Intermec

Programmer's Reference Manual

ESim v5.12 for EasyCoder® PD4 Bar Code Label Printer

Intermec Technologies Corporation
Corporate Headquarters
6001 36th Ave. W.
Everett, WA 98203
U.S.A.
www.intermec.com

The information contained herein is proprietary and is provided solely for the purpose of allowing customers to operate and service Intermec-manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Intermec.

Information and specifications contained in this document are subject to change without prior notice and do not represent a commitment on the part of Intermec Technologies Corporation.

The word Intermec, the Intermec logo, Norand, ArciTech, CrossBar, Data Collection Browser, dcBrowser, Duratherm, EasyADC, EasyCoder, EasyLAN, Enterprise Wireless LAN, EZBuilder, Fingerprint, i-gistics, INCA (under license), InterDriver, Intermec Printer Network Manager, IRL, JANUS, LabelShop, Mobile Framework, MobileLAN, Nor*Ware, Pen*Key, Precision Print, PrintSet, Ready-to-Work, RoutePower, Smart-Systems, TE 2000, Trakker Antares, and Virtual Wedge are either trademarks or registered trademarks of Intermec Technologies Corporation.

Throughout this manual, trademarked names may be used. Rather than put a trademark (™ or ®) symbol in every occurrence of a trademarked name, we state that we are using the names only in an editorial fashion, and to the benefit of the trademark owner, with no intention of infringement.

There are U.S. and foreign patents pending.

The name Centronics is wholly owned by GENICOM Corporation.
Windows is a trademark of Microsoft Corporation.

Contents

1 General Information

2 Commands List

3 Setting Up the Printer

4 Editing in Direct Mode

5 Editing in Form Edit Mode

6 Retrieving and Printing a Form

7 Commands

A Error-Handling

1 General Information

This chapter explains the basic functions and modes of the ESim v5.12 protocol and helps understanding the more comprehensive explanations in the other chapters in this manual.

Introduction

The EasyCoder PD4 printers from Intermec are provided with a built-in protocol (ESim) by which you can use any computer, terminal, scanner or keyboard, that can produce ASCII characters, to control the printer. This is a useful alternative to the Intermec InterDriver, which requires a PC operating under Microsoft Windows.

With the ESim protocol, you can use any editor to control the printer, either by means of the serial RS-232 channel, the parallel Centronics channel, or the USB channel.

Note that EasyCoder PD4 has a flash memory for forms and graphics, which requires special consideration. Avoid storing frequently changing data in flash (see **GM** and **GW** commands in Chapter 7) and use printer drivers developed for EasyCoder C4 and PD4.

This manual will assist you in designing labels using the ESim protocol. It has been organized to provide you with an understanding of the printer's functions and command structure.

The manual describes version 5.12 of the ESim protocol.

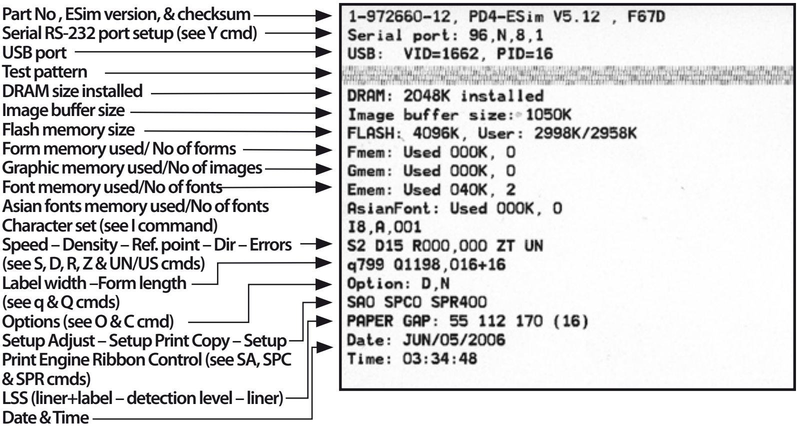
If you have any questions regarding the protocol or this manual, please contact your Intermec distributor for technical assistance.

Test Label

The best way to start programming is to find out how the printer is set up. If you have an EasyCoder PD4 with a display, you can browse the Setup Mode and see some parameters, but the most comprehensive information is obtained by printing the test labels.

To print test labels:

- Switch off the power to the printer.
- For best result, load the printer with full width labels or tags.
- Hold down the **Feed** key and turn on the power again.
- Release the **Feed** key within 3 seconds.
- The printer emits a beep and "Self test" is shown in the display, if any.
- A single test label is printed.
- After the test label has been printed, the printer automatically enters the Direct Mode.



```
1-972660-12, PD4-ESim V5.12 , F67D
Serial port: 96,N,8,1
USB:  VID=1662, PID=16

DRAM: 2048K installed
Image buffer size: 1050K
FLASH: 4096K, User: 2998K/2958K
Fmem: Used 000K, 0
Gmem: Used 000K, 0
Emem: Used 040K, 2
AsianFont: Used 000K, 0
I8,A,001
S2 D15 R000,000 ZT UN
q799 Q1198,016+16
Option: D,N
SA0 SPC0 SPR400
PAPER GAP: 55 112 170 (16)
Date: JUN/05/2006
Time: 03:34:48
```

Part No , ESim version, & checksum
Serial RS-232 port setup (see Y cmd)
USB port
Test pattern
DRAM size installed
Image buffer size
Flash memory size
Form memory used/ No of forms
Graphic memory used/No of images
Font memory used/No of fonts
Asian fonts memory used/No of fonts
Character set (see I command)
Speed – Density – Ref. point – Dir – Errors (see S, D, R, Z & UN/US cmds)
Label width –Form length (see q & Q cmds)
Options (see O & C cmd)
Setup Adjust – Setup Print Copy – Setup Print Engine Ribbon Control (see SA, SPC & SPR cmds)
LSS (liner+label – detection level – liner)
Date & Time

Example of a test label from an 203.2 dpi (8 dots/mm) EasyCoder PD4.

Dump Mode

The printer has the capability to perform in dump mode, which means that the printer will print out the echo of the received ASCII. Use this capability to debug your software when the printer does not perform as you expect.

To enter Dump Mode:

- Switch off the power to the printer.
- Hold down the **Feed** key and turn on the power again.
- Keep on holding **Feed** key for more than 2 seconds.
- Send a string of characters or a label form to the printer and tap the Feed key to produce a printout. Dump mode will also print control characters, see character set table on at the end of Chapter 9.
- To return to normal mode, briefly tap the Feed key.

Memory

The firmware has dynamic memory allocation for print image buffer, form, graphic, and external font memory. The first time the printer is used, it is automatically initialized to default settings, see Chapter 3.

Direct Mode

You can print a label without using a predefined format by sending write commands (text, bar codes, graphics, lines and boxes) to the printer after having cleared the image buffer using an **N** command. The label remains stored in the image buffer and can be printed over and over again by sending new **P** print commands, until the buffer is cleared by an **N** command, or by retrieving and printing a Form (see **FR** command).

The Direct Mode is also used for retrieving and printing preprogrammed label formats, for the issuing of global setup commands, for deleting forms and graphics from memory, and to make the printer produce a number of different reports.

Form Edit Mode

This mode is used to permanently store label forms and graphics in the printer memory. In addition to plain text, bar codes, graphics, lines, and boxes, form edit mode also allows the use of variables and counters, which are not available in the Direct Mode. The individual label forms can be retrieved and printed in the Direct Mode.

Some setup parameters can be included in forms in order to adapt the printer for different applications. However, such setup parameters will affect the global setup after the form has been retrieved and printed.

Form

Every label is made up of various fields. A form is the complete set of commands that define the content and the design of the label. A form can be saved in memory and retrieved when required.

Text Editor

Use any ASCII output device with a parallel, serial, or USB port and a text editor to design the form and programming the printer. Communication is based on the 8 bit ASCII characters 10 dec. and 32-255 dec.

Commands

The command syntax is based on upper and lower case characters, numeric characters, commas (as separators), quotation marks and line feeds (LF; ASCII 10 dec.). The LF in this manual is listed as ↵ in the command descriptions. Some commands require to be appended by both a carriage return and a line feed (<CR/LF>).

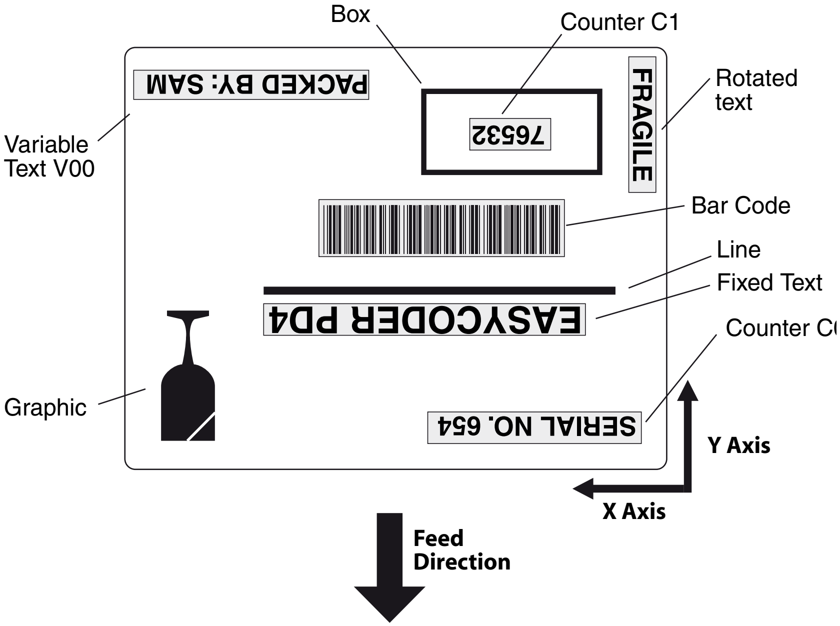
Note that all programming examples start with LF (depicted as ↵). It is strongly recommended to start any sequence of command lines with a Line Feed (LF).

Most PC based systems send CR/LF when pressing the <Enter> key. The CR (carriage return) sent in a CR/LF sequence will be ignored. CR alone will not work.

Refer to Chapter 2 for a list showing for which purposes the various commands can be used.

Field

Each command line of printable data will create a field, which is defined in regard of start position, rotation, magnification, etc.



The illustration shows how a label is printed and fed out when using the default direction.

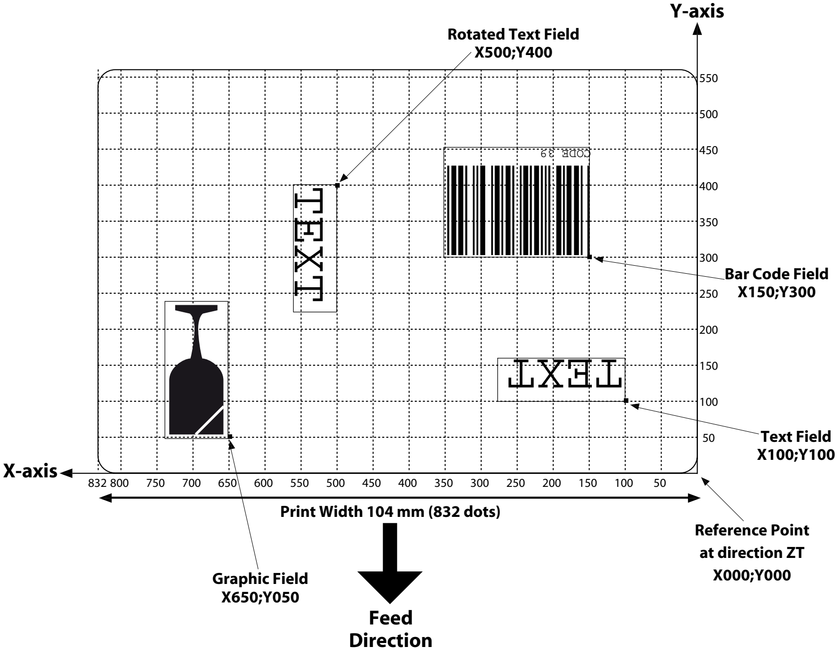
Field Positioning

The printable area of the label forms a grid, where the X-axis runs across the media and the Y-axis runs along the media path. Dots are used as the unit for establishing position of the upper left corner of each field in relation to a specified reference point.

The printer is available with two different printhead densities (the 203.2 dpi printer and the 300 dpi printer have different firmware versions):

Density	Medium	High
X-axis	203.2 dpi (8 dots/mm)	300 dpi (11.81 dots/mm)
Y-axis	203.2 dpi (8 dots/mm)	304.8 dpi (12 dots/mm)
Number of active dots:	832	1208
Max. print width:	104 mm	102.2 mm
Dot line to tear bar:	136 dots	204 dots
Dot line to cutter blade:	256 dots	384 dots

Text and bar code fields can be rotated around their insertion points, whereas lines, boxes and graphics cannot be rotated. However, the entire print image can be rotated 180°. The illustration below shows coordinates for the default print direction (**ZT**).



Example of field positioning when using an 203.2 dpi (8 dots per mm) printhead.

2 Commands List

This chapters lists which commands can be used in the Direct Mode and the Form Edit Mode respectively.

Direct Mode

Setup Commands

Used to set up the printer globally, that is, affect both the Direct Mode and Forms.

D	Density
eR	User-Defined Error/Status Character Control
f	Cut Position
I	Character Set Selection
i	Asian Character Spacing
JB	Disable Top of Form Backup
JF	Enable Top of Form Backup
j	Media Feed Adjustment
O	Options Select
oR	Character Substitution
Q	Set Form Length
q	Set Label Width
R	Set Reference Point
r	Set Relative Reference Point
S	Speed Select
SA	Setup Adjust
SPC	Setup Print Copy
SPR	Setup Print Engine Ribbon Control
TD	Set Date Format
TS	Set Real Time Clock
TT	Set Time Format
UC	Command Reply Accept Character Control
UN	Disable Error Reporting
US	Enable Error Reporting
W	Windows Mode
Y	Serial Port Setup
Z	Print Direction

Store Commands

GM	Store Graphics in Memory
GW	Store Graphics in Image Buffer
ES	Store Soft Font

Clear and Delete Commands

Used to erase data from the printer's memory.

EK	Delete Soft Font

FK	Delete Form
GK	Delete Graphics
N	Clear Image Buffer
^default	Restore Factory Default
^@	Reset Printer

Editing Commands

Used to edit labels in the Direct Mode.

A	Print Text
B	Print Standard Bar Codes
b	Print Two-Dimensional Codes
GG	Print Graphics
LE	Line Draw Exclusive
LO	Line Draw Black
LS	Line Draw Diagonal
LW	Line Draw White
X	Draw Box

Print Commands

Used to produce printouts of labels edited in the Direct or retrieved form edited in the Form Edit Mode.

C	Cut Immediate
FR	Retrieve Form
P	Print
PF	Feed Media
?	Download Variables

Report Commands

Return information on serial channel and/or produce printed information.

EI	List Soft Fonts
FI	Print Form Information
GI	Print Graphics Information
GRP	Reply Graphics from Printer Buffer
U	Print Configuration
UE	Soft Fonts Information Inquiry
UF	Form Information Inquiry
UG	Graphics Information Inquiry
UI	Enable Prompts/Code Page Inquiry
UM	Code Page & Memory Inquiry
UP	Code Page & Memory Inquiry/Print
UV	Product Identity and Asian Font Types
^ee	Immediate Error Report

Form Edit Mode

Setup Commands in Forms

Will affect the global setup after printing a form including such a command.

D	Density
Q	Set Form Length
R	Set Reference Point
S	Speed Select
Z	Print Direction

Editing Commands

Used to edit forms.

A	Print Text
B	Print Standard Bar Code
b	Print Two-Dimensional Codes
C	Counter
FE	End Form Store
FS	Form Store
GG	Print Graphics
LE	Line Draw Exclusive
LO	Line Draw Black
LS	Line Draw Diagonal
LW	Line Draw White
PA	Print Automatic
V	Define Variable
X	Draw Box

3 Setting Up the Printer

This chapter shows the default setup of the printer and gives an example of how to change the setup.

Default Setup

Parameter	Command	Meaning
Error/Status Char.	**eR**0,0	XOFF/XON-handling
Density	**D**10	10
Character Set	**I**8,0,001	8 bits, code page 0, country code 001
Top of Form backup	**JF**	Enabled
Media feed adjust	**j**136 **j**204	136 dots (at 203.2 dpi) 204 dots (at 300 dpi)
Options	**O**N,D	LTS disabled, DT mode, reverse gap disabled, cutter disabled
Char. substitution	**oR**	No substitution
Form Length	**Q**24,0	Continuous mode, print gap 24 dots
Label Width	**q**832 **q**1208	832 dots (at 203.2 dpi) 1208 dots (at 300 dpi)
Reference Point	**R**000,000	X:000;Y000
Print Speed	**S**3	75 mm/sec. (3 inches/sec.)
Date Format	**TD**me/dd/Y4	For example JAN/15/2004
Time Format	**TT**h:m:s	For example 13:25:00
Reply character	**UC**0	None
Error Handling	**US**	Error reporting enabled
Windows Mode	**W**N	Disabled
Serial Port	**Y**96,N,8,1	9600 baud, no parity, 8 data bits, 1 stop bit
Print Direction	**Z**T	Start printing at top of image buffer

Printhead Resolution	Measures
203.2 dpi printheads	203.2 dpi (8 dots/mm) in both directions
300 dpi printheads	300 dpi (11.81 dots/mm) across the media path 304.8 dpi (12 dots/mm) along the media path

The setup will be reset to default values if the **Cancel** key is pressed more than 3 seconds in the Dump Mode.

Some commands may also affect the values of other command, for example if a configuration label is printed (see **U** and **UP** commands), the print direction is reset to **ZT**, and if an **R** Reference Point command is executed, the label width (see **q** command) will be changed.

Example

Let us assume that we will use an EasyCoder PD4 for direct thermal printing. We will print full width Thermal Top labels in the peel-off mode without using the label taken sensor. The default communication setup and character set are acceptable.

Thus, a few setup parameters should be changed in the Direct Mode:

- Density from 10 to 8
- Media feed adjustment from 136 to 110
- Label Taken Sensor from enabled to disabled

Enter the following commands:

Command	Explanation
↵	CR/LF to start command structure
D8 ↵	Set density
j110 ↵	Set media feed adjustment for peel-off operation
ON ↵	Disable label taken sensor

4 Editing in Direct Mode

This chapter gives an example of how to create a simple label in the Direct Mode.

Example

Assuming that...

- the printer has been set up for the application (see Chapter 3),
- the length of the label and the gap has been determined using the Autosensing Mode (see *EasyCoder PD4 User's Guide*),
- and the graphic used in the example has been downloaded using a **GM** command[1]),

...we will now print two copies of a label which we will edit in the Direct Mode. The label will look like the example in Chapter 5.

The label can be printed as many times as you want, as long as it still is stored in the image buffer. Once replaced, it cannot be retrieved. It also implies that counters and variables cannot be used.

Command	Explanation
↵	CR/LF to start command structure
N↵	Clear image memory
X0,0,4,752,584↵	Draw a box
LO0,144,752,4↵	Draw a line
LO440,232,4,160↵	Draw a line
A40,400,1,1,1,1,N,"Made in Sweden"↵	Write a 90° text line of fixed data
A24,160,0,5,1,1,R,"EASYCODER"↵	Write a text line of fixed data
A24,250,0,4,1,1,N,"MODEL: 501SA"↵	Write a text line of fixed data
A472,312,0,4,1,1,N,"Checked by: Dan"↵	Write a text line of fixed data
A24,312,0,4,1,1,N,"SERIAL#: 000001"↵	Write a text line of fixed data
B280,440,0,1,2,3,96,B,"S 000001"↵	Write barcode representing fixed data
GG24,12,"LOGO"↵	Write a graphic from graphics memory[1]
P2↵	Print command to image buffer; Print 2 copies

[1]/. The Intermec logotype is not included in the software package and is only included in the example to demonstrate how to print a graphics field. You can substitute it with any graphics of approximately the same size. If you find it difficult to download graphics, you could omit the **GG** command from the example until you have become more familiar with the concept.

5 Editing in Form Edit Mode

This chapter gives an example of how to create a simple label in the Form Edit Mode.

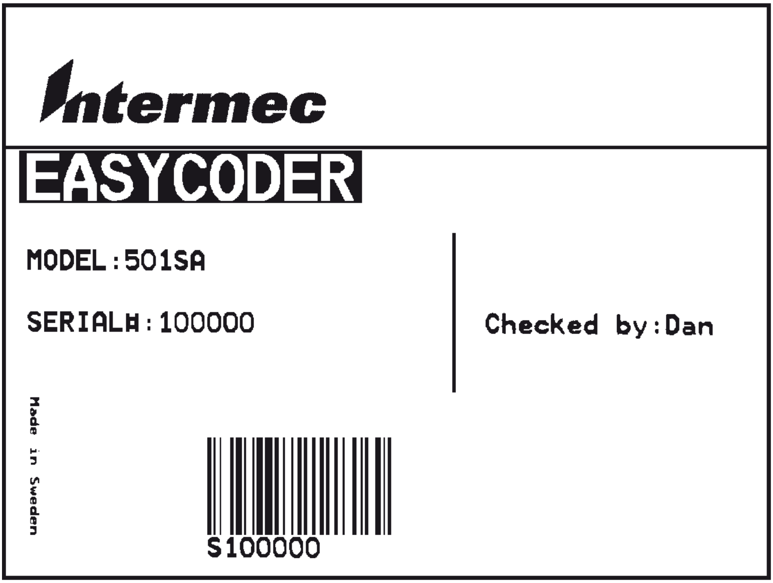
Example

Assuming that...

- the printer has been set up for the application (see Chapter 3),
- the length of the label and the gap has been determined using the Autosensing Mode (see *EasyCoder PD4, User's Guide*),
- and the graphic used in the example has been downloaded using a **GM** command[1]),

...we will now edit a label that can be saved as a form in the printer's memory and retrieved when so required. It also means that we can use counters and variables.

When we are finished, the label will look like this on an EasyCoder PD4 with a 200 dpi printhead:



EASYCODER

MODEL:501SA

SERIAL#:100000

Checked by:Dan

Made in Sweden

S100000

[1]/. The Intermec logotype is not included in the software package and is only included in the example to demonstrate how to print a graphics field. You can substitute it with any graphics of approximately the same size. If you find it difficult to download graphics, you could omit the **GG** command from the example until you have become more familiar with the concept.

Name the Form

Name of this form is TEST.

Command	Explanation
↵	CR/LF to start command structure
FK"TEST" ↵	Delete any existing form named TEST
FS"TEST" ↵	Start store form named TEST

Define Variables

The first variable (V00) has a maximum size of 15 characters.
The second variable (V01) has 10 characters and prints in reverse.
The third variable (V02) has a maximum size of 8 characters.

Command	Explanation
V00,15,N,"Enter Product name:" ↵	Define first variable
V01,10,L,"Enter Model number:" ↵	Define second variable
V02,8,N,"Checked by:" ↵	Define third variable

The text within quotes are prompts, which will be sent from the printer to the host when the label form is retrieved (serial communication only).

Define a Counter

The counter has maximum 6 digits.

Command	Explanation
C0,6,L,+1,"Enter Serial Number:"↵	Define counter

Note: The variables (V00, V01, V02) and counter (C0), are defined within this label form named TEST. The next label form containing variables and counters, will again start with V00 and C0. If variable data is being sent from an external data base, omit the text between the quotes and replace with a space character, for example **`V00,15,N," "`**.

Draw a Box and two Lines

Start to draw the surrounding box using the X command and then draw the two lines using the LO command.

Command	Explanation
X0,0,4,752,584↵	Draw a box
LO0,144,752,4↵	Draw a black line
LO440,232,4,160↵	Draw a black line

Place a Text Line with Fixed Data

Enter a 90° rotated text line containing the fixed data "Made in Sweden" in text size 1. The quotation marks enclosing the fixed data will not be printed. The text size (1) is the smallest resident font in the printer.

