**Руководство пользователя**

**Терминальная программа (версия 3.2.1)**

[Поддержать проект](https://andreyabdulkayumov.github.io/TerminalProgram_Website/donate.html)

andrey.abdulkayumov@gmail.com

Оглавление

[Краткое описание 3](#_Toc195474109)

[Без протокола 4](#_Toc195474110)

[Обычный режим работы 5](#_Toc195474111)

[Цикличный режим работы 5](#_Toc195474112)

[Modbus 6](#_Toc195474113)

[Обычный режим работы 7](#_Toc195474114)

[Чтение регистров Modbus 7](#_Toc195474115)

[Запись регистров Modbus 8](#_Toc195474116)

[0x05 Запись одного флага 8](#_Toc195474117)

[0x0F Запись нескольких флагов 8](#_Toc195474118)

[0x06 Запись одного регистра 9](#_Toc195474119)

[0x10 Запись нескольких регистров 9](#_Toc195474120)

[Цикличный режим работы 11](#_Toc195474121)

[Представления 12](#_Toc195474122)

[Modbus сканер 13](#_Toc195474123)

[Макросы 15](#_Toc195474124)

[Редактирование макроса 16](#_Toc195474125)

[Статьи на Хабр 20](#_Toc195474126)

[Ссылка на скачивание 20](#_Toc195474127)

[История версий 21](#_Toc195474128)

[3.2.1 21](#_Toc195474129)

[3.2.0 21](#_Toc195474130)

[3.1.0 21](#_Toc195474131)

[3.0.0 22](#_Toc195474132)

[2.7.0 22](#_Toc195474133)

# 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 "Цикличный опрос". Между ними можно переключаться во время работы. Данные на вкладках не теряются при переключении.

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

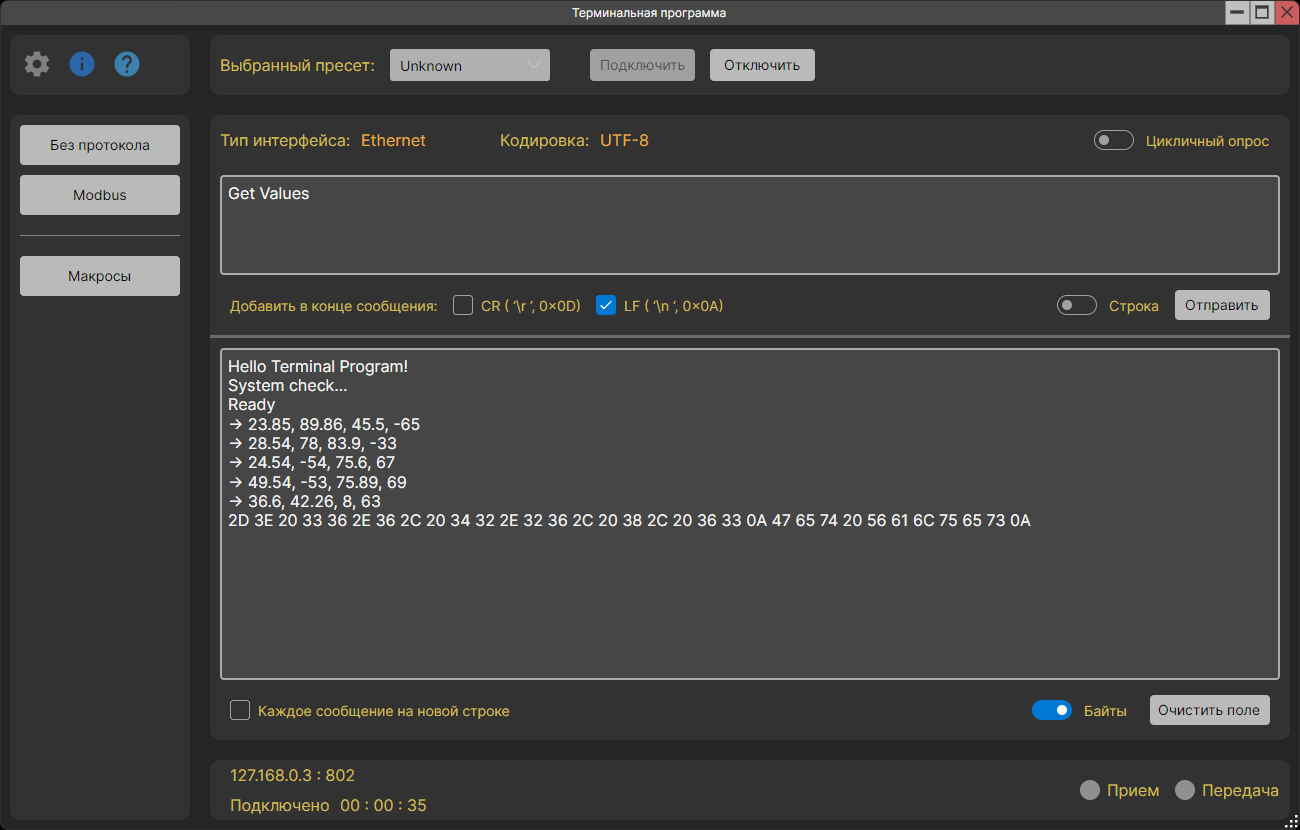
Также есть возможность работы с расширенными макросами для обоих режимов.

