SII Print Class Library for iOS
Application Programmer's Guide

Rev.01

[Products]

MP-A40 Series

Seiko Instruments Inc.

Rev.01 May 2026

Copyright © 2026 Seiko Instruments Inc. All rights reserved.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.
iPad®, iPad Air®, iPad mini™, iPhone®, and iPod® are trademarks of Apple Inc., registered in the U.S. and other countries and regions.
App Store℠ is a service mark of Apple Inc.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.

All other trademarks are the properties of their respective companies.

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 iOS" (hereinafter referred to as "SII print class library") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Printers

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

Printer	Interface
MP-A40 Series	Bluetooth

Terms

The terms used in this manual are described below.

Term	Description
Printer command	Command for controlling the printer described in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

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 Development of Application that Performs Bluetooth Communication with SII Printer 1-2

Chapter 2 Product Specifications 2-1

2.1 Operating Environment 2-1
2.1.1 Applicable iOS Devices 2-1
2.1.2 Applicable iOS Versions 2-2
2.2 Printer Settings 2-2
2.3 Precaution 2-3

Chapter 3 How to Use library 3-1

3.1 iOS Application Development Environment 3-1
3.2 Provided Files 3-2
3.3 Build the Library to Xcode Project 3-3
3.3.1 Objective-C 3-3
3.3.2 Swift 3-6

Chapter 4 Functions of Library 4-1

4.1 Standard Mode and Page Mode 4-1
4.1.1 Basic Operation 4-1
(1) Standard mode 4-1
(2) Page mode 4-2
4.1.2 Text Data Printing in Standard Mode 4-3
4.1.3 Mapping Position of Print Data in Page Mode 4-4
(1) Print area of page mode 4-4
(2) Print direction 4-4
(3) Reference point 4-5
4.1.4. Print Data Process at Out of Print Area of Page Mode 4-6
4.2 Log File Output Function 4-7
4.2.1 How to Set Log Output 4-7
4.2.2 Log Output Settings 4-7
4 2.3 Log File 4-7
4.3 API Reference 4-8
4.3.1 SIIPrinterManager Class 4-9
(1) Method List 4-9

① Common method to standard mode and page mode 4-9
② Dedicated method for standard mode 4-10
③ Dedicated method for page mode 4-10
(2) Common property list to standard mode and page mode 4-11
(3) Constant List 4-12
① Printer model 4-12
② Port type 4-12
③ Response type 4-12
④ Battery remaining capacity level 4-13
⑤ International character set 4-13
⑥ Codepage 4-14
⑦ Barcode and PDF417 4-14
(4) Enumerated Constant List 4-15
① Dithering (Dithering) 4-15
② Batch processing selection (TransactionFunction) 4-15
③ Bold print (CharacterBold) 4-15
④ Underline (CharacterUnderline) 4-15
⑤ Reverse print (CharacterReverse) 4-16
⑥ Inversion print (CharacterInversion) 4-16
⑦ Character font (CharacterFont) 4-16
⑧ Character scale (CharacterScale) 4-17
⑨ Alignment (PrintAlignment) 4-18
⑩ Barcode symbol (BarcodeSymbol) 4-18
⑪ Module size (ModuleSize) 4-19
⑫ HRI character print position (HriPosition) 4-21
⑬ N:W ratio (NwRatio) 4-21
⑭ Error correction level (ErrorCorrection) 4-21
⑮ PDF417 symbol (Pdf417Symbol) 4-22
⑯ QR Code Model (QrModel) 4-22
⑰ Data Matrix module (DataMatrixModule) 4-22
⑱ MaxiCode Mode (MaxiCodeMode) 4-23
⑲ Cutting method (CuttingMethod) 4-23
⑳ Form feed position (FeedPosition) 4-24
㉑ Image rotation direction (Rotate) 4-24
㉒ Image scaling (ImageScale) 4-25
㉓ Print direction (Direction) 4-25
㉔ Line style (LineStyle) 4-25
(5) Method Details 4-26
① Common method to standard mode and page mode 4-26
init Instance 4-26
connect Start communicating with printer 4-26
disconnect Stop communicating with printer 4-27
openDrawer Open cash drawer 4-27
buzzer Sound buzzer 4-27
externalBuzzer Sound external buzzer 4-28
getStatus Get printer status 4-28
abort Abort waiting state of printer 4-29
registerLogo Register logo 4-29

unregisterLogo Delete registered logo 4-30
registerStyleSheet
Register style sheet 4-30
unregisterStyleSheet
Delete registered style sheet 4-30
resetPrinter Reset printer 4-31
getPrinterResponse
Get various responses from printer 4-31
startDiscoveryPrinter
Start printer search (Bluetooth) 4-33
startDiscoveryPrinter
Start printer search (TCP/IP) 4-34
cancelDiscoveryPrinter
Cancel printer search 4-34
getFoundPrinter Get found printer information 4-34
getVersion Get SDK version 4-34
controlTransaction
Start/End batch processing 4-34
② Dedicated method for standard mode 4-37
sendText Send text data 4-37
sendTextEx Send format specified text data 4-37
printBarcode Print barcode 4-38
printPDF417 Print PDF417 4-42
printQRcode Print QR Code 4-43
printDataMatrix Print Data Matrix 4-44
printMaxiCode Print MaxiCode 4-45
printGS1DataBarStacked
Print GS1 Databar Stacked 4-45
printGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional 4-46
printGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked 4-46
printAztecCode Print Aztec Code 4-47
cutPaper Cut paper 4-47
feedPosition Paper form feed 4-47
sendBinary Send binary data 4-48
sendDataFile Send specified file 4-48
sendImage Send image 4-49
printPDF Print PDF page 4-50
printLogo Print logo 4-51
printSmartLabelImageData
Print label 4-51
③ Dedicated method for page mode 4-52
enterPageMode Start page mode 4-53
exitPageMode End page mode 4-53
setPageModeArea
Specify print area of page mode 4-53

setPageModeDirection
Specify print direction of page mode 4-55
setPageModeLineSpacing
Specify line spacing of page mode..... 4-55
printPageMode Print page mode 4-55
printPageModeText
Send text data of page mode 4-56
printPageModeTextEx
Send format specified text data of page mode..... 4-56
printPageModeBarcode
Print barcode of page mode..... 4-57
printPageModePDF417
Print PDF417 of page mode..... 4-61
printPageModeQRcode
Print QR Code of page mode..... 4-62
printPageModeDataMatrix
Print Data Matrix of page mode..... 4-63
printPageModeMaxiCode
Print MaxiCode of page mode..... 4-63
printPageModeGS1DataBarStacked
Print GS1 Databar Stacked of page mode..... 4-64
printPageModeGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional of page mode 4-65
printPageModeGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked of page mode 4-65
printPageModeAztecCode
Print Aztec Code of page mode 4-66
sendPageModeBinary
Send binary data of page mode 4-66
printPageModeImageFile
Draw Image file of page mode 4-67
printPageModeRectangle
Draw rectangle image of page mode..... 4-67
printPageModeLine
Print ruled line of page mode 4-68
printPageModeLogo
Print logo of page mode..... 4-70
(6) Common property detail to standard mode and page mode..... 4-71
sendTimeout Get/Set send timeout period 4-71
receiveTimeout Get/Set receive timeout period..... 4-71
internationalCharacter
Get/Set international character set..... 4-71
codePage Get/Set codepage..... 4-72
printerModel Get printer model..... 4-72
portType Get connecting port type..... 4-72
isConnect Verify connection state with printer..... 4-72

socketKeepingTime
Get/Set socket keeping time 4-73
delegate Register delegate 4-73
PrintSmartLabelMode
Get/Set paper when printing label file 4-73
4.3.2 SIIPrinterInfo Class 4-74
4.3.3 SIIPrinterIdentifier Class 4-75
(1) Method List 4-75
(2) Property List 4-75
(3) Method Details 4-75
SIIPrinterIdentifier
Constructor 4-75
(4) Property Details 4-76
printerModelID Get printer model ID 4-76
typeID Get type ID 4-76
romVersionID Get ROM version ID 4-76
mainFirmwareVersion
Get firmware version (main) 4-76
manufacturer Get manufacturer 4-76
modelName Get model name 4-76
serialNumber Get serial number 4-76
bootFirmwareVersion
Get firmware version (boot) 4-77
bootFirmwareChecksum
Get firmware checksum (boot) 4-77
mainFirmwareChecksum
Get firmware checksum (main) 4-77
firmwareChecksum
Get firmware checksum (main + boot) 4-77
4.3.4 SIIPrinterException Class 4-78
(1) Method List 4-78
(2) Property List 4-78
(3) Constant List 4-79
① Error code 4-79
(4) Method Details 4-80
SIIPrinterException
Constructor 4-80
(5) Property Details 4-80
errorCode Get error code 4-80
errorMessage Get error message 4-80
4.3.5 SIIPrinterManagerDelegate Protocol 4-81
(1) Method List 4-81
(2) Method Details 4-81
didStatusChange Notify printer status 4-81
4.3.6 SIISmartLabelManager Class 4-82

Chapter 5 Sample Program 5-1

5.1 Screen Layout 5-1
5.1.1 Main Screen 5-1
5.1.2 [Settings] Screen 5-3
5.2 Precaution 5-3

Appendix A Character Set A-1

A-1 Codepage Table (Character Code Table) A-1
A-2 International Character Set A-11

Appendix B Barcode Size List B-1

B.1 Barcode Size List B-1
B.1.1 printBarcode, printPageModeBarcode B-1
B.1.2 printPDF417, printPageModePDF417 ..... B-7
B.1.3 printQRCode, printPageModeQRCode B-8
B.1.4 printDataMatrix, printPageModeDataMatrix B-9
B.1.5 printMaxicode, printPageModeMaxicode B-11
B.1.6 printGS1DataBarStacked, printPageModeGS1DataBarStacked B-12
B.1.7 printGS1DataBarStackedOmnidirectional, printPageModeGS1DataBarStackedOmnidirectional B-13
B.1.8 printGS1DataBarExpandedStacked, printPageModeGS1DataBarExpandedStacked B-14

Appendix C Open Source Software License C-1

C.1 MIT License C-1
C.2 Apache License 2.0 ..... C-2

Chapter 1
Product Overview

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

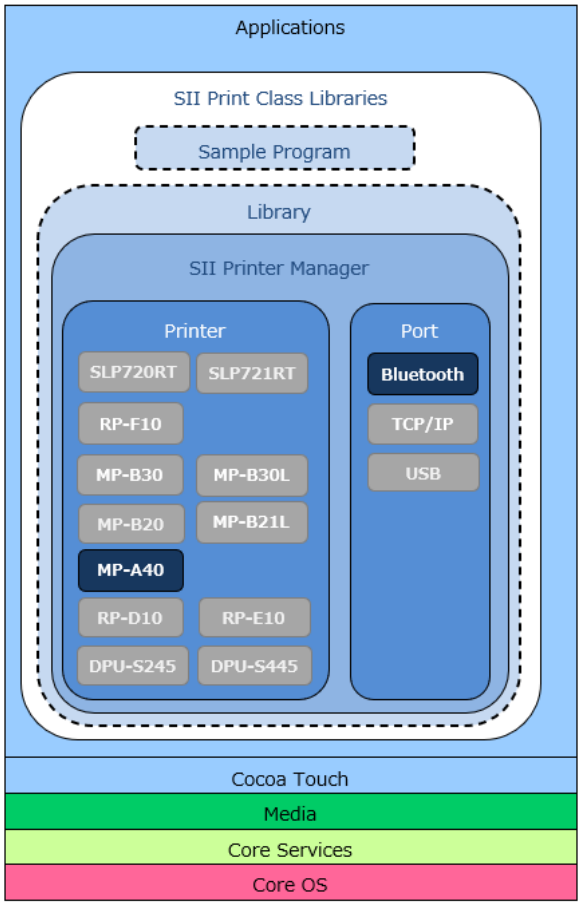
1.1 Functions Provided by SII Print Class Library

The SII print class library including the library and the sample program provides the functions to use SII printer MP-A40 Series (hereinafter referred to as "printer") in iOS applications.
Moreover, the SII print class library provides the library sample program in Xcode project.

1.2 SII Print Class Library Overview

1.2.1 SII Print Class Library Configuration

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



1.2.2 Functions Provided by Library

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

The library provides the following functions:

- Connecting to / disconnecting from a printer
- Sending data to a printer (print data and/or printer commands[*1])
- Printing barcode and 2-dimensional barcode
- Sending a data file to a printer (print data and/or printer commands[*1])
- Getting the printer status
- Aborting the waiting state of a printer
- Getting various responses from a printer
- Bulk registration of print commands
- Registering a printer status call back function
- Outputting a log file

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

(NOTE) MP-A40 does not support the APIs of Display or the barcode scanner.

1.2.3 Development of Application that Performs Bluetooth Communication with SII Printer

When registering an application that performs Bluetooth communication with a printer to App Store, advance application from SII to Apple is necessary. For details, please contact SII.

Chapter 2
Product Specifications

This chapter describes the product specifications of the library.

2.1 Operating Environment

2.1.1 Applicable iOS Devices

Applicable iOS devices for the library are shown in the following list.

(1) MP-A40-B06JK3x

iPhone models

- iPhone 16 Pro Max
- iPhone 16 Pro
- iPhone 16 Plus
- iPhone 16
- iPhone 15 Pro Max
- iPhone 15 Pro
- iPhone 15 Plus
- iPhone 15
- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone SE (2nd generation)

iPad models

- iPad mini (A17 Pro)
- iPad Pro 13-inch (M4)
- iPad Pro 11-inch (M4)
- iPad Air 13-inch (M2)
- iPad Air 11-inch (M2)
- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11-inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)

2.1.2 Applicable iOS Versions

Applicable iOS versions for the library are shown in the following list.

- iOS 17 to 17.6.1
- iPadOS 17 to 17.6.1
- iOS 18 to 18.6
- iPadOS 18 to 18.6
- iOS 26 to 26.4
- iPadOS 26 to 26.4

2.2 Printer Settings

Set the memory switches and the Bluetooth communication setting of the printer to [Value] in the following table when using the library.
The printer memory switches and Bluetooth communication settings can be set in the iOS app "SII Printer Utility" on the App Store.

See "MP-A40 SERIES Thermal Printer USER'S GUIDE" for details about the memory switches and the factory default settings.

Memory Switch

MS	Function	Value
1-1	Interface Selection (Interface)	1: USB/Wireless
1-2	Mark Mode Selection (Mark Mode)	1: Disable*1 0: Enable*2
1-3 to 1-5	Command System Selection (Command System)	000B: ESC/POS
1-6	Data Discard Selection When Error Occurs (Error Through)	0: Enable
2-2	Realtime Command Selection (Realtime Command)	0: Enable
9-1	Automatic Status Response Selection<ESC/POS> (Auto Status Back<ESC/POS>)	0: Enable
9-2	Initialized Response Selection<ESC/POS> (Init. Response<ESC/POS>)	0: Enable

*1: Select "Disable" when using **`cutPaper`**.
*2: Select "Enable" when using **`feedPosition`**.

See the printer command "Set Bluetooth Communication" described in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCES" for the Bluetooth communication setting.

Bluetooth Communication Setting

Function	Value
iOS Auto Connection	Enable*1 Disable

*1: Select "Enable" when using **`resetPrinter`**.

2.3 Precaution

This library is not thread safe. When this library is used on multiple threads, abnormal termination may occur.

A concurrent connection from multiple apps to one printer is not supported when multiple apps are worked simultaneously by Multitasking on iPad with iPadOS.

Chapter 3
How to Use Library

This chapter describes the development environment of iOS application and how to use the library.

3.1 iOS Application Development Environment

In order to develop iOS applications, the following tools are required.

- Xcode 15.0 or later

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.

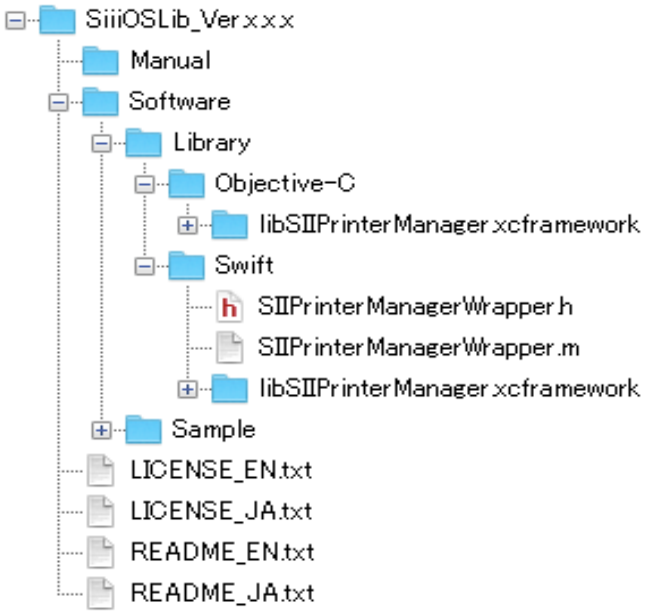


Figure 3-1

The file format of the library is XCFramework. The file name of the library is libSIIPrinterManager.xcframework.

3.3 Build the Library into Xcode Project

Using the project of the sample program (SIILibSample) included in the SII print class library as an example, this section describes by development language how to build the library into the project.

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

Development Language	Description
Objective-C	See "3.3.1 Objective-C" for details to build the library as Objective-C.
Swift	See "3.3.2 Swift" for details to build the library as Swift.

(NOTE) If the following libraries provided SII Print Class Library for iOS Ver. 3.8.0 or earlier versions are included in the target project, delete them all.

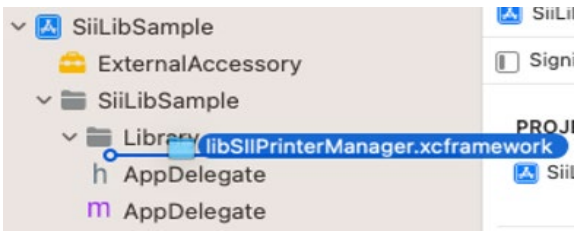
- **libSIIPrinterManager.a**
- **SIIPrinterEnum.h**
- **SIIPrinterException.h**
- **SIIPrinterManager.h**
- **SIISmartLabelManager.h**

3.3.1 Objective-C

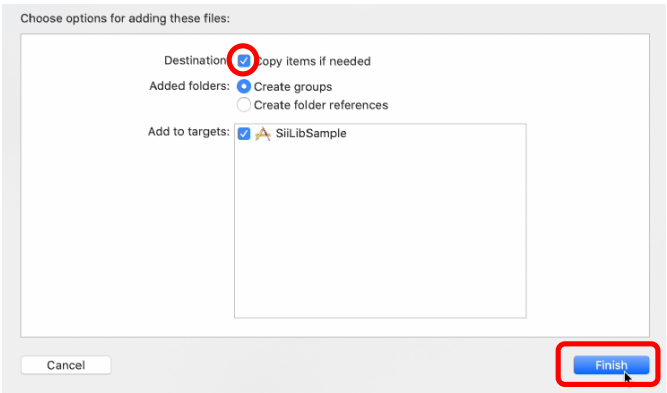
(1) Open the Xcode project.

(2) Drag the following files to any hierarchy in the target project in [Project Navigator] of the navigator window.

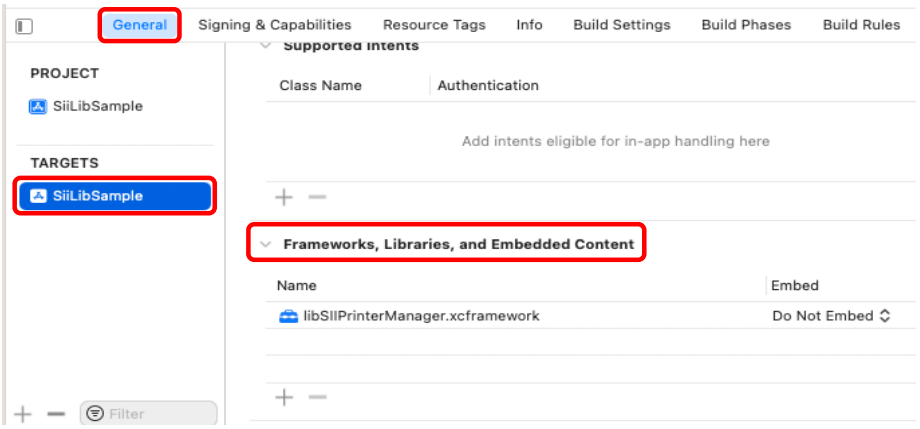
- libSIIPrinterManager.xcframework



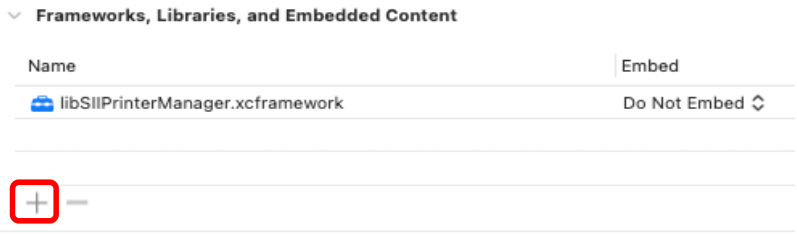
(3) Check the box [Copy items if needed], and click the [Finish] button.



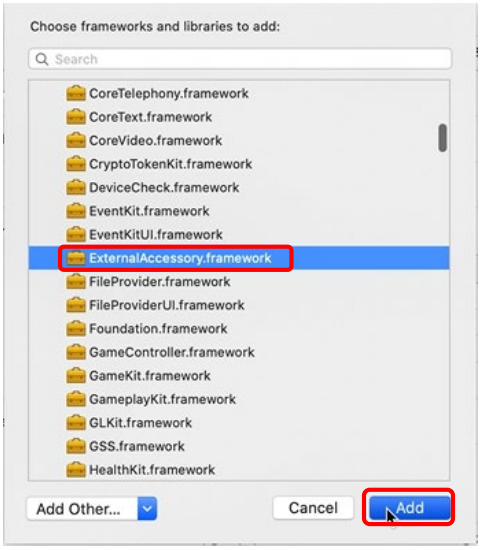
(4) Build the ExternalAccessory.framework into the project.
Select the target project in the [TARGETS], and open the [General] - [Frameworks, Libraries and Embedded Content].

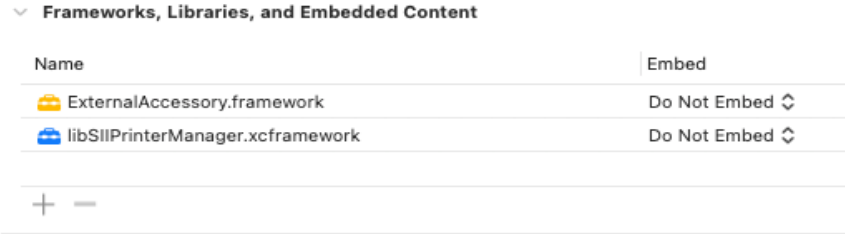


(5) Click the [+] button opened the [Frameworks, Libraries and Embedded Content].

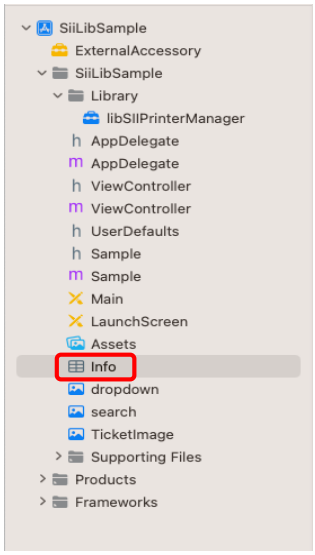


(6) Select the ExternalAccessory.framework from the list and click the [Add] button.

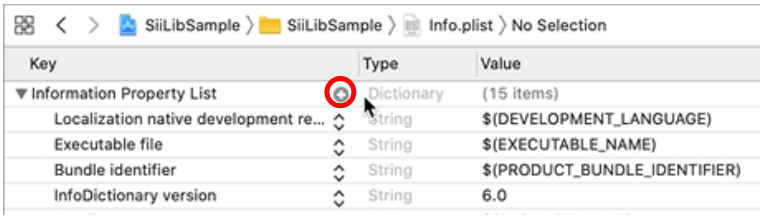




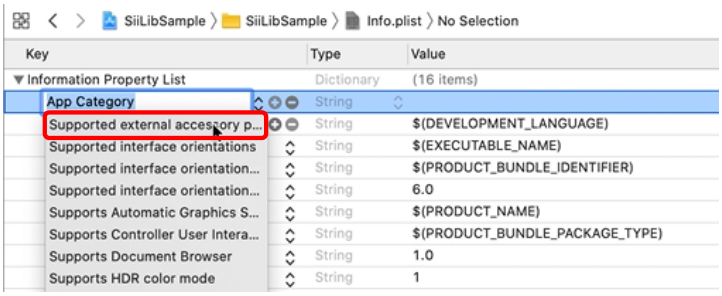
(7) Set the protocol name to use in the ExternalAccessory.framework. Select property list (.plist) in the [Project Navigator].



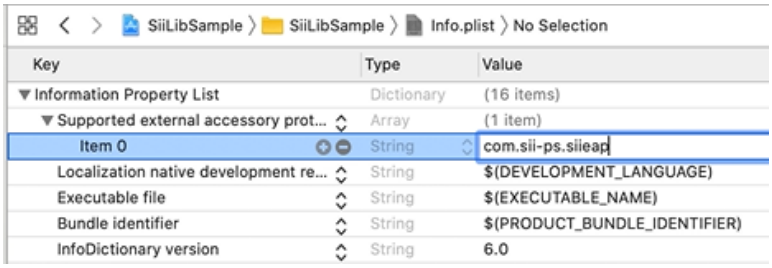
(8) Select the [Information Property List] - ⊕.



(9) Select the [Supported external accessory protocols] from the list.



(10) Open the added [Supported external accessory protocols].
The [Item 0] displayed in the opened [Supported external accessory protocols], enter com.sii-ps.siieap as the Value.



Use the following import statement when importing libraries.
#import <SIIPrinterManager/SIIPrinterManager.h>

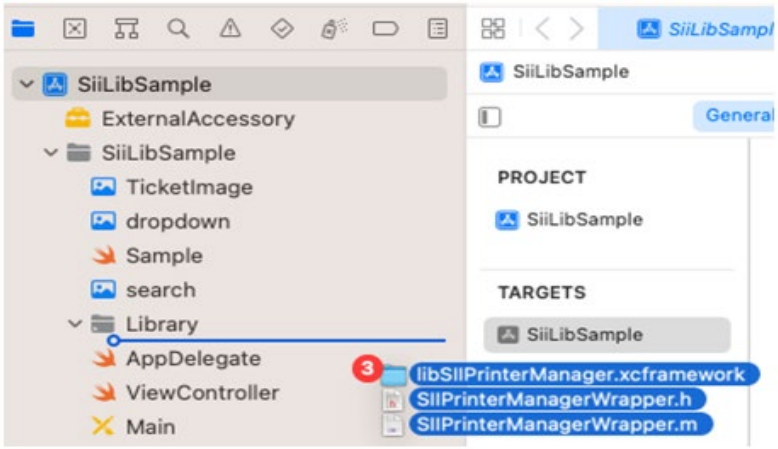
By completing these procedures, the library function becomes available.

3.3.2 Swift

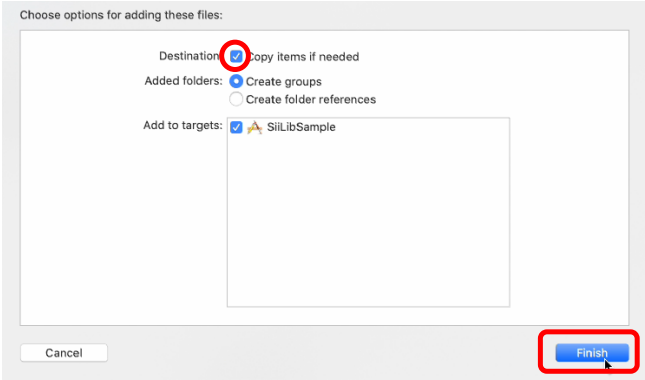
(1) Open the Xcode project.

