CDN Series Displays

with Profibus-DP ELEN UNI-TXT (ND) protocol

PARAMETRIZATION OF PROFIBUS-DP COMMUNICATION INTERFACE

CONTENT

Communication protocol description.

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 an existing Step7 project, as a first step it is necessary to download the .GSD file and then add display UNI-TXT to 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 provided on the CD with display. It can be also downloaded from ELEN web site:
http://www.elen.sk/files/download/ndit08db.zip

If you work with display NDI for the first time, we recommend you to study the example Step7 project „ElenCDN100", or make a test workspace, which 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 in the example project. As PROFIBUS-DP **Master** is used **CPU 315-2DP, Profibus address 2**. To Master is connected PROFIBUS-DP **Slave display CDN100 with factory predefined Profibus address 8**. (Profibus address can be set different as per request, or can be changed by customer with service programming cable.)

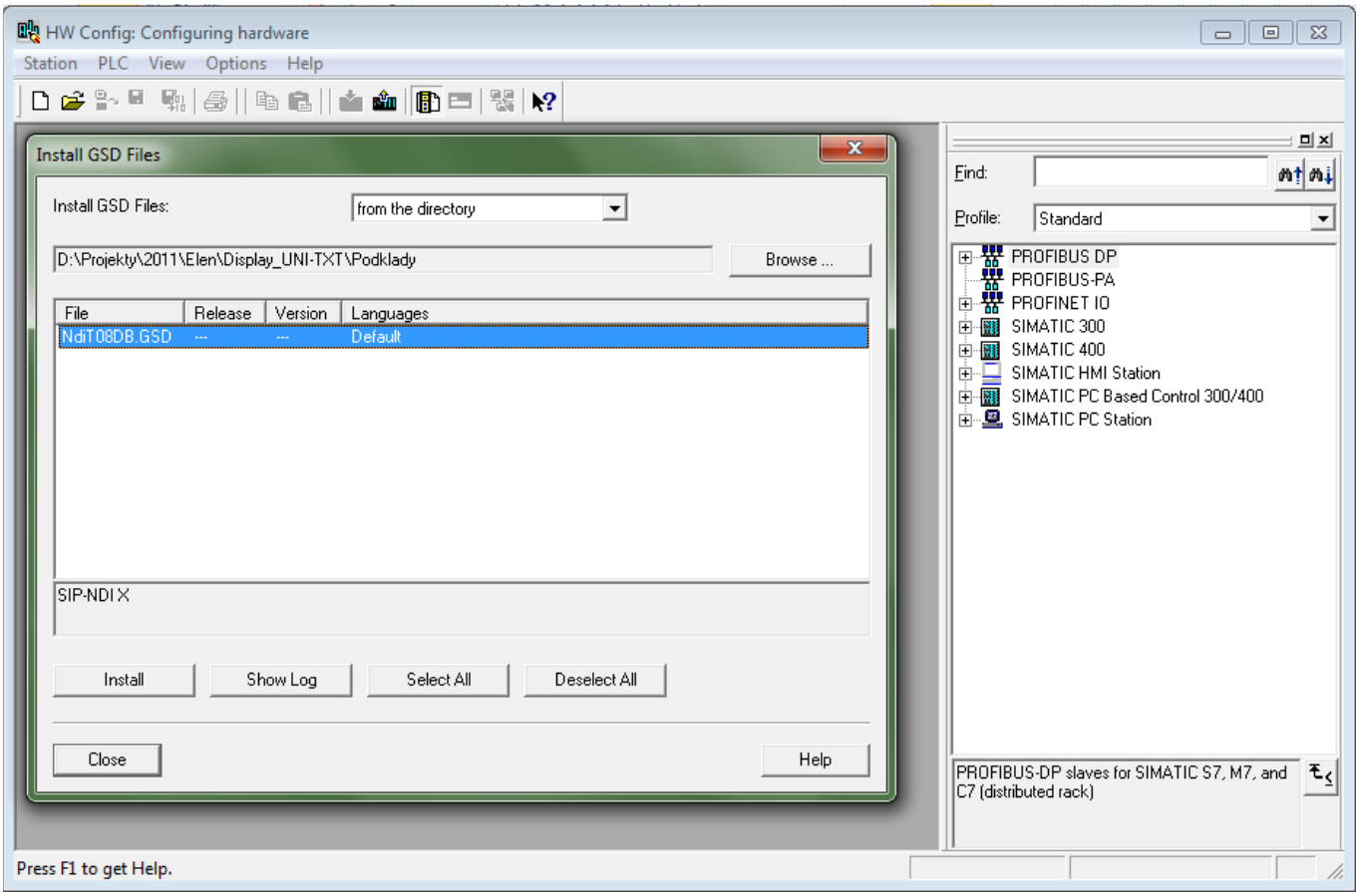


Figure 1 – Installing GSD file

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

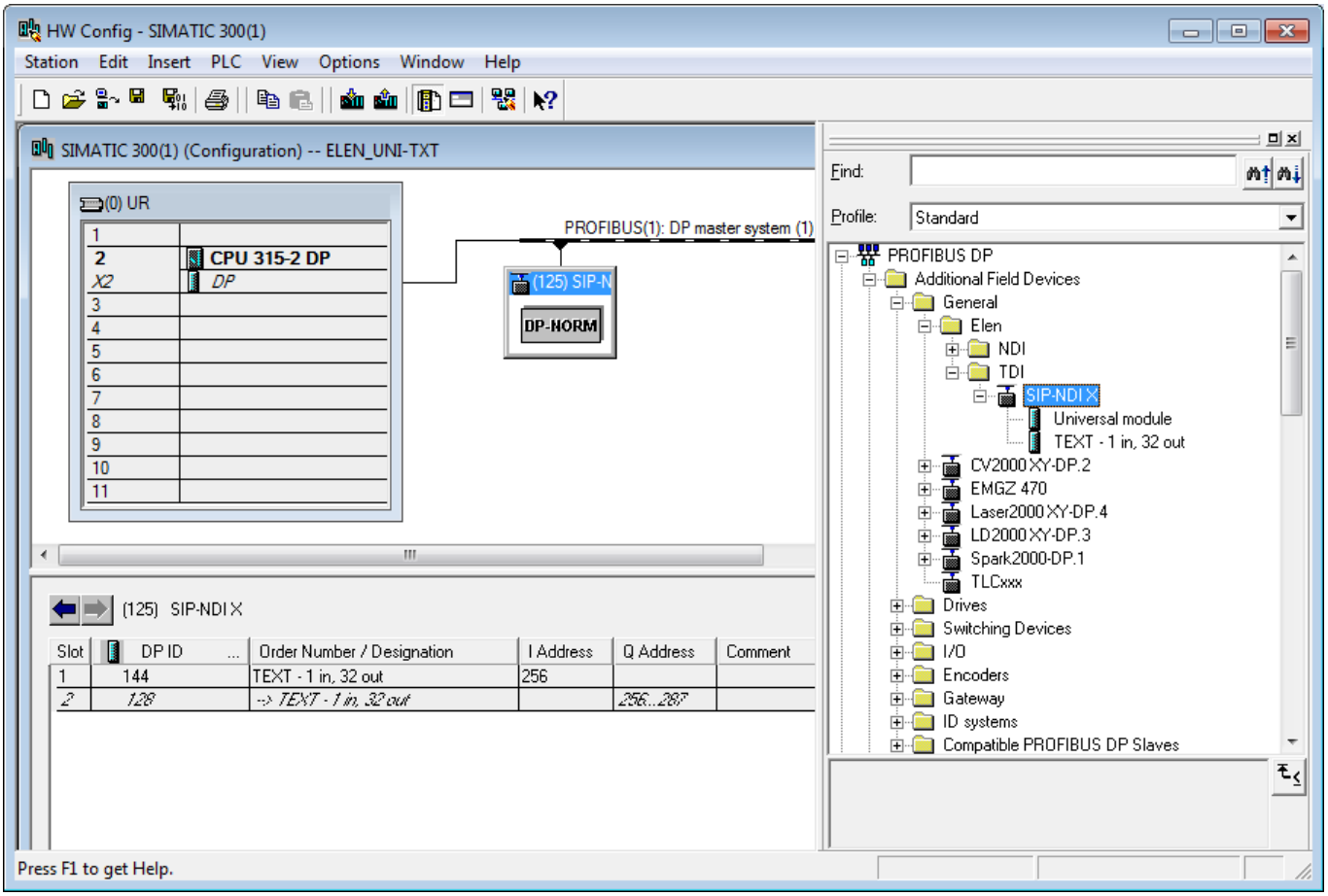


Figure 2 – HW catalog

The PLC output address space “**TEXT 32 out**“ is used for writing data into display. Display CDN100 is equipped with a control module which controls display. This module is connected to display’s Profibus DP Slave module via internal serial bus RS485. Control module contains buffer for storing characters which are displayed on display. **Buffer size of display’s control module for numerical displays is 128 bytes**. Since Profibus DP Slave module can send a **maximum of 32 bytes**, **communication between PLC and display is based on sending 32-byte blocks**.

