

# 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](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) WriteSingle Register
(0x0F) WriteMultipleCoils, (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
--debug	Может указываться в любой позиции и включает отладку, выводя на экран шестнадцатеричные коды отправляемых и принимаемых данных.	нет	
-m	<p>Определяет тип используемого протокола:</p> <ul style="list-style-type: none"> <li>▪ <b>-mrtu</b> — Modbus RTU,</li> <li>▪ <b>-mtcp</b> — Modbus TCP.</li> </ul> <p>Он должен указываться первым в командной строке, или вторым, если первый аргумент — --debug или имя файла порта RS-485.</p>	да	
-a	Задаёт Modbus-адрес устройства, к которому мы обращаемся.	нет	1
-c	Определяет, какое количество элементов мы запрашиваем.	нет	1
-r	Задаёт начальный адрес для чтения или записи.	нет	100
-t	Указывает код функции Modbus. Кратко они перечислены в выводе modbus_client, подробнее значения кодов описаны на странице Протокол Modbus.	да	
-o	Задаёт таймаут в миллисекундах.	нет	1000
-0	Ноль. Уменьшает на единицу адрес, задаваемый аргументом -r. 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	<p>Контроль четности:</p> <ul style="list-style-type: none"> <li>▪ <b>-pnone</b> — нет проверки,</li> <li>▪ <b>-peven</b> — передается бит контроля на четность,</li> <li>▪ <b>-podd</b> — передается бит контроля на нечетность.</li> </ul>	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 Wipen 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 occured!
```

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:

## 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><02><55><15><1A><97>
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.

# Как зайти на контроллер Wiren Board по SSH

## Contents

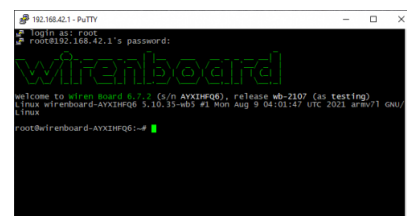
Введение

Логин и пароль

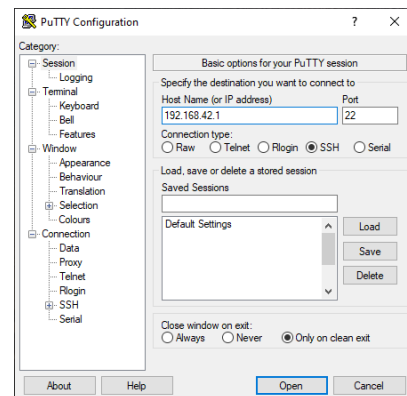
Программы

Windows

Linux



Консоль контроллера Wiren Board



Настройка SSH-соединения в программе PuTTY

## Введение

SSH — это протокол, при помощи которого можно получить доступ к консоли Wiren Board через локальную сеть или Интернет. Смотрите описание в Википедии ([http://en.wikipedia.org/wiki/Secure\\_Shell](http://en.wikipedia.org/wiki/Secure_Shell)).

Кроме SSH, получить доступ к консоли можно через Debug UART.

## Логин и пароль

Логин и пароль по умолчанию:

- Логин: **root**
- Пароль: **wirenboard**

## Программы

Windows

## Windows

Для операционной системы Windows, используйте бесплатную программу PuTTY.

## Linux

В операционной системе Linux, используйте PuTTY или просто выполните в консоли команду:

