Communication Protocol Description

Protocol ELEN UNI-ND

Version 1.06

ELEN

1. Introduction

This manual provides information about controlling numerical LED displays through local data network based on communication interface RS485 or Ethernet. Displays are produced by ELEN, s.r.o. company.

2. LAN Description

Individual displays are interconnected with a control unit into a local computer network with a bus topology. Physical layer is based on RS485 or Ethernet interface.

Transmission via **RS485** is set as **asynchronous half-duplex** communication with the following parameters: **9 600, 8, N, 1**. The bus assignment is managed as a "MASTER – SLAVE" system. The MASTER is a control unit. All SLAVE devices have unique addresses from 1 to 127. Address 0 is a global address for all devices. Communication telegrams are secured with XOR checksum. Message repeat is controlled by MASTER.

For transmission via **Ethernet** has each display assigned its IP address and operates as **device server**.

3. Communication Telegrams

3.1 Sending Frame

Sending frame has a fixed structure with variable length of information field. Sending frame is used by MASTER only. Telegram length is limited to 128 bytes. If reply message is not used it is necessary to **wait at least 100 ms** before sending next message.

<STX><ADR><I><ETX><CSUMH><CSUML>

<STX>	0x02	• start of telegram flag
<ADR>		• address of SLAVE device, from 1 to 127 • 0 – global address for all SLAVE devices • D7 bit is always 1 • for communication via TCP the address is 127 (or is specified)
<I>		• information message content • variable length 0 ÷ 123 bytes
<ETX>	0x03	• end of frame flag
<CSUMH>	0x8...	• upper 4 bits of CSUM byte – bit D7 always 1
<CSUML>	0x8...	• lower 4 bits of CSUM byte – bit D7 always 1

CSUM is defined as logical (bitwise) exclusive sum of all bytes of message including STX.

3.2 Receiving Frame

SLAVE device, whose address has been used, can respond to sending frame in 3 ways:

<ACK>	• acknowledge of reception and execution of command
<NAK>	• refusing frame (BUSY, ...)
"nothing"	• reception of frame with global address • refusing frame (CSUM error) • reply is not required

4. Control Commands and Text Format

Information part of telegram contains commands which tell display what to perform with the specified text message. These commands are designated with "**$**" character.

4.1 Commands for Writing Text

Each sending frame can contain only one command.

4.1.1 Direct Writing of Text

Display will show required text message directly, while its previous content is erased. Characters which cannot be displayed are replaced with empty character.

ttttttt	• simple writing of ASCII text characters

4.1.2 Writing Text to Exact Position

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

<p10X><p1X>	• beginning position of ASCII text format range: position 0 to 64
<n10X><n1X>	• number of bytes to be displayed range: 0 – 64
text	• string of characters to be displayed

String characters can contain numbers, some 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. To display decimal point only use sequence: empty 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 writing text to exact position, see chapter 4.1.2.

4.2.1 Blinking of Message

$F1	• following characters will be blinking
$F0	• following characters will NOT be blinking

ASCII format, default 0.

Command **$F** is valid until: The next change of value with another **$F** command. Display is turned off. Changing settings to default parameters with command **$R**.

4.2.2 Changing Color of Text Message

This is valid for multi-color LED displays only. 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. Set parameters will be preserved even when 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 • 0 – brightness control by setting direct value of PWM without automatic brightness control • 1 – brightness control by setting the slope of regulation curve • 2 – brightness control by setting direct value of PWM without automatic brightness control, value is not written into EEPROM, valid until display reset • ASCII format, default 1
<brightness level>	• level value, while bit D7 is 1 • range 0 – 100%, default value is 80%

4.3.2 Setting Communication TIMEOUT

```
$T<time-out value>
```

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

After TIMEOUT period expiration, display will show dashes (- - - -). This can be used to control the validity of currently displayed data. TIMEOUT does not affect the part of display which shows information about time or date. Setting the TIMEOUT value to 0 results in permanent display of data.

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). Switching between summer/winter times is performed by display automatically.

```
$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 Controlling Checksum Verification

Enables or disables checksum verification during communication. Factory default setup for checksum is disabled. This verification is mainly used for RS485 network. For Ethernet TCP/IP communication it is not needed because verification of data is already performed by TCP link layer.

```
$D<checksum verification control>
```

<checksum verification control>	• 0 – checksum is not verified • 1 – checksum is verified • ASCII format, default 0

4.3.5 Controlling Reply of Display

Allows to set whether display should acknowledge received packets during communication.

```
$E<reply control>
```

<reply control>	• 0 – display does not reply • 1 – display replies • ASCII format, default 1

4.3.6 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 value is 15
<filling>	• what percentage of blinking period is display ON • range 0 – 100%, value of bit D7 is 1 • default value is 66%

5. Examples of Using ELEN UNI-ND Protocol for Numerical LED Displays

STX	• Start of telegram flag
ADR	• Address of display
INF	• Information content
ETX	• End of telegram flag

Example 1: 3-digit, one line display, direct writing of text "123", without CRC.

Sent string [Hex]:

```
02H,FFH,31H,32H,33H,03H
```

Sent string [Dec]:

```
2,255,49,50,51,3
```

Example 2: 3-digit, one line display, direct writing of text "123", with CRC.

Sent string [Hex]:

```
02H,FFH,31H,32H,33H,03H,8CH,8EH
```

Sent string [Dec]:

```
2,255,49,50,51,3,140,142
```

Notes:
In case of communication interface RS485 it is advised to use CRC checksum control verification. For Ethernet TCP/IP communication it is not needed to add CRC because verification of data is already performed by TCP protocol already has its own control of sent and received data packets.

Example 3: 4-digit, one line display, direct writing of text "1234", without CRC.

Sent string [Hex]:

```
02H,FFH,31H,32H,33H,34H,03H
```

Sent string [Dec]:

```
2,255,49,50,51,52,3
```

Example 4: 4-digit, one line display, direct writing of text "1234", with CRC.

Sent string [Hex]:

```
02H,FFH,31H,32H,33H,34H,03H,8FH,8AH
```

Sent string [Dec]:

```
2,255,49,50,51,52,3,143,138
```

Example 5: 4-digit, one line display, direct writing of text "1234" **with blinking**, without CRC.

Sent string [Hex]:

```
02H,FFH,24H,46H,31H,31H,32H,33H,34H,24H,46H,30H,03H
```

Sent string [Dec]:

```
2,255,36,70,49,49,50,51,52,36,70,48,3
```

Example 6: Real-time synchronization packet.

Sent string [Hex]:

```
02H,FFH,24H,53H,31H,31H,30H,32H,31H,30H,31H,32H,35H,39H,35H,33H,03H
```

Will set date to **10.02.2018** and time to **12:59:53**

Example 7: Direct writing of text into 6 rows at once in the following format.

Sent string [Hex]:

```
02H,FFH,30H,30H,30H,31H,31H,31H,31H,32H,32H,32H,32H,33H,33H,33H,
33H,34H,34H,34H,34H,35H,35H,35H,35H,03H
```

Example 8: Writing text "1234" to position number 11 with length 4 characters.

Sent string [Hex]:

```
02H,FFH,24H,50H,31H,31H,30H,34H,31H,32H,33H,34H,03H
```

Messages were sent with address 127.