In the first 2 bytes of address space “**TEXT 32 out**” is a number, 16 bit integer, which is a pointer to byte address position in the buffer to which we are writing. Remaining 30 bytes is used for the data itself. Communication between PLC and Profibus DP Slave module is done in cycles. If we want to completely fill the buffer of the CDN100 display, we have to transfer 4x30 bytes + 8 bytes, which is 5 blocks. Each block is being written to Profibus DP Slave module of display in one cycle OB1 – Main PLC. Individual blocks are stored in the buffer of Profibus DP Slave module. **The Profibus Slave module buffer has 1024 bytes**, because Profibus DP Slave module is universal and is also used for displays with larger buffer size (e.g. textual displays).

After transferring the last block into Profibus DP Slave module, in the next PLC cycle is sent telegram, which contains “**FFFF” Hex in the first 2 bytes** of address space “**TEXT 32 out**”. Remaining 30 bytes, used for data, are ignored in the Profibus DP Slave module. This special telegram is used as a command to start writing data, transmitted in all previous blocks from Profibus DP Slave module into display’s control module via serial bus RS485.

The PLC input address space “**TEXT 1 in**” is used for reading display’s status. When trying to write above the pointer’s range (in the first 2 bytes “**TEXT 32 out**” is number higher than 994) Profibus DP Slave modul will return character **NAK – “15” Hex**.

When writing with correct value of the pointer, Profibus DP Slave module will return “**00” Hex**. After sending the last telegram **“FFFF”** Hex, transfer of data will start and internal timer will be also started

(about 10 sec.). Transfer of data between display's Profibus DP Slave module and display's control module lasts 10 to 1000 ms, depending on size of transferred data and the speed of the RS485 bus line.

