NDI series displays

with Profibus-DP UNI-TXT (ND) protocol

PARAMETRIZATION OF PROFIBUS-DP COMMUNICATION INTERFACE

CONTENT

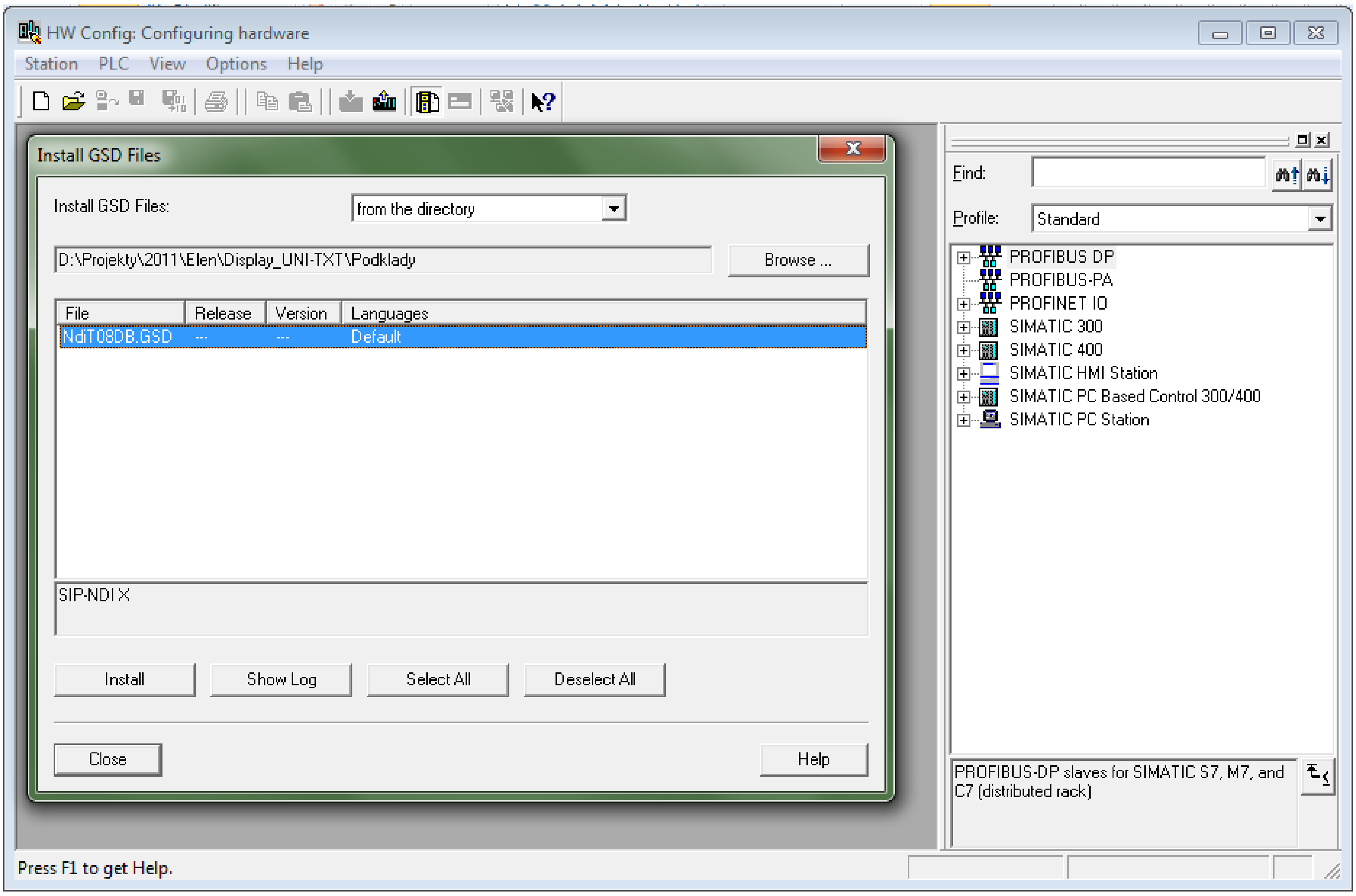
1. Introduction

This manual is aimed for programmers of SIEMENS, SIMATIC S7 control systems. In addition to this manual there is also example Step7 project “ELEN_UNI-TXT“, compiled in STEP7 ver. 5.5 programming environment, which is a development environment for SIMATIC S7-300 and S7-400 control system. Example Step7 project contains a simple configuration of SIMATIC S7-300 control system, consisting of CPU 315-2DP as a Profibus-DP Master and textual or alphanumeric display with communication protocol UNI-TXT and PROFIBUS-DP interface as a slave device.

