How to diagnose

Диагностика ошибок в работе контроллера Wiren Board

Руководство по эксплуатации

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

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

Содержание

How to diagnose
journalctl — утилита просмотра системного журнала
Обновление прошивки контроллера Wiren Board
Просмотр файлов контроллера с компьютера
MQTT
Драйвер wb-mqtt-serial
Watchdog
Движок правил wb-rules
Программное обеспечение Wiren Board
Как зайти на контроллер Wiren Board по SSH
Debug UART
Утилита «modbus_client»
Модули ввода-вывода
Центр документации
Ответы на часто задаваемые вопросы

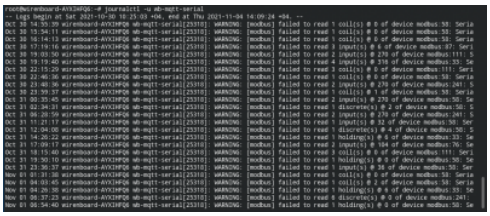
How to diagnose

Contents

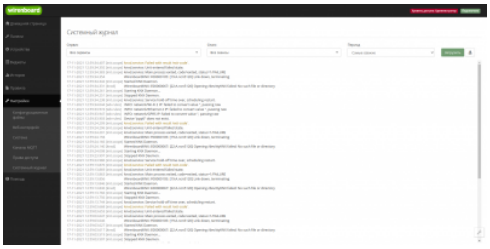
Введение
Основы
- Как устроено ПО контроллера
- Полезные команды
  - Информация о контроллере
  - Управление сервисами

Примеры типовых неисправностей
- Serial-устройства работают нестабильно
- Не работают модули WBIO
- Не открывается страница конфигурации устройств
- Контроллер перезагружается
- Контроллер не включается

Полезные ссылки



Просмотр системного журнала с помощью journalctl



Просмотр системного журнала в веб-интерфейсе контроллера Wiren Board

Введение

Если в работе контроллера Wiren Board возникают ошибки — найти причину поможет анализ системного журнала.

В контроллерах Wiren Board используется системный сервис журналов journald, который обрабатывает события на запись в лог и сохраняет их в бинарные файлы. По умолчанию настроена очистка старых сообщений журнала, которая не позволит ему занять всё свободное пространство.

Читать журналы можно двумя способами:

- из консоли с помощью journalctl.
- из веб-интерфейса с помощью инструмента Системный журнал.

Цель диагностики — локализовать и устранить неисправность, общий алгоритм:

1. Сформулируйте проблему: контроллер перезагружается или не включается, Modbus-устройство работает нестабильно, не работает 4G-модем, не работает веб-интерфейс и т.д.
2. Смотрите, есть ли в системном журнале ошибки. Прочитайте разделы Основы и Примеры типовых неисправностей — это даст вам понимание того, как всё устроено и поможет понять, что именно искать.
3. Если нашли в журнале ошибки — внимательно прочитайте их, они дадут понимание того, что происходит и кто виноват. Дополнительно поищите упоминание проблемы на портале техподдержки, возможно, кто-то уже с ней сталкивался и есть готовое решение.
4. Если из текста ошибки и результатов поиска на форуме непонятно, что происходит — проверьте, что вы используете свежую версию ПО. Если это не так — обновите прошивку контроллера. Подробнее читайте в статье Обновление прошивки контроллера.

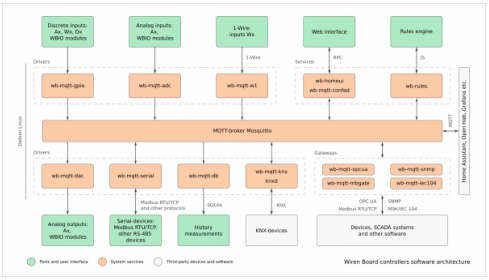
В случае, если вы не смогли самостоятельно определить и устранить причину проблемы, создайте тему на форуме техподдержки (https://support.wirenboard.com): подробно опишите проблему и приложите записи системного журнала или архив с диагностической информацией. Рекомендуем перед созданием темы прочитать Советы по оформлению сообщений (https://support.wirenboard.com/faq).

Обмен файлами с контроллером можно организовать по протоколу SFTP(SSH File Transfer Protocol), смотрите инструкцию в статье Просмотр файлов контроллера с компьютера.

Основы

Как устроено ПО контроллера

В основе программной архитектуры Wiren Board — MQTT-брокер, в который пишут информацию различные сервисы wb:



Через MQTT работают драйверы внутренних функций, внешних устройств, веб-интерфейс, система правил

- wb-mqtt-adc — драйвер аналоговых входов
- wb-mqtt-dac — драйвер аналоговых выходов
- wb-mqtt-confed — конфигуратор в WebUI
- wb-mqtt-db — хранение архива данных
- wb-mqtt-gpio — драйвер для работы с GPIO
- wb-mqtt-knx — мост KNX
- wb-mqtt-mbgate — шлюз MQTT-ModbusRTU/TCP
- wb-mqtt-opcua — шлюз MQTT-OPC UA
- wb-mqtt-serial — драйвер для serial-устройств
- wb-mqtt-w1 — драйвер для 1-Wire

Кроме этого, есть сервисы, которые реализуют дополнительные функции:

- watchdog — сторожевой таймер
- wb-hwconf-manager — драйвер модулей расширения
- wb-rules — движок правил

Подробнее читайте в статье Программное обеспечение Wiren Board.

На контроллер Wiren Board можно установить ПО сторонних разработчиков, которое может сбоить и блокировать нормальную работу контроллера — помните об этом. В большинстве случаев стороннее программное обеспечение так же оставляет записи в системном журнале.

Полезные команды

Команды ниже выполняются в консоли контроллера, подключиться к которой можно через SSH или отладочный порт.

Информация о контроллере

Посмотреть, когда был запущен контроллер:

```
# who -b
         system boot  Nov  4 13:43
```

Узнать, когда контроллер перезагружался:

```
# last -x reboot
reboot   system boot  5.10.35-wb6      Thu Nov  4 13:43   still running
reboot   system boot  5.10.35-wb6      Thu Nov  4 10:39   still running
```

Посмотреть текущую версию прошивки контроллера:

```
# wb-release
Wirenboard release wb-2108 (as stable), target wb6/stretch
```

Узнать версию пакета можно с помощью команды `apt policy <имя пакета>`, например, узнаем версию *wb-mqtt-serial*:

```
# apt policy wb-mqtt-serial
wb-mqtt-serial:
  Installed: 2.22.1-wb5
  Candidate: 2.22.1-wb5
  Version table:
 *** 2.22.1-wb5 990
        990 http://deb.wirenboard.com/wb6/stretch stable/main armhf Packages
        100 /var/lib/dpkg/status
```

Узнать, сколько свободного места на eMMC:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       976M  581M  329M  64% /
devtmpfs        240M     0  240M   0% /dev
tmpfs           248M     0  248M   0% /dev/shm
tmpfs           248M  584K  248M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           248M     0  248M   0% /sys/fs/cgroup
/dev/mmcblk0p6  4.7G  1.1G  3.4G  24% /mnt/data
tmpfs            50M     0   50M   0% /run/user/0
```

А так можно узнать, сколько занимают места подкаталоги:

```
# du -h --max-depth=1
6.9M    ./.cache
12K     ./.ssh
46M     ./.npm
328M    ./zigbee2mqtt
20K     ./.config
16K     ./.local
4.0K    ./.nano
32K     ./tmp
381M    .
```

Посмотреть нагрузку на процессор и оперативную память в разрезе загруженных программ:

```
# top
top - 08:52:05 up 16:32,  1 user,  load average: 0.36, 0.54, 0.60
Tasks:  96 total,   3 running,  68 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.9 us,  3.3 sy,  0.0 ni, 88.5 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem :   507820 total,    33124 free,   130160 used,   344536 buff/cache
KiB Swap:   262140 total,   262140 free,        0 used.   398184 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1119 root      20   0  913636  22576  11176 S  2.3  4.4  36:08.75 wb-rules
 3642 root      20   0   59272   5408   4812 S  1.6  1.1  12:52.91 wb-mqtt-adc
14919 root      20   0    5396   2380   1892 R  1.6  0.5   0:00.55 top
  349 mosquit+  20   0    8176   5008   3912 S  1.3  1.0  11:31.22 mosquitto
    1 root      20   0   25728   5060   3704 S  0.7  1.0   3:14.82 systemd
 1129 root      20   0   62212   8256   5368 S  0.7  1.6   7:12.36 main
 4262 root      20   0   59320   6180   5580 S  0.7  1.2   4:20.00 wb-mqtt-gpio
…
```

Посмотреть, какое ядро загружено:

```
# uname -a
Linux wirenboard-AYXIHFQ6 5.10.35-wb6 #1 Thu Sep 30 00:33:57 UTC 2021 armv7l GNU/Linux
```

Управление сервисами

Проверить статус сервиса и посмотреть его последние 10 сообщений в системном журнале можно командой `systemctl status <имя сервиса>`:

```
# 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: active (running) since Mon 2021-11-08 22:21:59 +04; 12h ago
 Main PID: 8180 (wb-mqtt-serial)
   CGroup: /system.slice/wb-mqtt-serial.service
           └─8180 /usr/bin/wb-mqtt-serial

Nov 09 02:29:54 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 6 coil(s) @ 0 of device modbus:241: Serial
protocol error: request timed out
Nov 09 02:46:10 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 2 input(s) @ 270 of device modbus:58: Serial
protocol error: request timed out
Nov 09 02:46:41 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 1 coil(s) @ 2 of device modbus:58: Serial
protocol error: request timed out
Nov 09 02:53:57 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 6 discrete(s) @ 0 of device modbus:241: Serial
protocol error: request timed out
Nov 09 03:54:33 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 6 discrete(s) @ 0 of device modbus:241: Serial
protocol error: request timed out
Nov 09 05:39:59 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 3 discrete(s) @ 0 of device modbus:111: Serial
protocol error: request timed out
Nov 09 06:50:47 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 1 holding(s) @ 0 of device modbus:58: Serial
protocol error: request timed out
Nov 09 08:47:36 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 2 input(s) @ 270 of device modbus:58: Serial
protocol error: request timed out
Nov 09 10:11:34 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 1 discrete(s) @ 4 of device modbus:58: Serial
protocol error: request timed out
Nov 09 10:12:15 wirenboard-AYXIHFQ6 wb-mqtt-serial[8180]: WARNING: [modbus] failed to read 2 input(s) @ 270 of device modbus:58: Serial
protocol error: request timed out
```

Получить список запущенных сервисов и их статусы:

```
systemctl list-units --type=service
```

Управление сервисом:

```
systemctl start <имя сервиса>
systemctl stop <имя сервиса>
systemctl restart <имя сервиса>
```

Включить или выключить автозапуск сервиса:

```
systemctl disable <имя сервиса>
systemctl enable <имя сервиса>
```

Примеры типовых неисправностей

Serial-устройства работают нестабильно

Узнать, с каким именно устройством проблемы, можно:

- в веб-интерфейсе контроллера, на вкладке *Devices* — каналы такого устройства будут окрашены красным;
- в сообщениях драйвера *wb-mqtt-serial* — ищите ошибки обмена error и warning. Если сообщения об ошибках обмена возникают периодически — повод проверить физическое подключение устройств.

Если ошибок обмена в сообщениях драйвера нет, но устройство не работает — включите вывод отладочных сообщений. Как это сделать, смотрите в описании драйвера wb-mqtt-serial.

После того как вы определились с проблемным устройством — подключите его коротким проводом на отдельный порт RS-485, настройте и добейтесь стабильной работы: проверьте параметры подключения, снизьте скорость обмена, убедитесь, что в устройстве выставлен верный стоп-бит.

Диагностика устройств, которые работают по протоколу Mosbus TCP или Modbus Over TCP почти не отличается от тех, что подключены к контроллеру по RS-485. Разница лишь в том, что у вас добавляется ещё одно звено — локальная сеть, через которую подключены устройства.

Частые симптомы и методы диагностики			
Порядок проверки	**Симптомы**	**Возможная причина**	**Диагностика**
1	Устройство не работает или работает нестабильно	Неверные настройки подключения	Проверьте, что настройки порта RS-485 и устройства одинаковые. Если устройство работает нестабильно, обратите внимание на стоп-бит
2	Устройство не работает или работает нестабильно. В системном журнале периодически возникают ошибки обмена	Проблемы с шиной RS-485: плохой контакт, обрывы или наводки на шине	Подключите устройство коротким проводом напрямую к контроллеру. Если проблема ушла — проверяйте шину RS-485 на отсутствие физических повреждений и соответствие рекомендациям: RS-485:Физическое подключение
3	Устройство работает нестабильно, в системном журнале есть ошибки таймаута и *invalid CRC*	На шине два устройства с одинаковыми адресами	▪ Физически отключите проблемное устройство от шины, запомните его адрес. ▪ Считайте с помощью утилиты modbus_client один из регистров по тому же адресу. Для устройств Wiren Board удобно считывать регистр 128. ▪ Если ответ пришёл — на линии есть двойник. Смените устройствам адреса на свободные.
4	Устройство стороннего производителя работает нестабильно	Проблемы в устройстве	▪ Подключите устройство на отдельный порт и добейтесь стабильной работы. ▪ Верните устройство на шину к другим устройствам. ▪ Если проблемы вернулись — производитель устройства не полностью реализовал Modbus-протокол. Решение: использовать устройство на отдельном порту или не использовать его совсем.

Не работают модули WBIO

Сперва проверьте правильность подключения и настройки.

Если модули правильно подключены и настроены, но не работают:

1. В настройках контроллера удалите все модули ввода-вывода.
2. Выключите контроллер и физически отключите от него все модули, кроме одного.
3. Включите контроллер и настройте оставшийся модуль по инструкции.
4. Если модуль появился в веб-интерфейсе на странице *Devices* — проверьте его работу.

Если подключённый модуль работает без проблем — добавляйте новые модули по одному и не забывайте конфигурировать их в веб-интерфейсе. Дальше у вас или все модули заработают, или вы найдёте тот, который влияет на работу других.

Не открывается страница конфигурации устройств

Перед началом диагностики, попробуйте просто перезагрузить страницу с очисткой кэша браузера — часто помогает. Обычно это можно сделать нажатием клавиш `Ctrl`+`Shift`+`R` или `Ctrl`+`F5` — комбинация зависит от вашего браузера.

Если это не помогло:

- ищите в логах сообщения от *wb-mqtt-serial* и *wb-mqtt-confed*. Особенно интересны сообщения типов *error* и *warning*.
- проверьте файл конфигурации и шаблоны по инструкции.

Контроллер перезагружается

Контроллер может перезагружаться по ряду причин:

- Нестабильное питание — просадки напряжения питания ниже допустимого значения могут вызвать перезагрузку.
- Нехватка места на eMMC.
- Зависание программ и сервисов — сработает watchdog, которые перезагрузит контроллер.
- Перезагрузка вызвана пользователем, например, командой `shutdown -r now`.

Сперва стоит проверить качество питания: уровень напряжения, отсутствие «просадок». Попробуйте подключить контроллер к другому блоку питания.

Если питание стабильно, то причину перезагрузки ищите в сообщениях watchdog и ядра ОС Linux (dmesg). Если контроллер перезагружается в цикле и вы не можете попасть в консоль, попробуйте отключить watchdog.

Контроллер не включается

Здесь нужно определиться, что происходит — подайте питание на контроллер, включите его кнопкой и следите за индикатором:

- прошла минута, но индикатор не загорается — аппаратная проблема: нет питания, неисправно «железо» контроллера. Если питание в норме и контроллер на гарантии — обратитесь в техподдержку для замены или ремонта.
- индикатор загорается, но даже через две-три минуты остаётся красного или оранжевого цвета — не загружается ОС: подключитесь к отладочному порту контроллера, перезагрузите его и смотрите сообщения в консоли.
- индикатор загорается и спустя пару минут начинает мигать зелёным — ОС контроллера загрузилась: попробуйте подключиться к нему по SSH. Если это удалось, точнее сформулируйте неисправность: не заходит в веб-интерфейс, не работают подключённые устройства и т.п. Потом начинайте диагностику.

Описание состояний индикатора контроллера смотрите в документации. Если у вас не получилось выяснить причину поломки, создайте тему на портале технической поддержки (https://support.wirenboard.com) и приложите всю собранную информацию.

Полезные ссылки

- Документация устройств Wiren Board — описание устройств, схемы подключения, инструкции по обновлению и т.п.
- Портал технической поддержки (https://support.wirenboard.com) — техподдержка и помощь сообщества.
- Ответы на часто задаваемые вопросы (FAQ) — сборник полезных советов и ссылок.
- Онлайн-переводчик от Google (https://translate.google.ru/) — если у вас трудности с переводом сообщений журнала, воспользуйтесь переводчиком.

journalctl — утилита просмотра системного журнала

Contents

Описание

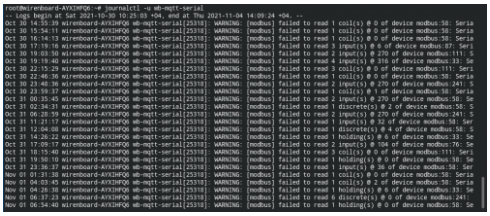
Архив

Просмотр журнала

Фильтрация результатов

Сохранение в текстовый файл

Занимаемое логами место



Просмотр системного журнала с помощью journalctl

Описание

journalctl — это консольная утилита для просмотра системного журнала ОС Linux, поэтому перед вводом команд подключитесь к контроллеру по SSH или отладочный порт.

Здесь приведены примеры, которые решают большинство задач. Полный список параметров смотрите командой `journalctl --help` и в документации на утилиту (https://manpages.debian.org/bullseye/systemd/journalctl.1.en.html).

Перемещаться по выводу утилиты можно с помощью клавиш:

- `q` — выйти из просмотра.
- `↑` — вверх на одну строку.
- `↓` — вниз на одну строку.
- `b` — вверх на одну страницу
- `Пробел` — вниз на одну страницу
- `g` — перейти на первую строку
- `/` — поиск по журналу
- `n` — найти следующее вхождение
- `N` — найти предыдущее вхождение

Архив

journald при каждой загрузке создаёт новый журнал, а старый закрывает. Список доступных журналов можно посмотреть командой:

```
# journalctl --list-boots
-2 301af672ad2b400fa9c6562a3403d179 Sat 2021-10-30 10:25:03 +04—Thu 2021-11-04 10:39:39 +04
-1 1238af7b5cfb4dd9a10bc54a1dd63067 Thu 2021-11-04 10:39:53 +04—Thu 2021-11-04 13:42:19 +04
 0 cc99bb41d7524321a70bddda34c1cceb Thu 2021-11-04 13:43:31 +04—Thu 2021-11-04 14:15:20 +04
```

По умолчанию journalctl выводит сообщения из всех доступных журналов, но вы можете сократить выборку, для этого укажите номер журнала в параметре `-b`:

```
journalctl -b -1
```

Просмотр журнала

Вывести все сообщения:

```
journalctl
```

Выводить новые сообщения:

```
journalctl -f
```

Открыть журнал и перемотать его к последней записи:

```
journalctl -e
```

Вывести последние 5 строк журнала:

```
journalctl -n 5
```

Фильтрация результатов

Лог последней загрузки операционной системы:

```
journalctl -b
```

Посмотреть сообщения ядра (dmesg):

```
journalctl -k
```

Посмотреть журнал определённого сервиса, например, wb-mqtt-serial:

```
journalctl -u wb-mqtt-serial
```

Просмотр сообщений от определённого сервиса в реальном времени:

```
journalctl -u wb-mqtt-serial -f
```

Фильтр по времени:

- записи за последние сутки

  ```
  journalctl --since -1d
  ```

  можно использовать: w — недели, d — дни, h — часы, m — минуты

- записи с определённого времени:

  ```
  journalctl --since "2020-02-13 07:00:00"
  ```

- записи за период

  ```
  journalctl --since "2020-02-13 07:00:00" --until "2020-02-14 07:00:00"
  ```

- записи за последний час

  ```
  journalctl --since "1 hour ago"
  ```

Вывод команды можно фильтровать по уровню важности сообщений:

- 0: emergency — чрезвычайная ситуация
- 1: alerts — предупреждения, требуется вмешательство

- 2: critical — критическое состояние
- 3: errors — ошибки
- 4: warning — предупреждения
- 5: notice — уведомления
- 6: info — информационные сообщения
- 7: debug — отладочные сообщения

Например, чтобы вывести все ошибки:

```
journalctl -p 3
```

Параметры можно комбинировать, например:

- показать предупреждения, записанные драйвером wb-mqtt-serial с момента последней загрузки:

  ```
  journalctl -b 0 -p 4 -u wb-mqtt-serial
  ```

- показать ошибки, записанные драйвером wb-mqtt-serial за последний час:

  ```
  journalctl --since "1 hour ago" -p 3 -u wb-mqtt-serial
  ```

Сохранение в текстовый файл

Вывод утилиты journalctl можно сохранить в файл, для этого добавьте в конец команды `>> filename.txt`, например, сохраним в файл сообщения драйвера wb-mqtt-serial:

```
journalctl -u wb-mqtt-serial >> /tmp/log-file.txt
```

Так как вывод команды будет сохранён в файл `/tmp/log-file.txt`, то на консоли вы ничего не увидите.

О том, как скопировать файл с контроллера на компьютер, читайте в статье Просмотр файлов контроллера с компьютера.

Занимаемое логами место

Посмотреть занимаемое системным журналом место:

```
journalctl --disk-usage
```

Удалить все сообщения, старше 1 недели:

```
journalctl --vacuum-time=1weeks
```

Удалить все сообщения таким образом, чтобы журнал занял 100 Мбайт:

```
journalctl --vacuum-size=100M
```

Настроить параметры ведения системного журнала можно в файле `/etc/systemd/journald.conf`.

Обновление прошивки контроллера Wiren Board

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.

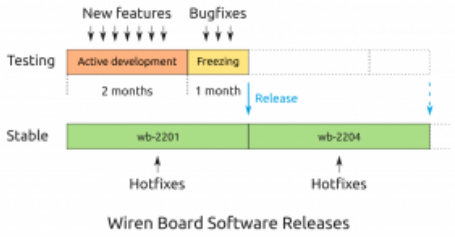
Контроллеры Wiren Board 4 и старее прошиваются через карту Micro-SD.

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.
- **Testing** — тестовый релиз: свежие версии пакетов с новыми функциями, а так же исправленными и новыми ошибками.

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



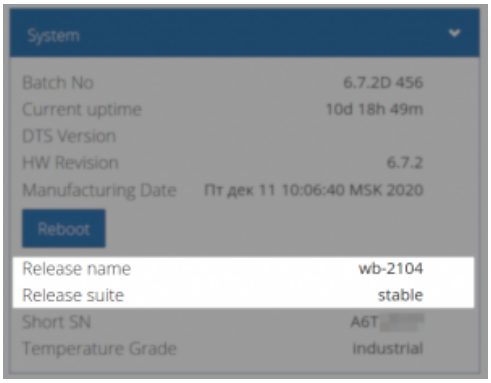
Релизный цикл ПО Wiren Board

Журналы изменений в релизах	
Ветка	**Имя**
stable	wb-2204 (апрель-май) • 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.



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

Переключение между релизами

Между релизами можно переключаться, так же можно заморозить ПО контроллера на определённом релизе — это может быть полезно на ответственных инсталляциях.

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
```

- Чтобы заморозить релиз и отказаться от новых функций, укажите версию релиза, например:

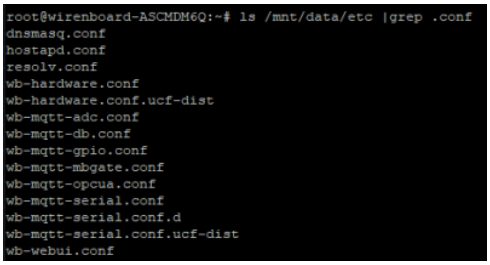
```
wb-release -t wb-2104
```

После смены релиза рекомендуем перезагрузить контроллер на случай, если обновилось ядро.

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:



Файлы настроек, которые сохраняются при обновлении через fit-файл

- 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`.

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, просто скопируйте содержимое `/mnt/data` 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/) или в консоли контроллера:

1. Подключитесь к контроллеру через SSH.
2. Выполните команду:

```
apt update
```

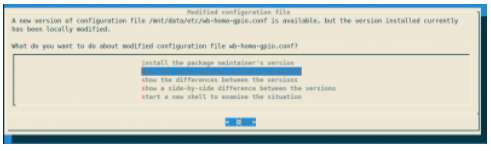
3. Если есть пакеты для обновления, то можете посмотреть их список:

```
apt list --upgradable
```

Обновление

В консоли через apt

> Рекомендуем обновлять прошивку контроллера через apt: будут сохранены настройки, ваши файлы и установленное ПО.



Окно *Modified configuration Files*

Apt — 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) apt-репозитория Wiren Board.

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.

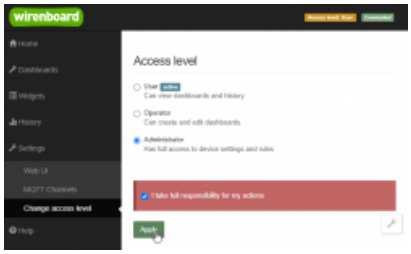
Так же в консоль будут выводиться служебные сообщения и запросы на действия, которые могут повредить систему или удалить пользовательские настройки — внимательно читайте вопросы перед тем, как ввести Y.

Через веб-интерфейс

Вам понадобится fit-файл прошивки для вашей версии контроллера, который можно скачать в нашем репозитории (http://fw-releases.wirenboard.com/?prefix=fit_image/stable/) или по прямой ссылке на stable-релиз:

- 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)

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



Уровень «Администратор»

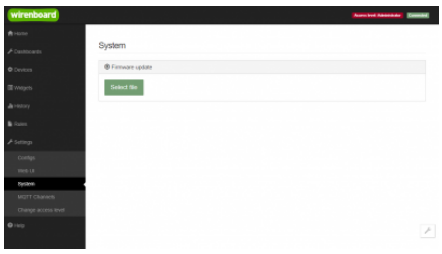
Изменить его можно в разделе **Settings → Change access level**.

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

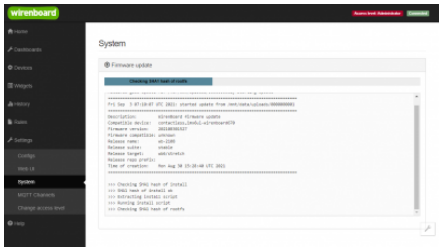
При обновлении сохраняются пользовательские файлы и настройки в `/mnt/data`, но стороннее ПО будет удалено. Рекомендуем сделать резервную копию.