If data are transferred ok, display's control module will send via RS485 bus line confirmation – character **ACK** – "**06**" **Hex** to the Profibus DP Slave module, which will write it into address space "**TEXT 1 in**", which is available for the PLC.

If data are not transferred ok and there is timeout of the internal timer in the Profibus DP Slave module, into the address space of "**TEXT 1 in**" will be written value **"FF" hex**, which means **timeout**. Timeout will also occur in case there is a hardware problem in display (problem on the internal bus line, display board, power supply, or if display board will not accept received data – e.g. exceeding buffer capacity of displaying board, etc.)

3. Example project ElenCDN100

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_NDI.
In this example Step7 project there is in the programming folder „**Blocks**" example function block **FB1** for controlling display. To call it, it is necessary to enter the following input-output parameters:

This function block transfers via Profibus DP interface required number (in the range of 1 to 33) of 30 byte blocks from Simatic to buffer memory of display. Display CDN100 makes possible to transfer a maximum of 5 blocks. Textual information for display are in Simatic stored in data block DB10 ("DATA_DISP"). This DB has a size 1024 bytes and contains textual strings, which should be displayed on display and control command characters of the protocol ELEN UNI-TXT. Setting the start bit "Write" will start the transfer of predefined number of 30 byte blocks and at the end FFFF Hex is sent for displaying the content of buffer on display. After transfer is finished the Write bit will be automatically reset inside the FB body.

