

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

ESC/POS

Ver. 1.14

シチズン・システムズ株式会社

目次

1. はじめに
2. メソッド
3. OLE POSコマンドのPrintNormal()関数でサポートされるコマンド一覧

履歴

バージョン	対象ライブラリ	説明
1.00	1.051	公開
1.01	1.051	日本語印刷のための設定例の説明を追加
1.02	1.052	関数を追加 PrintPDF47、printQRCode 関数の引数追加オプション printBitmap、printNVBitmap
1.03	1.056	printerCheck関数にタイムアウト機能を追加 printBitmap関数の引数のオプションを追加
1.04	1.058	QRコードのモジュールサイズの範囲を1-16に修正
1.05	1.062	printTextのattributeにCMP_FNT_UNDERLINE2を追加 getEmulation関数を追加 印字の行ごとの速度変化をスムーズになるよう改善
1.06	1.063	下線などの文字修飾が不安定になる不具合を修正
1.07	1.064	USBのサポートを追加
1.08	1.071	WiFiMultiConnectionサポートを追加 printerResults関数を追加 printAndroidFont関数を追加 printTextのattributeに白黒反転を追加 printQRcodeの引数にversionを追加 日本語文字の下線印字の不具合を修正 printTextの日本語文字修飾の解除不具合を修正 Font Bのサイズ拡大が出来ない不具合を修正
1.09	1.073	printBitmapの引数のオプションを追加 printerCheckの引数のオプションを追加 setText関数を追加 setAlignment関数を追加
1.10	1.083	表紙のロゴを変更 setCharSetとsetCodepage関数を追加 printerCheckの追加引数の説明の間違いを修正 サポートするBluetoothモジュールの追加
1.11	1.087	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)
1.12	1.088	ウェイクアップのためのダミーデータを送るタイミングにより起きていた問題解決 タイプ2で電源ON直後のステータスがおかしい問題を解決 本マニュアルには変更なし。
1.13	1.090	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。
1.14	1.092	printBitmapがJPEG/PNG/GIFを扱えるようになった。 setDitheringを追加した。

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。

2. メソッド

ESCPOSPrinter Classで定義されています。定数は、ESCPOSConst Interfaceに定義されています。

2.1. ESCPOSPrinter

これは、Constructorメソッドです。ESCPOSPrinter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。

ESCPOSPrinter()

ESCPOSPrinter(String charset)

ESCPOSPrinter(DeviceConnection connection)

ESCPOSPrinter(String charset, DeviceConnection connection)

[パラメーター]

* charset

- 文字セット名

* connection

- デバイス接続方法 (USBPortConnectionまたはWiFiMultiConnection)

注: USBPortConnectionについては、別書のAndroid Port Package Manualに説明があります。

使用例: 日本語印刷に対応する場合

ESCPOSPrinter("Shift-JIS")

2.2. printNormal

この関数は、テキストの印刷およびOPOSコマンドをサポートするために使用されます。

```
void printNormal(String data)
```

[パラメーター]

* data

- nullで終了するUnicode文字列へのポインター。OLE POSコマンドのPrintNormal関数と同じです。

2.3. printString

この関数は、ESCコマンドでテキストの印刷をサポートするために使用されます。

```
void printString(String data)
```

[パラメーター]

* data

- nullで終了するUnicode文字列へのポインター。印刷するUnicode文字列を設定します。

注: 日本語印刷対応のために、ESCPOSPrinter("Shift-JIS") とすると、ESCコマンドとともに送った1バイトコードの
0x80~0xA0, 0xE0~0xFF

が0x3F('?')に変換され意図したようにESCコマンドが動作しません。

その対策としては、ESCPOSPrinter("ISO-8859-1") などと生成し、印字の際のバイト配列のエンコーディングに
"ISO-8859-1" を指定します。2つのESCPOSPrinterを切り替えることで、問題を避けることができます。

2.4. printText

この関数は、テキストの印刷をサポートするために使用されます。

```
void printText(String data, int alignment, int attribute, int textSize)
```

[パラメーター]

* data

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

* alignment

- この値は配置を指定します。テキストの配置を設定します。

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

* attribute

- この値はテキストの属性です。印刷するテキスト属性を設定します。

変数	説明
CMP_FNT_DEFAULT	FontA、標準として設定
CMP_FNT_FONTB	FontBとして設定
CMP_FNT_BOLD	太字として設定
CMP_FNT_UNDERLINE	1ドット構成の下線として設定
CMP_FNT_UNDERLINE2	2ドット構成の下線として設定
CMP_FNT_REVERSE	白黒反転印字として設定

