



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

Rev.01

[Products]

DSP-A01 Series

Seiko Instruments Inc.

Copyright © 2019 by Seiko Instruments Inc.
All rights reserved.

Android™ is a trademark of Google LLC.

Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Other names may be trademarks of their respective owners.

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 Product

The product covered by this manual is as follows.

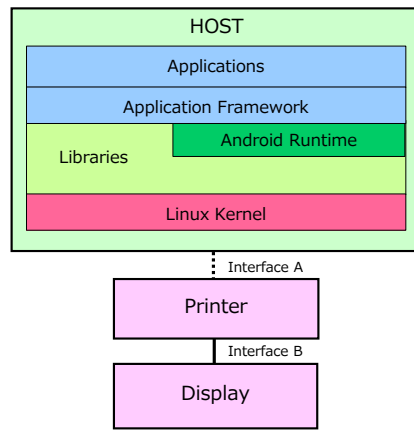
Product	Interface	Description in this manual
DSP-A01-1	USB	Display

Supported Driver

SII driver is required for using Display.
Supported SII drivers are described below.
There are 2 types for connecting Display.

1. When Display is used by connecting to SII printer (hereinafter described as "use via a printer")

SII Driver	Interface A	Printer	Interface B
"SII Printer Driver for Windows" for RP-F10/G10 series	Bluetooth TCP/IP USB	RP-F10 series	USB



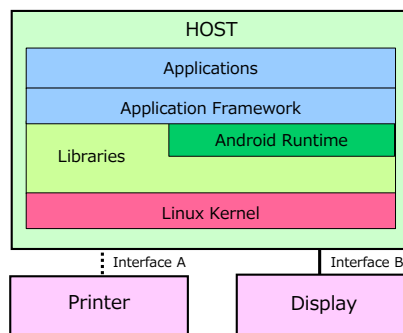
Configuration image

2. When Display is used alone (hereinafter described as "use alone")

SII Driver	Interface A	Printer ^{*1}	Interface B
"SII Printer Driver for Windows" for DSP-A01 series "SII Printer Driver for Windows" for RP-F10/G10 series	-	-	USB

*1: SII printer that supports Display is as follows.

- RP-F10 series



Configuration image

Terms

The terms used in this manual are defined as below.

Printer

Term	Description
Technical Reference for Printer	The following Technical Reference. · "RP-F10/G10 SERIES THERMAL PRINTER TECHNICAL REFERENCE"
Printer command	Command for controlling the printer described in "Technical Reference for Printer".

Display

Term	Description
Technical Reference for Display	The following Technical Reference. · "DSP-A01 SERIES CUSTOMER DISPLAY TECHNICAL REFERENCE"
Display command	Command for controlling Display described in "Technical Reference for Display".
Slide	The image data of the screen size (Width 480 px × Height 272 px). Displays as a standby screen and as a backscreen superimposed on templates.
Template	The stylized form having elements that can set attributes such as drawing areas and mapping positions. The elements include text elements (text data), img elements (image data), barcode elements (barcode data), and qr elements (QR Code data). To register templates, define a map ID each for an element to place. Registered image data or text data is shown on Display by updating the screen after selecting a template and specifying its map ID. The data in the template is required to be specified XML file format. The maximum size of template data is width 480 px × height 272 px.
Map ID	An ID defined to an element which is holding positional information or modification information when a template is registered.
Macro	A function to register multiple APIs in order of execution, and execute automatically when an event occurs.
Event	An event which is defined by "Event Notification" in Display commands.

Table of Contents

Chapter 1	Product Overview	1-1
1.1	Functions Provided by SII Print Class Library.....	1-1
1.2	SII Print Class Library Overview	1-1
1.2.1	SII Print Class Library Configuration	1-1
1.2.2	Functions Provided by Library.....	1-2
1.2.3	Registered Data in Display at the Shipping.....	1-2
Chapter 2	Product Specifications	2-1
2.1	Operating Environment.....	2-1
Chapter 3	How to Use the Library	3-1
3.1	Android Application Development Environment	3-1
3.2	Provided Files	3-2
3.3	Build the Library into Android Studio Project	3-3
3.4	Use Developed Android Application on Android Device	3-5
Chapter 4	Functions of the Library	4-1
4.1	API Reference	4-1
4.1.1	PrinterManager Class.....	4-2
PrinterManager	Constructor	4-10
connect	Start communicating with printer (Bluetooth)	4-10
connect	Start communicating with device (USB)	4-11
connect	Start communicating with printer (TCP/IP)	4-11
disconnect	Stop communicating with device	4-12
sendBinary	Send binary data.....	4-12
sendDataFile	Send specified file	4-12
getPrinterResponse	Get various responses from printer	4-13
startDiscoveryPrinter	Start printer search (Bluetooth)	4-14
startDiscoveryPrinter	Start printer search (USB)	4-14
startDiscoveryPrinter	Start printer search (TCP/IP)	4-15
startDiscoveryDevice	Start device search (USB)	4-15
cancelDiscoveryPrinter	Cancel printer search	4-16
getFoundPrinter	Get found printer information.....	4-16
getFoundDevice	Get found device information	4-16
getSendTimeout	Get send timeout period	4-16
setSendTimeout	Set send timeout period.....	4-16
getReceiveTimeout	Get receive timeout period	4-17
setReceiveTimeout	Set receive timeout period.....	4-17
getInternationalCharacter	Get international character set	4-17

setInternationalCharacter	Set international character set.....	4-17
getCodePage	Get codepage	4-18
setCodePage	Set codepage	4-18
getPrinterModel	Get device model.....	4-18
getPortType	Get connecting port type	4-19
isConnect	Verify connection state with device.....	4-19
getSocketKeepingTime	Get socket keeping time	4-19
setSocketKeepingTime	Set socket keeping time	4-19
defragment	Optimize memory area	4-20
initializeMemoryArea	Initialize memory area	4-20
showTemplate	Display template	4-21
showSlide	Display slide.....	4-22
enterStandbyMode	Display standby	4-22
executeMacro	Execute macro	4-23
turnOnScreen	Turn on/off screen	4-23
selectTemplate	Select template.....	4-24
setTemplateImageData	Set image data	4-24
selectTemplateTextObject	Select text element.....	4-25
setTemplateTextAlignment	Alignment of text data.....	4-26
setTemplateTextLeftMargin	Set left margin of text data	4-26
setTemplateTextLineSpacing	Set line spacing of text data	4-27
setTemplateTextBold	Set bold character of text data.....	4-27
setTemplateTextUnderline	Set underline of text data.....	4-28
setTemplateTextSize	Set character size of text data	4-29
setTemplateTextFont	Set character font of text data	4-29
setTemplateTextRegisteredFont	Set registered font of text data	4-30
setTemplateTextRightSpacing	Set right space of text data.....	4-30
setTemplateTextColor	Set character color of text data	4-31
setTemplateTextData	Input text data.....	4-32
setTemplateBarcodeData	Input barcode data.....	4-32
setTemplateQrCodeData	Input QR Code data.....	4-33
registerTemplate	Register template	4-35
unregisterTemplate	Delete template	4-35
registerImageData	Register image data	4-36
unregisterImageData	Delete image data	4-37
registerSlideData	Register slide data.....	4-37
unregisterSlideData	Delete slide data.....	4-38

registerUserDefinedCharacter	Register user-defined character	4-39
unregisterUserDefinedCharacter	Delete user-defined character	4-39
registerOptionFont	Register optional font.....	4-40
unregisterOptionFont	Delete optional font.....	4-41
controlMacroRegistration	Start/Finish of macro registration	4-41
getDisplayResponse	Get various response from Display	4-43
4.1.2 PrinterEvent Class.....		4-45
getEventType	Get end event	4-45
4.1.3 PrinterListener Interface.....		4-46
finishEvent	End event of device search	4-46
4.1.4 PrinterInfo Class.....		4-47
getDevicePath	Get device path	4-47
4.1.5 PrinterException Class.....		4-48
PrinterException	Constructor	4-49
getErrorCode	Get error codes.....	4-49

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.

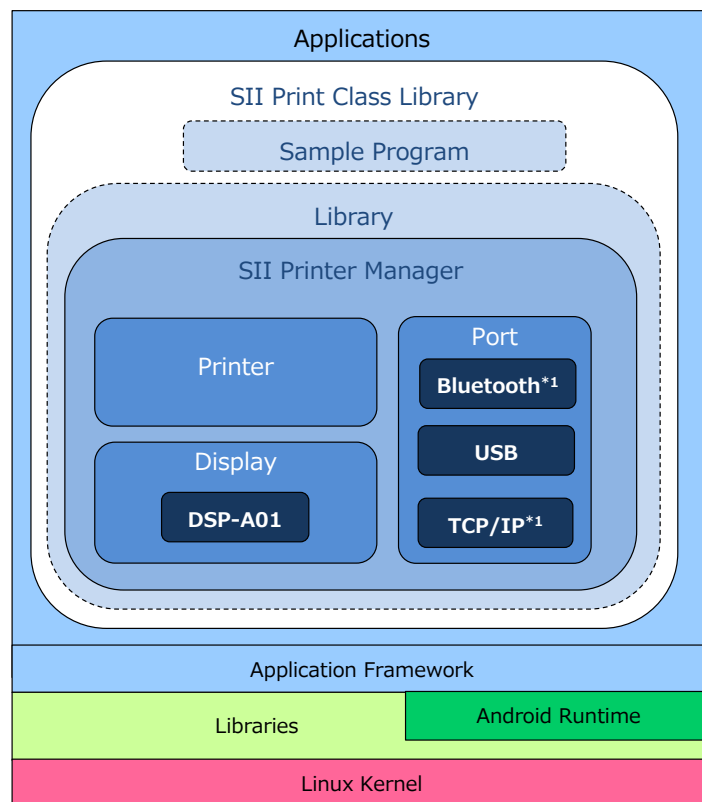
1.1 Functions Provided by SII Print Class Library

The SII print class library including the library and the sample program provides the functions to use Display described in "Introduction Target Product" in Android applications.
In addition, 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: It is valid only when Display is used via a printer.

1.2.2 Functions Provided by Library

By using the library, Android applications can easily send display data and Display commands to a device through communication port (Bluetooth^{*1}, USB or TCP/IP^{*1}) on an Android device. In addition, the applications can get the Display response.

The library provides the following functions:

- Connecting to / disconnecting from a device
- Sending data to a device (display data and/or Display commands^{*2})
- Sending a data file to a device (display data and/or Display commands^{*2})
- Getting various responses from Display
- Screen display control

*1: It is valid only when Display is used via a printer.

*2: To read responses from Display, use **getDisplayResponse**.

1.2.3 Registered Data in Display at the Shipping

In case of connecting Display to the printer to use templates, registered data in Display at the shipping may be added or changed without prior notice for the quality improvement.