The Profibus write-up itself is realized by calling of system function SFC15 (DPWR_DAT). Entire communication, that is transfer of 33 blocks max., is made one block at a time, one block in 1 PLC cycle.

```
CALL  "UNI-TXT" , "IDB_FB2"
 PerAdr  :=256                  // Peripheral address from HW configuration
 Blocks  :=5                    // Number of transferred 30 byte blocks
 DataDisp:="DATA_DISP"          // Data block with texts for display
 Timer_timeout:=T100            // Timer for timing timout error
 Time_timeout:=S5T#2S           // Timing constant for timing timout error
 Write   :="Start_disp"         // Start bit for writing data from DataDisp into display buffer
 Active  :="Start_disp_active"  // Flag bit meaning "transfer of data is running"
 Timeout :="Timeout"            // Output parameter for identification of timeout error
 StatDisp :="StatusByteDisp"    // Output parameter for display status (content "TEXT 1 in")
```

Profibus DP display interface has besides 32 byte output interface (output from the PLC viewpoint) also the output interface with 1 byte size for signaling the communication status. After power up of display this byte is 00 Hex and during communication this byte can contain values: 06Hex, 15Hex, FFHex, read explanation above.

Calling FB2 with corresponding instant data block DB2 is in the main program cycle OB1

List of all blocks from the example project is in Fig. 3 – Step7 project ElenCDN100.

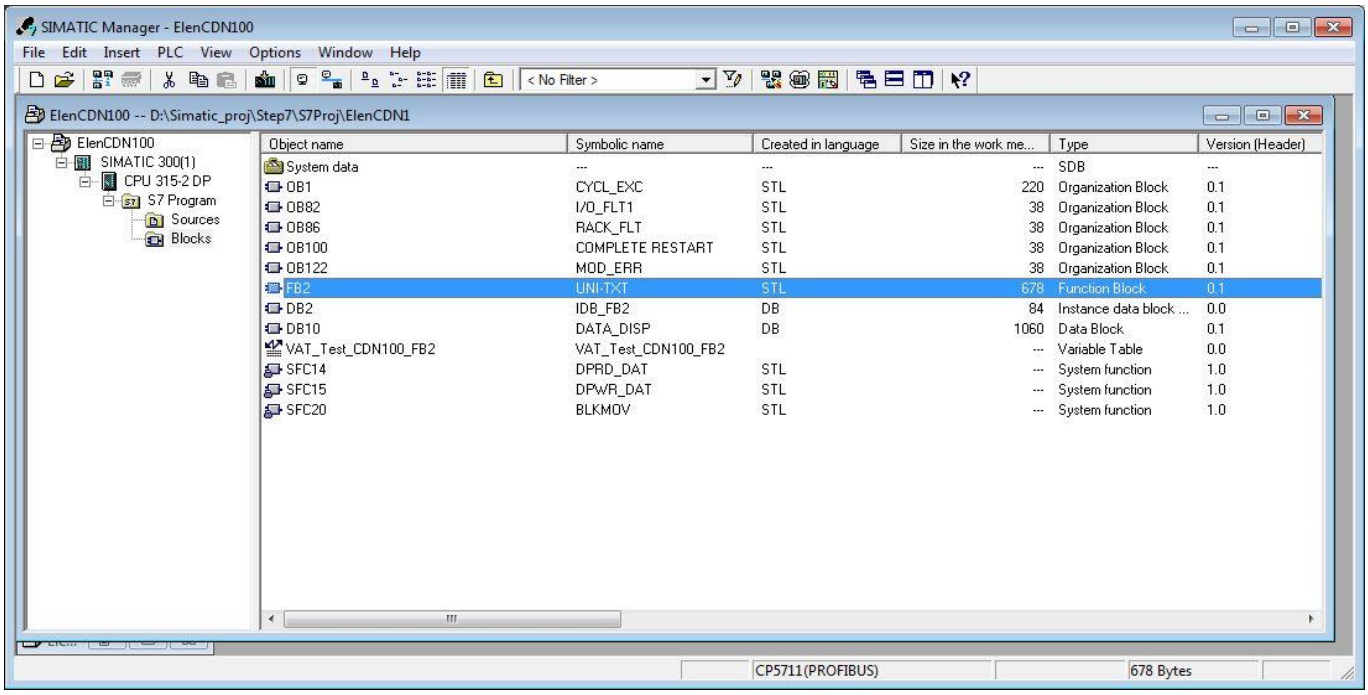


Figure 3 – Step7 project ElenCDN100

4. Control commands and text format

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 sent the last time.

