Programmer's Guide

P/N 1-960413-01
Edition 2
September,1998

EasyCoder 91
Bar Code Label Printer

Intermec

A UNOVA Company

Contents

cont'd

Intermec EasyCoder 91
Programming Manual
Edition 2, September 1998
Part No. 1-960413-01

Contents, cont'd.

Introduction

The *EasyCoder 91* printers from *Intermec* are provided with a built-in protocol 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 *Windows Driver*, which requires a PC operating under *Microsoft Windows*.

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

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

The manual describes version 2.24 of the protocol.

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

Information in this manual is subject to change without prior notice and does not represent a commitment on the part of Intermec Printer AB.

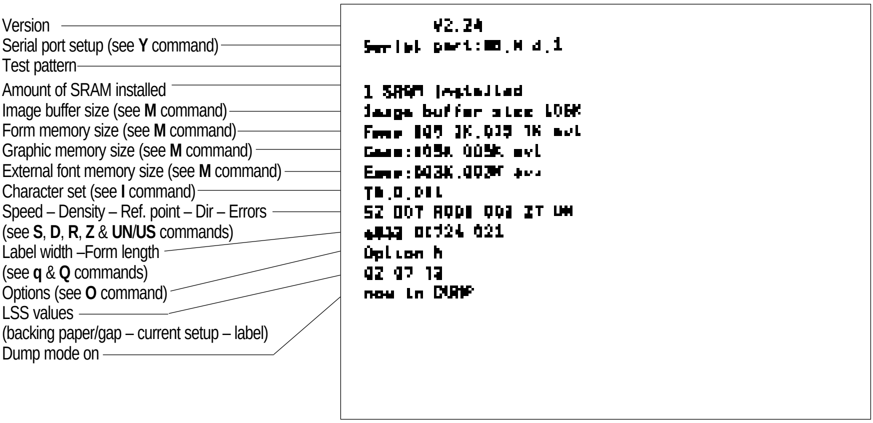
General Information

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:

- Turn 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.
- Hold the **Feed** key until the printer starts to feed. Then immediately release the **Feed** key. This procedure also adjusts the label stop sensor and produces a test label, see below.


Version
Serial port setup (see **Y** command)
Test pattern
Amount of SRAM installed
Image buffer size (see **M** command)
Form memory size (see **M** command)
Graphic memory size (see **M** command)
External font memory size (see **M** command)
Character set (see **I** command)
Speed – Density – Ref. point – Dir – Errors (see **S**, **D**, **R**, **Z** & **UN**/**US** commands)
Label width –Form length (see **q** & **Q** commands)
Options (see **O** command)
LSS values (backing paper/gap – current setup – label)
Dump mode on

```
        V2.24
Serial port:[illegible].N d.1

1 SRAM installed
Image buffer size 100K
Fmem 105 3K.035 1K avl
Gmem:105K 005K avl
Emem:043K.003K [illegible]
Tb.0.DIL
S2 D0T A001 001 ZT UN
[illegible] 0C724 021
Option h
02 07 13
now in DUMP
```

The Test Label contains useful information on the printer's current setup.

Note: If a real-time clock circuit is fitted in an inserted optional memory cartridge, the present time and date according to the clock circuit will also be printed on the Test Label.

Dump Mode, cont'd.

You can also enter the Dump Mode, when an error occurs and the control lamp shines orange, by pressing the **Feed** key and keep it depressed a few seconds (as opposed to tapping the key, which just resets the printer).

In the Dump Mode, the output is the same label as produced by means of a **U** command, but an extra line will be appended saying *"now in DUMP"*. Then the printer waits for ASCII dump printing.

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 page 108.

To return to normal mode, press the **Feed** key. A label with the message *"out of DUMP"* will be printed.

Memory

The firmware has memory allocation for print image buffer, form, graphic, and external font (soft font) memory. The first time the printer is used, it is automatically initialized to default settings, see page 12.

The **M** memory command sets the image buffer, the form memory, and graphic memory area. The remaining memory space, if any, is allocated to the external font memory, which is presently not used.

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.

The optional Keyboard Display Unit (KDU) can retrieve a stored form, making it possible to use the *printer* as a stand alone system, i.e. without connection to a computer.

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 or serial port and a text editor to design the form and programming the printer. Communication is based on the 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.

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).

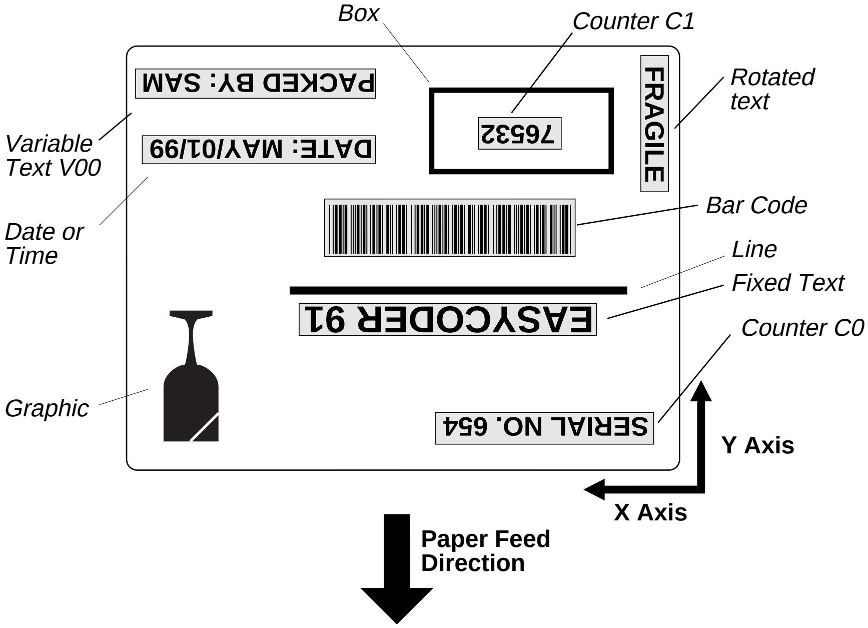
Note: Line Feed (LF) is required to be sent at the end of most command lines!

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 page 9 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.



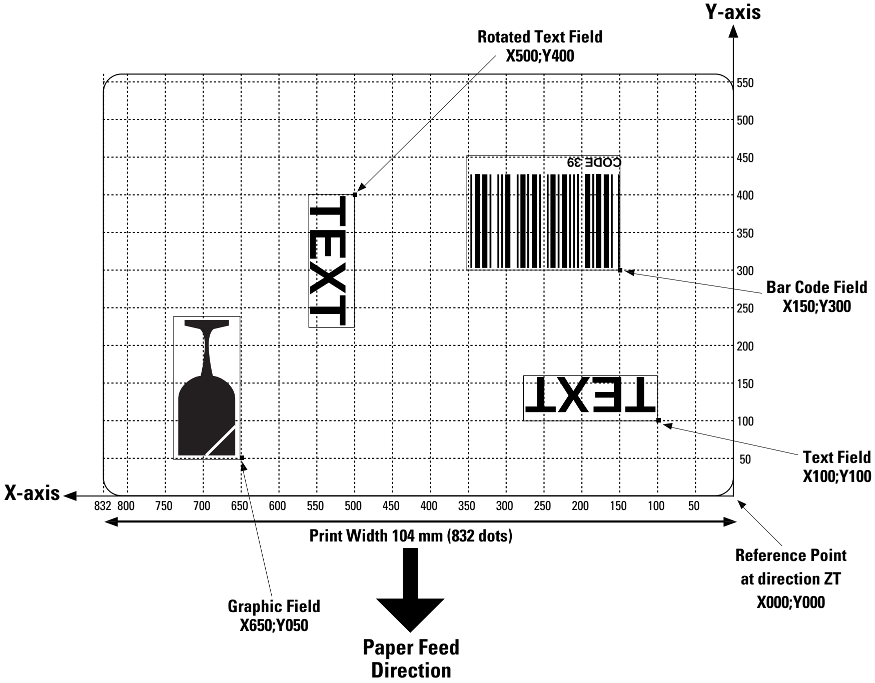
The illustration shows how a label is printed and fed out when using the default direction.

Field Positioning

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

For example, as the printhead density is eight dots per millimetre (203 dots per inch), a field that starts 5 mm (0.197 in.) inside of the left margin and 3 mm (0.118 in.) down should be expressed as 40 dots on the X axis and 24 dots on the Y axis.

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).



Commands List

The following list illustrates which commands can be used in the Direct Mode and the Form Edit Mode and for what purposes.

Direct Mode

- **Setup Commands**

  Used to set up the printer globally, i.e. affects both the Direct Mode and Forms.

D	Density	40
I	Character Set Selection	52
JB	Disable Top of Form Backup	53
JF	Enable Top of Form Backup	54
j	Paper Feed Adjustment	55
M	Memory Allocation	60
O	Options Select	66
Q	Set Form Length	70
q	Set Label Width	74
R	Set Reference Point	75
S	Speed Select	76
TD	Define Date Layout	77
TS	Set Real Time Clock	78
TT	Define Time Layout	79
UN	Disable Error Reporting	85
US	Enable Error Reporting	87
W	Windows Mode	90
Y	Serial Port Setup	92
Z	Print Direction	93

- **Store Command**

  Used to store graphic files in printer's memory.

GM	Store Graphics	51

- **Clear and Delete Commands**

  Used to erase data from the printer's memory.

FK	Delete Form	44
GK	Delete Graphics	50

Direct Mode, cont'd.

N	Clear Image Buffer	65

- **Editing Commands**

  Used to edit labels in the Direct Mode.

A	Print Text	24
B	Print Standard Bar Codes	28
b	Print Two-Dimensional Codes	33
GG	Print Graphics	48
LE	Line Draw Exclusive	56
LO	Line Draw Black	57
LS	Line Draw Diagonal	58
LW	Line Draw White	59
X	Draw Box	91

- **Print Commands**

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

FR	Retrieve Form	45
?	Download variables	94
P	Print	68

- **Report Commands**

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

FI	Print Form Information	43
GI	Print Graphics Information	49
U	Print Configuration	80
UF	Form Information Inquiry	81
UG	Graphics Information Inquiry	82
UI	Enable Prompts/Codepage Inquiry	83

Form Edit Mode

UM	Codepage & Memory Inquiry	84
UP	Codepage & Memory Inquiry/Print	86

- **Setup Commands in Forms**

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

D	Density	40
Q	Set Form Length	70
R	Set Reference Point	75
S	Speed Select	76
TD	Define Date Layout	77
TT	Define Time Layout	79
Z	Print Direction	93

- **Editing Commands**

  Used to edit forms.

A	Print Text	24
B	Print Standard Bar Codes	28
b	Print Two-Dimensional Codes	33
C	Counter	37
FE	End Form Store	42
FS	Form Store	46
GG	Print Graphics	48
LE	Line Draw Exclusive	56
LO	Line Draw Black	57
LS	Line Draw Diagonal	58
LW	Line Draw White	59
PA	Print Automatic	69
V	Define Variable	88
X	Draw Box	91

Setting Up the Printer

Default Setup

At delivery, the printers are set up as follows.

Parameter	Command	EC 91/200 dpi	EC 91/300 dpi	Notes
Density	**D**	7	4	
Character Set	**I**	8 bits, code page 0, country code 001		
Top of Form backup	**JB/JF**	Enabled	Enabled	
Paper feed adj.	**j**	140 dots (tear-off)	214 dots (tear-off)	
Image buffer	**M**	106 Kbytes	188 Kbytes	
Form memory	**M**	5.1 Kbytes	5.1 Kbytes	
Graphic Memory	**M**	5 Kbytes	5 Kbytes	
Ext. Font Memory	**M**	3 Kbytes	52 Kbytes	
Label Stop Sensor	**O**	Normal	Normal	
Label Taken Sensor	**O**	Enabled (if fitted)	Enabled (if fitted)	
DT/TT Printing	**O**	TT	TT	DT in pure DT printers
Form Length	**Q**	Auto-detection	Auto-detection	
Label Width	**q**	832 dots (full width)	1248 dots (full width)	
Reference Point	**R**	X:024;Y000	X:024;Y000	
Print Speed	**S**	50 mm/sec (2"/sec.)	37 mm/sec (1.5"/sec.)	
Error Handling	**UN/US**	Disabled	Disabled	
Windows Mode	**W**	Disabled	Disabled	
Serial Port	**Y**	9600 baud, no parity, 8 data bits, 1 stop bit		
Print Direction	**Z**	ZT	ZT	Start at top of form

The setup will be reset to default values if:

- The memory backup battery is disconnected or discharged.
- An optional memory cartridge has been removed.
- The printer's memory is reformatted using an **M** command.

Some commands may also affect the values of other command, e.g. 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 reset to default.

Example

Let us assume that we will use an *EasyCoder 91* 200 dpi thermal transfer printer (peel-off model) without any memory cartridge. We will print full vellum labels with GP91 transfer ribbon 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 7 to 4
- Paper feed adjustment from 140 to 110
- Label Taken Sensor from enabled to disabled

Enter the following commands:

Command	Explanation
↵	CR/LF to start command structure
D4 ↵	Set density
j110 ↵	Set paper feed adjustment for peel-off operation
ON ↵	Disable label taken sensor

Editing in Direct Mode

Example

Assuming that...

- The printer has been set up for the application (see page 12)
- The length of the label and the gap has been determined by printing a Test Label (see page 4)
- The graphic used in the example has been downloaded to the printer as described on page 51 (GM command[1]).

...we will now print two copies of a label which we will edit in the Direct Mode.

This means that 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

The label will look like the example on page 15.

