

# Printer Browser プログラミングマニュアル

2023 年 11 月 1 日 Rev1.0.0

このマニュアルは、Printer Browser に関するお客様が WEB アプリケーションを構築する上で必要なプログラミング情報を示しています。



本書の改定履歴

| 改定    | 日付              | アプリ Ver  | 改定内容   |
|-------|-----------------|----------|--------|
| 1.0.0 | 2023 年 11 月 1 日 | Rev 1.00 | 初版リリース |
|       |                 |          |        |

## ご注意

- Printer Browser は三栄電機株式会社(以下三栄電機といいます。)の著作物であり、本製品にかかる著作権その他の権利は三栄電機に帰属する。
- 三栄電機は Printer Browser に関して欠陥がないこと、このマニュアルに記載されている情報の使用に起因するいかなる損害に対しても責任を負うものではありません。
- 三栄電機は Printer Browser の使用に関連して生じる直接的または、間接的な損失、損害などについていかなる場合も一切責任を負わないものとする。

三栄電機株式会社

無断転載を禁じます。

本書の内容は断り無く変更することがあります。

Android は、Google Inc.の商標です。

# 目次

|  |    |
|--|----|
| 1. はじめに .....                            | 1  |
| 2. システム要件 .....                          | 1  |
| 3. 適用機種 .....                            | 1  |
| 4. API について .....                        | 2  |
| 5. 関数一覧 .....                            | 3  |
| 5-1. discoverUsbPrinter.....             | 4  |
| 5-2. discoverBlePrinter.....             | 4  |
| 5-3. discoverBthPrinter .....            | 5  |
| 5-4. connectUsbPrinter.....              | 5  |
| 5-5. connectBlePrinter.....              | 6  |
| 5-6. connectBthPrinter .....             | 6  |
| 5-7. connectNetworkPrinter.....          | 7  |
| 5-8. disconnectPrinter .....             | 7  |
| 5-9. isPrinterConnected.....             | 7  |
| 5-10. printString .....                  | 7  |
| 5-11. printString (エンコード指定有り).....       | 8  |
| 5-12. setFontStyle.....                  | 8  |
| 5-13. setFontStyle (アンダーラインの太さ指定有り)..... | 8  |
| 5-14. setFontType .....                  | 8  |
| 5-15. setFontMagnification .....         | 9  |
| 5-16. setFontColor .....                 | 9  |
| 5-17. setFontSmoothing .....             | 9  |
| 5-18. setAlignment.....                  | 9  |
| 5-19. setCodePage.....                   | 10 |
| 5-20. setInternationalChar .....         | 11 |
| 5-21. printBarcode .....                 | 11 |
| 5-22. printQR .....                      | 12 |
| 5-23. printPDF417 .....                  | 12 |
| 5-24. setBarcodeModification.....        | 13 |
| 5-25. printBitmap .....                  | 13 |
| 5-26. printBitmap (位置指定、ディザ指定あり).....    | 13 |
| 5-27. linefeed.....                      | 13 |
| 5-28. paperFeed .....                    | 14 |
| 5-29. printStringAndroidFont.....        | 14 |

|                                 |    |
|---------------------------------|----|
| 5-30. initPrinter .....         | 14 |
| 5-31. startPageMode.....        | 14 |
| 5-32. endPageMode.....          | 15 |
| 5-33. setPageAttribute .....    | 15 |
| 5-34. getStatus.....            | 16 |
| 5-35. outputRawDataByte .....   | 17 |
| 5-36. outputRawDataBase64 ..... | 17 |
| 5-37. inputRawData .....        | 17 |
| 5-38. inputRawData .....        | 17 |