(2) Drag the following files to any hierarchy in the target project in [Project Navigator] of the navigator window.

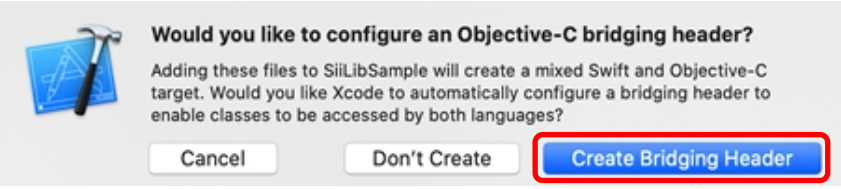
- libSIIPrinterManager.xcframework
- SIIPrinterManagerWrapper.h
- SIIPrinterManagerWrapper.m



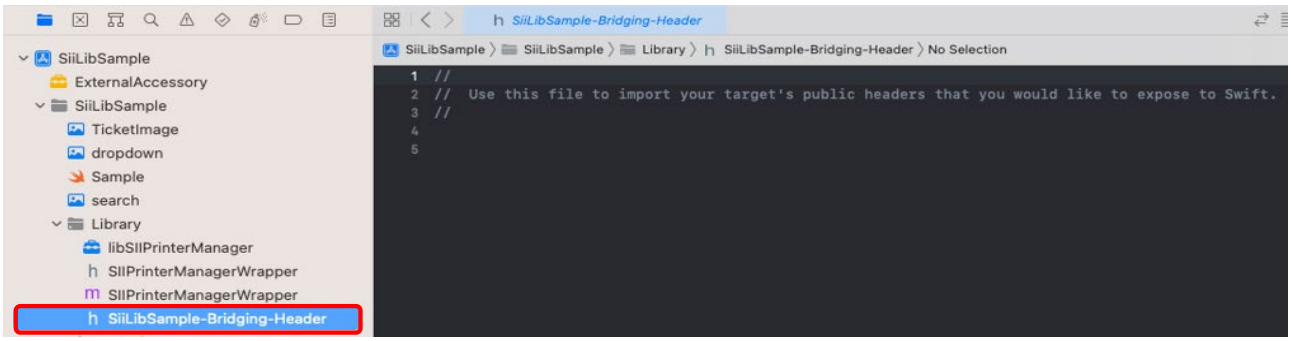
(3) Check the box [Copy items if needed], click the [Finish] button.



(4) The dialog is displayed. Select the [Create Bridging Header] button and create xxxxxxxx-Bridging-Header.h.



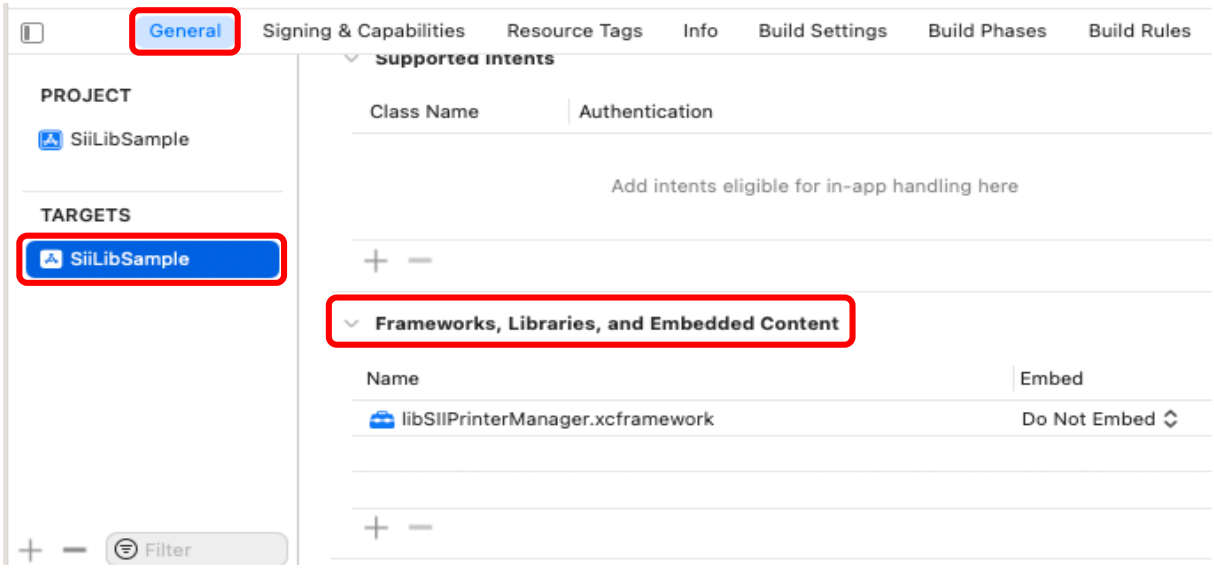
(5) Select the created xxxxxxxx-Bridging-Header.h.



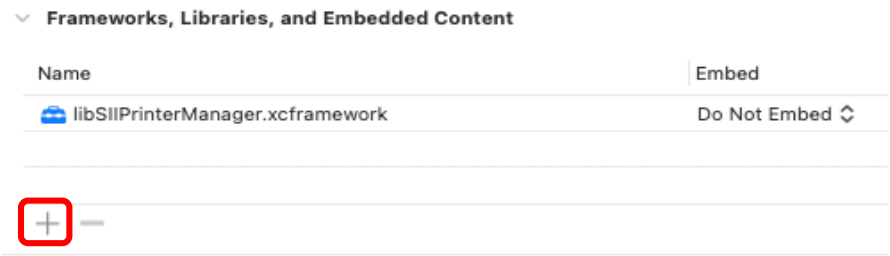
(6) Import the SIIPrinterManager.h and the SIIPrinterManagerWrapper.h into the xxxxxxxx-Bridging-Header.h.

```
1 //
2 //  Use this file to import your target's public headers that you would like to expose to Swift.
3 //
4
5 #import <SIIPrinterManager/SIIPrinterManager.h>
6 #import "SIIPrinterManagerWrapper.h"
7
```

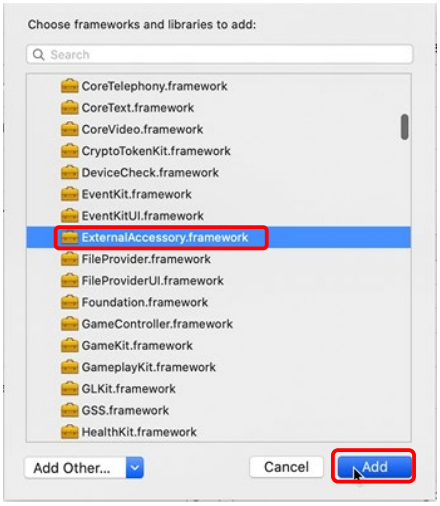
(7) Build the ExternalAccessory.framework.
Select the target project in the [TARGETS], and open the [General] - [Frameworks, Libraries and Embedded Content].

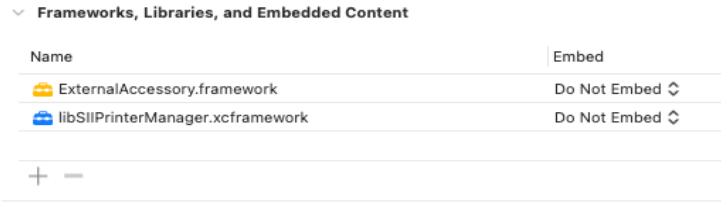


(8) Click the [+] button opened the [Frameworks, Libraries and Embedded Content].

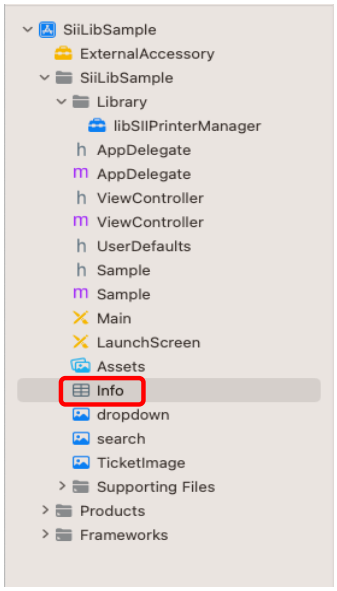


(9) Select the ExternalAccessory.framework from the list and click the [Add] button.

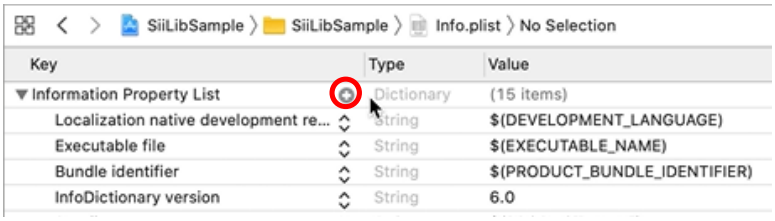




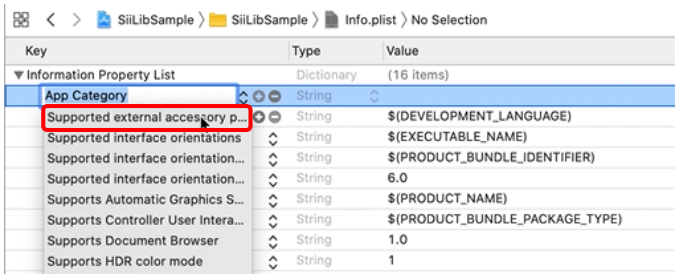
(10) Set the protocol name to use in the ExternalAccessory.framework. Select property list (.plist) in the [Project Navigator].



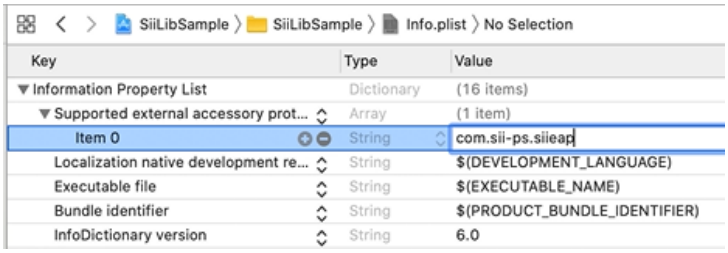
(11) Select the [Information Property List] - ⊕.



(12) Select the [Supported external accessory protocols] from the list.



(13) Open the added [Supported external accessory protocols].
The [Item 0] displayed in the opened [Supported external accessory protocols], enter com.sii-ps.siieap as the Value.



By completing these procedures, the library function becomes available.

Chapter 4
Functions of Library

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

4.1 Standard Mode and Page Mode

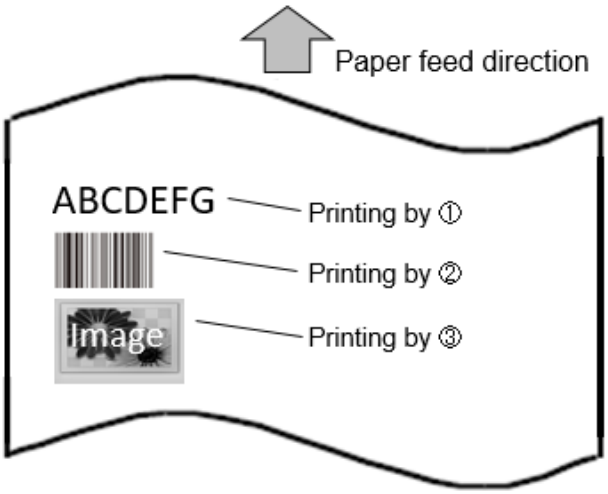
4.1.1 Basic Operation

There are two printing modes "Standard mode" and "Page mode" in the library.
The "Standard mode" and "Page mode" are described below.

(1) Standard mode

Standard mode is the mode to perform the printing in sequence.

Sample print command
① Send text data
② Print barcode
③ Send specified file (Specify an image file)



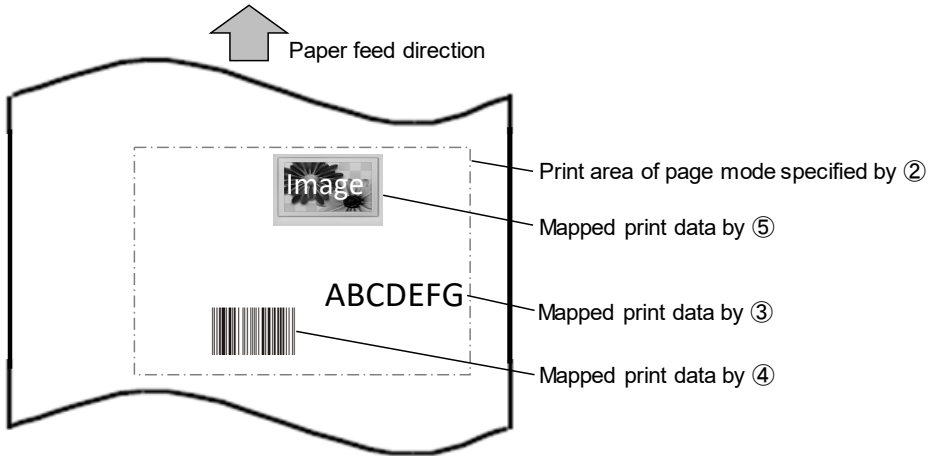
Standard mode suits the printing with an unfixed length such as a receipt.

(2) Page mode

Page mode is the mode to perform the printing on a per-page basis.

In page mode, the print area of page mode is allocated at first, and then print data is mapped on an arbitrary position of the print area.
The mapped print data is printed by the print method of page mode.

Sample print command
① Start page mode
② Specify print area of page mode
③ Send text data of page mode
④ Print barcode of page mode
⑤ Draw image file of page mode
⑥ Print page mode (print the data of ③④⑤ on the print area of ②)
⑦ End page mode



Page mode suits the printing for the followings.
・The printing with a fixed length.
・The printing with the coordinate determination of the character starting position or the ruled line printing position.

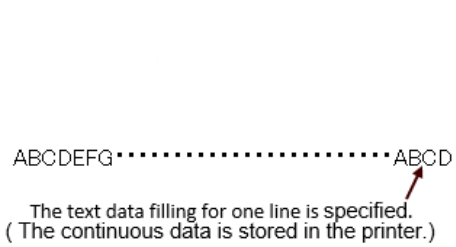
4.1.2 Text Data Printing in Standard Mode

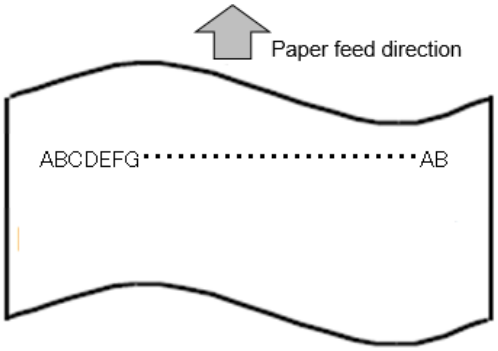
The text data in standard mode is printed each one line.

The text data is stored in the printer when the text data less than one line is specified.
The stored text data is printed by either the following conditions.

・The text data filling for one line is specified.
・Line Feed code is specified.

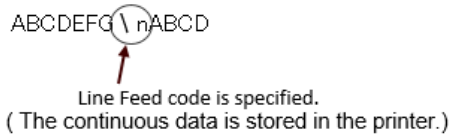
- The print process when the text data filling for one line is specified.

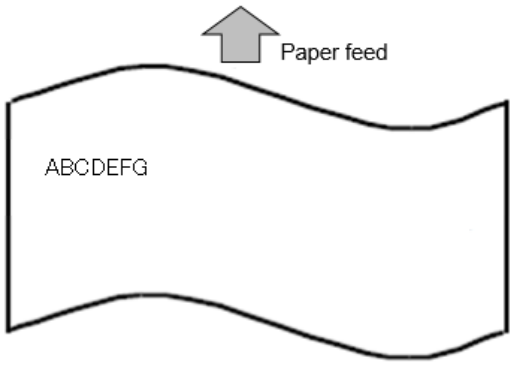




Input data **Print result**

- The print process when Line Feed code is specified.





Input data **Print result**

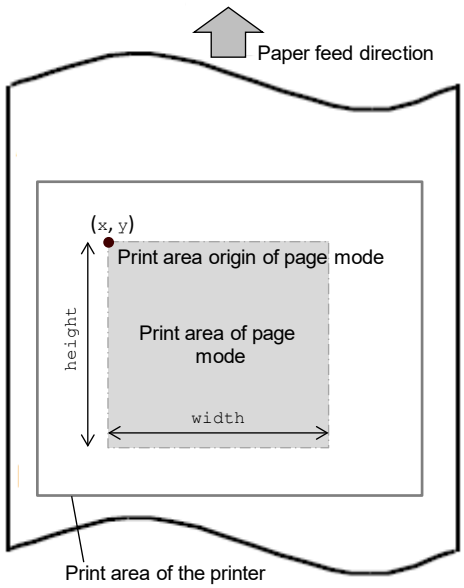
4.1.3 Mapping Position of Print Data in Page Mode

In page mode, the mapping position of print data is determined by print area, print direction, and reference point.
This section describes the print area, print direction, and reference point.

(1) Print area of page mode

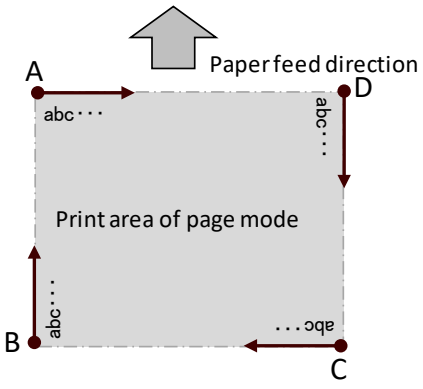
The print area of page mode is specified against the print area of the printer by the print area origin, and the width and the height of page mode. The view of the print area is shown in the following figures.

The print area of page mode can be specified more than one.



(2) Print direction

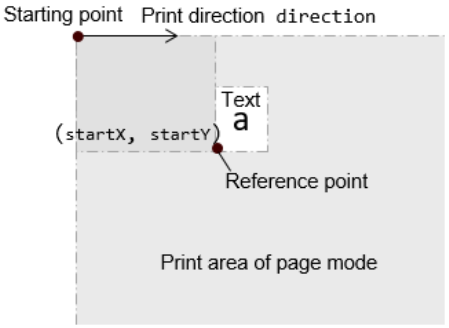
Specify the print direction at setting the print area of page mode.
The starting point is changed depending on specifying the print direction for each direction.
The relation between the print direction and the starting point is shown in the figure below.

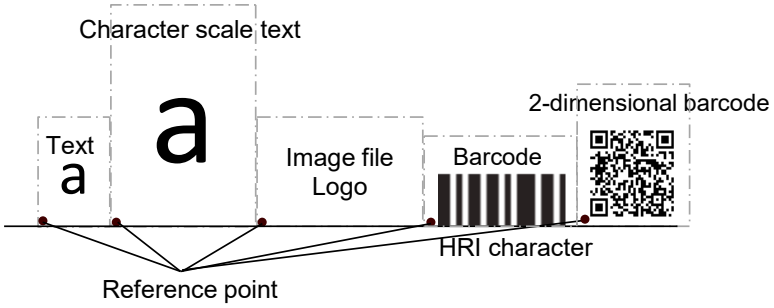


・Starting point: Upper left (A on the figure), Print direction: Left to Right
・Starting point: Left below (B on the figure), Print direction: Below to Upper
・Starting point: Right below (C on the figure), Print direction: Right to Left
・Starting point: Upper right (D on the figure), Print direction: Upper to Below

(3) Reference point

The relation between the reference point for mapping data and each print element (text, image file, logo, and barcode, etc.) is shown in the figures below.





(NOTE) The reference point cannot be specified out of the print area of page mode.

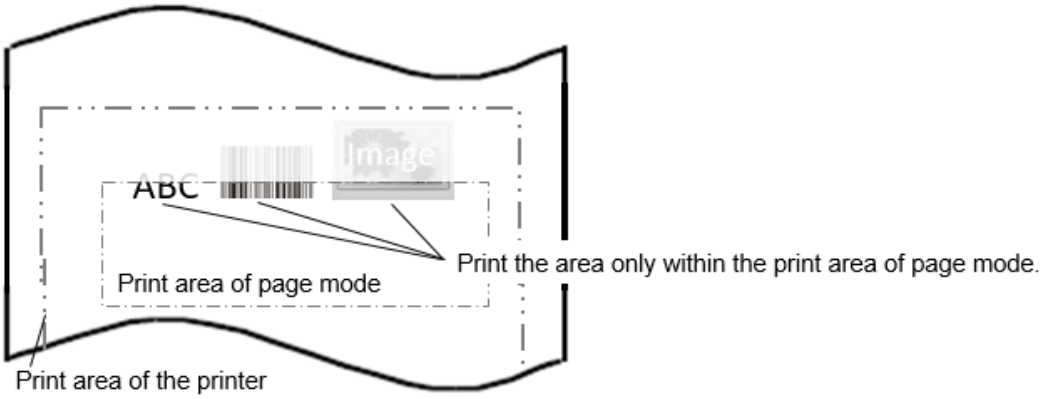
4.1.4 Print Data Process at Out of Print Area of Page Mode

This section describes the process when mapped data is to be mapped on out of the print area of page mode.

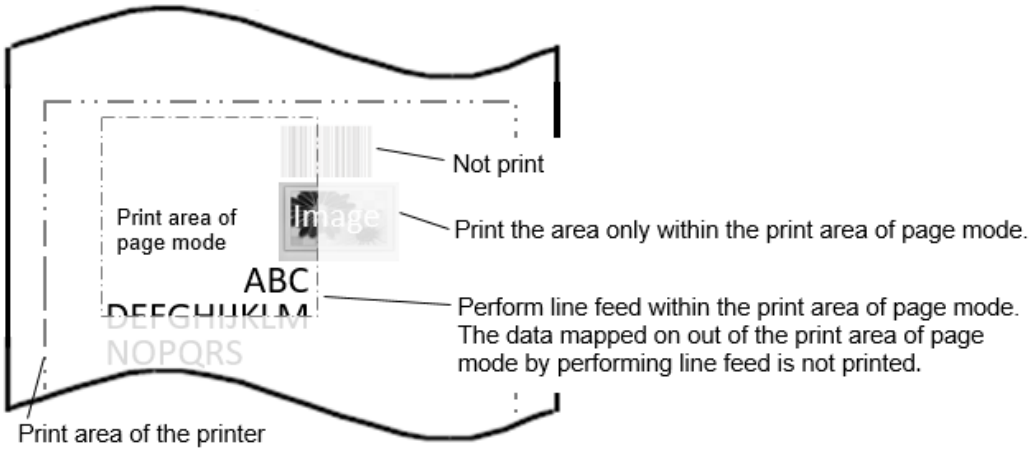
Type of Print Data

Text	Barcode, 2-dimensional Barcode	Image File, Logo, Rectangle, Ruled Line
ABC		Image

(1) The print data is mapped on the upper of the print area of page mode.



(2) The print data is mapped on the right of print area of page mode.



(NOTE)	**Read error or incorrect reading may occur when the part of mapped barcode data is on out of the print area of page mode.**

4.2 Log File Output Function

The logs can be retrieved and the log files can be output using the library.

4.2.1 How to Set Log Output

Log output settings can be configured by adding the config.ini file with the following content to the specific directory of the ios application that incorporates the library.

config.ini

```
LOGLEVEL=x
LOGSIZEMAX=xMB
LOGOUTPUT=x
```

Reference: See "4.2.2 Log Output Settings" fot details on the settings for x.

4.2.2 Log Output Settings

Item	Description	Settings
LOGLEVEL	Log level	0 : Not record the log. 1 : Records an error log when PrinterException occurs. 2 : Records API execution history.
LOGSIZEMAX	Log file maximum size	1MB : Log file maximum size is 1 MB 5MB : Log file maximum size is 5 MB 10MB : Log file maximum size is 10 MB 50MB : Log file maximum size is 50 MB
LOGOUTPUT	Console output enabled/disabled	0 : Console output is disabled 1 : Console output is enabled

4.2.3 Log File

Log files are saved as local files in the Android application that incorporates the library.

Log file name : PrinterManagerX.log (range of X is 0 to 4)

The 1st log file is created as PrinterManager0.log. If the log file maximum size is exceeded, changes the file name to PrinterManager1.log and creates a new PrinterManager0.log.

Up to 5 log files can be created.

4.3 API Reference

This library includes the following classes and protocol.

Name	Description	Supported*1
SIIPrinterManager	Provides the APIs used for communication with the printer and for printing. See "**4.3.1 SIIPrinterManager Class**" for details.	✓
SIIPrinterInfo	Stores the printer information found by **startDiscoveryPrinter**.	-
SIIPrinterIdentifier	Stores the printer ID of the printer retrieved by **getPrinterResponse**. See "**4.3.3 SIIPrinterIdentifier Class**" for details.	✓
SIIPrinterException	Exception class that is thrown at API call. See "**4.3.4 SIIPrinterException Class**" for details.	✓
SIIPrinterManagerDelegate	Provides the API to get notice from the printer. See "**4.3.5 SIIPrinterManagerDelegate Protocol**" for details.	✓
SIISmartLabelManager	Provides the API to specify label files or replace data.	-

*1: ✓: Supported, - : Not supported

(NOTE) MP-A40 does not support the APIs of Display or the barcode scanner.

4.3.1 `SIIPrinterManager` Class

(1) Method List

Methods provided by the **`SIIPrinterManager`** class are shown in the following table.
"Standard mode" or "Page mode" can be selected in the **`SIIPrinterManager`** class.

Method	Description
Common method to standard mode and page mode	The valid methods in standard mode and page mode. See "4.3.1(1)① Common method to standard mode and page mode" for the methods.
Dedicated method for standard mode	The valid methods in standard mode. See "4.3.1(1)② Dedicated method for standard mode" for the methods.
Dedicated method for page mode	The valid methods in page mode. See "4.3.1(1)③ Dedicated method for page mode" for the methods.

① Common method to standard mode and page mode

Methods provided by the common method to standard mode and page mode are shown in the following table. See "4.3.1(5)① Common method to standard mode and page mode" for details of the common methods.

Name	Description	Supported *1
`init`	Instance	✓
`connect`	Start communicating with printer	✓
`disconnect`	Stop communicating with printer	✓
`openDrawer`	Open cash drawer	-
`buzzer`	Sound buzzer	-
`externalBuzzer`	Sound external buzzer	-
`getStatus`	Get printer status	✓
`abort`	Abort waiting state of printer	✓
`registerLogo`	Register logo	✓
`unregisterLogo`	Delete registered logo	✓
`registerStyleSheet`	Register style sheet	✓
`unregisterStyleSheet`	Delete registered style sheet	✓
`resetPrinter`	Reset printer	✓
`getPrinterResponse`	Get various responses from printer	✓
`startDiscoveryPrinter`	Start printer search (Bluetooth)	✓
`startDiscoveryPrinter`	Start printer search (TCP/IP)	-
`cancelDiscoveryPrinter`	Cancel printer search	-
`getFoundPrinter`	Get found printer information	-
`getVersion`	Get SDK version	✓
`controlTransaction`	Start/End batch processing	✓

*1: ✓: Supported, -: Not supported

(NOTE) MP-A40 does not support the APIs of Display.

② Dedicated method for standard mode

Methods provided by the dedicated method for standard mode are shown in the following table.
See "4.3.1(5)② Dedicated method for standard mode" for details of the specified methods.

Name	Description	Supported *1
sendText	Send text data	✓
sendTextEx	Send format specified text data	✓
printBarcode	Print barcode	✓
printPDF417	Print PDF417	✓
printQRcode	Print QR Code	✓
printDataMatrix	Print Data Matrix	✓
printMaxiCode	Print MaxiCode	✓
printGS1DataBarStacked	Print GS1 Databar Stacked	✓
printGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional	✓
printGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked	✓
printAztecCode	Print Aztec Code	−
cutPaper	Cut paper*2	✓
feedPosition	Paper form feed	✓
sendBinary	Send binary data	✓
sendDataFile	Send specified file	✓
sendImage	Send image	✓
printPDF	Print PDF page	✓
printLogo	Print logo	✓
printSmartLabelImageData	Print label	−

*1: ✓: Supported, -: Not supported
*2:: Only the paper feed operation to the paper cut position is performed.

