MP-A40シリーズ Android™向けSDK
アプリケーションプログラマーズガイド

U00136203608

セイコーインスツル株式会社

U00136203600	2016年 8月
U00136203601	2016年 11月
U00136203602	2018年 1月
U00136203603	2019年 2月
U00136203604	2019年 10月
U00136203605	2022年 5月
U00136203606	2024年 7月
U00136203607	2024年 12月
U00136203608	2026年 6月

©セイコーインスツル株式会社　2016-2026
無断転載を禁じます。

Microsoft®、Windows®、および Microsoft Store は、米国およびその他の国における Microsoft Corporation の商標または登録商標です。
Android™ は、Google LLC.の商標です。
Bluetooth® ワードマークおよびロゴは、Bluetooth SIG, Inc. が所有する登録商標です。
Oracle および Java® は、Oracle Corporation およびその関連会社の米国およびその他の国における登録商標です。
その他の製品名及び会社名は、各社の商標または登録商標です。

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

はじめに

本書では、セイコーインスツル株式会社（以降、SII）製MP-A40シリーズ用Android向けSDK（以降、本SDK)について説明します。

対象プリンタについて

本SDKでサポートするプリンタ製品を下記に記載します。

プリンタ製品	通信インターフェース
MP-A40(Bluetooth搭載モデル)	Bluetooth
	USB
MP-A40(無線LAN搭載モデル)	無線LAN
	USB

目次

1章　製品概要　1-1

1.1　機能 1-1
1.2　構成 1-1
1.2.1　SII プリントクラスライブラリ 1-2
1.2.2　サンプルプログラム 1-2

2章　製品仕様　2-1

2.1　製品仕様
2.1.1　動作環境 2-1
2.1.2　動作条件 2-2
2.1.3　注意事項 2-2

3章　本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　作成した Android アプリケーションの Android デバイスでの利用 3-2
3.3　提供ファイル 3-3
3.4　Android Studio のプロジェクトへの本ライブラリの組み込み 3-4
3.5　注意事項 3-7

4章　本ライブラリの機能　4-1

4.1　本ライブラリの概要 4-1
4.2　本ライブラリの形体 4-1
4.3　本ライブラリのデータ受信処理と制限事項 4-1
4.4　API リファレンス 4-2
4.4.1　インターフェース 4-4
(1)　CallbackFunctionListener 4-4
onStatusChanged　プリンタステータス変化時の処理登録 4-4
(2)　PrinterListener 4-5
onDiscoveryFinished　プリンタ探索の終了イベント 4-5
4.4.2　クラス 4-6
(1)　PrinterManager 4-6
PrinterManager　コンストラクタ 4-8
open　プリンタの利用開始 4-8
close　プリンタの利用終了 4-10
isOpened　プリンタの利用状態取得 4-10
setWriteTimeout　送信タイムアウト時間設定 4-10
getWriteTimeout　送信タイムアウト時間取得 4-11
setResponseTimeout　受信タイムアウト時間設定 4-11
getResponseTimeout　受信タイムアウト時間取得 4-11

write バイナリデータ送信.....4-11
read 受信データの取得.....4-12
getReadSize 取得可能な受信データサイズの取得.....4-12
writeAndWaitResponse バイナリデータ送受信.....4-13
reset プリンタのリセット.....4-14
getStatus プリンタステータスの取得.....4-14
startCallbackFunction
プリンタステータス変化のコールバック開始.....4-15
stopCallbackFunction プリンタステータス変化のコールバック終了.....4-15
registerStyleSheet プリンタへのスタイルシートの登録.....4-15
deleteStyleSheet プリンタ上のスタイルシートの削除.....4-16
registerLogo プリンタへのロゴの登録.....4-16
deleteLogo プリンタ上のロゴの削除.....4-17
getPrinterInformation
getPrinterInformationNumber
getPrinterInformationString
プリンタ情報の取得.....4-18
getVersion SDK バージョンの取得 4-19
controlTransaction 印字データー括送信の開始・終了.....4-20
selectStandardMode スタンダードモードの開始.....4-21
selectPageMode ページモードの開始.....4-22
printPageModeData ページモードデータの印字.....4-23
setPageModeVerticalPosition
ページモードの縦方向絶対位置指定.....4-23
setStandardModeArea スタンダードモードの印字領域設定.....4-23
setStandardModeAlignment
位置揃え.....4-24
setHorizontalPosition
絶対位置指定.....4-25
setStandardModeBarcodeDirection
バーコード、2 次元バーコードの印字方向の選択 4-25
setLineSpacing 改行量の設定.....4-26
setCharacterRightSpace
文字の右スペース量の設定.....4-26
selectCharacterSet 文字セット選択.....4-26
getCharacterSet 設定されている文字セット取得.....4-27
selectInternationalCharacterSet
国際文字セット選択.....4-27
getInternationalCharacter
設定されている国際文字セット取得.....4-27
setCharacterFormatting
文字修飾.....4-28
printText テキストデータ送信.....4-29
printLogo プリンタに登録したロゴの印字.....4-29
sendDataFile ファイルデータ送信.....4-30
sendImage イメージデータ送信.....4-31
printPDF PDF ページの印字 4-32
printBarcode バーコードの印字.....4-33

print2DCode 2次元バーコードの印字 4-37
printPageModeRectangle
ページモードの矩形の描画 4-39
feedLine 行単位の紙送り 4-40
feedDotLine ドット単位の紙送り 4-40
feedCutPosition 用紙カット位置までの移動 4-40
feedMarkPosition マーク紙の頭出し 4-41
setLog ログ出力設定 4-41
(2) PrinterStatus 4-42
getErrOffline オフラインエラー状態の取得 4-42
getErrHardware ハードウェアエラー状態の取得 4-42
getErrVoltage Vp電圧エラー状態の取得 4-43
getErrHeadTemperature
ヘッド温度エラー状態の取得 4-43
getErrOutOfPaper 紙無しエラー状態の取得 4-43
getErrMarkPaperJam マーク検出時紙ジャムエラー状態の取得 4-43
getErrCoverOpen ペーパーカバーオープンエラー状態の取得 4-43
getErrBattery バッテリエラー状態の取得 4-44
getStateFeedSwitch フィードスイッチ状態の取得 4-44
getStatePaperFeed 紙送り状態の取得 4-44
getStateReturnWaiting
復帰待ち状態の取得 4-44
getStateFlashMemoryRewriting
フラッシュメモリ書き換え状態の取得 4-44
getStateBattery バッテリ電圧状態の取得 4-45
(3) PrinterDiscovery 4-46
PrinterDiscovery コンストラクタ 4-46
startDiscoveryPrinter
プリンタの探索開始 4-46
cancelDiscoveryPrinter
プリンタの探索中断 4-47
getFoundPrinter 発見されたプリンタ情報リストの取得 4-48
(4) DiscoveredPrinter 4-49
getPrinterModel プリンタモデル名の取得 4-49
getBluetoothAddress Bluetoothアドレスの取得 4-49
getIpAddress IPアドレスの取得 4-49
getMacAddress MACアドレスの取得 4-49
getSerialNumber シリアルナンバーの取得 4-50
4.4.3 列挙型 4-51
(1) PrinterInterface 4-51
(2) PrinterModel 4-51
(3) Dithering 4-51
(4) PrinterInformation 4-51
(5) TransactionFunction 4-52
(6) Direction 4-53
(7) Alignment 4-53
(8) CharacterSet 4-54
(9) InternationalCharacterSet 4-55

(10) CharacterType....4-55
(11) CharacterScale....4-56
(12) Underline....4-56
(13) Bold....4-56
(14) Reverse....4-56
(15) Rotate....4-57
(16) TypeBarcode....4-58
(17) ModuleWidthBarcode....4-58
(18) HriPositionBarcode....4-59
(19) NwRatioBarcode....4-59
(20) Type2DCode....4-59
(21) Mode2Dcode....4-59
(22) ModuleSize2Dcode....4-61
(23) ErrorCorrect2Dcode....4-63
(24) LogFileSize....4-64
(25) ErrorCode....4-64
4.4.4 例外....4-65
(1) PrinterException....4-65
getErrorCode エラーコードの取得....4-65

5章 サンプルプログラム 5-1

5.1 サンプルプログラムの概要....5-1
5.2 サンプルプログラムの利用方法....5-1
5.3 サンプルプログラムの機能....5-4
5.4 注意事項....5-5

6章 免責 6-1

付録A 文字セット（文字コード表） A-1

A.1 文字コード表（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、本SDKの製品概要について説明します。

1.1　機能

本SDKに含まれる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MP-A40シリーズ（以降、プリンタ）を利用するための機能を提供します。
また、本SDKは、SIIプリントクラス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　構成

本SDKに含まれるSIIプリントクラスライブラリとサンプルプログラムは、Android OSの構成図上の破線で囲まれた部分に位置します。（図 1-1）

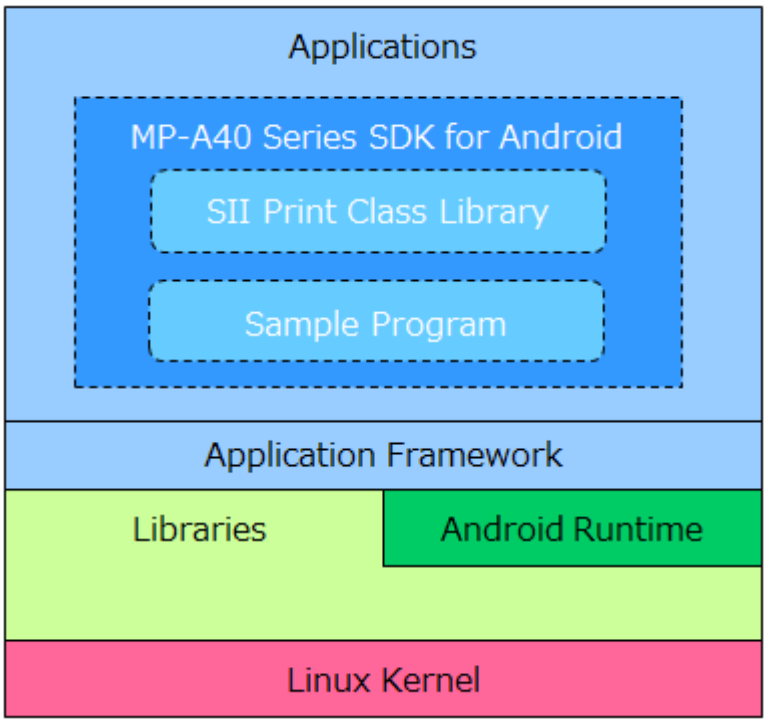


図 1-1

1.2.1 SIIプリントクラスライブラリ

SIIプリントクラスライブラリ（以降、本ライブラリ）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、プリンタに対して印字データやプリンタコマンドを容易に送信することができます。また、プリンタステータスを取得することができます。

本ライブラリでは、下記の機能を提供します。

- プリンタとの接続及び切断
- プリンタへのデータ送信(印字データやプリンタコマンド)
- バーコードの印字、2次元バーコードの印字
- プリンタへのデータファイル送信（印字データやプリンタコマンド）
- プリンタステータスの取得
- プリンタからの各種応答取得
- Bluetooth、またはTCP/IPによるプリンタの探索

1.2.2 サンプルプログラム

本ライブラリを使用したAndroidアプリケーションの実装サンプルとして提供します。

2章　製品仕様

本章では、本ライブラリの製品仕様について説明します。

2.1　製品仕様

2.1.1　動作環境

本ライブラリの動作環境を下記に示します。

<table>
<tr><td rowspan="3">プリンタ</td><td colspan="2">モデル</td><td colspan="3">MP-A40</td></tr>
<tr><td colspan="2">F/Wバージョン</td><td colspan="3">1.00以降</td></tr>
<tr><td colspan="2">通信インターフェース</td><td>Bluetooth</td><td>USB</td><td>無線LAN</td></tr>
<tr><td rowspan="15">Android
デバイス</td><td colspan="2">通信ポート</td><td>Bluetooth*1</td><td>USB*2</td><td>TCP/IP*3</td></tr>
<tr><td rowspan="14">OS</td><td>Android 7.0 (API 24)</td><td rowspan="14">対応</td><td rowspan="14">対応</td><td rowspan="14">対応</td></tr>
<tr><td>Android 7.1 (API 25)</td></tr>
<tr><td>Android 8.0 (API 26)</td></tr>
<tr><td>Android 8.1 (API 27)</td></tr>
<tr><td>Android 9.0 (API 28)</td></tr>
<tr><td>Android 10.0 (API 29)</td></tr>
<tr><td>Android 11.0 (API 30)</td></tr>
<tr><td>Android 12.0 (API 31)</td></tr>
<tr><td>Android 12.1 (API 32)</td></tr>
<tr><td>Android 13.0 (API 33)</td></tr>
<tr><td>Android 14.0 (API 34)</td></tr>
<tr><td>Android 15.0 (API 35)</td></tr>
<tr><td>Android 16.0 (API 36)</td></tr>
<tr><td colspan="3">サポート言語</td><td colspan="3">日本語
英語</td></tr>
</table>

（注意）　*1: SPP(Serial Port Profile)によりBluetooth接続が確立されている必要があります。
*2: AndroidデバイスがUSBホスト機能をサポートしている必要があります。
*3: Androidデバイスとプリンタが同じネットワークに接続されている必要があります。
プリンタがクライアントモードのときはアクセスポイント経由で、シンプルAP
モードのときは直接接続してください。

2.1.2 動作条件

本ライブラリの動作条件を下記に示します。
本ライブラリの利用時には、予めプリンタの機能設定を下記に示す値に設定してください。
機能設定の詳細については、「MP-A40シリーズ サーマルプリンタ 技術説明書」を参照してください。

MS	機能	設定	設定内容
1-1	通信選択 (Interface)	0 / 1	0: USBのみ (USB) 1: USB/Wireless*1
1-2	マークモード選択 (Mark Mode)	0 / 1	0: 有効 (Enable)*2 1: 無効 (Disable)
1-3	コマンド体系選択 (Command System)	0	ESC/POS
1-4		0	
1-5		0	
2-2	リアルタイムコマンド選択 (Realtime Command)	0	有効 (Enable)
9-1	自動ステータス応答機能選択<ESC/POS> (Auto Status Back)	0	有効 (Enable)
9-2	イニシャライズ完了応答選択<ESC/POS> (Init. Response)	0	有効 (Enable)

（注意） *1: Bluetoothおよび無線LANインターフェースでの使用時は「USB/Wireless」を選択してください。
*2: マーク紙の頭出しを行う場合は「有効」を選択してください。

2.1.3 注意事項

本ライブラリはTCP/IP接続時、プリンタのポートを占有します。そのため、プリンタドライバ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　本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本ライブラリの利用方法について説明します。

3.1　Androidアプリケーションの開発環境

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

- Android Studio
  http://developer.android.com/sdk/index.html
- Windows用USBドライバ（Windows環境で開発する場合）
  http://developer.android.com/tools/extras/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　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（実機）上で利用するためには、Androidデバイス（実機）で下記の設定を行ってください。

（注意）　本手順はAndroid 6.0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。

(a)　[設定] – [セキュリティ] – [提供元不明のアプリ] を有効にしてください。（図 3-1）



図 3-1

(b)　[設定] – [開発者向けオプション] – [USBデバッグ]にチェックを入れてください。（図 3-2）



図 3-2

3.3 提供ファイル

本ライブラリのファイル構成は、下記の通りです。

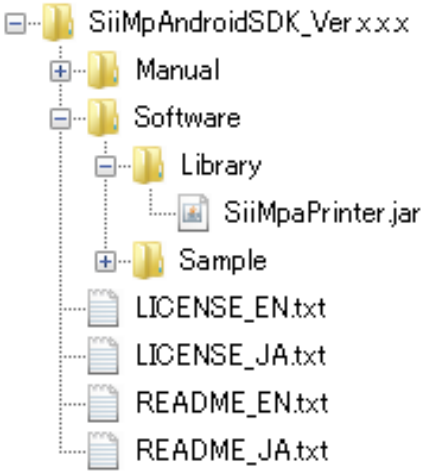


図 3-3

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

3.4　Android Studioのプロジェクトへの本ライブラリの組み込み

本SDK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、本ライブラリをAndroid Studioのプロジェクトに組込む手順を説明します。

本SDK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。

(a)　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 3-4)、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 3-5)。

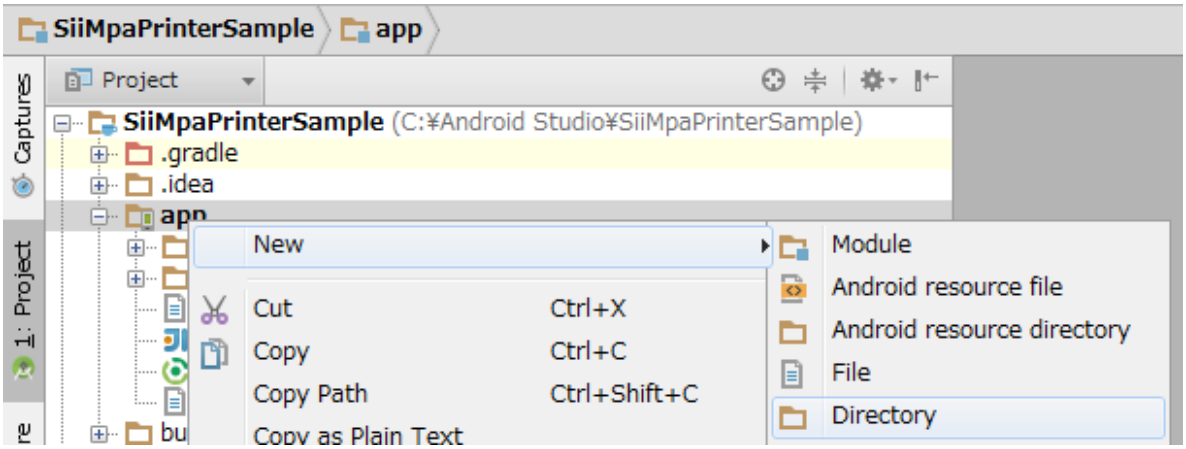


図 3-4

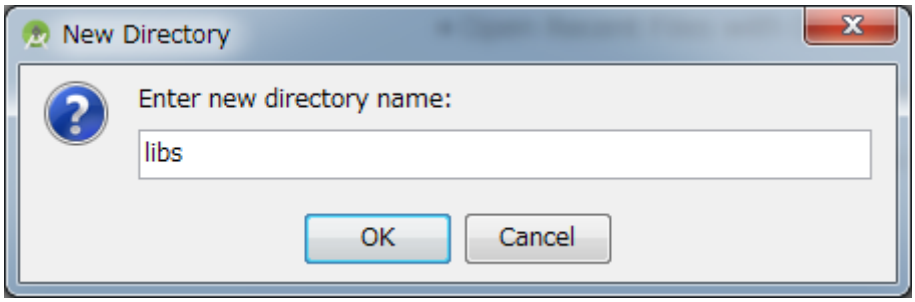


図 3-5

(b)　(a)で作成したフォルダ(¥SiiMpaPrinterSample¥app¥libs)に本ライブラリファイル(SiiMpaPrinter.jar)をコピーしてください。

