BIXOLON

API Reference Guide
Flutter Plugin

Ver. 1.00

http://www.bixolon.com

Table of Contents

Copyright

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer "OFF", before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer "OFF".

1. Manual Guide

This manual provides descriptions of contents necessary for developing applications using React Native.

2. Supported Devices

The below table summarizes the supported devices that are available.

Receipt Printers	Resolution	Supported Communication	Max Printable Width (dots)
SPP-R200II	203 dpi	Bluetooth, Wi-Fi	384
SPP-R200III	203 dpi		384
SPP-R210	203 dpi		384
SPP-R300	203 dpi		576
SPP-R310	203 dpi		576
SPP-R400	203 dpi		832
SPP-R410	203 dpi		832
SPP-C200	203 dpi	Bluetooth	384
SPP-C300	203 dpi		576
SRP-350IIOBE	180 dpi	Bluetooth, Ethernet	512
SRP-350plusIII	180 dpi	Bluetooth, Ethernet, Wi-Fi	512
SRP-352plusIII	203 dpi		576
SRP-380	180 dpi		512
SRP-382	203 dpi		576
SRP-383	300 dpi		864
SRP-Q200	203 dpi		432
SRP-Q300	180 dpi		512
SRP-Q302	203 dpi		576
SRP-F310II	180 dpi		512
SRP-F312II	203 dpi		576
SRP-F313II	203 dpi		640
SRP-S300	203 dpi		576
SRP-S320	203 dpi		576
SRP-350plusV	180 dpi		512
SRP-352plusV	203 dpi		576
SRP-380II	180 dpi		512
SRP-382II	203 dpi		576
SRP-350III	180 dpi	Ethernet	512
SRP-352III	203 dpi		576
SRP-330II	180 dpi		512
SRP-332II	203 dpi		576
SRP-S200	203 dpi		432
SRP-QE300	180 dpi		512
SRP-QE302	203 dpi		576
SRP-E300	180 dpi		512
SRP-E302	203 dpi		576

SRP-350V	180 dpi		512
SRP-352V	203 dpi		576
SRP-330III	180 dpi		512
SRP-332III	203 dpi		576
SRP-S3000	203 dpi	Ethernet, Wi-Fi	576

Label Printers	Resolution	Supported Communication	Max Printable Width
SLP-DX220	203 dpi	Bluetooth, Wi-Fi, Ethernet	432
SLP-DX223	300 dpi		672
SLP-TX220	203 dpi		432
SLP-TX223	300 dpi		672
SLP-DX420	203 dpi		864
SLP-DX423	300 dpi		1248
SLP-TX420	203 dpi		864
SLP-TX423	300 dpi		1248
SLP-TX400(RFID)	203 dpi		864
SLP-TX403(RFID)	300 dpi		1248
SRP-770III	203 dpi		832
SLP-DL410	203 dpi		864
SLP-DL413	300 dpi		1248
XT5-40(RFID)	203 dpi		832
XT5-43(RFID)	300 dpi		1248
XT5-46(RFID)	600 dpi		2496
XD5-40d	203 dpi		864
XD5-43d	300 dpi		1248
XD5-40t(RFID)	203 dpi		864
XD5-43t(RFID)	300 dpi		1248
XL5-40	203 dpi		864
XL5-43	300 dpi		1248
XT3-40	203 dpi		864
XT3-43	300 dpi		1248
SRP-E770III	203 dpi	Ethernet	832
SPP-L3000	203 dpi	Bluetooth, Wi-Fi	576
SPP-L310	203 dpi		576
SPP-L410	203 dpi		832
XM7-20	203 dpi	Bluetooth, Wi-Fi, * Ethernet (Cradle Required)	384
XM7-40(RFID)	203 dpi		832
SRP-S3000_LABEL	203 dpi	Wi-Fi, Ethernet	576

3. Flutter Plugin Installation

3-1 Supported Platforms

Platform	Description
Android	Android 4.0.3(API Level 15: ice cream sandwich) or later
iOS	iOS 12.0 or later

3-2 Installing Flutter Plugin

No.	Installation Steps

1	To install this Flutter Plugin, open the "**pubsepec.yaml**" file in your Flutter APP project and add "bxlflutterbgatelib" and "path" as shown below.
	`// pubspec.yaml file`
	`...`
	`dependencies:`
	`  flutter:`
	`    sdk: flutter`
	`  bxlflutterbgatelib:`
	`    path: ..`
	`...`

2	Run the command below **flutter pub get** If there is a problem in updating the plugin or running the app, additionally execute the command below. **rm –rf ios/Pods && rm ios/Podfile.lock && flutter clean** **flutter build ios or flutter build android**

3-3 Required permissions

iOS

Update the **Info.plist file** by opening it in XML editor or XCode's editor. Add the following to the file. This is to allow your app to access the Bluetooth communication.
`<key>UISupportedExternalAccessoryProtocols</key>`
`<array><string>com.bixolon.protocol</string></array>`
`<key>NSBluetoothAlwaysUsageDescription</key>`
`<string>Communication with the printer for printing.</string>`
`<key>NSBluetoothPeripheralUsageDescription</key>`
`<string>Communication with the printer for printing.</string>`

Multicast Networking Entitlement Request If the iOS version is 14 or later, you must need to request multicast permission from Apple to discover devices. See the link below. 1) https://developer.apple.com/news/?id=0oi77447 2) https://developer.apple.com/contact/request/networking-multicast - IP address used for discovering devices: 255.255.255.255 - Port numbers used for discovering devices: 48780, 48781, 9000, 3337a

Android
Add following permissions your Android Manifest file, located in <project root>/android/app/src/main/AndroidManifest.xml. - BLUETOOTH, BLUETOOTH_ADMIN, BLUETOOTH_PRIVILEGED - ACCESS_WIFI_STATE, CHANGE_WIFI_STATE - ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION

```
<manifest                          xmlns:android="http://schemas.android.com/apk/res/android"
android:installLocation="auto">

<uses-sdk android:minSdkVersion="15" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" />
<uses-permission android:name="android.permission. ACCESS_WIFI_STATE " />
<uses-permission android:name="android.permission. CHANGE_WIFI_STATE " />
<uses-permission android:name="android.permission. ACCESS_COARSE_LOCATION " />
<uses-permission android:name="android.permission. ACCESS_FINE_LOCATION " />
<uses-feature android:name="android.hardware.usb.host" />

<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
        <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
        <category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
<meta-data                    android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
android:resource="@xml/device_filter" />

</manifest>
```

4. Common API

4-1 Overview

• Methods below in 4-2 are commonly used in “MPosControllerPrinter”, “MPosControllerLabelPrinter” and “MPosControllerConfig” classes.

4-2 Methods

4-2-1 selectInterface

Configures the interface type and address of the device(printer) to connect.

This method must be used before calling the “openService” method.
USB communication is only available on Android.

[Syntax]
Future<int?> **selectInterface**(int interfaceType, String address) async

[Parameters]
• interfaceType: Communication type to connect the device.

1	: Wi-Fi
2	: Ethernet
4	: Bluetooth Classic
8	: BLE (Bluetooth Low Energy)
16	: USB

• address: Address Information

In case of network (Wi-Fi or Ethernet) communication, it should be used in the form of "IP address [: port number]". Port number is 9100 by default and can be omitted.

In case of Bluetooth communication, it should be entered as the value described in the table below according to platform.

Platform	Address value
iOS	Bluetooth Serial Number
Android	Bluetooth MAC Address

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

4-2-2 selectCommandMode

Selects whether to communicate directly with the device or via "B-gate".

[Syntax]
Future<int?> **selectCommandMode**(int commandMode) async

[Parameters]
• commandMode: Communication method

0	: Communication via B-gate
1	: Direct communication

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

4-2-3 openService

Establishes a communication connection with the device.

[Syntax]
Future<int?> **openService**({ int deviceId = -1, int timeout = 3}) async
[Parameters]
• deviceId: Device ID assigned by B-gate or -1 for the communication without B-gate
• timeout: Maximum wait time for device connection (in seconds)

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

4-2-4 closeService

Disconnects the communication with the device.

