

Руководство пользователя

CoreBus (версия 3.3.3)

[Поддержать проект](#)

andrey.abdulkayumov@gmail.com

Оглавление

Краткое описание.....	3
Без протокола	5
Одиночная отправка	6
Цикличная отправка	6
Отправка файлов	7
Modbus	8
Обычный режим работы.....	9
Чтение регистров Modbus	9
Запись регистров Modbus.....	10
0x05 Запись одного флага	10
0x0F Запись нескольких флагов	10
0x06 Запись одного регистра	11
0x10 Запись нескольких регистров.....	11
Циклический режим работы	13
Представления.....	14
Modbus сканер.....	15
Макросы.....	17
Редактирование макроса	18
Статьи на Хабр	22
Ссылка на скачивание	22
История версий	23
3.3.3.....	23
3.3.2.....	23
3.3.1.....	23
3.3.0.....	24
3.2.1.....	24
3.2.0.....	24
3.1.0.....	25
3.0.0.....	25
2.7.0.....	25

Краткое описание

CoreBus (старое название «Терминальная программа») — кроссплатформенный терминал для работы с COM-портами и TCP-сокетами с поддержкой протоколов Modbus TCP / RTU / ASCII.

Основные возможности приложения:

1. Два режима работы: "Без протокола" и "Modbus".
2. "Без протокола":
 - Работа с данными в строковом или байтовом формате.
 - Поддержка разных кодировок.
 - Три режима отправки: одиночная, циклическая, отправка файла.
3. "Modbus":
 - Поддержка различных вариаций протокола Modbus: TCP, RTU, ASCII и RTU / ASCII over TCP.
 - Удобная работа с функциями записи.
 - Возможность работы с числами типа float.
 - Возможность работы с бинарными данными.
 - Циклический опрос.
 - Modbus сканер, который осуществляет поиск устройств на линии связи.
4. Макросы:
 - Отдельные макросы для каждого режима работы.
 - Макрос состоит из неограниченного количества команд (действий).
 - Для Modbus макросов предусмотрена возможность выставления общего Slave ID для всего макроса.
 - Импорт и экспорт макросов.
5. Темная и светлая темы приложения.
6. Пресеты с пользовательскими настройками.
7. Руководство пользователя.
8. Кроссплатформенность: Windows, Linux.

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

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 расширенными 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.

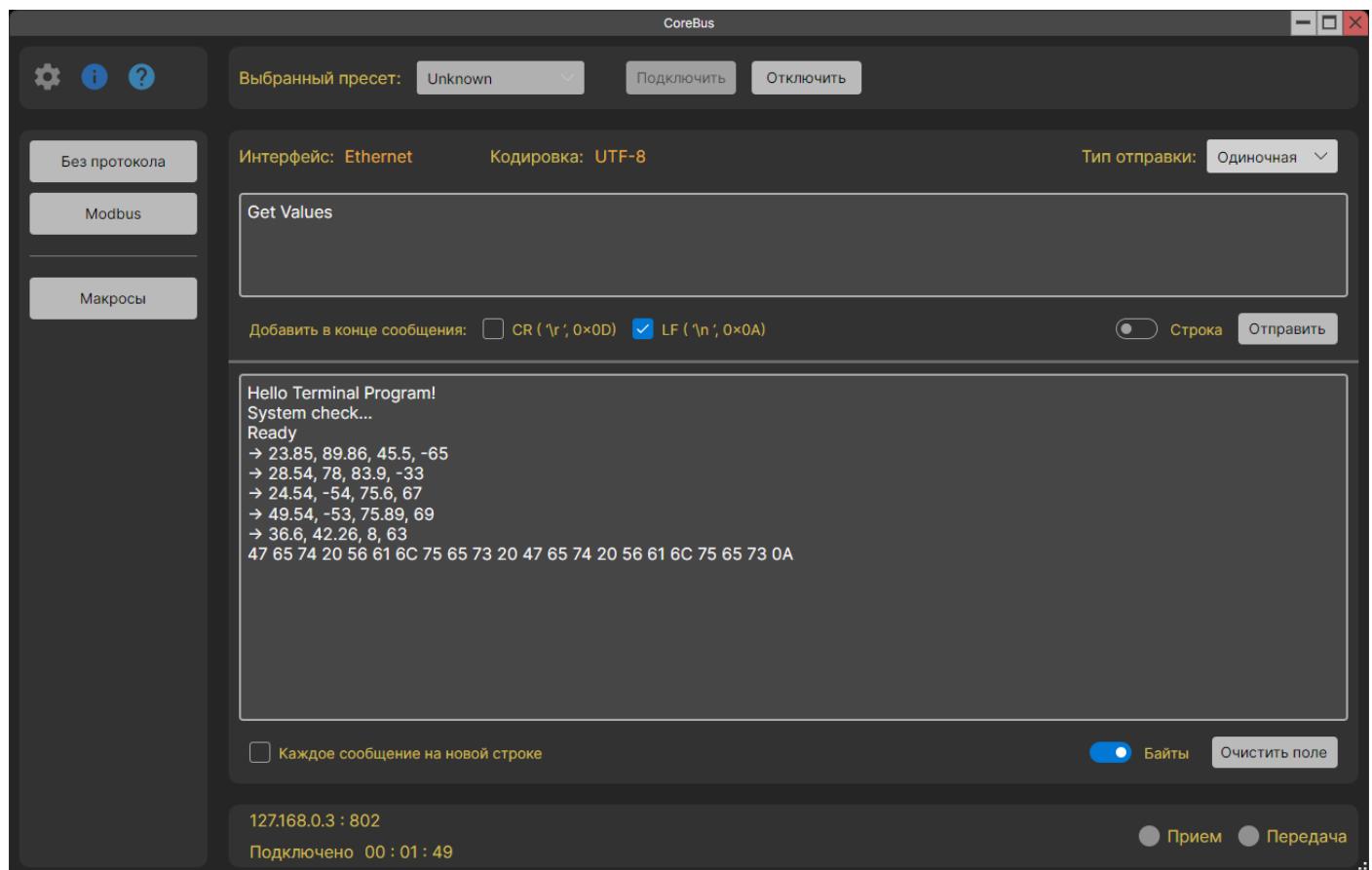
Поддерживаются протоколы:

- UART
- TCP

Есть три режима отправки:

- Одиночная
- Циклическая
- Файлы

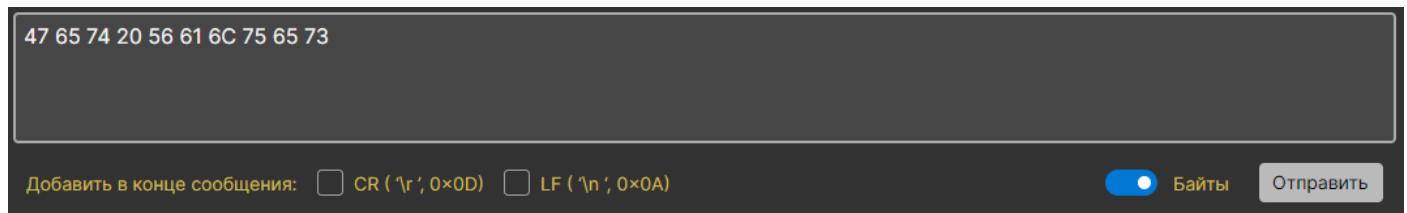
Тип отправки выбирается в выпадающем списке в правом верхнем углу.



Одиночная отправка

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.

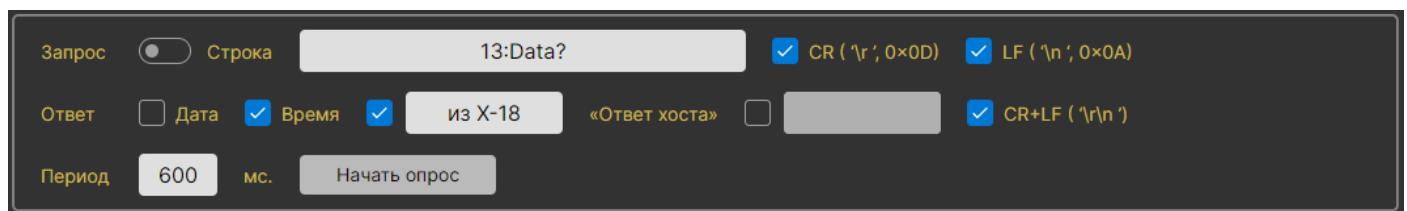


Циклическая отправка

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.

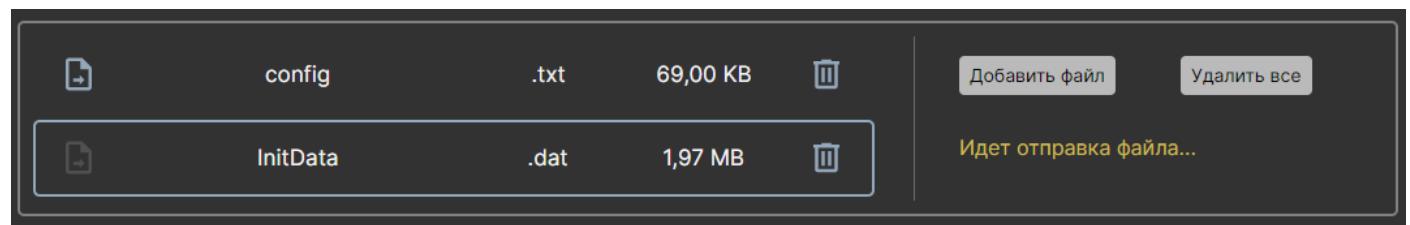


Отправка файлов

Этот режим позволяет отправлять файлы. Заданная кодировка в этом режиме не используется.

Файлы можно использовать многократно. Они добавляются в специальную папку, поэтому их не нужно добавлять каждый раз после запуска приложения.

Во время отправки файла появляется надпись «Идет отправка файла...». После она пропадает.

