ソフトウェア開発者マニュアル

FBPL コマンドレファレンス

TD-4420TN/4520TN

TD-4650TNWB/4750TNWB/4750TNWBR

RJ-2035B/3035B

TJ-4021TN/4121TN/4121TNR

TJ-4420TN/4520TN/4620TN

Version 10

著作権

© 2019 Brother Industries, Ltd. All rights reserved.

本文書の情報は予告無く変更されることがあります。本文書に記載されているソフトウェアは使用許諾契約に従って提供されています。ソフトウェアは、これらの契約条項に従ってのみ使用またはコピーできます。本文書のいかなる部分も、ブラザー工業株式会社の書面による事前の許可なしに、いかなる形式または手段によっても複製することはできません。

商標

ブラザー製品および関連資料等に記載されている社名及び商品名はそれぞれ各社の商標または登録商標です。

重要なお知らせ

本資料は、お客様が直接ラベルプリンター（対応機種については、「4. 対応機種」をご参照ください）を制御される場合に必要な情報を提供するものです。一部の機能は機種によってご利用いただけない場合があります。

お客様が以下の内容にご同意いただいた場合のみ、本資料のご利用が可能です。もしご同意いただけない場合は、お客様は本資料をご利用いただけません。

無保証

a. 対応モデルおよび本資料のバージョンアップや修正等はブラザーが任意で行うものとします。

b. ブラザーは本資料に関し、明示または黙示であるかを問わず、瑕疵がないこと、特定の目的に適合することを含め、その他いかなる保証も行いません。

c. ブラザーは本資料および本資料に基づきお客様が作成したプログラムに起因して発生した直接的または間接的損害について、お客様に対し、いかなる責任も負わないものとします。

目次

1 コマンドの見方 1
2 表記の説明 2
3 印字位置 3
4 対応機種 4
5 セットアップとシステムコマンド 5
5.1 SIZE 5
5.2 GAP 6
5.3 GAPDETECT 8
5.4 BLINEDETECT 9
5.5 AUTODETECT 10
5.6 BLINE 11
5.7 OFFSET 12
5.8 SPEED 13
5.9 DENSITY 14
5.10 DIRECTION 15
5.11 REFERENCE 16
5.12 SHIFT 17
5.13 COUNTRY 18
5.14 CODEPAGE 19
5.15 CLS 21
5.16 FEED 22
5.17 BACKFEED 23
5.18 FORMFEED 24
5.19 HOME 25
5.20 PRINT 26
5.21 SOUND 27
5.22 CUT 28
5.23 LIMITFEED 29
5.24 SELFTEST 30
5.25 EOJ 33
5.26 DELAY 34
5.27 DISPLAY 35
5.28 INITIALPRINTER 37
5.29 MENU 38
6 ラベルフォーマットコマンド 41
6.1 BAR 41
6.2 BARCODE 42
6.3 TLC39 49
6.4 BITMAP 50
6.5 BOX 52
6.6 CIRCLE 53
6.7 ELLIPSE 54
6.8 CODABLOCK F MODE 55
6.9 DMATRIX 56
6.10 ERASE 58
6.11 MAXICODE 59
6.12 PDF417 61
6.13 AZTEC 64
6.14 MPDF417 65
6.15 PUTBMP 66
6.16 PUTPCX 69
6.17 QRCODE 71
6.18 RSS 77
6.19 REVERSE 82
6.20 DIAGONAL 83
6.21 TEXT 84
6.22 BLOCK 88

6.23 FONT_REDIRECT 91
7 ステータス取得と即時実行コマンド 92
7.1 <ESC>!? 92
7.2 <ESC>!C 93
7.3 <ESC>!D 94
7.4 <ESC>!O 95
7.5 <ESC>!P 96
7.6 <ESC>!Q 97
7.7 <ESC>!R 98
7.8 <ESC>!S 99
7.9 <ESC>!F 101
7.10 <ESC>!. 102
7.11 ~!@ 103
7.12 ~!A 104
7.14 ~!D 106
7.15 ~!E 107
7.16 ~!F 108
7.17 ~!I 109
7.18 ~!T 110
7.19 <ESC>Y 111
7.20 <ESC>Z 112
8 Windows ドライバー用コマンド 113
8.1 !B 113
8.2 !J 114
8.3 !N 115
9 ファイル管理コマンド 116
9.1 DOWNLOAD 116
9.2 EOP 119
9.3 FILES 120
9.4 KILL 121
9.5 MOVE 122
9.6 RUN 123
10 基本コマンドと制御コマンド 124
10.1 ABS() 124
10.2 ASC() 125
10.3 CHR$() 126
10.4 XOR$() 127
10.5 END 128
10.6 EOF() 129
10.7 OPEN 130
10.8 CLOSE 132
10.9 WRITE 133
10.10 READ 134
10.11 SEEK 135
10.12 LOF() 136
10.13 LOC() 137
10.14 FREAD$() 138
10.15 PUT 139
10.16 GET 140
10.17 COPY 141
10.18 FOR…NEXT LOOP 142
10.19 WHILE…WEND 144
10.20 DO…LOOP 146
10.21 IF…THEN…ELSE…ENDIF LOOP 149
10.22 GOSUB…RETURN 152
10.23 GOTO 153
10.24 INP$() 154
10.25 INP() 155
10.26 LOB() 156

10.27 INPUT 157
10.28 PREINPUT 158
10.29 POSTINPUT 159
10.30 SET FILTER 160
10.31 REM 161
10.32 OUT 162
10.33 OUTR 163
10.34 GETKEY() 164
10.35 INT() 165
10.36 LEFT$() 166
10.37 LEN() 167
10.38 MID$() 168
10.39 RIGHT$() 169
10.40 STR$() 170
10.41 STRCOMP() 171
10.42 INSTR() 172
10.43 TRIM$() 173
10.44 LTRIM$() 174
10.45 RTRIM$() 175
10.46 TEXTPIXEL() 176
10.47 BARCODEPIXEL() 177
10.48 VAL() 178
10.49 NOW$() 179
10.50 NOW 180
10.51 FORMAT$() 181
10.52 DATEADD() 185
10.53 FSEARCH() 187
10.54 TOUCHPRESS() 188
10.55 RECORDSET$() 189
10.56 REPLACE$() 191
11 デバイス動作設定コマンド 192
11.1 SET COUNTER 192
11.2 SET CUTTER 193
11.3 SET PARTIAL_CUTTER 194
11.4 SET BACK 195
11.5 SET BACK_BEFORE_PRINT 196
11.6 SET KEYN 197
11.7 SET LEDN 199
11.8 SET PEEL 200
11.9 SET REWIND 201
11.10 SET TEAR 202
11.11 SET GAP 203
11.12 SET BLINE REVERSE / SET BLINE OBVERSE 205
11.13 SET BLINE PRINTSIDE / SET BLINE BACKSIDE 206
11.14 SET HEAD 207
11.15 SET RIBBON 208
11.16 SET ENCODER 209
11.17 SET RIBBONEND 210
11.18 SET COM1 211
11.19 SET PRINTKEY 212
11.20 SET REPRINT 214
11.21 SET FEED_LEN 215
11.22 GETSENSOR() 216
11.23 GETSETTING$() 219
11.24 SET USBHOST 222
11.25 SET AUTORUN 223
11.26 SET RESPONSE 224
11.27 SET DAYLIGHT_SAVE 226
11.28 SET REGISTRATION 227

11.29 PEEL 228
11.30 LED1, LED2, LED3 229
11.31 KEY1, KEY2, KEY3 230
11.32 SET SENSOR_REF 231
12 プリンターのグローバル変数 232
12.1 @LABEL 232
12.2 YEAR 233
12.3 MONTH 234
12.4 DATE 235
12.5 WEEK 236
12.6 HOUR 237
12.7 MINUTE 238
12.8 SECOND 239
12.9 @YEAR 240
12.10 @MONTH 241
12.11 @DATE 242
12.12 @DAY 243
12.13 @HOUR 244
12.14 @MINUTE 245
12.15 @SECOND 246
12.16 _MODEL$ 247
12.17 _SERIAL$ 248
12.18 _VERSION$ 249
13 Bluetooth 設定コマンド 250
13.1 BT NAME 250
13.2 BT PINCODE 251
13.3 BT MODE 252
13.4 SET BTLINKBACK 254
14 Wi-Fi 設定コマンド 255
14.1 WLAN SSID 255
14.2 WLAN WPA 256
14.3 WLAN WEP 257
14.4 WLAN DHCP 258
14.5 WLAN IP 259
14.6 WLAN PORT 260
14.7 WLAN THROUGH 261
15 内部イーサネット設定コマンド 262
15.1 NET DHCP 262
15.2 NET IP 263
15.3 NET PORT 264
15.4 NET NAME 265
15.5 NET THROUGH 266
16 RJ-2035B/3035B 向け設定コマンド 267
16.1 SET PRINTQUALITY 267
16.2 SET STANDBYTIME 268
16.3 SET SLEEPTIME 269
17 RFID 設定コマンド 270
17.1 RFID READ / RFID WRITE 270
17.2 RFIDDETECT 275
18 General purpose input output (GPIO) 設定コマンド 276
18.1 GPIO をお使いの前に 276
18.2 SET GPO 277
18.3 SET GPI 280
18.4 GPIO ご使用上の注意 283
付録 A: 機種別対応コマンド一覧 285
付録 B: 日本語テキストを印刷する 295
付録 C: 外部機器（PLC、バーコードリーダー等）を使用して印刷する 298

1　コマンドの見方

1.1 Print

コマンド

コマンドの説明

説明

現在イメージバッファに保存されているラベルフォーマットを印刷します。

構文

```
PRINT m[,n]
```

コマンド構文

各パラメーターの詳細説明

パラメーター	説明
m	印刷するラベルのセット数を指定します 1 ≤ m ≤ 999999999
n	各特定のラベルセットに対し、印刷するコピー枚数を指定します 1 ≤ n ≤ 999999999

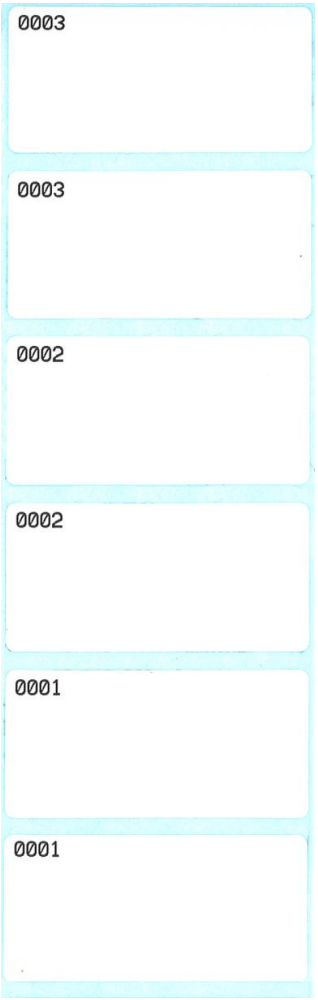
例

サンプルコードと印字結果

サンプルコード

```
SIZE 50 mm,25 mm
GAP 3 mm,0
DIRECTION 1
SET COUNTER @1 1
@1="0001"
CLS
TEXT 10,10, "3",0,1,1,@1
PRINT 3,2
```

結果



1 セット, 2 部

用紙送り方向

関連コマンド

SET COUNTER, INPUT, DOWNLOAD

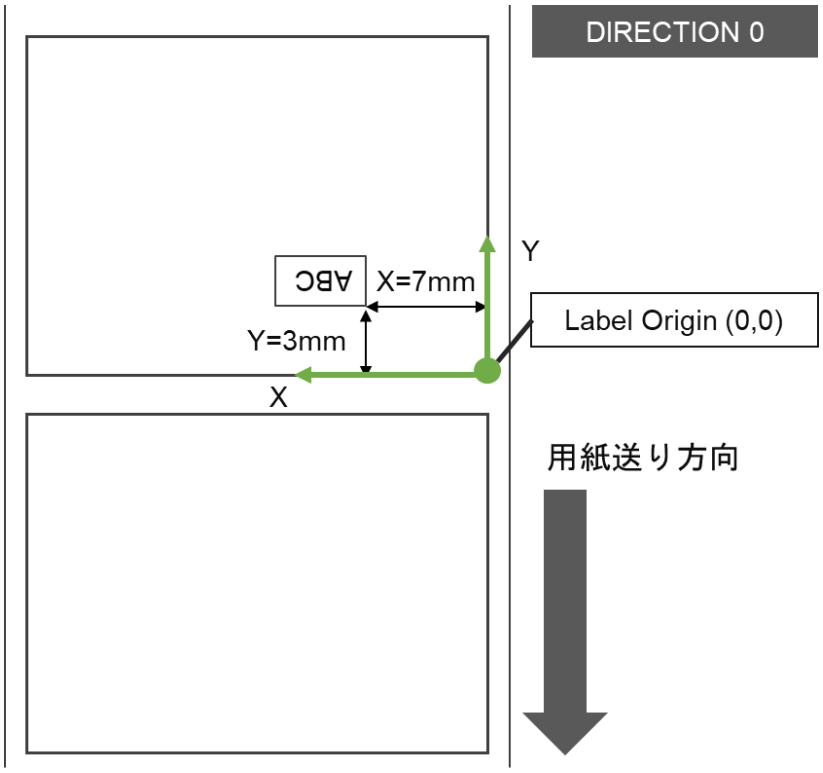
関連するコマンド

2　表記の説明

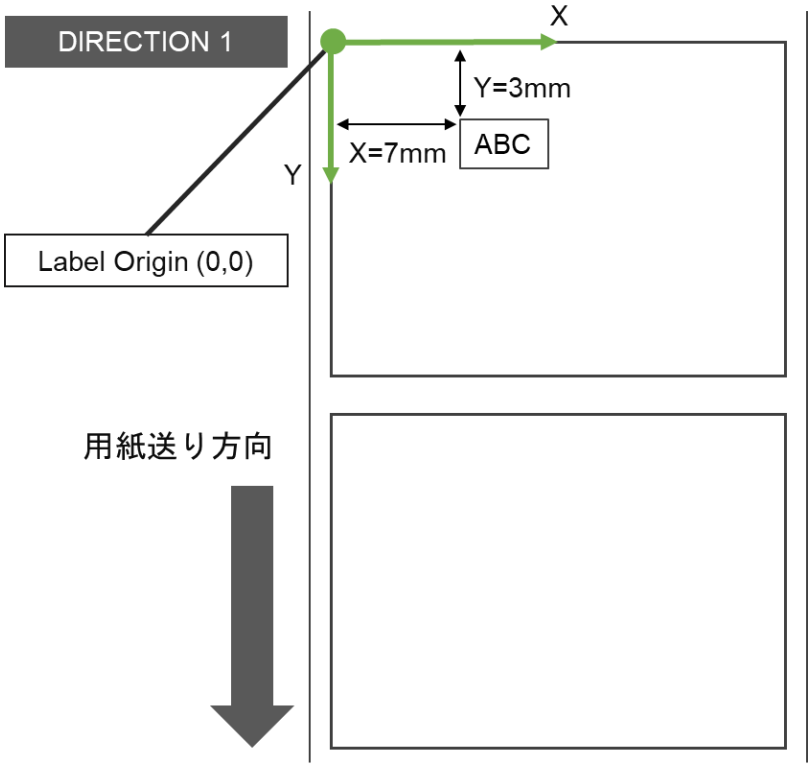
本マニュアルでは以下の記号が使用されます。

記号	説明
[expression list]	大括弧内の設定項目は任意、式は最大 2*1024 バイトの長さ
<ESC>	ASCII 27、ステータスポーリングコマンド、プリンターのステータスを即座に戻します/実行します
~	ASCII 126、ステータスポーリングコマンド、プリンターステータスが準備完了状態の時のみ戻り値を返します
Space	ASCII 32、コマンド行内では無視されます
"	ASCII 34、式の最初と最後
CR, LF	ASCII 13、ASCII 10、コマンド行の末尾であることを示しています
NULL	ASCII 0、式の中で利用可能です
備考: 203 dpi: 1mm = 8 ドット	備考

3　印字位置




```
DIRECTION 0
CLS
TEXT 56,24,"3",0,1,1,"ABC"
PRINT 1
```




```
DIRECTION 1
CLS
TEXT 56,24,"3",0,1,1,"ABC"
PRINT 1
```

備考：

1. 203 dpi, 1 mm = 8 ドット；300 dpi, 1 mm = 12 ドット；600 dpi, 1mm = 24 ドット
2. 整数部のみ有効です。例：2mm = 23.6 ドットの場合、23 ドットが用いられます

4　対応機種

本書に対応しているプリンターのシリーズ、機種名は下記のとおりです。

シリーズ	機種
TD-4T シリーズ	TD-4420TN/4520TN
	TD-4650TNWB/4750TNWB/4750TNWBR
RJ シリーズ	RJ-2035B/3035B
TJ シリーズ	TJ-4021TN/4121TN/4121TNR TJ-4420TN/4520TN/4620TN

詳細は、付録 A: 機種別対応コマンド一覧をご参照ください。

5　セットアップとシステムコマンド

5.1 SIZE

説明

ラベルの幅と長さを定義します。

構文

SIZE m[,n]	インチ
SIZE m mm[,n mm]	ミリ
SIZE m dot[,n dot]	ドット

パラメーター	説明
M	ラベル幅（インチ／ミリ／ドット）
[N]	ラベル長（インチ／ミリ／ドット）※オプション

備考：

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット
- ミリとドットの場合、パラメーターと「ミリ」や「ドット」の間にスペースを入れます。

例

サンプルコード

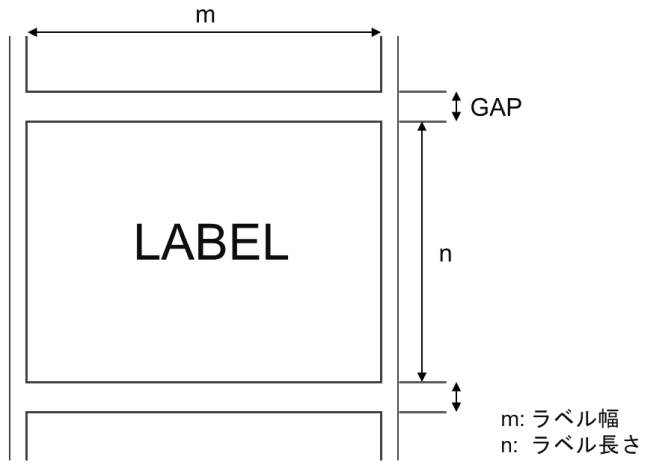
インチ:

```
SIZE 3.5,3.00
```

ミリ:

```
SIZE 100 mm,100 mm
```

結果



関連コマンド

GAP, BLINE

5.2 GAP

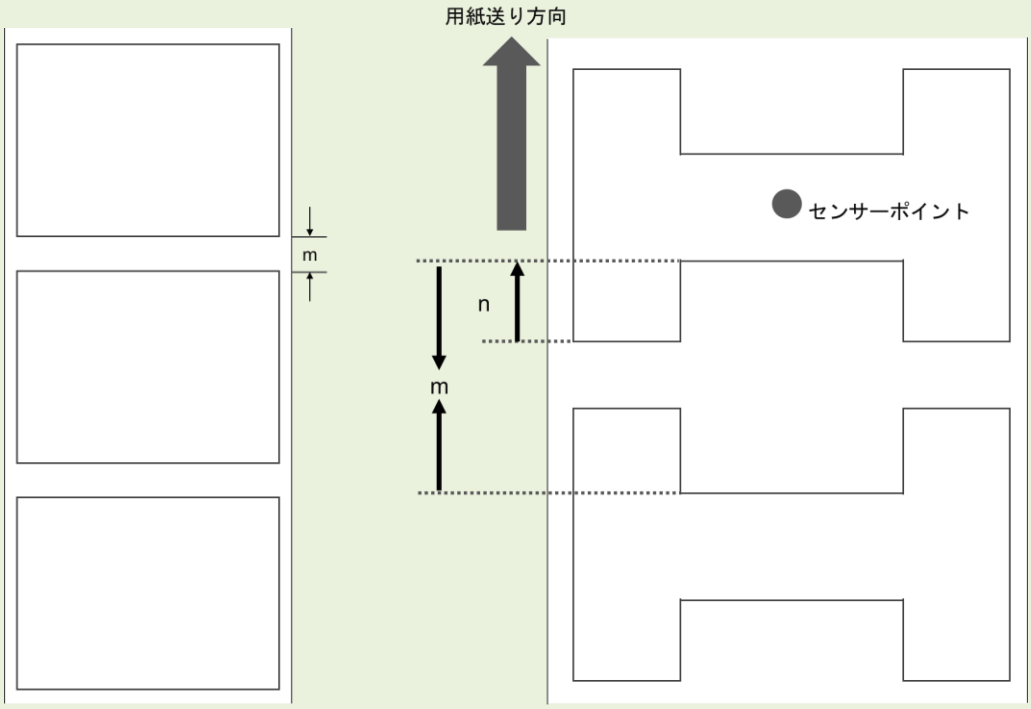
説明

2 枚のラベル間のギャップ（間隔）距離を定義します。

構文

GAP m,n	インチ
GAP m mm,n mm	ミリ
GAP m dot,n dot	ドット

パラメーター	説明
M	ラベル間のギャップ（間隔）距離 0 ≤ m ≤1 (inch), 0 ≤ m ≤ 25.4 (mm) 0 ≤ m ≤5 (inch), 0 ≤ m ≤ 127 (mm)
N	ギャップのオフセット（離間）距離 n ≤ ラベル長 (インチまたはミリ)
0, 0	無定長ラベル

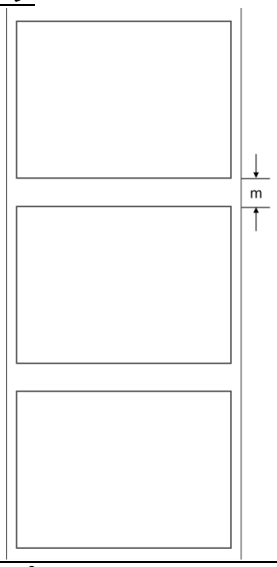


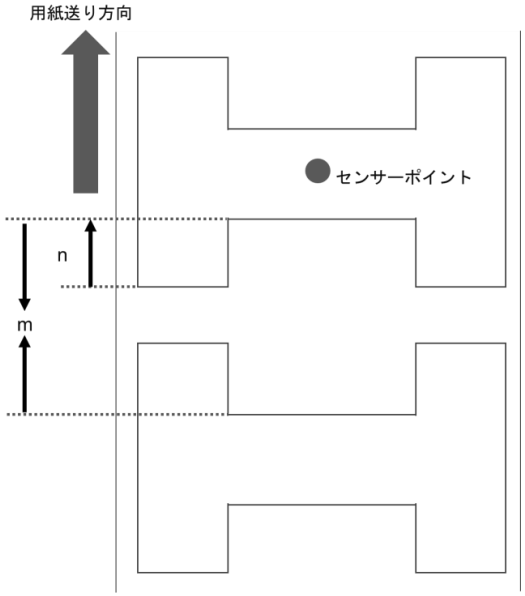
備考

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット
- ミリとドットの場合、パラメーターと「mm」や「dot」の間にスペースを入れます。
- センサータイプが「ブラックマーク」から「ギャップ」へ変更された場合、最初にプリンターへ **GAP** コマンドを送信します。

例

サンプルコード	結果

通常のギャップ
インチ:
`GAP 0.12,0`
 	
ミリ:
`GAP 3 mm,0 mm`
 	
無定長ラベル:
`GAP 0,0`
	通常のギャップ
	 

特殊なギャップ
インチ
`GAP 0.30,0.10`
 	
ミリ
`GAP 7.62 mm,2.54 mm`
	特殊なギャップ
	 

関連コマンド

SIZE, BLINE

5.3 GAPDETECT

説明

ラベル長とギャップ長を定めるために用紙を送ります。このコマンドはユーザー指定のパラメーターを参照するため、パラメーターと実際の長さが異なる場合、**GAPDETECT** コマンドは正しく動作しません。パラメーターX で指定した用紙長間は用紙端を検出しないため、ロゴやテキストがプレ印刷されているラベルにも対応可能です。

構文

```
GAPDETECT [x,y]
```

パラメーター	説明
X	用紙長（ドット）
Y	ギャップ長 （ドット）

備考：
ｘとｙのパラメーターが無効な場合、プリンターはキャリブレーションを実行し、自動的に用紙の長さとギャップのサイズを決定します。

関連コマンド

GAP, SIZE, BLINEDETECT, AUTODETECT

5.4 BLINEDETECT

説明

用紙とブラックマークの長さを定めるために、用紙をブラックマークセンサーまで送ります。このコマンドはユーザー指定のパラメーターを参照するため、パラメーターと実際の長さと異なる場合、**BLINEDETECT** コマンドは正しく動作しません。パラメーターx で指定した用紙長間は用紙端を検出しないため、ロゴやテキストがプレ印刷されているラベルにも対応可能です。

構文

```
BLINEDETECT [x,y]
```

パラメーター	説明
x	用紙長（ドット）
y	ギャップ長（ドット）

備考：
x と y のパラメーターが無効な場合、プリンターはキャリブレーションを実行し、自動的に用紙の長さとギャップのサイズを決定します。

関連コマンド

GAP, SIZE, GAPDETECT, AUTODETECT

5.5 AUTODETECT

説明

用紙、ギャップ、ブラックマークの長さを定めるために、用紙をギャップ/ブラックマークセンサーまで送ります。このコマンドはユーザー指定のパラメーターを参照するため、パラメーターと実際の長さと異なる場合、**AUTODETECT** コマンドは正しく動作しません。パラメーターX で指定した用紙長間は用紙端を検出しないため、ロゴやテキストがプレ印刷されているラベルにも対応可能です。

構文

```
AUTODETECT [x,y]
```

パラメーター	説明
x	用紙長（ドット）
y	ギャップ長 （ドット）

備考：

- x と y のパラメーターが無効な場合、プリンターは自動的にキャリブレーションを実行し、用紙の長さとギャップ/ブラックマークのサイズを決定します。
- このコマンドを使用する際、プリンターは正しいセンサータイプによりラベルを検出するため、**GAP** や **BLINE** コマンドをご自身のプログラムへ設定しないようにしてください。

関連コマンド

GAP, SIZE, GAPDETECT, BLINEDETECT

5.6 BLINE

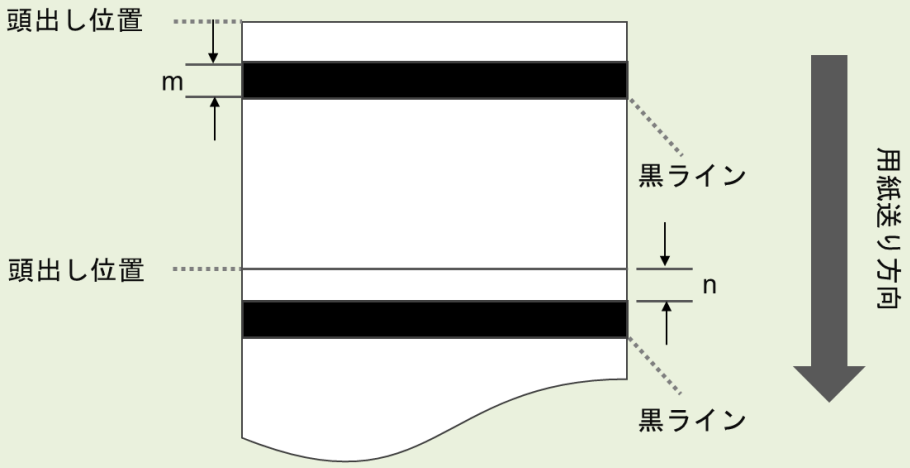
説明

用紙の頭出しをする際の黒ラインの高さとユーザー定義の追加のラベル送りの長さを設定します。

構文

BLINE m,n	インチ
BLINE m mm,n mm	ミリ
BLINE m dot,n dot	ドット

パラメーター	説明
m	黒ラインの高さ（インチまたはミリ） 0 ≤ m ≤ 1 (inch), 0 ≤ m ≤ 25.4 (mm) 0 ≤ m ≤5 (inch), 0 ≤ m ≤ 127 (mm)
n	追加のラベル送り長さ 0 ≤ n ≤ ラベル長
0,0	無定長ラベル



備考：

- ミリの場合、パラメーターとミリの間にスペースを入れます。
- センサータイプが「ギャップ」から「ブラックマーク」へ変更された場合、最初にプリンターへ **BLINE** コマンドを送信します。
- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi：1 mm = 24 ドット

例

サンプルコード

インチ:

```
BLINE 0.20,0.50
```

ミリ:

```
BLINE 5.08 mm,12.7 mm
```

関連コマンド

SIZE, GAP

5.7 OFFSET

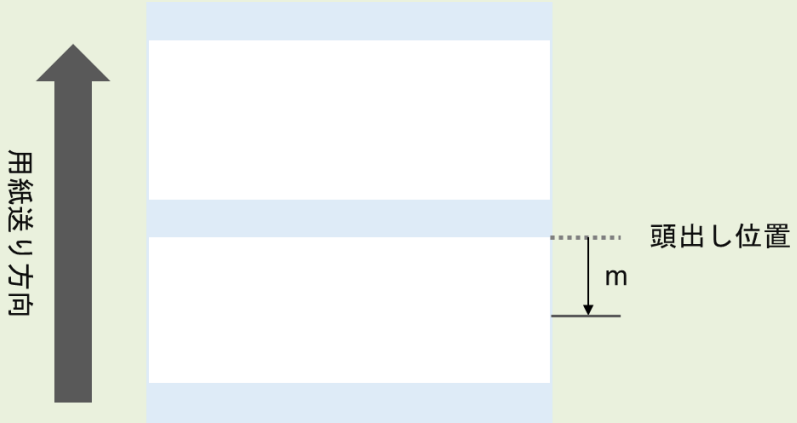
説明

用紙の停止位置を微調整します。特にカッターモードやハクリモード時のラベルの停止位置を調整するために使用します。プリンターは次の印刷を実行する前にオフセット分のラベル長を逆送りします。

構文

`OFFSET m`	インチ
`OFFSET m mm`	ミリ
`OFFSET m dot`	ドット

パラメーター	説明
m	オフセット距離（インチまたはミリ） -1 ≤ m ≤ 1 (inch)



備考：

- 不適切なオフセット値は紙詰まりを起こす可能性があります。
- ミリの場合、パラメーターとミリの間にスペースを入れます。
- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット

例

サンプルコード

インチ:

```
OFFSET 0.5
```

ミリ:

```
OFFSET 12.7 mm
```

関連コマンド

SIZE, GAP, SET PEEL, SET CUTTER

5.8 SPEED

説明

印刷速度を設定します。

構文

```
SPEED n
```

パラメーター	説明
n	インチ毎秒単位の印刷速度

Model / IPS	1	1.5	2	2.5	3	3.5	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
TD-4420TN		V	V		V		V	V	V												
TD-4520TN		V	V		V		V	V													
TD-4650TNWB		V	V		V		V	V	V	V	V										
TD-4750TNWB, TD-4750TNWBR		V	V		V		V	V	V												
RJ-2035B	V	V	V		V		V														
RJ-3035B	V	V	V		V		V														
TJ-4021TN	V	V	V		V		V	V	V	V	V	V	V								
TJ-4121TN, TJ-4121TNR	V	V	V		V		V	V	V	V											
TJ-4420TN			V		V		V	V	V	V	V	V	V	V	V	V	V				
TJ-4520TN			V		V		V	V	V	V	V	V	V	V	V						
TJ-4620TN		V	V		V		V	V	V												

例

サンプルコード

```
SPEED 10
```

関連コマンド

DENSITY

5.9 DENSITY

説明

印字濃度を設定します。

構文

```
DENSITY n
```

パラメーター	説明
n	0~15 0:　最小濃度 15:　最大濃度

備考：
初期値の濃度設定値は 8 です。

例

サンプルコード

```
DENSITY 7
```

5.10 DIRECTION

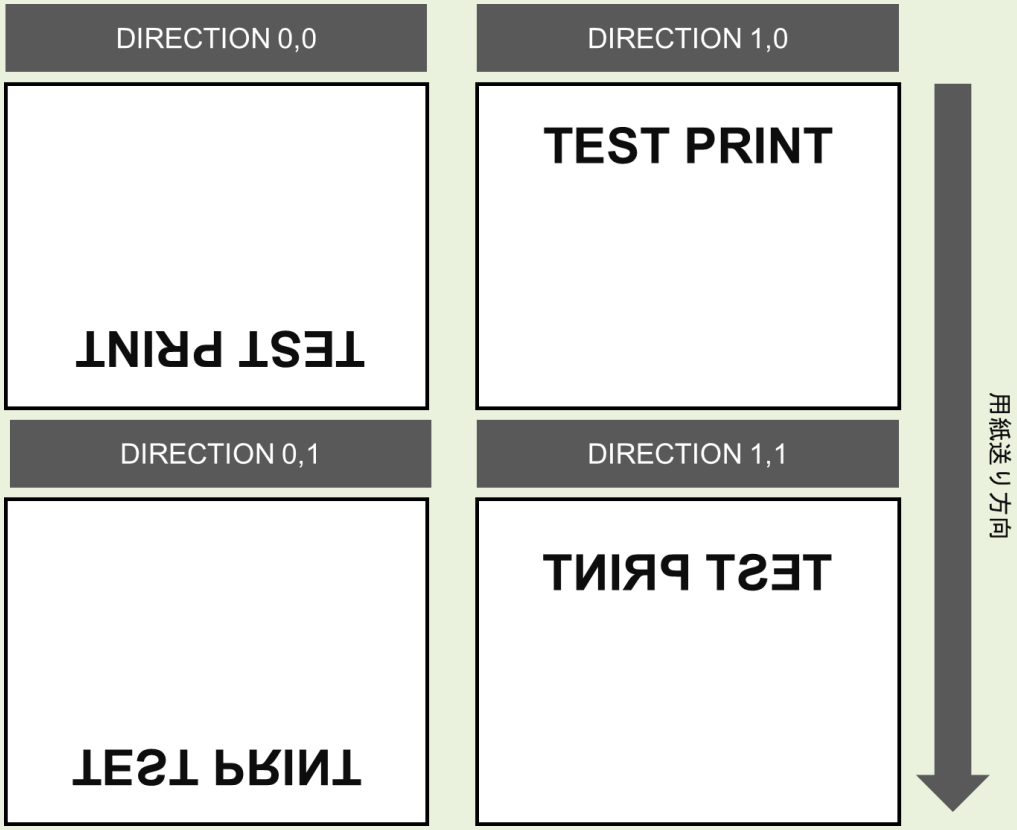
説明

印刷方向とミラーイメージを定義します。プリンターのメモリ内に記憶されます。

構文

```
DIRECTION n[,m]
```

パラメーター	説明
n	0 または 1（下図参照）
m	0: 通常のイメージを印刷する 1: ミラーイメージを印刷する



例

サンプルコード

- `DIRECTION 0`
- `DIRECTION 0,1`

関連コマンド

REFERENCE

5.11REFERENCE

説明

ラベルの基準点を定義します。基準点は印刷方向によって変わります。（下図参照）

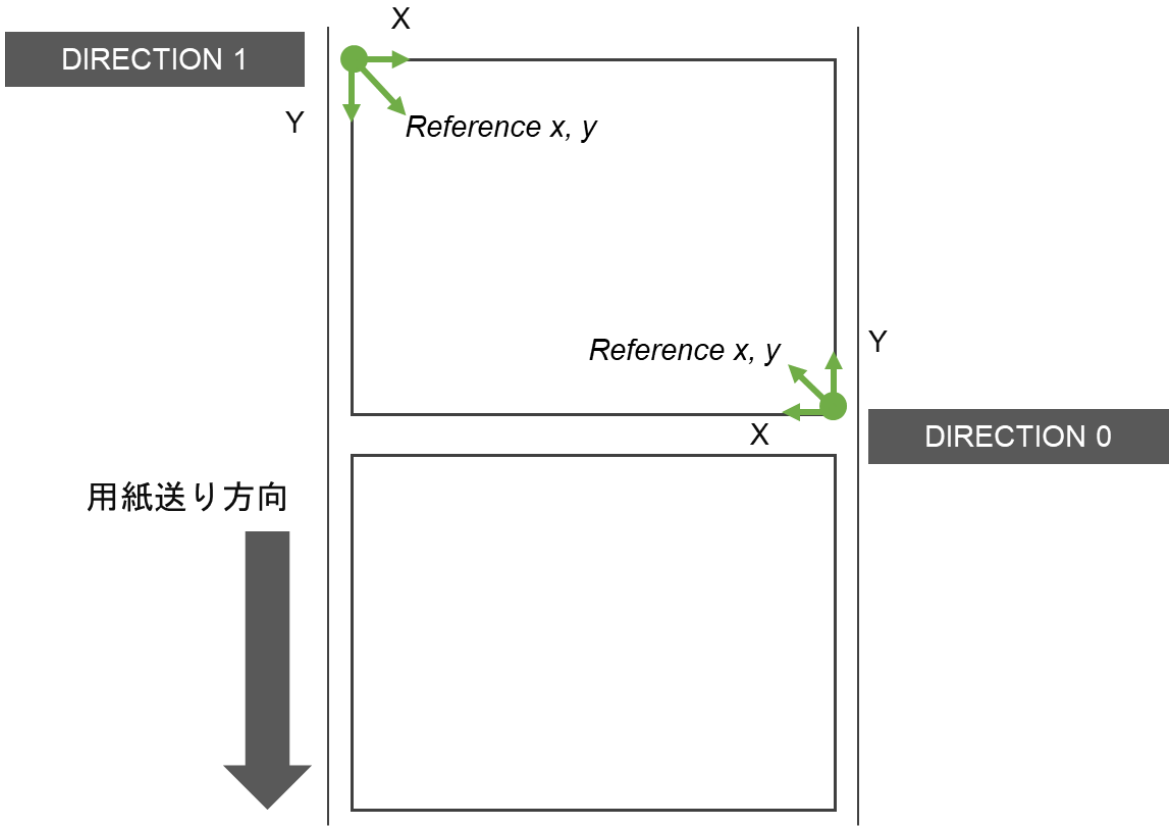
構文

```
REFERENCE x, y
```

パラメーター	説明
x	水平座標（ドット）
y	垂直座標（ドット）

備考：

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット



例

サンプルコード

```
REFERENCE 10,10
```

関連コマンド

DIRECTION

5.12 SHIFT

説明

搬送方向に対してラベルを移動させます。正の値はラベルを搬送方向と反対に移動させ、負の値はラベルを搬送方向に移動させます。

構文

```
SHIFT [x,] y
```

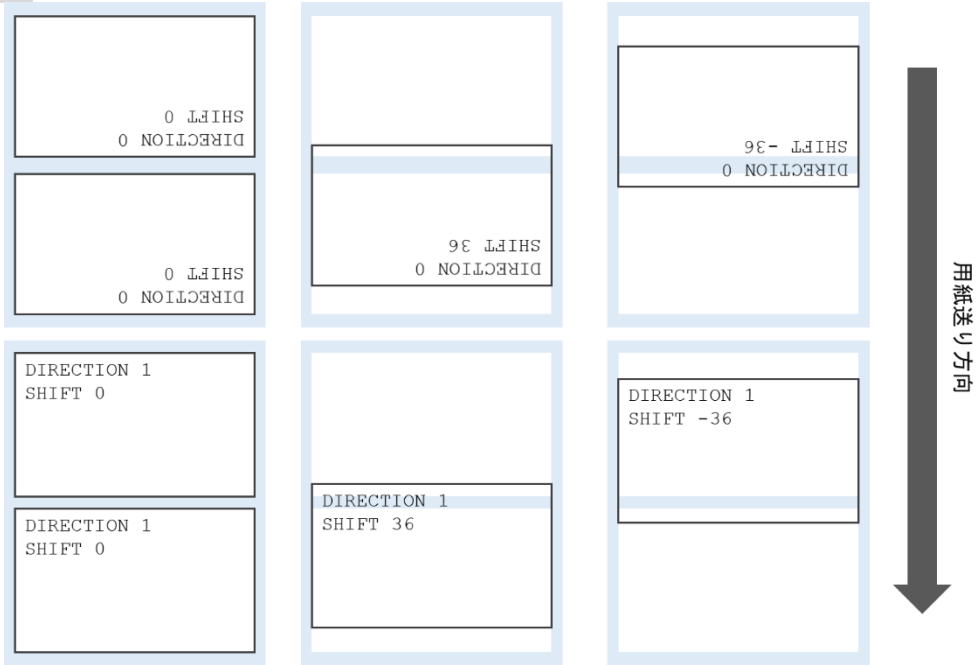
パラメーター	説明
x	オプション。最大値は 1 インチ。 • 200 dpi: –203 to 203 （ドット） • 300 dpi: -300 to 300 （ドット）
Y	最大値は 1 インチ。 • 200 dpi: –203 to 203 （ドット） • 300 dpi: -300 to 300 （ドット）

例

サンプルコード

```
SIZE 4,2.5
GAP 2 mm,0
DIRECTION 0
SHIFT 36
OFFSET 0
CLS
TEXT 400,200, "3",0,1,1, "DIRECTION 0"
TEXT 400,250, "3",0,1,1, "SHIFT 36"
BOX 10,0,780,490,8
PRINT 3,1
```

結果



関連コマンド

OFFSET, REFERENCE

5.13 COUNTRY

説明

外付け機器（キーボード等）上の特殊文字を定義することにより、様々な国の仕様でキーボードを使用できるようになります。

構文

```
COUNTRY n
```

パラメーター	説明	
n	001	USA
	003	Spanish (Latin America)
	007	Russia
	031	Dutch
	033	French (France)
	034	Spanish
	061	English (international)

例

サンプルコード

```
COUNTRY 001
```

関連コマンド

CODEPAGE, ~!l

5.14 CODEPAGE

説明

国際文字セット（コードページ）を定義します。

構文

```
CODEPAGE n
```

パラメーター

n

説明

コードページの名前や数は 7 ビットコードページと 8 ビットコードページに分割することが可能です。

7-bit		8-bit		Windows		ISO	
n	Name	n	Name	n	Name	n	Name
USA	USA	**437**	United States	**1250**	Central Europe	**8859-1**	Latin 1
BRI	British	**737**	Greek	**1251**	Cyrillic	**8859-2**	Latin 2
GER	German	**850**	Multilingual	**1252**	Latin I	**8859-3**	Latin 3
FRE	French	**851**	Greek 1	**1253**	Greek	**8859-4**	Baltic
DAN	Danish	**852**	Slavic	**1254**	Turkish	**8859-5**	Cyrillic
ITA	Italian	**855**	Cyrillic	**1255**	Hebrew	**8859-6**	Arabic
SPA	Spanish	**857**	Turkish	**1256**	Arabic	**8859-7**	Greek
SWE	Swedish	**860**	Portuguese	**1257**	Baltic	**8859-8**	Hebrew
SWI	Swiss	**861**	Icelandic	**1258**	Vietnam	**8859-9**	Turkish
		862	Hebrew	**932**	Japanese Shift-JIS	**8859-10**	Latin 6
		863	Canadian/ French	**936**	Simplified Chinese GBK	**8859-15**	Latin 9
		864	Arabic	**949**	Korean		
		865	Nordic	**950**	Traditional Chinese Big5		
		866	Russian	**UTF-8**	UTF 8		
		869	Greek 2				

備考：

- データ長は 7 ビットか 8 ビットの通信パラメーターにより決定されます。
- 一部の文字が印刷されなかったり、文字化けしたりする場合は、以下をお試し下さい。

  プリンターの **CODEPAGE** 設定が印刷データの文字コードと一致しているか確認してください。必要な場合は、どちらかの文字コードを変更してください。

  プリンターの **CODEPAGE** 設定が、フォントの対応文字コードでない可能性があります。プリンターの **CODEPAGE** 設定を変更するか、文章で使用するフォントを変更してみてください。
- フォント「K1.SFT /K2.SFT /K3.SFT /K4.SFT /K8.SFT /K9.SFT /KA.SFT /KB.SFT」は、シフト JIS に対応しています。使用する場合は、**CODEPAGE** コマンドで「932（Shift-JIS)」を指定ください。

サンプルコード

```
DOWNLOAD "TEST.BAS"

str1$ = " "
J = 0
y = 50

CODEPAGE 1252
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT
10,10,"COUR.TTF",0,12,12,"CODEPAGE
1252"
FOR I=32 TO 255
str1$=str1$+CHR$(I) + " "
J=J+1
IF J=16 THEN GOSUB drawTEXT
NEXT

PRINT 1
END

drawTEXT:
TEXT 10,y,"COUR.TTF",0,12,12,str1$
str1$=" "
J=0
y=y+40
RETURN
EOP
TEST
```

結果

```
CODEPAGE 1252
  ! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ 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 { | } ~
€   ‚ ƒ „ … † ‡ ˆ ‰ Š ‹ Œ   Ž
  ‘ ’ “ ” • – — ˜ ™ š › œ   ž Ÿ
  ¡ ¢ £ ¤ ¥ ¦ § ¨ © ª « ¬ - ® ¯
° ± ² ³ ´ µ ¶ · ¸ ¹ º » ¼ ½ ¾ ¿
À Á Â Ã Ä Å Æ Ç È É Ê Ë Ì Í Î Ï
Ð Ñ Ò Ó Ô Õ Ö × Ø Ù Ú Û Ü Ý Þ ß
à á â ã ä å æ ç è é ê ë ì í î ï
ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ
```

関連コマンド

COUNTRY, ~!I

5.15CLS

説明

イメージバッファを消去します。

構文

```
CLS
```

パラメーター	説明
None	N/A

備考：
このコマンドは **SIZE** コマンドの後に使用します。

例

サンプルコード

```
CLS
```

関連コマンド

SIZE, GAP, BLINE

5.16 FEED

説明

指定した長さのラベルを送ります。長さはドットで指定します。

構文

```
FEED n
```

パラメーター	説明
n	単位：ドット 1 ≤ n ≤ 9999

備考：

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi：1 mm = 24 ドット

例

サンプルコード

```
FEED 40
```

関連コマンド

BACKFEED, SIZE, GAP, BLINE, HOME, FORMFEED

5.17 BACKFEED

説明

指定した長さのラベルを逆送りします。長さはドットで指定します。

構文

```
BACKFEED n
```

パラメーター	説明
n	単位：ドット 1 ≤ n ≤ 9999

備考：

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi：1 mm = 24 ドット
- 不適切な値を指定した場合、紙詰まりやしわの原因となる可能性があります。

例

サンプルコード

```
BACKFEED 40
```

関連コマンド

FEED, SIZE, GAP, BLINE, HOME, FORMFEED

5.18 FORMFEED

説明

次のラベルの先端までラベルを送ります。

構文

```
FORMFEED
```

パラメーター	説明
None	N/A

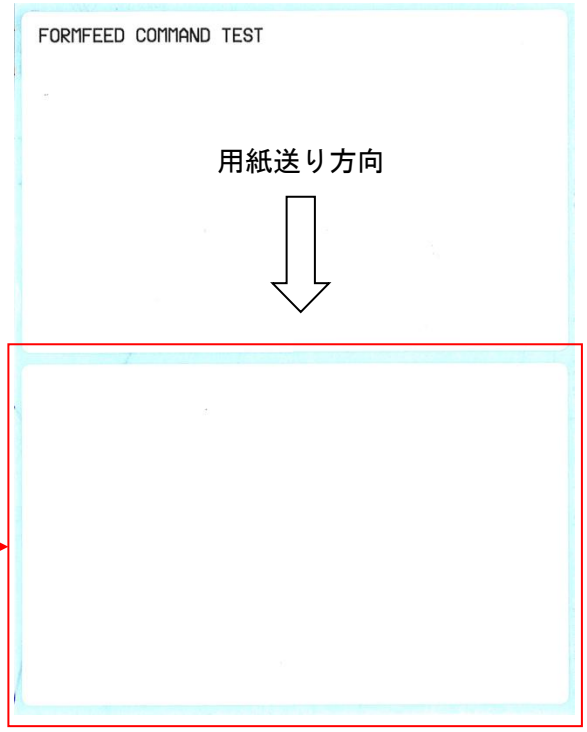
備考：
このコマンドは **SIZE** コマンドの後に使用します。

例

サンプルコード

```
SIZE 4,2.5
GAP 2 mm,0
DIRECTION 1
FORMFEED
CLS
TEXT  25,25,  "3",0,1,1,  "FORMFEED
COMMAND TEST"
PRINT 1,1
```

結果



関連コマンド

FEED, SIZE, GAP, BLINE, HOME, BACKFEED

5.19 HOME

説明

内部センサーが原点を決め終わるまでラベルを送ります。

構文

```
HOME
```

パラメーター	説明
None	N/A

備考：
このコマンドを使用する前に、ラベルのサイズとギャップを定義する必要があります。

例

サンプルコード

```
SIZE 4,2.5
GAP 2 mm,0
SET COUNTER @0 +1
@0="000001"
HOME
CLS
BOX 1,1,360,65,12
TEXT 25,25, "3",0,1,1, "HOME COMMAND TEST"
TEXT 25,80, "3",0,1,1,@0
PRINT 3,1
```

関連コマンド

FEED, SIZE, GAP, BLINE, FORMFEED

5.20 PRINT

説明

現在イメージバッファに保存されているラベルフォーマットを印刷します。

構文

```
PRINT m[,n]
```

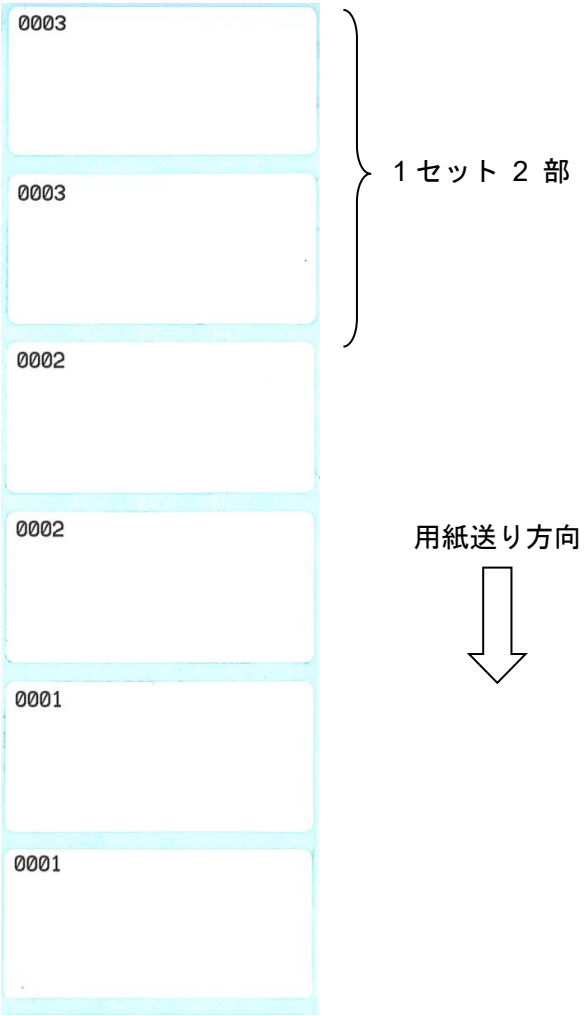
パラメーター	説明
m	印刷するラベルのセット数を指定します 1 ≤ m ≤ 999999999
n	各特定のラベルセットに対し、印刷するコピー枚数を指定します 1 ≤ n ≤ 999999999

例

サンプルコード

```
SIZE 50 mm,25 mm
GAP 3 mm,0
DIRECTION 1
SET COUNTER @1 1
@1="0001"
CLS
TEXT 10,10, "3",0,1,1,@1
PRINT 3,2
```