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

To send the next message it is important to wait at least 100 ms.

4.1. Commands for writing text

Each sending frame can have only one command.

4.1.1 Direct writing of text

Display will show required text message directly, while its previous content is erased.

tttttt simple writing of ASCII text characters, which can also contain commands for text formatting described in chapter 4.2 (blinking of message, change of color, clearing display).

4.1.2 Writing text to exact position

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

<p10X><p1X> – beginning position of ASCII text format
range 0. – 64. position
<n10X><n1X> – number of bytes to be displayed
range 0. – 64. position
text – characters string, which has to be displayed

String characters can contain numbers, some special alphanumeric characters and decimal point. To display decimal point it is possible to use code 0x2C as well as 0x2D. Decimal point character is considered as a stand-alone character. It is possible to use sequence: empty character and decimal point character to display a decimal point only.

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. Blinking of message

$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 and message background

This is valid for multi-color LED displays only.

Text message color (default value is C1):

$C1 red
$C2 green
$C3 yellow

4.2.3. Clearing display

$0 (zero)

Clearing of user display area and all text attributes. Display clearing does not affect the part of display which shows information about time or date.

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>

< type of brightness control > – brightness control type sign
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 80

4.3.2. Setting message TIMEOUT

$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

After TIME-OUT expiration hyphens will be displayed in the user area of display, which can control that displayed data are actual. TIME-OUT does not affect the part of display, which shows information about time or date. Setting the TIME-OUT value to 0, causes data to be displayed permanently.

