

# **RCL** リファレンスマニュアル

**Ver. 4.03**

**RCL : R Command Language**

日東電工株式会社

デュラプリンタのコマンド (以下、このコマンドのことを“**RCL (R Command Language)**”と表記します) に対応したプリンタには、以下のモデルがあります。

●DURA PRINTER R	検証機内蔵プリンタ
●DURA PINTER R4	
●DURA PRINER SG	
●DURA PRINER SR	検証機内蔵プリンタ
●DURA PRINER SR (RSS)	
●DURA PRINTER LSP5300	ライナーレスプリンタ
●DURA PRINER SRS	
●DURA PRINER SRS (RSS)	
●DURA PRINER LP5320	
●KP4300, KP3000	感熱紙プリンタ

#### ご注意

- (1) 本書の内容の一部または全部を無断複製、改変・転載することは禁止されています。
- (2) 本書の内容に関して将来予告なしに変更することがあります。
- (3) 本書の内容について万全を期して作成いたしましたが、万一ご不審な点や、記載漏れなどお気づきのことがありましたらご連絡ください。

**COPYRIGHT (C) 1990, 2005 NITTO DENKO CO.,LTD.**

**COPYRIGHT (C) 1990, 1998 TOHOKU RICOH CO.,LTD.**

**RCL**で印字するアウトラインフォントはリョービ株式会社の字母をもとにデザイン制作されたものです。

# 目 次

1. はじめに	1
1.1. RCL とは	1
1.2. 本書の読み方	1
1.3. 印刷位置の寸法単位 (ピクセル)	2
1.4. コマンドの形式について	2
1.5. プログラムの書式	4
1.6. RCL を使用するために	5
2. ヘッダ	6
2.1. 構文	6
2.2. ヘッダ開始文字	6
2.3. ラベル名	7
2.4. 印刷するラベルの数	8
2.5. ラベル間スペース	8
2.6. 印字領域の長さ	8
2.7. 左端開始点	8
3. プログラム・コントロール	10
3.1. バッファ処理	10
4. 印刷位置移動コマンド	13
4.1. 原点位置	13
4.2. 印刷開始基準位置	14
4.3. 絶対位置移動コマンド	14
4.4. 相対位置移動コマンド	15
5. 水平線と垂直線	18
5.1. 水平線	18
5.2. 垂直線	19
5.3. 対角線	20
6. 枠 (ボックス)	22
6.1. 枠コマンドの構文	22
6.2. 枠の描画位置の指定	22
6.3. 輪郭だけの枠	23
6.4. 塗りつぶし	24
6.5. 反転枠描画	25
6.6. 枠内消去 (領域クリア)	25
7. バーコード	26
7.1. バーコードの種類	27
7.2. バーコードの印刷	28
7.3. CODE39 と ITF	31
7.4. UPC と EAN	34

7.5. CODA BAR .....	37
7.6. CODE93 .....	40
7.7. CODE128 .....	42
7.8. CASECODE CODE128 (EAN-128) .....	49
8. 検証機能 .....	52
8.1. 検証機制御コマンド .....	52
8.2. ハーフドット .....	54
8.3. 検証対象バーコードについて .....	56
9. 2次元シンボル .....	57
9.1. 2次元シンボルの選択 .....	57
9.2. 2次元シンボルの印字 .....	58
9.3. CODE49 .....	59
9.4. QR CODE .....	61
9.5. Reduce symbol (RSS)シンボル .....	65
10. テキストと可読文字 .....	73
10.1. ベクトルフォント .....	73
10.2. ドットフォント .....	84
10.3. アウトラインフォント .....	93
11. 連番 (連続番号) .....	99
11.1. ルーピング .....	99
11.2. 増分 .....	103
11.3. バーコードの連番 .....	104
11.4. テキストの連番 .....	106
12. プリンタ制御 .....	110
12.1. 機能設定コマンド .....	110
12.2. フォームの長さ (ラベルピッチ) .....	110
12.3. 上下反転印刷 .....	111
12.4. 遠隔操作 .....	112
13. ロゴ (LOGO) .....	116
13.1. ロゴ (LOGO) .....	116
13.2. ロゴの展開方法 .....	116
13.3. データを指定したロゴコマンド .....	117
13.4. ロゴの例 .....	118
14. 拡張メモリ・オペレーション .....	125
14.1. 拡張メモリへのフォーマットの書き込み .....	125
14.2. 拡張メモリ・モードでの印刷 .....	127
14.3. ロゴの拡張メモリへの書き込み .....	131
14.4. 外字ファイルの書き込み .....	137
14.5. 拡張メモリを使うときの注意点 .....	141

15. カッター機能 .....	142
15.1.カッター制御コマンド .....	142
16. トラブルシューティングガイド .....	144
16.1.テキスト印字／16進ダンプ .....	144
16.2.一般的な問題とその対策 .....	145
17. コマンド概要 .....	149
18. プリンタ仕様 .....	152
18.1. モデル毎仕様 .....	152
19. モデル毎のバーコード密度一覧表 .....	153
19.1. 8ドット/mm ヘッド機 .....	153
19.2. 7.6ドット/mm ヘッド機 .....	156
19.3. 12ドット/mm (11.8ドット/mm) ヘッド機 .....	159
19.4. 16ドット/mm (15.75ドット/mm) ヘッド機 .....	162
20. モデル毎のドットフォントサイズ一覧表 .....	165
20.1. 8ドット/mm ヘッド機 (DURA PRINTER R) .....	165
20.2. 7.6ドット/mm ヘッド機 (DURA PRINTER SG) .....	165
20.3. 12ドット/mm ヘッド機 (DURA PRINTER LSP5300) .....	166
20.4. 16ドット/mm ヘッド機 (DURA PRINTER SR SRs) .....	167
21. フォント選択表 .....	168
21.1.Dura Printer R .....	168
21.2.Dura Printer SG .....	168
21.3.Dura Printer SR,SRs,LSP5300 .....	169
22. バーコード選択表 .....	170
22.1.BSYM の設定値 .....	170
22.2.BDEF の設定値 .....	171
23. 2次元シンボルの選択表 .....	173
23.1.2次元シンボル用コマンド .....	173
23.2.PDF417 .....	175
24. ステータス応答フォーマット一覧 .....	178
25. コマンド対応一覧表 .....	181

# 1.はじめに

---

## 1.1. RCLとは

RCLはバーコードプリンタ（以降プリンタと呼びます）の持つすべての機能を利用するためのインタープリタ型言語です。RCLを利用することで、ニーズに合わせていろいろなバーコード・ラベルを作成することができます。

RCLとプリンタを使って印刷できるのは、漢字を含む文字や図形、ロゴ、バーコード（UPC、EAN、Code39、Interleaved 2 of 5、Code93、Code128、CodaBar）2次元シンボル（QRCode、Code49等）です。

## 1.2. 本書の読みかた

本書では、初めてRCLを使うソフトウェアプログラマーを対象に、RCLのコマンドや構文の細かな内容について説明しているソフトウェアマニュアルです。また、すでにRCLをお使いの場合、コマンド構文やより優れた機能を探す時に、参考資料としてお役に立ちます。

注：

本書では、RCLが持っているすべてのコマンドについて書かれているため、プリンタのモデルによっては対応されていない場合があります。25章のコマンド対応一覧表をご覧の上、お使いください。また、プリンタの搭載されているROMバージョンによっては対応されていないコマンドがあります。

### 1.2.1. 常に必要なコマンド

RCLには、ラベルを印刷するためには絶対に必要なコマンドと、必要に応じて使用するコマンドがあります。常に必要なコマンドは、ヘッダ、バッファ処理開始（Start Processing Buffer : SPB）、コマンド列終了（TeRMinate : TRM）の3つで、省略することはできません。そこで、本書のラベル印刷のサンプルプログラムは、いずれもヘッダ、SPB、TRMが必ず付加されるものとします。

### 1.2.2. データ書式

RCLでは、すべてのデータはセミコロン（;）で区切ります。データが、整数、10進数またはASCII文字列いずれの場合でも当てはまります。

セミコロン記号が無いと、データ・フィールドの区切りがわかりません。

本書のサンプルプログラムでは、下の例のようにセミコロン（;）でデータ・フィールドを分離して示します。

HBR ; 100 ; VBR ; 50 ; DBBX ; 10 ; 25 ; 600 ; 300 ;

### 1.2.3. コマンドの書式

本書ではコマンド（ニモニツタ）を大文字で（たとえばSPB ; TRM ; のように）表記します。しかし、小文字でも構いません。

## NITTO DENKO

なお、本書においてコマンド全体を示すときは、**TeRMinate** のようにニモニツクになる部分を大文字で表記します。この場合のニモニツクは **TRM** です。**RCL** では、**HOME** のような例外を除いて、コマンド名全体を記述しても受け付けられません。ニモニツクを使ってください（例えば、**TERMINATE** ではなく、**TRM**）。

### 1.2.4. 本書で使用する記号

**ASCII** コードの **5CH** は多くのコンピュータでは **¥** ですが、**IBM** 系など海外のコンピュータでは **\** (バックスラッシュ) です。本書では特に断りのない限り **5CH** は **¥** で表記しますが、**IBM** 系など海外の機種では **\** (バックスラッシュ) に置き換えてください。

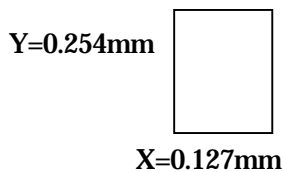
## 1.3. 印刷位置の寸法単位 (ピクセル)

**RCL** では、印刷位置の寸法はピクセルと呼ぶ単位で表します。

水平方向の最小ピクセル単位と垂直方向のピクセル単位は、以下の通りです。

水平方向最小ピクセル (**X**) = **0.127mm**

垂直方向最小ピクセル (**Y**) = **0.254mm**



例えば、水平方向に**10**ピクセルとは印刷位置が**1.27mm**移動することを示します。また垂直方向に**10**ピクセルとは、印刷位置が、**2.54mm**移動することを示します。

### 注

寸法単位 (ピクセル) はプリンタの分解能に依存しません。例えば **200dpi**、**300dpi**、**400dpi** の分解能の異なるプリンタでも印刷位置の寸法指定は **X=0.127mm**、**Y=0.254mm** 単位で指定します。

## 1.4. コマンドの形式について

**RCL** は高水準インタープリタ言語として設計されています。この言語には、バーコード・ラベルを定義して印刷するための **50** 以上のコマンドがあります。これらのコマンドのほとんどは引き数を必要とし、いくつかのコマンドは特定のデータを必要とします。コマンドによって異なりますが、データはピクセル単位、**10** 進数または **16** 進数で指定します。コメント (実行されない) をコマンド列の中に入れておくと、プログラムが読みやすくなります。

### 1.4.1. コマンドの名称

**RCL** のコマンドは長さ **3~4** 文字の英語のニモニツクです。多くは各コマンドの英語による完全な表記の頭文字です。例えば、**SPB** は **Start Processing Buffer** の略です。また、**DVL** は **Draw Vertical Line** の略です。いくつかのコマンドでは1つの単語からいくつかの文字を抜き出しています。例えば、**MRK** は **MaRK** の略で、

## NITTO DENKO

TRM は TeRMinate の略です。その他に、HOME、LOGO、BOTH、FLIP などのように完全な英単語をニモニックとして使う場合があります。

### 1.4.2. 引き数

ほとんどのコマンドは、あるコマンドの特定の機能を指定したり、寸法を指定するために引き数を使います。引き数が必要なコマンドでは、正しい順番で引き数を指定しなければなりません。そうでないと、エラーになります。引き数はコマンドによって、ピクセル単位、10 進数 または 16 進数で指定してください。

コマンドのいろいろな形式の例を以下に示します。

表 1

ニモニック	意 味	引き数	単位	例
HBR	<b>Horizontal Base Reference</b> 水平絶対位置移動	Xの位置	Xピクセル	HBR;100;
VBR	<b>Vertical Base Reference</b> 垂直絶対位置移動	Yの位置	Yピクセル	VBR;50;
BCLC	<b>Bar Code Label Count</b> リピート枚数設定	印刷枚数	10進整数	BCLC;5;
IDF	<b>Increment/Decrement Field</b> テキストフィールド連番 インクリメント・デクリメント設定	量	10進整数	IDF;3;
DHR	<b>Define Human Readable</b> ベクトルフォント指定	フォント番号	16進整数 10進整数	DHR;\$8000; DHR;32768;
SPB	<b>Start Processing Buffer</b> バッファ処理開始	なし	なし	SPB;

### 1.4.3. コマンドと引き数の指定

表 1 の例で示されたように、コマンドと引き数は常にセミコロン記号 (;...) で区切ります。引き数を必要とするコマンドでは、すべての引き数が定義されなければなりません。例えば、ボックス描画 (DBOX) コマンドは、X開始位置、Y開始位置、幅、高さの 4 つの引き数が必要です。コマンドを指定する順序も重要です。ヘッダとプロセッサはコマンド列の最初になければならず、位置決めコマンドは位置決めされるオブジェクトより前になければなりません。又、オブジェクトはラベル上に配置される前に定義しなければなりません。そしてコマンド列は TRM; で終わらなければなりません。

### 1.4.4. データの指定

データには、プリンタにあるフォントの中のいずれかの英数字、または漢字の文字を使うことが出来ます。データもコマンドのようにセミコロン (;...) で区切り、さらに、ダブルクォーテーションマーク (") で囲まなければなりません。正しいデータの指定のしかたを次に例示します。

DHR;1;"Label One"; BCST;"\*25D9B01\*";BSTP;  
"1st Entry"; "1234567890";



## NITTO DENKO

### 1.4.5. コメント

プログラマーのコメントは、#記号 (#..#) とセミコロンで区切ってコマンド列に入れることができます。コメントには次の3つの規則があります。

- ヘッダ前にコメントは挿入できない。
- コメントは1つのコマンドの引き数の中には挿入できない。
- テキスト文字列内に挿入できない。
- ヘッダ内及び機能設定コマンド内に入れられない。

以下に正しい例と間違っただけの例を示します。

#### 正しい例

```
~^"FILE"; 1; 0; 100; 0; #Header#;  
VBR; 10; HBR; 20; #Starting Point#;  
DHR; 1; "Data Field One"; #1#;
```

#### 間違っただけの例

```
~^"FILE"; #Header#; 10; 50; 0;  
VB # Starting Point#; 10; HBR; 20;  
DHR; 1; "Data Field #1# One";
```

#..#で囲まれてデータは無視され、ラベルには印刷されません。ただし、ラインプリントモードでコマンド列を印刷するときは、コメントも一緒に印刷されます。

(1.6.2.参照)

#### 注

コメントやデータを指定する時には、セミコロンの前に、コメントの終わりの#や、データの終わりの“ ”を入れるのを忘れてはなりません。終わりを示す記号を付け忘れますと、何も印刷されません。

## 1.5. プログラムの書式

**RCL** は、構造化されたプログラミング言語で、あるコマンドを使うためには、それに先立って関連する他のコマンドを使わなければならないことがあります。プログラムのフローチャートを使うと、コマンドの構成とコマンド間の関係を図示できます。

ここでは、必ず必要なコマンドと、連続番号（連番）をつけるために必要な事項について説明します。

### 1.5.1. 必要なコマンドとヘッダ

すでに説明したように、**RCL** では、各ラベルに対して必ず必要な3つのコマンドがあります。ヘッダ、**SPB** と **TRM** です。これらはコマンド列のヘッダとプロセッサ、コマンド列終了に対応します。これらのうち1つでも欠けると、なにも印刷されません。

オブジェクト（バーコードや文字列など、印刷する項目）を制御するコマンドのいくつかは、セットで使われなければなりません。例えば、バーコードを作るには、少なくともバーコード選択 (**BSYM/BDEF**)、バーコード開始 (**BCST**)、バーコード停止 (**BSTP**) 3つのコマンドが必要です。

## NITTO DENKO

ベクトルフォントで文字を印刷するには、ベクトルフォント指定 (**DHR**)、文字高さ設定 (**DCH**)、文字幅設定 (**DCW**)、文字間スペース (**ICS**) の4つのコマンドが最小限必要です。

位置移動のコマンドでは、水平絶対位置移動 (**HBR**) と垂直絶対位置移動 (**VBR**) をラベルごとに少なくとも1組はペアで使用しなければなりません。その他のコマンドは任意に目的に応じて追加したり除外しても構いません。

### 1.5.2. 連番データの印刷

複数のラベルを印刷するときに、バーコードやテキスト・フィールド内のデータは、連番 (連続番号) として印刷することができます。そのためには、マーク (**MRK**)、リターン (**RET**)、リピート枚数設定 (**BCLC**) を含む多数のコマンドを発行しなければなりません。バーコードに対しては、バーコード連番インクリメント・デクリメント設定 (**BCID**) とバーコードフィールド連番長さ保存 (**BSAL**) を定義しなければなりません。またテキストに対しては、テキストフィールド連番インクリメント・デクリメント設定 (**IDF**) とテキストフィールド連番長さ保存 (**SAL**) を定義しなければなりません。これらのコマンドの他に、枠内消去 (**DWBX**) コマンドを使って繰り返し印刷する領域をクリアしなければなりません。連番について詳しくは、第11章を参照ください。

## 1.6. RCL を使用するために

**RCL** は、シリアルインタフェースやセントロニクス・パラレルインタフェースを持ったコンピュータシステムで使用できるように設計されています。ここでは、**RCL** を活用するためのファイル管理とラインプリンタについて説明します。

### 1.6.1. ファイルの作成・保存と利用

プログラム・ファイルはエディタを使ってホスト上に作成します。このファイルは、プリンタに直接送るか、あるいは後でプリンタでダウンロードするためにホスト側で保存しておきます。すべてのプログラム・ファイルは、後で使用できるように、ホスト・コンピュータのファイル管理ユーティリティを使って保存しておかなければなりません。

### 1.6.2. ライン・プリンタの利用

プリンタでラベルを印刷する前、あるいは印刷後に、コマンド列を検査するとたいへん役に立ちます。このために、ラインプリンタまたはシリアルプリンタを使います。コマンド列を正しく接続されたプリンタに送信すると、そのコマンド列が印刷されます。これはプログラムをハードコピーしたい時や、フルスクリーンエディタがないときに役に立ちます。

## 2. ヘッダ

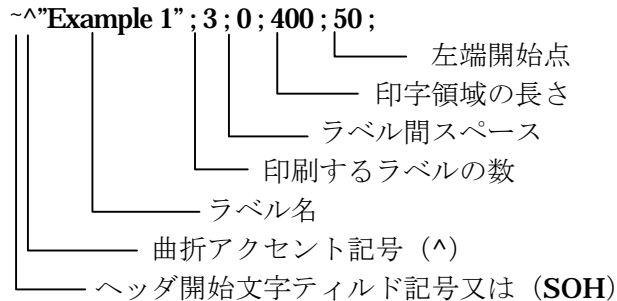
**RCL** のプログラムは、ヘッダで始めなければなりません。ヘッダには、名前、長さ、開始位置、印刷するラベル数など、ラベルについての重要な情報を定義します。これらの情報を定義しないと、他のラベルデータを指定することができません。

### 2.1. 構文

ヘッダには次の7つの部分があり、すべてを定義しなければなりません。

1. ヘッダ開始文字 (**SOH**)
2. 曲折アクセント記号 (^)
3. ラベル名 (引用符 “” で囲む)
4. 印刷するラベルの数
5. ラベル間スペース (未使用)
6. 印字領域の長さ (**Y** ピクセル単位で指定)
7. 左端開始点 (**X** ピクセル単位で指定)

ヘッダ開始文字を除いて、これらの各フィールドは、**RCL** の他のフィールドと同じように、下の例に示すようにセミコロン記号 (;) で区切らねばなりません。



フィールドの値がゼロであっても、それぞれのフィールドはすべて定義しなければなりません。

### 2.2. ヘッダ開始文字

ヘッダ開始文字には、ティルド記号 (~=**7EH**) と **SOH** (**01H**) のどちらか片方を使用します。両方同時には使えません。この開始文字の直後に曲折アクセント記号 (^=**5EH**) を付加します。

ヘッダ開始文字はプログラムの最初の文字でなければなりません。

**注**

**7EH** の **H** は 1 バイトの 16 進数であることを示しています。

## NITTO DENKO

**SOH** またはティルド記号 (~) 前にアスタリスク記号 (\*) があると、プリンタは拡張メモリ通信モードに切り換わります。

**SOH** またはティルド記号 (~) の前に機能設定コマンド、または **DIR** や **LOAD** といった拡張メモリ関連コマンドがあると、プリンタはそのコマンドを実行します。詳しくは **12.1.章**、**14 章**をご覧ください。

### 2.3. ラベル名

ヘッダの第2フィールドは、ラベルの名前です。ラベル名は必要な長さだけ定義することができ、常に引用符 (“...”) で囲みます。

#### 注

ラベル名は引用符に囲まれていなければなりません。引用符がありませんとコマンド列が処理されないので、プリンタはそのコマンド列に対して何もしません。

#### 2.3.1. ラベル名の形式

ラベル名はラベルを識別するために使います。

ラベル名には **ASCII** 文字が使えますが、引用符は名前の始めと終わりにだけ使います。

正しいラベル名の例は次の通りです。

**"Labell" "ALAG20" "!@#\$%^&\*()\_+" "test 2"**

拡張メモリ通信モードでは、ラベル名はファイル名としても使われます。拡張メモリに **RCL** のコマンド列を書き込むように指示すると、プリンタに送ったコマンド

列は、テキスト・ファイルとして拡張メモリに書き込まれます。

その後、拡張メモリ通信モードの、\*コマンドで拡張メモリ上のファイルが指定されると、このラベル名と \*コマンドで指定したファイル名を比較してデータを捜します。

このとき、ラベル名は以下のように扱われます。

- 1) ラベル名が 9 文字以上の場合、最初の 8 文字をファイル名と比較します。  
"NameOfFileA" は "NameOfFi" とみなされます。
- 2) スペースコード (20<sub>H</sub>) を含むラベル名は、スペースコードまでをファイル名と比較します。"File A" は "File" と同じです。

#### 注

コマンド列を拡張メモリに書き込む場合は、トラブルを避けるためにラベル名を 8 文字以下にしてください。

## NITTO DENKO

次の例は **2** 組の引用符が使われているので、間違った名前です。

””TEST3””

### 2.4. 印刷するラベルの数

これは **1** 回の印刷を実行するとき印刷するラベルの総数です。最小値は **1** で、最大値は **16,777,215** です。

### 2.5. ラベル間スペース

現在のところこの機能は備わっていないので、ゼロを指定してください。

### 2.6. 印字領域の長さ

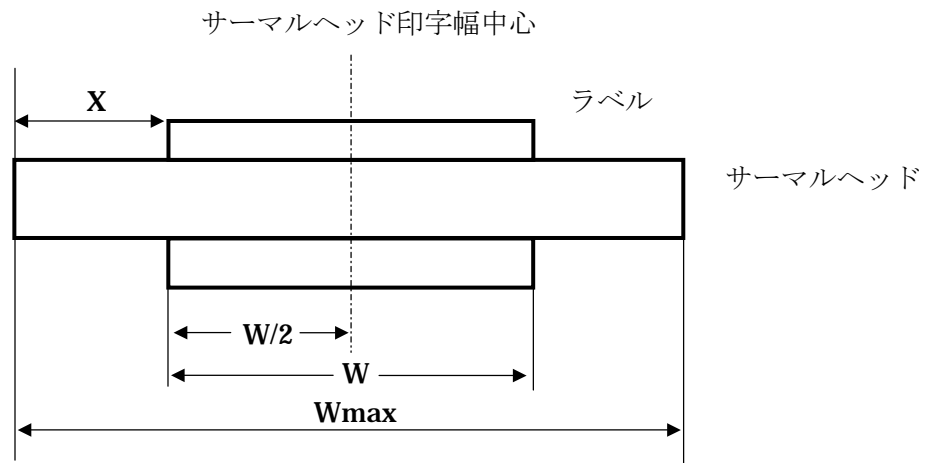
最初から最後までオブジェクトを印刷するためのプリント・バッファの全長を垂直ピクセル (**0.254mm**) 単位で指定します。これはラベルそのものの物理的な長さではなく、ラベルの上で実際に印刷される領域の長さです。原点位置は左上隅です。印字領域の長さはプリンタにより違います。印字長は、プリンタ仕様を参照ください。

### 2.7. 左端開始点

このパラメータを使うと、水平方向の印刷開始点（座標原点）を設定できます。プリンタは、機種によりラベルのセット方法が左合わせのものとセンター振り分けのものがあります。左合わせのプリンタの場合は、常に **0** を指定します。センター振り分けのプリンタでは、使用するラベルの幅がプリンタの最大印字幅より小さい場合には、次の式で求められる値を指定します。センター振り分けのプリンタでもラベル幅がプリンタの最大印字幅よりも大きい場合は、**0** を指定します。

$$X = ((W_{\max} - W) / 2) / 0.127$$

**X:** 左端開始点  
**W:** ラベル幅 (mm)  
**Wmax:** プリンタの最大印字幅 (mm)



左端開始点が設定されると、水平絶対位置移動（**HBR**）コマンドは、この左端開始点を基準にして計算されます。さらに、水平相対位置移動（**HPR**）コマンドは **HBR** に対して相対的に計算されます。

**注**

各々プリンタのラベルセット方法と最大印字幅（**Wmax**）は、**18**章のプリンタ仕様をご覧ください。

### 3. プログラム・コントロール

---

コマンド列は始まりと終わりの位置を指示しなければなりません。  
開始位置は、コマンド・ヘッダに続く第1行目の第1文字で定義します。終わりの位置は、コマンド列の最終行の最終コマンドです。3.1では、プログラム処理を制御する次の1つのコマンドについて説明します。

バッファ処理開始 (Start Processing Buffer:SPB)

バッファ処理再開 (ReStart Processing Buffer:RSPB)

コマンド列 (バッファ処理) 終了 (TeRMinate Processing Buffer:TRM)

ブレーク (Break:BRK)

#### 3.1. バッファ処理

プログラムの処理は、バッファ処理開始 (SPB)、バッファ処理再開 (RSPB)、コマンド列終了 (TRM)、ブレーク (BRK) の4つのコマンドで制御します。このうち、SPB と TRM の2つは、すべてのプログラムで必要です。RSPB は必要に応じて使用するコマンドで、処理開始前にプリントバッファから前のコマンド列のビットマップ・イメージをクリアしたくない場合に SPB の代わりに使います。BRK は長いコマンド列を複数のブロックに分割するために使います。

##### 3.1.1. バッファ処理開始 (Start Processing Buffer : SPB)

SPB コマンドはプリント・バッファをクリアしてコマンド列の処理を開始します。

SPB に続くすべてのコマンドとデータは、ヘッダであらかじめ定義されたラベルのデータとして処理されます。SPB の位置は次の例に示すように、ヘッダのデータの直後です。

```
~^"Example1";1;0;400;50;
```

```
SPB;
```

##### 3.1.2. バッファ処理再開 (ReStart Processing Buffer : RSPB)

RSPB コマンドは SPB と類似していますが、RSPB はプリント・バッファをクリアしません。RSPB は SPB の代わりに次の2つの方法で使われます。

第1に、バッファをクリアしないので、直前に印刷したラベルイメージを消去しないで再処理するのに使うことができます。これはより高速な処理能力になります。

2番目の使い方は、ブレークコマンド (3.1.4 参照) との併用です。SPB と RSPB は、バーコードや他のラベル・データを連番にして印刷するために使います。

##### 3.1.3. バッファ処理終了 (TeRMinate processing buffer : TRM)

TRM コマンドは、コマンド列の終わりを示し、プリンタにラベルを処理して印刷を開始するよう命令します。TRM コマンドは、プリンタのバッファにプリントする内容がある時に、それを印刷するために使います。

TRM はコマンド列の最後のコマンドでなければならず、その直後に次の例のようにセミコロン (;) と¥をつけます。

## NITTO DENKO

```
~^ "Example1";1;0;400;50;
```

```
SPB;
```

```
TRM;¥
```

ヘッダと **SPB** コマンド、**TRM** コマンドは、最小限必要なものです。これらの3つのどれか1つでもないと、**RCL** は処理を開始しません。もちろん、上に示した例は、バーコードや文字、矩形といったラベルのオブジェクトが定義されていないので、何も印刷しません。しかし、このコマンド列を送ると、プリンタはコマンド列を処理し、セットされているラベル用紙を「印字領域の長さ」のパラメータで指定した距離だけ、移動させます。

### 3.1.4 ブレーク (BreA : BRK)

**RCL** は、ニモニックを **16** 進数のコマンド列に変換して、サイズを約 **1/3** に圧縮した後に一度制御用バッファメモリーに格納します。長いプログラムの場合には、このメモリ容量を越えて書き込まれることがないように **TRM** コマンドの代わりにブレークコマンド (**BRK**) を使ってコマンド列を複数のブロックに分割しなければなりません。メモリ容量については、**18** 章のプリンタの仕様をご覧ください。

ブレークコマンド (**BRK**) は、プリンタに対して **BRK** で終了するコマンド列を処理するように指示します。このため、次のブロックを **RSPB** で始めることによりオーバーレイが可能になります。そして、それ以降のブロックで **TRM** を受信するまで印刷しないで待機します。必要ならば、**BRK** は何回でも使えます。**BRK** はコマンド列の **TRM** のあるべき位置に置かなければなりません (最後の **TRM** は除く)。そして、最初のバッファ処理開始コマンドを除いて、**SPB** を **RSPB** に置き換えなければなりません。**TRM** 同様、**BRK** の直後には ¥ (バックスラッシュ) を続けなければなりません。

ロゴ (第 **13** 章参照) を印刷する 2 つのコマンド列を、**BRK** と **RSPB** を使って記述した例を示します。

```
~^ "REGISTER";1;0;50;0;
```

```
SPB;
```

```
VBR;5;HBR;150;
```

```
LOGO;4;19;
```

```
0;$0F;$F8;0;
```

```
0;$70;$07;0;
```

```
1;$C0;1;$C0;
```

```
7;0;0;$70;
```

```
$0E;$0F;$F0;$38;
```

```
$1C;$0F;$F8;$1C;
```

```
$38;$0C;$18;$0E;
```

```
$30;$0C;$18;6;
```

```
$70;$0C;$38;7;
```

```
$70;$0F;$F0;7;
```



**NITTO DENKO**

\$70;\$0F;\$F0;7;

\$30;\$0C;\$38;6;

\$38;\$0C;\$18;\$0E;

\$1C;\$0C;\$18;\$1C;

\$0E;\$0C;\$18;\$38;

\$7;0;0;\$70;

1;\$C0;1;\$C0;

0;\$70;\$07;0;

0;\$0F;\$F8;0;

BRK;¥

~ ^ "COPYRIGHT";1;0;50;0;

RSPB;

VBR;5;HBR;200;

LOGO;4;20;

1;\$FF;\$FF;\$C0;

3;\$FF;\$FF;\$E0;

7;0;0;\$70;

\$0E;0;0;\$38;

\$1C;\$0F;\$F8;\$1C;

\$1C;\$1F;\$FC;\$1C;

\$1C;\$38;\$0E;\$1C;

\$1C;\$70;\$07;\$1C;

\$1C;\$70;0;\$1C;

\$1C;\$70;0;\$1C;

\$1C;\$70;0;\$1C;

\$1C;\$70;0;\$1C;

\$1C;\$70;7;\$1C;

\$1C;\$38;\$0E;\$1C;

\$1C;\$1F;\$FC;\$1C;

\$1C;\$0F;\$F8;\$1C;

\$0E;0;0;\$38;

7;0;0;\$70;

3;\$FF;\$FF;\$E0;

1;\$FF;\$FF;\$C0;

TRM;¥

## 4. 印刷位置移動コマンド

ラベル上に配置されるオブジェクト（印刷するバーコードや文字）の位置は、絶対位置移動コマンドと相対位置移動コマンドで制御します。このコマンドは、本書の第5章～8章で説明するすべてのオブジェクトを正しい位置に印刷するために必要ですので、ここでは印刷位置移動の方法について詳しく説明します。ラベルの上に何か印刷しようと試みる前に、まずこの項を読んでください。

### 4.1. 原点位置

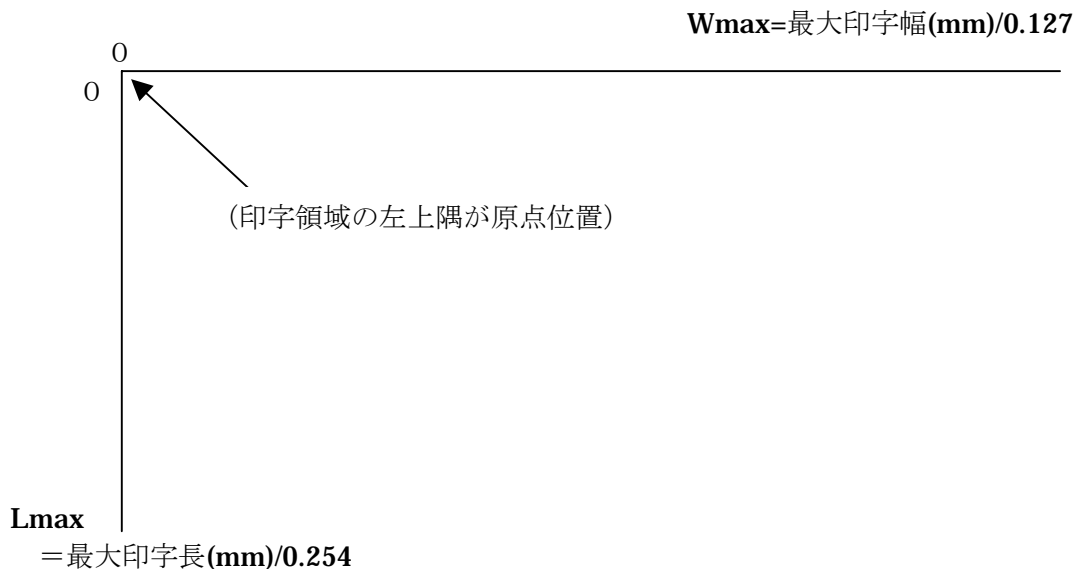
オブジェクトの印刷位置は、**X-Y**の座標系に基づいて計算されます。これは普通の直交座標系とは異なり、負の値は使用できません。原点位置**(0,0)**は印字領域の左上隅です。すべての絶対位置移動コマンドは、この原点位置を基準に計算されます。一方、相対位置移動コマンドは、相対的な位置で計算します。

#### 4.1.1. XとYの大きさ

**X**（水平方向）の大きさは、プリンタの最大印字幅によって決まります。これを水平ピクセル単位**(0.127mm)**で割った値が水平ピクセルの最大値です。オブジェクトは**0**～水平最大範囲の平面上のどこかに印刷することができます。

**Y**（垂直方向）の寸法は、垂直ピクセル単位**(0.254mm)**で計算します。印字領域の最大長（印刷できる範囲）は、最大印字長（mm）を垂直ピクセル単位**(0.254mm)**で割った値になります。

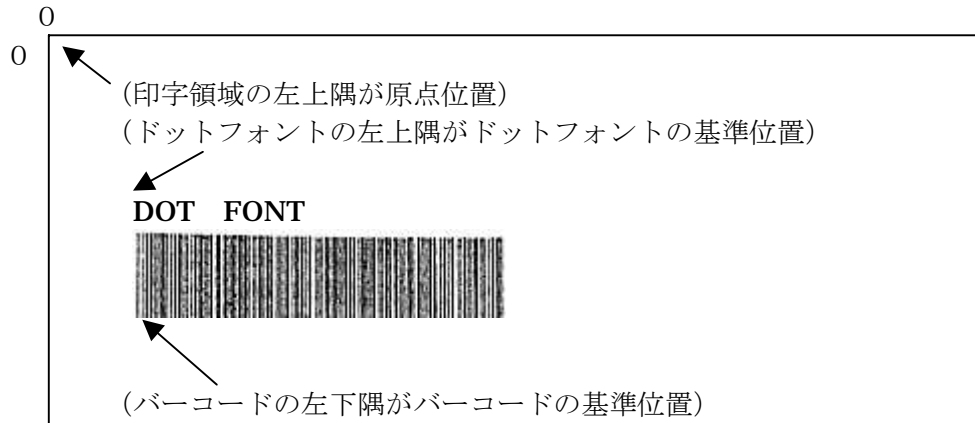
**RCL**座標系の最大値を以下に示します：（実寸ではありません）。



## NITTO DENKO

### 4.2. 印刷開始基準位置

オブジェクトの種類により、印刷開始基準位置は異なります。ドットフォント文字、線、枠、ロゴ、2次元シンボルのマトリックスタイプは、左上隅が基準位置です。ベクトルフォント文字、バーコードは左下隅が基準位置です。



### 4.3. 絶対位置移動コマンド

オブジェクトは、水平絶対位置移動 (**Horizontal Base Reference:HBR**) と垂直絶対位置移動 (**Vertical Base Reference:VBR**) を使用して **X-Y** 座標上の絶対位置に配置されます。このコマンドは、水平位置と垂直位置を決定し、その後、それに基づいてすべての位置が計算されます。

#### 4.3.1. 水平絶対位置移動 (**Horizontal Base Reference : HBR**)

**HBR** コマンドは、**X** 軸上の絶対位置を指定します。

大きさは水平ピクセル (**0.127mm**) 単位で指定します。指定できる値の範囲は0～最大印字幅 (mm) /**0.127** です。

**HBR** コマンドが定義されていない場合、水平絶対位置の初期値は **20** です。

このコマンドの構文は以下の通りです。

**HBR;XX;**

**XX**=水平ピクセル (**0.127mm**) 数

次の例では、水平絶対位置を **12.7mm** に設定します。

**HBR;100;**

#### 4.3.2. 垂直絶対位置移動 (**Vertical Base Reference : VBR**)

**VBR** コマンドは、**Y** 軸上の絶対位置を指定します。

大きさは垂直ピクセル (**0.254mm**) 単位で指定します。指定できる値の範囲は、0～最大印字長(mm)/**0.254** です。

## NITTO DENKO

**VBR** コマンドが定義されていない場合、垂直絶対位置の初期値は **50** です。

このコマンドの構文は以下の通りです。

**VBR;YY;**

**YY**=垂直ピクセル (**0.254mm**) 数

次の例では、垂直絶対位置を **3.81mm** に設定します。

**VBR;15;**

### 4.4. 相対位置移動コマンド

水平相対位置移動 (**Horizontal Position Relative:HPR**) と垂直相対位置移動 (**Vertical Position Relative:VPR**) の2つのコマンドは、最後の印刷位置 (カーソル位置) からの相対的な位置にオブジェクトを配置するために使います。これは、オブジェクト同志の位置関係で、あるいはラベル上の絶対位置を基準にオブジェクトの位置を決定しなければならぬ時に役に立ちます。

#### 4.4.1. 水平相対位置移動 (**Horizontal Position Relative : HPR**)

**HPR** コマンドは、印刷位置を最後のカーソル位置から **X** 軸方向に相対的に移動させます。**HPR** の最小値は1水平ピクセル (**0.127mm**) で、最大値は下記の公式で計算できます。

**HPR max.=W max-HBR**

例えば、**Wmax** が **1008** で **HBR** で水平絶対位置を **100** に設定した場合、カーソル位置の最大値は **1008-100**、つまりは **908** ピクセルです。

**HPR** コマンドを何回か発行すると、それぞれのコマンドで指定した移動量を加算した量だけ位置が移動します。以下の例では、カーソルが **X** 軸に沿って **2** 回移動します。

**HBR;20;HPR;30;HPR;100;**

この場合、最初の位置は **X** 軸上で **20** で、カーソルはまず位置 **50** まで **30** ピクセル移動します。次に位置 **150** まで **100** ピクセル移動します。**HPR** コマンドには累加性があるため、プログラムで使用するすべての **HPR** コマンドで指定した移動量の総和を正確に把握することが大切です。

移動した後の絶対位置が負にならなければ、**HPR** のパラメータに負の値を指定することができます。例えば、次のコマンドは有効です。

**HBR;100;HPR;-50;**

このコマンドの構文は次の通りです。

## NITTO DENKO

**HPR;XX;**

**XX**=水平ピクセル (0.127mm) 数

### 4.4.2 垂直相対位置移動 (Vertical Position Relative : VPR)

**VPR** コマンドは、最後のカーソルの位置から **Y** 方向にカーソルの位置を相対的に移動させます。**VPR** の最小値は1ピクセル (0.254mm) です。最大値はヘッダ (2.6 参照) 内で指定される印字領域の長さによって決まり、以下の式を使って計算できます。

**VPR max.** = (印字領域の長さ) - **VBR**

例えば、ヘッダでラベルの印字領域の長さを **100** ピクセルに定義した場合、**VBR** を **10** ピクセルに設定すると、ラベル上の最大の **VPR** 値は **100-10**、つまり **90** ピクセルです。

**HPR** と同様に、**VPR** コマンドには累加性があります。この例を以下に示します。

**VBR;50;VPR;25;VPR;60;**

上の例では、**Y** 軸上の絶対位置 **50** から、カーソルはまず **25** ピクセル下へ移動して位置 **75** に移動し、次に **60** ピクセル下へ移動して位置 **135** に移動します。移動した後の絶対位置が負にならなければ、**VPR** に負の値を指定することができます。

このコマンドの構文は以下の通りです。

**VPR;YY;**

**YY**=垂直ピクセル (0.254mm) 数

### 4.4.3. 水平位置リセット (End Of Line : EOL)

**EOL** コマンドは、水平印刷位置を、現在設定されている絶対水平位置に再設定します。これは復帰 (キャリッジリターン) と同じ効果があり改行は起きません。

このコマンドの構文は次の通りです。

**EOL;**

このコマンドは、復帰とほとんど同じように使われます。通常、テキストの行末に置いて、カーソルを現在の行の始めに戻らせる効果があります。このコマンドは垂直方向の印刷位置には影響を与えません。したがって、次の行にテキストを印刷する場合、垂直位置移動コマンドを発行してテキストを次のプリント位置まで下へ移動させる必要があります。

以下に **EOL** の使用例を示します。

~ ^ "IL4-6";1;0;100;0;

**SPB;VBR;505;HBR;100;**

## NITTO DENKO

```
“This is the first line.”;EOL;VPR;30;  
“This is the next line.”;  
TRM;¥
```

```
This is the first line.  
This is the next line.
```

EOL が取り除かれると結果は以下のようになります。

```
This is the first line.  
This is the next line.
```

### 4.4.4. ホームポジション (HOME)

HOME コマンドは、水平印刷位置と垂直印刷位置の両方に作用して、カーソルを HBR と VBR で最後に設定した位置へ戻します。

このコマンドの構文は以下の通りです。

```
HOME;
```

次の例では、テキストとアンダーラインを確実に同じ位置から開始するようにするために HOME が使われています。

```
~^”IL4-7”;1;0;200;0;  
SPB;VBR;100;HBR;200;  
“12345”;HOME;  
HLT;1;DHL;;0;0;112;  
TRM;¥
```

```
1 2 3 4 5
```

## 5. 水平線と垂直線

水平線太さ設定 (**Horizontal Line Thickness:HLT**)、水平線描画 (**Draw Horizontal Line:DHL**)、垂直線太さ設定 (**Vertical Line Thickness:VLT**)、垂直線描画 (**Draw Vertical Line:DVL**)の4つのコマンドを使ってさまざまな長さや太さの水平線や垂直線をラベル上に描くことができます。

5.1.では水平線、5.2.では垂直線を説明します。

### 5.1. 水平線

ラベル上のどの位置でも、長さが **0.127mm** から最大印字幅までの範囲、太さが **0.254mm** から **25.4mm** までの範囲の水平線を描くことができます。はじめに水平線の太さを設定し、それからラベル上に線を描きます。水平線の太さ設定 (**HLT**) は、枠 (ボックス) の水平部分の線の太さを設定するためにも使います。プリンタの最大印字幅は、**18** 章プリンタ仕様をご覧ください。

#### 5.1.1. 水平線太さ設定 (**Horizontal Line Thickness:HLT**)

このコマンドは、水平線の太さと枠の水平線の太さを設定します。線の太さは、線あるいは枠を描く前に定義しなければなりません。このコマンドの構文は次の通りです。

**HLT;YY;**

**YY**=垂直ピクセル(**0.254mm**)数

設定できる値は、1ピクセル(**0.254mm**)から 100ピクセル(**25.4mm**)までの範囲です。**HLT** が設定されていない場合、初期値は 3ピクセル(**0.762mm**)です。一度 **HLT** が設定されると、新たに **HLT** が設定されるまですべての水平線は同じ太さで描かれます。