Command	Explanation
A40,400,1,1,1,1,N,"Made in Sweden"↵	90 degree. text line, fixed data

Place a Variable Text

The next line is a text line, using text size 5 in reverse and prints the variable V00. The data printed in this field must be sent to the printer at the time of form retrieval.

Command	Explanation
A24,160,0,5,1,1,R,V00 ↵	Write a text line, 1:st variable

Place a Combination of Fixed Data and a Variable

The following two command lines consist of a combination of fixed data enclosed in quotation marks and variable data.

Command	Explanation
A24,250,0,4,1,1,N,"MODEL: "V01↵	Text line, fixed data + 2:nd variable
A472,312,0,4,1,1,N,"Checked by: "V02↵	Text line, fixed data + 3:rd variable

Place a Combination of Fixed Data and a Counter

The next command line is a text line containing fixed data and the counter (C0). The first time this label form is retrieved for printing, the start value for this counter must be sent to the printer. The printer will store the value of the counter for this form and automatically continue to print the next value the next time this form is retrieved. Reset or set to another value by sending a new start value.

Note: The value of the counter will be kept in the memory even if another form is retrieved or the printer is switched off.

Command	Explanation
A24,312,0,4,1,1,N,"SERIAL#: "C0↵	Text line, fixed data + 1:st counter

Place a Bar Code with Fixed Data and a Counter

Below a Bar Code Command line is used to enter a Code 128 bar code, containing the fixed data "S" in combination with the actual counter value. It is also set for printing the human readable text below the bar code.

Note: The narrow to wide ratio is not relevant for Code 128. The printer will use the value for the narrow bar to define the bar code. (Value 3 for wide bar definition is ignored.)

Command	Explanation
B280,440,0,1,2,3,96,B,"S"C0↵	Bar code, fixed data + 1:st counter

Place Graphics

The next line writes a graphic named "Intermec" from memory and positions it on the form.

Command	Explanation
GG24,12,"LOGO"↵	Write graphic from graphics memory

End Programming of this Form

The closing command that flags the end of form, see the full program listing later in this chapter.

Command	Explanation
FE↵	Closing command to store form

Complete List of the Example

Command	Explanation
↵	CR/LF to start command structure
FK"TEST"↵	Delete current form named TEST
FS"TEST"↵	Start store form named TEST
V00,15,N,"Enter Product name:"↵	Define 1:st variable
V01,10,L,"Enter Model number:"↵	Define 2:nd variable
V02,8,N,"Checked by:"↵	Define 3:rd variable
C0,6,L,+1,"Enter Serial Number:"↵	Define counter
X0,0,4,752,584↵	Draw a box
LO0,144,752,4↵	Draw a line
LO440,232,4,160↵	Draw a line
A40,400,1,1,1,1,N,"Made in Sweden"↵	Write a 90° text line of fixed data
A24,160,0,5,1,1,R,V00↵	Write 1:st variable text field
A24,250,0,4,1,1,N,"MODEL: "V01↵	Write text line, fixed data + 2:nd variable
A472,312,0,4,1,1,N,"Checked by: "V02↵	Write text, fixed data + 3:rd variable
A24,312,0,4,1,1,N,"SERIAL#: "C0↵	Write text line, fixed data + 1:st counter
B280,440,0,1,2,3,96,B,"S"C0↵	Write barcode, fixed data + 1:st counter
GG24,12,"LOGO"↵	Write graphic from graphics memory
FE↵	Closing command to store form

6 Retrieving and Printing a Form

This chapter gives a step-by-step example of how a form is retrieved. The variable data fields are manually provided with information after the operator has been prompted to do so on the screen of the host. Finally, the operator orders a label to be printed.

Example

The form “TEST”, edited in the previous chapter, can be retrieved and printed from any ASCII sending device using this sequence:

Command	Explanation
↵	CR/LF to start command structure
FR"TEST"↵	Retrieve form
?↵	Call for variables
EASYCODER↵	Substitute variable V00
501SA↵	Substitute variable V01
Dan↵	Substitute variable V02
100000↵	Counter start value C0
P1,2	Print 2 copies of a single label

In this example we have manually substituted variables for testing purposes.

Note: It is critical to the syntax to send exactly the same number of variable lines as defined for this label form.

Provided you use the serial interface for communication between printer and host[1], you can make the printer return prompts that appear on the screen, requesting the operator to enter input data, by sending a **UI** command after each power-up. The optional Keyboard Display Unit automatically sends the **UI** command at power-up.

Printer Sends...	Command	Explanation
	↵	CR/LF to start command structure
	UI	Enable prompts command (optional)
UI80,001		Printer returns code page status
	FR"TEST"↵	Retrieve form
	?↵	Call for variables
Enter Product name:	EASYCODER↵	Substitute variable V001
Enter Model number:	501SA↵	Substitute variable V01
Checked by:	Dan↵	Substitute variable V02
Enter SERIAL#:		
100001	100000↵	Reset, accept, or enter2 counter start value C0
Number of labels sets		Prompt
P1		Ignore
	P1↵	Enter P + Quantity of labels
Copies of each label		Prompt
1	2↵	Enter number of copies + ↵

[1]/. The font selected in this example allows uppercase characters only.

The example below demonstrates that it is not necessary to set the counter start value again. The counter internally keeps track of the last number issued as long as the power remains switched on and is updated according to instructions in the form.

Command	Explanation
↵	CR/LF to start command structure
FR"TEST"↵	Retrieve form
?↵	Call for variables
EASYCODER↵	Substitute variable V00
501SA↵	Substitute variable V01
Dan↵	Substitute variable V02
↵	CR/LF to use existing counter value
P1,2↵	Print 2 copies of 1 label

Once a form has been retrieved, it can be used over and over again until another form is retrieved. All variable input data and counter values are stored in the volatile memory, which means they will be lost if the printer is switched off or at a power failure. If prompts are enabled, existing data and counter values will be displayed on the screen after the related prompt. Any input data can be overwritten at will.

Command	Explanation
?↵	Call for variables in same form
↵	CR/LF to use existing data in V00
↵	CR/LF to use existing data in V01
Sam↵	Substitute data in variable V02
200000↵	Substitute counter start value
P1,1↵	Print 1 copy of 1 label

Note: The question mark (**?**) following the **FR** command is essential for the printing of certain fields edited in the Form Edit Mode, that is fields containing variables or counters. Variables and counter start values must be entered or accepted as described above. If no question mark is transmitted, all fields containing variable input, that is variables and counters, will be completely omitted from the printout.

7 Commands

This chapter lists the various commands in alphabetic order. For each command, a short description is given, followed by the syntax for the command and an explanation the of parameters included in the syntax.

Examples of how to use the commands are also given.

Syntax

In the command syntax, there are a few conventions for substituting data or indicating how data can be used:

- **`p1 - pn`**
  Indicates parameters listed separately below the command syntax.
- **`[ ..... ]`**
  Square brackets indicate optional parameters or data.
- |
  A straight vertical lines indicates alternatives.
- **`"Name"`**
  Enter the name of the form or graphic within double quotation marks (ASCII 34 dec.), for example "Intermec".
- **`"DATA"`**
  The data could be from another source such as a .PCX file, a database, or entered by the operator. "Data" designates the place in the command sequence to input the data.

  Because the firmware uses " " (ASCII 34 dec.), you need a special designator if you need to print text or bar codes which include these quotation marks[1]. The backslash character "\" (ASCII 92 dec.) serves that purpose:

To print:	"	enter:	" \" "
To print:	"ABC"	enter:	" \"ABC\" "
To print:	\	enter:	" \\ "
To print:	\code\	enter:	" \\code\\ "

  [1]/. If a 7 bit character set is selected, this syntax will not be supported. All backslash (\) characters will be printed as entered.
- →
  Trailing arrow indicate that data continues on the next line.
  Leading arrow indicate that data is continued from the previous line.
- ↵
  Indicates a linefeed character (LF; ASCII 10 dec.), also see Commands in Chapter 1.
- **`<name>`**
  Alternative method for writing unprintable ASCII characters, for example **`<ACK>`** = ASCII 06 dec.

A – Print Text

Description This command is used to print an ASCII text string.

Syntax $\mathtt{Ap_1,p_2,p_3,p_4,p_5,p_6,p_7,"DATA"}$

Parameters

p_1 Horizontal start position (X) in dots.

p_2 Vertical start position (Y) in dots.

p_3
0 No Rotation. Left to right.
1 90 degrees rotation. Left to right
2 180 degrees rotation. Left to right
3 270 degrees rotation. Left to right
4 No Rotation. Top to bottom. Asian fonts (p4=8) only
5 90 degrees rotation. Top to bottom. Asian fonts (p4=8) only
6 180 degrees rotation. Top to bottom. Asian fonts (p4=8) only
7 270 degrees rotation. Top to bottom. Asian fonts (p4=8) only

p_4 Font Selection:

	203.2 dpi (8 dots/mm)		**300 dpi (11.81 dots/mm)**
1	20.3 cpi, 6 points (8 x 12 dots)		25 cpi, 4 points (12 x 20 dots)
2	16.9 cpi, 7 points (10 x 16 dots)		8.75 cpi, 6 points (16 x 28 dots)
3	14.5 cpi, 10 points (12 x 20 dots)		15 cpi, 8 points (20 x 36 dots)
4	12.7 cpi, 12 points (14 x 24 dots)		12.5 cpi, 10 points (24 x 44 dots)
5	5.6 cpi, 24 points (32 x 48 dots)		6.25 cpi, 21 points (48 x 80 dots)
8	Asian fonts :		
	- Korean	Korean.24	(24 x 24 dots in double-byte)
	- Chinese GB	GB.24	(24 x 24 dots in double-byte)
	- Chinese BIG-5	Big5.24	(24 x 24 dots in double-byte)
	- Japanese	Japanese.24	(24 x 24 dots in double-byte)
	- Latin		(12 x 24 dots in single-byte, included in all .24 fonts)
a–z	Soft Fonts		(dot size controlled by ES command)

p_5 Horizontal multiplier 1, 2, 3, 4, 6, 8.

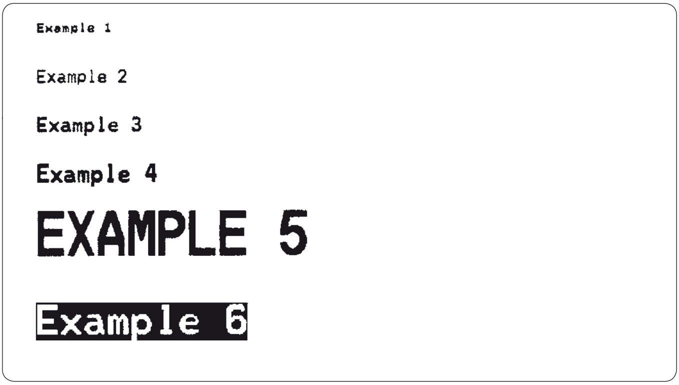
p_6 Vertical multiplier 1, 2, 3, 4, 5, 6, 7, 8, 9.

p_7
N Normal image
R Reverse image

"DATA" Represents a fixed data field.
When using Asian double-byte fonts, specify both bytes as ASCII decimal values, starting with the first value being larger than ASCII 127 dec (ASCII 7F hex), or below ASCII 127 (ASCII 7F hex) decimal to specify a Latin character.

Example

```
↵
N↵
A50,0,0,1,1,1,N,"Example 1"↵
A50,50,0,2,1,1,N,"Example 2"↵
A50,100,0,3,1,1,N,"Example 3"↵
A50,150,0,4,1,1,N,"Example 4"↵
A50,200,0,5,1,1,N,"EXAMPLE 5"↵
A50,300,0,3,2,2,R,"Example 6"↵
P1↵
```



Example 1

Example 2

Example 3

Example 4

EXAMPLE 5

Example 6

Note: Font size 5 only supports uppercase characters, see example 5 in the label above.

Remarks

The "DATA" field can be replaced by or combined with the commands below:

Variable:
Vnn Prints the contents of variable "nn" at this position, where nn is a 2 digit number from 00-99.

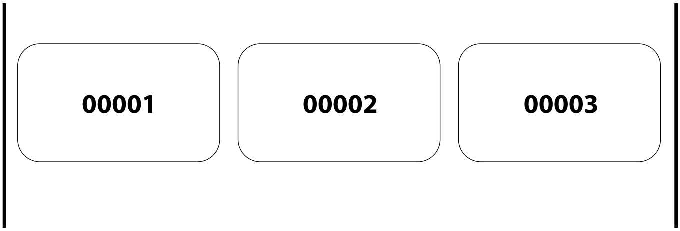
Consecutive Number Counter:
Cn Prints the contents of counter "n" at this position, where n is a 1 digit number from 0- 9.
Cn±x Prints the contents of counter "n" at this position while setting the counter's start value to "x". n and x are 1 digit numbers from 0-9
Enter + to increment or - to decrement.

Example:
When labels with consecutive numbers are printed next to each other across the media, it is done by using a single counter in a single form.

The command Cn±x in our example will be used twice and count up the single counter by one (1) in each position (last two A-command lines).

Set the Form Step Value p4 to +3 for the counter Cn used in our example (see the C-command line). Also refer to “**C – Counter**”.

```
↵
FK"TEST2"↵
FS"TEST2"↵
C0,5,L,+3,"Counter 0"↵
A180,50,0,3,1,1,N,C0↵
A380,50,0,3,1,1,N,C0+1↵
A580,50,0,3,1,1,N,C0+2↵
FE↵
```



Time:

TT Prints the current time at this position in the pre-defined format. See the **TT** command for format selection. This command is only available if a Real Time Clock is installed.

TT+nnn Prints “sell by” time. Adds nnn number of minutes (must be three digits) to the current time and places it on the form using time layout defined.

TD Prints the current date at this position in the pre-defined format. See the **TD** command for format selection. This command is only available if a Real Time Clock is installed.

TD+nn Prints “sell by” date. Adds nn number of days (must be two digits) to the current date and places it on the form using date layout defined.

This example illustrates how fixed text, variable text, counters, time and date can be used in text fields in the Form Edit Mode:

```
↵
FK"TEST1"↵
FS"TEST1"↵
V00,25,l,"Product name"↵
C0,4,L,+1,"Start serial No"↵
A50,50,0,4,1,1,N,"COMPANY NAME"↵
A50,100,0,3,1,1,N,"Product: "V00↵
A50,150,0,3,1,1,N,"Serial No: "C0↵
A50,200,0,3,1,1,N,"Expiry date: "TD+05↵
A50,250,0,3,1,1,N,"Packed :"TD"_"TT↵
FE↵
```

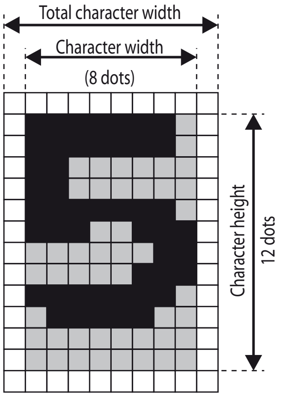
Combination of several options can also be used in a single text field:

```
A50,300,0,3,2,2,R,"Deluxe"V01C1"Combo"→
→TDV01TT↵
```

:Writes the text “Deluxe” + the contents of variable 01 + the contents of counter 2 + the text “Combo” + the current date + the contents of variable 01 + by the current time

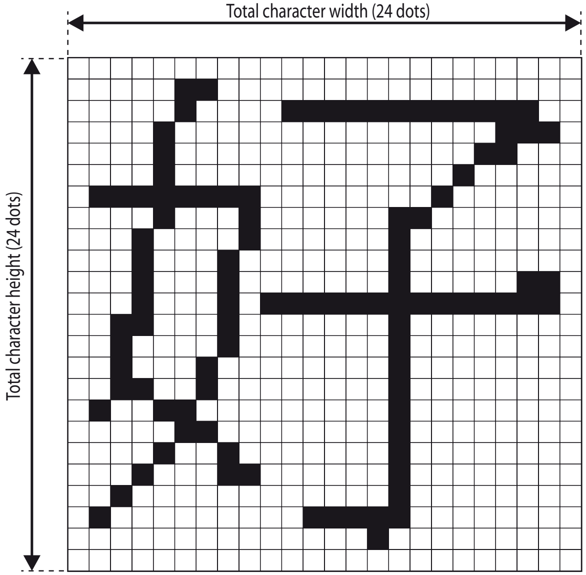
The Latin font characters (1–5) are dot-mapped differently than the Asian font characters (8). Asian characters do not have any built-in gap between individual characters and will thus be placed adjacently, whereas Latin characters include a single dot border around each character. To create inter-character spacing for Asian fonts, use the i command.

Latin Characters

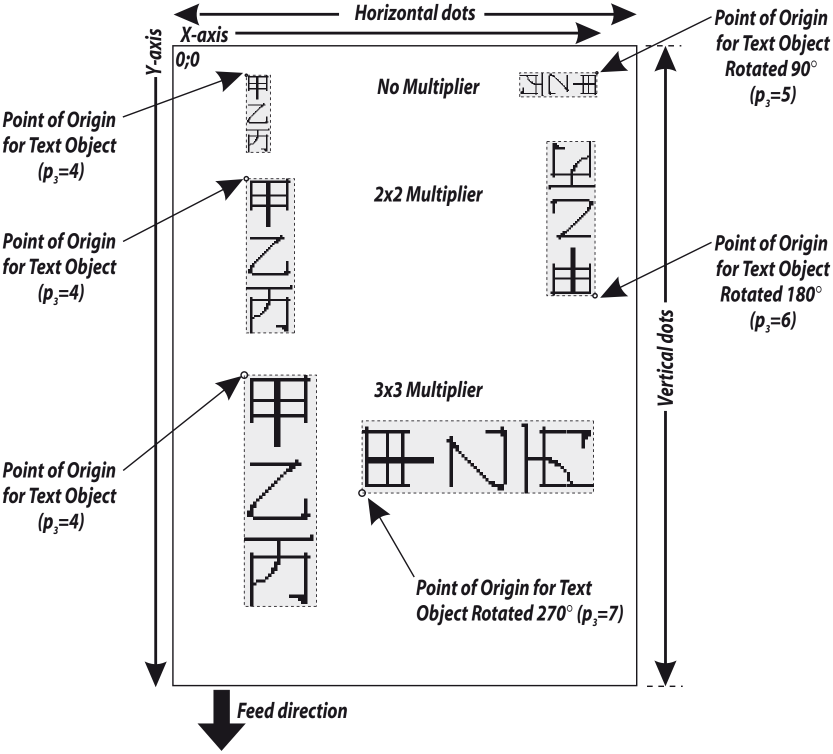


□ White dots (character spacing)

Asian Characters


Total character width (24 dots)
Total character height (24 dots)

The Asian fonts can print character strings running from top to bottom (p3 = 4–7), as well as the standard Latin word orientation from left to right (p3 = 0–3). The characters will print in the sequence that they are entered into the data field of the **A** command.


Horizontal dots
X-axis
Y-axis
0;0
No Multiplier
2x2 Multiplier
3x3 Multiplier
Point of Origin for Text Object (p_3=4)
Point of Origin for Text Object (p_3=4)
Point of Origin for Text Object (p_3=4)
Point of Origin for Text Object Rotated 90° (p_3=5)
Point of Origin for Text Object Rotated 180° (p_3=6)
Point of Origin for Text Object Rotated 270° (p_3=7)
Vertical dots
Feed direction

B – Standard Bar Codes

Description This command is used to print standard bar codes.

Syntax `Bp1,p2,p3,p4,p5,p6,p7,p8,"DATA"`

Parameters

p_1 Horizontal start position (X) in dots.

p_2 Vertical start position (Y) in dots.

p_3
- 0 No rotation.
- 1 90 degrees rotation clockwise.
- 2 180 degrees rotation clockwise.
- 3 270 degrees rotation clockwise.

p_4 Bar code select. See Bar Code Type table below.

p_5 Narrow bar width in dots. See Bar Code Type table below.

Bar Code Type	**"p_4"**	**"p_5"**
Code 39 std. or extended	3	1-10
Code 39 with check digit	3C	1-10
Code 93	9	1-10
Code 128 UCC case code	0	1-10
Code 128 A, B, C	1	1-10
Codabar	K	1-10
EAN8	E80	2-4
EAN8 2 digit add-on	E82	2-4
EAN8 5 digit add-on	E85	2-4
EAN13	E30	2-4
EAN 13 2 digit add-on	32	2-4
EAN13 5 digit add-on	E35	2-4
Interleaved 2 of 5	2	1-10
Interleaved 2 of 5 with check digit	2C	1-10
Interleaved 2 of 5 w human readable check digit	2D	1-10
Postnet 5, 6, 8 & 11 digit	P	n.a.
UCC/EAN 128	1E	1-10
UPC A	UA0	2-4
UPC A 2 digit add-on	UA2	2-4
UPC A 5 digit add-on	UA5	2-4
UPC E	UE0	2-4
UPC E 2 digit add-on	UE2	2-4
UPC E 5 digit add-on	UE5	2-4
UPC SCC	2U	1-10

p_6 Wide bar width in dots (2 -30).

p_7 Barcode height in dots.

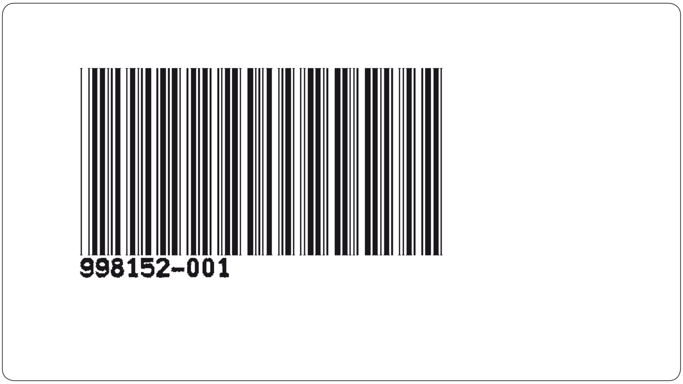
p_8
- B Human readables ON.
- N Human readables OFF.

"DATA" Represents a fixed data field.

Example

This example produces a Code 39 bar code:

```
↵
N↵
B50,50,0,3,2,6,200,B,"998152-001"↵
P1↵
```



Remarks

The "DATA" field can be replaced by or combined with the commands below:

Variable:

Vnn Prints the contents of variable “nn” at this position, where nn is a 2 digit number from 00-99.

Consecutive Number Counter:

Cn Prints the contents of counter “n” at this position, where n is a 1 digit number from 0-9.

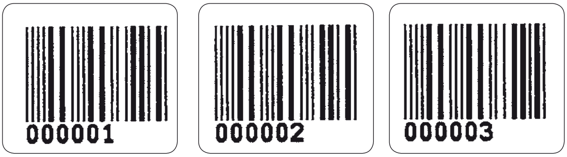
Cn±x Prints the contents of counter “n” at this position while setting the counter’s start value to “x”. n and x are 1 digit numbers from 0-9.
Enter + to increment or - to decrement.

When labels with consecutive numbers are printed next to each other across the web, it is done by using a single counter in a single form.

The command Cn±x in the following example will be used twice and count up the single counter by one (1) in each position (last two B-command lines).

Set the Form Step Value p4 to +3 for the counter Cn used in our example (see the C-command line). Also refer to “C command – Counter”.

```
↵
FK"TEST3"↵
FS"TEST3"↵
C0,6,L,+3,"Counter 0"↵
B120,50,0,2,3,6,100,B,C0↵
B320,50,0,2,3,6,100,B,C0+1↵
B520,50,0,2,3,6,100,B,C0+2↵
FE↵
```



Time:

TT Prints the current time at this position in the predefined format. See the **TT** command for format selection. This command is only available if a Real Time Clock is installed.

TT+nnn Prints "sell by" time. Adds nnn number of minutes (must be three digits) to the current time and places it on the form using time layout defined.

TD Prints the current date at this position in the predefined format. See the **TD** command for format selection. This command is only available if a Real Time Clock is installed.

TD+nn Prints "sell by" date. Adds nn number of days (must be two digits) to the current date and places it on the form using date layout defined.