```
ssh root@192.168.42.1
```

Где 192.168.42.1 — IP-адрес контроллера, а root — имя пользователя. Если вы подключаетесь к контроллеру в первый раз, то система предложит принять сертификат — введите yes.

IP-адрес зависит от способа подключения и настроек контроллера. Подробнее читайте в статье [Как узнать IP-адрес контроллера](#).

---

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

wirenboard

## WB-MIR v1 - Modbus IR Remote Control

---

wirenboard

[https://wirenboard.com/wiki/WB-MIR\\_v1\\_-\\_Modbus\\_IR\\_Remote\\_Control](https://wirenboard.com/wiki/WB-MIR_v1_-_Modbus_IR_Remote_Control)  
14-12-2021 13:15

# Устройство IR-управления WB-MIR v.1

## Руководство по эксплуатации

Самая актуальная документация всегда доступна на нашем сайте по ссылке: [https://wirenboard.com/wiki/WB-MIR\\_v1\\_-\\_Modbus\\_IR\\_Remote\\_Control](https://wirenboard.com/wiki/WB-MIR_v1_-_Modbus_IR_Remote_Control)

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

## Содержание

[WB-MIR v1 - Modbus IR Remote Control](#)

[Устройство ИК-управления WB-MIR v.2](#)

[Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board](#)

[Инструкция по работе с ИК-приёмопередатчиком в устройствах WB-MSW, WB-MIR](#)

[Управление датчиками Wirenboard по протоколу Modbus](#)

[Утилита «modbus\\_client»](#)

[Как зайти на контроллер Wiren Board по SSH](#)

## WB-MIR v1 - Modbus IR Remote Control

---

Эта страница описывает устройство WB-MIR v.1, снятое с производства в 11.2017. Прямая замена - улучшенный модуль [WB-MIR v.2](#)

Обучаемый модуль с ИК-приёмопередатчиком для управления по ИК-каналу различными устройствами, включая кондиционеры, телевизоры и т.п. Управление модулем производится с контроллера или ПК по шине RS-485 командами по протоколу MODBUS.

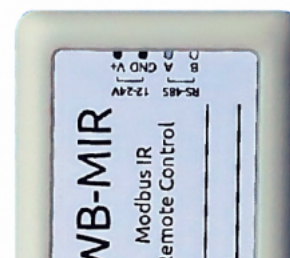
### Contents

---

[Общий принцип работы](#)

[Монтаж](#)

[Технические характеристики](#)



## Обучение и использование модуля

### Дополнительные Функции

#### Карты регистров

#### Сигнатура устройства

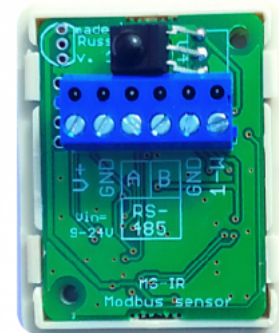
#### Управление модулем из веб-интерфейса

#### Управление модулем из командной строки

Пример чтения регистров



Модуль WB-MIR



Внутреннее устройство модуля. Видны подписи контактов на разъёмах



Отверстие ИК-передатчика на задней стороне модуля

## **Общий принцип работы**

Питание модуля подается на винтовые зажимы V+ и GND, линия RS-485 подключается ко входам A и B. Внешний датчик температуры DS18B20 подключается по паразитной схеме питания, когда выводы +5В и GND объединены и подключены к клемме GND, вывод данных подключается к зажиму 1-W. В некоторых моделях присутствует внутренний датчик температуры DS18B20 в корпусе TO-92, распаянный в верхней левой части платы модуля (вид сверху корпуса датчика нанесен в этом месте шелкографией).

## **Монтаж**

Модуль монтируется на кондиционер при помощи двустороннего скотча таким образом, чтобы небольшое отверстие излучающего диода с обратной стороны располагалось напротив ИК-приемника кондиционера, не закрывая приемник от основного пульта устройства. Положение подбирается экспериментальным путем ввиду огромного множества конструкций корпусов кондиционеров. То же относится и к любым бытовым устройствам, управляемым по ИК-каналу.

## **Технические характеристики**



Параметр	Значение
<b>Питание</b>	
Напряжение питания	9 — 24 В постоянного тока
Потребляемая мощность	0.1 Вт
Дальнейшее действие передатчика	< 1м
Внешние датчики	Возможность подключения цифрового датчика температуры DS18B20
<b>Управление</b>	
Интерфейс управления	RS-485
Изоляция интерфейса	Неизолированный
Протокол обмена данными	Modbus RTU, адрес задается программно, заводские настройки указаны на наклейке
Параметры интерфейса RS-485	<p>Скорость 9600 бит/с; данные — 8 бит; четность N; стоп-биты 2; Начиная с версии прошивки <b>3.1.0</b> параметры интерфейса могут быть настроены программно:</p> <ul style="list-style-type: none"> <li>▪ Скорость: 1200, 2400, 4800, 9600 (по умолчанию), 19200, 38400, 57600, 115200 бит/с (Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board)</li> <li>▪ Данные: 8 бит</li> <li>▪ Проверка чётности: нет (по умолчанию), 1 - нечётный (odd), 2 - чётный (even)</li> <li>▪ Стоповых бит: 2 (по умолчанию), 1</li> </ul>
<b>Габариты</b>	
Габариты	45x35x14 мм
<b>Условия эксплуатации</b>	
Температура воздуха	-40°C до +80°C
Относительная влажность воздуха	До 98%, без конденсата влаги

## Обучение и использование модуля

Подробное описание и руководство по использованию модуля, а также карта регистров, описаны в разделе [Инструкция по работе с ИК-приёмопередатчиком в устройствах WB-MS, WB-MSW, WB-MIR](#).

## Дополнительные Функции

**Смена адреса** производится широкоспешательный (slave\_id 0) командой записи (WRITE\_SINGLE\_REGISTER) в holding register с адресом 128 (0x80).

По адресу 200 лежит **сигнатура** длиной 6 байт. Сигнатура уникальна для каждой модели устройства и позволяет идентифицировать модель. Для получения сигнатуры нужно выполнить READ\_HOLDING\_REGISTERS адрес 200 длина 6. Запись в эту область памяти не поддерживается, при попытке записи в недопустимое место возвращается ошибка 3.