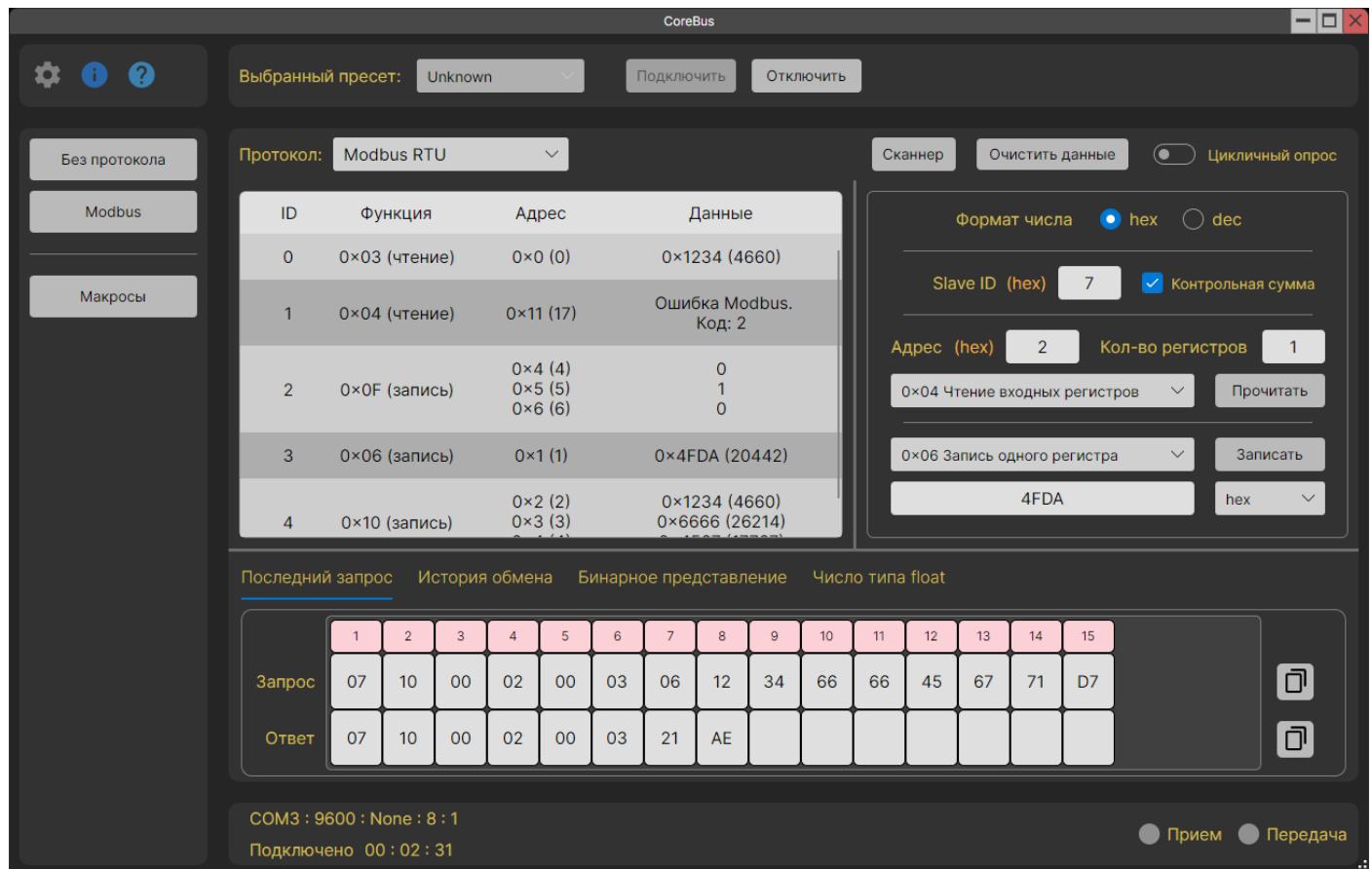


Modbus

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

Поддерживаются протоколы:

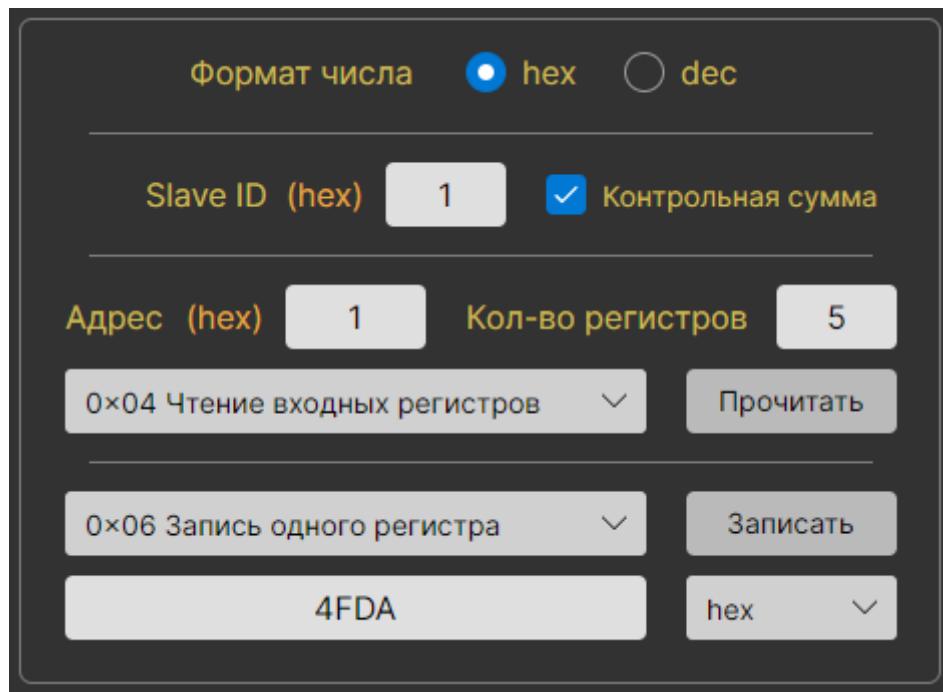
- Modbus TCP
- Modbus RTU
- Modbus ASCII
- Modbus RTU over TCP
- Modbus ASCII over TCP



Обычный режим работы

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

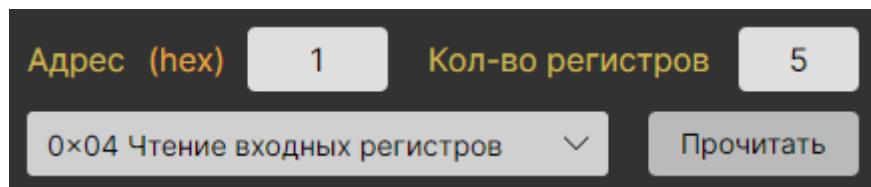
Переключатели «Формат числа» меняют формат числа в полях «Slave ID» и «Адрес». В скобочках у этих полей отображается выбранный формат.



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

Чтение регистров Modbus

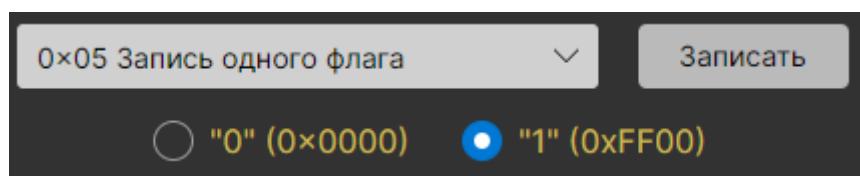
Выбираем функцию, начальный адрес, количество регистров и нажимаем кнопку «Прочитать».