A template which is specified appropriate encode is required to use depending on language settings or character codes to specify. See SII's Website for details about the data to be registered at the shipping.
<http://www.sii-ps.com/dspa01/>

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.

Display	Model		DSP-A01
	Communication Interface		USB
Android Device	Communication Port		USB ^{*1}
	OS	Android 5.0 (API 21)	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)	
	Supported Language		Japanese/English

^{*1}: The Android device needs to support USB host function.

Chapter 3

How to Use the 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.

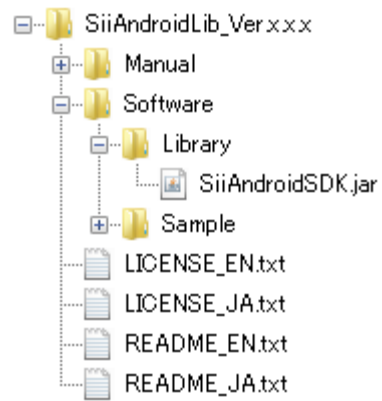


Figure 3-1

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

3.3 Build the 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).

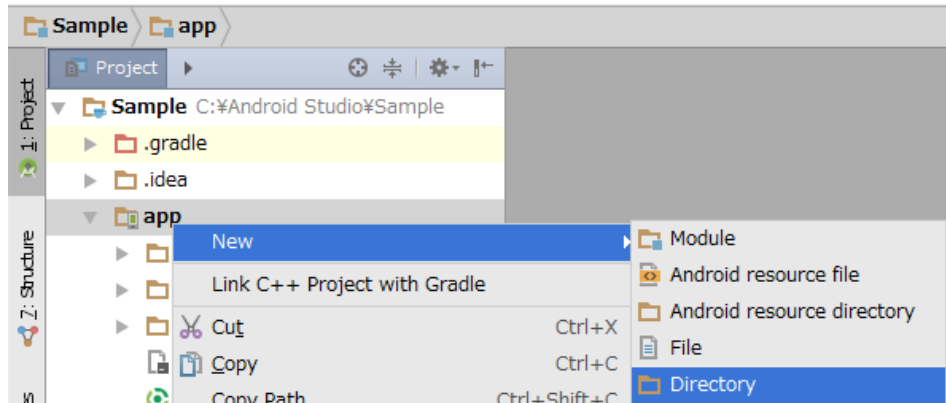


Figure 3-2

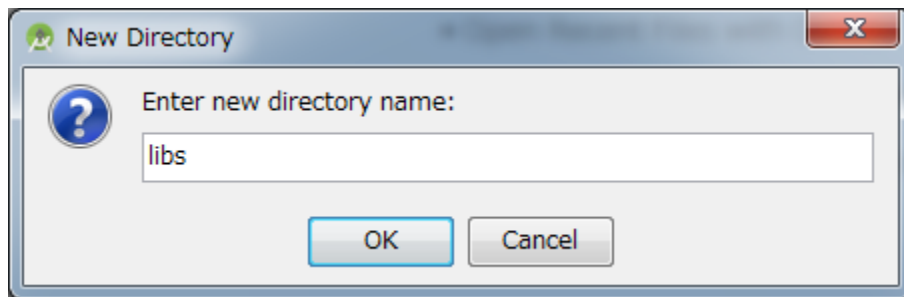


Figure 3-3

- (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.

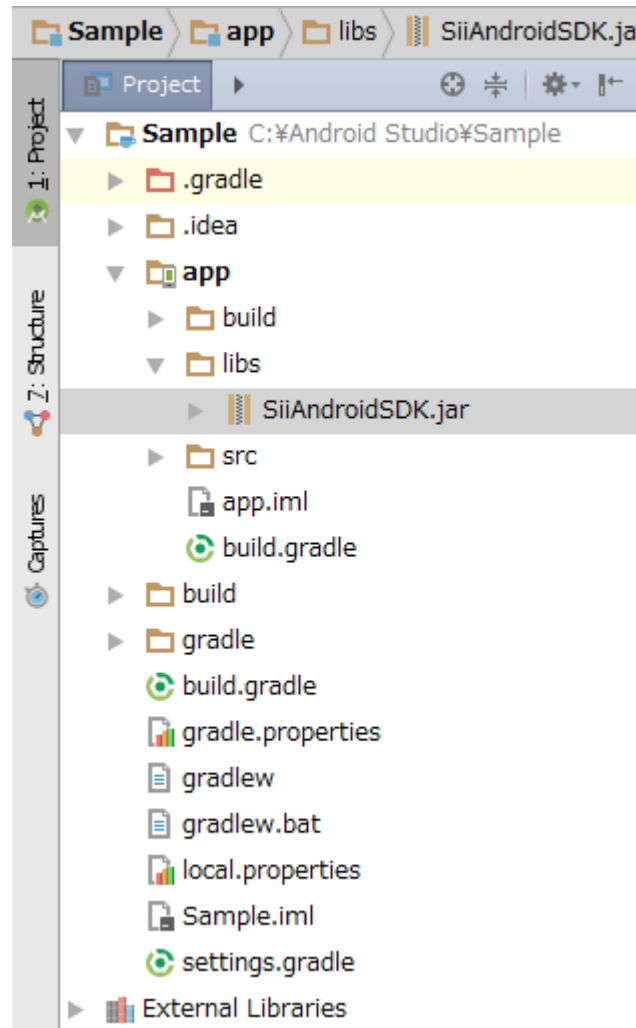


Figure 3-4

- (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)

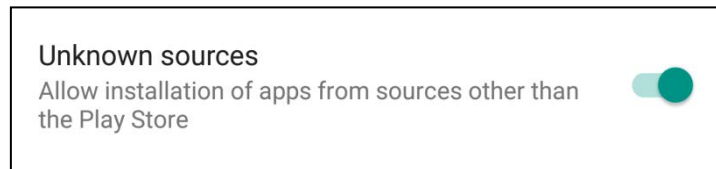


Figure 3-5

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

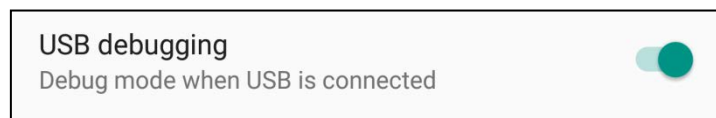


Figure 3-6

Chapter 4

Functions of the Library

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

4.1 API Reference

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

Class Name	Description	Supported ^{*1}
PrinterManager	Provides the API used for communication with the device and for displaying. See "4.1.1 PrinterManager Class " for more details.	✓
PrinterEvent	Provides the API that gets the end event when startDiscoveryPrinter is completed. See "4.1.2 PrinterEvent Class " for more details.	✓
PrinterListener	Interface for getting the end event when startDiscoveryPrinter or startDiscoveryDevice is completed. See "4.1.3 PrinterListener Interface " for more details.	✓
PrinterInfo	Stores the device information found by startDiscoveryPrinter or startDiscoveryDevice . See "4.1.4 PrinterInfo Class ".	✓
PrinterException	Exception class that is thrown at API call. See "4.1.5 PrinterException Class " for more details.	✓

*1: ✓: Supported, -: Not supported in DSP-A01

4.1.1 PrinterManager Class

(1) Method List

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

Name	Description	Supported ^{*1}
PrinterManager	Constructor	✓
connect	Start communicating with printer (Bluetooth)	✓
connect	Start communicating with device (USB)	✓
connect	Start communicating with printer (TCP/IP)	✓
disconnect	Stop communicating with device	✓
sendBinary	Send binary data	✓
sendDataFile	Send specified file	✓
getPrinterResponse	Get various responses from printer	✓
startDiscoveryPrinter	Start printer search (Bluetooth)	✓
startDiscoveryPrinter	Start printer search (USB)	✓
startDiscoveryPrinter	Start printer search (TCP/IP)	✓
startDiscoveryDevice	Start device search (USB)	✓
cancelDiscoveryPrinter	Cancel printer search	–
getFoundPrinter	Get found printer information list	✓
getFoundDevice	Get found device information list	✓
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 device model	✓
getPortType	Get connecting port type	✓
isConnect	Verify connection state with device	✓
getSocketKeepingTime	Get socket keeping time	✓
setSocketKeepingTime	Set socket keeping time	✓
defragment	Optimize memory area	✓
initializeMemoryArea	Initialize memory area	✓
showTemplate	Display template	✓
showSlide	Display slide	✓
enterStandbyMode	Display standby	✓
executeMacro	Execute macro	✓
turnOnScreen	Turn on/off screen	✓
selectTemplate	Select template	✓
setTemplateImageData	Set image data	✓

Name	Description	Supported ^{*1}
selectTemplateTextObject	Select text element	✓
setTemplateTextAlignment	Alignment of text data	✓
setTemplateTextLeftMargin	Set left margin of text data	✓
setTemplateTextLineSpacing	Set line spacing of text data	✓
setTemplateTextBold	Set bold character of text data	✓
setTemplateTextUnderline	Set underline of text data	✓
setTemplateTextSize	Set character size of text data	✓
setTemplateTextFont	Set character font of text data	✓
setTemplateTextRegisteredFont	Set registered font of text data	✓
setTemplateTextRightSpacing	Set right space of text data	✓
setTemplateTextColor	Set character color of text data	✓
setTemplateTextData	Input text data	✓
setTemplateBarcodeData	Input barcode data	✓
setTemplateQrCodeData	Input QR Code data	✓
registerTemplate	Register template	✓
unregisterTemplate	Delete template	✓
registerImageData	Register image data	✓
unregisterImageData	Delete image data	✓
registerSlideData	Register slide data	✓
unregisterSlideData	Delete slide data	✓
registerUserDefinedCharacter	Register user-defined character	✓
unregisterUserDefinedCharacter	Delete user-defined character	✓
registerOptionFont	Register optional font	✓
unregisterOptionFont	Delete optional font	✓
controlMacroRegistration	Start/Finish macro registration	✓
getDisplayResponse	Get various responses from Display	✓

*1: ✓ : Supported, - : Not supported in DSP-A01

(2) Constant List

① International character set

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

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

② Codepage

Constants used for setting/getting the 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 ^{*1}	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 ^{*12}	Arabic (Code Page864)	37
CODE_PAGE_1250	Central European (Code Page1250)	45
CODE_PAGE_1251	Cyrillic (Code Page1251)	46
CODE_PAGE_1253 ^{*3}	Greek (Code Page1253)	47

Constant Name	Description	Value
CODE_PAGE_1254	Turkish (Code Page1254)	48

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

*2: Font B cannot be displayed.

*3: 00AAh of the Unicode cannot be displayed.

③ Device model

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

Constant Name	Description	Value
PRINTER_MODEL_RP-FG10^{*1}	RP-F10	301
PRINTER_MODEL_DSP_A01	DSP-A01	303
PRINTER_MODEL_DEFAULT	Default value of device model	284

*1: It is valid only when Display is used via a printer.