#### 5.1.2. 水平線描画 (**Draw Horizontal Line : DHL**)

**DHL** は、プリンタでラベル上に水平線を引くためのコマンドです。このコマンドには **3** つの引き数が必要です。

**DHL;X;Y;XX;**

**X**=開始点の **X** 軸方向の位置

**Y**=開始点の **Y** 軸方向の位置

**XX**=水平ピクセル(**0.127mm**)単位で設定する線の全長

現在のカーソル位置から **X** 及び **Y** 離れた位置から線が描かれます。現在のカーソル位置から線を描きたい場合、**X** と **Y** に **0** を指定しなければなりません。

#### 注

**HLT**、**VLT** で太線を描いた場合、熱転写ヘッドが局所的にオーバーヒートを起こし、熱転写リボンのベースフィルムを溶かし、リボンが切れたり印字品質の低下を起こしたりすることがあります。この場合、太さを細くしてください。

## NITTO DENKO

線の位置を決めるときは、指定した開始点の位置が、線の左上になることに注意してください。太さが増すにつれて、水平線が左上開始点からラベルの下側に広くなります。水平線と垂直線のもう 1 つの特徴は、線が引かれた後に、カーソル位置が最初の位置から移動しないことです。カーソルは線が引かれると直ちに前の位置に戻ります。

### 5.2. 垂直線

垂直線を描くときには、垂直線太さ設定 (VLT) と垂直線描画 (DVL) の 2 つのコマンドを使います。垂直線は、太さ 1 ピクセル(0.217mm)から 200 ピクセル(25.4mm)まで、長さ 2 ピクセル(0.254mm)から最大印字長までの範囲内で描くことができます。はじめに VLT を使って太さを定義し、次にその太さの線を DVL を使って引きます。プリンタの最大印字長は、18 章プリンタ仕様をご覧ください。

#### 5.2.1. 垂直線太さ設定 (Vertical Line Thickness : VLT)

VLT コマンドは、垂直線の太さと枠の垂直線の太さを設定するために使います。このコマンドの構文は以下の通りです。

VLT;XX;

XX=水平ピクセル(0.127mm)数

線の最小の太さは、1 ピクセル(0.127mm)で、最大は 200(25.4mm)です。VLT が発行されていない場合、初期値は 2 ピクセル(0.254mm)です。一度 VLT が設定されると、新たに VLT を発行して垂直線の太さを変更するまで、同じ太さになります。

#### 注

水平線と垂直線を同じ太さで引くには、垂直線太さのピクセル数を水平線のピクセル値の 2 倍にしてください。これは、プリンタのピクセルは、縦が横の 2 倍の長さだからです。例えば HLT を 2 にすると、線の太さは  $2 \times 0.254 = 0.508\text{mm}$  になるので、VLT は 4 にしてください。すると  $4 \times 0.127 = 0.508\text{mm}$  で同じ太さになります。

#### 5.2.2. 垂直線描画 (Draw Vertical Line : DVL)

垂直線は DVL コマンドを使ってラベル上のどこにでも描くことができます。パラメータは 3 つあり、いずれも省略することはできません。

DVL;X;Y;YY;

X=開始点の X 軸方向の位置

Y=開始点の Y 軸方向の位置

YY=垂直ピクセル(0.254mm)単位で指定する線の全長



## NITTO DENKO

現在のカーソル位置から **X** および **Y** 離れた位置から線が描かれます。現在のカーソル位置から線を描きたい場合、**X** と **Y** に **0** を指定しなければなりません。

### 5.3. 対角線

対角線は、対角線描画 (**Draw Diagonal Line:DDL**)

**DHL** コマンドや **DVL** コマンドは、**HLT** コマンドや **VLT** コマンドで太さを設定しますが、**DDL** で描く線の太さはプリンタ・ソフトウェアによって設定されます。**DDL** コマンドでプリントされる線の幅は、描く線の方法 (傾斜) によって異なり、約 **0.254mm** から約 **0.356mm** まで変化します。

#### 5.3.1. 対角線描画 (**Draw Diagonal Line : DDL**)

対角線はラベル上で任意の位置から任意の長さで引くことができます。

**DDL** コマンドの書式は次の通りです。

**DDL:X1;Y1;X2;Y2;**

**X1** = 開始点の **X** 軸上の位置

**Y1** = 開始点の **Y** 軸上の位置

**X2** = 終了点の **X** 軸上の位置

**Y2** = 終了点の **Y** 軸上の位置

現在のカーソル位置から開始点 (**X1**, **Y1**) と終了点 (**X2**, **Y2**) で指定した長さを加えた点が始点と終点です。カーソル位置は、対角線を引いた後も変わりません。

**DDL** コマンドを使ってプリントした例を次に示します。

~ ^ "DDL";1;0;400;0;SPB;UT0F;4100;

**HBR; 0;VBR;100;**

**DDL;400; 25;100; 75;**

**DDL;100; 75;400;125;**

**DDL;400; 125;600; 75;**

**DDL;600; 75;400; 25;**

**DDL;400; 38;100; 75;**

**DDL;100; 75;400;113;**

**DDL;400; 113;600; 75;**

**DDL;600; 75;400; 38;**

**DDL;400; 50;100; 75;**

**DDL;100; 75;400;100;**

**DDL;400; 100;600; 75;**

**DDL;600; 75;400; 50;**

**DDL;400; 63;100; 75;**

## NITTO DENKO

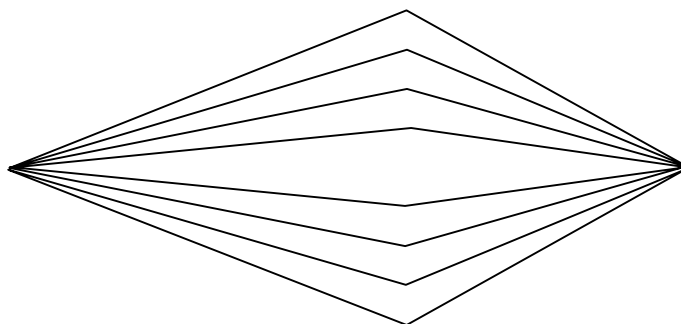
DDL;100; 75;400; 88;

DDL;400; 88;400; 88;

DDL;400; 88;600; 75;

DDL;600; 75;400; 63;

TRM;¥



### 注

DDL コマンドは、 $X1=X2$  (水平線) または  $Y1=Y2$  (垂直線) の場合、線を描きません。このような条件の場合には、DHL コマンドまたは DVL コマンドを使用してください。

## 6. 枠（ボックス）

---

RCL では、ラベル上にいろいろな大きさの枠（ボックス）を簡単に描くことができます。描くことのできる枠の種類は、輪郭だけの枠、黒枠（完全に塗り潰した長方形）、反転枠（反転印刷）、白枠（すでに印刷された領域を消去）の4種類です。これらの枠を描くコマンドは、枠描画（**Draw BOX:DBOX**）、塗りつぶし（**Draw Black BoX:DBBX**）、反転枠描画（**Draw Complement BoX:DCBX**）、枠内消去（**Draw White BoX:DWBX**）です。はじめに、大きさや位置の決定などの、これらのコマンドに共通した事項を説明し、その後で各コマンドについて詳しく説明します。

### 6.1. 枠コマンドの構文

4つの枠のコマンドの構文は同じです。枠を描く度に全部で4つのパラメータを指定しなければなりません。これらのパラメータは次の通りです。

**DBOX;X;Y;XX;YY;**

**X**=枠の描画開始点の **X** 方向の位置

**Y**=枠の描画開始点の **Y** 方向の位置

**XX**=水平ピクセル数で指定した枠の幅

**YY**=垂直ピクセル数で指定した枠の高さ

枠はラベル上の任意の位置に描くことができます。枠の最小寸法は、幅 **0.508mm**、高さ **0.508mm** です。最大寸法は作成するラベルの大きさによって決まります。

### 6.2. 枠の描画位置の指定

枠の位置は、枠のコマンドを実行するときのカーソルの位置と、コマンドのパラメータで指定した水平及び垂直方向の印刷位置によって決まります。枠の印刷開始点は、現在のカーソルの位置プラス（またはマイナス）枠コマンドの最初の**2**つのパラメータで指定された値です。

例えば、現在のカーソル位置が **10-20** でつぎの **DBOX** コマンドが発行されたとします。

**HBR;10;VBR20;**

**DBOX;0;0;600;300;**

すると、水平方向の位置は **10**（水平方向ピクセル）、垂直方向の位置は **20**（垂直方向ピクセル）から枠を描き始めます。枠は、幅 **76.2mm**（**600×0.127mm**）、高さ **76.2mm**（**300×0.254mm**）になります。つまり、**X**の位置が **10** と **610**、**Y**の位置が **20** と **320** の間に長方形が描かれます。

枠を描いた後、カーソル位置は枠を描く前の場所に戻ります。上の例では、カーソルは位置 **10-20** に戻ります。枠や線やロゴだけは、カーソルの位置が変わりません。

## NITTO DENKO

これはいろいろなオブジェクトをラベル上に配置するときに忘れてはならない重要な点です。

### 6.3. 輪郭だけの枠

輪郭だけの枠は、4つの黒線を使って枠の境界線だけを描き、内部には何も描きません。このような枠は、枠描画 (DBOX) コマンドを使って描きます。

#### 6.3.1. 枠作成 (Draw BOX : DBOX)

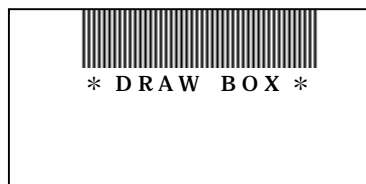
DBOX コマンドは、ラベル上に輪郭だけの枠を描きます。枠の辺にあたる線の太さは、HLT コマンドと VLT コマンドで設定します (5.1.1、5.2.1参照)。太さが設定されないと、水平線は 0.762mm、垂直線は 0.254mm の初期値で描きます。枠の水平方向の辺と垂直方向の辺を同じ太さにするには、HLT コマンドまたは VLT コマンドを使って、枠を描画する前に線の太さを設定してください。

#### 6.3.2. 枠描画コマンドの使用法

バーコードラベルでは、DBOX コマンドは重要なコマンドです。第1に、ラベルの外側の境を描くのに使います。

次にその例を示します。

```
~^ "DBOX";1:0;27;266;SPB;UT0F;0354;  
HBR;0;VBR;0;  
HBR; 30;VBR; 17;BDEF; 1;BNEW; 2;BWEW; 5;BICG; 2;BCSH; 17;BCST;  
"*DRAW BOX*";  
BSTP;  
HBR; 30;VBR; 19;DDF; 3; 10;DFM; 1; 1;DF0; 1;DFS; 10;  
"*DRAW BOX*";  
HLT; 1;VLT; 2;  
TRM;¥
```



## 6.4. 塗りつぶし

塗りつぶしは、枠を描いてその中を黒く塗りつぶすものです。塗りつぶすと他のものは見えなくなるので、他のオブジェクトと重なるような領域を塗りつぶしてはなりません。

### 注

大きな枠を描いて塗りつぶす場合、熱転写ヘッドが局所的にオーバーヒートを起こし、熱転写リボンのベース・フィルムを溶かし、リボンを切ったり、印刷品質の低下を起こしたりします。この場合、枠の大きさを小さくするか、その他の方法でヘッドの負担を少なくしてください。

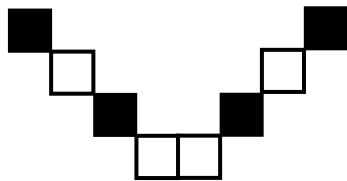
### 6.4.1. 塗りつぶし (Draw Black BOX : DBBX)

DBBX コマンドの構文は、DBOX と同じです。2つのコマンドの唯一の相違は、DBBX は境界を指定すると、内部を塗りつぶすことです。

### 6.4.2. 塗りつぶしコマンドの利用例

次の例は、いくつかの同じ大きさのボックスを描き、半数は外形線だけ (DBOX) で、残りの半数は黒で塗りつぶすプログラムです。

```
~ ^ "DBBX";1;0;200;20;SPB;UTOF;2100;  
HBR; 0;VBR; 0;  
HLT; 5; VLT; 10;  
DBBX;100; 0; 50; 25;  
DBOX;150; 25; 50; 25;  
DBBX;200; 50; 50; 25;  
DBOX;250; 75; 50; 25;  
DBBX;450; 0; 50; 25;  
DBOX;400; 25; 50; 25;  
DBBX;350; 50; 50; 25;  
DBOX;300; 75; 50; 25;  
TRM;¥
```



## NITTO DENKO

### 6.5. 反転枠描画

反転枠は、枠内のすべての黒い部分が白色に反転し、白色の部分が黒色に反転するボックスです。これは枠内のすべてのビットを反転することで実現できます。例えば、1（黒）のビットは0（白）になり、0のビットは1になります。

#### 6.5.1. 反転枠描画 (Draw Complement BoX : DCBX)

**DCBX** コマンドでは反転する領域を指定します。**RCL** では、すべての **ASCII** テキスト、線や枠を反転させることができますが、バーコードは反転できません。

### 6.6. 枠内消去 (領域クリア)

枠内消去は、いくつかのオブジェクトが描かれたラベルの領域をクリアします。

**DWBX** コマンドは、パラメータで指定した領域を消去します。これは主に連続番号（連番）を使用するときを使い、新しい番号を増加（インクリメント）あるいは減少（デクリメント）する前に、前のデータをクリアします（連番については第 11 章をご覧ください）。

#### 6.6.1. 枠内消去 (Draw White BoX : DWBX)

**DWBX** コマンドは、1つのコマンド列で複数のラベルを印刷しようとする時に役に立ちます。すでに定義されたラベルのうち、一般的にはバーコードの部分を **DWBX** で消去し、前のバーコードを新しいデータのバーコードと入れ換えます。連番のために **DWBX** で指定した領域は完全にクリアしますが、ラベルの他のオブジェクトはクリアしないので注意してください。

#### 注

大きな面積の塗りつぶし (**DBBX**)、反転枠画面 (**DCBX**) を使用した印字は避けてください。インクリボンが切れたり、リボンにシワが寄ったりすることがあります。

## 7. バーコード

RCL は、普及しているバーコードのうち、**Code 39**、**Interleaved 2 of 5 (ITF)**、**UPC**、**EAN**、**CodaBar**、**Code93**、**Code128** の7つをサポートします。

バーコードは、**0°**、**90°**、**180°**、**270°** の4つの方向に印刷することができます。**0°** と **180°** 方向のバーコードは各エレメントの長辺がラベル搬送方向に対して平行に並ぶためピケットフェンスバーコードと呼ばれます。これに対し **90°**、**270°** の方向のバーコードは各エレメントの長辺がラベル搬送方向に対して直角に並ぶためラダーバーコードと呼ばれます。

バーコードは RCL の心臓部ですので、バーコードを作成するために役に立つ多くのコマンドがあります。すべてのバーコードは少なくとも3つのコマンドを必要とします。それは、バーコード選択 (**Barcode SYMBOL:BSYM** / **Bar code DEFINITION:BDEF**)、バーコードデータ開始 (**Bar Code StarT:BCST**)、バーコードデータ終了 (**Bar code StoP:BSTP**) で **7.1** と **7.2** で説明します。

さらに、各バーコードはそれぞれ特有のコマンドを必要とします。

ピケットフェンス **Code39** と **ITF** には、バーの細線幅設定 (**Bar Narrow Element Width:BNEW**)、バーの太線幅設定 (**Bar Wide Element Width:BWEW**)、バーコード高さ設定 (**Bar Code Symbol Height:BCSH**) の3つのコマンドが使われます。さらに、**Code39** は、バーコード・インターキャラクタギャップ (**Barcode InterCharacter Gap:BICG**) を使用します。

この2つのバーコードをラダーバーコードで印刷するときは、**BNEW** と **BWEW** と **BICG** の代わりにバーコード密度設定 (**Barcode Characters Per Inch:BCPI**) を使います。

**UPC** と **EAN** には、**UPC** 拡大率設定 (**Upc MAGinfication:UMAG**) が、**Code93** と **Code128**、**CodaBar** には、バーコード密度設定 (**Barcode Characters Per Inch:BCPI**) とバーコード高さ設定 (**Bar Code Symbol Height:BCSH**) が必要です。これらのコマンドは、**7.3**、**7.4**、**7.5** でそれぞれ説明します。

最後にバーコードを連続番号で印刷するには、もっと多くのコマンドが必要になります。それについては第 **11** 章で説明します。

### 注



(ラベル搬送方法)

(ピケットフェンス)

(ラダー)

極力ピケットフェンスのバーコードを使用してください。

高密度のラダーバーコードは、読み取りが出来ないことがあります。

## NITTO DENKO

### 7.1. バーコードの種類

RCL で使用できるバーコードは、Code39、ITF、UPC、EAN、Code93、Code128、CodaBar の7種類で、これらは最も広く使われるバーコードです。

Code39 はチェックキャラクタの有無によって細分されます。また、UPC と EAN もそれぞれ細分されています。

#### 注

プログラマーは、使用するバーコードの最新の規格を調査し、またそこに定められた指針に従う責任があります。

#### 7.1.1. バーコードの選択 (Barcode SYMBOL : BSYM)

BSYM コマンドは、必ず必要なバーコード・コマンドで、使用するバーコードの種類と印刷方向を設定するために使います (BSYM の代わりに、RCL との互換性を保つために用意されている BDEF を使用しても構いません。BDEF については 7.1.3 参照)。バーコードの種類はバーコードを生成する前に選択しなければなりません。コマンドの構文は次の通りです。

#### BSYM;A;B;

A=バーコードの種類を決める整数。下の表によって決まる。

B=バーコードの方向を設定する 1~4 までの整数。

バーコードの種類と、BSYM コマンドのパラメータ値を次に示します。

バーコードの種類	A	バーコードの種類	A
Code39	1	EAN13	16
CodaBar	3	EAN8	17
Code39 (MOD43 付き)	5	EAN13 2 Char.	18
ITF	8	EAN13 5 Char.	19
UPC-A	10	EAN8 2 Char.	20
UPC-A 2 Char.	11	EAN8 5 Char.	21
UPC-A 5 Char.	12	Code128	25
UPC-E	13	Casecode Code128	27
ITF(MOD10 Wait 3 付き)	52	CodaBar(MOD16 付き)	54
UPC-E 2 Char.	14	Code93	30
UPC-E 5 Char.	15		

バーコードの方向	B
0°	1
90°	2
180°	3
270°	4



## NITTO DENKO

### 7.1.2. バーコード選択 (Bar code DEFINition : BDEF)

BDEF コマンドは、BSYM コマンドと同様なコマンドです。BDEF コマンドでは 0° と 270° の 2 方向の印刷だけが可能です。

バーコードの種類は、バーコードを生成する前に選択しなければなりません。コマンドの構文は次の通りです。

**BDEF;A;**

A=1～55 までの整数で、下に示された記号表によって決まります。

バーコードの種類と BDEF コマンドのパラメータ値を次に示します。

バーコードの種類	A	バーコードの種類	A
Code39	1	EAN 8 5 Char.	21
Code39 ラダー	2	Code128	25
CodaBar	3	Code128 ラダー	26
CodaBar ラダー	4	Casecode Code128	27
Code39(MOD43 付き)	5	同上ラダー	28
同上 ラダー	6	Code93	30
ITF	8	Code93 ラダー	31
ITF ラダー	9	UPC A ラダー	40
UPC-A	10	UPC A 2 Char.ラダー	41
UPC-A 2 Char.	11	UPC A 5 Char.ラダー	42
UPC-A 5 Char.	12	UPC E ラダー	43
UPC-E	13	UPC E 2 Char.ラダー	44
UPC-E 2 Char.	14	UPC E 5 Char.ラダー	45
UPC-E 5 Char.	15	EAN 13 ラダー	46
EAN-13	16	EAN 8 ラダー	47
EAN-8	17	EAN 13 2 Char.ラダー	48
EAN 13 2 Char.	18	EAN 13 5 Char.ラダー	49
EAN 13 5 Char.	19	EAN 8 2 Char.ラダー	50
EAN 8 2 Char.	20	EAN 8 5 Char.ラダー	51
ITF(MOD10 Wait3 付き)	52	ITF(MOD10 Wait3 付き)ラダー	53
CodaBar(MOD16 付き)	54	CodaBar(MOD16 付き)ラダー	55

BDEF コマンドも BSYM コマンドも発行されない場合は、規定値として 1 (Code39) を割り当てます。さらに、上の表に列挙された以外の値が指定された場合には、自動的に 1 であるとみなします。一度 BDEF コマンドまたは BSYM コマンドでバーコードを選択すると、新たに別のバーコードが選択されるまで同じバーコードを使用します。

## 7.2. バーコードの印刷

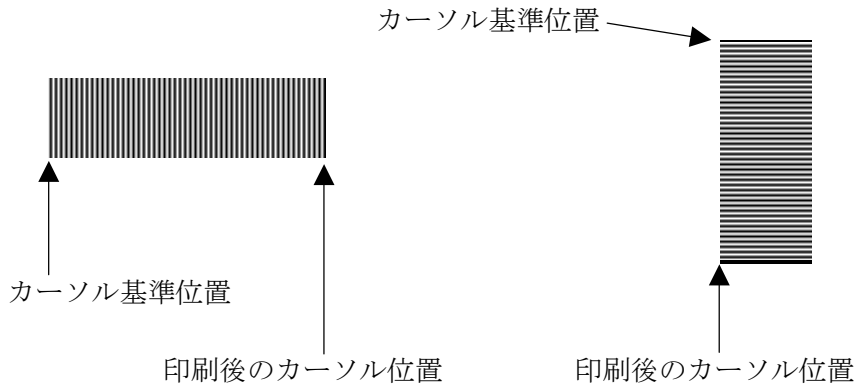
バーコードは、HBR、VBR、HPR、VPR、HOME、EOL コマンドを使って印刷位置を設定します (これらのコマンドについては第 4 章をご覧ください)。バーコードは左下隅がカーソルの基準位置です。

つまり、回転方向 0° のピケットフェンスバーコードは、カーソルから上方右側に向かって印刷され、回転方向 270° のラダーバーコードは、カーソルの下方右側に向かって印刷されます。バーコードの印刷が完了すると、次に印刷されるオブジェクトの基準位置は

## NITTO DENKO

ピケットフェンスバーコードではバーコードの右下隅になり、ラダーバーコードでは左下隅になります。

原点位置は、印字領域の左上隅です。



検証機付きモデルのプリンタでバーコードを正しくスキャンできるようにするには、バーコードの位置を設定する時に、次のような規則を守らなければなりません。

1. バーコードの前後に、少なくとも **3mm** の余白を設けなければなりません。
2. バーコードは、最新の規格に従って設定しなければなりません。特にバーコードの高さ、エレメントの寸法、バーコードの密度に注意してください。

### 7.2.1. バーコード・データの入力

バーコード・データは、常に引用符("...")で囲わなければなりません。バーコードの種類によっては、特定の **ASCII** 文字だけを使ったり、決められた数の文字を入力しなければならないので注意してください。**Code39** と **CodaBar** では、スタートキャラクタとストップキャラクタは、プログラマーが定義しなければなりません。

**CodaBar** では、必要に応じてチェックキャラクタを追加できます。**ITF** と **UPC** と **EAN** では、スタートキャラクタ、ストップキャラクタは自動的に決まります。

**Code128** では、バーコードデータの後ろに **Modulo 103** チェックキャラクタが付加されます。**Casecode Code128** のデータは、制限があります。これについては別の部分で説明します。

印字領域内で印刷できるバーコードの数には制限がありません。

**UPC** と **EAN** の場合、バーコードのデータを可読文字に変換し同時に印刷することができます。**Code93**、**ITF**、**Code39**、**Code128**、**CodaBar** では、可読文字はテキスト・コマンドを使って別に印刷しなければなりません (第 10 章参照)。この場合、バーコードと可読文字は同じデータになるようにしてください。

### 7.2.2. バーコードデータの開始 (Bar Code StarT : BCST)

**BCST** コマンドは、プリンタに対して、このコマンドの直後に続くデータをバーコードとして印刷するように指示します。**BCST** コマンドがないと、そのデータは通常のテキストであるとみなし、それをそのまま印刷します。このコマンドの構文は次の通りです。

**BCST;**

## NITTO DENKO

### 7.2.3. バーコードデータ終了 (BSTP)

**BSTP** コマンドは、**BCST** コマンドと似ていて、プリンタにバーコード・データの終わりを伝えます。このコマンドは、最後のバーコード文字の直後に置かなければなりません。**BSTP** コマンドが発行されない場合、プリンタはどこでバーコードが終わっているかわかりません。このコマンドの構文は次の通りです。

**BSTP;**

次の例は、**Code39** バーコードを選択し、データを入力したものです。

**BDEF;1;BCST;"\*A1ZOB2Y9\*";BSTP;**

### 7.3. Code39とITF

Code39 と ITF のバーコードは、両方とも同じように定義します。BSYM/BDEF を使ってバーコードの種類を定義し、BCST と BSTP を使ってバーコード・データの開始と停止を示し、バーコードを印刷するために、バーの細線幅設定 (BNEW)、バーの太線幅設定 (BWEW)、バーコードの高さ (BCSH) を設定しなければなりません。

Code39 では、バーコード文字間の間隔は、BICG (バーコード・インターキャラクタギャップ設定) コマンドを使って設定することができますが、このコマンドは通常は必要ありません。

前頁で説明した BNEW、BWEW、BICG コマンドは、ピケットフェンスバーコードの密度を設定するのに使います。ラダーバーコードには、これら 3 つのコマンドの代わりにバーコード密度設定 (BCPI) コマンドを使います。

Code39 のバーコードはチェックキャラクタの有無によって 2 種類あります。

BSYM;1;x、BDEF;1、BDEF;2;では、チェックキャラクタが付加されません。つまり、バーコード・データとして引用符で囲ったデータをそのまま、Code39 のバーコードにします。BSYM;5;x、BDEF;5、BDEF;6;では、プリンタは入力バーコード・データからモジュール 43 チェックキャラクタを計算し、ストップキャラクタの直前に付加します。そのため、符号化されたデータの桁は、入力された桁数より多くなります。Code39 を使用する HIBC ラベルは、この種類のモジュール 43 チェックキャラクタを使用します。モジュール 43 チェックキャラクタの計算方法は次の通りです。

1. 各データ文字に、次の表に示した数を割り当てます。

キャラクタ	値	キャラクタ	値	キャラクタ	値	キャラクタ	値	キャラクタ	値
0	0	9	9	I	18	R	27	-	36
1	1	A	10	J	19	S	28	.	37
2	2	B	11	K	20	T	29	スペース	38
3	3	C	12	L	21	U	30	\$	39
4	4	D	13	M	22	V	31	/	40
5	5	E	14	N	23	W	32	+	41
6	6	F	15	O	24	X	33	%	42
7	7	G	16	P	25	Y	34		
8	8	H	17	Q	26	Z	35		

2. すべての文字の値の合計を 43 で割った余りを求めます。

3. 計算した値に等しい文字をチェックキャラクタにします。

例

データが"DATA"の場合、

D=13、A=10、T=29、A=10

$13+10+29+10=62$

$62 \text{Mod} 43=19$

19="J"

したがって、チェックキャラクタをつけたデータは"DATAJ"になります。

このセクションでは、Code39 と ITF について説明します。

## NITTO DENKO

### 7.3.1. バーの細線幅設定 (Bar Narrow Element Width : BNEW)

**BNEW** コマンドは、ピケットフェンス **Code39** と **ITF** のバーコードの細いバーと狭いスペースの幅を設定します。**Code39** では、**BICG** コマンドがない場合、文字間間隔も設定します。

このコマンドの構文は次の通りです。

**BNEW;X;**

**X**= ヘッド密度単位で表したエレメント幅

**BNEW** の最大値と最小値は使用するバーコードの種類によって決まります。より詳しい内容については関連する規格を調べてください。**BNEW** コマンドが発行されない場合の初期値は、**3** です。

### 7.3.2. バーの太線幅設定 (Bar Wide Element Width : BWEW)

**BWEW** コマンドは、ピケットフェンス **Code39** と **ITF** のバーコードの太いバーと広いスペースの幅を設定します。

どちらのバーコードも、細いエレメントに対する広いエレメントの比率は、一般に **2:1** ~ **3:1** の間です。このコマンドの構文は次の通りです。

**BWEW;X;**

**X**= ヘッド密度単位で表したエレメント幅

**BWEW** コマンドが発行されない場合初期値は **9** です。

### 7.3.3. バーコード高さ (Bar Code Symbol Height : BCSH)

**BCSH** コマンドは、可読文字を含まないバーコードの高さを設定します。

**Code39**、**ITF**、**Code93**、**Code128**、**CodaBar** などのバーコードでは、このコマンドで高さを設定しなければなりません。**UPC** と **EAN** のバーコードでは、**UMAG** コマンドによって高さが自動的に決定されるので設定する必要はありません。しかし、**UPC** や **EAN** のバーコードで、印刷するバーコードの高さを制限したい場合には **BCSH** を使います。これにより、正規の高さより低いバーコードを印刷できます。

このコマンドの構文は次の通りです。

ピケットフェンスバーコードの場合

**BCSH;YY;**

**YY**=垂直ピクセル(**0.254mm**)の単位で設定したバーコードの高さ

ラダーバーコードの場合

**BCSH;XX;**

**XX**=水平ピクセル(**0.127mm**)の単位で設定したバーコードの高さ

## NITTO DENKO

BCSH の最大値と最小値は、バーコードの種類によって異なります。したがって、最近の関連する規格を調査して正しい値を決定しなければなりません。

BCSH コマンドが発行されない場合、初期値は、ピケットフェンスバーコードで 50(12.7mm)、ラダーバーコードでは 50 (6.35mm)です。

### 注

BSYM/BDEF コマンドでバーコードの方向を再設定する場合には、BCSH コマンドも再設定されなければなりません。再設定されていない場合、ピケットフェンスバーコードがラダーバーコードに変わると、生成されるラダーバーコードはピケットフェンスバーコードの 0.5 倍の高さで印刷されます。逆の場合には、ピケットフェンスバーコードがラダーバーコードの 2.0 倍の高さで印刷されます。

### 7.3.4. バーコード・インターキャラクタギャップ設定 (Bar code InterCharacter Gap : BICG)

BICG コマンドは、Code39 の文字の間隔を設定します。

BICG コマンドの構文は次の通りです。

BICG;X;

X= ヘッド密度単位で設定した文字の間隔

BICG の最大値と最小値は、バーコードの種類によって決まります。詳細については、関連する規格を調べてください。BICG コマンドが発行されない場合、バーコードの文字の間隔は細いエレメントの幅と同じになります。

### 7.3.5. Code39 と ITF の利用例

今までに説明したコマンドでバーコードを作成する例を以下に示します。

これは、20mm×70mm のラベルに Code39 を印字する例です。

印字領域はラベルの長さ (20mm) 分取っています。(20/0.254=79ピケル)。

ラベルピッチは 23mm で UTOF は 354 となります (23/0.254×10=905)。

### 注

Code39 の開始コードと停止コードのアスタリスク記号 (\*) を入れるのを忘れてはなりません。

^^"SAMPLE";5;0;79;187

SPB;UTOF;905;

BSYM;1;1;

#CODE 39 ; 0 deg. #;

BNEW;3;BWEW;7;BICG;3;

DDF;4;10;DFM;2;2;DFS;3;DFO;1;1;

#1 FONT ; 0 deg. #;

MRK;

#LOOP#;

HBR;0;VBR;0;DWBX;0;0;394;79;NUM;

## NITTO DENKO

BCLC;1;BCID;1;

IDF;1;

HBR;33;VBR;25;BCSHY;25;

#BarCode Date#;

BCST; " \*ABCDE01234500";BSAL;2;" \*";BSTP;

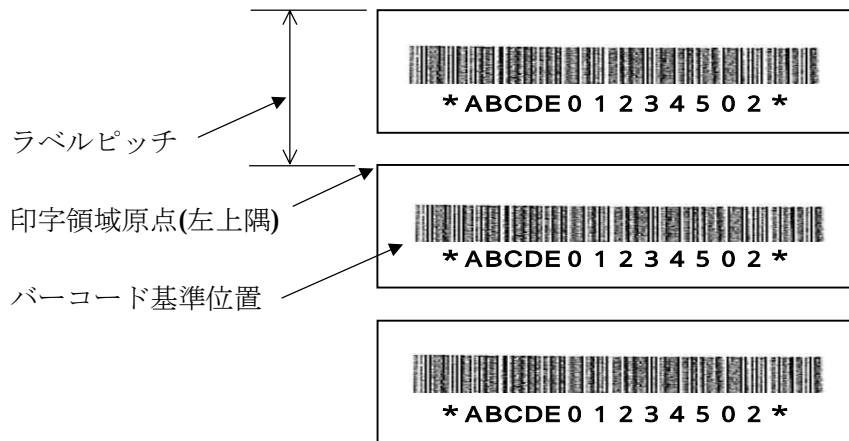
HBR;66;VBR;27;

#1Font Date#;

" \*ABCDE01234500";SAL;2;" \*";

RET;

TRM;¥



### 7.3.6. Code39 と ITF のラダーバーコード

Code39 と ITF のラダーバーコードの密度は、BNEW、BWEW、BISG コマンドの代わりに、BCIP コマンドを使って設定します。BCIP コマンドの書式と詳細については 7.5.1 をご覧下さい。

Code39 ラダーバーコードでは、バーコード・インターキャラクタギャップは細線幅と同じです。

## 7.4. UPCとEAN

UPC とそのヨーロッパ版である EAN は、同じように印刷することができます。

BSYM/BDEF を使ってバーコードの種類を指定することと、データの開始と停止を BCST と BSTP を使って指定することは、他のバーコードの設定と同じです。

これらのバーコードに必要な追加コマンドは、UPC 拡大設定率 (Upc MAGnification:UMAG) です。

### 7.4.1. 公称サイズと拡大

プリンタでは、ピケットフェンス UPC/EAN を 8 種類の大きさで、ラダーUPC/EAN を 3 種類の大きさで印刷することができます。その大きさは、公称サイズに対する拡大率で表します。ピケットフェンスの場合、拡大率の最小値は 0.76 倍、最大値は 2.08 倍で、ラダーの場合、拡大率の最小値は 0.77 倍、最大値は 1.54 倍です。

## NITTO DENKO

### 7.4.2. UPC 拡大率設定 (Upc MAGnification : UMAG)

UMAG コマンドは、プリンタで UPC/EAN のバーコードを印刷するときに使用する拡大率を指定します。拡大率が大きければ大きいほど、バーコードは大きくなります。このコマンドの構文は次の通りです。

UMAG;A;

A=別表の値に該当する整数

#### 注

UMAG で同じ値を指定しても、ピケットフェンスバーコードとラダーバーコードは印刷される大きさが異なります。

UMAG コマンドは、バーとスペースの寸法、バーコードの高さ、大きさ、可読文字の配置を自動的に決定します。

### 7.4.3. UPC と EAN の使用例

UMAG を指定したら、次に、バーコードデータを BCST コマンドと BSTP コマンドの間に、引用符で囲って指定します。スタートキャラクタ、ストップキャラクタとチェックデジットは自動的に計算されるので、指定する必要はありません。UPC と EAN は、桁数に応じたそれぞれのサブタイプごとの厳しい規則があります。UPC と EAN のサブタイプごとに、引用符の中に囲わなければならない桁数を示します。なお、UPC-E は短縮前、短縮後のどちらの桁数でも入力が可能です。

BSYMA 番号	BDEF 番号	タイプ	桁数	BSYMA 番号	BDEF 番号	タイプ	桁数
10	10,40	UPC-A	11	16	16,46	EAN 13	12
11	11,41	UPC-A2	13	17	17,47	EAN 8	7
12	12,42	UPC-A5	16	18	18,48	EAN 13 2	14
13	13,43	UPC-E	11 か 7	19	19,49	EAN 13 5	17
14	14,44	UPC-E 2	13 か 9	20	20,50	EAN 8 2	9
15	15,45	UPC-E 5	16 か 12	21	21,51	EAN 8 5	12

選択した拡大率ではバーコードの高さが高すぎる場合、BSYM/BDEF コマンドと BCST コマンドの間に、BCSH コマンドを入れることで、バーコードの高さを低くすることができます。BCSH で垂直ピクセル (0.254mm) 単位で高さを指定すると、特定の拡大率に対して、必要な高さに制限することができます。高さ制限をする UPC/EAN のバーコードの拡大率は、UMAG を使って定義しなければなりません。

次の例は、RCL を使って印刷できるいろいろな UPC/EAN バーコードを印刷した例です。



## NITTO DENKO

第1の例

~^ "UMAG1";1;0;400;0;SPB;UTOF;4100;  
HBR; 50;VBR;130;BDEF;10;UMAG;1;BCST;  
"42345678901";BSTP;  
HBR; 50;VBR;350;BDEF;13;UMAG;7;BCST;  
"04500000123";BSTP;  
TRM;¥



第2の例

~^ "UMAG2";1;0;400;0;SPB;UTOF;4100;  
HBR; 50;VBR; 80;BDEF;13;UMAG;2;BCST;  
"01129000003";BSTP;  
HBR; 50;VBR;350;BDEF;16;UMAG;2;BCST;  
"067890123455";BSTP;  
TRM;¥



## 7.5. CodaBar

CodaBar バーコードは、**BSYM/BDEF**、**BCST**、**BSTP** の3つの省略できないバーコード・コマンドと、**BCSH** コマンドおよびバーコード密度設定 (**Bar code Characters Per Inch:BCIP**) を使って定義します。

### 7.5.1. バーコード密度設定 (Bar code Characters Per Inch : BCPI)

**BCPI** コマンドは、**CodaBar** のときに、1インチ当りの文字の数を指定します。このコマンドは、細いエレメントと拾いエレメントのそれぞれの幅と、その比率を自動的に決定します。このコマンドの構文は次の通りです。

**BCPI;A;**

**A**=別表に示した **CPI** 密度の **1** つを表す整数

密度の設定値は、**19** 章モデル毎のバーコード密度一覧表をご覧ください。

### 7.5.2. CodaBar の使用例

**CodaBar** では、任意の桁数の文字をバーコードにして印刷することができます。スタートキャラクタ/ストップキャラクタは、プログラマーがデータ内に指定しなければなりません。可読文字を印刷するなら、別に指定してテキスト・コマンドを使って印刷しなければなりません。(第 **8** 章参照)。

**CodaBar** のバーコードは、チェックキャラクタの無い形式、**Modulo 11** チェックキャラクタをつけた形式、**Modulo 10** チェックキャラクタをつけた形式、**Modulo 10** 及び **Modulo 11** チェックキャラクタをつけた形式の **4** 種類の中から **1** つを選択して印刷することができます。**Modulo 11** を指定するには、データ列のチェックキャラクタが必要な位置に **@** を入れ、**Modulo 10** を指定するには、データ列に **#** を入れます。

チェックキャラクタを使うときは、特別な規則を守らなければなりません。以下にこの規則を示します。

#### **Modulo 11** チェックキャラクタ (@)

1. バーコードのデータとして使用できる文字は、**0-9** までの数字だけです。
2. ひとつのバーコードのデータは、数字 **6** 桁です。この **6** つの数字に、スタートキャラクタ、ストップキャラクタ、チェックキャラクタが加算され、バーコード **1** つについて合計で **9** 文字になります。
3. バーコード内に **6** 桁以上の数字が含まれている場合、最初の **6** 桁のデータ数字で、チェックキャラクタの計算を行いません。残りの数字はバーコードに印刷されますが、**Modulo 11** の計算には含まれません。数字 (**0-9**) ではない文字が含まれる場合、その文字はチェックキャラクタの計算で無視され、バーコードに印刷されません。
4. チェックキャラクタはソフトウェアが計算し、**@** 文字のある位置に印刷されます。
5. **Modulo 11** チェックキャラクタは次のように計算します。

ステップ **1**: スタートキャラクタから **6** 番目までの数字を取り出します。

## NITTO DENKO

ステップ 2: ステップ 1 で取り出した数字に、先頭から **7, 6, 5, 4, 3, 2** の重み係数を順に掛けます。そしてそれを加えて総和 **S** を求めます。

ステップ 3: ステップ 2 で得た数を **11** で割った余り (**R**) を求めます。

ステップ 4: **11** からステップ 3 の結果を引きます。この数が求めるチェックキャラクタです。

例

入力データ="A@237352B"

最初の 6 桁の数字="237352"

重み係数    データ

7	x	2=14
6	x	3=18
5	x	7=35
4	x	3=12
3	x	5=15
+) 2	x	2=4

**S=98**

**R=98/11** の余り =10

**C=11-10=1** ← **Modulo 11** チェックキャラクタ

**Modulo 10** チェックキャラクタ (#)

1. バーコードのデータとして使用できる文字は、**0-9** までの数字だけです。
2. 数字 (**0-9**) でない文字が含まれる場合、その文字はチェックキャラクタの計算で無視され、バーコードに印刷されません。
3. チェックキャラクタはソフトウェアが計算し、**#** 文字のある位置に印刷されます。
4. **Modulo 10** チェックキャラクタは次のように計算します。

ステップ 1: スタートキャラクタ、ストップキャラクタ、チェックキャラクタを除いた数字に、後ろから順番に番号 (**1,2,3...**) を付けます。

ステップ 2: 奇数番目の数値を加えて **2** 倍します。

ステップ 3: 偶数番目の数値を加えます。

ステップ 4: ステップ 2 とステップ 3 の結果を加えます。

ステップ 5: ステップ 4 の結果から、その値より大きい最小の **10** の倍数から、その値を引きます。

これがチェックキャラクタ **C** です。

## NITTO DENKO

例

入力データ="A123541#B"

位置	6	5	4	3	2	1
データ	1	2	3	5	4	1

---

$$\begin{array}{r} 2 + 5 + 1 = 8 \times 2 = 16 \\ 1 + 3 + 4 = 8 \end{array}$$

合計 24

$C=30-24=6$  ← Modulo 10 チェックキャラクタ  
したがって、バーコードのデータは"A1235416B"

次に **CodaBar** で印刷できるいろいろなバーコードの例を示します。

~^"CODABAR";1;0;400;0;SPB;UTOF;4100;

HBR;150;VBR; 50;BDEF;3;BCPI; 1;BCSH; 40;

BCST;"C34567890123D";BSTP;

HBR;210;VBR; 70;DHR;\$8002;DCW;30;ICS; 2;

"C34567890123D";

HBR;150;VBR;200;BCPI; 0;BCSH; 75;

BCST;B123456@C;BSTP;

HPR;380;HBR;190;VBR;220;

"B123456C";

TRM;¥



C34567890123D



B123456C

## 7.6. Code93

Code93 バーコードは、CodaBar と同じように、BSYM/BDEF、BCST、BSTP の 3 つのバーコード・コマンドに加えて、BCSH コマンドと BCPI コマンドが必要です。

### 7.6.1. Code93 用バーコード密度設定 (Bar code Character Per Inch : BCPI)

CodaBar と同じ様に、BCPI コマンドは、Code93 バーコードの 1 インチ当りの文字数を指定します。このコマンドは、Code93 バーコードの各エレメントの幅を自動的に決定します。このコマンドの構文は次の通りです。

BCPI;A;

A=別表の CPI 密度の 1 つです。

密度の設定値は、19 章モデル毎のバーコード密度一覧表をご覧ください。

### 7.6.2. Code93 の使用例

Code93 では、43 文字 (0-9、A-Z、6 つの特殊文字とスペース)、スタート/ストップ キャラクタ、4 つの制御文字からなるデータキャラクタセットを組み合わせて、ASCII の 128 のキャラクタ全部を表すことができます。

Code93 には、2 つのチェックデジットが含まれるので、読み取りエラーが最小限になります。また、必要な桁数の文字をバーコードにすることができます。スタート/ストップキャラクタは自動的に付加されますが、可読文字は別にテキスト・コマンドを使って印刷しなければなりません (第 8 章参照)。

#### 注

BCPI を 0 にすると、読み取り可能な品質のバーコードを印刷できないことがあります。

NULL(00H)、ENQ(05H)、DC2(12H)、DC4(14H)、CAN(18H)を除くすべての制御文字をバーコードや可読文字として印刷できます。

## NITTO DENKO

Code93 で印刷できるいろいろなバーコードの例を次に示します。

```
~ ^ "CODE93";1;0;400;0;SPB;UTOF;4100;  
HBR;300;VBR; 20;DHR; 1;DCH;12;DCW;24;ICS; 4;  
"CODE 93 SYMBOLOGY";  
HBR; 20;VBR; 90;BDSH;50;BCPI; 2;BCST;  
"ABCDEFGH";BSTP;  
HBR; 70;VBR;110;  
"ABCDEFGH";  
HBR; 20;VBR;180;BCST;  
"Jklmnopqr";BSTP;  
HBR; 70;VBR;200;  
"Jklmnopqr";  
HBR; 20;VBR;270;BCST;  
"!@#%*";BSTP;  
HBR; 70;VBR;290;  
"!@#%^&*";  
TRM;¥
```

