マーク紙を指定した場合、**printSmartLabelImageData**実行時に用紙の逆方向紙送りを伴う印字開始位置への紙送りを行います。

マークなし用紙を指定した場合、**printSmartLabelImageData**実行時に印字開始位置への紙送りは行いません。

getVersion　SDKバージョンの取得

SDKのバージョンを文字列で取得します。

構文　　public String **getVersion**()

戻り値　　SDKバージョン文字列（例：SDKバージョンがVer.1.0.0の場合、戻り値は”1.0.0”となります）

説明　　本メソッドは**isConnect**がtrue/falseに関わらず取得が可能です。

printSmartLabelImageData　ラベルの印字

ラベルを印字します。

構文　　public void **printSmartLabelImageData**(SmartLabelManager *labelManager)*
throws **PrinterException**

パラメータ　*labelManager*　　**SmartLabelManager**クラスのインスタンス

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　　本メソッドを利用した印字例については、「**4.3.9 SmartLabelManagerクラス**」を参照してください。

controlTransaction　一括処理の開始・終了

一括処理を開始、または終了します。

構文　public void **controlTransaction**(TransactionFunction *transactionFunction)*
throws **PrinterException**

パラメータ　*transactionFunction*　一括処理の選択
利用可能な定数は、「4.3.1(3)㉖ 一括処理選択（TransactionFunction）」を参照してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。
データ送信に失敗すると、プリンターとの通信を終了し、**PrinterException**をスローすることがあります。プリンターとの通信状態の確認は、**isConnect**を参照してください。

説明　一括処理の手順は下記の通りです。

① 一括処理を開始します。
TRANSACTION_STARTを指定してください。

② メソッドを実行します。
一括処理対象メソッドの場合、送信データのバッファリングを開始します。
バッファリング中に実行した一括処理対象メソッドの送信データは、プリンターへ送信せずに送信バッファーにバッファリングします。
バッファリングできる送信データの最大サイズはシステムに依存します。
バッファリングされた送信データが最大サイズを超える場合、超えた時点の一括対象メソッドがエラーとなります。エラーとなった場合は、エラー前までの送信データが保持されます。
保持された送信データは、手順③で一括処理を終了してください。

一括処理対象以外のメソッドの場合は、送信データはバッファリングされずに即実行されます。

③ 一括処理を終了します。
TRANSACTION_PRINTを指定するとバッファリングされた送信データをプリンターへ送信します。バッファリングされた送信データはプリンター送信後も保持されます。
保持されている送信データは下記のいずれかにより破棄されます。
・**TRANSACTION_CLEAR**の指定
・**TRANSACTION_START**の指定
・**disconnect**の実行

一括処理対象メソッドを下記に示します。
・**sendText**
・**sendTextEx**
・**printBarcode**
・**printPDF417**
・**printQRcode**
・**printDataMatrix**
・**printMaxiCode**
・**printGS1DataBarStacked**
・**printGS1DataBarStackedOmnidirectional**

・printGS1DataBarExpandedStacked
・cutPaper
・openDrawer
・externalBuzzer
・feedPosition
・sendBinary
・sendDataFile
・sendImage
・printPDF
・printLogo[*1]
・printSmartLabelImageData

*1: 一括処理中のメソッドは登録済みのロゴが存在しない場合でもエラーを通知しません。

4.3.2 PrinterEventクラス

PrinterEventクラスは、**startDiscoveryPrinter**が終了した際に発生する終了イベントを取得するクラスです。

(1) メソッド一覧

PrinterEventクラスで提供されるメソッドを下記に示します。

名前	説明
getEventType	終了イベントの取得

(2) 終了イベント定数

終了イベントの取得で利用する定数を下記に示します。

定数名	説明	値
EVENT_FINISHED_DISCOVERY	**startDiscoveryPrinter**の終了	1
EVENT_CANCELED_DISCOVERY	**cancelDiscoveryPrinter**による中断	2

(3) メソッド詳細

getEventType 終了イベントの取得

startDiscoveryPrinterが終了した際の終了イベントを取得します。

構文　public int **getEventType**()

戻り値　値については、「4.3.2(2) 終了イベント定数」を参照してください。

説明　終了イベントにより、**startDiscoveryPrinter**が終了したのか、**cancelDiscoveryPrinter**により中断されたのかを判定できます。
プリンターが発見できなかった場合も**EVENT_FINISHED_DISCOVERY**が返ります。

4.3.3 PrinterListenerインターフェイス

PrinterListenerインターフェイスは、**startDiscoveryPrinter**が終了した際の終了イベントを取得するためのインターフェイスです。

(1) メソッド一覧

PrinterListenerインターフェイスのメソッドを下記に示します。

名前	説明
finishEvent	プリンター探索の終了イベント

(2) メソッド詳細

finishEvent プリンター探索の終了イベント

startDiscoveryPrinterが終了した場合、または**cancelDiscoveryPrinter**が実行された場合に呼び出される終了イベントです。

構文 public void **finishEvent**(**PrinterEvent** *event*)

パラメータ *event* 終了イベント
PrinterEventクラスで指定されます。

説明 インターフェイスのため実装はありません。
本メソッドを**startDiscoveryPrinter**の終了や**cancelDiscoveryPrinter**による中断によって、終了イベントの通知を受け取るユーザアプリケーションに実装してください。**PrinterEvent**クラスの**getEventType**により、終了イベントの種別を判定してください。

4.3.4 PrinterInfoクラス

PrinterInfoクラスは、**startDiscoveryPrinter**で発見されたプリンターのプリンター情報を格納するクラスです。

(1) メソッド一覧

プリンターモデル名（Bluetoothデバイス名）、Bluetoothアドレス、MACアドレス、IPアドレス、ポート名（デバイスパス）及びペアリング状態を取得できます。**PrinterInfo**クラスで提供されるメソッドの一覧を下記に示します。

名前	説明
getPrinterModelName	プリンターモデル名の取得
getBluetoothAddress	Bluetoothアドレスの取得
getMacAddress	MACアドレスの取得
getIpAddress	IPアドレスの取得
getIsBonded	ペアリング状態の取得
getDevicePath	デバイスパスの取得

(2) メソッド詳細

getPrinterModelName プリンターモデル名の取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報からプリンターモデル名（Bluetoothデバイス名）の文字列を取得します。
SLP721RTのみサポートします。

構文　public String **getPrinterModelName**()

戻り値　プリンターモデル名（Bluetoothデバイス名）

getBluetoothAddress Bluetoothアドレスの取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報からBluetoothアドレスの文字列を取得します。
SLP721RTのみサポートします。

構文　public String **getBluetoothAddress**()

戻り値　Bluetoothアドレス

getMacAddress MACアドレスの取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報からMACアドレスの文字列を取得します。

構文　public String **getMacAddress**()

戻り値　MACアドレス

getIpAddress　IPアドレスの取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報からIPアドレスの文字列を取得します。

構文　public String **getIpAddress**()

戻り値　IPアドレス

getIsBonded　ペアリング状態の取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報からペアリング状態を取得します。
SLP721RTのみサポートします。

構文　public boolean **getIsBonded**()

戻り値　true　ペアリング済み
　　false　未ペアリング

getDevicePath　デバイスパスの取得

startDiscoveryPrinterで発見されたプリンターのプリンター情報からUSBデバイスファイルのパスの文字列を取得します。

構文　public String **getDevicePath**()

戻り値　デバイスパス

4.3.5 PrinterIdentifierクラス

PrinterIdentifierクラスは、**getPrinterResponse**で取得されたプリンターのプリンタIDを格納するクラスです。

(1) メソッド一覧

機種ID、タイプID、ROMバージョンID、ファームウェアバージョン(main)、メーカ名、モデル名、シリアル番号、ファームウェアバージョン(boot)、ファームウェアチェックサム(boot)、ファームウェアチェックサム(main)及びファームウェアチェックサム(main + boot)を取得できます。**PrinterIdentifier**クラスで提供されるメソッドの一覧を下記に示します。

名前	説明
getPrinterModelID	機種IDの取得
getTypeID	タイプIDの取得
getRomVersionID	ROMバージョンIDの取得
getMainFirmwareVersion	ファームウェアバージョン(main)の取得
getManufacturer	メーカ名の取得
getModelName	モデル名の取得
getSerialNumber	シリアル番号の取得
getBootFirmwareVersion	ファームウェアバージョン(boot)の取得
getBootFirmwareChecksum	ファームウェアチェックサム(boot)の取得
getMainFirmwareChecksum	ファームウェアチェックサム(main)の取得
getFirmwareChecksum	ファームウェアチェックサム(main + boot)の取得

(2) メソッド詳細

getPrinterModelID 機種IDの取得

getPrinterResponseで取得されたプリンターのプリンタIDから機種IDを取得します。

構文　public byte **getPrinterModelID**()

戻り値　機種ID

getTypeID タイプIDの取得

getPrinterResponseで取得されたプリンターのプリンタIDからタイプIDを取得します。

構文　public byte **getTypeID**()

戻り値　タイプID

getRomVersionID ROMバージョンIDの取得

getPrinterResponseで取得されたプリンターのプリンタIDからROMバージョンIDを取得します。

構文　public byte **getRomVersionID**()

戻り値　　ROMバージョンID

getMainFirmwareVersion　ファームウェアバージョン(main)の取得

getPrinterResponseで取得されたプリンターのプリンタIDからファームウェアバージョン(main)の文字列を取得します。

構文　　public String **getMainFirmwareVersion**()

戻り値　　ファームウェアバージョン(main)

getManufacturer　メーカ名の取得

getPrinterResponseで取得されたプリンターのプリンタIDからメーカ名の文字列を取得します。

構文　　public String **getManufacturer**()

戻り値　　メーカ名

getModelName　モデル名の取得

getPrinterResponseで取得されたプリンターのプリンタIDからモデル名の文字列を取得します。

構文　　public String **getModelName**()

戻り値　　モデル名

getSerialNumber　シリアル番号の取得

getPrinterResponseで取得されたプリンターのプリンタIDからシリアル番号の文字列を取得します。

構文　　public String **getSerialNumber**()

戻り値　　シリアル番号

getBootFirmwareVersion　ファームウェアバージョン(boot)の取得

getPrinterResponseで取得されたプリンターのプリンタIDからファームウェアバージョン(boot)の文字列を取得します。

構文　　public String **getBootFirmwareVersion**()

戻り値　　ファームウェアバージョン(boot)

getBootFirmwareChecksum　ファームウェアチェックサム(boot)の取得

getPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(boot)を取得します。

構文　public byte[] **getBootFirmwareChecksum**()

