

GSM/GPRS

Contents

Общая информация

Получение информации о модеме

Включение и начало работы

Утилита `wb-gsm`

Переключение активной sim-карты

Низкоуровневая работа по `uart`

Подключение в `linux`

Отправка AT-команд

Работа с `sms` и `ussd`

Настройка

Примеры команд `gammu`

SMS и USSD на русском

SMS-уведомления

Интернет через PPP

Быстрый выход в интернет

Автоматический запуск подключения

Автоматическое восстановление подключения

Резервирование канала связи

Интернет с 4G-модемом (LTE)

Настройка модема как сетевой карты

Настройка виртуальной сетевой карты

Включение

Отключение

Мультиплексирование

Документация на модемы

GPRS на модемах SIM7000E 2G/NB-IoT

Общая информация

В контроллеры Wigen Board могут быть установлены 2G/3G/4G(LTE)/NB-IoT модемы — зависит от комплектации.

С помощью модемов можно:

- отправлять и принимать SMS,
- подключаться к интернету по протоколу PPP для 2G- и 3G-модемов, а также настраивать 4G-модем как сетевую карту с выходом в интернет.

Модемы подключаются к процессору по `uart` и `usb`, исключение — модемы 2G. Подробнее о подключении модемов и низкоуровневом взаимодействии с ними можно почитать в разделе Низкоуровневая работа по `uart`.

Управление питанием и переключением активных sim-карт (если их две) производится с помощью `grio`, процесс описан в разделе Переключение активной sim-карты.

Включение и отключение модема рекомендуем делать с помощью утилиты `wb-gsm`.

Получение информации о модеме

В контроллерах, начиная с 2019 года, некоторая информация о модеме заносится в память. Получить её можно с помощью чтения файлов в директории `/proc/device-tree/wigenboard/gsm/`.

В контроллерах версии 6.7 модем устанавливается модулем расширения. После физического подключения модема его нужно добавить в конфигурацию контроллера:

1. В веб-интерфейсе перейдите в раздел **Settings** → **Configs** → **Hardware Modules Configuration**, выберите **Modem slot**.
2. В раскрывающемся списке **Module type** выберите тип установленного модуля.
3. Нажмите кнопку **Save**. Контроллер включит нужные для работы модема порты.

Для удаления модуля выберите тип **None**.

После конфигурирования включите модем командой `wb-gsm on` и настройте подключение к оператору связи.

Например, чтобы узнать модель модема, нужно выполнить команду

```
cat /proc/device-tree/wirenboard/gsm/model
```

Подробнее о файлах внутри директории можно узнать из таблицы:

| Файл | Описание |
|---|-------------------------|
| /proc/device-tree/wirenboard/gsm/model | модель модема |
| /proc/device-tree/wirenboard/gsm/type | поддерживаемые сети |
| /proc/device-tree/wirenboard/gsm/status | статус модема в системе |

Включение и начало работы

Чтобы начать работу с модемом, нужно:

1. Вставьте sim-карту.
2. Подключите антенну.
3. Перезапустите модем, выполнив команду

```
wb-gsm restart_if_broken
```

После каждой смены sim-карты необходимо перезапускать модем.

Утилита wb-gsm

Для упрощения работы с модемами была написана утилита `wb-gsm`, которая входит в пакет `wb-utils` (<https://github.com/wirenboard/wb-utils>).

Утилита `wb-gsm` входит в состав пакета `wb-utils`, который предустановлен на все контроллеры Wiren Board.

С помощью `wb-gsm` вы сможете:

- управлять питанием модемов, команды `on`, `off`, `restart_if_broken`;
- настраивать baudrate связи по `uart`, команды `init_baud`, `set_speed`;
- получать `imei`, команда `imei`.

Все команды `wb-gsm` можно посмотреть в репозитории на Github по ссылке в начале раздела.

Пример использования утилиты `wb-gsm` для получения `imei` модема, флаг `DEBUG=true` — выводить отладочную информацию:

```
DEBUG=true wb-gsm imei
```

Переключение активной sim-карты

По умолчанию активна `Sim1` — в каждый момент времени **только одна sim-карта может быть активной**.

Переключить модем на другую sim-карту можно с помощью `gpio` процессора. Узнать его номер можно двумя способами:

- **выполнить команду**

```
echo $WB_GPIO_GSM_SIMSELECT
```

- найти `SIM Slot Select gpio` в таблице `gpio` контроллера.

По умолчанию, этот `gpio` уже экспортирован в `sysfs`, поэтому, для переключения активной sim-карты с 1 на 2, нужно выполнить команду (в примере, номер `gpio` для переключения sim-карт - 88):

```
echo 1 > /sys/class/gpio/gpio88/value
```

Соответственно, для переключения обратно на `sim1`, нужно записать 0.

Подробнее о работе с `gpio` можно узнать из статьи [Работа с GPIO](#).

Для того чтобы новая sim-карта стала активной, нужно **выполнить следующие AT-команды** (см. раздел о работе с AT-командами):

```
AT+CFUN=0  
AT+CFUN=1
```

Низкоуровневая работа по uart

Любое взаимодействие с модемом так или иначе сводится к отправке AT-команд через последовательный порт модема. Все модемы подключаются к порту `/dev/ttyGSM`. 3G модемы, помимо этого, подключаются к портам `/dev/ttyACM*` (порты создаются usb-драйвером `cdc_acm`).

Подключение в linux

- uart: Порт `/dev/ttyGSM` является ссылкой на `/dev/ttyMX` (uart процессора) и создается с помощью правил `udev`. Конечный порт может быть разным для разных версий контроллера (подробнее можно посмотреть на нашем [github](https://github.com/wirenboard/wb-configs/tree/master/configs/usr/share/wb-configs/udev) (<https://github.com/wirenboard/wb-configs/tree/master/configs/usr/share/wb-configs/udev>)).
- usb: Порты `/dev/ttyUSB*` или `/dev/ttyACM*` (в случае 3G-модема) появляются автоматически после подачи питания на модем. Обычно, порты 3G-модема — это `/dev/ttyACM0-6`, однако **точно** определить, к каким портам модем подключен по USB можно, выполнив команды:

```
wb-gsm restart_if_broken; dmesg | tail
```

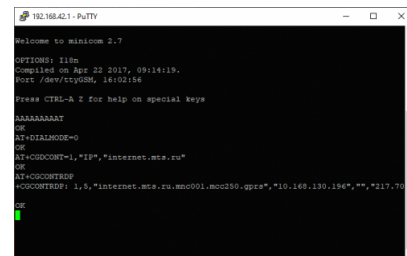
Примерный вывод команды:

```
[ 6102.978383] usb 2-1.2: New USB device found, idVendor=1e0e, idProduct=0020
[ 6102.985653] usb 2-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 6102.993108] usb 2-1.2: Product: SIMCOM_PRODUCT
[ 6102.997728] usb 2-1.2: Manufacturer: SIMCOM_VENDOR
[ 6103.002644] usb 2-1.2: SerialNumber: 004999010640000
[ 6103.002093] cdc_acm 2-1.2:1.0: ttyACM0: USB ACM device
[ 6103.008228] cdc_acm 2-1.2:1.2: ttyACM1: USB ACM device
[ 6103.116769] cdc_acm 2-1.2:1.4: ttyACM2: USB ACM device
[ 6103.132688] cdc_acm 2-1.2:1.6: ttyACM3: USB ACM device
[ 6103.145157] cdc_acm 2-1.2:1.8: ttyACM4: USB ACM device
[ 6103.163705] cdc_acm 2-1.2:1.10: ttyACM5: USB ACM device
[ 6103.182338] cdc_acm 2-1.2:1.12: ttyACM6: USB ACM device
```

Соответственно, в данном случае 3G модем подключен к портам `/dev/ttyACM0 - /dev/ttyACM6`.

Отправка AT-команд

| Параметры соединения по умолчанию | | |
|-----------------------------------|-----------------------|---|
| Значение | Параметр | Описание |
| Auto-bauding | Baud rate | Скорость, бит/с. В настройках программы подключения установите 115200. После подключения — отправьте модему АААААААТ и он определит скорость автоматически. |
| 8 | Data bits | Количество битов данных |
| None | Parity | Бит чётности |
| 1 | Stop bits | Количество стоповых битов |
| Off | Hardware flow control | Аппаратный контроль потока |
| Off | Software flow control | Программный контроль потока |



Отправка AT-команд для модема в терминале программы `minicom`

Для работы в интерактивном режиме рекомендуем использовать утилиту `minicom`:

1. Подключитесь к контроллеру по SSH.
2. Перезапустите модем командой:

```
wb-gsm restart_if_broken
```

3. Подключитесь к модему через `minicom`:

```
minicom -D /dev/ttyGSM -b 115200 -8 -a off
```

о параметрах командной строки читайте в статье о `minicom`.

4. Введите команду АААААААТ — с её помощью модем распознает скорость, с которой мы к нему обращаемся и ответит OK.

Модем готов к передаче AT-команд.

Чтобы закрыть `minicom`, нажмите на клавиатуре клавиши `Ctrl+A`, затем клавишу `X` и подтвердите выход клавишей `Enter`.

Работа с sms и ussd

Работать с sms и ussd проще всего при помощи программы `Gammu` (<http://wammu.eu/gammu/>) (это форк утилиты `gnokii`, которую перестали развивать).

Полную документацию смотрите на сайте проекта, ниже дана краткая инструкция.

Настройка

Все контроллеры WB6, начиная с 2018 года, поставляются с уже настроенной утилитой *gammu*. Если *gammu* не настроена, то можно выбрать один из 2-х способов настройки:

- Обновить пакет *wb-configs*. Для этого, нужно выполнить команды

```
apt update && apt install wb-configs
```

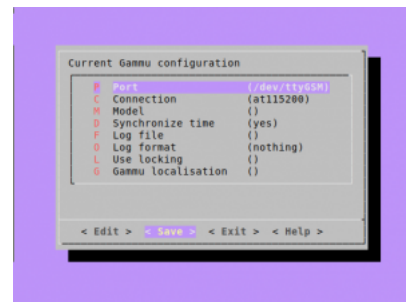
- Настроить *gammu* вручную:

1. Выполнить команду

```
gammu-config
```

2. В параметре Port укажите `/dev/ttyXXX` — файл модема, соответствующий вашей модели контроллера.

3. В параметре Connection укажите `at115200`



Настройка *gammu* вручную (*gammu-config*)

Примеры команд *gammu*

Перед использованием утилиты убедитесь, что соединение с интернетом по протоколу *ppp* завершено (см. раздел Интернет через *PPP*)

```
$ gammu networkinfo # посмотреть сеть и базовую станцию, к которой вы подключены
$ gammu geteasms # вывести все SMS
$ gammu getussd '#100#' # запросить баланс на МТС в транслите
$ gammu sendsms TEXT +79154816102 -unicode -text 'Привет' # отправить на номер сообщение с текстом
```

SMS и USSD на русском

SMS и USSD на русском в *gammu* пока работают не всегда хорошо, поэтому могут пригодиться команды для переключения языка USSD и перекодирования входящих и исходящих SMS в транслит:

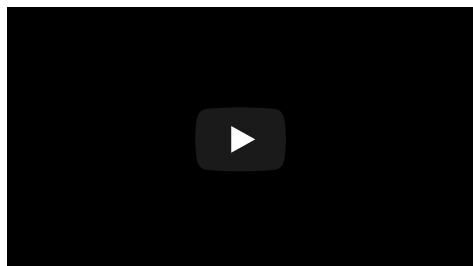
| Оператор | USSD | | SMS | |
|----------|-----------|-----------|------------|------------|
| | транслит | русский | транслит | русский |
| МТС | *100*6*2# | *100*6*1# | неизвестно | неизвестно |
| Мегафон | *105*0# | *105*9# | неизвестно | неизвестно |
| Билайн | *111*6*2# | *111*6*1# | неизвестно | неизвестно |
| Теле2 | *120# | *120*1# | неизвестно | неизвестно |

Для надежной отправки SMS на русском надо проверить локаль и установить `LC_ALL=ru_RU.utf8`

SMS-уведомления

Отправка *sms*-уведомлений об изменении состояния какого-либо устройства реализована в ПО *Wiren Board* с помощью сервиса уведомлений. Также можно отправлять SMS из движка правил *wb-rules*, вызывая соответствующую функцию. Подробнее в статье «Модуль уведомлений».

Интернет через PPP



Настройка интернета через *PPP* с помощью 2G-модема

Быстрый выход в интернет

Настройки быстрого подключения сбрасываются после перезагрузки контроллера. Если вам нужен постоянный доступ к интернету — настройте автоматический запуск подключения.

В стандартное ПО контроллера входят настройки подключения для операторов МТС, Мегафон и Билайн по протоколу *ppp*. Если вы пользуетесь одним из них, то для быстрого подключения к интернету нужно перезапустить модем и подключиться с использованием одной из настроек:

1. Перезапустите модем:

```
wb-gsm restart_if_broken
```

2. Установите соединение, например, для оператора МТС:

```
pon mts
```

mts можно заменить на megafon или beeline — зависит от вашего оператора связи.

3. Если соединение больше не нужно — вы можете его завершить командой:

```
poff mts
```

Если на контроллере установлен модуль 3G- или 4G-модем, то для увеличения пропускной способности соединения, демону pppd нужно указать другой порт. Для этого в файле /etc/ppp/peers/<ваш_провайдер_связи> замените устройство /dev/ttyGSM на (в зависимости от модема) /dev/ttyUSB0 или /dev/ttyACM0.

Например, изменим порт для провайдера МТС:

1. Откройте файл /etc/ppp/peers/mts

```
mcedit /etc/ppp/peers/mts
```

2. Закомментируйте старый порт и добавьте новый:

```
#/dev/ttyGSM  
/dev/ttyUSB0
```

3. Сохраните изменения и закройте файл.

Порт /dev/ttyUSB0 появляется автоматически после включения модема командой `wb-gsm on`.

Автоматический запуск подключения

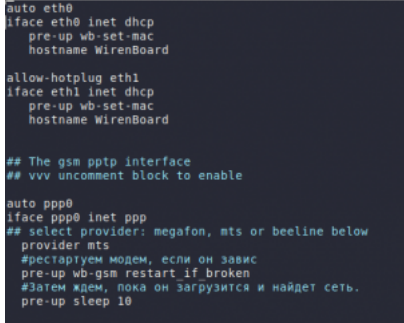
Чтобы подключение запускалось автоматически:

1. Откройте файл /etc/network/interfaces для редактирования:

```
mcedit /etc/network/interfaces
```

2. Раскомментируйте или отредактируйте следующие строки:

```
auto ppp0  
iface ppp0 inet ppp  
  provider mts # можно заменить mts на megafon или beeline  
  #перезапускаем модем, если он завис  
  pre-up wb-gsm restart_if_broken  
  #ждем, пока он загрузится и найдет сеть.  
  pre-up sleep 10
```



```
auto eth0  
iface eth0 inet dhcp  
  pre-up wb-set-mac  
  hostname WirenBoard  
  
allow-hotplug eth1  
iface eth1 inet dhcp  
  pre-up wb-set-mac  
  hostname WirenBoard  
  
## The gsm ppp interface  
## vvv uncomment block to enable  
  
auto ppp0  
iface ppp0 inet ppp  
  ## select provider: megafon, mts or beeline below  
  provider mts  
  #рестартуем модем, если он завис  
  pre-up wb-gsm restart_if_broken  
  #затем ждем, пока он загрузится и найдет сеть.  
  pre-up sleep 10
```

Файл /etc/network/interfaces, автоматически запускающий подключение к МТС

3. Сохраните изменения и закройте файл.

4. Теперь запустите интерфейс ppp0 командой:

```
ifup ppp0
```

через 10-15 секунд интерфейс ppp0 будет доступен.

5. Настройка завершена, теперь при перезагрузке контроллера подключение к интернету восстановится автоматически.

Параметры протокола пакетной передачи данных и номера для соединения для каждого провайдера хранятся в директории /etc/chatscripts. В большинстве случаев ничего менять в этих файлах не придется.

Для ppp-интерфейсов существуют директории, исполняемые файлы из которых также запускаются на разных фазах установления соединения. Но, если, например, для ethernet-интерфейсов эти скрипты должны находиться в директориях /etc/network/if-down.d, if-post-down.d, if-pre-up.d, if-up.d, то соответствующие директории для ppp-интерфейсов находятся в /etc/ppp/ip-down.d, ip-up.d и т.п. Подробнее об их назначении и функционировании можно узнать в документе PPP HOWTO (http://citforum.ru/operating_systems/linux/HOWTO/PPP-HOWTO.shtml).

Автоматическое восстановление подключения

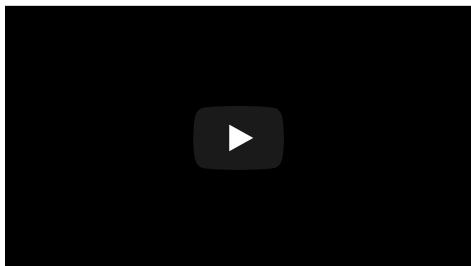
Скрипт позволяет восстановить интернет-соединение после сбоя. Пример автоматического запуска скрипта можете посмотреть в статье про WiFi

```
#!/bin/sh
echo
echo WAN CONTROLL RESTART
echo
PINGRESORCE1="ya.ru"
PINGRESORCE2="google.com"
if (! ping -q -c3 ${PINGRESORCE1} > /dev/null 2>&1)
then
if (! ping -q -c3 ${PINGRESORCE2} > /dev/null 2>&1)
then
wb-gsm restart_if_broken
else
echo 'internet ok'
fi
else
echo 'internet ok'
fi
```

Резервирование канала связи

Особенности резервирования выхода в интернет описаны в статье [Сетевые настройки контроллера](#).

Интернет с 4G-модемом (LTE)



Настройка модема WBC-4G на контроллере Wiren Board 6.7.2

Настройка модема как сетевой карты

В отличие от 2G- и 3G-модулей, WBC-4G поддерживает выход в интернет через виртуальную сетевую карту по протоколу RNDIS.

Настройка с помощью minicom:

1. Убедитесь, что модем правильно сконфигурирован. Подробнее смотрите на странице модуля WBC-4G.
2. Подключитесь к контроллеру по SSH.
3. Перезапустите модем командой:

```
wb-gsm restart_if_broken
```

4. Подключитесь к модему через minicom:

```
minicom -D /dev/ttyGSM -b 115200 -8 -a off
```

о параметрах командной строки читайте в статье о minicom.

5. Введите команду AAAAAAAAAAT — с её помощью модем распознает скорость, с которой мы к нему обращаемся и ответит ОК.
6. Отправьте из терминала minicom AT-команды для модема:

- Настроить автоматическое подключение: AT+DIALMODE=0.
- Установить APN: AT+CGDCONT=1,"IP","xxx", где xxx — точка подключения (APN). Имя точки подключения зависит от оператора, например, у МТС она выглядит так: internet.mts.ru.
- Проверить получение IP адреса: AT+CGCONTRDP.

7. Закройте minicom, для этого нажмите на клавиатуре клавиши **Ctrl**+**A**, затем клавишу **X** и подтвердите выход клавишей **Enter**.

После этого интернет будет доступен через интерфейс usb0, который можно настроить как обычную сетевую карту.

Настройка с помощью chat:

1. Убедитесь, что модем правильно сконфигурирован. Подробнее смотрите на странице модуля WBC-4G.
2. Подключитесь к контроллеру по SSH.
3. Перезапустите модем командой:

```
wb-gsm restart_if_broken
```

Отправка AT-команд для модема в терминале программы minicom

Выход из программы minicom

4. Замените в строке ниже APN_INTERNET на точку подключения вашего провайдера, вставьте изменённую строку консоль контроллера и нажмите на клавиатуре `Enter`:

```
PORT=/dev/ttyGSM; /usr/sbin/chat -s TIMEOUT 20 ABORT "ERROR" ECHO ON "" "AAAAAAAAAAAAAT" OK "AT+CMGF=1" OK "AT+DIALMODE=0" OK "AT+CGDCONT=1,\"IP\", \"APN_INTERNET\""" OK "AT+CGCONTRDP" "OK" > $PORT < $PORT
```

Этот способ можно использовать при написании скриптов.

Настройка виртуальной сетевой карты

Включение

После того как мы настроили модем, нужно настроить виртуальную сетевую карту:

1. Откройте файл `/etc/network/interfaces`:

```
nano /etc/network/interfaces
```

2. Добавьте в него строки:

```
auto usb0
allow-hotplug usb0
iface usb0 inet dhcp
pre-up wb-gsm restart_if_broken
pre-up sleep 10
```

автоматически запускать модем, интерфейс и получать IP-адрес.

3. Сохраните и закройте файл `interfaces`, для этого нажмите клавиши `Ctrl+O`, затем `Enter` и `Ctrl+X`.
4. Запустите интерфейс командой:

```
ifup usb0
```

Настройка завершена, теперь модем по DHCP назначит контроллеру IP-адрес в подсети `192.168.0.1`, а после перезагрузки контроллера соединение с интернетом восстановится автоматически.

Отключение

1. Остановите интерфейс командой:

```
ifdown usb0
```

2. Откройте файл `/etc/network/interfaces`:

```
nano /etc/network/interfaces
```

3. Закомментируйте или удалите строки:

```
#auto usb0
#allow-hotplug usb0
#iface usb0 inet dhcp
# pre-up wb-gsm restart_if_broken
# pre-up sleep 10
```

4. Сохраните и закройте файл `interfaces`, для этого нажмите клавиши `Ctrl+O`, затем `Enter` и `Ctrl+X`.

Мультиплексирование

Модем поддерживает режим мультиплексирования — создания виртуальных портов, через которые можно одновременно работать с модемом. Например, через один порт можно открыть сессию PPP для GPRS, а через другой — получать и отправлять SMS, проверять баланс и т.д. Подробнее смотрите CMUX. Этот режим не поддерживается для 2G-модемов.

Документация на модемы

| Модель | Режимы сети | краткое описание | hardware design | АТ-команды |
|----------|-------------|---|---|---|
| SIM800 | 2G | pdf (http://www.mt-system.ru/sites/default/files/documents/sim800_spec_20140423.pdf) | pdf (http://www.mt-system.ru/sites/default/files/documents/sim800_hardware_design_v1.10.pdf) | pdf (http://www.mt-system.ru/sites/default/files/documents/sim800_series_at_command_manual_v1.12.pdf) |
| SIM5300E | 2G/3G | pdf (http://www.mt-system.ru/sites/default/files/documents/sim5300e_spec_v1611_rus_0.pdf) | pdf (http://www.mt-system.ru/sites/default/files/documents/sim5300e_hardware_design_v1.09.pdf) | pdf (http://www.mt-system.ru/sites/default/files/documents/sim5300e_at_command_manual_v1.01.pdf) |
| SIM7000E | 2G/NB-IoT | pdf (http://www.mt-system.ru/sites/default/files/documents/sim7000e_spec_v1706_rus.pdf) | pdf (http://www.mt-system.ru/sites/default/files/documents/sim7000_hardware_design_v1.07.pdf) | pdf (http://www.mt-system.ru/sites/default/files/documents/sim7000_series_at_command_manual_v1.06.pdf) |
| 7600E | 2G/3G/4G | WBC-4G | | |

GPRS на модемах SIM7000E 2G/NB-IoT

Модем SIM7000E 2G/NB-IoT по умолчанию настроен на автоматический выбор GSM- и LTE-сетей. Однако, в сети или с SIM-картой без поддержки NB-IoT модем не регистрируется в сети GSM (GPRS). Для того, чтобы модем смог зарегистрироваться в сети GSM, необходимо принудительно перевести его в режим GSM only.

В терминальном режиме работы с модемом, например, в программе minicom (смотрите раздел Отправка АТ-команд), введите команду выбора режима:

```
AT+CNMP=13
```

Возможные варианты значений (команда AT+CNMP=?):

- 2 — Automatic,
- 13 — GSM Only,
- 38 — LTE Only,
- 51 — GSM And LTE Only.

Установить режим нужно один раз — он запоминается и активен даже после отключения питания.

Чтобы вернуться в режим IoT, выполните команду:

```
AT+CNMP=51
```

Wi-Fi

| Contents |
|---|
| Режимы работы |
| Первое подключение по Wi-Fi |
| Антенны |
| Подключение к точке доступа |
| Настройка Wi-Fi на контроллере Wiren Board |
| Настройка в режиме точки доступа |
| Установка пароля на подключение к точке доступа |
| Настройка в режиме точки доступа и клиента одновременно |
| Отключение режима точки доступа |
| Настройка в режиме клиента |
| Подключение к Wi-Fi точке доступа вручную |
| Универсальный файл настроек Wi-Fi |
| Автоматическое переключение при проблемах с соединением |

Режимы работы

Wi-Fi в Wiren Board можно настроить на работу в одном из двух или трёх режимов:

- Режим точки доступа (включён по умолчанию). Работает относительно медленно. Скорости вполне хватит для работы с веб-интерфейсом, но не стоит использовать как замену роутера.
- Режим клиента.
- Одновременная работа в режиме и точки доступа, и клиента.

В очень редких случаях возможна несовместимость адаптера Wi-Fi в Wiren Board с некоторыми другими устройствами Wi-Fi. Это общая проблема реализаций Wi-Fi на чипсетах разных производителей. Если вы столкнулись с необъяснимыми проблемами при работе, рекомендуем поменять настройки шифрования, ширины канала и т.п.

Первое подключение по Wi-Fi

Антенны

Прикрутите антенну к разъёму для антенны Wi-Fi.

Без антенны Wi-Fi в контроллерах Wiren Board работает на расстоянии не более одного метра. Чтобы получить стандартный для Wi-Fi радиус работы, нужно подключить к соответствующему разъёму контроллера антенну. Если контроллер находится в щитке (особенно в металлическом) или отдельной комнате, лучше использовать выносную антенну. Разъём для антенны — стандартный для Wi-Fi RP-SMA (https://en.wikipedia.org/wiki/SMA_connector#Reverse_polarity_SMA) ("гнездо", у GSM-антенн - наоборот).



Сравнение разъемов для антенн Wi-Fi (RP-SMA) и GSM (SMA)

Подключение к точке доступа

Контроллер создает Wi-Fi точку доступа и мы можем подключиться к ней:

- Откройте на ноутбуке или телефоне список WiFi точек доступа.
- Выберите из списка точку доступа с именем WirenBoard-XXXXXXX. Где XXXXXXX - серийный номер контроллера.

При подключении по Wi-Fi контроллер будет доступен по IP-адресу **192.168.42.1**.

По умолчанию, для подключения к контроллеру по Wi-Fi не требуется пароль, но вы можете это изменить.

Настройка Wi-Fi на контроллере Wiren Board

Настройка производится стандартным для Linux Debian способом - через файл `/etc/network/interfaces`. Краткие инструкции для типовых задач даны ниже, на сайте Linux Debian есть подробная документация (<https://wiki.debian.org/ru/NetworkConfiguration>).

Настройка в режиме точки доступа

Режим точки доступа включён по умолчанию. Работа в режиме точки доступа обеспечивается демоном **hostapd** (<https://wireless.wiki.kernel.org/en/users/documentation/hostapd>).

Сперва настраиваем демон hostapd:

1. в файле `/etc/default/hostapd` раскомментируйте строку (то есть удалите знак `#` в начале строки)

```
DAEMON_CONF="/etc/hostapd.conf"
```

2. отредактируйте файл `/etc/hostapd.conf`, чтобы он выглядел так:

```
interface=wlan0
#driver=nl80211 # оставьте эту строку закомментированной
ssid=WirenBoard # вместо WirenBoard можете подставить другое имя для создаваемой точки доступа
channel=1
```

Теперь нужно настроить сам интерфейс. Настройка делается в файле `/etc/network/interfaces`:

1. раскомментируйте и отредактируйте (или добавьте, если их не было) строки, относящиеся к настройке в режиме точки доступа:

```
iface wlan0 inet static
address 192.168.42.1 # здесь 192.168.42.1 - адрес, по которому в новой сети будет находиться Wiren Board; можете указать другой адрес
netmask 255.255.255.0
```

2. закомментируйте строки, относящиеся к работе в режиме клиента:

```
#auto wlan0
#iface wlan0 inet dhcp
# wpa-ssid {ssid}
# wpa-psk {password}
```

Выполните команду

```
/etc/init.d/hostapd restart
```

В итоге у нас получилась открытая точка доступа, для подключения к которой не требуется пароль.

Установка пароля на подключение к точке доступа

После установки пароля в файле `/etc/hostapd.conf`

подключитесь к контроллеру по SSN и откройте файл настроек `/etc/hostapd.conf`, для этого введите команду:

```
nano /etc/hostapd.conf
```

Добавьте в конец файла строки:

```
wpa=2
wpa_passphrase=your_password
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
rsn_pairwise=TKIP CCMP
```

Придумайте свой пароль и замените в файле `your_password` на него. Сохраните файл нажатием клавиш `Ctrl+O` и выйдете из редактора `Ctrl+X`.

После этого выполните команду:

```
systemctl restart hostapd
```

Контроллер применит новые настройки и связь с ним будет потеряна. Нужно будет заново подключиться к контроллеру по WiFi с указанным паролем. Если изменения настроек вы не можете подключить к контроллеру по WiFi — подключитесь к нему по Ethernet и проверьте настройки в файле `/etc/hostapd.conf`.

Настройка в режиме точки доступа и клиента одновременно

Режим одновременной работы модуля Wi-Fi и в режиме точки доступа, и в режиме клиента, называется *Concurrent Mode* или *STA+SoftAP*, и поддерживается не всеми Wi-Fi модулями. Он работает на всех версиях Wiren Board 6 и на некоторых ревизиях WB5. Проверено, что он работает из коробки на Wiren Board с чипом Realtek 8723BU и ядром Linux 4.1.15. Чтобы проверить, выполняются ли эти условия, выполните команды:

```
uname -a
lsmod | grep 8723bu
```

Если условия не выполнены, возможно, на вашем Wiren Board, всё равно, можно настроить *Concurrent Mode*. В качестве отправной точки используйте инструкцию (<http://randomstuffidosometimes.blogspot.ru/2016/03/rtl8192cu-and-rtl8188cus-in-station-and.html>).

Если условия выполнены:

1. Выполните команду

```
iwconfig
```

В её выводе должны быть показаны два интерфейса Wi-Fi: `wlan0` и `wlan1`.

2. Настройте по двум предыдущим инструкциям подключение в режиме клиента и подключение в режиме точки доступа, но используйте для них разные интерфейсы. Например, оставьте `wlan0` для точки доступа, а клиента сделайте на `wlan1`. Соответствующая часть файла `/etc/network/interfaces` должна выглядеть так:

```
# Wireless interfaces
auto wlan1
iface wlan1 inet dhcp
    wpa-ssid {ssid} # вместо {ssid} подставьте имя точки доступа
    wpa-psk {password} # вместо {password} подставьте пароль

auto wlan0
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

Отключение режима точки доступа

Если вы хотите перевести адаптер в режим клиента, подключиться к Wi-Fi точке доступа в ручном режиме или совсем отключить Wi-Fi на контроллере — отключите режим точки доступа:

1. Отключите автоматический запуск сервиса `hostapd`:

```
systemctl disable hostapd
```

2. Остановите демон `hostapd`

```
service hostapd stop
```

3. Теперь прокомментируйте настройки точки доступа и задайте настройки WiFi-клиента:

- откройте файл для редактирования

```
mcedit /etc/network/interfaces
```

- **закомментируйте строки, относящиеся к настройке в режиме точки доступа:**

```
#allow-hotplug wlan0
#iface wlan0 inet static
# address 192.168.42.1
# netmask 255.255.255.0
```

4. Сохраните и закройте файл настроек.
5. Запретите раздачу IP-адресов, для этого остановите DHCP-сервер:

```
systemctl disable dnsmasq
service dnsmasq stop
```

Режим точки доступа отключен, чтобы его включить, выполните инструкции из раздела Настройка в режиме точки доступа.

Настройка в режиме клиента

После настройки точки доступа в режиме клиента, контроллер будет подключаться к точке доступа автоматически при каждой загрузке операционной системы.

Вы можете настроить автоматическое подключение контроллера к Wi-Fi точке доступа:

1. Отключите точку доступа по инструкции в разделе Отключение режима точки доступа
2. Откройте файл настроек:

```
mcedit /etc/network/interfaces
```

3. Раскомментируйте и отредактируйте строки (или добавьте, если их не было):

```
auto wlan0
iface wlan0 inet dhcp
wpa-ssid ssid # вместо ssid подставьте имя точки доступа
wpa-psk password # вместо password подставьте пароль
```

4. Если точка доступа скрыта, то добавьте параметр:

```
wpa-scan-ssid 1
```

5. Сохраните и закройте файл настроек.
6. Завершите настройку, для этого перезапустите беспроводной интерфейс командами:

```
ifdown wlan0 && ifup wlan0
```

Подключение к Wi-Fi точке доступа вручную

Подключение в ручном режиме будет разорвано после перезагрузки контроллера.

Если у вас возникла проблема с настройкой автоматического подключения, то вы можете попробовать подключиться к Wi-Fi точке доступа вручную:

1. Отключите точку доступа по инструкции в разделе Отключение режима точки доступа
2. Запустите поиск доступных точек доступа с помощью команды `iwlist wlan0 scanning`:

```
~# iwlist wlan0 scanning | grep -i essid
ESSID:"DIR-615"
ESSID:"MT5Router_2.4GHz_072433"

ESSID:"Smart_box-40B598"
ESSID:"TP-Link_0E5AW"
ESSID:"TP-LINK_78DC"
```

в примере контроллер «видит» пять точек доступа.

3. Этот шаг зависит от типа сетевой аутентификации, выбранной в настройках точки доступа, к которой вы хотите подключиться:

- **WPA-PSK:**

1. Задайте параметры подключения:

```
iwconfig wlan0 essid ИмяТочкиДоступа key Пароль0тТочкиДоступа
```

2. Запустите сетевой интерфейс:

```
ifconfig wlan0 up
```

■ WPA2-PSK:

1. Сгенерируйте файл с учётной записью для подключения к точке доступа:

```
wpa_passphrase ИмяТочкиДоступа Пароль0ТочкиДоступа > /root/wpa.conf
```

2. Установите подключение с использованием сгенерированного файла:

```
wpa_supplicant -Dwext -iwlan0 -c/root/wpa.conf &
```

4. Подождите 15 секунд и проверьте подключение командой iwconfig wlan0:

```
~# iwconfig wlan0 | grep -i essid  
wlan0 IEEE 802.11bgn ESSID:"DIR-615" Nickname:"<WIFI@REALTEK>"
```

в примере контроллер подключён к точке доступа с именем DIR-615. Если в строке будет unassociated, то контроллер не смог подключиться.

5. Если контроллер успешно подключился к точке доступа и на ней запущен DHCP-сервер, то запустите dhclient:

```
dhclient wlan0
```

6. Проверьте, получил ли контроллер IP адрес, для этого используйте команду ip a:

```
~# ip a | grep wlan0  
5: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
inet 192.168.2.83/24 brd 192.168.2.255 scope global wlan0
```

в примере контроллер получил ip-адрес 192.168.2.83.

Настройка подключения контроллера к точке доступа завершена.

Универсальный файл настроек Wi-Fi

Ниже приведен текст файла с настройками для подключения к сетям с разными параметрами шифрования. Оригинал файла можно посмотреть на сайте www.raspberrypi.org (<https://www.raspberrypi.org/forums/viewtopic.php?t=7592>).

```
#####  
#; start of wireless bits  
#; this command stays for all configs  
auto wlan0  
#####  
#; comments indicated by #;  
#; commands indicated by #  
#; remove the # to enable the command  
#####  
#; if using static IP then...#  
#iface wlan0 inet static  
# address UR_IP  
#gateway UR_ROUTER_IP  
#netmask 255.255.255.0  
#####  
#; otherwise use dhcp #  
#iface wlan0 inet dhcp  
#####  
#; OPEN wireless config #  
#wireless-essid UR_ESSID  
#wireless-mode managed  
#####  
#; WEP wireless config #  
#wireless-essid UR_ESSID  
#wireless-key UR_KEY  
#; end of WEP config  
#####  
#; WPA and WPA2 wireless config #  
#; all command config lines above HERE to be #'ed except the entry auto wlan0  
#####  
wpa-driver wext  
wpa-ssid UR_ESSID  
#; wpa-ap-scan is 1 for visible and 2 for hidden hubs  
wpa-ap-scan 1  
#; wpa-proto is WPA for WPA1 (aka WPA) or RSN for WPA2  
wpa-proto WPA  
#; wpa-pairwise and wpa-group is TKIP for WPA1 or CCMP for WPA2  
wpa-pairwise TKIP  
wpa-group TKIP  
wpa-key-mgmt WPA-PSK  
#; use "wpa_passphrase UR_ESSID UR_KEY" to generate UR_HEX_KEY  
#; enter the result below  
wpa-psk UR_HEX_KEY
```



```
#####  
# end of wireless bits
```

Автоматическое переподключение при проблемах с соединением

Способ заимствован на сайте alexba.in (<http://alexba.in/blog/2015/01/14/automatically-reconnecting-wifi-on-a-raspberrypi/>).

Допустим, контроллер подключён к роутеру с адресом 192.168.0.1 через интерфейс wlan1:

1. Создайте в папке /root скрипт wifi_autoconnect.sh:

```
mcedit /root/wifi_autoconnect.sh
```

с содержанием:

```
#!/bin/bash  
  
# Подставьте имя интерфейса  
WLANINTERFACE=wlan1  
# Подставьте адрес роутера или сервера в интернете, доступ к которому будет проверяться  
SERVER=192.168.0.1  
  
PATH="/bin:/sbin:/usr/local/sbin:/usr/sbin:$PATH"  
# Only send two pings, sending output to /dev/null  
ping -I ${WLANINTERFACE} -c2 ${SERVER} > /dev/null  
  
# If the return code from ping ($) is not 0 (meaning there was an error)  
if [ $? != 0 ]  
then  
# Restart the wireless interface  
ifdown --force ${WLANINTERFACE}  
ifup ${WLANINTERFACE}  
fi
```

2. Сделайте файл исполняемым, выполнив команду

```
chmod +x /root/wifi_autoconnect.sh
```

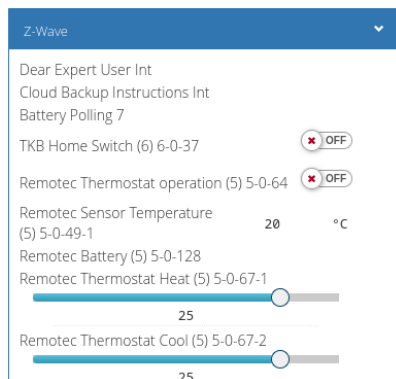
3. Запланируйте выполнение скрипта каждую минуту:

Добавьте в конец файла /etc/crontab строку

```
* * * * * root /root/wifi_autoconnect.sh  
# Обязательно добавьте пустую строку в конец файла
```

Подключение Z-Wave устройств к контроллеру Wiren Board

- English
- русский



Представление Z-Wave устройств в веб-интерфейсе контроллера Wiren Board

Contents

Описание

Установка и настройка z-way-server

- Описание
- Установка
- Настройка

Установка Wiren Board MQTT Integration

Описание
Принцип работы
Установка интеграции

Работа с Z-Wave устройствами

Добавление
Для модуля интеграции версии 1.4.2
Для модуля интеграции версии 1.5.0
Управление
Удаление
Для модуля интеграции версии 1.4.2
Для модуля интеграции версии 1.5.0

Полезные ссылки

Описание



Логотип Z-Wave

Z-Wave — это беспроводной протокол для домашней и коммерческой автоматизации.

Для использования Z-Wave устройств с контроллером Wiren Board понадобится:

1. Модуль расширения WBE2R-R-ZWAVE-ZWAY или USB-стик Z-Wave.Me UZB1
2. Дополнительное ПО:
 - z-way-server — Z-Way сервер, который работает с устройствами.
 - Wiren Board MQTT Integration — интеграция, которая позволяет управлять Z-Way устройствами из основного веб-интерфейса контроллера Wiren Board.

Инструкции по установке смотрите ниже на этой странице.

Установка и настройка z-way-server

Описание

z-way-server — это комплексное программное обеспечение Z-Way, поддерживающее радиопротocolы Z-Wave и EnOcean, а также устройства на основе Wi-Fi, MQTT и HTTP. Пакет есть в репозитории контроллера Wiren Board.

Установка

1. Обновите пакеты на контроллере:

```
apt update && apt upgrade
```

2. Установите ПО для работы с Z-Wave:

```
apt install z-way-server
```

Веб-интерфейс сервера Z-Way будет доступен по адресу <http://192.168.42.1:8083>, где 192.168.42.1 — IP-адрес контроллера, укажите свой. При первом подключении сервер предложит задать пароль администратора — запомните его.

Настройка

Теперь нужно вставить в контроллер USB-стик или модуль расширения и указать порт в программном обеспечении:

1. Вставьте в контроллер USB-стик или модуль расширения. Модуль расширения перед использованием нужно конфигурировать.

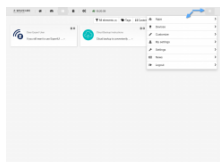
2. Зайдите в веб-интерфейс Z-Way.

3. Перейдите в **Меню справа вверху** → **Apps** → **Z-Wave Network Access** и в поле **Serial port to Z-Wave dongle** укажите одно из значений:

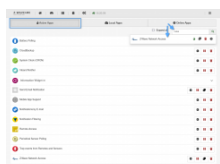
- USB-стик — /dev/ttyUZH;
- модуль расширения — /dev/ttyMODx, где x — номер разъёма, куда вставлен модуль.

Теперь вы можете работать с устройствами Z-Wave из веб-интерфейса ПО Z-Way. Если вы хотите видеть Z-Wave устройства в стандартном веб-интерфейсе контроллера и использовать их в автоматизациях, установите интеграцию *Wiren Board MQTT Integration*.

- Настройка порта



Вызов меню и переход в раздел Apps



Поиск Z-Wave Network Access

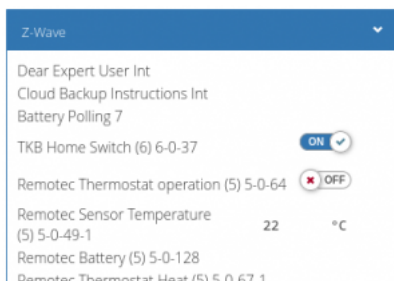


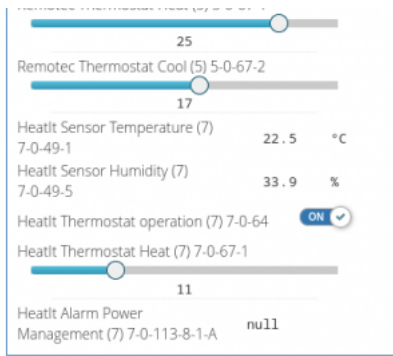
Настройки Z-Wave Network Access, порт



Настройки Z-Wave Network Access

Установка Wiren Board MQTT Integration





Представление устройств Z-Wave в веб-интерфейсе контроллера Wiren Board

Описание

Чтобы упростить обработку данных с Z-Wave устройств и управление ими, мы написали модуль интеграции *Wiren Board MQTT Integration*, который преобразовывает данные Z-Wave сервера в топик виртуального устройства, которое будет доступно в веб-интерфейсе контроллера на вкладке *Devices*.

После установки интеграции вы сможете управлять Z-Wave устройствами через стандартный веб-интерфейс контроллера, или настраивать автоматизации с помощью *wb-rules*.

Принцип работы

Преобразовывает описание настроенных в ПО Z-Wave устройств в каналы устройства Z-Wave, которое находится в веб-интерфейсе контроллера на вкладке **Devices**.

Можно получать с устройств данные и управлять ими.

| Поддерживаемые типы устройств | | |
|-------------------------------|-----------------|---|
| Тип Z-Way | Тип Wiren Board | Описание |
| sensorBinary | switch | Значения с бинарных датчиков |
| sensorMultilevel | value | Значения с датчиков температуры, влажности, освещенности и т.п. |
| battery | value | Значение с датчиков уровня батареи |
| doorlock | switch | Переключатель открытия/закрытия двери |
| switchBinary | switch | Переключатель ON/OFF |
| switchMultilevel | range | Переключатель с несколькими позициями |
| thermostat | range | Установка значения температуры в термостате |
| toggleButton | pushbutton | Нажатие на кнопку |

Установка интеграции

Установите модуль интеграции:

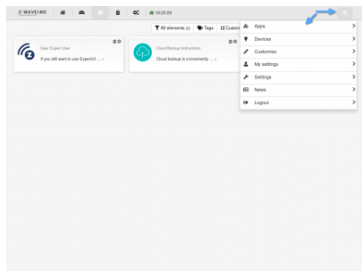
1. Зайдите в интерфейс сервера Z-Way.
2. Перейдите в **Меню справа вверху** → **Apps** → **Online Apps** .
3. Введите в поиске **Wiren Board MQTT Integration** и установите найденное приложение.
4. На этом этапе может потребоваться установить **BaseModule**, если это так, то в настройках модуля интеграции будет сообщение **please install** и кнопка **Base Module** — установите его нажатием на кнопку. Если сообщения нет — просто продолжайте настройку.
5. Если вы настраиваете интеграцию впервые и не знаете назначения настроек, то оставьте значения по умолчанию и нажмите внизу кнопку **Save**.

Если в процессе сохранения настроек модуля интеграции вы получите ошибку *Something went wrong* — перезапустите контроллер, найдите *Wiren Board MQTT Integration* теперь уже на вкладке **Local Apps**, нажмите на плюсик — откроется карточка с настройками интеграции, нажмите кнопку **Save**.

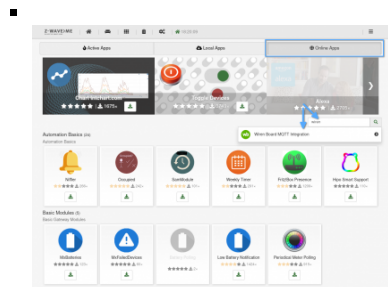
После этого во вкладке *Devices* веб-интерфейса контроллера Wiren Board будут появляться Z-Wave устройства, добавленные через интерфейс Z-Way.

— Установка интеграции для работы с Wiren Board

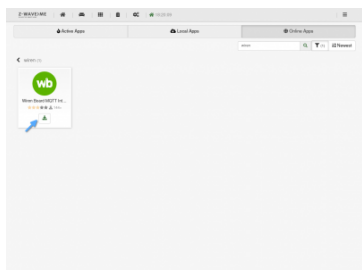
- установка интеграции для работы с wigen board



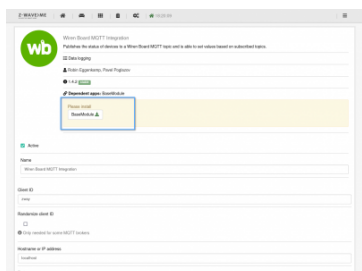
Вызов меню и переход в раздел *Apps*



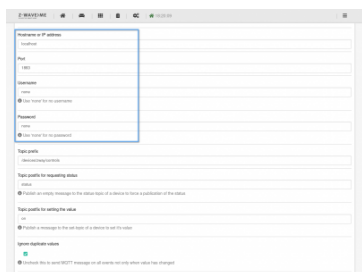
Поиск и выбор *Wired Board MQTT Integration*



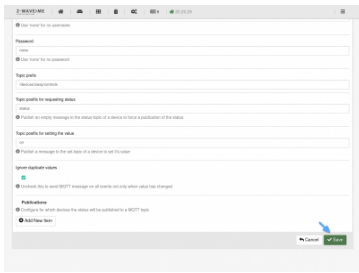
Установка *Wired Board MQTT Integration*



Запрос на установку *BaseModule*, будет только при первой установке интеграции



Настройка доступа к MQTT-брокеру



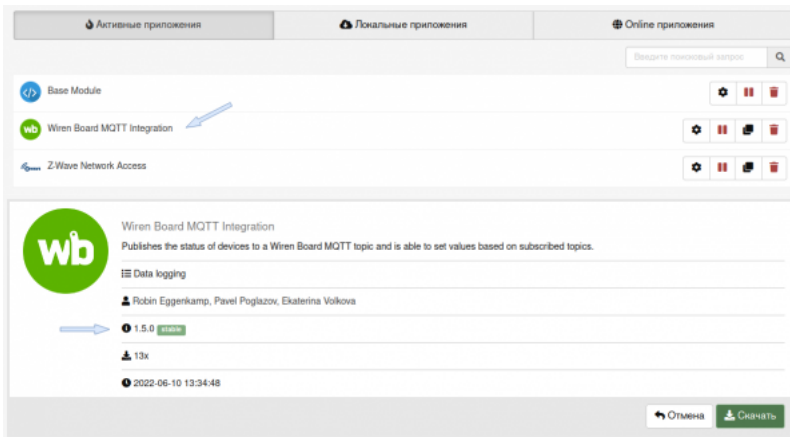
Сохранение настроек

Работа с Z-Wave устройствами

Добавление

Новое устройство добавляется в веб-интерфейсе Z-Way:

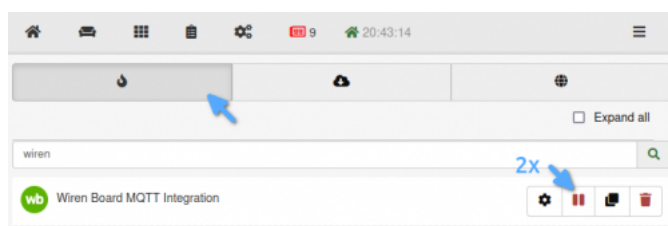
1. Зайдите в интерфейс сервера Z-Way.
2. Перейдите в **Меню справа вверху** → **Devices**
3. Найдите в списке **Z-Wave** и нажмите кнопку **Add new**.
4. Далее нажмите на кнопку **Add new Z-Wave Device and identify it automatically**
5. Если вы добавляете новое устройство, то нажмите зелёную кнопку **Start** и активируйте режим сопряжения на устройстве, обычно, для этого нужно нажать три раза на кнопку сопряжения. Если устройство уже было сопряжено с другим сервером Z-Way, то перед выполнением сопряжения его нужно сбросить, для этого нажмите на кнопку **Reset Device** и включите на устройстве режим сопряжения. После сообщения об успешном сбросе, выполните инструкцию в начале этого пункта.
6. После того, как устройство будет сопряжено, откроется страница, где можно задать устройству имя и настроить его. После этого нажмите кнопку **Save**.
7. Проверьте версию вашего модуля интеграции WirenBoard. Для этого откройте установленное приложение **Wiren Board MQTT Integration** на вкладке **Активные приложения**



Для модуля интеграции версии 1.4.2

После того, как вы добавите все устройства, перезапустите интеграцию Wiren Board MQTT Integration:

1. Перейдите в **Меню справа вверху** → **Apps** → **Active Apps**
2. Найдите в поиске **Wiren Board MQTT Integration**
3. Нажмите на кнопку **Activate/Deactivate** — это выключит интеграцию.
4. Снова нажмите на кнопку **Activate/Deactivate** — это включит интеграцию.

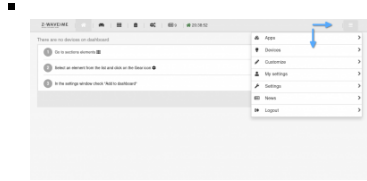


5. Добавление устройства завершено.

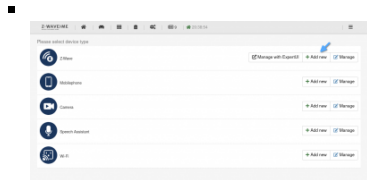
Для модуля интеграции версии 1.5.0

Дополнительных действий не требуется. Все вновь добавленные устройства автоматически добавляются в систему WirenBoard.

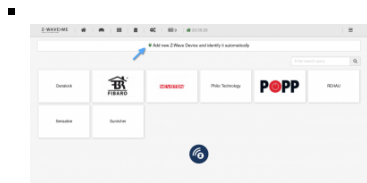
- Добавление нового Z-Wave устройства



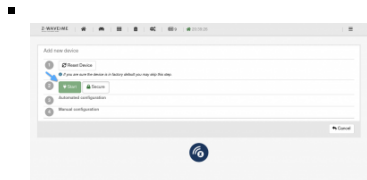
Вызов меню и переход в раздел *Devices*



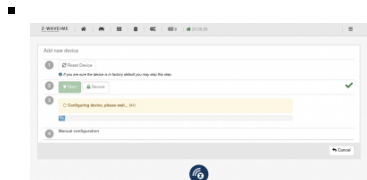
Добавление устройства типа Z-Wave



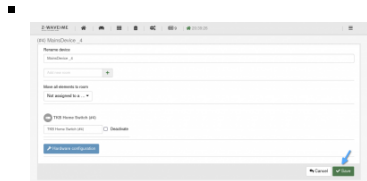
Переход с процедуре сопряжения устройств



Запуск сопряжения



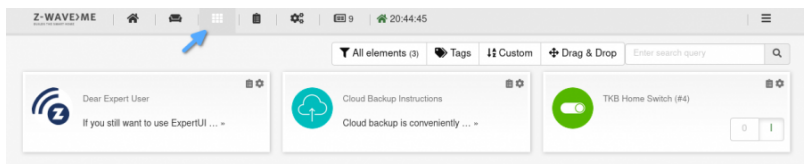
Сопряжение с устройством



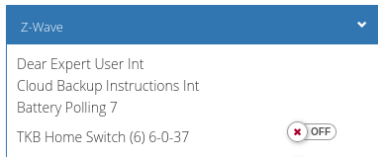
Настройка сопряжённого устройства

Список добавленных устройств можно найти в разделе *Elements*, кнопка перехода к которому находится вверху окна. Так же устройства будут доступны в веб-интерфейсе контроллера Wiren Board на вкладке *Devices* в карточке Z-Way.

- Представление Z-Wave устройства



Устройство в веб-интерфейсе Z-Way



Устройство в карточке Z-Wave в веб-интерфейсе контроллера Wiren Board

Управление

Устройствами можно управлять их веб-интерфейсов ПО Z-Wave и стандартного веб-интерфейса контроллера Wiren Board.

Кроме этого вы можете настраивать автоматизации с помощью правил `wb-rules` и стороннего ПО, которое работает с MQTT, например, Node-RED.

Удаление

Чтобы удалить Z-Wave устройство:

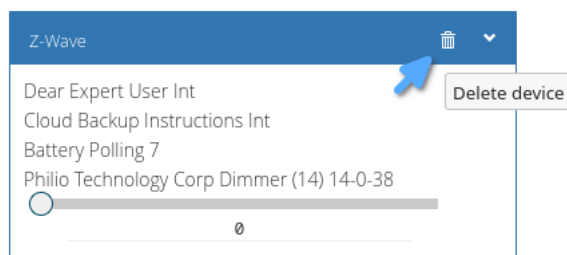
1. Зайдите в интерфейс сервера Z-Way.
2. Перейдите в **Меню справа вверху** → **Devices**
3. Найдите в списке **Z-Wave** и нажмите кнопку **Manage**.
4. Нажмите **кнопку с красной корзиной** рядом с названием устройства, которое хотите удалить.
5. Нажмите кнопку **Reset and Remove**, откроется всплывающее окно с инструкцией: вам нужно сделать на устройстве те же действия, которые включают режим сопряжения, например, три раза нажать на кнопку сопряжения.
6. Устройство сброшено и удалено.

Проверьте установленную версию модуля WirenBoard MQTT Intergation (инструкция о том, как это сделать, есть в разделе добавления Z-Wave устройств).

Для модуля интеграции версии 1.4.2

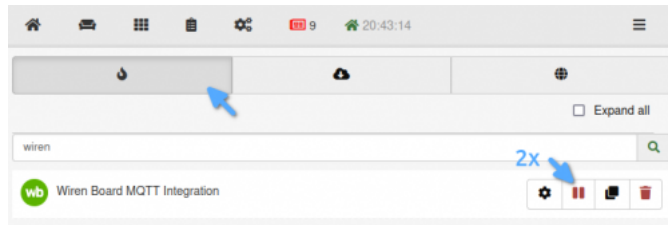
Теперь нужно удалить устройство в стандартном веб-интерфейсе Wiren Board:

1. Зайдите в стандартный веб-интерфейс контроллера
2. Перейдите в раздел **Devices** и найдите там устройство **Z-Wave**, наведите мышку на заголовок — появится корзина — нажмите на неё.



3. Теперь перезапустите интеграцию Wiren Board MQTT Integration:

1. Перейдите в **Меню справа вверху** → **Apps** → **Active Apps**
2. Найдите в поиске **Wiren Board MQTT Integration**
3. Нажмите на кнопку **Activate/Deactivate** — это выключит интеграцию.
4. Снова нажмите на кнопку **Activate/Deactivate** — это включит интеграцию, которая пересоздаст устройство Z-Wave в стандартном веб-интерфейсе контроллера.



4. Удаление устройства завершено.

Для модуля интеграции версии 1.5.0

Дополнительных действий не требуется. Устройство, которое вы удалили при помощи Z-Way сервера, также автоматически удаляется в системе WirenBoard.

- Удаление Z-Wave устройства



Вызов меню и переход в раздел *Devices*



Переход к управлению Z-Wave устройствами, кнопка *Manage*



Список Z-Wave устройств и кнопка удаления одного из них



Кнопка *Reset and Remove*, которая сбрасывает устройство и удаляет его из ПО

Полезные ссылки

- Официальный сайт программного обеспечения Z-Way
- Подключение реле Rubetek TZ78
- Подключение диммера Philio PAD07-RU
- Подключение термостата Heatit Z TEMP2

- подключение термостата peace z-temgz
- Подключение устройства управления кондиционером Philio PAR01-RU

| Поддерживаемые контроллером Wiren Board протоколы, устройства и системы верхнего уровня | |
|---|--|
| Поддерживаемые протоколы | |
| Опрос датчиков и работа с устройствами (в базовой комплектации) | 1-Wire • DLMS/COSEM • Modbus RTU/TCP Master • ГОСТ МЭК 61107 • СПОДЭС (ГОСТ Р 58940-2020) |
| Опрос датчиков и работа с устройствами (с помощью модулей расширения) | KNX • eBUS • OpenTherm • Z-Wave • Zigbee |
| Системы верхнего уровня | KNX • Modbus RTU/TCP Slave • MQTT • OPC UA • SNMP • Zabbix • МЭК 104 • SmartWeb |
| ПО верхнего уровня | Grafana • Home Assistant • IntraHouse • IntraSCADA • IRidium Server • MasterSCADA • Nagios • SimpLight SCADA • Rapid SCADA • SAYMON • Zabbix |
| Голосовые ассистенты и облачные решения | Сири и Apple HomeKit • Салют от SberDevices • Яндекс Алиса |
| Протестированные устройства сторонних производителей | |
| Адаптеры протоколов | Шлюз ECODim DALI GW2 |
| Датчики климата | DS18B20 и клоны • Kvaдро 1WIRE-RS485 • RLDA NL-3DPAS-M • RLDA NL-1S111 • Wellpro WP3066ADAM • Даджет MT8057/MT8057S • РД MSU21 • РД MSU24 • РД MSU34+TLP • РД MSU34+THLP • Эксис ИВТМ-7 М 3 |
| Датчики уровня | ЭККОРТ ДБ-2 |
| Диммеры | DALI • Philio PAD07-RU • Uniel UCH-M131RC/0808 • Uniel UCH-M141RC/0808 • РД DDL04R • РД DDL24 • РД DDL84R-V • РД DDM845R |
| Конвекторы | Varmann QTherm |
| Кондиционеры | Haier YCJ-A002 • Z-Wave ИК-передатчик PAR01-RU |
| Контроллеры вентиляции и климата | Mautomatics JL204C5 (Breezart 550 Lux) • GTC (General Thermo Controllers) Syberia 5.0 • SystemAir VR 300 |
| Контроллеры холодильного оборудования | Carel BASIC(PYEZ)/EASY(PJEZ) • Danfoss EKC 204A1 / AK-CC 210 • Danfoss EKC 202B • Danfoss EKC 202D • Danfoss ERC 211/ERC 213/ERC 214 • Eliwell IDPlus 974 |
| Метеостанции | Netatmo Urban Weather Station |
| Модули ввода-вывода | Wellpro WP8026ADAM • Wellpro WP8027ADAM • Wellpro WP8028ADAM • Wellpro WP9038ADAM |
| Модули реле | РД DRB88 • Rubetek TZ78 • ICP DAS tM-P3R3 • ICP DAS LC-103 • Uniel UCH-M111RX/0808 • Uniel UCH-M121RX/0808 |
| Моторы для штор/Электрокарнизы | Akko AM82 • Dooya DM35EQ • Dooya DT82 • WinDeco • Somfy SDN • SunFlower KT82TV • Somfy RS485 RTS transmitter |
| Управление двигателями (преобразователи частоты) | Vacon/Danfoss 10 • Danfoss VLT Microdrive FC51 • T13-400W-12-H |
| Счётчики воды | Пульсар • Пульсар-М • Элехант СВД-15 • Элехант СВД-20 • Счётчики с импульсным выходом |
| Счётчики тепла | Пульсар |
| Счётчики электроэнергии | CSQ PD561Z-9SY • Peacefair PZEM-016 • Eastron SDM120M • Eastron SDM220M • Меркурий 200 • Меркурий 201 • Меркурий 203.2Т • Меркурий 204 • Меркурий 206 • Меркурий 208 • Меркурий 230 • Меркурий 231 • Меркурий 234 • Меркурий 236 • Меркурий 238 • Милур 104 • Милур 105 • Милур 107 • Милур 305 • Милур 307 • Нева МТ 113 • Нева МТ 123 • Нева МТ 124 • Нева МТ 323 • Нева МТ 324 • Энергомера CE301 • Энергомера CE102М • Энергомера CE303 • Энергомера CE308 |
| Термостаты | BAC-6000 Series • BHT-6000 Series • Cityron ПУ-3 (Modbus) • Heatit Z-TEMP2 • Hessway • Siemens RDF302 |
| Увлажнители | CAREL Humisonic |
| Прочее | DIY • Shelly UNI • Tasmota • ESPHome |
| Устройства с аналоговым или цифровым выходом | |
| Низковольтная нагрузка | Реле с управляющим напряжением 12-24 В • Светодиоды • Низковольтные вентиляторы • Низковольтные сигнальные лампы |
| Датчики с аналоговым выходом | Датчики температуры, давления и другие, имеющие на выходе ток или напряжение |
| Счётчики с импульсным выходом | Счётчики электроэнергии, воды, тепла и другие с импульсным выходом |
| Устройства с выходом «открытый коллектор» | Устройства с выходом «открытый коллектор» |
| Устройства с питанием 220 В | Лампы • Контактторы • Другое оборудование, питающееся от 220 В |

wirenboard

Wiren Board 5

wirenboard

https://wirenboard.com/wiki/Wiren_Board_5
21-06-2022 10:10

Wiren Board 5

Руководство по эксплуатации

Самая актуальная документация всегда доступна на нашем сайте по ссылке: https://wirenboard.com/wiki/Wiren_Board_5

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

[Wiren Board 5](#)

[MQTT](#)

[SVG-панели \(Dashboards\)](#)

[Веб-интерфейс Wiren Board](#)

[Модуль уведомлений](#)

[Движок правил wb-rules](#)

[Zabbix](#)

[Шлюз Modbus RTU/TCP](#)

[Шлюз OPC UA](#)

[МЭК 104](#)

[Агент SNMP](#)

[Установка Node-RED на контроллер Wiren Board](#)

[Установка Home Assistant на контроллер Wiren Board](#)

[Установка Docker на контроллер Wiren Board](#)

[Обновление прошивки контроллера Wiren Board](#)

[Как разрабатывать ПО для Wiren Board](#)

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

[CryptodevATECCx08 Auth](#)

[Модули расширения](#)

[Модули ввода-вывода](#)

[WBMZ-BATTERY - модуль резервного питания](#)

[GSM/GPRS](#)

[Wi-Fi](#)

[Подключение Z-Wave устройств к контроллеру Wiren Board](#)

[Радиоинтерфейс 433 МГц](#)

[Питание USB-портов](#)

[RS-485](#)

[Драйвер wb-mqtt-serial](#)

[CAN](#)

[1-Wire в контроллерах Wiren Board](#)

[Подключение периферийных устройств к контроллеру Wiren Board](#)

[ADC](#)

[Дискретные входы](#)

[Отладочный порт](#)

[Зуммер \(звуковой излучатель\)](#)

[Consumer IR](#)

[Watchdog](#)

[Power over Ethernet](#)

[Центр документации](#)

[Устройства, протоколы и программы, с которыми может работать контроллер Wiren Board](#)

[Работа с GPIO](#)

[Wiren Board 5: Восстановление пароля пользователя root](#)

Wiren Board 5

Contents

[Общие характеристики](#)

[Программное обеспечение](#)

[Внутренние и внешние модули](#)

[Беспроводные интерфейсы](#)

[Проводные интерфейсы](#)



Универсальные входы/выходы A1-A4

Резистивные входы R1 и R2

Клеммники

Другие интерфейсы

Сторожевой таймер

Питание

Поддерживаемые устройства

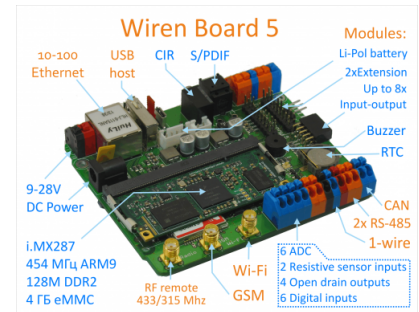
Подробное тех.описание платы контроллера

Изображения и чертежи устройства



Wiren Board 5

Общие характеристики



Wiren Board ревизии 5.3

| | |
|--------------------------|--|
| Процессор | Freescallе i.MX28 (http://www.freescale.com/products/arm-processors/i.mx-applications-processors-based-on-arm-cores/i.mx28-processors/multimedia-applications-processors-dual-ethernet-dual-can-lcd-touch-screen-arm9-core:i.MX287) 454 MHz ARM9 |
| Память оперативная | DDR2 SDRAM 128 MB |
| Память энергонезависимая | 4 GB eMMC |
| Операционная система | Debian Linux 7.0. Mainline kernel 4.1 |
| Габаритные размеры | Корпус на DIN-рейку шириной 6М. 106.25x90.2x57.5 мм. Размер без корпуса: 103x87x20 мм |
| Эксплуатация | Рабочая температура - -25...+70C (-40..+70C по запросу) |
| Питание | 7-28VDC |

Страница продукта и магазин: [Wiren Board 5 \(https://wirenboard.com/ru/product/wiren-board-5/\)](https://wirenboard.com/ru/product/wiren-board-5/)

Для начала работы с контроллером рекомендуется прочитать статью [Первое включение](#).

Программное обеспечение

Wiren Board работает под управлением стандартной сборки Debian Linux 9 Stretch. Для архитектуры используемого процессора есть официальный порт (<https://www.debian.org/ports/arm/>). Поэтому почти любой пакет найдётся в стандартном репозитории, и его можно установить одной командой `apt-get install имя_пакета`.

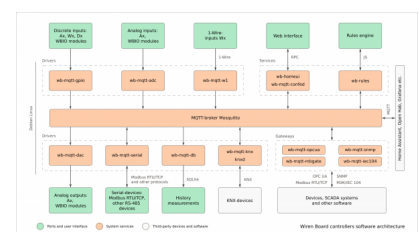
Есть две ветки ПО Wiren Board: **stable** и **testing**.

Исходный код программного обеспечения доступен на [GitHub \(https://github.com/contactless/\)](https://github.com/contactless/). Там можно почерпнуть примеры для разработки собственного ПО.

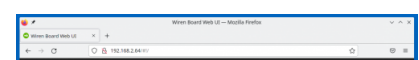
Очередь сообщений MQTT — «скелет» программной архитектуры Wiren Board.

Веб-интерфейс Wiren Board работает непосредственно на контроллере. В нём можно:

- следить за состоянием контроллера и подключённых устройств и управлять ими,
- подключать устройства к контроллеру,
- настраивать контроллер и обновлять его ПО,
- писать правила на встроенном движке,
- настраивать SMS- и email-уведомления,



Структура ПО контроллера. В центре очередь сообщений MQTT, которая используется для обмена информацией между разными частями ПО



- смотреть графики истории значений параметров: температуры, напряжения и т.п.

Движок правил wb-rules позволяет создавать собственные правила для контроллера, например: «Если температура датчика меньше 18°C, включи нагреватель». Правила создаются через веб-интерфейс и пишутся на простом Javascript-подобном языке.

Для работы с SCADA-системами есть:

- Агент Zabbix
- Шлюз Modbus TCP/RTU
- Шлюз OPC UA
- Шлюз МЭК 104
- Агент SNMP

Дополнительно:

- Node-RED — инструмент визуального программирования.
- Home Assistant — open-source платформа для автоматизации.
- Docker — программное обеспечение для запуска приложений в изолированной среде.

Полезные ссылки

- Обновление прошивки контроллера
- Как разрабатывать ПО для Wiren Board — статья для программистов.
- Обновление прошивок в Modbus-устройствах Wiren Board
- Использование встроенного чипа ATECCx08 и статья по теме на Хабре (<https://habr.com/ru/company/oleg-bunin/blog/476304/>)
- Сборка образа для загрузки в режиме USB Mass-Storage

Внутренние и внешние модули

Внутренние модули расширения - это небольшие платы, устанавливаемые внутрь корпуса Wiren Board 5 и расширяющие его функциональность - дополнительные порты RS-485, RS-232, релейные выходы и т. д.

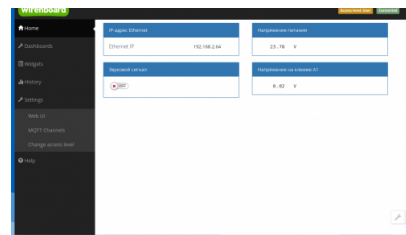
В контроллере есть два слота для подключения модулей расширения. На каждый модуль отводится 3 внешних клеммника.

Модули ввода-вывода стыкуются с боковым разъемом на WB5 с правой стороны, добавляя от 8 до 16 цифровых или аналоговых портов.

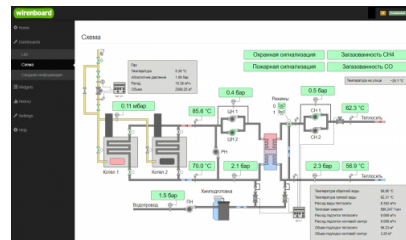
Последовательно можно подключать до 8 модулей: до 4 модулей ввода (типа I) и до 4-х модуля вывода (типа O и IO).

Модуль резервного питания - дополнительная мезонинная плата, устанавливается вторым этажом в корпус на DIN-рейку.

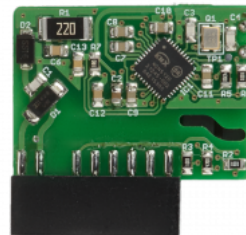
Содержит Li-Ion (Li-Pol) аккумулятором емкостью 1800 mAh, обеспечивает работу контроллера до 3 часов.



Главная страница веб-интерфейса



Пример графического SVG-дашборда



Модуль расширения KNX



Подключение модуля ввода-вывода к контроллеру

Беспроводные интерфейсы

Модуль сотовой связи - модем 2G (GPRS), 3G (UMTS) или NB-IoT устанавливается в контроллер при производстве. Требуется SIM-карта формата microSIM.

Модем позволяет отправлять и принимать SMS, подключаться к интернету.

Модуль Wi-Fi в Wiren Board можно настроить на работу в одном из трёх режимов:

- режим точки доступа, включён по умолчанию (имя WirenBoard, без пароля, адрес контроллера в созданной сети: 192.168.42.1).

- режим клиента;
- одновременная работа в режиме и точки доступа, и клиента.

Модуль Bluetooth 4.0 (Bluetooth Low Energy) - можно отслеживать приближение других Bluetooth устройств - например, телефона владельца или метки.

USB-стик **Z-Wave** дает поддержку устройств стандарта [Z-Wave](#).

Пакетное радио 433 МГц - радиомодуль RFM69H устанавливается в контроллер при производстве.

Позволяет подключать к контроллеру устройства Noolite, датчики Oregon.

Антенны Wi-Fi, GSM и радио 315/433MHz подключаются к SMA разъемам.

При слабом сигнале GSM рекомендуется использовать выносную антенну и располагать ее вдали от контроллера.

Проводные интерфейсы

Интерфейс Ethernet поддерживает скорость 10/100 Мбит/с. В ревизиях с 5.8 есть второй порт Ethernet.

Также присутствует один порт USB 2.0. Работает в режиме USB Host или USB Device. Управление питанием отдельных USB-устройств описано в статье [Питание USB-портов](#).

Интерфейс RS-485 - стандарт коммуникации по двухпроводной шине.

Контроллер имеет 2 порта RS-485 + можно добавить еще 2 порта [модулями расширения RS-485](#).

Статья [RS-485: Физическое подключение](#) поможет вам правильно выбрать и проложить кабеля.

[Настройка подключения устройств](#) осуществляется в веб интерфейсе.

Стандартно в Wigen Board с подключёнными по RS-485 устройствами работает [Драйвер wb-mqtt-serial](#) через систему [MQTT-сообщений](#).

CAN - это стандарт коммуникации по двухпроводной шине. На контроллере мультиплексирован (выведен на те же клеммники) со вторым портом RS-485.

1-Wire - шина для подключения внешних датчиков по двум или трём проводам. Так как это шина, можно подключить несколько устройств на один порт 1-Wire. ПО контроллера поддерживает подключение температурных датчиков типа DS18B20.

Для питания датчиков удобно использовать выход +5V. Он защищен от КЗ и подачи повышенного напряжения. При питании контроллера от аккумулятора выход +5V остается активным. Так же есть программное управление этим выходом (его можно отключать).

Универсальные входы/выходы A1-A4

Универсальный канал Ax объединяет в себе три функции и может работать как:

- Выход "[открытый коллектор](#)" - ключ (3А/30В), замыкающий выход на землю.
- [Аналоговый вход](#) с диапазоном измерений 0-28 В.
- [Дискретный вход](#) - срабатывает при напряжении на клемме 5В.

Резистивные входы R1 и R2

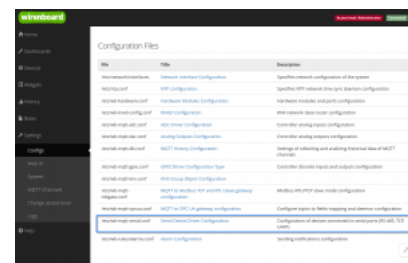
Могут работать в режимах:

- измерения сопротивления 1-30кОм
- измерения напряжение в диапазоне 0-3.0В
- цифровой вход

Вход R2 есть только в версии 5.3.

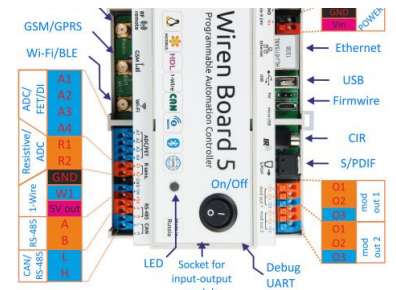
[ADC: Измерение сопротивлений - технические детали](#)

Клеммники

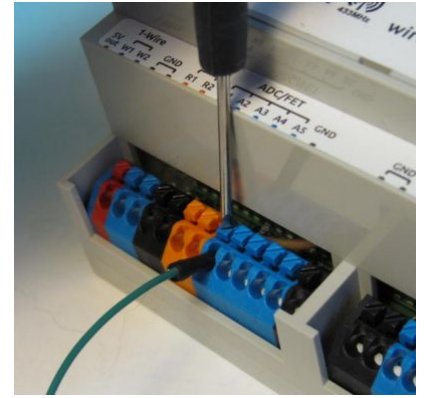


Настройка происходит через страницу [Configs](#) веб-интерфейса





Wiren Board 5



Работа с самозажимными клеммами

| подпись | Max. V, I | состояние по умолчанию | Функции |
|---------------------------------|-----------|------------------------|--|
| Верхний ряд, левый блок | | | |
| Vin | 28V | | Питание, защита от переплюсовки |
| GND | | | "земля", минус блок питания |
| Верхний ряд, правый блок | | | |
| O1-O3 | | | Входы/выходы 1-го модуля расширения |
| O1-O3 | | | Входы/выходы 2-го модуля расширения |
| Нижний ряд, правый блок | | | |
| A1-A4 | 28V , 2A | High Z | Выходы "открытый коллектор", ADC, цифровые входы, защита ключей от индуктивной нагрузки. |
| R1-R2* | 5V | High Z | Резистивные датчики, ADC, цифровые входы |
| GND | | | Для удобства подключения внешних датчиков |
| W1 | 30V | 3.3V | 1-Wire, GPIO |
| 5V out | 5V, 0.5A | 5V | Выход 5V. Защита от КЗ. Программное вкл/выкл. |
| A | 30 V | 0V | порт RS-485 (/dev/ttyAPP1) |
| B | 30 V | +5V | |
| L | 30 V | 0V | Порт CAN или RS-485 (/dev/ttyAPP4). Подключение RS-485: А к клемме L , В к клемме H . |
| H | 30 V | +5V | |
| Vout* | 1A | | Выход питания. Входное напряжение, подключенное через диод. |

* зависит от аппаратной ревизии

Все входы защищены от подачи напряжения питания (до 28 В) и импульсных помех.

В качестве интерфейсных клемм в контроллере применены клеммы "тип 250". Это самозажимные клипсы. При вставке очищенного одножильного провода в гнездо, он автоматически зажимается пружинной защёлкой. Для вставки многожильных проводов, необходимо отжать пружину, нажав на кнопку клипсы. Кнопка имеет паз под шлицевую отвертку. Для извлечения провода, нужно отжать пружину, нажав на кнопку клипсы, и вытащить провод.

Другие интерфейсы

Отладочный порт - подключившись к нему можно получить прямой доступ к консоли контроллера. Через него можно также взаимодействовать с загрузчиком и следить за загрузкой операционной системы (последовательная консоль, serial console).

"Пищалка" - издает звуковой сигнал, частота настраивается.

Часы реального времени RTC питаются от внутренней резервной батареи.

В контроллерах ревизии 5.3 и 5.6 так же есть:

- ИК-порт
- Аудиовыход цифрового звука - S/PDIF, разъём TOSLINK.

Сторожевой таймер

Отдельный аппаратный watchdog, перезагружающий контроллер целиком по питанию при зависании ПО.

Отключение аппаратного сторожевого таймера

Питание

На контроллере есть три внешних входа для подключения питания:

- DC jack - стандартный штекерный разъем (5.5x2.1мм) на левой стороне корпуса.
- Клеммники Vin и GND - в контроллерах ревизии 5.8 и выше: две клеммы Vin, к которым можно подключить два независимых блока питания для резервирования. Земли блоков питания должны быть соединены и подключены к клемме GND.
- Питание по кабелю Ethernet. Подробнее в Power over Ethernet.

Возможно одновременное подключения источников к разным входам, в том числе с разным напряжением.

Для резервного питания можно подключить внутренний модуль WBMZ-BATTERY с Li-Ion (Li-Pol) аккумулятором.

Допустимый диапазон питания **7-28В**. Среднее потребление платы - 1,5-2 Вт. Но т.к. модуль GSM потребляет импульсно до 8 Вт, рекомендуется использовать блоки питания с мощностью не менее **10 Вт**.

Поддерживаемые устройства

Устройства нашего производства с интерфейсом RS-485

Поддерживаемые устройства

Подключение периферийных устройств

Подробное тех.описание платы контроллера

В статье Wiren Board 5:Схемотехника описаны некоторые особенности работы и устройства входов, схемы питания контроллера.

Аппаратные ошибки/особенности Wiren Board 5 найденные при эксплуатации контроллера.

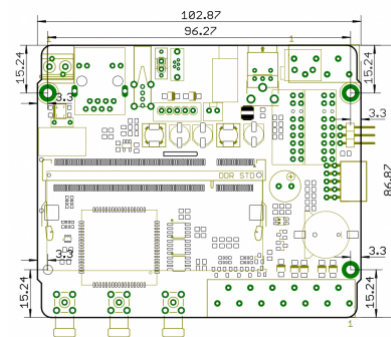
Wiren Board: Аппаратные ревизии - описание изменений в плате контроллера.

Таблицы соответствия GPIO процессора и сигналов на плате для ревизий:

rev. 5.5 (5.6, 5.6.1), rev. 5.8 (5.9), rev. 5.3

Работа с GPIO -как работать с GPIO напрямую.

Wiren Board 5:Восстановление пароля пользователя root



Размеры платы контроллера

Изображения и чертежи устройства

По ссылкам ниже вы можете скачать изображения и чертежи устройства **Wiren Board-5.8**.

Corel Draw 2018: [Wiren Board-5.8.cdr.zip](#)

Corel Draw PDF: [Wiren Board-5.8.cdr.pdf](#)

Autocad 2013 DXF: Мы еще не подготовили чертеж этого устройства. Вы можете запросить чертеж устройства "Контроллер Wiren Board 5.8" (<https://support.wirenboard.com/new-topic?category=featurerequests/blueprints&title=Чертеж%20устройства%20Wiren Board-5.8>) на портале техподдержки Wiren Board (необходима регистрация).

По ссылкам ниже вы можете скачать изображения и чертежи устройства **Wiren Board-5.9**.

Corel Draw 2018: [Wiren Board-5.9 new.cdr.zip](#)

Corel Draw PDF: [Wiren_Board-5.9_new.cdr.pdf](#)

Autocad 2013 DXF: Мы еще не подготовили чертеж этого устройства. Вы можете запросить чертеж устройства "Контроллер Wiren Board 5.9" (https://support.wirenboard.com/new-topic?category=featurerequests/blueprints&title=Чертеж%20устройства%20Wiren_Board-5.9) на портале техподдержки Wiren Board (необходима регистрация).

Contents

Описание

Примеры работы через MQTT

- [Получение значения от датчика температуры и вывод его в веб-интерфейс](#)
- [Нажатие кнопки в веб-интерфейсе и переключение реле на внешнем модуле](#)
- [Отображение устройств в структуре сообщений](#)
- [Пример подписки](#)
- [Структура сообщения — команды на изменение состояния](#)

Локальная работа с сообщениями MQTT

- [Работа из командной строки](#)
 - [Управление устройствами из командной строки](#)
 - [Слежение за состоянием устройства / подписка на топик](#)
- [Метасимволы](#)
- [Очистка очереди сообщений](#)
- [Работа с MQTT из внешних программ](#)
- [Просмотр MQTT-каналов в веб-интерфейсе](#)

Работа с сообщениями MQTT с внешнего устройства

- [Настройка моста с MQTT брокером Cloudmqtt](#)
- [Настройка моста с MQTT брокером Clusterfly](#)
- [Другие облачные брокеры](#)

Создание своего брокера MQTT

- [Установка брокера](#)
- [Настройка моста на контроллере](#)

Описание

MQTT — протокол обмена данными, использующийся в программном обеспечении [Wired Board](#). [Базовая информация по MQTT на Википедии \(http://en.wikipedia.org/wiki/MQTT\)](#).

Драйверы, которые отвечают за аппаратную часть контроллера (цифровые и транзисторные входы, АЦП и т.п.) и функции внешних подключенных устройств публикуют их состояние по MQTT в виде сообщений. Веб-интерфейс читает эти сообщения и на их основе отображает состояние устройств.

Действия пользователя в веб-интерфейсе также публикуются по MQTT, где их получает драйвер и передает команду пользователю устройства.

Через MQTT работают: веб-интерфейс, движок правил и встроенные драйверы. Если вы разрабатываете собственное ПО в дополнение к предустановленному — мы рекомендуем использовать MQTT.

Примеры работы через MQTT

Получение значения от датчика температуры и вывод его в веб-интерфейс

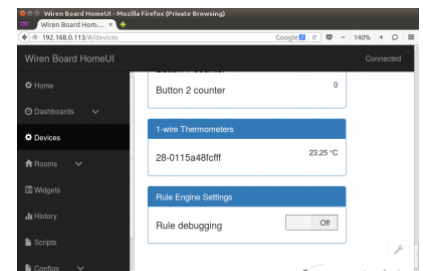
К [Wired Board](#) подключён датчик температуры по шине [1-Wire](#). Проследим, как данные с него через MQTT попадают в веб-интерфейс:

- Драйвер [wb-mqtt-w1](https://github.com/wireboard/wb-mqtt-w1) (<https://github.com/wireboard/wb-mqtt-w1>), отвечающий за данную аппаратную функцию, опрашивает подключенные к контроллеру датчики 1-Wire.
- При получении значения драйвер публикует MQTT сообщение вида:

```
/devices/wb-w1/controls/28-0115a48fcfff 23.25
```

Оно значит, что от устройства 1-Wire с идентификатором `28-0115a48fcfff` получено значение `23.25 °C`.

- Веб-интерфейс, который подписан на все сообщения из MQTT, получает это сообщение и выводит значение датчика на страницу.



Показание датчика и его уникальный идентификатор на странице *Devices* веб-интерфейса

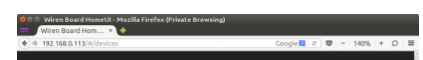
Нажатие кнопки в веб-интерфейсе и переключение реле на внешнем модуле

К контроллеру по шине RS-485 подключён релейный модуль [WB-MRM2](#). Пользователь в веб-интерфейсе нажимает кнопку включения реле. Проследим, как команда из веб-интерфейса попадает на внешний модуль:

- После нажатия кнопки веб-интерфейс публикует по MQTT сообщение вида:

```
/devices/wb-mrm2_130/controls/Relay 1/on 1
```

Оно значит, что устройство [WB-MRM2](#) с адресом `130` должно перевести *Реле 1* в



Это означает, что не удалось получить температуру термометра с адресом `28-0115a48fcfff`.

Драйвер `wb-mqtt-serial` устанавливает признак `r`, если не удалось запросить значение параметра устройства, признак `w` — не удалось передать значение устройству.

Драйвер `wb-mqtt-adc` (<https://github.com/wirenboard/wb-homa-adc>) устанавливает признак `r`, если не удалось получить значение соответствующего канала АЦП.

Пример подписки

Клиенты, которые хотят следить за значением температуры, «подписываются» на этот топик, и им приходят все новые сообщения — меняющиеся значения температуры. Один из таких клиентов — веб-интерфейс.

Подписаться на сообщения можно и из консоли Linux при помощи утилиты `mosquitto_sub`:

```
~# mosquitto_sub -t '/devices/wb-w1/controls/28-0115a48fcfff' -v //получить сообщения из топика устройства 1-Wire с идентификатором 28-0115a48fcfff
/devices/wb-w1/controls/28-0115a48fcfff 22.75 //в этой строке и ниже - вывод утилиты, полученные сообщения
/devices/wb-w1/controls/28-0115a48fcfff 22.75
/devices/wb-w1/controls/28-0115a48fcfff 22.75
```

Полное описание работы с MQTT из командной строки смотрите ниже.

Структура сообщения — команды на изменение состояния

Подпишемся на сообщения о состоянии первого реле подключённого по RS-485 релейного модуля WB-MRM2:

```
~# mosquitto_sub -t "/devices/wb-mrm2_130/controls/Relay 1/#" -v
/devices/wb-mrm2_130/controls/Relay 1/meta/type switch
/devices/wb-mrm2_130/controls/Relay 1/meta/order 1
/devices/wb-mrm2_130/controls/Relay 1 0
```

Тут стоит отметить, что MQTT сохраняет часть сообщений (а именно те, которые при отправке были помечены флагом *retained*) вечно, поэтому после подписки вы получите даже те сообщения, которые были отправлены раньше, чем вы подписались.

Релейный модуль управляется драйвером `Драйвер wb-mqtt-serial`. У него есть соответствующий топик-«канал» («контроль») *Relay 1*. У него самого есть значение — `0` (реле выключено), и есть два подтопика. Один из них — служебный: в `/meta/type` записан тип «контроля». Здесь он *switch* — выключатель. Второй подтопик `/on` — интереснее: в него клиенты пишут то состояние, в которое они хотят установить реле. Заметим, что оно может не совпадать некоторое время (затрачиваемое на процесс переключения) с тем состоянием, в котором реле находится. Драйвер при этом ведёт себя следующим образом: при получении сообщения в топик `/devices/wb-mrm2_130/controls/Relay 1/on` он физически включает реле на релейном модуле, а лишь затем записывает новое состояние реле в топик `/devices/wb-mrm2_130/controls/Relay`.

Например, если мы сейчас нажмём на кнопку реле в веб-интерфейсе (переключим его состояние), то получим новые сообщения:

```
/devices/wb-mrm2_130/controls/Relay 1/on 1
/devices/wb-mrm2_130/controls/Relay 1 1
```

- веб-интерфейс сначала «даёт указание» включить реле, потом драйвер его включает и записывает актуальное состояние в «канал» («контроль»).

Локальная работа с сообщениями MQTT

Программа (демон), отвечающая за рассылку сообщений от одних клиентов другим, называется брокером сообщений. В Wiren Board используется брокер сообщений `Mosquitto` (<http://mosquitto.org/>). Фактически, все драйверы и веб-интерфейс передают свои сообщения именно демону-брокеру `Mosquitto`.

Работа из командной строки

Управление устройствами из командной строки

Для управления устройством (изменения значения канала), необходимо отправить сообщение в топик `/devices/<device-id>/controls/<control-id>/on` (обратите внимание на `/on` в конце). Это делается с помощью консольной команды `mosquitto_pub`. Пример:

```
~# mosquitto_pub -t "/devices/wb-mrm2_130/controls/Relay 1/on" -m "1"
```

команда отправляет сообщение «1» (логическую единицу, «включить») в топик, соответствующий подключённому по RS-485 релейному модулю WM-MRM2 с адресом 130.

Слежение за состоянием устройства / подписка на топик

Клиенты, которые хотят следить за значением температуры, «подписываются» на этот топик, и им приходят все новые сообщения - меняющиеся значения температуры. Один из таких клиентов - веб-интерфейс.

Подписаться на сообщения можно и из консоли Linux при помощи утилиты **mosquitto_sub** (полное описание утилиты смотрите на http://mosquitto.org/man/mosquitto_sub-1.html (http://mosquitto.org/man/mosquitto_sub-1.html)):

```
~# mosquitto_sub -t '/devices/wb-w1/controls/28-0115a48fcfff' -v //получить сообщения из топика устройства 1-Wire с идентификатором 28-0115a48fcfff
/devices/wb-w1/controls/28-0115a48fcfff 22.75 //в этой строке и ниже – вывод утилиты, полученные сообщения
/devices/wb-w1/controls/28-0115a48fcfff 22.75
/devices/wb-w1/controls/28-0115a48fcfff 22.75
```

Метасимволы

Подписаться можно не только на один топик, но и на группу топиков по метасимволу. В MQTT применяется два метасимвола: **#** и **+**. Метасимвол **#** означает любое количество уровней вложенных топиков. Выполним команду

```
~# mosquitto_sub -t '/devices/wb-w1/#' -v
/devices/wb-w1/meta/name 1-wire Thermometers
/devices/wb-w1/controls/28-0115a48fcfff 22.812
/devices/wb-w1/controls/28-0115a48fcfff/meta/type temperature
/devices/wb-w1/controls/28-0115a48fcfff 22.75
```

В результате мы получили не только значения с «контроля» устройства, но и топика с метаданными — название драйвера устройства и тип «контроля» - *temperature*. Существует так же метасимвол **+**, который обозначает один уровень, а не произвольное количество, как **#**:

```
mosquitto_sub -v -t "/config/widgets+/name"
```

В этом случае мы получим имена всех виджетов.

Полное описание системы топиков и подписок (<http://mosquitto.org/man/mqtt-7.html>).

Очистка очереди сообщений

Ненужные retained-сообщения могут остаться в системе MQTT после удаления неиспользуемых драйверов или отключения каких-либо устройств. Это приводит к тому, что несуществующие больше устройства могут отображаться в разделе *Devices* веб-интерфейса.

Для удаления топиков можно воспользоваться командой `mqtt-delete-retained`.

Например, удалим все топика, начинающиеся на `/devices/noolite_tx_1234/`

```
~# mqtt-delete-retained '/devices/noolite_tx_1234/#'
```

Для удаления топиков «по маске», можно циклично вызывать `runShellCommand` из правил. Таким образом, задача сводится к задаче работы со строками в `js`.