### CODE 93 SYMBOLOGY



ABCDEFGH



Jklmnopqr



!@#%^&\*

## 7.7. Code128

**Code128** バーコードは、**Code93** や **CodaBar** と同じように、3つの標準的なバーコードコマンド、**BSYM**/**BDEF**、**BCST**、**BSTP**に加え、**BCSH** コマンドと **BCPI** コマンドが必要です。

### 7.7.1. Code128 用バーコード密度設定 (Bar code Character Per Inch : BCPI)

**Code93** と同じ様に、**BCPI** コマンドは、**Code128** バーコードの 1 インチ当りの文字数を指定します。このコマンドは、**Code128** バーコードの各エレメントの幅を自動的に決定します。このコマンドの構文は次の通りです。

**BCPI;A;**

**A**=別表の **CPI** 密度の 1 つです。

密度の設定値は、19 章モデル毎のバーコード密度一覧表をご覧ください。

**注**

**BCPI** を **0** にすると、読み取り可能な品質のバーコードを印刷できないことがあります。

### 7.7.2. Code128 の使用例

**Code128** はその名前が示すように、**ASCII** キャラクタ・セットの **128** 文字すべてをバーコードとして印刷できます。**Code93** も **ASCII** キャラクタ・セット全部をコード化することができますが、**Code128** は、小文字、特殊文字、制御文字に対して、より少ない記号ですみます。さらに、数字だけをコード化する場合、倍密度モード (**CODE SUBSET C**) を使用できます。

**Code128** は、**Code A**、**Code B**、**Code C**、の 3 つの異なるサブセットがあります。使用するサブセットは、それに対応する **START** コードを選択するか、コードの途中であるサブセットから別のサブセットに切り換える特別な **CODE** キャラクタを使うか、あるいはそれに続く 1 文字だけを他のサブセットに切り換える **SHIFT** キャラクタを使うかのいずれかの方法で選択できます。

次に 3 つのサブセットを簡単に説明しますが、詳細については **AIM** の **USD-6 Code128** の説明を参照してください。

- **CODE SUBSET A** は、標準 **ASCII** キーボードのアルファベット大文字と数字、制御文字 (**00H**~**1FH**) と特殊文字が使えます。
- **CODE SUBSET B** は、標準 **ASCII** キーボードの小文字を含む英数字と特別な **Code128** キャラクタが使えます。
- **CODE SUBSET C** は、文字ごとに倍密度の数を定義できる **00** から **99** までの **100** の 2 桁の数字と、**Code128** の特殊文字が使用できます。

**Code128** では、必要な桁数の文字をバーコードにすることができます。

## NITTO DENKO

START キャラクタ (CODE SUBSET 指示子) を指定しなければなりません。可読文字は、別にテキスト・コマンドを使って印刷しなければなりません。(第 8 章参照)。

### 7.7.2.1. サブセットの選択と利用

CODE SUBSET は、選択したい CODE SUBSET に対応する指示子を下記の表から選び、バーコードデータの最初の文字として指定することで選択してください。

CODE SUBSET	指示子
A	A
B	B
C	C

#### 注

間違った指示子を指定すると、CODE SUBSET を A (規定値) であるものとみなします。

各 CODE SUBSET の特殊文字をデータとして指定するための方法を次の表に示します (データの途中で CODE SUBSET を変更する場合、この表を参照してください)。

例えば、CODE SUBSET A から C に変更するには A と C のデータの間 @D を挿入します)。

CODE A	CODE B	CODE C	文字	CODE A	CODE B	CODE C	文字
@	@		@@	CODE C	CODE C		@D
FNC 3	FNC 3		@A	CODE B	FNC 4	CODE B	@E
FNC 2	FNC 2		@B	FNC 4	CODE A	CODE A	@F
SHIFT	SHIFT		@C	FNC 1	FNC 1	FNC 1	@G

制御文字あるいは小文字を送信するのが難しいホスト (例 IBM/EBCDIC) を使いやすくするために、CODE SUBSET とその該当データに対して次のように拡張されています。

- 1) CODE SUBSET A、B のどちらでも、ASCII キャラクタ・セットをすべて送信することが可能です。次の例は、コマンド列 16 進数 (CR=0DH; LF=0AH) を挿入しないで、バーコードに印刷したい復帰(CR)と改行(LF)を、等価なバーコードキャラクタ (CODE SUBSET A[CR]=CODE SUBSET B[m]; CODE SUBSET A[LF]=CODE SUBSET B[j]) に置き換えて送る方法です。

例

```
BCST;"ATOHOKUmjINC";BSTP;
```

- 2) 下に示す方法で、CODE SUBSET C のコードを CODE SUBSET A、B の中に挿入することが可能です。



## NITTO DENKO

@Y=CODE SUBSET C によるデータの開始

@Z=CODE SUBSET C によるデータの終了

### 注

@Y と@Z それ自体は、バーコードとして印刷されません。これらは CODE SUBSET C の文字で CODE SUBSET A、B の文字を送っていることを示す単なるソフトウェアのフラグにすぎません。

### 例

CODE SUBSET A の中で、@Y/@Z を使用する例を示します。

BCST;ATEST@Y7774@Z@E@Y737867@Z;BSTP;

文字	意味
A	CODE SUBSET A 選択
TEST	CODE SUBSET A で示したバーコードデータ
@Y	後ろに続くデータが、CODE SUBSET C のコードで表現されていることを示すフラグ
77	復帰の CODE SUBSET C による表現 (CODE SUBSET A データ列内)
74	改行の CODE SUBSET C による表現 (CODE SUBSET A データ列内)
@Z	CODE SUBSET C のコードによるデータ終了
@E	CODE SUBSET B 選択
@Y	CODE SUBSET C によるデータの始まり
73	i の CODE SUBSET C による表現 (CODE SUBSET B データ列内)
78	n の CODE SUBSET C による表現 (CODE SUBSET B データ列内)
67	c の CODE SUBSET C による表現 (CODE SUBSET B データ列内)
@Z	CODE SUBSET C のコードによるデータ終了

### 注

NULL (00H)、ENQ (05H)、BS (08H)、FF (0CH)、DC2 (12H)、DC (14H)、GAN (18H)、DEL (7FH) を除くすべての制御文字をバーコードとして印刷できます。

Code128 のいろいろなバーコードの例を次ページ以降に示します。

例) CODE SUBSET A を CODE SUBSET C に変更。

BCST;"ATEST@D123";BSTP;

## **NITTO DENKO**

例) **CODE SUBSET C** を **CODE SUBSET B** に変更。

**BCST;"C1234@EEST";**

**Code128** のいろいろなバーコードの例を次ページ以降に示します。

**NITTO DENKO**

~ ^ "CODE128A";1;0;400;0;SPB;UTOF;4100;  
HBR; 10;VBR; 20;DHR; 32768;DCH;12;DCW;22;ICS; 4;  
"CODE 128 SYMBOLOGY - SUBSET A WITH FUNCTIONS";  
HBR; 50;VBR; 90;BDEF;50;BCSH;50;BCPI; 2;  
BCST; "AABCDE@FGHIJKLM";BSTP;  
HBR;100;VBR;110;  
"ABCDEFGHJKLM";  
HBR; 50;VBR;170;  
BCST;"ANOPQR@GSTUVWXYZ";BSTP;  
HBR;100;VBR;190;  
"NOPQRSTUVWXYZ";  
HBR; 50;VBR;250;  
BCST;"ABCDEFGHIJK@BLM";BSTP;  
HBR;100;VBR;270;  
"ABCDEFGHJKLM";  
HBR; 50;VBR;330;  
BCST;"ANOPQR@STUVWXYZ";BSTP;  
HBR;100;VBR;350;  
"NOPQRSTUVWXYZ";  
TRM;¥

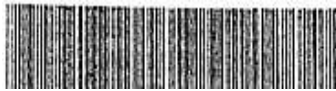
CODE 128 SYMBOLOGY - SUBSET A WITH FUNCTIONS



A B C D E F G H I J K L M



N O P Q R S T U V W X Y Z



A B C D E F G H I J K L M



N O P Q R S T U V W X Y Z

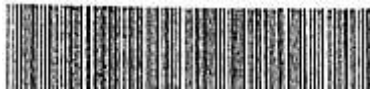
**NITTO DENKO**

~ ^ "CODE128B";1;0;400;0;SPB;UTOF;4100;  
HBR; 50;VBR; 20;DHR; 32768;DCH;12;DCW;22;ICS; 4;  
"CODE 128 SYMBOLOGY - SUBSET B WITH FUNCTIONS";  
HBR; 50;VBR; 90;BDEF;50;BCSH;50;BCPI; 2;  
BCST; "Babcde@Afg hijklm";BSTP;  
HBR;100;VBR;110;  
"abcdefghijklm";  
HBR; 50;VBR;170;  
BCST;"Bnopqr@stuvwxyz";BSTP;  
HBR;100;VBR;190;  
"nopqrstuvwxyz";  
HBR; 50;VBR;250;  
BCST;"Babcdefghijk@Blm";BSTP;  
HBR;100;VBR;270;  
"abcdefghijklm";  
HBR; 50;VBR;330;  
BCST;"Anopqr@stuvwxyz";BSTP;  
HBR;100;VBR;350;  
"nopqrstuvwxyz";  
TRM;¥

CODE 128 SYMBOLOGY - SUBSET B WITH FUNCTIONS



abcdefghijklm



nopqrstuvwxyz



abcdefghijklm



nopqrstuvwxyz

**NITTO DENKO**

~ ^ "CODE128C";1;0;400;0;SPB;UTOF;4100;  
HBR;200;VBR; 20;DHR; 32768;DCH;12;DCW;22;ICS; 4;  
"SUBSET C - NO CHANGE - C656667686970";  
HBR; 50;VBR; 90;BDEF;25;BCSH;50;BCPI; 2;  
BCST; "C656667686970";BSTP;  
HBR;150;VBR;110;  
"656667686970";  
HBR;200;VBR;150;  
"SUBSET C TO A - C656667@FABCDE";  
HBR; 50;VBR;220;  
BSTP;"C656667@FABCDE";BSTP;  
HBR;150;VBR;240;  
"656667ABCDE";  
HBR;200;VBR;280;  
"SUBSET C TO B - C504251@Eabcde";  
HBR; 50;VBR;350;  
BCST;"C504251@Eabcde";BSTP;  
HBR;150;VBR;370;  
"504251abcde";  
TRM;¥

SUBSET C - NO CHANGE - C656667686970



656667686970

SUBSET C TO A - C656667@FABCDE



656667ABCDE

SUBSET C TO B - C504251@Eabcde



504251abcde

## 7.8. Casecode Code128 (EAN-128)

**Casecode Code128** は、**Code128** を利用した物流関係のアプリケーションです。ここでは、その仕様とチェックキャラクタについて説明します。

**Casecode Code128** について詳しくは、**UNIFORM COUNCIL INC.**の **APPLICATION SPECIFICATION FOR THE UCC-128 SERIAL SHIPPING CONTAINER CODE:(WITH SYMBOL AND SHIPPING LABEL GUIDELINE)**をご覧ください。

### 7.8.1. Casecode Code128 の仕様

以下で説明するように、**Casecode Code128** は、スタートキャラクタ **C**、機能コード 1、クオリファイア、データ、**2** つのチェックキャラクタ、ストップキャラクタで構成されます。

シリアル SHIPPING・コンテナコードを含む **UCC-128** のシンボルは、**CODE SUBSET** を使用し、倍密度で印刷することでバーコードの大きさを小さくしています。さらに、スタートキャラクタ **C** の直後に機能コード 1 を付加することで、このバーコードが **UCC-128** のアプリケーションであることを示しています。

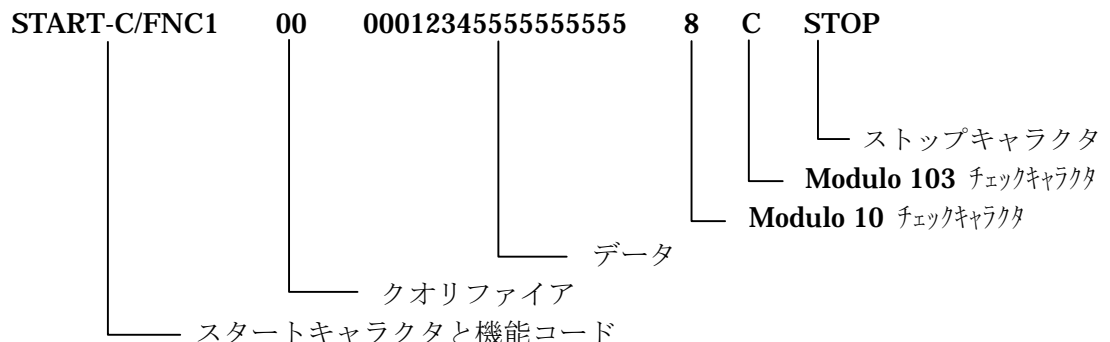
クオリファイアは **2** 桁の数で、**UCC-128** のどのアプリケーションが使われているかを示しています。シリアル SHIPPING コンテナ (標準カートン ID) のコードは **00** です。

クオリファイアに続いて **17** 桁の数値データが必要です。この数字はさらに **3** つのセクションに分けられます。最初の桁はパッキングタイプを示し、次の **7** 桁は **UCC** の業者 ID (**UCC Manufacture ID Number**)、残りの **9** 桁は SHIPPING コンテナのシリアルナンバーです。

**UCC-128** には、**Modulo 10** と **Modulo 103** の **2** つのチェックキャラクタが必要です。**Modulo 10** チェックキャラクタはデータにクオリファイアを加えた **19** 桁の数字から計算します。そして、この **19** 桁に **Modulo 10** チェックキャラクタを加えた **20** 桁が、**10** 桁の倍密度コードで印刷されます。**Modulo 103** チェックキャラクタは **Code128** バーコードの仕様に従って計算されます。

**Modulo 103** チェックキャラクタの後にはストップキャラクタが必要です。

**UCC-128** のデータの例



## NITTO DENKO

### Modulo 10 チェックキャラクタ

UCC-128 の Modulo 10 チェックキャラクタは CodaBar の Modulo 10 チェックキャラクタとは異なり、重み係数として **3** を掛けます。

UCC-128 の Modulo 10 チェックキャラクタは次のように計算します。

- ステップ 1: クオリアライアにデータを加えた **19** 桁に、最後の桁から順番に番号を付けます。
- ステップ 2: 奇数番目の数字を加えて、その結果に **3** を掛けます。
- ステップ 3: 偶数番目の数字を加えます。
- ステップ 4: ステップ 2 とステップ 3 の結果を加えます。
- ステップ 5: ステップ 4 で求めた値より大きい最小の **10** の倍数から、ステップ 4 で求めた値を引きます。これがチェックキャラクタ **C** です。

例

入力データ = "0000012345555555555"

位置	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
データ	0	0	0	0	0	1	2	3	4	5	5	5	5	5	5	5	5	5	5

---

0	+	0	+	0	+	2	+	4	+	5	+	5	+	5	+	5	+	5	+	5	=	31	x	3	=	93
		0	+	0	+	1	+	3	+	5	+	5	+	5	+	5	+	5	+	5	=					29

---

合計 122

C=130-122=8 ← Modulo 10 チェックキャラクタ

### 7.8.2. Casecode Code128 の使用例

Casecode Code128 のバーコードの印刷方法は、入力データの桁数の設定が異なるだけで、Code128 のバーコードと同じです。つまり、BSYM/BDEF、BCST、BSTP、BCPI、BCSH の 5 つのコマンドが必要です。なお、印刷できるバーコード密度と BCPI のパラメータとの関係については、7.7.1.を参照してください。

Casecode Code128 が選択されると、スタートキャラクタ、機能コード 1、Modulo 10 と Modulo 103 のチェックキャラクタ、ストップキャラクタが入力データに自動的に付加されます。UCC-128 のシンボルからこれらのデータを除いた残りの 19 桁を入力してください。19 桁より少ないデータを入力した場合、後ろに足りない数だけ "0" を加えます。また、データが 19 桁より多い場合、先頭から 19 桁を採用し、それより多い部分は無視されます。なお、データの中に数字以外の文字（文字や記号）がある場合、コマンドエラーを表示します。

## NITTO DENKO

可読文字を印刷する場合は、バーコードとは別に指示しなければなりません。

RCLには **Casecode Code128** の可読文字を印刷するために、文字列に **Modulo 10** チェックキャラクタを加えて印刷する機能があります。

詳しくは、21章のフォント選択表をご覧ください。

```
~^"CASECODE";1;0;400;0;SPB;UTOF;4100;  
HBR;150;VBR;200;BDEF; 27;BCSH;50;BCPI; 3;  
BCST;"0000052177177444130";BSTP;  
HBR;150;VBR;230;DHR;$12;DCH; 15;DCW; 30;ICS; 10;  
"0000052177177444130";  
TRM;¥
```



**00000521771774441303**

```
~^"CASECODE";1;0;400;0;SPB;UTOF;4100;  
HBR;300;VBR; 20;BDEF; 28;BCSH;100;BCPI; 3;  
BCST;"00000123455555555555";BSTP;  
HBR;270;VBR; 20;DHR;$612;DCH; 15;DCW; 30;ICS; 5;  
"00000123455555555555";  
TRM;¥
```





## 8. 検証機能

ここでは検証機能搭載プリンタの検証関連機能に関するコマンドについて説明します。

検証機を搭載したプリンタは印字したバーコードを検証して、欠陥が見つかった場合、そのバーコード上にボイドマークを印刷し、ラベルを再印字する機能を持っています。また、バーコードの検査を行なうことによって印刷エネルギーを最適化する **APPC (Automatic Print Process Control)** 機能を持っています。

なおこれらのコマンドは、検証機付きのプリンタのみ有効です、検証機なしのプリンタでは、単に無視されます。

検証についての詳細は取扱説明書をご覧ください。

### 8.1. 検証機制御コマンド

#### 8.1.1. 検証モード設定 (VeriFY Mode : VFYM)

**VFYM** コマンドは、バーコードを良品と判断する回数と検証のギャップを設定することができます。

このコマンドの構文は以下の通りです。

**VFYM;XX;YY;**

**XX** 検証モード

- 0** : 2回検証の内、1回 **OK** で検査 **OK** とする。
- 1** : 2回検証の内、2回 **OK** で検査 **OK** とする。
- 2** : 3回検証の内、3回 **OK** で検査 **OK** とする。

**YY** 検証ギャップ

- 2** 回目、及び **3** 回目の検証開始位置を指定します。
- 単位は **LF** ピッチに依存しますので、機種別の仕様を参照ください。

\* 1回目のスキャン位置は、**0.8~1.4mm** の間です。

#### 注

検証モードを **1**、**2** に変えると、ボイド発生率が高くなります。  
通常は、検証モード **0** で使用してください。

**VFYM** を指定しないときは、検証モード **0**、検証ギャップ **0.5mm** になります。

#### 8.1.2. 検証機オフ (APpc OFF : APOF)

**APOF** コマンドは、検証機を停止させ、個々のバーコードの検査を行わないようにします。一度、**APOF** コマンドが発行されると、**APON** コマンドを受取るまで、印刷したバーコードを検査しません。つまり、ラベル内の特定のバーコードの検査を行うことができます。

## NITTO DENKO

検証機能プリンタは、バーコードの検査を行なうことによって印刷エネルギーを最適化する機能を持っています。すべてのバーコードに **APOF** コマンドを使用した場合、印刷エネルギーは印刷ヘッド温度だけで制御されます。

### 8.1.3. 検証機オン (APpc ON : APON)

**APON** コマンドは、検証機を動作させます。一度 **APON** コマンドが発行されると、**APOF** コマンドが発行されるまですべてのバーコード検査を行います。プリンタでは、(電源オン時の) 初期値は **APPC** オンです。

**APOF** と **APON** コマンドの構文は次の通りです。

```
HBR; 10;VBR;100;BCST;"*123456*";BSTP;#VERIFICATION
ENABLED#;HBR; 10;VBR;200; APOF;BCST;"*123456*";BSTP;
APON;#DISABLE VERIFICATION#;
```

### 8.1.4. 検証機能無効 (Disable VeriFication : DVFY)

**DVFY** コマンドは、ラベル上のすべてのバーコードに対する検査を行わないようにします。**DVFY** を使って検証機能を無効にすることと、**APOF** を使って検査をしないことの違いは、**APOF** では実際に検証機構の動作を停止させ、**DVFY** ではスキャンしてバーコードを判断しますが、欠陥が見つかってボイドマークを印刷したり再印刷しません。ただし、スキャンした結果により、印刷エネルギーを制御します。

#### 注

検証機能を無効にするには、プリンタの機能設定でも設定できます。通常は、**APON**、**APOF**、**DVFY** コマンドを使用せずに、プリンタの機能設定で設定してください。(コマンドよりもプリンタの設定が優先されます)。

### 8.1.5. AQL レベル設定 (Assured Quality Level set : AQL)

**AQL** コマンドは、バーコードの検査の際にバーコードの要素の許容誤差を設定します。許容誤差は、プリンタのソフトウェアで設定した値に、このコマンドで指定した **0.0125mm** から **0.0875mm** までの範囲の数を加えた値で設定します。

コマンドの構文は次の通りです。

**AQL;X;**

**X=0~7** までの整数。

設定される値は、最小単位 **0.0125mm** の **X**倍になります。

プリンタのソフトウェアが決定する許容値は、それぞれのバーコードの規格に定義された値と同じですので、ほとんどの場合、**AQL;0;**で十分です。しかし、リボンやラベル品質のために、インクの転写が不十分の場合や、高密度のバーコードを印刷中にボイドマークが度々印刷される場合、許容幅を大きくすれば、エラー率を減少させることができます。この場合でも、一般的なバーコード・リーダーで十分使用できるラベルを作成できます。

## NITTO DENKO

コマンド列の最後の **AQL** 設定が、すべてのラベルに適用されます。つまり、コマンド列に複数の **AQL** コマンドを入れた場合、一番最後の **AQL** コマンドだけが実行されます。

**AQL** コマンドのパラメータとして **8** 以上の値を指定した場合、プリンタの正常な動作を保証できません。**8** 以上の値は使用しないで下さい。

### 8.1.6. 再印字禁止機能 (NO RePrint : NORP)

**NORP** コマンドは、エラーのために印刷を中断したラベルをエラー復帰後に再印刷させないための機能です。

プリンタはこのコマンドを受信すると、印字中にエラー（サプライエンプティ等）が発生すると復帰後の印刷再開はエラー発生時に印刷していたラベルの次のラベルより行うことになります。

なお、検証 **NG** については通常どおりボイドを印字して再印字を行います。

このコマンドの構文と用法は次の通りです（**SPB**、**UTOF** コマンドのあとに **NORP** コマンドを挿入）。

```
~^"SAMPLE1";1;0;24;187;  
  
SPB;  
  
UTOF;0354;  
  
NORP;  
  
MRK;HBR;0;VBR;0;DWBX;0;0;398;24;NUM;BCLC;1;  
  
~  
  
RET;  
  
TRM;¥
```

#### 注

- **NORP** コマンドを使用すると、検証を行わないラベルが出たり、ラベルの欠番、印字枚数の不足が起こる場合があります。
- **NORP** コマンドを使用しない場合は、検証を行ったデータを印字完了と判断します。従って印字後、検証までにエラー（サプライエンプティ塔）が発生した場合、復帰後の印刷再開時、エラー発生時に印字していた同内容のデータより行うことになります。

## 8.2. ハーフドット

ハーフドット制御を使ってバーコードを印字することができます。ハーフドット制御とは、サーマルヘッドの発熱体に通常よりも大きなエネルギーを加えることにより発熱体の周囲のインクも溶かして太いバーを印字したり、逆に通常よりも小さなエネルギーを加えることで細いバーを印字したりすることです。感熱紙ではインクリボンが無いので機能しません。

## NITTO DENKO

UPC/EAN では、ハーフドット制御の印字が必要かどうかはソフトウェアが自動的に決定し、ハーフドット制御にしたりしなかったりします。プログラマーがハーフドット制御を指定できるバーコードは、ピケットフェンス **Code39** とピケットフェンス **ITF** の 2 つです。

通常は、できるだけこの機能を使用しないでバーコードを印刷するほうが良いでしょう。通常とは異なった、ある特定の密度で印刷する必要がある時には、ハーフドット制御を試してください。

### 8.2.1. ハーフドット・オン (HALF dot on : HALF)

普通のラベル用紙に熱転写インクリボンを使って印字する時に、バーを太くして印刷するために使います。このコマンドの構文は次の通りです。

HALF;

**HALF** コマンドはハーフドット制御で印刷しようとするバーコードの先頭より前に指定しなければならず、後で説明します **HOFF (Half dot OFF:HOFF)** コマンドを使ってハーフドット制御が解除されるまでオンのままになります。**HALF** の効果は 3 つあります。

- (1) 各バーの両側に約 1/4 ドットずつインクの幅を増加させ、合計で 1/2 ドットを各バーに付加します。例えば、2 ドット幅のバーコードは 2.5 ドット幅のバーに広がります。
- (2) さらに白のハーフドットをバーコードの全スペースに加えます。例えば、2 ドット幅のスペースは 2.5 ドット幅のスペースに広がります。
- (3) バーコード誤差の許容値の表から新しい値を割当て、ハーフドット・バーコードが、プリンタに内蔵されている検証機で正しく走査できるようにします。

### 8.2.2. ハーフドット・オン・バー (HALF dot on Bar : HLFB)

インクの広がり難いラベル用紙に熱転写インクリボンを使って印字する時に、バーを細くして印刷するために使います。このコマンドの構文は次の通りです。

HLFB;

**HFLB** コマンドもハーフドット制御で印刷しようとするバーコードの先頭より前に指定しなければならず、後で説明します **HOFF (Half dot OFF:HOFF)** コマンドを使ってハーフドット制御が解除されるまでオンのままになります。**HALF** の効果は 3 つあり **HALF** コマンドと同じです。

### 8.2.3. ハーフドット・オフ (Half dot OFF : HOFF)

**HOFF** コマンドは、ハーフドット機能をオフにします。構文は次の通りです。

HOFF;

**HOFF** は、ラベルの中でハーフドット機能が必要な最後のバーコードの後に指定しなければなりません。**HALF** が実行されないと、ハーフドット機能は働きません。(既定値は **HOFF**)。

## NITTO DENKO

同じラベル上に **HALF** と **HOFF** のバーコードを混在させることはできません。**HALF** を必要とするバーコードが 1 つでもある場合、バーコード全部にハーフドット処理を実行しなければなりません。特に重要なのは、**UPC/EAN** バーコードは、特定の拡大率でハーフドット処理を自動的に行い、他の場合は行いません。

### 注

ハーフドット機能は、検証機を内蔵したプリンタ以外では使用しないで下さい。

### 8.3. 検証対象バーコードについて

プリンタで検証可能なバーコードは 1 ラベル内で 15 個までです。ただしこの数には、**APOF** コマンドを使って検証機を無効にしたバーコードは含まれません。

検証機でスキャンするには、バーコードの高さ (**BCSH**) には制限があります。機種別の仕様を参照ください。

プリンタは、バーコードをスキャンすることはできますが、可読文字とバーコードが実際に一致するかどうか判断できません。

検証機付きのプリンタでも、ラダーバーコードは検証できません。

## 9. 2次元シンボル

RCL は普及している 2 次元シンボルのうち **QRCode, Code49** をサポートしています。**Code49** は  $0^\circ$ 、 $90^\circ$ 、 $180^\circ$ 、 $270^\circ$ 、の 4 方向に印刷することができます。 $0^\circ$  と  $180^\circ$  方向のバーコードは、各エレメントがラベル搬送方向に対して平行に並ぶためピケットフェンスバーコードと呼びます。これに対して  $90^\circ$ 、 $270^\circ$  の方向のバーコードは各エレメントがラベル搬送方向に対して直角に並ぶためラダーバーコードと呼びます。RCL では、2 次元シンボルを作成するために役に立つ多くのコマンドが在ります。全ての 2 次元シンボルは、少なくとも 3 つのコマンドを必要とします。

それは、2 次元シンボル選択 (**Barcode SYMBOL:BSYM**)、  
シンボルデータ開始 (**Barcode Start:BCST**)、  
バーコードデータ終了 (**Barcode StoP:BSTP**) で説明します。

さらに、各 2 次元シンボルはそれ自体のコマンドを必要とします。

**QR Code** には、セル幅設定、エラー修復レベル設定、マスクパターン設定の 3 つのコマンドが使われます。

**Code49** には、バーコード密度設定、バーコード高さ設定が必要です。

### 9.1. 2次元シンボルの選択

2 次元シンボルを作成するには、適切な種類の 2 次元シンボルを選択しなければなりません。2 次元シンボルの選択コマンドは、使用する 2 次元シンボルの種類を選択するのに使います。

#### 9.1.1. 2次元シンボルの種類

RCL で使用できる 2 次元シンボルは、**QR CODE**、**Code49**、の 2 種類です。

#### 注

プログラマーは、使用する 2 次元シンボルの規格を調査し、またそこに定められた指針に従う責任があります。

#### 9.1.2. 2次元シンボルの選択 (**Barcode SYMBOL : BSYM**) (**Barcode DEFine : BDEF**)

**BSYM** コマンドは、2 次元シンボルの種類と印刷方向を設定するために使います。

2 次元シンボルの種類は、2 次元シンボルを生成する前に選択しなければなりません。

コマンドの構文は次の通りです。

**BSYM;A;B;**

**A**=2 次元シンボルの種類を決める整数、次表によって決まる。

**B**=2 次元シンボルの方向を設定する 1~4 の整数

(**QR CODE** では、**B** の値は意味を持ちません)

## NITTO DENKO

2次元シンボルの種類と、**BSYM** コマンドのパラメータの値を以下に示します。

2次元シンボルの種類	A
QR Code モデル 1	100
QR Code モデル 2	101
Micro QR Code	102
Code49	60

シンボルの方向	B
0°	1
90°	2
180°	3
270°	4

**BDEF** コマンドは、**BSYM** コマンドと同様なコマンドです。但し、**BDEF** コマンドでは 0° と 270° の 2 方向の印刷だけが可能です。(QR CODE は 0° のみです) 2次元シンボルの種類は、2次元シンボルを生成する前に選択しなければなりません。

コマンドの構文は次の通りです。

**BDEF;A;**

A=60、61 と 100、101、102 の整数、下に示された表で決まります。

2次元シンボルの種類と、**BDEF** コマンドのパラメータの値を示します。

2次元シンボルの種類	A
Code49	60
Code49 ラダー	61
QR Code モデル 1	100
QR Code モデル 2	101
Micro QR	102

一度 **BDEF** や **BSYM** コマンドで 2次元シンボルを選択すると、新たに別のシンボルが選択されるまで同じ 2次元シンボルを使用します。

### 9.2. 2次元シンボルの印字

2次元シンボルは、バーコード同様 **HBR,VBR,HPR,VPR,HOME,EOL** コマンドを使って印刷位置を設定します。**Code49** はシンボルの左下端がカーソルの基準位置です。つまり、回転方向 0° のピケットフェンスシンボルの場合はカーソルから上方右側に向かって印刷され、回転方向 270° のラダーシンボルは、カーソルの下方右側に向かって印刷されます。シンボルの印刷が完了すると次に印刷されるオブジェクトの印字開始基準位置は、ピケットフェンスシンボル (0°) ではシンボルの右下端になり、ラダーシンボルでは、左下端になります。印刷の原点位置は、印字領域の左上端です。

**QR Code** は左上端が基準位置になります。つまり、シンボルはカーソルから下方右側に向かって印刷されます。

## NITTO DENKO

### 9.2.1. シンボルデータの入力

シンボルデータは、常に引用符 (“ ”) で囲わなければなりません。

**Code49** では、**ASCII** キャラクタセットの **128** 文字全てをデータとして使用できます。

この他に、**FUNC1,FUNC2,FUNC3** の特殊コードが使用できます。

これらの入力方法は、別の部分で説明します。

又、**Code49** では、シンボルの各段に **Modulo49** のチェックキャラクタ及び最終段にシンボルのチェックキャラクタが **4** から **6** キャラクタ付加されます。

**QR Code** は、英数字、バイナリー、漢字データを入力することができます。これらのデータの入力方法は、別の部分で説明します。

**Code49**、**QR Code** で、可読文字は、テキストコマンドを使って別に印刷しなければなりません。この場合シンボルと可読文字は同じデータになるようにして下さい。

### 9.2.2. シンボルデータ開始 (**BarCode StarT : BCST**)

**BCST** コマンドは、プリンタに対して、このコマンドの直後に続くデータを、**2** 次元シンボルとして印刷するように指示します。**BCST** コマンドがないと、そのデータは通常のテキストであるとみなし、それをそのまま印刷します。

このコマンドの構文は次の通りです。

```
BCST;
```

### 9.2.3. シンボルデータ終了 (**Barcode StoP : BSTP**)

**BSTP** コマンドは、**BCST** と対になっていて、プリンタに **2** 次元シンボルのデータの終了を知らせます。このコマンドは、最後のシンボルデータの直後に置かなければなりません。**BSTP** コマンドが発行されない場合プリンタはどこでシンボルが終わっていいか判りません。

このコマンドの構文は次の通りです。

```
BSTP;
```

次の例は、**Code49** シンボルを選択し、データを入力したものです。

```
BSYM;60;1;
```

```
BCST;"CODE49 BARCODE";BSTP;
```

## 9.3. Code49

**Code49** は、**DTDS** または、**BSYM/BDEF** を使ってシンボルの種類を定義し、**BCST,BSTP** を使ってシンボルデータの開始と停止を示し、シンボルを印刷するために、シンボルの密度 (**BCPI**)、シンボルの高さ (**BCSH**) を設定する必要があります。

### 9.3.1. Code49 用シンボル密度設定 (**Barcode Character Par Inch : BCPI**)

**BCPI** コマンドは、**Code49** の **1** インチ当りの文字数を指定します。

このコマンドは、**Code49** のシンボルの各エレメントの幅を自動的に決定します。



## NITTO DENKO

このコマンドの構文は次の通りです。

**BCPI;A;**

A=次の表の CPI 密度の 1 つです。

密度の設定値は、19 章モデル毎のバーコード密度一覧表をご覧ください。

### 9.3.2. Code49 用シンボル高さ設定 (BarCode Symbol Height : BCSH)

BCSH コマンドは、Code49 の 1 段のシンボル高さを設定します。

シンボルのセパレータバーの高さは、BCPI コマンドで設定されたモジュール寸法で展開されます。つまりシンボルとしての全体の高さ H は、次式により決まります。

$$H=(h+1)*R+1)*X$$

H=シンボル高さ

h はそれぞれのバーの高さ、モジュールの倍数 (整数倍)

R は行 (段) の数

注: セパレータバーは X モジュールとする。

このコマンドの構文は次の通りです。

- ・ピケットフェンスシンボルの場合

**BCSH;YY;**

YY=0.254 の単位で設定したシンボル高さ

- ・ラダーシンボルの場合

**BCSH;XX;**

XX=0.127 の単位で設定したシンボル高さ

Code49 では、最小のシンボル高さは、エレメントのモジュール寸法を X とすると 8X 以上が推奨高さです。

初期値は 50 です。

### 9.3.3. Code49 の使用例

Code49 は、ASCII キャラクタセットの 128 文字全てと機能コードとして FUNC1、FUNC2、FUNC3 のコードをバーコードとして印刷できます。

Code49 では、数字が 5 桁以上連続している場合、数字モードとしてデータを短縮処理してコード化を行います。

Code49 は、2 段から 8 段の多段式のバーコードで、各々段は、セパレータバーにより分割されます。各段には、段毎のチェックキャラクタが付加されます。

また、シンボルの最終段には 4 キャラクタから 6 キャラクタのシンボル全体のチェックキャラクタが付加されます。

注

NULL(00H)、ENQ(05H)、DC2(12H)、DC4(14H)、CAN(18H)を除く全ての制御文字をバーコードとして印刷できます。

機能コードの入力方法

機能コードの FUNC1、FUNC2、FUNC3 の 3 つの特殊コードをデータとして指定する

## NITTO DENKO

方法を次の表に示します。

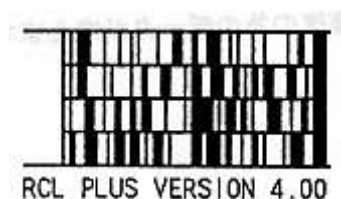
データコード	入力データ文字
@	@@
FUNC1	@A
FUNC2	@B
FUNC3	@C

入力データが上記で表した以外の@キャラクタとのペアの場合は、@とそれに続くキャラクタをそれぞれ1つのデータとして処理します。

つまり@@Dと@Dと入力された場合は、どちらもシンボルデータとしては、@とDの2つのデータとして展開されます。

次に、Code49で印刷できるバーコードの例を示します。

```
~^"CODE49";1;0;120;0;  
SPB;HBR;0;VBR; 90;  
BSYM;60;1;BCPI;5;BCSH;16;  
BCST;"RCL Ver. 4.00";BSTP;  
HBR;0;VBR;95;DDF;3;1;DFM;2;2;  
"RCL Ver. 4.00";  
TRM;¥
```



### 9.4. QR Code

QR Code シンボルは、シンボルの選択コマンド **BSYM/BDEF** とシンボルの開始/終了を示す **BCST,BSTP** コマンドに加えて、**QREL** または **TDEL,QR** マスクパターン設定コマンド **QRMP** が必要です。

また、QR Code は、他のバーコードと違いイメージの展開開始位置が、左下ではなく左上になっていることに注意してください。

#### 9.4.1. QR Code シンボル用セルサイズ設定 (QR code Cell Size : QRCS)

QRCS コマンドは、QR Code の1つのセル (正方形) の大きさを設定する。  
このコマンドの構文は以下の通りです。

QRCS;X;

X=セルのドット数を設定する整数  
デフォルト値は、2ドットです。

#### 9.4.2. QR Code エラー修復レベル設定 (QR code Error Level : QREL)

QR Code には、エラー修復レベルの設定があります。QREL コマンドまたは TDEL コマンドを使用します。

## NITTO DENKO

QREL コマンドは QR Code のモデル 1 のみ対応しています。TDEL コマンドは QR Code のモデル 1 とモデル 2 に対応しています

QREL コマンドはエラー修復レベルの設定を行います。  
エラーの修復率により以下の 4 つのレベルがあります  
なお、エラー修復レベルが高くなればそれだけ修復のためのデータが増えます。  
このコマンドの構文は次の通りです。

QREL;*L*;

*L*=次の表のエラーレベルの 1 つ

<i>L</i>	誤り修復レベル		
1	高密度レベル	(修復レベル L)	約 7%復元
2	標準レベル	(修復レベル M)	約 15%復元
3	高信頼度レベル	(修復レベル Q)	約 25%復元
4	超高信頼度レベル	(修復レベル H)	約 30%復元

設定値が上記以外の値の場合には、3 がデフォルト値となります。

### 9.4.3. 2次元シンボルエラー修復レベル設定 (Two Dimensional Error Level : TDEL)

TDEL コマンドはエラー修復レベルの設定を行います。  
エラーの修復率により以下の 4 つのレベルがあります  
なお、エラー修復レベルが高くなればそれだけ修復のためのデータが増えます。  
このコマンドの構文は次の通りです。

TDEL;*L*;

*L*=次の表のエラーレベルの 1 つ

<i>L</i>	誤り修復レベル		
1	高密度レベル	(修復レベル L)	約 7%復元
2	標準レベル	(修復レベル M)	約 15%復元
3	高信頼度レベル	(修復レベル Q)	約 25%復元
4	超高信頼度レベル	(修復レベル H)	約 30%復元

設定値が上記以外の値の場合には、4 がデフォルト値となります。

### 9.4.4. QR Code マスクパターン設定コマンド (QR code Mask Pattern : QRMP)

QRMP コマンドは、シンボルのマスクパターンの選択を行います。QR Code では、読み取りを確実にを行うために、シンボル内に白セル、黒セルが、バランス良く配置されるように、また位置検出パターンに特徴的な“1011101”なる系列がシンボル内になるべく存在しないようにマスキングを行う。このマスキングのパターンには 0 から 7 までの 8 種類のマスクパターンが在ります。

このコマンドの構文は次の通りです。

QRMP;*X*;

*X*=以下の指定による。(0 から 9 の整数)

## NITTO DENKO

- X: マスクパターン
- 0~7: マスク 0~7 を選択
- 8 : マスク無し
- 9 : 自動選定

設定値が上記以外の値の場合には、**9** がデフォルト値となります。

自動選定を選んだ場合は、プリンタが **8** 種類のマスクパターンで連続同色パターン、同色ブロック、**1011101** 系列、全体の黒セルの比率に対してペナルティを科して減点法で評価を行い最高点を上げたパターンを採用します。

マスク **No.0** から **7** で指定された場合は、指定のパターンでマスク処理を行います。

**8** では、マスク処理を行いません。

### 注

特別な理由が無い限りマスクパターンは、自動選定を選択してください。

#### 9.4.5. QR Code の印字データの入力方法

QR Code で、データとして用いることのできる入力文字の種類には

- ・ 数字モード (N) : 10 進数のみ (0~9)
- ・ 英数字モード (A) : 10 コの数字、26 個の大英文字、9 個の記号の計 45 文字
- ・ バイナリーモード (B) : JIS 8 単位符号
- ・ 漢字モード (K) : シフト JIS コードで漢字を含むデータの 4 種類のデータを用いることができる。  
又、データとしてはこれらのデータを混在して使用することも可能です。

印字データの入力時にはデータの先頭に、入力モードの識別子を付加して入力してください。

プリンタはこの識別子によりデータの種別を判別してエンコード処理を行います。

各モードを混在して入力する場合は“,”によりモードを区切って入力してください。

例 : BCST;"N1234567,AABCD01,B0002ab";BSTP;

▼ バイナリのデータ数を 4 桁で記述します。

この場合は、**1234567** が数字モードのデータ、**ABCD01** が英数字モードのデータ、**ab** がバイナリーモードのデータとなります。

QR Code では、数字、英字、バイナリ、漢字のデータを混在して入力できます。

#### 9.4.6. QR Code の使用例

QR Code では、英数字、漢字、バイナリーのデータを印字することができます。

QR Code では、構成セルに応じて、**1~22** の **22** 種類のバージョンがあります。

印刷するバージョンは、プリンタが印字するデータ数、誤り訂正レベルにより格納可能なバージョンを選択して印字を行います。

## NITTO DENKO

次に QR Code の印刷例を示します。

```
~ ^ "QRCODE";1;0;150;200;  
SPB;UTOF;1500;HBR;0;VBR;0;  
DDF;8;10;DFM;2;2;  
HBR;5;VBR;10;"LEVEL=H:MASK=AUTO";  
HBR;5; VBR;25;"DATA=K 漢字,N1234,B0004test";  
BDEF;100;QREL;3;QRCS;8;QRMP;9;  
HBR;80;VBR;50;  
BCST;"漢字,N1234,B0004test";BSTP;  
TRM;
```

```
LEVEL=H:MASK=AUTO  
DATA=K 漢字,N1234,B0004test
```



## NITTO DENKO

### 9.5. Reduced Space Symbol (RSS)シンボル

RCL では、Reduced Space Symbol (RSS) 及びRSS + 2D Composite Componets (2D CC) をサポートしています。(対応機種は注を参照してください) これらのシンボルは  $0^{\circ}$ 、 $90^{\circ}$ 、 $180^{\circ}$ 、 $270^{\circ}$  の方向に印刷することが出来ます。 $0^{\circ}$  と  $180^{\circ}$  方向のバーコードは、各エレメントがラベル搬送方向に対して平行に並ぶためピケットフェンスバーコードと呼ばれます。これに対して  $90^{\circ}$  と  $270^{\circ}$  の方向のバーコードは各エレメントがラベル搬送方向に対して直角に並ぶためラダーバーコードと呼ばれます。

RCL では、RSS シンボルを作成するために役に立つ多くのコマンドがあります。全ての RSS シンボルは、少なくとも3つのコマンドを必要とします。

それは、シンボル選択 (BSYM:Bar code SYMBol 及びBDEF: Bar code DEFine)、バーコードデータ開始 (BCST:BarCode StarT)、バーコードデータ終了 (BSTP:Barcode StoP) で説明します。