戻り値　ファームウェアチェックサム(boot)

getMainFirmwareChecksum　ファームウェアチェックサム(main)の取得

getPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(main)を取得します。

構文　public byte[] **getMainFirmwareChecksum**()

戻り値　ファームウェアチェックサム(main)

getFirmwareChecksum　ファームウェアチェックサム(main + boot)の取得

getPrinterResponseで取得されたプリンターのプリンタIDからファームウェアチェックサム(main + boot)を取得します。

構文　public byte[] **getFirmwareChecksum**()

戻り値　ファームウェアチェックサム(main + boot)

4.3.6 PrinterExceptionクラス

(1) メソッド一覧

PrinterExceptionクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
PrinterException	コンストラクタ
getErrorCode	エラーコードの取得

(2) 定数一覧

① エラーコード

エラーコードの取得で利用される定数を下記に示します。

定数名	説明	値
ERROR_ACCESS_DENIED	ハンドルの取得に失敗しました。*1	-1
	利用できないポートが指定されました。	
	サポートされていないメソッドが指定されました。	
ERROR_SHARING_VIOLATION	すでにオープン済みのポートが指定されました。	-11
ERROR_PORT_NOT_OPENED	ポートがオープンされていません。	-12
ERROR_DEVICE_NOT_CONNECTED	Androidデバイスとプリンターの接続に問題があります。	-21
ERROR_DEVICE_INITIALIZE_FAILED	プリンターの設定変更に失敗しました。プリンターへのデータ送信が送信タイムアウト時間内に完了してないか、プリンターからのデータ受信が受信タイムアウト時間内に完了していません。	-31
ERROR_DATA_SIZE_ZERO	データサイズが0バイトのデータが指定されました。	-101
ERROR_OVER_MAX_DATA_SIZE	データサイズが最大値を超過しています。	-102
ERROR_ENCODE_FAILED	テキストデータのエンコードでエラーが発生しました。*1	-111
ERROR_TIMEOUT	送信タイムアウトが発生しました。	-201
	受信タイムアウトが発生しました。	
ERROR_FILE_NOT_FOUND	指定されたファイルが存在しません。	-301
ERROR_FILE_USED	指定されたファイルは別プロセスで使用中です。	-302
ERROR_FILE_INVALID	指定されたファイルは無効です。	-303
ERROR_LOW_MEMORY	イメージファイルの読み込みでメモリ不足が発生しました。	-311
ERROR_OVER_MAX_IMAGE	イメージファイルの幅と高さのいずれか、または両方が印字可能な最大ドット数を超過しています。	-312
ERROR_LOGO_NOT_DEFINED	ロゴが登録されていません。	-313
ERROR_LOW_USER_AREA	ユーザ領域の残り容量が不足しています。	-401
ERROR_LOW_EXTERNAL_RAM	RAMの残り容量が不足しています。	-402
ERROR_LABEL_FILE_NOT_SELECTED	ラベルファイルが選択されていません。	-521
ERROR_GET_LABEL_IMAGE	ラベルイメージの作成に失敗しました。	-522

定数名	説明	値
ERROR_INVALID_PARAM	指定されたパラメータは無効です。	-9999

*1: 異常処理の可能性があります。

(3) メソッド詳細

PrinterException コンストラクタ

com.seikoinstruments.sdk.thermalprinter.PrinterExceptionクラスのコンストラクタです。

構文　public **PrinterException**(int *code*, String *message*)

getErrorCode エラーコードの取得

スローされた例外のエラーコードを取得します。

構文　public int **getErrorCode**()

戻り値　エラーコードの内容は、「4.3.6(2) 定数一覧」を参照してください。

4.3.7 CallbackFunctionListenerインターフェイス

CallbackFunctionListenerインターフェイスは、プリンターステータスの変化イベントを取得するためのインターフェイスです。

(1) メソッド一覧

CallbackFunctionListenerインターフェイスのメソッドを下記に示します。

名前	説明
onStatusChanged	プリンターステータスの変化イベント

(2) メソッド詳細

onStatusChanged プリンターステータスの変化イベント

構文 public void **onStatusChanged**(int *status*)

パラメータ *status* プリンターステータス

説明 本メソッドは下記のタイミングで呼び出されます。
- **setCallbackFunctionListener**が実行された場合
- プリンターステータスが変化した場合

プリンターステータスの変化イベントは、**isConnect**がtrueの場合に通知されます。

インターフェイスのため実装はありません。
プリンターステータス変化のコールバックを受け取るクラスで任意の処理を実装してください。

本メソッド内で**PrinterManager**のAPIを実行しないでください。

4.3.8 BarcodeScannerListenerインターフェイス

BarcodeScannerListenerインターフェイスは、バーコードスキャナーの接続、バーコードスキャナーの切断、または受信したバーコードデータを取得するためのインターフェイスです。

SLP720RT/SLP721RTは、本インターフェイスをサポートしていません。

4.3.9 SmartLabelManagerクラス

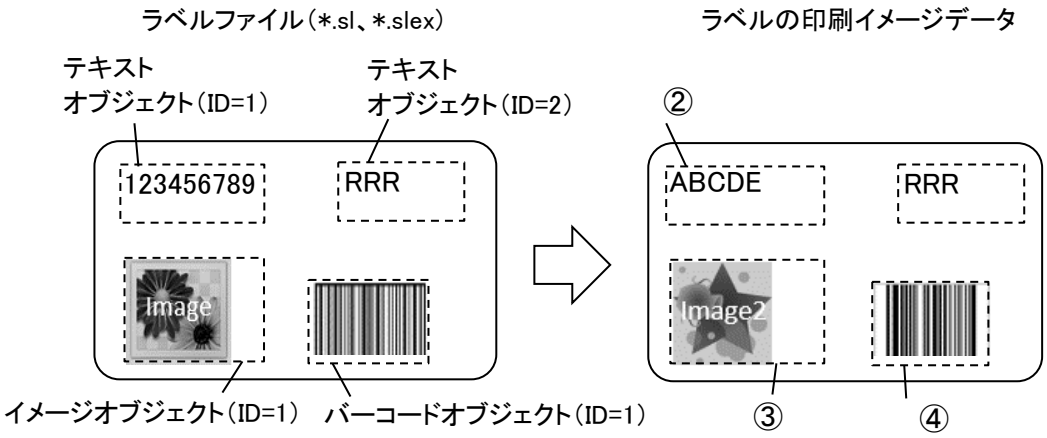
SmartLabelManagerクラスは、アプリで作成したラベルファイル（*.sl、*.slex）を、プリンターで印字可能なデータに変換する機能を提供するクラスです。

(1) メソッド一覧

SmartLabelManagerクラスで提供されるメソッドの一覧を下記に示します。

名前	説明
SmartLabelManager	コンストラクタ
selectSmartLabelFile	ラベルファイルの指定
replaceSmartLabelTextData	ラベルのテキストデータの置換
replaceSmartLabelImageData	ラベルのイメージデータの置換
replaceSmartLabelBarcodeData	ラベルのバーコードデータの置換

ラベルファイルを利用してデータを置き換えし、印字する手順例を下記に示します。



① 印字またはデータの置き換えをするラベルファイルを指定します。

```
smartLabelManager.selectSmartLabelFile(labelfilePath);
```

② テキストデータを置き換えます。

```
smartLabelManager.replaceSmartLabelTextData(1, "ABCDE");
```

③ イメージデータを置き換えます。

```
smartLabelManager.replaceSmartLabelImageData(1, bitmap1);
```

④ バーコードデータを置き換えます。

```
smartLabelManager.replaceSmartLabelBarcodeData(1, "123456789");
```

⑤ ラベルを印字します。

```
printerManager.printSmartLabelImageData(smartLabelManager);
```

(2) メソッド詳細

SmartLabelManager　コンストラクタ

com.seikoinstruments.sdk.thermalprinter.SmartLabelManagerクラスのコンストラクタです。

構文　public **SmartLabelManager**(Context *context*)

パラメータ　*context*　本メソッドを呼び出すアプリケーションのコンテキストを指定してください。
例： **MainActivity.this**

selectSmartLabelFile　ラベルファイルの指定

ラベルファイル（*.sl、*.slex）を指定します。

構文　public void **selectSmartLabelFile**(String *filePath)* throws **PrinterException**

パラメータ　*filePath*　使用するラベルファイル（*.sl、*.slex）のファイルパス（*.slは非推奨）
入力可能な形式について下記に説明します。

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列
  Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。
  詳細については、「3.5 注意事項 – 対象範囲別ストレージについて」を参照してください。

- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
  ・file://
  ・content://
  本パラメータには"ストレージ アクセス フレームワーク"から取得したURI文字列を指定する必要があります。"ストレージ アクセス フレームワーク"から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

説明　指定したラベルファイル（*.sl、*.slex）を内部で保持します。
ラベルファイル指定後は、各オブジェクトのデータの置き換えができます。

使用できるラベルファイルには制限があります。制限については、「0 ラベルファイルの構造」を参照してください。

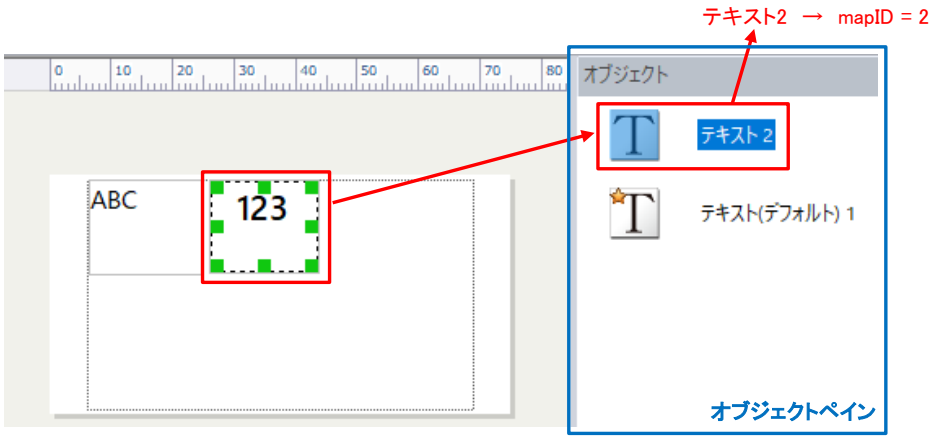
注意　ラベルファイル（*.sl）は、将来的に非サポートとなるラベルファイルです。
SII Layout Editor（アプリバージョン1.5.0以降）で作成できるラベルファイル（*.slex）をお使いください。

replaceSmartLabelTextData　ラベルのテキストデータの置換

ラベルファイル(*.sl、*.slex)のテキストオブジェクトの値を置き換えます。

構文　public void **replaceSmartLabelTextData**(int *mapID*, String *text)* throws **PrinterException**