## 1. はじめに

Printer Browser は WEB プログラム (Javascript) から直接弊社製プリンタへ印字を可能にするブラウザアプリです。

本ブラウザアプリを使用するに当たり、サンプルプログラムは、以下の URL に置きました。

<https://www.sanei-elec.co.jp/wp/wp-content/uploads/2021/02/sample.zip>

Javascript を使用した簡単な印字サンプルです。

このサンプルプログラムを参考に、印刷に必要な Javascript の命令構文について本書で解説します。

## 2. システム要件

ランタイムバージョン: 1.8.0 以上

Android OS: バージョン 7.1.1 以上

## 3. 適用機種

| 対応プリンタモデル       |         | F/W バージョン   | 対応インターフェース     | 対応コマンド |
|-----------------|---------|-------------|----------------|--------|
| KIOSK<br>プリンタ   | SK4-21  | V1.02.00 以降 | USB            | MODE-A |
|                 | SK4-31  | V1.02.00 以降 | USB            | MODE-A |
|                 | SK1-2x1 | V2.60.00 以降 | USB            | MODE-A |
|                 | SK1-3x1 | V2.60.00 以降 | USB            | MODE-A |
|                 | SK1-21H | V2.60.00 以降 | USB            | MODE-A |
|                 | SK1-31H | V2.60.00 以降 | USB            | MODE-A |
| Desktop<br>プリンタ | SD3-22  | V1.05.00 以降 | USB            | MODE-A |
|                 | SD3-21  | V1.05.00 以降 | USB            | MODE-A |
| Mobile<br>プリンタ  | SM4-21  | V1.01.00 以降 | USB, BLE, WLAN | MODE-A |
|                 | SM4-31  | V1.01.00 以降 | USB, BLE, WLAN | MODE-A |

### ★ メモリスイッチの設定

全てのプリンタモデルに対するメモリスイッチの設定を以下のとおり設定して使用する。

OFFLINE BUSY = OFF

Act. For Driver = INVALID

## 4. API について

### (1) データの種類

Javascript API で取り扱うデータは、数値、文字列、Boolean、オブジェクトとなります。  
API の種類によって扱い難い文字列は、JSON 形式や Base64 に変換して使用します。

### (2) API の呼び出し

呼び出しの先頭に `Printer.` を付ける事により簡単に API を呼び出す事ができます。  
外部ライブラリを必要とせず、API を呼び出すための準備も必要ありません。

### (3) 戻り値の取得

API から直接戻り値が取得できます。  
戻り値は、主に JSON 形式又は Base64 で返すため変換して使用します。  
Boolean は変換せずに、そのまま使用できます。

