

---

## ESC Sequences

### Programming Reference

for

### Diagraph – NGT-Series

#### Copyright

The present document and the described devices therein belong to the company Diagraph GmbH.

All rights remain reserved. Reproduction of this document, in full or in part, requires prior explicit and written approval of the company Diagraph GmbH.

©Copyright 2021 Diagraph GmbH, all rights reserved.

#### Disclaimer

The company Diagraph 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 Diagraph GmbH does not accept liability for possible damages that may have been caused by faulty information in this document. The company Diagraph 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

Diagraph GmbH

Friedrich-Bergius-Ring 30

D-97076 Würzburg

Direct +49 931 250 76 – 0

Fax +49 931 250 76 – 50

Email: [info@diagraph.de](mailto:info@diagraph.de)

Website: <http://www.diagraph.de/>

## Index

Copyright .....	1
Disclaimer .....	1
Manufacturer .....	1
<b>INDEX.....</b>	<b>2</b>
<b>INTRODUCTION .....</b>	<b>6</b>
<b>CONVENTIONS USED.....</b>	<b>6</b>
<b>NORMAL OPERATIONS .....</b>	<b>6</b>
<b>MESSAGE HEADER .....</b>	<b>6</b>
Examples .....	7
<b>FIELD PARAMETERS.....</b>	<b>10</b>
Content Data .....	10
Variable Field - Input: (content element) .....	14
Variable Field - Counter: (content element) .....	15
Variable Field - Date: (content element) .....	16
Variable Field - Formula: (content element) .....	20
Variable Field - Free: (content element) .....	23
Text field: (representation element) .....	24
Barcode field: (representation element) .....	25
Graphics field: (representation element) .....	37
Line / box field: (representation element) .....	39
Invert: (representation element) .....	40
<b>PRINT CONTROL PARAMETERS .....</b>	<b>41</b>
Special Setting Parameters .....	41
Set Print speed: Intermittent specific .....	41
Set Print density as percent: .....	41
Set Print quantity: .....	41
Standard Setting Parameters .....	42
Set Printhead Resistance: .....	42
Set Ribbon length: .....	42
Set Warning length: .....	42
Set Date Language: .....	42
Set Printhead Serial Number: .....	43
Reset printhead maintenance count: .....	43
Reset print count: .....	43
Set minimum print speed: .....	43
Set Print Delay: .....	43
Set Print Pulse: .....	43
Get Printer ID: .....	43
Set Image Orientation: .....	44
Additional Settings .....	44
Set time .....	44
Set date .....	44

Set density as a percent .....	44
Print trigger level, .....	44
Auto print .....	44
Set the NGT processor temperature control .....	44
Set the NGT+ processor temperature threshold (in °C) .....	44
Trigger a print .....	44
Delete resident format .....	44
Set print signal buffer .....	44
Set the host machine maximal speed (mm/s) .....	45
Set inhibit level .....	45
Set the image horizontal offset (pixel) .....	45
Set the IM mode image vertical offset (pixel) .....	45
Set the IM mode post-print delay (ms) .....	45
Set the ribbon rewind state and offset in pixel (only NGT++) .....	45
Set the ribbon spare mode (new since N7.3.01-R and MES005.0-R) .....	45
Select communication interface .....	45
General error relay .....	45
Load format .....	45
Set memory mode .....	45
Set preheater temperature (in °C) .....	45
Delete formats .....	45
Find homesensor .....	46
Motor steps (forward/reverse) .....	46
Set intermittent/continuous mode (new printer start muss be done) .....	46
Send "ready"-string after a format-transmission or after a print .....	46
Set low foil level .....	46
Reset error .....	46
Set connection timeout (network) in millisecond .....	46
Set connection timeout (serial) in millisecond .....	46
Set the max. format length in millimeter .....	46
Set ribbon type .....	46
Set counter attributes .....	46
Announce error .....	47
Set content of content element .....	47
Set post-valve delay (thermal bar delay) .....	47
Suppress Print Interrupted Error .....	47
Set the number and the coordinates of the areas to print. ....	47
Set the ribbon error state .....	47
Set the first week day to either Sunday or Monday .....	47
Save a format at a given slot number without any data processing .....	47
Reset all SRAM printer settings .....	48
Set the XPort (Ethernet) basic settings .....	48
Set hysteresis scale factor .....	48
Standard Request Parameters: .....	51
Request control parameters: .....	51
Request printer status: .....	51
Request maintenance count: .....	51
Request Printhead History .....	51
Additional Requests: .....	52
Request print head resistance (Ω) .....	52
Request print count .....	52
Request date .....	52
Request density as a percentage .....	52
Request ribbon length (m) .....	52
Request warning length (m) .....	52
Request password .....	52

