



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.05

[対象製品]

RP-F10シリーズ

RP-G10シリーズ

セイコーインスツル株式会社

Rev.01	2019年	2月
Rev.02	2019年	7月
Rev.03	2019年	8月
Rev.04	2020年	2月
Rev.05	2020年	6月

©セイコーインスツル株式会社 2019 - 2020

無断転載を禁じます。

Android™は、Google LLC.の商標です。

Bluetooth® は、Bluetooth SIG, Inc.の登録商標です。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

文中の社名、商品名等は各社の商標または登録商標である場合があります。

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以下に記載します。

プリンター	インターフェイス
RP-F10シリーズ	Bluetooth
	USB
	TCP/IP
RP-G10シリーズ	USB

用語一覧

本書で使用する用語について以下の通りに定義します。

用語	内容
プリンターコマンド	「RP-F10シリーズ サーマルプリンタ 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

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	2-3
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
4章	ライブラリの機能	4-1
4.1	スタンダードモードとページモード	4-1
4.1.1	基本動作	4-1
4.1.2	スタンダードモードにおけるテキストデータの印字	4-3
4.1.3	ページモードにおける印字データの展開位置	4-4
4.1.4	ページモードの印字領域外の印字データ処理	4-6
4.2	APIリファレンス	4-7
4.2.1	PrinterManagerクラス	4-8
	①スタンダードモード・ページモード共通メソッド	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)	4-26
	disconnect プリンターとの通信切断	4-27
	setBarcodeScannerListener バーコードスキャナーのコールバック開始・終了	4-27
	cutPaper 用紙のカット	4-28
	openDrawer キャッシュドロワを開く	4-28
	buzzer ブザーの鳴動	4-28
	externalBuzzer 外部ブザーの鳴動	4-29
	getStatus プリンターステータスの取得	4-29
	setCallbackFunctionListener	

プリンターステータス変化のコールバック開始・終了	4-30
abort	プリンターのデータ待ち状態解除 4-31
registerLogo	ロゴの登録 4-31
unregisterLogo	ロゴの登録削除 4-32
registerStyleSheet	スタイルシートの登録 4-32
unregisterStyleShee	スタイルシートの登録削除 4-32
resetPrinter	プリンターのリセット 4-32
getPrinterResponse	プリンターからの各種応答取得 4-33
startDiscoveryPrinter	プリンターの探索開始 (Bluetooth) 4-34
startDiscoveryPrinter	プリンターの探索開始 (USB) 4-34
startDiscoveryPrinter	プリンターの探索開始 (TCP/IP) 4-35
cancelDiscoveryPrinter	プリンターの探索中断 4-35
getFoundPrinter	発見されたプリンター情報リストの取得 4-36
getSendTimeout	送信タイムアウト時間の取得 4-36
setSendTimeout	送信タイムアウト時間の設定 4-36
getReceiveTimeout	受信タイムアウト時間の取得 4-36
setReceiveTimeout	受信タイムアウト時間の設定 4-37
getInternationalCharacter	国際文字セットの取得 4-37
setInternationalCharacter	国際文字セットの設定 4-37
getCodePage	コードページの取得 4-38
setCodePage	コードページの設定 4-38
getPrinterModel	プリンターモデルの取得 4-38
getPortType	接続中ポート種別の取得 4-38
isConnect	プリンターとの通信状態の確認 4-39
getSocketKeepingTime	ソケット維持時間の取得 4-39
setSocketKeepingTime	ソケット維持時間の設定 4-39
controlTransaction	一括処理の開始・終了 4-40
②スタンダードモード専用メソッド	4-42
sendText	テキストデータの送信 4-42
sendTextEx	書式指定テキストデータの送信 4-42
printBarcode	バーコードの印字 4-44
printPDF417	PDF417の印字 4-47
printQRcode	QRコードの印字 4-48
printDataMatrix	Data Matrixの印字 4-49
printMaxiCode	MaxiCodeの印字 4-50
printGS1DataBarStacked	GS1 Databar Stackedの印字 4-50

printGS1DataBarStackedOmnidirectional	GS1 Databar Stacked Omni-directionalの印字	4-51
printGS1DataBarExpandedStacked	GS1 Databar Expanded Stackedの印字	4-52
sendBinary	バイナリデータの送信	4-52
sendDataFile	指定ファイルの送信	4-53
printLogo	ロゴの印字	4-54
③ページード専用メソッド		4-55
enterPageMode	ページモードの開始	4-56
exitPageMode	ページモードの終了	4-56
setPageModeArea	ページモードの印字領域の指定	4-56
setPageModeDirection	ページモードの印字方向の指定	4-58
setPageModeLineSpacing	ページモードの改行量の指定	4-58
printPageMode	ページモードの印字	4-58
printPageModeText	ページモードのテキストデータの送信	4-59
printPageModeTextEx	ページモードの書式指定テキストデータの送信	4-59
printPageModeBarcode	ページモードのバーコードの印字	4-60
printPageModePDF417	ページモードのPDF417の印字	4-63
printPageModeQRcode	ページモードのQRコードの印字	4-65
printPageModeDataMatrix	ページモードのData Matrixの印字	4-66
printPageModeMaxiCode	ページモードのMaxiCodeの印字	4-66
printPageModeGS1DataBarStacked	ページモードのGS1 Databar Stackedの印字	4-67
printPageModeGS1DataBarStackedOmnidirectional	ページモードのGS1 Databar Stacked Omni-directionalの印字	4-68
printPageModeGS1DataBarExpandedStacked	ページモードのGS1 Databar Expanded Stackedの印字	4-68
sendPageModeBinary	ページモードのバイナリデータの送信	4-69
printPageModeImageFile	ページモードのイメージファイルの描画	4-70
printPageModeRectangle	ページモードの矩形の描画	4-70
printPageModeLine	ページモードの罫線の印字	4-72
printPageModeLogo	ページモードのロゴの印字	4-73
4.2.2 PrinterEventクラス		4-74
getEventType	終了イベントの取得	4-74
4.2.3 PrinterListenerインターフェイス		4-75
finishEvent	プリンター探索の終了イベント	4-75
4.2.4 PrinterInfoクラス		4-76
getPrinterModelName	プリンターモデル名の取得	4-76
getBluetoothAddress	Bluetoothアドレスの取得	4-76
getMacAddress	MACアドレスの取得	4-76

getIpAddress	IPアドレスの取得	4-77
getIsBonded	ペアリング状態の取得	4-77
getDevicePath	デバイスパスの取得	4-77
4.2.5 PrinterExceptionクラス		4-78
PrinterException	コンストラクタ	4-79
getErrorCode	エラーコードの取得	4-79
4.2.6 CallbackFunctionListenerインターフェイス		4-80
onStatusChanged	プリンタステータスの変化イベント	4-80
4.2.7 BarcodeScannerListenerインターフェイス		4-81
onBarcodeScannerChangedOnline	バーコードスキャナーの接続イベント	4-81
onBarcodeScannerChangedOffline	バーコードスキャナーの切断イベント	4-81
onBarcodeScannerReadData	バーコードデータの受信イベント	4-82

5章 サンプルプログラム 5-1

5.1	インストール	5-1
5.2	画面構成	5-3
5.2.1	メイン画面	5-3
5.2.2	[SETTINGS]画面	5-4
5.3	注意事項	5-4

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

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

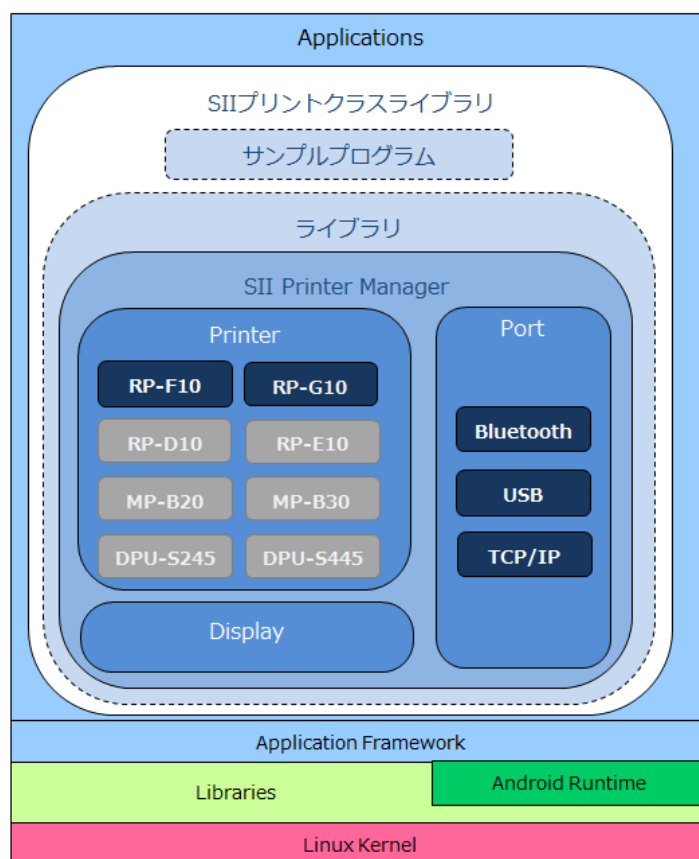
ライブラリとサンプルプログラムを含むSIIプリントクラスライブラリは、Androidアプリケーションに「はじめに 対象プリンター」に記載のSII製プリンターを利用するための機能を提供します。

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。



1.2.2 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、以下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})
- 用紙のカット
- 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
- プリンターのハードウェアリセット
- ドロワの動作制御
- ブザーの鳴動制御
- ディスプレイの表示制御^{*2}
- バーコードスキャナーのコールバック関数登録

*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。

*2: ディスプレイの表示制御については、DSP-A01 シリーズ用「SII Print Class Library for Android™ アプリケーションプログラマーズガイド」を参照してください。

(注意) 推奨バーコードスキャナー、及びバーコードスキャナーの設定については、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

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。

プリンター	モデル		RP-F10		RP-G10
	通信インターフェイス		Bluetooth	TCP/IP	USB
Android デバイス	通信ポート		Bluetooth*1	TCP/IP*2	USB*3
	OS	Android 5.0 (API 21)	対応	対応	対応
		Android 5.1 (API 22)			
		Android 6.0 (API 23)			
		Android 7.0 (API 24)			
		Android 7.1 (API 25)			
		Android 8.0 (API 26)			
		Android 8.1 (API 27)			
		Android 9.0 (API 28)			
		Android 10.0 (API 29)			
サポート言語			日本語/英語		

*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。

*3: 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。

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以下の[値]に設定してください。

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、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

• Bluetooth 接続の場合

MS	機能	値
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
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
38-1	スキャナー自動ステータス応答機能選択 ^{*1} (Scanner Auto Status Back)	0 : 有効 ^{*2}

*1: バーコードスキャナーをサポートするプリンターのファームウェアバージョンは、Ver.1.10以降です。

*2: バーコードスキャナーを使用する場合は、「有効」を選択してください。

バーコードスキャナーを含む周辺機器の組み合わせについては、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

• USB 接続の場合

MS	機能	値
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
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
38-1	スキャナー自動ステータス応答機能選択 ^{*1} (Scanner Auto Status Back)	0 : 有効 ^{*2}

*1: バーコードスキャナーをサポートするプリンターのファームウェアバージョンは、Ver.1.10以降です。

*2: バーコードスキャナーを使用する場合は、「有効」を選択してください。

バーコードスキャナーを含む周辺機器の組み合わせについては、「RP-F10シリーズ サーマルプリンタ 取扱説明書」を参照してください。

• TCP/IP 接続の場合

MS	機能	値
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
13-3	リアルタイムコマンド選択 (Realtime Command)	1 : 有効
38-1	スキャナー自動ステータス応答機能選択 ^{*1} (Scanner Auto Status Back)	0 : 有効 ^{*2}

*1: バーコードスキャナーをサポートするプリンターのファームウェアバージョンは、Ver.1.10以降です。

*2: バーコードスキャナーを使用する場合は、「有効」を選択してください。

バーコードスキャナーを含む周辺機器の組み合わせについては、「RP-F10シリーズ サーマルプリンタ
取扱説明書」を参照してください。

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、以下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、以下の通りです。

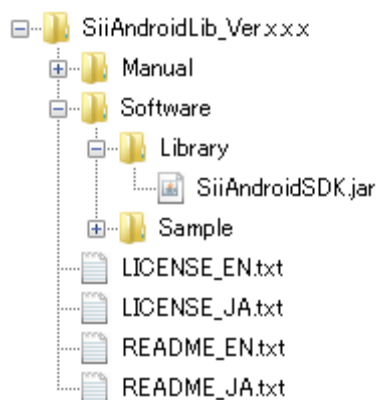


図 3-1

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に含まれるサンプルプログラムのプロジェクトを例にして、ライブラリをAndroid Studioのプロジェクトに組み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のAndroid Projectビューに表示されているモジュール(app)を選択した状態で右クリックし、[New] - [Directory]を選択してください。(図 3-2)
表示されるダイアログのフォルダ名に"libs"を入力して[OK]ボタンをクリックしてください。(図 3-3)

