

Руководство пользователя
Терминальная программа (версия 3.1.0)

[Поддержать проект](#)

andrey.abdulkayumov@gmail.com

Оглавление

Краткое описание.....	3
Без протокола	4
Обычный режим работы.....	5
Циклический режим работы	5
Modbus	6
Обычный режим работы.....	7
Чтение регистров Modbus	7
Запись регистров Modbus	8
0x05 Запись одного флага	8
0x0F Запись нескольких флагов	8
0x06 Запись одного регистра	9
0x10 Запись нескольких регистров.....	9
Циклический режим работы	11
Представления	12
Макросы.....	13
Без протокола	14
Modbus	15
Статьи на Хабр	16
Ссылка на скачивание	16
История версий	17
3.1.0.....	17
3.0.0.....	17
2.7.0.....	17

Краткое описание

Терминальная программа – это многофункциональный кроссплатформенный инструмент инженера, который позволяет взаимодействовать с внешним устройством или сервером. Приложение поддерживает различные вариации протокола Modbus, а также дает возможность работы с данными в строковом или байтовом формате.

Приложение тестировалось на Windows 10/11, Ubuntu и Astra Linux.

Поддерживаются темная и светлая темы оформления.

Есть два основных режима работы: «Без протокола» и «Modbus».

У каждого из них свою очередь есть два способа взаимодействия с хостом: "Обычный" и "Циклический опрос". Между ними можно переключаться во время работы. Данные на вкладках не теряются при переключении.

Важно: если переключиться в обычный режим пока идет циклический опрос, то сам опрос прекратится.

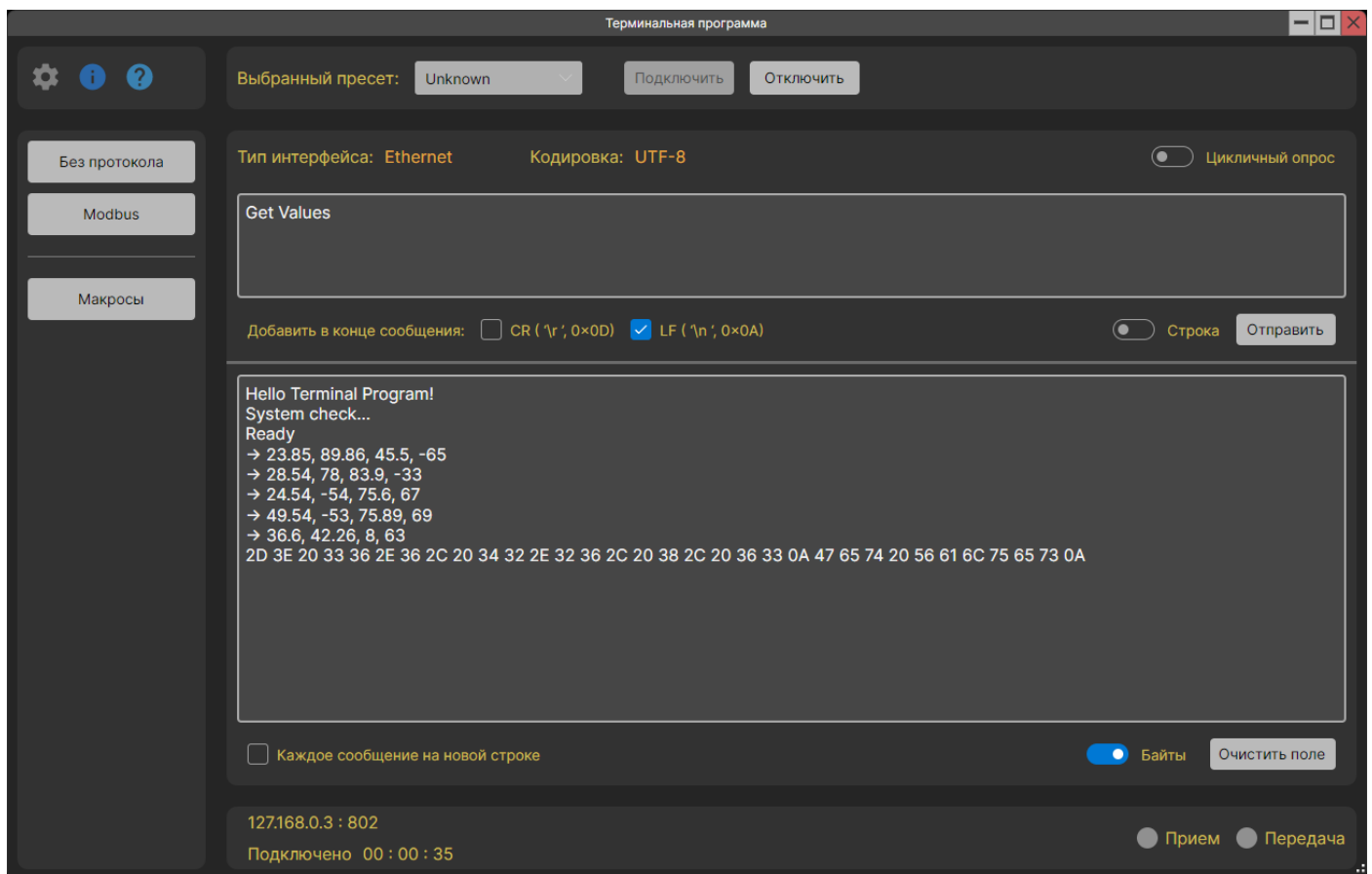
Также есть возможность работы с макросами для обоих режимов.