③ Dedicated method for page mode

Methods provided by the dedicated method for page mode are shown in the following table.
See "4.3.1(5)③ Dedicated method for page mode" for details of the specified methods.

Name	Description	Supported *1
enterPageMode	Start page mode	✓
exitPageMode	End page mode	✓
setPageModeArea	Specify print area of page mode	✓
setPageModeDirection	Specify print direction of page mode	✓

Name	Description	Supported *1
setPageModeLineSpacing	Specify line spacing of page mode	✓
printPageMode	Print page mode	✓
printPageModeText	Send text data of page mode	✓
printPageModeTextEx	Send format specified text data of page mode	✓
printPageModeBarcode	Print barcode of page mode	✓
printPageModePDF417	Print PDF417 of page mode	✓
printPageModeQRcode	Print QR Code of page mode	✓
printPageModeDataMatrix	Print Data Matrix of page mode	✓
printPageModeMaxiCode	Print MaxiCode of page mode	✓
printPageModeGS1DataBarStacked	Print GS1 Databar Stacked of page mode	✓
printPageModeGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional of page mode	✓
printPageModeGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked of page mode	✓
printPageModeAztecCode	Print Aztec Code of page mode	–
sendPageModeBinary	Send binary data of page mode	✓
printPageModeImageFile	Draw image file of page mode	✓
printPageModeRectangle	Draw rectangle image of page mode	✓
printPageModeLine	Print ruled line of page mode	✓
printPageModeLogo	Print logo of page mode	✓

*1: ✓: Supported, -: Not supported

(2) Common property list to standard mode and page mode

Properties provided by **SIIPrinterManager** class are shown in the following table.

Name	Access	Description	Supported*1
sendTimeout	R/W	Get/Set send timeout period	✓
receiveTimeout	R/W	Get/Set receive timeout period	✓
internationalCharacter	R/W	Get/Set international character set	✓
codePage	R/W	Get/Set codepage	✓
printerModel	R	Get printer model	✓
portType	R	Get connecting port type	✓
isConnect	R	Verify connection state with printer	✓
socketKeepingTime	R/W	Get/Set socket keeping time	-

Name	Access	Description	Supported*1
delegate	R/W	Register delegate	✓
printSmartLabelMode	R/W	Get/Set paper when printing label file	−

*1: ✓: Supported, - : Not supported

(3) Constant List

① Printer model

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

Constant Name	Description	Value
SII_PM_PRINTER_MODEL_MP_A40	MP-A40	297

② Port type

Constants used for starting communication with the printer and getting the connection port type are shown in the following table.

Constant Name	Description	Value
SII_PM_PRINTER_PORT_TYPE_BLUETOOTH	Bluetooth	0

③ Response type

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

Constant Name	Description	Value
SII_PM_PRINTER_RESPONSE_REQUEST	Execution response request	0
SII_PM_PRINTER_RESPONSE_USER_AREA	Send remaining capacity of user area	1
SII_PM_PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity of user area after defragment	2
SII_PM_PRINTER_RESPONSE_NV_GRAPHICS	Send NV graphics memory capacity	3
SII_PM_PRINTER_RESPONSE_KEY_CODE	Send key code list of defined NV graphics	4
SII_PM_PRINTER_RESPONSE_BATTERY_STATUS	Battery remaining capacity level	5
SII_PM_PRINTER_RESPONSE_FIRMWARE_VERSION	Send firmware version	6
SII_PM_PRINTER_RESPONSE_MAINTENANCE_COUNTER	Send maintenance counter	7
SII_PM_PRINTER_RESPONSE_PRINTER_IDENTIFIER	Send printer ID	8

④ Battery remaining capacity level

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

Constant Name	Description	Value
SII_PM_BATTERY_STATUS_FULL	Battery remaining capacity: approx. 80%	0
SII_PM_BATTERY_STATUS_MIDDLE	Battery remaining capacity: approx. 40%	1
SII_PM_BATTERY_STATUS_LOW	Battery remaining capacity: approx. 10%	2
SII_PM_BATTERY_STATUS_EMPTY	No battery	3
SII_PM_BATTERY_STATUS_AC_POWERED	Powered by AC	4

⑤ International character set

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

Constant Name	Description	Value
SII_PM_COUNTRY_USA	USA	0
SII_PM_COUNTRY_FRANCE	France	1
SII_PM_COUNTRY_GERMANY	Germany	2
SII_PM_COUNTRY_ENGLAND	United Kingdom	3
SII_PM_COUNTRY_DENMARK_1	Denmark I	4
SII_PM_COUNTRY_SWEDEN	Sweden	5
SII_PM_COUNTRY_ITALY	Italy	6
SII_PM_COUNTRY_SPAIN	Spain I	7
SII_PM_COUNTRY_JAPAN	Japan	8
SII_PM_COUNTRY_NORWAY	Norway	9
SII_PM_COUNTRY_DENMARK_2	Denmark II	10
SII_PM_COUNTRY_SPAIN_2	Spain II	11
SII_PM_COUNTRY_LATIN_AMERICA	Latin America	12
SII_PM_COUNTRY_ARABIA	Arabia	17

⑥ Codepage

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

Constant Name	Description	Value
SII_PM_CODE_PAGE_437	USA, Standard Europe (Code Page 437)	0
SII_PM_CODE_PAGE_KATAKANA	Katakana	1
SII_PM_CODE_PAGE_850	Multilingual (Code Page 850)	2
SII_PM_CODE_PAGE_860	Portuguese (Code Page 860)	3
SII_PM_CODE_PAGE_863	Canadian-French (Code page 863)	4
SII_PM_CODE_PAGE_865	Nordic (Code Page 865)	5
SII_PM_CODE_PAGE_857[*1]	Turkish (Code Page 857)	13
SII_PM_CODE_PAGE_737	Greek (Code Page 737)	14
SII_PM_CODE_PAGE_1252	Latin (Code Page 1252)	16
SII_PM_CODE_PAGE_866	Russian (Code Page 866)	17
SII_PM_CODE_PAGE_852	Eastern Europe (CodePage 852)	18
SII_PM_CODE_PAGE_858	Euro (Code Page 858)	19
SII_PM_CODE_PAGE_855	Cyrillic (Code Page 855)	34
SII_PM_CODE_PAGE_864[*1*2]	Arabic (Code Page 864)	37
SII_PM_CODE_PAGE_1250	Central European (Code Page 1250)	45
SII_PM_CODE_PAGE_1251	Cyrillic (Code Page 1251)	46
SII_PM_CODE_PAGE_1253[*3]	Greek (Code Page 1253)	47
SII_PM_CODE_PAGE_1254	Turkish (Code Page 1254)	48

*1: 20ACh of the Unicode cannot be printed.
*2: Font B cannot be printed.
*3: 00AAh of the Unicode cannot be printed.

⑦ Barcode and PDF417

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

Constant Name	Description	Value
SII_PM_BARCODE_HEIGHT_DEFAULT	Default value of barcode height	162
SII_PM_PDF417_MODULE_HEIGHT_DEFAULT	Default value of PDF417 height	10
SII_PM_PDF417_ROW_AUTO	Automatic selection of the number of rows	0
SII_PM_PDF417_COLUMN_AUTO	Automatic selection of the number of columns	0

(4) Enumerated Constant List

① Dithering (`Dithering`)

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

Constant Name	Description
SII_PM_DITHERING_DISABLE	Dithering is disabled
SII_PM_DITHERING_ERRORDIFFUSION	Dithering is enabled

② Batch processing selection (`TransactionFunction`)

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

Constant Name	Description
SII_PM_TRANSACTION_CLEAR	Cancel batch processing
SII_PM_TRANSACTION_START	Start batch processing
SII_PM_TRANSACTION_PRINT	Finish batch printing and batch processing

③ Bold print (`CharacterBold`)

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

Constant Name	Description
SII_PM_BOLD_CANCEL	Cancel bold print
SII_PM_BOLD	Specify bold print

④ Underline (`CharacterUnderline`)

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

Constant Name	Description
SII_PM_UNDERLINE_CANCEL	Cancel underline print
SII_PM_UNDERLINE_1	Specify 1-dot width underline print
SII_PM_UNDERLINE_2	Specify 2-dot width underline print

⑤ Reverse print (`CharacterReverse`)

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

Constant Name	Description
SII_PM_REVERSE_CANCEL	Cancel reverse print
SII_PM_REVERSE	Specify reverse print

⑥ Inversion print (`CharacterInversion`)

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

Constant Name	Description
SII_PM_INVERSION_CANCEL	Cancel inversion print
SII_PM_INVERSION	Specify inversion print

⑦ Character font (`CharacterFont`)

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

Constant Name	Description
SII_PM_FONT_A	Font A (24 × 12)
SII_PM_FONT_B	Font B (16 × 8)

⑧ Character scale (`CharacterScale`)

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

Constant Name	Description
SII_PM_VARTICAL_1_HORIZONTAL_1	Height × 1 and width × 1
SII_PM_VARTICAL_1_HORIZONTAL_2	Height × 1 and width × 2
SII_PM_VARTICAL_1_HORIZONTAL_3	Height × 1 and width × 3
SII_PM_VARTICAL_1_HORIZONTAL_4	Height × 1 and width × 4
SII_PM_VARTICAL_2_HORIZONTAL_1	Height × 2 and width × 1
SII_PM_VARTICAL_2_HORIZONTAL_2	Height × 2 and width × 2
SII_PM_VARTICAL_2_HORIZONTAL_3	Height × 2 and width × 3
SII_PM_VARTICAL_2_HORIZONTAL_4	Height × 2 and width × 4
SII_PM_VARTICAL_2_HORIZONTAL_6	Height × 2 and width × 6
SII_PM_VARTICAL_3_HORIZONTAL_1	Height × 3 and width × 1
SII_PM_VARTICAL_3_HORIZONTAL_2	Height × 3 and width × 2
SII_PM_VARTICAL_3_HORIZONTAL_3	Height × 3 and width × 3
SII_PM_VARTICAL_3_HORIZONTAL_4	Height × 3 and width × 4
SII_PM_VARTICAL_4_HORIZONTAL_1	Height × 4 and width × 1
SII_PM_VARTICAL_4_HORIZONTAL_2	Height × 4 and width × 2
SII_PM_VARTICAL_4_HORIZONTAL_3	Height × 4 and width × 3
SII_PM_VARTICAL_4_HORIZONTAL_4	Height × 4 and width × 4
SII_PM_VARTICAL_4_HORIZONTAL_6	Height × 4 and width × 6
SII_PM_VARTICAL_4_HORIZONTAL_8	Height × 4 and width × 8
SII_PM_VARTICAL_6_HORIZONTAL_2	Height × 6 and width × 2
SII_PM_VARTICAL_6_HORIZONTAL_4	Height × 6 and width × 4
SII_PM_VARTICAL_6_HORIZONTAL_6	Height × 6 and width × 6
SII_PM_VARTICAL_6_HORIZONTAL_8	Height × 6 and width × 8
SII_PM_VARTICAL_8_HORIZONTAL_4	Height × 8 and width × 4
SII_PM_VARTICAL_8_HORIZONTAL_6	Height × 8 and width × 6
SII_PM_VARTICAL_8_HORIZONTAL_8	Height × 8 and width × 8

⑨ Alignment (`PrintAlignment`)

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

Constant Name	Description
SII_PM_ALIGNMENT_LEFT	Aligned left
SII_PM_ALIGNMENT_CENTER	Centered
SII_PM_ALIGNMENT_RIGHT	Aligned right

⑩ Barcode symbol (`BarcodeSymbol`)

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

Constant Name	Description	Syntax*1
SII_PM_BARCODE_UPC_A	UPC-A	(a)
SII_PM_BARCODE_UPC_E	UPC-E	(a)
SII_PM_BARCODE_EAN13	EAN13	(a)
SII_PM_BARCODE_JAN13	JAN13	(a)
SII_PM_BARCODE_EAN8	EAN8	(a)
SII_PM_BARCODE_JAN8	JAN8	(a)
SII_PM_BARCODE_CODE39	CODE39	(a), (b)
SII_PM_BARCODE_CODE93	CODE93	(c)
SII_PM_BARCODE_CODE128	CODE128	(c)
SII_PM_BARCODE_ITF	ITF	(a), (b)
SII_PM_BARCODE_CODABAR	CODABAR	(a), (b)
SII_PM_BARCODE_EAN13_ADDON	EAN13 add-on	(a)
SII_PM_BARCODE_JAN13_ADDON	JAN13 add-on	(a)
SII_PM_BARCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	(a)
SII_PM_BARCODE_GS1_TRUNCATED	GS1 Databar Truncated	(a)
SII_PM_BARCODE_GS1_LIMITED	GS1 Databar Limited	(a)
SII_PM_BARCODE_GS1_EXPANDED	GS1 Databar Expanded	(a)

*1: See **`printBarcode`** or **`printPageModeBarcode`** for details of syntax.

⑪ Module size (`ModuleSize`)

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

Constant Name	Description	Method to Use
SII_PM_BARCODE_MODULE_WIDTH_2	Fine element 2 dots Module width 0.250 mm	● **printBarcode** ● **printPageModeBarcode**
SII_PM_BARCODE_MODULE_WIDTH_3	Fine element 3 dots Module width 0.375 mm	
SII_PM_BARCODE_MODULE_WIDTH_4	Fine element 4 dots Module width 0.500 mm	
SII_PM_BARCODE_MODULE_WIDTH_5	Fine element 5 dots Module width 0.625 mm	
SII_PM_BARCODE_MODULE_WIDTH_6	Fine element 6 dots Module width 0.750 mm	
SII_PM_PDF417_MODULE_WIDTH_2	Nominal fine element width 2 dots	● **printPDF417** ● **printPageModePDF417**
SII_PM_PDF417_MODULE_WIDTH_3	Nominal fine element width 3 dots	
SII_PM_PDF417_MODULE_WIDTH_4	Nominal fine element width 4 dots	
SII_PM_PDF417_MODULE_WIDTH_5	Nominal fine element width 5 dots	
SII_PM_PDF417_MODULE_WIDTH_6	Nominal fine element width 6 dots	
SII_PM_PDF417_MODULE_WIDTH_7	Nominal fine element width 7 dots	
SII_PM_PDF417_MODULE_WIDTH_8	Nominal fine element width 8 dots	
SII_PM_QR_MODULE_SIZE_2	2 dots	● **printQRcode** ● **printPageModeQRcode**
SII_PM_QR_MODULE_SIZE_3	3 dots	
SII_PM_QR_MODULE_SIZE_4	4 dots	
SII_PM_QR_MODULE_SIZE_5	5 dots	
SII_PM_QR_MODULE_SIZE_6	6 dots	
SII_PM_QR_MODULE_SIZE_7	7 dots	
SII_PM_QR_MODULE_SIZE_8	8 dots	
SII_PM_QR_MODULE_SIZE_9	9 dots	
SII_PM_QR_MODULE_SIZE_10	10 dots	
SII_PM_QR_MODULE_SIZE_11	11 dots	
SII_PM_QR_MODULE_SIZE_12	12 dots	
SII_PM_QR_MODULE_SIZE_13	13 dots	
SII_PM_QR_MODULE_SIZE_14	14 dots	
SII_PM_QR_MODULE_SIZE_15	15 dots	
SII_PM_QR_MODULE_SIZE_16	16 dots	

Constant Name	Description	Method to Use
SII_PM_DATAMATRIX_MODULE_SIZE_2	2 dots	● **printDataMatrix** ● **printPageModeDataMatrix**
SII_PM_DATAMATRIX_MODULE_SIZE_3	3 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_4	4 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_5	5 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_6	6 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_7	7 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_8	8 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_9	9 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_10	10 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_11	11 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_12	12 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_13	13 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_14	14 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_15	15 dots	
SII_PM_DATAMATRIX_MODULE_SIZE_16	16 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_2	2 dots	● **printGS1DataBarStacked** ● **printGS1DataBarStackedOmnidirectional** ● **printGS1DataBarExpandedStacked** ● **printPageModeGS1DataBarStacked** ● **printPageModeGS1DataBarStackedOmnidirectional** ● **printPageModeGS1DataBarExpandedStacked**
SII_PM_GS1DATABAR_MODULE_SIZE_3	3 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_4	4 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_5	5 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_6	6 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_7	7 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_8	8 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_9	9 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_10	10 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_11	11 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_12	12 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_13	13 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_14	14 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_15	15 dots	
SII_PM_GS1DATABAR_MODULE_SIZE_16	16 dots	

⑫ HRI character print position (`HriPosition`)

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

Constant Name	Description
SII_PM_HRI_NONE	Not printed
SII_PM_HRI_POSITION_ABOVE	Above barcode
SII_PM_HRI_POSITION_BELOW	Below barcode
SII_PM_HRI_POSITION_ABOVE_BELOW	Above and below barcode (both)

⑬ N:W ratio (`NwRatio`)

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

Constant Name	Description
SII_PM_NWRATIO_1TO2	1:2
SII_PM_NWRATIO_1TO2_5	1:2.5
SII_PM_NWRATIO_1TO3	1:3

⑭ 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
SII_PM_PDF417_ERROR_CORRECTION_0	Error correction level 0	● **printPDF417** ● **printPageModePDF417**
SII_PM_PDF417_ERROR_CORRECTION_1	Error correction level 1	
SII_PM_PDF417_ERROR_CORRECTION_2	Error correction level 2	
SII_PM_PDF417_ERROR_CORRECTION_3	Error correction level 3	
SII_PM_PDF417_ERROR_CORRECTION_4	Error correction level 4	
SII_PM_PDF417_ERROR_CORRECTION_5	Error correction level 5	
SII_PM_PDF417_ERROR_CORRECTION_6	Error correction level 6	
SII_PM_PDF417_ERROR_CORRECTION_7	Error correction level 7	
SII_PM_PDF417_ERROR_CORRECTION_8	Error correction level 8	
SII_PM_QR_ERROR_CORRECTION_L	Error correction level L	● **printQRcode** ● **printPageModeQRcode**
SII_PM_QR_ERROR_CORRECTION_M	Error correction level M	
SII_PM_QR_ERROR_CORRECTION_H	Error correction level H	
SII_PM_QR_ERROR_CORRECTION_Q	Error correction level Q	

⑮ PDF417 symbol (`Pdf417Symbol`)

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

Constant Name	Description
SII_PM_PDF417_STANDARD	PDF417
SII_PM_PDF417_COMPACT	Compact PDF417

⑯ QR Code Model (`QrModel`)

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

Constant Name	Description
SII_PM_QR_MODEL_1	QR Code Model 1
SII_PM_QR_MODEL_2	QR Code Model 2

⑰ Data Matrix module (`DataMatrixModule`)

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

Constant Name	Description
SII_PM_DATA_MATRIX_AUTO	Number of modules: Automatic
SII_PM_DATA_MATRIX_10_10	Number of modules: 10 × 10
SII_PM_DATA_MATRIX_12_12	Number of modules: 12 × 12
SII_PM_DATA_MATRIX_14_14	Number of modules: 14 × 14
SII_PM_DATA_MATRIX_16_16	Number of modules: 16 × 16
SII_PM_DATA_MATRIX_18_18	Number of modules: 18 × 18
SII_PM_DATA_MATRIX_20_20	Number of modules: 20 × 20
SII_PM_DATA_MATRIX_22_22	Number of modules: 22 × 22
SII_PM_DATA_MATRIX_24_24	Number of modules: 24 × 24
SII_PM_DATA_MATRIX_26_26	Number of modules: 26 × 26
SII_PM_DATA_MATRIX_32_32	Number of modules: 32 × 32
SII_PM_DATA_MATRIX_36_36	Number of modules: 36 × 36
SII_PM_DATA_MATRIX_40_40	Number of modules: 40 × 40
SII_PM_DATA_MATRIX_44_44	Number of modules: 44 × 44
SII_PM_DATA_MATRIX_48_48	Number of modules: 48 × 48
SII_PM_DATA_MATRIX_52_52	Number of modules: 52 × 52
SII_PM_DATA_MATRIX_64_64	Number of modules: 64 × 64
SII_PM_DATA_MATRIX_72_72	Number of modules: 72 × 72
SII_PM_DATA_MATRIX_80_80	Number of modules: 80 × 80

Constant Name	Description
SII_PM_DATA_MATRIX_88_88	Number of modules: 88 × 88
SII_PM_DATA_MATRIX_96_96	Number of modules: 96 × 96
SII_PM_DATA_MATRIX_104_104	Number of modules: 104 × 104
SII_PM_DATA_MATRIX_120_120	Number of modules: 120 × 120
SII_PM_DATA_MATRIX_132_132	Number of modules: 132 × 132
SII_PM_DATA_MATRIX_144_144	Number of modules: 144 × 144
SII_PM_DATA_MATRIX_8_18	Number of modules: 8 × 18
SII_PM_DATA_MATRIX_8_32	Number of modules: 8 × 32
SII_PM_DATA_MATRIX_12_26	Number of modules: 12 × 26
SII_PM_DATA_MATRIX_12_36	Number of modules: 12 × 36
SII_PM_DATA_MATRIX_16_36	Number of modules: 16 × 36
SII_PM_DATA_MATRIX_16_48	Number of modules: 16 × 48

⑱ MaxiCode Mode (`MaxiCodeMode`)

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

Constant Name	Description
SII_PM_MAXI_CODE_2	Mode2
SII_PM_MAXI_CODE_3	Mode3
SII_PM_MAXI_CODE_4	Mode4
SII_PM_MAXI_CODE_5	Mode5

⑲ Cutting method (`CuttingMethod`)

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

Constant Name	Description
SII_PM_CUT_FULL	No cut Paper feed operation to the paper cut position
SII_PM_CUT_PARTIAL	
SII_PM_CUT_NONE[*1]	No cut

*1: Supported only by **`printPageMode`**.

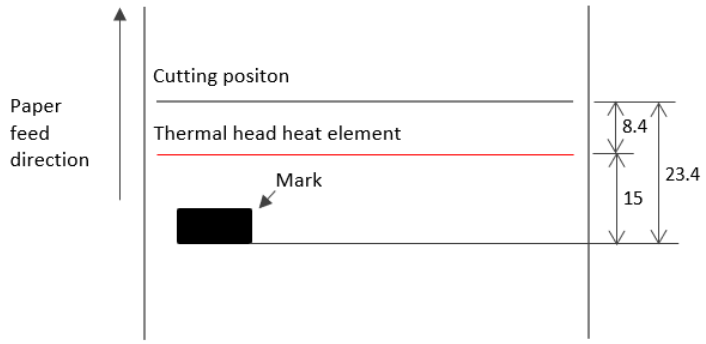
⑳ Form feed position (`FeedPosition`)

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

Constant Name	Description
SII_PM_FEED_CUTTER	After detecting the mark, feeds the paper to the cutting position. The paper feed length is the length of the memory switches MS 21 to 22 (Mark Position Correction) of the printer. The default of the paper feed is 0 dots (0 mm).

Referenece See "MP-A40 SERIES Thermal Printer USER'S GUIDE" for the details of the memory switch of the printer.
The memory switch of the printer can be changed in the iOS app "SII Printer Utility" on the App Store.

The relation between the mark position and the initial values of the printer memory switches MS21 to 22 (Mark Position Correction) is shown in the following figure.



Unit : mm

㉑ Image rotation direction (`Rotate`)

Constants of enumerated type used for image rotation direction are shown in the following table.

Constant Name	Description
SII_PM_ROTATE_NONE	No rotation
SII_PM_ROTATE_180	Rotate 180 degrees

㉒ Image scaling (`ImageScale`)

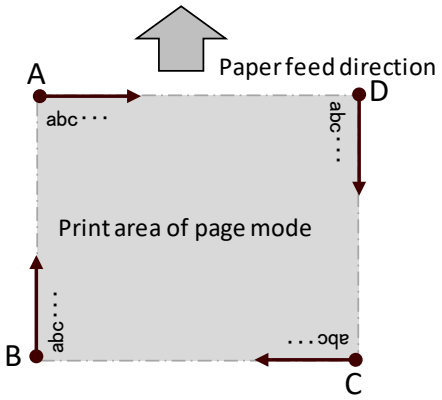
Constants of enumerated type used for image scaling are shown in the following table.

Constant Name	Description
SII_PM_IMAGE_SCALE_NONE	No scaling
SII_PM_IMAGE_SCALE_WIDTH_FIT	Scale to fit print width

㉓ Print direction (`Direction`)

Constants of enumerated type used for print direction in page mode are shown in the following table.

Constant Name	Description
SII_PM_DIRECTION_LEFT_TO_RIGHT	Starting point: Upper left (A on the figure), Print direction: Left to Right
SII_PM_DIRECTION_BOTTOM_TO_TOP	Starting point: Left below (B on the figure), Print direction: Below to Upper
SII_PM_DIRECTION_RIGHT_TO_LEFT	Starting point: Right below (C on the figure), Print direction: Right to Left
SII_PM_DIRECTION_TOP_TO_BOTTOM	Starting point: Upper right (D on the figure), Print direction: Upper to Below



㉔ Line style (`LineStyle`)

Constants of enumerated type used for line style in page mode are shown in the following table.

Constant Name	Description
SII_PM_LINESTYLE_THIN	Thin solid line (2 dots)
SII_PM_LINESTYLE_MEDIUM	Medium solid line (4 dots)
SII_PM_LINESTYLE_THICK	Thick solid line (8 dots)

(5) Method Details

① Common method to standard mode and page mode

The following methods are valid in standard mode and page mode. Standard mode is set immediately after **`connect`** is executed.

`init` Instance

Syntax
```
- (id) init;
```

Description This method initializes the instance of **`SIIPrinterManager`** class.

Return value When succeeded, the initialized instance of **`SIIPrinterManager`** class is returned. When failed, nil is returned.

Example of use
```
SIIPrinterManager *printerManager =
 [[SIIPrinterManager alloc] init];
```

`connect` Start communicating with printer

Starts communicating with the printer.

Syntax
```
- (void) connect:(NSInteger)printerModel
                 address:(NSString)address
                 portType:(NSInteger)portType;
```

Parameter

`printerModel`	Printer model constant. See "4.3.1(3)① Printer model" for available constants.
`address`	Depends on the setting of `portType`. ・For **SII_PM_PRINTER_PORT_TYPE_BLUETOOTH**: Specify the Bluetooth device name (Bluetooth Accessory). Example: "MP-A40"
`portType`	Port type See "4.3.1(3)② Port type" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description Call this method before using other **`SIIPrinterManager`** class methods.
In order to make this library work properly, this method may change the printer settings when connecting.

For Bluetooth connection:
Communication with a printer paired with iOS device starts through Bluetooth connection. Connection is made to the paired Bluetooth device (Bluetooth accessory) specified by `address`.

Note A concurrent connection from multiple apps to one printer is not supported.

disconnect — Stop communicating with printer

Stops communicating with the printer.

Syntax

```
- (void) disconnect;
```

Error

SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException** Class" for details on the error.

Note

It is recommended to get execution response by **SII_PM_PRINTER_RESPONSE_REQUEST** of **getPrinterResponse** before executing this method. If not, the following problems may occur:

- The communication is disconnected before the print data sending from iOS device to the printer is completed, and a part of the data may be lost.
- In Bluetooth connection, when either **disconnect** or **connect** is executed while the printer is in the buffer full state*1, the communication between iOS device and the printer may be disconnected.

  *1: The state of buffer full means that the buffer of the printer is filled with print data. The size to be in buffer full state is approximately 4K bytes.

If you do not execute **getPrinterResponse** in your program, please fully evaluate your program to confirm no problems arise.

openDrawer — Open cash drawer

This method is not supported. When executed, **SIIPrinterException** is thrown.

Syntax

```
- (void) openDrawer:(DrawerNum)drawerNum
                onOffTime:(PulseWidth)onOffTime;
```

buzzer — Sound buzzer

This method is not supported. When executed, **SIIPrinterException** is thrown.

Syntax

```
- (void) buzzer:(NSInteger)onTime
                offTime:(NSInteger)offTime;
```

externalBuzzer — Sound external buzzer

This method is not supported. When executed, **SIIPrinterException** is thrown.