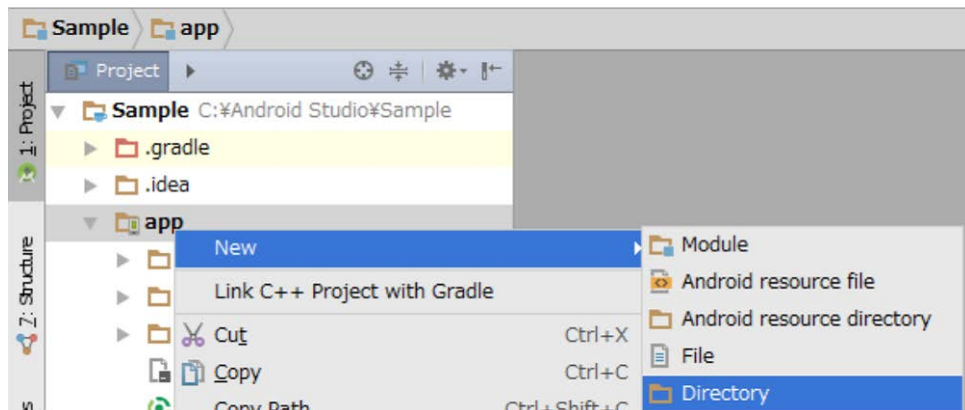


図 3-2

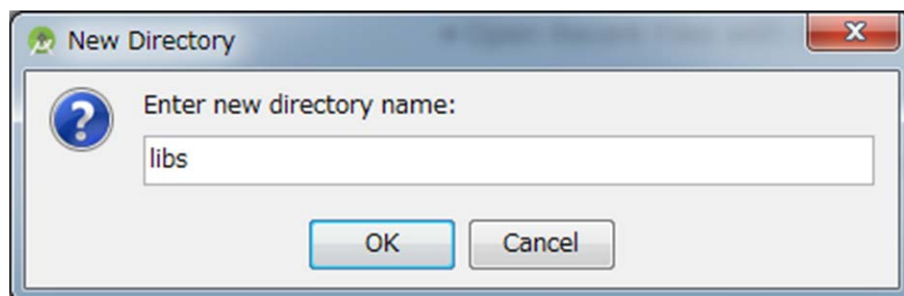


図 3-3

- (2) (1)で作成したフォルダ(¥Sample¥app¥libs)にライブラリファイル(SiiAndroidSDK.jar)をコピーしてください。

(3) ライブラリの追加が完了すると図 3-4の状態となります。

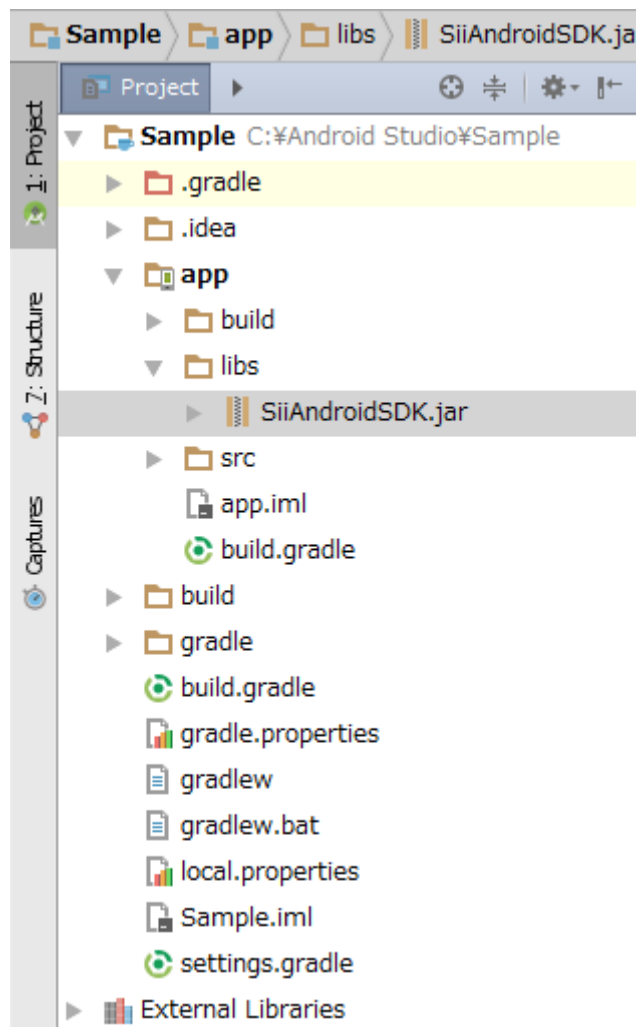


図 3-4

(4) メインとなるソースファイルの先頭に以下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.PrinterEvent;  
import com.seikoinstruments.sdk.thermalprinter.PrinterException;  
import com.seikoinstruments.sdk.thermalprinter.PrinterInfo;  
import com.seikoinstruments.sdk.thermalprinter.PrinterListener;  
import com.seikoinstruments.sdk.thermalprinter.PrinterManager;
```

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デバイスで以下の設定を行ってください。

(注意) 本手順はAndroid 7.1のメニューを基に記載しています。ご利用のAndroidデバイスにより、メニュー内容が異なる場合があります。

- (1) [設定] - [セキュリティ] - [提供元不明のアプリ]をオンにしてください。(図 3-5)



図 3-5

- (2) [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 3-6)

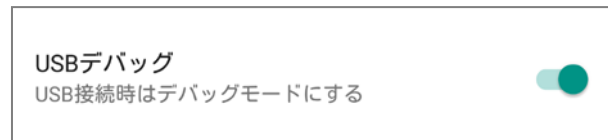


図 3-6

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 スタANDARDモードとページモード

4.1.1 基本動作

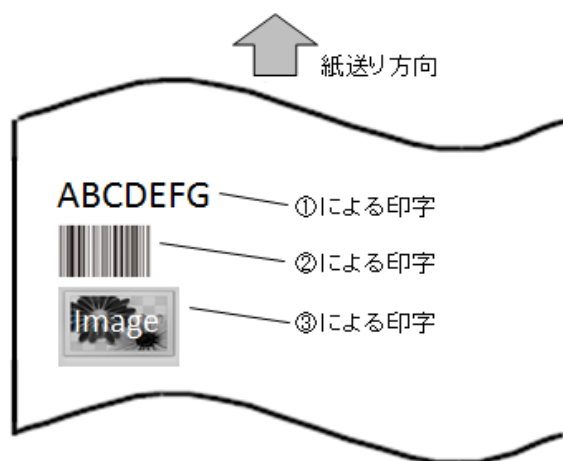
ライブラリには、「スタンダードモード」と「ページモード」の2つの印字モードがあります。
以下で「スタンダードモード」と「ページモード」について説明します。

(1) スタANDARDモード

スタンダードモードは、順次印字を行うモードです。

印字命令例

- ① テキストデータの送信
- ② バーコードの印字
- ③ 指定ファイルの送信(イメージファイルを指定)



スタンダードモードは、レシートなどのように非定型の長さでの印字に適しています。

(2) ページモード

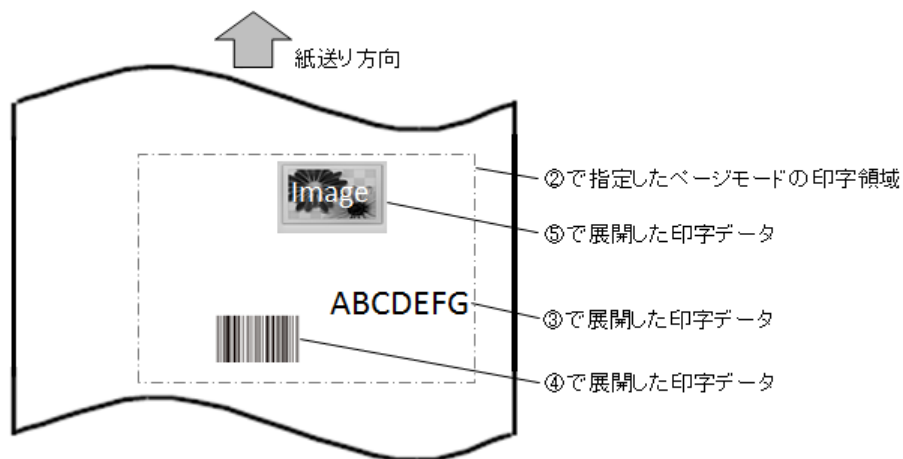
ページモードは、1ページ毎に印字を行うモードです。

ページモードでは、初めにページモードの印字領域を確保し、ページモードの印字領域上の任意の位置に印字データを展開していきます。

展開した印字データは、ページモードの印字命令により印字します。

印字命令例

- ① ページモードの開始
- ② ページモードの印字領域の指定
- ③ ページモードのテキストデータの送信
- ④ ページモードのバーコードの印字
- ⑤ ページモードのイメージファイルの描画
- ⑥ ページモードの印字 (③④⑤の印字データを②の印字領域で印字)
- ⑦ ページモードの終了



ページモードは、以下の印字に適しています。

- ・定型の長さでの印字
- ・文字の書き出しや罫線の印字位置を座標指定したい印字

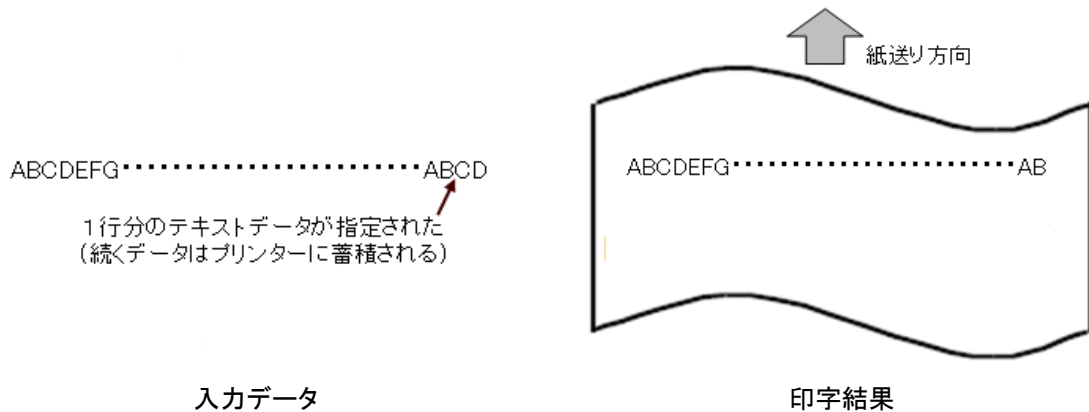
4.1.2 スタンドモードにおけるテキストデータの印字

スタンドモードにおけるテキストデータは1行毎に印字します。

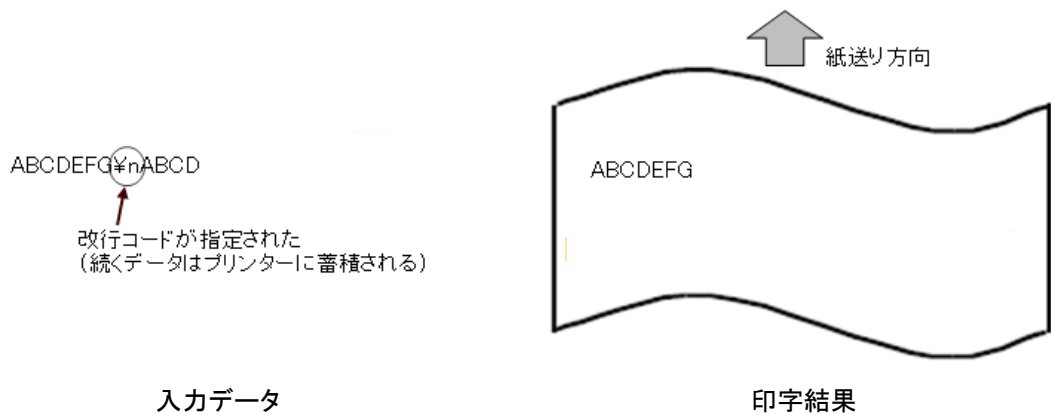
1行分に満たないテキストデータを指定した場合は、テキストデータはプリンターに蓄積されます。
蓄積されたテキストデータは、以下のどちらかの条件で印字されます。

- ・1行分のテキストデータを指定した場合
- ・改行コードを指定した場合

• 1行分のテキストデータを指定した場合の印字処理



• 改行コードを指定した場合の印字処理



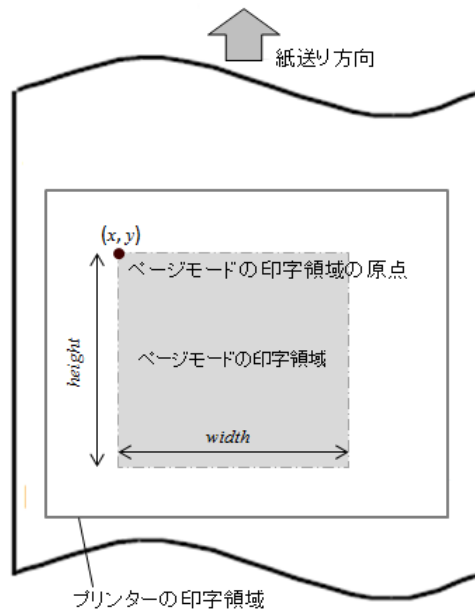
4.1.3 ページモードにおける印字データの展開位置

ページモードでは、印字領域、印字方向、基準点により印字データの展開位置が決まります。
印字領域、印字方向、基準点について説明します。

(1) ページモードの印字領域

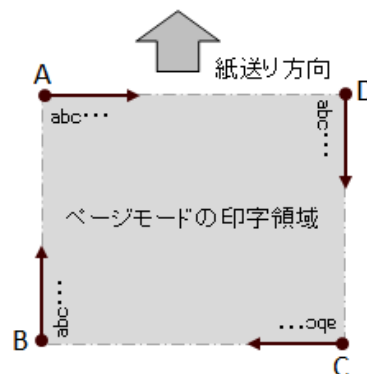
プリンターの印字領域に対して、ページモードの印字領域は、ページモードの印字領域の原点とその領域に対する幅と高さを指定します。下図に領域の考え方を示します。

ページモードの印字領域は、複数指定できます。



(2) 印字方向

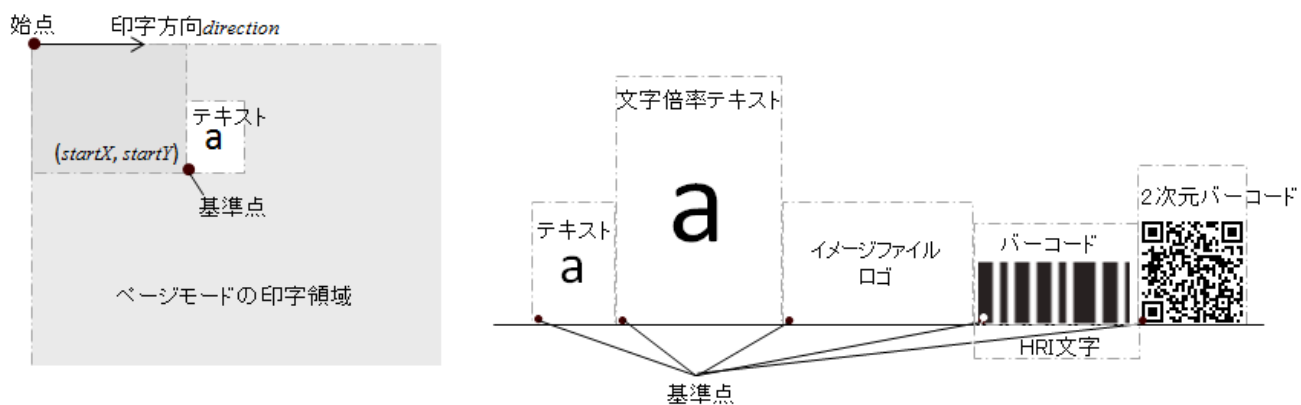
ページモードの印字領域を設定する際に、印字方向を指定します。
各方向の印字方向を指定すると、始点の位置も変わります。
印字方向と始点の関係を下図に示します。



- ・始点: 左上(図のA)、印字方向: 左→右
- ・始点: 左下(図のB)、印字方向: 下→上
- ・始点: 右下(図のC)、印字方向: 右→左
- ・始点: 右上(図のD)、印字方向: 上→下

(3) 基準点

ページモードの印字を行う場合の、データを展開する基準点と各印字要素(テキスト、イメージファイル、ロゴ、バーコード等)の関係を下図に示します。



(注意) 基準点は、ページモードの印字領域外に指定することはできません。

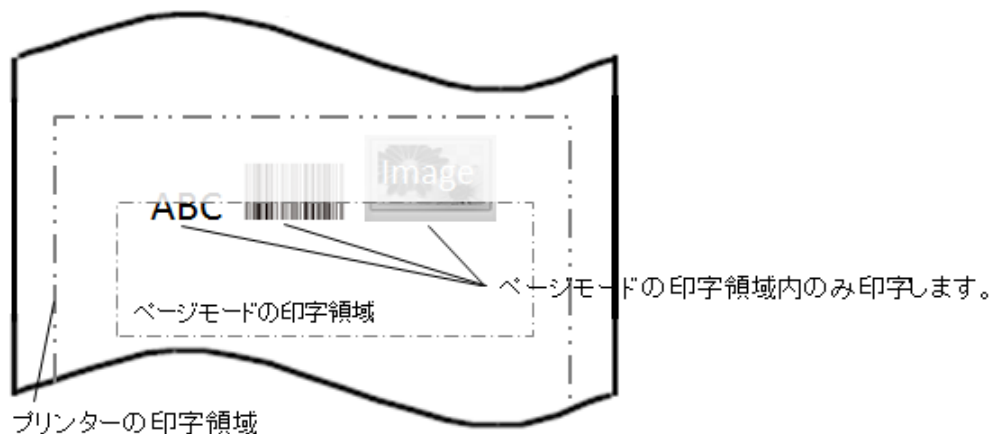
4.1.4 ページモードの印字領域外の印字データ処理

ページモードにおいて、展開した印字データがページモードの印字領域外になった場合の処理について説明します。

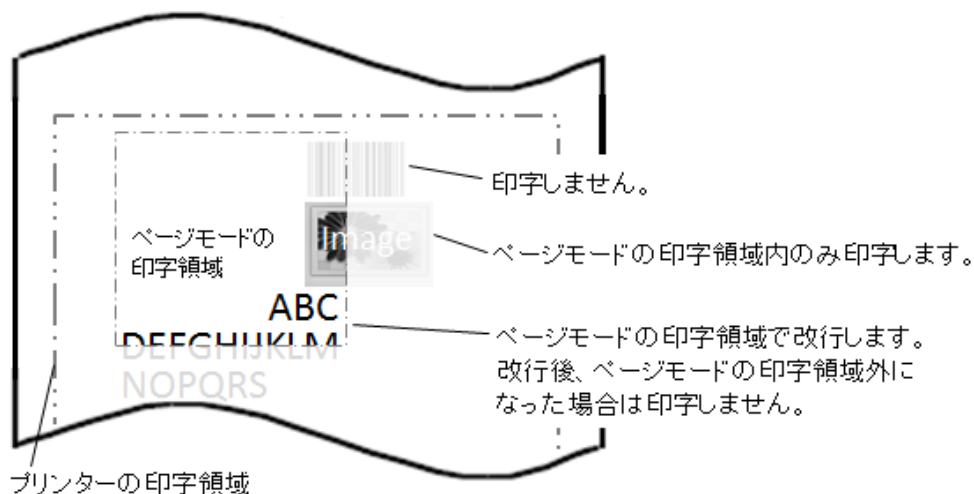
印字データ種類

テキスト	バーコード、 2次元バーコード	イメージファイル、ロゴ、 矩形、罫線
ABC		

- (1) ページモードの印字領域の上部に印字データが展開される場合



- (2) ページモードの印字領域の右側に印字データが展開される場合



(注意) 展開したバーコード印字データの一部がページモードの印字領域外になった場合、読み取りエラーや誤読が発生することがあります。

4.2 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。 「4.2.1 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。 「4.2.2 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。 「4.2.3 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。 「4.2.4 PrinterInfo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。 「4.2.5 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。 「4.2.6 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。 「4.2.7 BarcodeScannerListenerインターフェイス」を参照してください。	✓

*1: ✓: サポート、-: RP-F10/G10は非サポート

4.2.1 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。

PrinterManagerクラスでは、「スタンダードモード」「ページモード」を選択できます。

メソッド	説明
スタンダードモード・ページモード共通メソッド	スタンダードモード及びページモードで有効なメソッドです。 メソッドについては、「4.2.1(1)① スタンダードモード・ページモード共通メソッド」を参照してください。
スタンダードモード専用メソッド	スタンダードモードで有効なメソッドです。 メソッドについては、「4.2.1(1)② スタンダードモード専用メソッド」を参照してください。
ページモード専用メソッド	ページモードで有効なメソッドです。 メソッドについては、「4.2.1(1)③ ページモード専用メソッド」を参照してください。

① スタンダードモード・ページモード共通メソッド

スタンダードモード・ページモード共通メソッドを以下に示します。共通メソッドの詳細については、「4.2.1(4)① スタンダードモード・ページモード共通メソッド」を参照してください。

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)	✓
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	✓
cutPaper	用紙のカット	✓
openDrawer	キャッシュドロワを開く	✓
buzzer	ブザーの鳴動	－
externalBuzzer	外部ブザーの鳴動	✓
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	✓
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)	✓
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	✓
controlTransaction	一括処理の開始・終了	✓

*1: ✓: サポート、-: RP-F10/G10は非サポート

② スタンダードモード専用メソッド

スタンダードモード専用メソッドを以下に示します。専用メソッドの詳細については、「4.2.1(4)② スタンダードモード専用メソッド」を参照してください。

名前	説明	サポート*1
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	✓

名前	説明	サポート*1
printGS1DataBarExpandedStacked	GS1 Databar Expanded Stackedの印字	✓
sendBinary	バイナリデータの送信	✓
sendDataFile	指定ファイルの送信	✓
printLogo	ロゴの印字	✓

*1: ✓: サポート、-: RP-F10/G10は非サポート

③ ページモード専用メソッド

ページモード専用メソッドを以下に示します。専用メソッドの詳細については、「4.2.1(4)③ ページモード専用メソッド」を参照してください。

名前	説明	サポート*1
enterPageMode	ページモードの開始	✓
exitPageMode	ページモードの終了	✓
setPageModeArea	ページモードの印字領域の指定	✓
setPageModeDirection	ページモードの印字方向の指定	✓
setPageModeLineSpacing	ページモードの改行量の指定	✓
printPageMode	ページモードの印字	✓
printPageModeText	ページモードのテキストデータの送信	✓
printPageModeTextEx	ページモードの書式指定テキストデータの送信	✓
printPageModeBarcode	ページモードのバーコードの印字	✓
printPageModePDF417	ページモードのPDF417の印字	✓
printPageModeQRcode	ページモードのQRコードの印字	✓
printPageModeDataMatrix	ページモードのData Matrixの印字	✓
printPageModeMaxiCode	ページモードのMaxiCodeの印字	✓
printPageModeGS1DataBarStacked	ページモードのGS1 Databar Stackedの印字	✓
printPageModeGS1DataBarStackedOmnidirectional	ページモードのGS1 Databar Stacked Omni-directionalの印字	✓
printPageModeGS1DataBarExpandedStacked	ページモードのGS1 Databar Expanded Stackedの印字	✓
sendPageModeBinary	ページモードのバイナリデータの送信	✓
printPageModeImageFile	ページモードのイメージファイルの描画	✓
printPageModeRectangle	ページモードの矩形の描画	✓
printPageModeLine	ページモードの罫線の印字	✓
printPageModeLogo	ページモードのロゴの印字	✓

*1: ✓: サポート、-: RP-F10/G10は非サポート

(2) 定数一覧

① 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	0
COUNTRY_FRANCE	フランス	1
COUNTRY_GERMANY	ドイツ	2
COUNTRY_ENGLAND	イギリス	3
COUNTRY_DENMARK_1	デンマーク I	4
COUNTRY_SWEDEN	スウェーデン	5
COUNTRY_ITALY	イタリア	6
COUNTRY_SPAIN	スペイン I	7
COUNTRY_JAPAN	日本	8
COUNTRY_NORWAY	ノルウェー	9
COUNTRY_DENMARK_2	デンマーク II	10
COUNTRY_SPAIN_2	スペイン II	11
COUNTRY_LATIN_AMERICA	ラテンアメリカ	12
COUNTRY_ARABIA	アラビア	17

② 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
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}	Arabic (Code Page864)	37
CODE_PAGE_1250	Central European (Code Page1250)	45

定数名	説明	値
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。

③ 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

④ 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_RP_FG10	RP-F10/G10	301
PRINTER_MODEL_DEFAULT	プリンターモデルの初期値	284

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	Bluetooth	0
PRINTER_TYPE_USB	USB	1
PRINTER_TYPE_TCP	TCP/IP	2

⑥ 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

(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 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)