## Карты регистров

Подробно описаны в разделе [Инструкция по работе с ИК-приёмопередатчиком в устройствах WB-MS, WB-MSW, WB-MIR](#).

Регистр/адрес	Тип	Чтение/запись	Значение по умолчанию	Формат	Назначение	Версии прошивки
6	input	R	0x7FFF	°C × 16 (signed)	Температура с встроенного датчика 1-Wire	
7	input	R	0x7FFF	°C × 16 (signed)	Температура с внешнего датчика 1-Wire	
110	holding	RW	96	baud rate / 100	Скорость порта RS-485, <b>делённая на 100</b> . Допустимые скорости: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	>3.1.0
111	holding	RW	0		Настройка бита чётности порта RS-485. Допустимые значения: 0 - нет бита чётности (none), 1 - нечётный (odd), 2 - чётный (even)	
112	holding	RW	2		Количество стоп-битов порта RS-485. Допустимые значения: 1, 2	
120 (0x78)	holding	RW	0	отличное от 0	Запись в регистр вызывает перезагрузку модуля без сохранения состояния	
121 (0x79)	input	R	-	mV	Текущее напряжение питания модуля	

128 (0x80)	holding	RW	1		Modbus-адрес устройства
200-206	input	R	см. Сигнатура устройства		Сигнатура
220-241	input	R		__date__ __time__	Дата сборки прошивки
250-269	input	R		строка, null-terminated	Версия прошивки
270-271	input	R		32-bit unsigned int	Уникальный идентификатор (S/N)

## Сигнатура устройства

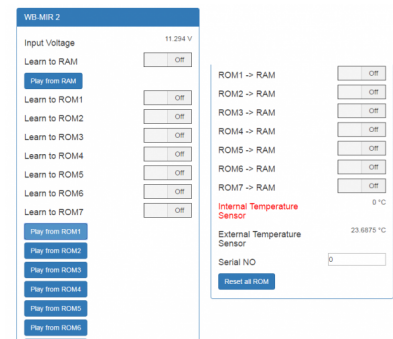
По адресу 200 лежит сигнатура модуля длиной 6 байт. Сигнатура уникальна для каждой модели устройств Wiren Board и позволяет идентифицировать модель по Modbus:

Модуль	Сигнатура
WB-MIR	WBMIR'0x00'

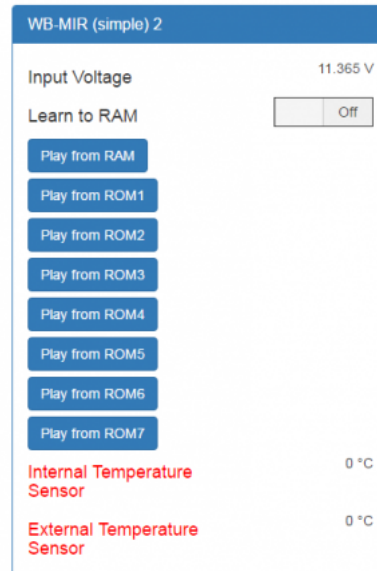
Для получения сигнатуры нужно выполнить команду READ\_HOLDING\_REGISTERS по адресу 200 (длина 6 регистров). Запись в эту область памяти не поддерживается, при попытке записи в недопустимое место возвращается ошибка 3. Пустые места в тексте сигнатуры забиваются значением 0x00. Начиная с регистра 220 записана дата сборки прошивки, в формате "число**00**время**00**" в виде строки (пример: 'Jan 27 2017 17:01:13'). Длина записи — 21 регистр.

## Управление модулем из веб-интерфейса

В настоящий момент для подключения модуля WB-MIR имеется два шаблона **WB-MIR** и **WB-MIR (simple)**. Шаблон **WB-MIR** предоставляет полный набор к функциям модуля из веб-интерфейса, включая обучение, а упрощенная версия **WB-MIR (simple)** может использоваться с уже обученным пультом и содержит гораздо меньшее количество элементов управления.



Веб-интерфейс модуля WB-MIR



Упрощенный веб-интерфейс модуля WB-MIR

На примере полнофункционального модуля рассмотрим основные органы управления модулем и их назначение.

Название элемента управления	Назначение
Input Voltage	Напряжение питания модуля
	Запись ИК-сигнала в оперативный буфер (см. раздел Оперативный режим). Для обучения контрол переводится в положение

Learn to RAM	Запись ИК-сигнала в оперативный буфер (см. раздел <a href="#">Оперативный режим</a> ). Для обучения контрол переводится в положение ON, после обучения — OFF.
Play from RAM	Воспроизведение ИК-сигнала из оперативного буфера (см. раздел <a href="#">Оперативный режим</a> ).
Learn to ROMx	Запись ИК-сигнала в соответствующую ячейку ПЗУ (см. раздел <a href="#">Постоянный режим</a> ). Для обучения контрол переводится в положение ON, после обучения — OFF.
Play from ROMx	Воспроизведение ИК-сигнала из соответствующей ячейки ПЗУ (см. раздел <a href="#">Постоянный режим</a> ). Для обучения контрол переводится в положение ON, после обучения — OFF.
ROMx -> RAM	Копирование данных из соответствующей ячейки ПЗУ в оперативный буфер для редактирования (см. раздел <a href="#">Постоянный режим</a> ). Для редактирования контрол переводится в положение ON, после редактирования — OFF.
Internal/External Temperature Sensor	Температура внутреннего/внешнего цифрового датчика. Если датчик отсутствует или неисправен, его название выделяется красным цветом.
Serial NO	Серийный номер модуля WB-MIR, может потребоваться при обращении в техподдержку.
Reset all ROM	Сброс содержимого всех ячеек памяти. <b>Будьте внимательны, очистка ПЗУ будет выполнена без запроса на подтверждение!</b>

## Управление модулем из командной строки

Программное обеспечение контроллера Wiren Board включает в себя утилиту `modbus_client` для работы с устройствами, подключенными к выходам RS-485, по протоколу Modbus. Подробное описание команды находится на странице [Modbus-client](#). Доступ к командной строке описан в статье [SSH](#).

### Пример чтения регистров

Проверим, верно ли, что мы подключились именно к модулю нужного типа. (Модуль `wb-mqtt-serial` должен быть предварительно остановлен). По адресу 200 хранится уникальная сигнатура длиной в 6 регистров. Прочтем 6 регистров, начиная с адреса 200, и преобразуем полученный ответ из шестнадцатеричных цифр в символьную строку:

```
echo -e `modbus_client --debug -mrtu -pnone -s2 /dev/ttyAPP1 \
-al -t0x03 -r200 -c 6 | \
grep Data | sed -e 's/0x00/\\x/g' -e 's/Data:/' -e 's/\\s//g'`
```

В результате выполнения команды получаем строку, например, **WBMIR**. В этом примере модуль WB-MIR имеет Modbus-адрес 1.

## Устройство ИК-управления WB-MIR v.2

- [English](#)
- [русский](#)

### Купить в интернет-магазине

Эта страница описывает устройство WB-MIR v.2, которое выпускается с 11.2017. Описание предыдущей модели: [WB-MIR v.1](#).

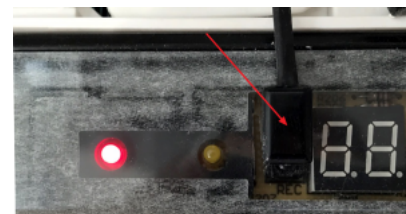
<b>Contents</b>
<a href="#">Назначение</a>
<a href="#">Технические характеристики</a>
<a href="#">Общий принцип работы</a>
<a href="#">Монтаж</a>
<a href="#">Представление в веб-интерфейсе контроллера WB</a>
<a href="#">Выбор шаблона</a>
<a href="#">Управление устройством и просмотр значений</a>
<a href="#">Настройка</a>
<a href="#">Способы настройки</a>
<a href="#">Обучение и использование модуля</a>
<a href="#">Режим работы цифрового входа</a>
<a href="#">Работа по Modbus</a>
<a href="#">Параметры порта по умолчанию</a>
<a href="#">Modbus-адрес</a>
<a href="#">Карта регистров</a>
<a href="#">Обновление прошивки и сброс настроек</a>



Модуль WB-MIR v.2. IR Learn — ИК-приёмник для записи команд

## Назначение

Обучаемый модуль с ИК-приёмопередатчиком для управления по ИК-каналу различными устройствами, включая кондиционеры, телевизоры и т.п. Управление производится с контроллера или ПК по шине RS-485 командами по протоколу Modbus RTU.



Пример монтажа ИК-излучателя модуля WB-MIR v.2 на приемнике кондиционера под декоративной крышкой

## Технические характеристики

Параметр	Значение
<b>Питание</b>	
Напряжение питания	9 — 28 В постоянного тока
Потребляемая мощность	0.1 Вт
Дальнодействие передатчика	< 1м
Внешние датчики	Можно подключить цифровой датчик температуры DS18B20.
<b>ИК-команды</b>	
Частота передатчика	38 кГц
Количество команд	<ul style="list-style-type: none"> <li>▪ Прошивки до версии <b>3.7.2</b> включительно: <b>7</b></li> <li>▪ Прошивки с версии <b>4.4.0: 40</b></li> <li>▪ Прошивки версии старше 3.7.2 и до 4.4.0 поддерживали до ~80 команд, но некорректно работали с некоторыми устройствами.</li> </ul>
Длительность команд	Максимальная длина команды - 508 регистров, плюс два регистра - признак окончания команды. Каждый регистр кодирует длительность высокого или низкого уровня сигнала (последовательно) в микросекундах.
<b>Управление</b>	
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
<b>Условия эксплуатации</b>	

Температура воздуха	От -40 до +80 °C
Относительная влажность	До 92 %, без конденсации влаги
<b>Клемники и сечение проводов</b>	
Рекомендуемое сечение провода с НШВИ	0.35 - 1 мм <sup>2</sup> — одинарные, 0.35 - 0.5 мм <sup>2</sup> - двоянные провода
Длина стандартной втулки НШВИ	8 мм
Момент затяжки винтов	0.2 Н•м
<b>Габариты</b>	
Д x Ш x В	40x15x11 мм; длина разъема кабеля передатчика добавляет +30 мм к ширине корпуса
Масса (с коробкой)	25 г

## Общий принцип работы

Записываете управляющий сигнал с пульта дистанционного управления устройством (ПДУ) в память модуля и потом воспроизводите его.

Для записи сигнала направьте ПДУ на ИК-приёмник модуля, который подписан как **IR learn**. Для передачи управляющего сигнала, подключите к входу **IR out** (3.5мм) внешний ИК-передатчик, который нужно разместить на ИК-приёмнике управляемого устройства.

К входу **1W** можно подключить внешний датчик температуры DS18B20 или изменить режим входа и использовать его как счётный вход. Сигнал с датчика удобно использовать для контроля за состоянием управляемых устройств с помощью правил [wb-rules](#) или сценариев [Node-Red](#). Например, можно настроить включение/отключение кондиционера в зависимости от температуры с датчика DS18B20.

Подробнее о записи и воспроизведении сигналов читайте в статье [Инструкция по работе с ИК-приёмопередатчиком](#).

## Монтаж

Модуль устанавливается рядом или внутри корпуса управляемого устройства, например, кондиционера. Если планируется дальнейшее обучение или переобучение WB-MIR v.2, то расположите модуль в доступном месте. ИК-передатчик, вынесенный на проводе длиной ~140 см, крепится рядом с ИК-приёмником управляемого устройства на двусторонний скотч (в комплекте). Не загораживайте ИК-приёмник устройства полностью, чтобы сохранить управление с пульта. Проверьте устойчивую передачу ИК-команд.

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](#).

Внешний датчик температуры DS18B20 подключается по паразитной схеме питания, когда выводы датчика +5В и GND объединены и подключены к клемме GND, вывод данных подключается к клемме 1W.

## 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-MIR v.2**.

### Управление устройством и просмотр значений

В веб-интерфейсе вы можете управлять выходами устройства и просматривать полученные с него значения. Список отображаемых каналов можно изменить через настройки устройства, доступные на [странице выбора шаблона](#).

Рассмотрим основные органы управления WB-MIR v.2 и их назначение.

Название элемента управления	Назначение
Input Voltage	Напряжение питания модуля
Learn to RAM	Запись ИК-сигнала в оперативный буфер (см. раздел <a href="#">Оперативный режим</a> ). Для обучения контрол переводится в положение ON, после обучения — OFF.
Play from RAM	Воспроизведение ИК-сигнала из оперативного буфера (см. раздел <a href="#">Оперативный режим</a> ).
Learn to ROMx	Запись ИК-сигнала в соответствующую ячейку ПЗУ (см. раздел <a href="#">Постоянный режим</a> ). Для обучения контрол переводится в положение ON, после обучения — OFF.
Play from ROMx	Воспроизведение ИК-сигнала из соответствующей ячейки ПЗУ (см. раздел <a href="#">Постоянный режим</a> ). Для обучения контрол переводится в положение ON, после обучения — OFF.