パラメータ　*mapID*　テキストオブジェクトのID
アプリのラベルファイル(*.sl、*.slex)に配置したテキストオブジェクトのIDを指定してください。テキストオブジェクトのIDはアプリのUI画面で確認できます。
指定した*mapID*が選択中のラベルに定義されていない場合は無視されます。



Smart Label CreatorのUI画面



アプリのUI画面

text　置き換えるテキストデータ

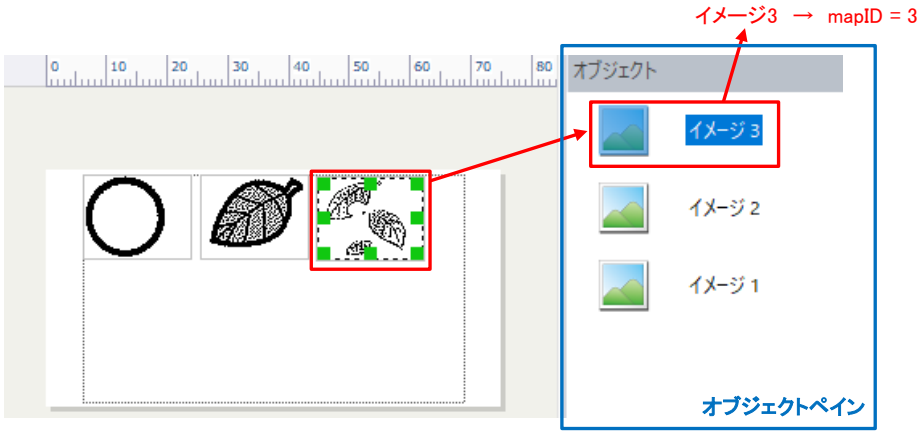
例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

replaceSmartLabelImageData　ラベルのイメージデータの置換

ラベルファイル(*.sl、*.slex)のイメージオブジェクトの値を置き換えます。

構文　public void **replaceSmartLabelImageData**(int *mapID*, Bitmap *bitmap)* throws **PrinterException**

パラメータ　*mapID*　イメージオブジェクトのID
アプリのラベルファイル(*.sl、*.slex)に配置したイメージオブジェクトのIDを指定してください。イメージオブジェクトのIDはアプリのUI画面で確認できます。
指定した*mapID*が選択中のラベルに定義されていない場合は無視されます。



Smart Label CreatorのUI画面



アプリのUI画面

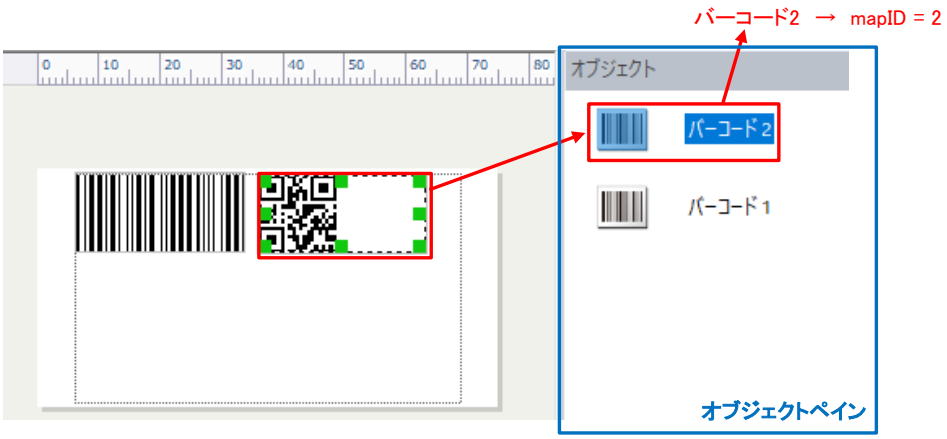
bitmap　置き換えるイメージデータ
Androidに用意されたクラス「android.graphics.Bitmap」に準するイメージデータを指定してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

ラベルファイル(*.sl、*.slex)のバーコードオブジェクトの値を置き換えます。

構文　public void **replaceSmartLabelBarcodeData**(int *mapID*, String *text)* throws **PrinterException**

パラメータ　*mapID*　バーコードオブジェクトのID
アプリのラベルファイル(*.sl、*.slex)に配置したバーコードオブジェクトのIDを指定してください。バーコードオブジェクトのIDはアプリのUI画面で確認できます。
指定した*mapID*が選択中のラベルに定義されていない場合は無視されます。



Smart Label CreatorのUI画面



アプリのUI画面

text　置き換えるテキストデータ
置き換えるテキストデータが無効なバーコードデータの場合でも、エラーにはなりません。有効なバーコードデータをご確認の上、指定してください。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。
エラーについての詳細は、「**4.3.6 PrinterExceptionクラス**」を参照してください。

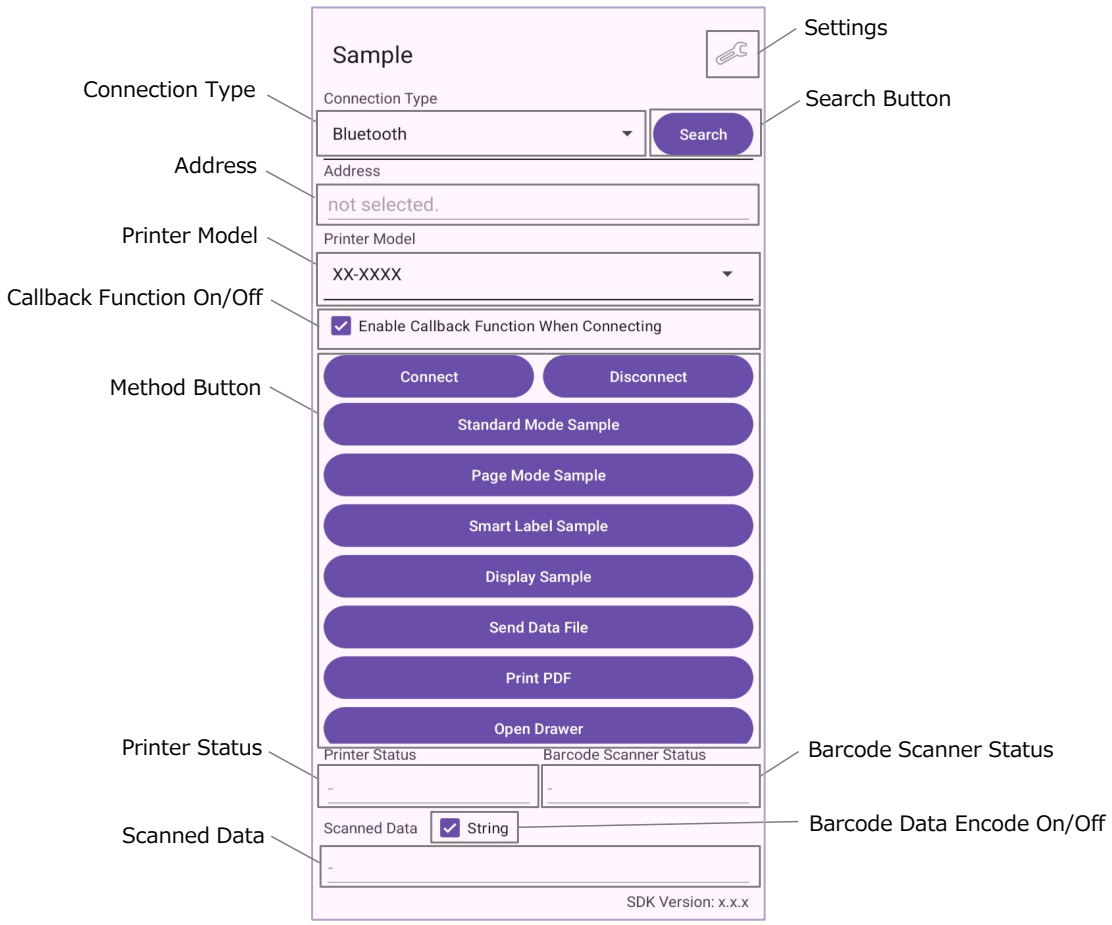
5章　サンプルプログラム

本章では、SIIプリントクラスライブラリで提供するサンプルプログラムについて説明します。

5.1　画面構成

SIIプリントクラスライブラリには、Android Studioのプロジェクト形式のサンプルプログラムが含まれています。
サンプルプログラムの画面について説明します。

5.1.1　メイン画面

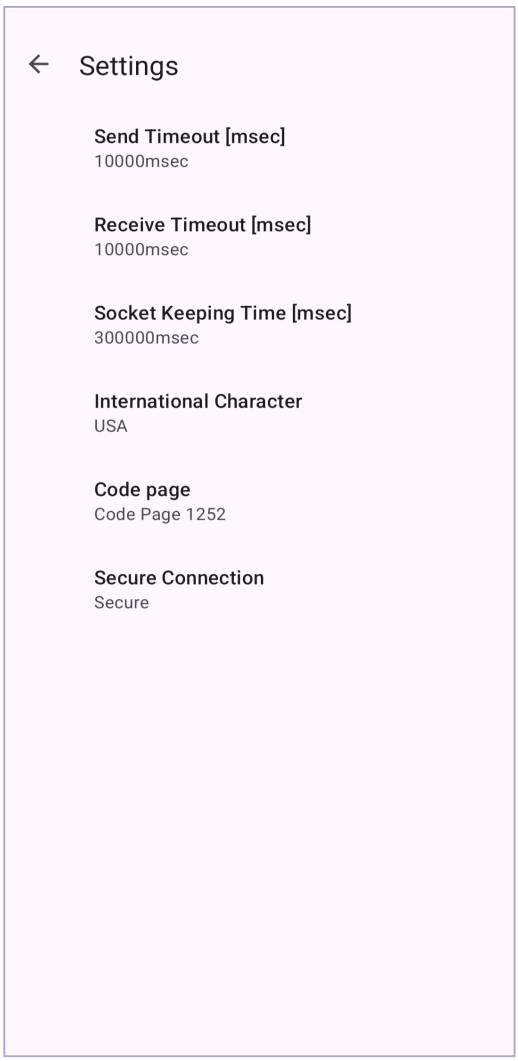


項目	説明
Settings	[Settings]ボタンをタップすると、機能設定画面が開きます。メイン画面に戻る場合は、画面左上の ← ボタンをタップしてください。
Connection type	プリンターへの接続種別を選択します。
Search Button	[Connection Type]で指定した種別のプリンターの探索を開始します。 プリンター探索画面に遷移します。探索されたプリンターは一覧に表示されます。探索済みのプリンターをタップするとプリンターが選択され、メイン画面に戻ります。
Address	選択したプリンターの情報を表示します。
Printer Model	プリンターモデルを指定します。
Callback Function On/Off	プリンターとの接続時にコールバック機能を有効にするか選択します。 On：接続時にコールバックを開始します。 Off：コールバックは応答しません。
Method Button*1	**connect**及び**disconnect**を行うメソッドボタンの他、各メソッドを組み合わせたサンプルの印字や周辺デバイスの動作確認ができます。
Printer Status	プリンターステータスを表示します。 [Callback Function On/Off]がOnの場合は、最新のプリンターステータスを表示します。
Barcode Scanner Status	バーコードスキャナーの接続状態を表示します。 SLP720RT/SLP721RTはバーコードスキャナーをサポートしていません。
Barcode Data Encode On/Off	バーコードスキャナーで読み取ったバーコードデータの変換を選択します。 SLP720RT/SLP721RTはバーコードスキャナーをサポートしていません。
Scanned Data	バーコードスキャナーで読み取ったバーコードデータを表示します。 SLP720RT/SLP721RTはバーコードスキャナーをサポートしていません。

