Terminal by Koval

created by a developer for developers

ver. 1.01

Summary

Target Applications

This terminal is a convenient and free tool for an engineer, which is designed to work with a COM-port (RS-232 interface, microcontroller UART interface), and if there are appropriate converters, also RS-485, RS-422 interfaces. The terminal will be useful when operating a variety of equipment that supports exchange via RS-232, RS-485, RS-422 interfaces. However, it is more "sharpened" for the needs of development engineers who are engaged in the development of electronic devices, their testing and software creation, including for microcontrollers.

The terminal can operate in one of two main modes:

1 - ***classic terminal mode***. Designed for data exchange with devices connected to the computer's RS-232 interface;

2 - ***monitoring mode***. Allows, ***when using a special cable*** and having two COM ports on the computer, to view the exchange of information between two devices operating via the RS-232 interface, for example, between the control controller and a GSM-modem.

Features

– without installation, only single and small .exe - file

– visual - spatial separation of the received and sent data (or data received by different ports - in the monitoring mode), which makes it easier for the operator to perceive them, since the data is presented in the form of a dialogue, as in a messenger;

– presentation of data in text (TEXT), decimal (DEC), hexadecimal (HEX) and binary (BIN) formats in tabular form (Tables) and free text (Memo). In Memo mode, you can choose the font, size and color of the text;

– a table of 1000 macros and comments to them, each of which can be instantly sent to the port by double-clicking on its number;

– the terminal is convenient for working with both text (character) and binary exchange protocols;

– various modes of splitting the received data into lines: upon detecting a fixed terminator ($0D, $0A, $0D$0A, $0D or $0A), by timeout after the end of the last byte reception, by receiving a fixed number of bytes;

– automatic addition to the sent data of various checksums (CRC) and terminators: $0D, $0A or $0D$0A (optional);

– detailing of the last received data block (DLRB) - a tool that allows you to split the received data into separate bytes in real time, defining for each byte its own sequence number in the message and setting a separate color for highlighting each byte or group of bytes, which makes it easier to visualize the data. This is convenient when working with positional protocols, when the position of each byte in a message is always fixed and does not change over time;

– the ability to calculate in real time on the basis of the received data one-, two- and four-byte values and, taking them as arguments, calculate on their basis complex functions given by a text formula;

– tools to implement support for the Modbus protocol, both in Master and Slave modes. 1, 2, 3, 4, 5, 6 and 16 protocol functions are supported;

– the mode of saving all received / sent to the log-file;

– the "Programs" tool, which allows you to implement the execution of the simplest programs based on sending specified macros with certain pauses between them with the possibility of looping the execution of individual blocks;

– a "String" tool that allows you to analyze a string in detail by copying it through the clipboard and breaking it down by bytes, determine its length, highlight individual bytes with a certain color and attach your own comment to each, calculate various checksums of a given string, translate a given number between different number systems (DEC <> HEX <> BIN), having decomposed it into separate bytes, perform a bitwise shift of these bytes left and right;

– quick display of reference information: tables of ASCII characters, descriptions of COM port pins (connector pinouts), cable wiring diagrams for implementing the monitoring mode of COM ports.

Installation

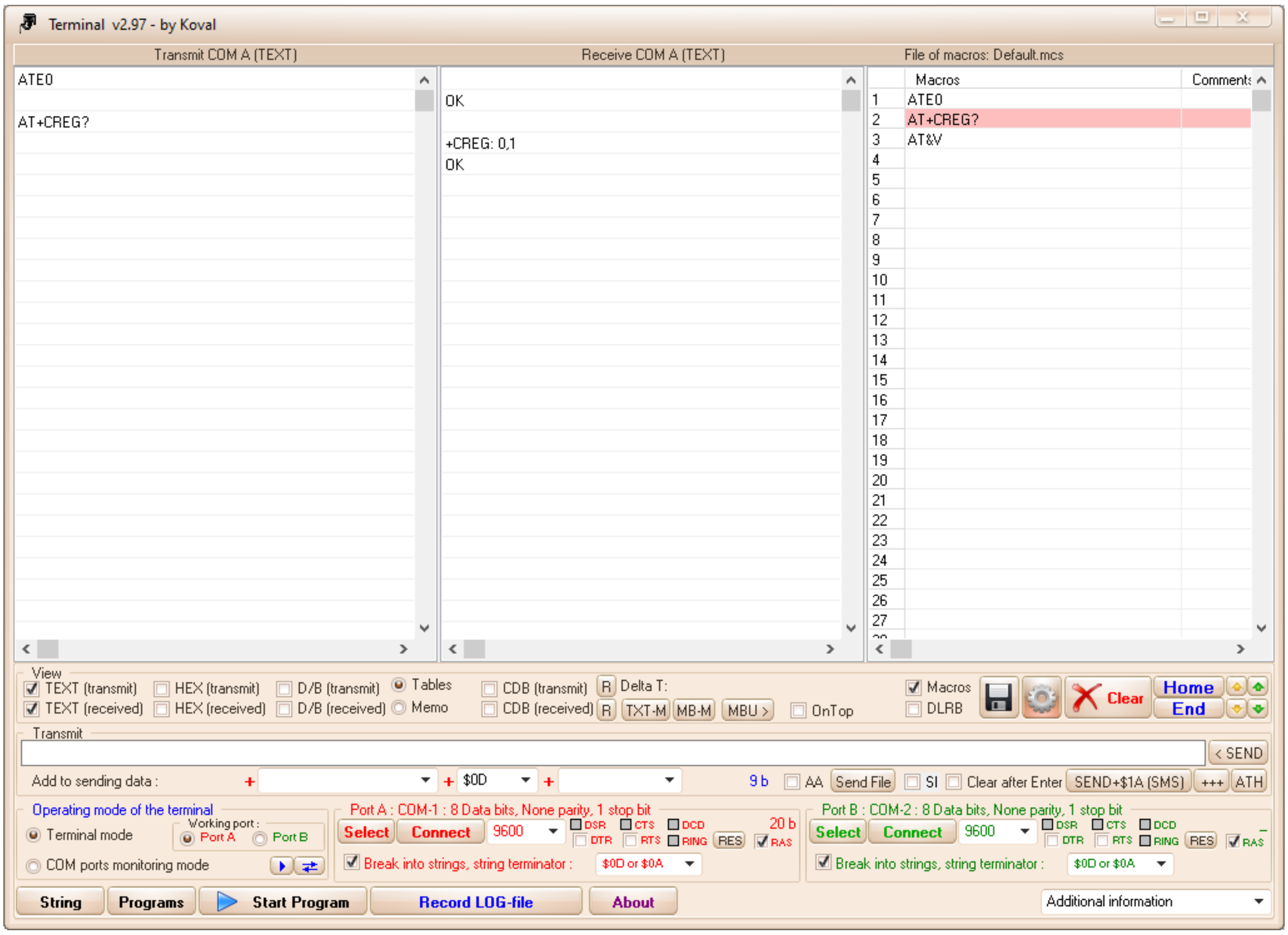
Unzip and copy the exe-file to a ***separate folder*** and run it.

Please note that the program, when launched, creates at least nine folders in the current directory for its work.

In the future, in the event of an update, it is possible to simply replace the exe-file with a new one. In this case, it is better to first delete the tbk.ini file from the "INI" folder.

Brief description of the interface

The terminal interface is shown in the following figure:



Transmit: ATE0, AT+CREG?

Receive: OK, +CREG: 0,1, OK

	Macros	Comments
1	ATE0	
2	AT+CREG?	
3	AT&V	

All data sent and received by the terminal are entered into tables or Memo located in the upper part of the working area of the window, and all settings are made in the fields and components in the lower area of the terminal.

Quick start

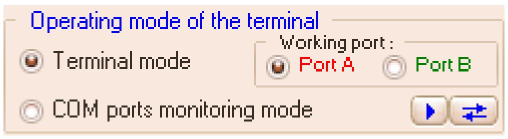
In order to quickly start working with the terminal in the mode of polling any equipment via the COM port, you need to perform just a few steps:

1. Start the program.
2. In the "View" field: to work with text protocols (work with a GSM-modem, GPS-module, etc.), press the button TXT-M (Text-mode), to work with binary protocols (for example, Modbus protocol) press the button MB-M (Modbus-mode).
3. In the "Terminal mode" field, select "Terminal mode", "Port A".
4. In the "Port A" field, click the Select and select the required COM port from the list of available ports. If the port is already selected (its number is indicated in the field header), i.e. if you have already worked with it before, then you just need to press the button Connect.

The terminal is now ready for use. In order to send something to the port, type the data in the text line and press the Enter button on the keyboard or the mouse button < SEND, the typed text + byte-terminator $0D will be sent to the port. Terminator byte is selected in the field + $0D. To send non-printable characters, the standard construction $XX is used, where XX are two hex characters representing the byte to be sent in hexadecimal notation (example: $0D). In order for the input line to be automatically cleared after sending the line to accept new text, you must enable the checkbox ☐ Clear after Enter.

All sent data are entered in the left column of the table, all data received by the port - in the right. Timestamp display is configured here: → Tab "View Text" → field "Time".

"Operating mode of the terminal" field


Operating mode of the terminal

[✓ Terminal mode]

Working port : [✓ Port A] Port B

COM ports monitoring mode

This is the main field in which one of the two main operating modes of the terminal is selected:

– *Terminal mode* - classic terminal mode, in which both sending and receiving data through the selected COM port of the computer is performed. The terminal can SIMULTANEOUSLY open two COM ports on the computer, they are conventionally called **Port A** and **Port B**.

In the "***Working port***" field, the user can choose which of the open ports to work with. Thus, you can physically connect two different devices to the computer (with different baud rates and other port settings) and, by switching the working port, work with each of them in turn.

Warning! Do not forget to control which port is currently working, otherwise you may get the impression that for some reason the data is not received or the terminal is not working correctly!

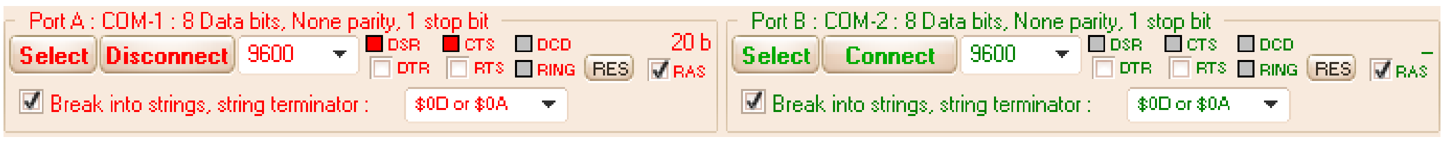
– *COM ports monitoring mode* - a mode that allows, ***if there is a special cable*** and two COM-ports on the computer, to view the exchange of information between two devices operating via the RS-232 interface. The wiring diagram of the cable can be viewed in the terminal itself, in the "Additional information" field. The cable is connected to break the connection of two devices. In this case, the terminal works only in the RECEIVE mode, displaying the data received by ***port A*** on the left, and the data received by ***port B*** on the right.

▶, ■ - opens / closes both ports ***at the same time***.

⇄ - swaps the numbers of ports A and B. That is, if COM-1 was opened under port A, and COM-2 was opened under port B, then after pressing this button, COM-2 will become open under port A, and under port B it will become open COM-1. This function makes it possible to quickly replace open ports in places without reconnecting the cable, so that, for example, requests (as usual) are displayed on the left, and responses on the right (if the cable was not initially connected very well).

Warning! This function is only available when both ports ***are open***.

Fields "Port A" "Port B"


Port A : COM-1 : 8 Data bits, None parity, 1 stop bit
Select Disconnect 9600 · DSR · CTS · DCD · DTR · RTS · RING · RES · ✓ RAS · 20 b
✓ Break into strings, string terminator : $0D or $0A

Port B : COM-2 : 8 Data bits, None parity, 1 stop bit
Select Connect 9600 · DSR · CTS · DCD · DTR · RTS · RING · RES · ✓ RAS
✓ Break into strings, string terminator : $0D or $0A

These are the fields for selecting and configuring COM ports that the terminal works with.

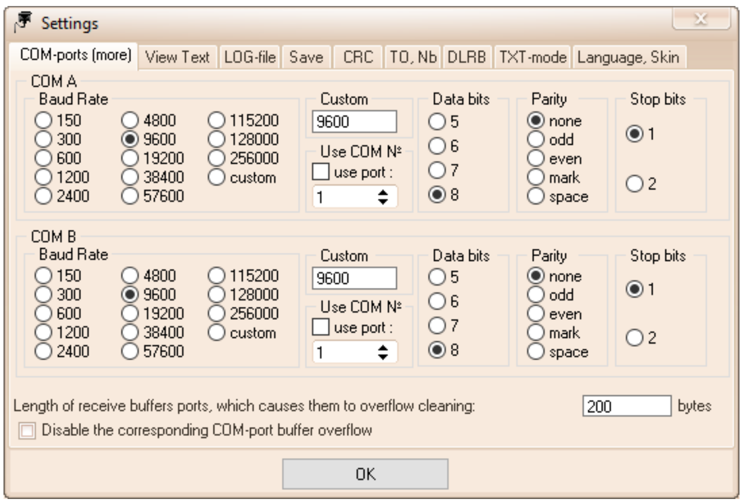
Select - button for selecting the required COM-port from the list of available ones.

Connect, Disconnect - COM-port open/close button. If, when starting the program, the required port has already been selected (after the previous session), then just click on this button ("Connect"). While the terminal is running, at any time, you can free a busy port for use by another application by pressing this button again ("Disconnect").

DSR CTS DCD DTR RTS RI - field for displaying DSR, CTS, DCD, RING inputs and controlling DTR and RTS outputs.

20 b RES ✓ RAS - counter of the number of bytes received on this COM-port, its reset button, a checkbox for setting the auto-reset mode of this counter after each act of data transmission through this port.

9600 - a drop-down list defining the baud rate of the COM-port. The "custom" option, like other COM-port settings, become available after calling the context menu Additional configuration of COM ports right click on the "Port A" field. This opens the "COM-ports (more)" tab in the "Settings" window:


Settings
COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

COM A
Baud Rate: 150, 300, 600, 1200, 2400, 4800, [✓ 9600], 19200, 38400, 57600, 115200, 128000, 256000, custom
Custom: 9600
Use COM №: use port : 1
Data bits: 5, 6, 7, [✓ 8]
Parity: [✓ none], odd, even, mark, space
Stop bits: [✓ 1], 2

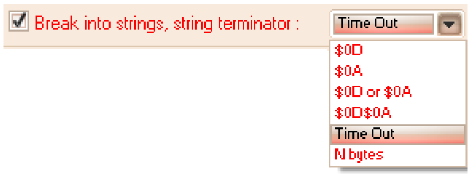
COM B
Baud Rate: 150, 300, 600, 1200, 2400, 4800, [✓ 9600], 19200, 38400, 57600, 115200, 128000, 256000, custom
Custom: 9600
Use COM №: use port : 1
Data bits: 5, 6, 7, [✓ 8]
Parity: [✓ none], odd, even, mark, space
Stop bits: [✓ 1], 2

Length of receive buffers ports, which causes them to overflow cleaning: 200 bytes
Disable the corresponding COM-port buffer overflow
OK

This tab is intended for a more complete configuration of COM-ports, namely, select: non-standard speed, number of data bits, parity, number of stop bits.