(c)　SiiMpaPrinter.jarを右クリックして、[Add as library…]を選択し、モジュール(app)を選択してください。（図 3-6）

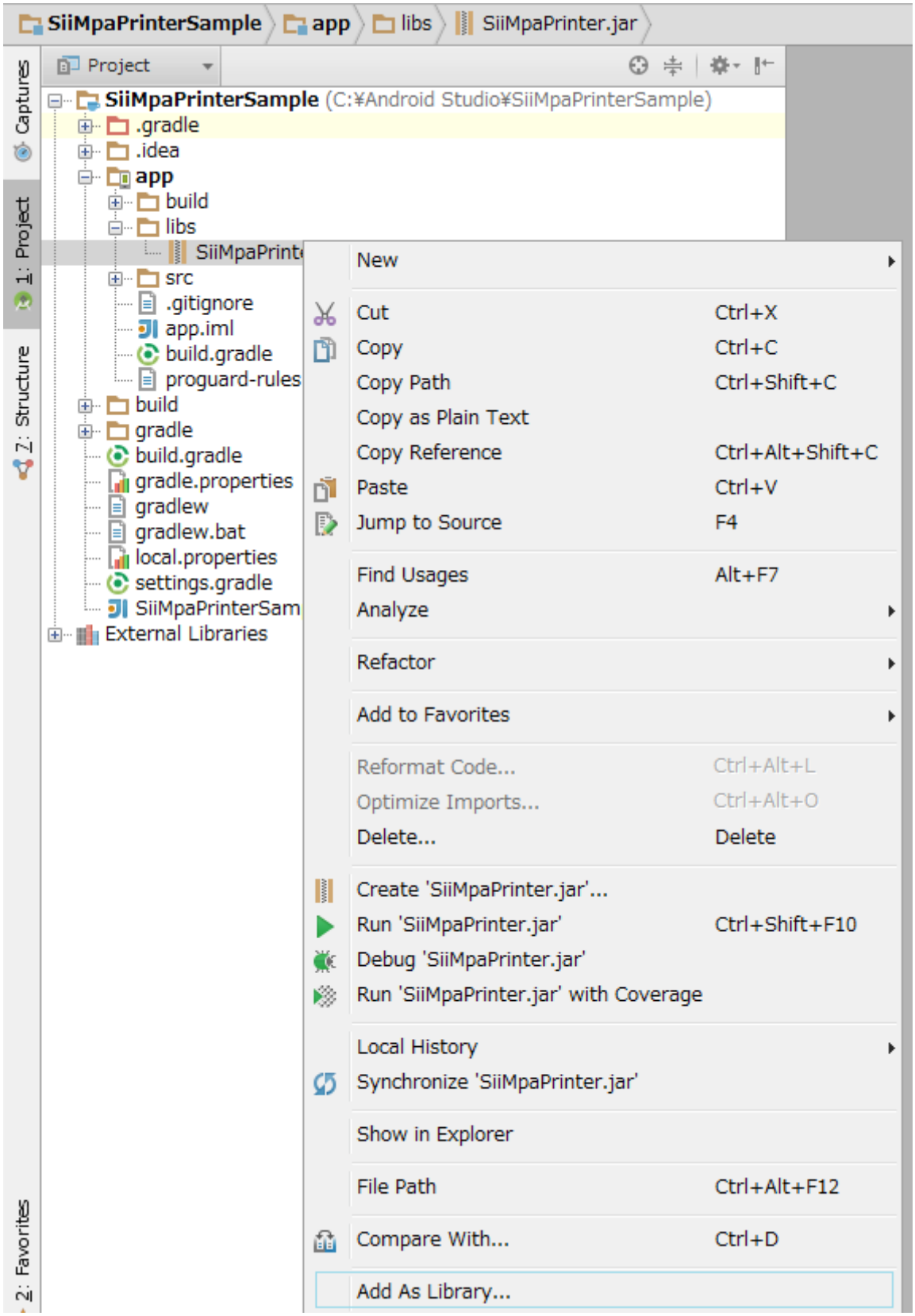


図 3-6

(d)　本ライブラリの追加が完了すると(図 3-7)の状態となります。

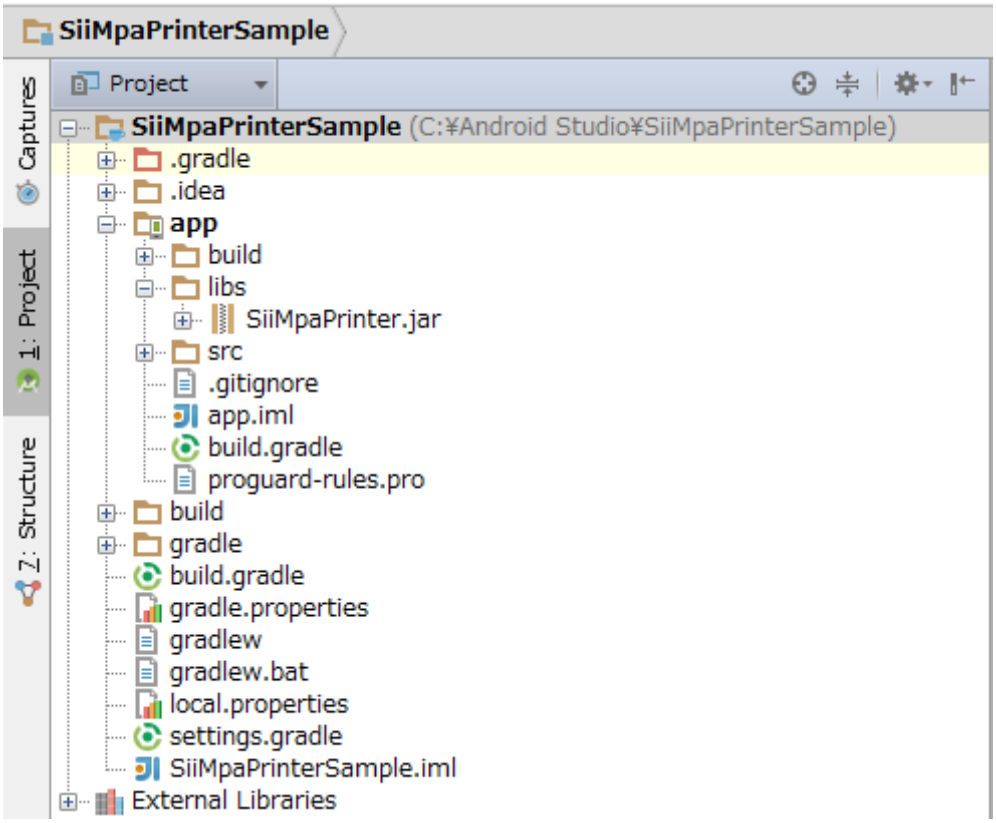


図 3-7

(e)　メインとなるソースファイルの先頭に下記を追加してください。
（SiiMpaPrinterSampleでは、MainActivity.java）

```
import com.seikoinstruments.sdk.mobileprinter.Alignment;
import com.seikoinstruments.sdk.mobileprinter.Bold;
import com.seikoinstruments.sdk.mobileprinter.CharacterScale;
import com.seikoinstruments.sdk.mobileprinter.CharacterSet;
import com.seikoinstruments.sdk.mobileprinter.CharacterType;
import com.seikoinstruments.sdk.mobileprinter.Direction;
import com.seikoinstruments.sdk.mobileprinter.ErrorCorrect2Dcode;
import com.seikoinstruments.sdk.mobileprinter.ErrorCode;
import com.seikoinstruments.sdk.mobileprinter.LogFileSize;
import com.seikoinstruments.sdk.mobileprinter.HriPositionBarcode;
import com.seikoinstruments.sdk.mobileprinter.InternationalCharacterSet;
import com.seikoinstruments.sdk.mobileprinter.Mode2Dcode;
import com.seikoinstruments.sdk.mobileprinter.ModuleSize2Dcode;
import com.seikoinstruments.sdk.mobileprinter.ModuleWidthBarcode;
import com.seikoinstruments.sdk.mobileprinter.NwRatioBarcode;
import com.seikoinstruments.sdk.mobileprinter.PrinterException;
import com.seikoinstruments.sdk.mobileprinter.PrinterInterface;
import com.seikoinstruments.sdk.mobileprinter.PrinterInformation;
import com.seikoinstruments.sdk.mobileprinter.PrinterManager;
import com.seikoinstruments.sdk.mobileprinter.PrinterModel;
import com.seikoinstruments.sdk.mobileprinter.PrinterStatus;
import com.seikoinstruments.sdk.mobileprinter.Reverse;
import com.seikoinstruments.sdk.mobileprinter.Rotate;
import com.seikoinstruments.sdk.mobileprinter.TransactionFunc;
import com.seikoinstruments.sdk.mobileprinter.Type2DCode;
import com.seikoinstruments.sdk.mobileprinter.TypeBarcode;
import com.seikoinstruments.sdk.mobileprinter.Underline;
```

以上で、本ライブラリの機能が利用可能になります。

3.5 注意事項

- **対象範囲別ストレージについて**

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

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

4章　本ライブラリの機能

本章では、本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　本ライブラリの概要

本ライブラリは、Androidアプリケーションに対して、プリンタを利用するための機能を提供します。

4.2　本ライブラリの形体

本ライブラリは、jarファイル形式です。本ライブラリのファイル名は、**SiiMpaPrinter.jar**です。
Androidアプリケーションで、本ライブラリを利用する場合、Android Studioのプロジェクトに本ライブラリを組み込んでください。
本ライブラリをAndroid Studioのプロジェクトに組み込む方法は『3章　本ライブラリの利用方法』を参照してください。

4.3　本ライブラリのデータ受信処理と制限事項

本ライブラリは、**open**によるプリンタの利用開始後、プリンタからの受信データをSDK内部の受信バッファ（以降、バッファ）にバッファリングします。プリンタは状態が変化するたびに自動ステータス応答を返すため、受信データはバッファに順次バッファリングされます。バッファリングされる受信データのサイズは最大4096バイトです。プリンタが再接続された場合、接続前にプリンタにバッファリングされていたデータをまとめて受信する場合があります。
バッファにバッファリングされた受信データは、**read**により取得が可能です。**read**により取得した受信データは、バッファから削除されます。
バッファリングされた受信データが4096バイトを超えた場合、古いデータからバッファのデータが破棄されますので、4096バイトを超えないように**read**を実行してください。受信データのサイズは、**getReadSize**で取得が可能です。
各メソッドの詳細は、『4.4 APIリファレンス』を参照してください。
各種応答の詳細は、『MP-A40シリーズ　サーマルプリンタ　技術説明書』を参照してください。

4.4 APIリファレンス

本ライブラリのパッケージは、com.seikoinstruments.sdk.mobileprinterです。

com.seikoinstruments.sdk.mobileprinterは、下記の機能を提供します。

- **インターフェース**

名前	説明
CallbackFunctionListener	プリンタステータス変化時の処理を実装するためのインターフェース
PrinterListener	プリンタの探索の終了イベントを取得するためのインターフェース

- **クラス**

名前	説明
PrinterManager	プリンタとの通信、印字を行うためのAPIを提供するクラス
PrinterStatus	プリンタステータスを格納するためのクラス
PrinterDiscovery	プリンタ探索を行うためのクラス
DiscoveredPrinter	プリンタ探索で発見されたプリンタ情報の格納クラス

- **列挙型**

名前	説明
PrinterInterface	**open**の通信インターフェース指定で利用する列挙型定数
PrinterModel	**open**のプリンタモデル指定で利用する列挙型定数
Dithering	**registerLogo**、**sendDataFile**、**printPDF**のディザリングで利用する列挙型定数
PrinterInformation	**getPrinterInformation**、**getPrinterInformationNumber**、**getPrinterInformationString**のプリンタ情報指定で利用する列挙型定数
TransactionFunction	**controlTransaction**の一括処理制御方法指定で利用する列挙型定数
Direction	**selectPageMode**の印字方向指定で利用する列挙型定数
Alignment	**setStandardModeAlignment**の位置揃え指定で利用する列挙型定数
CharacterSet	**selectCharacterSet**、及び**getCharacterSet**の文字セット選択で利用する列挙型定数
InternationalCharacterSet	**selectInternationalCharacterSet**、及び**getInternationalCharacter**の国際文字セット選択で利用する列挙型定数
CharacterType	**setCharacterFormatting**の文字フォント指定で利用する列挙型定数
CharacterScale	**setCharacterFormatting**の文字サイズ指定で利用する列挙型定数
Underline	**setCharacterFormatting**のアンダーライン指定で利用する列挙型定数
Bold	**setCharacterFormatting**の強調印字指定で利用する列挙型定数
Reverse	**setCharacterFormatting**の白黒反転印字指定で利用する列挙型定数
Rotate	**setCharacterFormatting**の回転印字指定、**setStandardModeBarcodeDirection**の印字方向指定、**printPDF**のイメージの回転方向指定で利用する列挙型定数
TypeBarcode	**printBarcode**のバーコードの種類指定で利用する列挙型定数
ModuleWidthBarcode	**printBarcode**のバーコードのモジュール幅または細エレメント指定で利用する列挙型定数
HriPositionBarcode	**printBarcode**のバーコードのHRI文字指定で利用する列挙型定数

名前	説明
NwRatioBarcode	**printBarcode**のバーコードのN:W比指定で利用する列挙型定数
Type2DCode	**print2DCode**の2次元バーコードの種類指定で利用する列挙型定数
Mode2Dcode	**print2DCode**の2次元バーコードのモードで利用する列挙型定数
ModuleSize2Dcode	**print2DCode**の2次元バーコードのモジュールサイズ指定で利用する列挙型定数
ErrorCorrect2Dcode	**print2DCode**の2次元バーコードのエラー訂正レベル指定で利用する列挙型定数
LogFileSize	**setLog**のログファイルの最大サイズ指定で利用する列挙型定数
ErrorCode	**PrinterException**で取得できるエラーコードの列挙型定数

- **例外**

名前	説明
PrinterException	API呼び出し時にスローされる例外クラス

4.4.1　インターフェース

(1) **CallbackFunctionListener**

- **概要**

  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ースです。
  下記の機能を提供します。

  Public Methods

名前	説明
onStatusChanged	プリンタステータス変化時の処理登録

- **Public Methods**

onStatusChanged　　プリンタステータス変化時の処理登録

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

パラメーター　*status*
　　PrinterStatusクラスのインスタンス

説明　　本メソッドは、**startCallbackFunction**によりプリンタステータス変化のコールバックが開始し、プリンタステータスが変化した場合に呼び出されます。

　　インターフェースのメソッドであるため、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にて、任意の処理を実装してください。

　　*status*には、変化した時点のプリンタステータスを含んだ**PrinterStatus**クラスのインスタンスを代入します。プリンタステータスは、*status*から**PrinterStatus**クラスのメソッドにより取得できます。**PrinterStatus**クラスのメソッドは、『**4.4.2**(2) **PrinterStatus**』を参照してください。

(2) **PrinterListener**

- **概要**

イベントを取得するためのインターフェースです。
下記の機能を提供します。

Public Methods

名前	説明
onDiscoveryFinished	プリンタ探索の終了イベント

Constants

名前	説明
EVENT_FINISHED_DISCOVERY	プリンタ探索の終了イベント 定数値=0
EVENT_CANCELED_DISCOVERY	プリンタ探索の中断 定数値=1

- **Public Methods**

onDiscoveryFinished　プリンタ探索の終了イベント

構文　public void **onDiscoveryFinished** (int *event*)

パラメーター　*event*
　プリンタイベント

説明　本メソッドは、**startDiscoveryPrinter**による探索が終了した場合、または**cancelDiscoveryPrinter**により探索が中断された場合に呼び出されます。

インターフェースのメソッドであるため、実装はありません。プリンタ探索の終了や中断のイベント通知を受け取るクラスにて、任意の処理を実装してください。

探索の終了により呼び出された場合、*event*には**EVENT_FINISHED_DISCOVERY**が代入されます。

探索の中断により呼び出された場合、*event*には**EVENT_CANCELED_DISCOVERY**が代入されます。

4.4.2 クラス

(1) **PrinterManager**

- **概要**

  プリンタとの通信、印字を行うための API を提供するクラスです。
  下記の機能を提供します。

  Public Methods

名前	説明	スタンダードモード[*1]	ページモード[*1]
PrinterManager	コンストラクタ	✓	✓
open	プリンタの利用開始	✓	✓
close	プリンタの利用終了	✓	✓
isOpened	プリンタの利用状態取得	✓	✓
setWriteTimeout	送信タイムアウト時間設定	✓	✓
getWriteTimeout	送信タイムアウト時間取得	✓	✓
setResponseTimeout	受信タイムアウト時間設定	✓	✓
getResponseTimeout	受信タイムアウト時間取得	✓	✓
write	バイナリデータ送信	✓	✓
read	受信データの取得	✓	✓
getReadSize	取得可能な受信データサイズの取得	✓	✓
writeAndWaitResponse	バイナリデータ送受信	✓	✓
reset	プリンタのリセット	✓	✓
getStatus	プリンタステータスの取得	✓	✓
startCallbackFunction	プリンタステータス変化のコールバック開始	✓	✓
stopCallbackFunction	プリンタステータス変化のコールバック終了	✓	✓
registerStyleSheet	プリンタへのスタイルシートの登録	✓	✓
deleteStyleSheet	プリンタ上のスタイルシートの削除	✓	✓
registerLogo	プリンタへのロゴの登録	✓	✓
deleteLogo	プリンタ上のロゴの削除	✓	✓
getPrinterInformation	プリンタ情報の取得	✓	✓
getPrinterInformationNumber		✓	✓
getPrinterInformationString		✓	✓
getVersion	SDKバージョンの取得	✓	✓
controlTransaction	印字データー括送信の開始・終了	✓	✓
selectStandardMode	スタンダードモードの開始	−	✓
selectPageMode	ページモードの開始	✓	−
printPageModeData	ページモードデータの印字	−	✓[*2]
setPageModeVerticalPosition	ページモードの縦方向絶対位置設定	−	✓[*2]
setStandardModeArea	スタンダードモードの印字領域設定	✓[*3]	−
setStandardModeAlignment	位置揃え	✓[*3]	−

名前	説明	スタンダードモード*1	ページモード*1
setHorizontalPosition	絶対位置設定	✓	✓
setStandardModeBarcodeDirection	バーコード、2次元バーコードの印字方向の選択	✓*3	–
setLineSpacing	改行量の設定	✓*4	✓*4
setCharacterRightSpace	文字の右スペース量の設定	✓*4	✓*4
selectCharacterSet	文字セット選択	✓	✓
getCharacterSet	設定されている文字セット取得	✓	✓
selectInternationalCharacterSet	国際文字セット選択	✓	✓
getInternationalCharacter	設定されている国際文字セット取得	✓	✓
setCharacterFormatting	文字修飾	✓	✓
printText	テキストデータ送信	✓	✓
printLogo	プリンタに登録されたロゴの印字	✓	✓
sendDataFile	ファイルデータ送信	✓	✓
sendImage	イメージデータ送信	✓	✓
printPDF	PDFページの印字	✓*3	–
printBarcode	バーコードの印字	✓	✓
print2DCode	2次元バーコードの印字	✓	✓
printPageModeRectangle	ページモードの矩形の描画	–	✓*2
feedLine	行単位の紙送り	✓	✓
feedDotLine	ドット単位の紙送り	✓	✓
feedCutPosition	用紙カット位置までの移動	✓	✓
feedMarkPosition	マーク紙の頭出し	✓	✓
setLog	ログ出力設定	✓	✓

*1　✓：使用可、－：使用不可
*2　本メソッドを使用する場合、事前に**selectPageMode**を実行し、ページモードを開始してください。
*3　本メソッドを使用する場合、事前に**selectStandardMode**を実行し、スタンダードモードを開始してください。
*4　スタンダードモードとページモードでそれぞれ独立した設定が可能です。