Чтобы обновить прошивку контроллера:

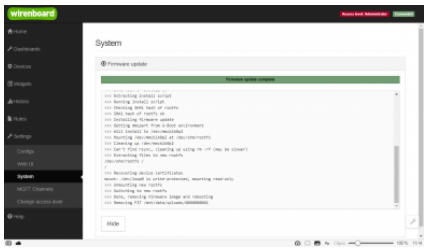
1. Скачайте fit-файл прошивки на компьютер.
2. Зайдите в веб-интерфейс контроллера и перейдите в раздел **Settings → System**. В старых версиях WebUI кнопка *Select file* находится в разделе **Settings**.
3. Нажмите кнопку **Select File** и выберите скачанный ранее fit-файл.
4. Файл с прошивкой загрузится 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.

- 

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 apt 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.

Просмотр файлов контроллера с компьютера

Для внесения изменений в конфигурацию Wiren Board иногда нужно изменять файлы на контроллере. Это можно сделать несколькими способами.

ВНИМАНИЕ: будьте осторожны — повреждение некоторых файлов может нарушить работу контроллера.

Contents

Windows
- Настройка подключения
- Операции с файлами

Mac OS X

Ubuntu

Контроллер

Windows

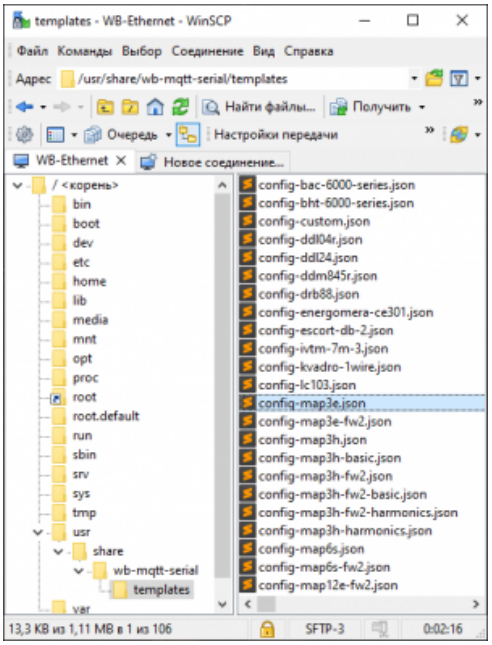
Настройка подключения

Мы рекомендуем использовать WinSCP — это бесплатный графический клиент SFTP (SSH File Transfer Protocol) для Windows.

Скачать WinSCP можно на сайте производителя: winscp.net (https://winscp.net/eng/docs/lang:ru).

Перед использованием WinSCP нужно установить и настроить:

1. Перейдите на сайт программы, загрузите последнюю версию и установите ее. В процессе установки рекомендуем выбрать режим «Проводник».
2. После установки перезагрузите компьютер.
3. Запустите WinSCP и создайте новое соединение:
   - Протокол передачи — **SFTP**. Порт — **22**.
   - Имя хоста — ip-адрес контроллера. см.Как узнать IP-адрес контроллера.
   - Имя пользователя и пароль — по умолчанию: **root** и **wirenboard**.
   - Нажмите **Сохранить** и **Войти**.



Файловая система контроллера в проводнике WinSCP

Если параметры подключения заданы верно, то WinSCP подключится к контроллеру и отобразит дерево каталогов.

Операции с файлами

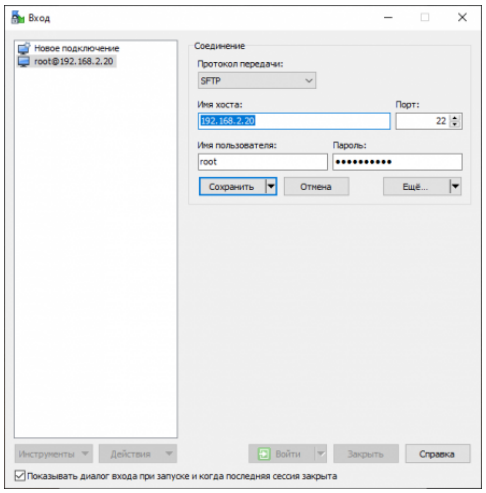
С помощью WinSCP вы можете:

- перемещать файлы между компьютером и контроллером,
- редактировать файлы прямо на контроллере.

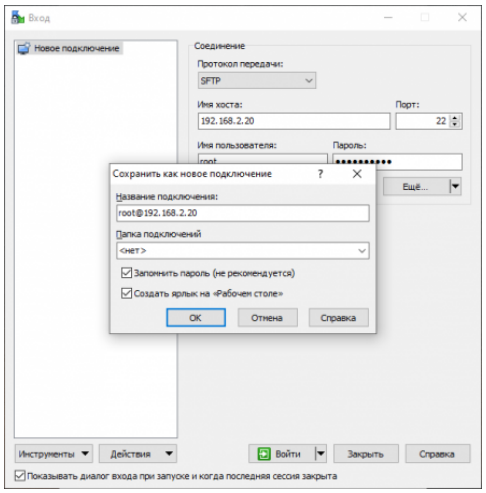
Для обмена файлами их достаточно перетащить в окно или из окна программы в windows-проводник.

Чтобы отредактировать текстовый файл прямо на контроллере, дважды кликните в окне программы на изображении файла. Файл откроется во встроенном редакторе.

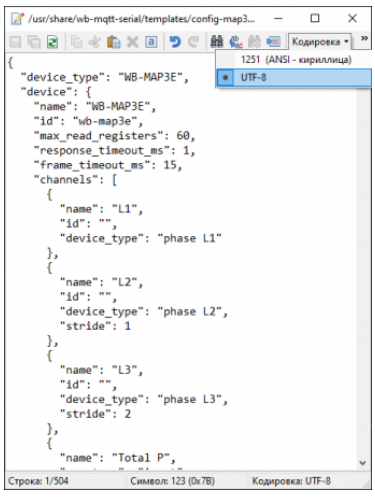
ВНИМАНИЕ: Файлы на контроллере хранятся в кодировке UTF-8. При открытии файла во встроенном редакторе, проверьте кодировку: **меню Кодировка → UTF-8**.

- 

  Настройка соединения с контроллером

- 

  Сохранение настроек соединения

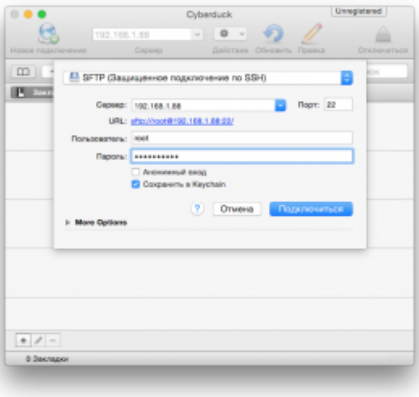
- 

  Встроенный редактор текстовых файлов, установите кодировку UTF-8

Mac OS X

Установите и запустите Cyberduck — http://cyberduck.ch/.

1. Нажмите кнопку **Новое подключение**
2. Заполните поля:
    1. Выпадающее меню: SFTP (Защищенное подключение по SSH)
    2. Сервер — введите IP-адрес (см. Как узнать IP-адрес контроллера)
    3. Порт — оставьте 22
    4. Пользователь — введите root
    5. Пароль — введите ваш пароль (по умолчанию "wirenboard")
3. Нажмите **Подключиться**
4. Добавьте SSH ключ в список ключей
5. В меню выбора папки перейти в корень



Настройка подключения через SFTP для Mac OS X с помощью Cyberduck

Для повторного подключения к серверу, с которым уже было установлено соединение нужно:

1. Нажать на значок книжки
2. Перейти в раздел **Журнал**
3. Выбрать нужный адрес и кликнуть по нему 2 раза
4. В меню выбора папки перейти в корень

Для изменения файла без скачивания на локальной компьютер и повторной загрузки нажать правой кнопкой по файлу, который хотите изменить и выбрать пункт **Открывать в**. Файл будет автоматически загружен на локальный компьютер, а после закрытия автоматически выгружен на Wiren Board;

Теперь вы можете работать с файловой системой контроллера как с обычным диском, в том числе открывать и редактировать файлы.

Ubuntu

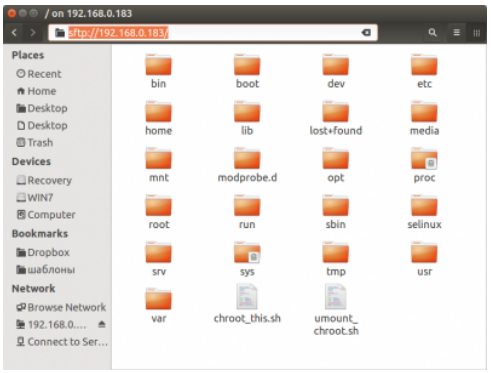
Откройте менеджер файлов **Nautilus**, нажмите `Ctrl+L`, в появившейся адресной строке введите

```
sftp://192.168.42.1    (подставьте ip-адрес-контроллера)
```

(см. Как узнать IP-адрес контроллера)

Появится окно для ввода имени пользователя и пароля — введите *root* — *wirenboard*.

Теперь вы можете работать с файловой системой контроллера как с обычной папкой, в том числе открывать и редактировать файлы.



