



# 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.06

[対象製品]  
MP-B20シリーズ

セイコーインスツル株式会社

Rev.01	2017年	3月
Rev.02	2018年	2月
Rev.03	2019年	2月
Rev.04	2019年	8月
Rev.05	2020年	3月
Rev.06	2022年	3月

©セイコーインスツル株式会社 2017-2022

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

---

本書では、セイコーインスツル株式会社（以下、SII）製MP-B20シリーズ用「SII Print Class Library for Android™」（以下、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
MP-B20シリーズ	Bluetooth
	USB

## 用語

---

本書で使用する用語について以下に記載します。

用語	内容
プリンターコマンド	「MP-B20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	動作条件	2-2
3章	ライブラリの利用方法	3-1
3.1	Androidアプリケーションの開発環境	3-1
3.2	提供ファイル	3-2
3.3	Android Studioのプロジェクトへの組み込み	3-3
3.4	作成したAndroidアプリケーションのAndroidデバイスでの利用	3-5
3.5	注意事項	3-5
4章	ライブラリの機能	4-1
4.1	APIリファレンス	4-1
4.1.1	PrinterManagerクラス	4-2
(1)	メソッド一覧	4-2
(2)	定数一覧	4-3
①	プリンターモデル	4-3
②	応答種別	4-3
③	バッテリー残量レベル	4-4
④	国際文字セット	4-4
⑤	コードページ	4-5
⑥	ポート種別	4-5
⑦	バーコード、PDF417専用	4-6
(3)	列挙型定数一覧	4-6
①	強調印字 (CharacterBold)	4-6
②	アンダーライン (CharacterUnderline)	4-6
③	白黒反転印字 (CharacterReverse)	4-6
④	倒立印字 (CharacterInversion)	4-7
⑤	文字フォント (CharacterFont)	4-7
⑥	文字倍率 (CharacterScale)	4-7
⑦	位置揃え (PrintAlignment)	4-8
⑧	保留データの出力指定 (OutputPendingData)	4-8
⑨	バーコードシンボル (BarcodeSymbol)	4-8

⑩	モジュールサイズ (ModuleSize)	4-9
⑪	HRI文字印字位置 (HriPosition)	4-11
⑫	NW比 (NwRatio)	4-11
⑬	エラー訂正レベル (ErrorCorrection)	4-12
⑭	PDF417シンボル (Pdf417Symbol)	4-12
⑮	QRコードモデル (QrModel)	4-12
⑯	Data Matrixモジュール (DataMatrixModule)	4-13
⑰	MaxiCodeモード (MaxiCodeMode)	4-14
⑱	カット方法 (CuttingMethod)	4-14
⑲	ディザリング (Dithering)	4-14
⑳	一括処理選択 (TransactionFunction)	4-14
(4)	メソッド詳細	4-15
	PrinterManager	4-15
	connect	4-15
	connect	4-16
	disconnect	4-16
	setBarcodeScannerListener	4-17
	バーコードスキャナーのコールバック	
	開始・終了	4-17
	sendText	4-17
	sendTextEx	4-17
	printBarcode	4-19
	printPDF417	4-22
	printQRcode	4-23
	printDataMatrix	4-24
	printMaxiCode	4-25
	printGS1DataBarStacked	4-25
	GS1 Databar Stackedの印字	4-25
	printGS1DataBarStackedOmnidirectional	4-26
	GS1 Databar Stacked Omni-directionalの印字	4-26
	printGS1DataBarExpandedStacked	4-27
	GS1 Databar Expanded Stackedの印字	4-27
	printAztecCode	4-27
	cutPaper	4-28
	feedPosition	4-28
	openDrawer	4-28
	buzzer	4-28
	externalBuzzer	4-28
	sendBinary	4-29
	sendDataFile	4-29
	getStatus	4-30
	setCallbackFunctionListener	4-32
	プリンタステータス変化のコールバック	
	開始・終了	4-32
	abort	4-32
	registerLogo	4-33
	printLogo	4-34

unregisterLogo	ロゴの登録削除	4-34
registerStyleSheet	スタイルシートの登録	4-34
unregisterStyleSheet	スタイルシートの登録削除	4-34
resetPrinter	プリンターのリセット	4-35
getPrinterResponse	プリンターからの各種応答取得	4-35
startDiscoveryPrinter	プリンターの探索開始 (Bluetooth)	4-36
startDiscoveryPrinter	プリンターの探索開始 (USB)	4-37
startDiscoveryPrinter	プリンターの探索開始 (TCP/IP)	4-37
cancelDiscoveryPrinter	プリンターの探索中断	4-38
getFoundPrinter	発見されたプリンター情報リストの取得	4-38
getSendTimeout	送信タイムアウト時間の取得	4-38
setSendTimeout	送信タイムアウト時間の設定	4-38
getReceiveTimeout	受信タイムアウト時間の取得	4-38
setReceiveTimeout	受信タイムアウト時間の設定	4-39
getInternationalCharacter	国際文字セットの取得	4-39
setInternationalCharacter	国際文字セットの設定	4-39
getCodePage	コードページの取得	4-40
setCodePage	コードページの設定	4-40
getPrinterModel	プリンターモデルの取得	4-40
getPortType	接続中ポート種別の取得	4-40
isConnect	プリンターとの通信状態の確認	4-41
getSocketKeepingTime	ソケット維持時間の取得	4-41
setSocketKeepingTime	ソケット維持時間の設定	4-41
getVersion	SDKバージョンの取得	4-41
controlTransaction	一括処理の開始・終了	4-41
4.1.2 PrinterEventクラス		4-43
(1) メソッド一覧		4-43
(2) 終了イベント定数		4-43
(3) メソッド詳細		4-43
getEventType	終了イベントの取得	4-43
4.1.3 PrinterListenerインターフェイス		4-44
(1) メソッド一覧		4-44
(2) メソッド詳細		4-44
finishEvent	プリンター探索の終了イベント	4-44
4.1.4 PrinterInfoクラス		4-45
(1) メソッド一覧		4-45
(2) メソッド詳細		4-45
getPrinterModelName	プリンターモデル名の取得	4-45

getBluetoothAddress	Bluetoothアドレスの取得	4-45
getMacAddress	MACアドレスの取得	4-45
getIsBonded	ペアリング状態の取得	4-46
getDevicePath	デバイスパスの取得	4-46
4.1.5	PrinterExceptionクラス	4-47
(1)	メソッド一覧	4-47
(2)	定数一覧	4-47
①	エラーコード	4-47
(3)	メソッド詳細	4-48
PrinterException	コンストラクタ	4-48
getErrorCode	エラーコードの取得	4-48
4.1.6	CallbackFunctionListenerインターフェイス	4-49
(1)	メソッド一覧	4-49
(2)	メソッド詳細	4-49
onStatusChanged	プリンタステータスの変化イベント	4-49
4.1.7	BarcodeScannerListenerインターフェイス	4-50

---

## 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	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

---

## 6章      免責 6-1

---



---

## 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

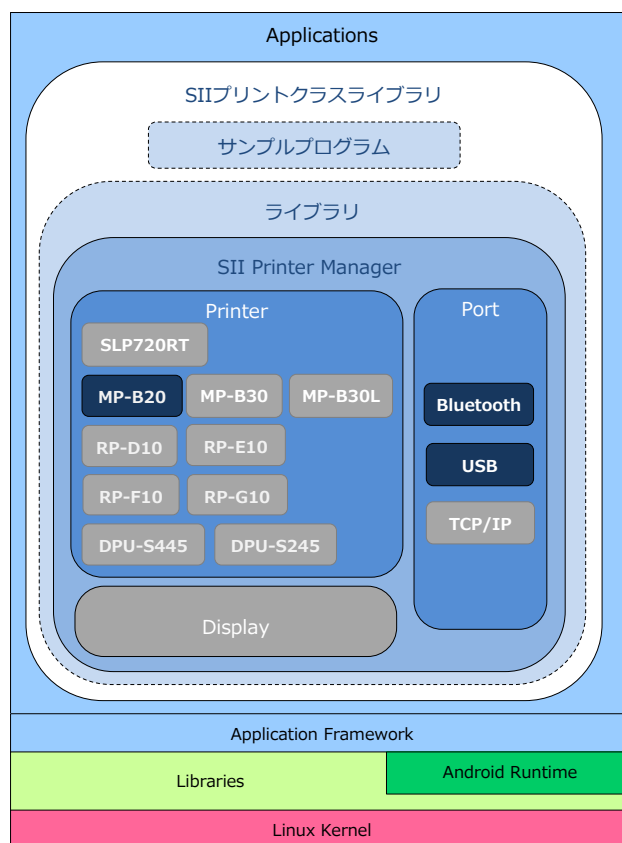
ライブラリとサンプルプログラムを含むSIIプリントクラスライブラリは、AndroidアプリケーションにSII製プリンターMP-B20シリーズ（以下、プリンター）を利用するための機能を提供します。