*1: サポートする機能は機種により異なります。サポートしている機能以外は動作しません。

5.1.2 [Settings]画面

[Settings]では各種設定機能を表示します。



5.2 注意事項

サンプルプログラムは予告なしに変更される場合があります。

サンプルプログラムに対しての動作保証及びサポートは行っておりません。

付録 A　文字セット

A.1 コードページ表（文字コードテーブル）

国際文字セットでCOUNTRY_USA設定時のコードページを下記に示します。
設定する国際文字セットにより、特定の文字コードの印字結果が異なります。
特定の文字コードについては、「A.2 国際文字セット」を参照してください。

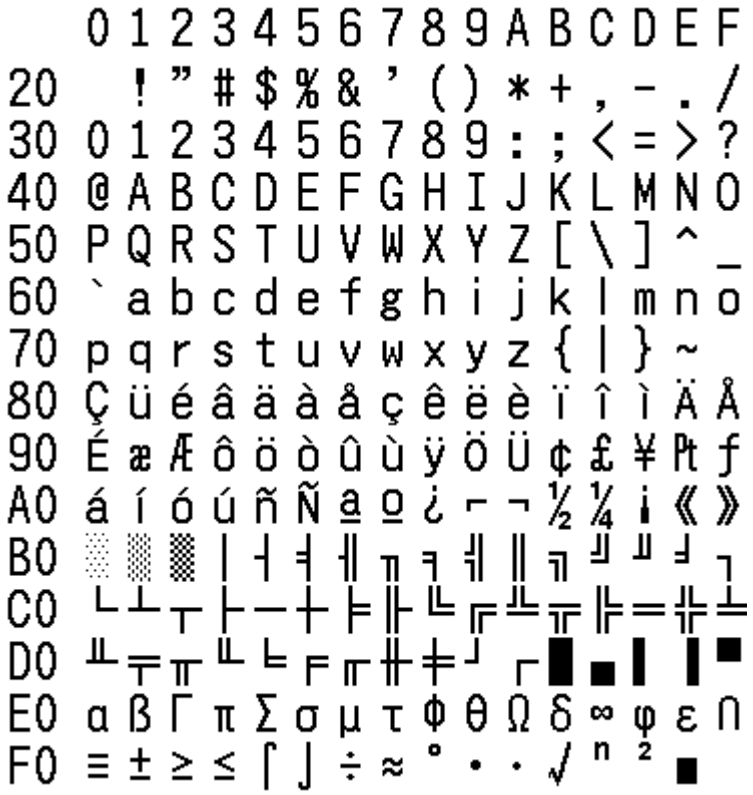


図 A-1　**CODE_PAGE_437** (USA, Standard Europe)

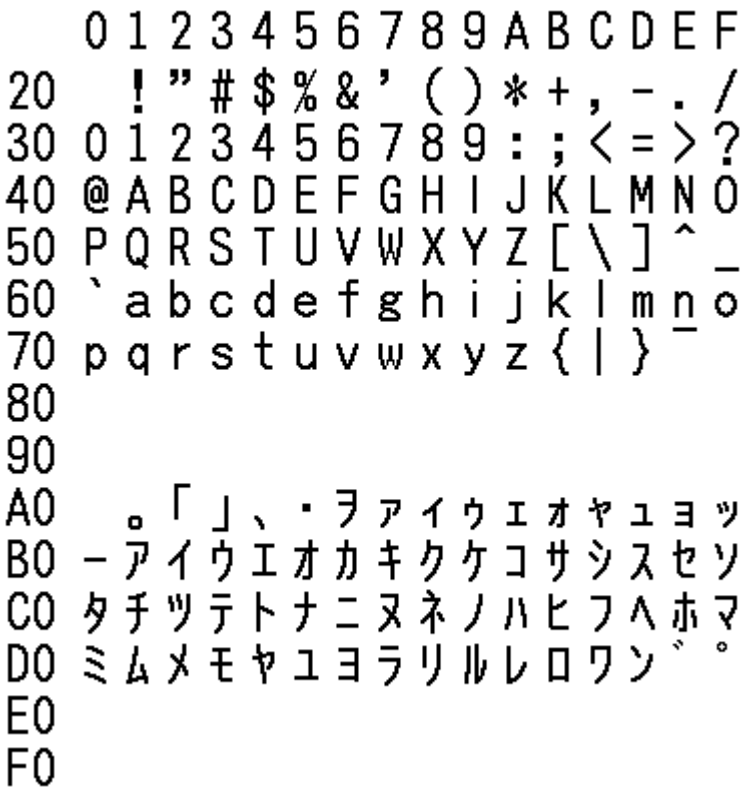


図 A-2 **CODE_PAGE_KATAKANA**

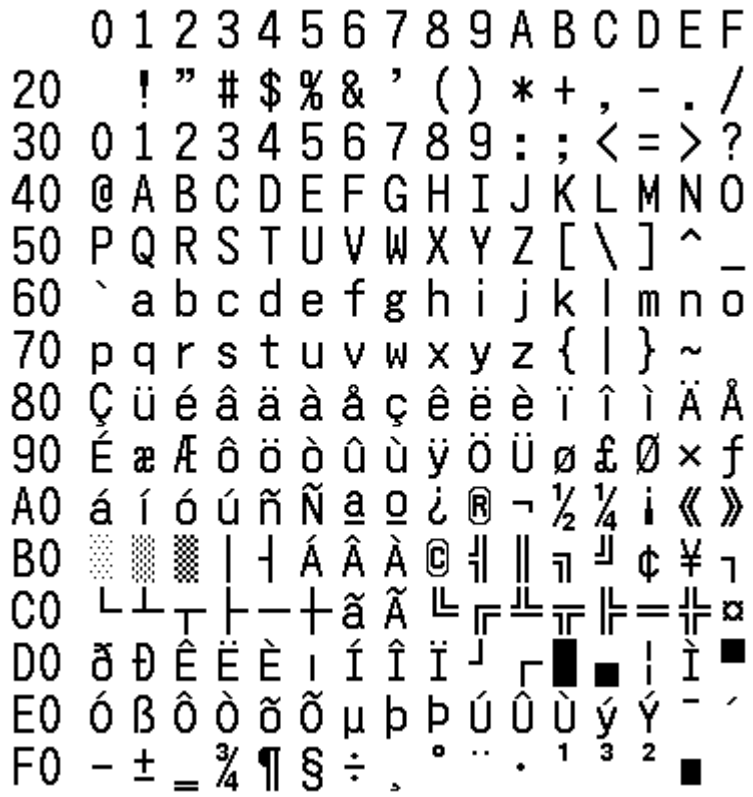


図 A-3 **CODE_PAGE_850** (Multilingual)

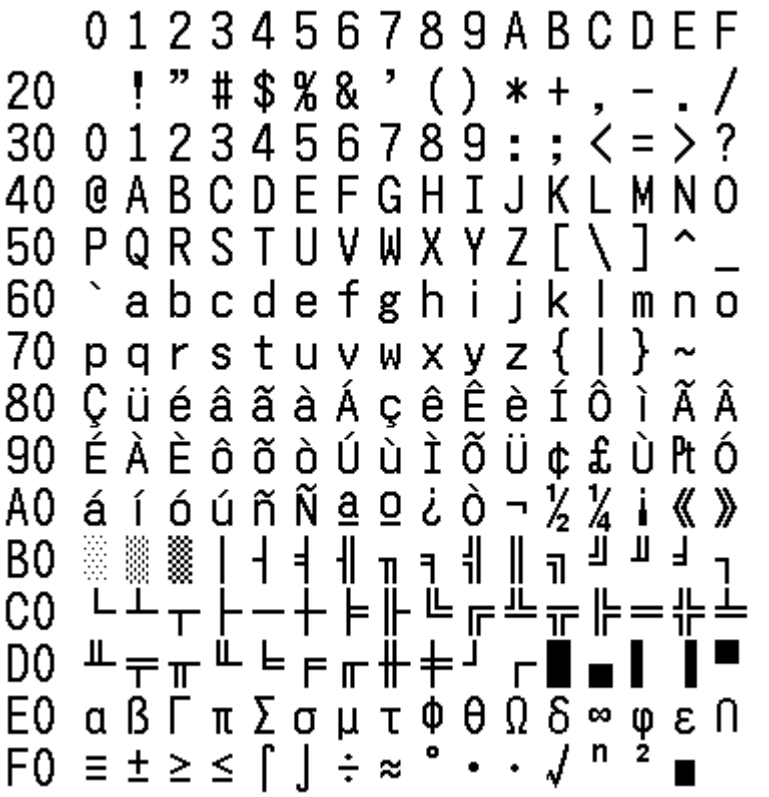


図 A-4 **CODE_PAGE_860** (Portuguese)

