ESC Sequences

Programming Reference

for

Allen Coding – NGT-Series

Copyright

Disclaimer

The company Allen Coding GmbH undertakes all measures to ensure that the published document is up to date at the time of publication. However no guarantee may be given that the document is error-free. The Allen Coding GmbH does not accept liability for possible damages that may have been caused by faulty information in this document. The company Allen Coding GmbH reserves the right to change the present document at any time without prior notice and to alter its content in regards to continuous product improvement.

Manufacturer

Allen Coding GmbH
Friedrich-Bergius-Ring 30
D-97076 Würzburg
Direct +49 931 250 76 – 0
Fax +49 931 250 76 – 50
Email: info@AllenCoding.com
Website: www.AllenCoding.com

Index

Introduction

This document describes commands used with the NGT range of thermal printers of Allen Coding with firmware version N5.0-D and newer.

This guide is intended for programmers who are familiar with the functionality of the NGT range!

Conventions used

In the escape sequences shown: Spaces and curly brackets are included for clarity only. All non-printable ASCII characters are indicated by the ASCII number in hexadecimal notation enclosed in normal brackets (with the exception of the escape character itself, which is denoted by the word ESC or <ESC>). The pipe symbol is used to indicate alternative forms for a parameter where a choice exists.

Where positional information is given: A top left origin is assumed, vertical measurements are made to the top of the field (with the exception of text fields which require the baseline of the character string instead).

All units of measurement are assumed to be 0.083mm (1/12mm) unless otherwise stated.

Normal Operations

There are two capabilities for sending “field parameter” and “print control parameters” from a control unit (PC, PLC, …) to the printer:

Via serial connection (RS232) or Ethernet connection (TCP/IP) -> see also NGT manual!

Items to be printed may include barcodes, text, graphics, lines and boxes. Each item is defined using the relevant “field parameter” escape sequence. These are sent as part of the “print format” escape sequence.

The set of “print control parameters”, which define how the print is to be made, is sent separately if required.

After the format has been sent, the “print quantity” escape sequence must be sent in order to initiate printing when the product sensor is triggered. The “format ready” LED will be lit when a valid format exists in memory and the print quantity is set. The “CPU ready” LED extinguishes while a print or a transmission is being made to indicate the printer is busy. Once the task has been made, the LED will light again to indicate the end of the task. If more prints are required, only the “print quantity” escape sequence must be sent again, as the existing format will remain resident in memory until a new one is sent (see also "Set Memory Mode").

Message header

A message header is considered mandatory for all message exchange. Each escape sequence, generally starting with the <ESC> character, is preceded by the following header.

The message header has a fixed length of 8 bytes.

Message start magic byte	**Acknowledge request flag**	**Message size bytes 1-6**
Offset 0	**Offset 1**	**Offset 2-7**
'~' (0x7E)	**'0' (0x30) or '1' (0x31)**	**ASCII numerals from '0' – '9'**

Message start magic byte is a constant value. It marks the beginning of a new message.

Acknowledge request flag can be set if an acknowledge response from the printer in the form of a "<ESC>ZOK" Message is desired following successful processing of a message that would not otherwise return a response. User drivers are not required to implement processing of this flag. Messages originating from the printer do not set this flag.

Message size is an ASCII-formatted decimal number equal to the byte size of the payload following the header.

Notes:

When the acknowledge request flag is set, the printer responds as it normally would to the message. Only if the issued command would not normally return a response, a response in the form of "~0000004<ESC>ZOK" is sent back. If the issued command causes an error condition, the printer does not necessarily (but may in special cases) return a "ZOK" message. A corresponding error response is returned instead, or in addition.

The size field of the header allows for a maximum message size of 999999 bytes. In the case of large format dimensions, especially for 8 Inch models, the size of a "Postview" packet can exceed this limit by one or two orders of magnitude. The overflow behavior is documented with the Postview command. Similarly, in the case of large formats, especially those containing lots of image data, the size of a format definition can exceed this limit. The overflow behavior is documented with the "Send format" and "Receive format" commands.

Format definitions generally consist of a range of <ESC> commands defining the format followed by one or up to three additional commands – Quantity (required), Density and Speed (optional). In these cases, the additional optional messages (Density and Speed) are processed and discarded – they are not stored as part of the format. The print quantity is stored as part of the format and set when the format is loaded from memory.

Only these messages, when transferred as part of a format (“<ESC>F”) message may be transferred within the same packet. It is not generally possibly to transfer several messages in the same packet.

Examples

Example 1:

~0000005<ESC>S120

~ – message start indicator
0 – no acknowledge receipt is requested
000005 – message payload length of 5 bytes
<ESC>S120 – message payload: set the printing speed to 120 mm/s

No printer response is expected or received.

Example 2:

~1000007<ESC>X36180

~ – message start indicator
1 – acknowledge receipt is requested
000007 – message payload length of 5 bytes
<ESC>X36180 – message payload: extended command 36, set slot number to 180

A printer response (acknowledge receipt) is expected and received:

~0000004<ESC>ZOK

Example 3:

```
~1000004<ESC>R46
```

~ – message start indicator

1 – acknowledge receipt is requested

000004 – message payload length of 4 bytes

<ESC>R46 – message request the main firmware version string

A printer response is expected and received:

```
~0000014<ESC>ZOKN7.2.07V-D
```

Since this command does return data, no additional ZOK acknowledge receipt is expected or received.

Example 4:

```
~1000054<ESC>FTESTP00001<ESC>#T01000400004000FTestPrint(0x0B)12L000(0xFF)<ESC>Q00000
```

~ – message start indicator

1 – an acknowledge receipt is requested

000054 – message payload length of 54 bytes

<ESC>FTESTP00001

– a format definition upload of a format with the name TESTP00001 is initiated.

<ESC>#T01000400004000FTestPrint(0x0B)12L000

– the format contains a text field with the fixed string content “TestPrint”. The string content is terminated by ASCII 0x0B (VT).Additional text formatting information follows.

(0xFF)

– the format definition is terminated with 0xFF.

<ESC>Q00000

– a regular “Set Print Quantity” command is included in the format definition. (Unlimited Quantity)

A printer response (acknowledge receipt) is expected and received:

~0000004<ESC>ZOK

Example 5:

~1000054<ESC>FTESTP00001<ESC#T01000010000100FTestPrint(0x0B)12L000 (0xFF)<ESC>Q00000

~ – message start indicator

1 – an acknowledge receipt is requested

000054 – message payload length of 54 bytes

<ESC>FTESTP00001

– a format definition upload of a format with the name TESTP00001 is initiated.

<ESC#T01000010000100FTestPrint(0x0B)12L000

– As Example 4, but the text field definition has been manipulated to produce an error in the format. As text field coordinates are counted from the font baseline, the positional parameter results in a negative coordinate, which causes an error.

(0xFF)

– the format definition is terminated with 0xFF.

<ESC>Q00000

– a regular “Set Print Quantity” command is included in the format definition. (Unlimited Quantity)

An error response is received, no additional ZOK acknowledge receipt is expected or received.