2. GSD file installation

If we want to add display UNI-TXT into existing Step7 project, as a first step it is necessary to download the .GSD file and then add display UNI-TXT into HW catalog of STEP7. To do this open HW config editor in Step7 environment and in menu “Options – Install GSD file…” using function “Browse” open folder which contains file NdiT08DB.GSD, see figure 1 – Installing GSD file. Next, using the Install button, install GSD file into HW catalog of Step7 development environment. File NdiT08DB.GSD is available for download from ELEN company web site (www.elen.sk).

If you work with display NDI for the first time, we recommend to study the example Step7 project „ELEN_UNI-TXT“, or make a test workspace, that means: CPU with Profibus DP interface and display NDI, download example Step7 project into CPU and test display functions. There are two participants on the Profibus DP network. As the PROFIBUS-DP Master is used CPU 315-2DP, Profibus address 2. To Master is connected PROFIBUS-DP Slave display UNI-TXT with factory predefined Profibus address 125.



Install GSD Files: from the directory

D:\Projekty\2011\Elen\Display_UNI-TXT\Podklady

File	Release	Version	Languages
NdiT08DB.GSD	---	---	Default

SIP-NDI X

Find:

Profile: Standard

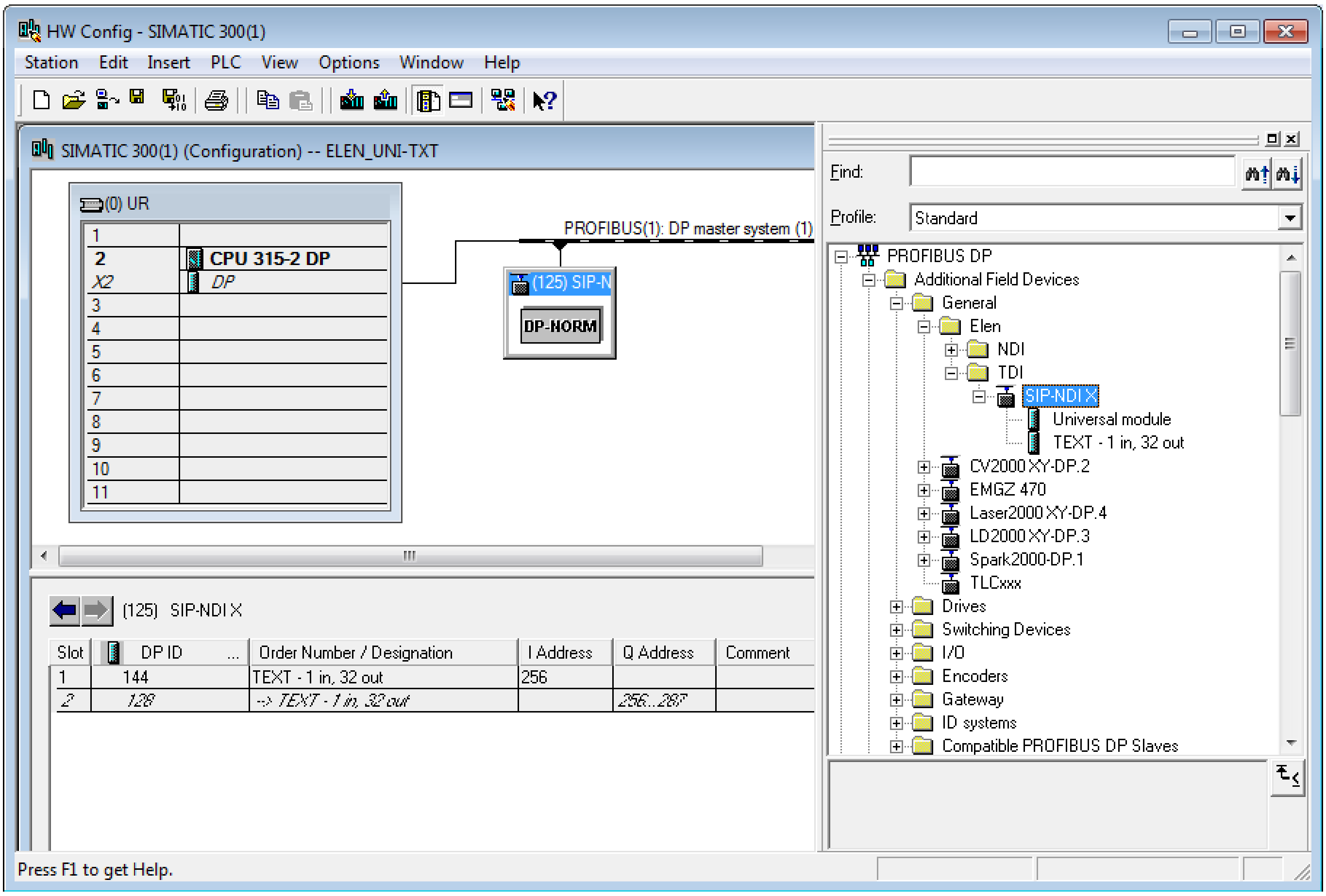
PROFIBUS-DP slaves for SIMATIC S7, M7, and C7 (distributed rack)

