

RS-485

Contents

Описание

Как правильно проложить шину

Добавление устройства в веб-интерфейс

Как ускорить опрос устройств

Работа с портом RS-485 контроллера из собственного ПО

Описание

RS-485 — стандарт коммуникации по двухпроводной шине.

Теоретически на шину можно подключать до 256 устройств. Длина линии может быть до 1200 метров, но она сильно влияет на скорость передачи данных.

Энциклопедия АСУ ТП. Интерфейс RS-485 (https://www.bookasutp.ru/Chapter2_3.aspx) — подробно про работу интерфейса.

В устройствах Wiren Board используется Протокол Modbus поверх RS-485. Пожалуйста, ознакомьтесь с ним для лучшего понимания работы устройств.

Максимальная скорость передачи данных в периферийных устройствах Wiren Board — до 115 200 бит/с.

Как правильно проложить шину

В статье RS-485:Физическое подключение описано, как правильно проложить шину.

Добавление устройства в веб-интерфейс

RS-485:Настройка через веб-интерфейс — что сделать для появления устройства в веб-интерфейсе контроллера.

Как ускорить опрос устройств

Для ускорения опроса устройств по шине RS-485 рекомендуем:

1. Увеличить скорость обмена до 115200 бит/с. На разумных длинах и топологии сети все должно нормально работать. Если на шине есть устройства, не поддерживающие эту скорость, см. пункт 3.
2. Отключить через веб-интерфейс в настройках устройства ненужные каналы.
3. Разделить устройства по типам и портам, контроллере 2 порта RS-485 и еще 3 можно добавить модулями расширения:
 - Устройства, не поддерживающие скорость 115200, подключите отдельно.
 - Счетчики MAP так же подключите отдельно или с оборудованием, не требующим быстрой реакции. В счетчиках очень много параметров, опрос идет медленно.
 - При большом количестве устройств разделите их на несколько портов. При прочих равных скорость вырастет кратно количеству портов.

Работа с портом RS-485 контроллера из собственного ПО

- Стандартно в Wiren Board с подключёнными по RS-485 устройствами работает Драйвер `wb-mqtt-serial` (ранее `wb-homa-modbus`). Он позволяет работать с подключёнными устройствами RS-485 через систему MQTT-сообщений.
- Если вы хотите работать с портом RS-485 напрямую, не используя этот драйвер — отключите его, иначе он будет писать в порт RS-485.
- Работа с последовательным портом из Linux
- Доступ к порту RS-485 контроллера Wiren Board с компьютера
- Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board

Настройка параметров подключения по RS-485 для Modbus-устройств Wiren Board

- English
- русский

Contents

Введение

Параметры порта по умолчанию

Изменение скорости обмена

Смена уровня доступа к веб-интерфейсу

Настройка

Настройка параметров обмена

Если параметры подключения неизвестны

Введение

Устройства Wiren Board управляются по протоколу Modbus RTU и на физическом уровне подключаются через интерфейс RS-485.

Параметры порта по умолчанию

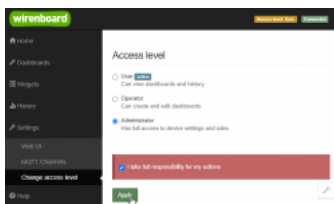
Значение по умолчанию	Название параметра в веб-интерфейсе	Параметр
9600	Baud rate	Скорость, бит/с
8	Data bits	Количество битов данных
None	Parity	Бит чётности
2	Stop bits	Количество стоповых битов

Изменение скорости обмена

Для ускорения отклика устройств на шине RS485 рекомендуем поднять скорость обмена до 115 200 бит/с.

Отметим, что низкая скорость обмена прощает многие ошибки построения шины, но на высоких скоростях выполнение рекомендаций по построению шины обязательно.

Смена уровня доступа к веб-интерфейсу



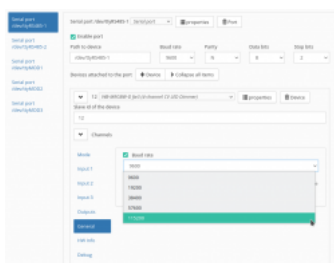
Уровень «Администратор»

Для изменения настроек контроллера у вас должен быть уровень доступа *Administrator*.

Изменить его можно в разделе **Settings** → **Change access level**.

После завершения настроек рекомендуем поставить уровень доступа *User* или *Operator* — это поможет не совершить случайных ошибок при ежедневной работе с веб-интерфейсом.

Настройка



Выбор желаемой скорости обмена в настройках устройства

Увеличим скорость обмена в Modbus-устройствах Wiren Board со значения по умолчанию до 115 200 бит/с:

1. Подключите и настройте все устройства на скорости 9600 бит/с, которая стоит у них по умолчанию.
2. Убедитесь, что все работает как надо: данные идут со всех устройств, каналы не горят красным, в системном журнале нет ошибок порта.
3. Откройте веб-интерфейс контроллера и перейдите **Settings** → **Configs** → **Serial Device Driver Configuration**.
4. Выберите нужный порт, в параметрах устройства в группе **General** поставьте флажок **Baud rate** и выберите желаемую скорость обмена: 115 200 бит/с. Скорость порта пока оставьте прежней.
5. Вверху страницы нажмите на кнопку **Save**, это запишет новое значение скорости в устройство. Но так как порт работает на старой скорости, то устройства отвечать не будут.
6. Укажите в настройках порта ту же скорость, которую вы выбрали в настройках устройства: 115 200 бит/с.
7. Снова сохраните настройки. Теперь настройки устройства и порта совпадают, устройство должно начать отвечать.

Настройка параметров обмена

Чтобы изменить параметры подключения, нам понадобится:

- знать текущие настройки подключения устройства;
- контроллер с утилитой `modbus_client` или компьютер с адаптером USB-RS485 и программой для работы с Modbus;
- номера регистров, которые описаны в таблице общих регистров.

Подготовка:

1. Подключите устройство по шине RS-485 к контроллеру или другому оборудованию, где будете выполнять команды.
2. Если вы выполняете команды на контроллере:

- откройте консоль контроллера по SSH,
- остановите драйвер `wb-mqtt-serial`.

3. Можно менять настройки устройств.

Допустим, у нас есть Modbus-устройство Wiren Board с заводскими параметрами подключения, Modbus-адресом 1 и подключённое к порту `/dev/ttyRS485-1`.

Изменим адрес устройства, для этого запишем в регистр 128 новый адрес, например 12:

```
modbus_client --debug -mrtu -b9600 -pnone -s2 /dev/ttyRS485-1 -a1 -t0x06 -r128 12
```

Теперь изменим скорость порта устройства с 9600 бит/с на 115 200 бит/с, для этого запишем в регистр 110 новое значение, формат которого можно посмотреть в таблице общих регистров:

```
modbus_client --debug -mrtu -b9600 -pnone -s2 /dev/ttyRS485-1 -a1 -t0x06 -r110 1152
```

Теперь устройство передаёт и принимает данные на скорости 115 200 бит/с.

Остальные параметры меняются аналогично: смотрите, в каком регистре хранится значение и записываете в него новое.

Если параметры подключения неизвестны

Бывает так, что параметры подключения устройства неизвестны, то можно или сбросить их к заводским, или узнать перебором, для этого загрузите на контроллер скрипт `Perebor.sh.tar.gz` и выполните его. Если адрес, к которому подключено устройство отличается от `/dev/ttyRS485-1`, измените его в теле скрипта.

Как это работает: мы обращаемся к регистру 128, в котором во всех modbus-устройствах Wiren Board хранится modbus-адрес. Вывод скрипта будет содержать строки, подобные этим:

```
Speed:9600 Stop bits:1 Parity:none Modbus address:0x0001
Speed:9600 Stop bits:2 Parity:none Modbus address:0x0001
```

Для стоп-битов, скорее всего, вы получите два значения: 1 и 2. Уточнить настройку можно считав значение из регистра 112 с уже известным адресом, скоростью, четностью:

```
modbus_client --debug -mrtu -b9600 -pnone -s2 /dev/ttyAPP1 -a0x01 -t0x03 -r112
```

или

```
modbus_client --debug -mrtu -b9600 -pnone -s1 /dev/ttyAPP1 -a0x01 -t0x03 -r112
```

SUCCESS: read 1 of elements:
Data: 0x0002

Если при чтении из регистра 112 вы получаете ошибку — устройство не поддерживает изменение параметров подключения. В этом случае для подключения используется значение по умолчанию, 2 стоп-бита.

wireboard

WB-MR6F Modbus Relay Module

wireboard

https://wireboard.com/wiki/WB-MR6F_Modbus_Relay_Module
21-06-2022 12:10

Модуль реле WB-MR6F

Руководство по эксплуатации

Самая актуальная документация всегда доступна на нашем сайте по ссылке: https://wireboard.com/wiki/WB-MR6F_Modbus_Relay_Module

Этот документ составлен автоматически из основной страницы документации и ссылок первого уровня.

Содержание

[WB-MR6F Modbus Relay Module](#)

[WB-MR6F Modbus Relay Module](#)

[Утилита «modbus_client»](#)

[Работа с Modbus-устройствами Wiren Board без контроллера](#)

[Протокол Modbus](#)

[Mapping-матрица](#)

[RS-485](#)

[Настройка параметров подключения по RS-485 для Modbus-устройств Wiren Board](#)

[Modbus-адрес устройства Wiren Board](#)

[Карта регистров модулей реле](#)

[Обновление прошивки Modbus-устройств Wiren Board](#)

WB-MR6F Modbus Relay Module

- English
- русский

Contents

[Назначение](#)



Технические характеристики

Общий принцип работы

- [Входы](#)
- [Выходы](#)

Монтаж

Настройка

- [Способы настройки](#)
- [Режим работы реле при возобновлении питания](#)
- [Безопасный режим](#)
- [Режимы взаимодействия входов и реле](#)
- [Антидребезг](#)

Представление в веб-интерфейсе контроллера WB

- [Выбор шаблона](#)
- [Управление устройством и просмотр значений](#)

Работа по Modbus

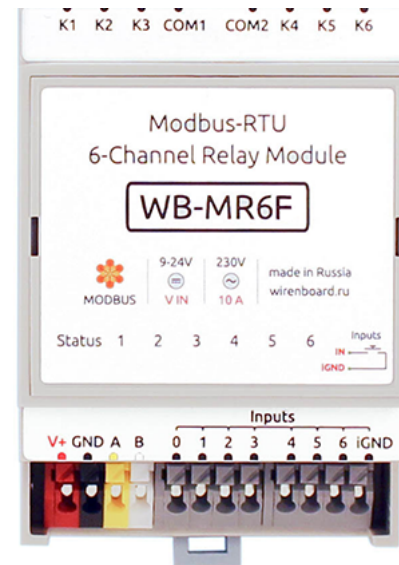
- [Параметры порта по умолчанию](#)
- [Modbus-адрес](#)
- [Карта регистров](#)

Обновление прошивки и сброс настроек

Известные неисправности

Ревизии устройства

Изображения и чертежи устройства



Релейный модуль WB-MR6F

Назначение

Специализированный двухканальный модуль реле WB-MR6F предназначен для ступенчатого управления нагрузками, например, для трёхступенчатого управления скоростью вращения вентиляторов конвекторов.

Устройство спроектировано таким образом, что неправильные программные настройки и механические неисправности реле не могут вызвать короткого замыкания входных напряжений V_x между собой.

Каналы рассчитаны на нагрузку в 3А.

Наша подробная статья на [habr.com](https://habr.com/ru/company/wirenboard/blog/422197/) о выборе параметров реле для коммутируемой нагрузки:

Технические характеристики

Параметр	Значение
Питание	
Напряжение питания	9 — 28 В постоянного тока
Потребляемая мощность	<ul style="list-style-type: none">В режиме холостого хода (со всеми выключенными реле) — 0,1 ВтСо всеми включенными реле — 1 ВтПиковое значение — до 4 Вт в течение 20 мс
Выходы	
Количество выходов	2
Тип выходов	Контакты механического реле
Конфигурация контактов	Четырёхпозиционные, нормально открытые (состояния: Open, V1, V2, V3)
Конфигурация выходов	-
Максимальное коммутируемое напряжение, AC	250 В
Максимальное коммутируемое напряжение, DC	30 В
Максимальный коммутируемый ток на каждый канал	3 А
Сопротивление контактов	< 100 миллиом
Напряжение изоляции между контроллером и выходом	1500 В (среднеквадратичное значение)
Срок жизни	100 000 переключений для нагрузки 3 А/230В переменного тока
Входы	
Количество входов (Inputs 0—6)	7 (6+1) Вход 0 отключает одновременно все реле модуля
Тип входов	"Сухой контакт", групповая изоляция. Напряжение на входе ~4,5В. Ток при замыкании входа ~2МА.
	<ul style="list-style-type: none">Входы общего назначения

Функции	<ul style="list-style-type: none"> Счет сигналов Прямое управление каналами реле Одновременное отключение всей нагрузки Запоминание состояния реле при отключении напряжения
Индикация	
Индикация питания и обмена данными	Зеленый светодиод Status (расположен под поверхностью верхней наклейки)
Индикация состояния каналов реле	Красно-оранжевые светодиоды 1 — 6 (расположены под поверхностью верхней наклейки)
Управление	
Интерфейс управления	RS-485
Изоляция интерфейса	Неизолированный
Протокол обмена данными	Modbus RTU, адрес задается программно, заводские настройки указаны на наклейке
Параметры интерфейса RS-485	Задаются программно, по умолчанию: скорость — 9600 бит/с; данные — 8 бит; бит чётности — нет (N); стоп-биты — 2
Готовность к работе после подачи питания	~2 с
Условия эксплуатации	
Температура воздуха	От -40 до +80 °C
Относительная влажность	До 92 %, без конденсации влаги
Клеммники и сечение проводов	
Рекомендуемое сечение провода с НШВИ	для входов управления: 0.35 - 1 мм ² — одинарные, 0.35 - 0.5 мм ² — сдвоенные провода, для силовых входов: до 2.5 мм ² — одинарные, до 1.5 мм ² — сдвоенные провода
Длина стандартной втулки НШВИ	8 мм
Момент затяжки винтов	для входов управления: 0.2 Н•м, для силовых выходов: 0.5 Н•м
Габариты	
Ширина, DIN-юнитов	3
Габаритные размеры (Д x Ш x В)	53 x 90 x 58 мм

Общий принцип работы

Входы

Входы WB-MR6F выведены на разъемные клеммники и работают по принципу "сухой контакт". Кнопки или выключатели подключаются между входами блока и iGND. Можно использовать кнопки с фиксацией или без нее. Есть дополнительный вход «0» — по умолчанию настроен на отключение всех реле.

Дискретные входы можно использовать для прямого управления каналами реле или настроить внутреннюю логику взаимодействия входов с выходами. Подробнее смотрите в разделе [Режимы взаимодействия входов и реле](#). Реле обычно применяются для подключения настенного клавишного выключателя, чтобы управлять освещением напрямую.