## 5. 関数一覧

プリンタデバイスの関数一覧は、以下の通りです。

| カテゴリー | 関数                                  | 機能                                     |
|-------|-------------------------------------|--|
| 接続・切断 | <code>discoverUsbPrinter</code>     | 接続可能なUSBプリンタポート名を取得する。                 |
|       | <code>discoverBlePrinter</code>     | 接続可能なBluetooth(BLE)プリンタポート名を取得する。      |
|       | <code>discoverBthPrinter</code>     | 接続可能なBluetooth(SPP)プリンタポート名を取得する。      |
|       | <code>connectUsbPrinter</code>      | 指定のUSBプリンタポートに接続する。                    |
|       | <code>connectBlePrinter</code>      | 指定の Bluetooth(BLE)プリンタポートに接続する。        |
|       | <code>connectBthePrinter</code>     | 指定の Bluetooth(SPP)プリンタポートに接続する。        |
|       | <code>connectNetworkPrinter</code>  | 指定のネットワークプリンタポートに接続する。                 |
|       | <code>disconnectPrinter</code>      | 接続されているプリンタポートを切断する。                   |
|       | <code>isPrinterConnected</code>     | プリンタポートの接続を確認する。                       |
| 標準印字  | <code>printString</code>            | 文字列を指定してプリンタフォントで印字する。                 |
|       | <code>setFontStyle</code>           | プリンタフォントに対する文字修飾を設定する。                 |
|       | <code>setFontType</code>            | プリンタフォントのタイプ(フォントA又はフォントB)を設定する。       |
|       | <code>setFontMagnification</code>   | プリンタフォントのサイズ(縦倍数と横倍数)を設定する。            |
|       | <code>setFontColor</code>           | プリンタフォントに対する白黒反転を指定または解除する。            |
|       | <code>setFontSmoothing</code>       | プリンタフォントに対するスムージング処理を指定または解除する。        |
|       | <code>setAlignment</code>           | 印字データに対する印刷位置((左揃い、センタリング、右揃い)を指定する。   |
|       | <code>setCodePage</code>            | プリンタフォントに対するコードページを指定する。               |
|       | <code>setInternationalChar</code>   | プリンタフォントに対する国際キャラクタ文字を指定する。            |
|       | <code>printBarcode</code>           | 1次元バーコードを印字する。                         |
|       | <code>printQR</code>                | QRコードを印字する。                            |
|       | <code>printPDF417</code>            | PDF417を印字する。                           |
|       | <code>setBarcodeModification</code> | 1次元バーコードの修飾情報(高さ、HRI文字)を設定する。          |
|       | <code>printBitmap</code>            | ビットマップを印字する。                           |
|       | <code>lineFeed</code>               | 指定された回数だけ改行を実行する。                      |
|       | <code>paperFeed</code>              | 指定されたドットライン数だけ紙送りを実行する。                |
|       | <code>printStringAndroidFont</code> | 指定された文字列をAndroidフォントに変換し、ビットイメージで印字する。 |
|       | <code>initPrinter</code>            | プリンタにセットされている修飾情報を初期化する。               |
| ページ印字 | <code>startPageMode</code>          | 標準モードからページモードに移行する。                    |
|       | <code>endPageMode</code>            | ページメモリを印字し、ページモードから標準モードに移行する。         |
|       | <code>setPageAttribute</code>       | ページモードの属性情報(印字方向、印字領域)を設定する。           |
| ステータス | <code>getStatus</code>              | プリンタのステータスを取得する。                       |
| バイナリー | <code>outputRawDataByte</code>      | バイナリーデータ(Byte)をプリンタポートに送信する。           |
|       | <code>outputRawDataBase64</code>    | バイナリーデータ(Base64)をプリンタポートに送信する。         |
|       | <code>inputRawdata</code>           | プリンタポートからバイナリーデータを取得する。                |



## 5-1. discoverUsbPrinter

接続可能な USB プリンタポート名を取得する。

関数: Printer.discoverUsbPrinter(timeoutMillis)  
引数: timeoutMillis      タイムアウト時間(ミリ秒)  
戻り値: JSON 形式の USB プリンタポート名(配列)  
備考: 取得した USB プリンタポート名は、「JSON.parse」で元の形に戻して使用する。

例:      //USB プリンタポート名を取得  
         var result = Printer.discoverUsbPrinter(0);  
         //JSON 形式の USB プリンタポート名を元の形式に変換する  
         var printers = JSON.parse(result);  
         //一番最初の USB プリンタポート名を取得する  
         var first\_printer = printers[0];

## 5-2. discoverBlePrinter

接続可能な Bluetooth(BLE)プリンタポート名を取得する。

関数: Printer.discoverBlePrinter(timeoutMillis)  
引数: timeoutMillis      タイムアウト時間(ミリ秒)  
戻り値: JSON 形式の Bluetooth(BLE)プリンタポート名(配列)  
備考: 取得した Bluetooth(BLE)プリンタポート名は、「JSON.parse」で元の形に戻して使用する。

例:      // Bluetooth(BLE)プリンタポート名を取得  
         var result = Printer.discoverBlePrinter(0);  
         //JSON 形式の Bluetooth(BLE)プリンタポート名を元の形式に変換する  
         var printers = JSON.parse(result);  
         //一番最初の Bluetooth(BLE)プリンタポート名を取得する  
         var first\_printer = printers[0];