~0000004<ESC>ZER

~0000007<ESC>ZERROR06

Print format

ESC F {name} {escape sequences} (FF)

{name}	=	Name of the format. This parameter must be exactly ten characters long and may include all printable ASCII characters.
{escape sequences}	=	Field parameter escape sequences.
(FF)	=	

The terminating character, required as the length of the format is variable.

The print control parameters must be sent in a certain order (described in the relevant section of this document), and must be after all field parameters. The field parameters however, maybe sent in any order, with any field type being used more than once where necessary.

Once the print format is saved to the printer this becomes your label template, if you have more fields that do not need to be updated for every print you can just update the variable fields. To do this you will need to send an esc code (“ESC X56”) followed by the variable field you wish to update. The benefit of this is speed. If your label template is very big then it would take a longer time to update the printer for the next print job.

Field Parameters

The field parameters are split into parameters containing data (called content elements) and parameters containing representation (called representation elements).

Content elements are counters, input variables, formulas, dates and fixed text. Without using them inside any representation element, they won't be visible inside the printed format.
Content elements currently consist of various types of barcodes and also plain text fields. The representation elements can contain content elements or fixed data.

All representation elements ("ESC #T", „ESC #B", "ESC G" and "ESC L") must be placed within the print area or the appropriate "field placed outside of print area" error will occur. Note that this may happen also when changing some content elements later, since they may affect the size of representation elements (more data makes a barcode longer, which then can exceed the print area) or even when the content of e.g. a counter changes during printing.

Content Data

Certain parameters are labeled to contain content data. For text, this data will be interpreted as Windows-1252 encoded. It may consist of every byte value (0-255) excluding the special characters with hexadecimal numbers 0x0B (VT, used to terminate content data), 0x1B (ESC, used to start printer escape sequences) and 0x5C (backslash '\', used to enter escape mode).

Escape mode can be used to produce special symbols or the reserved characters. After entering escape mode, one of the available escape sequences must follow. Escape mode is left on end of an escape sequence ('>' most of the time). Some of the special symbols are only displayable in text, respectively barcodes. Further it is important to note, that a sequence like "\<FNC1>" doesn't count as 7 characters when applying padding, but one. A description of each symbol is given in the following table.

For example the content "10,53\<UNIC(0x20AC)>/kg" with 0x20AC being the Unicode number for the Euro currency symbol results in the string "10,53€/kg" when printed as text.

Escape sequence	*Description*
\\	Produces backslash
\<ESC>	Produces ASCII escape character (0x1B)
\<VT>	Produces ASCII vt character (0x0B)
\<ASC({nnn})>	Produce arbitrary byte values, which will be interpreted as Windows-1252, argument given as decimal number
\<UNIC({n…n})>	Produce arbitrary Unicode symbols; this is only applicable in text fields and not in barcodes; n…n must be in hexadecimal format with optional leading “0x”; e.g. \<UNIC(0x20AC)> to produce Euro sign: €

Special control codes for barcodes

These codes are valid for all barcodes though they might be ignored if the selected symbology doesn’t support their function. It is recommended to use the barcode specific versions to avoid mistakes.

Escape sequence	*Description*
\< FNC1>	Special function symbol (FNC1)
\< ECI({nnnnnn})>	Extended Channel Interpretation: nnnnnn: values: 000000-999999; channel number
\< RP>	Reader Programming

Special control codes specific to Code128 barcodes

Escape sequence	***Description***
\<C128_FNC1>	Special function symbol (FNC1)
\<C128_FNC2>	Special function symbol (FNC2)
\<C128_FNC3>	Special function symbol (FNC3)
\<C128_FNC4>	Special function symbol (FNC4)
\<C128_START_A>	Start C128 symbol with code A
\<C128_START_B>	Start C128 symbol with code B
\<C128_START_C>	Start C128 symbol with code C
\<C128_A>	Select encoding A for C128 barcode
\<C128_B>	Select encoding B for C128 barcode
\<C128_C>	Select encoding C for C128 barcode
\<C128_SHIFT>	SHIFT symbol for C128 barcode

Special control codes specific to Datamatrix barcodes

Escape sequence	***Description***
\<DMX_FNC1>	Special function symbol (FNC1)
\<DMX_RP>	Special function symbol for barcodes (Reader Programming)
\<DMX_MCR05>	Special function symbol for barcodes (05 Macro)
\<DMX_MCR06>	Special function symbol for barcodes (06 Macro)
\<DMX_SA({pp}{nn}{fff}{ggg})>	Structured append: pp: values: 01-16; symbol position nn: values: 02-16; total number of symbols fff: values: 001-254; file id 1 ggg: values: 001-254; file id 2
\<DMX_ECI({nnnnnn})>	Extended Channel Interpretation: nnnnnn: values: 000000-999999; channel number

Variable Field - Input: (content element)

ESC VI {N..N} (0x0B) {a..a} (0x0B) {q..q} (0x0B) {p} {f} {llll} {yyyy} {g..g} (0x0B) {i..i} (0x0B)

{N..N} = Name of this variable field (alphanumeric)

{a..a} = content data

{q..q} = Prompt string (alphanumeric)

{p} = Pad character requirement

0 = Assert min length
L = pad to min length on left side of the content
R = pad to min length on right side of the content

{f} = Pad character

{llll} = Min. length of the content

{yyyy} = Max. length of the content

{g..g} = Prefix (max. 40 character of content data)

{i..i} = Suffix (max. 40 character of content data)

(0x0B) = Terminator character

Notes:

- If “Max. length of the content “ = “0000”, then the content length determines the maximum length allowed for the variable Information.

Variable Field - Counter: (content element)

ESC VC {N..N} (0x0B) {S|T|R} {aaaaaaaaa} { bbbbbbbbb} { ccccccccc} {ddddddddd} {eeeeeeeee} {j..j} (0x0B) {p} {f} {llll} {g..g} (0x0B) {i..i} (0x0B)

{N..N} = Name of this variable field (alphanumeric)

{S|T|R} = Sequential information. (S: Normal, T: Trigger, R: Reset)

{aaaaaaaaa} = Start (-99999999 to 999999999).

{bbbbbbbbb} = Stop (-99999999 to 999999999).

{ccccccccc} = Step (-99999999 to 099999999).

{ddddddddd} = Duration (0000001 to 999999999).

{eeeeeeeee} = Reset sequential number to.

{j..j} = Base (User defined)

“0123456789” or empty = decimal
“01” = binary
“ABCD...YZ” = alpha
...

{p} = Pad character requirement

0 = Assert min length

L = pad to min length on left side of the content

R = pad to min length on right side of the content

{f} = Pad character

{llll} = Pad to a total of “llll” characters

{g..g} = Prefix (max. 40 character of content data)

{i..i} = Suffix (max. 40 character of content data)

(0x0B) = Terminator character.

Notes:

- If a negative number is sent for the Start-, Stop, and/or Step-value, either of the following forms is acceptable: “-000099“or “0000-99“.