# 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



## 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.



## Цикличный 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.

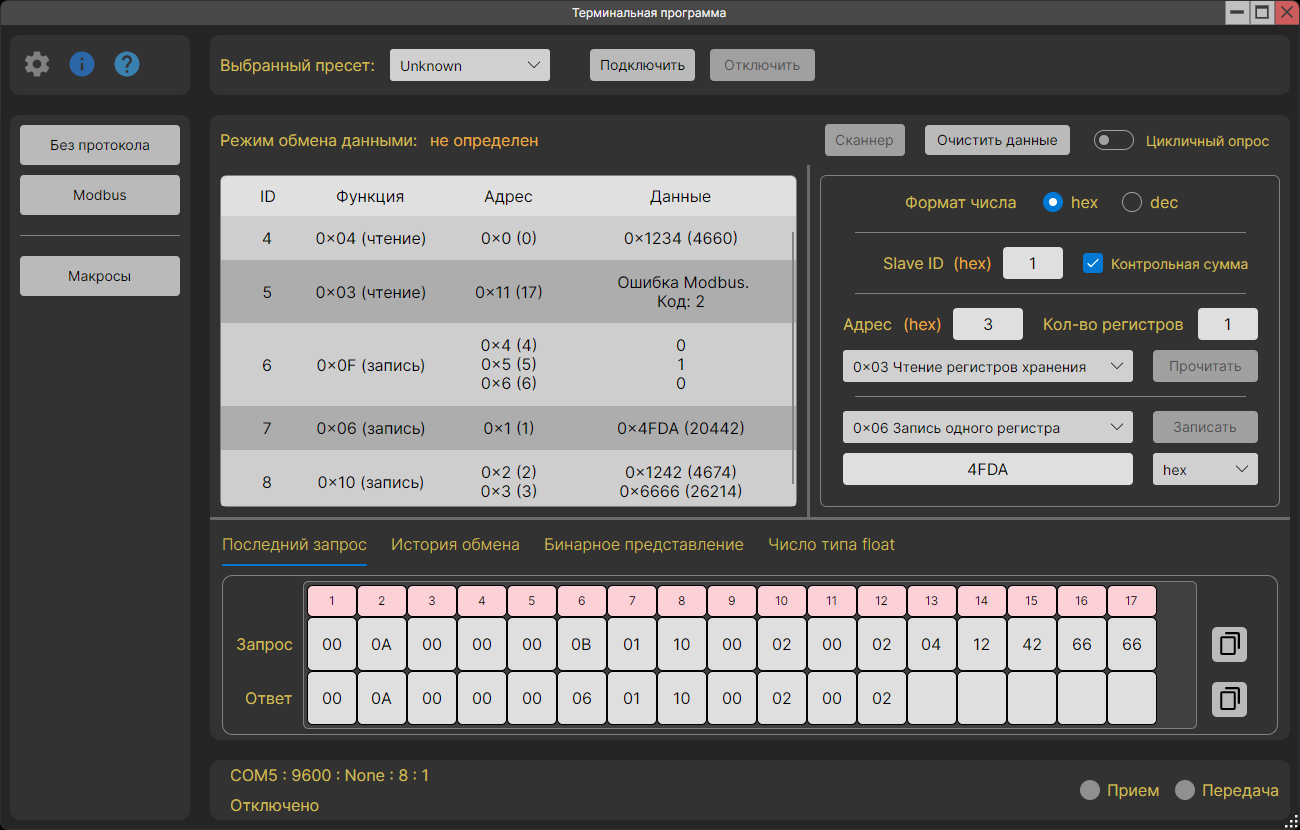


# Modbus

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



## 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 находится 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



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



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.

## Цикличный режим работы

В данном режиме можно только читать регистры с заданным периодом.



## 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



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.

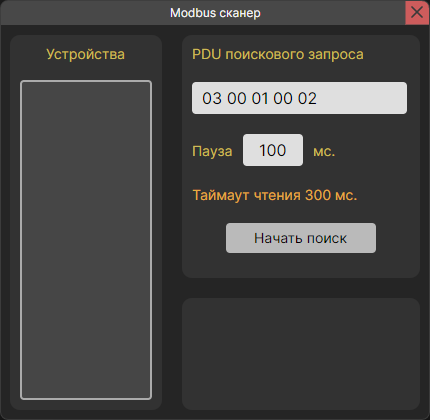
В поле «Устройства» будут отображаться адреса найденных устройств. Если после окончания поиска это поле пустое, то это означает, что ни одно устройство не ответило во всем диапазоне допустимых адресов (1 – 255). Широковещательный адрес 0 не учитывается, т.к. согласно документации устройства не должны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