Syntax

```
- (void) externalBuzzer:(BuzzerPattern)buzzerPattern
                 buzzerCount:(NSInteger)buzzerCount;
```

getStatus — Get printer status

Gets the latest printer status.

Syntax

```
- (void) getStatus:(NSInteger[])buf;
```

Parameter `buf` Status retrieved from the printer

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException** Class" for details on the error.

Description The status retrieved from the printer is stored in an NSInteger array.

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

Bit	Function	Value	
		0	1
0	Voltage error	No error	Error
1	Hardware error	No error	Error
2	Head temperature error	No error	Error
3	Reserved	Fixed	-
4	Out-of-paper error	No error	Error
5	Reserved	Fixed	-
6	Paper jam error while detecting mark	No error	Error
7	Cover open error	No error	Error
8	FEED Switch status	OFF	ON
9	Reserved	Fixed	-
10	Paper feed status	Stop	Operating
11	Return-waiting status	Not waiting	Waiting
12	Reserved	Fixed	-
13	Reserved	-	Fixed
14	Reserved	-	Fixed
15	Reserved	-	Fixed
16	FLASH memory rewriting	Not rewriting	Rewriting
17	Reserved	-	Fixed
18	Reserved	-	Fixed
19	Reserved	-	Fixed

Bit	Function	Value	
		0	1
20 to 22	Battery remaining capacity level	000: No battery 001: Level 4, 5 010: Level 3 011: Level 2 100: Level 1	
23	Battery error	No error	Error
24 to 31	Reserved	-	Fixed

abort — Abort waiting state of printer

Aborts the waiting state of the printer.

Syntax
```
- (void) abort;
```

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException** Class" for details on the error.

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

registerLogo — Register logo

Registers image file to the NV graphics memory in the printer as a logo.
In syntax (a), dithering can be specified.
In syntax (b), dithering is fixed to be disabled.

Syntax
```
(a) - (void) registerLogo:(NSString *)fileName
                   logoId:(NSString *)logoId
                   dithering:(Dithering)dithering;

(b) - (void) registerLogo:(NSString *)fileName
                   logoId:(NSString *)logoId;
```

Parameter

`fileName` File path of image file to be registered as a logo
Supported image file file extensions are .bmp, .jpg, .jpeg, and .png. Colored image is converted to monochrome image by binarization and registered.

`logoId` ID of the logo to be registered (key code)
Specify the ID of the logo to be registered as a two-character string.
Valid characters are ASCII character codes from 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z').

`dithering` Dithering
See "4.3.1(4)① Dithering (`Dithering`)" for available constants.

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException** Class" for details on the error.

unregisterLogo — Delete registered logo

Deletes the registered logo.

Syntax
```
- (void) unregisterLogo:(NSString *)logoId;
```

Parameter

`logoId`	ID of the logo to be deleted (key code) Specify the ID of the registered logo as a character string.

Error
`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

registerStyleSheet — Register style sheet

Registers CSS file to the printer as a style sheet.

Syntax
```
- (void) registerSytleSheet:(NSString *)fileName
                     cssId:(NSInteger)cssId;
```

Parameter

`fileName`	File path of CSS file to be registered as a style sheet The style sheet is effective when the style sheet language is CSS (Cascading Style Sheets) and the file extension is .css. The maximum number of styles that can be registered in a single CSS file is 64.
`cssId`	ID of the style sheet to be registered Set the ID of the style sheet to be registered. The valid range is 1 to 4.

Error
`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Note
See "Tag Processing Mode" in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for more details about style sheet.

unregisterStyleSheet — Delete registered style sheet

Deletes the registered style sheet.

Syntax
```
- (void) unregisterStyleSheet:(NSInteger)cssId;
```

Parameter

`cssId`	ID of the style sheet to be deleted Specify the ID of the registered style sheet.

Error
`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

resetPrinter — Reset printer

Performs a hardware reset of the printer.

Syntax
```
- (void) resetPrinter;
```

Error
SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description
For Bluetooth connection:
The printer hardware reset is performed by the printer command "Printer Reset". When using this method, enable iOS Auto Connection in the iOS app "SII Printer Utility" on the App Store. When it is disabled, this method fails to reconnect after reset and **SIIPrinterException** is thrown.
This method takes about 10 seconds to complete reconnection with the printer after performing the reset. Use this method after setting a sufficient receive timeout period.

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

getPrinterResponse — Get various responses from printer

Gets response data from the printer.

Syntax
```
- (void) getPrinterResponse:(NSInteger)responseId
                      param:(NSObject *)param
                   response:(void *)response;
```

Parameter

`responseId` — Response type constant
See "4.3.1(3)③ Response type" for available constants.

`param` — Command parameter
The value to be specified varies with the response type constant.
See the following table for description of the value to be specified.

`response` — Buffer for storing the retrieved response data
The buffer type varies with the response type constant.
See the following table for the buffer type.

Response Type Constant		
	Parameter	**Description**
SII_PM_PRINTER_RESPONSE_REQUEST (Execution response request)		
	`param`	Specify 0 to 15 (00h to 0Fh) in NSData type.
	`response`	Specify an NSInteger array of length 1. When the response is retrieved successfully, the response code of the execution response request is stored with 128 to 143 (80h to 8Fh).
SII_PM_PRINTER_RESPONSE_USER_AREA (Send remaining capacity of user area)		
	`param`	Specify nil.
	`response`	Specify an NSInteger 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.

Response Type Constant		
	Parameter	**Description**
SII_PM_PRINTER_RESPONSE_ARRANGE_USER_AREA (Send remaining capacity of user area after defragment)		
	param	Specify nil.
	response	Specify an NSInteger 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.
SII_PM_PRINTER_RESPONSE_NV_GRAPHICS (Send NV graphics memory capacity)		
	param	Specify nil.
	response	Specify an NSInteger array of length 1. When the response is retrieved successfully, the NV graphics memory capacity is stored as a numerical value in bytes.
SII_PM_PRINTER_RESPONSE_KEY_CODE (Send key code list of defined NV graphics)		
	param	Specify nil.
	response	Specify an NSMutableArray array. When the response is retrieved successfully, the key code of NV graphics is stored as a string array.
SII_PM_PRINTER_RESPONSE_BATTERY_STATUS (Battery remaining capacity level)		
	param	Specify nil.
	response	Specify an NSInteger array of length 1. When the response is retrieved successfully, the battery remaining capacity level is stored as a value. See "4.3.1(3)④ Battery remaining capacity level" for details of the value. Battery remaining capacity level: **SII_PM_BATTERY_STATUS_FULL** : Full (Battery remaining capacity: approx. 80%) **SII_PM_BATTERY_STATUS_MIDDLE**: Middle (Battery remaining capacity: approx. 40%) **SII_PM_BATTERY_STATUS_LOW**: Low (Battery remaining capacity: approx. 10%) **SII_PM_BATTERY_STATUS_EMPTY**: No battery **SII_PM_BATTERY_AC_POWERED**: Powered by AC
SII_PM_PRINTER_RESPONSE_FIRMWARE_VERSION (Send firmware version)		
	param	Specify nil.
	response	Specify an NSMutableArray array. When the response is retrieved successfully, the firmware version is stored as a string array.

Response Type Constant		
	Parameter	**Description**
SII_PM_PRINTER_RESPONSE_MAINTENANCE_COUNTER (Send maintenance counter)		
	`param`	Specify nil.
	`response`	Specify an NSMutableArray array. When the response is retrieved successfully, the maintenance counter value is stored as an NSNumber object as shown below. response[0] : Number of paper feed lines (unit: 100 dot-line) response[1] : Number of thermal head activation times (unit: 100 dot-line) response[2] : Number of autocutter drive times*1 response[3] : Drive time of printer (unit: minute) response[4] : Number of paper feed lines (integrated value) (unit: 100 dot-line) response[5] : Number of thermal head activation times (integrated value) (unit: 100 dot-line) response[6] : Number of autocutter drive times (integrated value)*1 response[7] : Drive time of printer (integrated value) (unit: minute)
SII_PM_PRINTER_RESPONSE_PRINTER_IDENTIFIER (Send printer ID)		
	`Param`	Specify nil.
	`response`	Specify an NSMutableArray array. When the response is retrieved successfully, **`SIIPrinterIdentifier`** class object is stored in the first element of response. Example: response[0] = "(**`SIIPrinterIdentifier`** class object)" See "**4.3.3 `SIIPrinterIdentifier` Class**" for the **`SIIPrinterIdentifier`** class object.

*1: Not supported. Always returns 0.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

`startDiscoveryPrinter` Start printer search (Bluetooth)

Searches for the Bluetooth device (Bluetooth accessory).

Syntax
```
- (void) startDiscoveryPrinter:(NSPredicate *)predicate
      completion:(EABluetoothAccessoryPickerCompletion)completion;
```

Parameter

`predicate`	Specify nil.
`completion`	Completion event of **`EABluetoothAccessoryPickerCompletion`** Specify ^(`NSError *error`) to receive the completion event of **`EABluetoothAccessoryPickerCompletion`**.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.

Description This method searches for the Bluetooth device (Bluetooth accessory). This method is calling `showBluetoothAccessoryPickerWithNameFilter` of `EAAccessoryManager` internally. Pairing with the Bluetooth device can be performed in the window displayed at execution of this method.

Example when specifying ^(NSError *error)
(Statement of EABluetoothAccessoryPickerCompletion)

```
typedef
void(^EABluetoothAccessoryPickerCompletion)(NSError *error);
```

startDiscoveryPrinter — Start printer search (TCP/IP)

This method is not supported. When executed, SII printers other than the MP-A40 are searched for.

Syntax

```
- (void) startDiscoveryPrinter:(NSInteger)retryCount
                       timeout:(NSInteger)timeout
                    completion:(SIIDiscoveryPrinterCompletion)completion;
```

cancelDiscoveryPrinter — Cancel printer search

This method is not supported. When executed, the ongoing **startDiscoveryPrinter** (TCP/IP) process is interrupted.

Syntax

```
- (void) cancelDiscoveryPrinter;
```

getFoundPrinter — Get found printer information

This method is not supported. When executed, the printer information discovered by **startDiscoveryPrinter** (TCP/IP) is returned as an NSArray.

Syntax

```
- (NSArray *) getFoundPrinter;
```

getVersion — Get SDK version

Gets the SDK version as a character string.

Syntax

```
- (NSString *) getVersion;
```

Return value: SDK version character string (Example: When the SDK version is Ver.1.0.0, the return value is "1.0.0")

Description: This method can be executed regardless of whether **isConnect** is YES or NO.

controlTransaction — Start/End batch processing

Starts or ends batch processing.

Syntax

```
- (void) controlTransaction:(TransactionFunction)control;
```

Parameter

control		Batch processing selection See "4.3.1(4)② Batch processing selection (TransactionFunction)" for available constants.

Error: **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description: The procedure of batch processing is as follows:

(1) Start batch processing.
Specify **SII_PM_TRANSACTION_START.**

(2) Execute the method.
In the case of the batch processing target method, buffering of transmission data is started.
The transmission data of the batch processing target method executed during buffering is buffered in the transmission buffer without being sent to the printer.
The maximum size of transmission data to be buffered is system dependent.
If the buffered transmission data exceeds the maximum size, the batch processing target method at the time of exceeding becomes an error. If an error occurs, the transmission data up to the error is retained.
As for the retained transmission data, finish the batch processing in step (3).
In the case of a method other than the batch processing target method, transmission data is immediately executed without being buffered.

(3) Finish batch processing.
When **SII_PM_TRANSACTION_PRINT** is specified, the buffered transmission data is sent to the printer. The buffered transmission data is retained even after sent to the printer.
The retained transmission data is discarded by any of the following:
・Specify **SII_PM_TRANSACTION_CLEAR**
・Specify **SII_PM_TRANSACTION_START**
・Execute **disconnect**

The batch processing target methods are as follows:
- **sendText**
- **sendTextEx**
- **printBarcode**
- **printPDF417**
- **printQRcode**
- **printDataMatrix**
- **printMaxiCode**
- **printGS1DataBarStacked**
- **printGS1DataBarStackedOmnidirectional**
- **printGS1DataBarExpandedStacked**
- **cutPaper**
- **feedPosition**
- **openDrawer**
- **buzzer**
- **sendBinary**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printLogo**[*1]
- **enterPageMode**
- **exitPageMode**
- **setPageModeArea**
- **setPageModeDirection**
- **setPageModeLineSpacing**
- **printPageMode**
- **printPageModeText**
- **printPageModeTextEx**
- **printPageModeBarcode**
- **printPageModePDF417**
- **printPageModeQRcode**
- **printPageModeDataMatrix**
- **printPageModeMaxiCode**

- **printPageModeGS1DataBarStacked**
- **printPageModeGS1DataBarStackedOmnidirectional**
- **printPageModeGS1DataBarExpandedStacked**
- **sendPageModeBinary**
- **printPageModeImageFile**
- **printPageModeRectangle**
- **printPageModeLine**
- **printPageModeLogo**[*1]

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

② Dedicated method for standard mode

The following methods are valid in standard mode. **SIIPrinterException** is thrown when the dedicated method for standard mode are executed in page mode.

sendText — Send text data

Sends text data.

Syntax

```
- (void) sendText:(NSString *)text;
```

Parameter

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

Error

SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description

This method encodes the specified text data to printable text data based on **internationalCharacter** and **codePage**, and sends it to the printer.

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

sendTextEx — Send format specified text data

Sends text data with specified formatting to the printer.
In syntax (a), bold, underline, reverse printing, font, character scale and alignment can be specified as text formatting.
In syntax (b), bold, underline, font and character scale can be specified as text formatting.
In syntax (c), bold, underline, inversion printing, reverse printing, font, character scale and alignment can be specified as text formatting.

Syntax

```
(a)- (void) sendTextEx:(NSString *)text
                  bold:(CharacterBold)bold
                  underline:(CharacterUnderline)underline
                  reverse:(CharacterReverse)reverse
                  font:(CharacterFont)font
                  scale:(CharacterScale)scale
                  alignment:(PrintAlignment)alignment;

(b)- (void) sendTextEx:(NSString *)text
                  bold:(CharacterBold)bold
                  underline:(CharacterUnderline)underline
                  font:(CharacterFont)font
                  scale:(CharacterScale)scale;

(c)- (void) sendTextEx:(NSString *)text
                  bold:(CharacterBold)bold
                  underline:(CharacterUnderline)underline
                  reverse:(CharacterReverse)reverse
                  inversion:(CharacterInversion)inversion
                  font:(CharacterFont)font
                  scale:(CharacterScale)scale
                  alignment:(PrintAlignment)alignment;
```

Parameter

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

	`bold`	Bold print See "4.3.1(4)③ Bold print (`CharacterBold`)" for available constants.
	`underline`	Underline See "4.3.1(4)④ Underline (`CharacterUnderline`)" for available constants.
	`reverse`	Reverse print See "4.3.1(4)⑤ Reverse print (`CharacterReverse`)" for available constants.
	`inversion`	Inversion print See "4.3.1(4)⑥ Inversion print (`CharacterInversion`)" for available constants.
	`font`	Character font See "4.3.1(4)⑦ Character font (`CharacterFont`)" for available constants.
	`scale`	Character scale See "4.3.1(4)⑧ Character scale (`CharacterScale`)" for available constants.
	`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description This method encodes the formatted text data to printable text data based on **`internationalCharacter`** and **`codePage`**, and sends it to the printer.

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

`printBarcode` Print barcode

Prints the barcode.
In syntax (a), barcode data can be specified as a character string.
In syntax (b), barcode data can be specified as a character string, and the N:W ratio of the barcode can also be specified.
In syntax (c), barcode data can be specified as a byte array.
The syntax (d) is not supported.

Syntax

```
(a) - (void) printBarcode:(BarcodeSymbol)barcodeSymbol
                     text:(NSString *)text
                     moduleSize:(ModuleSize)moduleSize
                     moduleHeight:(NSInteger)moduleHeight
                     hriPosition:(HriPosition)hriPosition
                     hriFont:(CharacterFont)hriFont
                     alignment:(PrintAlignment)alignment;
```

```
(b) - (void) printBarcode:(BarcodeSymbol)barcodeSymbol
                    text:(NSString *)text
                    moduleSize:(ModuleSize)moduleSize
                    moduleHeight:(NSInteger)moduleHeight
                    hriPosition:(HriPosition)hriPosition
                    hriFont:(CharacterFont)hriFont
                    alignment:(PrintAlignment)alignment
                    nwRatio:(NwRatio)nwRatio;

(c) - (void) printBarcode:(BarcodeSymbol)barcodeSymbol
                    data:(NSData*)data
                    moduleSize:(ModuleSize)moduleSize
                    moduleHeight:(NSInteger)moduleHeight
                    hriPosition:(HriPosition)hriPosition
                    hriFont:(CharacterFont)hriFont
                    alignment:(PrintAlignment)alignment;

(d) - (void) printBarcode:(BarcodeSymbol)barcodeSymbol
                    text:(NSString *)text
                    moduleSize:(ModuleSize)moduleSize
                    alignment:(PrintAlignment)alignment;
```

Parameter

`barcodeSymbol` Barcode symbol
See "4.3.1(4)⑩ Barcode symbol (`BarcodeSymbol`)" for available constants and corresponding syntax.

`text(data)` Barcode data to send to the printer
The input conditions for barcode are as follows.

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
UPC-A	11 to 12 characters	'0' to '9'	
UPC-E	11 to 12 characters	'0' to '9'	
EAN13 JAN13	12 to 13 characters '	'0' to '9'	
EAN8 JAN8	7 to 8 characters	'0' to '9'	
CODE39	1 to 150 characters	'0' to '9' 'A' to 'Z' ' ', '$', '%', '+', '-', '.', '/'	Start code and stop code ('*') are automatically added.
CODE93	1 to 150 bytes	(0x00 to 0x2E)	Input data with 0x2F or more at the end.
CODE128	2 to 150 bytes	(0x00 to 0x66)	When inputting the start code (0x67 to 0x69) of the CODE128 code set. Input data with 0x67 or more at the end.
		(0x00 to 0x7F)	When starting with a CODE128 special code start code ("{A", "{B", "{C").
ITF	2 to 150 characters (However, an even number)	'0' to '9'	

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
CODABAR	1 to 150 characters	'0' to '9' '$', '+', '-', '.', '/', ':'	It is needed to specify one of 'A' to 'D' at the beginning and end.
EAN13 add-on JAN13 add-on	Add-on 2: 14 to 15 characters Add-on 5: 17 to 18 characters	'0' to '9'	
Customer Bar Code_JP	-	-	Not supported.
GS1 Databar Omni-directional	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Truncated	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Limited	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Expanded	2 to 255 characters	' ' to "" '%' to '?' 'A' to 'Z' '_' 'a' to 'z' '{'	

`moduleSize` Barcode width
See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

`moduleHeight` Barcode height (dot)

- When `barcodeSymbol` is below, the valid range is 1 to 255.
  SII_PM_BARCODE_UPC_A
  SII_PM_BARCODE_UPC_E
  SII_PM_BARCODE_EAN13
  SII_PM_BARCODE_JAN13
  SII_PM_BARCODE_EAN8
  SII_PM_BARCODE_JAN8
  SII_PM_BARCODE_CODE39
  SII_PM_BARCODE_CODE93
  SII_PM_BARCODE_CODE128
  SII_PM_BARCODE_ITF
  SII_PM_BARCODE_CODABAR
  SII_PM_BARCODE_EAN13_ADDON
  SII_PM_BARCODE_JAN13_ADDON

- When `barcodeSymbol` is below, the valid range varies depending on `barcodeSymbol` and `moduleSize`.

`barcodeSymbol`	`moduleSize`	**Valid Range**
SII_PM_BARCODE_GS1_OMNI_DIRECTIONAL		
	SII_PM_BARCODE_MODULE_WIDTH_2	66 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	99 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	132 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	165 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	198 to 255
SII_PM_BARCODE_GS1_TRUNCATED		
	SII_PM_BARCODE_MODULE_WIDTH_2	26 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	39 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	52 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	65 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	78 to 255
SII_PM_BARCODE_GS1_LIMITED		
	SII_PM_BARCODE_MODULE_WIDTH_2	20 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	30 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	40 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	50 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	60 to 255
SII_PM_BARCODE_GS1_EXPANDED		
	SII_PM_BARCODE_MODULE_WIDTH_2	68 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	102 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	136 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	170 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	204 to 255

`hriPosition` HRI character print position
See "4.3.1(4)⑫ HRI character print position (`HriPosition`)" for available constants.

`hriFont` HRI character font
See "4.3.1(4)⑦ Character font (`CharacterFont`)" for available constants.

`alignment` Alignment
See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

`nwRatio` N:W ratio
See "4.3.1(4)⑬ N:W ratio (`NwRatio`)" for available constants.
Depending on the specified `nwRatio` and `moduleSize`, the width of the wide element is set as shown in the following table.

`moduleSize`	`nwRatio`		
	SII_PM_ NWRATIO_1TO2	**SII_PM_ NWRATIO_1TO2_5**	**SII_PM_ NWRATIO_1TO3**
SII_PM_BARCODE_MODULE_WIDTH_2	0.500 mm (4 dots)	0.625 mm (5 dots)	0.750 mm (6 dots)
SII_PM_BARCODE_ MODULE_WIDTH_3	0.750 mm (6 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)
SII_PM_BARCODE_ MODULE_WIDTH_4	1.000 mm (8 dots)	1.250 mm (10 dots)	1.500 mm (12 dots)
SII_PM_BARCODE_MODULE_WIDTH_5	1.250 mm (10 dots)	1.625 mm (13 dots)	1.875 mm (15 dots)
SII_PM_BARCODE_MODULE_WIDTH_6	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

`printPDF417` Print PDF417

Prints PDF417.
In syntax (a), the PDF417 symbol can be specified.
In syntax (b), the PDF417 symbol is fixed to standard PDF417.

Syntax

```
(a) - (void) printPDF417:(NSString *)text
                    errorCorrection:(ErrorCorrection)errorCorrection
                    row:(NSInteger)row
                    column:(NSInteger)column
                    moduleSize:(ModuleSize)moduleSize
                    moduleHeight:(NSInteger)moduleHeight
                    alignment:(PrintAlignment)alignment
                    pdf417Symbol:(Pdf417Symbol)pdf417Symbol;

(b) - (void) printPDF417:(NSString *)text
                    errorCorrection:(ErrorCorrection)errorCorrection
                    row:(NSInteger)row
                    column:(NSInteger)column
                    moduleSize:(ModuleSize)moduleSize
                    moduleHeight(NSInteger)moduleHeight
                    alignment:(PrintAlignment)alignment;
```

Parameter

`text` Barcode data to send to the printer

`errorCorrection` Error correction level
See "4.3.1(4)⑭ Error correction level (`ErrorCorrection`)" for available constants.

	`row`	Number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
	`column`	Number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
	`moduleSize`	Nominal fine element width See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
	`moduleHeight`	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
	`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.
	`pdf417Symbol`	PDF417 symbol See "4.3.1(4)⑮ PDF417 symbol (`Pdf417Symbol`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

`printQRcode` Print QR Code

Prints QR Code.
In syntax (a), QR Code Model can be specified.
In syntax (b), QR Code Model is fixed to QR Code Model 2.

Syntax

```
(a) - (void) printQRcode:(NSString *)text
                    errorCorrection(ErrorCorrection)errorCorrection
                    moduleSize:(ModuleSize)moduleSize
                    alignment:(PrintAlignment)alignment
                    model:(QrModel)model;

(b) - (void) printQRcode:(NSString *)text
                    errorCorrection:(ErrorCorrection)errorCorrection
                    moduleSize:(ModuleSize)moduleSize
                    alignment:(PrintAlignment)alignment;
```

Parameter

`text`	Barcode data to send to the printer The version is automatically set depending on the number of data bytes set with `text` in either syntax (a) and (b).
`errorCorrection`	Error correction level See "4.3.1(4)⑭ Error correction level (`ErrorCorrection`)" for available constants.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

	`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.
	`model`	QR Code Model See "4.3.1(4)⑯ QR Code Model (`QrModel`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

`printDataMatrix` Print Data Matrix

Prints Data Matrix.

Syntax
```
- (void) printDataMatrix:(NSString *)text
                  dataMatrixModule:(DataMatrixModule)dataMatrixModule
                  moduleSize:(ModuleSize)moduleSize
                  alignment:(PrintAlignment)alignment;
```

Parameter	`text`	Barcode data to send to the printer
	`dataMatrixModule`	Number of Data Matrix modules See "4.3.1(4)⑰ Data Matrix module (`DataMatrixModule`)" for available constants.
	`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
	`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printMaxiCode — Print MaxiCode

Prints MaxiCode.

Syntax

```
- (void) printMaxiCode:(NSString *)text
                maxiCodeMode:(MaxiCodeMode)maxiCodeMode
                alignment:(PrintAlignment)alignment;
```

Parameter

`text`	Barcode data to send to the printer • When `maxiCodeMode` is **SII_PM_MAXI_CODE_2**: Add service class (3 digits), country code (3 digits), and postal code (9 digits) to the beginning of the data. • When `maxiCodeMode` is **SII_PM_MAXI_CODE_3**: Add service class (3 digits), country code (3 digits), and postal code (6 digits) to the beginning of the data.
`maxiCodeMode`	MaxiCode Mode See "4.3.1(4)⑱ MaxiCode Mode (`MaxiCodeMode`)" for available constants.
`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference

See "Appendix B Barcode Size List" for details of the barcode size.

printGS1DataBarStacked — Print GS1 Databar Stacked

Prints GS1 Databar Stacked.

Syntax

```
- (void) printGS1DataBarStacked:(NSString *)text
                moduleSize:(ModuleSize)moduleSize
                alignment:(PrintAlignment)alignment;
```

Parameter

`text`	Barcode data to send to the printer Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference

See "Appendix B Barcode Size List" for details of the barcode size.

printGS1DataBarStackedOmnidirectional — Print GS1 Databar Stacked Omni-directional

Prints GS1 Databar Stacked Omni-directional.

Syntax

```
- (void) printGS1DataBarStackedOmnidirectional:(NSString *)text
                    moduleHeight:(NSInteger)moduleHeight
                    moduleSize:(ModuleSize)moduleSize
                    alignment:(PrintAlignment)alignment;
```

Parameter

`text`	Barcode data to send to the printer Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
`moduleHeight`	Barcode module height (number of modules) The valid range is 33 to 255.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Reference

See "Appendix B Barcode Size List" for details of the barcode size.

printGS1DataBarExpandedStacked — Print GS1 Databar Expanded Stacked

Prints GS1 Databar Expanded Stacked.

Syntax

```
- (void) printGS1DataBarExpandedStacked:(NSString *)text
                    column:(NSInteger)column
                    moduleSize:(ModuleSize)moduleSize
                    alignment:(PrintAlignment)alignment;
```

Parameter

`text`	Barcode data to send to the printer Enter any number of characters using the following: ' ', '!', '"', '%', '&', ''', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z' Enter '{1' for FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.
`column`	Number of columns Specify the number of columns in 1 line. An even number from 2 to 20 is valid.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printAztecCode Print Aztec Code

This method is not supported. When executed, **SIIPrinterException** is thrown.

Syntax
```
- (void) printAztecCode:(NSString *)text
                  layer:(NSInteger)layer
         errorCorrection:(NSInteger)errorCorrection
              moduleSize:(ModuleSize)moduleSize
             aztecSymbol:(AztecSymbol)aztecSymbol
               alignment:(PrintAlignment)alignment;
```

cutPaper Cut paper

Feeds the paper to the paper cut position. The paper is not cut.

Syntax
```
- (void) cutPaper:(CuttingMethod)cuttingMethod;
```

Parameter

cuttingMethod	Cutting method See "4.3.1(4)⑲ Cutting method (CuttingMethod)" for available constants.

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

feedPosition Paper form feed

Performs the paper form feed of marked paper.

Syntax
```
- (void) feedPosition:(FeedPosition)feedPosition;
```

Parameter

feedPosition	Form feed position See "4.3.1(4)⑳ Form feed position (FeedPosition)" for available constants.

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Note The paper form feed is not performed when this method is executed at the form feed position of the marked paper.

sendBinary — Send binary data

Sends binary data to the printer.

Syntax

```
- (void) sendBinary:(NSData*)data;
```

Parameter

`data`	Binary data to send to the printer Data size that can be specified at one time is 3 MB(3145728 bytes).