Запись регистров Modbus

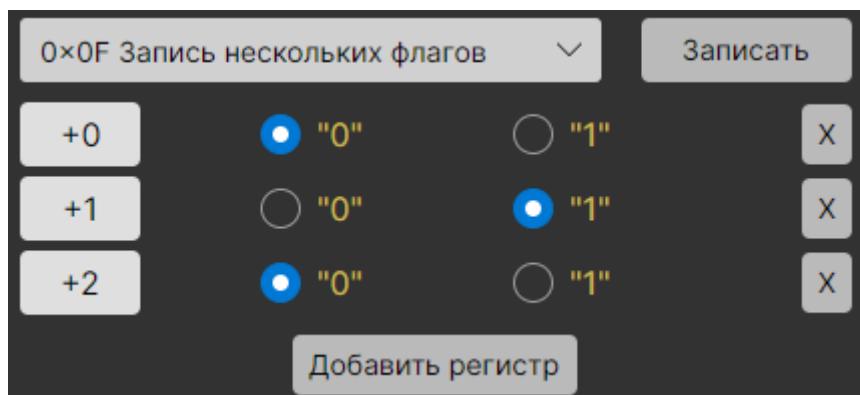
Для каждой функции предусмотрен свой вариант дизайна. Начальным адресом для всех функций является значение из поля «Адрес».

0x05 Запись одного флага



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



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

0x06 Запись одного регистра

Записать

4FDA

hex

dec

hex

bin

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

0x10 Запись нескольких регистров

Записать

+0	FD	hex	X
+1	36.6	float	X
+3	5434	dec	X
+4	1010101001	bin	X

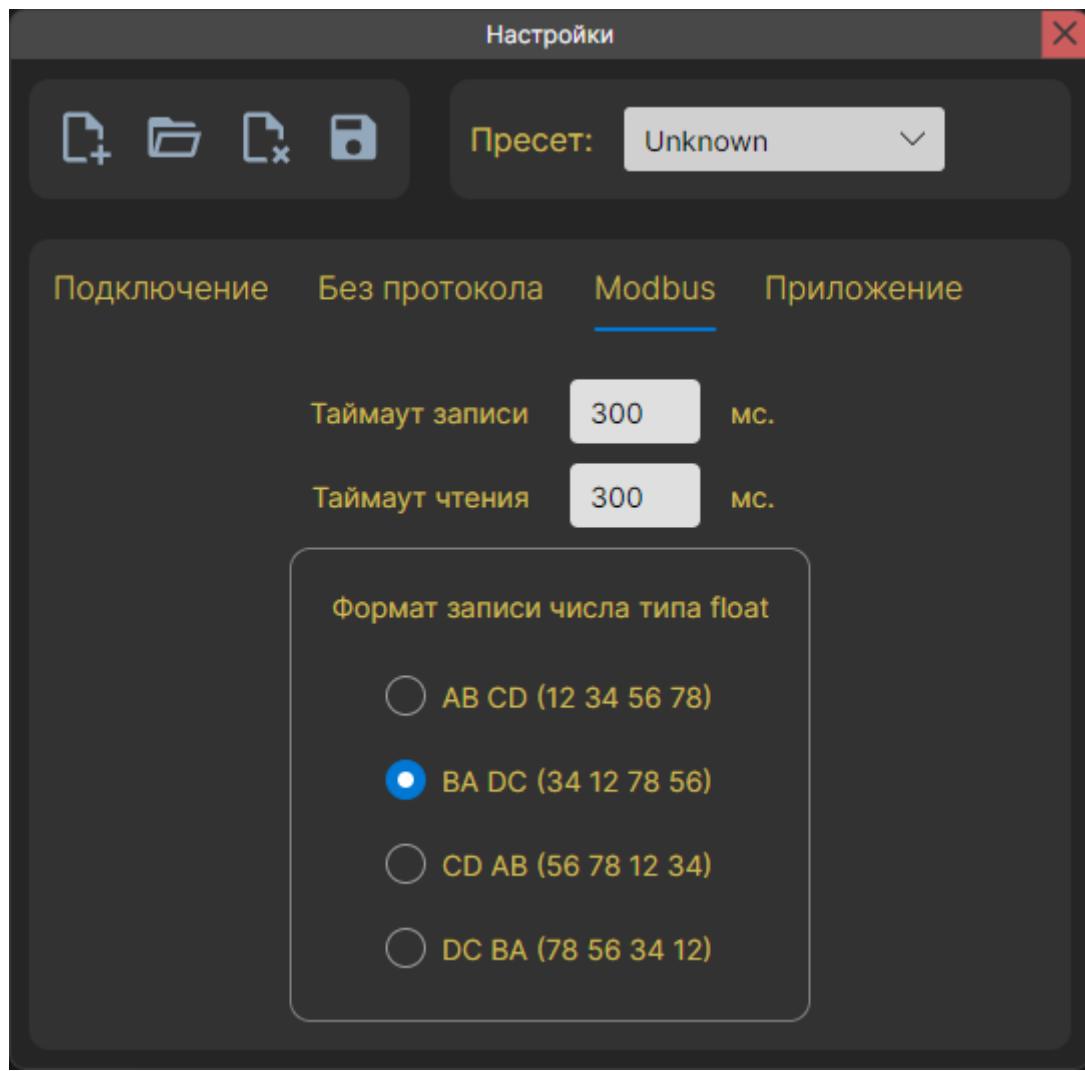
Добавить регистр

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.



Страница с настройками режима Modbus.

Циклический режим работы

В данном режиме можно только читать регистры с заданным периодом.

Формат числа hex dec

Slave ID (hex) Контрольная сумма

Функция

Период мс.