Доступ к файлам через встроенный файловый менеджер Ubuntu

Контроллер

Файлы контроллера можно просматривать и редактировать через консоль контроллера. Для этого:

1. Зайдите в консоль контроллера.
2. `mcedit /etc/wb-mqtt-serial.conf #открыть файл в псевдографическом редакторе; замените имя файла на нужное`
3. Нажмите F2 для сохранения изменений, F10 — для выхода из редактора.

MQTT

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

Принцип работы MQTT

- Отображение устройств в структуре сообщений
- Клиенты MQTT
- Структура сообщения о состоянии устройства
- Структура сообщения об ошибке опроса устройства
- 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 web-интерфейсе

Работа с сообщениями MQTT с внешнего устройства

- Настройка MQTT моста (bridge)
  - 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 Wiren 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 пользователя устройству.

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

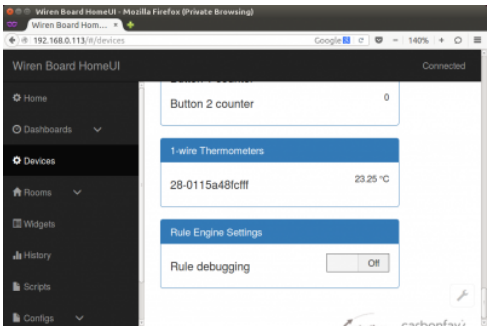
Получение значения от датчика температуры и вывод его в веб-интерфейс

К Wiren Board подключён датчик температуры по шине 1-Wire. Проследим, как данные с него через MQTT попадают в веб-интерфейс:

1. Драйвер wb-mqtt-w1 (https://github.com/wirenboard/wb-mqtt-w1), отвечающий за данную аппаратную функцию, опрашивает подключенные к контролеру датчики 1-Wire.
2. 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.
3. Веб-интерфейс, который подписан на все сообщения из MQTT, получает это сообщение и выводит значение датчика на страницу.



Показания 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:

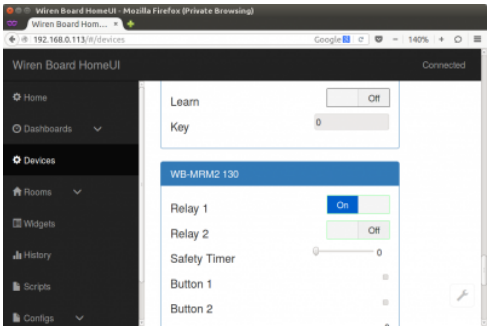
1. После нажатия кнопки веб-интерфейс публикует по MQTT сообщение вида:

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

   Оно значит, что устройство WB-MRM2 с адресом *130* должно перевести *Реле 1* в состояние логической единицы — «включено».
2. Драйвер wb-mqtt-serial, отвечающий за данную аппаратную функцию, получает это сообщение (он *подписан* на все сообщения, относящиеся к подключённым по RS-485 устройствам) и посылает по шине RS-485 релейному модулю команду на включение первого реле.
3. Релейный модуль WB-MRM2 получает команду от контроллера, переключает реле и посылает обратно уведомление «Реле 1 включено».
4. Драйвер wb-mqtt-serial получает это уведомление по RS-485 и публикует по MQTT сообщение:

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

   Оно значит, что первое реле на устройстве WB-MRM2 с адресом *130* находится (уже переведено) в состоянии логической единицы — «включено».



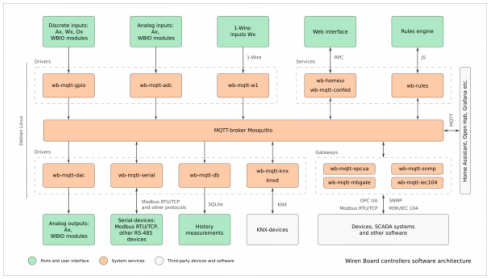
Веб-интерфейс после нажатия кнопки включения Реле 1 на подключённом по RS-485 релейном модуле WB-MRM2

Принцип работы MQTT

Система сообщений MQTT построена по следующему принципу:

- есть иерархическая система «топиков» (как на обычных форумах в интернете).
- в эти топики клиенты (в случае Wiren Board это драйверы устройств и веб-интерфейс) могут писать сообщения и читать оттуда.
- чтобы следить за изменениями нужного топика (например, температуры на датчике), клиент может на него «подписаться» — тогда он получит все сообщения в этом топике.

Читать полное описание системы топиков и подписок (http://mosquitto.org/man/mqtt-7.html).



Через MQTT работают драйверы внутренних функций, внешних устройств, веб-интерфейс, система правил

Отображение устройств в структуре сообщений

Логика организации топиков, соответствующих разным устройствам и их параметрам, в Wiren Board следует определённым правилам — так называемым соглашениям (Wiren Board MQTT Conventions (https://github.com/wirenboard/conventions/blob/main/README.md)).

Полный список MQTT-топиков можно увидеть на странице *Settings* веб-интерфейса в разделе *MQTT Channels* (появилось в последних версиях прошивки).

Клиенты MQTT

- драйверы внутренних аппаратных функций,
- драйверы внешних подключённых устройств,
- веб-интерфейс,
- движок правил,
- (если есть) собственные программы пользователя.

Структура сообщения о состоянии устройства

Вот сообщение от драйвера температурного датчика 1-Wire из примера выше:

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

Часть до пробела — название топика, после — само сообщение.

Название топика состоит из вложенных друг в друга «подтопиков»:

- `/devices` — коренной топик для всех «устройств» — как встроенных функций Wiren Board (цифровые, АЦП, ...), так и подключённых внешних (например, модулей реле).
- `/wb-w1` — подтопик, который наполняется драйвером 1-Wire.
- `/controls` — подтопик, который есть у всех устройств — именно в него записываются все их параметры, которые меняются («включено-выключено», значение датчика, ...).
- `/28-0115a48fcfff` — непосредственно сам «канал» («контрол») — топик, куда записывается значение с датчика. Его название совпадает с адресом 1-Wire датчика на шине.

Содержание сообщения:

- `23.25` — значение температуры

Если вы хотите самостоятельно написать драйвер устройства, и хотите, что оно отображалось на вкладке **Devices** и его можно было использовать в правилах, вам необходимо придерживаться такой же структуры топиков.

Структура сообщения об ошибке опроса устройства

Каждый «канал» («контрол») имеет «подтопик» `/meta/error`, в котором содержится информация о наличии ошибок взаимодействия с устройством. Ошибки получения данных (чтения) обозначаются символом **r**, ошибки записи — **w**.

Пример ошибки получения данных:

```
/devices/wb-w1/controls/28-0115a48fcfff/meta/error r
```

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 соотвествующего 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» («контрол») *Relay 1*. У него самого есть значение — *0* (реле выключено), и есть два подтопика. Один из них — служебный: в `/meta/type` записан тип «контрола». 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» («контрол»).

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 релейном модуле 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 метасиволу. 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 «контрола» устройства, но и топики с метаданными — название драйвера устройства и тип «контрола» - *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 топики, 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

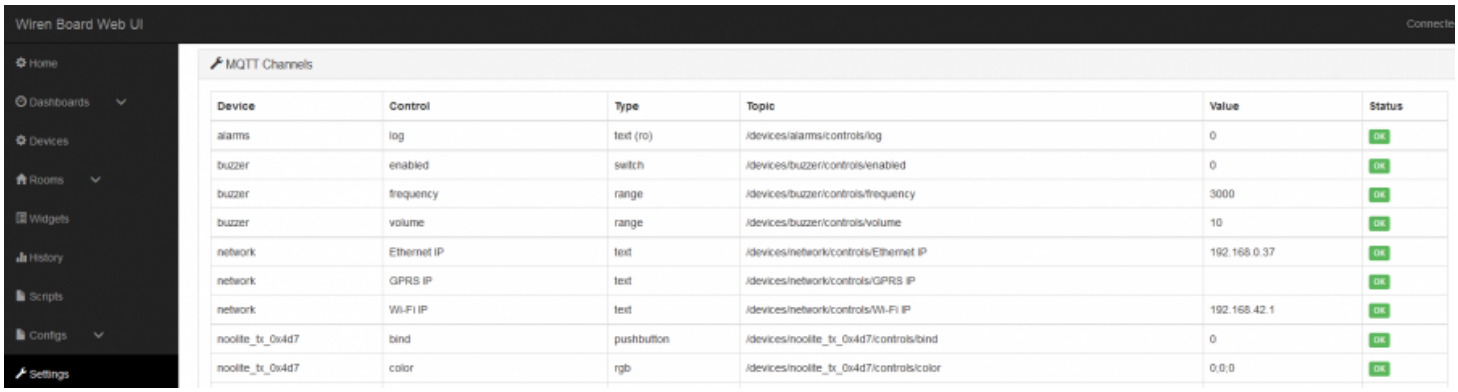
Если вы разрабатываете собственное ПО для Wiren Board, взаимодействовать с его аппаратными ресурсами лучше всего через протокол MQTT — ваша программа передаёт сообщение по MQTT, драйвер управляет устройством и вашей программе не нужно напрямую взаимодействовать с устройством на низком уровне.