④ Port type

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

Constant Name	Description	Value
PRINTER_TYPE_BLUETOOTH^{*1}	Bluetooth	0
PRINTER_TYPE_USB	USB	1
PRINTER_TYPE_TCP^{*1}	TCP/IP	2

*1: It is valid only when Display is used via a printer.

⑤ Printer response type

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

Constant Name	Description	Value
PRINTER_RESPONSE_REQUEST	Execution response request	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

⑥ Display response type

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

Constant Name	Description	Value
DISPLAY_RESPONSE_REQUEST	Execution response request	0
DISPLAY_RESPONSE_USER_AREA	Send remaining capacity of user area	1
DISPLAY_RESPONSE_TEMPLATE_ID_LIST	Send template ID	2
DISPLAY_RESPONSE_IMAGE_ID_LIST	Send image ID	3
DISPLAY_RESPONSE_SLIDE_ID_LIST	Send slide ID	4
DISPLAY_RESPONSE_TEMPLATE_LABEL	Send template name	5
DISPLAY_RESPONSE_IMAGE_LABEL	Send image name	6
DISPLAY_RESPONSE_SLIDE_LABEL	Send slide name	7

(3) Enumerated Constant List

① Bold character (CharacterBold)

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

Constant Name	Description
BOLD_CANCEL	Cancel bold character
BOLD	Specify bold character

② Underline (CharacterUnderline)

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

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

③ Character font (CharacterFont)

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

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

④ 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_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

⑤ Alignment (PrintAlignment)

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

Constant Name	Description
ALIGNMENT_LEFT	Left aligned
ALIGNMENT_CENTER	Align center
ALIGNMENT_RIGHT	Right aligned

⑥ Module size (ModuleSize)

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

Constant Name	Description	Method to Use
QR_MODULE_SIZE_2	2 dots	setTemplateQrCodeData
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	

⑦ 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
QR_ERROR_CORRECTION_L	Error correction level L	setTemplateQrCodeData
QR_ERROR_CORRECTION_M	Error correction level M	
QR_ERROR_CORRECTION_H	Error correction level H	
QR_ERROR_CORRECTION_Q	Error correction level Q	

⑧ Memory area (MemoryArea)

Constants of enumerated type used for operating memory area are shown in the following table.

Constant Name	Description
MEMORY_DISPLAY_USERMEMORY	User area

⑨ Registered font (RegisteredFont)

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

Constant Name	Description
FONT_STANDARD	Standard font
FONT_OPTION	Optional font

⑩ QR data mode (QrDataMode)

Constants of enumerated type used for QR data mode are shown in the following table.

Constant Name	Description
QRDATAMODE_NUMERIC	Numeric mode
QRDATAMODE_ALPHANUMERIC	Alphanumeric mode
QRDATAMODE_8BITBYTE	8-bit byte mode
QRDATAMODE_KANJI	Kanji mode
QRDATAMODE_MIXTURE	Mixed mode

⑪ QR quiet zone (QrQuietZone)

Constants of enumerated type used for QR quiet zone are shown in the following table.

Constant Name	Description
QRQUIETZONE_EXIST	Set QR quiet zone on
QRQUIETZONE_NONE	Set QR quiet zone off

⑫ Macro registration processing (MacroRegistrationFunction)

Constants of enumerated type used for macro registration processing are shown in the following table.

Constant Name	Description
MACRO_REGISTRATION_CLEAR	Cancel macro registration processing
MACRO_REGISTRATION_START	Start macro registration processing
MACRO_REGISTRATION_REGIST	Finish macro registration and macro registration processing

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)**

Starts communication with a printer by Bluetooth connection.

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

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

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

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

Parameter *printerModel* Device model constant for Bluetooth connection
See "4.1.1(2)③ Device model" for available constants.

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

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

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

Description This method is valid only when Display is used via a printer.

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

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

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

Starts communication with a device by USB connection.

The method of syntax (a) communicates with a device of the specified model.

The method of syntax (b) communicates with a specified model and a device of path.

Syntax	(a) public void connect (int <i>printerModel</i>) throws PrinterException
	(b) public void connect (int <i>printerModel</i> , String <i>address</i>) throws PrinterException
Parameter	<i>printerModel</i> Device model constant for USB connection See "4.1.1(2)③ Device model" for available constants.
	<i>address</i> Path of USB device file Example: "/dev/bus/usb/001/002"
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class " for details of the error.
Description	Call this method before using other PrinterManager class methods. The device specified by <i>printerModel</i> and/or <i>address</i> is connected. Also, device initial setting is performed at the connection based on the specified <i>printerModel</i> .

Starts communication with a printer by TCP/IP connection.

Syntax	public void connect (int <i>printerModel</i> , String <i>address</i>) throws PrinterException
Parameter	<i>printerModel</i> Device model constant for Ethernet connection See "4.1.1(2)③ Printer model" for available constants.
	<i>address</i> IP address Example: "192.168.0.190"
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class " for details of the error.
Description	This method is valid only when Display is used via a printer. 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 <i>address</i> . TCP ports 9100 and 26100 are used for communication. Also, printer initial setting is performed at the connection based on the specified <i>printerModel</i> ..

• **About socket creation/deletion of the library under TCP/IP connection**

The library retains the created socket until **disconnect** after **connect**.
 Also, it is impossible to connect to the same printer from another application until **disconnect**.

Based on the completion of data transmission to the printer, the socket is deleted once after socket keeping time set by **socketKeepingTime** elapses, but the socket is created again immediately.

disconnect**Stop communicating with device**

Stops communicating with the device.

Syntax `public void disconnect() throws PrinterException`

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

Note It is recommended to get the execution response by **PRINTER_RESPONSE_REQUEST** of **getPrinterResponse** or **DISPLAY_RESPONSE_REQUEST** of **getDisplayResponse** before executing this method. If not, the communication is disconnected by this method before all data sending from Android device to the device is completed, and a part of the data may be lost.
When **getPrinterResponse** or **getDisplayResponse** is not executed, evaluate that there is no problem in your program before using.

sendBinary**Send binary data**

Sends binary data to the device.

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

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

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

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

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

sendDataFile**Send specified file**

Sends file data.

Syntax `public void sendDataFile(String fileName) throws PrinterException`

Parameter *fileName* Name of the data file to send to the device
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.

- .bin, .dat
Data is sent to the device as the binary data without conversion.

Exception **PrinterException**

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

See "4.1.5 **PrinterException Class**" for details of the error.

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

getPrinterResponse

Get various responses from printer

Gets response data from the printer.

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

Parameter *id* Printer response type constant
See "4.1.1(2)⑤ Printer 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 printer response type constant.
See the following table for buffer types.

Response Type Constant	
Parameter	Description
PRINTER_RESPONSE_REQUEST (Execution response request)	
<i>buf</i>	Specify an int type array of length 1. Specify 0 to 15 (00h to 0Fh) for <i>buf[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).
PRINTER_RESPONSE_USER_AREA (Send remaining capacity of user area)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the remaining capacity of the user area is stored as a numerical value in bytes.
PRINTER_RESPONSE_ARRANGE_USER_AREA (Send remaining 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.
PRINTER_RESPONSE_NV_GRAPHICS (Send NV graphics memory capacity)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the NV graphics memory capacity is stored as a numerical value in bytes.
PRINTER_RESPONSE_KEY_CODE (Send key code list of defined NV graphics)	
<i>buf</i>	Specify an ArrayList<String> type array. When the response is retrieved successfully, the key code of NV graphics is stored as a string array. Example: <i>buf.size()</i> = 3, <i>buf[0]</i> = "22", <i>buf[1]</i> = "23", <i>buf[2]</i> = "24", etc.

Searches for SII printer.

Syntax	public void startDiscoveryPrinter (PrinterListener <i>listener</i> , int <i>retry</i> , int <i>timeout</i>) throws PrinterException	
Parameter	<i>listener</i>	Instance of PrinterListener Completion of this method or cancellation by cancelDiscoveryPrinter is notified to the user application as an end event by finishEvent through the instance set in <i>listener</i> .
	<i>retry</i>	Number of retry (times) Sends the local broadcast packet the number of times set by <i>retry</i> . 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.
	<i>timeout</i>	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 <i>timeout</i> 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.
Error	PrinterException is thrown when an error occurs while calling this method.	
Description	This method searches for SII printer. The printer information of the found printer is stored to PrinterInfo 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 <i>listener</i> .	

Searches for SII device.

Syntax	public void startDiscoveryDevice (PrinterListener <i>listener</i> , int <i>deviceType</i>) throws PrinterException	
Parameter	<i>listener</i>	Instance of PrinterListener Completion of this method or cancellation by cancelDiscoveryDevice is notified to the user application as an end event by finishEvent through the instance set in <i>listener</i> .
	<i>deviceType</i>	Port type Specify PRINTER_TYPE_USB .
Error	PrinterException is thrown when an error occurs while calling this method.	
Description	This method searches for SII device. The device information of the found device is stored to PrinterInfo 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 <i>listener</i> .	

cancelDiscoveryPrinter

Cancel printer search

This method is not supported.

Syntax public void **cancelDiscoveryPrinter**()

getFoundPrinter

Get found printer information

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

Syntax public ArrayList<**PrinterInfo**> **getFoundPrinter**()

Return value ArrayList of **PrinterInfo** class

getFoundDevice

Get found device information

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

Syntax public ArrayList<**PrinterInfo**> **getFoundDevice**()

Return value ArrayList of **PrinterInfo** class

getSendTimeout

Get send timeout period

Gets the send timeout period.

Syntax public int **getSendTimeout**()

Return value Send timeout period (millisecond: ms)

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

setSendTimeout

Set send timeout period

Sets the send timeout period.

Syntax public void **setSendTimeout**(int *sendTimeout*)

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

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

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

The set timeout period becomes effective at the next data sending.

getReceiveTimeout

Get receive timeout period

Gets the receive timeout period.

Syntax `public int getReceiveTimeout()`

Return value Receive timeout period (millisecond: ms)

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

setReceiveTimeout

Set receive timeout period

Sets the receive timeout period.

Syntax `public void setReceiveTimeout(int receiveTimeout)`

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

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

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

The set timeout period becomes effective at the next data receiving.

getInternationalCharacter

Get international character set

Gets the value of international character set.

Syntax `public int getInternationalCharacter()`

Return value See "4.1.1(2)① International character set" for details of the value.

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

Character codes whose display result varies depending on the international character set configuration

0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E

setInternationalCharacter

Set international character set

Sets the value of international character set.

Syntax `public void setInternationalCharacter(int internationalCharacter)`

Parameter *internationalCharacter* International character set constant
See "4.1.1(2)① International character set" for available constants.
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)② Codepage" for details of the value.

setCodePage	Set codepage
--------------------	---------------------

Sets the value of codepage.