⑤ 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
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

⑥ 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。

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

⑦ 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)
BARCODE_SYMBOL_CODE128	CODE128	(c)
BARCODE_SYMBOL_ITF	ITF	(a),(b)
BARCODE_SYMBOL_CODABAR	CODABAR	(a),(b)
BARCODE_SYMBOL_EAN13_ADDON	EAN13 アドオン	(a)
BARCODE_SYMBOL_JAN13_ADDON	JAN13 アドオン	(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またはprintPageModeBarcodeを参照してください。

⑧ 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	<ul style="list-style-type: none"> ● printBarcode ● printPageModeBarcode
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	<ul style="list-style-type: none"> ● printPDF417 ● printPageModePDF417
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	<ul style="list-style-type: none"> ● printQRcode ● printPageModeQRcode
QR_MODULE_SIZE_2	2ドット	
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	

定数名	説明	使用メソッド
DATAMATRIX_MODULE_SIZE_2	2ドット	<ul style="list-style-type: none"> ● printDataMatrix ● printPageModeDataMatrix
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	
GSIDATABAR_MODULE_SIZE_2	2ドット	<ul style="list-style-type: none"> ● printGS1DataBarStacked ● printGS1DataBarStackedOmnidirectional ● printGS1DataBarExpandedStacked ● printPageModeGS1DataBarStacked ● printPageModeGS1DataBarStackedOmnidirectional ● printPageModeGS1DataBarExpandedStacked
GSIDATABAR_MODULE_SIZE_3	3ドット	
GSIDATABAR_MODULE_SIZE_4	4ドット	
GSIDATABAR_MODULE_SIZE_5	5ドット	
GSIDATABAR_MODULE_SIZE_6	6ドット	
GSIDATABAR_MODULE_SIZE_7	7ドット	
GSIDATABAR_MODULE_SIZE_8	8ドット	
GSIDATABAR_MODULE_SIZE_9	9ドット	
GSIDATABAR_MODULE_SIZE_10	10ドット	
GSIDATABAR_MODULE_SIZE_11	11ドット	
GSIDATABAR_MODULE_SIZE_12	12ドット	
GSIDATABAR_MODULE_SIZE_13	13ドット	
GSIDATABAR_MODULE_SIZE_14	14ドット	
GSIDATABAR_MODULE_SIZE_15	15ドット	
GSIDATABAR_MODULE_SIZE_16	16ドット	

⑨ 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)

⑩ 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

⑪ 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	<ul style="list-style-type: none"> ● printPDF417 ● printPageModePDF417
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	<ul style="list-style-type: none"> ● printQRcode ● printPageModeQRcode
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	

⑫ 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

⑬ 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_1	QRコードモデル1
QR_MODEL_2	QRコードモデル2

⑭ 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
CUT_FULL	用紙カット位置までの紙送り フルカット
CUT_PARTIAL	用紙カット位置までの紙送り パーシャルカット
CUT_NONE*1	カットなし

*1: printPageModeのみサポート

⑮ 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

定数名	説明
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

⑩ 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

⑪ 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

⑱ 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

⑲ 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

⑳ 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

②① 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

②② 保留データの出力指定 (OutputPendingData)

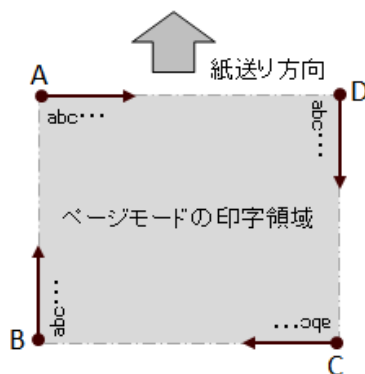
保留データの出力指定で利用する列挙型定数を下記に示します。

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

②③ 印字方向 (Direction)

ページモードの印字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
DIRECTION_LEFT_TO_RIGHT	始点: 左上(図のA)、印字方向: 左→右
DIRECTION_BOTTOM_TO_TOP	始点: 左下(図のB)、印字方向: 下→上
DIRECTION_RIGHT_TO_LEFT	始点: 右下(図のC)、印字方向: 右→左
DIRECTION_TOP_TO_BOTTOM	始点: 右上(図のD)、印字方向: 上→下



②④ 線種類 (LineStyle)

ページモードの線種類で利用する列挙型定数を下記に示します。

定数名	説明
LINestyle_THIN	細実線 (2ドット)
LINestyle_MEDIUM	中太実線 (4ドット)
LINestyle_THICK	太実線 (8ドット)

(4) メソッド詳細

① スタンダードモード・ページモード共通メソッド

スタンダードモード及びページモードで有効なメソッドです。connect直後はスタンダードモードになります。

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。

構文(a)のメソッドは、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)のメソッドは、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、「4.2.1(2)④ 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、「4.2.5 **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またはバーコードスキャナーに関する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。

バーコードスキャナーからのバーコードデータ受信はonBarcodeScannerReadData、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

バーコードスキャナーの接続はonBarcodeScannerChangedOnline、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

バーコードスキャナーの切断はonBarcodeScannerChangedOffline、及びsetBarcodeScannerListener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、「4.2.1(2)④ 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、「4.2.5 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またはバーコードスキャナーに関する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。

バーコードスキャナーからのバーコードデータ受信はonBarcodeScannerReadData、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

バーコードスキャナーの接続はonBarcodeScannerChangedOnline、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

バーコードスキャナーの切断はonBarcodeScannerChangedOffline、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

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、「4.2.1(2)④ 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、「4.2.5 **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。

• TCP/IP 接続中のライブラリのソケットの作成・削除について

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、socketKeepingTime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削除します。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またはバーコードスキャナーに関する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。

バーコードスキャナーからのバーコードデータ受信はonBarcodeScannerReadData、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

バーコードスキャナーの接続はonBarcodeScannerChangedOnline、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

バーコードスキャナーの切断はonBarcodeScannerChangedOffline、及びsetBarcodeScannerListenerにより、イベントとして通知できます。

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、「**4.2.5 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。
setBarcodeScannerListenerによって保持されている**BarcodeScannerListener**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。

バーコードスキャナーのコールバックを開始、または終了します。

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

パラメータ *listener* BarcodeScannerListener 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、「**4.2.5 PrinterExceptionクラス**」を参照してください。

説明 コールバックの対象は以下の通りです。

- ・バーコードデータの受信
- ・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

`listener`に`BarcodeScannerListener`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BarcodeScannerListenerインターフェイスのインスタンスは以下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。

cutPaper 用紙のカット

cutPaper 用紙のカット

用紙カット位置までの紙送りを行い、用紙をカットします。

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、「4.2.1(3)⑭ 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、「4.2.5 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。

openDrawer
キャッシュドローを開く

openDrawer キャッシュドロウを開く

指定されたキャッシュドローをオープンします。

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、「4.2.1(3)⑰ 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、「4.2.1(3)⑱ 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、「4.2.5 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
