

CITIZEN

モバイルプリンター

Androidモジュールプログラムマニュアル

CPCL

Ver. 1.07

シチズン・システムズ株式会社

履歴

| バージョン | 対象ライブラリ | 説明 |
|-------|---------|--|
| 1.00 | 1.051 | 公開 |
| 1.01 | 1.052以降 | 日本語印刷のための設定例等の説明を追加 ステータス機能の戻り値をESCPOS SDKのものと別名に修正 関数を追加 setCountry, resetCountry 関数を削除 sendPrinter, setFormMode, setJournalMode |
| 1.02 | 1.055以降 | 実際のSDKとマニュアルでメソッド名が食い違っていたメソッド名を printerCheckに統一(ライブラリ側を修正) printerCheckを使ったパワーセーブモードから復帰に対応 そのための注意書きを追記 ほか細かい修正 関数を追加 temperature, voltage 2.26の関数を差替え printGraphic(削除) printBitmap(追加) |
| 1.03 | 1.064以降 | printBatmapの関数の引数のオプションを追加 関数を追加 getEmulation 削除になった章を除いて章番号を振りなおした。 |
| 1.04 | 1.071以降 | CPCLprinterメソッドに引数connectionを追加 printerResultsメソッドを追加 printAndroidFontメソッドを追加 過去に無効にして取り消し線で消していたメソッドを削除した |
| 1.05 | 1.083以降 | 表紙のロゴを変更 座標指定形式のprintAndroidFontメソッドを追加 サポートするBluetoothモジュールの追加 |
| 1.06 | 1.087以降 | Bluetooth デバイス検索対応のため、ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATIONのパーミッションが追加 電源オフ直後のステータスの不具合の解消のため、printerCheck(), status(), printerResults() の戻り値拡張(互換性問題あり) |
| 1.07 | 1.090以降 | Power Saveモードから瞬時に復帰するCMP-30IIIには、ダミーデータをおくらないようにした。 |

1. はじめに

この「Androidモジュールプログラムマニュアル」では、Androidモバイルアプリケーションの開発に必要なJarパッケージファイルから得られるメソッドについて説明します。

説明はラベル紙をベースに行われております。

ブラックマーク紙を使う場合は、ラベルの長さ(高さ)＝ブラックマーク間の距離などと読み替えてください。

プリンターの仕様につき不明の点は、CMP-30の技術マニュアル、CPCLのコマンドマニュアルを参考にして頂きますよう、お願いいたします。

2. メソッド

CPCLPrinterクラスで定義されています。

定数は、CPCLConst Interfaceに定義されています。

2.1. CPCLPrinter

これは、Constructorメソッドです。CPCLPrinterオブジェクトは、パラメーターを使用して文字セットを選択します。パラメーターを使用しない場合、デフォルトの文字セットはUTF-8となります。

CPCLPrinter() , CPCLPrinter(String charset)

CPCLPrinter(DeviceConnection connection) ,

CPCLPrinter(String charset, DeviceConnection connection

[パラメーター]

* charset

- 文字セット名

* connection

- デバイス接続 (USBPortConnection、WiFiMultiConnection)

使用例: 日本語印刷に対応する場合

CPCLPrinter ("Shift-JIS")

2.2. setForm

この関数は、用紙フォームを定義するために使用されます。

void setForm(int HorizonOffset, int XResol, int YResol, int LabelHeight, int Quantity)

[パラメーター]

* HorizonOffset

- nullで終了する文字列を持つUnicode。ラベル全体の水平方向のオフセットを係数として受け取ります。

これは、左側の印刷の絶対位置の値として考えることができます。

* XResol

- 水平方向の解像度を係数として受け取ります。

* YResol

- 垂直方向の解像度を受け取ります。

* LabelHeight

- 印刷用のラベルの長さの値を係数として受け取ります。(単位ドット)

* Quantity

- 印刷するラベルの枚数を係数として受け取ります。[最大数 =< 1024]

注:必ず、setForm

setMedia

printForm の組み合わせで使用すること

2.3. printForm

この関数は、印刷をし、紙送りをしてフォームの頭出しをするために使用されます。

void printForm()

2.4. printerCheck

この関数は、プリンターのステータスチェック用に使用されます。

int printerCheck()

int printerCheck(int timeout)

[パラメーター]

* timeout: ミリ秒 (デフォルト5000ms)

[戻り値]

CMP_SUCCESS : 関数が適切に処理されると、この値が戻ります。

CMP_FAIL : 関数が適切に処理されなかった場合は、この値が戻ります。

CMP_STS_CPCL_PRINTEROFF : プリンターがオフの場合、この値が戻ります。← 1.087にて仕様追加

CMP_STS_CPCL_TIMEOUT : プリンターから返信がない場合にこの値が戻ります。←1.087にて仕様追加

注: パワーセーブモードからの復帰のためには、printerCheckを実行すること。

2.5. status

この関数は、プリンターのステータスを取得するために使用されます。

int status()

