

Supplement to Escape Sequences

Vision System Integration – V1.1

for

Allen Coding – NG-Series

Copyright

The present document and the described devices therein belong to the company Allen Coding GmbH. All rights remain reserved. Reproduction of this document, in full or in part, requires prior explicit and written approval of the company Allen Coding GmbH.

©Copyright 2012 Allen Coding GmbH, all rights reserved.

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@allen.de
Website: www.allen.de

Index:

COPYRIGHT 1

DISCLAIMER..... 1

MANUFACTURER 1

INDEX: 2

INTRODUCTION..... 3

CONVENTIONS USED 3

OPERATIONS 4

 PREPARATION/GENERAL INFORMATION 4

 GENERATE THE LABEL TEMPLATE REQUIRED..... 4

 HOW THE SEQUENCE SHOULD WORK..... 7

 HANDSHAKING FOR CHANGE DATA AFTER EVERY PRINT 9

 SET SPEED AND DENSITY 10

 EXAMPLE FOR SERIALIZATION IN GS1 DATA MATRIX CODE..... 10

Introduction

This document describes commands and functions used with the **NG** range of thermal printers of Allen Coding with firmware version N5.0-D and newer (N7.3 and newer is recommended).

Most vision systems will be the main control for the printer, the vision system will use the printers ESC Sequences (ASCII language) to send it “print format” (label information) and “print control parameters” to the printer.

This guide will help you understand the information required to integrate with the vision system.

Software you should use:

a:design (alternative Codesoft or NiceLabel)

a:control

Software not provided or supported by Allen or ITW:

Notepad++ (freeware download at your own risk).

This guide is intended for programmers who are familiar with the functionality of the NG range and the corresponding software!

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.

Key:

n, n1, n2, ... – digit

n(x) – x-digits (if the length is not defined: x = various)

c, c1, c2, ... – ASCII character

c(x) – a string of length x (if the length is not defined: x = various)

(0x00) – hexadecimal value

Operations

Preparation/General Information

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

Via serial connection (RS232) or Ethernet connection (TCP/IP) -> see also NG 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 to send 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

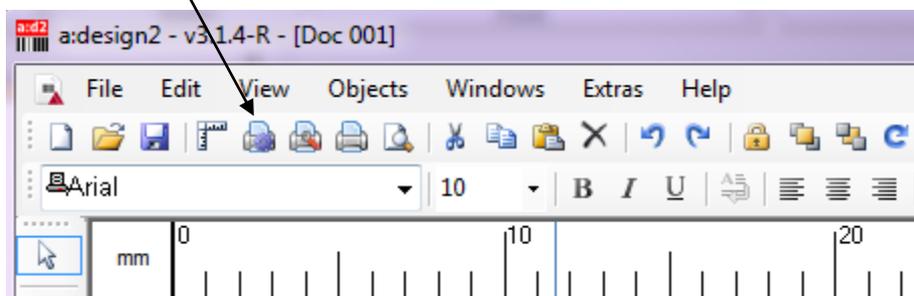
Generate the label template required

To start you should generate the label template required in a:design.