This example illustrates how fixed text, variable text, counters, time and date can be used in text fields in the Form Edit Mode:

```
↵
FK"TEST4"↵
FS"TEST4"↵
V00,25,l,"Product name"↵
C0,4,L,+1,"Start serial No"↵
B50,50,0,3,2,6,100,B,"TEXT"↵
B50,200,0,3,2,6,100,B,V00↵
B50,350,0,3,2,6,100,B,C0↵
B50,500,0,3,2,6,100,B,TT↵
B50,650,0,3,2,6,100,B,TD↵
FE↵
```

Combination of several options can also be used, for example:

```
B50,300,0,3,1,2,50,B,"Deluxe"V01C2"Combo"TDV
01TT↵
```

:Writes a Code 39 bar code containing the information “Deluxe” + the contents of variable 01 + the contents of counter 2 + the “Combo” + the current date + the contents of variable 01 + by the current time.

Bar Code 128 family:

The Code 128 family has three unique data character subsets: A, B and C. The choice of data character subset depends on the start character representing code A, B or C. The code subset can be redefined within the symbol by code set control characters A, B or C, or shift. In addition, a set of special functions are available via FNC1, FNC2, FNC3 and FNC4. All these can be invoked in the data field. The human readable part in the symbol, FNC and code subset control characters will be represented by a space. The input data string is a fixed string of maximum 64 alpha-numerical characters.

Code 128 with auto selection of subset and UCC/EAN:

Code 128 and EAN128 use automatic subset selection of subset A B or C, both at the start and within the barcode.

Code 128 with manual selection of subset:

By starting the Code 128 symbol with manual selection of subset A, B or C, the user has complete control of composing the symbol data output. By invoking special control characters in the data string, it is possible to switch between subsets or other special cases inside the symbol, see table below.

String invocation rules

Command Type ▶	Text Print	Bar Code C128 Manual Selection			Bar Code C128 Auto UCC/EAN128[4]
Command ▶	A	B	B	B	B, b[5]
Param. p_4 ▶	-	1A	1B	1C	1, 1E[4], 1EC[5]
Starts with ▶	-	Start A	Start B	Start C	Start A, B or C (+FNC1[4])
Invocation in String ▼	Text Output▼	Bar Code Output▼	Bar Code Output▼	Bar Code Output▼	Bar Code Output▼
\"	"	"	"	NA[2]	"
\\	\	\	\	NA[2]	\
\1	1	<FNC1>	<FNC1>	<FNC1>	<FNC1>
\2	2	<FNC2>	<FNC2>	NA[2]	<FNC2>
\3	3	<FNC3>	<FNC3>	NA[2]	<FNC3>
\4	4	<FNC4>	<FNC4>	NA[2]	<FNC4>
\A	A	Ignore[1]	<Code A>	<Code A>	A
\B	B	<Code B>	Ignore[1]	<Code B>	B
\C	C	<Code C>	<Code C>	Ignore[1]	C
\S	S	<Shift>	<Shift>	NA[2]	S
Char 06[3]	Char 06	Char 06	NA[2] in Code B	NA[2]	<FNC1>

b – Two-Dimensional Bar Codes, General Part

Description

This command is used to print two of three complex bar codes; Data Matrix, PDF417, MaxiCode, and QR Code. The command consists of two parts; a leading set of general positioning and bar type select parameters, and a trailing code-specific part defining the bar code's appearance and its input data.

The general part of the syntax is indicated by italic characters in the code-specific syntax descriptions on the following pages.

Syntax

bp$_1$,p$_2$,p$_3$,[code specific options]

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
p_3	Code type: D Selects Data Matrix M Selects MaxiCode P Selects PDF417 Q selects QR Code

[code specific options], see the following two pages.

Remarks

The standard program packages contains all the two-dimensional bar code mentioned above.

b – MaxiCode

Description The following MaxiCode specific options should append the general part of the two-dimensional code command (indicated by italic characters, see **b** command, general part). Only mode 2 and 3 are supported.

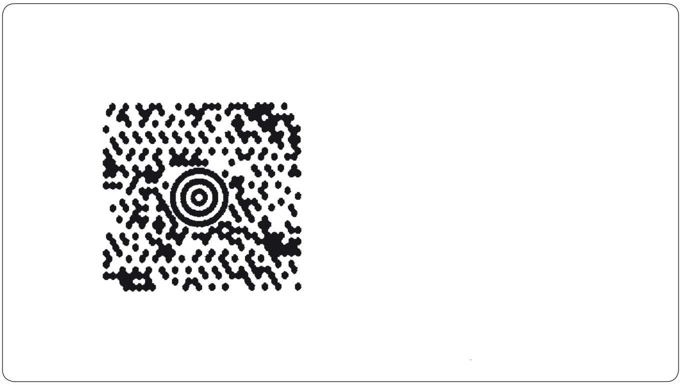
Syntax $bp_1,p_2,p_3,$ **["CL,CO,PC,LPM"]**

Parameters

CL	Class Code (3 digit number).
CO	Country Code (3 digit number).
PC	Postal Code (type of input decides mode 2 or 3): Mode 2: U.S.A. (5 digits, 4 digits). Note the separating comma sign! Mode 3: International (6 alphanumeric characters).
LPM	Low Priority Message (up to 84 alphanumeric characters).

Example

```
N↵
b100,100,M,"300,400,93065,1692,This is→
→MaxiCode"↵
P1↵
```



b – PDF417

Description

The following PDF417 bar code specific options should append the general part of the two-dimensional code command (indicated by italic characters, see **b** command, general part).

Syntax

$bp_1,p_2,p_3,$ **[www,hhh,s,c,p,f,d,x,y,r,l,t,o],"DATA"**

Parameters

www Maximum print width in dots (3 digits).

hhh Maximum print height in dots (3 digits).

s Sets error correction level. Legal values are 0 thru 8.
If level is not specified, a level that will generate about 1/8 as many ECC code words as data code words is selected.

c Selects data compression method:
- 0 Selects auto-encoding (default).
- 1 Selects binary mode.

p Print human readable code appended by additional variables:
- xxx horizontal start location (3 digits).
- yyy vertical start location (3 digits).
- mmm maximum characters per line (3 digits).

f Center pattern in area:
- 0 The pattern will print upper left justified in the area defined by the w and h values.
- 1 The pattern is printed in middle of the area defined by the w and h values (default).

d- Print code words:
- 0 Values of code words not printed (default).
- 1 Values of code words printed.

x- Module width. Legal values are 2-9.

y- Set bar height. Legal values are 4-99 dots high.

r- Maximum row count (refer to PDF 417 specifications).

l- Maximum column count (refer to PDF 417 specifications). Note that this character is lowercase L (ASCII 108 dec.).

t- Truncated flag:
- 0 Not truncated.
- 1 Truncated.

o- Rotation:
- 0 0° rotation clockwise.
- 1 90° rotation clockwise.
- 2 180° rotation clockwise.
- 3 270° rotation clockwise.

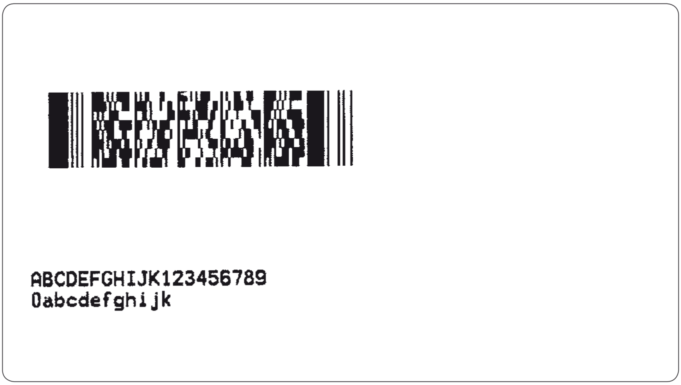
"DATA" Represents a fixed data field.

Remarks If parameter www (max. print width) gives less space than required by the sum of parameters x- (module width) and l- (max. column count), error condition 50 will occur.

Likewise, if parameter hhh (max. print height) gives less space than required by the sum of parameters y- (set bar height) and r- (max. row count), error condition 50 will also occur.

Example

```
↵
N↵
b40,40,P,400,300,p40,340,20,f1,x3,y10,r60,→
→l5,"ABCDEFGHIJK1234567890abcdefghijk"↵
P1↵
```



```
ABCDEFGHIJK123456789
0abcdefghijk
```

Note: The last parameter in the b command above (l5) is lowercase L + the digit 5, not 15!

b – Data Matrix

Description

The following Data Matrix specific options should append the general part of the two-dimensional code command (indicated by italic characters, see **b** command, general part).

Syntax

$bp_1,p_2,p_3,$ **[P$_4$,"DATA"]**

Parameters

P$_4$	Narrow bar from 1 to 15.
"DATA"	Represents a fixed data field, maximum 125 characters long.

Example

```
N↵
b40,80,D,5,"123456789012345678901234567890→
→1234567890123456789012345678901234567890→
→12345678901234567890"↵
P1↵
```

b – QR Code

Description

QR Code is a matrix symbology, defined in AIM International ITS/97-001.

Basic characteristics for QR Code:

QR Code can be encoded as two models;	Model 1	Model 2
Model Version	original	enhanced
Symbol size in cell, (increment of 4 cells)	21*21 to 73*73	21*21 to 177*177
1. Numeric data mode: Digits 0-9;	Max: 1167 char	Max: 7089 char
2. Alphanumeric data mode: Digits 0-9; upper case letters A-Z; nine other characters	Max: 707 char	Max: 4296 char
3. 8-bit byte data mode: JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201	Max: 486 char	Max: 2953 char
4. Kanji data mode: Shift JIS X 0208	Max: 299 char	Max: 1817 char
5. Mixing mode: Mix of mode 1-4 above	NA	Applicable

Note: The maximum data length in ESim is approximately 780 bytes.

Error correction:

Four levels of Reed-Solomon error correction allowing recovery of:

L	7%
M	15%
Q	25%
H	30%

Masking (intrinsic):

This enables the ratio of dark to light modules in the symbol to be approximated to 1:1 whilst minimizing the occurrence of arrangements of close modules which would prevent efficient decoding.

The following QR Code bar code specific options should append the general part of the two-dimensional code command (indicated by italic characters, see **b** command, general part).

Syntax *bp*$_1$*,p*$_2$*,p*$_3$*,* **[cn,mn,on,sn,xn,ynn],"DATA"**

Parameters

c Fixed data mode character set:

1 Numerical data mode

2 Alphanumerical data mode

3 8-bit data mode (JIS 0201)

4 Kanji data mode (JIS 0208)

5 Mixing mode (any of mode c1-c4, model type 2)

Default: Model 1 Most optimal alternative of c1 to c4

Model 2 c5

Internally during rendering the QR Code, the mode can be switch within a bar code to minimize the bit stream area. By using parameter cn, the data mode character set is fixed.

m Model type is defined and unchanged within one QR Code.

1 Original version

2 Enhanced version (recommended). Default.

o Rotation:

0 0° rotation clockwise. Default.

1 90° rotation clockwise.

2 180° rotation clockwise.

3 270° rotation clockwise.

s Sets correction level.

L 7 % Default.

M 15%

Q 25%

H 30%

x Masking factor 0-7 or 8 (auto).

Default: 8

y Cell pixel size magnification 02-64.

Default: 10

"DATA" Represents a fixed data field. Maximum data length is relative chosen model type and data mode but is limited to approximately 780 bytes. Ordinary string controls, like \\ for \ and \" for ".

Errors

If data not suitable to optional parameters:	Error 01; Syntax error
If data too long or out of range:	Error 03; Data length error
If data does not fit in area specified	Error 50

Example

```
N↵
b40,40,Q,sM,x1,"ABCDEFGHIJKLMNOPQRSTUVWHYZ→
 12345678901234567890 1234ABCDEFGHIJKLMNOPQR→
 STUVWHYZ1234"↵
P1↵
```

C – Counter

Description This command is used to define one of max. 10 automatic counters used in consecutive numbering applications, for example serial numbers. Counters can only be used in the Form Edit Mode, not in the Direct Mode.

Syntax **Cp_1,p_2,p_3,p_4,[p_5,]"PROMPT"**

Parameters

p_1 Counter number (0-9).

p_2 Maximum number of digits for the counter (1-29).

p_3 Field justification:

L Left justification.
R Right justification.
C Center justification.
N No justification.

p_4 Step value. Plus or minus sign followed by a single digit (1-9):

+ Incrementation.

- Decrementation.

p_5 Alphanumerical support (optional):

N Numerical support. Anything else than digit and leading spaces in DATA will give error 03.

A Alpha support: Base 10 on digits, base 26 on letters. Default
Anything else than digit, leading spaces, and uppercase alpha characters in DATA will give error 03.
Wrap around: 0 ⟶ 9 goes back to 0
A ⟶ Z goes back to A

B Alphanumerical support: Base 36
Anything else than digit, leading spaces, and uppercase alpha characters in DATA will give error 03.
Wrap around: 0 ⟶ 9 A ⟶ Z 9 goes to A
Z goes to 0

[-] A single leading minus sign in the prompt field will cause the prompt to be sent one time only after the form is retrieved (Keyboard Display Unit only, see below).

[--] A double leading minus sign in the prompt field will cause the prompt to be suppressed (Keyboard Display Unit only, see below).

"PROMPT" An ASCII text field that will be transmitted to the Keyboard Display Unit or host via the serial interface each time a form containing this command is retrieved. It usually requests the operator to enter the starting value for the counter.

Remarks

This command is used in forms that require sequential numbering. When initializing counters, they must be defined in order (for example C0, C1, C2, etc.) after possible variables.

To print the contents of the counter, the counter number (C0-C9) is entered in the "DATA" field of **A** (Print Text) or **B** (Print Bar Code) commands.

Prompts will only be displayed if a **UI** command has been issued after last power-up. The Keyboard Display Unit sends the **UI** command automatically.

The field justification parameter (p_3) affects the way the counter will be printed. When p_3 = L, R, or C, the counter value will be printed left, right or center justified in an area with a width defined by p_2 (number of digits). If no justification is selected (p_3 = N), the field will truncated from the right side so as to not exceed the set maximum field length, which may be useful when using a counter as input data to a bar code.

If the start value entered, when the form is retrieved for printing, is started by one or several zeros (0), the entire area specified by p_2 (number of digits) will be padded with leading zeros, that is p_3 (field justification) will have no effect.

Note: If a single counter is stepped up several times on the same form, then the step value p_4 must be set to the number of times the counter is used in the form or equivalent to what the step values for the single counter add up to in this form. A Cn±x command must also be used when designing the actual form. See the **A** and **B** commands.

The table below illustrates how leading space characters ("") in DATA will be influenced by next character at wrap around. If next position is a digit with wrap around (9→0), space will be incremented to digit 1. If next position is a letter with wrap around (Z→A), space will be incrementred to letter A.

P_5	Counter Command	Start Data	Data after p_4= +1		Data after p_4= +2	
N	**C**0,3,C,+1,N,"Numerical"	`" 99"`	→	`"100"`	→	`"101"`
		`"999"`	→	`"000"`	→	`"001"`
A	**C**0,3,C,+1,A,"Alpha" (if no P_5 parameter is included, method A is selected by default).	`" A9"`	→	`" B0"`	→	`" B1"`
		`" Z9"`	→	`"AA0"`	→	`"AA1"`
		`"0Z9"`	→	`"1A0"`	→	`"1A1"`
		`" ZZ"`	→	`"AAA"`	→	`"AAB"`
		`"ZZ9"`	→	`"AA0"`	→	`"AA1"`
B	**C**0,3,C,+1,B,"Alpha-num"	`" 99"`	→	`" 9A"`	→	`" 9B"`
		`" A9"`	→	`" AA"`	→	`" AB"`
		`" 9Z"`	→	`" A0"`	→	`" A1"`
		`" ZZ"`	→	`"100"`	→	`"101"`
		`"ZZZ"`	→	`"000"`	→	`"001"`

Example This form lets you test field justifications by entering various start values when the form is retrieved for printing.

```
↵
FK"TEST5"↵
FS"TEST5"↵
C0,5,L,+1,"Start value CNT 0"↵
C1,5,R,+1,N,"Start value N-CNT 1"↵
C2,5,C,+1,A,"Start value A-CNT 2"↵
C3,5,N,+1,B,"Start value B-CNT 3"↵
A50,050,0,3,1,1,N,"Cnt Default,   left →
→justified  :"C0":"↵
A50,100,0,3,1,1,N,"Cnt Numerical, right →
→justified :"C1":"↵
A50,150,0,3,1,1,N,"Cnt Alpha,     center →
→justified:"C2":"↵
A50,200,0,3,1,1,N,"Cnt Alpha-num, not →
→justified   :"C3":"↵
FE↵

FR"TEST5"↵
?↵
A9↵
 99↵
 ZZ9↵
 ZZ9↵
P3↵
```

Protect Counters

When the optional Keyboard Display Unit (KDU) is used, the label form can be designed to “skip” a consecutive number prompt, thereby protecting the data. This feature is especially useful when the counter represents a serial number or other types of number, that should never be repeated.

By placing one (1) minus sign as the first character of the prompt, the prompt will appear only once after the form is retrieved.

Example:

```
C0,10,L+1,"-Enter Serial Number:"↵
```

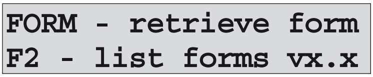
By placing two (2) minus signs as the first two characters of the prompt, the prompt will never be displayed.

Example:

```
C0,10,L+1,"- -Enter Serial Number:"↵
```

The protected consecutive number is accessed and modified from the optional Keyboard Display Unit only.

Enter the following when the KDU is displaying:



```
FORM - retrieve form
F2 - list forms vx.x
```

1 If necessary, press <Exit> key to display above.

2 Press <F1> key.

3 Press 4 9 1 6.

4 Press the <Form> key.

5 Key in Form name and press <Enter> to retrieve.

6 Enter or modify the consecutive number.

7 When complete, print label to store new number in memory.

C – Cut Immediate

Description This command is used to make the cutter perform a cutting cycle without printing a form.

Syntax

```
C
```

Remarks The command C is used for two purposes. C appended by parameters is used to define counters in the Form Edit mode, whereas C without any appending parameters is used to initiate a cutting operation.

C (Cut Immediate) can not be used inside a form or in connection with a keyboard/display unit (KDU).

To perform self-cleaning of the cutter blade, issue five consecutive C commands without any media loaded.

Example

```
C↵                                    :Performs a cutting cycle
```

D – Density

Description This command is used to select the print density.

Syntax `Dp1`

Parameters

p_1	Density setting (0-15). Default: 10. 0 is the lightest printing and 15 is the darkest.

Remarks The density command is used to control the energy to the printhead. A number of factors affect the actual darkness of the printout:

- Direct thermal printing or thermal transfer printing
- Print speed
- Different brands of direct thermal media
- Different combination between transfer ribbons and receiving face materials
- Different ambient temperature/humitity

The printed information may also require the density to be adjusted. Typically, this applies to different bar code orientations and densities.

Test after the print speed has been set (see **S** command) and make further adjustments until you have found the settings which best apply to your unique application.

Example

```
D9↵                                    :Selects density 9
```

EI – List Soft Fonts

Description	This command makes the printer print a list of all soft fonts that are stored in memory.
Syntax	`EI`
Remarks	This command is related to **ES** (Store Soft Fonts) and **EK** (Delete Soft Fonts).
Example	`EI↵` :Print a soft font list

EK – Delete Soft Font

Description This command is used to delete soft fonts from memory.

Syntax `EK["name"|"*"]`

Remarks Soft fonts are stored using the **ES** command and listed using the **EI** command.

Soft fonts can also be deleted from the printer using for example Intermec LabelShop or Intermec InterDriver.

Example

```
EK"a"↵                                  :Deletes font "a"
EK"*"↵                                  :Deletes all soft fonts
```

ES – Store Soft Font

Description This command is used to download and store soft fonts in memory.

Syntax ES"name"$p_1p_2p_3a_1b_1c_1$"data$_{1\ldots}$ $a_nb_nc_n$"data$_n$"

Parameters

"name$_{1\text{-}n}$" Name of the soft font (one lowercase letter only in the range a–z). Lower case named fonts minimize soft font memory usage to only store fonts downloaded and have 256 character limit.

p_1 Number of characters to be downloaded using hexadecimal coding. Range 00–FF hex (1–256 characters per soft font set).

p_2 Character rotation using hexadecimal coding:

00 hex:	0 and 180 degrees
01 hex:	90 and 270 degrees clockwise
02 hex:	All for directions (2 pairs)

p_3 Font height measured in dots and specified using hexadecimal coding. Range 00–FF hex. Font height includes accentors and dissenters of character and need to fit in the character cell of 256 dots = 32.03 mm (1.26 inches).

a Map position of character using hexadecimal coding. Range 00–FF hex.

b Spacing to next print character in dots using hexadecimal coding. Range 00–FF hex. Must be greater than or equal to the character width specified by parameter c.

c Width of character in dots using hexadecimal coding. Range 00–FF hex.

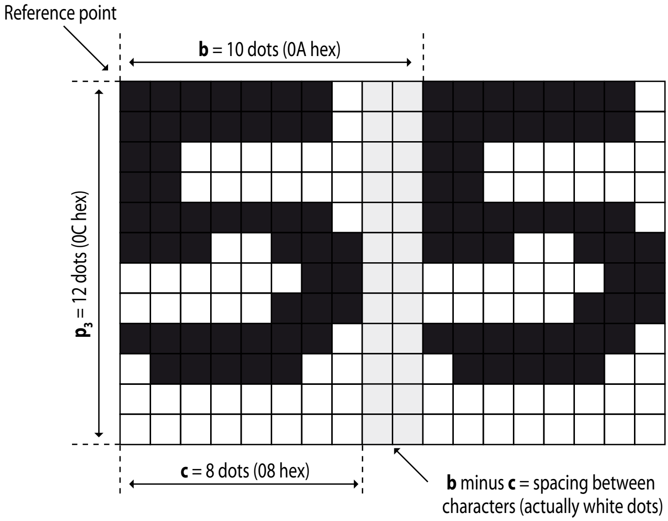
"data" p3 x c1 = bit map data (in bytes). Data is received in bytes on a line by line basis. The font character's 0,0 cell map position is in the top left corner of the map as viewed in the 0 degree rotation.

Repeat parameters a, b, c, and data for each character until all characters in the set have been downloaded.

For fonts with the rotation parameter p_2 set to 02 hex (all directions), repeat the individual font character download for each 90° rotated character from the start of the character set until all rotated characters in the set have been down loaded. The number of individual character maps downloaded will be double the characters in the font set (p_1).

Remarks

This picture illustrates the parameters p_3, b, and c:

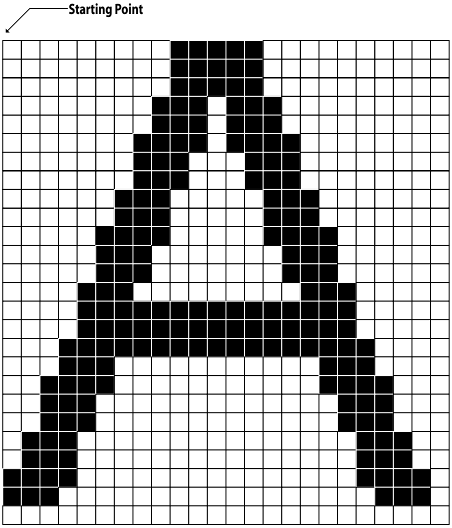

Reference point
b = 10 dots (0A hex)
$\mathbf{p_3}$ = 12 dots (0C hex)
c = 8 dots (08 hex)
b minus **c** = spacing between characters (actually white dots)

The black and white bitmap that represents the character must be converted to hexadecimal code. The bitmap is described line by line from left to right, starting from the upper left corner of the character cell. A white dot is represented by 0 and a black dot by 1. Each byte (that is 8 dots) will thus form a binary number, that is converted to hexadecimal code. The last byte in a line is padded with zeros to complete the line and data byte. The data is the sent to the printer as a continuous string of hexadecimal byte representations in line order.

