



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

Rev.01

[Products]

SLP720RT Series

Seiko Instruments Inc.

Copyright©2022 by Seiko Instruments Inc.  
All rights reserved.

Android™ is a trademark of Google LLC.  
Oracle and Java are registered trademarks of Oracle and/or its affiliates.  
Other names may be trademarks of their respective owners.

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.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

## Introduction

This manual describes "SII Print Class Library for Android™" (hereinafter referred to as "SII print class library") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

### Target printers

The printers supported by SII print class library are listed below.

Printers	Interface
SLP720RT Series	USB
	TCP/IP

### Terms

The terms used in this manual are defined as below.

Terms	Description
Printer command	Command for controlling the printer described in "SLP720RT SERIES THERMAL PRINTER TECHNICAL REFERENCE".

### Supported Paper and Names in This Manual

The supported paper by the SII print class library and their names in this manual are listed below.

All Type	By type	By function	Abbreviation	Support
Paper	Receipt	Receipt	Receipt	✓
	Linerless label	Linerless label	Label	✓
		Marked linerless label	Marked paper	✓
	SLP Label	SLP Label	Label	✓

# Table of Contents

<b>Chapter 1</b>	<b>Product Overview</b>	<b>1-1</b>
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 Library.....	1-2
<b>Chapter 2</b>	<b>Product Specifications</b>	<b>2-1</b>
2.1	Operating Environment .....	2-1
2.2	Printer Settings .....	2-2
2.3	Precautions .....	2-2
<b>Chapter 3</b>	<b>How to Use Library</b>	<b>3-1</b>
3.1	Android Application Development Environment.....	3-1
3.2	Provided Files .....	3-2
3.3	Build Library into Android Studio Project .....	3-3
3.4	Use Developed Android Application on Android Device.....	3-5
3.5	Precautions .....	3-5
<b>Chapter 4</b>	<b>Functions of Library</b>	<b>4-1</b>
4.1	API Reference.....	4-1
4.1.1	PrinterManager Class.....	4-2
(1)	Method List .....	4-2
(2)	Constant List .....	4-3
①	Printer model.....	4-3
②	Response type.....	4-3
③	International character set .....	4-4
④	Codepage.....	4-4
⑤	Port type.....	4-5
⑥	Barcode or PDF417.....	4-5
(3)	Enumerated Constant List .....	4-6
①	Bold print (CharacterBold) .....	4-6
②	Underline (CharacterUnderline) .....	4-6
③	Reverse print (CharacterReverse) .....	4-6
④	Inversion print (CharacterInversion).....	4-6
⑤	Character font (CharacterFont).....	4-6
⑥	Character scale (CharacterScale).....	4-7
⑦	Alignment (PrintAlignment).....	4-7
⑧	Pending data output specifying (OutputPendingData).....	4-8
⑨	Barcode symbol (BarcodeSymbol).....	4-8
⑩	Module size (ModuleSize) .....	4-9

⑪	HRI character print position (HriPosition) .....	4-10
⑫	N:W ratio (NwRatio) .....	4-11
⑬	Error correction level (ErrorCorrection) .....	4-11
⑭	PDF417 symbol (Pdf417Symbol) .....	4-11
⑮	QR Code Model (QrModel) .....	4-11
⑯	Data Matrix module (DataMatrixModule) .....	4-12
⑰	MaxiCode Mode (MaxiCodeMode) .....	4-13
⑱	Cutting method (CuttingMethod) .....	4-13
⑲	Form feed position (FeedPosition) .....	4-13
⑳	Dithering (Dithering) .....	4-13
㉑	Batch processing selection (TransactionFunction) .....	4-14
(4)	Method Details .....	4-15
	PrinterManager	
	Constructor .....	4-15
	connect	
	Start communicating with printer (Bluetooth) ....	4-15
	connect	
	Start communicating with printer (USB) .....	4-15
	connect	
	Start communicating with printer (TCP/IP) .....	4-15
	disconnect	
	Stop communicating with printer .....	4-16
	setBarcodeScannerListener	
	Start/End callback of barcode scanner .....	4-16
	sendText	
	Send text data .....	4-16
	sendTextEx	
	Send format specified text data .....	4-17
	printBarcode	
	Print barcode .....	4-18
	printPDF417	
	Print PDF417 .....	4-21
	printQRcode	
	Print QR Code .....	4-22
	printDataMatrix	
	Print Data Matrix .....	4-23
	printMaxiCode	
	Print MaxiCode .....	4-23
	printGS1DataBarStacked	
	Print GS1 Databar Stacked .....	4-24
	printGS1DataBarStackedOmnidirectional	
	Print GS1 Databar Stacked Omni-directional ....	4-24
	printGS1DataBarExpandedStacked	
	Print GS1 Databar Expanded Stacked .....	4-25
	printAztecCode	
	Print Aztec Code .....	4-25
	cutPaper	
	Cut paper .....	4-26
	feedPosition	
	Paper form feed .....	4-26
	openDrawer	
	Open cash drawer .....	4-26
	buzzer	
	Sound buzzer .....	4-26
	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) .....	4-33
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 .....	4-38
getVersion	Get SDK version .....	4-38
controlTransaction	Start/End batch processing.....	4-38
4.1.2 PrinterEvent Class .....		4-40
(1) Method List .....		4-40
(2) End event constant.....		4-40
(3) Method Details .....		4-40
getEventType	Get end event .....	4-40
4.1.3 PrinterListener Interface .....		4-41
(1) Method List .....		4-41
(2) Method Details .....		4-41
finishEvent	End event of printer search .....	4-41
4.1.4 PrinterInfo Class.....		4-42
(1) Method List .....		4-42
(2) Method Details .....		4-42
getPrinterModelName	Get printer model name .....	4-42
getBluetoothAddress	Get Bluetooth address.....	4-42
getMacAddress	Get MAC address.....	4-42
getIPAddress	Get IP address .....	4-42
getIsBonded	Get pairing status .....	4-43
getDevicePath	Get device path.....	4-43
4.1.5 PrinterException Class .....		4-44
(1) Method List .....		4-44
(2) Constant List .....		4-44
① Error code .....		4-44
(3) Method Details .....		4-45

PrinterException	Constructor .....	4-45
getErrorCode	Get error codes .....	4-45
4.1.6	CallbackFunctionListener Interface.....	4-46
(1)	Method List .....	4-46
(2)	Method Details .....	4-46
onStatusChanged	Change event of printer status.....	4-46
4.1.7	BarcodeScannerListener Interface.....	4-47

---

Chapter 5	Sample Program	5-1
-----------	----------------	-----

---

5.1	Installation.....	5-1
5.2	Screen Layout .....	5-3
5.2.1	Main screen.....	5-3
5.2.2	[SETTINGS] screen.....	5-4
5.3	Precaution.....	5-4

---

Appendix A	Character Set	A-1
------------	---------------	-----

---

A.1	Codepage 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.

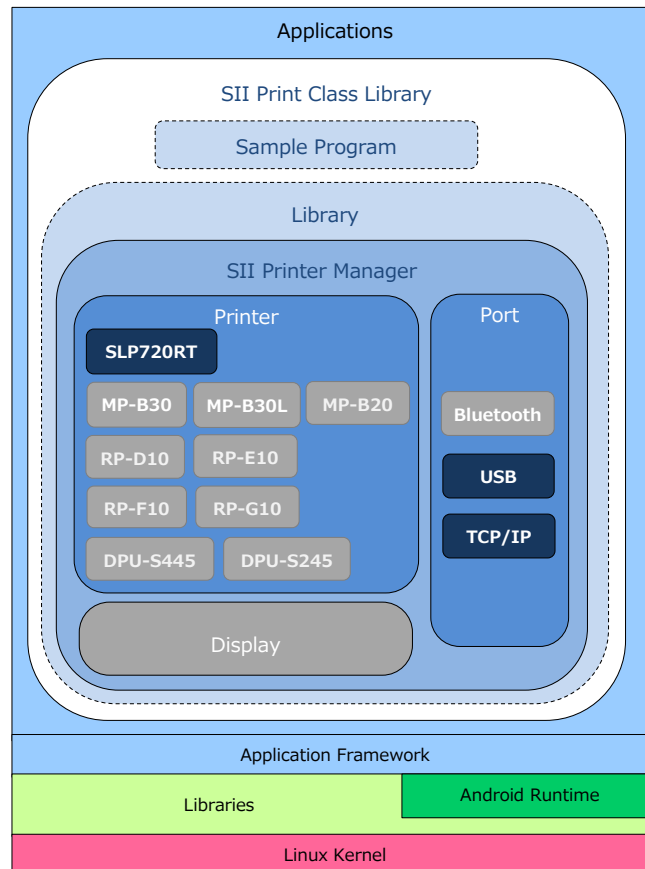
### 1.1 Functions Provided by SII Print Class Library

The SII print class library including the library and the sample program provides the functions to use SII printer SLP720RT series (hereinafter referred to as "printer") in Android applications. Moreover, the SII print class library provides the library sample program in Android Studio project.