Request date language .....	52
Request print head serial number .....	52
Request print head maintenance count (m) .....	52
Request print quantity .....	52
Request print speed (mm/s).....	52
Request time .....	52
Request ribbon count (m) .....	52
Request format name.....	52
Request the network card firmware version.....	52
Request print trigger Level .....	52
Request the NGT processor temperature (°C) .....	53
Request the CPLD firmware version .....	53
Request software version on Head .....	53
Request software version on MES.....	53
Request printer ID .....	53
Request print delay .....	53
Request print buffer .....	53
Request inhibit level .....	53
Retrieve the host machine maximal speed (mm/s) .....	53
Retrieve 2 colour raw bitmap of label.....	53
Load test format .....	53
Load printer diagnostic.....	53
Request the NGT processor temperature threshold (°C) .....	53
Request the error announce state .....	53
Request parser software version.....	53
Request the image horizontal offset (pixel) .....	54
Request the IM mode image vertical offset (mm) .....	54
Request the IM mode post-print delay (ms) .....	54
Request the ribbon rewind state and offset (pixel) .....	54
Request the print area number and coordinates .....	54
Request the ribbon spare mode (new since N7.3.01-R and MES005.0-R) .....	54
Request communication interface .....	54
Request general error relay level .....	54
Request auto print status .....	54
Request format orientation.....	54
Request the XPort (Ethernet) basic settings.....	54
Request “hysteresis scale factor” .....	54
Request PrintQuantityCounter .....	54
Request SuppressAirPressureError .....	55
Request IgnoreSpeedDensityInFormat .....	55
Request EnableFixedOffset .....	55
Request FoilMotorBreakOff .....	55
Request ignore print quantity .....	55
Request ignore off format .....	55
Request Intermittent/Continuous status .....	55
Request carriage position (mm) .....	55
Request network timeout (ms) .....	56
Request serial timeout (ms) .....	56
Request head software version.....	56
Request desired Preheater value (°C) .....	56
Request actual Preheater value (°C).....	56
Request actual Pressure .....	56
Request actual Printhead Voltage (V) .....	56
Request max. format length of used printer in intermittent mode (mm) .....	56
Request user set max. format length in intermittent mode (mm).....	56
Request MES-print software version.....	56

---

Request MES-foil software version .....	56
Request low foil level .....	56
Request start of the week .....	56
Request ribbon type.....	56
Request counter attributes .....	56
Request a block of format names.....	57
Request minimum print speed (mm/s) .....	57
Request a list of the names of all variable fields .....	57
Request content + lengths of a content element .....	57
Request result of a content element.....	57
Request escape sequence for a stored format.....	58
Return the number and ids of all pending errors .....	58
Return details about pending errors .....	58
Return details about errors in history .....	58
Request post-valve delay (thermal bar delay).....	58
Request Suppress Print Interrupted Error.....	58
Request the loaded format state. ....	58
Request Error Status.....	59
Test Formats Commands.....	60
Error Status.....	61

## Introduction

This document describes commands used with the NGT range of thermal printers of Diagraph 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

New functions are marked red

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.

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 possible 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 #I", "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} {aaaaaaaa} {bbbbbbbb} {cccccccc} {dddddddd} {eeeeeeee} {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)
{aaaaaaaa}	=	Start (-99999999 to 999999999).
{bbbbbbbb}	=	Stop (-99999999 to 999999999).
{cccccccc}	=	Step ( -99999999 to 099999999).
{dddddddd}	=	Duration (0000001 to 999999999).
{eeeeeeee}	=	Reset sequential number to.
{j..j}	=	Base (User defined) <div> <div>"0123456789" or empty</div> <div>=</div> <div>decimal</div> </div> <div> <div>"01"</div> <div>=</div> <div>binary</div> </div> <div> <div>"ABCD...YZ"</div> <div>=</div> <div>alpha</div> </div> <div>...</div>
{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}}

*(alternative with hour codes new since N8.700-R)*

ESC VD {N..N} (0x0B) {data} (0x0B) {{{O} {ddd} {mmm} {yyy} {sss} {MMM} {hhh}} | {{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
Hour Code	=	22 (new since N8.700-R)