❗	If the print time is long and the timeout is too small, the data transmission may fail. Therefore, set an appropriate timeout value.

[Syntax]
Future<int?> **closeService**({ int timeout = 3}) async

[Parameters]
• timeout: Timeout in second for disconnecting a device.

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

4-2-5 directIO

Sends user-defined data to the device.

[Syntax]
Future<int?> **directIO**(List<int> data) async

[Parameters]
• data: Data to be sent to device.

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

4-2-6 setTransaction

Enters or Leaves 'Transaction' mode, and then sends the data in the buffer.

It is recommended for output devices such as printers to use this mode.

[Syntax]
Future<int?> **setTransaction**(int transaction) async

[Parameters]
• transaction: the transaction mode

0: Leave transaction mode after data transmission
1: Enter transaction mode

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

4-2-7 isOpen

Check that communication with the device is possible or not.

[Syntax]
Future<int?> **isOpen**() async

[Parameters]
None

[Returns]
An instance of Future<int?> representing the result of the method execution.

If it can communicate with the HOST, 1 is passed, otherwise 0 is passed as the execution result value.

4-2-8 setReadMode

Set the data reading mode of the device connected to B-gate.

No need to call this method if HOST is communicating with the device directly without a B-gate.

[Syntax]
Future<int?> **setReadMode**(int mode) async

[Parameters]
• mode: data reading mode

0: Always Receives data from B-gate
1: Only once receives data from B-gate
2: No receives data from B-gate

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5. MPosControllerPrinter API

5-1 Overview

• The “MPosControllerPrinter” is a class which provides methods for controlling receipt printers.

5-2 Methods

5-2-1 printText

Prints characters.

[Syntax]

```
Future<int?> printText(String data,
        {
           int fontType = 0,
           int fontWidth = 0,
           int fontHeight = 0,
           int bold = 0,
           int underline = 0,
           int reverse = 0,
           int alignment = 0
        }) async
```

[Parameters]

• data: Character string
• fontType: Font type

0: Font A (12 × 24 dots)
1: Font B (9 × 17 dots)
2: Font C (9 × 24 dots)

• fontWidth: Font width

$0 \le fontWidth \le 7$

• fontHeight: Font height

$0 \le fontHeight \le 7$

• bold: Bold

0: Emphasized mode off
1: Emphasized mode on

• underline: Underline

0: No underline
1: Underline (1 dot)
2: Underline (2 dot)

• reverse: Turns white/black reverse print on or off

```
0 : off
1 : on
```

• alignment: Alignmet in standard mode.

```
0: Left
1: Center
2: Right
```

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-2 setCharacterset

Sets the code page used for encoding characters.

[Syntax]
Future<int?> **setCharacterset**(int characterset) async

[Parameters]
• characterset: Code page for encoding a character string

0: USA, Standard Europe
1: Katakana
2: Western European
3: Portuguese
4: Canadian-French
5: Nordic
16: Latin 1
17: Cyrillic (PC866)
18: Latin 2 (PC852)
19: Euro (PC858)
21: Hebrew DOS code (PC862)
22: Arabic (PC864)
34: Thai character code 11
31: Thai character code 14
39: Thai character code 16
35: Thai character code 18
23: Thai character code 42
24: Greek (WPC1253)
25: Turkiye (WPC1254)
26: Baltic (WPC1257)
27: Persian
28: Cyrillic (WPC1251)
29: Greek (PC737)
30: Baltic (PC775)
32: Hebrew Old
33: Hebrew New code (WPC1255)
36: Cyrillic (PC855)
37: Turkiye (PC857)
38: Greek (PC928)
40: Arabic (WPC1256)
41: Vietnamese (PC1258)
42: Khmer
47: Central European (PC1250)
48: Latin 9
49: Vietnamese (TCVN3)

50: Vietnamese (TCVN3 Capital)
51: Vietnamese (VISCII)
52: Albanian (PC912)
949: Korean (KS5601)
932: Japanese (ShiftJIS)
950: Chinese (BIG5)
936: Chinese (GB2312)
54936: Chinese (GB18030)
54937: Unicode (UTF8)
54938: Unicode (UTF16)
54939: Unicode (UTF32)

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-3 setInternationalCharacterset

Sets the character set used for encoding international characters.

[Syntax]
Future<int?> **setInternationalCharacterset**(int characterset) async

[Parameters]
• characterset: International character set

0:	USA Code
1:	FRANCE Code
2:	GERMANY Code
3:	UK Code
4:	DENMARK1 Code
5:	SWEDEN Code
6:	ITALY Code
7:	SPAIN1 Code
8:	JAPAN Code
9:	NORWAY Code
10:	DENMARK 2 Code
11:	SPAIN 2 Code
12:	LATIN AMERICA Code
13:	KOREA Code
14:	SLOVENIA / CROATIA Code
15:	CHINA Code

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-4 setPagemode

Enters or Leaves 'Page' mode, and then sends the data in the buffer.

Pagemode cannot be used simultaneously with transaction mode.

[Syntax]
Future<int?> **setPagemode**(int mode) async

[Parameters]
• mode: Enter or leave page mode

0: Leave Pagemode and then start printing
1: Enter Pagemode

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-5 setPagemodePrintArea

Sets the printing area of pagemode.

[Syntax]
Future<int?> **setPagemodePrintArea**(int x, int y, int width, int height) async

[Parameters]
• x: The starting X axis coordinate, in dot unit
• y: The starting Y axis coordinate, in dot unit
• width: Printing width, in dot unit
• height: Printing height, in dot unit

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-6 setPagemodeDirection

Sets the printing direction in pagemode.

[Syntax]
Future<int?> **setPagemodeDirection**(int direction) async

[Parameters]
• direction: Printing direction

0:	0 degree turn
1:	90 degree turn (clockwise)
2:	180 degree turn
3:	90 degree turn (counterclockwise)

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-7 setPagemodePosition

Sets the coordinates of the object to be printed in page mode.

[Syntax]
Future<int?> **setPagemodePosition**(int x, int y) async

[Parameters]
• x: X axis coordinate, in dot unit, of the object(text, barcode,image) to be printed.
• y: Y axis coordinate, in dot unit, of the object(text, barcode,image) to be printed.

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-8 printImageFile

Prints Bitmap file specified by the path.

[Syntax]

```
Future<int?> printImageFile(String fileName,
        int width,
        {
          int alignment = 0,
          int threshold = 128,
          int ditheringType = 1,
          int compressType = 1
        }) async
```

[Parameters]

• fileName: Path of the image file.
• width: Print width
• alignment: Alignmet

0: Left 1: Center 2: Right

• threshold: Print threshold

0 ≤ threshold ≤ 255

• ditheringType: Dithering type

0: No dither 1: Applying dither

• compressType: Data compression method

0: No Compress 1: Algorithm - RLE 2: Algorithm - LZMA

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-9 printBase64Image

Prints an image encoded with BASE64

[Syntax]

```
Future<int?> printBase64Image(String base64String,
        int width,
        {
          int alignment = 0,
          int threshold = 128,
          int ditheringType = 1,
          int compressType = 1
        }) async
```

[Parameters]

- base64String: String data in BASE64 format
- width: Print width
- alignment: Alignmet

  0: Left
  1: Center
  2: Right

- threshold: Print threshold

  0 ≤ threshold ≤ 255

- ditheringType: Dithering type

  0: No dither
  1: Applying dither

- compressType: Data compression method

  0: No Compress
  1: Algorithm - RLE
  2: Algorithm - LZMA

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-10 printPDFFile

Prints PDF file specified by the path.