The "***Use COM №***" field is intended for forced setting of the COM port number. In some rare cases, the terminal may not show the required port in the list of available ones, although in fact there is a port in the system. In this case, you can select this check box and select any port number in the range from 1 to 255.

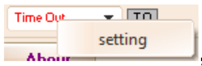
All received data are entered into special receive buffers, the contents of which are displayed in real time in tables and Memo. However, if the condition of dividing the received data into lines is not met, or the splitting is disabled, then the lines can take on very large sizes, which is very inconvenient for displaying on the screen. Therefore, the terminal has an adjustable limit on the number of bytes received, after exceeding which the buffer is considered full. When such an overflow occurs, the contents of the buffer are forced into the current row in the table with the "BUFFER OVERLOAD!>" Marker, after which the receive buffer is cleared to receive the next data. The terminal may regard such a buffer overflow as an abnormal situation (port "spam") and forcibly close this COM-port. These parameters apply to both ***Port A*** and ***Port B*** at the same time and are configured at the bottom of the tab.



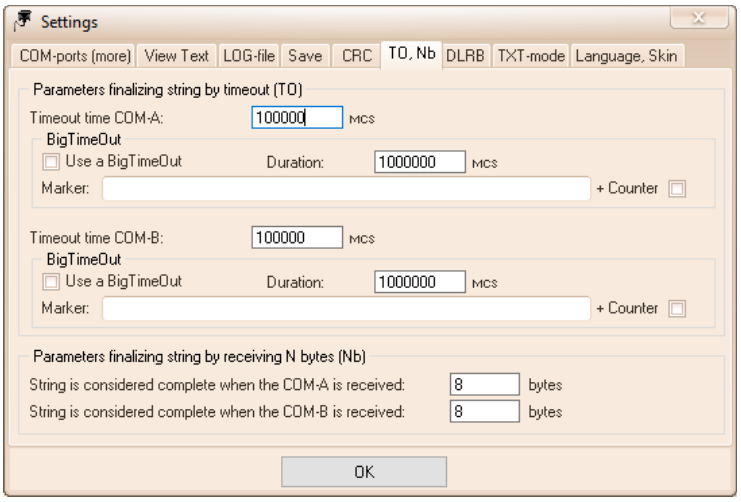
- setting this flag allows you to split the incoming data stream into separate lines according to a certain criterion: receiving a specified terminator byte - $0D, $0A, $0D or $0A, $0D$0A (2 bytes in sequence), expiration of a silence timeout (time interval, during which there was no data reception) or on the reception of a given number of bytes.

Warning! If this check box is not checked, then all bytes of the received message are entered into each new line as the data is received (at each event of receiving) by the component of the terminal COM-port. At the same time, the division into such block lines is almost random and depends on the version of the operating system, its load and the intensity of the received digital stream.

The mode of splitting the received data by timeout is convenient to use when working with binary protocols, in particular the Modbus protocol. Each message in such protocols often ends with a checksum (CRC), which is constantly changing. Therefore, the mode of splitting by a given byte-terminator does not work in this case, and the only way to split such a stream into lines is by the silence timeout on the bus. In particular, when devices operate using the Modbus protocol and a master polls one or more slave devices cyclically, it is possible to split packets (request + response) into separate lines, and at not very high exchange rates, even split the request and response among themselves.

Timeout parameters and reception of a given number of bytes are configured by calling the context menu 

, which opens the tab "TO, Nb" in window "Settings":


Settings

COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

Parameters finalizing string by timeout (TO)

Timeout time COM-A: 100000 MCS

BigTimeOut

☐ Use a BigTimeOut   Duration: 1000000 MCS

Marker: + Counter ☐

Timeout time COM-B: 100000 MCS

BigTimeOut

☐ Use a BigTimeOut   Duration: 1000000 MCS

Marker: + Counter ☐

Parameters finalizing string by receiving N bytes (Nb)

String is considered complete when the COM-A is received: 8 bytes

String is considered complete when the COM-B is received: 8 bytes

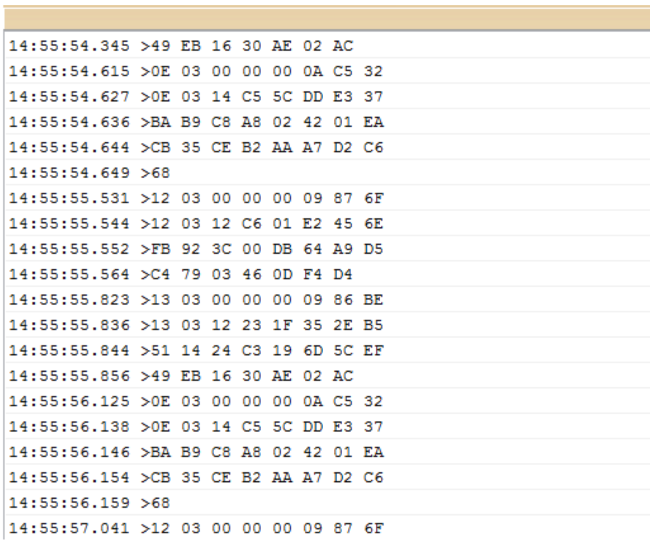
OK

On this tab, the timeout value is set, which is entered in microseconds, but the real resolution is lower and depends on the performance of the computer and the load of the operating system with other tasks.

If during the specified timeout no new bytes have been received, then the line is considered finalized and the next bytes will be received on a new line. To count the time intervals, the program uses the ***QueryPerformanceCounter*** functions based on the resources of the ***High Precision Event Timer*** system timer, which is hardware implemented on the computer, and the time counting procedure itself is implemented to improve the measurement accuracy in a separate thread. Nevertheless, the timing accuracy is much greater than 1 microsecond and the timeout value should be selected experimentally.

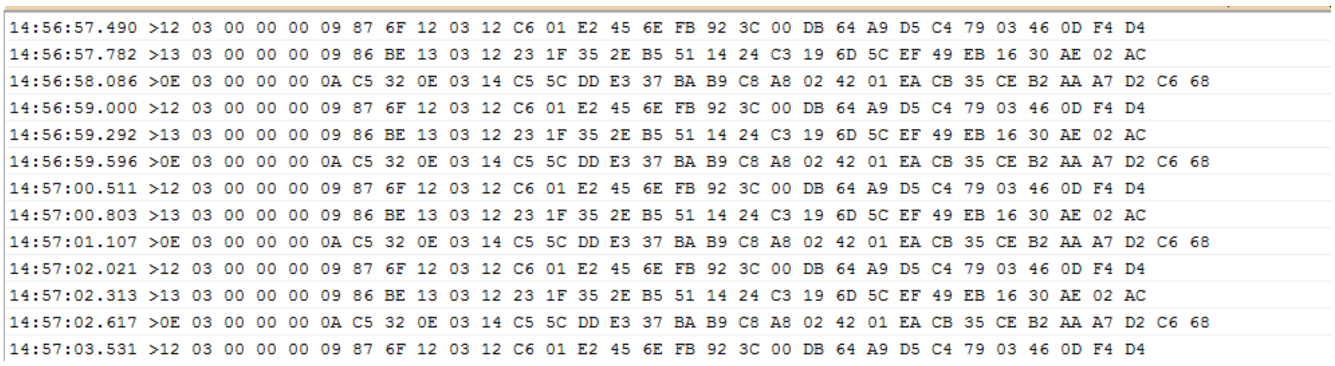
It is also possible to enable ***BigTimeOut***, after which the specified marker (plus, optionally, an incremental counter) will be inserted into the received data. This can significantly improve the perception of the data by the operator, for example, when a cyclic poll by one modbus master of several slaves is monitored, allowing you to mark the beginning of each new cycle.

This figure shows the results of the terminal receiving data from the RS-485 interface bus, through which the modbus master polls three slaves with network addresses 18, 19 and 14 at a speed of 9600 with a cycle period of 1500 ms ***with the line splitting mode disabled***:



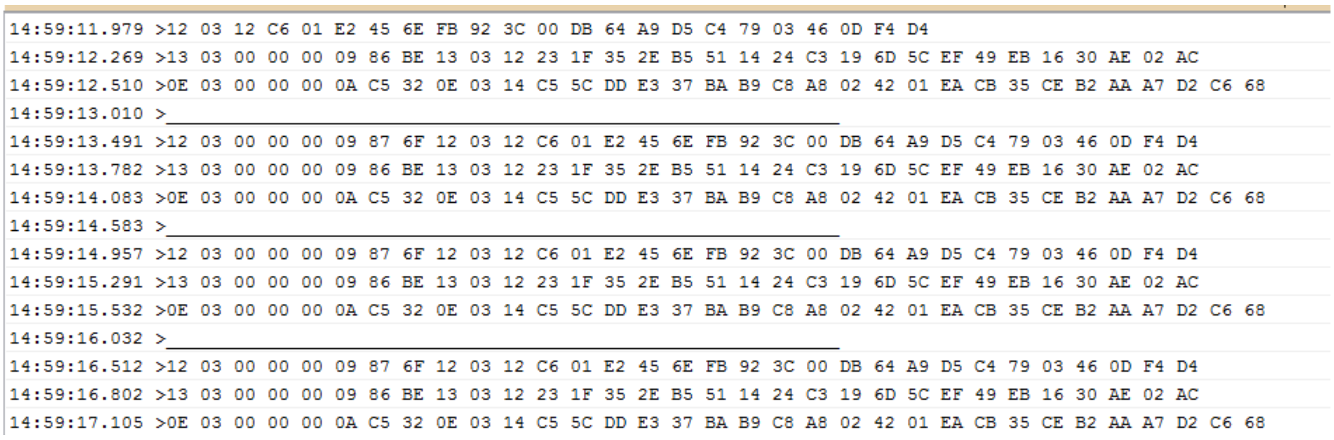
```
14:55:54.345 >49 EB 16 30 AE 02 AC
14:55:54.615 >0E 03 00 00 00 0A C5 32
14:55:54.627 >0E 03 14 C5 5C DD E3 37
14:55:54.636 >BA B9 C8 A8 02 42 01 EA
14:55:54.644 >CB 35 CE B2 AA A7 D2 C6
14:55:54.649 >68
14:55:55.531 >12 03 00 00 00 09 87 6F
14:55:55.544 >12 03 12 C6 01 E2 45 6E
14:55:55.552 >FB 92 3C 00 DB 64 A9 D5
14:55:55.564 >C4 79 03 46 0D F4 D4
14:55:55.823 >13 03 00 00 00 09 86 BE
14:55:55.836 >13 03 12 23 1F 35 2E B5
14:55:55.844 >51 14 24 C3 19 6D 5C EF
14:55:55.856 >49 EB 16 30 AE 02 AC
14:55:56.125 >0E 03 00 00 00 0A C5 32
14:55:56.138 >0E 03 14 C5 5C DD E3 37
14:55:56.146 >BA B9 C8 A8 02 42 01 EA
14:55:56.154 >CB 35 CE B2 AA A7 D2 C6
14:55:56.159 >68
14:55:57.041 >12 03 00 00 00 09 87 6F
```

When the ***mode of splitting the received data by timeout (100 ms) is enabled***, it is already possible to separate separate "request-response" blocks addressed to different devices:



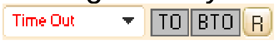
```
14:56:57.490 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:56:57.782 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:56:58.086 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:56:59.000 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:56:59.292 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:56:59.596 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:57:00.511 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:57:00.803 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:57:01.107 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:57:02.021 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:57:02.313 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:57:02.617 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:57:03.531 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
```

When the ***BigTimeOut marker is enabled (500ms)***, it becomes easier to see the beginning and end of each polling cycle of all slaves:



```
14:59:11.979 >12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:59:12.269 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:59:12.510 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:59:13.010 >__________________________________________________________________
14:59:13.491 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:59:13.782 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:59:14.083 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:59:14.583 >__________________________________________________________________
14:59:14.957 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:59:15.291 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:59:15.532 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
14:59:16.032 >__________________________________________________________________
14:59:16.512 >12 03 00 00 00 09 87 6F 12 03 12 C6 01 E2 45 6E FB 92 3C 00 DB 64 A9 D5 C4 79 03 46 0D F4 D4
14:59:16.802 >13 03 00 00 00 09 86 BE 13 03 12 23 1F 35 2E B5 51 14 24 C3 19 6D 5C EF 49 EB 16 30 AE 02 AC
14:59:17.105 >0E 03 00 00 00 0A C5 32 0E 03 14 C5 5C DD E3 37 BA B9 C8 A8 02 42 01 EA CB 35 CE B2 AA A7 D2 C6 68
```

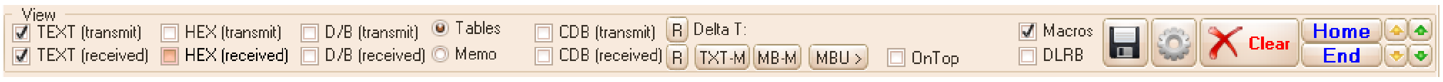
With a further decrease in the line splitting timeout, you can try to achieve splitting the master's request and the slave's response into separate lines, however, due to the short time pause between the end of the request and the beginning of the response and the peculiarities of the COM-port operation under Windows OS, this is not always possible do it correctly.

When finalizing lines by timeout, the terminal displays indicators of the timeout (and long timeout) countdown: 

, which are highlighted in the corresponding (red or green, depending on the port) color during the timeout countdown. The rest of the time (in a state of expectation or rest), they are colored gray. There is also a button to reset both timeouts ("R") in case of timing error. The *BigTimeOut* function works correctly with text protocols as well.

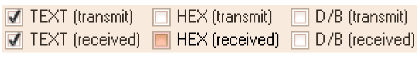
Warning! For more correct operation of the terminal in the timeout counting mode, it is recommended not to run other applications on the computer at this time.

"View" field



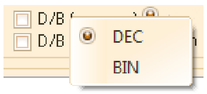
View
[✓] TEXT (transmit) [ ] HEX (transmit) [ ] D/B (transmit) (●) Tables [ ] CDB (transmit) R Delta T: [✓] Macros
[✓] TEXT (received) [ ] HEX (received) [ ] D/B (received) ( ) Memo [ ] CDB (received) R TXT-M MB-M MBU > [ ] OnTop [ ] DLRB

This field sets how the transmitted and received data is displayed, and also provides quick access to some important settings and functions.