結果



関連コマンド

SET COUNTER, INPUT, DOWNLOAD

5.21 SOUND

説明

ビープ音を鳴らします。音量とタイミングを調整できます。

構文

```
SOUND level,interval
```

パラメーター	説明
level	音量レベル 0~9
interval	インターバル 1~4095 （ミリ秒）

例

サンプルコード

```
・ SOUND 5,200
・ SOUND 3,200
・ SOUND 3,200
・ SOUND 4,200
・ SOUND 2,200
・ SOUND 2,200
・ SOUND 1,200
・ SOUND 2,200
・ SOUND 3,200
・ SOUND 4,200
・ SOUND 5,200
```

5.22 CUT

説明

ラベルを逆送りすることなく、カットします。（カッターオプション装着時のみ）

構文

```
CUT
```

パラメーター	説明
None	N/A

例

サンプルコード

```
SIZE 3,3
GAP 0,0
CLS
BOX 0,0,866,866,5
TEXT 100,100, "5",0,1,1, "FEED & CUT"
TEXT 100,200, "5",0,1,1, "300 dpi"
PRINT 1,1
FEED 260
CUT
```

関連コマンド

SET CUTTER, SET BACK, SET PARTITAL_CUTTER

5.23 LIMITFEED

説明

適切なセンサー感度が設定されていない場合、プリンターはラベル搬送中にギャップの正しい位置を特定できません。このコマンドは、プリンターがラベル搬送中にギャップの位置を特定できない場合、ラベル送りを停止させ、赤い LED を点滅させます。

構文

LIMITFEED n[,minpaper,maxgap]	インチ
LIMITFEED n mm[,minpaper mm,maxgap mm]	ミリ
LIMITFEED n dot[,minpaper dot,maxgap dot]	ドット

パラメーター	説明
n	センサー検知の最大の長さ
Minpaper	用紙の最大の長さ
Maxgap	ギャップの最大の長さ

備考：

- この設定はメモリ内に記憶されます。
- ミリの場合、パラメーターn とミリの間にスペースを入れます。
- プリンター—を初期化すると、初期値 10 インチに戻ります。
- プレ印刷された用紙や穴のある用紙をキャリブレーションできない場合は、Minpaper または Maxgap を入力します。

例

サンプルコード

```
LIMITFEED 12
LIMITFEED 10, 2.36, 0.12
LIMITFEED 250 mm, 60 mm, 3 mm
LIMITFEED 2000 dot, 480 dot, 24 dot
```

5.24 SELFTEST

説明

プリンター情報を印刷します。

構文

```
SELFTEST [page]
```

パラメーター	説明
page	None: 全てのプリンター情報を含むセルフテストページを印刷します PATTERN: 印字ヘッドの状態を確認するためパターンを印刷します ETHERNET: イーサネット設定を印刷します WLAN: Wi-Fi 設定を印刷します RS232: RS-232C 設定を印刷します SYSTEM: プリンター設定を印刷します Z: エミュレーション言語設定を印刷します BT: Bluetooth 設定を印刷します

例

次のページに続きます。

サンプルコード	結果
SELFTEST	---------- SYSTEM INFORMATION ---------- MODEL: XXXXXX FIRMWARE: XXXXXX CHECKSUM: XXXXXX S/N: XXXXXX TCF: NO DATE: 1970/01/01 TIME: 00:04:18 NON-RESET: 110 m (TPH) RESET: 110 m (TPH) NON-RESET: 0 (CUT) RESET: 0 (CUT) ---------- ---------- PRINTING SETTING ---------- SPEED: 5 IPS DENSITY: 8.0 WIDTH: 4.00 INCH HEIGHT: 4.00 INCH GAP: 0.00 INCH INTENSION: 5 CODEPAGE: 850 COUNTRY: 001 ---------- ---------- Z SETTING ---------- DARKNESS: 16.0 SPEED: 4 IPS WIDTH: 4.00 INCH TILDE: 7EH (~) CARET: 5EH (^) DELIMITER: 2CH (,) POWER UP: NO MOTION HEAD CLOSE: NO MOTION ---------- ---------- RS232 SETTING ---------- BAUD: 9600 PARITY: NONE DATA BIT: 8 STOP BIT: 1 ---------- ---------- DRAM FILE (0 FILES) ---------- PHYSICAL 8192 KBYTES AVAILABLE 256 KBYTES ---------- ---------- FLASH FILE (0 FILES) ---------- PHYSICAL 4096 KBYTES AVAILABLE 2560 KBYTES ----------
SELFTEST PATTERN	
SELFTEST ETHERNET	---------- ETHERNET SETTING ---------- NAME: XXXXXX MAC ADDR: XXXXXX DHCP: ON IP ADDR: XXXXXX SUBNET: XXXXXX GATEWAY: XXXXXX PORT: 9100 ----------
SELFTEST WLAN	---------- WLAN SETTING ---------- MAC ADDR: XXXX-XXXXX SSID: TEST-AP DHCP: OFF IP ADDR: 10.0.10.138 SUBNET: 255.255.255.0 GATEWAY: 10.0.10.252 PORT: 9100 ----------

SELFTEST RS232	---------- RS232 SETTING ---------- BAUD: 9600 PARITY: NONE DATA BIT: 8 STOP BIT: 1 ----------
SELFTEST SYSTEM	---------- SYSTEM INFORMATION ---------- MODEL: XXXXXX FIRMWARE: XXXXXX CHECKSUM: XXXXXX S/N: XXXXXX TCF: NO DATE: 2013/01/11 TIME: 14:57:55 NON-RESET: 145 m (TPH) RESET: 145 m (TPH) NON-RESET: 0 (CUT) RESET: 0 (CUT) ----------
SELFTEST PRINTER	---------- PRINTING SETTING ---------- SPEED: 5 IPS DENSITY: 8.0 WIDTH: 4.00 INCH HEIGHT: 1.00 INCH GAP: 0.00 INCH INTENSION: 5 CODEPAGE: 850 COUNTRY: 001 ----------
SELFTEST Z	---------- Z SETTING ---------- DARKNESS: 16.0 SPEED: 4 IPS WIDTH: 4.00 INCH TILDE: 7EH (~) CARET: 5EH (^) DELIMITER: 2CH (,) POWER UP: NO MOTION HEAD CLOSE: NO MOTION ----------
SELFTEST BT	---------- BT SETTING ---------- MAC ADDR: XXXXXX111111 NAME: BROTHER01 PIN CODE: 0000 PRINTER NAME: PAIR MODE: LEGACY MODULE: XXXX XXX MFi SUPPORTED: YES ----------

5.25 EOJ

説明

（EOJ 前の）コマンド処理が完了するまでプリンターを待機させ、次のコマンドに進みます。

構文

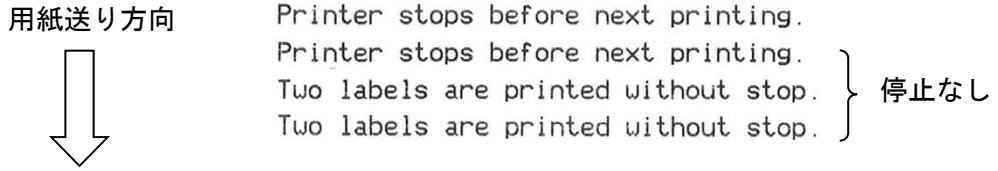
```
EOJ
```

例

サンプルコード

```
SIZE 4,0.2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"3",0,1,1,"Two labels are printed without stop."
PRINT 1
PRINT 1

SIZE 4,0.2
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1,"Printer stops before next printing."
PRINT 1
EOJ
PRINT 1
```

結果



5.26 DELAY

説明

特定の時間プリンターを待機させ、次のコマンドに進みます。

構文

```
DELAY ms
```

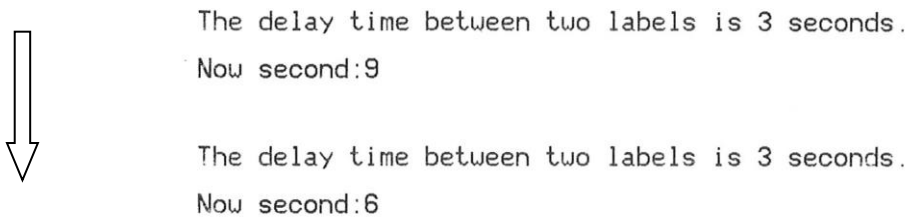
パラメーター	説明
ms	ミリ秒（1000 ms = 1 second）

例

サンプルコード

```
SIZE 4,0.7
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"3",0,1,1,"The delay time between two labels is 3 seconds."
TEXT 10,60,"3",0,1,1,"Now second:" +@SECOND
PRINT 1
DELAY 3000
PRINT 1
```

結果



5.27 DISPLAY

説明

液晶パネル上にプリンターのイメージバッファの画像を表示します。

構文

```
DISPLAY IMAGE/OFF/CLS/……
```

パラメーター	説明
IMAGE	液晶パネル上にプリンターのイメージバッファの画像を表示します
OFF	このコマンドを無効にします
CLS	背景色を表示し、液晶パネル上のプリンターのイメージバッファ項目を消去します
forecolor,backcolor	液晶パネル上のプリンターのイメージバッファに項目と背景のための色（10 進数）を設定します
x,y,width,height	液晶パネル上のプリンターのイメージバッファにバーを描きます
x,y,width,height, thick	
x,y,width,height, thick, radius	
x,y,”bmpfile”	液晶パネル上のプリンターのイメージバッファに.bmp を表示します
x,y,”font”,”content”	液晶パネル上のプリンターのイメージバッファにテキストを表示します
x,y,”font”,rotate,”content”	
x,y,”font”,rotate,multi,”content”	
x,y,”font”,rotate,x-multi,y-multi,”content”	
x,y,”font”,rotate,x-multi,y-multi,align,”content”	

forecolor	テキストやバー（10 進数）のための RGB カラーコード
backcolor	背景（10 進数）のための RGB カラーコード
x	水平方向の乗算
y	垂直方向の乗算
width	フレームの幅
height	フレームの高さ
thick	フレームの厚み
radius	フレームの丸角
bmpfile	.bmp ファイル名
font	フォント名
rotate	回転(0, 90, 180, 270)
x-multi	水平方向の乗算
y-multi	垂直方向の乗算

align	テキスト位置 (1:左揃え, 2:中央揃え, 3:右揃え)
content	テキスト文字列の内容

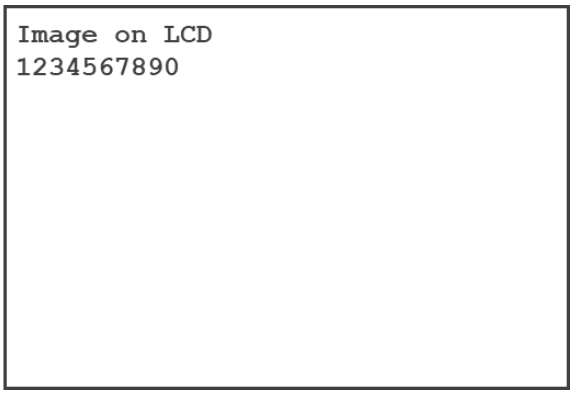
備考：
液晶パネル搭載モデルのみ。
詳細は付録 A: 機種別対応コマンド一覧をご参照ください。

例

サンプルコード

```
CLS
TEXT 1,10, "1",0,1,1, "Image on
LCD"
TEXT 1,30, "1",0,1,1,
"1234567890"
DISPLAY IMAGE
DELAY 5000
DISPLAY OFF
```

結果




```
CLS
DISPLAY 15128749,16711680
DISPLAY CLS
DISPLAY 10,30, "1","1234567890"
DELAY 5000
DISPLAY OFF
```

備考：
16進数「FF0000」を10進数「16711680」に変換して、背景色を定義します。

5.28 INITIALPRINTER

説明

プリンター設定を初期値に戻します。

構文

```
INITIALPRINTER
```

パラメーター	説明
None	N/A

例

サンプルコード

```
INITIALPRINTER
```

5.29 MENU

説明

プリンターに内部のデータベースを用いてユーザー独自のメニューをデザインできます。

構文

```
MENU title$, list$, selected
```

パラメーター	説明
title$	液晶パネル状に表示する文字列
list$	項目一覧（CRLF により識別される）
selected	変数 0 が選択された場合、オペレーターが ESC（USB キーボード）かメニューボタンを押したことになります。

備考：
液晶パネル搭載モデルのみ。
詳細は付録 A: 機種別対応コマンド一覧をご参照ください。

例

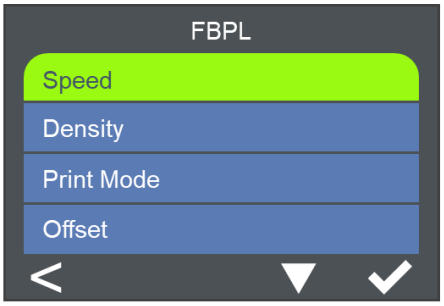
サンプルコード

```
DOWNLOAD F,"FBPL"
Speed
Density
Print Mode
Offset
Country
EOP

DOWNLOAD F,"Speed"
4
5
6
EOP

DOWNLOAD F,"Density"
6
7
8
9
10
11
12
EOP

DOWNLOAD F,"Print Mode"
NONE
TEAR OFF
PEEL OFF
CUT OFF
EOP
```

```
DOWNLOAD F,"Country"
007
031
033
034
045
EOP

DOWNLOAD F,"DEMO.BAS"

DPI = VAL(GETSETTING$("SYSTEM","INFORMATION","DPI"))

:MAINLOOP
OPEN "FBPL",0
LIST$ = FREAD$(0, LOF("FBPL"))
CLOSE 0
MENU "FBPL", LIST$, OPTION$

IF LEN(OPTION$) = 0 THEN END

IF OPTION$ = "Speed" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","SPEED")
IF OPTION$ = "Density" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","DENSITY")
IF OPTION$ = "Print Mode" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","PRINT MODE")
IF OPTION$ = "Offset" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","OFFSET")
IF OPTION$ = "Country" THEN SETTING$ =
GETSETTING$("CONFIG","FBPL","COUNTRY CODE")

IF LOF(OPTION$) <> 0 THEN
     OPEN OPTION$,0
     LIST$ = FREAD$(0, LOF(OPTION$))
     CLOSE 0
     MENU OPTION$, LIST$, SETTING$
ELSE
     IF OPTION$ = "Offset" THEN INPUT "Offset", SETTING$
ENDIF

IF LEN(SETTING$) <> 0 THEN
     IF OPTION$ = "Speed" THEN SPEED VAL(SETTING$)
     IF OPTION$ = "Density" THEN DENSITY VAL(SETTING$)
     IF OPTION$ = "Print Mode" THEN GOSUB SET_PRINT_MODE
     IF OPTION$ = "Offset" THEN OFFSET VAL(SETTING$) / DPI
     IF OPTION$ = "Country" THEN GOSUB SET_COUNTRY
ENDIF

GOTO MAINLOOP

:SET_PRINT_MODE
IF SETTING$ = "NONE" THEN SET TEAR OFF
IF SETTING$ = "TEAR OFF" THEN SET TEAR ON
IF SETTING$ = "PEEL OFF" THEN SET PEEL ON
IF SETTING$ = "CUT OFF" THEN SET CUTTER ON
RETURN
```

```
:SET_COUNTRY
IF SETTING$ = "007" THEN COUNTRY 007
IF SETTING$ = "031" THEN COUNTRY 031
IF SETTING$ = "033" THEN COUNTRY 033
IF SETTING$ = "034" THEN COUNTRY 034
IF SETTING$ = "045" THEN COUNTRY 045
RETURN

EOP

RUN "DEMO.BAS"
```

結果



6　ラベルフォーマットコマンド

6.1 BAR

説明

罫線を描きます。

構文

```
BAR x,y,width,height
```

パラメーター	説明
x	左上隅の x 座標（ドット）
y	左上隅の y 座標（ドット）
width	罫線の幅（ドット）
height	罫線の高さ（ドット）

備考:

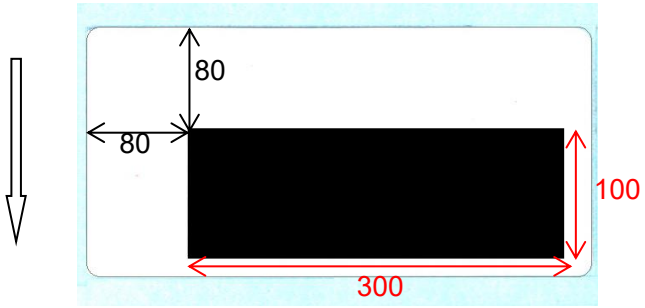
- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット
- 推奨される罫線の最大高さは 4 インチ幅で 12 ミリです。12 ミリを超える罫線の高さは電源供給に障害をきたし、また印字品質に影響を及ぼす可能性があります。
- 最大印字比率はモデルにより異なります。デスクトッププリンター—は最大 20%で、インダストリアルプリンター—は最大 30%です。

例

サンプルコード

```
SIZE 50 mm,25 mm
GAP 3 mm,0
DIRECTION 1
CLS
BAR 80,80,300,100
PRINT 1,1
```

結果



関連コマンド

BOX

6.2 BARCODE

説明

バーコードを定義します。利用可能なバーコードは下記の一覧表に記載されています。

Code Type	説明	狭さ：幅					最大データ長
		1:1	1:2	1:3	2:5	3:7	
128	Code 128、コードサブセット自動切替	V					
128M	Code 128、コードサブセット手動切替	V					
EAN128	EAN-128、コードサブセット自動切替	V					
EAN128M	EAN-128（GS1-128）、コードサブセット手動切替	V					
25	Interleaved 2-of-5		V	V	V		長さは偶数
25C	Interleaved 2-of-5 チェックディジット付き		V	V	V		長さは奇数
25S	Standard 2-of-5		V	V	V		
25I	Industrial 2-of-5		V	V	V		
39	Code 39、スタンダードとフルアスキーモードとの自動切替		V	V	V		
39C	Code 39 チェックディジット付き		V	V	V		
93	Code 93			V			
EAN13	EAN-13	V					12
EAN13+2	2 桁付加 EAN-13	V					14
EAN13+5	5 桁付加 EAN-13	V					17
EAN8	EAN-8	V					7
EAN8+2	2 桁付加 EAN-8	V					9
EAN8+5	5 桁付加 EAN-8	V					12
CODA	Codabar (NW-7)		V	V	V		
POST	POSTNET	V					5, 9, 11
UPCA	UPC-A	V					11
UPCA+2	2 桁付加 UPC-A	V					13
UPA+5	5 桁付加 UPC-A	V					16
UPCE	UPC-E	V					6
UPCE+2	2 桁付加 UPC-E	V					8
UPE+5	5 桁付加 UPC-E	V					11
MSI	MSI		V	V	V		
MSIC	MSI チェックディジット付き		V	V	V		
PLESSEY	Plessey		V	V	V		
CPOST	Datalogic 2-of-5 (China post)					V	
ITF14	ITF-14		V	V	V		13
EAN14	EAN-14	V					13
11	Code 11 (USD-8)		V	V	V		
TELEPEN	Telepen		V	V	V		
TELEPENN	Telepen 数字のみ		V	V	V		
PLANET	Planet	V					
CODE49	Code 49	V					
DPI	Deutsche Post Identcode		V	V	V		11
DPL	Deutsche Post Leitcode		V	V	V		13
LOGMARS	Code 39 特別仕様		V	V	V		

構文

```
BARCODE X,Y, "code type",height,human readable,rotation,narrow,wide,[alignment,]
"content "
```

パラメーター	説明
X	バーコードの x 座標
Y	バーコードの y 座標
code type	
128	Code 128、コードサブセット A, B, C を自動切替
128M	Code 128、コードサブセット A, B, C を手動切替

制御コード	A	B	C
096	FNC3	FNC3	NONE
097	FNC2	FNC2	NONE
098	SHIFT	SHIFT	NONE
099	CODE C	CODE C	NONE
100	CODE B	FNC4	CODE B
101	FNC4	CODE A	CODE A
102	FNC1	FNC1	FNC1
103	Start (CODE A)		
104	Start (CODE B)		
105	Start (CODE C)		

３つの制御コードが後に続く制御コードの最初の文字として「！」を使用します。サブセット開始が設定されていない場合、初期値のサブセット開始は B となります。

EAN128	EAN128: Code 128、コードサブセット A, B, C を自動切替
EAN128M	EAN128(GS1-128): Code 128、コードサブセット A, B, C を手動切替
25	Interleaved 2-of-5
25C	Interleaved 2-of-5 チェックディジット付き
25S	Standard 2-of-5
25I	Industrial 2-of-5
39	Code 39 フルアスキー
39C	Code 39 チェックディジット付き フルアスキー
39S	Code 39 スタンダード
93	Code 93
EAN13	EAN-13
EAN13+2	2 桁付加 EAN-13
EAN13+5	5 桁付加 EAN-13
EAN8	EAN-8
EAN8+2	2 桁付加 EAN-8
EAN8+5	5 桁付加 EAN-8
CODA	Codabar(NW-7)
POST	POSTNET
UPCA	UPC-A
UPCA+2	2 桁付加 UPC-A
UPCA+5	5 桁付加 UPC-A
UPCE	UPC-E
UPCE+2	2 桁付加 UPC-E
UPCE+5	5 桁付加 UPC-E
CPOST	China post code
MSI	MSI code
MSIC	MSI チェックディジット付き
PLESSEY	Plessey code
ITF14	ITF-14 code
EAN14	EAN-14 code
11	Code 11(USD-8)
TELEPEN	Telepen code

TELEPEN N	Telepen code、数字のみ
PLANET	Planet code
CODE49	Code 49
DPI	Deutsche Post Identcode
DPL	Deutsche Post Leitcode

Height　バーコード高さ　（ドット）

human readable
0: 非表示
1: 左揃えの可読文字
2: 中央揃えの可読文字
3: 右揃えの可読文字

rotation
0　：回転なし
90　：時計回りに 90 度回転
180：時計回りに 180 度回転
270：時計回りに 270 度回転

narrow　ナローバーの幅　（ドット）

wide　ワイドバーの幅（ドット）

	n : w 1:1	n : w 1:2	n : w 1:3	n : w 2:5	n : w 3:7
128	10x	-	-	-	-
EAN128	10x	-	-	-	-
EAN128M	10x				
25	-	10x	10x	5x	-
25C	-	10x	10x	5x	-
25S		10x	10x	5x	
25I		10x	10x	5x	
39	-	10x	10x	5x	-
39C	-	10x	10x	5x	-
93	-	-	10x	-	-
EAN13	8x	-	-	-	-
EAN13+2	8x	-	-	-	-
EAN13+5	8x	-	-	-	-
EAN 8	8x	-	-	-	-
EAN 8+2	8x	-	-	-	-
EAN 8+5	8x	-	-	-	-
CODA	-	10x	10x	5x	-
POST	1x	-	-	-	-
UPCA	8x	-	-	-	-
UPCA+2	8x	-	-	-	-
UPCA+5	8x	-	-	-	-
UPCE	8x	-	-	-	-
UPCE+2	8x	-	-	-	-
UPCE+5	8x	-	-	-	-
CPOST	-	-	-	-	1x
MSI	-	-	10x	-	-
MSIC			10x		-
PLESSY	-	-	10x	-	-
ITF14	-	10x	10x	5x	-
EAN14	8x	-	-	-	-
11	-	10x	10x	5x	-

alignment　バーコードの配置を指定します
0：初期値（左）
1：左
2：中央
3：右

content　バーコードの内容

備考：
バーコード内容の最大桁数に留意してください。

コードタイプ	文字設定	最大データ長
128	CODE128 用の文字設定を参照	-
128M	CODE128 用の文字設定を参照	-
EAN128	CODE128 用の文字設定を参照	-
EAN128M	CODE128 用の文字設定を参照	-
25	0123456789	長さは偶数
25C	0123456789	長さは奇数
25S	0123456789	
25I	0123456789	
39 I	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ-.$/+%	-
39 I Full ASCII	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&’()*+,-./:;⇔?@[\]^_`abcdefghijklmnopqrstuvwxyz{|}~	-
93	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&’()*+,-./:;⇔?@[\]^_`abcdefghijklmnopqrstuvwxyz{|}~	-
EAN13	0123456789	12
EAN13+2	0123456789	14
EAN13+5	0123456789	17
EAN8	0123456789	7
EAN8+2	0123456789	9
EAN8+5	0123456789	12
CODA	0123456789-$:/.+	-
POST	0123456789	5, 9, 11
UPCA	0123456789	11
UPCA+2	0123456789	13
UPA+5	0123456789	16
UPCE	0123456789	6
UPCE+2	0123456789	8
UPE+5	0123456789	11
MSI	0123456789	-
MSIC	0123456789	-
PLESSEY	0123456789	-
CPOST	0123456789	-
ITF14	0123456789	13
EAN14	0123456789	13
11	0123456789-	-
TELEPEN	ASCII 0 to 127	30
TELEPENN	0123456789	60
PLANET	0123456789	38
CODE49	ASCII 0 to 127	81
DPI	0123456789	11
DPL	0123456789	13
LOGMARS	0123456789[Space]ABCDEFGHIJKLMNOPQRSTUVWXYZ-.$/+%	-

CODE 128 用の文字設定

値	128A	128B	128C	値	128A	128B	128C	値	128A	128B	128C
0	space	space	00	36	D	D	36	72	BS	h	72
1	!	!	01	37	E	E	37	73	HT	i	73
2	“	“	02	38	F	F	38	74	LF	j	74
3	#	#	03	39	G	G	39	75	VT	k	75
4	$	$	04	40	H	H	40	76	FF	l	76
5	%	%	05	41	I	I	41	77	CR	m	77
6	&	&	06	42	J	J	42	78	SO	n	78
7	‘	‘	07	43	K	K	43	79	SI	o	79
8	(	(	08	44	L	L	44	80	DLE	p	80
9	)	)	09	45	M	M	45	81	DC1	q	81
10	*	*	10	46	N	N	46	82	DC2	r	82
11	+	+	11	47	O	O	47	83	DC3	s	83
12	,	,	12	48	P	P	48	84	DC4	t	84
13	-	-	13	49	Q	Q	49	85	NAK	u	85
14	.	.	14	50	R	R	50	86	SYN	v	86
15	/	/	15	51	S	S	51	87	ETB	w	87
16	0	0	16	52	T	T	52	88	CAN	x	88
17	1	1	17	53	U	U	53	89	EM	y	89
18	2	2	18	54	V	V	54	90	SUB	z	90
19	3	3	19	55	W	W	55	91	ESC	{	91
20	4	4	20	56	X	X	56	92	FS	|	92
21	5	5	21	57	Y	Y	57	93	GS	}	93
22	6	6	22	58	Z	Z	58	94	RS	~	94
23	7	7	23	59	[	[	59	95	US	DEL	95
24	8	8	24	60	\	\	60	96	FNC 3	FNC 3	96
25	9	9	25	61	]	]	61	97	FNC 2	FNC 2	97
26	:	:	26	62	^	^	62	98	Shift B	Shift A	98
27	;	;	27	63	_	_	63	99	Code C	Code C	99
28	<	<	28	64	NUL	`	64	100	Code B	FNC4	Code B
29	=	=	29	65	SOH	a	65	101	FNC 4	Code A	Code A
30	>	>	30	66	STX	b	66	102	FNC 1	FNC 1	FNC 1
31	?	?	31	67	ETX	c	67	103	Start Code A		
32	@	@	32	68	EOT	d	68	104	Start Code B		
33	A	A	33	69	ENQ	e	69	105	Start Code C		
34	B	B	34	70	ACK	f	70				
35	C	C	35	71	BEL	g	71				

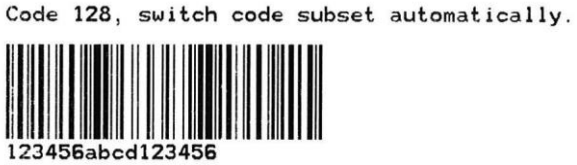
例

サンプルコード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "2",0,1,1, "Human readable
alignment"
BARCODE 10,50, "128",100,1,0,2,2,"left"
BARCODE 10,50,
  "128",100,2,0,2,2,"center"
BARCODE 610,50,
 "128",100,3,0,2,2,"right"
PRINT 1
```

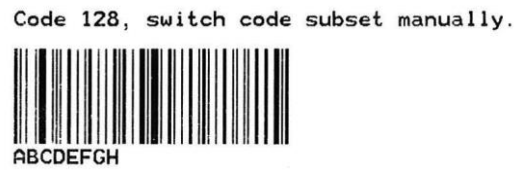
結果

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "2",0,1,1, "Code 128,
switch code subset automatically. "
BARCODE 10,50, "128",100,1,0,2,2,
"123456abcd123456"
PRINT 1
```



```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "2",0,1,1, "Code 128,
switch code subset manually."
BARCODE 10,50, "128M",100,1,0,2,2,
"!104!096ABCD!101EFGH"
PRINT 1
```

備考：
上の CODE 128M はスタートキャラクタ CODE-B を使用しています。次のCODE 128のファンクションキャラクタ FNC3、そしてCODE A サブセットとしてコード化されたABCDキャラクタ・EFGHキャラクタが続きます。




```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 100,10, "2",0,1,1, "EAN-128M(GS1-
128)"
BARCODE
100,50,"EAN128M",100,1,0,2,4,"011458901
2345673"+"!102"+"17161231"+"!102"+"3010
"+"!102"+"10ABC"
PRINT 1
```

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "2",0,1,1, "TELEPEN"
BARCODE 10,50, "TELEPEN",100,1,0,2,6,
"abcd1234ABCD"
PRINT 1
```

```
SIZE 4,4
GAP 0,0
DIRECTION 1
CLS
TEXT 400,26, "2",0,1,1,2, "TELEPEN
Number"
BARCODE 400,50,
"TELEPENN",60,2,0,2,6,2, "1234567890"
TEXT 400,136, "2",0,1,1,2, "Code 11"
BARCODE 400,160, "11",60,2,0,2,6,2,
"1234567890"
TEXT 400,246, "2",0,1,1,2, "PLANET"
BARCODE 400,270, "PLANET",60,2,0,2,2,2,
"12345678901"
TEXT 400,356, "2",0,1,1,2, "Deutsche
Post Identcode."
BARCODE 400,380, "DPI",60,2,0,2,6,2,
"12345678901"
TEXT 400,466, "2",0,1,1,2, "Deutsche
Post Leitcode. "
BARCODE 400,490, "DPL",60,2,0,2,6,2,
"123456789012"
```

TEXT 400,576, "2",0,1,1,2, "Code 49" BARCODE 400,600, "CODE49",60,2,0,2,2,2, "1234567890" PRINT 1	

6.3 TLC39

説明

TLC39（TCIF Linked Barcode 3 of 9）バーコードを定義します。

構文

```
TLC39 x,y,rotation,[height,]narrow,]wide,]cellwidth,]cellheight,] "ECI
number,Serial number & additional data"
```

パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
rotation	0 ：回転なし 90 ：時計回りに 90 度回転 180：時計回りに 180 度回転 270：時計回りに 270 度回転
height	ドット数での Code39 の高さ（初期値は 40）
narrow	ドット数での Code39 のナローバーの幅（初期値は 2）
wide	ドット数での Code39 のワイドバーの幅（初期値は 4）
cellwidth	ドット数での MicroPDF417 のセルの幅（初期値は 2）
cellheight	ドット数ででの MicroPDF417 のセルの幅（初期値は 4）
ECI number	Code39 生成時に使用される場合は 6 桁
Serial number & additional data	英数字は Micro-PDF417 用

備考：
コンマ（“）は ECI number と Serial number & additional data の間で必要となります。

例

サンプルコード

```
SIZE 4,1.2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "TLC39 code"
TLC39 10,50,0, "123456,SN00000001,00601,01501"
TLC39 310,50,0,80,3,6,3,4, "123456,SN00000001,00601,01501"
PRINT 1
```

結果

6.4 BITMAP

説明

ビットマップイメージを（BMP グラフィックファイルとは対照的に）描きます。

構文

```
BITMAP X,Y,width,height,mode,bitmap data…
```

パラメーター	説明
X	開始位置の x 座標（ドット）
Y	開始位置の y 座標（ドット）
width	イメージ幅（バイト）
height	イメージ高さ（ドット）
mode	グラフィックモードは下記 0: OVERWRITE（上書き） 1: OR（論理和） 2: XOR（排他的論理和）
bitmap data	ビットマップデータ

例

→ X size　2 bytes

→ Y size　16 dots

	1-byte								2-byte							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
6	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1
7	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1
8	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1
9	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1	1
10	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
11	0	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1
12	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1
13	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	1
14	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1
15	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
16	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

	X 軸			
Y 軸	1 バイト		2 バイト	
	バイナリ	16 進数	バイナリ	16 進数
1	00000000	00	00000000	00
2	00000000	00	00000000	00
3	00000000	00	00000000	00
4	00000111	07	11111111	FF
5	00000011	03	11111111	FF
6	00010001	11	11111111	FF
7	00011000	18	11111111	FF
8	00011100	1C	01111111	7F
9	00011110	1E	00111111	3F
10	00011111	1F	00011111	1F
11	00011111	1F	10001111	8F
12	00011111	1F	11000111	C7
13	00011111	1F	11100011	E3
14	00011111	1F	11110111	F7
15	00011111	1F	11111111	FF
16	00011111	1F	11111111	FF

サンプルコード (ASCII)	16 進数	結果
SIZE 4,2 GAP 0,0 CLS BITMAP 200,200,2,16,0, ?? ??-????? PRINT 1,1	53 49 5A 45 20 34 2C 32 0D 0A 47 41 50 20 30 2C 30 0D 0A 43 4C 53 0D 0A 42 49 54 4D 41 50 20 32 30 30 2C 32 30 30 2C 32 2C 31 36 2C 30 2C 00 00 00 00 00 00 07 FF 03 FF 11 FF 18 FF 1C 7F 1E 3F 1F 1F 1F 8F 1F C7 1F E3 1F E7 1F FF 1F FF 0D 0A 50 52 49 4E 54 20 31 2C 31 0D 0A	

関連コマンド

PUTBMP, PUTPCX

6.5 BOX

説明

長方形を描きます。

構文

```
BOX x,y,x_end,y_end,line thickness[,radius]
```

パラメーター	説明
x	左上隅の x 座標（ドット）
y	左上隅の y 座標（ドット）
x_end	右下隅の x 座標（ドット）
y_end	右下隅の y 座標（ドット）
line thickness	線の太さ（ドット）
radius	オプション：丸角を指定します。初期値は 0。

備考：

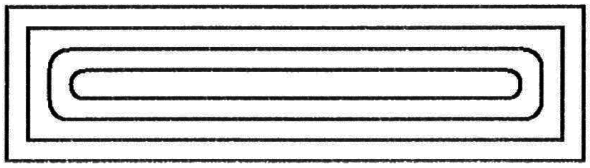
- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi：1 mm = 24 ドット
- 推奨される長方形の最大太さは 4 インチ幅ラベルで 12mm です。12mm を超える長方形の太さは電源供給に障害をきたし、印字品質に影響を及ぼす可能性があります。
- 最大印字比率はモデルにより異なります。デスクトッププリンター—は最大 20%で、インダストリアルプリンター—は最大 30%です。

例

サンプルコード

```
SIZE 4,1.1
CLS
BOX 60,60,610,210,4
BOX 80,80,590,190,4
BOX 100,100,570,170,4,20
BOX 120,120,550,150,4,20
PRINT 1
```

結果



関連コマンド

BAR

6.6 CIRCLE

説明

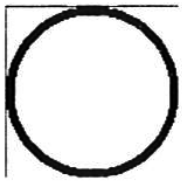
円を描きます。

構文

```
CIRCLE X_start,Y_start,diameter,thickness
```

パラメーター	説明
X_start	左上隅の x 座標（ドット）
Y_start	左上隅の y 座標（ドット）
diameter	円の直径（ドット）
thickness	円の太さ（ドット）

例

サンプルコード	結果
SIZE 80 mm,30 mm GAP 0,0 DIRECTION 1 CLS BAR 250,20,100,1 BAR 250,20,1,100 CIRCLE 250,20,100,5 PRINT 1	

6.7 ELLIPSE

説明

楕円（長円）を描きます。

構文

```
ELLIPSE x,y,width,height,thickness
```

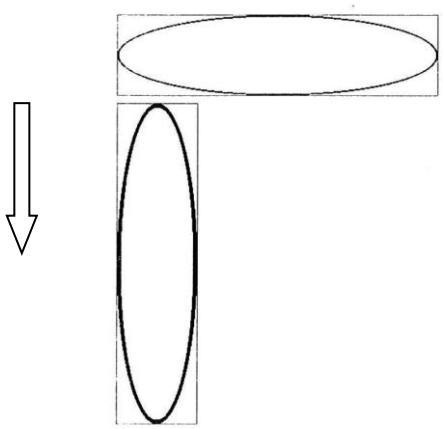
パラメーター	説明
x	左上隅の x 座標（ドット）
y	左上隅の y 座標（ドット）
width	楕円の幅（ドット）
height	楕円の高さ（ドット）
thickness	楕円の太さ（ドット）

例

サンプルコード

```
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
BOX 10,10,410,110,1
ELLIPSE 10,10,400,100,2
BOX 10,120,110,520,1
ELLIPSE 10,120,100,400,5
PRINT 1
```

結果



6.8 CODABLOCK F mode

説明

CODABLOCK F モードのバーコードを定義します。

構文

```
CODABLOCK x,y,rotation,[row height,]module width,] "content"
```

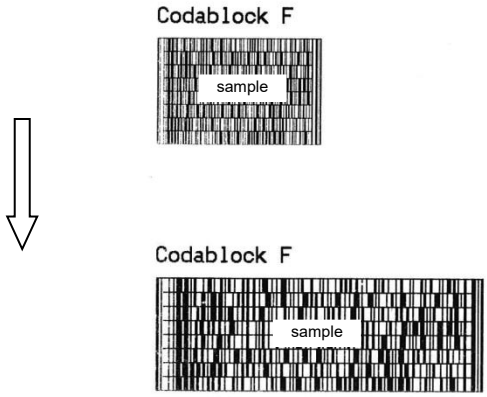
パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
rotation	0 ：回転なし 90 ：時計回りに 90 度回転 180：時計回りに 180 度回転 270：時計回りに 270 度回転
row height	行の高さ x モジュール幅に相当する個々の行の高さ（初期値：8）
module width	CODABLOCK のナローバーの幅　ドット数（初期値：2）
content	CODABLOCK バーコードの内容

例

サンプルコード

```
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Codablock F"
CODABLOCK 10,50,0, "With the "At your side." spirit in mind. the Brother
Group aims to continually create value."
PRINT 1
CLS
TEXT 10,10, "3",0,1,1, "Codablock F"
CODABLOCK 10,50,0,16,1, "With the "At your side." spirit in mind. the
Brother Group aims to continually create value."
PRINT 1
```

結果



6.9 DMATRIX

説明

DataMatrix 2D バーコードを定義します。ECC200 エラー修正のみサポートされます。

構文

```
DMATRIX x,y,width,height,[c#,x#,r#,a#,row,col,] "content"
```

パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
width	バーコード領域の想定幅（ドット）
height	バーコード領域の想定高さ（ドット）
c#	エスケープシーケンス制御文字（10 進数） 例：C126 は以下を意味する一 (1) ~X は制御文字のためのシフト文字です。

~X	Hex	ASCII	~X	HEX	ASCII	~X	HEX	ASCII	~X	HEX	ASCII
~@	00	NUL	~H	08	BS	~P	10	DLE	~X	18	CAN
~A	01	SOH	~I	09	HT	~Q	11	DC1	~Y	19	EM
~B	02	STX	~J	0A	LF	~R	12	DC2	~Z	1A	SUB
~C	03	ETX	~K	0B	VT	~S	13	DC3	~[	1B	ESC
~D	04	EOT	~L	0C	FF	~T	14	DC4	~\	1C	FS
~E	05	ENQ	~M	0D	CR	~U	15	NAK	~]	1D	GS
~F	06	ACK	~N	0E	SO	~V	16	SYN	~^	1E	RS
~G	07	BEL	~O	0F	SI	~W	17	ETB	~_	1F	US

(2) ~1 は FNC1 を意味します。
(3) ~dNNN は符号語に対して ASCII10 進数 NNN を作成します。これは 000 から 255 までの 3 桁を指定します。
(4) ~ in data は~~により符号化されます。

X#	モジュールサイズ（ドット）
r#	回転 0 ：回転なし 90 ：時計回りに 90 度回転 180：時計回りに 180 度回転 270：時計回りに 270 度回転
a#	0 ：正方形（初期値） 1 ：長方形
row	行のシンボルサイズ: 10～144
col	列のシンボルサイズ: 10～144
content	DataMatrix 2D バーコードの内容

備考:
DataMatrix ２次元バーコードのスタンダードシンボルサイズの場合、下記のリストをご参照ください。

正方形			長方形
10 x 10	26 x 26	72 x 72	8 x 18
12 x 12	32 x 32	80 x 80	8 x 32
14 x 14	36 x 36	88 x 88	12 x 26
16 x 16	40 x 40	96 x 96	12 x 36
18 x 18	44 x 44	104 x 104	16 x 36
20 x 20	48 x 48	120 x 120	16 x 48
22 x 22	52 x 52	132 x 132	
24 x 24	64 x 64	144 x 144	

例

サンプルコード

```
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
DMATRIX 10,110,400,400, "DMATRIX Example 1"
DMATRIX 310,110,400,400,x6, "DMATRIX Example 2"
DMATRIX 10,310,400,400,x8,18,18, "DMATRIX Example 3"
PRINT 1,1
```

結果

サンプルコード （FNC）

```
SIZE 4,1
GAP 0,0
CLS
DIRECTION 1
DMATRIX
100,50,100,100,c126,x6,18,18, "~1241sPn~110sLot~130sQ
ty "
PRINT 1
```

サンプルコード （長方形）

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
DMATRIX 100,110,600,600,a1,"DMATRIX Example 1"
PRINT 1,1
```

6.10 ERASE

説明

イメージバッファ内の指定領域を消去します。

構文

```
ERASE x,y,x_width,y_height
```

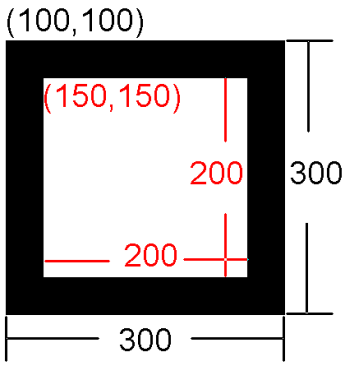
パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
x_width	x 軸方向の領域幅（ドット）
y_height	y 軸方向の領域高さ（ドット）

例

サンプルコード

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
BAR 100,100,300,300
ERASE 150,150,200,200
PRINT 1,1
```

結果



関連コマンド

CLS

6.11MAXICODE

説明

2D Maxicode を定義します。

構文

MAXICODE x,y,mode,[class,country,post,Lm,] "content"	
MAXICODE x,y,mode,class,country,postal code, "content"	モード 2 または 3 では、country が 840 の場合、郵便コードは 99999,9999 形式になります。他の国の場合、このコード 6 つの英数字までとなります。
MAXICODE x,y,mode,[Lm,] "content"	モード 4 または 5 では、AIM 特別形式が対応されています

パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
mode	2,3,4,5
class	サービスのクラス、3 桁（モード 2 または 3）
country	国コード、3 桁（モード 2 または 3）
post	郵便コード（モード 2 または 3） Mode 2(USA): 5 桁+4 桁の数 Mode 3(Canada): 6 つの英数字郵便コードは二重引用符に含まれます
Lm	式の長さ（二重引用符は無視されます）、1≤m≤138（このパラメーターはモード 4 と 5 専用です）
content	2D Maxicode の内容 **備考:** パラメーター Lm が使用された場合、二重引用符 (")は不要です。

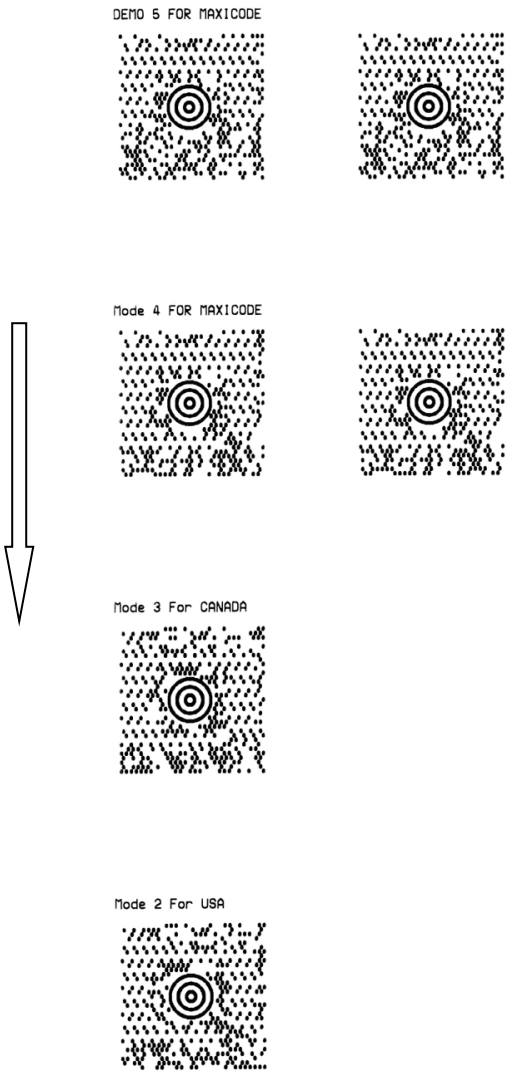
例

サンプルコード

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS

REM *****Mode 2 For USA*****
MAXICODE 110,100,2,300,840,06810,7317, "DEMO 2 FOR USA MAXICODE"
TEXT 100,50, "3",0,1,1, "Mode 2 For USA"
PRINT 1,1

REM *****Mode 3 For Canada*****
CLS
MAXICODE 110,100,3,300,863, "107317","DEMO 3 FOR CANADA MAXICODE"
TEXT 100,50, "3",0,1,1, "Mode 3 For CANADA"
PRINT 1,1

REM *****MODE4*****
CLS
```

```
MAXICODE 110,100,4, "DEMO 4 FOR MAXICODE"
MAXICODE 600,100,4,L19,DEMO 4 FOR MAXICODE
TEXT 100,50, "3",0,1,1, "Mode 4 FOR MAXICODE"
PRINT 1,1

REM *****MODE 5*****
CLS
MAXICODE 110,100,5, "DEMO 5 FOR MAXICODE"
MAXICODE 600,100,5,L19,DEMO 5 FOR MAXICODE
TEXT 100,50, "3",0,1,1, "DEMO 5 FOR MAXICODE"
PRINT 1
```

結果



6.12 PDF417

説明

PDF417 バーコードを定義します。

構文

```
PDF417 x,y,width,height,rotate,[option], "content"
```

パラメーター		説明
x		開始位置の x 座標（ドット）
y		開始位置の y 座標（ドット）
width		想定幅（ドット）
height		想定高さ（ドット）
rotate		反時計回り 0 ： 回転なし 90 ：90 度回転 180 ：180 度回転 270 ：270 度回転
option		
	P	データ圧縮方法 0: 自動エンコーディング 1: バイナリモード
	E	エラー修正レベル（範囲: 0~8）
	M	バーコード領域のセンターパターン 0: このパターンは領域の左上部に印刷されます 1: このパターンは領域の中央に印刷されます
	Ux,y,c	可読文字（ヒューマンリーダブル） x: x 座標での可読文字 y: y 座標での可読文字 c: 1 行毎の可読文字の最大文字数
	W	ドット数でのモジュール幅（範囲: 2~9）
	H	ドット数でのバーの高さ（範囲: 4~99）
	R	行の最大数
	C	列の最大数
	T	打ち切り 0: 打ち切りなし 1: 打ち切り
	Lm	式の長さ、1≤m≤2048 （“ は含まない）
content		PDF417 2D バーコードの内容 **備考：** パラメーター Lm が使用された場合、二重引用符 (")は不要です

例

サンプルコード	結果
SIZE 4,1 GAP 0,0 DIRECTION 1 REM *****WITHOUR OPTIONS***** CLS PDF417 50,50,400,200,0, "Without Options" PRINT 1,1	
SIZE 4,1.5 GAP 0,0 DIRECTION 1 REM *****OPTION:E3***** CLS PDF417 50,50,400,200,0,E3, "Error correction level:3" PRINT 1,1 REM *****OPTION:E4***** CLS PDF417 50,50,400,200,0,E4, "Error correction level:4" PRINT 1,1	
SIZE 4,1.5 GAP 0,0 DIRECTION 1 REM *****OPTION:E4 W4***** CLS PDF417 50,50,600,600,0,E4,W4, "Error correction level:4 module width 4 dots" PRINT 1,1 REM *****OPTION:E4 W4 H4***** CLS PDF417 50,50,600,600,0,E4,W4,H4, "Error correction level:4 module width 4 dots bar height 4 dots" PRINT 1,1	

```
SIZE 4,1.5
GAP 0,0
DIRECTION 1

REM *****OPTION:E4 W4 H4 R40 C4
T1*****
CLS
PDF417
50,50,800,800,0,E4,W4,H4,R40,C4,T1
, "Error correction level:4
Module Width 4 dots
Bar Height 4 dots
Maximum Number of Rows:5 Rows
Maximum number of columns:90 Cols
Truncation:1"
PRINT 1,1
```

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1

REM *****OPTION:P1 E4 M1
U50,300,50,W4,H4,R60,C4,T0,L297***
**
CLS
PDF417
50,50,900,600,0,P1,E4,M1,U50,300,5
0,W4,H4,R60,C4,T0,L297,Data
compression method: P1
Error correction level: E4
Center pattern in barcode area: M1
Human Readable: Yes: U50,300,50
Module Width 4 dots: W4
Bar Height 4 dots: H4
Maximum Number of Rows: 60 Rows:
R60
Maximum number of columns: 4 Cols:
C4
Truncation:1: T0
Expression length:297: L297
PRINT 1,1
```

Data compression method: P1 Error correction leve
l: E4 Center pattern in barcode area: M1 Human R
eadable: Yes: U50,300,50 Module Width 4 dots: W4
Bar Height 4 dots: H4 Maximum Number of Rows: 60
Rows: R60 Maximum number of columns: 4 Cols: C4
Truncation:1: T0 Expression length:297: L297

6.13AZTEC

説明

AZTEC バーコードを定義します。

構文

```
AZTEC x,y,rotate,[size,]ecp,]flg,]menu,]multi,]rev,] "content"
AZTEC x,y,rotate,size,ecp,flg,menu,multi,rev,bytes,content
```

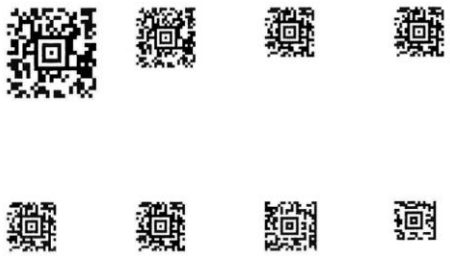
パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
rotate	回転 0 ：回転なし 90 ：90 度回転 180：180 度回転 270：270 度回転
size	要素モジュールサイズ(1～20)、初期値：6
ecp	誤り訂正と印字可能データ数 0 ：誤り訂正（初期値） 1～99 ：誤り訂正のパーセント 101～104 ：データレイヤー数が 1～4 のコンパクトシンボル 201～232 ：データレイヤー数が 1～32 のフルサイズシンボル 300 ：簡素なシンボル（ルーン）
flg	0：ストレートバイト 1：FLG(n)には"<Esc>n"、"<Esc>"には"<Esc><Esc>"を使用します
menu	メニューシンボル（0：いいえ, 1：はい)、初期値：0
multi	シンボル数（1～26)、初期値： 6
rev	反転（0：いいえ, 1：はい)、初期値：0
bytes	コンテンツの長さ
content	AZTEC 2D バーコードのコンテンツ **備考：** bytes パラメーターが使用された場合、二重引用符 (")は不要です

例

サンプルコード