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またはUSB)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)
- 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によるプリンターの探索

\*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いません。プリンターから応答を読み取るには、`getStatus`、または `getPrinterResponse` を使用してください。

(注意) MP-B20は、ページモード、ディスプレイ及びバーコードスキャナーのAPIをサポートしていません。

---

## 2章 製品仕様

---

本章では、ライブラリの製品仕様について説明します。

### 2.1 動作環境

ライブラリの動作環境を下記に示します。

プリンター	モデル		MP-B20	
	F/Wバージョン		1.00以降	
	通信インターフェース		Bluetooth	USB
Android デバイス	通信ポート		Bluetooth* <sup>1</sup>	USB* <sup>2</sup>
	OS	Android 5.0 (API 21)	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)		
		Android 11.0 (API 30)		
		Android 12.0 (API 31)		
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USBホスト機能をサポートしている必要があります。

## 2.2 動作条件

ライブラリの動作条件を下記に示します。

ライブラリの利用時には、予めプリンターの機能設定を以下の[値]に設定してください。

機能設定の詳細、および工場出荷時の設定については、「MP-B20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。

MS	機能	値
1-1	通信選択 (Interface)	0 : USB 1 : Wireless
3-1	自動ステータス応答機能選択 (Auto Status Back)	0 : 有効
3-2	イニシャライズ完了応答選択 (Init. Response)	0 : 有効
3-3	リアルタイムコマンド選択 (Realtime Command)	0 : 有効
3-4	エラー時データ破棄選択 (Error Through)	0 : 有効
3-5	出力バッファフル時データ破棄選択 (Response Data Discarding)	1 : 無効

---

## 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。

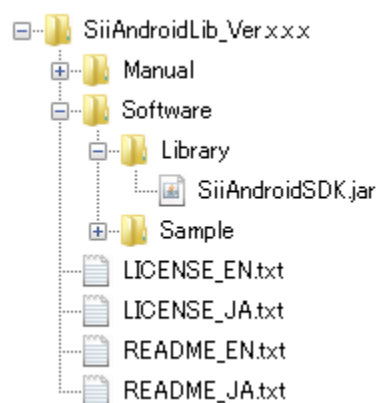


図 3-1

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(図 3-2)、フォルダ名に"libs"を入力してフォルダを作成してください。(図 3-3)

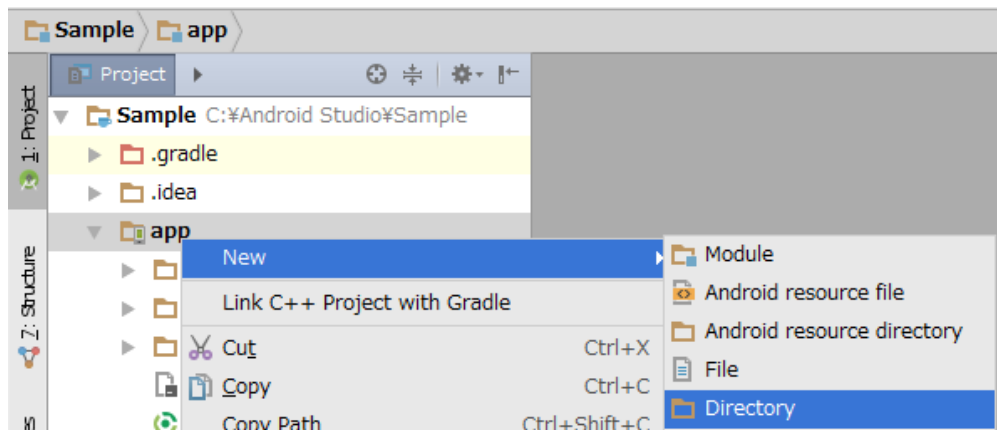


図 3-2

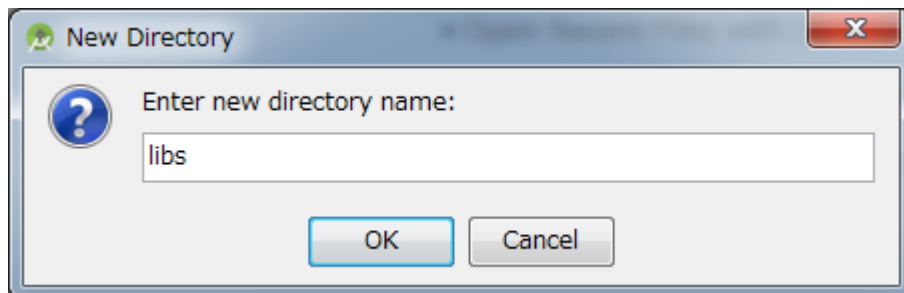


図 3-3

- (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。

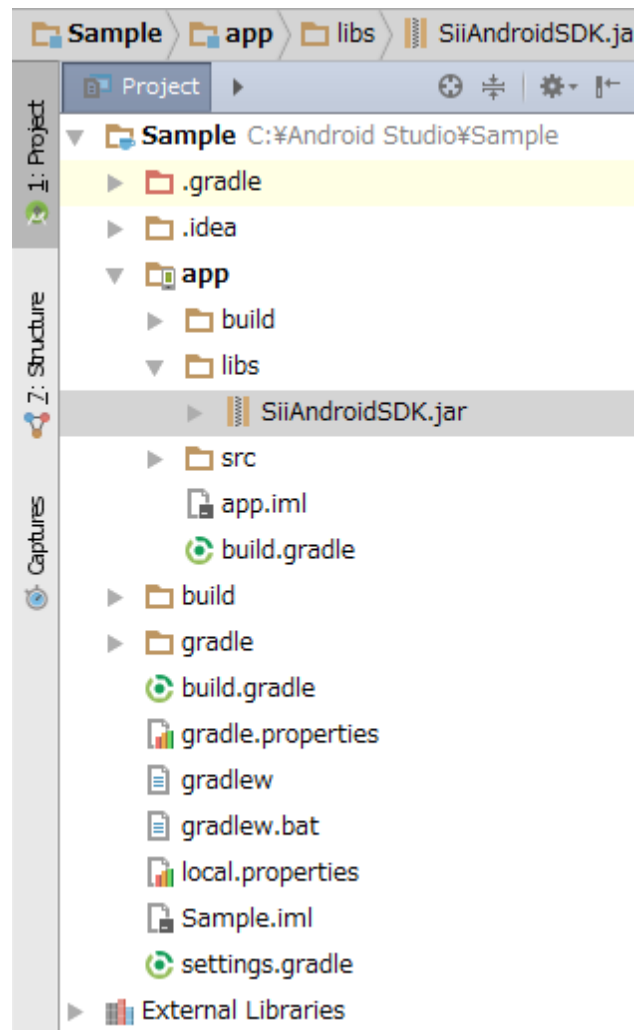


図 3-4

(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)

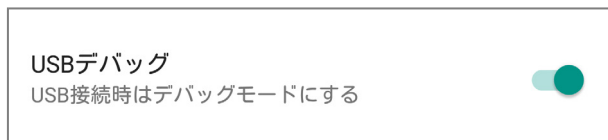


図 3-6

### 3.5 注意事項

- 対象範囲別ストレージについて

Android 10から導入された“対象範囲別ストレージ”ではアプリ固有のストレージとアプリ外部のストレージが区別されます。

Android 10 (API 29)以降をアプリのターゲットとする場合、アプリ外部のストレージ内にあるメディアファイルに該当しないファイルを直接扱うことができません。メディアファイルに該当しないファイルは“ストレージアクセス フレームワーク”を利用すると扱うことができます。