Начальный адрес (hex)

Кол-во регистров

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 запрос История обмена Бинарное представление Число типа float																	
Запрос	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Запрос	00	00	00	00	00	06	01	04	00	01	00	04					
Ответ	00	00	00	00	00	0B	01	04	08	45	67	89	01	00	00	00	00

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																	
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](#)

0x0000	0x0002	0x0004	0x0006	0x0008
AB CD 6,244814E-39	AB CD -6,380643E+34	AB CD 1,83765E-40	AB CD 6,134182E-28	AB CD 2,4934507E-27
BA DC 512	BA DC 1997	BA DC 9,423183E-38	BA DC 36,6	BA DC 2355,0344
CD AB 9,5E-44	CD AB 1,4783099E-38	CD AB 6,152981E-39	CD AB 2,7162027E+23	CD AB -5,425738E-31
DC BA 2,4394E-41	DC BA -1,0864843E-19	DC BA 128,00781	DC BA 2,7184077E+23	DC BA 1,0278732E-09

Представление числа типа float

Modbus сканер

Modbus сканер служит для поиска подчиненных устройств на линии связи. Эта функция доступна только при подключении по последовательному порту, т.к. при подключении по TCP/IP в ней смысла нет.

Поиск может происходить по протоколу Modbus RTU или Modbus ASCII. Тип протокола выбирается в выпадающем списке.

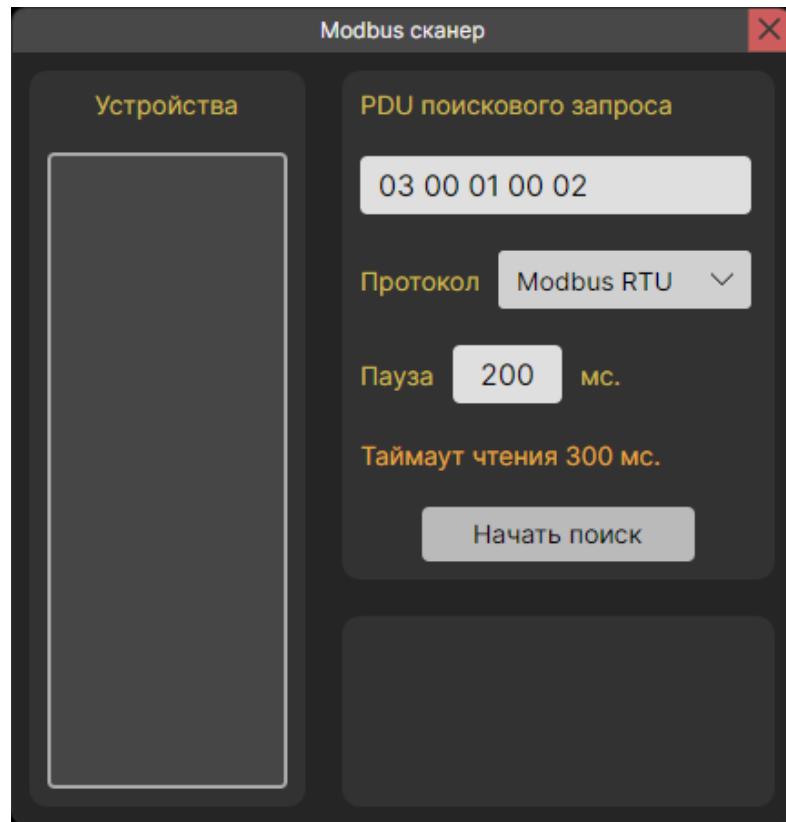
В поле «Устройства» будут отображаться адреса найденных устройств. Если после окончания поиска это поле пустое, то это означает, что ни одно устройство не ответило во всем диапазоне допустимых адресов (1 – 255). Широковещательный адрес 0 не учитывается, т.к. согласно документации устройства не должны на него отвечать.

В правом верхнем поле указан PDU, с которым проводится опрос.

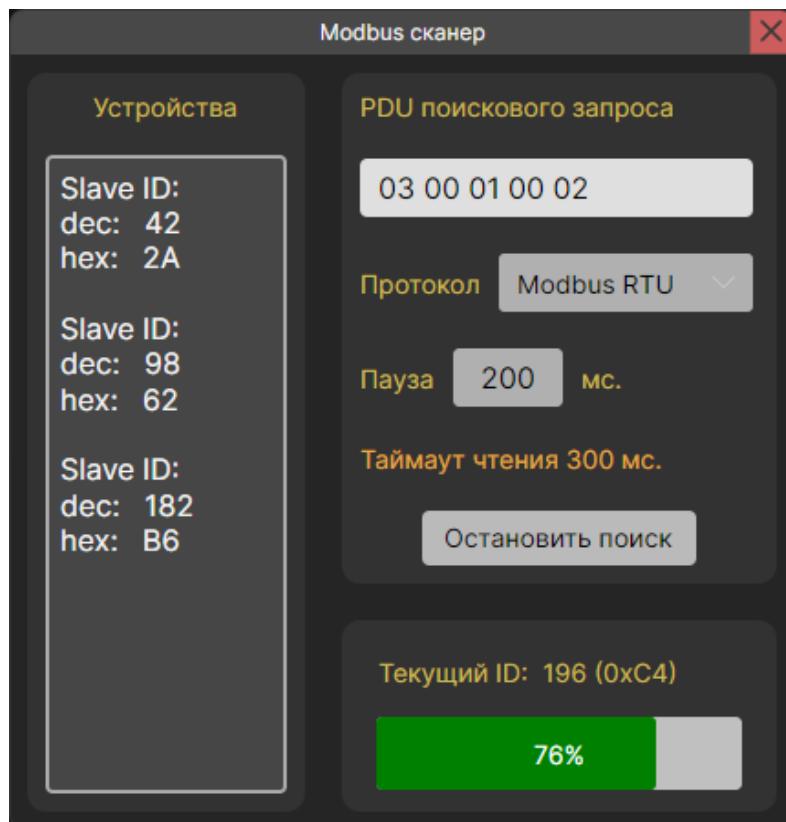
Важно понять, как работает пауза между отправками сообщений. Эта пауза состоит из двух составляющих – пользовательский таймаут, который указывается в поле «Пауза», и таймаут чтения, в течении которого приложение ждет ответа от устройства. Таймаут чтения задается в настройках.

Важно!!!

