

Canon

コントロールROM

CR-MES/MEN/HIS

コントロールフォントROM

CFR-M1/M2

バーコードフォント ユーザーズガイド

ご使用前に必ずこの取扱説明書をお読みください。
将来いつでも使用できるように大切に保管してください。

JPN

はじめに

本書では、コントロールROM、コントロールフォントROMで使用できるバーコードフォントの種類、印字方法、印字サンプル等を説明いたします。

本ROMを正しくご使用いただくために、ご使用になる前に必ず本書をお読み下さい。

■ 注意

1. 本書の内容の一部または全部を無断転載することは、禁止されています。
2. 本書の内容は、将来予告なしに変更する場合があります。
3. 本書の内容は、万全を期して制作しております。万一、ご不審な点や誤り、あるいは記載もれなど、お気づきのことがありましたらご連絡ください。
4. 運用した結果の影響につきましては、上記3.項にかかわらず、一切の責任を負いかねます。

■ 商標について

Canon、Canonロゴ、iR、LBP、LIPSはキヤノン株式会社の商標です。

Microsoft、Windows は、米国Microsoft Corporationの米国および他の国における登録商標です。

その他の各会社、各製品は各社の商標または登録商標です。

目次

1. ご使用になる前に	1.1 ご注意-----	1
2. 文字セット	2.1 文字セットの種類-----	2
	2.2 文字セットの選択-----	2
3. バーコードの印刷方法	3.1 CODE 3 9、NW- 7の印刷方法-----	3
	3.2 CODE 3 9の印刷サンプル例-----	4
	3.3 NW- 7の印刷サンプル例-----	8
	3.4 JAN、EAN、UPCの印刷方法-----	12
	3.5 JAN、EAN、UPCの印刷サンプル例-----	21
	3.6 EAN- 1 2 8の印刷方法 (CR-MENのみ)-----	32
	3.7 EAN- 1 2 8の印刷サンプル例 (CR-MENのみ)-----	42
	3.8 カスタマバーコードの印刷方法-----	54
	3.9 カスタマバーコードの印刷サンプル例-----	55
	3.10 VC++用共通サンプルメインプログラム-----	58
4. バーコードフォント仕様	4.1 共通仕様-----	60
	4.2 JAN、EAN、UPC用フォント仕様-----	61
	4.3 CODE 3 9用フォント仕様-----	64
	4.4 NW- 7用フォント仕様-----	66
	4.5 EAN- 1 2 8フォント仕様 (CR-MENのみ)-----	67
	4.6 OCR_B用フォント仕様-----	69
	4.7 カスタマバーコードフォント仕様-----	69
5. カスタマバーコードについて	5.1 新郵便番号による郵便処理-----	70
	5.2 カスタマバーコードとは-----	71
	5.3 カスタマバーコードの仕様-----	72
	5.4 バーコードに必要な文字情報の抜き出し法-----	77
6. 付録	6.1 バーコードフォントカード 文字セット名称一覧-----	85
	6.2 フォント仕様一覧-----	86
	6.3 JANコードチェックデジット計算方法-----	95
	6.4 UPC短縮コードチェックデジット計算方法-----	96
	6.5 EAN- 1 2 8チェックデジット計算方法-----	97
	6.6 EAN- 1 2 8アプリケーション識別子一覧-----	98

1. ご使用になる前に

1.1 ご注意

お使いのLBPシリーズ／i Rシリーズによっては、本ROMを使用できない機種があります。

本ROMが使用できるかどうかについては、LBPシリーズに付属の「設置ガイド」または「ユーザズガイド」、i Rシリーズに付属の「LIPS プリントガイド」または「LIPS リファレンスガイド」をご覧ください。

本ROMは、以下のトランスレータでのみバーコード印刷にご使用になれます。

- LIPSトランスレータ
- N201トランスレータ
- ESC/Pトランスレータ
- F359トランスレータ
- I5577トランスレータ
- N5273トランスレータ

本ROMにて印刷されたバーコードは、紙の質やトナーの濃度等の原因で全ての読み取り機では読み取れない場合がありますので、予めご了承下さい。

動作確認をした機種

- | | |
|-------------|-------------|
| ■ バーコード検査装置 | CodaScan II |
| ■ ハンディターミナル | ST-100 |
| | HT-970 |

- (1) バーコードの印字可能領域は、LBPシリーズ／i Rシリーズの仕様上、用紙の端から上下左右5mmを除く領域です。
詳細はLBPシリーズ／i Rシリーズのマニュアルをご覧ください。
- (2) 印字するバーコードのサイズによっては、バーコードフォントの仕様上、印字可能領域を外れて印字されない場合もあります。
- (3) 反射率の低い用紙（再生紙等）では、読み取り機での精度が悪化します。
そのような場合は、ドット幅を太くするか、反射率の高い用紙をご使用になることをおすすめします。
- (4) LBPシリーズ／i Rシリーズのトナーの濃度によって、印刷されたバーコードの読み取り機で精度の変化が生じる場合があります。通常の場合、工場出荷時の設定で支障はありませんが、読み取り機の精度が悪く、(3)の調整を行った後でも、改善されない場合は、濃度調整を行うことで、読み取り機の精度が向上することがあります。

2. 文字セット

2.1 文字セットの種類

本フォントROMの文字セットで印字出来るバーコードは、以下の6書体となります。

- ・CODE 3 9
- ・NW-7
- ・JAN (EAN、UPC)
- ・EAN-128 (CR-MENのみ)
- ・標準料金代理収納用EAN-128 (CR-MENのみ)
- ・カスタマバーコード (郵便番号バーコード)

2.2 文字セットの選択

文字セットを選択するには、文字セット名称で選択して下さい。

詳細は、付録の文字セット名称一覧をご覧下さい。

 初めてLBPシリーズ出力ソフトウェアを開発される場合には、プログラマーズマニュアルが必要です。

- ・プログラマーズマニュアルソフトウェア解説書
- ・プログラマーズマニュアルコマンドリファレンス
- ・プログラマーズマニュアルクックブック

3. バーコードの印刷方法

3.1 CODE 39、NW-7の印刷方法

- 1** LIPSモード（スケラブルフォント）での印刷方法
- ① 文字セット名称選択命令にて、印字するバーコードの文字セットを選択してプリンタに送って下さい。
 - ② バーコードデータをプリンタに送る事によってバーコードの印刷ができます。

- 2** 各エミュレーションモード（ドットフォント）での印刷方法
- ① 文字セット名称選択命令 I のコマンドをプリンタに送る事によって印刷します。
（例）ESC/Pエミュレーションモード時、文字セット名称“C3060. BAR”
でバーコードデータ *12345* を印刷する場合
`[ESC] I [SP] _ _ f C3060. BAR, 100, 000, 000, *12345*`

注意：エミュレーションモードとは、各オプションコントロールROMご使用時のモードです。

LIPSモードは、LIPSⅢ又はLIPSⅣでご使用になれます。

LIPSの詳細は、別売りのソフトウェア概要書、コマンドリファレンスマニュアルをご参照下さい。

3.2 CODE 39の印刷サンプル例

例 1

LIPSモードで、「*1234567890*」のCODE 39のバーコードとOCR_B文字でコード番号を印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****  
//  
// 関数 : OutPutStuff ( HANDLE)  
//  
// 目的 : PASSTHROUGHでプリンタに直接コマンドを送信する。  
//  
//*****  
void OutPutStuff ( HANDLE hDC) {  
  
    COM          Com ;                                //Escape 出力構造体  
    short Count ;                                    //ループカウンタ  
    char *TextMode = "%x1E] p%x1E" ;                //ピッチ終了&テキストモード移行命令  
    char *CrLf1 = "%xD%xA" ;                        //改行命令  
    char *CrLf2 = "%x1B%x4B" ;                      //半改行  
    char *MojiSet1 = "%x1BPzBARcode39. BAR%x1B%x" ; //文字セット登録命令  
    char *MojiSet2 = "%x1BPzOCR-B. OCR_B%x1B%x" ;  //文字セット登録命令  
    char *CodeNumber = "*1234567890*" ;            //コード番号  
    char *KaiPage = "%xC" ;                         //改ページ  
    char *VectorMode = "%x1B [0&] %" ;              //文字サイズ指定  
    char *TextStart = "%x1B%@" ;                   //テキストモード開始  
    char *ShiftIn = "%x0F" ;                        //シフトイン  
    char *MojiSize1 = "%x1B [240 C" ;               //文字サイズ指定  
    char *MojiSize2 = "%x1B [100 C" ;              //文字サイズ指定  
    char *SizeTani = "%x1B [2 ; 300 l" ;           //文字サイズ単位指定  
    char *MojiPicchi1 = "%x1B [ ? 1428 K" ;        //文字ピッチ指定  
    char *MojiPicchi2 = "%x1B [ ? 1000 K" ;        //文字ピッチ指定  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) TextMode) ;  
    Com.l = strlen (TextMode) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //ピッチ終了&テキストモード移行  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) TextStart) ;  
    Com.l = strlen (TextStart) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //テキストモード開始  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) ShiftIn) ;  
    Com.l = strlen (ShiftIn) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //シフトイン  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSet1) ;  
    Com.l = strlen (MojiSet1) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字セット登録  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) SizeTani) ;  
    Com.l = strlen (SizeTani) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字サイズ単位指定  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiPicchi1) ;  
    Com.l = strlen (MojiPicchi1) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字ピッチ指定  
  
    lstrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSize1) ;  
    Com.l = strlen (MojiSize1) ;  
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字サイズ指定
```

```

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) CrLf1);
Com.l = strlen (CrLf1);
for (Count = 0 ; Count < 4 ; Count++) { //4 行改行
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL);
}

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) CodeNumber);
Com.l = strlen (CodeNumber);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //コード番号

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) CrLf1);
Com.l = strlen (CrLf1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //改行

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) CrLf2);
Com.l = strlen (CrLf2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //半改行

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSet2);
Com.l = strlen (MojiSet2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字セット登録

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ単位指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiPicchi2);
Com.l = strlen (MojiPicchi2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピッチ指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSize2);
Com.l = strlen (MojiSize2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) CodeNumber);
Com.l = strlen (CodeNumber);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //コード番号

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) KaiPage);
Com.l = strlen (KaiPage);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //改ページ

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) VectorMode);
Com.l = strlen (VectorMode);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ベクタモード移行
}

```

印刷結果



1234567890

例 2

LIPSモードで、「*1234567890*」のCODE 39のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** CODE 39 LIPSモード 印刷 *****
20 LPRINT CHR$(&H1B);CHR$(&H25);CHR$(&H40);          'テキスト開始
30                                                    'ジョブ開始命令
40 LPRINT CHR$(&H1B);CHR$(&H50);"31";CHR$(&H3B);"300";CHR$(&H3B);"1";CHR$(&H4A);CHR$(&H1B);CHR$(&H5C);
50 LPRINT CHR$(&H1B);CHR$(&H5B);"0";CHR$(&H22);"p";
60 LPRINT CHR$(&H1B);CHR$(&H3C);                          'ソフトレット
70 LPRINT CHR$(&H0F);                                    'シフトイン
80 LPRINT CHR$(&H1B);"PzBARcode39.BAR";CHR$(&H1B);"¥";  '文字セット名称選択
90                                                    'サイズ単位選択
100 LPRINT CHR$(&H1B);CHR$(&H5B);"2";CHR$(&H3B);"300";" |";
110 LPRINT CHR$(&H1B);CHR$(&H5B);"?1428";"K";           '文字ピッチ指定
120 LPRINT CHR$(&H1B);CHR$(&H5B);"240";"C";           '文字サイズ指定
130 LPRINT CHR$(&HD);CHR$(&HA);                          '改行
140 LPRINT CHR$(&HD);CHR$(&HA);                          '改行
150 LPRINT CHR$(&HD);CHR$(&HA);                          '改行
160 LPRINT CHR$(&HD);CHR$(&HA);                          '改行
170 LPRINT "*1234567890*";                               'バーコード印字
180 LPRINT CHR$(&HD);CHR$(&HA);                          '改行
190 LPRINT CHR$(&H1B);CHR$(&H4B);                        '半改行
200 REM ***** OCR_B 文字 印刷 *****
210 LPRINT CHR$(&H1B);"PzOCR-B.OCR_B";CHR$(&H1B);"¥";  '文字セット名称選択
220                                                    'サイズ単位選択
230 LPRINT CHR$(&H1B);CHR$(&H5B);"2";CHR$(&H3B);"300";" |";
240 LPRINT CHR$(&H1B);CHR$(&H5B);"?1000";"K";          '文字ピッチ指定
250 LPRINT CHR$(&H1B);CHR$(&H5B);"100";"C";            '文字サイズ指定
260 LPRINT "*1234567890*";                               'コード番号印字
270 LPRINT CHR$(&HC);                                    '改ページ
280                                                    'ジョブ終了命令
290 LPRINT CHR$(&H1B);CHR$(&H50);CHR$(&H30);CHR$(&H4A);CHR$(&H1B);CHR$(&H5C);
300 END

```

印刷結果



1234567890

例 3

ESC/Pのエミュレーションモードで、3ドットモジュール、高さ100ドットの「*1234567890*」のCODE39のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** CODE39 ESC/Pエミュレーションモード 印刷 *****
20 LPRINT CHR$ (&HD); CHR$ (&HA); ' 改行
30 LPRINT CHR$ (&HD); CHR$ (&HA); ' 改行
40 LPRINT CHR$ (&HD); CHR$ (&HA); ' 改行
50 LPRINT CHR$ (&HD); CHR$ (&HA); ' 改行
60 ' 文字セット名称選択 (印字)
70 LPRINT CHR$ (&H1B); CHR$ (&H7C); CHR$ (&H20); CHR$ (&H00); CHR$ (&H00); CHR$ (&H66);
"C3100.BAR,100,100,000,00012*1234567890*";
80 LPRINT CHR$ (&HD); CHR$ (&HA); ' 改行
90 REM ***** OCR_B 文字 印刷 *****
100 ' 文字セット名称選択 (印字)
110 LPRINT CHR$ (&H1B); CHR$ (&H7C); CHR$ (&H20); CHR$ (&H00); CHR$ (&H00); CHR$ (&H66);
"A2032.OCR_B,100,100,000,00012*1234567890*";
120 LPRINT CHR$ (&HC); ' 改ページ
130 END

```

印刷結果



3.3 NW-7の印刷サンプル例

例 1

LIPSモードで、「A1234567890A」のNW-7のバーコードとOCR_B文字でコード番号を印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****
//
// 関数： OutPutStuff ( HANDLE)
//
// 目的： PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff ( HANDLE hDC) {

COM Com ; //Escape出力構造体
short Count ; //ループカウンタ
char *TextMode = "%x1E] p%x1E" ; //ビッチ終了&テキストモード移行命令
char *CrLf1 = "%xD%xA" ; //改行命令
char *CrLf2 = "%x1B%x4B" ; //半改行
char *MojiSet1 = "%x1BPzBARnw7. BAR%x1B%x" ; //文字セット登録命令
char *MojiSet2 = "%x1BPzOCR-B. OCR_B%x1B%x" ; //文字セット登録命令
char *CodeNumber = "A1234567890A" ; //コード番号
char *KaiPage = "%xC" ; //改ページ
char *VectorMode = "%x1B [0&] %" ; //ベクタモード移行命令
char *TextStart = "%x1B%@" ; //テキストモード開始
char *ShiftIn = "%x0F" ; //シフトイン
char *MojiSize1 = "%x1B [240 C" ; //文字サイズ指定
char *MojiSize2 = "%x1B [100 C" ; //文字サイズ指定
char *SizeTani = "%x1B [2; 300 I" ; //文字サイズ単位指定
char *MojiPicchi1 = "%x1B [? 0 K" ; //文字ピッチ指定
char *MojiPicchi2 = "%x1B [? 1000 K" ; //文字ピッチ指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) TextMode) ;
Com.l = strlen (TextMode) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //ビッチ終了&テキストモード移行

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) TextStart) ;
Com.l = strlen (TextStart) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //テキストモード開始

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) ShiftIn) ;
Com.l = strlen (ShiftIn) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //シフトイン

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSet1) ;
Com.l = strlen (MojiSet1) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字セット登録

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) SizeTani) ;
Com.l = strlen (SizeTani) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字サイズ単位指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiPicchi1) ;
Com.l = strlen (MojiPicchi1) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字ピッチ指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSize1) ;
Com.l = strlen (MojiSize1) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字サイズ指定
```

```

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CrLf1);
Com.l = strlen (CrLf1);
for (Count = 0; Count < 4; Count++) { //4行改行
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL);
}

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CodeNumber);
Com.l = strlen (CodeNumber);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //コード番号

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CrLf1);
Com.l = strlen (CrLf1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //改行

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CrLf2);
Com.l = strlen (CrLf2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //半改行

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSet2);
Com.l = strlen (MojiSet2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi2);
Com.l = strlen (MojiPicchi2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピッチ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize2);
Com.l = strlen (MojiSize2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CodeNumber);
Com.l = strlen (CodeNumber);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //コード番号

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) KaiPage);
Com.l = strlen (KaiPage);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //改ページ

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) VectorMode);
Com.l = strlen (VectorMode);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ベクタモード移行
}