対象範囲別ストレージの詳細については、以下を参照してください。

- データ ストレージとファイル ストレージの概要

<https://developer.android.com/training/data-storage>



## 4章 ライブラリの機能

本章では、ライブラリに実装されている各クラスのAPIについて説明します。

### 4.1 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 <sup>*1</sup>
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。 「4.1.1 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。 「4.1.2 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 「4.1.3 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。 「4.1.4 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 「4.1.5 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。 「4.1.6 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。 「4.1.7 BarcodeScannerListenerインターフェイス」を参照してください。	-

\*1: ✓: サポート、-: MP-B20は非サポート

(注意) MP-B20は、ページモード、ディスプレイ及びバーコードスキャナーのAPIをサポートしていません。

#### 4.1.1 PrinterManagerクラス

##### (1) メソッド一覧

PrinterManagerクラスで提供されるメソッドを下記に示します。

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

名前	説明	サポート*1
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	接続中ポート種別の取得	✓
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	-
getVersion	SDKバージョンの取得	✓
controlTransaction	一括処理の開始・終了	✓

\*1: ✓: サポート、-: MP-B20は非サポート

\*2: 用紙カット位置までの紙送り動作のみ行います。

## (2) 定数一覧

### ① プリンターモデル

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

定数名	説明	値
PRINTER_MODEL_MP_B20	MP-B20	298
PRINTER_MODEL_DEFAULT	プリンターモデルの初期値	284

### ② 応答種別

プリンターからの各種応答取得で利用する定数を下記に示します。

定数名	説明	値
PRINTER_RESPONSE_REQUEST	実行応答リクエスト	0
PRINTER_RESPONSE_USER_AREA	ユーザ領域の残り容量の送信	1
PRINTER_RESPONSE_ARRANGE_USER_AREA	ユーザ領域の整理後の残り容量の送信	2
PRINTER_RESPONSE_NV_GRAPHICS	NVグラフィックスメモリ容量の送信	3

定数名	説明	値
PRINTER_RESPONSE_KEY_CODE	定義済みNVグラフィックスの キーコード一覧の送信	4
PRINTER_RESPONSE_BATTERY_STATUS	バッテリー残量レベル	5

### ③ バッテリー残量レベル

プリンターから取得するバッテリー残量レベルの定数を下記に示します。

定数名	説明	値
BATTERY_STATUS_FULL	バッテリー残量目安：約80%	0
BATTERY_STATUS_MIDDLE	バッテリー残量目安：約40%	1
BATTERY_STATUS_LOW	バッテリー残量目安：約10%	2
BATTERY_STATUS_EMPTY	バッテリーなし	3

### ④ 国際文字セット

国際文字セットの設定/取得で利用する定数を下記に示します。

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

## ⑤ 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
<b>CODE_PAGE_437</b>	USA, Standard Europe (Code Page 437)	0
<b>CODE_PAGE_KATAKANA</b>	Katakana	1
<b>CODE_PAGE_850</b>	Multilingual (Code Page 850)	2
<b>CODE_PAGE_860</b>	Portuguese (Code Page 860)	3
<b>CODE_PAGE_863</b>	Canadian-French (Code Page 863)	4
<b>CODE_PAGE_865</b>	Nordic (Code Page 865)	5
<b>CODE_PAGE_857</b>	Turkish (Code Page 857)	13
<b>CODE_PAGE_737</b>	Greek (Code Page 737)	14
<b>CODE_PAGE_1252</b>	Latin (Code Page 1252)	16
<b>CODE_PAGE_866</b>	Russian (Code Page 866)	17
<b>CODE_PAGE_852</b>	Eastern Europe (Code Page 852)	18
<b>CODE_PAGE_858</b>	Euro (Code Page 858)	19
<b>CODE_PAGE_855</b>	Cyrillic (Code Page 855)	34
<b>CODE_PAGE_864<sup>*1*2</sup></b>	Arabic (Code Page 864)	37
<b>CODE_PAGE_1250</b>	Central European (Code Page 1250)	45
<b>CODE_PAGE_1251</b>	Cyrillic (Code Page 1251)	46
<b>CODE_PAGE_1253</b>	Greek (Code Page 1253)	47
<b>CODE_PAGE_1254</b>	Turkish (Code Page 1254)	48

\*1: Unicodeの20AChは印字できません。

\*2: フォントBは印字できません。

## ⑥ 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
<b>PRINTER_TYPE_BLUETOOTH</b>	Bluetooth	0
<b>PRINTER_TYPE_USB</b>	USB	1

⑦ バーコード、PDF417専用

バーコードの印字と、PDF417の印字で利用する定数を下記に示します。

定数名	説明	値
<b>BARCODE_HEIGHT_DEFAULT</b>	バーコード高さのデフォルト値	162
<b>PDF417_MODULE_HEIGHT_DEFAULT</b>	PDF417高さデフォルト値	10
<b>PDF417_ROW_AUTO</b>	行数自動選択	0
<b>PDF417_COLUMN_AUTO</b>	カラム数自動選択	0

(3) 列挙型定数一覧

① 強調印字(CharacterBold)

強調印字で利用する列挙型定数を下記に示します。

定数名	説明
<b>BOLD_CANCEL</b>	強調印字を解除
<b>BOLD</b>	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
<b>UNDERLINE_CANCEL</b>	アンダーライン印字を解除
<b>UNDERLINE_1</b>	1ドット幅アンダーライン印字を指定
<b>UNDERLINE_2</b>	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
<b>REVERSE_CANCEL</b>	白黒反転印字を解除
<b>REVERSE</b>	白黒反転印字を指定

④ 倒立印字(CharacterInversion)

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

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

⑤ 文字フォント(CharacterFont)

文字フォントで利用する列挙型定数を下記に示します。

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

⑥ 文字倍率(CharacterScale)

文字倍率で利用する列挙型定数を下記に示します。

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

定数名	説明
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

⑦ 位置揃え (PrintAlignment)

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

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

⑧ 保留データの出力指定 (OutputPendingData)

保留データの出力指定で利用する列挙型定数を下記に示します。

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

⑨ バーコードシンボル (BarcodeSymbol)

バーコードシンボルで利用する列挙型定数を下記に示します。

定数名	説明	構文 <sup>*1</sup>
BARCODE_SYMBOL_UPC_A	UPC-A	(a)
BARCODE_SYMBOL_UPC_E	UPC-E	(a)
BARCODE_SYMBOL_EAN13	EAN13	(a)
BARCODE_SYMBOL_JAN13	JAN13	(a)



定数名	説明	構文 <sup>*1</sup>
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 add-on	(a)
BARCODE_SYMBOL_JAN13_ADDON	JAN13 add-on	(a)
BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	(a)
BARCODE_SYMBOL_GS1_TRUNCATED	GS1 Databar Truncated	(a)
BARCODE_SYMBOL_GS1_LIMITED	GS1 Databar Limited	(a)
BARCODE_SYMBOL_GS1_EXPANDED	GS1 Databar Expanded	(a)

\*1: 構文の内容は、「4.1.1(4)メソッド詳細 printBarcode」を参照してください。

#### ⑩ モジュールサイズ (ModuleSize)

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

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

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

定数名	説明	使用メソッド
GSIDATABAR_MODULE_SIZE_2	2ドット	<ul style="list-style-type: none"> <li>● printGS1DataBarStacked</li> <li>● printGS1DataBarStackedOmnidirectional</li> <li>● printGS1DataBarExpandedStacked</li> </ul>
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	

#### ⑪ HRI文字印字位置 (HriPosition)

HRI文字印字位置で利用する列挙型定数を下記に示します。

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

#### ⑫ NW比 (NwRatio)

NW比で利用する列挙型定数を下記に示します。

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

⑬ エラー訂正レベル(ErrorCorrection)

エラー訂正レベルで利用する列挙型定数を下記に示します。

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

⑭ PDF417シンボル(Pdf417Symbol)