さらに、各RSS シンボルはそれ自体のコマンドを必要とします。

RSS シンボルには、バーコード密度、バーコード高さの設定、及びRSS Expanded Stacked では、1 段当たりのセグメント数の設定コマンドが使われます。

RSS シンボルを作成するには、適切な種類のRSS シンボルを選択しなければなりません。

RSS シンボル選択コマンドは、使用するRSS シンボルの種類を選択するのに使います。

RCL で使用できるRSS シンボルは、RSS-14、RSS-14 Trancated、FSS-14 Stacked、RSS-14 StackedOmnidirectional、RSS Limited、RSS Expanded、RSS Expanded Stacked の7種類です。

注：

- ・ プログラマーは、使用するRSS シンボルの規格を調査し、またそこに定められた指針に従う責任があります。
- ・ 対応機種は DURA PRINTER SR(RSS)、DURA PRINTER SR(RSS)で特別対応のプリンタファームウェア(ROM)交換が必要です。

## NITTO DENKO

### 9.5.1 RSS シンボルの選択 (Barcode SYMBOL:BSYM) (Barcode DEFine:BDEF)

**BSYM** コマンドは、**RSS** シンボルの種類と印刷方向を設定するために使います。**RSS** シンボルの種類は、**RSS** シンボルを生成する前に選択しなければなりません。

コマンドの構文は次の通りです。

**BSYM;A;B;**

**A**=**RSS** シンボルの種類を決める整数、下表によって決まります。

**B**=**RSS** シンボルの方向を設定する**1** から**4** の整数

**RSS** シンボルの種類と、**BSYM** コマンドのパラメータの値を以下に示します。

RSS 元シンボルの種類	<i>A</i>
<b>RSS-14</b>	<b>64</b>
<b>RSS-14 Truncated</b>	<b>65</b>
<b>RSS-14 Stacked</b>	<b>66</b>
<b>RSS-14 Stacked Omnidirectional</b>	<b>67</b>
<b>RSS Limited</b>	<b>68</b>
<b>RSS Expanded</b>	<b>69</b>
<b>RSS Expanded Stacked</b>	<b>70</b>

シンボルの方向	<i>B</i>
<b>0°</b>	<b>1</b>
<b>90°</b>	<b>2</b>
<b>180°</b>	<b>3</b>
<b>270°</b>	<b>4</b>

## NITTO DENKO

**BDEF** コマンドは、**BSYM** コマンドと同様なコマンドです。但し、**BDEF** コマンドでは **0°** と **270°** の2 方向の印刷だけが可能です。**RSS** シンボルの種類は、**RSS** シンボルを生成する前に選択しなければなりません。

コマンドの構文は次の通りです。

**BDEF;A;**

A=シンボルを決める整数、下に示された表で決まります。

**RSS** シンボルの種類と、**BDEF** コマンドのパラメータの値を示します。

<b>RSS</b> シンボルの種類	<b>A</b>
<b>RSS-14</b>	<b>64</b>
<b>RSS-14 Truncated</b>	<b>65</b>
<b>RSS-14 Stacked</b>	<b>66</b>
<b>RSS-14 Stacked Omnidirectional</b>	<b>67</b>
<b>RSS Limited</b>	<b>68</b>
<b>RSS Expanded</b>	<b>69</b>
<b>RSS Expanded Stacked</b>	<b>70</b>
<b>RSS-14 ラダー</b>	<b>71</b>
<b>RSS-14 Truncated ラダー</b>	<b>72</b>
<b>RSS-14 Stacked ラダー</b>	<b>73</b>
<b>RSS-14 Stacked Omnidirectional ラダー</b>	<b>74</b>
<b>RSS Limited ラダー</b>	<b>75</b>
<b>RSS Expanded ラダー</b>	<b>76</b>
<b>RSS Expanded Stacked ラダー</b>	<b>77</b>

一度**BSYM** や**BDEF** コマンドで**RSS** シンボルを選択すると、新たに別のシンボルが選択されるまで同じ**RSS** シンボルを使用します。

### 9.5.2 **RSS** シンボルの印刷

**RSS** シンボルは、他のバーコード同様**HBR**、**VBR**、**HPR**、**VPR**、**HOME**、**EOL** コマンドを使って印刷位置を設定します。

展開原点はシンボルの左下端がカーソルの基準位置です。つまり、回転方向**0°** のピケットフェンスシンボルの場合はカーソルから上方右側に向かって印刷され、回転方向**270°** のラダーシンボルは、カーソルの下方右側に向かって印刷されます。

シンボルの印刷が完了すると次に印刷されるオブジェクトの印刷開始基準位置は、ピケットフェンスシンボル (**0°**) ではシンボルの右下端になり、ラダーシンボルでは、左下端になります。印刷の原点位置は、印刷領域の左上端です。



## NITTO DENKO

### 9.5.3 シンボルデータの入力

シンボルデータは、常に引用符（” “）で囲わなければなりません。

**RSS-14** 及び**RSS Limit** では、**0~9** の数字を**13** 桁、**Application Identifier (AI)** の”**01**”はプリンタが内部的に付加しますのでバーコードデータとして入力の必要がありません。**RSS Expanded** は数字、英大小文字と**21** 個の句読文字に加えて特殊キャラクター**FNC1** がデータとして使用出来ます。**RSS Expanded** では**AI** もデータとして入力する必要があります。これらの入力方法は、別のところで説明します。

又、**RSS** シンボルが選択されている場合は、シンボルデータを以下のように入力することで、**RSS** コンポジットシンボルの印刷を行う事が出来ます。コンポジットシンボルのデータとしては、数字、英大小文字、スペース、**20** 個の句読文字に加えて特殊キャラクター**FNC1** がデータとして使用できます。

**RSS** コンポジットシンボルを印刷する場合のデータの入力方法**RSS** シンボルのデータに続けて”|”（バーチカルバー）を入力し、これに続けてコンポジットシンボルのデータを入力することで**RSS** コンポジットシンボルを印刷出来ます。

**RSS** シンボルで可読文字は、テキストコマンドを使って別に印刷しなければなりません。この場合シンボルと可読文字は同じデータになるようにしてください。

### 9.5.4 バーコードデータ開始 (BarCode Start:BCST)

**BCST** コマンドは、プリンタに対して、このコマンドの直後に続くデータを、**RSS** シンボルとして印刷するように指示します。**BCST** コマンドがないと、そのデータは通常のテキストであるとみなし、それをそのまま印刷します。

このコマンドの構文は次の通りです。

**BCST;**

### 9.5.5 バーコードデータ終了 (Barcode SToP:BSTP)

**BSTP** コマンドは、**BCST** と対になっていて、プリンタに**RSS** シンボルのデータの終了を知らせます。このコマンドは、最後のシンボルデータの直後に置かなければなりません。**BSTP** コマンドが発行されない場合プリンタはどこでシンボルが終わっていいか判りません。

このコマンドの構文は次の通りです。

**BSTP;**

次の例は、**RSS** シンボルを選択し、データを入力したものです

```
BSYM;64;1;  
BCST;"0361234567890";BSTP;
```

### 9.5.6 **RSS** シンボル

**RSS** シンボルは、**BSYM/BDEF** を使ってシンボルの種類を定義し、**BCST/BSTP** を使ってシンボルのデータの開始と停止を示し、シンボルを印刷するために、シンボルの密（**BCPI**）、シンボルの高さ（**BCSH**）を設定をする必要があります。

## NITTO DENKO

### 9.5.7 RSS 用のシンボル密度設定 (Barcode Character Par Inch:BCPI)

BCPI コマンドは、RSS のモジュール寸法を指定します。

このコマンドは、RSS のシンボルの各エレメントの幅を自動的に決定します。

このコマンドの構文は、次の通りです。

#### BCPI;A;

A=次の表のBCPI 密度のひとつです。

(ピケットフェンスバーコード)

BCPI 番号	モジュール寸法
0	2 ドット
1	3 ドット
2	4 ドット
3	5 ドット
4	6 ドット
5	7 ドット
6	8 ドット

(ラダーバーコード)

BCPI 番号	モジュール寸法
0	2 ドット
1	3 ドット
2	4 ドット
3	5 ドット
4	6 ドット

※IP6300V 互換モードのオン/オフの設定に関わらず上記の設定となります。

### 9.5.8 RSS シンボル高さ設定 (BarCode Symbol Height:BCSH)

BCSH コマンドは、RSS シンボルのシンボル高さを設定します。

スタック型シンボルの場合は、1 段当たりの高さとなります。

シンボルのセパレータバーの高さは、BCPI コマンドで設定されたモジュール寸法で展開されます。

このコマンドの構文は次の通りです。

- ・ピケットフェンスシンボルの場合

#### BCSH;YY;

YY=0.254mm の単位で設定したシンボル高さ

- ・ラダーシンボルの場合

#### BCSH;XX;

XX=0.127mm の単位で設定したシンボル高さ

初期値は、ピケットフェンスでは、50 (12.5mm)、ラダーでは、100 (12.7mm) になります。

## NITTO DENKO

### 9.5.9 Composite Components シンボル高さ設定(Two Dimension Cell Height:TDCH)

TDCH コマンドは、コンポジットシンボルの2D シンボル (PDF417) の1 段当たりのシンボル高さを設定します。

このコマンドの構文は次の通りです。

**TDCH;YY;**

YY=ドット数でシンボル高さ指定

初期値は、9 ドットです。

### 9.5.10 RSS Expanded Stacked の1 段当たりのセグメント数設定 ( Two Dimensional symbol Row andColumn:TDRC)

TDRC コマンドは、RSS Expanded Stacked の1 段あたりのセグメント数を設定します。

入力されたデータは設定された1 段あたりのセグメント数に従って展開され、1 段当たりに納まらない場合は、シンボルは2 段から11 段で展開されます。

このコマンドの構文は次の通りです。

**TDRC;R;C;**

R=常に0: (0 以外の値が設定されても無視されます)

C=2~22 の偶数 (奇数が設定された場合は、偶数に切り捨てを行います)

2 以下の場合は、2 に、22 以上の場合は22 として処理されます。

RSS コンポジットシンボルを展開する場合は、4 以上に設定する必要があります。

セグメントとは、バーコード符号化の最小単位です。
--------------------------

## NITTO DENKO

### 9.5.11 HRI のチェックデジット印刷

**RSS** シンボルでは、**HRI** としてチェックデジットを明示的に印刷する必要があります。  
**RCL** では、ドットフォント及びローテータブルフォントで**RSS** チェックデジット付きフォントに対応しておりプリンタが自動的に付加して印刷を行えます。  
ドットフォントを指定する場合は、**DDF** コマンドの国別指定のパラメータで設定が出来ます。

- ・ ドットフォントで指定する場合は、**DDF** コマンドで設定します。

このコマンドの構文は以下のようになります。

**DDF;A;B;**

**A**=印刷に使用するフォント種を指定して下さい。

**B=145** を指定して下さい。

上記以外の設定については、ドットフォント指定の項を参照ください。

- ・ ローテータブルフォントで指定する場合は、**DHR** コマンドで設定します。

このコマンドの構文は、以下のようになります。

**DHR;A;**

**A** の値は、下表の通り

<b>A</b>	フォント種
<b>\$13</b>	ローテータブル <b>RSS</b> チェックデジット付き
<b>\$213</b>	ローテータブル <b>90 x RSS</b> チェックデジット付き
<b>\$413</b>	ローテータブル <b>180 x RSS</b> チェックデジット付き
<b>\$613</b>	ローテータブル <b>270 x RSS</b> チェックデジット付き

### 9.5.12. RSS の使用例

**RSS** シンボルは、**RSS-14** と **RSS Limit** が0~9 の数字で**13** 桁 (固定) を**RSS Expanded** では数字、英大小文字、スペース、**20** 個の句読記号と機能コードとして**FNC1** を数字**74** 桁まで、又英数字は**41** 文字までデータを入力できます。

**RSS-14**、**RSS Limit** は、**14** 桁目にチェックキャラクターが付加されます。

**RSS Expanded** は、**AI** として**01** が入力された場合、これに続く**13** 桁でチェックデジットを求めて**14** 桁目をチェックデジットで書き換えを行いますのでダミーデータを入力しておいてください。

機能コードの入力方法

機能コードの**FNC1** をデータとして

指定する方法を次の表に示します。

データコード	入力データ文字
<b>FNC1</b>	<b>@A</b>

入力データが上記で表した以外の **@** キャラクターとのペアの場合は、**@** とそれに続くキャラクターを無視します。

## NITTO DENKO

次に、RSS-14 で印刷できるバーコードの例を示します。

```
~^"RSS14";1;0;120;0;  
SPB;HBR;0;VBR;90;  
BSYM;64;1;BCPI;4;BCSH;50;  
BCST;"0361234567890";BSTP;  
HBR;0;VBR;95;DDF;3;1;DFM;2;2;  
"(01)";  
DDF;3;145;"0361234567890";  
TRM;¥
```



## 10. テキストと可読文字

---

**RCL** は、バーコード・データと同じようにテキストデータ用チェックキャラクタを計算して付加する機能があります。

このセクションでは、テキストやバーコードの可読文字を制御するコマンドについて、ベクトルフォントとドットフォント及びアウトラインフォントに分けて説明します。

### 10.1. ベクトルフォント

(**KP4300**/**KP3000** はベクトルフォントに対応していません)

#### 10.1.1. テキスト・コマンド構文

ベクトルフォントを利用したテキストを指定するためには、ベクトルフォント指定 (**Define Human Readable:DHR**)、文字高さ設定 (**Define Character Height:DCH**)、文字幅設定 (**Define Character Width:DCW**)、文字間スペース (**Inter Character Space:ICS**) の 4 つのコマンドの構文は次の通りです。

**DHR;AA;DHC;YY;DCW;XX;ICS;XX;**

**AA=10** 進数 (16 進数で指定することも可能です。「10.1.4 ベクトルフォント指定」参照)

**YY=**垂直ピクセル (0.254mm) 数

**XX=**水平ピクセル (0.127mm) 数

これらコマンドのどれかがコマンド列の中にない場合、初期値を採用します。各コマンドの初期値 (デフォルト値) は次の通りです。

**DHR;32768;DCH;20;DCW;40;ICS;4;**

#### 10.1.2. ベクトルフォント

ベクトルフォントは、ティプトン・ゴシック・プロポーションナル、ティプトン・ゴシック・モノスペースとローテータブルの 3 種類です。ティプトン・ゴシックは、ボールド、イタリック、縮小 (通常の 80%) あるいは拡大 (通常の 120%) のスタイルでも印刷できます。ローテータブルフォントは、8 つの方向に回転でき、7 つのヨーロッパの言語を印刷できます。

## NITTO DENKO

### 10.1.3. キャラクタ・セット

ティプトン・ゴシックは、標準的な **96** 文字の **ASCII** 文字セットです。ASCII 文字の **32** (スペース) から **127 (DEL)** までの文字を印刷することができます。ローテータブルフォントは標準の **ASCII** 文字セットのうち、**6** つの句読点 ( , { , ~ , | , \ ) の代わりに **6** つの外国文字を印刷できます。この **2** 種類のキャラクタ・セットを下に示します。

```
TIPTON GOTHIC
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~

ROTATABLE:
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ^_
`abcdefghijklmnopqrstuvwxyzR8AG
```

### 10.1.4. ベクトルフォント指定 (Define Human Readable : DHR)

ベクトルフォントのフォント種類、スタイル、文字の回転は、**DHR** コマンドで指定します。

フォントは **10** 進数あるいは **16** 進数で指定します。このコマンドの構文は次の通りです。

**DHR;AAAA;**                    または    **DHR;\$HHHH;**  
**AAAA=10** 進数                    **HHHH=16** 進数  
**\$=16** 進符号

ベクトルフォント及びローテータブルフォントのスタイル、回転方向に対する **16** 進数と **10** 進数による値を下の表に示します。

フォント名	スタイル	16 進数の値	10 進数の値
ティプトン・ゴシック・プロボーション		\$8000	32768
ティプトン・ゴシック・プロボーション	ボールド	\$C000	49152
ティプトン・ゴシック・プロボーション	拡大	\$8800	34816
ティプトン・ゴシック・プロボーション	縮小	\$9000	36864
ティプトン・ゴシック・プロボーション	イタリック	\$2000	8192
ティプトン・ゴシック・プロボーション	ボールド、イタリック	\$4000	16384
ティプトン・ゴシック・プロボーション	拡大イタリック	\$0800	2048
ティプトン・ゴシック・プロボーション	縮小イタリック	\$1000	4096
ティプトン・ゴシック・モンスペース		\$8002	32770
ティプトン・ゴシック・モンスペース	ボールド	\$C002	49154
ティプトン・ゴシック・モンスペース	拡大	\$8802	34818
ティプトン・ゴシック・モンスペース	縮小	\$9002	36866
ティプトン・ゴシック・モンスペース	イタリック	\$0002	2
ティプトン・ゴシック・モンスペース	ボールド、イタリック	\$4002	16386
ティプトン・ゴシック・モンスペース	拡大イタリック	\$0802	2050
ティプトン・ゴシック・モンスペース	縮小イタリック	\$1002	4098

**NITTO DENKO**

フォント名	スタイル	16進数の値	10進数の値
ローテータブル		\$ 1	1
ローテータブル 45x	左->右	\$ 101	257
ローテータブル 90x	下->上	\$ 201	513
ローテータブル 135x	右->左	\$ 301	769
ローテータブル 180x	上下反転	\$ 401	1025
ローテータブル 225x	右->左	\$ 501	1281
ローテータブル 270x	上->下	\$ 601	1537
ローテータブル 315x	左->右	\$ 701	1793
ローテータブル MOD10,MOD11 付き			
ローテータブル		\$ 10	
ローテータブル 90x	下->上	\$ 210	16
ローテータブル 180x	上下反転	\$ 410	5
ローテータブル 270x	上->下	\$ 610	28
			10
			40
			15
			52
ローテータブル MOD43 付き			
ローテータブル		\$ 11	17
ローテータブル 90x	下->上	\$ 211	529
ローテータブル 180x	上下反転	\$ 411	1041
ローテータブル 270x	上->下	\$ 611	1553
ローテータブル MOD43 付き 読取可能な Casocode Code 128)			
ローテータブル		\$ 12	18
ローテータブル 90x	下->上	\$ 212	530
ローテータブル 180x	上下反転	\$ 412	1042
ローテータブル 270x	上->下	\$ 612	1554
各国語キャラクタセット			
英語キャラクタセット		\$ 3	3
フランス語キャラクタセット		\$ 4	4
スウェーデン/フィンランド語キャラクタセット		\$ 5	5
デンマーク語キャラクタセット		\$ 6	6
イタリア語キャラクタセット		\$ 7	7
ドイツ語キャラクタセット		\$ 8	8
スペイン語キャラクタセット		\$ 9	9

**注**

チェックキャラクタ付きの Modulo10,Modulo11,Modulo43 をそれぞれ MOD10、MOD11、MOD43 と記述しています。



## NITTO DENKO

RCL で、印刷できるいろいろなベクトルフォントとスタイルを示します。

Tipton Gothic Proportional:  
Normal: ABCD~~e~~fgh12345  
Bold: ABCD~~e~~fgh12345  
Expanded: ABCD~~e~~fgh12345  
Condensed: ABCD~~e~~fgh12345  
*Italic* ABCD~~e~~fgh12345  
*Bold-Italic* ABCD~~e~~fgh12345  
*Expanded Italic* ABCD~~e~~fgh12345  
*Condensed Italic* ABCD~~e~~fgh12345

Tipton Gothic Monospace:  
Normal: ABCD~~e~~fgh12345  
Bold: ABCD~~e~~fgh12345  
Expanded: ABCD~~e~~fgh12345  
Condensed: ABCD~~e~~fgh12345  
*Italic*: ABCD~~e~~fgh12345  
*Bold-Italic*: ABCD~~e~~fgh12345  
*Expanded Italic*: ABCD~~e~~fgh12345  
*Condensed Italic*: ABCD~~e~~fgh12345

Rotatable:  
Normal: ABCD~~e~~fgh12345  
Upside Down: ABCD~~e~~fgh12345

**NITTO DENKO**

次の表は、外国語のキャラクタ・セットの16進数による値のリストです。(例 21H=!)

US CHARACTER SET		DHR 1															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/		
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	U	H	X	Y	Z	[	ç	]	^	↑	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	ä	ö	å	ü		

UK CHARACTER SET		DHR 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2	!	"	£	\$	%	&	'	(	)	*	+	,	-	.	/		
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	U	H	X	Y	Z	[	\	]	^	_	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	ä	;	å	ü		

FRENCH CHARACTER SET		DHR 4															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/		
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	à	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	U	H	X	Y	Z	'	ç	]	^	↑	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	é	ù	è	"		

SWEDISH/FINNISH CHARACTER SET		DHR 5															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/		
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	é	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	U	H	X	Y	Z	Ä	Ö	Å	Ü	↑	
6	é	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	ä	ö	å	ü		

NITTO DENKO

DANISH CHARACTER SET DHR 6

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	U	W	X	Y	Z	Æ	Ø	À	^	↑
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	æ	ø	à	ü	.

ITALIAN CHARACTER SET DHR 7

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	U	W	X	Y	Z	·	ç	é	^	↑
6	ù	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	à	ò	è	ì	

GERMAN CHARACTER SET DHR.8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	U	W	X	Y	Z	Ä	ö	Ü	^	↑
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	

SPANISH CHARACTER SET DHR 9

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	U	W	X	Y	Z	Í	Ñ	¿	^	↑
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	í	ñ	ü	ü	

## NITTO DENKO

### 10.1.5. キャラクタ・サイズを選択

文字の大きさを選択するには、文字高さ設定 (**Define Character Height:DCH**) と文字幅設定 (**Define Character Width:DCW**) の 2 つのコマンドを使います。フォントによって異なりますが、高さは **1.27mm**～最大印字長の範囲、幅は **1.27mm**～最大印字幅の範囲で設定できます。さらに、文字間スペース (**Inter Character Space:ICS**) コマンドは、**0.254mm** から **63.5mm** までの範囲で文字の間のスペースの量を設定します。これらのコマンドの構文は次の通りです。

**DCH;YY;DCW;XX;ICS;XX;**

**YY**=垂直ピクセル(**0.254mm**)数

**XX**=水平ピクセル(**0.127mm**)数

#### 10.1.5.1. 文字高さ設定 (**Define Character Height : DCH**)

**DCH** コマンドは、ベクトルフォントを使ったテキストと可読文字の高さをピクセル単位で設定します。このコマンドは **DHR** と **DCW** と共に使って、いろいろなフォントと大きさのテキストを印刷します。**DHR** コマンドでティプトン・ゴシックを選んだときは、文字の最小の高さは **10** ピクセル(**2.54mm**)、最大の高さは **330** ピクセル(**83.8mm**) です。ローテータブルフォントを選択したときは、最小高さは **5** ピクセル(**1.27mm**)、最大の高さは最大印字長です。

#### 注

ティプトン・ゴシック・フォントの縮小やイタリックを使用する場合は、**DCH;30;** 以上の文字サイズを使用してください。

文字の実際の高さは、**DCH** コマンドで指定した値の **70%**になります。その理由は、**pypgj** のような小文字のディセンダ (文字のベースラインより下に出る部分) のためのスペースが必要だからです。大文字は、**DCH** で設定した高さの上側 **70%**に印刷され、**y** のような小文字のディセンダ文字は下側 **70%**に印刷されます。小文字 (a, c, e, o, u など) は、指定された高さの訳 **50%**の高さで印刷されます。

#### 10.1.5.2. 文字幅設定 (**Define Character Width : DCW**)

**DCW** コマンドは、ベクトルフォントを使ったテキストと可読文字の幅を、水平ピクセル(**0.127mm**)の半分の単位で設定します。その公式は次の通りです。

幅 (X ピクセル) = **DCW/2**

ティプトン・ゴシック・フォントの **DCW** は、最小値が **20** で、最大値は **1000** です。ローテータブルフォントは、最小値 **10**、最大値 **1536** です。

#### 注

ティプトン・ゴシック・フォントの縮小やイタリックを使用する場合は、**DCW;60;** 以上の文字サイズで使用してください。

## NITTO DENKO

### 10.1.5.3. 文字間スペース (Inter Character Space : ICS)

ICS コマンドは、水平ピクセル (0.127mm) 単位で文字間の距離を設定します。最小値は 2 ピクセル (0.254mm) で、最大値は 500 ピクセル (63.5mm) です。初期値は 4 (0.508mm) です。ICS で指定された値は、次の文字を印刷する前に、DCW コマンドで指定された各文字の幅に加算されます。ローテータブルとティプトン・ゴシック・モノスペース・フォントでは、文字間スペースは通常同じですが、ティプトン・ゴシック・プロポーションアル・フォントでは、文字間スペースはいくらか違います。それはいくつかの文字 (W、M など) は幅が広く、他の文字 (I、T など) は幅が狭くなるからです。

#### 注

ティプトン・ゴシック・モノスペース・フォントで、縮小あるいは拡大印刷する場合、ICS コマンドを使ってスペースの調整をしてください。そうしないと、文字のスペースが不適切になります。縮小印刷では、ICS 値を例えば 4 から 2 へと半分にし、また拡大印刷では、ICS 値を例えば 4 から 12 に 3 倍すると良いでしょう。

ICS コマンドでは、通常、偶数のピクセルで指定します。奇数で指定すると、自動的に次の偶数に切り上げられます。これは、ICS で 3 を指定しても 4 と同じになり、9 では 10 を指定したのと同じであることを意味します。

## NITTO DENKO

### 10.1.5.4. 文字の位置決め

ラベル上のテキストと可読文字の位置は、絶対位置移動と相対位置移動コマンドの両方を使って指定します。

ベクトルフォントの文字の印刷開始点は、第1文字の左下隅です。それ以降の文字は、この点から順に印刷されます。

下の例のベクトルフォントの場合、大文字 **A** の左下隅が印刷開始点を示します。

```
~ ^ "POINT";1;0;160;0;SPB;UTOF;1600;  
HBR; 50;VBR; 10;DHR; 1;DCH; 10;DCW; 20;ICS; 4;  
"TOF";  
HBR; 0;VBR; 0;DHL;100; 0; 40;  
HBR; 0;VBR; 0;DVL;120; 0;150;  
HBR; 0;VBR; 0;DHL;100;147; 40;  
HBR; 75;VBR;115;DHR; $201;DCH; 10;DCW; 20;ICS; 4;  
"DCH=150";  
HBR;125;VBR;105;DHR;$8000;DCH;150;DCW;300;ICS; 10;  
"ABC";  
HBR; 0;VBR; 0;DHR;120;105;550;EOL;  
HBR;125;VBR;120;DHR; $1;DCH; 10;DCW; 20;ICS;10;  
"VBR=105";  
TRM;¥
```

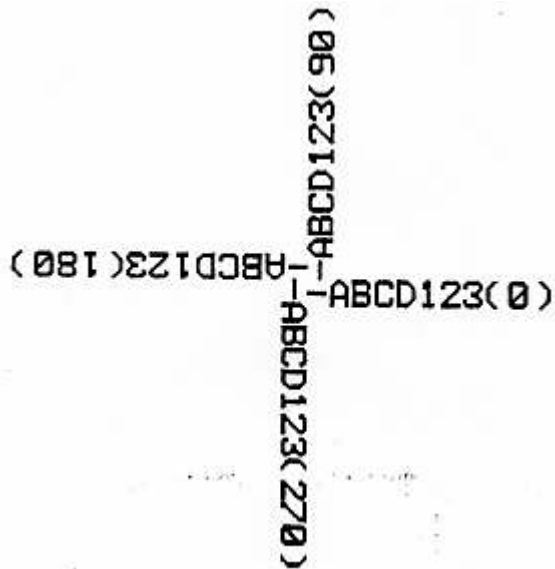


## NITTO DENKO

### 10.1.5.5. 回転テキストの印刷開始位置

ローテータブルフォントでテキストを回転するときは、開始基準点は回転の方向によって変わります。回転させたテキストの印刷開始位置を次に示します。

```
~ ^ "KAITEN";1;0;350;0;SPB;UTOF;3600;  
HBR;350;VBR;165;  
DHR; $1;DCH;20;DCW;40;ICS; 4; # orient 0 #;  
"-ABCD123(0)";HOME;  
DHR;$201;DCH;20;DCW;40;ICS; 4; # orient 90 #;  
"-ABCD123(90)";HOME;  
DHR;$401;DCH;20;DCW;40;ICS; 4; # orient 180 #;  
"-ABCD123(180)";HOME;  
DHR;$601;DCH;20;DCW;40;ICS; 4; # orient 270 #;  
"-ABCD123(270)";HOME;  
TRM;¥
```



### 10.1.6. 文字データの入力

文字は引用符("...")で囲って指定します。1つの文字列に必要なだけの文字を指定することができます。ただし、文字列がラベル上の一定のスペース内に納まるように注意しなければなりません。テキストの入力に必要な構文の例を示します。

```
DHR;$8000;DHC;20;DCW;40;ICS;4;  
"Text String";
```

## NITTO DENKO

### 10.1.7. チェックキャラクタ付ローテータブルフォント

ローテータブルフォントでは、プリンタ・ソフトウェアを使ってチェックキャラクタを計算し、テキスト・データに付加することができます。チェックキャラクタには、モジュラス 10、モジュラス 11、モジュラス 43 の 3 種類があります。

#### 10.1.7.1. MOD10、MOD11 チェックキャラクタを付けたローテータブルフォント

DHR のパラメータに、\$10、\$210、\$410、\$610 のいずれかを指定した場合、MOD10 か MOD11 のいずれか、または両方のチェックキャラクタが計算され、印刷するテキスト・データの指定した位置に付加されます。位置の指定は@と#によって行います。

@は MOD11 チェックキャラクタ、#は MOD10 チェックキャラクタです。

例

DHR;\$10;---;"#5678";---

この印刷結果は次のようになります。

□ □ 5 6 7 8  
↑ ↑  
データ"5678"に対する MOD10 チェックデジット  
データ"5678"に対する MOD11 チェックデジット

この機能は、CodaBar の可読文字に便利です。

Modulo10、Modulo11 の計算方法については、7.5.2 を参照してください。

#### 10.1.7.2. MOD43 チェックキャラクタを付けたローテータブルフォント

DHR のパラメータに、\$11、\$211、\$411、\$611 のいずれかを指定した場合、MOD43 チェックキャラクタが計算されて、データの終わりに加算されます。

例

DHR;\$11;---;"ABC123"---

このプリント結果は次のようになります。

A B C 1 2 3 □  
↑  
データ"ABC123"に対する MOD43 チェックキャラクタ

この機能は、Code39 MOD43 の可読テキストにべんりです。

Modulo 43 の計算方法については、7.3 を参照してください。



## NITTO DENKO

### 10.1.7.3. MOD10 チェックキャラクタ付きローテータブルフォント (Casecode Code 128 の可読文字用)

DHR のパラメータに \$12、\$212、\$412、\$612 を指定した場合、入力データに対して MOD10 チェックキャラクタが計算されてデータの後ろに付加されます。この機能は Casecode Code 128 の可読文字に使用します。データの桁数を検査しません。そこで、データの長さについてはプログラマーが責任を持たなければなりません。

Casecode Code 128 の可読文字に使う MOD10 チェックキャラクタは次のようにして計算します。

ステップ 1: データの最も小さい桁から順番に 1,2,3.. と番号を付けます。

ステップ 2: 奇数番目の桁の数を加えてその結果を 3 倍します。

ステップ 3: 偶数番目の桁の数を加えます。

ステップ 4: ステップ 2 とステップ 3 の結果を加えます。

ステップ 5: ステップ 4 の結果より大きい最小の 10 の倍数から、ステップ 4 の結果を引きます。これが MOD10 チェックキャラクタ、C です。

例

入力データ = "123541"

位置	6	5	4	3	2	1
データ	1	2	3	5	4	1
	<hr/>					
	2	+	5	+	1	= 8 × 3 = 24
	1	+	3	+	4	= 8
	<hr/>					
	合計					32

$C = 40 - 32 = 8 \leftarrow \text{MOD}10$

したがって、可読文字として "1235418" が印刷されます。

## 10.2. ドットフォント

### 10.2.1. テキスト・コマンド構文

ドットフォントを使ったテキストを指定するには、ドットフォント指定 (Define Dot Font:DDF)、文字拡大率指定 (Dot Font Magnification:DFM)、文字間スペース (Dot Font Space:DFS)、印刷方向指定 (Dot Font Orientation:DFO) の 4 つのコマンドを使用します。

これら 4 つのコマンドの構文は以下の通りです。

**DDF;A;B;DFM;XM;YM;DFO;O;K;DFS;X;**

A=ドットフォントタイプ

B=国別コード

XM=横方向への文字拡大率

YM=縦方向への文字拡大率

O=文字列の印刷方向

K=文字の方向 (縦書き/横書き)

## NITTO DENKO

**X**=文字間のスペース

これらコマンドのどれかがコマンド列の中にない場合、初期値を採用します。  
各コマンドの初期値は次の通りです。

**DDF;3;1;DFM;1;1;DFO;1;1;DFS;2;**

ドットフォントを指定しなかった場合は、ベクトルフォントで印刷されます。

### 10.2.2. フォント

ドットフォントは拡大したり、文字の間隔を指定したり印字することができます。

各ドットフォントの大きさは **20** 章のモデル毎文字サイズ一覧表をご覧ください。

### 10.2.3. キャラクタ・セット

英数字のドットフォント (**XS**、**SS**、**S**、**M**、**L**、**OCR-B** の各サイズ) は、標準的な **96** 文字の **ASCII** 文字セットです。**ASCII** 文字の **32** (スペース) から **127** (**DEL**) までの文字を印刷することができます。

英数字ドットフォントの印刷例を下に示します。



## NITTO DENKO

```
L SIZE
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^
_`abcdefghijklmnopqrstuvwxyz{|}~
```

OCR-B フォントは、JIS-X9001 光学式文字認識のための字形（英数字）で定められているサイズ1、サブセット3から、通貨記号と分離記号を除いた94文字が使用できます。ただし、これらの文字は光学式読み取り装置で読めることを保証するものではありません。印刷例を以下に示します。

```
OCR-B
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^
_`abcdefghijklmnopqrstuvwxyz{|}¥
```

漢字（16）、漢字（24）は、ともにJIS-X0208 情報交換用漢字符号系で定められている第一水準、第二水準の文字を使用することができます。2種類のサイズの漢字印刷例を下に示します。

```
漢字 16ドット
五 啞 娃 阿 哀 愛 挨 埃 始 達 葵 西 菴 蕪 握 澀 旭 董 芦 雙 梓 压 靴
```

```
漢字 24ドット
五 啞 娃 阿 哀 愛 挨 埃 始 達 葵 西 菴 蕪 握 澀 旭 董 芦 雙 梓 压 靴
```

### 10.2.4. ドットフォント指定 (Define Dot Font : DDF)

ドットフォントの種類、国別コードはドットフォント指定コマンド (Define Dot Font:DDF) で指定します。各フォントの種類は、21章のフォントの選択表をご覧ください。

英数字フォントを指定した時には、さらに国別コードを指定することによって、各国別の通貨記号やウムラウト文字のような、国によって固有の文字を印刷することができます。このとき、国別コードで日本を選択すると、JIS-X0201 情報交換用符号で定められているローマ文字用図形キャラクタを印刷することができます。

OCR-B 及び漢字フォントを指定した時は、国別コードは10（日本）を指定してください。英数字フォントに対し、国別コード10（日本）を指定した時に印刷されるコードとフォントは次の通りです。

## NITTO DENKO

次の表は外国語のキャラクタ・セットの 16 種類による値のリストです。  
 (例 21H=!)

US CHARACTER SET		DDF; 4; 1;
	0	1 2 3 4 5 6 7 8 9 A B C D E F
2	!	" # \$ % & ' ( ) * + , - . /
3	0	1 2 3 4 5 6 7 8 9 : ; < = > ?
4	@	A B C D E F G H I J K L M N O
5	P	Q R S T U V W X Y Z [ \ ] ^ _
6	`	a b c d e f g h i j k l m n o
7	p	q r s t u v w x y z Ä Ö Å Æ Ü

UK CHARACTER SET		DDF; 4; 3;
	0	1 2 3 4 5 6 7 8 9 A B C D E F
2	!	" £ \$ % & ' ( ) * + , - . /
3	0	1 2 3 4 5 6 7 8 9 : ; < = > ?
4	@	A B C D E F G H I J K L M N O
5	P	Q R S T U V W X Y Z [ \ ] ^ _
6	`	a b c d e f g h i j k l m n o
7	p	q r s t u v w x y z Ä Å Æ Ü

FRENCH CHARACTER SET		DDF; 4; 4;
	0	1 2 3 4 5 6 7 8 9 A B C D E F
2	!	" # \$ % & ' ( ) * + , - . /
3	0	1 2 3 4 5 6 7 8 9 : ; < = > ?
4	à	A B C D E F G H I J K L M N O
5	P	Q R S T U V W X Y Z [ \ ] ^ _
6	`	a b c d e f g h i j k l m n o
7	p	q r s t u v w x y z é û è "

NITTO DENKO

SWEDISH/FINNISH CHARACTER SET DDF: 4; 5;	
	0 1 2 3 4 5 6 7 8 9 A B C D E F
2	! " # \$ % & ' ( ) * + , - . /
3	0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4	É Á B C D E F G H I J K L M N O
5	P Q R S T U V W X Y Z Ä Ö Å Ü †
6	é á b c d e f g h i j k l m n o
7	p q r s t u v w x y z ä ö å ü

DENISH CHARACTER SET DDF: 4; 6;	
	0 1 2 3 4 5 6 7 8 9 A B C D E F
2	! " # \$ % & ' ( ) * + , - . /
3	0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4	@ A B C D E F G H I J K L M N O
5	P Q R S T U V W X Y Z Æ Ø Å ^ †
6	` a b c d e f g h i j k l m n o
7	p q r s t u v w x y z æ ø å ü

ITALIAN CHARACTER SET DDF: 4; 7;	
	0 1 2 3 4 5 6 7 8 9 A B C D E F
2	! " # \$ % & ' ( ) * + , - . /
3	0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4	@ A B C D E F G H I J K L M N O
5	P Q R S T U V W X Y Z ` ¢ é ^ †
6	ù a b c d e f g h i j k l m n o
7	p q r s t u v w x y z à ò è ì

GERMAN CHARACTER SET DDF: 4; 8;	
	0 1 2 3 4 5 6 7 8 9 A B C D E F
2	! " # \$ % & ' ( ) * + , - . /
3	0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4	@ A B C D E F G H I J K L M N O
5	P Q R S T U V W X Y Z Ä Ö Ü ^ †
6	` a b c d e f g h i j k l m n o
7	p q r s t u v w x y z ä ö ü ß

**NITTO DENKO**

SPANISH CHARACTER SET		DDF; 4; 9;
	0 1 2 3 4 5 6 7 8 9 A B C D E F	
2	! " # \$ % & ' ( ) * + , - . /	
3	0 1 2 3 4 5 6 7 8 9 : ; < = > ?	
4	@ A B C D E F G H I J K L M N O	
5	P Q R S T U V W X Y Z i ñ ÿ ^ ↑	
6	` a b c d e f g h i j k l m n o	
7	p q r s t u v w x y z " ñ Ü Ü	

次の表は、**OCR-B** コードの **16** 進数による値のリストです。

OCR-B CHARACTER SET		DDF; 6; 1;
	0 1 2 3 4 5 6 7 8 9 A B C D E F	
2	! " # \$ % & ' ( ) * + , - . /	
3	0 1 2 3 4 5 6 7 8 9 : ; < = > ?	
4	@ A B C D E F G H I J K L M N O	
5	P Q R S T U V W X Y Z [ \ ] ^ _	
6	` a b c d e f g h i j k l m n o	
7	p q r s t u v w x y z {   } ¥	

漢字（16）、漢字（24）フォントを印刷する場合、データは **JIS** 又はシフト **JIS** 漢字コードで入力します。又 **JIS** コードを使用する場合は例のように漢字文字列の前後に漢字シフトイン(**KI:ESC K,16** 進数で **1B4B**)と漢字シフトアウト(**KO:ESC H, 16** 進数で **1B48**)を付加しなければなりません。

例： 漢字を印刷する場合

**DDF;8;10;"K 漢字 H"; JIS** コードの場合

(引用符で囲まれた文字列 “**・K 漢字 H**” の **16** 進数は、**1B4B**、**3441**、**3B7A**、**1B48** です。)

**DDF;8;10;"漢字"; シフト JIS** コードの場合

(引用符で囲まれた文字列 “漢字” の **16** 進数は、**8ABF**、**8E9A** です。)

## NITTO DENKO

### 10.2.5. キャラクタ・サイズを選択

ドットフォントの文字の大きさを拡大するには、文字拡大率指定コマンド (**Dot Font Magnification:DFM**) を使用します。ドットフォントは、縦横ともに **1 倍から 16 倍**までの整数倍で拡大することができます。

#### 10.2.5.1. ドットフォントの文字拡大 (**Dot Font Magnification : DFM**)

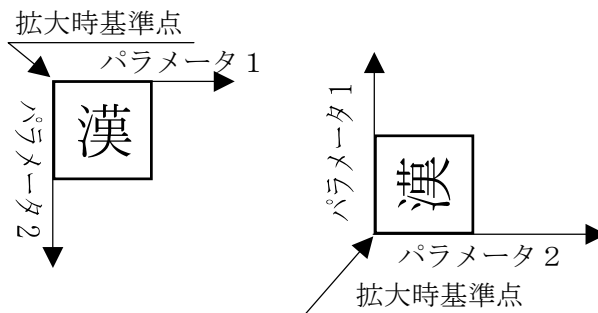
**DFM** コマンドは、縦横それぞれの方向の文字の拡大率を設定します。このコマンドの構文は次の通りです。

**DFM;XM;YM;**

**XM**=文字の横方向の拡大率。1～16 までの整数で指定。

**YM**=文字の縦方向の拡大率。1～16 までの整数で指定。

印刷開始の基準点と、拡大する場合の拡大の方向の関係は次のようになります。



#### 10.2.5.2. ドットフォントの文字間スペース (**Dot Font Space : DFS**)

ドットフォントの文字間スペースは、**DFS** コマンドで **0.127mm** 単位で設定します。このコマンドの構文は次の通りです。

**DFS;X;**

**X**=スペース量 (**0.127mm** 単位)

実際のスペースは  **$X \times 0.127\text{mm}$**  になります。

**DFS** で文字間スペースを指定しないときのドットフォントの文字間スペースの初期値は **2 (0.254mm)** です。

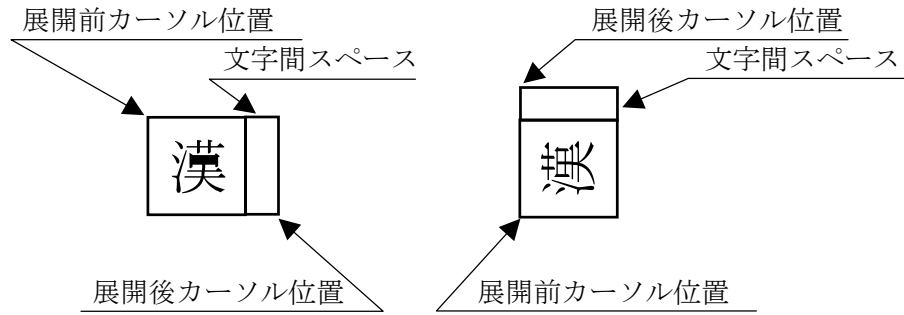
#### 10.2.5.3. 文字の位置決め

ラベル上のテキストと可読文字の位置は、絶対位置移動と相対位置移動コマンドの両方を使って指定します。(第 4 章参照)

ドットフォントの文字の印刷開始基準点は、第 1 文字の左上隅です。それ以降の文字は、この点から順に印刷されます。

ドットフォントの印刷開始点を示します。

NITTO DENKO



10.2.5.4. ドットフォントの印刷方向指定 (Dot Font Orientation : DFO)

ドットフォントで印刷する際の、文字列の印刷方向と文字の方向 (縦書き / 横書き) は、**DFO** コマンドで設定します。  
このコマンドの構文は次の通りです。