[戻り値]

CMP_STS_CPCL_NORMAL: プリンターはエラーなし状態です。(MSRの場合、MSRは準備できていません)

CMP_STS_CPCL_BUSY : プリンターはビジー状態です。

CMP_STS_CPCL_PAPER_EMPTY : プリンターは用紙なし状態です。

CMP_STS_CPCL_COVER_OPEN : プリンターのカバーが開いた状態です。

CMP_STS_CPCL_BATTERY_LOW : プリンターのバッテリー容量が少なくなっています。

CMP_STS_CPCL_PRINTEROFF : プリンターがオフの場合、この値が戻ります。← 1.087にて仕様追加

CMP_STS_CPCL_TIMEOUT : Tプリンターから返信がない場合にこの値が戻ります。←1.087にて仕様追加

注: status関数の前にprinterCheckを実行すること

2.6. setMeasure

この関数は、コマンドの単位を設定するために使用されます。

```
void setMeasure(int Measure)
```

[パラメーター]

* Measure

- 単位セットコマンドを係数として受け取ります。

| 変数 | 説明 |
|----------------|------------------------|
| CMP_CPCL_INCH | 単位をインチとして設定します。 |
| CMP_CPCL_CENTI | 単位をセンチメートルとして設定します。 |
| CMP_CPCL_MILLI | 単位をミリメートルとして設定します。 |
| CMP_CPCL_DOTS | 単位をドットとして設定します(デフォルト)。 |

2.7. setJustification

この関数は、フィールドの位置揃えを設定するために使用されます。

```
void setJustification (int Justify)
```

[パラメーター]

* Justify

- フィールドの位置揃えを設定するための係数を受け取ります。

| 変数 | 説明 |
|-----------------|------|
| CMP_CPCL_LEFT | 左揃え |
| CMP_CPCL_CENTER | 中央揃え |
| CMP_CPCL_RIGHT | 右揃え |

2.8. printCPCLText

この関数は、フォーム上の指定の場所にテキストを印刷するために使用されます。

```
void printCPCLText(int Rotation,int FontType,int FontSize,int PrintX,int PrintY, String Data,int count)
```

[パラメーター]

* Rotation

- 印刷するテキストの印刷方向の値を係数として受け取ります。

| 変数 | 説明 |
|-----------------------|----------------------------|
| CMP_CPCL_NO_ROTATION | テキストを回転せずに印刷します。 |
| CMP_CPCL_90_ROTATION | テキストを90度回転(反時計方向)して印刷します。 |
| CMP_CPCL_180_ROTATION | テキストを180度回転(反時計方向)して印刷します。 |
| CMP_CPCL_270_ROTATION | テキストを270度回転(反時計方向)して印刷します。 |

* FontType

- 印刷するテキストのフォントの種類を0-7で指定します。半角文字と日本語が混在する場合は7を推奨します。

* FontSize

- 印刷するテキストのフォントのサイズを0-7で指定します。

* PrintX

- 印刷するテキストのx座標の値を係数として受け取ります。

* PrintY

- 印刷するテキストのy座標の値を係数として受け取ります。

* Data

- nullで終了する文字列を持つUnicode。印刷するテキストを係数として受け取ります。

* count

- 印刷するテキストのカウント関数を係数として受け取ります。

注:フォントの種類とサイズについては、CMP-30の技術マニュアルを参照ください。

2.9. setConcat

この関数は、テキスト連結の開始と印刷位置を定義するために使用されます。

```
void setConcat(int ConcatMode,int PrintX,int PrintY)
```

[パラメーター]

* ConcatMode

- 連結するテキストの連結方法を係数として受け取ります。

| 変数 | 説明 |
|------------------|---------------|
| CMP_CPCL_CONCAT | テキストを横に連結します。 |
| CMP_CPCL_VCONCAT | テキストを縦に連結します。 |

* PrintX

- 印刷するテキストのx座標の値を係数として受け取ります。

* PrintY

- 印刷するテキストのy座標の値を係数として受け取ります。

2.10. concatText

この関数は、連結するフォントを定義するために使用されます。

```
void concatText(int FontType,int FontSize,int Offset,String Data)
```

[パラメーター]

* FontType

- 印刷するテキストのフォントの種類を係数として受け取ります。

* FontSize

- 印刷するテキストのフォントのサイズの値を係数として受け取ります。

* Offset

- 開始点の単位値を受け取ります。

* Data

- nullで終了する文字列を持つUnicode。印刷するテキストを係数として受け取ります。

2.11. resetConcat

この関数は、テキスト連結を取り消すために使用されます。

```
void resetConcat()
```

2.12. setMultiLine

この関数は、同じ長さと同じフォントで複数行の印刷を定義するために使用されます。

```
void setMultiLine(int LineHeight)
```