Без протокола

В поле передачи пользователь пишет данные, которые нужно отправить. В поле приема находятся данные, которые прислал сервер или внешнее устройство. Можно работать как с байтами, так и со строковыми данными в разных кодировках.

Поддерживаются протоколы:

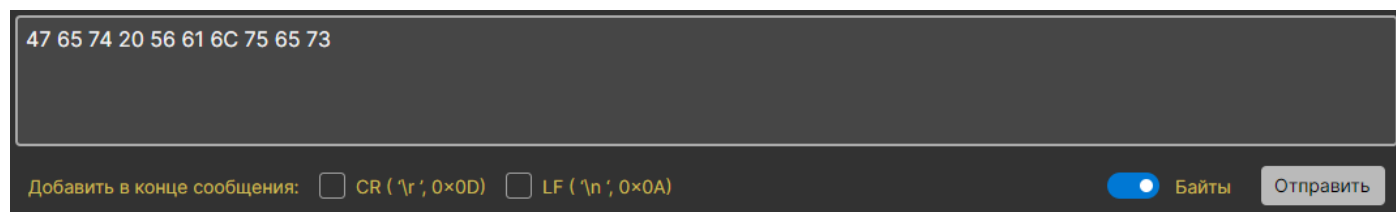
- UART
- TCP



Обычный режим работы

В этом режиме можно отправлять байты или строку подключенному хосту. Отправка происходит один раз по нажатию на кнопку «Отправить». Также можно добавить служебные символы в конце сообщения.

Кодировка строки задается в настройках режима.

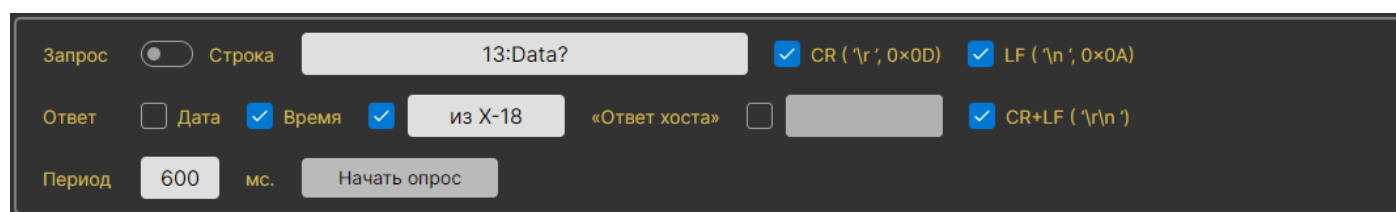


Циклический режим работы

Данный режим позволяет отправлять сообщение хосту автоматически с заданным промежутком времени.

Функционал строки «Запрос» аналогичен нормальному режиму работы. А в строке «Ответ» к самому сообщению можно добавить служебную информацию: дату получения сообщения в формате ДД.ММ.ГГГГ, время в формате ЧЧ:ММ:СС, пользовательские строки в начале и/или конце сообщения и служебные символы.

Кодировка строки такая же, как и у обычного режима.



Modbus

Пользователь может взаимодействовать с выбранными регистрами Modbus, используя соответствующие элементы интерфейса. Для дополнительной расшифровки транзакции существует раздел с представлениями.