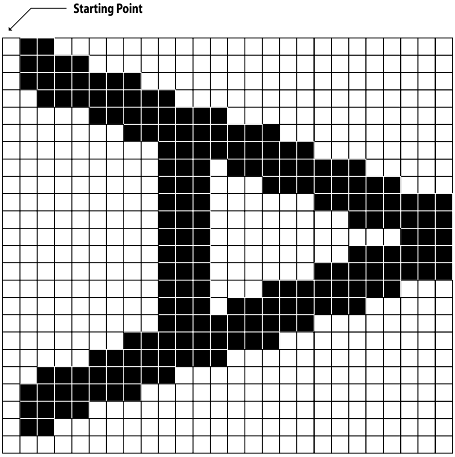
Soft fonts can also be downloaded to the printer using for example Intermec LabelShop or Intermec InterDriver.

Soft fonts can be listed using the **EI** command.

This example shows how a character in 0° and 90° rotation is downloaded to the printer:



```
Line 1:   00000000 01111100 00000000 = 00 7C 00
Line 2:   00000000 01111100 00000000 = 00 7C 00
Line 3:   00000000 01111100 00000000 = 00 7C 00
Line 4:   00000000 11111110 00000000 = 00 7E 00
.....
.....
```



```
Line 1:   01 10000000 00000000 00000000 = 01 80 00 00
Line 2:   01 11100000 00000000 00000000 = 01 E0 00 00
Line 3:   01 11111100 00000000 00000000 = 01 FC 00 00
Line 4:   00 11111111 00000000 00000000 = 01 FF 00 00
.....
.....
```

eR – User-Defined Error/Status Character Control

Description

This command controls the format of error messages, mostly when command **US** has enabled the printer's status reporting feature, but also at some high priority error types when command **UN** has disabled status report.

Syntax

```
eRp1,p2[,p3]
```

Parameters

p_1 Reply error action single character when error occurs:
Value of p_1 is one single ASCII-character, except hex 00. Default: 0

p_2 Mode of User Defined Error/Status Character reply (one ASCII-numerical):

$p_2 = 0$ Mode of Reply Default, where error number is NN:
Output at error action: <NAK>NN<XOFF>
Output at error recovery: <XON>

$p_2 = 1$ Mode of Reply only with character, defined by p_1:
Output at error action: <p_1><CRLF>
Output at error recovery: <p_3><CRLF>

$p_2 = 2$ Mode of Reply with character p_1 and error number NN:
Output at error action: <p_1>NN<CRLF>
Output at error recovery: <p_3>00<CRLF>

$p_2 = 6$ Mode of <XOFF> and <XON> replacement:
Like default mode (0), but with all <XOFF> replaced by p_1 and all <XON> replaced by optional p_3:
Output at error action: <NAK>NN<p1>
Output at error recovery: <p_3>NN

p_3 Optional recovery action single character:
The normal recovery character <XON> will be replaced by character p_3.
Value of p_3 is one single ASCII-character, except hex 00.
If no p_3 is available, p_1 is used instead as recovery action character.

Remarks

Exception of output: Error 07 "Out of media or ribbon", is added after 07 with Pnnn or Rnnn, where nnn = numbers of remaining labels:
Output at error media action (mode 0):
`<NAK>07Pnnn<XOFF>`
Output at error ribbon action (mode 0):
`<NAK>07Rnnn<XOFF>`
Output at error recovery:
`<XON>`
Mode 1, 2 and 6 changes the output in a similar way.

All parameters are saved in Flash. If more than one error occur at the same time, there will be an output for each error recovery, like **`<p1>00<CRLF>`** on mode 2. Refer to Appendix A for more information on error-handling.

Example

Command	Manual action	Reply	Comments
US↵		<ACK>	Enable Error Reporting
eRC,2,Q↵		*(Nothing happens)*	Reply error with character ‘C’ and error NN and at recovery with character ‘Q’
	Lift the print-head	C11<CRLF>	Manual action generates error 11
	Close the print-head	Q00<CRLF>	Manual recovery on error 11

f – Cut Position

Description

This command is used to adjust the cutting position along the Y-axis in relation to the media according to characteristics of individual printers and batches of media.

Syntax

```
fp1
```

Parameters

p_1 Cut position index measured in dots:

203.2 dpi (8 dots/mm):

Recommended:	070–130.
Min/max:	020–180 (lower or higher values ignored).
Default:	100

300 dpi (11.81 dots/mm):

Recommended:	050–150.
Min/max:	000–200 (lower or higher values ignored).
Default:	100

Remarks

When using labels on liner, the printer will advance each printed label to the cutting position (between two labels) according to the program before the liner is cut. Due to differences between batches of media, the printer may not cut exactly between labels, but either cut a slice at the last part of the printed label or a slice at the front part of next label.

Cutting through labels should be avoided, because the adhesive will stick to the cutting parts and prevent them from operating properly.

If the cut occurs in the already printed label, increase the cut position index value.

If the cut occurs in the next label, decrease the cut index value.

Example

```
f110↵                    :Increases the cut position index value.
```

FE – End Store Command

Description This command is used to end a Form Store sequence.

Syntax

```
FE
```

Remarks The Form Store sequence is started with the **FS** command and ended with the **FE** command.

Example

```
FS"formname"↵                    :Starts Form Store
.....
FE↵                              :Ends Form Store
```

FI – Print Form Information

Description This command makes the printer produce a list of all forms stored in memory.

Syntax

```
FI
```

Remarks The **FI** command will be executed directly, without appending any Linefeed.

Hint:
Issue a **FI** command after having stored a form to make sure the storing was successful and to check the amount of user flash memory.

FK – Delete Form

Description This command is used to delete a specified form or all forms from memory.

Syntax

```
FK"name"│"*"
```

Parameters

"name"	By entering a name of a form, that form only will be deleted from memory.
"*"	By entering an asterisk (*) as wildcard, all forms will be deleted from memory.

Examples

```
FK"FORM1"↵                               :Deletes "FORM1"
FK"*"↵                                   :Deletes all forms
```

FR – Retrieve Form

Description This command is used to retrieve a form that was previously stored in memory.

Syntax `FR"name"`

Parameters

"name"	This is the form name used when the form was stored. The printer is case -sensitive, that is, the use of upper and lower case letters must match the original name.

Remarks To print a list of the forms currently stored in memory, use the **FI** command.

Example

```
FR"Test1"↵                    :Retrieves the form named "Test1"
```

FS – Form Store

Description This command is used to begin a Form Store sequence.

Syntax `FS"name"`

Parameters

"name"	This is the form name that will be used when retrieving the stored form. The name may be from 1 to 8 characters. The printer is case sensitive, that is form names will be stored with the exact case entered on the FS command line.

Remarks All commands following FS will be stored in the user flash memory until a FE command is received, ending the form store process. If a form with the same name is already stored in memory, the FS command will result in an error and the old form will be retained. When updating a form, use the **FK** command to delete the old version before storing the new version. To print a list of the forms currently stored in memory, use the **FI** command.

Global commands, such as EI, EK, ES, FI, FK, GI, GK, GM, I, N, P, U, UE, UF, UG, UI, UM, UP, UN, US, Y, W, ?, ^@, or ^ee should not be used in a form store sequence.

Startup Form A special case of forms is the startup form, that is automatically retrieved and prompted for variables (if necessary) each time power is applied to the printer. A startup form is created by naming the form "AUTOFR". To exit the "AUTOFR" mode, send XOFF or NULL to the printer on any channel.

Note: Always test the form using another name before making it a startup form. If a startup form causes an error, there are two ways of clearing it:

- If the Ready lamp shines green, send XOFF or NULL to exit "AUTOFR" mode. Then delete the startup file using FK "AUTOFR"
- If the Error lamp shines yellow, there is no communication and the memory must be erased by pressing the Cancel button for more than 3 seconds in the Dump Mode.

Example

`FS"TEST1"↵`	:Begins the form store sequence of "TEST1"
`.....`	
`FE↵`	:Ends the form store sequence of "TEST1"

GG – Print Graphics

Description

This command is used to print a graphic that has been previously stored in the user flash memory.

Syntax

```
GGp1,p2,"name"
```

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
"name"	This is the name used when the graphic was stored. The name may be from 1 to 8 characters. The printer is case sensitive, that is the use of upper and lower case letters must match the original name.

Remarks

A graphic can only be printed in same direction and size as when it was saved. There are no means of magnification or rotation of an individual graphic. However, the entire print image including all text, bar codes, graphics, lines, and boxes can be rotated 180° using the **Z** command.

Note: EasyCoder PD4 has a flash memory for forms and graphics, which requires special consideration. Avoid storing frequently changing data in flash (see **GM** and **GW** commands) and use printer drivers developed for EasyCoder C4 and PD4.

Example

```
GG50,50,"LOGO"↵
```

:Prints the graphic "LOGO"

GI – Print Graphics Information

Description

This command will cause the printer to print a list of all graphics stored in the user flash memory.

Syntax

```
GI
```

Remarks

The **GI** command will be executed directly, without appending any Linefeed.

Hint:
Issue a **GI** command after having stored a graphic to make sure the storing was successful and to check the amount of free graphic memory.

Example

```
GI                                        :Prints graphics list
```

GK – Delete Graphics

Description	This command is used to delete a specified graphic or all graphics from the user flash memory.
Syntax	`GK"name"│"*"`
Parameters	"name" By entering a name of a form, that form only will be deleted from memory. "*" By entering an asterisk (*) as wildcard, all forms will be deleted from the user flash memory.
Examples	`GK"LOGO"↵` :Deletes "LOGO" `GK"*"↵` :Deletes all graphics

GM – Store Graphics in Memory

Description This command is used to store PCX graphics files in the user flash memory.

Syntax **GM"name"p$_1$↵"DATA"**

Parameters

"name" This is the name that will be used when retrieving the stored graphic (max. 8 characters). The printer is case sensitive, that is, graphic names will be stored with the exact case entered on the GM command line.

p_1 This is the size of the original .PCX file in bytes. In DOS, the DIR command can be used to determine the exact file size.

"DATA" The graphics data in 1-bit (black & white) PCX format. The resolution of the graphics must match resolution of the printer.

Remarks The **GM** command saves the graphics in the user flash memory, so it will not be lost at power off. Use it for graphics that are used frequently and do not change, for example the logotype of your company. Compare with **GW** command. In a DOS system, the "DATA" portion can be sent to the printer via the parallel port using the DOS COPY command.

Examples Assume you have a PCX file named LOGO.PCX in your current directory. Use a text editor to create a text file called for example STOREIT.TXT and store it in the same directory as the .PCX file.

```
↵
GM"LOGO"1421↵
```

To store the image in the default printer, at the DOS prompt type:

```
COPY STOREIT.TXT PRN
COPY LOGO.PCX PRN /b
```

To store the image in the printer connected to port LPT1, at the DOS prompt type:

```
COPY STOREIT.TXT LPT1:
COPY LOGO.PCX LPT1: /b
```

The **GI** command can be used to verify that the graphic was successfully stored. If not, check that the .PCX file is in 1-bit (black & white) format and that the printer’s free graphics memory is large enough to accommodate the graphics.

Caution

Always make backup copies on the host! EasyCoder PD4 has a flash memory for forms and graphics, which requires special consideration. Avoid storing frequently changing data in flash and use printer drivers developed for EasyCoder C4 and PD4.

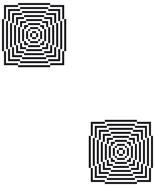
The following sequences show how to generate the image ”PYRAM” and print two copies. Image “PYRAM” is saved as a PCX-file (version 2) in monochrome mode with a size of 32*32 dots. Data below is shown in hex and ASCII format:

```
0000  0D 0A 47 4D 22 50 59 52 41 4D 22 32 39 34 0D 0A  ..GM”PYRAM”294..  Save image
0010  0A 02 01 01 00 00 00 00 1F 00 1F 00 0E 00 0E 00  ................  PCX-header
0020  00 00 00 FF FF FF 00 00 1D 04 00 00 00 00 00 00  ................
0030  38 42 13 00 EC E9 12 00 03 00 00 00 90 41 13 00  8B...........A..
0040  FF FF FF FF 00 00 00 00 FF FF FF FF DC E9 12 00  ................
0050  00 01 04 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................  PCX-header end
0090  C1 FF C1 00 00 C1 FF 80 C1 FF C1 FF 01 BF 80 01  ................  PCX data
00A0  C1 FD A0 7F C1 FE 05 AF C1 C0 03 C1 F5 A8 3F C1  ..............?.
00B0  FC 15 AB C1 E0 07 C1 D5 AA 1F C1 F8 55 6A C1 F0  ............Uj..
00C0  0F 56 5A 8F C1 F1 5A 56 B8 1D 6A 55 A7 C1 E5 AA  .VZ...ZV..jU....
00D0  55 6C 36 AA 55 5B C1 DA AA 55 56 6A AA C1 55 55  Ul6.U[...UVj..UU
00E0  C1 AA AA C1 55 55 C1 AA AA 55 56 6A AA 55 5B C1  ....UU...UVj.U[.
00F0  DA AA 55 6C 36 AA 55 A7 C1 E5 AA 56 B8 1D 6A 5A  ..Ul6.U....V..jZ
0100  8F C1 F1 5A 6A C1 F0 0F 56 AA 1F C1 F8 55 AB C1  ...Zj...V....U..
0110  E0 07 C1 D5 A8 3F C1 FC 15 AF C1 C0 03 C1 F5 A0  .....?..........
0120  7F C1 FE 05 BF 80 01 C1 FD 80 C1 FF C1 FF 01 C1  ................
0130  FF C1 00 00 C1 FF                                ......            PCX data end
```

Now, the image is stored on FLASH and can be used anywhere and several times on the printout, as this example below.

```
N↵
GG0,0,"PYRAM" ↵
GG50,50,"PYRAM" ↵
P↵
```

The result will be a printout shown below:



GRP – Reply Graphics from Printer Buffer

Description This command is used to dump part of the printer buffer back to the host via the serial port.

Syntax **GRP**p_1,p_2,p_3,p_4[,p_5]

Parameters

p_1	X-position in printer dots.
p_2	Y-position in printer dots.
p_3	Number of bytes across the graphic (8 dots = 1 byte).
p_4	Number of dot rows going down the graphic.
p_5	Parameter setting the format of the data. Allowed values are: B Binary, not inversed b Binary, inversed (default) H Hexadecimal, not invesed h Hexadecinmal, inversed

Remarks This command is the reverse of **GW** and can be used to verify that data has been rendered correctly. What part of the print buffer is sent back to the host is specified by the the parameters p_1-p_4 above, and the format of this data by p_5.

Note: There is no flow control or handshaking when sending this data to the host, so use this command only for small areas.

Example The command

GRP20,10,4,32,h↵

will give a reply in hexadecimal format like the one shown below.

```
00 FF FF 00 3F FF FF FE 5F FF FF FE 6F FF FF FE
77 FF FF FE 7B FF FF FE 7D FF FF FE 7E FF FF FE
FF 7F FF FF FF BF FF FF FF DF FF FF FF EF FF FF
FF F7 FF FF FF FB FF FF FF FD FF FF FF FE FF FF
FF FF 7F FF FF FF BF FF FF FF DF FF FF FF EF FF
FF FF F7 FF FF FF FB FF FF FF FD FF 7F FF FE FE
7F FF FF 7E 7F FF FF BE 7F FF FF DE 7F FF FF EE
7F FF FF F6 7F FF FF FA 7F FF FF FC 00 00 00 00
```

GW – Store Graphics in Image Buffer

Description

This command is used to store binary graphics files directly in the image buffer.

Syntax

GW p_1,p_2,p_3,p_4,"DATA"

Parameters

p_1 X-position in printer dots.

p_2 Y-position in printer dots.

p_3 Number of bytes across the graphic (8 dots = 1 byte).

p_4 Number of dot rows going down the graphic.

"DATA" The graphic data in 1-bit (black & white) binary format. First data-byte represent dots in upper left corner and next data-byte next to the right. "1" bits are white dot-positions and "0"-bits are black dot-positions. The total amount of "DATA"–bytes are $p_3 * p_4$.

Remarks

Use this command instead of **GM** for temporarily used graphics, for example images that change between each label. Not only is this method faster, but it also prolongs the life of the flash memory as the graphics are downloaded directly to DRAM.

The printer's firmware will calculate exactly how much data to expect based on p_3 and p_4.

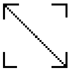
Note that EasyCoder PD4 has a flash memory for forms and graphics, which requires special consideration. Avoid storing frequently changing data in flash and use printer drivers developed for EasyCoder C4 and PD4.

Example

The following sequence shows how to generate and print an image of a reduced square of 32*32 dots, with a diagonal line, all lines 1 dot wide. Data below is shown in hex format:

```
0000   4E 0D 0A 47 57 32 30 2C 31 30 2C 34 2C 33 32 2C  N..GW20,10,4,32,
0010   00 FF FF 00 3F FF FF FE 5F FF FF FE 6F FF FF FE  ....?..._...o...
0020   77 FF FF FE 7B FF FF FE 7D FF FF FE 7E FF FF FE  w...{...}...~...
0030   FF 7F FF FF FF BF FF FF FF DF FF FF FF EF FF FF  ................
0040   FF F7 FF FF FF FB FF FF FF FD FF FF FF FE FF FF  ................
0050   FF FF 7F FF FF FF BF FF FF FF DF FF FF FF EF FF  ................
0060   FF FF F7 FF FF FF FB FF FF FF FD FF 7F FF FE FE  ................
0070   7F FF FF 7E 7F FF FF BE 7F FF FF DE 7F FF FF EE  ...~............
0080   7F FF FF F6 7F FF FF FA 7F FF FF FC 00 00 00 00  ................
0090   50 0D 0A                                         P..
```

The result will be a printout shown below:



I – Character Set Selection

Description This command is used to select the proper character set.

Syntax `Ip`$_1$`,p`$_2$`,p`$_3$

Parameters

p_1	Number of data bits (7 or 8). Default 8.
p_2	Printer Code Page (1 digit, see table 1 below). Default 0.
p_3	KDU Country Code (3 digits, see table 2 below). Default 001. (Only if p_1= 8.)

Table 1. Printer Code Page (p_2)

7 data bits (p_1=7)		8 data bits (p_1=8)		
p_2	Country	p_2	Code Page	Country
0	U.S.A	0	437	United Kingdom
1	United Kingdom	1	850	Multilingual (Latin 1)
2	Germany	4	863	Canada (French)
3	France	5	865	Norway
4	Denmark			
7	Sweden			
8	Switzerland			

In case code pages 437, 863, or 865 cannot produce the desired characters, use code page 850 Multilingual.

Table 2. KDU Country Code (p_3)

Code	Country	Code	Country
001	U.S.A.	041	Switzerland
002	Canada	044	United Kingdom
003	Latin America	045	Denmark
027	South Africa	046	Sweden
031	Netherlands	047	Norway
032	Belgium	049	Germany
033	France	351	Portugal
034	Spain	358	Finland
039	Italy		

For additional code page examples, refer to **Chapter 9**.

Example `I8,1,046↵` :Selects 8 bit character set for use in Sweden with a Keyboard Display Unit.

i – Asian Character Spacing

Description This command places an adjustable inter-character space between Asian font characters. The inter-character spacing gets multiplied with the text string by the selected font's horizontal and vertical multiplier values (See **A** command).

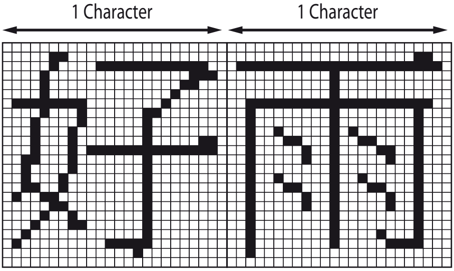
Syntax `ip1`

Parameters p_1 Space in dots between Asian characters (0–9). Default 0

Example `i8↵` :Selects an 8 dots spacing between Asian characters.

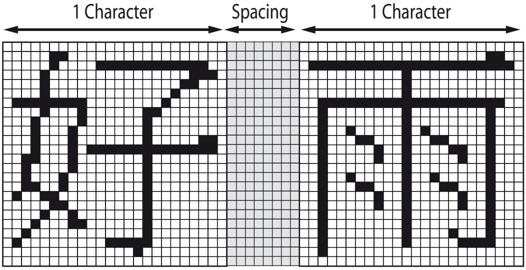
i Command

Parameter p_1 set to 0 (default)



i Command

Parameter p_1 set to 8 (dots)



JB – Disable Top of Form Backup

Description

This command disables automatic top of form backup of the media.

Syntax

```
JB
```

Remarks

Top of form backup is used in connection with the **j** command, which makes the printer feed out an extra amount of media after printing the label, so as to allow the media to be torn or peeled off properly. It does not work with a cutter.

By default, the media is pulled back before printing the first label in next batch as to allow the printing to start at the top of the label, see **JF** command.

The JB command will disable this function, that is any **j** command will be ignored, and the printer will stop feeding when the end of the label becomes aligned with the printhead's dot line. However, the **j** command is kept stored in memory and can be enabled again using a **JF** command. To disabled top of form baclup with a cutter, first send an **O** command followed by a **JB** command.

Example

```
JB↵                                  :Disables top of form backup
```

JF – Enable Top of Form Backup

Description

This command enables automatic top of form backup of the media.

Syntax

```
JF
```

Remarks

Top of form backup is used in connection with the **j** command, which makes the printer feed out an extra amount of media after printing the label, as to allow the media to be torn, peeled, or cut off properly.

By default, top of form is enabled, that is the media is pulled back before printing the first label in next batch as to allow the printing to start at the top of the label.

Top of form backup can be disabled by a **JB** command, that is, any **j** command will be ignored, and the printer will stop feeding when the end of the label becomes aligned with the printhead's dot line. In case of a cutter, the **JB** command must be preceeded by an **O** command. However, the j command is kept stored in memory and can be enabled again using a **JF** command.

Activating the cutter using an **OC** command has the same effect as issuing a **JF** command.

Example

```
JF↵                          :Enables top of form backup
```

j – Media Feed Adjustment

Description

This command makes it possible to set the media feed for tear-off (straight-through), peel-off (self-strip), or cut-off operation.

Syntax

jp$_1$

Parameters

p$_1$ Length of media feed after printing in dots (0-240).

Recommended values at 203.2 dpi (8 dots/mm):

- Tear-off (straight-through) operation: 136 (default)
- Peel-off (self-strip) operation: 110

Recommended values at 300 dpi (11.81 dots/mm):

- Tear-off (straight-through) operation: 204 (default)
- Peel-off (self-strip) operation: 165

Remarks

When using peel-off operation, the labels should remain slightly stuck to the liner (backing paper) so they do not fall off by their own weight, still can be manually removed with ease.

In case of tear-off operation, the media should be fed so the preperforation between tags or the gap between labels become aligned with the tear bar. The **j** command allows the media feed to be adjusted accordingly, that is after the printer has been printed and the rear edge becomes aligned with the printhead's dot line, an extra amount of media feed is performed.

Caution

Do not use extremely small or large values for the j command, since they may cause the printer to feed or pull back the media continuously.

The extra media feed set by the j command can be enabled or disabled using **JF** and **JB** "Top of Form Backup" commands respectively. By default "Top of Form Backup" is enabled.

Examples

`j110↵`	:Adjustment for peel-off operation at 8 dots/mm
`j136↵`	:Adjustment for tear-off operation at 8 dots/mm

LE – Line Draw Exclusive

Description This command is used to draw black lines where the line will be white when intersecting a black area or object and vice versa.