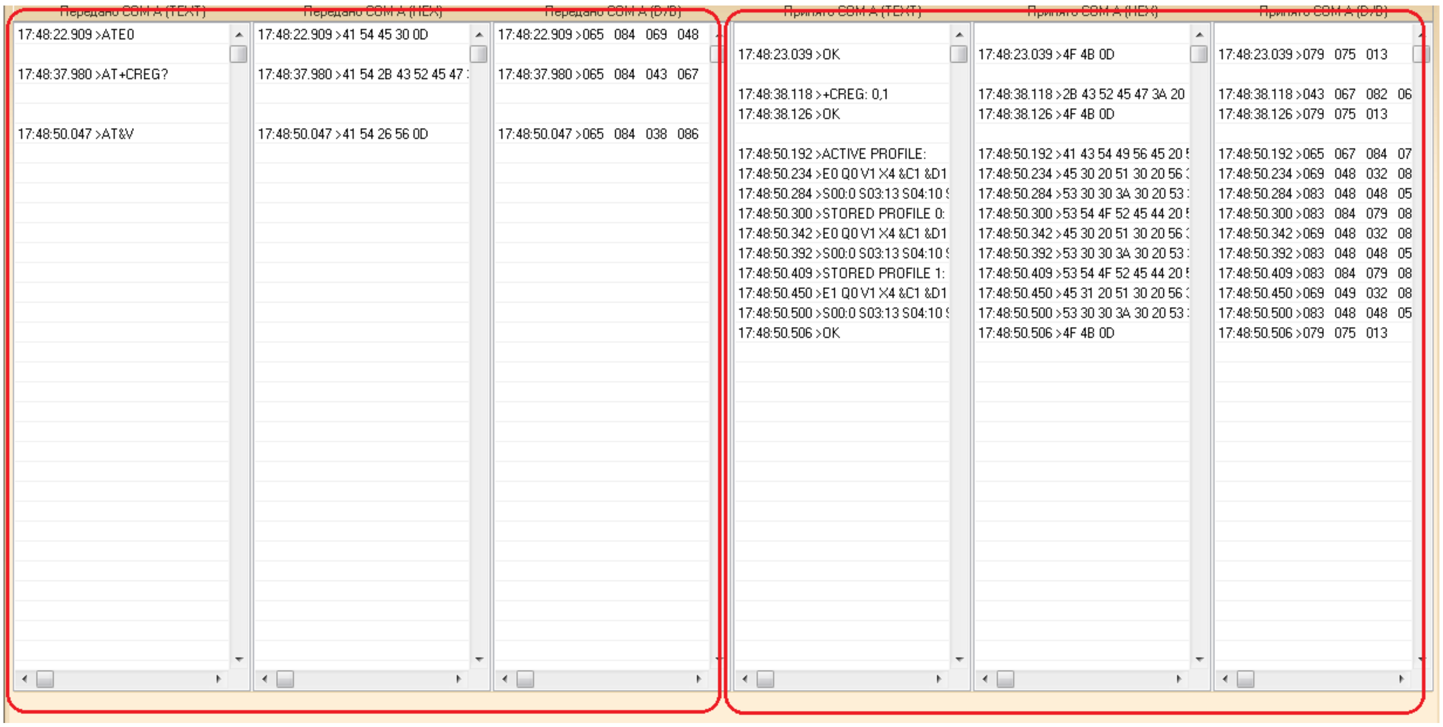


[✓] TEXT (transmit) [ ] HEX (transmit) [ ] D/B (transmit)
[✓] TEXT (received) [ ] HEX (received) [ ] D/B (received)

- checkboxes for selecting columns of number systems for displaying data. An additional choice of decimal or binary number system is performed from the context menu of the "D / B" field:



When all checkboxes are enabled, the working area looks like this:



Передано COM A (TEXT)
```
17:48:22.909 >ATE0

17:48:37.980 >AT+CREG?


17:48:50.047 >AT&V
```

Передано COM A (HEX)
```
17:48:22.909 >41 54 45 30 0D

17:48:37.980 >41 54 2B 43 52 45 47 :


17:48:50.047 >41 54 26 56 0D
```

Передано COM A (D/B)
```
17:48:22.909 >065  084  069  048

17:48:37.980 >065  084  043  067


17:48:50.047 >065  084  038  086
```

Принято COM A (TEXT)
```

17:48:23.039 >OK

17:48:38.118 >+CREG: 0,1
17:48:38.126 >OK

17:48:50.192 >ACTIVE PROFILE:
17:48:50.234 >E0 Q0 V1 X4 &C1 &D1
17:48:50.284 >S00:0 S03:13 S04:10 S
17:48:50.300 >STORED PROFILE 0:
17:48:50.342 >E0 Q0 V1 X4 &C1 &D1
17:48:50.392 >S00:0 S03:13 S04:10 S
17:48:50.409 >STORED PROFILE 1:
17:48:50.450 >E1 Q0 V1 X4 &C1 &D1
17:48:50.500 >S00:0 S03:13 S04:10 S
17:48:50.506 >OK
```

Принято COM A (HEX)
```

17:48:23.039 >4F 4B 0D

17:48:38.118 >2B 43 52 45 47 3A 20
17:48:38.126 >4F 4B 0D

17:48:50.192 >41 43 54 49 56 45 20 5
17:48:50.234 >45 30 20 51 30 20 56 :
17:48:50.284 >53 30 30 3A 30 20 53 :
17:48:50.300 >53 54 4F 52 45 44 20 5
17:48:50.342 >45 30 20 51 30 20 56 :
17:48:50.392 >53 30 30 3A 30 20 53 :
17:48:50.409 >53 54 4F 52 45 44 20 5
17:48:50.450 >45 31 20 51 30 20 56 :
17:48:50.500 >53 30 30 3A 30 20 53 :
17:48:50.506 >4F 4B 0D
```

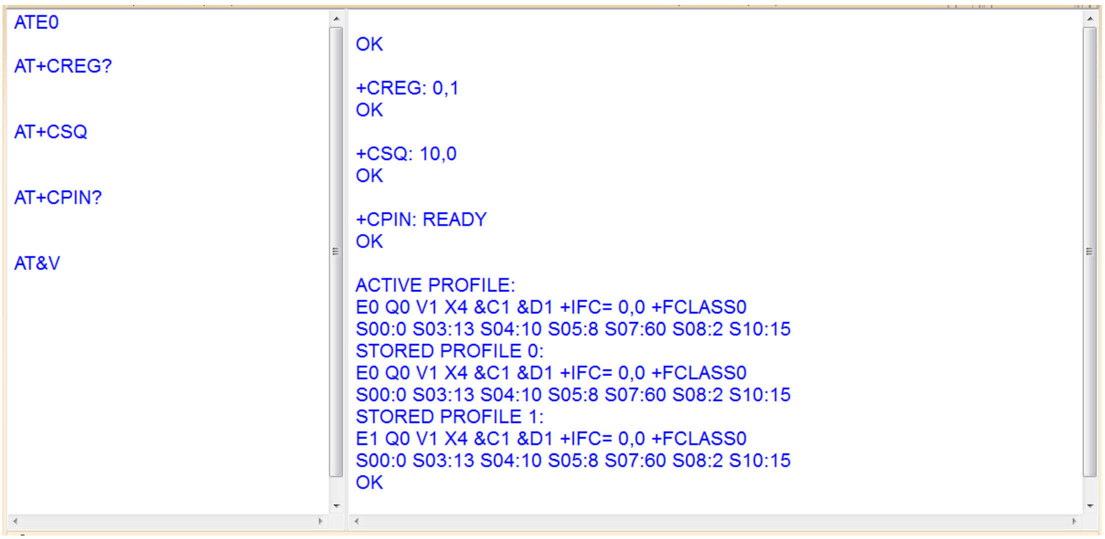
Принято COM A (D/B)
```

17:48:23.039 >079  075  013

17:48:38.118 >043  067  082  06
17:48:38.126 >079  075  013

17:48:50.192 >065  067  084  07
17:48:50.234 >069  048  032  08
17:48:50.284 >083  048  048  05
17:48:50.300 >083  084  079  08
17:48:50.342 >069  048  032  08
17:48:50.392 >083  048  048  05
17:48:50.409 >083  084  079  08
17:48:50.450 >069  049  032  08
17:48:50.500 >083  048  048  05
17:48:50.506 >079  075  013
```

Data from different calculation systems are always entered in separate columns, but *always data **transmitted** in terminal mode or data received in monitoring mode via port A is entered into the **left side** of the terminal, and data **received** in terminal mode or data received in monitoring mode via port B are entered on the **right side** of the terminal.*

(●) Tables
( ) Memo

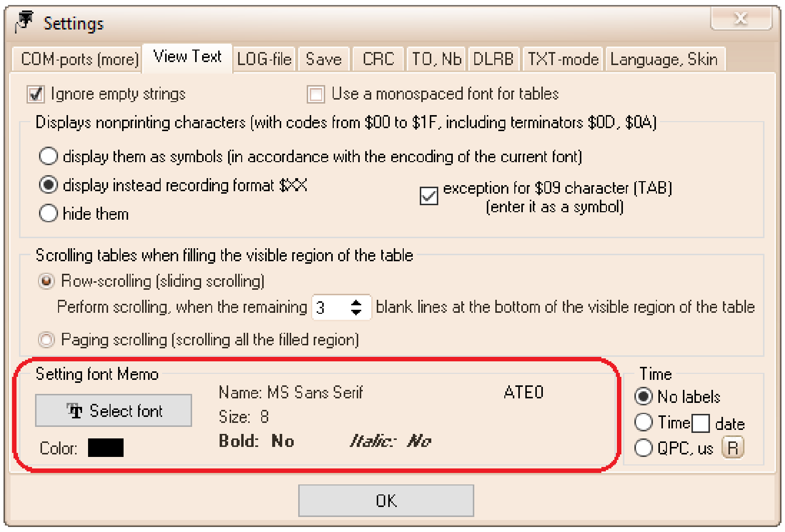
- selection of the type of containers where the data is entered. By default, data is entered into tables, and the columns of all calculation systems are filled in, regardless of whether their display is enabled or not. Tables have well-defined lines to make the data easier to read. This is the main mode of operation.

Memo - text containers that do not have explicitly separated cells. Initially, the terminal was developed using only tables, but later Memo components were added, since for some users they are more traditional and familiar. A distinctive feature of Memo components is the ability to customize the font, color, size of the text displayed in them, this can be useful when the operator is far away from the monitor:



```
ATE0                OK

AT+CREG?            +CREG: 0,1
                    OK

AT+CSQ              +CSQ: 10,0
                    OK

AT+CPIN?            +CPIN: READY
                    OK

AT&V                ACTIVE PROFILE:
                    E0 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
                    S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
                    STORED PROFILE 0:
                    E0 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
                    S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
                    STORED PROFILE 1:
                    E1 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
                    S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
                    OK
```

Setting the type of text in Memo is available as follows: button , further:



Settings

COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

[✓] Ignore empty strings   [ ] Use a monospaced font for tables

Displays nonprinting characters (with codes from $00 to $1F, including terminators $0D, $0A)
- ○ display them as symbols (in accordance with the encoding of the current font)
- ● display instead recording format $XX   [✓] exception for $09 character (TAB) (enter it as a symbol)
- ○ hide them

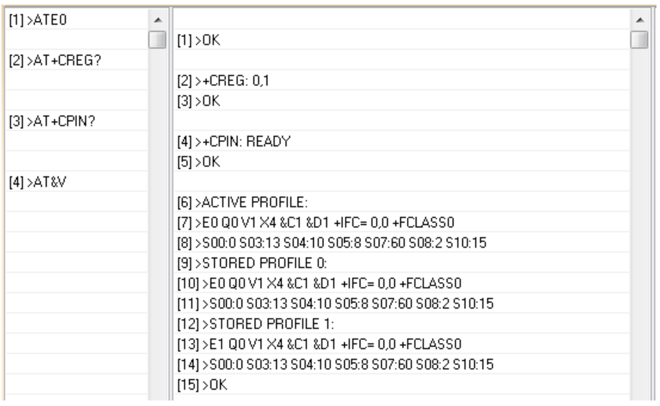
Scrolling tables when filling the visible region of the table
- ● Row-scrolling (sliding scrolling)
  Perform scrolling, when the remaining 3 blank lines at the bottom of the visible region of the table
- ○ Paging scrolling (scrolling all the filled region)

Setting font Memo
Select font   Name: MS Sans Serif   ATE0
Size: 8
Color: ■   **Bold: No**   ***Italic: No***

Time
- ● No labels
- ○ Time [ ] date
- ○ QPC, us R

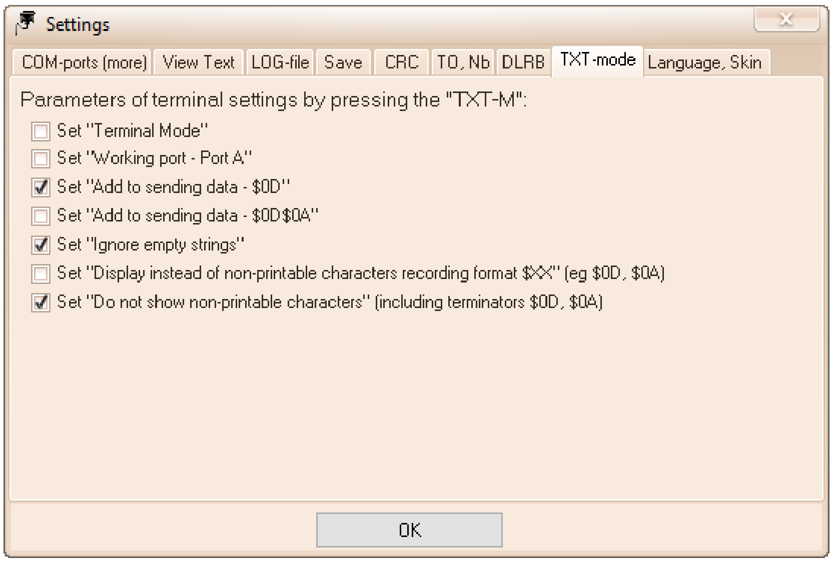
OK

CDB (transmit) R
CDB (received) R - counters of blocks of transmitted and received data, as well as buttons for their reset. The counting range is cyclically from 0 to 65535. The counter value is entered at the very beginning of the line:



```
[1]>ATE0              [1]>OK

[2]>AT+CREG?          [2]>+CREG: 0,1
                      [3]>OK
[3]>AT+CPIN?
                      [4]>+CPIN: READY
                      [5]>OK
[4]>AT&V
                      [6]>ACTIVE PROFILE:
                      [7]>E0 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
                      [8]>S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
                      [9]>STORED PROFILE 0:
                      [10]>E0 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
                      [11]>S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
                      [12]>STORED PROFILE 1:
                      [13]>E1 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
                      [14]>S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
                      [15]>OK
```

TXT-M - Text-Mode - button for quick configuration of the terminal for working with text protocols. The settings that are performed with the terminal when this button is pressed can be further configured on the tab "TXT-mode" : , further:


Settings

COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

Parameters of terminal settings by pressing the "TXT-M":

- ☐ Set "Terminal Mode"
- ☐ Set "Working port - Port A"
- [✓ Set "Add to sending data - $0D"]
- ☐ Set "Add to sending data - $0D$0A"
- [✓ Set "Ignore empty strings"]
- ☐ Set "Display instead of non-printable characters recording format $XX" (eg $0D, $0A)
- [✓ Set "Do not show non-printable characters" (including terminators $0D, $0A)]

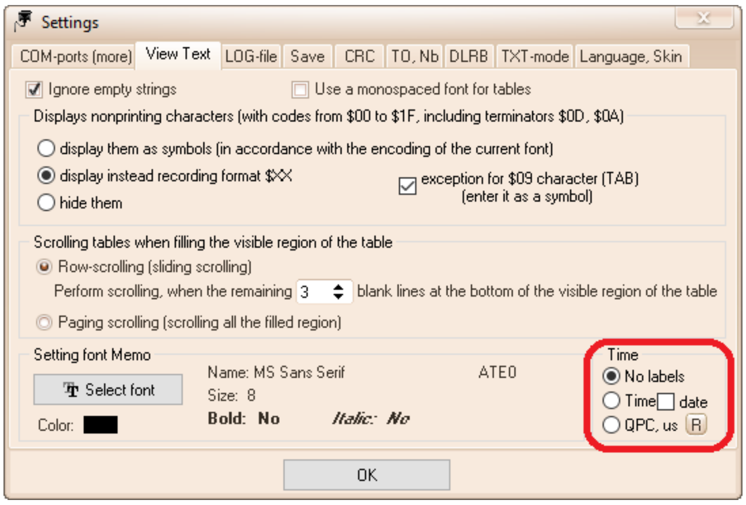
OK

MB-M - Modbus-MODE - button for quick configuration of the terminal for working with binary protocols, in particular, with the Modbus protocol.