```
SIZE 4,2
GAP 0,0
CLS
AZTEC 10,10,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
AZTEC 210,10,0,4,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
AZTEC 410,10,0,4,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
AZTEC 610,10,0,4,1,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
AZTEC
10,310,0,4,1,0,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
AZTEC
210,310,0,4,1,0,0,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
AZTEC
410,310,0,4,1,0,0,1,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
"
AZTEC 610,310,0,4,1,0,0,1,1,10,1234567890
PRINT 1
```

結果



6.14MPDF417

説明

Micro PDF 417 を定義します。

構文

```
MPDF417 x,y,rotate,[Wn,][Hn,][Cn,] "content"
```

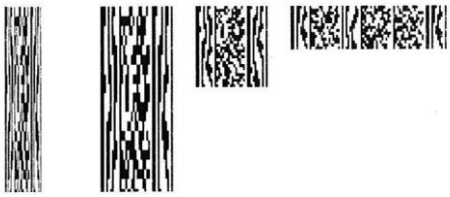
パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
rotate	回転 0 ：回転なし 90 ：90 度回転 180：180 度回転 270：270 度回転
Wn	オプション：ドット数によるモジュールサイズ（初期値：1）
Hn	オプション：ドット数によるモジュール高さ（初期値：10）
Cn	オプション：列数。パラメーターが設定されると、プリンターは自動的にバーコードベースのために適切な行を計算します。 0: 自動モード. 1: 列は 1 で、計算される適切な行は 11, 14, 17, 20, 24, 28 2: 列は 2 で、計算される適切な行は 8, 11, 14, 17, 20, 23 , 26 3: 列は 3 で、計算される適切な行は 6, 8, 10, 12, 15, 20, 26, 32, 38, 44 4: 列は 4 で、計算される適切な行は 4, 6, 8, 10, 12, 15, 20, 26, 32, 38, 44
Content	Micro PDF 417 バーコードの内容

例

サンプルコード

```
SIZE 4,1
GAP 0,0
CLS
MPDF417 10,10,0, "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "
MPDF417 110,10,0,W2, "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "
MPDF417 210,10,0,W2,H3, "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "
MPDF417 310,10,0,W2,H3,C3, "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 "
PRINT 1
```

結果



6.15 PUTBMP

説明

プリンターにダウンロードした 1 ビット（白黒）または 8 ビット（256 色）の BMP ファイルを印刷します。感熱モードでは白黒とグレースケール印刷に対応しています。熱転写モードは白黒印刷のみに対応しています。

構文

```
PUTBMP x,y, "filename" [, bpp][, contract]
```

パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
filename	ダウンロードされた BMP ファイル名
bpp	オプション：1 画素のビット数（初期値：1） 1: 1 ビットグラフィック（モノクロモード） 8: 8 ビットグラフィック（256 色）
contrast	オプション：グレースケールのコントラスト（初期値：80） 推奨範囲（60～100）

備考：

- グレースケール印刷は感熱モード時のみ使用できます。
- RJ-2035B/RJ-3035B はモノクロ印刷（1 ビット）のみ対応しています。
- ファイル名と拡張子は、必ず**大文字（半角英数字）**にしてください。（例：FILENAME.BMP）

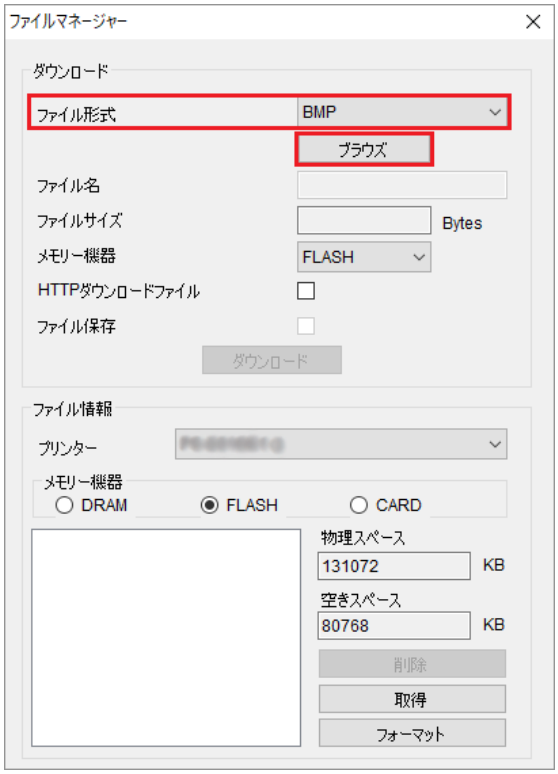
例

次のページに続きます。

DOWNLOAD コマンドまたは BPM (Brother Printer Management Tool) を使用してプリンターに BMP ファイルを転送することができます。

BPM を使用してダウンロードするには、次の手順に従ってください。

1. パソコンとプリンターを接続し、プリンターの電源を入れます。
2. BPM を起動し、「**ファイルマネージャー**」をクリックします。
3. 「**ファイル形式**」から「**BMP**」を選択します。
4. 「**ブラウズ**」をクリックして BMP ファイルを選択し、「**開く**」をクリックします。ファイル名と拡張子は、必ず**大文字（半角英数字）**にしてください。（例：FILENAME.BMP）
5. 「**メモリー機器**」からファイルをダウンロードする場所を選択し、「**ダウンロード**」をクリックしてダウンロードを開始します。

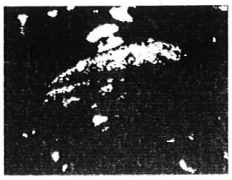


サンプルコード

```
SPEED 2
DENSITY 3
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
PUTBMP 10,10,"SAMPLE.BMP"
BLOCK 10,180,240,100,"2",0,1,1,"bpp and contrast are omitted."
PUTBMP 300,10, "SAMPLE.BMP",1,80
BLOCK 300,180,240,100,"2",0,1,1, "bpp = 1
contrast = 80"
PUTBMP 590,10, "SAMPLE.BMP",8,80
BLOCK 590,180,240,100,"2",0,1,1,"bpp = 8
contrast = 80"
PRINT 1
```

結果

bpp and contrast
are omitted.



bpp = 1
contrast = 80

bpp = 8
contrast = 80

サンプルコード

```
SIZE 2,2
GAP 0,0
CLS
PUTBMP 10,10, "SAMPLE.GRF"
PRINT 1
```

関連コマンド

DOWNLOAD, BITMAP, PUTPCX

6.16 PUTPCX

説明

プリンターにダウンロードした 1 ビット（白黒）または 8 ビット（256 色）の PCX ファイルを印刷します。

構文

```
PUTPCX x,y, "filename"
```

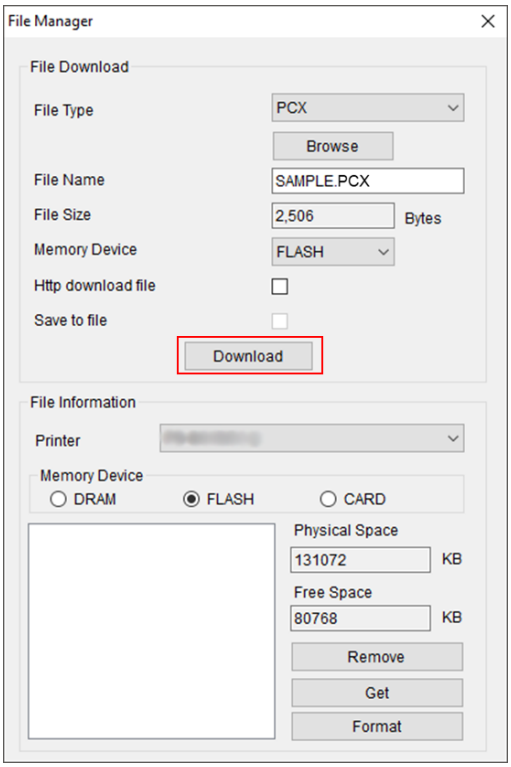
パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
filename	ダウンロードされた PCX ファイル名 ファイル名と拡張子は、必ず**大文字（半角英数字）**にしてください。（例：FILENAME.PCX）

例

DOWNLOAD コマンドまたは BPM (Brother Printer Management Tool) を使用してプリンターに PCX ファイルを転送することができます。

BPM を使用してダウンロードするには、次の手順に従ってください。

1. パソコンとプリンターを接続し、プリンターの電源を入れます。
2. BPM を起動し、「**ファイルマネージャー**」をクリックします。
3. 「**ファイル形式**」から「**PCX**」を選択します。
4. 「**ブラウズ**」をクリックして PCX ファイルを選択し、「**開く**」をクリックします。ファイル名と拡張子は、必ず**大文字（半角英数字）**にしてください。（例：FILENAME.PCX）
5. 「**メモリー機器**」からファイルをダウンロードする場所を選択し、「**ダウンロード**」をクリックしてダウンロードを開始します。



サンプルコード

```
SPEED 2
DENSITY 3
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
PUTPCX 10,10, "SAMPLE.PCX"
PRINT 1
```

結果

関連コマンド

DOWNLOAD, BITMAP, PUTPCX

6.17 QRCODE

説明

QR コードを定義します。

構文

```
QRCODE x,y,ECC Level,cell width,mode,rotation,[justification,]model,]mask,]area]
"content"
```

パラメーター	説明
x	QR コードの左上隅の x 座標
y	QR コードの左上隅の y 座標
ECC level	誤り訂正レベル L　: 7% M　: 15% Q　: 25% H　: 30%
cell width	1~10
mode	自動/手動エンコード A　: 自動 M　: 手動
rotation	0　: 0 度 90　: 90 度 180 : 180 度 270 : 270 度
[justification]	バーコード行端揃え（J1～J9：**サンプルコード**参照)
[model]	M1: オリジナルバージョン（初期値） M2: 機能拡張バージョン(多くのスマートフォンに対応)
[mask]	S0~S8（初期値：S7）
[area]	バーコード領域の最大サイズ(Xdots; e.g. X100)
content	コード化可能な文字セットは以下のとおりです。 コード化可能な文字セット 1) 数値データ(桁数 0~9) 2) 英数字データ 桁数 0-9、大文字 A-Z 9 個の記号（space $ % * + - . / :) 3) 8 ビットバイトデータ JIS X 0201 に準拠する JIS 8 ビット文字セット（ラテンと仮名） 4) 漢字 Shift JIS は 8140_{HEX} –$9FFC_{HEX}$ と $E040_{HEX}$ –$EAA4_{HEX}$ の値を付けます。 詳細は JIS X 0208 の添付資料 1 Annex1 Shift Coded Representation for detail をご参照ください。. シンボル毎のデータ文字（最大シンボルサイズ用）

	モデル 1 (Version 14-L)	モデル 2 (Version 40-L)
数値データ	1,167 文字	7,089 文字
英数字データ	707 文字	4,296 文字
8 ビットバイト	486 文字	2,953 文字
漢字データ	299 文字	1,817 文字

*データ文字列の最初の文字が"A"の場合、その後は英数字データが続きます
*データ文字列で最初の文字が"N"の場合、その後は数値データが続きます
*データ文字列で"最初の文字 B"の場合、その後に続く 4 桁の数字はデータ

数を指定するために使われます。4 桁の数字の後はコード化されるバイナリーデータのバイト数になります。
*データ文字列で最初の文字"K"の場合、その後に続くデータは漢字データです
*データ文字列に "!"を含み、"N", "A", "B", "K"の後に続く場合、特定のコード化可能な文字セットに切り替えられます

手動モード（例）
最初の文字「A」: 英数字データ
```
QRCODE 100,10,L,7,M,0,M1,S1, "ATHE FIRMWARE HAS BEEN UPDATED"
```
最初の文字「N」: 数値データ
```
QRCODE 100,10,M,7,M,0,M1,S2, "N123456"
```
最初の文字「N」: 数値データ、「!」あり、「A」: 英数字データ
```
QRCODE 100,10,Q,7,M,0,M1,S3, "N123456!ATHE FIRMWARE HAS BEEN UPDATED"
```
最初の文字「B」: バイナリデータ、0012 : 12 バイト）
```
QRCODE 100,10,H,7,M,0,M1,S3, "B0012Product name"
```
最初の文字「K」: 漢字データ）
```
QRCODE 100,10,M,7,M,0,M1,S3, "K"
```

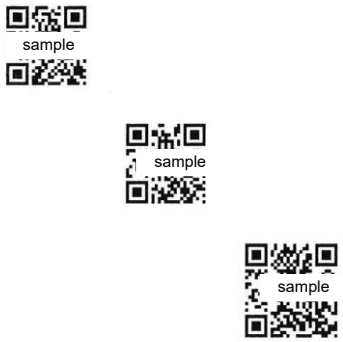
自動モード（例）
```
QRCODE 100,10,M,7,A,0, "THE FIRMWARE HAS BEEN UPDATED"
```

例

サンプルコード	結果
自動モード	
一般文字列 SIZE 4,2.5 GAP 0,0 DIRECTION 1 CLS QRCODE 10,10,H,4,A,0, "ABCabc123" QRCODE 160,160,H,4,A,0, "123ABCabc" QRCODE 310,310,M,4,A,0,M2, "Printer ABCabc123" PRINT 1,1	

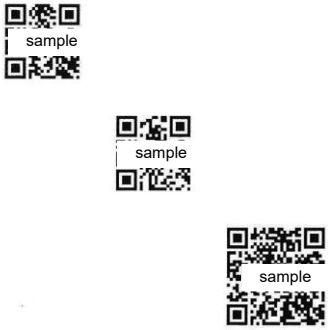
<Enter> character (0Dh, 0Ah)を含む文字列

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,4,A,0, "ABC<Enter>
abc<Enter>
123 "
QRCODE 160,160,H,4,A,0, "123<Enter>
ABC<Enter>
abc"
QRCODE 310,310,H,4,A,0, "Printer<Enter>
ABC<Enter>
abc<Enter>
123"
PRINT 1,1
```



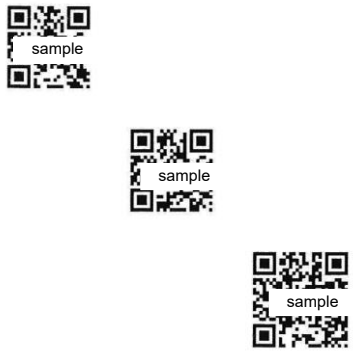
データ文字列連結
(DOWNLOAD ... EOP コマンドを使用します**)**

```
DOWNLOAD "DEMO.BAS"
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,4,A,0, "ABCabc123"
+STR$(1234)
QRCODE 160,160,H,4,A,0, "123ABCabc" +"1234"
QRCODE 310,310,H,4,A,0,
"PrinterABCabc123"+"1234"+"abcd"
PRINT 1,1
EOP
DEMO
```



二重引用符（”）を含む文字列
（(“)ではなく、\["]を使用します）

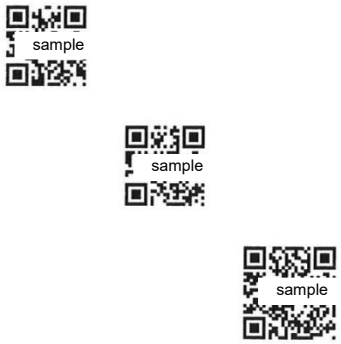
```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,4,A,0, "ABC\["]abc\["]123"
QRCODE 160,160,H,4,A,0, "123\["]ABC\["]abc"
QRCODE 310,310,H,4,A,0,
"\["]Printer\["]ABCabc123"
PRINT 1,1
```



手動モード

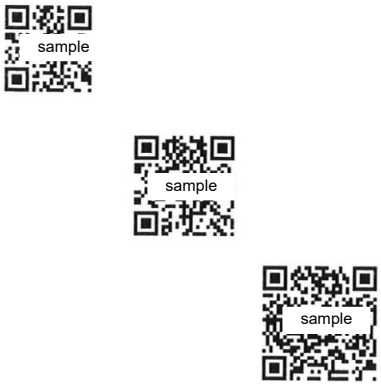
<u>一般文字列</u>

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,4,M,0, "AABC!B0003abc!N123"
QRCODE 160,160,H,4,M,0, "N123!AABC!B0003abc"
QRCODE 310,310,H,4,M,0, "K
Printer!AABC!B0006abc123"
PRINT 1,1
```



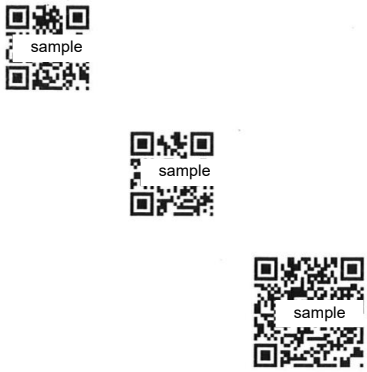
<u><Enter> character を含む文字列</u>
<u>（ <Enter> は 8 ビットバイトデータです）</u>

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,4,M,0,"AABC!B0007<Enter>
abc<Enter>
!N123"
QRCODE 160,160,H,4,M,0,"N123!B0002<Enter>
!AABC!B0005<Enter>
abc"
QRCODE 310,310,H,4,M,0, "K
Printer!B0002<Enter>
!AABC!B0010<Enter>
abc<Enter>
123"
PRINT 1,1
```



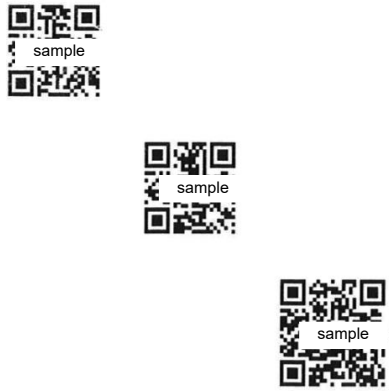
<u>データ文字列連結</u>
<u>**(DOWNLOAD ... EOP コマンドを使用します)**</u>

```
DOWNLOAD "A.BAS"
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE
10,10,H,4,M,0,"AABC!B0006abc123!N"+STR$(1234)
QRCODE
160,160,H,4,M,0,"N123!AABC!B0007abc"+"1234"
QRCODE 310,310,H,4,M,0, "K
Printer!AABC!B0014abc123"+" 1234"+"abcd"
PRINT 1,1
EOP
A
```



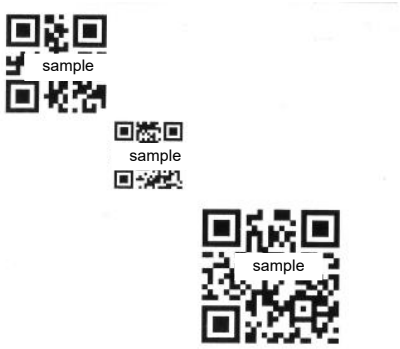
二重引用符（”）を含む文字列
（(“)ではなく、\["]を使用します）

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,4,M,0,
"AABC!B0005\["]abc\["]!N123"
QRCODE 160,160,H,4,M,0,
"N123!B0001\["]!AABC!B0004\["]abc"
QRCODE 310,310,H,4,M,0, "B0001\["]!K
Printer!B0010\["]ABCabc123"
PRINT 1,1
```



スマートフォンデータ文字列

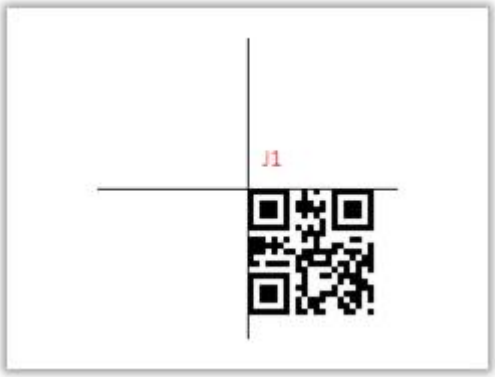
```
DOWNLOAD "A.BAS"
SIZE 3,3
GAP 0,0
DIRECTION 1
CLS
QRCODE 10,10,H,7,M,0,M2,S7,"Aabcd"
QRCODE 170,170,H,4,M,0, M2,"B0008 日本語"
QRCODE 300,300, L, 8, M, 0,
M2,"B0026http://www.brother.com"
PRINT 1,1
EOP
A
```

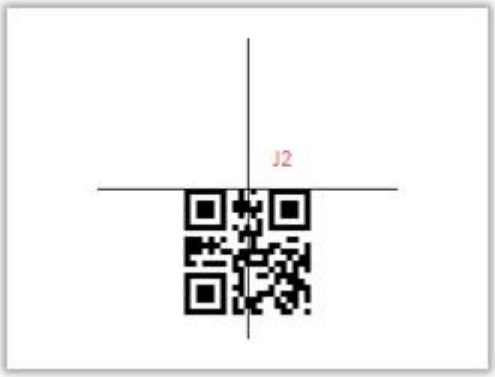


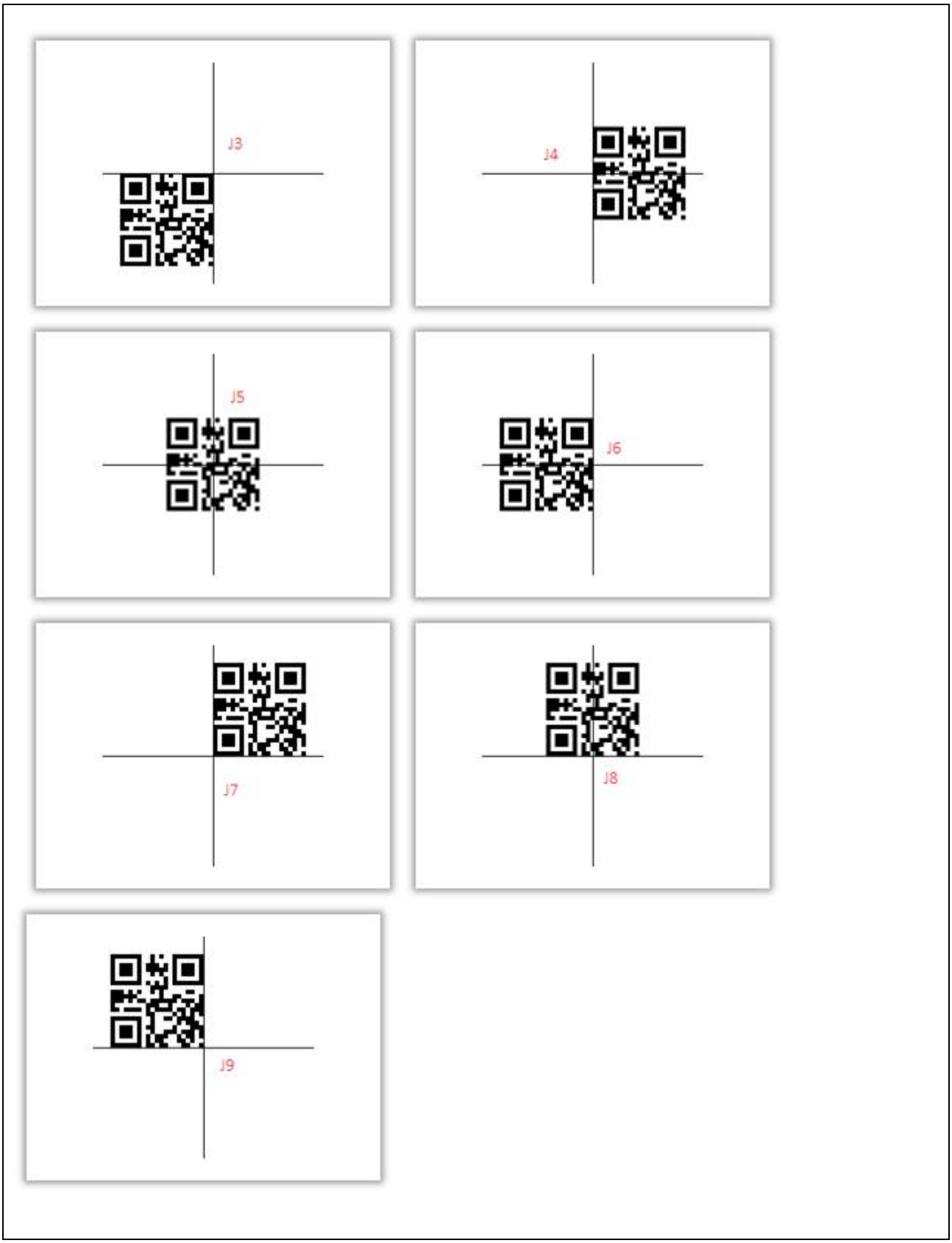
パラメーター [justification] & [area]の文字列

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
BAR 60,120,200,1
BAR 160,20,1,200
QRCODE 160,120,H,10,A,0,X100,J5,"123456789"
PRINT 1,1
```

その他の行端揃え [justification] の結果 (J1~J9)






J3
J4
J5
J6
J7
J8
J9

6.18 RSS

説明

GS1 Databar（RSS）バーコードを定義します。

構文

```
RSS x,y, "sym",rotate,pixMult,sepHt, "content"
RSS x,y, "RSSEXP",rotate,pixMult,sepHt,segWidth, "content"
RSS x,y, "UCC128CCA",rotate,pixMult,sepHt,linHeight, "content"
RSS x,y, "UCC128CCC",rotate,pixMult,sepHt,linHeight, "content"
```

パラメーター — **説明**

x — 開始位置の x 座標（ドット）

y — 開始位置の y 座標（ドット）

sym — シンボロジータイプ

RSS14	RSS14
RSS14T	RSS14 Truncated
RSS14S	RSS14 Stacked
RSS14SO	RSS14 Stacked Omnidirectional
RSSLIM	RSS Limited
RSSEXP	RSS Expanded
UPCA	UPC-A
UPCE	UPC-E
EAN13	EAN-13
EAN8	EAN-8
UCC128CCA	UCC/EAN-128 & CC-A/B
UCC128CCC	UCC/EAN-128 & CC-C

rotate — 回転（0, 90, 180, 270）

pixMult — モジュール幅（ドット、1～10）

下記のバーコードの高さがプリンターにより計算されます

RSS14	33 × pixMult
RSS14T	13 × pixMult.
RSS14S	13 × pixMult.
RSS14SO	33 × pixMult.
RSSLIM	13 × pixMult.
RSSEXP	33 × pixMult.
EAN8	60 × pixMult.
EAN13	74 × pixMult.
UPCA	74 × pixMult.
UPCE	74 × pixMult.

sepHt — セパレーターの行の高さ（1 または 2）

pixMult times sepHt は実際のセパレーターの行の高さです。プリンターにより計算されます。

segWidth — 拡張 RSS のセグメント幅（2～22 の偶数）

linHeight — UCC/EAN-128 の高さ（ドット、1～500）

content — バーコード内容と文字形式

UPCE の内容は下記のとおりです。

*00abc0000hij = abhijc, where c = 0-2

*00abc00000ij = abcij3

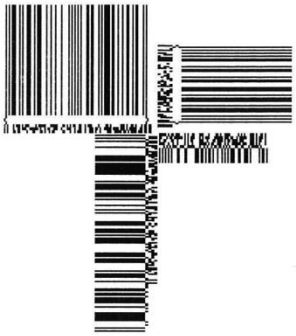
*00abcd00000j = abcdj4

*00abcde0000j = abcdej where j = 5-9

備考：

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット
- 反転された黒色領域の推奨最大高さは 4 インチ幅用紙で 12mm です。12mm を超える反転領域は電源供給に障害をきたし、印字品質に影響を及ぼす可能性があります。
- 最大印字比率はモデルにより異なります。デスクトッププリンター—は最大 20%で、インダストリアルプリンター—は最大 30%です。

例

サンプルコード	結果
SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSS14",0,2,2, "1234567890|ABCDEFG" RSS 300,300,"RSS14T",90,2,2,"1234567890|ABCDE FG" RSS 300,300,"RSS14S",180,2,2,"1234567890|ABCD EFG" RSS 300,300, "RSS14SO",270,2,2, "1234567890|ABCDEFG" PRINT 1,1	
SIZE 100 mm,100 mm GAP 0,0 DIRECTION 1 CLS RSS 300,300, "RSSLIM",0,2,2, "1234567890|ABCDEFG" RSS 300,300, "RSSEXP",90,2,2,22, "1234567890|ABCDEFG" RSS 300,300, "UPCA",180,2,2, "1234567890|ABCDEFG" RSS 300,300, "UPCE",270,2,2, "000|ABCDEFG" PRINT 1,1	

```
SIZE 100 mm,100 mm
GAP 0,0
DIRECTION 1
CLS
RSS
300,300,"EAN13",0,2,2,"123456789012|ABCDE
FG"
RSS
300,300,"EAN8",90,2,2,"1234567|ABCDEFG"
RSS
300,300,"UCC128CCA",180,2,2,25,"123456789
0|ABCDEFG"
RSS
300,300,"UCC128CCC",270,2,2,25,"123456789
0|ABCDEFG"
PRINT 1,1
```

```
SIZE 100 mm, 100 mm
GAP 0,0
DIRECTION 1
CLS
RSS 300,10, "RSSEXP",90,2,2,12,
"8110106141411234562891101201212085010048
0002140256100480003101231910 00"
PRINT 1
```

UPCE モード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
REM UPCE Rule 1: 00abc0000hij = abhijc,
where c = 0-2
RSS
10,10,"UPCE",0,2,2,"001200000456|ABCDEFG"
RSS
210,10,"UPCE",0,2,2,"001210000456|ABCDEFG
"
RSS
410,10,"UPCE",0,2,2,"001220000456|ABCDEFG
"
PRINT 1
```

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
REM UPCE Rule 2: 00abc00000ij = abcij3
RSS
10,10,"UPCE",0,2,2,"001230000045|ABCDEFG"
PRINT 1
SIZE 4,1
CLS
REM UPCE Rule 3: 00abcd00000j = abcdj4
RSS
10,10,"UPCE",0,2,2,"001234000005|ABCDEFG"
PRINT 1
```

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
REM UPCE Rule 4: 00abcde0000j = abcdej
where j = 5-9
RSS
10,10,"UPCE",0,2,2,"001234500005|ABCDEFG"
RSS
160,10,"UPCE",0,2,2,"001234500006|ABCDEFG
"
RSS
310,10,"UPCE",0,2,2,"001234500007|ABCDEFG
"
RSS
460,10,"UPCE",0,2,2,"001234500008|ABCDEFG
"
RSS
610,10,"UPCE",0,2,2,"001234500009|ABCDEFG
"
PRINT 1
```

EAN8 EAN13 UPCA UPCE バーコード高さ

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
RSS 10,10,"EAN8",0,1,1,"1234567|ABCDEFG"
RSS 210,10,
"EAN8",0,2,1,"1234567|ABCDEFG"
RSS 410,10,
"EAN8",0,3,1,"1234567|ABCDEFG"
PRINT 1
```

SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"EAN13",0,1,1,"123456789012|ABCDEFG " RSS 210,10,"EAN13",0,2,1,"123456789012|ABCDEF G" RSS 410,10,"EAN13",0,3,1,"123456789012|ABCDEF G" PRINT 1	
SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"UPCA",0,1,1,"12345678901|ABCDEFG" RSS 210,10,"UPCA",0,2,1,"12345678901|ABCDEFG" RSS 410,10,"UPCA",0,3,1,"12345678901|ABCDEFG" PRINT 1	
SIZE 4,2 GAP 0,0 DIRECTION 1 CLS RSS 10,10,"UPCE",0,1,1,"001200000456|ABCDEFG" RSS 210,10,"UPCE",0,2,1,"001210000456|ABCDEFG " RSS 410,10,"UPCE",0,3,1,"001220000456|ABCDEFG " PRINT 1	

6.19 REVERSE

説明

イメージバッファ内の特定領域を反転させます。

構文

```
REVERSE x_start,y_start,x_width,y_height
```

パラメーター	説明
x_start	開始位置の x 座標（ドット）
y_start	開始位置の y 座標（ドット）
x_width	Ｘ軸の領域幅（ドット）
y_height	Ｙ軸の領域高さ（ドット）

備考：

- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット
- 反転された黒色領域の推奨最大高さは 4 インチ幅用紙で 12mm です。12mm を超える反転領域は電源供給に障害をきたし、印字品質に影響を及ぼす可能性があります。
- 最大印字比率はモデルにより異なります。デスクトッププリンタ—は最大 20%で、インダストリアルプリンタ—は最大 30%です。

例

サンプルコード

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
TEXT 100,100,"3",0,1,1,"REVERSE"
REVERSE 90,90,128,40
PRINT 1,1
```

結果

6.20 DIAGONAL

説明

斜線を描きます。

構文

```
DIAGONAL x1, y1, x2, y2, thickness
```

パラメーター	説明
x1	開始位置の x1 座標（ドット）
y1	開始位置の y1 座標（ドット）
x2	終了位置の x2 座標（ドット）
y2	終了位置の y2 座標（ドット）
thickness	斜線の太さ

備考：

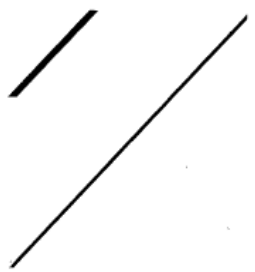
- 200 dpi : 1 mm = 8 ドット
  300 dpi : 1 mm = 12 ドット
  600 dpi : 1 mm = 24 ドット

例

サンプルコード

```
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
DIAGONAL 50, 200, 200, 50, 16
DIAGONAL 50, 500, 500, 50, 8
PRINT 1,1
```

結果



6.21 TEXT

説明

テキストを定義します。

構文

```
TEXT x,y, "font",rotation,x-multiplication,y-multiplication,[alignment,]
"content"
```

パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
font	フォント名

内蔵フォント	
0	CG Triumvirate Bold Condense (Monotype Imaging®) フォント幅と高さは伸縮可能です
1	8×12 ドット固定ピッチフォント
2	12×20 ドット固定ピッチフォント
3	16×24 ドット固定ピッチフォント
4	24×32 ドット固定ピッチフォント
5	32×48 ドット固定ピッチフォント
6	14×19 ドット固定ピッチフォント OCR-B
7	21×27 ドット固定ピッチフォント OCR-B
8	14×25 ドット固定ピッチフォント OCR-A
ROMAN.TTF	CG Triumvirate Bold Condensed (Monotype Imaging®) フォント幅と高さの比率は固定です
1.EFT	EPL2 font 1
2.EFT	EPL2 font 2
3.EFT	EPL2 font 3
4.EFT	EPL2 font 4
5.EFT	EPL2 font 5
A.FNT	ZPL2 font A
B.FNT	ZPL2 font B
D.FNT	ZPL2 font D
E8.FNT	ZPL2 font E8
F.FNT	ZPL2 font F
G.FNT	ZPL2 font G
H8.FNT	ZPL2 font H8
GS.FNT	ZPL2 font GS
K1.SFT	16×16 ドット漢字
K2.SFT	24×24 ドット漢字
K3.SFT	22×22 ドット漢字
K4.SFT	32×32 ドット漢字
K8.SFT	16×16 ドット半角・全角漢字
K9.SFT	24×24 ドット半角・全角漢字
KA.SFT	22×22 ドット半角・全角漢字
KB.SFT	32×32 ドット半角・全角漢字
KD.SFT	40×40 ドット半角・全角漢字 (RJ-2035B、RJ-3035B を除く)
ダウンロードフォント	
ダウンロードフォントも指定することができます。ダウンロードフォントの転送方法は、付録 B: 日本語テキストを印刷するをご参照ください。	

rotation	0 ：回転なし 90 ：90 度時計回り 180：180 度時計回り 270：270 度時計回り
x-multiplication	水平方向の乗算（推奨範囲：1~10） • フォント"ROMAN.TTF"では、このパラメーターは無効です。 • フォント"0"やダウンロードした True Type フォントでは、幅（ポイント）を指定します（1 ポイント=1/72 インチ）。 • *.TTF フォントのみ、浮動小数点数に対応しています。
y-multiplication	垂直方向の乗算（推奨範囲：1~10） • フォント"0"やダウンロードした True Type フォントでは、高さ（ポイント）を指定します（1 ポイント=1/72 インチ）。 • *.TTF フォントのみ、浮動小数点数に対応しています。
alignment	オプション：テキストの位置合わせ 0: 初期値（左揃え） 1: 左揃え 2: 中央揃え 3: 右揃え
content	テキスト文字列の内容

備考：

- テキスト内に二重引用符 (") がある場合は、\["]に変更します。
  例："Brother \["]At your side\["]"

- 一部の文字が印刷されなかったり、文字化けしたりする場合は、以下をお試し下さい。

  プリンターの **CODEPAGE** 設定が印刷データの文字コードと一致しているか確認してください。必要な場合は、どちらかの文字コードを変更してください。

  プリンターの **CODEPAGE** 設定が、フォントの対応文字コードでない可能性があります。プリンターの **CODEPAGE** 設定を変更するか、文章で使用するフォントを変更してください。

- フォント「K1.SFT /K2.SFT /K3.SFT /K4.SFT /K8.SFT /K9.SFT /KA.SFT /KB.SFT」は、シフト JIS に対応しています。使用する場合は、**CODEPAGE** コマンドで「932（Shift-JIS）」を指定ください。

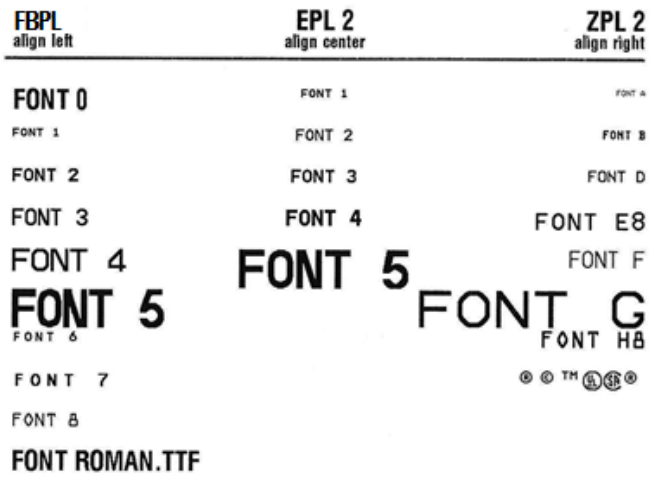
例

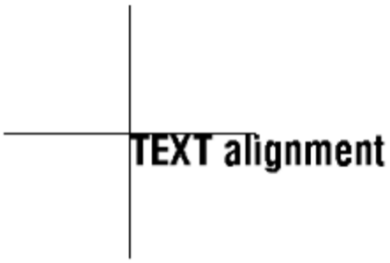
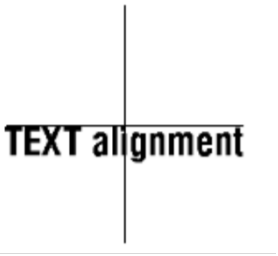
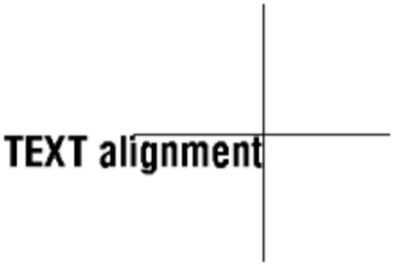
次のページに続きます。

サンプルコード

```
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10,"0",0,12,12,"FBPL"
TEXT 10,40,"0",0,8,8,"align left"
BAR 0,70,800,4
TEXT 10,110,"0",0,12,12,"FONT 0"
TEXT 10,160,"1",0,1,1,"FONT 1"
TEXT 10,210,"2",0,1,1,"FONT 2"
TEXT 10,260,"3",0,1,1,0,"FONT 3"
TEXT 10,310,"4",0,1,1,0,"FONT 4"
TEXT 10,360,"5",0,1,1,0,"FONT 5"
TEXT 10,410,"6",0,1,1,1,"FONT 6"
TEXT 10,460,"7",0,1,1,1,"FONT 7"
TEXT 10,510,"8",0,1,1,1,"FONT 8"
TEXT 10,560,"ROMAN.TTF",0,12,12,"FONT ROMAN.TTF"

TEXT 400,10,"0",0,12,12,2,"EPL 2"
TEXT 400,40,"0",0,8,8,2,"align center"
TEXT 400,110,"1.EFT",0,1,1,2,"FONT 1"
TEXT 400,160,"2.EFT",0,1,1,2,"FONT 2"
TEXT 400,210,"3.EFT",0,1,1,2,"FONT 3"
TEXT 400,260,"4.EFT",0,1,1,2,"FONT 4"
TEXT 400,310,"5.EFT",0,1,1,2,"FONT 5"

TEXT 800,10,"0",0,12,12,3,"ZPL 2"
TEXT 800,40,"0",0,8,8,3,"align right"
TEXT 800,110,"A.FNT",0,1,1,3,"FONT A"
TEXT 800,160,"B.FNT",0,1,1,3,"FONT B"
TEXT 800,210,"D.FNT",0,1,1,3,"FONT D"
TEXT 800,260,"E8.FNT",0,1,1,3,"FONT E8"
TEXT 800,310,"F.FNT",0,1,1,3,"FONT F"
TEXT 800,360,"G.FNT",0,1,1,3,"FONT G"
TEXT 800,410,"H8.FNT",0,1,1,3,"FONT H8"
TEXT 800,460,"GS.FNT",0,1,1,3,"ABCDEF"
PRINT 1
```

結果



サンプルコード	結果
SIZE 4,2 GAP 0,0 DIRECTION 1 CLS BAR 60,120,200,1 BAR 160,20,1,200 TEXT 160,120,"0",0,12,12,1,"TEXT alignment" PRINT 1,1	
サンプルコード	結果
SIZE 4,2 GAP 0,0 DIRECTION 1 CLS BAR 60,120,200,1 BAR 160,20,1,200 TEXT 160,120,"0",0,12,12,2,"TEXT alignment" PRINT 1,1	
サンプルコード	結果
SIZE 4,2 GAP 0,0 DIRECTION 1 CLS BAR 160,120,200,1 BAR 260,20,1,200 TEXT 260,120,"0",0,12,12,3,"TEXT alignment" PRINT 1,1	

6.22 BLOCK

説明

文章の段落を定義します。

構文

```
BLOCK x,y,width,height,"font",rotation,x-multiplication,y-multiplication,
[space,]align,]fit,]"content"
```

パラメーター	説明
x	開始位置の x 座標（ドット）
y	開始位置の y 座標（ドット）
width	段落のブロックの幅（ドット）
height	段落のブロックの高さ（ドット）
font	フォント名

内蔵フォント	
0	CG Triumvirate Bold Condense (Monotype Imaging®) フォント幅と高さは伸縮可能です
1	8×12 ドット固定ピッチフォント
2	12×20 ドット固定ピッチフォント
3	16×24 ドット固定ピッチフォント
4	24×32 ドット固定ピッチフォント
5	32×48 ドット固定ピッチフォント
6	14×19 ドット固定ピッチフォント OCR-B
7	21×27 ドット固定ピッチフォント OCR-B
8	14×25 ドット固定ピッチフォント OCR-A
ROMAN.TTF	CG Triumvirate Bold Condensed (Monotype Imaging®) フォント幅と高さの比率は固定です
1.EFT	EPL2 font 1
2.EFT	EPL2 font 2
3.EFT	EPL2 font 3
4.EFT	EPL2 font 4
5.EFT	EPL2 font 5
A.FNT	ZPL2 font A
B.FNT	ZPL2 font B
D.FNT	ZPL2 font D
E8.FNT	ZPL2 font E8
F.FNT	ZPL2 font F
G.FNT	ZPL2 font G
H8.FNT	ZPL2 font H8
GS.FNT	ZPL2 font GS
K1.SFT	16×16 ドット漢字
K2.SFT	24×24 ドット漢字
K3.SFT	22×22 ドット漢字
K4.SFT	32×32 ドット漢字
K8.SFT	16×16 ドット半角・全角漢字
K9.SFT	24×24 ドット半角・全角漢字
KA.SFT	22×22 ドット半角・全角漢字
KB.SFT	32×32 ドット半角・全角漢字
KD.SFT	40×40 ドット半角・全角漢字 (RJ-2035B、RJ-3035B を除く)
ダウンロードフォント	
ダウンロードフォントも指定することができます。ダウンロードフォント	

の転送方法は、付録 B: 日本語テキストを印刷するをご参照ください。

rotation	0 ：回転なし 90 ：90 度時計回り 180：180 度時計回り 270：270 度時計回り
x-multiplication	水平方向の乗算（推奨範囲：1~10） • フォント"ROMAN.TTF"では、このパラメーターは無効です。 • フォント"0"やダウンロードした True Type フォントでは、幅（ポイント）を指定します（1 ポイント=1/72 インチ）。 • *.TTF フォントのみ、浮動小数点数に対応しています。
y-multiplication	垂直方向の乗算（推奨範囲：1~10） • フォント"0"やダウンロードした True Type フォントでは、高さ（ポイント）を指定します（1 ポイント=1/72 インチ）。 • *.TTF フォントのみ、浮動小数点数に対応しています。
[space]	行間のスペースの追加または削除（ドット）
[align]	テキストの位置合わせ 0: 初期値（左揃え） 1: 左揃え 2: 中央揃え 3: 右揃え
[fit]	ブロック内に収まるようテキストを縮小する 0：縮小なし（初期値） 1：縮小あり
content	ブロック内のデータ（最大データ長は 4092 バイト）

備考：

- テキスト内に二重引用符 (") がある場合は、\["]に変更します。
  例："Brother \["]At your side\["]"
- 一部の文字が印刷されなかったり、文字化けしたりする場合は、以下をお試し下さい。

  プリンターの **CODEPAGE** 設定が印刷データの文字コードと一致しているか確認してください。必要な場合は、どちらかの文字コードを変更してください。

  プリンターの **CODEPAGE** 設定が、フォントの対応文字コードでない可能性があります。プリンターの **CODEPAGE** 設定を変更するか、文章で使用するフォントを変更してください。
- フォント「K1.SFT /K2.SFT /K3.SFT /K4.SFT /K8.SFT /K9.SFT /KA.SFT /KB.SFT / KD.SFT」は、シフト JIS に対応しています。使用する場合は、**CODEPAGE** コマンドで「932（Shift-JIS)」を指定ください。
- KD.SFT フォントは、最新のファームウェアバージョンでお使いいただけます。

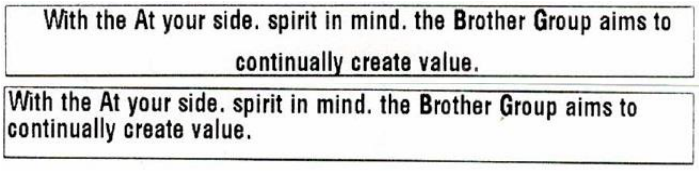
例

次のページに続きます。

サンプルコード

```
SIZE 4,0.5
GAP 0,0
DIRECTION 1
CLS
BOX 10,10,800,100,2
BLOCK 15,15,790,90, "0",0,8,8,"
With the At your side. spirit
in mind. the Brother Group aims
to continually create value."
PRINT 1
CLS
BOX 10,10,800,100,2
BLOCK
15,15,790,90,"0",0,8,8,20,2,"
With the At your side. spirit
in mind. the Brother Group aims
to continually create value."
PRINT 1
```

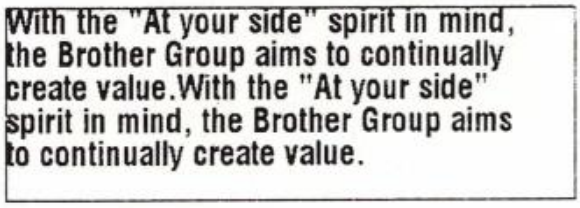
結果

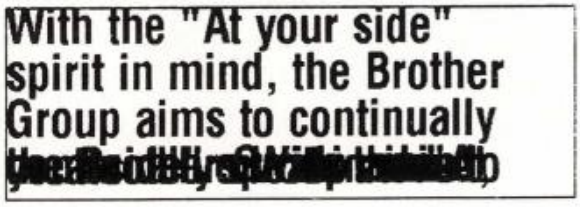


サンプルコード [fit] パラメーター

```
DATA$ = "With the \["]At your
side\["] spirit in mind, the
Brother Group aims to
continually create value.With
the \["]At your side\["] spirit
in mind, the Brother Group aims
to continually create value."
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
BLOCK
20,20,500,170,"0",0,10,10,0,0,1
,DATA$
BOX 20,20,500+20,170+20,2
PRINT 1

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
BLOCK
20,20,500,170,"0",0,10,10,0,0,0
,DATA$
BOX 20,20,500+20,170+20,2
PRINT 1
```

結果





6.23 FONT_REDIRECT

説明

印刷ジョブ内で指定されているフォントを特定のフォントに読み替えます。

構文

```
FONT_REDIRECT "current font","replacement font"
```

パラメーター	説明
Current font	印刷ジョブ内で指定されているフォント名
Replacement font	置き換えるフォント名

備考：

- 対応しているフォント形式は .SFT と .TTF です。
- ダウンロードフォントも指定することができます。ダウンロードフォントの転送方法は、付録 B: 日本語テキストを印刷するをご参照ください。

- 本機能は、最新のファームウェアバージョンでお使いいただけます。
- 本機能は、FBPL および SBPL のプログラムにのみご利用いただけます。
  印刷ジョブ送信前に本コマンドを送信して、本機能を有効にしてください。

例

サンプルコード

```
FONT_REDIRECT "K1.SFT","K4.SFT"
FONT_REDIRECT "K2.SFT","ARIAL.TTF"
```

7　ステータス取得と即時実行コマンド

7.1 <ESC>!?

説明

プリンター情報を取得します。最初の制御文字として、<ESC> （ASCII 27, escape 文字）を送信します。1 バイト文字が返信され、プリンター状態のフラグが設定されます。0 はプリンターがラベルを印刷できる状態であることを表しています。

構文

<ESC>!?

16 進数	プリンターステータス
00	通常
01	ヘッドオープン
02	用紙詰まり
03	用紙詰まりとヘッドオープン
04	用紙切れ
05	用紙切れとヘッドオープン
08	リボン切れ
09	リボン切れとヘッドオープン
0A	リボン切れと用紙詰まり
0B	リボン切れ、用紙詰まりとヘッドオープン
0C	リボン切れと用紙切れ
0D	リボン切れ、用紙切れとヘッドオープン
10	一時停止
20	印刷
80	その他のエラー

関連コマンド

<ESC>!S

7.2 <ESC>!C

説明

プリンターを再起動させ、AUTO.BAS の実行を省略します。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

```
<ESC>!C
```

備考：

AUTO.BAS がない場合もプリンターは再起動します。

関連コマンド

<ESC>!Q

7.3 <ESC>!D

説明

<ESC>!で開始される即時実行コマンド (例：<ESC>!R <RSC>!? <ESC>!C) を無効にします。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

<ESC>!D

関連コマンド

~!E

7.4 <ESC>!O

説明

プリンターの一時停止状態をキャンセルします。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

```
<ESC>!O
```

関連コマンド

<ESC>!P

7.5 <ESC>!P

説明

プリンターを一時停止させます。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

```
<ESC>!P
```

関連コマンド

<ESC>!O

7.6 <ESC>!Q

説明

プリンターを再起動させ、AUTO.BAS の実行を省略します。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

```
<ESC>!Q
```

備考：
プリンター内に AUTO.BAS が無い場合、プリンターは再起動しません。

関連コマンド

<ESC>!C

7.7 <ESC>!R

説明

プリンターを再起動します。このコマンドの最初は ESCAPE 文字（ASCII 27）です。ダウンロードされたメモリ内のファイルは消去されます。このコマンドはダンプモードでは送信することができません。

構文