```
var deviceName = ['name1', ..., 'nameN'];
var controlName = 'Temperature';

for (var i = 0; i < deviceName.length; i++) {
  runShellCommand ('mqtt-delete-retained /devices/' + deviceName[i] + '/controls/controlName/#');
}
```

Работа с MQTT из внешних программ

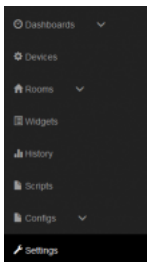
Если вы разрабатываете собственное ПО для Wiren Board, взаимодействовать с его аппаратными ресурсами лучше всего через протокол MQTT — ваша программа передаёт сообщение по MQTT, драйвер управляет устройством и вашей программе не нужно напрямую взаимодействовать с устройством на низком уровне.

Для того, чтобы отправлять сообщения MQTT, для многих языков программирования есть библиотеки:

- Python - [1] (<https://github.com/contactless/mqtt-tools>)
- C - [2] (<http://mosquitto.org/man/libmosquitto-3.html>)

Просмотр MQTT-каналов в веб-интерфейсе

MQTT-названия устройств, их элементов управления и последние значения можно найти в разделе **Settings** веб-интерфейса:



| Device | Control | Type | Topic | Value | Status |
|-----------------|-------------|------------|---|--------------|--------|
| alarms | log | text (ro) | idevices/alarms/controls/log | 0 | OK |
| buzzer | enabled | switch | idevices/buzzer/controls/enabled | 0 | OK |
| buzzer | frequency | range | idevices/buzzer/controls/frequency | 3000 | OK |
| buzzer | volume | range | idevices/buzzer/controls/volume | 10 | OK |
| network | Ethernet IP | text | idevices/network/controls/Ethernet IP | 192.168.0.37 | OK |
| network | GPRS IP | text | idevices/network/controls/GPRS IP | | OK |
| network | Wi-Fi IP | text | idevices/network/controls/Wi-Fi IP | 192.168.42.1 | OK |
| nooite_bt_0x4d7 | bind | pushbutton | idevices/nooite_bt_0x4d7/controls/bind | 0 | OK |
| nooite_bt_0x4d7 | color | rgb | idevices/nooite_bt_0x4d7/controls/color | 0.0.0 | OK |

Информация об MQTT-названиях устройств

Работа с сообщениями MQTT с внешнего устройства

Установленный на контроллер брокер mosquitto по умолчанию принимает подключения внешних клиентов по порту 1883 без пароля.

Например, если контроллер имеет адрес 192.168.0.67, его топики можно прочесть с другого компьютера с Linux, находящегося в той же сети:

```
mosquitto_sub -h 192.168.0.67 -p 1883 -v -t "/devices/power_status/controls/Vin" -i Test_Client
```

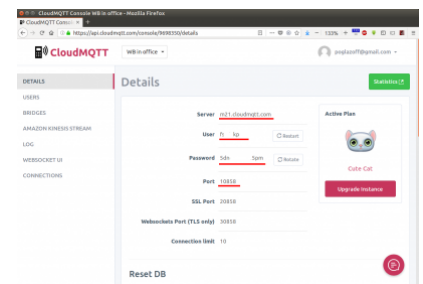
Альтернативный вариант — использовать [MQTT Explorer](#).

=== Настройка MQTT моста (bridge) ===

MQTT мост (bridge) — это функция MQTT-брокера, позволяющая пересылать все или часть сообщений на другой MQTT-брокер, и получать сообщения с другого брокера обратно.

Эту функцию удобно применять в следующей ситуации: хотя на самом контроллере уже есть MQTT-брокер, к нему часто неудобно подключаться, так как контроллер может не иметь белого IP-адреса, а иногда может быть выключен или не в сети. В таком случае удобно иметь отдельный брокер в облаке с фиксированным адресом, который будет всегда онлайн, и на который будут пересылаться сообщения с брокера контроллера.

На контроллере эта функция настраивается в конфигурационных файлах *mosquitto*. Самый простой вариант конфигурации приведён ниже.



Настройки брокера Cloud MQTT

Настройка моста с MQTT брокером Cloudmqtt

Задача: настроить пересылку всех сообщений MQTT на популярный дешёвый облачный MQTT брокер <http://cloudmqtt.com/> и обратно.

Решение:

1. Зарегистрируйтесь на <http://cloudmqtt.com/>
2. Зайдите в свой аккаунт на <http://cloudmqtt.com/> и посмотрите настройки: сервер, порт, логин, пароль.
3. Зайдите на контроллер и добавьте в конец файла `/etc/mosquitto/mosquitto.conf` следующие строки:

```
connection cloudmqtt
address m21.cloudmqtt.com:10858
remote_username fs_user_kp
remote_password 5dn_pass_pm
clientid pavel_test
try_private false
start_type automatic
topic # both
```

(последняя строка говорит, что нужно пересылать все сообщения (метасимвол #, смотрите описание выше) в обе (**both**) стороны (с брокера контроллера на облачный брокер и обратно)

Более подробное описание всех опций смотрите на <https://mosquitto.org/man/mosquitto-conf-5.html>.

4. Перезапустите *mosquitto*, выполнив в консоли:

```
service mosquitto restart
```

Настройка моста с MQTT брокером Clusterfly

Задача: настроить пересылку всех сообщений MQTT на бесплатный облачный MQTT брокер <https://clusterfly.ru/> и обратно.

Решение:

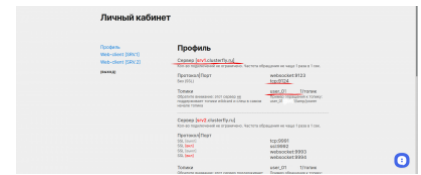
1. Зарегистрируйтесь на <https://clusterfly.ru/>
2. Зайдите в свой аккаунт на <https://clusterfly.ru/> и выберите "Профиль" посмотрите



настройки: сервер, порт, логин и сгенерируйте пароль. для пересылки используйте сервер `srv1.clusterfly.ru`.

- Зайдите на контроллер и добавьте в конец файла `/etc/mosquitto/mosquitto.conf` следующие строки:

```
connection clusterfly
address srv1.clusterfly.ru:9124
remote_username user_XXXXXX
remote_password pass_XXXXXX
try_private false
notifications true
notification_topic /client/wb_6/bridge_status
start_type automatic
topic /# both 0 "" user_XXXXXX
bridge_insecure true
cleansession false
```



Настройки брокера CLUSTERFLY

строка `'topic /# both 0 "" user_XXXXXX'` говорит, что нужно пересылать все сообщения (метасимвол `#`, смотрите описание выше) в обе (**both**) стороны (с брокера контроллера на облачный брокер и обратно) с префиксом (**user_XXXXXX**). Пример обращения к топике: `user_XXXXXX/devices/wb-тг6с_200/controls/K2`.

- Перезапустите `mosquitto`, выполнив в консоли:

```
service mosquitto restart
```

Потребуется подождать некоторое время пока брокер `mosquitto` сможет организовать соединение. Подписавшись на контроллере к топике `/client/wb_6/bridge_status` можно увидеть статус соединения.

```
mosquitto_sub -v -t "/client/wb_6/bridge_status"
/client/wb_6/bridge_status 0
/client/wb_6/bridge_status 1
```

Другие облачные брокеры

Список облачных брокеров, в том числе бесплатных: https://github.com/mqtt/mqtt.github.io/wiki/public_brokers (https://github.com/mqtt/mqtt.github.io/wiki/public_brokers)

Задача: настроить пересылку топика MQTT на другой контроллер. Есть два контроллера в одной сети:

- DestinationController* с адресом `10.0.0.40`, на этот контроллер получать топик.
- SourceController* с адресом `10.0.0.70`, с этого контроллера будем забирать топик.

На *SourceController* есть `/client/temp1`, но его нужно видеть на *DestinationController* в `/devices/temp1`.

Решается двумя способами, можно с *SourceController* публиковать на *DestinationController* или с *DestinationController* подписаться на топик *SourceController* и забирать изменения. От выбора стратегии зависит на каком контроллере будем проводить настройки.

Мы будем настраивать *DestinationController*.

Решение: На контроллере *DestinationController* добавьте в конфиг:

```
mcedit /etc/mosquitto/conf.d/bridge.conf
```

Строки:

```
connection wb_40
address 10.0.0.70
notifications true
notification_topic /client/wb_40/bridge_status
keepalive_interval 20
restart_timeout 20

topic /temp1/# in 2 /devices /client
```

Перезапустите `mosquitto` на *DestinationController*:

```
systemctl restart mosquitto; systemctl status mosquitto
```

ВАЖНО: перед перезапуском желательно остановить watchdog. В случае ошибки в конфигурационных файлах брокер не запустится и `watchdog` вызовет перезапуск контроллера.

Рассмотрим подробнее строчку `topic /temp1/# in 2 /devices /client` где:

- `/temp1/#` это топик от «корня». На брокере-источнике `/client/temp1`.
- `in` — только забираем, изменения на контроллере не передадутся на сервер.
- `/devices` — «корень» **куда** располагаем локально (на контроллере на котором **настраиваем**). На контроллере *DestinationController* это `/devices` и полный путь будет выглядеть как `/devices/temp1`.
- `/client` — «корень» откуда забираем на удаленном. На контроллере *SourceController* это `/client` и полный путь будет выглядеть как `/client/temp1`.

Проверка: Дожидаемся статуса бриджа «1» в топике /client/wb_40/bridge_status на контроллере *SourceController*. На нем же публикуем:

```
for i in {1..25}
do
mosquitto_pub -t "/client/temp1/temp" -m "$i" -r
done
```

Подписавшись на контроллере *DestinationController* на целевой топик можно видеть:

```
mosquitto_sub -v -t /devices/temp1/#
/devices/temp1/temp
/devices/temp1/temp 1
/devices/temp1/temp 2
/devices/temp1/temp 3
/devices/temp1/temp 4
/devices/temp1/temp 5
/devices/temp1/temp 6
/devices/temp1/temp 7
/devices/temp1/temp 8
/devices/temp1/temp 9
/devices/temp1/temp 10
/devices/temp1/temp 11
/devices/temp1/temp 12
/devices/temp1/temp 13
/devices/temp1/temp 14
/devices/temp1/temp 15
/devices/temp1/temp 16
/devices/temp1/temp 17
/devices/temp1/temp 18
/devices/temp1/temp 19
/devices/temp1/temp 20
/devices/temp1/temp 21
/devices/temp1/temp 22
/devices/temp1/temp 23
/devices/temp1/temp 24
/devices/temp1/temp 25
```

Создание своего брокера MQTT

Вы можете создать отдельный брокер на компьютере или на VDS-сервере в интернете и собирать на нем данные с контроллеров.

Инициировать соединение будет контроллер, поэтому контроллеру не нужен «белый» IP-адрес. Если контроллеров несколько, вы можете разделить данные от них на брокере, для этого в настройках моста укажите для каждого контроллера отдельный корневой топик.

Установка брокера

1. Установите mosquitto:

```
sudo apt update && sudo apt install mosquitto mosquitto-clients -y
```

2. Отключите возможность анонимного входа, для этого:

- Откройте файл конфигурации в редакторе

```
sudo nano /etc/mosquitto/mosquitto.conf
```

- Добавьте в конец файла строки:

```
#Turn on port listening
listener 1883
#Disable anonymous login:
allow_anonymous false
#Password file:
password_file /etc/mosquitto/mosquitto.pwd
```

3. Создайте пароль для пользователя, в примере использован пользователь test с паролем wbpasword:

```
sudo mosquitto_passwd -c /etc/mosquitto/mosquitto.pwd test
```

4. Введите пароль дважды и запомните его, он вам пригодится ниже.

5. Перезапустите mosquitto и проверьте его состояние:

```
sudo systemctl restart mosquitto && sudo systemctl status mosquitto
```

6. Подключитесь к брокеру для проверки, в примере адрес брокера 127.0.0.1:

```
mosquitto_sub -v -h 127.0.0.1 -u test -P wbpasword -t "/#"
```

7. Запустите в другой консоли команду ниже и убедитесь, что топик меняется:

```
for i in {1..25}; do mosquitto_pub -h 127.0.0.1 -u test -P wpassword -t "/client/templ/temp" -m "$i" -r; done
```

Брокер установлен и доступен с контроллера. Для подключения нужно ввести логин и пароль.

Настройка моста на контроллере

Создайте файл конфигурации моста, для этого:

1. Создайте файл `/etc/mosquitto/conf.d/bridge1.conf`

```
nano /etc/mosquitto/conf.d/bridge1.conf
```

2. Вставьте в него строки, где `10.0.0.105` — адрес брокера:

```
connection bridge1
#address of server
address 10.0.0.105
notifications true
notification_topic /clientnotification/bridge1_status
remote_username test
remote_password wpassword

topic /templ/# both 2 /devices /controller
```

Содержимое топика `/devices/templ/#` контроллера будет отображаться на брокере в `/controller`. Вместо `/controller` можете указать уникальное имя контроллера, например, серийный номер.

Contents

Введение

Подготовка

- [Смена уровня доступа к веб-интерфейсу](#)
- [Требования к изображению](#)

Создание SVG-панели

Синтаксис

- [Значение MQTT-топика](#)
- [Условные выражения](#)
- [Арифметические операции](#)
- [Округление значений](#)

Редактор связей

- [Read](#)
- [Write](#)
- [Visible](#)
- [Style](#)

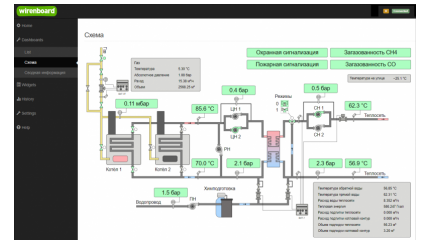
Редактирование

- [Панель](#)
- [SVG-изображение](#)

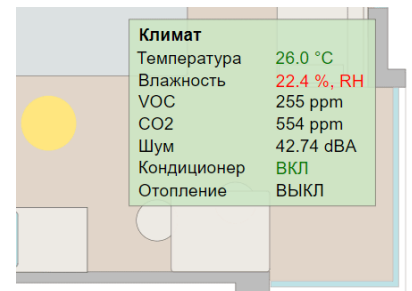
Перенос на другой контроллер

- [Сохранение](#)
- [Загрузка](#)

Полезные ссылки и материалы



Пример SVG-панели



Вывод текстовой информации, статусов и взаимодействие с элементом

Введение

В веб-интерфейсе контроллера Wipro Board можно создавать интерактивные графические панели. С их помощью удобно визуализировать схемы управления автоматикой.

Возможности:

- вывод текстовой или логической информации;
- изменение внешнего вида svg-элемента в зависимости от полученной информации: вы можете изменять стиль элемента, подменить полученное значение своим или скрыть/показать элементы;
- воздействие на исполнительные механизмы: включать и выключать нагрузку, менять режимы работы устройств.

SVG-панели работают на мобильных устройствах, но нужно учитывать размер изображения. Если элементы будут слишком мелкими — ими будет сложно пользоваться. Хорошим решением может стать разделение графических панелей на десктопные и мобильные, которые созданы с учетом особенностей мобильного устройства.

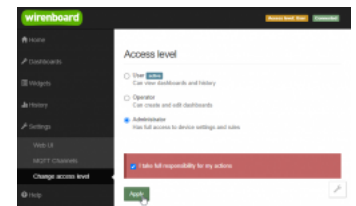
Подготовка

Смена уровня доступа к веб-интерфейсу

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.



Уровень «Администратор»

Требования к изображению

Основой для SVG-панели служит svg-изображение. Если готового изображения нет, то его можно нарисовать в любом векторном редакторе, например, в свободном [Inkscape](https://inkscape.org/ru/) (<https://inkscape.org/ru/>).

Вы можете выводить информацию в любые текстовые элементы и настраивать взаимодействия с любыми фигурами и текстом.

Ограничения:

- Нельзя взаимодействовать с прямоугольником. Если это необходимо — преобразуйте его в кривые (окоптурите).
- Редактор SVG-панели игнорирует группировку. Поэтому если нужно настроить взаимодействие с составным элементом, то преобразуйте его в одну фигуру, состоящую из контуров.
- Внутри svg-файла тег *content* не должен содержать другие теги. Если вы загрузили файл на контроллер, сделали привязку, скачали этот файл на компьютер и он не открылся в редакторе, откройте svg-файл текстовым редактором и удалите содержимое тега *content*. После этого вы сможете редактировать файл в svg-редакторе.

удалите содержимое тега `content`. После этого вы сможете редактировать файл в SVG-редакторе.

Изображение может быть любого размера и с любым количеством элементов.

Создание SVG-панели

Создайте новую SVG-панель:

1. Перейдите в раздел **Dashboards** → **List**.
2. В разделе **SVG Dashboards** нажмите кнопку **Add**, откроется страница **New SVG Dashboard**.
3. Выберите файл с svg-изображением и нажмите кнопку **Upload**.

После загрузки изображение отобразится на открытой странице:

1. Укажите связи и поведение svg-элементов с помощью редактора связей.
2. Сохраните изменения и посмотрите результат, для этого нажмите на кнопку **View dashboard**.

Не забудьте указать в полях **ID** и **Name** уникальный ID новой панели и имя.

Если нужно вписать дашборд в размер окна браузера, то можно использовать один из двух способов:

1. Опция в настройках дашборда **Full width** растянёт изображение по всю ширину.
2. Прописать размеры в SVG-файле. В Inkscape это можно сделать так:
 - Выберите на верхней панели **Посмотреть и изменить XML-дерево документа** (Меню **Edit - XML editor**), справа появится редактор XML.
 - Выделите самый первый тег `<svg>` и измените значение атрибутов `width` и `height`: `width: auto; height: 80vh`.
 - Сохраните документ.

Мы рекомендуем второй способ, так как он позволяет подстраивать автоматически не только ширину, но и высоту рисунка.

Синтаксис

В полях **Value** Редактора связей можно составлять простое выражение, которые содержат условия и простые арифметические операции.

Значение MQTT-топика

Значение из MQTT-топика, который указан в поле **Channel** помещается в переменную **val**.

Условные выражения

Доступные операции:

- `>` — больше
- `>=` — больше или равно
- `<` — меньше
- `<=` — меньше или равно
- `==` — равно
- `!=` — не равно

Синтаксис:

```
(УСЛОВИЕ) ? 'ЗНАЧЕНИЕ1' : 'ЗНАЧЕНИЕ2'
```

Например, если значение MQTT-топика будет равно 1, то вывести значение ВКЛ, иначе — ОТКЛ:

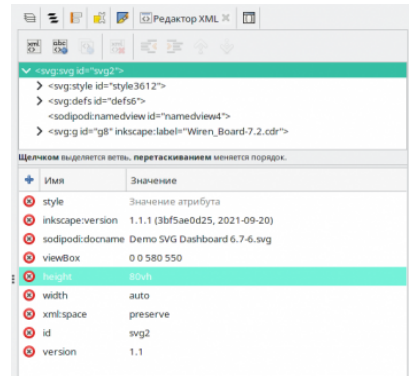
```
(val == 1) ? 'ВКЛ' : 'ОТКЛ'
```

Арифметические операции

Над полученным из MQTT-топика можно совершать простые арифметические операции:

- `+` — сложение
- `-` — вычитание
- `*` — умножение
- `/` — деление

Например, выведем значение АВАРИЯ, если значение в MQTT-топике, умноженное на 0.1 больше 20 и НОРМА в остальных



Настройка ширины и высоты дашборда в редакторе Inkscape

например, выводом значения 'АВАРИЯ', если значение в MQTT-топике, умноженное на 0.1 больше 20 и 'НОРМА' в остальных случаях:

```
(val*0.1>20) ? 'АВАРИЯ' : 'НОРМА'
```

Округление значений

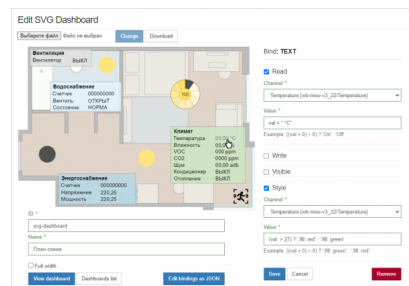
Часто с датчиков поступают значения с несколькими знаками после запятой, если вам не нужна такая точность в svg-панели, то их можно округлить:

```
val.toFixed(n) //округлить значение переменной val до n знаков после запятой  
val.toFixed(2) //округлить значение переменной val до двух знаков после запятой 220.238 -> 220.24
```

Редактор связей

Чтобы связать svg-элемент изображения со значениями или событиями, выберите его мышкой, после этого рядом с изображением появятся поля настройки поведения элемента.

После настройки выбранного svg-элемента нажмите кнопку **Save**.



Редактор SVG-панели

Read

Отображение значений из источника в текстовом поле.

Значение из выбранного в поле **Channel** источника помещается в переменную **val**, которую нужно указать в поле **Value**.

Чтобы вывести значение, просто укажите переменную:

```
val
```

К значению можно добавить произвольный текст, например, единицу измерения:

```
val + ' °C'
```

Также можно подменить полученное значение, например, заменить его на понятный человеку статус:

```
(val == 1) ? 'ВКЛ' : 'ОТКЛ'
```

Write

Реакция на клик пользователя по элементу изображения. Доступно два состояния ON и OFF. При клике пользователя на элементе, значение в MQTT-топике будет меняться на противоположное.

Выберите в поле **Channel** MQTT топик для записи значений, а в полях **Value on** и **Value off** укажите какие значения соответствуют состояниям.

Например, для переключения реле нужно указать:

```
Value on = 1  
Value on = 0
```

Visible

Здесь можно указать при каком условии показывать элемент изображения. По умолчанию элемент будет скрыт и отобразится только пока выполняется условие.

Выберите в поле **Channel** MQTT-топик, на информацию из которого будет реагировать элемент, в поле **Condition** выберите условие, а в поле **Value** — значение.

Style

Позволяет изменять стиль элементов изображения. Можно использовать для создания обратной связи для действий, оповещения пользователя о событии т.д.

Доступные для стилизации свойства элемента:

- fill — цвет заливки;
- fill-opacity — непрозрачность заливки, значения от 0.0 до 1.0 или в процентах;
- stroke — цвет обводки;

- `stroke-width` — толщина обводки;
- `stroke-dasharray` — вид пунктирной обводки, задается в единицах длины или процентах.
- `stroke-opacity` — непрозрачность обводки, значения от 0.0 до 1.0 или в процентах.

Цвета можно задавать именами `black`, `red`, `yellow` и т.п. или шестнадцатеричными значениями `#ffe680ff`, `#4f4d4dff` и т.п..

Изменим цвет элемента в зависимости от состояния реле, если в топике значение 1, то закрасиваем объект желтым, иначе — серым:

```
(val == 1) ? 'fill: yellow' : 'fill: gray'
```

Сделаем так, чтобы при значении больше 20 обводка элемента стала красного цвета, пунктиром и толщиной 2px. При значениях меньше или равно 20 стиль элемента оставим по умолчанию:

```
(val > 20) ? 'stroke: red; stroke-width:2px; stroke-dasharray:2,1' : ''
```

Редактирование

Панель

Чтобы внести изменения:

1. Откройте SVG-панель.
2. Нажмите кнопку **Edit dashboard**.
3. Внесите изменения и сохраните их нажатием на кнопку **View dashboard**.

SVG-изображение

При загрузке в редактор SVG-панели исходное изображение преобразуется: группы разбираются на составные элементы, в метаданные элементов прописываются ID, поэтому для редактирования изображения нужно его загрузить с контроллера на компьютер:

1. Откройте SVG-панель для редактирования.
2. В верхней части окна нажмите кнопку **Download**, на компьютер загрузится svg-файл.

При редактировании учтите, что если на svg-элемент была назначена связь, то он содержит скрытый ID и при его дублировании создается новый элемент с тем же ID.

Отредактируйте полученный svg-файл и вновь загрузите его на контроллер:

1. В окне редактирования нажмите кнопку **Выберите файл**.
2. После того, как кнопка **Change** станет активной, нажмите на нее.

Перенос на другой контроллер

Иногда нужно сделать резервную копию svg-панели, например, для переноса на другой контроллер. Для этого нужно сохранить преобразованный редактором svg-файл и описание его связей и загрузить их на другой контроллер.

Учтите, что на имена MQTT-топиков на обоих контроллерах должны совпадать. Если это не так — svg-элементы нужно будет привязать к новым MQTT-топикам.

Сохранение

Сохраните svg-файл и описание связей:

1. Загрузите файл на компьютер, как это описано в разделе [Редактирование](#).
2. В редакторе svg-панели нажмите кнопку **Edit bindings as JSON**, скопируйте и сохраните в файл содержимое поля **SVG Bindings**.

Загрузка

1. Создайте новую SVG-панель по инструкции в разделе [Создание SVG-панели](#).
2. В редакторе svg-панели нажмите кнопку **Edit bindings as JSON**, скопируйте из сохраненного файла описание связей и вставьте текст в поле **SVG Bindings**.
3. Нажмите кнопку **Save**.

Полезные ссылки и материалы

- [Архив с примерами SVG-панелей](#)
- [Статья о веб-интерфейсе контроллера Wiren Board \(https://wirenboard.com/ru/pages/wb-software/\)](https://wirenboard.com/ru/pages/wb-software/)

- [Описание веб-интерфейса контроллера на вики](#)
- [Свободный SVG-редактор Inkscape \(https://inkscape.org/ru/\)](https://inkscape.org/ru/)

- English
- русский

Contents

Возможности

Как зайти в веб-интерфейс

Работа с веб-интерфейсом

Разделы интерфейса

[Home \(Главная страница\)](#)

[Dashboards \(Панели\)](#)

[Devices \(Устройства\)](#)

[Widgets \(Виджеты\)](#)

[Пример создания виджетов](#)

[History \(История показаний\)](#)

[Rules \(Правила-скрипты\)](#)

[Settings -> Configs \(Настройки -> Конфигурирование\)](#)

[Settings -> WebUI \(Настройки -> Веб-интерфейс\)](#)

[Settings -> System \(Настройки -> Системные\)](#)

[Settings -> MQTT Channels \(Настройки -> MQTT-каналы\)](#)

[Settings -> Change access level \(Настройки -> Права доступа\)](#)

[Settings -> Logs \(Настройки -> Логи\)](#)

Стандартные задачи, решаемые через веб-интерфейс

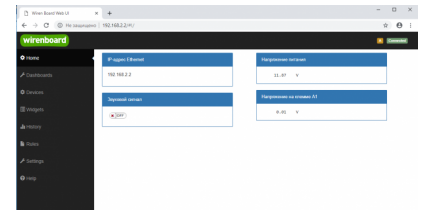
[Подключить устройство RS-485 Modbus и создать кнопки управления на главной панели](#)

[Обновить прошивку контроллера](#)

Облачный интерфейс

Настройка авторизованного доступа к веб-интерфейсу контроллера

Основные отличия версии 2.x от 1.0



Главная страница веб-интерфейса

Возможности

Контроллер Wiren Board имеет встроенный веб-интерфейс. Через интерфейс можно:

- следить за состоянием контроллера и подключённых устройств и управлять ими;
- подключать устройства к контроллеру;
- настраивать контроллер и обновлять его ПО;
- писать правила на встроенном движке;
- настраивать SMS и email-уведомления;
- смотреть на графике историю значений (например, температуры).

Веб-интерфейс работает непосредственно на Wiren Board. В качестве веб-сервера работает [nginx](#), сайт взаимодействует с MQTT через [WebSocket](#).

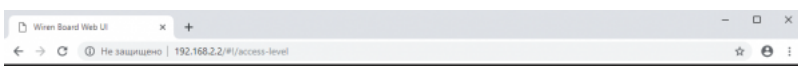
Ниже дано описание версии 2.0. Про предыдущую версию можно найти на странице [Веб-интерфейс Wiren Board 1.0](#).

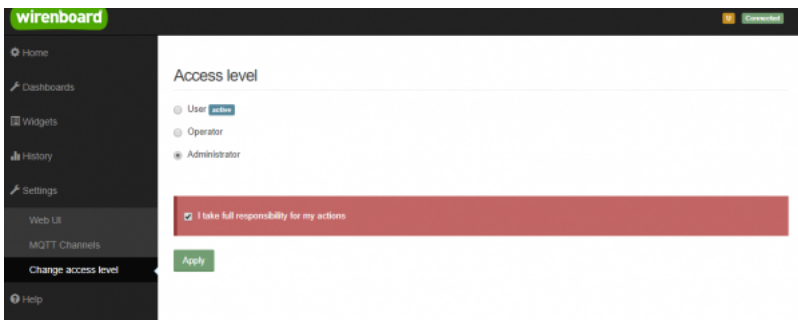
Как зайти в веб-интерфейс

Чтобы зайти в веб-интерфейс контроллера Wiren Board, введите в адресную строку браузера IP-адрес контроллера.

Если вы находитесь в одной сети с контроллером и используете устройства Apple, компьютер с Linux или Windows 10 и выше — введите в адресную строку `wirenboard-XXXXXXX.local`, где XXXXXXXX — восьмизначный серийный номер контроллера.

Работа с веб-интерфейсом





Выбор уровня доступа

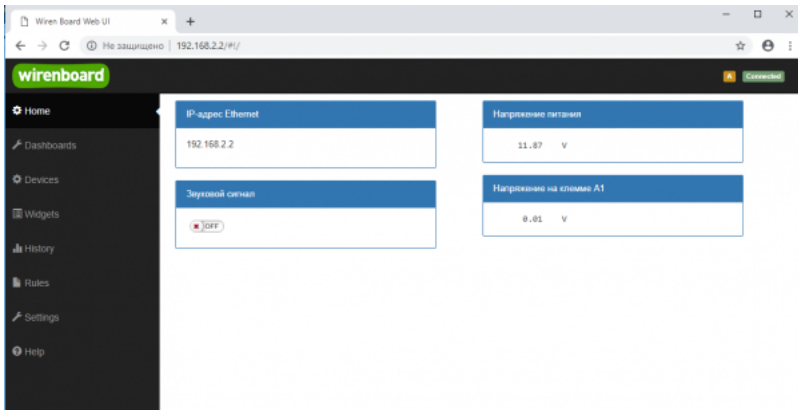
Для начала работы с веб-интерфейсом выберите уровень доступа. Для этого на вкладке **Settings -> Change access level -> Access Level** выберите один из пунктов **User**, **Operator** или **Administrator**. Уровни ограничивают доступ к функционалу веб-интерфейса: например, пользователь **User** может просматривать только настроенные виджеты, их редактирование и изменение настроек контроллера недоступно. Пользователь **Operator** получает доступ к контролам устройств, управляемых контроллером, может добавлять виджеты в панели (dashboards) (см. далее). Пользователь **Administrator** обладает всеми правами. Изменение текущего уровня доступа может быть изменено любым пользователем и предназначено больше для защиты от неверных действий, чем для разграничения прав.

В дальнейшем изложении мы предполагаем, что все действия выполняются пользователем **Administrator**.

Чтобы получить уровень доступа **Administrator**, на вкладке **Access Level** выберите опцию **Administrator**, подтвердите выбор, установив флажок в поле **"I take full responsibility for my actions"**, и нажмите кнопку **Apply** (как показано на рисунке "Выбор уровня доступа").

Разделы интерфейса

Home (Главная страница)

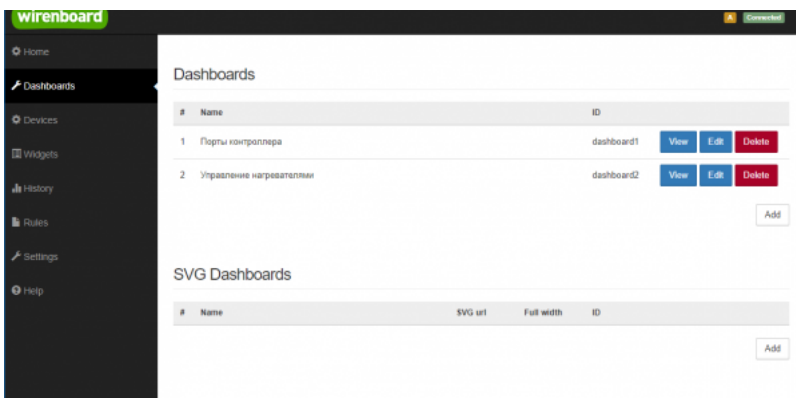


Home - главная страница

Это главная страница пользователя. На неё выводятся элементы интерфейса - так называемые "виджеты" (widget). Это могут быть показания датчиков (например, датчика температуры), кнопки включения света, управления подключёнными реле. Набор виджетов на главной странице полностью настраивается пользователем в меню **Settings -> Web UI -> Common Info**, где можно выбрать панель, которая будет отображаться во вкладке **Home** по умолчанию.

Dashboards (Панели)



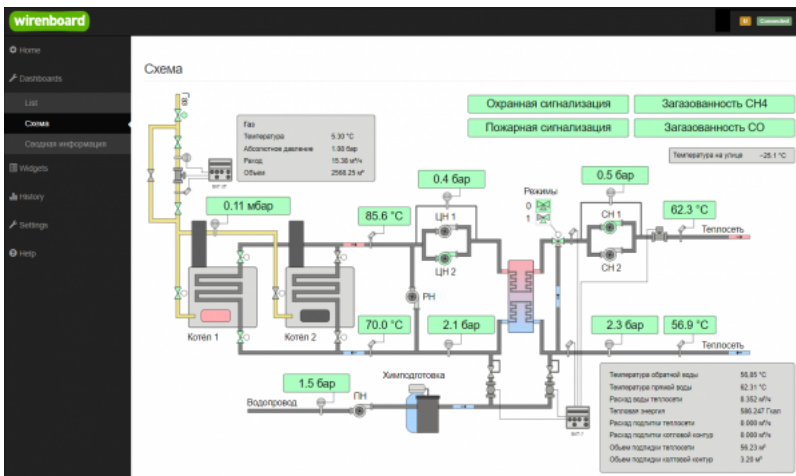


Dashboards - список панелей

Виджеты со схожим назначением можно группировать в панели, где на одном экране находятся все необходимые кнопки, настройки и показатели датчиков. Например, можно объединить виджеты включения подогрева, кондиционера, отображения температуры и влажности. В разделе **Dashboards** можно увидеть все созданные панели. Раздел Home тоже отображает одну из панелей, выбранную в настройках.

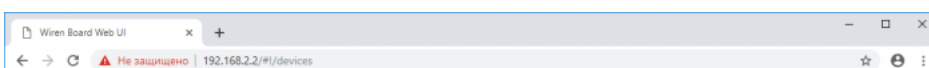
Дважды щелкнув по вкладке **Dashboards** на боковой панели, можно раскрыть список всех панелей, созданных в веб-интерфейсе.

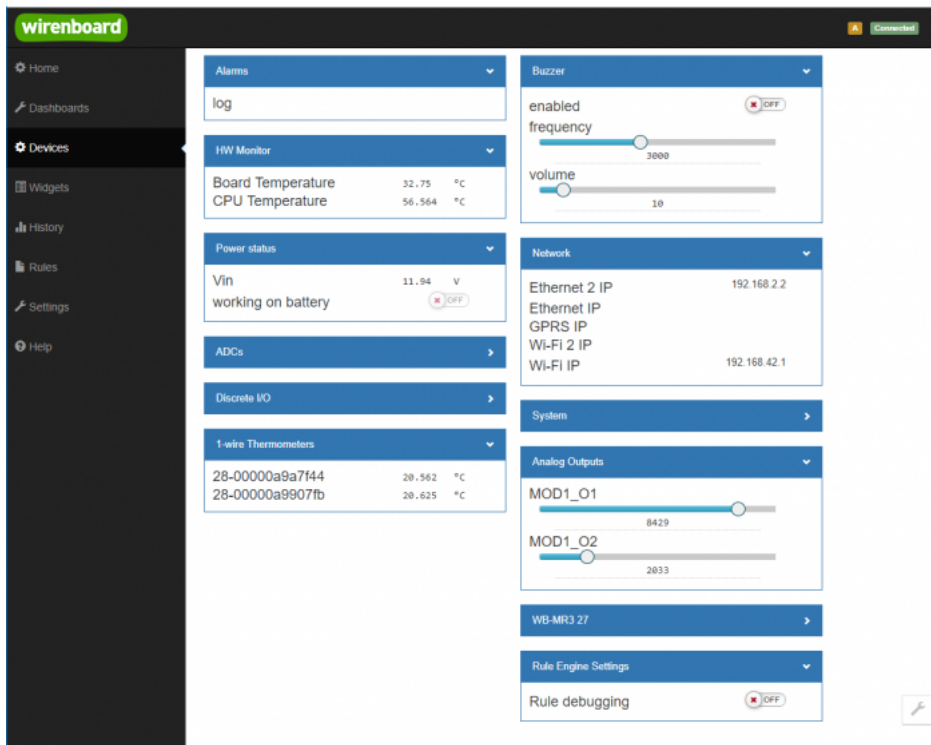
Кроме текстовых панелей с виджетами, можно создавать интерактивные SVG-панели (SVG Dashboards).



Пример SVG-панели

Devices (Устройства)





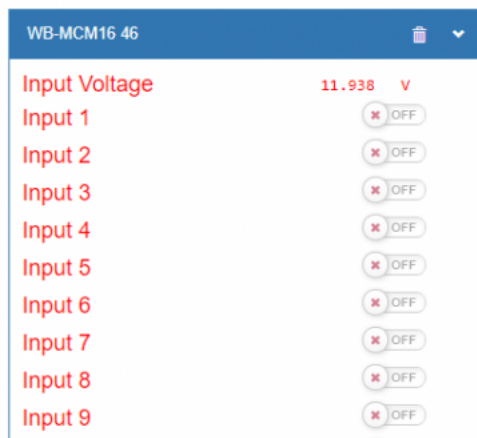
Devices - список всех аппаратных функций контроллера и подключённых устройств

На странице **Devices** отображаются все аппаратные возможности контроллера (состояние входов, выходов, напряжение питания), а также подключённых датчиков и устройств. Если вы подключили к контроллеру внешний модуль, все его меняющиеся значения будут отображены тут.

Каждый элемент устройства (показание значения напряжения, сетевой адрес, кнопка управления реле, флажок состояния входа и т.п.) -- называется "контрол". Несколько контролов могут быть объединены в один виджет. Подробнее смотрите в разделе Widgets (Виджеты).

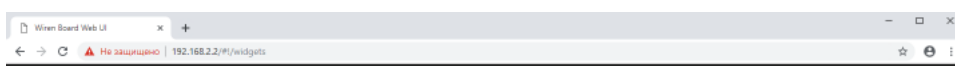
Подключаемые устройства (Modbus-модули, боковые и внутренние модули) **не** определяются контроллером автоматически. Чтобы на этой странице появились аппаратные возможности подключённых устройств (например, внешних модулей реле), сначала нужно настроить их через раздел Configs (Конфигурирование).

Удалить отключенные/неработающие устройства из веб-интерфейса можно с помощью кнопки **Delete** в виде значка с изображением мусорной корзины, в верхней строке плитки устройства. Кнопка появляется, когда указатель мыши находится над плиткой устройства.



Удаление отключенного устройства

Widgets (Виджеты)



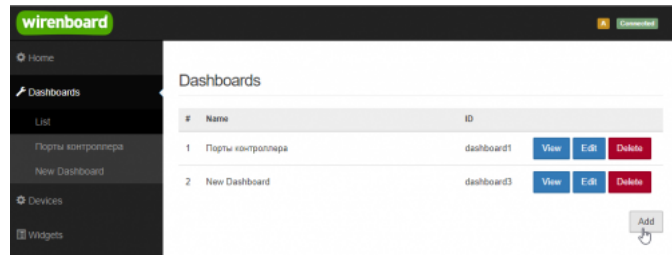
| # | Name | Cells | Types | Values | Graph | Description | Dashboards |
|---|-------------------------|----------------------------------|-----------------------|------------------|-------|-------------------------|---|
| 1 | Напряжение питания | Vin | voltage | 11.00 V | ↓ | Панель контроллера | Панель контроллера Add to dashboard |
| 2 | Звуковой сигнал | enabled | switch | OFF | ↓ | Панель контроллера | Панель контроллера Add to dashboard |
| 3 | IP-адрес Ethernet | Ethernet 2 IP | text | 192.168.2.2 | ↓ | Панель контроллера | Панель контроллера Add to dashboard |
| 4 | Напряжение на клемме A1 | A1 | voltage | 0.02 V | ↓ | Панель контроллера | Панель контроллера Add to dashboard |
| 5 | Температура ЦПУ | CPU Temperature | temperature | 56.564 °C | ↓ | Управление нагревателем | Управление нагревателем Add to dashboard |
| 6 | Комната 1 | Температура воздуха Конвектор | temperature switch | 20.312 °C OFF | ↓ | Управление нагревателем | Управление нагревателем Add to dashboard |
| 7 | Комната 2 | Температура воздуха Конвектор | temperature switch | 20.375 °C OFF | ↓ | Управление нагревателем | Управление нагревателем Add to dashboard |

Widgets - страница управления виджетами

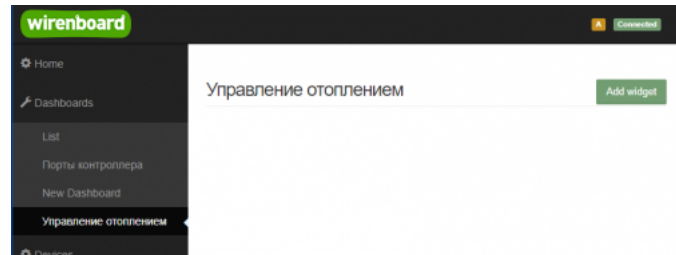
Виджеты - комбинированные элементы интерфейса контроллера, включающие в себя набор контролов, то есть аппаратных параметров контроллера и подключённых к нему устройств - тех, что отображаются на странице [Devices \(Устройства\)](#).

На странице Widgets представлен список всех виджетов, созданных в системе. Сами виджеты создаются в настройках панелей, на этой странице ими можно только управлять: просматривать, удалять и добавлять к существующим панелям-дашбордам.

Пример создания виджетов



Создать новую панель

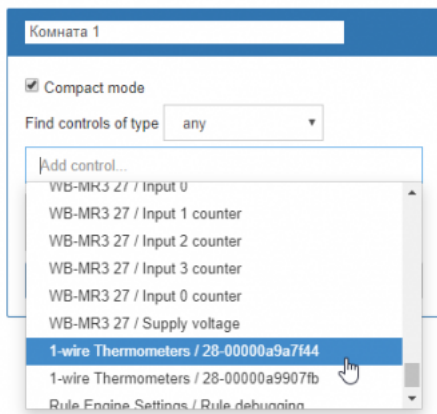


Создать новый виджет

Для примера создадим два виджета с показаниями температуры и переключателями управления отопительными конвекторами для двух комнат "Комната 1" и "Комната 2".

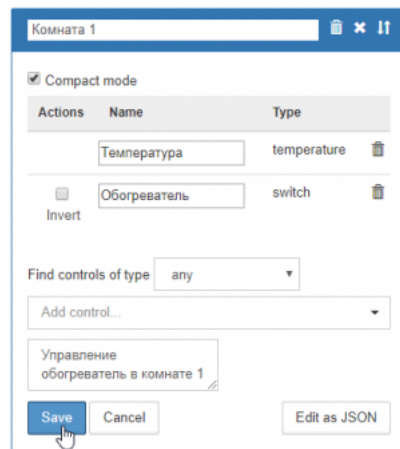
- На боковой панели щелкнем на вкладке **Dashboards**, в раскрывшемся списке выберем элемент **List** и на открывшейся странице нажмем кнопку **Add**.
- В поле **Name** напишем название новой панели, "Управление отоплением" и нажмем кнопку **Save**.
- В списке на странице **Dashboards** щелкнем по кнопке **View** напротив новой панели "Управление отоплением".
- В открывшемся окне с названием панели щелкнем по кнопке **Add widget** в правом верхнем углу окна (см. Рис. "Создать новый виджет").
- В заголовке виджета укажем название, в нашем случае "Комната 1", в списке **Add control...** выберем контрол, соответствующий термометру в первой комнате, еще раз в этом списке выберем реле, которое будет включать нагреватель.
- В поле **Name** виджета можно задать осмысленные названия для контролов, например: "Температура" и "Обогреватель". Снимите флажок **Compact mode**, чтобы эти названия контролов отображались в виджете.
- В поле **Widget description** можно написать назначение виджета.
- Аналогично создадим виджет для управления отоплением в комнате 2.
- Для внесения изменений подведите курсор к заголовку виджета и нажмите кнопку **Edit widget**, внесите изменения и нажмите кнопку **Save**.

управление отоплением



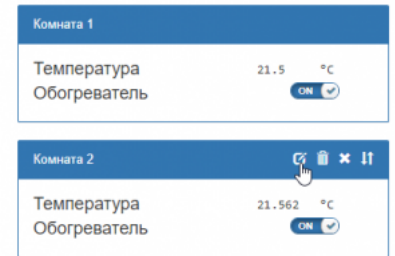
Добавить новый контрол в виджет

Управление отоплением



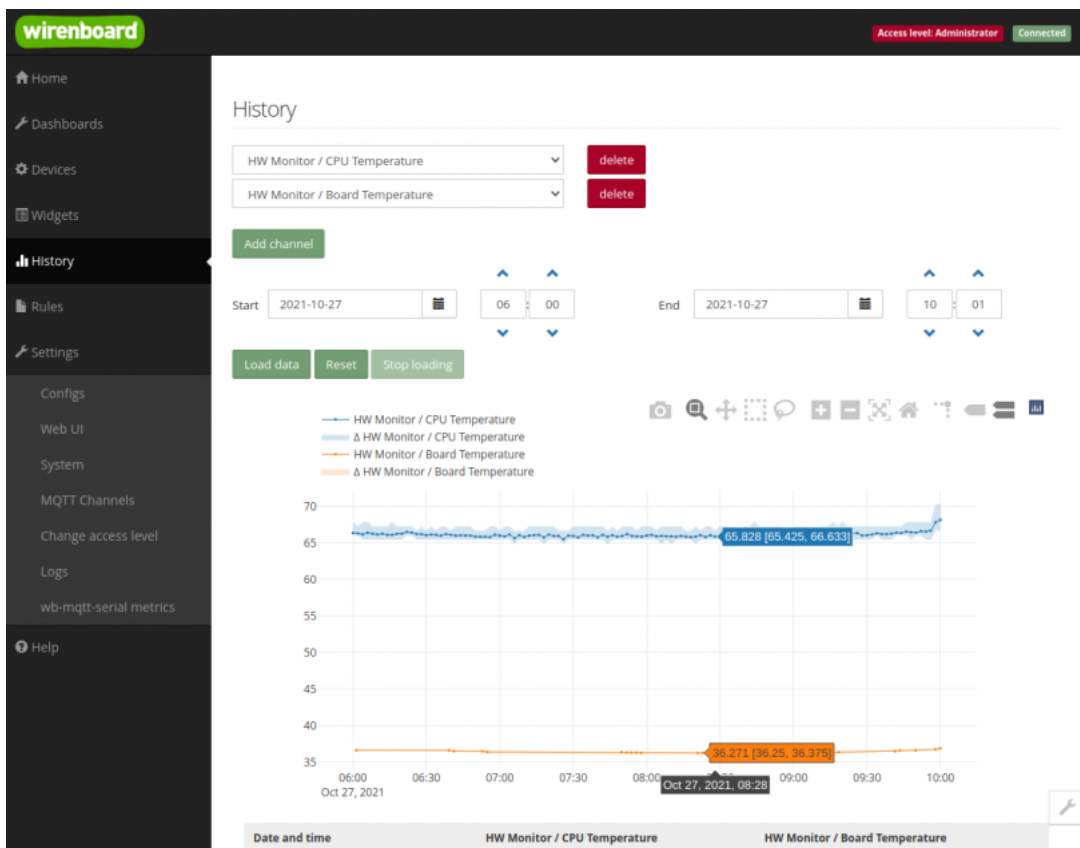
Пример создания виджета

Управление отоплением



Пример готовых виджетов на панели

History (История показаний)



Пример отображения исторических данных

На странице *History* можно просмотреть историю изменения значений аппаратных ресурсов (например, датчиков температуры, напряжения, показаний счётчиков). История представляется одновременно в виде графика и таблицы значений с метками времени.

Возможности просмотра исторических данных:

- Указание интервала времени для отображения данных
- Добавление и удаление нескольких показателей (кнопки Add channel и delete) на график
- Просмотр данных в виде графика и в виде таблицы
- Загрузка данных за выбранный период в csv-формате.

При наведении указателя мыши на область изображения становятся доступными дополнительные функции (кнопки в верхней правой части графика):

- Сохранение графика в формате .png
- Переключение между режимами Zoom (увеличения/уменьшения отрезка данных и масштаба с помощью выделения нужной области указателем мыши) и панорамирования Pan (перемещения области видимости с зажатой левой кнопкой мыши)

- Уменьшение и увеличение отображаемого временного интервала (Zoom in и Zoom out)
- Автоматический выбор масштаба графика по обеим осям
- Возвращение масштаба осей к исходному
- Включение/выключение указателя координат

Утилита для извлечения исторических данных из внутренней базы данных

Rules (Правила-скрипты)