Variable Field - Date: (content element)

ESC VD {N..N} (0x0B) {data} (0x0B) {{{F} {ddd} {mmm} {yyy}} | {{VI | VX} {N..N} (0x0B) {t}}

{N..N} = Name of this variable field (alphanumeric)

{data } Items in the field are defined by ASCII 2 digits number

Day = 01

Day without leading zero = 02

Week of year, ISO8601,leading zeros = 03

Day of week, ISO8601 = 04

Numeric Month = 05

Numeric Month without leading zero = 06

Alpha Month = 07

2 digits year = 08

2 digits year without leading zero = 09

4 digits year = 10

Last digit of year = 11

European Julian Date = 12

USA Julian Date = 13

Hours = 14

Minutes = 15

Seconds = 16

Text = 17{a....a}(0x03)

Week of Year, ISO 8601, no leading zeros = 18

Day of week, Sunday is 1 = 19

Week of Year, US rule = 20

Week of Year, US rule, leading zeros = 21

Reserved = 22…49

Day Code = 50???????

where ?s are codes for Monday – Sunday (ASCII 20h-5Ah)

Day Of Month Code = 51???????????????????????????????

where the ?s are the codes for each day number from 1 to 31 (ASCII 20h - 5Ah)

Month Code = 52????????????

where ?s are the codes for January – December(ASCII 20h - 5Ah)

Year Code = 53??????????

where ?s are the codes for the last digit of the year i.e. replacing 0 to 9 (ASCII 20h - 5Ah)

Day of week Code extender = 54{a..a}(0x03){b..b}(0x03){c..c}(0x03){d..d}…(0x03)

Where (0x03) is the separator character and each sequence (always 7-sequences) is for Monday – Sunday

Day of month Code extender = 55{a..a}(0x03){b..b}(0x03){c..c}(0x03){d..d}…(030x03

Where (0x03) is the separator character and each sequence (always 31-sequences) is for each day month from 1 to 31

Month Code extender = 56{a..a}(0x03){b..b}(0x03){c..c}(0x03){d..d}…(0x03)

Where (0x03) is the separator character and each sequence (always 12-sequences) is for January – December

Year Code extender = 57{a..a}(0x03){b..b}(0x03){c..c}(0x03){d..d}…(0x03)

Where (0x03) is the separator character and each sequence (always 20-sequences is for 2000-2020

12 hour format = 58

AM/PM (Alpha) = 59x

If selected, “AM” or “PM” will be printed
x=0: lower case, x=1: upper case

{F} = Fixed information for the offset value.

{ddd} = -99 - 999 number of advanced days

{mmm} = -99 - 999 number of advanced months

{yyy} = -99 - 999 number of advanced years

{VI} = Variable Field – Input information for the offset value

{VX} = Variable Field – Fixed information for the offset value

{N..N} = Name of the variable field (alphanumeric)

{t} = Interpretation of the content from VI {N..N} or VX {N..N}:

0 = number of advanced days
1 = number of advanced months
2 = number of advanced years

(0x0B) = Terminator character

Variable Field - Formula: (content element)

ESC VF {N..N} (0x0B) {dd} {e} {p} {f} {llll} {formula} (0x0B)

{N..N} = Name of this variable field (alphanumeric)

{dd} = Number of fractional digits; note that this parameter might affect precision of intermediate results

{e} = Decimal separator

{p} = Pad character requirement

0 = Assert min length
L = pad to min length on left side of the content
R = pad to min length on right side of the content

{f} = Pad character

{llll} = Pad up to "llll" characters

{formula} = Can include fixed data, type and name of variable fields, and control codes. Between the fixed data field type and name of variable fields, and control codes must be placed the operators. If no operator is given, then the "append" operator is used.

(0x0B) = Terminator character

Structure samples of a formula :

{operand} ({operator} {operand}) ({operator} {operand}) …

Where operand can be any of:

F{n..n}(0x0B) = fixed data; n..n is content data

VI{N..N}(0x0B) = use an arbitrary content element
VC{N..N}(0x0B)
VD{N..N}(0x0B)
VF{N..N}(0x0B)
VX{N..N}(0x0B)

#MOD10({operand}) = checksum, GS-1 modulo 10

And operator can be any of:

& = append string
+ = addition
- = subtraction
* = multiplication
/ = division

Notes:

- in a formula, values get converted from string to number and vice versa if needed. For arithmetic operations, numbers are needed and for string operations (like concatenation) and for the final result, a string is needed. Conversion from string to number might fail at runtime if the concerned object isn't the string description of a number. Conversion from number to string adheres to the rules (precision, padding, etc.) given for the formula object inside which the conversion happens.
- number format: leading zeros are allowed; sign (minus) may follow or lead the leading zeros; decimal separator must be either period or comma;
- setting the decimal separator for the formula only affects the output. Note that the separator of a formula must be set to either the period or the comma if the output shall be fed into another formula object
- the precedence of the operators in decreasing order is (* and /), (+ and -), &; for equal operators, they are assumed to be left-associative;

e.g.:

A&B+C = A&(B+C)
A+B*C = A+(B*C)
A-B-C-D = (((A-B)-C)-D)
A*B/C&E+F = ((A*B)/C)&(E+F)

Example:

```
<ESC>VFprice_string(0x0B)02.000000FPrice: (0x0B)&VIweight(0x0B)*F0.50(0x0B)&F€(0x0B)(0x0B)
```

The previous formula, together with input variable “weight” set to “1.5” would result in the following text string

“Price: 0.75€”

and with “weight” set to 2 (with setting number of fractional digits to two)

“Price: 1.00€”.

Variable Field - Free: (content element)

ESC VX {N..N} (0x0B) {a..a} (0x0B)

{N..N} = Name of this variable field (alphanumeric)

{a..a} = content data

(0x0B) = Terminator character.

Text field: (representation element)

ESC #T {ff} {xxxxx} {yyyyy} {o} {r} {{F {n..n}} | {{VI | VC | VD | VF | VX} {N..N}} (0x0B)} {ss} {t} {kkk}

{ff} = Font:

00 = OCR-B
01 = Gill Bold
02 = Helvetica
03 = Century Gothic
04 = Arial Bold
05 = Arial
06 = OCR A
07-50 = User definable

{xxxxx} = Horizontal position

{yyyyy} = Vertical position

{o} = Orientation

0 = Normal
1 = 90o
2 = 180o
3 = 270o

{r} = Reverse outline option:

0 = Off
1 = On

{ss} = Scaling factor (point size).

{t} = Alignment
L= left
R= right
C= centered

{kkk} = kerning modification:
additional space between characters in dots (1/12 mm)

{F} = Fixed information.

{n..n} = content data

{VI} = Variable Field - Input

{VC} = Variable Field - Counter

{VD} = Variable Field - Date

{VF} = Variable Field - Formula

{VX} = Variable Field – Free (Constant)

{N..N} = Name of the variable field (alphanumeric)

(0x0B) = Terminator character.

Barcode field: (representation element)

For some barcodes, the option of printing the content in human readable form exists. This prints the human readable representation in a certain position (usually below the barcode) and with a certain font. If a different representation is desired (different font, above the barcode, etc.) then it shall be turned off and replaced by a text field.