### 1.2 SII Print Class Library Overview

#### 1.2.1 SII Print Class Library Configuration

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





### 1.2.2 Functions Provided by Library

By using the library, Android applications can easily send print data and printer commands to a printer through communication port (USB or TCP/IP) on an Android device. Also, the applications can get the 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\*<sup>1</sup>)
- Printing barcode and 2-dimensional barcode
- Sending a data file to a printer (print data and/or printer commands\*<sup>1</sup>)
- 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 TCP/IP

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

<b>(NOTE) SLP720RT does not support the APIs of page mode, Display, or the barcode scanner.</b>
---

## 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		SLP720RT	
		Communication Interface		USB	TCP/IP
Android Device	Communication Port		USB*1	TCP/IP*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)			
		Android 11.0 (API 30)			
		Android 12.0 (API 31)			
Supported Language			Japanese, English		

<sup>\*1</sup>: Android device needs to support USB host function.

<sup>\*2</sup>: The wireless LAN access point that the Android device is connected and the printer need to be connected to the same network.

## 2.2 Printer Settings

Set the memory switches of the printer to [Value] in the following table when using the library.  
The memory switch of the printer can be set in the Android app "SII Printer Utility" on the Google Play.  
See "SLP720RT SERIES THERMAL PRINTER USER'S GUIDE" for details about the memory switches and the factory default settings.

MS	Function	Value
1-2	Taken Mode Selection (Taken Mode)	0 : Enable <sup>*1</sup> 1 : Disable <sup>*2</sup>
1-3	Mark Mode Selection (Mark Mode)	0 : Enable <sup>*3</sup> 1 : Disable <sup>*4</sup>
4-6	Paper Auto Detection Selection (Paper Auto Detection)	0 : Enable <sup>*3</sup> 1 : Disable <sup>*3*4</sup>
5-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable
5-2	Initialized Response Selection (Init. Response)	0 : Enable
5-3	Data Discard Selection When Error Occurs (Error Through)	0 : Enable
5-4	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1 : Disable
7	Thermal Paper Selection (Thermal Paper)	00B : Receipt 01B : Linerless label 10B : SLP Label
13-3	Realtime Command Selection (Realtime Command)	1 : Enable

\*1: When printing continuously on linerless label, set this value to "Enable".

The status response of the taken sensor is responded when this value is set to "Enable".

\*2: When printing continuously on receipt or SLP Label, set this value to "Disable".

\*3: When using **feedPosition**, one of the following settings is necessary.

- To automatically detect paper, set the memory switch MS 4-6 (Paper Auto Detection Selection) of the printer to "Enable".

- To specify the paper, set MS 4-6 (Paper Auto Detection Selection) to "Disable" and set MS 1-3 (Mark Mode Selection) to "Enable".

In addition, select the paper to use as follows.

- For marked linerless label:

- Set MS 7 (Thermal Paper Selection) to "Linerless label".

- For SLP Label:

- Set MS 7 (Thermal Paper Selection) to "SLP Label".

\*4: Set this value to "Disable" and select the paper to be used in the memory switch MS 7 (Thermal Paper Selection) of the printer when using the receipt (other than marked paper) or the Linerless label (other than the marked paper).

## 2.3 Precautions

Communication ports cannot be shared with the printer driver and other libraries when using TCP/IP.

## Chapter 3

### How to Use Library

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

#### 3.1 Android Application Development Environment

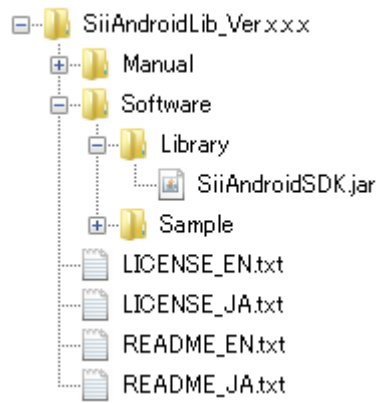
In order to develop Android applications, the following tools are required.  
See each of the following URLs for more details.

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

The description in and after this chapter is on the premise that the environment where each tool is available is prepared.

## 3.2 Provided Files

The file configuration of the 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 Build Library into Android Studio Project

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

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

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

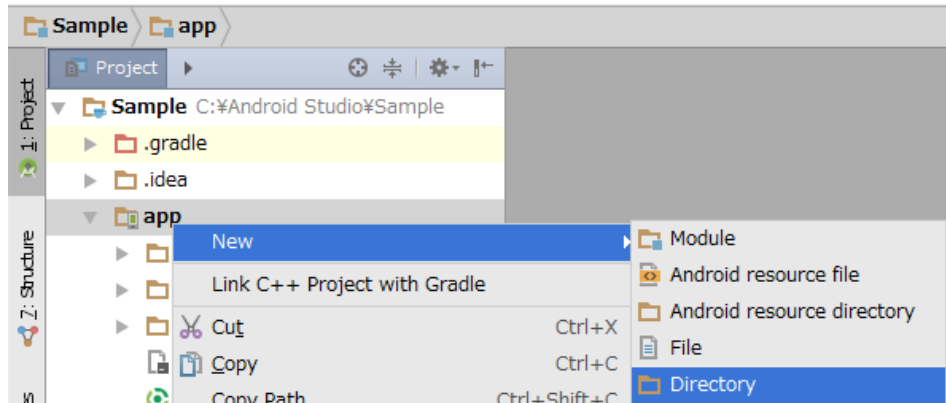


Figure 3-2

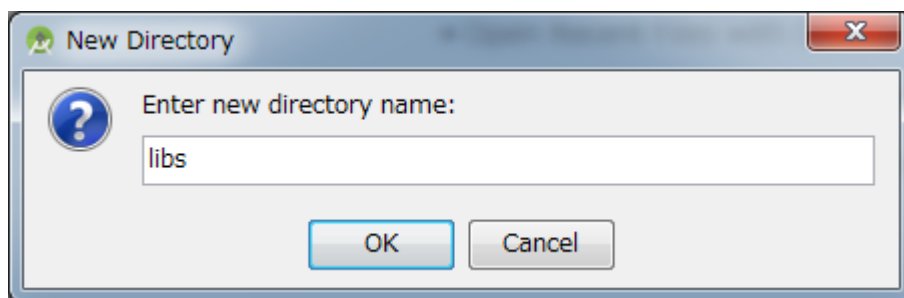
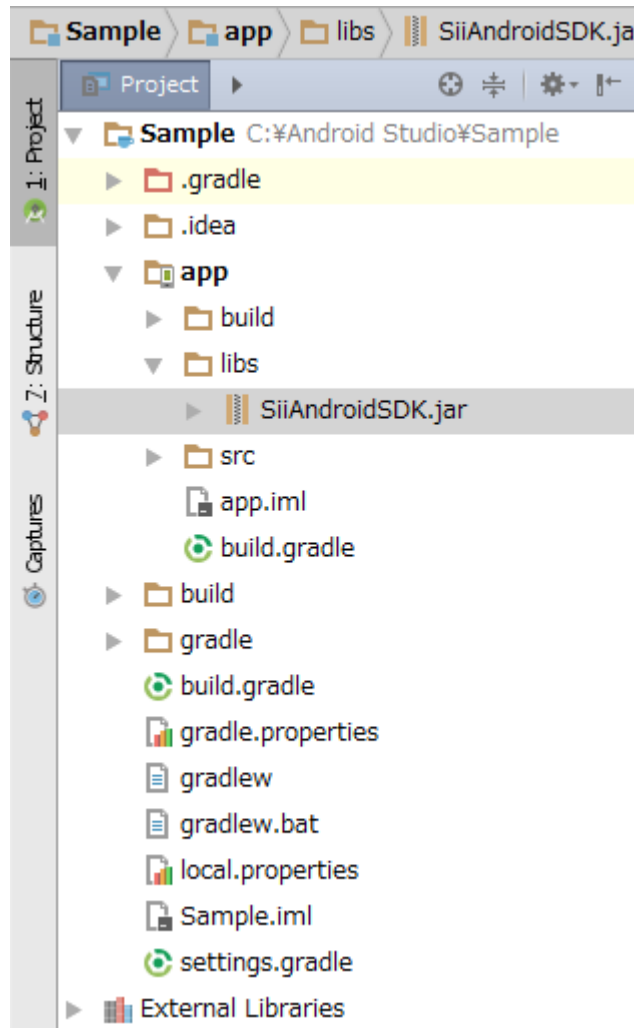