--------	--------

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**

外部ブザーを鳴動します。

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

パラメータ <i>buzzerPattern</i>	ブザーパターン 利用可能な定数は、「4.2.1(3)⑬ 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以下のいずれかの条件で停止します。 <ul style="list-style-type: none"> ・<i>buzzerCount</i>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
<i>buzzerCount</i>	鳴動回数(回) <i>buzzerCount</i> 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、「4.2.5 **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。

最新のプリンタステータスを取得します。

構文 (a) のメソッドは、プリンタステータスを戻り値で返します。

構文(b)のメソッドは、プリンタステータスをint型の配列に格納します。

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、「4.2.5 **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	オートカッタエラー	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	予約済み	固定	-
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	予約済み	-	固定
14	予約済み	-	固定
15	ドロフスイッチ入力状態	Low	High
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	予約済み	プリンター	その他
18	予約済み	固定	-
19～31	予約済み	-	固定

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、「4.2.5 **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以下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、「4.2.5 `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グラフィックスメモリに登録します。

構文(a)のメソッドは、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)のメソッドは、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

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、「4.2.1(3)② 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、「4.2.5 **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、「4.2.5 **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、「4.2.5 **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。

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。

プリンターからの各種応答取得

構文

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

パラメータ <i>id</i>	<p>応答種別定数</p> <p>利用可能な定数は、「4.2.1(2)⑥ 応答種別」を参照してください。</p>
<i>buf</i>	<p>取得した応答データを格納するバッファ</p> <p><i>id</i>で指定した応答データを<i>buf</i>で指定されたオブジェクトに格納します。</p> <p>応答種別定数によりバッファの型が異なります。</p> <p>バッファの型は、下表を参照してください。</p>

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）	
<i>buf</i>	長さ1のint型の配列を指定してください。 <i>buff[0]</i> には0～15(00h～0Fh)を指定してください。 応答取得成功時には、 <i>buff[0]</i> 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。
PRINTER_RESPONSE_USER_AREA （ユーザ領域の残り容量の送信）	
<i>buf</i>	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）	
<i>buf</i>	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）	
<i>buf</i>	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）	
<i>buf</i>	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: <i>buf.size()</i> = 3、 <i>buff[0]</i> = "22"、 <i>buff[1]</i> = "23"、 <i>buff[2]</i> = "24"など

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、「**4.2.5 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。

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、「**4.2.5 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cancelDiscoveryDevice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、「**4.2.5 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。

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。

エラー 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。

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。

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。

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。

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、「4.2.1(2)① 国際文字セット」を参照してください。

説明 `sendText`、`sendTextEx`、`sendDataFile`、`printPageModeText`、及び`printPageModeTextEx`にてテキストデータを送信した場合は、以下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

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、「4.2.1(2)① 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以下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`

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、「4.2.1(2)② コードページ」を参照してください。

説明 `sendText`、`sendTextEx`、`sendDataFile`、`printPageModeText`、及び`printPageModeTextEx`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。

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、「4.2.1(2)② 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以下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`

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、「4.2.1(2)④ 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。

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、「4.2.1(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。

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に関わらず設定が可能です。

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に関わらず設定または取得が可能です。

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 RP 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。

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、「4.2.1(3)㉑ 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、「**4.2.5 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

一括処理の手順は以下の通りです。

- 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以下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
- extarnalBuzzer
- sendBinary
- sendDataFile
- printLogo^{*1}
- enterPageMode
- exitPageMode
- setPageModeArea
- setPageModeDirection
- setPageModeLineSpacing
- printPageMode
- printPageModeText
- printPageModeTextEx
- printPageModeBarcode
- printPageModePDF417
- printPageModeQRcode
- printPageModeDataMatrix
- printPageModeMaxiCode
- printPageModeGS1DataBarStacked
- printPageModeGS1DataBarStackedOmnidirectional
- printPageModeGS1DataBarExpandedStacked
- sendPageModeBinary
- printPageModeImageFile
- printPageModeRectangle
- printPageModeLine
- printPageModeLogo^{*1}

*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。

② スタンダードモード専用メソッド

スタンダードモードで有効なメソッドです。スタンダードモード専用メソッドをページモード中に実行すると、**PrinterException**をスローします。

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、「**4.2.5 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)のメソッドは、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)のメソッドは、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**

パラメータ	<i>text</i>	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。
	<i>bold</i>	強調印字 利用可能な定数は、「4.2.1(3)① 強調印字(CharacterBold)」を参照してください。
	<i>underline</i>	アンダーライン 利用可能な定数は、「4.2.1(3)② アンダーライン(CharacterUnderline)」を参照してください。
	<i>reverse</i>	白黒反転印字 利用可能な定数は、「4.2.1(3)③ 白黒反転印字(CharacterReverse)」を参照してください。
	<i>font</i>	文字フォント 利用可能な定数は、「4.2.1(3)④ 文字フォント(CharacterFont)」を参照してください。
	<i>scale</i>	文字倍率 利用可能な定数は、「4.2.1(3)⑤ 文字倍率(CharacterScale)」を参照してください。
	<i>alignment</i>	位置揃え 利用可能な定数は、「4.2.1(3)⑥ 位置揃え(PrintAlignment)」を参照してください。
	<i>output</i>	保留データの出力指定 利用可能な定数は、 「4.2.1(3)㉔ 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、「4.2.5 **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)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。

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)のメソッドは、バーコードデータを文字列で指定します。

構文(b)のメソッドは、バーコードデータを文字列で指定し、バーコードの位置揃えとNW比を指定します。

構文(c)のメソッドは、バーコードデータをバイト配列で指定し、バーコードの位置揃えを指定します。

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
```