**DFO;O;K;**

**O** = 文字列の印刷方向

**K** = 文字の方向 (縦書き / 横書き)

文字列の印刷方向	<b>O</b>
0°	<b>1</b>
90°	<b>2</b>
180°	<b>3</b>
270°	<b>4</b>

文字の方向	<b>K</b>
横書き	<b>1</b>
縦書き	<b>2</b>



**NITTO DENKO**

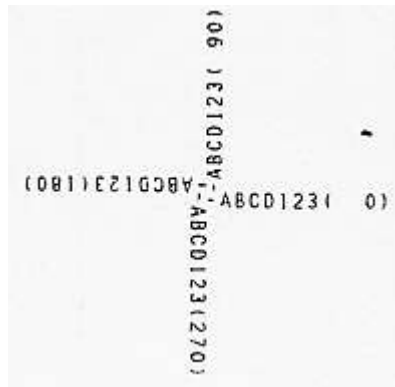
DFO コマンドでいろいろな印刷方向を指定した場合の印刷例を示します。

K \ O	1	2	3	4
1	漢 字	漢 字	字 漢	漢 字
2	漢 字	漢 字	字 漢	漢 字

以下にドットフォントを用いたプログラムとその印刷サンプルを示します。

```

~^ "DOTFONT";1;0;400;0;SPB;UTOF;4100;
HBR;300;VBR;200;
DDF; 4; 1;DFM; 1; 1;DFS; 5;
DFO; 1; 1; # rot 0 #;
"-ABCD123( 0)";HOME;
DFO; 2; 1; # rot 90 #;
"-ABCD123( 90)";HOME;
DFO; 3; 1; # rot 180 #;
"-ABCD123( 180)";HOME;
DFO; 4; 1; # rot 270 #;
"-ABCD123( 270)";HOME;
TRM;¥
    
```



### 10.3. アウトラインフォント

アウトラインフォントを使うことでギザギザのない文字を印刷することができます。

#### 10.3.1. テキスト・コマンド構文

アウトラインフォントを使ったテキストを指定するためには、ドットフォント指定、(Define Dot Font:DDF)、文字サイズ指定 (Define Character Size:DCS)、印字方向 (Dot Font Orientation:DFO)、文字間スペース (Dot Font Space:DFS) の 4 つのコマンドを使います。

これら 4 つのコマンドの構文は次の通りです。

**DDF;A;B;DCS;XD;YD;DFO;O;K;DFS;X;**

**A** = フォントの種類

**B** = 国別コード

**XD** = 水平ドット

**YD** = 垂直ドット

**O** = 文字列の印字方向

**K** = 文字の方向 (縦書き/横書き)

**X** = スペース量 **L(0.127mm)**

#### 10.3.2. フォント

アウトラインフォントは、**48~256** ドットの範囲で、縦/横自由に文字サイズを指定できます。アウトラインフォントは回転したり、文字の間隔を指定したりして印字することができます。

#### 10.3.3. キャラクタ・セット

英数字のアウトラインフォントは、標準的な **96** 文字の **ASCII** 文字セットです。

**ASCII** 文字の **32 (20H : スペース)** から **127 (7FH : DEL)** までの文字を印字することができます。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	“	#	\$	%	&	‘	(	)	*	+	,	—	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	—	

## NITTO DENKO

漢字は、**JIS-X0208** 情報交換用漢文字符号系で定められている第一水準、第二水準の文字を使用することができます。

亜 啞 娃 阿 哀 愛 挨 始 逢 葵 茜 穉 惡 握 渥
-------------------------------

### 10.3.4. フォントの選択

アウトラインフォントはドットフォント指定コマンド (**Define Dot Font:DDF**) で指定します。

このコマンドの構文は次の通りです。

**DDF:A;B;**

**A**=フォントの種類

**B**=国別コード

フォントの種類	A
アウトラインフォント (漢字コード入力)	100
アウトラインフォント (ASCII コード入力)	101

フォントの種類が、アウトラインフォント以外の場合は、ドットフォントが選択されません。

アウトラインフォントを指定したときは、国別コードは **10**(日本)を指定してください。

### 10.3.5. キャラクタ・サイズの選択

#### 10.3.5.1. アウトラインフォントの文字サイズ指定 (**Define Character Size : DCS**)

アウトラインフォントの文字の大きさを指定するには、文字サイズ指定コマンド (**DCS**) を使用します。アウトラインフォントは、縦横ともに **48** から **256** ドットまで設定することができます。

**DCS** コマンドは、縦横それぞれの方向の文字のピクセル数を設定します。

このコマンドの構文は次の通りです。

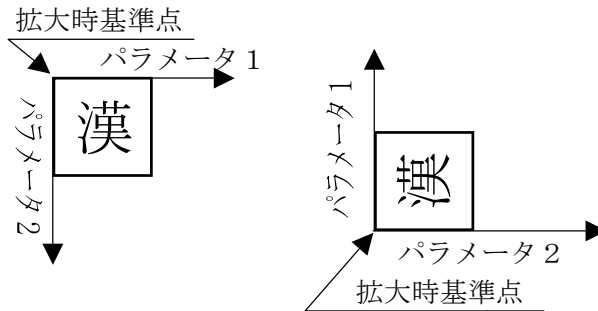
**DCS;XD;YD;**

**XD**=文字の横方向のドット数。 **48**~**256** までの整数で指定。

**YD**=文字の縦方向のドット数。 **48**~**256** までの整数で指定。

## NITTO DENKO

印字開始の基準点と文字方向の関係は次のようになります。



文字方向  $0^\circ$  の場合

文字方向  $90^\circ$  の場合

### 10.3.5.2. アウトラインフォントの文字間スペース (Dot Font Space : DFS)

アウトラインフォントの文字間スペースは、**DFS** コマンドで、**0.127mm** 単位で設定します。

このコマンドの構文は次の通りです。

**DFS;X;**

**X**=スペース量 (**0.127mm**)

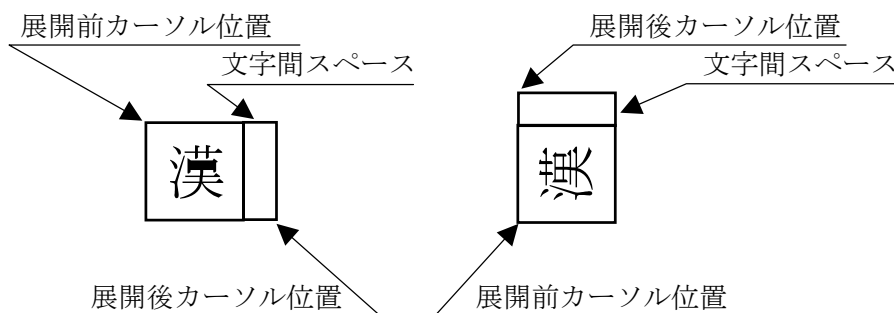
実際のスペース量は **X**×**0.127mm** になります。

**DFS** で文字間スペースを指定しないときのアウトラインフォントの文字間スペースの初期値は **2 (0.254mm)** です。

### 10.3.5.3. 文字の位置決め

アウトラインフォントの文字の印刷開始基準点は、第 1 文字の左上隅です。それ以降の文字は、この点から順に印刷されます。

アウトラインフォントの印字開始点を示します。



### 10.3.5.4. フォントの印刷方向指定 (Dot Font Orientation : DFO)

アウトラインフォントで印字する際の、文字列の印字方向と文字の方向（縦書き／横書き）は、**DFO** コマンドで設定します。

このコマンドの構文は次の通りです。

NITTO DENKO

DFO;O;K;

O=文字列の印刷方向

K=文字の方向 (縦書き/横書き)

文字列の印刷方向	O
0°	1
90°	2
180°	3
270°	4

文字の方向	K
横書き	1
縦書き	2

DFO コマンドでいろいろな印刷方向を指定した場合の印刷例を示します。

O \ K	1	2	3	4
1	漢   字	漢 字	字   漢	漢 字
2	漢 字	漢   字	漢 字	漢 字

10.3.6. 文字データの入力

文字は引用符("...")で囲って指定します。1つの文字列に必要なだけの文字を指定することができます。ただし、文字列がラベル上の一定のスペース内に納まるように注意しなければなりません。テキストの入力に必要な構文の例を示します。

DDF;101;10;DCS;48;48;DFO;1;1;DFS;2;

"Text String"

アウトラインフォントを漢字コード入力で印刷する場合、JIS またはシフト JIS 漢字コードで入力します。また、JIS コードを使用する場合は例のように漢字文字列の前後に漢字シフトイン (KI:ECK K,16 進数で 1B4B) と漢字シフトアウト (KO:ESC H,16 進数で 1B48) を付加しなければなりません。

## NITTO DENKO

例： 「漢字」を印字する場合

**DDF;100;10;"漢字";**                      **JIS** コードの場合  
(引用符で囲まれた文字列"漢字"の **16** 進数は、**1B4B 3441 3B7A 1B48** です。)

**DDF;100;10;"漢字";**                      シフト **JIS** コードの場合  
(引用符で囲まれた文字列"漢字"の **16** 進数は、**8ABF 8E9A** です。)

以下にアウトラインフォントを使った印字例を示します。

### 注

文字によって上下左右の余白のため文字の大きさが異なります。

NITTO DENKO

~ ^ "OUTLINE2";1;0;400;0;

SPB;VBR;10;HBR;50;

DDF;100;10;DFS; 5;DFO; 1; 1;DCS;48;48;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"48 x 48";

VBR;40;HBR;90;DDF;100;10;DCS;56;56;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"56 x 56";

VBR;70;HBR;140;DDF;100;10;DCS;64;64;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"64 x 64";

VBR;110;HBR;200;DDF;100;10;DCS;72;72;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"72 x 72";

VBR;150;HBR;250;DDF;100;10;DCS;80;80;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"80 x 80";

VBR;190;HBR;300;DDF;100;10;DCS;88;88;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"88 x 88";

VBR;240;HBR;350;DDF;100;10;DCS;128;128;

"漢字";

DDF; 3; 1;DFM; 1; 1;DFS; 5;DFO; 1; 1;

"128 x 128";

TRM;¥

漢字 48 × 48

漢字 56 × 56

漢字 64 × 64

漢字 72 × 72

漢字 80 × 80

漢字 88 × 88

漢字 128 × 128

## 11. 連番（連続番号）

---

バーコードを使用するときに、ラベルに連続番号（連番）を付けたい場合があります。**RCL** では、連番をつける方法が2種類あります。ラベルを印刷するごとにラベル全体を再構成方法と、ループを利用する方法です。ループは、連番を付けてラベルを印刷する部分に使い、必要な数のラベルが印刷されるまで繰り返します。

**RCL** では、数字連番 (**NUMeric:NUM**) と英字連番 (**ALPHabetic:ALPH**) と英数字連番 (**alphanumeric:BOTH**) の3種類の連続番号を指定できます。

ループを制御するコマンドの他に、連番データのタイプを指定することが必要です。バーコードの連番と、テキストあるいは可読文字の連番には、それぞれコマンドを発行しなければなりません。例えば **Code39**、**ITF**、**CodaBar** のバーコードと可読文字の両方を連番にしなければならない場合は、両方に1組ずつの連番コマンドを使わなければなりません。

### 11.1 ルーピング

複数のラベルを印刷する場合、バーコードやテキストをインクリメントまたはデクリメントするためにループを使うと、効率が非常に良くなります。これは、プリンタが毎回ラベル全体のイメージを生成するのではなく、ループの部分だけを再生成するからです。ここでは、ループをプログラムするときに使う、マーク (**MaRK:MRK**) とリピート枚数設定 (**Bar Code Label Count:BCLC**) とリターン (**RETurn:RET**) の3つのコマンドについて説明します。

ループの始まりと終わりを定義するには、マーク (**MaRK:MRK**) とリターン (**RETurn:RET**) の2つのコマンドが必要です。名前が示すように、**MRK** はプログラム内のループが開始される位置を示し、**RET** はプログラムの終わり、**TRM** の直前に置いて、**MRK** に戻って次のラベルを処理するようにします。

印刷するラベルの合計数は、印刷ラベルの数を定義するヘッダー・コマンド部分で制御します。フィールドをインクリメントまたはデクリメントするには、それぞれの番号のラベルの印刷枚数を指定する必要があるため、リピート枚数設定 (**Bar Code Label Count:BCLC**) コマンドを使います。

連番として印刷するバーコードとテキストは、次のラベルの印刷を行う前に枠内消去 (**Draw White BoX:DWBX**) コマンドで消去しなければなりません。

#### 11.1.1. マーク (MaRK : MRK)

**MRK** コマンドは、ループを使って繰り返す一連のコマンドの開始を定義します。**MRK** の構文と用法は次の通りです。

```

~^ "Example";3;0;400;50;
SPB;DHR;1;
MRK;HBR;100;VBR;50;"12345";

```



## NITTO DENKO

この場合、テキスト・フィールドの"12345"は、3枚のラベルを印刷する際にインクリメントまたはデクリメントされます。

1つのラベルで連番を設定できるフィールド数は32までです。ただし、複数フィールドを連番にする場合でも、MRKは最初のフィールドの前にだけ置きます。MRKを1つ指定したら、次のMRKを挿入する必要はありません。常にRETの前にある(最後の)MRKに戻るので、2つ目のMRKを挿入すると、最初のMRKは無視されます。

### 11.1.2. リピート枚数設定 (Bar Code Label Count : BCLC)

BCLC コマンドは、連番を印刷する際に印刷する同一ラベルの数を設定します。構文は次の通りです。

BCLC;A;

A=0~65,535 までの整数

BCLC コマンドがない場合、リピート枚数の既定値は1なので、ラベル1枚印刷するごとに連番の内容が増加/減少します。リピート枚数に2を指定すると、同じ連番のラベルを2枚ずつ印刷します。BCLC はバーコードと可読文字の両方のラベル数を制御しますので注意してください。

### 11.1.3. リターン (RETurn : RET)

RET コマンドは、RET より前にあるMRKにプログラムの制御を戻し、MRKとTRMの間の内容を繰り返し印刷します。MRKより前にあるデータは繰り返されず、印刷中は同じラベルを印刷します。印刷されたラベルの数がヘッダー内に指定した印刷枚数に達するまで繰り返しが行われます。

RETの使い方を次に示します。この短いプログラムは、数を1つずつ増やしながら3枚のラベルを印刷します。

```
~ ^ "IL9-1";3;0;100;0;          1 2 3 4 7
SPB;MRK;VBR;30;HBR;0;BCLC;1;
DWBX;0;-30;200;50;
"12345";SAL;1;EOL;
RET;TRM;¥                          1 2 3 4 6

                                      1 2 3 4 5
```

MRKと同じように、RET コマンドは1枚のラベルに1つでなければなりません。RETはMRKの後であればどこにでも置くことができますが、TRM コマンドの直前に置くのが最適です。

注

**BRK** コマンドを利用して 1 枚のラベルイメージを複数のコマンドブロックに分割している場合、**MRK**～**RET** は **TRM** で終了する最後のコマンドブロック内に入れる必要があります。

```

~^~      #分割された最初のブロックヘッダ# ;
SPB;
~
BRK;¥;  #最初のブロック終了# ;
~^~      #最初のブロックのヘッダ# ;
RSPB;
~
MRK;    #ループ開始# ;
~
RET;    #ループ終了# ;
TRM;¥   #最後のブロック終了# ;
    
```

11.1.4. **DWBX** を使った連番フィールドのクリア

バーコードを連番で印刷するときは、次のビットマップを印刷する前に、枠内消去 (**Draw White Box:DWBX**) コマンドを使ってバーコードと可読文字をビットマップからクリアしなければなりません。そうしないと、前のフィールドに新しいフィールドを重ねて印刷します。

**DWBX** コマンドは、**MRK** コマンドの後、連番フィールドを増加/減少させる前に置きます。**DWBX** は引数で指定した領域を消去するので、新しいバーコード・データをその位置に印刷できます。消去する位置を決めるときは、消去領域は左上隅が開始位置であり、一方、バーコードは左下隅が開始位置であることに注意してください。

次の例は、**DWBX** を使った **Code39** バーコードの消去と連番の使い方です。

```

~^ "DWBX";3;0;39;187;SPB;UTOF;0512;
MRK;HBR; 0;VBR; 0;DWBX; 0; 0;394;39;
HBR; 27;VBR; 28;BDEF; 1;BNEW; 2;BWEW; 5;BICG; 2;BCSH; 28;BCST;
"*RCL 00";BSAL;2;"*";BSTP;
HBR; 27;VBR; 30;DDF; 6;10;DFM; 1; 1;DFO; 1; 1;DFS; 8;
"*RCL 00";SAL;2;"*";
RET;
TRM;¥
    
```



## NITTO DENKO

### 11.1.5. プリンタの自動消去を使った連番フィールドのクリア

#### 11.1.5.1 自動消去オン (Erase Mode ON : EMON)

**RCL** の **Ver.4.00** より以前のモデルでは、連番を印刷するとき、次のビットマップを印刷する前に、枠内消去コマンド **DWBX** でイメージの消去を行う必要がありました。**RCL Ver.4.00** 以降では、連番部のイメージの消去をプリンタ自身に行わせ **DWBX** を省略することができます。この機能を有効にするためのコマンドが自動消去オンです。このモードをオンにした場合は、連番部の枠内消去が不要になります。

プリンタはこのコマンドにより自動消去の設定がオンされると、**SAL,BSAL、VLP、BVLP** で指定された、連番フィールドを次のビットマップ描画前にプリンタ自身で消去処理を行います。

#### 注

自動消去オンで、枠内消去コマンドが記述されている場合は、**2** 度同じイメージの消去を実行することになります。自動消去を使用される場合は、枠内消去コマンドは取り除いて下さい。

このコマンドの構文は次の通りです。

**EMON;**

次の例は、前の **DWBX** を使った例を自動消去オン (**EMON**) に変えた場合の例です。

```
~ ^ "DWBX";3;0;39;187;SPB;EMON;UTOF;0512;  
MRK;HBR; 0;VBR; 0;  
HBR; 27;VBR; 28;BDEF; 1;BNEW; 2;BWEW; 5;BICG; 2;BCSH; 28;BCST;  
"*RCL 00";BSAL;2;"*";BSTP;  
HBR; 27;VBR; 30;DDF; 6;10;DFM; 1; 1;DFO; 1; 1;DFS; 8;  
"*RCL 00";SAL;2;"*";  
RET;TRM;¥
```

## 11.2. 増分

**RCL** では、連番に数字、英字、英数字の **3** 種類のデータが使用できます。ここでは、増分を制御する **3** つのコマンドについて説明します。

### 11.2.1. 数字 (NUMeric : NUM)

**NUM** コマンドは、数字データ (**0-9**) だけをインクリメント (増加) またはデクリメント (減少) させるために使います。**NUM** は増分の既定値なので、**ALPH** あるいは **BOTH** コマンドを発行後、数字連番に戻る必要がない限り、**NUM** コマンドは必要ありません (下参照)。

**NUM** では、数字の **0123456789** 以外の文字は使用できません。その他の文字を使うと連番を印刷しません。このコマンドの構文は次の通りです。

**NUM;**

増分コマンドは連番させるフィールドの前になければなりません。複数の連番フィールドを印刷する場合には、別の増分コマンドが指定されない限り、すべてのフィールドに対して **NUM** コマンドが有効です。

次の例は、文字を **2** ずつ増加させる (**IDF=2**) 例で、正しい例と間違っただけの例を示します。

正しい例

**BCLC;1;IDF;2;NUM;**  
**"999898";SAL;3;**

間違っただけの例

**BCLC;1;IDF;2;NUM;**  
**"999A98";SAL;3;**

結果 (3 枚のラベルを印刷)

**999898**

**999A98**

**999900**

**999A00** [A は連番にならないことに注意してください]

**999902**

**999A02**

### 11.2.2. 英字 (ALPHabetic : ALPH)

**ALPH** コマンドは、大文字の **A** から **Z** までを連番として印刷するように指示します。小文字は連番にできません。このコマンドの構文は次の通りです。

**ALPH;**

英字増分は、**A** から **B** そして **C** へと **Z** まで進み、再び **A** まで戻って次の位の文字位置へ切り上げます。**NUM** と同じように、**ALPH** は連番するフィールドより前に指定しなければならず、別の増分コマンドが発行されるまで有効です。

次の例は、文字を **2** ずつ増加させる **ALPH** の使用例で、正しい例と間違っただけの例を示します。

正しい例

**BCLC;1;IDF;2;ALPH;**  
**"ZZZZZX";SAL;3;**

間違っただけの例

**BCLC;1;IDF;2;ALPH;**  
**"ZZZ3ZX";SAL;3;**

## NITTO DENKO

結果 (3 枚のラベルを印刷)

**ZZZZZX**

**ZZZZZZ**

**ZZZAAB**

**ZZZ3ZX**

**ZZZ3ZZ** [3 は連番にならないことに注意してください]

**ZZZ3AB**

### 11.2.3. 英数字 (BOTH)

**BOTH** コマンドは、数字と英字データの両方を連番にします。数字データとしては整数の **0** から **9** まで、英字文字は大文字の **A** から **Z** までが正しい文字として使用できます。小文字は連番にできません。このコマンドの構文は次の通りです。

**BOTH;**

**BOTH** ではデータは次の順序で文字が変わります。

**0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ012...**

**BOTH** コマンドは、連番にするフィールドの前に置かなければならず、他の増分コマンドが発行されるまで有効です。次の例は、文字を **12** ずつ増加させた (**IDF=12**) **BOTH** の使用例で、正しい例と間違っただけの例を示します。

正しい例

**BCLC;1;IDF;12;BOTH;**

**"ZZZ0Z96";SAL;4;**

間違っただけの例

**BCLC;1;IDF;12;BOTH;**

**"ZZZ0Z9b";SAL;4;**

結果 (ラベルを 4 枚印刷)

**ZZZ0Z96**

**ZZZ0ZA8**

**ZZZ0ZBA**

**ZZZ0ZCC**

**ZZZ0Z9 b** [小文字 **b** があるので連番ができません]

**ZZZ0Z9 b**

**ZZZ0Z9b**

**ZZZ0Z9b**

## 11.3. バーコードの連番

バーコードを連番にするには、すでに説明したコマンドの他にバーコード連番インクリメント・デクリメント設定 (**Bar Code Increment/Decrement:BCID**) とバーコードフィールド連番長さ保存 (**Bar Save Address Length:BSAL**) の 2 つのコマンドを使用します。**BCID** は各ラベルの増分または減分の量を定義し、**BSAL** は連番によって変更する文字数を指定します。

### 11.3.1. 連番インクリメント・デクリメント設定 (Bar Code Increment/Decrement : BCID)

**BCID** コマンドは、バーコードの連番によって値が変わるときに、減少または増加させる値を定義します。正の数は増加を指示し、負の数は減少を指示します。このコマンドの構文は次の通りです。

**BCID;A;**

**A=-32,768 から+32,767 の間の整数**

## NITTO DENKO

**BCID** コマンドが発行されない場合、既定値は **1** なのでバーコードの値が **1** ずつ増えます。例えば、**0** から **1** へ、そして **2** になります。次に、**BCID** コマンドの増分機能と減分機能を示します。

**BCID:10;#**バーコードを印刷するごとに **10** ずつインクリメントとする#;

**BCID:-10;#**バーコードを印刷するごとに **10** ずつデクリメントとする#;

### 11.3.2. バーコードフィールド連番長さ保存 (Bar Save Address Length : BSAL)

**BSAL** コマンドは、バーコードの連番にする値の直後に置きます。**BSAL** コマンドはバーコード・データのすべてのデータの中で連番にする数字の桁数を指示します。このコマンドには初期値がないので、連番を実行するためには必ず定義しなければなりません。このコマンドの構文は次の通りです。

**BSAL;A;**

**A=0** から **255** までの整数

**BSAL** コマンドに続けて、連番フィールドの最後の桁の位置から数えた数で、連番にする文字や数字の数を指定します。

**BSAL** は連番にする数の直後に置かなければなりません。

次に **BSAL** のいくつかの応用を示します。

**BDEF;1;BCSH;50;BCST;"\*123456789";BSAL;4;"\*";BSTP;**

この例では、**Code39** バーコードの後ろから **4** つの数字 (**6789**) が連番に使われています。**BSAL** はストップキャラクター (\*) の前に置かれていることに注意してください。**BSAL** をストップキャラクターの後ろに置くと、ストップキャラクターは特殊文字なので連番が停止します。

**BDEF;10;UMAG;0;BCST;"01234567890";BSAL;5;BSTP;**

2 番目の例では、**UPC-A** バーコードで終わりの **5** 桁の数字 (**67890**) が連番に使われます。最初の **6** 桁分の数字 (**012345**) は変化しません。

**BDEF;3;BCSH;50;BCPI;1;BCST;"A123456";BSAL;6;"@C";BSTP;**

3 番目の例では、**CodaBar** のバーコードで、**6** つのデータ文字 (**123456**) を連番に使っています。スタートキャラクター (**A**) とチェックキャラクター (**@**) とストップキャラクター (**C**) は連番になりません。

**BDEF;1;BCSH;75;NUM;BCST;"\*0123";BSAL;3;ALPH;"ABC";BSAL;3;"\*";BSTP;**

最後の例では、**Code39** のバーコードで、連番を **2** つの方法で使っています。整数の **123** は番号順に連番にし、文字の **ABC** はアルファベット順に連番にします。

## NITTO DENKO

### 11.3.3. バーコードフィールド連番長さ及び連番桁位置指定 (Barcode Variable field Length and Position : BVLP)

**BVLP** コマンドは、フィールド内のインクリメント又はデクリメントされる文字の桁数、及び文字列の桁位置を指示するために使用します。コマンドはバーコードフィールドと共に使います。このコマンドはインクリメント (デクリメント) するフィールドの最後の数の直後に指定します。

**BVLP;A;B;**

**A**=連番長さ      **0** から **255** までの整数で指定

**B**=連番桁位置    **1** から **255** までの整数で指定

**BVLP** コマンドに続く数 **B** でフィールドの最後の文字から数えて何桁目からを連番にするのかを示します (フィールドの最後の文字が **1** 桁目) コマンドに続く数 **A** では **B** で指定された桁位置の文字から数えた連番にする文字や数字の桁数を示します。

連番長さは (フィールドの総桁数 - (連番桁位置 - 1)) を超えることはできません。

又、連番桁位置はフィールドの総桁数を超えることはできません。

## 11.4. テキストの連番

テキストとバーコードの可読文字は、テキストが整数あるいは大文字であるかぎり連番になります。テキストの連番では、すでに説明したコマンド (**MRK**、**TRM**、**BCLC**、**DWBX**)に加え、テキストフィールド連番インクリメント・デクリメント設定 (**Increment / Decrement Field:IDF**) とテキストフィールド連番長さ保存 (**Save Address Length:SAL**) を使います。

ここでは **IDF** と **SAL** コマンドについて詳しく説明し、テキストの連番の例を示します。

### 11.4.1. テキストフィールド連番インクリメント・デクリメント設定 (Increment/Decrement Field : IDF)

**IDF** コマンドは、**BCID** コマンドがバーコードに作用するのと同様にテキストに対して作用し、テキストが増分される量を定義します。**IDF** の構文は次の通りです。

**IDF;A;**

**A**=**-32,768** から **+32,767** の間の整数

正の数は増加を指示し、負の数は減少を指示します。次の例では、テキストを増加したり減少させるための **IDF** の使い方を示しています。

**IDF:100;**#ラベルを印刷するごとにテキストを **100** ずつインクリメントとする#;

**IDF:-50;**#ラベルを印刷するごとにテキストを **50** ずつデクリメントとする#;

### 11.4.2. テキストフィールド連番長さ保存 (Save Address Length : SAL)

**SAL** コマンドは、フィールド内のインクリメントまたはデクリメントされる文字の桁数を指示するために使用します。**BSAL** がバーコードと共に使われたように、**SAL** はテキスト・フィールドと共に使います。どちらのコマンドも、インクリメント (デクリメント) するフィールドの最後の数の直後に指定します。

## NITTO DENKO

SAL の構文は次の通りです。

SAL;A;

A=0 から 255 までの整数

SAL コマンドに続く数で、フィールドの最後の文字から数えた連番にする文字や数字の数を示します。連番にする数字の総数は、フィールドの合計文字数を超えることはできません。例えば、**234789** という内容のフィールドを連番にする場合、指定できる数字の最大数は **6** です。

次に、SAL の使い方の例を示します。

```
~^ "Example1";3;0;400;50;
```

```
SPB;BCLC;1;IDF;-1;
```

```
MRK;"12345";SAL;1;
```

この例では、数字フィールド **12345** の一番最後の数字 (5) がデクリメントされます。

### 11.4.3. テキストフィールド連番長さ及び連番桁位置指定 (Variable field Length and Position : VLP)

VLP コマンドは、フィールド内のインクリメント又は、デクリメントされる文字の桁数及び文字列の桁位置を指示するために使用します。このコマンドはテキストフィールドと共に使います。

このコマンドはインクリメント (デクリメント) するフィールドの最後の数の直後に指定します。

VLP の構文は次の通りです。

VLP;A;B;

A=連番長さ            0 から 255 までの整数で指定

B=連番桁位置        1 から 255 までの整数で指定

VLP コマンドに続く数 **B** でフィールドの最後の文字から数えて何桁目からを指定された桁位置の文字から数えた連番にする文字や数字の桁数を示します。

連番長さは (フィールドの総桁数 - (連番桁位置 - 1)) を超えることはできません。

又、連番桁位置はフィールドの総桁数を超えることはできません。

次に VLP の使い方を示します。

```
~^ "Example1";3;0;400;50;
```

```
SPB;IDF;1;
```

```
MRK;"1234567890";VLP;3;5;
```

この例では数字フィールドの **1234567890** の一番最後から 5 番目の数字 **6** から **3** 桁 (**456**) をインクリメントします。



## NITTO DENKO

### 11.4.4. 指定文字削除連番機能（スキップ機能）（EXCePt : EXCP）

**EXCP** コマンドにより連番中、特定の文字のみ印字しないようにする（特定文字スキップ機能）ことができます。

**EXCP;CCC;**

CCC=削除したい文字を昇順に並べる

**C** で設定できる文字は **0~9**、**A~Z** の文字です。

数字連番 (**NUM**)、英字連番 (**ALPH**)、英数字連番 (**BOTH**) と組み合わせる事により、連番中、特定の文字を削除します。

削除文字として **0~9**、**A~Z** 以外の文字がある場合、削除文字が昇順に並んでいない場合は、コマンドエラーとなります。

- ・ **1** フォーマット内で複数の削除文字の、違う設定はできません。  
複数個の **EXCP** コマンドが入力された場合は、最後の **EXCP** コマンドでの指定が有効になります。
- ・ 数字連番 (**NUM**)、英字連番 (**ALPH**)、英数字連番 (**BOTH**) 以外の組み合わせはできません。また、一つのフォーマット内で、これらの連番の混在はできません。

例)

(A) 数字連番 (**NUM**) を選んだ場合

**NUM** コマンドは、数字データ (**0~9**) を連番にします。

**EXCP** コマンドで設定された数字データを飛ばしてインクリメント・デクリメントを行います。

また、**EXCP** コマンドで削除文字に数字データ以外が設定されていてもそのデータは無視されます。

**EXCP** を使わない例  
**BCLC;1;IDF;1;NUM;**  
**"000005";SAL;3;**

**EXCP** を使った例  
**BCLC;1;IDF;1;EXCP;6;NUM;**  
**"000005";SAL;3;**

結果（ラベル 3 枚印刷）

**000005**  
**000006**  
**000007**

**000005**      **6** は削除文字であるから  
**000007** ← 左のようにインクリメントされま  
**000008**      す。

(B) 英字連番 (**ALPH**) を選んだ場合

**ALPH** コマンドは、大文字の **A~Z** までを連番にします。

**EXCP** コマンドで設定された英字データを飛ばしてインクリメント・デクリメントを行います。

また、**EXCP** コマンドで削除文字に英字データ以外が設定されていてもそのデータは無視されます。

## NITTO DENKO

**EXCP を使わない例**  
**BCLC;1;IDF;1;ALPH;**  
**“AAAAAA”;SAL;2;**

**EXCP を使った例**  
**BCLC;1;IDF;1;EXCP;B;ALPH;**  
**“AAAAAA”;SAL;2;**

結果（ラベル 3 枚印刷）

**AAAAAA**  
**AAAAAB**  
**AAAAAC**

**AAAAAA** B は削除文字であるから  
**AAAAAC** ← 左のようにインクリメントされま  
**AAAAAD** す。

(C) 英数字連番 (BOTH) を選んだ場合

**BOTH** コマンドは、数字の **0~9** と大文字の **A~Z** までの両方を連番にします。  
**EXCP** コマンドで設定された英数字データを飛ばしてインクリメント・デクリメントを行います。

**BOTH** ではデータは以下の順番に文字が変わります。

**EXCP** コマンドで **I** と **O** が設定された場合 (IDF が 1)

**0123456789ABCDEFGHIJKLMNPQRSTUVWXYZ0123**  
のように **I** と **O** が抜けます。

**EXCP を使わない例 (IDF が 3)**  
**BCLC;1;IDF;3;BOTH;**  
**“77777C”;SAL;1;**

**EXCP を使った例 (IDF が 3)**  
**BCLC;1;IDF;3;EXCP;IO;BOTH;**  
**“77777C”;SAL;1;**

結果（ラベル 3 枚印刷）

**77777C**  
**77777F**  
**77777I**  
**77777L**  
**77777O**

**77777C**  
**77777F**  
**77777J**  
**77777M**  
**77777Q**

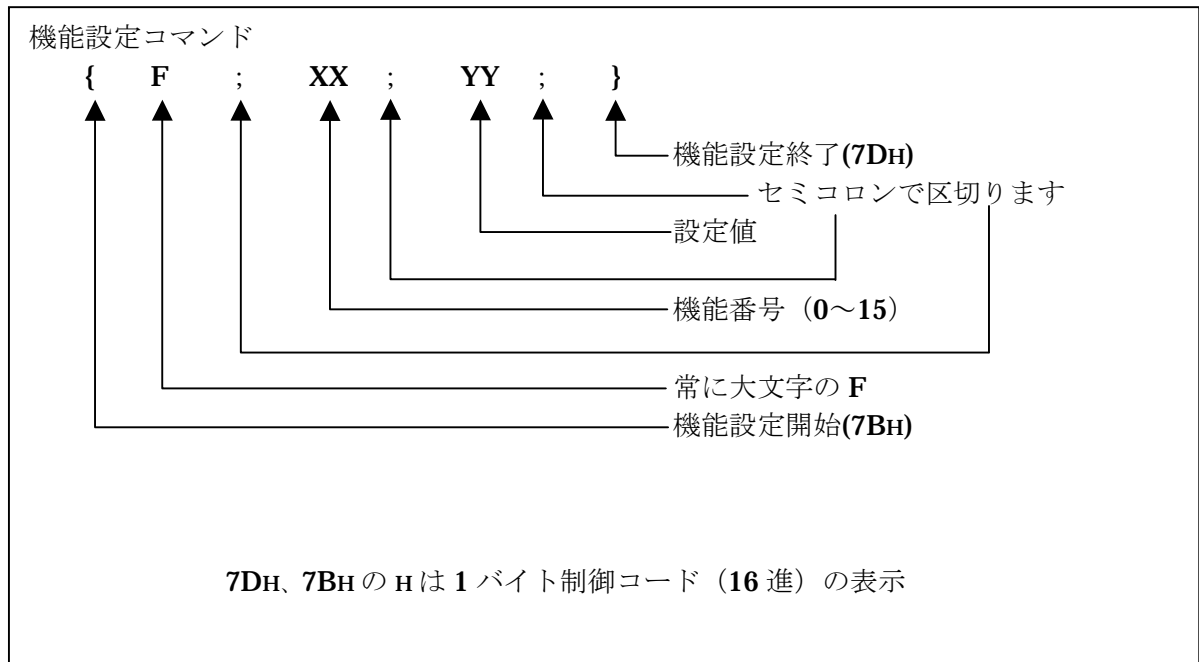
8 9 A B C D E F G H J K L M N P Q R S T U... \*  
                  ①   ②   ③   ④   ⑤

削除文字については、増減の文字データとしてカウントしません。  
つまり、上図にあるように始めから指定文字を削除した状態での増減用文字順番として動作します。**NUM**、**ALPH** も同様です。

## 12. プリンタ制御

### 12.1. 機能設定コマンド

コンピュータからコマンドを送信し、機能設定値の変更ができます（このコマンドを機能設定コマンドと呼びます）。



#### 注

- ・ 機能番号と設定値はプリンタの機能設定表を参照下さい。
- ・ 機能設定コマンドは **RCL** コマンドヘッダの前に置いてください。
- ・ 機能設定コマンド内にコメントの記述はできません

### 12.2. フォームの長さ (ラベルピッチ)

ヘッダ (2.6 参照) 内にはプリント・バッファの長さを指定しなければなりません。また、ラベル自体の物理的長さをラベル長設定 (**Use Top Of Form:UTOF**) コマンドによって指定しなければなりません。ここでは、このコマンドについて説明します。

## NITTO DENKO

### 12.2.1. 使用方法

コマンド・ヘッダ内の印字領域の長さが、フォームの長さと同じである場合には、**UTOF** コマンドを省略しても構いません。それらが異なる場合には、プリンタにフォームの長さと同様にラベル上のイメージの正しい位置を指示するために、**UTOF** コマンドが必要になります。

### 12.2.2. ラベル長設定 (Use Top Of Form : UTOF)

**UTOF** コマンドは、ラベルにラベル間のスペースを加えた物理的な長さ、つまり 1 つのラベルの先端から次のラベルの先端まで (ラベルピッチ) を設定します。この構文は次の通りになります。

```
UTOF;1968;
```

**UTOF** は、**SPB** コマンドの直後に指定しなければなりません。次の例では、長さ **47.1mm** のラベル上に **Code39** バーコードを印刷します。この長さのうち、印刷するために実際に使われるのはヘッダ部印字領域の長さ **180** ピクセル (**45.72mm**) 分です。

```
~ ^ "IL10-3";1;0;180;0;  
SPB;UTOF;1855;VBR;50;HBR;50;  
BDEF;1;BCSH;50;BNEW;3;BWEW;8;  
BCST;"*100100*";BSTP;EOL;  
VPR;20;HPR;110;DHR;$8002;DCH;15;DCW;30;ICS;2;  
"*100100*";  
TRM;¥
```



## 12.3. 上下反転印刷

### 12.3.1. 使用方法

プリンタは、印刷イメージの下側から上側に向かって印刷しますので、オペレータは正しい方向でラベルを見ることができます。しかし、上側から逆さまに印刷を始めたい場合があります。**FLIP** コマンドは、プリンタ内においてイメージを **180** 度逆転させます。

## NITTO DENKO

### 12.3.2. フリップ (FLIP)

**FLIP** コマンドは、印刷前に印刷イメージを **180** 度回転させるようにプリンタに命令を与えます。このコマンドの構文は次の通りです。

**FLIP;XX;**

**XX**=ラベルの幅。水平ピクセル (**0.127mm**) 単位。

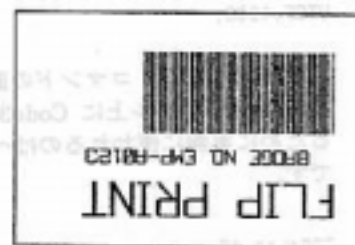
この引き数に拘らず常にすべてが上下反転されます。

### 12.3.3. 位置

**FLIP** は、ラベルを上下逆さにして前方に紙送りしながら印刷する効果があります。最初にラベルの上端が印刷され、ラベルの底部は最後に印刷されます。またラベルの右側が左側に、左は右に印刷されます。ラベル上のオブジェクト (印刷されるバーコードや文字など) の相対的な位置は変化しません。

**FLIP** コマンドを使った印刷例を次に示します。

```
~ ^ "IL10-4";1;0;150;0;  
SPB;FLIP;420;HLT;1;VLT;2;VBR;0;HBR;0;  
DBOX;0;0;420;145;  
VBR;40;HBR;35;  
DHR;$8000;DCH;30;DCW;60;ICS;4;  
"FLIP PRINT";EOL;  
VPR;20;HPR;30;DCH;12;DCW;28;ICS;2;"BADGE NO. EMP-A0123";EOL;  
VPR;60;BDEF;1;BCSH;50;BNEW;2;BWEW;6;  
BCST;"*EMP-A0123*";BSTP;EOL;  
TRM;¥
```



## 12.4. 遠隔操作

プリンタはホスト・コンピュータからの命令によって、プリンタの初期化と一時停止／一時停止の解除や、プリンタとホストがシリアルインターフェースで接続されている場合、プリンタからステータスデータを返させるためにステータス要求コマンドを使うことができます。

遠隔操作のコマンドは、これまでに説明したコマンドと異なって、プリンタが受け取るのと同時に実行されます。一方、通常のコマンドは、受信バッファに蓄えられ、入ってきた順に処理されます。もう **1** つの違いは、通常のコマンドは英数文字記号列であり、遠隔操作コマンドは **1** バイト制御文字です。ここでは遠隔操作コマンド群とその使い方について説明します。

### 注

これらのコマンドは、受信バッファが一杯の場合には、受信されません。

## NITTO DENKO

### 12.4.1. プリンタの初期化 (CAN : 18H)

プリンタ初期化 (CAN:18H) は、すべてのコマンドのパラメータを既定値に戻し、コマンド・エラーのようなエラー状態をリセットするのに使います。プリンタが **CAN** コマンドを受信すると、すべてのプリンタ制御パラメータがクリアされ、システムを電源が入れられた後と同じ状態になります。

### 12.4.2. 一時停止オン (DC2 : 12H)

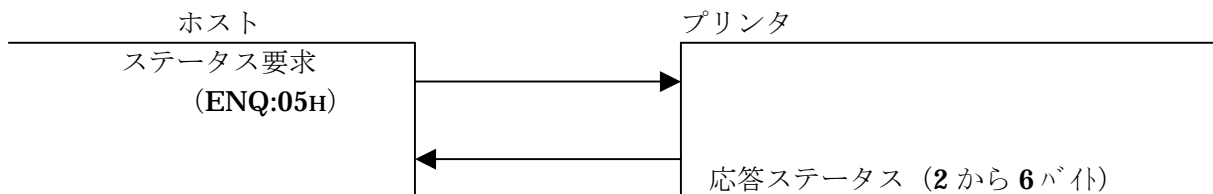
プリンタが印刷可能 (レディー) 状態の時に **DC2** コマンドを受け取ると、一時停止状態になります。これはプリンタの前面パネル上の一時停止スイッチを押すことと同じです。印刷可能 (レディー) 状態に戻すには、一時停止スイッチを押すか、あるいは一時停止オフ・コマンドを送信します。

### 12.4.3. 一時停止オフ (DC4 : 14H)

プリンタが一時停止状態の時に **DC4** コマンドを受け取ると、印刷可能 (レディー) 状態に復帰します。これはプリンタ前面パネル上の一時停止スイッチを押すことと同じです。一時停止がプリンタ・エラーによって引き起こされた場合には、**DC4** コマンドは無視されますので注意してください。

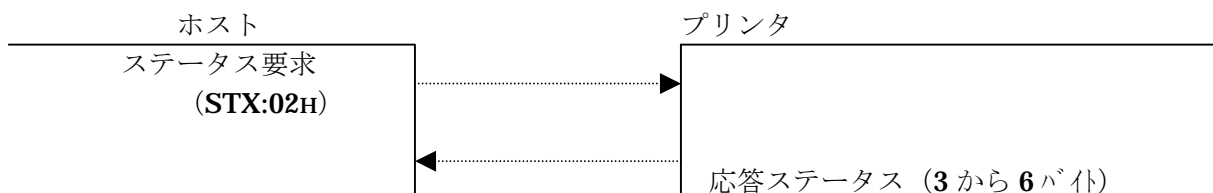
### 12.4.4. ステータス要求 (ENQ : 05H)

プリンタとホストが、シリアル・インターフェースで接続されていて **ENQ** を受信したら、その時点におけるプリンタのステータスをホストに送ります。シリアル・インターフェースは、二重通信 (ホストからプリンタへ、プリンタからホストへ) をサポートしますが、これに対し、セントロニクス・パラレル・インターフェースは、ホストからプリンタへのみ通信できます。したがって、セントロニクス・インターフェースでは、**ENQ** コマンドは無視されます。応答ステータスは、**24** 章のステータス一覧表をご覧ください。



### 12.4.5. ステータス要求 (STX : 02H)

プリンタとホストが、シリアル・インターフェースで接続されていて **STX** を受信したら、その時点におけるプリンタのステータスをホストに送ります。応答ステータスは、**24** 章のステータス一覧表をご覧ください。