### 5-3. discoverBthPrinter

接続可能な Bluetooth(SPP)プリンタポート名を取得する。

関数: Printer.discoverBthPrinter(timeoutMillis)  
引数: timeoutMillis      タイムアウト時間(ミリ秒)  
戻り値: JSON 形式の Bluetooth(SPP)プリンタポート名(配列)  
備考: 取得した Bluetooth(SPP)プリンタポート名は、「JSON.parse」で元の形に戻して使用する。

例:      // Bluetooth(SPP)プリンタポート名を取得  
         var result = Printer.discoverBthPrinter(0);  
         //JSON 形式の Bluetooth(SPP)プリンタポート名を元の形式に変換する  
         var printers = JSON.parse(result);  
         //一番最初の Bluetooth(SPP)プリンタポート名を取得する  
         var first\_printer = printers[0];

### 5-4. connectUsbPrinter

指定の USB プリンタポートに接続する。

関数: Printer.connectUsbPrinter(printer)  
引数: printer      JSON 形式の USB プリンタポート名  
戻り値: 接続の成功または失敗を Boolean で返す。  
         true      接続成功  
         false     接続失敗  
備考: USB プリンタポート名は、「JSON.stringify」で JSON 形式にして使用する。

例:      //USB プリンタポート名を JSON 形式に変換  
         var jprinter = JSON.stringify(printer);  
         //USB プリンタポートに接続  
         var status = Printer.connectUsbPrinter(jprinter)

## 5-5. connectBlePrinter

指定の Bluetooth(BLE)プリンタポートに接続する。

関数: `Printer.connectBlePrinter(printer)`

引数: `printer`      JSON 形式の Bluetooth(BLE)プリンタポート名

戻り値: 接続の成功または失敗を Boolean で返す。

`true`      接続成功

`false`      接続失敗

備考: Bluetooth(BLE)プリンタポート名は、「JSON.stringify」で JSON 形式にして使用する。

```
例: // Bluetooth(BLE)プリンタポート名を JSON 形式に変換
    var jprinter = JSON.stringify(printer);
    // Bluetooth(BLE)プリンタポートに接続
    var status = Printer.connectBlePrinter(jprinter)
```

## 5-6. connectBthPrinter

指定の Bluetooth(SPP)プリンタポートに接続する。

関数: `Printer.connectBthPrinter(printer)`

引数: `printer`      JSON 形式の Bluetooth(SPP)プリンタポート名

戻り値: 接続の成功または失敗を Boolean で返す。

`true`      接続成功

`false`      接続失敗

備考: Bluetooth(SPP)プリンタポート名は、「JSON.stringify」で JSON 形式にして使用する。

```
例: // Bluetooth(SPP)プリンタポート名を JSON 形式に変換
    var jprinter = JSON.stringify(printer);
    // Bluetooth(SPP)プリンタポートに接続
    var status = Printer.connectBthPrinter(jprinter)
```

### 5-7. connectNetworkPrinter

指定のネットワークプリンタポートに接続する。

関数: Printer.connectNetworkPrinter(address,port)

引数: address IP アドレス(文字列)

port ポート番号(数値)

戻り値: 接続の成功または失敗を Boolean で返す。

true 接続成功

false 接続失敗

### 5-8. disconnectPrinter

接続されているプリンタポートを切断する。

関数: Printer.disconnectPrinter()

引数: なし

戻り値: なし

### 5-9. isPrinterConnected

プリンタポートの接続を確認する。

関数: Printer.isPrinterConnected()

引数: なし

戻り値: 現在の接続状態を Boolean で返す

true 接続している

false 接続していない

### 5-10. printString

プリンタフォントで文字列を印字する。

関数: Printer.printString(text)

引数: text 印字する文字列をユニコードで指定する

戻り値: なし

### 5-11. printString (エンコード指定有り)

プリンタフォントで文字列をエンコード指定有りで印字する。

関数: `Printer.printString(text,charname)`

引数: `text`           文字列  
      `charname`       文字列のエンコード名

戻り値: なし

備考: `charname` は Java で使用するエンコード名を指定する。  
      例. “UTF-8”、“SJIS”などのエンコード名を文字列で指定する。