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、「4.2.1(3)⑦ バーコードシンボル (BarcodeSymbol)」を参照してください。

<code>text(data)</code>	プリンターに送信するバーコードデータ
-------------------------	--------------------

<i>moduleSize</i>	<p>バーコードの幅</p> <p>利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ (ModuleSize)」を参照してください。</p>
-------------------	--

<i>moduleHeight</i>	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。

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	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

<i>hriPosition</i>	HRI文字印字位置 利用可能な定数は、「4.2.1(3)⑨ HRI文字印字位置 (HriPosition)」を参照してください。
<i>hriFont</i>	HRI文字フォント 利用可能な定数は、「4.2.1(3)④ 文字フォント (CharacterFont)」を参照してください。
<i>alignment</i>	位置揃え 利用可能な定数は、「4.2.1(3)⑥ 位置揃え (PrintAlignment)」を参照してください。
<i>nwRatio</i>	NW比 利用可能な定数は、「4.2.1(3)⑩ NW比 (NwRatio)」を参照してください。 指定した <i>moduleSize</i> と <i>nwRatio</i> により、太エメントの幅が下表のように設定されます。

<i>moduleSize</i>	<i>nwRatio</i>		
	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、「4.2.5 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。

PDF417を印字します。

構文(a)のメソッドは、PDF417シンボルを指定します。

構文(b)のメソッドは、PDF417シンボルは通常のPDF417固定です。

```

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 <i>text</i>	プリンターに送信するバーコードデータ
<i>errorCorrection</i>	エラー訂正レベル 利用可能な定数は、「4.2.1(3)⑪ エラー訂正レベル(ErrorCorrection)」を参照してください。
<i>row</i>	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。
<i>column</i>	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。
<i>moduleSize</i>	公称細エレメント幅 利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ(ModuleSize)」を参照してください。
<i>moduleHeight</i>	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。
<i>alignment</i>	位置揃え 利用可能な定数は、「4.2.1(3)⑥ 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、「4.2.1(3)⑫ 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、「4.2.5 **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。

printQRcode

QRコードの印字

QRコードを印字します。

構文 (a) のメソッドは、QRコードモデルはQRコードモデル2固定です。

構文 (b) のメソッドは、QRコードモデルを指定します。

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、「4.2.1(3)⑪ 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、「4.2.1(3)⑧ 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、「4.2.1(3)⑥ 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、「4.2.1(3)⑬ QRコードモデル (QrModel)」を参照してください。

MaxiCodeを印字します。

構文	<pre>public void printMaxiCode(String text, MaxiCodeMode maxiCodeMode, PrintAlignment alignment) throws PrinterException</pre>	
パラメータ	<i>text</i>	<p>プリンターに送信するバーコードデータ</p> <ul style="list-style-type: none"> ● <i>maxiCodeMode</i>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。 ● <i>maxiCodeMode</i>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。
	<i>maxiCodeMode</i>	<p>MaxiCodeモード</p> <p>利用可能な定数は、「4.2.1(3)⑯ MaxiCodeモード(MaxiCodeMode)」を参照してください。</p>
	<i>alignment</i>	<p>位置揃え</p> <p>利用可能な定数は、「4.2.1(3)⑥ 位置揃え(PrintAlignment)」を参照してください。</p>
例外	<p>PrinterException</p> <p>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、「4.2.5 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。</p>	
注意	<p>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。</p>	

GS1 Databar Stackedを印字します。

構文	<pre>public void printGS1DataBarStacked(String text, ModuleSize moduleSize, PrintAlignment alignment) throws PrinterException</pre>	
パラメータ	<i>text</i>	<p>プリンターに送信するバーコードデータ</p> <p>'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。</p>
	<i>moduleSize</i>	<p>モジュールサイズ</p> <p>利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ(ModuleSize)」を参照してください。</p>

位置揃え

利用可能な定数は、「4.2.1(3)⑥ 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、「4.2.5 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。

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、「4.2.1(3)⑧ 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、「4.2.1(3)⑥ 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、「4.2.5 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。

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。

column カラム数
1行のセグメント数を指定します。
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、「4.2.1(3)⑧ 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、「4.2.1(3)⑥ 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、「4.2.5 **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。

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、「4.2.5 **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)のメソッドは、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)のメソッドは、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

指定可能なファイルサイズは、最大1MB(1048576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いません。
- .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、「4.2.1(3)⑥ 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、「4.2.1(3)⑩ 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、「4.2.5 **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

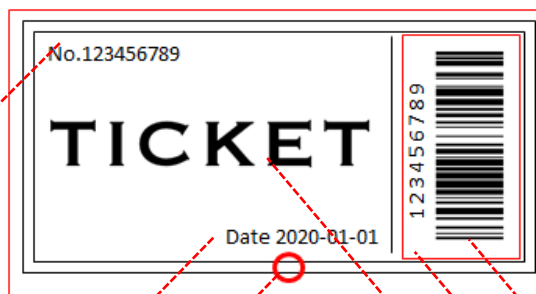
位置揃え
利用可能な定数は、「4.2.1(3)⑥ 位置揃え(PrintAlignment)」を参照してください。

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、「4.2.5 **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。

③ ページモード専用メソッド

ページモードを利用するための専用メソッドです。ページモードで印字する手順例を下記に示します。



① ページモードを開始します

```
mPrinterManager.enterPageMode();
```

② ページモードの印字領域を指定します

```
mPrinterManager.setPageModeArea(0, 0, 576, 355);
```

③ 矩形と罫線を指定します

```
mPrinterManager.printPageModeRectangle(0, 0, 575, 344, LineStyle.LINESTYLE_THIN);  
mPrinterManager.printPageModeRectangle(7, 7, 567, 336, LineStyle.LINESTYLE_THIN);  
mPrinterManager.printPageModeLine(404, 11, 404, 334, LineStyle.LINESTYLE_THIN);
```

④ 文字を指定します

```
mPrinterManager.printPageModeText(21, 37, "NO.123456789");  
mPrinterManager.printPageModeText(212, 330, "Date 2020-01-01");
```

⑤ イメージファイルを指定します

```
mPrinterManager.printPageModeImageFile(  
    10,  
    212,  
    Environment.getExternalStorageDirectory().getPath() + "/TicketImage.jpg",  
    Dithering.DITHERING_DISABLE);
```

⑥ ページモードの印字領域を指定します

```
mPrinterManager.setPageModeArea(404, 0, 163, 345);
```

⑦ 印字方向を指定します

```
mPrinterManager.setPageModeDirection(Direction.DIRECTION_BOTTOM_TO_TOP);
```

⑧ バーコードを指定します

```
mPrinterManager.printPageModeBarcode(  
    20,  
    132,  
    BarcodeSymbol.BARCODE_SYMBOL_CODE128,  
    new byte[]{0x67, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19x, 0x68},  
    ModuleSize.BARCODE_MODULE_WIDTH_2,  
    80,  
    HriPosition.HRI_POSITION_ABOVE,  
    CharacterFont.FONT_A);
```

⑨ ページモードを印字します

```
mPrinterManager.printPageMode(CuttingMethod.CUT_PARTIAL);
```

⑩ ページモードを終了します

```
mPrinterManager.exitPageMode();
```


ページモードを開始します。

構文 `public void enterPageMode() 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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドによりページモードが開始されます。以後は、ページモード専用メソッドとスタンダードモード・ページモード共通メソッドが使用できます。

exitPageModeを実行すると、ページデータバッファに保持している印字データを破棄し、スタンダードモードに移行します。

printPageModeを実行すると、ページデータバッファに保持している印字データを印字します。

ページモードを終了し、スタンダードモードに移行します。

構文 `public void exitPageMode() 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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 ページデータバッファに保持している印字データを破棄し、スタンダードモードに移行します。

ページモードの印字領域を指定します。

構文 `public void setPageModeArea(int x,
 int y,
 int width,
 int height) throws PrinterException`

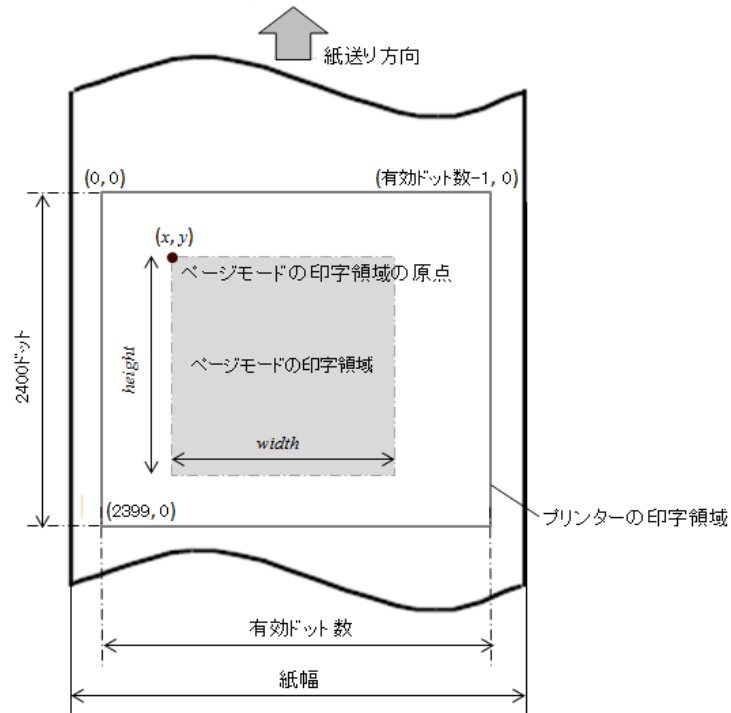
パラメータ *x* ページモードの印字領域の横方向の原点(ドット)
0はプリンターの印字領域の左端になります。

y ページモードの印字領域の縦方向の原点(ドット)
有効範囲は、0～2399です。
0は紙送りをしていない位置になります。

width ページモードの印字領域幅(ドット)

height ページモードの印字領域高さ(ドット)
有効範囲は、1～(2400-*y*)です。

x 及び $width$ の有効範囲を下記に示します。



プリンターのメモリスイッチ設定		有効ドット数	setPageModeArea	
MS4-4 (紙幅選択)	MS4-5 (有効ドット数選択)		x	$width$
80 mm	576	576	0 ~ 575	1 ~ (576 - x)
	512	512	0 ~ 511	1 ~ (512 - x)
58 mm	432	432	0 ~ 431	1 ~ (432 - x)
	360	360	0 ~ 359	1 ~ (360 - x)

有効ドット数はプリンターのメモリスイッチ設定により異なります。

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、「RP-F10シリーズ サーマルプリンタ取扱説明書」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、「4.2.5 **PrinterException**クラス」を参照してください。

説明

本メソッドの実行前に**enterPageMode**でページモードを開始してください。

enterPageModeでページモードを開始し、ページモード専用メソッドを実行した後に本メソッドを実行すると、追加してページモードの印字領域を指定できます。それまでに展開されているデータは保持されます。

本メソッド実行後は本メソッドで追加したページモードの印字領域に対して、ページモード専用メソッドのデータが展開されます。

enterPageModeメソッドを実行すると、ページモードの印字領域は、 $x=0$ 、 $y=0$ 、 $width$ =有効ドット数、 $height=2400$ になります。

ページモードの印字方向を指定します。

構文 `public void setPageModeDirection(Direction direction) throws PrinterException`

パラメータ *direction* 印字方向
利用可能な定数は、「4.2.1(3)②③ 印字方向 (Direction)」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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。
enterPageModeメソッドを実行すると、印字方向は左→右になります。

ページモードの改行量を指定します。

構文 `public void setPageModeLineSpacing(int lineSpacing) throws PrinterException`

パラメータ *lineSpacing* ページモードの改行量(ドット)
有効範囲は、0～255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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。
enterPageModeメソッドを実行すると、改行量は34ドットになります。

ページデータバッファに保持している印字データを印字します。

構文 `public void printPageMode(CuttingMethod cuttingMethod) throws PrinterException`

パラメータ *cuttingMethod* カット方法
利用可能な定数は、「4.2.1(3)⑭ 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、「4.2.5 **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 印字後も印字データは保持されます。印字データは以下のタイミングで破棄されます。
- `enterPageMode`を実行した場合
 - `disconnect`を実行した場合
 - `exitPageMode`を実行した場合

printPageModeText

ページモードのテキストデータの送信

テキストデータをページモードの印字領域に展開します。

構文 `public void printPageModeText(int startX, int startY, String text) throws PrinterException`

パラメータ *startX* 始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

startY 始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。

text 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、「4.2.5 **PrinterException**クラス」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ます。

本メソッドの実行前に`enterPageMode`でページモードを開始してください。

printPageModeTextEx

ページモードの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ページモードの印字領域に展開します。

構文 `public void printPageModeTextEx(int startX, int startY, String text, CharacterBold bold, CharacterUnderline underline, CharacterReverse reverse, CharacterFont font, CharacterScale scale) throws PrinterException`

パラメータ *startX* 始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

startY 始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。

text 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。

<i>bold</i>	強調印字 利用可能な定数は、「4.2.1(3)① 強調印字(CharacterBold)」を参照してください。
<i>underline</i>	アンダーライン 利用可能な定数は、「4.2.1(3)② アンダーライン(CharacterUnderline)」を参照してください。
<i>reverse</i>	白黒反転印字 利用可能な定数は、「4.2.1(3)③ 白黒反転印字(CharacterReverse)」を参照してください。
<i>font</i>	文字フォント 利用可能な定数は、「4.2.1(3)④ 文字フォント(CharacterFont)」を参照してください。
<i>scale</i>	文字倍率 利用可能な定数は、「4.2.1(3)⑤ 文字倍率(CharacterScale)」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、「4.2.5 PrinterException クラス」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ます。 本メソッドの実行前に enterPageMode でページモードを開始してください。

printPageModeBarcode

ページモードのバーコードの印字

バーコードをページモードの印字領域に展開します。

構文(a)のメソッドは、バーコードを文字列で指定します。

構文(b)のメソッドは、バーコードを文字列で指定し、バーコードのNW比を指定します。

構文(c)のメソッドは、バーコードをバイト配列で指定します。

構文 (a) `public void printPageModeBarcode(int startX,
int startY,
BarcodeSymbol barcodeSymbol,
String text,
ModuleSize moduleSize,
int moduleHeight,
HriPosition hriPosition,
CharacterFont hriFont) throws PrinterException`