Figure 3-3

- (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 to the beginning of the main source file.  
(Add to 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;
```

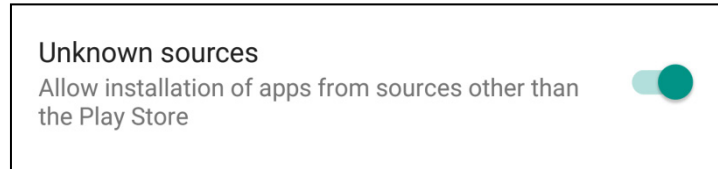
By completing these procedures, the library function becomes available.

### 3.4 Use Developed Android Application on Android Device

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

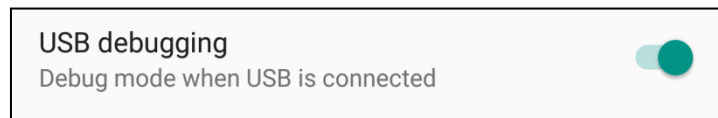
**(NOTE) This procedure is based on the menu of Android 7.1. Menu contents may vary depending on the Android device to use.**

- (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**

### 3.5 Precautions

- **About Scoped Storage**

"Scoped Storage" that is introduced in Android 10 distinguishes between app-specific storage and external storage.

When targeting Android 10 (API 29) or later, files that do not correspond to media files in the external storage cannot be handled directly. Files that do not correspond to media files can be handled by using the "Storage Access Framework".

See below for details of Scoped Storage.

- Data and file storage overview  
<https://developer.android.com/training/data-storage>



## Chapter 4

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

\*1: ✓: Supported, -: Not supported

**(NOTE)** SLP720RT does not support the APIs of page mode, Display, or the barcode scanner.

#### 4.1.1 PrinterManager Class

##### (1) Method List

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

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

Name	Description	Supported *1
cancelDiscoveryPrinter	Cancel printer search	✓
getFoundPrinter	Get found printer information	✓
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	✓
getVersion	Get SDK version	✓
controlTransaction	Start/End batch processing	✓

\*1: ✓ : Supported, -: Not supported

## (2) Constant List

### ① 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_SLP720RT	SLP720RT	305
PRINTER_MODEL_DEFAULT	Default of printer model	284

### ② 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 of user area	1
PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity of 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

③ 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

④ Codepage

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

Constant Name	Description	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 <sup>*1*2</sup>	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.

#### 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
<b>PRINTER_TYPE_BLUETOOTH<sup>*1</sup></b>	Bluetooth	0
<b>PRINTER_TYPE_USB</b>	USB	1
<b>PRINTER_TYPE_TCP</b>	TCP/IP	2

\*1: Not supported.

#### ⑥ Barcode or PDF417

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

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

### (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
<b>BOLD_CANCEL</b>	Cancel bold print
<b>BOLD</b>	Specify bold print

#### ② Underline (CharacterUnderline)

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

Constant Name	Description
<b>UNDERLINE_CANCEL</b>	Cancel underline print
<b>UNDERLINE_1</b>	Specify 1-dot width underline print
<b>UNDERLINE_2</b>	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
<b>REVERSE_CANCEL</b>	Cancel reverse print
<b>REVERSE</b>	Specify reverse print

#### ④ Inversion print (CharacterInversion)

Constants of enumerated type used for inversion print are shown in the following table.  
Inversion print cannot be added to the text data before inserting a new line feed.

Constant Name	Description
<b>INVERSION_CANCEL</b>	Cancel inversion print
<b>INVERSION</b>	Specify inversion print

#### ⑤ Character font (CharacterFont)

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

Constant Name	Description
<b>FONT_A</b>	Font A (24 × 12)
<b>FONT_B</b>	Font B (16 × 8)

⑥ Character scale (CharacterScale)

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

Constant Name	Description
VERTICAL_1_HORIZONTAL_1	Height × 1 and width × 1
VERTICAL_1_HORIZONTAL_2	Height × 1 and width × 2
VERTICAL_1_HORIZONTAL_3	Height × 1 and width × 3
VERTICAL_1_HORIZONTAL_4	Height × 1 and width × 4
VERTICAL_2_HORIZONTAL_1	Height × 2 and width × 1
VERTICAL_2_HORIZONTAL_2	Height × 2 and width × 2
VERTICAL_2_HORIZONTAL_3	Height × 2 and width × 3
VERTICAL_2_HORIZONTAL_4	Height × 2 and width × 4
VERTICAL_2_HORIZONTAL_6	Height × 2 and width × 6
VERTICAL_3_HORIZONTAL_1	Height × 3 and width × 1
VERTICAL_3_HORIZONTAL_2	Height × 3 and width × 2
VERTICAL_3_HORIZONTAL_3	Height × 3 and width × 3
VERTICAL_3_HORIZONTAL_4	Height × 3 and width × 4
VERTICAL_4_HORIZONTAL_1	Height × 4 and width × 1
VERTICAL_4_HORIZONTAL_2	Height × 4 and width × 2
VERTICAL_4_HORIZONTAL_3	Height × 4 and width × 3
VERTICAL_4_HORIZONTAL_4	Height × 4 and width × 4
VERTICAL_4_HORIZONTAL_6	Height × 4 and width × 6
VERTICAL_4_HORIZONTAL_8	Height × 4 and width × 8
VERTICAL_6_HORIZONTAL_2	Height × 6 and width × 2
VERTICAL_6_HORIZONTAL_4	Height × 6 and width × 4
VERTICAL_6_HORIZONTAL_6	Height × 6 and width × 6
VERTICAL_6_HORIZONTAL_8	Height × 6 and width × 8
VERTICAL_8_HORIZONTAL_4	Height × 8 and width × 4
VERTICAL_8_HORIZONTAL_6	Height × 8 and width × 6
VERTICAL_8_HORIZONTAL_8	Height × 8 and width × 8

⑦ Alignment (PrintAlignment)

Constants of enumerated type used for alignment are shown in the following table.  
Alignment cannot be added to the text data before inserting a new line feed.

Constant Name	Description
ALIGNMENT_LEFT	Aligned left
ALIGNMENT_CENTER	Centered
ALIGNMENT_RIGHT	Aligned right

⑧ 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

⑨ Barcode symbol (BarcodeSymbol)

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

Constant Name	Description	Syntax* <sup>1</sup>
BARCODE_SYMBOL_UPC_A	UPC-A	(a)
BARCODE_SYMBOL_UPC_E	UPC-E	(a)
BARCODE_SYMBOL_EAN13	EAN13	(a)
BARCODE_SYMBOL_JAN13	JAN13	(a)
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 `printBarcode` for details of syntax.



⑩ 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	
PDF417_MODULE_WIDTH_4	Nominal fine element width 4 dots	
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	printQRcode
QR_MODULE_SIZE_2	2 dots	
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	

Constant Name	Description	Method to Use
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	
GS1DATABAR_MODULE_SIZE_2	2 dots	<ul style="list-style-type: none"> <li>● printGS1DataBarStacked</li> <li>● printGS1DataBarStackedOmnidirectional</li> <li>● printGS1DataBarExpandedStacked</li> </ul>
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	

⑪ 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)

⑫ 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

⑬ 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	

⑭ 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

⑮ 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

⑩ 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
DATA_MATRIX_16_36	Number of modules: 16 × 36
DATA_MATRIX_16_48	Number of modules: 16 × 48

⑰ MaxiCode Mode (MaxiCodeMode)

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

Constant Name	Description
<b>MAXI_CODE_2</b>	Mode2
<b>MAXI_CODE_3</b>	Mode3
<b>MAXI_CODE_4</b>	Mode4
<b>MAXI_CODE_5</b>	Mode5

⑱ Cutting method (CuttingMethod)

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

Constant Name	Description
<b>CUT_FULL</b>	Paper feed operation to the paper cut position Full cut
<b>CUT_PARTIAL</b>	Paper feed operation to the paper cut position Partial cut

⑲ Form feed position (FeedPosition)

Constants of enumerated type used for form feed position of marked paper or label are shown in the following table.

Constant Name	Description
<b>FEED_CUTTER</b>	After detecting the mark, feeds the paper to the cutting position. The paper feed length is the length of the memory switches MS 8 to 9 (Mark Detection Cut Position Correction) of the printer. The default of the paper feed length is 58 dots (7.25 mm).