Возможна ситуация, когда после опроса всех адресов поле «Устройства» пустое. При этом достоверно известно, что на линии связи находятся исправно работающие устройства. Это объясняется тем, что подчиненное устройство по каким-либо причинам не успело обработать сообщение и отправить ответ. В таких случаях рекомендуется увеличить пользовательский таймаут в поле «Пауза».



Неактивное состояние



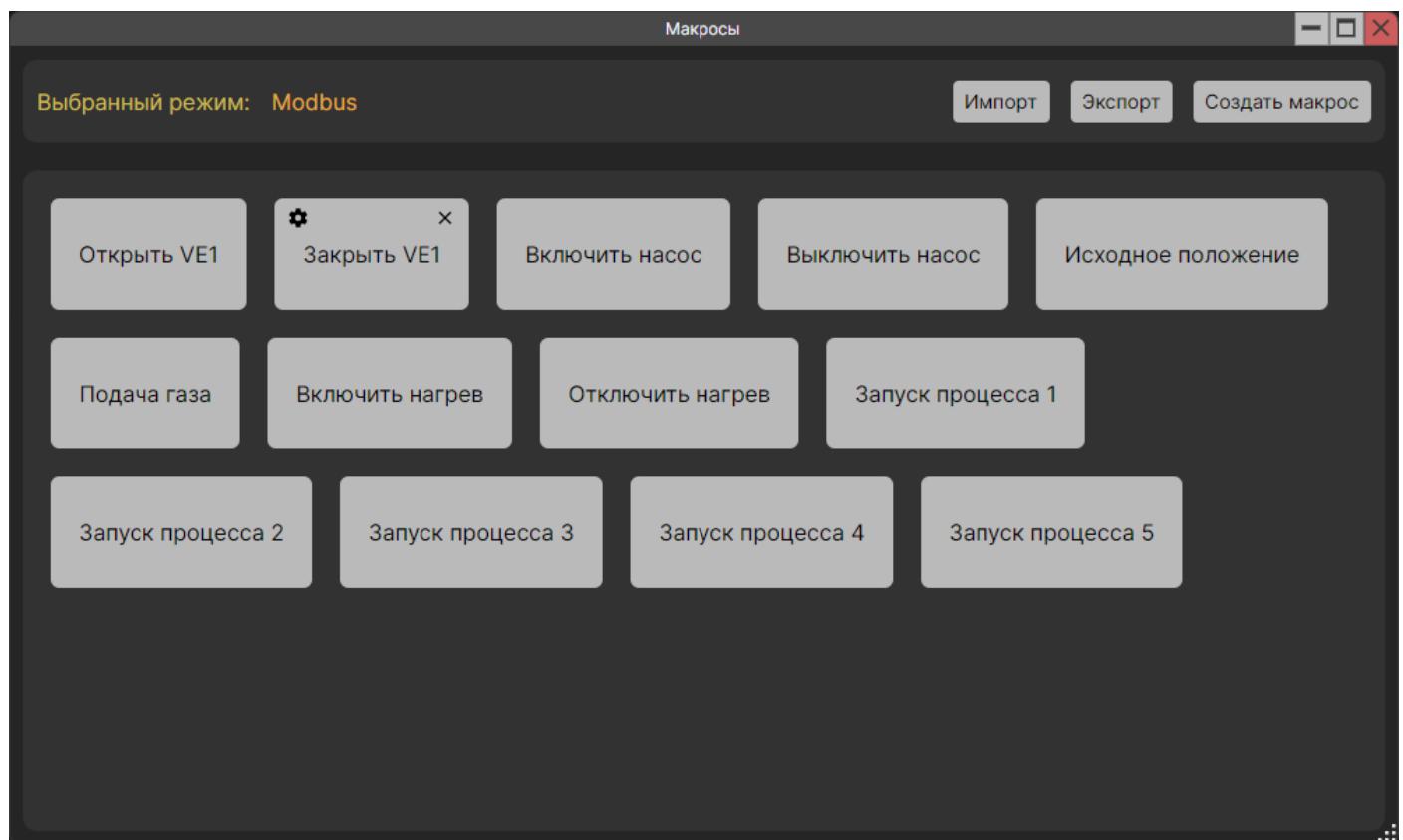
Состояние поиска

Макросы

В приложении предусмотрена работа с макросами. Они доступны для всех режимов. Макросы поддерживают отправку сразу нескольких сообщений за раз.

Все макросы представлены на рабочем поле в виде кнопок с соответствующими названиями. При наведении курсора на любой из макросов 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.



Редактирование макроса

Макрос разделен на команды. Каждая команда – это отправка одного сообщения.

При открытии окна редактирования макроса первоначальное окно с рабочим полем скрывается. А после закрытия окна редактирования, оно появляется вновь. Это сделано для более независимой работы с окнами.