```
(b) public void printPageModeBarcode(int startX,
                                     int startY,
                                     BarcodeSymbol barcodeSymbol,
                                     String text,
                                     ModuleSize moduleSize,
                                     int moduleHeight,
                                     HriPosition hriPosition,
                                     CharacterFont hriFont,
                                     NwRatio nwRatio) throws PrinterException
```

```
(c) public void printPageModeBarcode(int startX,
                                     int startY,
                                     BarcodeSymbol barcodeSymbol,
                                     byte[] data,
                                     ModuleSize moduleSize,
                                     int moduleHeight,
                                     HriPosition hriPosition,
                                     CharacterFont hriFont) throws PrinterException
```

パラメータ <i>startX</i>	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
<i>startY</i>	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
<i>barcodeSymbol</i>	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、「4.2.1(3)⑦ バーコードシンボル (BarcodeSymbol)」を参照してください。
<i>text(data)</i>	バーコードデータ
<i>moduleSize</i>	バーコードの幅 利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ (ModuleSize)」を参照してください。
<i>moduleHeight</i>	バーコードの高さ(ドット) <ul style="list-style-type: none"> ● <i>barcodeSymbol</i>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。

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	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

<i>hriPosition</i>	<p>HRI文字印字位置</p> <p>利用可能な定数は、「4.2.1(3)⑨ HRI文字印字位置(HriPosition)」を参照してください。</p>
<i>hriFont</i>	<p>HRI文字フォント</p> <p>利用可能な定数は、「4.2.1(3)④ 文字フォント(CharacterFont)」を参照してください。</p>
<i>nwRatio</i>	<p>NW比</p> <p>利用可能な定数は、「4.2.1(3)⑩ NW比(NwRatio)」を参照してください。</p> <p>指定した<i>moduleSize</i>と<i>nwRatio</i>により、太エレメントの幅が下表のように設定されます。</p>

<i>moduleSize</i>	<i>nwRatio</i>		
	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、「4.2.5 **PrinterException**クラス」を参照してください。
- 説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。
- 注意 バーコードの印字データは、その他の印字データと重ならないように展開してください。
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。

printPageModePDF417

ページモードのPDF417の印字

PDF417をページモードの印字領域に展開します。

構文(a)のメソッドは、PDF417シンボルを指定します。

構文(b)のメソッドは、PDF417シンボルは通常のPDF417固定です。

- 構文 (a) `public void printPageModePDF417(int startX,`
`int startY,`
`String text,`
`ErrorCorrection errorCorrection,`
`int row,`
`int column,`
`ModuleSize moduleSize,`
`int moduleHeight,`
`Pdf417Symbol pdf417Symbol)` throws **PrinterException**
- (b) `public void printPageModePDF417(int startX,`
`int startY,`
`String text,`
`ErrorCorrection errorCorrection,`
`int row,`
`int column,`
`ModuleSize moduleSize,`
`int moduleHeight)` throws **PrinterException**

パラメータ	<i>startX</i>	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>startY</i>	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>text</i>	バーコードデータ
	<i>errorCorrection</i>	エラー訂正レベル 利用可能な定数は、「4.2.1(3)⑪ エラー訂正レベル(ErrorCorrection)」を参照してください。
	<i>row</i>	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。
	<i>column</i>	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。
	<i>moduleSize</i>	公称細エレメント幅 利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ(ModuleSize)」を参照してください。
	<i>moduleHeight</i>	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。
	<i>pdf417Symbol</i>	PDF417のシンボル 利用可能な定数は、「4.2.1(3)⑫ 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、「4.2.5 PrinterException クラス」を参照してください。
説明		本メソッドの実行前に enterPageMode でページモードを開始してください。
注意		バーコードの印字データは、その他の印字データと重ならないように展開してください。 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。

QRコードをページモードの印字領域に展開します。

構文(a)のメソッドは、QRコードモデルはQRコードモデル2固定です。

構文(b)のメソッドは、QRコードモデルを指定します。

構文 (a) `public void printPageModeQRcode(int startX,
int startY,
String text,
ErrorCorrection errorCorrection,
ModuleSize moduleSize) throws PrinterException`

(b) `public void printPageModeQRcode(int startX,
int startY,
String text,
ErrorCorrection errorCorrection,
ModuleSize moduleSize,
QrModel model) throws PrinterException`

パラメータ	<i>startX</i>	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>startY</i>	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>text</i>	バーコードデータ バージョンは構文(a)、(b)のいずれでも <i>text</i> で設定したデータ数に応じて自動設定されます。
	<i>errorCorrection</i>	エラー訂正レベル 利用可能な定数は、「4.2.1(3)⑪ エラー訂正レベル(ErrorCorrection)」を参照してください。
	<i>moduleSize</i>	モジュールサイズ 利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ(ModuleSize)」を参照してください。
	<i>model</i>	QRコードモデル 利用可能な定数は、「4.2.1(3)⑬ 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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。

注意 バーコードの印字データは、その他の印字データと重ならないように展開してください。
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。

Data Matrixをページモードの印字領域に展開します。

```
構文      public void printPageModeDataMatrix(int startX,
                                             int startY,
                                             String text,
                                             DataMatrixModule dataMatrixModule,
                                             ModuleSize moduleSize) throws PrinterException
```

パラメータ	<i>startX</i>	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>startY</i>	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>text</i>	バーコードデータ
	<i>dataMatrixModule</i>	Data Matrixモジュール数 利用可能な定数は、「4.2.1(3)⑮ Data Matrixモジュール (DataMatrixModule)」を参照してください。
	<i>moduleSize</i>	モジュールサイズ 利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ (ModuleSize)」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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。

注意 バーコードの印字データは、その他の印字データと重ならないように展開してください。
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。

MaxiCodeをページモードの印字領域に展開します。

```
構文      public void printPageModeMaxiCode(int startX,
                                             int startY,
                                             String text,
                                             MaxiCodeMode maxiCodeMode) throws PrinterException
```

パラメータ	<i>startX</i>	始点からの横方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>startY</i>	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
	<i>text</i>	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、「4.2.1(3)⑯ MaxiCodeモード(MaxiCodeMode)」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、「4.2.5 **PrinterException**クラス」を参照してください。
- 説明** 本メソッドの実行前に`enterPageMode`でページモードを開始してください。
- 注意** バーコードの印字データは、その他の印字データと重ならないように展開してください。
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。

printPageModeGS1DataBarStacked

ページモードのGS1 Databar Stackedの印字

GS1 Databar Stackedをページモードの印字領域に展開します。

構文 `public void printPageModeGS1DataBarStacked(int startX,
int startY,
String text,
ModuleSize moduleSize) throws PrinterException`

- パラメータ** *startX* 始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。
- startY* 始点からの縦方向の基準点(ドット)
有効範囲は、0～2399です。
- text* 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、「4.2.1(3)⑧ モジュールサイズ(ModuleSize)」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、「4.2.5 **PrinterException**クラス」を参照してください。
- 説明** 本メソッドの実行前に`enterPageMode`でページモードを開始してください。
- 注意** バーコードの印字データは、その他の印字データと重ならないように展開してください。

ページモードのGS1 Databar Stacked Omni-directionalの印字

```
public void printPageModeGS1DataBarStackedOmnidirectional(int startX,
                                                         int startY,
                                                         String text,
                                                         int moduleHeight,
                                                         ModuleSize moduleSize) throws PrinterException
```

例外	<p>PrinterException</p> <p>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、「4.2.5 PrinterExceptionクラス」を参照してください。</p>
説明	<p>本メソッドの実行前にenterPageModeでページモードを開始してください。</p>
注意	<p>バーコードの印字データは、その他の印字データと重ならないように展開してください。</p>

ページモードのGS1 Databar Expanded Stackedの印字

```
public void printPageModeGS1DataBarExpandedStacked(int startX,
                                                    int startY,
                                                    String text,
                                                    int column,
                                                    ModuleSize moduleSize) throws PrinterException
```

パラメータ *startX* 始点からの横方向の基準点(ドット)
有効範囲は、0～2399です。

<i>startY</i>	始点からの縦方向の基準点(ドット) 有効範囲は、0～2399です。
<i>text</i>	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。
<i>column</i>	カラム数 1行のセグメント数を指定します。 有効範囲は、2～20の偶数です。
<i>moduleSize</i>	モジュールサイズ 利用可能な定数は、「4.2.1(3)⑧ モジュールサイズ (ModuleSize)」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、「4.2.5 PrinterException クラス」を参照してください。
説明	本メソッドの実行前に enterPageMode でページモードを開始してください。
注意	バーコードの印字データは、その他の印字データと重ならないように展開してください。

sendPageModeBinary

ページモードのバイナリデータの送信

バイナリデータをページモードの印字領域に展開します。

構文 `public void sendPageModeBinary(byte [] binary) throws PrinterException`

パラメータ *binary* バイナリデータ
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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。

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。

注意 送信するデータによっては期待しない動作をする場合があります。実機にて事前に確認してください。

```
構文 public void printPageModelImageFile(int startX,
                                           int startY,
                                           String fileName,
                                           Dithering dithering) 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、「**4.2.5 PrinterExceptionクラス**」を参照してください。

説明 本メソッドの実行前にenterPageModeでページモードを開始してください。

```
構文      public void printPageModeRectangle(int startX,
                                           int startY,
                                           int endX,
                                           int endY,
                                           LineStyle lineStyle) throws PrinterException
```

startY 始点からの縦方向の描画開始位置(ドット)
有効範囲は、0～2399です。

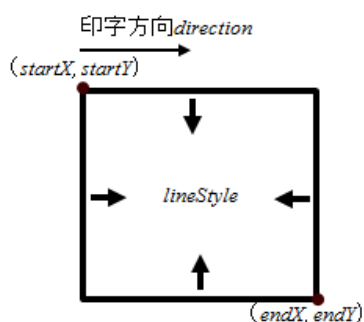
endX 始点からの横方向の描画終了位置(ドット)
有効範囲は、0～2399です。

endY 始点からの縦方向の描画終了位置(ドット)
有効範囲は、0～2399です。

lineStyle 線種類
利用可能な定数は、「4.2.1(3)④ 線種類 (LineStyle)」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、「4.2.5 **PrinterException**クラス」を参照してください。

説明 本メソッドの実行前に**enterPageMode**でページモードを開始してください。
矩形は**setPageModeDirection**の*direction*に対して下記のように展開します。



描画イメージに対するパラメータの設定例を下記に示します。
例: 始点から240ドット(30mm)、線種類を中太実線(4ドット)の正四角形を描画する場合

描画イメージ	パラメータ
<p>(<i>startX=0, startY=0</i>)</p> <p><i>lineStyle=</i> LINESTYLE_MEDIUM (4ドット)</p> <p>(<i>endX=239, endY=239</i>)</p>	<p><i>startX</i> 0</p> <p><i>startY</i> 0</p> <p><i>endX</i> 239</p> <p><i>endY</i> 239</p> <p><i>lineStyle</i> LINESTYLE_MEDIUM</p>

ページモードの印字領域に罫線を展開します。

```
構文      public void printPageModeLine(int startX,
                                     int startY,
                                     int endX,
                                     int endY,
                                     LineStyle lineStyle) throws PrinterException
```

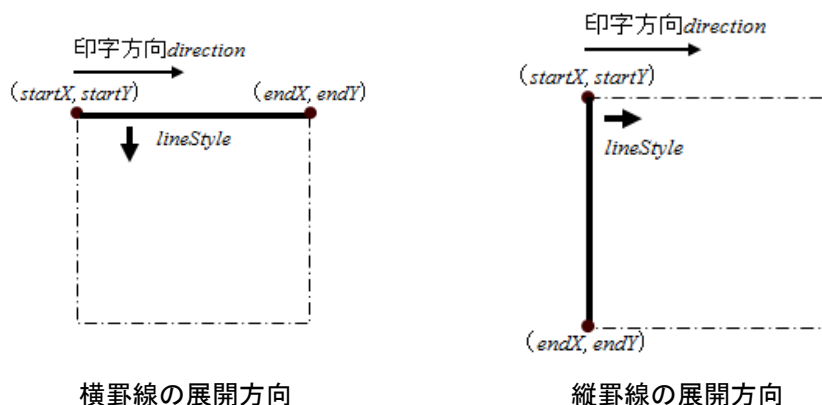
パラメータ	<i>startX</i>	始点からの横方向の罫線開始位置(ドット) 有効範囲は、0～2399です。
	<i>startY</i>	始点からの縦方向の罫線開始位置(ドット) 有効範囲は、0～2399です。
	<i>endX</i>	始点からの横方向の罫線終了位置(ドット) 有効範囲は、0～2399です。
	<i>endY</i>	始点からの縦方向の罫線終了位置(ドット) 有効範囲は、0～2399です。
	<i>lineStyle</i>	線種類 利用可能な定数は、「4.2.1(3)④ 線種類(LineStyle)」を参照してください。