```

印刷結果



A1234567890A

例 3

ESC/Pのエミュレーションモードで、3ドットモジュール、高さ100ドットの「A1234567890A」のNW-7のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** NW-7 ESC/Pエミュレーションモード 印刷 *****
20 LPRINT CHR$ (&HD) ; CHR$ (&HA) ; ' 改行
30 LPRINT CHR$ (&HD) ; CHR$ (&HA) ; ' 改行
40 LPRINT CHR$ (&HD) ; CHR$ (&HA) ; ' 改行
50 LPRINT CHR$ (&HD) ; CHR$ (&HA) ; ' 改行
60 ' 文字セット名称選択 (印字)
70 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H20) ; CHR$ (&H00) ; CHR$ (&H00) ; CHR$ (&H66) ;
"N3100.BAR,100,100,000,00012A1234567890A";
80 LPRINT CHR$ (&HD) ; CHR$ (&HA) ; ' 改行
90 REM ***** OCR_B 文字 印刷 *****
100 ' 文字セット名称選択 (印字)
110 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H20) ; CHR$ (&H00) ; CHR$ (&H00) ; CHR$ (&H66) ;
"A2032.OCR_B,100,100,000,00012A1234567890A";
120 LPRINT CHR$ (&HC) ; ' 改ページ
130 END

```

印刷結果



3.4 JAN、EAN、UPCの印刷方法

JAN、EAN、UPCのバーコードを印字する為には、JAN、EAN、UPCそれぞれの仕様にあったバーコードフォントの文字セットを選択してバーコードを印刷します。

本フォントROMでは、JAN、EAN、UPC用のバーコードの文字セットが内蔵されていますので、バーコードの仕様にあわせた文字セットを選択する事で、確実にバーコードが印刷できます。(文字セットの切り換えによる印字)

さらに、印刷するバーコードの文字コードを本フォントカード用の文字コードに置き換える事によって1つの文字セットだけで印刷する事もできます。(文字コードの置き換えによる印字)

文字コードの置き換えによる印字の方が文字セットの切り換えによる印字よりも制御コマンドを少なくして印字できます。

特にエミュレーションモードで印字する場合には、制御の方法がより簡単に行う事ができます。

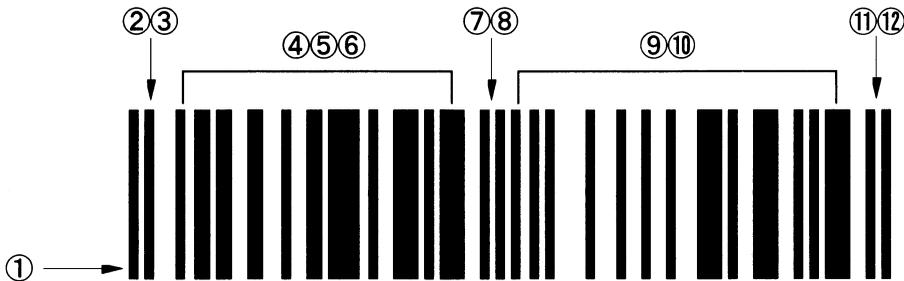
尚、LIPSモードとエミュレーションモードでは、同じバーコードを印刷する場合でも、印刷する為の制御方法が異なりますのでご注意ください。

1 文字セットの切り換えによる印刷方法

JAN、EAN、UPCコードのデータキャラクタは0～9までですが、それぞれのデータパターンA～D（15ページ参照）にしたがって、文字セットを切り換える必要があります。これは、データパターンが異なる場合、同じコードでもバーコードのデザインが違うためです。たとえばJAN8桁は、1桁目（パターンA）の位置での“0”と5桁目（パターンC）の位置での“0”とでは、バーコードデザインが異なります。以下の方法は、それぞれのデータパターンA～Dに該当する文字セットを切り換えながら印刷します。

1-1 LIPSモードでの印刷方法

以下の手順でプリンタに送る事によって印刷できます。



①	印字開始位置CAPの設定（行桁モードにて、CAPの移動を行います。または、サイズ単位モード設定、サイズ単位の選択、垂直水平絶対位置移動命令等のサイズ単位モードにてCAPの移動を行います。）
②	ガードバー用の文字セット（BARjan-D、BAR）を文字セット名称選択命令にて選択します。
③	“L”の文字を送ります。
④	左側のデータキャラクタパターン用の文字セット（BARjan-A、BARまたはBARjan-B、BAR）を文字セット名称選択命令にて選択します。
⑤	左側のデータキャラクタを送ります。
⑥	④⑤の組み合わせで左側のデータキャラクタパターンを印刷します。
⑦	ガードバー用の文字セットを文字セット名称選択命令にて選択します。
⑧	“M”の文字を送ります。
⑨	右側のデータキャラクタパターンの文字セットを文字セット名称選択命令にて選択します。
⑩	右側のデータキャラクタを送ります。
⑪	ガードバー用の文字セットを文字セット名称選択命令にて選択します。
⑫	“R”の文字を送ります。（UPC短縮バージョンの場合、“E”を送ります。）

1-2 各エミュレーションモードでの印刷

次の手順で、プリンタにコマンドを送る事によって印刷できます。

補足：②のコマンド体系にて、L I P Sの命令を各々プリンタに送ります。

L I P Sの命令は、L I P Sモードでの印刷方法をご参照下さい。

①	L I P Sモードのユーザーページ登録開始命令
②	イメージデータ転送命令 + L I P S命令
③	L I P Sモードのユーザーページ登録終了命令
④	ユーザーページ、オーバーレイ印字開始命令

④のコマンドの後、印字データをプリンタに送る事によってバーコードが印刷されます。

注意：エミュレーションモードとは、各オプションコントロールROMご使用時のモードです。

エミュレーションモードで印刷する場合には必ずバーコードフォント、OCR-Bフォントの印刷するページの先頭で行って下さい。(ページの途中で印刷する事はできません。) また、1ページに複数個のバーコードを印刷する場合には②のイメージデータ転送命令+L I P S命令で印刷する位置にCAPを移動してバーコードの印刷を行って下さい。

イメージデータの転送命令のパラメータのバイト数は必ずL I P S命令のコマンドのバイト数を指定して下さい。

バイト数の指定が誤っている場合には、正常にバーコードが印刷できなくなります。エミュレーションモードのコマンド(L I P Sモードのユーザーページ登録開始、終了命令、イメージデータの転送命令、ユーザーページオーバーレイ印字開始等)の詳細は各コントロールROMのユーザーズガイドをご参照下さい。

L I P Sの詳細は、別売りのソフトウェア概説書、コマンドリファレンスマニュアルをご参照下さい。

JAN、EAN、UPC印刷の文字セットの切り換えは、以下の組み合わせになります。

JAN、EAN、UPC標準バージョン（13桁）の場合

先頭1桁	文字セットの切り換えパターン	
0	DAAAAADCCCCCD	・・・UPCコード
1	DAABABBDCCCCCD	
2	DAABBABDCCCCCD	
3	DAABBBADCCCCCD	・・・JANコード
4	DABAABBDCCCCCD	
5	DABBAABDCCCCCD	
6	DABBBAADCCCCCD	
7	DABABABDCCCCCD	
8	DABABBADCCCCCD	
9	DABBABADCCCCCD	

JAN、EAN短縮バージョン（8桁）の場合

文字セットの切り換えパターン
DAAAADCCCCD

UPC短縮バージョンの場合

文字セットの切り換えパターン	末尾1桁（チェックデジット）
DBBBAAAD	0
DBBABAAD	1
DBBAABAD	2
DBBAAABD	3
DBABBAAD	4
DBAABBAD	5
DBAAABBD	6
DBABABAD	7
DBABAABD	8
DBAABABD	9

補足：A = J__ __ __ __ A. BAR 又は BARjan-A. BAR
 B = J__ __ __ __ B. BAR 又は BARjan-B. BAR
 C = J__ __ __ __ C. BAR 又は BARjan-C. BAR
 D = J__ __ __ __ D. BAR 又は BARjan-D. BAR
 の文字セットの略称です。

JAN、EAN、UPCの末尾がチェックデジットになっておりますが、チェックデジットが正しくないデータの場合、バーコードが読み取れないことがあります。
 正当なデータにするためにチェックデジットを付加する時は、ユーザーアプリケーションで行って下さい。
 チェックデジットの算出方法は、付録JANコードチェックデジット計算方法のページをご参照下さい。

2 文字コードの切り換えによる印刷方法

JAN、EAN、UPCコードはデータパターンによって、同一コードでもバーコードのデザインが異なることは前項で説明しましたが、当フォントROMでは、異なるデータパターンのデザインを、別のコードとして1つの文字セット（J____A. BAR又はBARjan-A. BAR）に格納しています。たとえば、データパターンCでのコード“0”のデザインはコード“P”として格納しています。

以下の方法は、印字する文字コードを置き換えることにより、1つの文字セットで印字します。

2-1 LIPSモードでの印刷方法

以下の手順でプリンタに送る事によって印刷できます。

- ① 文字セット（BARjan-A. BAR）を文字セット名称選択命令にて選択します。
- ② JAN、EAN、UPC等印刷するバーコードに合わせて文字コードを指定します。
文字コードの置換方法は、17ページ以下をご参照下さい。

2-2 各種エミュレーションモードでの印刷方法

文字セット名称選択命令Iのコマンドを用いて印刷します。

注意：文字セット名称選択命令Iの各パラメータは、以下の様に設定して下さい。

文字セット名称 : J____A. BARの文字セット

横 拡 大 率 : 1/2～2倍

縦 拡 大 率 : 1/2～2倍

文 字 ピ ッ チ : 文字セット固有の文字ピッチ

但し、縦横の拡大率を1/2、2倍の組み合わせでの指定時はそれぞれ等倍になりますので、ご注意下さい。

文字コードのバイト数 : バーコードの文字コードのバイト数

文 字 コ ー ド : バーコードの文字コード

バーコードの文字コードは、JAN、EAN、UPC等によって文字コードを置換して指定する必要があります。

文字コードの置換方法は、17ページ以下をご参照下さい。

文字セット名称選択命令Iは、各コントロールROMのユーザズガイドをご参照下さい。

2-3 文字コードの置換

① JAN、EAN、UPC標準バージョンの場合

コード番号1桁目によって以下の文字コード置換パターンに従って文字コードを指定します。

先頭1桁	文字コードの置換パターン	
0	DLAAAAAA DM CCCCCCD _R	・・・UPCコード
1	DLAABABB DM CCCCCCD _R	
2	DLAABBAB DM CCCCCCD _R	
3	DLAABBBA DM CCCCCCD _R	
4	DLABAABB DM CCCCCCD _R	・・・JANコード
5	DLABBAAB DM CCCCCCD _R	
6	DLABBBAA DM CCCCCCD _R	
7	DLABABAB DM CCCCCCD _R	
8	DLABABBA DM CCCCCCD _R	
9	DLABBABA DM CCCCCCD _R	

解説：コード番号の1桁目によって2桁目から13桁目を上記の置換パターンで各桁のバーコードを指定します。

DL = L (文字コード4 C(h))レフトガードバー

DM = M (文字コード4 D(h))センターガードバー

DR = N (文字コード4 E(h))ライトガードバー

A = データキャラクタ 0～9 (文字コード3 0(h)～3 9(h))をそのまま指定します。

B = データキャラクタ 0～9 (文字コード3 0(h)～3 9(h))を@～I (文字コード4 0(h)～4 9(h))に置換して指定します。

C = データキャラクタ 0～9 (文字コード3 0(h)～3 9(h))をP～Y (文字コード5 0(h)～5 9(h))に置換して指定します。

尚、A、B、Cパターンの置換は、20ページ表1-1の置換表をご参照下さい。

② JAN、EAN短縮バージョンの場合

以下の文字コード置換パターンに従って文字コードを指定します。

文字コードの置換パターン
DLAAAADMCCCCDR

- 解説：DL = L (文字コード4 C(h))レフトガードバー
DM = M (文字コード4 D(h))センターガードバー
DR = N (文字コード4 E(h))ライトガードバー
A = データキャラクタ 0～9 (文字コード3 0(h)～3 9(h))を
そのまま指定します。
C = データキャラクタ 0～9 (文字コード3 0(h)～3 9(h))をP～Y
(文字コード5 0(h)～5 9(h))に置換
して指定します。
尚、A、Cパターンの置換は、20ページ表1-1の置換表をご参照下さい。

③ U P C短縮バージョンの場合

コード番号の末尾の番号によって、以下の文字コードの置換パターンに従って文字コードを指定します。

文字コードの置換パターン	末尾 (CD)
DL BBBAAA DE	0
DL BBABAA DE	1
DL BBAABA DE	2
DL BBAAAB DE	3
DL BABBAA DE	4
DL BAABBA DE	5
DL BAAABB DE	6
DL BABABA DE	7
DL BABAAB DE	8
DL BAABAB DE	9

CD = チェックデジット

解説：文字コードの末尾はチェックデジット（以下CD）となっています。

CDによって上記の文字コードの置換パターンに従って各桁の文字コードを指定します。UPC短縮のCDの計算方法は、付録UPC短縮のチェックデジット計算方法のページをご参照下さい。また、コード番号2桁目から、7桁目の6桁をバーコードとして印字、先頭1桁目と末尾のCDは、バーコードとして印字しません。

DL = L (文字コード4 C(h))

DE = 0 (文字コード4 F(h))

A = 文字コード 0～9 (文字コード3 0(h)～3 9(h))を
そのまま指定します。

B = 文字コード 0～9 (文字コード3 0(h)～3 9(h))を@～I
(文字コード4 0(h)～4 9(h))に置換
して指定します。

尚、A、Bパターンの置換は、20ページ表1-1の置換表をご参照下さい。

3.5 JAN、EAN、UPCの印刷サンプル例

文字セットの切り換え方法による印刷サンプル

例 1

LIPSモードで、文字セットの切り換えによる方法でJAN13桁のバーコードとOCR_B文字でコード番号を印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****
//
// 関数 : OutPutStuff ( HANDLE )
//
// 目的 : PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff ( HANDLE hDC ) {

COM Com ;

short Count ;
char *TextMode = "%x1E p%x1E" ;
char *CrLf = "%xDxA" ;
char *MojiSetA = "%x1BPzBARjan-A. BAR%x1B%Y" ;
char *MojiSetB = "%x1BPzBARjan-B. BAR%x1B%Y" ;
char *MojiSetC = "%x1BPzBARjan-C. BAR%x1B%Y" ;
char *MojiSetD = "%x1BPzBARjan-D. BAR%x1B%Y" ;
char *MojiSetO = "%x1BPzOCR-B. OCR_B%x1B%Y" ;
char *Moji_L = "L" ;
char *Moji_M = "M" ;
char *Moji_N = "N" ;
char *Moji_9 = "9" ;
char *Moji_1 = "1" ;
char *Moji_23 = "23" ;
char *Moji_45 = "45" ;
char *Moji_678904 = "678904" ;
char *CodeNumber = "4912345678904" ;
char *KaiPage = "%C" ;
char *VectorMode = "%x1B [0&" ;
char *TextStart = "%x1B%@" ;
char *ShiftIn = "%xOF" ;
char *MojiSize1 = "%x1B [240 C" ;
char *MojiSize2 = "%x1B [100 C" ;
char *SizeTani = "%x1B [2;300 I" ;
char *MojiPicchi1 = "%x1B [?0 K" ;
char *MojiPicchi2 = "%x1B [?1428 K" ;
char *MojiPicchi3 = "%x1B [?1000 K" ;

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) TextMode) ;
Com.l = strlen (TextMode) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //ピッチ終了&テキストモード移行

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) TextStart) ;
Com.l = strlen (TextStart) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //テキストモード開始

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) ShiftIn) ;
Com.l = strlen (ShiftIn) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //シフトイン

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSetD) ;
Com.l = strlen (MojiSetD) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //カートバー用文字セット登録

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) SizeTani) ;
Com.l = strlen (SizeTani) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字サイズ単位指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiPicchi1) ;
Com.l = strlen (MojiPicchi1) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字ピッチ指定

Istrcpy ((LPSTR) &Com.com [0] , (LPCSTR) MojiSize1) ;
Com.l = strlen (MojiSize1) ;
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL) ; //文字サイズ指定

//Escape 出力構造体
//レフカク
//ピッチ終了&テキストモード移行命令
//改行命令
//文字セット登録命令
//文字セット登録命令
//文字セット登録命令
//文字セット登録命令
//カートバー
//カートバー
//カートバー
//バーコード印字文字
//バーコード印字文字
//バーコード印字文字
//バーコード印字文字
//コード番号
//改ページ
//ベクタモード移行命令
//テキストモード開始
//シフトイン
//文字サイズ指定
//文字サイズ指定
//文字サイズ単位指定
//文字ピッチ指定
//文字ピッチ指定
//文字ピッチ指定
```

```

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CrLf);
Com.l = strlen (CrLf);
for (Count = 0; Count < 4; Count++) { //4行改行
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL);
}

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_L);
Com.l = strlen (Moji_L);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //カ-ド`ハ`-印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetA);
Com.l = strlen (MojiSetA);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //左側用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi1);
Com.l = strlen (MojiPicchi1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピク`チ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_9);
Com.l = strlen (Moji_9);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ハ`-コ-ド`印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetB);
Com.l = strlen (MojiSetB);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //左側用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi2);
Com.l = strlen (MojiPicchi2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピク`チ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_1);
Com.l = strlen (Moji_1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ハ`-コ-ド`印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetA);
Com.l = strlen (MojiSetA);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //左側用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi1);
Com.l = strlen (MojiPicchi1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピク`チ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_23);
Com.l = strlen (Moji_23);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ハ`-コ-ド`印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetB);
Com.l = strlen (MojiSetB);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //左側用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ`単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi2);
Com.l = strlen (MojiPicchi2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピク`チ指定

```

```

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_45);
Com.l = strlen (Moji_45);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // ハ'-コト' 印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetD);
Com.l = strlen (MojiSetD);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // カ'-ト'ハ'-用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi1);
Com.l = strlen (MojiPicchi1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字ピクチャ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_M);
Com.l = strlen (Moji_M);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // カ'-ト'ハ'-印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetC);
Com.l = strlen (MojiSetC);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 右側用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi2);
Com.l = strlen (MojiPicchi2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字ピクチャ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_678904);
Com.l = strlen (Moji_678904);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // ハ'-コト' 印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSetD);
Com.l = strlen (MojiSetD);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // カ'-ト'ハ'-用文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi1);
Com.l = strlen (MojiPicchi1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字ピクチャ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize1);
Com.l = strlen (MojiSize1);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_N);
Com.l = strlen (Moji_N);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // カ'-ト'ハ'-印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CrLf);
Com.l = strlen (CrLf);
for (Count = 0; Count < 2; Count++) { // 2行改行
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL);
}

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSet0);
Com.l = strlen (MojiSet0);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ 単位指定

```

```
Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi3);  
Com.l = strlen (MojiPicchi3);  
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字ビッパ指定  
  
Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize2);  
Com.l = strlen (MojiSize2);  
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 文字サイズ指定  
  
Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CodeNumber);  
Com.l = strlen (CodeNumber);  
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // コード番号  
  
Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) KaiPage);  
Com.l = strlen (KaiPage);  
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // 改ページ  
  
Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) VectorMode);  
Com.l = strlen (VectorMode);  
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); // ベクタモード移行
```

印刷結果



4912345678904

例 2

LIPSモードで、文字セットの切り換えによる方法でJAN13桁のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** JAN 13桁 LIPSモード 印刷 *****
20 LPRINT CHR$ (&H1B); CHR$ (&H25); CHR$ (&H40);          ; 待機モード 開始
30 ; ショット開始命令
40 LPRINT CHR$ (&H1B); CHR$ (&H50); "31"; CHR$ (&H3B); "300"; CHR$ (&H3B); "1"; CHR$ (&H4A); CHR$ (&H1B);
CHR$ (&H5C);
50 LPRINT CHR$ (&H1B); CHR$ (&H5B); "0"; CHR$ (&H22); "p";
60 LPRINT CHR$ (&H1B); CHR$ (&H3C); ; ソフトリフト
70 LPRINT CHR$ (&H0F); ; シフトイン
80 LPRINT CHR$ (&HD); CHR$ (&HA); ; 改行
90 LPRINT CHR$ (&HD); CHR$ (&HA); ; 改行
100 LPRINT CHR$ (&HD); CHR$ (&HA); ; 改行
110 LPRINT CHR$ (&HD); CHR$ (&HA); ; 改行
120 LPRINT CHR$ (&H1B); "PzBARjan-D. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
130 ; サイズ単位選択
140 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
150 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?0"; "K"; ; 文字ビッパ指定
160 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
170 LPRINT "L"; ; ガートバー印字
180 LPRINT CHR$ (&H1B); "PzBARjan-A. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
190 ; サイズ単位選択
200 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
210 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?0"; "K"; ; 文字ビッパ指定
220 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
230 LPRINT "9"; ; ハーコード印字
240 LPRINT CHR$ (&H1B); "PzBARjan-B. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
250 ; サイズ単位選択
260 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
270 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?1428"; "K"; ; 文字ビッパ指定
280 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
290 LPRINT "1"; ; ハーコード印字
300 LPRINT CHR$ (&H1B); "PzBARjan-A. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
310 ; サイズ単位選択
320 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
330 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?0"; "K"; ; 文字ビッパ指定
340 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
350 LPRINT "23"; ; ハーコード印字
360 LPRINT CHR$ (&H1B); "PzBARjan-B. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
370 ; サイズ単位選択
380 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
390 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?1428"; "K"; ; 文字ビッパ指定
400 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
410 LPRINT "45"; ; ハーコード印字
420 LPRINT CHR$ (&H1B); "PzBARjan-D. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
430 ; サイズ単位選択
450 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
460 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?0"; "K"; ; 文字ビッパ指定
470 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
480 LPRINT "M"; ; ガートバー印字
490 LPRINT CHR$ (&H1B); "PzBARjan-C. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
500 ; サイズ単位選択
510 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
520 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?1428"; "K"; ; 文字ビッパ指定
530 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
540 LPRINT "678904"; ; ハーコード印字
550 LPRINT CHR$ (&H1B); "PzBARjan-D. BAR"; CHR$ (&H1B); "¥"; ; 文字セット名称選択
560 ; サイズ単位選択
570 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; "1";
580 LPRINT CHR$ (&H1B); CHR$ (&H5B); "?0"; "K"; ; 文字ビッパ指定
590 LPRINT CHR$ (&H1B); CHR$ (&H5B); "240"; "C"; ; 文字サイズ指定
600 LPRINT "N"; ; ガートバー印字
610 LPRINT CHR$ (&HD); CHR$ (&HA); ; 改行
620 LPRINT CHR$ (&HD); CHR$ (&HA); ; 改行

```

```

630 REM      ***** OCR_B 文字 印刷 *****
640 LPRINT CHR$ (&H1B); "PzOCR-B. OCR_B"; CHR$ (&H1B); "*" ;
650
660 LPRINT CHR$ (&H1B); CHR$ (&H5B); "2"; CHR$ (&H3B); "300"; " 1";
670 LPRINT CHR$ (&H1B); CHR$ (&H5B); "? 1000"; " K";
680 LPRINT CHR$ (&H1B); CHR$ (&H5B); "100"; " C";
690 LPRINT "4912345678904";
700 LPRINT CHR$ (&HC);
710
720 LPRINT CHR$ (&H1B); CHR$ (&H50); CHR$ (&H30); CHR$ (&H4A); CHR$ (&H1B); CHR$ (&H5C);
730 END

```