[1]/. *The Intermec logotype is not included in the firmware 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.*

Editing in Form Edit Mode

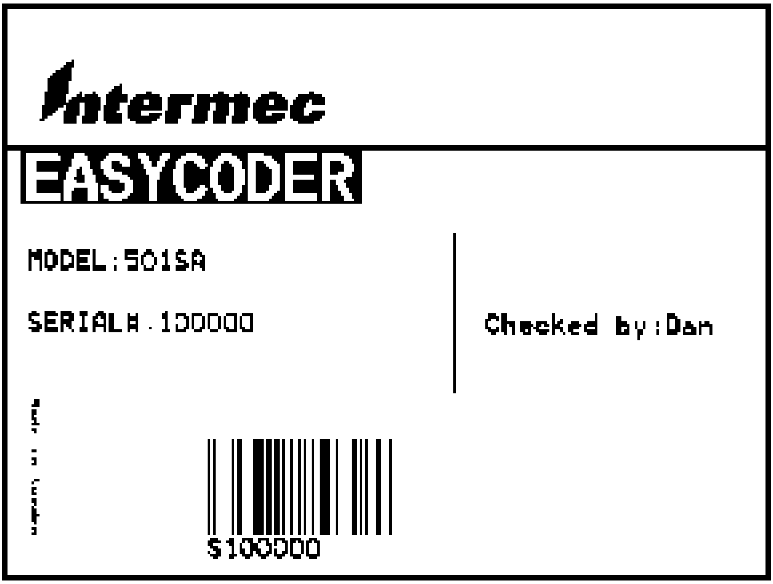
Example

Assuming that...

- The printer has been set up for the application (see page 12)
- The length of the label and the gap has been determined by printing a Test Label (see page 4)
- The graphic (i.e. the Intermec logotype) used in the example has been downloaded to the printer as described on page 51 (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:



Intermec

EASYCODER

MODEL:501SA

SERIAL#.100000

Checked by:Dan

$100000

*[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.*

Example, cont'd.

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 (**V**00, **V01,** **V**02) and counter (**C**0), are defined within this label form named TEST. The next label form containing variables and counters, will again start with **V**00 and **C**0.*

If variable data is being sent from an external data base, omit the text between the quotes and replace with a space character, e.g. ***V**00,15,N," "*.

Example, cont'd.

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 deg. 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 **V**00. 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

Example, cont'd.

Place a Combination of Fixed Data and a Counter

The next command line is a text line containing fixed data and the counter (**C**0). 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#: "C 0↵	Text line, fixed data + 1:st counter

Place a Bar Code with Fixed Data and a Counter

Below Bar Code Command line is entering 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"C 0↵	Bar code, fixed data + 1:st counter

Place Graphics

The next line writes a graphic named "Intermec" from graphics 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

On next page, there is a complete list of this example.

Example, cont'd.

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#: "C 0↵	Write text line, fixed data + 1:st counter
B280,440,0,1,2,3,96,B,"S"C 0↵	Write barcode, fixed data + 1:st counter
GG24,12,"LOGO"↵	Write graphic from graphics memory
FE↵	Closing command to store form

Retrieving and Printing a Form

Example

Retrieve and Print Form

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.

Example,
cont'd.

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.

Printer Sends	**Command**	**Explanation**
	↵	CR/LF to start command structure
	UI	Enable prompts command (optional)
UI80,001		Printer returns codepage status
	FR"TEST"↵	Retrieve form
	?↵	Call for variables
Enter Product name:	EASYCODER↵	Substitute variable V00[1]
Enter Model number:	501SA↵	Substitute variable V01
Checked by:	Dan↵	Substitute variable V02
Enter SERIAL#:		
100001	100000↵	Reset ,accept or enter[2]
		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 Quantity of copies + ↵

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

Example, cont'd.

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 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 memory. 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***

IMPORTANT!
*Note that the question mark (**?**) following the **FR** command is essential for the printing of certain fields edited in the Form Edit Mode, i.e. fields containing variables, counters, time and/or date. Variables and counter start values must be entered or accepted as described above, whereas time and date will be read from the optional real-time clock circuit. If no question mark is transmitted, all fields containing variable input, i.e. variables, counters, time and date, will be completely omitted from the printout.*

Commands

Introduction

This chapter lists the various commands in alphabetical 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 Descriptions

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

- **$p_1 – p_n$**
  Indicates parameters listed separately below the syntax description.
- **[** ..... **]**
  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.), i.e. "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 software program uses " " (ASCII 34 dec.), you need a special designator if you need to print text or bar code which includes 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.

A – Print Text

Description

This command is used to print an ASCII text string.

Syntax

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[1]*

p_3
0 *No Rotation*
1 *90 degrees rotation clockwise*
2 *180 degrees rotation clockwise*
3 *270 degrees rotation clockwise*

p_4 *Font selection[2]:*

	8 dots/mm (203.2 dpi)	***11.81 dots/mm (300 dpi)***
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)*	*18.75 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)*

p_5 *Horizontal multiplier 1, 2, 3, 4, 6, 8.
(Magnifies the text horizontally)*

p_6 *Vertical multiplier 1, 2, 3, 4, 5, 6, 7, 8, 9.
(Magnifies the text vertically)*

p_7
N *Normal image*
R *Reverse image*

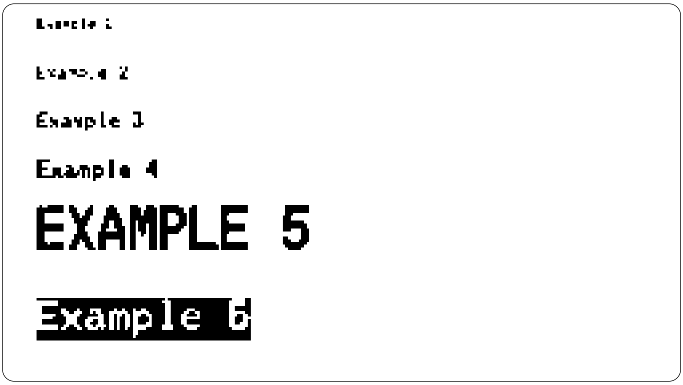
"DATA" *Represents a fixed data field.*

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 ↵
```

A – Print Text, cont'd.

Example, cont'd.



Example 1

Example 2

Example 3

Example 4

EXAMPLE 5

Example 6

Note:
Font size 5 only supports uppercase characters, as illustrated by example 5 above.

Remarks

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

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 web, it is done by using a single counter in a single form.

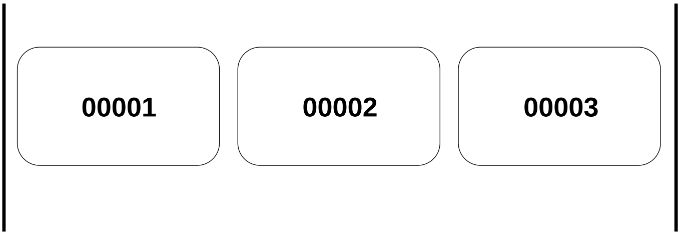
The command **Cn±x** in our example will be used twice and count up the single counter by one (1) in each position (last two A-command lines).

Set the Form Step Value $\mathbf{p}_4$ to +3 for the counter **Cn** used in our example (see the C-command line). Also refer to “*C Command – Counter”*.

A – Print Text, cont'd.

Remarks, cont'd.

```
↵
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 predefined format. See the TT command for format selection. This command is only available if a Memory Cartridge with 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 Memory Cartridge with a Real Time Clock is installed.

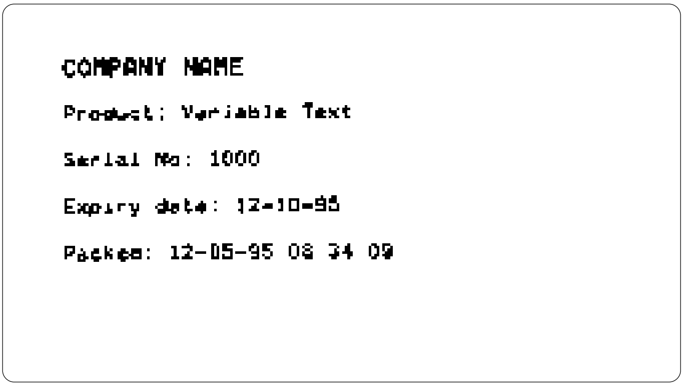
TD+nn Prints “sell by” date. Adds **nn** number of days (must be two digits) to the current date and places it on the form using date layout defined.

A – Print Text, cont'd.

Remarks, cont'd. *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 ↵
```

After retrieving and printing the form, the label may e.g. look like this:



COMPANY NAME

Product: Variable Text

Serial No: 1000

Expiry date: 12-10-95

Packed: 12-05-95 08 34 09

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

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 *Barcode select. See table below.*

p_5 *Narrow bar width in dots. See table below.*

Barcode 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 128UCC 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	*E32*	*2–4*
EAN13 5 digit add-on	*E35*	*2–4*
German Postcode	*2G*	*1-10*
Interleaved 2 of 5	*2*	*1–10*
Interleaved 2 of 5 with check digit	*2C*	*1–10*
Interleaved 2 of 5 with 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 Interleaved 2 of 5	*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*

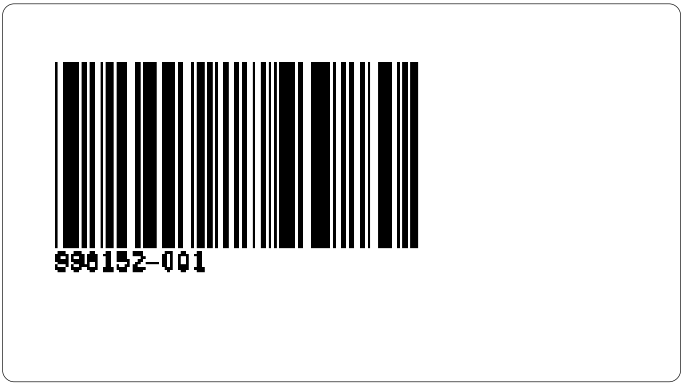
"DATA" *Represents a fixed data field*

B – Standard Bar Codes, cont'd.

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 below commands:

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.

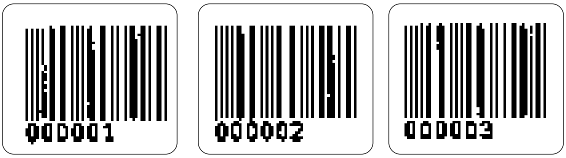
Example:

When labels with consecutive numbers are printed next to each other across the web, 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 B-command lines).

B – Standard Bar Codes, cont'd.

Remarks, cont'd. Set the Form Step Value **p_4** 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 Memory Cartridge with 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 Memory Cartridge with 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.

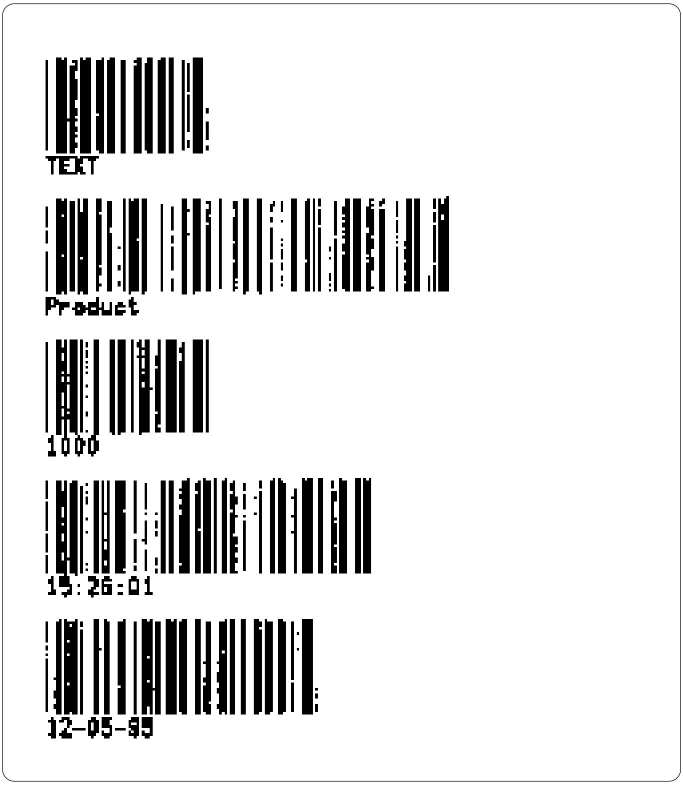
B – Standard Bar Codes, cont'd.

Example

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 ↵
```

After retrieving and printing the form, the label may look like this:



B – Standard Bar Codes, cont'd.

Example, cont'd. *Combination of several options can also be used, e.g:*

```
B50,300,0,3,1,2,50,B,"Deluxe"V01C2"Combo"TDV01TT↵
```

:Writes a Code 39 bar code containing the information "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.

b – Two-Dimensional Bar Codes, General Part

Description

This command is used to print two complex bar codes; *PDF 417* and *MaxiCode*. 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.

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:*	
	M	*Selects MaxiCode*
	P	*Selects PDF417*

[code specific options], see the following two pages

Remarks

If the amount of data will not fit in the area specified, the indicator will light orange, indicating an error.

b – MaxiCode

Description

The following *MaxiCode* specific options should append the general part of the two-dimensional code command (**b** command), see page 33.

Syntax

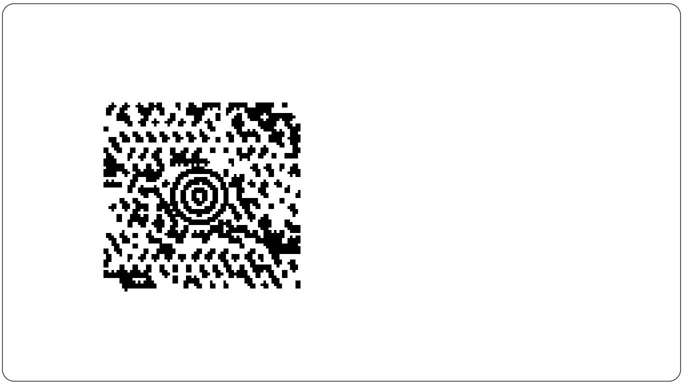
["CL,CO,PC,LPM"]

Parameters

CL	*Class Code (3 digit number)*
CO	*Country Code (3 digit number)*
PC	*Postal Code:* *U.S.A (5 digits**,**4 digits) Note the separating comma sign!* *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 PDF 417 bar code specific options should append the general part of the two-dimensional code command (**b** command), see page 33.