⑳ Dithering (Dithering)

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

Constant Name	Description
<b>DITHERING_DISABLE</b>	Dithering is disabled
<b>DITHERING_ERRORDIFFUSION</b>	Dithering is enabled

②① 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

#### (4) Method Details

### PrinterManager

### Constructor

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

Syntax      `public PrinterManager(Context context)`

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

### connect

### Start communicating with printer (Bluetooth)

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

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

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

### connect

### Start communicating with printer (USB)

Starts communication with a printer by USB connection.

Syntax      `public void connect(int printerModel)` throws **PrinterException**

Parameter    *printerModel*                      Printer model constant for USB connection  
See "4.1.1(2)① Printer model" for available constants.

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 PrinterException Class**" for details of the error.

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

The printer specified by *printerModel* is connected.  
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`.

### connect

### Start communicating with printer (TCP/IP)

Starts communication with a printer by TCP/IP connection.

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

Parameter    *printerModel*                      Printer model constant for Ethernet connection  
See "4.1.1(2)① Printer model" for available constants.

*address*                              IP address  
Example: "192.168.0.190"

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 PrinterException Class**" for details of the error.

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

Starts communication with a printer connected to the same network as the Android device by TCP/IP connection.

Connects to the IP address specified by *address*. TCP ports 9100 and 26100 are used for communication. Also, printer initial setting is performed at the connection based on *printerModel* specified.

- **Creating/discarding of socket in TCP/IP connection of the library**

After **connect**, the library retains the created socket until **disconnect**.

And connecting to the same printer from other applications is not possible until **disconnect**.

Based on the completion of data transmission to the printer, the socket is once discarded after elapsing socket keeping time set by **setSocketKeepingTime**. Then the new socket is created immediately and used for the next connection.

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**.

## **disconnect**

## Stop communicating with printer

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 this method is being called. See "**4.1.5 PrinterException Class**" for details of the error.

Description This method discards the print data kept by **controlTransaction**. The instance of **CallbackFunctionListener** interface kept by **setCallbackFunctionListener** is discarded and the callback is stopped.

Note          It is recommended to get the execution response by **PRINTER\_RESPONSE\_REQUEST** of **getPrinterResponse** 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.

## **setBarcodeScannerListener**

## Start/End callback of barcode scanner

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

Syntax      `public void setBarcodeScannerListener(BarcodeScannerListener listener) throws PrinterException`

## **sendText**

## Send text data

Sends text data.

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

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



Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.1.5 PrinterException Class</b>" for details of the error.</p> <p>When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Description	<p>This method encodes the specified text data to printable text data based on <b>setInternationalCharacter</b> and <b>setCodePage</b>, and then sends it to the printer.</p> <p>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.</p>

<b>sendTextEx</b>	<b>Send format specified text data</b>
-------------------	--

Sends format specified text data to the printer.  
The method of syntax (a) or (c) 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	<p>(a) public void <b>sendTextEx</b>(String <i>text</i>,  CharacterBold <i>bold</i>,  CharacterUnderline <i>underline</i>,  CharacterReverse <i>reverse</i>,  CharacterFont <i>font</i>,  CharacterScale <i>scale</i>,  PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></p> <p>(b) public void <b>sendTextEx</b>(String <i>text</i>,  CharacterBold <i>bold</i>,  CharacterUnderline <i>underline</i>,  CharacterReverse <i>reverse</i>,  CharacterFont <i>font</i>,  CharacterScale <i>scale</i>,  PrintAlignment <i>alignment</i>,  OutputPendingData <i>output</i>) throws <b>PrinterException</b></p> <p>(c) public void <b>sendTextEx</b>(String <i>text</i>,  CharacterBold <i>bold</i>,  CharacterUnderline <i>underline</i>,  CharacterReverse <i>reverse</i>,  CharacterInversion <i>inversion</i>,  CharacterFont <i>font</i>,  CharacterScale <i>scale</i>,  PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></p>	
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).
	<i>bold</i>	Bold print See "4.1.1(3)① Bold print (CharacterBold)" for available constants.
	<i>underline</i>	Underline See "4.1.1(3)② Underline (CharacterUnderline)" for available constants.
	<i>reverse</i>	Reverse print See "4.1.1(3)③ Reverse print (CharacterReverse)" for available constants.

<i>inversion</i>	Inversion print See "4.1.1(3)④ Inversion print (CharacterInversion)" for available constants.
<i>font</i>	Font See "4.1.1(3)⑤ Character font (CharacterFont)" for available constants.
<i>scale</i>	Character scale See "4.1.1(3)⑥ Character scale (CharacterScale)" for available constants.
<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.
<i>output</i>	Pending data output specifying See "4.1.1(3)⑧ Pending data output specifying (OutputPendingData)" for available constants.

Exception	<p><b>PrinterException</b>  <b>PrinterException</b> is thrown when an error occurs while this method is being called.  See "4.1.5 PrinterException Class" for details of the error.  When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Description	<p>This method encodes the specified text data to printable text data based on <b>setInternationalCharacter</b> and <b>setCodePage</b>, and then sends it to the printer.</p> <p>For laying out text data by sending following printer commands with <b>sendBinary</b> or <b>sendDataFile</b>, specify <b>PENDING_DATA_OUTPUT_TOGETHER</b> at <i>output</i> in the method of syntax (b).</p> <ul style="list-style-type: none"> <li>· "Horizontal Tab"</li> <li>· "Specify Absolute Position"</li> <li>· "Specify Relative Position"</li> </ul> <p>When the method of syntax (a) or (c) is executed or <b>PENDING_DATA_OUTPUT_FIRST</b> is specified at <i>output</i> in the method of syntax (b), the print position set in above becomes invalid.</p> <p>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.</p>

## 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	<p>(a) public void <b>printBarcode</b>(BarcodeSymbol <i>barcodeSymbol</i>,  String <i>text</i>,  ModuleSize <i>moduleSize</i>,  int <i>moduleHeight</i>,  HriPosition <i>hriPosition</i>,  CharacterFont <i>hriFont</i>,  PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></p>
--------	---

(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 "4.1.1(3)⑨ 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 "4.1.1(3)⑩ Module size (ModuleSize)" for available constants.
