



SII Print Class Library for Android™ Application Programmer's Guide

Rev.05

[Products]

MP-B20 Series

Seiko Instruments Inc.

Rev.01	March 2017
Rev.02	February 2018
Rev.03	February 2019
Rev.04	August 2019
Rev.05	March 2020

Copyright © 2017-2020 by Seiko Instruments Inc.
All rights reserved.

Android™ is a trademark of Google LLC.
Bluetooth® is registered trademark of Bluetooth SIG, Inc.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.
Other names may be trademarks of their respective owners.

Eclipse is a trademark of Eclipse Foundation, Inc.

Seiko Instruments Inc. (hereinafter referred to as "SII") has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

Introduction

This document describes the "SII Print Class Library for Android™" for MP-B20 Series (hereinafter referred to as "SII print class library") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target printers

This section lists the printers supported by SII print class library.

Printers	Interface
MP-B20 Series	Bluetooth
	USB

Terms

This section describes terms used in this manual.

Terms	Description
Printer Command	Command for controlling the printer described in "MP-B20 Series Thermal Printer Technical References"

Table of Contents

Chapter 1	Product Overview	1-1
1.1	Functions Provided by SII Print Class Library	1-1
1.2	SII Print Class Library Overview	1-1
1.2.1	SII Print Class Library Configuration	1-1
1.2.2	Functions Provided by the Library	1-2
Chapter 2	Product Specifications	2-1
2.1	Operating Environment.....	2-1
2.2	Operating Conditions.....	2-2
Chapter 3	How to Use the Library	3-1
3.1	Development Environment for Android Application.....	3-1
3.2	Provided Files	3-2
3.3	Add the Library to Android Studio Projects.....	3-3
3.4	Use Developed Android Application on Android Device	3-5
Chapter 4	Functions of the Library	4-1
4.1	API Reference.....	4-1
4.1.1	CallbackFunctionListener Interface	4-2
	onStatusChanged Change event of printer status.....	4-2
4.1.2	BarcodeScannerListener Interface.....	4-3
4.1.3	PrinterManager Class.....	4-4
	PrinterManager Constructor	4-16
	connect Start communicating with printer (Bluetooth).....	4-16
	connect Start communicating with printer (USB)	4-17
	disconnect Stop communicating with printer	4-17
	sendText Send text data	4-18
	sendTextEx Send format specified text data.....	4-18
	printBarcode Print barcode.....	4-19
	printPDF417 Print PDF417.....	4-22
	printQRcode Print QR Code.....	4-23
	printDataMatrix Print Data Matrix.....	4-24
	printMaxiCode Print MaxiCode.....	4-24
	printGS1DataBarStacked Print GS1 Databar Stacked.....	4-25
	printGS1DataBarStackedOmnidirectional Print GS1 Databar Stacked Omni-directional	4-25
	printGS1DataBarExpandedStacked Print GS1 Databar Expanded Stacked	4-26
	cutPaper Cut paper	4-26

openDrawer	Open cash drawer	4-27
buzzer	Sound buzzer	4-27
externalBuzzer	Sound external buzzer.....	4-27
sendBinary	Send binary data	4-27
sendDataFile	Send specified file	4-27
getStatus	Get printer status.....	4-28
setCallbackFunctionListener	Start/End callback of printer status change	4-29
abort	Abort waiting state of printer	4-30
registerLogo	Register logo	4-30
printLogo	Print logo.....	4-31
unregisterLogo	Delete registered logo	4-31
registerStyleSheet	Register style sheet.....	4-31
unregisterStyleSheet	Delete registered style sheet	4-31
resetPrinter	Reset printer.....	4-32
getPrinterResponse	Get various responses from printer	4-32
startDiscoveryPrinter	Start printer search (Bluetooth).....	4-33
startDiscoveryPrinter	Start printer search (USB)	4-34
startDiscoveryPrinter	Start printer search (TCP/IP)	4-34
cancelDiscoveryPrinter	Cancel printer search	4-34
getFoundPrinter	Get found printer information	4-34
getSendTimeout	Get send timeout period	4-35
setSendTimeout	Set send timeout period.....	4-35
getReceiveTimeout	Get receive timeout period.....	4-35
setReceiveTimeout	Set receive timeout period	4-35
getInternationalCharacter	Get international character set.....	4-36
setInternationalCharacter	Set international character set	4-36
getCodePage	Get codepage.....	4-36
setCodePage	Set codepage	4-36
getPrinterModel	Get printer model.....	4-37
getPortType	Get connecting port type.....	4-37
isConnect	Verify connection state with printer	4-37
getSocketKeepingTime	Get socket keeping time	4-37
setSocketKeepingTime	Set socket keeping time.....	4-37
controlTransaction	Start/End batch processing.....	4-38
4.1.4 PrinterEvent Class.....		4-39
getEventType	Get end event.....	4-39
4.1.5 PrinterListener Interface		4-40
finishEvent	End event of the printer search	4-40
4.1.6 PrinterInfo Class.....		4-41
getPrinterModelName	Get printer model name.....	4-41
getBluetoothAddress	Get Bluetooth address.....	4-41
getMacAddress	Get MAC address.....	4-41
getIsBonded	Get pairing status	4-41
getDevicePath	Get device path	4-42
4.1.7 PrinterException Class		4-43

PrinterException	Constructor	4-44
getErrorCode	Get error codes	4-44

Chapter 5	Sample Program	5-1
<hr/>		
5.1	Installation.....	5-1
5.2	Screen	5-3
5.2.1	Main screen.....	5-3
5.2.2	[SETTINGS] screen.....	5-4
5.3	Precaution.....	5-4
Chapter 6	Disclaimer	6-1
<hr/>		
Appendix A	Character Set	A-1
<hr/>		
A.1	Code Page Table (Character Code Table).....	A-1
A.2	International Character Set.....	A-11

Chapter 1

Product Overview

This chapter describes the product overview of SII print class library.

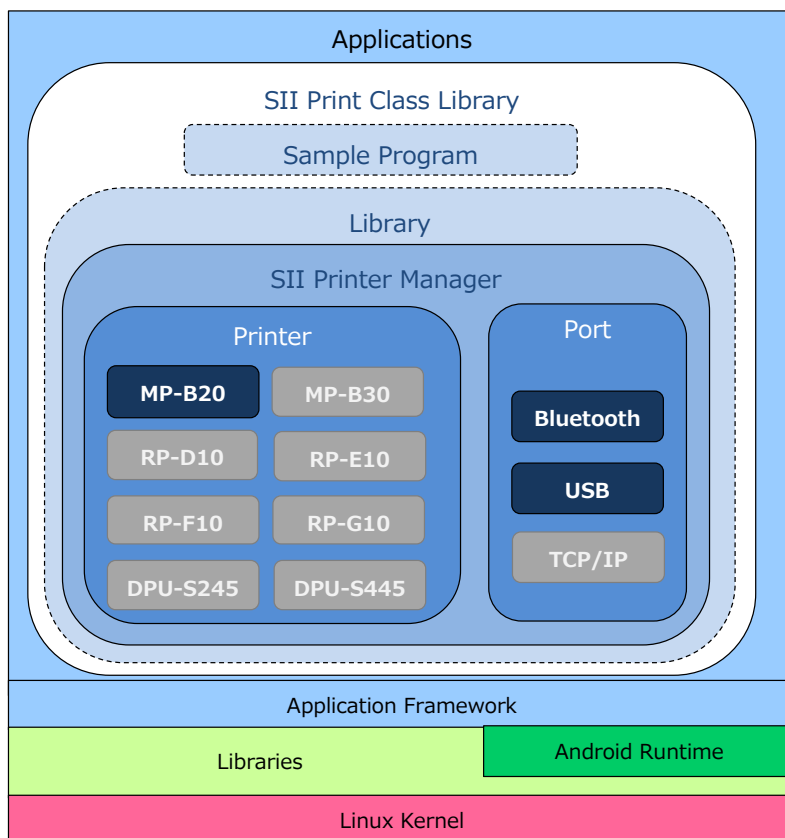
1.1 Functions Provided by SII Print Class Library

The SII print class library including the library and the sample programs provides the functions to use SII printer MP-B20 Series (hereinafter referred to as "printer") in Android applications. Moreover, the SII print class library provides Android Studio projects as a sample program for SII print class library.

1.2 SII Print Class Library Overview

1.2.1 SII Print Class Library Configuration

The library and sample program in SII print class library are indicated with dashed lines in the figure below.



1.2.2 Functions Provided by the Library

By using the library, Android applications can easily send print data and printer commands to a printer through communication port (Bluetooth or USB) on an Android device. Also, the applications can get printer status.

The library provides the following functions.

- Connecting to / disconnecting from a printer
- Sending data to a printer (print data and/or printer commands^{*1})
- Printing barcode and 2-dimensional barcode
- Sending a data file to a printer (print data and/or printer commands^{*1})
- Getting the printer status
- Aborting the waiting state of a printer
- Getting various responses from a printer
- Bulk registration of print commands
- Registering a printer status call back function
- Searching the printer by Bluetooth

*1: Commands that read responses from the printer are not available. In order to read responses from a printer, use `getStatus` or `getPrinterResponse`.

(NOTE) MP-B20 does not support the APIs of Display or the barcode scanner.

Chapter 2

Product Specifications

This chapter describes the product specifications of the library.

2.1 Operating Environment

Operating environment for the library is shown in the following table.

Printer	Model		MP-B20	
	F/W Version		1.00 or later	
	Communication Interface		Bluetooth	USB
Android Device	Communication Port		Bluetooth ^{*1}	USB ^{*2}
	OS	Android 5.0 (API 21)	Supported	Supported
		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)		
Supported Language		Japanese, English		

^{*1}: Bluetooth connection needs to be established by SPP (Serial Port Profile).

^{*2}: Android device needs to support USB host function.

2.2 Operating Conditions

This section describes the operating conditions for the library.

Set the Function Setting of the printer from [value] in the following table before using the library.

See "MP-B20 SERIES THERMAL PRINTER USER'S GUIDE" for details about Function Setting and the factory default settings.

MS	Function	Value
1-1	Interface Selection (Interface)	0 : USB 1 : Wireless
3-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable
3-2	Initialized Response Selection (Init. Response)	0 : Enable
3-3	Realtime Command Selection (Realtime Command)	0 : Enable
3-4	Data Discard Selection When Error Occurs (Error Through)	0 : Enable
3-5	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1 : Disable

Chapter 3

How to Use the Library

This chapter describes the development environment for Android application and how to use the library.

3.1 Development Environment for Android Application

In order to develop Android applications, the following tools are required. See each of the following URLs for more details.

- Android Studio
<https://developer.android.com/studio/index.html>
- USB driver for Windows (When develop on Windows environment)
<https://developer.android.com/studio/run/oem-usb.html>

In this chapter and after in this document, it is required to set up an environment where each tool is available.

3.2 Provided Files

The file configuration of SII print class library is as follows.

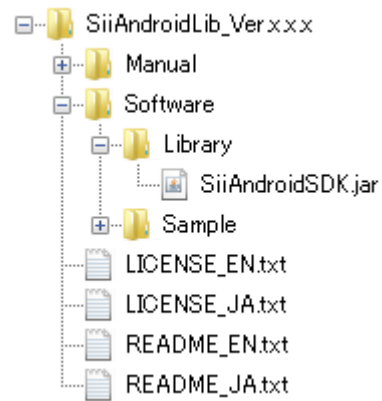


Figure 3-1

The file format of the library is jar. The file name of the library is SiiAndroidSDK.jar.

3.3 Add the Library to Android Studio Projects

Using the project of the sample program included in SII print class library as an example, this section describes how to add the library to Android Studio projects.

See "Chapter 5 Sample Program" for sample programs included in SII print class library.