[パラメーター]

* LineHeight

- 印刷するテキストの行の高さの値を係数として受け取ります。

2.13. multiLineText

この関数は、SetMultiLine()として定義されるテキストを印刷するために使用されます。印刷するテキストのフォント、サイズ、および印刷位置を係数として受け取ります。

```
void multiLineText(int Rotation,int FontType,int FontSize,int PrintX,int PrintY)
```

[パラメーター]

* Rotation

- 印刷するテキストの印刷方向の値を係数として受け取ります。

| 変数 | 説明 |
|-----------------------|----------------------------|
| CMP_CPCL_NO_ROTATION | テキストを回転せずに印刷します。 |
| CMP_CPCL_90_ROTATION | テキストを90度回転(反時計方向)して印刷します。 |
| CMP_CPCL_180_ROTATION | テキストを180度回転(反時計方向)して印刷します。 |
| CMP_CPCL_270_ROTATION | テキストを270度回転(反時計方向)して印刷します。 |

* FontType

- 印刷するテキストのフォントの種類を係数として受け取ります。

* FontSize

- 印刷するテキストのフォントのサイズの値を係数として受け取ります。

* PrintX

- 印刷するテキストのx座標の値を係数として受け取ります。

* PrintY

- 印刷するテキストのy座標の値を係数として受け取ります。

* Data

- nullで終了する文字列を持つUnicode。印刷するテキストを係数として受け取ります。

2.14. multiLineData

この関数は、MultiLineText()で定義される値のテキストを印刷するために使用されます。

```
void multiLineData(String Data)
```

[パラメーター]

* Data

- nullで終了する文字列を持つUnicode。印刷するテキストを係数として受け取ります。

2.15. resetMultiLine

この関数は、setMultiLine()で定義された関数を取り消すために使用されます。

```
void resetMultiLine()
```

2.16. setMagnify

この関数は、テキストの表示倍率を設定するために使用されます。

```
void setMagnify(int Width,int Height)
```

[パラメーター]

* Width

- テキストの幅の比率を係数として受け取ります。

| 変数(幅の比率を設定) | 説明 |
|----------------------|---------------|
| CMP_CPCL_TXT_1WIDTH | 幅の比率を1倍として設定 |
| CMP_CPCL_TXT_2WIDTH | 幅の比率を2倍として設定 |
| CMP_CPCL_TXT_3WIDTH | 幅の比率を3倍として設定 |
| CMP_CPCL_TXT_4WIDTH | 幅の比率を4倍として設定 |
| CMP_CPCL_TXT_5WIDTH | 幅の比率を5倍として設定 |
| CMP_CPCL_TXT_6WIDTH | 幅の比率を6倍として設定 |
| CMP_CPCL_TXT_7WIDTH | 幅の比率を7倍として設定 |
| CMP_CPCL_TXT_8WIDTH | 幅の比率を8倍として設定 |
| CMP_CPCL_TXT_9WIDTH | 幅の比率を9倍として設定 |
| CMP_CPCL_TXT_10WIDTH | 幅の比率を10倍として設定 |
| CMP_CPCL_TXT_11WIDTH | 幅の比率を11倍として設定 |
| CMP_CPCL_TXT_12WIDTH | 幅の比率を12倍として設定 |
| CMP_CPCL_TXT_13WIDTH | 幅の比率を13倍として設定 |
| CMP_CPCL_TXT_14WIDTH | 幅の比率を14倍として設定 |
| CMP_CPCL_TXT_15WIDTH | 幅の比率を15倍として設定 |
| CMP_CPCL_TXT_16WIDTH | 幅の比率を16倍として設定 |

* Height

- テキストの高さの比率を係数として受け取ります。

| 変数 (高さの比率を設定) | 説明 |
|-----------------------|----------------|
| CMP_CPCL_TXT_1HEIGHT | 高さの比率を1倍として設定 |
| CMP_CPCL_TXT_2HEIGHT | 高さの比率を2倍として設定 |
| CMP_CPCL_TXT_3HEIGHT | 高さの比率を3倍として設定 |
| CMP_CPCL_TXT_4HEIGHT | 高さの比率を4倍として設定 |
| CMP_CPCL_TXT_5HEIGHT | 高さの比率を5倍として設定 |
| CMP_CPCL_TXT_6HEIGHT | 高さの比率を6倍として設定 |
| CMP_CPCL_TXT_7HEIGHT | 高さの比率を7倍として設定 |
| CMP_CPCL_TXT_8HEIGHT | 高さの比率を8倍として設定 |
| CMP_CPCL_TXT_9HEIGHT | 高さの比率を9倍として設定 |
| CMP_CPCL_TXT_10HEIGHT | 高さの比率を10倍として設定 |
| CMP_CPCL_TXT_11HEIGHT | 高さの比率を11倍として設定 |
| CMP_CPCL_TXT_12HEIGHT | 高さの比率を12倍として設定 |
| CMP_CPCL_TXT_13HEIGHT | 高さの比率を13倍として設定 |
| CMP_CPCL_TXT_14HEIGHT | 高さの比率を14倍として設定 |
| CMP_CPCL_TXT_15HEIGHT | 高さの比率を15倍として設定 |
| CMP_CPCL_TXT_16HEIGHT | 高さの比率を16倍として設定 |