MBU > - button for calling Modbus - utilities that allow the terminal to work in Modbus – Master mode or Modbus – Slave mode. For more details, see the "*Modbus – utilities*" section.

☐ OnTop - a checkbox that enables the terminal window to remain on top of other Windows windows in non-blocking mode.

Delta T:00:00:00.004 - ***time difference indicator*** of the last two received or two selected blocks. To use it, you must first enable adding timestamps to the data displayed by the terminal, the button , further:


Settings

COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

[✓ Ignore empty strings] ☐ Use a monospaced font for tables

Displays nonprinting characters (with codes from $00 to $1F, including terminators $0D, $0A)
- ○ display them as symbols (in accordance with the encoding of the current font)
- [✓ display instead recording format $XX] [✓ exception for $09 character (TAB) (enter it as a symbol)]
- ○ hide them

Scrolling tables when filling the visible region of the table
- [✓ Row-scrolling (sliding scrolling)]
  Perform scrolling, when the remaining 3 blank lines at the bottom of the visible region of the table
- ○ Paging scrolling (scrolling all the filled region)

Setting font Memo
Select font
Name: MS Sans Serif
Size: 8
Color:
Bold: No *Italic: No*
ATE0

Time
- [✓ No labels]
- ○ Time ☐ date
- ○ QPC, us R

OK

The terminal can add two types of timestamps:

"Time" - stamps of the absolute (system) time set in the operating system.

"QPC" - relative time stamps using the *QueryPerformanceCounter* functions based on the resources of the system *High Precision Event Timer*, which is implemented on the computer hardware and allows you to distinguish between microsecond time intervals.

Warning! The ***time difference indicator*** works ***only*** in conjunction with the ***"Time"*** mode!

To reset the time difference indicator, select an empty cell (no text) in the table. This will clear the indicator. Then it is necessary to sequentially select two necessary cells with data, which have the correct time stamps created in the ***"Time"*** mode. The indicator will display the difference in the time of their fixation.

☐ Macros - checkbox to enable displaying of a column (table) of macros. More details - in the section "Macros".

☐ DLRB - checkbox for enabling the display of the DLRB table (Detail Last Received Block) - detailing the last received block. More details - in the "DLRB" section.

- button for calling the "Settings" dialog box with an open tab for saving data entered in tables and Memo.

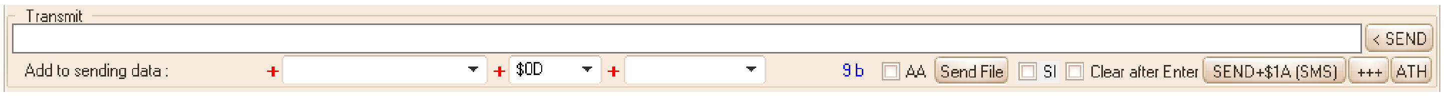
- button for launching the "Settings" dialog box.

Clear - button to clear tables and Memo. ***Warning!*** Cleaning is done completely and without confirmation warning.

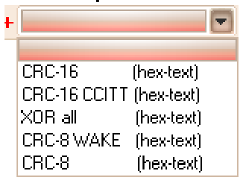
Home
End - buttons for scrolling tables and Memo to the beginning (Home) and to the end of the *area filled with data* (End).

- buttons for scrolling tables and Memo: 10 lines each (red, left) and line by line (green, right).

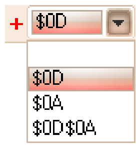
"Transmit" field



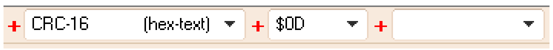
In the "Transmit" field, the data to be transferred is entered, the final data block is selected, which will be added at the end of this data, and there are also buttons here, by pressing which direct data transfer is performed.



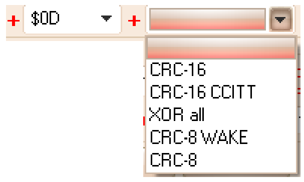
- adding checksums, which are calculated over the entire transmitted string and added as text HEX-characters to the end of the string.



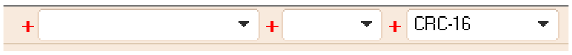
- adding characters that finalize the string. $0D - carriage return, $0A - line translation.

For example, with this terminator setup: 

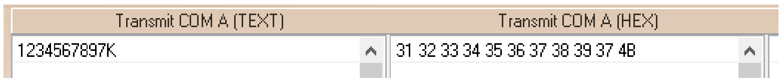
a string consisting of digits 1 through 9 will be transmitted as follows: 123456789374B$0D, where 374B is the CRC-16 checksum calculated over the string "123456789" and appended to the end as HEX-characters. This is more applicable to text-based protocols.



- adding checksums, which are calculated over the entire transmitted string and added as one or two bytes to the end of the string.

For example, with this setup of terminators: 

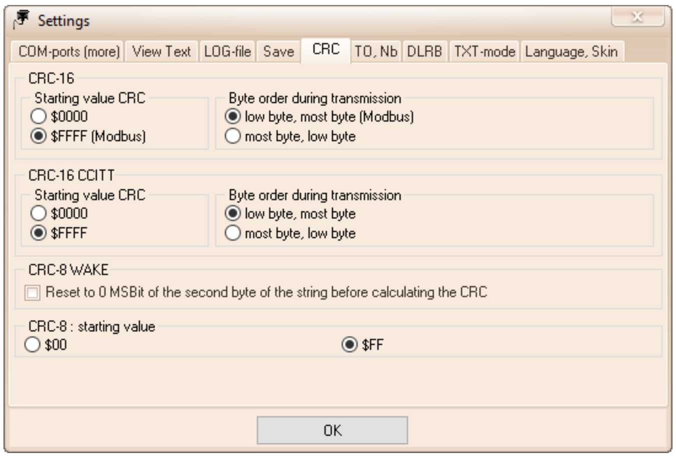
a string consisting of digits from 1 to 9 will be transmitted as follows:



Transmit COM A (TEXT)	Transmit COM A (HEX)
1234567897K	31 32 33 34 35 36 37 38 39 37 4B

that is, the bytes $37 and $4B of the CRC-16 checksum calculated above the string "123456789" are appended to the end as exactly two bytes. This is more applicable to binary protocols.

The CRC calculation is configured from the "settings" context menu, when selected, the "CRC" tab of the "Settings" window opens:



Settings

COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

CRC-16
- Starting value CRC: ○ $0000 [✓ $FFFF (Modbus)]
- Byte order during transmission: [✓ low byte, most byte (Modbus)] ○ most byte, low byte

CRC-16 CCITT
- Starting value CRC: ○ $0000 [✓ $FFFF]
- Byte order during transmission: [✓ low byte, most byte] ○ most byte, low byte

CRC-8 WAKE
- ☐ Reset to 0 MSBit of the second byte of the string before calculating the CRC

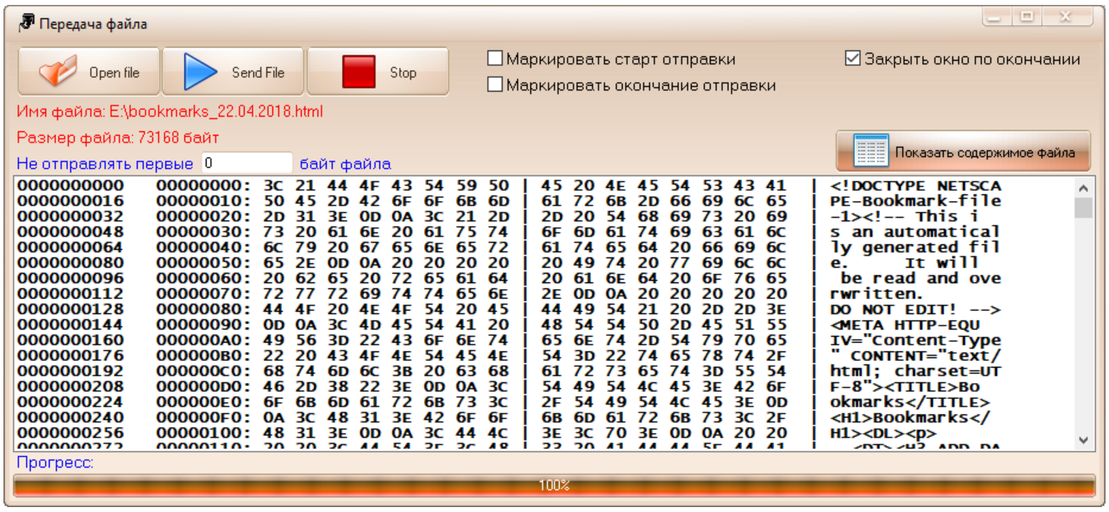
CRC-8 : starting value
- ○ $00 [✓ $FF]

OK

9b - indicator of the number of bytes transmitted in the last packet through either of the two ports.

AA - AutoAnswer – activation of the "auto answer" mode. If the mode of splitting the received data into lines by timeout is selected (Time Out TO), then when the "auto answer" is turned on, each time after the timeout expires, the terminal will automatically press the button < SEND , that is, the data entered in the "Transfer" line will be sent.

Send File - calling an additional window to transfer the selected file to the port:



☐ Маркировать старт отправки
☐ Маркировать окончание отправки
☑ Закрыть окно по окончании

Имя файла: E:\bookmarks_22.04.2018.html
Размер файла: 73168 байт
Не отправлять первые 0 байт файла

```
0000000000   00000000: 3C 21 44 4F 43 54 59 50 | 45 20 4E 45 54 53 43 41  | <!DOCTYPE NETSCA
0000000016   00000010: 50 45 2D 42 6F 6F 6B 6D | 61 72 6B 2D 66 69 6C 65  | PE-Bookmark-file
0000000032   00000020: 2D 31 3E 0D 0A 3C 21 2D | 2D 20 54 68 69 73 20 69  | -1><!-- This i
0000000048   00000030: 73 20 61 6E 20 61 75 74 | 6F 6D 61 74 69 63 61 6C  | s an automatical
0000000064   00000040: 6C 79 20 67 65 6E 65 72 | 61 74 65 64 20 66 69 6C  | ly generated fil
0000000080   00000050: 65 2E 0D 0A 20 20 20 20 | 20 49 74 20 77 69 6C 6C  | e.     It will
0000000096   00000060: 20 62 65 20 72 65 61 64 | 20 61 6E 64 20 6F 76 65  |  be read and ove
0000000112   00000070: 72 77 72 69 74 74 65 6E | 2E 0D 0A 20 20 20 20 20  | rwritten.
0000000128   00000080: 44 4F 20 4E 4F 54 20 45 | 44 49 54 21 20 2D 2D 3E  | DO NOT EDIT! -->
0000000144   00000090: 0D 0A 3C 4D 45 54 41 20 | 48 54 54 50 2D 45 51 55  | <META HTTP-EQU
0000000160   000000A0: 49 56 3D 22 43 6F 6E 74 | 65 6E 74 2D 54 79 70 65  | IV="Content-Type
0000000176   000000B0: 22 20 43 4F 4E 54 45 4E | 54 3D 22 74 65 78 74 2F  | " CONTENT="text/
0000000192   000000C0: 68 74 6D 6C 3B 20 63 68 | 61 72 73 65 74 3D 55 54  | html; charset=UT
0000000208   000000D0: 46 2D 38 22 3E 0D 0A 3C | 54 49 54 4C 45 3E 42 6F  | F-8"><TITLE>Bo
0000000224   000000E0: 6F 6B 6D 61 72 6B 73 3C | 2F 54 49 54 4C 45 3E 0D  | okmarks</TITLE>
0000000240   000000F0: 0A 3C 48 31 3E 42 6F 6F | 6B 6D 61 72 6B 73 3C 2F  | <H1>Bookmarks</
0000000256   00000100: 48 31 3E 0D 0A 3C 44 4C | 3E 3C 70 3E 0D 0A 20 20  | H1><DL><p>
```

Прогресс:
100%

With this tool, you can select, view the contents and send to the port any file located on the computer.

☐ SI ☐ Clear after Enter - setting the transfer mode of data dialed in the input line:

SI – Send Immediately – a mode in which the code of each pressed key on the keyboard is immediately sent to the port (at the moment the key is pressed). In normal mode, data transfer to the port is carried out at once by the entire line at the moment the user presses the "Enter" key.

Clear after Enter - a mode in which after the user presses the "Enter" key, the input line is automatically cleared.

SEND+$1A (SMS) +++ ATH - buttons to quickly send certain characters:

"SEND+$1A" – sends, the text typed in the input line, at the end of which the byte $1A is added. It is necessary to send SMS text when working with a GSM-modem.