ESC #B {tt} {xxxxx} {yyyyy} {o} {r} {{F {n..n}} | {{VI | VC | VD | VF | VX} {N..N}} (0x0B)} {"code specific parameter"}

{tt} = Type (symbology):

00 = EAN-8
01 = EAN-13
02 = 2/5 Interleaved
03 = CODE39
04 = CODE128
05 = UPC_A
06 = UPC_E
07 = Data Matrix
08 = PDF417
09 = DataBar *(new since N8.1.00-R)*
10 = QR Code *(new since N8.1.00-R)*

{xxxxx} = Horizontal position.

{yyyyy} = Vertical position.

{o} = Orientation:

0 = Normal
1 = 90°
2 = 180°
3 = 270°

{r} = Reverse outline option:

0 = Off
1 = On

{F} = Fixed information

{n..n} = content data

{VI} = Variable Field - Input

{VC} = Variable Field - Counter

{VD} = Variable Field - Date

{VF} = Variable Field - Formula

{VX} = Variable Field – Free (Constant)

{N..N} = Name of the used variable field (alphanumeric)

(0x0B) = Terminator character

Although some types of barcodes have a fixed length, the terminator character should still be included for consistency.

{“code specific parameter”} =

EAN 8, EAN 13, UPC A, UPC E:

{{hhhhh} {bb} {r} {u}}

{hhhhh} = Height (including human readable character).

{bb} = Widths of the thinnest bar and space element.

{r} = Human readable option:

0 = Off
1 = On

{u} = GS1 conformity checking:

0 = Off
1 = On

Code 39 / 2/5 Interleaved:

{{hhhhh} {nn} {ww} {r} {c}}

{hhhhh} = Height (including human readable character).

{nn} = Widths of narrow bar and space element

{ww} = Width of the wide element

{r} = Human readable option:

0 = Off
1 = On

{c} = Checksum:

0 = None
1 = In barcode and human readable line

ITF 14 (not implemented yet):

ITF-14 is subset of code Interleaved 2/5 used by the GS-1. Its length is fixed to 14 characters. To improve readability, bars surrounding the barcode (called bearer bars) are mandatory above and below the symbol. The optional bars to the left and to the right are always printed by the printer. The width of the bars matches the minimum width of twice the module size (width of narrow element).

{{hhhhh} {nn} {ww} {r} {u}}

{hhhhh} = Height (including human readable character).

{nn} = Widths of narrow bar and space element

{ww} = Width of the wide element

{r} = Human readable option:

0 = Off
1 = On

{u} = GS1 conformity checking:

0 = Off
1 = On

Code 128:

{{hhhhh} {bb} {r} {u} {dd}}

{hhhhh} = Height (including human readable character).

{bb} = Widths of the thinnest bar and space element.

{r} = Human readable option:

0 = Off
1 = On

{u} = GS1 conformity checking:

0 = Off
1 = On

{dd} = Encoding type:

00= Auto
01= Code A
02= Code B
03= Code C

GS1 128:

{{hhhhh} {bb} {r} {u} {dd}}

{hhhhh} = Height (including human readable character).

{bb} = Widths of the thinnest bar and space element.

{r} = Human readable option:

0 = Off
1 = On

{u} = GS1 conformity checking:

0 = Off
1 = On (without AI-checking)

{dd} = Encoding type:

00= Auto
01= Code A
02= Code B
03= Code C

Datamatrix:

{{mm} {ccc} {rrr} {eee} {s} {u}}

{mm} = Module size

{ccc} = Number of columns

{rrr} = Number of rows

{eee} = Error Checking and Correction Algorithm

{s} = Character set:

1 = field data is numeric + space (0..9,”) – no \&’’

2 = field data is uppercase alphanumeric + space (A..Z,’’) – no \&’’

3 = field data is uppercase alphanumeric + space, period, comma, dash, and slash (0..9,A..Z,“.-/”)

4 = field data is upper-case alphanumeric + space (0..9,A..Z,’’) – no \&’’

5 = field data is full 128 ASCII 7-bit set

6 = field data is full 256 ISO 8-bit set

{u} = GS1 conformity checking:

0 = Off
1 = On (without AI-checking)

PDF 417: (not implemented yet):

{{ww} {zz} ccc} {rrr} {l}}

{ww} = Module width

{zz} = Module height

{ccc} = Number of columns

{rrr} = Number of rows

{l} = security level

Databar *(new since N8.1.00-R)*:

{{zzzz} {hhhhh} {nn} {rrrrr} {ss} {pp}}

{zzzz} = Symbology:

0029 = Omnidirectional/Truncated
0030 = Limited
0031 = Expanded
0079 = Stacked
0080 = Stacked omnidirectional
0081 = Expanded stacked

{hhhhh} = Height (<u>excluding</u> human readable line)

{nn} = Widths of narrow bar and space element

{rrrrr} = Height of the human readable line:

00000 = No human readable line

For all stacked codes the human readable line is switched off!

{pp} = Segments per row (just for "expanded stacked" type)

00 = Not used!

QR Code *(new since N8.1.00-R)*:

{{ss} {mm} {hhhhh} {e} {i} {u}}

{ss} = QRCode size :

01 = 21x21 (also automatic)
02 = 25x25
03 = 29x29
04 = 33x33
05 = 37x37
06 = 41x41
07 = 45x45
08 = 49x49
09 = 53x53
10 = 57x57
11 = 61x61
12 = 65x65
13 = 69x69
14 = 73x73
15 = 77x77
16 = 81x81
17 = 85x85
18 = 89x89
19 = 93x93
20 = 97x97
21 = 101x101
22 = 105x105
23 = 109x109
24 = 113x113
25 = 117x117
26 = 121x121
27 = 125x125
28 = 129x129
29 = 133x133
30 = 137x137
31 = 141x141
32 = 145x145
33 = 149x149
34 = 153x153
35 = 157x157

36 = 161x161
37 = 165x165
38 = 169x169
39 = 173x173
40 = 177x177

{mm} = Min. module size (1/12 mm; If the min. symbol size is not reached with this value, this value is automatically increased!)

{hhhhh} = Min. symbol size (1/12 mm; If the min. module size is not reached with this value, this value is automatically increased!)
! Currently without function !
00000 = auto size

{e} = Error Checking and Correction Algorithm

1 = L
2 = M
3 = Q
4 = H

{i} = InputMode

0 = data (Uses full ASCII range interpreted as Latin-1 or binary data)
1 = Unicode (Uses pre-formatted UTF-8 input)
! Currently without function !
2 = GS1 (Encodes GS1 data using FNC1 characters)

{u} = GS1 conformity checking:

0 = Off
1 = On (without AI-checking)
! Currently without function !

Graphics field: (representation element)

ESC G {xxxxx} {yyyyy} {wwwww} {hhhhh} {s} {d..d}

{xxxxx}	=	Horizontal position.
{yyyyy}	=	Vertical position.
{wwwww}	=	Width (bytes).
{hhhhh}	=	Height (bits).
{s}	=	Scaling factor: 0 = 1:1 (normal dimensions).
{d..d}	=	Content (ASCII data).

A terminator character is not used, as the content may cover the full range of eight bit ASCII data from 0 to 255. In this case, the number of bytes expected for the contents is determined by the width multiplied by the height, so no terminator is required.