2.17. resetMagnify

この関数は、SetMagnify()で定義された関数を取り消すために使用されます。

```
void resetMagnify()
```

2.18. printCPCLBarCode

この関数は、バーコードを印刷するために使用されます。

```
void printCPCLBarCode(int Rotation,int BarCodeType,int NB,int Ratio,int BarHeight, int PrintX,int PrintY,String Data,int count)
```

[パラメーター]

* Rotation

- 印刷するバーコードの印刷方向の値を係数として受け取ります。

| 変数 | 説明 |
|-----------------------|-----------------------------|
| CMP_CPCL_NO_ROTATION | バーコードを回転せずに印刷します。 |
| CMP_CPCL_90_ROTATION | バーコードを90度回転(反時計方向)して印刷します。 |
| CMP_CPCL_180_ROTATION | バーコードを180度回転(反時計方向)して印刷します。 |
| CMP_CPCL_270_ROTATION | バーコードを270度回転(反時計方向)して印刷します。 |

* BarCodeType

- 印刷するバーコードの種類を係数として受け取ります。

| 変数 | 説明 |
|-----------------------|----------------------------------|
| CMP_CPCL_BCS_39 | バーコード3/9標準 |
| CMP_CPCL_BCS_39C | バーコード3/9標準 チェックデジット付加 |
| CMP_CPCL_BCS_39F | バーコード3/9標準 Full ASCII |
| CMP_CPCL_BCS_39FC | バーコード3/9 Full ASCII チェックデジット付加 |
| CMP_CPCL_BCS_93 | バーコード Code 93 |
| CMP_CPCL_BCS_128 | バーコード Code 128 |
| CMP_CPCL_BCS_EAN128 | バーコード UCC-128 |
| CMP_CPCL_BCS_CODABAR | バーコード Codabar |
| CMP_CPCL_BCS_CODABARC | バーコード Codabar チェックデジット付加 |
| CMP_CPCL_BCS_EAN8 | バーコード EAN8 |
| CMP_CPCL_BCS_EAN82 | バーコード EAN8 2桁アドオン |
| CMP_CPCL_BCS_EAN85 | バーコード EAN8 5桁アドオン |
| CMP_CPCL_BCS_EAN13 | バーコード EAN13 |
| CMP_CPCL_BCS_EAN132 | バーコード EAN13 2桁アドオン |
| CMP_CPCL_BCS_EAN135 | バーコード EAN13 5桁アドオン |
| CMP_CPCL_BCS_I2OF5 | バーコード Interleaved 2 of 5. |
| CMP_CPCL_BCS_POSTNET | バーコード PostNet |
| CMP_CPCL_BCS_UPCA | バーコード UPCA |
| CMP_CPCL_BCS_UPCA2 | バーコード UPCA 2桁アドオン |
| CMP_CPCL_BCS_UPCA5 | バーコード UPCA 5桁アドオン |
| CMP_CPCL_BCS_UPCE | バーコード UPCE |
| CMP_CPCL_BCS_UPCE2 | バーコード UPCE 2桁アドオン |
| CMP_CPCL_BCS_UPCE5 | バーコード UPCE 5桁アドオン |
| CMP_CPCL_BCS_MSI | バーコード Plessey(MSI-1) |
| CMP_CPCL_BCS_MSI1C | バーコード Plessey(MSI-1) チェックデジット付加 |
| CMP_CPCL_BCS_MSI2C | バーコード Plessey(MSI-2) チェックデジット付加 |
| CMP_CPCL_BCS_MSI11C | バーコード Plessey(MSI-11) チェックデジット付加 |
| CMP_CPCL_BCS_PLUS2 | Plus 2 Extension |
| CMP_CPCL_BCS_PLUS5 | Plus 5 Extension |