Press F1 to get Help.

Figure 1 – Installing GSD file

After successfully adding GSD file there will be a new Profibus slave device “SIP-NDI X” added into HW catalog, see figure 2 – HW catalog. In its virtual slots there are two blocks:

1. Input PLC ***„TEXT 1 in“*** – 1 byte
2. Output PLC ***“TEXT 32 out”*** – 32 byte



1	
2	CPU 315-2 DP
X2	DP
3	
4	
5	
6	
7	
8	
9	
10	
11	

Slot	DP ID	... Order Number / Designation	I Address	Q Address	Comment
1	144	TEXT - 1 in, 32 out	256		
2	*128*	*-> TEXT - 1 in, 32 out*		*256...287*	

Figure 2 – HW catalog

3. Example project ELEN_UNI-TXT (ND)

After installing GSD file into HW catalog of Step7 environment we can insert new DP slave „SIP-NDI X“ into HW configuration of existing Step7 project, or we can open example Step7 project ELEN_UNI-TXT_Profibus. In this example project there are two functional blocks in the programming folder „Blocks“ for controling display. First is FB1. To call it, it is necessary to enter the following input-output parameters:

```
CALL  "UNI-TXT_full" , "IDB_FB1"
 PerAdr  :=256                  // Periferal address from HW configuration
 Blocks  :=5                    // Number of transfered 30 byte blocks
 DataDisp:="DATA_DISP"          // Data block with texts for display
 Write   :="Start_disp"         // Start bit for writing from DataDisp into display buffer
```