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description

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

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

sendDataFile — Send specified file

Sends file data.
In syntax (a), dithering can be specified.
In syntax (b), dithering is fixed to disabled.

Syntax

```
(a) - (void) sendDataFile:(NSString *)fileName
                alignment:(PrintAlignment)alignment
                dithering:(Dithering)dithering;

(b) - (void) sendDataFile:(NSString *)fileName
                alignment:(PrintAlignment)alignment;
```

Parameter

`fileName`	File path of data to sent to the printer The maximum file size that can be specified is 16 MB(16777216 bytes). The file extensions that can be sent and the file transmission are described below. • .bmp, .jpg, .jpeg, .png Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and sent to the printer. Printing is performed in batch after mapping the image file on the memory of the printer. • .txt Data is sent to the printer as text data. Text data format supports UTF-8. This method encodes the text data to printable text data based on the settings of **`internationalCharacter`** and **`codePage`**, and then sends it to the printer. This method does not add a line feed code at the end of the text data. In order to print to the end, add a line feed code at the end of the text data. • .bin, .dat Data is sent to the printer as binary data without conversion. • .htm, .html Data is sent to the printer as html data without conversion.

	`alignment`	Alignment It is valid when the extension of the file specified by `fileName` is .bmp, .jpg, .jpeg, .png, or .txt. See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.
	`dithering`	Dithering It is valid when the extension of the file specified by `fileName` is .bmp, .jpg, .jpeg, or .png. See "4.3.1(4)① Dithering (`Dithering`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

`sendImage` Send image

Prints the specified UIImage-type image.
In syntax (a), image alignment and dithering can be specified.
In syntax (b), image alignment can be specified, and dithering is fixed to disabled.
In syntax (c), image alignment is fixed to left alignment, and dithering is fixed to disabled.

Syntax

```
(a) - (void) sendImage:(UIImage *)image
                    alignment:(PrintAlignment)alignment
                    dithering:(Dithering)dithering;

(b) - (void) sendImage:(UIImage *)image
                    alignment:(PrintAlignment)alignment;

(c) - (void) sendImage:(UIImage *)image;
```

Parameter

`image`	UIImage-type object to be sent to the printer. Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and sent to the printer. Printing is performed in batch after mapping the image file on the memory of the printer.
`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.
`dithering`	Dithering See "4.3.1(4)① Dithering (`Dithering`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

printPDF — Print PDF page

Prints the specified page of the selected PDF file.

Syntax

```
- (void) printPDF:(NSString *)fileName
                  startIndex:(NSInteger)startIndex
                  endIndex:(NSInteger)endIndex
                  rotate:(Rotate)rotate
                  imageScale:(ImageScale)imageScale
                  bottomMargin:(NSInteger)bottomMargin
                  alignment:(PrintAlignment)alignment
                  dithering:(Dithering)dithering;
```

Parameter

`fileName`	Path of the PDF file The file extension for supporting PDF is .pdf. The maximum file size that can be specified is 16 MB(16777216 bytes).
`startIndex`	Start number of the printed page The range is -1, 1 to the number of pages in the PDF file. When -1 is specified for `startIndex`, all pages are printed. When -1 is specified for `startIndex`, the value of `endIndex` is ignored. When 1 or more is specified for `startIndex`, pages from the page number specified in `startIndex` to the page number specified in `endIndex` are printed. When a value more than the value specified for `endIndex` is specified for `startIndex`, an error occurs. When an out-of-range value is specified for `startIndex` or `endIndex`, an error occurs.
`endIndex`	End number of the printed page The range is 1 to 2147483647. When a value more than the number of pages in the PDF file is specified for `endIndex`, pages from the page number specified in `startIndex` to the last page of the PDF file are printed.
`rotate`	Rotation direction of the image See "4.3.1(4)㉑ Image rotation direction (`Rotate`)" for available constants.
`imageScale`	Image scaling See "4.3.1(4)㉒ Image scaling (`ImageScale`) " for available constants. When **SII_PM_IMAGE_SCALE_WIDTH_FIT** is specified for `imageScale`, the image width is converted to the print width of the printer while maintaining the aspect ratio.
`bottomMargin`	Bottom margin (dot) The range is -1, 0 to 2400. When -1 is specified for `bottomMargin`, the image is created and printed while maintaining the bottom margin. When a value between 0 and 2400 is specified for `bottomMargin`, the bottom margin is changed to the specified size. When a value between 0 and 2400 is specified for `bottomMargin`, blank pages are not printed.
`dithering`	Dithering See "4.3.1(4)① Dithering (`Dithering`)" for available constants.

	`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description PDF is converted to printable format for the printer and is sent to the printer.

The color image data is converted to monochrome image by binarization.

When the memory switches MS 1-2 (Mark Mode Selection) of the printer is set to "Enable", form feed of marked paper is performed after printing the PDF page.

Note No guarantee of printing operation when more than 100 pages are printed at a time.

`printLogo` Print logo

Prints the registered logo.

Syntax
```
- (void) printLogo:(NSString *)logoId
                 alignment(PrintAlignment)alignment;
```

Parameter	`logoId`	ID of the logo to be printed (key code) Specify the ID of the registered logo as a character string.
	`alignment`	Alignment See "4.3.1(4)⑨ Alignment (`PrintAlignment`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

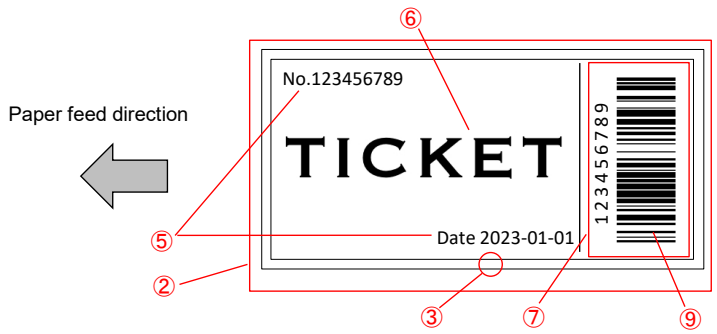
`printSmartLabelImageData` Print label

This method is not supported. When executed, **`SIIPrinterException`** is thrown.

Syntax
```
- (void *) printSmartLabelImageData:
                                (SIISmartLabelManager *)labelManager;
```

③ Dedicated method for page mode

The following methods are dedicated methods to use page mode. An example for the print process in page mode is shown below.



① Start page mode

```
[printerManager enterPageMode];
```

② Specify print area of page mode

```
[printerManager setPageModeArea:0 y:0 width:355 height:576];
```

③ Specify a rectangle and a ruled line

```
[printerManager printPageModeRectangle:0 startY:0 endX:344 endY:575 lineStyle:SII_PM_LINESTYLE_THIN];
[printerManager printPageModeRectangle:7 startY:7 endX:336 endY:567 lineStyle:SII_PM_LINESTYLE_THIN];
[printerManager printPageModeLine:11 startY:404 endX:334 endY:404 lineStyle:SII_PM_LINESTYLE_THIN];
```

④ Specify print direction of page mode

```
[printerManager setPageModeDirection: SII_PM_DIRECTION_ TOP _TO_ BOTTOM];
```

⑤ Specify a character

```
[printerManager printPageModeText:21 startY:47 text:@"NO.123456789"];
[printerManager printPageModeText:212 startY:340 text:@"Date 2023-01-01"];
```

⑥ Specify an image file

```
[NSString *filePath = [[NSBundle mainBundle] pathForResource:@"TicketImage" ofType:@"jpg"];
[printerManager printPageModeImageFile:10 startY:222 fileName:filePath
dithering:SII_PM_DITHERING_DISABLE];
```

⑦ Specify print area of page mode

```
[printerManager setPageModeArea:0 y:404 width:345 height:163];
```

⑧ Specify print direction

```
[printerManager setPageModeDirection:SII_PM_DIRECTION_LEFT_TO_RIGHT];
```

⑨ Specify a barcode

```
[printerManager printPageModeBarcode:20 startY:132 barcodeSymbol:SII_PM_BARCODE_CODE128
data:[@"{B123456789" dataUsingEncoding:NSASCIIStringEncoding]
moduleSize:SII_PM_BARCODE_MODULE_WIDTH_2 moduleHeight:80
hriPosition:SII_PM_HRI_POSITION_ABOVE hriFont:SII_PM_FONT_A];
```

⑩ Print in page mode

```
[printerManager printPageMode:SII_PM_CUT_PARTIAL];
```

⑪ Ends page mode

```
[printerManager exitPageMode];
```

enterPageMode — Start page mode

Starts page mode.

Syntax
```
- (void) enterPageMode;
```

Error
SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException** Class" for details on the error.

Description
This method starts page mode. The dedicated method for page mode and common methods to standard mode and page mode can be used after this method execution.

Executing **exitPageMode** discards the print data kept in the page data buffer and changes the mode to standard mode.
Executing **printPageMode** prints the print data kept in the page data buffer.

exitPageMode — End page mode

Ends page mode and changes the mode to standard mode.

Syntax
```
- (void) exitPageMode;
```

Error
SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException** Class" for details on the error.

Description
Discards the print data kept in the page data buffer and changes the mode to standard mode.

setPageModeArea — Specify print area of page mode

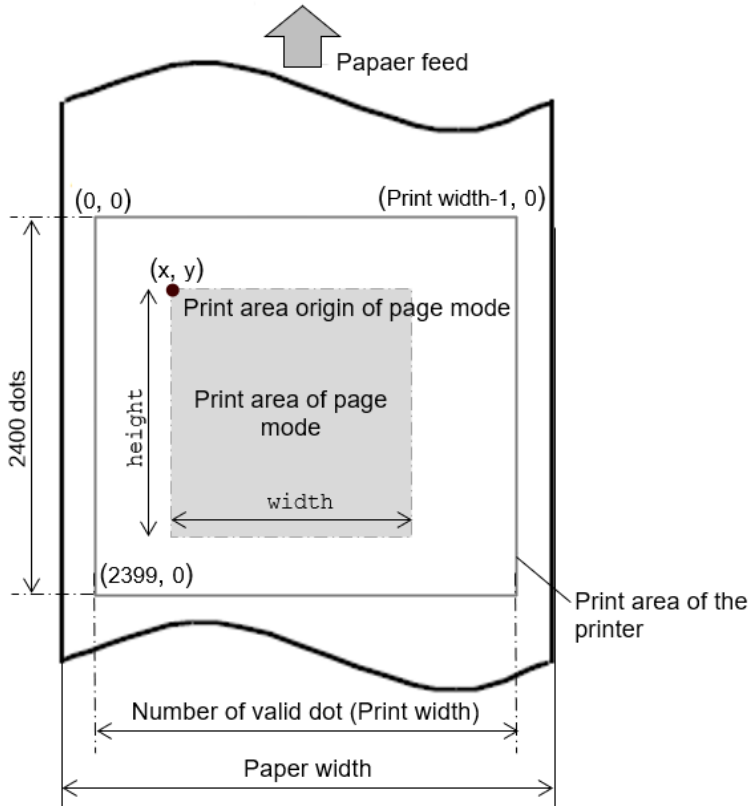
Specifies print area of page mode.

Syntax
```
- (void) setPageModeArea:(NSInteger)x
                       y:(NSInteger)y
                   width:(NSInteger)width
                  height:(NSInteger)height;
```

Parameter

`x`	The horizontal origin (dot) of the print area of page mode The valid range is 0 to (print width - 1). 0 represents the left edge on the print area of the printer.
`y`	The vertical origin (dot) of the print area of page mode The valid range is 0 to 2399. 0 represents the position where paper feed has not been performed.
`width`	The print area width (dot) of page mode The valid range is 1 to (print width - `x`).
`height`	The print area height (dot) of page mode The valid range is 1 to (2400 - `y`).

The relation between the print area of page mode and the print area of the printer is shown in figure below.



Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description

Start page mode by **`enterPageMode`** before executing this method.

Starting page mode by **`enterPageMode`** and executing this method after the dedicated method for page mode is executed, the print area of page mode can be additionally specified. The data that has been mapped is kept.
The data of the dedicated method for page mode is mapped to the print area of page mode added by this method after executing this method.

The print area of page mode is `x` = 0, `y` = 0, `width` = print width, `height` = 2400 after executing **`enterPageMode`**.

Reference

The print width for MP-A40 is the printing width of the memory switch MS 3-1 to 6 (Paper Width) of the printer.

setPageModeDirection — Specify print direction of page mode

Specifies print direction of page mode.

Syntax
```
- (void) setPageModeDirection: (Direction)direction;
```

Parameter

direction	Print direction See "4.3.1(4)㉓ Print direction (Direction)" for available constants.

Error
SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description
Start page mode by **enterPageMode** before executing this method.

The print direction is left to right after executing **enterPageMode**.

setPageModeLineSpacing — Specify line spacing of page mode

Specifies line spacing of page mode.

Syntax
```
- (void) setPageModeLineSpacing:(NSInteger)lineSpacing;
```

Parameter

lineSpacing	Line spacing (dot) of page mode The valid range is 0 to 255.

Error
SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description
Start page mode by **enterPageMode** before executing this method.

The line spacing is 34 dots after executing **enterPageMode**.

printPageMode — Print page mode

Prints the print data kept in page data buffer.

Syntax
```
- (void) printPageMode: (CuttingMethod)CuttingMethod;
```

Parameter

cuttingMethod	Cutting method See "4.3.1(4)⑲ Cutting method (CuttingMethod)" for available constants.

Error
SIIPrinterException is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description
The print data is kept after printing. The print data is discarded at the timing of the following:

- Execute **enterPageMode**
- Execute **disconnect**
- Execute **exitPageMode**

printPageModeText — Send text data of page mode

Maps the text data on the print area of page mode.

Syntax

```
- (void) printPageModeText:(NSInteger)startX
                    startY:(NSInteger)startY
                    text:(NSString *)text;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description

This method encodes the specified text data to printable text data based on **`internationalCharacter`** and **`codePage`**.

Start page mode by **`enterPageMode`** before executing this method.

printPageModeTextEx — Send format specified text data of page mode

Maps the format specified text data on the print area of page mode.

Syntax

```
- (void) printPageModeTextEx:(NSInteger)startX
                      startY:(NSInteger)startY
                      text:(NSString *)text
                      bold:(CharacterBold)bold
                      underline:(CharacterUnderline)underline
                      reverse:(CharacterReverse)reverse
                      font:(CharacterFont)font
                      scale:(CharacterScale)scale;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
`bold`	Bold print See "4.3.1(4)③ Bold print (`CharacterBold`)" for available constants.
`underline`	Underline See "4.3.1(4)④ Underline (`CharacterUnderline`)" for available constants.

	`reverse`	Reverse print See "4.3.1(4)⑤ Reverse print (`CharacterReverse`)" for available constants.
	`font`	Font See "4.3.1(4)⑦ Character font (`CharacterFont`)" for available constants.
	`scale`	Character scale See "4.3.1(4)⑧ Character scale (`CharacterScale`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description This method encodes the specified text data to printable text data based on **`internationalCharacter`** and **`codePage`**.

Start page mode by **`enterPageMode`** before executing this method.

`printPageModeBarcode` Print barcode of page mode

Maps the barcode on the print area of page mode.
In syntax (a), barcode data can be specified as a character string.
In syntax (b), barcode data can be specified as a character string, and the N:W ratio can also be specified.
In syntax (c), barcode data can be specified as a byte array.
The syntax (d) is not supported.

Syntax

```
(a) - (void) printPageModeBarcode:(NSInteger)startX
                  startY:(NSInteger)startY
                  barcodeSymbol:(BarcodeSymbol)barcodeSymbol
                  text:(NSString *)text
                  moduleSize:(ModuleSize)moduleSize
                  moduleHeight:(NSInteger)moduleHeight
                  hriPosition:(HriPosition)hriPosition
                  hriFont:(CharacterFont)hriFont;

(b) - (void) printPageModeBarcode:(NSInteger)startX
                 startY:(NSInteger)startY
                 barcodeSymbol:(BarcodeSymbol)barcodeSymbol
                 text:(NSString *)text
                 moduleSize:(ModuleSize)moduleSize
                 moduleHeight:(NSInteger)moduleHeight
                 hriPosition:(HriPosition)hriPosition
                 hriFont:(CharacterFont)hriFont
                 nwRatio:(NwRatio)nwRatio;

(c) - (void) printPageModeBarcode:(NSInteger)startX
                  startY:(NSInteger)startY
                  barcodeSymbol:(BarcodeSymbol)barcodeSymbol
                  data:(NSData*)data
                  moduleSize:(ModuleSize)moduleSize
                  moduleHeight:(NSInteger)moduleHeight
                  hriPosition:(HriPosition)hriPosition
                  hriFont:(CharacterFont)hriFont;
```

```
(d) - (void) printPageModeBarcode:(NSInteger)startX
                  startY:(NSInteger)startY
                  barcodeSymbol:(BarcodeSymbol)barcodeSymbol
                  barcodeSymbol:(BarcodeSymbol)barcodeSymbol
                  text:(NSString *)text
                  moduleSize:(ModuleSize)moduleSize;
```

Parameter

`startX` The horizontal reference point (dot) from the starting point
The valid range is 0 to 2399.

`startY` The vertical reference point (dot) from the starting point
The valid range is 0 to 2399.

`barcodeSymbol` BarcodeSymbol
See "4.3.1(4)⑩ Barcode symbol (`BarcodeSymbol`)" for available constants and correspondent syntax.

`text(data)` Barcode data to send to the printer
The input conditions for barcode are as follows.

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
UPC-A	11 to 12 characters	'0' to '9'	
UPC-E	11 to 12 characters	'0' to '9'	
EAN13 JAN13	12 to 13 characters '	'0' to '9'	
EAN8 JAN8	7 to 8 characters	'0' to '9'	
CODE39	1 to 150 characters	'0' to '9' 'A' to 'Z' ' ', '$,', '%', '+', '-', '.', '/'	Start code and stop code ('*') are automatically added.
CODE93	1 to 150 bytes	(0x00 to 0x2E)	Input data with 0x2F or more at the end.
CODE128	2 to 150 bytes	(0x00 to 0x66)	When inputting the start code (0x67 to 0x69) of the CODE128 code set. Input data with 0x67 or more at the end.
		(0x00 to 0x7F)	When starting with a CODE128 special code start code ("{A", "{B", "{C").
ITF	2 to 150 characters (However, an even number)	'0' to '9'	
CODABAR	1 to 150 characters	'0' to '9' '$', '+', '-', '.', '/', ':'	It is needed to specify one of 'A' to 'D' at the beginning and end.
EAN13 add-on JAN13 add-on	Add-on 2: 14 to 15 characters Add-on 5: 17 to 18 characters	'0' to '9'	
Customer Bar Code_JP	-	-	Not supported.

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
GS1 Databar Omni-directional	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Truncated	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Limited	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Expanded	2 to 255 characters	' ' to "" '%' to '?' 'A' to 'Z' '_' 'a' to 'z' '{'	

`moduleSize` Barcode width
See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

`moduleHeight` Barcode height (dot)

- When `barcodeSymbol` is set to the following, the valid range is 1 to 255.
  SII_PM_BARCODE_UPC_A
  SII_PM_BARCODE_UPC_E
  SII_PM_BARCODE_EAN13
  SII_PM_BARCODE_JAN13
  SII_PM_BARCODE_EAN8
  SII_PM_BARCODE_JAN8
  SII_PM_BARCODE_CODE39
  SII_PM_BARCODE_CODE93
  SII_PM_BARCODE_CODE128
  SII_PM_BARCODE_ITF
  SII_PM_BARCODE_CODABAR
  SII_PM_BARCODE_EAN13_ADDON
  SII_PM_BARCODE_JAN13_ADDON

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

`barcodeSymbol`		
	`moduleSize`	**Valid Range**
SII_PM_BARCODE_GS1_OMNI_DIRECTIONAL		
	SII_PM_BARCODE_MODULE_WIDTH_2	66 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	99 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	132 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	165 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	198 to 255

barcodeSymbol		
	moduleSize	**Valid Range**
SII_PM_BARCODE_GS1_TRUNCATED		
	SII_PM_BARCODE_MODULE_WIDTH_2	26 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	39 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	52 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	65 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	78 to 255
SII_PM_BARCODE_GS1_LIMITED		
	SII_PM_BARCODE_MODULE_WIDTH_2	20 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	30 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	40 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	50 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	60 to 255
SII_PM_BARCODE_GS1_EXPANDED		
	SII_PM_BARCODE_MODULE_WIDTH_2	68 to 255
	SII_PM_BARCODE_MODULE_WIDTH_3	102 to 255
	SII_PM_BARCODE_MODULE_WIDTH_4	136 to 255
	SII_PM_BARCODE_MODULE_WIDTH_5	170 to 255
	SII_PM_BARCODE_MODULE_WIDTH_6	204 to 255

`hriPosition` HRI character print position
See "4.3.1(4)⑫ HRI character print position (`HriPosition`)" for available constants.

`hriFont` HRI character font
See "4.3.1(4)⑦ Character font (`CharacterFont`)" for available constants.

`nwRatio` N:W ratio
See "4.3.1(4)⑬ N:W ratio (`NwRatio`)" for available constants.
Depending on specified `nwRatio` and `moduleSize`, the wide element width is set as shown in the following table.

moduleSize	nwRatio		
	SII_PM_ NWRATIO_1TO2	**SII_PM_ NWRATIO_1TO2_5**	**SII_PM_ NWRATIO_1TO3**
SII_PM_BARCODE_MODULE_WIDTH_2	0.500 mm (4 dots)	0.625 mm (5 dots)	0.750 mm (6 dots)
SII_PM_BARCODE_MODULE_WIDTH_3	0.750 mm (6 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)
SII_PM_BARCODE_MODULE_WIDTH_4	1.000 mm (8 dots)	1.250 mm (10 dots)	1.500 mm (12 dots)
SII_PM_BARCODE_MODULE_WIDTH_5	1.250 mm (10 dots)	1.625 mm (13 dots)	1.875 mm (15 dots)
SII_PM_BARCODE_MODULE_WIDTH_6	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description Start page mode by **`enterPageMode`** before executing this method.

Note Map the print data of the barcode not to overlap the other print data.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

`printPageModePDF417` Print PDF417 of page mode

Maps PDF417 on the print area of page mode.
In syntax (a), PDF417 symbol can be specified.
In syntax (b), PDF417 symbol is fixed to standard PDF417.

Syntax

(a) –
```
(void) printPageModePDF417:(NSInteger)startX
                    startY:(NSInteger)startY
                    text:(NSString *)text
                    errorCorrection:(ErrorCorrection)errorCorrection
                    row:(NSInteger)row
                    column:(NSInteger)column
                    moduleSize:(ModuleSize)moduleSize
                    moduleHeight:(NSInteger)moduleHeight
                    pdf417Symbol:(Pdf417Symbol)pdf417Symbol;
```

(b) –
```
(void) printPageModePDF417:(NSInteger)startX
                    startY:(NSInteger)startY
                    text:(NSString *)text
                    errorCorrection:(ErrorCorrection)errorCorrection
                    row:(NSInteger)row
                    column:(NSInteger)column
                    moduleSize:(ModuleSize)moduleSize
                    moduleHeight:(NSInteger)moduleHeight;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data
`errorCorrection`	Error correction level See "4.3.1(4)⑭ Error correction level (`ErrorCorrection`)" for available constants.
`row`	The number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
`column`	The number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
`moduleSize`	Nominal fine element width See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

	`moduleHeight`	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
	`pdf417Symbol`	Symbol of PDF417 See "4.3.1(4)⑮ PDF417 symbol (`Pdf417Symbol`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description Start page mode by **`enterPageMode`** before executing this method.

Note Map the print data of the barcode not to overlap the other print data.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

`printPageModeQRcode` Print QR Code of page mode

Maps QR Code on the print area of page mode.
In syntax (a), QR Code Model can be specified.
In syntax (b), QR Code Model is fixed to QR Code Model 2.

Syntax

```
(a) - (void) printPageModeQRcode:(NSInteger)startX
                  startY:(NSInteger)startY
                  text:(NSString *)text
                  errorCorrection:(ErrorCorrection)errorCorrection
                  moduleSize:(ModuleSize)moduleSize
                  model:(QrModel)model;

(b) - (void) printPageModeQRcode:(NSInteger)startX
                  startY:(NSInteger)startY
                  text:(NSString *)text
                  errorCorrection:(ErrorCorrection)errorCorrection
                  moduleSize:(ModuleSize)moduleSize;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data The version for either syntax (a) or (b) is automatically set depending on the number of data specified on `text`.
`errorCorrection`	Error correction level See "4.3.1(4)⑭ Error correction level (`ErrorCorrection`)" for available constants.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
`model`	QR Code Model See "4.3.1(4)⑯ QR Code Model (`QrModel`)" for available constants.

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description Start page mode by **enterPageMode** before executing this method.

Note Map the print data of the barcode not to overlap the other print data.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeDataMatrix — Print Data Matrix of page mode

Maps Data Matrix on the print area of page mode.

Syntax
```
- (void) printPageModeDataMatrix:(NSInteger)startX
                          startY:(NSInteger)startY
                          text:(NSString *)text
                          dataMatrixModule:(DataMatrixModule)dataMatrixModule
                          moduleSize:(ModuleSize)moduleSize;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data
`dataMatrixModule`	The number of Data Matrix modules See "4.3.1(4)⑰ Data Matrix module (`DataMatrixModule`)" for available constants.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

Error **SIIPrinterException** is thrown when an error occurs while this method is being called.
See "**4.3.4 SIIPrinterException Class**" for details on the error.

Description Start page mode by **enterPageMode** before executing this method.

Note Map the print data of the barcode not to overlap the other print data.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeMaxiCode — Print MaxiCode of page mode

Maps MaxiCode on the print area of page mode.

Syntax
```
- (void) printPageModeMaxiCode:(NSInteger)startX
                        startY:(NSInteger)startY
                        text:(NSString *)text
                        maxiCodeMode:(MaxiCodeMode)maxiCodeMode;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.

	`text`	Barcode data • When **`maxiCodeMode`** is **SII_PM_MAXI_CODE_2** Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data. • When **`maxiCodeMode`** is **SII_PM_MAXI_CODE_3** Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.
	`maxiCodeMode`	MaxiCode Mode See "4.3.1(4)⑱ MaxiCode Mode (`MaxiCodeMode`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description Start page mode by **`enterPageMode`** before executing this method.

Note Map the print data of the barcode not to overlap the other print data.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

`printPageModeGS1DataBarStacked` Print GS1 Databar Stacked of page mode

Maps GS1 Databar Stacked on the print area of page mode.

Syntax
```
- (void) printPageModeGS1DataBarStacked:(NSInteger)startX
                                startY:(NSInteger)startY
                                text:(NSString *)text
                                moduleSize:(ModuleSize)moduleSize;
```

Parameter	`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`text`	Barcode data Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
	`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description Start page mode by **`enterPageMode`** before executing this method.

Note Map the print data of the barcode not to overlap the other print data.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional of page mode

Maps GS1 Databar Stacked Omni-directional on the print area of page mode.

Syntax

```
- (void) printPageModeGS1DataBarStackedOmnidirectional:
                                                (NSInteger)startX
                startY:(NSInteger)startY
                text:(NSString *)text
                moduleHeight:(ModuleHeight)moduleHeight
                moduleSize:(ModuleSize)moduleSize;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
`moduleHeight`	Barcode module height (the number of the modules) The valid range is 33 to 255.
`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description

Start page mode by **`enterPageMode`** before executing this method.

Note

Map the print data of the barcode not to overlap the other print data.

Reference

See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked of page mode

Maps GS1 Databar Expanded Stacked on the print area of page mode.

Syntax

```
- (void) printPageModeGS1DataBarExpandedStacked:(NSInteger)startX
                startY:(NSInteger)startY
                text:(NSString *)text
                column:(NSInteger)column
                moduleSize:(ModuleSize)moduleSize;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.