	<i>moduleHeight</i>	Barcode height (dot)

- When *barcodeSymbol* 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
<b>BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL</b>		
	<b>BARCODE_MODULE_WIDTH_2</b>	66 to 255
	<b>BARCODE_MODULE_WIDTH_3</b>	99 to 255
	<b>BARCODE_MODULE_WIDTH_4</b>	132 to 255
	<b>BARCODE_MODULE_WIDTH_5</b>	165 to 255
	<b>BARCODE_MODULE_WIDTH_6</b>	198 to 255

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

<i>hriPosition</i>	HRI character print position See "4.1.1(3)⑪ HRI character print position (HriPosition)" for available constants.
<i>hriFont</i>	HRI character font See "4.1.1(3)⑤ Character font (CharacterFont)" for available constants.
<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.
<i>nwRatio</i>	N:W ratio See "4.1.1(3)⑫ N:W ratio (NwRatio)" for available constants. Depending on specified <i>nwRatio</i> and <i>moduleSize</i> , the wide element width is set as shown in the following table.

<i>moduleSize</i>	<i>nwRatio</i>		
	<b>NWRATIO_1TO2</b>	<b>NWRATIO_1TO2_5</b>	<b>NWRATIO_1TO3</b>
<b>BARCODE_MODULE_WIDTH_2</b>	0.500 mm (4 dots)	0.625 mm (5 dots)	0.750 mm (6 dots)
<b>BARCODE_MODULE_WIDTH_3</b>	0.750 mm (6 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)
<b>BARCODE_MODULE_WIDTH_4</b>	1.000 mm (8 dots)	1.250 mm (10 dots)	1.500 mm (12 dots)
<b>BARCODE_MODULE_WIDTH_5</b>	1.250 mm (10 dots)	1.625 mm (13 dots)	1.875 mm (15 dots)
<b>BARCODE_MODULE_WIDTH_6</b>	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.1.5 <b>PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.

## printPDF417

## Print PDF417

Prints PDF417.

The method of syntax (a) specifies PDF417 symbol.

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

Syntax	<p>(a) public void <b>printPDF417</b>(String <i>text</i>,  ErrorCorrection <i>errorCorrection</i>,  int <i>row</i>,  int <i>column</i>,  ModuleSize <i>moduleSize</i>,  int <i>moduleHeight</i>,  PrintAlignment <i>alignment</i>,  Pdf417Symbol <i>pdf417Symbol</i>) throws <b>PrinterException</b></p> <p>(b) public void <b>printPDF417</b>(String <i>text</i>,  ErrorCorrection <i>errorCorrection</i>,  int <i>row</i>,  int <i>column</i>,  ModuleSize <i>moduleSize</i>,  int <i>moduleHeight</i>,  PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></p>	
Parameter	<i>text</i>	Barcode data to send to the printer
	<i>errorCorrection</i>	Error correction level See "4.1.1(3)⑬ Error correction level (ErrorCorrection)" for available constants.
	<i>row</i>	The number of rows (row) The valid range is 0, 3 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 "4.1.1(3)⑩ 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 "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.

*pdf417Symbol*

Symbol of PDF417

See "4.1.1(3)<sup>⑭</sup> PDF417 symbol (Pdf417Symbol)" for available constants.

Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.1.5 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.

## printQRcode

## Print QR Code

Prints QR Code.

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

The method of syntax (b) specifies QR Code Model.

Syntax	(a) public void <b>printQRcode</b> (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>  (b) public void <b>printQRcode</b> (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> , QrModel <i>model</i> ) throws <b>PrinterException</b>										
Parameter	<table><tr><td><i>text</i></td><td>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>.</td></tr><tr><td><i>errorCorrection</i></td><td>Error correction level See "4.1.1(3)<sup>⑬</sup> Error correction level (ErrorCorrection)" for available constants.</td></tr><tr><td><i>moduleSize</i></td><td>Module size See "4.1.1(3)<sup>⑩</sup> Module size (ModuleSize)" for available constants.</td></tr><tr><td><i>alignment</i></td><td>Alignment See "4.1.1(3)<sup>⑦</sup> Alignment (PrintAlignment)" for available constants.</td></tr><tr><td><i>model</i></td><td>QR Code Model See "4.1.1(3)<sup>⑮</sup> QR Code Model (QrModel)" for available constants.</td></tr></table>	<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 "4.1.1(3) <sup>⑬</sup> Error correction level (ErrorCorrection)" for available constants.	<i>moduleSize</i>	Module size See "4.1.1(3) <sup>⑩</sup> Module size (ModuleSize)" for available constants.	<i>alignment</i>	Alignment See "4.1.1(3) <sup>⑦</sup> Alignment (PrintAlignment)" for available constants.	<i>model</i>	QR Code Model See "4.1.1(3) <sup>⑮</sup> QR Code Model (QrModel)" for available constants.
<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 "4.1.1(3) <sup>⑬</sup> Error correction level (ErrorCorrection)" for available constants.										
<i>moduleSize</i>	Module size See "4.1.1(3) <sup>⑩</sup> Module size (ModuleSize)" for available constants.										
<i>alignment</i>	Alignment See "4.1.1(3) <sup>⑦</sup> Alignment (PrintAlignment)" for available constants.										
<i>model</i>	QR Code Model See "4.1.1(3) <sup>⑮</sup> QR Code Model (QrModel)" for available constants.										
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.1.5 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.										
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.										

Prints Data Matrix.

Syntax	public void <b>printDataMatrix</b> (String <i>text</i> , DataMatrixModule <i>dataMatrixModule</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>text</i>	Barcode data to send to the printer
	<i>dataMatrixModule</i>	The number of Data Matrix modules See "4.1.1(3)⑩ Data Matrix module (DataMatrixModule)" for available constants.
	<i>moduleSize</i>	Module size See "4.1.1(3)⑩ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.1.5 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	

Prints MaxiCode.

Syntax	public void <b>printMaxiCode</b> (String <i>text</i> , MaxiCodeMode <i>maxiCodeMode</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>text</i>	Barcode data to send to the printer
		<ul style="list-style-type: none"> <li>When <i>maxiCodeMode</i> is <b>MAXI_CODE_2</b> Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data.</li> <li>When <i>maxiCodeMode</i> is <b>MAXI_CODE_3</b> Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.</li> </ul>
	<i>maxiCodeMode</i>	MaxiCode Mode See "4.1.1(3)⑪ MaxiCode Mode (MaxiCodeMode)" for available constants.
	<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.

Exception	<p><b>PrinterException</b>  <b>PrinterException</b> is thrown when an error occurs while this method is being called.  See "4.1.5 PrinterException Class" for details of the error.  When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.

## printGS1DataBarStacked Print GS1 Databar Stacked

Prints GS1 Databar Stacked.

Syntax	<pre>public void <b>printGS1DataBarStacked</b>(String <i>text</i>,                                      ModuleSize <i>moduleSize</i>,                                      PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></pre>	
Parameter	<i>text</i>	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.
	<i>moduleSize</i>	Module size See "4.1.1(3)⑩ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.
Exception	<p><b>PrinterException</b>  <b>PrinterException</b> is thrown when an error occurs while this method is being called.  See "4.1.5 PrinterException Class" for details of the error.  When the data transmission is failed, the communication with the printer is ended and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>	

## printGS1DataBarStackedOmnidirectional Print GS1 Databar Stacked Omni-directional

Prints GS1 Databar Stacked Omni-directional.

Syntax	<pre>public void <b>printGS1DataBarStackedOmnidirectional</b>(String <i>text</i>,   int <i>moduleHeight</i>,   ModuleSize <i>moduleSize</i>,   PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></pre>	
Parameter	<i>text</i>	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.