PDF417シンボルで利用する列挙型定数を下記に示します。

定数名	説明
PDF417_STANDARD	PDF417
PDF417_COMPACT	コンパクトPDF417

⑮ QRコードモデル(QrModel)

QRコードモデルで利用する列挙型定数を下記に示します。

定数名	説明
QR_MODEL_1	QRコードモデル1
QR_MODEL_2	QRコードモデル2

⑩ Data Matrixモジュール(DataMatrixModule)

Data Matrixモジュールで利用する列挙型定数を下記に示します。

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

⑰ MaxiCodeモード (MaxiCodeMode)

MaxiCodeモードで利用する列挙型定数を下記に示します。

定数名	説明
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

⑱ カット方法 (CuttingMethod)

カット方法で利用する列挙型定数を下記に示します。どちらの定数を指定してもカットは行わず、用紙カット位置までの紙送りを行います。

定数名	説明
CUT_FULL	フルカット
CUT_PARTIAL	パーシャルカット

⑲ 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

⑳ 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

**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 (a) `public PrinterManager(Context context)`

(b) `public PrinterManager()`

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`

注意 新規に本メソッドを使用する場合は、構文(a)をご使用ください。  
構文(b)は将来的に非サポートとなるメソッドです。

**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、「4.1.1(2)① プリンターモデル」を参照してください。

*address* Bluetoothアドレス  
例: `"00:11:22:AA:BB:CC"`

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

例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。  
エラーの内容は、「4.1.5 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のモニタリングを開始します。最新のプリンターステータスは  
`getStatus`で取得が可能です。

プリンターステータスの変化は、`onStatusChanged`、及び`setCallbackFunctionListener`により、イベントとして通知できます。

USB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、プリンターモデルを指定します。

構文(b)のメソッドは、プリンターモデルを指定し、アプリケーションを呼び出すコンテキストを指定します。

構文 (a) `public void connect(int printerModel) throws PrinterException`  
 (b) `public void connect(int printerModel, Context context) 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、「4.1.1(2)① プリンターモデル」を参照してください。  
*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`

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、「4.1.5 **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のモニタリングを開始します。最新のプリンターステータスは `getStatus`で取得が可能です。

プリンターステータスの変化は、`onStatusChanged`、及び`setCallbackFunctionListener`により、イベントとして通知できます。

注意 新規に本メソッドを使用する場合は、構文(a)をご使用ください。  
 構文(b)は将来的に非サポートとなるメソッドです。

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、「4.1.5 **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。

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開始・終了

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

**構文**

```
public void setBarcodeScannerListener(BarcodeScannerListener listener)
```

**throws PrinterException**

**sendText**

## テキストデータの送信

テキストデータを送信します。

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

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

## 例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。  
エラーについての詳細は、「**4.1.5 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)及び(c)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
```

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

パラメータ <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、「4.1.1(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、「4.1.1(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、「4.1.1(3)③ 白黒反転印字(CharacterReverse)」を参照してください。
<i>inversion</i>	倒立印字 利用可能な定数は、「4.1.1(3)④ 倒立印字(CharacterInversion)」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、「4.1.1(3)⑤ 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、「4.1.1(3)⑥ 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、「4.1.1(3)⑦ 位置揃え(PrintAlignment)」を参照してください。

*output*

保留データ出力指定

利用可能な定数は、「4.1.1(3)⑧ 保留データの出力指定 (OutputPendingData)」を参照してください。

例外

**PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。

エラーについての詳細は、「4.1.5 **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  
 利用可能な定数は、「4.1.1(3)⑨ バーコードシンボル (BarcodeSymbol)」を参照してください。

*text(data)*                      プリンターに送信するバーコードデータ

*moduleSize*                        バーコードの幅  
 利用可能な定数は、「4.1.1(3)⑩ モジュールサイズ (ModuleSize)」を参照してください。

*moduleHeight*                      バーコードの高さ(ドット)

- *barcodeSymbol*が下記の場合の有効範囲は、1～255です。  
**BARCODE\_SYMBOL\_UPC\_A**  
**BARCODE\_SYMBOL\_UPC\_E**  
**BARCODE\_SYMBOL\_EAN13**  
**BARCODE\_SYMBOL\_JAN13**  
**BARCODE\_SYMBOL\_EAN8**  
**BARCODE\_SYMBOL\_JAN8**  
**BARCODE\_SYMBOL\_CODE39**  
**BARCODE\_SYMBOL\_CODE93**  
**BARCODE\_SYMBOL\_CODE128**  
**BARCODE\_SYMBOL\_ITF**  
**BARCODE\_SYMBOL\_CODABAR**  
**BARCODE\_SYMBOL\_EAN13\_ADDON**  
**BARCODE\_SYMBOL\_JAN13\_ADDON**
- *barcodeSymbol*が下記の場合は、*barcodeSymbol*と*moduleSize*により有効範囲が異なります。

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	有効範囲
<b>BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL</b>		
	<b>BARCODE_MODULE_WIDTH_2</b>	66 ～ 255
	<b>BARCODE_MODULE_WIDTH_3</b>	99 ～ 255
	<b>BARCODE_MODULE_WIDTH_4</b>	132 ～ 255
	<b>BARCODE_MODULE_WIDTH_5</b>	165 ～ 255
	<b>BARCODE_MODULE_WIDTH_6</b>	198 ～ 255

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	有効範囲
<b>BARCODE_SYMBOL_GS1_TRUNCATED</b>		
	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
<b>BARCODE_SYMBOL_GS1_LIMITED</b>		
	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
<b>BARCODE_SYMBOL_GS1_EXPANDED</b>		
	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、「4.1.1(3)⑪ 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、「4.1.1(3)⑤ 文字フォント (CharacterFont)」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、「4.1.1(3)⑦ 位置揃え (PrintAlignment)」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、「4.1.1(3)⑫ 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、「4.1.5 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。

## printPDF417

## PDF417の印字

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 *text*

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、「4.1.1(3)⑬ 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、0～90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、「4.1.1(3)⑩ 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、「4.1.1(3)⑦ 位置揃え(PrintAlignment)」を参照してください。
<i>pdf417Symbol</i>	PDF417のシンボル 利用可能な定数は、「4.1.1(3)⑭ PDF417シンボル(Pdf417Symbol)」を参照してください。

#### 例外

#### **PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。

エラーについての詳細は、「4.1.5 **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。

## 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コードモデルを指定します。

構文(b)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

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

```
(b) public void printQRcode(String text,
                             ErrorCorrection errorCorrection,
                             ModuleSize moduleSize,
                             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 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、「4.1.1(3)⑬ 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、「4.1.1(3)⑩ モジュールサイズ(ModuleSize)」を参照してください。
	<i>alignment</i>	位置揃え 利用可能な定数は、「4.1.1(3)⑦ 位置揃え(PrintAlignment)」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、「4.1.1(3)⑮ QRコードモデル(QrModel)」を参照してください。
例外	<b>PrinterException</b>	本メソッド呼び出し時にエラーが発生した場合は、 <b>PrinterException</b> をスローします。 エラーについての詳細は、「4.1.5 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、 <b>PrinterException</b> 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、 <b>isConnect</b> を参照してください。

## printDataMatrix

## Data Matrixの印字

Data Matrixを印字します。

構文	<pre>public void printDataMatrix(String text,                              DataMatrixModule dataMatrixModule,                              ModuleSize moduleSize,                              PrintAlignment alignment) throws PrinterException</pre>	
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>dataMatrixModule</i>	Data Matrixモジュール数 利用可能な定数は、「4.1.1(3)⑯ 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、「4.1.1(3)⑩ モジュールサイズ(ModuleSize)」を参照してください。
	<i>alignment</i>	位置揃え 利用可能な定数は、「4.1.1(3)⑦ 位置揃え(PrintAlignment)」を参照してください。