Constants

定数名	説明
PAPER_WIDTH_80mm	プリンタのメモリスイッチMS3-1～6（紙幅選択）が80mmの時の印字幅（ドット） 定数値=576
PAPER_WIDTH_100mm	プリンタのメモリスイッチMS3-1～6（紙幅選択）が100mmの時の印字幅（ドット） 定数値=736
PAPER_WIDTH_105mm	プリンタのメモリスイッチMS3-1～6（紙幅選択）が105mmの時の印字幅（ドット） 定数値=776
PAPER_WIDTH_112mm	プリンタのメモリスイッチMS3-1～6（紙幅選択）が112mmの時の印字幅（ドット） 定数値=832

- **Public Methods**

PrinterManager　コンストラクタ

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

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

パラメーター　*context*
本メソッドを呼び出すAndroidアプリケーションのコンテキスト。
getApplicationContext()で取得できる*context*を指定してください。

open　プリンタの利用開始

プリンタの利用を開始します。

構文　public void **open**(PrinterInterface *prnIf*,
PrinterModel *prnModel*,
String *address*,
int *socketKeepingTime*,
boolean *secure*) throws **PrinterException**

パラメーター　*prnIf*
通信インターフェース定数
利用可能な設定は、『**4.4.3**(1) **PrinterInterface**』を参照してください。

prnModel
プリンタモデル定数
利用可能な設定は、『**4.4.3**(2) **PrinterModel**』を参照してください。

address
*prnIf*の設定により異なります。
・**PRN_IF_TCP**を指定する場合
*address*には接続するプリンタのIPアドレス、またはホスト名を指定してください。
例：″192.168.0.190″、″host″
・**PRN_IF_BT**を指定する場合
*address*には接続するプリンタのBluetoothアドレスを指定してください。
例：″00:11:22:AA:BB:CC″
・**PRN_IF_USB**を指定する場合
指定した値は無視されます。

socketKeepingTime

*prnIf*の設定により異なります。

・**PRN_IF_TCP**を指定する場合

*socketKeepingTime*の有効範囲は60～300です。
60未満の値を設定した場合は60に、300を超える値を指定した場合は300に設定されます。*socketKeepingTime*には、プリンタの無線LAN設定である接続無通信タイムアウト時間と同じ値を指定してください。
接続無通信タイムアウト時間のデフォルト値は300秒です。
接続無通信タイムアウト時間の詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

・**PRN_IF_BT**を指定する場合

指定した値は無視されます。

・**PRN_IF_USB**を指定する場合

指定した値は無視されます。

secure

*prnIf*の設定により異なります。

・**PRN_IF_TCP**を指定する場合

指定した値は無視されます。

・**PRN_IF_BT**を指定する場合

trueのときセキュアなモードでプリンタと接続します。
falseのときインセキュアなモードでプリンタと接続します。
通常はtrueを指定してください。
Androidデバイスの一部には、セキュアなモードでの接続に失敗するものがあるため、その様なAndroidデバイスとプリンタを接続する場合には、falseを指定してください。

・**PRN_IF_USB**を指定する場合

指定した値は無視されます。

例外　**PrinterException**

本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　本メソッドは、*prnIf*で指定した通信インターフェースにて、*prnModel*で指定したプリンタに接続します。

本メソッドにより、プリンタステータスのモニタリングを開始します。最新のプリンタステータスは**getStatus**で取得が可能です。プリンタステータスの変化は、**onStatusChanged**、**startCallbackFunction**、及び**stopCallbackFunction**により、イベントとして通知することが可能です。

注意　プリンタコマンド「自動ステータス送信の有効・無効」や機能設定により、自動ステータス応答を無効にしないでください。プリンタステータスのモニタリングができず、関連する機能が動作しません。自動ステータス応答の詳細や機能設定については『MP-A40シリーズ サーマルプリンタ技術説明書』を参照してください。

TCP/IP接続でのプリンタへのデータ送信後、ソケット維持時間が経過すると本ライブラリ内部の通信ソケットを破棄します。その後、再度、通信ソケットを作成し接続します。そのため、通信ソケット破棄の時点でプリンタが同じネットワーク上の別のホストから接続要求を受けていた場合、プリンタはそちらのホストとの通信を確立するため、再接続に失敗する可能性があります。

close　プリンタの利用終了

プリンタの利用を終了し、プリンタステータスのモニタリングを終了します。

構文　public void **close**() 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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　**controlTransaction**によって保持している印字データは、破棄されます。

isOpened　プリンタの利用状態取得

openによるプリンタの利用開始状態を取得します。

構文　public boolean **isOpened**()

戻り値　true　:プリンタの利用が開始されている。
false　:プリンタの利用が開始されていない。

setWriteTimeout　送信タイムアウト時間設定

送信タイムアウト時間を設定します。

構文　public void **setWriteTimeout**(int *writeTimeout*)

パラメーター　*writeTimeout*
送信タイムアウト時間(ミリ秒)
有効範囲は、1000～90000です。
1000未満の値を設定した場合は1000に設定されます。
90000を超える値を指定した場合は90000に設定されます。
本メソッドで設定しない場合、送信タイムアウト時間は、初期値の10000になります。

説明　設定した値は、**getWriteTimeout**で取得できます。

本メソッドの設定は、下記のメソッドに有効です。

- **write**
- **writeAndWaitResponse**（送信処理部分）
- **registerLogo**
- **registerStyleSheet**
- **controlTransaction**（*control*で**TRANSACTION_PRINT**選択時）
- **printText**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printBarcode**
- **print2DCode**

getWriteTimeout　送信タイムアウト時間取得

送信タイムアウト時間を取得します。

構文　public int **getWriteTimeout**()

戻り値　送信タイムアウト時間(ミリ秒)

setResponseTimeout　受信タイムアウト時間設定

受信タイムアウト時間を設定します。

構文　public void **setResponseTimeout**(int *respTimeout*)

パラメーター　*respTimeout*
受信タイムアウト時間(ミリ秒)
有効範囲は、1000～90000です。
1000未満の値を設定した場合は1000に設定されます。
90000を超える値を指定した場合は90000に設定されます。
本メソッドで設定しない場合、受信タイムアウト時間は、初期値の10000になります。

説明　設定した値は、**getResponseTimeout**で取得できます。

本メソッドの設定は、下記のメソッドに有効です。
- **writeAndWaitResponse**
- **getPrinterInformation**
- **getPrinterInformationNumber**
- **getPrinterInformationString**

getResponseTimeout　受信タイムアウト時間取得

受信タイムアウト時間を取得します。

構文　public int **getResponseTimeout**()

戻り値　受信タイムアウト時間(ミリ秒)

write　バイナリデータ送信

バイナリデータを送信します。

構文　public void **write**(byte[] *binary*, int *offset*) throws **PrinterException**

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

offset
送信するデータの開始位置の指定

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) PrinterException』を参照してください。

説明　　**controlTransaction**を使用しない場合、本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

送信データには、プリンタコマンド「プリンタの初期化」以外のプリンタを初期化するプリンタコマンドを含めないでください。ハードウェアリセットを行う場合、**reset**を使用してください。プリンタの初期化については、『MP-A40シリーズ　サーマルプリンタ 技術説明書』を参照してください。

本メソッドは**reset**により中断されます。

read　受信データの取得

バッファにバッファリングされた受信データを取得します。

構文　　public byte[] **read**(int *bufferSize*) throws **PrinterException**

パラメーター　*bufferSize*
受信データサイズ(バイト)
有効範囲は、1～4096です。
4096を超える値を指定した場合、4096に設定されます。
0以下の値を指定した場合、エラーを通知します。

戻り値　　受信データ

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　バッファにバッファリングされた受信データを全て取得する場合は、*bufferSize*に**getReadSize**で取得した値を指定してください。

本メソッドによる受信データの取得後、再度本メソッドを実行すると、バッファから続きの受信データを取得します。

受信データの処理と制限事項については、『4.3 本ライブラリのデータ受信処理と制限事項』を参照してください。

getReadSize　取得可能な受信データサイズの取得

取得可能な受信データサイズを取得します。

構文　　public int **getReadSize**() throws **PrinterException**

戻り値　　取得可能な受信データサイズ(バイト)

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

バイナリデータを送信し、本メソッドが実行された時点からの受信データを取得します。

構文　public byte[] **writeAndWaitResponse**(byte[] *sendBinary*,
int *bufferSize*,
boolean *respControl*,
boolean *incAsbData*) throws **PrinterException**

パラメーター　*sendBinary*
送信データ
指定可能なデータサイズは、最大3MB(3145728バイト)です。

bufferSize
受信データの最大サイズ(バイト)
有効範囲は、1～10485760です。
10485760を超える値を指定した場合、10485760に設定されます。
0以下の値を指定した場合、エラーを通知します。

respControl
受信処理の動作選択
true　:何らかのデータを受信するか、タイムアウトになるまで受信を継続します。
false　:*bufferSize*で指定したサイズのデータを受信するか、タイムアウトになるまで受信を継続します。

incAsbData
受信データに自動ステータス応答を含むか否か
true　:含む
false　:含まない

戻り値　受信データ

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　本メソッドは文字列や容量などの値を応答するプリンタコマンドの送信と、その応答取得に適しています。

本メソッドの実行後、次のメソッドを実行しても本メソッドの受信が完了するまでは処理されません。実行応答や進捗応答を取得する場合、本メソッドではなく**read**を使用してください。実行応答や進捗応答の詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

本メソッドの受信タイムアウト時間は、**setResponseTimeout**で設定した値になります。

送信データには、プリンタコマンド「プリンタの初期化」以外のプリンタを初期化するプリンタコマンドを含めないでください。ハードウェアリセットを行う場合、**reset**を使用してください。プリンタの初期化については、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

本メソッドは**reset**により中断されます。

reset　プリンタのリセット

プリンタをリセットします。

構文　public void **reset**() 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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　下記のメソッド処理中に本メソッドを実行すると、処理が中断されます。下記のメソッドの未送信データ、未受信データは破棄されます。

- **write**
- **writeAndWaitResponse**
- **controlTransaction**（*control*で**TRANSACTION_PRINT**選択時）
- **printText**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **getPrinterInformation**
- **getPrinterInformationNumber**
- **getPrinterInformationString**

本メソッド実行中、プリンタステータスはオフラインとなります。

本メソッド実行後、数秒経過してからデータ送信のメソッドを実行してください。リセット中にデータ送信を行うと、データ抜けの原因となります。

Bluetooth接続で、プリンタがデータを受け付けない状態で本メソッドを実行した場合、本メソッドは成功しますが、リセットはプリンタが印字可能状態となるまで実行されません。 また、それまでの間データ送信は行えません。

getStatus　プリンタステータスの取得

最新のプリンタステータスを取得します。

構文　public PrinterStatus **getStatus**() throws **PrinterException**

戻り値　プリンタステータスを**PrinterStatus**クラスで返します。
PrinterStatusクラスのメソッドにより、プリンタステータスの内容が取得できます。
PrinterStatusクラスのメソッドは、『**4.4.2**(2) **PrinterStatus**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

startCallbackFunction　プリンタステータス変化のコールバック開始

プリンタ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します。

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

パラメーター　*listener*
CallbackFunctionListener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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』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**にて登録します。

stopCallbackFunction　プリンタステータス変化のコールバック終了

startCallbackFunctionで開始されたコールバックを終了します。

構文　public void **stopCallbackFunction**() 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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

registerStyleSheet　プリンタへのスタイルシートの登録

スタイルシートをプリンタに登録します。

構文　public void **registerStyleSheet**(String *filePath*, int *regNum*) throws **PrinterException**

パラメーター　*filePath*
スタイルシートのファイルパス
入力可能な形式について下記に説明します。

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

・Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
・file://
・content://
本パラメーターには"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。

regNum
スタイルシート番号
有効範囲は、1～4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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　最大4シートを登録可能です。

サポートするスタイルシートのファイル拡張子は、.cssです。

登録可能なファイルサイズは、最大1MB(1048576バイト)です。

一つのスタイルシートに登録できるスタイル数は最大64です。

既にスタイルシートが登録されているスタイルシート番号を指定した場合、スタイルシートを上書きします。

登録可能なスタイルシートの詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』のプリンタコマンド「スタイルシート登録」を参照してください。

本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

deleteStyleSheet　プリンタ上のスタイルシートの削除

プリンタに登録されているスタイルシートを削除します。

構文　　public void **deleteStyleSheet**(int *regNum*) throws **PrinterException**

パラメーター　*regNum*
スタイルシート番号
有効範囲は、1～4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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

registerLogo　プリンタへのロゴの登録

イメージデータをロゴとしてプリンタに登録します。
構文(a)は、ディザリングは有効に固定されます。
構文(b)は、ディザリングを指定できます。

構文　　(a)　public void **registerLogo**(String *filePath*, int *regNum*) throws **PrinterException**

(b)　public void **registerLogo**(String *filePath*,
int *regNum*,
Dithering *dithering*) throws **PrinterException**

パラメーター　*filePath*
イメージデータのファイルパス
入力可能な形式について下記に説明します。

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

・Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
・file://
・content://
本パラメーターには"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。

regNum
ロゴ番号
有効範囲は、0～99です。

dithering
ディザリング
利用可能な設定は、『**4.4.3**(3) **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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明 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。

カラーのイメージデータは、二値化処理によりモノクロイメージに変換します。

登録可能なファイルサイズは、最大1MB(1048576バイト)です。

登録可能な画像サイズは、最大幅8192ドット、最大高さ2304ドットです。

本メソッドで登録したロゴの登録状況は、**getPrinterInformation**または**getPrinterInformationString**で、*prnInfo*に**GET_NV_MEM_KEYCODE_LIST**を指定して実行することで確認できます。

既にロゴが登録されているロゴ番号を指定した場合、ロゴを上書きします。

本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

deleteLogo プリンタ上のロゴの削除

プリンタに登録されているロゴを削除します。

構文 public void **deleteLogo**(int *regNum*) throws **PrinterException**

パラメーター *regNum*
ロゴ番号
有効範囲は、0～99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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　本メソッドで削除したロゴの削除状況は、**getPrinterInformation**または**getPrinterInformationString**で、*prnInfo*に**GET_NV_MEM_KEYCODE_LIST**を指定して実行することで確認できます。

getPrinterInformation
getPrinterInformationNumber
getPrinterInformationString　プリンタ情報の取得

プリンタ情報を取得します。

構文　　(a)　public byte[] **getPrinterInformation**(PrinterInformation *prnInfo*)
throws **PrinterException**

(b)　public int **getPrinterInformationNumber**(PrinterInformation *prnInfo*)
throws **PrinterException**

(c)　public String **getPrinterInformationString**(PrinterInformation *prnInfo*)
throws **PrinterException**

パラメーター　*prnInfo*
取得するプリンタ情報の種類
利用可能な設定、および取得できるプリンタ情報の一覧は、『**4.4.3**(4) **PrinterInformation**』を参照してください。
取得できるプリンタ情報の詳細については、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

戻り値　　プリンタ情報

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　*prnInfo*の指定に基づき、プリンタ情報を応答させるプリンタコマンドをプリンタへ送信し、プリンタからの応答データを応答識別子に基づき解析し、数値配列、数値、または文字列で返します。

全ての**PrinterInformation**において構文(a)を用いて数値配列でプリンタ情報を取得できます。

次の**PrinterInformation**において構文(b)を用いて数値でプリンタ情報を取得できます。

- **GET_NV_MEM_CAP**
- **GET_NV_MEM_REM_CAP**
- **GET_REM_USER_MEM_CAP**
- **GET_REM_USER_MEM_CAP_DEFRAG**
- **GET_PRN_ID_MODEL**
- **GET_PRN_ID_TYPE**
- **GET_PRN_ID_ROM_VER**
- **GET_PRN_ID_FIRM_CHECKSUM_BOOT**
- **GET_PRN_ID_FIRM_CHECKSUM_MAIN**
- **GET_PRN_ID_FIRM_CHECKSUM**
- **GET_MAINT_NUM_FEED_LINE**
- **GET_MAINT_NUM_HEAD_ACTIVE**
- **GET_MAINT_DRIVE_TIME**
- **GET_MAINT_NUM_FEED_LINE_INTEGRATION**
- **GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION**
- **GET_MAINT_DRIVE_TIME_INTEGRATION**
- **GET_HFONT_24_CHECKSUM**
- **GET_HFONT_16_CHECKSUM**
- **GET_FFONT_CHECKSUM**
- **GET_FFONT_DATA_SIZE**

次の**PrinterInformation**において構文(c)を用いて文字列でプリンタ情報を取得できます。

- **GET_NV_MEM_KEYCODE_LIST**
- **GET_PRN_ID_FIRM_VER_MAIN**
- **GET_PRN_ID_MFR**
- **GET_PRN_ID_MODEL_NAME**
- **GET_PRN_ID_FIRM_VER_BOOT**
- **GET_HFONT_24_ID**
- **GET_HFONT_24_INT_CHAR**
- **GET_HFONT_16_ID**
- **GET_HFONT_16_INT_CHAR**
- **GET_FFONT_LANG**
- **GET_FFONT_STANDARD**
- **GET_FFONT_COMPANY**

本メソッドの受信タイムアウト時間は、**setResponseTimeout**で設定した値になります。

本メソッドは**reset**により中断されます。

getVersion　SDKバージョンの取得

SDKのバージョンを文字列で取得します。

構文　public String **getVersion**()

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

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

controlTransaction　印字データー括送信の開始・終了

一括処理対象メソッドの印字データをバッファリングし、バッファ内のデータをプリンタへ送信します。

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

パラメーター　*control*

一括処理の動作選択

利用可能な設定は、『**4.4.3**(5) **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**PrinterException**をスローします。

詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　一括処理対象メソッドは下記の通りです。

- **write**
- **selectStandardMode**
- **selectPageMode**
- **printPageModeData**
- **setPageModeVerticalPosition**
- **setStandardModeArea**
- **setStandardModeAlignment**
- **setHorizontalPosition**
- **setStandardModeBarcodeDirection**
- **setLineSpacing**
- **setCharacterRightSpace**
- **setCharacterFormatting**
- **printText**
- **printLogo**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printBarcode**
- **print2DCode**
- **printPageModeRectangle**
- **feedLine**
- **feedDotLine**
- **feedCutPosition**
- **feedMarkPosition**

*control*により、一括処理対象メソッドの印字データのバッファリングの開始、終了を指定します。

・バッファリングの開始

本メソッドの*control*を**TRANSACTION_START**で実行すると、一括処理対象メソッドの印字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印字データは、プリンタへ送信されずバッファに溜め込まれます。バッファリング中に実行した一括処理対象外のメソッドは、即時に実行されます。

・バッファリングの終了
バッファリング中に本メソッドの*control*を**TRANSACTION_PRINT**で実行すると、一括処理対象メソッドの印字データのバッファリングを終了し、バッファ内のデータをプリンタへ送信します。
バッファリング中に本メソッドの*control*を**TRANSACTION_CLEAR**で実行すると、バッファリングを中止し、バッファ内のデータを破棄します。

バッファリングを開始せずに本メソッドの*control*を**TRANSACTION_PRINT**、**TRANSACTION_CLEAR**で実行した場合、エラーとなります。

バッファリング中に本メソッドの*control*を**TRANSACTION_START**で実行すると、バッファ内のデータを破棄し、バッファリングは継続します。

バッファ内にデータがない状態で、本メソッドの*control*を**TRANSACTION_PRINT**で実行すると、プリンタへの送信は行わず、バッファリングを終了します。

