**Руководство пользователя**

**Терминальная программа (версия 3.1.0)**

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

Оглавление

[Краткое описание 3](#_Toc186987923)

[Без протокола 4](#_Toc186987924)

[Обычный режим работы 5](#_Toc186987925)

[Цикличный режим работы 5](#_Toc186987926)

[Modbus 6](#_Toc186987927)

[Обычный режим работы 7](#_Toc186987928)

[Чтение регистров Modbus 7](#_Toc186987929)

[Запись регистров Modbus 8](#_Toc186987930)

[0x05 Запись одного флага 8](#_Toc186987931)

[0x0F Запись нескольких флагов 8](#_Toc186987932)

[0x06 Запись одного регистра 9](#_Toc186987933)

[0x10 Запись нескольких регистров 9](#_Toc186987934)

[Цикличный режим работы 11](#_Toc186987935)

[Представления 12](#_Toc186987936)

[Макросы 13](#_Toc186987937)

[Без протокола 14](#_Toc186987938)

[Modbus 15](#_Toc186987939)

[Статьи на Хабр 16](#_Toc186987940)

[История версий 17](#_Toc186987941)

[3.1.0 17](#_Toc186987942)

[3.0.0 17](#_Toc186987943)

[2.7.0 17](#_Toc186987944)

# Краткое описание

**Терминальная программа** – это многофункциональный кроссплатформенный инструмент инженера, который позволяет взаимодействовать с внешним устройством или сервером.

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

Приложение поддерживает темную и светлую темы.

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.

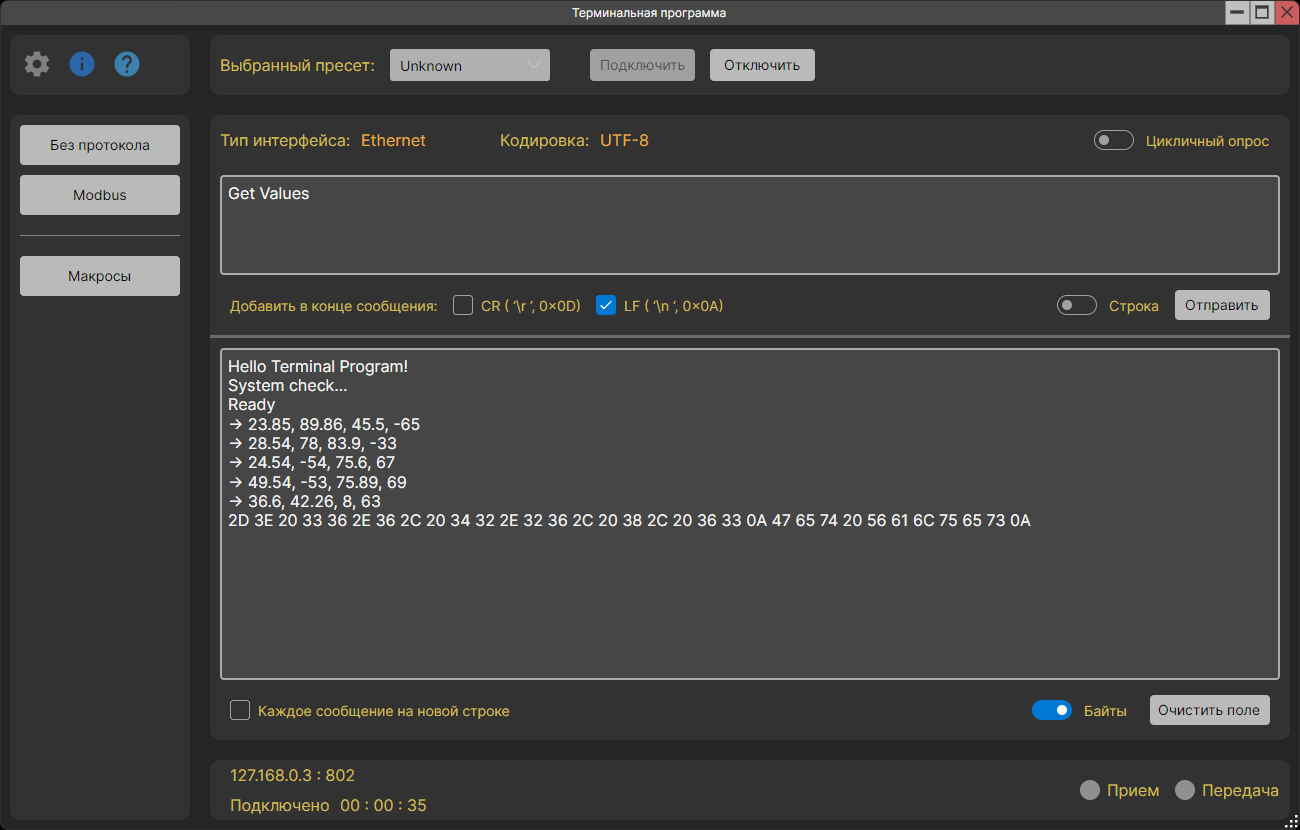
Также есть возможность работы с макросами для обоих режимов.