## 例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。

エラーについての詳細は、「[4.1.5 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。

**printMaxiCode**

## MaxiCodeの印字

MaxiCodeを印字します。

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

パラメータ *text* プリンターに送信するバーコードデータ

- *maxiCodeMode*が**MAXI\_CODE\_2**の場合  
データの先頭にサービスクラス(3桁)、国コード(3桁)、郵便番号(9桁)を付加してください。
- *maxiCodeMode*が**MAXI\_CODE\_3**の場合  
データの先頭にサービスクラス(3桁)、国コード(3桁)、郵便番号(6桁)を付加してください。

*maxiCodeMode*      MaxiCodeモード  
利用可能な定数は、「4.1.1(3)⑰ MaxiCodeモード(MaxiCodeMode)」を参照してください。

*alignment*      位置揃え  
利用可能な定数は、「4.1.1(3)⑦ 位置揃え (PrintAlignment)」を参照してください。

## 例外 **PrinterException**

本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。

エラーについての詳細は、「4.1.5 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。

```
printGS1DataBarStacked
```

## GS1 Databar Stackedの印字

GS1 Databar Stackedを印字します。

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

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

*moduleSize* モジュールサイズ  
利用可能な定数は、「4.1.1(3)⑩ モジュールサイズ (ModuleSize)」を参照してください。

*alignment* 位置揃え  
利用可能な定数は、「4.1.1(3)⑦ 位置揃え (PrintAlignment)」を参照してください。

**例外** **PrinterException**  
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。  
エラーについての詳細は、「4.1.5 **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*      バーコードの段高さ(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、「4.1.1(3)⑩ モジュールサイズ (ModuleSize)」を参照してください。

*alignment*      位置揃え  
利用可能な定数は、「4.1.1(3)⑦ 位置揃え (PrintAlignment)」を参照してください。

**例外** **PrinterException**  
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。  
エラーについての詳細は、「4.1.5 **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。チェックデジットはプリンターにより自動計算されませんので必ず入力してください。

*column*      カラム数  
1行のセグメント数を指定してください。  
有効範囲は2～20の偶数です。

*moduleSize*      モジュールサイズ  
利用可能な定数は、「4.1.1(3)⑩ モジュールサイズ (ModuleSize)」を参照してください。

*alignment*      位置揃え  
利用可能な定数は、「4.1.1(3)⑦ 位置揃え (PrintAlignment)」を参照してください。

例外      **PrinterException**  
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。  
エラーについての詳細は、「4.1.5 **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。

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

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

cutPaper 用紙のカット

## 用紙のカット

用紙カット位置までの紙送りを行います。用紙のカットは行いません。

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、「4.1.1(3)⑩ 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、「**4.1.5 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。

feedPosition	用紙の頭出し
0	用紙の左端
1	用紙の右端
2	用紙の中央
3	用紙の左端から1/4
4	用紙の右端から1/4
5	用紙の中央から1/4
6	用紙の左端から1/2
7	用紙の右端から1/2
8	用紙の中央から1/2
9	用紙の左端から3/4
10	用紙の右端から3/4
11	用紙の中央から3/4
12	用紙の左端から4/5
13	用紙の右端から4/5
14	用紙の中央から4/5
15	用紙の左端から5/6
16	用紙の右端から5/6
17	用紙の中央から5/6
18	用紙の左端から6/7
19	用紙の右端から6/7
20	用紙の中央から6/7
21	用紙の左端から7/8
22	用紙の右端から7/8
23	用紙の中央から7/8
24	用紙の左端から8/9
25	用紙の右端から8/9
26	用紙の中央から8/9
27	用紙の左端から9/10
28	用紙の右端から9/10
29	用紙の中央から9/10
30	用紙の左端から10/11
31	用紙の右端から10/11
32	用紙の中央から10/11
33	用紙の左端から11/12
34	用紙の右端から11/12
35	用紙の中央から11/12
36	用紙の左端から12/13
37	用紙の右端から12/13
38	用紙の中央から12/13
39	用紙の左端から13/14
40	用紙の右端から13/14
41	用紙の中央から13/14
42	用紙の左端から14/15
43	用紙の右端から14/15
44	用紙の中央から14/15
45	用紙の左端から15/16
46	用紙の右端から15/16
47	用紙の中央から15/16
48	用紙の左端から16/17
49	用紙の右端から16/17
50	用紙の中央から16/17
51	用紙の左端から17/18
52	用紙の右端から17/18
53	用紙の中央から17/18
54	用紙の左端から18/19
55	用紙の右端から18/19
56	用紙の中央から18/19
57	用紙の左端から19/20
58	用紙の右端から19/20
59	用紙の中央から19/20
60	用紙の左端から20/21
61	用紙の右端から20/21
62	用紙の中央から20/21
63	用紙の左端から21/22
64	用紙の右端から21/22
65	用紙の中央から21/22
66	用紙の左端から22/23
67	用紙の右端から22/23
68	用紙の中央から22/23
69	用紙の左端から23/24
70	用紙の右端から23/24
71	用紙の中央から23/24
72	用紙の左端から24/25
73	用紙の右端から24/25
74	用紙の中央から24/25
75	用紙の左端から25/26
76	用紙の右端から25/26
77	用紙の中央から25/26
78	用紙の左端から26/27
79	用紙の右端から26/27
80	用紙の中央から26/27
81	用紙の左端から27/28
82	用紙の右端から27/28
83	用紙の中央から27/28
84	用紙の左端から28/29
85	用紙の右端から28/29
86	用紙の中央から28/29
87	用紙の左端から29/30
88	用紙の右端から29/30
89	用紙の中央から29/30
90	用紙の左端から30/31
91	用紙の右端から30/31
92	用紙の中央から30/31
93	用紙の左端から31/32
94	用紙の右端から31/32
95	用紙の中央から31/32
96	用紙の左端から32/33
97	用紙の右端から32/33
98	用紙の中央から32/33
99	用紙の左端から33/34
100	用紙の右端から33/34
101	用紙の中央から33/34
102	用紙の左端から34/35
103	用紙の右端から34/35
104	用紙の中央から34/35
105	用紙の左端から35/36
106	用紙の右端から35/36
107	用紙の中央から35/36
108	用紙の左端から36/37
109	用紙の右端から36/37
110	用紙の中央から36/37
111	用紙の左端から37/38
112	用紙の右端から37/38
113	用紙の中央から37/38
114	用紙の左端から38/39
115	用紙の右端から38/39
116	用紙の中央から38/39
117	用紙の左端から39/40
118	用紙の右端から39/40
119	用紙の中央から39/40
120	用紙の左端から40/41
121	用紙の右端から40/41
122	用紙の中央から40/41
123	用紙の左端から41/42
124	用紙の右端から41/42
125	用紙の中央から41/42
126	用紙の左端から42/43
127	用紙の右端から42/43
128	用紙の中央から42/43
129	用紙の左端

## 用紙の頭出し

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 feedPosition(FeedPosition feedPosition)` throws **PrinterException**

openDrawer キャッシュドロウを開く

## キャッシュドローを開く

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 openDrawer(DrawerNum drawerNum, PulseWidth onOffTime)` throws **PrinterException**