[Syntax]

```
Future<int?> printPDFFile(String fileName,
      int width,
      int startPage,
      int endPage,
      {
        int alignment = 0,
        int threshold = 128,
        int ditheringType = 0,
        int compressType = 1,
      }) async
```

[Parameters]

• fileName: Path of the PDF file.
• width: Print width
• alignment: Alignmet

  0: Left
  1: Center
  2: Right

• startPage: The first page number

  0 ≤ startPage < The number of pages

• endPage: The last page number

  startPage ≤ endPage

• threshold: Print threshold

  0 ≤ threshold ≤ 255

• ditheringType: Dithering type

  0: No dither
  1: Applying dither

• compressType: Data compression method

  0: No Compress
  1: Algorithm - RLE
  2: Algorithm - LZMA

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-11 print1DBarcode

Prints an one dimenional barcode.

[Syntax]

```
Future<int?> print1DBarcode(String data,
        int symbology,
        int barWidth,
        int height,
        {
          int alignment = 0,
          int textPostion = 0
        }) async
```

[Parameters]

• data: Barcode data

• symbology: Barcode symbol type

101: UPC A
102: UPC E
103: EAN 8/JAN 8
104: EAN 13/JAN 13
106: ITF
107: CODABAR
108: CODE39
109: CODE93
110: CODE128
111: GS128
112: GS1 Databar (Omnidirectional)
113: GS1 Databar (Truncated)
114: GS1 Databar (Limited)

• barWidth: Barcode width

2 ≤ barWidth ≤ 6

• height: height

1 ≤ height ≤ 255

• alignment: Alignmet

0: Left
1: Center
2: Right

• textPosition: Printing position of HRI (Human Readable Interface)

0: No HRI print
1: Print above barcode
2: Print below barcode

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-12 printQRCode

Prints the QR Code.

[Syntax]

Future<int?> **printQRCode**(String data,
    int model,
    int moduleSize,
    int eccLevel,
    { int alignment = 0 }) async

[Parameters]

• data: QR Code data
• model: QR Code model

204: MODEL 1
205: MODEL 2

• moduleSize: Module size

1 ≤ moduleSize ≤ 8

• eccLevel: Error correction level

48: Level: L
49: Level: M
50: Level: Q
51: Level: H

• alignment: Alignmet

0: Left
1: Center
2: Right

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-13 printPDF417

Prints the PDF417 code.

[Syntax]

```
Future<int?> printPDF417(String data,
        int symbol,
        int columnNumber,
        int rownumber,
        int moduleWidth,
        int moduleHeight,
        int eccLevel,
        { int alignment = 0 }) async
```

[Parameters]

• data: PDF417 data

• symbol: PDF417 type

201: PDF417 Standard 202: PDF417 Simplified

• columnNumber: Number of columns

1 ≤ columnNumber ≤ 30

• rowNumber: Number of rows

3 ≤ rowNumber ≤ 90

• moduleWidth: Module width

1 ≤ moduleWidth ≤ 4

• moduleHeight: Module height

2 ≤ moduleHeight ≤ 8

• eccLevel: The error correction level

48: Error Correction Level 0 49: Error Correction Level 1 50: Error Correction Level 2 51: Error Correction Level 3 52: Error Correction Level 4 53: Error Correction Level 5 54: Error Correction Level 6 55: Error Correction Level 7 56: Error Correction Level 8

• alignment: Alignmet

0: Left 1: Center 2: Right

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-14 printDataMatrix

Prints the Data Matrix code.

[Syntax]

Future<int?> **printDataMatrix**(String data,
    int moduleSize,
    { int alignment = 0 }) async

[Parameters]

• data: DataMatrix code data
• moduleSize: Number of columns

    2 ≤ moduleSize ≤ 3

• alignment: Alignmet

    0: Left
    1: Center
    2: Right

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-15 printGS1Databar

Prints the GS1 databar.

• This method works only in the following printer models.
- SRP-380/382/383, SRP-Q300/Q302

[Syntax]

Future<int?> **printGS1Databar**(String data,
int symbol,
int size,
{ int alignment = 0 }) async

[Parameters]

• data: GS1 data bar data

• symbol: GS1 data bar symbol

72: GS1 Databar Stacked
73: GS1 Databar Stacked Omnidirectional

• size: Module size

1 ≤ size ≤ 8

• alignment: Alignmet

0: Left
1: Center
2: Right

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-16 printGS1DatabarMobile

Prints the GS1 databar.

This method only works for mobile printers whose model name starts with “SPP”.

[Syntax]

Future<int?> **printGS1DatabarMobile**(String data,
    String cData,
    int symbol,
    int moduleWidth,
    int moduleHeight,
    int segmentHeight,
    int separatorHeight,
    { int alignment = 0 }) async

[Parameters]

• data: data for 1D barcode
• cData: data for 2D code
• symbol: symbol for 1D barcode

50: GS1 Databar Omnidirectional barcode
51: GS1 Databar truncated barcode
52: GS1 Databar stacked barcode
53: GS1 Databar stacked omnidirectional barcode
56: UPC-A barcode
57: UPC-E barcode
58: EAN-13 barcode
59: EAN-8 barcode
60: UCC/EAN-128 CC-A/B
61: UCC/EAN-128 CC-C

• moduleWidth: module width

1 ≤ moduleWidth ≤ 8

• moduleHeight: module height

1 ≤ moduleHeight ≤ 8

• segmentHeight: segment height

1 ≤ segmentHeight ≤ 2

• separatorHeight: separator height

1 ≤ separatorHeight ≤ 2

• alignment: Alignmet

0: Left
1: Center
2: Right

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-17 printCompositeSymbology

Prints the composite symbology.

• This method works only in the following printer models.
  - SRP-380/382/383, SRP-Q300/Q302

[Syntax]

```
Future<int?> printCompositeSymbology(String data,
        String cData,
        int symbol,
        int cSymbol,
        int size,
        { int alignment = 0 }) async
```

[Parameters]

• data: data for 1D barcode
• cData: data for 2D code
• symbol: GS1 Databr symbol

65:	EAN8
66:	EAN13
67:	UPC-A
69:	UPC-E
70:	GS1 Databar Omnidirectional
71:	GS1 Databar truncated
72:	GS1 Databar stacked
73:	GS1 Databar stacked omnidirectional
74:	GS1 Databar limited
75:	GS1 Databar expanded
77:	GS1 128

• cSymbol: 2-dimentional synthetic element symbol

65:	Auto
66:	CC-C fixed (GS1 128 Only)

• size: Number of columns

1 ≤ size ≤ 8

• alignment: Alignmet

```
0: Left
1: Center
2: Right
```

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-18 printMaxicode

Prints the maxicode.

This method only works for mobile printers whose model name starts with “SPP”.

[Syntax]

Future<int?> **printMaxicode**(String data,
    int mode,
    { int alignment = 0 }) async

[Parameters]

• data: Maxicode data
• mode: Maxicode mode

50:	Mode 2
51:	Mode 3
52:	Mode 4

• alignment: Alignmet

0:	Left
1:	Center
2:	Right

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-19 printAztec

Prints the Aztec code.

This method only works for mobile printers whose model name starts with “SPP”.

[Syntax]

Future<int?> **printAztec**(String data,
    int moduleSize,
    int eccLevel,
    int mode,
    { int alignment = 0 }) async

[Parameters]

• data: Aztec code data

• moduleSize: Number of columns

    1 ≤ moduleSize ≤ 8

• eccLevel: Error correction level

    48: Error correction level 10%
    49: Error correction level 23%
    50: Error correction level 36%
    51: Error correction level 50%

• mode: Aztec code mode

    0: Data mode
    1: GS1 mode
    2: Unicode mode

• alignment: Alignmet

    0: Left
    1: Center
    2: Right

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-20 printLine

Prints the straight line.

• This method is available in page mode, and print direction must be left to right (no rotation), and only straight line can be printed.
• This method only works for mobile printers whose model name starts with “SPP”.

[Syntax]

Future<int?> **printLine**(int x1,
    int y1,
    int x2,
    int y2,
    int thickness) async

[Parameters]

• x1: x-coordinate, in dot unit, of the start point of the straight line
• y1: y-coordinate, in dot unit, of the start point of the straight line
• x2: x-coordinate, in dot unit, of the end point of the straight line
• y2: y-coordinate, in dot unit, of the end point of the straight line
• thickness: Thickness of the straight line

0 ≤ thickness ≤ 16

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-21 printBox

Prints the rectangle.