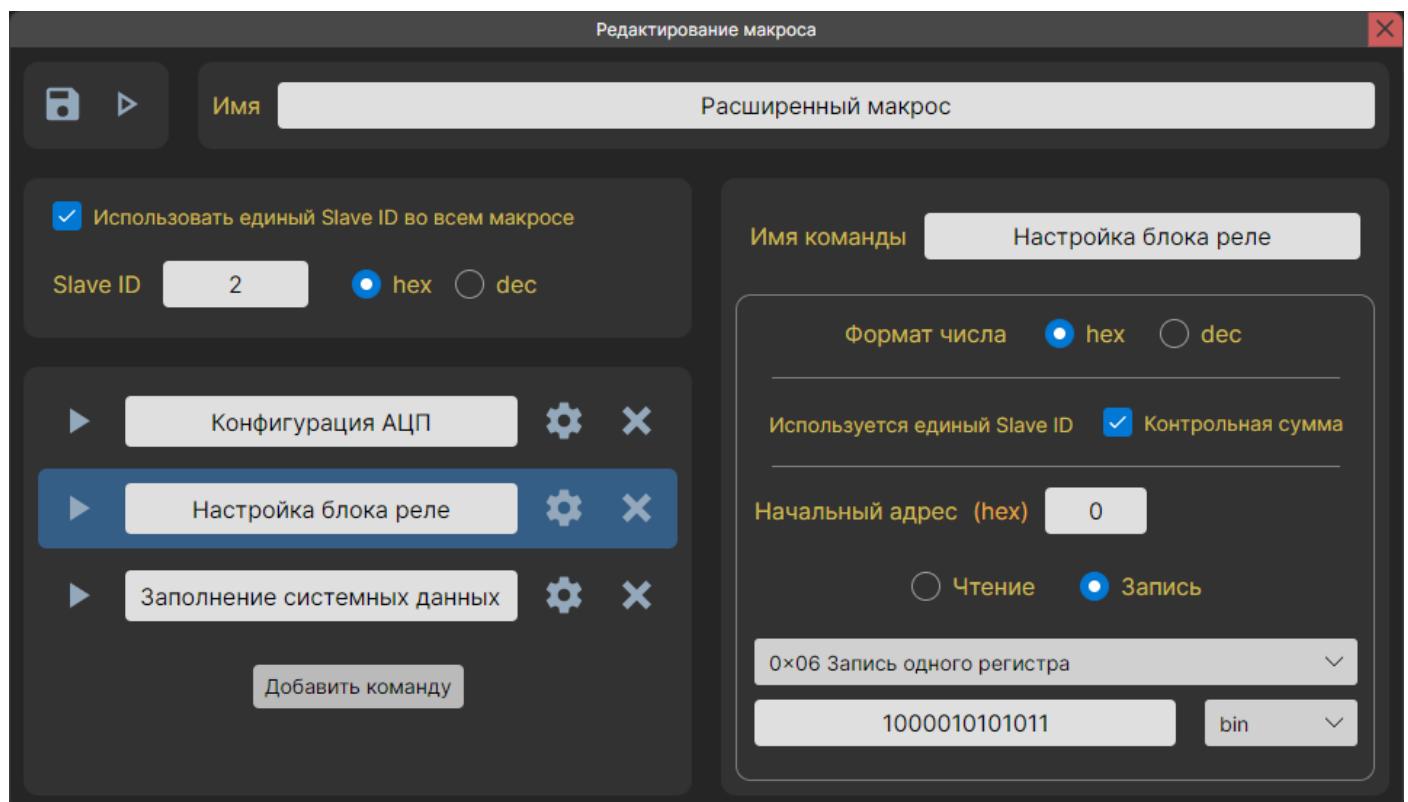
В режиме редактирования есть возможность отправки отдельных команд или всего макроса полностью. Для этого предусмотрены соответствующие кнопки в шапке макроса и у каждой команды в списочной форме.

Рабочее поле разделено на четыре части.

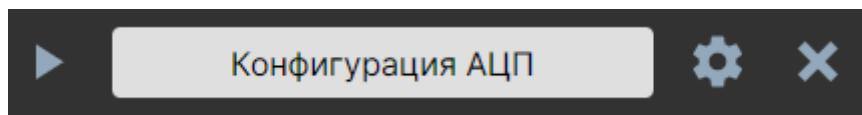
Сверху находится поле редактирования имени макроса, а левее от него кнопки сохранения и запуска макроса. Перед сохранением или запуском происходит проверка. В случае наличия ошибок, появляется сообщение с перечислением всех найденных ошибок.

Нижняя часть разделена пополам. Слева находится список со всеми командами макроса и кнопка добавления команды, а справа форма редактирования выбранной команды. Редактируемая команда подсвечивается в списке.

Для режима «Modbus» предусмотрена возможность выставить общий Slave ID для всего макроса.



Рассмотрим элементы каждой команды из списка.



Слева направо.

- **Кнопка запуска команды.**

Если команда не содержит ошибок и хост подключен, то сообщение будет отправлено. В противном случае появится сообщение с описанием ошибки.

- **Поле с названием команды.**

Оно доступно только для выделения и копирования. Изменить название команды можно только в форме редактирования.

- **Кнопка открытия/закрытия формы редактирования команды.**

Редактируемая команда подсвечивается в списке. Также можно не закрывая форму редактирования текущей команды, нажать на эту же кнопку у другой команды и редактировать уже её.

- **Кнопка удаления команды.**

Просто так удалить нельзя, нужно подтвердить удаление в диалоговом окне.

Для каждого режима работы предусмотрена своя форма редактирования команды. Рассмотрим их по отдельности.

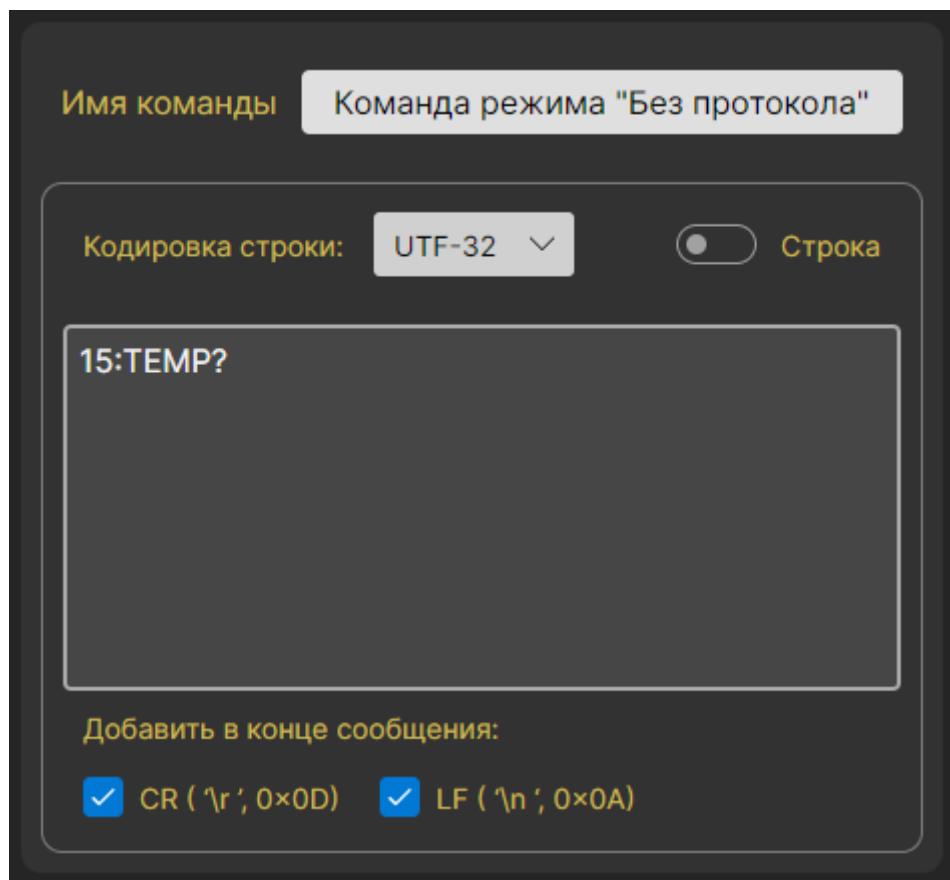
Команда режима «Без протокола»