Syntax public void **setCodePage**(int *codePage*)

Parameter *codePage* Codepage constant
 See "4.1.1(2)② Codepage" for available constants.
 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

When the text data is sent by **sendDataFile** or **setTemplateTextData**, an encoding to be used is changed depending on this property setting.

getPrinterModel	Get device model
------------------------	-------------------------

Gets the value of the connecting device model.

Syntax public int **getPrinterModel**()

Return value See "4.1.1(2)③ Device model" for details of the value.
PRINTER_MODEL_DEFAULT is returned when **isConnect** is false.

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

getPortType

Get connecting port type

Gets the port type used for connecting with the device.

Syntax public int **getPortType**()

Return value See "4.1.1(2)④ Port type" for details of the value.
PRINTER_TYPE_BLUETOOTH is returned when **isConnect** is false.

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

isConnect

Verify connection state with device

Verifies connection state with the device.

Syntax public boolean **isConnect**()

Return value true: Connected to a device
 false: Not connected to a device

Description When the data transmission is failed, the communication with the device 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 is valid only when Display is used via a printer.
Setting is possible by this method regardless of whether **isConnect** is true or false.

setSocketKeepingTime

Set socket keeping time

Sets the socket keeping time.

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

Valid range 60000 to 300000 (millisecond: ms)
When the value is specified less than 60000, the time is set to 60000 ms.
When the value is specified more than 300000, the time is set to 300000 ms.

Default 300000

Description This method is valid only when Display is used via a printer.
Setting is possible by this method regardless of whether **isConnect** is true or false.

For the socket keeping time, specify a time equal to Receive Timeout of the printer to be connected. See the printer command "Set Wireless LAN Communication" described in "RP-F10/G10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details of Receive Timeout.

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

Optimizes the memory area.

Syntax public void **defragment**(MemoryArea *memoryArea*) throws **PrinterException**

Parameter *memoryArea* Memory area
See "4.1.1(3)⑧ Memory area (MemoryArea)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description Display is changed to Standby mode when this method is executed. A selecting template is deselected.
It may take several minutes for optimization.
Do not turn the device power off during optimization.

In use via a printer, this method is ignored when Display is not connected to the printer.

Initializes the memory area.

Syntax public void **initializeMemoryArea**(MemoryArea *memoryArea*) throws **PrinterException**

Parameter *memoryArea* Memory area
See "4.1.1(3)⑧ Memory area (MemoryArea)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description Display is changed to Standby mode when this method is executed. A selecting template is deselected.
It may take several minutes for initialization.
Do not turn the device power off during initialization.

In use via a printer, this method is ignored when Display is not connected to the printer.

Note Registered data in following methods is deleted when the memory area is initialized after specifying **MEMORY_DISPLAY_USERMEMORY**.

- **registerTemplate**
- **registerImageData**
- **controlMacroRegistration**
- **registerUserDefinedCharacter**
- **registerOptionFont**

In addition, part of data which is registered at the shipping to use for the system is deleted either.

Therefore, the guide screen cannot be displayed when an error occurs in the printer.
The used memory area can be reused after executing **defragment**.

Displays the template on the screen.

Syntax public void **showTemplate**(int *time_ms*) throws **PrinterException**

Parameter *time_ms* Display time (ms: millisecond)
Specify display time on the screen with *time_ms* (ms).
The valid range is 0 to 25500.
When the value exceeds 0 and less than 100 is specified, the time is set to 100 ms.
When the value exceeding 25500 is specified, the time is set to 25500 ms.

For macro registration:
When 0 is specified with *time_ms*, the template is shown continuously.
When other than 0 is specified with *time_ms*, a next template is shown after the display time is elapsed.

For other than macro registration:
When 0 is specified with *time_ms*, the template is shown continuously.
When other than 0 is specified with *time_ms*, the template returns to a previous template after the display time is elapsed.
In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the template which was updated with the display time 0.

Exception **PrinterException**

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

See "4.1.5 **PrinterException Class**" for details of the error.

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

Description Updates the screen, and displays data being specified with the following methods.

- **selectTemplate**
- **setTemplateImageData**
- **selectTemplateTextObject**
- **setTemplateTextAlignment**
- **setTemplateTextLeftMargin**
- **setTemplateTextLineSpacing**
- **setTemplateTextBold**
- **setTemplateTextUnderline**
- **setTemplateTextSize**
- **setTemplateTextFont**
- **setTemplateTextRightSpacing**
- **setTemplateTextColor**
- **setTemplateTextData**
- **setTemplateBarcodeData**
- **setTemplateQrCodeData**

In use via a printer, this method is ignored when Display is not connected to the printer.

Displays the slide on the screen.

Syntax	public void showSlide (int <i>slideID</i> , int <i>time_ms</i>) throws PrinterException	
Parameter	<i>slideID</i>	<p>Slide ID</p> <p>Specify the ID of slide data which was registered at registerSlideData. The valid range is 0 to 91.</p> <p>This method is ignored when slide data is not registered in the specified ID.</p>
	<i>time_ms</i>	<p>Display time (ms: millisecond)</p> <p>Specify display time on the screen with <i>time_ms</i> (ms). The valid range is 0 to 25500.</p> <p>When the value exceeds 0 and less than 100 is specified, the time is set to 100 ms.</p> <p>When the value exceeding 25500 is specified, the time is set to 25500 ms.</p> <p>For macro registration:</p> <p>When 0 is specified with <i>time_ms</i>, the slide is shown continuously. When other than 0 is specified with <i>time_ms</i>, a next slide is shown after the display time is elapsed.</p> <p>For other than macro registration:</p> <p>When 0 is specified with <i>time_ms</i>, the slide is shown continuously. When other than 0 is specified with <i>time_ms</i>, the slide returns to a previous slide after the display time is elapsed.</p> <p>In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the slide which was updated with the display time 0.</p>
Exception	PrinterException	<p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error.</p> <p>When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>A selecting template is deselected.</p> <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>	

Changes Display to Standby mode.

Syntax	public void enterStandbyMode () throws PrinterException	
Exception	PrinterException	<p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error.</p> <p>When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>This method is ignored during Standby mode or Guide mode is being displayed.</p> <p>A selecting template is deselected.</p>	

In use via a printer, this method is ignored when Display is not connected to the printer.

executeMacro

Execute macro

Executes the macro.

Syntax public void **executeMacro**(int *macroID*, int *repeatCount*) throws **PrinterException**

Parameter	<i>macroID</i>	Macro ID Specify the macro ID which was registered at controlMacroRegistration . The valid range is 0 to 127. This method is ignored when the macro is not registered in the specified ID.
	<i>repeatCount</i>	The number of execution times Specify times to execute the macro. The valid range is 0 to 255. Continues the repeating when 0 is specified.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description A selecting template is deselected.

In use via a printer, this method is ignored when Display is not connected to the printer.

turnOnScreen

Turn on/off screen

Sets the screen backlight on/off.

Syntax public void **turnOnScreen**(boolean *isOn*) throws **PrinterException**

Parameter	<i>isOn</i>	Screen status Specify the screen status from following. true : backlight on false : backlight off
-----------	-------------	---

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

Description In use via a printer, this method is ignored when Display is not connected to the printer.

Selects the template to show on Display.

The method of syntax (a) selects slide data to be used for the template or the template background.

The method of syntax (b) selects a template.

Syntax (a) public void **selectTemplate**(int *templateID*, int *SlideID*) throws **PrinterException**

(b) public void **selectTemplate**(int *templateID*) throws **PrinterException**

Parameter	<i>templateID</i>	<p>Template ID</p> <p>Specify the ID of template to select.</p> <p>The valid range is 0 to 127.</p> <p>This method is ignored when the template is not registered in the specified ID.</p>
	<i>slideID</i>	<p>Slide ID</p> <p>Specify the ID of slide data to use for the background of the template.</p> <p>The valid range is 0 to 91.</p> <p>This method is ignored when slide data is not registered in the specified ID.</p>

Exception **PrinterException**

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

See "4.1.5 **PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the device is ended, and

PrinterException may be thrown. See **isConnect** for verifying the connection state with the device.

Description The data on the template is cleared when this method is executed.
The selected template is displayed to the screen when **showTemplate** is executed.

The selecting template is deselected when **showSlide**, **enterStandbyMode**, or **executeMacro** is executed.

The selecting template is deselected when the specified time of display is executed at **showTemplate**.

Use following templates depends on the values of **codePage** when characters other than 20h to 7Eh of ASCII character are input with **setTemplateTextData**.

·When **codePage** is **CODE_PAGE_KATAKANA**:

Use the template which encoding specifying is Shift_JIS.

·When **codePage** is other than **CODE_PAGE_KATAKANA**:

Use the template which encoding specifying is ISO-2022-JP.

In use via a printer, this method is ignored when Display is not connected to the printer.

Sets image data to show on a selecting template.

Syntax public void **setTemplateImageData**(int *mapID*, int *imageID*) throws **PrinterException**

Parameter	<i>mapID</i>	<p>Map ID</p> <p>The valid range is 0 to 63.</p> <p>This method is ignored when the specified map ID is not defined in the template.</p>
-----------	--------------	--

imageID Image ID
Specify the ID of image data which was registered at **registerImageData**.
The valid range is 0 to 63.
This method is ignored when image data is not registered in the specified ID.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description After specifying the map ID of the selecting template with this method, specify the image ID to map.
The specified image data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

This method is ignored when a template is not selected.
In use via a printer, this method is ignored when Display is not connected to the printer.

selectTemplateTextObject

Select text element

Selects the text element to start editing.

Syntax public void **selectTemplateTextObject**(int *mapID*) throws **PrinterException**

Parameter *mapID* Map ID
The valid range is 0 to 63.
This method is ignored when the specified map ID is not defined in the template.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description After specifying the map ID of the selecting template with this method, start to edit characters.

When a scroll is set with the text element of the specified map ID and this method is executed after **showTemplate**, the scroll is executed.
When a scroll is not set with the text element of the specified map ID and this method is executed after **showTemplate**, input text data is discarded.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When **showTemplate** is executed.

This method is ignored when a template is not selected.
In use via a printer, this method is ignored when Display is not connected to the printer.

Sets alignment to text data shown on Display.

Syntax public void **setTemplateTextAlignment**(PrintAlignment *alignment*) throws **PrinterException**

Parameter *alignment* Alignment
See "4.1.1(3)⑤ Alignment (PrintAlignment)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description Alignment is valid only as following cases.
·Text data is not entered in the specified map ID at **selectTemplateTextObject**.
·Text data entered in the map ID which is specified at **selectTemplateTextObject** is registered right after a line feed.

Specify the map ID at **selectTemplateTextObject** before executing this method.

Input text data at **setTemplateTextData** after executing this method.

The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

Sets left margin to text data shown on Display.

Syntax public void **setTemplateTextLeftAlignment**(int *margin*) throws **PrinterException**

Parameter *margin* Left margin (pixel: px)
The valid range is 0 to 479.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description The left margin setting is valid only as following cases.
·Text data is not entered in the specified map ID at **selectTemplateTextObject**.
·Text data entered in the map ID which is specified at **selectTemplateTextObject** is registered right after a line feed.