The width and height are also used to determine how the data should be translated into pictorial information. This is achieved in the following way:

The data forms a bitmapped image, with the first byte describing the top left portion of the image. The following data continues to describe the image horizontally until the byte count matches the stated width. At this point, one complete row will have been defined. The process starts again from the leftmost position of the next row starting with the next byte in sequence. This process continues until both the width and height have been exceeded.

Line / box field: (representation element)

ESC L {xxxxx} {yyyyy} {wwwww} {hhhhh} {vvvvv} {rrrrr}

{xxxxx} = Horizontal position.

{yyyyy} = Vertical position.

{wwwww} = Width.

{hhhhh} = Height.

{vvvvv} = Vertical thickness.

{rrrrr} = Horizontal thickness.

Together these parameters may describe vertical lines, horizontal lines, filled blocks and boxes with sides of different widths. No terminator character is required as the data is of fixed length.

Lines or filled blocks are described by supplying width and height parameters as required and setting both thickness parameters to zero. Boxes are described by supplying all parameters, where the thickness parameters set the thickness of the vertical and horizontal sides of the box respectively.

Print control parameters

Special Setting Parameters

These parameters may be sent once as part of the format (ESC S and ESC X07 in front of the terminating character “FF”; ESC Q after the terminating character “FF”), or as many times as required when sent independently.

Set Print speed: Intermittent specific

ESC S {nnn} Where {nnn} = 100 to 600 mm/sec (dependent on printhead), in steps of 1

Set Print density as percent:

ESC X07 {nnn} Where {nnn} = 000 to 100 %, in steps of 1

Set Print quantity:

ESC Q {nnnnn} Where {nnnnn} = 00000 to 99999.
00000 set it to an unlimited number of prints.
-0000 set it to a real zero.

For release a format the ESC Qxxxxx is needed at the end of the format.

Standard Setting Parameters

These parameters can be sent independently as many times as required.

Set Printhead Resistance:

ESC P {nnnn} Where {nnnn} = 1000 to 1500 ohms, in steps of 1

This parameter defines the printhead average resistance in ohms. It limits the maximum power in order to protect the thermal printhead!

Set Ribbon length:

ESC Z L {xxxx} Where {xxxx} = 0100 to 1000 meters.

Sets the length of the ribbon roll to be used in the cartridge.

Set Warning length:

ESC Z W {xx} Where {xx} = 05 to 25 meters.

Sets the length at which the ribbon left in the cartridge will trigger a ribbon low error.

Set Date Language:

ESC Z U {x} Where {x}
1= England,
2 = France,
3 = Sweden,
4 = Germany

5 = Italy,
6 = Spain,
7 = Holland.

This command selects the language for the abbreviated month names. Every abbreviation consists of three capital letters. Please refer to the following table.

Month	***1 England***	***2 France***	***3 Sweden***	***4 Germany***	***5 Italy***	***6 Spain***	***7 Holland***
January	JAN	JAN	JAN	JAN	GEN	ENE	**JAN**
February	FEB	FEV	FEB	FEB	FEB	FEB	**FEB**
March	MAR	MAR	MAR	MAR	MAR	MAR	**MAA**
April	APR	AVR	APR	APR	APR	ABR	**APR**
May	MAY	MAI	MAJ	MAI	MAG	MAY	**MEI**
June	JUN	JUN	JUN	JUN	GIU	JUN	**JUN**
July	JUL	JUL	JUL	JUL	LUG	JUL	**JUL**
August	AUG	AOU	AUG	AUG	AGO	AGO	**AUG**
September	SEP	SEP	SEP	SEP	SET	SEP	**SEP**
October	OCT	OCT	OKT	OKT	OTT	OCT	**OKT**
November	NOV	NOV	NOV	NOV	NOV	NOV	**NOV**
December	DEC	DEC	DEC	DEZ	DIC	DIC	DEC

Set Printhead Serial Number:

ESC Z N {xxxx} Where {xxxx} = 0000 to 9999

This parameter allows the operator to enter the printhead serial number.

Reset printhead maintenance count:

ESC Z M

Reset print count:

ESC Z Q

Set minimum print speed:

ESC Z I S {xxx}	Where {xxx}	= 050 to MAXIMUM (mm/sec)

Set Print Delay:

ESC Z G {xxxx}	Where {xxxx}	In IM-mode: 0000 to 5000 millisecond (ms) in increments of 1 In CM-mode: 0000 to 0250 millimeter (mm) in increments of 1 *(new since N7.3.00-R and MES004.6-D – before: ms – like IM mode)*

Set Print Pulse:

ESC Z P {x}	Where {x}	0= low (falling edge), 1= high (rising edge)

This parameter set the used edge of the incoming print signal.

Set Image Orientation:

ESC Z F {x}	Where {x}	1= normal 2= mirror 3= normal 180 degrees turned 4= mirror 180 degrees turned

This parameter is used to set the image orientation in print area memory

Additional Settings

Key:

ni, **i={0, 1, 2, …}** – digit (ASCII 0x30 – 0x39)
ni(x) x-digits (if the length is not defined: x = various)

ci, **i={0, 1, 2, …}** – ASCII character (0x20 – 0xFF)
ci(x) a string of length x (if the length is not defined: x = various)

[s](x) square brackets structured groups ()
x is the total length of the group (if the length is not defined: x = various)

(0x00) – hexadecimal value

Terminal sets **ESC X Command Data**, printer sends an ***error string*** if the data is wrong

Command	Data	Description
ESC X00	n0(6)	**Set time** n0=hhmmss
ESC X01	n0(9)	**Set date** n0=DDMMYYYYW (W=Day of the week)

Command	Data	Description
ESC X07	n0(3)	**Set density as a percent** n0=000 – 100 degrees
ESC X08	n0(1)	**Print trigger level,** 0=low, 1=high
ESC X09	n0(1)	**Auto print** 0=off, 1=on
ESC X10	n0(1)	**Set the NGT processor temperature control** 0 = disable, 1 = enable
ESC X11	n0(2)n1(2)	**Set the NGT+ processor temperature threshold** (in °C) n0: lower threshold value n1: upper threshold value
ESC X12		**Trigger a print**
ESC X13		**Delete resident format**
ESC X14	n0(1)	**Set print signal buffer** 0=enable, 1=disable
ESC X15	n0(3)	**Set the host machine maximal speed** (mm/s) *(new since N7.3.00-R and MES004.6-D)* For speed compensation in CM - mode
ESC X16	n0(1)	**Set inhibit level** 0=Low, 1=High
ESC X17	n0(5)	**Set the image horizontal offset (pixel)** *(new since N7.3.01-R)*
ESC X18	n0(3)	**Set the IM mode image vertical offset (pixel)** *(new since N7.3.01-R and MES005.0-R)*