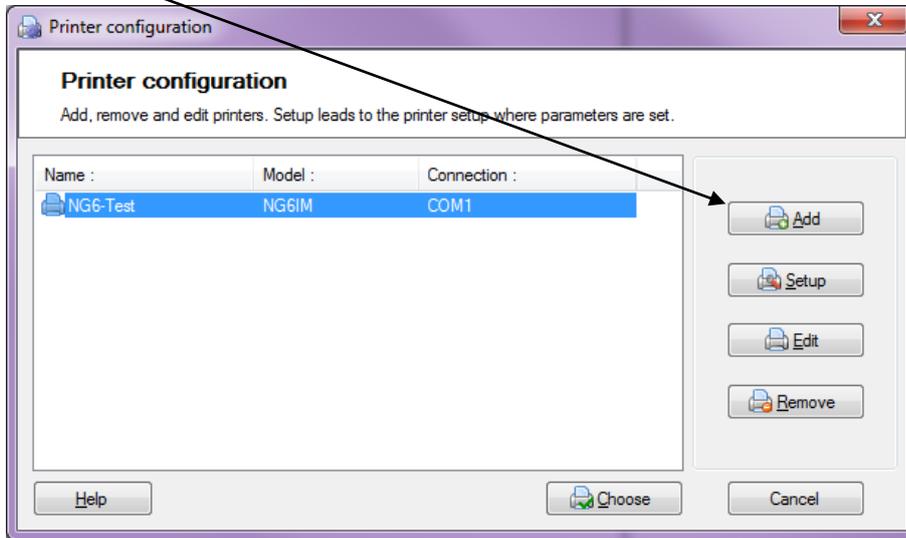
By printing this Information to file you can generate the label template, if you view the txt file with Notepad++ get a visual representation of the ASCII symbols in the string.

The first step is to set the printer up in a:design making sure you use the firmware version 5.xxx and above.