Specify the map ID at **selectTemplateTextObject** before executing this method.

Input text data at **setTemplateTextData** after executing this method.

The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

setTemplateTextLineSpacing

Set line spacing of text data

Sets line spacing to text data shown on Display.

Syntax public void **setTemplateTextLineSpacing**(int *spacing*) throws **PrinterException**

Parameter *spacing* Line spacing (pixel: px)
 The valid range is 0 to 255.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description The line spacing setting is valid only as following cases.
 ·Text data is not entered in the specified map ID at **selectTemplateTextObject**.
 ·Text data entered in the map ID which is specified at **selectTemplateTextObject** is registered right after a line feed.

When a scroll is set to a text element of the specified map ID, the line spacing to text data is not reflected.

Specify the map ID at **selectTemplateTextObject** before executing this method.

Input text data at **setTemplateTextData** after executing this method.

The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

setTemplateTextBold

Set bold character of text data

Sets bold characters to text data shown on Display.

Syntax public void **setTemplateTextBold**(CharacterBold *bold*) throws **PrinterException**

Parameter *bold* Bold character
 See "4.1.1(3)① Bold character (CharacterBold)" for available constants.

Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>From the text data after this method is executed, the bold characters are applied. The bold character can be set one by one.</p> <p>Specify the map ID at selectTemplateTextObject before executing this method. Input text data at setTemplateTextData after executing this method. The input text data is displayed to the screen when showTemplate is executed.</p> <p>This method setting is cleared under the following conditions.</p> <ul style="list-style-type: none"> • When selectTemplate is executed. • When other than 0 is specified at <i>time_ms</i> of showTemplate, and the specified display time has elapsed. • When showTemplate registered in executeMacro is executed. <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>

setTemplateTextUnderline

Set underline of text data

Sets underline to text data shown on Display.

Syntax	public void setTemplateTextUnderline (CharacterUnderline <i>underline</i>) throws PrinterException
Parameter	<p><i>underline</i> Underline</p> <p>See "4.1.1(3)② Underline (CharacterUnderline)" for available constants.</p>
Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>From the text data after this method is executed, the underlines are applied. The underline can be set one by one.</p> <p>Specify the map ID at selectTemplateTextObject before executing this method. Input text data at setTemplateTextData after executing this method. The input text data is displayed to the screen when showTemplate is executed.</p> <p>This method setting is cleared under the following conditions.</p> <ul style="list-style-type: none"> • When selectTemplate is executed. • When other than 0 is specified at <i>time_ms</i> of showTemplate, and the specified display time has elapsed. • When showTemplate registered in executeMacro is executed. <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>

Sets character size to text data shown on Display.

Syntax public void **setTemplateTextSize**(CharacterScale *scale*) throws **PrinterException**

Parameter *scale* Character scale
See "4.1.1(3)④ Character scale (CharacterScale)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description From the text data after this method is executed, the character sizes are applied.
The character size can be set one by one.

Specify the map ID at **selectTemplateTextObject** before executing this method.
Input text data at **setTemplateTextData** after executing this method.
The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

Sets a character font to text data shown on Display.

Syntax public void **setTemplateTextFont**(CharacterFont *font*) throws **PrinterException**

Parameter *font* Character font
See "4.1.1(3)③ Character font (CharacterFont)" for available constants.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description From the text data after this method is executed, the character fonts are applied.
The character font can be set one by one.

Specify the map ID at **selectTemplateTextObject** before executing this method.
Input text data at **setTemplateTextData** after executing this method.
The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

setTemplateTextRegisteredFont

Set registered font of text data

Sets the registered font used for text data to show on Display.

Syntax public void **setTemplateRegisteredFont**(RegisteredFont *font*) throws **PrinterException**

Parameter *font* Registered font
See "4.1.1(3)⑨ Registered font (RegisteredFont)" for available constants.
The registered font specifying is ignored when the optional font is not registered.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description From the text data after this method is executed, the registered fonts are applied.
The registered font can be set one by one.

Input text data at **setTemplateTextData** after executing this method.
The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

setTemplateTextRightSpacing

Set right space of text data

Sets the amount of right space to text data shown on Display.

Syntax public void **setTemplateTextRightSpacing**(int *spacing*) throws **PrinterException**

Parameter *space* The amount of character right space (pixel: px)
The valid range is 0 to 255.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description From the text data after this method is executed, the amount of character right space is applied.
The amount of right space can be set one by one.

Specify the map ID at **selectTemplateTextObject** before executing this method.
Input text data at **setTemplateTextData** after executing this method.
The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

setTemplateTextColor	Set character color of text data
-----------------------------	---

Sets the character color used for text data to show on Display.

Syntax `public void setTemplateTextColor(int color) throws PrinterException`

Parameter *color* Character color
The valid range is 0 to 0xfffff.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description This method can set character colors to text data.
The character colors can be set in RGB24 bit color. The set color is displayed in the color converted to 16 bit RGB555.

From the text data after this method is executed, the character colors are applied.
The character color can be set one by one.

Specify the map ID at **selectTemplateTextObject** before executing this method.
Input text data at **setTemplateTextData** after executing this method.
The input text data is displayed to the screen when **showTemplate** is executed.

This method setting is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

Inputs text data to show on Display.

Syntax public void **setTemplateTextData**(String *text*) throws **PrinterException**

Parameter *text* Text data to show on Display
Data size which is able to be specified it once is 1 to 1020 bytes.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description This method encodes input text data into text data which is enable to display on the basis of settings at **internationalCharacter** and **codePage**, and displays.

Select the template in **selectTemplate** before executing this method.
After specifying the map ID of the selecting template with **selectTemplateTextObject**, input text data with this method.
This method is ignored when the map ID is not specified in **selectTemplateTextObject**.
The input text data is displayed to the screen when **showTemplate** is executed.

The input text data is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

In use via a printer, this method is ignored when Display is not connected to the printer.

Specifies a map ID of the barcode element on a selecting template, and inputs barcode data.
The method of syntax (a) inputs data with character strings to display barcode.
The method of syntax (b) inputs data with byte arrays to display barcode.

Syntax (a) public void **setTemplateBarcodeData**(int *mapID*, String *text*) throws **PrinterException**

(b) public void **setTemplateBarcodeData**(int *mapID*, byte [] *data*) throws **PrinterException**

Parameter *mapID* Map ID
The valid range is 0 to 7.
This method is ignored when the specified map ID is not defined in the selecting template.

text Barcode data
The characters that can be input are ASCII characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').
The available number of characters is 1 to 150.
Barcode data which is not complying with barcode specification is ignored.

data Barcode data
The value that can be input is 00h to 7Fh.
The available number of data is 1 to 150.
Barcode data which is not complied with barcode specification is ignored.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description After specifying the map ID of the selecting template with this method, input barcode data.
The input barcode data is displayed to the screen when **showTemplate** is executed.

The input barcode data is cleared under the following conditions.

- When **selectTemplate** is executed.
- When other than 0 is specified at *time_ms* of **showTemplate**, and the specified display time has elapsed.
- When **showTemplate** registered in **executeMacro** is executed.

This method is ignored when a template is not selected.

In use via a printer, this method is ignored when Display is not connected to the printer.

setTemplateQrCodeData

Input QR Code data

Specifies a map ID of the qr element on a selecting template, and inputs QR Code data.

The method of syntax (a) inputs QR Code data with character strings.

The method of syntax (b) inputs QR Code data with character strings. The setting of selecting template is reflected to *moduleSize*, *errorCorrection*, *mode*, and *qrQuietZone*.

The method of syntax (c) inputs QR Code data with byte array.

The method of syntax (d) inputs QR Code data with byte array. The setting of selecting template is reflected to *moduleSize*, *errorCorrection*, *mode*, and *qrQuietZone*.

Syntax (a) public void **setTemplateQrCodeData**(int *mapID*,
ModuleSize *moduleSize*,
ErrorCorrection *errorCorrection*,
QrDataMode *mode*,
QrQuietZone *qrQuietZone*,
String *text*) throws **PrinterException**

(b) public void **setTemplateQrCodeData**(int *mapID*, String *text*) throws **PrinterException**

(c) public void **setTemplateQrCodeData**(int *mapID*,
ModuleSize *moduleSize*,
ErrorCorrection *errorCorrection*,
QrDataMode *mode*,
QrQuietZone *qrQuietZone*,
byte [] *data*) throws **PrinterException**

(d) public void **setTemplateQrCodeData**(int *mapID*, byte [] *data*) throws **PrinterException**

Parameter *mapID* Map ID
The valid range is 0 to 7.
This method is ignored when the specified map ID is not defined in the selecting template.

moduleSize Module size
See "4.1.1(3)⑥ Module size (ModuleSize)" for available constants.

<i>errorCorrection</i>	Error correction level See "4.1.1(3)⑦ Error correction level (ErrorCorrection)" for available constants.
<i>mode</i>	Data mode See "4.1.1(3)⑩ QR data mode (QrDataMode)" for available constants.
<i>qrQuietZone</i>	Quiet zone See "4.1.1(3)⑪ QR quiet zone (QrQuietZone)" for available constants.
<i>text</i>	QR Code data The characters that can be input are as follows: <ul style="list-style-type: none"> • ASCII code characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z'). • 8 bits Latin / Katakana based on JIS X 0201 • Shift-JIS code based on JIS X 0208 The available data size is 1 to 3909 bytes. QR Code data which is not complied with QR Code specification is ignored.
<i>data</i>	QR Code data QR Code data shown on Display. The value that can be input is 00h to FFh. The available number of data is 1 to 3909. QR Code data which is not complied with QR Code specification is ignored.

registerTemplate

Registers a template.

<i>label</i>	<p>Template name</p> <p>A name for identification can be specified to the template to be registered.</p> <p>The characters that can be specified are ASCII code characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').</p> <p>Do not use Unicode 00A5h ('¥').</p> <p>The available number of characters is 0 to 32.</p> <p>This <i>label</i> is optional. Specify null when the template name is not registered.</p> <p>The specified template name can be retrieved with getDisplayResponse.</p>
<i>filePath</i>	<p>File path of the template data to register in Display</p> <p>Supported file extension is .xml.</p> <p>The maximum data size that can be registered is 8192 bytes.</p> <p>See "Technical Reference for Display" for details on registration of the template data.</p>

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description Display is changed to Standby mode when this method is executed. A selecting template is deselected.

In use via a printer, this method is ignored when Display is not connected to the printer.

Note **Registered data at the shipping may be added or changed without prior notice for quality improvement.**

unregisterTemplate	Delete template
---------------------------	------------------------

Deletes the registered template.

Syntax public void **unregisterTemplate**(int *templateID*) throws **PrinterException**

Parameter *templateID* Template ID
Specify an ID of template to delete.
The valid range is 0 to 127.
This method is ignored when a template is not registered in the specified ID.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

Description Display is changed to Standby mode when this method is executed. A selecting template is deselected.