ROMx -> RAM	Копирование данных из соответствующей ячейки ПЗУ в оперативный буфер для редактирования (см. раздел <a href="#">Постоянный режим</a> ). Для редактирования контрол переводится в положение ON, после редактирования — OFF.
Internal/External Temperature Sensor	Температура внутреннего/внешнего цифрового датчика. Если датчик отсутствует или неисправен, его название выделяется красным цветом.

Serial NO	Сериинный номер модуля WB-MIR может потребоваться при обращении в техподдержку.
Reset all ROM	Сброс содержимого всех ячеек памяти. <b>Будьте внимательны, очистка ПЗУ будет выполнена без запроса на подтверждение!</b>

Веб-интерфейс WB-MIR v.2

## Настройка

### Способы настройки

Чтобы настроить модуль:

- укажите параметры в веб-интерфейсе контроллера Wiren Board: перейдите на [страницу настройки serial-устройств](#), выберите порт, найдите или добавьте устройство и измените параметры.
- запишите настройки в [Modbus-регистры](#) модуля:
  - в веб-интерфейсе контроллера Wiren Board — через [пользовательские параметры](#);
  - в консоли — с помощью утилиты [modbus\\_client](#);
  - если у вас нет контроллера Wiren Board, используйте [стороннее ПО](#).

### Обучение и использование модуля

ИК-приемник для обучения расположен прямо под красным круглым окном на корпусе модуля, подписанным “IR learn”

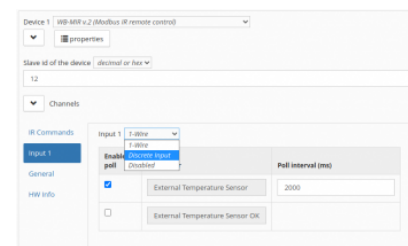
Подробное описание и использование WB-MIR v.2 приведены в разделе [Инструкция по работе с ИК-приёмопередатчиком в устройствах WB-MS, WB-MSW, WB-MIR](#).

### Режим работы цифрового входа

Вход **1W** устройства можно сконфигурировать как счётный вход, для этого нужно изменить режим в параметре **Inpit x**.

В версиях прошивки до 4.14.1 включительно, данные счетчиков срабатывания записываются в энергонезависимую память 1 раз в 600 сек.

Начиная с версии прошивки 4.15.0, алгоритм записи данных счетчика изменился. Если средняя частота срабатывания счетчиков меньше, чем 1 раз в 300 сек., то запись в память происходит с каждым изменением счетчика. Но если частота больше, то энергонезависимая память обновляется не чаще 1 раза в 300 сек. В частности, если счетчик меняется несколько раз в течение короткого времени (например в течение минуты), а потом продолжительное время не меняется (например в течение 5 часов), то после каждого изменения произойдет запись в память. Такой подход позволяет хранить актуальные данные счетчиков и гарантирует продолжительную работу энергонезависимой памяти.



Выбор режима для входа W1 в веб-интерфейсе контроллера Wiren Board

## Работа по Modbus

Устройства Wirenboard 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
8	Data bits	Количество битов данных
None	Parity	Бит чётности
2	Stop bits	Количество стоповых битов

При необходимости их можно изменить, смотрите инструкцию в статье [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 модуля WB-MIR v.2](#)

## Обновление прошивки и сброс настроек

Большинство устройств Wiren Board поддерживают обновление прошивки (микропрограммы) по протоколу Modbus. Это даёт возможность расширять функциональные возможности устройств и устранять ошибки в микропрограмме непосредственно на месте монтажа.

Инструкции:

- [Обновление прошивки](#)
- [Настройка параметров подключения](#)
- [Modbus-адрес: узнать, сбросить или изменить](#)
- [Сброс устройства к заводским настройкам](#)

Узнать о выходе новой версии прошивки можно в [Журнале изменений прошивок](#).

## Известные неисправности

Аппаратные ошибки/особенности [WB-MIR v.2](#), найденные при эксплуатации устройства.

## Ревизии устройства

Номер партии (Batch №) указан на наклейке, на боковой поверхности корпуса, а также на печатной плате.

Ревизия	Партии	Дата выпуска	Отличия от предыдущей ревизии
1.8	v.1.8K - ...	08.2021 - ...	▪ В качестве ИК-приемника IRM-56384
1.8	v1.8I, v.1.8J, v1.8L	06.2021 - 08.2021, 10.2021 - 11.2021	▪ на микроконтроллере GD32
1.8	1.8A - 1.8H	12.2019 - 06.2021	▪ Улучшена компоновка платы
1.6	338, 1.6A - 1.6C	03.2019 - 10.2019	▪ До 40 команд
1.6 - 1.7	277, 323?	до 02.2019	▪ До 80 команд (версии прошивки от 3.7.2 до 4.2.0), работает корректно не со всеми устройствами. ▪ С версии прошивки 4.0.0 можно использовать вход 1W как счётный.
1.6 - 1.7	238, 323?	до 02.2019	▪ Первая версия: до 7 команд (FW: 3.7.2)

# Изображения и чертежи устройства

---

Corel Draw 2018: [Файл:WB-Library.cdr.zip](#)

Corel Draw PDF: [Файл:WB-MIR.cdr.pdf](#)

Autocad 2013 DXF: [Файл:WB-MIR.dxf.zip](#)

Autocad PDF: [Файл:WB-MIR.pdf](#)

## Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board

- [English](#)
- русский

### Регистры параметров обмена данными по RS-485

---

В заводской конфигурации все устройства Wiren Board поставляются со следующими настройками RS-485: 9600 бит/с, бит чётности отсутствует (none), количество стоп-битов — 2. Однако в прошивках почти всех современных устройств Wiren Board поддерживаются дополнительные регистры настройки параметров обмена данными по RS-485:

Регистр / адрес	Тип	Чтение/ запись	Значение по умолчанию	Формат	Назначение
110	holding	RW	96	baud rate / 100	скорость порта RS-485, <b>делённая на 100</b> . Допустимые скорости: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
111	holding	RW	0		настройка бита чётности порта RS-485. Допустимые значения: 0 - нет бита чётности (none), 1 - нечётный (odd), 2 - чётный (even)
112	holding	RW	2		количество стоп-битов порта RS-485. Допустимые значения: 1, 2

### Изменение параметров обмена данными

---

Чтобы иметь возможность обратиться к устройству, поддерживающему протокол modbus RTU по шине RS-485, необходимо знать его modbus-адрес, а так же установленные на устройстве скорость обмена данными, четность, количество стоп-битов. **Важно: Перед выполнением команд, описанных ниже, остановите сервис `wb-mqtt-serial: service wb-mqtt-serial stop`**

Узнать текущие настройки параметров обмена данными, можно, прочитав значение указанных выше регистров, например:

```
echo -n '110 \*' | echo -e `modbus-client --debug -mrtu -b9600 -pnone -s? /dev/ttyAPP1 -a0x01 -t0x03 -r110 | grep Data | sed -e 's/Data:/' -e 's/s//g' | xargs
```



```
printf "%d" | xargs expr
```

```
9600
```

Подробнее смотрите описание команды `modbus_client`. **Важно: в примерах ниже используется порт `/dev/ttyAPP1`. Если устройство подключено к другому порту, необходимо заменить `/dev/ttyAPP1` на название этого порта!**

Записать новое значение можно с помощью следующей команды:

```
modbus_client --debug -mrtu -b9600 -pnone -s2 /dev/ttyAPP1 -a0x01 -t0x06 -r110 1152
```

Теперь устройство передает и принимает данные на скорости 115200 бит/с.

Записью в соответствующие регистры изменяется четность и количество стоп-битов.

При этом возникает парадоксальная ситуация: мы не можем узнать значение скорости, если мы заранее не укажем её значение при обращении к устройству! Поэтому стоит внимательно относиться к тому, какие коммуникационные параметры вы устанавливаете. Наклейте на устройство наклейку с новыми параметрами. В случае, если все коммуникационные параметры неизвестны, найти их можно только перебором:

```
#!/bin/bash
for l in {1,2};
do
  for k in {none,odd,even};
  do
    for j in {1200,2400,4800,9600,19200,38400,57600,115200};
    do
      for i in {1..247}; do
        modbus_client -mrtu /dev/ttyAPP1 --debug -o 300 -asi -t3 -r0x80 -b$j -s$l -p$k
        done 2>/dev/null | grep Data: | sed -e 's/ //g' -e 's/\n//' | xargs -I {} printf "Speed:$j\tStop bits:$l\tParity:$k\tModbus address:{}" | grep Data: | sed -e 's/Data:/'
      done
    done
  done
done
```

В этом скрипте мы обращаемся к регистру `0x80`, 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, то устройство не поддерживает установку коммуникационных параметров. В этом случае для коммуникации используется значение по умолчанию, 2 стоп-бита.

# Инструкция по работе с ИК-приёмопередатчиком в устройствах WB-MSW, WB-MIR

- [English](#)
- [русский](#)

## Contents

---

### Назначение

#### Принцип работы

Описание

Управление телевизором

Управление климатической техникой

Управление другой техникой по ИК

#### Способ представления и хранения сигнала

#### Управление модулем по Modbus

Обмен данными

#### Постоянный режим

Запись сигнала в постоянную память с пульта ДУ

Воспроизведение сигнала из постоянной памяти

Редактирование сигнала в постоянной памяти

Удаление всех сигналов из постоянной памяти

#### Оперативный режим

Запись сигнала в оперативную память с пульта ДУ

Воспроизведение сигнала из оперативной памяти

Удаление всех сигналов из постоянной памяти

Чтение/запись банков команд

Подготовка

Чтение команд в файлы

Запись команд из файлов

Карта регистров

Карта регистров флагов и входов (Coils и Inputs)

## Назначение

Инфракрасный приёмопередатчик, установленный в устройствах WB-MSW модификации I и WB-MIR, предназначен для записи сигналов с ИК-пультов ДУ бытовых устройств и дальнейшего управления бытовой техникой: кондиционерами, телевизорами и т.д. Модули WB-MIR v.2 и WB-MSW v.3 поддерживают около 80 ИК-команд (зависит от объема занимаемой в памяти).

**Внимание! Все модули, кроме WB-MSW v.3, обеспечивают управление возможно только с небольшого расстояния (< 1 метра).**

## Принцип работы

---

### Описание

Записываете управляющий сигнал с пульта дистанционного управления устройством в память модуля и потом воспроизводите его.

Вы можете записать сигнал в одну из двух видов памяти:

- Постоянную — записанные команды хранятся в ROM-буферах, которые записываются в ПЗУ модуля и сохраняются при отключении питания.
- Оперативную — это RAM-буфер модуля, данные теряются при отключении питания.

Помните, что при использовании постоянной памяти вы расходуете ресурс ПЗУ, каждую ячейку можно перезаписать не более 1000 раз.

Подробнее о записи и воспроизведении сигналов читайте в разделах [Постоянный режим](#) и [Оперативный режим](#).

### Управление телевизором

Обычно пульт телевизора работает в одиночном режиме — передаёт на устройство код нажатой клавиши. Поэтому, достаточно записать сигналы используемых кнопок пульта и вы сможете полностью управлять устройством. Для записи сигнала в модуль, направьте пульт на ИК-приёмник и нажмите нужную кнопку на пульте.

В итоге у вас получится в памяти модуля набор команд, которые соответствуют клавишам на пульте ДУ:

- Включить / выключить
- Прибавить громкость
- Убавить громкость
- Листать каналы вперёд
- Листать каналы назад
- Нажата кнопка 1

Также можно записывать целые режимы, например, включить 13 программу.

### Управление климатической техникой

Обычно пульт климатической техники передаёт на устройство набор команд, который соответствует выбранному на пульте режиму. Притом, состояние устройства (включено или выключено) может передаваться отдельно.

Например, чтобы управлять кондиционером, нужно сперва выставить нужный режим на пульте, а потом записать его в модуль. Для записи сигнала в модуль, направьте пульт на ИК-приёмник и измените один из параметров, например, температуру.

В итоге получится в памяти модуля набор готовых режимов:

- Выключен
- Включён, вентилятор на максимальную скорость, охлаждаем до 22 °С
- вентилятор на максимальную скорость, охлаждаем до 20 °С
- вентилятор на малую скорость, охлаждаем до 20 °С
- вентилятор на среднюю скорость, нагреваем до 27 °С

### Управление другой техникой по ИК

С помощью модуля вы можете управлять любой техникой, которая принимает команды по ИК. Для этого нужно выяснить: передаёт пульт одиночные сигналы на каждую нажатую клавишу или целиком режим и выбрать одну из описанных выше стратегий.

## Способ представления и хранения сигнала

Сигнал хранится в памяти устройства виде последовательности натуральных чисел (каждое от 0 до 65535), где каждое число - продолжительность логического 0 или 1 в сигнале в квантах по 10 микросекунд. Следовательно, максимальная продолжительность каждого нуля или единицы должна быть не более  $65535 * 10 \text{ мкс} = 655350 \text{ мкс} = 0,655 \text{ секунд}$ . Первое число последовательности — длительность первой логической единицы. Конец последовательности маркируется двумя периодами продолжительностью 0.

В оперативном режиме IR-сигнал (только один) находится в оперативной памяти устройства. Он записывается туда с пульта управления при подаче соответствующей Modbus-команды или путем записи значений в соответствующие регистры с внешнего устройства по Modbus (например, с контроллера Wiren Board), также IR-сигнал может быть скопирован из Flash-ПЗУ. IR-сигнал может быть воспроизведён из оперативной памяти и затем перезаписан; ресурс ПЗУ при этом не расходуется. При отключении питания сигнал в оперативной памяти не сохраняется.

В последних версиях прошивок модулей WB-MIR v.2 и WB-MSW v.3 последовательности команд сжимаются по специальному алгоритму, чтобы обеспечить более высокую плотность хранения. Перед воспроизведением последовательности разархивируются.

## Управление модулем по Modbus

**Внимание!** Номера используемых Modbus регистров менялись, начиная с версии прошивки **3.2.0**. В инструкции указаны новые номера регистров. Если вы пользуетесь устройством со старой прошивкой, общий принцип остаётся прежним, но номера регистров сдвигаются; как именно — смотрите в Карта регистров флагов и входов (Coils и Inputs). Полное описание регистров ИК-устройств можно найти в этой статье в разделе Карта регистров.

### Обмен данными

На физическом уровне модуль подключается через интерфейс RS-485. Для управления модулем используется протокол Modbus RTU. В устройствах Wirenboard данные Modbus передаются по линиям связи RS-485. Подробнее смотрите страницу Протокол Modbus. Modbus-адрес модуля задается на заводе и нанесен на наклейке на его боковой стороне. Адрес может быть изменен программно.

В устройствах с версией прошивки **3.1.0** и позднее также можно изменить настройки параметров обмена данными — смотрите таблицу регистров Modbus.

Управление ИК-приёмопередатчиком производится через регистры флагов Modbus; полный список можно найти в описании карты регистров. Есть 3 типа операций с сигналами:

- запись сигнала в память с пульта ДУ,
- воспроизведение сигнала из памяти,
- редактирование сигнала в оперативной памяти.

В каждый момент времени можно выполнять только одну операцию только над одним сигналом. Так как во время выполнения любой операции в один из регистров флагов ставится 1, а после завершения операции туда же ставится 0, в каждый момент времени среди всех регистров флагов, относящихся к ИК-приёмопередатчику, только один регистр может быть установлен в единицу. Попытка записи 1 в регистры флагов, связанных с ИК, при уже имеющейся единице приведёт к ошибке BUSY.

## Постоянный режим

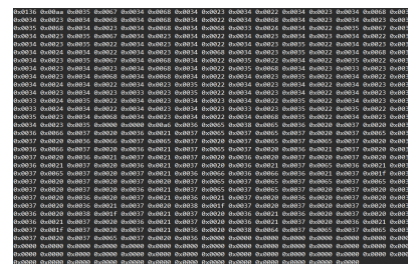
В постоянном режиме ИК-команды сохраняются в памяти устройства и доступны после перезагрузки по питанию. Каждая команда хранится в отдельном "банке" памяти. Количество банков памяти зависит от прошивки устройства. До прошивки 3.7.2 включительно устройства имели 7 банков памяти ИК-команд. Начиная с версии 4.4.0 поддерживается 32 банка. Промежуточные версии позволяли записывать до 80 команд, но в ряде случаев делали это некорректно.

### Запись сигнала в постоянную память с пульта ДУ

**Примечание:** Подробно о работе в режиме командной строки рассказано в разделе Управление модулем из командной строки.

Чтобы записать сигнал под номером  $i$  с пульта ДУ в память, запишите 1 в регистр флагов по адресу  $5300 + i$ . Затем направьте пульт ДУ на ИК-приёмник и нажмите кнопку на пульте. После отпускания кнопки, завершите запись сигнала, записав 0 в регистр флагов по адресу  $5300 + i$ .