* textSize

- この値はテキストのサイズです。印刷するテキストサイズを設定します。

変数(幅の比率を設定)	説明
CMP_TXT_1WIDTH	幅の比率を 1 倍として設定
CMP_TXT_2WIDTH	幅の比率を 2 倍として設定
CMP_TXT_3WIDTH	幅の比率を 3 倍として設定
CMP_TXT_4WIDTH	幅の比率を 4 倍として設定
CMP_TXT_5WIDTH	幅の比率を 5 倍として設定
CMP_TXT_6WIDTH	幅の比率を 6 倍として設定
CMP_TXT_7WIDTH	幅の比率を 7 倍として設定
CMP_TXT_8WIDTH	幅の比率を 8 倍として設定

変数(高さの比率を設定)	説明
CMP_TXT_1HEIGHT	高さの比率を 1 倍として設定
CMP_TXT_2HEIGHT	高さの比率を 2 倍として設定
CMP_TXT_3HEIGHT	高さの比率を 3 倍として設定
CMP_TXT_4HEIGHT	高さの比率を 4 倍として設定
CMP_TXT_5HEIGHT	高さの比率を 5 倍として設定
CMP_TXT_6HEIGHT	高さの比率を 6 倍として設定
CMP_TXT_7HEIGHT	高さの比率を 7 倍として設定
CMP_TXT_8HEIGHT	高さの比率を 8 倍として設定

2.5. printBitmap

この関数は、画像ファイル[Bmp/JPEG/PNG/GIFファイル]の印刷のために使用されます。

```
void printBitmap(String bitmapName, int alignment)
void printBitmap(String bitmapName, int alignment, int size)
void printBitmap(Bitmap bmp, int alignment)
void printBitmap(Bitmap bmp, int alignment, int size)
```

[パラメーター]

* BitmapName

- nullで終了するUnicode文字列へのポインター。この値は、ビットマップファイルの完全パスを付けたビットマップファイル名です。

* Bmp

- android.graphics.Bitmap クラスのBitmapオブジェクトです。

* Alignment

- この値は配置です。画像の配置を設定します。

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

* Size

- この値は画像のサイズです。印刷する画像のサイズを設定します。

変数	説明
CMP_BITMAP_NORML	通常サイズ
CMP_BITMAP_DOUBLE_WIDTH	幅 2 倍
CMP_BITMAP_DOUBLE_HEIGHT	高さ 2 倍
CMP_BITMAP_QUADRUPLE	4 倍

2.6. printBarCode

この関数は、バーコードの印刷をサポートするために使用されます。

```
void printBarCode(String data,int symbology,int height,int width,int alignment,int textPosition)
```

[パラメーター]

* Data

- nullで終了するUnicode文字列へのポインター。印刷するバーコードデータを設定します。

* Symbology

- この値はバーコード記号のタイプです。印刷するバーコードタイプを設定します。

変数	説明
CMP_BCS_UPCA	UPC Aバーコードを印刷
CMP_BCS_UPCE	UPC E バーコードを印刷
CMP_BCS_EAN8	8桁の EAN バーコード
CMP_BCS_EAN13	13桁の EAN バーコード
CMP_BCS_JAN8	8桁の JAN バーコード
CMP_BCS_JAN13	13桁の JAN バーコード
CMP_BCS_ITF	Interleaved 2/ 5を印刷
CMP_BCS_Codabar	Codabar バーコードを印刷
CMP_BCS_Code39	Code 39 バーコードを印刷
CMP_BCS_Code93	Code 93 バーコードを印刷
CMP_BCS_Code128	Code 128 バーコードを印刷

* Height

- この値はドット単位のバーコードの高さです。印刷するバーコードの高さを設定します。

* Width

- この値はバーコード幅 [2 <= 値 <= 6] です。印刷するバーコードの全体幅を設定します。

* Alignment

- この値は配置です。バーコードの配置を設定します。

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

* TextPosition

- この値は、バーコードHRI文字(バーコードデータ)の印刷位置です。

変数	説明
CMP_HRI_TEXT_NONE	バーコードデータの印刷を行いません
CMP_HRI_TEXT_ABOVE	バーコードデータをバーコードの上に印刷
CMP_HRI_TEXT_BELOW	バーコードデータをバーコードの下に印刷