Syntax

```
[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 codewords as data codewords 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 *Centre 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 codewords:*
- ***0*** *Values of codewords not printed (default)*
- ***1*** *Values of codewords 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.

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

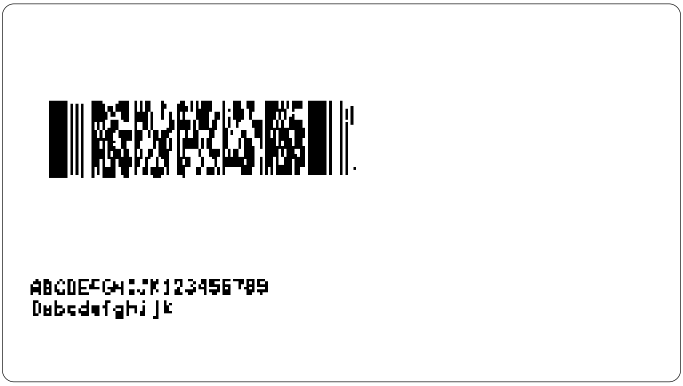
o- *Rotation:*
- ***0*** *0° rotation clockwise*
- ***1*** *90° rotation clockwise*
- ***2*** *180° rotation clockwise*
- ***3*** *270° rotation clockwise*

DATA *Represents a fixed data field.*

b – PDF417, cont'd.

Example:

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



ABCDEFGHIJK123456789
0abcdefghijk

C – Counter

Description

This command is used to define one of max. 10 automatic counters used in consecutive numbering applications (e.g. 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,"[-][- -]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*** *Centre justification*
- ***N*** *No justification*

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

- ***+*** *Incrementation*
- ***-*** *Decrementation*

[-] *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 (e.g. 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 (KDU) sends the **UI** command automatically.

C – Counter, cont'd.

Remarks, cont'd. 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 centre 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, i.e. **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.*

Example *This form lets you test field justifications by entering various start values when the form is retrieved for printing. Test various number of digits, with and without leading zeros.*

```
↵
FK"TEST5"↵
FS"TEST5"↵
C0,5,L,+1,"Start value CNT 0"↵
C1,5,R,+1,"Start value CNT 1"↵
C2,5,C,+1,"Start value CNT 2"↵
C3,5,N,+1,"Start value CNT 3"↵
A50,50,0,3,1,1,N,"Cnt left justified:"↵
A50,100,0,3,1,1,N,"Cnt right justified:"↵
A50,150,0,3,1,1,N,"Cnt centre justified:"↵
A50,200,0,3,1,1,N,"Cnt not justified: "↵
FE↵
```

C – Counter, cont'd.

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:" ↵
```

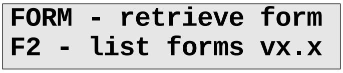
By placing two (2) minus signs as the first two characters of the prompt, the prompt will never be displayed.

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 <**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.

D – Density

Description

This command is used to select the print density.

Syntax

Dp_1

Parameters

p_1 *Density setting (0 – 15).*
0 is the lightest printing and 15 is the darkest.

Remarks

The density command is used to control the energy to the printhead. There are a number of factors that affect the actual darkness of the printout:

- Direct thermal printing or thermal transfer printing
- Print speed
- Different brands of direct thermal paper
- Different combination between transfer ribbons and receiving face materials

The printed information may also require the density to be adjusted. Typically, this applies to horizontal (picket fence) and vertical (ladder) bar codes, but text and graphics may also require adjustment. Thus, we recommend the settings listed on next page to be used initially.

D – Density, cont'd.

Type of		**Rec. Density**
Printing		**at S = 2 (50 mm/sec)**
Direct thermal printing:		
Intermec Economy		9
Intermec Premium		9
Thermal transfer printing (Europe):		
GP91 ribbon	Intermec Vellum paper	4
GP91 ribbon	Intermec Matte coated paper	4
HP91 ribbon	Intermec Matte coated paper	7
HP91 ribbon	Semi gloss paper	6
HR91 ribbon	Synthetic gloss	8
Thermal transfer printing (USA):		
GP92 ribbon	Bond paper	3
GP92 ribbon	Matte coated paper	0
HP92 ribbon	Matte coated paper	4
HP92 ribbon	Semi gloss paper	8
HR91 ribbon	Synthetic gloss	8

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

Example **D9** ↵ *:Selects density 9*

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

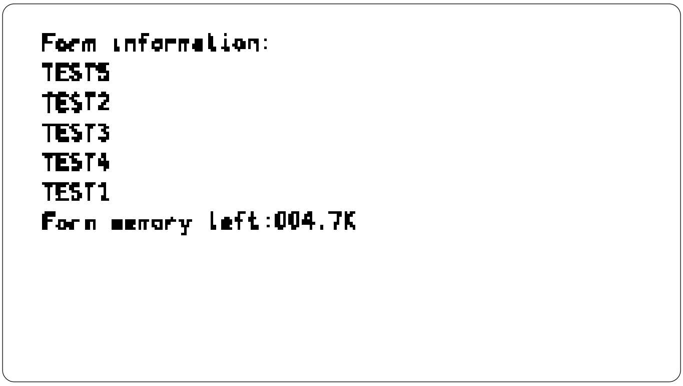
Description This command makes the printer produce a list of all forms stored in memory.

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 free form memory.*



```
Form information:
TEST5
TEST2
TEST3
TEST4
TEST1
Form memory left:004.7K
```

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, i.e. 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, i.e. 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 Forms 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.

Important!
*Always make backup copies on the host! If you need to change the memory allocation (see **M** command), or if the RAM backup batteries should run out, all formats and graphics stored in the printer and memory cartridge will be lost.*

FS – Form Store, cont'd.

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 the serial interface.

Important!
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 indicator lamp shines **green**, send XOFF or NULL to exit "AUTOFR" mode. Then delete the startup file using FK "AUTOFR"*
- *If the indicator lamp shines **orange**, there is no communication and the RAM memory must be erased by removing the backup battery on the CPU board and possibly also in the optional Memory Cartridge.*

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 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, i.e. 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.

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 memory.

Syntax

```
GI
```

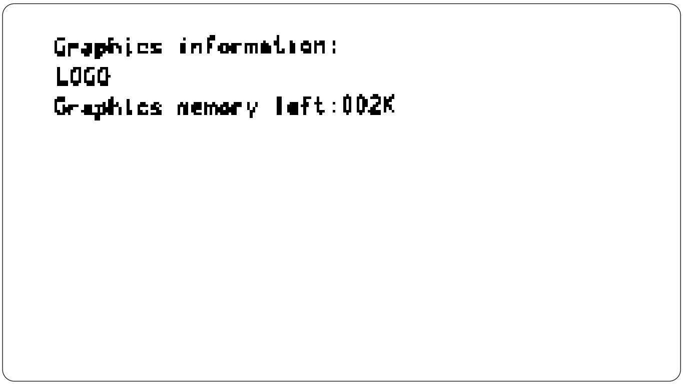
Remarks

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*



```
Graphics information:
LOGO
Graphics memory left:002K
```

GK – Delete Graphics

Description

This command is used to delete a specified graphic or all graphics from 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 memory*

Examples

`GK"LOGO" ↵`	*:Deletes "LOGO"*
`GK"*" ↵`	*:Deletes all graphics*

GM – Store Graphics

Description

This command is used to store PCX graphics files in memory.

Syntax

GM"name"p_1 ↵
"DATA"

Parameters

"name" *This is the graphic name that will be used when retrieving the stored graphic. The name may be from 1 to 8 characters. The printer is case sensitive, i.e. 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 graphic data in 1-bit (black & white) PCX format.*

Remarks

In a DOS system, the "DATA" portion can be sent to the printer via the parallel port using the DOS COPY command.

Example

Let us assume you have a PCX file named LOGO.PCX in your current directory. Use a text editor, e.g. Windows Notepad, to create a text file called e.g. STOREIT.TXT and store it in the same directory as the .PCX file:

```
↵
GM"LOGO"1421 ↵
```

At the DOS prompt, type:

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

(Stores the image in the default printer).
or...

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

(Stores the image in the printer connected to port LPT1).

After downloading, the **GI** command can be used to verify that the graphic was successfully stored. If the downloading did not succeed, first check that the .PCX file is in 1-bit (black & white) format and that the free graphics memory in the printer is large enough to store the graphics.

Important!
Always make backup copies on the host! If you need to change the memory allocation (see ***M*** *command), or if the RAM backup batteries should run out, all formats and graphics stored in the printer and memory cartridge will be lost.*

I – Character Set Selection

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

Syntax

Ip_1,p_2,p_3 ***(I is uppercase i)***

Parameters

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

Table 1: Printer Code Page

7 data bits (p_1=7)		**8 data bits (p_1=8)**		
p_2	***Country***	***p_2***	***Code Page***	***Country***
0	USA	0	437	English
1	British	1	850	Multilingual (Latin 1)
2	German	4	863	Canadian (French)
3	French	5	865	Norwegian
4	Danish	*(In case code pages 437, 863, or 865 do not produce the desired characters, use code page 850 Multilingual)*		
5	Italian			
6	Spanish			
7	Swedish			
8	Swiss			

Table 2: KDU Country Code (8 bits only)

Code	***Country***	***Code***	***Country***
001	U.S.A.	041	Switzerland
002	Canada	044	U.K.
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		

The default setting is **I**8,0,001. For additional information and 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 (KDU)

JB – Disable Top of Form Backup

Description

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

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 paper after printing the label, so as to allow the paper to be torn off or peeled off properly.

By default, the paper 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, i.e. 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.

Example

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

JF – Enable Top of Form Backup

Description

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

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 paper after printing the label, as to allow the paper to be torn off or peeled off properly.

By default, top of form is enabled, i.e. the paper 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, i.e. 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.

Example

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

j – Paper Feed Adjustment

Description

This command makes it possible to set the paper feed for either tear-off or peel-off operation.

Syntax

```
jp1
```

Parameters

p_1 *Length of paper feed after printing in dots (0–999)*
Recommended values:

Tear-off operation:	*140 (at 203.2 dpi)*	*214 (at 300 dpi)*
Peel-off operation:	*110 (at 203.2 dpi)*	*173 (at 300 dpi)*

Remarks

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

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

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

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

Examples

```
j110 ↵                    :Adjustment for peel-off operation1
j140 ↵                    :Adjustment for tear-off operation1
```

[1]/. Values refer to an EasyCoder 91 with 203.2 dpi printhead density.

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

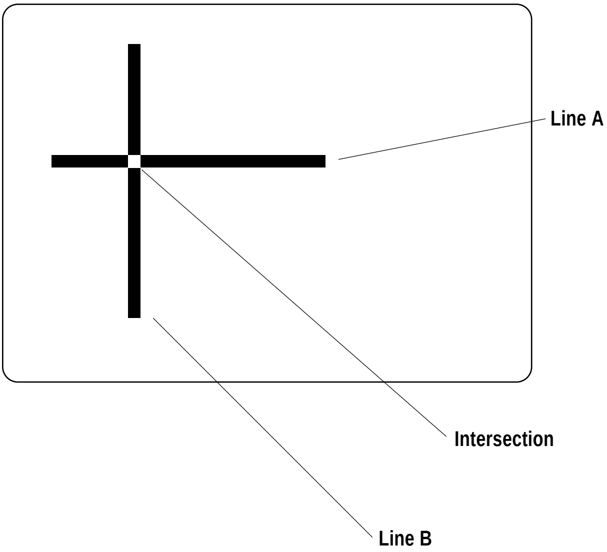
LEp_1,p_2,p_3,p_4

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.

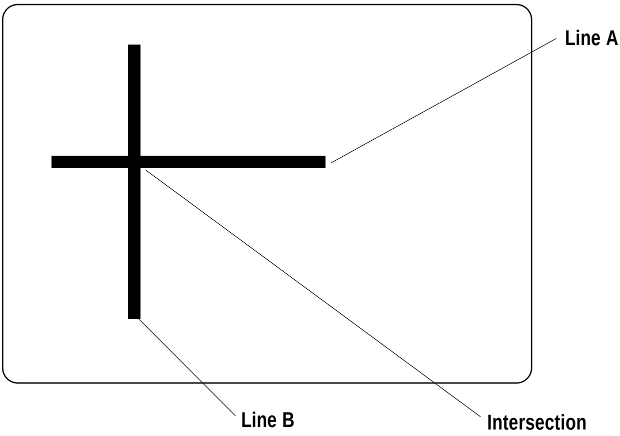
Syntax **LOp_1,p_2,p_3,p_4**

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

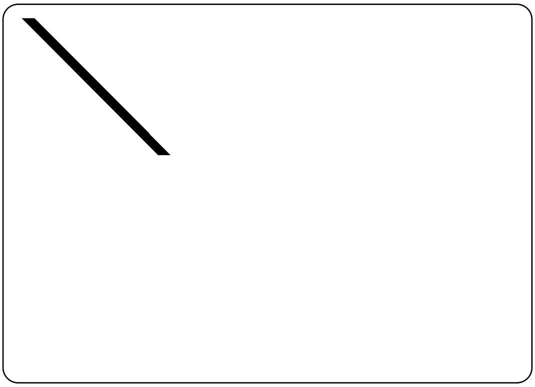
LSp$_1$,p$_2$,p$_3$,p$_4$,p$_5$

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 (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.