本メソッドの*control*を**TRANSACTION_PRINT**で実行中に、別スレッドから本メソッドの*control*を**TRANSACTION_START**で実行しても、本メソッドの*control*を**TRANSACTION_PRINT**で実行した送信は継続され、別スレッドから本メソッドの*control*を**TRANSACTION_START**で実行して新たにバッファリングを開始します。

バッファの最大サイズはシステムに依存します。データが最大サイズを超える場合、一括処理対象メソッドの実行がエラーとなり、バッファのデータは保持されます。

バッファリングされたデータを**TRANSACTION_PRINT**で送信している場合の送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

TRANSACTION_PRINTでデータの送信中にエラーとなった場合、未送信のデータは破棄されます。

TRANSACTION_PRINTでデータの送信中に**reset**が実行された場合、送信は中断され、未送信のデータは破棄されます。

selectStandardMode　スタンダードモードの開始

スタンダードモードを開始します。

構文　public void **selectStandardMode**() 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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　**open**が実行された直後はスタンダードモードです。

selectPageModeによりページモードが開始されていた場合、ページモードを終了し、プリンタのページモードデータを破棄します。

selectPageMode　ページモードの開始

ページモードを開始し、印字領域と印字方向を指定します。

構文　public void **selectPageMode**(int *startX*,
int *startY*,
int *width*,
int *height*,
Direction *direction*) throws **PrinterException**

パラメーター　*startX*
横方向始点(ドット)
有効範囲は、0～2399です。

startY
縦方向始点(ドット)
有効範囲は、0～2399です。

width
印字領域幅(ドット)
有効範囲は、1～2400です。

height
印字領域高さ(ドット)
有効範囲は、1～2400です。

direction
印字方向
利用可能な設定は、『**4.4.3**(6) **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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　*startX*が印字幅を超える場合、本メソッドによる印字領域の設定は無効となります。
(*startX* + *width*)が印字幅を超える場合、*startX*は指定した値に設定され、*width*は
(印字幅 － *startX*)に設定されます。
(*startY* + *height*)が2400以上の場合、*startY*は指定した値に設定され、*height*は
(2400 － *startY*)に設定されます。
印字幅は機能設定により指定が可能です。印字幅と機能設定の詳細は『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

指定した印字領域内において、**setHorizontalPosition**、及び**setPageModeVerticalPosition**によりキャラクタ、イメージ、バーコード、2次元バーコードを任意の印字位置に配置し、ページモードデータを構成してください。

selectStandardModeによりページモードが中断された場合は、ページモードのデータは破棄されます。

プリンタがリセットされた場合、スタンダードモードに戻ります。

printPageModeData　ページモードデータの印字

ページモードデータを印字します。

構文　public void **printPageModeData**() 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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　本メソッドは、**selectPageMode**によりページモードを開始し、印字領域内にページモードデータを構成した後、実行してください。

本メソッドは、スタンダードモードでは無視されます。

setPageModeVerticalPosition　ページモードの縦方向絶対位置指定

ページモードの印字領域内で、印字データの縦方向の絶対位置を指定します。

構文　public void **setPageModeVerticalPosition**(int *verticalPosition*) throws **PrinterException**

パラメーター　*verticalPosition*
縦方向のデータ展開開始位置(ドット)
有効範囲は、0～2399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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　始点は、**selectPageMode**の*direction*(印字方向)設定により変わります。印字方向と始点については、『**4.4.3**(6) **Direction**』を参照してください。

本メソッドの設定は、スタンダードモードでは無効です。

印字領域を超える指定は無視されます。

プリンタがリセットされた場合、本メソッドの設定は無効になります。

setStandardModeArea　スタンダードモードの印字領域設定

スタンダードモードの印字領域を設定します。

構文　public void **setStandardModeArea**(int *leftMargin*, int *prnAreaWidth*) throws **PrinterException**

パラメーター　*leftMargin*
左マージン(ドット)
有効範囲は、0～831です。

prnAreaWidth
印字領域幅(ドット)
有効範囲は、1～831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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　左マージンと印字領域幅の関係を図 4-1に示します。印字データは、網掛けで示す印字領域に展開されます。

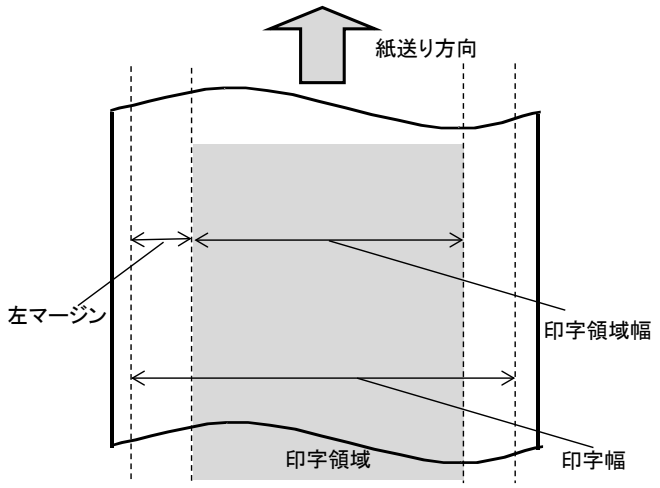


図 4-1

本メソッドの設定は、ページモードでは無効です。本メソッドをページモードで実行した場合、設定はスタンダードモード開始後の印字に反映されます。

プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

*leftMargin*に印字幅を超える値を指定した場合、印字幅が設定されます。*prnAreaWidth*に印字幅を超える値を指定した場合、印字幅と左マージンの差が設定されます。
印字幅は機能設定により指定が可能です。印字幅と機能設定の詳細は『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

プリンタがリセットされた場合、*leftMargin*は0、*prnAreaWidth*は印字幅に初期化されます。

setStandardModeAlignment　位置揃え

スタンダードモードの位置揃えを設定します。

構文　　public void **setStandardModeAlignment**(Alignment *align*) throws **PrinterException**

パラメーター　*align*

位置揃え
利用可能な設定は、『**4.4.3**(7) **Alignment**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　**setStandardModeArea**で印字領域を設定している場合、設定している印字領域内で位置を揃えます。

本メソッドの設定は、ページモードでは無効です。本メソッドをページモードで実行した場合、設定はスタンダードモード開始後の印字に反映されます。

プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

プリンタがリセットされた場合、本メソッドの設定はデフォルト値に戻ります。

setHorizontalPosition　絶対位置指定

横方向の絶対位置を指定します。

構文　public void **setHorizontalPosition**(int *horizontalPosition*) throws **PrinterException**

パラメーター　*horizontalPosition*
横方向の印字開始位置(ドット)
有効範囲は、0～2399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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードの場合、**setStandardModeArea**で設定した左マージン位置を基準とします。

ページモードの場合、**selectPageMode**の*direction*により設定する印字方向により始点が変わります。印字方向と始点については、『**4.4.3**(6) **Direction**』を参照してください。

setStandardModeArea、**selectPageMode**で設定した印字領域を超える指定は無視されます。

プリンタがリセットされた場合、本メソッドの設定は無効になります。

setStandardModeBarcodeDirection　バーコード、2次元バーコードの印字方向の選択

スタンダードモードでのバーコード、2次元バーコードの印字方向を設定します。

構文　public void **setStandardModeBarcodeDirection**(Rotate *rotate*) throws **PrinterException**

パラメーター　*rotate*
印字方向
利用可能な設定は、『**4.4.3**(15) **Rotate**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　**printBarcode**、または**print2DCode**を実行すると、本メソッドの設定は、デフォルト値に戻ります。

本メソッドの設定は、ページモードでは無効です。本メソッドをページモードで実行した場合、設定はスタンダードモード開始後の印字に反映されます。

*rotate*で**ROTATE_90_TO_RIGHT**、**ROTATE_90_TO_LEFT**を指定した場合、印字可能なバーコード、2次元バーコードの幅は最大300mmです。300mmを超える場合、バーコード、2次元バーコードは印字されません。

プリンタがリセットされた場合、本メソッドの設定はデフォルト値に戻ります。

setLineSpacing　改行量の設定

改行量を設定します。

構文　public void **setLineSpacing**(int *lineSpacing*) throws **PrinterException**

パラメーター　*lineSpacing*
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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードとページモードにそれぞれ独立した改行量の設定が可能です。

プリンタがリセットされた場合、*lineSpacing*は34に設定されます。

setCharacterRightSpace　文字の右スペース量の設定

文字の右スペース量を設定します。

構文　public void **setCharacterRightSpace**(int *space*) throws **PrinterException**

パラメーター　*space*
右スペース量(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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードとページモードにそれぞれ独立した右スペース量の設定が可能です。

半角文字及び全角文字に有効です。全角文字の場合、左スペースは0に設定されます。

プリンタがリセットされた場合、*space*は0に設定されます。

selectCharacterSet　文字セット選択

文字セットを選択します。

構文　public void **selectCharacterSet**(CharacterSet *charSet*) throws **PrinterException**

パラメーター　*charSet*
文字セット
利用可能な選択は、『**4.4.3**(8) **CharacterSet**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　文字セットのデフォルト値は、Androidデバイスの言語設定により下記の通りです。

日本語の場合	**CODEPAGE_KATAKANA**
日本語以外の場合	**CODEPAGE_1252**

本メソッドを実行した後、**reset**によるリセットまたはプリンタが電源オフされた場合も、本メソッドの設定は有効です。

getCharacterSet　設定されている文字セット取得

設定されている文字セットを取得します。

構文　　public CharacterSet **getCharacterSet**()

戻り値　　文字セット

説明　　取得可能な設定は、『**4.4.3**(8) **CharacterSet**』を参照してください。

selectInternationalCharacterSet　国際文字セット選択

国際文字セットを選択します。

構文　　public void **selectInternationalCharacterSet**(InternationalCharacterSet *intCharSet*)
throws **PrinterException**

パラメーター　*intCharSet*
国際文字セット
利用可能な選択は、『**4.4.3**(9) **InternationalCharacterSet**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　国際文字セットのデフォルト値は、Androidデバイスの言語設定により下記の通りです。

日本語の場合	**INT_CHAR_SET_JAPAN**
日本語以外の場合	**INT_CHAR_SET_USA**

本メソッドを実行した後、**reset**によるリセットまたはプリンタが電源オフされた場合も、本メソッドの設定は有効です。

getInternationalCharacter　設定されている国際文字セット取得

設定されている国際文字セットを取得します。

構文　　public InternationalCharacterSet **getInternationalCharacter**()

戻り値　　国際文字セット

説明　　取得可能な設定は、『**4.4.3**(9) **InternationalCharacterSet**』を参照してください。

setCharacterFormatting　　文字修飾

文字修飾を設定します。

構文　　public void **setCharacterFormatting**(CharacterType *type*,
CharacterScale *verticalScale*,
CharacterScale *horizontalScale*,
Underline *underline*,
Bold *bold*,
Reverse *reverse*,
Rotate *rotate*) throws **PrinterException**

パラメーター　*type*
文字フォント
利用可能な設定は、『**4.4.3**(10) **CharacterType**』を参照してください。

verticalScale
文字サイズ（縦方向）
利用可能な設定は、『**4.4.3**(11) **CharacterScale**』を参照してください。

horizontalScale
文字サイズ（横方向）
利用可能な設定は、『**4.4.3**(11) **CharacterScale**』を参照してください。

underline
アンダーライン
利用可能な設定は、『**4.4.3**(12) **Underline**』を参照してください。

bold
強調印字
利用可能な設定は、『**4.4.3**(13) **Bold**』を参照してください。

reverse
白黒反転印字
利用可能な設定は、『**4.4.3**(14) **Reverse**』を参照してください。

rotate
回転印字
ページモードでは、設定は無効です。設定した場合、スタンダードモード開始後の印字に反映されます。
スタンダードモードの場合、プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。
スタンダードモードで**ROTATE_90_TO_RIGHT**、**ROTATE_90_TO_LEFT**を指定した場合、下記の動作となります。
・アンダーライン印字は行われません。
・*verticalScale*と*horizontalScale*の縦横の倍率は逆になります。
利用可能な設定は、『**4.4.3**(15) **Rotate**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　プリンタがリセットされた場合、本メソッドの設定はデフォルト値に戻ります。

printText　テキストデータ送信

テキストデータを送信します。

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

パラメーター　*text*
プリンタに送信するテキストデータ

例外　　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　本メソッドは、*text*で指定したテキストデータを、**selectCharacterSet**で指定した文字セット設定を基に、プリンタが認識可能なテキストデータへエンコードを行った後、プリンタに送信します。

selectCharacterSet、**selectInternationalCharacterSet**の指定に基づき、送信データにヘッダとしてプリンタコマンド「文字コードテーブルの選択」「国際文字の選択」「漢字コード体系の選択」「漢字モード解除」を付加します。プリンタコマンドの詳細は、『MP-A40シリーズ　サーマルプリンタ　技術説明書』を参照してください。

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。

指定可能なデータサイズは、最大16KB(16384バイト)です。

controlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

本メソッドは**reset**により中断されます。

printLogo　プリンタに登録したロゴの印字

プリンタに登録したロゴを印字します。

構文　　public void **printLogo**(int *regNum*) throws **PrinterException**

パラメーター　*regNum*
ロゴ番号
有効範囲は、0～99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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　登録は**registerLogo**により行います。

指定された*regNum*にロゴが登録されていない場合、無視されます。

sendDataFile　ファイルデータ送信

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

構文　(a)　public void **sendDataFile**(String *filePath*) throws **PrinterException**

(b)　public void **sendDataFile**(String *filePath,*
Dithering *dithering*) throws **PrinterException**

パラメーター　*filePath*
プリンタに送信するファイルのパス
入力可能な形式について下記に説明します。

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

・Androidに用意されたクラス「android.net.Uri」が扱う下記スキーム名のURI文字列
・file://
・content://
本パラメーターには"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。

dithering
ディザリング
*filePath*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。
利用可能な設定は、『**4.4.3**(3) **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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　指定したファイルの拡張子に応じて、下記の処理を行います。
・ ファイル拡張子が.bmp、.jpg、.jpeg、.pngの場合
イメージデータを、プリンタで印字可能な形式に変換してプリンタに送信します。
カラーのイメージデータは、二値化処理によりモノクロイメージに変換します。

・ ファイル拡張子が.txtの場合
テキストデータのフォーマットはUTF-8をサポートします。
テキストデータを、**selectCharacterSet**で指定した文字セット設定を基に、プリンタが認識可能なテキストデータへエンコードを行った後、プリンタに送信します。
selectCharacterSet、**selectInternationalCharacterSet**の指定に基づき、送信データにヘッダとしてプリンタコマンド「文字コードテーブルの選択」「国際文字の選択」「漢字コード体系の選択」「漢字モード解除」を付加します。プリンタコマンドの詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。
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。

・ ファイル拡張子が.bin、.datの場合
無変換のままプリンタに送信します。

・ ファイル拡張子が.htm、.htmlの場合
無変換のままプリンタに送信します。
ファイルデータの終わりは、HTML終了タグを必ず指定してください。

指定可能なファイルサイズは、最大16MB(16777216バイト)です。

controlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

送信データには、プリンタコマンド「プリンタの初期化」以外のプリンタを初期化するプリンタコマンドを含めないでください。ハードウェアリセットを行う場合、**reset**を使用してください。プリンタの初期化については、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

本メソッドは**reset**により中断されます。

sendImage　イメージデータ送信

プリンタにイメージデータを送信します。

構文　public void **sendImage**(Bitmap *bitmap*
Dithering *dithering*) throws **PrinterException**

パラメーター　*bitmap*
プリンタに送信する「android.graphics.Bitmap」型オブジェクト
イメージファイルとしてプリンタに送信します。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。

dithering
ディザリング
利用可能な設定は、『**4.4.3**(3) **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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　**controlTransaction**を使用しない場合、本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

本メソッドは**reset**により中断されます。

選択したPDFファイルの指定したページを印字します。

構文 public void **printPDF**(String *filePath*,
int *startIndex*,
int *endIndex*,
Rotate *rotate*,
int *width*,
int *bottomMargin*,
Dithering *dithering*) throws **PrinterException**

パラメーター *filePath*
PDFのファイルのパス

startIndex
印字するページの開始番号
有効範囲は、-1、及び1～PDFファイルのページ数です。

endIndex
印字するページの終了番号
有効範囲は、1～2147483647です。

rotate
イメージの回転方向
利用可能な設定は、『**4.4.3**(15) **Rotate**』を参照してください。

width
イメージの幅(ドット)
有効範囲は、-1、及び1～832です。

bottomMargin
下余白(ドット)
有効範囲は、-1、及び0～2400です。

dithering
ディザリング
利用可能な設定は、『**4.4.3**(3) **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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

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

サポートするPDFのファイル拡張子は、.pdfです。
指定可能なファイルサイズは、最大16MB(16777216バイト)です。
カラーのイメージデータは、二値化処理によりモノクロイメージに変換します。

*startIndex*に-1を指定した場合、全てのページを印字します。
*startIndex*に-1を指定した場合、*endIndex*の値は無視されます。
*startIndex*に1以上を指定した場合、*startIndex*に指定したページ番号から*endIndex*に指定したページ番号のページまでを印字します。
指定するページ番号に関わらず、PDFは一度に100枚まで印字できます。
*startIndex*に*endIndex*で指定した値より大きい値を指定した場合はエラーとなります。
*endIndex*にPDFファイルページ数より大きい値を指定した場合、*startIndex*に指定したページ番号からPDFファイルの最後のページまでを印字します。
*startIndex*及び*endIndex*に範囲外の値を指定した場合、エラーとなります。
*width*に-1を指定した場合、サイズ変換は行いません。
*width*に1～832の値を指定した場合、アスペクト比を維持しながら画像の幅が指定された値に変換します。
*bottomMargin*に-1を指定した場合、下余白を維持してイメージを作成し印字します。
*bottomMargin*に0～2400の値を指定した場合、下余白を指定したサイズに変更します。
*bottomMargin*に0～2400の値を指定した場合、空白ページの印字は行いません。
マーク紙が選択されている場合、PDF印字後マーク紙の頭出しを行います。
マーク紙の使用方法の詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

controlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は1ページ印字毎に**setWriteTimeout**で設定した値になります。

本メソッドは**reset**により中断されます。

注意　・本メソッドはページモードではサポートしていません。
・一度に100枚より多く印字した場合の印字動作保証は行っておりません。

printBarcode　バーコードの印字

バーコードを印字します。

構文　(a) public void **printBarcode**(TypeBarcode *type*,
String *text*,
ModuleWidthBarcode *moduleWidth*,
int *moduleHeight*,
HriPositionBarcode *hri*,
NwRatioBarcode *nwRatio*) throws **PrinterException**

(b) public void **printBarcode**(TypeBarcode *type*,
byte[] *binary*,
ModuleWidthBarcode *moduleWidth*,
int *moduleHeight*,
HriPositionBarcode *hri*,
NwRatioBarcode *nwRatio*) throws **PrinterException**