- (1) Select and right click the module (app) displayed in the Android Project view of Android Studio, select [New] and [Directory] (Figure 3-2), and enter "libs" in the folder name to create a folder. (Figure 3-3)

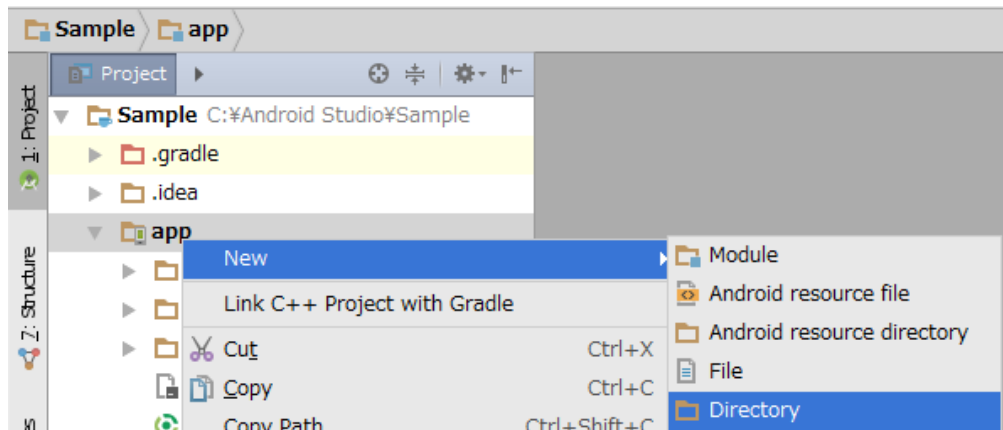


Figure 3-2

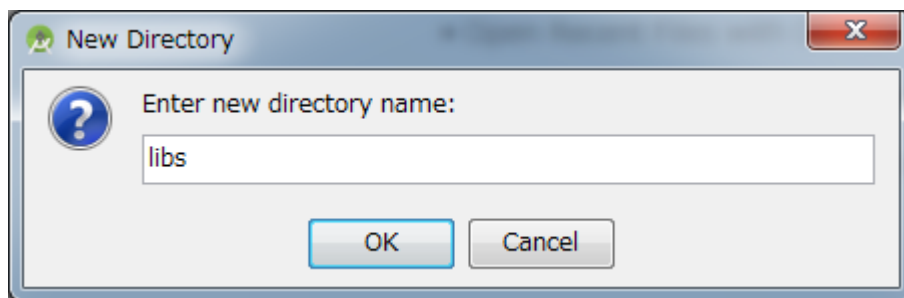


Figure 3-3

- (2) Copy the library file (SiiAndroidSDK.jar) into the folder (¥Sample¥app¥libs) created in step (1).

- (3) After adding the library, the view looks like Figure 3-4.

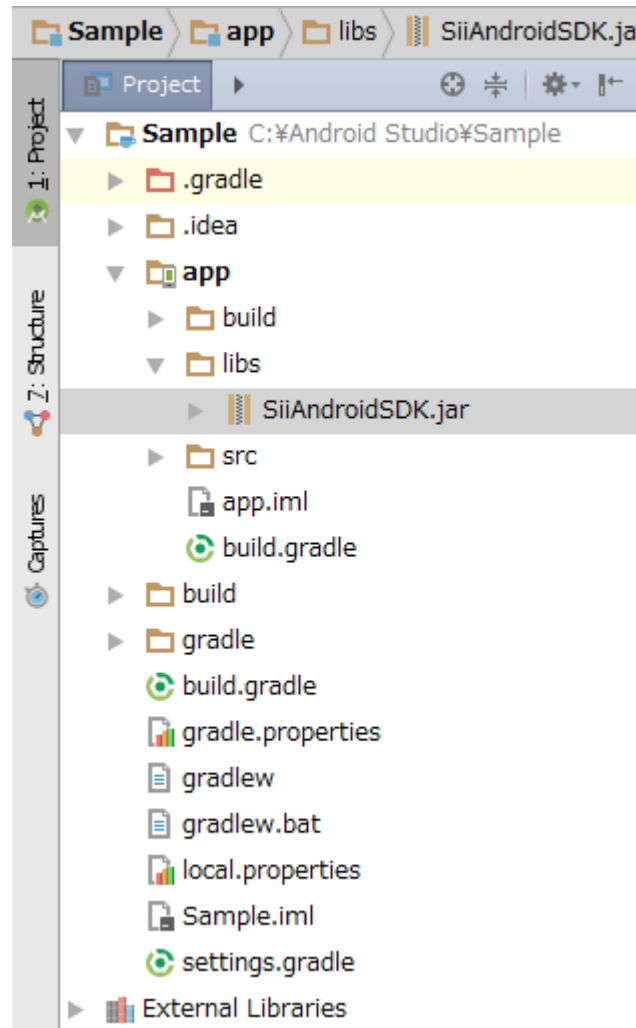


Figure 3-4

- (4) Add the following codes at the beginning of the main source file.
(Add at the beginning of MainActivity.java for the sample program.)

```
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;
```

By completing these procedures, functions of the library become available.

3.4 Use Developed Android Application on Android Device

In order to use the developed Android applications on the Android device, configure the following settings on the Android device.

(NOTE) This procedure is based on the menu of Android 7.1. Menu contents may vary depending on the using Android device.

- (1) Select [Settings], [Security], and turn on [Unknown sources]. (Figure 3-5)

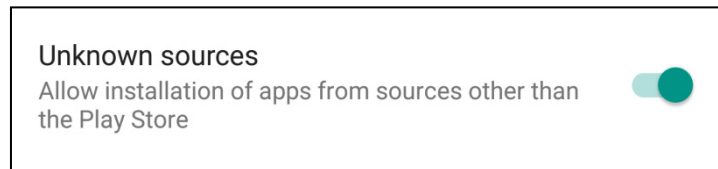


Figure 3-5

- (2) Select [Settings], [Developer options], and turn on [USB debugging]. (Figure 3-6)

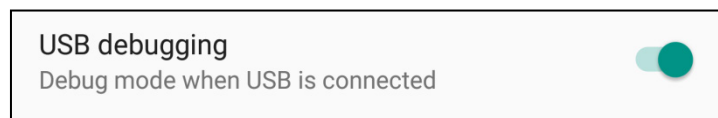


Figure 3-6

Chapter 4

Functions of the Library

This chapter describes the APIs of each class implemented in the library.

4.1 API Reference

The package of the library is `com.seikoinstruments.sdk.thermalprinter`.
`com.seikoinstruments.sdk.thermalprinter` includes the following classes.

Class Name	Description	Supported ^{*1}
CallbackFunctionListener	Interface for getting the change event of printer status. See "4.1.1 CallbackFunctionListener Interface " for more details.	✓
BarcodeScannerListener	Interface for getting barcode scanner connection or barcode scanner disconnection, or received barcode data. See "4.1.2 BarcodeScannerListener Interface " for more details.	-
PrinterManager	Provides the API used for communication with the printer and for printing. See "4.1.3 PrinterManager Class " for more details.	✓
PrinterEvent	Provides the API that gets the end event when <code>startDiscoveryPrinter</code> is terminated. See "4.1.4 PrinterEvent Class " for more details.	✓
PrinterListener	Interface for getting the end event when <code>startDiscoveryPrinter</code> is terminated. See "4.1.5 PrinterListener Interface " for more details.	✓
PrinterInfo	Stores the printer information found by <code>startDiscoveryPrinter</code> . See "4.1.6 PrinterInfo Class ".	✓
PrinterException	Exception class that is thrown at API call. See "4.1.7 PrinterException Class " for more details.	✓

*1: ✓: Supported, -: Not supported in MP-B20

(NOTE) MP-B20 does not support the APIs of Display or the barcode scanner.

4.1.1 CallbackFunctionListener Interface

CallbackFunctionListener Interface is an interface for getting the change event of printer status.

(1) Method List

Method of **CallbackFunctionListener** Interface is shown below.

Name	Description
onStatusChanged	Change event of printer status

(2) Method Details

onStatusChanged	Change event of printer status
------------------------	--------------------------------

Syntax `public void onStatusChanged(int status)`

Parameter *status* Printer status

Description This method is called at the following timing.
 ·When **setCallbackFunctionListener** is executed.
 ·When the printer status is changed.

The change event of printer status is notified when **isConnect** is true.

This method is an interface, so it is not implemented.
Implement the optional process in the class that receives a callback of the printer status change.

Do not execute the APIs of **PrinterManager** within this method.

4.1.2 BarcodeScannerListener Interface

BarcodeScannerListener Interface is an interface for getting barcode scanner connection, barcode scanner disconnection, or received barcode data.

MP-B20 does not support this interface.

4.1.3 PrinterManager Class

(1) Method List

Methods provided by the **PrinterManager** class are shown in the following table.

Name	Description	Supported ^{*1}
PrinterManager	Constructor	✓
connect	Start communicating with printer (Bluetooth)	✓
connect	Start communicating with printer (USB)	✓
disconnect	Stop communicating with printer	✓
sendText	Send text data	✓
sendTextEx	Send format specified text data	✓
printBarcode	Print barcode	✓
printPDF417	Print PDF417	✓
printQRcode	Print QR Code	✓
printDataMatrix	Print Data Matrix	✓
printMaxiCode	Print MaxiCode	✓
printGS1DataBarStacked	Print GS1 Databar Stacked	✓
printGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional	✓
printGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked	✓
cutPaper	Cut paper ^{*2}	✓
openDrawer	Open cash drawer	-
buzzer	Sound buzzer	-
externalBuzzer	Sound external buzzer	-
sendBinary	Send binary data	✓
sendDataFile	Send specified file	✓
getStatus	Get printer status	✓
setCallbackFunctionListener	Start/End callback of printer status change	✓
abort	Abort waiting state of printer	✓
registerLogo	Register logo	✓
printLogo	Print logo	✓
unregisterLogo	Delete registered logo	✓
registerStyleSheet	Register style sheet	-
unregisterStyleSheet	Delete registered style sheet	-
resetPrinter	Reset printer	✓
getPrinterResponse	Get various responses from printer	✓
startDiscoveryPrinter	Start printer search (Bluetooth)	✓
startDiscoveryPrinter	Start printer search (USB)	✓
startDiscoveryPrinter	Start printer search (TCP/IP)	-
cancelDiscoveryPrinter	Cancel printer search	✓
getFoundPrinter	Get found printer information	✓

Name	Description	Supported ^{*1}
getSendTimeout	Get send timeout period	✓
setSendTimeout	Set send timeout period	✓
getReceiveTimeout	Get receive timeout period	✓
setReceiveTimeout	Set receive timeout period	✓
getInternationalCharacter	Get international character set	✓
setInternationalCharacter	Set international character set	✓
getCodePage	Get codepage	✓
setCodePage	Set codepage	✓
getPrinterModel	Get printer model	✓
getPortType	Get connecting port type	✓
isConnect	Verify connection state with printer	✓
getSocketKeepingTime	Get socket keeping time	-
setSocketKeepingTime	Set socket keeping time	-

*1: ✓ : Supported, -: Not supported in MP-B20

*2: Only the paper feed operation to the paper cut position is performed.

(2) Constant List

① International character set

Constants used for setting/getting international character set are shown in the following table.

Constant Name	Description	Value
COUNTRY_USA	USA	0
COUNTRY_FRANCE	France	1
COUNTRY_GERMANY	Germany	2
COUNTRY_ENGLAND	United Kingdom	3
COUNTRY_DENMARK_1	Denmark I	4
COUNTRY_SWEDEN	Sweden	5
COUNTRY_ITALY	Italy	6
COUNTRY_SPAIN	Spain I	7
COUNTRY_JAPAN	Japan	8
COUNTRY_NORWAY	Norway	9
COUNTRY_DENMARK_2	Denmark II	10
COUNTRY_SPAIN_2	Spain II	11
COUNTRY_LATIN_AMERICA	Latin America	12
COUNTRY_ARABIA	Arabia	17

② Codepage

Constants used for setting/getting codepage are shown in the following table.