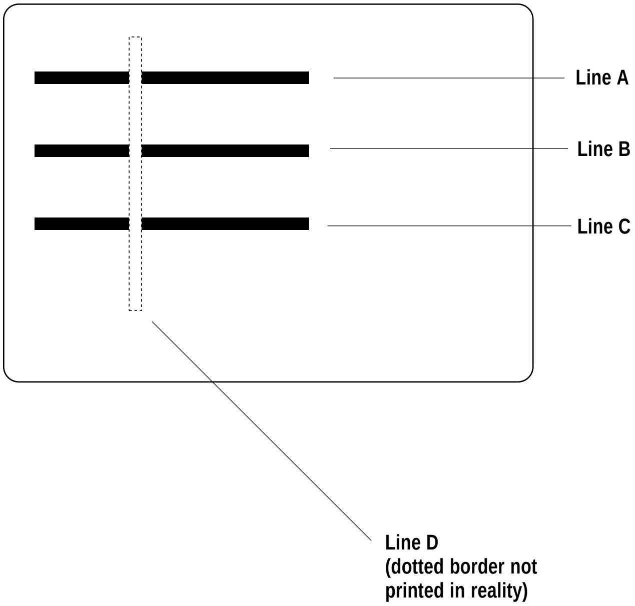
Syntax LWp_1,p_2,p_3,p_4

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
```



M – Memory Allocation

Description

This command is used to allocate or partition the printer's memory into separate areas for image buffer, forms, graphics, and soft fonts (not used).

Syntax

Mp_1,p_2,p_3

Parameters

p_1	*Image buffer area in whole Kbytes*
p_2	*Form memory area in whole Kbytes*
p_3	*Graphic memory area in whole Kbytes*
	All remaining memory will be allocated as soft font memory.

Remarks

The command to allocate the memory may have to be performed to initialize the printer if the current memory areas are too small.

Important:
The M command will also erase all forms and graphics and return printer default settings.

Default Memory Allocation

The **M** command line will set image buffer, form and graphic memory area. The remainder will automatically be allocated to a *Soft Fonts Memory* (also called “E-memory”), a feature presently not used in *EasyCoder 91*. As standard, the printer's memory is allocated like this:

***EasyCoder 91* 8 dots/mm (203.2 dpi):**

Internal memory size:	128 Kbytes
Memory cartridge:	Optional (+128 or +386 Kbytes)
Image buffer:	106 Kbytes
Form memory:	5.1 Kbytes
Graphics memory:	5 Kbytes
Soft fonts memory:	3 Kbytes

The printer's firmware requires approx. 9 Kbytes regardless of printhead density.

***EasyCoder 91* 11.81 dots/mm (300 dpi):**

Internal memory size:	128 Kbytes
Memory cartridge:	+128 Kbytes (standard)
Image buffer:	188 Kbytes
Form memory:	5.1 Kbytes
Graphics memory:	5 Kbytes
Soft fonts memory:	52 Kbytes

The printer's firmware requires approx. 9 Kbytes regardless of printhead density.

M – Memory Allocation, cont'd.

Remarks, cont'd.

Memory Cartridges

The printers can be fitted with a memory cartridge containing an additional 128 kbytes (standard in the 300 dpi versions) or 384 kbytes of RAM. When memory is allocated, the printer's internal memory is used first. Thus, when using memory cartridges, allocate at least 118 kbytes to the image buffer to make sure that forms and graphics are stored in the removable memory cartridge.

An *EasyCoder 91* printer can detect if a memory cartridge if inserted or not:

- If a cartridge is present, the printer will take the setup information from the cartridge.
- If a cartridge is not present, then the printer will take the setup information from its internal RAM memory.
- If a memory cartridge is removed, the printer will use its default setup, see page 12.

Therefore, memory cartridges can be moved from printer to printer and function the same way in each one.

Checking the Memory Allocation

The amount of memory and the current allocation can be printed on a label using the **U** command, or by printing a test label in the Test Mode, see Installation & Operation manual.

When to Re-allocate the Memory

- If your label size is larger then the current image buffer.
- If you need to change the size of the forms memory to accommodate more or less forms.
- If you need to change the size of the graphics memory to accommodate more or less graphics.
- If you have replaced the EPROM
- If you have fitted an unformatted memory cartridge.

Image Buffer

The image buffer is the area where the active print image is temporarily stored. Calculate the memory size needed for your *image* area by measuring the largest form intended to be printed (take future needs into consideration).

For less than full width labels, also refer to the **q** command, which allows trading off print width for increased label length with the same image buffer size.

M – Memory Allocation, cont'd.

Remarks, cont'd.

Formulas:
[(Height in mm x Dots per mm) x (Width in mm x Dots per mm)]/ (1024 x 8) = Kbytes required
or
[(Height in inches x Dots per inch) x (Width in inches x Dots per inch)]/ (1024 x 8) = Kbytes required

The printhead has a density of either 8 dots per mm (203.2 dots per inch) or 11.81 dots per mm (300 dots per inch).

Rule of thumb for <u>full</u> width labels (Density 8 dots/mm):
Label height in inches × 22Kb (Min. 44Kb)
Label height in mm × 1Kb (Min. 44Kb)
Round off to the next higher whole number.

Rule of thumb for <u>full</u> width labels (Density 11.81 dots/mm):
Label height in inches × 46.8 Kb (Min. 22Kb)
Label height in mm × 1.842 Kb (Min. 22Kb)
Round off to the next higher whole number.

Form Memory

The Form memory is for permanent storage of label forms. A typical form requires 1 kbyte of memory. The size of each form can, for example, be displayed with a DIR command at the DOS prompt.

Graphics Memory

The Graphics memory is for permanent storage of label graphics. Graphic files can vary greatly in size. The size of each PCX file can, for example, be displayed with a DIR command at the DOS prompt.

Examples

Resetting the memory via the serial port:
The example below formats the memory for a 127 mm (5)" long **full width** label for a printer with 8 dots/mm printhead density.

Note that the memory allocation values returned e.g. by a **U** command may differ slightly from the values entered using an **M** command because of certain round off calculations in the firmware.

This should have few practical consequences and can generally be ignored.

```
M104,5,5 ↵            :Sets the memory to the value 106,5.1.5,3
```

M – Memory Allocation, cont'd.

Examples, cont'd.

Resetting the memory via the parallel port (Windows driver): When installing a memory cartridge, or when you need to print extra long labels (see below), you may want to change the memory allocation without having to set up a serial communication. Using the *MS-DOS Prompt* in *Windows 3.1x*, you can send the necessary **M** command via the parallel port as follows. The example assumes that MS Windows 3.1x is installed in drive C:\ and that the printer is connected to LPT1:.

In a text editor, e.g. *Windows Notepad*, write the **M** command, e.g.:

```
M118,20,106 ↵
```

Save the text file in the directory **c:\windows\** under a suitable name (e.g. **memsetup.txt**).

In the ***Main*** group of *Windows 3.1x* ***Program Manager***, double-click the ***MS-DOS Prompt*** icon.

In ***MS-DOS***, the directory **c:\windows\** is selected by default:

```
C:\WINDOWS>_
```

Enter the following ***DOS*** command:

```
C:\WINDOWS>copy memsetup.txt lpt1: ↵
```

MS-DOS responds by displaying:

```
1 file(s) copied
C\:WINDOWS>
```

Exit ***MS-DOS*** by typing:

```
C\:WINDOWS> exit ↵
```

M – Memory Allocation, cont'd.

Examples, cont'd.

Maximizing the Image Buffer

When using the *Windows* printer driver, or the Direct Mode only, you have no need for any form or soft font (E) memory. In the *Windows* printer driver, you do not need any graphics memory at all, and possibly you can also dispense with graphics in the Direct Mode. Thus, to be able to print as long labels as possible, you can allocate most of or the entire RAM memory to the image buffer:

```
M117,0,0 ↵                    :Sets max. image buffer for
                               printer w/o memory cartridge
M245,0,0 ↵                    :Sets max. image buffer for
                               printer w. 128 Kbyte cartridge
M501,0,0 ↵                    :Sets max. image buffer for
                               printer w. 384 Kbyte cartridge
```

This table illustrates the connection between the **M** command, the memory allocated to the print buffer and the maximum print length at full print width in the Direct Mode in an *EasyCoder 91* with a printhead density of 8 dots per mm (203.2 dots per inch):

Command	*Image Buffer*	*Maximum Print Length*
`M117,0,0 ↵`	119 Kbyte	1150 dots = 143.75 mm (5.65")
`M245,0,0 ↵`	250 Kbyte	2400 dots = 300.00 mm (11.81")
`M501,0,0 ↵`	513 Kbyte	4930 dots = 616.25 mm (24.26")

N – Clear Image Buffer

Description

This command is used to clear the image buffer before building a new image.

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 disable or enable various printer options.

Syntax

```
O[S, N, D]
```

Parameters

S *Reverse operation of label gap sensor, so the sensor will interpret blockage of light as a gap.*
N *Disable label-taken sensor.*
D *Disable the ribbon end sensor (EasyCoder 91 TT only).*

Remarks

An **O** command without any trailing parameter resets options to their respective the default settings.

An **O** command supplemented by one or several trailing parameters changes the settings for the parameters included in the **O** command.

S Parameter:
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 91* printer with the appropriate type of paper. By default, the sensor will interpret blockage of light as a label or similar.

N Parameter:
EasyCoder 91 Peel-off models are fitted with a label taken sensor. When the label taken sensor is enabled (default), the communication to the printer will be BUSY as long as the sensor detects a label in the outfeed slot.

D parameter:
The ribbon end sensor of the thermal transfer capable versions of *EasyCoder 91* detects reflections from the trailing silvery part of the transfer ribbon. Once the ribbon has been removed, the error is cleared and you can either load a new supply of transfer ribbon, or change to direct ther-mal paper. However, switching between thermal transfer printing and direct thermal printing requires the heat density to be adjusted using a **D** command, see page 40.

O – Options Select, cont'd.

Examples *These examples show how to enable or disable the various options:*

```
O ↵                         :All options set to default
ON ↵                        :Normal label gap sensor operation
                            :LTS disabled
                            :Ribbon end sensor enabled
ON,D ↵                      :Normal label gap sensor operation
                            :LTS disabled
                            :Ribbon end sensor disabled
OS,N,D ↵                    :Reverse label gap sensor operation
                            :LTS disabled
                            :Ribbon end sensor disabled
```

P – Print

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

Syntax

Pp_1[,p_2]

Parameters

p_1	*Numbers of label sets (1–65535). Default 0.*
p_2	*Number of copies of each label (1–65535). Used in combination with counters to print multiple copies of the same label.*

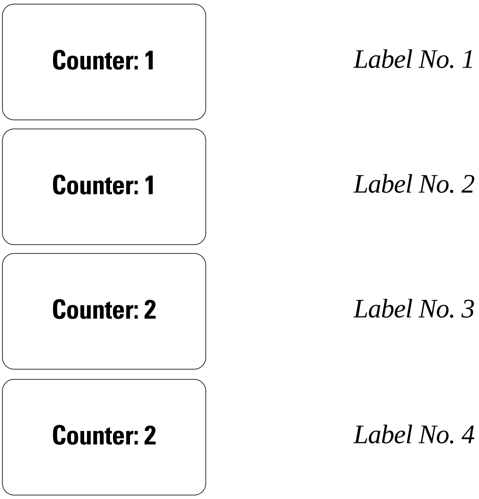
Remarks

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). Default 0.*

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.

Warning!
*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 “1”
FS"TEST6" ↵                    :Starts form store sequence
V00,50,N,"Enter text" ↵                  :Defines variable
A24,24,0,4,1,1,N,V00 ↵           :Writes text w. variable
PA1 ↵                      :Prints 1 label automatically
FE ↵                            :Ends form store sequence

FR"TEST6" ↵                            :Retrieves form “1”
? ↵                                        :Gets variables
This is variable text             :Data for variable 00
```

Q – Set Form Length (std)

Description

This command is used to set the form and gap length when using the standard label gap sensor.

Syntax

Qp_1,p_2[±p_3]

Parameters

p_1	*Label length measured in dots*
p_2	*Gap length measured in dots*
$\pm p_3$	*Optional offset length measured in dots*

Remarks

As standard, all *EasyCoder 91* printers have a label gap sensor designed to detect the top of each label or tag. It does this in two ways:

- By looking through the semi-transparent backing paper in the gap between labels, or
- By looking through a hole in the tag.

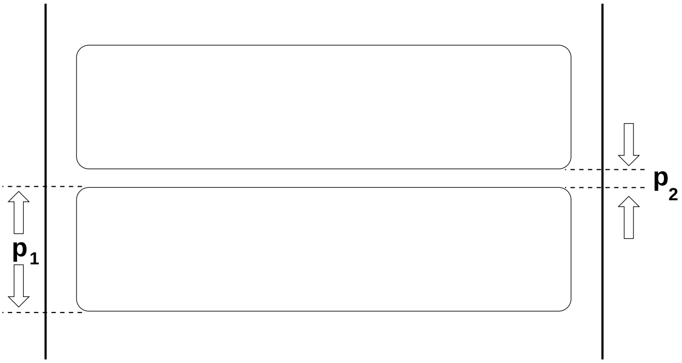
The sensor is located in the centre of the label path.

When entering the Dump Mode (see page 4), or when printing a form for the first time after power-up using the *Windows Driver*, 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.

Examples

Rectangular label:

$\mathbf{p_1}$ =	20.0 mm	(160 dots)
$\mathbf{p_2}$ =	3.0 mm	(24 dots)



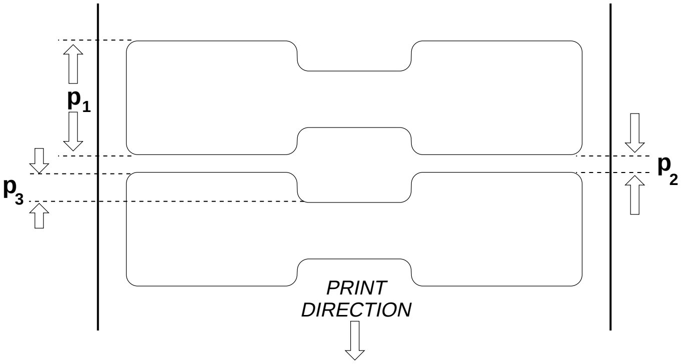
The **Q** command would be:

```
Q160,24 ↵
```

Q – Set Form Length (std), cont'd.