С версии прошивки 1.12.0 изменился режим работы с выключателями — теперь по умолчанию модуль настроен на выключатели с фиксацией — состояние контактов реле повторяют состояние контактов выключателя. До этого модули были по умолчанию настроены на выключатели без фиксации — каждое замыкание входа меняло состояние реле на противоположное.

Для каждого входа можно изменить параметр времени антидребезга. Подробнее смотрите в разделе [Антидребезг](#).

На каждом канале доступно измерение количества срабатываний, а начиная с версии прошивки 1.15.0 можно измерять и частоту импульсов на входе.

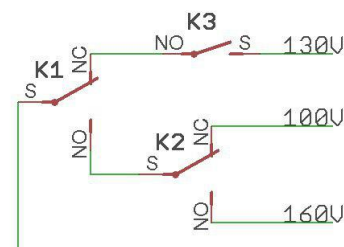
Максимальное значение измеряемой частоты сигнала зависит от ревизии устройства и от количества измеряемых частотных сигналов. На испытании проводилось измерение шести сигналов одновременно на частоте 2.5 кГц.

Выходы

В модуле установлены 6 реле HF32F ([Datasheet](#)).

Каждый из двух каналов состоит из двух перекидных реле и одного реле с нормально-разомкнутыми контактами. В зависимости от положения реле, каждый из выходов OUTx либо соединяется с одним из трёх входов Vx, либо отключается от входов Vx (выключенное состояние).

Каждый выход защищен от образования дуги при замыкании/размыкании варистором.



Монтаж

Устройство монтируется на стандартную DIN-рейку шириной 35 мм. Клеммный блок «V+ GND A B» с шагом 3.5 мм служит для подключения питания и управления по шине RS-485. Для стабильной связи с устройством важно правильно организовать подключение к шине RS-485, читайте об этом в статье [RS-485: Физическое подключение](#).

Рекомендуем для монтажа использовать гибкие многожильные провода с обжатием концов втулочными наконечниками (НШВИ — наконечник штыревой втулочный изолированный).

При снятии изоляции провод должен зачищаться ровно по длине гильзы (можно зачистить больше, а потом откусить выступающий излишек). Для обжима (опрессовывания) используйте пресс-клещи (кримпер, «обжимка»). При монтаже обжатый наконечником провод не разрушается винтовым зажимом и надежно фиксируется.

Не прикладывайте чрезмерное усилие при завинчивании клеммы — это приводит к разрушению винтового разъема.

При включении реле модуль импульсно потребляет большую мощность (см. потребление). Используйте блок питания с запасом по мощности. При питании по длинному кабелю учитывайте падение напряжения на нем.

Срабатывание дискретных входов происходит при их замыкании на землю iGND. Возникающий ток невелик (~2 мА), так что тип кнопки или выключателя может быть любым. Также не играет роли длина и сечение кабеля. Но для предотвращения наводок от близко расположенных силовых линий лучше использовать витую пару и подключать землю iGND ко второй жиле пары.

Сечение проводов, подключаемых к винтовым зажимам выходов реле, должно соответствовать мощности коммутируемой нагрузки. Винтовые зажимы принимают провод сечением 2.5–4 мм². Выбрать правильное сечение провода для подключения нагрузки поможет таблица [Допустимый длительный ток для проводов и шнуров](#).

Модуль реле необходимо устанавливать таким образом, чтобы удовлетворять требованиям электробезопасности и не допускать случайного касания контактов, находящихся под высоким напряжением. Устройство должно эксплуатироваться при рекомендованных условиях окружающей среды.

Настройка

Способы настройки

1. Указать параметры в веб-интерфейсе контроллера Wiren Board. Перейдите на [страницу настройки serial-устройств](#), выберите порт, найдите или добавьте устройство и измените параметры. Если нужный параметр отсутствует в шаблоне, его можно задать через пользовательские параметры.
2. Записать настройки в [Modbus-регистры](#) модуля из консоли контроллера с помощью утилиты [modbus_client](#).
3. Если нет контроллера Wiren Board, используйте [адаптер USB-RS485](#).

Режим работы реле при возобновлении питания

Устройство запоминает состояние выходов при отключении питания, но вы можете это изменить.

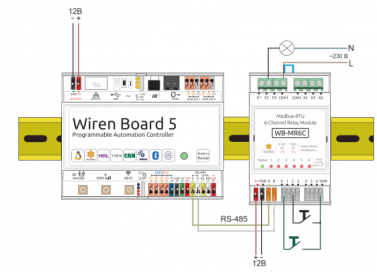
Выберите нужный режим в параметре **Restore Last Outputs State After Power On**.

Безопасный режим

Безопасный режим позволяет остановить технологические процессы в случае, если контроль над модулем утрачен, например, перебит кабель RS-485. Рекомендуем учитывать возможность потери связи с контроллером при проектировании систем управления.

Таймер этого режима начинает отсчёт после каждого успешно обработанного (принятого) пакета [Modbus](#). При достижении установленного времени — выходы реле отключаются. Если была включена маппинг-матрица, то реле можно управлять напрямую от его входов.

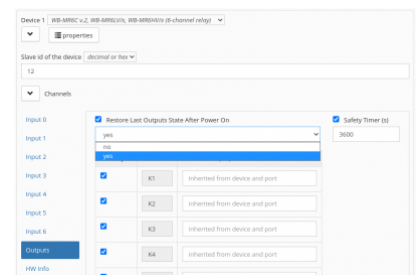
Значение таймера указывается в параметре **Safety Timer (s)** — значение «0» отключает безопасный режим. По умолчанию безопасный режим включен.



Образец монтажа и подключения модуля WB-MR6C. Кнопка с проводами зеленого цвета, подключенная ко входу 0, отключает все каналы одновременно



Как обжимать наконечники НШВИ



Пример настройки реле WB-MR6C v.2:

отключает безопасный режим. По умолчанию безопасный режим выключен.

Пример настройки реле WB-MR6C v2: таймер безопасного режима и выбор состояния выходов при возобновлении питания

Режимы взаимодействия входов и реле

В модулях для каждого дискретного входа можно настроить внутреннюю логику, которая позволяет управлять выходами реле. Изменить режим можно в параметре **Input x Mode** или настроить логику через **Mapping-матрицу**.

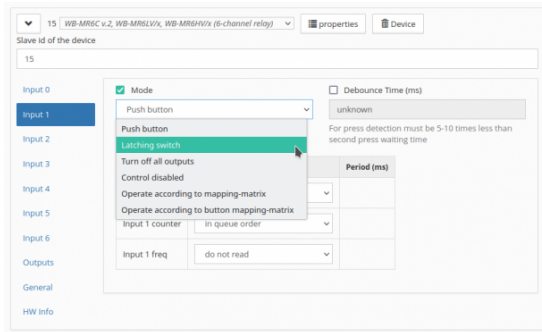
Режимы по умолчанию:

- каждый вход управляет соответствующим реле, т.е. вход номер 2 управляет реле номер 2
- режим работы для входов — выключатель с фиксацией (до середины 2019 года, кнопка без фиксации)
- нулевой вход отключает все реле.

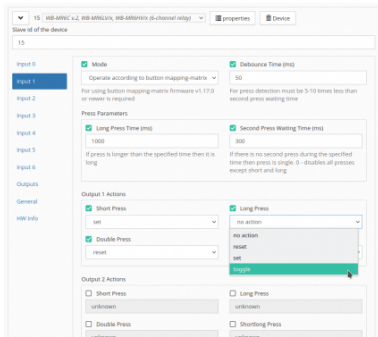
Начиная с версии прошивки 1.17.0 появились дополнительные матрицы с возможностью настроить входы как кнопки для детектирования различных типов нажатий: одинарное, двойное, длительное и т.п.

Кроме этого, можно полностью отключить обработку состояния любого из входов, притом, в веб-интерфейсе и регистрах можно будет отслеживать их состояние и обрабатывать программно на контроллере.

- Пример настройки реле WB-MR6C v.2



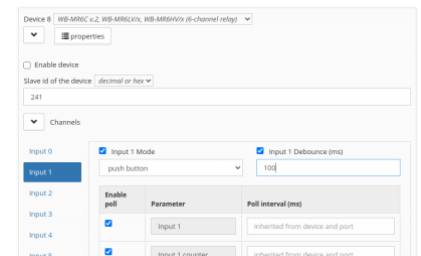
Выбор режима для 1 входа реле



Привязка действий к типам нажатий

Антидребезг

Для любого из входов реле вы можете настроить фильтр антидребезга в параметре **Input x Debounce (ms)**. Возможные значения от 0 до 250 мс (от 0 до 100 в прошивках до 1.17.8), значение по умолчанию — 50 мс.



Пример установки времени антидребезга

Представление в веб-интерфейсе контроллера WB

Выбор шаблона

Чтобы устройство появилось на вкладке *Devices* в веб-интерфейсе контроллера Wiren Board, добавьте новое serial-устройство и выберите шаблон **WB-MR6C**.

Управление устройством и просмотр значений

Каждому каналу соответствуют три переключателя, управляющие тремя реле в группе: K1, K2, K3 относятся к первому каналу, K4, K5, K6 - ко второму.

Соответствие реле положению выходов:

K1/ K4	K2/ K5	K3/ K6	OUT1/ OUT2
OFF	OFF ON	OFF	Отключен
ON	OFF	ON OFF	V1
ON	ON	ON OFF	V3
OFF	ON OFF	ON	V2

The screenshot shows the web interface for WB-MR6C 145. It features a dropdown menu for the device name. Below it, there are three channel control buttons labeled K1, K2, and K3. K1 is currently OFF, while K2 and K3 are ON. To the right, there are three input status flags labeled Input 1, Input 2, and Input 3, all currently OFF. Further right, there are three input counters labeled Input 1 counter, Input 2 counter, and Input 3 counter, all showing a value of 0. On the far right, there is a Safety Timer control labeled 'Sa' and 'C'.

С помощью виртуальных выключателей в веб-интерфейсе K1-K6 можно управлять соответствующими выходами WB-MR6F и следить за их состоянием. Если реле будет включено или выключено через внешний вход, это отразится и в веб-интерфейсе. Текущее состояние входов показывают флажки Input1 — Input6 (на них нельзя щелкнуть, чтобы изменить состояние). Счетчики нажатий/включений отображаются в полях Input 1 counter - Input 6 counter. Значения счетчиков хранятся в оперативной памяти микроконтроллера модуля реле и обнуляются при выключении/включении питания и сбросе.

Ползунок Safety Timer позволяет задать время отключения (в секундах) всех выходов при отсутствии обмена данными с Wiren Board (0 - таймер безопасности отключен). При возобновлении связи с контроллером выключившиеся реле останутся выключенными. Ползунком удобно управлять с помощью клавиш со стрелками, предварительно сфокусировавшись на нем мышью - перемещение ползунка с помощью мыши не обеспечивает достаточную точность. Таймер безопасности позволяет остановить технологические процессы в случае, если контроль над устройством утрачен (например, перебит кабель RS-485).

Кнопка, подключенная ко входу 0, позволяет отключить все включенные реле модуля одним нажатием. Повторное нажатие включит все реле, которые были включены изначально. Состояние реле хранится в оперативной памяти, поэтому если повторное нажатие кнопки произойдет после перебора питания, восстановления состояния реле не произойдет. Возможность восстановления состояния реле после перебора питания описана в разделе Режим работы реле в случае отключения питания.

Работа по Modbus

Устройства Wiren Board управляются по протоколу Modbus RTU. На физическом уровне подключаются через интерфейс RS-485.

Поддерживаются все основные команды чтения и записи одного или нескольких регистров. Смотрите список доступных команд в описании протокола Modbus.

Настроить параметры модуля можно в веб-интерфейсе контроллера Wiren Board, или через сторонние программы.

Параметры порта по умолчанию

Значение по умолчанию	Название параметра в веб-интерфейсе	Параметр
9600	Baud rate	Скорость, бит/с

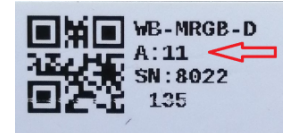
Значение	Битовая скорость	Скорость, бит/с
8	Data bits	Количество битов данных
None	Parity	Бит чётности
2	Stop bits	Количество стоповых битов

Для ускорения отклика устройств **рекомендуем поднять скорость обмена до 115 200 бит/с**, см. [Настройка параметров обмена данными](#)

Modbus-адрес

Каждое устройство на линии имеет уникальный адрес в диапазоне от 1 до 247. Адрес устройства, установленный на заводе, указан на отдельной наклейке со штрихкодом. На заводе устройствам Wirenboard в одной партии присваиваются разные адреса, поэтому в вашем заказе, скорее всего, адреса не будут повторяться.

О том, как узнать, изменить или сбросить Modbus-адрес устройства, читайте в статье [Modbus-адрес устройства Wiren Board](#).



Modbus-адрес, установленный на производстве

Карта регистров

[Карта регистров модулей реле](#)

Обновление прошивки и сброс настроек

При обновлении прошивки устройства пользовательские настройки удаляются. Если вы задавали настройки в веб-интерфейсе контроллера, то они будут автоматически восстановлены при первом опросе устройства.

В устройствах Wiren Board выпущенных с 2019 года можно обновлять прошивку по протоколу Modbus. Это даёт возможность расширять функциональные возможности устройств и устранять ошибки в прошивке прямо на месте монтажа.

Инструкции:

- [Обновление прошивки](#)
- [Настройка параметров подключения](#)
- [Modbus-адрес: узнать, сбросить или изменить](#)

Известные неисправности

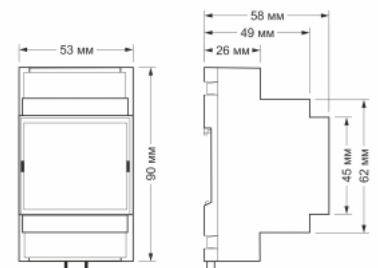
[Список известных неисправностей](#)

Ревизии устройства

Номер партии (Batch №) указан на наклейке на боковой поверхности корпуса или на печатной плате.

Ревизия	Партии	Дата выпуска	Отличия от предыдущей ревизии
			▪
			▪

Изображения и чертежи устройства



Габаритные размеры WB-MR6F в корпусе 3 DIN

WB-MR6F Modbus Relay Module

- English
- русский

Contents

Purpose

Outputs

Inputs

Technical specifications

== Module dimensions ==

[Data exchange](#)

[Installation](#)

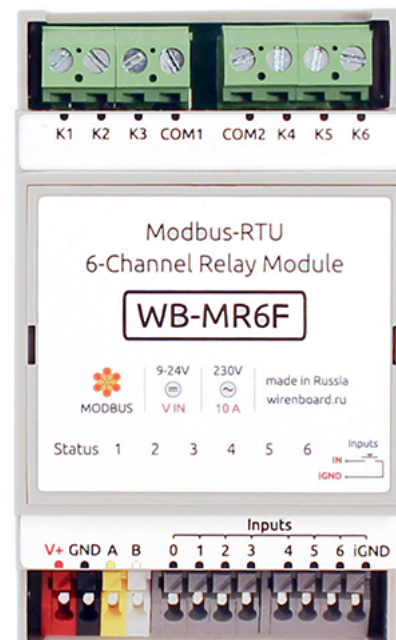
[Control of the module via the Wiren Board web interface](#)

[Relay operation in case of power failure](#)

[Modbus Control](#)

[Device firmware update](#)

[Known faults and bugs](#)



Purpose

Specialized two-channel relay module WB-MR6F is designed for step load control, for example, for three-stage speed control of convector fans.