Constant Name	Codepage	Value
CODE_PAGE_437	USA, Standard Europe (Code Page437)	0
CODE_PAGE_KATAKANA	Katakana	1
CODE_PAGE_850	Multilingual (Code Page850)	2
CODE_PAGE_860	Portuguese (Code Page860)	3
CODE_PAGE_863	Canadian-French (Code Page863)	4
CODE_PAGE_865	Nordic (Code Page865)	5
CODE_PAGE_857	Turkish (Code Page857)	13
CODE_PAGE_737	Greek (Code Page737)	14
CODE_PAGE_1252	Latin (Code Page1252)	16
CODE_PAGE_866	Russian (Code Page866)	17
CODE_PAGE_852	Eastern Europe (Code Page852)	18
CODE_PAGE_858	Euro (Code Page858)	19
CODE_PAGE_855	Cyrillic (Code Page855)	34
CODE_PAGE_864^{*1*2}	Arabic (Code Page864)	37
CODE_PAGE_1250	Central European (Code Page1250)	45
CODE_PAGE_1251	Cyrillic (Code Page1251)	46
CODE_PAGE_1253	Greek (Code Page1253)	47
CODE_PAGE_1254	Turkish (Code Page1254)	48

*1: 20ACh of the Unicode cannot be printed.

*2: Font B cannot be printed.

③ Barcode or PDF417

Constants used for printing barcode and printing PDF417 are shown in the following table.

Constant Name	Description	Value
BARCODE_HEIGHT_DEFAULT	Default value of barcode height	162
PDF417_MODULE_HEIGHT_DEFAULT	Default value of PDF417 height	10
PDF417_ROW_AUTO	Automatic selection of the number of rows	0
PDF417_COLUMN_AUTO	Automatic selection of the number of columns	0

④ Printer model

Constants used for starting communicating with a printer and getting the printer model are shown in the following table.

Constant Name	Description	Value
PRINTER_MODEL_MP_B20	MP-B20	298
PRINTER_MODEL_DEFAULT	Default value of printer model	284

⑤ Port type

Constants used for starting communicating with a printer and getting the connecting port type are shown in the following table.

Constant Name	Description	Value
PRINTER_TYPE_BLUETOOTH	Bluetooth	0
PRINTER_TYPE_USB	USB	1

⑥ Response type

Constants used for getting various responses from a printer are shown in the following table.

Constant Name	Description	Value
PRINTER_RESPONSE_REQUEST	Request of execution response	0
PRINTER_RESPONSE_USER_AREA	Send remaining capacity response in user area	1
PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity response in user area after defragment	2
PRINTER_RESPONSE_NV_GRAPHICS	Send NV graphics memory capacity	3
PRINTER_RESPONSE_KEY_CODE	Send key code list of defined NV graphics	4
PRINTER_RESPONSE_BATTERY_STATUS	Battery remaining capacity level	5

⑦ Battery remaining capacity level

Constants of the battery remaining capacity level retrieved from a printer are shown in the following table.

Constant Name	Description	Value
BATTERY_STATUS_FULL	Battery remaining capacity: approx. 80%	0
BATTERY_STATUS_MIDDLE	Battery remaining capacity: approx. 40%	1
BATTERY_STATUS_LOW	Battery remaining capacity: approx. 10%	2
BATTERY_STATUS_EMPTY	No battery	3

(3) Enumerated Constant List

① Bold print (CharacterBold)

Constants of enumerated type used for bold print are shown in the following table.

Constant Name	Description
BOLD_CANCEL	Cancel bold print
BOLD	Specify bold print

② Underline (CharacterUnderline)

Constants of enumerated type used for underlining are shown in the following table.

Constant Name	Description
UNDERLINE_CANCEL	Cancel underline print
UNDERLINE_1	Specify 1-dot width underline print
UNDERLINE_2	Specify 2-dot width underline print

③ Reverse print (CharacterReverse)

Constants of enumerated type used for reverse print are shown in the following table.

Constant Name	Description
REVERSE_CANCEL	Cancel reverse print
REVERSE	Specify reverse print

④ Character font (CharacterFont)

Constants of enumerated type used for character font are shown in the following table.

Constant Name	Description
FONT_A	Font A (24×12)
FONT_B	Font B (16×8)

⑤ Character scale (CharacterScale)

Constants of enumerated type used for character scale are shown in the following table.

Constant Name	Description
VARTICAL_1_HORIZONTAL_1	Height × 1 and width × 1
VARTICAL_1_HORIZONTAL_2	Height × 1 and width × 2
VARTICAL_1_HORIZONTAL_3	Height × 1 and width × 3
VARTICAL_1_HORIZONTAL_4	Height × 1 and width × 4
VARTICAL_2_HORIZONTAL_1	Height × 2 and width × 1
VARTICAL_2_HORIZONTAL_2	Height × 2 and width × 2
VARTICAL_2_HORIZONTAL_3	Height × 2 and width × 3
VARTICAL_2_HORIZONTAL_4	Height × 2 and width × 4
VARTICAL_2_HORIZONTAL_6	Height × 2 and width × 6
VARTICAL_3_HORIZONTAL_1	Height × 3 and width × 1
VARTICAL_3_HORIZONTAL_2	Height × 3 and width × 2
VARTICAL_3_HORIZONTAL_3	Height × 3 and width × 3
VARTICAL_3_HORIZONTAL_4	Height × 3 and width × 4
VARTICAL_4_HORIZONTAL_1	Height × 4 and width × 1
VARTICAL_4_HORIZONTAL_2	Height × 4 and width × 2
VARTICAL_4_HORIZONTAL_3	Height × 4 and width × 3
VARTICAL_4_HORIZONTAL_4	Height × 4 and width × 4
VARTICAL_4_HORIZONTAL_6	Height × 4 and width × 6
VARTICAL_4_HORIZONTAL_8	Height × 4 and width × 8
VARTICAL_6_HORIZONTAL_2	Height × 6 and width × 2
VARTICAL_6_HORIZONTAL_4	Height × 6 and width × 4
VARTICAL_6_HORIZONTAL_6	Height × 6 and width × 6
VARTICAL_6_HORIZONTAL_8	Height × 6 and width × 8
VARTICAL_8_HORIZONTAL_4	Height × 8 and width × 4
VARTICAL_8_HORIZONTAL_6	Height × 8 and width × 6
VARTICAL_8_HORIZONTAL_8	Height × 8 and width × 8

⑥ Alignment (PrintAlignment)

Constants of enumerated type used for alignment are shown in the following table.

Constant Name	Description
ALIGNMENT_LEFT	Aligned left
ALIGNMENT_CENTER	Centered
ALIGNMENT_RIGHT	Aligned right

⑦ Barcode symbol (BarcodeSymbol)

Constants of enumerated type used for barcode symbol are shown in the following table.

Constant Name	Description	Syntax ^{*1}
BARCODE_SYMBOL_UPC_A	UPC-A	(a)
BARCODE_SYMBOL_UPC_E	UPC-E	(a)
BARCODE_SYMBOL_EAN13	EAN13	(a)
BARCODE_SYMBOL_JAN13	JAN13	(a)
BARCODE_SYMBOL_EAN8	EAN8	(a)
BARCODE_SYMBOL_JAN8	JAN8	(a)
BARCODE_SYMBOL_CODE39	CODE39	(a), (b)
BARCODE_SYMBOL_CODE93	CODE93	(c)
BARCODE_SYMBOL_CODE128	CODE128	(c)
BARCODE_SYMBOL_ITF	ITF	(a), (b)
BARCODE_SYMBOL_CODABAR	CODABAR	(a), (b)
BARCODE_SYMBOL_EAN13_ADDON	EAN13 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: See "4.1.3(4) Method Details printBarcode" for more details of syntax.

⑧ Module size (ModuleSize)

Constants of enumerated type used for width, nominal fine element width, and module size of barcode are shown in the following table.

Constant Name	Description	Method to Use
BARCODE_MODULE_WIDTH_2	Fine element 2 dots Module width 0.250 mm	printBarcode
BARCODE_MODULE_WIDTH_3	Fine element 3 dots Module width 0.375 mm	
BARCODE_MODULE_WIDTH_4	Fine element 4 dots Module width 0.500 mm	
BARCODE_MODULE_WIDTH_5	Fine element 5 dots Module width 0.625 mm	
BARCODE_MODULE_WIDTH_6	Fine element 6 dots Module width 0.750 mm	
PDF417_MODULE_WIDTH_2	Nominal fine element width 2 dots	printPDF417
PDF417_MODULE_WIDTH_3	Nominal fine element width 3 dots	

Constant Name	Description	Method to Use
PDF417_MODULE_WIDTH_4	Nominal fine element width 4 dots	printPDF417
PDF417_MODULE_WIDTH_5	Nominal fine element width 5 dots	
PDF417_MODULE_WIDTH_6	Nominal fine element width 6 dots	
PDF417_MODULE_WIDTH_7	Nominal fine element width 7 dots	
PDF417_MODULE_WIDTH_8	Nominal fine element width 8 dots	
QR_MODULE_SIZE_2	2 dots	printQRcode
QR_MODULE_SIZE_3	3 dots	
QR_MODULE_SIZE_4	4 dots	
QR_MODULE_SIZE_5	5 dots	
QR_MODULE_SIZE_6	6 dots	
QR_MODULE_SIZE_7	7 dots	
QR_MODULE_SIZE_8	8 dots	
QR_MODULE_SIZE_9	9 dots	
QR_MODULE_SIZE_10	10 dots	
QR_MODULE_SIZE_11	11 dots	
QR_MODULE_SIZE_12	12 dots	
QR_MODULE_SIZE_13	13 dots	
QR_MODULE_SIZE_14	14 dots	
QR_MODULE_SIZE_15	15 dots	
QR_MODULE_SIZE_16	16 dots	
DATAMATRIX_MODULE_SIZE_2	2 dots	printDataMatrix
DATAMATRIX_MODULE_SIZE_3	3 dots	
DATAMATRIX_MODULE_SIZE_4	4 dots	
DATAMATRIX_MODULE_SIZE_5	5 dots	
DATAMATRIX_MODULE_SIZE_6	6 dots	
DATAMATRIX_MODULE_SIZE_7	7 dots	
DATAMATRIX_MODULE_SIZE_8	8 dots	
DATAMATRIX_MODULE_SIZE_9	9 dots	
DATAMATRIX_MODULE_SIZE_10	10 dots	
DATAMATRIX_MODULE_SIZE_11	11 dots	
DATAMATRIX_MODULE_SIZE_12	12 dots	
DATAMATRIX_MODULE_SIZE_13	13 dots	
DATAMATRIX_MODULE_SIZE_14	14 dots	
DATAMATRIX_MODULE_SIZE_15	15 dots	
DATAMATRIX_MODULE_SIZE_16	16 dots	

Constant Name	Description	Method to Use
GS1DATABAR_MODULE_SIZE_2	2 dots	<ul style="list-style-type: none"> • printGS1DataBarStacked • printGS1DataBarStackedOmnidirectional • printGS1DataBarExpandedStacked
GS1DATABAR_MODULE_SIZE_3	3 dots	
GS1DATABAR_MODULE_SIZE_4	4 dots	
GS1DATABAR_MODULE_SIZE_5	5 dots	
GS1DATABAR_MODULE_SIZE_6	6 dots	
GS1DATABAR_MODULE_SIZE_7	7 dots	
GS1DATABAR_MODULE_SIZE_8	8 dots	
GS1DATABAR_MODULE_SIZE_9	9 dots	
GS1DATABAR_MODULE_SIZE_10	10 dots	
GS1DATABAR_MODULE_SIZE_11	11 dots	
GS1DATABAR_MODULE_SIZE_12	12 dots	
GS1DATABAR_MODULE_SIZE_13	13 dots	
GS1DATABAR_MODULE_SIZE_14	14 dots	
GS1DATABAR_MODULE_SIZE_15	15 dots	
GS1DATABAR_MODULE_SIZE_16	16 dots	

⑨ HRI character print position (HriPosition)

Constants of enumerated type used for HRI character print position are shown in the following table.

Constant Name	Description
HRI_NONE	Not printed
HRI_POSITION_ABOVE	Above barcode
HRI_POSITION_BELOW	Below barcode
HRI_POSITION_ABOVE_BELOW	Above and below barcode (both)

⑩ N:W ratio (NwRatio)

Constants of enumerated type used for N:W ratio are shown in the following table.