2.7. lineFeed

この関数は、改行コマンドをプリンターに送信するために使用されます。

```
void lineFeed(int LFCount)
```

[パラメーター]

* LFCount

- この値は改行を行う行数です。改行カウンタを設定します。

2.8. 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)
```

```
int printerCheck(int timeout, Boolean bCheckResults)
```

[パラメーター]

* timeout: ミリ秒 (デフォルト:5000ms)

* bCheckResults:

- printerResults()使用の可否

変数	説明
True	printerResult()にてプリンターの状態確認可
False	printerResults()でプリンターの状態確認不可

[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_PRINTEROFF : プリンターがオフの場合に、この値が戻ります。 ← 1.087にて追加

CMP_STS_TIMEOUT : プリンターからの返信が無い場合に、この値が戻ります。 ← 1.087にて追加

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。

2.9. 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_NORMAL: プリンターステータスは「エラーなし」で、MSRは「準備ができていません」。

CMP_STS_PAPER_EMPTY : プリンターステータスは「用紙がありません」です。

CMP_STS_COVER_OPEN : プリンターカバーが開いています。

CMP_STS_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_MSR_READ : 現在のMSRは読み取りモードで、印刷はできません。

CMP_STS_PRINTEROFF : プリンターの電源がOFFです。 ← 1.087にて追加

CMP_STS_TIMEOUT : プリンターの返信がありません。 ← 1.087にて追加

注: status関数の前にprinterCheckを実行すること

2.10. printNVBitmap

この関数は、フラッシュメモリに保存されたビットマップ画像の印刷をサポートするために使用されます。

```
void printNVBitmap(int NVImageNumber)
```

```
void printNVBitmap(int NVImageNumber, int size)
```

[パラメーター]

* NVImageNumber

- フラッシュメモリ内に保存された印刷する番号の画像を設定します。

* Size

- この値は画像のサイズです。印刷する画像のサイズを設定します。

変数	説明
CMP_BITMAP_NORML	通常サイズ
CMP_BITMAP_DOUBLE_WIDTH	幅 2 倍
CMP_BITMAP_DOUBLE_HEIGHT	高さ 2 倍
CMP_BITMAP_QUADRUPLE	4 倍

2.11. printPDF417

この関数は、PDF417バーコード印刷をサポートするために使用されます。

```
void printPDF417(String pdfData, int dataLength, int numberOfColumns, int cellWidth,int alignment)
```

[パラメーター]

* pdfData

- 印刷するPDF417バーコードデータです。

* dataLength

- バーコードデータの長さです。

* numberOfColumns

- 桁数です。(範囲 0-30)

0の場合は、自動処理となります。

* cellWidth

- セルの幅です。(範囲 2-8)

* alignment

- この値は配置を指定します。テキストの配置を設定します。

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

注: 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。

2.12. printQRCode

この関数は、QRコード印刷をサポートするために使用されます。

```
void printQRCode(String data, int dataLength, int moduleSize, int ecLevel, int alignment)
```

```
void printQRCode(String data, int dataLength, int version, int moduleSize, int ecLevel, int alignment)
```

[パラメーター]

* data

- 印刷するQRコードデータです。

* dataLength

- QRコードデータの長さです。

* version

- QRコードのバージョン（自動 = 0, 固定 = 1 - 40）

* moduleSize

- モジュールサイズです。(1-16)

* ecLevel

- エラー訂正レベルです。

変数	説明
CMP_QRCODE_EC_LEVEL_L	エラー訂正レベル L (7%)
CMP_QRCODE_EC_LEVEL_M	エラー訂正レベル M (15%)
CMP_QRCODE_EC_LEVEL_Q	エラー訂正レベル Q (25%)
CMP_QRCODE_EC_LEVEL_H	エラー訂正レベル H (30%)

* alignment

- この値は配置を指定します。QRコードの配置を設定します。

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

注: 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。

2.13. getEmulation

この関数は、CMP-30の現在のエミュレーション設定情報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/POSエミュレーション

注) この機能は、CMP-20では無効です。

2.14. printerResults

この関数は、印刷の結果を知るために使います。

int printerResults()

int PrinterResults(int timeout)

[パラメーター]

* timeout

- 印刷結果の待ち時間です。

[戻り値]

変数	説明
CMP_SUCCESS	印刷成功
CMP_STS_PAPER_EMPTY	紙無し状態
CMP_STS_COVER_OPEN	カバーが空いた状態
CMP_STS_PRINTEROFF	オフ状態 (1.087にて仕様追加)
CMP_STS_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

2.15. printAndroidFont

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 alignment)
```