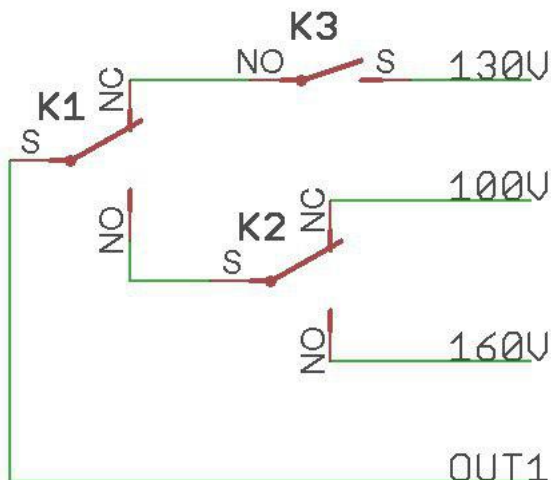
Incorrect software settings and mechanical faults of the relay can not cause a short circuit of the input voltages V_x between themselves because of the device design.



Релейный модуль WB-MR6F

The channels are designed for a 3A load.

Outputs



The module has 6 HF32F relays ([Datasheet](#)).

Each of the two channels of the module consists of two rocker relays and one relay with normally open contacts. Depending on the position of the relay, each of the OUT_x outputs is either connected to one of the three V_x inputs or disconnected from the V_x inputs (off state).

The module can be timer enabled safe mode, which at the termination of the data exchange through Modbus module disables all the relays after a specified time.

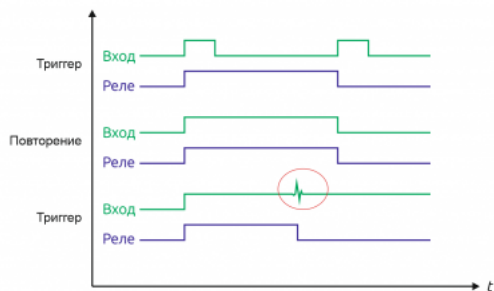
The table below lists the specifications and conditions of use of the WB-MR6F relay module. Each output of the module is protected from arcing when the varistor closes/opens. Depending on the Modbus register settings, the status of the inputs can be resumed after a power interruption (see [Relay operation mode in case of power failure](#)).

Inputs

The inputs of the modules WB-MR6F displayed on the screwless terminal blocks. Inputs are organized on the principle of dry contact. Buttons or switches are connected between the corresponding inputs and iGND inputs. There is a special input for disconnecting all relays of the module at the same time.

Each input controls the state of the corresponding relay.

The inputs have two modes of operation: trigger mode (for control with buttons) and repeat mode (for control with switches). In the trigger mode, to change the state of the relay, it is enough to close and open the input once. The change of state occurs on the rising edge (that is, when the circuit). In the repeat mode, the relay state is determined by the input state: the input is closed — the relay is on, open — off. Details about the mode switching written in [Relay Module Modbus Management](#).



Closed input in trigger mode may be unstable

It is not recommended to use switches (e.g. buttons with fixation) in the trigger mode of inputs. Input interference may be perceived as a short

release/push of the button and the relay status will change, although the switch will remain closed.

Technical specifications

Our detailed article in russian on [habr.com](https://habr.com/ru/company/wirenboard/blog/422197/) about the choice of relay parameters for the switched load: <https://habr.com/ru/company/wirenboard/blog/422197/>

Parameter	Value
Power	
Supply voltage	9 — 28 V DC
Power consumption	<ul style="list-style-type: none"> ▪ In idle mode (with all relays switched off) — 0.1 W ▪ With all relays included — 1 W ▪ Peak value — up to 4 W for 20 MS
Outputs	
Number of outputs	2
Output type	Mechanical relay contacts
Contact configuration	Four-position, normally open (States: Open, V1, V2, V3)
Output configuration	-
Maximum switching voltage, AC	250 V
Maximum switching voltage, DC	30 Volts
Maximum switching current per channel	3 A
Contact resistance	< 100 milliohms
Voltage isolation between the controller and the output	1500 V (RMS value)
Term life:	100 000 switchings for 3 A/230V AC load
Inputs	

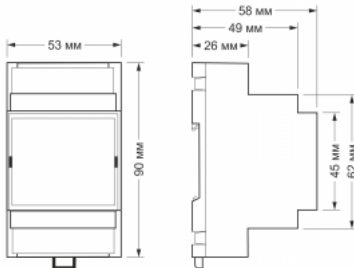
Number of inputs (Inputs 0—6)	7 (6+1) Input 0 disconnects all relays of the module simultaneously
Input type	"Dry contact", group insulation. The input voltage is ~4.5 V. Current closing input ~2mA.
Functions	<ul style="list-style-type: none"> ▪ General purpose inputs ▪ Signal count ▪ Direct control of relay channels ▪ Simultaneous disconnection of the entire load ▪ Remember the relay status in case of power cuts
Control	
Management interface	RS-485
Interface isolation	Uninsulated
Communication protocol	Modbus RTU, address is set by software, factory settings are indicated on the label
RS-485 interface parameters	set programmatically (see register Card); default: speed — 9600 bit/s; data — 8 bit; parity bit — none (N); stop bits — 2
Ready for operation after power supply	~0.03 c
dimensions	
Width, DIN units	3
Overall dimensions (l x W x h)	53.3 x 56.3 x 57.5 mm
Display	
Power and data exchange indication	Green Status led (located under the surface of the top sticker)
Relay channel status indication	Red-orange LEDs 1 — 6 (located under the surface of the top sticker)
operating Conditions	
Air temperature	
Relative humidity	Up to 92%, no condensation

== Module dimensions ==

The overall dimensions of the module are 53.3 x 90.2 x 57.5 mm (L x W x H), see features.

Data exchange

At the physical level, the module is connected via RS-485 interface. Modbus RTU Protocol is used to control the WB-MR6F. In Wiren Board devices, Modbus data is transmitted over RS-485 communication lines. See the [Modbus Protocol](#) page for details. Modbus address of the device is set at the factory and marked on the label. The address can be changed programmatically. Details refer to the [Modbus control](#).



the dimensions of the modules in the housing 3 DIN

Installation

The relay module is mounted on a standard 35 mm wide DIN rail and occupies a width of 3 DIN modules.

The screw-free terminal blocks on the relay board are used to connect power lines, control (RS-485) and input contacts. When using handpieces for installation of wires in screw-free clamps, it is necessary that the diameter of the insulated cuffs does not exceed 3.6 mm, the wire section — 0.75 mm², and the length of the conductive sleeve — 5-6 mm. Screw clamps accept wire section 2.5—4 mm². The wire section must correspond to the switched load.

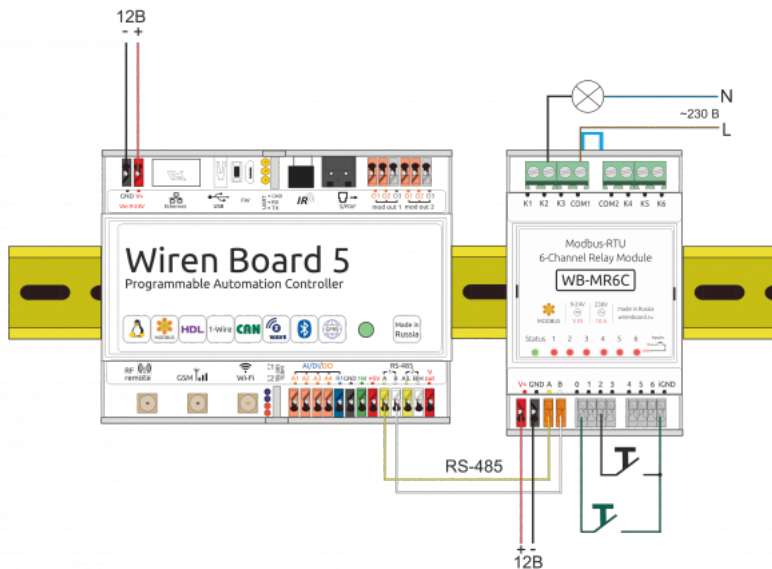
When connecting the switching devices to the relay contacts, it is important not to exceed the recommended current of **3 A**.

The current flowing when the button or switch is closed is small, so the type of button or switch can be any. Despite the built-in protection against rattle, choose quality buttons and switches to avoid false alarms due to bad contacts.

If the device is the last one on the RS-485 line, a 120 Ohm resistor-terminator must be installed between its inputs A and B. Practice shows that in case of bench tests with a small length of the RS-485 line and a small number of terminator devices, it's not necessary to install terminator on the last device in the line

to install terminals on the PCB device in the case.

The module must be installed to meet the requirements of electrical safety and to prevent accidental contact of contacts under high voltage. The module must be operated under recommended environmental conditions.



Sample installation and connection of WB-MR6C module. Scheme for WB-MR6F is the same. The choice of a quality power supply is very important for the operation of the module. A button with green wires connected to the module input 0 disconnects all channels simultaneously.

Control of the module via the Wiresh Board web interface

The device uses the WB-MR6F template. Each channel corresponds to three switches that control three relays in the group: K1, K2, K3 belong to the first channel, K4, K5, K6 - to the second.

The relay corresponds to the position of the outputs:

K1/ K4	K2/ K5	K3/ K6	OUT1/ OUT2
OFF	OFF ON	OFF	OFF
ON	OFF	ON OFF	V1
ON	ON	ON OFF	V3
OFF	ON OFF	ON	V2

Using virtual switches K1-K6 in the web-interface, you can control the corresponding outputs of the module and monitor their status. If the relay is turned on or off via the external input, it will be reflected in the web-interface. The current state of the inputs is indicated by the Input1 — Input6 checkboxes (they cannot be clicked to change the state). Press/enable counters are displayed in the Input 1 counter - Input 6 counter fields. The meter values are stored in the memory of the relay module microcontroller and are reset when the power is turned on/off and reset.

The Safety Timer slider allows you to set the shutdown time (in seconds) of all outputs when there is no communication with the Wiresh Board (0 - the safety timer is disabled). When communication with the controller is resumed, the switched-off relays will

remain switched off. The slider is easy to control with the arrow keys, previously focusing on it with the mouse: moving the slider with the mouse does not provide sufficient accuracy. The safety timer allows stopping technological processes in case the control over the module is lost (for example, the RS-485 cable is broken).

A button connected to input 0 allows you to turn off all the relays included in the module with a single press. Pressing again will turn on all the relays that were switched on initially. The status of the relay is stored in memory, so if you press the button again after a power failure, the restoration of the relay will not happen. The ability to restore relay status after power supply interruption as described in section [Relay operation mode in case of power failure](#).

Relay operation in case of power failure

Depending on the value written to the storage register 0x06, after the power is turned off and restored, the relays will remain off (if the value is 0, or turn on those that were turned on before the power interruption if the register has a value of 1). This feature appeared in the modules with the firmware, starting with version 1.5.3.

Modbus Control

Details of the work with the module via Modbus Protocol is written in [Relay Modules Modbus Management](#). There you can also find a map of the device registers.

[Back to peripheral devices list](#)

Device firmware update

Starting with the new firmware versions of the device, the firmware update functionality (firmware) via the Modbus Protocol is supported. This makes it possible to extend the functionality of the devices and eliminate errors in the firmware directly at the installation site. Read more about flashing devices described in the article [Download firmware to Wiren Board devices](#). A list of firmware versions that support the update, as well as devices for which this feature is implemented, can be found in [Firmware Changelog](#).

Known faults and bugs

[List of known faults](#)

Утилита «modbus_client»

Contents

[Описание](#)

[Подготовка к работе](#)

[Контроллер Wiren Board](#)

[Настольный компьютер с Linux](#)

[Аргументы командной строки](#)

[Примеры использования с оборудованием Wiren Board](#)

[Проверка подключения к устройству и считывание адреса](#)

[Запись нового адреса](#)

[Чтение сигнатуры устройства](#)

[Чтение версии прошивки](#)

[Настройка параметров трансформаторов](#)

[Включение реле релейного модуля](#)

[Одновременное включение нескольких реле](#)

[Настройка взаимодействия входов и выходов реле](#)

Описание

modbus_client — утилита для опроса устройств по протоколам Modbus RTU и Modbus TCP из командной строки.

Подготовка к работе

Контроллер Wiren Board

Утилита `modbus_client` предустановлена на все контроллеры Wiren Board. Для использования утилиты нужно подключиться к контроллеру по протоколу [SSH](#).

Обычно порт RS-485 занят драйвером `wb-mqtt-serial`, поэтому перед запуском `modbus_client` этот драйвер надо остановить:

```
service wb-mqtt-serial stop # для Wiren Board 5 и позднее
service wb-homa-modbus stop # для Wiren Board 4
```

После завершения работы с `modbus_client` запустите драйвер обратно:

```
service wb-mqtt-serial start # для Wiren Board 5 и позднее
service wb-homa-modbus start # для Wiren Board 4
```

Настольный компьютер с Linux

Скачайте пакет для настольных компьютеров с Linux (https://github.com/contactless/modbus-utils/releases/download/1.2/modbus-utils_1.2_amd64.deb).

Перейдите в папку со скаченным пакетом и установите его командой:

```
sudo apt install ./modbus-utils_1.2_amd64.deb
```

Также автоматически должен установиться пакет `libmodbus`, если этого не произошло — установите его из репозитория `apt`.

Аргументы командной строки

Значения параметров (адрес устройства или регистра, таймаут, тип функции, значение для записи в регистр и т.д.) можно указывать как в шестнадцатеричном `0x**`, так и в десятичном виде.