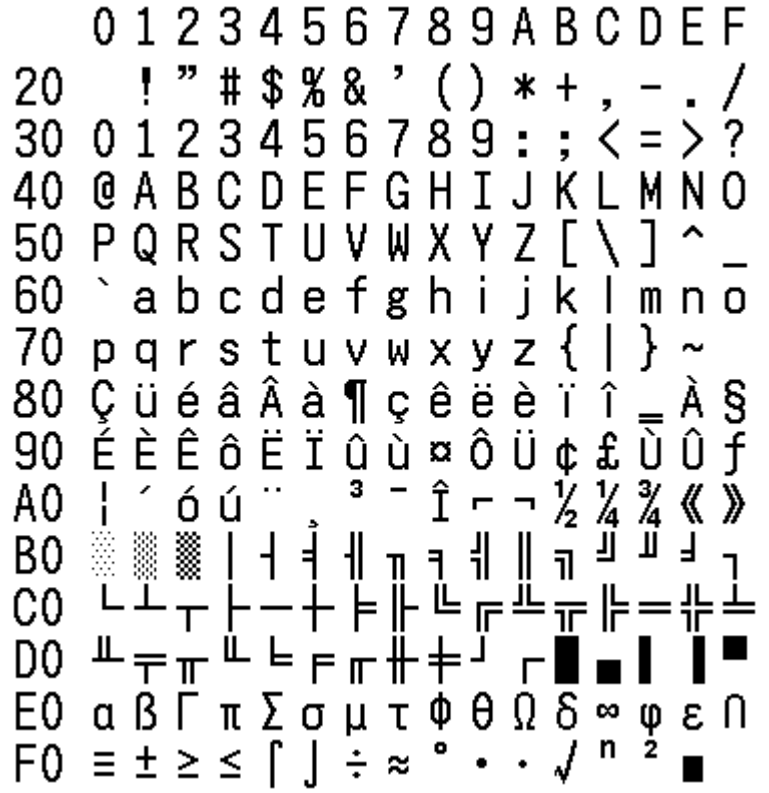


図 A-5 **CODE_PAGE_863** (Canadian-French)

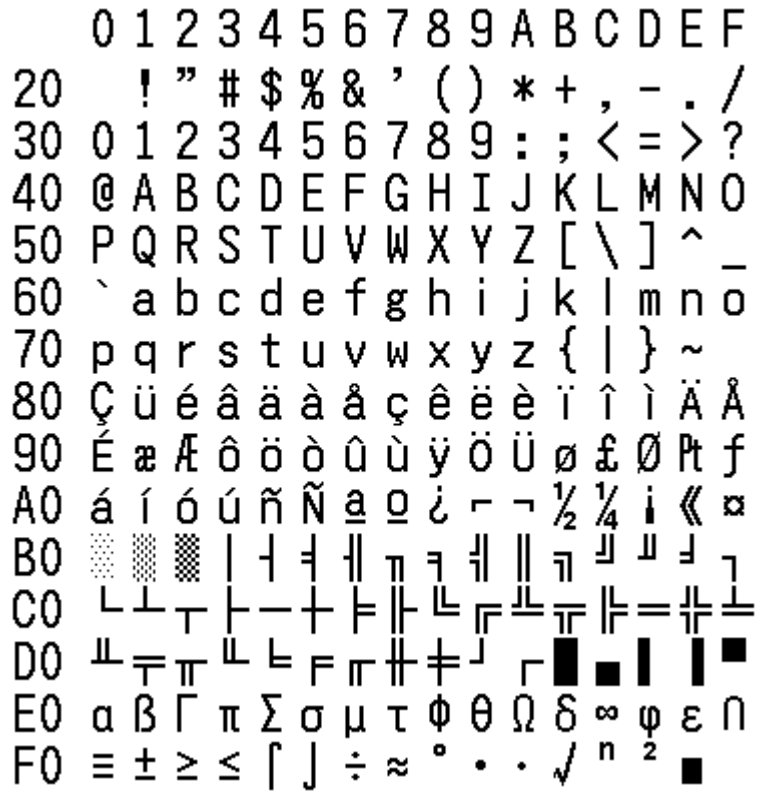


図 A-6 **CODE_PAGE_865** (Nordic)

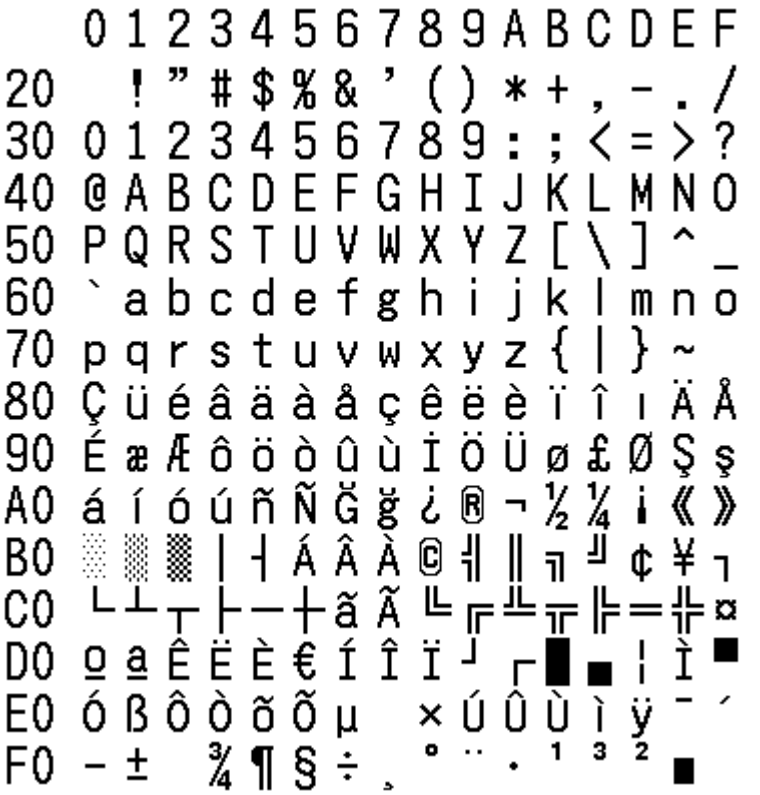


図 A-7 **CODE_PAGE_857** (Turkish)

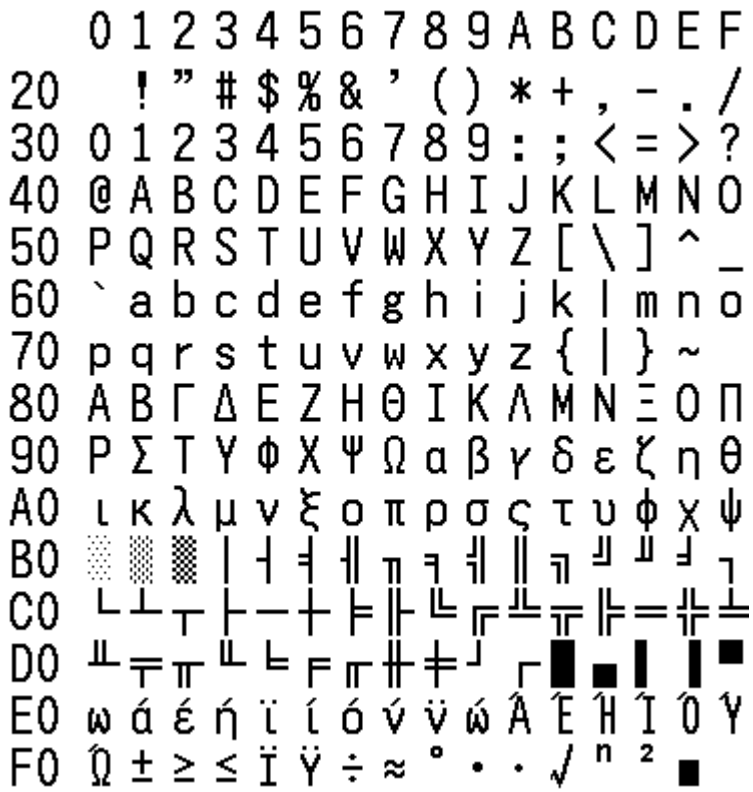


図 A-8 **CODE_PAGE_737** (Greek)

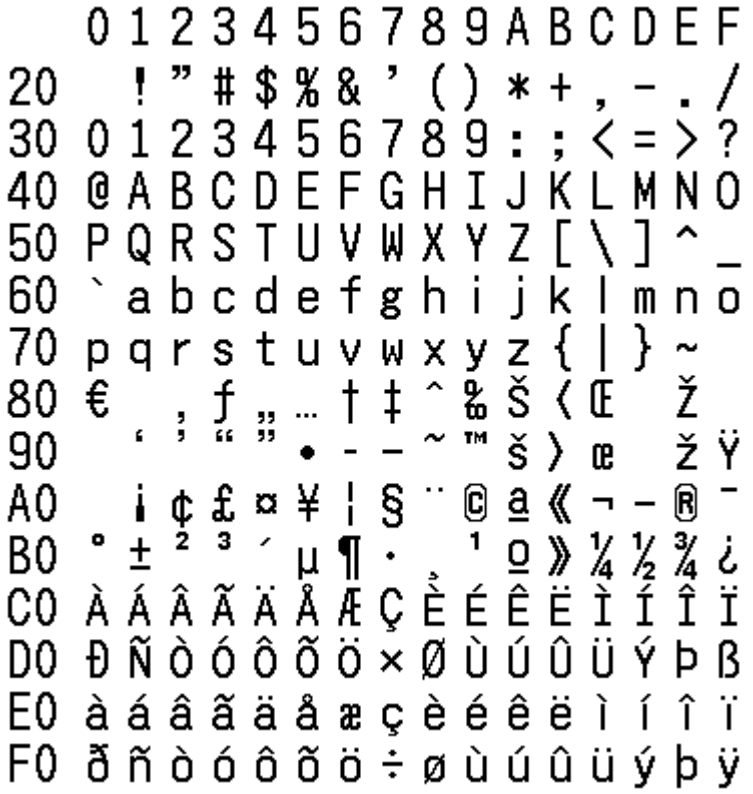


図 A-9 **CODE_PAGE_1252** (Latin)