Поддерживаются протоколы:

- Modbus TCP
- Modbus RTU
- Modbus ASCII
- Modbus RTU over TCP
- Modbus ASCII over TCP

Терминальная программа

Выбранный пресет: Unknown Подключить Отключить

Режим обмена данными: не определен

ID	Функция	Адрес	Данные
4	0x04 (чтение)	0x0 (0)	0x1234 (4660)
5	0x03 (чтение)	0x11 (17)	Ошибка Modbus. Код: 2
6	0x0F (запись)	0x4 (4) 0x5 (5) 0x6 (6)	0 1 0
7	0x06 (запись)	0x1 (1)	0x4FDA (20442)
8	0x10 (запись)	0x2 (2) 0x3 (3)	0x1242 (4674) 0x6666 (26214)

Формат числа: ☒ hex ☐ dec

Slave ID (hex): 1 ☒ Контрольная сумма

Адрес (hex): 3 Кол-во регистров: 1

0x03 Чтение регистров хранения Прочитать

0x06 Запись одного регистра Записать

4FDA hex

Последний запрос История обмена Бинарное представление Число типа float

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Запрос	00	0A	00	00	00	0B	01	10	00	02	00	02	04	12	42	66	66
Ответ	00	0A	00	00	00	06	01	10	00	02	00	02					

COM5 : 9600 : None : 8 : 1

Отключено ☐ Прием ☐ Передача

Обычный режим работы

В обычном режиме работы можно читать или записывать в регистры Modbus.

Переключатели «Формат числа» меняют формат числа в полях «Slave ID» и «Адрес». В скобках у этих полей отображается выбранный формат.

Формат числа ☒ hex ☐ dec

Slave ID (hex) ☒ Контрольная сумма

Адрес (hex) Кол-во регистров

0x04 Чтение входных регистров

0x06 Запись одного регистра

Данный режим является самым богатым по функционалу, поэтому рассмотрим его подробнее.

Чтение регистров Modbus

Выбираем функцию, начальный адрес, количество регистров и нажимаем кнопку «Прочитать».

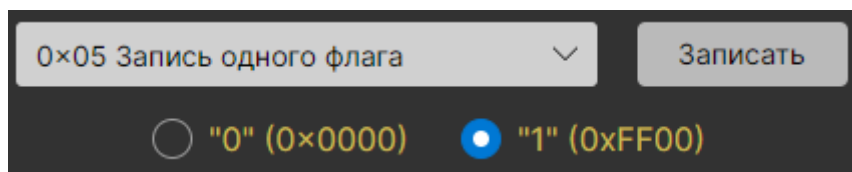
Адрес (hex) Кол-во регистров

0x04 Чтение входных регистров

Запись регистров Modbus

Для каждой функции предусмотрен свой вариант дизайна. Начальным адресом для всех функций является значение из поля «Адрес».

0x05 Запись одного флага



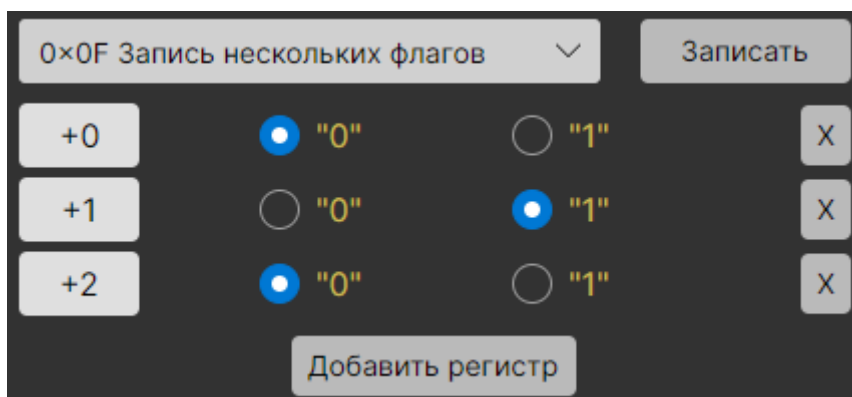
0x05 Запись одного флага

Записать

☐ "0" (0x0000) ☒ "1" (0xFF00)

Согласно документации на протокол, в поле данных должно находиться только одно из двух значений. 0x0000 – это логический ноль, а 0xFF00 – это логическая единица. Поэтому выбираем желаемое значение и нажимаем кнопку «Записать».