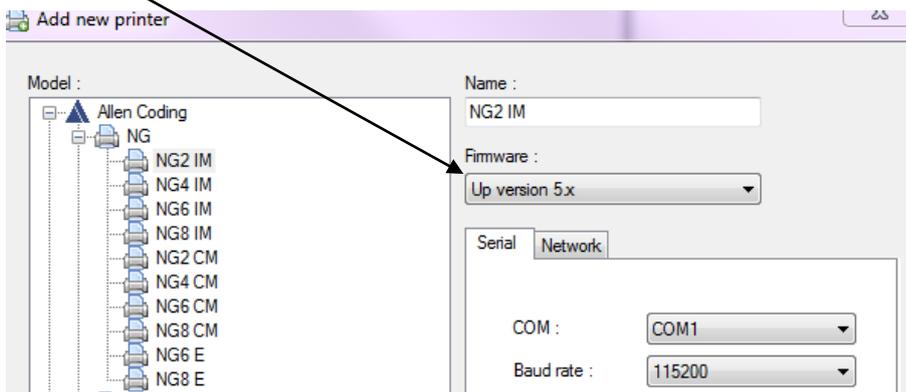
Go to printer set up



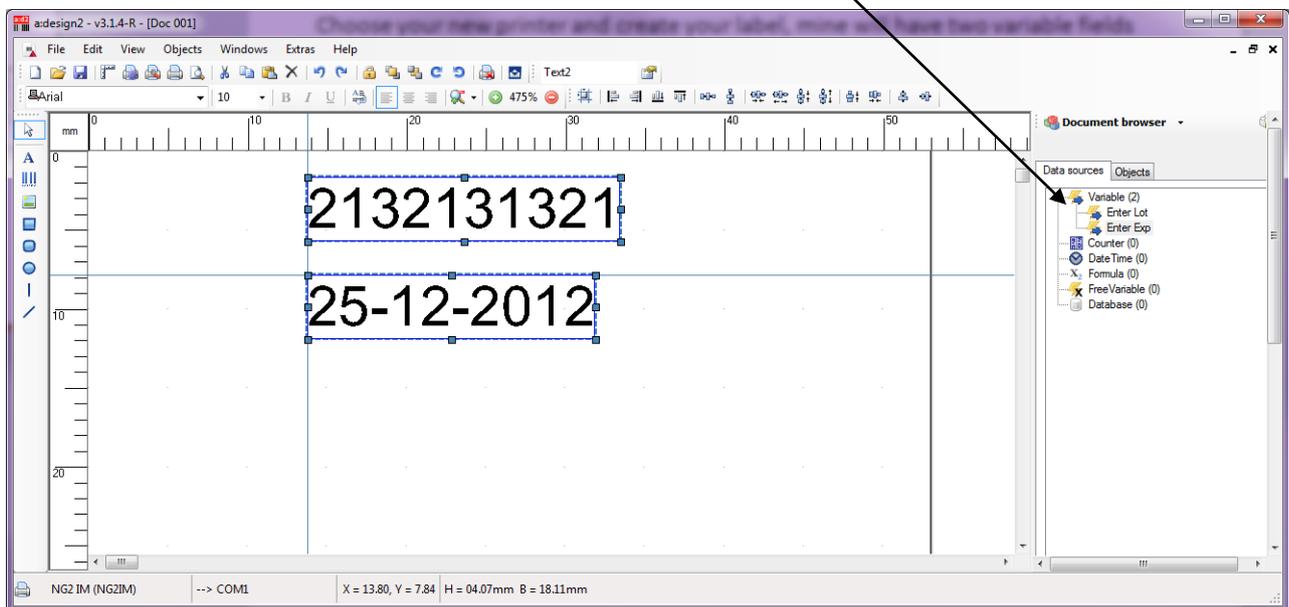
Click add



Up version 5

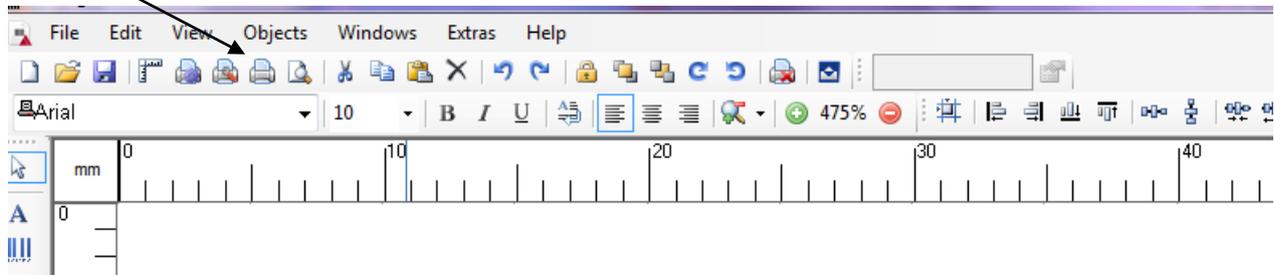


Choose your new printer and create your label, mine will have two variable fields

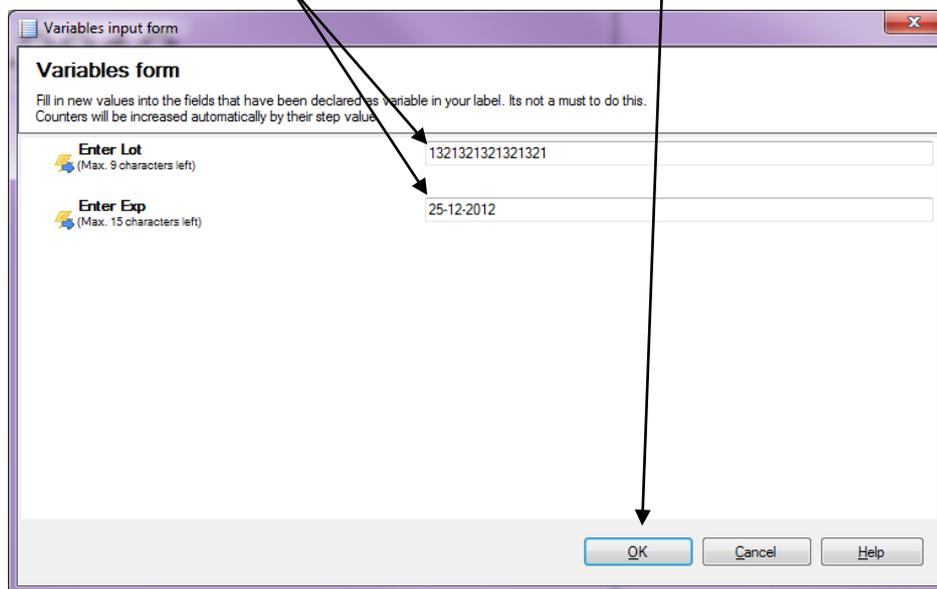


Once your label is set up you can send it directly to the printer and update just the variable information or you can print to a text file and use the ESC sequence and send it from the vision system.