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、「**4.2.5 PrinterExceptionクラス**」を参照してください。

説明 本メソッドの実行前にenterPageModeでページモードを開始してください。

本メソッドでは、斜線は描画できません。

罫線はsetPageModeDirectionのdirectionに対して下記のように展開します。



例：始点から240ドット（30mm）、線種類を中太実線（4ドット）の正四角形の横罫線を描画する場合

例：始点から240ドット(30mm)、線種類を中太実線(4ドット)の正四角形の縦罫線を描画する場合

ページモードのロゴの印字

構文 `public void printPageModeLogo(int startX, int startY, String id) throws PrinterException`

説明 本メソッドの実行前にenterPageModeでページモードを開始してください。

4.2.2 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、「4.2.2(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。

4.2.3 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。

4.2.4 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（プリンタ名）、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（プリンタ名）の文字列を取得します。

構文 public String getPrinterModelName()

戻り値 プリンターモデル名（プリンタ名）

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。

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

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

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。

構文 `public boolean getIsBonded()`

戻り値 true ペアリング済み
 false 未ペアリング

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

4.2.5 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Bluetooth接続に問題があります。	-21
	AndroidデバイスとプリンターのUSB接続に問題があります。	
ERROR_OFFLINE	切断状態またはプリンターがオフラインです。	-22
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いないか、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

定数名	説明	値
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_PAGE_MODE_SPECIFIED	ページモードが指定されています。	-511
ERROR_PAGE_MODE_NOT_SPECIFIED	ページモードが指定されていません。	-512
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、「4.2.5(2) 定数一覧」を参照してください。

4.2.6 CallbackFunctionListenerインターフェイス

CallbackFunctionListenerインターフェイスは、プリンタ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	プリンタステータスの変化イベント

(2) メソッド詳細

onStatusChanged プリンタ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* プリンタステータス

説明 本メソッドは、以下のタイミングで呼び出されます。
 ・setCallbackFunctionListenerが実行された場合
 ・プリンタステータスが変わった場合

プリンタ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。

4.2.7 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。

(1) メソッド一覧

BarcodeScannerListener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
<code>onBarcodeScannerReadData</code>	バーコードデータの受信イベント
<code>onBarcodeScannerChangedOnline</code>	バーコードスキャナーの接続イベント
<code>onBarcodeScannerChangedOffline</code>	バーコードスキャナーの切断イベント

(2) メソッド詳細

`onBarcodeScannerReadData` バーコードデータの受信イベント

構文 `public void onBarcodeScannerReadData(byte[] data)`

パラメータ `data` バーコードスキャナーから受信したバーコードデータ

説明 本メソッドは、`setBarcodeScannerListener`によりバーコードスキャナーからのバーコードデータ受信のコールバックが開始され、バーコードデータを受信した場合に呼び出されます。

バーコードデータの受信は、`isConnect`が`true`でプリンターと通信中に通知されます。

インターフェイスのため実装はありません。

バーコードスキャナーイベント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。

`onBarcodeScannerChangedOnline` バーコードスキャナーの接続イベント

構文 `public void onBarcodeScannerChangedOnline()`

説明 本メソッドは、以下のタイミングで呼び出されます。

- ・バーコードスキャナーがプリンターと接続状態で`setBarcodeScannerListener`が実行された場合
- ・プリンターにバーコードスキャナーが接続された場合

バーコードスキャナーの接続イベントは、`isConnect`が`true`でプリンターと通信中に通知されます。

インターフェイスのため実装はありません。

バーコードスキャナーイベント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。

構文 `public void onBarcodeScannerChangedOffline()`

説明 本メソッドは、以下のタイミングで呼び出されます。

- ・バーコードスキャナーがプリンターと非接続状態でsetBarcodeScannerListenerが実行された場合
- ・プリンターからバーコードスキャナーが切断された場合

バーコードスキャナーの切断イベントは、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。

バーコードスキャナーイベント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。

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。

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。

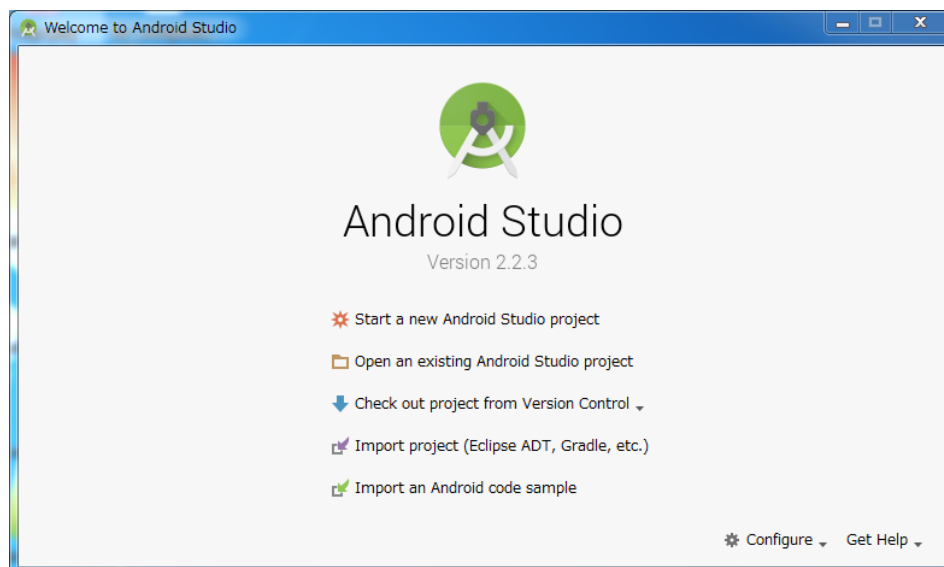
5.1 インストール

サンプルプログラムをインストールしてください。

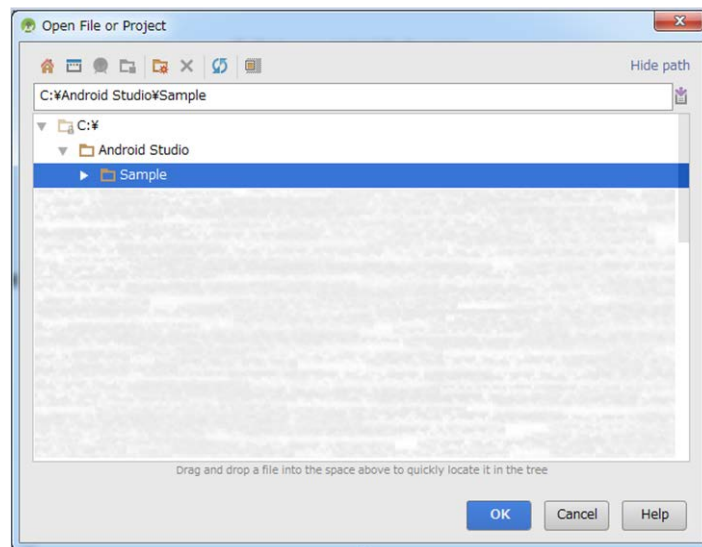
Androidアプリケーションの開発環境が整っていることを確認してください。必要な開発環境については、「3章 ライブラリの利用方法」を参照してください。

インストール手順は以下の通りです。

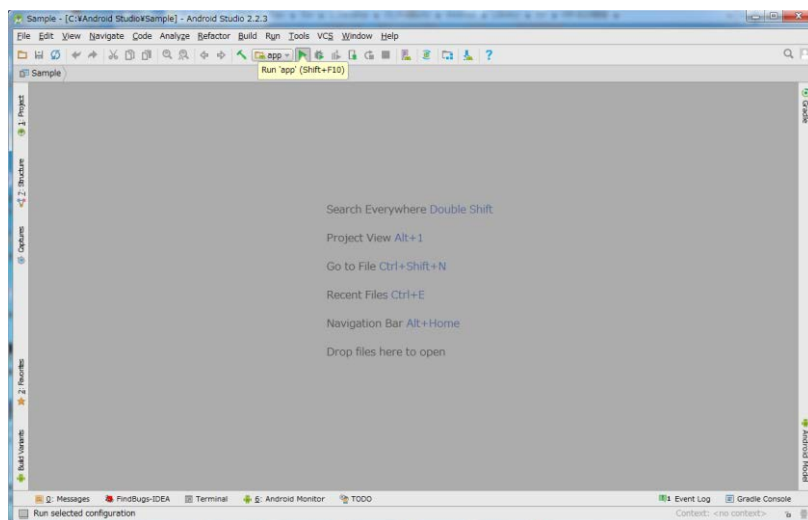
- (1) Sampleフォルダを任意の位置に配置してください。
- (2) Android Studioを起動し、「Open an existing Android Studio project」をクリックしてください。