0x0F Запись нескольких флагов



0x0F Запись нескольких флагов

Записать

+0	<input checked="" type="radio"/> "0"	<input type="radio"/> "1"	X
+1	<input type="radio"/> "0"	<input checked="" type="radio"/> "1"	X
+2	<input checked="" type="radio"/> "0"	<input type="radio"/> "1"	X

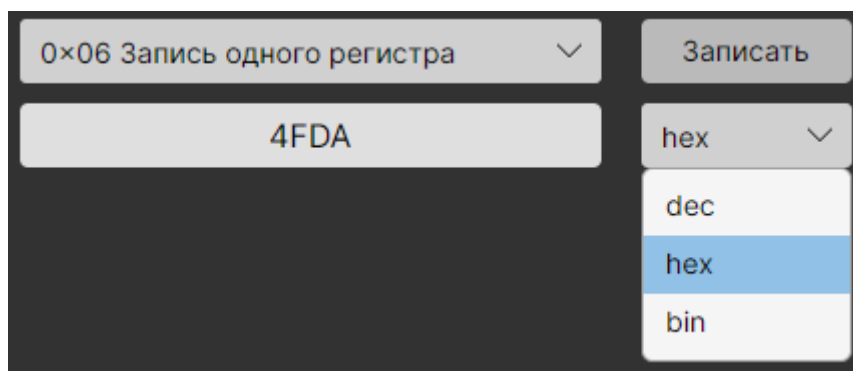
Добавить регистр

С помощью кнопки «Добавить регистр» создаем нужное количество флагов, задаем значение и нажимаем кнопку «Записать».

Слева от значений регистров у нас находятся значения смещения относительно начального адреса.

Справа находятся кнопки удаления для каждого регистра.

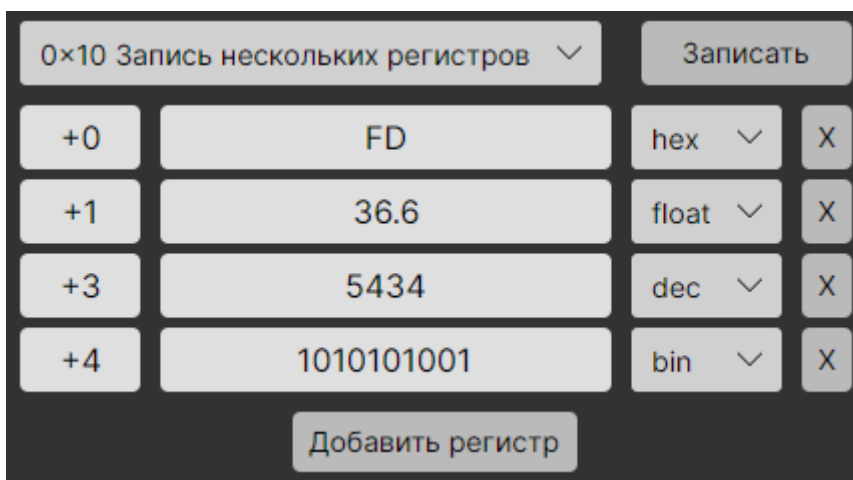
0x06 Запись одного регистра



С помощью этой функции можно записывать в 16-ти разрядные регистры.

Формат записываемого числа выбирается в выпадающем списке справа от поля ввода. При смене формата число автоматически преобразуется.

0x10 Запись нескольких регистров



Offset	Value	Format	Confirm
+0	FD	hex	X
+1	36.6	float	X
+3	5434	dec	X
+4	1010101001	bin	X

Управление тут аналогично функции «0x0F Запись нескольких флагов».

В этой функции появляется возможность записи чисел типа float.

Такие числа занимают 2 слова или же 4 байта. Поэтому у следующего регистра смещение уже не +1, а +2 адреса.

Иногда бывает, что устройство может использовать нетипичный формат для расшифровки чисел типа float. И чтобы подстроиться под конкретное устройство в настройках можно выбрать нужный формат записи.

Настройки

Пресет: Unknown

Подключение

Без протокола

Modbus

Приложение

Таймаут записи

300

мс.

Таймаут чтения

300

мс.

Формат записи числа типа float

☐ AB CD (12 34 56 78)

☒ BA DC (34 12 78 56)

☐ CD AB (56 78 12 34)

☐ DC BA (78 56 34 12)

Страница с настройками режима Modbus.