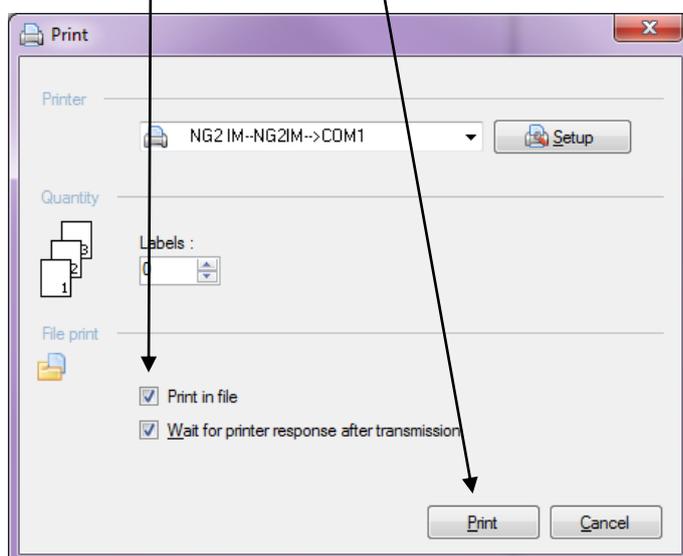
Select print



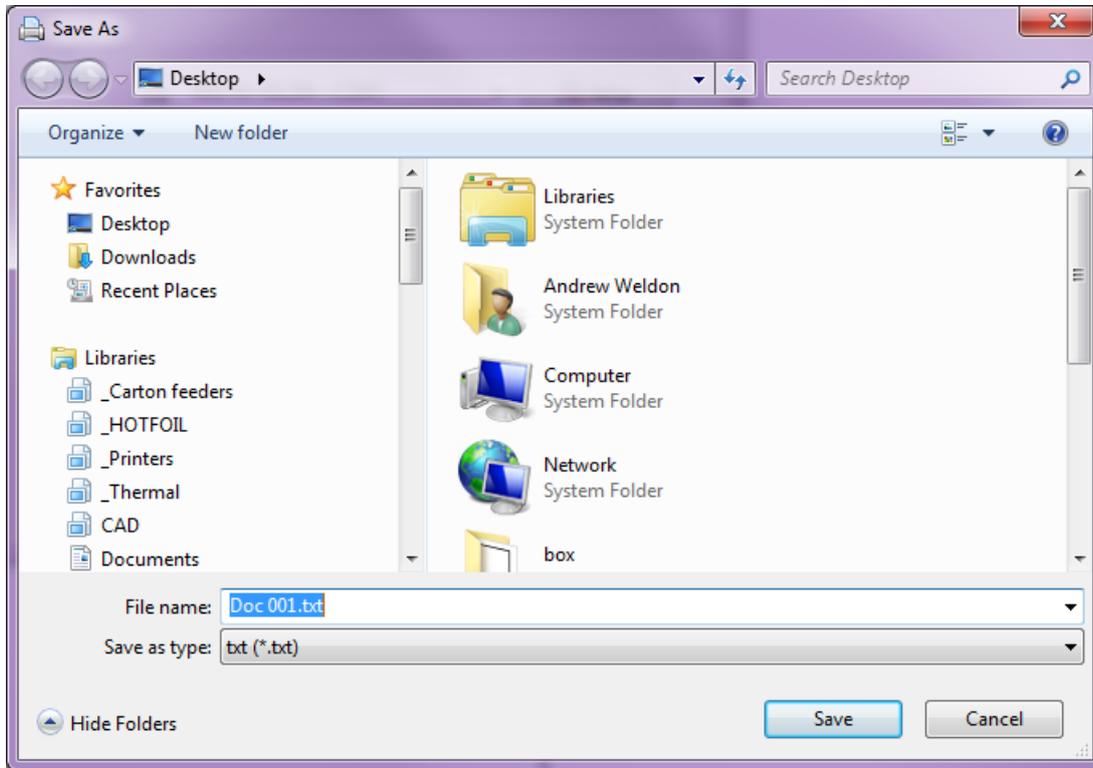
Update your variable information if required and Press OK



Select print to file and click the print button



You will be asked to save the file in a location on your PC



Once the ESC string or label 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 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 long time to update the printer for the next print job.