	`text`	Barcode data Enter any number of characters using the following: ' ', '!', '"', '%', '&', '', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.
	`column`	The number of columns Specifies the number of the columns in 1 line. The valid range is the even number from 2 to 20.
	`moduleSize`	Module size See "4.3.1(4)⑪ Module size (`ModuleSize`)" for available constants.
Error	**`SIIPrinterException`** is thrown when an error occurs while this method is being called. See "**4.3.4 `SIIPrinterException` Class**" for details on the error.	
Description	Start page mode by **`enterPageMode`** before executing this method.	
Note	Map the print data of the barcode not to overlap the other print data.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

`printPageModeAztecCode` Print Aztec Code of page mode

This method is not supported. When executed, **`SIIPrinterException`** is thrown.

Syntax

```
- (void) printPageModeAztecCode:(NSInteger)startX
                         startY:(NSInteger)startY
                           text:(NSString *)text
                          layer:(NSInteger)layer
                errorCorrection:(NSInteger)errorCorrection
                     moduleSize:(ModuleSize)moduleSize
                    aztecSymbol:(AztecSymbol)aztecSymbol;
```

`sendPageModeBinary` Send binary data of page mode

Maps binary data on the print area of page mode.

Syntax

```
- (void) sendPageModeBinary:(NSData*)data;
```

Parameter	`binary`	Binary data Data size that can be specified at 1 time is 3 MB (3145728 bytes).
Error	**`SIIPrinterException`** is thrown when an error occurs while this method is being called. See "**4.3.4 `SIIPrinterException` Class**" for details on the error.	
Description	Start page mode by **`enterPageMode`** before executing this method. This method sends the specified binary data to the printer without conversion. By sending printer commands as binary data with this method, printer functions which are not supported in the library become available.	
Note	This method may execute unexpected performance depending on the data to send. Please ensure the performance with your actual device in advance.	

printPageModeImageFile — Draw Image file of page mode

Maps the image file on the print area of page mode.

Syntax

```
- (void) printPageModeImageFile:(NSInteger)startX
                         startY:(NSInteger)startY
                         fileName:(NSString *)fileName
                         dithering:(Dithering)dithering;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`fileName`	Path of the image file The maximum file size that can be specified is 16 MB (16777216 bytes). The image files that can be sent are .bmp, .jpg, .jpeg, .png. Colored image file is converted to monochrome image by binarization and registered.
`dithering`	Dithering See "4.3.1(4)① Dithering (`Dithering`)" for available constants.

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description

Start page mode by **`enterPageMode`** before executing this method.

printPageModeRectangle — Draw rectangle image of page mode

Maps the rectangle image on the print area of page mode.

Syntax

```
- (void) printPageModeRectangle:(NSInteger)startX
                         startY:(NSInteger)startY
                         endX:(NSInteger)endX
                         endY:(NSInteger)endY
                         lineStyle:(LineStyle)lineStyle;
```

Parameter

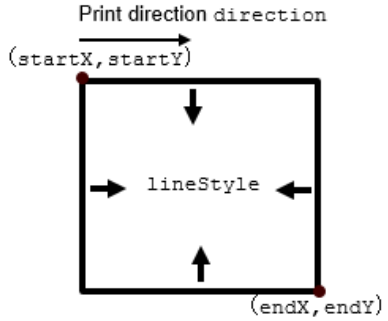
`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`endX`	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
`endY`	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
`lineStyle`	Line style See "4.3.1(4)㉔ Line style (`LineStyle`)" for available constants.

Error

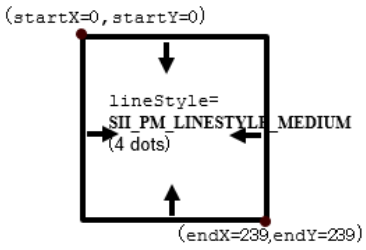
`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description Start page mode by **`enterPageMode`** before executing this method.

The rectangle is mapped to `direction` of **`setPageModeDirection`** as shown in the figure below.



The example of the parameter setting to the image is shown below.
Example: Draw a square with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point.

Image	Parameter
	`startX` 0 `startY` 0 `endX` 239 `endY` 239 `lineStyle` **SII_PM_LINESTYLE_MEDIUM**

`printPageModeLine` Print ruled line of page mode

Maps the ruled line on the print area of page mode.

Syntax

```
- (void) printPageModeLine:(NSInteger)startX
                    startY:(NSInteger)startY
                      endX:(NSInteger)endX
                      endY:(NSInteger)endY
                 lineStyle:(LineStyle)lineStyle;
```

Parameter

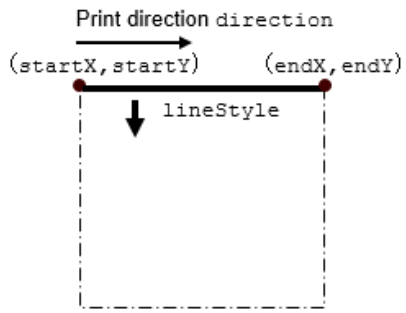
`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`endX`	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
`endY`	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
`lineStyle`	Line style See "4.3.1(4)㉔ Line style (`LineStyle`)" for available constants.

Error **`SIIPrinterException`** is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

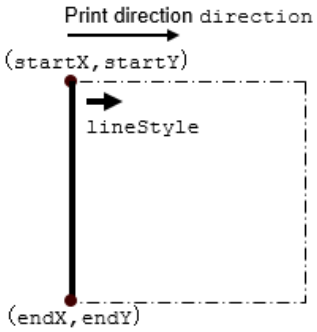
Description Start page mode by **`enterPageMode`** before executing this method.

A diagonal stroke cannot be drawn by this method.

The ruled line is mapped to the `direction` of **`setPageModeDirection`** as shown in the figure below.

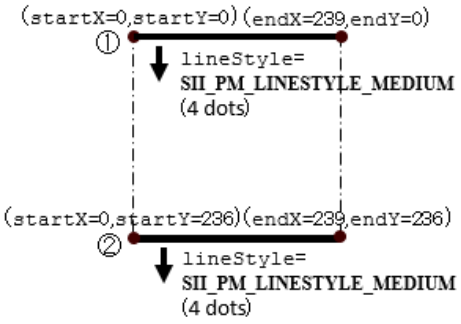


Mapping direction of horizontal ruled line

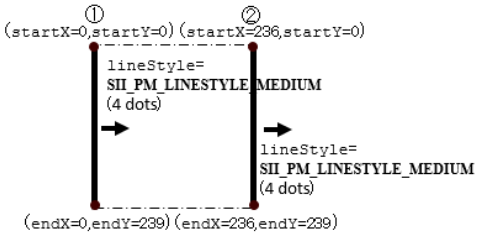


Mapping direction of vertical ruled line

The setting example of the parameter to the image is shown below.
Example: Draw a horizontal ruled line of a square with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point.

Image	Parameter
	①
	`startX` 0
	`startY` 0
	`endX` 239
	`endY` 0
	`lineStyle` **SII_PM_LINESTYLE_MEDIUM**
	②
	`startX` 0
	`startY` 236
	`endX` 239
	`endY` 236
	`lineStyle` **SII_PM_LINESTYLE_MEDIUM**

Example: Draw a vertical ruled line of a square with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point.

Image	Parameter
	① startX 0 startY 0 endX 0 endY 239 lineStyle **SII_PM_LINESTYLE_MEDIUM** ② startX 236 startY 0 endX 236 endY 239 lineStyle **SII_PM_LINESTYLE_MEDIUM**

printPageModeLogo — Print logo of page mode

Maps the registered logo on the print area of page mode.

Syntax

```
- (void) printPageModeLogo:(NSInteger)startX
                    startY:(NSInteger)startY
                    logoId:(NSString *)logoId;
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`logoId`	Logo ID to print (key code) Specify the ID of the registered logo as a character string.

Error

`SIIPrinterException` is thrown when an error occurs while this method is being called.
See "**4.3.4 `SIIPrinterException` Class**" for details on the error.

Description

Start page mode by **`enterPageMode`** before executing this method.

(6) Common property detail to standard mode and page mode

sendTimeout — Get/Set send timeout period

Gets or sets the timeout period in sending data.

Syntax `@property NSInteger **sendTimeout;**`

Valid range 100 to 300000 (millisecond: ms)
When the value is specified less than 100, the period is set to 100 ms.
When the value is specified more than 300000, the period is set to 300000 ms.

Default 10000

Description This property can get or set the timeout period regardless of whether **isConnect** is YES or NO.
The set timeout period becomes effective at the next data sending.

receiveTimeout — Get/Set receive timeout period

Gets or sets the timeout period in receiving data.

Syntax `@property NSInteger **receiveTimeout;**`

Valid range 100 to 300000 (millisecond: ms)
When the value is specified less than 100, the period is set to 100 ms.
When the value is specified more than 300000, the period is set to 300000 ms.

Default 10000

Description This property can get or set the timeout period regardless of whether **isConnect** is YES or NO.
The set timeout period becomes effective at the next data receiving.

internationalCharacter — Get/Set international character set

Gets or sets the value of international character set.

Syntax `@property NSInteger **internationalCharacter;**`

Description See "4.3.1(3)⑤ International character set" for configurable constants. When an invalid value is specified, it is ignored.
When this property is not set, the international character set is as follows depending on the language setting of iOS device.

When the language setting of iOS device is Japanese: **SII_PM_COUNTRY_JAPAN**
When the language setting of iOS device is other than Japanese:
SII_PM_COUNTRY_USA

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

Character codes with the varying print result depending on the configuration of the international character:
0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E

codePage — Get/Set codepage

Gets or sets the value of codepage.

Syntax
```
@property NSInteger codePage;
```

Description
See "4.3.1(3)⑥ Codepage" for configurable constants. When an invalid value is specified, it is ignored.
When this property is not set, the codepage is as follows depending on the language setting of iOS device.

When the language setting of iOS device is Japanese:
SII_PM_CODE_PAGE_KATAKANA
When the language setting of iOS device is other than Japanese:
SII_PM_CODE_PAGE_1252

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

printerModel — Get printer model

Gets the value of the connecting printer model.

Syntax
```
@property(readonly) NSInteger printerModel;
```

Default
-1

Return value
See "4.3.1(3)① Printer model" for available constants.
When **isConnect** is NO, -1 is returned.

portType — Get connecting port type

Gets the value of the port type used for connection with the printer.

Syntax
```
@property(readonly) NSInteger portType;
```

Default
-1

Return value
See "4.3.1(3)② Port type" for available constants.
When **isConnect** is NO, -1 is returned.

isConnect — Verify connection state with printer

Verifies connection state with the printer.

Syntax
```
@property(readonly) BOOL isConnect;
```

Return value
YES Connected to the printer
NO Not connected to the printer

Description
This property retains the **connect** state as a BOOL value.
When **connect** succeeds, this property is YES. After **connect**, when **disconnect** succeeds, this property becomes NO.

socketKeepingTime — Get/Set socket keeping time

This property is not supported and should not be used.

Syntax `@property NSInteger **socketKeepingTime;**`

delegate — Register delegate

Registers a delegate object that receives notifications from the printer.

Syntax `@property(weak, nonatmic) id<SIIPrinterManagerDelegate> **delegate;**`

Description Specify an object conforming to **`SIIPrinterManagerDelegate`** protocol.
When this property is executed with the delegate object registered, the already registered delegate object becomes disabled, and a new delegate object is registered.

When nil is specified for this property, the notification of the printer status is stopped.

PrintSmartLabelMode — Get/Set paper when printing label file

This property is not supported and should not be used.

Syntax `@property NSInteger **printSmartLabelMode;**`

4.3.2 `SIIPrinterInfo` Class

This class stores the printer information found by printer searching method. It gets the printer model name, MAC address, and IP address from the found printer information.

Do not use this class because it is not supported.

4.3.3 `SIIPrinterIdentifier` Class

`PrinterIdentifier` class stores the printer ID of the printer retrieved by **`getPrinterResponse`**.

(1) Method List

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

Name	Description
`SIIPrinterIdentifier`	Constructor

(2) Property List

Properties provided by **`PrinterIdentifier`** class are shown in the following table.

Name	Access	Description
`printerModelID`	R	Get printer model ID
`typeID`	R	Get type ID
`romVersionID`	R	Get ROM version ID
`mainFirmwareVersion`	R	Get firmware version (main)
`manufacturer`	R	Get manufacturer
`modelName`	R	Get model name
`serialNumber`	R	Get serial number
`bootFirmwareVersion`	R	Get firmware version (boot)
`bootFirmwareChecksum`	R	Get firmware checksum (boot)
`mainFirmwareChecksum`	R	Get firmware checksum (main)
`firmwareChecksum`	R	Get firmware checksum (main + boot)

(3) Method Details

`SIIPrinterIdentifier` Constructor

This method stores the printer ID of the printer obtained via **`getPrinterResponse`**.

Syntax **`SIIPrinterIdentifier`**

(4) Property Details

printerModelID — Get printer model ID

This property gets the printer model ID from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property UInt8 **printerModelID;**`

typeID — Get type ID

This property gets the type ID from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property UInt8 **typeID;**`

romVersionID — Get ROM version ID

This property gets the ROM version ID from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property UInt8 **romVersionID;**`

mainFirmwareVersion — Get firmware version (main)

This property gets the firmware version (main) as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property NSString *__mainFirmwareVersion;__`

manufacturer — Get manufacturer

This property gets the manufacturer name as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property NSString *__manufacturer;__`

modelName — Get model name

This property gets the model name as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property NSString *__modelName;__`

serialNumber — Get serial number

This property gets the serial number as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax `@property NSString *__serialNumber;__`

bootFirmwareVersion — Get firmware version (boot)

This property gets the firmware version (boot) as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax

```
@property NSString *bootFirmwareVersion;
```

bootFirmwareChecksum — Get firmware checksum (boot)

This property gets the firmware checksum (boot) from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax

```
@property NSData *bootFirmwareChecksum;
```

mainFirmwareChecksum — Get firmware checksum (main)

This property gets the firmware checksum (main) from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax

```
@property NSData *mainFirmwareChecksum;
```

firmwareChecksum — Get firmware checksum (main + boot)

This property gets the firmware checksum (main + boot) from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax

```
@property NSData *firmwareChecksum;
```

4.3.4 `SIIPrinterException` Class

(1) Method List

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

Name	Description
`SIIPrinterException`	Constructor

(2) Property List

Properties provided by **`SIIPrinterException`** class are shown in the following table.

Name	Access	Description
`errorCode`	R	Get error code
`errorMessage`	R	Get error message

(3) Constant List

① Error code

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

Constant Name	Description	Value
SII_PM_ERROR_ACCESS_DENIED	Failed to get the handle.*1	-1
	An unavailable port was specified.	
	An unsupported method was specified.	
SII_PM_ERROR_SHARING_VIOLATION	An already opened port was specified.	-11
SII_PM_ERROR_PORT_NOT_OPENED	The port is not open.	-12
SII_PM_ERROR_DEVICE_NOT_CONNECTED	There is a problem with the Bluetooth connection between the iOS device and the printer.	-21
SII_PM_ERROR_OFFLINE	Disconnected state or the printer is offline.	-22
SII_PM_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
SII_PM_ERROR_DATA_SIZE_ZERO	0-byte data was specified.	-101
SII_PM_ERROR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.	-102
SII_PM_ERROR_ENCODE_FAILED	An error occurred in encoding text data.*1	-111
SII_PM_ERROR_TIMEOUT	Send timeout occurred.	-201
	Receive timeout occurred.	
SII_PM_ERROR_FILE_NOT_FOUND	The specified file is not found.	-301
SII_PM_ERROR_FILE_USED	The specified file is in use by another process.	-302
SII_PM_ERROR_FILE_INVALID	The specified file is invalid.	-303
SII_PM_ERROR_LOW_MEMORY	Memory shortage occurred when loading image file.	-311
SII_PM_ERROR_OVER_MAX_IMAGE	Either or both of width and height of image file exceeds the number of printable maximum dots.	-312
SII_PM_ERROR_LOGO_NOT_DEFINED	The logo is not registered.	-313
SII_PM_ERROR_LOW_USER_AREA	Remaining user area is insufficient.	-401
SII_PM_ERROR_LOW_EXTERNAL_RAM	Remaining RAM capacity is insufficient.	-402
SII_PM_ERROR_INVALID_NO	The specified value for the logo ID or the style sheet number is invalid.	-501
SII_PM_ERROR_OVER_STYLE_NUM	The number of styles registered in the specified file exceeds the rated value (64).	-502
SII_PM_ERROR_PAGE_MODE_SPECIFIED	Page mode is specified.	-511
SII_PM_ERROR_PAGE_MODE_NOT_SPECIFIED	Page mode is not specified.	-512
SII_PM_ERROR_INVALID_PARAM	The specified parameter is invalid.	-9999

*1: Abnormal processing might have occurred.

(4) Method Details

SIIPrinterException — Constructor

This is the exception class that is thrown when API of **SIIPrinterManager** class is called.

Syntax **SIIPrinterException**

(5) Property Details

errorCode — Get error code

Gets the error code of the thrown exception.

Syntax `@property NSInteger` **errorCode;**

Return value See "4.3.4(3) Constant List".

errorMessage — Get error message

Gets the error message of the thrown exception.

Syntax `@property NSString` ***errorMessage;**

Description A character string that supplements the contents of **errorCode** can be retrieved.

4.3.5 `SIIPrinterManagerDelegate` Protocol

(1) Method List

Methods provided by **`SIIPrinterManagerDelegate`** protocol are shown in the following table.

Name	Description
`didStatusChange`	Notify printer status

(NOTE) MP-A40 does not support the APIs of the barcode scanner.

(2) Method Details

`didStatusChange` Notify printer status

Notifies changes in the printer status.

Syntax
```
- (void) didStatusChange:(SIIPrinterManager *)printerManager
                  status:(NSInteger)status;
```

Parameter

`printerManager`	Calling `SIIPrinterManager` object
`status`	Printer status

Description

This method is called the latest status at the following timing.
・When **`connect`** is executed.
・When the printer status is changed.

This method is called when **`isConnect`** is YES.

The notification of the printer status is stopped by **`disconnect`**.
The notification of the printer status is stopped by setting nil to **`delegate`**.

When communication with the printer is disconnected, this method notifies 0x80000000. After disconnection from the printer, the library attempts to resume communication with the printer until **`disconnect`** is executed. When communication with the printer becomes possible, this method notifies the latest printer status.
See **`getStatus`** for description of the printer status.

Do not execute the APIs of `SIIPrinterManager` within this method.

4.3.6 `SIISmartLabelManager` Class

`SIISmartLabelManager` class provides the function to covert the label file (*.sl, *.slex) created using SII Layout Editor into the printable data from the printer.

Do not use this class because it is not supported.

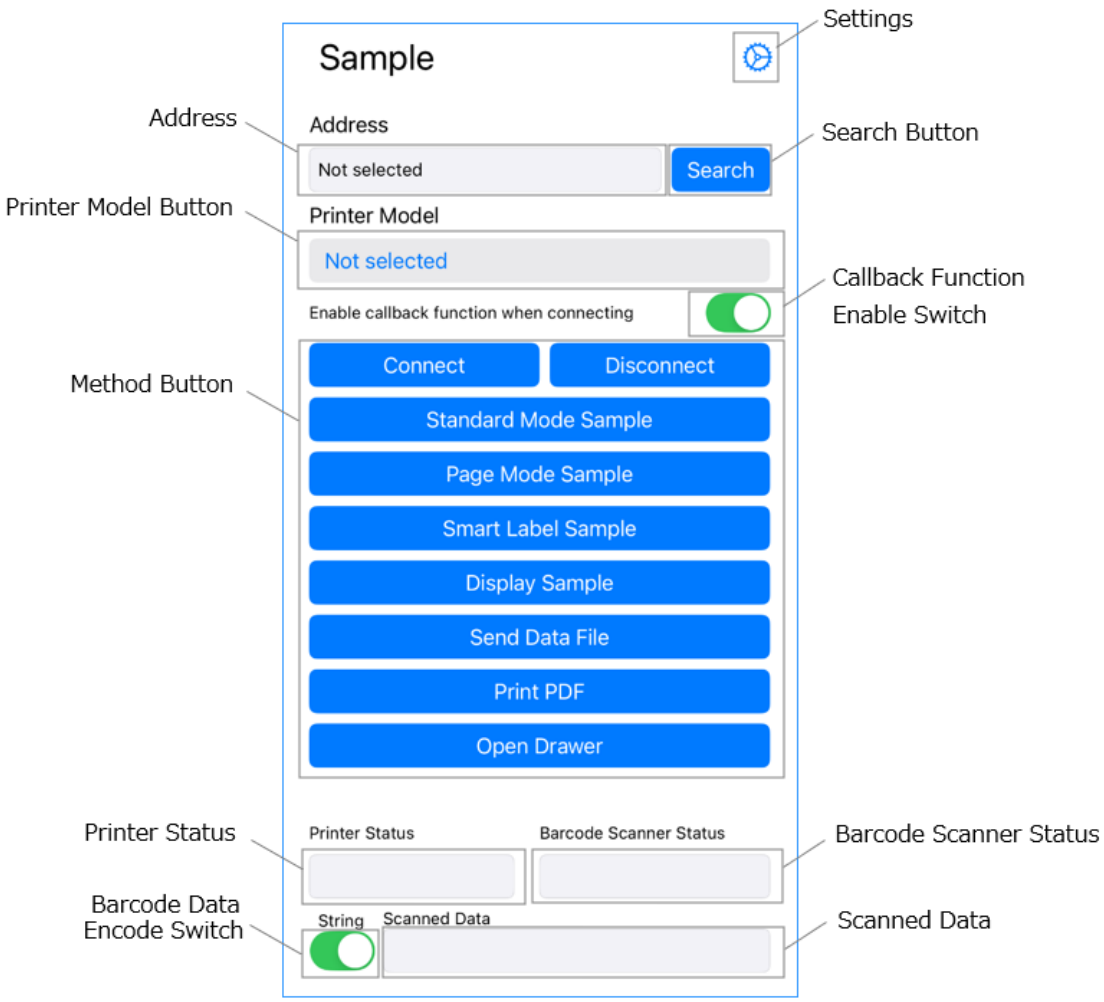
Chapter 5
Sample Program

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

5.1 Screen Layout

SII print class library includes SiiLibSample, a sample program in Xcode project format. This section describes the screen of SiiLibSample.

5.1.1 Main Screen

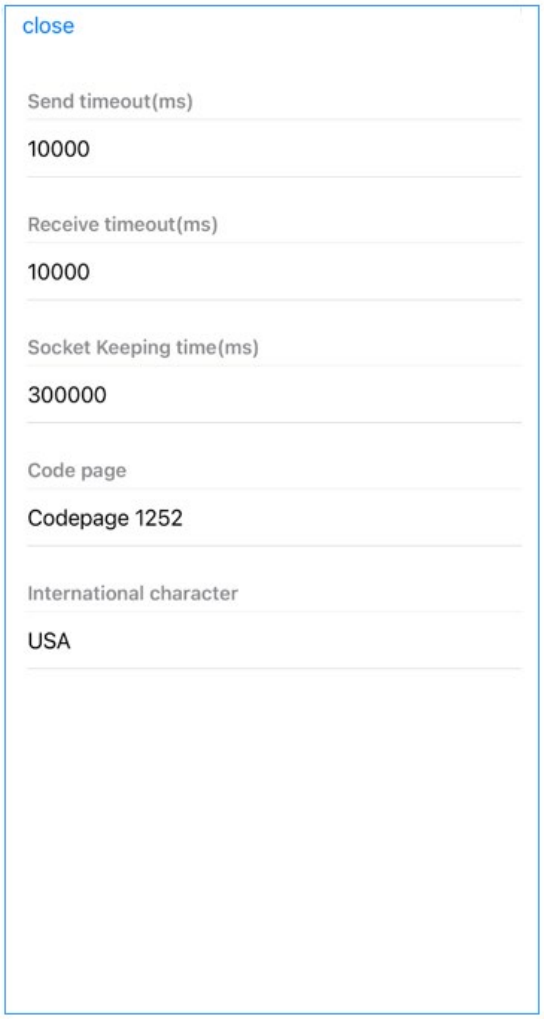


Item	Description
Settings	Tapping the [Settings] button opens the function setting screen. In order to go back to the main screen, tap [close] on the top left of the screen.
Address	Displays the information about the selected printer.
Printer Model Button	Specifies the printer model. When tapping [Printer Model Button], a list of printer models is displayed. By selecting from the list, the printer model can be entered. When the printer is selected from the printer search screen, the printer model is automatically displayed.
Search Button	Starts searching for printers. Transits to the printer search screen. A list of the searched printers is displayed. The printer is selected by tapping the searched printer and returns to the main screen.
Callback Function Enable Switch	Select whether to enable the callback function when connecting to the printer. On : Starts the callback function when connecting. Off : The callback function does not respond.
Method Button*1	In addition to the method buttons for executing **`connect`** and **`disconnect`**, the sample by the combination of some methods can be printed and checked for the operation of peripheral devices.
Printer Status	Displays the printer status. When [Callback Function Enable Switch] is On, the latest status is displayed.
Barcode Scanner Status	Displays the connection status of the barcode scanner. MP-A40 does not support the barcode scanner.
Barcode Data Encode Switch	Selects the conversion of barcode data read by the barcode scanner. MP-A40 does not support the barcode scanner.
Scanned Data	Displays the barcode data read by the barcode scanner. MP-A40 does not support the barcode scanner.

*1: Supported functions vary by model. Only supported functions can be operated.

5.1.2 **[Settings] Screen**

Various setting functions are displayed in [Settings].