4.3.3. Blinking text parameters

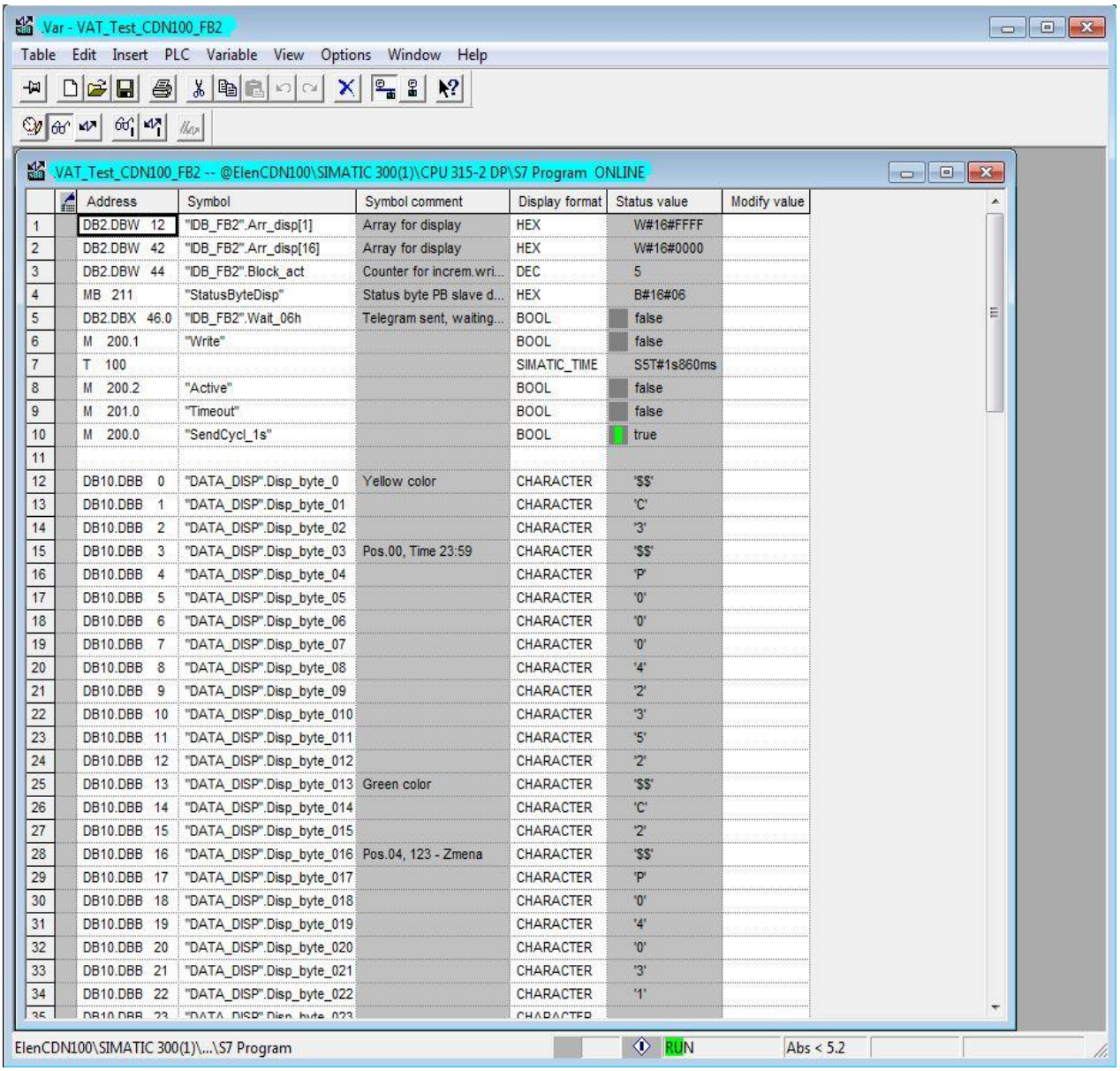
$G<period><filling>

<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 66

5. VAT table

In Fig. 4 – Vat_Test_CDN100_FB2 is VAT table for rewriting variables, declared when calling FB2 in main block OB1. Table can be used to test display functions. Setting the “Write” bit will execute onetime data writing from DB10 to display. Setting bit “SendCycl_1s” will start writing of data to display repeatedly with period of 1 sec.

It can be used to test all display functions.