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 web-интерфейсе

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



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 прочитать с другого компьютера с Linux, находящегося в той же сети:

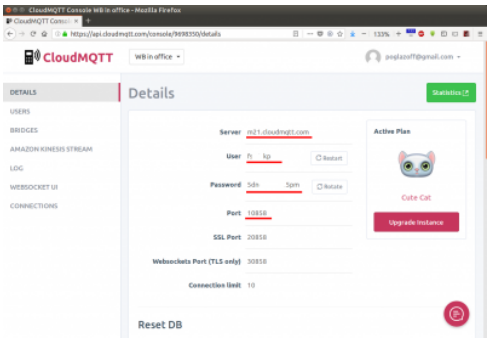
```
mosquitto_sub -h 192.168.0.67 -p 1883 -v -t "/devices/power_status/controls/Vin" -i Test_Client
```

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.



Настройки брокера Cloud MQTT

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

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:

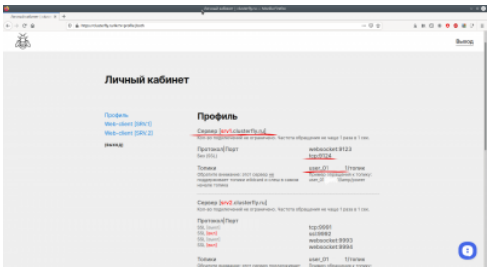
```
connection cloudmqtt
address m21.cloudmqtt.com:10858
remote_username fs_user_kp
remote_password 5dn_pass_pm
clientid pavel_test
try_private false
start_type automatic
topic # both
```

(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



Настройки брокера 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`.
3. 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
```

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 топику: `user_xxxxxx/devices/wb-mr6c_200/controls/K2`.

4. Перезапустите `mosquitto`, выполнив в консоли:

```
service mosquitto restart
```

Потребуется подождать некоторое время пока брокер `mosquitto` сможет организовать соединение. Подписавшись на контроллере к топику `/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:

1. *DestinationController* с адресом `10.0.0.40`, на этот контроллер получать топик.
2. *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 выгляджеть как **/devices**/temp1.
- `/client` — «корень» откуда забираем на удаленном. На контроллере *SourceController* это `/client` и полный путь будет выгляджеть 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 anonomous login:
     allow_anonymous false
     #Password file:
     password_file /etc/mosquitto/mosquitto.pwd
     ```

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

   ```
   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 wbpassword -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 wbpassword -t "/client/temp1/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 wbpassword

topic /temp1/# both 2 /devices /controller
```

Содержимое топика `/devices/temp1/#` контроллера будет отображаться на брокере в `/controller`. Вместо `/controller` можете указать уникальное имя контроллера, например, серийный номер.

Драйвер wb-mqtt-serial

Contents

Описание
- Общая информация
- Особенности

Управление драйвером

Диагностика неполадок

Включение отладки

Полезные ссылки

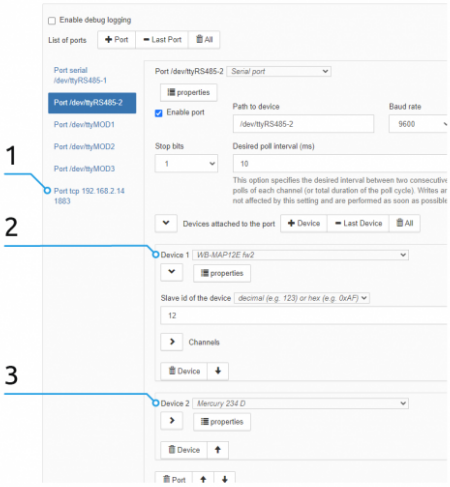
Описание

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 топики MQTT-брокера. Устройства настраиваются через веб-интерфейс.



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

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 Wiren 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.

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.

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'
```

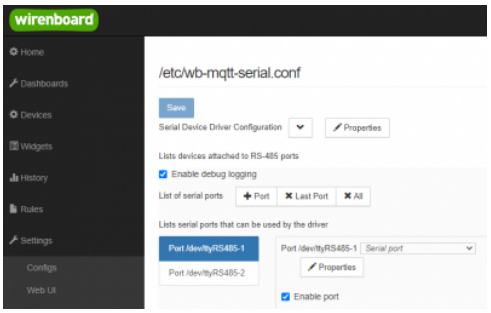
При необходимости, можно добавить путь к файлу, который нужно проверить:

```
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.



Веб-интерфейс. Флажок *Enable debug logging* установлен, отладка включена

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.

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

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

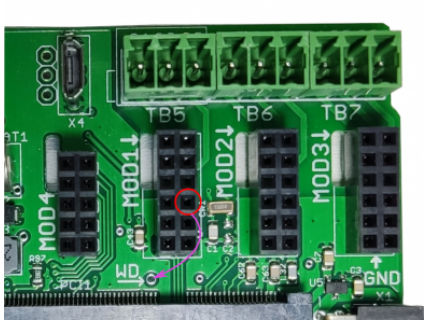
Для отключения требуется разобрать корпус контроллера и:

- 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 разъема ON/OFF с GND.

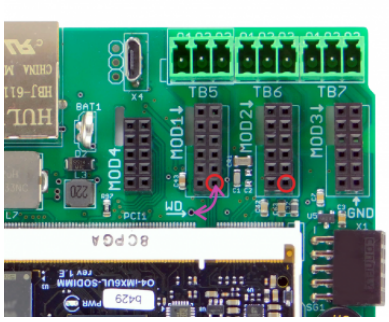
На иллюстрациях красными кружками показаны контакты, к которым нужно подключить контакт WB или контакт из разъёма ON/OFF. Фиолетовой стрелкой показан пример подключения.

- 

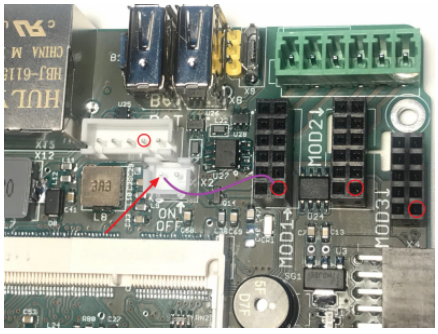
  Wiren Board 7.2. Запаять перемычку Watchdog OFF

- 

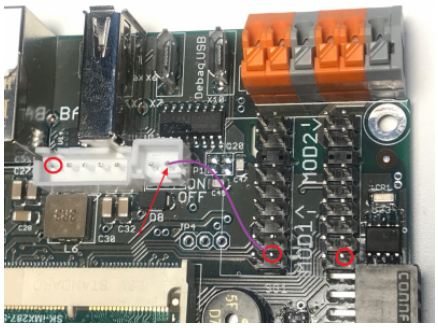
  Wiren Board 6.8. Подключить WD к +5V

- 

  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
```

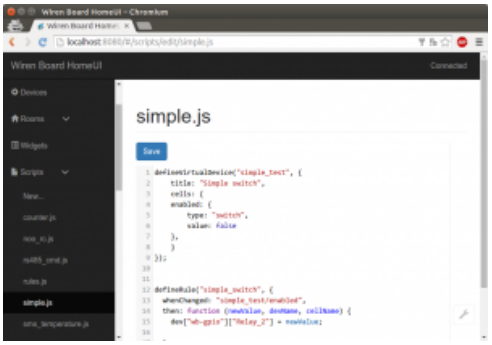
а затем выполните

```
systemctl restart watchdog
```

Движок правил wb-rules

Описание

> Читайте полное описание движка правил на Github (https://github.com/wirenboard/wb-rules).



Редактирование правил в веб-интерфейсе

Wb-rules это возможность писать правила на языке JS. В первую очередь нужно понимать, что такое JS. Знать синтаксис, как происходит работа с функциями, переменными и основными языковыми конструкциями. Подробнее про язык можно узнать в официальном учебнике https://learn.javascript.ru/

Если вы не готовы программировать, возможно вам стоит попробовать создавать правила в среде Node-RED.

Если ваше правило не работает или показывает красным строку и вы не понимаете причину, то возможно вы можете получить дополнительную информацию в Системном журнале, который можно отфильтровать по имени сервиса — wb-rules.

По умолчанию движок правил wb-rules предустановлен на контроллер и запускается автоматически. Но вы можете управлять им самостоятельно из консоли контроллера, читайте подробнее в статье Диагностика ошибок в работе контроллера Wiren Board.

Как создавать и редактировать правила

Правила хранятся на контроллере в папке `/etc/wb-rules/`, поэтому вы можете редактировать и загружать их напрямую с компьютера или использовать веб-интерфейс, вкладка **Rules**.

Если в правиле нет ошибок, оно начинает работать сразу после сохранения файла.

Подробнее о создании правил и возможностях wb-rules, читайте в документации на Github (https://github.com/wirenboard/wb-rules).

Примеры правил смотрите:

- в статье Примеры правил;
- в специальной теме на портале техподдержки (http://forums.contactless.ru/t/dvizhok-pravil-primery-koda/483);
- в исходниках написанного на wb-rules конвертера esphome2wb (https://github.com/wb-adegtyarev/esphome2wb).