Used memory is not released even the template is deleted. The used memory can be reused after executing **defragment**.

In use via a printer, this method is ignored when Display is not connected to the printer.

Registers image data.

Syntax	public void registerImageData (int <i>imageID</i> , String <i>label</i> , String <i>filepath</i>) throws PrinterException	
Parameter	<i>imageID</i>	Image ID The valid range is 0 to 63. Do not specify the image IDs of 49 to 63 because they are being used for the system.
	<i>label</i>	Image name A name for identification can be specified to image data to be registered. The characters that can be specified are ASCII characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z'). Do not use Unicode 00A5h ('¥'). The available number of characters is 0 to 32. This <i>label</i> is optional. Specify null when the template name is not registered. The specified image name can be retrieved with getDisplayResponse .
	<i>filePath</i>	File path Specify the file name of image data to register. Supported file extensions are .jpg, .jpeg, and .png. However, even the supported extensions may not be registered depending on the format. The maximum file size that can be registered is 786432 bytes. The maximum data size that can be registered is 480 horizontal x 272 vertical pixels (px).
Exception	PrinterException	PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class " for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.
Description	Display is changed to Standby mode when this method is executed. A selecting template is deselected. In use via a printer, this method is ignored when Display is not connected to the printer.	
Note	<u>Registered data at the shipping may be added or changed without prior notice for quality improvement.</u>	

unregisterImageData	Delete image data
----------------------------	-------------------

unregisterImageData	Delete image data
----------------------------	-------------------

Deletes registered image data.

Syntax `public void unregisterImageData(int imageID) throws PrinterException`

Parameter	<i>imageID</i>	Image ID The valid range is 0 to 63. This method is ignored when image data is not registered in the specified ID.
-----------	----------------	--

Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error.</p> <p>When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
-----------	--

Description	Display is changed to Standby mode when this method is executed. A selecting template is deselected.
-------------	--

Used memory is not released even image data is deleted. The used memory can be reused after executing **defragment**.

In use via a printer, this method is ignored when Display is not connected to the printer.

registerSlideData	Register slide data
--------------------------	---------------------

registerSlideData	Register slide data
--------------------------	---------------------

Registers slide data.

Syntax public void **registerSlideData**(int *slideID*,
 String *label*,
 String *filePath*) throws **PrinterException**

Parameter	<i>slideID</i>	Slide ID The valid range is 0 to 91. Do not specify the slide IDs of 80 to 90 because they are being used for the system.
-----------	----------------	---

label

Slide name

A name for identification can be specified to slide data to be registered.

The characters that can be specified are ASCII characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').

Do not use Unicode 00A5h ('¥').

The available number of characters is 0 to 32.

This *label* is optional. Specify null when the template name is not registered.

The specified slide name can be retrieved with **getDisplayResponse**.

filePath

File path
Specify the file name of slide data to register.
Supported file extensions are .jpg, .jpeg, and .png.
However, even the supported extensions may not be registered depending on the format.
The maximum file size that can be registered is 786432 bytes.
The data size that can be registered is fixed to 480 horizontal x 272 vertical pixels (px).

Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>Display is changed to Standby mode when this method is executed. A selecting template is deselected.</p> <p>Execute showSlide to show the registered slide data. Specify the slide ID at showTemplate to use the registered slide data as a backscreen of the template.</p> <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>
Note	<p><u>Registered data at the shipping may be added or changed without prior notice for quality improvement.</u></p>

unregisterSlideData	Delete slide data
----------------------------	-------------------

Deletes registered slide data.

Syntax	public void unregisterSlideData (int <i>imageID</i>) throws PrinterException	
Parameter	<i>slideID</i>	<p>Slide ID</p> <p>The valid range is 0 to 91.</p> <p>This method is ignored when slide data is not registered in the specified ID.</p>
Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>	
Description	<p>Display is changed to Standby mode when this method is executed. A selecting template is deselected.</p> <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>	

registerUserDefinedCharacter	Register user-defined character
-------------------------------------	---------------------------------

Registers user-defined characters in Display.

Syntax	public void registerUserDefinedCharacter (String <i>filePath</i>) throws PrinterException	
Parameter	<i>filePath</i>	<p>File path</p> <p>Specify the file name of the user-defined characters to register. Supported file extension is .bin.</p> <p>See "Register User-Defined Character" of the Display command in "Technical Reference for Display" for details on the user-defined character data.</p>

Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>Display is changed to Standby mode when this method is executed. A selecting template is deselected. User-defined characters are overwritten when this method is executed with user-defined character registered status.</p> <p>Use the template which encoding specifying is Shift_JIS for displaying user-defined characters. Specify CODE_PAGE_KATAKANA for <i>codePage</i> of setCodePage before user-defined characters are displayed. Specify the character codes that can be specified for <i>text</i> of setTemplateTextData when user-defined characters are displayed. The character codes that can be specified are E000h to E05Dh.</p> <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>

unregisterUserDefinedCharacter	Delete user-defined character
---------------------------------------	-------------------------------

Deletes registered user-defined characters.

Syntax	public void unregisterUserDefinedCharacter () throws PrinterException
Exception	<p>PrinterException</p> <p>PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.</p>
Description	<p>Display is changed to Standby mode when this method is executed. A selecting template is deselected.</p> <p>All registered user-defined characters are deleted. Used memory is not released even the user-defined characters are deleted. The used memory can be reused after executing defragment.</p> <p>In use via a printer, this method is ignored when Display is not connected to the printer.</p>

registerOptionFont	Register optional font
---------------------------	------------------------

Registers optional fonts in Display.

Syntax	public void registerOptionFont (int <i>startCode</i> , int <i>endCode</i> , int <i>width</i> , int <i>height</i> , String <i>filePath</i>) throws PrinterException	
Parameter	<i>startCode</i>	Character code for registration starting The valid range is 20h to FFh of ASCII character code.
	<i>endCode</i>	Character code for registration finishing The valid range is 20h to FFh of ASCII character code.

<i>width</i>	Character width (pixel: px) The valid range is 1 to 255.
<i>height</i>	Character height (pixel: px) The valid range is 1 to 255.
<i>filePath</i>	File path Specify the file name of the optional font to register. Supported file extension is .bin. See "Register Optional Font" of the Display command in "Technical Reference for Display" for details on optional font data.

Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class " for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.
Description	Display is changed to Standby mode when this method is executed. A selecting template is deselected. When this method is executed with optional font registered status, the registered optional fonts are deleted, and a new memory area is allocated to register optional fonts. Used memory is not released even the registered optional fonts are deleted. The used memory can be reused after executing defragment . In use via a printer, this method is ignored when Display is not connected to the printer.

unregisterOptionFont

Delete optional font

Deletes registered optional fonts.

Syntax	public void unregisterOptionFont () throws PrinterException
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class " for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.
Description	Display is changed to Standby mode when this method is executed. A selecting template is deselected. All registered optional fonts are deleted. Used memory is not released even the optional fonts are deleted. The used memory can be reused after executing defragment . In use via a printer, this method is ignored when Display is not connected to the printer.

Specifies start or finish of macro registration.

Syntax	public void controlMacroRegistration (int <i>macroID</i> , MacroRegistrationFunction <i>control</i>) throws PrinterException	
Parameter	<i>macroID</i>	Macro ID The valid range is -1 to 127. Do not specify the macro IDs of 120 to 126 because they are being used for the system.
	<i>control</i>	Macro registration processing See "4.1.1(3)⑫ Macro registration processing (MacroRegistrationFunction)" for available constants.
Exception	PrinterException PrinterException is thrown when an error occurs while calling this method. See "4.1.5 PrinterException Class " for details of the error. When the data transmission is failed, the communication with the device is ended, and PrinterException may be thrown. See isConnect for verifying the connection state with the device.	
Description	The procedures of the macro registration processing are as follows: (1) Starts macro registration processing. Specify -1 for <i>macroID</i> . Specify MACRO_REGISTRATION_START for <i>control</i> . (2) Executes methods. Starts buffering of transmit data when methods are targeted in macro registration processing. The transmit data of a macro registration processing target method which is executed during the buffering is not sent to the device, buffered in macro data buffer. The maximum transmit data size to be able to buffer is 1024 bytes. When the buffered transmit data exceeds the maximum size, a macro registration processing target method at the point of exceeding is to be error. When the error occurs, data under the registration is discarded and canceled the macro mode. Regarding transmit data which is held, finish the macro registration processing by procedure (3). When a method is out of the macro registration processing target, it is executed immediately without buffering the transmit data.	

Methods for the macro registration processing target are shown below.

- **showTemplate**
- **showSlide**
- **selectTemplate**
- **setTemplateImageData**
- **selectTemplateTextObject**
- **setTemplateTextAlignment**
- **setTemplateTextLeftMargin**
- **setTemplateTextLineSpacing**
- **setTemplateTextBold**
- **setTemplateTextUnderline**
- **setTemplateTextSize**
- **setTemplateTextFont**
- **setTemplateTextRegisteredFont**
- **setTemplateTextRightSpacing**
- **setTemplateTextColor**
- **setTemplateTextData**
- **setTemplateBarcodeData**
- **setTemplateQrCodeData**

(3) Finishes macro registration processing.

Specify a macro ID (0 to 127) to register at *macroID*.

When **MACRO_REGISTRATION_REGIST** is specified at *control*, buffered transmit data is sent to the device. The buffered transmit data is held even after transmitting to the device.

Display is changed to Standby mode when this method is executed.

A selecting template is deselected.

The holding transmit data is discarded by following processes.

- Specify **MACRO_REGISTRATION_CLEAR**.
- Specify **MACRO_REGISTRATION_START**.
- Execute **disconnect**.

The registered macro can be executed at **executeMacro**.

The process to delete the registered macro is as follow.

Specify **MACRO_REGISTRATION_START** at *control* and specify -1 for *macroID* to call this method.

Specify **MACRO_REGISTRATION_START** at *control* and specify the macro ID to delete, and then call this method.

Display is changed to Standby mode when this method is executed.

A selecting template is deselected.

In use via a printer, this method is ignored when Display is not connected to the printer.

Note **Registered data at the shipping may be added or changed without prior notice for quality improvement.**

getDisplayResponse

Get various response from Display

Gets response data from Display.

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

Parameter *id* Display response type constant
See "4.1.1(2)⑥ Display response type" for available constants.

param Command parameter
The buffer type varies depending on the Display response type constant.
See the following table for description of the value to be specified.

buf

Buffer for storing the retrieved response data
Stores the response data specified with *id* in the object specified with *buf*.
The buffer type varies depending on the Display response type constant.
See the following table for the buffer type.