• This method is available in page mode, and print direction must be left to right (no rotation).
• This method only works for mobile printers whose model name starts with “SPP”.

[Syntax]

Future<int?> **printBox**(int left,
    int top,
    int right,
    int bottom,
    int thickness) async

[Parameters]

• left: x-coordinate, in dot unit, of the upper-left corner of the rectangle.
• top: y-coordinate, in dot unit, of the upper-left corner of the rectangle.
• right: x-coordinate, in dot unit, of the lower-right corner of the rectangle.
• bottom: y-coordinate, in dot unit, of the lower-right corner of the rectangle.
• thickness: Thickness of the rectangle

    0 ≤ thickness ≤ 16

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-22 checkPrinterStatus

Checks the printer’s current status.

[Syntax]
Future<int?> **checkPrinterStatus**() async

[Parameters]
None

[Returns]
An instance of Future<int?> representing the current printer's status

0	: Printing is possible
1	: No paper
2	: Cover open
4	: Insufficient paper
8	: Error (offline or unknown error)
64	: Battery level is low
256	: Cash Drawer Signal - High
512	: Cash Drawer Signal - Low
-1	: Failed in checking the status

5-2-23 checkBattStatus

Checks the printer’s current battery level.

This method works only on models with batteries.

[Syntax]
Future<int?> **checkBattStatus**() async

[Parameters]
None

[Returns]
An instance of Future<int?> representing the current status of the printer's battery.

0	: Battery Power Level - Full
16	: Battery Power Level - High
32	: Battery Power Level - Middle
64	: Battery Power Level - Low

5-2-24 asbEnable

Enable or disable ASB (Automatic Status Back).

[Syntax]
Future<int?> **asbEnable**(int enable) async

[Parameters]
• enable

0: Disable ASB (Auto Status Back)	
1: Enable ASB (Auto Status Back)	

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-25 cutPaper

Cuts the paper.

❗ Call this method only on printers equipped with auto-cutters.

[Syntax]
Future<int?> **cutPaper**(int cutType) async

[Parameters]
• cutType: Cut option

0	: No paper feed + partial cut
1	: No paper feed + full cut
65	: Auto paper feed + partial cut
66	: Auto paper feed + full cut

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-26 openDrawer

Outputs pulse to Drawer kick-out port.

Please note that the cash drawer has two PIN numbers (#2 and #5), as the cash drawer may not open depending on the first parameter.

[Syntax]
Future<int?> **openDrawer**(int pinNumber, { int onTime = 25, int offTime = 255 }) async

[Parameters]
• pinNumber: Drawer kick-out connector Pin number

0: Pin Number 2
1: Pin Number 5

• onTime: off time (×2ms)

0 ≤ onTime ≤ 255

• offTime: off time (×2ms)

0 ≤ offTime ≤ 255

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

5-2-27 getModelName

Gets the printer’s model name.

[Syntax]
Future<String?> **getModelName**() async

[Parameters]
None

[Returns]
An instance of Future<String?> representing the model name of the printer.

5-2-28 getFirmwareVersion

Gets the printer’s firmware version.

[Syntax]
Future<String?> **getFirmwareVersion**() async

[Parameters]
None

[Returns]
An instance of Future<String?> representing the firmware version.

5-2-29 getStatisticsData

Gets the printer’s statistical data information.

[Syntax]
Future<String?> **getStatisticsData**(int info) async

[Parameters]
• info: Statistical data type

0:	Motor
2:	TPH
4:	Cutter

[Returns]
An instance of Future<String?> representing the requested statistical data information.

6. MPosControllerLabelPrinter API

6-1 Overview

• The “MPosControllerLabelPrinter” is a class which provides methods for controlling label printers.

6-2 Methods

6-2-1 setCharacterset

Sets the codepage and international characterset used for encoding characters.

[Syntax]

Future<int?> **setCharacterset**(int characterset, int internationalCharacterset) async

[Parameters]

• characterset: Code page for encoding a character string

0	: USA, Standard Europe
1	: Western European
2	: Latin 2
3	: Portuguese
4	: Canadian-French
5	: Nordic
6	: Latin 1
8	: Turkiye
9	: Greek (PC 737)
10	: Central European
11	: Greek (ANSI 1253)
12	: Turkiye
13	: Cyrillic
14	: Hebrew DOS code
15	: Cyrillic #2
16	: Cyrillic
17	: Hebrew New code
18	: Greek
19	: Arabic
20	: Baltic
21	: Baltic
22	: Euro

• internationalChararacterset: International character set

0	: USA Code
1	: FRANCE Code

2	: GERMANY Code
3	: UK Code
4	: DENMARK1 Code
5	: SWEDEN Code
6	: ITALY Code
7	: SPAIN1 Code
8	: JAPAN Code
9	: NORWAY Code
10	: DENMARK 2 Code
11	: SPAIN 2 Code
12	: LATIN AMERICA Code
13	: KOREA Code
14	: SLOVENIA / CROATIA Code
15	: CHINA Code

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-2 checkPrinterStatus

Gets the printer’s current status.

[Syntax]
Future<int?> **checkPrinterStatus**() async

[Parameters]
None

[Returns]
An instance of Future<int?> representing the current printer's status

0	: Printing is possible
1	: No paper
2	: Cover is open
4	: Cutter error
8	: TPH overheating
16	: GAP sensor error
32	: No ribbon
64	: Building label buffer
128	: Printing
256	: Paper stuck to filler
512	: Board overheating
1024	: Motor overheating
2048	: Waiting for label to be taken
4096	: RFID writing Error
-1	: Status checking failed

6-2-3 printBuffer

Starts printing the contents saved in the printer buffer.

[Syntax]
Future<int?> **printBuffer**(int numberOfCopies) async

[Parameters]
• numberOfCopies: The number of copies

1 ≤ numberOfCopies ≤ 65535

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-4 drawTextDeviceFont

Saves data of the characters to the printer buffer using the bitmap font.

[Syntax]

```
Future<int?> drawTextDeviceFont(String data,
        int xPosition,
        int yPosition,
        String fontType,
        int fontWidth,
        int fontHeight,
        int rightSpace,
        {
          int rotation = 0,
          int reverse = 0,
          int bold = 0,
          int rightToLeft = 0,
          int alignment = 0
        }) async
```

[Parameters]

• data: Data for characters to be printed
• xPosition: In dot unit, x axis coordinates of the character string
• yPosition: In dot unit, y axis coordinates of the character string
• fontType: Font type

‘0’	: 1 byte character 9 X 15 (dots)
‘1’	: 1 byte character 12 X 20 (dots)
‘2’	: 1 byte character 16 X 25 (dots)
‘3’	: 1 byte character 19 X 30 (dots)
‘4’	: 1 byte character 24 X 38 (dots)
‘5’	: 1 byte character 32 X 40 (dots)
‘6’	: 1 byte character 48 X 76 (dots)
‘7’	: 1 byte character 22 X 34 (dots)
‘8’	: 1 byte character 28 X 44 (dots)
‘9’	: 1 byte character 37 X 58 (dots)
‘a’	: Korean 16 X 16 (dots) (English. Numbers 9 X 15)
‘b’	: Korean 24 X 24 (dots) (English. Numbers 12 X 24)
‘c’	: Korean 20 X 20 (dots) (English. Numbers 12 X 20)
‘d’	: Korean 26 X 26 (dots) (English. Numbers 16 X 30)
‘e’	: Korean 20 X 26 (dots) (English. Numbers 16 X 30)
‘f’	: Korean 38 X 38 (dots) (English. Numbers 22 X 34)
‘m’	: Chinese, GB2312 24 X 24 (dots) (English. Numbers 12 X 24)
‘n’	: Chinese, BIG5 24 X 24 (dots) (English. Numbers 12 X 24)
‘j’	: Japanese 24 X 24 (dots) (English. Numbers 12 X 24)

• fontWidth: The width expansion ratio

1 ≤ fontWidth ≤ 4

• fontHeight: The height expansion ratio

1 ≤ fontHeight ≤ 4

• rightSpace: The space on the right of a character

• rotation: The printing direction of characters

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

• reverse: White/Black reverse print on or off

0	: No reverse printing
1	: Black and white reverse printing

• bold: Bold or not

0	: Emphasized mode off
1	: Emphasized mode on

• rightToLeft: Direction of printing characters

0	: Left to right
1	: Right to left

• alignment: Alignment

0	: Align to the left
1	: Align to the right

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-5 drawTextVectorFont

Saves data of the character strings to the printer buffer using the vector font.