Constant Name	Description
NWRATIO_1TO2	1:2
NWRATIO_1TO2_5	1:2.5
NWRATIO_1TO3	1:3

⑪ Error correction level (ErrorCorrection)

Constants of enumerated type used for error correction level are shown in the following table.

Constant Name	Description	Method to Use
PDF417_ERROR_CORRECTION_0	Error correction level 0	printPDF417
PDF417_ERROR_CORRECTION_1	Error correction level 1	
PDF417_ERROR_CORRECTION_2	Error correction level 2	
PDF417_ERROR_CORRECTION_3	Error correction level 3	
PDF417_ERROR_CORRECTION_4	Error correction level 4	
PDF417_ERROR_CORRECTION_5	Error correction level 5	
PDF417_ERROR_CORRECTION_6	Error correction level 6	
PDF417_ERROR_CORRECTION_7	Error correction level 7	
PDF417_ERROR_CORRECTION_8	Error correction level 8	
QR_ERROR_CORRECTION_L	Error correction level L	printQRcode
QR_ERROR_CORRECTION_M	Error correction level M	
QR_ERROR_CORRECTION_H	Error correction level H	
QR_ERROR_CORRECTION_Q	Error correction level Q	

⑫ PDF417 symbol (Pdf417Symbol)

Constants of enumerated type used for PDF417 symbol are shown in the following table.

Constant Name	Description
PDF417_STANDARD	PDF417
PDF417_COMPACT	Compact PDF417

⑬ QR Code Model (QrModel)

Constants of enumerated type used for QR Code Model are shown in the following table.

Constant Name	Description
QR_MODEL_1	QR Code Model 1
QR_MODEL_2	QR Code Model 2

⑭ Cutting method (CuttingMethod)

Constants of enumerated type used for cutting method are shown in the following table. Whichever constant of enumerated type is specified, the printer only feeds the paper to the paper cut position without cutting.

Constant Name	Description
CUT_FULL	Full cut
CUT_PARTIAL	Partial cut

⑮ Data Matrix module (DataMatrixModule)

Constants of enumerated type used for Data Matrix module are shown in the following table.

Constant Name	Description
DATA_MATRIX_AUTO	Number of modules: Automatic
DATA_MATRIX_10_10	Number of modules: 10×10
DATA_MATRIX_12_12	Number of modules: 12×12
DATA_MATRIX_14_14	Number of modules: 14×14
DATA_MATRIX_16_16	Number of modules: 16×16
DATA_MATRIX_18_18	Number of modules: 18×18
DATA_MATRIX_20_20	Number of modules: 20×20
DATA_MATRIX_22_22	Number of modules: 22×22
DATA_MATRIX_24_24	Number of modules: 24×24
DATA_MATRIX_26_26	Number of modules: 26×26
DATA_MATRIX_32_32	Number of modules: 32×32
DATA_MATRIX_36_36	Number of modules: 36×36
DATA_MATRIX_40_40	Number of modules: 40×40
DATA_MATRIX_44_44	Number of modules: 44×44
DATA_MATRIX_48_48	Number of modules: 48×48
DATA_MATRIX_52_52	Number of modules: 52×52
DATA_MATRIX_64_64	Number of modules: 64×64
DATA_MATRIX_72_72	Number of modules: 72×72
DATA_MATRIX_80_80	Number of modules: 80×80
DATA_MATRIX_88_88	Number of modules: 88×88
DATA_MATRIX_96_96	Number of modules: 96×96
DATA_MATRIX_104_104	Number of modules: 104×104
DATA_MATRIX_120_120	Number of modules: 120×120
DATA_MATRIX_132_132	Number of modules: 132×132
DATA_MATRIX_144_144	Number of modules: 144×144
DATA_MATRIX_8_18	Number of modules: 8×18
DATA_MATRIX_8_32	Number of modules: 8×32
DATA_MATRIX_12_26	Number of modules: 12×26
DATA_MATRIX_12_36	Number of modules: 12×36

Constant Name	Description
DATA_MATRIX_16_36	Number of modules: 16×36
DATA_MATRIX_16_48	Number of modules: 16×48

⑩ MaxiCode Mode (MaxiCodeMode)

Constants of enumerated type used for MaxiCode Mode are shown in the following table.

Constant Name	Description
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

⑪ Dithering (Dithering)

Constants of enumerated type used for dithering are shown in the following table.

Constant Name	Description
DITHERING_DISABLE	Dithering is disabled
DITHERING_ERRORDIFFUSION	Dithering is enabled

⑫ Batch processing selection (TransactionFunction)

Constants of enumerated type used for batch processing selection are shown in the following table.

Constant Name	Description
TRANSACTION_CLEAR	Cancel batch processing
TRANSACTION_START	Start batch processing
TRANSACTION_PRINT	Finish batch printing and batch processing

⑬ Pending data output specifying (OutputPendingData)

Constants of enumerated type used for pending data output specifying are shown in the following table.

Constant Name	Description
PENDING_DATA_OUTPUT_FIRST	Output pending data at first and start the processing
PENDING_DATA_OUTPUT_TOGETHER	Output pending data at the same time as the processing

PrinterManager**Constructor**

Constructor for **com.seikoinstruments.sdk.thermalprinter.PrinterManager** class.

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

(b) public **PrinterManager**()

Parameter *context* Specify application context to call this method.
Example: **MainActivity.this**

Note Use the syntax (a) when using this method newly.
Syntax (b) is a method that will be unsupported in the future.

connect**Start communicating with printer (Bluetooth)**

Starts communication with a printer by Bluetooth connection.

The method of syntax (a) always communicates with a printer in secure mode.

The method of syntax (b) communicates with a printer by specifying secure mode or insecure mode.

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

Parameter *printerModel* Printer model constant for Bluetooth connection
See "4.1.3(2)④ Printer model" for available constants.

address Bluetooth address
Example: "00:11:22:AA:BB:CC"

secure true Communicates with a printer in secure mode
false Communicates with a printer in insecure mode
Normally, communication in secure mode is recommended.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.

Description Call this method before using other **PrinterManager** class methods.

The printer specified by *printerModel* is connected to the Bluetooth address specified by *address*.

Also, printer initial setting is performed at the connection based on *printerModel* specified.

Monitoring of the printer status is started with this method. The latest printer status can be retrieved from **getStatus**.

Changes of the printer status can be notified as events by **onStatusChanged** and **setCallbackFunctionListener**.

Starts communication with a printer by USB connection.

The method of syntax (a) specifies the printer model.

The method of syntax (b) specifies the printer model and specifies the context to call the application.

Syntax	(a) public void connect (int <i>printerModel</i>) throws PrinterException	
	(b) public void connect (int <i>printerModel</i> , Context <i>context</i>) throws PrinterException	
Parameter	<i>printerModel</i>	Printer model constant for USB connection See "4.1.3(2)④ Printer model" for available constants.
	<i>context</i>	Specify application context to call this method. Example: MainActivity.this
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.7 PrinterException Class " for details of the error.	
Description	Call this method before using other PrinterManager class methods. The printer specified by <i>printerModel</i> is connected. Also, printer initial setting is performed at the connection based on <i>printerModel</i> specified. Monitoring of the printer status is started with this method. The latest printer status can be retrieved from getStatus . Changes of the printer status can be notified as events by onStatusChanged and setCallbackFunctionListener .	
Note	Use the syntax (a) when using this method newly. Syntax (b) is a method that will be unsupported in the future.	

Stops communicating with the printer and monitoring the printer status.

Syntax	public void disconnect () throws PrinterException	
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.7 PrinterException Class " for details of the error.	
Description	This method discards the print data kept by controlTransaction .	
Note	It is recommended to get the execution response by PRINTER_RESPONSE_REQUEST of getPrinterResponse method before executing this method. If not, the communication is disconnected by this method before the print data sending from Android device to the printer is completed, and a part of the data may be lost. If you do not execute getPrinterResponse in your program, evaluate your program to confirm no problems arise.	

sendText	Send text data
-----------------	----------------

sendText	Send text data
----------	----------------

Sends text data.

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

Parameter	<i>text</i>	Text data to send to the printer Data size that can be specified at 1 time is 16 KB (16384 bytes).
-----------	-------------	---

Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.7 PrinterException Class" for details of the error.</p> <p>When the data transmission is failed, the communication with the printer is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the printer.</p>
-----------	--

Description	This method encodes the specified text data to printable text data based on <code>setInternationalCharacter</code> and <code>setCodePage</code> , and then sends it to the printer.
-------------	---

This method does not add any line feed code at the end of the text data. In order to print to the end, add a line feed code to the end of the text data.

sendTextEx	Send format specified text data
-------------------	---------------------------------

sendTextEx	Send format specified text data
-------------------	---------------------------------

Sends format specified text data to the printer.

The method of syntax (a) outputs the pending data at first and starts processing.

The method of syntax (b) starts processing according to the constants of the pending data output specifying.

Syntax (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**

Parameter	<i>text</i>	Text data to send to the printer Data size that can be specified at 1 time is 16 KB (16384 bytes).
-----------	-------------	---

bold Bold print
See "4.1.3(3)① Bold print (CharacterBold)" for available constants.

underline Underline
See "4.1.3(3)② Underline (CharacterUnderline)" for available constants.

reverse Reverse print
See "4.1.3(3)③ Reverse print (CharacterReverse)" for available constants.

<i>font</i>	Font See "4.1.3(3)④ Character font (CharacterFont)" for available constants.
<i>scale</i>	Character scale See "4.1.3(3)⑤ Character scale (CharacterScale)" for available constants.
<i>alignment</i>	Alignment See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.
<i>output</i>	Pending data output specifying See "4.1.3(3)⑱ Pending data output specifying (OutputPendingData)" for available constants.

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling this method.

See "4.1.7 **PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description This method encodes the specified text data to printable text data based on **setInternationalCharacter** and **setCodePage**, and then sends it to the printer.

For laying out text data by sending following printer commands with **sendBinary** or **sendDataFile**, specify **PENDING_DATA_OUTPUT_TOGETHER** at *output* in the method of syntax (b).

- "Horizontal Tab"
- "Specify Absolute Position"
- "Specify Relative Position"

When the method of the syntax (a) is executed or **PENDING_DATA_OUTPUT_FIRST** is specified at *output* in the method of syntax (b), the print position set in above becomes invalid.

This method does not add any line feed code at the end of the text data. In order to print to the end, add a line feed code to the end of the text data.

printBarcode

Print barcode

Prints barcode.

The method of syntax (a) specifies the barcode data by character string.

The method of syntax (b) specifies the barcode data by character string and specifies the alignment and N:W ratio of the barcode.

The method of syntax (c) specifies the barcode data by the array of bytes and specifies the alignment of the barcode.

Syntax (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**

Parameter	<i>barcodeSymbol</i>	Barcode symbol See "4.1.3(3)⑦ Barcode symbol (BarcodeSymbol)" for available constants and corresponding syntax.
	<i>text (data)</i>	Barcode data to send to the printer
	<i>moduleSize</i>	Barcode width See "4.1.3(3)⑧ Module size (ModuleSize)" for available constants.
	<i>moduleHeight</i>	Barcode height (dot) <ul style="list-style-type: none"> When <i>barcodeSymbol</i> is set to the following, the valid range is 1 to 255. 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

- When *barcodeSymbol* is set to the following, the valid range is different by *barcodeSymbol* and *moduleSize*.