Response Type Constant	
Parameter	Description
DISPLAY_RESPONSE_REQUEST (Execution response request)	
<i>param</i>	Specify 0 to 15 (00h to 0Fh) in int type.
<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 with 64 to 79 (40h to 4Fh).
DISPLAY_RESPONSE_USER_AREA (Send remaining capacity of user area)	
<i>param</i>	Specify 0 in int type.
<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.
DISPLAY_RESPONSE_TEMPLATE_ID_LIST (Send template ID)	
<i>param</i>	Specify 0 in int type.
<i>buf</i>	Specify an ArrayList<Integer> type array. When the response is retrieved successfully, the registered template ID is stored as an int array.
DISPLAY_RESPONSE_IMAGE_ID_LIST (Send image ID)	
<i>param</i>	Specify 0 in int type.
<i>buf</i>	Specify an ArrayList<Integer> type array. When the response is retrieved successfully, the registered image ID is stored as an int array.
DISPLAY_RESPONSE_SLIDE_ID_LIST (Send slide ID)	
<i>param</i>	Specify 0 in int type.
<i>buf</i>	Specify an ArrayList<Integer> type array. When the response is retrieved successfully, the registered slide ID is stored as an int array.
DISPLAY_RESPONSE_TEMPLATE_LABEL (Send template name)	
<i>param</i>	Specify 0 to 127 (00h to 7Fh) in int type.
<i>buf</i>	Specify an ArrayList<String> type array. When the response is retrieved successfully, the template name specified at template registration is stored as a character string.
DISPLAY_RESPONSE_IMAGE_LABEL (Send image name)	
<i>param</i>	Specify 0 to 63 (00h to 3Fh) in int type.
<i>buf</i>	Specify an ArrayList<String> type array. When the response is retrieved successfully, the image name specified at image data registration is stored as a character string.

Response Type Constant	
Parameter	Description
DISPLAY_RESPONSE_SLIDE_LABEL (Send slide name)	
<i>param</i>	Specify 0 to 91 (00h to 5bh) in int type.
<i>buf</i>	Specify an ArrayList<String> type array. When the response is retrieved successfully, the slide name specified at slide data registration is stored as a character string.

Exception **PrinterException**
PrinterException is thrown when an error occurs while calling this method.
See "4.1.5 **PrinterException Class**" for details of the error.
When the data transmission is failed, the communication with the device is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the device.

4.1.2 PrinterEvent Class

PrinterEvent class gets the end event that occurs when **startDiscoveryPrinter** or **startDiscoveryDevice** is completed.

(1) Method List

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

Name	Description
getEventType	Get end event

(2) End event constant

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

Constant Name	Description	Value
EVENT_FINISHED_DISCOVERY	Completion of startDiscoveryPrinter or startDiscoveryDevice .	1

(3) Method Details

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

Gets the end event when **startDiscoveryPrinter** or **startDiscoveryDevice** is completed.

Syntax public int **getEventType()**

Return value See "4.1.2(2) End event constant" for details of the value.

Description The judgement conditions for the end event are as follows.

- **startDiscoveryPrinter** has been completed

- **startDiscoveryDevice** has been completed

Even when the device was not discovered, **EVENT_FINISHED_DISCOVERY** is returned.

4.1.3 PrinterListener Interface

PrinterListener interface is for getting the end event when **startDiscoveryPrinter** or **startDiscoveryDevice** is completed.

(1) Method List

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

Name	Description
finishEvent	End event of device search

(2) Method Details

finishEvent	End event of device search
--------------------	----------------------------

End event that is called when **startDiscoveryPrinter** or **startDiscoveryDevice** is completed.

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 **startDiscoveryDevice**. Determine the type of the end event by **getEventType** in **PrinterEvent** class.

4.1.4 PrinterInfo Class

PrinterInfo class stores the information of the device found by **startDiscoveryPrinter** or **startDiscoveryDevice**.

(1) Method List

Port name can be retrieved. Methods of **PrinterInfo** class are shown in the following table.

Name	Description
getDevicePath	Get device path

(2) Method Details

getDevicePath	Get device path
----------------------	-----------------

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

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
PrinterException	Constructor
getErrorCode	Get error code

(2) Constant List

① Error code

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

Constant Name	Description	Value
ERROR_ACCESS_DENIED	Failed to get the handle. ^{*1}	-1
	An unavailable port was specified.	
	An unsupported method was specified.	
ERROR_SHARING_VIOLATION	An already opened port was specified.	-11
ERROR_PORT_NOT_OPENED	The port is not open.	-12
ERROR_DEVICE_NOT_CONNECTED	There is a problem with Bluetooth connection between the Android device and the printer.	-21
	There is a problem with USB connection between the Android device and the printer.	
ERROR_OFFLINE	Disconnected state or the printer is offline.	-22
ERROR_EXTERNAL_DEVICE_NOT_CONNECTED	Display is not connected.	-23
ERROR_DEVICE_INITIALIZE_FAILED	Failed to change the printer settings. Data sending to the printer is not completed within the send timeout period, or data receiving from the printer is not completed within the receive timeout period.	-31
ERROR_DATA_SIZE_ZERO	0-byte data was specified.	-101
ERROR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.	-102
ERROR_DATA_SIZE_INVALID	Data size is invalid.	-103
ERROR_ENCODE_FAILED	An error occurred in encoding text data. ^{*1}	-111
ERROR_TIMEOUT	Send timeout occurred.	-201
	Receive timeout occurred.	
ERROR_FILE_NOT_FOUND	The specified file is not found.	-301
ERROR_FILE_USED	The specified file is in use by another process.	-302
ERROR_FILE_INVALID	The specified file is invalid.	-303
ERROR_LOW_MEMORY	Memory shortage occurred when loading image data file.	-311

Constant Name	Description	Value
ERROR_OVER_MAX_IMAGE	Either or both of width and height of image data exceeds the number of printable maximum dots.	-312
ERROR_LOGO_NOT_DEFINED	The logo is not registered.	-313
ERROR_LOW_USER_AREA	Remaining user area is insufficient.	-401
ERROR_LOW_EXTERNAL_RAM	Remaining RAM capacity is insufficient.	-402
ERROR_NOT_REGISTERED	The template is not registered. Image data is not registered. Slide data is not registered. The optional font is not registered. The user-defined character is not registered.	-403
ERROR_NOT_UNREGISTERED	The template is not deleted. Image data is not deleted. Slide data is not deleted. The optional font is not deleted. The user-defined character is not deleted.	-404
ERROR_INVALID_NO	The specified value for the logo ID is invalid.	-501
ERROR_INVALID_DATA	The specified data is invalid.	-503
ERROR_INVALID_PARAM	The specified parameter is invalid.	-9999

*1: Abnormal processing might have occurred.

(3) Method Details

PrinterException Constructor

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

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

getErrorCode Get error codes

Gets the error code for thrown exception.

Syntax `public int getErrorCode()`

Return value See "4.1.5(2) Constant List" for details of the error.

Chapter 5

Sample Program

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

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

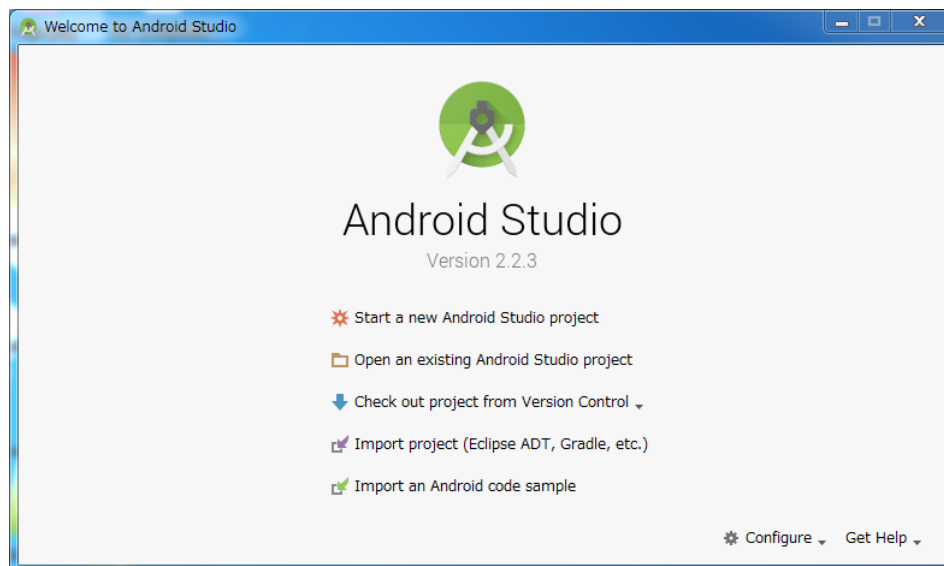
5.1 Installation

Install the sample program.

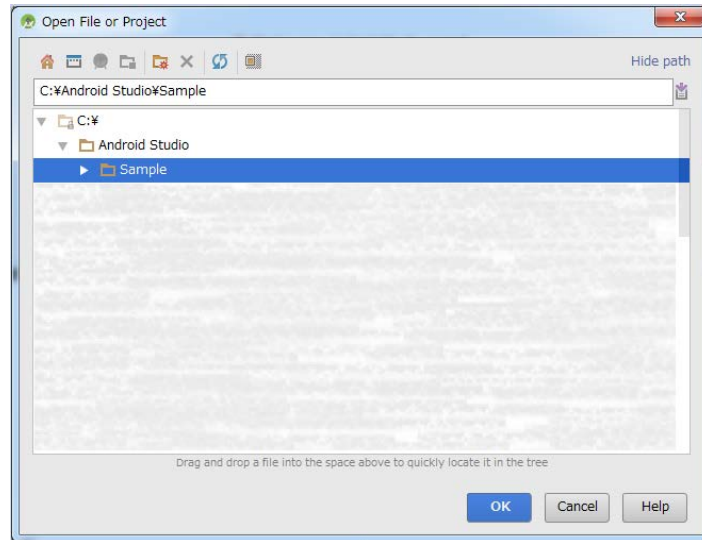
Ensure that the environment for developing Android application is prepared. See "Chapter 3 How to Use the Library" for details about required development environment.

The procedures are shown below.

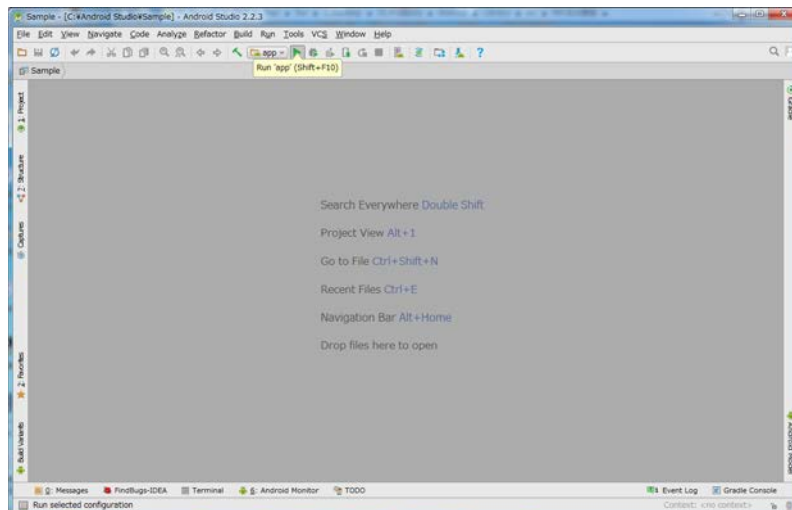
- (1) Place a sample folder at any location.
- (2) Start Android Studio, and click "Open an existing Android Studio project".