**buzzer** ブザーの鳴動

## ブザーの鳴動

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

### 構文

```
public void buzzer(int onTime, int offTime) throws PrinterException
```

**externalBuzzer** 外部ブザーの鳴動

## 外部ブザーの鳴動

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 externalBuzzer(BuzzerPattern buzzer pattern, int buzzerCount)  
throws PrinterException
```

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、「4.1.5 **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することができます。ただし、本メソッドで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いません。

ファイルデータを送信します。

構文 (a) のメソッドは、ディザリングを指定できます。

構文 (b) 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      (a) `public void sendDataFile(String fileName,  
  PrintAlignment alignment,  
  Dithering dithering) throws PrinterException`  
  
            (b) `public void sendDataFile(String fileName,  
  PrintAlignment alignment) 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名  
入力可能な形式について下記に説明します。

- Javaの標準的なクラス「`java.io.File`」が扱う絶対パス文字列  
Android 10 (API 29)以降をアプリのターゲットとする場合には、一部のファイルが直接扱えなくなりますのでご注意ください。  
詳細については、「3.5 注意事項 - 対象範囲別ストレージについて」を参照してください。
- Androidに用意されたクラス「`android.net.Uri`」が扱う下記スキーム名のURI文字列
  - `file://`
  - `content://`
 本パラメータには“ストレージ アクセス フレームワーク”から取得したURI文字列を指定する必要があります。“ストレージ アクセス フレームワーク”から取得せずに作成されたURIはファイルを開けない可能性がありますのでご注意ください。

指定可能なファイルサイズは、最大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  
イメージデータ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。イメージデータをプリンターのメモリ上に展開してから一括で印字を行ないます。
- .txt  
テキストデータとしてプリンターに送信します。テキストデータのフォーマットはUTF-8をサポートしています。setInternationalCharacter、及びsetCodePageの設定を基にプリンターで印字可能なテキストデータにエンコードしてプリンターに送信します。  
本メソッドではテキストデータの最後に改行コードは付加しません。最後まで印字する場合は、テキストデータの最後に改行コードを入れてください。
- .bin、.dat  
バイナリデータとして変換せずにプリンターに送信します。

*alignment*

位置揃え

*fileName*で指定されたファイルの拡張子が.bmp、.jpg、.jpeg、.png、.txtの場合に有効です。

利用可能な定数は、「4.1.1(3)⑦ 位置揃え(PrintAlignment)」を参照してください。

*dithering*

ディザリング

*fileName*で指定されたファイルの拡張子が.bmp、.jpg、.jpeg、.pngの場合に有効です。

利用可能な定数は、「4.1.1(3)⑱ 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、「4.1.5 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。

## getStatus

## プリンターステータスの取得

最新のプリンターステータスを取得します。

構文(a)のメソッドは、プリンターステータスを戻り値で返します。

構文(b)のメソッドは、プリンターステータスを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、「**4.1.5 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

プリンターから取得したステータスは、int型の配列に格納されます。

プリンターステータスを下記に示します。

通信失敗時は、0x80000000を示します。

ビット	機能	値	
		0	1
0	電圧エラー	エラーなし	エラーあり
1	ハードウェアエラー	エラーなし	エラーあり
2	ヘッド温度エラー	エラーなし	エラーあり
3	予約済み	固定	-
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	予約済み	固定	-
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	予約済み	-	固定
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	-	固定
18	予約済み	-	固定
19	予約済み	-	固定
20～22	バッテリー残量レベル	000 : バッテリなし 001 : Low (残量目安: 約10%) 011 : Middle (残量目安: 約40%) 111 : Full (残量目安: 約80%)	
23	バッテリーエラー	エラーなし	エラーあり
24～31	予約済み	-	固定

**構文**      public void **setCallbackFunctionListener**(CallbackFunctionListener *listener*)  
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、「**4.1.5 PrinterExceptionクラス**」を参照してください。

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、jsConnectがtrueの時に使用してください。

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

**説明** **sendDataFile**によるイメージデータの送信が中断した場合、プリンターは指定したイメージデータの受信完了まで、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、イメージデータ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、未印字のイメージデータの一部分が印字されることがあります。



イメージデータ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、ディザリングを指定できます。

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

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

パラメータ *fileName*

ロゴとして登録するイメージデータのファイル名  
入力可能な形式について下記に説明します。

- Javaの標準的なクラス「java.io.File」が扱う絶対パス文字列  
Android 10 (API 29)以降をアプリのターゲットとする場合には、一部の  
ファイルが直接扱えなくなりますのでご注意ください。  
詳細については、「3.5 注意事項 - 対象範囲別ストレージについて」  
を参照してください。
- Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名の  
URI文字列
  - file://
  - content://
 本パラメータには“ストレージ アクセス フレームワーク”から取得した  
URI文字列を指定する必要があります。“ストレージ アクセス フレーム  
ワーク”から取得せずに作成されたURIはファイルを開けない可能性が  
ありますのでご注意ください。

サポートするイメージデータ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、「4.1.1(3)⑱ 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、「4.1.5 **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。

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、「4.1.1(3)⑦ 位置揃え(PrintAlignment)」を参照してください。

例外      **PrinterException**  
本メソッド呼び出し時にエラーが発生した場合は、**PrinterException**をスローします。  
エラーについての詳細は、「4.1.5 **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。

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、「4.1.5 **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。

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`

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`

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、「4.1.5 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接続の場合：

USBプリンタークラスのSOFT\_RESET機能によりプリンターをリセットします。

プリンターの応答データを取得します。

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

パラメータ    *id*

応答種別定数

利用可能な定数は、「4.1.1(2)② 応答種別」を参照してください。

*buf*

取得した応答データを格納するバッファー

*id*で指定した応答データを*buf*で指定されたオブジェクトに格納します。

応答種別定数によりバッファーの型が異なります。

バッファーの型は、下表を参照してください。

応答種別定数	
パラメータ	説明
<b>PRINTER_RESPONSE_REQUEST</b> (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>buf[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>buf[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。