We tend to use this more when we are updating fields for every print such as a serialized barcode, the vision system would send us the information to populate the barcode for every print.

How the sequence should work

The next few slides will explain how the sequence should work, for this example I will talk about a plain vision system application.

Our vision system runs software – this software will load our complete label templates (text files). These templates are assigned to specific jobs.

An example of the label template in ESC sequence is below.

```
~1000194ESCDoc_001ESCVIEnter LotVT2132131321VTEnter LotVT0NU00000025VTVTESCVIEnter  
ExpVT25-12-2012VTEnter ExpVT0NU00000025VTVTESC#T05001660005600VIEnter  
LotVT10L000ESC#T05001660013000VIEnter ExpVT10L000ESC#S160ESCX07050yESCQ00000
```

Note:

When the string is accepted the printer will respond with ESC Z OK (because in the "Message Header" the acknowledge request flag "1 = Receipt wanted" is set). If there is an error the printer will respond with ZER. If "Announce Error" is activated (ESC X 55 1) and an error occurs, ESC ZERROR followed by the error number will be sent automatically when an error occurs, and <ESC>ZERCLR after the pending error has been cleared.

Attention! If you change the length of the label template string (e.g. change the content of a text field) the message size in the header must also change!

The label template is sent at the start of every batch. This loads the print template on the printer in the same way we load a print template when sending a label via a:design, NiceLabel, Codesoft, AllenCraft ...

After the batch is loaded it is no longer necessary to load the complete label for every print. We can just update the input variable fields (VI) by sending the following commands:

```
1 ~1000027ESCX56VIEnter LotVT3456789012VT
```

```
1 ~1000027ESCX56VIEnter ExpVT31.12.2013VT
```

The sequence broken down:

Each variable field is updated with ESCX56

The field is identified by the name "VIEnter Lot" and "VIEnter Exp"

The end of the variable name is identified by 0x0B (VT)

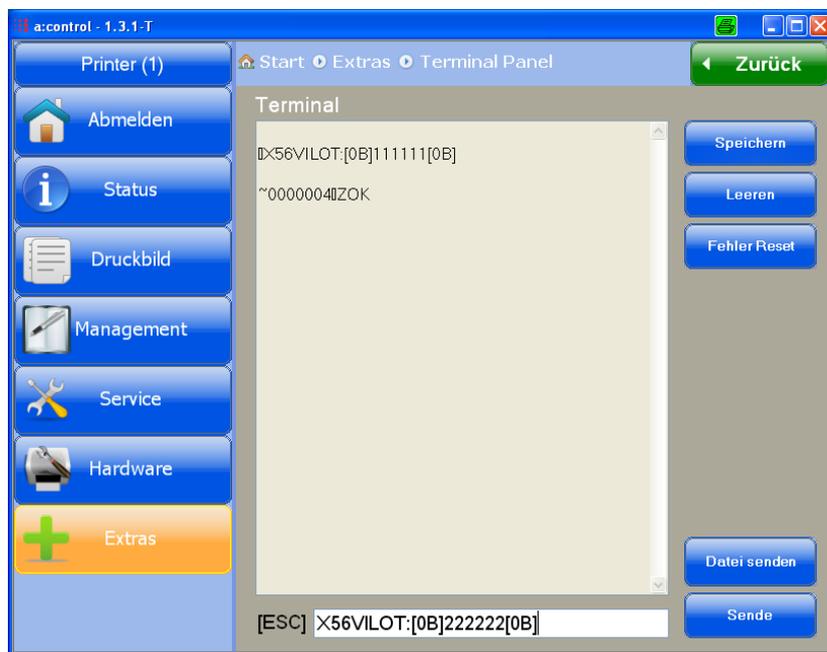
The end of the variable string is identified by 0x0B (VT)

ESC (0x1B) and VT (0x0B) are examples of the ASCII codes that should be used. To see the ESC string correctly a hex editor should be used.