Examples, cont'd. **Butterfly label:**

$\mathbf{p_1}$ =	12.5 mm	(100 dots)
$\mathbf{p_2}$ =	3.0 mm	(24 dots)
$\mathbf{p_3}$ =	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 optional black mark sensor, and to specify the location and height of the black marks on the back of the paper.

Syntax

```
Qp1Bp2[±p3]
```

Parameters

p_1	*Distance between black marks measured in dots*
B	*Disables LSS, enables BMS*
p_2	*Height of black mark measured in dots*
$\pm p_3$	*Optional offset length measured in dots*

Remarks

As standard, all *EasyCoder 91* printers have a label gap sensor designed to detect the top of each label or tag. It can be supplemented with an optional black mark sensor (factory installed option only) that determines the top of each label or tag by sensing a preprinted black mark on the back of the paper.

The black marks should be centre-aligned on the paper and have the following dimensions:

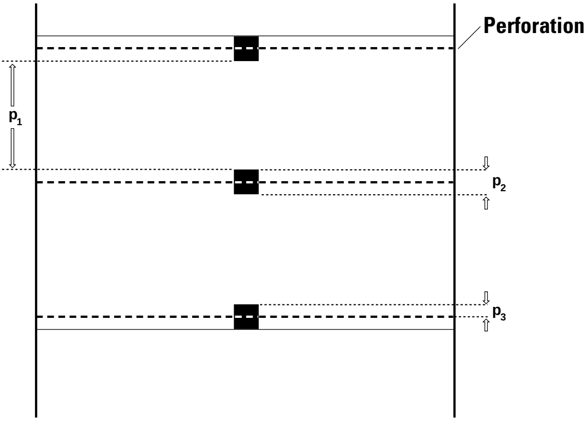
Maximum height:	15 mm	(0.59")
Minimum height:	3 mm	(0.12")
Recommended height:	5 mm	(0.20")
Recommended width:	≥ 10 mm	(0.39")

Q – Set Form Length (Black Mark), cont'd.

Examples

On this tag, the black marks are printed on the perforation:

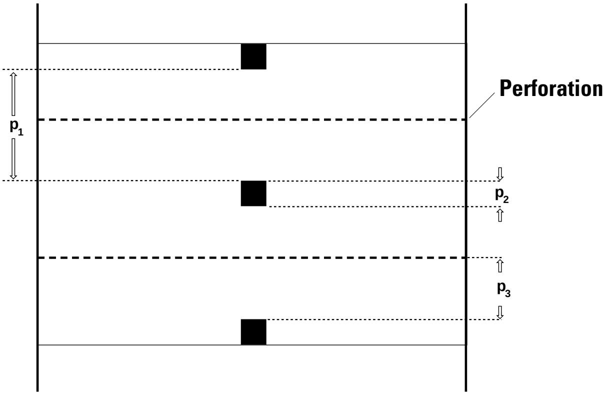
$\mathbf{p_1}$ =	31.0 mm	(248 dots)
$\mathbf{p_2}$=	7.0 mm	(56 dots)
$\mathbf{p_3}$=	0.5 mm	(4 dots)



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

On the tag below, the black marks are printed between the perforations:

$\mathbf{p_1}$ =	31.0 mm	(248 dots)
$\mathbf{p_2}$=	7.0 mm	(56 dots)
$\mathbf{p_3}$=	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*

Remarks

The **q** command will cause the image buffer (see **M** command) to be formatted to match the label width, i.e. width is traded off for increased length within the same memory size. This allows printing long narrow labels with a minimum of memory.

The **q**-value will automatically be rounded off to the closest multiple of 8 or 12, e.g. if **q** is entered as 500, the actual **q**-value will be 496 (62 × 8) in case of an 8 dots/mm printer.

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

(No. of dots on printhead – label width in dots)/2 (centre aligned)

There are either 8 dots per mm and 203.2 dots per inch, or 11.81 dots per mm and 300 dots per inch, depending on printer model.

Important!
*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

Rp_1,p_2

Parameters

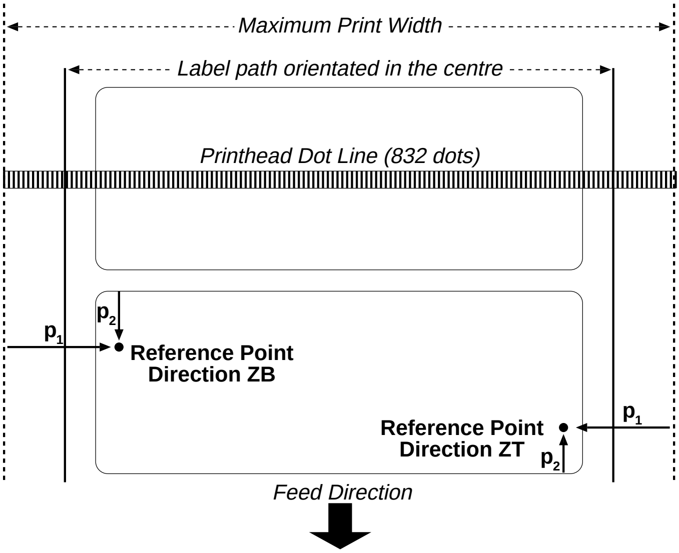
p_1	*Horizontal (left) margin measured in dots (default 000)*
p_2	*Vertical (top) margin measured in dots (default 000)*

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.

Warning!
Repeated printing off the edge of the label can cause excessive printhead wear.

Note that 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:



Example

`R50,100 ↵` *:Creates a 50 dot left margin and a 100 dot top margin on an 8 dots/mm printer.*

S – Speed Select

Description This command is used to select the label speed while printing.

Syntax

```
Sp1
```

Parameters ***p_1*** *Speed select value:*

0	*25 mm/sec. (1"/sec.)*
1	*37 mm/sec. (1.5"/sec.)*
2	*50 mm/sec. (2"/sec.)*

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

Example **`S0 ↵`** *:Sets the print speed to 25 mm/sec (1"/sec.).*

TD – Define Date Layout

Description

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

Syntax

TDp_1[/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, e.g. 96*
y4	*Year displayed as 4 digits, e.g. 1996*
me	*Month displayed as a 3-letter English abbreviation, e.g. JAN, FEB, MAR, APR, MAY etc.*
mn	*Month displayed as 2 digits, e.g. 01*
dd	*Day displayed as 2 digits, e.g. 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.*

Remarks

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

Examples

If the current date is June 15, 1998:

`TDy2/me/dd` ↵	*:Prints as 98/JUN/15*
`TDdd-me-y4` ↵	*:Prints as 15-JUN-1998*
`TDdd,mn,y4` ↵	*:Prints as 15,06,1998*
`TDy4-mn-dd` ↵	*:Prints as 1998-06-15*

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

```
TSp1,p2,p3,p4,p5,p6
```

Parameters

p_1	*Month (01–12)*
p_2	*Day (01–31)*
p_3	*Year, two last digits (e.g. 96)*
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 a Memory Cartridge containing an optional real-time clock circuit (RTC).

Example

`TS06,15,98,12,45,23` ↵ *:Sets the date to June 15, 1998 and the time to 12:45:23 p.m.*

TT – Define Time Layout

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

Syntax

```
TTp1[/p2/p3][+]
```

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, e.g. 12*
m	*Minutes displayed as 2 digits, e.g. 15*
s	*Seconds displayed as 2 digits, e.g. 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.*

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

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

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

U – Print Configuration (General)

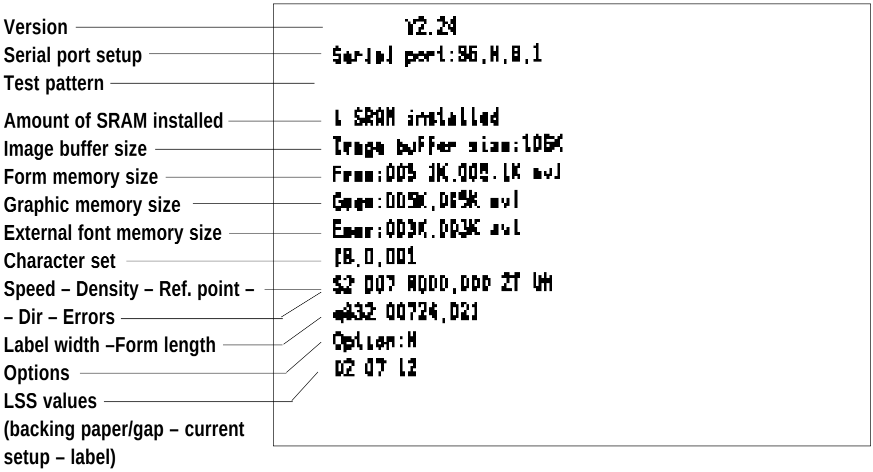
Description This command is used to print the current printer configuration.

Syntax

```
U
```

Remarks This command produces a single label identical to the one printed in the Dump Mode (see page 4), but without entering the Dump Mode.

Example **U ↵** *:Produces e.g. the following label*



```
        V2.24
Serial port:96,N,8,1

1 SRAM installed
Image buffer size:106K
Form:003 1K,005.1K avl
Gmem:005K,065K avl
Emem:003K,003K avl
C8,0,001
S2 D07 R000,000 2T UN
W432 00724,021
Option:N
02 07 12
```

Note:
If a real-time clock circuit is fitted, the present time and date according to the clock circuit will also be printed at the bottom of the label.

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 serial RS 232C port.

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

Example

UF *:Returns number of forms and all form names, e.g.:*

```
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 serial RS 232C port.

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

Example

UG *:Returns number of graphics and all graphic names, e.g.:*

```
UG001
LOGO
```

UI – Enable Prompts/Codepage 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 serial RS 232C port.

Syntax

UI

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

UIp_1,p_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, e.g.:

```
UI80,001
```

Also see

I and **U** commands

UM – Codepage & Memory Inquiry

Description

This command will cause the printer to send the currently selected code page and memory status to the host through the serial RS 232C 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:

UM $p_1,p_2,p_3,p_4,p_5,p_6,p_7$,UI p_8,p_9,p_{10}

Parameters

p_1	*Image buffer size in Kbytes*
p_2	*Form memory allocation size in Kbytes*
p_3	*Form memory free in Kbytes*
p_4	*Graphic memory allocation size in Kbytes*
p_5	*Graphic memory free in Kbytes*
p_6	*External font memory allocation size in Kbytes*
p_7	*External font memory free in Kbytes*
p_8	*Number of data bits*
p_9	*Code page*
p_{10}	*Country code*

Example

`UM ↵` *:Returns memory status and current code page, e.g.:*

```
UM106,005.1,005.0,005,005,003,003
                                UI80,001
```

Also see

I, **M**, **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 – Codepage & 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 serial RS 232C 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:

$\mathbf{UM}\ p_1,p_2,p_3,p_4,p_5,p_6,p_7,\mathbf{UI}\ p_8,p_9,p_{10}$

Parameters

p_1	*Image buffer size in Kbytes*
p_2	*Form memory allocation size in Kbytes*
p_3	*Form memory free in Kbytes*
p_4	*Graphic memory allocation size in Kbytes*
p_5	*Graphic memory free in Kbytes*
p_6	*External font memory allocation size in Kbytes*
p_7	*External font memory free in Kbytes*
p_8	*Number of data bits*
p_9	*Code page*
p_{10}	*Country code*

Example

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

Also see

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

US – Enable Error Reporting

Description

This command is used to enable the printer's status reporting feature.

Syntax

```
US
```

Remarks

Serial Port:
If an error occurs while using the serial port, the printer will send a NAK (ASCII 21 dec.), followed by the error number, back to the computer. If no error occur, the printer will echo ACK (ASCII 06 dec.) after each **P** (print) command.

If out-of-paper or out-of-ribbon occurs, the printer will send, through the serial port, a “**-07**” and “**Pnnn**” where **nnn** is the number of labels remaining to print.

Parallel Port:
While using the parallel port, the printer will print the error number and the control lamp will go orange (error).

The default setting is off (also see **UN**).

Error Messages

Message	*Meaning*
ERR01	Syntax Error
ERR02	Object exceeds image buffer border
ERR03	Data length error (e.g. EAN 13 is 12 or 13 bytes only)
ERR04	Insufficient memory to store forms or graphics
ERR05	Memory configuration error
ERR06	RS 232C error
ERR07	Out of paper and/or ribbon
ERR08	Form or PCX name duplicate
ERR09	Form or PCX not found
ERR16	No form was retrieved before “**?** ↵” was entered.
ERR50	Does not fit in area specified
ERR51	Data length too long

HINT!
Tap the Feed key three times to resume printing after an error.

Example

US ↵ *:Enables error reporting*

V – Define Variable

Description

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

Syntax

```
Vp1,p2,p3,"[-][- -]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	*Centre 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 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 centre justification are selected, the counter value will be printed left, right or centre justified in an area with a width defined by the $\mathbf{p_2}$ parameter. If the number of digits in the counter value is less than the number of digits defined by $\mathbf{p_2}$, 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.

V – Define Variable, cont'd.

Example

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

```
FK"TEST7" ↵
FR"TEST7" ↵
V00,10,L,"Variable 00" ↵
V02,10,R,"Variable 00" ↵
V03,10,C,"Variable 00" ↵
V04,10,N,"Variable 00" ↵
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":Centre justified" ↵
A50,200,0,3,1,1,N,"TEXT"V03":No justification" ↵
FE ↵
```

Refer to the **?** *command on page 94 for continuation of this example!*

W – Windows Mode

Description

This command is used to enable/disable the Windows command mode.

Syntax

Wp_1

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.