## NITTO DENKO

### 12.4.6. プリンタ情報要求コマンド (INFOmation : INFO)

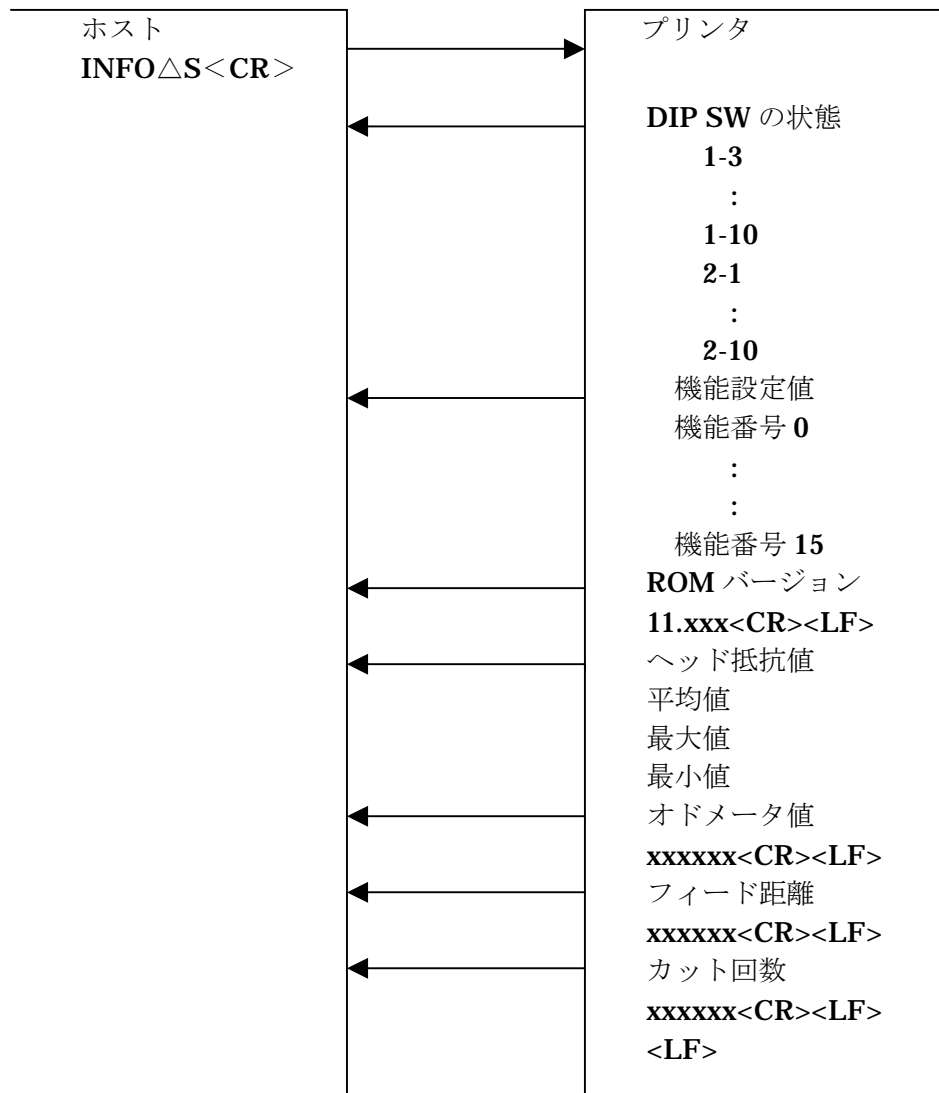
このコマンドは、プリンタの機能設定値、ROM バージョン等を問い合わせる為に使用します。

ホストから **INFO** コマンドが送られるとプリンタは、下図のように **DIP SW** の状態、機能設定値、**ROM** バージョン、ヘッド抵抗値、オドメータ値、フィード距離値、カット回数を送信します。

**INFO** コマンドのフォーマットとその応答は次の通りです。

**INFO△S<CR>**

注：△はスペースです。



各情報は、**CR** (0DH)、**LF** (0AH) で分離され、一番最後の情報のカット回数の後には、**CR1** つと **LF2** つが送られます。

各情報の内容は、以下の形式で送信されます。

**DIP SW** 情報

**ON** の場合 : **ON△△<CR><LF>**

**OFF** の場合 : **OFF△<CR><LF>**

## NITTO DENKO

機能設定値

0～9 の場合 : 03<CR><LF>  
10 以上の場合 : 13<CR><LF>  
マイナスの場合 : -10<CR><LF>

ROM バージョン

11.xxx<CR><LF>  
xxx が ROM のバージョンになります。

ヘッド抵抗値 (ohm)

xxx<CR><LF>

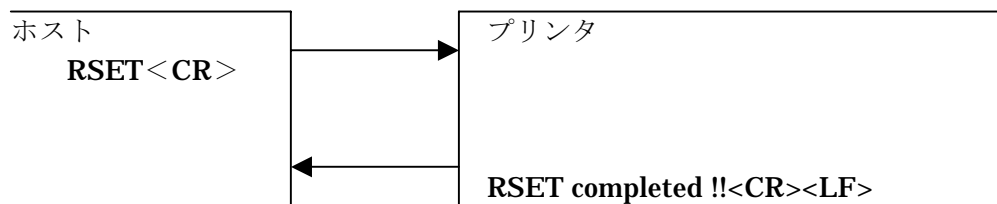
カット回数 (回)

xxx<CR><LF>

### 12.4.7. フィード距離リセットコマンド (ReSET command : RSET)

このコマンドは、プリンタのバックアップメモリに格納されているフィード距離をリセット (フィード距離を **0** クリア) するために使用するコマンドです。ホストから **RSET** コマンドが送られるとプリンタは、フィード距離のクリアを実行しクリアが終了するとメッセージを返します。

**RSET** コマンドとその応答は下図のようになります。





## 13. ロゴ LOGO

線や枠コマンドで生成できない特殊なオブジェクトやイメージは、**LOGO** コマンドで作成できます。この章では、**LOGO** コマンドについて説明し、**RCL** で作成されたロゴの例を示します。

### 13.1. ロゴ (LOGO)

**LOGO;B;R;DATA;...;DATA;**

**B**=1 列のデータの中のバイト数。

**R**=データの列の数。

**DATA**=10 進数あるいは 16 進数で表したデータ (バイト単位)

**LOGO** コマンドのデータバイト数は、**B**×**R** に等しくなければなりません。例えば、次のようなコマンドを考えてみます。

**LOGO;15;10;**

15 バイト×10 列でデータは合計 150 バイトです。したがって、**LOGO** コマンドのデータ部分は、150 バイトの情報がなければなりません。

データは、通常、バイトごとに 10 進数で指定しますが、各データ・バイトの前にドル記号 (\$) をおいて 16 進数で指定しても構いません。また、1 つのロゴのデータに 10 進数のデータと 16 進数のデータを混合することも可能です。

#### 注

ビットマップ・データは、論理的ピクセルではなく、物理的ピクセルで印字されます。物理的ピクセルの大きさは、ヘッド密度に依存します。

#### 注

1 列のデータの中のバイト数は、印綫幅を超える値を使用しないで下さい。

**LOGO** コマンド使用時フォーマットプログラムの容量が大きくなるような場合は、**BRK** コマンドを使用してください。(3.1.4 参照)

### 13.2. ロゴの展開方法

**RCL Ver.4.00** より以前のモデルでは、ロゴデータのように大量にデータをプリンタに送信する場合、プリンタ内部の、制御用バッファのサイズの関係から **BRK;RSPB** を使いいくつかのフォーマットデータに分割して、データを送信する必要がありましたが、**RCL Ver.4.00** 以降では、ロゴのデータは、制御用バッファにデータを格納してからの描画ではなく直接イメージバッファに描画するように変更されています。これにより、従来はいくつかのフォーマットファイルに分割して入力していたデータも、**BRK;RSPB** を使用して分割することなく、1 つのフォーマットとして入力が可能に

## NITTO DENKO

なっています。この機能に関しては、設定コマンドは存在しません。  
(プリンタ内部の処理が変更されただけです)

ロゴは他のオブジェクトと同様に絶対位置移動と相対位置移動コマンドでラベル上の位置を決定します。プリンタは通常バイトの左端でロゴを描き始めることを忘れてはなりません。ロゴを印刷する前に、カーソルが **1** バイトのデータの間にある場合、プリンタはそのバイトの左隅からロゴを描き始めます。線や枠と同じように、ロゴはカーソルの位置を変更しません。

### 13.3. データを指定したロゴコマンド

#### 13.3.1. ロゴデータフォーマット指定つきコマンド (LOGo Data format : LOGD)

ロゴコマンドとして描画データ部の表記方法を選択して送信できるコマンドです。新たなフォーマットを選択することにより、従来のロゴデータの約 **1/2** で入力が可能になります。

このコマンドの構文は、以下のようになります。

**LOGD;F;C;R;Data...Data;**

**F**=ロゴデータのフォーマットを設定する。

**0** : **10** 進、**16** 進識別タイプ (従来のデータ形式)

**1** : **16** 進キャラクタタイプ (データ短縮形式)

(短縮形式データは、**1** バイトのバイナリーデータを **2** バイトの文字変数で表します。)

例 : バイナリ **1110 0011** (**E3hex**) は、“**E3**” で表す。

**C**=**1** 行のバイト数

**R**=ロゴのライン数

**Data**=**10** 進数あるいは **16** 進数で表したデータ (バイト単位)

ロゴコマンドのデータのバイト数は、**C×R** に等しくなければなりません。

フォーマット : **0** の場合

通常は、**1** バイトデータを **10** 進数で表現します。**1** バイト後のデータは ; (セミコロン) で区切られます。またデータの前に \$ を付けることにより **16** 進数で表現することができます。また、ロゴのデータ内で **2** つの表現を混在することも可能です。

フォーマット : **1** の場合

データの表現は、全て **16** 進数表記になり、**1** バイトごとの区切りの ;、**16** 進を意味する \$ は必要ありません。しかしデータの最後に終了の ; をつけて下さい。

#### 注

ロゴデータフォーマットの **1** : **16** 進数キャラクタタイプで作成されたデータを拡張メモリのロゴの登録フォーマットとしてそのまま使用できません。拡張メモリへ登録する場合は、**LOGO** コマンドによりデータを作成してください。

## NITTO DENKO

### 13.4. ロゴの例

2つの特殊文字、登録商標記号と著作権記号を印刷する **LOGO** の使用例を示します。

~ ^ "REGISTER";1;0;150;0;

SPB;

VBR;5;HBR;120;

LOGO;4;19;

0;\$0F;\$F8;0;

0;\$70;\$07;0;

1;\$C0;1;\$C0;

7;0;0;\$70;

\$0E;\$0F;\$F0;\$38;

\$1C;\$0F;\$F8;\$1C;

\$38;\$0C;\$18;\$0E;

\$30;\$0C;\$18;6;

\$70;\$0C;\$38;7;

\$70;\$0F;\$F0;7;

\$70;\$0F;\$F0;7;

\$30;\$0C;\$38;6;

\$38;\$0C;\$18;\$0E;

\$1C;\$0C;\$18;\$1C;

\$0E;\$0C;\$18;\$38;

\$7;0;0;\$70;

1;\$C0;1;\$C0;

0;\$70;\$07;0;

0;\$0F;\$F8;0;

BRK;¥

~ ^ "COPYRIGHT";1;0;50;0;

RSPB;

VBR;5;HBR;200;

LOGO;4;20;

1;\$FF;\$FF;\$C0;

3;\$FF;\$FF;\$E0;

7;0;0;\$70;

\$0E;0;0;\$38;

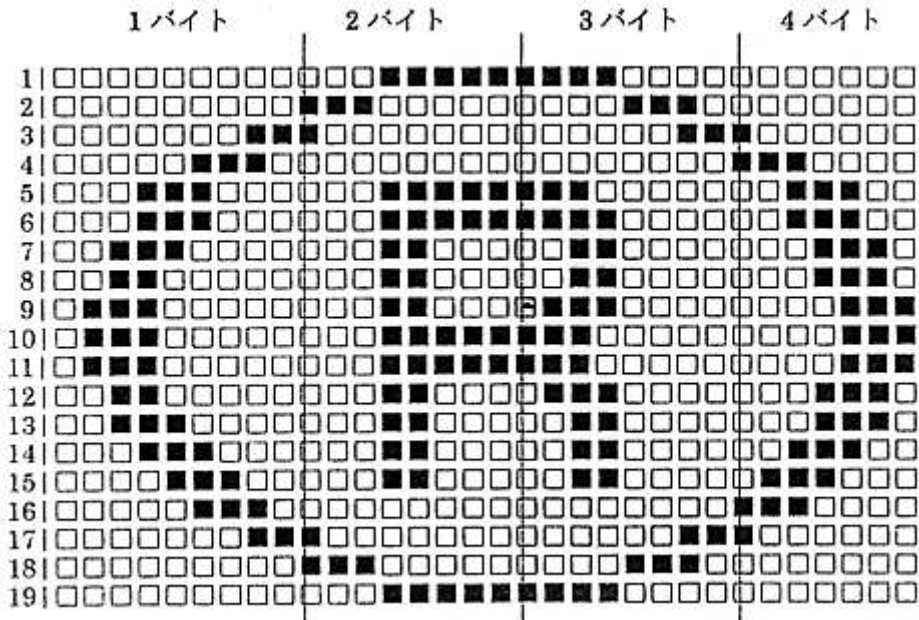
\$1C;\$0F;\$F8;\$1C;

NITTO DENKO

\$1C;\$1F;\$FC;\$1C;  
 \$1C;\$38;\$0E;\$1C;  
 \$1C;\$70;\$07;\$1C;  
 \$1C;\$70;0;\$1C;  
 \$1C;\$70;0;\$1C;  
 \$1C;\$70;0;\$1C;  
 \$1C;\$70;0;\$1C;  
 \$1C;\$70;7;\$1C;  
 \$1C;\$38;\$0E;\$1C;  
 \$1C;\$1F;\$FC;\$1C;  
 \$1C;\$0F;\$F8;\$1C;  
 \$0E;0;0;\$38;  
 7;0;0;\$70;  
 3;\$FF;\$FF;\$E0;  
 1;\$FF;\$FF;\$C0;  
 TRM;¥

注

- この例では 1 ドットで印字している部分がありますが、実際は 2 ドット以上で作成してください。
- 黒の部分が大きい場合、印字品質が低下する場合があります。



# NITTO DENKO

1      
2      
3      
4      
5      
6      
7      
8      
9

A      
B      
C      
D      
E      
F

NITTO DENKO

~ ^ "LOGO";5;0;39;170;SPB;UTOF;0512;

HBR;160;VBR; 20;

LOGO;12;87;

\$00;\$00;\$00;\$00;\$00;\$1f;\$c0;\$00;\$00;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$03;\$ff;\$c0;\$00;\$00;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$1f;\$ff;\$c0;\$00;\$00;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$7f;\$ff;\$c0;\$00;\$00;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$03;\$ff;\$ff;\$c0;\$00;\$18;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$0f;\$ff;\$e0;\$00;\$00;\$1e;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$1f;\$fe;\$00;\$00;\$00;\$1f;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$7f;\$f0;\$00;\$00;\$00;\$1f;\$c0;\$00;\$00;\$00;  
\$00;\$00;\$00;\$ff;\$80;\$00;\$00;\$00;\$1f;\$e0;\$00;\$00;\$00;  
\$00;\$00;\$01;\$fe;\$00;\$00;\$00;\$00;\$1f;\$f0;\$00;\$00;\$00;  
\$00;\$00;\$07;\$fc;\$00;\$00;\$00;\$00;\$1f;\$fc;\$00;\$00;\$00;\$00;  
\$00;\$00;\$0f;\$f0;\$00;\$00;\$00;\$00;\$1f;\$fe;\$00;\$00;\$00;\$00;  
\$00;\$00;\$1f;\$e0;\$00;\$00;\$c0;\$00;\$1f;\$ff;\$00;\$00;\$00;\$00;  
\$00;\$00;\$3f;\$80;\$00;\$00;\$c0;\$00;\$1f;\$3f;\$80;\$00;\$00;\$00;  
\$00;\$00;\$7f;\$00;\$00;\$00;\$00;\$00;\$1f;\$1f;\$c0;\$00;\$00;\$00;  
\$00;\$00;\$fe;\$00;\$00;\$00;\$00;\$00;\$1f;\$0f;\$e0;\$00;\$00;\$00;  
\$00;\$00;\$fc;\$00;\$00;\$00;\$00;\$00;\$1f;\$07;\$e0;\$00;\$00;\$00;  
\$00;\$01;\$f8;\$00;\$00;\$00;\$00;\$00;\$1f;\$03;\$f0;\$00;\$00;\$00;  
\$00;\$03;\$f0;\$00;\$00;\$00;\$00;\$00;\$1f;\$01;\$f8;\$00;\$00;\$00;  
\$00;\$07;\$e0;\$00;\$00;\$00;\$00;\$00;\$1f;\$00;\$fc;\$00;\$00;\$00;  
\$00;\$07;\$e0;\$00;\$00;\$00;\$00;\$00;\$1f;\$00;\$fc;\$00;\$00;\$00;  
\$00;\$0f;\$c0;\$00;\$00;\$00;\$00;\$00;\$1f;\$00;\$7e;\$00;\$00;\$00;  
\$00;\$1f;\$80;\$00;\$00;\$01;\$e0;\$00;\$1f;\$00;\$3f;\$00;\$00;\$00;  
\$00;\$1f;\$80;\$00;\$00;\$01;\$e0;\$00;\$1f;\$00;\$3f;\$00;\$00;\$00;  
\$00;\$3f;\$00;\$00;\$00;\$01;\$e0;\$00;\$1f;\$00;\$1f;\$80;\$00;\$00;  
\$00;\$3e;\$00;\$00;\$00;\$01;\$e0;\$00;\$1f;\$00;\$0f;\$80;\$00;\$00;  
\$00;\$3e;\$00;\$00;\$00;\$01;\$f0;\$00;\$3f;\$00;\$0f;\$80;\$00;\$00;  
\$00;\$7c;\$00;\$00;\$00;\$01;\$f0;\$00;\$3e;\$00;\$07;\$c0;\$00;\$00;  
\$00;\$7c;\$00;\$00;\$00;\$01;\$f8;\$00;\$7e;\$00;\$07;\$c0;\$00;\$00;  
\$00;\$fc;\$00;\$00;\$00;\$01;\$fc;\$00;\$fc;\$00;\$07;\$e0;\$00;\$00;  
\$00;\$f8;\$00;\$00;\$00;\$01;\$ff;\$03;\$f8;\$00;\$03;\$e0;\$00;\$00;  
\$00;\$f8;\$00;\$00;\$00;\$01;\$ff;\$ff;\$f0;\$00;\$03;\$e0;\$01;\$f8;\$00;\$00;  
\$00;\$01;\$ff;\$ff;\$e0;\$00;\$03;\$f0;\$01;\$f0;\$00;\$00;\$00;\$01;\$ef;\$ff;\$c0;\$00;\$01;\$f0;

NITTO DENKO

\$01;\$f0;\$00;\$00;\$00;\$01;\$e3;\$ff;\$00;\$00;\$01;\$f0;  
\$01;\$f0;\$00;\$00;\$00;\$01;\$e0;\$00;\$00;\$00;\$01;\$f0;  
\$01;\$f0;\$00;\$00;\$00;\$01;\$e0;\$00;\$00;\$00;\$01;\$f0;  
\$03;\$e0;\$00;\$00;\$00;\$01;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$00;\$01;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$00;\$01;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$00;\$01;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$0f;\$f1;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$3f;\$fd;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$7f;\$ff;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$00;\$ff;\$ff;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$01;\$fc;\$3f;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$01;\$f0;\$0f;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$03;\$e0;\$07;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$03;\$c0;\$03;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$03;\$e0;\$00;\$07;\$c0;\$03;\$e0;\$00;\$00;\$00;\$00;\$f8;  
\$01;\$f0;\$00;\$07;\$80;\$01;\$e0;\$00;\$00;\$00;\$01;\$f0;  
\$01;\$f0;\$00;\$07;\$80;\$01;\$e0;\$00;\$00;\$00;\$01;\$f0;  
\$01;\$f0;\$00;\$07;\$c0;\$01;\$e0;\$00;\$00;\$00;\$01;\$f0;  
\$01;\$f0;\$00;\$07;\$c0;\$01;\$e0;\$00;\$00;\$00;\$01;\$f0;  
\$01;\$f8;\$00;\$03;\$e0;\$01;\$e0;\$00;\$00;\$00;\$03;\$f0;  
\$00;\$f8;\$00;\$03;\$e0;\$01;\$e0;\$00;\$00;\$00;\$03;\$e0;  
\$00;\$f8;\$00;\$01;\$f0;\$01;\$e0;\$00;\$00;\$00;\$03;\$e0;  
\$00;\$fc;\$00;\$01;\$f8;\$00;\$00;\$00;\$00;\$00;\$07;\$e0;  
\$00;\$7c;\$00;\$00;\$fc;\$00;\$00;\$00;\$00;\$00;\$07;\$c0;  
\$00;\$7c;\$00;\$00;\$7e;\$00;\$00;\$00;\$00;\$00;\$07;\$c0;  
\$00;\$3e;\$00;\$00;\$3f;\$00;\$00;\$00;\$00;\$00;\$0f;\$80;  
\$00;\$3e;\$00;\$00;\$1f;\$80;\$00;\$00;\$00;\$00;\$0f;\$80;  
\$00;\$3f;\$00;\$00;\$0f;\$80;\$00;\$00;\$00;\$00;\$1f;\$80;  
\$00;\$1f;\$80;\$00;\$07;\$c0;\$00;\$00;\$00;\$00;\$3f;\$00;  
\$00;\$1f;\$80;\$00;\$07;\$c0;\$00;\$00;\$00;\$00;\$3f;\$00;  
\$00;\$0f;\$c0;\$00;\$03;\$e0;\$00;\$00;\$00;\$00;\$7e;\$00;  
\$00;\$07;\$e0;\$00;\$03;\$e0;\$00;\$00;\$00;\$00;\$fc;\$00;  
\$00;\$07;\$e0;\$00;\$01;\$e0;\$00;\$00;\$00;\$00;\$fc;\$00;  
\$00;\$03;\$f0;\$00;\$01;\$e0;\$00;\$00;\$00;\$01;\$f8;\$00;  
\$00;\$01;\$f8;\$00;\$01;\$e0;\$00;\$00;\$00;\$03;\$f0;\$00;  
\$00;\$00;\$fc;\$00;\$03;\$e0;\$00;\$00;\$00;\$07;\$e0;\$00;  
\$00;\$00;\$fe;\$00;\$03;\$c0;\$00;\$00;\$00;\$0f;\$e0;\$00;

## NITTO DENKO

\$00;\$00;\$7f;\$00;\$07;\$c0;\$00;\$00;\$00;\$1f;\$c0;\$00;  
\$00;\$00;\$3f;\$80;\$0f;\$c0;\$00;\$00;\$00;\$3f;\$80;\$00;  
\$00;\$00;\$1f;\$c0;\$3f;\$80;\$00;\$00;\$00;\$ff;\$00;\$00;  
\$00;\$00;\$0f;\$f0;\$ff;\$80;\$00;\$00;\$01;\$fe;\$00;\$00;  
\$00;\$00;\$07;\$ff;\$ff;\$00;\$00;\$00;\$07;\$fc;\$00;\$00;  
\$00;\$00;\$03;\$ff;\$fe;\$00;\$00;\$00;\$0f;\$f0;\$00;\$00;  
\$00;\$00;\$01;\$ff;\$fc;\$00;\$00;\$00;\$3f;\$e0;\$00;\$00;  
\$00;\$00;\$00;\$7f;\$f0;\$00;\$00;\$01;\$ff;\$c0;\$00;\$00;  
\$00;\$00;\$00;\$1f;\$80;\$00;\$00;\$0f;\$ff;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$00;\$00;\$00;\$ff;\$fe;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$00;\$00;\$ff;\$ff;\$f8;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$00;\$00;\$ff;\$ff;\$c0;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$00;\$00;\$ff;\$ff;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$00;\$00;\$ff;\$f8;\$00;\$00;\$00;\$00;  
\$00;\$00;\$00;\$00;\$00;\$00;\$ff;\$00;\$00;\$00;\$00;\$00;  
TRM;¥



次に上記ロゴをデータ短縮形で設定した場合のコマンド例を以下に示します。

```
~ ^"LOGO";5;0;39;170;SPB;UTOF;0512;HBR;160;VBR; 20;  
LOGO;1;12;87;  
0000000001fc0000000000000000000003ffc0000000000  
00000001fffc00000000000000000007fffc0000000000  
0000003ffffc00018000000000000ffe00001e00000  
000001ffe000001f0000000000007ff0000001fc0000  
00000f80000001fe0000000001fe00000001ff0000  
00007fc00000001ffc00000000ff000000001ffe0000  
0001fe000000001fff00000003f8000000001f3f8000  
0007f000000001f1fc000000fe0000000001f0fe000  
000fc0000000001f07e000001f80000000001f03f000  
0003f0000000001f01f800007e00000000001f00fc00  
0007e00000000001f00fc00000fc00000000001f007e00  
001f8000001e0001f003f00001f8000001e0001f003f00  
003f0000001e0001f001f80003e0000001e0001f00f80  
003e0000001f0003f000f80007c00000001f0003e0007c0  
007c0000001f8007e0007c000fc0000001fc00fc0007e0  
00f80000001ff03f80003e000f80000001ffff00003e0  
01f80000001fff00003f001f00000001efffc00001f0  
01f00000001e3ff000001f001f00000001e00000001f0
```



**NITTO DENKO**

01f000000001e000000001f003e000000001e000000000f8  
03e000000001e000000000f803e000000001e000000000f8  
03e000000001e000000000f803e000000ff1e000000000f8  
03e000003ffde000000000f803e000007ffe000000000f8  
03e00000fffe000000000f803e00001fc3fe000000000f8  
03e00001f00fe000000000f803e00003e007e000000000f8  
03e00003c003e000000000f803e00007c003e000000000f8  
01f000078001e000000001f001f000078001e000000001f0  
01f00007c001e000000001f001f00007c001e000000001f0  
01f80003e001e000000003f000f80003e001e000000003e0  
00f80001f001e000000003e000fc0001f800000000007e0  
007c0000fc00000000007c0007c00007e00000000007c0  
003e00003f00000000000f80003e00001f80000000000f80  
003f00000f80000000001f80001f800007c0000000003f00  
001f800007c0000000003f00000fc00003e0000000007e00  
0007e00003e000000000fc000007e00001e000000000fc00  
0003f00001e000000001f8000001f80001e000000003f000  
0000fc0003e000000007e0000000fe0003c00000000fe000  
00007f0007c00000001fc00000003f800fc00000003f8000  
00001fc03f80000000ff00000000ff0ff80000001fe0000  
000007ffff00000007fc0000000003fffe000000ff00000  
000001fffc0000003fe000000000007ff0000001ffc00000  
0000001f8000000ff000000000000000000fffe000000  
000000000000ffff800000000000000000000fffc0000000  
000000000000ffff000000000000000000000ff800000000  
000000000000ff0000000000;

TRM;¥

## 14. 拡張メモリ・オペレーション

プリンタは、すでに説明した **RCL** を使って個々のラベルフォーマットを指定するコマンドモードに加えて、ラベル・データやロゴデータを前もって拡張メモリに書き込みできるもう **1** つのモード (拡張メモリモード) があります。拡張メモリモードでは、フォーマット内のデータ・フィールドのデータを指定するだけで印刷することができます。この章では、拡張メモリモードで操作するのに必要なコマンドについて説明します。

### 14.1. 拡張メモリへのフォーマット書き込み

拡張メモリへ書き込むラベルフォーマットは、**RCL** のコマンドで作成します。プリンタは、ラベルフォーマットを直接拡張メモリへ書き込むことができます。プリンタでラベルフォーマットデータを拡張メモリへ書き込む手順は次の通りです。

1. 拡張メモリに保存されるラベルフォーマットデータは、**RCL** を使って作成します。印刷時にデータが決定されるテキストとバーコード・フィールド (可変フィールド) のあるラベルには、従来の"...”の代わりに **FLD** コマンドを使用します。
2. ダウンロード開始コマンド (**FS:1Ch**) を送信します。プリンタが **FS** コマンドを受取った後、ダウンロード終了 (**EM:19H**) コマンドを受け取るまで、全てのコマンド列が拡張メモリに書き込まれます。
3. ステップ **1** で作成したコマンド列をホストから送信してください。1 枚の拡張メモリの中に、複数のラベルフォーマットが保存できます。複数のラベルフォーマットをカードに書き込む場合には、(**FS:1Ch**) (フォーマット) (**EM:19H**) を **1** つずつダウンロードしてください。書き込み中に拡張メモリ容量を越える場合には、プリンタのエラー **LED** が点灯して拡張メモリのオーバーフローを示します。

#### 注

ダウンロード開始コマンド、終了コマンドは **1** バイトの制御コードで、開始が **1Ch** (**16** 進表示)、終了が **19H** (**16** 進表示) を送信してください。

つまり、書き込みのためホストから送られるコマンド列は次のような構成になります。

```
(1Ch)
  ~ ^ "file1.RCL";...;TRM;¥
(19H)
(1Ch)
  ~ ^ "file2.RCL";...;TRM;¥
(19H)
(1Ch)
  ~ ^ "file3.RCL";...;TRM;¥
(19H)
```

## 注

- (FS:1Ch) と (EM:19H) の間に必ず 1 フォーマットのみ挿入してください。
- ファイル名の拡張子 (RCL) は省略可能です。プリンタ内部で (RCL) をつけたファイル名で拡張メモリに書き込まれます。
- ファイル名は 8 文字以内としてください。

## 14.1.1. フィールド (FieLD : FLD)

**FLD** コマンドは、変数データ・フィールドを指定するために使います。**FLD** コマンドのフォーマットは次の通りです。

**FLD**;( +/-offset)<user prompt;NN>=field name

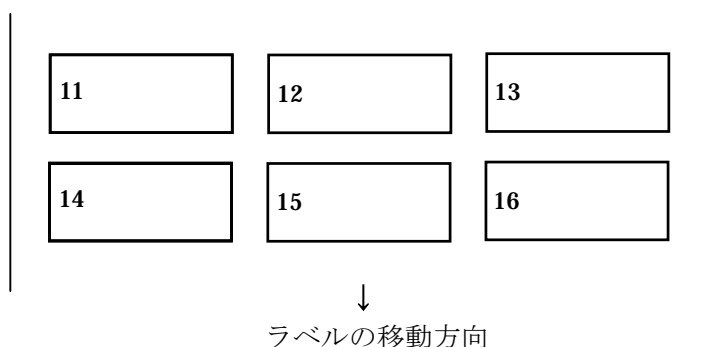
**offset** :0 から 32,767 までの整数

**user prompt** :LOAD コマンドに応答して送信される文字列

**NN** :フィールド内の最大文字数を指定する 2 桁の数

**field name** :FLD コマンドで参照されるフィールド名

オフセットは複数のラベルを横に並べて印刷するためのパラメータです。下の番号を参照してください。



横に 3 つのラベルを縦 2 列に、合計で 6 つの連番を含むラベルを印刷する場合、横に並んだラベルは 1 組のものとしてフォーマットしなければなりません。この場合、横に並んだ 3 つのラベルは、1 セットで取り扱われ、それぞれのラベルに対して +3 ずつインクリメントするコマンドを連続して使わなければなりません。印刷する連番フィールドが変数データ・フィールドである場合、最初に印刷される 3 つのラベルそれぞれの変数データフィールドに初期値を割り当てなければなりません。しかし、このような例では、2 番目と 3 番目の初期値は最初のフィールドの初期値で決まるので、ホストからは第 1 (基準) フィールドを送信する必要があり、残りはこの基準からのオフセットで決定できます。

上述の図において、ラベル番号 11 を基準とすると、12 と 13 のラベルのそれぞれの初期値は、**FLD** コマンドでオフセットで指定できます。すなわち、次のようになります。

**FLD**;( +1),**FLD**;( +2)

基準ラベルの **FLD** コマンド構文は、**FLD**;( +0) です。

## 14.2. 拡張メモリモードでの印刷

拡張メモリ・モードでは、前もって拡張メモリにラベルフォーマットを保存し、ホストから拡張メモリ内のフォーマット・ファイルを指定し、フォーマット内の変数フィールドの初期値を入力し、印刷するラベルの枚数を指示します。これらのコマンドが、拡張メモリ通信コマンドです。ここでは、**3**つの拡張メモリ通信コマンドと、拡張メモリに書き込まれた内容を調べるための**DIR**コマンド、**LOAD**コマンドについて説明します。

### 14.2.1. ファイル名指定コマンド (\*)

この拡張メモリ通信コマンドは、拡張メモリ上のファイルの中の印刷すべきファイルを指定します。コマンドフォーマットは単純で、アスタリスク記号 (\*) の後にファイル名を指定します。ファイル名の終わりには、復帰 (**0DH**) をつけなければなりません。

**\*filename<CR>**  
**(\*filename;RCL <CR>)**

電源を投入後と、ある印刷を終了した後でコマンド待ちの状態の時には、プリンタは**RCL**のヘッダー・コマンドまたはアスタリスク記号のいずれかを受け入れることができます。ヘッダー・コマンドを受信した場合、コマンドモードになり、アスタリスクを受け取った場合、拡張メモリ・モードになります。拡張メモリ通信コマンドと**RCL**のコマンドモードのコマンドは混在できませんので注意してください。

アスタリスク記号で指定したファイルが拡張メモリに存在しない場合、拡張メモリ・エラーになります。

### 14.2.2. 変数データ・フィールド初期値設定コマンド (/)

この拡張メモリ通信コマンドは、フォーマット内の各変数データ・フィールドの初期値を設定します。このコマンドの構文は次のとおりです。

**/fieldname=initial\_value<CR>**

**fieldname** は、拡張メモリにフォーマットデータを書き込んだときに、**FLD**コマンドで入力したフィールド名です。

**1**つのファイル内に複数の変数データ・フィールドがある場合、必要な回数だけ / コマンドを必要なだけ使うことができます。各行は復帰 (**0DH**) で終わらなければなりません。

**/fieldname1=initial\_value1<CR>**

**/fieldname2=initial\_value2<CR>**

**/fieldname3=initial\_value3<CR>**

## NITTO DENKO

### 14.2.3. バッチカット枚数 (%)

3 つ目の拡張メモリ通信コマンドは、バッチカット枚数を指定するコマンドです。これも 1 バイト文字です。行の終わりは復帰 (0DH) でなければなりません。

%N<CR>

N=1 から 999999 までの整数

%コマンドを受け取ると、プリンタは、バッチカット枚数を設定します。

### 14.2.4. 印刷枚数指定コマンド (!)

4 つ目の拡張メモリ通信コマンドは、印刷枚数指定コマンドです。これも 1 バイト文字です。行の終わりは復帰 (0DH) でなければなりません。

!N<CR>

N=1 から 999999 までの整数

!コマンドを受け取ると、プリンタは印刷を開始します。変数データ・フィールドのいずれかに初期値が割り当てられていない場合、フィールドはデータがないもの (Null) として扱います。

以下に FLD コマンドを用いたラベルフォーマット例を示します。

#

#### SAMPLE PROGRAM IC CARD

LABEL HIGHT	6mm	LENGTH	50mm
LABEL PITCH	9mm		

#;

#Start of Down load #;

# 1byte の制御コード(1CH) [プリントアウトでは見えません] #;

~^"ICCARD";1;0;24;187;SPB;UTOF;0354;

BSYM; 1; 1;BNEW; 3;BWEW; 7;BICG; 3;

DDF; 6; 10; DFM; 1; 1;DFS; 3;DFO; 1; 1;

MRK;

HBR; 0;VBR; 0;DWBX; 0; 0;394; 24;NUM;

BCLC; 1;BCID; 1;

IDF; 1;

## NITTO DENKO

```
HBR; 33;VBR; 14;BCSH; 18;  
BCST;  
  “*”;FLD;(0)<;11>=DATA;BSAL;2;”*”;  
BSTP;  
HBR; 66;VBR; 15;  
  FLD;(0)<;11>=DATA;SAL;2;  
RET;  
TRM;¥  
#End of Down load #;  
# 1byte の制御コード(19H) [プリントアウトでは見えません] #;
```

次にこのフォーマットにデータを与えた拡張メモリ通信コマンド例を示します。

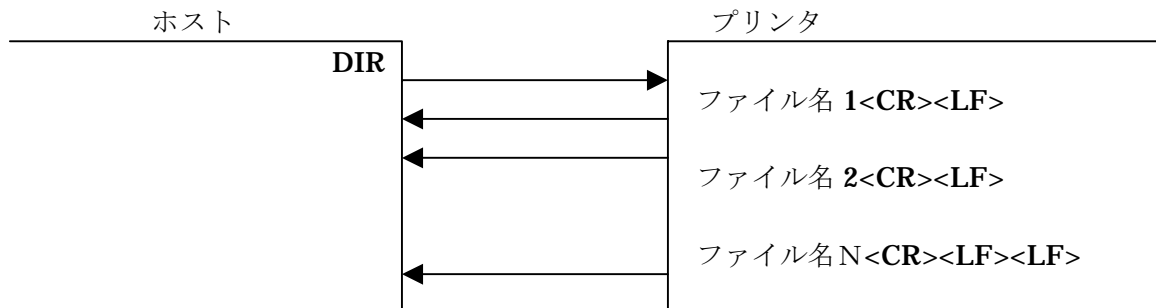
```
*ICCARD <CR>  
/DATA=ABCDEFGHII00<CR>  
!10<CR>
```

<p><b>注</b> 機能設定コマンドを拡張メモリに登録することはできません。</p>
--

## NITTO DENKO

### 14.2.5. ファイル名要求コマンド (DIR)

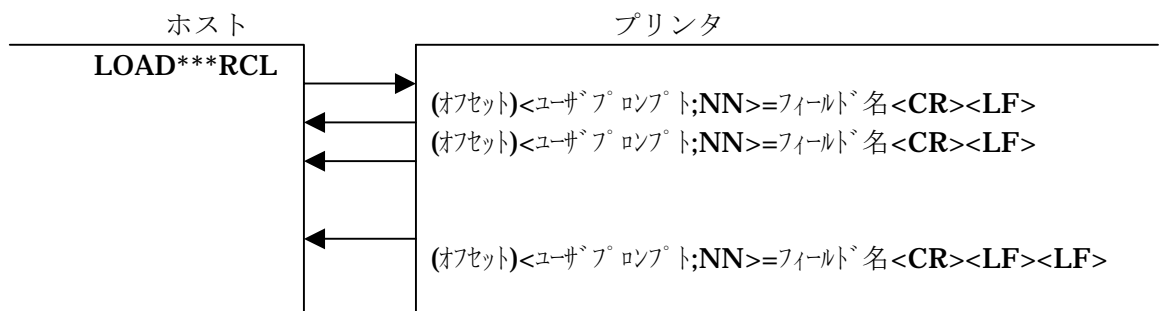
**DIR** コマンドは、拡張メモリに保存されているファイル名を問い合わせるために使用します。ホストから **DIR** コマンドを送ったとき、プリンタは拡張メモリ上に保存されているすべてのファイル名を返します。このコマンドは、二重通信が必要なのでシリアル・インターフェースでのみサポートされています。**DIR** コマンドとその応答は、下の図に示すように取り扱われます。



### 14.2.6. フィールド名要求コマンド (LOAD)

**DIR** コマンドがファイル名を返すのと同じように、**LOAD** コマンドは **FLD** コマンドで指定された変数データ・フィールドの内容を返します。これは指定したファイルのみに適用されます。**LOAD** コマンドは、シリアル・インターフェースでのみにサポートされています。**LOAD** コマンドのフォーマットとその応答は次のとおりです。

**LOAD (filename)**



ユーザ・プロンプトと NN とフィールド名は、**FLD** コマンドで指定された内容です。

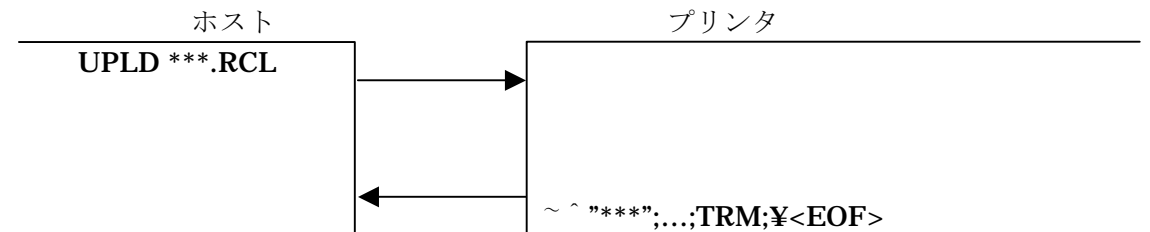
フィールド名は **CR** (0DH) と **LF** (0AH) で分離されます。一番最後のフィールド名の後ろには **CR**1 つと **2** つの **LF** コードが続きます。

## NITTO DENKO

### 14.2.7. フォーマット アップロードコマンド (UP Load : UPLD)

UPLD コマンドは、拡張メモリに保存されているフォーマットをアップロードするために使用します。UPLD コマンドは、シリアルインターフェースのみでサポートされています。

UPLD コマンドのフォーマットとその応答は次のとおりです。



アップロードの最後のデータは、EOF (1AH) でファイルへの格納ができるようになっています。

### 14.2.8. ファイル削除コマンド (DElete File : DELF)

DELF コマンドは拡張メモリ内のファイルを削除するために使用します。ファイル名は、拡張子も含めて指定します。

例)

`DELF TEST.RCL`

## 14.3. ロゴの拡張メモリへの書き込み

拡張メモリの中にロゴデータを書き込むことができます。

前もって拡張メモリにロゴデータを書き込むことにより、ホストからロゴデータを送信するより早く印字が開始できます。

書き込まれたロゴデータは、拡張メモリの中でファイルとして (拡張子 **RLG**) 取り扱われます。

### 14.3.1. ロゴファイルの書き込み

・コマンド (MAX) : 横 : ヘッド総ドット数(ドット) × 縦 : 最大印字ライン数(ドット)

#### (1) コマンド形式 : (1CH) “ファイル” ;x;y;n;データ;(19H)

ファイル名は、拡張子を含めて指定します。

$x = 1$  列のバイト数 ( $x = 1 \sim$  総ドット数 / 8)

$y =$  データの列数 ( $y = 1 \sim$  最大印字長ライン数)

$n =$  ロゴの個数 ( $n = 1$  以上)

#### (2) ロゴの個数について

同じサイズのロゴであれば 1 つのファイル内に 1 個以上の任意の個数を格納できます。ただし、何番目のロゴがどのロゴに当たるかなどは、ユーザー側で把握しておく必要があります (通常  $n = 1$  としてください)。



## NITTO DENKO

### (3)データ部分について

- ログデータの形式は、**10**進数表記または\$のついた**16**進数表記とします。従来のログデータと同じです。
- プリンタは拡張メモリに対して**(1CH)**から**(19H)**までのデータを書き込むときに、データの過不足のチェックは行いますが、正当性のチェックは行いません。



## NITTO DENKO

### 14.3.2. ロゴの印字

(1) コマンド形式 : **LOGF**;"ファイル名";

↑  
使用したいロゴの入ったファイル名の指定 (ロゴの登録時に指定したファイル名)。  
拡張子の省略が可能です。

**XLOG**;n;

↑  
ロゴ番号 : **LOGF** で指定してファイルの何番目のロゴを使用するか指定します (n=1 以上)

(2) コマンド例

~^"LOGOTEST";1;0;100;0;

**SPB**;UTOF;1100;HBR;100;VBR;50;

**LOGF**;"LOGFILE1";

← ログファイル指定"LOGFILE1.RLG"  
というファイル名のロゴを使用します。

**XLOG**;1;

← ファイル内の 1 番目のロゴを使用します。

**TRM**;¥

- ・ ロゴを使用するときは、“**LOGF**”と“**XLOG**”は対にして使用することとします。
- ・ ファイルの指定がない場合には、拡張メモリエラーとなります。
- ・ 他のロゴファイルのロゴを使用した場合には、改めて“**LOGF**”でそのファイルを指定しなくてはなりません。
- ・ ログファイルを使って展開できるロゴイメージのサイズの制限はありません。拡張メモリのサイズ、及び展開時のラベルサイズを超えない範囲でロゴデータを作成してください。拡張メモリにロゴファイルを登録し、それを **LOGF** コマンドで参照する方法は、**LOGO** コマンドを直接 **RCL** コマンド列に埋め込む方法と比べて、次の特徴があります。
  1. ログイメージデータを送信する必要がないので印字開始時間が早いです。
  2. ログファイルのデータは **LOGO** コマンドのように **CMDBUF** バッファを経由せずに、直接イメージバッファに書き込まれるので、**CMDBUF** サイズによるコマンド長の制限を受けません。