(alphanumeric hours i.e. replacing 0 to 24 -> A to X)

Reserved = 23...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)

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

// French "JAN", "FEV", "MAR", "AVR", "MAI", "JUN", "JUL", "AOU", "SEP", "OCT", "NOV", "DEC"

// Sweden "JAN", "FEB", "MAR", "APR", "MAJ", "JUN", "JUL", "AUG", "SEP", "OKT", "NOV", "DEC"

// German "JAN", "FEB", "MAR", "APR", "MAI", "JUN", "JUL", "AUG", "SEP", "OKT", "NOV", "DEZ"

// Italian "GEN", "FEB", "MAR", "APR", "MAG", "GIU", "LUG", "AGO", "SET", "OTT", "NOV", "DIC"

// Spanish "ENE", "FEB", "MAR", "ABR", "MAY", "JUN", "JUL", "AGO", "SEP", "OCT", "NOV", "DIC"

// Dutch "JAN", "FEB", "MAA", "APR", "MEI", "JUN", "JUL", "AUG", "SEP", "OKT", "NOV", "DEC"

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
International week	=	60
Hour Code extender		61{a..a}{0x03}{b..b}{0x03}{c..c}{0x03}{d..d}...(0x03)
		Where (0x03) is the separator character and each sequence (always 24-sequences) is for 0-24 (new since N8.700-R)

{O}	=	Fixed information for the offset value with hour codes. (new since N8.700-R)
{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
{sss}	=	-99 - 999      number of advanced seconds (new since N8.700-R)
{MMM}	=	-99 - 999      number of advanced minutes (new since N8.700-R)
{hhh}	=	-99 - 999      number of advanced hours (new since N8.700-R)
{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)&Vlweight(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 = (Reserved)
		09 = DataBar ( <i>new since N8.1.00-R</i> )
		10 = QR Code ( <i>new since N8.1.00-R</i> )
		11 = Hanxin Code ( <i>new since N8.700-R</i> )
		12 = PDF417 ( <i>new since N8.700-R</i> )
		13 = (Reserved)
		14 = Denso Wave Micro QR Code ( <i>new since N8.900-R</i> )
{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

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 ( <i>new since N8.1.00-R</i> ):		
{zzzz} {hhhhh} {nn} {rrrr} {ss} {pp}		
{zzzz}	=	<p>Symbology:</p> <p>0029 = Omnidirectional/Truncated</p> <p>0030 = Limited</p> <p>0031 = Expanded</p> <p>0079 = Stacked</p> <p>0080 = Stacked omnidirectional</p> <p>0081 = Expanded stacked</p>
{hhhhh}	=	Height ( <u>excluding</u> human readable line)
{nn}	=	Widths of narrow bar and space element
{rrrr}	=	<p>Height of the human readable line:</p> <p>00000 = No human readable line</p> <p>For all stacked codes the human readable line is switched off!</p>
{pp}	=	<p>Segments per row (just for "expanded stacked" type)</p> <p>00 = Not used!</p>
QR Code ( <i>new since N8.1.00-R</i> ):		
{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 !**

Hanxin Code (*new since N8.7.00-R*):

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

{ss} = HanxinCode size :  
 21x21 (also automatic) = 00  
 23x23 = 01  
 25x25 = 02  
 27x27 = 03  
 29x29 = 04  
 31x31 = 05  
 33x33 = 06  
 35x35 = 07

---

37x37 = 08  
39x39 = 09  
41x41 = 10  
43x43 = 11  
45x45 = 12  
47x47 = 13  
49x49 = 14  
51x51 = 15  
53x53 = 16  
55x55 = 17  
57x57 = 18  
59x59 = 19  
61x61 = 20  
63x63 = 21  
65x65 = 22  
67x67 = 23  
69x69 = 24  
71x71 = 25  
73x73 = 26  
75x75 = 27  
77x77 = 28  
79x79 = 29  
81x81 = 30  
83x83 = 31  
85x85 = 32  
87x87 = 33  
89x89 = 34  
91x91 = 35  
93x93 = 36  
95x95 = 37  
97x97 = 38  
99x99 = 39  
101x101 = 40  
103x103 = 41  
105x105 = 42  
107x107 = 43  
109x109 = 44  
111x111 = 45  
113x113 = 46  
115x115 = 47  
117x117 = 48  
119x119 = 49  
121x121 = 50  
123x123 = 51