Command	Data	Description
ESC X19	n0(2)	**Set the IM mode post-print delay (ms)** *(new since N7.3.01-R and MES005.0-R)*
ESC X20	n0(1)n1(3)	**Set the ribbon rewind state and offset in pixel (<u>only NGT++</u>)** *(new since N7.3.01-R)* n0= 0: disabled, n0 = 1: enabled n1: rewind offset (min: -99, max: 127)
ESC X21	n0(1)	**Set the ribbon spare mode** *(new since N7.3.01-R and MES005.0-R)* CM: “lower ribbon speed” n0(1) = 0: deactivated, n0(1) = 1: 25%, n0(1) = 2: 50% (**<u>only NGT++</u>**) IM: “Side Step Function (together with multiple areas)” n0(1) = 0: deactivated, n0(1) = 1: activated
ESC X23	n0(1)	**Select communication interface** 0=Serial (RS232), 1=Network (Ethernet)
ESC X24	n0(1)	**General error relay** 0=Active High, 1=Active Low
ESC X36	n0(3)	**Load format** n0: index of the format to load.
ESC X38	n0(1)	**Set memory mode** 0: standard 1: PC-mode 2: volatile mode When printer is in volatile mode, only one format slot is available. Formats in this slot won’t be saved if printer is turned off. In PC-mode the transmitted formats every time are saved in slot 0. In standard mode the transmitted format are saved in the currently selected slot.

Command	Data	Description
ESC X41	n0(2)	**Set preheater temperature** (in °C) 01 - 60 degrees
ESC X42	n0(3)	**Delete formats** 000=All Formats, 001=Format 1, 002=Format 2, 003=Format 3, …
ESC X43		**Find homesensor**

ESC X44	n0(1)n1(4)	**Motor steps (forward/reverse)**
		n0=1: forward n0=0: reverse
		n1=0001-1200pixels

ESC X45	n0(1)	**Set intermittent/continuous mode** (new printer start muss be done) 1=intermittent, 0=continuous
ESC X47	n0(1)n1(1)	**Send "ready"-string after a format-transmission or after a print** n0=0- All ready-strings off n0=1- All ready strings on n0=2- ready-string for format transmission n0=3- ready-string for print n0=4- ready-string for serialization low buffer n1=1- enable ready-string, n1=0- disable ready-string After a print, ESC RD1 is sent After receiving a format, ESC RD2 is sent After a print and if serialization buffer is low, ESC RD3 is sent
ESC X48	n0(1)	**Set low foil level** 0=Low, 1=High
ESC X49	n0(1)	**Reset error** 1=reset, x=nothing
ESC X50	n0(5)	**Set connection timeout (network) in millisecond** n0=00100-20000 ms (Default: 6000 ms)

Command	Data	Description
ESC X51	n0(5)	**Set connection timeout (serial) in millisecond** n0=00100-20000 ms (Default: 700 ms)
ESC X52	n0(4)	**Set the max. format length in millimeter** n0: max. length of the format
ESC X53	n0(1)	**Set ribbon type** 0=Wax/Resin, 1=Resin
ESC X54	n0(2)n1(7)n2(7)n3(7)n4(7)n5(7)	**Set counter attributes** n0: number of counter to change n1: start value n2: stop value n3: step n4: duration n5: reset sequential number to
ESC X55	n0(1)	**Announce error with [ESC]ZERROR followed by the error number n1(2) and [ESC]ZERCLR after the pending error has been cleared** 0=deactivate, 1=activate Automatically returns an error code following the error response in the form of [ESC]ZERRORnn, where nn is the two digit error code. Note: Before firmware version N7.3.00-R, [ESC]ZERRORnn responses return with the wrong header length (7) instead of 9.
ESC X56	n0(2)c0(various)(0x0B)c1(various)(0x0B)	**Set content of content element** n0: Type of the variable Field (VI, VC, VX) c0: Name of the variable Field (0x0B): Terminating character c1: VI and VX: Content (alphanumeric) VC: Start value (max 9 digits) (current value)

Command	Data	Description
ESC X57	n0(3)	**Set post-valve delay (thermal bar delay)** In IM-mode: 000 to 999 millisecond (ms) in increments of 1 Default: 000 ms In CM-mode: 000 to 021 millimeter (mm) in increments of 1 *(new since N7.3.03-R and MES004.6-D)* Default: 005 mm Sets the length of a delay between triggering the air-valve and beginning to print (increase the ribbon index!).
ESC X58	n0(1)	**Suppress Print Interrupted Error** When this is set to 1, the “Print Interrupted” error is suppressed. Default is 0.
ESC X59	n0(2)[n1(3) n2(3)](n)	**Set the number and the coordinates of the areas to print.** n0: Number of areas to print n1: Coordinates in mm of an area starting line n2: Coordinates in mm of an area ending line. It is possible to split a format in multiple areas – each of them is printed with one print trigger. The area coordinates are expressed in mm. The starting line of an area must be greater than it´s ending line. An ending line should not exceed the image´s ending line.
ESC X60	n0(1)	**Set the ribbon error state.** n0=0: warning disabled and error enabled. n0=1: warning enabled and error enabled. n0=2: warning disabled and error disabled.
ESC X62	n0(1)	**Set the first week day to either Sunday or Monday** n0=0: first week day is Sunday n0=1: first week day is Monday
ESC X63	n0(3)n1(6)	**Save a format at a given slot number without any data processing** n0: slot number n1: format length
ESC X64		**Reset all SRAM printer settings**

Command	Data	Description
ESC X65	n0(various)(0x0B)n1(various)(0x0B)n2(various)(0x0B)n3(various)(0x0B)	**Set the XPort (Ethernet) basic settings** n0: IP address n1: Subnet mask n2: Gateway address n3: local port
ESC X66	n0(1)n1(3)	**Set hysteresis scale factor** n0=0: no change info requested n0=1: change info requested n1=001-255 (“Hysteresis scale factor”) standard: 51 *(new since N8.1.00-R)*
ESC X67	n0(2)	**Set serialization buffer max elements** n0=00: serialization disabled n0=01-30 maximum elements, standard : 30 If serialization is enabled, serialization content has to be provided for each print *(new since N8.6.02-R)*

ESC X68	n0(2)
	n1(various)(0x0B)[n2(various)(0x0B)](n)
		Append serialization content to buffer
		n0=number of variable elements
		n1=variable name, i.e VIVariable1
		n2=variable contents
		ESC X68 {ff} {{VI | VF | VX} {N..N}} (0x0B)} {N..N} (0x0B)}
		(new since N8.6.02-R)

ESC X69	n0(2) [n1(various) (0x0B)](n)	**Delete serialization content for variable element** n0=number of variable elements n1=variable names, i.e VIVariable1 *(new since N8.6.02-R)*

Command	Data	Description
ESC X70	n0(2)	**Set low buffer RD3 warning** n0=Serialization buffer minimum elements Printer sends RD3 signal when the element number in the serialization buffer is smaller than this value *(new since N8.6.02-R)*
ESC X71	n0(1)	Set buffer underrun enabled 1= printer stops if serialization buffer underrun occurs 0= printer ignores buffer underrun error (new since N8.6.02-R)

Standard Request Parameters:

These take the form: "ESC Z R {x} . . . ", where the "{x}" parameter(s) define the type of action required.

The 'acknowledge' Escape sequence of 'ESC Z OK' will be returned by the printer before any other data is returned.