~^"FileA";1;0;100;0;

**SPB**;

**LOGF**;"LogoA";

**XLOG**;1;

**TRM**;¥

← LogoA.RLG 内のロゴサイズの制限はありません。

(3) 1 枚のラベル上に複数のロゴファイルを展開する例

## NITTO DENKO

RCL コマンドヘッダーとバッファ処理終了 (TRM;¥)、ブレーク (BRK;¥)の間には複数のロゴファイルのロゴデータが使用できます。しかし、同時にオープンできるファイルの数には制限があります。例えば同時に開けるファイルの数 (拡張メモリファイルオープン最大数) が 8 個の場合の例を示します。

1. 1 枚のラベル上のロゴデータが 7 個以下の時は、以下のように 7 個のロゴファイルを呼び出すことができます。すでに RCL のファイルを開いているのでファイルが 1 個オープン済みであることに注意してください。

```
~ ^ "FileA"1;0;100;0;
```

```
SPB;
```

```
HBR;10;VBR;10;LOGF;"Logo1";XLOG;1;
```

```
HBR;20;VBR;20;LOGF;"Logo2";XLOG;1;
```

```
HBR;30;VBR;30;LOGF;"Logo3";XLOG;1;
```

```
HBR;40;VBR;40;LOGF;"Logo4";XLOG;1;
```

```
HBR;50;VBR;50;LOGF;"Logo5";XLOG;1;
```

```
HBR;60;VBR;60;LOGF;"Logo6";XLOG;1;
```

```
HBR;70;VBR;70;LOGF;"Logo7";XLOG;1;
```

TRM;¥      TRM;¥でオープンされていた 7 個のロゴファイルがクローズされます。

2. 1 枚のラベル上のロゴデータが 8 個以上の時は、フォーマットファイルを分割することにより、複数のフォーマットファイルを重ね合わせてラベルイメージを作成できます。次の例は拡張メモリに格納されている 2 個のフォーマットファイルを使って、14 個のロゴを印字する例です。

(1CH)

```
~ ^ "FileC";1;0;100;0;
```

```
SPB;
```

```
HBR;10;VBR;10;LOGF;"Logo1";XLOG;1;
```

```
HBR;20;VBR;20;LOGF;"Logo2";XLOG;1;
```

```
HBR;30;VBR;30;LOGF;"Logo3";XLOG;1;
```

```
HBR;40;VBR;40;LOGF;"Logo4";XLOG;1;
```

```
HBR;50;VBR;50;LOGF;"Logo5";XLOG;1;
```

```
HBR;60;VBR;60;LOGF;"Logo6";XLOG;1;
```

```
HBR;70;VBR;70;LOGF;"Logo7";XLOG;1;
```

```
BRK;¥
```

(19H)

(1CH)

```
~ ^ "FileD";1;0;100;0;
```

```
RSPB;
```

```
HBR;10;VBR;20;LOGF;"Logo8";XLOG;1;
```

```
HBR;20;VBR;30;LOGF;"Logo9";XLOG;1;
```

## NITTO DENKO

HBR;20;VBR;30;LOGF;"Logo9";XLOG;1;  
HBR;30;VBR;40;LOGF;"Logo10";XLOG;1;  
HBR;40;VBR;50;LOGF;"Logo11";XLOG;1;  
HBR;50;VBR;60;LOGF;"Logo12";XLOG;1;  
HBR;60;VBR;70;LOGF;"Logo13";XLOG;1;  
HBR;70;VBR;10;LOGF;"Logo14";XLOG;1;  
TRM;¥  
(19H)

### \*FileC

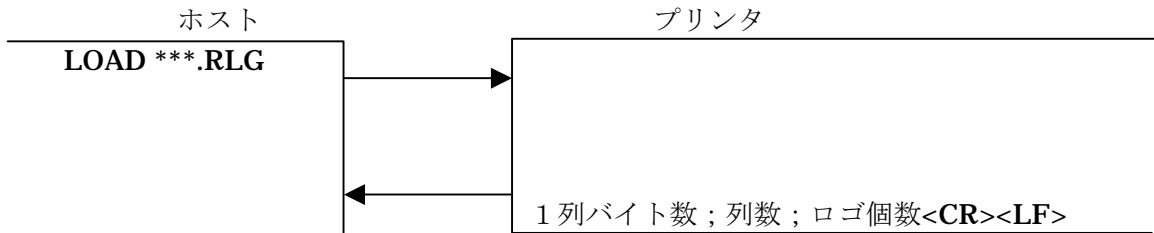
! 1           **FileC** をイメージ展開する過程で 7 個のロゴファイルがオープンされます。

### \*FileD

! 1           **FileD** をイメージ展開する過程で 7 個のロゴファイルがオープンされます。  
印字されたラベルには、**14** 個のロゴファイルイメージが展開されます。

### 14.3.3. ロゴの大きさ要求コマンド (LOAD)

ロゴの **LOAD** コマンドは 1 列のバイト数、データ個数、ロゴの個数を返します。



### 14.3.4. ロゴデータのアップロード (UPLD)

ロゴの **UPLD** コマンドは、拡張メモリに保存されているロゴデータをアップロードするために使用します。

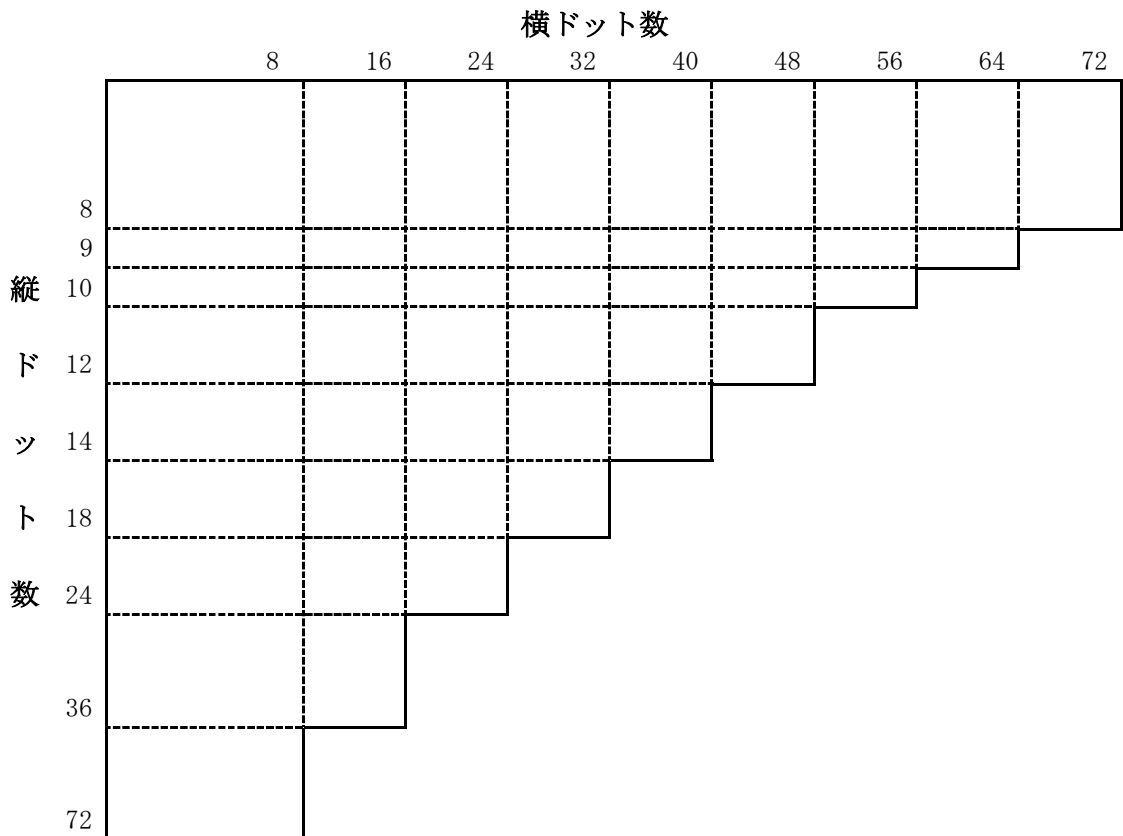


#### 14.4. 外字ファイルの書き込み

ユーザーが設計したドットフォントデータを外字ファイルとして登録することにより、標準対応のドットフォントと同じ手順で印字することができます。文字の回転や拡大、インクリメント・デクリメントも可能です。

登録できるフォントサイズは下図枠内の縦×横の組み合わせとなります。

外字フォントサイズ (1文字を構成する縦、横のドット数)



外字ファイルの登録コマンドは次のとおりです。

<FS> “ファイル名” ;  
**0**;X;Y;C1;N1;C2;N2;.....;SFF;データ;<EM>

ファイル名は、拡張子を含めて指定します。

**0**=機能設定フラグ。常に**0**を設定します。

**X=1** 文字を構成する横ドット数

**Y=1** 文字を構成する縦ドット数

**C1;N1;C2;N2;.....**=コード表中の外字を登録するエリア指定。次頁で詳述します。

**SFF**=エリア指定部分のターミネータ

データ=外字データ部分。次頁で詳述します。

横ドット数、縦ドット数、エリア指定、データは **10** 進数表記または**\$**のついた **16** 進数表記で指定します。

## NITTO DENKO

### 14.4.1. 外字登録とコード表

外字は、ユーザーが書体を設計したドットフォントで、共通の書体を持ったひとまとまりの外字を文字セットとして拡張メモリに登録することができます。フル ASCII の文字セットは、**20hex**～**7Ehex** の **95** 文字を登録します。又、数字だけの文字セットや英大文字だけの文字セット、あるいはこれらの組み合わせの文字セットを作る事もできます。さらに、カナと英数字を組み合わせた文字セットも可能です。外字は **1** バイトコード文字で、使用できるコードは、**20hex**～**7Ehex**、**80hex**～**0FFhex** の範囲です。

どのコードに文字を割り当てるかは、コマンドの変数にある“コード表中の外字を登録するエリア指定”で行います。**Cx;Nx;**はコード **Cx** から **Nx** 個の連続したエリアに外字を割り付けることを表します。**Cx, Nx,**を複数組み合わせる事により、連続しない任意の位置に必要な文字数の外字を登録することができます。例えば、数字 **10** 文字 (**30hex** を先頭に連続配置する) と英大文字 **26** 文字 (**41hex** を先頭に連続配置する) の計 **36** 文字で構成される外字セットを、登録するときのコマンドは次のようになります。

<FS> “ファイル名” ;**0;X;Y;\$30;10;\$41;26;\$FF;**データ;<EM>

### 14.4.2. データ部分について

**1** 文字のデータは必ずバイト単位で指定しなければなりません。また、外字データはコードの小さいほうから順に、指定しなければなりません。

次の例は **12×12** ドットの文字“**A**”を外字登録するときのデータ例を示しています。

” A” (41h) の外字データ (12×12 ドット)

	第1バイト目								第2バイト目								第1 バイト目	第2 バイト目
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
1																	→ 00 H	00 H
2																	→ 06 H	00 H
3																	→ 0F H	00 H
4																	→ 1F H	80 H
5																	→ 19 H	80 H
6																	→ 30 H	C0 H
7																	→ 3F H	C0 H
8																	→ 7F H	E0 H
9																	→ 60 H	60 H
10																	→ 60 H	60 H
11																	→ 60 H	60 H
12																	→ 00 H	00 H

## NITTO DENKO

実フォントサイズは **12×12** ドットですが、横方向の区切りをバイト単位にするために **16×12** ドットが **1** 文字の登録単位（データサイズ）となります。  
文字のデータは、各行左詰で **1** 行毎にバイト境界になければなりません。

上の“**A**”のデータを **16** 進数表記したものが下図左側です。フル **ASCII** の外字ファイルこれが下図右側 **41h** のデータになります。**41h** 以外のデータについても同様に作成し、コマンドのデータ部に埋め込まれると、外字登録コマンド全体は、下図右側全体のようになります。

	<b>&lt;FS&gt;"GAIJI1.RFN";0;12;12;\$20;95;\$FF;</b>
<b>\$00;\$00;</b>	<b>20h のデータ (32 バイト)</b>
<b>\$06;\$00;</b>	
<b>\$0F;\$00;</b>	
<b>\$1F;\$80;</b>	
<b>\$19;\$80;</b>	<b>21h のデータ (32 バイト)</b>
<b>\$30;\$C0;</b>	
<b>\$3F;\$C0;</b>	
<b>\$7F;\$E0;</b>	
<b>\$60;\$60;</b>	<b>41h のデータ (32 バイト)</b>
<b>\$60;\$60;</b>	
<b>\$60;\$60;</b>	
<b>\$00;\$00;</b>	
	<b>&lt;EM&gt;</b>
	<b>7Eh のデータ (32 バイト)</b>

フォントサイズについて

外字を拡大や回転して印字するときにはフォントサイズに次の制限が生じます。

外字 **1** 文字のデータの横バイト数を **m**、縦ドットを **n** とすると、

拡大時： $m \times n \leq 1152$  (バイト) であること。

回転時： $m \times n \leq 184$  (バイト) であること。

外字の印字 (**FNTF**)

外字の印字は、外字種類指定コマンド (**FNTF**) とドットフォント指定コマンド (**DDF**) を組み合わせて行います。外字種類指定コマンドは、登録されている複数の外字ファイルのどれを使うかを指定します。このコマンドは拡張メモリに登録されている外字ファイルが **1** つだけの場合でも必要です。反対に、**1** 枚のラベル内に複数の外字セットを使用することもできます。

**FNTF**; “ファイル名” ;**N**;

**N=DDF** の設定値。ただし、**200** 以上 **300** 未満の任意の値。

ファイル名=使用する外字ファイルの指定。フォント登録時に指定したファイル名を使います。拡張子は省略することができます。



## NITTO DENKO

**DDF;A;B;**

**A**=ドットフォント種類を指定します。**FNTF** で設定した値 **N** を用います。

**B**=現在は意味を持ちません。**10** を指定してください。

次の例では外字種類指定コマンドで“**GAIJI1.RFN**”というファイルの外字を **DDF;200;** に割り付けて使用しています。

```
~^ "GAIJITST";1;0;100;0;  
SPB;HBR;100;VBR;50;  
FNTF;"GAIJI1";200;  
DDF;200;10;  
DFM;1;2;DFO;2;2;DFS;3;  
"GAIJI FONT TEST PRINT";  
TRM;¥
```

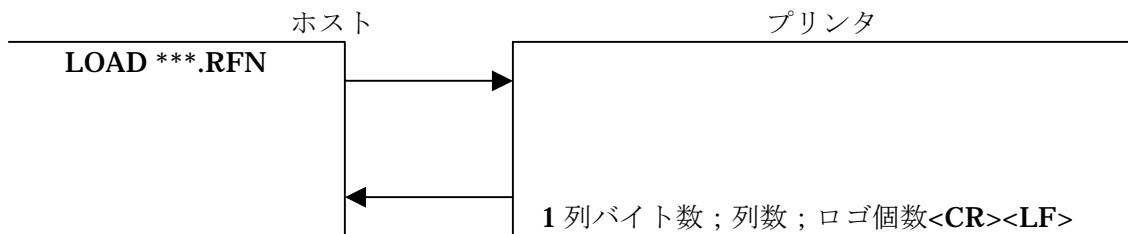
外字を展開する位置は **HBR,VBR** で指定します。

外字を使用する場合には、**FNTF** を指定してからドットフォント指定 (**DDF**) を指定してください。

**FNTF** の指定がない場合には、ドットフォントのデフォルト値である英数字 **M (DDF;3;)** で印字を行います。

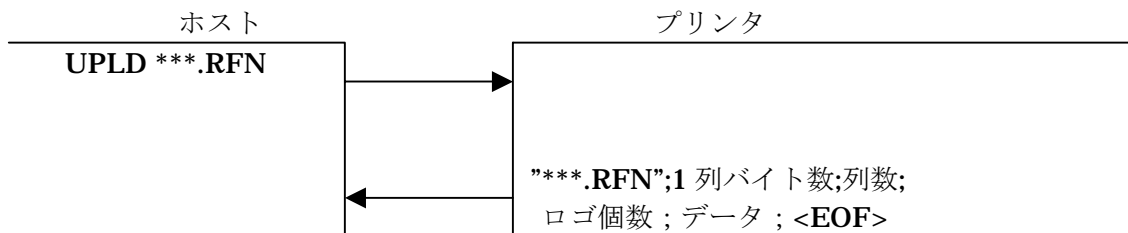
### 14.4.3. 外字の大きさ要求コマンド (LOAD)

外字の **LOAD** コマンドは 1 列のバイト数、データ个数、ロゴの個数を返します。



### 14.4.4. ロゴデータのアップロード (UPLD)

ロゴの **UPLD** コマンドは、拡張メモリに保存されているロゴデータをアップロードするために使用します。



## 14.5. 拡張メモリを使うときの注意点

拡張メモリ内部のデータフォーマットは、フォーマットファイルを **RCL** コマンド列はテキスト形式のまま格納し、ロゴファイルは、参照時の処理速度向上のためにバイナリー形式で格納されます。バイナリー形式に変換されたロゴファイルのサイズは、ホストが送信したファイルサイズの約 **1/4** になります。

拡張メモリに登録できるファイル数は、プリンタのモデル（使用する拡張メモリ）により変わります。詳細は付録のプリンタ仕様をご覧ください。

拡張メモリに格納されているフォーマットファイルを呼び出すときは、**RCL** コマンド列の終了を示す”**¥**”マークをもってファイルの終了と判断します。従って、**1**つのフォーマットファイルの中には **RCL** ヘッダーと **TRM;¥**（または **BRK;¥**）は、それぞれ **1**つだけ記述できます。

拡張メモリに登録するファイルのファイル名は、**DOS** ファイル名に使用できる文字が使えます。**RCL** コマンドヘッダーのファイル名に許されている記号の **1**部は使えません。

## 15. カッター機能

---

ここではカッター搭載プリンタのカッター関連機能について説明します。

### 15.1. カッター制御コマンド

カッター制御コマンドとして、カッター動作 (**CuTter ON:CTON**) /カッター停止 (**CuTter Off:CTOF**) コマンドと、バッチカット枚数指定 (**NUmber of Batch Cut:NUBC**) コマンドがあります。

なおこれらのコマンドはカッター付きのプリンタのみ有効です。カッターなしまたは、プリンタがカッター動作しない設定の場合には、単に無視されます。

#### 15.1.1. カッター動作 (**CuTter ON : CTON**)

**CTON** コマンドは、印字したラベルをカットするように指示します。

このコマンドを送ると、バッチカット枚数指定 (**NUBC**) コマンドで設定された枚数ごとにラベルをカットします。電源オンしたときの初期設定は **CTON** の状態です。

このコマンドの構文は次のとおりです。

**CTON;**

#### 15.1.2. カッター停止 (**CuTter Off : CTOF**)

**CTOF** コマンドは、印字したラベルのカットを停止します。

このコマンドを送ると、カッター動作を行いません。この状態は **CTON** コマンドや **CAN** コマンドを送るか、プリンタの電源がオフされるまで続きます。

このコマンドの構文は次のとおりです。

**CTOF;**

**CTON**/**CTOF** コマンドは、どちらか **1** つをバッチ※内で指定します。コマンドが省略されたときは、以前に実行したバッチの設定が継続されます。また、同一バッチ内で異なる設定を指定したときは、あとから指定された動作が有効となります。

※**RCL** のコマンドヘッダーと **TRM;¥** (又は **BRK;¥**) で囲まれている一連のコマンド列

#### 15.1.3. バッチカット枚数指定 (**NUmber of Batch Cut : NUBC**)

**NUBC** コマンドは、カットするまでに印字するラベルの数(カット単位)を指定します。

このコマンドが省略されたときは、それ以前のバッチで指定されたパラメータが継続して使用されます。ただし、ここで指定した枚数に達しなくても、コマンドヘッダーで指定した印刷枚数に達した時にカットが行われます。

**NUBC** コマンド指定しない場合のカット枚数の既定値は **1** です。

## NITTO DENKO

このコマンドの構文は次のとおりです。

**NUBC;N;**

N=バッチカット枚数 (1~65536)。

次の例では、10枚印字のバッチに対して、4枚目、8枚目、10枚目を印字後にカットしています。

~^"CT1";10;0;100;0;

SPB;CTON;NUBC;4;

HBR;50;VBR;70;

BDEF;1;BCSH;50;BNEW;2;BWEW;5;BCST;

"\*1234567890\*";

BSTP;

TRM;¥

## 16. トラブルシューティングガイド

あらゆるプログラム言語と同様、**RCL**でも不正確な操作によってエラーが生じます。最も一般的なエラーは、コマンドの構文の間違い、タイプミス、通信インターフェースの障害、あるいはラベル上のオブジェクトが適正に配置されていないことなどにより起こされるエラーです。この章では、役に立つデバッグの方法と **16** 進ダンプについて説明し、最も一般的な問題とその解決策を示します。

### 16.1. テキスト印字/16進ダンプ

プリンタの設定を変更することによりプリンタで受信したデータのテキスト印字、または **16** 進ダンプ印字ができます。(印字の為にプリンタの最大印字幅の連続紙を用意し、プリンタの用紙設定を連続紙モードにしてください)。

#### ① テキスト印字

- ・ プリンタをテキスト印字の状態にする方法は、本体の取扱説明書をご覧ください。
- ・ ホストよりデータを送信します。下記のようなテキスト印字が行われます。

```

0^"QRCODE";1;0;420;000;SPB;HBR;0;UBR;0;D
DF;8;10;DFH;2;2;1BR;80;UBR;0;" .B.K#Q#R#3
I<#I.H.A";DFH;1;1;HBR;5;UBR;15;"LEVEL=H:
MASK=AUTO:DATA=K.B.K4A;z.H.A,N1234,B0004
test";DTDS;100;1;QRCS;8;QRHP;9;HBR;50;UB
R;30;TDEL;1;BCS1;"K.B.K4A;z.H.A,N1234,B0
004test";BSTP;HI;R;50;UBR;130;TDEL;2;BCST

```

#### ② 16進ダンプ印字

- ・ プリンタをテキスト印字の状態にする方法は、本体の取扱説明書をご覧ください。
- ・ ホストよりデータを送信します。下記のようなテキスト印字が行われます。

```

56 42 52 3B 31 35 3B 22 UBR;15;"
4C 45 56 45 4C 3D 48 3A LEVEL=H:
4D 41 53 4B 3D 41 55 54 MASK=AUT
4F 3A 44 41 54 41 3D 4B O:DATA=K
1C 42 1B 4B 34 41 3B-7A .B.K4A;z
1B 4B 1C 41 2C 4E 31 32 .H.A,N12
33 34 2C 42 30 30 30 34 34,B0004
74 65 73 74 22 3B 44 54 test";DT
44 53 3B 31 30 30 3B 31 DS;100;1

```

## 16.2. 一般的な問題とその対策

最も一般的に起こり得る問題と、その原因、対策を以下に示します。

### 16.2.1. 問題

プリンタにコマンド列を送ったが、何も印刷されない。  
そして、エラー**LED**は点灯になっていない。

#### 16.2.1.2. 原因

- a. 引用符が閉じていない（コマンド列のどこかで、閉じる引用符(“)を忘れている）。
- b. 開いたコメントがある（コマンド列のどこかで、閉じるコメント記号(#)を忘れている）。
- c. ヘッダで指定したラベルの印刷長さが、ラベル長設定（**UTOF** 値）よりも大きい。

#### 16.2.1.3. 対策

- a. 閉じていない引用符を調べて閉じる。
- b. 閉じていないコメントを調べて閉じる。
- c. ヘッダで指定したプリント長さを調べ、ラベル長設定（**UTOF** 値）の大きさかそれ以下にする。

### 16.2.1.4. 問題

プリンタにコマンド列を送ると、ラベルの代わりに、コマンド列の **1** 部または全体を **ASCII** テキスト形式で印刷し始める。

#### 16.2.1.5. 原因

- a. 認識できないコマンドかパラメータがある。
- b. ホストとプリンタのハンドシェイクに問題がある。

#### 16.2.1.6. 対策

- a. **ASCII** テキストで印刷されている最初の文字の前のプログラムを調べる。**RCL** では、認識できないデータがあるとテキスト・ダンプ・モードに変わります。この場合、セミコロン記号が抜けていたり、コマンドの引き数が抜けていたり、あるいは引き数のパラメータが多すぎたりする場合があります。この種の誤りを訂正して、送信してください。
- b. ホスト・コンピュータとのプロトコルが、ハードウェアによるものか、**XON/XOFF** によるものかを調べてください。ホストの条件設定が適切に行われているか確かめてください。ホストとプリンタの間のケーブルと接続を調べてください。それでも問題が解決しないならば、コマンド列の **16** 進ダンプを行い、第 **1** 行を調べて、プロトコルに問題がないか調べてください。そのためには、プログラム・コードと **16** 進ダンプの違いを見つけ、例えばコマンドが欠けていたり、あるいはパラメータが欠けていれば、プロトコルに問題があります。

### 16.2.1.7. 問題

プリンタにコマンド列を送ると、ラベルの代わりに「^"FILE NAME";」で始まるコマンドのテキスト・ダンプが印刷される。

#### 16.2.1.8. 原因

- a. ヘッダの開始文字がない。
- b. おそらくパリティが原因で、プリンタがヘッダの開始文字を認識していない。
- c. ホストが **SOH** を送れない。

## NITTO DENKO

### 16.2.1.9. 対策

- a. コマンド列の先頭にヘッダ文字開始、**SOH (01H)** または (~) を入れる。
- b. ホストとプリンタが、同じパリティに設定されていることを確かめる。
  - 8 ビット、1 ストップ、パリティなし
  - 7 ビット、1 ストップ、偶数パリティ
  - 7 ビット、1 ストップ、奇数パリティ
- c. **SOH (01H)** の代わりに波形符の (~) をヘッダ文字の開始として使用する。

### 16.2.1.10. 問題

紙が正しくセットされているにも関わらず、ペーパーアウトやペーパージャム・エラーが発生する。

### 16.2.1.11. 原因

- a. センサが用紙を検出していない。
- b. フォームの長さの設定が正しくない。
- c. 使われている用紙の透明度が、プリンタの仕様に一致していない。

### 16.2.1.12. 対策

- a. 用紙を再度装着し、キャリッジ機構にきちんとセットされているか確かめる。
- b. **UTOF** コマンドを使って、フォームの長さを再設定する（より正確に）。
- c. 異なったタイプの用紙を試してみる。

### 16.2.1.13. 問題

ほとんどのラベルは問題なく印刷するが、時々印刷位置が狂う。

### 16.2.1.14. 原因

使用している用紙の透明度が、プリンタの仕様に適合しない。

### 16.2.1.15. 対策

用紙を変える。

### 16.2.1.16. 問題

一連のラベルを印刷する際に、ラベルが 1 枚おきに余白になる。

### 16.2.1.17. 原因

- a. ヘッダで指定したプリント長さが、使用している用紙の長さと同様か、ほとんど違う。
- b. フォームの長さが、使用している用紙よりも長く設定されている。

### 16.2.1.18. 対策

- a. プリント長さを、用紙の長さよりも短くする。
- b. **UTOF** を使ってフォームの長さを、用紙の長さと同じになるように再設定する（非常に正確に）。

## NITTO DENKO

### 16.2.1.19. 問題

ラベルの 1 部だけが印刷され、ラベルの下部が余白になる。

### 16.2.1.20. 原因

ヘッダのプリント長さが小さすぎ、ラベルの下部の印刷内容を印字しない。

### 16.2.1.21. 対策

- a. ヘッダのプリント長さを増やす。
- b. 印刷されないオブジェクトの位置を変えて、ラベルの上の方に印刷するようにする。

### 16.2.1.22. 問題

ラベルの上部に印刷される最初のオブジェクト(特にテキストあるいはバーコード)が、正しく印刷されない。

### 16.2.1.23. 原因

そのオブジェクトがラベルの上部に近接しすぎて配置されており、印刷するための十分な余地がない。

### 16.2.1.24. 対策

**VPR** または **VBR** の初期設定を大きくして、最初のオブジェクトを下へ移動する。

### 16.2.1.25. 問題

印刷されたオブジェクトが用紙の左端または右端で重なって印刷されているか、オブジェクトが用紙の左端か右端で切断されている。

### 16.2.1.26. 原因

- a. フォームの左端が用紙の左端から離れすぎた位置に設定されているために、ラベルの左端に近いオブジェクトを印刷する余地がない。
- b. **HPR** または **HBR** の設定が正しくない。通常、値が大きすぎる。

### 16.2.1.27. 対策

- a. フォームの左端を使用している用紙の左端に正しく設定されているか確かめてください。
- b. すべての **HPR** と **HBR** コマンドを調べ、値が正しいことを確かめてください。正しくないときは値を小さくし、再び転送してください。

### 16.2.1.28. 問題

ティプトン・ゴシック・フォントを使っている時に、小さいテキストがラベルに印刷できない。

### 16.2.1.29. 原因

おそらく、文字がこのフォントの最小の大きさよりも小さすぎる。ティプトン・ゴシック・フォントは **DCH;10;DCW;20**;よりも小さいサイズでは印刷できません。

### 16.2.1.30. 対策

**DCH;10;DCW;20**;よりも小さい文字には、ティプトン・ゴシック・フォントの代わりにローテータブルフォントまたはドットフォントを使用する。

### 16.2.2.31. 問題

**Code39** または **CodaBar** のバーコードで、プリンタではスキャンできたのに、他のバーコードリーダーでは正しく読み取れない。



## NITTO DENKO

### 16.2.1.32. 原因

- a. 開始文字や停止文字が間違っているか、無い。
- b. エレメントの幅や密度が正しくない。

### 16.2.1.33. 対策

- a. **Code39** バーコードのスタートキャラクタとストップキャラクタとして、アスタリスク記号(\*)を使用する。これはプログラマーが指定しなければならない。
- b. **CodaBar** バーコードのスタートキャラクタとストップキャラクタとして、**A、B、C、D**を使用する。これらはプログラマーが指定しなければならない。
- c. **Code39** の **BWEW** と **BNEW** コマンドのパラメータ値、**CodaBar** の **BCPI** コマンドのパラメータ値を、最新の規格と比較して、これらのコマンドに正しい値が定義されていることを確かめる。

### 16.2.1.34. 問題

バーコードを連番で印刷している時に、最初のラベルは問題なく印刷されるが、残りのラベルが無効になり、連番を印刷したバーコードのスキャンができない。

### 16.2.1.35. 原因

**DWBX** (枠内消去) コマンドで前のバーコードを消していないため、次のバーコードを前のバーコードの上に印刷する。

### 16.2.1.36. 対策

- a. **DWBX** コマンドが発行されていることを確かめる。
- b. **DWBX** が正しく配置され、バーコードの全体を消去することを確かめる。
- c. **DWBX** で消去する領域を拡大して、バーコードの全体を完全に消去させる。

### 16.2.1.37. 問題

バーコードを連番で印刷している時、インクリメント/デクリメントの量が正しくないか、連番を印刷する前に間違った数のラベルを印刷する。

### 16.2.1.38. 原因

- a. **BCID** と **IDF** または **BCLC** コマンドが、コマンド列内の連番フィールドの後ろにあるため、指定した値の代わりにこれらコマンドの初期値で印刷される。
- b. **BCID** と **IDF** または **BCLC** コマンドで定義した値が正しくないか、これらのコマンドの **1** つが無い。

### 16.2.1.39. 対策

- a. **BCID** と **IDF** または **BCLC** コマンドを連番フィールドの前に配置する。
- b. コマンド列を再検査して、**BCID** と **IDF** と **BCLC** が適切に定義されているかどうか確かめる。

## 17. コマンド概要

---

ALPH : ALPHabetic	英字連番	95
APOF : APpc OFF	APPC オフ	52
APON : APpc ON	APPC オン	53
AQL : Assure Quality Level set	APPC レベル設定	53
BCID : Bar Code Increment/Decrement	バーコード連番インクリメント・デクリメント設定	96
BCLC : Bar Code Label Count	リピート枚数設定	92
BCPI : Bar code Character Per Inch	バーコード密度設定	37,40,42,59
BCSH : Bar Code Symbol Height	バーコード高さ設定	32,60
BCST : Bar Code StarT	バーコードデータ開始	29
BDEF : Bar code DEFinition	バーコード選択	28
BCIG : Bar Code Intercharacter Gap	バーコードインターキャラクターギャップ設定	33
BNEW : Bar Narrow Element Width	バーの細線幅設定	32
BOTH : BOTH	英数字連番	96
BRK : BReaK	ブレーク	11
BSAL : Bar Save Address Length	バーコードフィールド連番長さ保存	97
BSTP : Bar code SToP	バーコードデータ終了	30,59
BSYM : Bar code SYMbol	バーコード選択	27
BVLP : Bar code Variable field Length and Position	バーコードフィールド連番長さ 及び連番桁位置指定	98
BWEW : Bar Wide Element Width	バーの太線幅設定	32
CAN : Initialize Printer	プリンタ初期化	105
CTOF : CuTter OFF	カッター停止	134
CTON : CuTter ON	カッター動作	134
DBBX : Draw Black BoX	塗りつぶし	24
DBOX : Draw BOX	枠描画	23
DC2 : Pause ON	一時停止	105
DC4 : Pause OFF	一時停止解除	105
DCBX : Draw Complement BoX	反転枠描画	25
DCH : Define Character Height	文字高さ設定	71
DCS : Define Character Size	アウトラインフォントキャラクターサイズ指定	86
DCW : Define Character Width	文字幅設定	71
DDF : Define Dot Font	ドットフォント指定	78
DDL : Draw Diagonal Line	対角線描画	20
DELf : DEL File	ファイル削除	123
DFM : Dot Font Magnification	文字拡大率指定	82

## NITTO DENKO

<b>DFO : Dot Font Orientation</b>	印字方向指定	<b>83,87</b>
<b>DFS : Dot Font Space</b>	文字間スペース量設定	<b>82,87</b>
<b>DHL : Draw Horizontal Line</b>	水平線描画	<b>18</b>
<b>DHR : Define Human Readable</b>	ベクトルフォント指定	<b>66</b>
<b>DIR : file DIRectory</b>	ファイル名要求	<b>122</b>
<b>DVfy : Disable Verification</b>	検証機能無効	<b>53</b>
<b>DVL : Draw Vertical Line</b>	垂直線描画	<b>19</b>
<b>DWBX : Draw White BoX</b>	枠内消去	<b>25</b>
<b>EMON : Erase Mode ON</b>	自動消去オン	<b>94</b>
<b>ENQ : Status request</b>	ステータス要求	<b>105</b>
<b>EOL : End Of Line</b>	水平位置リセット	<b>16</b>
<b>EXCP : EXCePt</b>	指定文字削除	<b>100</b>
<b>FLD : FieLD</b>	可変フィールド設定	<b>118</b>
<b>FLIP : FLIP</b>	上下反転	<b>104</b>
~ ^	ヘッダ	<b>6</b>
<b>HALF : HALF dot on</b>	ハーフドット・オン	<b>55</b>
<b>HBR : Horizontal Base Reference</b>	絶対水平位置移動	<b>14</b>
<b>HLFB : HaLF dot on Bar</b>	ハーフドット・オン・バー	<b>55</b>
<b>HLT : Horizontal Line Thickness</b>	水平線太さ設定	<b>18</b>
<b>HOFF : Half dot OFF</b>	ハーフドット・オフ	<b>55</b>
<b>HOME : HOME</b>	ホームポジション	<b>17</b>
<b>HPR : Horizontal Position Relative</b>	相対水平位置移動	<b>15</b>
<b>ICS : Inter Character Space</b>	文字間スペース	<b>72</b>
<b>IDF : Increment/Decrement Field</b>	テキストフィールド・インクリメント・デクリメント設定	<b>98</b>
<b>INFO : INFOrmation</b>	プリンタ情報	<b>106</b>
<b>LOAD : LOAD field name</b>	フィールド名要求	<b>122,128,132</b>
<b>LOGD : LOGo Data format</b>	ロゴデータフォーマット指定つきコマンド	<b>109</b>
<b>LOGF : LOGo File</b>	拡張メモリロゴファイル	<b>126</b>
<b>LOGO : LOGO</b>	ロゴ	<b>108</b>
<b>MRK : MaRK</b>	マーク	<b>91</b>
<b>NORP : NO RePrint</b>	再印字禁止機能	<b>54</b>
<b>NUBC : NUmBer of Cut</b>	バッチカット枚数指定	<b>134</b>
<b>NUM : NUMeric</b>	数字連番	<b>95</b>
<b>QRCS : QR code Cell Size</b>	<b>QR Code</b> シンボル用セルサイズ設定	<b>61</b>
<b>QREL : QR code Error Level</b>	<b>QR Code</b> エラー修復レベル設定	<b>61</b>
<b>QRMP : QR code Mask Pattern</b>	<b>QR Code</b> マスクパターン設定コマンド	<b>62</b>
<b>RET : RETurn</b>	リターン	<b>92</b>
<b>RSET : ReSET</b>	フィード距離リセットコマンド	<b>107</b>

## NITTO DENKO

<b>RSPB : ReStart Processing Buffer</b>	バッファ処理再開	<b>10</b>
<b>SAL : Save Address Length</b>	テキストフィールド <sup>※</sup> 連番長さ保存	<b>98</b>
<b>SPB : Start Processing Buffer</b>	バッファ処理開始	<b>10</b>
<b>STX : Status request</b>	ステータス要求	<b>105</b>
<b>TDEL : Two Dimensional Error Level</b>	2次元シンボルエラー修復レベル設定	<b>62</b>
<b>TRM : TeRMinate</b>	コマンド列終了	<b>10</b>
<b>UMAG : Upc MAGnification</b>	UPC 拡大率設定	<b>35</b>
<b>UPLD : UP LoaD</b>	拡張メモリアップロード <sup>※</sup>	<b>123,128,132</b>
<b>UTOF : Use Top Of Form</b>	ラベル長設定	<b>103</b>
<b>VBR : Vertical Base Reference</b>	絶対垂直位置移動	<b>14</b>
<b>VFYM : VeriFY Mode</b>	検証モード設定	<b>52</b>
<b>VLP : Variable field</b>	テキストフィールド <sup>※</sup> 連番長さ	<b>99</b>
<b>Length and PositiOn</b>	及び連番桁位置指定	
<b>VLT : Vertical Line Thickness</b>	垂直線太さ	<b>19</b>
<b>VPR : Vertical Position Relative</b>	相対垂直位置移動	<b>16</b>
<b>XLOG : X LOGo</b>	ロゴファイル内データ個数	<b>126</b>
<b>{F;XX;YY;} : Function mode command</b>	機能設定コマンド	<b>102</b>

## 18. プリンタ仕様

ここでは、**Dura Printer** シリーズの各モデル毎の仕様（印字最大幅、最大長等）および各モデル毎のバーコード密度一覧表と文字サイズ表を提供します。

### 18.1. モデル毎仕様

モデル名	R	SG	SR	SRS
ヘッドドット密度	<b>8 dot/mm</b>	<b>7.6dot/mm</b>	<b>15.75dot/mm</b>	<b>15.75dot/mm</b>
<b>L/Fピッチ</b>	<b>0.127mm</b>	<b>0.1314mm</b>	<b>0.0635mm</b>	<b>0.0635mm</b>
ヘッド総ドット数	<b>1216</b>	<b>800</b>	<b>1536</b>	<b>1536</b>
印字領域の長さ	<b>6~256mm</b>	<b>20~256mm</b>	<b>2~100mm</b>	<b>5~100mm</b>
最大印字幅	<b>156mm</b>	<b>105.6mm</b>	<b>95.75mm</b>	<b>95.75mm</b>
検証機能	あり	なし	あり	なし
ラベルセット	センター振り分け	センター振り分け	センター振り分け	センター振り分け
制御バッファメモリ	<b>3KB</b>	<b>7KB</b>	<b>7KB</b>	<b>7KB</b>
アウトラインフォント	工場オプション	標準搭載	工場オプション	標準搭載
拡張メモリ種類	メモ리카ード	メモ리카ード	メモ리카ード	メモ리카ード
拡張メモリファイル登録数	メモリ容量による	<b>128</b>	<b>128</b>	<b>128</b>
オープン最大数	<b>1</b>	<b>8</b>	<b>8</b>	<b>8</b>

モデル名	LSP5300/LP5320	KP4300	KP3000
ヘッドドット密度	<b>11.8dot/mm</b>	<b>12dot/mm</b>	<b>12dot/mm</b>
<b>L/Fピッチ</b>	<b>0.0833mm</b>	<b>0.0833mm</b>	<b>0.0833mm</b>
ヘッド総ドット数	<b>1536</b>	—	—
印字領域の長さ	<b>20~254mm</b>	<b>25~400mm</b>	<b>40~180mm</b>
最大印字幅	<b>127mm</b>	<b>109mm</b>	<b>76mm</b>
検証機能	なし	なし	なし
ラベルセット	センター振り分け	左合せ	センター振り分け
制御バッファメモリ	<b>21KB</b>	<b>21KB</b>	<b>21KB</b>
アウトラインフォント	標準搭載	標準搭載	標準搭載
拡張メモリ種類	フラッシュメモリーボード	フラッシュメモリーボード	フラッシュメモリーボード
拡張メモリファイル登録数	<b>1024</b>	<b>1024</b>	<b>1024</b>
オープン最大数	<b>16</b>	<b>16</b>	<b>16</b>

## 19. モデル毎のバーコード密度一覧表

---

### 19.1. 8ドット/mmヘッド機

#### 19.1.1. Code39

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	6.90
1	0.254	0.762	3.0:1	6.25
2	0.381	0.889	2.33:1	4.76
3	0.381	1.016	2.67:1	4.44
4	0.508	1.143	2.25:1	3.64

#### 19.1.2. ITF

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	12.50
1	0.254	0.762	3.0:1	11.11
2	0.381	0.889	2.33:1	8.70
3	0.381	1.016	2.67:1	8.00
4	0.508	1.143	2.25:1	6.67

#### 19.1.3. UPC/EAN/JAN

ピケットフェンスバーコード

UMAG	モジュール寸法 (mm)	拡大率
0	0.25	75.8
1	0.3125	94.2
2	0.375	113.6
3	0.4375	132.6
4	0.5	151.5
5	0.5625	170.5
6	0.625	189.4
7	0.6875	208.3

ラダーバーコード

UMAG	モジュール寸法 (mm)	拡大率
0	0.254	76.9
1	0.3175	96.2
2	0.381	115.5
3	0.4445	134.7
4	0.508	153.9

## NITTO DENKO

### 19.1.4. CodaBar

ピケットフェンスバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.25	0.625	2.5:1	9.24
1	0.25	0.75	3.0:1	8.47
2	0.375	0.875	2.33:1	6.35
3	0.375	1.0	2.67:1	5.98
4	0.5	1.125	2.25:1	4.84
5	0.5	1.375	2.75:1	4.42

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	9.09
1	0.254	0.762	3.0:1	8.33
2	0.381	0.889	2.33:1	6.25
3	0.381	1.016	2.67:1	5.88
4	0.508	1.143	2.25:1	4.76

### 19.1.5. Code93

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.125	22.5
1	0.25	11.29
2	0.375	7.53
3	0.5	5.64
4	0.625	4.52

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.254	11.11
1	0.381	7.41
2	0.508	5.56
3	0.635	4.44
4	0.762	3.70

## NITTO DENKO

### 19.1.6. Code128

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.125	18.47	36.95
1	0.25	9.24	18.47
2	0.375	6.16	12.32
3	0.5	4.62	9.24
4	0.625	3.69	7.39
5	0.75	3.08	6.16
6	0.875	2.64	2.64

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.254	9.09	18.18
1	0.381	6.06	12.12
2	0.508	4.55	9.09
3	0.635	3.64	7.27
4	0.762	3.03	6.06

### 19.1.7. Code49

ピケット&ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	数字のみ 密度 (CPI)	英数字 密度 (CPI)
0	0.25	117.57	71.12
1	0.375	78.38	47.47
2	0.5	58.78	35.56
3	0.625	47.03	28.45
4	0.75	39.19	1.71



NITTO DENKO

19.2. 7.6ドット/mmヘッド機

19.2.1. Code39

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.264	0.66	2.5:1	6.64
1	0.264	0.792	3.0:1	6.01
2	0.396	0.924	2.33:1	4.58
3	0.396	1.056	2.67:1	4.28
4	0.528	1.188	2.25:1	3.50