[Syntax]

```
Future<int?> drawTextVectorFont(String data,
        int xPosition,
        int yPosition,
        String fontType,
        int fontWidth,
        int fontHeight,
        {
          int rightSpace = 0,
          int rotation = 0,
          int reverse = 0,
          int bold = 0,
          int italic = 0,
          int rightToLeft = 0,
          int alignment = 0
        }) async
```

[Parameters]

• data: Data for characters to be printed
• xPosition: In dot unit, x axis coordinates of the character string
• yPosition: In dot unit, y axis coordinates of the character string
• fontType: Font type

‘U’	: ASCII (1Byte code)
‘K’	: Korean, KS5601 (2Byte code)
‘B’	: Chinese, BG5 (2Byte code)
‘G’	: Chinese, GB2312 (2Byte code)
‘J’	: Japanese, Shift-JIS (2Byte code)
‘a’	: OCR-A (1Byte code)
‘b’	: OCR-B (1Byte code)

• fontWidth: In dot unit, character width
• fontHeight: In dot unit, character height
• rightSpace: In dot unit, right space of character
• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

• reverse: Reverse or not

0	: No black and white reverse printing
1	: Black and white reverse printing

• bold: Bold or not

0	: Emphasized mode off
1	: Emphasized mode on

• italic: Italic or not

0	: No italic
1	: Print characters with italic

• rightToLeft: Direction of printing characters

0	: Left to right
1	: Right to left

• alignment: Alignment

0	: Align to the left
1	: Align to the right
2	: Align at the center

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-6 drawBarcode1D

Saves data of the 1d barcode to the printer buffer.

[Syntax]

```
Future<int?> drawBarcode1D(String data,
        int xPosition,
        int yPosition,
        int barcodeType,
        int widthNarrow,
        int widthWide,
        int height,
        int hri,
        {
          int rotation = 0,
          int quietZoneWidth = 0
        }) async
```

[Parameters]

• data: Data of the barcode
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• barcodeType: Barcode type

0	: CODE39
1	: CODE128
2	: Interleaved 2of 5
3	: CODABAR
4	: CODE93
5	: UPC-A
6	: UPC-E
7	: EAN13
8	: EAN8
9	: UCC/EAN128
10	: CODE11
11	: PLANET
12	: Industrial 2 of 5
13	: Standard 2 of 5
14	: LOGMARS
15	: UPC/EAN Extensions
16	: POSTNET

• widthNarrow: In dot unit, narrow bar width
• widthWide: In dot unit, wide bar width
• height: In dot unit, barcode height
• hri: HRI (Human Readable Interface) printing position

0	: No HRI print
1	: above barcode
2	: below barcode
3	: above barcode (size: 2)
4	: below barcode (size: 2)
5	: above barcode (size: 3)
6	: below barcode (size: 3)
7	: above barcode (size: 4)
8	: below barcode (size: 4)

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

• quietZoneWidth: Quiet zone width

0 ≤ quietZoneWidth ≤ 20

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-7 drawBarcodeMaxiCode

Saves data of the Maxicode to the printer buffer.

[Syntax]

Future<int?> **drawBarcodeMaxiCode**(String data,
    int xPosition,
    int yPosition,
    int mode) async

[Parameters]

• data: Data of MaxiCode
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• mode: Maxicode mode

0	: MaxiCode Mode 0
2	: MaxiCode Mode 2
3	: MaxiCode Mode 3
4	: MaxiCode Mode 4

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-8 drawBarcodePDF417

Saves data of the PDF417 to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodePDF417(String data,
        int xPosition,
        int yPosition,
        int maximumRowCount,
        int maximumColumnCount,
        int eccLevel,
        int dataCompressionMethod,
        int hri,
        int originPoint,
        int moduleWidth,
        int barHeight,
        {int rotation = 0}) async
```

[Parameters]

• data: Data of PDF417
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• maximumRowCount: Number of barcode rows

3 ≤ maximumRowCount ≤ 90

• maximumColumnCount: Number of barcode columns

1 ≤ maximumColumnCount ≤ 30

• eccLevel: Error correction level

0	: Error correction level 0
1	: Error correction level 1
2	: Error correction level 2
3	: Error correction level 3
4	: Error correction level 4
5	: Error correction level 5
6	: Error correction level 6
7	: Error correction level 7
8	: Error correction level 8

• dataCompressionMethod: Data compression method

0	: 2 characters per codeword
1	: 2.93 characters per codeword
2	: 1.2 characters per codeword

• hri: HRI (Human Readable Interface) printing position

0	: No HRI print
2	: Below

• originPoint: starting point of a barcode

0	: Coordinate value based on middle of barcode
1	: Coordinate value based on top left of barcode

• moduleWidth: Width of the barcode module

2 ≤ moduleWidth ≤ 9

• barHeight: Height of the bar

4 ≤ barHeight ≤ 99

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-9 drawBarcodeQRCode

Saves data of the QRCode to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeQRCode(String data,
        int xPosition,
        int yPosition,
        int size,
        int model,
        int eccLevel,
        {int rotation = 0}) async
```

[Parameters]

• data: Data of QRCode
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• size: QRCode size

1 ≤ size ≤ 9

• model: QRCode model

1	: Model 1
2	: Model 2

• eccLevel: Error correction level

48	: Error correction rate: 7%
49	: Error correction rate: 15%
50	: Error correction rate: 25%
51	: Error correction rate: 35%

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-10 drawBarcodeDataMatrix

Saves data of the DataMatrix to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeDataMatrix(String data,
        int xPosition,
        int yPosition,
        int size,
        {
          int reverse = 0,
          int rotation = 0
        }) async
```

[Parameters]

• data: Data of DataMatrix
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• size: size

1 ≤ size ≤ 4

• reverse: Reverse or normal
• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-11 drawBarcodeAztec

Saves data of the Aztec to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeAztec(String data,
        int xPosition,
        int yPosition,
        int size,
        int extendedChannel,
        int eccLevel,
        int menuSymbol,
        int numberOfSymbols,
        String optionalID,
        { int rotation = 0}) async
```

[Parameters]

• data: Data of Aztec
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• size: Aztec size

1 ≤ size ≤ 9

• extendedChannel: Extended channel interpretation code

0	: Disable
1	: Enable

• eccLevel: Error correction level

0: Automatically configures the error correction level.
1 to 99: Directly enter the error correction level.
101 to 104: 1~4 layer compact symbol
201 to 232: 1~32 layer compact symbol
300: Simple Aztec ”Rune”

• menuSymbol: Enable or disable the menu symbol

0	: Disable
1	: Enable

• numberOfSymbols: The number of symbols

1 ≤ numberOfSymbols ≤ 26

• optionalID: Optional ID field (Maximum 24 letters)

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-12 drawBarcodeCode49

Saves data of the Code49 to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeCode49(String data,
        int xPosition,
        int yPosition,
        int widthNarrow,
        int widthWide,
        int height,
        int hri,
        int startingMode,
        { int rotation = 0 }) async
```

[Parameters]

• data: Data of the barcode
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• widthNarrow: Width of narrow Bar
• widthWide: Width of wide Bar
• height: height
• hri: HRI (Human Readable Interface) printing position

0	: No HRI print
1	: Above
2	: Below

• startingMode: Starting mode of a barcode

0	: Regular Alphanumeric Mode
1	: Multiple Read Alphanumeric
2	: Regular Numeric Mode
3	: Group Alphanumeric Mode
4	: Regular Alphanumeric Shift 1
5	: Regular Alphanumeric Shift 2
7	: Automatic Mode

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-13 drawBarcodeCodaBlock

Saves data of the CodaBlock to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeCodaBlock(String data,
        int xPosition,
        int yPosition,
        int widthNarrow,
        int widthWide,
        int height,
        int securityLevel,
        int dataColumns,
        String mode,
        int rowsToEncode) async
```