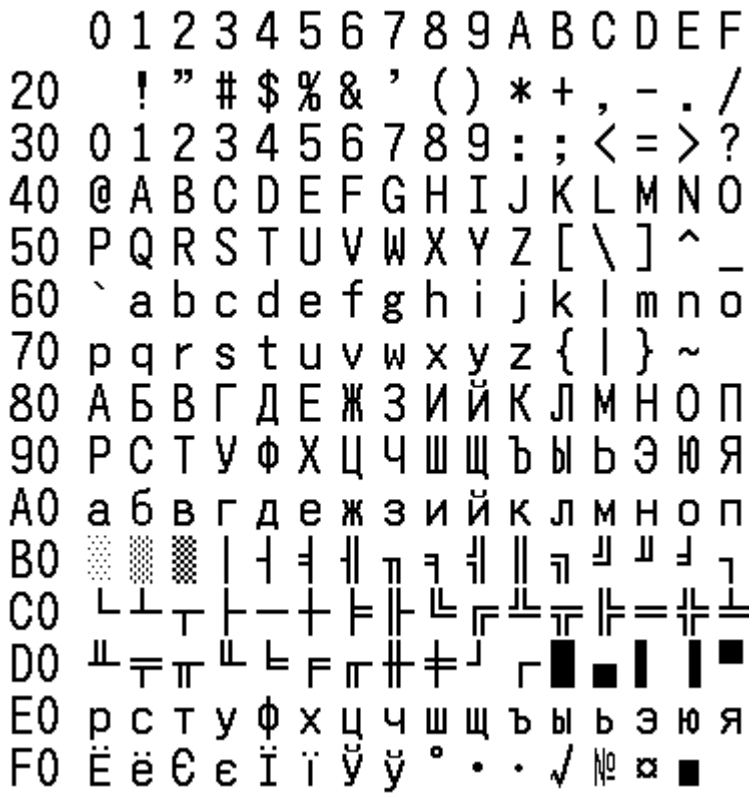


図 A-10 **CODE_PAGE_866** (Russian)

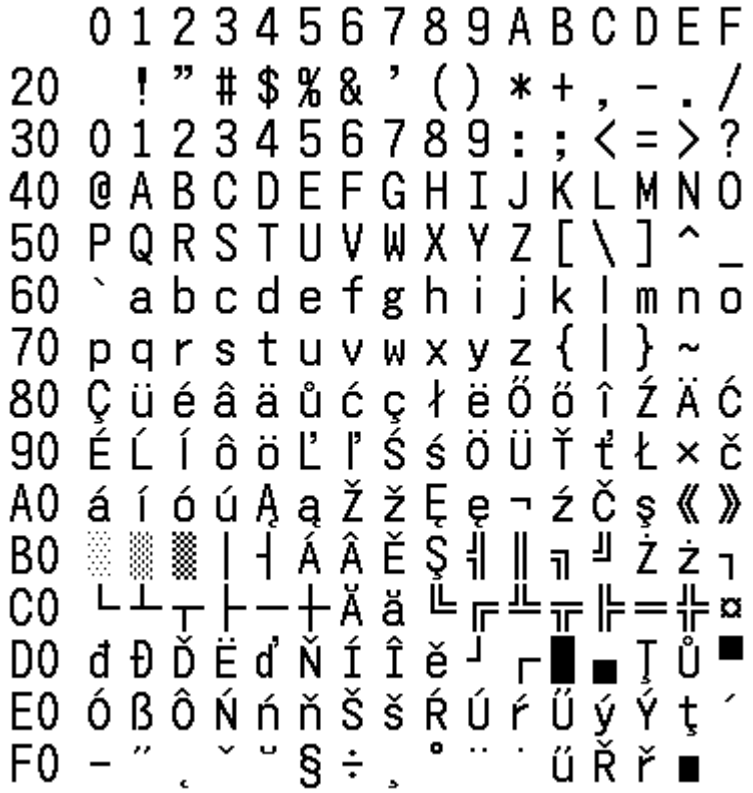


図 A-11 **CODE_PAGE_852** (Eastern Europe)

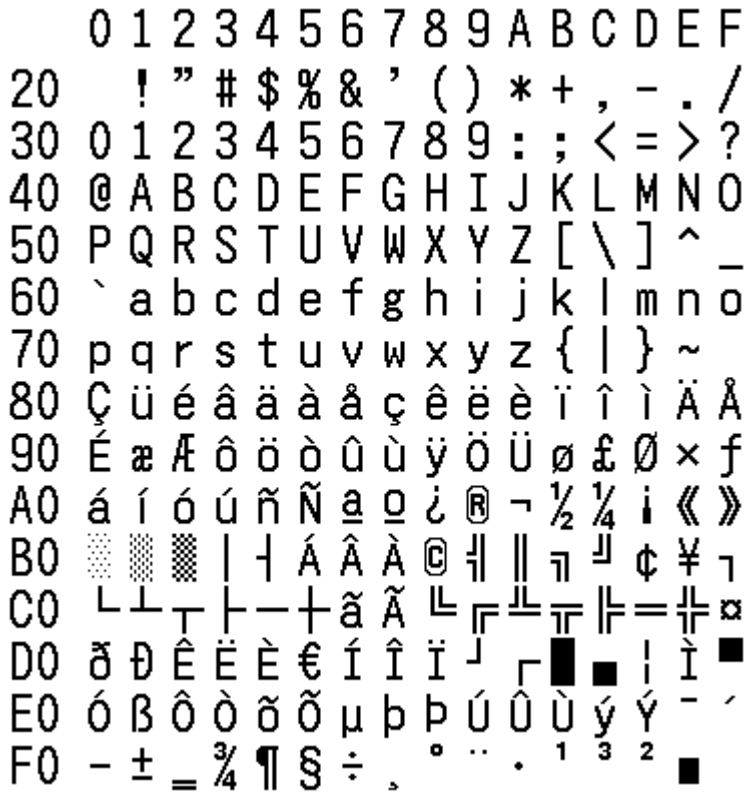


図 A-12 **CODE_PAGE_858** (Euro)

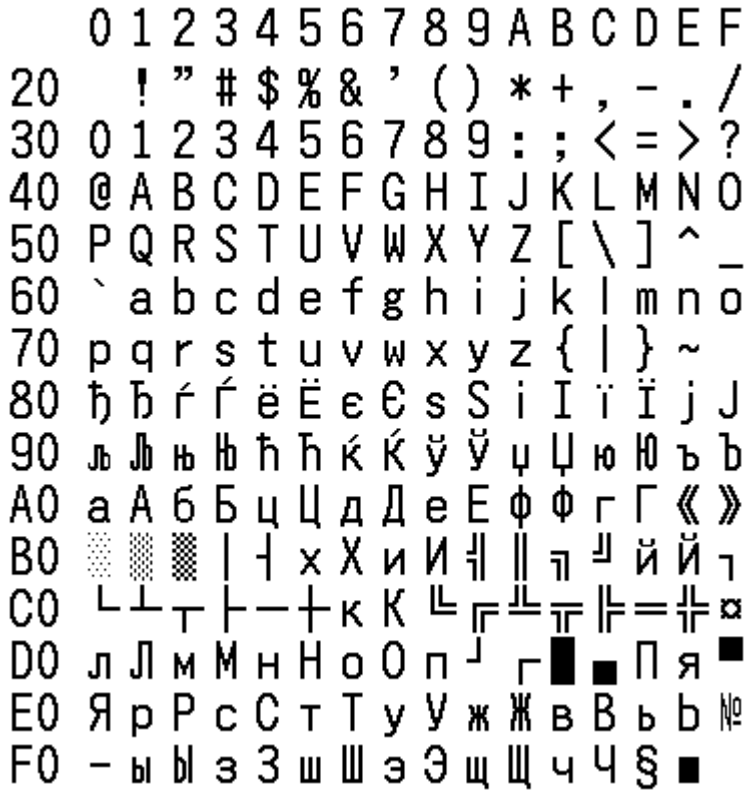


図 A-13 **CODE_PAGE_855** (Cyrillic)

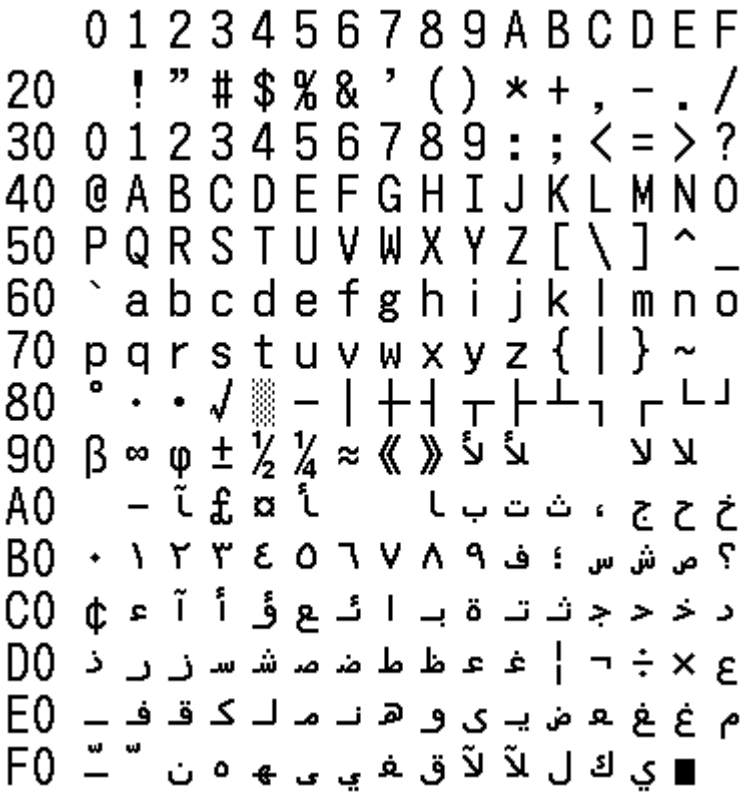


図 A-14 **CODE_PAGE_864** (Arabic)



図 A-15 **CODE_PAGE_1250** (Central European)



図 A-16 **CODE_PAGE_1251** (Cyrillic)

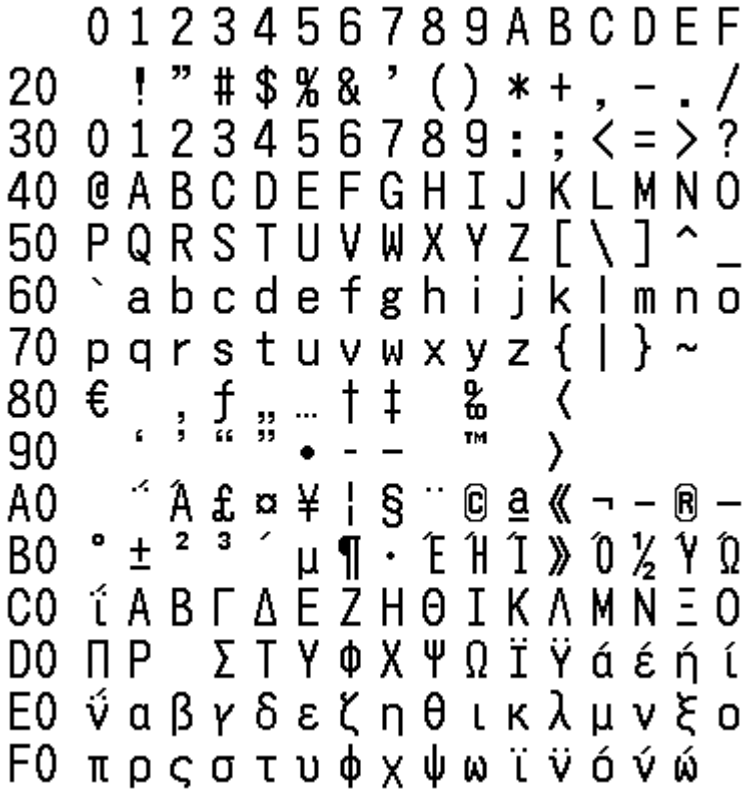


図 A-17 **CODE_PAGE_1253** (Greek)