19.2.2. ITF

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.264	0.66	2.5:1	12.03
1	0.264	0.792	3.0:1	10.69
2	0.396	0.924	2.33:1	8.37
3	0.396	1.056	2.67:1	7.70
4	0.528	1.188	2.25:1	6.41

19.2.3. UPC/EAN/JAN

ピケットフェンスバーコード

ラダーバーコード

UMAG	モジュール寸法 (mm)	拡大率
0	0.264	80.0
1	0.33	100.0
2	0.396	120.0
3	0.462	140.0
4	0.528	160.0
5	0.594	180.0
6	0.66	200.0
7	0.726	220.0

UMAG	モジュール寸法 (mm)	拡大率
0	0.264	80.0
1	0.33	100.0
2	0.396	120.0
3	0.462	140.0
4	0.528	160.0

NITTO DENKO

19.2.4. CodaBar

ピケットフェンスバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.264	0.66	2.5:1	8.75
1	0.264	0.792	3.0:1	8.02
2	0.396	0.924	2.33:1	6.01
3	0.396	1.056	2.67:1	5.66
4	0.528	1.188	2.25:1	4.58
5	0.528	1.452	2.75:1	4.18

ラダーバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.264	0.66	2.5:1	8.75
1	0.264	0.792	3.0:1	8.02
2	0.396	0.924	2.33:1	6.01
3	0.396	1.056	2.67:1	5.66
4	0.528	1.188	2.25:1	4.58

19.2.5. Code93

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.132	21.39
1	0.264	10.69
2	0.396	7.13
3	0.528	5.35
4	0.66	4.28

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.264	10.69
1	0.396	7.13
2	0.528	5.35
3	0.66	4.28
4	0.792	3.56

## NITTO DENKO

### 19.2.6. Code128

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.132	17.49	34.99
1	0.264	8.75	17.49
2	0.396	5.83	11.66
3	0.528	4.37	8.75
4	0.66	3.50	7.00
5	0.792	2.92	5.83
6	0.924	2.50	5.00

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.264	8.75	17.49
1	0.396	5.83	11.66
2	0.528	4.37	8.75
3	0.66	3.50	7.00
4	0.792	2.92	5.83

### 19.2.7. Code49

ピケット&ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	数字のみ 密度 (CPI)	英数字 密度 (CPI)
0	0.264	111.33	67.35
1	0.396	74.22	44.90
2	0.528	55.67	33.67
3	0.66	44.53	26.94
4	0.792	37.11	22.45

NITTO DENKO

19.3. 12ドット/mm (11.8ドット/mm) ヘッド機

(DURA PRINTER LSP5300/LP5320/KP4300/KP3000)

19.3.1. Code39

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.167	0.417	2.5:1	10.5
1	0.167	0.5	3.0:1	9.5
2	0.25	0.583	2.33:1	7.2
3	0.25	0.666	2.67:1	6.7
4	0.33	0.75	2.25:1	5.5

19.3.2. ITF

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.167	0.417	2.5:1	19.1
1	0.167	0.5	3.0:1	17.0
2	0.25	0.583	2.33:1	13.3
3	0.25	0.666	2.67:1	12.2
4	0.33	0.75	2.25:1	10.2

19.3.3. UPC/EAN/JAN

ピケットフェンスバーコード

ラダーバーコード

UMAG	モジュール寸法 (mm)	拡大率
0	0.1666	50.5
1	0.208	63.1
2	0.2499	75.7
3	0.292	88.4
4	0.333	101.0
5	0.375	113.6
6	0.417	126.2
7	0.458	138.8

UMAG	モジュール寸法 (mm)	拡大率
0	0.166	50.5
1	0.208	63.1
2	0.249	75.7
3	0.292	88.4
4	0.333	101.0

NITTO DENKO

19.3.4. CodaBar

ピケットフェンスバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.166	0.416	2.5:1	13.86
1	0.166	0.499	3.0:1	12.71
2	0.2499	0.583	2.33:1	9.53
3	0.2499	0.666	2.67:1	8.97
4	0.333	0.749	2.25:1	7.26
5	0.333	0.916	2.75:1	6.63

ラダーバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.166	0.416	2.5:1	13.86
1	0.166	0.499	3.0:1	12.71
2	0.2499	0.583	2.33:1	9.53
3	0.2499	0.666	2.67:1	8.97
4	0.333	0.749	2.25:1	7.26

19.3.5. Code93

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.0833	33.88
1	0.166	16.94
2	0.2499	11.29
3	0.333	8.47
4	0.416	6.78

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.1666	16.94
1	0.249	11.29
2	0.333	8.47
3	0.416	6.78
4	0.499	5.65

NITTO DENKO

19.3.6. Code128

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.0833	27.72	55.44
1	0.1666	13.86	27.72
2	0.249	9.24	18.48
3	0.333	6.93	13.86
4	0.416	5.54	11.09
5	0.499	4.62	9.24
6	0.583	3.96	7.92

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.166	13.86	27.72
1	0.249	9.24	18.48
2	0.333	6.93	13.86
3	0.416	5.54	11.09
4	0.499	4.62	9.24

19.3.7. Code49

ピケット&ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	数字のみ 密度 (CPI)	英数字 密度 (CPI)
0	0.166	176.42	106.72
1	0.249	117.61	71.15
2	0.333	88.21	53.36
3	0.416	70.57	42.69
4	0.499	58.81	36.57

NITTO DENKO

19.4. 16ドット/mm (15.75ドット/mm) ヘッド機  
(DURA PRINTER SR/SRs)

19.4.1. Code39

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	6.90
1	0.254	0.762	3.0:1	6.25
2	0.381	0.889	2.33:1	4.76
3	0.381	1.016	2.67:1	4.44
4	0.508	1.143	2.25:1	3.64

19.4.2. ITF

ラダーバーコード

BCPI 番号	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	12.50
1	0.254	0.762	3.0:1	11.11
2	0.381	0.889	2.33:1	8.70
3	0.381	1.016	2.67:1	8.00
4	0.508	1.143	2.25:1	6.67

19.4.3. UPC/EAN/JAN

ピケットフェンスバーコード

ラダーバーコード

UMAG	モジュール寸法 (mm)	拡大率
0	0.254	77.0
1	0.3175	96.2
2	0.381	115.5
3	0.4445	134.7
4	0.508	153.9
5	0.5715	173.2
6	0.635	192.4
7	0.6985	211.7
8	0.127	38.5
9	0.191	48.1

UMAG	モジュール寸法 (mm)	拡大率
0	0.254	76.9
1	0.3175	96.2
2	0.381	115.5
3	0.4445	134.7
4	0.508	153.9
5	0.127	38.5
6	0.191	48.1

※1

※1

※1

※1

※1) DURA PRINTER SR ,ROM Ver 00.53 以上  
DURA PRINTER SRS ,ROM Ver 10.31 以上

NITTO DENKO

19.4.4. CodaBar

ピケットフェンスバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	9.09
1	0.254	0.762	3.0:1	8.33
2	0.381	0.889	2.33:1	6.25
3	0.381	1.016	2.67:1	5.88
4	0.508	1.143	2.25:1	4.76
5	0.508	1.397	2.75:1	4.35

ラダーバーコード

BCPI 値	細線幅 (mm)	太線幅 (mm)	比率 (W:N)	密度 (CPI)
0	0.254	0.635	2.5:1	9.09
1	0.254	0.762	3.0:1	8.33
2	0.381	0.889	2.33:1	6.25
3	0.381	1.016	2.67:1	5.88
4	0.508	1.143	2.25:1	4.76

19.4.5. Code93

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.127	22.2
1	0.254	11.11
2	0.381	7.41
3	0.508	5.56
4	0.635	4.44
10	0.191	14.81
11	0.318	8.89
12	0.445	6.35

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	密度 (CPI)
0	0.254	11.11
1	0.381	7.41
2	0.508	5.56
3	0.635	4.44
4	0.762	3.70
10	0.191	14.81
11	0.381	8.89
12	0.445	6.35



NITTO DENKO

19.4.6. Code128

ピケットフェンスバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.127	18.18	36.36
1	0.254	9.09	18.18
2	0.381	6.06	12.12
3	0.508	4.55	9.09
4	0.635	3.64	7.27
5	0.762	3.03	6.06
6	0.889	2.60	5.19
10	0.191	12.09	24.18
11	0.318	7.26	14.52
12	0.445	5.19	10.38

ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	A と B 密度 (CPI)	C 密度 (CPI)
0	0.254	9.09	18.18
1	0.381	6.06	12.12
2	0.508	4.55	9.09
3	0.635	3.64	7.27
4	0.762	3.03	6.06
10	0.191	12.09	24.18
11	0.318	7.26	14.52
12	0.445	5.19	10.38

19.4.7. Code49

ピケット&ラダーバーコードの場合

BCPI 番号	モジュール寸法 (mm)	数字のみ 密度 (CPI)	英数字 密度 (CPI)
0	0.191	93.3	154.3
1	0.245	70.0	115.7
2	0.318	56.0	92.6
3	0.318	46.7	77.1
4	0.445	40.0	66.1

## 20. モデル毎のドットフォントサイズ一覧表

### 20.1. 8ドット/mmヘッド機 (DURA PRINTER R)

DDF 値	フォント 名称	ドットサイズ 幅×高さ (dot)	ドットサイズ 幅×高さ (dot)	ドット (dot)	文字サイズ 幅×高さ (dot)	文字サイズ 幅×高さ (dot)	文字種
1	英数字 XS	5×9	0.63×1.14	2	5×7	0.63×0.89	英数字
2	英数字 S	7×14	0.88×1.78	3	7×11	0.88×1.40	英数字
3	英数字 M	10×20	1.25×2.54	4	10×16	1.25×2.03	英数字
4	英数字 L	15×30	1.88×3.81	6	15×24	1.88×3.05	英数字
5	英数字 SS	6×8	0.75×1.02	—	6×8	0.75×1.02	英大文字 数字
6	擬似 OCR-B	14×27	1.75×2.79	5	14×22	1.75×2.16	英数字
7	漢字(16)	16×16	2.00×2.03	—	16×16	2.00×2.03	漢字
8	漢字(24)	24×24	3.00×3.05	—	24×24	3.00×3.05	漢字
9	英数字 2バイト	16×16	2.00×2.03	—	16×16	2.00×2.03	英数字 2バイト
10	英数字 1バイト	16×16	2.00×2.03	—	16×16	2.00×2.03	英数字 1バイト
11	英数字 B	14×11	1.75×1.40	—	14×11	1.75×1.40	英数字
12	英数字 N1	10×16	1.25×2.03	3	10×13	1.25×1.65	英数字
13	英数字 N2	10×16	1.25×2.03	2	10×14	1.25×1.78	英数字

### 20.2. 7.6ドット/mmヘッド機 (DURA PRINTER SG)

DDF 値	フォント 名称	ドットサイズ 幅×高さ (dot)	ドットサイズ 幅×高さ (dot)	ドット (dot)	文字サイズ 幅×高さ (dot)	文字サイズ 幅×高さ (dot)	文字種
1	英数字 XS	5×9	0.66×1.18	2	5×7	0.66×0.92	英数字
2	英数字 S	7×14	0.92×1.84	3	7×11	0.92×1.45	英数字
3	英数字 M	10×20	1.32×2.63	4	10×16	1.32×2.10	英数字
4	英数字 L	15×30	1.98×3.95	6	15×24	1.98×3.16	英数字
5	英数字 SS	6×8	0.79×1.05	—	6×8	0.79×1.05	英大文字 数字
6	擬似 OCR-B	14×27	1.85×2.89	5	14×22	1.85×2.24	英数字
7	漢字(16)	16×16	2.11×2.11	—	16×16	2.11×2.11	漢字
8	漢字(24)	24×24	3.17×3.17	—	24×24	3.17×3.17	漢字
9	擬似 OCR-A	15×22	1.98×2.89	—	15×22	1.98×2.89	英数字
10	英数字 1バイト	16×16	2.11×2.11	—	16×16	2.11×2.11	英数字 1バイト

## 20.3. 12ドット/mmヘッド機 (DURA PRINTER LSP5300/LP5320/KP4300/KP3000)

DDF 値	フォント 名称	ドットサイズ 幅×高さ (dot)	ドットサイズ 幅×高さ (dot)	ディセンダ (dot)	文字サイズ 幅×高さ (dot)	文字サイズ 幅×高さ (dot)	文字種
1	英数字 XS	5×9	0.42×0.75	2	5×7	0.42×0.58	英数字
2	英数字 S	7×14	0.58×0.17	3	7×11	0.58×0.92	英数字
3	英数字 M	10×20	0.83×1.67	4	10×16	0.83×1.33	英数字
4	英数字 L	15×30	1.25×2.50	6	15×24	1.25×2.00	英数字
5	英数字 SS	6×8	0.50×0.67	—	6×8	0.50×0.67	英大文字 数字
6	擬似 OCR-B	14×27	1.17×1.83	5	14×22	1.17×1.83	英数字
7	漢字(16)	16×16	1.33×1.33	—	16×16	1.33×1.33	漢字
8	漢字(24)	24×24	2.00×2.00	—	24×24	2.00×2.00	漢字
9	英数字 2バイト	16×16	1.33×1.33	—	16×16	1.33×1.33	英数字 2バイト
10	英数字 1バイト	16×16	1.33×1.33	—	16×16	1.33×1.33	英数字 1バイト
11	英数字 B	14×11	1.17×0.92	—	14×11	1.17×0.92	英数字
12	英数字 N1	10×16	0.83×1.33	3	10×13	0.83×1.08	英数字
13	英数字 N2	10×16	0.83×1.33	2	10×14	0.83×1.17	英数字
14	英数字 NO1	12×20	1.00×1.67	2	11×18	0.92×1.50	英数字
15	英数字 NO2	12×20	1.00×1.67	—	12×20	1.00×1.67	英数字
16	英数字 NO3	15×24	1.25×2.00	2	15×22	1.25×1.83	英数字
17	英数字 NO4	20×32	1.67×2.67	2	20×30	1.67×2.50	英数字
18	英数字 NO5	24×36	2.00×3.00	2	24×34	2.00×2.83	英数字
19	英数字 NA1	16×20	1.33×1.67	2	16×18	1.33×1.50	英数字
20	英数字 NA2	20×28	1.67×2.33	4	20×24	1.67×2.00	英数字

\*LP5320/KP4300/KP3000 は、DDF 値は、1 から 10 までで、11 以降はありません。

また、DDF9 は、擬似 OCR-A フォントでサイズは下記の様になります。

9	擬似 OCR-A	15×22	1.25×1.83	—	15×22	1.25×1.83	英数字
---	-------------	-------	-----------	---	-------	-----------	-----

## 20.4. 16ドット/mmヘッド機 (DURA PRINTER SR,SRs)

DDF 値	フォント 名称	ドットサイズ 幅×高さ (dot)	ドットサイズ 幅×高さ (dot)	ドット (dot)	文字サイズ 幅×高さ (dot)	文字サイズ 幅×高さ (dot)	文字種
1	英数字 XS	5×9	0.32×0.57	2	5×7	0.32×0.44	英数字
2	英数字 S	7×14	0.44×0.89	3	7×11	0.44×0.70	英数字
3	英数字 M	10×20	0.64×1.27	4	10×16	0.64×1.02	英数字
4	英数字 L	15×30	0.95×1.91	6	15×24	0.95×1.52	英数字
5	英数字 SS	6×8	0.38×0.51	—	6×8	0.38×0.51	英大文字 数字
6	擬似 OCR-B	14×27	0.89×1.71	5	14×22	0.89×1.08	英数字
7	漢字(16)	16×16	1.02×1.02	—	16×16	1.02×1.02	漢字
8	漢字(24)	24×24	1.52×1.52	—	24×24	1.52×1.52	漢字
9	英数字 2バリエーション	16×16	1.02×1.02	—	16×16	1.02×1.02	英数字 2バリエーション
10	英数字 1バリエーション	16×16	1.02×1.02	—	16×16	1.02×1.02	英数字 1バリエーション
11	英数字 B	14×11	0.89×0.70	—	14×11	0.89×0.70	英数字
12	英数字 N1	10×16	0.64×1.02	3	10×13	0.64×0.83	英数字
13	英数字 N2	10×16	0.64×1.02	2	10×14	0.64×0.89	英数字
14	英数字 NO1	12×20	0.76×1.27	2	11×18	0.76×1.14	英数字
15	英数字 NO2	12×20	0.76×1.27	—	12×20	0.76×1.27	英数字
16	英数字 NO3	15×24	0.95×1.52	2	14×22	0.95×1.40	英数字
17	英数字 NO4	20×32	1.27×2.03	2	19×30	1.27×1.91	英数字
18	英数字 NO5	24×36	1.52×2.29	2	23×34	1.52×2.16	英数字
19	英数字 NA1	16×20	1.02×1.27	2	15×18	1.02×1.14	英数字
20	英数字 NA2	20×28	1.27×1.78	4	20×24	1.27×1.52	英数字

## 21. フォント選択表

---

各プリンタ毎のフォント選択表を以下に示します。

### 21.1. Dura Printer R

フォントの種類	
英数字 XS	1
英数字 SS	5
英数字 S	2
英数字 M	3
英数字 L	4
疑似 OCR-B	6
漢字(16)	7
漢字(24)	8
英数字 2 バイバカ	9
英数字 1 バイバカ	10
英数字 B	11
英数字 N1	12
英数字 N2	13

国別コード	
アメリカ	1
イギリス	3
フランス	4
スウェーデン・フィンランド	5
デンマーク	6
イタリア	7
ドイツ	8
スペイン	9
日本	10

### 21.2. Dura Printer SG/LP5320/KP4300/KP3000

フォントの種類	
英数字 XS	1
英数字 SS	5
英数字 S	2
英数字 M	3
英数字 L	4
疑似 OCR-B	6
漢字(16)	7
漢字(24)	8
OCR-A	9
英数字 1 バイバカ	10

国別コード	
アメリカ	1
イギリス	3
フランス	4
スウェーデン・フィンランド	5
デンマーク	6
イタリア	7
ドイツ	8
スペイン	9
日本	10

NITTO DENKO

21.3. Dura Printer SR、SR s、LSP5300

フォントの種類	
英数字 XS	1
英数字 SS	5
英数字 S	2
英数字 M	3
英数字 L	4
疑似 OCR-B	6
漢字(26)	7
漢字(24)	8
英数字 2 バイ付	9
英数字 1 バイ付	10
英数字 B	11
英数字 N1	12
英数字 N2	13
英数字 NO1	14
英数字 NO2	15
英数字 NO3	16
英数字 NO4	17
英数字 NO5	18
英数字 NA1	19
英数字 NA2	20

国別コード		チェックデジット付き	
アメリカ	1	code39 MOD43	17
イギリス	3	ITF MOD10Wait3	33
フランス	4	codabar MOD10	49
スウェーデン・フィンランド	5	codabar MOD11	65
デンマーク	6	codabar MOD16	81
イタリア	7	codabar MOD10&11	97
ドイツ	8	case code128	113
スペイン	9		
日本	10		

※SR (RSS) ,SRS (RSS) では  
 英数字 NO3,NO4,NO5,NA1,NA2 は印字できません。

## 22. バーコード選択表

BSYM、BDEF の設定値と対応機種一覧表を以下に示します。

### 22.1. BSMYの設定値

バーコードの種類	設定値	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS
Code39	1	○	○	○	○	○
Codabar	3	○	○	○	○	○
Code39(MOD43 付き)	5	○	○	○	○	○
ITF	8	○	○	○	○	○
UPC-A	10	○	○	○	○	○
UPC-A 2 Char.	11	○	○	○	○	○
UPC-A 5 Char.	12	○	○	○	○	○
UPC-E	13	○	○	○	○	○
UPC-E 2 Char.	14	○	○	○	○	○
UPC-E 5 Char.	15	○	○	○	○	○
EAN13	16	○	○	○	○	○
EAN8	17	○	○	○	○	○
EAN13 2 Char.	18	○	○	○	○	○
EAN13 5 Char.	19	○	○	○	○	○
EAN8 2 Char.	20	○	○	○	○	○
EAN8 5 Char.	21	○	○	○	○	○
Code128	25	○	○	○	○	○
Cascode Code128	27	○	○	○	○	○
Code93	30	○	○	○	○	○
ITF (MOD10 Wait 3 付き)	52	×	○	○	○	○
Codabar (MOD16 付き)	54	×	○	○	○	○
UPC-A (HRI なし)	110	×	○	○	○	○
UPC-A 2 Char. (HRI なし)	111	×	○	○	○	○
UPC-A 5 Char. (HRI なし)	112	×	○	○	○	○
UPC-E (HRI なし)	113	×	○	○	○	○
UPC-E 2 Char. (HRI なし)	114	×	○	○	○	○
UPC-E 5 Char. (HRI なし)	115	×	○	○	○	○
EAN13 (HRI なし)	116	×	○	○	○	○
EAN8 (HRI なし)	117	×	○	○	○	○
EAN13 2 Char. (HRI なし)	118	×	○	○	○	○
EAN13 5 Char. (HRI なし)	119	×	○	○	○	○
EAN8 2 Char. (HRI なし)	120	×	○	○	○	○
EAN8 5 Char. (HRI なし)	121	×	○	○	○	○
EAN13 (チェックデジット計算なし)	216	×	○	×	○	○
EAN8 (チェックデジット計算なし)	217	×	○	×	○	○

## 22.2. BDEFの設定値

バーコードの種類	設定値	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS
Code39	1	○	○	○	○	○
Code39 ラダー	2	○	○	○	○	○
Codabar	3	○	○	○	○	○
Codabar ラダー	4	○	○	○	○	○
Code39(MOD43 付き)	5	○	○	○	○	○
同上ラダー	6	○	○	○	○	○
ITF	8	○	○	○	○	○
ITF ラダー	9	○	○	○	○	○
UPC-A	10	○	○	○	○	○
UPC-A 2 Char.	11	○	○	○	○	○
UPC-A 5 Char.	12	○	○	○	○	○
UPC-E	13	○	○	○	○	○
UPC-E 2 Char.	14	○	○	○	○	○
UPC-E 5 Char.	15	○	○	○	○	○
EAN13	16	○	○	○	○	○
EAN8	17	○	○	○	○	○
EAN13 2 Char.	18	○	○	○	○	○
EAN13 5 Char.	19	○	○	○	○	○
EAN8 2 Char.	20	○	○	○	○	○
EAN8 5 Char.	21	○	○	○	○	○
Code128	25	○	○	○	○	○
Code128 ラダー	26	○	○	○	○	○
Casecode Code128	27	○	○	○	○	○
同上ラダー	28	○	○	○	○	○
Code93	30	○	○	○	○	○
Code93 ラダー	31	○	○	○	○	○
UPC-A ラダー	40	○	○	○	○	○
UPC-A 2 Char. ラダー	41	○	○	○	○	○
UPC-A 5 Char. ラダー	42	○	○	○	○	○
UPC-E ラダー	43	○	○	○	○	○
UPC-E 2 Char. ラダー	44	○	○	○	○	○
UPC-E 5 Char. ラダー	45	○	○	○	○	○
EAN 13 ラダー	46	○	○	○	○	○
EAN 8 ラダー	47	○	○	○	○	○
EAN 13 2 Char. ラダー	48	○	○	○	○	○
EAN 13 5 Char. ラダー	49	○	○	○	○	○
EAN 8 2 Char. ラダー	50	○	○	○	○	○
EAN 8 5 Char. ラダー	51	○	○	○	○	○
ITF (MOD10 Wait 3 付き)	52	×	○	○	○	○
ITF (MOD10 Wait 3 付き)ラダー	53	×	○	○	○	○
Codabar (MOD16 付き)	54	×	○	○	○	○
Codabar (MOD16 付き)ラダー	55	×	○	○	○	○



NITTO DENKO

バーコード種類	設定値		SR	SG	LSP5300 /KP4300 /KP3000	SRS
UPC-A (HRI なし)	110	×	○	○	○	○
UPC-A 2 Char. (HRI なし)	111	×	○	○	○	○
UPC-A 5 Char. (HRI なし)	112	×	○	○	○	○
UPC-E (HRI なし)	113	×	○	○	○	○
UPC-E 2 Char. (HRI なし)	114	×	○	○	○	○
UPC-E 5 Char. (HRI なし)	115	×	○	○	○	○
EAN13 (HRI なし)	116	×	○	○	○	○
EAN8 (HRI なし)	117	×	○	○	○	○
EAN13 2 Char. (HRI なし)	118	×	○	○	○	○
EAN13 5 Char. (HRI なし)	119	×	○	○	○	○
EAN8 2 Char. (HRI なし)	120	×	○	○	○	○
EAN8 5 Char. (HRI なし)	121	×	○	○	○	○
UPC-A ラダー (HRI なし)	140	×	○	○	○	○
UPC-A 2 Char. ラダー (HRI なし)	141	×	○	○	○	○
UPC-A 5 Char. ラダー (HRI なし)	142	×	○	○	○	○
UPC-E ラダー (HRI なし)	143	×	○	○	○	○
UPC-E 2 Char. ラダー(HRI なし)	144	×	○	○	○	○
UPC-E 5 Char.ラダー (HRI なし)	145	×	○	○	○	○
EAN13 ラダー(HRI なし)	146	×	○	○	○	○
EAN8 ラダー(HRI なし)	147	×	○	○	○	○
EAN13 2 Char.ラダー (HRI なし)	148	×	○	○	○	○
EAN13 5 Char.ラダー (HRI なし)	149	×	○	○	○	○
EAN8 2 Char.ラダー (HRI なし)	150	×	○	○	○	○
EAN8 5 Char.ラダー (HRI なし)	151	×	○	○	○	○
EAN13 (チェックデジット計算なし)	216	×	○	○	○	○
EAN8 (チェックデジット計算なし)	217	×	○	○	○	○
EAN13 ラダー(チェックデジット計算なし)	246	×	○	○	○	○
EAN8 ラダー (チェックデジット計算なし)	247	×	○	○	○	○

## 23. 2次元シンボルの選択表

DTDS の設定値と対応機種一覧表

シンボル種類	設定値	R	SR	SG	LSP5300 /KP4300 /KP3000	SR
QR コード タイプ 1	100	×	○	○	○	○
QR コード タイプ 2	101	×	○	○	○	○
Micro QR コード	102	×	○	○	○	○
Code49	60	×	○	○	○	○
PDF417 標準タイプ	20	×	○	×	○	○
トランケート PDF417 切り詰めタイプ	21	×	○	×	○	○

DURA PRINTER シリーズでは、上表のように QR コード、Code49 のほかに PDF417、Data Matrix を印刷することができるモデルもあります。ここでは 9 章、2 次元シンボルのところでは、説明をしなかった、PDF417 用のコマンドについて説明します。

### 23.1. 2次元シンボル用コマンド

2 次元シンボル用コマンドの補足編として PDF417 のシンボルを印字する為のコマンドについて説明をします。

#### 注

QR CODE と CODE49 を印字する為のコマンドについては、第 9 章の 2 次元シンボルの章をご覧ください。

PDF417 を印字する為には、少なくとも 3 つのコマンドを必要とします。

それは、2 次元シンボル選択 (Define Two Dimensional Code:DTDC)、シンボルデータ開始 (BarCode StarT:BCST)、バーコードデータ終了 (Barcode SToP:BSTP) で説明します。

PDF417 には、シンボルのセルサイズ (TDCS)、シンボルの高さ (TDCH)、エラー修復レベル (TDEL)、データ入力モード設定 (TDDM) コマンドが必要です。

VeriCode には、シンボルのセルサイズ設定 (TDCS)、エラー修復レベル (TDEL)、データ入力モード設定 (TDDM) コマンドが必要です。

#### 注

プログラマーは、使用する 2 次元シンボルの規格を調査し、またそこに定められた指針に従う責任があります。当社は RCL を使って作成した 2 次元シンボルが、いずれか特定の規格を満足することを保証するものではありません。

## NITTO DENKO

### 23.1.1. 2次元シンボルの選択 (Define Two Dimensional Symbol : DTDS)

DTDS コマンドは、必ず必要な 2次元シンボル選択コマンドで、使用する 2次元シンボルの種類と印刷方向を設定するために使います。(QR Code の方向は常に 0° です。)

2次元シンボルの種類は、2次元シンボルを生成する前に選択しなければなりません。

このコマンドの構文は次のとおりです。

**DTDS;A;B;**

**A**=2次元シンボルの種類を決める整数、下の表によって決まる。

**B**=2次元シンボルの方向を設定する 1~4 の整数

(QR Code では、**B** の値は意味を持ちません)

設定値は下表のようになります。

シンボル種類	A の値
QR コード モデル 1	100
QR コード モデル 2	101
Micro QR コード	102
Code49	60
PDF417	20
PDF417 トランケートタイプ	21

回転方向	B の値
0 度	1
90 度	2
180 度	3
270 度	4

## 23.2. PDF417

**PDF417** シンボルは、シンボルの選択コマンド **DTDS** とシンボルの開始／終了を示す **BCST**、**BSTP** に加えて、シンボルのセルサイズ設定 (**TDCS**)、シンボルの高さ設定 (**TDCH**)、エラー修復レベル (**TDEL**)、データ入力モード設定 (**TDDM**) コマンドが必要です。**PDF417** は、バーコードと同じくイメージの展開位置が、左下になります。

### 23.2.1. PDF417 バー幅設定 (Two Dimensional symbol Cell Width : TDCW)

**TDCW** コマンドは、**PDF417** のナローバーのモジュール幅を設定するコマンドです。

このコマンドの構文は次のとおりです。

**TDCW;X;**

**X**=セルのドット数を設定する整数。

デフォルト値は、**2** ドットです。

### 23.2.2. PDF417 バー高さ設定 (Two Dimensional symbol Cell Height : TDCH)

**TDCH** コマンドは、**PDF417** シンボルの **1** 行あたりのシンボル高さを設定するコマンドです。**PDF417** は、**3** 行から **90** 行でシンボルの展開ができます。

シンボルとしての高さは、シンボルの行数：**R**、バー高さ：**H** より

シンボル高さ = **R** × **H** となります。

このコマンドの構文は次のとおりです。

**TDCH;Y;**

**Y**=ドット数で指定。

### 23.2.3. PDF417 エラー修復レベル設定 (Two Dimensional symbol Error Level : TDEL)

**TDEL** コマンドは、**PDF417** のエラー修復レベルの設定を行うコマンドです。

**PDF417** には、エラーの修復率により以下の **9** つのレベルがあります。

なお、エラー修復レベルが高くなればそれだけ修復の為のデータが増えます。

このコマンドの構文は次のとおりです。

**TDEL;L;**

**L**=次の表のエラーレベルの **1** つ

L	修復レベル
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8

設定値が上記以外の場合には、デフォルト値となります。  
デフォルト値は **0** : レベル **0** です。

#### 23.2.4. PDF417 マトリックス (行、列) サイズ設定 (Two Dimensional symbol Matrix Size : TDMS)

PDF417 は、行数及び **1** 行の列数をコマンドにより設定することができます。設定の中には、プリンタ自身にサイズを決める設定とユーザーの任意のマトリックスサイズに設定する方法があります。

しかし、ユーザー任意のサイズを指定した場合、入力されたデータが指定のマトリックスサイズに格納できない場合は、コマンドエラーとなりますので十分に注意してください。

このコマンドの構文は次のとおりです。

**TDMS;R;C;**

**R**=行数 (縦セル数)

**C**=1 行のデータ数 (横セル数)

行数、列数設定を以下に示します。

- ・ 行数、列数の設定値によりプリンタは以下のように **PDF417** の展開を行います。
- ① 行数または列数のいずれかにマイナスの値が設定された時及び、両方 **0** の場合は、自動設定モードとしてプリンタ自身が行数、**1** 行のデータ数を決定して印字を行います。このモードがデフォルト値となります。
- ② 列数が **0** で行数が整数値の場合は行数を設定値に固定して、データが格納可能なように **1** 行あたりのデータ数を決定して印字を行います。(行数は **3** から **90**)
- ③ 数が **0** で列数が整数値の場合は、**1** 行のデータ数を列数の設定値に固定して、データが格納可能なように行数を決定して印字を行います。(列数は **1** から **30**)
- ④ 行数と列数ともに値が設定された場合は、指定された行数、**1** 行のデータ数でシンボルを印字します。なお、指定サイズにデータが格納できない場合はコマンドエラーとします。

## NITTO DENKO

### 23.2.5. PDF417 の印字データの入力方法

PDF417 では、データとして用いることのできる入力文字の種類は、データ入力モードプリンタはこの識別子によりデータの種類を判別してエンコード処理を行います。

### 23.2.6. PDF417 の使用例

次に PDF417 の印刷例を示します。

```
~ ^ "PDFSAMPLE";1;0;150;200;  
SPB;UTOF;1500;HBR;0;VBR;0;  
DDF;8;10;DFM;2;2;  
HBR;5;VBR;85;"PDF417 の印字サンプル";  
DTDS;20;1;TDEL;3;TDCH;20;TDCW;3;  
HBR;80;VBR;50;  
BCST;"PDF417 の印字サンプル";BSTP;  
TRM;¥
```



PDF417 の印字サンプル

## 24. ステータス応答フォーマット一覧

要求コマンド	ビット	SG/KP4300/KP3000		R	
		ENQ (05H)	STX (02H)	ENQ (05H)	STX (02H)
ヘッダ	7 ～ 0	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ
ステータス 1	7 6 5 4 3 2 1 0	常に <b>0</b> 一時停止 ハードウェアエラー 通信エラー カバーオープン カッターエラー サブライエラー コマンドエラー	常に <b>0</b> 一時停止 ハードウェアエラー 通信エラー カバーオープン カッターエラー サブライエラー コマンドエラー	常に <b>0</b> 一時停止 ハードウェアエラー 通信エラー カバーオープン カッターエラー サブライエラー コマンドエラー	常に <b>0</b> 一時停止 ハードウェアエラー 通信エラー カバーオープン カッターエラー サブライエラー コマンドエラー
ステータス 2	7 6 5 4 3 2 1 0		未定義常に <b>0</b> 剥離位置ラベル有 未定義常に <b>0</b> 印字中 コマンドデータ有	常に <b>03Hex</b> フッター	未定義 常に <b>0</b>  バッファ残 50%以下 印字中 コマンドデータ有
ステータス 3	7 ～ 0		印字残り枚数 上位桁 バイナリで 送信		常に <b>03Hex</b> フッター
ステータス 4	7 ～ 0		印字残り枚数 中位桁 バイナリで 送信		
ステータス 5	7 ～ 0		印字残り枚数 下位桁 バイナリで 送信		

NITTO DENKO

要求コマンド	ビット	SR		SRS	
		ENQ (05H)	STX (02H)	ENQ (05H)	STX (02H)
ヘッダ バイト	7 ～ 0	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ
ステータス 1 バイト	7 6 5 4 3 2 1 0	常に <b>0</b> 一時停止 ハートウェアエラー 通信エラー カバーオープン ボルトエラー サプライエラー コマンドエラー	常に <b>0</b> 一時停止 ハートウェアエラー 通信エラー カバーオープン ボルトエラー サプライエラー コマンドエラー	常に <b>0</b> 一時停止 ハートウェアエラー 通信エラー カバーオープン カッターエラー サプライエラー コマンドエラー	常に <b>0</b> 一時停止 ハートウェアエラー 通信エラー カバーオープン カッターエラー サプライエラー コマンドエラー
ステータス 2 バイト	7 6 5 4 3 2 1 0	常に <b>03Hex</b> フッター	未定義常に <b>0</b> 剥離位置ラベル有 未定義常に <b>0</b> 機能設定中 バッファ残 <b>50%</b> 以下 印字中 コマンドデータ有	常に <b>03Hex</b> フッター	未定義常に <b>0</b> 剥離位置ラベル有 未定義常に <b>0</b> 機能設定中 バッファ残 <b>50%</b> 以下 印字中 コマンドデータ有
ステータス 3 バイト	7 ～ 0		常に <b>03Hex</b> フッター		常に <b>03Hex</b> フッター



NITTO DENKO

		LSP5300/LP5320	
要求コマンド	ビット	ENQ (05H)	STX (02H)
ヘッダ バイト	7 ~ 0	常に <b>02Hex</b> ヘッダ	常に <b>02Hex</b> ヘッダ
ステータス 1 バイト	7 6 5 4 3 2 1 0	常に <b>0</b> 一時停止 ハートウェアエラー 通信エラー ハットオープン カッターエラー サブライエラー コマンドエラー	常に <b>0</b> 一時停止 ハートウェアエラー 通信エラー ハットオープン カッターエラー サブライエラー コマンドエラー
ステータス 2 バイト	7 6 5 4 3 2 1 0	常に <b>03Hex</b> フッター	未定義 常に <b>0</b> 剥離位置ラベル有 ラベルニアンプティ リボンエンド バッファ残 50%以下 印字中 コマンドデータ有
ステータス 3 バイト	7 ~ 0	X	常に <b>03Hex</b> フッター

## 25. コマンド対応一覧表

表中の記号は、以下のようになっています。

○ は、対応

× は、未対応でコマンドが入力されるとコマンドエラーとなる。

△ は、機能的に対応していないのでコマンド自体が単に無視される。

機能：カーソル移動	シーモニック							
		R	SR	SG	LSP5300 /KP4300 /KP3000	SRS		
垂直相対位置移動	VPR	○	○	○	○	○		
水平相対位置移動	HPR	○	○	○	○	○		
絶対垂直位置指定	VBR	○	○	○	○	○		
絶対水平位置指定	HBR	○	○	○	○	○		
カーソルを HBR、VBR による最後の指定位置へ移動	HOME	○	○	○	○	○		
カーソルを HBR により最後の指定位置に移動	EOL	○	○	○	○	○		

機能：バーコード設定	シーモニック							
		R	SR	SG	LSP5300 /KP4300 /KP3000	SRS		
バーコードデータ終了	BSTP	○	○	○	○	○		
バーコードデータ開始	BCST	○	○	○	○	○		
バーの細線幅設定	BNEW	○	○	○	○	○		
バーの太線幅設定	BWEW	○	○	○	○	○		
バーコード連番保存長さ設定	BSAL	○	○	○	○	○		
バーコード種類設定 1	BDEF	○	○	○	○	○		
バーコードシンボル高さ	BCSH	○	○	○	○	○		
バーコードインターキャラクターギャップ指定	BICG	○	○	○	○	○		
検証オン (バーコード単位で指定可)	APON	○	○	△	△	△		
検証オフ (バーコード単位で指定可)	APOF	○	○	△	△	△		
ハーフドットオン (白バーに付加)	HALF	○	○	○	○	○		
ハーフドットオン (黒バーに付加)	HLFB	×	○	×	○	○		
ハーフドットオフ	HOFF	○	○	○	○	○		
UPC/EAN/JAN のバー拡大率指定	UMAG	○	○	○	○	○		
バーコードの密度設定	BCPI	○	○	○	○	○		
検証レベル設定	AQL	○	○	△	△	△		
検証 NG 判断なし	DVfy	○	○	△	△	△		
バーコード指定 2	BSYM	○	○	○	○	○		
マルチボイド枚数設定	NUMV	○	○	△	△	△		
検証のモード設定、リスキャンギャップ	VFYM	×	○	△	△	△		

機能：ドットフォント設定	シーモニック							
		R	SR	SG	LSP5300 /KP4300 /KP3000	SRS		
ドットフォント指定	DDF	○	○	○	○	○		
ドットフォント倍率設定	DFM	○	○	○	○	○		
ドットフォント文字間設定	DFS	○	○	○	○	○		
ドットフォント印字方向指定	DFO	○	○	○	○	○		
外字フォントファイル指定	FNTF	○	○	○	○	○		
半角コード (内部処理用)		×	○	○	○	○		

NITTO DENKO

機能：ベクトルフォント設定 (アウトラインフォント含む)							
	モニター	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS	
ベクトルフォント指定	DHR	○	○	○	○	○	
ベクトルフォントキャラクター高さ指定	DCH	○	○	○	○	○	
ベクトルフォントキャラクター幅指定	DCW	○	○	○	○	○	
ベクトルフォントのスペース間設定	ICS	○	○	○	○	○	
アウトラインフォントサイズ設定	DCS	○	○	○	○	○	

機能：ボックス描画							
	モニター	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS	
囲みボックス	DBOX	○	○	○	○	○	
黒塗りボックス描画	DBBX	○	○	○	○	○	
白塗りボックス描画	DWBX	○	○	○	○	○	
反転ボックス描画	DCBX	○	○	○	○	○	

機能：ライン描画							
	モニター	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS	
水平線太さ	HLT	○	○	○	○	○	
垂直線太さ	VLT	○	○	○	○	○	
垂直線描画	DVL	○	○	○	○	○	
水平線描画	DHL	○	○	○	○	○	
斜線描画	DDL	○	○	○	○	○	

機能：2次元シンボル設定							
	モニター	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS	
QRコードエラー訂正レベル指定	QREL	×	○	○	○	○	
QRコードセルサイズ指定	QRCS	×	○	○	○	○	
QRコードマスクパターン指定	QRMP	×	○	○	○	○	
2次元シンボル指定	DTDS	×	○	×	○	○	
2次元シンボルセル幅指定	TDCW	×	○	×	○	○	
2次元シンボル高さ指定(PDF417)	TDCH	×	○	×	○	○	
2次元シンボルエラー訂正レベル設定	TDEL	×	○	○	○	○	
2次元シンボルマトリックスサイズ設定	TDRC	×	○	×	○	○	
2次元シンボルデータ入力モード設定	TDDM	×	○	×	○	○	

機能：連番設定							
	モニター	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS	
バーコードの連番インク・ディクリメントの刻み設定	BCID	○	○	○	○	○	
バーコード連番リピート枚数設定	BCLC	○	○	○	○	○	
連番、アルファベットモード (A-Z)	ALPH	○	○	○	○	○	
連番数字モード (0-9)	NUM	○	○	○	○	○	
連番、数字アルファベットモード (0-9, A-Z)	BOTH	○	○	○	○	○	
テキストのインク・ディクリメントの刻み設定	IDF	○	○	○	○	○	
テキストの連番長さ指定	SAL	○	○	○	○	○	
コマンド解析繰り返し先頭	MRK	○	○	○	○	○	
コマンド解析繰り返し終了	RET	○	○	○	○	○	
可変エリア桁位置、桁数指定	VLP	○	○	○	○	○	
バーコード連番桁位置、桁数指定	BVLP	○	○	○	○	○	
連番削除文字設定	EXCP	○	○	○	○	○	
連番自動消去モード	EMON	×	×	×	○	○	

NITTO DENKO

機能：ロゴ設定、描画								
	ニモニック	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS		
ロゴ描画1	LOGO	○	○	○	○	○		
ロゴ描画2	LOGD	×	×	×	○	○		
ロゴファイル指定	LOGF	×	○	○	○	○		
描画ロゴ番号指定	XLOG	×	○	○	○	○		
ロゴ直接描画（内部処理）		×	×	×	○	○		

機能：動作設定								
	ニモニック	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS		
180度回転印字指示	FLIP	○	○	○	○	○		
描画開始ターミネーター	BRK	○	○	○	○	○		
バッファ処理開始(イメージバッファクリア、パラメータ初期値)	SPB	○	○	○	○	○		
バッファ処理再開(イメージバッファそのまま、パラメータも)	RSPB	○	○	○	○	○		
描画印字開始ターミネーター	TRM	○	○	○	○	○		
メモ리카ード可変データエリア	FLD	○	○	○	○	○		
カッターオン	CTON	○	△	○	○	○		
カッターオフ	CTOF	○	△	○	○	○		
バッチカット枚数	NUBC	○	△	○	○	○		
エラー発生での再印字なし	NORP	○	○	×	○	○		
ラベルピッチ長設定	UTOF	○	○	○	○	○		

機能：プリンタ制御								
	ニモニック	R	SR	SG	LSP5300 /KP4300 /KP3000	SRS		
ステータス1 (05Hex)	<ENQ>	○	○	○	○	○		
ステータス2 (02Hex)	<STX>	○	○	○	○	○		
プリンタ情報	INFO	×	×	×	○	×		
走行距離リセット	RSET	×	×	×	○	×		
一時停止オン (12Hex)	<DC2>	○	○	○	○	○		
一時停止オフ (14Hex)	,<DC4>	○	○	○	○	○		
プリンタ初期設定 (18Hex)	<CAN>	○	○	○	○	○		
メモ리카ード書き込み開始 (1CHex)	<FS>	○	○	○	○	○		
メモ리카ード書き込み終了 (19Hex)	<EM>	○	○	○	○	○		