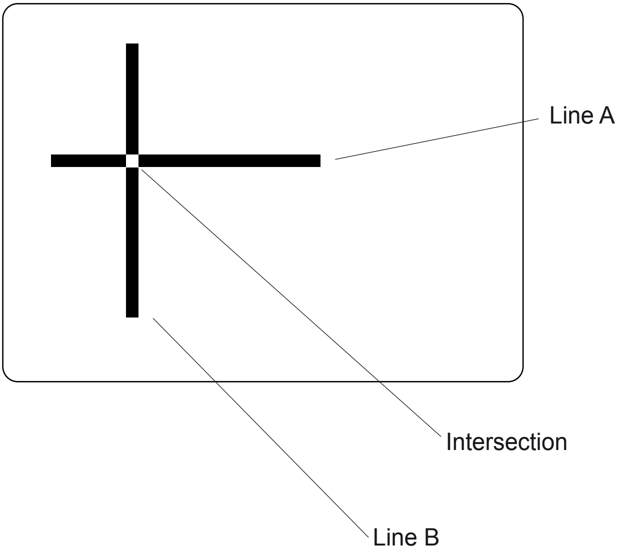
Syntax `LEp1,p2,p3,p4`

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
p_3	Horizontal length in dots.
p_4	Vertical length in dots.

Example

```
N↵                                  :Clears image buffer
LE50,200,400,20↵                    :Draws line A
LE200,50,20,400↵                    :Draws line B
P1↵                                 :Prints one label
```



LO – Line Draw Black

Description This command is used to draw black lines, overwriting previous information.

Syntax

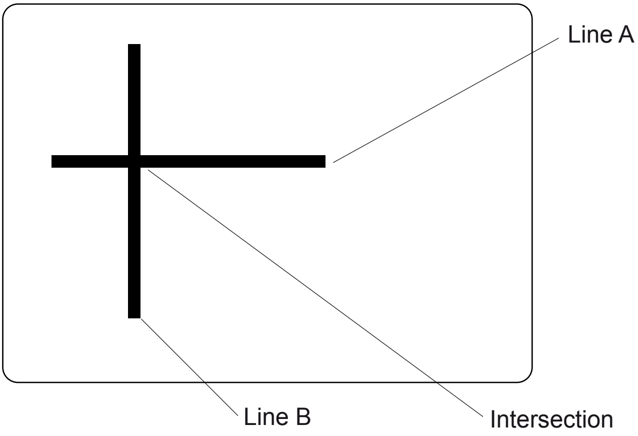
```
LOp1,p2,p3,p4
```

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
p_3	Horizontal length in dots.
p_4	Vertical length in dots.

Example

```
N↵                                   :Clears image buffer
LO50,200,400,20↵                           :Draws line A
LO200,50,20,400↵                           :Draws line B
P1↵                                     :Prints one label
```



LS – Line Draw Diagonal

Description This command is used to draw diagonal black lines overwriting previous information.

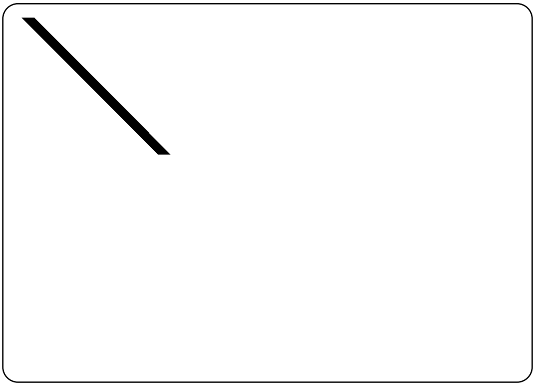
Syntax LSp_1,p_2,p_3,p_4,p5

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
p_3	Line thickness in dots.
p_4	Horizontal end position (X) in dots.
p_5	Vertical end position (Y) in dots.

Example

```
N↵                                  :Clears image buffer
LS10,10,20,200,200↵                 :Draws diagonal line
P1↵                                 :Prints one label
```



LW – Line Draw White

Description This command is used to draw white lines, effectively erasing previous information.

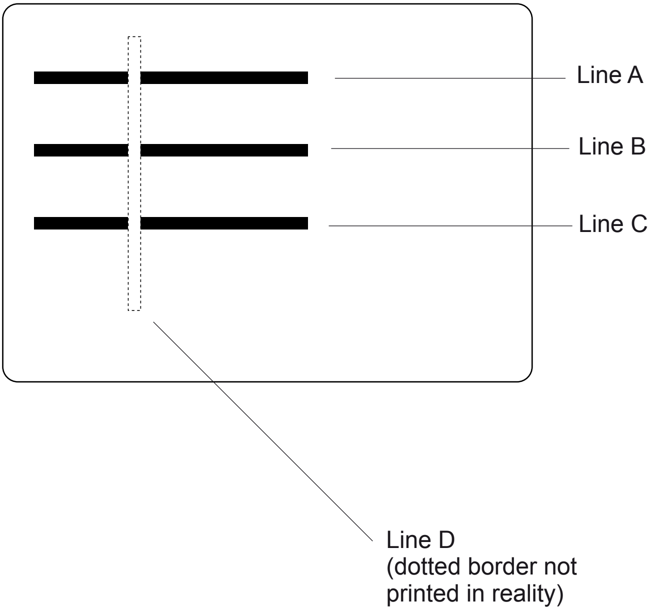
Syntax `LWp1,p2,p3,p4`

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
p_3	Horizontal length in dots.
p_4	Vertical length in dots.

Example

```
N↵                                  :Clears image buffer
LO50,100,400,20↵                    :Draws black line A
LO50,200,400,20↵                    :Draws black line B
LO50,300,400,20↵                    :Draws black line C
LW200,50,20,400↵                    :Draws white line D
P1↵                                 :Prints one label
```



N – Clear Image Buffer

Description This command is used to clear the image buffer before building a new image. It also clears any error list that not yet has been presented.

Syntax **N**

Remarks The **N** command is essential when printing labels in the Direct Mode. It is not necessary to use an **N** command before printing a form. An **N** command must not be used inside a form in the Form Edit Mode.

Example

```
N↵                                    :Clears image buffer
```

O – Options Select

Description

This command is used to enable or disable various sensors and the cutter.

Syntax

```
O[S[,N[,D[,Cnnn|Cb]]]]
```

Parameters

S	Enable reverse gap sensing.
N	Disable label taken sensor.
D	Disable ribbon end sensor
Cnnn	Print nnn labels (1-255) before cutting.
Cb	Print batch before cutting.

Parameters can be entered in any order. Default: ON

Remarks

S: Reverse Gap Sensing Enabled
This parameter reverses the operation of the label gap sensor so it interprets a blockage of light as a gap between labels or similar. Before using the S parameter, make sure to load the EasyCoder PD4 printer with the appropriate type of media. By default, the sensor will interpret blockage of light as a label or similar.

N: Label Taken Sensor Disable
When the label taken sensor is enabled, the communication to the printer will be BUSY as long as the sensor detects a label in the outfeed slot. (Does not work with a cutter—use ON to disable the sensor.)

D: Ribbon End Sensor Disable
The ribbon end sensor (thermal transfer models only) detects reflections from the trailing silvery part of the transfer ribbon.

Direct Thermal Mode: If the ribbon end sensor is disabled parameter, density settings (see **D** command) are interpreted via a table that gives an optimized printout for direct thermal printing.

Thermal Transfer Mode: If the ribbon end sensor is enabled, an error will occur if no reflection is detected. The density settings (see **D** command) are interpreted via a table that gives an optimized printout for thermal transfer printing.

C: Cutter Enabled
If option C is enabled, cutting will be performed after print. Commands **JF** and **JB** will be disabled, but their values will remain stored in memory.

If option C is disabled, no cutting will be performed after print. Commands **JF** and **JB** will be enabled. Command **f** will be disabled but its value will remain stored in memory.

The following table illustrates the results of some combinations of the various O command parameters:

Command	LTS (N)	DT/TTR Mode (D)	Reverse Gap (S)	Cutter (C)
O	Enable	TT Mode	Disable	Disable
OD	Enable	DT Mode	Disable	Disable
OS	Enable	TT Mode	Enable	Disable
OD,S	Enable	DT Mode	Enable	Disable
OS,Cnnn|b	Enable	TT Mode	Enable	Enable
OD,S,Cnnn|b	Enable	DT Mode	Enable	Enable
ON	Disable	TT Mode	Disable	Disable
ON,D	Disable	DT Mode	Disable	Disable
ON,D,S	Disable	DT Mode	Enable	Disable
ON, Cnnn|b	Disable	TT Mode	Disable	Enable
OCnnn|b	not used	TT Mode	Disable	Enable
OD,Cnnn|b	Disabled	DT Mode	Disable	Enable

Example

```
ON,D↵                :Enable DT Mode
                     :Disable LTS
                     :Disable reverse gap sensor
                     :Disable cutter
```

oR – Character Substitution

Description	This command allows the advanced programmer to substitute the Euro currency character (€) for any ASCII character in printer-resident fonts 1-5. The original character can be restored by sending the oR command.	
Syntax	`oR[p1[,p2]]`	
Parameters	p_1	If $p_1 = E$, the Euro character will be mapped to the code page position specified by p_2. If no p_1 or p_2 parameters are given, all code pages will be reset to original default character mapping.
	p_2	Specifies the code page position for the Euro character in the range ASCII 32-255 decimal for all code pages, provided $p_1 = E$. If p_2 is omitted, the Euro character will be mapped to the code page position ASCII 213 decimal for all code pages, provided $p_1 = E$.

Remarks

The **oR** command is a global printer command.

- It cannot be issued inside a form.
- It must be issued prior to issuing a text command and printing it.
- It affects a single character on all code pages. Changing the character position will restore the original character.
- Flash memory printer parameter data are preserved until they are changed by the **oR** command or the printer is reset to default.

Examples

`oRE↵`	**:** Places the Euro character in position ASCII 213 dec.
`oRE,128↵`	:Places the Euro character in position ASCII 128 dec.
`oR↵`	:Clears character substitution and restores default character maps

P – Print

Description This command is used to print the contents of the image buffer.

Syntax `Pp1[,p2]`

Parameters

p_1	Numbers of label sets (1-65535).
p_2	Number of copies of each label (1-65535). Used in combination with counters to print multiple copies of the same label.

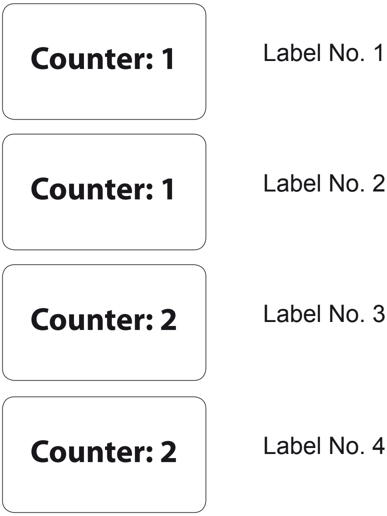
Remarks

Important!
The **P** command cannot be used inside a stored form sequence. For automatic printing of stored forms, use the **PA** command.

Examples

`P↵`	**:** Prints one label set
`P1↵`	:Prints one label set
`P2,1↵`	**:** Prints two label sets of one label each
`P5,2↵`	:Prints five label sets of two labels each

The principles for how counters are printed is illustrated by this example, where the print command is P2,2:



PA – Print Automatic

Description

This command is used in a stored form sequence to automatically print the form as soon as all variable data has been supplied.

Syntax

PAp_1[,p_2]

Parameters

p_1 Numbers of label sets (1-65535).

p_2 Number of copies of each label (1-65535). Used in combination with counters to print multiple copies of the same label.

Remarks

Refer to the **P** command for explanations on how to print multiple labels with counters. The **PA** command follows the same principles.

⚠ Caution

The PA command can only be used with forms containing at least one variable (see V command). If there is no variable in the form, the printer will enter a loop and print continuously!

Examples

```
FK"TEST6"↵                         :Deletes form “TEST6”
FS"TEST6"↵                    :Starts form store sequence
V00,50,N,"Enter text"↵                  :Defines variable
A24,24,0,4,1,1,N,V00↵            :Writes text with variable
PA1↵                         :Prints 1 label automatically
FE↵                             :Ends form store sequence

FR"TEST6"↵                       :Retrieves form “TEST6”
?↵                                        :Gets variables
This is variable text              :Data for variable 00
```

PF – Feed Media

Description This command is used to feed the media forward or backward a given length.

Syntax `PFp1[,p2[,p3]]`

Parameters

p_1 Numbers of steps to feed the media. Positive values will feed the media forward, negative values backward. Range -999 to 9999.

p_2 Position control after feeding. Allowed values:
0 = The media has unknown position. Printer needs label gap/black mark detecting before next print job (default).
1 = Media is located in printing position.
2 = Media feed will stop when Black mark/label gap is in sensor position or, if no black mark/gap is detected, when p_1 dots are stepped (and errror 99 will be reported).

p_3 Optional delay after feeding. Delay will be p_3 * 100 ms.

Remarks Media feeding is disrupted by pressing FEED-key or by opening cover.

Examples The following command will feed the media forward 360 steps, make the printer detect gap/black mark before next print job, and cause a delay of 3 seconds after the feed. Red LED will blink during the delay.

```
PF360,0,30↵
```

The following command will retract the media 360 steps, and assume that the media is in printing position.

```
PF-360,1↵
```

Q – Set Form Length (gap or slot)

Description This command is used to set the form and gap length when using the label gap sensor, or the amount of media feed after the print image in case of continuous stock.

Syntax $Qp_1,p_2[\pm p_3]$

Parameters

p_1	Form length measured in dots.		
	203.2 dpi (8 dots/mm):		
		Default:	800 dots
		Maximum:	10300 dots at full print width
	300 dpi (11.81 dots/mm):		
		Default:	800 dots
		Maximum :	7100 dots at full print width
p_2	Gap length measured in dots.		
	Default:		24 dots at at 203.2 dpi (8 dots/mm) 24 dots at at 300 dpi (11.81 dots/mm)
$\pm p_3$	Optional offset length measured in dots.		

Remarks

Gaps and slots:

The EasyCoder PD4 has a label gap sensor designed to detect the top of each form. It does this by looking either

- through the semi-transparent liner in the gap between labels,
- or through a hole in the media.

The position of the sensor is adjustable across the media path. Refer to the *EasyCoder PD4, User's Guide* for specifications of the size and location of detection slots and instructions for moving the sensor.

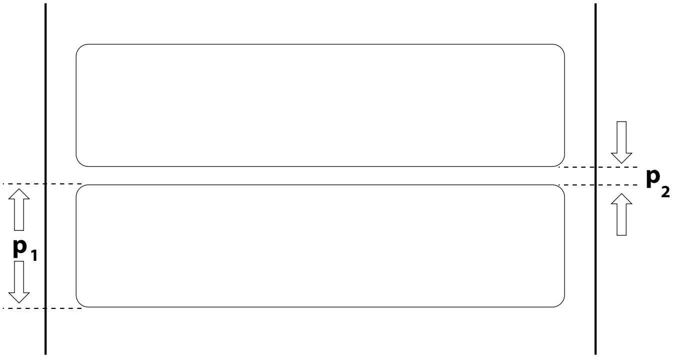
When entering Autosensing Mode by holding the Pause key at startup, the printer automatically determines the Q value while feeding a couple of labels. The current Q value is printed on the test label and the label produced by a **U** command.

Continuous stock:

In case of continuous stock, parameter p_1 decides the amount of media feed performed after the actual print image has been printed. Continuous stock is selected by setting parameter $p_2 = 0$.

Examples

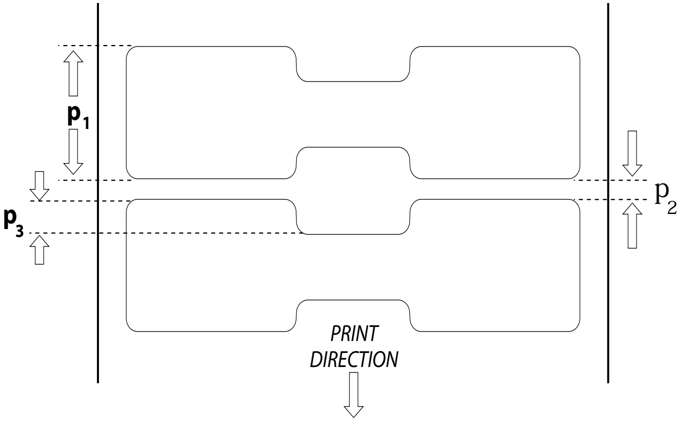
Rectangular label (203.2 dpi = 8 dots/mm printhead):
p_1 = 20.0 mm(160 dots)
p_2 = 3.0 mm (24 dots)



The **Q** command would be:

```
Q160,24↵
```

Butterfly label (203.2 dpi = 8 dots/mm printhead):
p_1 = 12.5 mm (100 dots)
p_2 = 3.0 mm (24 dots)
p_3 = 3.0 mm (24 dots)



The **Q** command would be:

```
Q100,24+24↵
```

Q – Set Form Length (Black Mark)

Description This command is used switch from label gap sensor to the black mark sensor, and to specify the location and height of the black marks on the back of the media.

Syntax

```
Qp1,Bp2[±p3]
```

Parameters

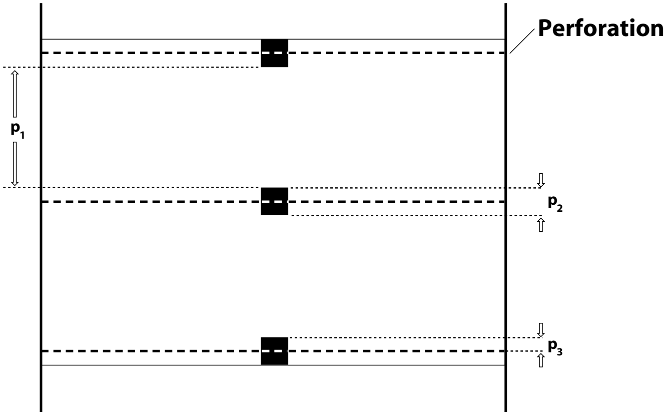
p_1	Distance between black marks measured in dots.
B	Disables label gap sensor, enables black mark sensor.
p_2	Height of black mark measured in dots.
$\pm p_3$	Optional offset length measured in dots.

Remarks In addition to the label gap sensor, all EasyCoder PD4 printers have a black mark sensor that determines the top of each form by sensing a preprinted black mark on the back of the media.

The position of the sensor is adjustable across the media path. Refer to the *EasyCoder PD4, User's Guide* for specifications of the size and location of black marks and instructions for moving the sensor.

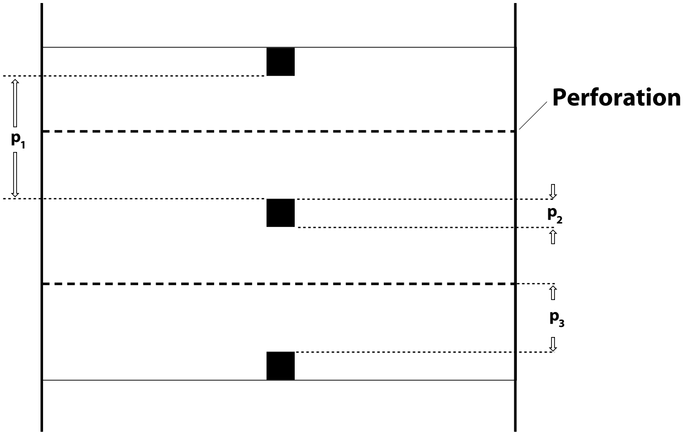
Examples

On this tag, the black marks are printed on the perforation in a 203.2 dpi (8 dots/mm) printer:
p_1 = 31.0 mm (248 dots)
p_2 = 7.0 mm (56 dots)
p_3 = 0.5 mm (4 dots)



The **Q** command would be: `Q248,B56+4↵`

On the tag below, the black marks are printed between the perforations. The printer has a 203.2 dpi (8 dots/mm) printhead.
p_1 = 31.0 mm (248 dots)
p_2 = 7.0 mm (56 dots)
p_3 = 17 mm (136 dots)



The **Q** command would be: `Q248,B56-136 ↵`

q – Set Label Width

Description This command is used to set the label width when using less than full width labels.

Syntax

```
qp1
```

Parameters

p_1	Width of label measured in dots. Default: 832 at 203.2 dpi (8 dots/mm) 1204 at 300 dpi (11.81 dots mm)

Remarks The **q** command will cause the image buffer to be formatted to match the label width, that is width is traded off for increased length within the same memory size.

The **q** command will also automatically set the margins according to the following rule:

(No. of dots on printhead - label width in dots)/2 (left-aligned)

Note: If an **R** command (Reference Point) is sent after a **q** command, the image buffer will be automatically reformatted to match the width of the printhead and the margins will be reset accordingly.

Example

```
q416↵                                  :Sets label width to 416 dots
```

R – Set Reference Point

Description This command is used to move the reference point for the X- and Y-axes. All horizontal and vertical measurements in other commands use the setting for R as the origin for measurements.

Syntax `Rp1,p2`

Parameters

p_1	Horizontal (left) margin measured in dots (default 000).
p_2	Vertical (top) margin measured in dots (default 000).

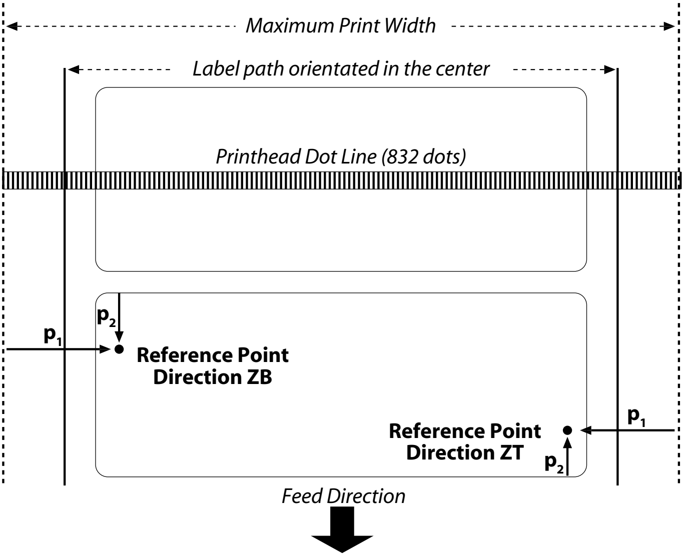
Remarks The reference point command is used to establish top and left margins to prevent printing off the edge of the label. A minimum margin of 1 mm should be used on all sides of the label.

Caution

Repeated printing outside the edge of the media can cause excessive printhead wear.

For narrow labels, the **R** command could be substituted by a **q** command, which has the benefit of making better use of a limited image buffer. However, the **q** command cannot affect the vertical margin. Any **R** command after a **q** command will revoke the latter.

The print direction commands **ZB** and **ZT** affect the location of the reference point, as illustrated below:


Maximum Print Width
Label path orientated in the center
Printhead Dot Line (832 dots)
p1
p2
Reference Point Direction ZB
p1
Reference Point Direction ZT
p2
Feed Direction

Example `R50,100↵` : Creates a 50 dot left margin and a 100 dot top margin.

r – Set Relative Reference Point

Description This command is used to move the reference point for the X- and Y-axes, relative to the label width set by a previous **q** command.

Syntax `rp1,p2`

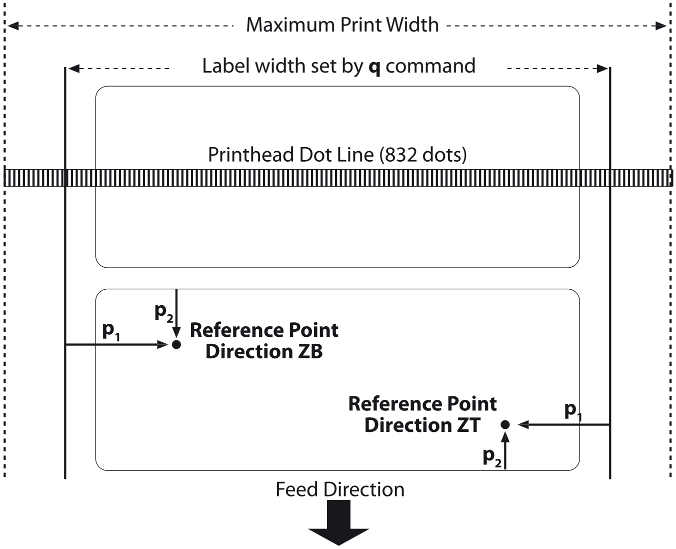
Parameters

p_1	Horizontal (left) margin measured in dots (default 000).
p_2	Vertical (top) margin measured in dots (default 000).