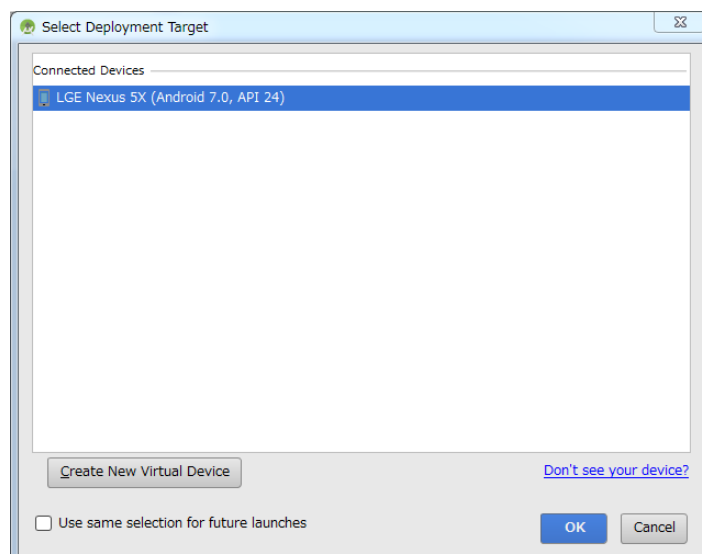
(3) Select the folder placed in the step (1), and click [OK].



(4) Click [Run 'app'].



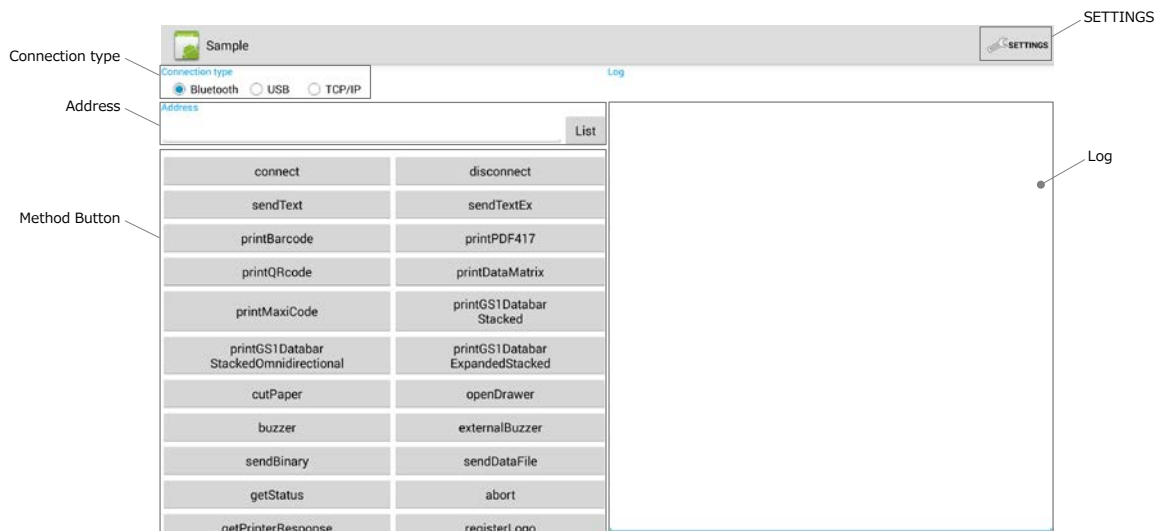
(5) Select the device, and click [OK].




5.2 Screen Layout

This section describes the screen of the sample program.

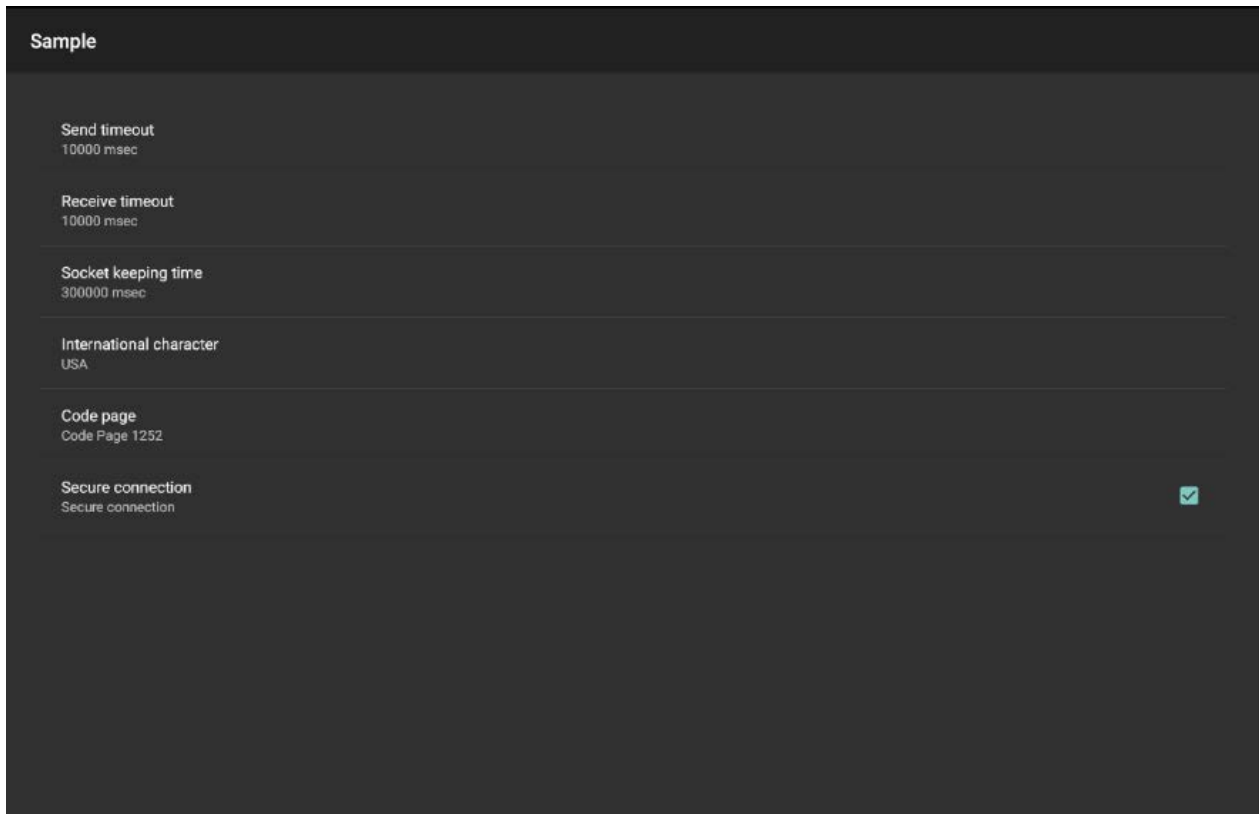
5.2.1 Main screen



Item	Description
Connection type	Selects connection form to a device.
Address	<p>This is valid only when Display is used via a printer. Specifies the printer address. For manual input: When connecting with Bluetooth, enter the Bluetooth address. When connecting with TCP/IP, enter the MAC address.</p> <p>For automatic input: By tapping the [List] button, the information of found printers is displayed in a list. When a printer is selected from the displayed list, the Bluetooth address is automatically entered.</p>
Method Button	<p>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.</p>
SETTINGS	<p>Tapping the [SETTINGS] button opens the function setting screen.</p> <p>In order to go back to the main screen, tap  on the screen.</p>
Log	Executing each method of "Method Button" displays the method execution logs.

5.2.2 [SETTINGS] screen

Various setting functions are displayed in [SETTINGS].



5.3 Precaution

The sample program is subject to change without notice.

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

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. Display 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	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ
D0	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ	ℒ
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	n	2	■	

Figure A-1 CODE_PAGE_437 (USA, Standard Europe)

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

Figure A-2 CODE_PAGE_KATAKANA

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

Figure A-3 CODE_PAGE_850 (Multilingual)

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

Figure A-4 CODE_PAGE_860 (Portuguese)

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

Figure A-8 CODE_PAGE_737 (Greek)

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

Figure A-9 CODE_PAGE_1252 (Latin)

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

Figure A-10 CODE_PAGE_866 (Russian)

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

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	ù	ÿ	Ö	Ü	ø	£	Ø	×	f
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	■	■	■	■	■
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	☐	☐	☐			x	X	и	И							
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	■	■	■	■	■
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	≈	«	»	لَا	لَا	لَا	لَا	لَا	لَا	لَا
A0	-	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل	ل
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	س	س	س	س	س
C0	¢	ء	آ	أ	ؤ	ع	ئ	ب	ة	ث	ج	ح	خ	د	ذ	ذ
D0	ذ	ز	ر	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	م	م
E0	-	ف	ق	ك	م	ل	م	ل	م	ل	م	ل	م	ل	م	ل
F0	-	ن	ه	ه	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي	ي

Figure A-14 CODE_PAGE_864 (Arabic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	;	;	;	...	†	†	™	š	<	š	š	š	š	š	š
90	;	;	;	...	•	-	-	™	š	>	š	š	š	š	š	š
A0	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘	˘
B0	°	±	±	±	±	±	±	±	±	±	±	±	±	±	±	±
C0	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á	Á
D0	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ	Đ
E0	á	á	á	á	á	á	á	á	á	á	á	á	á	á	á	á
F0	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ	đ

Figure A-15 CODE_PAGE_1250 (Central European)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	ѓ	;	ѓ	„	...	†	‡	€	‰	Љ	<	Њ	ќ	ћ	џ
90	ђ	‘	;	“	”	•	-	-	™	Љ	>	њ	ќ	ћ	џ	
A0	ÿ	ÿ	Ј	Ѡ	Г	І	Š	Ě	©	©	«	¬	-	®	İ	
B0	°	±	І	і	г	μ	¶	•	ё	№	е	»	ј	š	s	ï
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	ˆ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ı	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	€	‘	‚	ƒ	„	…	†	‡	^	‰	Š	‹	Œ			
90		‚	“	”	•	-	-	~	™	š	›	œ			ÿ	
A0	ı	ϕ	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

Figure A-18 CODE_PAGE_1254 (Turkish)

A.2 International Character Set

Display 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 display 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	£	\$	@	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

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.)