```
<ESC>!R
```

関連コマンド

<ESC>!?

7.8 <ESC>!S

説明

プリンター状態を取得します。最初の制御文字として、<ESC> （ASCII 27, escape 文字）をプリンターへ送信します。8 バイト文字が返り、プリンター状態のフラグが設定されます。

構文

<ESC>!S

戻り値フォーマット

<STX>[4-byte status]<ETX><CR><LF>

ステータスバイト #1: メッセージ											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	16 進数	10 進数	ASCII	意味
0	1	0	0	0	0	0	0	40	64	@	通常
0	1	1	0	0	0	0	0	60	96	`	一時停止
0	1	0	0	0	0	1	0	42	66	B	ラベル逆送り
0	1	0	0	0	0	1	1	43	67	C	カット
0	1	0	0	0	1	0	1	45	69	E	プリンターエラー
0	1	0	0	0	1	1	0	46	70	F	ラベル頭出し
0	1	0	0	1	0	1	1	4B	75	K	印刷キー押下待ち
0	1	0	0	1	1	0	0	4C	76	L	ラベル取得待ち
0	1	0	1	0	0	0	0	50	80	P	バッチ印刷中
0	1	0	1	0	1	1	1	57	87	W	画像処理中

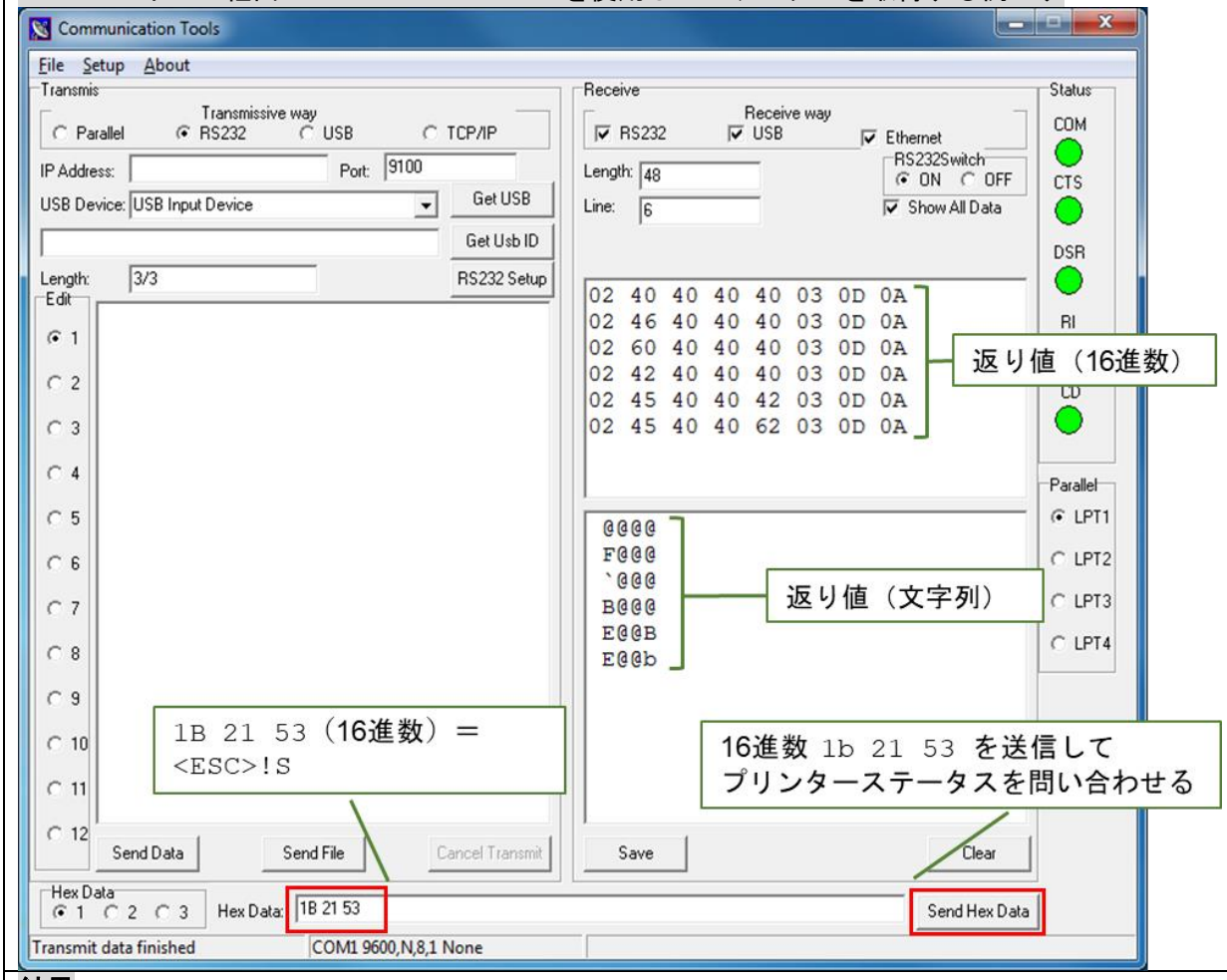
ステータスバイト #2: 警告											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	16 進数	10 進数	ASCII	意味
0	1	0	0	0	0	0	0	40	64	@	通常
0	1	0	0	0	0	0	1	41	65	A	用紙少量
0	1	0	0	0	0	1	0	42	66	B	リボン少量
0	1	0	0	0	1	0	0	44	68	D	（未使用）
0	1	0	0	1	0	0	0	48	72	H	受信バッファフル (RS-232)
0	1	1	0	0	0	0	0	60	96	`	（未使用）

ステータスバイト #3: エラー											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	16 進数	10 進数	ASCII	意味
0	1	0	0	0	0	0	0	40	64	@	通常
0	1	0	0	0	0	0	1	41	65	A	印字ヘッド高温
0	1	0	0	0	0	1	0	42	66	B	ステッピングモーター高温
0	1	0	0	0	1	0	0	44	68	D	印字ヘッドエラー
0	1	0	0	1	0	0	0	48	72	H	カッターの目詰まり
0	1	0	1	0	0	0	0	50	80	P	メモリ不十分

ステータスバイト #4: エラー											
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	16 進数	10 進数	ASCII	意味
0	1	0	0	0	0	0	0	40	64	@	通常
0	1	0	0	0	0	0	1	41	65	A	用紙切れ
0	1	0	0	0	0	1	0	42	66	B	用紙詰まり
0	1	0	0	0	1	0	0	44	68	D	リボン切れ
0	1	0	0	1	0	0	0	48	72	H	リボン詰まり
0	1	1	0	0	0	0	0	60	96	`	印字ヘッドオープン

例

RS-232C ポート経由で Brother CommTool を使用してステータスを取得する例です



結果

1: 開始、2: 4 バイトステータス（16 進数）、 3: 終了、4: 4 バイトステータス（文字列）

ステータス例	1		2								3						4			
通常	0	2	4	0	4	0	4	0	4	0	0	3	0	D	0	A	@	@	@	@
ラベル送り中	0	2	4	6	4	0	4	0	4	0	0	3	0	D	0	A	F	@	@	@
一時停止中	0	2	6	0	4	0	4	0	4	0	0	3	0	D	0	A	`	@	@	@
ラベル逆送り中	0	2	4	2	4	0	4	0	4	0	0	3	0	D	0	A	B	@	@	@
エラー：用紙詰まり	0	2	4	5	4	0	4	0	4	2	0	3	0	D	0	A	E	@	@	B
エラー：用紙詰まりとヘッドオープン	0	2	4	5	4	0	4	0	6	2	0	3	0	D	0	A	E	@	@	b

備考：

用紙詰まり<Hex 42>

ヘッドオープン<Hex 60>

0x42 | 0x60 = 62 <Hex b>

関連コマンド

<ESC>!?

7.9 <ESC>!F

説明

ラベルを送ります。**給紙**ボタンを押すことと同義です。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

```
<ESC>!F
```

7.10<ESC>!.

説明

全ての印刷ファイルをキャンセルします。このコマンドの最初は ESCAPE 文字（ASCII 27）です。

構文

```
<ESC>!.
```

7.11~!@

説明

プリンターの総印刷距離を問い合わせます。総印刷距離の整数部は ASCII 文字で返ります（小数部は返りません）。総印刷距離の最後の文字は `0x0D` です。

構文

```
~!@
```

例

```
~!@
```

7.12~!A

説明

プリンターの空き容量を問い合わせます。空き容量のバイト数は 10 進数で表され、末尾に `0x0d` がつきます。

構文

```
~!A
```

例

```
~!A
```

関連コマンド

FILES

7.14~!D

説明

プリンタ—はダンプモードになります。ダンプモードでは、プリンターは受信したコードをそのまま出力します。

構文

```
~!D
```

例

```
~!D
```

7.15~!E

説明

<ESC>!で始まる即時実行コマンド（例： <ESC>!R <RSC>!? <ESC>!C）を有効にします。

構文

```
~!E
```

例

```
~!E
```

関連コマンド

<ESC>!D

7.16~!F

説明

プリンターメモリ内にある全てのファイルとフォントを問い合わせます。ファイル名は ASCII 文字として返信されます。各ファイル名は `0x0D` で終わります。最後は `0x1A` で終わります。このコマンドを複数回入力するとメモリ内にあるファイルを繰り返し読み出します。

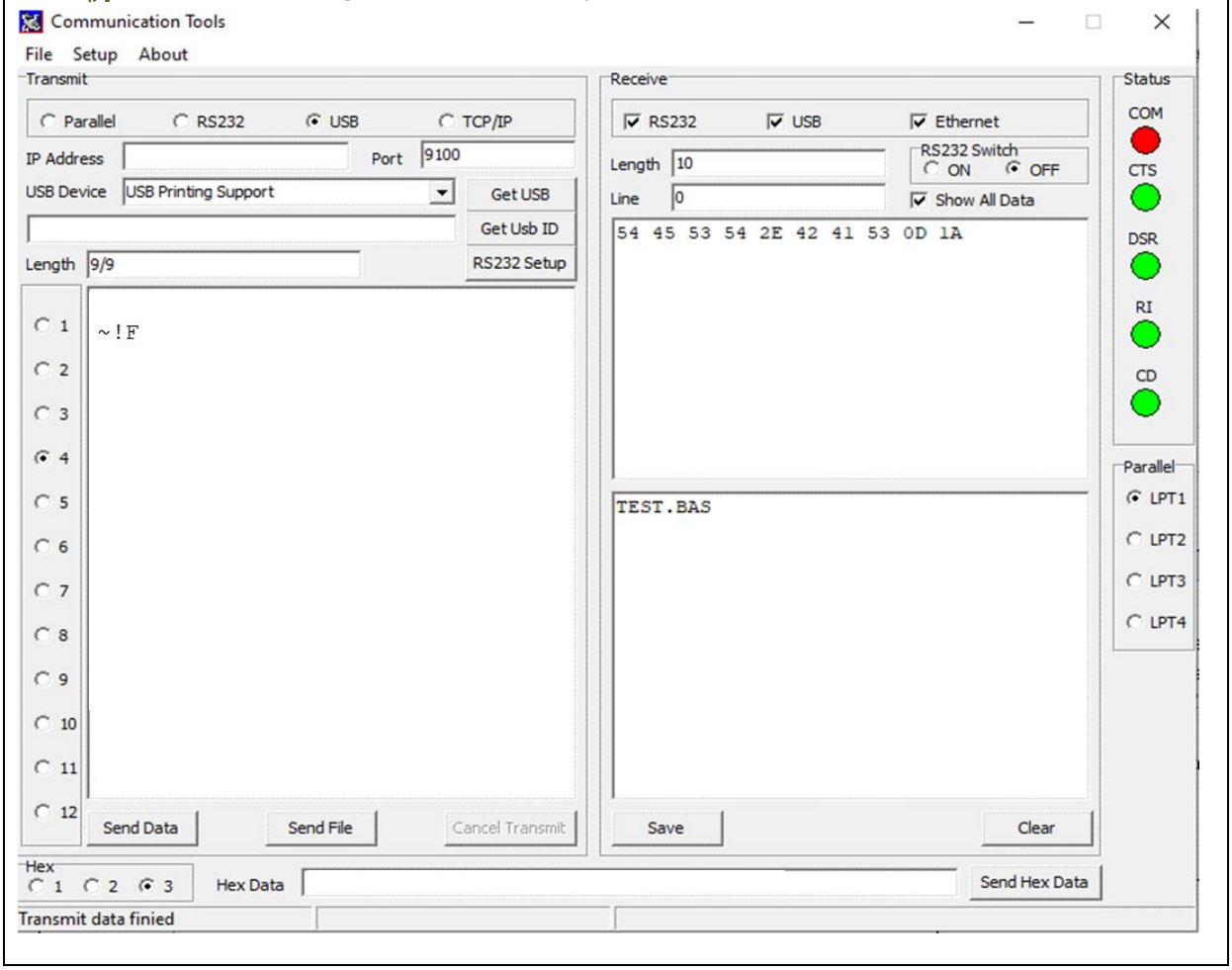
構文

```
~!F
```

例

サンプルコード

This 例 uses the Brother CommTool to test ~!F.



関連コマンド

FILES

7.17~!I

説明

プリンターのコードページと国コードの設定を問い合わせます。

構文

```
~!I
```

返り値のフォーマット：
コードページ, 国コード
例　8 bit: `437, 001`
　　7 bit: `USA, 001`

コードページの詳細は **CODEPAGE** コマンドをご参照ください。

例

```
~!I
```

関連コマンド

CODEPAGE

7.18~!T

説明

プリンターのモデル名とモデル番号を問い合わせます。この情報は ASCII 文字で返信されます。

構文

```
~ !T
```

例

```
~ !T
```

7.19<ESC>Y

説明

ラインモードに入ります（EZP (ESC/POS) ファームウェアのみ）

備考：

- EZP ファームウェアアップデートキットは、RJ-2035B/RJ-3035B のダウンロードページ (support.brother.com) からダウンロードできます。

構文

```
<ESC>Y
```

例

```
<ESC>Y
```

関連コマンド

```
<ESC>Z
```

7.20<ESC>Z

説明

ラインモードからページモードに戻ります（EZP (ESC/POS) ファームウェアのみ）

備考：

- EZP ファームウェアアップデートキットは、RJ-2035B/RJ-3035B のダウンロードページ (support.brother.com) からダウンロードできます。

構文

```
<ESC>Z
```

例

```
<ESC>Z
```

関連コマンド

```
<ESC>Y
```

8 Windows ドライバー用コマンド

8.1 !B

説明

メモリ内にビットマップイメージデータを保存します。後ろの nnn はビットマップデータになります。

構文

```
!Bnnn
```

パラメーター	説明
nnn	パソコンからプリンターに送られるイメージデータのバイト数は 3 桁の 10 進数で表されます。

例

```
!B100
```

関連コマンド

BITMAP

8.2 !J

説明

y 方向で特定の位置にビットマップデータを印刷します。

構文

```
!Jnnnn
```

パラメーター	説明
nnn	y 方向で特定の位置にイメージを印刷します。この位置は 4 桁の 10 進数で表されます。

例

```
!J0100
```

関連コマンド

FEED

8.3 !N

説明

指定枚数分のラベルを印刷します。

構文

```
!Nnnn
```

パラメーター	説明
nnn	印刷するコピー枚数を指定します

例

```
!N001
```

9 ファイル管理コマンド

9.1 DOWNLOAD

説明

プリンターのメモリ内にプログラムファイルやデータファイル（テキストデータファイル、グラフィックファイル、ビットマップフォントファイルを含む）を保存します。

保存できるファイル数
DRAM　: 50 ファイル
FLASH　: 256 ファイル

構文

- **データファイルを転送する場合：**
  `DOWNLOAD [n,] "FILENAME",DATA SIZE,DATA CONTENT`

パラメーター	説明
n	データファイルを保存するメモリを指定します。

ID	メモリ
指定なし	DRAM 電源を切る前に DRAM から FLASH メモリにファイルを保存したい場合は、**MOVE** コマンドを使用します。
F	FLASH
U	USB
E	SD CARD

FILENAME	プリンターメモリ内に保存するファイル名
DATA SIZE	ヘッダーを含まないデータファイルの実際のサイズ（バイト）
DATA CONTENT	プリンター内にダウンロードされるデータ

備考：
- テキストファイルでは、改行コード CR LF（0x0D 0x0A）がデータの区切りです。
- ファイル名の大文字と小文字は区別されます。
- DRAM 内のファイルは、プリンター—の電源オフ時に失われます。

備考：
BPM (Brother Printer Management Tool) を使用してプリンターにデータを転送することもできます。

1. パソコンとプリンターを接続し、プリンターの電源を入れます。
2. BPM を起動し、「**ファイルマネージャー**」をクリックします。
3. 「**ファイル形式**」には転送するファイルの種類を選択します。
4. 「**ブラウズ**」をクリックしてファイルを選択し、「**開く**」をクリックします。ファイル名と拡張子は、必ず**大文字（半角英数字）**にしてください。（例：FILENAME.BMP、FILENAME.PCX）
5. 「**メモリー機器**」をクリックして、ファイルをダウンロードする場所を選択し、「**ダウンロード**」ボタンをクリックして、ダウンロードを開始します。

- **プログラムファイル・テンプレートを転送する場合：**
  BAS ファイル（Printer BASIC File）として、本機内部へ転送します。
  `DOWNLOAD [n,] "FILENAME.BAS"`

パラメーター	説明
n	プログラムファイルを保存するメモリを指定します。

ID	メモリ
指定なし	DRAM 電源を切る前に DRAM から FLASH メモリにファイルを保存したい場合は、**MOVE** コマンドを使用します。
F	FLASH
U	USB
E	SD CARD

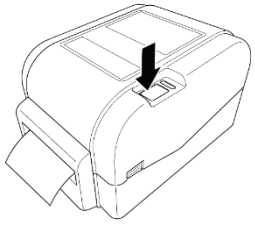
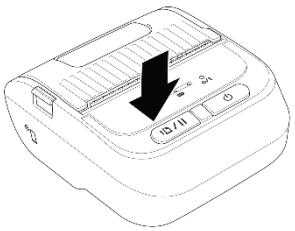
FILENAME.BAS　プリンターメモリ内に保存するファイル名

備考：

- ファイル名の大文字と小文字は区別されます。
- ファイルの拡張子を”.BAS”にします（例：DOWNLOAD F,"TEST.BAS や DOWNLOAD F,"DEMO.BAS"）。
- **EOP** コマンドと使用します。
- DRAM 内のファイルは、プリンター—の電源オフ時に失われます。
- .BAS 拡張子を除くファイル名（例：TEST や DEMO）、もしくは **RUN** コマンドを使用してダウンロードプログラムを実行します（例：RUN “TEST.BAS” や RUN “DEMO.BAS” ）。
- AUTO.BAS の優先度：DRAM > CARD (Ext. FLASH) > FLASH

備考：
プリンター起動時に自動的にプログラムを実行したい場合は､BAS ファイルを"AUTO.BAS"として保存します。”AUTO.BAS”ファイルを自動実行させたくない場合は、メモリ内から”AUTO.BAS”ファイルを削除するか、次の手順で”AUTO.BAS”ファイルの実行をスキップできます。

1. プリンター—の電源をオフにします。
2. 次のボタンを押しながらプリンターの電源を入れます。

TD-4420TN/4520TN	RJ シリーズ	TD-4650TNWB/4750TNWB TJ シリーズ
給紙/一時停止ボタン 	給紙/一時停止ボタン 	右側の選択ボタン

3. LED が緑色に点灯したら、ボタンを離します。

LED の色は次のように変化します。

TD-4420TN/4520TN
黄色（点灯）→ 赤色（5 回）→ 黄色（5 回）→ 緑色（5 回）→ 緑色/黄色（5 回）→ 赤色/黄色（5 回） → **緑色（点灯）**

TD-4650TNWB/4750TNWB/4750TNWBR
黄色（点灯）→ 赤色/黄色 （5 回） → 黄色 （5 回） → 緑色/赤色（5 回） →緑色（5 回） → 赤色 （5 回） → **緑色（点灯）**

RJ-2035B/3035B
黄色（点灯）→ 右/緑色 （5 回）→ 中央/緑色（5 回）→ 左/緑色（5 回）→ **緑色（点灯）**

TJ-4021TN/4121TN/4121TNR/4420TN/4520TN/4620TN
黄色（点灯）→ 赤色（5 回）→ 黄色（5 回）→ 緑色（5 回）→ 緑色/黄色（5 回）→ 赤色/黄色（5 回） → **緑色（点灯）**

例

サンプルコード プリンターの DRAM メモリにダウンロードする方法

```
DOWNLOAD "Example.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
TEXT 100,100, "3",0,1,1, "Example PROGRAM"
PRINT 1
EOP
```

サンプルコード プリンターの FLASH メモリにダウンロードする方法

```
DOWNLOAD F, "Example.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
TEXT 100,100, "3",0,1,1, "Example PROGRAM"
PRINT 1
EOP
```

AUTO.BAS ファイルを使用して印刷までの時間を早くする（RJ-2035B, RJ-3035B のみ）

RJ-2035B, RJ-3035B でダウンロードしたフォントを使用する場合、プリンター起動後 最初の印刷ではフォントを作業領域へ移動しているため、印刷を開始するまでに時間がかかることがあります。

次の AUTO.BAS ファイルを事前に転送して、プリンター起動時に自動的にフォントを印刷できる領域に移動させることで、改善する場合があります。

サンプルコード

```
DOWNLOAD F,"AUTO.BAS"
COPY F,"IPAEXG.TTF","IPAEXG.TTF"
EOP
AUTO
```

関連コマンド

EOP, RUN, PUTBMP, PUTPCX, INPUT, FILES, ~!F

9.2 EOP

説明

プログラムの最後を表します。プログラムの開始と終了を宣言するために、プログラムの最初の行に **DOWNLOAD** “FILENAME.BAS”を追加し、プログラムの最後の行に“**EOP**”を追加します。

構文

```
EOP
```

例

サンプルコード プリンターの DRAM メモリにダウンロードする方法

```
DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
TEXT 100,100, "3",0,1,1, "DEMO PROGRAM"
PRINT 1
EOP
```

関連コマンド

DOWNLOAD, INPUT, FILES, ~!F

9.3 FILES

説明

プリンターメモリ内（FLASH と DRAM の両方）にある合計メモリサイズ、利用可能なメモリサイズとファイルリスト（もしくは RS-232C 経由のファイルリスト）を印刷します。

構文

```
FILES
```

例

サンプルコード	結果

`FILES`	`----------`
	`DRAM FILE (0 FILES)`
	`----------`
	`PHYSICAL 8192 KBYTES`
	`AVAILABLE 256 KBYTES`
	`----------`
	
	`----------`
	`FLASH FILE (0 FILES)`
	`----------`
	`PHYSICAL 4096 KBYTES`
	`AVAILABLE 2560 KBYTES`
	`----------`

関連コマンド

~!F, KILL

9.4 KILL

説明

プリンターメモリ内のファイルを消去します。

構文

```
KILL [n], "FILENAME"
```

パラメーター	説明
n	消去するファイルがあるメモリ

ID	メモリ
指定なし	DRAM
F	FLASH
U	USB
E	SD CARD

FILENAME	消去するデータファイル名（大小文字区別）

備考：

- パラメーターn が指定されていない場合、DRAM 内のファイルを消去します
- 構文 例

```
KILL "FILENAME"        : DRAM 内の特定ファイルを消去します
KILL "*.PCX"           : DRAM 内のすべての PCX ファイルを消去します
KILL "* "              : DRAM 内のすべてのファイルを消去します
KILL F, "FILENAME"     : FLASH 内の特定のファイルを消去します
KILL E, " *.PCX "      : 拡張メモリカード内のすべての PCS ファイルを消去します
```

- **KILL** コマンドを送る前に、**MOVE** コマンドを使用します。

モデル	対応		
	KILL "*"	KILL "*" MOVE	KILL F, "*"
FBPL 言語	**V**		**V**

例

SELFTEST コマンドまたは FILES コマンドをプリンターに送信することで、プリンターのメモリ内に保存されたファイルの一覧を印刷できます。パラレルポート接続を通じて、プリンターメモリ内のファイルを消去するためには次の手順を行います。

```
C:\>COPY CON LPT1<ENTER>
FILES<ENTER>
<CTRL><Z><ENTER>
C:\>COPY CON LPT1<ENTER>
KILL "DEMO.BAS "  <ENTER>
<CTRL><Z><ENTER>
C:\>COPY CON LPT1<ENTER>
FILES<ENTER>
<CTRL><Z><ENTER>
```

備考： `<ENTER>` は PC キーボードの **ENTER** キーです。`<CTRL><Z>` は PC キーボードで **CTRL** キーを押した後、**Z** キーを押すことを意味しています。

関連コマンド

~!F, FILES

9.5 MOVE

説明

ダウンロードされているファイルを DRAM から FLASH メモリに移動させます。

構文

```
MOVE
```

関連コマンド

DOWNLOAD, EOP

9.6 RUN

説明

プリンターのメモリ内のプログラムを実行します。

構文

```
RUN "FILENAME.BAS"
```

例

<table>
<tr><th>サンプルコード</th><th>結果</th></tr>
<tr><td>

```
DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
TEXT   100,100,   "3",0,1,1,   "DEMO
PROGRAM"
PRINT 1
EOP
RUN "DEMO.BAS"
```

</td><td rowspan="2">

DEMO PROGRAM

</td></tr>
<tr><td>

```
DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
TEXT   100,100,   "3",0,1,1,   "DEMO
PROGRAM"
PRINT 1
EOP
DEMO
```

備考：
このコマンドは、".BAS"を含めないファイル名に置き換えることができます（例：RUN “DEMO.BAS” = DEMO）

</td></tr>
</table>

関連コマンド

DOWNLOAD, EOP

10 基本コマンドと制御コマンド

10.1ABS()

説明

整数、浮動小数点や変数の絶対値を返します。

構文

```
ABS (VARIABLE)
```

例

サンプルコード	結果

`DOWNLOAD "TEST.BAS"`
`SIZE 4,4`
`GAP 0,0`
`DIRECTION 1`
`SET TEAR ON`
`CLS`
`A=ABS(-100)`
`B=ABS(-50.98)`
`C=-99.99`
`TEXT 100,100, "3",0,1,1,STR$(A)`
`TEXT 100,150, "3",0,1,1,STR$(B)`
`TEXT 100,200, "3",0,1,1,STR$(ABSI)`
`PRINT 1`
`EOP`
`RUN "TEST.BAS"`
	100
	50.98
	99.99

関連コマンド

DOWNLOAD, EOP

10.2 ASC()

説明

文字の ASCII コードを返します。

構文

```
ASC (" A ")
```

例

サンプルコード	結果

`DOWNLOAD "TEST.BAS"`
`SIZE 4,4`
`GAP 0,0`
`DIRECTION 1`
`SET TEAR ON`
`CLS`
`CODE1=ASC(" A ")`
`TEXT`
`100,100, " 3 ",0,1,1,STR$(CODE1)`
`PRINT 1`
`EOP`
`RUN "TEST.BAS"`
	65

関連コマンド

DOWNLOAD, EOP, STR$()

10.3 CHR$()

説明

指定された ASCII コードを使い、文字を返します。

構文

```
CHR$(n)
```

パラメーター	説明
n	ASCII コード

例

サンプルコード

```
DOWNLOAD "TEST.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
SET TEAR ON
CLS
A=75
WORD$=CHR$(A)
TEXT 100,100, "3",0,1,1,WORD$
PRINT 1
EOP
RUN "TEST.BAS"
```

結果

K

関連コマンド

DOWNLOAD, EOP, STR$(), ASC$()

10.4XOR$()

説明

オリジナルデータをロジック XOR により新しいデータにコード化します。

構文

```
XOR$(data$,password$)
```

パラメーター	説明
data$	オリジナルデータは Password$でコード化する必要があります
Password$	新しいデータを作成するために使用します

例

サンプルコード

```
data$="1234"
password$="ABCD"
encoded$=XOR$(data$,password$)
deconded$=XOR$(encoded$,password$)

SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1, "Encoded data:
"+encoded$
TEXT 10,60, "3",0,1,1, "Decoded data:
"+deconded$
PRINT 1
```

結果

Encoded data: pppp

Decoded data: 1234

10.5END

説明

プログラムの終わりを宣言します。

構文

```
END
```

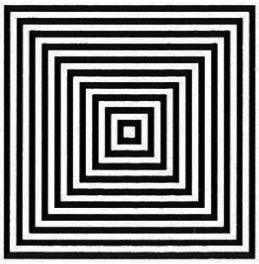
例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 200,60, "4",0,1,1, "END COMMAND
TEST"
X=300
Y=200
X1=500
Y1=400
GOSUB DR_LINE
PRINT 1
END

:DR_LINE
FOR I=1 TO 100 STEP 10
BOX X+I,Y+I,X1-I,Y1-I,5
NEXT
RETURN
EOP
DEMO
```

結果

END COMMAND TEST



関連コマンド

DOWNLOAD, EOP, GOSUB

10.6 EOF()

説明

オープンしているダウンロードファイルが終わりに到達したかどうかを確認するために使用します。

構文

```
EOF(File Handle)
```

パラメーター	説明
File handle	0 または 1

返り値	説明
None-zero	ファイルの終わり
0	ファイルの終わりでない

例

サンプルコード

```
DOWNLOAD "DATA",16,COMPUTER
2000

DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.0,0
DIRECTION 1
CLS
OPEN "DATA",0
SEEK 0,0
Y=110
TEXT 10,10, "3",0,1,1, "*****EOF TEST*****"
:A
Temp$=""
READ 0,ITEM$,P
TEXT 10,Y,"2",0,1,1,ITEM$+"$" +STR$(P)+"[EOF(0)="+STR$(EOF(0))+"]"
BARCODE 10,Y+25,"39",40,1,0,2,4,"PRICE-"+STR$(P)
Y=Y+100
IF EOF(0)=0 THEN GOTO A
PRINT 1
EOP
DEMO
```

結果

関連コマンド

DOWNLOAD, EOP, OPEN, READ, SEEK

10.7 OPEN

説明

ダウンロードされたデータファイルを開き、ファイルハンドル（システムがファイルを識別する情報）を確立します。最大 2 つのファイルまで同時に開くことができます。開くファイルはこのコマンドを使う前にダウンロードされている必要があります。

備考：
ファイルを開くとき、ファームウェアは自動的に基板 FLASH メモリや拡張メモリカードにファイルが存在するかどうかを確認するために検索します。もしこのファイルが存在しない時は、プリンターはこのファイルを基板 FLASH メモリ内に作成します。

構文

```
OPEN [memory ID,] "filename",file handle
```

パラメーター	説明
[memory ID]	オプション：指定メモリ内のファイルを開きます

ID	メモリ
指定なし	DRAM
F	FLASH
E	SD CARD
U	USB

filename	メモリ内のファイル名
file handle	0 または 1

例

サンプルコード

```
DOWNLOAD "DATA.DAT",18,Open file in
DRAM.
DOWNLOAD F, "DATA.DAT",19,Open file
in FLASH.
DOWNLOAD "TEST.BAS"
data1$=""
data2$=""
data3$=""
OPEN "DATA.DAT",0
READ 0,data1$
CLOSE 0
OPEN F, "DATA.DAT",0
READ 0,data2$
CLOSE 0
KILL F, "*"
OPEN "NEW.DAT",0
SEEK 0,0
WRITE 0, "Auto create a new file in
FLASH."
SEEK 0,0
READ 0,data3$
CLOSE 0
SIZE 4,1
GAP 0,0
CLS
```

結果

```
Open file in DRAM.
Open file in FLASH.
Auto create a new file in FLASH.
```

`TEXT 10,10,"3",0,1,1,data1$`
`TEXT 10,60,"3",0,1,1,data2$`
`TEXT 10,110,"3",0,1,1,data3$`
`PRINT 1`
`EOP`
`TEST`	

関連コマンド

DOWNLOAD, EOP, READ, WRITE, SEEK, CLOSE

10.8 CLOSE

説明

OPEN コマンドによりオープンとなっているファイルハンドルを閉じます。

構文

```
CLOSE file handle
```

パラメーター	説明
file handle	0 または 1

例

OPEN コマンドの例をご参照ください。

10.9WRITE

説明

ダウンロードされたデータファイルにデータを書き込みます。プリンターは 2 つのファイルハンドルに対応しているため、2 つのファイルを同時に開くことができます。

構文

```
WRITE file handle,variables
```

パラメーター	説明
file handle	0 または 1
variables	文字列、整数、浮動小数点変数

関連コマンド

READ, DOWNLOAD, EOP, OPEN, EOF, LOF, SEEK, FREAD$()

10.10　READ

説明

ダウンロードされたデータファイルからデータを読み取ります。

構文

```
READ file handle,variables
```

パラメーター	説明
file handle	0 または 1
variables	文字列、整数、浮動小数点変数

例

サンプルコード

```
DOWNLOAD "DATA1",20,COMPUTER
2000
12
DOWNLOAD "DATA2",16,Mouse
900
93
DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
DIRECTION 1
I=0
Y=100
OPEN "DATA1",0
OPEN "DATA2",1
SEEK 0,0
SEEK 1,0
:Start
CLS
TEXT      10,10,"3",0,1,1,"*****READ      COMMAND
TEST*****"
TEXT 10,50,"3",0,1,1,"OPEN-READ DATA"+STR$(I+1)
ITEM$=""
READ I,ITEM$,P,Q
TEXT 10,Y, "2",0,1,1,ITEM$+"$" +STR$(P)
BARCODE   10,Y+25,   "39   ",40,1,0,2,4,   "PRICE*
"+STR$(Q)+ "= "+STR$(P*Q)
Y=Y+100
PRINT 1
Y=100
IF I<=1 THEN
IF EOF(I)=1 THEN
I=I+1
GOTO Start
ELSE
GOTO Start
ENDIF
ELSE
END
ENDIF
EOP
DEMO
```

結果

関連コマンド

DOWNLOAD, EOP, OPEN, EOF, LOF, SEEK, FREAD$()

10.11 SEEK

説明

指定のファイルポインターを特定の位置に移します。

構文

```
SEEK file handle,offset
```

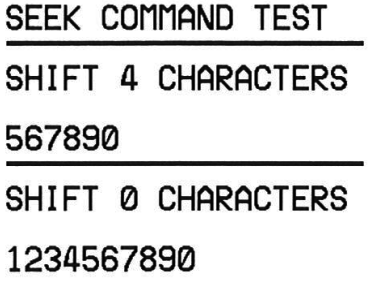
パラメーター	説明
file handle	0 または 1
offset	移動させるオフセット文字

例

サンプルコード

```
DOWNLOAD "DATA",12,1234567890
DOWNLOAD "TEST.BAS"
SIZE 4,1.5
GAP 0,0
DIRECTION 1
REFERENCE 0,0
CLS
OPEN "DATA",0
SEEK 0,4
READ 0,Num$
TEXT   100,10,"3",0,1,1,"SEEK   COMMAND
TEST"
BAR 100,40,300,4
TEXT     100,60,"3",0,1,1,"SHIFT     4
CHARACTERS"
TEXT 100,110,"3",0,1,1,Num$
BAR 100,140,300,4
SEEK 0,0
READ 0,Num$
TEXT     100,160,"3",0,1,1,"SHIFT     0
CHARACTERS"
TEXT 100,210,"3",0,1,1,Num$
PRINT 1
EOP
TEST
```

結果



関連コマンド

DOWNLOAD, EOP, OPEN, READ, EOF, LOF, FREAD$()

10.12　LOF()

説明

指定ファイルのサイズを返します。

構文

```
LOF("FILENAME")
```

パラメーター	説明
FILENAME	プリンターメモリ内のダウンロードファイル

例

サンプルコード

```
DOWNLOAD "DATA1",10,1234567890

DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO

DOWNLOAD "LofTest.BAS"
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
OPEN "DATA1",0
OPEN "DATA2",1
TEXT 10,20,"4",0,1,1,"LOF() FUNCTION
TEST"
J=LOF("DATA1")
K=LOF("DATA2")
TEXT 10,140,"3",0,1,1,"DATA1 IS:
"+STR$(J)+"Bytes"
TEXT 10,200,"3",0,1,1,"DATA2 IS:
"+STR$(K)+"Bytes"
PRINT 1
EOP
LofTest
```

結果

```
LOF() FUNCTION TEST

DATA1 IS: 10 Bytes

DATA2 IS: 15 Bytes
```

関連コマンド

DOWNLOAD, EOP, OPEN, READ, EOF, SEEK, FREAD$()

10.13 LOC()

説明

オープンファイル内で現在の読み/書き位置を返します。

構文

```
LOC(file handle)
```

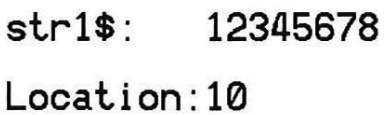
パラメーター	説明
file handle	0 または 1

例

サンプルコード

```
DOWNLOAD "DATA.DAT",30,12345678
12345678
12345678

DOWNLOAD "TEST.BAS"
str1$ = ""
location = 0
OPEN "DATA.DAT",0
READ 0,str1$
location = LOC(0)
CLOSE 0
SIZE 4,1
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1,"str1$:   "+str1$
TEXT
10,60,"3",0,1,1,"Location:"+STR$(location)
PRINT 1
EOP
TEST
```

結果



10.14 FREAD$()

説明

ファイルからバイトの指定数を読み取ります。

構文

```
FREAD$ (file handle,byte)
```

パラメーター	説明
file handle	0 または 1
byte	読み取るバイト数

例

サンプルコード

```
DOWNLOAD "DATA1",10,1234567890
DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO
DOWNLOAD "OPEN2.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
OPEN "DATA1",0
OPEN "DATA2",1
SEEK 0,0
SEEK 1,0
Y$=FREAD$(0,6)
Z$=FREAD$(1,6)
TEXT 10,100,"3",0,1,1,"FREAD$(0,6) IS: " +Y$
TEXT 10,150,"3",0,1,1,"FREAD$(1,6) IS: " +Z$
PRINT 1
EOP
OPEN2
```

結果

```
FREAD$(0,6) IS: 123456

FREAD$(1,6) IS: ABCDEF
```

関連コマンド

DOWNLOAD, EOP, OPEN, READ, EOF, LOF(), SEEK

10.15 PUT

説明

ファイルに 1 バイトを付加します。

構文

```
PUT file handle,var1$[, var2$][,var3$][, …]
PUT file handle,var1[, var2][,var3][, …]
PUT file handle,var1$[, var2$][,var3][, …]
```

パラメーター	説明
file handle	0 または 1
var$	文字データ
var	ASCII データ

例

サンプルコード

```
DOWNLOAD "DATA1",10,1234567890
DOWNLOAD "TEST.BAS"
str1$ = ""
str2$ = ""
OPEN "DATA1",0
SEEK 0,0
READ 0,str1$
PUT 0,"a","B",49
SEEK 0,0
READ 0,str2$
CLOSE 0

SIZE 4,0.5
GAP 0,0
CLS
TEXT 10, 10,"3",0,1,1,"Original data in DATA1: "+str1$
TEXT 10, 60,"3",0,1,1,"New data in Data1: "+str2$
PRINT 1
EOP
TEST
```

結果

```
Original data in DATA1: 1234567890
New data in Data1: 1234567890aB1
```

関連コマンド

DOWNLOAD, EOP, OPEN, READ, EOF, LOF(), SEEK, GET

10.16　GET

説明

ファイルから 1 バイトを取得します。

構文

```
GET file handle,var1$[,var2$][,var3$][, …]
GET file handle,var1[,var2][,var3][, …]
GET file handle,var1$[,var2$][,var3][, …]
```

パラメーター	説明
file handle	0 または 1
var$	文字の取得
var	ASCII 値の取得

例

サンプルコード

```
DOWNLOAD "DATA1",10,1234567890
DOWNLOAD "TEST.BAS"
a$=""
b$=""
c=0
d$=""
e$=""
OPEN "DATA1",0
SEEK 0,0
GET 0,a$,b$,c
SEEK 0,0
FOR I=1 TO 5
GET 0,d$
e$=e$+d$
NEXT

SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10,"3",0,1,1,"The first 3 characters in DATA1: "+ a$+b$+"
("+STR$(c)+")"
TEXT 10,60,"3",0,1,1,"The first 5 characters in DATA1: "+e$
PRINT 1
EOP
TEST
```

結果

```
The first 3 characters in DATA1: 12 (51)
The first 5 characters in DATA1: 12345
```

関連コマンド

DOWNLOAD, EOP, OPEN, READ, EOF, LOF(), SEEK, PUT

10.17　COPY

説明

ファイルを別のメモリにコピーします。

構文

```
COPY [memory ID of source,] "filename of source",[memory ID of new file,] "new filename"
```

パラメーター　**説明**

memory ID of source　オプション

ID	メモリ
指定なし	DRAM
F	FLASH
U	USB
E	SD CARD

source filename　現在のファイル名

Memory ID of new file　オプション

ID	メモリ
指定なし	DRAM
F	FLASH

new filename　コピー先のメモリで使用する新しいファイル名

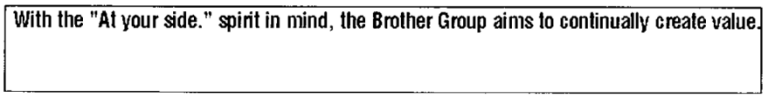
例

サンプルコード

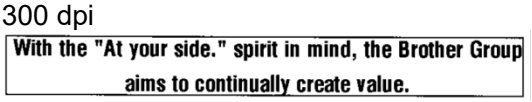
```
DOWNLOAD "DATA_D.DAT",105, With the "At your side." spirit in mind. the Brother Group aims to continually create value.
DOWNLOAD "TEST.BAS"
KILL F,"*"
COPY "DATA_D.DAT",F,"DATA_F.DAT"
OPEN "DATA_F.DAT",0
SEEK 0,0
data$=FREAD$(0,LOF("DATA_F.DAT"))
CLOSE 0
SIZE 4,0.5
GAP 0,0
CLS
BOX 10,10,800,100,2
BLOCK 15,15,790,90,"0",0,8,8,20,2,data$
PRINT 1
EOP
TEST
```

結果

203 dpi



300 dpi



関連コマンド

DOWNLOAD, EOP, OPEN, FREAD$(), EOF, LOF(), SEEK, CLOSE

10.18　FOR…NEXT LOOP

説明

1 つ以上のプログラムの行をそれぞれ実行するために使用します。ループカウンター値は実行の数を指定します。

構文

```
FOR variable = start TO end STEP increment
    statement; start < end
    [EXITFOR]
NEXT
```

パラメーター	説明
variable	変数名（8 文字まで）
start	整数か浮動小数点数
end	整数か浮動小数点数
increment	整数か浮動小数点、正数と負数
EXITFOR	ループを抜ける

備考：

・最大ネスト数は 39 個です。

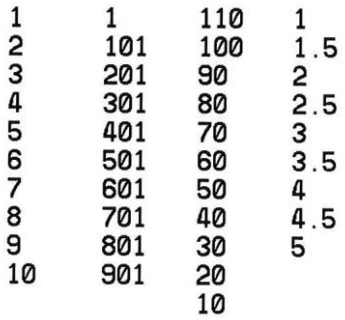
・**FOR…NEXT** ループ内に飛ぶことはできません。

例

サンプルコード

```
DOWNLOAD "TEST.BAS"
SIZE 4,2.5
GAP 0,0
CLS
FOR I=1 TO 10 STEP 1
TEXT 100,10+30*(I-1),"3",0,1,1,STR$(I)
NEXT

FOR I=1 TO 1000 STEP 100
TEXT                          200,10+((I-
1)/10)*3,"3",0,1,1,STR$(I)
NEXT

FOR I=110 TO 10 STEP -10
TEXT                        300,10+(ABS(I-
110))*3,"3",0,1,1,STR$(I)
NEXT

FOR I=1 TO 5 STEP 0.5
IF  I-INT(I)=0  THEN  Y=10+60*(I-1)  ELSE
Y=Y+30
TEXT 400,Y,"3",0,1,1,STR$(I)
NEXT

PRINT 1
EOP
TEST
```

結果



関連コマンド

DOWNLOAD, EOP

10.19 WHILE…WEND

説明

True 条件が与えられている間、指定されたプログラムを実行します。

構文

```
WHILE condition
[statement]
WEND
```

パラメーター	説明
condition	利用可能な関連演算子: <, >, =, <=, >=, <> *関連演算子 <>, 不等がサポートされました。
Statement	条件が True の間、1 つ以上のプログラム構成文が実行されます。

備考：

・最大ネスト数は 39 個です。

例

サンプルコード	結果

`DOWNLOAD "TEST.BAS"`
`I=0`
`TOTAL=0`
`WHILE I<100`
`I=I+1`
`TOTAL=TOTAL+I`
`WEND`
`SIZE 4,0.5`
`GAP 0,0`
`CLS`
`TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = "`
`+STR$(TOTAL)`
`PRINT 1`
`EOP`
`TEST`
	1+2+3+ ... + 100 = 5050

`DOWNLOAD "TEST.BAS"`
`data$ =""`
`SIZE 4,0.3`
`GAP 0,0`
`DIRECTION 1`
`INPUT "Data: ",data$`
`WHILE data$ <> "Quit"`
`CLS`
`TEXT 10,10, "3",0,1,1, "Data: "+data$`
`PRINT 1`
`INPUT "Data: ",data$`
`WEND`
`CLS`
`TEXT 10,10, "3",0,1,1, "Quit BAS"`
`PRINT 1`
`EOP`
`TEST`
	Quit BAS
	Data: quit
	Data: 67890
	Data: 12345

12345 67890 quit Quit	

10.20 DO…LOOP

説明

条件が True の間、プログラム構成文のブロックが繰り返されます。

構文

```
DO
    [statement]
    [EXITDO]
    [statement]
LOOP

DO WHILE condition
    [statement]
    [EXITDO]
    [statement]
LOOP

DO UNTIL condition
    [statement]
    [EXITDO]
    [statement]
LOOP

DO
    [statement]
    [EXITDO]
    [statement]
LOOP WHILE condition

DO
    [statement]
    [EXITDO]
    [statement]
LOOP UNTIL condition
```

パラメーター	説明
condition	利用可能な関連演算子: <, >, =, <=, >=, <>
	*関連演算子 <>, 不等
Statement	条件が True の間、1 つ以上のプログラム構成文が実行されます
EXITDO	ループから抜ける

次のページに続きます。

例

サンプルコード	結果
DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO I=I+1 TOTAL=TOTAL+I IF I=100 THEN EXITDO LOOP SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR$(TOTAL) PRINT 1 EOP TEST	1+2+3+ ... + 100 = 5050
DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO WHILE I<=100 TOTAL=TOTAL+I I=I+1 LOOP SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR$(TOTAL) PRINT 1 EOP TEST	1+2+3+ ... + 100 = 5050
DOWNLOAD “TEST.BAS” I=0 TOTAL=0 DO UNTIL I>100 TOTAL=TOTAL+I I=I+1 LOOP SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, “3”,0,1,1, “1+2+3+ … + 100“= " + STR$(TOTAL) PRINT 1 EOP TEST	1+2+3+ ... + 100 = 5050

DOWNLOAD "TEST.BAS" I=0 TOTAL=0 DO TOTAL=TOTAL+I I=I+1 LOOP WHILE I<101 SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 ="+ STR$(TOTAL) PRINT 1 EOP TEST	1+2+3+ ... + 100 = 5050
DOWNLOAD "TEST.BAS" I=0 TOTAL = 0 DO TOTAL = TOTAL + I I=I+1 LOOP UNTIL I>100 SIZE 4,0.5 GAP 0,0 CLS TEXT 10,10, "3",0,1,1, "1+2+3+ ... + 100 = " + STR$(TOTAL) PRINT 1 EOP TEST	1+2+3+ ... + 100 = 5050

10.21 IF…THEN…ELSE…ENDIF LOOP

説明

1 つ以上のプログラム構成文を条件的に実行するために **IF…THEN** ブロックを使用します。1 行の構文か複数行の“ブロック” 構文のどちらか一方が利用できます。

構文

```
IF condition THEN statement
Note the single-line form of IF …THEN does not use an ENDIF statement.

Or

IF condition THEN
    Statements
ENDIF

Or

IF condition THEN
    Statements
ELSE
    Statements
ENDIF

Or

IF condition 1 THEN
    Statement block 1
ELSEIF condition 2 THEN
    Statement block 2
    ...
ELSEIF condition n THEN
    Statement block n
ENDIF
```

備考：

IF…THEN…ELSE で 1 行（256 文字以内）のコマンドを入力します。

パラメーター	説明
condition	利用可能な関連演算子: <, >, =, <=, >=, <> *関連演算子 <>, 不等.
Statement	1 つのプログラム構成文のみ利用可能

次のページに続きます。

例

サンプルコード	結果

DOWNLOAD "DEMO.BAS"
SIZE 4,4
GAP 0,0
DIRECTION 1
CLS
A=0
B=0
C=0
D=0
E=0
F=0
G=0
H=0
J=0
K=0
L=0
FOR I=1 TO 100
IF I-INT(I/1)*1=0 THEN A=A+I
IF I-INT(I/2)*2=1 THEN B=B+I ELSE C=C+I
IF I-INT(I/3)*3=0 THEN
D=D+I
ENDIF
IF I-INT(I/5)*5=0 THEN
E=E+I
ELSE
F=F+I
ENDIF
IF I-INT(I/7)*7=0 THEN
G=G+I
ELSEIF I-INT(I/17)*17=0 THEN
H=H+I
ELSEIF I-INT(I/27)*27=0 THEN
J=J+I
ELSEIF I-INT(I/37)*37=0 THEN
K=K+I
ELSE
L=L+I
ENDIF
NEXT
TEXT 100,110,"3",0,1,1,"(1) 1+2+3+...+100="+STR$(A)
TEXT 100,160,"3",0,1,1,"(2) 1+3+5+...+99="+STR$(B)
TEXT 100,210,"3",0,1,1,"(3) 2+4+6+...+100="+STR$(C)
TEXT 100,260,"3",0,1,1,"(4) 3+6+9+...+99="+STR$(D)
TEXT 100,310,"3",0,1,1,"(5) 5+10+15+...+100="+STR$(E)
TEXT 100,360,"3",0,1,1, " (1)-(5)="+STR$(F)
TEXT 100,410,"3",0,1,1,"(6) 7+14+21+...+98="+STR$(G)
	(1) 1+2+3+...+100=5050
	(2) 1+3+5+...+99=2500
	(3) 2+4+6+...+100=2550
	(4) 3+6+9+...+99=1683
	(5) 5+10+15+...+100=1050
	(1)-(5)=4000
	(6) 7+14+21+...+98=735
	(7) 17+34+51+...+85=255
	(8) 27+54+...+81=162
	(9) 37+74=111
	(1)-(6)-(7)-(8)-(9)=3787

TEXT 100,460,"3",0,1,1,"(7)17+34+51+...+85=” +STR$(H) TEXT 100,510,"3",0,1,1,"(8)27+54+...+81="+STR$(J) TEXT 100,560,"3",0,1,1,"(9)37+74="+STR$(K) TEXT 100,610,"3",0,1,1," (1)-(6)-(7)-(8)-(9)=”+STR$(L) PRINT 1,1 EOP	
DOWNLOAD F, "TEST.BAS" SIZE 4,1 GAP 0,0 DIRECTION 1 CLS A=85 B=10 :START IF A<100 THEN GOTO L1 ELSE GOTO L2 :L1 CLS TEXT 100,10,"3",0,1,1,STR$(A) + " IS SMALLER THEN 100" PRINT 1 A=A+B GOTO START ENDIF :L2 CLS TEXT 100,10,"3",0,1,1,STR$(A) + "IS LAGER THEN 100" PRINT 1 EOP TEST	105 IS LAGER THEN 100 95 IS SMALLER THEN 100 85 IS SMALLER THEN 100

備考：
式の結果がゼロ以外の場合、THEN に続く構成文が実行されます。式の結果がゼロの場合、ELSE に続く構成文が表示され、それが実行されるか、あるいは構成文の次の行が実行されます。

IF…THEN …ELSE の中にプログラム構成文のブロックがある場合、IF…THEN …ELSE 構成文の最後に ENDIF が使用されなければなりません。.