Версии wb-rules и совместимость скриптов

Существует две версии движка:

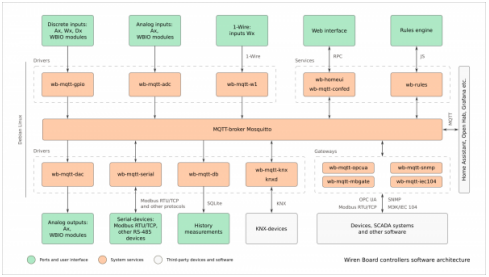
- wb-rules 1.7 — устаревшая версия, поддерживаются контроллеры Wiren Board 5 и 6.
- wb-rules 2.0 (https://github.com/wirenboard/wb-rules) — актуальная версия, поддерживаются контроллеры Wiren Board 6 и 7.

Если у вас контроллер Wiren Board 6 из первых партий и вы переходите с версии 1.7 на 2.0, то прочитайте статью Совместимость скриптов — в ней мы описали возможные проблемы и пути решения.

Программное обеспечение Wiren Board

Архитектура ПО Wiren Board

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`.



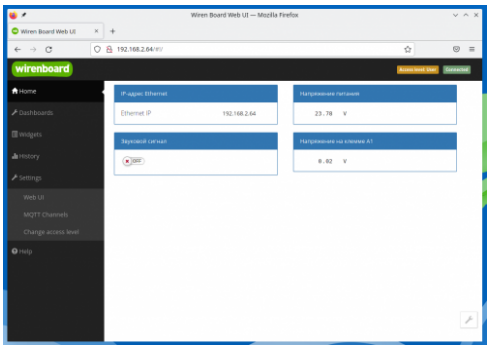
Структура ПО контроллера. В центре очередь сообщений MQTT, которая используется для обмена информацией между разными частями ПО

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

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

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

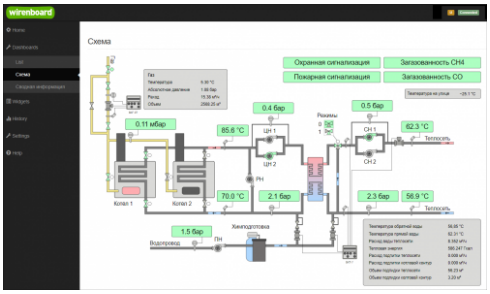
Веб-интерфейс Wiren Board работает непосредственно на контроллере. В нём можно:



Главная страница веб-интерфейса

- 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 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.



Пример графического SVG-дашборда

Для работы с SCADA-системами есть:

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

Node-RED — инструмент визуального программирования.

Полезные ссылки

- Обновление прошивки контроллера
- Как разрабатывать ПО для Wiren Board — статья для программистов.
- Обновление прошивок в Modbus-устройствах Wiren Board

Список сервисов и их назначение

Список сервисов, запущенных на контроллере, их статус и описание можно получить командой:

```
systemctl list-units --type=service
```

Про управление сервисами читайте в статье Диагностика ошибок в работе контроллера.

Имя сервиса	Описание
avahi-daemon.service	Avahi mDNS/DNS-SD Stack
bluetooth.service	Bluetooth service
cgmanager.service	Cgroup management daemon
cron.service	Regular background program processing daemon
dbus.service	D-Bus System Message Bus
dnsmasq.service	dnsmasq - A lightweight DHCP and caching DNS server
getty@tty1.service	Getty on tty1
hostapd.service	LSB: Advanced IEEE 802.11 management daemon
kmod-static-nodes.service	Create list of required static device nodes for the current kernel
knxd.service	KNX Daemon
mosquitto.service	Mosquitto MQTT v3.1/v3.1.1 Broker
netplug.service	LSB: Brings up/down network automatically
networking.service	Raise network interfaces
nginx.service	A high performance web server and a reverse proxy server
ntp.service	LSB: Start NTP daemon
rsyslog.service	System Logging Service
serial-getty@ttymxc0.service	Serial Getty on ttymxc0
ssh.service	OpenBSD Secure Shell server
systemd-fsck-root.service	File System Check on Root Device
systemd-fsck@dev-mmcblk0p6.service	File System Check on /dev/mmcblk0p6
systemd-journal-flush.service	Flush Journal to Persistent Storage
systemd-journald.service	Journal Service
systemd-logind.service	Login Service
systemd-modules-load.service	Load Kernel Modules
systemd-random-seed.service	Load/Save Random Seed
systemd-modules-load.service	Load Kernel Modules
systemd-random-seed.service	Load/Save Random Seed
systemd-remount-fs.service	Remount Root and Kernel File Systems
systemd-sysctl.service	Apply Kernel Variables
systemd-tmpfiles-setup-dev.service	Create Static Device Nodes in /dev
systemd-tmpfiles-setup.service	Create Volatile Files and Directories
systemd-udev-trigger.service	udev Coldplug all Devices
systemd-udevd.service	udev Kernel Device Manager
systemd-update-utmp.service	Update UTMP about System Boot/Shutdown
systemd-user-sessions.service	Permit User Sessions
user@0.service	User Manager for UID 0
watchdog.service	watchdog daemon
wb-configs-early.service	prepare mounts and symlinks to config files
wb-configs.service	watch config files
wb-gsm-rtc.service	LSB: initscript to use GSM modem integrated RTC
wb-homa-ism-radio.service	LSB: MQTT driver for WB HomA for RFM69 ISM radio
wb-hwconf-manager.service	LSB: Hardware configuration with Device Tree overlays
wb-init.service	LSB: board-specific initscript
wb-mqtt-adc.service	MQTT Driver for ADC
wb-mqtt-confed.service	LSB: Configuration Editor Backend
wb-mqtt-db.service	Wiren Board database logger
wb-mqtt-gpio.service	MQTT Driver for GPIO-controlled switches

wb-mqtt-knx.service	LSB: : Wiren Board MQTT KNX bridge
wb-mqtt-logs.service	Wiren Board journald to MQTT gateway
wb-mqtt-mbgate.service	Wiren Board MQTT to Modbus TCP gateway
wb-mqtt-opcua.service	Wiren Board MQTT to OPC UA gateway
wb-mqtt-serial.service	MQTT Driver for serial devices
wb-mqtt-w1.service	Kernel 1-Wire MQTT driver for WB-HomA
wb-prepare.service	initialize filesystems at first boot
wb-repart.service	prepare partitions at first boot
wb-rules.service	MQTT Rule engine for Wiren Board
wb-systime-adjust.service	Compensation of systime in PPM from value, stored in device-tree (with opposite sign)
wb-watch-update.service	LSB: Firmware update monitor

Как зайти на контроллер Wiren Board по SSH

Contents

Введение

Логин и пароль

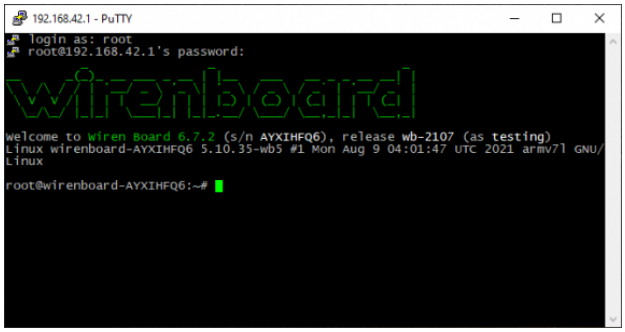
Программы

- Windows
- Linux

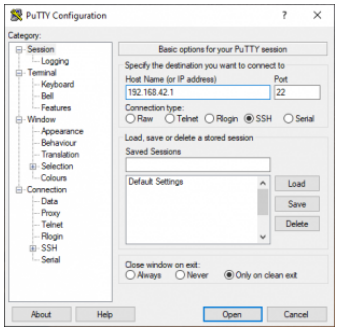
Введение

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

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



Консоль контроллера Wiren Board



Настройка SSH-соединения в программе PuTTY

Логин и пароль

Логин и пароль по умолчанию:

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

Рекомендуем сменить пароль по умолчанию, для этого введите в консоли контроллера команду `passwd` и дважды введите новый пароль:

```
# passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Программы

Windows

Для операционной системы Windows, используйте бесплатную программу PuTTY.

Linux

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