---

125x125 = 52  
127x127 = 53  
129x129 = 54  
131x131 = 55  
133x133 = 56  
135x135 = 57  
137x137 = 58  
139x139 = 59  
141x141 = 60  
143x143 = 61  
145x145 = 62  
147x147 = 63  
149x149 = 64  
151x151 = 65  
153x153 = 66  
155x155 = 67  
157x157 = 68  
159x159 = 69  
161x161 = 70  
163x163 = 71  
165x165 = 72  
167x167 = 73  
169x169 = 74  
171x171 = 75  
173x173 = 76  
175x175 = 77  
177x177 = 78  
179x179 = 79  
181x181 = 80  
183x183 = 81  
185x185 = 82  
187x187 = 83  
189x189 = 84  
191x191 = 85  
193x193 = 86  
195x195 = 87  
197x197 = 88

{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!) <b>! Currently without function !</b> 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 ) <b>! Currently without function !</b> 2 = GS1 (Encodes GS1 data using FNC1 characters)
{u}	=	GS1 conformity checking: 0 = Off 1 = On (without AI-checking) <b>! Currently without function !</b>

PDF 417:

{{tttt} {zz} ccc} {rrr} {l}}

{tttt}	=	Symbology
		24 = PDF417
		25 = PDF417 Truncated
		26 = PDF417 HIBC
		27 = PDF417 HIBC Micro
		28 = PDF417 Micro
{zz}	=	Module height
{ccc}	=	Number of columns
{rrr}	=	Number of rows
{l}	=	security level

## 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 {xxxx} {yyyy} {wwww} {hhhh} {vvvv} {rrrr}

{xxxx} = Horizontal position.

{yyyy} = Vertical position.

{wwww} = Width.

{hhhh} = Height.

{vvvv} = Vertical thickness.

{rrrr} = 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.

---

## Invert: (representation element)

ESC #I {xxxxx} {yyyyy} {wwwww} {hhhhh} {vvvvv} {rrrrr}

{xxxxx} = Horizontal position.

{yyyyy} = Vertical position.

{wwwww} = Width.

{hhhhh} = Height.

These parameters may describe a region which will be inverted. No terminator character is required as the data is of fixed length.



---

## 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 Regions Reset with external Signal:

ESC Z {x}                  Where {x}

0= off  
1= on

?= shows current value

### 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 ZN{xxxx}                      Where {xxxx}                      = 0000 to 9999

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

### Reset printhead maintenance count:

ESC ZM

**Reset print count:**

ESC ZQ

**Set minimum print speed:**

ESC ZIS{xxx}                      Where {xxx}                      = 050 to MAXIMUM (mm/sec)

### Set Print Delay:

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

**Set Print Pulse:**

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

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

**Get Printer ID:**

ESC ZRP	24: NG2 Intermittent 53mm
	28: NG2 Extender 53mm
	25: NG2 Continuous 53mm
	26: NG4 Intermittent 107mm
	29: NG4 Extender 107mm
	27: NG4 Continuous 107mm
	20: NG6 Intermittent 160mm
	30: NG6 Extender 160mm
	21: NG6 Continuous 160mm
	22: NG8 Intermittent 213mm
	31: NG8 Extender 213mm
	23: NG8 Continuous 213mm

### 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:**

<b>ni,</b>	<b>i={0, 1, 2, ...}</b>	– digit (ASCII 0x30 – 0x39)
<b>ni(x)</b>		x-digits (if the length is not defined: x = various)
<b>ci,</b>	<b>i={0, 1, 2, ...}</b>	– ASCII character (0x20 – 0xFF)
<b>ci(x)</b>		a string of length x (if the length is not defined: x = various)
<b>[s](x)</b>		square brackets structured groups ( )
		x is the total length of the group (if the length is not defined: x = various)
<b>(0x00)</b>		– 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)	<b>Set time</b>
		n0=hhmmss
ESC X01	n0(9)	<b>Set date</b>
		n0=DDMMYYYYW (W=Day of the week)
ESC X07	n0(3)	<b>Set density as a percent</b>
		n0=000 – 100 degrees
ESC X08	n0(1)	<b>Print trigger level,</b>
		0=low, 1=high
ESC X09	n0(1)	<b>Auto print</b>
		0=off, 1=on
ESC X10	n0(1)	<b>Set the NGT processor temperature control</b>
		0 = disable, 1 = enable
ESC X11	n0(2)n1(2)	<b>Set the NGT+ processor temperature threshold (in °C)</b>
		n0: lower threshold value
		n1: upper threshold value
ESC X12		<b>Trigger a print</b>
ESC X13		<b>Delete resident format</b>
ESC X14	n0(1)	<b>Set print signal buffer</b>
		0=enable, 1=disable