```

' 文字セット名称選択
' サイズ単位選択
' 文字ピッチ指定
' 文字サイズ指定
' コード番号印字
' 改ページ
' ショフ終了命令

```

印刷結果



4912345678904

例 3

ESC/Pのエミュレーションモードで、3ドットモジュール、高さ100ドットの文字セットの切り換えによる方法でJAN13桁のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** JAN 13桁 ESC/Pエミュレーションモード 印刷 *****
20 ' LIPSの制御命令によるページ登録開始
30 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H27) ; CHR$ (&H00) ; CHR$ (&H01) ; CHR$ (&H30) ;
40 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H08) ; CHR$ (&H00) ;
50 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
60 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
70 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
80 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
90 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H10) ; CHR$ (&H00) ;
100 LPRINT CHR$ (&H1B) ; "PzJ3100D.BAR" ; CHR$ (&H1B) ; "¥" ;
110 LPRINT "L" ;
120 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H10) ; CHR$ (&H00) ;
130 LPRINT CHR$ (&H1B) ; "PzJ3100A.BAR" ; CHR$ (&H1B) ; "¥" ;
140 LPRINT "9" ;
150 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H10) ; CHR$ (&H00) ;
160 LPRINT CHR$ (&H1B) ; "PzJ3100B.BAR" ; CHR$ (&H1B) ; "¥" ;
170 LPRINT "1" ;
180 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H11) ; CHR$ (&H00) ;
190 LPRINT CHR$ (&H1B) ; "PzJ3100A.BAR" ; CHR$ (&H1B) ; "¥" ;
200 LPRINT "23" ;
210 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H11) ; CHR$ (&H00) ;
220 LPRINT CHR$ (&H1B) ; "PzJ3100B.BAR" ; CHR$ (&H1B) ; "¥" ;
230 LPRINT "45" ;
240 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H10) ; CHR$ (&H00) ;
250 LPRINT CHR$ (&H1B) ; "PzJ3100D.BAR" ; CHR$ (&H1B) ; "¥" ;
260 LPRINT "M" ;
270 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H15) ; CHR$ (&H00) ;
280 LPRINT CHR$ (&H1B) ; "PzJ3100C.BAR" ; CHR$ (&H1B) ; "¥" ;
290 LPRINT "678904" ;
300 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H10) ; CHR$ (&H00) ;
310 LPRINT CHR$ (&H1B) ; "PzJ3100D.BAR" ; CHR$ (&H1B) ; "¥" ;
320 LPRINT "N" ;
330 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H04) ; CHR$ (&H00) ;
340 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
350 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ;
360 REM ***** OCR_B文字 印刷 *****
370 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H1D) ; CHR$ (&H00) ;
380 LPRINT CHR$ (&H1B) ; "PzA2032.OCR_B" ; CHR$ (&H1B) ; "¥" ;
390 LPRINT "4912345678904" ;
400 ' LIPSの制御命令によるページ終了開始
410 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H2E) ; CHR$ (&H00) ; CHR$ (&H00) ;
420 ' ユーザーページ登録開始
430 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H68) ; CHR$ (&H00) ; CHR$ (&H01) ; CHR$ (&H30) ;
440 END

```

印刷結果

※ LIPSモードのユーザーページ機能を使用する場合には、プリンタパネルから [ESC/Pの設定] - [LIPS7オプション] の設定で "LIPS4" を選択しておく必要があります。



4912345678904


```

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_L);
Com.l = strlen (Moji_L);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //レフト'ト'ハ'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_9);
Com.l = strlen (Moji_9);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //Aタイプ'ハ'コート'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_A);
Com.l = strlen (Moji_A);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //Bタイプ'ハ'コート'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_23);
Com.l = strlen (Moji_23);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //Aタイプ'ハ'コート'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_DE);
Com.l = strlen (Moji_DE);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //Bタイプ'ハ'コート'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_M);
Com.l = strlen (Moji_M);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //センサ'ト'ハ'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_VWXYPT);
Com.l = strlen (Moji_VWXYPT);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //Cタイプ'ハ'コート'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) Moji_N);
Com.l = strlen (Moji_N);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ライト'ト'ハ'印字

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CrLf1);
Com.l = strlen (CrLf1);
for (Count = 0; Count < 2; Count++) { //2行改行
    Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL);
}

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSet2);
Com.l = strlen (MojiSet2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字セット登録

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) SizeTani);
Com.l = strlen (SizeTani);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ'単位指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiPicchi2);
Com.l = strlen (MojiPicchi2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字ピッチ指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) MojiSize2);
Com.l = strlen (MojiSize2);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //文字サイズ'指定

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) CodeNumber);
Com.l = strlen (CodeNumber);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //コート'番号

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) KaiPage);
Com.l = strlen (KaiPage);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //改'ページ'

Istrcpy ((LPSTR) &Com.com [0], (LPCSTR) VectorMode);
Com.l = strlen (VectorMode);
Escape (hDC, PASSTHROUGH, NULL, (LPSTR) &Com, NULL); //ハ'カモード'移行

```

印刷結果



4912345678904

例 2

LIPSモードで、文字コードの置換による方法でJAN13桁のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** JAN 13桁 LIPSモード 印刷 *****
20 LPRINT CHR$ (&H1B) ; CHR$ (&H25) ; CHR$ (&H40) ;
30
40 LPRINT CHR$ (&H1B) ; CHR$ (&H50) ; "31" ; CHR$ (&H3B) ; "300" ; CHR$ (&H3B) ; "1" ; CHR$ (&H4A) ; CHR$ (&H1B) ;
CHR$ (&H5C) ;
50 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "0" ; CHR$ (&H22) ; "p" ;
60 LPRINT CHR$ (&H1B) ; CHR$ (&H3C) ;
70 LPRINT CHR$ (&H0F) ;
80 LPRINT CHR$ (&H1B) ; "PzBARjan-A. BAR" ; CHR$ (&H1B) ; "¥" ;
90
100 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "2" ; CHR$ (&H3B) ; "300" ; "1" ;
110 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "? 0" ; " K" ;
120 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "240" ; " C" ;
130 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
140 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
150 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
160 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
170 LPRINT "L" ;
190 LPRINT "9" ;
200 LPRINT "A" ;
210 LPRINT "23" ;
230 LPRINT "DE" ;
240 LPRINT "M" ;
250 LPRINT "VWXYPT" ;
260 LPRINT "N" ;
270 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
280 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
290 REM ***** OCR_B 文字 印刷 *****
300 LPRINT CHR$ (&H1B) ; "PzOCR-B. OCR_B" ; CHR$ (&H1B) ; "¥" ;
310
320 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "2" ; CHR$ (&H3B) ; "300" ; "1" ;
330 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "? 1000" ; " K" ;
340 LPRINT CHR$ (&H1B) ; CHR$ (&H5B) ; "100" ; " C" ;
350 LPRINT "4912345678904" ;
360 LPRINT CHR$ (&HC) ;
370
380 LPRINT CHR$ (&H1B) ; CHR$ (&H50) ; CHR$ (&H30) ; CHR$ (&H4A) ; CHR$ (&H1B) ; CHR$ (&H5C) ;
390 END

```

' テキストモード 開始
 ' ショボ 開始命令
 ' ソフトリセット
 ' シフトイン
 ' 文字セット名称選択
 ' サイズ 単位選択
 ' 文字ピッチ指定
 ' 文字サイズ 指定
 ' 改行
 ' 改行
 ' 改行
 ' 改行
 ' 改行
 ' リフトガードハ-印字
 ' Aタイプハ-コード 印字
 ' Bタイプハ-コード 印字
 ' Aタイプハ-コード 印字
 ' Bタイプハ-コード 印字
 ' センターハ-印字
 ' Cタイプハ-コード 印字
 ' ライトハ-印字
 ' 改行
 ' 改行
 ' 文字セット名称選択
 ' サイズ 単位選択
 ' 文字ピッチ指定
 ' 文字サイズ 指定
 ' コード 番号印字
 ' 改ページ
 ' ショボ 終了命令

印刷結果



4912345678904

例 3

ESC/Pのエミュレーションモードで、3ドットモジュール、高さ100ドットの文字コードの置換による方法でJAN 13桁のバーコードとOCR_B文字でコード番号を印刷する場合。

サンプルプログラム (BASIC)

```

10 REM ***** JAN 13桁 ESC/Pエミュレーションモード 印刷 *****
20                                     'LIPSの制御命令によるページ登録開始
30 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H27) ; CHR$ (&H00) ; CHR$ (&H01) ; CHR$ (&H30) ;
40 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H08) ; CHR$ (&H00) ;
50 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
60 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
70 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
80 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
90 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H1E) ; CHR$ (&H00) ;
100 LPRINT CHR$ (&H1B) ; "PzJ3100A.BAR" ; CHR$ (&H1B) ; "¥" ;
110 LPRINT "L" ;
120 LPRINT "9" ;
130 LPRINT "A" ;
140 LPRINT "23" ;
150 LPRINT "DE" ;
160 LPRINT "M" ;
170 LPRINT "VWXYPT" ;
180 LPRINT "N" ;
190 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H04) ; CHR$ (&H00) ;
200 LPRINT CHR$ (&HD) ; CHR$ (&HA) ;
210 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ;
220 REM ***** OCR_B 文字 印刷 *****
230 LPRINT CHR$ (&H1B) ; CHR$ (&H4B) ; CHR$ (&H1D) ; CHR$ (&H00) ;
240 LPRINT CHR$ (&H1B) ; "PzA2032.OCR_B" ; CHR$ (&H1B) ; "¥" ;
250 LPRINT "4912345678904" ;
260                                     'コート番号印字
270 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H2E) ; CHR$ (&H00) ; CHR$ (&H00) ;
280                                     'LIPSの制御命令によるページ終了開始
290 LPRINT CHR$ (&H1B) ; CHR$ (&H7C) ; CHR$ (&H68) ; CHR$ (&H00) ; CHR$ (&H01) ; CHR$ (&H30) ;
300 END                                     'ユーザーページ登録開始

```

印刷結果



4912345678904

3.6 EAN-128の印刷方法 (CR-MENのみ)

本ROMには、「EAN-128」および「標準料金代理収納用EAN-128」のバーコードフォントが収録されています。

以下にそれぞれの特長を記載します。

(1) EAN-128

EAN-128は、Code128バーコードを使用し、複数の管理情報を「アプリケーション識別子 (AI)」によって区別できるようにしたバーコード体系です。

以下のように、アプリケーション識別子とデータの組み合わせから成り立ちます。



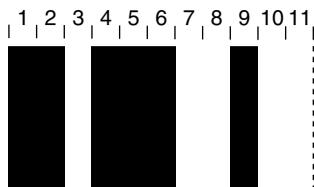
- ・ EAN-128のバーコードキャラクタ構成については、付録「6.2 フォント仕様一覧」を参照してください。
- ・ ストップキャラクタの直前に入れるチェックデジットは、「モジュラス103」を使用します。計算方法については、付録「6.5 EAN-128チェックデジット計算方法」を参照してください。また、このチェックデジットは目視文字には表示されません。
- ・ アプリケーション識別子 (AI) には様々な種類があります。詳しくは、付録「6.6 EAN-128アプリケーション識別子一覧」を参照してください。
- ・ 「FNC1」は、以下の位置に追加します。
 - スタートキャラクタの直後に追加。
 - データの桁数が可変長の場合、データの直後に追加。(データの桁数が可変長か固定長かは、アプリケーション識別子によって異なります)

注意：本ROMに収録されているバーコードフォントEAN-128は、48ポイントのサイズでの出力に適しています。48ポイント以外のサイズで出力したバーコードの読み取りは保証しません。

■ バーコードキャラクタについて

EAN-128のバーコードキャラクタは、黒バー3本と白バー3本で構成されています。

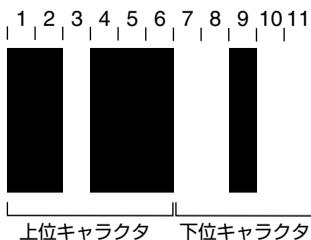
例) データキャラクタ"5"の場合



ただし、ストップキャラクタのみ、黒バー4本と白バー3本で構成されています。



本ROMでは、これらのキャラクタを上位キャラクタと下位キャラクタに分けて表現しています。



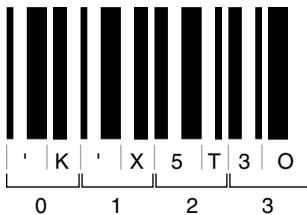
- ・上位キャラクタは、黒・白・黒のバーの組み合わせです。
(バーコードキャラクター一覧では、BSBと表記しています。)
- ・下位キャラクタは、白・黒・白のバーの組み合わせです。
(バーコードキャラクター一覧では、SBSと表記しています。)

たとえば、「0123」というバーコードを印刷したいときは、次のようになります。キャラクタマップおよびバーコードキャラクタ構成については、付録「6.2 フォント仕様一覧」を参照してください

- ・コードセット"A"およびコードセット"B"の場合

コード	キャラクタマップ		バーコードキャラクタ	
	上位	下位	上位	下位
0	'	K		
1	'	X		
2	5	T		
3	3	O		

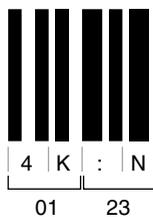
したがって、「0123」は「'K'X5T3O」とマッピングされ以下のように印字されます。



- ・コードセット"C"の場合

コード	キャラクタマップ		バーコードキャラクタ	
	上位	下位	上位	下位
01	4	K		
23	:	N		

したがって、「0123」は「4K:N」とマッピングされ以下のように印字されます。



■ 代表的なアプリケーション識別子を使った出力例

アプリケーション識別子	内容	桁数
00	標準カートンID	数字18桁

「00」は、ダンボールやパレットなど輸送単位に対して梱包シリアル番号を意味付ける識別子です。

データは18桁で以下のような構成になります。

梱包タイプ	共通企業コード	出荷単位ごとの梱包シリアル番号	チェックデジット
1桁	7桁	9桁	1桁

- ・「梱包タイプ」には、以下の数字を入力します。

数字	梱包タイプ
0	ケースまたはカートン
1	パレット（ケース、カートンよりも大きいもの）
2	コンテナ（パレットよりも大きいもの）
3	上記以外の梱包タイプ
4	内部規定による（社内用途）
5	取引企業間の相互規定による
6～9	使用禁止

- ・「共通企業コード」には、「国コード」 + 「メーカーコード」を入力します。
- ・「チェックデジット」は、「モジュラス10/3ウェイト」を使用します。計算方法については、付録「6.3 JANコードチェックデジット計算方法」を参照してください。

例)

アプリケーション識別子	00
梱包タイプ	1
共通企業コード	4912345
梱包シリアル番号	000012345
計算後のチェックデジット	6

↓順に並べる

00149123450000123456

↓キャラクタマップ変換 (コードセット"C")

OY&]BJ:N#LOYOY"^)L?J

↓以下のコードを先頭に追加

- ・スタートキャラクタコード (スタートC) : /^
- ・FNC1 : AN

/^ANOY&]BJ:N#LOYOY"^)L?J

↓以下のコードを末尾に追加

- ・チェックキャラクタコード* : 4X
- ・ストップキャラクタコード : {

/^ANOY&]BJ:N#LOYOY"^)L?J4X{

↓目視文字用の文字列を追加

/^ANOY&]BJ:N#LOYOY"^)L?J4X{

(00)149123450000123456

↓それぞれ文字セットを指定してバーコードを印刷

- ・バーコード : CI_CODE128.BAR
- ・目視文字 : OCR-B.OCR_B

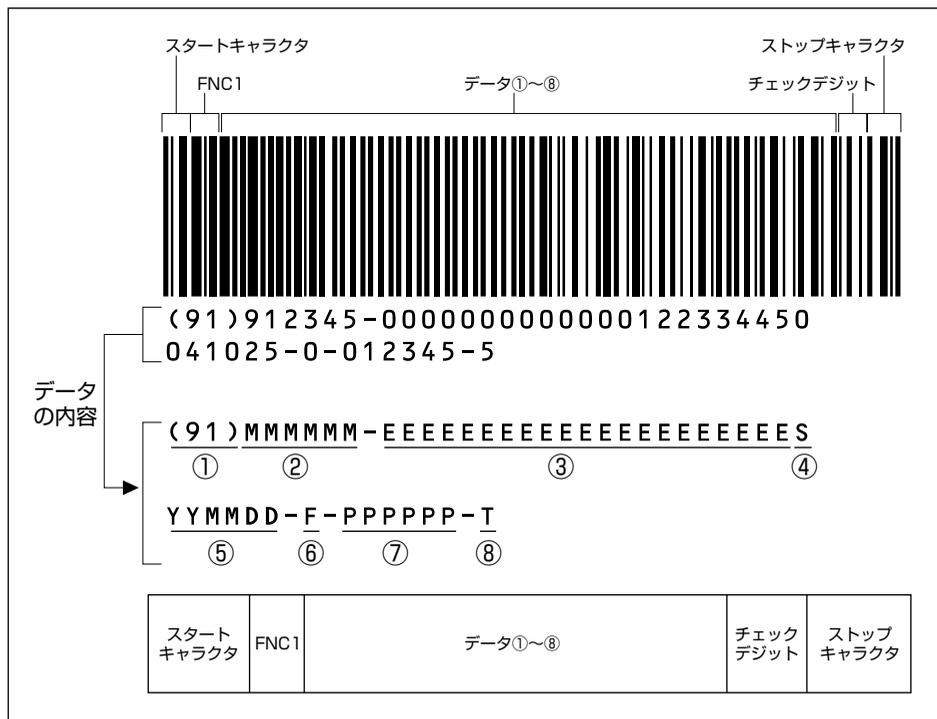


(00)149123450000123456

- * チェックキャラクタコードは、「モジュラス103」での計算結果より求めます。計算方法については、付録「6.5 EAN-128チェックデジット計算方法」を参照してください。

(2) 標準料金代理収納用EAN-128

標準料金代理収納用EAN-128は、バーコードとしてEAN-128を用い、「標準料金代理収納ガイドライン」でコード体系を規定したバーコードです。以下のように決められたデータの並び順でバーコードを作成します。



	項目	内容	桁数
①	(91)	アプリケーション識別子 (91固定)	2
②	MMMMMM	国コードの2桁目 (9または5) + メーカーコード (5桁) 請求書を発行する企業を示し、「JANメーカーコード」(7桁)のうち、先頭の「4」を除いた6桁を使用する。 * 9桁の「JANメーカーコード」の付番を受けている企業は、「代理収納」のために7桁のJANメーカーコードを追加申請する必要があります。	6
③	EEEE・・・EEEEEE	自由使用欄 請求書を発行する企業が、顧客番号などで使用する。 21桁未満の場合は、数字「0」を先頭に追加して21桁にする。	21
④	S	再発行区分 (再発行回数) 初回発行を「0」とし、再発行回数を1桁の数字で表す。 例) 0:初回発行 1:1回目の再発行 2:2回目の再発行	1
⑤	YYMMDD	支払期限日 (西暦下2桁+月+日) この期限日まで、バーコード印刷された収納票を使用した支払いが可能。 請求書を発行する企業が支払期限日を設けない場合は、「999999」と表す。	6
⑥	F	印紙フラグ 請求金額に対して、収入印紙が必要か否かを1桁の数字で表す。 0:収入印紙不要 (貼らない) 1:200円の収入印紙が必要 (貼る) 2~9:リザーブ (新税対応等で使用)	1
⑦	PPPPPP	請求金額 (円単位) 顧客に請求する金額 (正味の金額) を6桁の数字で表す。6桁未満の場合は、数字「0」を先頭に追加して6桁にする。	6
⑧	T	チェックデジット 読み取りミスを防ぐための1桁の数値で、①から⑦までの43桁の数値に対して「モジュラス10/3ウェイト」の計算方式で算出する。計算方法については、付録「6.3 JANコードチェックデジット計算方法」を参照してください。	1

- ・ EAN-128のバーコードキャラクタ構成については、付録「6.2 フォント仕様一覧」を参照してください。
- ・ ストップキャラクタの直前に入れるチェックデジットは、「モジュラス103」を使用します。計算方法については、付録「6.5 EAN-128チェックデジット計算方法」を参照してください。また、このチェックデジットは目視文字には表示されません。
- ・ 目視文字の大きさは8ポイント (半角数字) 以上で印刷してください。

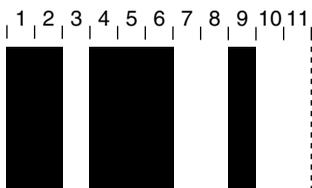
注意：本ROMに収録されているバーコードフォントEAN-128は、48ポイントのサイズでの出力に適しています。48ポイント以外のサイズで出力したバーコードの読み取りは保証しません。

バーコードを印刷するサイズは、幅：60mm以内、高さ：10mm以上が原則です。

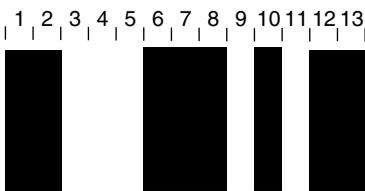
■ バーコードキャラクタについて

標準料金代理収納用EAN-128のバーコードキャラクタは、黒バー3本と白バー3本で構成されています。

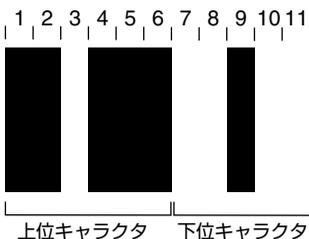
例) データキャラクタ"5"の場合



ただし、ストップキャラクタのみ、黒バー4本と白バー3本で構成されています。



本ROMでは、これらのキャラクタを上位キャラクタと下位キャラクタに分けて表現しています。



- ・上位キャラクタは、黒・白・黒のバーの組み合わせです。
(バーコードキャラクター一覧では、BSBと表記しています。)
- ・下位キャラクタは、白・黒・白のバーの組み合わせです。
(バーコードキャラクター一覧では、SBSと表記しています。)

たとえば、「0123」というバーコードを印刷したいときは、次のようになります。キャラクタマップおよびバーコードキャラクタ構成については、付録「6.2 フォント仕様一覧」を参照してください。

補足：標準料金代理収納用EAN-128は、コードセット"C"を使用します。

コード	キャラクタマップ		バーコードキャラクタ	
	上位	下位	上位	下位
01	4	K		
23	:	N		

したがって、「0123」は「4K:N」とマッピングされ以下のように印字されます。



■ 出力例

メーカーコード (MMMMMM)	4912345 (先頭の4は入力しません)
自由使用欄 (EEEE・・・EEE)	12233445 (21桁にするため、0を13個付けて入力します。)
再発行区分 (S)	0 (初回発行)
支払期限日 (YYMMDD)	2004年10月25日 (041025と入力します)
印紙フラグ (F)	0 (貼らない)
請求金額 (PPPPPP)	12,345円 (6桁にするため、0を1個付けて入力します。)
計算後のチェックデジット	5

↓アプリケーション識別子(91) を先頭に付け、数字を順に並べる

91 912345 000000000000012233445 0 041025 0 012345 5
 (実際は文字間のスペースは必要ありません)

↓キャラクタマップ変換 (コードセット"C")

BJBJ:N#LOYOYOYOYOYOY4K5G!f*N6N%e3b=KOY"^L9d

↓以下のコードを先頭に追加

- ・スタートキャラクタコード (スタートC) : /^
- ・FNC1 : AN

/^ANBJBJ:N#LOYOYOYOYOYOY4K5G!f*N6N%e3b=KOY"^L9d

↓以下のコードを末尾に追加

- ・チェックキャラクタコード* : &B
- ・ストップキャラクタコード : {

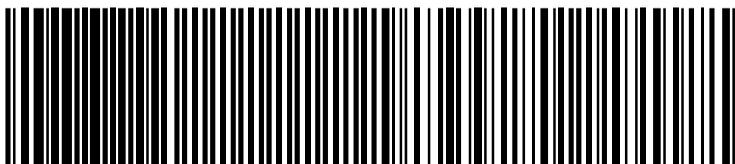
/^BJBJ:N#LOYOYOYOYOYOY4K5G!f*N6N%e3b=KOY"^L9d&B{

↓目視文字用の文字列を追加

/^BJBJ:N#LOYOYOYOYOYOY4K5G!f*N6N%e3b=KOY"^L9d&B{
 (91)912345-0000000000000122334450
 041025-0-012345-5

↓それぞれ文字セットを指定してバーコードを印刷

- ・バーコード : CI_CODE128S.BAR
- ・目視文字 : OCR-B.OCR_B



(91)912345-0000000000000122334450
 041025-0-012345-5

-
- * チェックキャラクタコードは、「モジユラス103」での計算結果より求めます。計算方法については、付録「6.5 EAN-128チェックデジット計算方法」を参照してください。

1 LIPSモード（スケーラブルフォント）での印刷方法

- ① 文字セット名称選択命令にて、印字するバーコードの文字セットを選択してプリンタに送って下さい。
EAN-128 : CI_CODE128.BAR
標準料金代理収納用EAN-128 : CI_EAN128S.BAR
- ② バーコードデータをプリンタに送る事によってバーコードの印刷ができます。

2 エミュレーションモード（ドットフォント）での印刷方法

- ① 文字セット名称選択命令Iのコマンドをプリンタに送る事によって印刷します。
EAN-128 : CI_CODE12848.BAR
標準料金代理収納用EAN-128 : CI_EAN128S48.BAR
- 例) ESC/Pエミュレーションモード時、文字セット名称“CI_CODE12848.BAR”でバーコードデータ /[^]ANOY&]BJ:N#LOYOY[^])L?J4X{ を印刷する場合