制限：
1 つのプログラム内においてネスト化された IF …THEN …ELSE の合計数は、最大 40 です。
1 つのプログラム内においてネスト化された IF …THEN …ELSE、FOR…NEXT や GOSUB RETURN の合計数は、最大 40 です。

関連コマンド

DOWNLOAD, EOP

10.22　GOSUB…RETURN

説明

サブルーチンに分岐し、**RETURN** に到達するまでプログラム構成文を実行します。

構文

```
GOSUB LABEL
    statement
END

:LABEL
    statement
RETURN
```

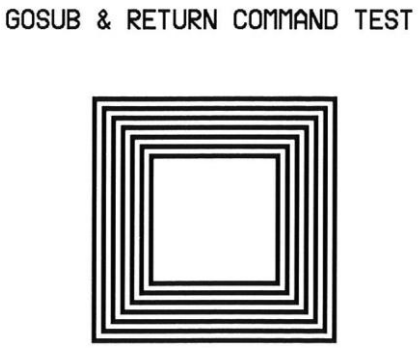
パラメーター	説明
LABEL	サブルーチンの最初。ラベルの最大の長さは 8 文字です。

例

サンプルコード

```
DOWNLOAD "GOSUB1.BAS"
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT   10,10,"3",0,1,1,"GOSUB   &   RETURN
COMMAND TEST"
GOSUB DR_BOX
PRINT 1
END
:DR_BOX
FOR I=21 TO 81 STEP 10
BOX 80+I,80+I,80+300-I,80+300-I,5
NEXT
RETURN
EOP
GOSUB1
```

結果



関連コマンド

DOWNLOAD, EOP, END, GOTO

10.23　GOTO

説明

指定ラベルに分岐させるために使用します。ラベルは 8 文字までです。

構文

```
GOTO LABEL

:LABEL
```

パラメーター	説明
LABEL	ポイントの最初。ラベルの最大の長さは 8 文字です。

例

サンプルコード

```
DOWNLOAD "GOTO1.BAS"
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
A=0
TOTAL=0
:START
IF A<100 THEN
GOTO SUM
ELSE
GOTO PRTOUT
ENDIF
:SUM
A=A+1
TOTAL=TOTAL+A
GOTO START
:PRTOUT
B$="THE  SUMMATION  OF  1..100  IS
"+STR$(TOTAL)
TEXT 10,100, "3",0,1,1,B$
PRINT 1
END
EOP
```

結果

```
THE SUMMATION OF 1..100 IS 5050
```

関連コマンド

DOWNLOAD, EOP, END, GOSUB…RETURN

10.24 INP$()

説明

コミュニケーションポートから 1 バイトを受信します。

構文

```
INP$(n)
```

パラメーター	説明
n	1：プリンターの com1 ポート

例

サンプルコード

```
DOWNLOAD "TEST.BAS"

T$=""
FOR I=1 TO 5
T$=T$+INP$(1)
NEXT

SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10, "3",0,1,1, "The received data is: "+T$
PRINT 1
EOP
TEST
12345
```

結果

```
The received data is: 12345
```

関連コマンド

INP()

10.25　INP()

説明

コミュニケーションポートから 1 バイト（ASCII 値）を受信します。

構文

```
INP(n)
```

パラメーター	説明
n	1：プリンターの com1 ポート

例

サンプルコード

```
DOWNLOAD "TEST.BAS"

155sci=0
str$=""

FOR I=1 TO 5
155sci=INP(1)
str$=str$+" " +STR$(155sci)
OUT 155sci
NEXT

SIZE 4,0.5
GAP 0,0
CLS
TEXT 10,10, "3",0,1,1, "The received data is: "+str$
PRINT 1
EOP
TEST
12345
```

結果

```
The received data is:  49 50 51 52 53
```

関連コマンド

INP$()

10.26 LOB()

説明

受信バッファ内のデータサイズを返します。

構文

```
LOB()
```

例

サンプルコード

```
DOWNLOAD "TEST.BAS"

DATA$=""

WHILE LOB()<>0
DATA$=DATA$+INP$(1)
WEND

SIZE 4,0.5
GAP 0,0
CLS
BOX 10,10,800,100,2
BLOCK 15,15,790,90, "0",0,8,8,DATA$
PRINT 1
EOP
TEST
With the "At your side." spirit in mind. the Brother Group aims to
continually create value.
```

結果

203 dpi:

With the "At your side." spirit in mind, the Brother Group aims to continually create value.

300 dpi:

With the "At your side." spirit in mind, the Brother Group aims to continually create value.

関連コマンド

INP$(), WHILE … WEND

10.27　INPUT

説明

変数を定義し、外部機器から入力された文字列を変数に代入します。

構文

```
INPUT ["Prompt string", number of digits], variables
```

コンマはセミコロンに置き換えることができます。

```
INPUT ["Prompt string"; number of digits]; variables
```

パラメーター	説明
Prompt string	最大 20 文字
Number of digits	最大 255 文字
Variables	変数名

例

サンプルテンプレート

```
DOWNLOAD F, "TEST.BAS"
SIZE 4,3
GAP 0,0
DIRECTION 1
:START
INPUT "CODE 39 : ",C39$
INPUT "EAN 13: ",12,E13$
CLS
TEXT 20,50, "3",0,1,1, "PLC or Barcode Scanner Test"
BARCODE 20,100, "39",48,1,0,2,5,C39$
BARCODE 20,200, "EAN13",48,1,0,4,4,E13$
PRINT 1
GOTO START
EOP
```

サンプル入力データ（PLC/バーコードスキャナー）

```
TEXT
123456
123456789012
```

結果

関連コマンド

DOWNLOAD, EOP, END, GOTO

10.28 PREINPUT

説明

INPUT コマンド用に開始文字を定義することができます。

構文

```
PREINPUT var$
PREINPUT CHR$(n)
```

パラメーター	説明
var$	データ前の特殊文字や文字列
N	n = 1 ~ 255

例

```
PREINPUT "<"
PREINPUT CHR$(2)
```

関連コマンド

POSTINPUT, INPUT, SET FILTER

10.29 POSTINPUT

説明

INPUT コマンド用に終わりの文字を定義することができます。

構文

```
POSTINPUT var$
POSTINPUT CHR$(n)
```

パラメーター	説明
var$	データ終わりの特殊文字や文字列
N	n = 1 ~ 255

例

```
POSTINPUT ">"
POSTINPUT CHR$(3)
```

関連コマンド

PREINPUT, INPUT, SET FILTER

10.30 SET FILTER

説明

PREINPUT と **POSTINPUT** コマンドを有効・無効にします。

構文

```
SET FILTER ON/OFF
```

パラメーター	説明
ON	**PREINPUT** と **POSTINPUT** を有効にします
OFF	**PREINPUT** と **POSTINPUT** を無効にします

備考：
LCD 搭載モデルで、**INPUT** コマンドと **SET FILTER** コマンドを USB ホスト機器とあわせて使用する場合は、**SET USBHOST** コマンドで USB ホスト機器のモードを設定する必要があります。

例

サンプルコード

```
DOWNLOAD "TEST.BAS"

PREINPUT "<="
POSTINPUT "=>"
SET FILTER ON

START:
INPUT "DATA",data1$
SIZE 4,0.25
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "DATA = "+data1$
PRINT 1
GOTO START
EOP
TEST
<=1234=><=5678=><=9012=>
```

結果

DATA = 9012

DATA = 5678

DATA = 1234

関連コマンド

PREINPUT, POSTINPUT, INPUT, SET USBHOST

10.31　REM

説明

コメント。プリンターからは無視されます。

構文

```
REM
```

例

サンプルコード

```
REM ********************************
REM This is a demonstration program*
REM ********************************
DOWNLOAD “REMARK.BAS”
SIZE 4,3
GAP 0,0
DIRECTION 1
CLS
TEXT 50,50, "3",0,1,1, "REMARK DEMO PROGRAM"
REM TEXT 50,100, "3",0,1,1, "REMARK DEMO PROGRAM"
PRINT 1,1
EOP
REMARK
```

結果

REMARK DEMO PROGRAM

関連コマンド

DOWNLOAD, EOP, END

10.32　OUT

説明

指定ポートを通じてデータを返します。

構文

```
OUT [port] "prompt",variable
OUT [port] "prompt";variable
```

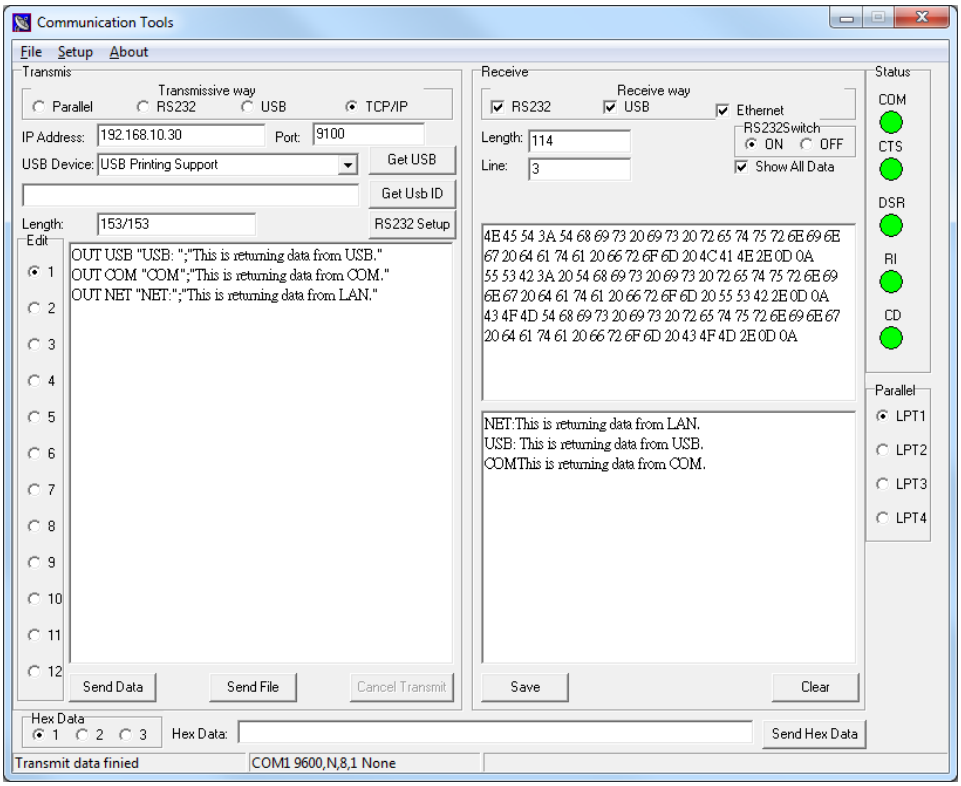
パラメーター	説明
port	オプション：データ/文字列を戻すためのポートが指定します（初期値：プリンターにデータを送るポートからデータ/文字列を返します） COM: COM ポートからデータ/文字列を返します USB: USB ポートからデータ/文字列を返します NET: LAN ポートからデータ/文字列を返します
Prompt	プロンプト文字列
Variable	アウトプットメッセージ
,	prompt と variable は<0x0D><0x0A>によって区切られます
;	variable は prompt のすぐ後ろに続きます

例

サンプルコード

```
OUT USB "USB: ";"This is returning data from USB. "
OUT COM "COM"; "This is returning data from COM. "
OUT NET "NET: ";"This is returning data from LAN. "
```

結果



10.33 OUTR

説明

RS-232C ポートのみを通じてデータを送ります。

構文

```
OUTR "prompt",variable
OUTR "prompt";variable
```

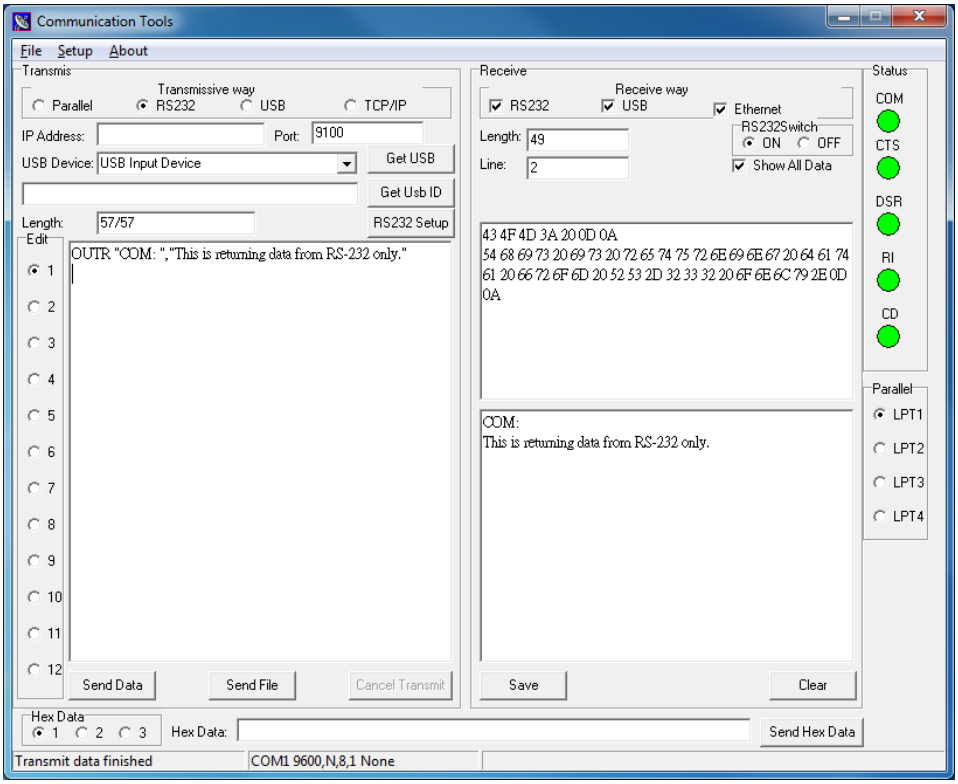
パラメーター	説明
prompt	プロンプト文字列
Variable	アウトプットメッセージ
,	prompt と variable は<0x0D><0x0A>によって区切られます
;	variable は prompt のすぐ後ろに続きます

例

サンプルコード

```
OUTR "COM: "," This is returning data from RS-232 only."
```

結果



10.34 GETKEY()

説明

停止と**給紙**キーのステータスを取得します。このコマンドはどちらかのキーが押されるまで状態を維持し、**停止**キーが押されると 0 が返り、**給紙**キーが押されると 1 が返ります。

構文

```
GETKEY()
```

停止キー	給紙キー
0	1

例

サンプルコード

```
DOWNLOAD "DEMO4.BAS"
SIZE 4,3
GAP 0,0
CLS
:START
A=GETKEY()
IF A=0 THEN GOTO PAUSEB
IF A=1 THEN GOTO FEEDB
:PAUSEB
CLS
TEXT 50,10, "4",0,1,1, "PAUSE key is pressed !"
PRINT 1
GOTO START
:FEEDB
CLS
TEXT 50,10, "4",0,1,1, "FEED key is pressed !"
PRINT 1
EOP
```

関連コマンド

DOWNLOAD, EOP, END, GOTO

10.35　INT()

説明

浮動小数点数を切り捨てます。

構文

```
INT(n)
```

パラメーター	説明
n	正か負の整数、浮動小数点数か数式

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
INPUT "Number: ",Num
CLS
REM **** To round up or down****
N=INT(Num+0.5)
IF N>Num THEN
TEXT 50,100, "3",0,1,1, "To round up= "
+STR$(N)
ELSE
TEXT 50,100, "3",0,1,1, "To round down= "
+STR$(N)
ENDIF
PRINT 1
EOP
56.2
```

結果

```
To round down= 56
```

関連コマンド

DOWNLOAD, EOP, END, ABS(), ASC(), STR$()

10.36 LEFT$()

説明

文字列の左側から、指定した文字数を取り出します。

構文

```
LEFT$(X$, n)
```

パラメーター	説明
X$	文字列
n	取り出す文字数

例

サンプルコード

```
DOWNLOAD "TEST.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="BARCODE PRINTER DEMO PRINTING"
C$=LEFT$(A$,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,100,"3",0,1,1, "10 LEFT 10 CHARS: "
+C$
PRINT 1
EOP
TEST
```

結果

```
BARCODE PRINTER DEMO PRINTING

10 LEFT 10 CHARS: BARCODE PR
```

関連コマンド

DOWNLOAD, EOP, END, RIGHT$(), MID$(), LEN(), STR$()

10.37　LEN()

説明

指定した文字列の文字数を返します。

構文

```
LEN(string)
```

パラメーター	説明
string	文字列

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
B=LEN(A$)
CLS
TEXT 10,10, "3",0,1,1,A$
TEXT 10,50, "3",0,1,1,"STRING LENGTH="
+STR$(B)
PRINT 1
EOP
DEMO
```

結果

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
STRING LENGTH=26
```

関連コマンド

DOWNLOAD, EOP, END, LEFT$(), LEN(), RIGHT$(), MID$(), STR$(), VAL()

10.38　MID$()

説明

文字列の m 番目の文字から、指定した文字数を取り出します。

構文

```
MID$(string,m,n)
```

パラメーター	説明
string	文字列
m	文字列の m 番目（1 <= m <= 文字列の数）
n	取り出す文字数

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
E$=MID$(A$,11,10)
CLS
TEXT 10,10, "3",0,1,1,A$
TEXT 10,200, "3",0,1,1,"10 MIDDLE CHARS:
"+E$
PRINT 1
EOP
DEMO
```

結果

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
10 MIDDLE CHARS: KLMNOPQRST
```

関連コマンド

DOWNLOAD, EOP, END, LEFT$(), LEN(), RIGHT$(), STR$(), VAL()

10.39 RIGHT$()

説明

文字列の右側から指定した文字数を取り出します。

構文

```
RIGHT$(X$,n)
```

パラメーター	説明
X$	文字列
n	取り出す文字数

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
D$=RIGHT$(A$,10)
CLS
TEXT 10,10,"3",0,1,1,A$
TEXT 10,150,"3",0,1,1, "10 RIGHT CHARS:
"+D$
PRINT 1
EOP
DEMO
```

結果

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
10 RIGHT CHARS: QRSTUVWXYZ
```

関連コマンド

DOWNLOAD, EOP, END, LEFT$(), LEN(), MID$(), STR$(), VAL()

10.40 STR$()

説明

指定された値や式を該当する文字列に変換します。

構文

```
STR$(n)
```

パラメーター	説明
n	正数、浮動小数点数、数式

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
F=100
G=500
H$=STR$(F+G)
CLS
TEXT 10,10, "3",0,1,1,A$
TEXT 10,60, "3",0,1,1, "F=" +STR$(F)
TEXT 10,110, "3",0,1,1, "G=" +STR$(G)
TEXT 10,160, "3",0,1,1, "F+G=" +H$
PRINT 1
EOP
DEMO
```

結果

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
F=100
G=500
F+G=600
```

関連コマンド

DOWNLOAD, EOP, END, LEFT$(), LEN(), RIGHT$(), MID$(), VAL()

10.41　STRCOMP()

説明

文字列比較の結果に基づき、-1、0、か 1 を返します。

構文

```
STRCOMP(str1$,str2$[,comp])
```

パラメーター	説明
str1$	有効な文字列式のいずれか
Str2$	有効な文字列式のいずれか
Comp	オプション：文字列比較のタイプ 0: バイナリー比較（初期値） 1: 原文比較（比較は大小文字区別）

条件	返り値
str1$ sorts ahead of str2$	-1
str1$ is equal to str2$	0
str1$ sorts after str2$	1

例

サンプルコード

```
DOWNLOAD "TEST.BAS"
STR1$ = "ABCD"
STR2$ = "abcd"

result1 = STRCOMP(STR1$,STR2$)
result2 = STRCOMP(STR1$,STR2$,1)
result3 = STRCOMP(STR2$,STR1$)

SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 100,10,"3",0,1,1,STR$(result1)+": \["]" +STR1$+"\["] sorts ahead of
\["]" +STR2$+" \["]"
TEXT 100,60,"3",0,1,1," " +STR$(result2)+": \["]"+STR1$+"\["] is equal to
\["]"+STR2$+"\["]"
TEXT 100,110,"3",0,1,1," "+STR$(result3)+": \["]"+STR2$+"\["] sorts after
\["]"+STR1$+"\["]"
PRINT 1
EOP
TEST
```

結果

```
-1: "ABCD" sorts ahead of "abcd"
 0: "ABCD" is equal to "abcd"
 1: "abcd" sorts after "ABCD"
```

関連コマンド

INSTR()

10.42 INSTR()

説明

一つの文字列または他の文字列の最初の開始位置を整数で返します。

構文

```
INSTR([start,]str1$,str2$)
```

パラメーター	説明
start	オプション：検索を始める位置を指定します。省略された場合、最初の文字位置を検索します。開始インデックスは 1 を基本とします。
Str1$	検索の対象となる文字列を指定します。
Str2$	Str1$の中から検索する文字列を指定します。

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
string$="ABC123ABC123"
searchfor$="123"
starpos=8

temp1=INSTR(string$,searchfor$)
temp2=INSTR(starpos,string$,searchfor$)

str1$=searchfor$+"in "+string$+"is "+STR$(temp1)
str2$=searchfor$+"in "+string$+"after"+STR$(starpos)+ " is "+STR$(temp2)

SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1,str1$
TEXT 10,60, "3",0,1,1,str2$
PRINT 1
EOP
DEMO
```

結果

```
123 in ABC123ABC123 is 4
123 in ABC123ABC123 after 8 is 10
```

関連コマンド

STRCOMP()

10.43 TRIM$()

説明

文字列から空白や指定文字を削除します。

構文

```
TRIM$(str$[,list$])
```

パラメーター	説明
str$	もとの文字列
List$	オプション：削除する空白や指定文字

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
data1$="1234567"
data2$="a1234567a"
data3$="[<12345>]"

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 50,020,"3",0,1,1,"LTRIM$(\["]" +data1$+" \["])        = " +LTRIM$(data1$)
TEXT 50,050,"3",0,1,1,"TRIM$ (\["]" +data1$+"\["])        = " +TRIM$(data1$)
TEXT 50,080,"3",0,1,1,"RTRIM$(\["]" +data1$+"\["])        = " +RTRIM$(data1$)
TEXT 50,110,"3",0,1,1,"LTRIM$(\["]" +data2$+ "\["], \["]a\["])    = " +LTRIM$(data2$,"a")
TEXT 50,140,"3",0,1,1,"TRIM$ (\["]" +data2$+ "\["], \["]a\["])    = " +TRIM$(data2$,"a")
TEXT 50,170,"3",0,1,1,"RTRIM$(\["]" +data2$+ "\["], \["]a\["])    = " +RTRIM$(data2$,"a")
TEXT 50,200,"3",0,1,1,"LTRIM$(\["]" +data3$+ "\["], \["][<>]\["]) = "
+LTRIM$(data3$,"[<>]")
TEXT 50,230,"3",0,1,1,"TRIM$ (\["]" +data3$+ "\["], \["][<>]\["]) = "
+TRIM$(data3$,"[<>]")
TEXT 50,260,"3",0,1,1,"RTRIM$(\["]" +data3$+ " \["], \["][<>]\["]) = "
+RTRIM$(data3$,"[<>]")
PRINT 1
EOP
DEMO
```

結果

```
LTRIM$(" 1234567 ")         = 1234567
TRIM$ (" 1234567 ")         = 1234567
RTRIM$(" 1234567 ")         =  1234567
LTRIM$("a1234567a", "a")    = 1234567a
TRIM$ ("a1234567a", "a")    = 1234567
RTRIM$("a1234567a", "a")    = a1234567
LTRIM$("[<12345>]", "[<>]") = 12345>]
TRIM$ ("[<12345>]", "[<>]") = 12345
RTRIM$("[<12345>]", "[<>]") = [<12345
```

関連コマンド

LTRIM$(), RTRIM$()

10.44 LTRIM$()

説明

文字列の左側（先頭）から空白や指定文字を削除します。

構文

```
LTRIM$(str$[,list$])
```

パラメーター	説明
str$	もとの文字列
List$	オプション：削除する空白や指定文字

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
data1$="1234567"
data2$="a1234567a"
data3$="[<12345>]"

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 50,020,"3",0,1,1,"LTRIM$(\["]" +data1$+" \["])        =" +LTRIM$(data1$)
TEXT 50,050,"3",0,1,1,"TRIM$ (\["]" +data1$+" \["])        =" +TRIM$(data1$)
TEXT 50,080,"3",0,1,1,"RTRIM$(\["]" +data1$+" \["])        =" +RTRIM$(data1$)
TEXT 50,110,"3",0,1,1,"LTRIM$(\["]" +data2$+" \["],\["]a\["])   =" +LTRIM$(data2$,"a")
TEXT 50,140,"3",0,1,1,"TRIM$ (\["]" +data2$+" \["],\["]a\["])    =" +TRIM$(data2$,"a")
TEXT 50,170,"3",0,1,1,"RTRIM$(\["]" +data2$+" \["],\["]a\["])    =" +RTRIM$(data2$,"a")
TEXT 50,200,"3",0,1,1,"LTRIM$(\["]" +data3$+" \["],\["][<>]\["])  ="
+LTRIM$(data3$,"[<>]")
TEXT 50,230,"3",0,1,1,"TRIM$ (\["] " +data3$+" \["],\["][<>]\["])  ="
+TRIM$(data3$,"[<>]")
TEXT 50,260,"3",0,1,1,"RTRIM$(\["]" +data3$+" \["],\["][<>]\["])  ="
+RTRIM$(data3$,"[<>]")
PRINT 1
EOP
DEMO
```

結果

```
LTRIM$(" 1234567 ")         = 1234567
TRIM$ (" 1234567 ")         = 1234567
RTRIM$(" 1234567 ")         =  1234567
LTRIM$("a1234567a", "a")    = 1234567a
TRIM$ ("a1234567a", "a")    = 1234567
RTRIM$("a1234567a", "a")    = a1234567
LTRIM$("[<12345>]", "[<>]") = 12345>]
TRIM$ ("[<12345>]", "[<>]") = 12345
RTRIM$("[<12345>]", "[<>]") = [<12345
```

関連コマンド

TRIM$(), RTRIM$()

10.45 RTRIM$()

説明

文字列の右側（末端）から空白や指定文字を削除します。

構文

```
RTRIM$(str$ [, list$])
```

パラメーター	説明
str$	もとの文字列
List$	オプション：削除する空白や指定文字

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
data1$="1234567"
data2$="a1234567a"
data3$="[<12345>]"

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
TEXT 50,020,"3",0,1,1, "LTRIM$(\["]" +data1$+" \["])        = " +LTRIM$(data1$)
TEXT 50,050,"3",0,1,1, "TRIM$ (\["]" +data1$+" \["])        = " +TRIM$(data1$)
TEXT 50,080,"3",0,1,1, "RTRIM$(\["]" +data1$+" \["])        = " +RTRIM$(data1$)
TEXT 50,110,"3",0,1,1, "LTRIM$(\["]" +data2$+" \["], \["]a\["])    = " +LTRIM$(data2$,"a")
TEXT 50,140,"3",0,1,1, "TRIM$ (\["]" +data2$+" \["], \["]a\["])    = " +TRIM$(data2$,"a")
TEXT 50,170,"3",0,1,1, "RTRIM$(\["]" +data2$+" \["], \["]a\["])    = " +RTRIM$(data2$,"a")
TEXT 50,200,"3",0,1,1, "LTRIM$(\["]" +data3$+" \["], \["][<>]\["]) = "
+LTRIM$(data3$,"[<>]")
TEXT 50,230,"3",0,1,1, "TRIM$ (\["]" +data3$+" \["], \["][<>]\["]) = "
+TRIM$(data3$,"[<>]")
TEXT 50,260,"3",0,1,1, "RTRIM$(\["]" +data3$+" \["], \["][<>]\["]) = "
+RTRIM$(data3$,"[<>]")
PRINT 1
EOP
DEMO
```

結果

```
LTRIM$(" 1234567 ")         = 1234567
TRIM$ (" 1234567 ")         = 1234567
RTRIM$(" 1234567 ")         =  1234567
LTRIM$("a1234567a", "a")    = 1234567a
TRIM$ ("a1234567a", "a")    = 1234567
RTRIM$("a1234567a", "a")    = a1234567
LTRIM$("[<12345>]", "[<>]") = 12345>]
TRIM$ ("[<12345>]", "[<>]") = 12345
RTRIM$("[<12345>]", "[<>]") = [<12345
```

関連コマンド

TRIM$(), LTRIM$()

10.46 TEXTPIXEL()

説明

ドット数でテキスト文字列の幅を返します。

構文

```
TEXTPIXEL(cont$,font$,size)
```

パラメーター	説明
cont$	テキスト文字列
Font $	フォントタイプ（詳細は **TEXT** コマンドの font をご参照ください）
Size	フォントサイズ（詳細は **TEXT** コマンドの x-multiplication をご参照ください）

例

サンプルコード

```
DOWNLOAD "TEST.BAS"

str$="ABCDEFG"
font$="3"
fontsize=3
strwidth=TEXTPIXEL(str$,font$,fontsize)

SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT
10,10,font$,0,fontsize,fontsize,str$
REVERSE 8,8,strwidth,72
PRINT 1
EOP
TEST
```

結果

関連コマンド

TEXT, BARCODEPIXEL()

10.47 BARCODEPIXEL()

説明

ドット数でバーコード幅を返します。

構文

```
BARCODEPIXEL(cont$, sym$, narrow, wide)
```

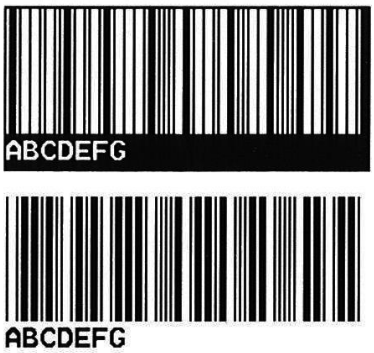
パラメーター	説明
cont$	バーコードの内容
Sym $	バーコードタイプ （詳細は **BARCODE** コマンドの code type をご参照ください）
Narrow	ナローバーの幅 （詳細は **BARCODE** コマンドの narrow をご参照ください）
Wide	ワイドバーの幅 （詳細は **BARCODE** コマンドの wide をご参照ください）

例

サンプルコード

```
DOWNLOAD "TEST.BAS"

cont$="ABCDEFG"
sym$="39"
narrow=2
wide=6
codewidth=BARCODEPIXEL(cont$,sym$,narrow,wide)

SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
BARCODE 10,10,sym$,100,1,0,narrow,wide,cont$
REVERSE 8,8,codewidth+8,132
BARCODE 10,160,sym$,100,1,0,narrow,wide,cont$
PRINT 1
EOP
TEST
```

結果



関連コマンド

BARCODE, TEXTPIXEL()

10.48 VAL()

説明

引数の数字を整数や浮動小数点数に変換します。

構文

```
VAL("numeric character")
```

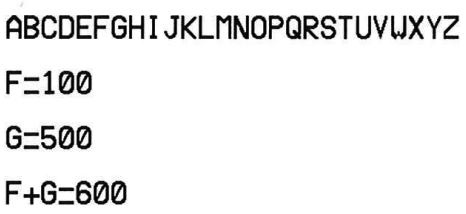
パラメーター	説明
numeric character	" 0~9"," "

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
F$="100"
G$="500"
CLS
H=VAL(F$)+VAL(G$)
I$=STR$(H)
TEXT 10,10, "3",0,1,1,A$
TEXT 10,60, "3",0,1,1, "F=" +F$
TEXT 10,110, "3",0,1,1, "G=" +G$
TEXT 10,160, "3",0,1,1, "F+G=" +I$
PRINT 1
EOP
DEMO
```

結果



関連コマンド

DOWNLOAD, EOP, END, LEFT$(), LEN(), RIGHT$(), MID$(), STR$()

10.49 NOW$()

説明

現在のプリンターの設定に従って、現在の日時を返します。返された値は常に **FORMAT$()**コマンドを使用します。

構文

```
NOW$( )
```

例

サンプルコード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Now is " +NOW$( )
TEXT 10,60, "3",0,1,1,FORMAT$(NOW$(),"Long Date")
PRINT 1
```

結果

Now is 1/9/2013 2:19:27 PM

Tuesday, January 09 2013

関連コマンド

FORMAT$( )

10.50 NOW

説明

A.D.から遡り、合計日数を返します。このグローバルな値は常に **FORMAT$()**と **DATEADD()**コマンドを使用します。

構文

```
NOW
```

例

サンプルコード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Total days since a.d. 1900: " +STR$(NOW)+ " days"
TEXT 10,50, "3",0,1,1, "Date Info in RTC: " +FORMAT$(NOW, "General Date")
TEXT 10,90, "3",0,1,1, "Date after a year: "
+FORMAT$(DATEADD("yyyy",1,NOW), "General Date")
PRINT 1
```

結果

```
Total days since a.d. 1900: 41283.597176 days
Date Info in RTC: 1/9/2013 2:19:56 PM
Date after a year: 1/9/2014 2:19:56 PM
```

関連コマンド

FORMAT$( ), DATEADD(), NOW

10.51 FORMAT$()

説明

現在のプリンターの設定に従い、現在の日付、時間、数値を返します。

構文

```
FORMAT$(expression[,style$])
```

パラメーター	説明
expression	全ての有効な式
Style$	オプション：有効な名前形式やユーザー定義形式の文字列式

所定の日付/時間形式	説明
General Date	日付・時間
Long Date	曜日・日付
Medium Date	dd-mmm-yy
Short Date	d/m/yyyy
Long Time	h:mm:ss AM/PM
Medium Time	hh:mm AM/PM
Short Time	hh:mm

ユーザー定義の日付/時間形式	説明
c	日付 ddddd・時間 ttttt
d	0 を読み取らず、日付を数字として表示する（1-31）
dd	0 を読み取り、日付を数字として表示する（01-31）
ddd	曜日（省略・Sun-Sat）
dddd	曜日（フル・Sunday-Saturday）
ddddd	現在のシステムの short date 形式設定に基づいて構成された日時（日付、月、年を含む）を表示する。初期値の short date 形式は m/d/yyyy。
dddddd	現在のシステムの long date 設定に基づいて構成された日時（日付、月、年）を表示する。初期値の long date 形式は dddd, mmmm dd, yyyy。
w	数字として曜日を表示する（1 は日曜日～7 は土曜日）
ww	数字として年の週を表示する（1-53）
m	0 を読み取らず、数字として月を表示する（1-12）。m が h や hh の直後に続く場合、月ではなく分が表示される。
mm	0 を読み取り、数字として月を表示する（01-12）。mm が h や hh の直後に続く場合、月ではなく分が表示される。
mmm	月（省略・Jan-Dec）
mmmm	月（フル・January–December）
q	数字として年間の四半期を表示する（1–4）
y	数字として年間の日付を表示する（1–366）
yy	2 桁の数字として年を表示する（00–99）
yyyy	4 桁の数字として年を表示する（100–9999）
h	複数の 0 を読み取らず、数字として時間を表示する（0–23）
hh	0 を読み取り、数字として時間を表示する（00–23）
n	複数の 0 を読み取らず、数字として分を表示する（0–59）
nn	0 を読み取り、数字として分を表示する（00–59）
s	複数の 0 を読み取らず、数字として秒を表示する（0–59）
ss	0 を読み取り、数字として秒を表示する（00–59）

ttttt	完全時間（時間、分、秒を含む）として時間を表示する（初期値の時間書式：h:mm:ss AM/PM）
AM/PM	午前中の全ての時間で大文字の AM を表示し、正午と 11:59 P.M.の間の時間は大文字 PM で表示する
am/pm	午前中の全ての時間で小文字の am を表示し、正午と 11:59 P.M.の間の時間は小文字 pm で表示する
A/P	午前中の全ての時間で大文字の A を表示し、正午と 11:59 P.M.の間の時間は大文字 P で表示する
a/p	午前中の全ての時間で小文字の a を表示し、正午と 11:59 P.M.の間の時間は小文字 p で表示する
AMPM	AMPM は大文字でも小文字でも利用できますが、表示文字列の活字ケースはお使いのシステム設定により定義された文字列と一致しなければなりません
\	フォーマット文字列中の次の文字を表示する
“string”	二重引用符の中の文字列を表示する

数字	説明
General Number	四捨五入とコンマを使うことなく、数字を入力された通りに表示する
Currency	小数点の右側に 2 桁で、ドル記号、コンマ（該当する場合）で数字を表示する。括弧の中の負の数を表す
Fixed	最低でも小数点の左側に 1 桁と右側に 2 桁で数字を表示する。コンマは表示しない
Standard	最低でも小数点の左側に 1 桁と右側に 2 桁とコンマ（該当する場合）で数字を表示する
Percent	値に 100 を掛けて、小数点の右側に 2 桁と最後に%を用いて表示する
Scientific	標準的な指数表記を使用する
Yes/No	ゼロ以外の数値：Yes、ゼロは：No
True/False	ゼロ以外の数値：True、ゼロは：False
On/Off	ゼロ以外の数値：On、ゼロは：Off

ユーザー定義の数字の形式	説明
0	桁のプレースホルダ　桁かゼロで表示する
#	桁のプレースホルダ　桁かなしで表示する
.	少数のプレースホルダ
%	パーセントのプレースホルダ　100 を掛ける
,	3 桁毎の区切り
E- E+ e- e+	指数形式
\	フォーマット文字列内の次の文字を表示する
"ABC"	二重引用符内の文字列を表示する

異なる数値のための異なる形式	説明
One section only	フォーマット式は全ての値に適用される
Two section	最初のセクションが正の値とゼロに適用され、2 番目のセクションは負の値に適用される
Three sections	最初のセクションが正の値に適用され、2 番目のセクションは負の値に適用され、そして 3 番目のセクションがゼロに適用される

例

サンプルコード

```
SIZE 800 dot,1900 dot
GAP 0,0
DIRECTION 1
CLS
TEXT 15,10, "3",0,1,1, "General Date: "+FORMAT$(NOW,"General
Date")
TEXT 15,60, "3",0,1,1, "Long Date: "+FORMAT$(NOW,"Long Date")
TEXT 15,110, "3",0,1,1, "Medium Date: "+FORMAT$(NOW,"Medium
Date")
TEXT 15,160, "3",0,1,1, "Short Date: "+FORMAT$(NOW,"Short
Date")
TEXT 15,210, "3",0,1,1, "Long Time: "+FORMAT$(NOW,"Long Time")
TEXT 15,260, "3",0,1,1, "Medium Time: "+FORMAT$(NOW,"Medium
Time")
TEXT 15,310, "3",0,1,1, "Short Time: "+FORMAT$(NOW,"Short
Time")
TEXT 15,360, "3",0,1,1, "c: "+FORMAT$(NOW,"c")
TEXT 15,410, "3",0,1,1, "d: "+FORMAT$(NOW,"d")
TEXT 15,460, "3",0,1,1, "dd: " +FORMAT$(NOW,"dd")
TEXT 15,510, "3",0,1,1, "ddd: " +FORMAT$(NOW,"ddd")
TEXT 15,560, "3",0,1,1, "dddd: " +FORMAT$(NOW,"dddd")
TEXT 15,610, "3",0,1,1, "ddddd: " +FORMAT$(NOW,"ddddd")
TEXT 15,660, "3",0,1,1, "dddddd: " +FORMAT$(NOW,"dddddd")
TEXT 15,710, "3",0,1,1, "w: " +FORMAT$(NOW,"w")
TEXT 15,760, "3",0,1,1, "ww: " +FORMAT$(NOW,"ww")
TEXT 15,810, "3",0,1,1, "m: " +FORMAT$(NOW,"m")
TEXT 15,860, "3",0,1,1, "mm: " +FORMAT$(NOW,"mm")
TEXT 15,910, "3",0,1,1, "mmm: " +FORMAT$(NOW,"mmm")
TEXT 15,960, "3",0,1,1, "mmmm: " +FORMAT$(NOW,"mmmm")
TEXT 15,1010, "3",0,1,1, "q: " +FORMAT$(NOW,"q")
TEXT 15,1060, "3",0,1,1, "y: " +FORMAT$(NOW,"y")
TEXT 15,1110, "3",0,1,1, "yy: " +FORMAT$(NOW,"yy")
TEXT 15,1160, "3",0,1,1, "yyyy: " +FORMAT$(NOW,"yyyy")
TEXT 15,1210, "3",0,1,1, "h: " +FORMAT$(NOW,"h")
TEXT 15,1260, "3",0,1,1, "hh: " +FORMAT$(NOW,"hh")
TEXT 15,1310, "3",0,1,1, "n: " +FORMAT$(NOW,"n")
TEXT 15,1360, "3",0,1,1, "nn: " +FORMAT$(NOW,"nn")
TEXT 15,1410, "3",0,1,1, "s: " +FORMAT$(NOW,"s")
TEXT 15,1460, "3",0,1,1, "ss: " +FORMAT$(NOW,"ss")
TEXT 15,1510, "3",0,1,1, "ttttt: " +FORMAT$(NOW,"ttttt")
TEXT 15,1560, "3",0,1,1, "AM/PM: " +FORMAT$(NOW,"AM/PM")
TEXT 15,1610, "3",0,1,1, "am/pm: " +FORMAT$(NOW,"am/pm")
TEXT 15,1660, "3",0,1,1, "A/P: " +FORMAT$(NOW,"A/P")
TEXT 15,1710, "3",0,1,1, "a/p: " +FORMAT$(NOW,"a/p")
TEXT 15,1760, "3",0,1,1, "AMPM: " +FORMAT$(NOW,"AMPM")
TEXT 15,1810, "3",0,1,1, "\:" +FORMAT$(NOW,"To\da\y i\s ddddd")
TEXT 15,1860, "3",0,1,1, "string: " +FORMAT$(NOW,"To\da\y i\s
ddddd")
PRINT 1
```

結果

```
General Date:1/9/2013 2:46:18 PM
Long Date:Tuesday, January 09 2013
Medium Date:09-Jan-13
Short Date:1/9/2013
Long Time:2:46:18 PM
Medium Time:02:46 PM
Short Time:14:46
c:1/9/2013 2:46:18 PM
d:9
dd:09
ddd:Tue
dddd:Tuesday
ddddd:1/9/2013
dddddd:Tuesday, January 09 2013
w:3
ww:2
m:1
mm:01
mmm:Jan
mmmm:January
q:1
y:9
yy:13
yyyy:2013
h:14
hh:14
n:46
nn:46
s:18
ss:18
ttttt:2:46:18 PM
AM/PM:PM
am/pm:pm
A/P:P
a/p:p
AMPM:PM
\:Today is 1/9/2013
string:Today is 1/9/2013
```

サンプルコード

```
SIZE 800 dot,850 dot
GAP 0,0
DIRECTION 1
CLS
TEXT 15,10, "3",0,1,1, "General Number:
"+FORMAT$(1234.5,"General Number")
TEXT 15,60, "3",0,1,1, "Currency: "+FORMAT$(1234.5,"Currency")
TEXT 15,110, "3",0,1,1, "Fixed: "+FORMAT$(1234.5,"Fixed")
TEXT 15,160, "3",0,1,1, "Standard: "+FORMAT$(1234.5,"Standard")
TEXT 15,210, "3",0,1,1, "Percent: "+FORMAT$(1234.5,"Percent")
TEXT 15,260, "3",0,1,1, "Scientific:
"+FORMAT$(1234.5,"Scientific")
TEXT 15,310, "3",0,1,1, "Yes/No: "+FORMAT$(1234.5,"Yes/No")
TEXT 15,360, "3",0,1,1, "Yes/No: "+FORMAT$(0,"Yes/No")
TEXT 15,410, "3",0,1,1, "True/False: "+FORMAT$(0,"True/False")
TEXT 15,460, "3",0,1,1, "On/Off: "+FORMAT$(0,"On/Off")
TEXT 15,510, "3",0,1,1, "00000.00: "+FORMAT$(1234.5,"00000.00")
TEXT 15,560, "3",0,1,1, "#####.##: "+FORMAT$(1234.5,"#####.##")
TEXT 15,610, "3",0,1,1, "##,##0.00:
"+FORMAT$(1234.5,"##,##0.00")
TEXT 15,660, "3",0,1,1, "$##0.00: "+FORMAT$(1234.5,"$##0.00")
TEXT 15,710, "3",0,1,1, "$0.00%: "+FORMAT$(1234.5,"0.00%")
TEXT 15,760, "3",0,1,1, "Yes/No: "+FORMAT$(-12.3,"Yes/No")
TEXT 15,810, "3",0,1,1, "0.00;(0.00): "+FORMAT$(-
12.3,"0.00;(0.00)")
PRINT 1
```

結果

```
General Number: 1234.5
Currency: $1,234.50
Fixed: 1234.50
Standard: 1,234.50
Percent: 123450.00%
Scientific: 1.23E+03
Yes/No: Yes
Yes/No: No
True/False: False
On/Off: Off
00000.00: 01234.50
#####.##: 1234.5
##,##0.00: 1,234.50
$##0.00: $1234.50
$0.00%: 123450.00%
Yes/No: Yes
0.00;(0.00): (12.30)
```

関連コマンド

NOW$(), DATEADD(), NOW

10.52 DATEADD()

説明

日付に指定した単位の時間を加算して返します。返された値は常に **FORMAT$()**コマンドと使用します。

構文

```
DATEADD(interval$,number,date)
```

パラメーター / **説明**

`interval$,`

加算する時間/日付の単位。以下の値のいずれかとなります。

Interval$	インターバル単位
"yyyy"	年
"q"	四半期
"m"	月
"y"	年間通算日
"d"	日付
"w"	曜日
"ww"	週
"h"	時
"n"	分
"s"	秒

`Number` 加算する時間

`Date` 日時

日付形式: `yyyy/mm/dd`

時間形式: `hh:nn:ss`

例

サンプルコード 1

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Current RTC info: " +NOW$()
TEXT 10,60, "3",0,1,1, "-1 year: " +FORMAT$(DATEADD("yyyy",-1, " 11/26/2012
10:08:00"), "yyyy/mm/dd hh:nn:ss")
TEXT 10,110, "3",0,1,1, "+9 months: " +FORMAT$(DATEADD("m",9,NOW), "Short Date")
TEXT 10,160, "3",0,1,1, "-8 hours: " +FORMAT$(DATEADD("h",-8,NOW), "Short Time")
TEXT 10,210, "3",0,1,1, "+5 mins: " +FORMAT$(DATEADD("n",5,NOW), "Short Time")
TEXT 10,260, "3",0,1,1, "+00 day: " +FORMAT$(NOW, "Short Date")
TEXT 10,310, "3",0,1,1, "+20 days: " +FORMAT$(DATEADD("d",20,NOW), "Short Date")
TEXT 10,360, "3",0,1,1, "-20 day: " +FORMAT$(DATEADD("d",-20,NOW), "Short Date")
PRINT 1
```

結果 1

```
Current RTC info: 1/9/2013 3:20:06 PM
-1 year: 2011/11/26 10:08:00
+9 months: 10/9/2013
-8 hours: 07:20
+5 mins: 15:25
+00 day: 1/9/2013
+20 days: 1/29/2013
-20 day: 12/20/2012
```

サンプルコード 2

```
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 10,60, "3",0,1,1, "-1 year:   "+FORMAT$(DATEADD("yyyy", -1, "11/26/2012
10:08"),"yyyy/mm/dd hh:nn AM/PM")
TEXT 10,110,"3",0,1,1,"+9 months: "+FORMAT$(DATEADD("m",9,"11/26/2012 10:08"),"yyyy/mm/dd
hh:nn AM/PM")
TEXT 10,160,"3",0,1,1,"+8 hours:  "+FORMAT$(DATEADD("h", +8,"11/26/2012
10:08"),"yyyy/mm/dd hh:nn AM/PM")
TEXT 10,210,"3",0,1,1,"+00 day:   "+FORMAT$("11/26/2012 10:08:00","yyyy/mm/dd hh:nn
AM/PM")
TEXT 10,260,"3",0,1,1,"+20 days:  "+FORMAT$(DATEADD("d",20,"11/26/2012 10:08"),"yyyy/mm/dd
hh:nn AM/PM")
TEXT 10,310,"3",0,1,1,"-20 days:  "+FORMAT$(DATEADD("d", -20,"11/26/2012
10:08"),"yyyy/mm/dd hh:nn AM/PM")
PRINT 1
```

結果 2

```
-1 year:   2011/11/26 10:08 AM
+9 months: 2013/08/26 10:08 AM
+8 hours:  2012/11/26 06:08 PM
+00 day:   2012/11/26 10:08 AM
+20 days:  2012/12/16 10:08 AM
-20 days:  2012/11/06 10:08 AM
```

10.53 FSEARCH()

説明

指定した文字列の位置を返します。

構文

```
FSEARCH(file handle, STR$)
```

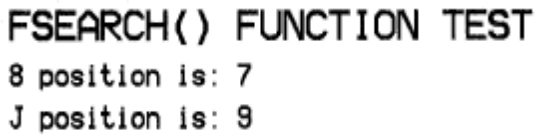
パラメーター	説明
file handle	0 または 1
STR$	文字列

例

サンプルコード

```
DOWNLOAD "DATA1",10,1234567890
DOWNLOAD "DATA2",15,ABCDEFGHIJKLMNO
DOWNLOAD "Test.BAS"
SIZE 4,1.5
GAP 0,0
DIRECTION 1
CLS
OPEN "DATA1",0
OPEN "DATA2",1
TEXT 10,90,"4",0,1,1,"FSEARCH() FUNCTION
TEST"
A=FSEARCH(0,"8")
B=FSEARCH(1,"J")
TEXT 10,140,"3",0,1,1,"8 position
is:"+STR$(A)
TEXT 10,180,"3",0,1,1,"J position
is:"+STR$(B)
PRINT 1
EOP
Test
```

結果



10.54 TOUCHPRESS()

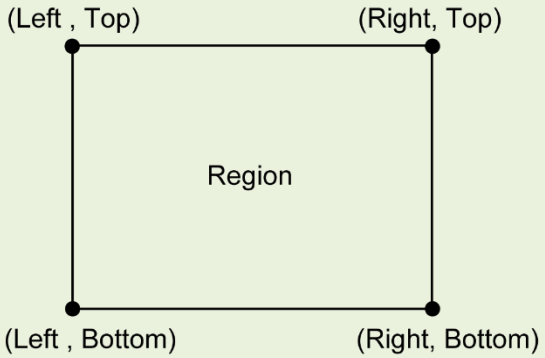
説明

タッチスクリーンの状態を検知します。指定した領域が押された場合 1 を返し、それ以外は 0 を返します。

構文

```
TOUCHPRESS (left, top, right, bottom)
```

パラメーター	説明
left	左座標（ピクセル）
top	上座標（ピクセル）
right	右座標（ピクセル）
bottom	下座標（ピクセル）



備考：TJ-4021TN/TJ-4121TN/TJ-4121TNR のみ

例

Sample Code

```
DOWNLOAD "DEMO.BAS"
:START
IF TOUCHPRESS(0,90,272,120) <> 0 THEN GOTO A
GOTO START
ENDIF
:A
CLS
SIZE 4,1
GAP 0,0
DIRECTION 1
TEXT 30,30,"3",0,1,1,"TOUCH TEST!!"
PRINT 1,1
EOP
DEMO
```

10.55 RECORDSET$()

説明

表から値を返します。表はグリッドフォーマットと行と列の表形式で表示されます。

構文

```
RECORDSET$(TABLE$, ROW, COLUMN [, DELIMITER])
```

パラメーター	説明
TABLE$	表の名前
ROW	行の番号
COLUMN	列の番号（もしくは名前）
DELIMITER	オプション：表の区切り文字（初期値 ：09H <Tab>）

備考：
ROW は常に数字ですが、COLUMN は数字もしくは名前が使用できます。

例

サンプルコード 1:

```
DOWNLOAD F,"TEST.CSV",75,3
Name,Age,Height,Weight
John,18,180,80
Mary,30,150,50
Mark,65,170,65


DOWNLOAD F,"TEST.BAS"
CLOSE 0
SIZE 4,2
GAP 0,0
DIRECTION 1
CLS
TEXT 100,50,"3",0,1,1,"Row 1 and Column 1 = "
+ RECORDSET$("TEST.CSV", 1, 1, ASC(","))
TEXT 100,100,"3",0,1,1,"Row 2 and Column 1 = "
+ RECORDSET$("TEST.CSV", 2, 1, ASC(","))
TEXT 100,150,"3",0,1,1,"John Age = " +
RECORDSET$("TEST.CSV", 1, 2,ASC(","))
TEXT 100,200,"3",0,1,1,"Mary Age = " +
RECORDSET$("TEST.CSV", 2, 2,ASC(","))
TEXT 100,250,"3",0,1,1,"John Height = " +
RECORDSET$("TEST.CSV",1,"Height", ASC(","))
TEXT 100,300,"3",0,1,1,"Mary Height = " +
RECORDSET$("TEST.CSV",2,"Height", ASC(","))
PRINT 1