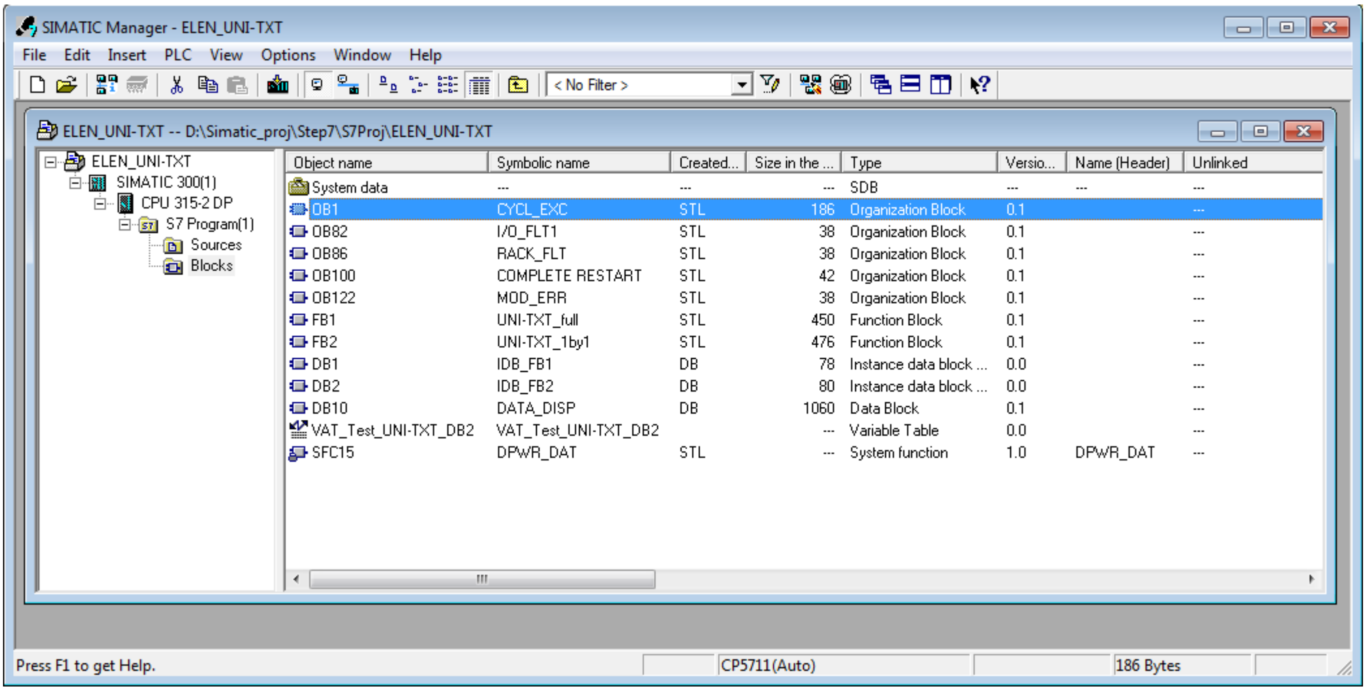
Using the Profibus-DP interface, this function block transfers required number of 30-byte blocks (range 1 to 34) from Simatic to display buffer memory. Text data for display are in Simatic stored in data block DB10 ("DATA_DISP"). Display buffer memory has size 1024 bytes and contains text strings, which should be displayed on display and control commands of UNI-TXT protocol. Setting the start bit „Write“ triggers transfer of given number of 30 byte blocks and at the end FFFFHex is sent for displaying the content of display buffer. Bit “Write“ is automatically reset after the end of transfer. Profibus writing is realized by calling system function SFC15 /DPWR_DAT/. The entire communication, that is transfer of max. 34 blocks is done in 1 PLC cycle, what could in some time critical PLC programs cause unacceptable duration of time cycle.

For this reason there is in the example project also a second block FB2, whose function is similar, but in one PLC cycle there is only 1 block with 30 bytes transfered into display buffer. Below is example on how to call block FB2 wiht corresponding instant data block DB2.

```
CALL  "UNI-TXT_1by1" , "IDB_FB2"
    PerAdr  :=256                    //Periferal address from HW configuration
    Blocks  :=5                      //Number of transfered 30-byte blocks
    DataDisp:="DATA_DISP"            //Data block with text for display
    Write   :="Start_disp"           //Start bit for writing data from DataDisp into display buffer.
    Active  :="Start_disp_active"    //Flag bit, transfer of data is active
```

Calling FB1 with corresponding instant data block DB1 (or FB2, DB2) is in the main program cycle OB1. In starting block OB100 is instruction to set bit „Write“, which serves for automatic displaying of text when PLC is starting.

List of all blocks from example project is in Fig. 3 – Step7 project ELEN_UNI-TXT.