```

1 defineVirtualDevice("spotlight", {
2   title: "Spotlight", //
3   cells: {
4     //
5     on: {
6       type: "switch",
7       value: false,
8       readonly: true
9     }
10  }
11 });
12
13 defineRule("spotlightRelayChange", {
14   whenChanged: "mb-mr11_111/K5",
15   then: function (newValue, devName, cellName) {
16     // if (newValue) { dev["radioRemote"]["K4"]=idev["radioRemote"]["K4"];
17     log(devName);
18     dev["spotlight"]["on"]=newValue;
19   }
20 });
21
22 defineRule("spotlightRadioRemoteChange", {
23   whenChanged: "radioRemote/K1",
24   then: function (newValue, devName, cellName) {
25     // if (newValue) { dev["mb-mr11_111"]["K5"]=idev["mb-mr11_111"]["K5"];
26     log(devName);
27     dev["mb-mr11_111"]["K5"]=newValue;
28   }
29 });

```

Скрипт, открытый для просмотра и редактирования

На странице **Rules** можно создавать и редактировать правила. Правила пишутся на простом языке, похожем на JavaScript и позволяют создавать правила ("включай свет с 10:00 до 18:00") или виртуальные устройства (например, кнопка в интерфейсе, которая включает и отключает всё освещение в здании вместе).

- [Подробнее про скрипты.](#)

Settings -> Configs (Настройки -> Конфигурирование)

| File | Title | Description |
|---------------------------|--|---|
| /etc/network/interfaces | Network Interface Configuration | Specifies network configuration of the system |
| /etc/ntp.conf | NTP configuration | Specifies NTP network time sync daemon configuration |
| /etc/wb-hardware.conf | Hardware Modules Configuration | Lists additional hardware modules configuration |
| /etc/wb-knxnd-config.conf | KNXD Configuration | |
| /etc/wb-mqst-adc.conf | ADC Driver Configuration | Analog inputs configuration |
| /etc/wb-mqst-dac.conf | Analog Outputs Configuration | |
| /etc/wb-mqst-db.conf | MQTT History Configuration | |
| /etc/wb-mqst-gpio.conf | GPIO Driver Configuration Type | Digital inputs and outputs configuration |
| /etc/wb-mqst-mbgate.conf | MQTT to Modbus TCP and RTU slave gateway configuration | |
| /etc/wb-mqst-opcu.conf | MQTT to OPC UA gateway configuration | Configure topics to fields mapping and daemon configuration |
| /etc/wb-mqst-serial.conf | Serial Device Driver Configuration | |
| /etc/wb-rules/alarms.conf | Alarm Configuration | Lists alarms |
| /etc/wb-webui.conf | WebUI Configuration | For internal use, edit with caution |

Страница Configs

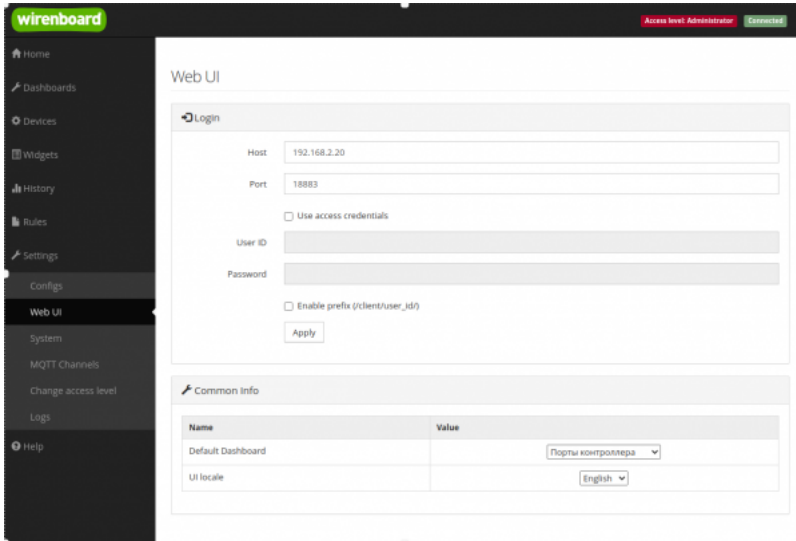
На странице **Settings -> Configs** производится конфигурирование контроллера и настройка подключения внешних устройств:

- настройка сетевых интерфейсов
- настройка серверов получения точного времени
- конфигурирование и настройка боковых и внутренних модулей
- настройка сервиса knxd
- настройка аналоговых входов
- настройка записи в историю
- настройка цифровых входов и выходов (GPIO): в последних версиях контроллера список GPIO по умолчанию пустой, все

вводы-выводы сконфигурированы системой. изменять назначение вводов-выводов следует, если вы хотите изменить их режим функционирования. Список номеров GPIO для последних версий контроллеров Wiren Board 6 представлен на странице [Подробное_тех.описание_платы_контроллера](#).

- настройка шлюза Modbus TCP / Modbus RTU
- настройка шлюза OPC UA
- настройка подключения устройств RS-485
- настройка предупреждений (alarms)
- доступ к редактированию JSON-файла настроек веб-интерфейса

Settings -> WebUI (Настройки -> Веб-интерфейс)

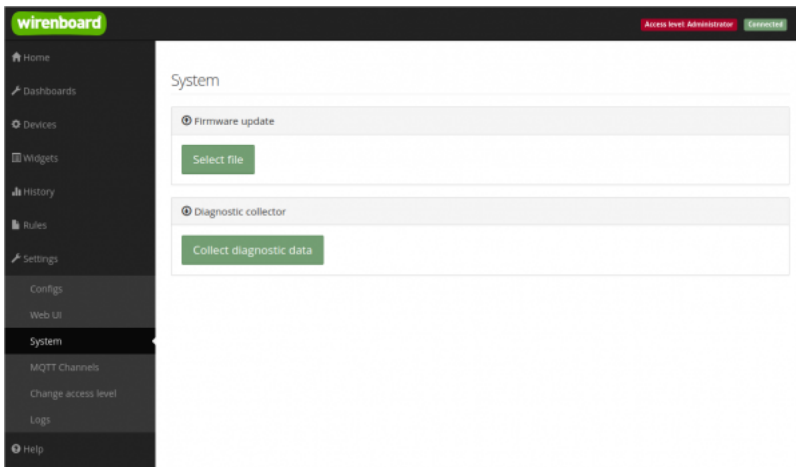


Страница Settings

На странице **Web UI** настраиваются параметры веб-интерфейса и контроллера. Здесь можно:

- Выбрать подключение к MQTT-брокеру (Web-sockets), если используется нелокальный брокер, а, например, облачный сервис
- При необходимости указать учетные данные на удаленном MQTT-брокере
- Указать префикс всех топиков, с которым данные охраняются в облачном сервисе
- Выбрать панель (Default Dashboard), которая будет отображаться на главной странице (Home)
- Выбрать язык веб-интерфейса.

Settings -> System (Настройки -> Системные)

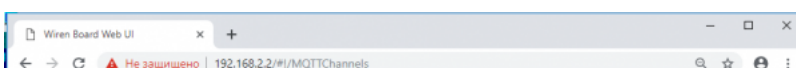


Страница System

На этой странице можно:

- Обновить прошивку контроллера, предварительно скачав ее на компьютер.
- Загрузить архив с диагностической информацией для отправки в техподдержку.

Settings -> MQTT Channels (Настройки -> MQTT-каналы)



| Device | Control | Type | Topic | Value | Status |
|--------------|-------------------|-------------|---|--------------|--------|
| alarms | log | text | /devices/alarms/controls/log | | OK |
| buzzer | enabled | switch | /devices/buzzer/controls/enabled | false | OK |
| buzzer | frequency | range | /devices/buzzer/controls/frequency | 3000 | OK |
| buzzer | volume | range | /devices/buzzer/controls/volume | 10 | OK |
| hwmom | Board Temperature | temperature | /devices/hwmom/controls/Board Temperature | 34.875 | OK |
| hwmom | CPU Temperature | temperature | /devices/hwmom/controls/CPU Temperature | 57.778 | OK |
| network | Ethernet 2 IP | text | /devices/network/controls/Ethernet 2 IP | 192.168.2.2 | OK |
| network | Ethernet IP | text | /devices/network/controls/Ethernet IP | | OK |
| network | GPRS IP | text | /devices/network/controls/GPRS IP | | OK |
| network | Wi-Fi 2 IP | text | /devices/network/controls/Wi-Fi 2 IP | | OK |
| network | Wi-Fi IP | text | /devices/network/controls/Wi-Fi IP | 192.168.42.1 | OK |
| power_status | Vin | voltage | /devices/power_status/controls/Vin | 11.91 | OK |

MQTT Channels

Settings → **MQTT Channels**. На этой странице приводится справочная информация о всех MQTT-топиках, полученных веб-интерфейсом контроллера, а также статус их получения (**OK** или **ERR** в последнем столбце).

Settings -> Change access level (Настройки -> Права доступа)

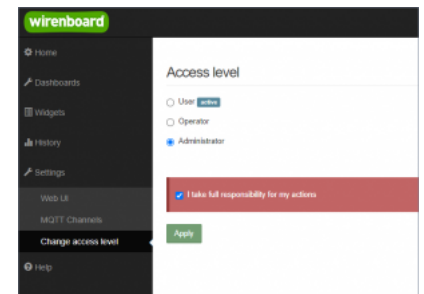
Уровни доступа призваны защитить пользователя от ошибок при регулярной работе с контроллером. Важно понимать, что это не полноценное разграничение прав, а способ защитить себя от необдуманных действий. Новых пользователей создавать нельзя.

Доступны следующие уровни:

- **User** — дашборды, виджеты, история, базовые настройки.
- **Operator** — права уровня User и раздел Devices.
- **Administrator** — полный доступ ко всем функциям.

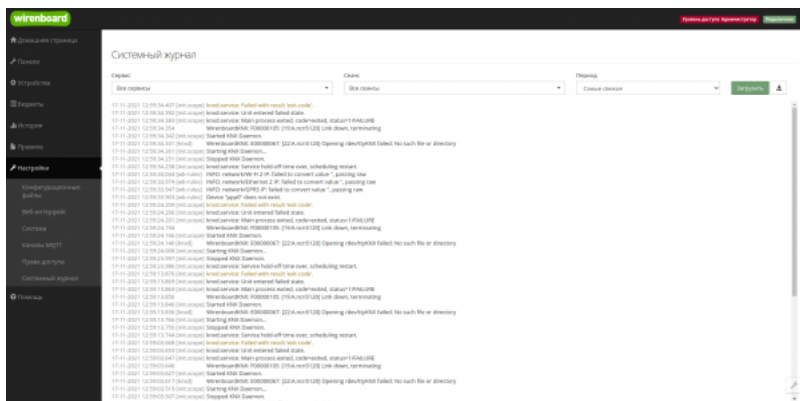
Чтобы изменить уровень доступа к настройкам веб-интерфейса:

- Зайдите в веб-интерфейс.
- Перейдите в раздел **Settings** и выберите пункт **Change access level**. Установите нужное значение и нажмите кнопку **Apply**.



Web UI 2.0 — смена уровня доступа текущего пользователя

Settings -> Logs (Настройки -> Логи)



Просмотр лог-файлов контроллера

Чтобы открыть инструмент, перейдите **Settings** → **Logs**.

После выбора параметров, нажмите **Load** для запроса данных из системного журнала.

Загруженные сообщения можно сохранить в файл, для этого нажмите кнопку **Save loaded log to file** и укажите куда сохранить. Строки из системного журнала подгружаются во время прокрутки списка сообщений, поэтому сперва прокрутите список до нужного места, а потом сохраняйте вывод в файл.

Для отправки сообщений системного журнала в техподдержку удобно использовать выгрузку диагностической информации.

Стандартные задачи, решаемые через веб-интерфейс

Подключить устройство RS-485 Modbus и создать кнопки управления на главной панели

RS-485. Примеры через веб-интерфейс

Обновить прошивку контроллера

Обновление прошивки через веб-интерфейс

Облачный интерфейс

Веб-интерфейс Wiren Board можно разместить не только на самом контроллере, но и на специальном сервере. Тогда на интерфейс можно будет заходить, используя всегда один и тот же IP-адрес.

Чтобы контроллер начал работать с веб-интерфейсом, размещённым на сервере, нужно внести некоторые изменения в конфигурацию контроллера.

Такой вариант удобен, если ваш контроллер находится за роутером и не имеет глобального IP-адреса, или если он подключён по GPRS - тогда он тоже, скорее всего, не имеет глобального IP, да ещё и работа с удалённым веб-интерфейсом израсходует слишком много трафика.

Пока что такой вариант доступен только корпоративным клиентам по запросу.

Настройка авторизованного доступа к веб-интерфейсу контроллера

В статье Защита паролем приводятся краткие инструкции по перенастройке контроллера, обеспечивающие авторизованный доступ к веб-интерфейсу контроллера.

Основные отличия версии 2.x от 1.0

- Каждый виджет может содержать произвольное число каналов, в виджете каналы можно переименовывать
- Отдельные устройства теперь автоматически сворачиваются в виде плиток, если не помещаются на экране. Плитки можно развернуть или свернуть
- Появились уровни доступа к интерфейсу (пользователь, оператор, администратор). Текущий уровень доступа отображается в правом верхнем углу интерфейса, рядом со значком состояния подключения
- Улучшенный интерфейс для мобильных устройств
- По клику на канал или значение название канала или его значение копируются в буфер обмена
- Историю значений можно посмотреть, нажав на кнопку, появляющуюся рядом со значением при наведении
- Историю значений можно скачивать в виде текстового файла
- Исторические данные загружаются постепенно; возможно сравнивать значения нескольких каналов
- Удаление лишних MQTT-топиков из интерфейса
- Все настройки отображения теперь хранятся в конфиг-файле /etc/wb-ui.conf в формате JSON. Теперь их можно редактировать и генерировать из сторонних программ и очень просто копировать с одного контроллера на другой
- Отсутствуют "Комнаты"
- Сохранение конфигурации интерфейса при обновлении предыдущей версии веб-интерфейса.

Модуль уведомлений (алармов, Alarm) предназначен для быстрой настройки отправки СМС и email в случае изменения параметров контроллера или подключённых устройств: показаний подключённого датчика температуры, сработки пожарной сигнализации, обнаружении протечки и т.п.

Contents

[Предварительная настройка для отправки email](#)

[Для ящика на сервере Google](#)

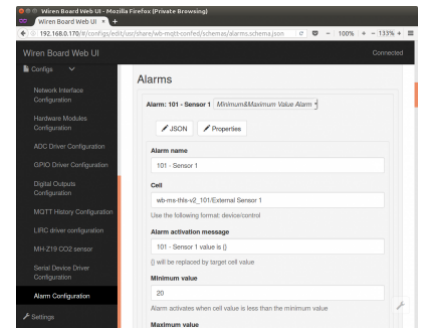
[Для ящика на сервере mail.ru](#)

[Проверка настроек](#)

[Предварительная настройка для отправки SMS](#)

[Настройка модуля уведомлений](#)

[Как настроить уведомления с более сложными условиями](#)



Настройка модуля уведомлений

Предварительная настройка для отправки email

На уровне Linux для отправки email используется утилита sSMTP (<https://linux.die.net/man/8/ssmtp>). Чтобы начать отправлять почту, нужно отредактировать её конфигурационный файл `/etc/ssmtp/ssmtp.conf`.

Электронные письма будут отправляться с одного из ваших почтовых ящиков, от которого потребуются:

- логин — например, `robot@contactless.ru`,
- пароль,
- адрес и порт SMTP сервера. Для ящиков на Gmail это `smtp.gmail.com:587`, для Mail.Ru — `smtp.mail.ru:465`).

Для ящика на сервере Google

Очистите содержимое файла `/etc/ssmtp/ssmtp.conf` и добавьте туда строки:

```
mailhub=smtp.gmail.com:587
UseTLS=Yes
UseSTARTTLS=Yes
AuthUser=robot@contactless.ru
AuthPass=qwerty12345
# Не забудьте добавить пустую пустую строку в конце файла
```

Обязательно оставьте пустую строку в конце файла — после строки с `AuthPass=`.

Для отправки писем через Google Mail необходимо разрешить отправку писем от недоверенных приложений для вашего Google-аккаунта

Для ящика на сервере mail.ru

Очистите содержимое файла `/etc/ssmtp/ssmtp.conf` и добавьте туда строки:

```
mailhub=smtp.mail.ru:465
FromLineOverride=YES
UseTLS=Yes
AuthUser=robot@contactless.ru
AuthPass=qwerty12345
# Не забудьте добавить пустую пустую строку в конце файла
```

В файл `/etc/ssmtp/revaliases` добавьте строку:

```
root:robot@contactless.ru:smtp.mail.ru:465
```

Проверка настроек

Чтобы проверить, что отправка email заработала, выполните в консоли команду:

```
echo 'Test message to check sSMTP new configuration' | ssmtp -v ivanov@gmail.com
# вместо ivanov@gmail.com подставьте адрес другого своего ящика (например того, куда собираетесь получать уведомления с контроллера)
```

Дополнительную информацию можно получить на сайтах askubuntu.com (<http://askubuntu.com/questions/185070/why-i-get-hostname-or-service-not-known-error>) и [wiki.archlinux.org](https://wiki.archlinux.org/index.php/SSMTP) (<https://wiki.archlinux.org/index.php/SSMTP>).

Предварительная настройка для отправки SMS

Убедитесь, что вы настроили модем вставили SIM-карту и подключили GSM антенну.

Перед настройкой в веб-интерфейсе можете протестировать отправку SMS из консоли.

Настройка модуля уведомлений

1. Зайдите на страницу **Settings** → **Configs** → **Alarm Configuraion** веб-интерфейса.

2. Добавьте получателей уведомлений в разделе **Recipients**:

- нажмите кнопку **+ Recipient**;
- В выпадающем списке *Recipient N* выберите один из типов: *E-mail recipient* или *SMS recipient*;
- введите почтовый адрес или номер телефона. Для email можно также указать тему письма.

3. Настройте проверяемый параметр и условие в разделе **Alarms**:

- нажмите кнопку **+ Item**;
- в появившемся разделе в выпадающем списке выберите один из четырёх типов уведомлений:
 1. **Expected Value Alarm** — уведомление, если значение перестанет равняться заданному.
 2. **Minimum Value Alarm** — уведомление, если значение станет меньше заданного.
 3. **Maximum Value Alarm** — уведомление, если значение станет больше заданного.
 4. **Minimum&Maximum Value Alarm** — уведомление, если значение выйдет из заданного диапазона.
- в **Alarm name** введите произвольное название для уведомления;
- в **Cell** введите параметр, за которым нужно следить. Параметр задаётся в виде Device/Control, где названия *Device* и *Control* для каждого параметра можно найти в веб-интерфейсе в разделе **Settings** → **MQTT Channels**. Например, для датчика 1-Wire, подключённого к внешнему modbus-сенсору WB-MS, эта строка будет выглядеть как `wb-ms-th1s-v2_101/External Sensor 2`;
- в **Alarm activation message** введите текст сообщения, которое будет отправлено. В него в произвольном месте можно ввести две фигурные скобки {}, вместо которых в сообщение будет подставлено новое значение отслеживаемого параметра.
- в зависимости от типа уведомления, заполните поля **Minimum Value**, **Maximum Value**, **Expected Value**;
- некоторые дополнительные настройки станут доступны, если нажать на кнопку **Properties** и отметить их.

4. Нажмите **Save** в самом верху страницы:

- если вы забыли указать какой-то обязательный параметр, кнопка *Save* будет неактивна, а рядом с параметром красным будет написана ошибка, которую нужно исправить;
- если всё в порядке, уведомления сразу начнут работать.

Как настроить уведомления с более сложными условиями

Модуль уведомлений предназначен для быстрой настройки простых уведомлений. Если вам нужны более сложные сценарии уведомлений, их можно запрограммировать через движок правил. Чтобы отправить сообщение, из правила нужно вызвать функцию

```
Notify.sendEmail("to", "subject", "text"); //отправляет почту указанному адресату (to), с указанной темой (subject) и содержимым (text); почтовый адрес, тема и содержимое указываются в кавычках
```

или

```
Notify.sendSMS("to", "text"); //отправляет SMS на указанный номер (to) с указанным содержимым (text); номер и текст указываются в кавычках
```


Смотрите также:

- Репозиторий с устаревшим нативным мостом в Zabbix (<https://github.com/contactless/wb-mqtt-zabbix>). Пакеты в разделе Releases (<https://github.com/wirenboard/wb-mqtt-zabbix/releases>).
- Статья на Хабре «Zabbix + Wirenboard: мониторинг производства (<https://habr.com/ru/post/525852/>)» — подробное описание создания системы с Wiren Board и Zabbix. В статье используется плагин, написанный пользователем — [zbx_mqtt](https://github.com/v-zhuravlev/zbx_mqtt) (https://github.com/v-zhuravlev/zbx_mqtt)

Подготовка

Нужно поставить zabbix-agent:

```
apt-get install zabbix-agent -y
```

Настройка Zabbix Agent

Для получения значений переменных в Zabbix используется чтение retained-значений каналов.

Для настройки zabbix-agent нужно создать файл zabbix-mqtt.conf в директории /etc/zabbix/zabbix_agentd.conf.d/ и записать в него строку:

```
UserParameter=mqtt.value[*],mosquitto_sub -t '$1' -C 1
```

Это можно сделать одной командой:

```
echo "UserParameter=mqtt.value[*],mosquitto_sub -t '\$1' -C 1" > /etc/zabbix/zabbix_agentd.conf.d/zabbix-mqtt.conf
```

Использование

Обращение к каналам в Zabbix происходит следующим образом:

```
mqtt.value[topic]
```

где topic - топик соответствующего канала в MQTT

например:

```
mqtt.value[/devices/sht1x/controls/humidity]
mqtt.value[/devices/sht1x/controls/temperature]
```

Пример теста агента, запускать на контроллере:

```
zabbix_agentd -t mqtt.value[/devices/wb-adc/controls/5Vout]
```

Шлюз Modbus RTU/TCP

Contents

Описание

Modbus TCP binding

Modbus RTU binding

MQTT connection

Общие для Modbus RTU/TCP параметры

Register bindings

Discrete inputs and Coils

Input- и Holding-registers

Описание

Контроллеры Wigen Board могут выступать в роли Slave-устройства и транслировать сообщения из выбранных MQTT-топиков по протоколам Modbus RTU, Modbus TCP.

Функция трансляции топиков может быть полезна для настройки взаимодействия между контроллером Wigen Board и внешним программным обеспечением, например, SCADA-системы с поддержкой Modbus RTU/TCP.

Для создания шлюза Modbus RTU/TCP служит сервис `wb-mqtt-mbgate` (<https://github.com/wireboard/wb-mqtt-mbgate>). Перед использованием нужно указать топик, отображаемые в регистры Modbus RTU/TCP, уточнить их адреса и выбрать формат регистров.


Настройки шлюза хранятся в файле `/etc/wb-mqtt-mbgate.conf` и могут быть изменены через веб-интерфейс. Чтобы открыть настройку шлюза, перейдите в веб-интерфейсе в раздел **Settings** → **Configs** → **MQTT to Modbus TCP and RTU slave gateway configuration**.

Обратите внимание, что порт, выбранный для работы в режиме Modbus-slave, не должен использоваться в настройках опроса драйвера `wb-mqtt-serial`!

Modbus TCP binding

Выберите здесь тип шлюза — **TCP**, а также укажите IP-адрес и порт создаваемого сервера:

- Bind address — IP-адрес по которому будет доступен сервер Modbus TCP. По умолчанию: * — возможно подключение к любому IP-адресу контроллера.
- Server TCP port — порт, на котором сервер будет ждать соединения. По умолчанию: 502 — стандартный для Modbus TCP порт.

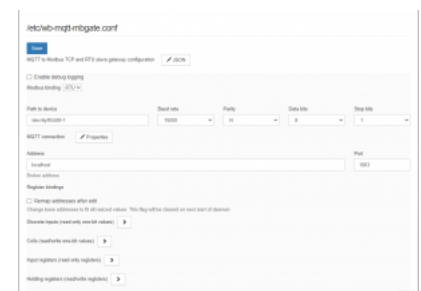


Настройка Modbus TCP шлюза в веб-интерфейсе контроллера Wigen Board

Modbus RTU binding

Выберите здесь тип шлюза — **RTU**, а также укажите serial-порт и параметры соединения:

- Path to device — адрес serial-порта контроллера, на котором будет доступен шлюз.
- Baud rate — скорость соединения.
- Parity — контроль четности.
- Data bits — биты данных.
- Stop bits — стоп-биты.



Настройка Modbus RTU шлюза в веб-интерфейсе контроллера Wigen Board

MQTT connection

Здесь можно настроить параметры для подключения к брокеру сообщений MQTT. По умолчанию шлюз использует MQTT-брокер контроллера. Также вы можете настроить аутентификацию при подключении к брокеру.

Обязательные параметры:

- Address — адрес брокера сообщений. По умолчанию: localhost — подключение к брокеру контроллера.
- Port — номер порта, по которому доступен брокер. По умолчанию: 1883.

Нажав на кнопку **Properties** вы можете включить дополнительные параметры:

- Keep-alive interval — интервал ожидания ответа от брокера. Если сообщение не пришло — запрос повторяется.
- Enable username+password authentication — если брокер требует аутентификации с помощью логина и пароля, установите этот флаг. Также потребуется указать данные учетной записи:
 - Login — имя пользователя.
 - Password — пароль.

Общие для Modbus RTU/TCP параметры

Register bindings

В этом разделе можно настроить соответствия топиков и регистров.

Раздел содержит список всех найденных каналов в очереди сообщений MQTT, распределённых по типу. Каналы определяются по имени топика: в список вносятся все топика, соответствующие шаблону `/devices+/controls/+`, при этом в списке они отображаются как `+/+`. Например, если в брокере сообщений обнаружено сообщение с топиком `/devices/alarms/controls/alarm_test`, то в этом списке канал будет записан как `alarms/alarm_test`.

Если в брокере сообщений появились новые каналы, например, при подключении нового устройства — они будут добавлены в список при перезапуске службы. Чтобы перезапустить службу, сохраните конфигурацию или перезапустите контроллер.

Каналы распределены по четырем группам — типам регистров в Modbus:

- Discrete inputs — дискретные входы. Двоичные значения 1/0, доступные только на чтение. Например, состояния входов GPIO.
- Coils — регистры флагов. Двоичные значения, доступные на чтение и на запись: переключатели состояний реле и т.п.
- Input registers — регистры ввода. Численные значения, доступные только на чтение: значение с датчика температуры, давления и т.п.
- Holding registers — регистры хранения. Численные значения, доступные на чтение и на запись: яркость освещения, уставки температуры и т.п.

Распределение происходит автоматически на основании описания канала в топиках: `/devices+/controls+/meta/+`.

Remap addresses after edit — если произошло наложение адресов, то обновить адреса устройств. Если установить флаг и сохранить настройки, то файл будет проанализирован на наложение адресов. Если наложения будут обнаружены, сервис попытается устранить их и после операции снимет флаг. Обновите страницу клавишами `Ctrl+Shift+R`, чтобы считать внесенные сервисом изменения.

ВНИМАНИЕ: используйте флаг **Remap addresses after edit** только на настраиваемой системе — адреса любых каналов могут измениться, в том числе и настроенных ранее.

Discrete inputs and Coils

Для дискретных и числовых регистров доступны параметры:

- Enabled — флаг, включающий привязку канала к регистру. По умолчанию все каналы отключены.
- MQTT Device — имя канала в формате `+/+`, как соответствие `/devices+/controls/+/`.
- Meta-type — тип канала, полученный из очереди MQTT. Параметр отображен только «для пользователя», не влияет на работу службы.
- Modbus unit ID и Start address — адрес slave-устройства в сети Modbus RTU/TCP и адрес его регистра. Определяются хешированием от имени канала. Таким образом, одинаковые каналы на разных контроллерах с большой вероятностью будут иметь одинаковые адресные поля. Адреса 1 и 2 зарезервированы за пользователем, поэтому ни один канал не будет назначен на них автоматически.

Важно для Modbus TCP: Так как контроллер является шлюзом, то он назначает разным устройствам, которые к нему подключены разные Unit ID. Некоторое программное обеспечение рассчитано на работу с одним Unit ID (например 1) на одном IP-адресе. В некоторых программах стартовый адрес считается начинающимся с 1, а не с 0. В таком случае к стартовому адресу из конфигурации шлюза надо добавлять 1 при указании его в вашем ПО.

Input- и Holding-registers

Для Input- и Holding-регистров, доступны все параметры дискретных и числовых, а также дополнительно:

- Register format — тип хранимых данных:
 - *Signed integer* — знаковое целое;
 - *Unsigned integer* — беззнаковое целое;
 - *BCD* — двоично-десятичное беззнаковое целое. Например, десятичное значение 123 будет кодироваться как `0x123 == 303`.

291;

- *IEEE 754 float* — число с плавающей точкой;
 - *Varchar* — текстовое поле фиксированного размера (1 символ на регистр).
- **Size (in bytes)** — размер данных в байтах. Нужно учитывать, что для всех типов, кроме текстового, в один регистр укладывается два байта:
- *для целочисленных типов* — 2, 4 и 8 байт — это будет 1, 2 и 4 регистра соответственно;
 - *для чисел с плавающей точкой* — 4 и 8 байт, то есть 2 и 4 регистра соответственно;
 - *для текстовых полей* — любое неотрицательное значение.
- **Maximum value** — максимальное значение поля. Требуется для каналов типа *range*. Все значения больше максимума будут заменены на максимальное.
- **Value scale** — множитель значения. Например, если множитель равен 2 и в канале появилось значение 10, значение регистра будет 20. И наоборот, если в регистр было записано значение 30, в брокер уйдёт сообщение со значением 15.
- **Byte swap** — поменять порядок байт в modbus-регистрах. По умолчанию значения записываются в регистры в формате *big-endian*. Если установить этот флаг, регистр будет интерпретироваться как *little-endian*. *Важно:* флаг действует только на отдельные регистры; в случае составных значений (размером больше 2 байт), меняется порядок только в каждом двухбайтном слове. Например, значение 0x12345678 будет записываться как 0x34127856,
- **Word swap** — поменять порядок слов для составных значений. Например, значение 0x12345678 будет записываться как 0x56781234.

Таким образом, чтобы записать значение в формате *little-endian*, нужно установить оба последних флага: *Byte swap* и *Word swap*.

Contents

Описание

Принцип работы

Настройка в веб-интерфейсе контроллера Wiren Board

Пример настройки получения данных в MasterScada 3.x

Полезные ссылки

Описание

Шлюз работает на контроллерах [Wiren Board 6](#) и [Wiren Board 7](#), предыдущие версии не поддерживаются.

Контроллеры Wiren Board могут транслировать сообщения из выбранных MQTT-топиков по протоколу OPC UA.

Функцию можно использовать для экспорта данных во внешнее программное обеспечение например, SCADA-системы с поддержкой OPC UA.

Трансляция организовывается с помощью сервиса [wb-mqtt-opcua](https://github.com/wirenboard/wb-mqtt-opcua) (<https://github.com/wirenboard/wb-mqtt-opcua>).

Сервис уже установлен на контроллере и доступен [стабильном релизе](#).

Принцип работы

При запуске шлюза происходит автоматическое создание конфигурационного файла `/etc/wb-mqtt-opcua.conf`. При последующих запусках шлюз анализирует доступные MQTT каналы (контроли) и добавляет их в файл. Активировать передачу данных конкретных каналов можно, редактируя `/etc/wb-mqtt-opcua.conf`, либо воспользовавшись онлайн-редактором настроек.

Шлюз подключается к заданному MQTT брокеру и подписывается на сообщения от каналов, указанных в конфигурационном файле. В системах с поддержкой протокола OPC UA выступает в роли сервера и принимает входящие TCP/IP соединения по указанному в конфигурационном файле локальному интерфейсу и порту.

Для контролов, доступных для записи (подтопик `/meta/readonly`равный 0), шлюз производит передачу значений, записанных в OPC UA узлы, в соответствующие оп-топики.

Настройка в веб-интерфейсе контроллера Wiren Board

Шлюз можно настроить в веб-интерфейсе, для этого перейдите в раздел **Settings** → **Configs** → **MQTT to OPC UA gateway configuration**.

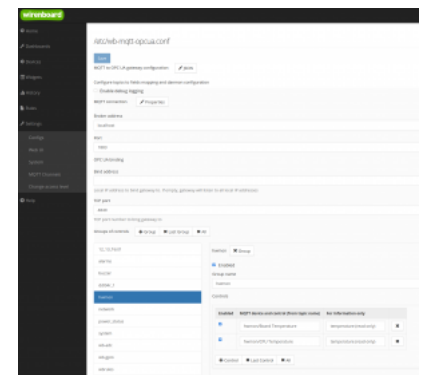
В конфигураторе можно указать:

- параметры подключения к MQTT-брокеру;
- дополнительные параметры, которые можно выбрать, нажав на кнопку **Properties**;
- локальный IP и порт, по которым шлюз будет ожидать подключения.

По умолчанию шлюз создаёт отдельную группу для каждого устройства. Список групп расположен слева, его можно самостоятельно редактировать.

При выборе конкретной группы в правой части появится список входящих в неё каналов. Каналы так же можно создавать, удалять и редактировать.

Столбец **MQTT device and control** указывает конкретный канал MQTT, который формируется из названия устройства и канала: для первого в списке канала соответствующий топик MQTT будет `/devices/hwmon/controls/Board Temperature`.

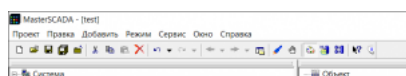


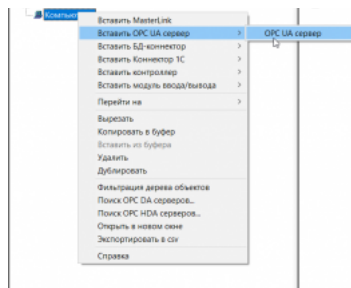
Настройка OPC UA

Пример настройки получения данных в MasterScada 3.x

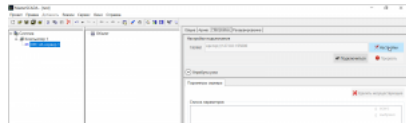
Рассмотрим процесс настройки получения данных по протоколу OPC UA на примере работы с MasterScada 3.x:

1. Создайте новый проект, в нём вставьте компьютер, далее **Вставить OPC UA сервер**

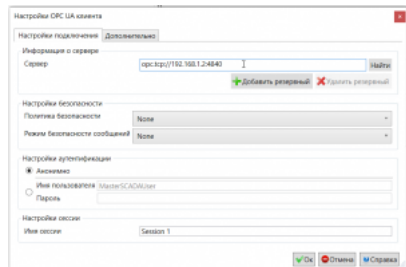




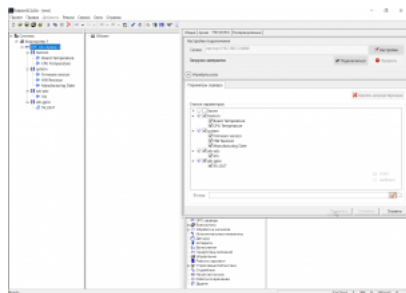
2. В настройках измените **Настройки подключения**



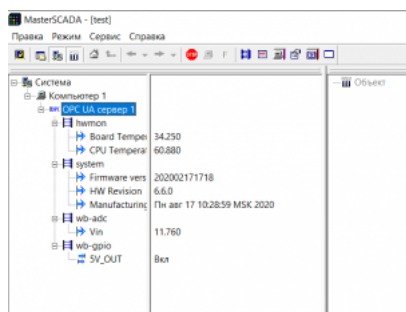
3. Укажите адрес и порт OPC-сервера контроллера.



4. После нажатия кнопки **Подключиться**, будут загружены доступные узлы. Выберите те, с которыми будете работать, и нажмите **Применить**. Они появятся в дереве слева.



5. Проверить получаемые данные можно, запустив проект.



Полезные ссылки

- [Тема на портале поддержки \(https://support.wirenboard.com/t/podderzhka-opc-ua/6927\)](https://support.wirenboard.com/t/podderzhka-opc-ua/6927)
- [Репозиторий wb-mqtt-opcua на Github \(https://github.com/wirenboard/wb-mqtt-opcua\)](https://github.com/wirenboard/wb-mqtt-opcua)

Contents

Описание

Принцип работы

Передача сообщений из MQTT в МЭК 60870-5-104

Передача команд МЭК 60870-5-104 в MQTT

Установка

Настройка в веб-интерфейсе контроллера Wiren Board

Полезные ссылки

Описание

МЭК 104, 104-ый — протокол по ГОСТ Р МЭК 60870-5-104

Разработан шлюз для трансляции сообщений между MQTT-брокером и системами с поддержкой протокола МЭК 60870-5-104. Работает с контроллером Wiren Board 6.

Принцип работы

Передача сообщений из MQTT в МЭК 60870-5-104

Сообщения MQTT передаются в МЭК 60870-5-104 блоками данных (ASDU) с причиной передачи “спорадически”(3). При подключении нового контролирующего устройства, шлюз автоматически высылает последние известные значения всех включенных каналов. В дальнейшем каждое новое MQTT-сообщение сразу же передаётся в МЭК 60870-5-104.

Передача команд МЭК 60870-5-104 в MQTT

Шлюз поддерживает ASDU с типами:

- одноэлементная команда (C_SC_NA_1);
- команда уставки, масштабированное значение (C_SE_NB_1);
- команда уставки, короткое число с плавающей запятой (C_SE_NC_1).

Обрабатывается первый объект информации в ASDU. Если в конфигурационном файле есть включенный канал для адреса этого объекта информации, шлюз произведёт запись полученного значения в соответствующую тему канала, например /devices/wb-gpio/controls/5V_OUT/on). Также поддерживается команда общего опроса станции (C_IC_NA_1, QOI равный 20), прочие команды не поддерживаются.

Установка

Обновляем список пакетов командой:

```
apt update
```

Устанавливаем командой:

```
apt install wb-mqtt-iec104
```

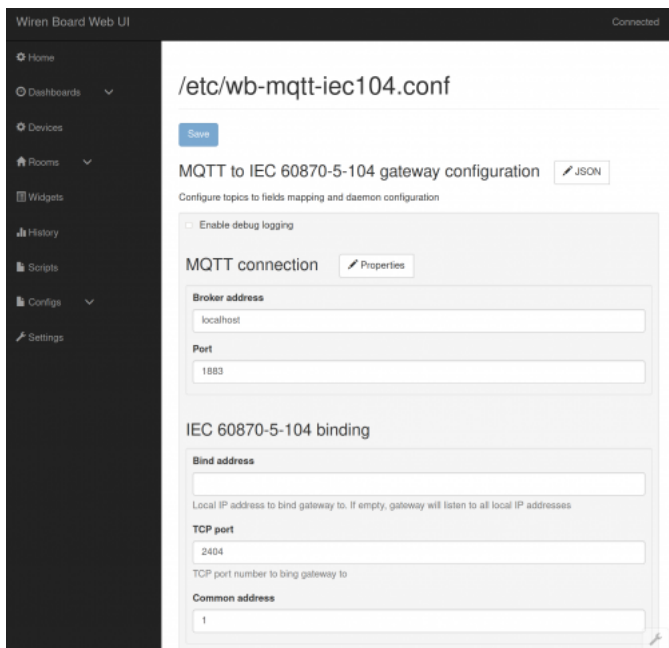
При запуске шлюза происходит автоматическое создание конфигурационного файла /etc/wb-mqtt-iec104.conf. При последующих запусках шлюз анализирует доступные MQTT каналы(контролы) и добавляет их в файл. Активировать передачу данных конкретных каналов можно, редактируя файл /etc/wb-mqtt-serial.conf, либо воспользовавшись онлайн-редактором настроек. Структура конфигурационного файла описана [тут](https://wirenboard.com) (<https://wirenboard.com>).

Шлюз подключается к заданому MQTT брокеру и подписывается на сообщения от каналов, указанных в конфигурационном файле. В системах с поддержкой протокола МЭК 60870-5-104 шлюз выступает в роли контролируемой станции и принимает входящие TCP/IP соединения по указаному в конфигурационном файле локальному интерфейсу и порту.

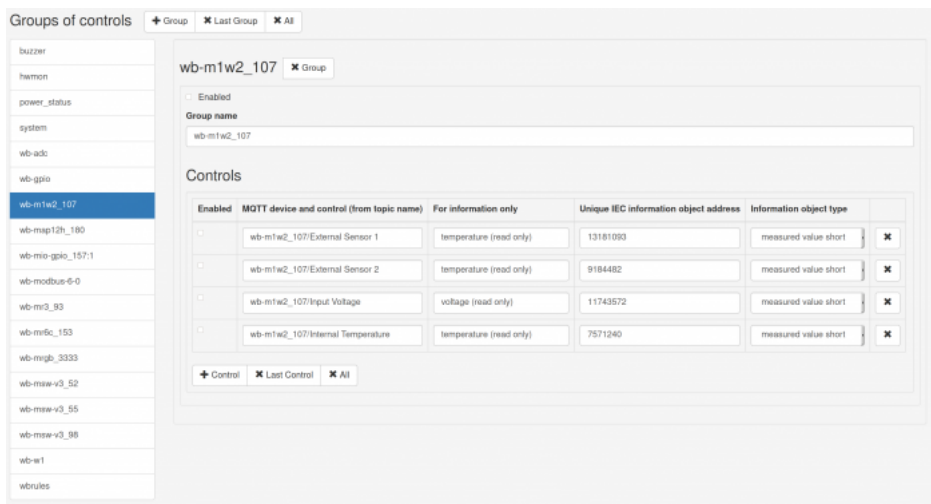
Настройка в веб-интерфейсе контроллера Wiren Board

После установки шлюза его можно настроить в интерфейсе онлайн-конфигуратора (https://wirenboard.com/wiki/Wiren_Board_Web_Interface#Settings_-_3E_Configs_.28.D0.9D.D0.B0.D1.81.D1.82.D1.80.D0.BE.D0.B9.D0.BA.D0.B8_-_3E_.D0.9A.D0.BE.D0.BD.D1.84.D0.B8.D0.B3.D1.83.D1.80.D0.B8.D1.80.D0.BE.D0.B2.D0.B0.D0.BD.D0.B8.D0.B5.29), выбрав из списка файл /etc/wb-mqtt-iec104.conf.

Онлайн-конфигуратор позволяет указать параметры подключения к MQTT-брокеру (дополнительные параметры можно выбрать, нажав на кнопку “Properties”) и выбрать локальный IP и порт, по которым шлюз будет ожидать подключения.



Ниже показан интерфейс редактирования списка групп и каналов для трансляции из MQTT в МЭК 60870-5-104. По умолчанию шлюз создаёт отдельную группу для каждого устройства. Список групп расположен слева, его можно самостоятельно редактировать.



При выборе конкретной группы в правой части появится список входящих в неё каналов. Каналы так же можно создавать, удалять и редактировать. Столбец **MQTT device and control** указывает конкретный канал MQTT. Он формируется из названия устройства и канала. Для первого в списке канала соответствующий топик MQTT будет `/devices/wb-m1w2_107/controls/External Sensor 1`. Уникальный адрес объекта информации согласно МЭК генерируется при старте сервиса, его можно изменить в столбце **Unique IEC information object address**. Также можно поменять тип объекта информации.

Полезные ссылки

- [Тема на форуме \(https://support.wirenboard.com/t/podderzhka-mek-60870-5-104/6465\)](https://support.wirenboard.com/t/podderzhka-mek-60870-5-104/6465)

Контроллер как источник SNMP

Иногда нужно получать какие-то параметры из контроллера по SNMP, например, значения топиков. Для реализации требуется установить и настроить на контроллере *агента* SNMP.

Решение базируется на демоне `snmpd`, который разрабатывается и поддерживается открытым сообществом. Здесь даны общие рекомендации, подробную информацию ищите в [официальной документации \(англ\)](http://www.net-snmp.org/docs/man/snmpd.html) (<http://www.net-snmp.org/docs/man/snmpd.html>).

Установка и предварительная настройка пакетов

Обновите источники и установите пакеты `snmp` и `snmpd`:

```
apt update && apt-get install snmp snmpd -y
```

Отредактируйте основной файл конфигурации демона:

```
mcedit /etc/snmp/snmpd.conf
```

В секцию `# ACCESS CONTROL` допишите строчки:

```
view systemonly included .1.3.6.1.4.1.2021.8
view systemonly included .1.3.6.1.4.1.2021.9
```

А в конец файла допишите ссылки на скрипты:

```
extend .1.3.6.1.4.1.2021.8 tt /var/lib/snmp/mqtt-snmp.sh
pass .1.3.6.1.4.1.2021.9 /var/lib/snmp/mqtt-snmp_pass.sh
```

Дополнительную информацию о параметрах конфигурационного файла можно в [документации snmpd](https://www.opennet.ru/cgi-bin/opennet/man.cgi?topic=snmpd.conf&category=5) (<https://www.opennet.ru/cgi-bin/opennet/man.cgi?topic=snmpd.conf&category=5>)

Добавление скрипта

Отредактируйте скрипт `/var/lib/snmp/mqtt-snmp.sh`

```
mcedit /var/lib/snmp/mqtt-snmp.sh
```

И приведите его к виду:

```
#!/bin/sh
echo "mosquitto topics"
# voltage Vout
echo $(/usr/bin/mosquitto_sub -C 1 -t "/devices/wb-adc/controls/5Vout" 2>/dev/null)
# voltage Vin
echo $(/usr/bin/mosquitto_sub -C 1 -t "/devices/wb-adc/controls/Vin" 2>/dev/null)
```

OID'ы отдаваемые этим скриптом могут иметь тип только *STRING*

Для демонстрации вывода других типов служит второй:

```
mcedit /var/lib/snmp/mqtt-snmp_pass.sh
```

Содержимое:

```
#!/bin/sh -f
echo .1.3.6.1.4.1.2021.9.1
echo integer
/bin/systemctl is-active wb-mqtt-serial > /dev/null && echo 1 || echo 0
```

Скрипт возвращает **INTEGER:1** если сервис `wb-mqtt-serial` запущен и **0** если нет.

Установите права:

```
chmod a+rx /var/lib/snmp/mqtt-snmp.sh
chmod a+rx /var/lib/snmp/mqtt-snmp_pass.sh
```

Теперь добавляем в этот скрипт нужные топики, или создаем несколько скриптов с разными наборами. Если вы создаете несколько скриптов, не забудьте [дописать ссылки на них в конец файла /etc/snmp/snmpd.conf](#).

После сохранения изменений в скрипте, перезапустите демон snmpd:

```
systemctl restart snmpd && systemctl status snmpd
```

Можно протестировать:

```
snmpwalk -On -v2c -c public 127.0.0.1 1.3.6.1.4.1.2021.8.4
```

и второй:

```
snmpwalk -On -v2c -c public 127.0.0.1 1.3.6.1.4.1.2021.9
```

Contents

[Описание](#)

[Установка](#)

[Создание сервиса](#)

[Удаление](#)

[Веб-интерфейс Node-RED](#)

[WB Nodes](#)

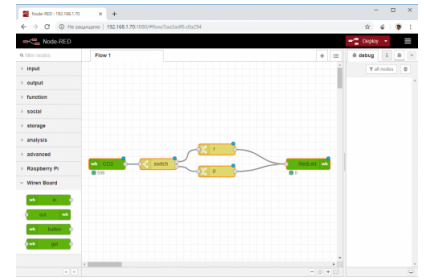
[Установка](#)

[Настройка](#)

[Проверка настроек](#)

[Примеры](#)

[Примеры использования](#)



Node-RED на WB6

Описание

Node-RED — это инструмент визуального программирования, основанный на Node.js и который разрабатывается и поддерживается открытым сообществом.

Подробное описание, инструкции и примеры использования можно найти на сайте nodered.org (<https://nodered.org/>), а поддержку сообщества в телеграм-канале [Node-RED](https://t.me/SprutAI_NodeRED) (https://t.me/SprutAI_NodeRED).

Копируйте команды из инструкции, вставляйте их в консоль контроллера с помощью клавиш **Shift+Insert** и запускайте клавишей **Enter**. В случае ошибок, внимательно читайте вывод, там будут инструкции и подсказки.

Установка

Для установки Node-RED на Wiren board 6 подключитесь к контроллеру по [SSH](#) и введите в консоль несколько команд.

Установите необходимые компоненты:

```
apt update && apt install -y nodejs git make g++ gcc build-essential
```

Запустите установку Node-RED:

```
npm install -g --unsafe-perm node-red
```

Если потребуется обновить версию Node-RED, выполните команду установки ещё раз.

Создание сервиса

Для автозапуска Node-RED необходимо настроить сервис. Для этого в папке `/etc/systemd/system/` создайте файл `nodered.service`:

```
mcedit /etc/systemd/system/nodered.service
```

Затем вставьте в этот файл строки:

```
[Unit]
Description=Node-RED graphical event wiring tool
Wants=network.target

[Service]
Type=simple
User=root
Group=root
WorkingDirectory=/home

Nice=5
Environment="NODE_OPTIONS=--max_old_space_size=256"
ExecStart=/usr/bin/env node-red $NODE_OPTIONS $NODE_RED_OPTIONS
KillSignal=SIGINT
Restart=on-failure
SyslogIdentifier=Node-RED

[Install]
WantedBy=multi-user.target
```

Сохраните изменения и закройте файл. Теперь включите сервис и запустите его командой:

```
systemctl enable nodered && systemctl start nodered
```

Проверьте статус сервиса командой, в выводе должна быть строчка `Active: active (running)`:

```
# systemctl status nodered
● nodered.service - Node-RED graphical event wiring tool
   Loaded: loaded (/etc/systemd/system/nodered.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-12-03 06:32:25 UTC; 24s ago
   Main PID: 2757 (node)
   CGroup: /system.slice/nodered.service
           └─2757 node /usr/bin/node-red --max_old_space_size=256

Dec 03 06:32:25 wrenboard-AWQBNTYP systemd[1]: Started Node-RED graphical event wiring tool.
Dec 03 06:32:36 wrenboard-AWQBNTYP Node-RED[2757]: 3 Dec 06:32:36 - [info]
Dec 03 06:32:36 wrenboard-AWQBNTYP Node-RED[2757]: Welcome to Node-RED
Dec 03 06:32:36 wrenboard-AWQBNTYP Node-RED[2757]: =====
Dec 03 06:32:36 wrenboard-AWQBNTYP Node-RED[2757]: 3 Dec 06:32:36 - [info] Node-RED version: v2.1.4
Dec 03 06:32:36 wrenboard-AWQBNTYP Node-RED[2757]: 3 Dec 06:32:36 - [info] Node.js version: v12.19.0
Dec 03 06:32:36 wrenboard-AWQBNTYP Node-RED[2757]: 3 Dec 06:32:36 - [info] Linux 5.10.35-wb6 arm LE
Dec 03 06:32:41 wrenboard-AWQBNTYP Node-RED[2757]: 3 Dec 06:32:41 - [info] Loading palette nodes
```

Удаление

Если вам больше не нужен Node-RED, то его можно удалить:

1. Остановите, отключите и удалите сервис:

```
systemctl stop nodered && systemctl disable nodered && rm /etc/systemd/system/nodered.service
```

2. Удалите сам Node-RED:

```
npm -g remove node-red && npm -g remove node-red-admin && rm -R ~/.node-red
```

Node.js, тоже можно удалить, но убедитесь, что у вас не установлены другие программы, которые его используют, например, zigbee2mqtt. Чтобы удалить node.js, выполните команду:

```
apt remove nodejs
```

Веб-интерфейс Node-RED

После запуска сервиса откройте браузер и введите адрес своего контроллера в сети и порт 1880. Например: <http://192.168.42.1:1880>

Инструкции по работе с Node-RED можно найти на [сайте nodered.org \(https://nodered.org/docs/getting-started/\)](https://nodered.org/docs/getting-started/)

WB Nodes

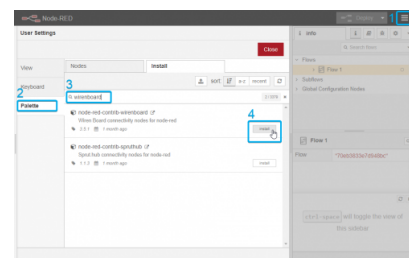
Чтобы упростить взаимодействие Node-RED с контроллером Wren Board, пользователь [Andrej Popov \(https://support.wrenboard.com/t/node-red-contrib-wrenboard/2019\)](https://support.wrenboard.com/t/node-red-contrib-wrenboard/2019) написал модуль `node-red-contrib-wrenboard`.

Установка

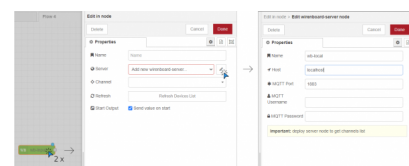
Установим его через [веб-интерфейс Node-RED](#):

1. Зайдите в веб-интерфейс Node-RED.
2. В правом верхнем углу вызовите меню.
3. В открывшемся окне выберите вкладку **Управление палитрой (Manager palette)**.
4. Перейдите на вкладку **Установка (Install)**, введите в поле поиска **wrenboard** и нажмите на клавиатуре **Enter**.
5. Установите пакет с названием **node-red-contrib-wrenboard**.
6. Закройте окно с настройками. Установка модуля завершена.

После установки, в левой панели появится секция Wren Board.



Установка WB Nodes на контроллер Wren Board



Настройка соединения с контроллером Wren Board

Настройка

После того как вы установили WB Nodes, вам нужно настроить соединение с MQTT-брокером контроллера Wren Board:

1. Добавьте в рабочую область ноду **WB-input**, которая находится в левой панели, в секции **Wren Board**.
2. Дважды кликните на ней, откроется окно **Edit in node**.
3. Найдите в нём поле **Server** и нажмите на кнопку с карандашом. Откроется окно **Add new wrenboard-server config node**.
4. Заполните поля **Name**, **Host** и **MQTT Port**. Если Node-RED установлен на контроллере — в поле Host пропишите `localhost`.
5. Сохраните настройки сервера нажатием на кнопку **Done**.
6. Закройте окно **Edit in node** нажатием на **Cancel**.

После настройки подключения, примените настройку сервера нажатием кнопки **Deploy** справа сверху. Только после этого будет доступен список каналов контроллера Wiren Board.

Вы настроили WB Nodes и можете писать автоматизации с использованием Node-RED.

Проверка настроек

Чтобы проверить правильность настроек:

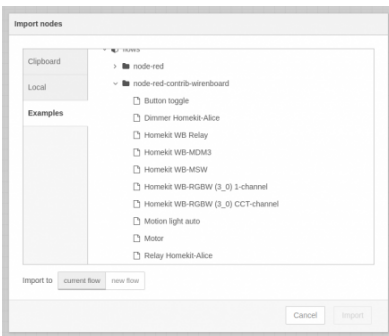
1. Добавьте в рабочую область любую ноду из группы Wiren Board, например, **WB-input**.
2. Дважды кликните на ноде **WB-input**, откроется окно **Edit in node**.
3. В окне **Edit in node** нажмите кнопку **Refresh Device List**.

Если соединение настроено верно, то в выпадающем списке поля **Channel** появится список доступных для использования каналов.

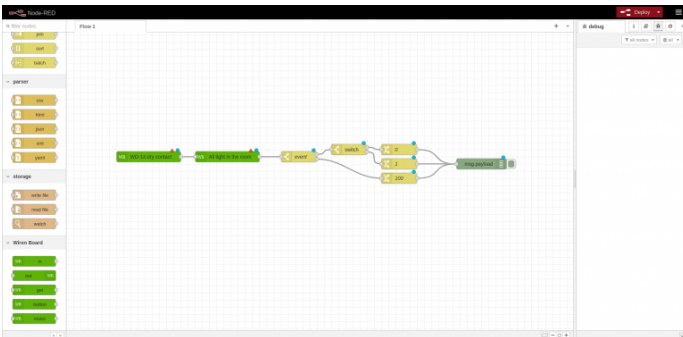
Примеры

В комплекте с WB-Nodes есть примеры для Apple HomeKit и Яндекс.Алисы, найти их можно в **Menu** → **Import** → **Examples**.

- Примеры из комплекта WB-Nodes



Список примеров



Пример в рабочей области

Примеры использования

Примеры использования Node-RED с WB-Nodes на контроллерах Wiren Board:

- Настройка Telegram-бота.
- Настройка Алисы на контроллере Wiren Board.
- Настройка Apple HomeKit на контроллере Wiren Board.
- Видео от нашего партнёра: WIRENBOARD + NODERED (<https://youtu.be/cKQmU4LZo4c>)

Установка Home Assistant на контроллер Wiren Board

Contents

Описание

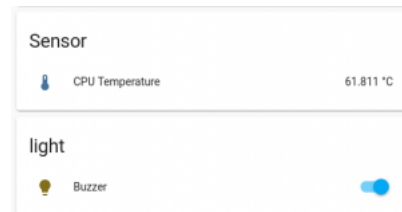
Установка

Настройка устройств Wiren Board

Установка MQTT-интеграции

Настройка связи с устройствами

Полезные ссылки



Устройства в Home Assistant

Описание

Home Assistant — это open-source платформа для автоматизации, которую разрабатывает и поддерживает сообщество [home-assistant.io](https://www.home-assistant.io/) (<https://www.home-assistant.io/>). Существует одноимённый мобильный клиент, который может управлять контроллером по локальной сети или через облако производителя.

Платформу можно установить на контроллер Wiren Board и использовать вместо стандартного веб-интерфейса для управления подключёнными устройствами.

Установка

Встроенный флеш-накопитель контроллера разбит на разделы и для пользователя отведён самый большой из них, который монтируется в папку `/mnt/data`. Нужно учесть эту особенность при установке программ, а также при обновлении прошивки контроллера, которое нужно делать только через менеджер пакетов apt.

Копируйте команды из инструкции, вставляйте их в консоль контроллера с помощью клавиш `Shift+Insert` и запускайте клавишей `Enter`. В случае ошибок, внимательно читайте вывод, там будут инструкции и подсказки.

Мы будем использовать рекомендуемый разработчиком платформы метод **Home Assistant Container**:

1. Установите docker по инструкции.
2. Создайте каталог под служебные файлы и сделайте на него симлинк:

```
mkdir /mnt/data/root/HA && ln -s /mnt/data/root/HA /HA
```

3. Запустите образ homeassistant — docker автоматически загрузит его из интернет и запустит:

```
docker run -d --name homeassistant --privileged --restart=unless-stopped -e TZ=Europe/Moscow -v /HA:/config --network=host ghcr.io/home-assistant/home-assistant:stable
```

После установки и запуска, откройте браузер и введите адрес своего контроллера в сети и порт 8123. Например: <http://192.168.42.1:8123> Консоль можно закрыть.

Ресурсы, потребляемые Home Assistant

```
root@wirenboard-AWE7DENS:~# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/root       ext4      979M  804M  108M  89% /
devtmpfs        devtmpfs  492M   0 492M   0% /dev
tmpfs           tmpfs     500M   0 500M   0% /dev/shm
tmpfs           tmpfs     500M  1.8M  498M   1% /run
tmpfs           tmpfs     5.0M   0 5.0M   0% /run/lock
tmpfs           tmpfs     500M   0 500M   0% /sys/fs/cgroup
/dev/mmcblk0p6  ext4      4.7G  1.8G  2.8G  39% /mnt/data
tmpfs           tmpfs     100M   0 100M   0% /run/user/0
overlay         overlay   4.7G  1.8G  2.8G  39% /mnt/data/root/docker/overlay2/
d1b02ce168fbd49df2a9b2268a/merged
```

После установки Home Assistant для пользовательских данных остаётся ещё 60 % пространства в разделе `/mnt/data`

```
top - 19:05:20 up 1:37, 1 user, load averages: 0.01, 0.53, 0.49
Tasks: 341 total, 2 running, 97 sleeping, 0 stopped, 0 zombie
Mem(s): 1.5 us, 2.1 sy, 0.0 ni, 95.8 id, 0.0 wa, 0.0 hi, 0.7 si, 0.0 st
MiB Mem: 1022812 total, 245298 free, 384876 used, 219328 buff/cache
KiB Swap: 262140 total, 262140 free, 0 used, 726976 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1760 root      20   0  75700 8496  6376  S  1.0  0.8  2:05.72  wd-mqtt-serial
 760 mosquito 20   0  9800 5676  2820  S  1.2  0.6  1:39:21  mosquitto
 1732 root      20   0  23428 11516 5600  S  1.3  1.1  1:09:40  wb-metrics
 1762 root      20   0  948752 21876 9872  S  1.3  2.1  2:41:46  wb-rules
 2740 root      20   0  3532 2428  1876  R  1.3  0.2  0:00:40  top
 1742 root      20   0  64652 6200 5260  S  0.7  0.6  0:52.83  main
 11 root      20   0  0 0 0  S  0 0 0:00:00  rcu_sched
 80 root     -SI  0  0  0  S  0 0 0:01:05  /usr/bin/sd
 989 root      20   0  0 0 0  S  0 0 0:10:16  RTM_CMD_THREAD
 1738 root      20   0  15710 12764 5980  S  0.2  1.2  0:04:31  wb-flaps-collect
 4443 root      20   0  63548 6220 5604  S  0.3  0.6  0:09:11  wb-mqtt-ogio
 1 root      20   0  27292 5600 3792  S  0.8  0.6  0:05:38  systemd
 2 root      20   0  0 0 0  S  0 0 0:00:00  kthreadd
```

Вывод утилиты `top` при запущенном Home Assistant

Настройка устройств Wiren Board

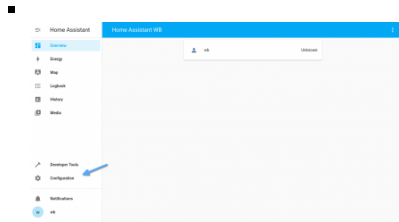
Установка MQTT-интеграции

Home Assistant может управлять Modbus-устройствами с помощью MQTT, для этого нужно установить MQTT-интеграцию и настроить получение и отправку данных.

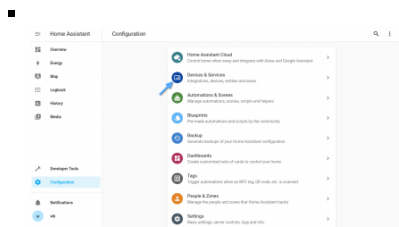
Установите MQTT-интеграцию:

1. Перейдите в веб-интерфейс Home Assistant.
2. В панели слева кликните на пункт **Configuration**.
3. На открывшейся странице найдите пункт **Devices & Services** и зайдите в него.
4. Нажмите внизу на кнопку *Add Integration*'.
5. Введите в поле поиска **MQTT** и выберите найденную интеграцию.
6. Через несколько секунд откроется окно настройки, в котором заполните поля **Broker** — localhost и **Port** — 1883. В стандартной конфигурации контроллера логин и пароль не используется, эти поля оставьте пустыми. Нажмите кнопку **Submit**.

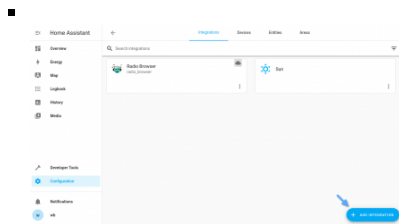
■ Установка MQTT-интеграции



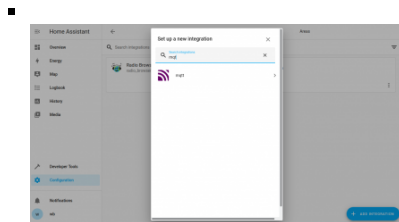
Configuration



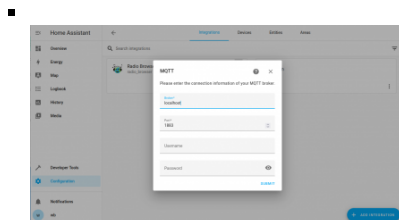
Devices & Services



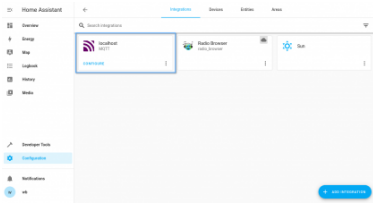
Add Integration



Поиск интеграции



Настройка интеграции



Установленная интеграция

Настройка связи с устройствами

Настройка интеграции завершена, теперь надо для каждого устройства создать описание в файле **configuration.yaml**, для этого откройте файл с помощью редактора:

```
mcedit /mnt/data/root/HA/configuration.yaml
```

В процессе настройки вы будете добавлять в файл секции и описывать в них параметры, например, для описания реле нужно использовать секцию **switch**.

После редактирования сохраните файл и перезапустите Home Assistant, для этого перейдите в его веб-интерфейсе **Configuration** → **Settings** и нажмите кнопку **Restart**. После нажатия на кнопку файл конфигурации будет проверен и, если он содержит ошибки, вы получите сообщение с подробностями. Чаще всего ошибки связаны с неверными отступами строк, которые имеют значение.

Так как место на контроллере не очень много, то рекомендуем ограничить количество сохраняемых в архиве. Добавьте для этого в файл настроек строки, где 14 — глубина архива в днях:

```
recorder:
  purge_keep_days: 14
```

Несколько моментов:

- Полный синтаксис и поддерживаемые типы устройств смотрите в описании [MQTT Discovery \(https://www.home-assistant.io/docs/mqtt/discovery/\)](https://www.home-assistant.io/docs/mqtt/discovery/).
- Адреса MQTT-топиков смотрите в стандартном веб-интерфейсе контроллера, раздел **Settings** → **MQTT Channels**.
- В *state_topic* пишете адрес MQTT-топика, а в *command_topic* — тот же адрес, только в конец добавляете */on*.

Выведем данные с датчика температуры процессора:

```
sensor:
- platform: mqtt
  name: "CPU Temperature"
  unique_id: wb_cpu_temperature
  state_topic: "/devices/hwmon/controls/CPU Temperature"
  unit_of_measurement: "°C"
```

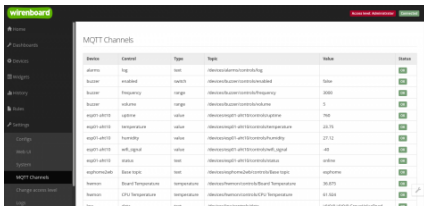
Настроим управление выходами **A1** и **A2**, этот пример можно использовать для управления модулями реле:

```
switch:
- platform: mqtt
  name: "WB A1"
  unique_id: wb_A1
  command_topic: "/devices/wb-gpio/controls/A1_OUT/on"
  state_topic: "/devices/wb-gpio/controls/A1_OUT"
  payload_on: "1"
  payload_off: "0"
  retain: true
- platform: mqtt
  name: "WB A2"
  unique_id: wb_A2
  command_topic: "/devices/wb-gpio/controls/A2_OUT/on"
  state_topic: "/devices/wb-gpio/controls/A2_OUT"
  payload_on: "1"
  payload_off: "0"
  retain: true
```

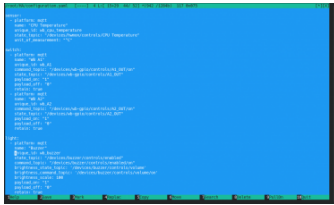
Настроим управление зуммером контроллера, этот пример можно использовать при подключении диммера светодиодных ламп:

```
light:
- platform: mqtt
  name: "Buzzer"
  unique_id: wb_buzzer
  state_topic: "/devices/buzzer/controls/enabled"
  command_topic: "/devices/buzzer/controls/enabled/on"
  brightness_state_topic: '/devices/buzzer/controls/volume'
  brightness_command_topic: '/devices/buzzer/controls/volume/on'
  brightness_scale: 100
  payload_on: "1"
  payload_off: "0"
  retain: true
```

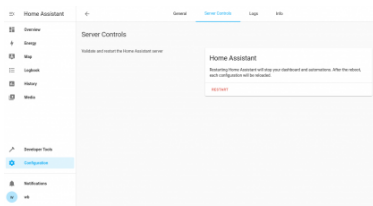
■ Настройка Modbus-устройства Wren Board



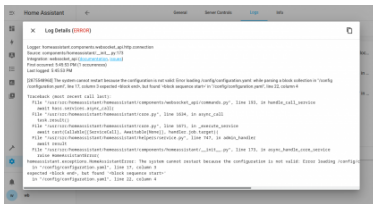
MQTT-топики в веб-интерфейсе контроллера



Редктирование файла конфигурации



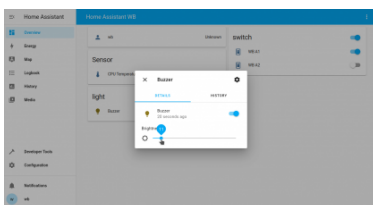
Кнопка перезагрузки Home Assistant



Пример сообщения об ошибке



Устройства в веб-интерфейсе Home Assistant



Управление зуммером

Полезные ссылки

- [Оригинальная инструкция по установке Home Assistant \(https://www.home-assistant.io/installation/linux#install-home-assistant-container\)](https://www.home-assistant.io/installation/linux#install-home-assistant-container)
- [Описание Home Assistant MQTT Discovery \(https://www.home-assistant.io/docs/mqtt/discovery/\)](https://www.home-assistant.io/docs/mqtt/discovery/)
- [Автоматическое добавление serial-устройств в Home Assistant, решение от пользователя \(https://support.wirenboard.com/t/integracziya-wirenboard-s-homeassistant/9008/31\)](https://support.wirenboard.com/t/integracziya-wirenboard-s-homeassistant/9008/31)
- [Wiren Board MQTT Conventions \(https://github.com/wirenboard/conventions/blob/main/README.md\)](https://github.com/wirenboard/conventions/blob/main/README.md)
- [Установка Docker на контроллер Wiren Board](#)
- [Преобразование RGB формата для диммера WB-MRGBW-D \(https://support.wirenboard.com/t/wb-mrgbw-d-i-homeassistant-cherez-mqtt/11566\)](https://support.wirenboard.com/t/wb-mrgbw-d-i-homeassistant-cherez-mqtt/11566)

Contents

Описание

Подготовка к установке

Предварительная настройка

Установка

Полезные ссылки

Описание

Docker (<https://docker.com>) — программное обеспечение для запуска приложений в изолированной среде. С его помощью можно упаковать приложение со всем его окружением и зависимостями в контейнер и потом развернуть его на любой ОС Linux. Такой подход позволяет избежать конфликта используемых приложением библиотек с установленными в основной системе.

С помощью docker-контейнеров распространяется много открытого ПО, например, [Home Assistant](https://www.home-assistant.io/) и [openHAB](https://www.openhab.org/) (<https://www.openhab.org/>).

```
root@wiredboard-AWE7DENS:~# docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (WARNING: Please verify the official sites to identify the image ID, digest and parent to assure security)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Проверка работы docker

```
root@wiredboard-AWE7DENS:~# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/root       ext4      979M  804M  108M  89% /
devtmpfs        devtmpfs  492M   0  492M   0% /dev
tmpfs           tmpfs     500M   0  500M   0% /dev/shm
tmpfs           tmpfs     500M  1.6M  498M   1% /run
tmpfs           tmpfs     5.0M   0  5.0M   0% /run/lock
tmpfs           tmpfs     500M   0  500M   0% /sys/fs/cgroup
/dev/mmcblk0p6  ext4      4.7G  358M  4.1G   8% /mnt/data
tmpfs           tmpfs     100M   0  100M   0% /run/user/0
```

После установки docker, в разделе /mnt/data остаётся много свободного места для образов и пользовательских данных

Подготовка к установке

Копируйте команды из инструкции, вставляйте их в консоль контроллера с помощью клавиш **Shift+Insert** и запускайте клавишей **Enter**. В случае ошибок, внимательно читайте вывод, там будут инструкции и подсказки.

Установите необходимые зависимости:

```
apt update && apt install ca-certificates curl gnupg lsb-release
```

Добавьте репозиторий с пакетами docker:

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Добавьте GPG ключ для репозитория:

```
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Предварительная настройка

Встроенный флеш-накопитель контроллера разбит на разделы и для пользователя отведён самый большой из них, который монтируется в папку /mnt/data. Нужно учесть эту особенность при установке программ, а также при обновлении прошивки контроллера, которое нужно делать только через [менеджер пакетов apt](#).

Настройте симлинк для папки конфигурации:

```
mkdir /mnt/data/etc/docker && ln -s /mnt/data/etc/docker /etc/docker
```

Мы будем хранить образы на встроенном накопителе, если вам нужно больше места под образы — используйте внешнюю флешку.

Создайте папку для хранения образов:

```
mkdir /mnt/data/.docker
```

Укажите в файле настроек **daemon.json** созданную выше папку:

1. Откройте файл в редакторе:

```
mcedit /etc/docker/daemon.json
```

2. Вставьте в него строки:

```
{  
  "data-root": "/mnt/data/.docker"  
}
```

3. Сохраните и закройте файл.

Установка

После того, как мы указали, где будут храниться контейнеры, устанавливаем сам docker:

```
apt update && apt install docker-ce docker-ce-cli containerd.io
```

Чтобы проверить, что всё работает — запустите контейнер hello-world:

```
docker run hello-world
```

Если в консоли появилась надпись Hello from Docker!, docker установлен и работает.

Полезные ссылки

- [Установка Home Assistant через docker](#)
- [Оригинальная инструкция по установке Docker \(https://docs.docker.com/engine/install/debian/\)](https://docs.docker.com/engine/install/debian/)

Contents

Общая информация

[Совместимость](#)

[Кратко о релизах](#)

[Какой релиз на вашем контроллере](#)

[Переключение между релизами](#)

Пользовательские настройки и файлы

[Где хранятся настройки](#)

[Резервное копирование](#)

Проверка обновлений

Обновление

[В консоли через apt](#)

[Через веб-интерфейс](#)

Удаление данных и другие способы обновления

Общая информация

Совместимость

Инструкции на этой странице подходят для контроллеров Wiren Board 5.x, 6.x, 7.x.

Исключения:

- Контроллеры Wiren Board 5.x с версией прошивки 0.46-20190613 — их можно обновить только через [восстановление прошивки](#).
- Контроллеры Wiren Board 7.2.1A, выпущенные в декабре 2021 г — перед обновлением или возвратом заводских настроек, [переключите их на релиз](#), а потом используйте инструкции на этой странице.

Контроллеры Wiren Board 4 и старше [прошиваются через карту microSD](#).

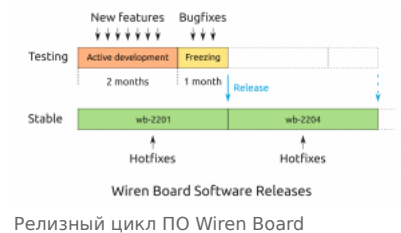
Кратко о релизах

Программное обеспечение контроллеров Wiren Board состоит из множества пакетов, которые мы объединяем в релизы:

- Stable** — стабильный релиз: обновление пакетов раз в три месяца и исправления критических ошибок. Рекомендуем для инсталляций, которые делает интегратор и которые работают сейчас.
- Testing** — тестовый релиз: свежие версии пакетов с новыми функциями и исправлениями, но возможно и с новыми ошибками. Рекомендуем для активных пользователей (DIY), для инсталляций на стадии стройки и пусконаладки.

Стабильные релизы имеют номер вида *wb-YYMM*, где *YY* — год, а *MM* — месяц выпуска. Например, *wb-2104* — релиз, выпущенный в апреле 2021 года.

Мы будем рады, если вы сможете присоединиться к использованию *testing*, ведь кроме получения новых фиш вы сможете уменьшить количество ошибок в *stable*.



| Журналы изменений в релизах | |
|-----------------------------|---|
| Ветка | Имя |
| stable | wb-2207 (июль-август) • wb-2204 • wb-2201 (https://wirenboard.com/statics/release-changelogs/wb-2201/changelog.html) • wb-2110 (https://wirenboard.com/statics/release-changelogs/wb-2110/changelog.html) • wb-2108 (https://wirenboard.com/statics/release-changelogs/wb-2108/changelog.html) • wb-2104 |
| testing | скользящий релиз, изменения (https://t.me/wirenboard_testing) |

Какой релиз на вашем контроллере

С завода на контроллерах Wiren Board установлен актуальный на момент производства стабильный релиз.

Узнать версию релиза можно в веб-интерфейсе контроллера в разделе *Devices* в карточке устройства *System* или в консоли командой `wb-release`. Если в веб-интерфейсе нет упоминания о *testing* или *stable* или команда `wb-release` не найдена — у вас старая версия ПО и нужно [сменить репозиторий](#).

Переключение между релизами

Можно вернуть можно переключатся вручную. Способы возврата к заводским настройкам



между релизами можно переключаться вручную. Сделайте резервную копию настроек и выполните одну из команд:

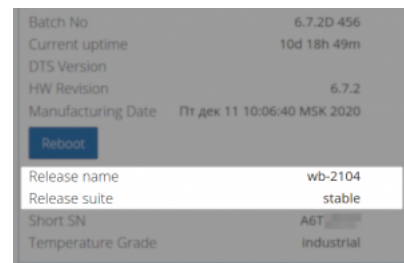
- Переход со стабильного на тестовый:

```
wb-release -t testing
```

- Переход с тестового на стабильный:

```
wb-release -t stable
```

После смены релиза рекомендуем перезагрузить контроллер на случай, если обновилось ядро, а также обновить страницу веб-интерфейса в браузере со сбросом кэша (Ctrl+F5 для Chrome), чтобы загрузилась обновленная версия.



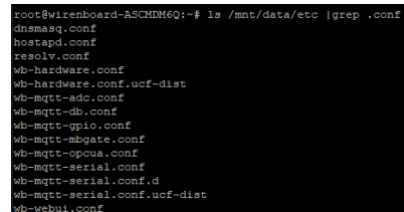
Версия ПО в веб-интерфейсе контроллера
WebUI → Devices → System

Пользовательские настройки и файлы

Где хранятся настройки

В контроллере Wiren Board есть отдельный раздел, который монтируется в каталог /mnt/data, в котором по адресу /mnt/data/etc находятся настройки:

- системные: сеть, часовой пояс, пароль к веб-интерфейсу, mosquitto, nginx;
- устройств, подключенных по RS-485 — wb-mqtt-serial.conf;
- модулей ввода-вывода и расширения — wb-hardware.conf;
- универсальных входов/выходов A1-A4 — wb-mqtt-adc.conf;
- выводов GPIO контроллера — wb-mqtt-gpio.conf;
- настройки архива данных — wb-mqtt-db.conf;
- шлюза OPC UA — wb-mqtt-opcua.conf;
- шлюза Modbus TCP/Slave — wb-mqtt-mbgate.conf.



Файлы настроек, которые сохраняются при обновлении через fit-файл

Кроме этого, по адресу /mnt/data/etc/ хранятся:

- wb-rules — пользовательские скрипты;
- wb-rules-module — модули, написанные на wb-rules;
- wb-mqtt-serial.conf.d/templates/ — пользовательские шаблоны.

Где хранятся настройки установленного стороннего ПО, уточняйте в его документации.

Резервное копирование

Чтобы сделать резервную копию настроек контроллера, скопируйте содержимое /mnt/data/etc на компьютер. Программы, установленные в /mnt/data также нужно сохранить на компьютер.

Если вы устанавливали на контроллер сторонние программы, или хранили файлы вне папки /mnt/data — их нужно сохранить отдельно.

Проверка обновлений

В контроллере нет механизма, который сообщит пользователю о доступном обновлении, поэтому о выходе новых версий вы можете узнать из новостей в [наших социальных сетях \(https://wirenboard.com/ru/pages/contacts/\)](https://wirenboard.com/ru/pages/contacts/) или в консоли контроллера:

1. Подключитесь к контроллеру через [SSH](#).
2. Выполните команду:

```
apt update
```

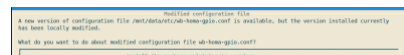
3. Если есть пакеты для обновления, то можете посмотреть их список:

```
apt list --upgradable
```

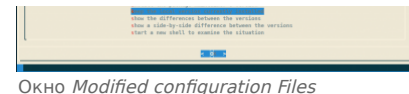
Обновление

В консоли через apt

Рекомендуемый способ



Арт — это менеджер пакетов операционной системы Debian, который обновляет изменившиеся пакеты и устанавливает новые, если это необходимо. Для работы нужен интернет или локальное зеркало (<https://wiki.debian.org/ru/CreateLocalRepo>) арт-репозитория Wiren Board.



Окно Modified configuration Files

Обновление контроллера через арт учитывает систему релизов — следуя инструкции, вы полностью перейдёте на новый релиз, а не только обновите отдельные пакеты.

При обновлении сохраняются установленные программы, пользовательские файлы и настройки.

Чтобы обновить прошивку контроллера:

1. Подключитесь к нему по SSH.
2. Выполните команды:

```
apt update && apt upgrade
```

В процессе обновления может появиться запрос на действие с файлами конфигурации. Если не знаете, что выбрать — оставьте значение по умолчанию **keep the local version currently installed**, так вы сохраните свои настройки.

Так же в консоль будут выводиться служебные сообщения и запросы на действия, которые могут повредить систему или удалить пользовательские настройки — внимательно читайте вопросы перед тем, как ввести Y.

Через веб-интерфейс

Вам понадобится fit-файл прошивки для вашей версии контроллера. Самые свежие stable-релизы находятся по ссылкам:

- Wiren Board 5.3 (http://fw-releases.wirenboard.com/fit_image/stable/5/latest_stretch.fit), Wiren Board 5.6.x (http://fw-releases.wirenboard.com/fit_image/stable/55/latest_stretch.fit), Wiren Board 5.8.x-5.9 (http://fw-releases.wirenboard.com/fit_image/stable/58/latest_stretch.fit);
- Wiren Board 6.3-6.6.0 (http://fw-releases.wirenboard.com/fit_image/stable/6x/latest_stretch.fit), Wiren Board 6.7.x-6.9.x (http://fw-releases.wirenboard.com/fit_image/stable/67/latest_stretch.fit);
- Wiren Board 7.x (http://fw-releases.wirenboard.com/fit_image/stable/7x/latest_stretch.fit)

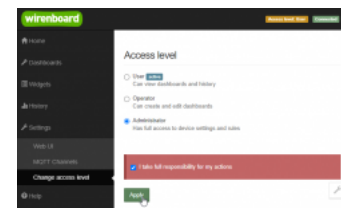
А все доступные образы лежат в общей папке в нашем репозитории (http://fw-releases.wirenboard.com/?prefix=fit_image/stable/).

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.

При обновлении сохраняются пользовательские файлы и настройки в `/mnt/data`, но стороннее ПО будет удалено. Рекомендуем сделать резервную копию.

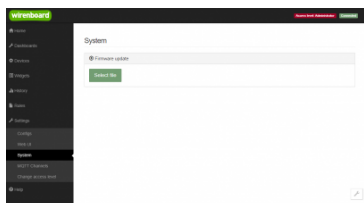


Уровень «Администратор»

Чтобы обновить прошивку контроллера:

1. Скачайте fit-файл прошивки на компьютер.
2. Зайдите в веб-интерфейс контроллера и перейдите в раздел **Settings** → **System**. В старых версиях WebUI кнопка *Select file* находится в разделе **Settings**.
3. Нажмите кнопку **Select File** и выберите скачанный ранее fit-файл.
4. Файл с прошивкой загрузится на контроллер и начнётся обновление, которое длится 5-10 минут. Не закрывайте страницу и не выключайте контроллер до завершения.
5. После обновления контроллер перезагрузится и на странице появится надпись **Firmware update complete** — обновление завершено.
6. Закройте страницу.

После завершения обновления перезагрузите контроллер на случай, если обновилось ядро, а также обновите страницу веб-интерфейса в браузере со сбросом кэша (Ctrl+F5 для Chrome), чтобы загрузилась обновленная версия.

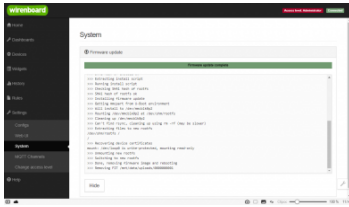


Нажмите кнопку **Select file** и выберите fit-файл





Во время обновления на страницу выводятся системные сообщения



Оповещение об успешном обновлении

Удаление данных и другие способы обновления

Обновление через арт или веб-интерфейс полностью покрывают задачи по обслуживанию контроллера.

Однако, если они не подходят или вам нужно удалить данные с возвратом к заводским настройкам — это тоже можно, читайте в инструкции для каждой версии контроллера по ссылкам [Wiren Board 5.x](#), [Wiren Board 6.x](#) и [Wiren Board 7.x](#).

Замечание: эта статья про то, как создавать новое программное обеспечение для Wiren Board, и предназначена для программистов;

- читайте описание стандартного ПО Wiren Board на странице [Программное обеспечение Wiren Board](#),
- используйте встроенную систему правил для написания правил и сценариев.

Contents

Окружение

[Общие соображения](#)

[Сборка пакетов и программ](#)

[Тулчейн \(toolchain\)](#)

[Окружение для разработки](#)

[Полезные ссылки](#)

Окружение

Все контроллеры Wiren Board поставляются с полноценным Debian Linux. Архитектура процессора: armhf (Wiren Board 6) или armel (Wiren Board 5 и раньше). На контроллере используется libc версии 2.13.

Установку ПО рекомендуется производить с помощью пакетного менеджера Debian, и упаковывать ПО в deb-пакеты

Общие соображения

В зависимости от модели контроллеры Wiren Board имеют от 64 до 1024 МВ оперативной памяти. Это значит, что необходимо учитывать потребление памяти при разработке своих программ. Про то, как контролировать потребление памяти в Linux читайте на сайте <http://www.linuxatemyram.ru/>.

Сборка пакетов и программ

Сборку программ рекомендуется производить на компьютере с Linux, используя специальное окружение для разработки. Компиляция и разработка непосредственно на контроллерах Wiren Board не рекомендуется из-за маленького объёма оперативной памяти.

Тулчейн (toolchain)

Если ваша программа не имеет внешних зависимостей от библиотек операционной системы, то для её сборки можно установить тулчейн.

Название тулчейна:

- для Wiren Board 6 и выше: **AArch32 target with hard float (arm-linux-gnueabihf)**
- для Wiren Board 5 и ниже: **arm-linux-gnueabi**

Скачать тулчейн для вашей операционной системы можно на сайте [ARM.COM \(https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-a/downloads\)](https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-a/downloads)

В дистрибутивах Linux необходимые тулчейны обычно уже доступны как пакеты. Например, для Ubuntu или Debian тулчейн можно установить командой:

```
apt install gcc-arm-linux-gnueabihf
```

Окружение для разработки

Окружение для разработки удобно использовать для сборки программ с внешними зависимостями от системных библиотек, для упаковки программ как Debian-пакетов и для удобной сборки под несколько версий Wiren Board. Это рекомендуемый и самый удобный способ.

Окружение предоставляется в виде Docker-контейнера, а для его использования в Linux существует удобный скрипт `wbdev`.

При использовании окружения, кросс-компиляция не нужна. Скрипт `wbdev` автоматически запускает соответствующие команды в виртуализованном окружении с архитектурой `armel`, соответствующей архитектуре процессора в контроллере Wiren Board.

Сборка пакетов производится с помощью вызова `wbdev`, например так:

```
$ wbdev chroot #запустить виртуализованное qemu chroot окружение с Debian 7 архитектуры armel
$ wbdev make # вызвать make в виртуализованном qemu chroot окружении

$ wbdev cdeb # собрать пакет, написанный на C++ в виртуализованном qemu chroot окружении
$ wbdev gdeb # собрать пакет, написанный на Go, с использованием кросс-компиляции Go
$ wbdev ndeb # собрать архитектурно-независимый пакет, например содержащий проект на Python
```

Полное описание работы с окружением смотрите в [документации на Github \(https://github.com/contactless/wirenboard/blob/master/README.md\)](https://github.com/contactless/wirenboard/blob/master/README.md).

Полезные ссылки

- [Сборка ядра](#)
- [Сборка образов прошивки](#)
- [Уникальные идентификаторы](#) — для идентификации устройства, привязки софта
- [Узнать степень износа внутреннего накопителя](#)
- [Пересборка Device Tree](#)

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

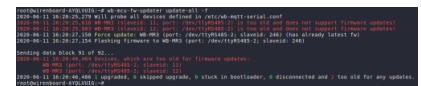
В наших modbus-устройствах реализован механизм загрузчика прошивок — bootloader. Он позволяет обновлять прошивки устройств и модулей Wiren Board по RS-485/Modbus RTU. В режиме загрузчика основные функции устройства отключаются, а коммуникационные параметры в режиме загрузчика фиксированы и не зависят 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 контроллеры 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 устройства можно использовать утилиту `wb-mcu-fw-flasher`.

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](#)

- English
- русский

Contents

Использование встроенного чипа АТЕССх08 для авторизации на внешних сервисах.

Настройка nginx.

Настройка openssl

Настройка mosquitto

Использование встроенного чипа АТЕССх08 для авторизации на внешних сервисах.

В контроллеры **WirenBoard** встроен чип **АТЕССх08**, назначением которого является генерация ключевых пар, хранение приватных ключей а также операции асимметричного шифрования с использованием эллиптических кривых. Приватный ключ после начальной инициализации хранится в микросхеме и не покидает ее, тем самым исключается его компрометация.

Используя данную микросхему можно организовать авторизацию контроллера и защиту соединения по SSL на внешних сервисах и быть уверенным, что запрос выполнен с использованием именно этого экземпляра микросхемы.

В данной статье пойдет речь о трех вариантах применения: **nginx**, **openssl**, **mosquitto**.

Для доступа к криптоустройству нужна библиотека libateccssl1.1, установим ее командой:

apt install libateccssl1.1

Далее нужно отредактировать файл **/etc/ssl/openssl.cnf**, добавив в нем следующие строчки:

```
openssl_conf = openssl_init

[openssl_init]
engines      = engine_section

[engine_section]
ateccx08     = ateccx08_section

[ateccx08_section]
init        = 1
```

Теперь приступим к созданию сертификатов. Для начала создадим свой центр сертификации (Certification Authority, **CA**):

Для этого сгенерируем ключевую пару:

```
openssl genrsa -out ca.key 2048
```

И сертификат нашего **CA**:

```
openssl req -x509 -new -days 3650 -key ca.key -out ca.crt -subj "/CN=MY CA"
```

CA является основой безопасности в данной схеме, поэтому эти операции выполняем на машине доступ к которой есть только у владельца **CA**.

Далее на контроллере **WB** создаем запрос на сертификат устройства:

```
openssl req -new -engine ateccx08 -keyform engine -key ATECCx08:00:04:C0:00 -subj "/CN=wirenboard-AP6V5MDG" -out device_AP6V5MDG.csr
```

В этой команде мы указываем, что запрос подписывается приватным ключом, находящимся в криптоустройстве ateccx08 ключом с идентификатором **АТЕССх08:00:04:C0:00** и публичным именем wirenboard-AP6V5MDG. Запрос помещаем в файл device_AP6V5MDG.csr.

Имя можно выбрать любое уникальное, удобно для этой цели спользовать идентификатор устройства, который по умолчанию прописывается в файле **/etc/hostname**:

```
cat /etc/hostname
wirenboard-AP6V5MDG
```

Далее этот запрос подписываем в нашем центре сертификации: **openssl x509 -req -in device_AP6V5MDG.csr -CA ca.crt -CAkey ca.key -out device_AP6V5MDG.crt -days 365 -CAcreateserial**

В этой команде мы уазываем файл запроса, и файлы **CA** необходимые для подписи. В итоге получаем сертификат устройства device_AP6V5MDG.crt. Файл **device_AP6V5MDG.crt** копируем на контроллер **WB**, он будет необходим для авторизации.

Давайте посмотрим, что содержится в файлах сертификатов **CA** и устройства:

```
openssl x509 -text -noout -in device_AP6V5MDG.crt
Certificate:
```

```
Data:
  Version: 1 (0x0)
  Serial Number:
    bf:85:29:be:19:67:f5:3e
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: CN = MY CA
  Validity
    Not Before: Feb  4 14:50:14 2019 GMT
    Not After : Feb  4 14:50:14 2020 GMT
  Subject: CN = wirenboard-AP6V5MDG
  Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (256 bit)
      pub:
        04:66:80:f6:83:ea:4f:88:a5:05:df:8f:2c:62:f3:
        ad:71:55:87:7f:ae:12:ae:b1:74:4b:68:68:fd:f7:
        e0:8a:f4:44:87:45:ab:c1:07:3f:54:2a:a9:ea:c6:
        71:1d:41:63:67:1b:75:f4:00:42:8d:fd:f6:d5:b6:
        52:38:e8:5a:a9
      ASN1 OID: prime256v1
      NIST CURVE: P-256
  Signature Algorithm: sha256WithRSAEncryption
  be:d1:f8:04:fb:34:a9:84:ff:25:b6:04:04:c0:f1:1d:4a:a4:
  04:b8:54:6c:a8:46:61:5f:6c:e7:ab:16:8f:ae:45:46:02:99:
  c6:d3:90:42:91:20:c7:89:d5:cf:4e:23:3a:33:64:ab:1b:c9:
  78:18:82:f4:39:8b:97:ae:6c:ee:a4:13:0c:5a:54:6b:69:c8:
  1e:fa:24:3d:48:2c:ea:0e:5c:0d:c3:43:c2:49:ea:b2:f8:5e:
  d7:0b:b5:4e:67:87:53:84:76:23:aa:10:77:5d:f1:21:9e:b0:
  4b:16:99:7c:d4:d3:d6:e7:00:9c:bf:53:a1:4b:f4:2c:fc:0b:
  64:10:fb:77:fc:3d:b2:71:cf:be:0b:b1:a2:62:ed:8c:92:e4:
  78:73:dc:69:c4:61:10:22:66:11:11:8b:d4:3c:b6:4f:7f:2c:
  24:07:61:47:15:2a:56:7e:71:69:59:15:8b:53:c8:e2:b5:ed:
  34:a0:78:70:d4:f6:cf:0f:6d:df:45:00:3b:0a:39:a2:fb:e7:
  89:f3:d9:88:7f:6b:bd:fa:ca:5e:44:94:74:70:5e:86:0b:93:
  ca:16:71:42:67:eb:77:bd:15:e3:90:2f:68:fd:bc:61:25:a3:
  a6:e7:8b:b1:42:bc:c2:36:d4:17:67:b3:77:fb:bd:06:e9:35:
  3b:8e:08:48
```

Видим, что сертификат выписан Центром сертификации с именем "MY CA" на 1 год, начиная с "Feb 4 14:50:14 2019 GMT" (для этого мы указывали `-days 365` в команде подписи), устройству с именем **wirenboard-AP6V5MDG**, имеющим приватный ключ соответствующий публичному Subject Public Key Info. Сертификат подписан цифровой подписью.

Цифровая подпись гарантирует, что никакая часть сертификата не может быть незаметно изменена. В случае изменения проверка публичным ключом **CA** даст ошибку.

```
openssl x509 -text -noout -in ca.crt
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      d9:e0:91:e7:d0:27:02:db
  Signature Algorithm: sha256WithRSAEncryption
  Issuer: CN = MY CA
  Validity
    Not Before: Feb  4 14:30:01 2019 GMT
    Not After : Feb  1 14:30:01 2029 GMT
  Subject: CN = MY CA
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
    Modulus:
      00:c8:5e:02:b8:55:e5:42:97:f7:c6:53:61:d3:df:
      66:bf:05:dd:7a:0c:61:a4:68:36:23:3f:3b:c7:83:
      ec:47:9b:5a:ed:78:8a:5b:f1:5f:88:3d:36:f2:3e:
      7b:84:9e:1b:e5:87:bf:3b:00:33:36:1c:0b:3a:16:
      2f:8d:be:0e:a4:9e:25:73:4d:93:8a:47:74:29:65:
      0e:4e:ea:44:fd:c4:c0:bf:fa:bc:11:d5:93:43:e2:
      65:18:bb:f7:e5:fc:16:8c:f9:11:97:76:2c:bb:cb:
      c0:94:7e:78:12:20:c9:8a:68:29:c1:e8:af:7e:d7:
      63:6e:a3:57:79:c9:b3:a8:8c:a3:2d:3e:15:1a:25:
      ea:f1:50:fc:ea:93:8f:14:5f:34:61:07:a9:dc:24:
      b8:11:de:9c:17:13:03:19:0d:0c:a3:e8:10:31:50:
      82:5b:cb:0e:26:d5:b1:fe:df:c3:f6:f9:e4:0f:b1:
      24:40:f2:8d:95:d5:ea:34:b3:27:a1:87:76:9d:f2:
      65:74:d5:40:47:dd:a1:32:46:c3:37:ec:a5:b3:09:
      30:73:99:d1:9c:bb:a8:05:61:2a:56:89:32:5e:c0:
      5d:1d:b6:a6:b6:74:17:be:74:69:9c:b0:e3:bc:b4:
      f9:96:6d:aa:60:ae:70:d1:ee:07:e5:2c:5d:0a:af:
      ce:b3
    Exponent: 65537 (0x10001)
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      53:4B:6D:7A:1A:1F:F8:BE:3F:59:64:70:2B:31:2F:7A:7F:2A:6B:14
    X509v3 Authority Key Identifier:
      keyid:53:4B:6D:7A:1A:1F:F8:BE:3F:59:64:70:2B:31:2F:7A:7F:2A:6B:14

    X509v3 Basic Constraints: critical
      CA:TRUE
  Signature Algorithm: sha256WithRSAEncryption
  30:c2:f3:e0:96:51:7d:13:be:06:1d:40:06:70:b8:36:e9:46:
  81:64:0c:f0:e7:69:6f:31:2c:e1:86:df:f8:ad:b2:84:6e:90:
  4a:38:48:7d:ae:92:a5:71:40:c8:8e:0f:7e:67:e5:66:e7:70:
  4d:52:92:fa:a6:54:45:3b:a6:b9:b3:f4:35:ad:1c:6e:6e:15:
  06:81:ef:13:54:80:89:2e:7d:75:06:22:59:89:44:a9:ad:25:
  30:6c:02:e1:3b:2e:e2:bc:46:90:d1:a5:00:eb:87:57:60:a4:
  cf:e0:03:4a:b5:32:c4:dc:c7:e3:34:d4:c8:af:e3:ce:20:8c:
  c4:7f:f0:b8:72:d7:65:3b:38:be:2b:b1:d0:4e:9e:e2:52:32:
  41:fe:22:d2:7c:13:60:fe:4a:13:4b:c5:09:f0:00:89:32:22:
  47:4d:2c:a1:21:8e:b2:7d:0f:1a:10:f5:94:ee:fb:18:d3:15:
  f1:9e:70:89:73:c4:41:71:0e:92:22:9c:18:ef:0b:b1:7c:42:
  41:e7:9f:e7:82:d5:db:f3:60:d3:2f:2a:86:e4:0c:c0:4c:0c:
  17:12:ec:e4:37:96:dc:2d:01:00:22:ac:b5:33:6b:97:41:d7:
  37:e5:75:fa:c9:6b:00:2a:d8:87:0f:9e:f3:aa:c5:23:4e:60:
  02:a9:5b:eb
```

Видим, что сертификат подписан сам своим-же приватным ключом (Issuer: CN = MY CA, Subject: CN = MY CA) и выписан на 10 лет: начиная с "Feb 4 14:30:01 2019 GMT" (для этого мы указывали `-days 3650` в команде подписи)

Таким образом цепочка доверия (проверки) выстраивается следующим образом:

Сертификат устройства подписан сертификатом CA и может быть им проверен:

```
openssl verify -CAfile ca.crt device_AP6V5MDG.crt
device_AP6V5MDG.crt: OK
```

Ну а сам сертификат подписан "собой":

```
openssl verify -CAfile ca.crt ca.crt
ca.crt: OK
```

Настройка nginx.

Допустим в интернете есть сервер, который должен обрабатывать запросы только от устройств обладающих сертификатами, выписанными нашим CA. Для включения такой проверки в конфигурационном файле nginx необходимо в секции http или server прописать следующие строчки:

```
ssl_client_certificate ca.crt;
ssl_verify_client on;
```

Теперь nginx будет требовать от клиента сертификат, который должен проходить проверку с помощью ca.crt, иначе сервер вернет клиенту ошибку 400:

```
curl https://example.com
<html>
<head><title>400 No required SSL certificate was sent</title></head>
<body bgcolor="white">
<center><h1>400 Bad Request</h1></center>
<center>No required SSL certificate was sent</center>
<hr><center>nginx/1.14.0 (Ubuntu)</center>
</body>
</html>
```

Теперь сделаем так, чтобы HTTP запросы с контроллера WB проходили данную проверку. Для простоты будем использовать nginx и на клиентской стороне. Это даст возможность работать с защищенными серверами клиентам не умеющим делать SSL соединения.

Для начала создадим на неиспользуемом локальном порту http сервер, который будет делать всю https "магию" за нас:

```
server {
    listen 8080;
    location / {
        proxy_pass https://example.com;
        proxy_ssl_name example.com;
        proxy_ssl_server_name on;
        proxy_ssl_certificate device_AP6V5MDG.crt;
        proxy_ssl_certificate_key engine:ateccx08:ATECCx08:00:04:C0:00;
    }
}
```

Добавим пользователя www-data в группу i2c для доступа к криптоустройству:

```
usermod -G www-data
```

Выполним команду **service nginx restart** для обновления конфигурации.

Теперь при обращении по http на локальный порт 8080 зашифрованные запросы с аутентификационной информацией будут отправляться на сервер example.com.

```
curl localhost:8080/
<html>
<body>
EXAMPLE.COM
</body>
</html>
```

Настройка openssl

Для начала установим пакет:

```
apt install openssl
```

Создадим файл req.cnf - он нам потребуется для создания сертификата сервера.

```
[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth
```

Серверу openssl также требуется файл с параметрами DH, сделаем его.

```
openssl dhparam -out dh2048.pem 2048
```

Создание данного файла может потребовать несколько минут.

Далее сделаем приватный ключ нашего сервера и запрос на сертификат. Предполагаем, для примера, что сервер имеет имя example.com:

```
openssl genrsa -out example.key 2048
openssl req -new -key example.key -subj "/CN=example.com" -out example.csr
```

Подписываем запрос в нашем центре сертификации:

```
openssl x509 -req -in example.csr -CA ca.crt -CAkey ca.key -out example.crt -days 365 -CAcreateserial -extfile req.cnf -extensions v3_req
```

Теперь приступим к настройке openvpn сервера на машине example.com.

Копируем файлы **ca.crt**, **example.crt**, **example.key** и **dh2048.pem** и редактируем файл конфигурации openvpn сервера. По умолчанию конфигурационный файл лежит в файле /etc/openvpn/server.conf

```
port 1194
proto tcp
dev tun
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/example.crt
key /etc/openvpn/server/example.key
dh /etc/openvpn/server/dh2048.pem
topology subnet
server 10.8.0.0 255.255.255.0
keepalive 10 60
key-direction 0
cipher AES-128-CBC
auth SHA256
comp-lzo
user nobody
group nogroup
persist-key
persist-tun
status /var/run/openvpn/openvpn-status.log
management 127.0.0.1 7500
log-append /var/log/openvpn.log
```

Добавим пользователя openvpn в группу i2c для доступа к криптоустройству:

```
usermod -G www-data
```

после этого запускаем сервер командой service openvpn start.

Далее на контроллере создаем файл конфигурации клиента: client.ovpn:

```
client
dev tun
proto tcp
remote example.com 1194
resolv-retry infinite
nobind
user openvpn
''group i2c''
persist-key
persist-tun
cipher AES-128-CBC
auth SHA256
key-direction 1
keepalive 1 10
remote-cert-tls server
comp-lzo
```

Обратите внимание на строчку group i2c. Она необходима для работы с криптоустройством.

После этого запускаем клиента:

```
openvpn --config example.ovpn --ca ca.crt --cert device_AP6V5MDG.crt --key engine:ateccx08:ATECCx08:00:04:C0:00
```

Если все хорошо, то в системе должен появиться интерфейс tun0 с адресом из подсети 10.8.0.0/24:

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
inet 10.8.0.2 netmask 255.255.255.0 destination 10.8.0.2
inet6 fe80::53b0:83f3:b1f5:b817 prefixlen 64 scopeid 0x20<link>
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100 (UNSPEC)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 6 bytes 288 (288.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Для проверки работоспособности запускаем ping:

```
ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data:
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=2.23 ms
```

```
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=2.83 ms
```

Настройка mosquitto

UPD:

- 1) openssl.cnf
- 2) mosquitto=1.4.15-1+wb7-4
- 3) libateccssl1.1=0.2.1
- 4) ca-certificates-contactless
- 5) usermod -a -G i2c mosquitto

```
root@wiredboard-APIJVTIG:~# cat /etc/mosquitto/conf.d/bridge-hw.conf
connection wb_devices_cloud.wiredboard-APIJVTIG
address contactless.ru:8884

bridge_cafile /etc/ssl/certs/WiredBoard_Root_CA.pem
bridge_certfile /etc/ssl/device/device_bundle.crt.pem
bridge_keyfile engine:ateccx08:ATECCx08:00:04:C0:00
bridge_capath /etc/ssl/certs/
bridge_insecure true

notifications true
notification_topic /client/wiredboard-APIJVTIG/bridge_status

topic /devices/# both 2 "" /client/wiredboard-APIJVTIG
topic /config/# both 2 "" /client/wiredboard-APIJVTIG
topic /rpc/# both 2 "" /client/wiredboard-APIJVTIG
```

```
root@wiredboard-APIJVTIG:~# cat fix.sh
#!/bin/bash -x
CERT_IN=/etc/ssl/certs/device_bundle.crt.pem
CERT_OUT=/etc/ssl/device/device_bundle.crt.pem
CERT_BKP=/etc/ssl/device/_device_bundle.crt.pem

prn() {
    cat $CERT_IN|grep "$1" -n|sed -n "$2p"|cut -d':' -f1
}

fix() {
    B1=$(prn "BEGIN CERTIFICATE" 1)
    B2=$(prn "BEGIN CERTIFICATE" 2)
    E1=$(prn "END CERTIFICATE" 1)
    E2=$(prn "END CERTIFICATE" 2)

    if [[ "$E1" -le "$B1" || "$E2" -le "$B2" || "$E1" -ge "$B2" ]]; then
        echo "ERROR in device cert bundle."
        exit 1
    fi

    cat $CERT_IN|sed -n "${B2},${E2}p"
    cat $CERT_IN|sed -n "${B1},${E1}p"
}

mkdir -p /etc/ssl/device

if [ -f "$CERT_IN" ]; then
    echo "backup device bundle certificate..."
    cp "$CERT_IN" "$CERT_BKP"
fi

if [ ! -f "$CERT_OUT" ]; then
    if [ ! -f "$CERT_IN" ]; then
        echo "ERROR: no such file: $CERT_IN"
        exit 1
    fi
    fix > "$CERT_OUT"
    echo "Device bundle certificate fix done."
    rm -f "$CERT_IN"
else
    echo "Device cert $CERT_OUT already fixed."
fi
```

Генерируем приватный ключ и запрос на сертификат:

```
openssl genrsa -out example.key 2048
openssl req -new -key example.key -subj "/CN=example.com" -out example.csr
```

Создаем сертификат сервера в CA.

```
openssl x509 -req -in example.csr -CA ca.crt -CAkey ca.key -out example.crt -days 365 -CAcreateserial -extfile req.cnf -extensions v3_req
```

Копируем файлы **ca.crt**, **mosquitto.crt**, **mosquitto.key** на сервер и редактируем файл конфигурации **/etc/mosquitto/conf.d/server.conf**

```
cafile /etc/mosquitto/ssl/ca.crt
certfile /etc/mosquitto/ssl/mosquitto.crt
keyfile /etc/mosquitto/ssl/mosquitto.key
require_certificate true
use_identity_as_username true
```

Запускаем сервис:

```
service mosquitto start
```

```
service mosquitto start
```

Также, если требуется, можно сделать чтобы локальный mosquitto сервер на контроллере форвардил некоторые топики на удаленный сервер. Для этого создаем файл бриджа: **/etc/mosquitto/bridge.conf**

```
connection main
topic test/# out
address example.com:1883

bridge_cafile      /etc/mosquitto/certs/ca.crt
bridge_certfile   /etc/mosquitto/certs/device_AP6V5MDG.crt
bridge_keyfile    engine:ateccx08:ATECCx08:00:04:C0:00
```

После перезапуска локального сервиса mosquitto топики /test/.. будут отправляться на удаленный сервер example.com по защищенному ssl каналу.

Примеры клиентских команд mosquitto. Отправка сообщения "message" в топик "test" на сервере example.com

```
mosquitto_pub -h example.com --cert device_AP6V5MDG.crt --key 'engine:ateccx08:ATECCx08:00:04:C0:00' --cafile ca.crt -t "test" -m "message"
```

Получение сообщений из топика "test" на сервере example.com

```
mosquitto_sub -h example.com --cert device_AP6V5MDG.crt --key 'engine:ateccx08:ATECCx08:00:04:C0:00' -t "test" --cafile ca.crt
```

- English
- русский

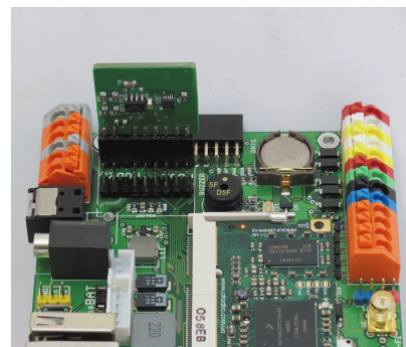
Модули расширения - небольшие электронные платы, устанавливаемые внутрь корпуса Wiren Board 5, и расширяющие его функциональность.

При заказе модулей расширения в комплекте с контроллером Wiren Board 5, модули расширения устанавливаются производителем.

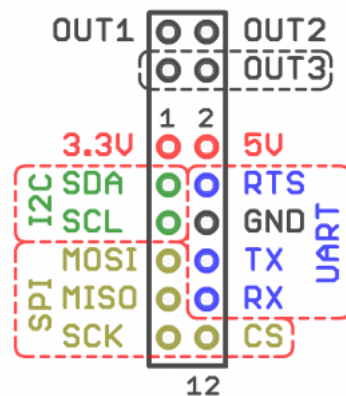
Если модули приобретены отдельно, и вы устанавливаете их самостоятельно, вам нужно:

1. Снять заднюю крышку корпуса.
2. Извлечь печатную плату.
3. Вставить модуль, соблюдая полярность.

Смотрите соответствующие статьи для описания подключения и работы в ПО:



Установка модуля расширения



Распиновка разъема под модули расширения

| Модули расширения интерфейсов | | | | |
|---|--|-------------------------|--------------------|----------|
| Артикул | Описание | Интерфейс | | Изоляция |
| WBE-I-RS232 | Модуль расширения RS-232 | RS-232 | | нет |
| WBE-I-CAN-ISO | Модуль расширения CAN (Изолированный) | CAN, UART-CAN | | 1.5 kV |
| WBE-I-RS485-ISO | Модуль расширения RS-485 (Изолированный) | RS-485 | | 1.5 kV |
| WBE-I-1WIRE | Модуль расширения 1-Wire | 1-Wire | | нет |
| WBE-I-KNX | Модуль расширения KNX (WBE-I-KNX) | KNX TP-UART (EIB) | | 5kV |
| Артикул | Описание | Тип входов | Количество входов | Изоляция |
| WBE-DI-DR-3 | Модуль входов "сухой контакт" | сухой контакт | 3 | нет |
| Модули расширения портов дискретных выходов | | | | |
| Артикул | Описание | Тип выходов | Количество выходов | Изоляция |
| WBE-DO-R6C-1 | Модуль релейных выходов WBE-DO-R6C-1 | механическое реле | 1 | 3 kV |
| WBE-DO-SSR-2 | Модуль выходов "сухой контакт" (оптореле) WBE-DO-SSR-2 | оптореле | 2 | нет |
| WBE-DO-OC-2 | Модуль выходов "Открытый коллектор" WBE-DO-OC-2 | открытый коллектор | 2 | нет |
| Модули расширения аналогового ввода-вывода | | | | |
| Артикул | Описание | Тип | Количество каналов | |
| WBE-AO-10V-2 | Модуль расширения ЦАП | аналоговые выходы 0-10В | 2 | |
| Модули расширения прочие | | | | |
| Артикул | Описание | | | |
| WBE-MICROSD | Модуль расширения microSD | | | |
| WBE-GPS | Модуль расширения GPS/Glonass | | | |

Модули ввода-вывода

Модули ввода-вывода стыкуются к контроллеру Wiren Board или преобразователю интерфейсов WB-MIO справа, через боковой разъём. Следите за попаданием всех штырей модуля в отверстия ответного разъёма. Зафиксируйте на DIN-рейке упорами (ограничителями) с боков.



Подключение модуля к контроллеру

Последовательно можно подключать до 8 штук: до 4 модулей ввода (типа «I») и до 4-х модулей вывода (тип «O»). Исключение WBIO-AI-DV-12 — этот модуль можно подключить только один.

Адреса раздаются последовательно. Подключать до 4 модулей можно в любой последовательности, а при большем числе следует подключать сначала один тип, потом другой.

| Артикул | Описание | Количество каналов | Тип каналов | Ширина, DIN юнитов | Тип |
|------------------------------|--|--------------------|---|--------------------|-----|
| Модули входов | | | | | |
| WBIO-DI-WD-14 | Универсальный модуль дискретных входов | 14 | Входы «сухой контакт» наличия напряжения 12/24В | 2U | I |
| WBIO-DI-HVD-8 | Модуль-детектор наличия сетевого напряжения (230В) | 8 | Входы напряжения 50-250В AC | 2U | I |
| WBIO-DI-HVD-16 | Модуль-детектор наличия сетевого напряжения (230В) | 16 | Входы напряжения 50-250В AC | 3U | I |
| WBIO-AI-DV-12 | Модуль аналоговых входов | 12 (6 дифф.) | Аналоговые входы 0-3В, -50..+50V | 3U | I |
| WBIO-AI-DV-12/4-20MA | Модуль аналоговых входов 4-20mA | 12 | Аналоговые входы 4-20mA | 3U | I |
| Модули выходов | | | | | |
| WBIO-DO-R10A-8 | Модуль релейных выходов 7A | 8 | Механические реле, SPST | 3U | O |
| WBIO-DO-R10R-4 | Модуль релейных выходов 3A для управления роллетами | 4 | Механические реле, SPCO | 3U | O |
| WBIO-DO-R1G-16 | Модуль релейных выходов 1A для контакторов | 16 | Механические реле, SPST | 3U | O |
| WBIO-DO-HS-8 | Модуль дискретных выходов High Side Switch | 8 | Выходы напряжения | 2U | O |
| WBIO-DO-SSR-8 | Модуль дискретных выходов с твёрдотельными реле на 30В | 8 | Оптореле, SPST, до 30В | 2U | O |
| WBIO-AO-10V-8 | Модуль аналоговых выходов WBIO-AO-10V-8 | 8 | Аналоговые выходы 0...10 В | 2U | O |
| Снятые с производства | | | | | |
| WBIO-DI-DR-8 | Модуль дискретных входов типа «сухой контакт» | 8 | Входы "сухой контакт" | 2U | I |
| WBIO-DI-DR-16 | Модуль дискретных входов типа «сухой контакт» | 16 | Входы "сухой контакт" | 3U | I |
| WBIO-DI-DR-14 | Модуль дискретных входов типа «сухой контакт» | 14 | Входы "сухой контакт" | 2U | I |
| WBIO-DI-LVD-8 | Модуль-детектор наличия напряжения 12/24В | 8 | Входы напряжения 9-50В AC/DC | 2U | I |
| WBIO-DI-LVD-16 | Модуль-детектор наличия напряжения 12/24В | 16 | Входы напряжения 9-50В AC/DC | 3U | I |
| WBIO-DO-R3A-8 | Модуль релейных выходов 3A | 8 | Механические реле, SPST и SPDT | 3U | O |
| WBIO-DIO-TTL-8 | Модуль ввода-вывода с TTL-уровнями | | 5V TTL GPIO | 2U | O |

Конфигурирование

После физического подключения модуля его нужно добавить в конфигурацию контроллера:

1. В веб-интерфейсе перейдите в раздел **Settings** → **Configs**.
2. Зайдите в раздел **Hardware Modules Configuration**, выберите из **External I/O module** тот, куда установлен модуль расширения.
3. В раскрывающемся списке **Module type** выберите тип установленного модуля.
4. Нажмите кнопку **Save**.

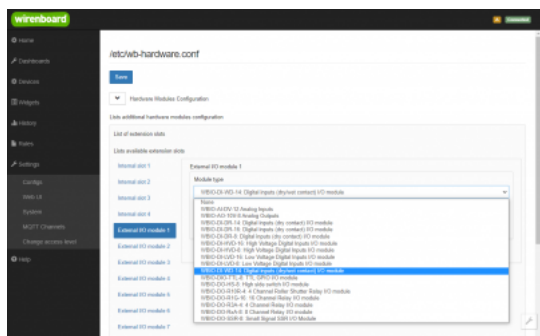
Для удаления модуля выберите тип **None**.

Убедитесь, что внешний модуль виден в веб-интерфейсе:

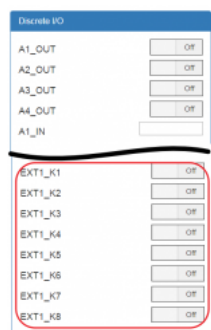
1. Перейдите в раздел **Devices**.
2. Найдите устройство **Discrete I/O** для дискретных и **ADCs** для аналоговых модулей. Каналы будут иметь вид EXTN_YYYY,

где n — порядковый номер модуля, а rrr — название канала модуля.

Адрес MQTT-топика будет иметь вид `/devices/wb-gpio/controls/EXTN_YYYY`.



Настройка бокового нового модуля



Каналы внешнего модуля в веб-интерфейсе

Распиновка разъёма



Распиновка разъёма «мама» для модулей ввода-вывода (на контроллере и модулях ввода-вывода)

- English
- русский

Дополнительная мезонинная плата, устанавливается вторым этажом в корпус на DIN-рейку.

Совместима с контроллером Wiren Board 5. Подключается шлейфом к внутреннему разъему контроллера.

Плата содержит схему заряда Li-Ion аккумулятора с защитой от заряда при низких и высоких температурах.

Отсутствует защита от переразряда подключенного аккумулятора - обязательно использовать аккумуляторы с встроенной схемой защиты.

Платы комплектуются аккумуляторами 1800 ма*ч.

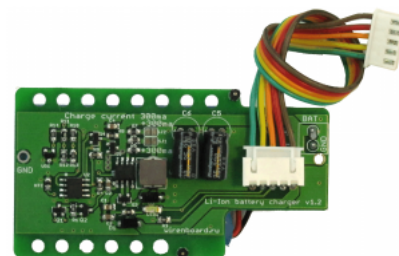
На контроллер идет два сигнала:

- STAT1 (active low) - плата подключена.
- STAT2 (active low) - идет заряд аккумулятора.

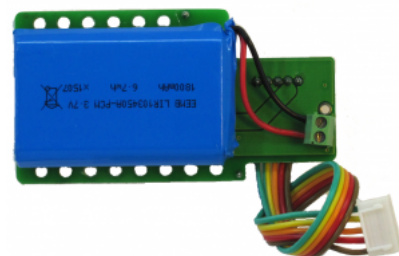
Зарядный ток устанавливается запаиваемыми перемычками:

- не запаяны - 300 ма.
- одна запаянна - 600 ма.
- две - 900 ма.

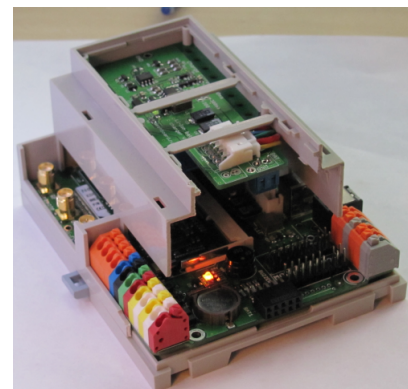
Стандартные модификации:



Модуль резервного питания. Вид сверху



Модуль резервного питания с аккумулятором LIR103450A



Установка модуля вторым этажом

| Артикул | Комплектация | Ёмкость | Вес |
|----------------------|--|----------|------|
| WBMZ-BATTERY-1800MAH | Модуль с аккумулятором LIR103450A-PCB-LD | 1800 mAh | 50 г |

Технические характеристики аккумуляторов LIR103450A

| | |
|------------------------|--------|
| Тип | Li-ion |
| Номинальное напряжение | 3,7 В |

| | |
|---------------------------------|--------------------------|
| Номинальная ёмкость (С) | 1800 мАч |
| Максимальный ток заряда/разряда | 1 С |
| Время заряда | 0,5 С — 5 часов |
| | 1 С — 2,5 часа; |
| Напряжение заряда | 4,2 В |
| Температура эксплуатации | Заряд: от 0°C до 45°C |
| | Разряд: от -20°C до 55°C |
| Температура хранения | от -5°C до 35°C |
| Циклов заряд/разряд | 300 |

Подключение модуля, идущего в комплекте с контроллером

Если аккумуляторный модуль поставляется в комплекте с контроллером, то его подключение потребует минимальных усилий. Контроллеры с аккумуляторным модулем внутри комплектуются крышкой с выключателем. Сам модуль отсоединен от контроллера во избежание разряда аккумулятора. Для подключения модуля крышка с выключателем аккуратно извлекается отверткой, вставленной в боковые пазы между крышкой и верхней частью корпуса. Выключатель должен быть подсоединен к разъему на плате контроллера. Далее необходимо подключить к модулю шлейф, соединяющий его с платой, как это показано на рисунках.



Выключатель на верхней крышке контроллера



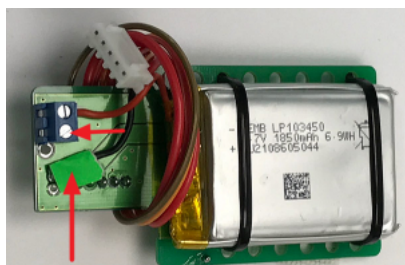
Изначально разъем шлейфа не подключен к модулю

Вставляем вилку шлейфа

И продвигаем внутрь разъема до упора

Установите верхнюю крышку контроллера на место. Контроллер готов к работе.

Подключение модуля, приобретенного отдельно



Аккумуляторный модуль, один контакт батареи отсоединен

Разбираем корпус контроллера по инструкции.

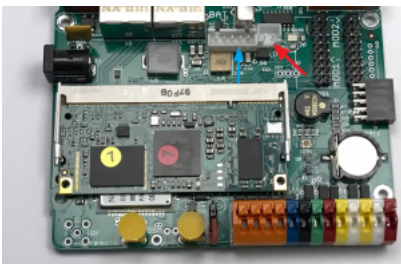
Аккумуляторный модуль WBMZ поставляется в комплекте с верхней крышкой корпуса, на которой смонтирован выключатель. Крышка устанавливается взамен идущей в комплекте контроллера. Старая крышка извлекается отверткой, вставленной в боковые пазы между крышкой и верхней частью корпуса.

Один контакт аккумуляторной батареи отсоединен от модуля для предотвращения разряда при хранении. Перед установкой необходимо закрепить отключенный контакт в винтовом зажиме модуля (показано на рисунке стрелками)



Верхняя крышка корпуса контроллера с выключателем

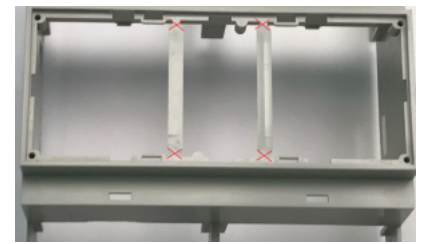




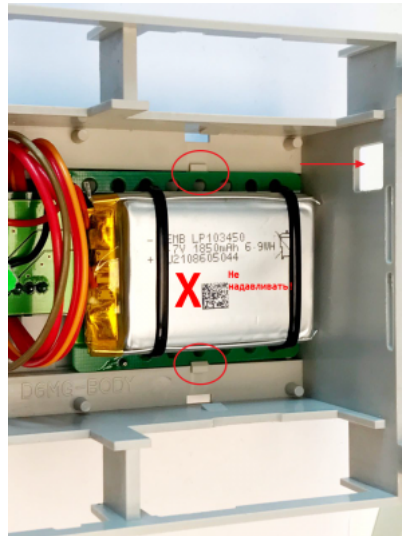
Плата контроллера с разъемами для аккумулятора и выключателя

стрелками.

Для удобства установки модуля перемычки в верхней части корпуса можно удалить (например, кусачками).



Перемычки под верхней крышкой можно удалить



Крепление модуля

Плата модуля крепится на двух защелках (обозначены на рисунке), аккумулятором вниз, в сторону платы. Аккумулятор должен быть расположен со стороны отверстия для бокового модуля питания (показано стрелкой). Не надавливайте на аккумулятор при установке — это может привести к внутреннему короткому замыканию и возгоранию!

После установки модуля подключите разъемы к плате, как показано стрелками на рисунке "Плата контроллера с разъемами для аккумулятора и выключателя": красная стрелка — разъем выключателя, голубая — разъем модуля.

При сборке контроллера с установленным модулем обратите внимание на провода модуля: они не должны находиться между индикатором и пластиковым световодом верхней крышки корпуса контроллера. Снятие модуля производится в обратной последовательности.

Радиоинтерфейс 433 МГц

- English
- русский

В настоящее время модуль WBE2S-R-433MHZ снят с производства, его поддержки нет и работоспособность в актуальных релизах ПО не гарантируется.

Контроллеры Wiren Board опционально комплектовались радиоинтерфейсом 433МГц на базе чипа RFM69. С его помощью можно принимать и отправлять сигналы пультов радиопередачи различными бытовыми устройствами. В базовой комплектации поддерживаются климатические датчики Oregon Scientific (протоколы версий 2 и 3) (см. таблицу совместимых устройств), а также устройства nooLite.

В бета-версии находится более продвинутый драйвер радиокommunikации `rfsniffer`, обеспечивающий поддержку более широкого спектра радиоуправляемых устройств.

Contents

Начало работы

Системный сервис

Устройства

Представление в Web-интерфейсе

Пример подключения

Обновление драйвера до `wb-homa-rfsniffer`

Настройки параметров `rfsniffer`

Группа параметров настройки драйвера — **Radio settings**

Группа параметров настройки **MQTT settings**

Группа параметров настройки **Device politics**

Группа параметров настройки **DEBUG settings**

Группа параметров настройки **Enabled features**

Начало работы

Поддержка радиоинтерфейса 433 МГц — опция, которую вы можете выбрать при покупке контроллера Wiren Board. На плате контроллера с поддержкой радиоинтерфейса распаян материнский разъем SMA (Sub-Miniature Version A), обозначенный "RF remote", на который накручивается ответный разъем антенны 433 МГц. При монтаже контроллера в металлическом щите антенну следует располагать за его пределами для обеспечения надежной радиосвязи.



Подключение антенны

Системный сервис

| ISM Radio | |
|-----------|---------------------------------|
| rx raw | <input type="text" value="32"/> |
| rx noo | <input type="text" value="5"/> |
| rx oregon | <input type="text" value="12"/> |

Параметры сервиса `wb-homa-ism-radio`

За поддержку радиоинтерфейса отвечает служба `wb-homa-ism-radio`. Она обслуживает драйвер радиомодуля, создает новые устройства, обновляет их состояния и передает команды управления. Параметр **rx raw** обозначает количество принятых радиопакетов от любых устройств, **rx noo** — пакетов от устройств Noolite, **rx oregon** — от датчиков Oregon Scientific. При наличии заведомо работоспособных устройств, функционирующих на частоте 433 МГц в зоне приема контроллера, эти параметры увеличиваются со временем. Например, нажатие кнопок на брелоке автосигнализации должно увеличить счетчик **rx raw**. При перезагрузке значения счетчиков обнуляются. Для распознанного оборудования в веб-интерфейсе создаются виртуальные устройства.

Устройства

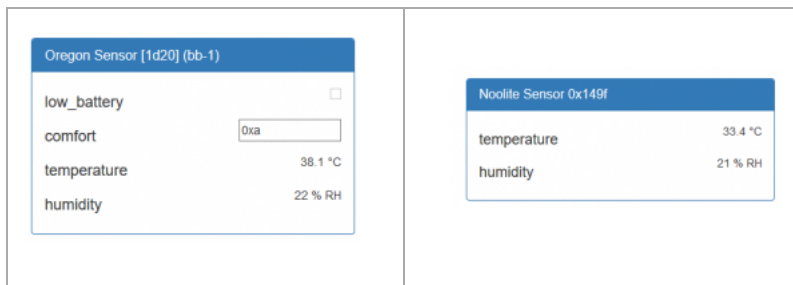
Радиомодуль контроллера Wiren Board поддерживает устройства различного типа: датчики температуры и влажности, релейные модули, пульты дистанционного управления, радиовыключатели и т.п. Для каждого поддерживаемого устройства в Web-интерфейсе контроллера создается соответствующее виртуальное устройство, а в очереди сообщений `mqtt` публикуются

его параметры.

Представление в Web-интерфейсе

При обнаружении нового устройства создаются новые устройства в веб-интерфейсе, а также новые mqtt-топики, описывающие состояние устройства.

Пример использования: для привязки релейного модуля NooLite к контроллеру выполните стандартную процедуру привязки релейного модуля к пульту, но вместо кнопки пульта нажмите кнопку **bind** в веб-интерфейсе. После этого вы сможете отправлять команды на включение и выключение реле.



Пример подключения

Рассмотрим подключение радио-устройства на примере модуля "Силовой блок NooLite SLF-1-300" (Модели с шифрованием и обратной связью (NooLite-F) несовместимы с контроллером, выбирайте те, которые поддерживают просто NooLite, без "-F"). Есть модели, которые поддерживают и тот, и другой протоколы связи).

Подключаем модуль в сеть и к нагрузке, нажимаем кнопку привязки, после этого заморгает светодиод на модуле. Затем на любом из пультов (для примера — на 0xb61) нажимаем кнопку ****Bind****, светодиод заморгает часто, значит, модуль готов привязаться. Еще раз нажимаем кнопку, светодиод станет мигать относительно редко — значит, модуль привязан к виртуальному пульту.

В случае модуля SLF-1-300 управлять им можно с виртуального пульта переключателем state (включать или выключать) и кнопкой switch (изменять состояние включено-выключено-включено-...):

В MQTT этим органам управления соответствуют топики `/devices/noolite_tx_0xb61/controls/state` и `/devices/noolite_tx_0xb61/controls/switch`

Попробуем в командной строке управлять нагрузкой: команда `mosquitto_pub -t "/devices/noolite_tx_0xb61/controls/state/on" -m 1` включит нагрузку, а команда `mosquitto_pub -t "/devices/noolite_tx_0xb61/controls/state/on" -m 0` — выключит. Команда `mosquitto_pub -t "/devices/noolite_tx_0xb61/controls/switch/on" -m 0` будет изменять состояние (в топик можно публиковать любое значение)

Поскольку обратной связи нет, переключатель "state" в веб-интерфейсе не отражает текущее состояние реле!

В движке правил управлять реле можно следующим образом:

```
//включить реле
dev["noolite_tx_0xb61"]["state"]=true;

//выключить реле
dev["noolite_tx_0xb61"]["state"]=false;