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	Valid Range
BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL		
	BARCODE_MODULE_WIDTH_2	66 to 255
	BARCODE_MODULE_WIDTH_3	99 to 255
	BARCODE_MODULE_WIDTH_4	132 to 255
	BARCODE_MODULE_WIDTH_5	165 to 255
	BARCODE_MODULE_WIDTH_6	198 to 255
BARCODE_SYMBOL_GS1_TRUNCATED		
	BARCODE_MODULE_WIDTH_2	26 to 255
	BARCODE_MODULE_WIDTH_3	39 to 255
	BARCODE_MODULE_WIDTH_4	52 to 255
	BARCODE_MODULE_WIDTH_5	65 to 255
	BARCODE_MODULE_WIDTH_6	78 to 255
BARCODE_SYMBOL_GS1_LIMITED		
	BARCODE_MODULE_WIDTH_2	20 to 255
	BARCODE_MODULE_WIDTH_3	30 to 255
	BARCODE_MODULE_WIDTH_4	40 to 255
	BARCODE_MODULE_WIDTH_5	50 to 255
	BARCODE_MODULE_WIDTH_6	60 to 255
BARCODE_SYMBOL_GS1_EXPANDED		
	BARCODE_MODULE_WIDTH_2	68 to 255
	BARCODE_MODULE_WIDTH_3	102 to 255
	BARCODE_MODULE_WIDTH_4	136 to 255
	BARCODE_MODULE_WIDTH_5	170 to 255
	BARCODE_MODULE_WIDTH_6	204 to 255

<i>hriPosition</i>	HRI character print position See "4.1.3(3)⑨ HRI character print position (HriPosition)" for available constants.
<i>hriFont</i>	HRI character font See "4.1.3(3)④ Character font (CharacterFont)" for available constants.
<i>alignment</i>	Alignment See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.
<i>nwRatio</i>	N:W ratio See "4.1.3(3)⑩ N:W ratio (NwRatio)" for available constants. Depending on specified <i>nwRatio</i> and <i>moduleSize</i> , the wide element width is set in the following table.

<i>moduleSize</i>	<i>nwRatio</i>		
	NWRATIO_ 1TO2	NWRATIO_ 1TO2_5	NWRATIO_ 1TO3
BARCODE_MODULE_WIDTH_2	0.500 mm (4 dots)	0.625 mm (5 dots)	0.750 mm (6 dots)
BARCODE_MODULE_WIDTH_3	0.750 mm (6 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)
BARCODE_MODULE_WIDTH_4	1.000 mm (8 dots)	1.250 mm (10 dots)	1.500 mm (12 dots)
BARCODE_MODULE_WIDTH_5	1.250 mm (10 dots)	1.625 mm (13 dots)	1.875 mm (15 dots)
BARCODE_MODULE_WIDTH_6	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling this method.

See "4.1.7 **PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printPDF417

Print PDF417

Prints PDF417.

The method of syntax (a) specifies PDF417 symbol.

The method of syntax (b) is fixed to standard PDF417.

Syntax (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**

Parameter *text* Barcode data to send to the printer

errorCorrection Error correction level
See "4.1.3(3)① Error correction level (ErrorCorrection)" for available constants.

row The number of rows (row)
The valid range is 0 to 90.
When 0 is specified, the number of rows is automatically set.

<i>column</i>	The number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
<i>moduleSize</i>	Nominal fine element width See "4.1.3(3)⑧ Module size (ModuleSize)" for available constants.
<i>moduleHeight</i>	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
<i>alignment</i>	Alignment See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.
<i>pdf417Symbol</i>	Symbol of PDF417 See "4.1.3(3)⑫ PDF417 symbol (Pdf417Symbol)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printQRcode Print QR Code

Prints QR Code.
The method of syntax (a) specifies QR Code Model.
The method of syntax (b) is fixed to QR Code Model 2.

Syntax	(a) public void printQRcode (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> , QrModel <i>model</i>) throws PrinterException
	(b) public void printQRcode (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i>) throws PrinterException
Parameter	<i>text</i> Barcode data to send to the printer The version for either syntax (a) or (b) is automatically set depending on the number of data specified on <i>text</i> .
	<i>errorCorrection</i> Error correction level See "4.1.3(3)⑪ Error correction level (ErrorCorrection)" for available constants.
	<i>moduleSize</i> Module size See "4.1.3(3)⑧ Module size (ModuleSize)" for available constants.
	<i>alignment</i> Alignment See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.
	<i>model</i> QR Code Model See "4.1.3(3)⑬ QR Code Model (QrModel)" for available constants.

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling this method.

See "4.1.7 PrinterException Class" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printDataMatrix

Print Data Matrix

Prints Data Matrix.

Syntax	<pre>public void printDataMatrix(String <i>text</i>, DataMatrixModule <i>dataMatrixModule</i>, ModuleSize <i>moduleSize</i>, PrintAlignment <i>alignment</i>) throws PrinterException</pre>
--------	--

Parameter	<i>text</i>	Barcode data to send to the printer
	<i>dataMatrixModule</i>	The number of Data Matrix modules See "4.1.3(3)⑮ Data Matrix module (DataMatrixModule)" for available constants.
	<i>moduleSize</i>	Module size See "4.1.3(3)⑧ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling this method.

See "4.1.7 PrinterException Class" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printMaxiCode

Print MaxiCode

Prints MaxiCode.

Syntax	<pre>public void printMaxiCode(String <i>text</i>, MaxiCodeMode <i>maxiCodeMode</i>, PrintAlignment <i>alignment</i>) throws PrinterException</pre>
--------	---

Parameter	<i>text</i>	Barcode data to send to the printer
		<ul style="list-style-type: none"> • When <i>maxiCodeMode</i> is MAXI_CODE_2 Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data. • When <i>maxiCodeMode</i> is MAXI_CODE_3 Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.
	<i>maxiCodeMode</i>	MaxiCode Mode See "4.1.3(3)⑩ MaxiCode Mode (MaxiCodeMode)" for available constants.
	<i>alignment</i>	Alignment See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printGS1DataBarStacked	Print GS1 Databar Stacked
-------------------------------	----------------------------------

Prints GS1 Databar Stacked.

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

Parameter *text* Barcode data to send to the printer
Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.

moduleSize Module size
See "4.1.3(3)⑧ Module size (ModuleSize)" for available constants.

alignment Alignment
See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional
--	---

Prints GS1 Databar Stacked Omni-directional.

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

Parameter *text* Barcode data to send to the printer
Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.

moduleHeight Barcode module height (the number of the modules)
The valid range is 33 to 255.

moduleSize Module size
See "4.1.3(3)⑧ Module size (ModuleSize)" for available constants.

alignment Alignment
See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.

openDrawer**Open cash drawer**

This method is not supported. When executing this method, **PrinterException** is thrown.

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

buzzer**Sound buzzer**

This method is not supported. When executing this method, **PrinterException** is thrown.

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

externalBuzzer**Sound external buzzer**

This method is not supported. When executing this method, **PrinterException** is thrown.

Syntax public void **externalBuzzer**(BuzzerPattern *buzzer pattern*, int *buzzerCount*)
throws **PrinterException**

sendBinary**Send binary data**

Sends binary data to the printer.

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

Parameter *binary* Binary data to send to the printer
Data size that can be specified at 1 time is 16 KB (16384 bytes).

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description This method sends the specified binary data to the printer without conversion.

By sending printer commands as binary data with this method, printer functions which are not supported in the library become available. However, this method does not support commands which get responses from the printer.

sendDataFile**Send specified file**

Sends file data.

The method of syntax (a), dithering can be specified.

The method of syntax (b), dithering is fixed to be disabled.

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

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

Parameter	<i>fileName</i>	<p>Name of the data file to send to the printer</p> <p>The maximum file size that can be specified is 1 MB (1048576 bytes).</p> <p>The file extensions that can be sent and the file transmission are described below.</p> <ul style="list-style-type: none"> • .bmp, .jpg, .jpeg, .png Data is sent to the printer as image data. Colored image data is converted to monochrome image by binarization and sent to the printer. Printing is performed at one time after mapping the image data in memory of the printer. • .txt Data is sent to the printer as text data. Text data format supports UTF-8. This method encodes the text data to printable text data based on the settings of setInternationalCharacter and setCodePage, and then sends it to the printer. This method does not add a line feed code at the end of the text data. In order to print to the end, add a line feed code to the end of the text data. • .bin, .dat Data is sent to the printer as the binary data without conversion.
	<i>alignment</i>	<p>Alignment</p> <p>The alignment is valid only when the file extension specified on <i>fileName</i> is .bmp, .jpg, .jpeg, .png, or .txt. See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.</p>
	<i>dithering</i>	<p>Dithering</p> <p>The dithering is valid only when the file extension specified on <i>fileName</i> is .bmp, .jpg, .jpeg, or .png. See "4.1.3(3)⑦ Dithering (Dithering)" for available constants.</p>
Exception	PrinterException	<p>PrinterException is thrown when an error occurs while calling this method. See "4.1.7 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the printer is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the printer.</p>

getStatus		Get printer status
<p>Gets the latest printer status.</p> <p>The method of syntax (a) returns the printer status with return value.</p> <p>The method of syntax (b) stores the printer status in an array of int type.</p>		
Syntax	<p>(a) public int getStatus() throws PrinterException</p> <p>(b) public void getStatus(int [] buf) throws PrinterException</p>	
Return value	Status retrieved from the printer	
Parameter	<i>buf</i>	Status retrieved from the printer
Exception	PrinterException	<p>PrinterException is thrown when an error occurs while calling this method. See "4.1.7 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the printer is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the printer.</p>

Description Status retrieved from the printer is stored to an integer array.

The printer status is shown below.

When the connection failed, the printer status is shown in 0x80000000.

Bit	Function	Value	
		0	1
0	Voltage error	No error	Error
1	Hardware error	No error	Error
2	Head temperature error	No error	Error
3	Reserved	Fixed	-
4	Out-of-paper error	No error	Error
5	Reserved	Fixed	-
6	Reserved	Fixed	-
7	Reserved	Fixed	-
8	FEED Switch status	OFF	ON
9	Reserved	Fixed	-
10	Paper feed status	Stop	Operating
11	Return-waiting status	Not waiting	Waiting
12	Reserved	Fixed	-
13	Reserved	-	Fixed
14	Reserved	-	Fixed
15	Reserved	-	Fixed
16	FLASH memory rewriting	Not rewriting	Rewriting
17	Reserved	-	Fixed
18	Reserved	-	Fixed
19	Reserved	-	Fixed
20 to 22	Battery remaining capacity level	000: No battery 001: Low (Battery remaining capacity: approx. 10%) 011: Middle (Battery remaining capacity: approx. 40%) 111: Full (Battery remaining capacity: approx. 80%)	
23	Battery error	No error	Error
24 to 31	Reserved	-	Fixed

setCallbackFunctionListener

Start/End callback of printer status change

Starts or ends callback to be executed according to changes of the printer status.

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

Parameter *listener* Instance of **CallbackFunctionListener** interface

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
 See "4.1.7 **PrinterException Class**" for details of the error.

Description Register the process executed by callback with **onStatusChanged**.
 When specify the instance of **CallbackFunctionListener** interface in *listener* and execute this method, the callback is started.
 When specify null in *listener* and execute this method, the callback is stopped.

The instance kept by **CallbackFunctionLister** interface is discarded by any of the following:

- Execute this method specifying null in *listener*
- Execute **disconnect**

This call of the method can be used when **connect** is executed and **isConnect** is true.

abort

Abort waiting state of printer

Aborts the waiting state of the printer.

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

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling this method.

See "4.1.7 **PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description When sending of image data by **sendDataFile** is aborted, the printer does not accept other processes until the specified image data is received completely. (Methods and transmission data are misinterpreted and recognized as a part of the image data.) To solve this situation, use this method to abort the waiting state of the printer.
 Note that when executing this method, a part of unprinted image data may be printed.

registerLogo

Register logo

Registers image data to NV graphics memory in the printer as a logo.

The method of syntax (a), dithering can be specified.

The method of syntax (b), dithering is fixed to be disabled.

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

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

Parameter *fileName* File name of image data to register as a logo
 The file extensions for supporting image data are .bmp, .jpg, .jpeg, and .png. When the image data is colored, it is converted to monochrome image by binarization and registered.

id Logo ID to register (key code)
 Specify the logo ID to register by character string of 2 characters.
 The valid characters are ASCII character code from 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z').

dithering Dithering
 See "4.1.3(3)⑪ Dithering (Dithering)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

printLogo	Print logo
------------------	-------------------

Prints the registered logo.