パラメーター　*type*
バーコードの種類
利用可能な設定は、『**4.4.3**(16) **TypeBarcode**』を参照してください。

(a) *text*
バーコードのテキストデータ

(b) *binary*

バーコードのバイナリデータ

text、*binary*で利用可能な値は、下記を参照してください。

バーコードの種類	データサイズ	入力可能なバーコードデータ	
UPC-A	11 ～ 12バイト	'0' ～ '9'	(0x30 ～ 0x39)
UPC-E	11 ～ 12バイト	'0' ～ '9'	(0x30 ～ 0x39)
JAN13 (EAN13)	12 ～ 13バイト	'0' ～ '9'	(0x30 ～ 0x39)
JAN8 (EAN8)	7 ～ 8バイト	'0' ～ '9'	(0x30 ～ 0x39)
CODE39	1 ～ 150バイト	' '	(0x20)
		'$'	(0x24)
		'%'	(0x25)
		'+'	(0x2B)
		'–'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' ～ '9'	(0x30 ～ 0x39)
		'A' ～ 'Z'	(0x41 ～ 0x54)
ITF	2 ～ 150バイト (ただし偶数個)	'0' ～ '9'	(0x30 ～ 0x39)
CODABAR	1 ～ 150バイト	'$'	(0x24)
		'+'	(0x2B)
		'–'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' ～ '9'	(0x30 ～ 0x39)
		':'	(0x3A)
		'A' ～ 'D'	(0x41 ～ 0x44)
CODE93	1 ～ 150バイト		(0x00 ～ 0x7F)
CODE128	2 ～ 150バイト		(0x00 ～ 0x7F)*1
JAN13 (EAN13) add-on 2	14 ～ 15バイト	'0' ～ '9'	(0x30 ～ 0x39)
JAN13 (EAN13) add-on 5	17 ～ 18バイト	'0' ～ '9'	(0x30 ～ 0x39)
GS1 Databar Omni-directional	13バイト	'0' ～ '9'	(0x30 ～ 0x39)
GS1 Databar Truncated	13バイト	'0' ～ '9'	(0x30 ～ 0x39)
GS1 Databar Limited	13バイト	'0' ～ '9'	(0x30 ～ 0x39)
GS1 Databar Expanded	2 ～ 150バイト	' '～ '"'	(0x20 ～ 0x22)
		'%' ～ '?'	(0x25 ～ 0x3F)
		'A' ～ 'Z'	(0x41 ～ 0x5A)
		'_'	(0x5F)
		'a' ～ 'z'	(0x61 ～ 0x7A)
		'{'	(0x7B)

*1： 指定したコードセットにより、入力可能なバーコードデータが異なります。
CODE A：0x00 ～ 0x50
CODE B：0x00 ～ 0x7F

CODE C : 00 ～ 99(0x00 ～ 0x63)の2桁の数字

moduleWidth

バーコードのモジュール幅または細エレメント

利用可能な設定は、『**4.4.3**(17) **ModuleWidthBarcode**』を参照してください。

moduleHeight

バーコードのモジュール高さ(ドット)

利用可能な設定は、下記を参照してください。

バーコード種類	*moduleHeight*（ドット）
UPC-A	0:初期値 (162 ドット)、1 ～ 255
UPC-E	
JAN13 (EAN13)	
JAN8 (EAN8)	
CODE39	
ITF	
CODABAR	
CODE93	
CODE128	
JAN13 (EAN13) add-on 2	
JAN13 (EAN13) add-on 5	
GS1 Databar Omni-directional	0:初期値 (162 ドット)、66 ～ 255
GS1 Databar Truncated	0:初期値 (162 ドット)、26 ～ 255
GS1 Databar Limited	0:初期値 (162 ドット)、20 ～ 255
GS1 Databar Expanded	0:初期値 (162 ドット)、68 ～ 255

hri

バーコードのHRI文字設定

利用可能な設定は、『**4.4.3**(18) **HriPositionBarcode**』を参照してください。

nwRatio

バーコードのNW比

利用可能な設定は、*type*の設定により異なります。

詳細は、『**4.4.3**(19) **NwRatioBarcode**』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**PrinterException**をスローします。

詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードの場合、プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

印字領域を超えた場合は、バーコードは印字されません。

本メソッドは、構文(a)の*text*で指定したテキストデータを、プリンタが認識可能なテキストデータにエンコードしてプリンタに送信します。
構文(b)の*binary*で指定したバイナリデータは、そのままプリンタに送信します。

CODE128で指定する特殊コードを下記に示します。

入力データ		コードセット毎の特殊コード		
		CODE A	CODE B	CODE C
'{S'	(0x7B, 0x53)	SHIFT	SHIFT	SHIFT
'{A'	(0x7B, 0x41)	−	CODE A	CODE A
'{B'	(0x7B, 0x42)	CODE B	−	CODE B
'{C'	(0x7B, 0x43)	CODE C	CODE C	−
'{1'	(0x7B, 0x31)	FNC1	FNC1	FNC1
'{2'	(0x7B, 0x32)	FNC2	FNC2	FNC2
'{3'	(0x7B, 0x33)	FNC3	FNC3	FNC3
'{4'	(0x7B, 0x34)	FNC4	FNC4	FNC4
'{{'	(0x7B, 0x7B)	'{'	'{'	−

下記のバーコードのチェックデジットは自動計算されます。
- UPC-A
- UPC-E
- JAN13 (EAN13)
- JAN8 (EAN8)
- JAN13 (EAN13) add-on 2
- JAN13 (EAN13) add-on 5
- GS1 Databar Omni-directional
- GS1 Databar Truncated
- GS1 Databar Limited

下記のバーコードでチェックデジットを含むデータを指定した場合、チェックデジットデータは無視され、プリンタ側で再計算します。
- UPC-A
- UPC-E
- JAN13 (EAN13)
- JAN8 (EAN8)

CODE39のスタートコード及びストップコード('*')は自動で付加されますので指定する必要はありません。

ITFのバーコードデータに奇数個のデータを指定した場合、最後のデータが切り捨てられます。

CODABARのバーコードデータにおいて、'A'～'D'はスタートコード及びストップコードとして入力してください。

GS1 Databar Expandedのバーコードデータにおいて、'{'は、FNC1を指定する場合にのみ使用してください。FNC1を指定する場合、'{1'(0x7B, 0x31)を入力します。

text、*binary*、*moduleWidth*、*moduleHeight*、*nwRatio*のいずれか一つでも、*type*で指定したバーコード種類に対応する値以外を設定した場合、エラーになります。

controlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

2次元バーコードを印字します。

構文 (a) public void **print2DCode**(Type2DCode *type*,
String *text*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
int *moduleHeight*,
int *column*,
int *row*,
ErrorCorrect2Dcode *errorCorrect*) throws **PrinterException**

(b) public void **print2DCode**(Type2DCode *type*,
byte[] *binary*,
Mode2Dcode *mode*,
ModuleSize2Dcode *moduleSize*,
int *moduleHeight*,
int *column*,
int *row*,
ErrorCorrect2Dcode *errorCorrect*) throws **PrinterException**

パラメーター *type*
2次元バーコードの種類
利用可能な設定は、『**4.4.3**(20) **Type2DCode**』を参照してください。

(a) *text*
バーコードのテキストデータ

(b) *binary*
バーコードのバイナリデータ

mode
2次元バーコードのモード
利用可能な設定は、*type*の設定により異なります。
詳細は、『**4.4.3**(21) **Mode2Dcode**』を参照してください。
MODE_2DCODE_MAXI_CODE_2を選択する場合、*text*、または*binary*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)のデータをそれぞれ付加してください。
MODE_2DCODE_MAXI_CODE_3を選択する場合、*text*、または*binary*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)のデータをそれぞれ付加してください。

moduleSize
2次元バーコードのモジュールサイズ
利用可能な設定は、*type*の設定により異なります。
詳細は、『**4.4.3**(22) **ModuleSize2Dcode**』を参照してください。

moduleHeight

2次元バーコードのモジュール高さ(ドット)

PDF417、及びGS1 Databar Stacked Omni-directionalで有効です。

・ PDF417

PDF段高さを指定します。有効範囲は、2～127です。

・ GS1 Databar Stacked Omni-directional

1段の高さを指定します。有効範囲は、33～255です。

column

2次元バーコードのカラム数

PDF417、及びGS1 Databar Expanded Stackedで有効です。

・ PDF417

データ領域のカラム数を指定します。有効範囲は、0～30です。

0を指定した場合、カラム数は自動設定されます。

・ GS1 Databar Expanded Stacked

1行のエレメント数を指定します。有効範囲は、2～20です。

row

2次元バーコードの段数

PDF417のみで有効です。PDF417以外の2次元バーコードでは無視されます。

有効範囲は、0、及び3～90です。

0を指定した場合、段数は自動設定されます。

errorCorrect

2次元バーコードのエラー訂正レベル

利用可能な設定は、*type*の設定により異なります。

詳細は、『**4.4.3**(23) **ErrorCorrect2Dcode**』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**PrinterException**をスローします。

詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードの場合、プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

本メソッドは、構文(a)の*text*で指定したテキストデータを、プリンタが認識可能なテキストデータにエンコードしてプリンタに送信します。
*text*には、*type*で指定する2次元バーコード種類の規格に従い、対応する文字データをASCII形式で入力してください。
構文(b)の*binary*で指定したバイナリデータは、そのままプリンタに送信します。
*binary*には、*type*で指定する2次元バーコード種類の規格に従い、対応する文字データをASCII形式の文字コードで入力してください。

text、*binary*、*mode*、*moduleSize*、*moduleHeight*、*column*、*errorCorrect*のいずれか一つでも、*type*で指定した2次元バーコード種類に対応する値以外を設定した場合、エラーになります。

2次元バーコードの幅が、印字領域幅を超える場合、本メソッドは無視されます。

controlTransactionを使用しない場合、本メソッドの送信タイムアウト時間は、**setWriteTimeout**で設定した値になります。

printPageModeRectangle　　ページモードの矩形の描画

ページモードで矩形を描画します。

構文　　public void **printPageModeRectangle**(int *startX*,
　　int *startY*,
　　int *width*,
　　int *height*,
　　int *thickness*) throws **PrinterException**

パラメーター　*startX*
　横方向始点(ドット)
　有効範囲は、0～2395です。

startY
　縦方向始点(ドット)
　有効範囲は、0～2395です。

width
　幅(ドット)
　有効範囲は、4～2400です。

height
　高さ(ドット)
　有効範囲は、4～2400です。

thickness
　線幅(ドット)
　有効範囲は、2～40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**PrinterException**をスローします。
　詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　　本メソッドは、スタンダードモードでは実行しないでください。

始点は、**selectPageMode**の*direction*により設定するページモードの印字方向により変わります。印字方向と始点については、『**4.4.3**(6) **Direction**』を参照してください。

startX、*startY*、*width*、*height*に、**selectPageMode**により設定するページモードの印字領域を超える指定をした場合、印字領域を超える部分は描画されません。

本メソッドではプリンタコマンド「罫線OFF」、「罫線クリア」を送信し、罫線をOFFにし罫線バッファをクリアします。プリンタコマンドの詳細は、『MP-A40シリーズ　サーマルプリンタ 技術説明書』を参照してください。

feedLine 行単位の紙送り

行単位で紙送りをします。

構文　public void **feedLine**(int *lines*) throws **PrinterException**

パラメーター　*lines*
紙送りする行数(行)
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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードでは、指定した行数の紙送りを行います。
プリンタの行バッファ内にデータがある場合には、一行印字した後に紙送りを行います。
ページモードでは、縦方向のデータ展開開始位置を指定した行数分、移動します。

一行の紙送り量(改行量)は、**setLineSpacing**で設定が可能です。

feedDotLine ドット単位の紙送り

ドット単位で紙送りをします。

構文　public void **feedDotLine**(int *dotLines*) throws **PrinterException**

パラメーター　*dotLines*
紙送りするドット数(ドット)
スタンダードモードでは有効範囲は、-48～8192です。
ページモードでは、*dotLines*に0～8192を指定した場合のみ有効です。
-48～-1を指定した場合は無視されます。

例外　**PrinterException**
本メソッド呼び出し時にエラーが発生した場合**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードでは、指定したドット数の紙送りを行います。
プリンタの行バッファ内にデータがある場合には、一行印字した後に紙送りを行います。
ページモードでは、縦方向のデータ展開開始位置を指定したドット数分、移動します。

feedCutPosition 用紙カット位置までの移動

カット位置まで紙送りをします。

構文　public void **feedCutPosition**() 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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードの場合、プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

feedMarkPosition　マーク紙の頭出し

マーク紙の頭出し、及び頭出し位置の補正を行います。

構文　public void **feedMarkPosition**(int *dotLines*) throws **PrinterException**

パラメーター　*dotLines*
補正量(ドット)
有効範囲は、-48～255です。
補正を行わない場合、0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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　スタンダードモードの場合、プリンタに印字データが残っていると、本メソッドは実行されません。印字を全て終わらせてから本メソッドを実行してください。

マーク紙が選択されている場合のみ有効です。
マーク紙の使用方法の詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

setLog　ログ出力設定

ログ出力を設定します。

構文　public void **setLog**(int *logLevel*, LogFileSize *logFileSize*) throws **PrinterException**

パラメーター　*logLevel*
ログの出力レベル
0を指定してください。0以外の値を指定して出力されたログの内容は保証いたしません。

logFileSize
ログファイルの最大サイズ
利用可能な設定は、『**4.4.3**(24) **LogFileSize**』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**PrinterException**をスローします。
詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

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

ログファイル名 : PrinterManager.log.x (xの範囲は、0～4です)

最初にログファイルはPrinterManager.log.0で作成されます。
ログファイルの最大サイズを超えた場合、ファイル名をPrinterManager.log.1に変更し、新たにPrinterManager.log.0を作成します。
作成されるログファイル数は最大5つです。

(2) **PrinterStatus**

- **概要**

プリンタステータスを格納するクラスです。
下記の機能を提供します。

Public Methods

名前	説明
getErrOffline	オフラインエラー状態の取得
getErrHardware	ハードウェアエラー状態の取得
getErrVoltage	Vp電圧エラー状態の取得
getErrHeadTemperature	ヘッド温度エラー状態の取得
getErrOutOfPaper	紙無しエラー状態の取得
getErrMarkPaperJam	マーク検出時紙ジャムエラー状態の取得
getErrCoverOpen	ペーパーカバーオープンエラー状態の取得
getErrBattery	バッテリエラー状態の取得
getStateFeedSwitch	フィードスイッチ状態の取得
getStatePaperFeed	紙送り状態の取得
getStateReturnWaiting	復帰待ち状態の取得
getStateFlashMemoryRewriting	フラッシュメモリ書き換え状態の取得
getStateBattery	バッテリ電圧状態の取得

- **Public Methods**

getErrOffline　オフラインエラー状態の取得

オフラインエラー状態を取得します。

構文　public boolean **getErrOffline**()

戻り値　true :オフラインエラー
false :オンライン

getErrHardware　ハードウェアエラー状態の取得

ハードウェアエラー状態を取得します。

構文　public boolean **getErrHardware**()

戻り値　true :エラー
false :OK

getErrVoltage　Vp電圧エラー状態の取得

Vp電圧エラー状態を取得します。

構文　public boolean **getErrVoltage**()

戻り値　true ：エラー
false ：OK

getErrHeadTemperature　ヘッド温度エラー状態の取得

ヘッド温度エラー状態を取得します。

構文　public boolean **getErrHeadTemperature**()

戻り値　true ：エラー
false ：OK

getErrOutOfPaper　紙無しエラー状態の取得

紙無しエラー状態を取得します。

構文　public boolean **getErrOutOfPaper**()

戻り値　true ：エラー
false ：OK

getErrMarkPaperJam　マーク検出時紙ジャムエラー状態の取得

マーク検出時紙ジャムエラー状態を取得します。

構文　public boolean **getErrMarkPaperJam**()

戻り値　true ：エラー
false ：OK

getErrCoverOpen　ペーパーカバーオープンエラー状態の取得

ペーパーカバーオープンエラー状態を取得します。

構文　public boolean **getErrCoverOpen**()

戻り値　true ：エラー
false ：OK

getErrBattery　バッテリエラー状態の取得

バッテリエラー状態を取得します。

構文　public boolean **getErrBattery**()

戻り値　true ：エラー
false：OK

getStateFeedSwitch　フィードスイッチ状態の取得

フィードスイッチ状態を取得します。

構文　public boolean **getStateFeedSwitch**()

戻り値　true ：ON
false：OFF

getStatePaperFeed　紙送り状態の取得

紙送り状態を取得します。

構文　public boolean **getStatePaperFeed**()

戻り値　true ：紙送り中
false：停止

getStateReturnWaiting　復帰待ち状態の取得

復帰待ち状態を取得します。

構文　public boolean **getStateReturnWaiting**()

戻り値　true ：復帰待ち状態
false：復帰待ち状態でない

getStateFlashMemoryRewriting　フラッシュメモリ書き換え状態の取得

フラッシュメモリ書き換え状態を取得します。

構文　public boolean **getStateFlashMemoryRewriting**()

戻り値　true ：フラッシュメモリ書き換え中
false：フラッシュメモリ書き換え中でない

getStateBattery　バッテリ電圧状態の取得

バッテリ電圧状態を取得します。

構文　public int **getStateBattery**()

戻り値　0:バッテリなし
1:バッテリ残量レベル4またはバッテリ残量レベル5 （要充電、または残量0%）
2:バッテリ残量レベル3　（残量目安： 約10%）
3:バッテリ残量レベル2　（残量目安： 約40%）
4:バッテリ残量レベル1　（残量目安： 約80%）

(3) **PrinterDiscovery**

- **概要**

プリンタの探索機能を提供するクラスです。
下記の機能を提供します。

Public Methods

名前	説明
PrinterDiscovery	コンストラクタ
startDiscoveryPrinter	プリンタの探索開始
cancelDiscoveryPrinter	プリンタの探索中断
getFoundPrinter	発見されたプリンタ情報リストの取得

- **Public Methods**

PrinterDiscovery　コンストラクタ

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

構文　public **PrinterDiscovery** (Context *context*)

パラメーター　*context*
本メソッドを呼び出すAndroidアプリケーションのコンテキスト。
getApplicationContext()で取得できる*context*を指定してください。

startDiscoveryPrinter　プリンタの探索開始

プリンタの探索を開始します。

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

パラメーター　*listener*
PrinterListenerのインスタンス

prnIf
通信インターフェース定数
利用可能な設定は、『**4.4.3**(1) **PrinterInterface**』を参照してください。
PRN_IF_USBはサポートされていません。

retry

リトライ回数(回)

*prnIf*の設定により、動作が異なります。

・**PRN_IF_TCP**の場合

*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として処理されます。

5を超える値を指定した場合は5として処理されます。

・**PRN_IF_BT**の場合

指定した値は無視され、探索は1回のみ行います。

timeout

探索1回のタイムアウト時間(秒)

*prnIf*の設定により、動作が異なります。

・**PRN_IF_TCP**の場合

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、プリンタからの応答を待ちます。

有効範囲は、3～60です。

3未満の値を指定した場合は3として処理されます。

60を超える値を指定した場合は60として処理されます。

・**PRN_IF_BT**の場合

指定した値は無視されます。

例外　**PrinterException**

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

詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　**cancelDiscoveryPrinter**で中断が可能です。探索を中断、または完了すると、*listener*で指定したインスタンスの**onDiscoveryFinished**を実行します。

探索結果は、**getFoundPrinter**で取得してください。

cancelDiscoveryPrinter　プリンタの探索中断

startDiscoveryPrinterで開始したプリンタの探索を中断します。

構文　public void **cancelDiscoveryPrinter**() 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**PrinterException**をスローします。

詳細は、『**4.4.4**(1) **PrinterException**』を参照してください。

説明　探索を中断すると、**startDiscoveryPrinter**の*listener*で指定したインスタンスの**onDiscoveryFinished**を実行します。

中断時までの探索結果は**getFoundPrinter**にて取得できます。

getFoundPrinter　発見されたプリンタ情報リストの取得

startDiscoveryPrinterで発見された全てのプリンタ情報を取得します。

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

戻り値　発見されたプリンタ情報を**DiscoveredPrinter**クラスのArrayListで返します。**DiscoveredPrinter**クラスのメソッドにより、プリンタモデル名、Bluetoothアドレス、IPアドレス、MACアドレス及びシリアルナンバーを取得できます。
DiscoveredPrinterクラスのメソッドは、『**4.4.2**(4) **DiscoveredPrinter**』を参照してください。