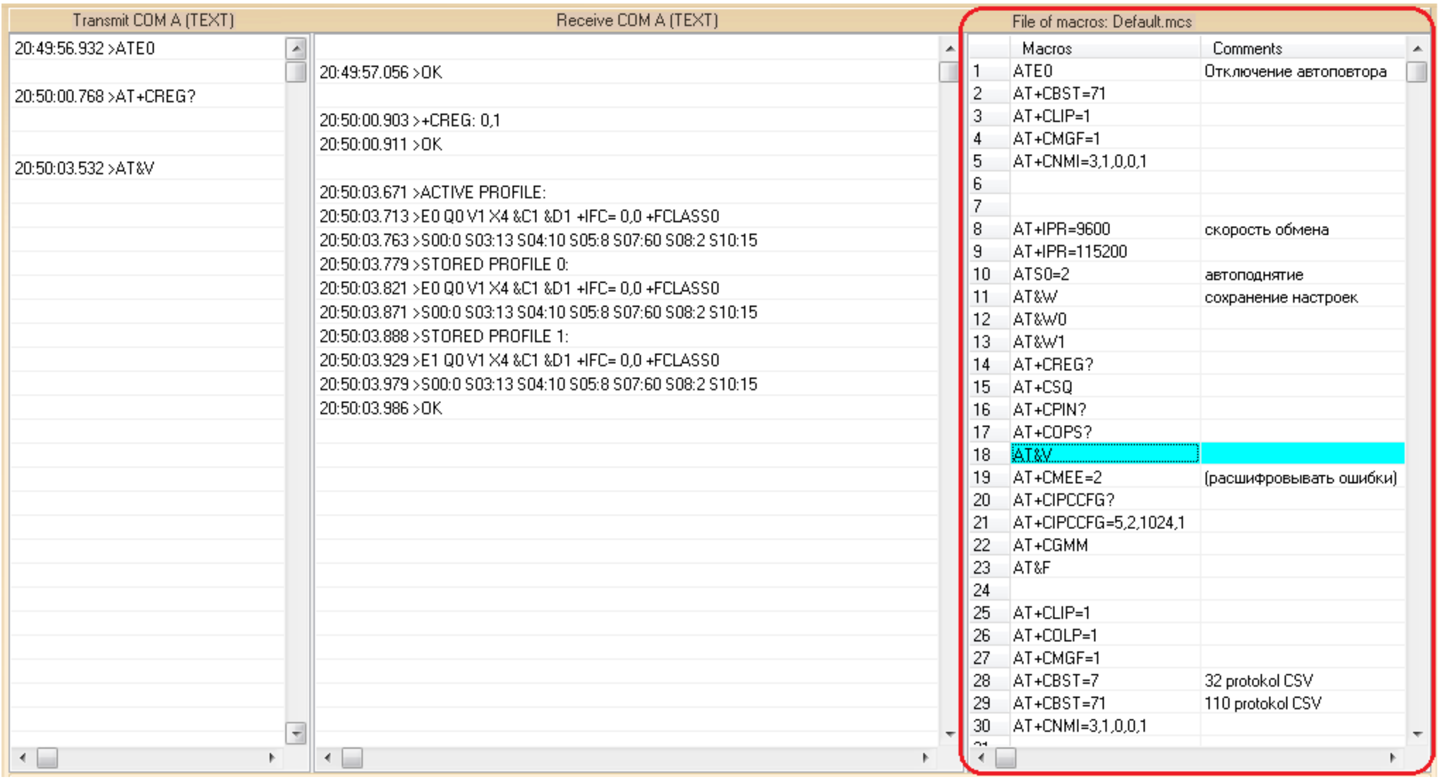
"+++" – sends three "+" characters. This is necessary to break the transparent connection during CSD-call to another terminal and switch the GSM-modem to command mode.

"ATH" – sends the string "ATH" + $0D to the port. This is necessary to end an outgoing call or reject an incoming call when working with a GSM-modem.

Macros

Macro support is implemented in the terminal as a separate table, the display of which is enabled by a checkbox ☐ Macros in the "View" field.

The macro table is **ALWAYS** displayed in the terminal as the rightmost column. It is designed for 1000 macros, each with its own comment. Any macro can be sent to the port at any time by double-clicking ***on its number*** in the table or by selecting it and then pressing the F9 key.



Transmit COM A (TEXT)

```
20:49:56.932 >ATE0

20:50:00.768 >AT+CREG?

20:50:03.532 >AT&V
```

Receive COM A (TEXT)

```
20:49:57.056 >OK

20:50:00.903 >+CREG: 0,1
20:50:00.911 >OK

20:50:03.671 >ACTIVE PROFILE:
20:50:03.713 >E0 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
20:50:03.763 >S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
20:50:03.779 >STORED PROFILE 0:
20:50:03.821 >E0 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
20:50:03.871 >S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
20:50:03.888 >STORED PROFILE 1:
20:50:03.929 >E1 Q0 V1 X4 &C1 &D1 +IFC= 0,0 +FCLASS0
20:50:03.979 >S00:0 S03:13 S04:10 S05:8 S07:60 S08:2 S10:15
20:50:03.986 >OK
```

File of macros: Default.mcs

	Macros	Comments
1	ATE0	Отключение автоповтора
2	AT+CBST=71	
3	AT+CLIP=1	
4	AT+CMGF=1	
5	AT+CNMI=3,1,0,0,1	
6		
7		
8	AT+IPR=9600	скорость обмена
9	AT+IPR=115200	
10	ATS0=2	автоподнятие
11	AT&W	сохранение настроек
12	AT&W0	
13	AT&W1	
14	AT+CREG?	
15	AT+CSQ	
16	AT+CPIN?	
17	AT+COPS?	
18	AT&V	
19	AT+CMEE=2	(расшифровывать ошибки)
20	AT+CIPCCFG?	
21	AT+CIPCCFG=5,2,1024,1	
22	AT+CGMM	
23	AT&F	
24		
25	AT+CLIP=1	
26	AT+COLP=1	
27	AT+CMGF=1	
28	AT+CBST=7	32 protokol CSV
29	AT+CBST=71	110 protokol CSV
30	AT+CNMI=3,1,0,0,1	

When the terminal starts, the macro table is loaded from a file with the .mcs extension, and when the program is closed, the table is automatically saved to the same file. By default, when the program starts, if the macro file is not found, then an empty Default.mcs file is generated and opened in the MACROS subdirectory (\MACROS\Default.mcs). At the same time, at any time you can select another macro file with which the terminal will work. The name of the file the terminal is currently working with is indicated above the table.

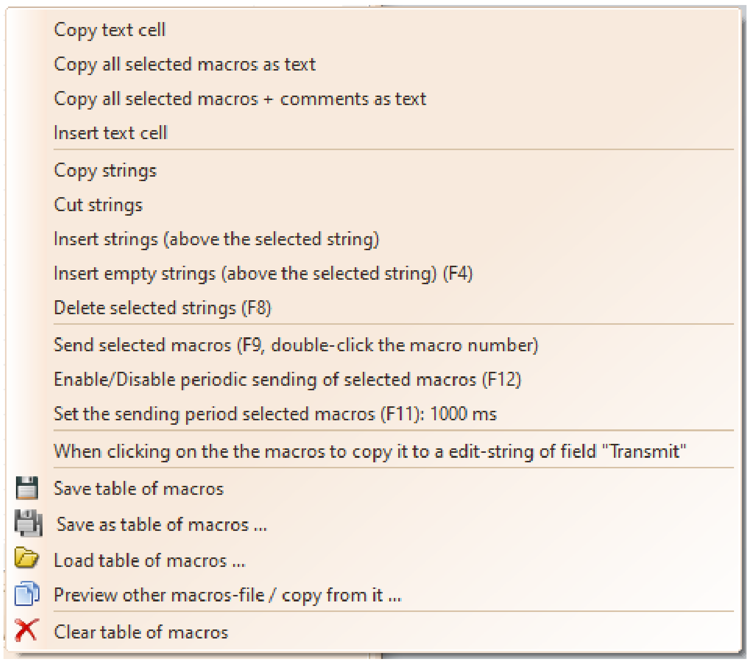
The macro table supports ***multi-selection*** of lines. If you select one macro and hold down the left mouse button, then moving the cursor over other macros will highlight them. In addition, the selection mode with the CTRL key is supported - while holding it, you can select/deselect individual macros. With the highlighted macros, you can do the following:

- copy, cut or delete (performed via the context menu);

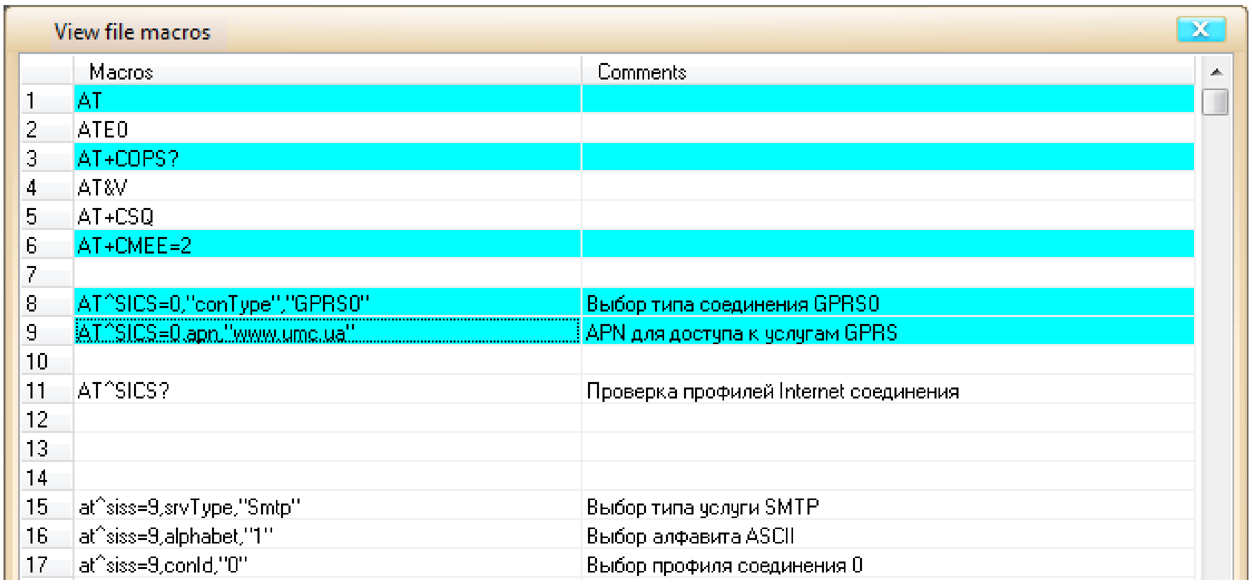
- send once with a certain period (F9 or context menu);

- enable cyclic sending with a certain period (F12 or context menu);

All operations that can be performed with macros are available from the context menu:


Copy text cell
Copy all selected macros as text
Copy all selected macros + comments as text
Insert text cell
Copy strings
Cut strings
Insert strings (above the selected string)
Insert empty strings (above the selected string) (F4)
Delete selected strings (F8)
Send selected macros (F9, double-click the macro number)
Enable/Disable periodic sending of selected macros (F12)
Set the sending period selected macros (F11): 1000 ms
When clicking on the the macros to copy it to a edit-string of field "Transmit"
Save table of macros
Save as table of macros ...
Load table of macros ...
Preview other macros-file / copy from it ...
Clear table of macros

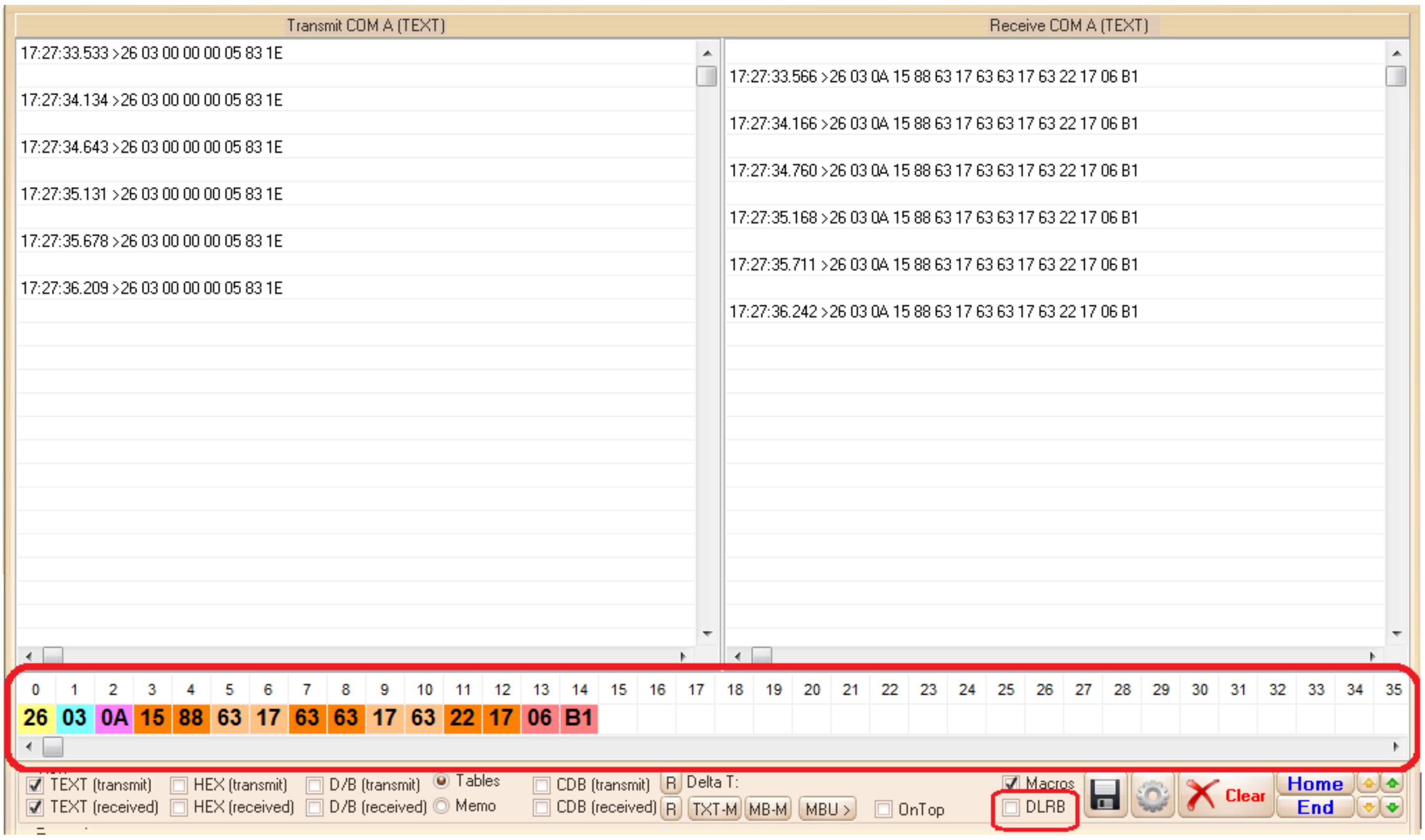
The meaning of all menu items is clear from their names. When you select the *"Preview another macros-file / copy from it ..."* menu item, a file selection dialog will open, and then an additional window in which you can mark the necessary macros and copy them to the main macro table from the context menu. This window also supports ***multi-selection*** of lines mode:


View file macros

	Macros	Comments
1	AT	
2	ATE0	
3	AT+COPS?	
4	AT&V	
5	AT+CSQ	
6	AT+CMEE=2	
7		
8	AT^SICS=0,"conType","GPRS0"	Выбор типа соединения GPRS0
9	AT^SICS=0,apn,"www.umc.ua"	APN для доступа к услугам GPRS
10		
11	AT^SICS?	Проверка профилей Internet соединения
12		
13		
14		
15	at^siss=9,srvType,"Smtp"	Выбор типа услуги SMTP
16	at^siss=9,alphabet,"1"	Выбор алфавита ASCII
17	at^siss=9,conId,"0"	Выбор профиля соединения 0

DLRB (Detail Last Received Block)

The terminal has the function of detailed display of the received data in the form of a table, where this data is placed. This is relevant when working with positional protocols, where the position of certain bytes in a message is fixed and does not change over time. This table is enabled by a checkbox ☐ DLRB in the "View" field and is always displayed at the bottom of the main tables or Memo:

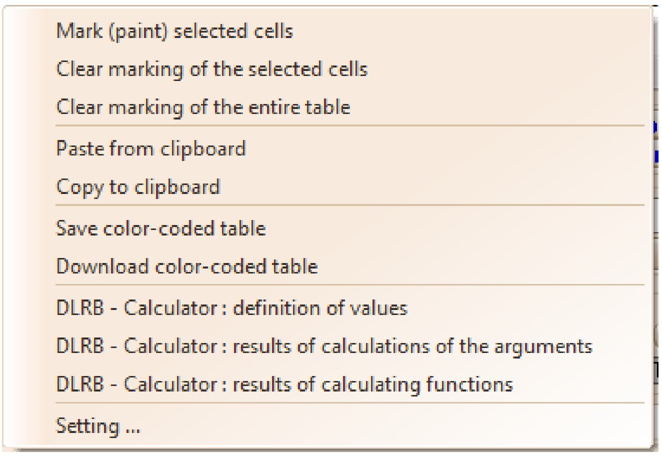


```
17:27:33.533 >26 03 00 00 00 05 83 1E
17:27:34.134 >26 03 00 00 00 05 83 1E
17:27:34.643 >26 03 00 00 00 05 83 1E
17:27:35.131 >26 03 00 00 00 05 83 1E
17:27:35.678 >26 03 00 00 00 05 83 1E
17:27:36.209 >26 03 00 00 00 05 83 1E
```

```
17:27:33.566 >26 03 0A 15 88 63 17 63 63 17 63 22 17 06 B1
17:27:34.166 >26 03 0A 15 88 63 17 63 63 17 63 22 17 06 B1
17:27:34.760 >26 03 0A 15 88 63 17 63 63 17 63 22 17 06 B1
17:27:35.168 >26 03 0A 15 88 63 17 63 63 17 63 22 17 06 B1
17:27:35.711 >26 03 0A 15 88 63 17 63 63 17 63 22 17 06 B1
17:27:36.242 >26 03 0A 15 88 63 17 63 63 17 63 22 17 06 B1
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
26	03	0A	15	88	63	17	63	63	17	63	22	17	06	B1																					

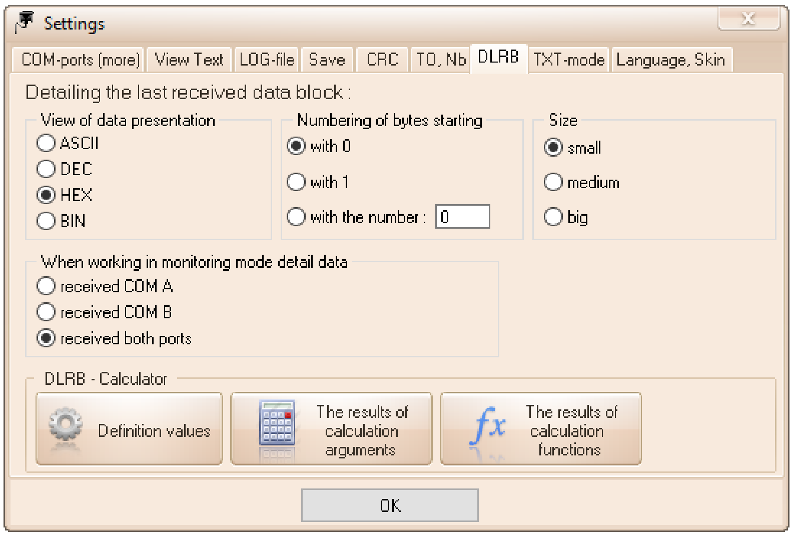
The block of received data is always split in the table by byte, and they can be displayed in the table in TEXT-, HEX-, DEC- or BIN- format. All bytes are numbered starting from 0, 1 or any given number. Each individual byte (individual table cell) can be filled with its own individual color to improve visual perception. All actions with the table, as well as its configuration, are available from the context menu:



Mark (paint) selected cells
Clear marking of the selected cells
Clear marking of the entire table
Paste from clipboard
Copy to clipboard
Save color-coded table
Download color-coded table
DLRB - Calculator : definition of values
DLRB - Calculator : results of calculations of the arguments
DLRB - Calculator : results of calculating functions
Setting ...

Cells are selected with the mouse pointer. The color scheme can be saved to a file under a specific name and then loaded later. You can also copy data from the table to the clipboard as plain text, as well as paste plain text from the clipboard into the table (insertion always occurs starting from the very first character in the table and does not depend on the selection of certain cells, and copying is performed taking into account the selection) ...

The DLRB table view is configured on the "DLRB" tab in the "Settings" window, which opens after selecting the "Settings ..." item in the context menu:


Settings

COM-ports (more) | View Text | LOG-file | Save | CRC | TO, Nb | DLRB | TXT-mode | Language, Skin

Detailing the last received data block :

View of data presentation: ASCII / DEC / [✓ HEX] / BIN

Numbering of bytes starting: [✓ with 0] / with 1 / with the number : 0

Size: [✓ small] / medium / big

When working in monitoring mode detail data: received COM A / received COM B / [✓ received both ports]

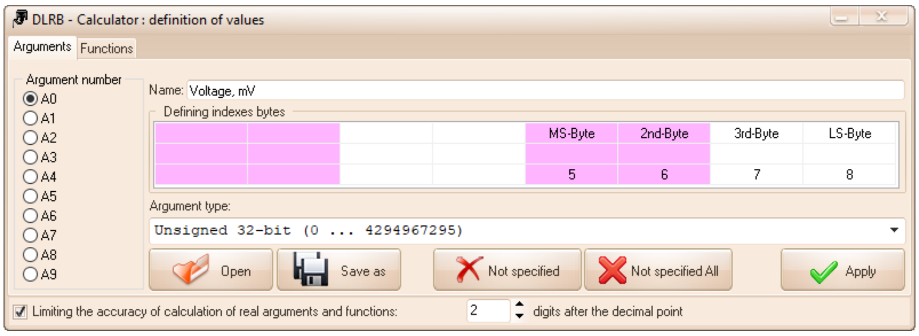
DLRB - Calculator
Definition values
The results of calculation arguments
The results of calculation functions
OK

Another tool is closely related to the work of the DLRB table - the ***DLRB-Calculator***, which allows you to calculate a set (up to 10 items) of arguments and a set of functions based on them in real time.

An ***argument*** is a number calculated based on one or more (two or four) bytes of the received data block, and bytes can be interpreted both in binary form and in text form (as two hex characters). In total, you can specify up to 10 arguments with fixed names from "A0" to "A9".

A ***function*** is a number calculated from a given formula, using the same arguments ("A0" - "A9") and math symbols and parentheses.

For example, the terminal receives a block of data in which bytes 5, 6, 7, and 8 contain the value of an unsigned 32-bit integer starting with the most significant (fifth) byte. Let's say this is the voltage in millivolts being transferred across a 50 ohm resistor. With the help of the DLRB-Calculator, it is possible to calculate this value in real time, as well as the quantities associated with it using formulas. So, the voltage value itself will be an argument and you can set it like this:


DLRB - Calculator : definition of values

Arguments | Functions

Argument number: [✓ A0] / A1 / A2 / A3 / A4 / A5 / A6 / A7 / A8 / A9

Name: Voltage, mV

Defining indexes bytes

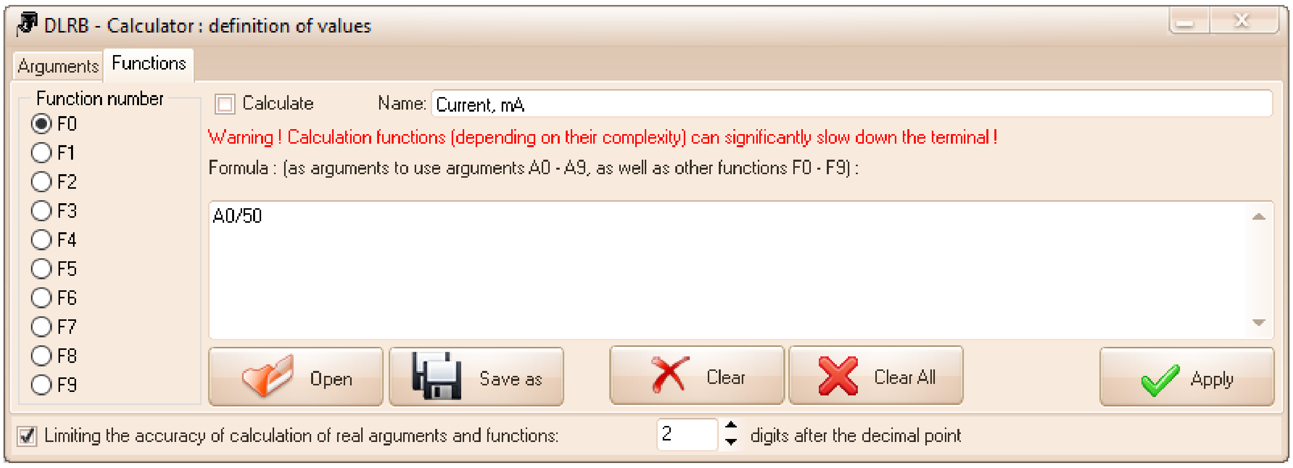
				MS-Byte	2nd-Byte	3rd-Byte	LS-Byte
				5	6	7	8

Argument type:

Unsigned 32-bit (0 ... 4294967295)

[✓] Limiting the accuracy of calculation of real arguments and functions: 2 digits after the decimal point

And functions in this way:


DLRB - Calculator : definition of values

Arguments | Functions

Function number: (● F0) F1 F2 F3 F4 F5 F6 F7 F8 F9

☐ Calculate Name: Current, mA

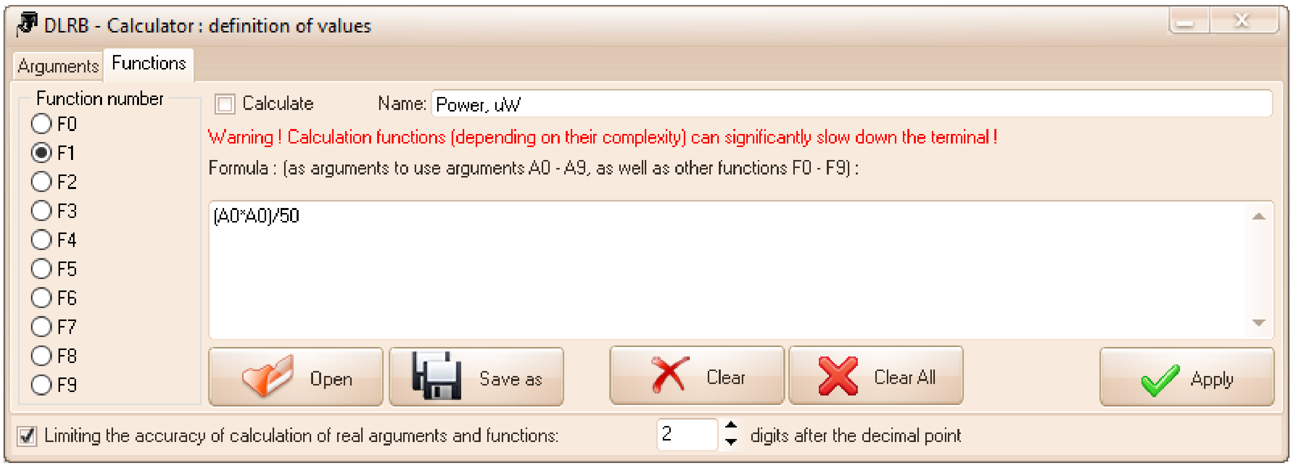
Warning ! Calculation functions (depending on their complexity) can significantly slow down the terminal !

Formula : (as arguments to use arguments A0 - A9, as well as other functions F0 - F9) :

```
A0/50
```

Open | Save as | Clear | Clear All | Apply

☑ Limiting the accuracy of calculation of real arguments and functions: 2 digits after the decimal point


DLRB - Calculator : definition of values

Arguments | Functions

Function number: F0 (● F1) F2 F3 F4 F5 F6 F7 F8 F9

☐ Calculate Name: Power, uW

Warning ! Calculation functions (depending on their complexity) can significantly slow down the terminal !

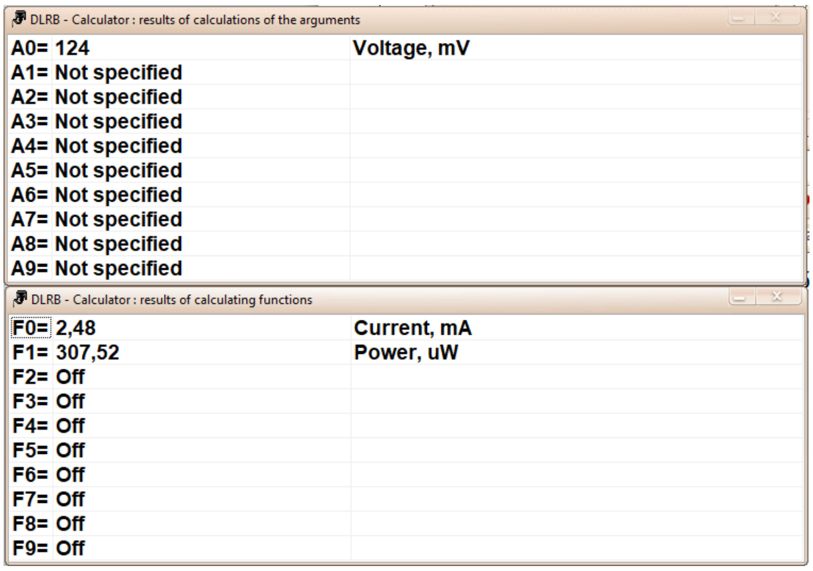
Formula : (as arguments to use arguments A0 - A9, as well as other functions F0 - F9) :