	<i>moduleHeight</i>	Barcode module height (the number of the modules) The valid range is 33 to 255.
	<i>moduleSize</i>	Module size See "4.1.1(3)⑩ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.



**Exception**     **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 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.

<b>printGS1DataBarExpandedStacked</b>	<b>Print GS1 Databar Expanded Stacked</b>
---------------------------------------	---

Prints GS1 Databar Expanded Stacked.

Syntax	public void <b>printGS1DataBarExpandedStacked</b> (String <i>text</i> , int <i>column</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>text</i>	Barcode data to send to the printer Enter any number of characters using the following: ' ', '!', '""', '%', '&', '"', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.
	<i>column</i>	The number of columns Specifies the number of the segments in 1 line. The valid range is the even number from 2 to 20.
	<i>moduleSize</i>	Module size See "4.1.1(3)⑩ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.1.5 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	

<b>printAztecCode</b>	<b>Print Aztec Code</b>
-----------------------	-------------------------

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

Syntax	public void <b>printAztecCode</b> (String <i>text</i> , int <i>layer</i> , int <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , AztecSymbol <i>aztecSymbol</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
--------	---	--

Feeds the paper to the paper cut position and cuts the paper.

Syntax	public void <b>cutPaper</b> (CuttingMethod <i>cuttingMethod</i> ) throws <b>PrinterException</b>	
Parameter	<i>cuttingMethod</i>	Cutting method See "4.1.1(3) <sup>18</sup> Cutting method (CuttingMethod)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.1.5 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	

Performs the paper form feed of marked paper or label to the cut position.

Syntax	public void <b>feedPosition</b> (FeedPosition <i>feedPosition</i> ) throws <b>PrinterException</b>	
Parameter	<i>feedPosition</i>	Form feed position See "4.1.1(3) <sup>19</sup> Form feed position (FeedPosition)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.1.5 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Note	The paper form feed is not performed when this method is executed at the form feed position of the marked paper or the label.	

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

Syntax	public void <b>openDrawer</b> (DrawerNum <i>drawerNum</i> , PulseWidth <i>onOffTime</i> ) throws <b>PrinterException</b>	
--------	--	--

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

Syntax	public void <b>buzzer</b> (int <i>onTime</i> , int <i>offTime</i> ) throws <b>PrinterException</b>	
--------	--	--

This method is not supported. When this method is executed, **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 this method is being called.  
See "**4.1.5 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 is fixed to be disabled.

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

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

Parameter    *fileName*                      Name of the data file to send to the printer  
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"  
When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.  
See "3.5 Precautions - About Scoped Storage" for details.
- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://
 It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

The maximum file size that can be specified is 1 MB (1048576 bytes). The file extensions that can be sent and the file transmission are described below.

- .bmp, .jpg, .jpeg, .png  
Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and registered. Printing is performed at one time after mapping the image file 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.

*alignment* Alignment  
The alignment is valid only when the file extension specified on *fileName* is .bmp, .jpg, .jpeg, .png, or .txt.  
See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.

*dithering* Dithering  
The dithering is valid only when the file extension specified on *fileName* is .bmp, .jpg, .jpeg, or .png.  
See "4.1.1(3)⑩ Dithering (Dithering) " for available constants.

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 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.

getStatus	Get printer status
-----------	--------------------

Gets the latest printer status.  
The method of syntax (a) returns the printer status with return value.  
The method of syntax (b) stores the printer status in an array of int type.

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

Return value Status retrieved from the printer

Parameter *buf* Status retrieved from the printer

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 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	Status retrieved from the printer is stored in an array of NSInteger type.
-------------	--

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	Cutter error	No error	Error
4	Out-of-paper error	No error	Error
5	Reserved	Fixed	-
6	Paper jam error while detecting mark	No error	Error
7	Cover open error	No error	Error
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	Taken sensor status	Paper removed	Paper removal waiting
14	Reserved	-	Fixed
15	Reserved	Fixed	-
16	FLASH memory rewriting	Not rewriting	Rewriting
17 to 18	Reserved	Fixed	-
19 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	<i>listener</i>	Instance of <b>CallbackFunctionListener</b> interface
-----------	-----------------	---

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
 See "**4.1.5 PrinterException Class**" for details of the error.

Description	<p>Register the process executed by callback with <b>onStatusChanged</b>.</p> <p>When specify the instance of <b>CallbackFunctionListener</b> interface in <i>listener</i> and execute this method, the callback is started.</p> <p>When specify null in <i>listener</i> and execute this method, the callback is stopped.</p>
-------------	--

The keeping instance kept by of **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.

Aborts the waiting state of the printer.

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

Exception      **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called.

See **"4.1.5 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 file by <b>sendDataFile</b> is aborted, the printer does not accept other processes until the specified image file is received completely. (Methods and transmission data are misinterpreted and recognized as a part of the image file.) To solve this situation, use this method to abort the waiting state of the printer. Note that when this method is executed, a part of unprinted image file may be printed.
-------------	---

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

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

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

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

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

Parameter	<i>fileName</i>
-----------	-----------------

File name of image file to register as a logo  
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"  
When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.  
See "3.5 Precautions - About Scoped Storage" for details.

- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android

- file://
- content://

It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

The file extensions supporting image file are .bmp, .jpg, .jpeg, and .png. When the image file 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 "4.1.1(3)Ⓣ Dithering (Dithering) " for available constants.