[Parameters]

• data: Data of CodaBlock
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• widthNarrow: Width of narrow Bar
• widthWide: Width of wide Bar
• height: height
• securityLevel: Barcode security level

0	: Disable
1	: Enable

• numberOfCharactersPerRow: Number of characters per row

2 ≤ numberOfCharactersPerRow ≤ 62

• mode: CodaBlock mode

'A'	: Creates a code that uses the Code 39 character set
'E'	: Creates a code that uses the Code 128 character set
'F'	: Creates a code that uses the Code 128 character set and that automatically has func1 added

• rowsToEncode

```
1 to 18 : Codablock mode A
2 to 4 : Codablock mode E
2 to 4 : Codablock mode F
```

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-14 drawBarcodeMicroPDF

Saves data of the MicroPDF to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeMicroPDF(String data,
        int xPosition,
        int yPosition,
        int moduleWidth,
        int height,
        int mode,
        { int rotation = 0 }) async
```

[Parameters]

• data: Data of MicroPDF
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• moduleWidth: Module width

2 ≤ moduleWidth ≤ 8

• height: height

1 ≤ height ≤ 99

• mode: Barcode mode

0 ≤ mode ≤ 33

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-15 drawBarcodeIMB

Saves data of the IMB to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeIMB(String data,
        int xPosition,
        int yPosition,
        int hri,
        { int rotation = 0 }) async
```

[Parameters]

• data: Data of IMB
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• hri: HRI (Human Readable Interface) printing position

0	: No HRI Print
1	: Print below barcode

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-16 drawBarcodeMSI

Saves data of the MSI to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeMSI(String data,
        int xPosition,
        int yPosition,
        int widthNarrow,
        int widthWide,
        int height,
        int checkDigit,
        int printCheckDigit,
        int hri,
        { int rotation = 0 }) async
```

[Parameters]

• data: data of MSI
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• widthNarrow: Width of narrow Bar
• widthWide: Width of wide Bar
• height: height
• checkDigit: Check digit mode

0	: Check digit not configured
1	: 1 Mod 10
2	: 2 Mode 10
3	: 1 Mod 11 and Mod 10

• printCheckDigit: Print check digit

0	: No print
1	: Prints check digits

• hri: HRI (Human Readable Interface) printing position.

0	: No HRI Print
1	: Above
2	: Below

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-17 drawBarcodePlessey

Saves data of the Plessey to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodePlessey(String data,
        int xPosition,
        int yPosition,
        int widthNarrow,
        int widthWide,
        int height,
        int printCheckDigit,
        int hri,
        { int rotation = 0 }) async
```

[Parameters]

• data: data of Plessey
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• widthNarrow: Width of narrow Bar
• widthWide: Width of wide Bar
• height: height
• printCheckDigit: Print check digit

0	: No print
1	: Prints check digits

• hri: HRI (Human Readable Interface) printing position.

0	: No HRI Print
1	: Above
2	: Below

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-18 drawBarcodeTLC39

Saves data of the TLC39 to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeTLC39(String data,
        int xPosition,
        int yPosition,
        int widthNarrow,
        int widthWide,
        int height,
        int rowHeightOfMicroPDF417,
        int narrowWidthOfMicroPDF417,
        { int rotation = 0 }) async
```

[Parameters]

• data: data of TLC39
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• narrowWidth: Width of narrow Bar
• widthWide: Width of wide Bar
• height: height
• rowHeightOfMicroPDF417: Height of Micro PDF417's row

1 ≤ rowHeightOfMicroPDF417 ≤ 255

• narrowWidthOfMicroPDF417: Width of Micro PDF417's Narrow bar

1 ≤ narrowWidthOfMicroPDF417 ≤ 10

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-19 drawBarcodeRSS

Saves data of the RSS barcode to the printer buffer.

[Syntax]

```
Future<int?> drawBarcodeRSS(String data,
        int xPosition,
        int yPosition,
        int barcodeType,
        int magnification,
        int separatorHeight,
        int height,
        int segmentWidth,
        { int rotation = 0 }) async
```

[Parameters]

- data: data of RSS
- xPosition: In dot unit, x axis coordinates
- yPosition: In dot unit, y axis coordinates
- barcodeType: RSS barcode type

0	: RSS14
1	: RSS14 truncated
2	: RSS14 stacked
3	: RSS14 Stacked omnidirectional
4	: RSS limited
5	: RSS Expanded
6	: RSS UPC A
7	: RSS UPC E
8	: EAN13
9	: EAN 8
10	: EAN128 CC-A/B
11	: EAN128 CC-C

- magnification: Magnification

  1 ≤ magnification ≤ 10

- separatorHeight: Separator height

  1 ≤ separatorHeight ≤ 2

- height: height

  # Applies only to UCC/EAN128 and CC-A/B/C

- segmentWidth: Segment width

  1 ≤ segmentWidth ≤ 22 (Even Number only)

• rotation: Printing direction

0	: No rotated
1	: Rotated 90 degree (clockwise)
2	: Rotated 180 degree
3	: Rotated 270 degree (clockwise)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-20 drawBlock

Saves data of a quadrangle or line to the printer buffer.

[Syntax]

```
Future<int?> drawBlock(int startPosX,
          int startPosY,
          int endPosX,
          int endPosY,
          String option,
          int thickness) async
```

[Parameters]

• startPosX: In dot unit, x axis coordinates of top left
• startPosY: In dot unit, y axis coordinates of top left
• endPosX: In dot unit, x axis coordinates of bottom right
• endPosY: In dot unit, y axis coordinates of bottom right
• option: drawing option

‘O’	: Repeat drawing on places where lines overlap
‘E’	: Do not draw on places where lines overlap
‘D’	: Delete line
‘S’	: Diagonal line
‘B’	: Quadrangle edge

• thickness: Line thickness

  # This only applies if the ‘option’ value is ‘S’ or ‘B’.

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-21 drawCircle

Saves data of a circle to the printer buffer.

[Syntax]

```
Future<int?> drawCircle(int startPosX,
        int startPosY,
        int size,
        int multiplier) async
```

[Parameters]

• startPosX: In dot unit, x axis coordinates of starting point of circle region
• startPosY: In dot unit, y axis coordinates of starting point of circle region
• size: Size of circle

1	: 40 x 40 dots
2	: 56 x 56 dots
3	: 72 x 72 dots
4	: 88 x 88 dots
5	: 104 x 104 dots
6	: 168 x 168 dots

• multiplier: Expands a circle by the scaling unit

1 ≤ multiplier ≤ 4

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-22 drawBase64Image

Saves data of an image to the printer buffer.

[Syntax]

```
Future<int?> drawBase64Image(String base64String,
        int xPosition,
        int yPosition,
        int width,
        {
          int threshold = 128,
          int ditheringType = 1,
          int compressType = 1
        }) async
```

[Parameters]

• base64String: String data in BASE64 format
• xPosition: In dot unit, x axis coordinates
• yPosition: In dot unit, y axis coordinates
• width: Print width
• threshold: Print threshold

0 ≤ threshold ≤ 255

• ditheringType: Whether or not to apply dithering

0	: No dither
1	: Applying dither

• compressType: Whether or not to compress data

0	: No Compress
1	: Algorithm - RLE
2	: Algorithm - LZMA

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-23 drawImageFile

Saves data of the image file specified by the path to the printer buffer.

[Syntax]