EOP
TEST
```

結果

```
Row 1 and Column 1 = John
Row 2 and Column 1 = Mary
John Age = 18
Mary Age = 30
John Height = 180
Mary Height = 150
```

表形式(TEST.CSV)

行の番号	3			
列の番号	Name	Age	Height	Weight
Row 1	John	18	180	80
Row 2	Mary	30	150	50
Row 3	Mark	65	170	65
	Column 1	Column 2	Column 3	Column 4

サンプルコード 2:

```
DOWNLOAD "TEST.CSV",121,6,
Number,String
1234,ABCD
"12,34","AB,CD"
"12
34","AB
CD"
"12""34","AB""CD"
"""1234","""ABCD"
"1234""","ABCD"""


OUT RECORDSET$("TEST.CSV", 1, "Number",
ASC(","))
OUT RECORDSET$("TEST.CSV", 2, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 3, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 4, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 5, 1, ASC(","))
OUT RECORDSET$("TEST.CSV", 6, 1, ASC(","))

OUT ""

OUT RECORDSET$("TEST.CSV", 1, "String",
ASC(","))
OUT RECORDSET$("TEST.CSV", 2, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 3, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 4, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 5, 2, ASC(","))
OUT RECORDSET$("TEST.CSV", 6, 2, ASC(","))
```

Table format (TEST.CSV)

行の番号	6	
列の番号	Number	String
Row 1	1234	ABCD
Row 2	12,34	AB,CD
Row 3	12 34	AB CD
Row 4	12"34	AB"CD
Row 5	"1234	"ABCD
Row 6	1234"	ABCD"
	Column 1	Column 2

返り値

```
1234
12,34
12
34
12"34
"1234
1234"

ABCD
AB,CD
AB
CD
AB"CD
"ABCD
ABCD"
```

10.56　REPLACE$()

説明

指定した文字列を別の文字列に置き換えます。

構文

```
REPLACE$(“str1$”,”sub1$”,”sub2$”)
```

パラメーター	説明
str1$	置き換える文字を含むもとの文字列
sub1$	置き換えたい文字列
sub2$	置き換える文字列

例

サンプルコード

```
DOWNLOAD F,"TEST.BAS"
SIZE 3,2
GAP 0,0
DIRECTION 1
INPUT A$
DATA$ = REPLACE$(A$,"ABC","123")
CLS
TEXT 100,100,"3",0,1,1,DATA$
PRINT 1
EOP
TEST
ABCDEFG
```

結果

11 デバイス動作設定コマンド

11.1SET COUNTER

説明

実際のカウンターや変数を設定します。プログラム内で数のカウントとその増加も設定できます。カウンターは、桁(0~9~0)、小文字(a~z~a)と大文字(A~Z~A)の 3 種類あります。

構文

```
SET COUNTER @n step
@n= "Expression "
```

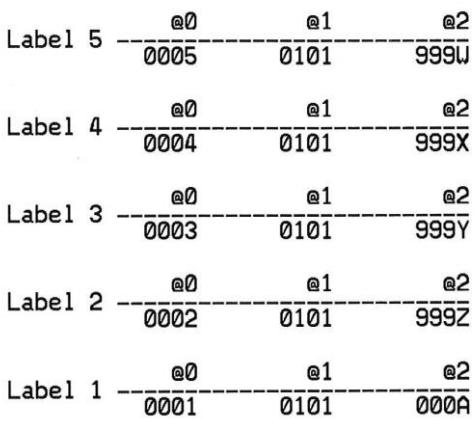
パラメーター	説明
@n	n: カウント数。プリンター—では 61 のカウンター (@0 ~ @60) が使用できます。@0 ～@50 はプリンター—を再起動すると消去されます。@51～@60 はプリンター—が初期化されるまでプリンター—内に記憶されます。
step	カウンターの増加は正または負に設定できます。 -999999999 <= step <= 999999999 固定変数としてカウンターを使用する場合、増加を 0 に設定します。
Expression	最初の文字列。文字列の長さは 101 バイトです。

例

サンプルコード

```
SET COUNTER @0 +1
SET COUNTER @1 +0
SET COUNTER @2 -1
SET COUNTER @3 1

@0=" 0001"
@1=" 0101"
@2=" 000A"
@3=" 1"

SIZE 4,0.5
GAP 0,0
DIRECTION 1
CLS
TEXT 600,10,"3",0,1,1,3,"@0       @1       @2"
TEXT 600,30,"3",0,1,1,3, "Label"  +@3+"  -------
-------------------"
TEXT 600,50,"3",0,1,1,3,@0+"      "  +@1+"      "
+@2
PRINT 5
```

結果



関連コマンド

PRINT, TEXT, BARCODE

11.2 SET CUTTER

説明

フルカットを有効/無効にして、何枚毎に印字ラベルをカットするかを指定します。この設定は電源を切った後もプリンターのメモリに記憶されます。

構文

```
SET CUTTER OFF/BATCH/pieces
```

パラメーター	説明
OFF	カッター機能をオフにする
BATCH	印刷ジョブの終わりにラベルをカットする設定にする
Pieces	カット毎枚数を指定する（0<= pieces <=65535）

例

サンプルコード	結果
SIZE 3,3 GAP 0,0 SET CUTTER OFF SET PEEL OFF CLS TEXT 50,50, "3",0,1,1, "SET CUTTER OFF" PRINT 3	カッター機能をオフにする
SET CUTTER BATCH CLS TEXT 50,50, "3",0,1,1, "SET CUTTER BATCH" PRINT 3,2	ジョブ（6 枚印刷する）ごとにカットする
SET CUTTER 1 CLS TEXT 50,50, "3",0,1,1, "SET CUTTER 1" PRINT 3,2	1 枚印刷するごとにカットする
CLS TEXT 50,50, "3",0,1,1, "SET CUTTER 2" PRINT 3,2	2 枚印刷するごとにカットする

関連コマンド

OFFSET, PRINT, SET PARTIAL_CUTTER

11.3 SET PARTIAL_CUTTER

説明

パーシャルカットを有効/無効にし、何枚毎に印字ラベルをカットするかを指定します。この設定は電源を切った後もプリンターのメモリに記憶されます。このコマンドはカット後の逆送りを防止します。

構文

```
SET PARTIAL_CUTTER OFF/BATCH/Pieces
```

パラメーター	説明
OFF	カッター機能をオフにする
BATCH	印刷ジョブの終わりにラベルをカットする設定にする
Pieces	カット毎枚数を指定する（0<= pieces <=65535）

備考： パーシャルカットが可能なカッターモジュールを装着したプリンター—のみに対応しています。

例

サンプルコード

```
REM **SET PARTIAL_CUTTER FUNCTION OFF EXAMPLE PROGRAM**
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET PARTIAL_CUTTER OFF
CLS
TEXT 50,50, "3",0,1,1, "SET PARTIAL_CUTTER OFF"
PRINT 3
REM ***This program cuts once at the batch***
SET PARTIAL_CUTTER BATCH
CLS
TEXT 50,50, "3",0,1,1, "SET PARTIAL_CUTTER BATCH"
PRINT 3,2
REM ***This program cuts every label***
SET PARTIAL_CUTTER 1
CLS
TEXT 50,50, "3",0,1,1, " SET PARTIAL_CUTTER 1"
PRINT 3,2
REM ***This program cuts 2 label***
SET PARTIAL_CUTTER 2
CLS
TEXT 50,50, "3",0,1,1, "SET PARTIAL_CUTTER 2"
PRINT 3,2
```

関連コマンド

OFFSET, PRINT, SET CUTTER

11.4SET BACK

説明

カット後のラベルの逆送りを有効/無効にします。**SET CUTTER** コマンドの後で使用します。

構文

```
SET BACK OFF/ON
```

パラメーター	説明
OFF	逆送り機能を無効にする
ON	逆送り機能を有効にする

例

サンプルコード

```
REM **SET BACK FUNCTION OFF 例 PROGRAM**
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 1
REFERENCE 0,0
SET CUTTER 1
SET BACK OFF
CLS
TEXT 50,50, "3",0,1,1, "SET BACK OFF "
PRINT 3
CLS
SET CUTTER 1
SET BACK ON
TEXT 50,50, "3",0,1,1, "SET BACK ON "
PRINT 3
```

関連コマンド

OFFSET, PRINT, SET CUTTER

11.5 SET BACK_BEFORE_PRINT

説明

カット後にラベルを逆送りするタイミングを変更します。この設定は、プリンターの再起動やリセット後も保持されます。ファームウェアの更新や工場出荷時設定を実行した場合は、初期値に戻ります。

構文

```
SET BACK_BEFORE_PRINT ON/OFF
```

パラメーター	説明
OFF	カット直後、用紙は印刷開始位置までバックフィードする（初期値）
ON	カット直後、用紙は印刷開始位置に戻らない。印刷直前（プリンターが次の印刷データを受け取ってから）、用紙をバックフィードする

備考：この機能は、次のファームウェアバージョンからお使いいただけます。

モデル名	ファームウェアバージョン
TD-4420TN/4520TN	V1.04 S57 以降
TD-4650TNWB/4750TNWB/4750TNWBR	V1.04 S53 以降
TJ-4021TN/4121TN/4121TNR	B2.12.S70 以降
TJ-4420TN/4520TN/4620TN	B1.17.S32 以降

例

```
SET BACK_BEFORE_PRINT OFF
SET BACK_BEFORE_PRINT ON
```

関連コマンド

SET CUTTER

11.6 SET KEYn

説明

KEYn 機能を有効/無効にします。**KEYn** 機能の設定前に、最初に **KEYn** の初期値を無効にします。この設定はプリンターの電源がオフになっても、プリンター内に記憶します。

構文

```
SET KEYn ON/OFF/DEFAULT/MENU/PAUSE/PRINT m/FEED/BACKFEED/FORMFEED/CUT/INPUT
"string "
```

パラメーター	説明
n	0, 1, 2, 3, 4, 5, 6
ON	KEYn 機能を有効にします
OFF	KEYn 機能を無効にします
DEFAULT	KEYn 機能を初期値に戻します
MENU	**メニュー**キーを設定します
PAUSE	**停止**キーを設定します
PRINT m	**印字**キーを設定します m: 印刷毎の印刷枚数を設定します(0 < m < 32000)
FEED	手動でラベルの送り量を調整できるように**給紙**キーを設定します
BACKFEED	手動でラベルの逆送り量を調整できるように**逆送り**キーを設定します
FORMFEED	次のラベルの先端に送るため**頭出し**キーを設定します 例：フォーマットが"size 4,6”の場合、プリンターは 6 送ります
CUT	**カット**キーを設定します
INPUT "string "	コマンドを送信する**押下**キーを設定します （例：SET KEY1 INPUT "CONFIG " + CHR$(13) + CHR$(10)）

KETn 機能の初期値は以下のとおりです

機種	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6
TD-4T シリーズ		給紙					
RJ シリーズ		給紙					
TJ シリーズ		給紙	メニュー	上	右	左	下

例

サンプルコード

```
DOWNLOAD “DEMO.BAS”
SIZE 3,1
GAP 0,0
DENSITY 8
SPEED 3
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET KEY1 OFF
SET KEY2 OFF
SET KEY3 OFF
KEY1=0
KEY2=0
KEY3=0
```

```
:START
IF KEY1=1 THEN
CLS
TEXT 100,10, “3”,0,1,1, “KEY1 (MENU key) is pressed!! “
PRINT 1,1
ELSEIF KEY2=1 THEN
CLS
TEXT 100,10, “3”,0,1,1, “KEY2 (PAUSE key) is pressed!! “
PRINT 1,1
ELSEIF KEY3=1 THEN
CLS
TEXT 100,10, “3”,0,1,1, “KEY3 (FEED key) is pressed!! “
TEXT 100,60, “3”,0,1,1, “End of test”
PRINT 1,1
SET KEY1 ON
SET KEY2 ON
SET KEY3 ON
END
ENDIF
GOTO START
EOP
DEMO
```

関連コマンド

OFFEST, PRINT

11.7 SET LEDn

説明

LED のオン/オフ機能を制御します。

構文

```
SET LED1 ON/OFF
SET LED2 ON/OFF
SET LED3 ON/OFF
```

LED no.	初期設定
LDE1	電源オン/オフ
LED2	プリンター—オンライン/オフライン
LED3	エラー/通常

パラメーター	説明
ON	**LEDn** 機能を有効にします
OFF	**LEDn** 機能を無効にします

LEDn の初期値は以下のとおりです

モデル	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED2 & LED3
TD-4T シリーズ **備考**：このシリーズは LED1=LED2	緑色	緑色	赤色					黄色
RJ シリーズ	緑色	緑色	赤色	バッテリ 1	バッテリ 2	バッテリ 3	WIFI/BT	黄色
TJ シリーズ	緑色	緑色	赤色					黄色

例

サンプルコード

```
DOWNLOAD "DEMO4.BAS "
SET LED1 OFF
SET LED2 OFF
SET LED3 OFF
FOR I=1 TO 100
LED1=0
LED2=0
LED3=0
IF I-INT(I/2)*2=0 THEN
LED1=1
ELSEIF I-INT(I/3)*3=0 THEN
LED2=1
ELSE
LED3=1
ENDIF
NEXT
LED1=1
LED2=1
LED3=0
SET LED1 ON
SET LED2 ON
SET LED3 ON
EOP
DEMO4
```

11.8 SET PEEL

説明

ハクリ機能を有効/無効にします。初期値は「オフ」です。 この機能がオンの時、プリンターはラベル印刷した後に毎回停止し、ハクリしたラベルが取り除かれるまで次の印刷をしません。この設定は電源をオフにしても、プリンターのメモリに保存されます。

構文

```
SET PEEL ON/OFF
```

パラメーター	説明
ON	ハクリ機能を有効にします
OFF	ハクリ機能を無効にします

例

サンプルコード

```
REM ***SELF-PEELING FUNCTION ON***
SIZE 4,4
GAP 0,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL ON
CLS
TEXT 50,100, "3 ",0,1,1, "SELF-PEELING FUNCTION TEST "
PRINT 5
```

関連コマンド

OFFEST, PRINT

11.9 SET REWIND

説明

内部巻き戻し機能を有効/無効にします。この設定は電源をオフにしてもプリンターのメモリに保存されます。

構文

```
SET REWIND ON/OFF/RS232
```

パラメーター	説明
ON	内部巻き戻し機能を有効にします
OFF	内部巻き戻し機能や外部巻き戻しモジュールコマンドを無効にします
RS232	外部巻き戻しモジュールコマンドを有効にします(RS-232C ポート/プル高信号を経由)

例

サンプルコード

```
REM ***REWIND FUNCTION ON***
SIZE 4,4
GAP 0.12,0
DENSITY 8
SPEED 6
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET REWIND ON
CLS
TEXT 50,100, "3 ",0,1,1, "REWIND FUNCTION TEST "
PRINT 500
```

関連コマンド

OFFEST, PRINT

11.10 SET TEAR

説明

ラベルのギャップ/ブラックマークを手切り位置まで送るかを指定します。この設定は電源をオフにしてもプリンターのメモリに保存されます。

構文

```
SET TEAR ON/OFF
```

パラメーター	説明
ON	印刷後、ラベルのギャップは手切り位置で停止します。
OFF	印刷後、ラベルのギャップは手切り位置で停止しません。ラベルを印字ヘッド下で停止させます。

例

サンプルコード

```
REM ***TEAR FUNCTION ON***
SIZE 3,3
GAP 0.08,0
DENSITY 8
SPEED 4
DIRECTION 0
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET TEAR ON
CLS
TEXT 50,100, "3 ",0,1,1, "TEAR FUNCTION TEST "
PRINT 1
```

関連コマンド

SET PEEL, SET CUTTER

11.11 SET GAP

説明

ギャップセンサー検知感度を設定します。**給紙/停止**ボタンを押しながらプリンターの電源を入れた時、プリンター—は自動ギャップセンサーキャリブレーションを開始します。ただし、センサーにより台紙の厚さとラベル有の台紙の厚さに大きな違いが検知された場合や、プレ印刷されたマークやパターンがあるラベルの場合、自動キャリブレーションが正常に動作しない可能性があります。その場合、正しい設定値を得るために、このコマンドでギャップセンサーをキャリブレーションします。この設定は電源を切る時にプリンターのメモリに保存されます。

構文

```
SET GAP n/AUTO/OFF/0,/REVERSE/OBVERSE
```

パラメーター	説明
N	ギャップセンサーの発光量（最低感度：0 ）を指定します。
AUTO	ギャップセンサーのキャリブレーションのため、2、3 枚ラベルを送ります。ラベルの切れ目が無い場合、プリンターは無定長ラベルかどうかを確認するために 254～508mm を限度にラベルを送ります。
OFF	**SET GAP AUTO** 機能を無効にします。
0,	自動的にギャップサイズを補正します。
REVERSE	ラベルの区切りにあるブラックマークがブラックマークセンサーで検知できない場合に使用します。ギャップセンサーを通過した媒体の一部を印刷可能領域あるいはギャップと定義します。
OBVERSE	**SET GAP REVERSE** 機能を無効にします。

プリンターモデル	ギャップセンサー範囲	ブラックマークセンサー範囲	SET GAP REVERSE SET GAP OBVERSE SET GAP AUTO
TD-4420TN / TD-4520TN	0~15	0~3	**V**
TD-4650TNWB TD-4750TNWB / TD-4750TNWBR	0~15	0~3	**V**
RJ-2035B / RJ-3035B	0~7	0~7	**V**
TJ-4021TN / TJ-4121TN / TJ-4121TNR	0~15	0~3	**V**
TJ-4420TN / TJ-4520TN / TJ-4620TN	0~15	0~3	**V**

備考：
SET HEAD OFF モードでは、印字ヘッドが開閉されても **SET GAP AUTO** 機能は動作しませんが、プリンターの電源がオンになると作動可能となります。

例

下の例はラベルサイズ、ギャップ距離とセンサー感度を設定するためにパラレルポート接続を通じて DOS 環境で動作しています。

```
C:\>COPY CON LPT1<ENTER>
SIZE 4,2.5<ENTER>
GAP 0.12,0<ENTER>
SET GAP 1<ENTER>
```

```
<CTRL><Z><ENTER>
C:\>
```

備考：
<ENTER>　はキーボードの **ENTER** キーを表します。上の例では<ENTER>　をキー入力するのではなく、**ENTER** キーを押してください。上の例では<CTRL>は **Ctrl**”キーを表します。

トラブルシューティング:
テストするために**給紙**キーを押します。エラー点滅せず、各ラベルが同じ位置で止まっていない場合、現在の設定より大きな数字を設定します。この設定を調整する時、まずは 0 から始めて、順に高い数値へ調整していきます。

関連コマンド

SIZE, GAP, BLINE

11.12　SET BLINE REVERSE / SET BLINE OBVERSE

説明

センサー機能を反転します。

構文

```
SET BLINE REVERSE/OBVERSE
```

パラメーター	説明
REVERSE	センサー機能を逆転します。反射領域は黒ラインで、非反射領域を用紙と再定義します。(通常、反射部分は用紙で、非反射部分は黒ラインです。)
OBVERSE	**SET BLINE REVERSE** 機能を無効にします。

11.13 SET BLINE PRINTSIDE / SET BLINE BACKSIDE

説明

上部のブラックマークセンサーまたは下部のブラックマークセンサーに設定します。

構文

```
SET BLINE PRINTSIDE / SET BLINE BACKSIDE
```

パラメーター	説明
PRINTSIDE	印刷面　：ブラックマークが印刷面にある場合
BACKSIDE	背面　：ブラックマークが裏面にある場合

備考：
TJ-4420TN、TJ-4520TN、TJ-4620TN のみ

11.14 SET HEAD

説明

ヘッドオープンセンサーを有効/無効にします。ヘッドオープンセンサーが無効の場合、オープン状態のプリンターヘッドはエラーメッセージを返しません。この設定はプリンターのメモリに保存されます。

構文

```
SET HEAD ON /OFF
```

パラメーター	説明
ON	"ヘッドオープン"センサーを有効にします
OFF	"ヘッドオープン"センサーを無効にします

例

SET HEAD ON
SET HEAD OFF

11.15 SET RIBBON

説明

リボンセンサー検知を有効/無効にします。熱転写／感熱プリンターは、プリンター起動時に、リボンの有無を検知して、感熱式・熱転式モードを判断します。この設定はプリンターのメモリ内に保存されません。

構文

```
SET RIBBON ON/OFF/INSIDE/OUTSIDE
```

パラメーター	説明
ON	熱転写式
OFF	感熱式

例

サンプルコード

```
REM *****Disable ribbon detection sensor for direct thermal printing.
SET RIBBON OFF
SIZE 4,1
GAP 0,0
CLS
TEXT 10,10, " 3 "  ,0,1,1, "  Direct thermal printing. "
PRINT 1

REM *****Enable ribbon detection sensor for thermal transfer printing.
SET RIBBON ON
SIZE 4,1
GAP 0,0
CLS
TEXT 10,10, " 3 "  ,0,1,1, "  Thermal transfer printing. "
PRINT 1
```

11.16 SET ENCODER

説明

リボンのエンコーダーセンサー検知を有効/無効にします。

構文

```
SET ENCODER ON/OFF
```

パラメーター	説明
ON	リボンエンコーダーセンサーを有効にします
OFF	リボンエンコーダーセンサーを無効にします

例

SET ENCODER ON
SET ENCODER OFF

11.17 SET RIBBONEND

説明

リボンエンドセンサー検知を有効/無効にします。

構文

```
SET RIBBONEND ON/OFF
```

パラメーター	説明
ON	リボンエンドセンサーを有効にします
OFF	リボンエンドセンサーを無効にします

例

SET RIBBONEND ON
SET RIBBONEND OFF

11.18 SET COM1

説明

プリンターシリアルポートの communication パラメーターを定義します。

構文

```
SET COM1 baud,parity,data,stop
```

パラメーター	説明
baud	ボー比率。利用可能なボー比率は下記のとおりです。 24: 2400 bps 48: 4800 bps 96: 9600 bps 19: 19200 bps 38: 38400 bps 57: 57600 bps 115: 115200 bps
parity	Parity check N: No parity check E: Even parity check O: Odd parity check
Data	Data bit 8: 8 bits data 7: 7 bits data
stop	Stop bit 1: 1 stop bit 2: 2 stop bits

例

パラレルポートは MS-DOS モードを通じてこの例の様にプリンターシリアルポートを設定するのに使用します。

```
C:\>COPY CON LPT1<ENTER>
SET COM1 19,N,8,1<ENTER>
<CTRL><Z><ENTER>
C:\>
```

備考：
<ENTER>は PC キーボードの **ENTER** キーを表します。<CTRL><Z>は PC キーボードの **CTRL** キーを押し、それから **Z** キーを押すことを意味しています。

11.19 SET PRINTKEY

説明

一つのラベルを印刷し、手切りのためにラベルギャップを手切り位置まで送り出します。次のラベルや一括ラベルを印刷するには、**給紙**ボタンを押します。ラベルの内容に連番テキストやバーコードが含まれている場合、連番処理されます。この設定はプリンターのメモリ内に保存されます。

構文

```
SET PRINTEKY OFF/ON/AUTO/<num>
```

パラメーター	説明
OFF	この機能を無効にします
ON	この機能を有効にします
AUTO	この機能を有効にします
<num>	**給紙**ボタンを押されたらラベル番号を印字します

例

サンプルコード

```
SIZE 4,2.5
GAP 0.12,0
SET PRINTKEY ON
SET COUNTER @0 1
@0= "0001"
CLS
TEXT 10,10, "5",0,1,1,@0
PRINT 1
```

Execute:

構文	Receive “PRINT m”	印刷
SET PRINTKEY ON or SET PRINTKEY AUTO	1.) PRINT 2	Label 1~2
	2.) 給紙キーを押す	Label 3~4

構文	Receive “PRINT m,n”	印刷
SET PRINTKEY ON or SET PRINTKEY AUTO	1.) PRINT 1,2	Label 1, Label 1
	2.) 給紙キーを押す	Label 2, Label 2

構文	Receive “PRINT -1,n”	印刷
SET PRINTKEY ON or SET PRINTKEY AUTO	1.) PRINT -1,2	Label 1, Label 1
	2.) 給紙キーを押す	Label 1, Label 1

構文	Receive “PRINT m”	印刷
SET PRINTKEY 5	1.) PRINT 2	Label 1~2
	2.) 給紙キーを押す	Label 3~7
構文	**Receive “PRINT m,n”**	**印刷**
SET PRINTKEY 5	1.) PRINT 1,2	Label 1, Label 1
	2.) 給紙キーを押す	Label 2~6

構文	Receive “PRINT –1,n”	印刷
SET PRINTKEY 5	1.) PRINT -1,2	Label 1, Label 1
	2.) 給紙キーを押す	Label 1, Label 1

11.20 SET REPRINT

説明

「用紙無し」、「リボン無し」、「キャリッジオープン」エラーの後に再印刷の試行を有効/無効にします。

構文

```
SET REPRINT OFF/ON
```

パラメーター	説明
OFF	この機能を無効にします
ON	この機能を有効にします

例

SET REPRINT ON

11.21 SET FEED_LEN

説明

給紙キーが押される時の送り出し長さを設定できます。この設定はプリンターにより記憶されます。初期値はラベルの長さです。

構文

```
SET FEED_LEN n
```

パラメーター	説明
n	送り出し長さ（ドット）

例

サンプルコード

```
SET FEED_LEN 100
```

結果

この設定後に**給紙**ボタンを押すと、送り出しの長さは 100 ドットになります。

11.22 GETSENSOR()

説明

センサーステータス/AD 値を取得します。センサー機能を確認するために使用できます。

構文

```
GETSENSOR(sensor$[,intension])
```

パラメーター / **説明**

`sensor$`

センサータイプ

GAP	ギャップセンサー
BLINE	ブラックマークセンサー
RIBBON	リボンエンドセンサー
PEEL	ハクリセンサー
HEAD UP	印字ヘッドオープンセンサー
HEAD TEMP	プリントヘッドの温度
HEAD VOLT	プリントヘッドの電圧
BATTERY VOLT	バッテリーの電圧
BATTERY CAP	バッテリーの容量

`intension`

Sensor intension

Gap	設定範囲は **SET GAP** コマンドをご参照ください
BLINE	設定範囲は **SET GAP** コマンドをご参照ください
RIBBON	0 ~ 3
PEEL sensor	無視されます
HEAD UP sensor	無視されます
HEAD TEMP	無視されます
HEAD VOLT	無視されます

`Returned value`

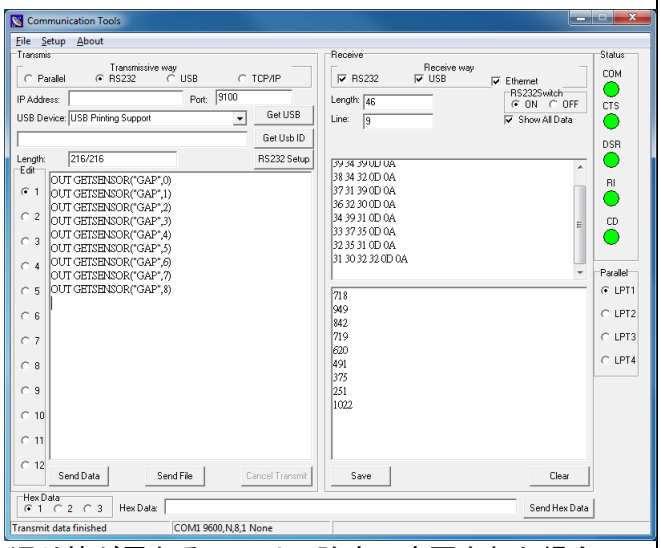
Gap	ギャップセンサーの AD 値を返します
BLINE	ブラックマークセンサーの AD 値を返します
RIBBON	リボンセンサーの AD 値を返します
PEEL	返り値 0: 用紙がセンサー上にありません 1: 用紙がセンサー上にあります
HEAD UP	返り値 0: 印字ヘッドが閉じています 1: 印字ヘッドが開いています
HEAD TEMP	印字ヘッドの温度を返します
HEAD VOLT	印字ヘッドの電圧を返します

例 (RS-232C 経由で Brother CommTool を使用してセンサーステータスを取得する例です)

サンプルコード

```
OUT GETSENSOR("GAP",0)
OUT GETSENSOR("GAP",1)
OUT GETSENSOR("GAP",2)
OUT GETSENSOR("GAP",3)
OUT GETSENSOR("GAP",4)
OUT GETSENSOR("GAP",5)
OUT GETSENSOR("GAP",6)
OUT GETSENSOR("GAP",7)
OUT GETSENSOR("GAP",8)
```

結果

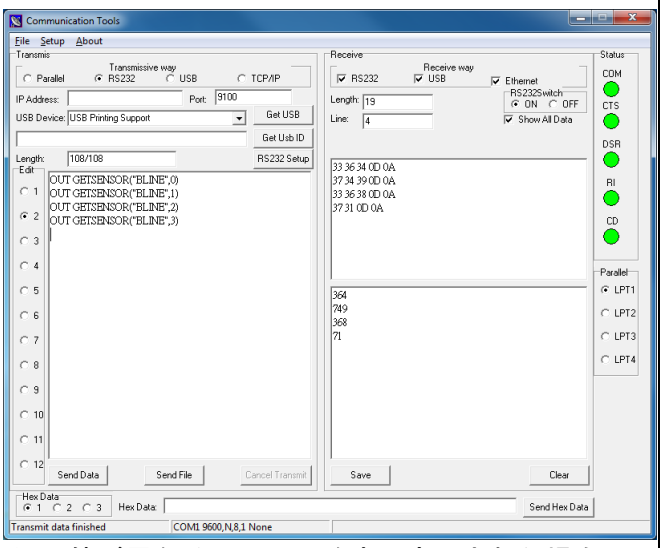


返り値が異なるセンサー強度に変更された場合、センサーは機能してます。

サンプルコード

```
OUT GETSENSOR("BLINE",0)
OUT GETSENSOR("BLINE",1)
OUT GETSENSOR("BLINE",2)
OUT GETSENSOR("BLINE",3)
```

結果

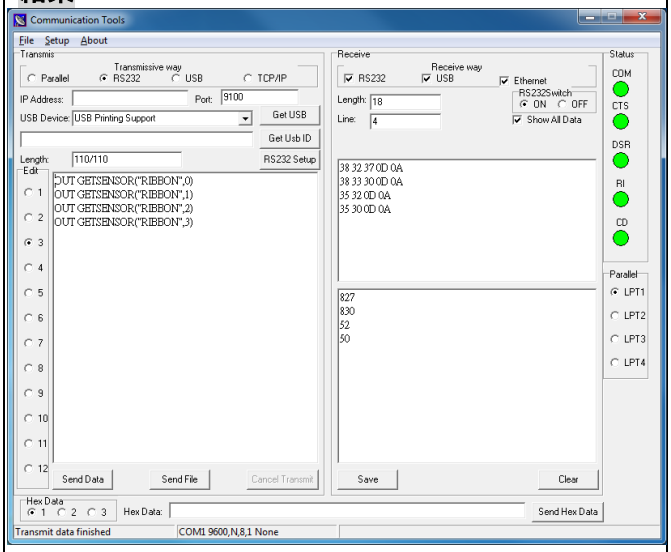


返り値が異なるセンサー強度に変更された場合、センサーは機能してます。

サンプルコード

```
OUT GETSENSOR("RIBBON",0)
OUT GETSENSOR("RIBBON",1)
OUT GETSENSOR("RIBBON",2)
OUT GETSENSOR("RIBBON",3)
```

結果

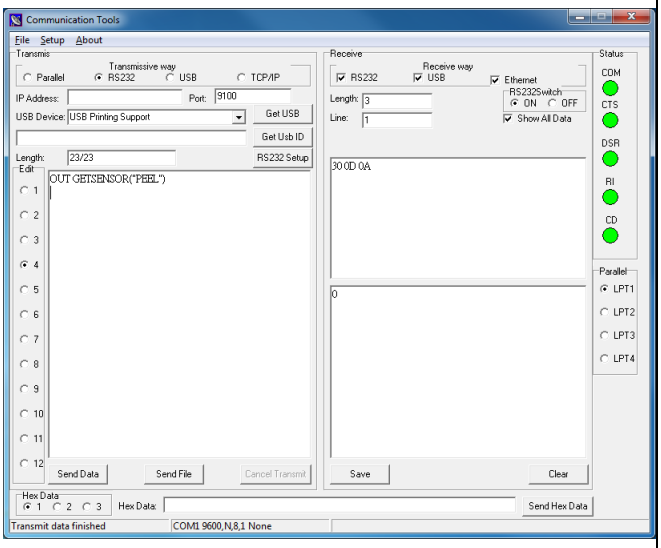


返り値が異なるセンサー強度に変更された場合、センサーは機能してます。

サンプルコード

```
OUT GETSENSOR("PEEL")
```

結果

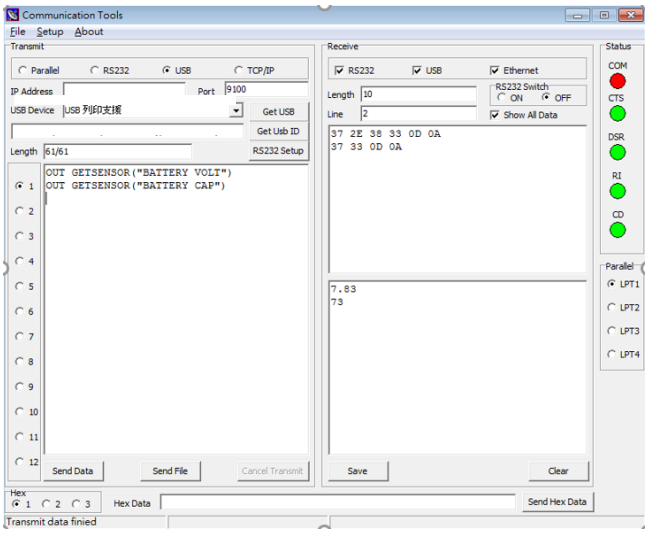


0: 用紙がセンサー上にありません
1: 用紙がセンサー上にあります

サンプルコード

```
OUT GETSENSOR("BATTERY VOLT")
OUT GETSENSOR("BATTERY CAP")
```

結果



RJ プリンターのバッテリーの電圧と容量を確認します

11.23 GETSETTING$()

説明

プリンター設定を取得します。

構文

```
GETSETTING$(app$,sec$,key$[,default$])
```

app$	sec$	key$	説明
SYSTEM	INFORMATION	DPI	解像度
		MODEL	モデル名
		SERIAL	シリアル番号
		VERSION	ファームウェアバージョン
		CHECKSUM	ファームウェアチェックサム
		PRINTQUALITY	印字モード（DRAFT, STANDARD または OPTIMUM : **SET PRINTQUALITY** 参照）RJ-2035B/3035B のみ
		STANDBYTIME	待機時間（OFF または数字）RJ-2035B/3035B のみ
		SLEEPTIME	スリープ時間（OFF または数字）RJ-2035B/3035B のみ
	RECORD	MILAGE	印字マイレージ（ドット）
		LABEL COUNTER	印字総枚数
		CUT COUNTER	カット数
FILE	DRAM	CAPACITY	DRAM の合計容量
		AVAILABLE	DRAM の利用可能容量
	FLASH	CAPACITY	FLASH の合計容量
		AVAILABLE	FLASH の利用可能容量
	CARD	CAPACITY	CARD の合計容量
		AVAILABLE	CARD の利用可能容量
		INSTALLED	カードステータス（1:あり： 0: なし）
CONFIG	NET	MAC ADDRESS	MAC アドレス
		IP ADDRESS	IP アドレス
		SUBNET MASK	サブネットマスク
		DEFAULT GATEWAY	デフォルトゲートウェイ
		RAW PORT	Raw（標準）ポート。
		NAME	プリンター名
		PRIMARY DNS	プライマリーDNS
		SECONDARY DNS	セカンダリーDNS
	WLAN	MAC ADDRESS	MAC アドレス
		IP ADDRESS	IP アドレス
		SUBNET MASK	サブネットマスク
		DEFAULT GATEWAY	デフォルトゲートウェイ
		RAW PORT	Raw（標準）ポート
	COM1	BAUD RATE	COM ポートのボー比率
		DATA BIT	COM ポートのデータビット
		PARITY	COM ポートのパリティ（伝送エラー検出法）
		STOP BIT	COM ポートのストップビット
	SENSOR	SENSOR TYPE	現在のセンサータイプ
		CARRIAGE	ヘッドオープンセンサーのステータス
		GAP INTENSION	ギャップセンサーの強度
		BLINE INTENSION	ブラックマークセンサーの強度
		CONTINUOUS INTENSION	継続センサーの強度
	FBPL	PRINT MODE	印字後動作のモード
		DENSITY	印刷濃度
		PAPER SIZE	用紙サイズ
		GAP SIZE	ギャップサイズ
		BLINE SIZE	ブラックマークサイズ
		DIRECTION	印刷方向

		MIRROR	ミラーステータス
		RIBBON	リボンステータス
		REPRINT	再印刷ステータス
		PAPER WIDTH	用紙幅
		LIMIT FEED	センサーキャリブレーションの最大長
		OFFSET	オフセット値
		REFERENCE X	リファレンス X 値
		REFERENCE Y	リファレンス Y 値
		SHIFT X	シフト X 値
		SHIFT Y	シフト Y 値
		SPEED	印字スピード
		COUNTRY CODE	国コード
		CODEPAGE	コードページ

パラメーター	説明
default$	オプション。key$設定で値が何も設定されていな場合、式は返信する値を含みます。省略された場合、初期値は長さがゼロの文字列(“”)と仮定されます。

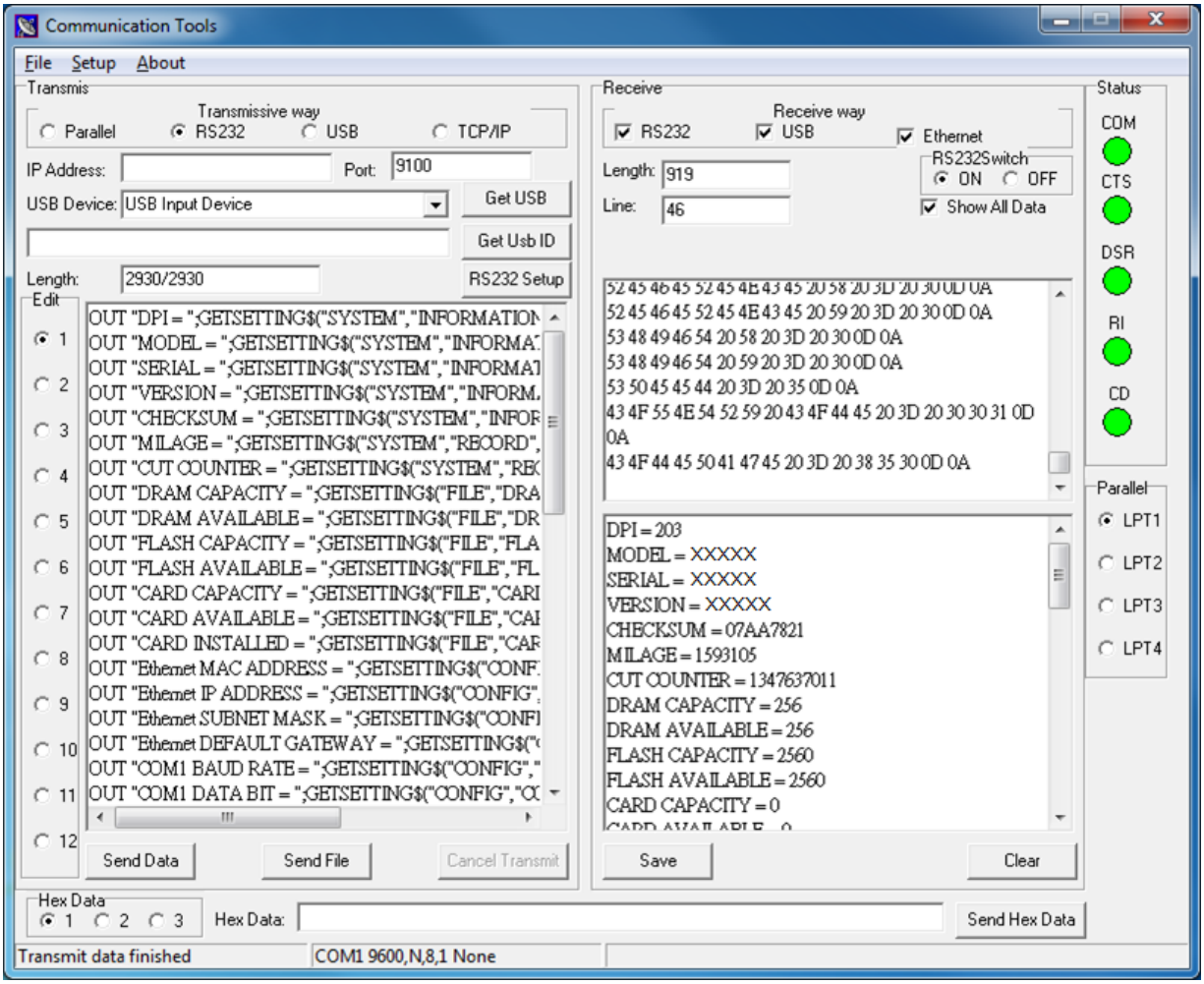
例

サンプルコード (RS-232C 経由で Brother CommTool を使用してプリンター設定を取得する例です)

```
OUT "DPI = ";GETSETTING$("SYSTEM","INFORMATION","DPI")
OUT " MODEL =";GETSETTING$("SYSTEM","INFORMATION","MODEL")
OUT "SERIAL = ";GETSETTING$("SYSTEM","INFORMATION","SERIAL")
OUT "VERSION = ";GETSETTING$("SYSTEM","INFORMATION","VERSION")
OUT "CHECKSUM = ";GETSETTING$("SYSTEM","INFORMATION","CHECKSUM")
OUT "MILAGE = ";GETSETTING$("SYSTEM","RECORD","MILAGE")
OUT "CUT COUNTER = ";GETSETTING$("SYSTEM","RECORD","CUT COUNTER")
OUT "DRAM CAPACITY = ";GETSETTING$("FILE","DRAM","CAPACITY")
OUT "DRAM AVAILABLE = ";GETSETTING$("FILE","DRAM","AVAILABLE")
OUT "FLASH CAPACITY = ";GETSETTING$("FILE","FLASH","CAPACITY")
OUT "FLASH AVAILABLE = ";GETSETTING$("FILE","FLASH","AVAILABLE")
OUT "CARD CAPACITY = ";GETSETTING$("FILE","CARD","CAPACITY")
OUT "CARD AVAILABLE = ";GETSETTING$("FILE","CARD","AVAILABLE")
OUT "CARD INSTALLED = ";GETSETTING$("FILE","CARD","INSTALLED")
OUT "Ethernet MAC ADDRESS = ";GETSETTING$("CONFIG","NET","MAC ADDRESS")
OUT "Ethernet IP ADDRESS = ";GETSETTING$("CONFIG","NET","IP ADDRESS")
OUT "Ethernet SUBNET MASK = ";GETSETTING$("CONFIG","NET","SUBNET MASK")
OUT "Ethernet DEFAULT GATEWAY = ";GETSETTING$("CONFIG","NET","DEFAULT GATEWAY")
OUT "Ethernet PRIMARY DNS = ";GETSETTING$("CONFIG","NET","PRIMARY DNS")
OUT "Ethernet SECONDARY DNS = ";GETSETTING$("CONFIG","NET","SECONDARY DNS")
OUT "COM1 BAUD RATE = ";GETSETTING$("CONFIG","COM1","BAUD RATE")
OUT "COM1 DATA BIT = ";GETSETTING$("CONFIG","COM1","DATA BIT")
OUT "COM1 PARITY = ";GETSETTING$("CONFIG","COM1","PARITY")
OUT "COM1 STOP BIT = ";GETSETTING$("CONFIG","COM1","STOP BIT")
OUT "SENSOR TYPE = ";GETSETTING$("CONFIG","SENSOR","SENSOR TYPE")
OUT "CARRIAGE = ";GETSETTING$("CONFIG","SENSOR","CARRIAGE")
OUT "GAP INTENSION = ";GETSETTING$("CONFIG","SENSOR","GAP INTENSION")
OUT "BLINE INTENSION = ";GETSETTING$("CONFIG","SENSOR","BLINE INTENSION")
OUT "CONTINUOUS INTENSION = ";GETSETTING$("CONFIG","SENSOR","CONTINUOUS
INTENSION")
OUT "PRINT MODE = ";GETSETTING$("CONFIG","FBPL","PRINT MODE")
OUT "DENSITY = ";GETSETTING$("CONFIG","FBPL","DENSITY")
OUT "PAPER SIZE = ";GETSETTING$("CONFIG","FBPL","PAPER SIZE")
OUT "GAP SIZE = ";GETSETTING$("CONFIG","FBPL","GAP SIZE")
OUT "BLINE SIZE = ";GETSETTING$("CONFIG","FBPL","BLINE SIZE")
OUT "DIRECTION = ";GETSETTING$("CONFIG","FBPL","DIRECTION")
OUT "MIRROR = ";GETSETTING$("CONFIG","FBPL","MIRROR")
OUT "RIBBON = ";GETSETTING$("CONFIG","FBPL","RIBBON")
```

```
OUT "REPRINT = ";GETSETTING$("CONFIG","FBPL","REPRINT")
OUT "PAPER WIDTH = ";GETSETTING$("CONFIG","FBPL","PAPER WIDTH")
OUT "LIMIT FEED = ";GETSETTING$("CONFIG","FBPL","LIMIT FEED")
OUT "OFFSET = ";GETSETTING$("CONFIG","FBPL","OFFSET")
OUT "REFERENCE X = ";GETSETTING$("CONFIG","FBPL","REFERENCE X")
OUT "REFERENCE Y = ";GETSETTING$("CONFIG","FBPL","REFERENCE Y")
OUT "SHIFT X = ";GETSETTING$("CONFIG","FBPL","SHIFT X")
OUT "SHIFT Y = ";GETSETTING$("CONFIG","FBPL","SHIFT Y")
OUT "SPEED = ";GETSETTING$("CONFIG","FBPL","SPEED")
OUT "COUNTRY CODE = ";GETSETTING$("CONFIG","FBPL","COUNTRY CODE")
OUT "CODEPAGE = ";GETSETTING$("CONFIG","FBPL","CODEPAGE")
```

結果



サンプルコード(NET, WLAN)

```
OUT "Ethernet DEFAULT RAW PORT = ";GETSETTING$("CONFIG","NET","RAW PORT")


OUT "WLAN MAC ADDRESS = ";GETSETTING$("CONFIG","WLAN","MAC ADDRESS")
OUT "WLAN IP ADDRESS = ";GETSETTING$("CONFIG","WLAN","IP ADDRESS")
OUT "WLAN SUBNET MASK = ";GETSETTING$("CONFIG","WLAN","SUBNET MASK")
OUT "WLAN DEFAULT GATEWAY = ";GETSETTING$("CONFIG","WLAN","DEFAULT GATEWAY")
OUT "WLAN DEFAULT RAW PORT = ";GETSETTING$("CONFIG","WLAN","RAW PORT")
OUT "NET Name = ";GETSETTING$("CONFIG","NET","NAME")
```

11.24 SET USBHOST

説明

LCD 搭載モデル向けのコマンドです。**INPUT** コマンドと **SET FILTER** コマンドを USB ホスト機器とあわせて使用する場合は、USB キーボードもしくは USB バーコードリーダーいずれかのモードを設定する必要があります。

構文

```
SET USBHOST KEYBOARD/SCANNER
```

パラメーター	説明
KEYBOARD	USB キーボード （デフォルト：プロンプトを LCD 上に表示します）
SCANNER	USB バーコードリーダー （プロンプトを LCD 上に表示しません）

備考：

- USB ホストのある LCD 搭載モデルのみ
- モードが使用する USB ホスト機器と一致していない場合は、USB ホスト機器からの入力が正しくプリンターに送信されない可能性があります。

例

サンプルコード

```
SET USBHOST KEYBOARD
DOWNLOAD "A.BAS"
:LOOP
SIZE 4,2
GAP 0,0
CLS
INPUT A$
TEXT 50,50,"0",0,20,20,A$
PRINT 1
GOTO LOOP
EOP
A.BAS
```

関連コマンド

INPUT, SET FILTER

11.25 SET AUTORUN

説明

プリンターを切り替えると同時に、自動的に動作する BAS ファイルを再定義します。初期値は AUTO.BAS です。

構文

```
SET AUTORUN "filename"
```

パラメーター	説明
filename	ファイルは AUTO-RUN ファイルと定義されます（初期値：AUTO.BAS）

例

サンプルコード

```
REM *****Step1: Send the following command to redefine the auto-run file from
"AUTO.BAS" to "TEST.BAS"
SET AUTORUN "TEST.BAS"

REM *****Step2: Send the following commands to download "TEST.BAS" file into
printer.
DOWNLOAD F, "TEST.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
BLOCK 10,10,600,200, "3",0,1,1,12, "TEST.BAS is running automatically while
turning on the printer. "
PRINT 1
EOP

REM *****Step3: Turn off and on the printer to run "TEST.BAS" automatically.
```

結果

```
"TEST.BAS" is running automatically
while turning on the printer.
```

11.26 SET RESPONSE

説明

指定タイミングで自動的にエラー内容を発信します。

構文

```
SET RESPONSE ["Job ID",] ON/OFF/BATCH
```

パラメーター	説明
["Job ID"]	オプション：Set job ID（初期値：Null）
ON	この機能を有効にします
OFF	この機能を無効にします（初期値：オフ）
BATCH	印刷ジョブの最後に応答します

レスポンス構文

```
{Status,######,ID}
```

ステータス

[16 進数]

00 通常
01 ヘッドオープン
02 用紙詰まり
03 用紙詰まりとヘッドオープン
04 用紙切れ
05 用紙切れとヘッドオープン
08 インクリボン切れ
09 インクリボン切れとヘッドオープン
0A インクリボン切れと用紙詰まり
0B インクリボン切れ、用紙詰まり、ヘッドオープン
0C インクリボン切れと用紙切れ
0D インクリボン切れ、用紙切れ、ヘッドオープン
10 停止
20 印字中
80 その他のエラー

#####: 00001 ~ 99999

例

サンプルコード

```
SET RESPONSE ON
SIZE 4,2
GAP 0,0
PRINT 3

{00,00001}{00,00002}{00,00003}
```

サンプルコード

```
SET RESPONSE "ID1",ON
SIZE 4,2
GAP 0,0
PRINT 3,2

{00,00001,ID1}{00,00002,ID1}{00,00003,ID1}{00,00004,ID1}{00,00005,ID1}{00
,00006,ID1}
```

サンプルコード

```
SET RESPONSE "CCCC ",BATCH
SIZE 4,2
GAP 0,0
PRINT 3,2

{00,00006,CCCC}
```

11.27 SET DAYLIGHT_SAVE

説明

サマータイム時間を設定するのに使われます。

構文

```
SET DAYLIGHT_SAVE ON/OFF
SET DAYLIGHT_SAVE "Start", "End"
```

パラメーター	説明
ON	有効にします
OFF	無効にします（初期値）
"Start"	"Start time"から 1 時間増やします
"End"	"End time"から 1 時間減らします

Month
"JAN","FEB","MAR","APR","MAY","JUN",
"JUL","AUG","SEP","OCT","NOV","DEC"

"JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE", "JULY",
"AUGUST", "SEPTEMBER", "OCTOBER", "NOVEMBER", "DECEMBER"

Week
"SUN","MON","TUE","WED","THU","FRI","SAT"
"SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY",
"FRIDAY", "SATURDAY"

Which Week
"FIRST","SECOND", "THIRD","FOURTH","LAST"
"1ST", "2ND", "3RD", "4TH", "LAST"

Date
1~31

Time
0:00~23:00

例

サンプルコード