Note: The number printed in brackets after each parameter indicates the length of the parameter returned.

Request control parameters:

ESC Z R C

The printer will return all ' print control parameters ' associated with the current format to the terminal. Parameters will be sent as a block of data in the following order:

Print speed(3),
Density(2),
Printhead resistance(4)

Request printer status:

ESC Z R S

The printer will return the following details as a block of data:

Format name(10),
Software version(10)
Print quantity(5)
Print count(5),
Ribbon count(4), (i.e. no. meters left),

Print speed(3),
Density(2),
Printhead resistance(4)
Error status(2) (as an error number),
Format Orientation(1)

Notes :

If no format is loaded, 'No Format!' will be returned as the format name.
If continuous print is set, 'CONTI' will be returned as the current print count.

Request maintenance count:

ESC Z R M

The printer will return a six digit value for the total number of meters covered by the print head.

Request Printhead History

ESC Z R I

After the acknowledge, 210 characters are sent to the terminal comprising 15 sets of Print head Resistance (4 digits), Date (6 digits) and Print head Serial Number (4 digits).

Additional Requests:

Key:

ni, i={0, 1, 2, …} – digit (ASCII 0x30 – 0x39)
ni(x) x-digits (if the length is not defined: x = various)

ci, i={0, 1, 2, …} – ASCII character (0x20 – 0xFF)
ci(x) a string of length x (if the length is not defined: x = various)

[s](x) square brackets structured groups ()
x is the total length of the group (if the length is not defined: x = various)

(0x00) – hexadecimal value

Terminal asks **ESC R Command**, Printer sends a ***ESC ZOK*** followed by **Data**

Command	Data (Printer sends)	Description
ESC R00	n0(4)	**Request print head resistance** (Ω)
ESC R02	n0(6)	**Request print count**
ESC R03	c0(10)	**Request date** c0=DD/MM/YYYY
ESC R04	n0(3)	**Request density as a percentage**
ESC R07	n0(4)	**Request ribbon length** (m)
ESC R08	n0(1)	**Request warning length** (m)
ESC R10	c0(10)	**Request password**
ESC R12	n0(1)	**Request date language**
ESC R13	n0(4)	**Request print head serial number**
ESC R14	n0(6)	**Request print head maintenance count** (m)
ESC R15	n0(5)	**Request print quantity**

Command	Data (Printer sends)	Description
ESC R17	n0(3)	**Request print speed** (mm/s)
ESC R18	c0(8)	**Request time** c0=hh:mm:ss
ESC R20	n0(4)	**Request ribbon count** (m)
ESC R21	c0(10)	**Request format name**
ESC R22	c0(3)	**Request the network card firmware version**
ESC R23	n0(1)	**Request print trigger Level**
ESC R24	n0(3)	**Request the NGT processor temperature** (°C)
ESC R25	n0(2)	**Request the CPLD firmware version**
ESC R27	c0(10)	**Request software version on Head**
ESC R28	c0(10)	**Request software version on MES**
ESC R29	c0(various)	**Request printer ID**
ESC R30	n0(4)	**Request print delay** IM-mode: millisecond (ms) CM-mode: millimeter (mm) (since version 7.3.00 – before: ms)
ESC R32	n0(1)	**Request print buffer**
ESC R33	n0(1)	**Request inhibit level**
ESC R34	n0(3)	**Retrieve the host machine maximal speed** (mm/s)
ESC R36	c0(various)	**Retrieve 2 colour raw bitmap of label** Scale 1:1. Bitmaps have 8 byte header. 4 width then 4 height. Note: when receiving messages larger than 999999 bytes, the message header will overflow to the left, into the acknowledge flag. In these cases, the header should be discarded, and the message length calculated from the width and length field according to the formula ((width*length)/8)+8

Command	Data (Printer sends)	Description
ESC R40n0(2)	Cannot request	**Load test format** n0=see test formats below
ESC R41n0(2)	Cannot request	**Load printer diagnostic** n0=see printer diagnostics below
ESC R42	n0(2)n1(2)	**Request the NGT processor temperature threshold** (°C) n0: lower threshold value n1: upper threshold value
ESC R45	n0(1)	**Request the error announce state** 0: Disabled, 1: Enabled
ESC R46	c0(10)	**Request parser software version**
ESC R47	n0(5)	**Request the image horizontal offset (pixel)** *(new since N7.3.01-R)*
ESC R48	n0(3)	**Request the IM mode image vertical offset (mm)** *(new since N7.3.01-R and MES005.0-R))*
ESC R49	n0(2)	**Request the IM mode post-print delay (ms)** *(new since N7.3.01-R and MES005.0-R)*
ESC R50	n0(1)n1(3)	**Request the ribbon rewind state and offset (pixel) (only NGT++)** *(new since N7.3.01-R)* n0(1) = 0: disabled, n0(1) = 1: enabled n1(3): rewind offset
ESC R51	n0(2)[n1(3)n2(3)](n)	**Request the print area number and coordinates** n0(2): number of print areas n1(3): print area upper coordinate n2(3): print area lower coordinate

Command	Data (Printer sends)	Description
ESC R52	n0(1)	**Request the ribbon spare mode** *(new since N7.3.01-R and MES005.0-R)* CM: “lower ribbon speed” n0(1) = 0: deactivated, n0(1) = 1: 25%, n0(1) = 2: 50% (**only NGT++**) IM: “Side Step Function (together with multiple areas)” n0(1) = 0: deactivated, n0(1) = 1: activated
ESC R53	n0(1)	**Request communication interface** 0=Serial (RS232), 1=Network (Ethernet)
ESC R54	n0(1)	**Request general error relay level**
ESC R55	n0(1)	**Request auto print status** 1-enabled 2-disabled
ESC R56	n0(1)	**Request format orientation**
ESC R57	c0(various)(0x0B)c1(various)(0x0B)c2(various)(0x0B)c3(various)(0x0B)	**Request the XPort (Ethernet) basic settings** c0: IP address c1: Subnet mask c2: Gateway address c3: local port
ESC R58	n0(3)	**Request “hysteresis scale factor”** *(new since N8.1.00-R)*
ESC R70	n0(3)c0(10)	**Request currently activated format** n0: ID (first format-ID is 000!) c0: format name
ESC R7400	n0(1)	**Request memory mode**
ESC R7401	n0(1)	**Request touchscreen availability status**
ESC R7403	n0(1)	**Request Intermittent/Continuous status**
ESC R7406	n0(4)	**Request carriage position** (mm)