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

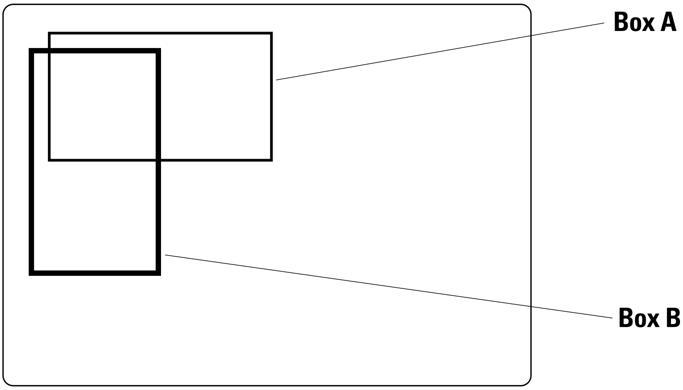
```
Xp1,p2,p3,p4,p5
```

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 (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 serial port communication parameters.

Syntax

```
Yp1,p2,p3,p4
```

Parameters

p_1 *Baud rate:*

19	*19,200 baud*
96	*9,600 baud*
48	*4,800 baud*
24	*2,400 baud*
12	*1,200 baud*

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 communication port.

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

If the current communication setup is not known, it can be checked by printing a test label (see page 4).

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

```
Zp1
```

Parameters ***p_1*** *Print orientation:*

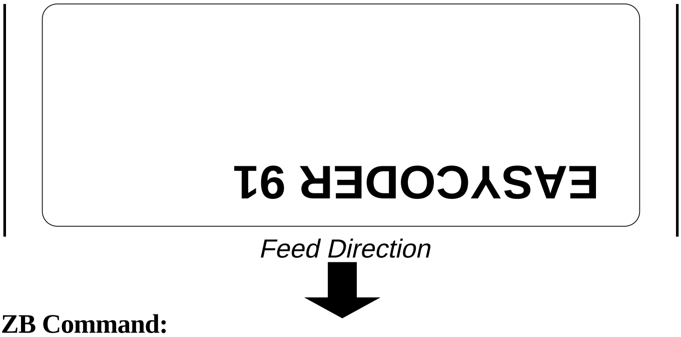
T *Start printing from the top of the label (default)*

B *Start printing from the bottom of the label*

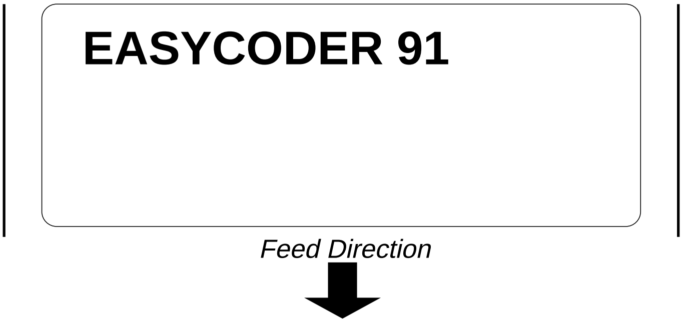
Remarks This command affects the complete print image, including text, bar codes, graphics, lines, and boxes, as well as the location of the reference point (see **R** command).

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, i.e. **ZT**.

ZT Command:



ZB Command:



Example **ZB** ↵ *:Starts printing from the bottom of the label*

? – Download Variables

Description

This command is used to signal to the printer that the data following are variable or counter values. It also makes the printer read the optional real-time clock circuit.

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.

If the form contains time and/or date fields, they will only be printed if the **FR** (form retrieve command) is followed by a **?** command line. The **?** command allows the printer to read its optional real-time clock circuit.

Important!
If the ? command is omitted, no variables, counter values, time fields or date field will be printed.

Example

```
FR"TEST7" ↵        :Retrieves the form “TEST7”, see page 90
? ↵                                      :Variables follow
12345 ↵                                :Variable 00 entered
abcde ↵                                :Variable 01 entered
ABCDE ↵                                :Variable 02 entered
99999 ↵                                :Variable 03 entered
P1 ↵                                      :Prints one label
```

Fonts

Resident Fonts

The *EasyCoder 91* printers support 160 different characters for font sizes 1 – 4 and 80 characters for font size 5. All fonts are non-proportional. The ASCII value of the different characters is determined by the I command setting.

Printhead Density 8 dots/mm (203.2 dpi)

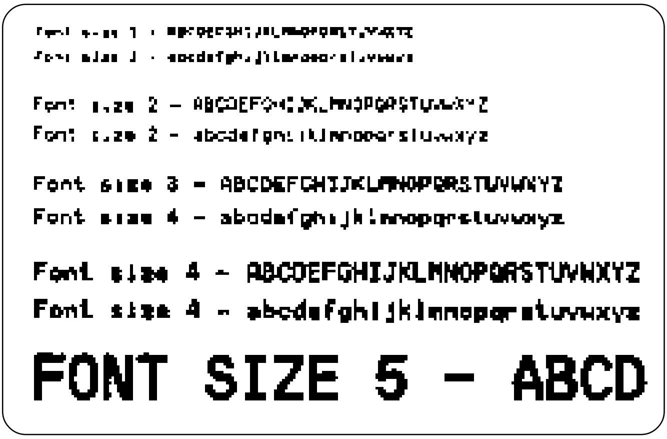
Font	Size (dots)	Size (points)	Char./inch
1	8 x 12	6	20.3
2	10 x 16	7	16.9
3	12 x 20	10	14.5
4	14 x 24	12	12.7
5	32 x 48	24	5.6

Printhead Density 11.81 dots/mm (300 dpi)

Font	Size (dots)	Size (points)	Char./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	21	6.25

Font Sizes 1-5

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



Font size 1 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 1 - abcdefghijklmnopqrstuvwxyz

Font size 2 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 2 - abcdefghijklmnopqrstuvwxyz

Font size 3 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 4 - abcdefghijklmnopqrstuvwxyz

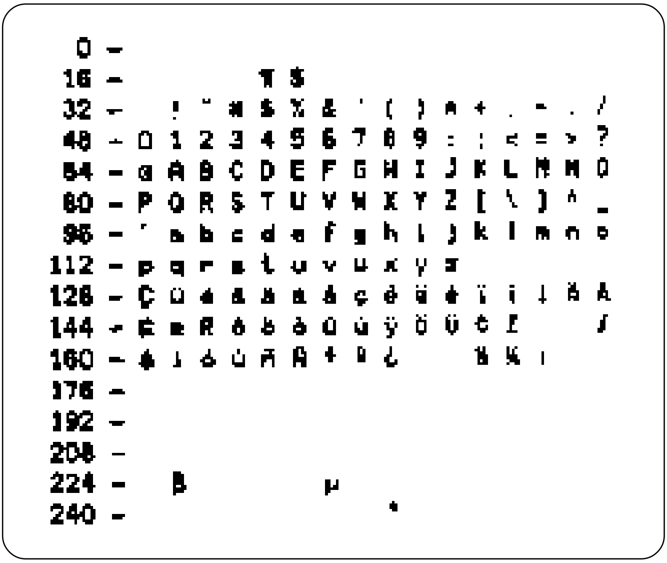
Font size 4 - ABCDEFGHIJKLMNOPQRSTUVWXYZ

Font size 4 - abcdefghijklmnopqrstuvwxyz

FONT SIZE 5 - ABCD

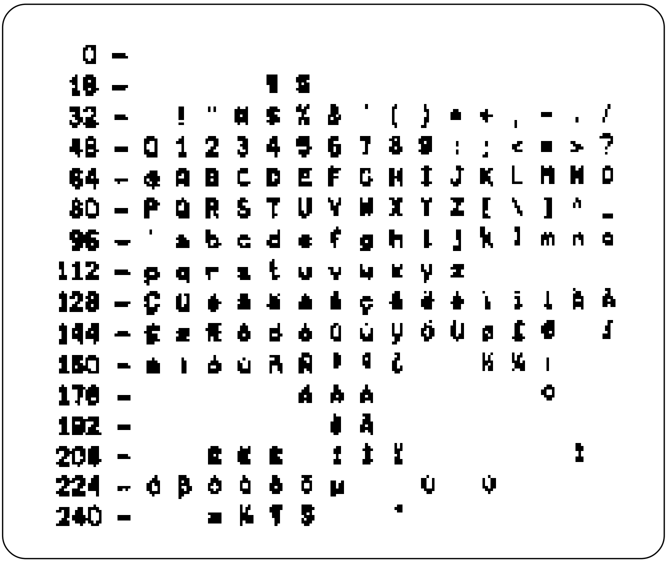
Codepages and Character Sets

Size 1–4 (8 bit); Codepage 437



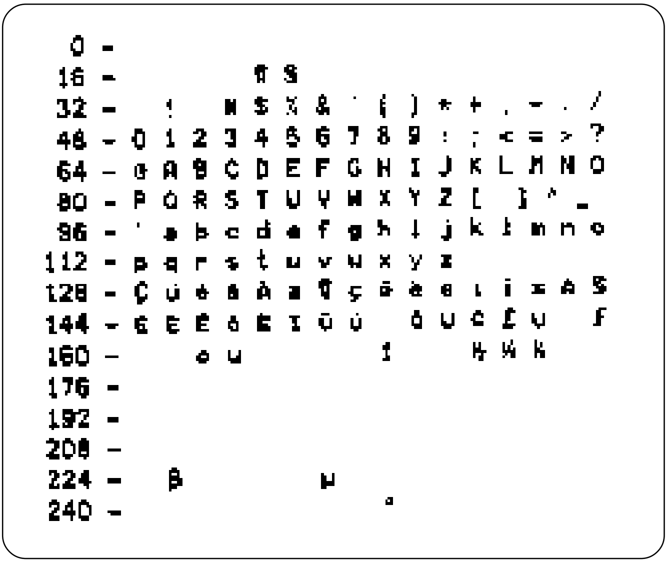
```
  0 –
 16 –         ¶ §
 32 –   ! " # $ % & ' ( ) * + , - . /
 48 – 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 – @ A B C D E F G H I J K L M N O
 80 – P Q R S T U V W X Y Z [ \ ] ^ _
 96 – ` a b c d e f g h i j k l m n o
112 – p q r s t u v w x y z
128 – Ç ü é â ä à å ç ê ë è ï î ì Ä Å
144 – É æ Æ ô ö ò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

Size 1–4 (8 bit); Codepage 850



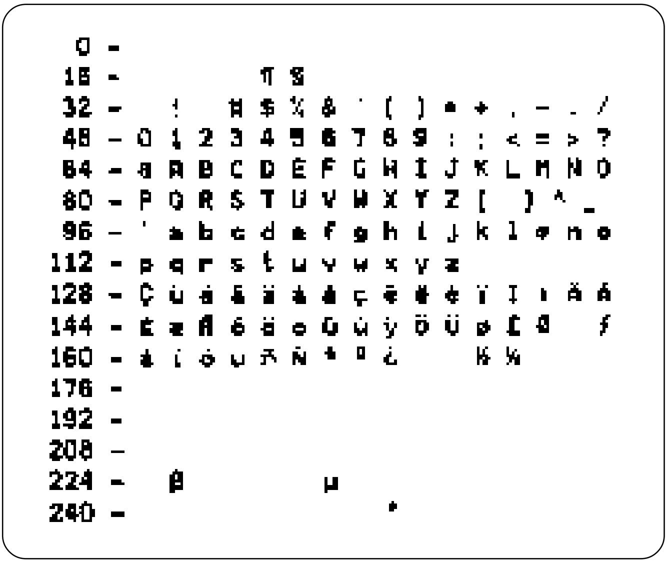
```
  0 –
 16 –         ¶ §
 32 –   ! " # $ % & ' ( ) * + , - . /
 48 – 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 – @ A B C D E F G H I J K L M N O
 80 – P Q R S T U V W X Y Z [ \ ] ^ _
 96 – ` a b c d e f g h i j k l m n o
112 – p q r s t u v w x y z
128 – Ç ü é â ä à å ç ê ë è ï î ì Ä Å
144 – É æ Æ ô ö ò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

Size 1–4 (8 bit); Codepage 863



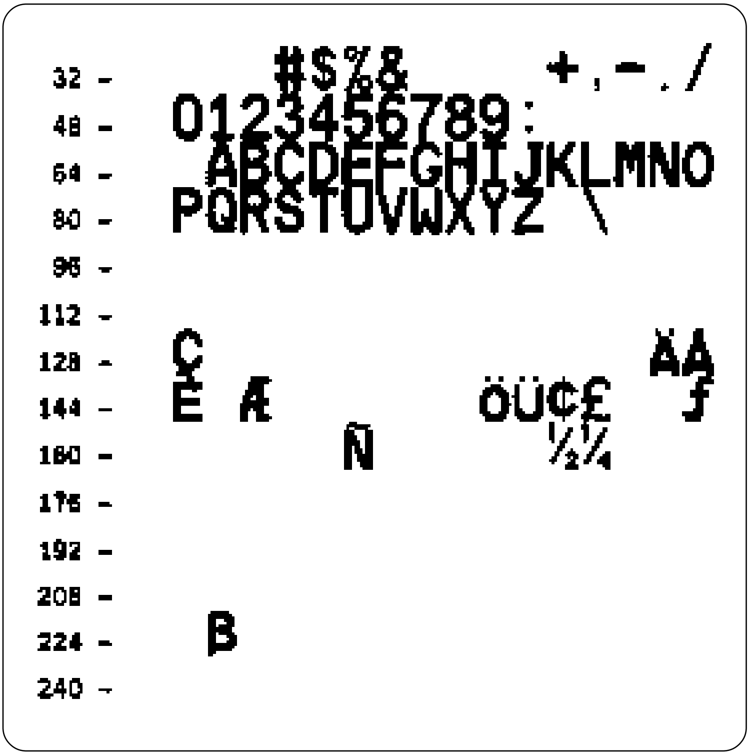
```
  0 -
 16 -         ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [   ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
128 - Ç ü é â Â à ¶ ç ê ë è ï î ‗ À §
144 - É È Ê ô Ë Ï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

Size 1–4 (8 bit); Codepage 865



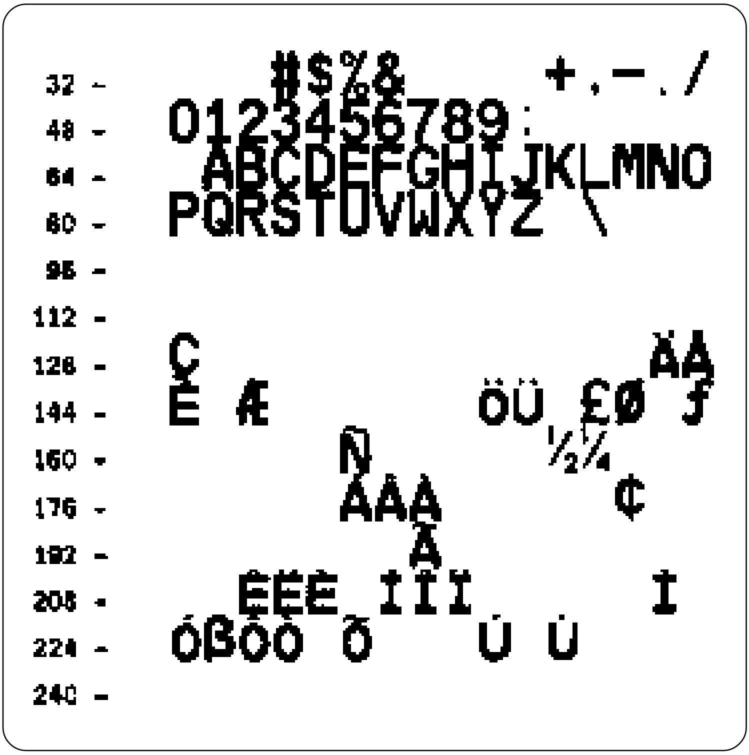
```
  0 -
 16 -         ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [   ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
128 - Ç ü é â ä à å ç ê ë è ï î ì Ä Å
144 - É æ Æ ô ö ò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