Remarks The relative reference point command is used to establish the top and left margins relative to the label width.

For narrow labels, the **r** command can be used after a **q** command, which has the benefit of making better use of a limited image buffer. The values for the latest **r** command (or **R**, whichever was last executed) are printed on the test label.

The print direction commands **ZB** and **ZT** affect the location of the relative reference point, as illustrated below:


Maximum Print Width
Label width set by q command
Printhead Dot Line (832 dots)
p1
p2
Reference Point Direction ZB
Reference Point Direction ZT
p1
p2
Feed Direction

Example `r50,100↵` **:** Creates a 50 dot left margin and a 100 dot top margin from label edge set by q command.

S – Speed Select

Description This command is used to select the print speed.

Syntax

```
Sp1
```

Parameters

p1 Speed select value:

0	30 mm/sec. (1.2 inches/sec.)
1	40 mm/sec. (1.6 inches/sec.)
2	50 mm/sec. (2 inches/sec.)
3	75 mm/sec. (3 inches/sec.)
4	100 mm/sec. (4 inches/sec.) Default
5	125 mm/sec. (5 inches/sec.) 203.2 dpi (8 dots/mm) printers only.
6	150 mm/sec. (6 inches/sec.) 203.2 dpi (8 dots/mm) printers only.

Remarks Changing the print speed will affect the blackness of the printout, which may have to be adjusted using a **D** command.

Example `S2↵` :Sets the print speed to 50 mm/sec. (2 inches/sec.).

SA – Setup Adjust

Description This command is used to control the activation and storage of setup commands / values.

Syntax `SAp1`

Parameters p_1 Setup Adjust value:

0 All setup commands remain active and values are saved when changed. The printer will use stored setup values (Default).

1 All setup parameters are locked and commands will be ignored. The printer will use previously stored setup values.

2 All setup commands within forms will be ignored.

Remarks The **SA** command affects the following setup commands: **R**, **r**, **D**, **j**, **J**, **q**, **Q**, **S**, **Z**, and **SPC**. The current **SA** value is stored in Flash memory and printed on the test label.

Example

```
SA2↵
```

:All of the commands listed above that are called within forms will be ignored.

SPC – Setup Print Copy

Description

This command is used to change the function of the **Feed** key to print a copy of the label residing in the printer's image buffer.

Syntax

`SPCp1`

Parameters

p_1 Setup Print Copy value:

0 Media will feed as usual when the Feed key is pressed (Default).

1 Print Copy is activated, but will be disabled if any of the commands listed below have been sent since the last printout.

2 Print Copy is activated. In this state, additional fields can be added to the existing image and printed by pressing the Feed key.

Remarks

The following commands will disable Print Copy when **SPC1** is set: **A, B, b, GW, GG, LE, LO, LS, LW, N, q, Q, r, R, X** and **Info**. The current **SPC** value is stored in Flash memory and printed on the test label.

Example

```
SPC1↵                    :Print Copy is activated.
```

SPR – Setup Print Engine Ribbon Control

Description

When the printer is operated in thermal transfer mode, this command allows the user to specify the number of steps executed by the stepping motor AFTER the ribbon supply hub has stopped rotating. Once the motor has executed the number of steps given by the command, the printer will signal "Out of Ribbon".

Syntax

```
SPRp1
```

Parameters

p_1	Number of steps executed by stepping motor	
	Default:	400 steps
	Maximum:	32737 steps

Remarks

This is a feature exclusive to the PD4 printer. The SPR value holds no relevance on EasyCoder PC4 printers.

Example

`SPR200↵`	:The stepper motor will count 200 steps before signalling Out of Ribbon.

TD – Define Date Format

Description This command is used to define the date format when printing.

Syntax `TDp1[/p2/p3]`

Parameters

p_1–p_3 The parameters describe the format of the date display. At least one parameter must be supplied. Each parameter can be any of the acceptable values listed below:

y2	Year displayed as 2 digits, for example 04
y4	Year displayed as 4 digits, for example 2004
me	Month displayed as a 3-letter English abbreviation, for example JAN, FEB, MAR, APR, MAY, etc.
mn	Month displayed as 2 digits, for example 01
dd	Day displayed as 2 digits, for example 15
/	Represents an optional separator character, which can be any character in the range between ASCII 32 dec. and ASCII 63 dec. The separator is printed between the results of each of the supplied parameters.

Default: me/dd/y4

Remarks This command works only if the printer is fitted with an optional real-time clock circuit (RTC).

Examples If the current date is January 15, 2004:

```
TDy2/me/dd↵          :Prints as 04/JAN/15
TDdd-me-y4↵          :Prints as 15-JAN-2004
TDdd,mn,y4↵          :Prints as 15,01,2004
TDy4-mn-dd↵          :Prints as 2004-01-15
TDme/dd/y4↵          :Prints as JAN/15/2004
```

TS – Set Real Time Clock

Description This command is used to set the time and date in the printer's optional real-time clock circuit.

Syntax `TS`p_1`,`p_2`,`p_3`,`p_4`,`p_5`,`p_6

Parameters

p_1	Month (01–12)
p_2	Day (01–31)
p_3	Year, two last digits (for example 04 for the year 2004)
p_4	Hour in 24 hour format (00–23)
p_5	Minutes (00-59)
p_6	Seconds (00-59)

Remarks This command works only if the printer is fitted with an optional real-time clock circuit (RTC).

Example `TS01,15,04,12,45,23↵` :Sets the date to JAN 15, 2004 and the time to 12:45:23 p.m.

TT – Define Time Format

Description This command is used to define the time format when printing.

Syntax **TTp$_1$ [/p$_2$/p$_3$] [+]**

Parameters

p$_1$–p$_3$ These parameters describe the format of the time display. At least one parameter must be supplied. Each parameter can be any of the acceptable values listed below:

h Hours displayed as 2 digits, for example 12

m Minutes displayed as 2 digits, for example 15

s Seconds displayed as 2 digits, for example 00

/ Represents an optional separator character, which can be any character in the range between ASCII 32 dec. and ASCII 63 dec. The separator is printed between the results of each of the supplied parameters.

+ Optionally selects 12-hour mode, where the time will be appended with an "AM" or "PM" indicator. If there is no trailing + sign in command, 24-hour mode will be selected.

Default: h:m:s

Remarks This command works only if the printer is fitted with an optional real-time clock circuit (RTC).

Example If the current time is 1:25:00 PM:

```
TTh:m:s↵                        :Prints as 13:25:00
TTh:m:s+↵                       :Prints as 01:25:00 PM
TTh,m↵                          :Prints as 13,25
TTh+↵                           :Prints as 01 PM
```

U – Print Configuration (General)

Description	This command is used to print the current printer configuration.
Syntax	`U`
Remarks	This command produces a single label identical to the one printed in the Dump Mode (see Chapter 1), but without entering the Dump Mode.
Example	`U↵` :Produces a test label.

UC – Command Reply Accept Character Control

Description This command specifies a prompt acknowledge character which the printer returns on the RS-232 or USB channel to the host after each command execution with no error reply before proceeding with next command.

Syntax `UCp1`

Parameters

p_1	Specifies the desired reply character by its ASCII decimal value (0-255). If p_1 is anything other than numerical characters or the value is zero, there will not be any Command Reply Accept Character (default value).

Remarks Default is no Command Reply Accept Character (command UC0).

Parameter p_1 saved in Flash.

If command UC is activated, it will replace the <ACK>-character at end of command **US** and **P**.

If error occurs and error report is activated, the error message will be sent instead.

Example

Command	Reply	Comments
UC062 ↵	>	Set Command Reply Accept Character to ‘>’
US ↵	>	Reply error
X50,200,5,400,20↵	>	Draw a box
P↵	>	Print box & reply ‘>’ at ready

UE – Soft Font Information Inquiry

Description

This command makes the printer send information back to the host on the soft fonts stored in memory.

Syntax

```
UE
```

Remarks

The printer sends the number of soft fonts and the name, height, and direction of each soft font through the RS-232 or USB port.

The **UE** command will be executed directly, without appending any Linefeed.

Example

```
UE
```

UF – Form Information Inquiry

Description This command will cause the printer to send information about forms currently stored in the printer back to the host.

Syntax `UF`

Remarks The printer will send the number of forms stored and the name of each form to the host through the RS-232 or USB port.

The **UF** command will be executed directly, without appending any Linefeed.

Example `UF` :Returns number of forms and all form names, for example:

```
UF006
TEST1
TEST2
TEST3
TEST4
TEST5
TEST6
```

UG – Graphics Information Inquiry

Description This command will cause the printer to send information about graphics currently stored in the printer back to the host.

Syntax `UG`

Remarks The printer will send the number of graphics and the name of each graphic to the host through the RS-232 or USB port.

The **UG** command will be executed directly, without appending any Linefeed.

Example `UG` :Returns number of graphics and all graphic names, for example:

```
UG001
LOGO
```

UI – Enable Prompts/Code Page Inquiry

Description This command will cause the printer to enable prompts to be sent to the host and to send the currently selected code page to the host through the RS-232 or USB port.

Syntax

```
UI
```

The printer will send information on the currently selected code page back to the host in the following format:

UIp_1p_2,p_3

Parameters

p_1	Number of data bits.
p_2	Code page.
p_3	Country code.

Remarks The KDU (Keyboard Display Unit) automatically sends this command each time power is applied. The **UI** command is disabled by removing power from the printer for 60 seconds.

Example

```
UI↵
```

: Enables prompts from host and returns current code page, for example

```
UI80,001
```

Also see **I** and **U** commands.

UM – Code Page & Memory Inquiry

Description This command will cause the printer to send the currently selected code page and memory status to the host through the RS-232 or USB port.

Syntax

```
UM
```

The printer will send information on the currently selected code page and memory status back to the host in the following format:

UMp_1**,**p_2**,**p_3**,**p_4**,**p_5**,**p_6**,**p_7 **UI**p_8p_9**,**p_{10}**[,**p_{11}**]**

Parameters

p_1	Image buffer size in kilobytes.
p_2	Total User Flash-memory size for [Form + Graphic + External font + Asian Font] in kilobytes incl. decimals.
p_3	Form memory used in kilobytes. decimals = forms file counter
p_4	Graphic memory allocation size in kilobytes.
p_5	Graphic file counter.
p_6	External font memory allocation size in kilobytes.
p_7	External font file counter
p_8	Number of data bits.
p_9	Code page.
p_{10}	Country code.

p_{11}	Asian Font file flag & used memory in kilobytes;
	Annn = Chinese traditional
	Bnnn = Chinese simplified
	Cnnn = Japanese
	Dnnn = Korean
	where nnn = memory used in kilobytes for Asian font

Example

UM↵ :Returns memory status and current code page, for example:

```
UM1050,998.0,8.3,40,1,37,2 UI80,001,B581
```

Also see **I**, **U**, **UI**, and **UP** commands.

UN – Disable Error Reporting

Description This command is used to disable error reporting.

Syntax

```
UN
```

Remarks Cancels **US** command.

Example

```
UN↵                                        :Disables error reporting
```

UP – Code Page & Memory Inquiry/Print

Description

This command will cause the printer to print and send the currently selected code page and memory status to the host through the RS-232 or USB port.

Syntax

```
UP
```

The printer will:

- Send information on the currently selected code page and memory status back to the host (same as **UM** command).
- Print the current printer configuration (same as **U** command).

The format of the data sent to the host is as follows:

UMp_1,p_2,p_3,p_4,p_5,p_6,p_7 **UI**$p_8$$p_9$,$p_{10}$[,$p_{11}$]

Parameters

p_1 Image buffer size in kilobytes.

p_2 Total User Flash-memory size for [Form + Graphic + External font + Asian Font] in kilobytes incl. decimals.

p_3 Form memory used in kilobytes. decimals = forms file counter

p_4 Graphic memory allocation size in kilobytes.

p_5 Graphic file counter.

p_6 External font memory allocation size in kilobytes.

p_7 External font file counter

p_8 Number of data bits.

p_9 Code page.

p_{10} Country code.

p_{11} Asian Font file flag & used memory in kilobytes;
Annn = Chinese traditional
Bnnn = Chinese simplified
Cnnn = Japanese
Dnnn = Korean
where nnn = memory used in kilobytes for Asian font

Example

UP↵ :Returns memory status and current code page and prints configuration on label.

Also see

I, **U**,**UI**, and **UM** commands.

US – Enable Error Reporting

Description This command is used to enable the printer's status reporting feature with optional direct recovery.

Syntax `US [A] [B] [E]`

Parameters

A	If any error of type A occurs, it will be directly recovered and continue in normal mode. No stop for recovery, often performed with FEED key.
B	If any error of type B occurs, it will be directly recovered and continue in normal mode. No stop for recovery, often performed with FEED key.
E	Error 13 LTS reply control (label taken sensor). Returns error 13 if LTS detects a label before and after printing.

Remarks Any or all parameters can be activated at the same time. Refer to Appendix A for information on error types.

Parameters are saved in the Flash memory.

The format of the error messages is controlled by the **eR** command.

Example

Command	Reply[1]	Comment
USABE↵	<ACK>	Activate all error flags
eRQ,6,R↵	*(no reply)*	Setup User-Defined Error/Status Character control to Q for <XOFF> and <R> for <XON>
N↵	*(no reply)*	Clear image buffer and errors
AA↵	*(no reply)*	Erroneous command type A; Syntax error
P↵	<NAK>01Q	Print image buffer with error 01
	R	Direct recovery error 01 (type A) and continue

[1]/. Reply only works on serial and USB ports.

UV – Product Identity and Asian Font Types

Description This command generates an output on the RS-232 or USB port about the software and font products stored in the user flash memory.

Syntax

```
UV
```

Remarks The first line returned after the **UV** command is the base software appended by a CR/LF. Then comes one line with information on the font appended by CR/LF.

Example

```
UV
```

returns for example...

```
"1-972660-00,PD4-ESim 5.12"           Base software

"1-972650-01,Font GB2312-80"           Chinese font
                                       GB 2312-80 or...

"1-972660-00,PD4-ESim 5.12"           Base software

"1-972651-01,Font BIG5"        Chinese Big5 level 1 & 2
                                               font or...

"1-972660-00,PD4-ESim 5.12"           Base software

"1-972652-00,Font KSX1001:1992"         Korean font

                                    KS X 1001:1992 or...

"1-972660-00,PD4-ESim 5.12"           Base software

"1-972653-00,Font JISX0208:1997"     Japanese font
                                       JISX0208:1997
```

V – Define Variable

Description

This command is used to define variable data fields for use in stored forms.

Syntax

Vp$_1$,p$_2$,p$_3$,"PROMPT"

Parameters

p$_1$ Variable reference number (00-99).
A maximum total of 1500 bytes of data for all variables is allowed.

p$_2$ Maximum number of digits for the variable (1-99).
A maximum total of 1500 bytes of data for all variables is allowed.

p$_3$ Field justification:
L Left justification.
R Right justification.
C Center justification.
N No justification.

[-] A single leading minus sign in the prompt field will cause the prompt to be sent one time only after the form is retrieved (Keyboard Display Unit only).

[- -] A double leading minus sign in the prompt field will cause the prompt to be suppressed (Keyboard Display Unit only).

"PROMPT" An ASCII text field that will be transmitted to the host or Keyboard Display Unit via the serial interface each time this command is executed. This prompt requests the operator to enter the value for the variable.

Remarks

This command is used in forms that require unique data on each label. When initializing variables, they must be defined in order (V00, V01, V02, etc.) immediately after the **FS** command.

The field justification parameter affects the way the variable will be printed. When left, right, or center justification are selected, the counter value will be printed left, right or center justified in an area with a width defined by the p$_2$ parameter. If the number of digits in the counter value is less than the number of digits defined by p$_2$, the area will be padded with space characters.

If no justification is selected, the field will adjust to fit the actual length of the data and will not exceed the set maximum field length, which may be useful when using a counter as input data to a bar code.

To print the contents of a variable, the number of the variable must be included in the "DATA" field of the **A** (Print Text) or **B** (Print Bar Code) commands.

Example This example shows how the field justification works in variable fields:

```
FK"TEST7"↵
FS"TEST7"↵
V00,10,L,"Variable 00"↵
V01,10,R,"Variable 01"↵
V02,10,C,"Variable 02"↵
V03,10,N,"Variable 03"↵
A50,50,0,3,1,1,N,"TEXT"V00":Left justified"↵
A50,100,0,3,1,1,N,"TEXT"V01":Right justified"↵
A50,150,0,3,1,1,N,"TEXT"V02":Center justified"↵
A50,200,0,3,1,1,N,"TEXT"V03":No justification"↵
FE↵
```

Refer to the **?** command for continuation of this example!

W – Windows Mode

Description This command is used to enable/disable the Windows command mode (special applications only).

Syntax `Wp1`

Parameters p_1 Windows Mode enable/disable:

Y Enables Windows Mode.

N Disables Windows Mode (default).

Remarks When enabled, the printer will accept Windows mode escape sequences to print data. When disabled, escape sequences will be ignored up to the next linefeed.

The Windows mode escape sequences are only used by the Windows Printer Driver. When working with a main frame or other non-Windows host, this mode can be disabled to prevent erratic operation.

Examples

`WY↵`	:Enables Windows Mode
`WN↵`	:Disables Windows Mode

X – Draw Box

Description This command is used to draw a box shape.

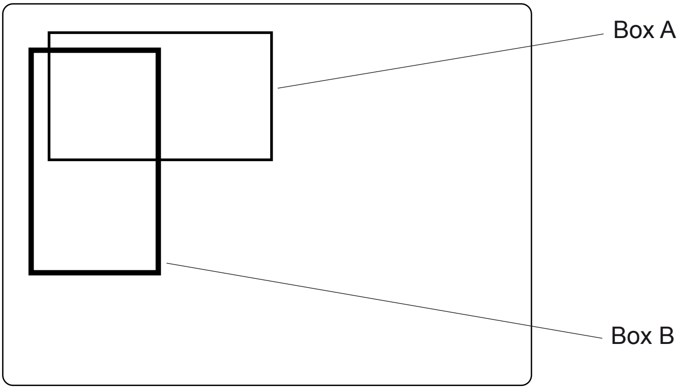
Syntax Xp_1,p_2,p_3,p_4,p_5

Parameters

p_1	Horizontal start position (X) in dots.
p_2	Vertical start position (Y) in dots.
p_3	Line thickness in dots.
p_4	Horizontal end position (X) in dots.
p_5	Vertical end position (X) in dots.

Example

```
N↵                                  :Clears image buffer
X50,200,5,400,20↵                   :Prints box A
X200,50,10,20,400↵                  :Prints box B
P1↵                                 :Prints a label
```



Y – Serial Port Setup

Description This command is used to establish the communication parameters of the serial RS-232 port.

Syntax `Yp1,p2,p3,p4`

Parameters

p_1	Baud rate:	
	48	4,800 bps
	96	9,600 bps
	19	19,200 bps
	38	38400 bps
p_2	Parity:	
	O	Odd. (O is uppercase o character; ASCII 79 dec.).
	E	Even.
	N	None.
p_3	Number of data bits:	
	7	7 data bits.
	8	8 data bits.
p_4	Number of stop bits:	
	1	1 stop bit.
	2	2 stop bits.

Remarks After receiving this command, the printer will automatically reset its communication on the serial RS-232 communication port. There are no such parameters for the USB port.

By default, the printer is set for 9600 baud, no parity, 8 data bits, 1 stop bit.

XON/XOFF handshaking is always used. The printer sends XOFF when an error occurs. RTS/CTS is not supported

If the current communication setup is not known, it can be checked by printing a test label (see Chapter 1).

Example

```
Y19,O,7,1↵
```

:Sets 19,200 baud, odd parity, 7 data bits, 1 stop bit

Z – Print Direction

Description This command is used to select the print orientation.

Syntax `Zp`$_1$

Parameters p$_1$ Print orientation:

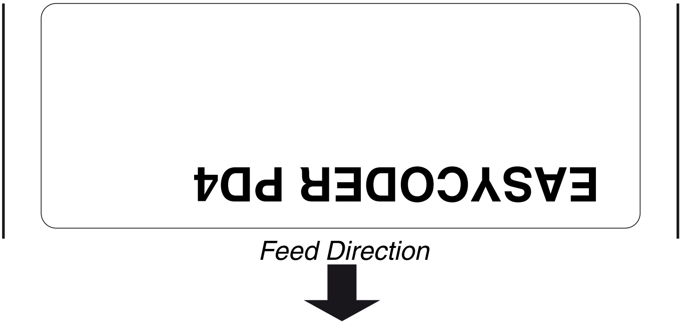
T Start printing from the top of image buffer (default).

B Start printing from the bottom of image buffer.

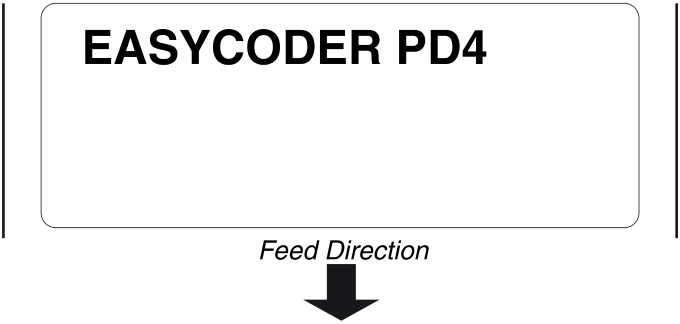
Remarks This command affects the complete print image, including text, bar codes, graphics, lines, and boxes, as well as the location of the reference point (see **R** command).

Note that printing a test label in the Test Mode, or by means of a **U** or **UP** command, will reset the print direction to default (= **ZT**).

ZT Command:



ZB Command:



Example `ZB↵` :Starts printing from the bottom of the image buffer.

? – Download Variables

Description

This command is used to signal to the printer that the data following are variable or counter values.

Syntax

```
?
```

Remarks

This command is used by the host system to send data representing variables and/or counters to the printer after a stored for containing variables and/or counters has been retrieved. The amount of data following the question mark line must match exactly the total number and order of variables and/or counters for that specific form.

Note: If the **?** command is omitted, no variables or counter values will be printed.

Example

```
FR"TEST7"↵          :Retrieves the form “TEST7”
?↵                  :Variables follow
12345↵              :Variable 00 entered
abcde↵              :Variable 01 entered
ABCDE↵              :Variable 02 entered
99999↵              :Variable 03 entered
P1↵                 :Prints one label
```

^@ – Reset Printer

Description This command resets the printer.

Syntax **^@**

Remarks The **^@** command resets the printer in the same way as a power off followed by a power on.

- The command must be followed by CR/LF to be correctly interpreted.
- The reset command is only available during command input, that is, not as data in PCX-graphics, soft fonts, or in format forms.
- The printer will be unavailable for a short time during which all commands sent to the printer will be lost.

Note: An error report has higher priority than other commands and is processed directly, both in position of normal command-input and in position of any error recovery.

An **^@** command cannot be performed anywhere else, such as
- during downloading of images, fonts, or forms, or
- inside a variable-input (?) or inside a string-area (""), or
- when the input data buffer is full (> 8000 bytes).

Example

```
^@ ↵                                   :Printer will be reset.
```

^default – Restore Factory Default

Description This command set all parameter back to the factory default.

Syntax

```
^default ↵
```

Remark Many recommendations, especially when troubleshooting or updating firmware, require the need to start from a known state. This command returns the printer to its default setup. See Default Setup in Chapter 3, for the default value of setup parameters. A delay of two seconds should be observed before inputting other commands as the printer restarts after cleanup to factory default.

Example Return to factory default setup.

```
^default↵
```

^ee – Immediate Error Report

Description This command makes the printer report error and status immediately via the RS-232 or USB port.

Syntax

```
^ee ↵
```

Remarks After the command is sent into printer via the RS-232or USB interface, the printer returns an error and status report on serial port, when reaching an position of normal command-input or of error-waiting-recovering.