Object name	Symbolic name	Created...	Size in the ...	Type	Versio...	Name (Header)	Unlinked
System data	---	---	---	SDB	---	---	---
OB1	CYCL_EXC	STL	186	Organization Block	0.1		---
OB82	I/O_FLT1	STL	38	Organization Block	0.1		---
OB86	RACK_FLT	STL	38	Organization Block	0.1		---
OB100	COMPLETE RESTART	STL	42	Organization Block	0.1		---
OB122	MOD_ERR	STL	38	Organization Block	0.1		---
FB1	UNI-TXT_full	STL	450	Function Block	0.1		---
FB2	UNI-TXT_1by1	STL	476	Function Block	0.1		---
DB1	IDB_FB1	DB	78	Instance data block ...	0.0		---
DB2	IDB_FB2	DB	80	Instance data block ...	0.0		---
DB10	DATA_DISP	DB	1060	Data Block	0.1		---
VAT_Test_UNI-TXT_DB2	VAT_Test_UNI-TXT_DB2		---	Variable Table	0.0		---
SFC15	DPWR_DAT	STL	---	System function	1.0	DPWR_DAT	---

Figure. 3 – Step7 project ELEN_UNI-TXT

4. <u>Control commands and text format</u>

As was mentioned in section 2, communication is realized through 1 input byte and 32 output bytes. Output block has in first 2 bytes location in buffer memory and in next 30 bytes is text content including control signs. Writing into buffer memory is done after 30 byte blocks. After sending all blocks with text data, the content of buffer memory is displayed on display when sending message, which has FFFF Hex in first 2 bytes. In case of communication loss, e.g. the Profibus communication line gets disconnected, display will show text information which was sen the last time.

Data content consists of commands for display to work with specific text information. These are separated with **$** sign.

4.1. Commands for writing text

4.1.1. Direct writing of text

Display will show required text message directly, while its previous content of the entire line is erased. If exceeding the display line length, writing continues on the next display line automaticaly.

Abcdef	simple writing of ASCII text characters, which can also contain commands for text formatting described in chapter 4.2.

4.1.2. Clearing display

$0	Clearing of user display area and all text message attributes

4.1.3. Writing variable to a specific location

$P<p10X><p1X><n10X><n1X>text

<p10X><p1X> – starting position of text in ASCII format
range 0 – 64th position
<n10X><n1X> – number of bytes to display
range 0 – 64th position
text – character string, which should be displayed

Character string can contain numbers, some alphanumeric characters and decimal point. Code 0x2C and also code 0x2D can be used to display decimal point. Decimal point is cosidered as independent character. To display decimal point it you can use empty space character and decimal point character.

4.2. Commands for message formatting

One sending frame can contain more commands of this type, commands can be nested only into command of direct writing of text, see chapter 4.1.1.

4.2.1. Controlling text blinking

$F1	following characters will be blinking
$F0	following characters will NOT be blinking ASCII format, default 0

Command $F is valid until the next change of value with $F command, or until display is turned off, or until changing settings to default manufacturing parameters with command $R.

4.2.2. Changing text message color

This is valid for multi-color LED displays only.

Text message color (default value is C1):

$C1	red
$C2	green
$C3	yellow

4.3. Commands for global functions of display

After reset, default values are set and command lasts until rewritten with a new command. Settings of these parameters will stay the same even if the display is restarted. Each sending frame can contain only one command.

4.3.1. Brightness control

$B<type of brightness control><brightness level>

<typ riadenie jasu> – je znak typu nastavenia jasu
0 – brightness control by setting direct value of PWM without automatic control
1 – brightness control by setting the steepness of regulation curve in ASCII format, default 0
<brightness level> – level value, while bit D7 is 1, range 0 – 100%, default 100

4.3.2. Setting communication Time-out

$T<time-out value>

<time-out value> – value of time-out from 0 to 127, while bit D7 is 1
range 0 – 127 seconds, default 0

4.3.3. Real time synchronization packet

Display will set the internal real time according to content of the synchronization packet. Display is expecting information about time and date, in a format, which contains correction for time zone and DST (summer/winter time).