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

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

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

```
void PrintAndroidFont(Typeface typeface, boolean isBold, boolean isItalic, boolean isUnderline, String textString, int widthDots, int textSize, 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]

* 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。

2.16. setCharSet

この関数は、文字コードを指定するために使用されます。

```
void setCharSet(String charSet)
```

```
void setCharSet(String charset, int iCodepage)
```

[パラメーター]

*charset

- 文字コード名を指定します。

*iCodepage

- setCodepageを参照してください。

2.17. setCodepage

この関数は、コードページを設定するために使用されます。

```
void setCodepage(int iCodepage)
```

[パラメーター]

*iCodepage - この値は、コードページになります。

Variable (Set up code page)	Description
CMP_CODEPAGE_CP437	Code page CP437
CMP_CODEPAGE_KATAKANA	Code page KATAKANA
CMP_CODEPAGE_CP850	Code page CP850
CMP_CODEPAGE_CP860	Code page CP860
CMP_CODEPAGE_CP863	Code page CP863
CMP_CODEPAGE_CP865	Code page CP865
CMP_CODEPAGE_CP1252	Code page CP1252
CMP_CODEPAGE_CP866	Code page CP866
CMP_CODEPAGE_CP852	Code page CP852
CMP_CODEPAGE_CP858	Code page CP858
CMP_CODEPAGE_CP1253	Code page CP1253
CMP_CODEPAGE_CP737	Code page CP737
CMP_CODEPAGE_CP857	Code page CP857
CMP_CODEPAGE_ISO_8859_9	Code page ISO_8859_9
CMP_CODEPAGE_CP864	Code page CP864
CMP_CODEPAGE_CP862MD	Code page CP862MD
CMP_CODEPAGE_ISO_8859_2	Code page ISO_8859_2
CMP_CODEPAGE_MOZOVIA	Code page MOZOVIA
CMP_CODEPAGE_CP1250	Code page CP1250
CMP_CODEPAGE_CP1254	Code page CP1254
CMP_CODEPAGE_CP1251	Code page CP1251
CMP_CODEPAGE_TIS14	Code page TIS14
CMP_CODEPAGE_TIS17	Code page TIS17
CMP_CODEPAGE_TIS11	Code page TIS11
CMP_CODEPAGE_CP1257	Code page CP1257
CMP_CODEPAGE_CP1258	Code page CP1258
CMP_CODEPAGE_ISO_8859_7	Code page ISO_8859_7
CMP_CODEPAGE_CP1256	Code page CP1256
CMP_CODEPAGE_ISO_8859_1	Code page ISO_8859_1

2.18. setText

この関数は、テキストの属性とサイズを設定するために使用されます。

```
void setText(int attribute, int textSize)
```

[パラメーター]

* attribute

- この値はテキストの属性です。印刷するテキスト属性を設定します。

変数	説明
CMP_FNT_DEFAULT	FontA、標準として設定
CMP_FNT_FONTB	FontBとして設定
CMP_FNT_BOLD	太字として設定
CMP_FNT_UNDERLINE	1ドット構成の下線として設定
CMP_FNT_REVERSE	白黒反転印字として設定

* textSize

- この値はテキストのサイズです。印刷するテキストサイズを設定します。

変数(幅の比率を設定)	説明
CMP_TXT_1WIDTH	幅の比率を 1 倍として設定
CMP_TXT_2WIDTH	幅の比率を 2 倍として設定
CMP_TXT_3WIDTH	幅の比率を 3 倍として設定
CMP_TXT_4WIDTH	幅の比率を 4 倍として設定
CMP_TXT_5WIDTH	幅の比率を 5 倍として設定
CMP_TXT_6WIDTH	幅の比率を 6 倍として設定
CMP_TXT_7WIDTH	幅の比率を 7 倍として設定
CMP_TXT_8WIDTH	幅の比率を 8 倍として設定

変数(高さの比率を設定)	説明
CMP_TXT_1HEIGHT	高さの比率を 1 倍として設定
CMP_TXT_2HEIGHT	高さの比率を 2 倍として設定
CMP_TXT_3HEIGHT	高さの比率を 3 倍として設定
CMP_TXT_4HEIGHT	高さの比率を 4 倍として設定
CMP_TXT_5HEIGHT	高さの比率を 5 倍として設定
CMP_TXT_6HEIGHT	高さの比率を 6 倍として設定
CMP_TXT_7HEIGHT	高さの比率を 7 倍として設定
CMP_TXT_8HEIGHT	高さの比率を 8 倍として設定

2.19. setAlignment

この関数は、テキストの位置揃えを設定するために使用されます。