```
ESC I [SP] _f CI_CODE12848.BAR, 100, 000, 000,  
/^ANOY&]BJ:N#LOYOY^)L?J4X{
```

注意：エミュレーションモードとは、各オプションコントロールROMご使用時のモードです。

LIPSモードは、LIPSⅢ又はLIPSⅣでご使用になれます。

LIPSの詳細は、別売りのソフトウェア概要書、コマンドリファレンスマニュアルをご参照下さい。

3.7 EAN-128の印刷サンプル例 (CR-MENのみ)

「EAN-128」の印刷サンプル



LIPSモードでスケーラブルフォントを使用し、EAN-128のバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****
//
//      OutPutStuff_Lips( HDC )
//      PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff_Lips04( HDC hDC )
{
    COM Com; // Esc出力構造体
    short Count; // ループカウンタ
    char *TextMode = "%\x1e)p\x1e"; // ピクチャ終了&テキストモード移行命令
    char *CrLf1 = "\r\n"; // 改行命令
    char *Mojiset1 = "\x1bPzG1_CODE128.BAR\x1b%"; // 文字セット名称選択命令
    char *BarCode = "/^AN0Y6U5G!f*N9d%kCFDT#Q&V{"; // バーコード・データ
    char *KaiPage = "%c"; // 改ページ
    char *VectorMode = "\x1b[0&]*"; // ベクタモード移行命令
    char *TextStart = "\x1b@"; // テキストモード開始
    char *ShiftIn = "\x0f"; // シフトイン
    char *Mojisize1 = "\x1b[240 G"; // 文字サイズ指定
    char *Spacing = "\x1b[600 G"; // スペーシング量指定
    char *Mojipitch1 = "\x1b[?600 K"; // 文字ピッチ指定

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)TextMode );
    Com.len = strlen( TextMode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // ピクチャ終了&テキストモード移行

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)TextStart );
    Com.len = strlen( TextStart );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // テキストモード開始

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)ShiftIn );
    Com.len = strlen( ShiftIn );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // シフトイン

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)Mojiset1 );
    Com.len = strlen( Mojiset1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字セット登録

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)Mojipitch1 );
    Com.len = strlen( Mojipitch1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字ピッチ指定

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)Mojisize1 );
    Com.len = strlen( Mojisize1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字サイズ指定
}
```

```
    strcpy( (LPSTR)&Com.com[0], (LPSTR)Spacing );
    Com.len = strlen( Spacing );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // スペーシング量指定

    strcpy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );
    Com.len = strlen( CrLf1 );
    for (Count = 0; Count < 4; Count++) {                          // 4行改行
        Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );
    }

    strcpy( (LPSTR)&Com.com[0], (LPSTR)BarCode );
    Com.len = strlen( BarCode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // バーコード・データ

    strcpy( (LPSTR)&Com.com[0], (LPSTR)KaiPage );
    Com.len = strlen( KaiPage );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // 改ページ

    strcpy( (LPSTR)&Com.com[0], (LPSTR)VectorMode );
    Com.len = strlen( VectorMode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // ベクタモード移行命令
}
```

印刷結果



例 2

LIPSモードでドットフォントを使用し、EAN-128のバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****  
//  
//      OutPutStuff_Lips( HDC )  
//      PASSTHROUGHでプリンタに直接コマンドを送信する。  
//  
//*****  
void OutPutStuff_Lips03( HDC hDC )  
{  
    COM Com; // Esc出力構造体  
    short Count; // ループカウンタ  
    char *TextMode = "%x1e)p%x1e"; // ピクチャ終了&テキストモード移行命令  
    char *CrLf1 = "%xd%xa"; // 改行命令  
    char *MojiSet1 = "%x1bPzC1_CODE12848.BAR%x1b%Y"; // 文字セット名称選択命令  
    char *BarCode = "/^ANOY6U5G!f*N9d%kCFDT#Q&V{"; // バーコード・データ  
    char *KaiPage = "%xc"; // 改ページ  
    char *VectorMode = "%x1b[O&]¥""; // ベクタモード移行命令  
    char *TextStart = "%x1b%@"; // テキストモード開始  
    char *ShiftIn = "%x0f"; // シフトイン  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)TextMode );  
    Com.len = strlen( TextMode );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // ピクチャ終了&テキストモード移行  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)TextStart );  
    Com.len = strlen( TextStart );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // テキストモード開始  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)ShiftIn );  
    Com.len = strlen( ShiftIn );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // シフトイン  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)MojiSet1 );  
    Com.len = strlen( MojiSet1 );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字セット登録  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );  
    Com.len = strlen( CrLf1 );  
    for (Count = 0; Count < 4; Count++) { // 4行改行  
        Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );  
    }  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)BarCode );  
    Com.len = strlen( BarCode );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // バーコード・データ  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)KaiPage );  
    Com.len = strlen( KaiPage );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 改ページ  
  
    Istrcpy( (LPSTR)&Com.com[0], (LPSTR)VectorMode );  
    Com.len = strlen( VectorMode );  
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // ベクタモード移行命令  
}
```

印刷結果



例 3

ESC/Pモードでドットフォントを使用し、EAN-128のバーコードを印刷する場合。
(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
/**
//
//      OutPutStuff_Escp( HDC )
//      PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff_Escp02( HDC hDC )
{
    COM Com; // Esc出力構造体
    short Count; // ループカウンタ
    char *PageStart = "\x1b\x7c\x27\x00\x01\x30"; // LIPSの制御命令によるページ登録開始
    char *Bit8Image = "\x1b\x4b\x08\x00"; // 8ビットイメージ命令
    char *CrLf1 = "\xd\xa"; // 改行命令
    char *Mojiset1 = "\x1bPzC1_CODE12848.BAR\x1b"; // 文字セット名称選択命令
    char *BarCode = "/^AN0Y6U5G!f*N9d%kCFDT#Q&V{"; // バーコード・データ
    char *PageEnd = "\x1b\x7c\x2e\x00\x00"; // LIPSの制御命令によるページ終了開始
    char *PageEntry = "\x1b\x7c\x68\x00\x01\x30"; // ユーザページ登録開始

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)PageStart );
    Com.len = strlen( PageStart );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // LIPSの制御命令によるページ登録開始

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)Bit8Image );
    Com.len = strlen( Bit8Image );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 8ビットイメージ命令

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );
    Com.len = strlen( CrLf1 );
    for (Count = 0; Count < 4; Count++) { // 4行改行
        Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );
    }

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)Bit8Image );
    Com.len = strlen( Bit8Image );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 8ビットイメージ命令

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)Mojiset1 );
    Com.len = strlen( Mojiset1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字セット名称選択

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)BarCode );
    Com.len = strlen( BarCode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // バーコード・データ

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );
    Com.len = strlen( CrLf1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 改行

    Istcopy( (LPSTR)&Com.com[0], (LPSTR)PageEnd );
    Com.len = strlen( PageEnd );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // LIPSの制御命令によるページ終了開始
}
```

```
lstrcpy( (LPSTR)&Com.com[0], (LPSTR)PageEntry );  
Com.len = strlen( PageEntry );  
Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // ユーザページ登録開始  
}
```

印刷結果



「標準料金代理収納用 E A N - 1 2 8」の印刷サンプル

例 1

L I P Sモードでスケーラブルフォントを使用し、標準料金代理収納用 E A N - 1 2 8 のバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
/**
//
//      OutPutStuff_Lips( HDC )
//      PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff_Lips02( HDC hDC )
{
    COM    Com;                // Esc出力構造体
    short  Count;             // ループカウンタ
    char   *TextMode          = "%x1e p%x1e"; // ピクチャ終了&テキストモード移行命令
    char   *CrLf1             = "%d%a";      // 改行命令
    char   *MojiSet1          = "%x1bPzC1_EAN128S.BAR%x1b%"; // 文字セット名称選択命令
    char   *BarCode           = "~/^ANOY6U5G!f*N9d%kCFDT#Q&V{"; // バーコード・データ
    char   *KaiPage           = "%xc";       // 改ページ
    char   *VectorMode        = "%x1b[0%";   // ベクタモード移行命令
    char   *TextStart         = "%x1b%@";    // テキストモード開始
    char   *ShiftIn           = "%x0f";     // シフトイン
    char   *MojiSize1         = "%x1b[240 C"; // 文字サイズ指定
    char   *Spacing           = "%x1b[600 G"; // スペーシング量指定
    char   *MojiPitch1        = "%x1b[?600 K"; // 文字ピッチ指定

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)TextMode );
    Com.len = strlen( TextMode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // ピクチャ終了&テキストモード移行

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)TextStart );
    Com.len = strlen( TextStart );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // テキストモード開始

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)ShiftIn );
    Com.len = strlen( ShiftIn );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // シフトイン

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)MojiSet1 );
    Com.len = strlen( MojiSet1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字セット登録

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)MojiPitch1 );
    Com.len = strlen( MojiPitch1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字ピッチ指定

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)MojiSize1 );
    Com.len = strlen( MojiSize1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字サイズ指定

    Istrcpy( LPSTR)&Com.com[0], (LPSTR)Spacing );
    Com.len = strlen( Spacing );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // スペーシング量指定
}
```

```
lstrcpy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );
Com.len = strlen(CrLf1);
for (Count = 0; Count < 4; Count++) { // 4行改行
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );
}

lstrcpy( (LPSTR)&Com.com[0], (LPSTR)BarCode );
Com.len = strlen(BarCode);
Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // バーコード・データ

lstrcpy( (LPSTR)&Com.com[0], (LPSTR)KaiPage );
Com.len = strlen(KaiPage);
Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 改ページ

lstrcpy( (LPSTR)&Com.com[0], (LPSTR)VectorMode );
Com.len = strlen(VectorMode);
Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // ベクタモード移行命令
}
```

印刷結果



例 2

LIPSモードでドットフォントを使用し、標準料金代理収納用EAN-128のバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```

//*****
//
//      OutPutStuff_Lips( HDC )
//      PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff_Lips01( HDC hDC )
{
    COM Com;                                // Esc出力構造体
    short Count;                             // ループカウンタ
    char *TextMode = "%\%x1e]p%\%x1e";      // ピクチャ終了&テキストモード移行命令
    char *CrLf1 = "\%xd%\%xa";             // 改行命令
    char *Mojiset1 = "\%x1bPzCI_EAN128S48.BAR%\%x1b%\%"; // 文字セット名称選択命令
    char *BarCode = "/^AN0Y6U5G!f*N9d%kCFDT#Q&V!"; // バーコード・データ
    char *KaiPage = "\%xc";                // 改ページ
    char *VectorMode = "\%x1b[0&]\%";      // ベクタモード移行命令
    char *TextStart = "\%x1b%@";          // テキストモード開始
    char *ShiftIn = "\%x0f";              // シフトイン

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) TextMode );
    Com.len = strlen( TextMode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // ピクチャ終了&テキストモード移行

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) TextStart );
    Com.len = strlen( TextStart );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // テキストモード開始

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) ShiftIn );
    Com.len = strlen( ShiftIn );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // シフトイン

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) Mojiset1 );
    Com.len = strlen( Mojiset1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // 文字セット登録

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) CrLf1 );
    Com.len = strlen( CrLf1 );
    for (Count = 0; Count < 4; Count++) { // 4行改行
        Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL );
    }

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) BarCode );
    Com.len = strlen( BarCode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // バーコード・データ

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) KaiPage );
    Com.len = strlen( KaiPage );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // 改ページ

    Istrcpy( (LPSTR) &Com.com[0], (LPSTR) VectorMode );
    Com.len = strlen( VectorMode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR) &Com, NULL ); // ベクタモード移行命令
}

```

印刷結果



例 3

ESC/Pモードでドットフォントを使用し、標準料金代理収納用EAN-128のバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****
//
//      OutPutStuff_Escp( HDC )
//      PASSTHROUGHでプリンタに直接コマンドを送信する。
//
//*****
void OutPutStuff_Escp01( HDC hDC )
{
    COM Com;                                // Esc出力構造体
    short Count;                             // ループカウンタ
    char *PageStart = "\x1b\x7c\x27\x00\x01\x30"; // LIPSの制御命令によるページ登録開始
    char *Bit8Image = "\x1b\x4b\x08\x00";        // 8ビットイメージ命令
    char *CrLf1 = "\xd\xa";                    // 改行命令
    char *MojiSet1 = "\x1bPzC1_EAN128S48.BAR\x1bYY"; // 文字セット名称選択命令
    char *BarCode = "/^AN0Y6U5G!f*N9d%kCFDT#Q&V{"; // バーコード・データ
    char *PageEnd = "\x1b\x7c\x2e\x00\x00";     // LIPSの制御命令によるページ終了開始
    char *PageEntry = "\x1b\x7c\x68\x00\x01\x30"; // ユーザーページ登録開始

    Istropy( (LPSTR)&Com.com[0], (LPSTR)PageStart );
    Com.len = strlen( PageStart );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // LIPSの制御命令によるページ登録開始

    Istropy( (LPSTR)&Com.com[0], (LPSTR)Bit8Image );
    Com.len = strlen( Bit8Image );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 8ビットイメージ命令

    Istropy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );
    Com.len = strlen( CrLf1 );
    for (Count = 0; Count < 4; Count++) { // 4行改行
        Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );
    }

    Istropy( (LPSTR)&Com.com[0], (LPSTR)Bit8Image );
    Com.len = strlen( Bit8Image );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 8ビットイメージ命令

    Istropy( (LPSTR)&Com.com[0], (LPSTR)MojiSet1 );
    Com.len = strlen( MojiSet1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 文字セット名称選択

    Istropy( (LPSTR)&Com.com[0], (LPSTR)BarCode );
    Com.len = strlen( BarCode );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // バーコード・データ

    Istropy( (LPSTR)&Com.com[0], (LPSTR)CrLf1 );
    Com.len = strlen( CrLf1 );
    Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL ); // 改行
}
```

```
lstrcpy( (LPSTR)&Com.com[0], (LPSTR)PageEnd );  
Com.len = strlen( PageEnd );  
Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // LIPSの制御命令によるページ終了開始  
  
lstrcpy( (LPSTR)&Com.com[0], (LPSTR)PageEntry );  
Com.len = strlen( PageEntry );  
Escape( hDC, PASSTHROUGH, NULL, (LPCSTR)&Com, NULL );           // ユーザページ登録開始  
}
```

印刷結果



3.8 カスタマバーコードの印刷方法

- ① 文字セット名称選択命令にて、印字するバーコードの文字セットを選択してプリンタに送って下さい。
- ② バーコードデータをプリンタに送る事によってバーコードの印刷ができます。

注意：LIPSモードは、LIPSⅢ又はLIPSⅣでご使用になれます。

LIPSの詳細は、別売りのソフトウェア概要書、コマンドリファレンスマニュアルをご参照下さい。

3.9 カスタマバーコードの印刷サンプル例

例 1

「26300233-30-8-403」のカスタマバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (VC++)

```
//*****  
//  
//          関数 : OutPutStuff( HDC )  
//  
//          目的 : PASSTHROUGH でプリンタに直接コマンドを送信する。  
//  
//*****  
void OutPutStuff( HDC hDC )  
{  
    COM Com;                                //Escape 出力構造体  
    short Count;                             //ループカウンタ  
    char *TextMode                           = "%x1E)p%x1E"; //ピクチャ終了 &テキストモード移行命令  
    char *CrLf1                              = "%xD%A";      //改行命令  
    char *MojiSet1                            = "%x1BPzBARpostal.BAR%x1B¥¥"; //文字セット登録命令  
    char *PostalCode                          = "(26300233-30-8-403DDD5)"; //バーコードデータ  
    char *KaiPage                             = "%xC";        //改ページ  
    char *VectorMode                          = "%x1B[0&¥"";   //ベクタモード移行命令  
    char *TextStart                           = "%x1B%@";     //テキストモード開始  
    char *ShiftIn                             = "%x0F";       //シフトイン  
    char *MojiSize1                            = "%x1B[100 C";  //文字サイズ指定  
    char *SizeTani                            = "%x1B[2,300 I"; //文字サイズ単位指定  
    char *MojiPicchi1                         = "%x1B[?720 K"; //文字ピッチ指定  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)TextMode );  
    Com.l = strlen( TextMode );  
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL ); //ピクチャ終了 &テキストモード移行  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)TextStart );  
    Com.l = strlen( TextStart );  
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL ); //テキストモード開始  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)ShiftIn );  
    Com.l = strlen( ShiftIn );  
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL ); //シフトイン  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)MojiSet1 );  
    Com.l = strlen( MojiSet1 );  
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL ); //文字セット登録  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)SizeTani );  
    Com.l = strlen( SizeTani );  
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL ); //文字サイズ単位指定  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)MojiPicchi1 );  
    Com.l = strlen( MojiPicchi1 );  
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL ); //文字ピッチ指定  
  