Exception     **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 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 print (key code)  
Specify the ID of the registered logo as a character string.

*alignment*     Alignment  
See "4.1.1(3)⑦ Alignment (PrintAlignment)" for available constants.

Exception     **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.1.5 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 this method is being called.  
See "**4.1.5 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 this method is executed, **PrinterException** is thrown.

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

## unregisterStyleSheet Delete registered style sheet

This method is not supported. When this method is executed, **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 this method is being called.

See "4.1.5 **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 USB connection:

The printer reset is performed by using the SOFT\_RESET function in USB printer class.

For TCP/IP connection:

The reset is performed to the connected printer by our proprietary command (reset request) to TCP port 26100.

The connection with the printer is retained even after this method is executed.

## 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 "4.1.1(2)② 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
<b>PRINTER_RESPONSE_REQUEST</b> (Execution response request)	
<i>buf</i>	Specify an int type array of length 1. Specify 0 to 15 (00h to 0Fh) for <i>buf[0]</i> . When the response is retrieved successfully, the response code of the execution response request is stored to <i>buf[0]</i> with 128 to 143 (80h to 8Fh).
<b>PRINTER_RESPONSE_USER_AREA</b> (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.
<b>PRINTER_RESPONSE_ARRANGE_USER_AREA</b> (Send remaining capacity of user area 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.





Start printer search (TCP/IP)

[illegible]

*retry*      Retry count (times)  
Sends the local broadcast packet the number of times set by *retry*.  
The valid range is 1 to 5.  
When the value is specified less than 1, the number is set to 1.  
When the value is specified more than 5, the number is set to 5.

*timeout* Search timeout period (millisecond: ms)  
Sets the timeout period per search. Each time the local broadcast packet is sent, this method waits for a response from the printer until the period specified by *timeout* elapses.  
The valid range is 3000 to 60000.  
When the value is specified less than 3000, the period is set to 3000 ms.  
When the value is specified more than 60000, the period is set to 60000 ms.

Description	This method searches for SII printer. The printer information of the found printer is stored to <b>PrinterInfo</b> class described later.
-------------	---

The completion or cancellation of the search is notified as an event to the user application by **finishEvent** through the instance set in *listener*.

Cancel printer search

Description	Cancellation by this method is notified as an end event to the user application by <b>finishEvent</b> through the instance set in <i>listener</i> of <b>startDiscoveryPrinter</b> .
-------------	---

## Get found printer information

**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   This method can get the send timeout period 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.  
When the value out of the valid range is specified, the value is set to 10000 ms.

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

This method can set the send timeout period 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   This method can get the receive timeout period 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.  
When the value out of the valid range is specified, the value is set to 10000 ms.

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

This method can set the receive timeout period 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 "4.1.1(2)③ International character set" for details of the value.

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

Character codes with the varying print result depending on the configuration of the international character:  
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 "4.1.1(2)③ International character set" for the configurable values.  
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 than Japanese:  
**COUNTRY\_USA**

## getCodePage

## Get codepage

Gets the value of codepage.

Syntax      `public int getCodePage()`

Return value   See "4.1.1(2)④ 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 "4.1.1(2)④ Codepage" for the configurable values.  
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 than Japanese:  
**CODE\_PAGE\_1252**

## getPrinterModel Get printer model

Gets the value of the connecting printer model.

**Syntax**      `public int getPrinterModel()`

**Return value** See "4.1.1(2)① 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 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 "4.1.1(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

Gets the socket keeping time.

**Syntax**      `public int getSocketKeepingTime()`

**Return value** Socket keeping time (millisecond: ms)

**Description** This method can get the socket keeping time regardless of whether **isConnect** is true or false.

## Set socket keeping time

**Syntax**      `public void setSocketKeepingTime(int socketKeepingTime)`

Default	300000
---------	--------

For the socket keeping time, specify a time equal to Receive Timeout of the printer to be connected. The setting of Receive Timeout can be changed in the Android app "SII Printer Utility" on the Google Play.

The set socket keeping time becomes effective at the next **connect** execution.

## Get SDK version

**Syntax**      `public String getVersion()`

Description	This method can get the SDK version regardless of whether <b>isConnect</b> is true or false.
-------------	--

## Start/End batch processing

[illegible]

Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.1.5 PrinterException Class</b>" for details of the error.</p> <p>When data transmission fails, communication with the printer may be terminated and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
-----------	--

Description	<p>The procedure of batch processing is as follows:</p> <ol style="list-style-type: none"> <li>(1) Start batch processing. Specify <b>TRANSACTION_START</b>.</li> <li>(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.</li> <li>(3) Finish batch processing. When <b>TRANSACTION_PRINT</b> 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: <ul style="list-style-type: none"> <li>· Specify <b>TRANSACTION_CLEAR</b></li> <li>· Specify <b>TRANSACTION_START</b></li> <li>· Execute <b>disconnect</b></li> </ul> </li> </ol> <p>The batch processing target methods are as follows:</p> <ul style="list-style-type: none"> <li>· <b>sendText</b></li> <li>· <b>sendTextEx</b></li> <li>· <b>printBarcode</b></li> <li>· <b>printPDF417</b></li> <li>· <b>printQRcode</b></li> <li>· <b>printDataMatrix</b></li> <li>· <b>printMaxiCode</b></li> <li>· <b>printGS1DataBarStacked</b></li> <li>· <b>printGS1DataBarStackedOmnidirectional</b></li> <li>· <b>printGS1DataBarExpandedStacked</b></li> <li>· <b>cutPaper</b></li> <li>· <b>feedPosition</b></li> <li>· <b>sendBinary</b></li> <li>· <b>sendDataFile</b></li> <li>· <b>printLogo</b><sup>*1</sup></li> </ul>
-------------	--

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

### 4.1.2 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
<b>getEventType</b>	Get end event

#### (2) End event constant

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

Constant Name	Description	Value
<b>EVENT_FINISHED_DISCOVERY</b>	Completion of <b>startDiscoveryPrinter</b>	1
<b>EVENT_CANCELED_DISCOVERY</b>	Cancellation by <b>cancelDiscoveryPrinter</b>	2

#### (3) Method Details

<b>getEventType</b>	Get end event