Command	Data (Printer sends)	Description
ESC R7407	n0(5)	**Request network timeout** (ms)
ESC R7408	n0(5)	**Request serial timeout** (ms)
ESC R7501	c0(various)	**Request head software version**
ESC R7502	n0(2)	**Request desired Preheater value** (°C)
ESC R7503	n0(2)	**Request actual Preheater value (°C)**
ESC R7504n0(2)	n1(6)	**Request actual Pressure** n0=unit: 00=bar, 01=Kpa, 02=Atm, 3=psi, 4=Kla
ESC R7505	n0(4)	**Request actual Printhead Voltage (V)**
ESC R7506	n0(3)	**Request max. format length of used printer in intermittent mode (mm)**
ESC R7507	n0(3)	**Request user set max. format length in intermittent mode (mm)**
ESC R7600	c0(10)	**Request MES-print software version**
ESC R7601	c0(10)	**Request MES-foil software version**
ESC R77	n0(1)	**Request low foil level**
ESC R78	n0(1)	**Request start of the week**
ESC R79	n0(1)	**Request ribbon type**
ESC R80n0(2)	n1(2 or 35)	**Request counter attributes** n0: number of requested counter (if n0=0: return number of counters in format!) n1: counter attributes

Command	Data (Printer sends)	Description
ESC R81n0(3)n1(2)	n2(3)+c1(n1*10)	**Request a block of format names** n0: number of the first format name to fetch n1: number of subsequent formats to fetch n2: available format slots (Standard: 200) c1: block of format names The first format has number 1. If you specify 0 as first format, then the first sent format will also be 1. If you request formats beyond the last available formats, then those formats won't be sent.
ESC R82	n0(3)	**Request minimum print speed** (mm/s)
ESC R83c0(2)	[c1(2)c2(various)(0x0B)](various)(0x0B)	**Request a list of the names of all variable fields** c0=00: Request all variable fields c0=VI: Request all variable input fields c0=VC: Request all variable counter fields c0=VD: Request all variable date fields c0=VF: Request all variable formula fields c0=VX: Request all variable free fields For each field: c1: Type of the variable field (VI, VC, VD, VF, VX) c2: Name of the variable field (0x0B): Terminating character (0x0B): Terminating character

Command	Data (Printer sends)	Description
ESC R84c0(2)c1(various)(0x0B)	c2(various)(0x0B)n0(4)n1(4)	**Request content + lengths of a content element** c0: Type of the variable Field (VI, VC, VX) c1: Name of the variable Field (0x0B): Terminating character c2: VI and VX - Content (alphanumeric) VC – current value (9 digits) n0: minimum length of input variable n1: maximum length of input variable Minimum and maximum lengths are only available for VI fields and are set to 0000 each for other fields. For an input variable, is only the variable part without prefix and suffix.
ESC R85c0(2)c1(various)(0x0B)	c2(various)(0x0B)	**Request result of a content element** c0: Type of the variable Field (VI, VC, VD, VF, VX) c1: Name of the variable Field (0x0B): Terminating character c2: result of the content element e.g. for an input variable, this includes prefix and suffix

Command	Data (Printer sends)	Description
ESC R86n0(3)	n1(8)c0(various)	**Request escape sequence for a stored format** n0: Number of format to return n1: length of data that is returned c0: the escape sequence The returned escape sequence starts with <ESC>F, thus no tilde header is included. A trailing <ESC>Q00000 is transmitted. If requesting format of an empty slot, data of length zero is returned. Note: When requesting a format that is larger than 999999 bytes, the header will overflow to the right. The header should be discarded. The <ESC> character can be used to mark the start of the payload) and the n1 parameter used instead to determine the message length.
ESC R880n0(3)	n1(3)n2(n1*3)	**Return the number and ids of all pending errors** n0: number of pending error ids to return. 000 for all. n1: number of pending errors n2: id of error (n1 times)
ESC R881n0(3)	n1(3)s1(various)[n2(3)c0(various)(0x0B)c1(various)(0x0B)](various)	**Return details about pending errors** n0: number of pending error detail to return. 000 for all. n1: number of pending errors for each pending error: n2: id of error c0: name string of error (terminated by (0x0B)) c1: detail message about error (terminated by (0x0B))

Command	Data (Printer sends)	Description
ESC R883n0(3)	n1(3)[n2(3)c0(various)(0x0B)c1(various)(0x0B)](various)	**Return details about errors in history** n0: number of history error details to return. 000 for all. n1: number of errors for each error: n2: id of error (three digits) c0: name string of error (terminated by (0x0B)) c1: detail message about error (terminated by (0x0B))
ESC R89	n0(3)	**Request post-valve delay (thermal bar delay)** IM-mode: millisecond (ms) CM-mode: millimeter (mm) - *(new since version 7.3.00)*
ESC R90	n0(1)	**Request Suppress Print Interrupted Error** If this is set to 1, the “Print Interrupted” error is suppressed. Default is 0
ESC R91	n0(1)	**Request the loaded format state.** n0=1: a format is loaded n0=0: no format is loaded
ESC R92	n0(1)	**Request the ribbon warning state.** n0=0: the warning is disabled and the error enabled n0=1: the warning is enabled and the error enabled n0=2: the warning is disabled and the error disabled
ESC R93	n0(2)	**Request serialization buffer max elements** *maximum elements count for serialization variables* *0 = Serialization disabled* *(new since N8.6.02-R)*
ESC R94	n0(2)	**Request current available serialization elements** *number of current available elements in the buffer* *(new since N8.6.02-R)*
ESC R95	n0(2)	**Request low buffer RD3 warning** *Low Buffer RD3 warning is sent by the printer if available element buffer count is lower than this value* *(new since N8.6.02-R)*

Command	Data (Printer sends)	Description
ESC R96	n0(1)	**Request buffer underrun enabled** *Printer stops if serialization buffer underrun occurs* *(new since N8.6.02-R)*
ESC R99	n0(2)	**Request Error Status** See error status table!

Test Formats Commands

Terminal asks **ESC R40 Command**, Printer load requested format and answer with ***ESC ZOK***

Test Formats are temporary; they are lost after power off!

Command	Format Name	Description
ESC R4000	**Legacy**	Standard Allen test format of shrinking filled boxes
ESC R4005	**Hatch**	Vertical and horizontal lines going through the whole format
ESC R4010n0(2)	**Box(mm)**	A box of given size in mm (10-90)
ESC R4023n0(3)	**Quality**	Test format to evaluate the print quality. n0: placeholder for ribbon type in print image.

Error Status

Error	Description
00	**No error**
01	**Ribbon failed**
03	**Ribbon low**
04	**Graphic off format**
05	**Line / Box off format**
06	**Text off format**
07	**Barcode off format**
08	**Out of memory** Flash memory is full.
10	**Communication time out**
17	**Homesensor not found**
30	**Head comm. time out**
31	**DPRAM comm. time out**
32	**Not allowed operation**
33	**SRAM Out of memory**
34	**Print release error** Either no format is loaded or the print quantity is consumed
36	**No supported ESC-SEQ**
45	**Print format empty**
46	**Maglatch not close**
47	**Air pressure too high/low**

Error	Description
48	**Print head voltage too high/low**
51	**Print signal too fast**
56	**In home position while printing**
58	**Print Interrupted** Issued when print process was interrupted. Most often caused by wrong substrate speed in CM-mode.
60	**Format length exceeded** Format is longer than the allowed length set inside the printer.
61	**Parameter out of bounds** Issued when trying to set a print parameter (e.g. speed, density) to an value that is out of range
62	**Invalid SRAM data**
63	**NGT+ processor temperature too high**
64	**XPort configuration error**
99	**Miscellaneous error**