```
Future<int?> drawImageFile(String fileName,
        int xPosition,
        int yPosition,
        int width,
        {
          int threshold = 128,
          int ditheringType = 1,
          int compressType = 1
        }) async
```

[Parameters]

- fileName: The path of the image file.
- xPosition: In dot unit, x axis coordinates
- yPosition: In dot unit, y axis coordinates
- width: Print width
- threshold: Print threshold

0 ≤ threshold ≤ 255

- ditheringType: Whether or not to apply dithering

0	: No dither
1	: Applying dither

- compressType: Whether or not to compress data

0	: No Compress
1	: Algorithm - RLE
2	: Algorithm - LZMA

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-24 drawPDFFile

Saves data of the PDF file specified by the path to the printer buffer.

[Syntax]

```
Future<int?> drawPDFFile(String fileName,
             int xPosition,
             int yPosition,
            int width,
            int pageNumber,
            {
              int threshold = 128,
              int ditheringType = 0,
              int compressType = 1,
            }) async
```

[Parameters]

• fileName: Path of the PDF file
• width: Print width
• pageNumber: A page number to print

0 ≤ pageNumber < The number of pages

• threshold: Print threshold

0 ≤ threshold ≤ 255

• ditheringType: Whether or not to apply dithering

0: No dither
1: Applying dither

• compressType: Data compression method

0: No Compress
1: Algorithm - RLE
2: Algorithm - LZMA

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-25 setPrintingType

Sets the printing type of the printer.

[Syntax]
Future<int?> **setPrintingType**(String printingType) async

[Parameters]
• printingType: Printing type

'd'	: Direct Thermal
't'	: Transfer Thermal

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-26 setMargin

Sets the margins in the printing area.

[Syntax]
Future<int?> **setMargin**(int horizontalMargin, int verticalMargin) async

[Parameters]
• horizontalMargin: In dot unit, left margin
• verticalMargin: In dot unit, top margin

[Returns]
If the method succeeds, the return value is zero(0). Other cases, returns non-zero.

6-2-27 setLength

Sets the label length, gap, media type, and offset.

[Syntax]

```
Future<int?> setLength(int labelLength,
        int gapLength,
        String mediaType,
        int offsetLength) async
```

[Parameters]

• labelLength: In dot unit, media length
• gapLength: In dot unit, gap length of the media
• mediaType: Media type

‘G’	: Gap paper
‘C’	: Continuous paper
‘B’	: Black mark paper

• offsetLength: In dot unit, offset length

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-28 setWidth

Sets the print width.

[Syntax]
Future<int?> **setWidth**(int labelWidth) async

[Parameters]
• labelWidth: In dot unit, print width

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-29 setSpeed

Sets the printing speed.

[Syntax]
Future<int?> **setSpeed**(int speed) async

[Parameters]
• speed: Printing speed

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-30 setDensity

Sets the print density.

[Syntax]
Future<int?> **setDensity**(int density) async

[Parameters]
• density: Print density

  1 ≤ density ≤ 20

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-31 setOrientation

Sets the printing orientation of the label.

[Syntax]
Future<int?> **setOrientation**(String orientation) async

[Parameters]
• orientation: Printing orientation

'T'	: Print from top to bottom
'B'	: Print from bottom to top

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-32 setOffset

Sets offset length between black mark(or gap) and dotted lines.

[Syntax]
Future<int?> **setOffset**(int length) async

[Parameters]
• length: In dot unit, offset length

-100 ≤ length ≤ 100

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-33 setCuttingPosition

Sets the cutting position or tear-off position after printing labels.

[Syntax]
Future<int?> **setCuttingPosition**(int length) async

[Parameters]
• position: In dot unit, cutting position adjustment length
  $-100 \le \text{length} \le 100$

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-34 setAutoCutter

Sets whether to use auto cutter.

This method only works if an auto cutter is equipped in the printer.

[Syntax]
Future<int?> **setAutoCutter**(int enableAutoCutter, int cuttingPeriod) async

[Parameters]
• enableAutoCutter: Whether or not to use auto cutter

0	: Disable
1	: Enable auto cutter

• cuttingPeriod: Cutting cycle

# When set to 2, it will cut every second paper.

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-35 setRewinder

Sets whether to use the printer’s rewinder.

[Syntax]
Future<int?> **setRewinder**(int enableRewinder) async

[Parameters]
• enableRewinder: Whether or not to use the printer’s rewinder

0	: Disable
1	: Enable printer’s rewinder

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-36 getModelName

Gets the printer’s model name.

[Syntax]
Future<String?> **getModelName**() async

[Parameters]
None

[Returns]
An instance of Future<String?> representing the model name.

6-2-37 getFirmwareVersion

Gets the printer’s firmware version.

[Syntax]
Future<String?> **getFirmwareVersion**() async

[Parameters]
None

[Returns]
An instance of Future<String?> representing the firmware version.

5-2-38 getStatisticsData

Gets the printer's statistical data information.

[Syntax]
Future<String?> **getStatisticsData**(int info) async

[Parameters]
• info: Statistical data type

0:	Motor
2:	TPH
4:	Cutter

[Returns]
An instance of Future<String?> representing the requested statistical data information.

5-2-39 getMaxWidth

Gets the maximum printable width value in dot unit.

[Syntax]
Future<int?> **getMaxWidth**() async

[Parameters]
None

[Returns]
An instance of Future<int?> representing the maximum printable width in dot unit.

5-2-40 getPrinterDPI

Gets the printer DPI (dots per inch) value.

[Syntax]
Future<int?> **getPrinterDPI**() async

[Parameters]
None

[Returns]
An instance of Future<int?> representing the printer DPI (dots per inch).

5-2-41 getSupportedSpeeds

Gets the supported print speeds.

[Syntax]
Future<List<int>?> **getSupportedSpeeds**() async

[Parameters]
None

[Returns]
A Future<List<int>?> instance representing a list of supported print speeds.

6-2-42 setupRFID

Set the RFID transponder type, the number of retry codings, the number of retry labels, and the signal strength.

[Syntax]