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

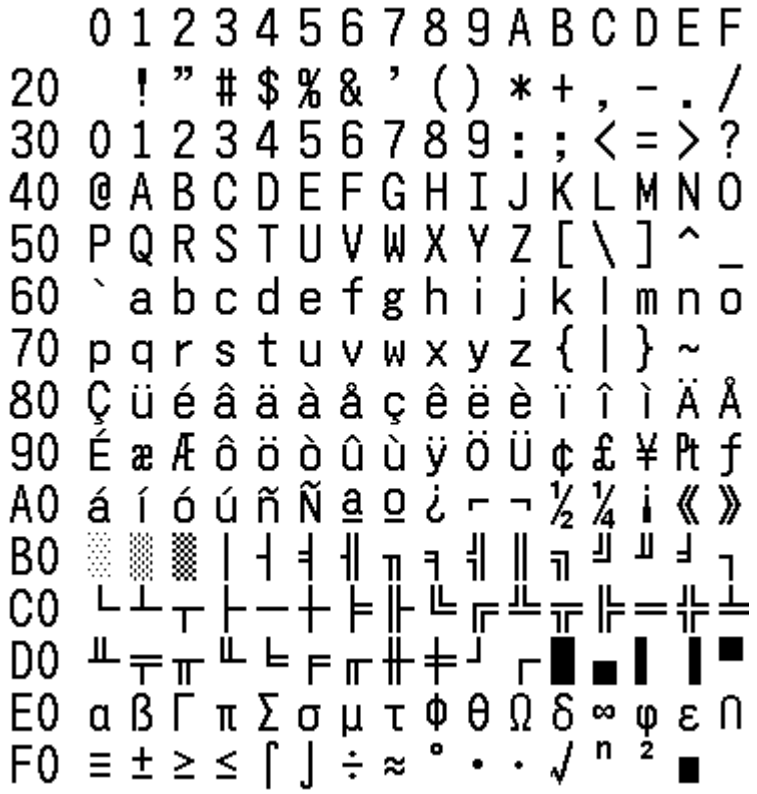


Figure A-1 SII_PM_CODE_PAGE_437 (USA, Standard Europe)

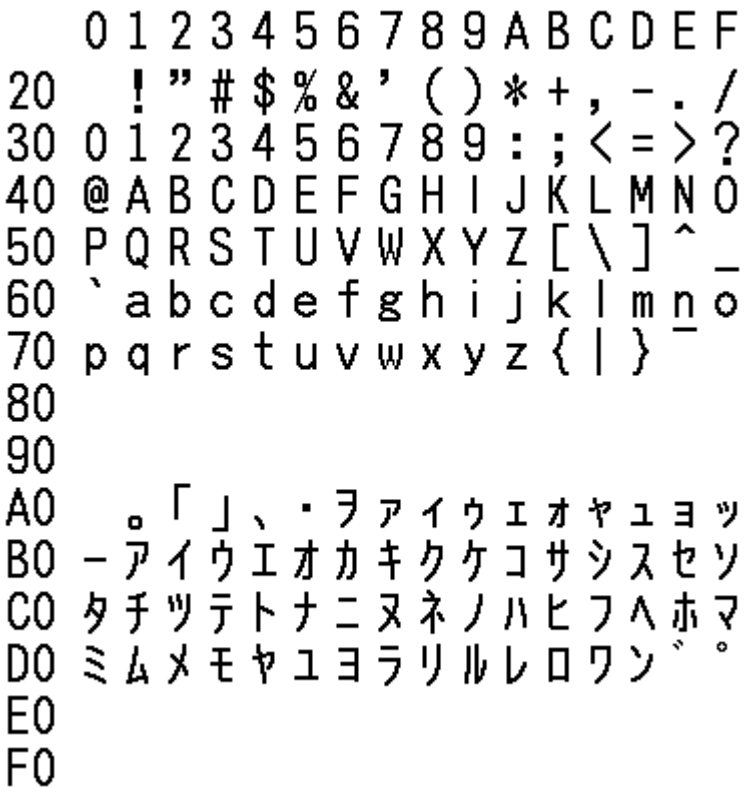


Figure A-2 SII_PM_CODE_PAGE_KATAKANA

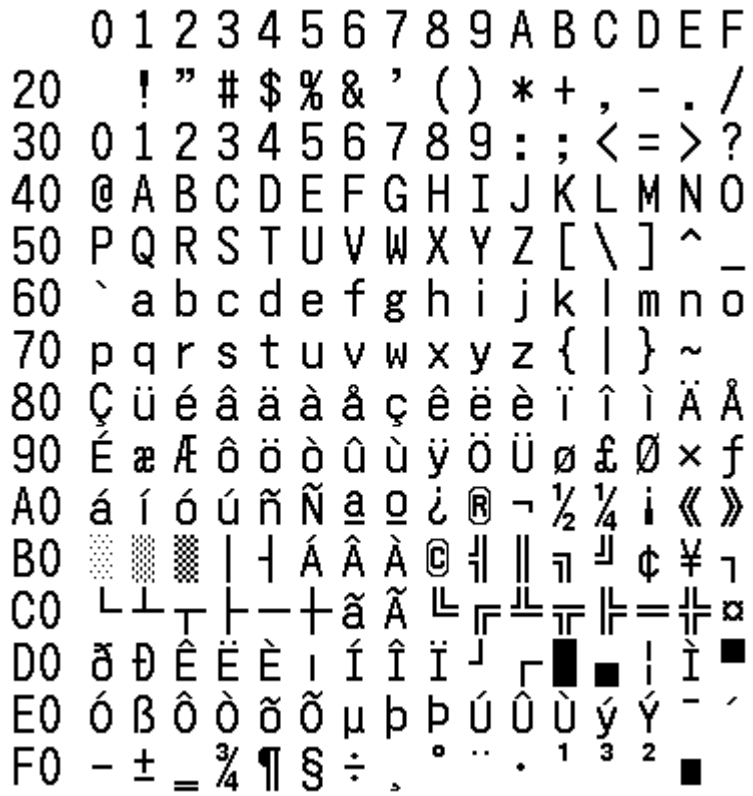


Figure A-3 SII_PM_CODE_PAGE_850 (Multilingual)

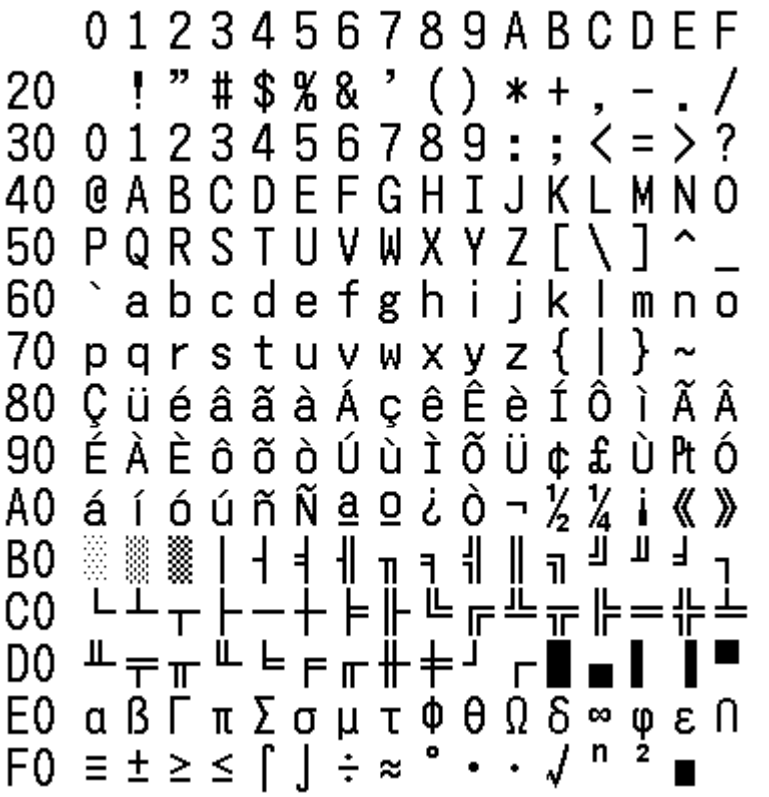


Figure A-4 SII_PM_CODE_PAGE_860 (Portuguese)

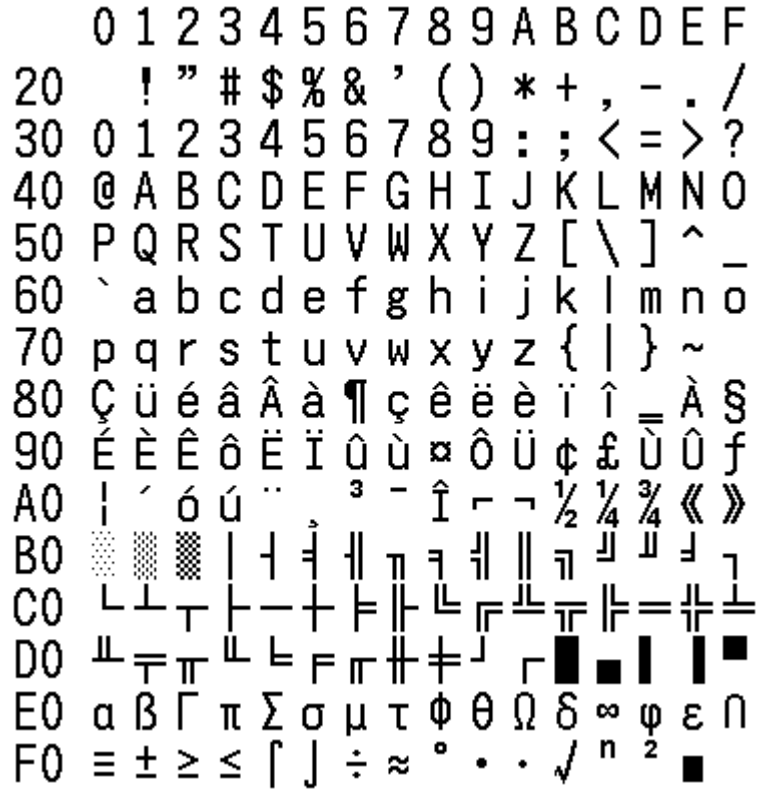


Figure A-5 SII_PM_CODE_PAGE_863 (Canadian-French)

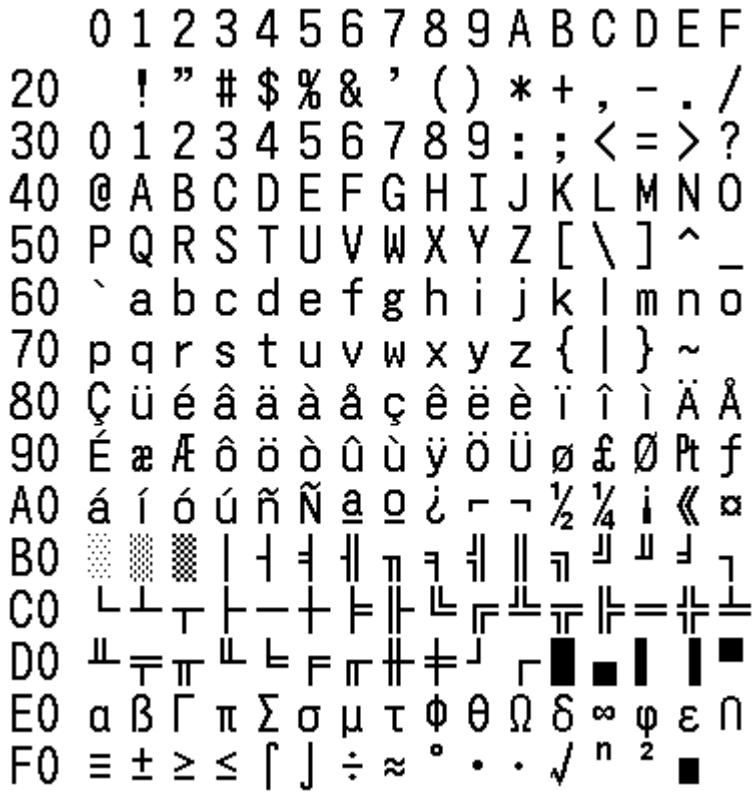


Figure A-6 SII_PM_CODE_PAGE_865 (Nordic)

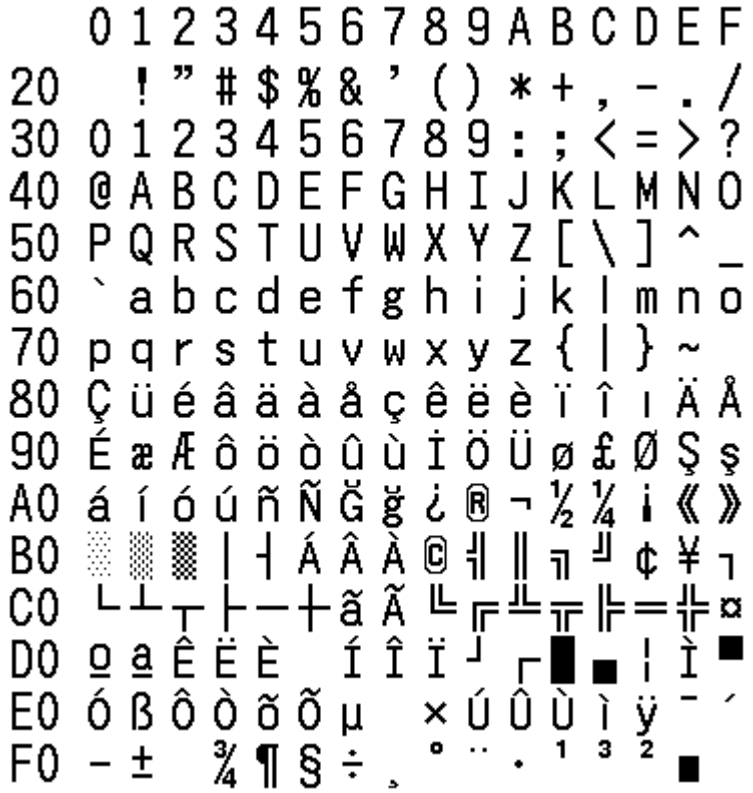


Figure A-7 SII_PM_CODE_PAGE_857 (Turkish)

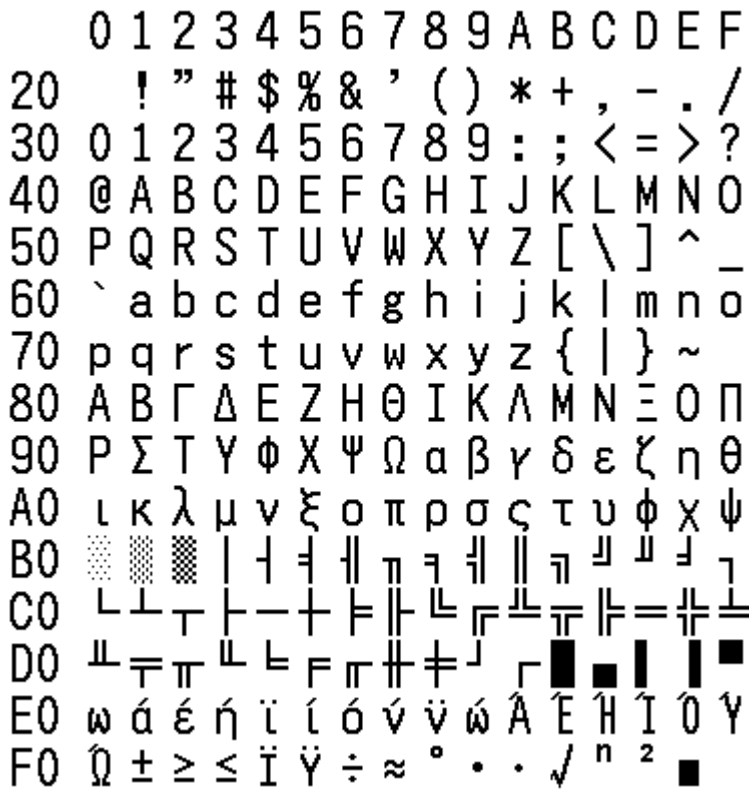


Figure A-8 SII_PM_CODE_PAGE_737 (Greek)

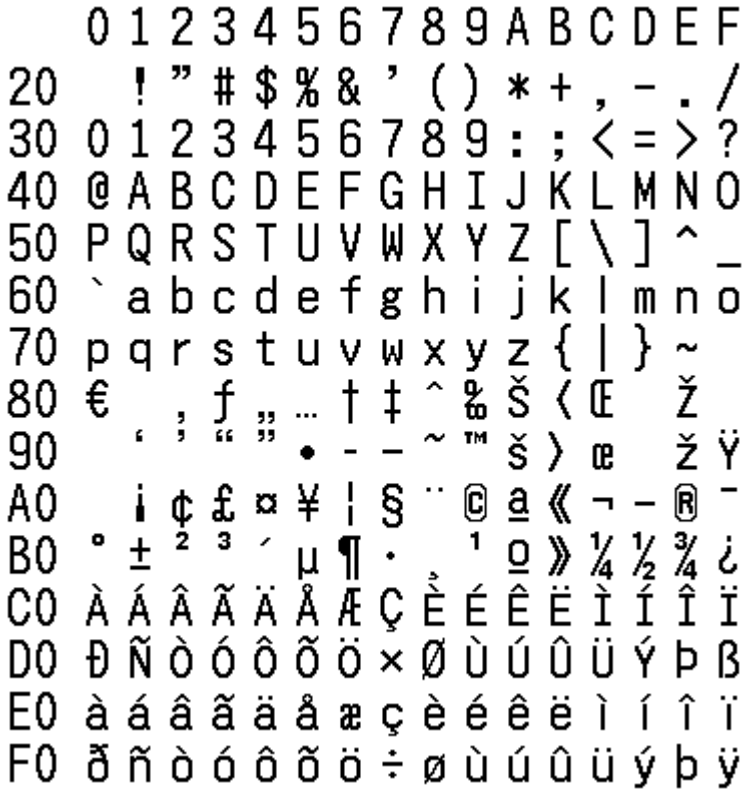


Figure A-9 SII_PM_CODE_PAGE_1252 (Latin)

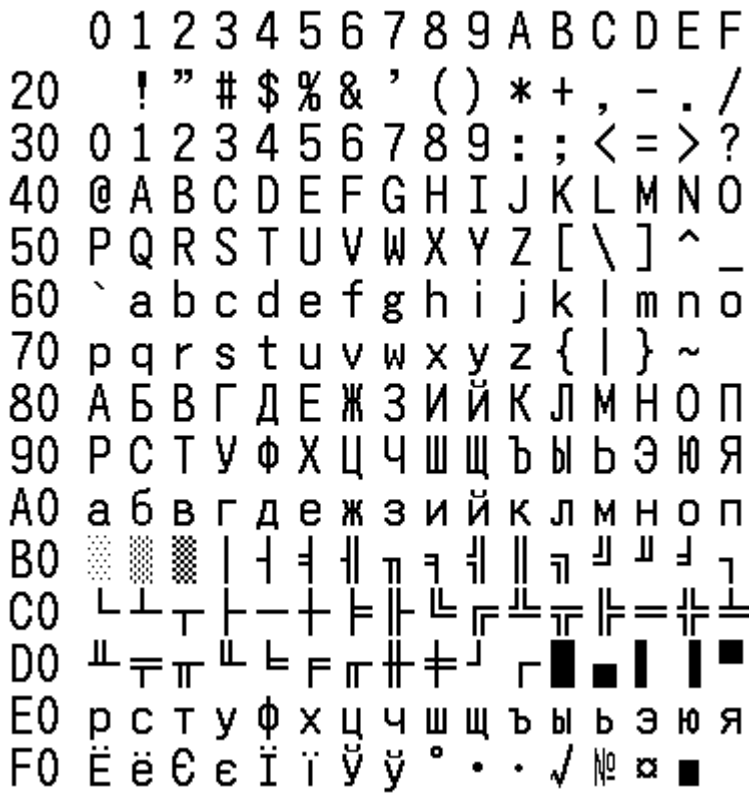


Figure A-10 SII_PM_CODE_PAGE_866 (Russian)

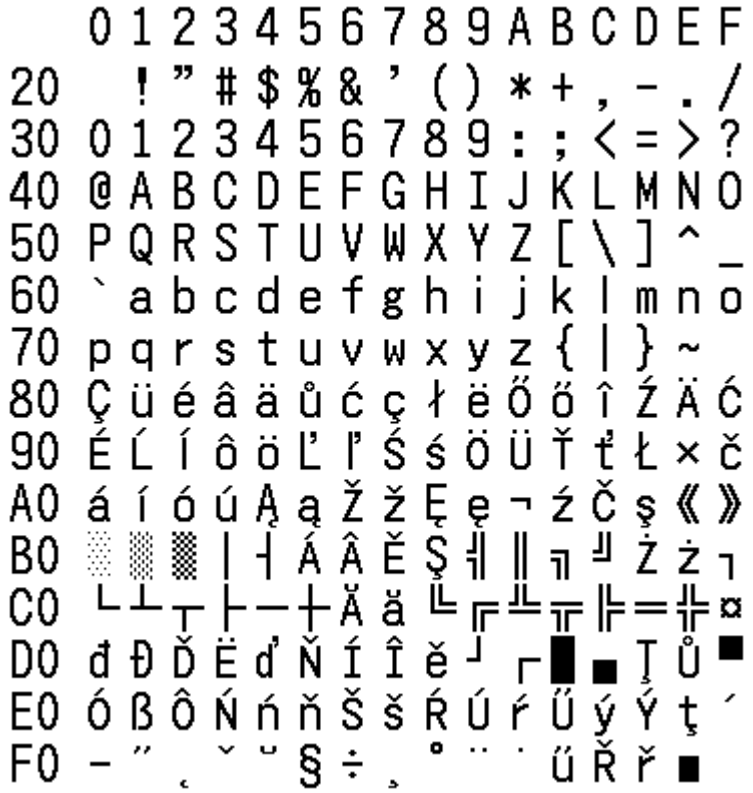


Figure A-11 SII_PM_CODE_PAGE_852 (Eastern Europe)

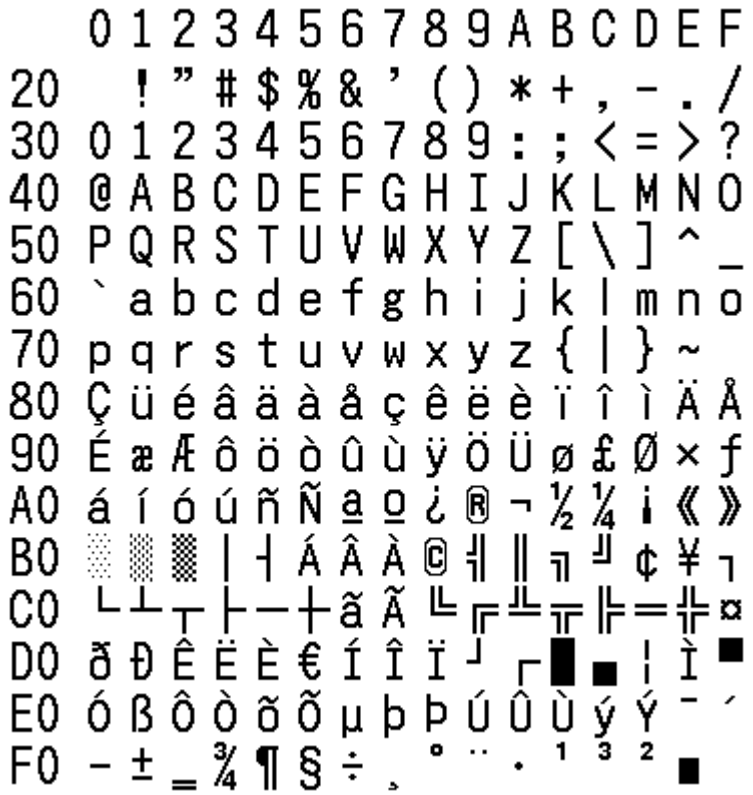


Figure A-12 SII_PM_CODE_PAGE_858 (Euro)



Figure A-13 SII_PM_CODE_PAGE_855 (Cyrillic)



Figure A-14 SII_PM_CODE_PAGE_864 (Arabic)



Figure A-15 SII_PM_CODE_PAGE_1250 (Central European)



Figure A-16 SII_PM_CODE_PAGE_1251 (Cyrillic)

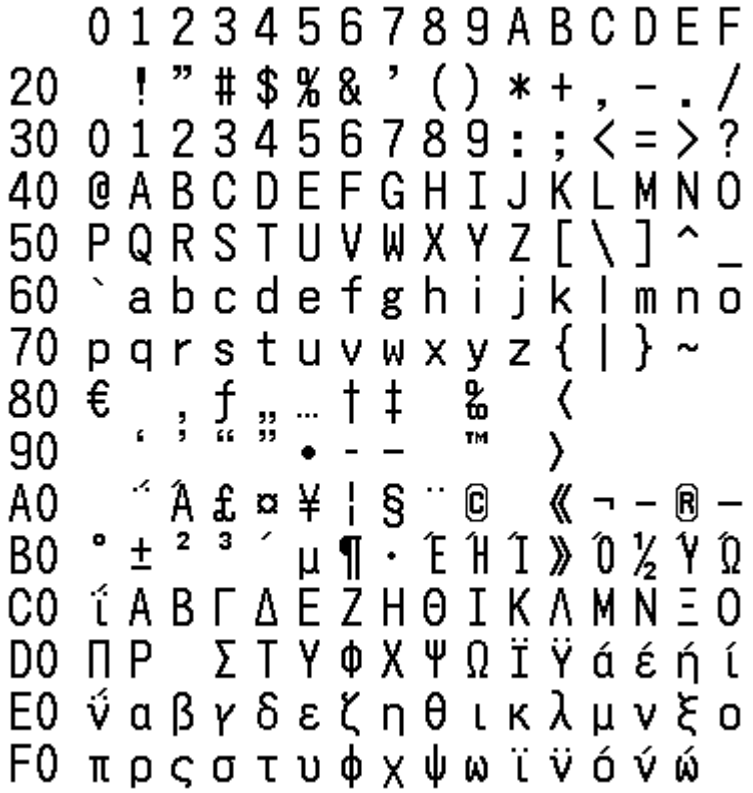


Figure A-17 SII_PM_CODE_PAGE_1253 (Greek)



Figure A-18 SII_PM_CODE_PAGE_1254 (Turkish)

A.2 International Character Set

Print results of the specific character codes vary depending on the setting of the international character set. The following table shows the specific character codes and their print results.

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

Figure A-19 International Character Set

Appendix B
Barcode Size List

B.1 Barcode Size List

B.1.1 `printBarcode, printPageModeBarcode`



(1) Height of the barcode image

hriFont	hriPosition	Length from Top of Barcode to Reference Point	Height of Barcode Image
SII_PM_FONT_A	SII_PM_HRI_NONE	moduleHeight	moduleHeight
	SII_PM_HRI_POSITION_ABOVE	moduleHeight + 32	moduleHeight + 32
	SII_PM_HRI_POSITION_BELOW	moduleHeight	moduleHeight + 32
	SII_PM_HRI_POSITION_ABOVE_BELOW	moduleHeight + 64	moduleHeight + 64
SII_PM_FONT_B	SII_PM_HRI_NONE	moduleHeight	moduleHeight
	SII_PM_HRI_POSITION_ABOVE	moduleHeight + 24	moduleHeight + 24
	SII_PM_HRI_POSITION_BELOW	moduleHeight	moduleHeight + 24
	SII_PM_HRI_POSITION_ABOVE_BELOW	moduleHeight + 48	moduleHeight + 48

(2) Width of the barcode image

barcodeSymbol	moduleSize	Width of Barcode Image
SII_PM_BARCODE_UPC_A	**SII_PM_BARCODE_MODULE_WIDTH_2**	226
	SII_PM_BARCODE_MODULE_WIDTH_3	339
	SII_PM_BARCODE_MODULE_WIDTH_4	452
	SII_PM_BARCODE_MODULE_WIDTH_5	565
	SII_PM_BARCODE_MODULE_WIDTH_6	678
SII_PM_BARCODE_UPC_E	**SII_PM_BARCODE_MODULE_WIDTH_2**	130
	SII_PM_BARCODE_MODULE_WIDTH_3	195
	SII_PM_BARCODE_MODULE_WIDTH_4	260
	SII_PM_BARCODE_MODULE_WIDTH_5	325
	SII_PM_BARCODE_MODULE_WIDTH_6	390
SII_PM_BARCODE_EAN13	**SII_PM_BARCODE_MODULE_WIDTH_2**	226
	SII_PM_BARCODE_MODULE_WIDTH_3	339
	SII_PM_BARCODE_MODULE_WIDTH_4	452
	SII_PM_BARCODE_MODULE_WIDTH_5	565
	SII_PM_BARCODE_MODULE_WIDTH_6	678
SII_PM_BARCODE_JAN13	**SII_PM_BARCODE_MODULE_WIDTH_2**	226
	SII_PM_BARCODE_MODULE_WIDTH_3	339
	SII_PM_BARCODE_MODULE_WIDTH_4	452
	SII_PM_BARCODE_MODULE_WIDTH_5	565
	SII_PM_BARCODE_MODULE_WIDTH_6	678
SII_PM_BARCODE_EAN8	**SII_PM_BARCODE_MODULE_WIDTH_2**	162
	SII_PM_BARCODE_MODULE_WIDTH_3	243
	SII_PM_BARCODE_MODULE_WIDTH_4	324
	SII_PM_BARCODE_MODULE_WIDTH_5	405
	SII_PM_BARCODE_MODULE_WIDTH_6	486
SII_PM_BARCODE_JAN8	**SII_PM_BARCODE_MODULE_WIDTH_2**	162
	SII_PM_BARCODE_MODULE_WIDTH_3	243
	SII_PM_BARCODE_MODULE_WIDTH_4	324
	SII_PM_BARCODE_MODULE_WIDTH_5	405
	SII_PM_BARCODE_MODULE_WIDTH_6	486
SII_PM_BARCODE_CODE93	**SII_PM_BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 96
	SII_PM_BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 144
	SII_PM_BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 192
	SII_PM_BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 240
	SII_PM_BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 288
SII_PM_BARCODE_CODE128	**SII_PM_BARCODE_MODULE_WIDTH_2**	22 × number of barcode data + 66
	SII_PM_BARCODE_MODULE_WIDTH_3	33 × number of barcode data + 99
	SII_PM_BARCODE_MODULE_WIDTH_4	44 × number of barcode data + 132
	SII_PM_BARCODE_MODULE_WIDTH_5	55 × number of barcode data + 165
	SII_PM_BARCODE_MODULE_WIDTH_6	66 × number of barcode data + 198

barcodeSymbol	moduleSize	**Width of Barcode Image**
SII_PM_BARCODE_GS1_OMNI_DIRECTIONAL	**SII_PM_BARCODE_MODULE_WIDTH_2**	192
	SII_PM_BARCODE_MODULE_WIDTH_3	288
	SII_PM_BARCODE_MODULE_WIDTH_4	384
	SII_PM_BARCODE_MODULE_WIDTH_5	480
	SII_PM_BARCODE_MODULE_WIDTH_6	576
SII_PM_BARCODE_GS1_TRUNCATED	**SII_PM_BARCODE_MODULE_WIDTH_2**	192
	SII_PM_BARCODE_MODULE_WIDTH_3	288
	SII_PM_BARCODE_MODULE_WIDTH_4	384
	SII_PM_BARCODE_MODULE_WIDTH_5	480
	SII_PM_BARCODE_MODULE_WIDTH_6	576
SII_PM_BARCODE_GS1_LIMITED	**SII_PM_BARCODE_MODULE_WIDTH_2**	158
	SII_PM_BARCODE_MODULE_WIDTH_3	237
	SII_PM_BARCODE_MODULE_WIDTH_4	316
	SII_PM_BARCODE_MODULE_WIDTH_5	395
	SII_PM_BARCODE_MODULE_WIDTH_6	474
SII_PM_BARCODE_GS1_EXPANDED[*1]	**SII_PM_BARCODE_MODULE_WIDTH_2**	number of barcode module × 2
	SII_PM_BARCODE_MODULE_WIDTH_3	number of barcode module × 3
	SII_PM_BARCODE_MODULE_WIDTH_4	number of barcode module × 4
	SII_PM_BARCODE_MODULE_WIDTH_5	number of barcode module × 5
	SII_PM_BARCODE_MODULE_WIDTH_6	number of barcode module × 6