```
#Начинаем запись сигнала с номером 1. Устройство подключено к порту /dev/ttyRS485-1 и имеет Modbus-адрес 16
root@wirenboard:~# modbus_client --debug -mrtu -b9600 -s2 -pnone /dev/ttyRS485-1 -a16 -t5 -r5301 1
#Направляем пульт на окно приемника и нажимаем кнопку на пульте ДУ
root@wirenboard:~# modbus_client --debug -mrtu -b9600 -s2 -pnone /dev/ttyRS485-1 -a16 -t5 -r5301 0
#Завершаем запись сигнала с номером 1
```



Пример содержимого оперативного буфера WB-MIR

## Воспроизведение сигнала из постоянной памяти

Чтобы воспроизвести сигнал под номером  $i$  из памяти, запишите 1 в регистр флага по адресу  $5100 + i$ . Если сигнал под номером  $i$  не был записан ранее, будет возвращена ошибка. Пока идёт воспроизведение сигнала, регистр флага по адресу  $5100 + i$  будет установлен в 1, а затем сам установится в 0. Пока идёт воспроизведение сигнала, никакие другие операции с ИК-приёмопередатчиком по Modbus не будут доступны.

```
#Отдаём команду воспроизвести сигнал с номером 1. Устройство подключено к порту /dev/ttyRS485-1 и имеет Modbus-адрес 16
root@wirenboard:~# modbus_client --debug -mrtu -b9600 -s2 -pnone /dev/ttyRS485-1 -a16 -t5 -r5101 1
```

## Редактирование сигнала в постоянной памяти

Чтобы отредактировать сигнал под номером  $i$  или записать его по Modbus (без пульта ДУ), запишите 1 в регистр флага по адресу  $5200 + i$ . После этого сигнал в виде последовательности натуральных чисел (смотрите рисунок "Пример содержимого оперативного буфера WB-MIR") будет записан в подряд идущие регистры хранения, начиная с регистра 2000. В конце последовательности будут два подряд идущих регистра хранения, содержащих 0. Если сигнал под номером  $i$  не был записан ранее, в регистрах хранения 2000 и 2001 будут находиться 0. Для внесения изменений в последовательность (в том числе для записи команды по Modbus) нужно записать правильную последовательность в данные регистры хранения с помощью команд WRITE\_SINGLE\_REGISTER и WRITE\_MULTI\_REGISTERS. В конце изменённой последовательности всё также должен быть маркер конца последовательности - два подряд идущих регистра хранения, содержащих 0. Чтобы удалить сигнал, запишите в первые два регистра хранения (2000 и 2001) 0. После окончания редактирования запишите 0 в регистр флагов по адресу  $5200 + i$ . До тех пор пока в регистре флагов по адресу  $5200 + i$  не будет стоять 0, никакие другие операции, в том числе и над другими сигналами, не будут доступны.

## Удаление всех сигналов из постоянной памяти

Чтобы удалить все записанные сигналы, запишите 1 в регистр флагов по адресу 5000.

## Оперативный режим

---

### Запись сигнала в оперативную память с пульта ДУ

Чтобы записать сигнал (только один) с пульта в оперативную память, не помещая при этом в ПЗУ - запишите 1 в регистр флага по адресу 5001. Далее действия происходят так же, как описано выше для записи сигнала в ПЗУ. После прекращения записи в регистрах хранения по адресу 2000 и дальше будет храниться сигнал в виде последовательности чисел. Эту последовательность можно редактировать, как описано выше. Внимание: содержимое этих регистров хранения не сохраняется при отключении питания устройства!

### Воспроизведение сигнала из оперативной памяти

Чтобы воспроизвести сигнал из оперативной памяти, запишите 1 в регистр флага по адресу 5002. Будет воспроизведён сигнал, записанный в регистрах хранения, начиная с 2000. В случае отсутствия маркера конца команды будет возвращена ошибка. Пока идёт воспроизведение сигнала, регистр флага по адресу 5002 будет установлен в 1, а затем сам установится в 0. Пока идёт воспроизведение сигнала, никакие другие операции с ИК-приёмопередатчиком по Modbus не будут доступны.

Управление ИК-приёмопередатчиком производится через регистры флагов Modbus (полный список можно найти в [#Карта регистров флагов и входов \(Coils and Inputs\)](#)). Есть 3 типа операций с сигналами:

- запись сигнала в память с пульта ДУ,
- воспроизведение сигнала из памяти,
- редактирование сигнала в памяти.
- в следующих версиях предполагается реализовать непосредственную запись из оперативного буфера в ПЗУ.

Как и в случае работы с регистрами постоянного хранения, в каждый момент времени можно выполнять только одну операцию только над одним сигналом. Так как во время выполнения любой операции в один из регистров флагов ставится 1, а после завершения операции туда же ставится 0, в каждый момент времени среди всех регистров флагов, относящихся к ИК-приёмопередатчику, только один регистр может быть установлен в единицу. Попытка записи 1 в регистры флагов, связанных с ИК, при уже имеющейся единице приведёт к ошибке BUSY.

### Удаление всех сигналов из постоянной памяти

Чтобы удалить все записанные сигналы, запишите 1 в регистр флагов по адресу 5000.

## Чтение/запись банков команд

---

Пользователь [rivcheg](#) написал скрипты чтения и записи IR-команд, которые [выложил на форуме](#).

Мы немного поправили им имена и опубликовали в архиве: [ir\\_backup.tgz](#)

## Подготовка

Содержимое архива:

Содержимое архива:

- `read_roms.pl` — скрипт сохранения ИК-команд в файлы
- `write_roms.pl` — скрипт записи ИК-команд из файлов в регистры датчика
- `compare_bufs.pl`, `getbuffer.pl`, `putbuffer.pl` — вспомогательные скрипты, запускаются автоматически.

Перед тем, как приступить к чтению и записи команд:

1. Зайдите в консоль контроллера по [SSH](#).
2. Перейдите в папку `/home`
3. Загрузите скрипт на контроллер и распакуйте его:

```
# wget -q0- https://wiringboard.com/wiki/images/1/19/Ir_backup.tgz | tar -xvz -C ./
ir_backup/
ir_backup/getbuffer.pl
ir_backup/write_roms.pl
ir_backup/putbuffer.pl
ir_backup/read_roms.pl
ir_backup/readme.txt
ir_backup/compare_bufs.pl
```

4. Перейдите в папку с распакованными файлами:

```
cd ir_backup
```

Настройки скриптов `read_roms.pl` и `write_roms.pl`:

- Скрипт читает/записывает только первые пять банков памяти: 0 — 4, если вам нужно считать/записать больше банков, то измените в скрипте значения 0..4.
- Скорость подключения и serial-порту можно указать в строке `modbus_client...`

## Чтение команд в файлы

Для чтения ИК-команд используется скрипт `read_roms.pl`.

Формат запуска скрипта:

```
./read_roms.pl directory modbus-address
```

Пример, в котором ИК-команды из устройства с адресом 22 сохраняются в папку `./roms`:

```
# ./read_roms.pl ./roms 22
Stopping wb-mqtt-serial
->1<->2<->3<->4<->5<-
Starting wb-mqtt-serial
```

## Запись команд из файлов

Для записи ИК-команд используется скрипт `write_roms.pl`.

Формат запуска скрипта:

```
./write_roms.pl directory modbus-address
```

Пример, в котором ИК-команды из папки `./roms` записываются в устройство с адресом 22:

```
# ./write_roms.pl ./roms/ 22
Stopping wb-mqtt-serial
->1<->2<->3<->4<->5<-
Starting wb-mqtt-serial
```

## Карта регистров

### Карта регистров флагов и входов (Coils и Inputs)

Начальный адрес	Тип	Количество	Назначение	В модификациях
5000	coil	1	Очистка всех банков ИК-команд	
5001	coil	1	Запись ИК-команды с ИК-приёмника в Modbus регистры хранения (2000-2509) (RAM), без сохранения в банк команд	
5002	coil	1	Воспроизведение ИК-команды из Modbus регистров хранения (2000-2509) (RAM)	

5100 (в прошивке 3.1.0 и ранее - 0)	coil	number_of_banks	Воспроизведение ИК-команды из соответствующего банка ROM (Play)	I
5200 (в прошивке 3.1.0 и ранее - 1000)	coil	number_of_banks	Редактирование ИК-команд (ROM) с копированием в RAM (ROM -> RAM)	I
5300 (в прошивке 3.1.0 и ранее - 2000)	coil	number_of_banks	Запись в банк ИК-команд с использованием IR-приёмника (Learn)	I
5400	input	number_of_banks	Размер данных IR-команды в байтах	I

В последних версиях прошивок значение number\_of\_banks равняется 80, то есть, например, регистры размера команд имеют значения от 5400 до 5479.

## Управление датчиками Wirenboard по протоколу Modbus

- English
- русский

### Contents

#### Карта регистров датчиков WB-MS и WB-MSW

- Карта регистров хранения (Holding Registers)
- Карта регистров флагов (Coils)
- Сигнатура устройства

## Карта регистров датчиков WB-MS и WB-MSW

---

### Карта регистров хранения (Holding Registers)

Регистр/ адрес	Тип	Чтение/ запись	Значение по умолчанию или при ошибке	Формат	Назначение	Под
						WB-MS (код модели)
0	input	R	0x7FFF	°C × 10 (signed)	Температура с встроенного датчика	T
1	input	R	0xFFFF	%RH × 10 (signed)	Относительная влажность с встроенного датчика	H
2	input	R	-	лк	<i>(регистр не используется в датчиках WB-MSW, вместо него регистры 9 и 10)</i> Освещённость с встроенного датчика	L
3	input	R	-	дБ × 100	Уровень шума, умноженный на 100	S
4	input	R	0x7FFF	°C × 100 (signed)	Температура с встроенного датчика	+
5	input	R	-	%RH × 100 (signed)	Относительная влажность с встроенного датчика	+
6	input	R	0x7FFF	°C × 16 (signed)	Температура с внешнего датчика	D
7	input	R	0x7FFF	°C × 16 (signed)	Температура с внешнего датчика	D
8	input	R	0xFFFF	PPM	Концентрация CO2	-
9-10	input	R		лк	<b>Освещенность, умноженная на 100</b> (9 - старший разряд, 10 — младший разряд значения освещенности)	-
11	input	R	0xFFFF	ppb	Качество воздуха	-
					<b>Служебный регистр,</b> отсутствует в большинстве устройств.	



86	input	R	0	0 или 1	Текущий канал работы датчика уровня шума: 0 - канал низкоуровневых шумов, 1 - канал высокоуровневых шумов	
87	input	R	0		Служебный регистр HOLD_REG_SPL_HIGHGAIN_RAW_INT	
88	holding	RW	0	ppm (400-1500)	<b>Служебный регистр,</b> отсутствует в большинстве устройств. Ручная калибровка датчика CO2. Для калибровки запишите сюда текущее значение концентрации CO2, измеренное другим прибором.	-
89	holding	RW	0xFF	дней	<b>Служебный регистр,</b> отсутствует в большинстве устройств. Длительность цикла для алгоритма самокалибровки датчика CO (ABC)	-
90	holding	RW	20 (200 ms)	× 10 ms	Время усреднения освещённости	L, B
91	holding	RW	20 (200 ms)	× 10 ms	Время усреднения шума	S
92	holding	RW	0	×0.0625	Служебный регистр HOLD_REG_SPL_RAW_OFFSET - сдвиг значения АЦП	-
93	holding	RW	0	×0.1dB	Служебный регистр HOLD_REG_SPL_OFFSET - поправка к значению уровня шума	-
95	holding	RW	1	1 or 0	Режим автокалибровки датчика CO2 (ABC) на атмосферный уровень CO2	-
96	holding	R	2000 / 5000	ppm	(доступно с версии прошивки 3.5.0) Диапазон измерения концентрации CO2. Допустимые значения: 2000, 5000, 10000.	-
97	holding	RW	0	0 — 10 с	Период между вспышками сигнальных светодиодов	-
98	holding	RW	0	0 — 50 мс	Длительность вспышки светодиодов	-
99	holding	RW	16	1 = 0.0625C	(доступно с версии прошивки 3.10.0) Фильтр подозрительных значений для датчиков 1-Wire (85.000C, 127.937C). Значения отбрасываются, если предыдущее отличается от подозрительного больше, чем на значение регистра*0.0625C. Запишите 0, чтобы отключить фильтр.	D
100	input	R			Служебный регистр HOLD_REG_SPL_RAW_INT	-
101	input	R			Служебный регистр HOLD_REG_TH_READS	
102	input	R			Служебный регистр HOLD_REG_TH_ERRORS	
103	holding	RW			Служебный регистр HOLD_REG_ILLUMINANCE_MODE	+
104-105	input	R		32-bit unsigned int	Время работы устройства (uptime counter) в секундах	
106	input	R			Служебный регистр INPUT_REG_SGPC3_BASELINE	-
107	input	R			Служебный регистр INPUT_REG_SGPC3_RAW_SIGNAL	-
108	input	R	0xFFFF		Служебный регистр SGPC3 VERSION	-
110	holding	RW	96	baud rate / 100	скорость порта RS-485, <b>делённая на 100</b> . Допустимые скорости: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 (Настройка параметров обмена данными по RS-485 для modbus-устройств Wiren Board)	

111	holding	RW	0		настройка бита чётности порта RS-485. Допустимые значения: 0 - нет бита чётности (none), 1 - нечётный (odd), 2 - чётный (even)	
112	holding	RW	2		количество стоп-битов порта RS-485. Допустимые значения: 1, 2	
113	holding	RW	8	мс	Задержка перед отправкой ответного пакета по RS-485	
120 (0x78)	holding	RW	0	отличное от 0	запись в регистр вызывает перезагрузку модуля без сохранения состояния	
121 (0x79)	input	R	-	mV	текущее напряжение питания модуля	
128 (0x80)	holding	RW	1		Modbus-адрес устройства	
129 (0x81)	holding	RW	0	0 или 1	Переход в режим загрузчика	
200-206	input	R	см. Сигнатура устройства		сигнатура	
220-241	input	R		__date__ __time__	дата сборки прошивки	
245	holding	RW	0	°C × 100 (signed)	Температурная компенсация самонагрева для датчика температуры и влажности (значение вычитается из измеренной температуры)	+
250-269	input	R		строка, null-terminated	версия прошивки	
270-271	input	R		32-bit unsigned int	уникальный идентификатор (S/N)	
280	input	R			Максимальное значение датчика движения за установленное время окна (Max motion)	-
281	input	R			Служебный регистр: сырое значение с АЦП датчика движения	-
282	holding	RW	10	с	Ширина временного окна для вычисления максимального усредненного значения датчика движения (от 1 до 60 секунд)	-
283	input	R			Текущее усредненное значение движения в условных единицах (Current motion)	-

Входы 1-W устройства можно сконфигурировать как счётные входы:

Регистр/адрес	Тип	Чтение/запись	Значение по умолчанию	Формат	Назначение	Версии прошивки
275	holding	RW	0	0 или 1	Режим входа №1: 0 - 1-wire, 1 - дискретный вход	>=4.0.0
276	holding	RW	0	0 или 1	Режим входа №2: 0 - 1-wire, 1 - дискретный вход	
277	input	R		16-bit unsigned int	Счетчик срабатываний для входа №1	
278	input	R		16-bit unsigned int	Счетчик срабатываний для входа №2	
0	discrete	R		0 или 1	Текущее состояние входа №1: 0 - разомкнут, 1 - замкнут на GND. Если для входа выбран режим 1-wire, значение всегда равно 0.	
1	discrete	R		0 или 1	Текущее состояние входа №2: 0 - разомкнут, 1 - замкнут на GND. Если для входа выбран режим 1-wire, значение всегда равно 0.	

## Карта регистров флагов (Coils)

Регистры, связанные с ИК-управлением, описаны в статье WB-MSx\_Consumer\_IR\_Manual.

Начальный адрес	Количество	Назначение	В модификациях	С версии прошивки
0	1	Включение пищалки (buzzer)	WB-MSW v.3, WB-MSW2	
1	1	Принудительная калибровка встроенного датчика CO2. Включать после 20 минут работы при уровне CO2, равном 400 ppm	WB-MSW v.3, WB-MSW2	
2	1	Включение подогрева (heater) в микросхеме сенсора температуры и влажности HDC1080	WB-MS (2017 г. и новее), WB-MSW v.3, WB-MSW2,	3.2.0
3	1	Включение датчика CO2 (1-вкл, 0-выкл)	WB-MSW v.3	4.1.0
10	1	Включение красного светодиода	WB-MSW v.3	
11	1	Включение зеленого светодиода	WB-MSW v.3	

## Сигнатура устройства

По адресу 200 хранится сигнатура модуля длиной 6 байт. Сигнатура уникальна для каждой модели устройств Wiren Board и позволяет идентифицировать модель по Modbus:

Модуль	Сигнатура
WB-MS	WBMS,'0x00','0x00'
WB-MSW	WBMSW,'0x00'
WB-MSW2	WBMSW2
WB-MSW v.3	WBMSW3

Для получения сигнатуры устройства нужно выполнить команду READ\_HOLDING\_REGISTERS по адресу 200 (длина 6 регистров). Запись в эту область памяти не поддерживается, при попытке записи в недопустимое место возвращается ошибка 3. Пустые места в тексте сигнатуры забиваются значением 0x00. Начиная с регистра 220 записана дата сборки прошивки, в формате "число**00**время**00**" в виде строки (пример: 'Jan 27 2017 17:01:13'). Длина записи — 21 регистр.

- 
- Privacy policy
  - About Wiren Board
  - Disclaimers
  -