<b>PRINTER_RESPONSE_USER_AREA</b> (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。
<b>PRINTER_RESPONSE_ARRANGE_USER_AREA</b> (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。

応答種別定数	
パラメータ	説明
<b>PRINTER_RESPONSE_NV_GRAPHICS</b> (NVグラフィックスメモリ容量の送信)	
<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グラフィックスメモリ容量(単位:バイト)が数値で格納されます。
<b>PRINTER_RESPONSE_KEY_CODE</b> (定義済み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>buf[0]</i> = "22"、 <i>buf[1]</i> = "23"、 <i>buf[2]</i> = "24"など
<b>PRINTER_RESPONSE_BATTERY_STATUS</b> (バッテリー残量レベル)	
<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、バッテリー残量レベルが値で格納されます。 値については、「4.1.1(2)③ バッテリー残量レベル」を参照してください。 バッテリー残量レベル <b>BATTERY_STATUS_FULL</b> : Full (残量目安: 約80%) <b>BATTERY_STATUS_MIDDLE</b> : Middle (残量目安: 約40%) <b>BATTERY_STATUS_LOW</b> : Low (残量目安: 約10%) <b>BATTERY_STATUS_EMPTY</b> : バッテリーなし

#### 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、「4.1.5 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(a)のメソッドは、**PrinterListener**のインスタンスを指定します。

構文(b)のメソッドは、**PrinterListener**のインスタンスを指定し、アプリケーションを呼び出すコンテキストを指定します。

#### 構文

(a) `public void startDiscoveryPrinter(PrinterListener listener) throws PrinterException`

(b) `public void startDiscoveryPrinter(PrinterListener listener, Context context) 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ーアプリケーションに通知されます。

*context*

アプリケーションのコンテキスト



## プリンターの探索中断

実行中のstartDiscoveryPrinterを中断します。

構文      `public void cancelDiscoveryPrinter()`

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

## 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

## 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。

## 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	<i>sendTimeout</i>	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。

## 受信タイムアウト時間の取得

データ受信時のタイムアウト時間を取得します。

構文      `public int getReceiveTimeout()`

戻り値 受信タイムアウト時間(ミリ秒:ms)

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

## setReceiveTimeout

## 受信タイムアウト時間の設定

データ受信時のタイムアウト時間を設定します。

構文      `public void setReceiveTimeout(int receiveTimeout)`

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

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

本メソッドはisConnectがtrue/falseに関わらず設定が可能です。  
設定したタイムアウト時間が有効になるのは、次回データ受信時となります。

## getInternationalCharacter

## 国際文字セットの取得

国際文字セットの値を取得します。

構文      `public int getInternationalCharacter()`

戻り値      値については、「4.1.1(2)④ 国際文字セット」を参照してください。

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

## setInternationalCharacter

## 国際文字セットの設定

国際文字セットの値を設定します。

構文      `public void setInternationalCharacter(int internationalCharacter)`

パラメータ *internationalCharacter*      国際文字セット定数  
設定可能な値は、「4.1.1(2)④ 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、「4.1.1(2)⑤ コードページ」を参照してください。

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

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、「4.1.1(2)⑤ 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、「4.1.1(2)① プリンターモデル」を参照してください。  
`isConnect`が`false`の場合は、`PRINTER_MODEL_DEFAULT`が返ります。

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

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、「4.1.1(2)⑥ 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。

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 int getSocketKeepingTime()`

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 setSocketKeepingTime(int socketKeepingTime)`

SDKのバージョンを文字列で取得します。

構文        `public String getVersion()`

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

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

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、「4.1.1(3)② 一括処理選択(TransactionFunction)」を参照してください。

例外	<p><b>PrinterException</b></p> <p>本メソッド呼び出し時にエラーが発生した場合は、<b>PrinterException</b>をスローします。</p> <p>エラーについての詳細は、「<b>4.1.5 PrinterExceptionクラス</b>」を参照してください。</p> <p>データ送信に失敗すると、プリンターとの通信を終了し、<b>PrinterException</b>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、<b>isConnect</b>を参照してください。</p>
説明	<p>一括処理の手順は以下の通りです。</p> <ol style="list-style-type: none"> <li>① 一括処理を開始します。 TRANSACTION_STARTを指定してください。</li> <li>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。</li> <li>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。 <ul style="list-style-type: none"> <li>・TRANSACTION_CLEARの指定</li> <li>・TRANSACTION_STARTの指定</li> <li>・disconnectの実行</li> </ul> </li> </ol> <p>一括処理対象メソッドを下記に示します。</p> <ul style="list-style-type: none"> <li>・sendText</li> <li>・sendTextEx</li> <li>・printBarcode</li> <li>・printPDF417</li> <li>・printQRcode</li> <li>・printDataMatrix</li> <li>・printMaxiCode</li> <li>・printGS1DataBarStacked</li> <li>・printGS1DataBarStackedOmnidirectional</li> <li>・printGS1DataBarExpandedStacked</li> <li>・sendBinary</li> <li>・sendDataFile</li> <li>・printLogo*1</li> </ul>

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

#### 4.1.2 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、「4.1.2(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。

### 4.1.3 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
<b>finishEvent</b>	プリンター探索の終了イベント

#### (2) メソッド詳細

<b>finishEvent</b>	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。

#### 4.1.4 PrinterInfoクラス

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

##### (1) メソッド一覧

プリンターモデル名 (Bluetoothデバイス名)、Bluetoothアドレス、MACアドレス、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
getIsBonded	ペアリング状態の取得
getDevicePath	デバイスパスの取得

##### (2) メソッド詳細

#### getPrinterModelName プリンターモデル名の取得

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