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

Parameter *id* Logo ID to be printed (key code)
Specify the ID of the registered logo as a character string.

alignment Alignment
See "4.1.3(3)⑥ Alignment (PrintAlignment)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

unregisterLogo	Delete registered logo
-----------------------	-------------------------------

Deletes the registered logo.

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

Parameter *id* Logo ID to delete (key code)
Specify the ID of the registered logo as a character string.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

registerStyleSheet	Register style sheet
---------------------------	-----------------------------

This method is not supported. When executing this method, **PrinterException** is thrown.

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

unregisterStyleSheet	Delete registered style sheet
-----------------------------	--------------------------------------

This method is not supported. When executing this method, **PrinterException** is thrown.

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

Resets the printer hardware.

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

Exception **PrinterException**

PrinterException is thrown when an error occurs while calling this method.

See "4.1.7 **PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description For Bluetooth connection:

The printer hardware reset is performed by the printer command "Printer Reset".

For USB connection:

The printer reset is performed by using the SOFT_RESET function in USB printer class.

getPrinterResponse

Get various responses from printer

Gets response data from the printer.

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

Parameter *id*

Response type constant

See "4.1.3(2)⑥ Response type" for available constants.

buf

Buffer that stores the retrieved response data

This method stores the response data specified by *id* to the object specified by *buf*.

The buffer type varies depending on the response type constant. See the following table for buffer types.

Response Type Constant	
Parameter	Description
PRINTER_RESPONSE_REQUEST (Execution response request)	
<i>buf</i>	Specify an int type array of length 1. Specify 0 to 15 (00h to 0Fh) for <i>buf</i> [0]. When the response is retrieved successfully, the response code of the execution response request is stored to <i>buf</i> [0] with 128 to 143 (80h to 8Fh).
PRINTER_RESPONSE_USER_AREA (Send remaining capacity of user area)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the remaining capacity of the user area is stored as a numerical value in bytes.
PRINTER_RESPONSE_ARRANGE_USER_AREA (Send remaining user area response after defragment)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the remaining capacity of the user area after defragment is stored as a numerical value in bytes.

Response Type Constant	
Parameter	Description
PRINTER_RESPONSE_NV_GRAPHICS (Send NV graphics memory capacity)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the NV graphics memory capacity is stored as a numerical value in bytes.
PRINTER_RESPONSE_KEY_CODE (Send key code list of defined NV graphics)	
<i>buf</i>	Specify an ArrayList<String> array. When the response is retrieved successfully, the key code of NV graphics is as a stored in string array. Example: <i>buf.size()</i> = 3, <i>buf[0]</i> = "22", <i>buf[1]</i> = "23", <i>buf[2]</i> = "24", etc.
PRINTER_RESPONSE_BATTERY_STATUS (Battery remaining capacity level)	
<i>buf</i>	Specify an integer array of length 1. When the response is retrieved successfully, the battery remaining capacity level is stored as a value. See "4.1.3(2)⑦ Battery remaining capacity level" for details of the value. Battery remaining capacity level BATTERY_STATUS_FULL : Full (Battery remaining capacity: approx. 80%) BATTERY_STATUS_MIDDLE : Middle (Battery remaining capacity: approx.40%) BATTERY_STATUS_LOW : Low (Battery remaining capacity: approx. 10%) BATTERY_STATUS_EMPTY : No battery

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

startDiscoveryPrinter Start printer search (Bluetooth)

Searches for the printer using the Bluetooth connection. The found printer information is stored in **PrinterInfo** class.

The method of syntax (a) specifies the instance of **PrinterListener**.

The method of syntax (b) specifies the instance of **PrinterListener** and specifies the context to call the application.

Syntax (a) public void **startDiscoveryPrinter**(PrinterListener *listener*) throws **PrinterException**

(b) public void **startDiscoveryPrinter**(PrinterListener *listener*, Context *context*)
throws **PrinterException**

Parameter *listener* Instance of **PrinterListener**
Completion of this method or cancellation by **cancelDiscoveryPrinter** is notified to the user application as an end event by **finishEvent** through the instance set in *listener*.

context Context of the application

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.7 **PrinterException Class**" for details of the error.

- Description** This method may discover other printers besides SII printer. In addition, the printers in which the Bluetooth connection is already established by the library or other applications are not found.
- Do not call this method from the main thread of the application.
- Note** Use the syntax (a) when using this method newly. Syntax (b) is a method that will be unsupported in the future.

startDiscoveryPrinter

Start printer search (USB)

Searches for the printer using the USB connection. The found printer information is stored in **PrinterInfo** class.

Syntax `public void startDiscoveryPrinter(PrinterListener listener; int deviceType)`
throws **PrinterException**

Parameter *listener* Instance of **PrinterListener**
Completion of this method or cancellation by **cancelDiscoveryDevice** is notified to the user application as an end event by **finishEvent** through the instance set in *listener*.

deviceType Port type
Specify **PRINTER_TYPE_USB**.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method. See "4.1.7 **PrinterException** Class" for details of the error.

Description This method searches for SII printer. The printer information of the found printer is stored to **PrinterInfo** class described later.

startDiscoveryPrinter

Start printer search (TCP/IP)

This method is not supported. When executing this method, it searches SII printer other than MP-B20.

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

cancelDiscoveryPrinter

Cancel printer search

Cancels **startDiscoveryPrinter** under execution.

Syntax `public void cancelDiscoveryPrinter()`

Description Cancellation by this method is notified as an end event to the user application by **finishEvent** through the instance set in *listener* of **startDiscoveryPrinter**.

getFoundPrinter

Get found printer information

Gets the information of the printer found by **startDiscoveryPrinter** in **ArrayList** from the **PrinterInfo** class, which is the storage destination.

Syntax `public ArrayList<PrinterInfo> getFoundPrinter()`

Return value **ArrayList** of **PrinterInfo** class

getSendTimeout

Get send timeout period

Gets the send timeout period.

Syntax `public int getSendTimeout()`

Return value Send timeout period (millisecond: ms)

Description Getting is possible by this method regardless of whether **isConnect** is true or false.

setSendTimeout

Set send timeout period

Sets the send timeout period.

Syntax `public void setSendTimeout(int sendTimeout)`

Parameter *sendTimeout* Send timeout period (millisecond: ms)
The valid range is 100 to 90000.
The value is set to 10000 ms when the value out of the valid range is specified.

Description When the send timeout period is not set by this method, the value is set to 10000.

Setting is possible by this method regardless of whether **isConnect** is true or false.
The set timeout period becomes effective at the next data sending.

getReceiveTimeout

Get receive timeout period

Gets the receive timeout period.

Syntax `public int getReceiveTimeout()`

Return value Receive timeout period (millisecond: ms)

Description Getting is possible by this method regardless of whether **isConnect** is true or false.

setReceiveTimeout

Set receive timeout period

Sets the receive timeout period.

Syntax `public void setReceiveTimeout(int receiveTimeout)`

Parameter *receiveTimeout* Receive timeout period (millisecond: ms)
The valid range is 100 to 90000.
The value is set to 10000 ms when the value out of the valid range is specified.

Description When the receive timeout period is not set by this method, the value is set to 10000.

Setting is possible by this method regardless of whether **isConnect** is true or false.
The set timeout period becomes effective at the next data receiving.

getInternationalCharacter

Get international character set

Gets the value of international character set.

Syntax `public int getInternationalCharacter()`

Return value See "4.1.3(2)① International character set" for details of the value.

Description When the text data is sent by **sendText**, **sendTextEx** or **sendDataFile**, the print result for the following character codes varies. See "Appendix A Character Set" for details about characters to be printed.

Character codes whose print result varies depending on the international character set configuration
0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E

setInternationalCharacter

Set international character set

Sets the value of international character set.

Syntax `public void setInternationalCharacter(int internationalCharacter)`

Parameter *internationalCharacter* International character set constant
See "4.1.3(2)① International character set" for the values available for setting.
When an invalid value is specified, it is ignored.

Description When the international character set is not set by this method, it is as follows depending on the language setting of an Android device.

When the language setting of the Android device is Japanese:

COUNTRY_JAPAN

When the language setting of the Android device is other languages than Japanese:

COUNTRY_USA

getCodePage

Get codepage

Gets the value of codepage.

Syntax `public int getCodePage()`

Return value See "4.1.3(2)② Codepage" for details of the value.

Description The encoder used for sending the text data by **sendText**, **sendTextEx**, or **sendDataFile** is changed. See "Appendix A Character Set" for details about characters to be printed.

setCodePage

Set codepage

Sets the value of codepage.

Syntax `public void setCodePage(int codePage)`

Parameter *codePage* Codepage constant
See "4.1.3(2)② Codepage" for the values available for setting.
When an invalid value is specified, it is ignored.

Description When the codepage is not set by this method, it is as follows depending on the language setting of an Android device.

When the language setting of the Android device is Japanese:

CODE_PAGE_KATAKANA

When the language setting of the Android device is other languages than Japanese:

CODE_PAGE_1252

getPrinterModel

Get printer model

Gets the value of the connecting printer model.

Syntax `public int getPrinterModel()`

Return value See "4.1.3(2)④ Printer model" for details of the value.
PRINTER_MODEL_DEFAULT is returned when **isConnect** is false.

Description Even when the printer is not connected, when **connect** has been succeeded once, the printer model value successfully connected last time is returned.

getPortType

Get connecting port type

Gets the port type used for connecting with the printer.

Syntax `public int getPortType()`

Return value See "4.1.3(2)⑤ Port type" for details of the value.
PRINTER_TYPE_BLUETOOTH is returned when **isConnect** is false.

Description Even when the printer is not connected, when **connect** has been succeeded once, the port type value successfully connected last time is returned.

isConnect

Verify connection state with printer

Verifies connection state with the printer.

Syntax `public boolean isConnect()`

Return value

true	Connected to a printer
false	Not connected to a printer

Description When the data transmission is failed, the communication with the printer is ended, and this method returns false. When false is returned, reconnect with the printer by **connect**.

getSocketKeepingTime

Get socket keeping time

This method is not supported. When executing this method, **PrinterException** is thrown.

Syntax `public int getSocketKeepingTime()`

setSocketKeepingTime

Set socket keeping time

This method is not supported. When executing this method, **PrinterException** is thrown.

Syntax `public void setSocketKeepingTime(int socketKeepingTime)`

Starts or ends batch processing.

Syntax	public void controlTransaction (TransactionFunction <i>transactionFunction</i>) throws PrinterException	
Parameter	<i>transactionFunction</i>	Batch processing selection See "4.1.3(3)⑩ Batch processing selection (TransactionFunction)" for available constants.
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.7 PrinterException Class " for details of the error. When data transmission fails, communication with the printer may be terminated and PrinterException may be thrown. See isConnect for verifying the connection state with the printer.	
Description	The procedure of batch processing is as follows: (1) Start batch processing. Specify TRANSACTION_START . (2) Execute the method. In the case of the batch processing target method, buffering of transmission data is started. The transmission data of the batch processing target method executed during buffering is buffered in the transmission buffer without being sent to the printer. The maximum size of transmission data to be buffered is system dependent. If the buffered transmission data exceeds the maximum size, the batch processing target method at the time of exceeding becomes an error. If an error occurs, the transmission data up to the error is retained. As for the retained transmission data, finish the batch processing in step (3). In the case of a method other than the batch processing target method, transmission data is immediately executed without being buffered. (3) Finish batch processing. When TRANSACTION_PRINT is specified, the buffered transmission data is sent to the printer. The buffered transmission data is retained even after sent to the printer. The retained transmission data is discarded by any of the following: • Specify TRANSACTION_CLEAR • Specify TRANSACTION_START • Execute disconnect The batch processing target methods are as follows: • sendText • sendTextEx • printBarcode • printPDF417 • printQRcode • printDataMatrix • printMaxiCode • printGS1DataBarStacked • printGS1DataBarStackedOmnidirectional • printGS1DataBarExpandedStacked • sendBinary • sendDataFile • printLogo ^{*1}	

*1: **printLogo** under batch processing does not notify the error even when the registered logo does not exist.

4.1.4 PrinterEvent Class

PrinterEvent class gets the end event that occurs when **startDiscoveryPrinter** is terminated.