Command	Data	Description
ESC X15	n0(3)	<b>Set the host machine maximal speed (mm/s)</b> (new since N7.3.00-R and MES004.6-D) For speed compensation in CM - mode
ESC X16	n0(1)	<b>Set inhibit level</b>  0=Low, 1=High
ESC X17	n0(5)	<b>Set the image horizontal offset (pixel)</b>  (new since N7.3.01-R)
ESC X18	n0(3)	<b>Set the IM mode image vertical offset (pixel)</b>  (new since N7.3.01-R and MES005.0-R)
ESC X19	n0(2)	<b>Set the IM mode post-print delay (ms)</b>  (new since N7.3.01-R and MES005.0-R)
ESC X20	n0(1)n1(3)	<b>Set the ribbon rewind state and offset in pixel (only NGT++)</b>  (new since N7.3.01-R) n0= 0: disabled, n0 = 1: enabled n1: rewind offset (min: -99, max: 127)
ESC X21	n0(1)	<b>Set the ribbon spare mode</b> (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 X23	n0(1)	<b>Select communication interface</b>  0=Serial (RS232), 1=Network (Ethernet)
ESC X24	n0(1)	<b>General error relay</b>  0=Active High, 1=Active Low
ESC X36	n0(3)	<b>Load format</b>  n0: index of the format to load.
ESC X38	n0(1)	<b>Set memory mode</b>  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.
ESC X41	n0(2)	<b>Set preheater temperature (in °C)</b>  01 - 60 degrees
ESC X42	n0(3)	<b>Delete formats</b>  000=All Formats, 001=Format 1, 002=Format 2, 003=Format 3, ...

Command	Data	Description
ESC X43		<b>Find homesensor</b>
ESC X44	n0(1)n1(4)	<b>Motor steps (forward/reverse)</b>  n0=1: forward n0=0: reverse n1=0001-1200pixels
ESC X45	n0(1)	<b>Set intermittent/continuous mode</b> (new printer start muss be done)
ESC X47	n0(1)n1(1)	1=intermittent, 0=continuous <b>Send "ready"-string after a format-transmission or after a print</b>  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)	<b>Set low foil level</b>  0=Low, 1=High
ESC X49	n0(1)	<b>Reset error</b>  1=reset, x=nothing
ESC X50	n0(5)	<b>Set connection timeout (network) in millisecond</b>  n0=00100-20000 ms (Default: 6000 ms)
ESC X51	n0(5)	<b>Set connection timeout (serial) in millisecond</b>  n0=00100-20000 ms (Default: 700 ms)
ESC X52	n0(4)	<b>Set the max. format length in millimeter</b>  n0: max. length of the format
ESC X53	n0(1)	<b>Set ribbon type</b>  0=Wax/Resin, 1=Resin
ESC X54	n0(2)n1(7)n2(7)n3(7)n4(7)n5(7)	<b>Set counter attributes</b>  n0: number of counter to change n1: start value n2: stop value n3: step n4: duration n5: reset sequential number to

Command	Data	Description
ESC X55	n0(1)	<p><b>Announce error with [ESC]ZERROR followed by the error number n1(2) and [ESC]ZERCLR after the pending error has been cleared</b></p> <p>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.</p>
ESC X56	n0(2)c0(various)(0x0B)c1(various)(0x0B)	<p><b>Set content of content element</b></p> <p>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)</p>
ESC X57	n0(3)	<p><b>Set post-valve delay (thermal bar delay)</b></p> <p>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 (<i>new since N7.3.03-R and MES004.6-D</i>) Default: 005 mm Sets the length of a delay between triggering the air-valve and beginning to print (increase the ribbon index!).</p>
ESC X58	n0(1)	<p><b>Suppress Print Interrupted Error</b></p> <p>When this is set to 1, the "Print Interrupted" error is suppressed. Default is 0.</p>
ESC X59	n0(2)[n1(3)n2(3)](n)	<p><b>Set the number and the coordinates of the areas to print.</b></p> <p>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.</p>
ESC X60	n0(1)	<p><b>Set the ribbon error state.</b></p> <p>n0=0: warning disabled and error enabled. n0=1: warning enabled and error enabled. n0=2: warning disabled and error disabled.</p>
ESC X62	n0(1)	<p><b>Set the first week day to either Sunday or Monday</b></p> <p>n0=0: first week day is Sunday n0=1: first week day is Monday</p>
ESC X63	n0(3)n1(6)	<p><b>Save a format at a given slot number without any data processing</b></p> <p>n0: slot number n1: format length</p>