    Istrcpy((LPSTR) &Com.com[ 0 ], (LPCSTR)MojiSize1 );  
    Com.l = strlen( MojiSize1 );
```

```

Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL );           // 文字サイズ指定

lstrncpy((LPSTR) &Com.com[ 0 ], (LPCSTR)CrLf1 );
Com.l = strlen( CrLf1 );
for ( Count = 0; Count < 4; Count ++){                          //4行改行
    Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL );
}

lstrncpy((LPSTR) &Com.com[ 0 ], (LPCSTR)PostalCode );
Com.l = strlen( PostalCode );
Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL );           //ハーフコードデータ

lstrncpy((LPSTR) &Com.com[ 0 ], (LPCSTR)KaiPage );
Com.l = strlen( KaiPage );
Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL );           // 改ページ

lstrncpy((LPSTR) &Com.com[ 0 ], (LPCSTR)VectorMode );
Com.l = strlen( VectorMode );
Escape( hDC, PASSTHROUGH, NULL, (LPSTR)&Com, NULL );           //ハーフタモト移行命令
}

```

印刷結果



例 2

「26300233-30-8-403」のカスタマバーコードを印刷する場合。(メインプログラムは章末を参照)

サンプルプログラム (BASIC)

```
10 rem ***** カスタマバーコード LIPSモード 印刷 *****
20 LPRINT CHR$(&H1B); CHR$(&H25); CHR$(&H40);          'テキストモード開始
30                                                    'ジョブ開始命令

40 LPRINT CHR$(&H1B); CHR$(&H50); "31"; CHR$(&H3B); "300"; CHR$(&H3B); "1"; CHR$(&H4A); CHR$
(&H1B); CHR$(&H5C);
50 LPRINT CHR$(&H1B); CHR$(&H5B); "0"; CHR$(&H22); "p";
60 LPRINT CHR$(&H1B); CHR$(&H3C);                          'ソフトリセット

70 LPRINT CHR$(&HF);                                       'シフトイン

80 LPRINT CHR$(&H1B); "PzBARpostal.BAR"; CHR$(&H1B); "¥"; '文字セット名称選択
90

100 LPRINT CHR$(&H1B); CHR$(&H5B); "2"; CHR$(&H3B); "300"; "1"; 'サイズ単位選択
110 LPRINT CHR$(&H1B); CHR$(&H5B); "?720"; "K";           '文字ピッチ指定
120 LPRINT CHR$(&H1B); CHR$(&H5B); "100"; "C";           '文字サイズ指定
130 LPRINT CHR$(&HD); CHR$(&HA);                          '改行
140 LPRINT CHR$(&HD); CHR$(&HA);                          '改行
150 LPRINT CHR$(&HD); CHR$(&HA);                          '改行
160 LPRINT CHR$(&HD); CHR$(&HA);                          '改行
170 LPRINT "(26300233-30-8-403DDD5)";                    'バーコード印刷
180 LPRINT CHR$(&HC);                                     '改ページ

190                                                    'ジョブ終了命令

200 LPRINT CHR$(&H1B); CHR$(&H50); CHR$(&H30); CHR$(&H4A); CHR$(&H1B); CHR$(&H5C);
210 END
```

印刷結果



3.10 VC++用共通サンプルメインプログラム

```
//*****  
// Windowsハ`コード`印刷 ソースファイル  
//-----  
// プログラム:MAIN.CPP  
//  
// 目的:Windowsハ`コード`印刷  
//*****  
//*****  
// インクルードファイル定義  
//*****  
#include "windows.h" //すべてのWindowsアプリケーションに必要  
#include <string.h>  
  
//*****  
// グローバル変数定義  
//*****  
typedef struct { //Escape 出力構造体  
    WORD l; //出力データサイズ  
    char com[128]; //出力データ  
} COM ;  
  
//*****  
//  
// 関数:WinMain(HANDLE, HANDLE, LPSTR, int)  
//  
// 目的:Windowsプログラムメイン  
//  
//*****  
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)  
{  
    char szPrinter[64]; //プリンタ情報取得  
    char *szDriver; //ドライバ名  
    char *szDevice; //デバイス名  
    char *szOutput; //出力先名  
    HDC hDC; //デバイスコンテキストのハンドル  
    DOCINFO DocInfo; //StartDoc 関数を使用する入力ファイル名と出力ファイル名を格納する構造体  
  
    int Ret; //MessageBox 関数の戻り値  
  
    if (hPrevInstance != NULL) { //ほかのインスタンスが実行中か?  
        return ( FALSE); //初期化に失敗した場合は終了  
    }  
  
    Ret = MessageBox ( NULL, //印刷開始メッセージ表示  
        "印刷を開始してよろしいですか？",  
        "Windows Print", MB_OKCANCEL);  
    if ( Ret == IDCANCEL) { //キャンセルボタン押下時  
        return( FALSE); //プログラム終了  
    }  
  
    GetProfileString( "windows", "device", "", szPrinter, 64); //WIN.INIより使用プリンタ情報取得  
    szDevice = strtok ( szPrinter, ",");  
    szDriver = strtok ( NULL, ",");  
    szOutput = strtok ( NULL, ",");  
    hDC = CreateDC (szDriver, szDevice, szOutput, NULL); //プリンタのデバイスコンテキスト作成
```

```
//DOCINFO 構造体 初期化
DocInfo.cbSize      = sizeof( DOCINFO);           // 構造体のサイズ指定
DocInfo.lpszDocName = "TEST";                   // 文書の名前指定
DocInfo.lpszOutput  = NULL;                      //ファイル名も指定可能

// ここで、印刷を行う
if ( StartDoc( hDC, &DocInfo) > 0) {           // 印刷ジョブの開始
    // プリントに出力する
    OutPutStuff( hDC);
    EndDoc( hDC);                               // 印刷ジョブを終了
}
else {                                           // StartDocエラー
    AbortDoc( hDC);                             // 現在の印刷ジョブの強制終了
    MessageBox(NULL,                            // エラーメッセージ表示
        "プリンタを起動できませんでした。",
        "Warning", MB_OK);
}
DeleteDC( hDC);                                 // プリンタのデバイスコンテキスト 削除