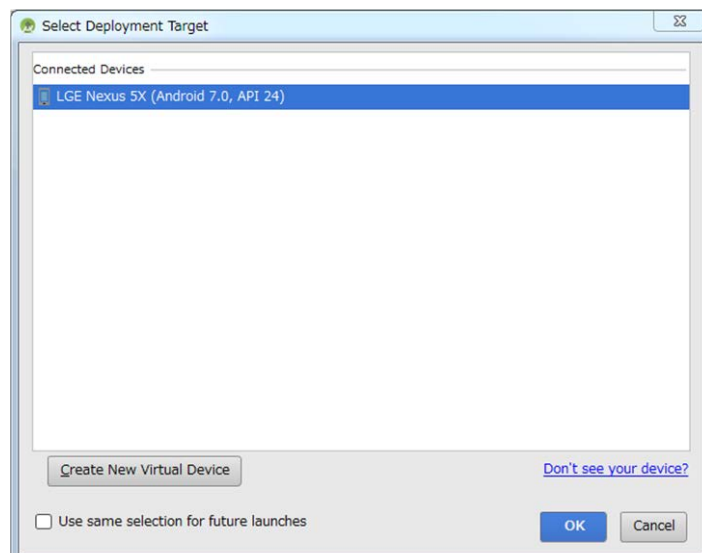
(3) 手順(1)で配置したフォルダを選択し、[OK]をクリックしてください。



(4) [Run 'app']をクリックしてください。



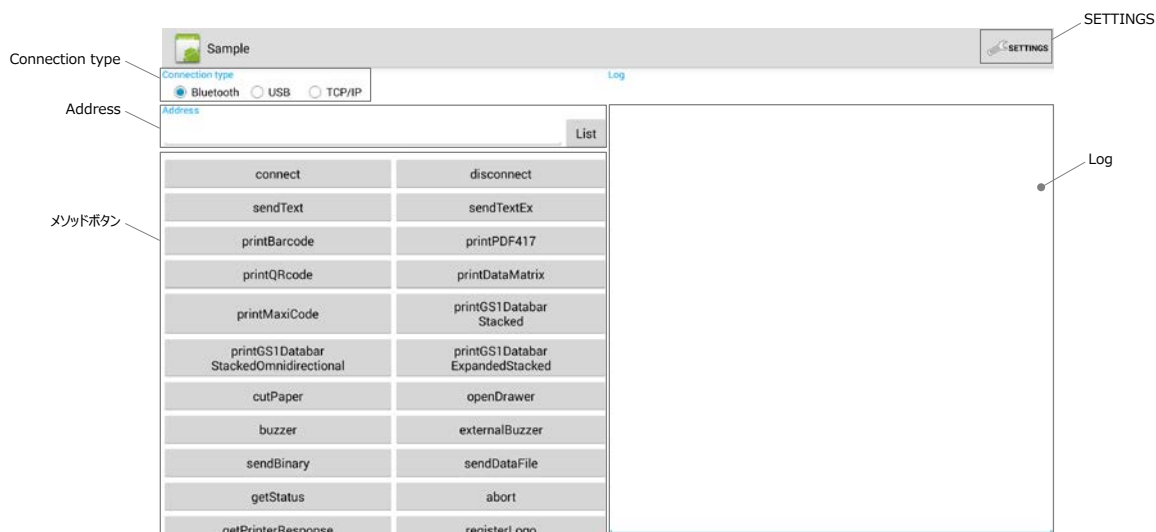
(5) デバイスを選択し、[OK]をクリックしてください。



5.2 画面構成

サンプルプログラムの画面について説明します。

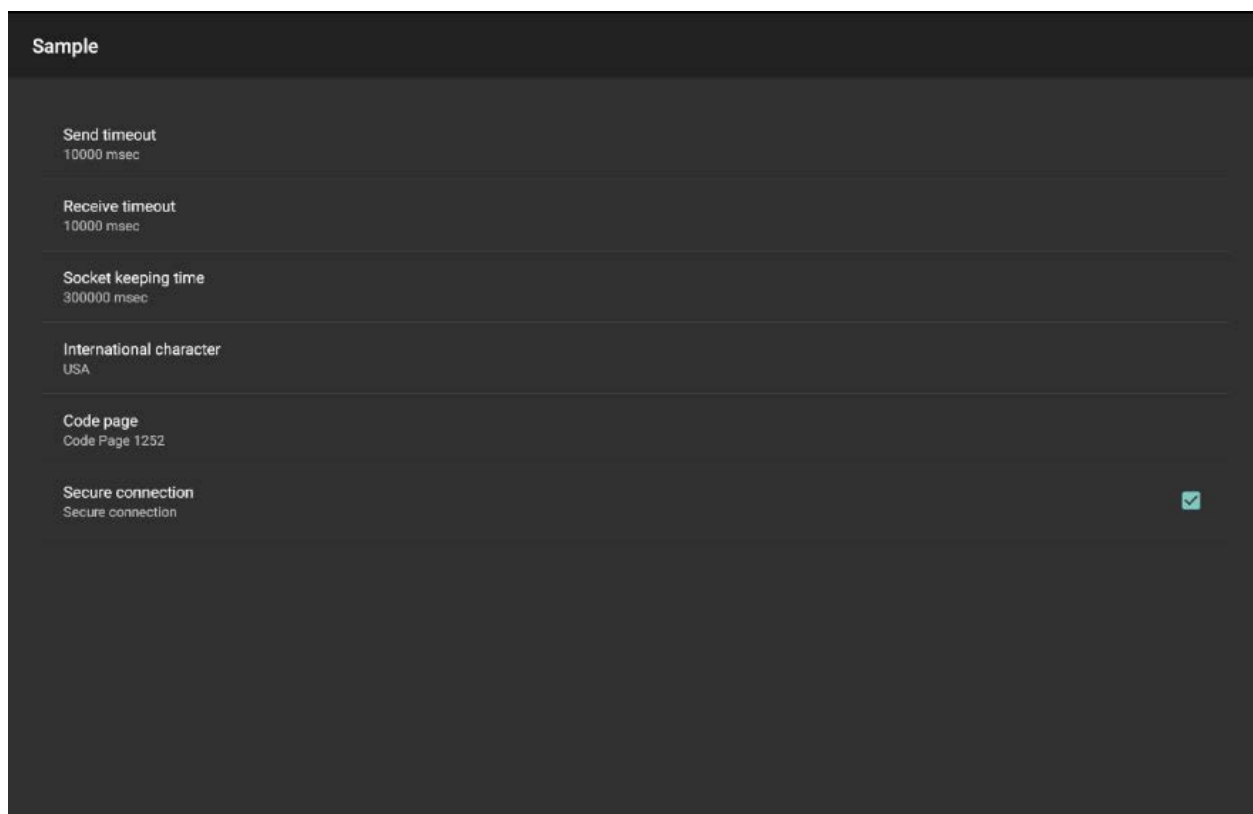
5.2.1 メイン画面



項目	説明
Connection type	プリンターへの接続種別を選択します。
Address	プリンターのアドレスを指定します。 手動入力の場合： Bluetoothで接続する場合は、Bluetoothアドレスを入力してください。 TCP/IPで接続する場合はMACアドレスを入力してください。 自動入力の場合： [List]ボタンをタップすると、startDiscoveryPrinter(Bluetooth)により発見されたプリンターのプリンター情報が一覧で表示されます。 表示された一覧の中からプリンターを選択すると、Bluetoothアドレスが自動入力されます。
メソッドボタン	各メソッドを実行できるボタンです。 スクロールすると、表示されないメソッド及びプロパティを見ることができます。 各メソッドの詳細については、「4章 ライブラリの機能」を参照してください。
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、画面上の◀をタップしてください。
Log	「メソッドボタン」の各メソッドを実行すると、メソッド実行ログが表示されます。

5.2.2 [SETTINGS]画面

[SETTINGS]では各種設定機能を表示します。



The screenshot displays a settings interface with a dark background. At the top left, the word "Sample" is written in white. Below it, several settings are listed, each with a label and a value:

- Send timeout**: 10000 msec
- Receive timeout**: 10000 msec
- Socket keeping time**: 300000 msec
- International character**: USA
- Code page**: Code Page 1252
- Secure connection**: Secure connection ☒

5.3 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。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	φ	£	¥	℔	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¿	»		
B0	☐	☐	☐		†	‡	§	¶	§	§	§	§	§	§	§	§
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	∩
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	n	2	■	

図 A-1 CODE_PAGE_437 (USA, Standard Europe)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80																
90																
A0	。	「	」	、	・	ヲ	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ッ	
B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	ゝ	。
E0																
F0																

図 A-2 CODE_PAGE_KATAKANA

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶	¶	¶	¶	¢	¥	₱
C0	⊥	⊥	⊥	⊥	⊥	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	α
D0	ð	Đ	Ê	Ë	È	Í	Î	Ï	⌋	⌋	■	■	■	■	■	■
E0	ó	β	ô	ò	õ	õ	μ	þ	þ	ú	û	ù	ý	ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	…	.	1	3	2	■	

図 A-3 CODE_PAGE_850 (Multilingual)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ã	à	Á	ç	ê	Ê	è	Í	Ô	ì	Ã	Â
90	É	À	È	ô	õ	ò	Ú	ù	Ì	Õ	Ü	¢	£	Ù	Þ	Ó
A0	á	í	ó	ú	ñ	Ñ	ä	ö	ï	ò	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	L	L	T		-	+	+	+	+	+	+	+	+	+	+	+
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

图 A-4 CODE_PAGE_860 (Portuguese)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	À	à	¶	ç	ê	ë	è	ï	î	≡	À	§
90	É	È	Ê	ô	Ë	Ï	Ô	Ù	⊗	Ô	Ü	¢	£	Ù	Ù	f
A0	!	'	ó	ú	..	3	-	î	¬	¬	½	¼	¾	«	»	
B0	☐	☐	☐													
C0	L	L	T		-	+	+	+	+	+	+	+	+	+	+	+
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

图 A-5 CODE_PAGE_863 (Canadian-French)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	í	«	»	
B0	☐	☐	☐		†	‡	§	¶	§		¶		¶		¶	
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

图 A-6 CODE_PAGE_865 (Nordic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Ş	ş
A0	á	í	ó	ú	ñ	Ñ	Ğ	ğ	¿	®	¬	½	¼	í	«	»
B0	☐	☐	☐		†	‡	§	¶	§		¶		¶		¶	
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	ó	β	ô	ò	õ	õ	μ		×	ú	û	ü	ì	ÿ	-	'
F0	-	±	¾	¶	§	÷			°	..	.	1	3	2	■	

图 A-7 CODE_PAGE_857 (Turkish)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π
90	P	Σ	T	Υ	Φ	X	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ
A0	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	ς	τ	υ	φ	χ	ψ
B0	⋈	⋈	⋈		†	‡		π	‡			π				
C0	L	⊥	T	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	ω	ά	έ	ή	ϊ	ί	ό	ύ	ϋ	ώ	Ά	Έ	Ή	Ί	Ό	Υ
F0	Ω	±	≥	≤	İ	ÿ	÷	≈	°	•	•	√	n	2	■	

图 A-8 CODE_PAGE_737 (Greek)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	•	-	-	~	™	š	<	Œ		ž	
90											š	>	œ		ž	ÿ
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

图 A-9 CODE_PAGE_1252 (Latin)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
90	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B0	␣	␣	␣													
C0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
D0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
E0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0	Ё	ё	Є	є	İ	ı	Ÿ	ÿ	°	•	•	√	№	α	■	

☒ A-10 CODE_PAGE_866 (Russian)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	û	ç	ł	ë	ő	ö	î	ž	Ä	Ć	
90	É	Í	í	ô	ö	Ĺ	ĺ	Š	š	Ö	Ü	Ť	ť	Ł	×	č
A0	á	í	ó	ú	À	à	Ž	ž	Ę	ę	¬	ž	Č	š	«	»
B0	␣	␣	␣			Á	Â	Ě	Š					Ž	ž	ı
C0	␣	␣	␣	␣	␣	Ä	ä	Ľ	ľ	␣	␣	␣	␣	␣	␣	␣
D0	đ	Đ	Ď	Ě	ď	Ň	í	î	ě	Ĵ	Ĵ	■	■	Ť	Ů	■
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	ú	ý	ý	ť	´	
F0	-	"	˘	˘	˘	§	÷	˚	˚	˚	˚	ú	ř	ř	■	

☒ A-11 CODE_PAGE_852 (Eastern Europe)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©			¶	¶	¢	¥	₱
C0	L	L	T	T	—	+	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	Ј	Г	■	■	І	Ì	■
E0	Ó	β	Ô	Ò	Õ	μ	þ	þ	Ú	Û	Ü	ý	Ý	-	'	
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2		■

☒ A-12 CODE_PAGE_858 (Euro)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	Ђ	ѓ	Ѓ	ё	Ё	є	Є	ѕ	Ѕ	і	І	ї	Ї	ј	Ј
90	љ	Љ	њ	Њ	ћ	Ћ	ќ	Ќ	џ	Џ	џ	џ	џ	џ	џ	џ
A0	а	А	б	Б	в	В	г	Г	д	Д	е	Е	ф	Ф	г	Г
B0	☐	☐	☐			х	Х	и	И			¶	¶	й	Й	₱
C0	L	L	T	T	—	+	к	К	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	л	Л	м	М	н	Н	о	О	п	П	Г	■	■	П	я	■
E0	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	Ь	№
F0	-	ы	Ы	э	Э	ш	Ш	э	Э	щ	Щ	ч	Ч	§		■

☒ A-13 CODE_PAGE_855 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	°	•	•	√	■	-		+	+	+	+	+	+	+	+	+
90	β	∞	φ	±	½	¼	≈	«	»	لَا	لَا	لَا	لَا	لَا	لَا	لَا
A0	-	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	س	س	س	س	س
C0	¢	ء	آ	أ	ؤ	ع	ئ	ب	ا	ث	ة	ج	ح	خ	د	ذ
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	م	ن
E0	-	ف	ق	ك	م	ن	ه	و	ي	ض	ع	غ	ف	ق	ك	م
F0	-	ن	ه	و	ي	ض	ع	غ	ف	ق	ك	م	ن	ه	و	ي

图 A-14 CODE_PAGE_864 (Arabic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	…	†	‡	§	Š	Š	Š	Š	Š	Š	Š
90	‘	’	“	”	•	-	-	-	™	š	š	š	š	š	š	š
A0	˘	˘	Ł	ł	Ą	ą	Ś	ś	©	§	«	»	®	Ž	ž	ž
B0	°	±	ı	ı	μ	¶	•	•	•	•	»	»	»	»	»	»
C0	Ř	Á	Â	Ă	Ä	Á	Ć	Ç	Č	É	Ê	Ë	Ě	Í	Î	Ď
D0	Đ	Ń	Ň	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Ů	Ů	Ý	Ů	Ů
E0	ř	á	â	ă	ä	í	ć	ç	č	é	ê	ë	ě	í	î	ď
F0	đ	ń	ň	ó	ô	õ	ö	÷	ř	ů	ú	ú	ú	ý	ı	ı

图 A-15 CODE_PAGE_1250 (Central European)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	ѓ	;	ѓ	„	...	†	‡	€	‰	Љ	<	Њ	ќ	ћ	џ
90	ђ	‘	;	“	”	•	-	-	™	Љ	>	њ	ќ	ћ	џ	
A0	ѳ	ѵ	Ј	Ѡ	Г	І	Ѕ	Ё	©	Є	«	¬	-	®	İ	
B0	°	±	І	і	г	μ	¶	•	ё	№	є	»	ј	ѕ	ѕ	ї
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

☒ A-16 CODE_PAGE_1251 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	;	ƒ	„	...	†	‡	‰		<					
90		‘	;	“	”	•	-	-	™		>					
A0	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
B0	°	±	²	³	´	μ	¶	•	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
C0	ı	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
D0	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί
E0	ύ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ό	ύ	ώ	

☒ A-17 CODE_PAGE_1253 (Greek)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	”	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	•	-	-	~	™	š	<	Œ			
90		‘	’	“	”	•	-	-	~	™	š	>	œ			ÿ
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

图 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	#	α	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	\$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	Pt	\$	@	ı	Ñ	ı	^	`	..	ñ	}	~
COUNTRY_JAPAN	#	\$	@	[¥]	^	`	{		}	~
COUNTRY_NORWAY	#	α	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	ı	Ñ	ı	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	\$	á	ı	Ñ	ı	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	\$	@	[\]	^	`	{		}	~

図 A-19 国際文字セット



セイコーインスツル株式会社

プリントシステム事業部

千葉県千葉市美浜区中瀬 1-8 〒261-8507

電話番号：043-211-1212（直通） ファクシミリ：043-211-8037

ホームページ：<https://www.sii-ps.com>

大阪営業所

大阪府大阪市北区豊崎 3-2-1 淀川 5 番館 〒531-0072

電話番号：06-7711-0858（直通） ファクシミリ：06-7711-0856

©本書の内容は、製品の改良に伴い、予告なしに変更することがあります。