Вызов `modbus_client` без аргументов выдает краткое описание возможных аргументов команды:

```
modbus_client [--debug] [-m {rtu|tcp}] [-a<slave-addr=1>] [-c<read-no>=1]
  [-r<start-addr>=100] [-t<f-type>] [-o<timeout-ms>=1000] [{rtu-params|tcp-params}] serialport|host [<write-data>]
NOTE: if first reference address starts at 0, set -0
f-type:
  (0x01) Read Coils, (0x02) Read Discrete Inputs, (0x05) Write Single Coil
  (0x03) Read Holding Registers, (0x04) Read Input Registers, (0x06) Write Single Register
  (0x0F) Write Multiple Coils, (0x10) Write Multiple register
rtu-params:
  b<baud-rate>=9600
  d{7|8}<data-bits>=8
  s{1|2}<stop-bits>=1
  p{none|even|odd}=even
tcp-params:
  p<port>=502
Examples (run with default mbServer at port 1502):
Write data:  modbus_client --debug -mtcp -t0x10 -r0 -p1502 127.0.0.1 0x01 0x02 0x03
Read that data: modbus_client --debug -mtcp -t0x03 -r0 -p1502 127.0.0.1 -c3
```

Общие аргументы

Параметр	Описание	Обязателен	Значение по умолчанию
<code>--debug</code>	Может указываться в любой позиции и включает отладку, выводя на экран шестнадцатеричные коды отправляемых и принимаемых данных.	нет	
<code>-m</code>	Определяет тип используемого протокола: <ul style="list-style-type: none">▪ <code>-mrtu</code> — Modbus RTU,▪ <code>-mtcp</code> — Modbus TCP. Он должен указываться первым в командной строке, или вторым, если первый аргумент — <code>--debug</code> или имя файла порта RS-485.	да	
<code>-a</code>	Задаёт Modbus-адрес устройства, к которому мы обращаемся.	нет	1
<code>-c</code>	Определяет, какое количество элементов мы запрашиваем.	нет	1
<code>-r</code>	Задаёт начальный адрес для чтения или записи.	нет	100
<code>-t</code>	Указывает код функции Modbus. Кратко они перечислены в выводе <code>modbus_client</code> , подробнее значения кодов описаны на странице Протокол Modbus .	да	
<code>-o</code>	Задаёт таймаут в миллисекундах.	нет	1000
<code>-0</code>	Ноль. Уменьшает на единицу адрес, задаваемый аргументом <code>-r</code> . Это может быть полезным при работе с устройствами с нестандартной адресацией, например, с диапазоном адресов 1 — 65536 вместо привычного 0 — 65535.	нет	

Затем указываются специфические параметры протокола (Modbus RTU или Modbus TCP). Несмотря на информацию,

выводимую в подсказке, эти параметры также начинаются со знака - (минус, дефис).

Для Modbus RTU

Параметр	Описание	Значение по умолчанию
-b	Скорость передачи данных по последовательной линии	9600
-d	Количество передаваемых бит данных, 7 или 8	8
-s	Количество стоповых битов, 1 или 2	1
-p	Контроль четности: <ul style="list-style-type: none">-pnone — нет проверки,-peven — передается бит контроля на четность,-podd — передается бит контроля на нечетность.	even

Для Modbus TCP

Параметр	Описание
-p	Номер TCP-порта устройства, с которым взаимодействует контроллер.

Далее следует имя файла порта RS-485 или адрес хоста, а в конце необязательный параметр — данные для функций записи.

Примеры использования с оборудованием Wiren Board

Проверка подключения к устройству и считывание адреса

Все устройства Wiren Board с протоколом Modbus RTU хранят адрес в регистре 128 — его удобно считывать для проверки подключения.

Читаем содержимое регистра 128 из устройства с адресом 2, подключенного к serial-порту /dev/ttyRS485-1, с помощью функции 0x03 (Read Holding Registers):

```
modbus_client --debug -mrtu -b9600 -pnone -s2 /dev/ttyRS485-1 -a2 -t0x03 -r128
```

Аргумент	Описание
--debug	отладка включена, будут выведены шестнадцатеричные коды отправляемых и принимаемых данных
-mrtu	выбран протокол Modbus RTU
-pnone	без проверки контроля четности
-s2	стоповых битов 2
/dev/ttyRS485-1	адрес serial-порта, к которому подключено опрашиваемое устройство
-a2	адрес устройства, 2
-t0x03	адрес функции чтения из holding-регистра
-r128	адрес регистра, значение которого мы запрашиваем

Ответ:

```
Opening /dev/ttyRS485-1 at 9600 bauds (N, 8, 2)
[02][03][00][80][00][01][85][D1]
Waiting for a confirmation...
<02><03><02><00><02><7D><85>
SUCCESS: read 1 of elements:
Data: 0x0002
```

Запись нового адреса

Записываем новый адреса устройства в регистр 128, используя функцию 0x06 (Write Single Register).

В примере используется широковещательный адрес 0. Использование примера в таком виде *изменит адрес на всех устройствах Wiren Board*, подключенных к порту /dev/ttyRS485-1. Чтобы этого не произошло — отсоедините другие устройства от шины.

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a0 -t0x06 -r128 2
```

Где 0 — широковещательный адрес, а 2 — адрес, который нужно задать.

Ответ:

```
Data to write: 0x2
Opening /dev/ttyRS485-1 at 9600 bauds (N, 8, 2)
[00][06][00][80][00][02][08][32]
Waiting for a confirmation...
ERROR Connection timed out: select
ERROR occurred!
```

Сообщение об ошибке возникает всегда, когда запись производится на специальный (широковещательный) адрес 0 (-a0). Теперь к устройству нужно обращаться по адресу 2.

Пример **неправильного** использования команды:

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a0 -t0x06 -r128
```

Здесь не указано значение, которое нужно записать в регистр адреса, поэтому устройство получит неизвестное значение.

Чтение сигнатуры устройства

Прочтем регистры релейного модуля WB-MR14 с адресом 1, содержащие сигнатуру (модель) устройства: WBMR14. Известно, что сигнатура хранится по адресу 200 и занимает 6 регистров.

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a1 -t0x03 -r200 -c 6
```

Ответ:

```
Opening /dev/ttyAPP1 at 9600 bauds (N, 8, 2)
[01][03][00][C8][00][06][44][36]
Waiting for a confirmation...
<01><03><0C><00><57><00><42><00><4D><00><52><00><31><00><34><D4><76>
SUCCESS: read 6 of elements:
Data: 0x0057 0x0042 0x004d 0x0052 0x0031 0x0034
```

В ответе мы получили шесть 16-битных значений, в каждом из которых содержится код одного ASCII-символа. Преобразуем их:

```
echo -e $(modbus_client -mrtu -pnone -s2 /dev/ttyRS485-1 -a1 -t0x03 -r200 -c 6 | grep Data | sed -e 's/. *Data://' -e 's/ 0x00/\\x/g')
```

Ответ:

```
WBMR14
```

Чтение версии прошивки

Прочтем версию прошивки из модуля с modbus-адресом 189. По адресу 250 хранится null-terminated строка максимальной длиной в 16 регистров. Прочтем 16 регистров, начиная с адреса 250, и преобразуем полученный шестнадцатеричный ответ в символьную строку:

```
echo -e $(modbus_client -mrtu -pnone -s2 /dev/ttyRS485-1 -a189 -t0x03 -r250 -c 16 | grep Data | sed -e 's/. *Data://' -e 's/ 0x00/\\x/g')
```

В результате выполнения команды получаем строку, например **1.3.1**.

Настройка параметров трансформаторов

Для настройки трансформаторов запишите нужные значения в регистры счётчика. Номера регистров смотрите в карте регистров счётчика.

В примере задаются параметры трёх трансформаторов, подключенных к первому каналу счётчика WB-MAP12E(H).

Трансформатор на фазе	Коэффициент трансформации	Фазовый сдвиг
L1	3001	501
L2	3002	502
L3	3003	503

Настройки записываются в память конкретного WB-MAP один раз:

```
$ modbus_client --debug -mrtu -pnone -b9600 -s2 /dev/ttyRS485-2 -a1 -t0x10 -r0x1460 3001 3002 3003 501 502 503
```

Включение реле релейного модуля

На модуле WB-MR14 включим реле с номером 6 (адреса регистров флагов начинаются с нуля, помним об этом!). Используем для этого команду 0x05 (Write Single Coil):

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a1 -t0x05 -r5 1
```

Ответ:

```
Data to write: 0x1
Opening /dev/ttyRS485-1 at 9600 bauds (N, 8, 2)
[01][05][00][05][FF][00][9C][3B]
Waiting for a confirmation...
<01><05><00><05><FF><00><9C><3B>
SUCCESS: written 1 elements!
```

Обратите внимание, утилита modbus_client при записи заменила 1 на 0x00FF, поскольку именно это значение служит для включения реле. Любое ненулевое значение будет заменено на 0x00FF, поэкспериментируйте.

Одновременное включение нескольких реле

Включим все нечетные реле и выключим все четные. Для этого используем функцию 0x0F (Write Multiple Coils). В модуле всего 14 реле, так что мы должны передать значения для 14 регистров с 0 по 13.

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a1 -t0x0F -r0 -c 14 255 0 255 0 255 0 255 0 255 0 255 0 255 0
```

Ответ:

```
Data to write: 0xff 0x00 0xff 0x00 0xff 0x00 0xff 0x00 0xff 0x00 0xff 0x00 0xff 0x00
Opening /dev/ttyRS485-1 at 9600 bauds (N, 8, 2)
[01][0F][00][00][00][0E][02][55][15][1A][97]
Waiting for a confirmation...
<01><0F><00><00><00><0E><D4><0F>
SUCCESS: written 14 elements!
```

Обратите внимание на структуру данных запроса:

- [01] — адрес
- [0F] — код функции Write Multiple Coils
- [00][00] — адрес первого регистра флагов для записи
- [00][0E] — количество элементов для записи (14)
- [02] — количество байт данных (14 бит помещаются в 2 байтах)

- [55][15] — 01010101 00010101 (первое реле — младший бит первого байта, 8 реле — старший бит первого байта, 9 реле — младший бит второго байта)
- [1A][97] — CRC16

А так же на структуру ответа:

- <01> — адрес
- <0F> — код функции Write Multiple Coils
- <00><00> — адрес первого регистра флагов для записи
- <00><0E> — количество записанных регистров флагов
- <D4><0F> — CRC16

Подробнее описание структуры данных запросов и ответов можно найти на странице [Протокол Modbus](#).

Настройка взаимодействия входов и выходов реле

Примеры смотрите в статье [Примеры настройки взаимодействия входов и выходов](#).

Работа с Modbus-устройствами Wiren Board без контроллера

Contents

Аппаратная часть

Подготовка к работе

Программы для работы по протоколу Modbus

Выбор порта

Работа с программой Rilheva Modbus Poll

Создание собственной конфигурации

Использование готовых конфигураций регистров для модулей Wirenboard

Файлы конфигураций регистров модулей Wirenboard

Работа с программой modbus_client

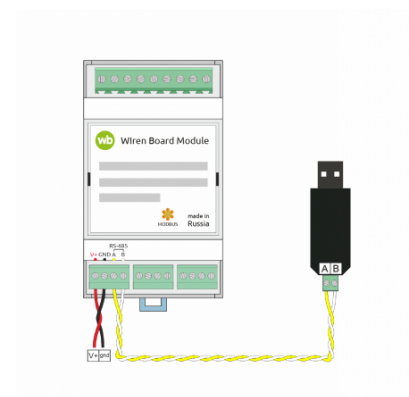


Схема подключения modbus-устройств через адаптер USB-RS485. Если у адаптера есть клемма GND — подключите её к клемме GND modbus-устройства

Аппаратная часть

Большинство устройств Wiren Board могут работать без управления контроллером — достаточно подать питание на клеммы

«V+» и «GND». Но для их настройки и считывания данных требуется подключиться к ним по протоколу Modbus. Для этого вы можете использовать компьютер с ОС Windows или Linux и адаптер USB-RS485.

Чтобы начать обмен с modbus-устройством, нужно клеммы **A** и **B** устройства подключить ко входам адаптера, подать на устройство питание и настроить программное обеспечение на компьютере.

Подготовка к работе

Перед настройкой устройства вам нужно знать:

1. modbus-адрес устройства,
2. коды функций чтения и записи регистров,
3. адреса регистров устройства.

Перечень общих для всех устройств Wiren Board регистров можно найти в таблице общих регистров. Полный список регистров для каждого устройства смотрите в документации к нему.

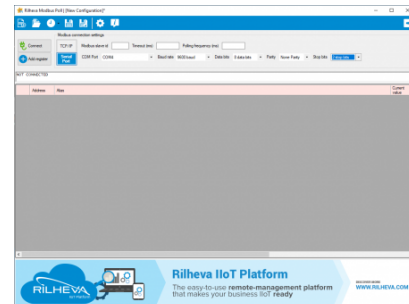
Программы для работы по протоколу Modbus

Для работы с устройствами по протоколу Modbus мы рекомендуем использовать программу **Rilheva Modbus Poll** (<https://www.rilheva.com/rilheva-modbus-poll-desktop-edition/>). Эта программа бесплатна, не требует установки, имеет понятный интерфейс и возможность сохранения настроек обмена и конфигурации регистров.

Программа написана для ОС Windows. Однако ОС Linux программа может быть запущена с помощью эмулятора wine (<https://www.winehq.org/>) (проверено на wine 7.0) и указанием пути к исполняемому файлу:

```
wine ~/RilhevaModbusPoll_1.0.7/RilhevaModbusPoll.exe
```

Также можно использовать утилиты Modbus Poll (<https://www.modbustools.com/download.html>) из комплекта Modbus Tools (программа платная, но с бесплатным пробным периодом в 30 дней) и termite (<http://s2-team.ru/wrkr/prods/modbus-tools/termite/>) (есть «Pro» и «Free» версия) в ОС Windows и утилиту modbus_client в ОС Linux.



Интерфейс программы Rilheva Modbus Poll

Выбор порта

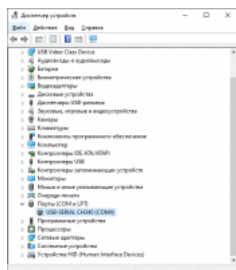
В ОС Windows узнать, к какому порту у вас подключен адаптер, можно в диспетчере устройств. Найдите в дереве тип устройств **Порты (COM и LPT)**, разверните ветку и найдите в ней свой адаптер. Если устройство выделено восклицательным знаком — это значит, что драйвер не был установлен автоматически. Установите его вручную по инструкции производителя.

В ОС Linux порт можно узнать с помощью команды команды `dmesg -wH`, которую нужно запустить перед физическим подключением USB-адаптера. В диагностических сообщениях будет выведено сообщение об обнаружении адаптера и назначении ему порта:

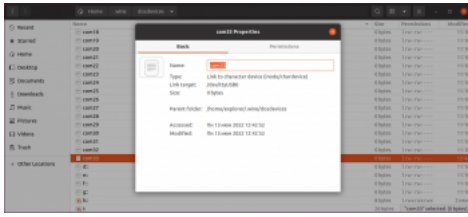
```
[ +7,705022] usb 3-7: new full-speed USB device number 6 using xhci_hcd
[ +0,149142] usb 3-7: New USB device found, idVendor=1a86, idProduct=7523, bcdDevice= 2.62
[ +0,000014] usb 3-7: New USB device strings: Mfr=0, Product=2, SerialNumber=0
[ +0,000005] usb 3-7: Product: USB2.0-Serial
[ +0,001500] ch341 3-7:1.0: ch341-uart converter detected
[ +0,000549] usb 3-7: ch341-uart converter now attached to ttyUSB0
```

В данном примере был назначен порт `/dev/ttyUSB0`.

В интерфейсе программы **Rilheva Modbus Poll** для адаптера обычно нужно выбрать COM-порт с самым большим номером. Чтобы проверить правильность выбора порта в Linux, нужно в пользовательской директории со списком портов эмулятора wine (`~/wine/dosdevices`) найти файл, который является символической ссылкой на реальный USB-порт, к которому подключен адаптер.



Определение номера COM-порта в диспетчере устройств ОС Windows



Определение номера COM-порта при работе через wine в ОС Linux

Работа с программой Rilheva Modbus Poll

Создание собственной конфигурации

Для начала работы с модулем по протоколу Modbus RTU в главном окне программы Rilheva Modbus Poll нужно выбрать подключение по последовательному порту, указать номер используемого порта и настроить параметры обмена: скорость, четность, количество стопбитов в соответствии с настройками модуля, а также таймаут ответа и частоту повторения запросов.

Затем добавьте регистры устройства, которые необходимо опрашивать. Нажмите кнопку **Add register** и в появившемся диалоговом окне заполните поля в соответствии с картой регистров используемого модуля:

- **Address** — адрес регистра в десятичном формате
- **Alias** - описание регистра
- **Read only** - запрет на запись значений в регистр. Для регистров, значения которых нужно изменять снимите галочку
- **Read function** - выбор типа регистра и соответствующей modbus-функции для его опроса
- **Type** - задает формат данных
- **Multiplier** - коэффициент, на который умножается значение регистра для удобного отображения
- **Offset** - значение, которое прибавляется к значению регистра для удобного отображения
- **Decimal places** - количество десятичных знаков при отображении
- **Add further registers with the same kind** - опция позволяет добавить сразу несколько регистров одного типа в список опроса

Запустите опрос устройства, для этого нажмите кнопку **Connect**.

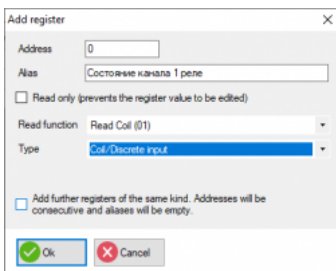
Чтобы изменить значение регистра типов holding или coil, задайте новое значение в поле **Current value**

Чтобы остановить опрос устройства, нажмите кнопку **Disconnect**.

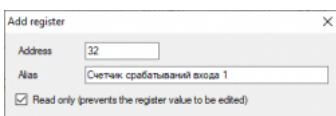
Рассмотрим работу с устройством по протоколу Modbus RTU на примере модуля реле WB-MR6C v.2. Считаем состояние выхода 1, счетчик срабатываний входа 1 и серийный номер модуля. Для этого в таблице регистров модуля реле находим данные этих регистров:

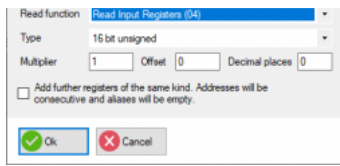
- состояние выхода 1 реле: адрес регистра 0, тип coil, формат bool
- счетчик срабатываний входа 1: адрес регистра 32, тип input, формат u16
- серийный номер модуля: адреса регистров 270, 271, тип input, формат u32

Задаем эти параметры в настройках регистров опроса программы и включаем опрос. Если параметры обмена и конфигурация регистров корректны, то в поле **Current value** отобразится содержимое регистров. При необходимости в дальнейшем использовать данную конфигурацию регистров сохраните ее.

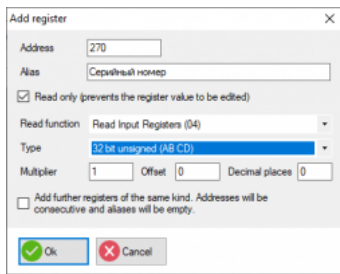


Конфигурирование опроса регистра состояния выхода 1 реле





Конфигурирование опроса регистра счетчика срабатываний входа 1



Конфигурирование опроса регистра серийного номера модуля

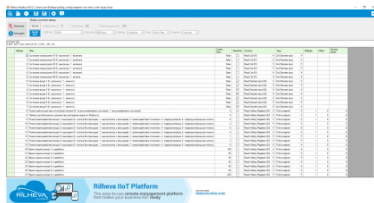


Опрос регистров устройства Wirenboard

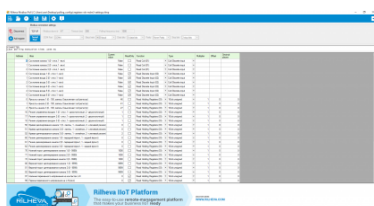
Использование готовых конфигураций регистров для модулей Wirenboard

Чтобы избавить пользователей от необходимости создавать шаблон опроса регистров самостоятельно мы подготовили файл с общими регистрами для всех устройств, а также файлы с набором основных регистров настройки для модулей реле WB-MR6xx, WB-MR3xx, модуля диммера WB-MDM3, модуля диммера светодиодных лент WB-MRGBW-D.

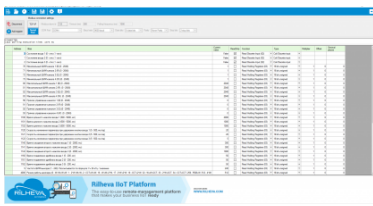
После запуска программы нужно выбрать файл конфигурации регистров соответственно модулю. Все настройки обмена будут заполнены для работы с параметрами модуля по умолчанию. Если какие-либо настройки устройства были изменены ранее, то установите соответствующие значения. Также нужно задать адрес устройства (у всех устройств он разный) и выбрать используемый COM-порт. Для начала опроса регистров нужно нажать кнопку **Connect**. Состояния регистров примут актуальные значения. При необходимости изменения значения регистра типа holding или coil задайте его в поле Current value. Для прекращения опроса регистров нужно нажать **Disconnect**.



Настройка модуля реле WB-MR6C



Настройка диммера WB-MDM3



Настройка диммера WB-MRGBW-D

Файлы конфигураций регистров модулей Wirenboard

Файл регистров, общих для всех Modbus-устройств Wiren Board

Файлы регистров и настроек для Modbus-модулей Wirenboard по типам:
[WB-MR6C v.2, WB-MR6LV/x, WB-MR6HV/x, WB-MR6C/NC \(6-канальный модуль реле\)](#)
[WB-MR3LV/x, WB-MR3HV/x, WB-MRWL3 \(3-канальный модуль реле\)](#)
[WB-MR6CU v.2, WB-MRPS6 \(6-канальный модуль реле\)](#)
[WB-MRGBW-D fw3 \(4-канальный диммер светодиодных лент с прошивкой 3.x\)](#)
[WB-MDM3 \(3-канальный диммер\)](#)

Работа с программой `modbus_client`

Для работы с modbus-устройством в ОС Linux также можно использовать консольную утилиту `modbus_client`.

Для этого скачайте пакет для настольных компьютеров с Linux (https://github.com/contactless/modbus-utils/releases/download/1.2/modbus-utils_1.2_amd64.deb).

Перейдите в папку со скаченным пакетом и установите его командой:

```
sudo apt install ./modbus-utils_1.2_amd64.deb
```

Также автоматически должен установиться пакет `libmodbus`, если этого не произошло — установите его из репозитория apt.

Как работать и примеры использования смотрите в статье [modbus_client](#).

Протокол Modbus

- [English](#)
- [русский](#)

Contents

[Основные понятия](#)

[Структуры данных Modbus](#)

[Модель данных Modbus](#)

[Адреса регистров](#)

[Нестандартная адресация](#)

[Пример описания регистров в документации](#)

[Коды функций чтения и записи регистров](#)

[Формат данных запросов и ответов Modbus](#)

[Коды исключений \(ошибки\) Modbus](#)

[Вычисление контрольной суммы Modbus](#)

Основные понятия

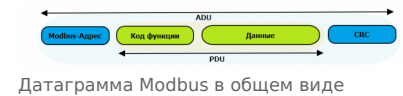
Modbus - это протокол прикладного (седьмого) уровня модели OSI. Чаще всего он служит для обмена данными между устройствами автоматизации и реализован в виде "протокола ответов на запросы (request-reply protocol)".

В устройствах Wiren Board данные Modbus передаются по последовательным линиям связи RS-485. В последовательных линиях связи протокол RS-485 полудуплексный и работает по принципу «клиент-сервер». Каждое устройство в сети (кроме ведущего см. далее) имеет адрес от 1 до 247, адрес 0 используется для широковещательной передачи данных всем устройствам, а адреса 248-255 считаются зарезервированными согласно спецификации Modbus, их использование не рекомендуется.

Существует две спецификации протокола: Modbus RTU и Modbus ASCII. В Modbus RTU передается 11-битный символ, состоящий из 1 стартового бита, 8 бит данных (начиная с младшего бита), бит четности (необязателен) и 2 стоповых бита - если бит четности не передается, или 1 стоповый бит - если бит четности передается. Такой символ передает 1 байт данных. В устройствах Wiren Board по умолчанию бит контроля четности не передается и используется 2 стоповых бита. В Modbus ASCII каждый байт передается двумя символами, представляющими ASCII-коды младшей и старшей четырехбитной группы байта (пример). Modbus RTU передает больше информации при той же скорости последовательной линии, и в устройствах Wiren Board используется именно он. Все дальнейшее описание относится к Modbus RTU.

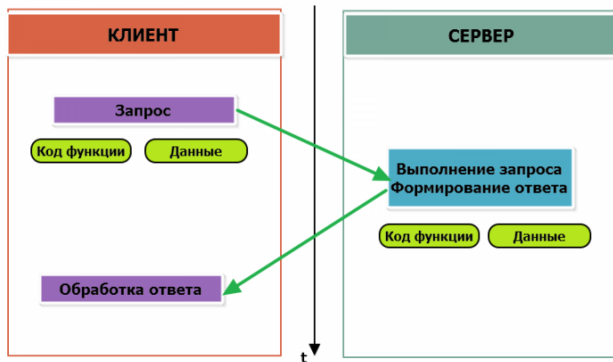
Ведущее устройство ("мастер", или "клиент") периодически опрашивает "ведомое", или "сервер". Ведущее устройство не имеет адреса, передача сообщений от устройства-сервера ведущему без запроса ведущего в протоколе не предусмотрена.

Пакет данных Modbus выглядит, как это показано на рисунке. **PDU** (Protocol Data Unit) — общая часть пакета MODBUS, включающая код функции и данные пакета. **ADU** (Application Data Unit) — полный пакет MODBUS. Включает в себя специфичную для физического уровня часть пакета и PDU. Для последовательных линий в заголовке ADU передается адрес устройства, а в конце — контрольная сумма CRC16. Максимальный размер ADU в последовательных коммуникационных линиях составляет **253 байта** (из максимальных, разрешенных спецификацией 256 байт вычитается 1 байт адреса и два байта контрольной суммы). Для справки — в Modbus TCP максимальная длина пакета составляет 260 байт.

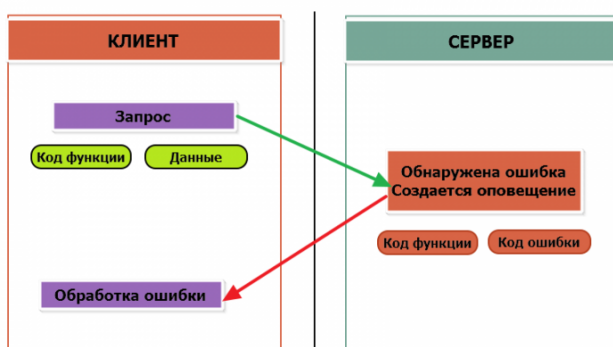


Датаграмма Modbus в общем виде

Функция кодируется одним байтом и определяет, какое действие должно выполнить устройство-сервер. Значение кодов функций лежат в диапазоне от 1 до 255, причем коды от 128 до 255 зарезервированы для сообщений об ошибках со стороны устройства-сервера. Код 0 не используется. Размер блока данных может варьироваться от нуля до максимально допустимого. Если обработка запроса прошла без ошибок, то устройство-сервер возвращает пакет ADU, содержащий запрошенные данные.



Modbus-транзакция, прошедшая без ошибок



При возникновении ошибки устройством возвращается код ошибки. При обычной транзакции код функции в ответе возвращается без изменений; при ошибке старший бит кода функции устанавливается в единицу (то есть *код функции* + 0x80). Так же есть таймаут ожидания ответа от ведомого устройства — бессмысленно долго ждать ответ, который, возможно, никогда и не придет.

Структуры данных Modbus

В Modbus принято кодировать адреса и данные в формате big-endian, то есть в формате, когда байты следуют, начиная со старшего: например, при передаче шестнадцатеричного числа 0x1234 сначала устройством будет принят байт 0x12, а затем — 0x34. Для передачи данных другого типа, например, чисел с плавающей запятой (float), текстовых строк, даты и времени суток и т.п. производитель может выбрать свой собственный способ кодирования — для расшифровки получаемых данных важно ознакомиться со спецификацией производителя устройства.

Модель данных Modbus

Обмен данными с Modbus-устройствами происходит через регистры. В протоколе Modbus определяется четыре типа регистров, показанных в таблице:

Таблица	Размер	Доступ
Регистры флагов (Coils)	1 бит	чтение и запись
Дискретные входы (Discrete Inputs)	1 бит	только чтение
Регистры хранения (Holding Registers)	16-битное слово	чтение и запись
Регистры ввода (Input Registers)	16-битное слово	только чтение

Регистры флагов (Coils) хранят однобитные значения - то есть могут находиться в состоянии 0 или 1. Такие регистры могут обозначать текущее состояние выхода (включено реле). Название "coil" буквально и означает обмотку-актюатор электромеханического реле. Регистры флагов допускают как чтение, так и запись.

Дискретные входы (Discrete Inputs) также являются однобитными регистрами, описывающими состояние входа устройства (например, подано напряжение — 1). Эти регистры поддерживают только чтение.

Регистры хранения (Holding Registers) и **регистры ввода (Input Registers)** представлены двухбайтовым словом и могут хранить значения от 0 до 65535 (0x0000 — 0xFFFF). Регистры ввода допускают только чтение (например, текущее значение температуры). Регистры хранения поддерживают как чтение, так и запись (для хранения настроек). В настоящее время во многих устройствах, в частности в устройствах Wipac Board, эти регистры не разделяются. Команды на чтение регистра хранения N и регистра ввода N обратятся к одному и тому же значению в адресном пространстве устройства.

Адреса регистров

Регистры в стандарте Modbus адресуются с помощью 16-битных адресов. Адресация начинается с нуля. Адрес регистра, таким образом, может принимать значения от 0 до 65535.

Адресные пространства регистров, также называемые таблицами или блоками, могут быть различны для всех четырёх типов регистров. Это значит, что значения регистров с одинаковым адресом, но разным типом, в общем случае разные.

Например, при чтении регистра флагов (coil) номер 42, регистра дискретного входа (Discrete), регистров ввода и хранения (Input и Holding) с теми же адресами, можно получить четыре разных значения.

Нестандартная адресация

В документации на некоторые, особенно старые, устройства адреса элементов (регистров) указываются в формате, не соответствующем стандарту. В этом формате тип элемента кодируется первой цифрой адреса, а адресация начинается не с нуля.

Например, регистр хранения с адресом 0 может записываться как 40001 или 400001, а Coil с адресом 0 как 000001.

В таблице перевода адресов в стандартный формат показаны диапазоны для двух разных нестандартных типов указания адресов и соответствующие им типы данных и диапазоны стандартных адресов.

Тип данных	Стандартные адреса	Стандартные адреса (hex)	Нестандартные адреса (5 цифр)	Нестандартные адреса (6 цифр)
Флагов (Coils)	0-65535	0x0000 - 0xFFFF	00001 - 09999	000001 - 065536
Дискретных входов (Discrete)	0-65535	0x0000 - 0xFFFF	10001 - 19999	100001 - 165536
Регистры входов (Input Registers)	0-65535	0x0000 - 0xFFFF	30001 - 39999	300001 - 365536
Регистры хранения (Holding Registers)	0-65535	0x0000 - 0xFFFF	40001 - 49999	400001 - 465536

признаки использования нестандартной адресации.

- Адреса записываются в десятичном формате
- Во всех адресах пять или шесть цифр
- Адреса с недискретными данными (показания датчиков и т.п.) начинаются на 30 или 40

Часто рядом с нестандартными адресами указываются и адреса соответствующие стандарту, обычно в шестнадцатеричном формате. Стоит отметить, что физически в пакете данных передаются адреса в стандартном формате, независимо от способа представления их в документации.

Пример описания регистров в документации

В готовых шаблонах устройств для контроллера Wiren Board есть шаблон для однофазного счетчика электроэнергии SDM220 (/usr/share/wb-mqtt-serial/templates/config-sdm220.json). В документации от производителя "Eastron SDM 220 Modbus Smart Meter Modbus Protocol Implementation V1.0" перечислены регистры и соответствующие им измеряемые параметры, например:

Address (Register)	Description	Units	Modbus Protocol Start Address Hex (Hi Byte Lo Byte)
30001	Line to neutral volts.	Volts	00 00
30007	Current.	Amps.	00 06
30013	Active power	Whatts	00 0C
30019	Apparent power	VoltAmps	00 12
...

Производитель в таблице приводит и логические, и физические адреса регистров, что позволяет нам с легкостью создать шаблон устройства и проиллюстрировать связь между логическими и физическими адресами Modbus-регистров.