### 5-12. setFontStyle

プリンタフォントの修飾を設定する。

関数: `Printer.setFontStyle(bold,italic,underline)`

引数: `bold`            ボールド書体の指定(true)又は解除(false)  
      `italic`          イタリック書体の指定(true)又は解除(false)  
      `underline`       アンダーラインの指定(true)又は解除(false)

戻り値: なし

### 5-13. setFontStyle (アンダーラインの太さ指定有り)

プリンタフォントの修飾を設定(アンダーラインの太さ指定有り)する。

関数: `Printer.setFontStyle(bold,italic,underline,dotpitch)`

引数: `bold`            ボールド書体の指定(true)又は解除(false)  
      `italic`          イタリック書体の指定(true)又は解除(false)  
      `underline`       アンダーラインの指定(true)又は解除(false)  
      `dotpitch`       アンダーラインの太さ 0~7ドットを指定する。

戻り値: なし

### 5-14. setFontType

プリンタフォントの種類を設定する。

関数: `Printer.setFontType(compact)`

引数: `compact`        スモールフォント(フォントB)の指定(true)又は解除(false)

戻り値: なし

### 5-15. setFontMagnification

プリンタフォントのサイズを設定する。

関数: Printer.setFontMagnification (horizontalRatio, verticalRatio)

引数: horizontalRatio プリンタフォントの横倍数(1~8)

verticalRatio プリンタフォントの縦倍数(1~8)

戻り値: なし

### 5-16. setFontColor

プリンタフォントに対する白黒反転を指定または解除する。

関数: Printer.setFontColor(reverse)

引数: reverse 白黒反転の指定(true)又は解除(false)

戻り値: なし

### 5-17. setFontSmoothing

縦倍又は横倍数の指定されたプリンタフォントに対するスムージング処理を指定または解除する。

関数: Printer.setFontSmoothing(smoothing)

引数: smoothing スムージングの指定(true)又は解除(false)

戻り値: なし

### 5-18. setAlignment

印字データに対する印刷位置(左揃い、センタリング、右揃い)を指定する。

関数: Printer.setAlignment(alignment)

引数: alignment 印字位置の指定

| alignment |   |
|-----------|---|
| 左揃い       | 0 |
| センタリング    | 1 |
| 右揃い       | 2 |

戻り値: なし

## 5-19. setCodePage

プリンタフォントに対するコードページを指定する。

関数: Printer.setCodePage(codePage)

引数: codepage      コードページの指定

| CodePage  |    |
|-----------|----|
| PC437     | 0  |
| katakana  | 1  |
| PC850     | 2  |
| PC852     | 3  |
| PC857     | 4  |
| PC858     | 5  |
| PC863     | 6  |
| PC865     | 7  |
| PC866     | 8  |
| WPC1252   | 9  |
| PC860     | 10 |
| WPC1252_2 | 11 |
| PC862     | 12 |
| WPC1254   | 13 |
| WPC1250   | 14 |
| WPC1251   | 15 |
| PC864     | 16 |
| PC737     | 18 |
| PC1253    | 20 |

戻り値: なし

## 5-20. setInternationalChar

プリンタフォントに対する国際キャラクタ文字を指定する。

関数: Printer.setInternationalChar(internationalCharset)

引数: internationalCharset 国際文字の指定

| InternationalCharset |   |
|----------------------|---|
| USA                  | 0 |
| France               | 1 |
| Germany              | 2 |
| England              | 3 |
| Denmark              | 4 |
| Sweden               | 5 |
| Italy                | 6 |
| Spain                | 7 |
| Japan                | 8 |

戻り値: なし

## 5-21. printBarcode

1次元バーコードを印字する。

関数: Printer.printBarcode(barcodeSystem,barcodeData)