```
SET DAYLIGHT_SAVE ON
SET DAYLIGHT_SAVE OFF
SET DAYLIGHT_SAVE "MAR 1 4:00","NOV 1 5:00"
SET DAYLIGHT_SAVE "MAR FIRST SUN 2:00", "NOV LAST SUN 3:00"
```

11.28 SET REGISTRATION

説明

１ロールに複数のラベルサイズがある場合のラベル位置モードを設定します。

構文

```
SET REGISTRATION mode
```

パラメーター	説明
mode	BYSIZE: デフォルト ACTUAL: １ロールに複数のラベルサイズがある場合

例

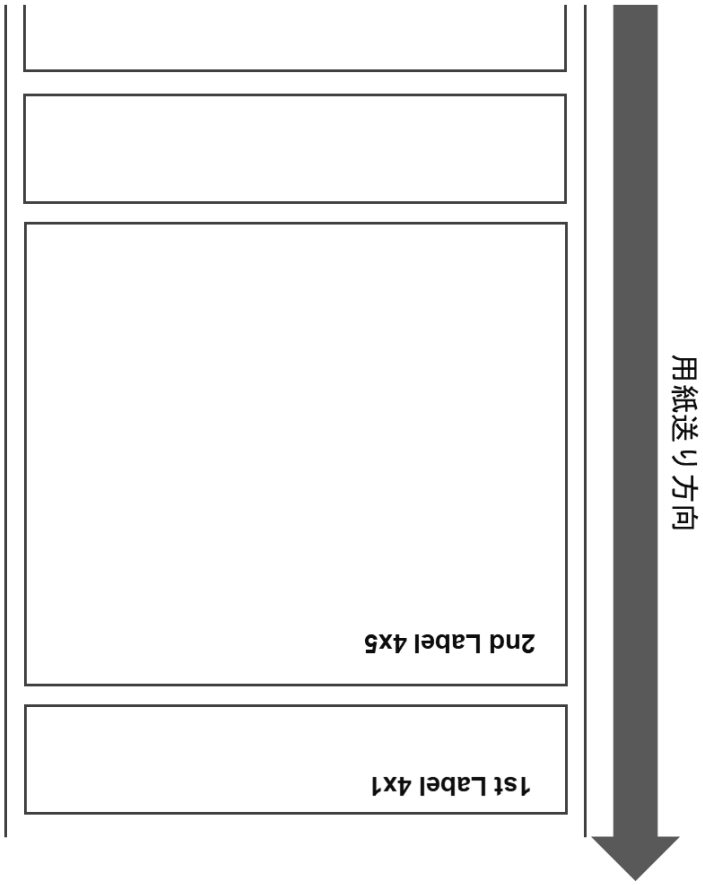
サンプルコード

ラベル2枚それぞれに異なる印刷ジョブ:

```
SET REGISTRATION ACTUAL
SIZE 4,1
GAP 0.12,0
CLS

TEXT 40,40,"0",0,10,10,"1st
Label 4x1" PRINT 1
SIZE 4,5
GAP 0.12,0
CLS
TEXT 40,40,"0",0,10,10,"2nd
Label 4x5" PRINT 1
```

ラベル2枚に同じ印刷ジョブ：

```
SET REGISTRATION ACTUAL
SIZE 4,6.12
GAP 0.12,0
CLS
TEXT 40,40,"0",0,10,10,"1st
Label 4x1" TEXT
40,267,"0",0,10,10,"2nd
Label 4x5"
PRINT 1
```



11.29 PEEL

説明

ハクリセンサーのステータスを取得します。読み取りのみです。

構文

```
PEEL
```

Return Value	説明
0	用紙がハクリセンサーの上部にありません。
1	用紙がハクリセンサーの上部にあります。

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
SET PEEL OFF
SET KEY1 OFF
SET LED1 OFF
SET LED3 OFF
:START
LED1=0
LED3=0
IF KEY1=1 THEN GOTO A
GOTO START
:A
LED1=1
CLS
TEXT 10,10, "3",0,1,1, "PEEL Function Test!! "
PRINT 1,1
:B
LED1=0
IF PEEL=1 THEN
LED3=1
GOTO B
ELSE
CLS
TEXT 10,10, "3",0,1,1, "The label is removed from the PEEL sensor!! "
PRINT 1,1
GOTO START
ENDIF
EOP
DEMO
```

11.30 LED1, LED2, LED3

説明

このコマンドは LED の有効/無効にします。書き込みのみです。LED のライトを有効にするには 1 を指定し、無効にするには 0 を指定します。このコマンドを使用する前に、LED 機能の初期値をキャンセルします。**SET LED** コマンドをご参照ください。

構文

```
LEDm = n
```

Return Value	説明
m	m=1, LED1 m=2, LED2 m=3, LED3
n	0: LED を無効にします 1: LED を有効にします

モデル	LED1	LED2	LED3	LED4	LED5	LED6	LED7	LED2 & LED3
TD-4T シリーズ **備考**：このシリーズは LED1=LED2	緑色	緑色	赤色					黄色
RJ シリーズ	緑色	緑色	赤色	バッテリ 1	バッテリ 2	バッテリ 3	WIFI/BT	黄色
TJ シリーズ	緑色	緑色	赤色					黄色

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 3,3
GAP 0.12,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
SET CUTTER OFF
SET PEEL OFF
SET LED1 OFF
SET LED2 OFF
SET LED3 OFF
LED1=0
LED2=1
LED3=0
EOP
DEMO
```

11.31 KEY1, KEY2, KEY3

説明

KEY1 と KEY2 と KEY3 のステータスを読み取ります。

機種	KEY0	KEY1	KEY2	KEY3	KEY4	KEY5	KEY6
TD-4T シリーズ		給紙					
RJ シリーズ		給紙					
TJ シリーズ		給紙	メニュー	上	右	左	下

構文

```
KEYm = n
```

Key	返り値
KEY1 (メニュー)	0: 離す
	1: 押す
KEY2 (停止)	0: 離す
	1: 押す
KEY3 (給紙)	0: 離す
	1: 押す

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 3,1
GAP 0,0
SPEED 4
DENSITY 8
DIRECTION 1
REFERENCE 0,0
SET LED1 OFF
SET KEY1 OFF
LED1=0
:START
IF KEY1=1 THEN
LED1=1
CLS
TEXT 100,10, "3",0,1,1, "KEY FUNCTION TEST"
PRINT 1,1
ELSE
LED1=0
ENDIF
GOTO START
EOP
DEMO
```

11.32 SET SENSOR_REF

説明

センサーのしきい値を設定します。

構文

```
SET SENSOR_REF AUTO
SET SENSOR_REF MANUAL
```

Key	Return Value
AUTO	用紙を送るとき、センサー値（High/Low）に応じて用紙のしきい値を自動的に調整します（初期値）
MANUAL	用紙を送るとき、センサー値（High/Low）に応じて用紙のしきい値を自動的に調整せず、しきい値を固定します

例

サンプルコード

```
SET SENSOR_REF AUTO

SET SENSOR_REF MANUAL
```

12 プリンターのグローバル変数

12.1@LABEL

説明

この変数は何枚のラベルが印刷されたかをカウントします。プリンターがリセットされると初期化されますが、プリンターの電源をオフにしても保持されます。

構文

書き込み: `@LABEL=n or @LABEL= "n"`
読み取り: `A=LABEL or A$=STR$(LABEL)`

パラメーター	説明
n	印刷されたラベル枚数 0<=n<=999999999

例

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,2.5
GAP 0,0
DIRECTION 1
CLS
TEXT 10,50, "3",0,1,1,@LABEL
TEXT 10,100, "3",0,1,1,
"@LABEL="+STR$(LABEL)
TEXT 10,150, "3",0,1,1,
"*****Statement 1*****"
IF LABEL>1000 THEN
TEXT 10,200, "3",0,1,1, "LABEL>1000"
ELSE
TEXT 10,200, "3",0,1,1, "LABEL<1000"
ENDIF
TEXT 10,250, "3",0,1,1,
"*****Statement 1*****"
A=LABEL
IF A>1000 THEN
TEXT 10,300, "3",0,1,1, "A>1000"
ELSE
TEXT 10,300, "3",0,1,1, "A<1000"
ENDIF
TEXT 10,350, "3",0,1,1,
"*****Statement 3*****"
A$=STR$(LABEL)
IF VAL(A$)>1000 THEN
TEXT 10,400, "3",0,1,1, "VAL(A$)>1000"
ELSE
TEXT 10,400, "3",0,1,1, "VAL(A$)<1000"
ENDIF
PRINT 1,1
EOP
DEMO
```

結果

```
1661
@LABEL=1661
*****Statement 1*****
LABEL>1000
*****Statement 1*****
A>1000
*****Statement 3*****
VAL(A$)>1000
```

12.2YEAR

説明

この変数は Real Time Clock (RTC)を通じて年のデータを読み取り/書き込みます。RTC は 4 桁の年表示に対応しています。

構文

書き込み：`YEAR = 02`
読み取り：`A = YEAR`
範囲：`00~50` = 2000~2050; `51~99` = 1951~1999

例

サンプルコード

```
DOWNLOAD "SetYear.BAS"
REM *****Set Year Parameter to RTC*****
YEAR=13
EOP
SetYear
```

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read YEAR Parameter from RTC*****
YEAR$=STR$(YEAR)
Y=YEAR

REM *****Print*****
TEXT 10,10, "3",0,1,1, "YEAR1="+YEAR$
TEXT 10,50, "3",0,1,1, "YEAR2="+STR$(Y)
TEXT 10,90, "3",0,1,1, "YEAR3="+STR$(YEAR)
PRINT 1
EOP
DEMO
```

結果

```
YEAR1=2013
YEAR2=2013
YEAR3=2013
```

関連コマンド

~!C, MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.3MONTH

説明

この変数は Real Time Clock (RTC)を通じて月のデータを読み取り/書き込みます。RTC は 2 桁（01-12）の月表示に対応しています。

構文

書き込み：MONTH = 01
読み取り：A = MONTH
範囲：01~12

例

サンプルコード

```
DOWNLOAD "SetMonth.BAS"
REM *****Set Month Parameter to RTC*****
MONTH=01
EOP
SetMonth
```

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Month Parameter form RTC*****
MONTH$=STR$(MONTH)
M=MONTH

REM *****Print*****
TEXT 10,10, "3",0,1,1, "MONTH1="+MONTH$
TEXT 10,50, "3",0,1,1, "MONTH2="+STR$(M)
TEXT 10,90, "3",0,1,1, "MONTH3="+STR$(MONTH)
PRINT 1
EOP
DEMO
```

結果

```
MONTH1=1
MONTH2=1
MONTH3=1
```

関連コマンド

MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.4DATE

説明

この変数は Real Time Clock (RTC)を通じて日付のデータを読み取り/書き込みます。RTC は 2 桁（01-31）の日付表示に対応しています。

構文

書き込み：DATE = 12
読み取り：A = DATE
範囲： 01~31

例

サンプルコード

```
DOWNLOAD "SetDate.BAS"
REM *****Set Date Paramter to RTC*****
DATE=10
EOP
SetDate
```

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Date Parameter form RTC*****
DATE$=STR$(DATE)
D=DATE

REM *****Print*****
TEXT 10,10, "3",0,1,1, "DATE1="+DATE$
TEXT 10,50, "3",0,1,1, "DATE2="+STR$(D)
TEXT 10,90, "3",0,1,1, "DATE3="+STR$(DATE)
PRINT 1
EOP
DEMO
```

結果

```
DATE1=10
DATE2=10
DATE3=10
```

関連コマンド

MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.5WEEK

説明

この変数は Real Time Clock (RTC)を通じて曜日のデータを読み取り/書き込みし、1 桁（1~7）で表現されます。

構文

読み取り：A = WEEK
範囲：1(日曜日)~7(土曜日)

例

サンプルコード

```
DOWNLOAD "DEMO.BAS "
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Week Parameter form RTC*****
WEEK$=STR$(WEEK)
W=WEEK

REM *****Print*****
TEXT 10,10, "3",0,1,1, "WEEK1="+WEEK$
TEXT 10,50, "3",0,1,1, "WEEK2="+STR$(W)
TEXT 10,90, "3",0,1,1, "WEEK3="+STR$(WEEK)
PRINT 1
EOP
DEMO
```

結果

```
WEEK1=5
WEEK2=5
WEEK3=5
```

関連コマンド

MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.6 HOUR

説明

この変数は Real Time Clock (RTC)を通じて時間のデータを読み取り/書き込みます。RTC は 24 時間表示（00-23）に対応しています。

構文

書き込み：HOUR = 12
読み取り：A = HOUR
範囲： 00~23

例

サンプルコード

```
DOWNLOAD "SetHour.BAS"
REM *****Set Hour Parameter to RTC*****
HOUR=10
EOP
SetHour
```

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Hour Parameter form RTC*****
HOUR$=STR$(HOUR)
H=HOUR

REM *****Print*****
TEXT 10,10, "3",0,1,1, "HOUR1="+HOUR$
TEXT 10,50, "3",0,1,1, "HOUR2="+STR$(H)
TEXT 10,90, "3",0,1,1, "HOUR3="+STR$(HOUR)
PRINT 1
EOP
DEMO
```

結果

```
HOUR1=10
HOUR2=10
HOUR3=10
```

関連コマンド

MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.7 MINUTE

説明

この変数は Real Time Clock (RTC)を通じて年のデータを読み取り/書き込みします。RTC は 2 桁（00~59）の分表示に対応しています。

構文

書き込み：MINUTE = 12
読み取り：A = MINUTE
範囲：00~59

例

サンプルコード

```
DOWNLOAD "SetMinute.BAS"
REM *****Set Minute Parameter to RTC*****
MINUTE=27
EOP
SetMinute
```

サンプルコード

```
DOWNLOAD "DEMO.BAS"
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Minute Parameter form RTC*****
MINUTE$=STR$(MINUTE)
MIN=MINUTE

REM *****Print*****
TEXT 10,10, "3",0,1,1, "MINUTE1="+MINUTE$
TEXT 10,50, "3",0,1,1, "MINUTE2="+STR$(MIN)
TEXT 10,90, "3",0,1,1, "MINUTE3="+STR$(MINUTE)
PRINT 1
EOP
DEMO
```

結果

```
MINUTE1=27
MINUTE2=27
MINUTE3=27
```

関連コマンド

MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.8 SECOND

説明

この変数は Real Time Clock (RTC)を通じて秒のデータを読み取り/書き込みます。RTC は 2 桁（00~59）の秒表示に対応しています。

構文

書き込み：SECOND = 12
読み取り：A = SECOND
範囲：00~59

例

サンプルコード

```
DOWNLOAD "SetSecond.BAS"
REM *****Set Second Parameter to RTC*****
SECOND=59
EOP
SetSecond
```

サンプルコード

```
DOWNLOAD "DEMO.BAS "
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS

REM *****Read Second Parameter form RTC*****
SECOND$=STR$(SECOND)
SEC=SECOND

REM *****Print*****
TEXT 10,10, "3",0,1,1, "SECOND1="+SECOND$
TEXT 10,50, "3",0,1,1, "SECOND2="+STR$(SEC)
TEXT 10,90, "3",0,1,1, "SECOND3="+STR$(SECOND)
PRINT 1
EOP
DEMO
```

結果

```
SECOND1=59
SECOND2=59
SECOND3=59
```

関連コマンド

MONTH, DATE, DAY, HOUR, MINUTE, SECOND

12.9@YEAR

説明

この変数は Real Time Clock (RTC)を通じて年のデータを読み取り/書き込みます。RTC は 2 桁の年表示に対応しています。**@YEAR** グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

書き込み：`@YEAR = "01"`
読み取り：`@YEAR`
範囲：00~99

例

サンプルコード	結果

`REM *****Set @YEAR*****`
`@YEAR="05"`
 	
`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1, "@YEAR"`
`TEXT 210,10, "3",0,1,1, @YEAR`
`PRINT 1`
	@YEAR 2005

関連コマンド

@MONTH, @DATE, @DAY, @HOUR, @MINUTE, @SECOND

12.10 @MONTH

説明

この変数は Real Time Clock (RTC)を通じて月のデータを読み取り/書き込みます。RTC は 2 桁（01~12）の月表示に対応しています。**@MONTH** グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

書き込み：`@MONTH = "01"`
読み取り：`@MONTH`
範囲：01~12

例

サンプルコード	結果

`REM *****Set @MONTH*****`
`@MONTH="12"`
 	
`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1, "@MONTH"`
`TEXT 210,10, "3",0,1,1,@MONTH`
`PRINT 1`
	@MONTH 12

関連コマンド

@YEAR, @DATE, @DAY, @HOUR, @MINUTE, @SECOND

12.11　@DATE

説明

この変数は Real Time Clock (RTC)を通じて日付のデータを読み取り/書き込みます。RTC は 2 桁（01~31）の日付形式に対応しています。**@DATE** グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

書き込み：`@DATE = "12"`
読み取り：`@DATE`
範囲：01~31

例

サンプルコード	結果

`REM *****Set @DATE*****`
`@DATE="31"`
 	
`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1, "@DATE"`
`TEXT 210,10, "3",0,1,1,@DATE`
`PRINT 1`
	@DATE　　31

関連コマンド

@YEAR, @MONTH, @DAY, @HOUR, @MINUTE, @SECOND

12.12　@DAY

説明

この変数は Real Time Clock (RTC)を通じて曜日のデータを読み取り/書き込みます。１桁（1~7）で表示されます。@DAY グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

読み取り：@DAY

範囲： 1(日曜日)~7(土曜日)

例

サンプルコード	結果

`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1, "@DAY"`
`TEXT 210,10, "3",0,1,1,@DAY`
`PRINT 1`
	@DAY　　7

関連コマンド

@YEAR, @MONTH, @DATE, @HOUR, @MINUTE, @SECOND

12.13　@HOUR

説明

この変数は Real Time Clock (RTC)を通じて時間のデータを読み取り/書き込みます。RTC は 24 時間表示（00~23）に対応しています。**@HOUR** グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

書き込み：`@HOUR = "12"`
読み取り：`@HOUR`
範囲：00~23

例

サンプルコード	結果

`REM *****Set @HOUR*****`
`@HOUR="23"`
 	
`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1,`
`"@HOUR"`
`TEXT 210,10, "3",0,1,1,@HOUR`
`PRINT 1`
	@HOUR　　23

関連コマンド

@YEAR, @MONTH, @DATE, @DAY, @MINUTE, @SECOND

12.14 @MINUTE

説明

この変数は Real Time Clock (RTC)を通じて分のデータを読み取り/書き込みます。RTC は 2 桁（00~59）の分表示に対応しています。**@MINUTE** グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

書き込み：`@MINUTE = "12"`
読み取り：`@MINUTE`
Range: `00~59`

例

サンプルコード	結果

`REM *****Set @MINUTE*****`
`@MINUTE="59"`
 	
`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1, "@MINUTE"`
`TEXT 210,10, "3",0,1,1,@MINUTE`
`PRINT 1`
	@MINUTE 59

関連コマンド

@YEAR, @MONTH, @DATE, @DAY, @HOUR, @SECOND

12.15　@SECOND

説明

この変数は Real Time Clock (RTC)を通じて秒のデータを読み取り/書き込みます。RTC は 2 桁（00~59）の秒表示に対応しています。**@SECOND** グローバル変数は BASIC 言語コマンドを使用することなく、直接アクセス可能です。

構文

書き込み：`@SECOND = "12"`
読み取り：`@SECOND`
範囲：00~59

例

サンプルコード	結果

`REM *****Set @SECOND*****`
`@SECOND = "59"`
 	
`REM *****Print*****`
`SIZE 4,1`
`GAP 0,0`
`DIRECTION 1`
`CLS`
`TEXT 10,10, "3",0,1,1,`
`"@SECOND"`
`TEXT 210,10, "3",0,1,1,@SECOND`
`PRINT 1`
	@SECOND　59

関連コマンド

@YEAR, @MONTH, @DATE, @DAY, @HOUR, @MINUTE

12.16 _MODEL$

説明

この変数は読み取りのみです。プリンターモジュール名の情報を含みます。

構文

```
_MODEL$
```

例

サンプルコード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

結果

Model:******

Serial No.:**********

F/W Version:*.**

関連コマンド

_SERIAL$, _VERSION$

12.17 _SERIAL$

説明

この変数は読み取りのみです。プリンターシリアル番号の情報を含みます。
プリンターシリアル番号は工場にてプリンタ本体に設定されます。

構文

```
_SERIAL$
```

例

サンプルコード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

結果

Model:******
Serial No.:**********
F/W Version:*.**

関連コマンド

_MODEL$, _VERSION$

12.18 _VERSION$

説明

変数は読み取りのみです。プリンターのファームウェアバージョンを含みます。

構文

```
_VERSION$
```

例

サンプルコード

```
SIZE 4,1
GAP 0,0
DIRECTION 1
CLS
TEXT 10,10, "3",0,1,1, "Model: " + _MODEL$
TEXT 10,60, "3",0,1,1, "Serial No.: " + _SERIAL$
TEXT 10,110, "3",0,1,1, "F/W Version: " + _VERSION$
PRINT 1
```

結果

Model:******

Serial No.:**********

F/W Version:*.**

関連コマンド

_MODEL$, _SERIAL$

13 Bluetooth 設定コマンド

13.1 BT NAME

説明

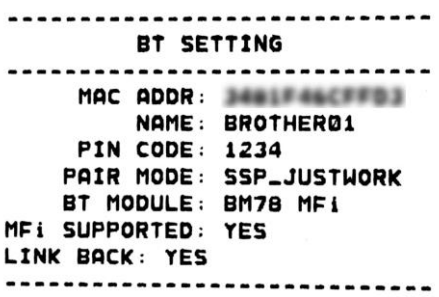
Bluetooth モジュール名を設定します（最大 15 バイト）。

構文

```
BT NAME "name"
```

備考：
SELFTEST BT を使用して正しく名前が更新されたかご確認いただけます。

例

サンプルコード	結果 (RJ-2035B)
`BT NAME "BROTHER01"` `SELFTEST BT`	

13.2 BT PINCODE

説明

Bluetooth モジュールの PIN コードを設定します（最大 15 バイト）。

構文

```
BT PINCODE “pincode”
```

備考：
SELFTEST BT を使用して設定が反映されたことをご確認いただけます。

例

サンプルコード	結果 (RJ-2035B)
`BT PINCODE “1234”` `SELFTEST BT`	---------- BT SETTING ---------- MAC ADDR: [illegible] NAME: BROTHER01 PIN CODE: 1234 PAIR MODE: LEGACY BT MODULE: BM78 MFi MFi SUPPORTED: YES LINK BACK: YES ----------

13.3 BT MODE

説明

Bluetooth Low Energy (BLE) を有効/無効にします。

構文

- **Bluetooth Low Energy (BLE) を有効にします**

[TD-4650TNWB/TD-4750TNWB/TD-4750TNWBR/TJ-4420TN/TJ-4520TN/TJ-4620TN]

次のコマンドを送信します。

```
BT MODE "BT4.0"
WLAN MODULE SAVECFG
DELAY 30000
INITIALPRINTER
```

30 秒後にプリンターが自動的に再起動します。

[RJ-2035B/RJ-3035B]

次のコマンドを送信します。

```
BT MODE BT4.0
SET BTLINKBACK OFF
```

BLE 機能を ON にすると、プリンターと iOS デバイス/Android デバイスを Bluetooth で自動で再接続する機能が無効になります。15 秒後にプリンターが自動的に再起動します。

- **Bluetooth Low Energy (BLE) を無効にします**

[TD-4650TNWB/TD-4750TNWB/TD-4750TNWBR/TJ-4420TN/TJ-4520TN/TJ-4620TN]

次のコマンドを送信します。

```
BT MODE "BT2.1"
WLAN MODULE SAVECFG
DELAY 30000
INITIALPRINTER
```

30 秒後にプリンターが自動的に再起動します。

[RJ-2035B/RJ-3035B]

プリンターと iOS デバイス/Android デバイスを Bluetooth で自動再接続する機能を有効にしたい場合
次のコマンドを送信します。
15 秒後にプリンターが自動的に再起動します。

```
BT MODE BT2.1
SET BTLINKBACK ON
```

プリンターと iOS デバイス/Android デバイスを Bluetooth で自動再接続する機能を使用しない場合
次のコマンドを送信します。

プリンターを再起動させます。

```
BT MODE BT2.1
```

備考：

[TD-4650TNWB/TD-4750TNWB/TD-4750TNWBR/TJ-4420TN/TJ-4520TN/TJ-4620TN]

- BLE（Bluetooth Low Energy）を有効・無効にする機能は、最新のファームウェアバージョンからご使用いただけます。
- 現在の「Bluetooth/Wi-Fi モジュールファームウェア」が、BLE に対応しているバージョン (5.2.1.0R5 以降) かを確認してください。

[RJ-2035B/RJ-3035B]

- プリンタ—と iOS デバイス/Android デバイスを Bluetooth で自動再接続する機能は、最新のファームウェアバージョンからご使用いただけます。
- 初期設定では、プリンタ—と iOS デバイス/Android デバイスを Bluetooth で自動再接続する機能が有効になっています。BLE 機能を ON にすると、プリンターと iOS デバイス/Android デバイスを Bluetooth で自動で再接続する機能が無効になります。

13.4 SET BTLINKBACK

説明

iOS デバイス/Android デバイスとの Bluetooth ペアリング再接続機能を有効/無効にします（MFi モデルのみ）。再接続機能が有効になっていると、前回接続した iOS デバイス/Android デバイスが自動でペアリングされます。

構文

```
SET BTLINKBACK ON      : 再接続機能を有効にします（デフォルト）
SET BTLINKBACK OFF     : 再接続機能を無効にします
```

備考：

- 初期設定では、プリンタ—と iOS デバイス/Android デバイスを Bluetooth で自動再接続する機能が有効になっています。
- プリンタ—と iOS デバイス/Android デバイスを Bluetooth で自動再接続する機能は、RJ-2035B/RJ-3035B の B1.00.Q33 以降のファームウェアバージョンからご使用いただけます。

14 Wi-Fi 設定コマンド

14.1WLAN SSID

説明

Wi-Fi モジュールにお使いの無線ネットワークの SSID を設定します。

構文

```
WLAN SSID "ssid"
```

パラメーター	説明
ssid	お使いの無線ネットワークの SSID

備考：
SSID は大小文字区別です。最大長さは 32 バイトです。

1. **WLAN SSID** コマンドを入力します。
2. 30 秒の待機時間を設定するために **DELAY** コマンドを入力します。次に電子音を鳴らす **SOUND** コマンドを入力します。
3. コマンドを送信します。
   例：
   ```
   WLAN SSID "TEST-AP"
   DELAY 30000
   SOUND 5,400
   ```
4. 約 30 秒待ちます。「ピー」という音が鳴った後に、プリンターを再起動してください。

備考：
Wi-Fi 設定を確認するには、**SELFTEST WLAN** コマンドを送信して Wi-Fi 設定情報を印刷します。

SELFTEST WLAN	---------- Wi-Fi SETTING ---------- APP VERSION: 4.6.1.0R7 MAC ADDRESS: [illegible] Region: Japan SSID: TEST-AP DHCP ENABLED: NO IP ADDRESS: 0.0.0.0 SUBNET MASK: 0.0.0.0 GATEWAY: 0.0.0.0 PRINTER NAME: [illegible] RAW PORT: 9100 ----------

14.2WLAN WPA

説明

WPA セキュリティーモードを設定します。

構文

```
WLAN WPA OFF
WLAN WPA "key"
```

パラメーター	説明
OFF	WPA セキュリティーモードを無効にします
Key	ネットワークセキュリティキー（8～63 文字） Key = パスワードまたは事前共有キー (パスワード：8～63 文字を含む文字列) (事前共有キー：32 バイトの 16 進数)

1. **WLAN WPA** コマンドを入力します。
2. 30 秒の待機時間を設定するために **DELAY** コマンドを入力します。次に電子音を鳴らす **SOUND** コマンドを入力します。
3. コマンドを送信します。
   例：
   ```
   WLAN WPA OFF
   WLAN WPA "123456789"
   DELAY 30000
   SOUND 5,400
   ```
4. 約 30 秒待ちます。「ピー」という音が鳴った後に、プリンターを再起動してください。

14.3WLAN WEP

説明

WEP セキュリティーモードを設定します。

構文

```
WLAN WEP OFF
WLAN WEP n, "key"
```

パラメーター	説明
OFF	WEP セキュリティーモードを無効にします
N	WEP キー（1～4）
Key	暗号キー（「5 文字または 13 文字の半角英数」または「10 桁または 26 桁の 16 進数」）

1. **WLAN WEP** コマンドを入力します。
2. 30 秒の待機時間を設定するために **DELAY** コマンドを入力します。次に電子音を鳴らす **SOUND** コマンドを入力します。
3. コマンドを送信します。
   例：

```
WLAN WEP OFF
WLAN WEP 1, "ABCDE"
WLAN WEP 2, "ABCDE"
WLAN WEP 3, "ABCDE"
WLAN WEP 4, "4142434445"
DELAY 30000
SOUND 5,400
```

4. 約 30 秒待ちます。「ピー」という音が鳴った後に、プリンターを再起動してください。

14.4WLAN DHCP

説明

DHCP サーバーから IP アドレスを取得するためにプリンターを設定します。

構文

```
WLAN DHCP
```

1. **WLAN DHCP** コマンドを入力します。
2. 30 秒の待機時間を設定するために **DELAY** コマンドを入力します。次に電子音を鳴らす **SOUND** コマンドを入力します。
3. コマンドを送信します。

   例：

```
WLAN SSID "TEST-AP"
WLAN WPA "123456789"
WLAN DHCP
WLAN PORT 9100
DELAY 30000
SOUND 5,400
```

4. 約 30 秒待ちます。「ピー」という音が鳴った後に、プリンターを再起動してください。

備考：

Wi-Fi 設定を確認するには、**SELFTEST WLAN** コマンドを送信して Wi-Fi 設定情報を印刷します。

SELFTEST WLAN	---------- Wi-Fi SETTING ---------- APP VERSION: 4.6.1.0R7 MAC ADDRESS: [illegible] Region: Japan SSID: TEST-AP DHCP ENABLED: YES IP ADDRESS: 10.0.10.138 SUBNET MASK: 255.255.255.0 GATEWAY: 10.0.10.252 PRINTER NAME: [illegible] RAW PORT: 9100 ----------

関連コマンド

WLAN IP

14.5WLAN IP

説明

プリンターに指定の Static IP アドレスを設定します。

構文

```
WLAN IP "ip", "mask", "gateway"
```

パラメーター	説明
ip	IP アドレス
Mask	サブネットマスク
Gateway	ゲートウェイ

1. **WLAN IP** コマンドを入力します。
2. 30 秒の待機時間を設定するために **DELAY** コマンドを入力します。次に電子音を鳴らす **SOUND** コマンドを入力します。
3. コマンドを送信します。

   例：

```
WLAN SSID "TEST-AP"
WLAN WPA "123456789"
WLAN IP "10.0.10.138",  255.255.255.0",  10.0.10.252"
WLAN PORT 9100
DELAY 30000
SOUND 5,400
```

4. 約 30 秒待ちます。「ピー」という音が鳴った後に、プリンターを再起動してください。

備考：

Wi-Fi 設定を確認するには、**SELFTEST WLAN** コマンドを送信して Wi-Fi 設定情報を印刷します。

SELFTEST WLAN	---------- Wi-Fi SETTING ---------- APP VERSION: 4.6.1.0R7 MAC ADDRESS: [illegible] Region: Japan SSID: TEST-AP DHCP ENABLED: NO IP ADDRESS: 10.0.10.138 SUBNET MASK: 255.255.255.0 GATEWAY: 10.0.10.252 PRINTER NAME: [illegible] RAW PORT: 9100 ----------

関連コマンド

WLAN DHCP

14.6 WLAN PORT

説明

Wi-Fi モジュールのポート番号を指定します。

構文

```
WLAN PORT number
```

パラメーター	説明
number	Base Raw ポート番号（初期値：9100）

1. **WLAN PORT** コマンドを入力します。
2. 30 秒の待機時間を設定するために **DELAY** コマンドを入力します。次に電子音を鳴らす **SOUND** コマンドを入力します。
3. コマンドを送信します。
   例：

```
WLAN SSID "TEST-AP"
WLAN WPA "123456789"
WLAN IP "10.0.10.138",  255.255.255.0",  10.0.10.252"
WLAN PORT 8000
DELAY 30000
SOUND 5,400
```

4. 約 30 秒待ちます。「ピー」という音が鳴った後に、プリンターを再起動してください。

備考：
Wi-Fi 設定を確認するには、**SELFTEST WLAN** コマンドを送信して Wi-Fi 設定情報を印刷します。

SELFTEST WLAN	---------- Wi-Fi SETTING ---------- APP VERSION: 4.6.1.0R7 MAC ADDRESS: [illegible] Region: Japan SSID: TEST-AP DHCP ENABLED: NO IP ADDRESS: 10.0.10.138 SUBNET MASK: 255.255.255.0 GATEWAY: 10.0.10.252 PRINTER NAME: [illegible] RAW PORT: 8000 ----------

14.7WLAN THROUGH

説明

TJ-4420TN, TJ-4520TN, TJ-4620TN の接続モードを Wi-Fi／Bluetooth 接続に設定します。

TJ-4420TN, TJ-4520TN, TJ-4620TN は、Wi-Fi／Bluetooth 接続と有線 LAN を同時に使用することはできません。これらのモデルでは同時に接続していても、有線／無線切り替えで設定されている接続が有効になります。

構文

```
WLAN THROUGH
```

- **有線 LAN 接続から Wi-Fi／Bluetooth 接続に切り替えます**

  次のコマンドを送信します。

  ```
  WLAN THROUGH
  ```

  プリンタ—が自動的に再起動します。

- **Wi-Fi／Bluetooth 接続から有線 LAN 接続に切り替えます**

  次のコマンドを送信します。

  ```
  NET THROUGH
  ```

  プリンタ—が自動的に再起動します。

関連コマンド

NET THROUGH

15 内部イーサネット設定コマンド

15.1NET DHCP

説明

DHCP サーバーから IP アドレスを取得するように設定します。このコマンドを設定するとプリンターは自動的に再起動します。

構文

```
NET DHCP
```

例

サンプルコード	結果
`NET DHCP` `SELFTEST ETHERNET`	---------- ETHERNET SETTING ---------- NAME: XXXXXX MAC ADDR: XXXXXX DHCP: ON IP ADDR: 192.168.0.107 SUBNET: 255.255.255.0 GATEWAY: 192.168.0.1 PORT: 9100 ----------

関連コマンド

NET IP

15.2 NET IP

説明

プリンターに指定の IP アドレスを設定します。このコマンドを設定するとプリンターは自動的に再起動します。

構文

```
NET IP "ip","mask","gateway"
```

パラメーター	説明
ip	IP アドレス
mask	サブネットマスク
gateway	デフォルトゲートウェイ

例

サンプルコード

```
NET IP "192.168.10.40","255.255.255.0","192.168.10.252"
SELFTEST ETHERNET
```

結果

```
------------------------------
      ETHERNET SETTING
------------------------------
        NAME: XXXXXX
    MAC ADDR: XXXXXX
        DHCP: OFF
     IP ADDR: 192.168.10.40
      SUBNET: 255.255.255.0
     GATEWAY: 192.168.10.252
        PORT: 9100
------------------------------
```

関連コマンド

NET DHCP

15.3 NET PORT

説明

イーサネットのポート番号を指定します。このコマンドを設定するとプリンターは自動的に再起動します。

構文

```
NET PORT number
```

パラメーター	説明
number	Base Raw ポート番号（初期値：9100）

例

サンプルコード	結果
NET PORT 9100 SELFTEST ETHERNET	---------- ETHERNET SETTING ---------- NAME: XXXXXX MAC ADDR: XXXXXX DHCP: OFF IP ADDR: 192.168.10.40 SUBNET: 255.255.255.0 GATEWAY: 192.168.10.252 PORT: 9100 ----------

15.4NET NAME

説明

プリンターサーバー名を設定します。

構文

```
NET NAME "printerserver"
```

パラメーター	説明
printerserver	プリンターサーバーの指定名

例

サンプルコード	結果
NET NAME "TEST" SELFTEST ETHERNET	---------- ETHERNET SETTING ---------- NAME: TEST MAC ADDR: XXXXXX DHCP: OFF IP ADDR: 192.168.10.40 SUBNET: 255.255.255.0 GATEWAY: 192.168.10.252 PORT: 9100 ----------

15.5 NET THROUGH

説明

TJ-4420TN, TJ-4520TN, TJ-4620TN の接続モードを有線 LAN に設定します。

TJ-4420TN, TJ-4520TN, TJ-4620TN は、有線 LAN と Wi-Fi／Bluetooth を同時に使用することはできません。これらのモデルでは同時に接続していても、有線／無線切り替えで設定されている接続が有効になります。

構文

```
NET THROUGH
```

- **Wi-Fi／Bluetooth 接続から有線 LAN 接続に切り替えます**

  次のコマンドを送信します。

  ```
  NET THROUGH
  ```

  プリンタ—が自動的に再起動します。

- **有線 LAN 接続から Wi-Fi／Bluetooth 接続に切り替えます**

  次のコマンドを送信します。

  ```
  WLAN THROUGH
  ```

  プリンタ—が自動的に再起動します。

関連コマンド

WLAN THROUGH

16 RJ-2035B/3035B 向け設定コマンド

16.1SET PRINTQUALITY

説明

RJ-2035B/3035B 向けに印刷モード（印字品質）を設定します。

構文

```
SET PRINTQUALITY DRAFT/STANDARD/OPTIMUM
```

パラメーター	説明
DRAFT	印字濃度を薄くして、印字速度を早めます
STANDARD	標準の印字速度・印字品質にします
OPTIMUM	ラベルの内容（例：バーコード、テキスト、グラフィック）に応じて、印字速度を遅くすることで、高い印字品質に調整します

備考：
初期値：STANDARD

例

サンプルコード

```
SET PRINTQUALITY DRAFT
SET PRINTQUALITY STANDARD
SET PRINTQUALITY OPTIMUM
```

16.2 SET STANDBYTIME

説明

RJ-2035B/3035B 向けに待機時間を設定します。指定した時間が経過すると、自動的にバッテリーLED がオフになります。

構文

```
SET STANDBYTIME OFF/XXXXX
```

パラメーター	説明
OFF	無効
XXXXX	0 ~ 625534 (秒)

備考：
初期値：120 (秒)

例

サンプルコード

```
SET STANDBYTIME OFF
SET STANDBYTIME 480
```

16.3 SET SLEEPTIME

説明

RJ-2035B/3035B 向けにスリープ時間を設定します。指定した時間が経過すると、自動的にプリンターの電源がオフになります。

構文

```
SET SLEEPTIME OFF/XXXXX
```

パラメーター	説明
OFF	無効
XXXXX	0 ~ 65534 (分)

備考：
初期値：30

例

サンプルコード

```
SET SLEEPTIME OFF
SET SLEEPTIME 20
```

17 RFID 設定コマンド

17.1 RFID READ / RFID WRITE

説明

RFID タグの読み取りや書き込みを行います。

重要:

- RFIDタグを選定する際には、購入前に必ずお使いのプリンターで印刷することをお勧めします。
- RFIDキャリブレーションを行う前に、用紙キャリブレーションを実行します。
  リボン（熱転写のみ）およびラベルが正しくセットされていることを、必ずご確認ください。
- 詳細は**製品マニュアル**の*ユーザーズガイド*をご参照ください（support.brother.com）。

構文

```
RFID READ,A,B,C,D,E,"Read Data:"
RFID WRITE,A,B,C,D,E,data
```

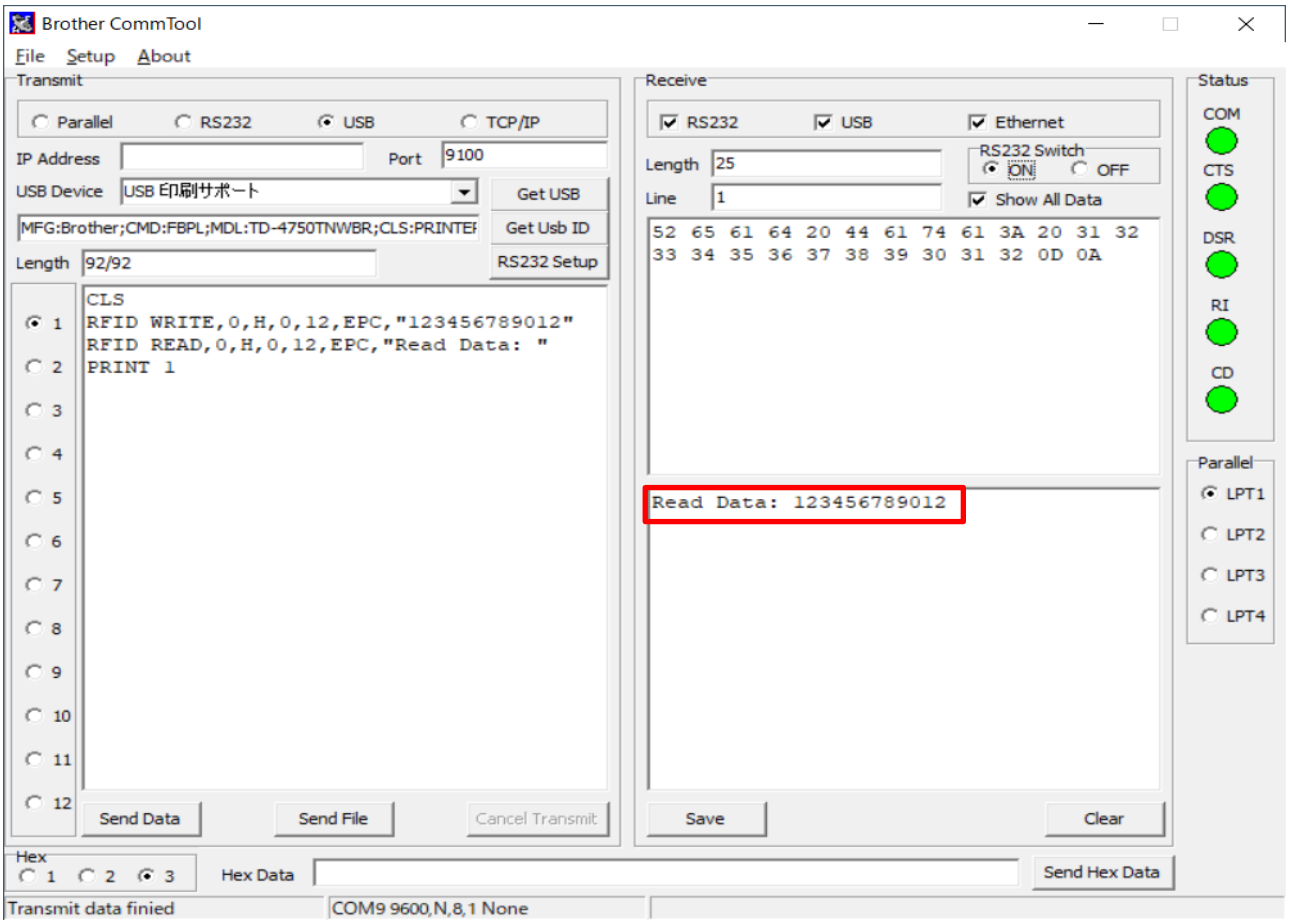
<table>
<tr><th>パラメーター</th><th>説明</th></tr>
<tr><td>RFID READ
RFID WRITE</td><td>READ = タグを読み取ります
WRITE = タグに書き込みます</td></tr>
<tr><td>A</td><td>
<table>
<tr><td>READ のみ</td><td>ロック解除</td><td>0 =ロック解除せずに読み取り
1～FFFFFFFF （16 進数）=
上書き可能にするため、データブロックを読み取り、ロックを解除します</td></tr>
<tr><td>WRITE のみ</td><td>ロック</td><td>0 = ロックせずに書き込み
1～FFFFFFFF （16 進数）=
上書きさせないようにするため、データブロックに書き込み、ロックします</td></tr>
</table>
</td></tr>
<tr><td>B</td><td>データフィールドの形式を指定します
A = ASCII
H = Hex</td></tr>
<tr><td>C</td><td>0 を指定します</td></tr>
<tr><td>D
Size</td><td>1 から n のデータサイズの読み取り/書き込みをします（10 進数）
備考：
- WRITE を使用する場合、「サイズ」が「データ」よりも大きいと、読み取るデータの前に 0 が付け加えられます。
- READ を使用する場合、「サイズ」が書き込みの「データ」より大きいと、読み取るデータの後ろに 0 が付け加えられます。
- サンプルコードの例 3 を参照。</td></tr>
<tr><td>E
Memory bank</td><td>EPC - EPC 領域（最大 496 ビットまで）
TID - Tag ID 領域 （RFID READ のみ）
USR - ユーザー領域 （最大 8k ビットまで）
ACS - アクセスコード
KIL - キルコード
PC - Protocol Control（PC）領域</td></tr>
<tr><td>"Read Data:"
or
data</td><td>READ = [データのプロンプト]
WRITE =データの内容

備考：
- RFID WRITE 文字列や基本変数（例：VAR$）に対応しています。
- [ ]= オプション：任意のパラメーター</td></tr>
</table>

例

サンプルコード (1)

```
CLS
RFID WRITE,0,H,0,12,EPC,"123456789012"
RFID READ,0,H,0,12,EPC,"Read Data: "
PRINT 1
```



サンプルコード (2)

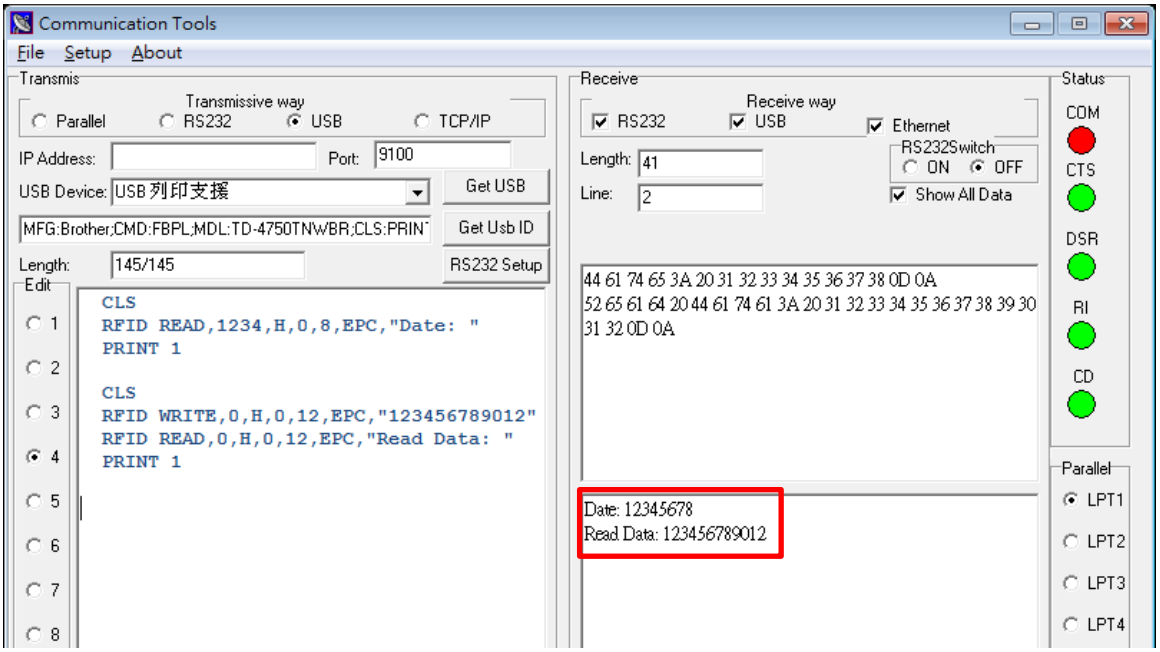
この例では、RFIDタグにパスワードロックでデータを書き込み、書き込まれたデータを読み取ります。

```
CLS
RFID WRITE,1234,H,0,8,EPC,"20191008"
RFID READ,0,H,0,8,EPC,"Date: "
PRINT 1
```

備考：
このロックされた RFID タグについて、RFID READ（ロック解除）コマンドを使用しない限り、データの上書きはできません。RFID WRITE コマンドを再送した場合、プリンターLCD 上で下記の様に表示されます。

このロックされたタグを上書きする場合、下記の例のように RFID READ（ロック解除）コマンドを使用します。RFID タグのパスワードのロックを解除することにより、上書きが可能になります。

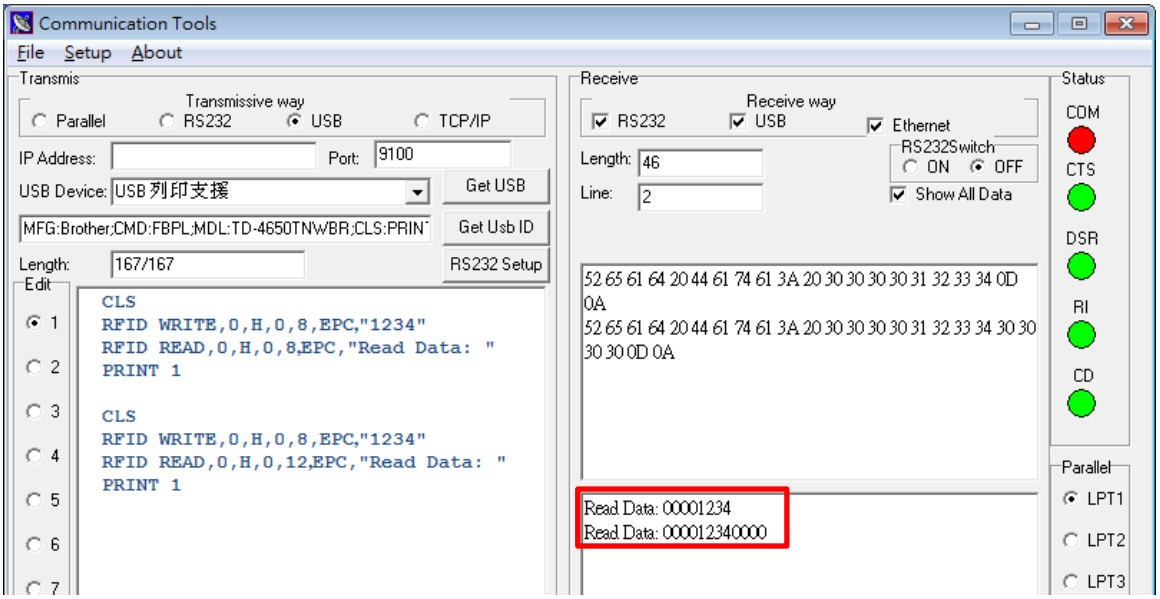
```
CLS
RFID READ,1234,H,0,8,EPC,"Date: "
PRINT 1

CLS
RFID WRITE,0,H,0,12,EPC,"123456789012"
RFID READ,0,H,0,12,EPC,"Read Data: "
PRINT 1
```



サンプルコード (3)

書き込む（RFID WRITE）場合、「サイズ」が「データ」よりも大きいと、読み取りをするデータの前に 0 が付け加えられます。また読み取る（RFID READ）場合、「サイズ」が書き込みの「データ」より大きいと、読み取るデータの後ろに 0 が付け加えられます。