```
Future<int?> setupRFID(int rfidType,
        int numberOfRetries,
        int numberOfLabels,
        int radioPower) async
```

[Parameters]

• rfidType: RFID transponder type

0	: None
1	: ISO 18000-6 Type A
2	: ISO 18000-6 Type B
3	: EPC Class 0
4	: EPC Class 1
5	: EPC Class 1 Generation 2

• numberOfRetries: Number of retries when coding fails

1 ≤ numberOfRetries ≤ 10

• numberOfLabel: Number of retry labels when coding fails

1 ≤ numberOfLabel ≤ 5

• radioPower: Signal strength

0 ≤ radioPower ≤ 30

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-43 calibrateRFID

The optimal coding position of the RFID label is calculated and stored in the printer.

[Syntax]
Future<int?> **calibrateRFID**() async

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-44 setRFIDPosition

Set the coding position directly on the RFID label.

[Syntax]
Future<int?> **setRFIDPosition**(int transPosition) async

[Parameters]
• transPosition: Coding position of RFID label (y-axis value)

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-45 setRFIDPassword

Set RFID Access Password and Kill Password.

[Syntax]

Future<int?> **setRFIDPassword**(String oldAccessPwd,
    String oldKillPwd,
    String newAccessPwd,
    String newKillPwd) async

[Parameters]

• oldAccessPwd: Access Password currently in use
• oldKillPwd: Kill Password currently in use
• newAccessPwd: Access Password to be modified
• newKillPwd: Kill Password to modify

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-46 setEPCDataStructure

Define the EPC data structure for writing EPC data.

[Syntax]
Future<int?> **setEPCDataStructure**(int totalSize, String fieldSizes) async

Example)
 # Total bits : 64 bits
 # Each field bits: 2 bits, 3 bits, 14 bits, 20 bits, 25 bits
"2,3,14,20,25"

[Parameters]
• totalSize: Total bits in the field
• fieldSizes: Bits of each field (Comma (',') is used for the separator)

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-47 writeRFID

Send data to RFID label tags.

[Syntax]

Future<int?> **writeRFID**(int dataType,
    int startBlockNumber,
    int dataLength,
    String data) async

Must be used after setEPCDataStructure.

[Parameters]

• dataType: RFID data type

65	: ASCII
72	: HEX
69	: EPC
85	: HEX (User area)

• startingBlockNumber: Block to start writing

4 ≤ startingBlockNumber ≤ 10

• dataLength: Number of bytes to write
• data: Data (Input according to dataType)

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

6-2-48 lockRFID

Lock Kill / Access / EPC Data using Access Password.

Be sure to use it after calling setRFIDPassword.

[Syntax]
Future<int?> **lockRFID**() async

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

7. MPosControllerConfig API

7-1 Overview

• The “MPosControllerConfig” is a class which provides methods for controlling peripherals connected with B-gate.

7-2 Methods

7-2-1 searchDevices

Returns a list of device identifiers connected to the B-gate as a string in JSON format.

[Syntax]
Future<String?> **searchDevices**() async

[Parameters]
None

[Returns]
An instance of Future<String?> representing a JSON string for the ID list information of the device connected to the B-gate.

7-2-2 getBgateSerialNumber

Gets B-gate’s serial number.

[Syntax]
Future<String?> **getBgateSerialNumber**() async

[Parameters]
None

[Returns]
An instance of Future<String?> representing B-gate serial number.

7-2-3 getUSBDevice

Get the USB VID&PID corresponding to the device identifier ‘deviceId’.

[Syntax]
Future<String?> **getUSBDevice**(int deviceId) async

[Parameters]
• deviceId: Device ID number

[Returns]
An instance of Future<List<String?>> representing the USB VID&PID corresponding to the device identifier ‘deviceId’.

7-2-4 getCustomDevices

Get the list of USB VID&PID corresponding to the device type.

[Syntax]

Future<List<String>?> **getCustomDevices**(int deviceType) async

[Parameters]

• deviceType:Device Type

1	: Periperal: Bixolon Label Printer
10	: Periperal: Bixolon Receipt Printer
30	: Periperal: MSR
40	: Periperal: Barcode Scanner
60	: Periperal: RFID Reaer
70	: Periperal: Dallas Key
80	: Periperal: NFC Reader
110	: Periperal: Customer Display
120	: Periperal: USB-Serial Device
130	: Periperal: Scale

[Returns]

An instance of Future<List<String?>> representing a list of USB VID&PIDs corresponding to device type.

7-2-5 addCustomDevice

Adds a custom device to B-gate.

[Syntax]
Future<int?> **addCustomDevice**(int deviceType, String vid, String pid) async

[Parameters]
• deviceType:Device Type

1	: Periperal: Bixolon Label Printer
10	: Periperal: Bixolon Receipt Printer
30	: Periperal: MSR
40	: Periperal: Barcode Scanner
60	: Periperal: RFID Reaer
70	: Periperal: Dallas Key
80	: Periperal: NFC Reader
110	: Periperal: Customer Display
120	: Periperal: USB-Serial Device
130	: Periperal: Scale

• VID: String to USB Vender ID
• PID: String to USB Product ID

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

7-2-6 deleteCustomDevice

Deletes a custom device from B-gate.

[Syntax]
Future<int?> **deleteCustomDevice**(int deviceType, String vid, String pid) async

[Parameters]
• deviceType:Device Type

1	: Periperal: Bixolon Label Printer
10	: Periperal: Bixolon Receipt Printer
30	: Periperal: MSR
40	: Periperal: Barcode Scanner
60	: Periperal: RFID Reaer
70	: Periperal: Dallas Key
80	: Periperal: NFC Reader
110	: Periperal: Customer Display
120	: Periperal: USB-Serial Device
130	: Periperal: Scale

• VID: String to USB Vender ID
• PID: String to USB Product ID

[Returns]
An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

7-2-7 reInitCustomDeviceType

Initializes all devices for the specified device type from B-gate

[Syntax]

Future<int?> **reInitCustomDeviceType**(int deviceType) async

[Parameters]

• deviceType:Device Type

1	: Periperal: Bixolon Label Printer
10	: Periperal: Bixolon Receipt Printer
30	: Periperal: MSR
40	: Periperal: Barcode Scanner
60	: Periperal: RFID Reaer
70	: Periperal: Dallas Key
80	: Periperal: NFC Reader
110	: Periperal: Customer Display
120	: Periperal: USB-Serial Device
130	: Periperal: Scale

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

7-2-8 getSerialConfiguration

Gets the information of USB-Serial configuration for the requested device ID.

[Syntax]
Future<List<int>?> **getSerialConfiguration**(int deviceId) async

[Parameters]
• deviceId: Device ID for USB-Serial device

[Returns]
A Future<List<int>?> instance representing SERIAL communication information for the specified device.

# index 0, Baud rate:
0: 2400
1: 4800
2: 9600
3: 19200
4: 38400
5: 57600
6: 115,200
7: 230,400

# index 1, data bit
0: 7 bits
1: 8 bits

# index 2, stop bit
0: 1 bit
1: 2 bits

# index 3, parity bit
0: None
1: Even
2: Odd

# index 3, flow control
0: Hardware
1: None

7-2-9 setSerialConfiguration

Sets the USB-Serial configuration for the requested device ID.

[Syntax]

Future<List<int>?> **setSerialConfiguration**(int deviceId,
    int baudRate,
    int dataBit,
    int stopBit,
    int parityBit,
    int flowControl) async

[Parameters]

• deviceId: Device ID for USB-Serial device

• baudRate: Baud rate

0:	2400
1:	4800
2:	9600
3:	19200
4:	38400
5:	57600
6:	115,200
7:	230,400

• dataBit: Data bit

0:	7 bits
1:	8 bits

• stopBit: Stop bit

0:	1 bit
1:	2 bits

• parityBit: Parity Bit

0:	None
1:	Even
2:	Odd

• flowControl: Flow control method

0:	Hardware
1:	None

[Returns]

An instance of Future<int?> representing the result of the method execution. A value of 0 on success and a non-zero value on failure is passed as the execution result value.

8. MPosLookup API

8-1 Overview

• The “MPosLookup” is a class which provides methods for discovering devices able to connect.

8-2 Methods

8-2-1 getNetworkDevices

Returns a list of connectable network devices as a JSON-formatted string.

[Syntax]
static Future<String?> **getNetworkDevices**() async

[Parameters]
None

[Returns]
An instance of Future<String?> that representing a JSON string of connectable device list information.

8-2-2 getBluetoothDevices

Returns a list of connectable Bluetooth devices as a JSON-formatted string.

[Syntax]
static Future<String?> **getBluetoothDevices**() async

[Parameters]
None

[Returns]
An instance of Future<String?> that representing a JSON string of connectable device list information.

8-2-3 getUSBDevices

Returns a list of connectable USB devices as a JSON-formatted string.

[Syntax]
static Future<String?> **getUSBDevices**() async

[Parameters]
None

[Returns]
An instance of Future<String?> that representing a JSON string of connectable device list information.

8-2-4 getBLEDevices

Returns a list of connectable BLE devices as a JSON-formatted string.

[Syntax]
static Future<String?> **getBLEDevices**() async

[Parameters]
None

[Returns]
An instance of Future<String?> that representing a JSON string of connectable device list information.

9. Appendix

9-1 Error Code Table

• The error code table below summarizes the values returned when calling the API provided in this manual.

Value	Description
0	Suceess of the method Operation
1	Already Device Open
1000	Failure of the method Operation
1001	Not Supported Communication Interface Type
1002	Tried to access the device which is not open
1003	Not supported API by the device
1004	Invalid parameter for the method
1005	No response from the device
1006	Failed to connect the device
1008	No file in the the path specified
1012	Not supported code page
1013	Not supported code international code page
1014	Already entered in page mode.
1015	Already entered in transactoin mode.
1017	Not supported escape sequence
5000	No device found
5001	Failed to get permission required
5003	Failed to decode image in BASE64

Revision history

Rev.	Date	Description
1.00	2022-08-19	New