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

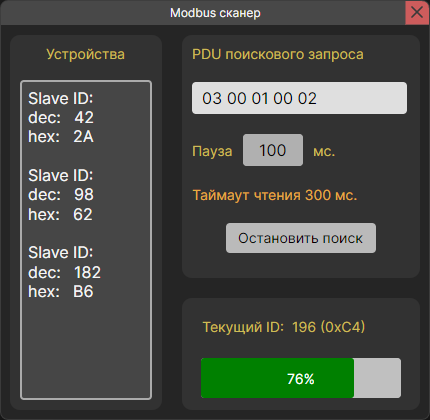
Важно понять, как работает пауза между отправками сообщений. Эта пауза состоит из двух составляющих – пользовательский таймаут, который указывается в поле «Пауза», и таймаут чтения, в течении которого приложение ждет ответа от устройства. Таймаут чтения задается в настройках.

**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

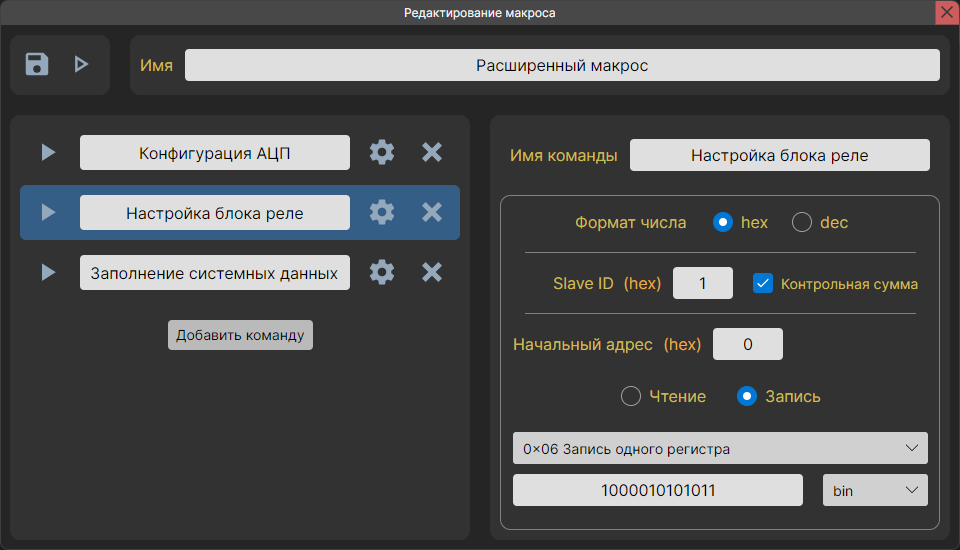
Макрос разделен на команды. Каждая команда – это отправка одного сообщения.

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.



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.

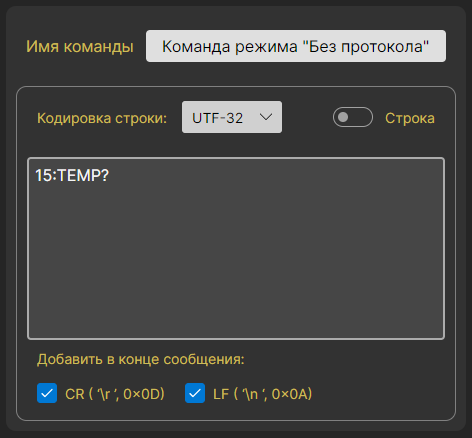
**Команда режима «Без протокола»**

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.



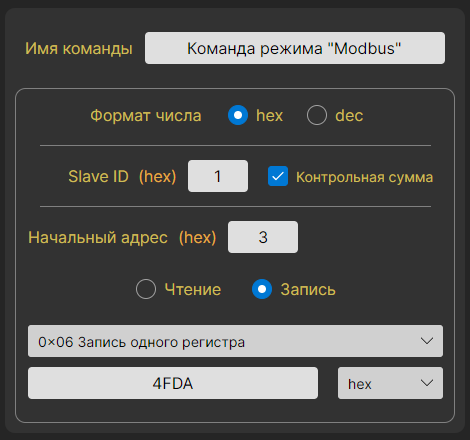
**Команда режима «Modbus»**

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

[Кроссплатформенный терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 4](https://habr.com/ru/articles/895692/)

[Кроссплатформенный терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 3](https://habr.com/ru/articles/871788/)

[Кроссплатформенный терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 2](https://habr.com/ru/articles/854824/)

[Терминал Modbus TCP / RTU / ASCII с открытым исходным кодом: Часть 1](https://habr.com/ru/articles/795387/)

# Ссылка на скачивание

[Все версии тут.](https://andreyabdulkayumov.github.io/TerminalProgram_Website/downloads.html)

# История версий

## 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