Size 5 (8 bit); Codepage 437



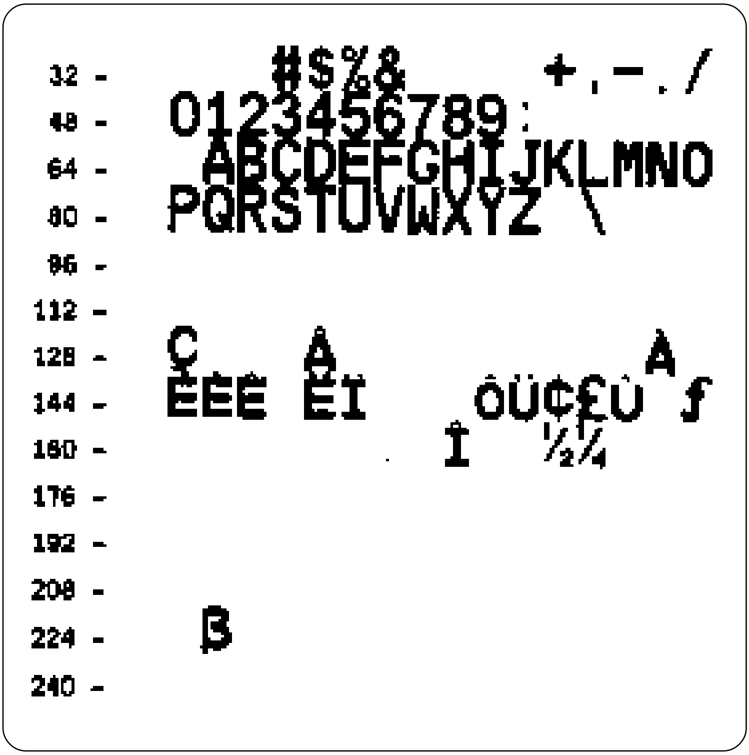
```
 32 -     #$%&    +,-./
 48 -  0123456789:
 64 -   ABCDEFGHIJKLMNO
 80 -  PQRSTUVWXYZ \
 96 -
112 -
128 -  Ç             ÄÅ
144 -  É Æ      ÖÜ¢£  ƒ
160 -       Ñ     ½¼
176 -
192 -
208 -
224 -   ß
240 -
```

Size 5 (8 bit); Codepage 850



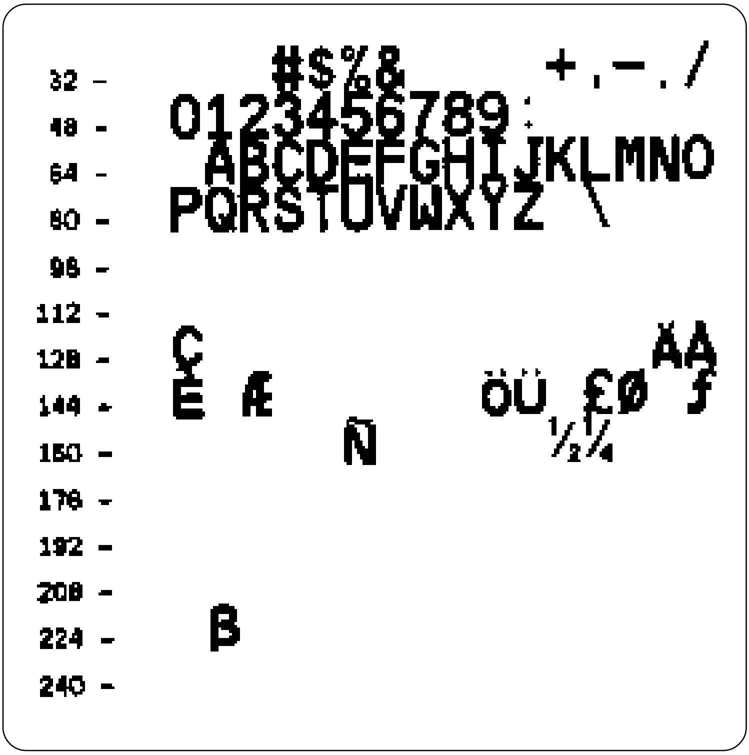
```
 32 -    #$%&    +,-./
 48 - 0123456789:
 64 -  ABCDEFGHIJKLMNO
 80 - PQRSTUVWXYZ \
 96 -
112 -
128 - Ç             ÄÅ
144 - É Æ      ÖÜ £Ø ƒ
160 -      Ñ     ½¼
176 -      ÁÂÀ     ¢
192 -        Ã
208 -   ÊËÈ ÍÎÏ     Ì
224 - ÓßÔÒ Õ   Ú Ù
240 -
```

Size 5 (8 bit); Codepage 863

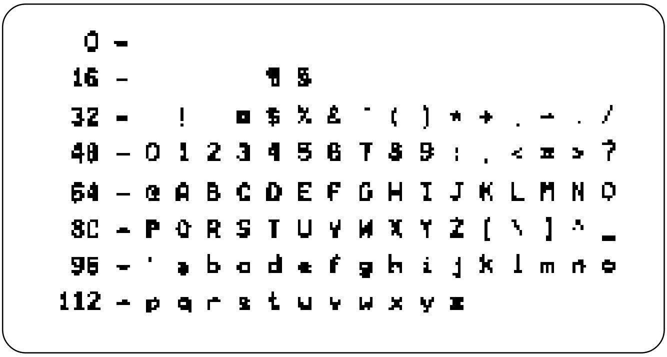


```
32 -     #$%&    +,-./
48 -  0123456789:
64 -   ABCDEFGHIJKLMNO
80 -  PQRSTUVWXYZ \
96 -
112 -
128 - Ç   Â         À
144 - ÉÈÊ ËÏ   ÔÜ¢£Ù ƒ
160 -       . Î  ½¼
176 -
192 -
208 -
224 -  ß
240 -
```

Size 5 (8 bit); Codepage 865

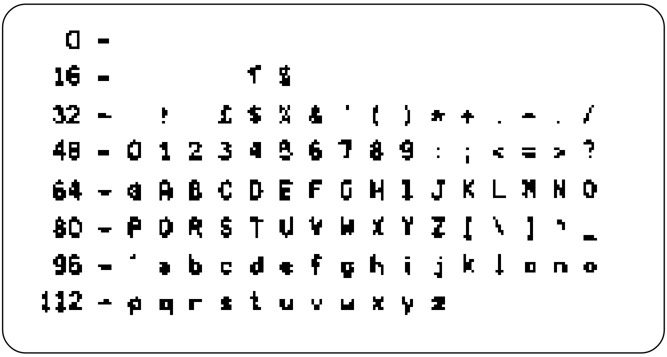

32 - #$%& +,-./
48 - 0123456789:;
64 - ABCDEFGHIJKLMNO
80 - PQRSTUVWXYZ \
96 -
112 -
128 - Ç ÄÅ
144 - É Æ ÖÜ £Ø ƒ
160 - Ñ ½¼
176 -
192 -
208 -
224 - β
240 -

Size 1-4 (7 bit); USA



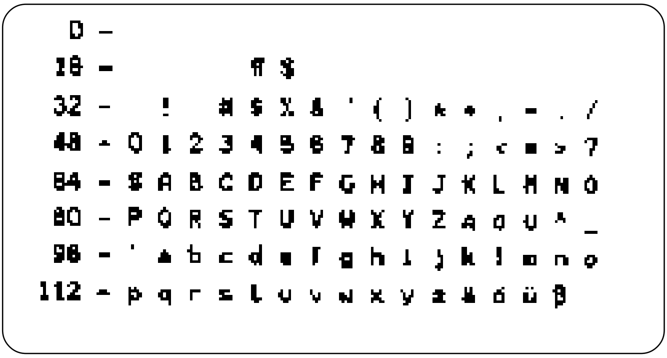
```
  0 -
 16 -              ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
```

Size 1-4 (7 bit); British



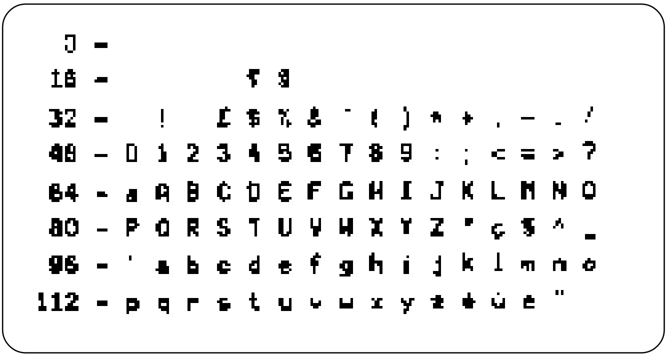
```
  0 -
 16 -              ¶ §
 32 -   !   £ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - @ A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z [ \ ] ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z
```

Size 1-4 (7 bit); German



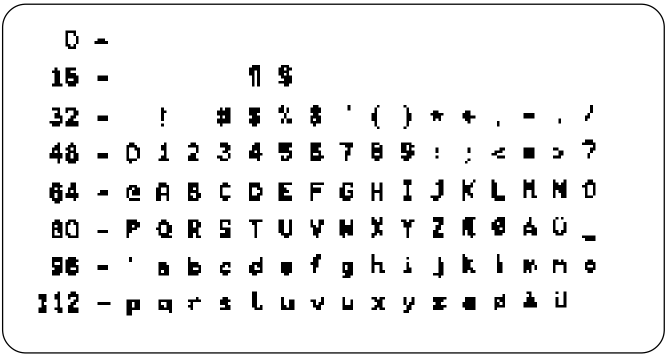
```
  0 -
 16 -              ¶ §
 32 -   !   # $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - § A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Ä Ö Ü ^ _
 96 - ` a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ä ö ü ß
```

Size 1-4 (7 bit); French



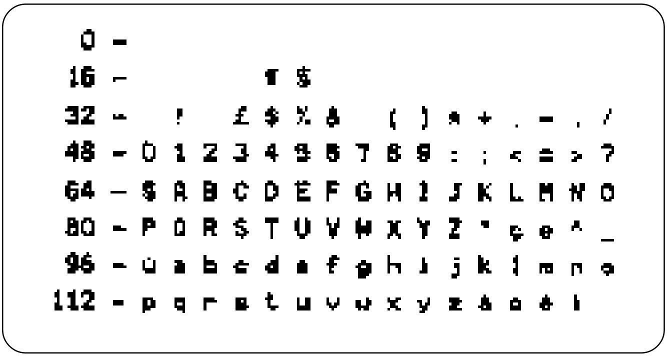
0 –																
16 –					¶	§										
32 –		!		£	$	%	&	'	(	)	*	+	,	-	.	/
48 –	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64 –	à	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80 –	P	Q	R	S	T	U	V	W	X	Y	Z	°	ç	§	^	_
96 –	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112 –	p	q	r	s	t	u	v	w	x	y	z	é	ù	è	¨	

Size 1-4 (7 bit); Danish



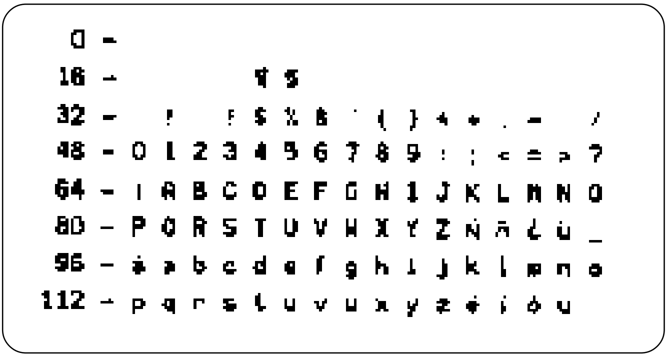
0 –																
16 –					¶	§										
32 –		!		#	$	%	&	'	(	)	*	+	,	-	.	/
48 –	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64 –	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80 –	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	Ü	_
96 –	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112 –	p	q	r	s	t	u	v	w	x	y	z	æ	ø	å	ü	