```
(A0*A0)/50
```

Open | Save as | Clear | Clear All | Apply

☑ Limiting the accuracy of calculation of real arguments and functions: 2 digits after the decimal point

Then the calculation results will be presented as follows:

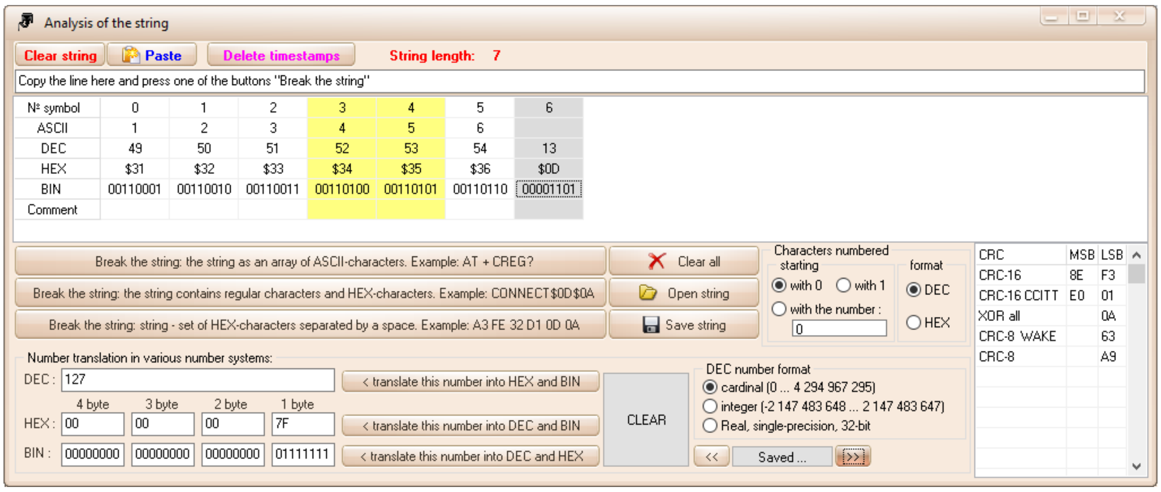

DLRB - Calculator : results of calculations of the arguments

A0= 124	Voltage, mV
A1= Not specified	
A2= Not specified	
A3= Not specified	
A4= Not specified	
A5= Not specified	
A6= Not specified	
A7= Not specified	
A8= Not specified	
A9= Not specified	

DLRB - Calculator : results of calculating functions

F0= 2,48	Current, mA
F1= 307,52	Power, uW
F2= Off	
F3= Off	
F4= Off	
F5= Off	
F6= Off	
F7= Off	
F8= Off	
F9= Off	

"Analysis of the string" tool (button "String")


Analysis of the string
Clear string
Paste
Delete timestamps

String length: 7

Copy the line here and press one of the buttons "Break the string"

№ symbol	0	1	2	3	4	5	6
ASCII	1	2	3	4	5	6	
DEC	49	50	51	52	53	54	13
HEX	$31	$32	$33	$34	$35	$36	$0D
BIN	00110001	00110010	00110011	00110100	00110101	00110110	00001101
Comment							

Break the string: the string as an array of ASCII-characters. Example: AT + CREG?
Break the string: the string contains regular characters and HEX-characters. Example: CONNECT$0D$0A
Break the string: string - set of HEX-characters separated by a space. Example: A3 FE 32 D1 0D 0A
Clear all
Open string
Save string

Characters numbered starting: [✓ with 0] with 1 with the number : 0

format: [✓ DEC] HEX

CRC	MSB	LSB
CRC-16	8E	F3
CRC-16 CCITT	E0	01
XOR all		0A
CRC-8 WAKE		63
CRC-8		A9

Number translation in various number systems:

DEC : 127

	4 byte	3 byte	2 byte	1 byte
HEX :	00	00	00	7F
BIN :	00000000	00000000	00000000	01111111

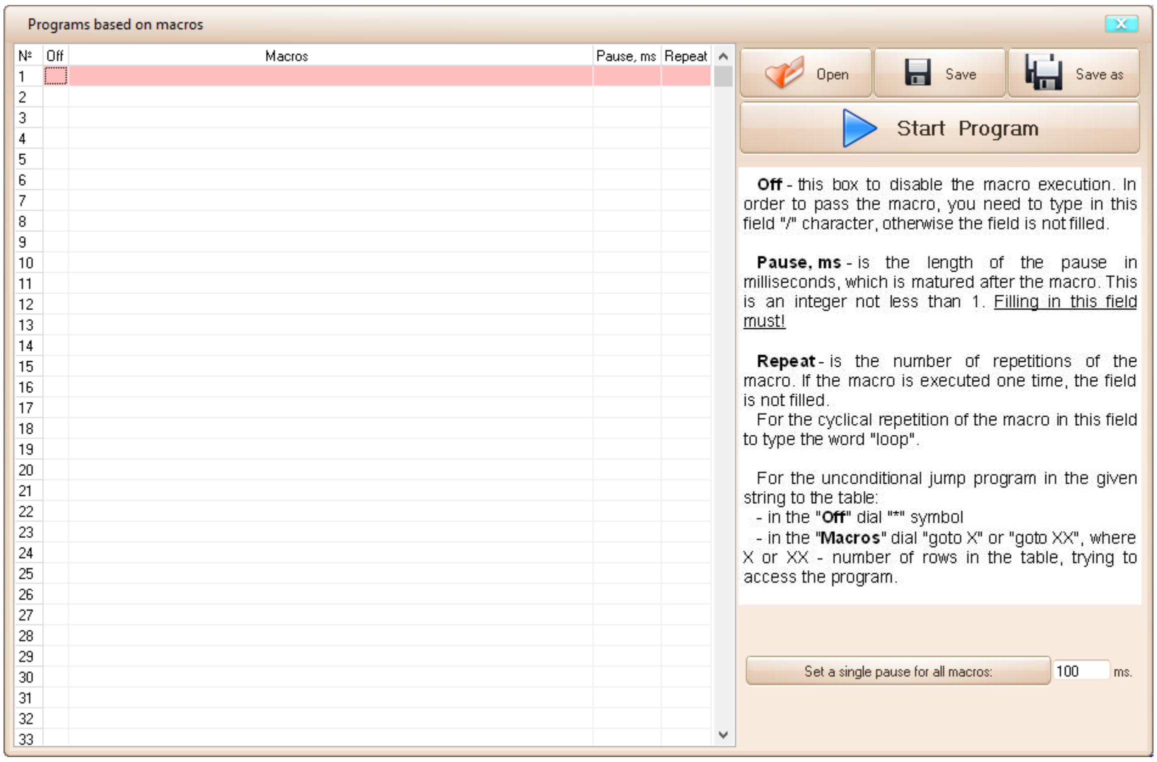
< translate this number into HEX and BIN
< translate this number into DEC and BIN
< translate this number into DEC and HEX
CLEAR
<<
Saved ...
>>

DEC number format: [✓ cardinal (0 ... 4 294 967 295)] integer (-2 147 483 648 ... 2 147 483 647) Real, single-precision, 32-bit

In this window, you can analyze the string in detail by copying it through the clipboard and breaking it down by bytes, determine its length, highlight individual bytes with a specific color and attach your own comment to each. Such a detailed analysis can be saved in a separate file and loaded later.

In addition, in this window, it is possible to calculate various checksums of a given string, translate a given number between different calculation systems (DEC <> HEX <> BIN), decompose it into separate bytes, and perform a bitwise shift of these bytes to the left and to the right.

"Programs based on macros" tool (button"Programs")



№	Off	Macros	Pause, ms	Repeat
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				

Off - this box to disable the macro execution. In order to pass the macro, you need to type in this field "/" character, otherwise the field is not filled.

Pause, ms - is the length of the pause in milliseconds, which is matured after the macro. This is an integer not less than 1. Filling in this field must!

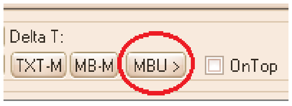
Repeat - is the number of repetitions of the macro. If the macro is executed one time, the field is not filled.
For the cyclical repetition of the macro in this field to type the word "loop".

For the unconditional jump program in the given string to the table:
- in the "**Off**" dial "*" symbol
- in the "**Macros**" dial "goto X" or "goto XX", where X or XX - number of rows in the table, trying to access the program.

Set a single pause for all macros: 100 ms.

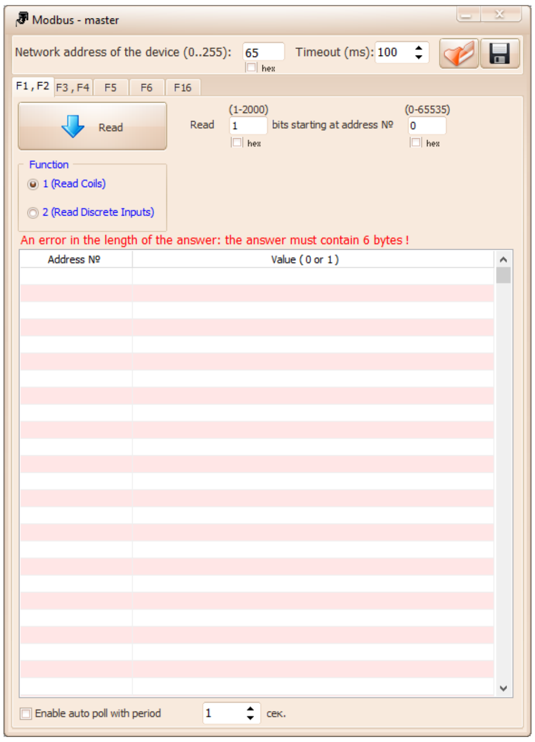
In this window, it is possible to implement the execution of the simplest programs based on sending specified macros with certain pauses between them with the possibility of looping the execution of individual blocks.

"Modbus utilities" tool (button "MBU>")

Pressing the button 

allows you to call one of two windows, which are designed to support work with the MODBUS protocol. In this case, a computer with a terminal can act both as a Master, which polls the Slave devices, and imitate a Slave device from itself.

Modbus-master window:


Modbus - master

Network address of the device (0..255): 65 [ ] hex   Timeout (ms): 100

F1 , F2 | F3 , F4 | F5 | F6 | F16

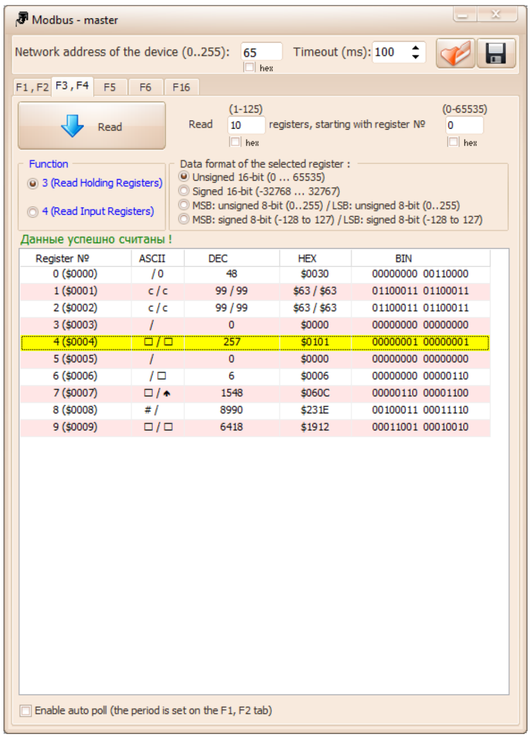
Read   Read (1-2000) 1 [ ] hex bits starting at address № (0-65535) 0 [ ] hex

Function: [✓ 1 (Read Coils)] 2 (Read Discrete Inputs)

An error in the length of the answer: the answer must contain 6 bytes !

Address №	Value (0 or 1)

[ ] Enable auto poll with period 1 сек.


Modbus - master

Network address of the device (0..255): 65 [ ] hex   Timeout (ms): 100

F1 , F2 | F3 , F4 | F5 | F6 | F16

Read   Read (1-125) 10 [ ] hex registers, starting with register № (0-65535) 0 [ ] hex

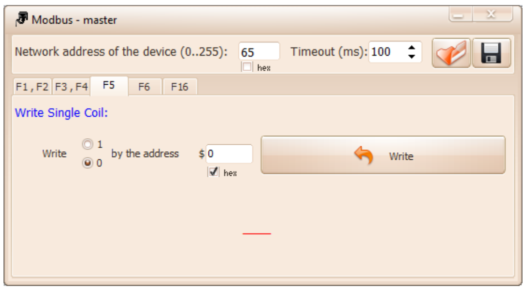
Function: [✓ 3 (Read Holding Registers)] 4 (Read Input Registers)

Data format of the selected register :
[✓ Unsigned 16-bit (0 … 65535)]
Signed 16-bit (-32768 … 32767)
MSB: unsigned 8-bit (0..255) / LSB: unsigned 8-bit (0..255)
MSB: signed 8-bit (-128 to 127) / LSB: signed 8-bit (-128 to 127)

Данные успешно считаны !

Register №	ASCII	DEC	HEX	BIN
0 ($0000)	/ 0	48	$0030	00000000 00110000
1 ($0001)	c / c	99 / 99	$63 / $63	01100011 01100011
2 ($0002)	c / c	99 / 99	$63 / $63	01100011 01100011
3 ($0003)	/	0	$0000	00000000 00000000
4 ($0004)	□ / □	257	$0101	00000001 00000001
5 ($0005)	/	0	$0000	00000000 00000000
6 ($0006)	/ □	6	$0006	00000000 00000110
7 ($0007)	□ / ♠	1548	$060C	00000110 00001100
8 ($0008)	# /	8990	$231E	00100011 00011110
9 ($0009)	□ / □	6418	$1912	00011001 00010010

[ ] Enable auto poll (the period is set on the F1, F2 tab)

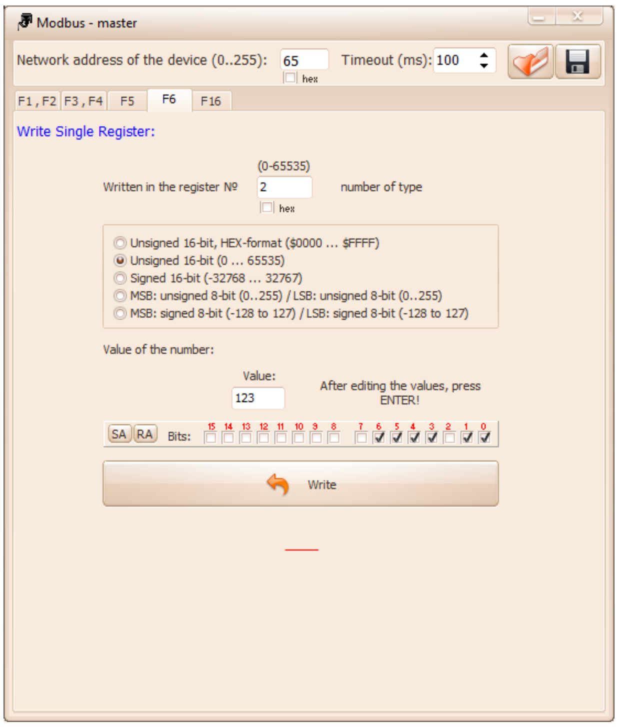

Modbus - master

Network address of the device (0..255): 65 [ ] hex   Timeout (ms): 100

F1 , F2 | F3 , F4 | F5 | F6 | F16

Write Single Coil:

Write 1 [✓ 0] by the address $ 0 [✓] hex   Write


Modbus - master

Network address of the device (0..255): 65 [ ] hex   Timeout (ms): 100

F1 , F2 | F3 , F4 | F5 | F6 | F16

Write Single Register:

Written in the register № (0-65535) 2 [ ] hex   number of type

Unsigned 16-bit, HEX-format ($0000 … $FFFF)
[✓ Unsigned 16-bit (0 … 65535)]
Signed 16-bit (-32768 … 32767)
MSB: unsigned 8-bit (0..255) / LSB: unsigned 8-bit (0..255)
MSB: signed 8-bit (-128 to 127) / LSB: signed 8-bit (-128 to 127)

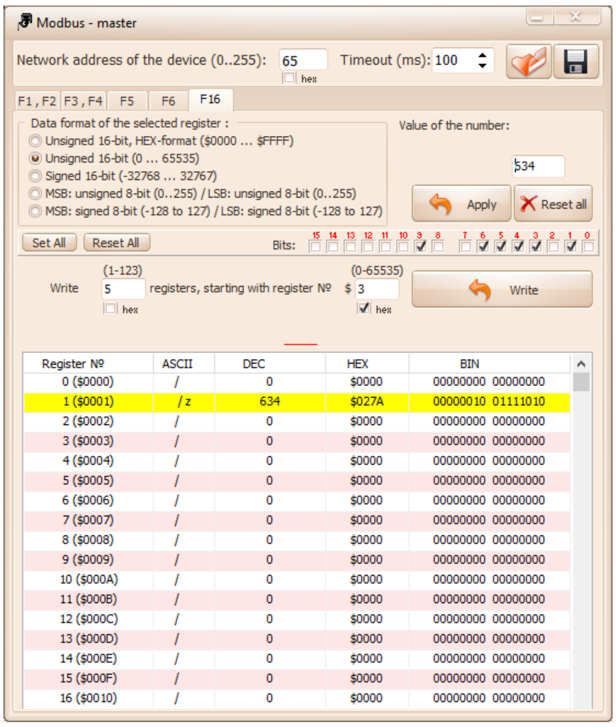
Value of the number:

Value: 123   After editing the values, press ENTER!

SA RA Bits:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
☐	☐	☐	☐	☐	☐	☐	☐	☐	✓	✓	✓	✓	☐	✓	✓

Write


Modbus - master

Network address of the device (0..255): 65 [ ] hex   Timeout (ms): 100

F1 , F2 | F3 , F4 | F5 | F6 | F16

Data format of the selected register :
Unsigned 16-bit, HEX-format ($0000 … $FFFF)
[✓ Unsigned 16-bit (0 … 65535)]
Signed 16-bit (-32768 … 32767)