	Address	Symbol	Symbol comment	Display format	Status value	Modify value
1	DB2.DBW 12	"IDB_FB2".Arr_disp[1]	Array for display	HEX	W#16#FFFF	
2	DB2.DBW 42	"IDB_FB2".Arr_disp[16]	Array for display	HEX	W#16#0000	
3	DB2.DBW 44	"IDB_FB2".Block_act	Counter for increm.wri...	DEC	5	
4	MB 211	"StatusByteDisp"	Status byte PB slave d...	HEX	B#16#06	
5	DB2.DBX 46.0	"IDB_FB2".Wait_06h	Telegram sent, waiting...	BOOL	false	
6	M 200.1	"Write"		BOOL	false	
7	T 100			SIMATIC_TIME	S5T#1s860ms	
8	M 200.2	"Active"		BOOL	false	
9	M 201.0	"Timeout"		BOOL	false	
10	M 200.0	"SendCycl_1s"		BOOL	true	
11						
12	DB10.DBB 0	"DATA_DISP".Disp_byte_0	Yellow color	CHARACTER	'$$'	
13	DB10.DBB 1	"DATA_DISP".Disp_byte_01		CHARACTER	'C'	
14	DB10.DBB 2	"DATA_DISP".Disp_byte_02		CHARACTER	'3'	
15	DB10.DBB 3	"DATA_DISP".Disp_byte_03	Pos.00, Time 23:59	CHARACTER	'$$'	
16	DB10.DBB 4	"DATA_DISP".Disp_byte_04		CHARACTER	'P'	
17	DB10.DBB 5	"DATA_DISP".Disp_byte_05		CHARACTER	'0'	
18	DB10.DBB 6	"DATA_DISP".Disp_byte_06		CHARACTER	'0'	
19	DB10.DBB 7	"DATA_DISP".Disp_byte_07		CHARACTER	'0'	
20	DB10.DBB 8	"DATA_DISP".Disp_byte_08		CHARACTER	'4'	
21	DB10.DBB 9	"DATA_DISP".Disp_byte_09		CHARACTER	'2'	
22	DB10.DBB 10	"DATA_DISP".Disp_byte_010		CHARACTER	'3'	
23	DB10.DBB 11	"DATA_DISP".Disp_byte_011		CHARACTER	'5'	
24	DB10.DBB 12	"DATA_DISP".Disp_byte_012		CHARACTER	'2'	
25	DB10.DBB 13	"DATA_DISP".Disp_byte_013	Green color	CHARACTER	'$$'	
26	DB10.DBB 14	"DATA_DISP".Disp_byte_014		CHARACTER	'C'	
27	DB10.DBB 15	"DATA_DISP".Disp_byte_015		CHARACTER	'2'	
28	DB10.DBB 16	"DATA_DISP".Disp_byte_016	Pos.04, 123 - Zmena	CHARACTER	'$$'	
29	DB10.DBB 17	"DATA_DISP".Disp_byte_017		CHARACTER	'P'	
30	DB10.DBB 18	"DATA_DISP".Disp_byte_018		CHARACTER	'0'	
31	DB10.DBB 19	"DATA_DISP".Disp_byte_019		CHARACTER	'4'	
32	DB10.DBB 20	"DATA_DISP".Disp_byte_020		CHARACTER	'0'	
33	DB10.DBB 21	"DATA_DISP".Disp_byte_021		CHARACTER	'3'	
34	DB10.DBB 22	"DATA_DISP".Disp_byte_022		CHARACTER	'1'	
35	DB10.DBB 23	"DATA_DISP".Disp_byte_023		CHARACTER		

Figure 3 – Vat_Test_CDN100_FB2

Communication protocol description

Converter of Profibus DP to UNI TXT protocol

Converter Profibus DP is a SLAVE type device connected between Profibus DP interface and display control module.

Conventions used:

- 9999 decimal number
- 9999h hexadecimal number
- 11111111b binary number
- „ABCabc“ string
- ‘A’ ASCII character

1. Communication description

Communication features are specified by GSD file NdiT08DB.GSD. Communication runs through defined memory space of Profibus DP MASTER device. Display message can be longer than defined memory space therefore transmission is made in two steps. Messages are first stored into buffer memory of the converter which has 1024 bytes. Each message contains position of location in final message for display. If all necessary data were sent, MASTER sends special command to send received data from converter into display control unit. Display answers with one byte. Converter then sends this answer to MASTER.

2. Communication module

Communication module has two parts – input 32 byte and output 1 byte.

Inputs:

Byte (offset)	0	1	2 – 31
Bit No	MSB Addresses	LSB Addresses	Data part (ASCII)
0. (LSB)	Adr	Adr	X
1.	Adr	Adr	X
2.	Adr	Adr	X
3.	Adr	Adr	X
4.	Adr	Adr	X
5.	Adr	Adr	X
6.	Adr	Adr	X
7. (MSB)	Adr	Adr	X