---------------------	---------------

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

Syntax      `public int getEventType()`

Return value    See "4.1.2(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.



### 4.1.3 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
<b>finishEvent</b>	End event of printer search

#### (2) Method Details

<b>finishEvent</b>	End event of 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.

#### 4.1.4 PrinterInfo Class

**PrinterInfo** class stores the information of the printer found by **startDiscoveryPrinter**.

##### (1) Method List

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

Name	Description
<b>getPrinterModelName</b>	Get printer model name
<b>getBluetoothAddress</b>	Get Bluetooth address
<b>getMacAddress</b>	Get MAC address
<b>getIpAddress</b>	Get IP address
<b>getIsBonded</b>	Get pairing status
<b>getDevicePath</b>	Get device path

##### (2) Method Details

#### **getPrinterModelName** Get printer model name

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

Syntax      `public String getPrinterModelName()`

#### **getBluetoothAddress** Get Bluetooth address

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

Syntax      `public String getBluetoothAddress()`

#### **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

#### **getIpAddress** Get IP address

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

Syntax      `public String getIpAddress()`

Return value   IP address

## **getIsBonded**

Get pairing status

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

Syntax      `public boolean getIsBonded()`

## **getDevicePath**

Get device path

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

#### 4.1.5 PrinterException Class

##### (1) Method List

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

Name	Description
<b>PrinterException</b>	Constructor
<b>getErrorCode</b>	Get error code

##### (2) Constant List

###### ① Error code

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

Constant Name	Description	Value
<b>ERROR_ACCESS_DENIED</b>	Failed to get the handle.*1	-1
	Unavailable port is specified.	
	Unsupported method is specified.	
<b>ERROR_SHARING_VIOLATION</b>	Already opened port is specified.	-11
<b>ERROR_PORT_NOT_OPENED</b>	Port is not opened.	-12
<b>ERROR_DEVICE_NOT_CONNECTED</b>	There is a problem with USB connection between the Android device and printer.	-21
<b>ERROR_DEVICE_INITIALIZE_FAILED</b>	Failed to change the printer setting. 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
<b>ERROR_DATA_SIZE_ZERO</b>	0-byte data has been specified.	-101
<b>ERROR_OVER_MAX_DATA_SIZE</b>	Maximum data size is exceeded.	-102
<b>ERROR_ENCODE_FAILED</b>	Error occurred in encoding text data.*1	-111
<b>ERROR_TIMEOUT</b>	Send timeout occurred.	-201
	Receive timeout occurred.	
<b>ERROR_FILE_NOT_FOUND</b>	Specified file is not found.	-301
<b>ERROR_FILE_USED</b>	Specified file is being used by another process.	-302
<b>ERROR_FILE_INVALID</b>	Specified file is invalid.	-303
<b>ERROR_LOW_MEMORY</b>	Insufficient memory occurred when loading image file.	-311
<b>ERROR_OVER_MAX_IMAGE</b>	Either or both of width and height of image file exceeds the number of printable maximum dots.	-312
<b>ERROR_LOGO_NOT_DEFINED</b>	Logo is not registered.	-313
<b>ERROR_LOW_USER_AREA</b>	Remaining user area is insufficient.	-401
<b>ERROR_LOW_EXTERNAL_RAM</b>	Remaining RAM capacity is insufficient.	-402
<b>ERROR_INVALID_PARAM</b>	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 "4.1.5(2) Constant List" for details of the error.

#### 4.1.6 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
<b>onStatusChanged</b>	Change event of printer status

##### (2) Method Details

<b>onStatusChanged</b>	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.

#### **4.1.7 BarcodeScannerListener Interface**

**BarcodeScannerListener** Interface is an interface for the barcode scanner connection, barcode scanner disconnection, or received barcode data obtaining.

SLP720RT does not support this interface.

## 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.

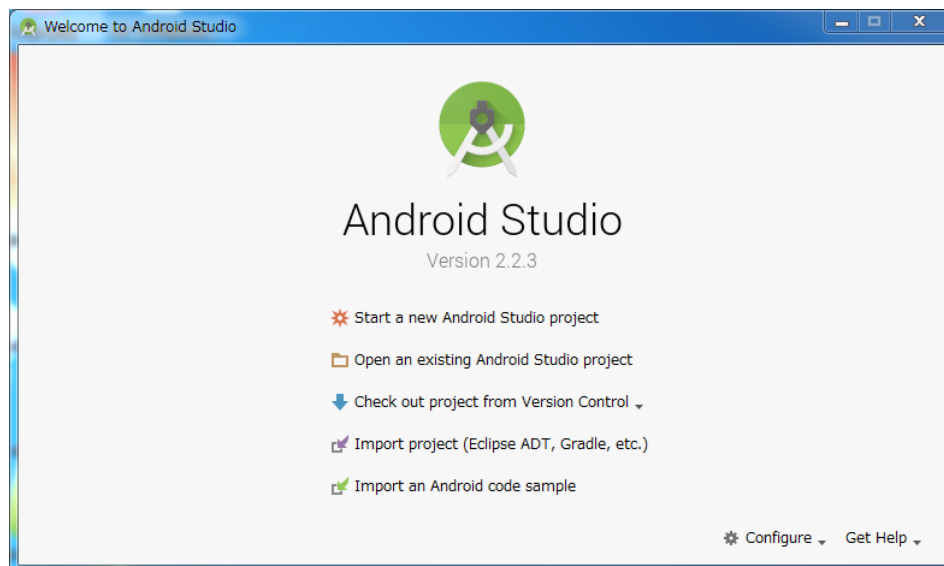
#### 5.1 Installation

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.

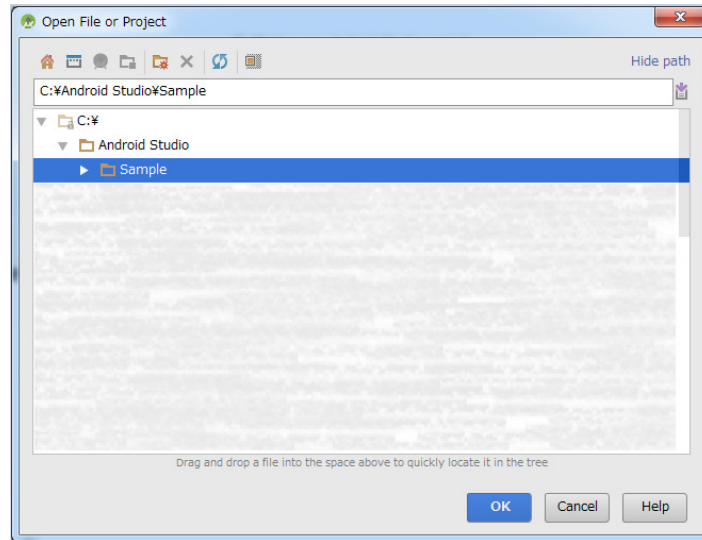
The procedures are shown below.

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

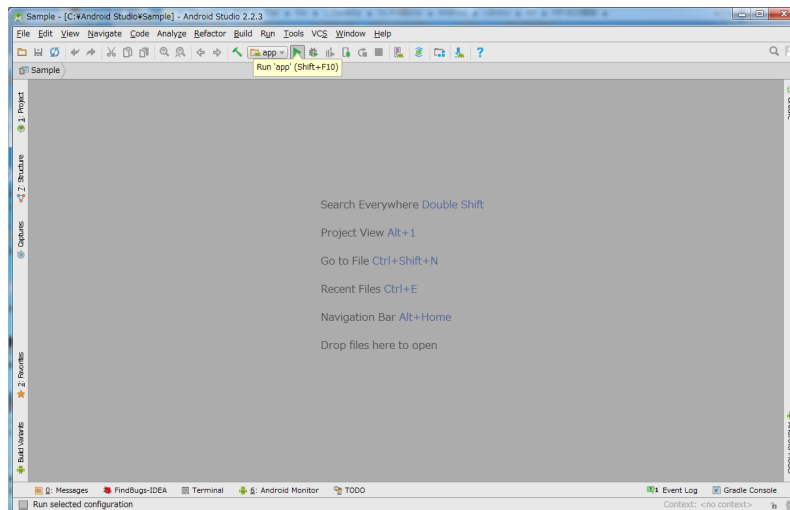




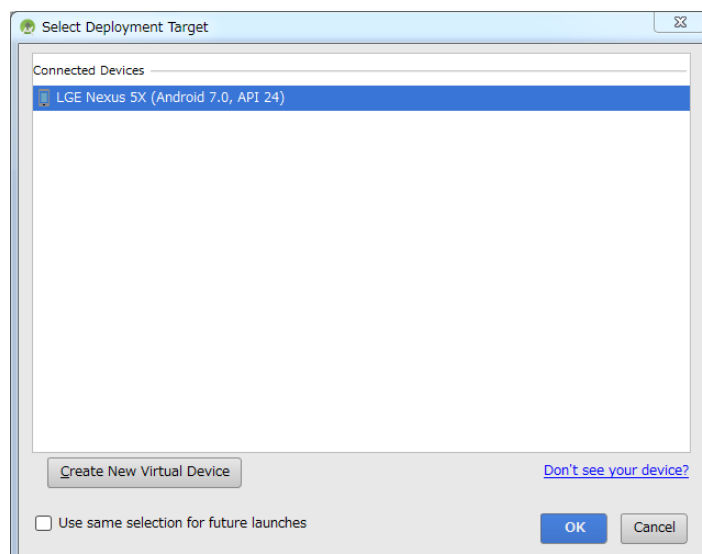
(3) Select the folder placed in the step (1), and click [OK].



(4) Click [Run 'app'].



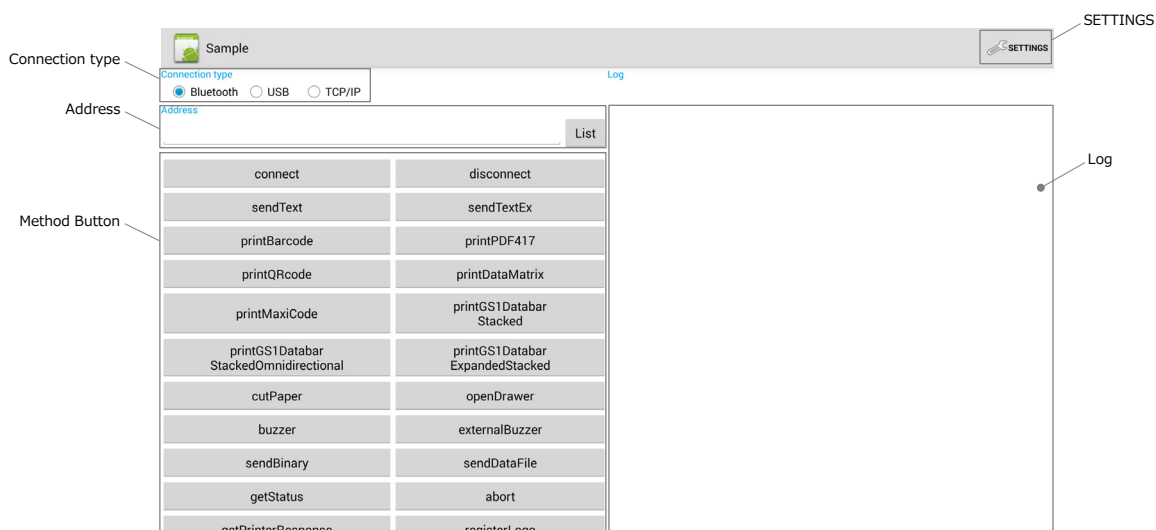
(5) Select the device, and click [OK].




## 5.2 Screen Layout

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 TCP/IP, enter the MAC address. For automatic input: By tapping the [List] button, the information of printers found by <b>startDiscoveryPrinter</b> is displayed in a list. When selecting a printer from the displayed list, the 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].

Sample	
Send timeout 10000msec	
Receive timeout 10000msec	
Socket keeping time 300000msec	
International character USA	
Code page Code Page 1252	
Secure connection Secure connection	<input checked="" type="checkbox"/>
SDK version X.X.X	

## 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.

## 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	£	¥	ℙ	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	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	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	Û	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	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¡	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	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	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Ş	ş
A0	á	í	ó	ú	ñ	Ñ	Ğ	ğ	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	ó	β	ô	ò	õ	õ	μ	×	ú	û	ü	ì	ÿ	-	'	
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	⌘	⌘	⌘		†	‡	§	¶	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
C0	L	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
E0	ω	ά	έ	ή	ï	í	ó	ύ	ÿ	ώ	À	É	Η	Ι	Ο	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	€	‘	’	“	”	•	-	-	~	™	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	Ĺ	ĺ	Š	š	Ö	Ü	ř	ť	Ł	×	č
A0	á	í	ó	ú	À	à	Ž	ž	Ę	ę	¬	ž	Č	š	«	»
B0	␣	␣	␣			Á	Â	Ě	Š					ž	ž	
C0	␣	␣	␣	␣	␣	Ě	ě	␣	␣	␣	␣	␣	␣	␣	␣	␣
D0	đ	Đ	Ď	Ě	ď	Ň	í	î	ě	␣	␣	␣	␣	␣	␣	␣
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	ú	ý	ý	ť	'	
F0	-	"	˘	˘	˘	§	÷	°	°	°	°	ú	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	Ã	©			¶	¶	¢	¥	₱
C0	L	⊥	T	└	└	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	Ј	Г	■	■	І	İ	■
E0	ó	β	ô	ò	õ	õ	μ	þ	þ	ú	û	ü	ý	ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2		■

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

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	¶	•	•	•	•	»	Ł	”	Ł	ž	
C0	Ř	Á	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
D0	Đ	Ň	Ň	Ó	Ô	Õ	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	õ	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	ъ	ѓ	;	ѓ	„	...	†	‡	€	‰	Љ	<	Њ	ќ	ћ	џ
90	ђ	‘	;	“	”	•	-	-	™	љ	>	њ	ќ	ћ	џ	
A0	ѣ	ѝ	Ј	Ѡ	Г	І	Ѕ	Ё	Є	«	¬	-	®	İ		
B0	°	±	І	і	г	μ	¶	•	ё	№	є	»	ј	ѕ	ѕ	ї
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

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	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
B0	°	±	²	³	´	μ	¶	•	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
C0	ı	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
D0	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί	
E0	ύ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ό	ύ	ώ	

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	Ń	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

**SII**



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](mailto: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.)