引数: barcodeSystem バーコードの種類

| BarcodeSystem |    |
|---------------|----|
| UPCA          | 65 |
| UPCE          | 66 |
| JAN13         | 67 |
| JAN8          | 68 |
| CODE39        | 69 |
| ITF           | 70 |
| CODABAR       | 71 |
| CODE93        | 72 |
| CODE128       | 73 |

barcodeData バーコードデータ

戻り値: なし



## 5-22. printQR

QRコードを印字する。

関数: Printer.printQR(size,eccLevel,barcodeData)

引数: size シンボルサイズ(1~40)を指定する。  
eccLevel エラーコントロールレベル(1~4)を指定する。  
1: L (7%)  
2: M (15%)  
3: Q (25%)  
4: H (30%)  
barcodeData バーコードデータ

戻り値: なし

## 5-23. printPDF417

PDF417を印字する。

関数: Printer.printPDF417(truncate,binaryEncode,eccLevel,size,barcodeData)

引数: truncate (コンパクト)PDF417に指定する(true)、指定しない(false)  
binaryEncode バイト符号化モード(true)、自動符号化モード(false)  
eccLevel 誤り訂正レベル(0~7)を指定する。  
size 組み合わせ表の通り、バーコードサイズを指定する。

| size | 詳細 (X:列 / Y=ステップ) | size | 詳細 (X:列 / Y=ステップ) |
|------|-------------------|------|-------------------|
| 0    | X 2: Y 4          | 8    | X 12: Y 4         |
| 1    | X 2: Y 9          | 9    | X 12: Y 9         |
| 2    | X 2: Y 15         | 10   | X 12: Y 15        |
| 3    | X 2: Y 20         | 11   | X 12: Y 20        |
| 4    | X 7: Y 4          | 12   | X 20: Y 4         |
| 5    | X 7: Y 9          | 13   | X 20: Y 9         |
| 6    | X 7: Y 15         | 14   | X 20: Y 15        |
| 7    | X 7: Y 20         | 15   | X 20: Y 20        |

barcodeData バーコードデータ

戻り値: なし

## 5-24. setBarcodeModification

1次元バーコードの修飾情報(高さ、HRI 文字)を設定する。

関数: Printer.setBarcodeModification (hri,width,height)

引数: hri HRI 文字の修飾を指定する。

| hri               |   |
|-------------------|---|
| 文字を印字しない          | 0 |
| 文字をバーコードの上に印字する   | 1 |
| 文字をバーコードの下に印字する   | 2 |
| 文字をバーコードの上と下に印字する | 3 |

Width バーコードのモジュール幅(1~4)を指定する。

Height バーコードの高さ(1~255ドット)を指定する。

戻り値: なし

## 5-25. printBitmap

Base64 文字列で指定されたビットマップデータをビットイメージコマンドに変換して印字する。

関数: Printer.printBitmap(data)

引数: data ビットマップデータ (Base64 文字列)を指定する。

戻り値: なし

## 5-26. printBitmap (位置指定、ディザ指定あり)

Base64 文字列で指定されたビットマップデータをビットイメージコマンドに変換して印字する。

関数: Printer.printBitmap(data,align,dither)

引数: data ビットマップデータ (Base64 文字列)を指定する。

align 印字位置 (Left, Center, Right) を指定する。

dither ディザリングを指定する (true)又は指定しない(false)

戻り値: なし

## 5-27. linefeed

指定された行数の改行を実行する。

関数: Printer.lineFeed(lines)

引数: lines 改行数(1~255)を指定する。

戻り値: なし

## 5-28. paperFeed

指定されたドットライン数の正方向または逆方向に紙送りを実行する。

関数: Printer.paperFeed(lines)

引数: lines            -255～255のドットピッチ値の紙送りを指定する。  
                          パラメータが負数の場合、コマンドは印刷と逆方向紙送りを実行する。  
                          整数の場合、印刷と正方向紙送りを実行する。