//изменить состояние реле (= true или =false)
dev["noolite_tx_0xb61"]["switch"]=true;
```

Обновление драйвера до wb-homa-rfsniffer

Новая версия wb-homa-rfsniffer драйвера 433 МГц охватывает более широкий спектр устройств, функционирующих на частоте 443 МГц, но находится в стадии разработки. Для установки новой версии драйвера требуется удалить wb-homa-ism-radio совместная работа этих драйверов не поддерживается. В настоящий момент rfsniffer НЕ включает в себя полный функционал ism-radio: основное его назначение — получение данных с датчиков (управления из веб-интерфейса нет). В настоящий момент поддерживаются протоколы:

- Oregon (V2 + V3)
- NooLite
- Livolo
- Raex
- RST
- Rubitek
- X10
- VHome
- EV1527

Удаление wb-homa-ism-radio выполняется командой `apt-get remove wb-homa-ism-radio`. Установка новой версии — командой `apt-get update && apt-get install wb-homa-rfsniffer`. Убедитесь, что новый драйвер запущен: `service wb-homa-rfsniffer status`.

На боковой панели Web-интерфейса в разделе Configs появится доступ (пункт **RFM69 Driver Configuration**) к настройкам драйвера, хранящимся в файле `/etc/wb-homa-rfsniffer.conf`.

Настройки параметров rfsniffer

Группа параметров настройки драйвера — Radio settings

- **lirc_device** — специальный файл символического устройства; создается для работы с битовым потоком данных (прием/передача) RFM69; значение по умолчанию — `/dev/lirc0`. Не требует изменений.
- **rfm_irq** — номер прерывания от чипа RFM69. Не требует изменений.
- **rssI** — максимальный уровень чувствительности приемника в dBi (чем больше по модулю отрицательное значение, тем выше чувствительность, максимум: -120). С помощью этого параметра можно ограничить прием удаленных слабых сигналов от ненужного радиоборудования.
- **spi_device** — специальный файл устройства для коммуникации по SPI-шине микропроцессора контроллера. Зависит от аппаратной модели контроллера. В Wiren Board 5 используется `/dev/spidev32766.0` — шина 32766, `chipselect` — 0.

Группа параметров настройки MQTT settings

Параметр **host** определяет адрес mqtt-брокера, куда драйвер будет публиковать сообщения от радиоустройств. По умолчанию — `localhost`. Предполагается, что брокер принимает соединения на порту 1883.

Группа параметров настройки Device politics

Параметры этой группы определяют, каким образом будут обнаруживаться и использоваться новые устройства.

- Список **Known devices** определяет, с какими устройствами будет работать драйвер, когда параметр **use_devices_list** установлен в значение `true`. Устройство имеет имя **name**, политику отображения **politics** и интервал времени **heartbeat**, по прошествии которого устройство считается неработоспособным, если от него не поступало сигналов. При значении этого параметра равного **-1**, устройство считается работоспособным всегда.
- **unknown_devices_politics** — отображать ли ранее неизвестные радиоустройства
- **use_devices_list** — отображать ли устройства только из разрешенного списка

Группа параметров настройки DEBUG settings

Данная группа отвечает за ведение логов отладочной информации.

- **console_level** — уровень отладочной информации в консоли
- **file_level** — уровень отладочной информации в журнальном файле
- **file_name** — путь к журнальному файлу драйвера (`/run/wb-homa-rfsniffer.log`)
- **log_time** — включать ли в отладочную информацию временные метки
- **name** — имя журнала

Возможно создание нескольких журнальных файлов с разными параметрами.

| RF Remote controls | |
|--------------------|------------------|
| Rubitek | 0101000100010001 |
| Livolo | 0 |
| Raex | 0 |

| EV1527 20801 | |
|--------------|-------------------------------------|
| cmd_flip | <input checked="" type="checkbox"/> |
| cmd | 5.00 |

Radio settings [JSON] [Properties]

lirc_device
/dev/lirc0
Address of lirc device

rfm_irq
38
IRQ

rssI
-90
Received signal strength indicator(RSSI)

spi_device
/dev/spidev32766.0
Address of SPI device

MQTT settings [JSON] [Properties]

host
localhost
MQTT host

Device politics [JSON] [Properties]

Known devices [v]

Device 1 [x] Device [JSON] [Properties]

name
noolite_rx_0x149f
Name as in mqtt tree

politics
show
What is needed to do with devices that are not mentioned in the list?

heartbeat
7300
Heartbeat (time in seconds which is needed to admit that device is dead. -1 for dis

unknown_devices_politics
show
What is needed to do with devices that are not mentioned in the list?

use_devices_list
false
Use list of known devices

- **save_path** — путь к директории, в которой сохраняются файлы дампа радиоконанд и отладочной информации

Группа параметров настройки **Enabled features**

Эта группа параметров используется для поддержки передачи данных на устройства NooLite. В списке **Adresses of transferring devices** можно указывать адреса устройств-передатчиков.

Список **Enabled protocols** позволяет выбрать протоколы тех поддерживаемых устройств, которые планируется подключать к контроллеру.

Питание USB-портов

- English
- русский

пакет hubpower из нашего репозитория.

Работа:

```
root@wirenboard:~# hubpower 1:2 status
Port 1 status: 0503 High-Speed Power-On Enabled Connected
Port 2 status: 0100 Power-On
Port 3 status: 0100 Power-On
Port 4 status: 0100 Power-On
Port 5 status: 0503 High-Speed Power-On Enabled Connected
```

Включение/выключение порта:

```
root@wirenboard:~# hubpower 1:2 power 4 off
Port 4 status: 0000 Power-Off
root@wirenboard:~# hubpower 1:2 power 4 on
Port 4 status: 0100 Power-On
```

Contents

Wiren Board 6

Wiren Board 5

Wiren Board 4

Wiren Board Smart Home rev. 3.5

Wiren Board 6

Второй внешний USB-порт:

```
# выключить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 4 off
# включить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 4 on
```

Модуль Wi-Fi:

```
# выключить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 1 off
# включить
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` power 1 on
```

Вместо

```
$ hubpower `lsusb | grep "0424:2514" | sed 's/^Bus 0*\([[[:digit:]]*\) Device 0*\([[[:digit:]]*\).*$/\1:\2/g'` ...
```

в большинстве случаев можно писать

```
$ hubpower 2:2 ...
```

Wiren Board 5

Внешний USB-порт:

```
# выключить
$ hubpower 1:1 power 1 off
# включить
$ hubpower 1:1 power 1 on
```

Модуль Wi-Fi:

```
# выключить
$ hubpower 2:1 power 1 off
# включить
$ hubpower 2:1 power 1 on
```

Wiren Board 4

Номера портов (для управления питанием):

4 - встроенный WiFi
3 - порты USB-Hub, выход 5V
1 - Ethernet-часть LAN9514

Пример (отключение Wi-Fi):

```
hubpower 1:2 power 4 off
```

Wiren Board Smart Home rev. 3.5

Номера портов (для управления питанием):

4 - встроенный WiFi
3 - порты USB-Hub
1 - Ethernet-часть LAN9514

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

Драйвер wb-mqtt-serial

Contents

Описание

Общая информация
Особенности

Управление драйвером

Диагностика неполадок

Включение отладки

Полезные ссылки

Описание

Общая информация

wb-mqtt-serial — драйвер master-slave протоколов для устройств, подключённых:

- к шине RS-485 — протокол Modbus RTU и других;
- через Ethernet — протоколы Modbus TCP, Modbus over TCP и другие;
- к разъёмам MOD1–MOD3 — при наличии модулей расширения, использующих обмен по UART.

Драйвер опрашивает serial-устройства и публикует данные в топике MQTT-брокера. Устройства настраиваются через веб-интерфейс.

wb-mqtt-serial использует систему JSON-шаблонов, которые описывают подключённые устройства: тип протокола, номера регистров, название параметров и контролов в веб-интерфейсе контроллера. В стандартной поставке есть шаблоны для всех устройств Wigen Board, а также некоторых сторонних устройств: счётчики электроэнергии, частотные преобразователи, холодильные контроллеры и другие.

Если ваше устройство работает по поддерживаемому драйвером протоколу, но в стандартной поставке под него нет шаблона — можете написать шаблон сами.

Файлы и папки:

- `/etc/wb-mqtt-serial.conf` — файл настроек драйвера, редактировать вручную не рекомендуем;
- `/usr/share/wb-mqtt-serial/templates` — папка с предустановленными шаблонами;
- `/etc/wb-mqtt-serial.conf.d/templates` — папка для пользовательских шаблонов, которые имеют приоритет на предустановленными.

Если вы добавили свой шаблон или изменили существующий, подождите 20 секунд, а потом перезагрузите страницу конфигуратора в веб-интерфейсе клавишами `Ctrl+F5`.

О том, как получить доступ к файлам и папкам, читайте в статье [Просмотр файлов контроллера с компьютера](#).

Полное описание драйвера, список поддерживаемых протоколов и примеры шаблонов, смотрите **в репозитории на Github** (<https://github.com/contactless/wb-mqtt-serial>).

Особенности

При работе с Modbus-устройствами, драйвер оптимизирует запросы к устройствам: считывает несколько регистров подряд, не выдерживает некоторые задержки, рекомендованные стандартом.

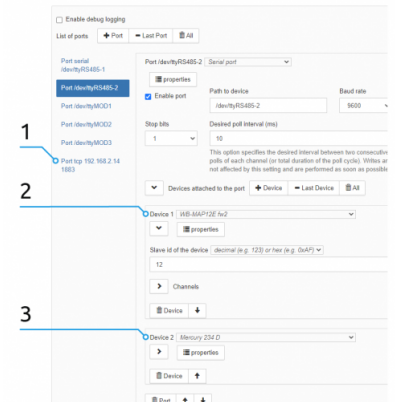
Поэтому при написании шаблона для сторонних Modbus-устройств, нужно указать параметр `guard_interval_us`, который рассчитывается по формуле:

```
guard_interval_us = (3.5*11*106)/(скорость в бит/с)
```

Так же этот параметр можно установить через веб-интерфейс:

- Guard interval (us) — для порта.
- Additional delay before each writing to port (us) — для устройства.

Если при чтении регистра устройства возникла ошибка, то соответствующий контрол в веб-интерфейсе будет окрашен в красный цвет. Аналогично с устройством — если оно давно не отвечает, то все его контролы будут окрашены красным.



Драйвер *wb-mqtt-serial* может одновременно опрашивать устройства, работающие по разным протоколам: 1 — виртуальный порт для устройств с протоколом Modbus TCP, 2 — устройство работает по протоколу Modbus RTU, 3 — устройство работает по протоколу DLMS

Управление драйвером

Обычно драйвер запускается автоматически при загрузке контроллера и перезапускается при сохранении файла конфигурации в веб-интерфейсе.

Также можно управлять драйвером в ручном режиме — это может быть полезно для поиска ошибок в конфигурационном файле или если вам нужно освободить порт для использования `modbus_client`.

Для выполнения команд подключитесь к контроллеру по SSH. Доступны команды:

```
systemctl stop wb-mqtt-serial # остановить
systemctl start wb-mqtt-serial # запустить
systemctl restart wb-mqtt-serial # перезапустить
wb-mqtt-serial -c /etc/wb-mqtt-serial.conf -d # запустить в отладочном режиме с указанием пути к конфигурационному файлу
```

Диагностика неполадок

Если возникли проблемы с запуском драйвера, например, новое устройство не появилось, то можно узнать причину: выполните команду `systemctl status wb-mqtt-serial` и в последних двух строчках ответа будет подсказка.

В примере файл конфигурации содержит синтаксическую ошибку во второй строке на 14 позиции:

```
# systemctl status wb-mqtt-serial
● wb-mqtt-serial.service - MQTT Driver for serial devices
   Loaded: loaded (/lib/systemd/system/wb-mqtt-serial.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2021-01-28 15:10:51 +04; 4s ago
   Process: 23682 ExecStart=/usr/bin/wb-mqtt-serial (code=exited, status=0/SUCCESS)
   Main PID: 23682 (code=exited, status=0/SUCCESS)

Jan 28 15:10:47 wirenboard-A6XXXT2R systemd[1]: Started MQTT Driver for serial devices.
Jan 28 15:10:51 wirenboard-A6XXXT2R wb-mqtt-serial[23682]: ERROR: [serial] Failed to parse JSON /etc/wb-mqtt-serial.conf:* Line 2, Column 14
Jan 28 15:10:51 wirenboard-A6XXXT2R wb-mqtt-serial[23682]: Syntax error: value, object or array expected.
```

Проверить только шаблоны, в том числе и не подключённые в файле конфигурации, можно командой:

```
# wb-mqtt-serial -g
<3>ERROR: [serial config] Failed to parse /usr/share/wb-mqtt-serial/templates/config-bac-6000-series.json
Failed to parse JSON /usr/share/wb-mqtt-serial/templates/config-bac-6000-series.json:* Line 12, Column 5
Missing ', ' or '}' in object declaration
```

Проверить файл конфигурации и шаблоны на ошибки:

```
# wb-mqtt-serial -j
<3>ERROR: [serial config] Failed to parse /usr/share/wb-mqtt-serial/templates/config-wb-mdm3.json
Failed to parse JSON /usr/share/wb-mqtt-serial/templates/config-wb-mdm3.json:* Line 8, Column 9
Missing ', ' or '}' in object declaration

<3>ERROR: [serial] Can't find template for 'WB-MDM3'
```

При необходимости, можно добавить путь к файлу, который нужно проверить:

```
wb-mqtt-serial -c /etc/wb-mqtt-serial.conf -j
```

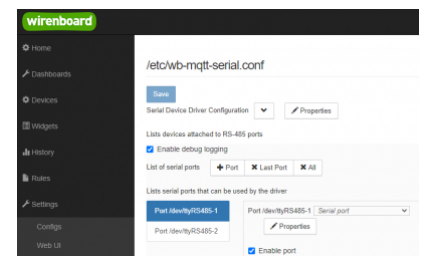
Включение отладки

Иногда нужно включить отладочный режим драйвера. Это можно сделать из командной строки или через веб-интерфейс.

При включённой отладке размер системного журнала будет быстро расти, поэтому не забудьте отключить отладку, когда необходимость в ней отпадет.

Включение отладки через веб-интерфейс:

1. Зайдите в веб-интерфейс контроллера
2. Если вы работаете под обычным пользователем, то смените уровень доступа
3. Перейдите **Settings** → **Configs** → **Serial Device Driver Configuration**
4. Установите флажок **Enable debug logging**
5. Нажмите на кнопку **Save**, чтобы сохранить настройки.



Веб-интерфейс. Флажок *Enable debug logging* установлен, отладка включена

Теперь в системный журнал будут записываться отправленные и принятые драйвером пакеты.

Чтобы посмотреть `debug`-вывод драйвера, выполните в консоли контроллера команду `journalctl -e -p 7`, где `-e` — отобразить последние записи, а `-p 7` задает уровень сообщений, где 7 — это `debug`. Подробнее о параметрах утилиты, читайте в статье `journalctl`.

Пример вывода команды:

```
~# journalctl -e -p 7
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [serial port driver] channel 'Urms L1' of device 'wb-modbus-0-0' <-- 224.647
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [modbus] read 2 input(s) @ 5136 of device modbus:142
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [port] Sleep 0 us
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [port] Write: 8e 04 14 10 00 02 6a c1
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [port] Sleep 10000 us
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [port] ReadFrame: 16 04 02 0f 3d 09 12
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [register handler] new val for input @ 3 of device modbus:22: f3d
Jan 29 14:27:24 wirenboard-A6ZZXT2R wb-mqtt-serial[1667]: DEBUG: [serial port driver] register value change: input @ 3 of device modbus:22 <- 39.01
```

Полезные ссылки

- Как подключить стороннее Modbus-устройство
- Описание wb-mqtt-serial на Github (<https://github.com/contactless/wb-mqtt-serial>)
- Описание протокола Modbus
- Описание шины RS-485

CAN

Работа происходит через стандартную подсистему Linux — SocketCAN. Порты CAN доступны в системе как сетевые интерфейсы `can0` или `can1`, в зависимости от модели контроллера. Для работы CAN на контроллерах до версии 6.7.x нужен джампер терминатора на порту. Начиная с версий 6.7.x, терминатор управляется программно и включается автоматически.

Contents

Настройка

- Через веб-интерфейс
- Стандартными средствами linux (автоматически)
- Стандартными средствами linux (вручную)

Работа с CAN

Настройка

Через веб-интерфейс

Для настройки через веб-интерфейс нужно обновить пакет `wb-mqtt-confed` до версии 1.2.3+

Действия происходят в разделе **Configs** веб-интерфейса:

- На вкладке **Network Interfaces** добавьте новый интерфейс `can0` (см. скриншот). Нажмите кнопку **Save**.
- Переключите порт RS-485/CAN в режим CAN: на вкладке **Hardware Modules Configuration** выберите в поле **Module type** «CAN interface» и нажмите кнопку **Save**.

CAN-интерфейс будет подниматься сам при:

- загрузке системы — за это отвечает сервис `wb-hwconf-manager`,
- смене режима порта RS-485/2 в «CAN».

Стандартными средствами linux (автоматически)

Раздел **Network Interfaces** веб-интерфейса — это обёртка вокруг файла `/etc/network/interfaces`, поэтому настройку CAN можно произвести с помощью службы управления сетями ОС `debian`. Для этого нужно:

- Добавить в `/etc/network/interfaces` запись вида:

```
allow-hotplug can0
iface can0 can static
bitrate 125000
```

- Переключить порт RS-485-2 в режим CAN

Получим результат, аналогичный настройке через веб-интерфейс.

Стандартными средствами linux (вручную)

После переключения порта в режим CAN, нужно выполнить:

```
ip link set can0 up type can bitrate 125000
```

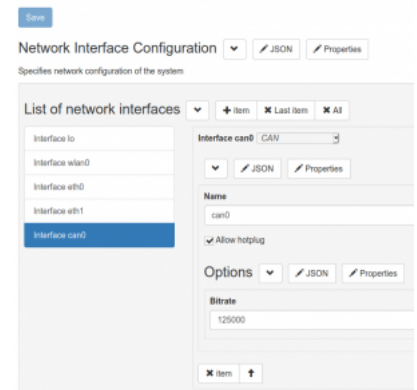
Работа с CAN

Утилиты `can send` и `can dump` есть в стандартном наборе ПО контроллера. Если по каким-то причинам их нет, можно поставить пакет `can-utils`:

```
apt update && apt install can-utils
```

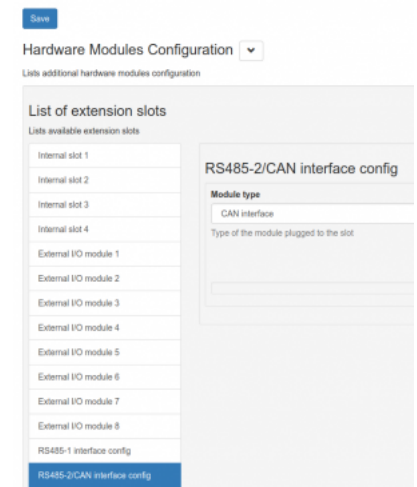
Примеры команд:

`/etc/network/interfaces`



Настройка `can`-интерфейса. `Allow-hotplug` позволяет интерфейсу подниматься автоматически

`/etc/wb-hardware.conf`



Перевод порта RS485-2 в режим CAN и включение драйвера `can` в linux

- Отправка четырех байт с адресом 123:

```
cansend can0 123#DEADBEEF
```

- Показывать входящие пакеты:

```
candump can0
```

- Показать статистику:

```
cat /proc/net/can/stats
```

Обратите внимание, что в случае проблем на шине (нет терминатора, нет принимающего устройства, короткое замыкание), контроллер может прекратить работу. Для того чтобы начать работу вновь, выполните:

```
ifconfig can0 down && ip link set can0 up type can bitrate 125000
```

Больше информации смотрите в вики проекта «AmadeuS», статья «CAN bus Linux driver (http://www.armadeus.com/wiki/index.php?title=CAN_bus_Linux_driver)».

1-Wire в контроллерах Wiren Board

- English
- русский

Купить датчик температуры DS18B20 1-Wire

Contents

Подключение

- Подключение по трём проводам
- Подключение по двум проводам
- Прокладка шины 1-Wire

Поддержка в ПО

Полезные ссылки



Самый популярный температурный датчик 1-Wire — DS18B20, установленный в герметичном корпусе.
Купить

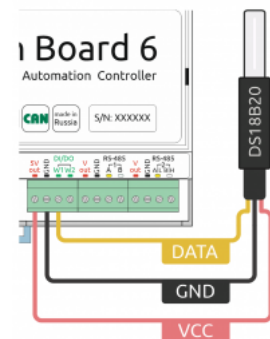
Подключение

В контроллере уже есть резистор 3 кОм подтяжки между шиной Data и VCC — внешний резистор не нужен.

Подключение по трём проводам

Датчик имеет три вывода. Их цвета могут меняться от модели к модели, желательно найти документацию на свою модель.

| Сигнал | Клеммник | Цвет: модель 1 | Цвет: модель 2 | Цвет: модель 3 |
|--------------------|----------|----------------|----------------|----------------|
| Vdd (VCC, питание) | +5V Out | Красный | Красный | Красный |
| GND (земля) | GND | Чёрный | Чёрный | Желтый |
| DQ (DATA, данные) | 1W | Синий | Жёлтый | Зелёный |



Подключение датчика 1-Wire к контроллеру Wiren Board

Подключение по двум проводам

Соедините контакты питания и земли датчика и подключите их к земле контроллера. При таком подключении датчик будет брать питание с канала данных.

Этот способ не рекомендуется, особенно для подключения нескольких датчиков: тока с линии данных может не хватить для всех датчиков, к тому же замедляется опрос — время тратится на зарядку внутренних емкостей датчиков напряжением от линии данных.

Прокладка шины 1-Wire

Количество возможных датчиков и надежность их работы зависит от длины шины, её топологии и кабеля.

Обычно в домашних условиях надежно работает до 20 датчиков по 5 метров кабеля, соединенных звездой.

Основной документ при проектировании шины — инструкция от разработчика 1-Wire. Основные тезисы:

- Длина шины при подключении одного датчика до 200 метров.
- При подключении нескольких датчиков, подключайте их к питанию 5 В (не используйте двухпроводную схему).
- Прокладка линии одной шиной лучше, чем прокладка звездой.
- Для прокладывания длинной шины или в условиях повышенных помех (например, в щитке) — используйте витую пару, например, Cat 5, лучше экранированную.

Подключение по витой паре — это сигнал по одной жиле пары, земля по второй и аналогично питание: плюс питания по одной жиле пары, минус по второй. Минус питания соединить с сигнальной землёй. Экран с одной стороны соединить с минусом питания.

Поддержка в ПО

Поддерживаемые протоколы

| | |
|---|--|
| Опрос датчиков и работа с устройствами (в базовой комплектации) | 1-Wire • DLMS/COSEM • Modbus RTU/TCP Master • ГОСТ МЭК 61107 • СПОДЭС (ГОСТ Р 58940-2020) |
| Опрос датчиков и работа с устройствами (с помощью модулей расширения) | KNX • eBUS • OpenTherm • Z-Wave • Zigbee |
| Системы верхнего уровня | KNX • Modbus RTU/TCP Slave • MQTT • OPC UA • SNMP • Zabbix • МЭК 104 • SmartWeb |
| ПО верхнего уровня | Grafana • Home Assistant • IntraHouse • IntraSCADA • IRidium Server • MasterSCADA • Nagios • SimpLight SCADA • Rapid SCADA • SAYMON • Zabbix |
| Голосовые ассистенты и облачные решения | Сири и Apple HomeKit • Салют от SberDevices • Яндекс Алиса |

Протестированные устройства сторонних производителей

| | |
|--|--|
| Адаптеры протоколов | Шлюз ECODim DALI GW2 |
| Датчики климата | DS18B20 и клоны • Kvaдро 1WIRE-RS485 • RLDA NL-3DPAS-M • RLDA NL-1S111 • Wellpro WP3066ADAM • Даджет MT8057/MT8057S • РД MSU21 • РД MSU24 • РД MSU34+TLP • РД MSU34+THLP • Эксис ИВТМ-7 М 3 |
| Датчики уровня | ЭККОРТ ДБ-2 |
| Диммеры | DALI • Philio PAD07-RU • Uniel UCH-M131RC/0808 • Uniel UCH-M141RC/0808 • РД DDL04R • РД DDL24 • РД DDL84R-V • РД DDM845R |
| Конвекторы | Varmann QTherm |
| Кондиционеры | Haier YCJ-A002 • Z-Wave ИК-передатчик PAR01-RU |
| Контроллеры вентиляции и климата | Mautomatics JL204C5 (Breezart 550 Lux) • GTC (General Thermo Controllers) Syberia 5.0 • SystemAir VR 300 |
| Контроллеры холодильного оборудования | Carel BASIC(PYEZ)/EASY(PJEZ) • Danfoss EKC 204A1 / AK-CC 210 • Danfoss EKC 202B • Danfoss EKC 202D • Danfoss ERC 211/ERC 213/ERC 214 • Eliwell IDPlus 974 |
| Метеостанции | Netatmo Urban Weather Station |
| Модули ввода-вывода | Wellpro WP8026ADAM • Wellpro WP8027ADAM • Wellpro WP8028ADAM • Wellpro WP9038ADAM |
| Модули реле | РД DRB88 • Rubetek TZ78 • ICP DAS tM-P3R3 • ICP DAS LC-103 • Uniel UCH-M111RX/0808 • Uniel UCH-M121RX/0808 |
| Моторы для штор/Электрокарнизы | Akko AM82 • Dooya DM35EQ • Dooya DT82 • WinDeco • Somfy SDN • SunFlower KT82TV • Somfy RS485 RTS transmitter |
| Управление двигателями (преобразователи частоты) | Vacon/Danfoss 10 • Danfoss VLT Microdrive FC51 • T13-400W-12-H |
| Счётчики воды | Пульсар • Пульсар-М • Элехант СВД-15 • Элехант СВД-20 • Счётчики с импульсным выходом |
| Счётчики тепла | Пульсар |
| Счётчики электроэнергии | CSQ PD561Z-9SY • Peacefair PZEM-016 • Eastron SDM120M • Eastron SDM220M • Меркурий 200 • Меркурий 201 • Меркурий 203.2Т • Меркурий 204 • Меркурий 206 • Меркурий 208 • Меркурий 230 • Меркурий 231 • Меркурий 234 • Меркурий 236 • Меркурий 238 • Милур 104 • Милур 105 • Милур 107 • Милур 305 • Милур 307 • Нева МТ 113 • Нева МТ 123 • Нева МТ 124 • Нева МТ 323 • Нева МТ 324 • Энергомера CE301 • Энергомера CE102M • Энергомера CE303 • Энергомера CE308 |
| Термостаты | BAC-6000 Series • BHT-6000 Series • Cityron ПУ-3 (Modbus) • Heatit Z-TEMP2 • Hessway • Siemens RDF302 |
| Увлажнители | CAREL Humisonic |
| Прочее | DIY • Shelly UNI • Tasmota • ESPHome |

Устройства с аналоговым или цифровым выходом

| | |
|---|--|
| Низковольтная нагрузка | Реле с управляющим напряжением 12-24 В • Светодиоды • Низковольтные вентиляторы • Низковольтные сигнальные лампы |
| Датчики с аналоговым выходом | Датчики температуры, давления и другие, имеющие на выходе ток или напряжение |
| Счётчики с импульсным выходом | Счётчики электроэнергии, воды, тепла и другие с импульсным выходом |
| Устройства с выходом «открытый коллектор» | Устройства с выходом «открытый коллектор» |
| Устройства с питанием 220 В | Лампы • Контакторы • Другое оборудование, питающееся от 220 В |

Подключение периферийных устройств к контроллеру Wiren Board

Contents

Управление низковольтной нагрузкой

Технические подробности

Подключение нагрузки

Датчики с аналоговым выходом по напряжению

Датчики с аналоговым токовым выходом

Датчики/счетчики с импульсными выходами/кнопки

Устройства с выходом открытый коллектор

Контакты с управляющим напряжением 220В

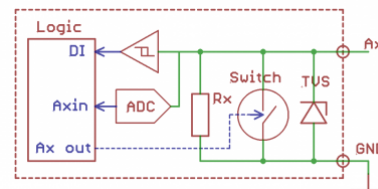


Схема входов/выходов A1-A4

Управление низковольтной нагрузкой

Технические подробности

Для управления низковольтной нагрузкой в Wiren Board предназначены так называемые «транзисторные выходы», они же FET или **открытый коллектор**. С их помощью можно управлять включением низковольтных ламп, светодиодных лент, внешних блоков реле и т.п.

Транзисторными выходами можно управлять из веб-интерфейса, там они называются соответственно клеммам: **A1_OUT** — **A4_OUT**.

Для сокращения общего числа клеммников каналы управления низковольтной нагрузкой совмещены с каналами АЦП. Поэтому выходы имеют большое, но конечное сопротивление — 100кОм. Это может вызывать, например, слабое свечение светодиодных лент, но проблему можно решить — подтянуть вывод резистором к питанию.

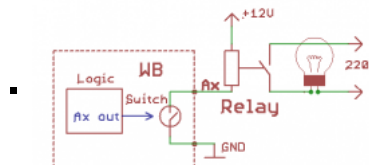
В контроллерах Wiren Board 6 выходы защищены от импульсных перенапряжений, короткого замыкания и перегрева.

Подключение нагрузки

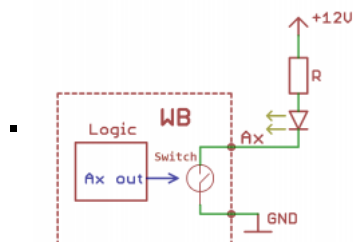
Чтобы подключить нагрузку, подключите «плюс» нагрузки к источнику питания, а «минус» к транзисторному входу. Нагрузка включается подачей высокого уровня на выход. Если суммарный ток на канале превышает 2 А — дополнительно подключите клемму **GND** к минусу источника питания.

При управлении индуктивной нагрузкой (реле), возникают всплески напряжения. Для защиты от перенапряжения в контроллер встроены защитные диоды — внешних защитных элементов не требуется.

Также для управления низковольтной нагрузкой можно использовать модуль дискретных выходов WBIO-DO-HS-8.



Пример подключения реле к выходам A1-A4



Датчики с аналоговым выходом по напряжению

Клеммы A1-A4 могут измерять напряжение, поэтому к ним можно подключить датчики с аналоговым выходом по напряжению, например, температурные сенсоры.

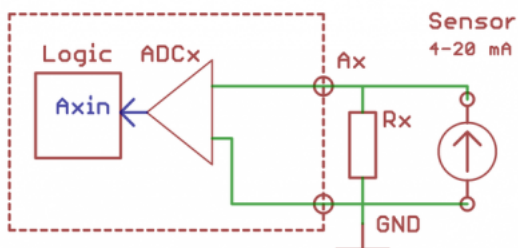
Подключите землю устройства к клемме GND, или соедините с общей земляной шиной. Выход датчика подключите к одной из клемм **Ax**.

Для точного измерения напряжения можно использовать модуль ввода-вывода WBIO-AI-DV-12_I/O_Module или модуль аналоговых входов WB-MAI11.

Датчики с аналоговым токовым выходом

Специальных токовых входов в WB нет, но можно, используя резистор $R_x = 100-300\text{ Ом}$, ток преобразовать в напряжение и подключить по аналогии с датчиком, имеющим аналоговый выход по напряжению.

Так же можно использовать модуль ввода-вывода WBIO-AI-DV-12_4-20mA или модуль аналоговых входов WB-MAI11.



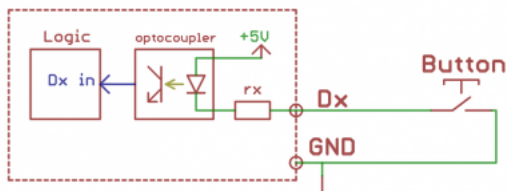
Датчики/счетчики с импульсными выходами/кнопки

Такие устройства формируют сигнал, замыкая подходящие к ним два провода.

Способы подключения к контроллеру:

1. С помощью клемм **Ax** контроллера. Подключите один провод к источнику питания 5-24 В, второй провод к клемме **Ax**. Подробнее смотрите на странице Подключение устройств с импульсными выходами.
2. С помощью модуля ввода-вывода WBIO-DI-WD-14 (14 каналов). Один из проводов подключите к GND, второй к клемме **Dx** модуля.
3. С помощью модуля расширения WBE2-DI-DR-3 (3 канала). Один из проводов подключите к GND, второй к клемме **Ox** соответствующей модулю расширения.

Некоторые счетчики имеют импульсный выход на оптроне, тогда два провода имеют полярность — «плюс» и «минус». В таком случае минус подключается к **GND**, а «плюс» ко входу. Либо для первого способа — «плюс» к питанию, а «минус» к **Ax**.



Устройства с выходом открытый коллектор

Есть три способа подключить такие устройства к контроллеру:

1. С помощью модуля ввода-вывода WBIO-DI-WD-14 (14 каналов). Выход «открытый коллектор» подключите к клемме **Dx** модуля. Землю устройства к iGND модуля.
2. С помощью модуля расширения WBE2-DI-DR-3 (3 канала). Выход «открытый коллектор» подключите к клемме **Ox** соответствующей модулю расширения, а землю устройства к GND контроллера.
3. Можно подключать к клеммам **A1-A4**, при этом нужно также подключить внешний подтягивающий резистор между **5Vout** и соответствующей клеммой **Ax** номиналом около 10 кОм. Соедините земли устройства и контроллера.

Контакты с управляющим напряжением 220В

Используйте модуль ввода-вывода с релейными выходами, например WBIO-DO-R1G-16.

Подключите управляющую катушку контактора через реле модуля расширения, схему подключения смотрите в разделе «Монтаж» на странице используемого модуля.

Модуль WBIO-DO-R1G-16 содержит TVS, защищающий контакты реле от искрения. Внешние защитные компоненты не требуются.

ADC

- English
- русский

Как получить напряжение с АЦП

1. Напряжение должно быть в диапазоне допустимых значений.
2. Клеммники Aх выполняют две функции: АЦП и управление низковольтной нагрузкой. Перед измерением напряжение, поставьте соответствующий выход управления низковольтной нагрузкой в положение "выключено". Например, если вы подключаетесь к клемме A1, выключите в веб-интерфейсе A1_OUT (раздел Relays & FETs).
3. Подключите ваш источник к клемме. Значение напряжения сразу появится в веб-интерфейсе, в устройстве ADCs. Также значение транслируется в систему сообщений MQTT.

Также значение можно получать в ручном режиме: Низкоуровневая работа с ADC.

Входное напряжение

Демон wb-homa-adc транслирует значение в очередь сообщений MQTT в топик /devices/wb-adc/controls/Vin . Таким образом, значение отображается в веб-интерфейсе как канал Vin устройства ADCs

Список АЦП для контроллера WB6

В Wiren Board 6 каналы АЦП процессора подключены к клеммникам A1-A4. Также на АЦП заведено входное напряжение (после входных диодов) и напряжение на клемме 5Vout.

См. Wiren Board 6.2: Peripherals#Каналы АЦП

Списки АЦП для старых версий контроллера

Дискретные входы

DI (*digital input* - цифровой, дискретный вход) - вход с двумя состояниями. Используется для подключения герконов, счётчиков воды и электричества, кнопок.

Входы бывают двух типов:

1. "**Наличия напряжения**" (Wet contact) - вход (Ах у контроллера) с двумя состояниями: либо на него "подано напряжение", либо "не подано". Для этого один контакт кнопки/геркона/счётчика подключается к источнику напряжения (например, к клемме Vout), а другой к DI.
2. "**Сухой контакт**" (Dry contact) - входы для подключения "**Сухой контакт**" (W1, W2 у контроллера, входы на модулях реле WB-MR и т. д.), которые проверяют, замкнуты ли они на "землю".

Один контакт кнопки/геркона/счётчика подключается к земле (к клемме GND), а другой к входу "сухой контакт". Так же можно подключать выходы типа "открытый коллектор" (соединить земли и вход к выходу). Состояние входа:

- замкнуто на землю - "включено"
- не замкнуто на землю - "выключено"

"Сухими" обычно называют контакты, которые не подключены ни к каким другим цепям внешней системы: контакты реле, герконы, выключатели. Выход "открытый коллектор", например, нельзя назвать сухим контактом, так как он гальванически не развязан с внешним устройством.

Входы некоторых устройств работают в обоих режимах (в зависимости от схемы подключения), например модуль ввода-вывода WD-14.

Отладочный порт

- English
- русский

Contents

Описание

Контроллеры Wiren Board 5.8, 5.9, 6.x и 7.x

- Драйвера адаптеров и названия виртуальных COM-портов
- Особенности подключения к WB6.4 и новее из Linux
- Параметры подключения

Физическая реализация в контроллерах

Контроллеры Wiren Board ≤5.6

Описание

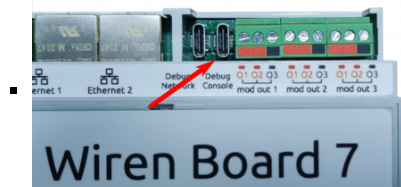
У контроллеров Wiren Board есть отладочный порт, через который можно получить доступ к консоли контроллера. Через него можно взаимодействовать с загрузчиком, следить за загрузкой операционной системы.

Отладочный порт — это инструмент доступа к контроллеру, когда вы не можете подключиться к нему через веб-интерфейс или по SSH. Мы не рекомендуем его для постоянного использования.

Для подключения к порту используйте одну из программ, перечисленных в статье Работа с последовательным портом.

Контроллеры Wiren Board 5.8, 5.9, 6.x и 7.x

- Расположение разъёма Debug Console на контроллерах Wiren Board



Wiren Board 7.x



Wiren Board 6.x



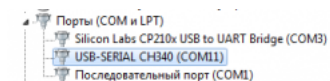
Wiren Board 5.8 и 5.9

Разъём отладочного порта подписан на корпусе контроллера как **Debug Console**.

В зависимости от версии контроллера, для подключения используйте стандартный кабель от смартфонов:

- **Wiren Board 5.8, 5.9 и 6** — USB (A) - Micro-USB (B).
- **Wiren Board 7** — USB (A) - USB-C.

Разъём USB (A) подключается к компьютеру, а Micro-USB (B) или USB-C — к контроллеру.



Номер порта в Windows

Кабели от зарядных устройств подходят, но бывают исключения. Лучше выбирать такие, на которых указано, что они предназначены для передачи данных. Если кабель выбран правильно, Windows сообщит звуком об обнаружении устройства.

Драйвера адаптеров и названия виртуальных COM-портов

Внутри контроллера установлен переходник USB-UART.

- **Linux:** адаптер обычно определяется автоматически, при подключении адаптера в выводе команды `dmesg` должна появиться строка:

```
usb 1-1: ch341-uart converter now attached to ttyUSB0
```

или, для версий контроллера 6.4 и младше:

```
cdc_acm:3-6:1.0: ttyACM0: USB ACM device
```

Номер порта `ttyUSBx` или `ttyACMx` может меняться, в зависимости от уже подключенных устройств. Смотрите список файлов в папке `/dev/`. Консольный порт контроллеров для версий 6.4 и младше определяется автоматически как устройство `/dev/ttyACM0`.

Смотрите также раздел Особенности подключения к WB6.4 и новее из Linux.

- **Windows:** может потребоваться установка драйвера (см. таблицу Поддержка в операционных системах). В Диспетчере устройств в разделе **Ports(COM&LPT)** появится виртуальный COM-порт контроллера.
- **macOS:** начиная с High Sierra отладочная консоль контроллеров определяется из коробки. Предыдущие версии операционной системы, возможно, потребуют установки драйверов. При подключении создается устройство `/dev/tty.usbserial-1410` или `/dev/tty.usbmodem00001` (для контроллеров версии 6.4 и младше).
- **Android:** поздние версии Android при подключении через OTG поддерживают отладочную консоль контроллеров без дополнительных драйверов. Тестировалось с приложением USB Serial Console.



Кабель USB (A) - Micro-USB (B)



Кабель USB (A) - USB-C

| Поддержка в операционных системах | | | | | |
|-----------------------------------|---|---|------------|-----------------------|--------------------------------|
| Версия контроллера | Linux | Windows XP, 7, 8 | Windows 10 | MacOS X (High Sierra) | Android 9 + USB Serial Console |
| 7.x 6.4 - 6.9 | Модуль ядра <code>cdc_acm</code> (все новые дистрибутивы — из коробки), | Нужны <code>inf</code> файлы: Медиа:HT42B534_inf.zip | Из коробки | Из коробки | Из коробки |
| 6.0 - 6.3 5.8 - 5.9, | Модуль ядра <code>ch341</code> (все новые дистрибутивы — из коробки) | Нужен драйвер CH341SER | | CH341SER_MAC | |

Особенности подключения к WB6.4 и новее из Linux

На некоторых Linux-системах процесс `ModemManager` автоматически открывает устройство `/dev/ttyACM0`. Чтобы `ModemManager` не мешал работе с портом, его надо отключить для этого устройства с помощью правил `udev`:

1. Создайте файл исключений `udev`:

```
sudo nano /etc/udev/rules.d/99-wb-debug-usb.rules
```

2. Добавьте в него строчку:

```
ATTRS{idVendor}=="04d9" ATTRS{idProduct}=="b534", ENV{ID_MM_DEVICE_IGNORE}="1"
```

3. После сохранения файла перезагрузите правило `udev`:

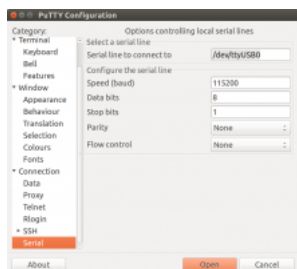
```
sudo udevadm control --reload-rules
```

4. В свежих операционных системах может понадобиться отредактировать файл `/lib/systemd/system/ModemManager.service`. Смотрите подробнее по ссылкам: [ModemManager does not honor blacklisted ttys](#) и [SystemD ModemManager: failed to set dtr/rts](#).

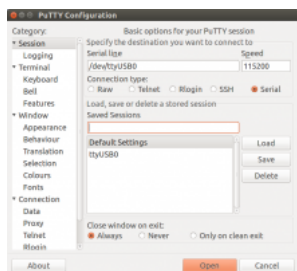
Параметры подключения

Параметры подключения к Debug-консоли контроллера Wiren Board

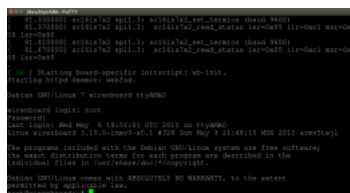
| Параметр | Значение | Описание |
|-----------------------|---|---------------------------------------|
| Serial Port | <ul style="list-style-type: none"> ■ Windows — COMx ■ Linux — /dev/ttyACM0 или до версии WB6.4 — /dev/ttyUSB0 | Подробнее смотрите выше в этой статье |
| Baud rate | 115200 | Скорость, бит/с |
| Data bits | 8 | Количество битов данных |
| Parity | None | Бит чётности |
| Stop bits | 1 | Количество стоповых битов |
| Hardware flow control | None | Аппаратный контроль потока |
| Software flow control | None | Программный контроль потока |



Настройка параметров для подключения к Debug-консоли контроллера в программе PuTTY



Открытие сессии в программе PuTTY



Приветственное сообщение контроллера при подключении к его Debug-консоли

Физическая реализация в контроллерах

| Версия контроллера | Тип разъема | Название разъема | Интерфейс | Чип USB/UART |
|----------------------|--------------------------|--|--------------|--------------|
| 7.x | USB-C | Debug Console | USB/UART | HT42B534 |
| 6.4–6.9 | Micro-USB | Debug Console | USB/UART | HT42B534 |
| 6.0 - 6.3, 5.8 - 5.9 | | | USB/UART | CH340 |
| 5.6.1 | 3-pin UART (Gnd, Rx, Tx) | UART | UART (3,3 В) | — |
| 5.3, 4 | | Debug UART | | |
| 3.5, 2.8 | 3-pin UART (Gnd, Tx, Rx) | UEXT1 (выводы 2,3,4) и 3-pin разъем без названия | | |

Программно во всех случаях это последовательный порт.

Контроллеры Wiren Board ≤5.6

Подключение к отладочному порту в Wiren Board 5.6 и старше

Зуммер (звуковой излучатель)

Contents

Описание

Управление из веб-интерфейса

Управление из движка правил

Управление из python

Низкоуровневая работа

О ШИМ и пересчёт параметров

Номер pwm-порта для sysfs

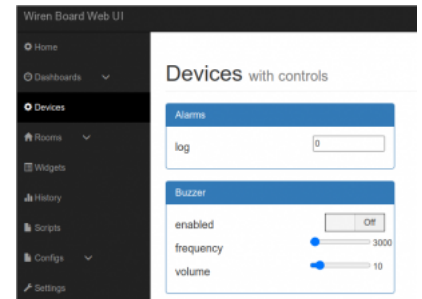
Работа из sysfs

Описание

Контроллер Wiren Board имеет на борту Зуммер (звуковой излучатель). Зуммер питается от 5В и управляется ножкой gpio процессора в режиме ШИМ. Управлять зуммером можно через sysfs-интерфейс ядра и различное ПО поверх него. Сейчас реализовано управление из веб-интерфейса, движка правил wb-rules и python.

Управление из веб-интерфейса

В веб-интерфейсе контроллера управление зуммером доступно во вкладке "Devices". Параметр "Frequency" - частота звука в Гц. "Volume" - громкость (в условных единицах, шкала линейная). Параметры сохраняются при перезагрузке контроллера.



Управление зуммером

Управление из движка правил

Управление зуммером, выведенное в веб-интерфейс - это виртуальное устройство, созданное системным правилом wb-rules при старте контроллера. Исходный код правила доступен на нашем github (<https://github.com/wirenboard/wb-rules-system/blob/master/rules/buzzer.js>).

О том, для чего нужны виртуальные устройства, можно узнать подробнее в описании движка правил.

Системное правило внутри реализует пересчёт тональности и громкости (см раздел о пересчёте) и работу с pwm через sysfs (см соответствующий раздел). Наружу пользователю доступно устройство "buzzer", имеющее несколько mqtt-контролов:

| Device | Control | Тип | Максимальное значение | Описание |
|--------|-----------|--------|-----------------------|----------------------|
| Buzzer | Frequency | Range | 7000 | Частота звука |
| | Volume | Range | 100 | Громкость, % |
| | Enabled | Switch | | Включение/отключение |

Контролы устройства можно использовать в собственных правилах. Подробнее о структуре mqtt-топиков виртуальных и физических устройств можно узнать из нашей mqtt-конвенции (<https://github.com/wirenboard/conventions/blob/main/README.md>).

Управление из python

На контроллерах Wiren Board работать с зуммером можно из python с помощью модуля *beeper* из пакета *wb_common*. Это обёртка вокруг интерфейса sysfs. Модуль предустановлен на все контроллеры в составе deb-пакета *python-wb-common*. Исходный код доступен на нашем github (https://github.com/wirenboard/wb-common/blob/master/wb_common/beeper.py).

Пример работы из python:

```
from wb_common import beeper
beeper.beep(0.5, 2)
```

Поддерживаются все настройки sysfs-интерфейса (пересчёт нужно проводить вручную; см раздел о пересчёте).

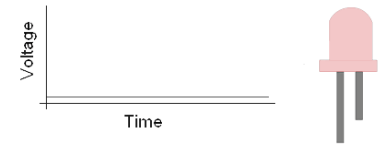
Низкоуровневая работа

О ШИМ и пересчёт параметров

ШИМ (PWM) - это распространённый способ управления мощностью, подаваемой к нагрузке.

В контексте управления зуммером, нас интересуют 2 параметра PWM:

- Коэффициент заполнения (duty cycle) - влияет на громкость звука. Обычно, считается в процентном соотношении от периода сигнала.
- Частота PWM (frequency) - влияет на высоту звука (чем выше частота, тем выше и звук). Единица, обратная периоду сигнала.



Duty Cycle: 0%

Duty cycle управляет яркостью светодиодов / громкостью Зуммера

Ядро Linux предоставляет интерфейс sysfs для pwm, который принимает частоту pwm и duty cycle в **наносекундах (10⁻⁹С)**! Поэтому, для низкоуровневого управления Buzzer'ом нужно производить пересчёт желаемой частоты из kHz в период в наносекундах по формуле: **$T(\text{ns}) = 1\,000\,000 / f(\text{kHz})$**

Номер pwm-порта для sysfs

Ножка gpio настраивается, как выход PWM в dts ядра linux. Подробнее можно посмотреть [на нашем github (<https://github.com/wirenboard/linux/blob/ef2d87e222b365848fe7262c022ca887b6449432/arch/arm/boot/dts/imx6ul-wirenboard61.dts#L495>)].

- Для контроллеров WB6.X.X номер порта = 0, (для всех контроллеров до WB6.X.X номер порта = 2)
- Номер порта можно узнать, выполнив команду

```
echo $WB_PWM_BUZZER
```

Во всех примерах далее будем считать, что номер pwm-порта = 0.

Работа из sysfs

Для работы с pwm через sysfs нужно:

1. Экспортировать порт

```
echo 0 > /sys/class/pwm/pwmchip0/export
```

После этого появляется директория /sys/class/pwm/pwmchip0/pwm0

2. Записать период pwm в наносекундах

```
echo 250000 > /sys/class/pwm/pwmchip0/pwm0/period # устанавливаем период в 250 000 нс, т.е. в 250мкс, что соответствует частоте 4кГц
```

3. Записать громкость (пересчитав из duty-cycle)

```
echo 125000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle # максимальная громкость достигается при duty_cycle = period / 2 => устанавливаем duty_cycle в 125 000 нс
```

4. Включить выход PWM

```
echo 1 > /sys/class/pwm/pwmchip0/pwm0/enable
```

Для выключения зуммера, нужно записать 0:

```
echo 0 > /sys/class/pwm/pwmchip0/pwm0/enable
```

Пример bash-скрипта для работы с pwm (<https://github.com/contactless/wirenboard/tree/master/examples/beeper>)

Установка периода в наносекундах. Пересчёт из частоты (в килогерцах) в период (в наносекундах) производится по формуле: **$T(\text{ns}) = 1\,000\,000 / f(\text{kHz})$**

Consumer IR

- English
- русский

В этой статье описывается прием и передача сигналов пультов управления бытовой техникой, работающих в инфракрасном диапазоне (то есть большинство пультов от телевизоров, кондиционеров, и т. д.).

Contents

Аппаратная часть

Технические детали

Программная часть

Установка и настройка

Скачивание существующего конфига пульта

Создание нового конфига пульта

MQTT-интерфейс

Устранение неисправностей

Не работает прием

Не работает передача

Аппаратная часть

Внимание! В ревизиях контроллера 5.8 и старше ИК-порт для подключения приемопередатчика отсутствует!

На Wiren Board 5 (до ревизии 5.6.1 включительно) необходимые сигналы выведены на разъем mini-jack 3.5mm с со следующей распиновкой:

- Tip (левый аудиоканал) - питание 5В/передача IR.
- Ring (правый аудиоканал) - прием IR
- Sleeve (земля) - земля

Распиновка разъёма позволяет подключать распространённые ИК-приёмники и передатчики от бытовой техники.

Для приема ИК-сигналов необходимо подключить приемник (совмещённый приёмопередатчик WB-CIR-TR). Номинальная частота несущей приемника должна примерно соответствовать частоте несущей используемого пульта, в противном случае возможно ухудшение или полная невозможность приема сигнала от пульта.

Для передачи ИК-сигналов рекомендуется использовать готовый совмещённый ИК-приёмопередатчик WB-CIR-TR, Samsung IR Blaster bn96/26652a или подобный.

Технические детали

Распиновка разъёма mini-jack 3.5mm на Wiren Board 5 :

- Tip (левый аудиоканал) - питание 5В/передача IR.
- Ring (правый аудиоканал) - прием IR
- Sleeve (земля) - земля

Для приема ИК-сигналов используются приемники типа TSOP1738, AX-1838HS или аналогичных. Для передачи ИК-сигналов используется ИК-светодиод, токоограничивающий резистор не обязателен (встроен в контроллер) При передаче напряжение питания будет промодулировано передаваемым сигналом и несущей. Приёмопередатчик WB-CIR-TR содержит приёмник типа TSOP и ИК-светодиод. Для питания приёмника используется паразитное питание.

Программная часть

На уровне ядра за работу с ИК на Wiren Board 5 отвечает драйвер **lirc-pwm**. Он предоставляет устройство `/dev/lirc1`.

С этим устройством взаимодействует демон **lircd** из состава пакета LIRC, осуществляющий декодирование последовательностей импульсов в события нажатых на пульте кнопок согласно файлам конфигурации пультов. Также этот демон способен осуществлять обратное преобразование для передачи сигналов нажатия кнопок управляемым устройством.

Наконец, трансляцию событий от **lircd** в сообщения MQTT и обратно осуществляет демон **wb-mqtt-lirc**



ИК-приемник



ИК-передатчик

Установка и настройка

Необходимые пакеты (`wb-mqtt-lirc`, `lirc-scripts`, `wb-mqtt-lirc`), а так же правильный конфиг-файл `/etc/lirc/hardware.conf` установлены в стандартном образе, начиная с версии 201512070633.

- Скачать или сгенерировать файл конфигурации пульта и положить его в `/etc/lirc/lircd.conf.d/`.

Скачивание существующего конфига пульта

```
root@wirenboard:~# irdb-get update
root@wirenboard:~# irdb-get find 687C
sony/RM-687C.lircd.conf
root@wirenboard:~# irdb-get download sony/RM-687C.lircd.conf
Downloaded https://sourceforge.net/p/lirc-remotes/code/ci/master/tree/remotes/sony/RM-687C.lircd.conf as RM-687C.lircd.conf
root@wirenboard:~# mv RM-687C.lircd.conf /etc/lirc/lircd.conf.d/
```

Создание нового конфига пульта

TODO: описать как пользоваться `irtecord`

После завершения конфигурации, следует перезапустить службы `lirc` и `wb-mqtt-lirc`

```
root@wirenboard:~# service lirc restart && service wb-mqtt-lirc restart
```

MQTT-интерфейс

`wb-mqtt-lirc` создает устройство `/devices/wb-lirc/`, принятые нажатия публикуются в топике `/devices/wb-lirc/controls/Key` в формате `<название_пульта>:<название_кнопки>:<повтор>`. После отпускания кнопки в этом топике публикуется сообщение с пустым телом.

Пример:

```
root@wirenboard:~# mosquitto_sub -v -t '/devices/wb-lirc/#'
/devices/wb-lirc/meta/name IR Remote Control
/devices/wb-lirc/controls/Key/meta/order 1
/devices/wb-lirc/controls/Key/meta/type text
/devices/wb-lirc/controls/Key Sony_RM-687C:KEY_7:0
/devices/wb-lirc/controls/Key Sony_RM-687C:KEY_7:1
/devices/wb-lirc/controls/Key Sony_RM-687C:KEY_7:2
/devices/wb-lirc/controls/Key Sony_RM-687C:KEY_7:3
/devices/wb-lirc/controls/Key (null)
```

Для отправки нажатия кнопки следует опубликовать сообщение в формате `<название_пульта>:<название_кнопки>` в топике `/devices/wb-lirc/controls/Key/on`

Устранение неисправностей

Не работает прием

- Убедиться что драйвер "видит" импульсы от приемника.
 - Остановить сервис `lirc` (командой `service lirc stop`)

```
root@wirenboard:~# mode2 -d /dev/lirc1
```

Должны бежать цифры при нажатии кнопок на пульте, если нет - проблема в "железе" или в несоответствии несущей пульта и приемника. Следует попробовать использовать другой приемник.

- Убедиться что `lirc` может декодировать последовательности импульсов согласно конфигу пульта:
 - Запустить обратно `lirc` (`service lirc start`)

```
root@wirenboard:~# irw
```

Должны бежать строчки при нажатии кнопок на пульте. Если `mode2` работает, а `irw` - нет, то проблема с конфигом пульта. Стоит попробовать использовать другой конфиг, либо создать новый.

Не работает передача

- Попробовать передавать обращаясь к LIRC напрямую:

```
root@wirenboard:~# irsend SEND_START Sony_RM-687C KEY_3
...
root@wirenboard:~# irsend SEND_STOP Sony_RM-687C KEY_3
```

(названия пульта и кнопки должны соответствовать используемому конфигу).

- Убедиться что аппаратная часть работает - например, можно посмотреть на ИК-светодиод через камеру смартфона (светодиод должен мигать при передаче)

Watchdog

- English
- русский

Сторожевой таймер (англ. *watchdog*) — аппаратно реализованная схема контроля за зависанием системы.

Представляет собой отдельную микросхему-компаратор, ведущую отсчёт времени. Если таймер досчитывает до заданного времени (около 15 секунд), происходит перезагрузка по питанию (выключение одной из линий питания на 3-4 секунды). В нормальном режиме таймер периодически сбрасывается подачей переменного сигнала, подаваемого на вход таймера с одного из выводов GPIO процессора. Этот GPIO контролируется специальным сервисом Linux `watchdog`. Интерфейс сторожевого таймера `/dev/watchdog1`, конфигурация сервиса хранится в файле `/etc/watchdog.conf`.

Отключение сторожевого таймера

Сторожевой таймер иногда требуется отключить:

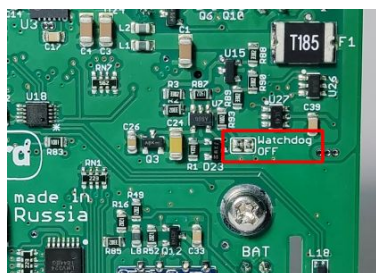
1. Если вам нужно полностью выключить контроллер, не снимая с него питание (например, по событию от источника бесперебойного питания) - если сторожевой таймер будет работать, то контроллер даже после команды `halt` через некоторое время перезагрузится по питанию, и начнёт работать. При отключенном сторожевом таймере возобновление работы контроллера будет возможно только при ручном сбросе питания контроллера.
2. Если вы неправильно настроили одну из важных служб контроллера, и он ушёл в циклическую перезагрузку: из-за неправильной конфигурации службы не будут запускаться, а сторожевой таймер будет замечать их отсутствие и перезапускать контроллер.

Отключение сторожевого таймера аппаратным способом

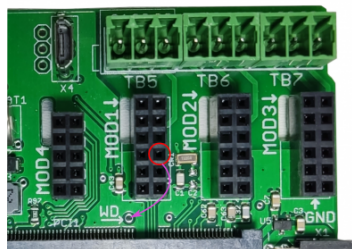
Для отключения требуется разобрать корпус контроллера и:

- в Wiern Board 7.2 — запаять перемычку Watchdog OFF;
- в Wiren Board 6.9 нет простого способа аппаратно отключить сторожевой таймер;
- в Wiern Board 6.8 — соединить контакт WD с +5V;
- в Wiern Board 6.7 — соединить контакт WD с GND;
- в Wiren Board 6.3 - 6.6 и Wiren Board 5 соединить один из выводов разъёма ON/OFF с GND.

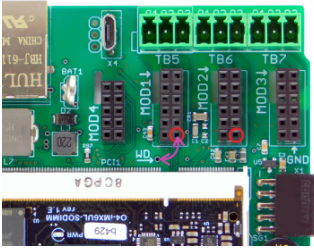
На иллюстрациях красными кружками показаны контакты, к которым нужно подключить контакт WB или контакт из разъёма ON/OFF. Фиолетовой стрелкой показан пример подключения.



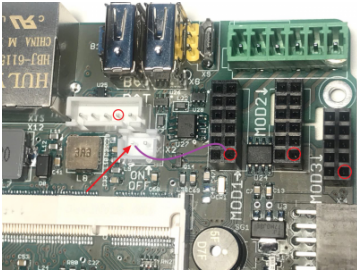
Wiren Board 7.2. Запаять перемычку Watchdog OFF



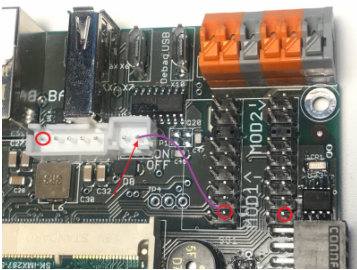
Wiren Board 6.8. Подключить WD к +5V



Wiren Board 6.7. Подключить WD к GND



Wiren Board 6. Подключить контакт ON/OFF к GND



Wiren Board 5. Подключить контакт ON/OFF к GND

Отключение сторожевого таймера программным способом

Этим способом вы сможете только остановить циклическую перезагрузку из-за неправильной работы ПО. Добиться им полного выключения контроллера при наличии питания не получится.

Чтобы отключить сторожевой таймер, остановите его службу:

```
systemctl stop watchdog
```

Но после перезагрузки контроллера служба сторожевого таймера запустится снова.

Если вы исправили ошибки в работе ПО и хотите запустить обратно сторожевой таймер без перезагрузки контроллера, выполните

```
systemctl start watchdog
```

Если вы хотите навсегда отключить слежение сторожевого таймера за одним из сервисов, отредактируйте конфигурационный файл `/etc/watchdog.conf`, закомментировав строки соответствующих сервисов (в этом примере отключено слежение за `nginx`):

```
# Test if vital daemons are running
pidfile = /var/run/syslogd.pid
pidfile = /var/run/sshd.pid
pidfile = /var/run/mosquitto.pid
#pidfile = /var/run/nginx.pid
```

а затем выполните

```
systemctl restart watchdog
```

Power over Ethernet

- English
- русский

Купить в интернет-магазине

Power over Ethernet - название, объединяющее несколько стандартов подачи питания по кабелю Ethernet.

Wireshark поддерживает так называемый Passive Power over Ethernet (Passive PoE) с напряжением в пределах номинального для контроллера. Питание передаётся по неиспользуемым парам кабеля Ethernet: "+" ("-") по паре 4-5 (синий, бело-синий), "-" ("+") по паре 7-8 (коричневый, бело-коричневый). Полярность не имеет значения.

Этот стандарт не совместим с распространёнными стандартами IEEE 802.3af и 802.3at, называемыми обычно просто Power over Ethernet.

Для подачи питания между роутером и контроллером ставится блок питания (инжектор), "добавляющий" питание в кабель Ethernet до контроллера. Желательно использовать инжектор с напряжением от 12 вольт и мощностью от 12 Вт.

Wireshark 4/5/6 может быть одновременно запитано и по Passive PoE, и через штекер/клеммники. Фактическое питание идет от источника с большим напряжением.

Примечание для контроллеров версий 2.8 и 3.5: по умолчанию устройства поставлялись без поддержки PoE. Версии с поддержкой PoE имеют надпись HLJ-6115ANL на Ethernet-разъёме. Подключение питания по Ethernet к Wireshark без поддержки PoE (с надписью HanRun на Ethernet-разъёме) приведёт к повреждению устройства.



PoE Injector



PoE Injector

Центр документации

- English
- русский

Контроллеры

Универсальные контроллеры автоматизации, работающие под управлением свободного программного обеспечения. Применяются в задачах мониторинга серверного и климатического оборудования, диспетчеризации и сбора данных с приборов учёта, в качестве основы для «умного дома» и автоматизации производств.

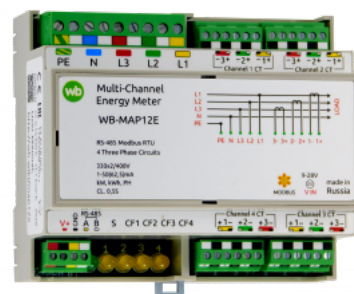
- Wiren Board 6 — универсальный контроллер для типовых задач.
- Wiren Board 7 — мощный универсальный контроллер для ресурсоёмких задач.
- Модули расширения устанавливаются внутрь корпуса контроллера, совместимы с Wiren Board 6 и Wiren Board 7.
- Модули ввода-вывода стыкуются к контроллеру Wiren Board справа через боковой разъём. Совместимы с Wiren Board 5, Wiren Board 6, Wiren Board 7.
- Поддерживаемые устройства и протоколы — стороннее оборудование, работающее с контроллером Wiren Board.
- Ответы на часто задаваемые вопросы (FAQ) — сборник готовых решений и советов, полезные ссылки
- Диагностика ошибок в работе контроллера Wiren Board — сборник советов по диагностике контроллера
- Подключение и настройка устройств по RS-485



Wiren Board 6

Счётчики электроэнергии и вольтметры

- WB-MAP12E — многоканальный счетчик электроэнергии (измерение всплесков тока и напряжения)
- WB-MAP6S — однофазный многоканальный счетчик электроэнергии
- WB-MAP3E — трехфазный счетчик электроэнергии (измерение всплесков тока и напряжения)
- WB-MAP3ET — трехфазный счетчик электроэнергии (измерение всплесков тока и напряжения) со встроенными трансформаторами
- WB-MAP3EV — трехфазный вольтметр
- WB-CT309 — сборка неразъемных трансформаторов для счетчиков MAP



WB-MAP12E

Релейные модули

О выборе модуля реле читайте в статье Рекомендации по выбору реле для нагрузки.

- WB-MR3LV/K(I), WB-MR6LV/K(I) — 3- и 6-канальные модули реле общего назначения с переключаемой группой контактов
- WB-MR3LV/S, WB-MR6LV/S — очень мощные 3- и 6-канальные модули реле с нормально открытыми контактами
- WB-MRPS6 — мощный 6-канальный модуль реле без входов
- WB-MRWL3 — очень мощный 3-канальный модуль реле
- WB-MR6C v.2 — модуль реле 6-канальный
- WB-MR6C v.3 — модуль реле 6-канальный со встроенным блоком питания
- WB-MR6C/NC — модуль реле 6-канальный с нормально-замкнутыми контактами
- WB-MR6CU v.2 — компактный модуль реле 6-канальный
- WB-MRM2-mini v.2 — компактный 2-канальный модуль реле
- WB-MRWM2 — мощный 2-канальный модуль реле с **измерением мощности**



WB-MRM-2mini

Датчики

- WB-MS v.2 — универсальный датчик температуры, влажности, освещённости, качества воздуха
- WB-MSW v.3 — датчик климата и CO2 в настенном исполнении v.3
- WB-MSW v.3 Zigbee — датчик климата и CO2 в настенном исполнении с Zigbee

- WB-MSW v.3 LoRa — датчик климата и CO2 в настенном исполнении с LoRa
- WB-MAI11 — модуль аналоговых входов

Диммеры

- WB-MRGBW-D — четырёхканальный диммер светодиодных лент
- WB-LED — четырёхканальный диммер светодиодных лент с защитой по току и четырьмя входами
- WB-AMPLED — четырёхканальный усилитель для светодиодных лент
- WB-MDM3 — трёхканальный диммер светодиодных ламп и ламп накаливания 230 В

Преобразователи интерфейсов

- WB-MIO — преобразователь интерфейса I2C (WBIO) в RS-485 с поддержкой Modbus RTU
- WB-MIO-E v.2 — преобразователь интерфейса I2C (WBIO) в RS-485 с поддержкой Modbus RTU и RS-485 (Modbus) в Ethernet с поддержкой Modbus RTU over TCP и Modbus TCP
- WB-MGE v.2 — преобразователь интерфейса RS-485 (Modbus) в Ethernet с поддержкой Modbus RTU over TCP и Modbus TCP

Сетевые карты для контроллеров холодильного оборудования

- WB-REF-U-CR — сетевая карта для контроллеров Carel BASIC(PYEZ)/EASY(PJEZ)
- WB-REF-DF-178A — сетевая карта для контроллеров Danfoss EKC 202/EKC 210
- WB-REF-DF-ERC21 — сетевая карта для контроллеров Danfoss ERC 211/ERC 213/ERC 214

Разное

- WB-MAO4 — модуль аналоговых выходов 0-10В 4-канальный
- WB-UPS v.2 — модуль бесперебойного питания на литий-полимерных аккумуляторах
- WB-MCM8 — модуль дискретных и счётных входов 8-канальный
- WB-MIR v.2 — устройство ИК-управления
- WB-M1W2 — преобразователь для термометров 1-Wire
- WB-MAI2-mini/CC — модуль измерения токового сигнала
- WB-MWAC — модуль для учета водопотребления и контроля протечек
- WB-DEMO-KIT v.3 — «Демо-чемодан»: набор интегратора, для демонстрации заказчику или самостоятельного быстрого освоения устройств Wiren Board
- Демонстрационный стенд — пример сборки демонстрационного стенда с оборудованием Wiren Board. Можно посмотреть в нашем офисе.



WB-MIR v.2

Снятые с производства устройства

- WB-DEMO-KIT v.2 — демонстрационный комплект оборудования, версия 2
- WB-DEMO-KIT v.1 — демонстрационный комплект оборудования, версия 1
- WB-UPS — модуль бесперебойного питания на литий-полимерных аккумуляторах
- WB-MR3HV, WB-MR6HV — мощные 3- и 6-канальные модули реле
- WB-MIO-E v.1 — устройство заменено WB-MIO-E v.2
- WB-MGE v.1 — устройство заменено WB-MGE v.2
- WBC-2G v.1 — модуль заменён WBC-2G v.2
- WBC-3G — модуль заменён WBC-4G
- WB-MSW2 — датчик климата и CO2 в настенном исполнении v.2
- WB-MSGR — электрохимические датчики газа WB-MSGR с встроенным реле
- WB-MDM2 — двухканальный диммер светодиодных ламп и ламп накаливания 230 В
- WB-MCM16 — модуль счетных входов 16-канальный
- WB-MRGB — диммер светодиодных лент
- WB-MRGB-D — диммер светодиодных лент (на дин-рейку)
- WB-MSW — универсальный датчик температуры, влажности, освещённости, шума в настенном исполнении v.1
- WB-MIR v1 — устройство ИК-управления
- WB-MAP12H — многоканальный счетчик электроэнергии (измерение гармонических составляющих тока и напряжения)
- WB-MAP3H — трехфазный счетчик электроэнергии (измерение гармонических составляющих тока и напряжения)



Wiren Board 4



Wiren Board NETMON-1

- WB-MR6F — модуль реле для ступенчатого управления двумя вентиляторами
- WB-MR11 — модуль реле 11-канальный
- WB-MR14 — модуль реле 14-канальный
- WB-MRM2 — модуль реле 2-канальный
- WBIO-AI-DCM-4 — модуль измерения токов и напряжения, заменён модулем WBIO-AI-DV-12
- WBE2S-R-433MHZ — модуль расширения 433 MHz. Доступен по запросу
- WB_AC_rev_E2.0 — автономный/сетевой IP-контроллер доступа со встроенным считывателем карт Mifare
- WB-MGW — преобразователь интерфейсов WB-MGW Wi-Fi — RS-485 предназначен для создания моста между сетями Wi-Fi и RS-485
- Wiren Board NETMON-2 — контроллер для автоматизации и мониторинга в 19" стойку. Состоит из Wiren Board 5 + модуль реле + модуль для «сухих контактов» + модуль резервного питания в корпусе под 19" стойку
- Wiren Board NETMON-1 — контроллер в 19" стойку. Программное обеспечение практически полностью совпадает с таковым у Wiren Board 5. Устройства отличаются набором портов и аппаратными характеристиками
- Wiren Board 5 — предыдущая модель контроллера
- Wiren Board 4 — устаревшая версия контроллера
- Wiren Board Smart Home rev. 3.5 — устаревшая версия контроллера
- Wiren Board rev. 2.8 — устаревшая версия контроллера

Устройства, протоколы и программы, с которыми может работать контроллер Wiren Board

Поддерживаемые контроллером Wiren Board протоколы, устройства и системы верхнего уровня

Поддерживаемые протоколы

| | |
|---|--|
| Опрос датчиков и работа с устройствами (в базовой комплектации) | 1-Wire • DLMS/COSEM • Modbus RTU/TCP Master • ГОСТ МЭК 61107 • СПОДЭС (ГОСТ Р 58940-2020) |
| Опрос датчиков и работа с устройствами (с помощью модулей расширения) | KNX • eBUS • OpenTherm • Z-Wave • Zigbee |
| Системы верхнего уровня | KNX • Modbus RTU/TCP Slave • MQTT • OPC UA • SNMP • Zabbix • МЭК 104 • SmartWeb |
| ПО верхнего уровня | Grafana • Home Assistant • IntraHouse • IntraSCADA • IRidium Server • MasterSCADA • Nagios • SimpLight SCADA • Rapid SCADA • SAYMON • Zabbix |
| Голосовые ассистенты и облачные решения | Сири и Apple HomeKit • Салют от SberDevices • Яндекс Алиса |

Протестированные устройства сторонних производителей

| | |
|--|--|
| Адаптеры протоколов | Шлюз ECODim DALI GW2 |
| Датчики климата | DS18B20 и клоны • Kvaдро 1WIRE-RS485 • RLDA NL-3DPAS-M • RLDA NL-1S111 • Wellpro WP3066ADAM • Даджет MT8057/MT8057S • РД MSU21 • РД MSU24 • РД MSU34+TLP • РД MSU34+THLP • Эксис ИВТМ-7 М 3 |
| Датчики уровня | ЭСКОРТ ДБ-2 |
| Диммеры | DALI • Philio PAD07-RU • Uniel UCH-M131RC/0808 • Uniel UCH-M141RC/0808 • РД DDL04R • РД DDL24 • РД DDL84R-V • РД DDM845R |
| Конвекторы | Varmann QTherm |
| Кондиционеры | Haier YCJ-A002 • Z-Wave ИК-передатчик PAR01-RU |
| Контроллеры вентиляции и климата | Mautomatics JL204C5 (Breezart 550 Lux) • GTC (General Thermo Controllers) Syberia 5.0 • SystemAir VR 300 |
| Контроллеры холодильного оборудования | Carel BASIC(PYEZ)/EASY(PJEZ) • Danfoss EKC 204A1 / AK-CC 210 • Danfoss EKC 202B • Danfoss EKC 202D • Danfoss ERC 211/ERC 213/ERC 214 • Eliwell IDPlus 974 |
| Метеостанции | Netatmo Urban Weather Station |
| Модули ввода-вывода | Wellpro WP8026ADAM • Wellpro WP8027ADAM • Wellpro WP8028ADAM • Wellpro WP9038ADAM |
| Модули реле | РД DRB88 • Rubetek TZ78 • ICP DAS tM-P3R3 • ICP DAS LC-103 • Uniel UCH-M111RX/0808 • Uniel UCH-M121RX/0808 |
| Моторы для штор/Электрокарнизы | Akko AM82 • Dooya DM35EQ • Dooya DT82 • WinDeco • Somfy SDN • SunFlower KT82TV • Somfy RS485 RTS transmitter |
| Управление двигателями (преобразователи частоты) | Vacon/Danfoss 10 • Danfoss VLT Microdrive FC51 • T13-400W-12-H |
| Счётчики воды | Пульсар • Пульсар-М • Элехант СВД-15 • Элехант СВД-20 • Счётчики с импульсным выходом |
| Счётчики тепла | Пульсар |
| Счётчики электроэнергии | CSQ PD561Z-9SY • Peacefair PZEM-016 • Eastron SDM120M • Eastron SDM220M • Меркурий 200 • Меркурий 201 • Меркурий 203.2Т • Меркурий 204 • Меркурий 206 • Меркурий 208 • Меркурий 230 • Меркурий 231 • Меркурий 234 • Меркурий 236 • Меркурий 238 • Милур 104 • Милур 105 • Милур 107 • Милур 305 • Милур 307 • Нева МТ 113 • Нева МТ 123 • Нева МТ 124 • Нева МТ 323 • Нева МТ 324 • Энергомера CE301 • Энергомера CE102M • Энергомера CE303 • Энергомера CE308 |
| Термостаты | BAC-6000 Series • BHT-6000 Series • Cityron ПУ-3 (Modbus) • Heatit Z-TEMP2 • Hessway • Siemens RDF302 |
| Увлажнители | CAREL Humisonic |
| Прочее | DIY • Shelly UNI • Tasmota • ESPHome |

Устройства с аналоговым или цифровым выходом

| | |
|---|--|
| Низковольтная нагрузка | Реле с управляющим напряжением 12-24 В • Светодиоды • Низковольтные вентиляторы • Низковольтные сигнальные лампы |
| Датчики с аналоговым выходом | Датчики температуры, давления и другие, имеющие на выходе ток или напряжение |
| Счётчики с импульсным выходом | Счётчики электроэнергии, воды, тепла и другие с импульсным выходом |
| Устройства с выходом «открытый коллектор» | Устройства с выходом «открытый коллектор» |
| Устройства с питанием 220 В | Лампы • Контакты • Другое оборудование, питающееся от 220 В |

Работа с GPIO

ВНИМАНИЕ: статья рассчитана на разработчиков или опытных пользователей и даёт общие рекомендации того, как использовать gpio в обход официального ПО WirenBoard.

Если вам нужно работать напрямую с gpio, то мы рекомендуем делать это через драйвер `wb-mqtt-gpio` (<https://github.com/wirenboard/wb-homa-gpio>).

Описание доступных ножек gpio для конкретной ревизии контроллера можете посмотреть в статье [GPIO](#).

Contents

Меры предосторожности

Именованние gpio

- Вычисление номера gpio

Работа из userspace

- Bash

 - Интерфейс sysfs

 - Чтение и запись

 - Работа с прерываниями

 - Работа через chardev

- Python

- Прямое обращение через память процессора

Работа в ядре Linux

- Рекомендации по Device Tree

 - Пример device-tree node

Меры предосторожности

- Убедитесь, что вашу задачу нельзя решить стандартными средствами программного обеспечения Wiren Board.
- Все порты Wiren Board, в том числе и GPIO, работают с напряжением 3.3V.
- Подключение сигнала с напряжением большим 3.3V к ножке GPIO грозит выходом из строя процессорного модуля.

В случае необходимости подключения устройств, работающих с более высоким напряжением, необходимо использовать схемы согласования или подключать (для 5V) через резистор в 20 кОм и более.

Именованние gpio

К сожалению, четкого стандарта по именованию gpio не существует, но при работе с контроллерами WirenBoard стоит придерживаться следующих правил:

- выводы gpio сгруппированы по банкам (*banks*; эквивалентно *gpiochips*)
- каждый банк содержит 32 gpio. Нумерация банков начинается с 0.

Вычисление номера gpio

Для управления ножкой gpio нужно знать её номер. В рассматриваемых примерах будем работать с gpio `A1_IN` контроллера WB6.7 (номер: 109; *gpiochip 3, offset 13*): Вычислим банк gpio и offset, зная номер (109):

```
# Поделим 109 на 32. Целая часть – номер банка, остаток - offset:  
109.0 / 32.0 = 3, остаток 13
```

То же самое справедливо и наоборот. Зная банк и offset (3 и 13, соответственно), можно вычислить номер gpio:

```
# Умножим номер банка на 32 и прибавим offset:  
3 * 32 + 13 = 109
```

Работа из userspace

Перед началом работы из userspace, необходимо убедиться, в том, что нужный gpio — свободен. Для этого можно посмотреть на вывод команды

```
cat /sys/kernel/debug/gpio
```

В выводе команды видим примерно следующее:

```
gpiochip0: GPIOs 0-31, parent: platform/209c000.gpio, 209c000.gpio:
gpio-0 ( ) |sysfs ( ) in hi IRQ
gpio-10 ( ) |? ( ) in lo
gpio-11 ( ) |wl ( ) in hi
gpio-13 ( ) |wl strong pullup ( ) out lo
gpio-26 ( ) |sysfs ( ) out lo
gpio-27 ( ) |sysfs ( ) out hi
```

Это значит, что gpio 0, 26 и 27 уже экспортированы в sysfs и доступны для управления. Gpio 11 и 13 заняты ядерным драйвером `openwire` и недоступны для использования. Остальные gpio банка 0 — свободны.

Если нужный gpio — занят, то можно остановить драйвер:

```
lsmod | grep wl # узнаем название драйвера
rmmod wl_gpio # выгружаем драйвер, название которого узнали
```

ВНИМАНИЕ: остановка драйверов может привести к неожиданному поведению контроллера. Теперь нужный gpio свободен до следующей перезагрузки.

Bash

В настоящий момент, для работы с gpio в userspace доступны 2 интерфейса: `sysfs` и `chardev` (начиная с версии ядра 4.8).

Различия между `chardev` и `sysfs` хорошо описаны в этой статье (<https://embeddedbits.org/new-linux-kernel-gpio-user-space-interfa>ce/). Sysfs имеет статус deprecated, поэтому, по возможности, стоит работать через chardev.

Интерфейс sysfs

Для работы через sysfs с определённым GPIO его надо экспортировать:

Здесь и далее N — номер gpio

```
echo N > /sys/class/gpio/export
```

Экспортированные gpio появляются в каталоге `/sys/class/gpio`:

```
~# ls -l /sys/class/gpio/
export
gpio32
gpiochip0
gpiochip120
gpiochip32
gpiochip64
unexport
```

В директории `/sys/class/gpioN` теперь находятся файлы для работы с GPIO (где N — номер GPIO, как и было сказано ранее):

```
~# ls -l /sys/class/gpio/gpioN/
active_low
device
direction
edge
power
subsystem
uevent
value
```

Установка направления GPIO (ввод/вывод) производится с помощью записи в файл `direction`

```
echo in > /sys/class/gpio/gpioN/direction # установим GPIO номер N на ввод
echo out > /sys/class/gpio/gpioN/direction # установим GPIO номер N на вывод
```

Чтение и установка значения GPIO производится с помощью файла `value`.

Чтение и запись

Чтение:

```
echo in > /sys/class/gpio/gpioN/direction # установим GPIO номер N на ввод
cat /sys/class/gpio/gpioN/value # вернёт 1 или 0
```

Запись:

```
echo out > /sys/class/gpio/gpioN/direction # установим GPIO номер N на вывод
echo 0 > /sys/class/gpio/gpioN/value # установим логический 0 (низкое напряжение) на GPIO номер N
echo 1 > /sys/class/gpio/gpioN/value # установим логический 1 (высокое напряжение) на GPIO номер N
```

Пример:

1. Находим номер GPIO, соответствующий вашей версии контролера нужному клеммнику в таблице WB2.8. Для клеммника номер 2 в версии 2.8 это GPIO 32.
2. Экспортируем GPIO в sysfs

```
echo 32 > /sys/class/gpio/export
```

3. Устанавливаем GPIO в режим вывода для управления транзистором. Это обязательно, т.к. GPIO может находиться в режиме ввода и иметь высокий импеданс, оставляя транзистор в неопределённом состоянии.

```
echo out > /sys/class/gpio/gpio32/direction
```

4. Открываем транзистор, подавая логический высокий уровень на затвор:

```
echo 1 > /sys/class/gpio/gpio32/value
```

5. Закрываем транзистор, подавая логический ноль на затвор:

```
echo 0 > /sys/class/gpio/gpio32/value
```

Работа с прерываниями

Через интерфейс sysfs можно запросить прерывания по изменению состояния процессора.

Установка прерывания производится путём записи значения в файл "edge". Значения могут быть:

- none — отключить прерывание
- rising — включить прерывание по нисходящему фронту
- falling — включить прерывание по восходящему фронту
- both — включить прерывание по обоим фронтам.

Пример работы с прерываниями:

```
~# echo 3 > /sys/class/gpio/export # экспортируем GPIO номер 3 (TB10 на WB3.3)
~# cat /sys/class/gpio/gpio3/edge # проверяем состояние прерывания
none
~# echo falling > /sys/class/gpio/gpio3/edge # устанавливаем прерывание по нисходящему фронту
~# cat /proc/interrupts | grep gpiolib # прерывание появилось в списке. 26 - внутренний номер прерывания, 0 - количество событий
26:      0      gpio-mxs      3      gpiolib
~# cat /proc/interrupts | grep gpiolib # после нескольких событий, 76 - количество событий
26:      76      gpio-mxs      3      gpiolib
```

Прерывания можно ловить из userspace с помощью системного вызова `epoll()` и `select()` на файл `value`. Пример работы см. [1] (<https://github.com/contactless/wiegand-linux-sysfs>)

См. также [elinux.org](http://elinux.org/GPIO) (<http://elinux.org/GPIO>)

Работа через chardev

Представленный в ядре 4.8 интерфейс chardev имеет C/Python библиотеку `libgpiod` и userspace-утилиты для работы с gpio. Исходный код библиотеки и документация доступны в репозитории `libgpiod` (<https://github.com/brgl/libgpiod>).

Утилиты распространяются в составе debian-пакетов `gpiod` и `libgpiod-dev` для debian buster и новее. К сожалению, для stretch пакетов в официальных репозиториях нет.

Если нужно установить `libgpiod` в debian stretch, можно воспользоваться сторонними репозиториями (например, этим (<https://github.com/rcn-ee/repos>)). **Используйте сторонние репозитории на свой страх и риск; компания WirenBoard не контролирует их содержимое.**

Для работы с gpio из bash в пакете `gpiod` поставляются следующие утилиты:

- `gpiodetect` — информация обо всех банках gpio в системе
- `gpioinfo` — подробная информация обо всех линиях gpio определённого банка
- `gpioget <чип> <линия>` — возвращает значение определённого gpio
- `gpioset <чип> <линия1>=<значение1> <линия2>=<значение2>` — устанавливает состояние на определенные линии gpio
- `gpiofind <название>` — возвращает номер gpio
- `gpiomon` — отслеживание событий gpio

Примеры использования `gpiod` можно посмотреть в [2] (<https://www.acmesystems.it/gpiod>) и [3] (<https://github.com/brgl/libgpiod>)

Python

Для управления gpio из python был написан модуль

```
wb_common.gpio
```

Модуль представляет собой обёртку вокруг sysfs. Исходный код доступен на нашем github. (https://github.com/wirenboard/wb-common/blob/master/wb_common/gpio.py)

Модуль позволяет работать с gpio в синхронном и асинхронном (с регистрацией коллбэков) режимах.

Прямое обращение через память процессора

Этот метод настоятельно НЕ РЕКОМЕНДУЕТСЯ для использования без достаточных оснований. Для работы из C/C++ стоит использовать работу через файлы в sysfs или chardev, как описано в предыдущих разделах.

Управлять GPIO можно с помощью прямого доступа к регистрам процессора, в обход Linux, через интерфейс /dev/mem. При этом, по сравнению с работой через sysfs минимизируются накладные расходы. Этот метод можно использовать, если вам необходим очень быстрый доступ к GPIO, например bitbang протоколов или ШИМ. Стоит иметь в виду, что планировщик процессоров всё ещё может вносить в работу программы значительные задержки. Рекомендуется выносить критичные ко времени задачи в ядро.

См. [4] (<http://olimex.wordpress.com/2012/09/11/imx233-olinuxino-gpios-faster-and-faster/>) , [5] (<https://github.com/OLIMEX/OLINUXINO/blob/master/SOFTWARE/iMX233/gpio-mmap.h>)

Работа в ядре Linux

Ознакомиться с ядром Linux, использующимся в контроллерах WirenBoard можно в нашем репозитории ядра (<https://github.com/wirenboard/linux>).

Рекомендации по Device Tree

Device-tree, использующиеся на контроллерах WirenBoard, доступны в репозитории ядра. Разные аппаратные ревизии контроллера используют разные dts (например, dts для WB6.X можно найти здесь (<https://github.com/wirenboard/linux/blob/dev/v4.9.x/arch/arm/boot/dts/imx6ul-wirenboard61.dts>))

Указывать GPIO в Device Tree необходимо для настройки работы GPIO в режиме программного SPI, I2C, для использования GPIO в качестве источника прерываний и т.д. Так, например, на пин 10@UEXT1 (CS) и пины 5@UEXT2 (SCL), 6@UEXT2 (SDA), 10@UEXT2 (CS) выведены линии GPIO процессора. Их можно сконфигурировать для использования, например, в качестве chip-select для SPI или в качестве I2C.

GPIO процессора и периферийных устройств разбиты на банки (gpiochip). GPIO процессора разбиты на 3 банка по 32 GPIO: gpio0, gpio1, gpio2. Адресация GPIO в Device Tree происходит по номеру банка и номеру GPIO *внутри* банка.

Пример device-tree node

Определим сигнал 6@UEXT2 (SDA) в качестве источника прерываний для драйвера mrf24j40. Согласно таблице Список GPIO, сигнал соответствует GPIO 53 процессора. 53 принадлежит второму банку gpio (от 32 до 63). Номер GPIO внутри банка 53-32=21 :

```
6lowpan@0 {
    compatible = "microchip,mrf24j40";
    spi-max-frequency = <1000000>;
    reg = <6>;
    interrupt-parent = <&gpio1>;
    interrupts = <21 0>;
};
```

Wiren Board 5: Восстановление пароля пользователя root

- English
- русский

```
U-Boot 2015.07-02364-gb119240 (Jun 10 2016 - 21:41:13 +0300)
CPU: Freescale i.MX28 rev1.2 at 454 MHz
BOOT: SSP/SD/MMC #0, 3V3
DRAM: 128 MiB
MMC: M0S MMC: 0
MMC0: Command 0 timeout (status 0xf03c4020)
MMC0: Command 55 timeout (status 0xf02c4020)
In: serial
Out: serial
Err: serial
Net: FEC0 [PRIME], FEC1
Hit any key to stop autoboot: 1
```

Остановка автозагрузки uboot

```
=> setenv optargs ro rootwait fixrtc single init=/bin/sh
=> run bootcmd
MMC0: Command 0 timeout (status 0xf03c4020)
MMC0: Command 55 timeout (status 0xf02c4020)
switch to partitions #0, OK
mmc0(part 0) is current device
MMC0: Command 0 timeout (status 0xf03c4020)
MMC0: Command 55 timeout (status 0xf02c4020)
SD/MMC found on device 0
43 bytes read in 189 ms (0 Bytes/s)
Importing environment from mmc (uEnv.txt)...
Checking if uenvcmd is set ...
Running default loadimage ...
```

Ввод команд

Процедура восстановления пароля root на Wiren Board 5

1. Подключаем USB-UART переходник контроллеру. Подробно подключение описано на странице Debug UART. Обратите внимание, что в последних версиях контроллера отладочный порт UART представлен в виде microUSB-разъема, так что дополнительного преобразователя USB-UART не требуется.
2. Включаем питание контроллера и в UART-консоли быстро нажимаем любую клавишу.
3. **Очень быстро** последовательно вводим команды (их лучше заранее скопировать в буфер обмена):
setenv optargs ro rootwait fixrtc single init=/bin/sh
run bootcmd
Если команды вводить медленно (вручную), то прерванный процесс загрузки продолжится.
4. Загружаемся и получаем системное приглашение #
5. Монтируем корневую файловую систему
mount -n -o remount,rw /dev/mmcblk0p2 /
и получаем сообщение EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
6. Монтируем раздел с резервной копии конфигурации:
mount /dev/mmcblk0p6 /mnt/data
и получаем сообщение EXT4-fs (mmcblk0p6): mounted filesystem with ordered data mode. Opts: (null)
7. Вводим команду passwd и меняем пароль. Успешное изменение пароля: passwd: password updated successfully
8. Копируем новый пароль в раздел с резервной копии конфигурации:
cp /etc/shadow /mnt/data/etc
9. Отмонтируем всё:
sync
umount /mnt/data
umount /
10. Выключаем и снова включаем питание. Пароль изменен.

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

- Privacy policy
- About Wiren Board
- Disclaimers
-