$S<Y10X><Y1X><M10X><M1X><D10X><D1X>H10X><H1X><MIN10X><MIN1X><SEC10X><SEC1X>

<Y10X>	tens of year in ASCII format
<Y1X>	ones of year in ASCII format
<M10X>	tens of month in ASCII format
<M1X>	ones of month in ASCII format
<D10X>	tens of day in ASCII format
<D1X>	ones of day in ASCII format
<H10X>	tens of hours in ASCII format
<H1X>	ones of hours in ASCII format
<MIN10X>	tens of minutes in ASCII format
<MIN1X>	ones of minutes in ASCII format
<SEC10X>	tens of seconds in ASCII format
<SEC1X>	ones of seconds in ASCII format

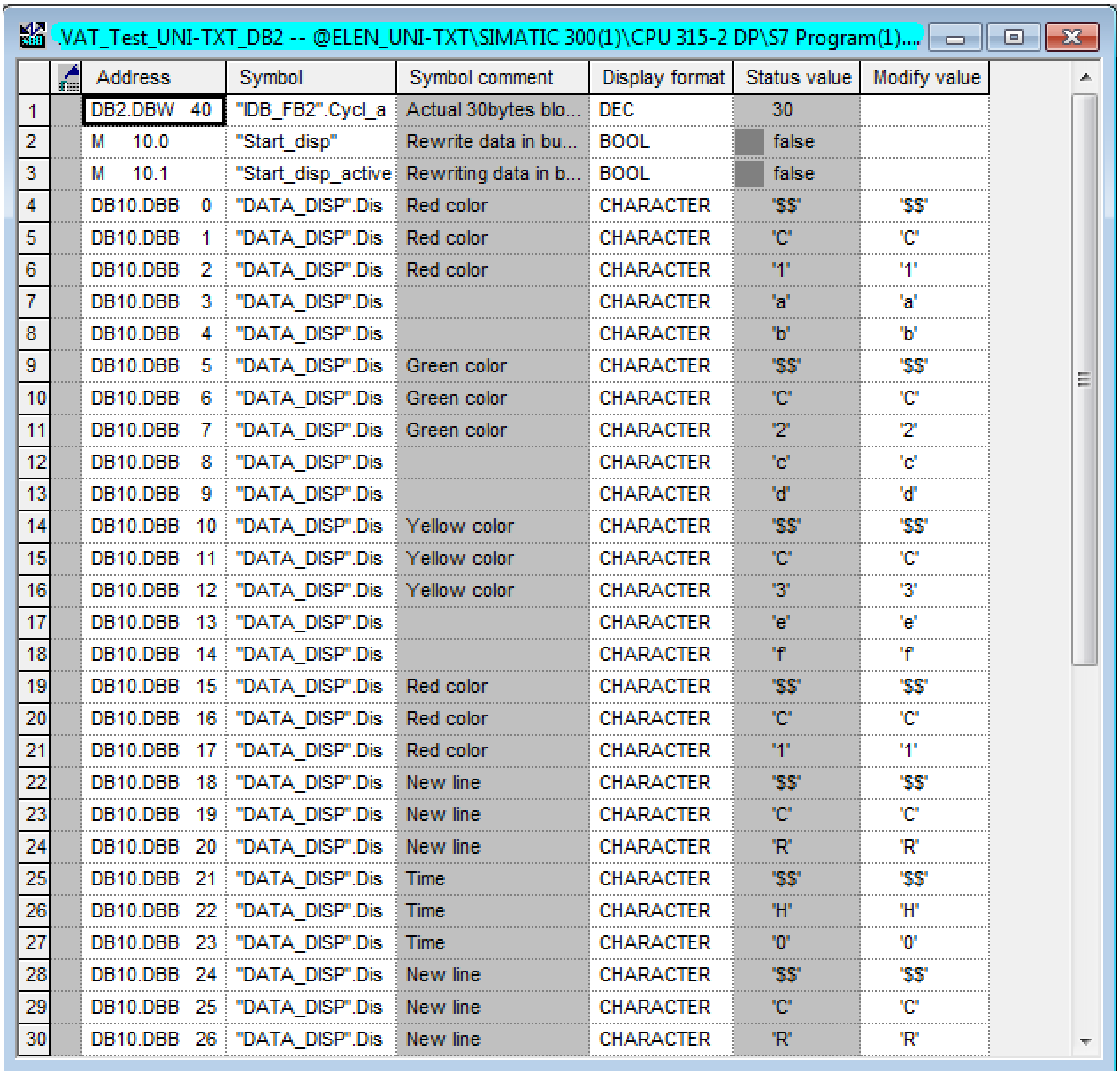
4.3.4. Blinking text parameters

$G<period><filling>

<period> – blinking time period x 100 ms
range 0 – 127 (that means 0.1 – 12.7 seconds), value of bit D7 is 1, default 5
<filling> – what percentage of blinking period is display ON
range 0 – 100%, value of bit D7 is 1, default 50

5. VAT table

In Fig. 4 – Vat_Test_UNI-TXT is VAT table for rewriting variables, declared when calling FB1 in main block OB1. It can be used to test all display functions.