# 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, 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.

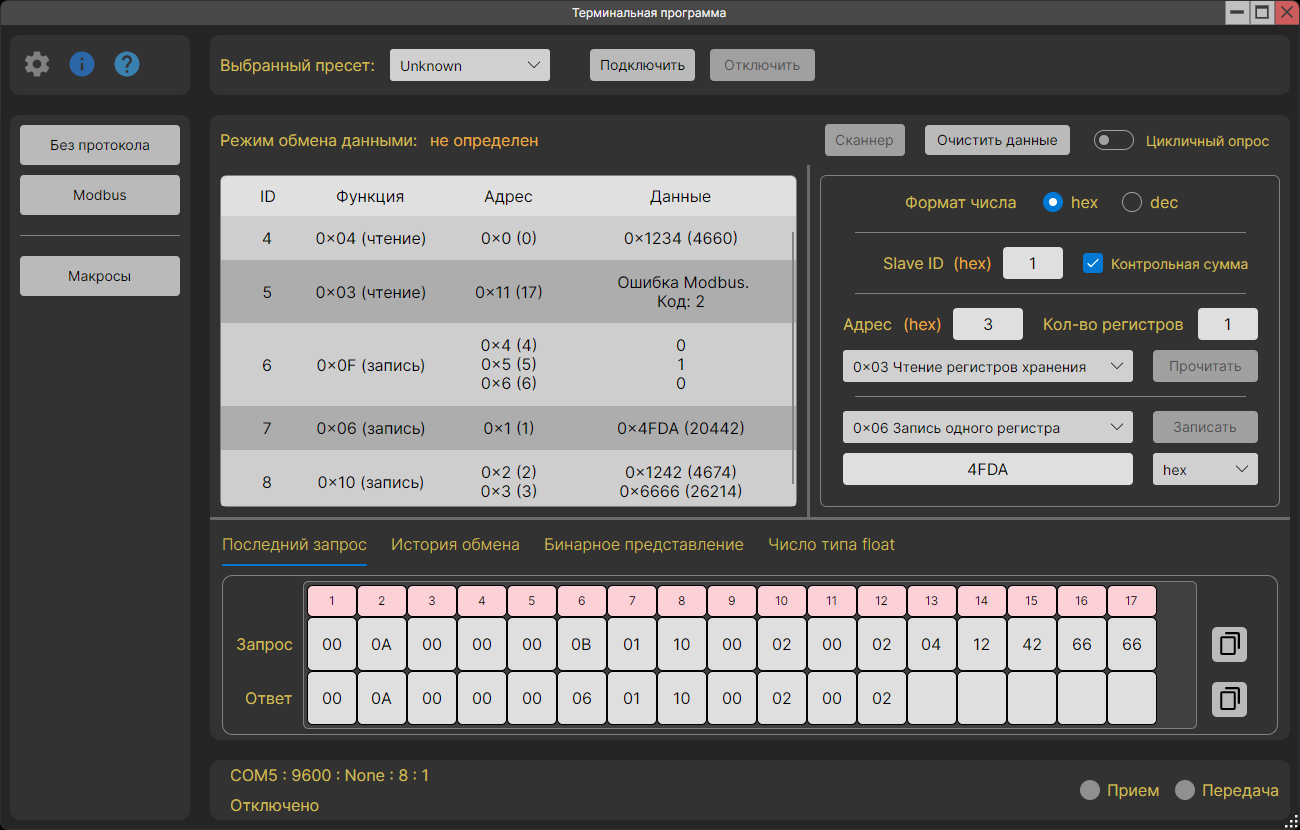


# Modbus

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

# 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.

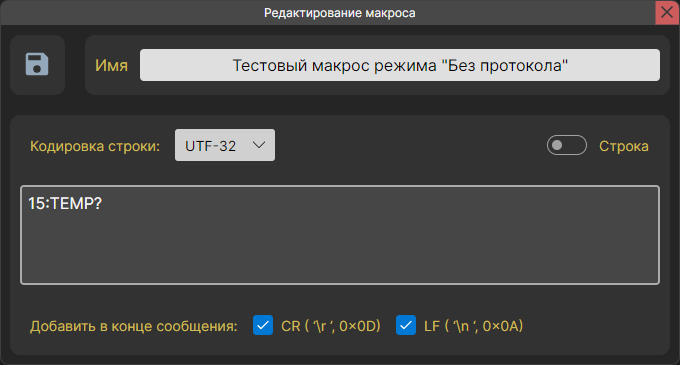
## Без протокола

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.



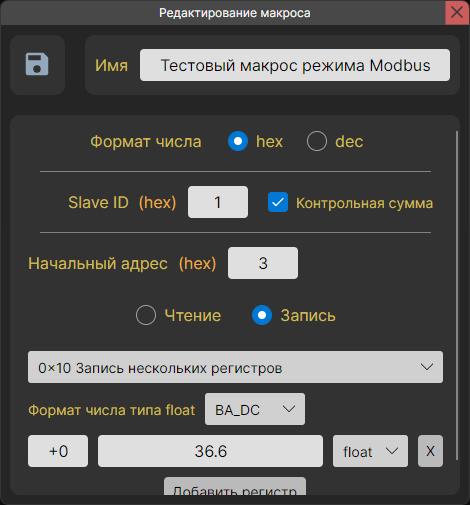
## Modbus

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 2](https://habr.com/ru/articles/854824/)

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

# История версий

## 3.1.0

## 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.

**Ссылки на скачивание приложения**

* [Win\_x64\_installer](https://disk.yandex.ru/d/Jv4LL3C19jmSNQ)
* [Win\_x64\_portable](https://disk.yandex.ru/d/cO_iBcdilOQ5YQ)
* [Linux\_x64\_portable](https://disk.yandex.ru/d/RSCfHxgfJ952YQ)

## 2.7.0

**Изменения**

* Первая публичная версия

**Ссылка на скачивание**

[Win\_x64\_installer](https://disk.yandex.ru/d/mBcN5eJWdL_cJA)