return( TRUE);                                  // 初期化に失敗した場合は終了
}
```

4. バーコードフォント仕様

4.1 共通仕様

- 各文字セットは、JAN、EAN、UPC、CODE39、NW-7、EAN-128 (CR-MENのみ)、標準料金代理収納用EAN-128 (CR-MENのみ)、OCR-B、カスタマバーコードフォントとも1バイトフォントです。
- 印字するバーコードの大きさによって文字セット名称を選択します。
- バーコードの印刷は、それぞれの文字セットの持つ属性により印刷します。
- 各バーコードの割当てられていない文字コードの印字は、スペースに置き換わります。
- それぞれのバーコードには、各種ドットフォントによるモジュールとスケラブルフォントによるモジュールのバーコードフォントがあります。
詳細は、付録 文字セット一覧をご参照下さい。

4.2 JAN、EAN、UPC用フォント仕様

1 文字セット仕様（ドットフォント）
JAN、EAN、UPC用の文字セットは以下の構成になっています。

- (1) 左側のデータキャラクタの奇数パリティ用のフォント、及び文字コード置き換えで使用する全フォントが割り当てられている文字セット
(= J X₁ X₂ X₃ X₄ A. BAR)
- (2) 左側のデータキャラクタの偶数パリティ用のフォントが割り当てられている文字セット
(= J X₁ X₂ X₃ X₄ B. BAR)
- (3) 右側のデータキャラクタ用のフォントが割り当てられている文字セット
(= J X₁ X₂ X₃ X₄ C. BAR)
- (4) レフトガードバー、センターガードバー、ライトガードバー用のフォントが割り当てられている文字セット
(= J X₁ X₂ X₃ X₄ D. BAR)

JAN、EAN、UPCコードを印刷する際には、上記4つの文字セットまたは、(1)の文字セットが必要になります。

各文字セットに持つフォントは以下のとおりです。

- J X₁ X₂ X₃ X₄ A. BARの文字セット—— 数字の0～9
英数の@～I, P～Y, L～O
- J X₁ X₂ X₃ X₄ B. BARの文字セット—— 数字の0～9
- J X₁ X₂ X₃ X₄ C. BARの文字セット——
- J X₁ X₂ X₃ X₄ D. BARの文字セット—— 英大のL、M、R、E

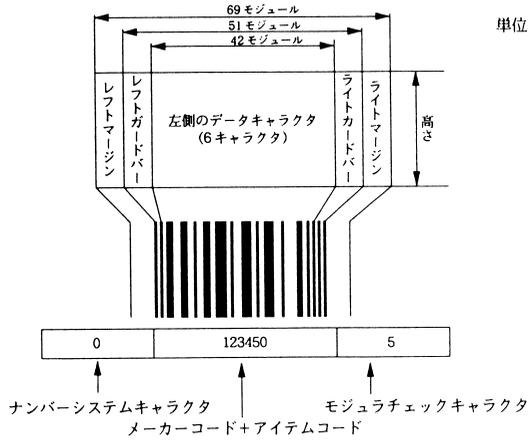
- X₁ ドットモジュール名
 - 2 : 3 ドットモジュール
 - 3 : 4 ドットモジュール
 - 5 : 6 ドットモジュール
- X₂ X₃ X₄ バーコードの高さ
 - 0 6 0 : 高さ 7 5 ドット (300dpi)
 - 0 8 0 : 高さ 1 0 0 ドット (300dpi)
 - 1 0 0 : 高さ 1 2 5 ドット (300dpi)

(例) 文字セット名称 J 2 0 6 0 C. BARの場合
モジュールの幅 = 3 ドット、バーコードの高さ = 7 5 ドット
右側にデータキャラクタの文字セット

バーコードシンボル

データキャラクタ … 数字(0~9)
 レフトガードバー … 英字(L)
 ライトガードバー … 英字(E)

UPC短縮バージョン (6桁)



4 バーコードの寸法

バーコードの寸法の算出方法は以下の通りです。

マージンを含まないシンボル長 (L) は次の計算式で求められます。

JAN、EAN、UPC標準バージョンの時・・・ L = 95 X

JAN、EAN短縮バージョンの時・・・ L = 67 X

UPC短縮バージョンの時・・・ L = 51 X

(X = ドットモジュール幅)

(例) 3ドットモジュール 標準バージョンの時

$$L = 95 \times 3 = 285 \text{ dot} (= 285 \times 0.0846 = 24.11 \text{ mm})$$

4.3 CODE 39用フォント仕様

1 文字セット仕様（ドットフォント）
CODE 39用の文字セットは、以下の構成になっています。

N₁ N₂ N₃ N₄. BARの文字セット

N₁ ドットモジュール名
2 : 3 ドットモジュール
3 : 4 ドットモジュール
5 : 6 ドットモジュール

N₂ N₃ N₄ バーコードの高さ
0 6 0 : 高さ 7 5 ドット (300dpi)
0 8 0 : 高さ 1 0 0 ドット (300dpi)
1 0 0 : 高さ 1 2 5 ドット (300dpi)

(例) C 2 0 6 0 . BARの場合
CODE 39細バーが3ドット、バーコードの高さが75ドットのバーコード用の文字セット

バーコードのシンボル（取り扱う文字）

スタート、ストップコード . . . *
データキャラクタ 数字（0～9）、英大（A～Z）
記号（-、.、スペース、\$、/、+、%）

構成

クワイエットゾーン（マージン）、スタートコード（*）、データキャラクタ文字、ストップコード（*）、クワイエットゾーンで構成されます。
クワイエットゾーンの最小値は（細バー）×10、または2.54mmのいずれか大きい方の値にします。

バーコードの寸法

- (1) キャラクタ間ギャップ
キャラクタ間ギャップは、細バーの太さと同様になっています。
各キャラクタにはギャップを含んだ文字ピッチが与えられます。
- (2) 細バーと太バーの比
細バーと太バーの構成比は以下の様になっています。
- (3) シンボル長（バーコードの長さ）

細バー	太バー	RATIO
3	7	2.3
4	10	2.5
6	15	2.5

RATIO=細バーと太バーの比

シンボル長（L）は次の計算式で求められます。

但し、クワイエットゾーンは含みません。

$$L = (C + 2)(6X + 3 + NX) + 1(C + 1)$$

X = 細バーの幅

C = スタート、ストップビットを含まない全キャラクタの総数

N = 細バーと太バーの比（例： 3 : 7 : 2 . 3）

1 = キャラクタ間ギャップ（例： 細バーが3ドットの時、3）

2 文字セット仕様（スケーラブルフォント）

CODE 39用の文字セットは以下の構成になっています。

文字の大きさは24ポイント指定時（300dpi）

42 × 100ドット

4.4 NW-7用フォント仕様

- 1** NW-7文字セット仕様（ドットフォント）
NW-7用の文字セット仕様は以下の構成になっています。

NC1 C2 C3 C4. B A Rの文字セット

C1 ドットモジュール名
2 : 2 ドットモジュール
3 : 4 ドットモジュール
5 : 6 ドットモジュール

C2 C3 C4 バーコードの高さ
0 6 0 : 高さ 7 5 ドット (300dpi)
0 8 0 : 高さ 1 0 0 ドット (300dpi)
1 0 0 : 高さ 1 2 5 ドット (300dpi)

(例) N 2 0 6 0 . B A Rの場合
NW-7細バーが2ドット、バーコードの高さが75ドットのバーコード用の文字セット

バーコードのシンボル（取り扱う文字）

スタート、ストップコード . . . A~D、a~e、n、t、*

データキャラクタ 数字（0~9）

記号（-、\$、:、/、.、+）

通常スタート、ストップコードはA~Dまたはa~dが使用され、同じキャラクタが使われ
ます。

バーコードの寸法

- (1) キャラクタ間ギャップ

キャラクタ間ギャップは、太バーの太さと同様になっています。

各キャラクタにはギャップを含んだ文字ピッチが与えられます。

- (2) 細バーと太バーの比

細バーと太バーの構成比は以下の様になっています。

- (3) シンボル長（バーコードの長さ）

細バー	太バー	R A C I O
2	6	3.0
4	10	2.5
6	17	2.8

R A C I O = 細バーと太バーの比

シンボル長 (L) は次の計算式で求められます。

$$L = \{(2N+5)(C+2) + (N-1)(W+2)\} X + I (C+1)$$

X = 細バーの幅

C = スタート、ストップビットを含まない全キャラクタの総数

W = 幅広キャラクタコードの総数 (:、 /、 .、 +)

N = 細バーと太バーの比 (例 : 2 : 6 = 3 . 0)

I = キャラクタ間ギャップ (例 : 細バーが 3 ドットの時、 3)

(例) 細バー 2 ドット、データキャラクタ 1 + 2 1 3 の時

$$\begin{aligned} L &= \{(2 \times 3.0 + 5)(5 + 2) + (3.0 - 1)(2 + 2)\} 2 + 2(5 + 1) \\ &= 182 \text{ dot} (= 15.4 \text{ mm}) \end{aligned}$$

2 文字セット仕様 (スケーラブルフォント)

NW-7 用の文字セットは以下の構成になっています。

文字の大きさは 24 ポイント指定時 (300 dpi)

WIDE : 32 × 100 ドット

NARROW : 28 × 100 ドット

4.5 EAN-128 フォント仕様 (CR-MENのみ)

1 文字セット仕様 (ドットフォント)

1-1 EAN-128

EAN-128 用の文字セットは、以下の構成になっています。

CI_CODE12848.BAR

└── 48 : ポイント数

バーコードのシンボル (取り扱う文字)

アスキーコード 128 文字 (数字、アルファベット大文字/小文字、記号、制御コード) すべて。

1-2 標準料金代理収納用E A N－1 2 8

標準料金代理収納用E A N－1 2 8用の文字セットは、以下の構成になっています。

CI_EAN128S48.BAR

└── 48：ポイント数

バーコードのシンボル（取り扱う文字）

数字（0～9）

2 文字セット仕様（スケーラブルフォント）

2-1 E A N－1 2 8

E A N－1 2 8用の文字セットは、以下の構成になっています。

文字の大きさ（高さ）は48ポイント指定時（600dpi）

240ドット

注意：本ROMに収録されているバーコードフォントE A N－1 2 8は、48ポイントのサイズでの出力に適しています。48ポイント以外のサイズで出力したバーコードの読み取りは保証しません。

2-2 標準料金代理収納用E A N－1 2 8

標準料金代理収納用E A N－1 2 8用の文字セットは、以下の構成になっています。

文字の大きさ（高さ）は48ポイント指定時（600dpi）

240ドット

注意：本ROMに収録されているバーコードフォントE A N－1 2 8は、48ポイントのサイズでの出力に適しています。48ポイント以外のサイズで出力したバーコードの読み取りは保証しません。

4.6 OCR_Bフォント仕様

- 1** 文字セット仕様（ドットフォント）
OCR-B用の文字セットは以下の構成になっています。

- A2540. OCR_Bの文字セット
300 dpi フォント : 25×40ドット
- A2032. OCR_Bの文字セット
600 dpi フォント : 50×80ドット

- 2** 文字セット仕様（スケーラブルフォント）
OCR-B用の文字セットは以下の構成になっています。

文字の大きさは10ポイントを指定した場合、ドットフォントと同じサイズになります。

本フォントROMのOCR_Bフォントが持つ文字コード、フォント仕様の詳細は、付録のフォント仕様一覧OCR_Bフォントのページをご参照下さい。

本フォントは、JIS規格に準拠したフォントになっておりますが、フォントデザインや文字ピッチ等の関係からOCRリーダー等で読み取りにくい場合もあります。予めご了承下さい。

4.7 カスタマバーコードフォント仕様

- 1** 文字セット仕様（スケーラブルフォント）

本フォントROMのカスタマバーコードフォントが持つ文字コード、フォント仕様の詳細は、付録のフォント仕様一覧カスタマバーコードフォントのページをご参照下さい。

なお、カスタマバーコード用のフォントには、ドットフォントはありません。

■ 印字ピッチ

印字ピッチは次の計算式で求められます。

- (1) 文字高さをaポイント、1ポイント=1/72インチとします。
- (2) 文字高さ(インチ) = $a * 1/72$
- (3) 文字高さ = 文字幅
- (4) 文字ピッチ(cpi) = $1 \text{インチ} / (a/72) = 72/a$ (cpi)

例)

- ・ 8ポイントの場合の文字ピッチ .. $72/8 = 9\text{cpi}$
- ・ 9ポイントの場合の文字ピッチ .. $72/9 = 8\text{cpi}$
- ・ 10ポイントの場合の文字ピッチ .. $72/10 = 7.2\text{cpi}$
- ・ 11ポイントの場合の文字ピッチ .. $72/11 = 6.55\text{cpi}$ (6.54545454....)

5. カスタマバーコードについて

本製品により、郵便物にカスタマバーコードを印字することができます。

ここでは、お客様が郵便物にカスタマバーコードを印字するために必要な情報を説明しておりますので、内容をご理解の上、郵便物へのカスタマバーコード印字を行ってください。

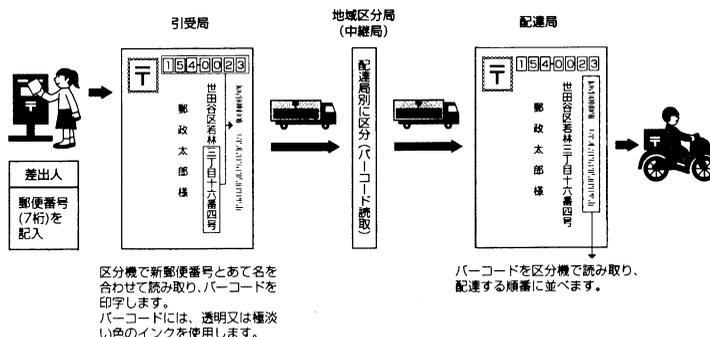
※「5 カスタマバーコードについて」は、郵便局発行の「新郵便番号制マニュアル」から抜粋しております。

5.1 新郵便番号による郵便処理

- (1) 引き受けた郵便局では、郵便局の区分機で新郵便番号とあて名を合わせて読み取り、住所全体を表すバーコードを印字し、以後の処理は、バーコードを読み取って行います。お客様があらかじめバーコードを印字して差し出した場合は、お客様の印字したバーコードを読み取って処理します。



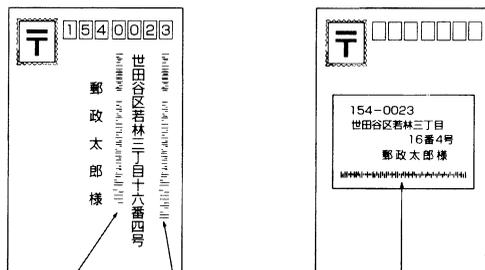
- (2) 配達局では、バーコードを読み取り、郵便物を配達順に並べるところまで機械化します。なお、バーコードがまだ印字されていない郵便物は、配達局でバーコードを印字します。



5.2 カスタマバーコードとは

郵便物に印字するバーコードには、局内バーコード、IDバーコード、カスタマバーコードの3種類があります。局内バーコードとIDバーコードは郵便局で印字するバーコードです。お客様があらかじめ郵便物に印字するバーコードがカスタマバーコードです。カスタマバーコードは、差出しの必要条件ではなく、料金割引（バーコード割引）を受けようとする場合に印字するもので、あて名印字と同時に印字できるような仕様としています。

〔葉書の場合〕



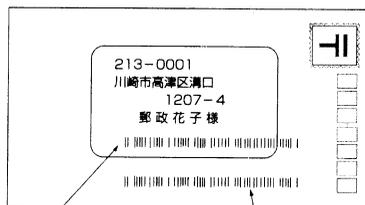
IDバーコード 局内バーコード

バーコードは透明又は極淡い色の特殊なインクで印字

カスタマバーコード

カスタマバーコードは、黒または濃い青のインクで印字（あて名と同時に印字可能）

〔封筒の場合〕



IDバーコード

局内バーコード



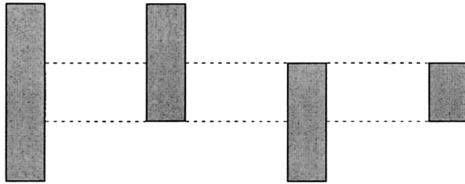
カスタマバーコード

5.3 カスタマバーコードの仕様

ア. カスタマバーコードの形

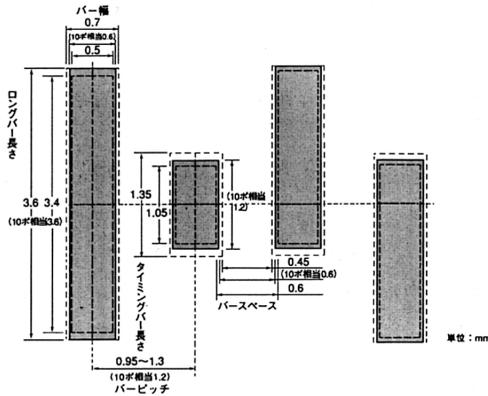
カスタマバーコードは、上下にバーを延ばしたロングバー、上方向のみにバーを延ばしたセミロングバー（上）、下方向のみにバーを延ばしたセミロングバー（下）及びタイミングバーの4つの形状のバーを3本組み合わせると1つのキャラクタを表す4ステイト3バーとします。

ロングバー:1 セミロングバー(上):2 セミロングバー(下):3 タイミングバー:4

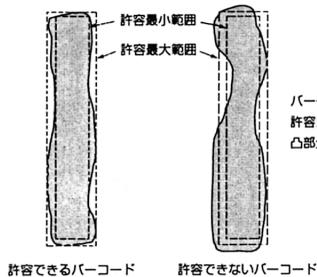


イ. カスタマバーコードの体系について

(カスタマバーコードの寸法) (10ボ相当)

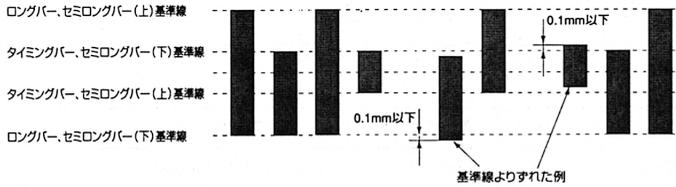


(バー印刷のみ出しに関する許容範囲)



バー一本一本に関しては、各ポイントにおける許容最小範囲内をバーが満たし、そこからの凸部分が許容最大範囲内に納めてください。

(バーコードの基準線からのばらつきに関する許容範囲)



バーコードの基準線からのばらつきは±0.1mmとします。

桁コード	かたまりバーコード	バー種類	本製品のコード番号
1	·	114	31
2		132	32
3		312	33
4		123	34
5	·	141	35
6		321	36
7		213	37
8		231	38
9	·	411	39
0	·	144	30
ハイフン	·	414	2D
CC1	·	324	41
CC2		342	42
CC3	·	234	43
CC4	·	432	44
CC5	·	243	45
CC6	·	423	46
CC7	·	441	47
CC8		111	48
スタート		13	28
ストップ		31	29

※本製品のコード番号は、Aケコードです。

桁コード	かたまりバーコード	バー種類	コード組合せ	本製品のコード番号
A	·	324144	CC1+0	4130
B	·	324114	CC1+1	4131
C	·	324132	CC1+2	4132
D	·	324312	CC1+3	4133
E	·	324123	CC1+4	4134
F	·	324141	CC1+5	4135
G	·	324321	CC1+6	4136
H	·	324213	CC1+7	4137
I	·	324231	CC1+8	4138
J	·	324411	CC1+9	4139
K	·	342144	CC2+0	4130
L	·	342114	CC2+1	4231
M	·	342132	CC2+2	4232
N	·	342312	CC2+3	4233
O	·	342123	CC2+4	4234
P	·	342141	CC2+5	4235
Q	·	342321	CC2+6	4236
R	·	342213	CC2+7	4237
S	·	342231	CC2+8	4238
T	·	342411	CC2+9	4239
U	·	234144	CC3+0	4330
V	·	234114	CC3+1	4331
W	·	234132	CC3+2	4332
X	·	234312	CC3+3	4333
Y	·	234123	CC3+4	4334
Z	·	234141	CC3+5	4335

ウ. カスタマバーコードの寸法

カスタマバーコードの寸法は、次表の通りとし（10ポイントの場合、 $a/10=1$ ）、 $8 \leq a \leq 11.5$ の大きさを許すものとします（8ポイントから11.5ポイント相当）。

aポイント	比率	基準寸法(mm)	許容範囲(mm)
ロングバー長さ	6	$3.6 \times a/10$	$3.40 \times a/10 \sim 3.60 \times a/10$
タイミングバー長さ	2	$1.2 \times a/10$	$1.05 \times a/10 \sim 1.35 \times a/10$
バーピッチ	2	$1.2 \times a/10$	$0.95 \times a/10 \sim 1.30 \times a/10$
バー幅	1	$0.6 \times a/10$	$0.50 \times a/10 \sim 0.70 \times a/10$
バースペース	1	$0.6 \times a/10$	$0.45 \times a/10 \sim 0.60 \times a/10$

$$8 \leq a \leq 11.5$$

エ. カスタマバーコードのフォーマット及びけた数

カスタマバーコードのフォーマットは次のとおりとします。ただし、新郵便番号の3けた目と4けた目の間のハイフン及び新郵便番号と住所表示番号を連結するハイフンは省きます。また、英字1文字は、制御コードと数字コードの組み合わせにより表現し、バーコード2けた分として扱います。

フォーマット： スタートコード + 新郵便番号 + 住所表示番号 + チェックデジット + ストップコード
バーコード桁数： (1) (7) (13) (1) (1)

住所表示番号が規定のけた数13けたに対して過不足のある場合には、次により調整します。

13けたを超える場合	13けたまでの住所表示番号をバーコードに変換し、それ以上の情報は含めません。ただし、制御コード+数字コードで表される英字の制御コードが13けた目に当たる場合は、この制御コードに該当するバーコードまでを含めるものとします。
13けたに満たない場合	13けたになるまで制御コードCC4に該当するバーコードで埋めるものとします。

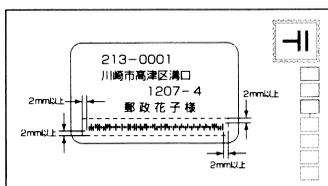
また、チェックデジットは、新郵便番号～住所表示番号に盛り込む情報の各キャラクタをチェック用数字に置き換え、その合計が19の倍数となるように生成します。各キャラクタのチェック用数字への置き換えは、次のとおりとします。

バーコード用キャラクタ	0	1	2	3	4	5	6	7	8	9
チェック用数字	0	1	2	3	4	5	6	7	8	9

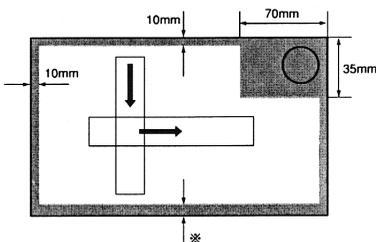
バーコード用キャラクタ	-	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8
チェック用数字	10	11	12	13	14	15	16	17	18

オ. 印字位置

カスタマバーコードの上下左右には、2mm以上の空白を設けるものとします。ただし、窓枠とカスタマバーコードの間の空白は、封筒と内容物のズレにかかわらず、常に上下左右とも2mm以上を確保するものとします。



あて名を横書きする場合は、最下行（あて先氏名の直下）にカスタマバーコードを単独で印字することとし、あて名を縦書きにする場合には、左右又は下部に単独で印字することとします。なお、カスタマバーコードは、次の図の様に、郵便物の表面の縁から10mm及び消印領域である70mm×35mmを除いた範囲内に印字することができます。また、次のような2つの方向で印字することができます。



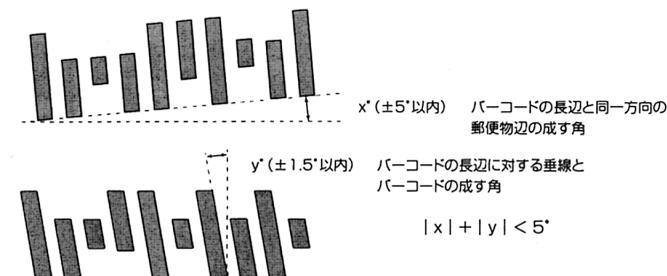
印字可能領域は、郵便物の縁から10mm（ただし、※部は、できる限り縁から15mm以上開けていただくようお願いします）及び消印領域である70mm×35mmを除いた範囲。印字方向は、矢印の方向。

カ. カスタマバーコードが印字される下地

カスタマバーコードが印字される下地は、白色又は地模様のない淡い色とします。

キ. カスタマバーコードの傾き

カスタマバーコードの傾きは、バーコードの長辺と同一方向の郵便物辺が成す角が5度以内になるようにします。また、バーコードの長辺に対する垂線とバーコードとの成す角は1.5度以内になるようにするものとし、上記2つが混在する場合には、2つの傾きの絶対値を加えたものが、5度以内になるようにします。



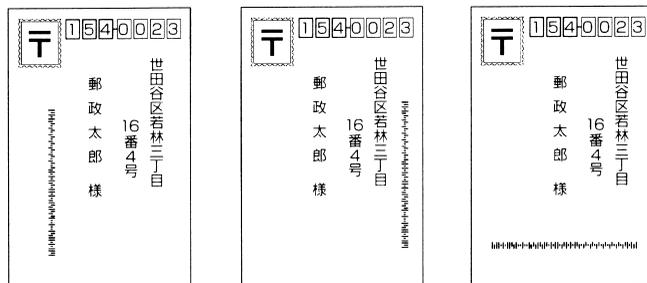
ク. 使用インク

カスタマバーコードの印字に使用されるインクは、プリンタによる通常のあて名打ち出しに使用されるインクと同等のものとし、黒又は濃い青色とします。

ケ. 印字品質

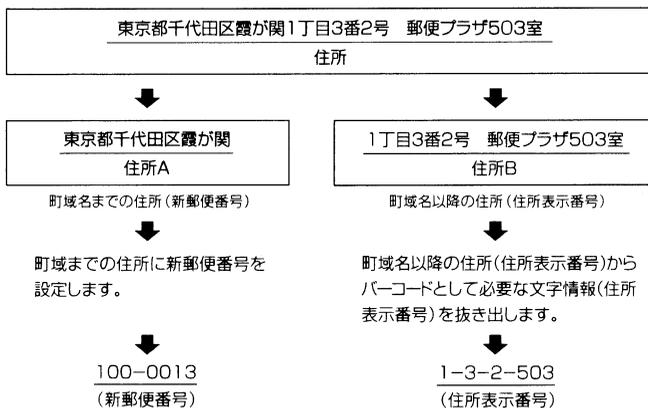
カスタマバーコード印字面には、反射率50%以上の紙を使用し、印字面とカスタマバーコードとの反射率PCS (Print Contrast Signal) は、0.6以上とします。また、カスタマバーコードにはインクのにじみやかすれなどが極力ないものとします。

(カスタマバーコード印刷例)

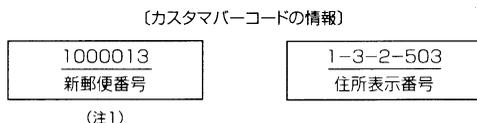


5.4 バーコードに必要な文字情報の抜き出し法

- (1) カスタマバーコードに必要な文字情報の抜き出しから生成までの処理概要