(4) **DiscoveredPrinter**

- **概要**

  startDiscoveryPrinter で発見されたプリンタ情報を格納するクラスです。発見されたプリンタ情報からプリンタモデル名、Bluetooth アドレス、IP アドレス、MAC アドレス及びシリアルナンバーを取得できます。
  下記の機能を提供します。

  Public Methods

名前	説明
getPrinterModel	プリンタモデル名の取得
getBluetoothAddress	Bluetoothアドレスの取得
getIpAddress	IPアドレスの取得
getMacAddress	MACアドレスの取得
getSerialNumber	シリアルナンバーの取得

- **Public Methods**

getPrinterModel　プリンタモデル名の取得

プリンタモデル名の文字列を取得します。

構文　　public String **getPrinterModel**()

戻り値　　プリンタモデル名

getBluetoothAddress　Bluetoothアドレスの取得

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

getIpAddress　IPアドレスの取得

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

getMacAddress　MACアドレスの取得

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

getSerialNumber　シリアルナンバーの取得

シリアルナンバーの文字列を取得します。

構文　　public String **getSerialNumber**()

戻り値　　シリアルナンバー

4.4.3 列挙型

(1) **PrinterInterface**

openの通信インターフェース指定で利用する列挙型定数を示します。

定数名	説明
PRN_IF_TCP	TCP/IP
PRN_IF_BT	Bluetooth
PRN_IF_USB	USB

(2) **PrinterModel**

openのプリンタモデル指定で利用する列挙型定数を示します。

定数名	説明
PRN_MODEL_MP_A40	MP-A40

(3) **Dithering**

registerLogo、**sendDataFile**、**printPDF**のディザリングで利用する列挙型定数を示します。

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

(4) **PrinterInformation**

getPrinterInformation、**getPrinterInformationNumber**、**getPrinterInformationString**のプリンタ情報指定で利用する列挙型定数を示します。

取得できるプリンタ情報の詳細は、『MP-A40シリーズ サーマルプリンタ 技術説明書』を参照してください。

定数名	説明（プリンタ情報）
GET_NV_MEM_CAP	NVグラフィックスメモリ容量
GET_NV_MEM_REM_CAP	NVグラフィックスメモリの残容量
GET_NV_MEM_KEYCODE_LIST	NVグラフィックスのキーコード一覧
GET_REM_USER_MEM_CAP_DEFRAG	ユーザ領域整理後の残り容量
GET_REM_USER_MEM_CAP	ユーザ領域の残り容量
GET_FUNC_SET_RESP	機能設定応答
GET_PRN_ID_MODEL	プリンタIDの送信(機種ID)
GET_PRN_ID_TYPE	プリンタIDの送信(タイプID)
GET_PRN_ID_ROM_VER	プリンタIDの送信(ROMバージョンID)
GET_PRN_ID_FIRM_VER_MAIN	プリンタIDの送信 (ファームウェアバージョン(main))
GET_PRN_ID_MFR	プリンタIDの送信(メーカ名)
GET_PRN_ID_MODEL_NAME	プリンタIDの送信(モデル名)
GET_PRN_ID_FIRM_VER_BOOT	プリンタIDの送信 (ファームウェアバージョン(boot))

定数名	説明（プリンタ情報）
GET_PRN_ID_FIRM_CHECKSUM_BOOT	プリンタIDの送信 (ファームウェアチェックサム(boot))
GET_PRN_ID_FIRM_CHECKSUM_MAIN	プリンタIDの送信 (ファームウェアチェックサム(main))
GET_PRN_ID_FIRM_CHECKSUM	プリンタIDの送信 (ファームウェアチェックサム(main+boot))
GET_MAINT_NUM_FEED_LINE	メンテナンスカウンタ (紙送り行数(100ドットライン単位))
GET_MAINT_NUM_HEAD_ACTIVE	メンテナンスカウンタ(サーマルヘッド通電回数 (100ドットライン単位))
GET_MAINT_DRIVE_TIME	メンテナンスカウンタ (製品の駆動時間(分単位))
GET_MAINT_NUM_FEED_LINE_INTEGRATION	メンテナンスカウンタ (紙送り行数(100ドットライン単位)(積算))
GET_MAINT_ NUM_HEAD_ACTIVE_INTEGRATION	メンテナンスカウンタ(サーマルヘッド通電回数 (100ドットライン単位)(積算))
GET_MAINT_DRIVE_TIME_INTEGRATION	メンテナンスカウンタ (製品の駆動時間(分単位)(積算))
GET_HFONT_24_CHECKSUM	半角フォントIDの送信 (24ドットフォント、チェックサム)
GET_HFONT_24_ID	半角フォントIDの送信(24ドットフォント、ID)
GET_HFONT_24_INT_CHAR	半角フォントIDの送信 (24ドットフォント、登録済み国際文字)
GET_HFONT_16_CHECKSUM	半角フォントIDの送信 (16ドットフォント、チェックサム)
GET_HFONT_16_ID	半角フォントIDの送信(16ドットフォント、ID)
GET_HFONT_16_INT_CHAR	半角フォントIDの送信 (16ドットフォント、登録済み国際文字)
GET_FFONT_LANG	全角フォントIDの送信(言語名)
GET_FFONT_STANDARD	全角フォントIDの送信(規格名)
GET_FFONT_COMPANY	全角フォントIDの送信(社名)
GET_FFONT_CHECKSUM	全角フォントIDの送信(チェックサム)
GET_FFONT_DATA_SIZE	全角フォントIDの送信(データサイズ)

(5) **TransactionFunction**

controlTransactionの一括処理制御方法指定で利用する列挙型定数を示します。

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

(6) **Direction**

selectPageModeの印字方向指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

始点が"左上"または"右下"の場合、紙送り方向と垂直方向に印字データを展開します。

始点が"右上"または"左下"の場合、紙送り方向に印字データを展開します。

定数名	説明
DIRECTION_LEFT_TO_RIGHT	印字方向:左→右、始点:左上(図 4-2 A)
DIRECTION_BOTTOM_TO_TOP	印字方向:下→上、始点:左下(図 4-2 B)
DIRECTION_RIGHT_TO_LEFT	印字方向:右→左、始点:右下(図 4-2 C)
DIRECTION_TOP_TO_BOTTOM	印字方向:上→下、始点:右上(図 4-2 D)

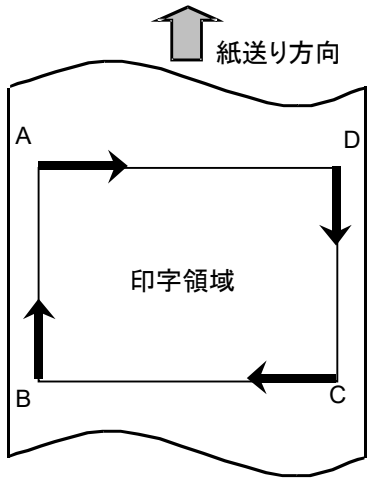


図 4-2

(7) **Alignment**

setStandardModeAlignmentの位置揃え指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

定数名	説明
ALIGN_LEFT	左揃え
ALIGN_CENTER	中央揃え
ALIGN_RIGHT	右揃え

(8) **CharacterSet**

selectCharacterSet、及び**getCharacterSet**の文字セット選択で利用する列挙型定数を示します。

デフォルト値は、Androidデバイスの言語設定により下記の通りです。

日本語の場合: **CODEPAGE_KATAKANA**

日本語以外の場合: **CODEPAGE_1252**

定数名	説明
CODEPAGE_437	USA, Standard Europe (Code Page 437)
CODEPAGE_KATAKANA	Katakana
CODEPAGE_850	Multilingual (Code Page 850)
CODEPAGE_860	Portuguese (Code Page 860)
CODEPAGE_863	Canadian-French (Code Page 863)
CODEPAGE_865	Nordic (Code Page 865)
CODEPAGE_857[*1]	Turkish (Code Page 857)
CODEPAGE_737	Greek (Code Page 737)
CODEPAGE_1252	Latin (Code Page 1252)
CODEPAGE_866	Russian (Code Page 866)
CODEPAGE_852	Eastern Europe (Code Page 852)
CODEPAGE_858	Euro (Code Page 858)
CODEPAGE_855	Cyrillic (Code Page 855)
CODEPAGE_864[*1]	Arabic (Code Page 864)
CODEPAGE_1250	Central European (Code Page 1250)
CODEPAGE_1251	Cyrillic (Code Page 1251)
CODEPAGE_1253	Greek (Code Page 1253)
CODEPAGE_1254	Turkish (Code Page 1254)

*1: UnicodeのU20ACHは印字できません。

(9) **InternationalCharacterSet**

selectInternationalCharacterSet、及び**getInternationalCharacter**の国際文字セット選択で利用する列挙型定数を示します。

デフォルト値は、Androidデバイスの言語設定により下記の通りです。

日本語の場合　　**INT_CHAR_SET_JAPAN**

日本語以外の場合 **INT_CHAR_SET_USA**

定数名	説明
INT_CHAR_SET_USA	アメリカ（USA）
INT_CHAR_SET_FRANCE	フランス（France）
INT_CHAR_SET_GERMANY	ドイツ（Germany）
INT_CHAR_SET_UNITED_KINGDOM	イギリス（United Kingdom）
INT_CHAR_SET_DENMARK_1	デンマークⅠ（Denmark I）
INT_CHAR_SET_SWEDEN	スウェーデン（Sweden）
INT_CHAR_SET_ITALY	イタリア（Italy）
INT_CHAR_SET_SPAIN_1	スペインⅠ（Spain I）
INT_CHAR_SET_JAPAN	日本（Japan）
INT_CHAR_SET_NORWAY	ノルウェー（Norway）
INT_CHAR_SET_DENMARK_2	デンマークⅡ（Denmark II）
INT_CHAR_SET_SPAIN_2	スペインⅡ（Spain II）
INT_CHAR_SET_LATIN_AMERICA	ラテンアメリカ（Latin America）
INT_CHAR_SET_ARABIA	アラビア（Arabia）

(10) **CharacterType**

setCharacterFormattingの文字フォント指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

定数名	説明
CHAR_TYPE_FONT_CURRENT[*1]	現在の設定（設定するプリンタコマンドの送信なし）
CHAR_TYPE_FONT_A	フォントA(24×12)・漢字フォントA(24×24)
CHAR_TYPE_FONT_B[*1]	フォントB(16×8)・漢字フォントB(16×16)

*1: **selectCharacterSet**で**CODEPAGE_864**を選択した場合、**setCharacterFormatting**の文字フォント指定に関わらず、フォントA(24×12)で印字します。

(11) **CharacterScale**

setCharacterFormattingの文字サイズ指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

定数名	説明
CHAR_SCALE_CURRENT	現在の設定(設定するプリンタコマンドの送信なし)
CHAR_SCALE_X1	1倍
CHAR_SCALE_X2	2倍
CHAR_SCALE_X3	3倍
CHAR_SCALE_X4	4倍
CHAR_SCALE_X5	5倍
CHAR_SCALE_X6	6倍
CHAR_SCALE_X7	7倍
CHAR_SCALE_X8	8倍

(12) **Underline**

setCharacterFormattingのアンダーライン指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

定数名	説明
CHAR_UNDERLINE_CURRENT	現在の設定(設定するプリンタコマンドの送信なし)
CHAR_UNDERLINE_NONE	アンダーラインなし
CHAR_UNDERLINE_1DOT	1 ドット幅のアンダーラインあり
CHAR_UNDERLINE_2DOT	2 ドット幅のアンダーラインあり

(13) **Bold**

setCharacterFormattingの強調印字指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

定数名	説明
CHAR_BOLD_CURRENT	現在の設定(設定するプリンタコマンドの送信なし)
CHAR_BOLD_OFF	強調印字なし
CHAR_BOLD_ON	強調印字あり

(14) **Reverse**

setCharacterFormattingの白黒反転印字指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。

定数名	説明
CHAR_REVERSE_CURRENT	現在の設定(設定するプリンタコマンドの送信なし)
CHAR_REVERSE_OFF	白黒反転印字なし
CHAR_REVERSE_ON	白黒反転印字あり

(15) **Rotate**

setCharacterFormattingの回転印字指定、**setStandardModeBarcodeDirection**の印字方向指定、**printPDF**のイメージの回転方向指定で利用する列挙型定数を示します。デフォルト値は、表中の網掛けで示す定数です。**setCharacterFormatting**により回転印字を指定した場合は、文字の印字位置も変化します。文字の印字位置と向きを、図 4-3、図 4-4、図 4-5、及び図 4-6に示します。

定数名	説明
ROTATE_CURRENT[*1]	現在の設定(設定するプリンタコマンドの送信なし)
ROTATE_NONE	回転なし
ROTATE_90_TO_RIGHT[*1]	90度右回転
ROTATE_180	180度回転
ROTATE_90_TO_LEFT[*1]	90度左回転

*1: **printPDF**はサポートしていません。

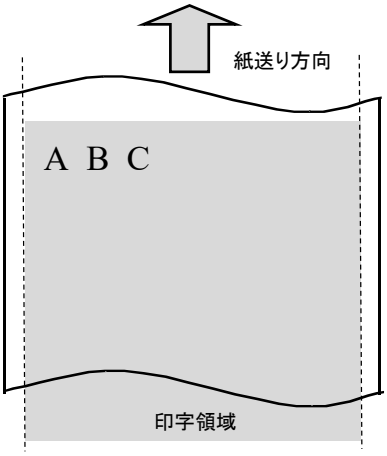


図 4-3 回転なし

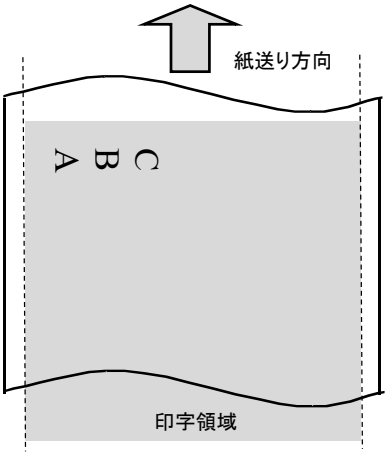


図 4-4 90度右回転

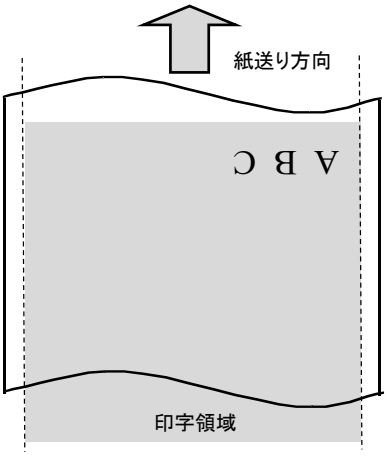


図 4-5 180度回転

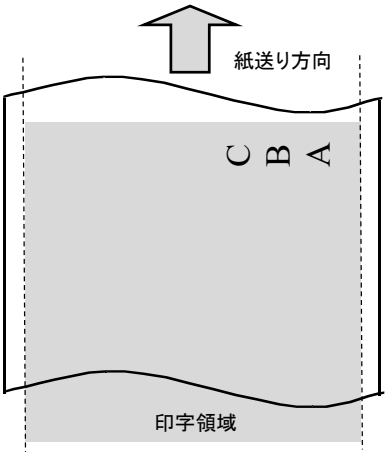


図 4-6 90度左回転

(16) **TypeBarcode**

printBarcodeのバーコードの種類指定で利用する列挙型定数を示します。

定数名	説明	
TYPE_BARCODE_UPC_A	UPC-A	マルチレベルバーコード
TYPE_BARCODE_UPC_E	UPC-E	マルチレベルバーコード
TYPE_BARCODE_JAN13	JAN13 (EAN13)	マルチレベルバーコード
TYPE_BARCODE_JAN8	JAN8 (EAN8)	マルチレベルバーコード
TYPE_BARCODE_CODE39	CODE39	2値レベルバーコード
TYPE_BARCODE_ITF	ITF	2値レベルバーコード
TYPE_BARCODE_CODABAR	CODABAR	2値レベルバーコード
TYPE_BARCODE_CODE128	CODE128	マルチレベルバーコード
TYPE_BARCODE_CODE93	CODE93	マルチレベルバーコード
TYPE_BARCODE_JAN13_ADDON2	JAN13 (EAN13) add-on 2	マルチレベルバーコード
TYPE_BARCODE_JAN13_ADDON5	JAN13 (EAN13) add-on 5	マルチレベルバーコード
TYPE_BARCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	マルチレベルバーコード
TYPE_BARCODE_GS1_TRUNCATED	GS1 Databar Truncated	マルチレベルバーコード
TYPE_BARCODE_GS1_LIMITED	GS1 Databar Limited	マルチレベルバーコード
TYPE_BARCODE_GS1_EXPANDED	GS1 Databar Expanded	マルチレベルバーコード

(17) **ModuleWidthBarcode**

printBarcodeのバーコードのマルチレベルバーコードにおけるモジュール幅または2値レベルバーコードにおける細エレメント指定で利用する列挙型定数を示します。

定数名	説明
MODULE_WIDTH_BARCODE_2	2ドット
MODULE_WIDTH_BARCODE_3	3ドット
MODULE_WIDTH_BARCODE_4	4ドット
MODULE_WIDTH_BARCODE_5	5ドット
MODULE_WIDTH_BARCODE_6	6ドット

(18) **HriPositionBarcode**

printBarcodeのバーコードのHRI文字指定で利用する列挙型定数を示します。

定数名	説明
HRI_BARCODE_NONE	HRI文字なし
HRI_BARCODE_TOP_FONT_A	バーコードの上(フォントA)
HRI_BARCODE_BOTTOM_FONT_A	バーコードの下(フォントA)
HRI_BARCODE_FONT_A	バーコードの上下(フォントA)
HRI_BARCODE_TOP_FONT_B	バーコードの上(フォントB)
HRI_BARCODE_BOTTOM_FONT_B	バーコードの下(フォントB)
HRI_BARCODE_FONT_B	バーコードの上下(フォントB)

(19) **NwRatioBarcode**

printBarcodeの2値レベルバーコードのN:W比指定で利用する列挙型定数を示します。

定数名	説明
NWRATIO_BARCODE_1TO2	1:2
NWRATIO_BARCODE_1TO2_5	1:2.5
NWRATIO_BARCODE_1TO3	1:3

(20) **Type2DCode**

print2DCodeの2次元バーコードの種類指定で利用する列挙型定数を示します。

定数名	説明
TYPE_2DCODE_QR_CODE	QR Code
TYPE_2DCODE_PDF417	PDF417
TYPE_2DCODE_DATA_MATRIX	Data Matrix
TYPE_2DCODE_MAXI_CODE	Maxi Code
TYPE_2DCODE_GS1_STACKED	GS1 Databar Stacked
TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Stacked Omni-directional
TYPE_2DCODE_GS1_EXPANDED_STACKED	GS1 Databar Expanded Stacked

(21) **Mode2Dcode**

print2DCodeの2次元バーコードのモードで利用する列挙型定数を示します。

定数名	説明	
MODE_2DCODE_QR_CODE_MODEL1	QR Code	Model1
MODE_2DCODE_QR_CODE_MODEL2	QR Code	Model2
MODE_2DCODE_PDF417_STANDARD	PDF417	通常モード
MODE_2DCODE_PDF417_COMPACT	PDF417	簡易モード
MODE_2DCODE_DATA_MATRIX_AUTO	Data Matrix	モジュール数自動
MODE_2DCODE_DATA_MATRIX_10_10	Data Matrix	モジュール数10×10
MODE_2DCODE_DATA_MATRIX_12_12	Data Matrix	モジュール数12×12