VAT_Test_UNI-TXT_DB2 -- @ELEN_UNI-TXT\SIMATIC 300(1)\CPU 315-2 DP\S7 Program(1)....

		Address	Symbol	Symbol comment	Display format	Status value	Modify value
1		DB2.DBW 40	"IDB_FB2".Cycl_a	Actual 30bytes blo...	DEC	30	
2		M 10.0	"Start_disp"	Rewrite data in bu...	BOOL	■ false	
3		M 10.1	"Start_disp_active	Rewriting data in b...	BOOL	■ false	
4		DB10.DBB 0	"DATA_DISP".Dis	Red color	CHARACTER	'$$'	'$$'
5		DB10.DBB 1	"DATA_DISP".Dis	Red color	CHARACTER	'C'	'C'
6		DB10.DBB 2	"DATA_DISP".Dis	Red color	CHARACTER	'1'	'1'
7		DB10.DBB 3	"DATA_DISP".Dis		CHARACTER	'a'	'a'
8		DB10.DBB 4	"DATA_DISP".Dis		CHARACTER	'b'	'b'
9		DB10.DBB 5	"DATA_DISP".Dis	Green color	CHARACTER	'$$'	'$$'
10		DB10.DBB 6	"DATA_DISP".Dis	Green color	CHARACTER	'C'	'C'
11		DB10.DBB 7	"DATA_DISP".Dis	Green color	CHARACTER	'2'	'2'
12		DB10.DBB 8	"DATA_DISP".Dis		CHARACTER	'c'	'c'
13		DB10.DBB 9	"DATA_DISP".Dis		CHARACTER	'd'	'd'
14		DB10.DBB 10	"DATA_DISP".Dis	Yellow color	CHARACTER	'$$'	'$$'
15		DB10.DBB 11	"DATA_DISP".Dis	Yellow color	CHARACTER	'C'	'C'
16		DB10.DBB 12	"DATA_DISP".Dis	Yellow color	CHARACTER	'3'	'3'
17		DB10.DBB 13	"DATA_DISP".Dis		CHARACTER	'e'	'e'
18		DB10.DBB 14	"DATA_DISP".Dis		CHARACTER	'f'	'f'
19		DB10.DBB 15	"DATA_DISP".Dis	Red color	CHARACTER	'$$'	'$$'
20		DB10.DBB 16	"DATA_DISP".Dis	Red color	CHARACTER	'C'	'C'
21		DB10.DBB 17	"DATA_DISP".Dis	Red color	CHARACTER	'1'	'1'
22		DB10.DBB 18	"DATA_DISP".Dis	New line	CHARACTER	'$$'	'$$'
23		DB10.DBB 19	"DATA_DISP".Dis	New line	CHARACTER	'C'	'C'
24		DB10.DBB 20	"DATA_DISP".Dis	New line	CHARACTER	'R'	'R'
25		DB10.DBB 21	"DATA_DISP".Dis	Time	CHARACTER	'$$'	'$$'
26		DB10.DBB 22	"DATA_DISP".Dis	Time	CHARACTER	'H'	'H'
27		DB10.DBB 23	"DATA_DISP".Dis	Time	CHARACTER	'0'	'0'
28		DB10.DBB 24	"DATA_DISP".Dis	New line	CHARACTER	'$$'	'$$'
29		DB10.DBB 25	"DATA_DISP".Dis	New line	CHARACTER	'C'	'C'
30		DB10.DBB 26	"DATA_DISP".Dis	New line	CHARACTER	'R'	'R'

Fig. 4 – Vat_Test_UNI-TXT