「住所」を町域名までの「住所A」と町域名以降の「住所B」に分割します。



前記の新郵便番号と住所表示番号を連結してカスタマバーコードの情報とします。



- (注1) バーコードの情報を生成するときは、新郵便番号の3～4けた目間のハイフンを省きます。

前記のカスタマバーコードの情報にチェックデジット、スタートコード、ストップコードを付加してカスタマバーコードを生成します。

スタートコード+新郵便番号+住所表示番号+チェックデジット+ストップコード

(STC)	1000013	1-3-2-503	CC4	CC4	CC4	CC4	9	SPC)
(1)	(7)	(13)	(注2)				(1)	(1) 桁数

- (注2) 「CC4」は、住所表示番号部分が13けたに満たない場合に充足する制御コードです。

- (注3) 「チェックデジット9」は、(7)の「チェックデジットの計算方法」を参照して下さい。

(2) カスタマバーコードを付番する住所データについて、次のAとBの部分に住所を2分割します。

住所A：町域名までの住所

住所B：町域名以降の住所

① 千葉市稲毛区緑町3丁目30-8 郵便ビル403号

┌──────────┴──────────┐
住所A 住所B

② 北海道札幌市東区北六条東4丁目 郵便センター6号館

┌──────────┴──────────┐
住所A 住所B

新郵便番号は、基本的に上記の住所A（町域名までの住所）毎に設定されますが、②のように、丁目により新郵便番号が異なる地域もあります。

北海道札幌市東区北六条東(1～7丁目) ⇒ 新郵便番号 060-0906

〃 (8～20丁目) ⇒ 〃 065-0006

┌──────────┴──────────┐
住所A

(3) 住所A(町域名までの住所)と住所B(町域名以降の住所)とに2分割した住所データを用いて、新郵便番号を設定します。

ア. 住所A(町域名までの住所)により新郵便番号が設定されている場合

① 千葉市稲毛区緑町3丁目30-8 郵便ビル403号

┌──────────┴──────────┐
住所A 住所B

↓
新郵便番号 263-0023

イ. 住所A(町域名までの住所)と住所B(町域名以降の住所)との組み合わせにより新郵便番号が設定されている場合

① 北海道札幌市東区北六条東4丁目 郵便センター6号館

┌──────────┴──────────┐
住所A 住所B

┌──────────┴──────────┐
住所A&住所B

↓
新郵便番号 060-0906

② 北海道札幌市東区北六条東8丁目 郵便センター10号館

┌──────────┴──────────┐
住所A 住所B

┌──────────┴──────────┐
住所A&住所B

↓
新郵便番号 065-0006

新郵便番号を設定する際に、代表の郵便番号（新郵便番号簿における‘以下に掲載がない場合’に該当する郵便番号）となった場合、カスタマバーコードは付番しません。代表の郵便番号は下記の例のように、新郵便番号の下2桁（6～7桁目）が原則として‘00’となります。

新郵便番号簿 草津市：以下に掲載がない場合 525-0000
和光市：以下に掲載がない場合 351-0100

- (4) 住所B (町域名以降の住所) のデータから、カスタマバーコードとして必要な文字情報を抜き出します。

ア. 抜き出しの基本ルール

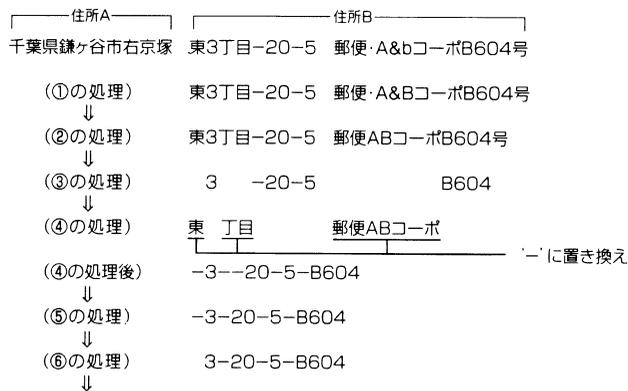
- ① まず、データ内にあるアルファベットの小文字は大文字に置き換えます。
- ② 同様に、データ内にある '&' 等の下記の文字は取り除き、後ろのデータを詰めます。

「&」(アンパサンド)、「/」(スラッシュ)、「・」(中グロ)、「。」(ピリオド)

- ③ ①および②で整理したデータから、算用数字、ハイフン及び連続していないアルファベット1文字を必要な文字情報として抜き出します。
- ④ 次に抜き出された文字の前にある下記の文字等は、ハイフン1文字に置き換えます。

「漢字」、「かな文字」、「カタカナ文字」、「漢数字」、「ブランク」、
「2文字以上連続したアルファベット文字」

- ⑤ ④の置き換えで、ハイフンが連続する場合は1つにまとめます。
- ⑥ 最後に、先頭がハイフンの場合は取り除きます。



バーコード・データ	3-20-5-B604
-----------	-------------

尚、後述の補足ルールにより最終のバーコード・データは下記となります。

バーコード・データ	3-20-5B604
-----------	------------

イ. 抜き出しの補足ルール

- ① 漢数字が下記の特定文字の前にある場合は抜き出し対象とし、算用数字に変換して抜き出します。

特定文字群(9種類)	‘丁目’ ‘丁’ ‘番地’ ‘番’ ‘号’ ‘地割’ ‘線’ ‘の’ ‘ノ’
------------	--

東京都青梅市河辺町十一丁目六番地一号 郵便タワー601
┌──住所A──┐┌──────────住所B──────────┐

バーコード・データ	11-6-1-601
-----------	------------

北海道帯広市稲田町南七線 西28
┌──住所A──┐┌──住所B──┐

バーコード・データ	7-28
-----------	------

- ② 連続していないアルファベット1文字は抜き出し対象となりますが、算用数字に続くアルファベット1文字‘F’に限っては抜き出し対象としません。

茨城県日立市宮田町6丁目7-14 ABCビル2F
┌──住所A──┐┌──住所B──┐

バーコード・データ	6-7-14-2
-----------	----------

- ③ ②に記述したように、算用数字に続くアルファベット1文字‘F’は抜き出し対象となりませんが、更に、‘F’以降のデータに抜き出し対象となる文字がある場合、‘F’はハイフン1文字に置き換えます。

茨城県日立市宮田町6丁目7-14 ABCビル2F201号室
┌──住所A──┐┌──住所B──┐

バーコード・データ	6-7-14-2-201
-----------	--------------

- ④ 抜き出し後のバーコードデータについて、アルファベット文字の前後にあるハイフンは取り除きます。

神戸市中央区港島中町9丁目7-6 郵便シティA棟1F1号
┌──住所A──┐┌──────────住所B──────────┐

9-7-6-A-1-1

バーコード・データ	9-7-6A1-1
-----------	-----------

- ⑤ ④の処理でアルファベット文字の前後にあるハイフンを取り除いた結果、2文字以上の連続したアルファベット文字が残った場合、取り除かないでそのままにします。

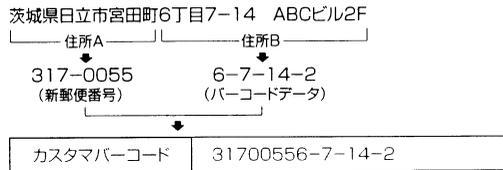
京都市綾部市青野町綾部6-7 LプラザB106
┌──住所A──┐┌──住所B──┐

6-7-L-B106

バーコード・データ	6-7LB106
-----------	----------

- (5) 新郵便番号とバーコードデータを連結し、チェックデジットを計算する前のカスタマバーコードを生成します。

連結の際、新郵便番号の3～4桁目の間のハイフンは取り除きます。

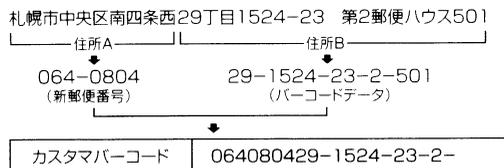


- (6) 生成したカスタマバーコード（チェックデジットは未計算）の合計桁数が20桁を越えた場合、以降の文字については切り捨てます。

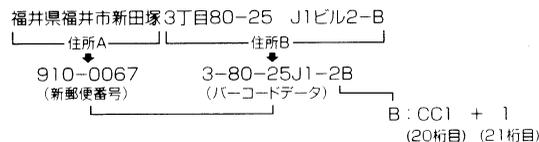
その際、20～21桁目がアルファベット文字となった場合、20桁目の英字用制御コード（CC1、CC2、CC3）は残して以降は切り捨てます。

注. アルファベット文字の場合、英字用制御コードと数字の2桁でアルファベット1文字を表します。

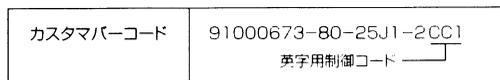
- ① 20桁目が数字あるいはハイフンとなる場合



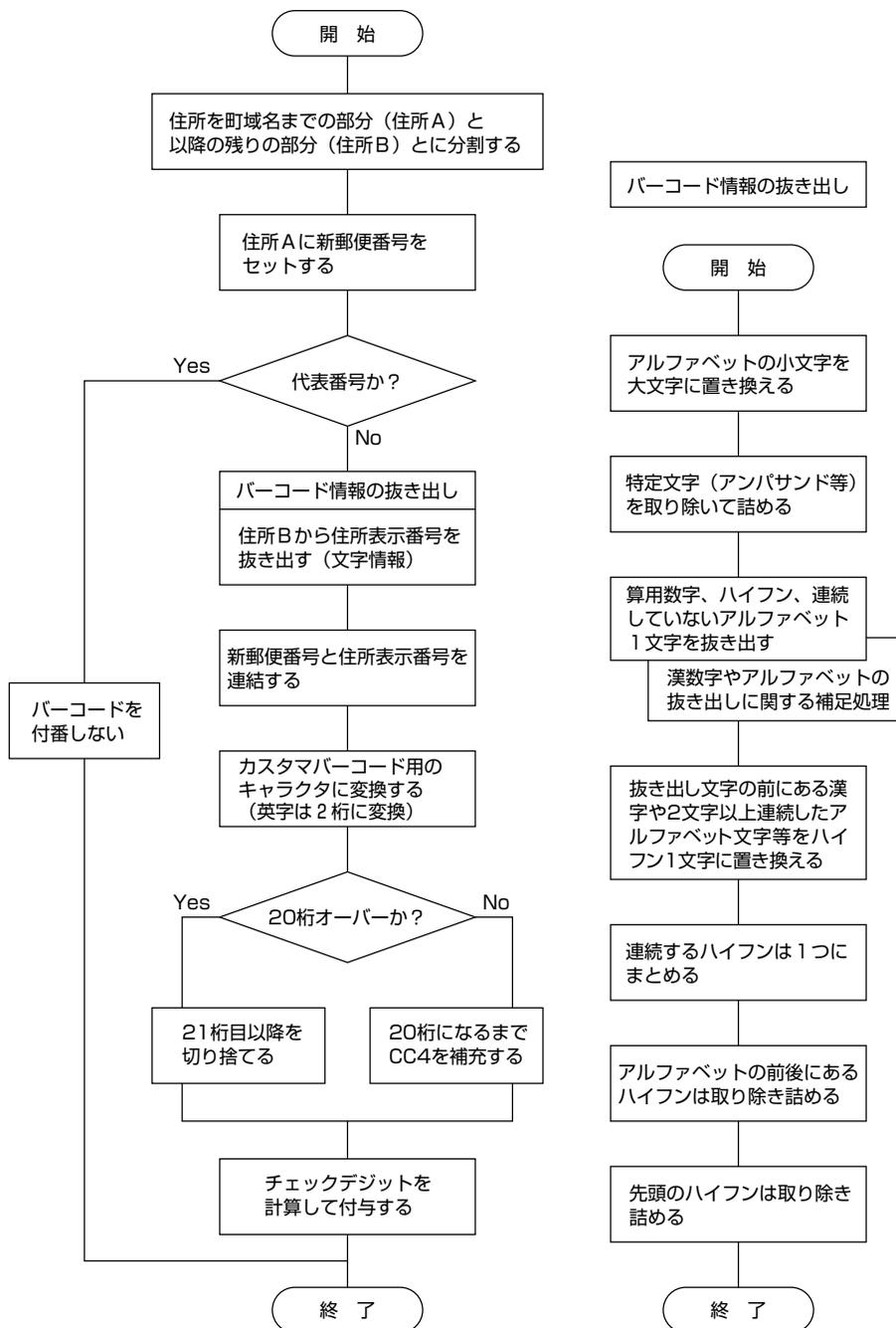
- ② 20～21桁目がアルファベット文字となる場合



前記のカスタマバーコードは下記となります。



カスタマバーコード生成フローチャート



バーコード情報の抜き出し

開始

アルファベットの小文字を
大文字に置き換える

特定文字（アンパサンド等）
を取り除いて詰める

算用数字、ハイフン、連続
していないアルファベット
1文字を抜き出す

漢数字やアルファベットの
抜き出しに関する補足処理

抜き出し文字の前にある漢
字や2文字以上連続したア
ルファベット文字等をハイ
フン1文字に置き換える

連続するハイフンは1つに
まとめる

アルファベットの前後にある
ハイフンは取り除き詰める

先頭のハイフンは取り除き
詰める

終了

カスタマバーコード付番事例（検証用）

以下に記載した住所及び文字列は、住所情報からカスタマバーコード作成に必要な文字情報の抜き出しが正しいかどうかを検証するためのチェックシートです。各項の住所情報がその下段の文字列に変換されていれば、その変換ソフトのロジックに問題はありません。

千葉市稲毛区緑町3丁目30-8 郵便ビル403号 STC 26300233-30-8-403 CC4 CC4 CC4 5 SPC
秋田県仙北郡仙北町堀見内 南田茂木 添60-1 STC 014011360-1 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC8 SPC
東京都台東区台東5-6-3 ABCビル10F STC 11000165-6-3-10 CC4 CC4 CC4 CC4 CC4 9 SPC
北海道札幌市東区北六条東4丁目 郵便センター6号館 STC 06009064-6 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 9 SPC
北海道札幌市東区北六条東8丁目 郵便センター10号館 STC 06500068-10 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 9 SPC
山梨県韮崎市龍岡町下條南割 韮崎400 STC 4070033400 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4-SPC
千葉県鎌ヶ谷市右京塚 東3丁目20-5 郵便・A&bコーポB604号 STC 27301023-20-5 CC1 1 604 CC4 CC4 0 SPC
東京都青梅市河辺町十一丁目六番地一号 郵便タワー601 STC 198003611-6-1-601 CC4 CC4 CC4 CC8 SPC
岩手県宮古市大字津軽石第二十一地割大淵川480 STC 027020321-480 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC5 SPC
大阪府堺市中田出井町四丁目六番十九号 STC 59000164-6-19 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC2 SPC
北海道帯広市稲田町南七線 西28 STC 08008317-28 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC4 CC7 SPC
茨城県日立市宮田町6丁目7-14 ABCビル2F STC 31700556-7-14-2 CC4 CC4 CC4 CC4 CC4 CC1 SPC
神戸市中央区港島中町9丁目7-6 郵便シティA棟1F1号 STC 65000469-7-6 CC1 01-1 CC4 CC4 CC4 5 SPC
京都府綾部市青野町綾部6-7 LプラザB106 STC 62300116-7 CC2 1 CC1 1 106 CC4 CC4 CC4 4 SPC
神奈川県座間市入谷6丁目3454-5 郵便ハイツ6-1108 STC 22800246-3454-5-6-112 SPC
札幌市中央区南四条西29丁目1524-23 第2郵便ハウス501 STC 064080429-1524-23-2-3 SPC
福井県福井市新田塚3丁目80-25 J1ビル2-B STC 91000673-80-25 CC1 9 1-2 CC1 9 SPC

STC（スタートコード） SPC（ストップコード） CC1～CC8（英字用制御コード）

6. 付録

6.1 バーコードフォントROM 文字セット名称一覧

バーコードの種類	ドット幅	高さ	文字セット名称
JAN系	3ドット	75ドット	J2060A.BAR J2060B.BAR J2060C.BAR J2060D.BAR
		100ドット	J2080A.BAR J2080B.BAR J2080C.BAR J2080D.BAR
		125ドット	J2100A.BAR J2100B.BAR J2100C.BAR J2100D.BAR
	4ドット	75ドット	J3060A.BAR J3060B.BAR J3060C.BAR J3060D.BAR
		100ドット	J3080A.BAR J3080B.BAR J3080C.BAR J3080D.BAR
		125ドット	J3100A.BAR J3100B.BAR J3100C.BAR J3100D.BAR
	6ドット	75ドット	J5060A.BAR J5060B.BAR J5060C.BAR J5060D.BAR
		100ドット	J5080A.BAR J5080B.BAR J5080C.BAR J5080D.BAR
		125ドット	J5100A.BAR J5100B.BAR J5100C.BAR J5100D.BAR
		スケーラブルフォント	BARjan-A.BAR BARjan-B.BAR BARjan-C.BAR BARjan-D.BAR
CODE39	3ドット	75ドット	C2060.BAR
		100ドット	C2080.BAR
		125ドット	C2100.BAR
	4ドット	75ドット	C3060.BAR
		100ドット	C3080.BAR
		125ドット	C3100.BAR
	6ドット	75ドット	C5060.BAR
		100ドット	C5080.BAR
		125ドット	C5100.BAR
		スケーラブルフォント	BARcode39.BAR
NW-7	2ドット	75ドット	C2060.BAR
		100ドット	N2080.BAR
		125ドット	N2100.BAR
	4ドット	75ドット	N3060.BAR
		100ドット	N3080.BAR
		125ドット	N3100.BAR
	6ドット	75ドット	N5060.BAR
		100ドット	N5080.BAR
		125ドット	N5100.BAR
		スケーラブルフォント	BARnw7.BAR
EAN-128 (CR-MENのみ)	104ドット	240ドット	CI_CODE12848.BAR
		スケーラブルフォント	CI_CODE128.BAR
標準料金代理収納用 EAN-128 (CR-MENのみ)	52ドット	240ドット	CI_EAN128S48.BAR
		スケーラブルフォント	CI_EAN128S.BAR

OCR-B	300dpi	A2540. OCR_B
	600dpi	A2032. OCR_B
	スケーラブルフォント	OCR-B. OCR_B

カスタマバーコード	スケーラブルフォント	BARpostal. BAR
-----------	------------	----------------

6.2 フォント仕様一覧

1. JAN、EAN、UPCバーコード、高さ60ドット用文字セット一覧

文字セット名称	フォントオリエンテーション	グラフィックセット番号 (HEX)	文字ピッチ (CPI)	文字サイズ (ポイント)	書体	書体番号	ストロークウェイト	文字スタイル	セル幅×セル高 (ドット)
J2060A. BAR	ポートレート	0F0B	PS	18.0	BARCODE	200	標準	直立体	24×75
J2060B. BAR	〃	〃	14.28	〃	〃	〃	〃	〃	〃
J2060C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J2060D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃
J3060A. BAR	ポートレート	0F0B	PS	18.0	BARCODE	200	標準	直立体	32×75
J3060B. BAR	〃	〃	10.71	〃	〃	〃	〃	〃	〃
J3060C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J3060D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃
J5060A. BAR	ポートレート	0F0B	PS	18.0	BARCODE	200	標準	直立体	48×75
J5060B. BAR	〃	〃	7.14	〃	〃	〃	〃	〃	〃
J5060C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J5060D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃

2. JAN、EAN、UPCバーコード、高さ80ドット用文字セット一覧

文字セット名称	フォントオリエンテーション	グラフィックセット番号 (HEX)	文字ピッチ (CPI)	文字サイズ (ポイント)	書体	書体番号	ストロークウェイト	文字スタイル	セル幅×セル高 (ドット)
J2080A. BAR	ポートレート	0F0B	PS	24.0	BARCODE	200	標準	直立体	24×100
J2080B. BAR	〃	〃	14.28	〃	〃	〃	〃	〃	〃
J2080C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J2080D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃
J3080A. BAR	ポートレート	0F0B	PS	24.0	BARCODE	200	標準	直立体	32×100
J3080B. BAR	〃	〃	10.71	〃	〃	〃	〃	〃	〃
J3080C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J3080D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃
J5080A. BAR	ポートレート	0F0B	PS	24.0	BARCODE	200	標準	直立体	48×100
J5080B. BAR	〃	〃	7.14	〃	〃	〃	〃	〃	〃
J5080C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J5080D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃

※PS=プロポーションアル文字ピッチ

3. JAN、EAN、UPCバーコード、高さ100ドット用文字セット一覧

文字セット名称	フォントイン テション	グラフィック セット番号 (HEX)	文字ピッチ (CPI)	文字サイズ (ポイント)	書体	書体 番号	スローウ ェイト	文字 スタイル	幅×高さ (ドット)
J2100A. BAR	ポートレート	0F0B	PS	30.0	BARCODE	200	標準	直立体	24×125
J2100B. BAR	〃	〃	14.28	〃	〃	〃	〃	〃	〃
J2100C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J2100D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃
J3100A. BAR	ポートレート	0F0B	PS	30.0	BARCODE	200	標準	直立体	32×125
J3100B. BAR	〃	〃	10.71	〃	〃	〃	〃	〃	〃
J3100C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J3100D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃
J5100A. BAR	ポートレート	0F0B	PS	30.0	BARCODE	200	標準	直立体	48×125
J5100B. BAR	〃	〃	7.14	〃	〃	〃	〃	〃	〃
J5100C. BAR	〃	〃	〃	〃	〃	〃	〃	〃	〃
J5100D. BAR	〃	〃	PS	〃	〃	〃	〃	〃	〃

4. JAN、EAN、UPCバーコード、スケーラブルフォント用文字セット一覧

文字セット名称	フォントイン テション	グラフィック セット番号 (HEX)	文字ピッチ (CPI)	文字サイズ (ポイント)	書体	書体 番号	スローウ ェイト	文字 スタイル	幅×高さ (ドット)
BARjan-A. BAR	ポートレート	0F0B	PS	スケーラブル	BARCODE	200			
BARjan-B. BAR	〃	〃	固定	〃	〃	〃			
BARjan-C. BAR	〃	〃	固定	〃	〃	〃			
BARjan-D. BAR	〃	〃	PS	〃	〃	〃			

※PS=プロポーショナル文字ピッチ

5. CODE 39 バーコード用文字セット一覧

文字セット名称	フォントイン テ-ション	グ-ラフィッ セ-ット番号 (HEX)	文字ビ-ツ (CPI)	文字サイズ (ポ-イント)	書体	書体 番号	スロ-ク ウ-イト	文字 スタイル	幅×高さ (ドット)
C2060. BAR	ポ-ト-レ-ト	0F0B	7. 14	18. 0	BARCODE	200	標準	直立体	40×75
C2080. BAR	〃	〃	〃	24. 0	〃	〃	〃	〃	40×100
C2100. BAR	〃	〃	〃	30. 0	〃	〃	〃	〃	40×125
C3060. BAR	ポ-ト-レ-ト	0F0B	5. 17	18. 0	BARCODE	200	標準	直立体	60×75
C3080. BAR	〃	〃	〃	24. 0	〃	〃	〃	〃	60×100
C3100. BAR	〃	〃	〃	30. 0	〃	〃	〃	〃	60×125
C5060. BAR	ポ-ト-レ-ト	0F0B	3. 44	18. 0	BARCODE	200	標準	直立体	88×75
C5080. BAR	〃	〃	〃	24. 0	〃	〃	〃	〃	88×100
C5100. BAR	〃	〃	〃	30. 0	〃	〃	〃	〃	88×125

6. CODE 39 バーコード、スケーラブルフォント用文字セット一覧

文字セット名称	フォントイン テ-ション	グ-ラフィッ セ-ット番号 (HEX)	文字ビ-ツ (CPI)	文字サイズ (ポ-イント)	書体	書体 番号	スロ-ク ウ-イト	文字 スタイル	幅×高さ (ドット)
BARcode39. BAR	ポ-ト-レ-ト	0F0B	固定	ス-ラ-ブル	BARCODE	200			

7. NW-7 バーコード用文字セット一覧

文字セット名称	フォントイン テ-ション	グ-ラフィッ セ-ット番号 (HEX)	文字ビ-ツ (CPI)	文字サイズ (ポ-イント)	書体	書体 番号	スロ-ク ウ-イト	文字 スタイル	幅×高さ (ドット)
N2060. BAR	ポ-ト-レ-ト	0F0B	PS	18. 0	BARCODE	200	標準	直立体	32×75
N2080. BAR	〃	〃	〃	24. 0	〃	〃	〃	〃	32×100
N2100. BAR	〃	〃	〃	30. 0	〃	〃	〃	〃	32×125
N3060. BAR	ポ-ト-レ-ト	0F0B	PS	18. 0	BARCODE	200	標準	直立体	48×75
N3080. BAR	〃	〃	〃	24. 0	〃	〃	〃	〃	48×100