Command must end with <CR> or <LF> to be correctly executed.

The printer will reply to host via serial port in one of the following formats:

1 If no error code is available (normal case):
`00<CR><LF>`
00 = Status code OK

2 If only one error code is available:
`XX<CR><LF>`
XX = Error or Status code

3 If more error or status codes available:
`XX,YY,ZZ,..<CR><LF>`
XX YY ZZ .. = Error or Status codes

Note: An error report has higher priority than other commands and is processed directly, both in position of normal command-input and in position of any error recovery.

An ^ee command cannot be performed anywhere else, such as
- during downloading of images, fonts, or forms, or
- inside a variable-input (?) or inside a string-area (""), or
- when the input data buffer is full (> 8000 bytes).

At “position error-waiting-recovering”, the command ^ee must be sent into printer via the RS-232 or USB interface, to get an error report.

Example

Serial command	Manual action	Reply on RS232	Comments
US↵		<ACK>	Enable Error Reporting
AA↵		*(Nothing happens)*	Command with syntax error (#01)
	Lift the printhead	<NAK>11<XOFF>	Manual action to generate error (#11) and wait in position of error-waiting-recovering.
UV↵		*(Nothing happens)*	Command to print product ID will be queued
^ee↵		01,11<CR><LF>	Command reports direct all errors/status
	Close the printhaed	<XON>1-972660-00,→ →PD4 5.12<CR><LF>	Manual action to release error #11. Command UV will now be performed.
P↵		<NAK>01<XOFF>	Command to print, reports syntax error (#01)
	Press FEED key	<XON>	Manual action to release error status

8 Fonts

This chapter lists the printer's resident fonts and illustrates the various font sizes.

Resident Fonts

The EasyCoder PD4 printers support upper- and lowercase characters for font sizes 1-4 and uppercase characters for font size 5. All fonts are non-proportional. The ASCII value of the different characters is determined by the **I** command setting.

203.2 dpi (8 dots/mm) (illustrated in Chapter 8 and 9)

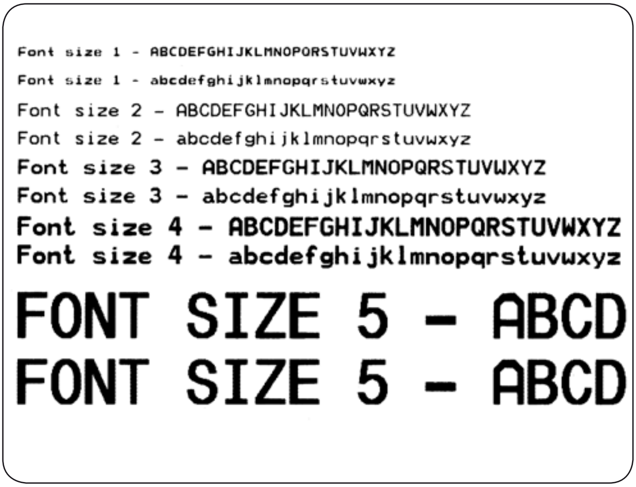
Font	Size (dots)	Size (points)	Characters/inch
1	8 x 12	6	20.3
2	10 x 16	7	16.9
3	12 x 20	10	14.5
4	14 x 24	12	12.7
5	32 x 48	24	5.6

300 dpi (11.81 dots/mm) (not illustrated)

Font	Size (dots)	Size (points)	Characters/inch
1	12 x 20	4	25
2	16 x 28	6	18.75
3	20 x 36	8	15
4	24 x 44	10	12.5
5	48 x 80	214	6.25

Font Sizes 1-5

Below, the various fonts are illustrated in real size as printed on an 8 dots/mm (203.2 dpi) printer.



Font size 1 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 1 - abcdefghijklmnopqrstuvwxyz

Font size 2 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 2 - abcdefghijklmnopqrstuvwxyz

Font size 3 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 3 - abcdefghijklmnopqrstuvwxyz

Font size 4 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 4 - abcdefghijklmnopqrstuvwxyz

FONT SIZE 5 - ABCD

FONT SIZE 5 - ABCD

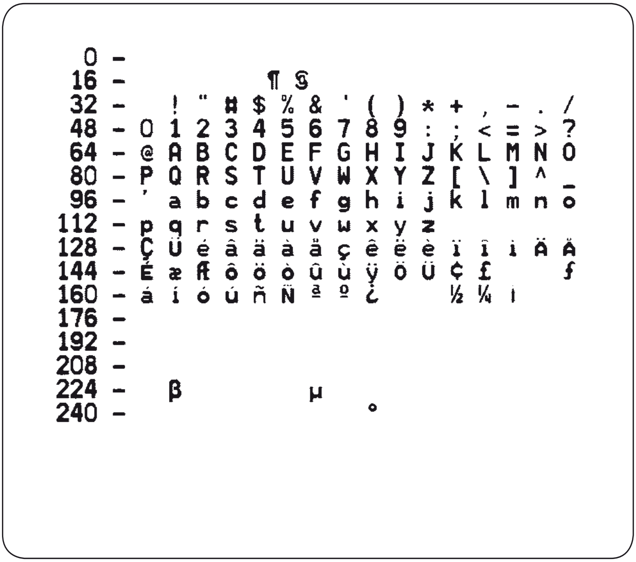
9 Code Pages and Character Sets

This chapter contains printout samples in 100% size of the various code pages and character sets for the printer's resident fonts. All printouts are produced on an EasyCoder PD4 with a printhead density of 203.2 dpi (8 dots/mm).

For information and printout samples on Asian fonts, plese refer to the manual *EasyCoder C4 Asian Fonts*.

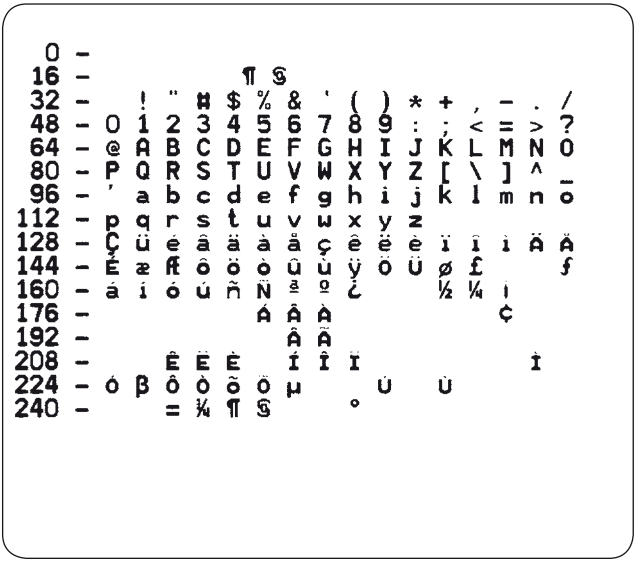
Printout Samples

Size 1-4 (8 bit); Code page 437
(printed in size 4)



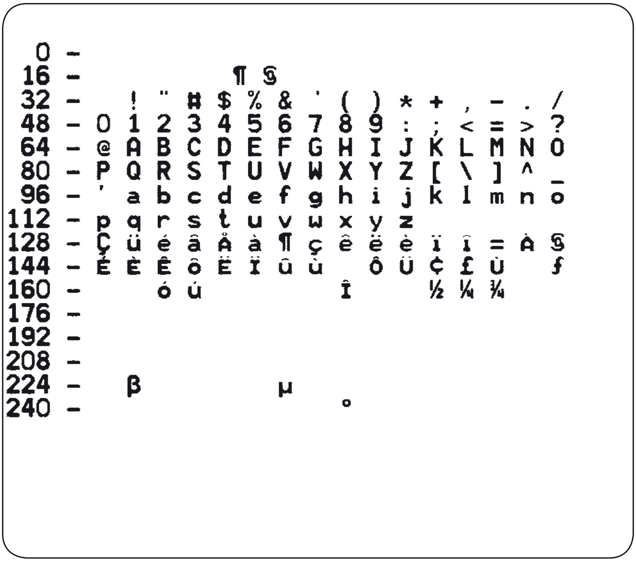
```
  0 -
 16 -         ¶ §
 32 -   ! " # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
128 - Ç Ü é â ä à å ç ê ë è ï î ì Ä Å
144 - É æ Æ ô ö ò û ù ÿ Ö Ü ¢ £     ƒ
160 - á í ó ú ñ Ñ ª º ¿     ½ ¼ ¡
176 -
192 -
208 -
224 -   β         µ
240 -                 °
```

Size 1-4 (8 bit); Code page 850
(printed in size 4)



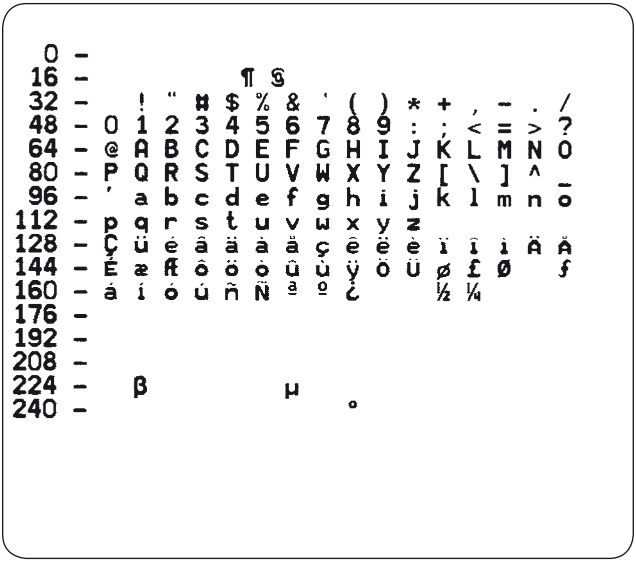
```
  0 -
 16 -         ¶ §
 32 -   ! " # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
128 - Ç ü é â ä à å ç ê ë è ï î ì Ä Å
144 - É æ Æ ô ö ò û ù ÿ Ö Ü ø £     ƒ
160 - á í ó ú ñ Ñ ª º ¿     ½ ¼ ¡
176 -           Á Â À           ¢
192 -             Â Ã
208 -     Ê Ë È   Í Î Ï           Ì
224 - ó β Ô Ò õ Õ µ     Ú   Ù
240 -     = ¾ ¶ §     °
```

Size 1-4 (8 bit);
Code page 863
(printed in size 4)



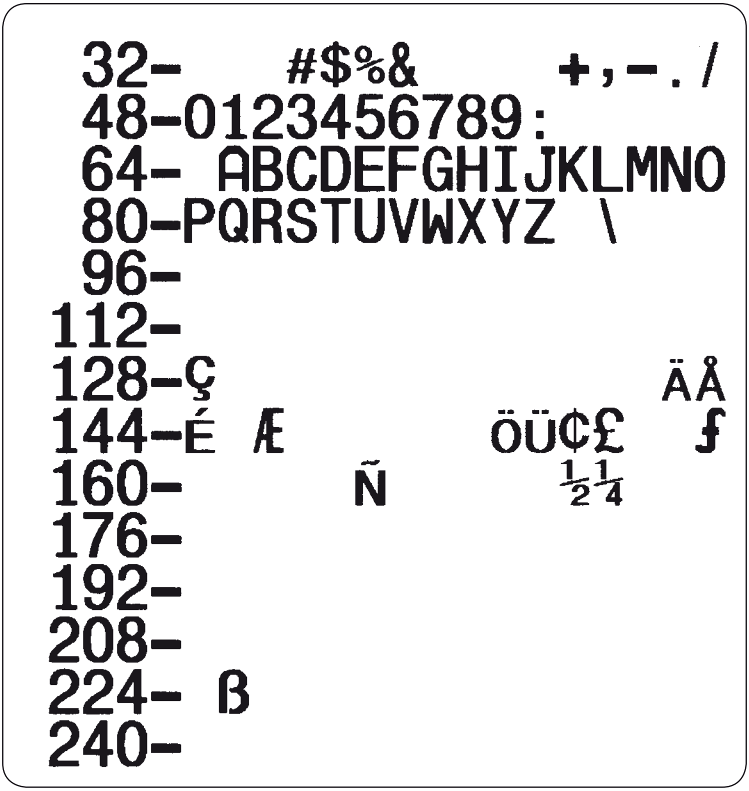
```
  0 -
 16 -         ¶ §
 32 -   ! " # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
128 - Ç ü é â Â à ¶ ç ê ë è ï î = À §
144 - É È Ê ô Ë Ï û ù   Ô Ü ¢ £ Ù   ƒ
160 -     ó ú         Î     ½ ¼ ¾
176 -
192 -
208 -
224 -   β         µ
240 -                 °
```

Size 1-4 (8 bit);
Code page 865
(printed in size 4)



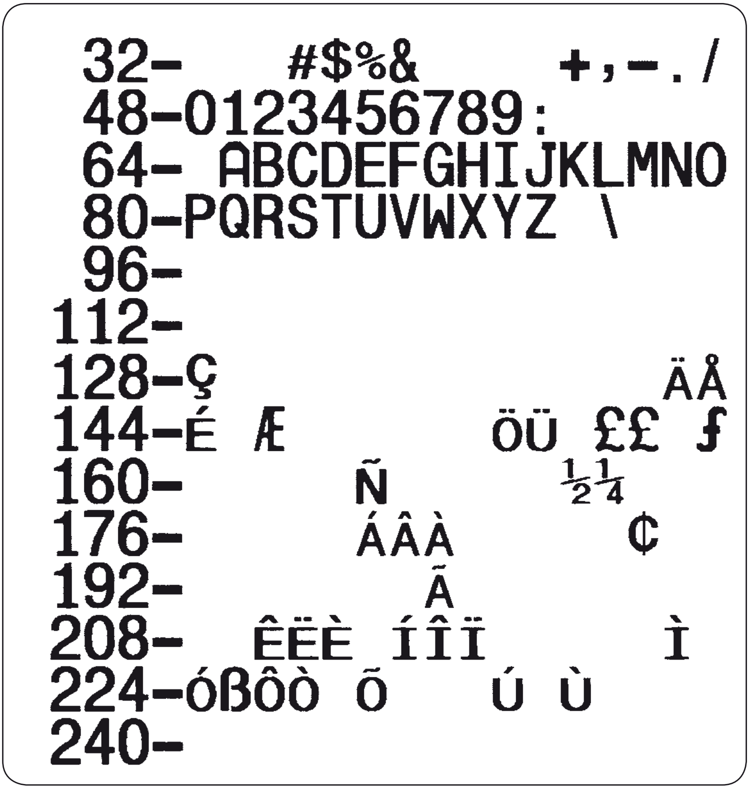
```
  0 -
 16 -         ¶ §
 32 -   ! " # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
128 - Ç ü é â ä à å ç ê ë è ï î ì Ä Å
144 - É æ Æ ô ö ò û ù ÿ Ö Ü ø £ Ø   ƒ
160 - á í ó ú ñ Ñ ª º ¿     ½ ¼
176 -
192 -
208 -
224 -   β         µ
240 -                 °
```

Size 5 (8 bit); Code page 437



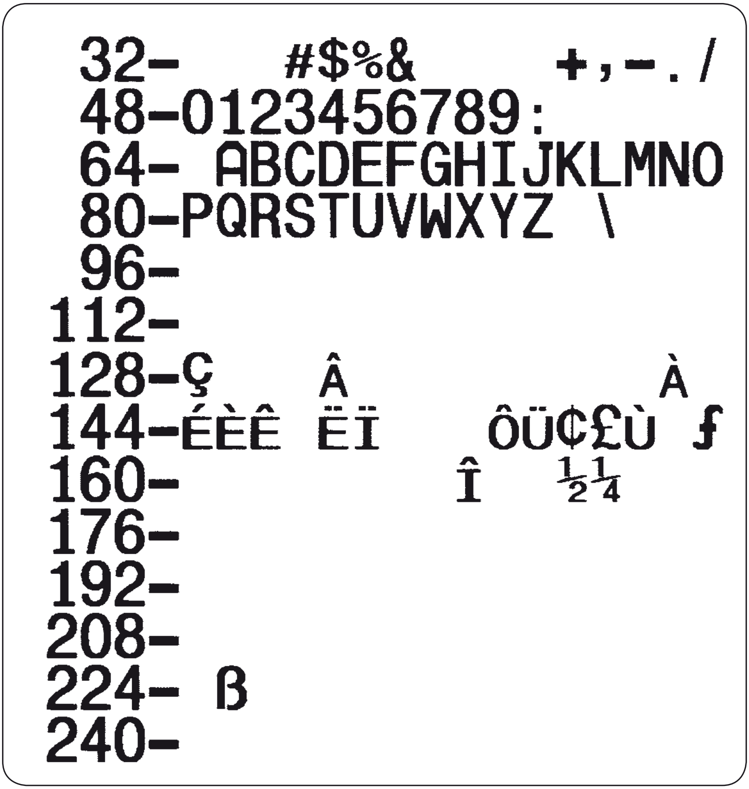
```
 32-   #$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
128-Ç             ÄÅ
144-É Æ      ÖÜ¢£  ƒ
160-     Ñ     ½¼
176-
192-
208-
224- ß
240-
```

Size 5 (8 bit); Code page 850



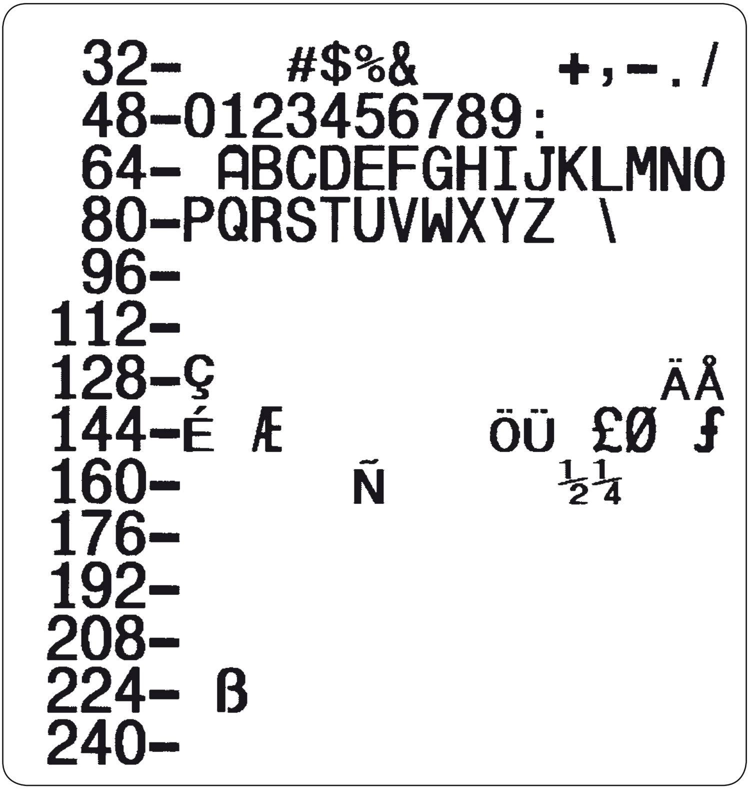
```
 32-   #$%&       +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
128-Ç             ÄÅ
144-É Æ      öÜ ££ ƒ
160-     Ñ     ½¼
176-     ÁÂÀ     ¢
192-       Ã
208-  ÊËÈ ÍÎÏ     Ì
224-óßÔÒ õ   Ú Ù
240-
```

Size 5 (8 bit); Code page 863



```
 32-   #$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
128-Ç   Â         À
144-ÉÈÊ ËÏ   ÔÜ¢£Ù ƒ
160-        Î  ½¼
176-
192-
208-
224- ß
240-
```

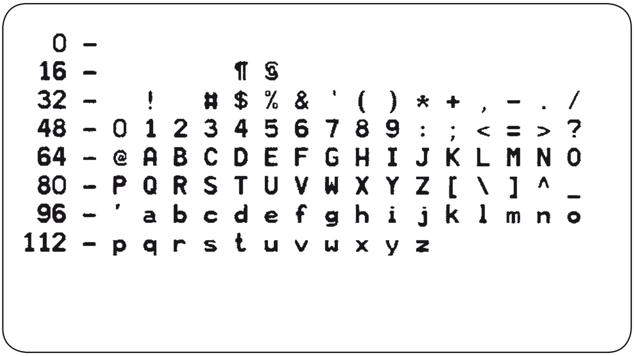
Size 5 (8 bit); Code page 865



```
 32-   #$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
128-Ç             ÄÅ
144-É Æ      ÖÜ £Ø ƒ
160-     Ñ     ½¼
176-
192-
208-
224- ß
240-
```

Size 1-4 (7 bit); USA

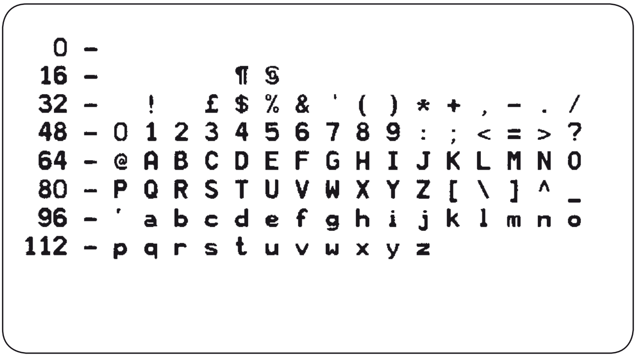
(printed in size 4)



```
  0 -
 16 -         ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
```

Size 1-4 (7 bit); British

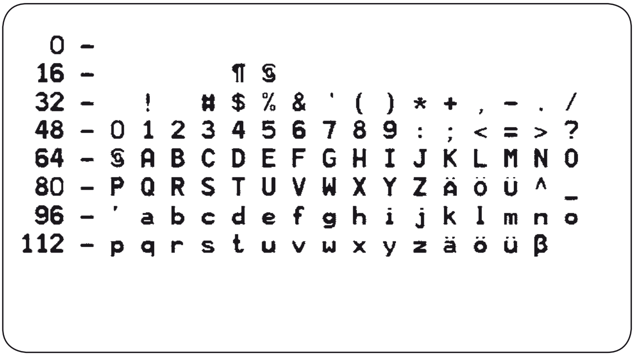
(printed in size 4)



```
  0 -
 16 -         ¶ §
 32 -   !   £ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
```

Size 1-4 (7 bit); German

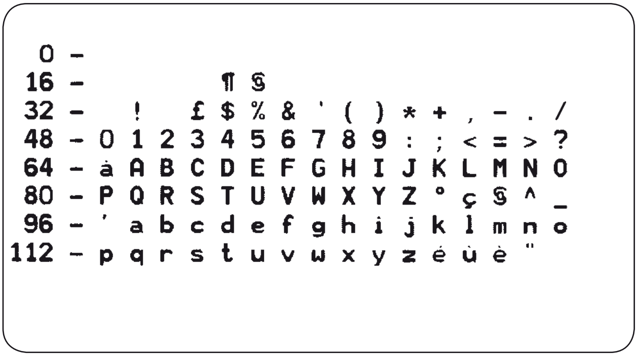
(printed in size 4)



```
  0 -
 16 -         ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - § A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Ä Ö Ü ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ä ö ü β
```

Size 1-4 (7 bit); French

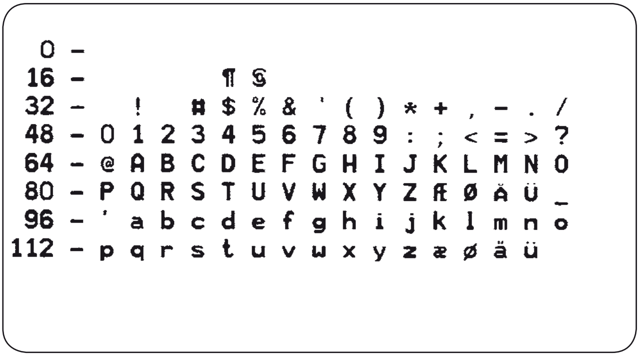
(printed in size 4)



```
  0 -
 16 -         ¶ §
 32 -   !   £ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - à A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z ° ç § ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z é ù è ¨
```