(1) Method List

Methods provided by the **PrinterEvent** class are shown in the following table.

Name	Description
getEventType	Get end event

(2) End event constant

Constants used for getting the end event are shown in the following table.

Constant Name	Description	Value
EVENT_FINISHED_DISCOVERY	Completion of startDiscoveryPrinter	1
EVENT_CANCELED_DISCOVERY	Cancellation by cancelDiscoveryPrinter	2

(3) Method Details

getEventType	Get end event
---------------------	---------------

Gets the end event when **startDiscoveryPrinter** is terminated.

Syntax `public int getEventType()`

Return value See "4.1.4(2) End event constant" for details of the value.

Description Whether **startDiscoveryPrinter** has been completed or the search has been canceled by **cancelDiscoveryPrinter** can be determined by the end event.
Even when the printer was not discovered, **EVENT_FINISHED_DISCOVERY** is returned.

4.1.5 PrinterListener Interface

PrinterListener interface is for getting the end event when **startDiscoveryPrinter** is terminated.

(1) Method List

Methods of the **PrinterListener** interface are shown in the following table.

Name	Description
finishEvent	End event of the printer search

(2) Method Details

finishEvent	End event of the printer search
--------------------	---------------------------------

End event that is called when **startDiscoveryPrinter** is completed, or when **cancelDiscoveryPrinter** is executed.

Syntax `public void finishEvent(PrinterEvent event)`

Parameter *event* End event
 It is specified by **PrinterEvent** class.

Description This method is an interface, so it is not implemented.
 Implement this method in the user application that receives the notification of the end event by completion of **startDiscoveryPrinter** or cancellation by **cancelDiscoveryPrinter**. Determine the type of the end event by **getEventType** in **PrinterEvent** class.

4.1.6 PrinterInfo Class

PrinterInfo class stores the information of the printer which has been searched by **startDiscoveryPrinter**.

(1) Method List

Printer model name (Bluetooth device name), Bluetooth address, MAC address, port name (device path) and pairing status can be retrieved. Methods of **PrinterInfo** class are shown in the following table.

Name	Description
getPrinterModelName	Get printer model name
getBluetoothAddress	Get Bluetooth address
getMacAddress	Get MAC address
getIsBonded	Get pairing status
getDevicePath	Get device path

(2) Method Details

getPrinterModelName Get printer model name

Gets the character string of the printer model name (Bluetooth device name) from the printer information found by **startDiscoveryPrinter**.

Syntax `public String getPrinterModelName()`

Return value Printer model name (Bluetooth device name)

getBluetoothAddress Get Bluetooth address

Gets the character string of the Bluetooth address from the printer information found by **startDiscoveryPrinter**.

Syntax `public String getBluetoothAddress()`

Return value Bluetooth address

getMacAddress Get MAC address

Gets the character string of the MAC address from the printer information found by **startDiscoveryPrinter**.

Syntax `public String getMacAddress()`

Return value MAC address

getIsBonded Get pairing status

Gets the status of pairing from the printer information found by **startDiscoveryPrinter**.

Syntax `public boolean getIsBonded()`

Return value true Paired
 false Not paired

Gets the character string of the USB device file path from the printer information found by **startDiscoveryPrinter**.

Syntax public String **getDevicePath()**

Return value Device path

4.1.7 PrinterException Class

(1) Method List

Methods provided by the **PrinterException** class are shown in the following table.

Name	Description
PrinterException	Constructor
getErrorCode	Get error codes

(2) Constant List

① Error code

Constants used for getting error codes are shown in following table.

Constant Name	Description	Value
ERROR_ACCESS_DENIED	Failed to get the handle.* ¹	-1
	An unavailable port was specified.	
	An unsupported method was specified.	
ERROR_SHARING_VIOLATION	An already opened port was specified.	-11
ERROR_PORT_NOT_OPENED	The port is not open.	-12
ERROR_DEVICE_NOT_CONNECTED	There is a problem with Bluetooth connection between the Android device and the printer.	-21
	There is a problem with USB connection between the Android device and the printer.	
ERROR_OFFLINE	Disconnected state or the printer is offline.	-22
ERROR_DEVICE_INITIALIZE_FAILED	Failed to change the printer settings. Data sending to the printer is not completed within the send timeout period, or data receiving from the printer is not completed within the receive timeout period.	-31
ERROR_DATA_SIZE_ZERO	0-byte data was specified.	-101
ERROR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.	-102
ERROR_ENCODE_FAILED	An error occurred in encoding text data.* ¹	-111
ERROR_TIMEOUT	Send timeout occurred.	-201
	Receive timeout occurred.	
ERROR_FILE_NOT_FOUND	The specified file is not found.	-301
ERROR_FILE_USED	The specified file is in use by another process.	-302
ERROR_FILE_INVALID	The specified file is invalid.	-303
ERROR_LOW_MEMORY	Memory shortage occurred when loading image data file.	-311
ERROR_OVER_MAX_IMAGE	Either or both of width and height of image data exceeds the number of printable maximum dots.	-312
ERROR_LOGO_NOT_DEFINED	The logo is not registered.	-313
ERROR_LOW_USER_AREA	Remaining user area is insufficient.	-401
ERROR_LOW_EXTERNAL_RAM	Remaining RAM capacity is insufficient.	-402

Constant Name	Description	Value
ERROR_INVALID_PARAM	The specified parameter is invalid.	-9999

*1: Abnormal processing might have occurred.

(3) Method Details

PrinterException Constructor

Constructor for the **com.seikoinstruments.sdk.thermalprinter.PrinterException** class.

Syntax `public PrinterException(int code, String message)`

getErrorCode Get error codes

Gets the error code for thrown exception.

Syntax `public int getErrorCode()`

Return value See "4.1.7(2) Constant List" for details of the error.

Chapter 5

Sample Program

This chapter describes the sample program provided by SII print class library.

SII print class library includes the sample program in Android Studio project format.

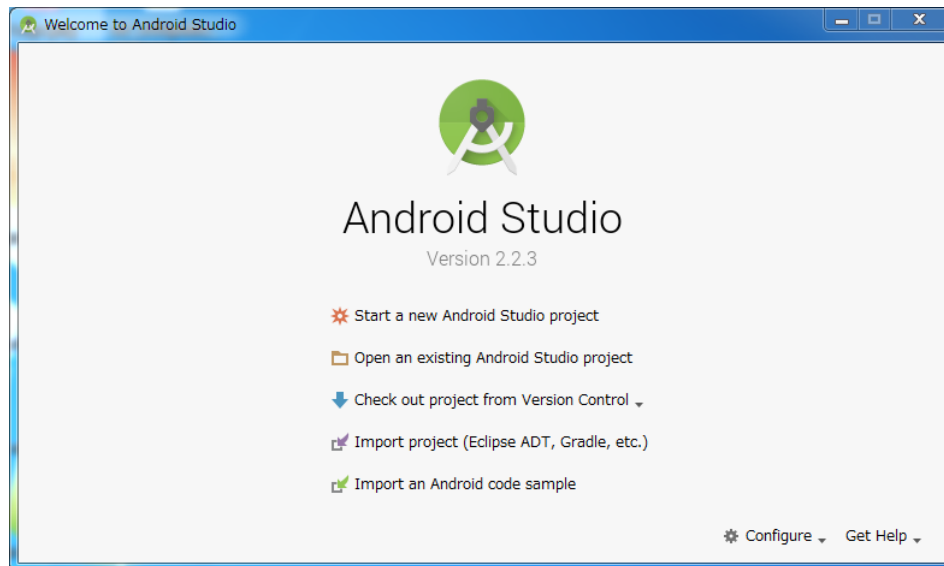
5.1 Installation

Install the sample program.

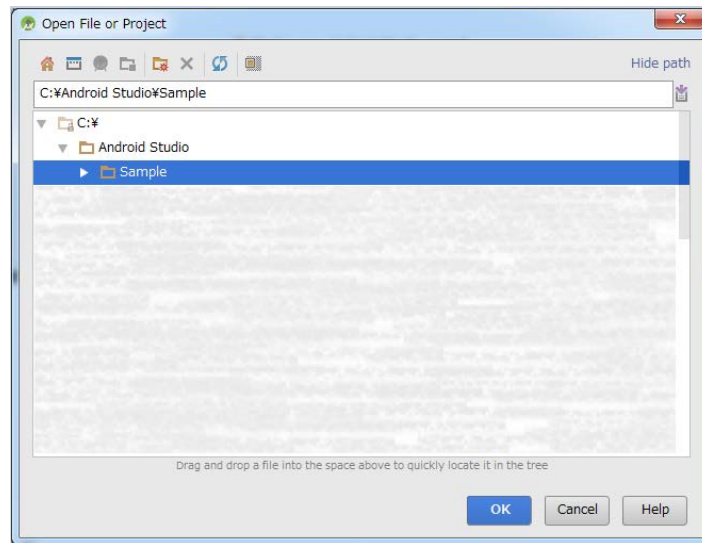
Ensure that the environment for developing Android application is prepared. See "Chapter 3 How to Use the Library" for details about required development environment.

The procedures are shown below.

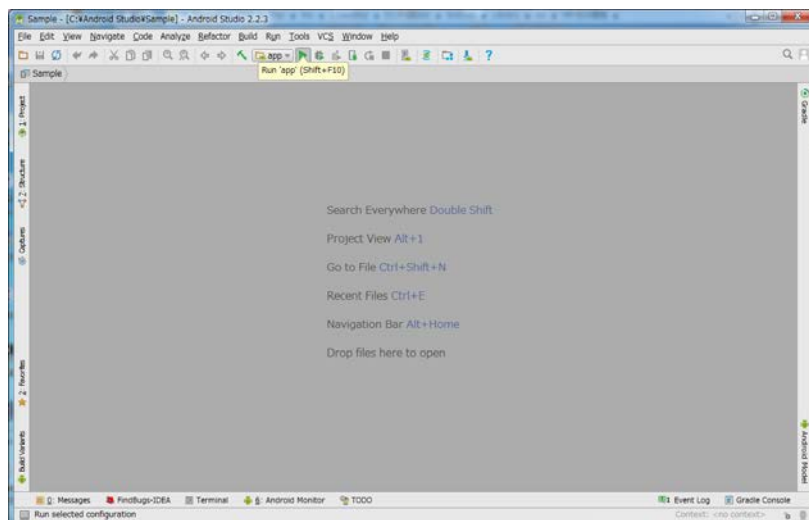
- (1) Place a sample folder at any location.
- (2) Start Android Studio, and click "Open an existing Android Studio project".



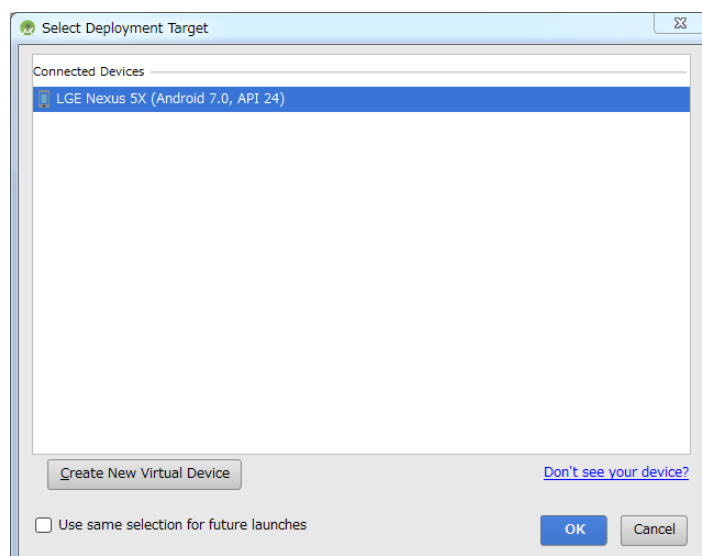
(3) Select the folder placed in the step (1), and click [OK].



(4) Click [Run 'app'].