Циклический режим работы

В данном режиме можно только читать регистры с заданным периодом.

Формат числа

☒ hex

☐ dec

Slave ID

3

(hex)

☒ Контрольная сумма

Функция

0x04 Чтение входных регистров

▼

Период

1000

мс.

Начальный адрес

21

(hex)

Кол-во регистров

2

Начать опрос

Представления

Просто значения регистров можно посмотреть в табличном представлении. Но, к сожалению, в этих числах не всегда есть смысл. И иногда их требуется "расшифровать". Поэтому для интерпретации данных в терминале предусмотрена область с представлениями.

Всего есть 4 типа представлений:

Последний запросИстория обменаБинарное представлениеЧисло типа float

Запрос

1234567891011121314151617

000000000006010400010004

00000000000B0104084567890100000000

Ответ

Последний запрос

	Последний запрос	История обмена	Бинарное представление	Число типа float
19:29:41.255	→	00 04 00 00 00 06 01 04 00 01 00 03		
19:29:41.256	←	00 04 00 00 00 09 01 04 06 45 67 89 01 00 00		
19:29:42.460	→	00 05 00 00 00 06 01 04 00 01 00 04		
19:29:42.461	←	00 05 00 00 00 0B 01 04 08 45 67 89 01 00 00 00 00		
19:29:46.063	→	00 06 00 00 00 06 01 03 00 01 00 03		