Note:

You can also update the content of a "Free variable field" VX and the start value of a "Counter variable field" VC. For details on the ESCX56 command, refer to the ESC-Sequences documentation.

To test these commands you can use a:control to update variable strings (e.g. Input variable field “Lot:”)



Handshaking for change data after every print

It is also possible to receive a print complete message over the communication interface. This must be turned on first. To test this you can use the terminal panel in a:control.

Use the following ESC sequence:

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

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.

Example:

ESC string sent: X4731 Now after every print the printer will return ESCRD1.

This can be used to trigger the vision system to send new data for the next print.

If you want to make sure that the printer prints every format (variable) only once send a ESCQ00001 after update the variable string (for print quantity 1).

Set Speed and Density

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 “0xFF”), or as many times as required when sent independently.

You can preset the speed and density by adjusting the following part of the string:

```
~1000194ESC FDoc 001 . . . ESCVEnter LotVT2132131321VTEnter LotVT0NU00000025VTVTESC#T05001660005600VTEnter  
ExpVT25-12-2012VTEnter ExpVT0NU00000025VTVTESC#T05001660005600VTEnter  
LotVT10L000ESC#T05001660013000VTEnter ExpVT10L000ESC S160ESC X07050yESC Q00000
```

Speed = S160, 160 is the print speed in mm/sec be aware that modifying the esc string incorrectly will cause errors!

Density = X07050, 050 is the density in % be aware that modifying the esc string incorrectly will cause errors!

To set the speed and density independently use following commands:

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

Please have a look in the ESC-Sequences document for setting and request others print control parameters!

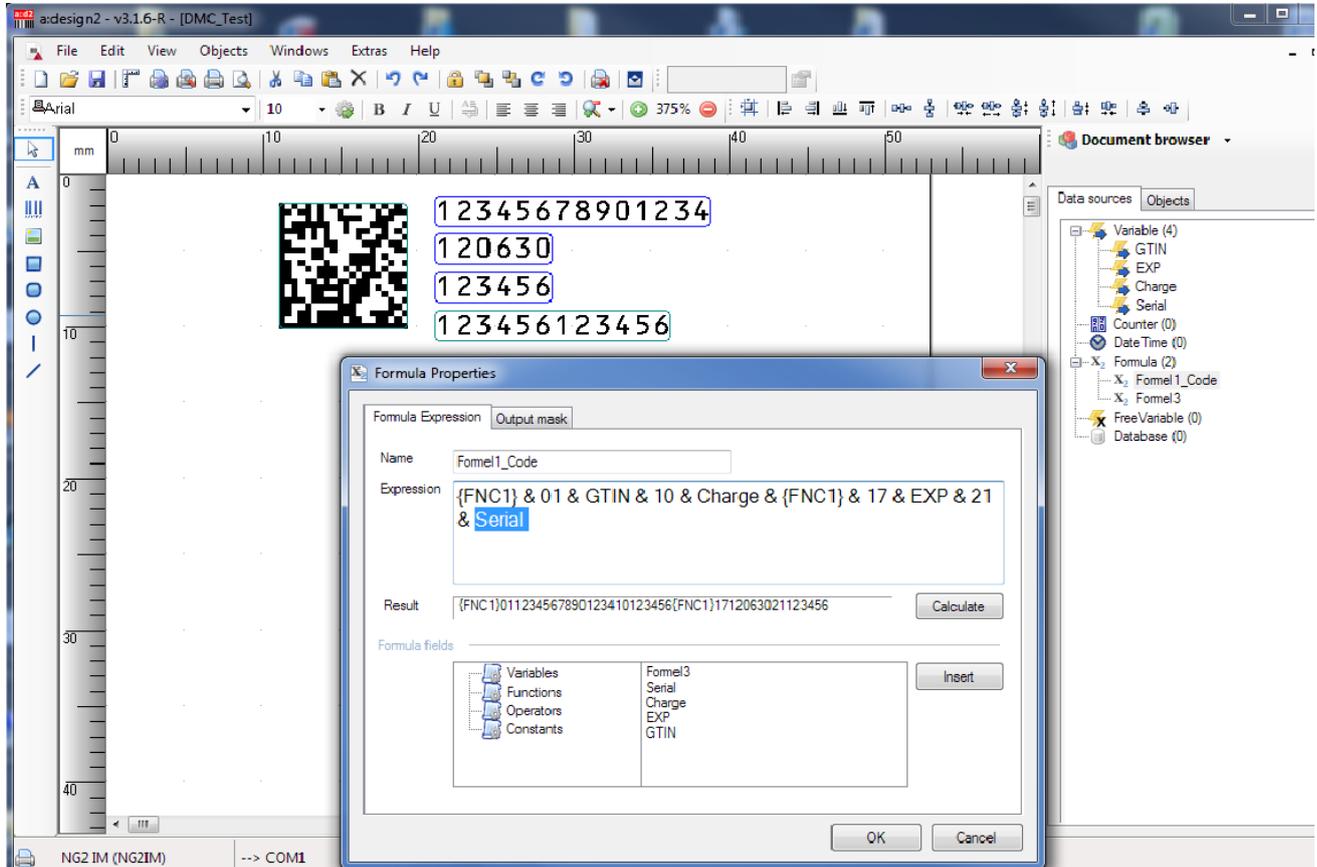
Example for serialization in GS1 Data Matrix Code