定数名	説明	
MODE_2DCODE_DATA_MATRIX_14_14	Data Matrix	モジュール数14×14
MODE_2DCODE_DATA_MATRIX_16_16	Data Matrix	モジュール数16×16
MODE_2DCODE_DATA_MATRIX_18_18	Data Matrix	モジュール数18×18
MODE_2DCODE_DATA_MATRIX_20_20	Data Matrix	モジュール数20×20
MODE_2DCODE_DATA_MATRIX_22_22	Data Matrix	モジュール数22×22
MODE_2DCODE_DATA_MATRIX_24_24	Data Matrix	モジュール数24×24
MODE_2DCODE_DATA_MATRIX_26_26	Data Matrix	モジュール数26×26
MODE_2DCODE_DATA_MATRIX_32_32	Data Matrix	モジュール数32×32
MODE_2DCODE_DATA_MATRIX_36_36	Data Matrix	モジュール数36×36
MODE_2DCODE_DATA_MATRIX_40_40	Data Matrix	モジュール数40×40
MODE_2DCODE_DATA_MATRIX_44_44	Data Matrix	モジュール数44×44
MODE_2DCODE_DATA_MATRIX_48_48	Data Matrix	モジュール数48×48
MODE_2DCODE_DATA_MATRIX_52_52	Data Matrix	モジュール数52×52
MODE_2DCODE_DATA_MATRIX_64_64	Data Matrix	モジュール数64×64
MODE_2DCODE_DATA_MATRIX_72_72	Data Matrix	モジュール数72×72
MODE_2DCODE_DATA_MATRIX_80_80	Data Matrix	モジュール数80×80
MODE_2DCODE_DATA_MATRIX_88_88	Data Matrix	モジュール数88×88
MODE_2DCODE_DATA_MATRIX_96_96	Data Matrix	モジュール数96×96
MODE_2DCODE_DATA_MATRIX_104_104	Data Matrix	モジュール数104×104
MODE_2DCODE_DATA_MATRIX_120_120	Data Matrix	モジュール数120×120
MODE_2DCODE_DATA_MATRIX_132_132	Data Matrix	モジュール数132×132
MODE_2DCODE_DATA_MATRIX_144_144	Data Matrix	モジュール数144×144
MODE_2DCODE_DATA_MATRIX_8_18	Data Matrix	モジュール数8×18
MODE_2DCODE_DATA_MATRIX_8_32	Data Matrix	モジュール数8×32
MODE_2DCODE_DATA_MATRIX_12_26	Data Matrix	モジュール数12×26
MODE_2DCODE_DATA_MATRIX_12_36	Data Matrix	モジュール数12×36
MODE_2DCODE_DATA_MATRIX_16_36	Data Matrix	モジュール数16×36
MODE_2DCODE_DATA_MATRIX_16_48	Data Matrix	モジュール数16×48
MODE_2DCODE_MAXI_CODE_2	Maxi Code	Mode2
MODE_2DCODE_MAXI_CODE_3	Maxi Code	Mode3
MODE_2DCODE_MAXI_CODE_4	Maxi Code	Mode4
MODE_2DCODE_MAXI_CODE_5	Maxi Code	Mode5
MODE_2DCODE_NONE	GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	設定なし

(22) **ModuleSize2Dcode**

print2DCodeの2次元バーコードのモジュールサイズ指定で利用する列挙型定数を示します。

定数名	説明	
MODULE_SIZE_2DCODE_DEFAULT	QR Code	初期値（6ドット）
	PDF417	初期値（4ドット）
	Data Matrix	初期値（6ドット）
	Maxi Code	設定なし
	GS1 Databar Stacked	初期値（6ドット）
	GS1 Databar Stacked Omni-directional	初期値（6ドット）
	GS1 Databar Expanded Stacked	初期値（6ドット）
MODULE_SIZE_2DCODE_2	QR Code	2ドット
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_3	QR Code	3ドット
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_4	QR Code	4ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_5	QR Code	5ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_6	QR Code	6ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_7	QR Code	7ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

定数名	説明	
MODULE_SIZE_2DCODE_8	QR Code	8ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_9	QR Code	9ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_10	QR Code	10ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_11	QR Code	11ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_12	QR Code	12ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_13	QR Code	13ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_14	QR Code	14ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
MODULE_SIZE_2DCODE_15	QR Code	15ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

定数名	説明	
MODULE_SIZE_2DCODE_16	QR Code	16ドット
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

(23) **ErrorCorrect2Dcode**

print2DCodeの2次元バーコードのエラー訂正レベル指定で利用する列挙型定数を示します。

定数名	説明	
ERR_CORRECTION_2DCODE_QR_CODE_L	QR Code	L
ERR_CORRECTION_2DCODE_QR_CODE_M	QR Code	M
ERR_CORRECTION_2DCODE_QR_CODE_Q	QR Code	Q
ERR_CORRECTION_2DCODE_QR_CODE_H	QR Code	H
ERR_CORRECTION_2DCODE_PDF417_0	PDF417	0
ERR_CORRECTION_2DCODE_PDF417_1	PDF417	1
ERR_CORRECTION_2DCODE_PDF417_2	PDF417	2
ERR_CORRECTION_2DCODE_PDF417_3	PDF417	3
ERR_CORRECTION_2DCODE_PDF417_4	PDF417	4
ERR_CORRECTION_2DCODE_PDF417_5	PDF417	5
ERR_CORRECTION_2DCODE_PDF417_6	PDF417	6
ERR_CORRECTION_2DCODE_PDF417_7	PDF417	7
ERR_CORRECTION_2DCODE_PDF417_8	PDF417	8
ERR_CORRECTION_2DCODE_NONE	Data Matrix Maxi Code GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	設定なし

(24) **LogFileSize**

setLogのログファイルの最大サイズ指定で利用する列挙型定数を示します。

定数名	説明
LOG_FILE_SIZE_1MB	1 MB (1048576バイト)
LOG_FILE_SIZE_5MB	5 MB (5242880バイト)
LOG_FILE_SIZE_10MB	10 MB (10485760バイト)
LOG_FILE_SIZE_50MB	50 MB (52428800バイト)

(25) **ErrorCode**

getErrorCodeで取得するエラー内容を示す列挙型定数を示します。

定数名	説明
ERR_PARAM	パラメーターが不正です。
ERR_OPENED	指定したプリンタはすでにオープン済みです。
ERR_NOT_OPENED	指定したプリンタはオープンされていません。
ERR_TIMEOUT	タイムアウト、またはビジー状態です。
ERR_OFFLINE	切断状態またはプリンタがオフラインです。
ERR_CLOSE_FAIL	プリンタの切断に失敗しました。
ERR_NOT_MONITORING	接続状態が監視されていません。
ERR_INIT_FAILED	初期化に失敗しました。
ERR_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。
ERR_OVER_MAX_DATA_SIZE	データサイズが最大値を超過しています。
ERR_INVALID_DATA	不正なデータが指定されています。
ERR_INVALID_STATE	コールバックに**PrinterManager**オブジェクトへのアクセスが指定されました。
ERR_ACCESS	プリンタにアクセスできません。
ERR_CANCELED	機能が取り消されました。
ERR_WRITE_FAULT	プリンタにデータを送信できません。
ERR_WORKAREA_NO_MEMORY	メモリの領域指定が不足しています。
ERR_FILE_INVALID	指定されたファイルは無効です。
ERR_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。
ERR_NOT_FOUND	指定したファイルが見つかりません。
ERR_TRANSACTION_STOPPED	一括処理が開始されていません。
ERR_PRINTER_STATUS_ERROR	プリンタステータスが正常ではありません。

4.4.4 例外

(1) **PrinterException**

- **概要**

  PrinterException クラスは、例外状況が発生したことを示すクラスです。
  下記の機能を提供します。

  Public Methods

名前	説明
getErrorCode	エラーコードの取得

- **Public Methods**

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 ErrorCode **getErrorCode**()

戻り値　　エラーコード

説明　　エラー内容詳細は、『**4.4.3**(25) **ErrorCode**』を参照してください。

5章　サンプルプログラム

本章では、本SDKで提供するサンプルプログラムについて説明します。

5.1　サンプルプログラムの概要

本SDK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。

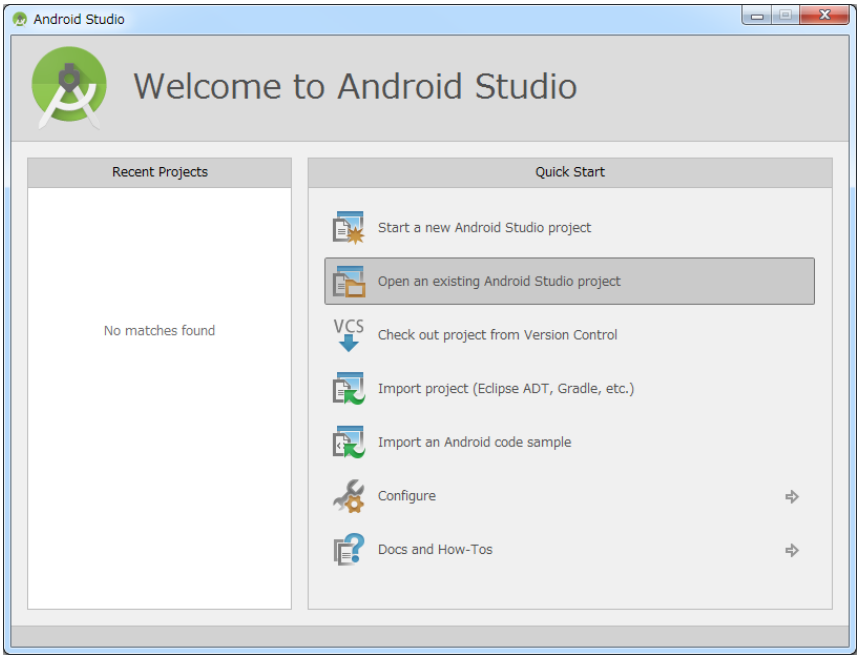
5.2　サンプルプログラムの利用方法

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 本ライブラリの利用方法」を参照してください。

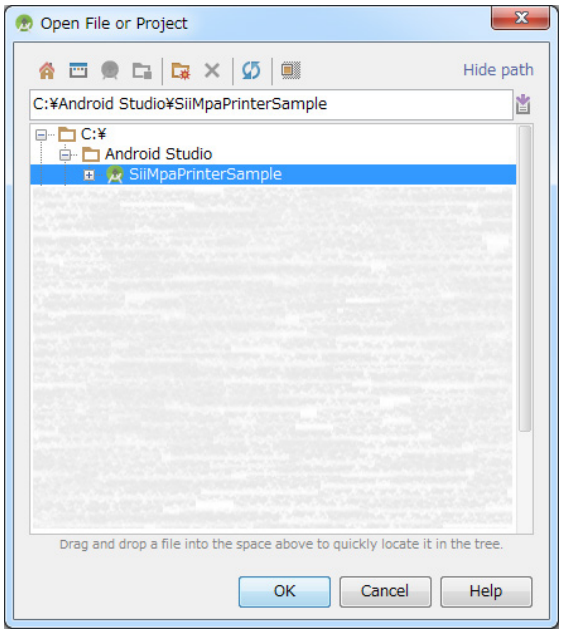
サンプルプログラムのインストール手順は下記の通りです。

(1) SiiMpaPrinterSampleフォルダを任意の位置に配置してください。

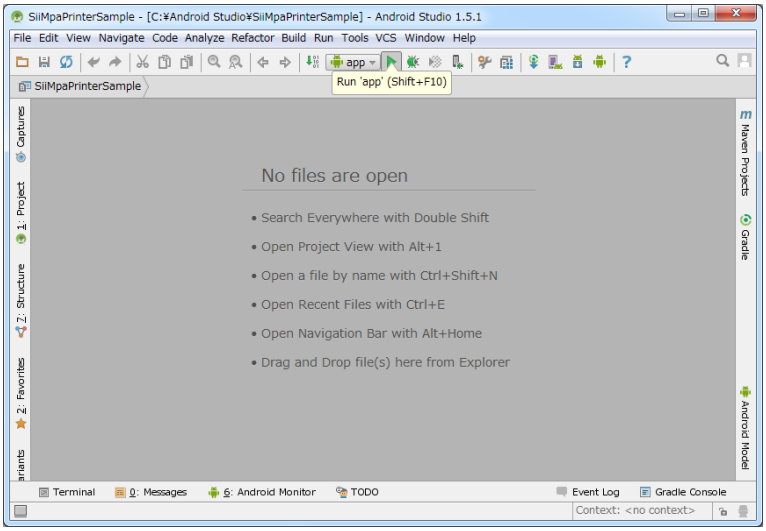
(2) Android Studioを起動し、「Open an existing Android Studio Project」をクリックしてください。



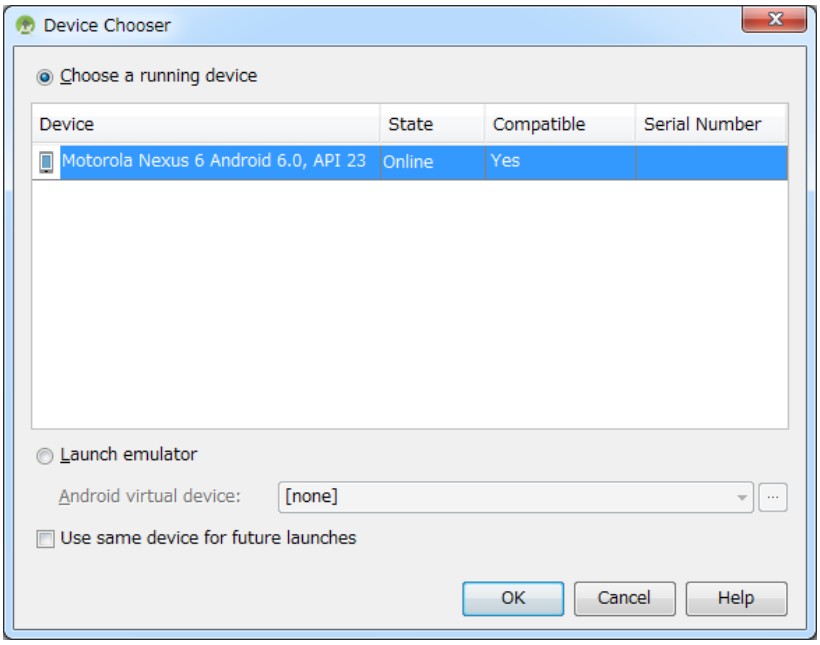
(3) 手順(1)で配置したフォルダを選択し、[OK]をクリックしてください。



(4) [Run 'app']をクリックしてください。

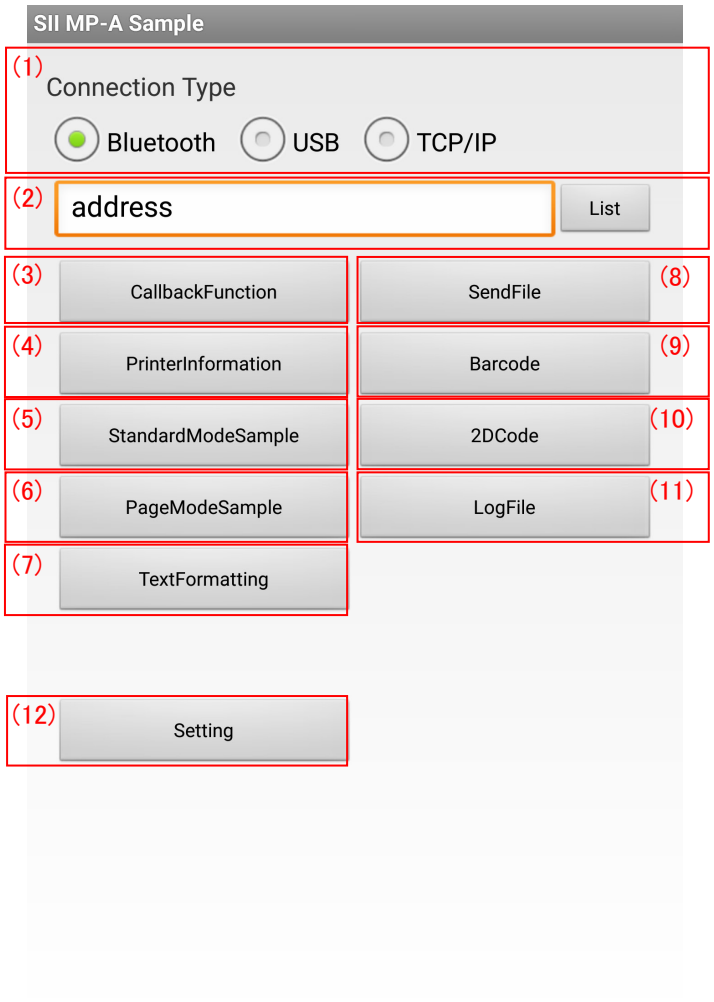


(5) デバイスを選択し、[OK]をクリックしてください。



5.3　サンプルプログラムの機能

SiiMpaPrinterSampleプロジェクトの画面を例に、サンプルプログラムの機能を説明します。



番号	機能名	詳細
(1)	Connection type	プリンタへの接続形式を選択します。
(2)	Address	プリンタのアドレスを指定します。 Bluetoothで接続する場合には、プリンタのBluetoothアドレスを入力してください。 [List]ボタンのタップにより、**startDiscoveryPrinter**(Bluetooth)により発見されたデバイスの一覧を表示します。一覧から接続対象のプリンタを選択することで、Bluetoothアドレスを入力できます。 例：″00:11:22:AA:BB:CC″ TCP/IPで接続する場合には、プリンタのIPアドレスを入力してください。 [List]ボタンのタップにより、**startDiscoveryPrinter**(TCP/IP)により発見されたデバイスの一覧を表示します。一覧から接続対象のプリンタを選択することで、IPアドレスを入力できます。 例：″192.168.0.190″
(3)	CallbackFunction	プリンタステータス変化のコールバックを**startCallbackFunction**に登録し、ステータス情報の変化をダイアログで表示します。
(4)	PrinterInformation	**getPrinterInformation**、**getPrinterInformationNumber**、または**getPrinterInformationString**を用いてプリンタからプリンタ情報を取得します。

番号	機能名	詳細
(5)	StandardModeSample	スタンダードモードにおいて、各種印字メソッドを用いてサンプルレシート印字コマンドをプリンタに送信します。
(6)	PageModeSample	ページモードにおいて、各種印字メソッドを用いてサンプルレシート印字コマンドをプリンタに送信します。
(7)	TextFormatting	任意のテキストを各種文字修飾にて修飾した印字コマンドをプリンタに送信します。
(8)	SendFile	**sendDataFile**を用いて指定したファイルをプリンタに送信します。
(9)	Barcode	**printBarcode**を用いてバーコード印字コマンドをプリンタに送信します。
(10)	2DCode	**print2Dcode**を用いて2次元バーコード印字コマンドをプリンタに送信します。
(11)	LogFile	**setLog**を用いて保存するログファイルサイズを設定します。
(12)	Setting	次の機能を設定します。 ・Socket Keeping Time TCP/IP接続のソケット維持時間をmsec(ミリ秒)単位で設定します。 **open**の引数*socketKeepingTime*に用います。 ・Write Timeout 送信タイムアウト発生までの時間をmsec(ミリ秒)単位で設定します。 ・Response Timeout 受信タイムアウト発生までの時間をmsec(ミリ秒)単位で設定します。 ・Codepage コードページを設定します。 テキスト印字時(TextFormatting及びSendFileでテキストファイル指定時)のエンコードとプリンタ側の文字セット設定値です。 ・International Character 国際文字セットを設定します。 テキスト印字時(TextFormatting及びSendFileでテキストファイル指定時)のプリンタ側の国際文字セット設定です。 ・Bluetooth Secure Bluetooth接続時にセキュアなモードで接続するか、インセキュアなモードで接続するかを設定します。