N3100. BAR	〃	〃	〃	30. 0	〃	〃	〃	〃	48×125
N5060. BAR	ポ-ト-レ-ト	0F0B	PS	18. 0	BARCODE	200	標準	直立体	88×75
N5080. BAR	〃	〃	〃	24. 0	〃	〃	〃	〃	88×100
N5100. BAR	〃	〃	〃	30. 0	〃	〃	〃	〃	88×125

8. NW-7 バーコード、スケーラブルフォント用文字セット一覧

文字セット名称	フォントイン テ-ション	グ-ラフィッ セ-ット番号 (HEX)	文字ビ-ツ (CPI)	文字サイズ (ポ-イント)	書体	書体 番号	スロ-ク ウ-イト	文字 スタイル	幅×高さ (ドット)
BARnw7. BAR	ポ-ト-レ-ト	0F0B	PS	ス-ラ-ブル	BARCODE	200			

※PS=プロポーショナル文字ピッチ

9. EAN-128バーコード用文字セット一覧 (CR-MENのみ)

文字セット名称	フォントリ ネーション	グラフィッ セット番 号 (HEX)	文字ピ ッチ (CPI)	文字サイ ズ (ポ イント)	書 体	書 体 番 号	スト ーク ウェ イト	文 字 ス タ イル	幅 × 高 (ド ット)
CI_CODE12848.BAR		0FOB	5.77	48	BARCODE	192	標準	直立体	104×240

10. EAN-128バーコード、スケーラブルフォント用文字セット一覧 (CR-MENのみ)

文字セット名称	フォントリ ネーション	グラフィッ セット番 号 (HEX)	文字ピ ッチ (CPI)	文字サイ ズ (ポ イント)	書 体	書 体 番 号	スト ーク ウェ イト	文 字 ス タ イル	幅 × 高 (ド ット)
CI_CODE128.BAR		0FOB	PS	ス ケ ー ラ ブ ル	BARCODE	190	標準	直立体	800×800

11. 標準料金代理収納用EAN-128バーコード用文字セット一覧 (CR-MENのみ)

文字セット名称	フォントリ ネーション	グラフィッ セット番 号 (HEX)	文字ピ ッチ (CPI)	文字サイ ズ (ポ イント)	書 体	書 体 番 号	スト ーク ウェ イト	文 字 ス タ イル	幅 × 高 (ド ット)
CI_EAN128S48.BAR		0FOB	11.54	24	BARCODE	193	標準	直立体	52×240

12. 標準料金代理収納用EAN-128バーコード、スケーラブルフォント用文字セット一覧 (CR-MENのみ)

文字セット名称	フォントリ ネーション	グラフィッ セット番 号 (HEX)	文字ピ ッチ (CPI)	文字サイ ズ (ポ イント)	書 体	書 体 番 号	スト ーク ウェ イト	文 字 ス タ イル	幅 × 高 (ド ット)
CI_EAN128S.BAR		0FOB	PS	ス ケ ー ラ ブ ル	BARCODE	191	標準	直立体	800×800

※PS=プロポーション文字ピッチ

13. EAN-128バーコードキャラクタ構成 (CR-MENのみ)

下記の「EAN-128バーコードキャラクタ構成」は、付属のCD-ROMの[EAN128_Character_Map]フォルダに収められています。

数値	キャラクタ			記号表示*1*2		Map	バーコードキャラクタ
	コト`セツ"A"	コト`セツ"B"	コト`セツ"C"	BSB	SBS		
0	(sp)	(sp)	00	212	222	0Y	
1	!	!	01	222	122	4K	
2	"	"	02	222	221	4X	
3	#	#	03	121	223	%Z	
4	\$	\$	04	121	322	%e	
5	%	%	05	131	222)Y	
6	&	&	06	122	213	&V	
7	'	'	07	122	312	&b	
8	((08	132	212	*U	
9))	09	221	213	3V	
10	*	*	10	221	312	3b	
11	+	+	11	231	212	6U	
12	,	,	12	112	232	"^	
13	-	-	13	122	132	&O	
14	.	.	14	122	231	&]	
15	/	/	15	113	222	#Y	
16	0	0	16	123	122	'K	
17	1	1	17	123	221	'X	
18	2	2	18	223	211	5T	
19	3	3	19	221	132	3O	
20	4	4	20	221	231	3]	
21	5	5	21	213	212	1U	
22	6	6	22	223	112	5G	
23	7	7	23	312	131	:N	
24	8	8	24	311	222	9Y	
25	9	9	25	321	122	=K	
26	:	:	26	321	221	=X	
27	;	;	27	312	212	:U	
28	<	<	28	322	112	>G	
29	=	=	29	322	211	>T	
30	>	>	30	212	123	0L	
31	?	?	31	212	321	0d	
32	@	@	32	232	121	7J	
33	A	A	33	111	323	!f	
34	B	B	34	131	123)L	
35	C	C	35	131	321)d	
36	D	D	36	112	313	"c	
37	E	E	37	132	113	*H	
38	F	F	38	132	311	*a	
39	G	G	39	211	313	/c	
40	H	H	40	231	113	6H	
41	I	I	41	231	311	6a	
42	J	J	42	112	133	"P	
43	K	K	43	112	331	"g	
44	L	L	44	132	131	*N	
45	M	M	45	113	123	#L	
46	N	N	46	113	321	#d	
47	O	O	47	133	121	+J	
48	P	P	48	313	121	;J	
49	Q	Q	49	211	331	/g	
50	R	R	50	231	131	6N	
51	S	S	51	213	113	1H	
52	T	T	52	213	311	1a	
53	U	U	53	213	131	1N	

*1 Bは黒バーを表し、Sは白バーを表す。

*2 数字は、エレメントを表す数値を示す。

54	V	V	54	311 123	9L	■■■■ ■■■■
55	W	W	55	311 321	9d	■■■■ ■■■■
56	X	X	56	331 121	?J	■■■■ ■■■■
57	Y	Y	57	312 113	:H	■■■■ ■■■■
58	Z	Z	58	312 311	:a	■■■■ ■■■■
59	[[59	332 111	@F	■■■■ ■■■■
60	*3	*3	60	314 111	<F	■■■■ ■■■■
61]]	61	221 411	3i	■■■■ ■■■■
62	^	^	62	431 111	EF	■■■■ ■■■■
63	~	~	63	111 224	![■■■■ ■■■■
64	NUL	~	64	111 422	!!	■■■■ ■■■■
65	SOH	a	65	121 124	%M	■■■■ ■■■■
66	STX	b	66	121 421	%k	■■■■ ■■■■
67	ETX	c	67	141 122	-K	■■■■ ■■■■
68	EOT	d	68	141 221	-X	■■■■ ■■■■
69	ENQ	e	69	112 214	"W	■■■■ ■■■■
70	ACK	f	70	112 412	"j	■■■■ ■■■■
71	BEL	g	71	122 114	&l	■■■■ ■■■■
72	BS	h	72	122 411	&i	■■■■ ■■■■
73	HT	i	73	142 112	.G	■■■■ ■■■■
74	LF	j	74	142 211	.T	■■■■ ■■■■
75	VT	k	75	241 211	8T	■■■■ ■■■■
76	FF	l	76	221 114	3l	■■■■ ■■■■
77	CR	m	77	413 111	CF	■■■■ ■■■■
78	SO	n	78	241 112	8G	■■■■ ■■■■
79	SI	o	79	134 111	,F	■■■■ ■■■■
80	DLE	p	80	111 242	!'	■■■■ ■■■■
81	DC1	q	81	121 142	%R	■■■■ ■■■■
82	DC2	r	82	121 241	%	■■■■ ■■■■
83	DC3	s	83	114 212	\$U	■■■■ ■■■■
84	DC4	t	84	124 112	(G	■■■■ ■■■■
85	NAK	u	85	124 211	(T	■■■■ ■■■■
86	SYN	v	86	411 212	AU	■■■■ ■■■■
87	ETB	w	87	421 112	DG	■■■■ ■■■■
88	CAN	x	88	421 211	DT	■■■■ ■■■■
89	EM	y	89	212 141	0Q	■■■■ ■■■■
90	SUB	z	90	214 121	2J	■■■■ ■■■■
91	ESC	{	91	412 121	BJ	■■■■ ■■■■
92	IS4(FS)		92	111 143	IS	■■■■ ■■■■
93	IS3(GS)	}	93	111 341	!h	■■■■ ■■■■
94	IS2(RS)	~*4	94	131 141)Q	■■■■ ■■■■
95	IS1(US)	DEL	95	114 113	\$H	■■■■ ■■■■
96	FNC 3	FNC 3	96	114 311	\$a	■■■■ ■■■■
97	FNC 2	FNC 2	97	411 113	AH	■■■■ ■■■■
98	SHIFT	SHIFT	98	411 311	Aa	■■■■ ■■■■
99	CODE C	CODE C	99	113 141	#Q	■■■■ ■■■■
100	CODE B	FNC 4	CODE B	114 131	\$N	■■■■ ■■■■
101	FNC 4	CODE A	CODE A	311 141	9Q	■■■■ ■■■■
102	FNC 1	FNC 1	FNC 1	411 131	AN	■■■■ ■■■■
103		スタートA		211 412	/j	■■■■ ■■■■
104		スタートB		211 214	/W	■■■■ ■■■■
105		スタートC		211 232	/^	■■■■ ■■■■
106		ストップ		233 1112	{	■■■■ ■■■■

*3 JIS X 0201では、'¥'に相当する。

*4 JIS X 0201では、'~'に相当する。

14. OCR_Bフォント用文字フォント一覧

文字セット名称	フォントリイ ネーション	グラフィック セット番号 (HEX)	文字ピッチ (CPI)	文字サイズ (ポイント)	書体	書体 番号	スローウ ェイト	文字 スタイル	幅×高さ (ドット)
A2032. OCR_B (600dpiフォント)	ポートレート	0F0B	12.0	9.6	BARCODE	200	標準	直立体	50×80
A2540. OCR_B (300dpiフォント)	〃	〃	〃	〃	〃	〃	〃	〃	25×40

15. OCR_Bスケラブルフォント用文字フォント一覧

文字セット名称	フォントリイ ネーション	グラフィック セット番号 (HEX)	文字ピッチ (CPI)	文字サイズ (ポイント)	書体	書体 番号	スローウ ェイト	文字 スタイル	幅×高さ (ドット)
OCR-B. OCR_B	ポートレート	0F0B	固定	スケラブル	BARCODE	200			

16. OCR_Bフォントコード表一覧

	0	1	2	3	4	5	6	7
0				0	@	P		
1			!	1	A	Q		
2			"	2	B	R		
3			#	3	C	S		
4			\$	4	D	T		
5			%	5	E	U		
6			&	6	F	V		
7			'	7	G	W		
8			(8	H	X		
9)	9	I	Y		
A			*	:	J	Z		
B			+	;	K	[
C			,	<	L	¥		
D			-	=	M]		
E			.	>	N	^		
F			/	?	O	_		

注意：空白の文字コードの印字は、スペースに置き換わります。

17. カスタマバーコードスケラブルフォント用文字フォント一覧

文字セット名称	フォントリ ネーション	グ ラ フ ィ ツ セ ツ 番 号 (HEX)	文字ピ ツ 寸 (CPI)	文字サ イ ズ (ポ ィ ツ)	書 体	書 体 番 号	ス ケ ー リ ン グ	文 字 ス タ イ ル	セ ル 幅 × セ ル 高 (ド ツ ツ)
BARpostal.BAR	ポ ー ト レ ー ト	0F0B	固 定	ス ケ ー ラ ブル	BARCODE	200			

18. カスタマバーコードフォントコード表一覧

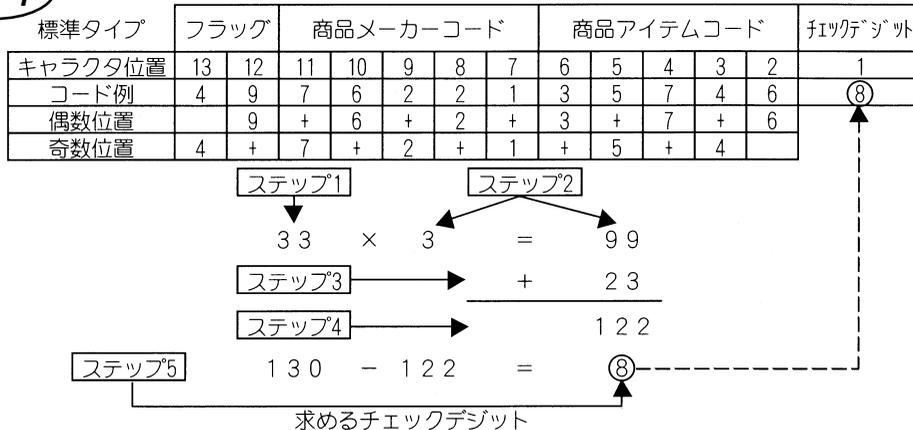
	0	1	2	3	4	5	6	7
0				▬▬				
1				▬▬	▬▬			
2				▬▬	▬▬			
3				▬▬	▬▬			
4				▬▬	▬▬			
5				▬▬	▬▬			
6				▬▬	▬▬			
7				▬▬	▬▬			
8			▬▬	▬▬	▬▬			
9			▬▬	▬▬				
A								
B								
C								
D			▬▬					
E								
F								

6.3 JANコードチェックデジット計算方法

各キャラクタはその位置に従い、右側から左方向に順次番号付けする（チェックデジットが1番目）。

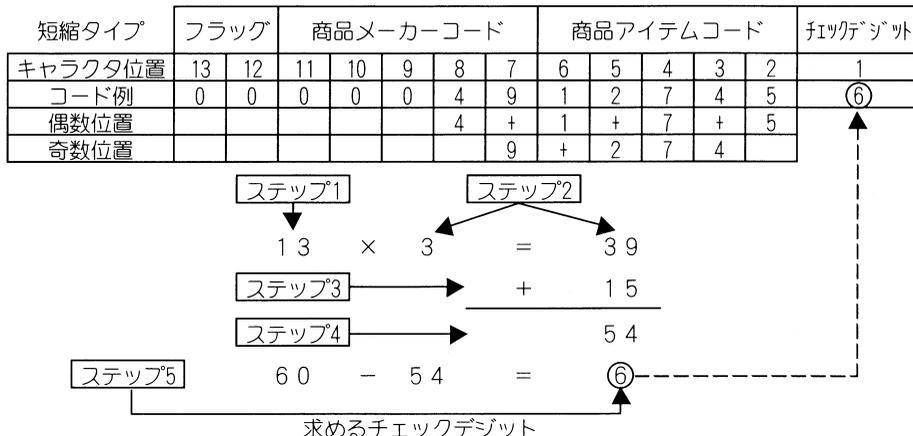
- ステップ1： 2番目のキャラクタから始めて、すべての偶数番キャラクタの値の和を取る。
- ステップ2： ステップ1の結果を3倍する。
- ステップ3： 3番目のキャラクタから始めて、すべての奇数番キャラクタの値の和を取る。
- ステップ4： ステップ2とステップ3の和を取る。
- ステップ5： ステップ4で得た値よりも大きく、かつ最も近い10の倍数を求める。その値とステップ4の値の差が求めるチェックデジットの数値となる。

例 1



例 2

短縮タイプでは、キャラクタ位置を右詰めとして(キャラクタ位置1～13に“0”を入れて)計算する。



注意：なお、ステップ4で得られた値が10の倍数の場合は、求めるチェックデジットは0となります。

6.4 UPC短縮バーコードチェックデジット計算方法

1. UPC短縮コードをメーカーコード+アイテムコードの6桁目によって表1の6通りの方法でUPC標準コードに変換します。
2. 以下、JANコードのチェックデジットの計算方法に従って、チェックデジットを求めます。

表1

UPC短縮コード (Eバージョン)

変換パターン	キャラクターシステム ナンバー (1桁)	メーカーコード+ アイテムコード (6桁)	チェックデジット (1桁)
1	0	F ₁ F ₂ A ₃ A ₄ A ₅ 0	CD
2	0	F ₁ F ₂ A ₃ A ₄ A ₅ 1	CD
3	0	F ₁ F ₂ A ₃ A ₄ A ₅ 2	CD
4	0	F ₁ F ₂ A ₃ A ₄ A ₅ 3	CD
5	0	F ₁ F ₂ A ₃ A ₄ A ₅ 4	CD
6	0	F ₁ F ₂ A ₃ A ₄ A ₅ A ₅ *	CD

※A₅ = 5~9

UPC標準コード (Aバージョン)

キャラクターシステム ナンバー (1桁)	メーカーコード (5桁)	アイテムコード (5桁)	チェックデジット (1桁)
0	F ₁ F ₂ 0 0 0	0 0 F ₃ F ₄ F ₅	CD
0	F ₁ F ₂ 1 0 0	0 0 F ₃ F ₄ F ₅	CD
0	F ₁ F ₂ 2 0 0	0 0 F ₃ F ₄ F ₅	CD
0	F ₁ F ₂ F ₃ 0 0	0 0 0 F ₄ F ₅	CD
0	F ₁ F ₂ F ₃ F ₄ 0	0 0 0 0 F ₅	CD
0	F ₁ F ₂ F ₃ F ₄ F ₅	0 0 0 0 F ₅	CD

例 1

UPC短縮コード 0 1 2 3 4 5 6 CDの場合
 UPC標準コード 0 1 2 3 4 5 0 0 0 6 CDに変換
 (表1のパターン6の変換)

以下、JANコードの計算方法により

ステップ1 $0 + 2 + 4 + 0 + 0 + 6 = 12$

ステップ2 $12 \times 3 = 36$

ステップ3 $1 + 3 + 5 + 0 + 0 = 9$

ステップ4 $36 + 9 = 45$

ステップ5 $50 - 45 = 5$

CD = 5 になります。

6.5 EAN-128チェックデジット計算方法

- ステップ1： ストップキャラクタ以外のすべてのキャラクタに対する数値を得る。数値は、「6.2 フォント仕様一覧」の「13. EAN-128バーコードキャラクタ構成 (CR-MENのみ)」より得る。
- ステップ2： スタートキャラクタおよび最初のデータキャラクタは数値×1をし、以降のキャラクタには数値×2、×3、・・・、×Nをする。
- ステップ3： ステップ2で求めた値の和を取る。
- ステップ4： ステップ3で求めた値を103で割り、余りを求める。
- ステップ5： ステップ4で求めた余りと同じ数値に対応するバーコードキャラクタをチェックキャラクタとする。

例

以下のバーコードデータの場合

スタートA	FNC1	7	B	K	チェックデジット	ストップキャラクタ
-------	------	---	---	---	----------	-----------

↑
ここを求める

	キャラクタ				
	スタートA	FNC1	7	B	K
ステップ1	103	102	23	34	43
ステップ2	103×1 =103	102×1 =102	23×2 =46	34×3 =102	43×4 =172
ステップ3	103+102+46+102+172=525				
ステップ4	525÷103=5余り 10				
ステップ5	数値10に対応するバーコードキャラクタがチェックキャラクタ				

6.6 EAN-128アプリケーション識別子一覧

項目	AI	フォーマット	固定 / 可変	備考	
標準カートン (SSCC-18)	00	n2+n18	固定	n18はチェックデジットを含む桁数	
出荷コンテナコード (SCC-14)	01	n2+n14	固定	n14はチェックデジットを含む桁数	
	02	n2+n14	固定	14桁未満は頭に0を付加。AI'37'と併用。	
バッチ/ロットナンバー	10	n2+an...20	可変		
年月日	製造年月日	n2+n6	固定	n6部分はYYMMDDの形式。 YYMMのみの表記の場合は、YYMM00となる。	
	包装年月日	n2+n6	固定		
	販売期限日	n2+n6	固定		
	保証期限日	n2+n6	固定		
リニューアル商品	20	n2+n2	固定		
連続番号	21	n2+an...20	可変		
	22	n2+an...29	可変		
	23*	n3+an...19	可変	データの長さ(最大19桁)を示す1桁(N)をAIに追加して、3桁表示となる。 Nは、「データ長=2N+1」で決定。 例: データが1234567の7桁の場合 7=2N+1→N=3 となり、「(233) 1234567」となる。	
	240	n3+an...30	可変		
	241	n3+an...30	可変		
	250	n3+an...30	可変		
数量	30	n2+n...8	可変		
計量表示	キログラム	310*	n4+n6	固定	小数点を示すために、AIに1桁追加。 重量14.123Kgを表現する場合、小数点以下3桁のデータなので、AI'310'をAI'3103'にし、14.123Kgは小数点なしの14123とし、さらに、n6桁固定なので、先頭に0を塚て'014123'とする。
	メートル	311*	n4+n6	固定	
		312*	n4+n6	固定	
		313*	n4+n6	固定	
		314*	n4+n6	固定	
	リットル	315*	n4+n6	固定	
	立方メートル	316*	n4+n6	固定	
	ポンド	320*	n4+n6	固定	
	インチ	321*	n4+n6	固定	
	フィート	322*	n4+n6	固定	
	ヤード	323*	n4+n6	固定	
	インチ	324*	n4+n6	固定	
	フィート	325*	n4+n6	固定	
	ヤード	326*	n4+n6	固定	
	インチ	327*	n4+n6	固定	
	フィート	328*	n4+n6	固定	
	ヤード	329*	n4+n6	固定	
	キログラム	330*	n4+n6	固定	
	メートル換算	331*	n4+n6	固定	
		332*	n4+n6	固定	
		333*	n4+n6	固定	
		334*	n4+n6	固定	
	リットル換算	335*	n4+n6	固定	
	メートル換算	336*	n4+n6	固定	
	キログラム/平方メートル	337*	n4+n6	固定	
	ポンド	340*	n4+n6	固定	
	インチ換算	341*	n4+n6	固定	
	フィート換算	342*	n4+n6	固定	
	ヤード換算	343*	n4+n6	固定	
	インチ換算	344*	n4+n6	固定	
	フィート換算	345*	n4+n6	固定	
	ヤード換算	346*	n4+n6	固定	
	インチ換算	347*	n4+n6	固定	
	フィート換算	348*	n4+n6	固定	
	ヤード換算	349*	n4+n6	固定	
平方インチ	350*	n4+n6	固定		
平方フィート	351*	n4+n6	固定		

計量表示	平方ヤード	352*	n4+n6	固定	
	平方インチ換算	353*	n4+n6	固定	
	平方フィート換算	354*	n4+n6	固定	
	平方ヤード換算	355*	n4+n6	固定	
	トロイオンス	356*	n4+n6	固定	
	クォート	360*	n4+n6	固定	
	ガロン	361*	n4+n6	固定	
	クォート	362*	n4+n6	固定	
	ガロン	363*	n4+n6	固定	
	立方インチ	364*	n4+n6	固定	
	立方フィート	365*	n4+n6	固定	
	立方ヤード	366*	n4+n6	固定	
	立方インチ	367*	n4+n6	固定	
	立方フィート	368*	n4+n6	固定	
立方ヤード	369*	n4+n6	固定		
数量	37	n2+n...8	可変	A1"02"と併用。	
業務管理番号	400	n3+an...30	可変		
	401	n3+an...30	可変		
ロケーションナンバー	410	n3+n13	固定		
	411	n3+n13	固定		
	412	n3+n13	固定		
	413	n3+n13	固定		
	414	n3+n13	固定		
国内の送り先所在地コード	420	n3+an...9	可変		
海外の送り先所在地コード	421	n3+a2+an...9	可変	a2=2桁のISOコード。	
運送貨物番号	43	n2+n4+n7+an6...10+n1	可変		
計量単位	8001	n4+n14	固定		
携帯電話連続番号	8002	n4+an...20	可変		
返品管理番号	8003	n4+n14+an...16	可変		
一般的な連続番号	8004	n4+an...30	可変		
価格表示	8005	n4+n6	固定		
構成部品の管理番号	8006	n4+n14+n2+n2	固定		
サービス関連番号	8018	n4+n18	固定		
クーポン拡張コード	8100	n4+n1+n5	固定		
	8101	n4+n1+n5+n4	固定		
	8102	n4+n1+n1	固定		
FACTデータ識別子	90	n2+an...4+an...26	可変		
企業内部使用	91	n2+an...30	可変		
	92	n2+an...30	可変		
	93	n2+an...30	可変		
	94	n2+an...30	可変		
	95	n2+an...30	可変		
	96	n2+an...30	可変		
	97	n2+an...30	可変		
	98	n2+an...30	可変		
	99	n2+an...30	可変		
a : 英字 n : 数字 an : 英数字 ... : 可変長					

Canon