Another example to change the content from a variable field after every print is the serialization number integrated in a Data Matrix Code (Track & Trace application).

In this case we use an input variable (name: Serial) as a part of two formulas. One formula is then the basis of the Data Matrix Code. The other formula is the basis of a text field.

That means, if we change the content of the variable "Serial" the Data matrix code and the text field would both be changed.

The corresponding label template would be look as follows:



... and the same label template in ESC sequences as follows (this information must be sent at the start of the batch):

```

1 ~1000526ESC FDMC_Test · ESC VIGTINVT12345678901234VTGT
eingeben: VTL000140014VTVTESE VIEXPVT120630VTEXP.-Date
eingebenVT0NUL00060006VTVTESE VICHARGEVT123456VTCh.-Nr.
eingebenVT0NUL00060020VTVTESE VISERIALVT123456VTSe.-Nr.
eingebenVT0NUL00060030VTVTESE VFFORMEL1_CODEVT02.000000F
\<FNC1>VT&F01VT&VIGTINVT&F10VT&VICHARGEVT&F\<FNC1>VT&F1
7VT&VIEXPVT&F21VT&VISERIALVTVTESE VFFORMEL3VT02.000000VI
ChargeVT&VISERIALVTVTESE #T00002540003500VIGTINVT05L000
ESC#B07001330002200VFFORMEL1_CODEVT0500000020010ESC#T000
02540006600VIEXPVT05L000ESC#T00002540009600VICHARGEVT05L
000ESC#T00002540012600VFFORMEL3VT05L000ESC S160ESC X07050ÿ
ESC Q00000
    
```

Note:

If you don't want to change the used speed and density send the ESC sequence without this information. But don't forget to change the size in the header!

After the batch is loaded you can just update the input variable fields (VISerial) by sending the following commands:

```
1 ~1000020ESCX56VISerialVT789789VT
```

```
1 ~1000007ESCQ00001
```

If you have turned on the handshake with ESCX4731 you will receive ESCRD1 from the printer after the print is done. Now you can send the next data.

The printer is ready to print again after it has received the next data and the next ESCQ00001 command.