```
CLS
RFID WRITE,0,H,0,8,EPC,"1234"
RFID READ,0,H,0,8,EPC,"Read Data: "
PRINT 1

CLS
RFID WRITE,0,H,0,8,EPC,"1234"
RFID READ,0,H,0,12,EPC,"Read Data: "
PRINT 1
```

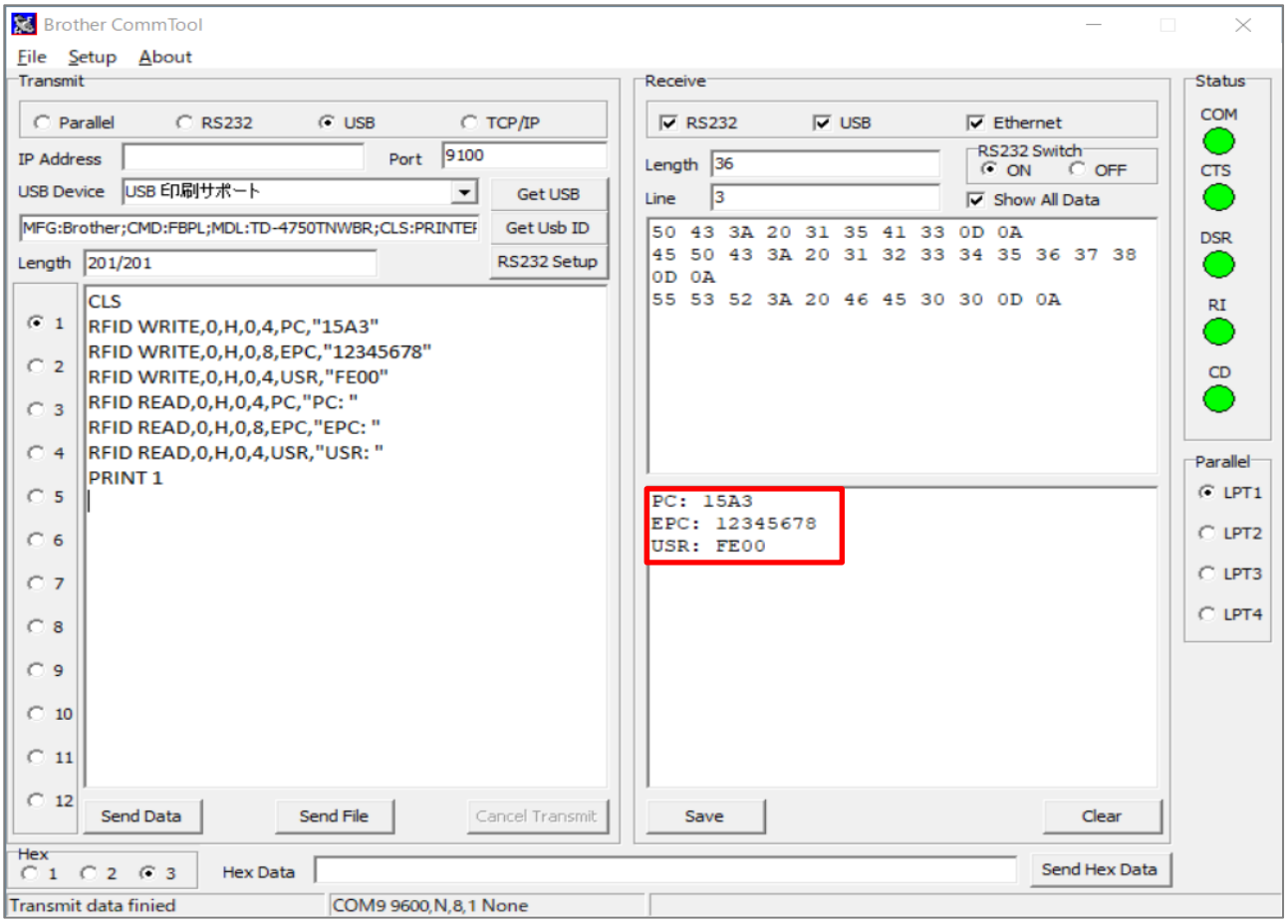


サンプルコード (4)

PC（Protocol Control）領域を書き換える場合の例です。

備考：
PC 領域への書き込みは、TD-4750TNWBR の V1.04 S21 以降のファームウェアバージョンからご使用いただけます。

```
CLS
RFID WRITE,0,H,0,4,PC,"15A3"
RFID WRITE,0,H,0,8,EPC,"12345678"
RFID WRITE,0,H,0,4,USR,"FE00"
RFID READ,0,H,0,4,PC,"PC: "
RFID READ,0,H,0,8,EPC,"EPC: "
RFID READ,0,H,0,4,USR,"USR: "
PRINT 1
```



17.2 RFIDDETECT

説明

RFID キャリブレーションを実行します。

18 General purpose input output (GPIO) 設定コマンド

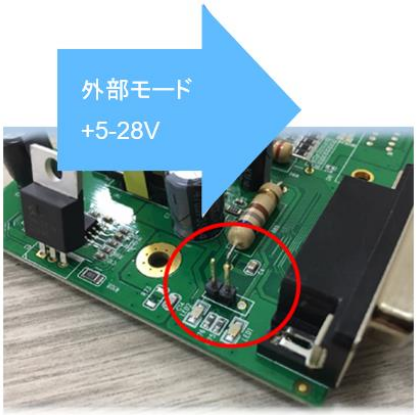
18.1GPIO をお使いの前に

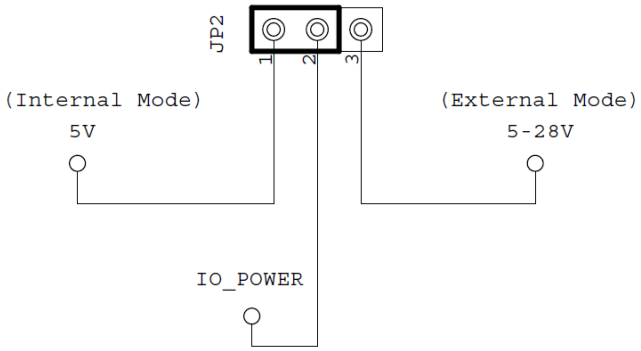
内部／外部モード（JP2）

- デフォルトは**内部モード**です。**外部モード**をご使用になる場合は、ご購入の前にお問合せください。
- **外部モード**では、必ず Pin 2 の電圧 **5-28V** の範囲で使用してください。この範囲を超える使用は、基板の故障の原因となります。

内部モード（デフォルト）
IO Power は 5V に設定されます

外部モード
PIN2 に入力した電圧レベルが IO Power に適用されます





- 本機のファームウェアが次のバージョンであることを確認します。
  SET GPO コマンドと **SET GPI** コマンドを使用して、GPIO を制御します。

モデル名	ファームウェアバージョン
TJ-4021TN/4121TN/4121TNR	B2.12.S70 以降
TJ-4420TN/4520TN/4620TN	B1.17.S54 以降

18.2 SET GPO

説明

指定したプリンターステータスを検知したときに電気信号を出力します。

構文

```
SET GPOn signal state, delay0, pulse0, delay1, pulse1, function condition
```

パラメーター **説明**

n
PIN 番号（n = 1 - 7）
7 本の GPO ピンから PIN 番号を指定します。

signal state

HIGH	function condition で指定したステータスを検知したとき、HIGH レベルの信号を出力します
LOW	function condition で指定したステータスを検知したとき、LOW レベルの信号を出力します
POS	function condition で指定したステータスを検知したとき、正パルス信号を出力します
NEG	function condition で指定したステータスを検知したとき、負パルス信号を出力します

delay0
function condition で指定したステータスを検知してから、”true”信号を出力するまでの待機時間を指定します。
（単位：ミリ秒、最大値：32000）

pulse0
function condition で指定したステータスを検知したときに出力されるパルス信号のパルス幅を指定します。
（単位：ミリ秒、最大値：32000）

delay1
function condition で指定したステータスを検知してから、”false”信号を出力するまでの待機時間を指定します。
（単位：ミリ秒、最大値：32000）

pulse1
function condition で指定したステータスが終了したときに出力されるパルス信号のパルス幅を指定します。
（単位：ミリ秒、最大値：32000）

function condition (warning, error, control…)	FAULT RIBBON	リボンエラー
	FAULT PAPER	用紙切れ・用紙詰まりエラー
	FAULT CARRIAGE	印字ヘッドオープン
	FAULT MEMORY	メモリフル
	FAULT CUTTER	カッターエラー（カッターモードのみ）
	FAULT OVERHEAT	ステッピングモーターまたは印字ヘッドのオーバーヒート
	TAKELABEL	ラベルがとられました（ハクリモードのみ）
	PAUSE	停止中
	IDLE	待機中
	PRINT	印字中
	FAULT	その他のプリンターエラー

例

サンプルコード (1)

```
SET GPO1 HIGH,100,0,100,0,FAULT RIBBON
SET GPO2 LOW,100,0,100,0,FAULT PAPER
SET GPO3 POS,100,100,100,100,PAUSE
SET GPO4 NEG,100,50,100,50,IDLE
```

サンプルコード (2)

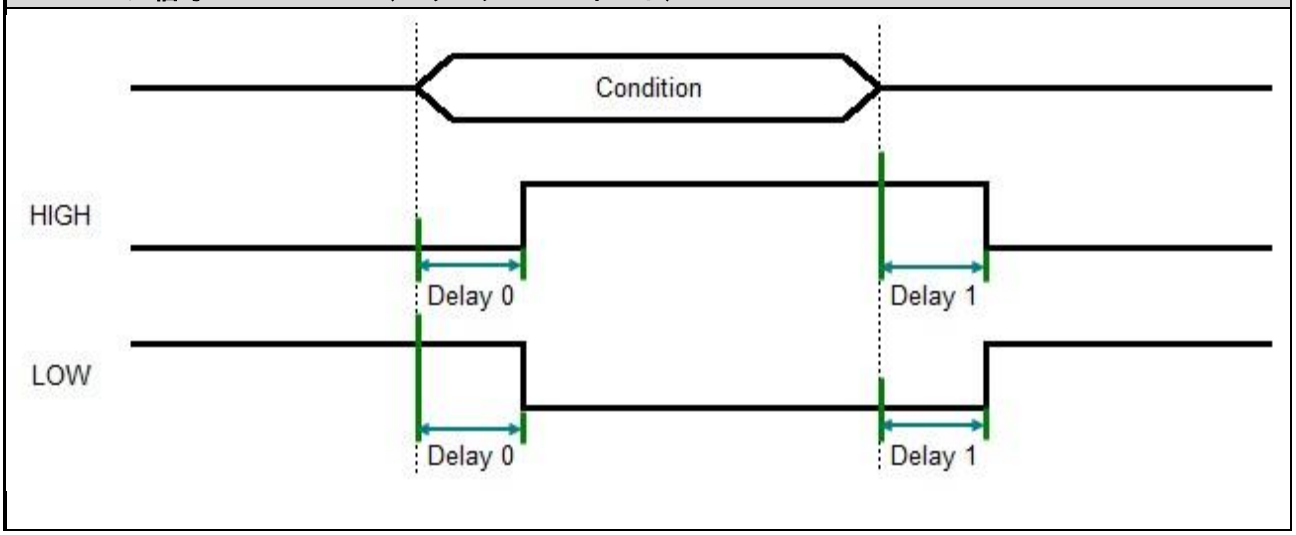
```
SET GPO1 HIGH,100,0,100,0,IDLE
SET GPO2 LOW,100,0,100,0,FAULT CARRIAGE
```

関連コマンド

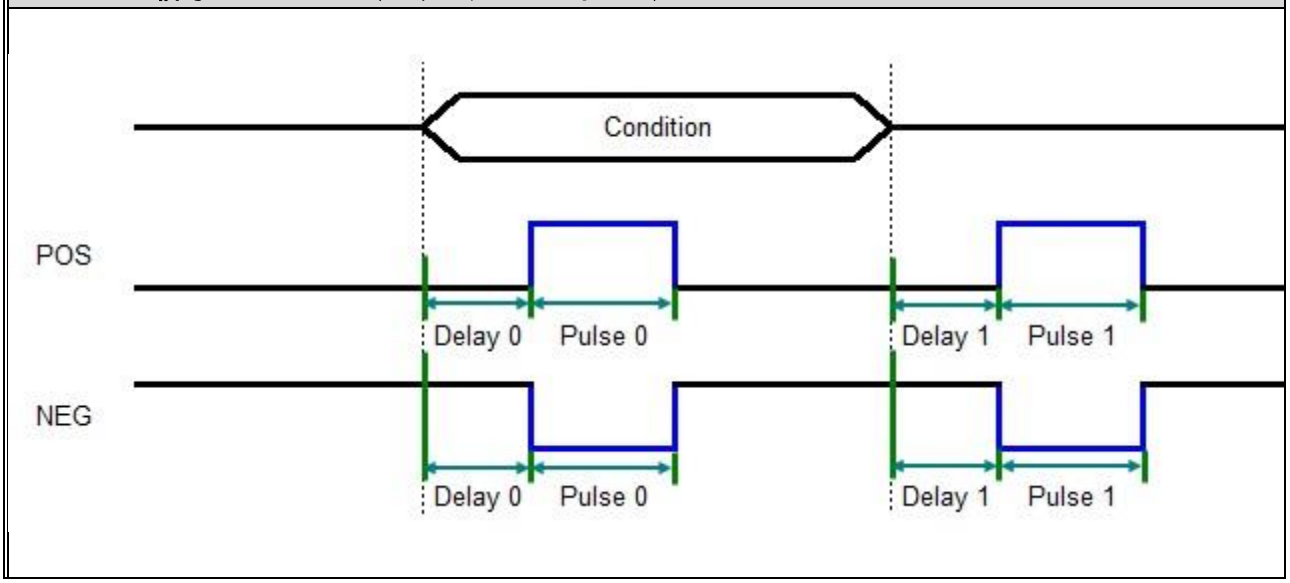
SET GPI

タイムチャート

GPO レベル信号：HIGH/LOW（プリンター → ホスト）



GPO パルス信号：POS/NEG（プリンター → ホスト）



18.3 SET GPI

説明

指定した電気信号を受信したときに実行するプリンターの制御を定義します。

構文

```
SET GPIn signal, pulse, function
```

例

パラメーター	説明
n	PIN 番号（n = 1 - 4） 4 本の GPI ピンから PIN 番号を指定します。
signal state	
pulse	パルス幅のしきい値を指定します。指定した幅より短いパルス幅の信号には応答しません。（単位：ミリ秒、最大値：32000）
function	

signal state

HIGH	HIGH レベルの信号を受信したとき、function で指定したプリンター制御を実行します
LOW	LOW レベルの信号を受信したとき、function で指定したプリンター制御を実行します
POS	正パルス信号を受信したとき、function で指定したプリンター制御を実行します
NEG	負パルス信号を受信したとき、function で指定したプリンター制御を実行します

function

PAUSE	停止状態をトグルします
PAUSE ON	停止状態に入ります
PAUSE OFF	停止状態をキャンセルします
PRINT	最後に印刷したラベルを再印刷します。プリンターの電源を切ると、再印刷用のデータは削除されます。印刷データがない場合は、白紙出力されます
PRINT n	最後に印刷したラベルを n 枚 再印刷します。印刷枚数を指定します。プリンターの電源を切ると、再印刷用のデータは削除されます。印刷データがない場合は、白紙出力されます（最大値：32000）
CUT	ラベルを即時カットします
FEED n	n ドット分 用紙を搬送します（最大値：32000）
BACKFEED n	n ドット分 用紙を逆送りします（最大値：32000）
FORMFEED	ラベルを 1 枚搬送します
INPUT n	BAS ファイル名（n）のあとに、改行コード「CHR$(13)+CHR$(10)」を入力しプログラムを開始します

サンプルコード (1)

```
SET GPI1 HIGH,0,PAUSE
SET GPI2 LOW,0,PAUSE ON
SET GPI3 POS,100,PAUSE OFF
SET GPI4 NEG,100,CUT
```

サンプルコード (2)

```
SET GPI1 NEG,100,INPUT "TEST.BAS"+CHR$(13)+CHR$(10)
```

サンプルコード (3)

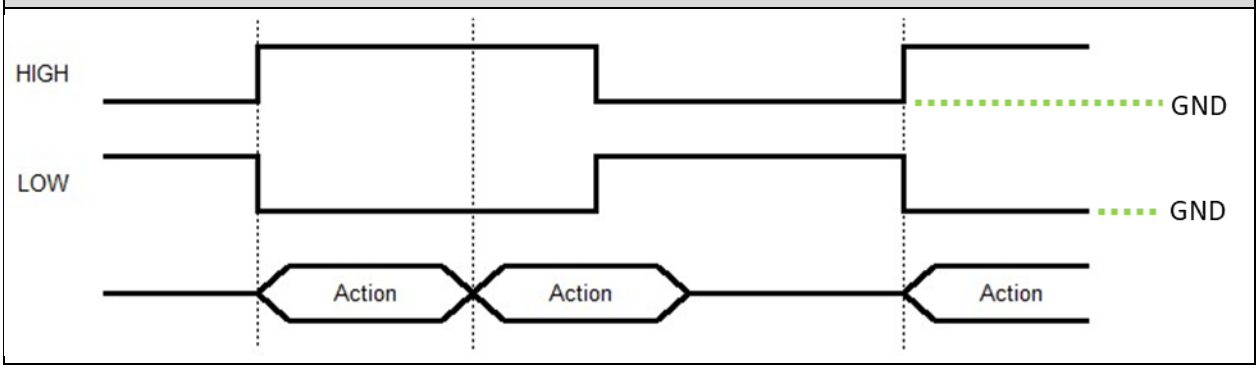
```
SET GPI1 POS,100,INPUT "TEST1.BAS" +CHR$(13)+CHR$(10)
SET GPI2 NEG,100,CUT
SET GPI3 POS,100,INPUT "TEST2.BAS" +CHR$(13)+CHR$(10)
```

関連コマンド

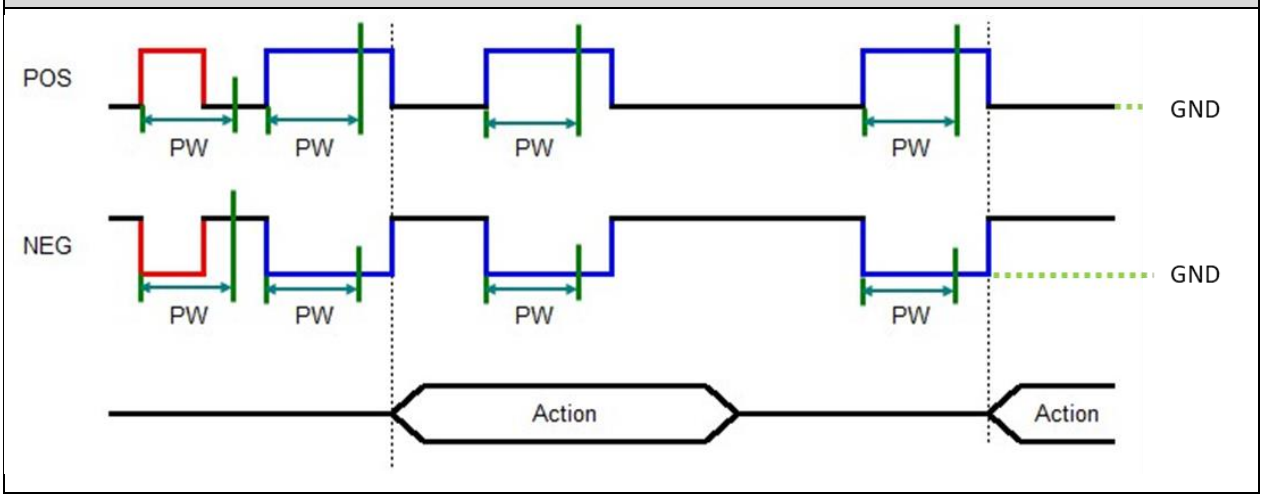
SET GPO

タイムチャート

GPI レベル信号：HIGH/LOW（ホスト → プリンター）

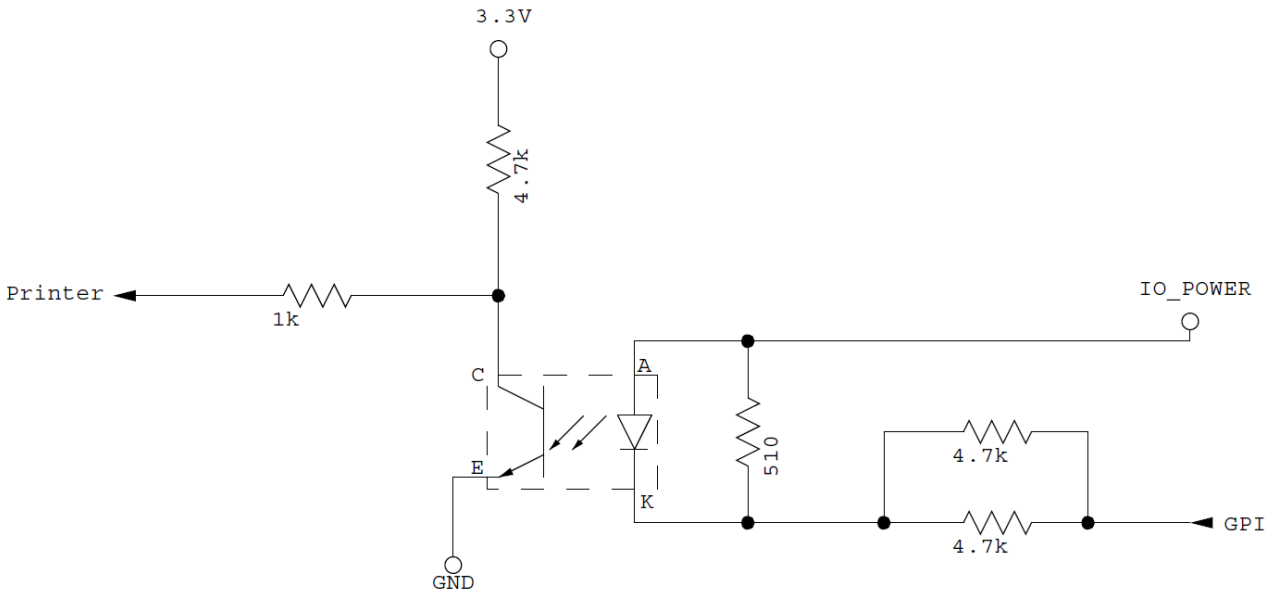


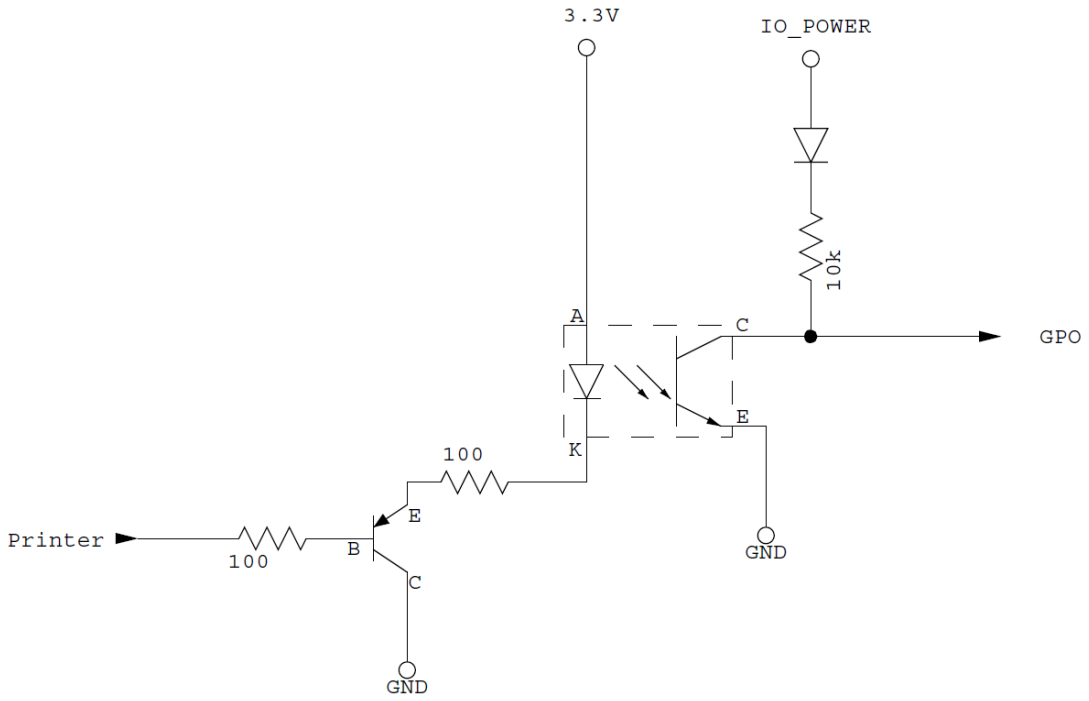
GPI パルス信号：POS/NEG（ホスト → プリンター）



18.4 GPIO ご使用上の注意

回路図



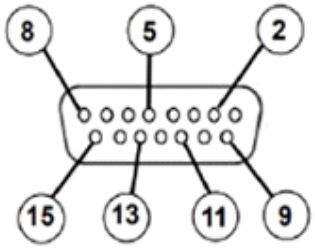


備考：

- フォトカプラは、PC357N1TJ000F (PC817) / SMD [IC]または PC357NT(PC817) / SMD [IC] が実装されます。これら 2 つのフォトカプラは同じ仕様です。
- IO_Power の電圧レベルの詳細は、**内部／外部モード（JP2）**をご参照ください。

PIN 配置

PIN 番号	信号	信号タイプ
1	GND	I/O Signal Ground
2	5V(JP2 short)	Power
3	GPI_1	Input
4	GPI_2	Input
5	GPI_3	Input
6	GPI_4	Input
7	24V	Power
8	GND	Power Ground
9	GPO_1	Output
10	GPO_2	Output
11	GPO_3	Output
12	GPO_4	Output
13	GPO_5	Output
14	GPO_6	Output
15	GPO_7	Output



備考：

- PIN7 に対して外部から信号や電圧を入力しないでください。基板が故障する恐れがあります。PIN7 は常時 24V レベルのリファレンス電圧が出力されています。通常は PIN7 を使用しないことを推奨します。
- 内部モードでは、PIN2 に対して外部から電圧を入力しないでください。基板が故障する恐れがあります。
- 外部モードでは、PIN2 に対して **5-28V** までの電圧を入力できます。入力した電圧レベルが IO_Power の電圧として使用されます。

付録 A: 機種別対応コマンド一覧

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
セットアップとシステムコマンド						
SIZE	○	○	○	○	○	○
GAP	○	○	○	○	○	○
GAPDETECT	○	○	○	○	○	○
BLINEDETECT	○	○	○	○	○	○
AUTODETECT	○	○	○	○	○	○
BLINE	○	○	○	○	○	○
OFFSET	○	○	○	○	○	○
SPEED	○	○	○	○	○	○
DENSITY	○	○	○	○	○	○
DIRECTION	○	○	○	○	○	○
REFERENCE	○	○	○	○	○	○
SHIFT	○	○	○	○	○	○
COUNTRY	○	○	○		○	○
CODEPAGE	○	○	○	○	○	○
CLS	○	○	○	○	○	○
FEED	○	○	○	○	○	○
BACKFEED	○	○	○	○	○	○
FORMFEED	○	○	○	○	○	○
HOME	○	○	○	○	○	○
PRINT	○	○	○	○	○	○
SOUND	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
CUT	○	○	○ RFID タグを除く		○ RFID タグを除く	○
LIMITFEED	○	○	○	○	○	○
SELFTEST	○	○	○	○	○	○
EOJ	○	○	○	○	○	○
DELAY	○	○	○	○	○	○
DISPLAY		○	○		○	○
INITIALPRINTER	○	○	○	○	○	○
MENU		○	○		○	○
ラベルフォーマットコマンド						
BAR	○	○	○	○	○	○
BARCODE	○	○	○	○	○	○
TLC39	○	○	○	○	○	○
BITMAP	○	○	○	○	○	○
BOX	○	○	○	○	○	○
CIRCLE	○	○	○	○	○	○
ELLIPSE	○	○	○	○	○	○
CODABLOCK F mode	○	○	○	○	○	○
DMATRIX	○	○	○	○	○	○
ERASE	○	○	○	○	○	○
MAXICODE	○	○	○	○	○	○
PDF417	○	○	○	○	○	○
AZTEC	○	○	○	○	○	○
MPDF417	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
PUTBMP	○	○	○	○ (1-bit モノクロ印刷のみ)	○	○
PUTPCX	○	○	○	○	○	○
QRCODE	○	○	○	○	○	○
RSS	○	○	○	○	○	○
REVERSE	○	○	○	○	○	○
DIAGONAL	○	○	○	○	○	○
TEXT	○	○	○	○	○	○
BLOCK	○	○	○	○	○	○
ステータス取得と即時実行コマンド						
<ESC>!?	○	○	○	○	○	○
<ESC>!C	○	○	○	○	○	○
<ESC>!D	○	○	○	○	○	○
<ESC>!O	○	○	○	○	○	○
<ESC>!P	○	○	○	○	○	○
<ESC>!Q	○	○	○	○	○	○
<ESC>!R	○	○	○	○	○	○
<ESC>!S	○	○	○	○	○	○
<ESC>!F	○	○	○	○	○	○
<ESC>!.	○	○	○	○	○	○
~!@	○	○	○	○	○	○
~!A	○	○	○	○	○	○
~!D	○	○	○	○	○	○
~!E	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
~!F	○	○	○	○	○	○
~!I	○	○	○	○	○	○
~!T	○	○	○	○	○	○
<ESC>Y				○		
<ESC>Z				○		
Windows ドライバー用コマンド						
!B	○	○	○	○	○	○
!J	○	○	○	○	○	○
!N	○	○	○	○	○	○
ファイル管理コマンド						
DOWNLOAD	○	○	○	○	○	○
EOP	○	○	○	○	○	○
FILES	○	○	○	○	○	○
KILL	○	○	○	○	○	○
MOVE	○	○	○	○	○	○
RUN	○	○	○	○	○	○
基本コマンドと制御コマンド						
ABS( )	○	○	○	○	○	○
ASC( )	○	○	○	○	○	○
CHR$( )	○	○	○	○	○	○
XOR$( )	○	○	○	○	○	○
END	○	○	○	○	○	○
EOF( )	○	○	○	○	○	○
OPEN	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
CLOSE	○	○	○	○	○	○
WRITE	○	○	○	○	○	○
READ	○	○	○	○	○	○
SEEK	○	○	○	○	○	○
LOF( )	○	○	○	○	○	○
LOC( )	○	○	○	○	○	○
FREAD$( )	○	○	○	○	○	○
PUT	○	○	○	○	○	○
GET	○	○	○	○	○	○
COPY	○	○	○	○	○	○
FOR…NEXT LOOP	○	○	○	○	○	○
WHILE…WEND	○	○	○	○	○	○
DO…LOOP	○	○	○	○	○	○
IF…THEN…ELSE…ENDIF LOOP	○	○	○	○	○	○
GOSUB…RETURN	○	○	○	○	○	○
GOTO	○	○	○	○	○	○
INP$( )	○	○	○	○	○	○
INP( )	○	○	○	○	○	○
LOB( )	○	○	○	○	○	○
PREINPUT	○	○	○	○	○	○
POSTINPUT	○	○	○	○	○	○
SET FILTER	○	○	○	○	○	○
REM	○	○	○	○	○	○
OUT	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
OUTR	○	○	○	○	○	○
GETKEY( )	○	○	○	○	○	○
INT( )	○	○	○	○	○	○
LEFT$( )	○	○	○	○	○	○
LEN( )	○	○	○	○	○	○
MID$( )	○	○	○	○	○	○
RIGHT$( )	○	○	○	○	○	○
STR$( )	○	○	○	○	○	○
STRCOMP( )	○	○	○	○	○	○
INSTR ( )	○	○	○	○	○	○
TRIM$( )	○	○	○	○	○	○
LTRIM$( )	○	○	○	○	○	○
RTRIM$( )	○	○	○	○	○	○
TEXTPIXEL( )	○	○	○	○	○	○
BARCODEPIXEL( )	○	○	○	○	○	○
VAL( )	○	○	○	○	○	○
NOW$( )	○	○	○	○	○	○
NOW	○	○	○	○	○	○
FORMAT$( )	○	○	○	○	○	○
DATEADD()	○	○	○	○	○	○
FSEARCH()	○	○	○	○	○	○
TOUCHPRESS()					○	
RECORDSET$ ( )	○	○	○		○	○
REPLACE$()	○	○	○		○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
デバイス動作設定コマンド						
SET COUNTER	○	○	○	○	○	○
SET CUTTER	○	○	○ RFID タグを除く		○ RFID タグを除く	○
SET PARTIAL_CUTTER	○	○	○ RFID タグを除く			
SET BACK	○	○	○		○	○
SET BACK_BEFORE_PRINT	○	○	○		○	○
SET KEYn	○	○	○	○	○	○
SET LEDn	○	○	○	○	○	○
SET PEEL	○	○	○ RFID タグを除く		○ RFID タグを除く	○
SET REWIND	○	○	○		○	○
SET TEAR & SET STRIPER	○	○	○	○	○	○
SET GAP	○	○	○	○	○	○
SET BLINE REVERSE / SET OBVERSE	○	○	○	○	○	○
SET BLINE PRINTSIDE / SET BLINE BACKSIDE						○
SET HEAD	○	○	○	○	○	○
SET RIBBON	○	○	○		○	○
SET ENCODER	○	○	○		○	○
SET RIBBONEND						○
SET COM1	○	○	○	○	○	○
SET PRINTKEY	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
SET REPRINT	○	○	○	○	○	○
SET FEED_LEN	○	○	○	○	○	○
GETSENSOR()	○	○	○	○	○	○
GETSETTING$()	○	○	○	○	○	○
SET USBHOST	○	○	○		○	○
SET AUTORUN	○	○	○	○	○	○
SET RESPONSE	○	○	○		○	○
SET DAYLIGHT_SAVE	○	○	○	○	○	○
SET REGISTRATION	○	○	○	○	○	○
PEEL	○	○	○ RFID タグを除く		○ RFID タグを除く	○
LED1, LED2, LED3	○	○	○	○	○	○
KEY1, KEY2, KEY3	○	○	○	○	○	○
SET SENSOR_REF	○	○	○	○	○	○
プリンターのグローバル変数						
@LABEL	○	○	○	○	○	○
YEAR	○	○	○	○	○	○
MONTH	○	○	○	○	○	○
DATE	○	○	○	○	○	○
WEEK	○	○	○	○	○	○
HOUR	○	○	○	○	○	○
MINUTE	○	○	○	○	○	○
SECOND	○	○	○	○	○	○
@YEAR	○	○	○	○	○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
@MONTH	○	○	○	○	○	○
@DATE	○	○	○	○	○	○
@DAY	○	○	○	○	○	○
@HOUR	○	○	○	○	○	○
@MINUTE	○	○	○	○	○	○
@SECOND	○	○	○	○	○	○
_MODEL$	○	○	○	○	○	○
_SERIAL$	○	○	○	○	○	○
_VERSION$	○	○	○	○	○	○
Bluetooth 設定コマンド						
BT NAME		○	○	○		○
BT PINCODE				○		
BT MODE				○		
SET BTLINKBACK				○		
Wi-Fi 設定コマンド						
WLAN SSID		○	○		○	○
WLAN WPA		○	○		○	○
WLAN WEP		○	○		○	○
WLAN DHCP		○	○		○	○
WLAN IP		○	○		○	○
WLAN PORT		○	○		○	○
WLAN THROUGH						○
内部イーサネット設定コマンド						
NET DHCP	○	○	○		○	○

コマンド	TD-4420TN TD-4520TN	TD-4650TNWB TD-4750TNWB	TD-4750TNWBR	RJ-2035B RJ-3035B	TJ-4021TN TJ-4121TN TJ-4121TNR	TJ-4420TN TJ-4520TN TJ-4620TN
NET IP	○	○	○		○	○
NET PORT	○	○	○		○	○
NET NAME	○	○	○		○	○
NET THROUGH						○
RJ-2035B/3035B 向け設定コマンド						
SET PRINTQUALITY				○		
SET STANDBYTIME				○		
SET SLEEPTIME				○		
RFID コマンド						
RFID READ			○		○ TJ-4121TNR のみ	
RFID WRITE			○		○ TJ-4121TNR のみ	
RFIDDETECT			○		○ TJ-4121TNR のみ	
GPIO コマンド						
SET GPO					○	○
SET GPI					○	○

付録 B: 日本語テキストを印刷する

内蔵フォント

本機は、日本語フォント「K1.SFT / K2.SFT / K3.SFT / K4.SFT / K8.SFT / K9.SFT / KA.SFT / KB.SFT / KD.SFT」を内蔵しています。内蔵フォントの詳細は **TEXT** コマンドを参照ください。

ダウンロードフォント

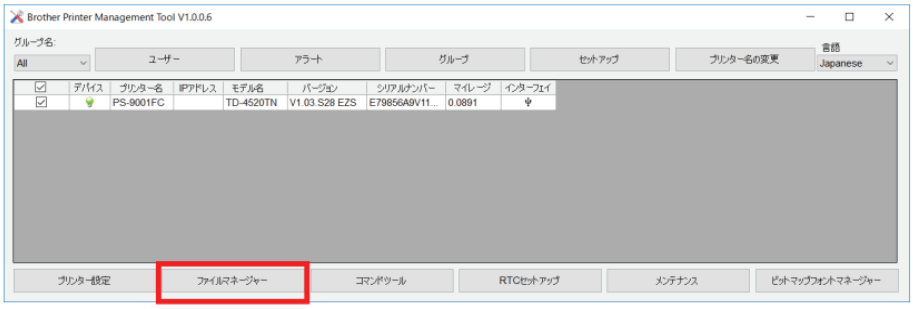
そのほかの日本語フォントを使用するには、あらかじめ日本語フォントをプリンターへ登録する必要があります。

★例として、オープンライセンスである IPAex フォントを用いる方法を説明します。IPAex フォントは次の URL よりダウンロード可能です。（IPAEXG.TTF：https://moji.or.jp/ipafont/）

★IPAex フォント（Ver.003.01)を使用ください。
本マニュアルの対象プリンターは IPAex フォント（Ver.004.01）には非対応のためご注意ください。

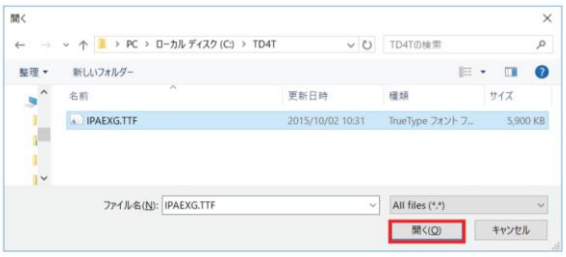
BPM ツール（Brother Printer Management）を用いて本機へ日本語フォントを転送します。

1. プリンターと本体を接続し、BPM を起動します。
2. [ファイルマネージャー]をクリックします。

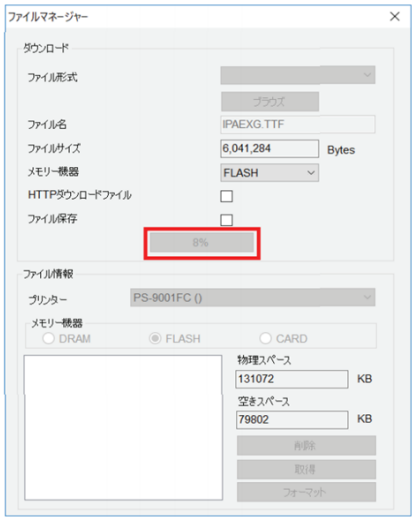


- Bluetoothで接続する場合、BPMは使用できません。

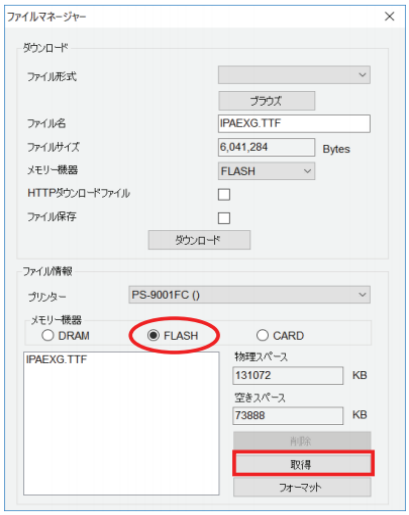
3. [ブラウズ]をクリックしてフォントファイルを選択します。プリンター本体へ送信するフォント名は**大文字（半角英数字）**で指定してください。（例：~~ipaexg.ttf~~ → IPAEXG.TTF）



4. [メモリー機器]ドロップダウンリストから[FLASH]を選択し、[ダウンロード]ボタンをクリックします。



5. [メモリー機器]のラジオボタンから[FLASH]を選択して[取得] ボタンを押します。左下にダウンロードしたフォントが表示されることを確認して、右上の「X」でファイルマネージャーを閉じます。



- 日本語を印刷する場合、印刷データの形式にあわせて**CODEPAGE**コマンドでコードページをShift JIS (932)またはUTF-8に定義したうえで、**TEXT**コマンドで日本語フォントを**大文字（半角英数字）**で指定する必要があります。

[Shift-JIS の場合]

```
CLS
CODEPAGE 932
TEXT 100,500,"IPAEXG.TTF",0,10,10,"ブラザー工業株式会社"
PRINT 1
```

[UTF-8 の場合] 第 3 水準、第 4 水準の日本語も印刷可能となります。

```
CLS
CODEPAGE UTF-8
TEXT 100,500,”IPAEXG.TTF”,0,10,10,”ブラザー工業株式会社”
PRINT 1
```

付録 C: 外部機器（PLC、バーコードリーダー等）を使用して印刷する

1. テンプレートを作成します

FBPL コマンドレファレンスを参照して、FBPL テンプレート用ファイルを作成します。

- **DOWNLOAD**コマンドを参照ください。
- テンプレートファイル名は**すべて大文字**にします。拡張子は必ず「.BAS」にします。
  （例：”TEST.BAS”、”DEMO.BAS”）

例：

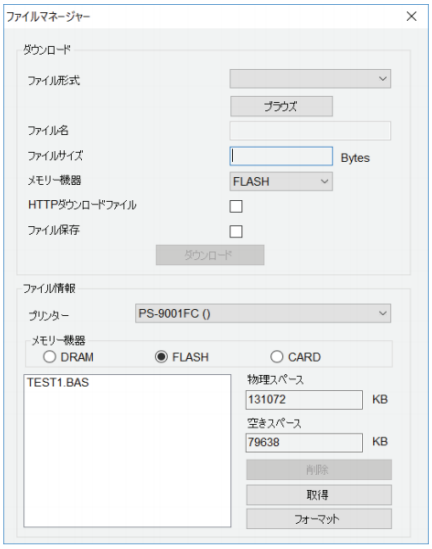
DOWNLOAD F,"DEMO.BAS"	"DEMO.BAS"テンプレートとしてプリンターに保存します
SIZE 48 mm, 50.8 mm	ラベルサイズを幅、長さの順に定義します
DIRECTION 0,0	印刷方向を定義します
REFERENCE 0,0	ラベルの基準点を X 座標：0 dot、Y 座標：0 dot と定義します
OFFSET 0 mm	用紙停止位置の調整することができます
SET TEAR ON	印刷後、ラベルを手切り位置まで送ります
CLS	イメージバッファを消去します
INPUT D1$	外部から入力したデータ 1 を文字列 D1$と定義します
INPUT D2$	外部から入力したデータ 2 を文字列 D2$と定義します
INPUT D3$	外部から入力したデータ 3 を文字列 D3$と定義します
INPUT D4$	外部から入力したデータ 4 を文字列 D4$と定義します
INPUT D5$	外部から入力したデータ 5 を文字列 D5$と定義します
BARCODE 67,287,"128M",53,1,0,2,4,D1$	文字列 D1$（外部から入力したデータ 1）のバーコード Code128 を印刷します
BAR 190,24, 3, 244	罫線を描きます
BAR 12,265, 369, 3	〃
CODEPAGE 932	コードページを「日本語 SHIFT-JIS」に設定します
TEXT 21,196,"K3.SFT",0,1,1,"日付"	文字列"日付"を印刷します
TEXT 21,224,"0",0,10,8,D2$	文字列 D2$（外部から入力したデータ 2）を印刷します
ERASE 234,203,115,50	イメージバッファ内の指定領域を消去します
TEXT 235,204,"0",0,17,15,"A3-01"	文字列"A3-01"を印刷します
REVERSE 234,203,115,50	イメージバッファ内の指定領域（"A3-01"部分）を反転させます
BAR 12,177, 369, 3	罫線を描きます
TEXT 25,58,"0",0,10,8,D3$	文字列 D3$（外部から入力したデータ 3）を印刷します
TEXT 25,97,"0",0,10,8,D4$	文字列 D4$（外部から入力したデータ 4）を印刷します
TEXT 199,58,"K3.SFT",0,1,1,"プリンター"	文字列"プリンター"を印刷します
TEXT 199,97,"0",0,10,8,"203dpi"	文字列"203dpi"を印刷します
TEXT 25,136,"0",0,10,8,"JAPAN"	文字列"JAPAN"を印刷します
BAR 10,24, 371, 3	罫線を描きます
BAR 378,27, 3, 363	〃
BAR 10,390, 371, 3	〃
BAR 10,27, 3, 363	〃
PRINT VAL(D5$)	VAL コマンドで文字列 D5$（外部から入力したデータ 5）を数値に変換して印刷枚数を指定します
EOP	プログラムの終了を宣言します
	改行します

2. テンプレートをプリンターへ転送します

テンプレートを BAS ファイル（Printer BASIC File）として、本機内部のフラッシュメモリへ記憶させます。

1. プリンターと本体を接続し、BPM（Brother Printer Management Tool）を起動します。
2. [コマンドツール]をクリックします。
3. [ブラウズ]をクリックして、テンプレート用ファイル（.BAS）を選択してから[開く]をクリックします。
4. [ファイル送信]をクリックして、テンプレート用ファイルをプリンターへ転送します。

- BPMツールの[ファイルマネージャー]を使用して、本機内部にBASファイルが保存されたことを確認できます。画面の表示に時間がかかる場合は、[取得]を押します。



3. プリンターと外部機器を接続します

●RC-232C 通信の設定方法

1. プリンターと本体を接続し、BPM（Brother Printer Management Tool）を起動します。
2. [プリンター情報]をクリックして[RS232]タブでシリアル情報を設定します。
3. プリンターと外部機器（PLC 等）を RC-232C 接続します。

- 外部機器（ホスト）とプリンターはストレートケーブルで通信可能です。

ホスト	9 Pin	25 Pin
RxD	2	3
TxD	3	2
DTR	4	20
GND	5	7
DSR	6	6
RTS	7	4
CTS	8	5

9 Pin	プリンター
1	+5V
2	TxD
3	RxD
4	DSR
5	GND
6	DTR
7	N/C
8	RTS
9	N/C

●GPIO の使用方法

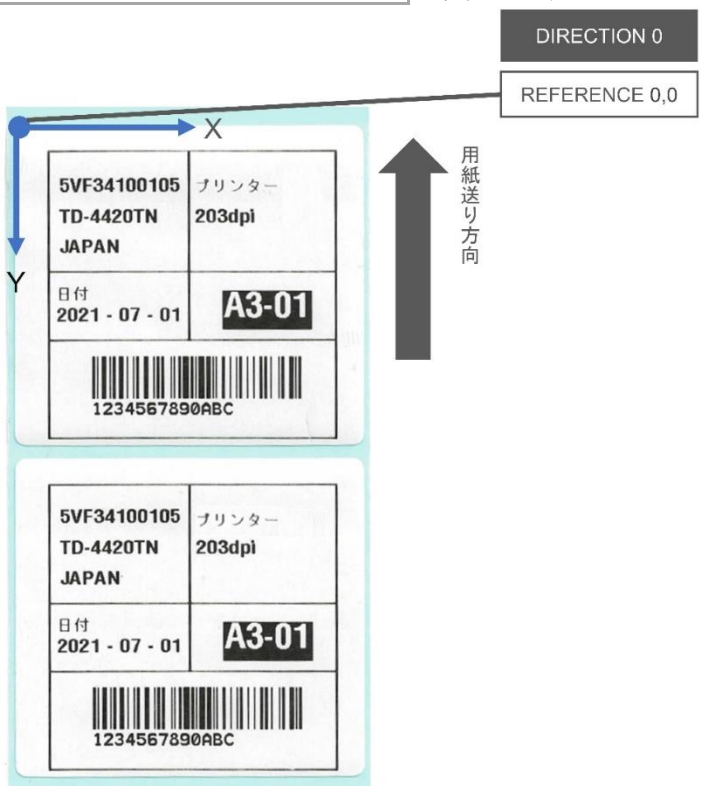
詳細は GPIO 設定コマンドを参照ください。

4. 印刷を実行します

外部機器（PLC、バーコードリーダー等）を使用して、プリンター内のテンプレートを呼び出し印刷します。

手順 1「テンプレートを作成します」のテンプレート例の場合は、外部機器からプリンターへ次のような値を送信します。最初にテンプレートの BAS ファイル名「DEMO」または **RUN** コマンド「RUN "DEMO.BAS"」を使用してテンプレートを呼びだします。次に、テンプレートの可変値（**INPUT** コマンド）を送信します。

1	DEMO	"DEMO.BAS"プログラムを開始します
2	!1051234567890!100ABC	外部から入力したデータ 1（文字列 D1$）
3	2021・07・01	外部から入力したデータ 2（文字列 D2$）
4	5VF34100105	外部から入力したデータ 3（文字列 D3$）
5	TD-4420TN	外部から入力したデータ 4（文字列 D4$）
6	2	外部から入力したデータ 5（文字列 D5$）
7		改行します



- プリンターと接続した外部機器から、テキストやバーコードなどの形式で、テンプレートの呼び出しとテンプレート内の可変値を入力できます。

```
DEMO
!1051234567890!100ABC
2021/07/01
5VF34100105
TD-4420TN
2
```

DEMO«CR»«LF»!1051234567890!100ABC
«CR»«LF»2021/07/01«CR»«LF»5VF34100105
«CR»«LF»TD-4420TN«CR»«LF»2«CR»«LF»

- 各値の末尾は＜改行＞コードを入れてください。

FAQ

Q: 本マニュアルの対象製品（TD-4T・TJ シリーズ）と TD-4D シリーズを両方使用したい場合、それぞれどのコマンドを使用するのが適切でしょうか。

A: 外部機器からの印刷の場合、TD-4D シリーズ向けには P-touch Template コマンドをご使用ください。（TD-4T・TJ シリーズは FBPL コマンドをご使用ください）

Q: 外部機器（PLC）からデータ送信時に、”00H”データが入ります。FBPL コマンド側に”00H”が含まれていても問題ありませんか。

A: 本機は”00H”データは読み飛ばすため、問題ありません。

Q: FBPL コマンドの最後に改行コードは必要ですか。

A: 必要です。各 FBPL コマンド間および FBPL コマンドの最後に改行コード（例：CRLF、1310、0x0D 0x0A）を追加します。

Q: PLC からの発行は可能ですか。また PLC からハブを経由して有線 LAN（イーサネット）ケーブルを接続して印刷できますか。

A: 本機（TD-4T・TJ シリーズ）のインターフェイスは RS-232C と有線 LAN に対応しています。本機は”00H”データは読み飛ばすため、問題ありません。

Q: プリンターに接続した USB バーコードリーダーを使用して、FBPL コマンドのバーコードを読み取りましたが、プリンターが反応しません。

A:下記を確認ください。

- 各 FBPL コマンド間および最後に改行コード（例：CRLF、1310、0x0D 0x0A）が必要です。
- LCD 搭載プリンターで、テンプレート内に **INPUT** コマンドと **SET FILTER** コマンドを使用されている場合は、プリンターを「USB バーコードリーダーモード」に設定する必要があります。テンプレートファイル（BAS ファイル）に **SET USBHOST SCANNER** を挿入してください。

brother