```
"channels" : [
  {
    "name" : "Voltage",
    "type" : "voltage",
    "reg_type" : "input",
    "address" : "0x00",
    "format" : "float"
  },
  {
    "name" : "Current",
    "type" : "current",
    "reg_type" : "input",
    "address" : "0x06",
    "format" : "float"
  },
  {
    "name" : "Active Power",
    "type" : "power",
    "reg_type" : "input",
    "address" : "0x0c",
    "format": "float"
  },
  {
    "name" : "Apparent Power",
    "type" : "power",
    "reg_type" : "input",
    "address" : "0x12",
    "format": "float"
  }
],
```

Фрагмент шаблона счетчика SDM220

Коды функций чтения и записи регистров

В следующей таблице приведены наиболее распространенные коды функций Modbus:

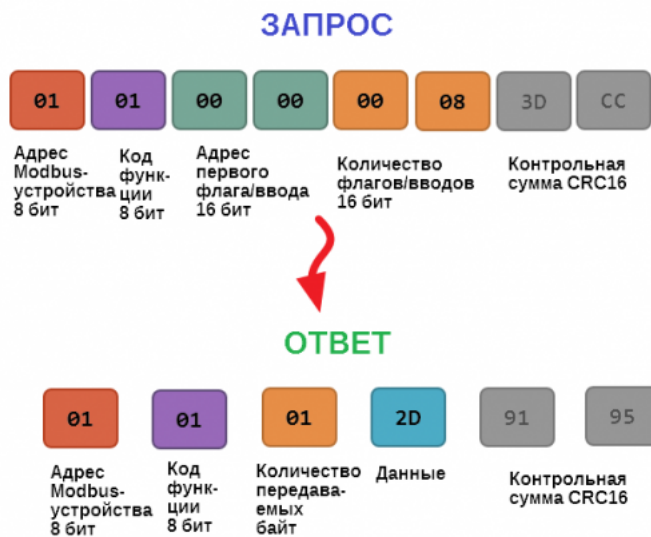
Код функции	HEX	Название	Действие
-------------	-----	----------	----------

1	0x01	Read Coils	Чтение значений нескольких регистров флагов
2	0x02	Read Discrete Inputs	Чтение значений нескольких дискретных входов
3	0x03	Read Holding Registers	Чтение значений нескольких регистров хранения
4	0x04	Read Input Registers	Чтение значений нескольких регистров ввода
5	0x05	Write Single Coil	Запись одного регистра флагов
6	0x06	Write Single Register	Запись одного регистра хранения
15	0x0F	Write Multiple Coils	Запись нескольких регистров флагов
16	0x10	Write Multiple Register	Запись нескольких регистров хранения

Команды условно можно разделить по типам: чтение значений — запись значений; операция с одним значением — операция с несколькими значениями.

Формат данных запросов и ответов Modbus

Рассмотрим подробнее, как происходит обмен данными между устройством-клиентом, отправляющим запрос, и устройством-сервером, отвечающим ему. На следующем рисунке показан обмен данными контроллера с устройством с адресом 0x01. Мы хотим прочесть 8 coil-регистров, начиная с первого.



Обмен данными в Modbus

В качестве данных мы получили шестнадцатеричное число 0x2D, то есть состояние восьми coil-регистров в двоичном виде такое: 0b10110100.

В следующей таблице приведены структуры данных запросов и ответов для основных функций Modbus.

Код функции	Запрос	Ответ
1 (Read Coils) и 2 (Read Discrete Inputs)	<ul style="list-style-type: none"> Адрес первого регистра флагов или входного регистра (16 бит) Количество данных (8 значений на байт) (16 бит) 	<ul style="list-style-type: none"> Число передаваемых байт (8 бит) Значения регистров флагов или входных регистров (8 значений на байт)
3 (Read Holding Registers) и 4 (Read Input Registers)	<ul style="list-style-type: none"> Адрес первого регистра (16 бит) Количество регистров, которые нужно прочесть 	<ul style="list-style-type: none"> Число передаваемых байт (8 бит) Значения регистров (16 бит на 1 регистр)
5 (Write Single Coil)	<ul style="list-style-type: none"> Адрес регистра (16 бит) Значение, которое нужно записать (0 — выключить, 0xFF00 — включить) 	Ответ аналогичен запросу
6 (Write Single Register)	<ul style="list-style-type: none"> Адрес регистра (16 бит) Новое значение регистра (16 бит) 	Ответ аналогичен запросу
15 (Write Multiple Coils)	<ul style="list-style-type: none"> Адрес первого регистра флагов для записи (16 бит) Количество регистров флагов для записи (16 бит) Количество передаваемых байт данных для регистров флагов (8 бит) Данные (8 регистров флагов на байт) 	<ul style="list-style-type: none"> Адрес первого coil-регистра (16 бит) Количество записанных coil-регистров (16 бит)
	<ul style="list-style-type: none"> Адрес первого регистра хранения для записи (16 бит) Количество регистров хранения для записи (16 бит) 	<ul style="list-style-type: none"> Адрес первого регистра хранения (16 бит)

16 (Write Multiple register)	бит) <ul style="list-style-type: none"> Количество передаваемых байт данных для регистров (8 бит) Данные (16 байт на регистр) 	<ul style="list-style-type: none"> Количество записанных регистров хранения(16 бит)
--------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------

Коды исключений (ошибки) Modbus

Если запрос не может по той или иной причине быть обработан устройством-сервером, то в ответ он отправляет сообщение об ошибке. Сообщение об ошибке содержит адрес Modbus-устройства, код функции, при выполнении которой произошла ошибка, увеличенный на 0x80, код ошибки и контрольную сумму:



Транзакция завершилась с ошибкой

В этом случае мы попытались обратиться к несуществующему адресу регистра 0xFFFF и попытались прочесть 8 регистров флагов. В результате мы получили код ошибки 0x03 — "В поле данных передано неверное значение".

Наиболее распространенные коды ошибок Modbus приведены в следующей таблице:

Код ошибки	Название ошибки	Что означает
1	Illegal Function	В запросе был передан недопустимый код функции
2	Illegal Data Address	Указанный в запросе адрес не существует
3	Illegal Data Value	Неверный формат запроса, например количество байт в запросе не соответствует ожидаемому. Примечание: несмотря на название, эта ошибка не говорит о том, что само значение регистра неправильное или ошибочное, и должна использоваться только для ошибок формата запроса.
4	Server Device Failure	Произошла невосстановимая ошибка на устройстве при выполнении запрошенной операции
5	Acknowledge	Запрос принят, выполняется, но выполнение потребует много времени; необходимо увеличить таймаут.
6	Server Device Busy	Устройство занято обработкой предыдущего запроса.
7	Negative Acknowledge	Устройство не может выполнить запрос, необходимо получить от устройства дополнительную диагностическую информацию. Возможно, требуется тех. обслуживание.
8	Memory Parity Error	Ошибка четности при обращении к внутренней памяти устройства.

Вычисление контрольной суммы Modbus

Для протокола Modbus RTU 16-битная контрольная сумма (CRC) вычисляется по алгоритму, описанному в спецификации Modbus, в документе "Modbus Serial Line Protocol and Implementation Guide", раздел "CRC-generation". Передающее устройство формирует два байта контрольной суммы на основе данных сообщения, а принимающее устройство заново вычисляет контрольную сумму и сравнивает с полученной. Совпадение принятой и вычисленной контрольной суммы Modbus RTU считается индикатором успешного обмена данными.

В случае ограниченных вычислительных ресурсов для вычисления контрольной суммы существует функция, использующая

В эту же группу точек в пользовательских ресурсах для выполнения контрольной суммы существует функция, использующая табличные значения (также приведена в спецификации).

Mapping-матрица

- English
- русский

Назначение

Релейные модули имеют несколько режимов работы реле от входов. И один из них — это взаимодействие согласно Mapping-матрице.

Режимы настраиваются индивидуально на каждый вход регистрами № 9-16, см. карту регистров модулей реле.

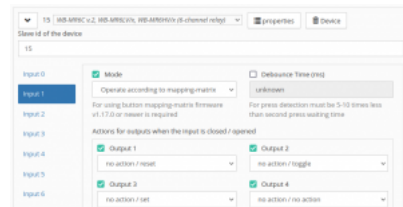
Матрица действий над выходами по изменению состояний входов позволяет гибко настраивать реакцию устройства в зависимости от изменения его входов.

Настроить Mapping-матрицы можно через запись значений в регистры или в веб-интерфейсе контроллера Wiren Board.

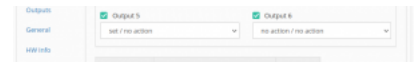
Классическая матрица

Классическая Mapping-матрица появилась в устройствах WB-MR и WB-MWAC с версии прошивки **1.9.0**.

Регистры	Выходы MR3-6							
	1	2	3	4	5	6	7	8
1	384	385	386	387	388	389	390	391
2	392	393	394	395	396	397	398	399



Входы	3	400	401	402	403	404	405	406	407
	4	408	409	410	411	412	413	414	415
	5	416	417	418	419	420	421	422	423
	6	424	425	426	427	428	429	430	431
	7	432	433	434	435	436	437	438	439
	0	440	441	442	443	444	445	446	447



Пример настройки mapping-матрицы в веб-интерфейсе контроллера Wiren Board

Mapping-матрица находится в регистрах начиная с адреса 384, и содержит 64 регистра из расчета 8 входов / 8 выходов. В каждом регистре записан код действия входа на выход. Если в устройстве количество входов и/или выходов меньше 8, то адреса регистров **не** меняются — просто строки/столбцы для отсутствующих регистров не учитываются. Исключение — устройства с 2 входами и выходами.

Регистры	Выходы MR2	
	1	2
Входы	1	384 385
	2	386 387

В ячейке на пересечении строки входа и столбца выхода — регистр действия входа на выход. В него записывается 16-битное слово **0b0000 0000 0000 уухх**, где биты **уу** описывают действия при замыкании входа (переход из 0 в 1, передний фронт), а биты **хх** — при размыкании (из 1 в 0, задний фронт).

Действия, в зависимости от битов:

- 00 — ничего не делать
- 01 — выключить
- 10 — включить
- 11 — инвертировать значение

Таким образом, можно запрограммировать реакцию каждого выхода на замыкание и размыкание любых входов.

Примеры программирования взаимодействия входов и выходов

		Задний фронт			
		00	01	10	11
Передний фронт	00	0000 (0) — Вход отключен, не управляет выходами	0001 (1) — Выключить при размыкании	0010 (2) — Включить при размыкании	0011 (3) — Изменить состояние выхода при размыкании
	01	0100 (4) — Выключить при замыкании	0101 (5) — Всегда выключать	0110 (6) — Работать как инвертированный выключатель с фиксацией	0111 (7) — Выключить при размыкании, потом изменить состояние при переключении
	10	1000 (8) — Включить при замыкании	1001 (9) — Работать как выключатель с фиксацией (повторять вход)	1010 (10, 0x0A) — Всегда включать	1011 (11, 0x0B) — Включить, потом изменить состояние при размыкании
	11	1100 (12, 0x0C) — Изменить состояние выхода при замыкании (выключатель без фиксации)	1101 (13, 0x0D) — Изменить состояние, потом выключить при размыкании	1110 (14, 0x0E) — Изменить состояние, потом включить при размыкании	1111 (15, 0x0F) — Изменить состояние при замыкании и вернуться к исходному состоянию при размыкании

		Биты хх — размыкание входа			
		00	01	10	
00					
	01				

Биты уу, замыкание входа	<p>(4) — Выключить при замыкании</p>	<p>(5) — Всегда выключать</p>	<p>(6) — Работать как инвертированный выключатель с фиксацией</p>	<p>(7) — Выключить и измениться при замыкании</p>
	<p>(8) — Включить при замыкании</p>	<p>(9) — Работать как выключатель с фиксацией (повторять вход)</p>	<p>(10, 0x0A) — Всегда включать</p>	<p>(11, 0x0B) — Всегда в состоянии</p>
	<p>(12, 0x0C) — Изменить состояние выхода при замыкании (выключатель без фиксации)</p>	<p>(13, 0x0D) — Изменить состояние, потом выключить при размыкании</p>	<p>(14, 0x0E) — Изменить состояние, потом включить при замыкании</p>	<p>(15, 0x0F) — Изменить состояние и вернуться при замыкании</p>

Матрицы для кнопок с распознаванием нажатий

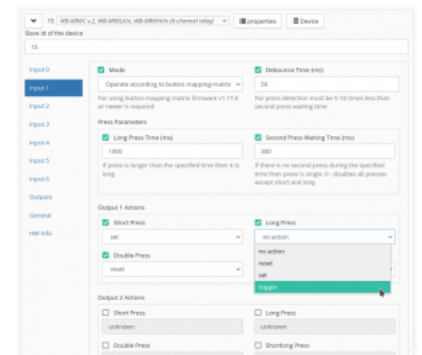
Начиная с версии прошивки 1.17.0 появились дополнительные матрицы с возможностью настроить входы как кнопки для детектирования четырёх типов нажатий:

- короткое,
- длинное,
- двойное
- и сначала короткое, затем длинное.

Настраиваемые действия аналогичны Mapping-матрице, также как и принцип выбора регистра на пересечении. Биты уу не используются.

Карты регистров для настройки матриц нажатий описаны ниже, про настройку можно почитать в статье Примеры программирования взаимодействия входов и выходов

Mapping-матрица коротких нажатий:



Пример настройки матрицы для кнопок в веб-интерфейсе контроллера Wiren Board

Регистры	Выходы MR3-6								
	1	2	3	4	5	6	7	8	
Входы	1	544	545	546	547	548	549	550	551
	2	552	553	554	555	556	557	558	559
	3	560	561	562	563	564	565	566	567
	4	568	569	570	571	572	573	574	575
	5	576	577	578	579	580	581	582	583
	6	584	585	586	587	588	589	590	591
	7	592	593	594	595	596	597	598	599
	0	600	601	602	603	604	605	606	607

Регистры	Выходы MR2		
	1	2	
Входы	1	544	545
	2	546	547

Mapping-матрица длинных нажатий:

Регистры	Выходы MR3-6								
	1	2	3	4	5	6	7	8	
Входы	1	608	609	610	611	612	613	614	615
	2	616	617	618	619	620	621	622	623
	3	624	625	626	627	628	629	630	631
	4	632	633	634	635	636	637	638	639
	5	640	641	642	643	644	645	646	647
	6	648	649	650	651	652	653	654	655
	7	656	657	658	659	660	661	662	663
	0	664	665	666	667	668	669	670	671

	6	648	649	650	651	652	653	654	655
	7	656	657	658	659	660	661	662	663
	0	664	665	666	667	668	669	670	671

Регистры		Выходы MR2	
		1	2
Входы	1	608	609
	2	610	611

Mapping-матрица двойных нажатий:

Регистры		Выходы MR3-6							
		1	2	3	4	5	6	7	8
Входы	1	672	673	674	675	676	677	678	679
	2	680	681	682	683	684	685	686	687
	3	688	689	690	691	692	693	694	695
	4	696	697	698	699	700	701	702	703
	5	704	705	706	707	708	709	710	711
	6	712	713	714	715	716	717	718	719
	7	720	721	722	723	724	725	726	727
	0	728	729	730	731	732	733	734	735

Регистры		Выходы MR2	
		1	2
Входы	1	672	673
	2	674	675

Mapping-матрица сначала коротких, потом длинных нажатий:

Регистры		Выходы MR3-6							
		1	2	3	4	5	6	7	8
Входы	1	736	737	738	739	740	741	742	743
	2	744	745	746	747	748	749	750	751
	3	752	753	754	755	756	757	758	759
	4	760	761	762	763	764	765	766	767
	5	768	769	770	771	772	773	774	775
	6	776	777	778	779	780	781	782	783
	7	784	785	786	787	788	789	790	791
	0	792	793	794	795	796	797	798	799

Регистры		Выходы MR2	
		1	2
Входы	1	736	737
	2	738	739

Modbus-адрес устройства Wiren Board

Contents

Общая информация

Определение адресов всех устройств на шине

Изменение адреса устройству с известным адресом

Изменение адреса устройству с неизвестным адресом

Восстановление доступа

Устройство питается от блока питания

Устройство питается от Vout контроллера

Полезные ссылки

Общая информация

Заводской Modbus-адрес устройства Wiren Board можно узнать на наклейке, которая находится на корпусе устройства.

Если заводской адрес был изменен, то можно воспользоваться одним из способов ниже, для работы вам понадобится утилита `Modbus_client`, которая доступна для контроллеров Wiren Board и компьютеров с ОС Linux. Если у вас компьютер с ОС Windows, то вы можете восстановить доступ к устройству.

ВНИМАНИЕ: если вы выполняете команды на контроллере, то перед началом работы остановите драйвер `wb-mqtt-serial`, а после окончания — запустите снова.



Modbus-адрес, установленный на производстве

Определение адресов всех устройств на шине

Если перебрать все доступные адреса и прочитать регистр с сигнатурой устройства — можно получить список устройств на шине:

1. Подключите устройства по шине RS-485 к контроллеру или другому оборудованию, где будете выполнять команды.
2. Если вы выполняете команды на контроллере:
 - откройте консоль контроллера по SSH,
 - остановите драйвер `wb-mqtt-serial`.
3. Замените в скрипте порт `/dev/ttyRS485-1` на тот, к которому подключены устройства, настройки соединения **9600N2** задаются параметрами `-b9600 -pnone -s2`:

```
for i in {1..247}; do echo -n "$i - "; D=`modbus_client -mrtu /dev/ttyRS485-1 --debug -b9600 -pnone -s2 -a$i -t3 -o100 -r200 -c6 2>/dev/null | grep Data: | awk 'gsub("Data:", "")' | sed -e 's/0x00/\\x/g' -e 's/\\s//g'; echo -e $D; done
```

4. Скопируйте и вставьте измененный скрипт в консоль контроллера, нажмите **Enter**.

Скрипт переберет все адреса с 1 по 247 и выведет в консоль результат для каждого адреса:

```
# for i in {1..247}; do echo -n "$i - "; D=`modbus_client -mrtu /dev/ttyRS485-1 --debug -b9600 -pnone -s2 -a$i -t3 -o100 -r200 -c6 2>/dev/null | grep Data: | awk 'gsub("Data:", "")' | sed -e 's/0x00/\\x/g' -e 's/\\s//g'; echo -e $D; done
1 -
2 -
3 -
4 -
5 -
6 - WBMWAC
7 -
8 -
9 - WBMRGB
10 -
11 -
12 -
...
```

Вариант скрипта с перебором не только адресов но и параметров связи: Если параметры подключения неизвестны

Изменение адреса устройству с известным адресом

Вы можете записать новый адрес в регистр 128(0x80):

1. Подключите устройство по шине RS-485 к контроллеру или другому оборудованию, где будете выполнять команды.
2. Если вы выполняете команды на контроллере:

- откройте консоль контроллера по SSH,
- остановите драйвер wb-mqtt-serial.

3. Чтобы назначить новый адрес 12 устройству с адресом 1 и подключенное к порту /dev/ttyRS485-1 выполните команду:

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a1 -t0x06 -r128 12
```

Пример успешного выполнения команды:

```
~# modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a1 -t0x06 -r128 12
Data to write: 0xc
Opening /dev/ttyRS485-1 at 9600 bauds (N, 8, 2)
[01][06][00][80][00][0C][88][27]
Waiting for a confirmation...
<01><06><00><80><00><0C><88><27>
SUCCESS: written 1 elements!
```

Изменение адреса устройству с неизвестным адресом

Если вам достаточно изменить адрес устройства, то вы можете сделать это отправив ему широковещательный запрос.

ВНИМАНИЕ: новый адрес будет установлен для всех устройств на шине, поэтому отключите те устройства, адреса которых вы не хотите менять.

Чтобы изменить адрес, выполните шаги:

1. Подключите устройство по шине RS-485 к контроллеру или другому оборудованию, где будете выполнять команды.
2. Если вы выполняете команды на контроллере:
 - откройте консоль контроллера по SSH,
 - остановите драйвер wb-mqtt-serial.
3. Замените в команде порт /dev/ttyRS485-1 на тот, к которому подключены устройства и выполните команду на контроллере:

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a0 -t0x06 -r128 1
```

Так как команда отправляет данные по широковещательному адресу — сообщение об ошибке в ответе является нормой.

Запишем всем устройствам на шине в регистр 128 (0x80) новый адрес 1:

```
modbus_client --debug -mrtu -pnone -s2 /dev/ttyRS485-1 -a0 -t0x06 -r128 1
Data to write: 0x1
Opening /dev/ttyRS485-1 at 9600 bauds (N, 8, 2)
[00][06][00][80][00][01][48][33]
Waiting for a confirmation...
ERROR Connection timed out: select
ERROR occured!
```

Восстановление доступа

Вы можете сбросить настройки приемопередатчика Modbus-устройства до заводских: скорость — 9600, чётность (parity) — N, количество стоп-бит — 2, Modbus-адрес — 1.

Это может быть полезно, если вам неизвестны все параметры подключения. Для сброса настроек используется утилита wb-mcu-fw-flasher, которая доступна для контроллеров Wiren Board, а также компьютеров с ОС Linux и Windows.

Устройство питается от блока питания

1. Подключите **только одно устройство** по шине RS-485 к контроллеру или другому оборудованию, где будете выполнять команды.
2. Если вы выполняете команды на контроллере:
 - откройте консоль контроллера по SSH,
 - остановите драйвер wb-mqtt-serial.
3. Отключите питание устройства.
4. Подайте питание на устройство и в течение двух секунд, пока устройство находится в режиме загрузчика, выполните команду, где /dev/ttyRS485-1 (COM1) — порт, к которому подключено устройство:

- на контроллере или компьютере с ОС Linux:

```
wb-mcu-fw-flasher -d /dev/ttyRS485-1 -a0 -u
```



Индикация режима загрузки

- на компьютере с ОС Windows перейдите в папку с утилитой, а потом выполните команду:

```
wb-mcu-fw-flasher_1.0.3.exe -d COM1 -a0 -u
```

5. Прошейте устройство новой прошивкой, или перезапустите, для этого отключите и включите питание устройства.

Пример успешного сброса настроек приемопередатчика:

```
root@wirenboard-A4DTZKTB:~# wb-mcu-fw-flasher -d /dev/ttyRS485-1 -a0 -u
/dev/ttyRS485-1 opened successfully.
Send reset UART settings and modbus address command...
Ok.
Device is in Bootloader now! To flash FW run
wb-mcu-fw-flasher -d <port> -f <firmware.wbfw>
```

Устройство питается от Vout контроллера

Если устройство питается от выхода *Vout* контроллера, то вы можете управлять его питанием программно. Этот способ доступен только для контроллеров Wiren Board.

1. Подключите **только одно устройство** по шине RS-485 к контроллеру.
2. Откройте консоль контроллера по SSH.
3. Остановите драйвер `wb-mqtt-serial`.
4. Выполните команду, которая перезагрузит устройство, подключенное к порту `/dev/ttyRS485-1` и сбросит настройки приемопередатчика:

```
mosquitto_pub -t '/devices/wb-gpio/controls/V_OUT/on' -r -m 0 && sleep 3 && mosquitto_pub -t '/devices/wb-gpio/controls/V_OUT/on' -r -m 1 && sleep 1 && wb-mcu-fw-flasher -d /dev/ttyRS485-1 -a0 -u
```

Пример успешного сброса настроек приемопередатчика:

```
~# mosquitto_pub -t '/devices/wb-gpio/controls/V_OUT/on' -r -m 0 && sleep 3 && mosquitto_pub -t '/devices/wb-gpio/controls/V_OUT/on' -r -m 1 && sleep 1 && wb-mcu-fw-flasher -d /dev/ttyRS485-1 -a0 -u
/dev/ttyRS485-1 opened successfully.
Send reset UART settings and modbus address command...
Ok.
Device is in Bootloader now! To flash FW run
wb-mcu-fw-flasher -d <port> -f <firmware.wbfw>
```

Полезные ссылки

- Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board
- Обновление прошивки Modbus-устройств Wiren Board
- Описание утилиты `modbus_client`
- Сервисная утилита `wb-mcu-fw-flasher`
- Описание драйвера `wb-mqtt-serial`

Карта регистров модулей реле

Описание

Общие регистры модулей реле

Адрес		Параметры регистра			Канал	Описание	Значения	FW
Dec	Hex	Тип	Доступ	Формат				
0	0x0000	Coil	RW	bool	Канал 1	Состояние канала реле	0 - разомкнут, 1 - замкнут	
1	0x0001				Канал 2			
2	0x0002				Канал 3			
3	0x0003				Канал 4			
4	0x0004				Канал 5			
5	0x0005				Канал 6			
0	0x0000	Discrete input	RO	bool	Вход 1	Состояние входа реле	0 - разомкнут, 1 - замкнут, Error: 0	
1	0x0001				Вход 2			
2	0x0002				Вход 3			
3	0x0003				Вход 4			
4	0x0004				Вход 5			
5	0x0005				Вход 6			
7	0x0007				Вход 0			
4	0x0004	Input	RW	u16		Статус внешнего источника питания (только для реле со встроенным блоком питания)	0: нет подключения внешнего источника питания 1: внешний источник питания подключен	1.18.4
5	0x0005	Holding	RW	u16		Служебный регистр, значение должно быть 0	0	
6	0x0006	Holding	RW	u16		Режим работы реле при отключении питания	0: не восстанавливать состояние реле, 1: восстанавливать состояние реле, 2: установить состояние реле согласно состоянию входа (только если настроен режим входа 1: выключатель с фиксацией) <i>До FW 1.18.0 были режимы 0 и 1</i>	1.5.3
8	0x0008	Holding	RW	u16		Таймаут для безопасного режима при пропадании связи по Modbus	с 0	
9	0x0009	Holding	RW	u16	Вход 1	Режим взаимодействия отдельного цифрового входа с соответствующим релейным выходом. В регистре 5 должно стоять значение по умолчанию (0).	0: кнопки без фиксации, 1: выключатель с фиксацией, 2: отключать все реле при нажатии, 3: отключить взаимодействие, 4: управлять по mapping-матрице, 5: не используется в реле 6: управлять по mapping-матрицам для кнопок <i>До FW 1.12.0 режимом по умолчанию был 0</i>	
10	0x000A				Вход 2			
11	0x000B				Вход 3			
12	0x000C				Вход 4			
13	0x000D				Вход 5			
14	0x000E				Вход 6			
16	0x0010				Вход 0			

							4: управлять по mapping-матрице, 6: управлять по mapping-матрицам для кнопок	
19	0x0013	Holding	RW	u16		Выбор причины перехода в безопасный режим. Только для WB-MR6C v.3	0: Потеря связи или пропадание низковольтного питания 1: Потеря связи 2: Пропадание низковольтного питания	1.18.4
20	0x0014	Holding	RW	u16	Вход 1	Время защиты входа от дребезга	мс 0 - 250 (50) 0 - 100 (50) до FW 1.17.8	1.13.0
21	0x0015				Вход 2			
22	0x0016				Вход 3			
23	0x0017				Вход 4			
24	0x0018				Вход 5			
25	0x0019				Вход 6			
27	0x001B				Вход 0			
32	0x0020	Input	RO	u16	Вход 1	Счетчик срабатываний входа	0 – 65535 по кругу Error: 0	
33	0x0021				Вход 2			
34	0x0022				Вход 3			
35	0x0023				Вход 4			
36	0x0024				Вход 5			
37	0x0025				Вход 6			
39	0x0027				Вход 0			
40	0x0020	Input	RO	u32	Вход 1	Целая часть значения частоты сигнала	$x1.52588 \cdot 10^{-05}$, Гц Error: 0	1.15.0
42	0x0028				Вход 2			
44	0x002A				Вход 3			
46	0x002C				Вход 4			
48	0x002E				Вход 5			
50	0x0030				Вход 6			
54	0x0032				Вход 0			
41	0x0029	Input	RO	u16	Вход 1	Дробная часть значения частоты сигнала 1 Гц = 65536 (2 ^ 16)	Error: 0	1.15.0
43	0x002B				Вход 2			
45	0x002D				Вход 3			
47	0x002F				Вход 4			
49	0x0031				Вход 5			
51	0x0033				Вход 6			
55	0x0037				Вход 0			
464	0x01D0	Input	RO	u16	Вход 1	Счётчик коротких нажатий	0 – 65535 по кругу	1.17.0
465	0x01D1				Вход 2			
466	0x01D2				Вход 3			
467	0x01D3				Вход 4			
468	0x01D4				Вход 5			
469	0x01D5				Вход 6			
471	0x01D7				Вход 0			
480	0x01E0	Input	RO	u16	Вход 1	Счётчик длинных нажатий	0 – 65535 по кругу	
481	0x01E1				Вход 2			
482	0x01E2				Вход 3			
483	0x01E3				Вход 4			
484	0x01E4				Вход 5			
485	0x01E5				Вход 6			
487	0x01E7				Вход 0			
496	0x01F0	Input	RO	u16	Вход 1	Счётчик двойных нажатий	0 – 65535 по кругу	
497	0x01F1				Вход 2			
498	0x01F2				Вход 3			
499	0x01F3				Вход 4			
500	0x01F4				Вход 5			
501	0x01F5				Вход 6			
503	0x01F7				Вход 0			
512	0x0200	Input	RO	u16	Вход 1	Счётчик короткого, а затем длинного нажатий	0 – 65535 по кругу	

513	0x0201				Вход 2			
514	0x0202				Вход 3			
515	0x0203				Вход 4			
516	0x0204				Вход 5			
517	0x0205				Вход 6			
519	0x0207				Вход 0			
123	0x007B	Input	RO	u16		Напряжение на микроконтроллере	mV Error: 0	1.16.0
124	0x007C	Input	RO	u16		Внутренняя температура микроконтроллера	x0.1, °C Error: 0	1.16.0
384 - 447	0x0180 - 0x01BF	Holding	RW	u16		Регистры mapping-матрицы	0	1.9.0
544 - 607	0x0220 - 0x025F	Holding	RW	u16		Регистры mapping-матрицы для коротких нажатий	0	1.17.0
608 - 671	0x0260 - 0x029F	Holding	RW	u16		Регистры mapping-матрицы для длинных нажатий	0	
672 - 735	0x02A0 - 0x02DF	Holding	RW	u16		Регистры mapping-матрицы для двойных нажатий	0	
736 - 799	0x02E0 - 0x031F	Holding	RW	u16		Регистры mapping-матрицы для сначала короткого, потом длинного нажатий	0	
930	0x03A2	Holding	RW	u16	Канал 1	Настройка безопасного состояния канала реле. Для всех сигнатур, кроме mg3, mg6, mg6c, mgw3	0: Контакты реле разомкнуты (реле отключено) 1: Контакты реле замкнуты (реле включено) <i>Для реле с контактами NO/NC речь здесь про контакты NO.</i>	1.18.4
931	0x03A3				Канал 2			
932	0x03A4				Канал 3			
933	0x03A5				Канал 4			
934	0x03A6				Канал 5			
935	0x03A7				Канал 6			
938	0x03AA	Holding	RW	u16	Канал 1	Действие канала при переходе в безопасный режим	0: Не реагировать 1: Перевести канал реле в безопасное состояние	1.18.4
939	0x03AB				Канал 2			
940	0x03AC				Канал 3			
941	0x03AD				Канал 4			
942	0x03AE				Канал 5			
943	0x03AF				Канал 6			
946	0x03B2	Holding	RW	u16	Канал 1	Влияние безопасного режима на управление с входов	0: Не влиять 1: Блокировать управление со входов при переходе в безопасный режим 2: Разрешить управление со входов только в безопасном режиме	1.18.4
947	0x03B3				Канал 2			
948	0x03B4				Канал 3			
949	0x03B5				Канал 4			
950	0x03B6				Канал 5			
951	0x03B7				Канал 6			
1100	0x044C	Holding	RW	u16	Вход 1	Время удержания входа в замкнутом состоянии для фиксации долгого нажатия	мс 500 - 5000 (2000)	1.17.0
1101	0x044D				Вход 2			
1102	0x044E				Вход 3			
1103	0x044F				Вход 4			
1104	0x0450				Вход 5			
1105	0x0451				Вход 6			
1107	0x0453				Вход 0			
1140	0x0474	Holding	RW	u16	Вход 1	Время двойного нажатия	мс 100 - 2000 (500)	
1141	0x0475				Вход 2			
1142	0x0476				Вход 3			

1143	0x0477			Вход 4	
1144	0x0478			Вход 5	
1145	0x0479			Вход 6	
1147	0x047B			Вход 0	

Общие для всех Modbus-устройств Wigen Board регистры

Адрес		Параметры регистра			Описание	Значения
Dec	Hex	Тип	Доступ	Формат		
104-105	0x0068 - 0x0069	Input	RO	u32	Время работы с момента загрузки	секунды
110	0x006E	Holding	RW	u16	Скорость порта RS-485. Настройка параметров подключения по RS-485	x100, Боды 12 — 1200 бит/с, 24 — 2400 бит/с, 48 — 4800 бит/с, 96 — 9600 бит/с , 192 — 19 200 бит/с, 384 — 38 400 бит/с, 576 — 57 600 бит/с, 1152 — 115 200 Кбит/с
111	0x006F	Holding	RW	u16	Настройка бита чётности порта RS-485	0 — нет бита чётности (none) , 1 — нечётный (odd), 2 — чётный (even)
112	0x0070	Holding	RW	u16	Количество стоп-битов порта RS-485	1, 2
120	0x0078	Holding	RW	u16	Перезагрузка устройства без сохранения состояния	любое, отличное от 0 перезагружает устройство
121	0x0079	Input	RO	u16	Текущее напряжение питания	мВ
128	0x0080	Holding	RW	u16	Modbus-адрес устройства (подробнее)	
129	0x0081	Holding	RW	u16	Перевод в режим обновления прошивки на 2 минуты	0 - выключен , >0 - включен
200-205	0x00C8 - 0x00CD	Input	RO	string	Модель устройства	
220-241	0x00DC - 0x00F1	Input	RO	string	Время и дата сборки прошивки	
220-248	0x00DC - 0x00F8	Input	RO	string	Хэш коммита и название ветки откуда собрана прошивка (2 символа в регистре)	
250-265	0x00FA - 0x0109	Input	RO	string	Версия прошивки	
266-269	0x010A - 0x010D	Input	RO	u64	Расширение серийного номера	
270-271	0x010E - 0x010F	Input	RO	u32	Серийный номер	
290-301	0x0122 - 0x012D	Holding	RO	string	Сигнатура прошивки	
330-336	0x014A - 0x0150	Holding	RO	string	Версия загрузчика	

Регистры настройки параметров обмена данными по RS-485 поддерживаются начиная с прошивки версии 1.6.0

Обновление прошивки Modbus-устройств Wiren Board

Contents

Общая информация

Автоматическое обновление

- Обновление всех устройств на шине
- Обновление одного устройства

Ручное обновление

- Особенности
- Подготовка устройства
- Загрузка прошивки в устройство

Восстановление прошивки устройства

- Автоматически
- Вручную

Полезные ссылки

Общая информация

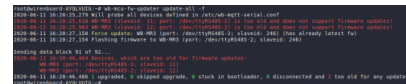
В наших modbus-устройствах реализован механизм загрузчика прошивок — bootloader. Он позволяет обновлять прошивки устройств и модулей Wiren Board по RS-485/Modbus RTU. В режиме загрузчика основные функции устройства отключаются, а коммуникационные параметры в режиме загрузчика фиксированы и не зависят от значений в памяти устройства: 9600 8N2.

Сами прошивки выпускаются в формате релизов, которые привязаны к релизам программного обеспечения контроллера. Это позволяет избежать ситуации, когда свежая версия прошивки без продолжительного тестирования попадает в очень ответственную инсталляцию.

Если ваши устройства подключены через шлюз WB-MGE или аналог, то для прошивки подключите их напрямую к контроллеру или используйте перенаправление socat, инструкция.

Автоматическое обновление

При обновлении прошивки удаляются ИК-команды, сохранённые в устройствах WB-MSW и WB-MIR. Рекомендуем сохранить банки команд перед обновлением с помощью скрипта.



Пример работы wb-fw-mcu-updater

Автоматическое обновление прошивки выполняется с помощью предустановленной на контроллере Wiren Board утилиты wb-mcu-fw-updater и позволяет установить свежую версию ПО сразу на все подключенные устройства или отдельно на каждое. Определение сигнатуры (модели) устройства, новой прошивки произойдет автоматически.

Утилита работает только на нашем контроллере и её нужен доступ в интернет, если у вас нет интернета или нашего контроллера, смотрите раздел Ручное обновление.

Обновление всех устройств на шине

Вы можете обновить все устройства, настроенные в разделе **Serial Devices Configuration** веб-интерфейса

1. Подключите устройства по шине RS-485 к контроллеру.
2. Настройте подключенные устройства в веб-интерфейсе.
3. Откройте консоль контроллера по SSH.
4. Обновите все настроенные устройства командой:

```
wb-mcu-fw-updater update-all
```

Обновление одного устройства

Чтобы обновить только одно устройство:

1. Подключите устройство по шине RS-485 к контроллеру.
2. Узнайте modbus-адрес устройства, которое хотите обновить.
3. Откройте консоль контроллера по SSH.
4. Запустите утилиту wb-mcu-fw-updater параметрами: ключ update-fw, а также порт и modbus-адрес.

Например, обновим прошивку устройства с modbus-адресом 70 и подключенного к порту /dev/ttyRS485-1:

```
wb-mcu-fw-updater update-fw /dev/ttyRS485-1 -a70
```

Полный список параметров и примеры работы смотрите на странице утилиты.

Ручное обновление

Особенности

Мы не рекомендуем этот способ, но если на объекте нет доступа в интернет, или у вас нет контроллера — это единственный вариант.

Ручное обновление можно сделать утилитой `wb-mcu-fw-flasher`, которую нужно предварительно установить. Способ установки отличается и зависит от используемой операционной системы и описан в документации.

Подготовка устройства

Прошивать устройства можно:

- по modbus-адресу устройства.
- по широковещательному адресу — 0.

Для прошивки нескольких устройств на шине нужно поочередно перевести их в режим загрузчика и прошить.

Загрузка прошивки в устройство

Для загрузки прошивки выполните шаги:

1. Подключите устройство по шине RS-485 к контроллеру или другому оборудованию, где установлена утилита прошивки.
2. Если вы выполняете команды на контроллере:

- Откройте консоль контроллера по SSH.
- Остановите драйвер `wb-mqtt-serial` или иное ПО, которое опрашивает устройство.

3. Скачайте из репозитория файл прошивки для вашего устройства по инструкции.
4. Загрузите файл прошивки на контроллер или другое устройство.
5. Перейдите в папку с файлом прошивки и выполните команду:

- на контроллере или компьютере с ОС Linux:

```
wb-mcu-fw-flasher -j -d /dev/ttyRS485-1 -a25 -f ./firmware.wbfw
```

- на компьютере с ОС Windows:

```
wb-mcu-fw-flasher_1.0.3.exe -j -d COM1 -a25 -f firmware.wbfw
```

6. Если вы выполняли команду с контроллера — запустите драйвер `wb-mqtt-serial`.

В команде выше мы флагом `-j` перевели устройство, подключенное к порту `/dev/ttyRS485-1` (COM1) с адресом 25 в режим загрузчика, а затем прошили его.

Успешный процесс прошивки выглядит так:

```
~# wb-mcu-fw-flasher -j -d /dev/ttyRS485-1 -a 25 -f mr6c_1.15.5_master_971fe50.wbfw
/dev/ttyRS485-1 opened successfully.
Send jump to bootloader command and wait 2 seconds...
Ok, device will jump to bootloader.
mr6c_1.15.5_master_971fe50.wbfw opened successfully, size 14720 bytes

Sending info block... OK

Sending data block 108 of 108... OK.
All done!
```

Если сигнатура устройства и файла прошивки не совпали, то вы получите сообщение об ошибке:

```
Sending info block...
Error while sending info block: Slave device or server failure
Data format is invalid or firmware signature doesn't match the device
```

Восстановление прошивки устройства

Если во время обновления произошел сбой, то устройство перейдет в режим загрузчика и вы можете восстановить его прошивку.

Автоматически

Для автоматического восстановления прошивки одного или нескольких устройств можно использовать утилиту `wb-mcu-fw-updater` в режимах **recover** и **recover-all**.

Чтобы восстановить устройство с адресом 10 и подключенное к порту `/dev/ttyRS485-1`, выполните команду:

```
wb-mcu-fw-updater recover /dev/ttyRS485-1 -a 10
```

Подробнее о режимах `recover` и `recover-all`, читайте в документации.

Вручную

Если вы не можете воспользоваться `wb-mcu-fw-updater`, то вы восстановить прошивку устройств можно с помощью сервисной утилиты `wb-mcu-fw-flasher`. Также этот способ могут использовать пользователи компьютеров с ОС Windows.

Для этого вам понадобится сама утилита и файл прошивки:

1. Подключите устройство по шине RS-485 к контроллеру или другому оборудованию, где установлена утилита прошивки.
2. Если вы выполняете команды на контроллере:
 - Откройте консоль контроллера по SSH.
 - Остановите драйвер `wb-mqtt-serial` или иное ПО, которое опрашивает устройство.
3. Скачайте из репозитория файл прошивки для вашего устройства.
4. Загрузите файл прошивки на контроллер или другое устройство, на котором установлена утилита прошивки.
5. Перейдите в папку с прошивкой и выполните команду:

- на контроллере или компьютере с ОС Linux:

```
wb-mcu-fw-flasher -d /dev/ttyRS485-1 -a 25 -f ./firmware.wbfw
```

- на компьютере с ОС Windows:

```
wb-mcu-fw-flasher_1.0.3.exe -d COM1 -a 25 -f firmware.wbfw
```

Здесь мы прошили находящееся в режиме загрузчика устройство с Modbus-адресом 25 и подключенное к порту `/dev/ttyRS485-1` (COM1) файлом `firmware.wbfw`.

Полезные ссылки

- Сброс Modbus-устройства Wiren Board к заводским настройкам
 - Modbus-адрес устройства Wiren Board
 - Утилита обновления и восстановления прошивок `wb-mcu-fw-updater`
 - Сервисная утилита `wb-mcu-fw-flasher`
 - Репозиторий прошивок для Modbus-устройств Wiren Board
-

Retrieved from "<https://wirenboard.com/wiki/Служебная:Print/>"

- Privacy policy
- About Wiren Board
- Disclaimers
-