* NB

- バーコードの小さいバーの幅を係数として受け取ります。

* Ratio

- バーコードの比率の値を係数として受け取ります。

| 変数 | 説明 |
|----------------------|---------------------------|
| CMP_CPCL_BCS_0RATIO | バーコードの比率を1.5 : 1として設定します。 |
| CMP_CPCL_BCS_1RATIO | バーコードの比率を2.0 : 1として設定します。 |
| CMP_CPCL_BCS_2RATIO | バーコードの比率を2.5 : 1として設定します。 |
| CMP_CPCL_BCS_3RATIO | バーコードの比率を3.0 : 1として設定します。 |
| CMP_CPCL_BCS_4RATIO | バーコードの比率を3.5 : 1として設定します。 |
| CMP_CPCL_BCS_20RATIO | バーコードの比率を2.0 : 1として設定します。 |
| CMP_CPCL_BCS_21RATIO | バーコードの比率を2.1 : 1として設定します。 |
| CMP_CPCL_BCS_22RATIO | バーコードの比率を2.2 : 1として設定します。 |
| CMP_CPCL_BCS_23RATIO | バーコードの比率を2.3 : 1として設定します。 |
| CMP_CPCL_BCS_24RATIO | バーコードの比率を2.4 : 1として設定します。 |
| CMP_CPCL_BCS_25RATIO | バーコードの比率を2.5 : 1として設定します。 |
| CMP_CPCL_BCS_26RATIO | バーコードの比率を2.6 : 1として設定します。 |
| CMP_CPCL_BCS_27RATIO | バーコードの比率を2.7 : 1として設定します。 |
| CMP_CPCL_BCS_28RATIO | バーコードの比率を2.8 : 1として設定します。 |
| CMP_CPCL_BCS_29RATIO | バーコードの比率を2.9 : 1として設定します。 |
| CMP_CPCL_BCS_30RATIO | バーコードの比率を3.0 : 1として設定します。 |

* BarHeight

- 印刷するバーコードの高さの値を係数として受け取ります。

* PrintX

- 印刷するバーコードのx軸の開始点を係数として受け取ります。

* PrintY

- 印刷するバーコードのy軸の開始点を係数として受け取ります。

* Data

- 印刷するバーコードデータを係数として受け取ります。

* Count

- 印刷するバーコードのカウント関数を係数として受け取ります。

2.19. printBox

この関数は、ボックス画像を印刷するために使用されます。

```
void printBox(int xs,int ys,int xx,int yx,int Thickness)
```

[パラメーター]

* xs

- ボックス印刷のためのx軸の開始点を係数として受け取ります。

* ys

- ボックス印刷のためのy軸の開始点を係数として受け取ります。

* xx

- ボックス印刷のためのx軸の終了点を係数として受け取ります。

* yx

- ボックス印刷のためのy軸の終了点を係数として受け取ります。

* Thickness

- 印刷するボックス線の太さを係数として受け取ります。

2.20. printLine

この関数は、線の画像を印刷するために使用されます。

```
void printLine(int xs,int ys,int xx,int yx,int Thickness)
```

[パラメーター]

* xs

- 線の印刷のためのx軸の開始点を係数として受け取ります。

* ys

- 線の印刷のためのy軸の開始点を係数として受け取ります。

* xx

- 線の印刷のためのx軸の終了点を係数として受け取ります。

* yx

- 線の印刷のためのy軸の終了点を係数として受け取ります。

* Thickness

- 印刷する線の太さを係数として受け取ります。

2.21. inverseLine

この関数は、特別な領域での白黒反転画像のために使用されます。PrintLine() メソッドと同様の構文を使用します。

```
void inverseLine(int xs,int ys,int xx,int yx,int Thickness)
```

[パラメーター]

* xs

-白黒反転画像の印刷のためのx軸の開始点を係数として受け取ります。

* ys

-白黒反転画像の印刷のためのy軸の開始点を係数として受け取ります。

* xx

-白黒反転画像の印刷のためのx軸の終了点を係数として受け取ります。

* yx

-白黒反転画像の印刷のためのy軸の終了点を係数として受け取ります。

* Thickness

-白黒反転画像の印刷のための線の厚さを係数として受け取ります。

2.22. setPattern

この関数は、パターン印刷のために使用されます。

```
void setPattern(int PatternNum)
```

[パラメーター]

* PatternNum

- 印刷するパターンの値を係数として受け取ります。

| 変数 | 説明 |
|---------------------------|---------------------|
| CMP_CPCL_DEFAULT_PATTERN | 塗りつぶし(黒/デフォルト値) |
| CMP_CPCL_HORIZON_PATTERN | パターンを水平線として印刷します。 |
| CMP_CPCL_VERTICAL_PATTERN | パターンを垂直線として印刷します。 |
| CMP_CPCL_RDIAGON_PATTERN | 斜線パターンを右側に印刷します。 |
| CMP_CPCL_LDIAGON_PATTERN | 斜線パターンを左側に印刷します。 |
| CMP_CPCL_SQUARE_PATTERN | パターンを正方形として印刷します。 |
| CMP_CPCL_CROSS_PATTERN | パターンを斜線として左右に印刷します。 |

2.23. printBitmap

この関数はビットマップイメージファイルを印刷するために使用されます。

サポートされているビットマップデータのフォーマットは、BMP/JPEG/PNG/GIFです。

```
void printBitmap(String FilePath,int PrintX,int PrintY)
```

```
void printBitmap(Bitmap BMP, int PrintX,int PrintY)
```