Command	Data	Description
ESC X64		<b>Reset all SRAM printer settings</b>
ESC X65	n0(0x0B)n1(0x0B)n2(0x0B)n3(0x0B)	<b>Set the XPort (Ethernet) basic settings</b> n0: IP address n1: Subnet mask n2: Gateway address n3: local port
ESC X66	n0(1)n1(3)	<b>Set hysteresis scale factor</b> n0=0: no change info requested n0=1: change info requested n1=001-255 ("Hysteresis scale factor") standard: 51 <i>(new since N8.1.00-R)</i>
ESC X67	n0(2)	<b>Set serialization buffer max elements</b> n0=00: serialization disabled n0=01-30 maximum elements, standard : 30 If serialization is enabled, serialization content has to be provided for each print <i>(new since N8.6.02-R)</i>
ESC X68	n0(2)n1(0x0B)n2(0x0B)n	<b>Append serialization content to buffer</b> 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} <i>(new since N8.6.02-R)</i>
ESC X69	n0(2)[n1(0x0B)](n)	<b>Delete serialization content for variable element</b> n0=number of variable elements n1=variable names, i.e VIVariable1 <i>(new since N8.6.02-R)</i>



Command	Data	Description
ESC X70	n0(2)	<b>Set low buffer RD3 warning</b> n0=Serialization buffer minimum elements Printer sends RD3 signal when the element number in the serialization buffer is smaller than this value <i>(new since N8.6.02-R)</i>
ESC X71	n0(1)	<b>Set buffer underrun enabled</b> 1= printer stops if serialization buffer underrun occurs 0= printer ignores buffer underrun error <i>(new since N8.6.02-R)</i>
ESC X72	n0(1)	<b>Set reverse print enabled</b> 0= normal print 1= the complete print image is printed reverse <i>(new since N8.7.03-R)</i>
ESC X73	n0(1)	<b>Set ignore print trigger</b> 0= the print triggers are enabled 1= the print triggers are ignored <i>(new since N8.7.04-R)</i>
ESC X74	n0(1)	<b>Set ignore off format</b> 0= Printing objects outside the permitted print area is not permitted 1= Printing objects outside the permitted print area generates no error <i>(new since N8.8.02-R)</i>
ESC X75	n0(1)	<b>Set ignore print quantity</b> 0= Default 1= always ignore print quantity <i>(new since N8.8.03-R)</i>
ESC X76	n0(6)	<b>Set PrintQuantityCounter</b> Set the print counter to specified value min: 0 max: 999998 <i>(new since N8.8.05-R)</i>

# NGT – Escape Sequences

V3.19 - Programming Reference



Command	Data	Description
ESC X77	n0(1)	<b>Set SuppressAirPressureError</b> 0= Air pressure errors enabled 1= Air pressure errors disabled <i>(new since N8.8.06-R)</i>
ESC X78	n0(1)	<b>Set IgnoreSpeedDensityInFormat</b> 0= Speed and Density in print formats is enabled 1= Speed and Density in print formats is disabled <i>(new since N8.9.01-R)</i>
ESC X79	n0(1)	<b>Set EnableFixedOffset</b> 0= Horizontal offset is format specific 1= Horizontal offset is stored for all formats <i>(new since N8.9.04-R)</i>
ESC X80	n0(1)	<b>Set FoilMotorBreakOff</b> 0= Foil Motor break is on 1= Set Foil motor break off <i>(new since N8.9.04-R)</i>

---

## 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 ( $\Omega$ )
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
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