19:29:46.065	←	00 06 00 00 00 09 01 03 06 00 00 00 00 00 00		

История обмена

Последний запрос

История обмена

Бинарное представление

Число типа float

Адрес 0x00

Значение

0

0

0

0

0

1

0

1

0

1

1

0

0

0

1

1

Адрес 0x01

Значение

0

0

0

0

0

0

0

0

1

1

1

1

0

1

0

1

Бинарное представление

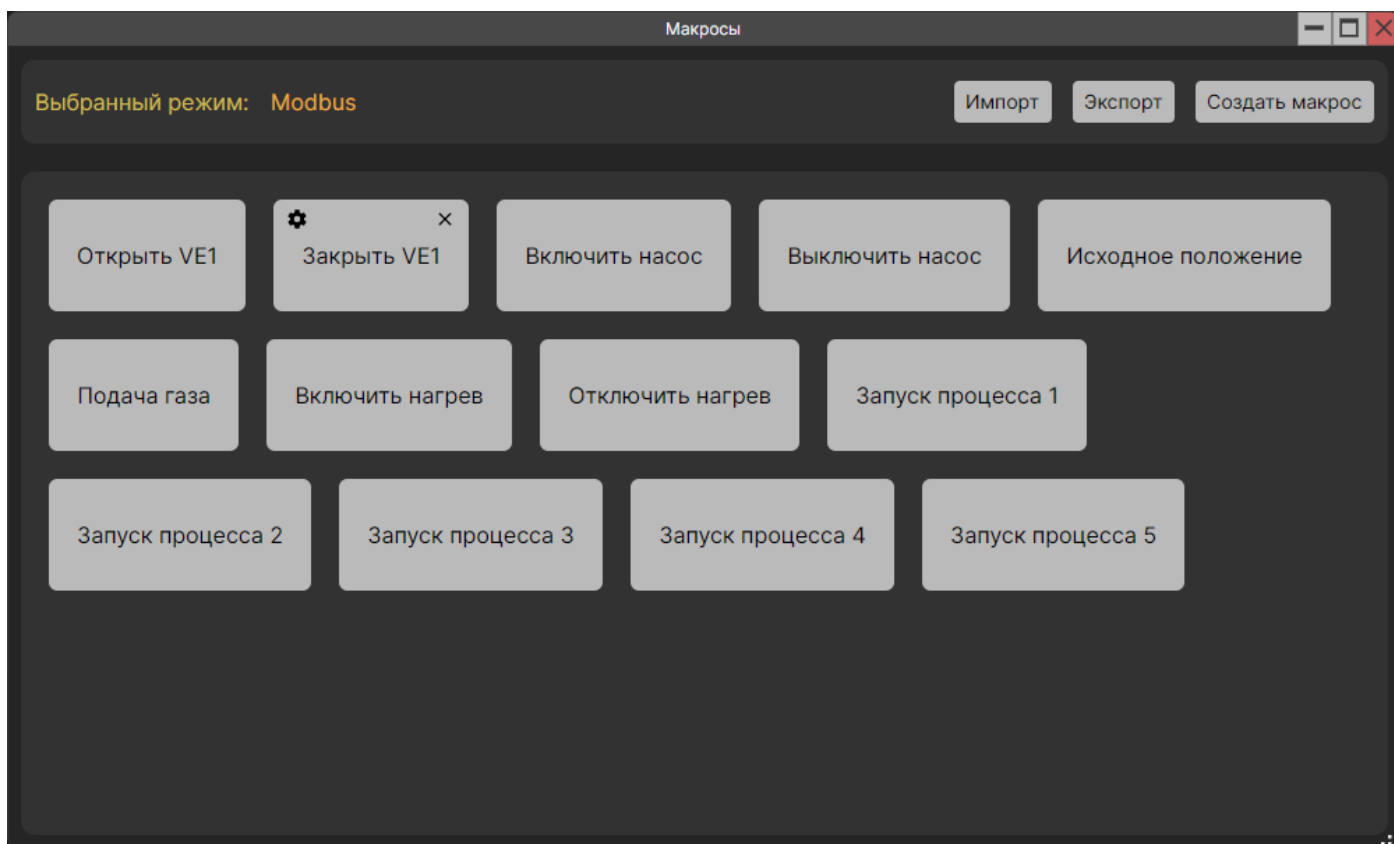
Последний запрос	История обмена	Бинарное представление	Число типа float	
<div>0×0000</div> <div>AB CD 6,244814E-39</div> <div>BA DC 512</div> <div>CD AB 9,5E-44</div> <div>DC BA 2,4394E-41</div>	<div>0×0002</div> <div>AB CD -6,360643E+34</div> <div>BA DC 1997</div> <div>CD AB 1,4783099E-38</div> <div>DC BA -1,0864843E-19</div>	<div>0×0004</div> <div>AB CD 1,83765E-40</div> <div>BA DC 9,423183E-38</div> <div>CD AB 6,152981E-39</div> <div>DC BA 128,00781</div>	<div>0×0006</div> <div>AB CD 6,134182E-28</div> <div>BA DC 36,6</div> <div>CD AB 2,7162027E+23</div> <div>DC BA 2,7184077E+23</div>	<div>0×0008</div> <div>AB CD 2,4934507E-27</div> <div>BA DC 2355,0344</div> <div>CD AB -5,425738E-31</div> <div>DC BA 1,0278732E-09</div>