図 A-18 **CODE_PAGE_1254** (Turkish)

A.2 国際文字セット

設定する国際文字セットにより、特定の文字コードの印字結果が異なります。
特定の文字コード、及びその印字結果を下記に示します。

	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
COUNTRY_GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_DENMARK_1	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
COUNTRY_JAPAN	#	$	@	[	¥	]	^	`	{	|	}	~
COUNTRY_NORWAY	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	$	á	¡	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	$	@	[	\	]	^	`	{	|	}	~

図 A-19 国際文字セット

付録 B　バーコードサイズ一覧

B.1 バーコードサイズ一覧

B.1.1 printBarcode



(1) バーコードイメージの高さ

hriFont	*hriPosition*	バーコードの上端から 基準点までの長さ	バーコードイメージの高さ
FONT_A	**HRI_NONE**	*moduleHeight*	*moduleHeight*
	HRI_POSITION_ABOVE	*moduleHeight*+32	*moduleHeight*+32
	HRI_POSITION_BELOW	*moduleHeight*	*moduleHeight*+32
	HRI_POSITION_ABOVE_BELOW	*moduleHeight*+64	*moduleHeight*+64
FONT_B	**HRI_NONE**	*moduleHeight*	*moduleHeight*
	HRI_POSITION_ABOVE	*moduleHeight*+24	*moduleHeight*+24
	HRI_POSITION_BELOW	*moduleHeight*	*moduleHeight*+24
	HRI_POSITION_ABOVE_BELOW	*moduleHeight*+48	*moduleHeight*+48

(2) バーコードイメージの幅

barcodeSymbol	*moduleSize*	バーコードイメージの幅
BARCODE_UPC_A	**BARCODE_MODULE_WIDTH_2**	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_UPC_E	**BARCODE_MODULE_WIDTH_2**	102
	BARCODE_MODULE_WIDTH_3	153
	BARCODE_MODULE_WIDTH_4	204
	BARCODE_MODULE_WIDTH_5	255
	BARCODE_MODULE_WIDTH_6	306
BARCODE_EAN13	**BARCODE_MODULE_WIDTH_2**	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_JAN13	**BARCODE_MODULE_WIDTH_2**	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_EAN8	**BARCODE_MODULE_WIDTH_2**	134
	BARCODE_MODULE_WIDTH_3	201
	BARCODE_MODULE_WIDTH_4	268
	BARCODE_MODULE_WIDTH_5	335
	BARCODE_MODULE_WIDTH_6	402
BARCODE_JAN8	**BARCODE_MODULE_WIDTH_2**	134
	BARCODE_MODULE_WIDTH_3	201
	BARCODE_MODULE_WIDTH_4	268
	BARCODE_MODULE_WIDTH_5	335
	BARCODE_MODULE_WIDTH_6	402
BARCODE_CODE93	**BARCODE_MODULE_WIDTH_2**	18×バーコードデータ数+56
	BARCODE_MODULE_WIDTH_3	27×バーコードデータ数+84
	BARCODE_MODULE_WIDTH_4	36×バーコードデータ数+112
	BARCODE_MODULE_WIDTH_5	45×バーコードデータ数+140
	BARCODE_MODULE_WIDTH_6	54×バーコードデータ数+168
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_2**	22×バーコードデータ数+26
	BARCODE_MODULE_WIDTH_3	33×バーコードデータ数+39
	BARCODE_MODULE_WIDTH_4	44×バーコードデータ数+52
	BARCODE_MODULE_WIDTH_5	55×バーコードデータ数+65

barcodeSymbol	*moduleSize*	バーコードイメージの幅
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_6**	66×バーコードデータ数+78
BARCODE_GS1_OMNI_DIRECTIONAL	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_TRUNCATED	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_LIMITED	**BARCODE_MODULE_WIDTH_2**	158
	BARCODE_MODULE_WIDTH_3	237
	BARCODE_MODULE_WIDTH_4	316
	BARCODE_MODULE_WIDTH_5	395
	BARCODE_MODULE_WIDTH_6	474
BARCODE_GS1_EXPANDED[*1]	**BARCODE_MODULE_WIDTH_2**	バーコードモジュール数×2
	BARCODE_MODULE_WIDTH_3	バーコードモジュール数×3
	BARCODE_MODULE_WIDTH_4	バーコードモジュール数×4
	BARCODE_MODULE_WIDTH_5	バーコードモジュール数×5
	BARCODE_MODULE_WIDTH_6	バーコードモジュール数×6

*1:　バーコードモジュール数は指定するバーコードデータによって決まります。

barcodeSymbol	*nwRatio*	*moduleSize*	バーコードイメージの幅
BARCODE_CODE39	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	26×バーコードデータ数+50
		BARCODE_MODULE_WIDTH_3	39×バーコードデータ数+75
		BARCODE_MODULE_WIDTH_4	52×バーコードデータ数+100
		BARCODE_MODULE_WIDTH_5	65×バーコードデータ数+125
		BARCODE_MODULE_WIDTH_6	78×バーコードデータ数+150
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	29×バーコードデータ数+56
		BARCODE_MODULE_WIDTH_3	45×バーコードデータ数+87
		BARCODE_MODULE_WIDTH_4	58×バーコードデータ数+112
		BARCODE_MODULE_WIDTH_5	74×バーコードデータ数+143
		BARCODE_MODULE_WIDTH_6	87×バーコードデータ数+168
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	32×バーコードデータ数+ 62
		BARCODE_MODULE_WIDTH_3	48×バーコードデータ数+93
		BARCODE_MODULE_WIDTH_4	64×バーコードデータ数+124
		BARCODE_MODULE_WIDTH_5	80×バーコードデータ数+155
		BARCODE_MODULE_WIDTH_6	96×バーコードデータ数+186
BARCODE_ITF	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	14×バーコードデータ数+16
		BARCODE_MODULE_WIDTH_3	21×バーコードデータ数+24
		BARCODE_MODULE_WIDTH_4	28×バーコードデータ数+32
		BARCODE_MODULE_WIDTH_5	35×バーコードデータ数+40
		BARCODE_MODULE_WIDTH_6	42×バーコードデータ数+48
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	16×バーコードデータ数+17
		BARCODE_MODULE_WIDTH_3	25×バーコードデータ数+26
		BARCODE_MODULE_WIDTH_4	32×バーコードデータ数+34
		BARCODE_MODULE_WIDTH_5	41×バーコードデータ数+43
		BARCODE_MODULE_WIDTH_6	48×バーコードデータ数+51
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	18×バーコードデータ数+18
		BARCODE_MODULE_WIDTH_3	27×バーコードデータ数+27
		BARCODE_MODULE_WIDTH_4	36×バーコードデータ数+36
		BARCODE_MODULE_WIDTH_5	45×バーコードデータ数+45
		BARCODE_MODULE_WIDTH_6	54×バーコードデータ数+54

barcodeSymbol	*nwRatio*	*moduleSize*	バーコードイメージの幅
BARCODE_CODABAR*1	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	20×データ数+2×(2+ワイドデータ数)-2
		BARCODE_MODULE_WIDTH_3	30×データ数+3×(2+ワイドデータ数)-3
		BARCODE_MODULE_WIDTH_4	40×データ数+4×(2+ワイドデータ数)-4
		BARCODE_MODULE_WIDTH_5	50×データ数+5×(2+ワイドデータ数)-5
		BARCODE_MODULE_WIDTH_6	60×データ数+6×(2+ワイドデータ数)-6
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	22×データ数+3×(2+ワイドデータ数)-2
		BARCODE_MODULE_WIDTH_3	34×データ数+5×(2+ワイドデータ数)-3
		BARCODE_MODULE_WIDTH_4	44×データ数+6×(2+ワイドデータ数)-4
		BARCODE_MODULE_WIDTH_5	56×データ数+8×(2+ワイドデータ数)-5
		BARCODE_MODULE_WIDTH_6	66×データ数+9×(2+ワイドデータ数)-6
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	24×データ数+4×(2+ワイドデータ数)-2
		BARCODE_MODULE_WIDTH_3	36×データ数+6×(2+ワイドデータ数)-3
		BARCODE_MODULE_WIDTH_4	48×データ数+8×(2+ワイドデータ数)-4
		BARCODE_MODULE_WIDTH_5	60×データ数+10×(2+ワイドデータ数)-5
		BARCODE_MODULE_WIDTH_6	72×データ数+12×(2+ワイドデータ数)-6

*1: データ数はスタートキャラクタ、ストップキャラクタを除いた全てのキャラクタ数です。
ワイドデータ数は「：／.＋」の数です。

barcodeSymbol	データ数	*moduleSize*	バーコード イメージの幅
BARCODE_EAN13_ADDON	14または15	**BARCODE_MODULE_WIDTH_2**	244
		BARCODE_MODULE_WIDTH_3	366
		BARCODE_MODULE_WIDTH_4	488
		BARCODE_MODULE_WIDTH_5	610
		BARCODE_MODULE_WIDTH_6	732
	17または18	**BARCODE_MODULE_WIDTH_2**	298
		BARCODE_MODULE_WIDTH_3	447
		BARCODE_MODULE_WIDTH_4	596
		BARCODE_MODULE_WIDTH_5	745
		BARCODE_MODULE_WIDTH_6	894
BARCODE_JAN13_ADDON	14または15	**BARCODE_MODULE_WIDTH_2**	244
		BARCODE_MODULE_WIDTH_3	366
		BARCODE_MODULE_WIDTH_4	488
		BARCODE_MODULE_WIDTH_5	610
		BARCODE_MODULE_WIDTH_6	732
	17または18	**BARCODE_MODULE_WIDTH_2**	298
		BARCODE_MODULE_WIDTH_3	447
		BARCODE_MODULE_WIDTH_4	596
		BARCODE_MODULE_WIDTH_5	745
		BARCODE_MODULE_WIDTH_6	894