Command	Data (Printer sends)	Description
ESC R24	n0(3)	<b>Request the NGT processor temperature (°C)</b>
ESC R25	n0(2)	<b>Request the CPLD firmware version</b>
ESC R27	c0(10)	<b>Request software version on Head</b>
ESC R28	c0(10)	<b>Request software version on MES</b>
ESC R29	c0(various)	<b>Request printer ID</b>
ESC R30	n0(4)	<b>Request print delay</b>
		IM-mode: millisecond (ms) CM-mode: millimeter (mm) (since version 7.3.00 – before: ms)
ESC R32	n0(1)	<b>Request print buffer</b>
ESC R33	n0(1)	<b>Request inhibit level</b>
ESC R34	n0(3)	<b>Retrieve the host machine maximal speed (mm/s)</b>
ESC R36	c0(various)	<b>Retrieve 2 colour raw bitmap of label</b>
		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 $((width * length) / 8) + 8$
ESC R40	n0(2)	<b>Load test format</b>
		n0=see test formats below
ESC R41	n0(2)	<b>Load printer diagnostic</b>
		n0=see printer diagnostics below
ESC R42	n0(2)n1(2)	<b>Request the NGT processor temperature threshold (°C)</b>
		n0: lower threshold value n1: upper threshold value
ESC R45	n0(1)	<b>Request the error announce state</b>
		0: Disabled, 1: Enabled
ESC R46	c0(10)	<b>Request parser software version</b>

Command	Data (Printer sends)	Description
ESC R47	n0(5)	<b>Request the image horizontal offset (pixel)</b>  (new since N7.3.01-R)
ESC R48	n0(3)	<b>Request the IM mode image vertical offset (mm)</b>  (new since N7.3.01-R and MES005.0-R)
ESC R49	n0(2)	<b>Request the IM mode post-print delay (ms)</b>  (new since N7.3.01-R and MES005.0-R)
ESC R50	n0(1)n1(3)	<b>Request the ribbon rewind state and offset (pixel) (only NGT++) (new since N7.3.01-R)</b>  n0(1) = 0: disabled, n0(1) = 1: enabled n1(3): rewind offset
ESC R51	n0(2)[n1(3)n2(3)](n)	<b>Request the print area number and coordinates</b>  n0(2): number of print areas n1(3): print area upper coordinate n2(3): print area lower coordinate
ESC R52	n0(1)	<b>Request the ribbon spare mode (new since N7.3.01-R and MES005.0-R)</b>  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)	<b>Request communication interface</b>  0=Serial (RS232), 1=Network (Ethernet)
ESC R54	n0(1)	<b>Request general error relay level</b>
ESC R55	n0(1)	<b>Request auto print status</b>  1-enabled 2-disabled
ESC R56	n0(1)	<b>Request format orientation</b>
ESC R57	c0(various)(0x0B)c1(various)(0x0B)c2(various)(0x0B)c3(various)(0x0B)	<b>Request the XPort (Ethernet) basic settings</b>  c0: IP address c1: Subnet mask c2: Gateway address c3: local port
ESC R58	n0(3)	<b>Request "hysteresis scale factor"</b>  (new since N8.1.00-R)
ESC R60	n0(various)	<b>Request PrintQuantityCounter</b>  (new since N8.8.5-R)

Command	Data (Printer sends)	Description
ESC R61	n0(1)	<b>Request SuppressAirPressureError</b> 0- Airpressure errors enabled 1- Airpressure errors disabled (new since N8.8.6-R)
ESC R62	n0(1)	<b>Request IgnoreSpeedDensityInFormat</b> 0- Speed and Density in print formats is enabled 1- Speed and Density in print formats is disabled (new since N8.9.1-R)
ESC R63	n0(1)	<b>Request EnableFixedOffset</b> 0-Horizontal offset is format specific 1- Horizontal offset is stored for all formats (new since N8.9.4-R)
ESC R64	n0(1)	<b>Request FoilMotorBreakOff</b> 0-Foil Motor break is on 1- Foil motor break is off (new since N8.9.4-R)
ESC R68	n0(1)	<b>Request ignore print quantity</b> 0-disabled 1-enabled (new since N8.8.3-R)
ESC R69	n0(1)	<b>Request ignore off format</b> 0-disabled 1-enabled (new since N8.8.2-R)
ESC R71n	String	<b>Request all formats name</b> n=0.
ESC R72	n(6)	<b>Request total variable content</b> (3Char.) and autocodes + variable content (3Char.)
ESC R 73nnn	string	<b>Send corresponding variable content</b> nnn= variable ID
ESC R7403	n0(1)	<b>Request Intermittent/Continuous status</b>
ESC R7406	n0(4)	<b>Request carriage position (mm)</b>