構文      `public String getPrinterModelName()`

戻り値    プリンターモデル名 (Bluetoothデバイス名)

#### getBluetoothAddress Bluetoothアドレスの取得

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

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ペアリング状態を取得します。

構文        `public String 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

#### 4.1.5 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。 <sup>*1</sup>	-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。 <sup>*1</sup>	-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	イメージデータ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	イメージデータの幅と高さのいずれか、または両方が印字可能な最大ドット数を超過しています。	-312
ERROR_LOGO_NOT_DEFINED	ロゴが登録されていません。	-313
ERROR_LOW_USER_AREA	ユーザ領域の残り容量が不足しています。	-401
ERROR_LOW_EXTERNAL_RAM	RAMの残り容量が不足しています。	-402

定数名	説明	値
<b>ERROR_INVALID_PARAM</b>	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、「4.1.5(2) 定数一覧」を参照してください。



#### 4.1.6 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

<b>onStatusChanged</b>	<b>プリンタステータスの変化イベント</b>
------------------------	-------------------------

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。

#### 4.1.7 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。

MP-B20は、本インターフェイスをサポートしていません。

---

## 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。

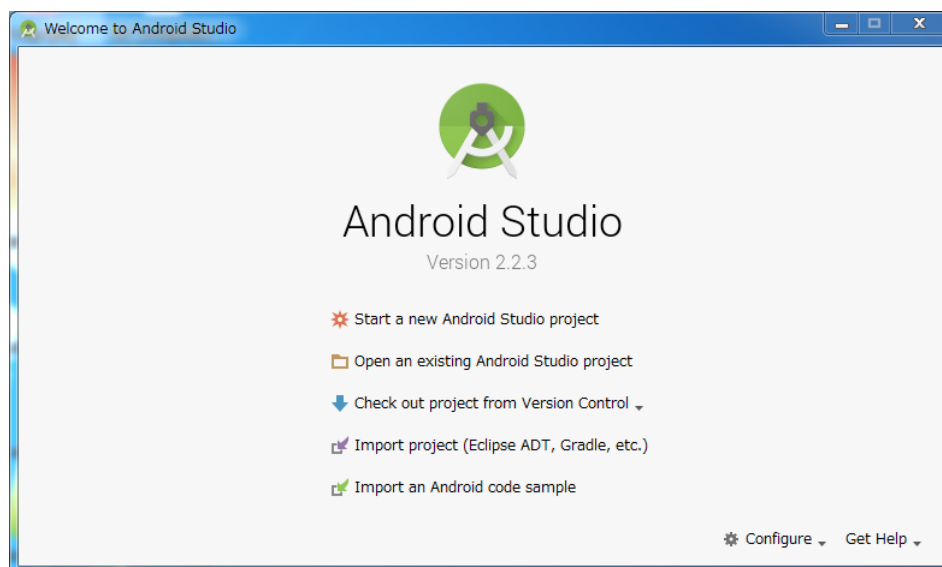
### 5.1 インストール

サンプルプログラムをインストールしてください。

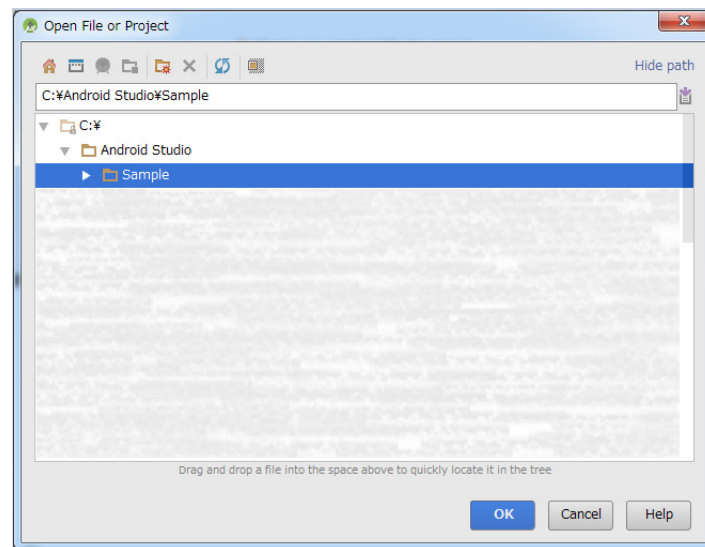
Androidアプリケーションの開発環境が整っていることを確認してください。必要な開発環境については、「3章 ライブラリの利用方法」を参照してください。

インストール手順は以下の通りです。

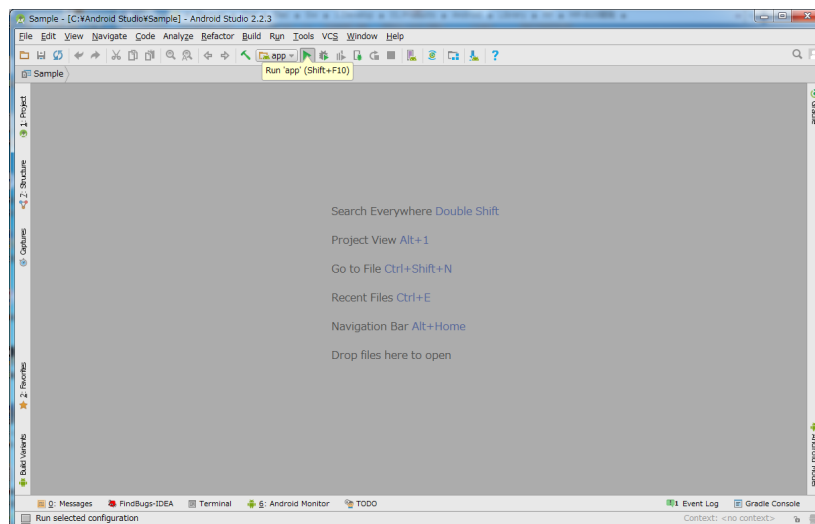
- (1) Sampleフォルダを任意の位置に配置してください。
- (2) Android Studioを起動し、「Open an existing Android Studio Project」をクリックしてください。



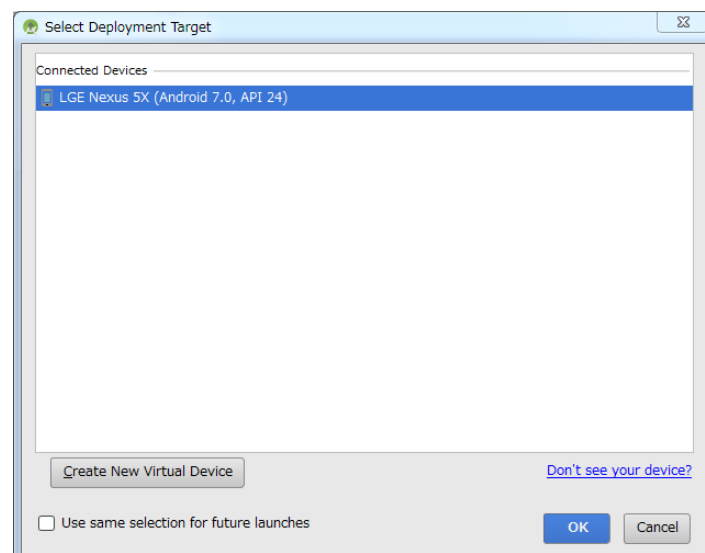
(3) 手順(1)で配置したフォルダを選択し、[OK]をクリックしてください。



(4) [Run 'app']をクリックしてください。



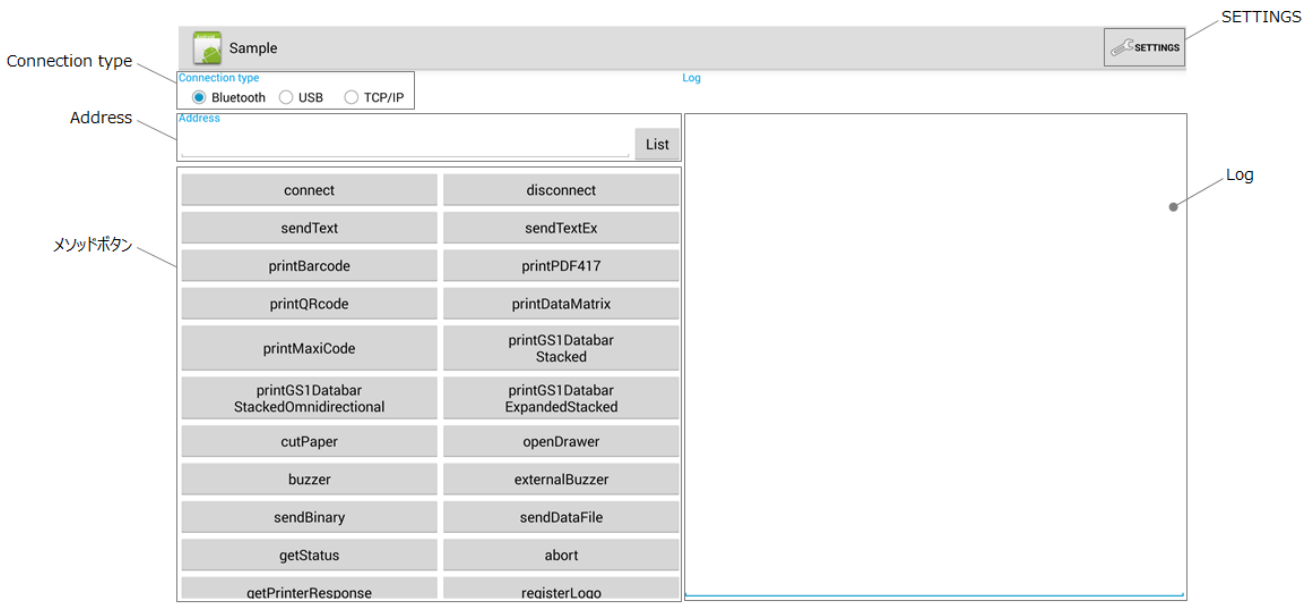
(5) デバイスを選択し、[OK]をクリックしてください。




# 5.2 画面

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。 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。

Sample

Send timeout  
10000msec

Receive timeout  
10000msec

Socket keeping time  
300000msec

International character  
USA

Code page  
Code Page 1252

Secure connection  
Secure connection

SDK version  
X.X.X

☒

### 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。

---

## 6章 免責

---

SIIプリントクラスライブラリは不具合の発生なきよう十分留意して開発しておりますが、万一、SIIプリントクラスライブラリによってお客様に損害を及ぼした場合、一切の責を負いかねますのでご了承ください。

## 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	ı	ı	½	¼	ı	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	⌌	⌌	π	⌌	⌌	F	π	⌌	⌌	J	Γ	■	■	■	■	■
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	∩
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	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	ü	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	ü	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	☐	☐	☐		†	‡		π	¶			¶	¶	¶	¶	¶
C0	L	L	T	†	†	†	†	¶	¶	¶	¶	¶	¶	¶	¶	¶
D0	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	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	Ü	f
A0	¡	´	ó	ú	¶	³	-	î	¶	¬	½	¼	¾	«	»	
B0	☐	☐	☐		†	‡		π	¶			¶	¶	¶	¶	¶
C0	L	L	T	†	†	†	†	¶	¶	¶	¶	¶	¶	¶	¶	¶
D0	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶	¶
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	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	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	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	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	¡	«	»	
B0	☐	☐	☐		†	‡	§	¶	§		¶		¶		¶	
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	ó	β	ô	ò	õ	õ	μ	×	ú	û	ü	ì	ÿ	-	'	
F0	-	±	¾	¶	§	÷	•	•	•	•	•	•	•	•	•	■

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	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
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	€	‚	ƒ	„	…	†	‡	^	‰	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	Ł	ł	Ś	ś	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	
C0	␣	␣	␣	␣	␣	Ä	ä	␣	␣	␣	␣	␣	␣	␣	␣	␣
D0	đ	Đ	Ď	Ě	ď	Ň	Í	Î	ě	␣	␣	␣	␣	Ť	Ů	■
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	Ů	ý	Ý	ť	'	
F0	-	"	˘	˘	˘	§	÷	˘	˘	˘	˘	Ů	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			ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	Ј	Г	■	■	І	İ	■
E0	ó	β	ô	ò	õ	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			к	К	ℓ	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	≈	«	»	لأ	لأ					لا
A0	-	£	£	£	£			ل	ب	ث	ج	ح	خ	ح	ج	خ
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	س	ش	ص	ش	؟
C0	φ	ء	آ	أ	ؤ	ع	ئ	ب	ة	ث	ت	ج	د	خ	د	خ
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ		÷	×	ع	
E0	-	ف	ق	ك	ل	م	ن	ه	و	ي	ض	ع	غ	غ	غ	م
F0	-	ن	ه	و	ي	ض	ع	غ	غ	غ	غ	غ	غ	غ	غ	■

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	Ť	Ž	Ž	
90		‘	’	“	”	•	-	-	™	š	>	š	ť	ž	ž	
A0	˘	˘	Ł	ł	Ą	ą	Ś	ś	©	§	«	¬	¬	®	Ž	
B0	°	±	ı	ı	μ	¶	•	ą	§	»	Ł	”	ł	ž	ž	
C0	Ř	Á	Ā	Ă	Ä	Ć	Č	Č	É	Ě	Ě	Í	Î	Ď		
D0	Đ	Ň	Ň	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Ú	Ú	Ý	Ť	ß
E0	ř	á	â	ă	ä	í	ć	č	é	ě	ě	í	î	ď		
F0	đ	ň	ň	ó	ô	õ	ö	÷	ř	ů	ú	ú	ú	ý	ť	·

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	Ў	ў	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	ˆ	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	\$	@	¡	Ñ	¿	^	`	..	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	¡	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	\$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	\$	@	[	\	]	^	`	{		}	~

図 A-19 国際文字セット



## 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

---

©この取扱説明書の内容は、製品の改良に伴い、予告なしに変更することがあります。

---