Size 1-4 (7 bit); Italian



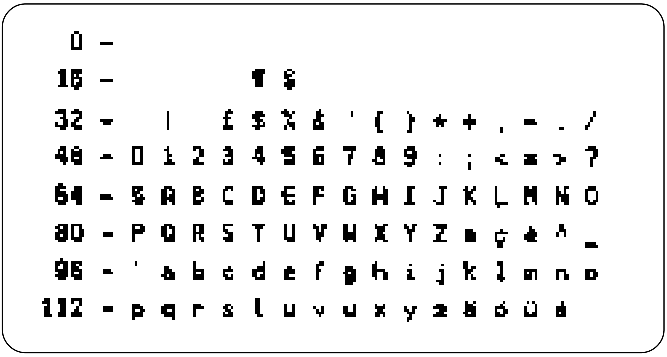
0 –																
16 –					¶	§										
32 –		!		£	$	%	&		(	)	*	+	,	-	.	/
48 –	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64 –	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80 –	P	Q	R	S	T	U	V	W	X	Y	Z	°	ç	é	^	_
96 –	ù	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112 –	p	q	r	s	t	u	v	w	x	y	z	à	ò	è	ì	

Size 1-4 (7 bit); Spanish



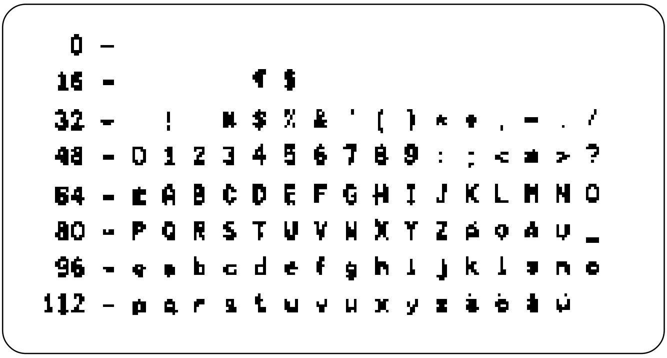
```
  0 -
 16 -         ¶ §
 32 -   !   £ $ % & ' ( ) * + . -   /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - | A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Ñ ñ ¿ ü _
 96 - à a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ë ¡ ö ü
```

Size 1-4 (7 bit); Swedish



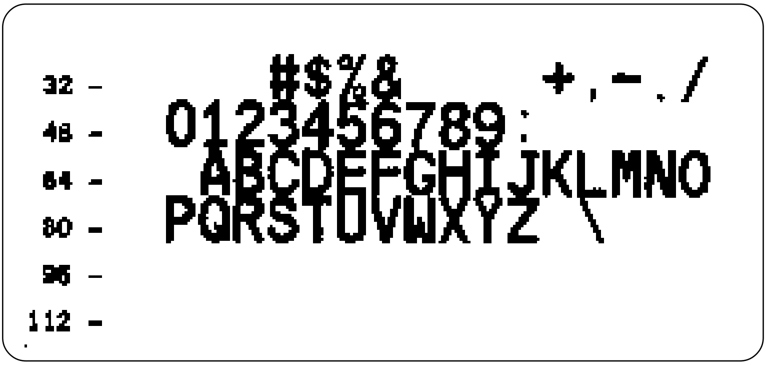
```
  0 -
 16 -         ¶ §
 32 -   |   £ $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - § A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z Ä ç Å ^ _
 96 - ' a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ä ó ü å
```

Size 1-4 (7 bit); Swiss



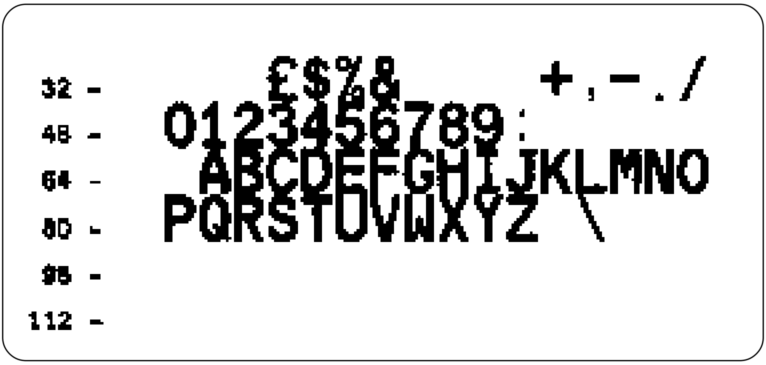
```
  0 -
 16 -         ¶ §
 32 -   !   M $ % & ' ( ) * + , - . /
 48 - 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
 64 - à A B C D E F G H I J K L M N O
 80 - P Q R S T U V W X Y Z é ç â ù _
 96 - ê a b c d e f g h i j k l m n o
112 - p q r s t u v w x y z ä ö ë ü
```

Size 5 (7 bit); USA



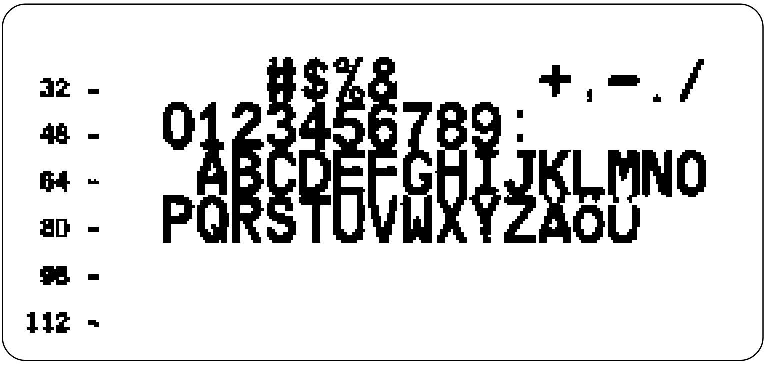
```
 32 -      #$%&    +,-./
 48 -  0123456789:
 64 -   ABCDEFGHIJKLMNO
 80 -  PQRSTUVWXYZ \
 96 -
112 -
```

Size 5 (7 bit); British



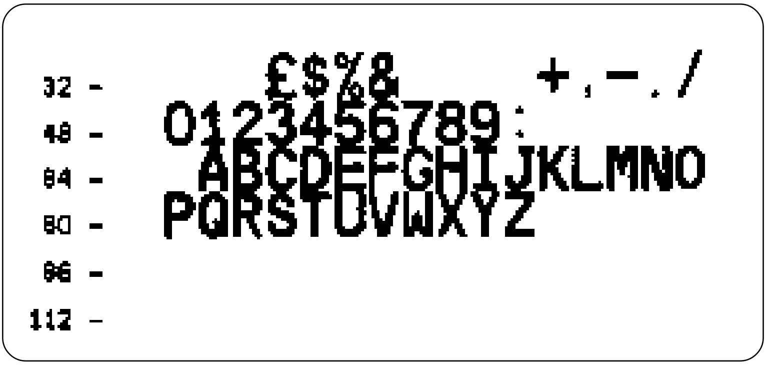
```
 32 -      £$%&    +,-./
 48 -  0123456789:
 64 -   ABCDEFGHIJKLMNO
 80 -  PQRSTUVWXYZ \
 96 -
112 -
```

Size 5 (7 bit); German



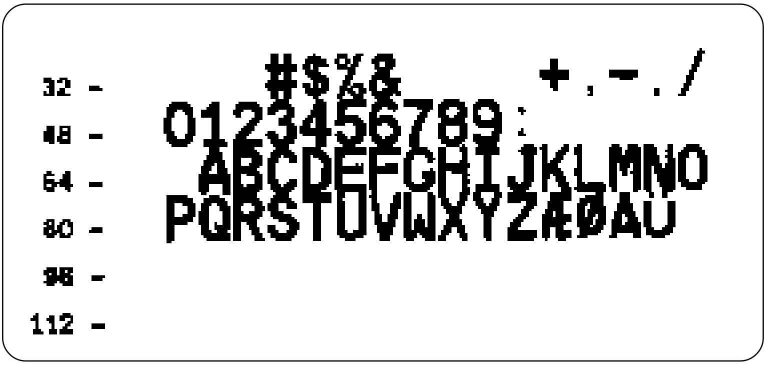
```
 32 -      #$%&    +,-./
 48 -  0123456789:
 64 -   ABCDEFGHIJKLMNO
 80 -  PQRSTUVWXYZÄÖÜ
 96 -
112 -
```

Size 5 (7 bit); French



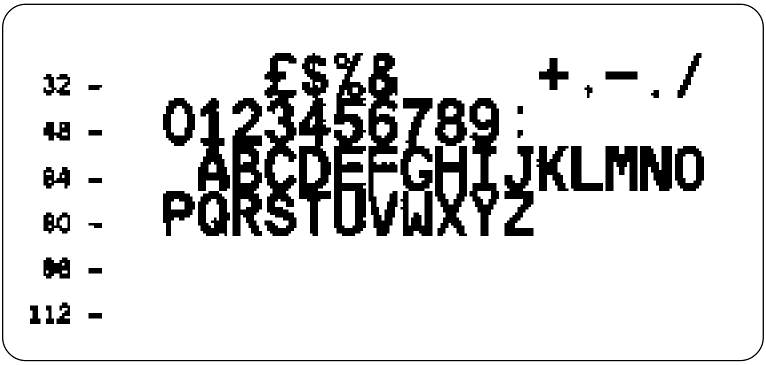
```
 32 -      £$%&      +,-./
 48 -   0123456789:
 64 -    ABCDEFGHIJKLMNO
 80 -   PQRSTUVWXYZ
 96 -
112 -
```

Size 5 (7 bit); Danish



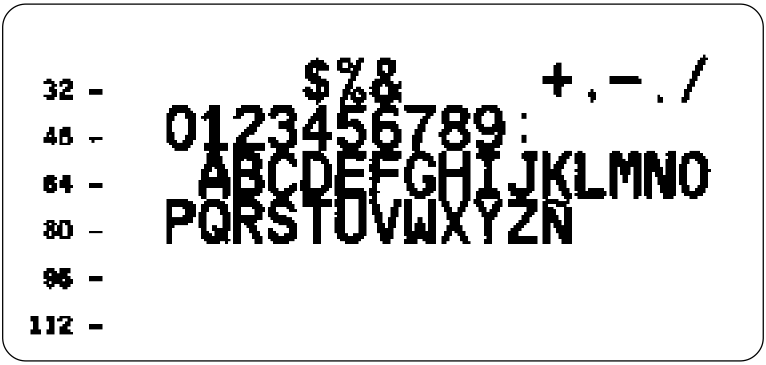
```
 32 -      #$%&      +,-./
 48 -   0123456789:
 64 -    ABCDEFGHIJKLMNO
 80 -   PQRSTUVWXYZÆØÅÜ
 96 -
112 -
```

Size 5 (7 bit); Italian

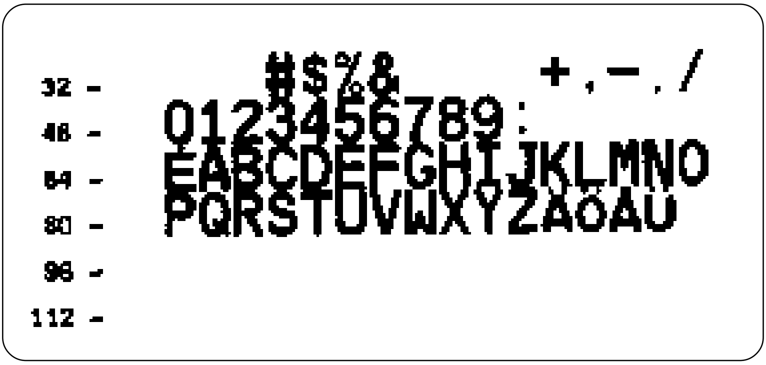


```
 32 -      £$%&      +,-./
 48 -   0123456789:
 64 -    ABCDEFGHIJKLMNO
 80 -   PQRSTUVWXYZ
 96 -
112 -
```

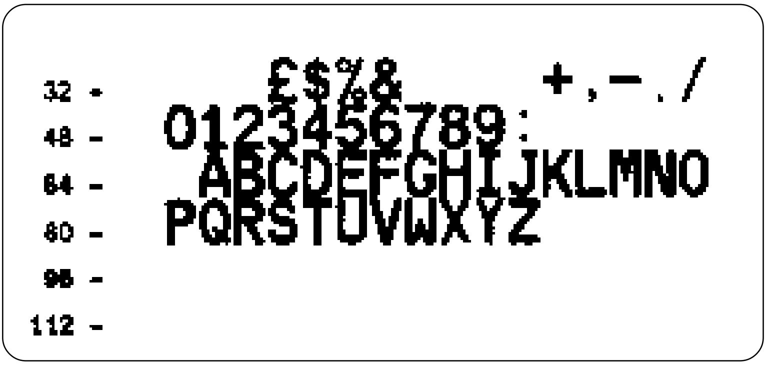
Size 5 (7 bit); Spanish



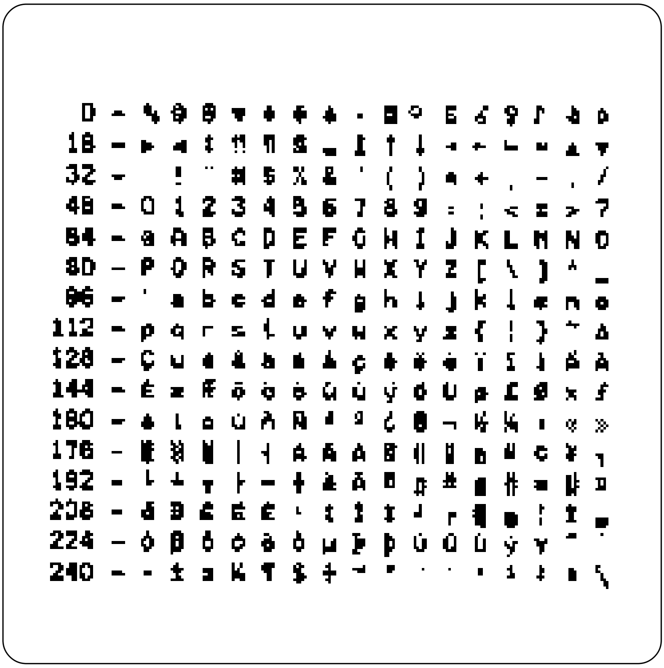
Size 5 (7 bit); Swedish



Size 5 (7 bit); Swiss



Size 5 (7 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 ■ ↘
```