Command	Data (Printer sends)	Description
ESC R7407	n0(5)	<b>Request network timeout (ms)</b>
ESC R7408	n0(5)	<b>Request serial timeout (ms)</b>
ESC R7501	c0(various)	<b>Request head software version</b>
ESC R7502	n0(2)	<b>Request desired Preheater value (°C)</b>
ESC R7503	n0(2)	<b>Request actual Preheater value (°C)</b>
ESC R7504n0(2)	n1(6)	<b>Request actual Pressure</b>
		n0=unit: 00=bar, 01=Kpa, 02=Atm, 3=psi, 4=Kla
ESC R7505	n0(4)	<b>Request actual Printhead Voltage (V)</b>
ESC R7506	n0(3)	<b>Request max. format length of used printer in intermittent mode (mm)</b>
ESC R7507	n0(3)	<b>Request user set max. format length in intermittent mode (mm)</b>
ESC R7600	c0(10)	<b>Request MES-print software version</b>
ESC R7601	c0(10)	<b>Request MES-foil software version</b>
ESC R77	n0(1)	<b>Request low foil level</b>
ESC R78	n0(1)	<b>Request start of the week</b>
ESC R79	n0(1)	<b>Request ribbon type</b>
ESC R80n0(2)	n1(2 or 35)	<b>Request counter attributes</b>
		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)	<b>Request a block of format names</b>  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)	<b>Request minimum print speed (mm/s)</b>
ESC R83c0(2)	[c1(2)c2(Various)(0x0B)](Various)(0x0B)	<b>Request a list of the names of all variable fields</b>  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
ESC R84c0(2)c1(Various)(0x0B)n0(4) n1(4)	c2(Various)(0x0B)n0(4) n1(4)	<b>Request content + lengths of a content element</b>  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)	<b>Request result of a content element</b>  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)	<b>Request escape sequence for a stored format</b>  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)	<b>Return the number and ids of all pending errors</b>  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)	<b>Return details about pending errors</b>  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))
ESC R883n0(3)	n1(3)[n2(3)c0(various)(0x0B)c1(various)(0x0B)](various)	<b>Return details about errors in history</b>  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)	<b>Request post-valve delay (thermal bar delay)</b>  IM-mode: millisecond (ms) CM-mode: millimeter (mm) - (new since version 7.3.00)
ESC R90	n0(1)	<b>Request Suppress Print Interrupted Error</b>  If this is set to 1, the "Print Interrupted" error is suppressed. Default is 0
ESC R91	n0(1)	<b>Request the loaded format state.</b>  n0=1: a format is loaded n0=0: no format is loaded
ESC R92	n0(1)	<b>Request the ribbon warning state.</b>  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

Command	Data (Printer sends)	Description
ESC R93	n0(2)	<b>Request serialization buffer max elements</b> maximum elements count for serialization variables <i>0 = Serialization disabled</i> <i>(new since N8.6.02-R)</i>
ESC R94	n0(2)	<b>Request current available serialization elements</b> number of current available elements in the buffer <i>(new since N8.6.02-R)</i>
ESC R95	n0(2)	<b>Request low buffer RD3 warning</b> Low Buffer RD3 warning is sent by the printer if available element buffer count is lower than this value <i>(new since N8.6.02-R)</i>
ESC R96	n0(1)	<b>Request buffer underrun enabled</b> Printer stops if serialization buffer underrun occurs <i>(new since N8.6.02-R)</i>
ESC R97	n0(1)	<b>Request reverse print</b> the complete print image is printed reverse <i>(new since N8.7.03-R)</i>
ESC R98	n0(1)	<b>Request ignore print trigger</b> print triggers are ignored <i>0 = print triggers are not ignored</i> <i>1 = print triggers are ignored</i> <i>(new since N8.7.04-R)</i>
ESC R99	n0(2)	<b>Request Error Status</b>  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	<b>Legacy</b>	Standard Allen test format of shrinking filled boxes
ESC R4005	<b>Hatch</b>	Vertical and horizontal lines going through the whole format
ESC R4010n0(2)	<b>Box(mm)</b>	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
48	Print head voltage too high/low
51	Print signal too fast

Error	Description
56	<b>In home position while printing</b>
58	<b>Print Interrupted</b> 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