5.4 注意事項

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。

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。

6章　免責

本ソフトウェアは不具合の発生のなきよう十分留意して開発しておりますが、万一、本ソフトウェアによってお客様に損害を及ぼした場合、一切の責任を負いかねますのでご了承ください。

付録 A　文字セット（文字コード表）

A.1 文字コード表（文字コードテーブル）

（国際文字設定でCOUNTRY_USA設定時）

	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	¢	£	¥	₧	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B0	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C0	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D0	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E0	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F0	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

表 A-1 **CODEPAGE_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	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	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	゛	゜
E0																
F0																

表 A-2 **CODEPAGE_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	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	ª	º	¿	®	¬	½	¼	¡	«	»
B0	░	▒	▓	|	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
C0	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
D0	ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	█	▄	¦	Ì	▀
E0	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
F0	-	±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	

表 A-3 **CODEPAGE_850** (Multilingual)

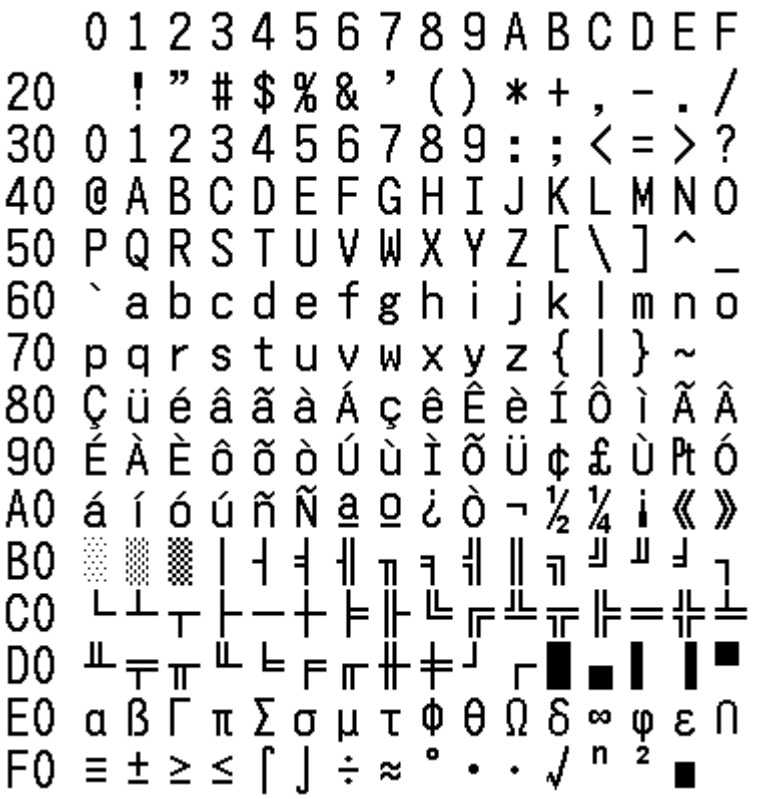


表 A-4 **CODEPAGE_860** (Portuguese)

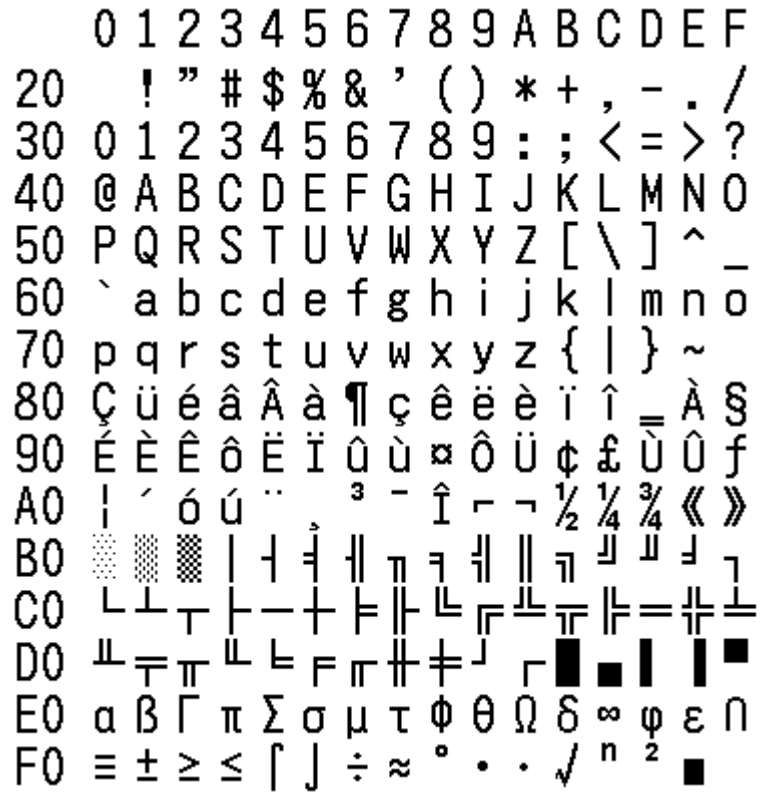


表 A-5 **CODEPAGE_863** (Canadian-French)

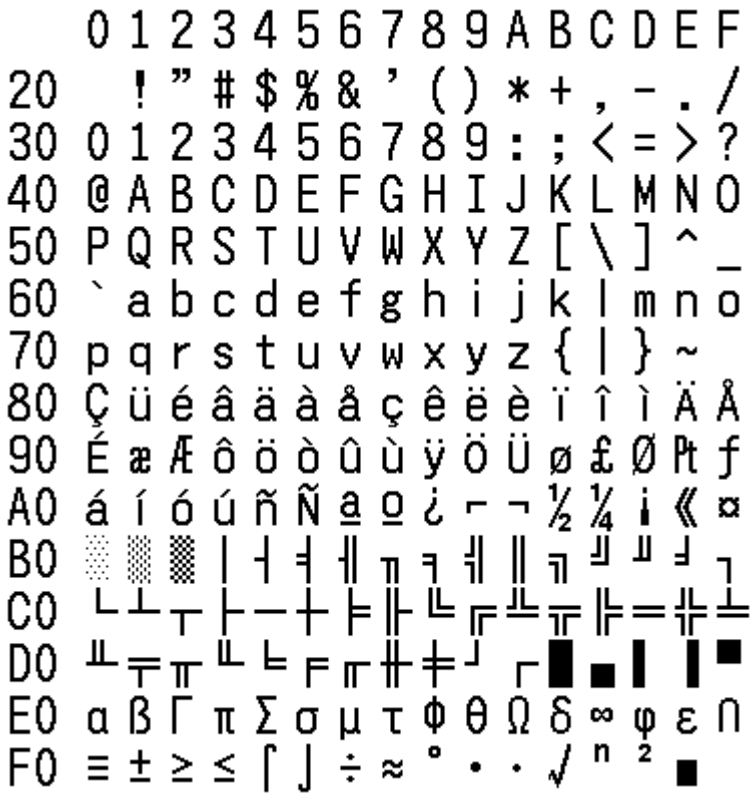


表 A-6 **CODEPAGE_865** (Nordic)

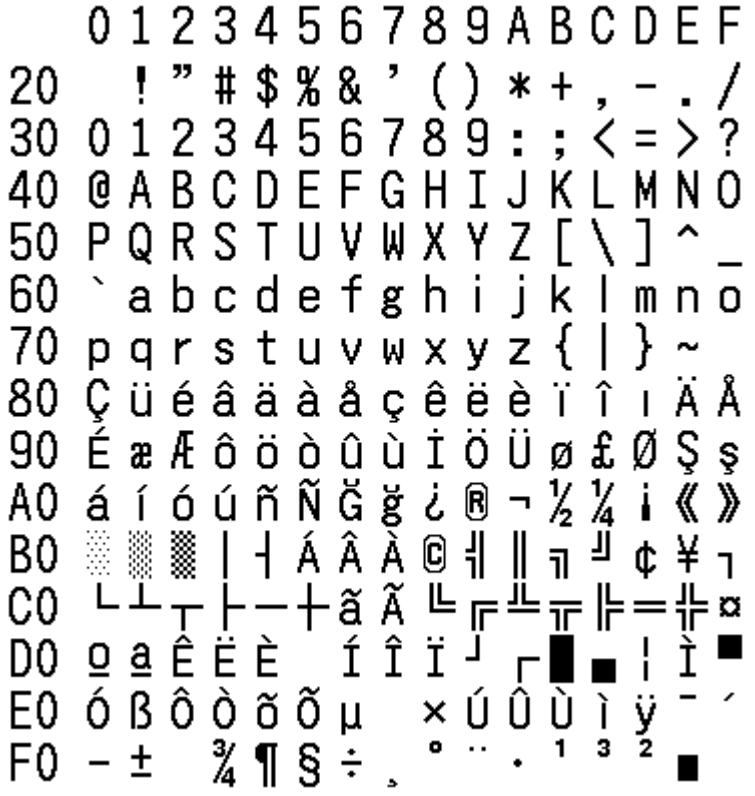


表 A-7 **CODEPAGE_857** (Turkish)

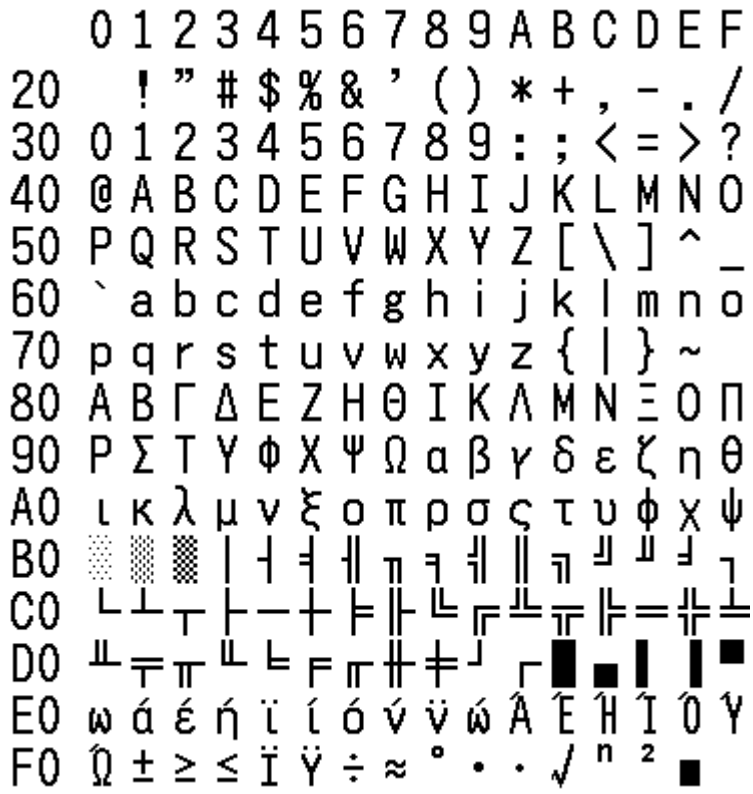


表 A-8 **CODEPAGE_737** (Greek)

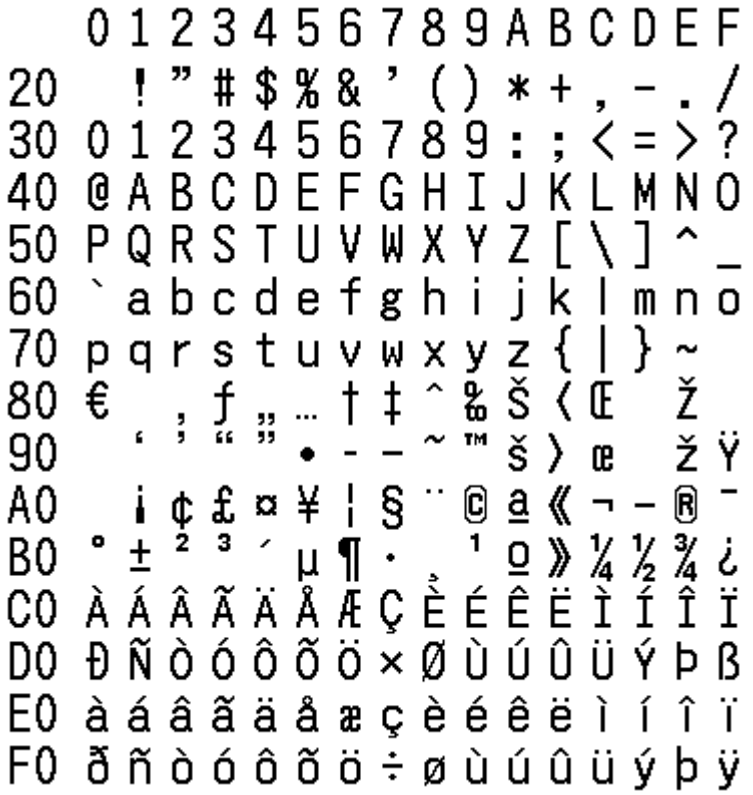


表 A-9 **CODEPAGE_1252** (Latin)

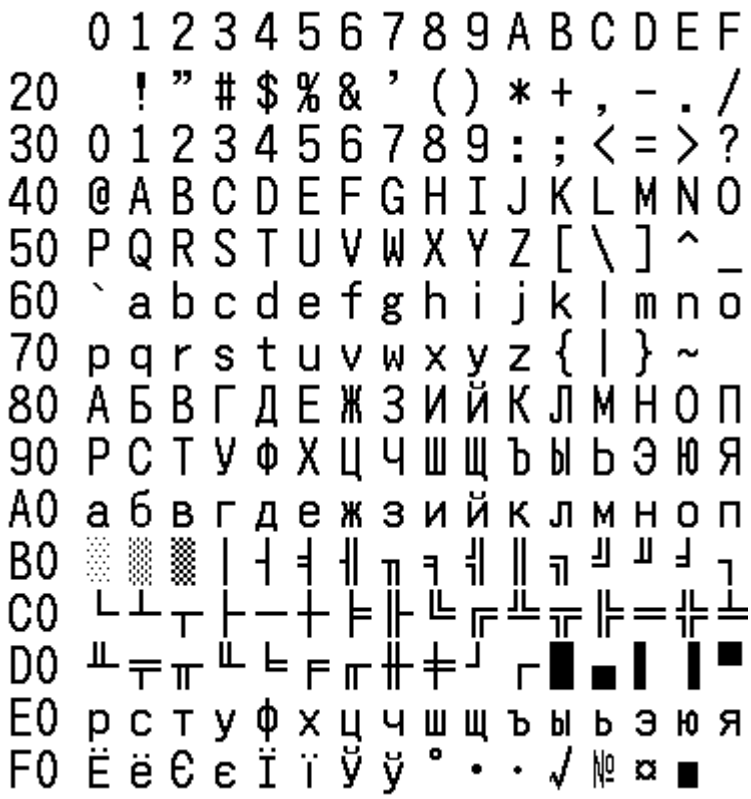


表 A-10 **CODEPAGE_866** (Russian)

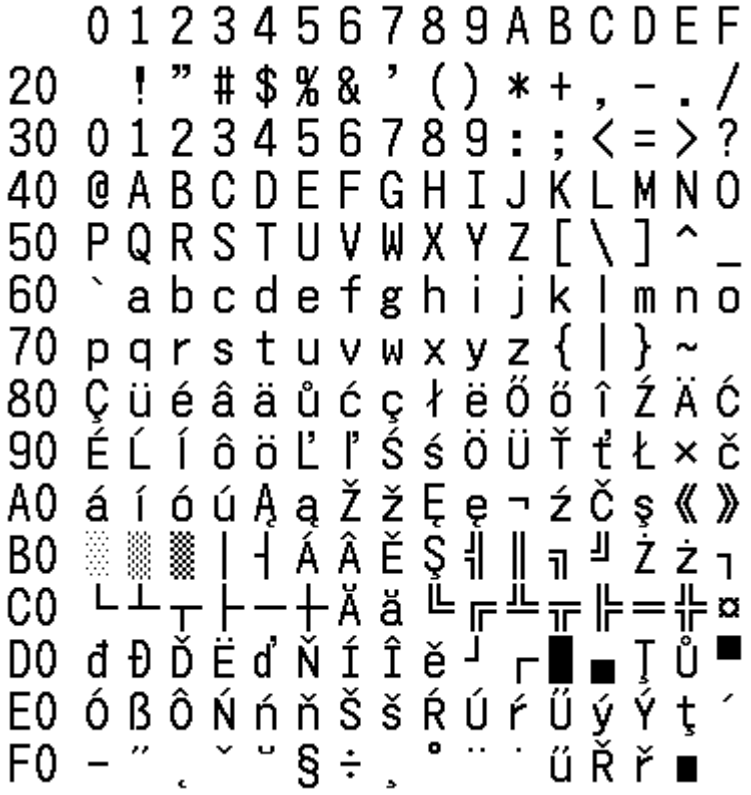


表 A-11 **CODEPAGE_852** (Eastern Europe)

	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	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	ª	º	¿	®	¬	½	¼	¡	«	»
B0	░	▒	▓	|	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
C0	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	┘	┌	█	▄	¦	Ì	▀
E0	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
F0	-	±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	

表 A-12 **CODEPAGE_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	s	S	i	I	ї	Ї	j	J
90	љ	Љ	њ	Њ	ћ	Ћ	ќ	Ќ	ў	Ў	џ	Џ	ю	Ю	ъ	Ъ
A0	а	А	б	Б	ц	Ц	д	Д	е	Е	ф	Ф	г	Г	«	»
B0	░	▒	▓	|	┤	х	Х	и	И	╣	║	╗	╝	й	Й	┐
C0	└	┴	┬	├	─	┼	к	К	╚	╔	╩	╦	╠	═	╬	¤
D0	л	Л	м	М	н	Н	о	О	п	┘	┌	█	▄	П	я	▀
E0	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	Ь	№
F0	-	ы	Ы	з	З	ш	Ш	э	Э	щ	Щ	ч	Ч	§	■	

表 A-13 **CODEPAGE_855** (Cyrillic)



表 A-14 **CODEPAGE_864** (Arabic)



表 A-15 **CODEPAGE_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	J	¤	Ґ	¦	§	Ё	©	Є	«	¬	-	®	Ї
B0	°	±	І	і	ґ	µ	¶	·	ё	№	є	»	j	S	s	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

表 A-16 **CODEPAGE_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		΅	Ά	£	¤	¥	¦	§	¨	©		«	¬	-	®	―
B0	°	±	²	³	΄	µ	¶	·	Έ	Ή	Ί	»	Ό	½	Ύ	Ώ
C0	ΐ	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	Ϊ	Ϋ	ά	έ	ή	ί
E0	ΰ	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	ϊ	ϋ	ό	ύ	ώ	

表 A-17 **CODEPAGE_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	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ			
90		‘	’	“	”	•	-	—	~	™	š	›	œ			Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	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

表 A-18 **CODEPAGE_1254** (Turkish)

A.2 国際文字セット

下記のコードが指定の国際文字によって異なります。

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

表 A-19 国際文字セット