[パラメーター]

* FilePath

- nullで終了する文字列を持つUnicode。

ファイル名の含まれるフルパスを係数として受け取ります。

* Bmp

- Androidのビットマップオブジェクト[android.graphics.Bitmap]

* PrintX

- 印刷するイメージのx軸の開始点を係数として受け取ります。

* PrintY

- 印刷するイメージのy軸の開始点を係数として受け取ります。

2.24. setContrast

この関数は、印刷するラベル全体のコントラストを調整するために使用されます。

```
void setContrast(int Darkness)
```

[パラメーター]

* Darkness

- 印刷するラベル全体のコントラストの値を係数として受け取ります。[0 <= 暗さ <=3]

| 変数 | 説明 |
|-------------------------|------------|
| CMP_CPCL_CONT_DEFAULT | デフォルトとして設定 |
| CMP_CPCL_CONT_MEDIUM | 中間の輝度で印刷 |
| CMP_CPCL_CONT_DARK | 暗く印刷 |
| CMP_CPCL_CONT_VERY_DARK | 非常に暗く印刷 |

2.25. setPageWidth

この関数は、用紙フォームの幅を定義するために使用されます。

```
void setPageWidth(int PageWidth)
```

[パラメーター]

* PageWidth

- 印刷するラベルのページ幅の値を係数として受け取ります。

2.26. printPCLLImage

この関数は、プリンターのフラッシュファイルシステムに保存されたpcx画像を印刷するために使用されます。

```
void printPCLLImage(String ImageName,int PrintX,int PrintY)
```

[パラメーター]

* ImageName

- nullで終了する文字列を持つUnicode。

フラッシュファイルシステムに保存されたpcxファイル名を係数として受け取ります。

* PrintX

- 印刷するx軸の開始点を係数として受け取ります。

* PrintY

- 印刷するy軸の開始点を係数として受け取ります。

2.27. setSpeed

この関数は、印刷速度を定義するために使用されます。

```
void setSpeed(int Speed)
```

[パラメーター]

* Speed

- 印刷速度を係数として受け取ります。[0 =< 速度 =<5]

2.28. setTone

この関数は、setContrast()の代わりに使用されるメソッドであり、印刷する全体的なラベルの暗さを調整するために使用されます。

```
void setTone(int Tone)
```

[パラメーター]

* Tone

- ラベル全体の暗さの色調を係数として受け取ります。[-99 =< 色調=< 200]

2.29. setCPCLBarCode

この関数は、HRI文字情報を設定するために使用されます。

```
void setCPCLBarCode(int FontNum,int FontSize,String Offset)
```

[パラメーター]

* FontNum

- 印刷するテキストのフォントの種類を係数として受け取ります。

* FontSize

- 印刷するテキストのフォントのサイズの値を係数として受け取ります。

* Offset

- 印刷するバーコードとテキストのフォントの間のオフセットを係数として受け取ります。

2.30. printCPCL2DBarCode

この関数は、二次元バーコードを印刷するために使用されます。

```
void printCPCL2DBarCode(int Rotation,int BarCodeType,String PrintX,String PrintY, int
UnitWidth,int UnitHeight,int Column,int SecurityLevel,String Data)
```

[パラメーター]

* Rotation

- 印刷するバーコードの印刷方向の値を係数として受け取ります。

| 変数 | 説明 |
|-----------------------|-----------------------------|
| CMP_CPCL_NO_ROTATION | バーコードを回転せずに印刷します。 |
| CMP_CPCL_90_ROTATION | バーコードを90度回転(反時計方向)して印刷します。 |
| CMP_CPCL_180_ROTATION | バーコードを180度回転(反時計方向)して印刷します。 |
| CMP_CPCL_270_ROTATION | バーコードを270度回転(反時計方向)して印刷します。 |

* BarCodeType

- 印刷するバーコードの種類を係数として受け取ります。

| 変数 | 説明 |
|-------------------------|------------------|
| CMP_CPCL_BCS_PDF417 | PDF417 バーコード |
| CMP_CPCL_BCS_QRCODE | QRCode バーコード |
| CMP_CPCL_BCS_DATAMATRIX | DataMatrix バーコード |

* PrintX

- 印刷するバーコードのx座標の値を係数として受け取ります。

* PrintY

- 印刷するバーコードのy座標の値を係数として受け取ります。

* UnitWidth

- バーコードの種類によります。

| バーコードの種類 | 説明 |
|-------------------------|------------------------|
| CMP_CPCL_BCS_PDF417 | セル幅。範囲は1～32です。デフォルトは2 |
| CMP_CPCL_BCS_QRCODE | セル幅。範囲は 1～24 です。自動 = 4 |
| CMP_CPCL_BCS_DATAMATRIX | セル幅。範囲は 1～24 です。自動 = 4 |