*1: The number of barcode module is determined by the barcode data to be specified.

barcodeSymbol	nwRatio	moduleSize	Width of Barcode Image
SII_PM_BARCODE_CODE39	**SII_PM_NWRATIO_1TO2**	**SII_PM_BARCODE_MODULE_WIDTH_2**	26 × number of barcode data + 90
		SII_PM_BARCODE_MODULE_WIDTH_3	39 × number of barcode data + 135
		SII_PM_BARCODE_MODULE_WIDTH_4	52 × number of barcode data + 180
		SII_PM_BARCODE_MODULE_WIDTH_5	65 × number of barcode data + 225
		SII_PM_BARCODE_MODULE_WIDTH_6	78 × number of barcode data + 270
	SII_PM_NWRATIO_1TO2_5	**SII_PM_BARCODE_MODULE_WIDTH_2**	29 × number of barcode data + 96
		SII_PM_BARCODE_MODULE_WIDTH_3	45 × number of barcode data + 147
		SII_PM_BARCODE_MODULE_WIDTH_4	58 × number of barcode data + 192
		SII_PM_BARCODE_MODULE_WIDTH_5	74 × number of barcode data + 243
		SII_PM_BARCODE_MODULE_WIDTH_6	87 × number of barcode data + 288
	SII_PM_NWRATIO_1TO3	**SII_PM_BARCODE_MODULE_WIDTH_2**	32 × number of barcode data + 102
		SII_PM_BARCODE_MODULE_WIDTH_3	48 × number of barcode data + 153
		SII_PM_BARCODE_MODULE_WIDTH_4	64 × number of barcode data + 204
		SII_PM_BARCODE_MODULE_WIDTH_5	80 × number of barcode data + 255
		SII_PM_BARCODE_MODULE_WIDTH_6	96 × number of barcode data + 306
SII_PM_BARCODE_ITF	**SII_PM_NWRATIO_1TO2**	**SII_PM_BARCODE_MODULE_WIDTH_2**	14 × number of barcode data + 56
		SII_PM_BARCODE_MODULE_WIDTH_3	21 × number of barcode data + 84
		SII_PM_BARCODE_MODULE_WIDTH_4	28 × number of barcode data + 112
		SII_PM_BARCODE_MODULE_WIDTH_5	35 × number of barcode data + 140
		SII_PM_BARCODE_MODULE_WIDTH_6	42 × number of barcode data + 168
	SII_PM_NWRATIO_1TO2_5	**SII_PM_BARCODE_MODULE_WIDTH_2**	16 × number of barcode data + 57
		SII_PM_BARCODE_MODULE_WIDTH_3	25 × number of barcode data + 86
		SII_PM_BARCODE_MODULE_WIDTH_4	32 × number of barcode data + 114

barcodeSymbol	nwRatio	moduleSize	Width of Barcode Image
SII_PM_BARCODE_ITF	**SII_PM_NWRATIO_1TO2_5**	**SII_PM_BARCODE_MODULE_WIDTH_5**	41 × number of barcode data + 143
		SII_PM_BARCODE_MODULE_WIDTH_6	48 × number of barcode data + 171
	SII_PM_NWRATIO_1TO3	**SII_PM_BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 58
		SII_PM_BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 87
		SII_PM_BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 116
		SII_PM_BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 145
		SII_PM_BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 174
SII_PM_BARCODE_CODABAR*1	**SII_PM_NWRATIO_1TO2**	**SII_PM_BARCODE_MODULE_WIDTH_2**	20 × number of data + 2 × (2 + number of wide data) + 38
		SII_PM_BARCODE_MODULE_WIDTH_3	30 × number of data + 3 × (2 + number of wide data) + 57
		SII_PM_BARCODE_MODULE_WIDTH_4	40 × number of data + 4 × (2 + number of wide data) + 76
		SII_PM_BARCODE_MODULE_WIDTH_5	50 × number of data + 5 × (2 + number of wide data) + 95
		SII_PM_BARCODE_MODULE_WIDTH_6	60 × number of data + 6 × (2 + number of wide data) + 114
	SII_PM_NWRATIO_1TO2_5	**SII_PM_BARCODE_MODULE_WIDTH_2**	22 × number of data + 3 × (2 + number of wide data) + 38
		SII_PM_BARCODE_MODULE_WIDTH_3	34 × number of data + 5 × (2 + number of wide data) + 57
		SII_PM_BARCODE_MODULE_WIDTH_4	44 × number of data + 6 × (2 + number of wide data) + 76
		SII_PM_BARCODE_MODULE_WIDTH_5	56 × number of data + 8 × (2 + number of wide data) + 95
		SII_PM_BARCODE_MODULE_WIDTH_6	66 × number of data + 9 × (2 + number of wide data) + 114
	SII_PM_NWRATIO_1TO3	**SII_PM_BARCODE_MODULE_WIDTH_2**	24 × number of data + 4 × (2 + number of wide data) + 38
		SII_PM_BARCODE_MODULE_WIDTH_3	36 × number of data + 6 × (2 + number of wide data) + 57
		SII_PM_BARCODE_MODULE_WIDTH_4	48 × number of data + 8 × (2 + number of wide data) + 76
		SII_PM_BARCODE_MODULE_WIDTH_5	60 × number of data + 10 × (2 + number of wide data) + 95
		SII_PM_BARCODE_MODULE_WIDTH_6	72 × number of data + 12 × (2 + number of wide data) + 114

*1: The number of data is the number of all characters except for the start and stop characters.
The wide data is the number of ": / . +".

barcodeSymbol	Number of Data	moduleSize	Width of Barcode Image
SII_PM_BARCODE_EAN13_ADDON	14 or 15	**SII_PM_BARCODE_MODULE_WIDTH_2**	276
		SII_PM_BARCODE_MODULE_WIDTH_3	414
		SII_PM_BARCODE_MODULE_WIDTH_4	552
		SII_PM_BARCODE_MODULE_WIDTH_5	690
		SII_PM_BARCODE_MODULE_WIDTH_6	828
	17 or 18	**SII_PM_BARCODE_MODULE_WIDTH_2**	330
		SII_PM_BARCODE_MODULE_WIDTH_3	495
		SII_PM_BARCODE_MODULE_WIDTH_4	660
		SII_PM_BARCODE_MODULE_WIDTH_5	825
		SII_PM_BARCODE_MODULE_WIDTH_6	990
SII_PM_BARCODE_JAN13_ADDON	14 or 15	**SII_PM_BARCODE_MODULE_WIDTH_2**	276
		SII_PM_BARCODE_MODULE_WIDTH_3	414
		SII_PM_BARCODE_MODULE_WIDTH_4	552
		SII_PM_BARCODE_MODULE_WIDTH_5	690
		SII_PM_BARCODE_MODULE_WIDTH_6	828
	17 or 18	**SII_PM_BARCODE_MODULE_WIDTH_2**	330
		SII_PM_BARCODE_MODULE_WIDTH_3	495
		SII_PM_BARCODE_MODULE_WIDTH_4	660
		SII_PM_BARCODE_MODULE_WIDTH_5	825
		SII_PM_BARCODE_MODULE_WIDTH_6	990

B.1.2 printPDF417, printPageModePDF417



(1) Height of the barcode image

moduleSize	**Height of the barcode image*1**
SII_PM_PDF417_MODULE_WIDTH_2	moduleHeight×row*2+8
SII_PM_PDF417_MODULE_WIDTH_3	moduleHeight×row*2+12
SII_PM_PDF417_MODULE_WIDTH_4	moduleHeight×row*2+16
SII_PM_PDF417_MODULE_WIDTH_5	moduleHeight×row*2+20
SII_PM_PDF417_MODULE_WIDTH_6	moduleHeight×row*2+24
SII_PM_PDF417_MODULE_WIDTH_7	moduleHeight×row*2+28
SII_PM_PDF417_MODULE_WIDTH_8	moduleHeight×row*2+32

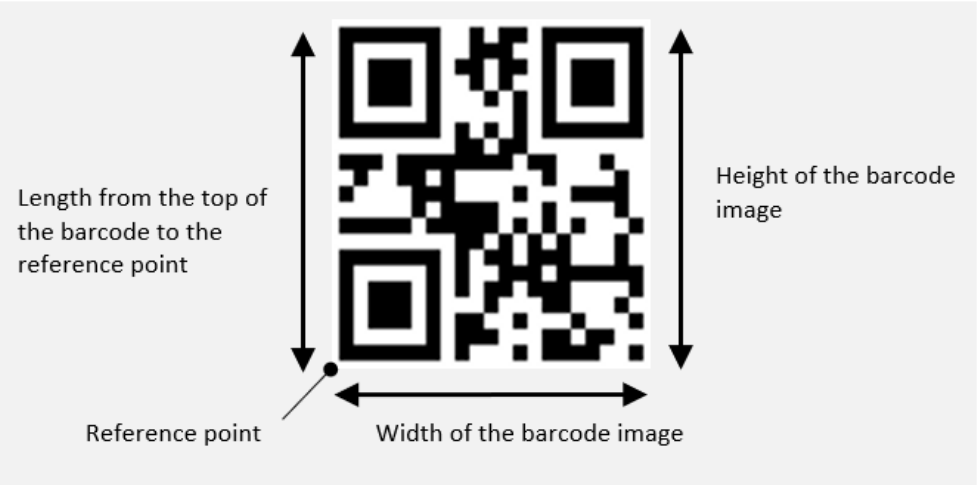
*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: row≠0

(2) Width of the barcode image

pdf417Symol	moduleSize	**Width of Barcode Image**
SII_PM_PDF417_STANDARD	**SII_PM_PDF417_MODULE_WIDTH_2**	(17×column*1+69)×2+8
	SII_PM_PDF417_MODULE_WIDTH_3	(17×column*1+69)×3+12
	SII_PM_PDF417_MODULE_WIDTH_4	(17×column*1+69)×4+16
	SII_PM_PDF417_MODULE_WIDTH_5	(17×column*1+69)×5+20
	SII_PM_PDF417_MODULE_WIDTH_6	(17×column*1+69)×6+24
	SII_PM_PDF417_MODULE_WIDTH_7	(17×column*1+69)×7+28
	SII_PM_PDF417_MODULE_WIDTH_8	(17×column*1+69)×8+32
SII_PM_PDF417_COMPACT	**SII_PM_PDF417_MODULE_WIDTH_2**	(17×column*1+35)×2+8
	SII_PM_PDF417_MODULE_WIDTH_3	(17×column*1+35)×3+12
	SII_PM_PDF417_MODULE_WIDTH_4	(17×column*1+35)×4+16
	SII_PM_PDF417_MODULE_WIDTH_5	(17×column*1+35)×5+20
	SII_PM_PDF417_MODULE_WIDTH_6	(17×column*1+35)×6+24
	SII_PM_PDF417_MODULE_WIDTH_7	(17×column*1+35)×7+28
	SII_PM_PDF417_MODULE_WIDTH_8	(17×column*1+35)×8+32

*1: column≠0

B.1.3 `printQRCode, printPageModeQRCode`



(1) Height and width of the barcode image

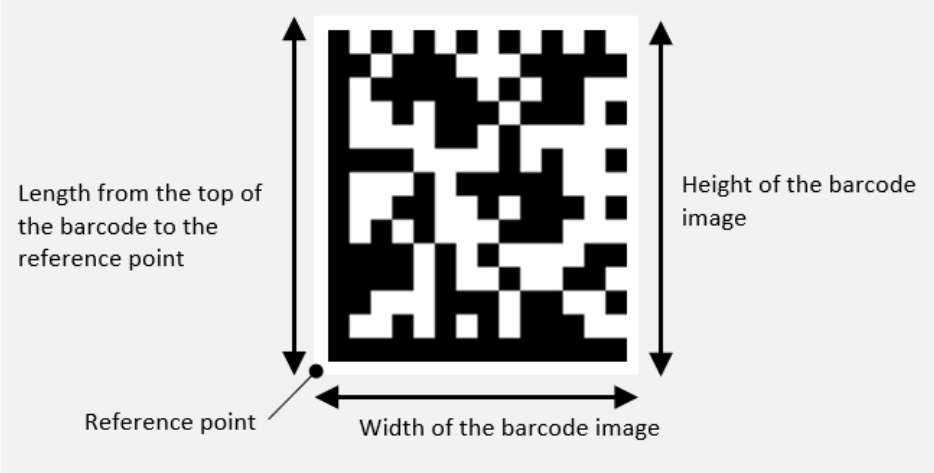
Height[*1] and width of the barcode image = (4 × version[*2] + 17 + 8) × module size value

*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: The version is determined by the content of the barcode data and the error correction level.

Module Size Value

moduleSize	Module Size Value
SII_PM_QR_MODULE_SIZE_2	2
SII_PM_QR_MODULE_SIZE_3	3
SII_PM_QR_MODULE_SIZE_4	4
SII_PM_QR_MODULE_SIZE_5	5
SII_PM_QR_MODULE_SIZE_6	6
SII_PM_QR_MODULE_SIZE_7	7
SII_PM_QR_MODULE_SIZE_8	8
SII_PM_QR_MODULE_SIZE_9	9
SII_PM_QR_MODULE_SIZE_10	10
SII_PM_QR_MODULE_SIZE_11	11
SII_PM_QR_MODULE_SIZE_12	12
SII_PM_QR_MODULE_SIZE_13	13
SII_PM_QR_MODULE_SIZE_14	14
SII_PM_QR_MODULE_SIZE_15	15
SII_PM_QR_MODULE_SIZE_16	16

B.1.4 `printDataMatrix, printPageModeDataMatrix`



(1) Height and width of the barcode image

Height of the barcode image = number of vertical module × module size value

Width of the barcode image = number of horizontal module × module size value

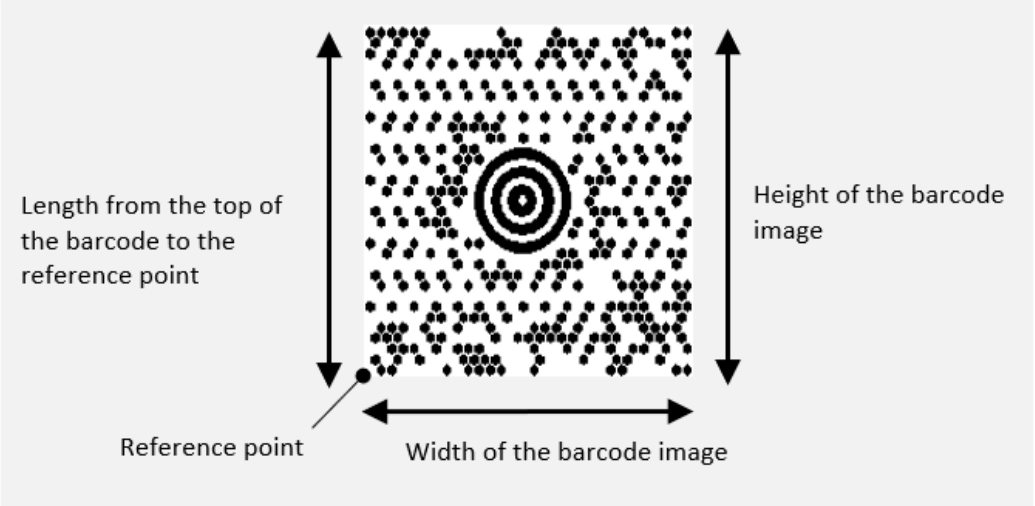
dataMatrixModule	Number of Vertical Module	Number of Horizontal Module
SII_PM_DATA_MATRIX_10_10	10	10
SII_PM_DATA_MATRIX_12_12	12	12
SII_PM_DATA_MATRIX_14_14	14	14
SII_PM_DATA_MATRIX_16_16	16	16
SII_PM_DATA_MATRIX_18_18	18	18
SII_PM_DATA_MATRIX_20_20	20	20
SII_PM_DATA_MATRIX_22_22	22	22
SII_PM_DATA_MATRIX_24_24	23	23
SII_PM_DATA_MATRIX_26_26	26	26
SII_PM_DATA_MATRIX_32_32	32	32
SII_PM_DATA_MATRIX_36_36	36	36
SII_PM_DATA_MATRIX_40_40	40	40
SII_PM_DATA_MATRIX_44_44	44	44
SII_PM_DATA_MATRIX_48_48	48	48
SII_PM_DATA_MATRIX_52_52	52	52
SII_PM_DATA_MATRIX_64_64	64	64
SII_PM_DATA_MATRIX_72_72	72	72
SII_PM_DATA_MATRIX_80_80	80	80
SII_PM_DATA_MATRIX_88_88	88	88
SII_PM_DATA_MATRIX_96_96	96	96
SII_PM_DATA_MATRIX_104_104	104	104
SII_PM_DATA_MATRIX_120_120	120	120
SII_PM_DATA_MATRIX_132_132	132	132

dataMatrixModule	Number of Vertical Module	Number of Horizontal Module
SII_PM_DATA_MATRIX_144_144	144	144
SII_PM_DATA_MATRIX_8_18	8	18
SII_PM_DATA_MATRIX_8_32	8	32
SII_PM_DATA_MATRIX_12_26	12	26
SII_PM_DATA_MATRIX_12_36	12	36
SII_PM_DATA_MATRIX_16_36	16	36
SII_PM_DATA_MATRIX_16_48	16	48

Module Size Value

moduleSize	Module Size Value
SII_PM_DATAMATRIX_MODULE_SIZE_2	2
SII_PM_DATAMATRIX_MODULE_SIZE_3	3
SII_PM_DATAMATRIX_MODULE_SIZE_4	4
SII_PM_DATAMATRIX_MODULE_SIZE_5	5
SII_PM_DATAMATRIX_MODULE_SIZE_6	6
SII_PM_DATAMATRIX_MODULE_SIZE_7	7
SII_PM_DATAMATRIX_MODULE_SIZE_8	8
SII_PM_DATAMATRIX_MODULE_SIZE_9	9
SII_PM_DATAMATRIX_MODULE_SIZE_10	10
SII_PM_DATAMATRIX_MODULE_SIZE_11	11
SII_PM_DATAMATRIX_MODULE_SIZE_12	12
SII_PM_DATAMATRIX_MODULE_SIZE_13	13
SII_PM_DATAMATRIX_MODULE_SIZE_14	14
SII_PM_DATAMATRIX_MODULE_SIZE_15	15
SII_PM_DATAMATRIX_MODULE_SIZE_16	16

B.1.5 `printMaxicode, printPageModeMaxicode`



(1) Height of the barcode image

Height of the barcode image[*1] = 215

*1: Height of the barcode image = Length from the top of the barcode to the reference point

(2) Width of the barcode image

Width of the barcode image = 225

B.1.6 `printGS1DataBarStacked, printPageModeGS1DataBarStacked`



(1) Height and width of the barcode image

Height of the barcode image[*1] = 13 × module size value

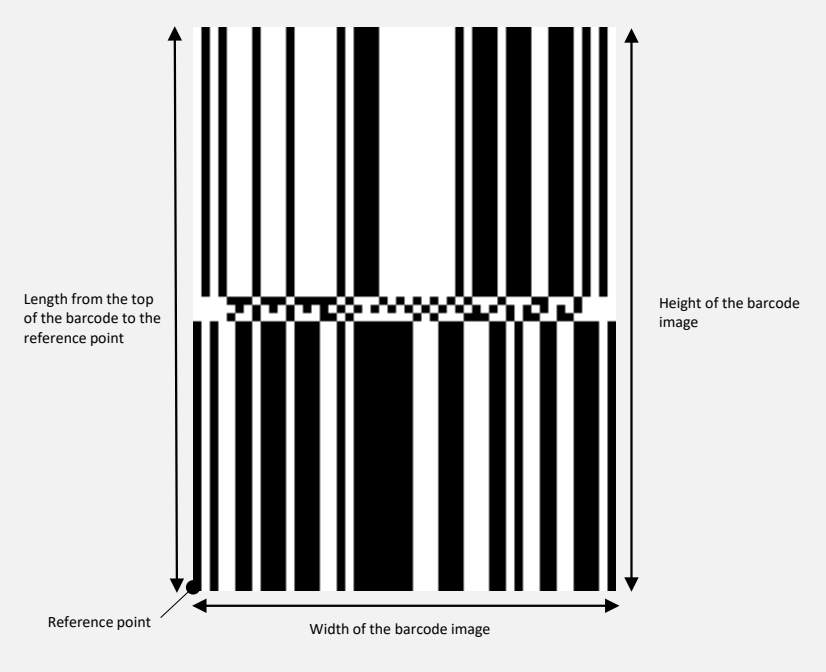
*1: Height of the barcode image = Length from the top of the barcode to the reference point

Width of the barcode image = 50 × module size value

Module Size Value

`moduleSize`	**Module Size Value**
SII_PM_GS1DATABAR_MODULE_SIZE_2	2
SII_PM_GS1DATABAR_MODULE_SIZE_3	3
SII_PM_GS1DATABAR_MODULE_SIZE_4	4
SII_PM_GS1DATABAR_MODULE_SIZE_5	5
SII_PM_GS1DATABAR_MODULE_SIZE_6	6
SII_PM_GS1DATABAR_MODULE_SIZE_7	7
SII_PM_GS1DATABAR_MODULE_SIZE_8	8
SII_PM_GS1DATABAR_MODULE_SIZE_9	9
SII_PM_GS1DATABAR_MODULE_SIZE_10	10
SII_PM_GS1DATABAR_MODULE_SIZE_11	11
SII_PM_GS1DATABAR_MODULE_SIZE_12	12
SII_PM_GS1DATABAR_MODULE_SIZE_13	13
SII_PM_GS1DATABAR_MODULE_SIZE_14	14
SII_PM_GS1DATABAR_MODULE_SIZE_15	15
SII_PM_GS1DATABAR_MODULE_SIZE_16	16

B.1.7 `printGS1DataBarStackedOmnidirectional`, `printPageModeGS1DataBarStackedOmnidirectional`



(1) Height and width of the barcode image

Height of the barcode image[*1] = (`moduleHeight` × 2 + 3) × module size value

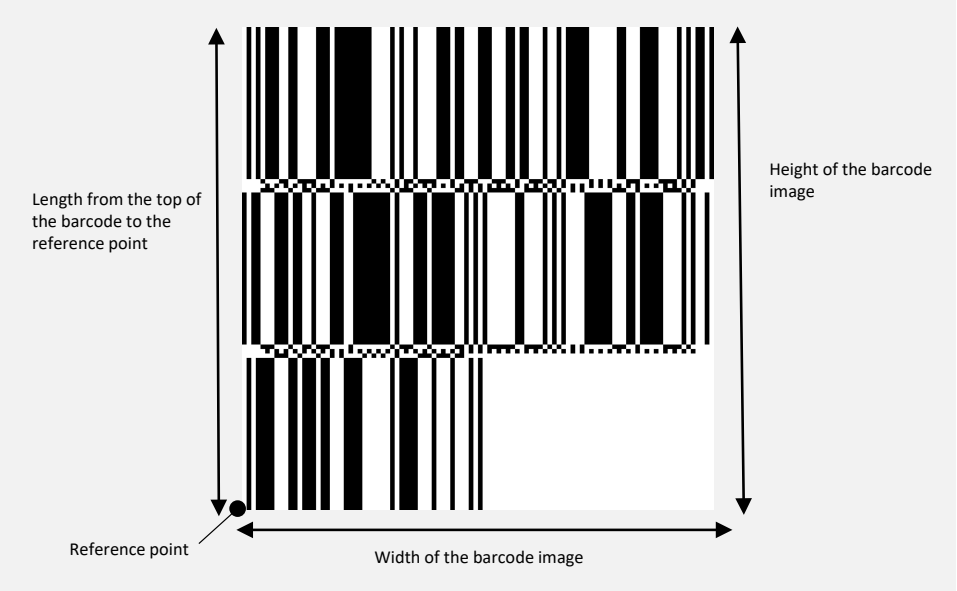
*1: Height of the barcode image = Length from the top of the barcode to the reference point

Width of the barcode image = 50 × module size value

Module Size Value

`moduleSize`	**Module Size Value**
SII_PM_GS1DATABAR_MODULE_SIZE_2	2
SII_PM_GS1DATABAR_MODULE_SIZE_3	3
SII_PM_GS1DATABAR_MODULE_SIZE_4	4
SII_PM_GS1DATABAR_MODULE_SIZE_5	5
SII_PM_GS1DATABAR_MODULE_SIZE_6	6
SII_PM_GS1DATABAR_MODULE_SIZE_7	7
SII_PM_GS1DATABAR_MODULE_SIZE_8	8
SII_PM_GS1DATABAR_MODULE_SIZE_9	9
SII_PM_GS1DATABAR_MODULE_SIZE_10	10
SII_PM_GS1DATABAR_MODULE_SIZE_11	11
SII_PM_GS1DATABAR_MODULE_SIZE_12	12
SII_PM_GS1DATABAR_MODULE_SIZE_13	13
SII_PM_GS1DATABAR_MODULE_SIZE_14	14
SII_PM_GS1DATABAR_MODULE_SIZE_15	15
SII_PM_GS1DATABAR_MODULE_SIZE_16	16

B.1.8 `printGS1DataBarExpandedStacked`, `printPageModeGS1DataBarExpandedStacked`



(1) Height and width of the barcode image

Height of the barcode image[*1] = ((34 + 3) × number of row[*2] + 34) × module size value

*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: The number of row is determined by the barcode data.

Width of the barcode image = (4 + 49 × `column` / 2) × module size value

Module Size Value

`moduleSize`	**Module Size Value**
SII_PM_GS1DATABAR_MODULE_SIZE_2	2
SII_PM_GS1DATABAR_MODULE_SIZE_3	3
SII_PM_GS1DATABAR_MODULE_SIZE_4	4
SII_PM_GS1DATABAR_MODULE_SIZE_5	5
SII_PM_GS1DATABAR_MODULE_SIZE_6	6
SII_PM_GS1DATABAR_MODULE_SIZE_7	7
SII_PM_GS1DATABAR_MODULE_SIZE_8	8
SII_PM_GS1DATABAR_MODULE_SIZE_9	9
SII_PM_GS1DATABAR_MODULE_SIZE_10	10
SII_PM_GS1DATABAR_MODULE_SIZE_11	11
SII_PM_GS1DATABAR_MODULE_SIZE_12	12
SII_PM_GS1DATABAR_MODULE_SIZE_13	13
SII_PM_GS1DATABAR_MODULE_SIZE_14	14
SII_PM_GS1DATABAR_MODULE_SIZE_15	15
SII_PM_GS1DATABAR_MODULE_SIZE_16	16

Appendix C
Open Source Software License

This chapter describes the License of open source software used in the library.

C.1 MIT License

- **SSZipArchive**

  Copyright (c) 2010-2012 Sam Soffes

  Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

  The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

C.2 Apache License 2.0

- **zxingify-objc**

  Copyright 2012 ZXing authors

  Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.