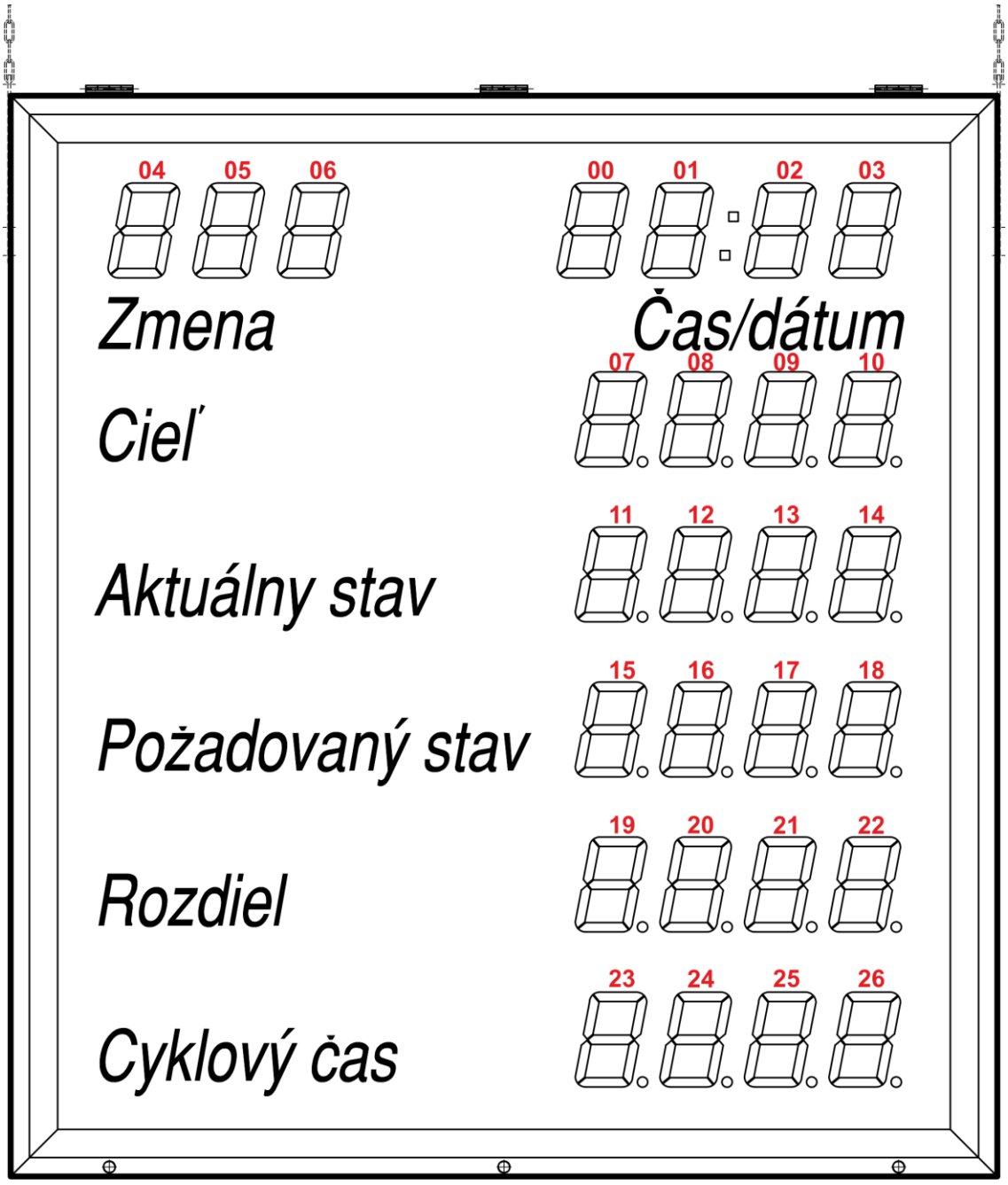
Position address can have values from 0 to 994.
Position 65535 (0xff, 0xff) is reserved for sending received data into the display control unit.

Output:

Byte (offset)	0
Bit No	Answer from display
0. (LSB)	X
1.	X
2.	X
3.	X
4.	X
5.	X
6.	X
7. (MSB)	X

If communication was ok, display control unit will return ACK (code 0x06).

3. Example



All messages are formated with command **$P** described in section **4.1.2**.
Displayed message can be sent as one full string, or by parts, which should be converted to blocks for transfer via communication interface Profibus DP.

$P00041245	Message field „Čas“ 12:45
$P0403123	Message field „Zmena“
$C1$P070512.34	Message field „Ciel“ (red color)
$C2$P110512.34	Message field „Aktuálny stav“ (green color)
$C3$P150512.34	Message field „Požadovaný stav“ (yellow)
$C1$P190545.34	Message field „Rozdiel“ (red)
$C3$P230578.25	Message field „Cyklový čas“ (yellow)