B.1.2 printPDF417



(1) バーコードイメージの高さ

バーコードイメージの高さ*1 ＝ *moduleHeight* × *row**2

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ
*2: *row*≠0

(2) バーコードイメージの幅

pdf417Symbol が **PDF417_STANDARD** の場合

バーコードイメージの幅 ＝（17 × *column**1 ＋ 69）× モジュールサイズ値

*1: *column*≠0

pdf417Symbol が **PDF417_COMPACT** の場合

バーコードイメージの幅 ＝（17 × *column**1 ＋ 35）× モジュールサイズ値

*1: *column*≠0

モジュールサイズ値

moduleSize	モジュールサイズ値
PDF417_MODULE_WIDTH_2	2
PDF417_MODULE_WIDTH_3	3
PDF417_MODULE_WIDTH_4	4
PDF417_MODULE_WIDTH_5	5
PDF417_MODULE_WIDTH_6	6
PDF417_MODULE_WIDTH_7	7
PDF417_MODULE_WIDTH_8	8

B.1.3 printQRCode



(1) バーコードイメージの高さ及び幅

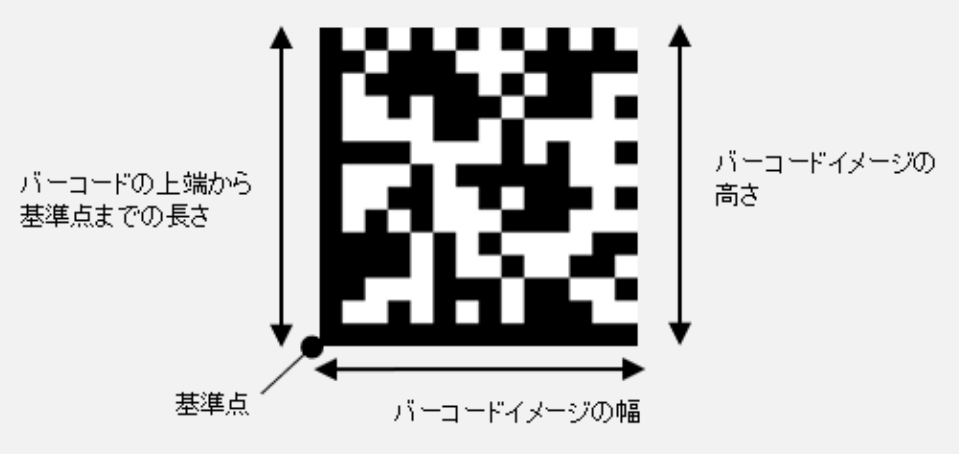
バーコードイメージの高さ*1 及び幅 ＝（4 × バージョン*2 ＋ 17）× モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ
*2: バージョンはバーコードデータの内容及びエラー訂正レベルにより決まります。

モジュールサイズ値

moduleSize	モジュールサイズ値
QR_MODULE_SIZE_2	2
QR_MODULE_SIZE_3	3
QR_MODULE_SIZE_4	4
QR_MODULE_SIZE_5	5
QR_MODULE_SIZE_6	6
QR_MODULE_SIZE_7	7
QR_MODULE_SIZE_8	8
QR_MODULE_SIZE_9	9
QR_MODULE_SIZE_10	10
QR_MODULE_SIZE_11	11
QR_MODULE_SIZE_12	12
QR_MODULE_SIZE_13	13
QR_MODULE_SIZE_14	14
QR_MODULE_SIZE_15	15
QR_MODULE_SIZE_16	16

B.1.4 printDataMatrix



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ ＝ 縦方向モジュール数 × モジュールサイズ値

バーコードイメージの幅 ＝ 横方向モジュール数 × モジュールサイズ値

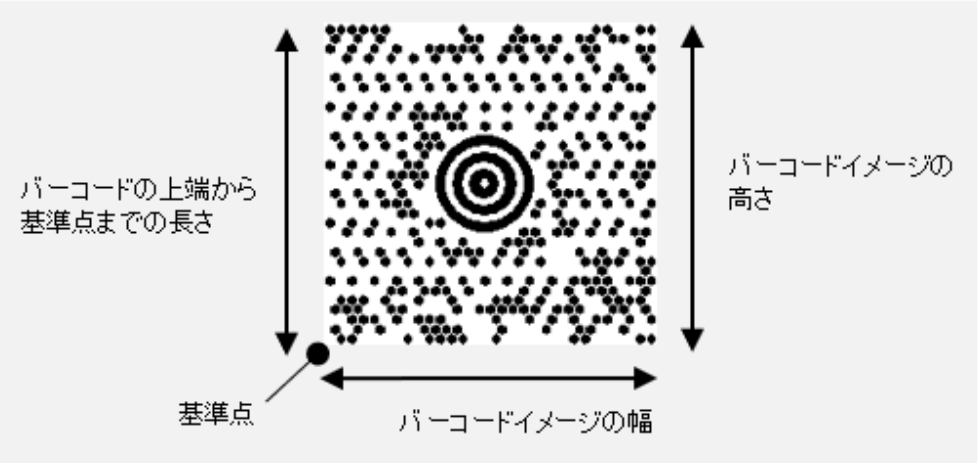
dataMatrixModule	縦方向モジュール数	横方向モジュール数
DATA_MATRIX_10_10	10	10
DATA_MATRIX_12_12	12	12
DATA_MATRIX_14_14	14	14
DATA_MATRIX_16_16	16	16
DATA_MATRIX_18_18	18	18
DATA_MATRIX_20_20	20	20
DATA_MATRIX_22_22	22	22
DATA_MATRIX_24_24	23	23
DATA_MATRIX_26_26	26	26
DATA_MATRIX_32_32	32	32
DATA_MATRIX_36_36	36	36
DATA_MATRIX_40_40	40	40
DATA_MATRIX_44_44	44	44
DATA_MATRIX_48_48	48	48
DATA_MATRIX_52_52	52	52
DATA_MATRIX_64_64	64	64
DATA_MATRIX_72_72	72	72
DATA_MATRIX_80_80	80	80
DATA_MATRIX_88_88	88	88
DATA_MATRIX_96_96	96	96
DATA_MATRIX_104_104	104	104

dataMatrixModule	縦方向モジュール数	横方向モジュール数
DATA_MATRIX_120_120	120	120
DATA_MATRIX_132_132	132	132
DATA_MATRIX_144_144	144	144
DATA_MATRIX_8_18	8	18
DATA_MATRIX_8_32	8	32
DATA_MATRIX_12_26	12	26
DATA_MATRIX_12_36	12	36
DATA_MATRIX_16_36	16	36
DATA_MATRIX_16_48	16	48

モジュールサイズ値

moduleSize	モジュールサイズ値
DATAMATRIX_MODULE_SIZE_2	2
DATAMATRIX_MODULE_SIZE_3	3
DATAMATRIX_MODULE_SIZE_4	4
DATAMATRIX_MODULE_SIZE_5	5
DATAMATRIX_MODULE_SIZE_6	6
DATAMATRIX_MODULE_SIZE_7	7
DATAMATRIX_MODULE_SIZE_8	8
DATAMATRIX_MODULE_SIZE_9	9
DATAMATRIX_MODULE_SIZE_10	10
DATAMATRIX_MODULE_SIZE_11	11
DATAMATRIX_MODULE_SIZE_12	12
DATAMATRIX_MODULE_SIZE_13	13
DATAMATRIX_MODULE_SIZE_14	14
DATAMATRIX_MODULE_SIZE_15	15
DATAMATRIX_MODULE_SIZE_16	16

B.1.5 printMaxicode



(1) バーコードイメージの高さ

バーコードイメージの高さ*1 ＝ 200

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

(2) バーコードイメージの幅

バーコードイメージの幅 ＝ 210

B.1.6 printGS1DataBarStacked



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝ 13 × モジュールサイズ値

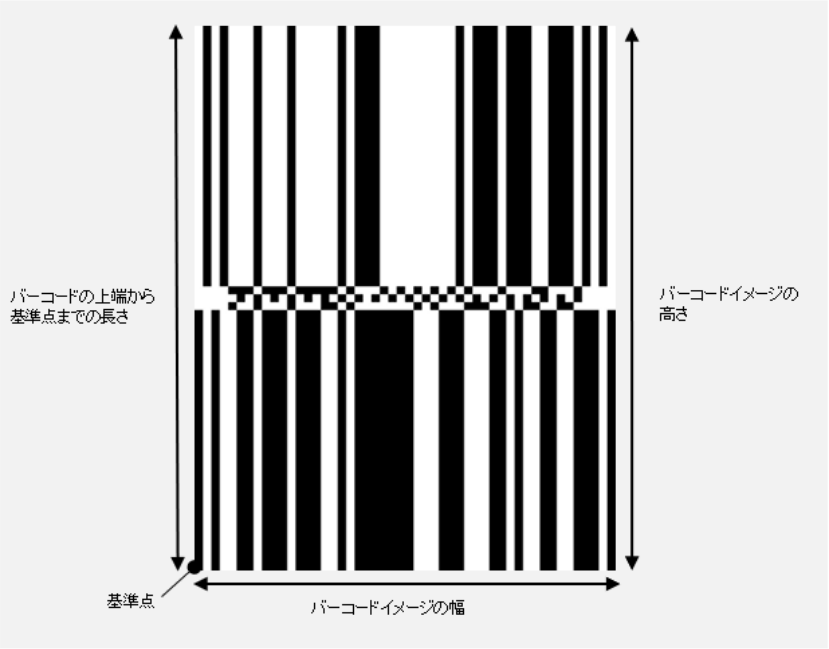
*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

バーコードイメージの幅 ＝ 50 × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.1.7 printGS1DataBarStackedOmnidirectional



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝（*moduleHeight* × 2 ＋ 3）× モジュールサイズ値

*1: バーコードイメージの高さ ＝ バーコードの上端から基準点までの長さ

バーコードイメージの幅 ＝ 50 × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

B.1.8 printGS1DataBarExpandedStacked



(1) バーコードイメージの高さ及び幅

バーコードイメージの高さ*1 ＝ ((34 ＋ 3) × 段数*2 ＋ 34) × モジュールサイズ値

*1: バーコードイメージの高さ＝バーコードの上端から基準点までの長さ
*2: 段数はバーコードデータによって決まります。

バーコードイメージの幅 ＝ (4 ＋ 49 × *column* / 2) × モジュールサイズ値

モジュールサイズ値

moduleSize	モジュールサイズ値
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16