MSB: unsigned 8-bit (0..255) / LSB: unsigned 8-bit (0..255)
MSB: signed 8-bit (-128 to 127) / LSB: signed 8-bit (-128 to 127)

Value of the number: 634   Apply   Reset all

Set All   Reset All   Bits:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
☐	☐	☐	☐	☐	☐	✓	☐	☐	✓	✓	✓	✓	☐	✓	☐

Write (1-123) 5 [ ] hex registers, starting with register № (0-65535) $ 3 [✓] hex   Write

Register №	ASCII	DEC	HEX	BIN
0 ($0000)	/	0	$0000	00000000 00000000
1 ($0001)	/ z	634	$027A	00000010 01111010
2 ($0002)	/	0	$0000	00000000 00000000
3 ($0003)	/	0	$0000	00000000 00000000
4 ($0004)	/	0	$0000	00000000 00000000
5 ($0005)	/	0	$0000	00000000 00000000
6 ($0006)	/	0	$0000	00000000 00000000
7 ($0007)	/	0	$0000	00000000 00000000
8 ($0008)	/	0	$0000	00000000 00000000
9 ($0009)	/	0	$0000	00000000 00000000
10 ($000A)	/	0	$0000	00000000 00000000
11 ($000B)	/	0	$0000	00000000 00000000
12 ($000C)	/	0	$0000	00000000 00000000
13 ($000D)	/	0	$0000	00000000 00000000
14 ($000E)	/	0	$0000	00000000 00000000
15 ($000F)	/	0	$0000	00000000 00000000
16 ($0010)	/	0	$0000	00000000 00000000

1, 2, 3, 4, 5, 6 and 16 Modbus-protocol functions are supported. To work with a device of the Slave-type, you need an adapter from the RS-232 to RS-485 type

or USB -> RS-485, type

Next, you need to set the network address of the device (1..255) and the timeout for waiting for a response.

To read data from the device, functions 1 - 4 also set the number of registers to be subtracted and the address of the first register from which reading begins.

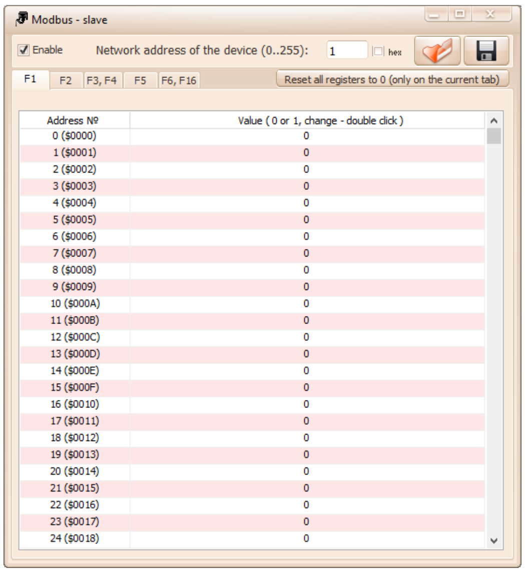
To write to the device by functions 5, 6 and 16, the value of the number that must be written and the address of the register into which it is written are set (for the 16th function, it is necessary to fill in the table of values and indicate the start register for writing).

Next, the button is pressed for reading Read, and for recording - the button Write.

Also, on the tabs 1 and 2 of the functions (below), it is possible to set the period and enable the mode of periodic polling by the terminal of the connected slave device.

Warning! Almost all fields for specifying the network address, register addresses and values support increasing and decreasing the current value using the "up arrow" and "down arrow" keyboard keys !!!

Modbus - slave window:

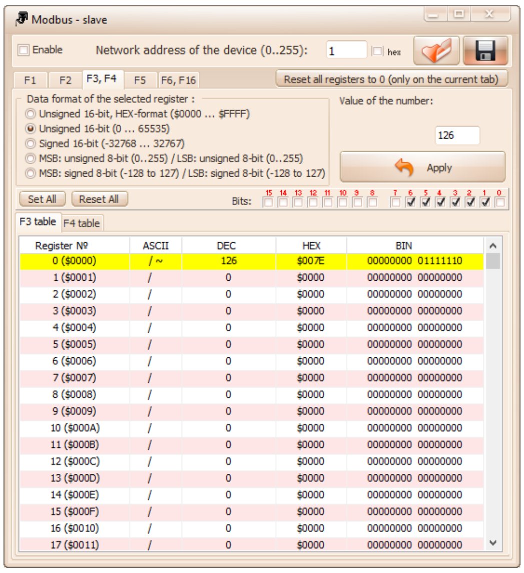


Modbus - slave

[✓] Enable   Network address of the device (0..255): 1   ☐ hex

Tabs: F1 | F2 | F3, F4 | F5 | F6, F16   Reset all registers to 0 (only on the current tab)

Address №	Value (0 or 1, change - double click)
0 ($0000)	0
1 ($0001)	0
2 ($0002)	0
3 ($0003)	0
4 ($0004)	0
5 ($0005)	0
6 ($0006)	0
7 ($0007)	0
8 ($0008)	0
9 ($0009)	0
10 ($000A)	0
11 ($000B)	0
12 ($000C)	0
13 ($000D)	0
14 ($000E)	0
15 ($000F)	0
16 ($0010)	0
17 ($0011)	0
18 ($0012)	0
19 ($0013)	0
20 ($0014)	0
21 ($0015)	0
22 ($0016)	0
23 ($0017)	0
24 ($0018)	0



Modbus - slave

☐ Enable   Network address of the device (0..255): 1   ☐ hex

Tabs: F1 | F2 | F3, F4 | F5 | F6, F16   Reset all registers to 0 (only on the current tab)

Data format of the selected register :
- Unsigned 16-bit, HEX-format ($0000 … $FFFF)
- [✓ Unsigned 16-bit (0 … 65535)]
- Signed 16-bit (-32768 … 32767)
- MSB: unsigned 8-bit (0..255) / LSB: unsigned 8-bit (0..255)
- MSB: signed 8-bit (-128 to 127) / LSB: signed 8-bit (-128 to 127)

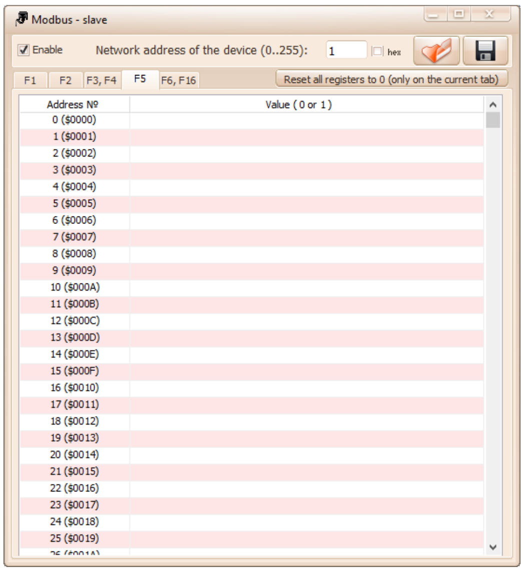
Value of the number: 126   Apply

Set All   Reset All   Bits:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
☐	☐	☐	☐	☐	☐	☐	☐	☐	✓	✓	✓	✓	✓	✓	☐

F3 table | F4 table

Register №	ASCII	DEC	HEX	BIN
0 ($0000)	/ ~	126	$007E	00000000 01111110
1 ($0001)	/	0	$0000	00000000 00000000
2 ($0002)	/	0	$0000	00000000 00000000
3 ($0003)	/	0	$0000	00000000 00000000
4 ($0004)	/	0	$0000	00000000 00000000
5 ($0005)	/	0	$0000	00000000 00000000
6 ($0006)	/	0	$0000	00000000 00000000
7 ($0007)	/	0	$0000	00000000 00000000
8 ($0008)	/	0	$0000	00000000 00000000
9 ($0009)	/	0	$0000	00000000 00000000
10 ($000A)	/	0	$0000	00000000 00000000
11 ($000B)	/	0	$0000	00000000 00000000
12 ($000C)	/	0	$0000	00000000 00000000
13 ($000D)	/	0	$0000	00000000 00000000
14 ($000E)	/	0	$0000	00000000 00000000
15 ($000F)	/	0	$0000	00000000 00000000
16 ($0010)	/	0	$0000	00000000 00000000
17 ($0011)	/	0	$0000	00000000 00000000

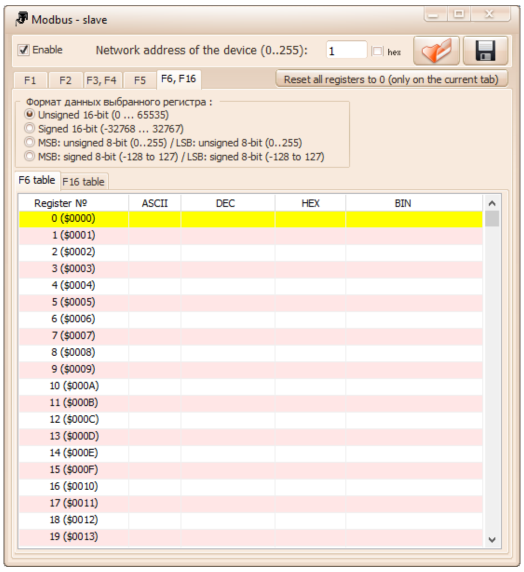


Modbus - slave

[✓] Enable   Network address of the device (0..255): 1   ☐ hex

Tabs: F1 | F2 | F3, F4 | F5 | F6, F16   Reset all registers to 0 (only on the current tab)

Address №	Value (0 or 1)
0 ($0000)	
1 ($0001)	
2 ($0002)	
3 ($0003)	
4 ($0004)	
5 ($0005)	
6 ($0006)	
7 ($0007)	
8 ($0008)	
9 ($0009)	
10 ($000A)	
11 ($000B)	
12 ($000C)	
13 ($000D)	
14 ($000E)	
15 ($000F)	
16 ($0010)	
17 ($0011)	
18 ($0012)	
19 ($0013)	
20 ($0014)	
21 ($0015)	
22 ($0016)	
23 ($0017)	
24 ($0018)	
25 ($0019)	



Modbus - slave

[✓] Enable   Network address of the device (0..255): 1   ☐ hex

Tabs: F1 | F2 | F3, F4 | F5 | F6, F16   Reset all registers to 0 (only on the current tab)

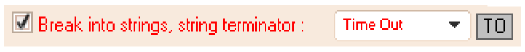
Формат данных выбранного регистра :
- [✓ Unsigned 16-bit (0 … 65535)]
- Signed 16-bit (-32768 … 32767)
- MSB: unsigned 8-bit (0..255) / LSB: unsigned 8-bit (0..255)
- MSB: signed 8-bit (-128 to 127) / LSB: signed 8-bit (-128 to 127)

F6 table | F16 table

Register №	ASCII	DEC	HEX	BIN
0 ($0000)				
1 ($0001)				
2 ($0002)				
3 ($0003)				
4 ($0004)				
5 ($0005)				
6 ($0006)				
7 ($0007)				
8 ($0008)				
9 ($0009)				
10 ($000A)				
11 ($000B)				
12 ($000C)				
13 ($000D)				
14 ($000E)				
15 ($000F)				
16 ($0010)				
17 ($0011)				
18 ($0012)				
19 ($0013)				

In this mode, the terminal imitates a Modbus-slave device that can poll an external Master via the RS-485 interface. 1, 2, 3, 4, 5, 6 and 16 functions are supported.

For the terminal to work correctly in this mode, you need to select the breakdown of the received data by timeout:


[✓] Break into strings, string terminator : Time Out TO

The value of this timeout will determine the time after which a response will be sent to the Master if a correct request is received from it.

Next, you need to set the network address of the device (1..255) and fill in the contents of the device registers in the corresponding tables.

For functions 5, 6 and 16, all data that the Master writes to the simulated device will be recorded and displayed in tables on the corresponding tabs.

At the end of the setup, in order for the terminal to start analyzing requests, form and send responses, you must activate the "Enable" checkbox in the upper left part of the window.

The contents of the tables can be saved to a separate file for later loading.

Warning! Almost all fields for specifying the network address, register addresses and values support increasing and decreasing the current value using the "up arrow" and "down arrow" keyboard keys !!!