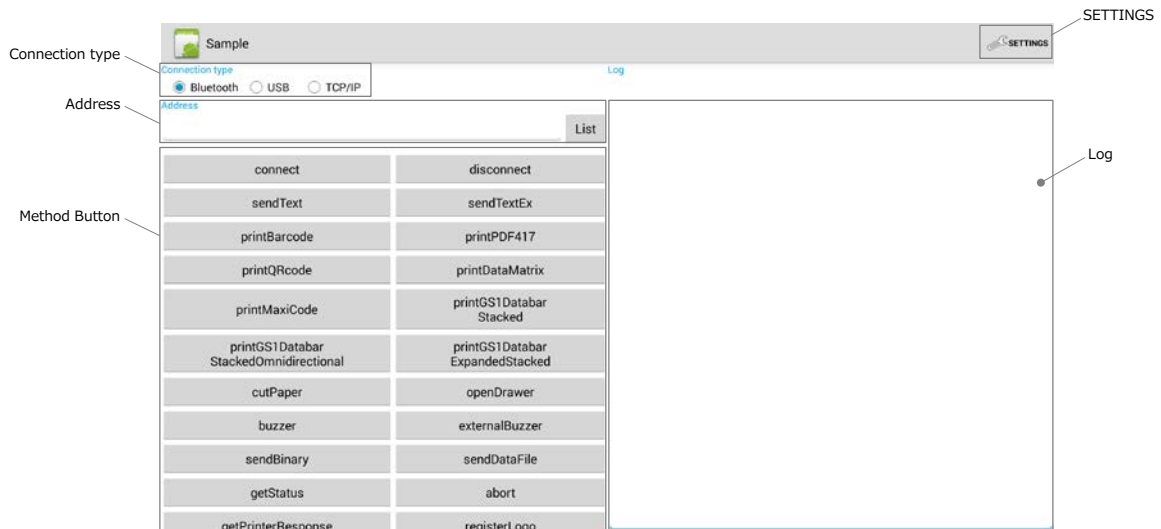
(5) Select the device, and click [OK].




5.2 Screen

This section describes the screen of the sample program.

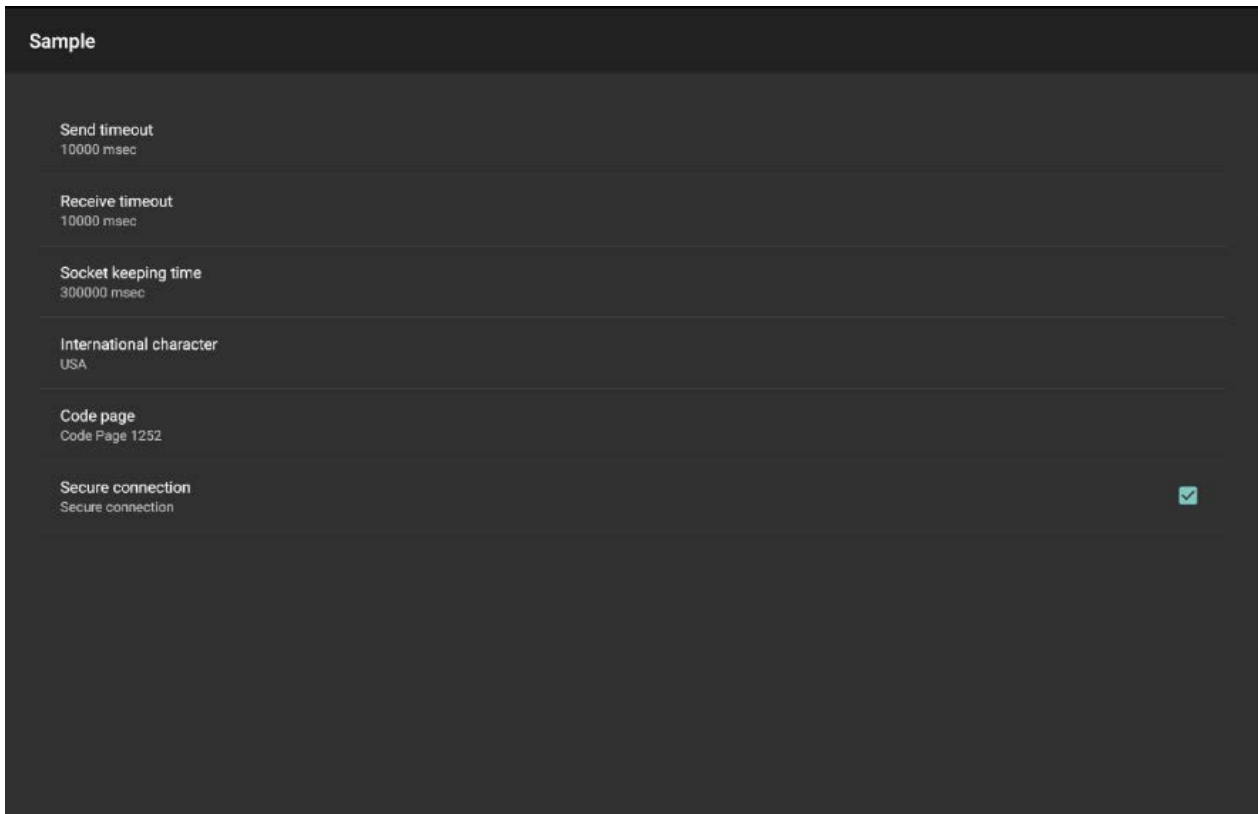
5.2.1 Main screen



Item	Description
Connection type	Selects connection form to a printer.
Address	Specifies the printer address. For manual input: When connecting with Bluetooth, enter the Bluetooth address. For automatic input: By tapping the [List] button, the information of printers found by startDiscoveryPrinter(Bluetooth) is displayed in a list. When selecting a printer from the displayed list, the Bluetooth address is automatically entered.
Method Button	The buttons for executing each method. When scrolling, it is possible to see the methods and properties that are not displayed. See "Chapter 4 Functions of the Library" for details of each method.
SETTINGS	Tapping the [SETTINGS] button opens the function setting screen. In order to go back to the main screen, tap  on the screen.
Log	Executing each method of "Method Button", displays the method execution logs.

5.2.2 [SETTINGS] screen

Various setting functions are displayed in [SETTINGS].



5.3 Precaution

The sample program is subject to change without notice.

No guarantee of proper operation and support are provided for the sample program.

Chapter 6

Disclaimer

We closely monitor the development of SII print class library in order to avoid problems. However, we are not responsible for any damages arising out of the use of SII print class library.

Appendix A

Character Set

A.1 Codepage Table (Character Code Table)

The codepages when **COUNTRY_USA** is set for the international character set are shown below. Print results of the specific character codes vary depending on the setting of the international character set. See "A.2 International Character Set" for the specific character codes.

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

Figure 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																

Figure A-2 CODE_PAGE_KATAKANA

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

Figure A-3 CODE_PAGE_850 (Multilingual)

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

Figure 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	Â	à	¶	ç	ê	ë	è	ï	î	≡	À	§
90	É	È	Ê	ô	Ë	Ï	Ô	Ù	⌘	Ô	Ü	¢	£	Ù	Ù	f
A0		'	ó	ú	¨	³	-	î	¬	¬	½	¼	¾	«	»	
B0	⌘	⌘	⌘													
C0	L	L	T		-	+	+	+	+	+	+	+	+	+	+	+
D0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	2	■	

Figure 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	í	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¡	«	»	
B0	☐	☐	☐		†	‡	§	¶	§		¶		¶		¶	
C0	L	⊥	T	└	├	┤	┘	┐	┌	└	├	┤	┘	┐	┌	└
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	J	÷	≈	°	•	•	√	n	2	■	

Figure 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	í	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	⊥	T	└	├	┤	┘	┐	┌	└	├	┤	┘	┐	┌	└
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	ó	β	ô	ò	õ	õ	μ	×	ú	û	ü	ì	ý	-	'	
F0	-	±	¾	¶	§	÷	,	°	•	•	•	1	3	2	■	

Figure A-7 CODE_PAGE_857 (Turkish)

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

Figure 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

Figure 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	■	

Figure 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	␣	␣	␣		†	‡	§	¶	‡	§	¶	␣	␣	␣	␣	␣
C0	␣	␣	␣		†	‡	§	¶	‡	§	¶	␣	␣	␣	␣	␣
D0	đ	Đ	Ď	Ě	ď	Ň	í	î	ě	Ĵ	Ĵ	■	■	Ť	Ů	■
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	Ů	ý	Ý	ť	´	
F0	-	"	˘	˘	˘	§	÷	˘	˘	˘	˘	Ů	Ř	ř	■	

Figure 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	ü	Ö	Ü	ø	£	Ø	×	ƒ	
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©			¶	¶	¢	¥	₱
C0	L	⊥	T	└	└	+	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	Ј	Г	■	■	І	Ì	■
E0	ó	β	ô	ò	õ	õ	μ	þ	þ	ú	û	ý	ý	-	'	
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2	■	

Figure 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	⊥	T	└	└	+	к	К	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	л	Л	м	М	н	Н	о	О	п	П	Г	■	■	П	я	■
E0	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	ь	№
F0	-	ы	Ы	э	Э	ш	Ш	э	Э	щ	Щ	ч	Ч	§	■	

Figure 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	ث	،	ج	ح	خ	خ
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	؛	س	ش	ص	؟
C0	φ	ء	آ	أ	ؤ	ع	ئ	ب	ة	ث	ث	ج	ح	خ	د	د
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ		÷	×	ع	ع
E0	-	ف	ق	ك	ل	م	ن	هـ	و	ي	ض	ع	غ	غ	م	م
F0	-	”	ن	هـ	ي	ي	ي	غ	ي	غ	ي	غ	ي	غ	ي	■

Figure 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		Ś	..	©	§	«	¬	-	®	Ž	
B0	°	±	ł	ł	μ	¶	•	ą	§	»	Ł	”	ł	ž	ž	
C0	Ř	Á	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
D0	Đ	Ň	Ň	Ó	Ô	Ö	Ö	×	Ř	Ů	Ú	Ú	Ú	Ú	Ý	Ť
E0	ř	á	â	ä	ä	í	ć	ç	č	é	ę	ě	ě	í	î	ď
F0	đ	ň	ň	ó	ô	ö	÷	ř	ů	ú	ú	ú	ú	ý	ť	·

Figure 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	ÿ	ÿ	Ј	Ѡ	Г	І	Š	Ě	©	©	«	¬	-	®	İ	
B0	°	±	İ	ı	г	μ	¶	•	ё	№	е	»	j	S	s	ı
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

Figure 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	Â	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-	
B0	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¸
C0	ı	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
D0	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί
E0	ύ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ό	ύ	ώ	

Figure 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

Figure A-18 CODE_PAGE_1254 (Turkish)

A.2 International Character Set

Print results of the specific character codes vary depending on the setting of the international character set.

The following table shows the specific character codes and their print results.

	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	ℙ	\$	@	ı	Ñ	ı	^	`	..	ñ	}	~
COUNTRY_JAPAN	#	\$	@	[¥]	^	`	{		}	~
COUNTRY_NORWAY	#	α	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	\$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	\$	á	ı	Ñ	ı	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	\$	á	ı	Ñ	ı	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	\$	@	[\]	^	`	{		}	~

Figure A-19 International Character Set



Seiko Instruments Inc.
1-8, Nakase, Mihama-ku, Chiba-shi,
Chiba 261-8507, Japan
Print System Division
Telephone:+81-43-211-1106
Facsimile:+81-43-211-8037

Seiko Instruments USA Inc.
Thermal Printer Div.
21221 S. Western Avenue, Suite 250, Torrance, CA 90501, USA
Telephone:+1-310-517-7778 Facsimile:+1-310-517-7779

Seiko Instruments GmbH
Siemensstrasse 9, D-63263 Neu-Isenburg, Germany
Telephone:+49-6102-297-0 Facsimile:+49-6102-297-222
info@seiko-instruments.de

Seiko Instruments (H.K.) Ltd.
4-5/F, Wyler Center 2,200 Tai Lin Pai Road, Kwai Chung, N.T., Kowloon, Hong Kong
Telephone:+852-2494-5160 Facsimile:+852-2424-0901

(Specifications are subject to change without notice.)