Size 1-4 (7 bit); Danish

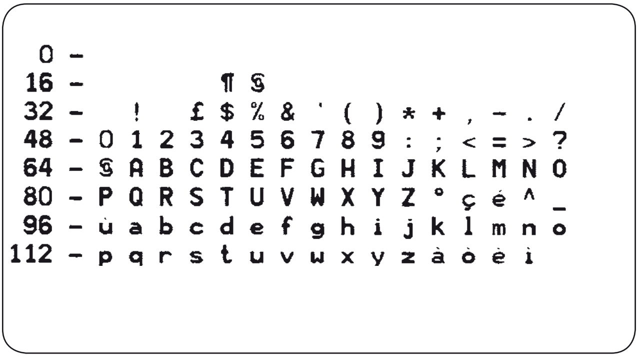
(printed in size 4)



```
  0 -
 16 -         ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Æ Ø Ä Ü _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z æ ø ä ü
```

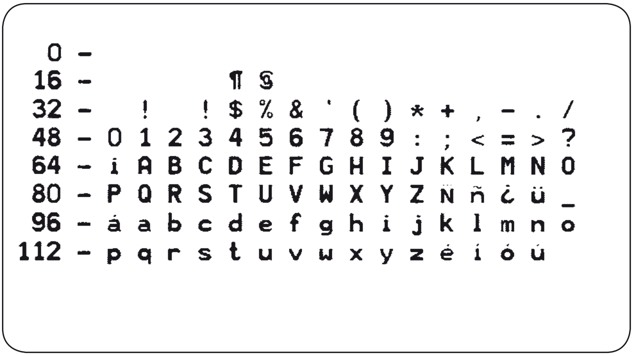
Size 1-4 (7 bit); Italian

(printed in size 4)



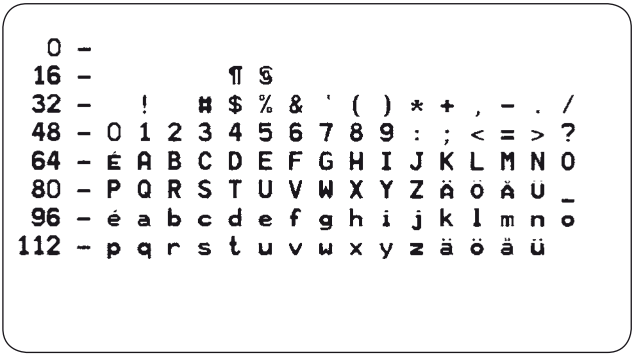
```
  0 -
 16 -         ¶ §
 32 -   !   £ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - § A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z ° ç é ^ _
 96 - ù a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z à ò è ì
```

Size 1-4 (7 bit); Spanish
(printed in size 4)



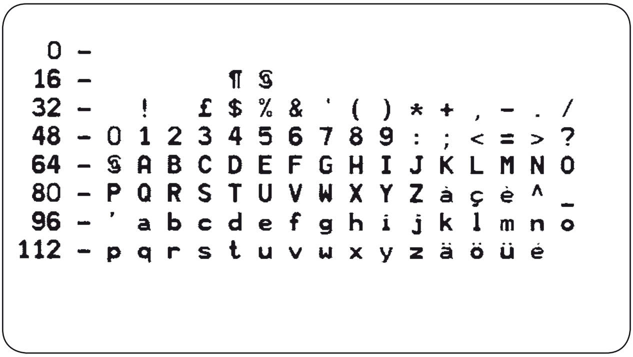
```
  0 -
 16 -                 ¶ §
 32 -     !     ! $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - ¡ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Ñ ñ ¿ ü _
 96 - á a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z é í ó ú
```

Size 1-4 (7 bit); Swedish
(printed in size 4)



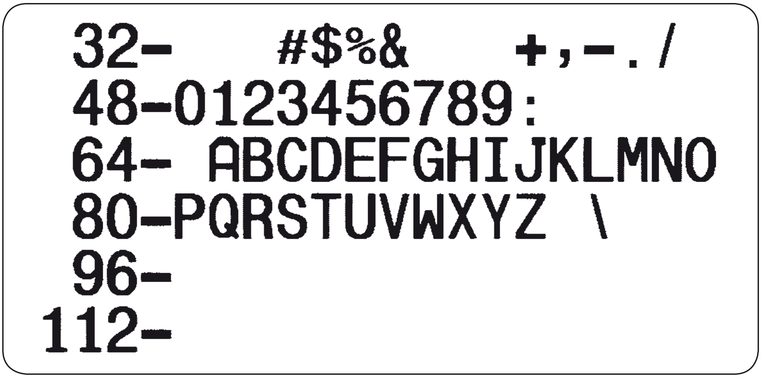
```
  0 -
 16 -                 ¶ §
 32 -     !     ¤ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - É A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Ä Ö Å Ü _
 96 - é a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ä ö å ü
```

Size 1-4 (7 bit); Swiss
(printed in size 4)



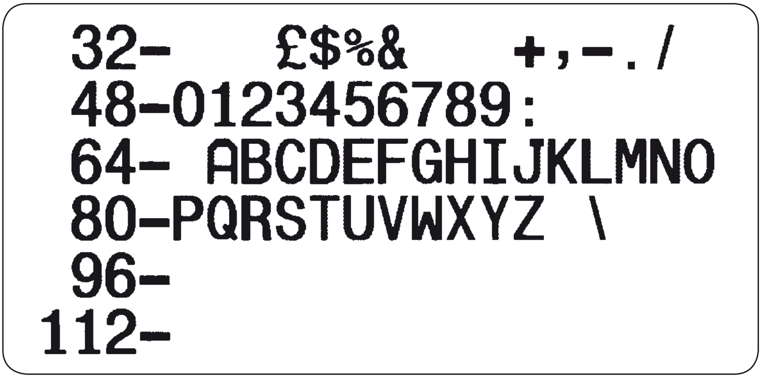
```
  0 -
 16 -                 ¶ §
 32 -     !     £ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - § A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z à ç è ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ä ö ü é
```

Size 5 (7 bit); USA



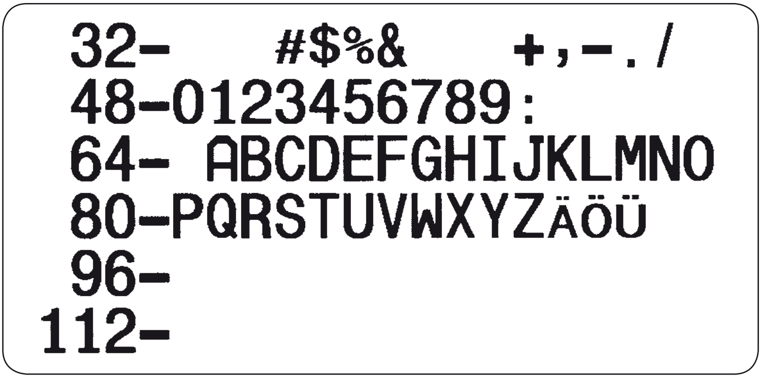
```
 32-   #$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
```

Size 5 (7 bit); British



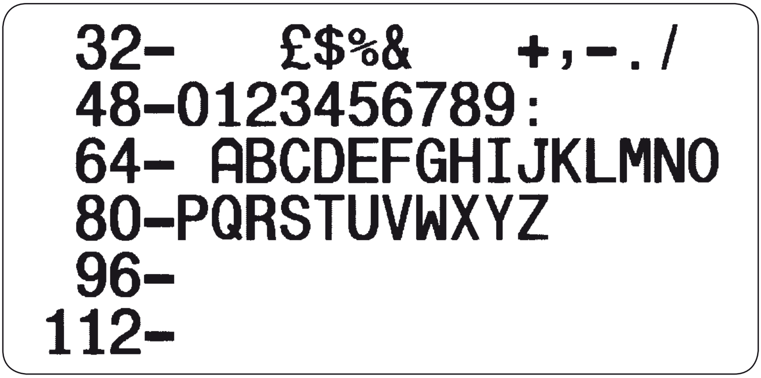
```
 32-   £$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ \
 96-
112-
```

Size 5 (7 bit); German



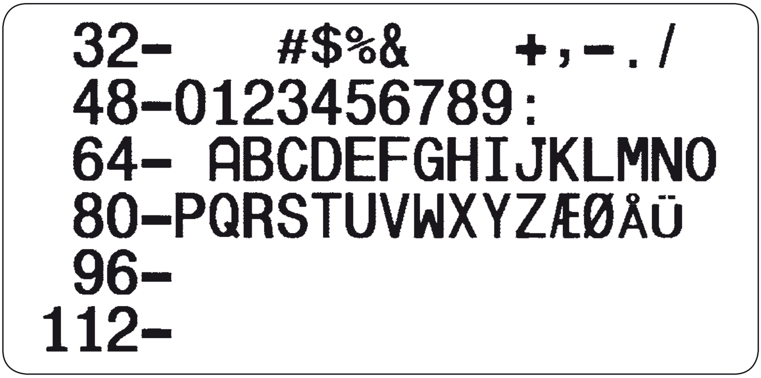
```
 32-   #$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÄÖÜ
 96-
112-
```

Size 5 (7 bit); French



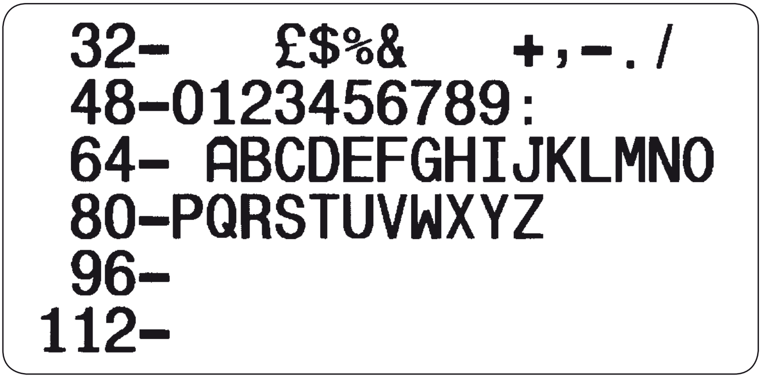
```
 32-   £$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ
 96-
112-
```

Size 5 (7 bit); Danish



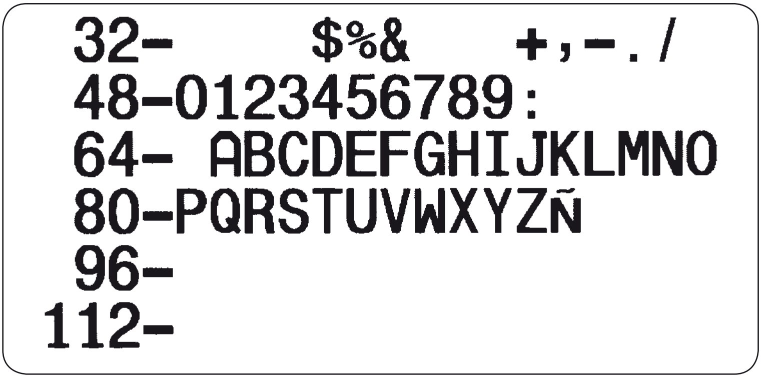
```
 32-   #$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÆØÅÜ
 96-
112-
```

Size 5 (7 bit); Italian



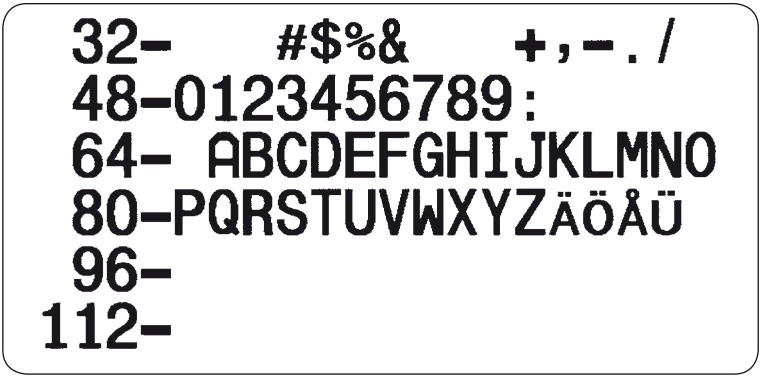
```
 32-   £$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ
 96-
112-
```

Size 5 (7 bit); Spanish



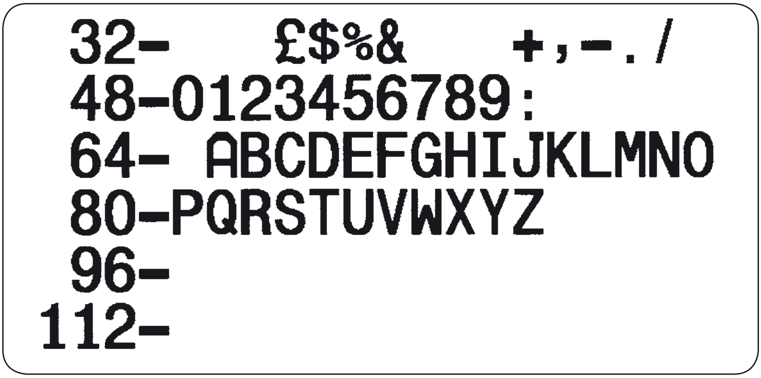
```
 32-    $%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÑ
 96-
112-
```

Size 5 (7 bit); Swedish



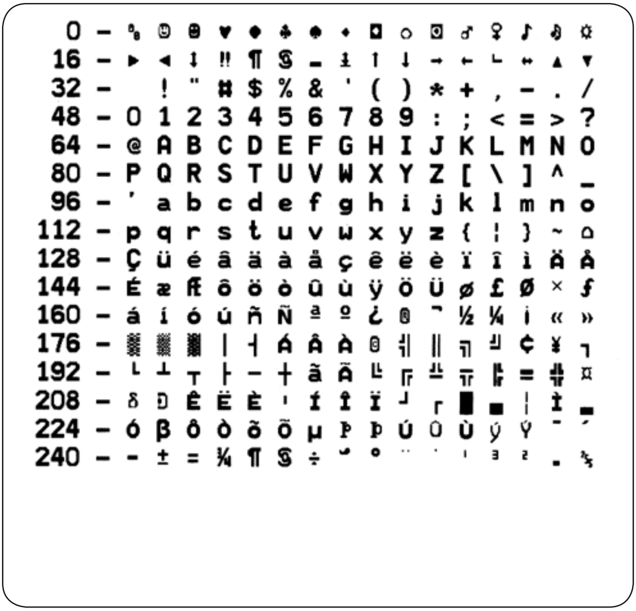
```
 32-   #$%&    +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZÄÖÅÜ
 96-
112-
```

Size 5 (7 bit); Swiss



```
 32-   £$%&   +,-./
 48-0123456789:
 64- ABCDEFGHIJKLMNO
 80-PQRSTUVWXYZ
 96-
112-
```

Size 4 (8 bit); Characters in Dump Mode



```
  0 -   ☺ ☻ ♥ ♦ ♣ ♠ • ◘ ○ ◙ ♂ ♀ ♪ ♫ ☼
 16 - ► ◄ ↕ ‼ ¶ § ▬ ↨ ↑ ↓ → ← ∟ ↔ ▲ ▼
 32 -   ! " # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z { ¦ } ~ ⌂
128 - Ç ü é â ä à å ç ê ë è ï î ì Ä Å
144 - É æ Æ ô ö ò û ù ÿ Ö Ü ø £ Ø × ƒ
160 - á í ó ú ñ Ñ ª º ¿ ® ¬ ½ ¼ ¡ « »
176 - ░ ▒ ▓ │ ┤ Á Â À © ╣ ║ ╗ ╝ ¢ ¥ ┐
192 - └ ┴ ┬ ├ ─ ┼ ã Ã ╚ ╔ ╩ ╦ ╠ ═ ╬ ¤
208 - ð Ð Ê Ë È ı Í Î Ï ┘ ┌ █ ▄ ¦ Ì ▄
224 - Ó ß Ô Ò õ Õ µ þ Þ Ú Û Ù ý Ý ¯ ´
240 - - ± = ¾ ¶ § ÷ ¸ ° ¨ · ¹ ³ ² ■
```

A Error-Handling

This appendix explains the principles of error handling and lists the various error types and all possible error replies that the printer can return to the host via an RS-232 or USB channel. It also shows how the LED control lamps will switch between various colors to notify the operator of an error condition. If the printer is fitted with an LCD display, error messages and prompts in plain text will also be shown.

Error-Handling Principles

The firmware is able to handle a number of error conditions. Code numbers represent the different error cases.

The printer controls output of the error code and its format and wait until the error recovery is performed for each special error-code.

Error action source

A: Command errors

The biggest group of error codes is command format and data information error. It exists errors of type "Syntax error" to more sophisticated like "Form or image name duplicate." (See error type A.)

B: Printing error

Print out error occurs, when the printing the label not is correctly performed, such as "Out of media", "printhead lifted", and "cutting not completed." (See error type B, C, and D.)

C: Status

A status error performs normally not a true error case, but in some matter it has to be detected. A typical case is "LTS-detection waiting." (See error type E.)

D: Special modes

Some special feature errors will hang the printer for ever or have a complete other output. Typical of this group errors are: "Flashing not completed" or "Dump mode." (See error type F and G.)

Error action output information

The error code will often be written in the upper left corner of the label, like "ERR01." The Error LED will go red and the display (if any) shows information on the error.

Serial Port (RS-232 or USB)

Output on serial line when error occurs is depending on some setup commands.

The commands **UN** and **US** turn off or on most of the error output information.

The command **eR** specifies the format of the error output information.

Parallel Port

At an error, the parallel port will mark error status by setting pin 12 in active state.

Error recovery/output

The normal action at any error is to stop in a waiting loop to recover the error feature. At this "Error Waiting Loop" only command **^ee** and **^@** can be performed and executed.

The error recovery, often performed by pressing the FEED key, releases the error and sends recovery information on the serial line (RS-232 or USB), normally <XON>.

See also commands **USA** or **USB** for direct recovery.

Error-handling commands

US[A][B][E]	Optional Error Reporting
UN	No Error Reporting
eR	User Defined Error/Status Character Control
^ee	Error Report - Immediate
^@	Reset

Error Types

Classified types of error & status codes for the Error Replies table later in this appendix.

Type	Feature	Generates	Recovered by
A	Minor Error of command sequences and similar.	At Error reporting (cmd **US**): Sets error on print command and sets Error LED to solid red.	At Error reporting (cmd **US**): Press Feed key orsend Reset command **^@**. Also see cmd **USA**.
		At no Error reporting (cmd **UN**): sets Ready LED to green at print command	At No Error reporting (cmd **UN**): No recover necessary, already done.
B	Print Stopping Error.	Sets Error LED to solid red	Feed Key or command Reset **^@**.See also command **USB**.
C	Paper End Stopping Error.	Sets Error LED to solid red at print.	Load new media and press Feed key or send Reset command **^@**.
D	Error of printing functionality and status.	Sets Error LED directly to red, sometimes blinking.	Manual or automatic (temperature lowered) measure or send Reset command **^@**.
E	Status changes.	No LED change.	No recover necessary/Delete Label from LTS. See also command **USE**.
F	Unrecoverable Error.	Sets Error LED directly to red.	Power Off/Power On or send Reset command **^@**.
G	Various modes		Not applicable

Error Replies

Explanation of text in the table on the next page:

Error	Referred error number divided in Code and Type.
Code	Referred error number
Type	Letter is a classifying of error.
Description	Short human explanation
Error Response...	Response depending on Error reporting flag controlled by US & UN.
LED	Designation and color
	Ready Green (GRN)
	Error Red (RED)
	Error Blinking Red (BLNK RED)
Serial	Output on Serial line (RS-232 or USB) at error response.
Response at ^ee	Output on serial line of error number, if only one error is available.
<LF>	0x0a
<CR>	0x0d
<CRLF>	0x0d + 0x0a
<XON>	0x11
<XOFF>	0x13
<NAK>	0x15
-	Nothing happens
[XOFF]	Flag for waiting on Error recovery. Default value is <XOFF> = 0x13 , but can be substituted with command eRp1,6,p_3 where p_1 is replacement of <XOFF> and where p_3 is replacement of <XON>.

Example Command **`eRF,6,N`** substitutes at error reply <XOFF> with the character ‘F’ and at error recovery <XON> with the character ‘N’.

Error Replies Table

Error		Displayed Error Message or *(Mode Status)*	Error Response depending on cmd US/UN				Response at cmd ^ee
Code	Type		command US: reporting on		command UN: reporting off		
			LED	Serial	LED	Serial	
00		No Error	GRN	-	GRN	-	00<CRLF>
01	A	Syntax error	RED	<NAK>01[XOFF]	GRN	-	01<CRLF>
02	B	Object exceeds image buffer border	RED	<NAK>02[XOFF]	RED	[XOFF]	02<CRLF>
03	A	Data length error	RED	<NAK>03[XOFF]	GRN	-	03<CRLF>
04	A	Insufficient memory to store Data	RED	<NAK>04[XOFF]	GRN	-	04<CRLF>
05	A	Memory configuration error	RED	<NAK>05[XOFF]	GRN	-	05<CRLF>
7	C	Out of media	RED	<NAK>07Pnnn[XOFF] (nnn = numbers of remaining labels)	RED	[XOFF]	07<CRLF>
08	A	Form or image name duplicate	RED	<NAK>08[XOFF]	GRN	-	08<CRLF>
09	A	Form or image not found	RED	<NAK>09[XOFF]	GRN	-	09<CRLF>
11	D	Print head lifted	RED	<NAK>11[XOFF]	RED	[XOFF]	11<CRLF>
13	E	Label taken sensor waiting	-	<NAK>13[XOFF] (if command USE has been preceded)	-	-	13<CRLF>
16	A	No form was retrieved before “?“	RED	<NAK>16[XOFF]	GRN	-	16<CRLF>
17	C	Out of ribbon	RED	<NAK>17Rnnn[XOFF] (nnn = numbers of remaining labels)	RED	[XOFF]	17<CRLF>
50	A	Does not fit in area specified	RED	<NAK>50[XOFF]	GRN	-	50<CRLF>
51	A	Data length too long	RED	<NAK>51[XOFF]	GRN	-	51<CRLF>
62	D	High print head temperature (WAIT)	BLNK RED	<NAK>62[XOFF]	BLNK RED	[XOFF]	62<CRLF>
71	F	Default Setup retrieved (RESET)	GRN	-	GRN	[XOFF]	71<CRLF>
72	F	Flashing not completed (RESET)	RED	<NAK>72[XOFF]	RED	[XOFF]	72<CRLF>
73	F	Download error (RESET)	RED	<NAK>73[XOFF]	RED	[XOFF]	73<CRLF>
81	B	Cutter jammed or not installed	BLNK RED	<NAK>81[XOFF]	BLNK RED	[XOFF]	81<CRLF>
89	G	*(Dump Mode)*	GRN	<NAK>89 XOFF]	GRN	-	89<CRLF>

90	G	*(Printing)*	GRN	<NAK>90 [XOFF]	GRN	[XOFF]	90<CRLF>
91	G	*(Setup)*	GRN	<NAK>91 [XOFF]	GRN	[XOFF]	91<CRLF>
92	G	*(Pause Printing Mode)*	GRN	<NAK>92 [XOFF]	GRN	[XOFF]	92<CRLF>
93	G	*(Pause Direct Mode)*	GRN	<NAK>93 [XOFF]	GRN	[XOFF]	93<CRLF>
94	G	*(Autosensing Mode)*	GRN	<NAK>94 [XOFF]	GRN	[XOFF]	94<CRLF>
98	G	*(Download Mode)*	GRN	<NAK><ACK>	GRN	<NAK><ACK>	98<CRLF>

Intermec Technologies Corporation
Corporate Headquarters
6001 36th Avenue West
Everett, WA 98203
U.S.A.

tel 425.348.2600

fax 425.355.9551

www.intermec.com

Intermec ESim v5.12 – Programmer's Reference Manual

P/N 1-960615-01