戻り値: なし

備考: 標準モードでは、ラインバッファ内のデータを印刷し、指定されたドットピッチだけ前方に紙送り  
      する。ページモードでは、指定したドットピッチだけ y 軸方向に移動する。

## 5-29. printStringAndroidFont

指定された文字列を Android フォントに変換しビットイメージで印字する。

関数: Printer.printStringAndroidFont(text, x, y, textSize)

引数: text            文字列を指定する。  
      x                水平方向の印字位置  
      y                垂直方向の印字位置  
      textSize        文字サイズ(単位:SP)を指定する。

戻り値: なし

備考: SP = スケール非依存ピクセル (Scale-independent pixel)

## 5-30. initPrinter

プリンタにセットされている修飾情報を初期化する。

関数: Printer.initPrinter()

引数: なし

戻り値: なし

## 5-31. startPageMode

標準モードからページモードに移行する。

関数: Printer.startPageMode()

引数: なし

戻り値: なし

### 5-32. endPageMode

ページメモリの指定領域を印字しページモードから標準モードに移行する。

関数: Printer.endPageMode()

引数: なし

戻り値: なし

### 5-33. setPageAttribute

ページモードの指定領域(印字方向、印字領域)を設定する。

関数: Printer.setPageAttribute(x,y,width,height,direction)

引数: x 印字領域 (x 軸) の始点

y 印字領域 (y 軸) の始点

width 印字領域 (x 軸) の横幅

height 印字領域 (y 軸) の高さ

direction 印字領域の展開方向

| direction |   |
|-----------|---|
| 正方向       | 0 |
| 左90度回転    | 1 |
| 逆方向       | 2 |
| 右90度回転    | 3 |

戻り値: なし

備考: 印字領域はページモードを印字するまでの間、何度でも指定ができる。

ページモードを印字する場合、プリンタ内で自動的にその時点の最大印字領域で印字を行う。

印字領域の指定可能範囲は、プリンタモデルとメモリスイッチの設定に依存する。

## 5-34. getStatus

プリンタの状態を取得する。

関数: Printer.getStatus()

引数: なし

戻り値: JSON 形式のステータス

備考: 取得したステータスは、「JSON.parse」で元の形に戻して使用する。  
ステータス値の仕様はコマンドマニュアル GS R1 コマンドを参照とする。

```
例: //ステータスを取得
var result = Printer.getStatus();
//取得した JSON 形式のステータスを元の形式に変換する
var status = JSON.parse(result);

switch (status.status1) {
case 82: //アスキーコードの'R'
    buffer+= "Status1: WAITING<br>";
    break;
case 66: //アスキーコードの'B'
    buffer+= "Status1: MOVING<br>";
    break;
case 69: //アスキーコードの'E'
    buffer+= "Status1: ERROR<br>";
    break;
default:
    buffer+= "Status1: " + status.status1 + "<br>";
}
```

### 5-35. outputRawDataByte

1バイトのバイトデータをプリンタポートに送信する。

関数: Printer.outputRawDataByte(byte)

引数: byte            バイトデータ

戻り値: なし

### 5-36. outputRawDataBase64

Base64 文字列でバイナリーデータをプリンタポートに送信する。

関数: Printer.outputRawDataBase64(base64)

引数: base64        Base64 文字列のバイナリーデータ

戻り値: なし

### 5-37. inputRawData

1バイトのバイトデータをプリンタポートから取得する。

関数: Printer.inputRawData(timeoutMillis)

引数: timeoutMillis    タイムアウト時間(ミリ秒)を指定する。

戻り値: 1バイトのバイトデータ

### 5-38. inputRawData

Base64 文字列でバイナリーデータをプリンタポートから取得する。

関数: Printer.inputRawData(timeoutMillis,size)

引数: timeoutMillis    タイムアウト時間(ミリ秒)を指定する

size                    受信するサイズを指定する

戻り値: Base64 文字列のバイナリーデータ