* UnitHeight

- バーコードの種類によります。

| バーコードの種類 | 説明 |
|-------------------------|---------------------------------|
| CMP_CPCL_BCS_PDF417 | セルの高さ。範囲は1～32です。デフォルトは6 |
| CMP_CPCL_BCS_QRCODE | QRCode バージョン。範囲は 1～40 です。自動 = 0 |
| CMP_CPCL_BCS_DATAMATRIX | デフォルト(ゼロ) |

* Column

- バーコードの種類によります。

| バーコードの種類 | 説明 |
|-------------------------|---|
| CMP_CPCL_BCS_PDF417 | 使用する列数。範囲は1～30です。 デフォルトは3 |
| CMP_CPCL_BCS_QRCODE | エラー修正レベル。範囲は 0～3 です。デフォルトは 1 |
| CMP_CPCL_BCS_DATAMATRIX | エラー修正レベル。範囲は 0、50、80、100、140、200 です。 |

* SecurityLevel

- バーコードの種類によります。

| バーコードの種類 | 説明 |
|-------------------------|--|
| CMP_CPCL_BCS_PDF417 | セキュリティレベルは、検出および/または修正されるエラー の最大数を示します。範囲は0～8です。 デフォルトは1 |
| CMP_CPCL_BCS_QRCODE | マスクパターン。範囲は 0～7 です。自動 = 8 |
| CMP_CPCL_BCS_DATAMATRIX | デフォルト(ゼロ) |

* Data

- nullで終了する文字列を持つUnicode。

印刷するテキストを係数として受け取ります。

注: 多くの変数があるため、印刷された2次元バーコードが正しく読み取られることは保証できません。
位置やサイズなどに気をつけて適切な変数をお使いください。

2.31. setMedia

この関数は、印刷するラベルの種類を設定するために使用されます。

```
void setMedia(int mode)
```

[パラメーター]

* Mode

- ラベルの種類を設定します。

| 変数 | 説明 |
|---------------------|---------------|
| CMP_CPCL_LABEL | 間にギャップのあるラベル紙 |
| CMP_CPCL_BLACKMARK | ブラックマーク紙 |
| CMP_CPCL_CONTINUOUS | 連続ラベル紙 |

注: ラベル、ブラックマーク(裏面)、ブラックマーク(表面)などの用紙種類を使用するには、上記パラメーター以外に、
プリンターのセンサー設定が正しく行われている必要があります。
プリンターのセンサー設定は、ユーティリティーで行います。使い方は、ユーティリティーのマニュアルを参照ください。

2.32. setCountry

この関数は、プリンターの文字コードを各国の適切な文字セットに設定するために使用されます。

void setCountry(Sting country)

[パラメーター]

* country

- 国別文字セットを設定します。

| 変数 | 説明 |
|--------------------------|----------------|
| CMP_CPCL_COUNTRY_USA | 国際文字: USA |
| CMP_CPCL_COUNTRY_GERMANY | 国際文字: GERMANY |
| CMP_CPCL_COUNTRY_FRANCE | 国際文字: FRANCE |
| CMP_CPCL_COUNTRY_SWEDEN | 国際文字: SWEDEN |
| CMP_CPCL_COUNTRY_SPAIN | 国際文字: SPAIN |
| CMP_CPCL_COUNTRY_NORWAY | 国際文字: NORWAY |
| CMP_CPCL_COUNTRY_ITALY | 国際文字: ITALY |
| CMP_CPCL_COUNTRY_UK | 国際文字: UK |
| CMP_CPCL_COUNTRY_CP850 | コードページ: CP850 |
| CMP_CPCL_COUNTRY_LATIN9 | コードページ: LATIN9 |

2.33. resetCountry

この関数は、国別文字セットをリセットするために使用されます。

void resetCountry()

2.34. temperature (Firmware バージョン 1.11cLB 以降のみ)

この関数は、印字ヘッドの温度を入手するために使用されます。

double temperature()

[戻り値]

印字ヘッドの温度の値

2.35. voltage (Firmware バージョン 1.11cLB 以降のみ)

この関数は、バッテリーの電圧を入手するために使用されます。

double voltage ()

[戻り値]

バッテリー電圧の値

2.36. getEmulation

この関数は、現在のプリンターのエミュレーション設定を知るために使います。

Int getEmulation ()

[戻り値]

EMUL_CPCL: CPCLエミュレーション

EMUL_ESCPOS: ESC/POSEミュレーション

2.37. printerResults

この関数は、印刷の結果を知るためのものです。

Int printerResults ()

int PrinterResults(int timeout)

[パラメーター]

* timeout

印字結果の待ち時間を指定します。(単位ミリ秒: デフォルト5000)