```
void setAlignment(int alignment)
```

[パラメーター]

* alignment

- この値は配置を指定します。テキストの配置を設定します。

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

2.20. setDithering

この関数は、ビットマップのディザリングのために使います。

```
void setDithering(int iDither)
```

[パラメーター]

* iDither

- この値は配置を指定します。テキストの配置を設定します。

変数	説明
CMP_BITMAP_NO_DITHER	単純2値化方式(輝度閾値)
CMP_BITMAP_ERROR_DIFFUSION	誤差拡散方式
CMP_BITMAP_ORDERED_DITHER	配列ディザ方式

3. printNormal()関数でサポートされている OPOS コマンド一覧

即実行コマンド すぐに指示とおりに実行します。

名称	データ	備考
Feed and Paper cut(用紙送りおよび用紙裁断)	ESC #fP	RecLinesToPaperCut行で用紙送りした後で、出てきた用紙を裁断します。文字'#'は、“Paper cut”エスケープシーケンスによって定義されます。
Print bitmap(ビットマップを印刷)	ESC #B	あらかじめ保存されたビットマップを印刷します。文字'#'は、ビットマップ番号によって置き換えられます。
Feed lines(行単位用紙送り)	ESC #fF	行数によって用紙を送ります。文字'#'は、用紙を送る行数を指示するASCIIの10進数の文字列によって置き換えられます。'#'が省略された場合、1行で用紙が送られます。
Feed units(紙送り単位)	ESC #uF	1/360ドット単位で用紙を送り出します。文字'#'は、用紙を送る単位数を指定したASCIIの10進数の文字列によって置き換えられます。

印刷モードコマンド 明示的に変更されるまで設定が保持される特性を設定します。

名称	データ	備考
Font typeface selection(フォント名の選択)	ESC #fT	次のデータに対して新しい書体を選択します。文字'#'の値: 0 = デフォルトの書体。 1 = FontTypefaceListプロパティから最初の書体を選択します。 2 = FontTypefaceListプロパティから2番目の書体を選択します。 3以降も同様になります。

行単位コマンド 各印刷メソッドの終わり、または"Normal"シーケンスによってリセットされる特性を設定します。

名称	データ	備考
Bold(太字)	ESC [!] bC	太字またはダブルストライクで印刷します。
Underline(下線)	ESC [!] [#]uC	下線付きで印刷します。文字 '#'は、プリンタードット単位での下線幅を指定した ASCIIの16進数の文字列によって置き換えられます。 '#'が省略された場合は、プリンター固有 のデフォルト幅が使用されます。
Reverse video(反転映像)	ESC [!] rvC	反転映像形式で印刷します。' 'が指定された場合、 反転映像は無効にされます。
Single high and wide(等倍 の高さと幅)	ESC 1C	標準サイズで印刷します。
Double wide(倍角)	ESC 2C	文字を2倍の幅で印刷します。
Double high(縦倍角)	ESC 3C	文字を2倍の高さで印刷します。
Double high and wide(縦 倍角と倍角)	ESC 4C	文字を2倍の高さと幅で印刷します。
Scale horizontally(横方向 に拡大/縮小)	ESC #hC	幅を通常サイズの'#'倍拡大/縮小して印刷しま す。ここで、'#'はASCIIの10進数の文字列によっ て置き換えられます。
Scale vertically(縦方向に 拡大/縮小)	ESC #vC	高さを通常サイズの'#'倍拡大/縮小して印刷しま す。ここで、'#'はASCIIの10進数の文字列によっ て置き換えられます。
Center(中央揃え)	ESC cA	続くテキストを中央で揃えます。
Right justify(右揃え)	ESC rA	続くテキストを右で揃えます。
Left justify(左揃え)	ESC lA	続くテキストを左で揃えます。
Normal(標準)	ESC N	プリンター特性を標準の状態に復元します。