Представление числа типа float

Макросы

Этот режим позволяет работать с макросами для режимов "Без протокола" и "Modbus". При наведении курсора на макрос появляются кнопки редактирования и удаления.

Также предусмотрена возможность импорта и экспорта файла макросов для каждого режима. Это удобно использовать, когда необходимо перенести макросы на несколько ПК.



Для каждого режима есть своя форма создания/редактирования макроса.

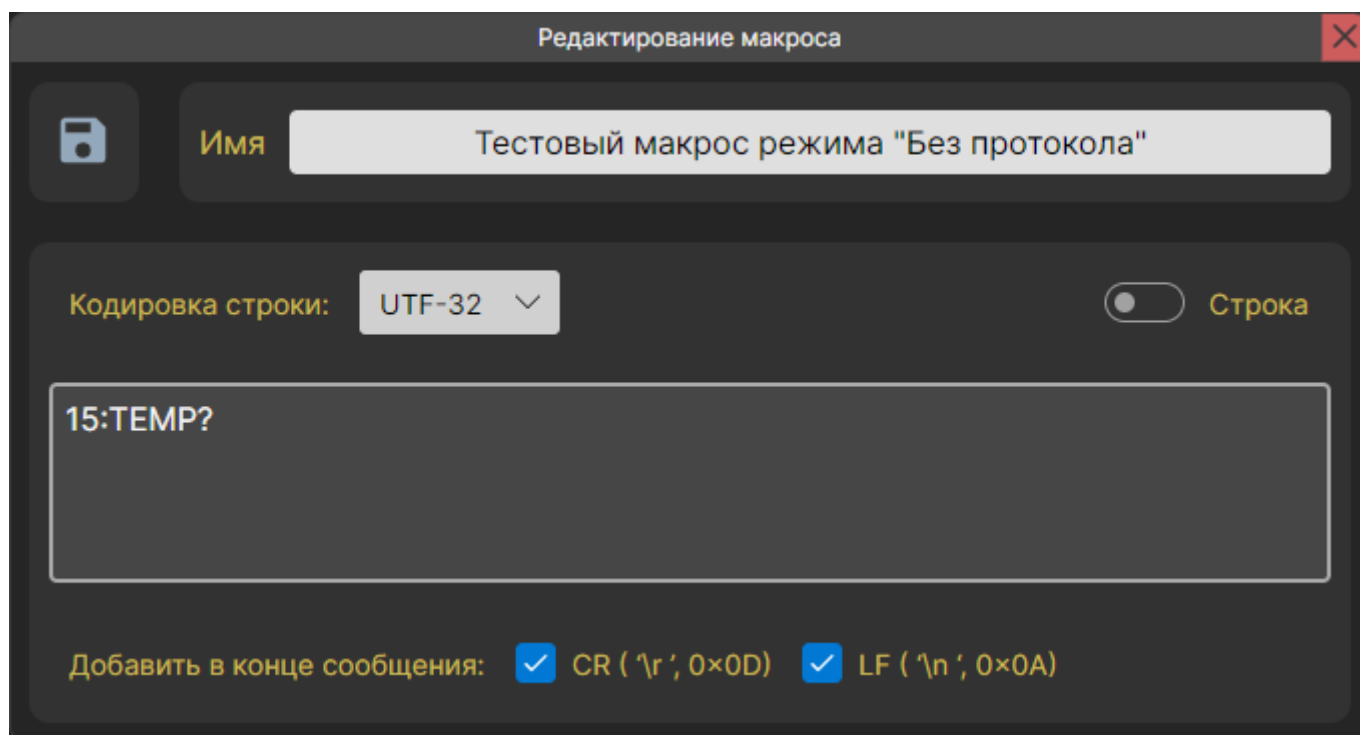
Без протокола

Взаимодействие в данном окне аналогично обычному режиму «Без протокола».

Важно!

Кодировка строки в макросе автономна.

Она не зависит от общей кодировки, заданной в настройках приложения.



The screenshot shows a window titled "Редактирование макроса" (Editing macro) with a close button in the top right corner. The window contains the following elements:

- A save icon (floppy disk) on the left.
- A label "Имя" (Name) followed by a text input field containing "Тестовый макрос режима "Без протокола"" (Test macro of "No protocol" mode).
- A label "Кодировка строки:" (String encoding:) followed by a dropdown menu showing "UTF-32".
- A toggle switch on the right, currently turned off, with the label "Строка" (String).
- A large text area containing the text "15:TEMP?".
- A label "Добавить в конце сообщения:" (Add to the end of the message:) followed by two checked checkboxes: "CR ('\r', 0x0D)" and "LF ('\n', 0x0A)".

Modbus

Управление в этом окне также схоже с обычным режимом «Modbus».

Важно!

Формат числа типа float в макросе автономен.

Он не зависит от формата, выбранного в настройках приложения.

Редактирование макроса

Имя: Тестовый макрос режима Modbus

Формат числа: ☒ hex ☐ dec

Slave ID (hex): 1 ☒ Контрольная сумма

Начальный адрес (hex): 3

☐ Чтение ☒ Запись

0x10 Запись нескольких регистров

Формат числа типа float: BA_DC

+0 36.6 float X

Добавить регистр

Статьи на Хабр

[Кроссплатформенный терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 3](#)

[Кроссплатформенный терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 2](#)

[Терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 1](#)

Ссылка на скачивание

[Все версии тут.](#)

История версий

3.1.0

Изменения

- Добавлена поддержка Modbus RTU over TCP.
- Добавлена поддержка Modbus ASCII over TCP.
- Добавлена возможность работы с байтами в режиме "Без протокола".
- Добавлены макросы.
- Добавлено руководство пользователя.
- Исправление ошибок, мелкие улучшения и рефакторинг.

3.0.0

Изменения

- Проект перенесен с WPF на AvaloniaUI.
- Изменен дизайн.
- Добавлен Modbus сканер.
- Modbus: для каждой функции записи сделан свой вариант дизайна.
- Modbus: добавлено ведение истории обмена.
- Modbus: добавлена возможность работы с бинарными данными.
- Modbus: добавлена возможность работы с данными типа float.
- Исправлены ошибки версии 2.7.0.

2.7.0

Изменения

- Первая публичная версия