[戻り値]

| 変数 | 説明 |
|--------------------------|--------------------------|
| CMP_STS_CPCL_NORMAL | 印刷は成功 |
| CMP_STS_CPCL_PAPER_EMPTY | プリンターは紙無し状態 |
| CMP_STS_CPCL_COVER_OPEN | プリンターカバーが空いた状態 |
| CMP_STS_CPCL_BATTERY_LOW | プリンターのバッテリーが低下した状態 |
| CMP_STS_CPCL_PRINTEROFF | プリンターはオフ状態 (1.087にて仕様追加) |
| CMP_STS_CPCL_TIMEOUT | プリンターから返信なし(1.087にて仕様追加) |
| その他の値 | プリンターはそれ以外のエラー状態 |

2.38. printAndroidFont (位置揃え指定)

この関数は、Androidの内蔵フォントを使うために使用されます。

void PrintAndroidFont(String textString, int widthDots, int textSize, int PrintY, int alignment)

void PrintAndroidFont(Typeface typeface, String textString, int widthDots, int textSize, int PrintY, int alignment)

void PrintAndroidFont(Typeface typeface, boolean isBold, String textString, int widthDots, int textSize, int PrintY, int alignment)

void PrintAndroidFont(Typeface typeface, boolean isBold, boolean isItalic, String textString, int widthDots, int textSize, int PrintY, int alignment)

void PrintAndroidFont(Typeface typeface, boolean isBold, boolean isItalic, boolean isUnderline, String textString, int widthDots, int textSize, intPrintY, int alignment)

[パラメーター]

* textString

- nullで終了するUnicode文字列へのポインター。印刷するテキストを指定します。

*widthDots

- 印刷するテキストの印字幅を指定します。[単位: dot]

* textSize

- 印刷するテキストのフォントサイズを指定します。[単位: dot]

* PrintY

-印刷するイメージの始点のY座標を指定する

* Alignment

- この値で印字位置を指定します。

| 変数 | 説明 |
|----------------------|------|
| CMP_ALIGNMENT_LEFT | 左揃え |
| CMP_ALIGNMENT_CENTER | 中央揃え |
| CMP_ALIGNMENT_RIGHT | 右揃え |

* typeface

- Android搭載フォントを指定します。

| 変数 | 説明 |
|------------|-----------------|
| SANS_SERIF | SANS_SERIF font |
| SERIF | SERIF font |
| MONOSPACE | MONOSPACE font |

* isBold

- Android搭載フォントの強調印字を指定します。

* isItalic

- Android搭載フォントの斜体印字を指定します。

* isUnderline

- Android搭載フォントの下線印字を指定します。

2.39. printAndroidFont（座標指定）

この関数は、Androidの内蔵フォントをX,Y座標指定で使うために使用されます。

void PrintAndroidFont(int printX, int printY, String textString, int widthDots, int textSize)

void PrintAndroidFont(int printX, int printY, Typeface typeface, String textString, int widthDots, int textSize)

void PrintAndroidFont(int printX, int printY,Typeface typeface, boolean isBold, String textString, int widthDots, int textSize)

void PrintAndroidFont(int printX, int printY, Typeface typeface, boolean isBold, boolean isItalic, String textString, int widthDots, int textSize)

void PrintAndroidFont(int printX, int printY,Typeface typeface, boolean isBold, boolean isItalic, boolean isUnderline, String textString, int widthDots, int textSize)

[パラメーター]

* printX, printY

- 印刷するx/y軸の開始点を係数として受け取ります。

* textString

- nullで終了するUnicode文字列へのポインタ。印刷するテキストを指定します。

*widthDots

- 印刷するテキストの印字幅を指定します。[単位: dot]

* textSize

- 印刷するテキストのフォントサイズを指定します。[単位: dot]

* typeface

- Android搭載フォントを指定します。

| 変数 | 説明 |
|------------|-----------------|
| SANS_SERIF | SANS_SERIF font |
| SERIF | SERIF font |
| MONOSPACE | MONOSPACE font |

* isBold

- Android搭載フォントの強調印字を指定します。

* isItalic

- Android搭載フォントの斜体印字を指定します。

* isUnderline

- Android搭載フォントの下線印字を指定します。

* Alignment

- この値で印字位置を指定します。

| 変数 | 説明 |
|----------------------|------|
| CMP_ALIGNMENT_LEFT | 左揃え |
| CMP_ALIGNMENT_CENTER | 中央揃え |
| CMP_ALIGNMENT_RIGHT | 右揃え |

* typeface

- Android搭載フォントを指定します。

| 変数 | 説明 |
|------------|-----------------|
| SANS_SERIF | SANS_SERIF font |
| SERIF | SERIF font |
| MONOSPACE | MONOSPACE font |

* isBold

- Android搭載フォントの強調印字を指定します。

* isItalic

- Android搭載フォントの斜体印字を指定します。

* isUnderline

- Android搭載フォントの下線印字を指定します。