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.



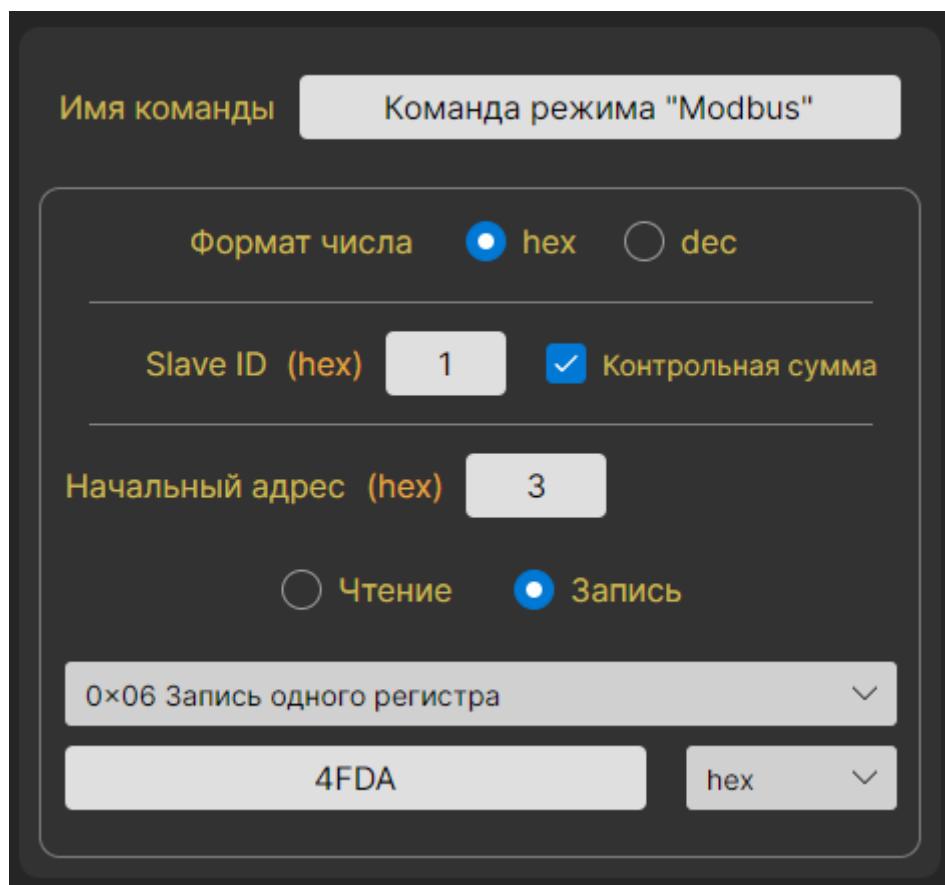
Команда режима «Modbus»

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.



Статьи на Хабр

[CoreBus: Часть 5 — попытка использования Native AOT](#)

[Кроссплатформенный терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 4](#)

[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

3.3.3

Новый функционал:

- Теперь при работе с макросами используется только одно окно (рабочее поле или редактирование). Неиспользуемое окно скрывается.
- Рефакторинг Unit тестов.

3.3.2

Новый функционал:

- Modbus: теперь если сработал таймаут, то выводится содержимое запроса.
- Расширен функционал окна "О приложении".

Исправление ошибок:

- Редкое зависание при записи/чтении Modbus.
- Продолжение работы циклического опроса в случае появления ошибки.
- Неправильное отображение количества записанных регистров Modbus после отправки команды макроса.
- При отправке отдельных команд макроса не учитывается единый Slave ID.
- При использовании единого Slave ID в макросе учитываются ошибки от полей Slave ID команд.

3.3.1

Изменения:

- В Modbus сканер добавлен выбор протокола (Modbus RTU или Modbus ASCII), по которому происходит поиск устройств.
- Сокращены некоторые имена в UI.

3.3.0

Изменения:

- Добавлена возможность использовать единый Slave ID для Modbus макроса.
- Теперь все ошибки, появившиеся при работе макроса, собираются в единое сообщение, а не показываются по отдельности.
- Добавлены улучшения для более удобной работы с окном макросов.
- Исправлен баг с получением некорректного пути при выборе папки или файла.
- Оптимизации.
- Рефакторинг.

3.2.1

Изменения:

- В MessageBox появились иконки, которые зависят от типа сообщения.
- В MessageBox стал доступен "Отчет об ошибке", который появляется у сообщений с типом ошибки. Отчет можно посмотреть в отдельном окне, копировать в буфер обмена или в текстовый файл.

3.2.0

Изменения:

- Расширена работа с макросами. Макрос разделен на команды. Теперь в одном макросе можно отправить несколько сообщений за раз.
- В проект внедрен DI.
- Рефакторинг. Уменьшена связность между компонентами.
- Исправлены мелкие ошибки.

[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