```
ssh root@192.168.42.1
```

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

IP-адрес зависит от способа подключения и настроек контроллера. Подробнее читайте в статье Как узнать IP-адрес контроллера.

Debug UART

Возможно, для специалистов это будет излишним, но я думаю,что можно привести ссылки на документацию по началу работы с Debug UART и описать, зачем вообще это нужно. Быстрый поиск в интернете меня не удовлетворил.

Утилита «modbus_client»

Contents

Описание

Подготовка к работе

- Контроллер Wiren Board
- Настольный компьютер с Linux

Аргументы командной строки

Примеры использования с оборудованием Wiren Board

- Проверка подключения к устройству и считывание адреса
- Запись нового адреса
- Чтение сигнатуры устройства
- Чтение версии прошивки
- Настройка параметров трансформаторов
- Включение реле релейного модуля
- Одновременное включение нескольких реле
- Настройка взаимодействия входов и выходов реле

Описание

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

Подготовка к работе

Контроллер Wiren Board

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

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

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

После завершения работы с `modbus_client` запустите драйвер обратно:

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

Настольный компьютер с Linux

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

Перейдите в папку со скаченным пакетом и установите его командой:

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

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

Аргументы командной строки

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

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

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

Общие аргументы

Параметр	Описание	Обязателен	Значение по умолчанию
--debug	Может указываться в любой позиции и включает отладку, выводя на экран шестнадцатеричные коды отправляемых и принимаемых данных.	нет	
-m	Определяет тип используемого протокола: ▪ **-mrtu** — Modbus RTU, ▪ **-mtcp** — Modbus TCP. Он должен указываться первым в командной строке, или вторым, если первый аргумент — `--debug` или имя файла порта RS-485.	да	
-a	Задает Modbus-адрес устройства, к которому мы обращаемся.	нет	1
-c	Определяет, какое количество элементов мы запрашиваем.	нет	1
-r	Задает начальный адрес для чтения или записи.	не	100
-t	Указывает код функции Modbus. Кратко они перечислены в выводе modbus_client, подробнее значения кодов описаны на странице Протокол Modbus.	да	
-o	Задает таймаут в миллисекундах.	нет	1000
-0	Ноль. Уменьшает на единицу адрес, задаваемый аргументом **-r**. Это может быть полезным при работе с устройствами с нестандартной адресацией, например, с диапазоном адресов 1 — 65536 вместо привычного 0 — 65535.	нет	

Затем указываются специфические параметры протокола (Modbus RTU или Modbus TCP). Несмотря на информацию, выводимую в подсказке, эти параметры также начинаются со знака `-` (минус,дефис).

Для Modbus RTU

Параметр	Описание	Значение по умолчанию
-b	Скорость передачи данных по последовательной линии	9600
-d	Количество передаваемых бит данных, 7 или 8	8
-s	Количество стоповых битов, 1 или 2	1
-p	Контроль четности: ▪ **-pnone** — нет проверки, ▪ **-peven** — передается бит контроля на четность, ▪ **-podd**— передается бит контроля на нечетность.	even

Для Modbus TCP

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

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

Примеры использования с оборудованием Wiren Board

Проверка подключения к устройству и считывание адреса

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

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

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

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

Ответ:

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

Запись нового адреса

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

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

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

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

Ответ:

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

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

Пример **неправильного** использования команды:

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

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

Чтение сигнатуры устройства

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

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

Ответ:

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

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

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

Ответ:

```
WBMR14
```

Чтение версии прошивки

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

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

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

Настройка параметров трансформаторов

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

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

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

Настройки записываются в память конкретного WB-MAP один раз:

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

Включение реле релейного модуля

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

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

Ответ:

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

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

Одновременное включение нескольких реле

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

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

Ответ:

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

Обратите внимание на структуру данных запроса:

- [01] — адрес
- [0F] — код функции Write Multiple Coils
- [00][00] — адрес первого регистра флагов для записи
- [00][0E] — количество элементов для записи (14)
- [02] — количество байт данных (14 бит помещаются в 2 байтах)
- [55][15] — 01010101 00010101 (первое реле — младший бит первого байта, 8 реле — старший бит первого байта, 9 реле — младший бит второго байта)
- [1A][97] — CRC16

А так же на структуру ответа:

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

Подробнее описание структуры данных запросов и ответов можно найти на странице Протокол Modbus.

Настройка взаимодействия входов и выходов реле

Примеры смотрите в статье Примеры настройки взаимодействия входов и выходов.

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 0-3V,-50..+50V	3U	I
WBIO-AI-DV-12/4-20MA	Модуль аналоговых входов 4-20мА	12	Аналоговые входы 4-20мА	3U	I
Модули выходов					
WBIO-DO-R10A-8	Модуль релейных выходов 7А	8	Механические реле, SPST	3U	O
WBIO-DO-R10R-4	Модуль релейных выходов 3А для управления роллетами	4	Механические реле, SPCO	3U	O
WBIO-DO-R1G-16	Модуль релейных выходов 1А 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 3А	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:

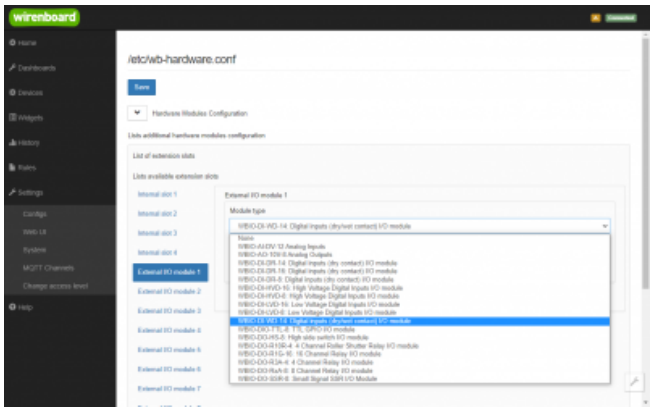
1. В веб-интерфейсе перейдите в раздел **Settings** → **Configs**.
2. Зайдите в раздел **Hardware Modules Configuration**, выберите из **External I/O module** тот, куда установлен модуль расширения.
3. В раскрывающемся списке **Module type** выберите тип установленного модуля.
4. Нажмите кнопку **Save**.

Для удаления модуля выберите тип **None**.

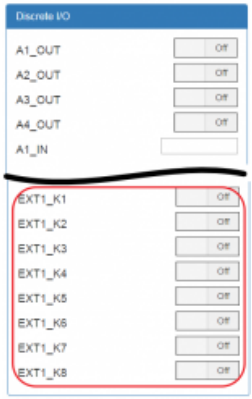
Убедитесь, что внешний модуль виден в веб-интерфейсе:

1. - Перейдите в раздел **Devices**.
   - Найдите устройство Discrete I/O. Каналы внешних модулей будут иметь вид EXT**N_YYYY**, где N — порядковый номер модуля, а YYY — название канала модуля.

     Адрес MQTT-топика внешнего модуля будет иметь вид : /devices/wb-gpio/controls/EXT**N_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)

Центр документации

- English
- русский

Контроллеры

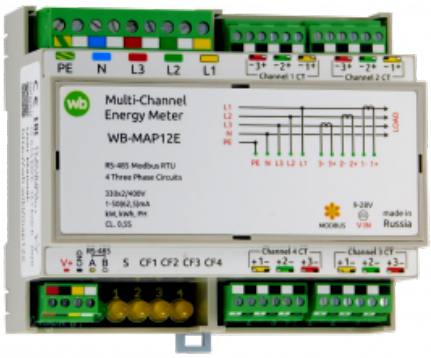
Универсальные контроллеры автоматизации, работающие под управлением свободного программного обеспечения. Применяются в задачах мониторинга серверного и климатического оборудования, диспетчеризации и сбора данных с приборов учёта, в качестве основы для «умного дома» и автоматизации производств.

Wiren Board 6

- 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

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, WB-MR6LV/K — 3- и 6-канальные модули реле общего назначения с переключаемой группой контактов
- WB-MR3LV/I, WB-MR6LV/I — мощные 3- и 6-канальные модули реле 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 — компактный 2-канальный модуль реле
- WB-MRWM2 — мощный 2-канальный модуль реле с **измерением мощности**

WB-MRM-2mini

Датчики

- WB-MS — 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-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 счетных 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.
- **Как подключить устройство RS-485**

WB-MIR v.2

Снятые с производства устройства

- 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)
- 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

Wiren Board 4


Wiren Board NETMON-1

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

Ответы на часто задаваемые вопросы

Contents

С чего начать

Подбор оборудования

Выбор расходных материалов

Технологии

Работа с контроллером

- Обслуживание
- Настройка контроллера
- Веб-интерфейс
- Установка и настройка стороннего ПО
- Информация
- Решение проблем

Модули расширения

Боковые модули ввода-вывода

Работа с Modbus-модулями от Wirenboard

Подключение сторонних устройств по Modbus

С чего начать

В зависимости от ваших навыков и потребностей, начало может быть разным:

- Вам нужна информация о возможностях устройств Wiren Board и схемы их подключения — смотрите документацию. В ней подробно описаны характеристики устройств и даны типовые схемы подключения. Частые вопросы по работе с нашим оборудованием и общие рекомендации приведены ниже на данной странице.
- Вы начинающий и хотите узнать азы — почитайте нашу статью Общие принципы проектирования умного дома (https://wirenboard.com/ru/pages/sh-principles/) и статьи наших клиентов (https://wirenboard.com/ru/pages/articles/). В них описаны примеры реализованных проектов.
- Вы разбираетесь в строительстве и электрике, но не знаете как подступиться к автоматизации — посмотрите реальные проекты нашего партнёра, компании XIOT (https://xiot.ru):
  - Электрический щит для квартиры 200 м^2 (https://wirenboard.com/statics/content/files/5fb53c8992702.pdf)
  - Автоматизация квартир: 56 м^2 (https://wirenboard.com/statics/content/files/6065db27f198b.pdf), 60 м^2 (https://wirenboard.com/statics/content/files/5f85b0427d359.pdf), 85 м^2 (https://wirenboard.com/statics/content/files/60f198d75b223.pdf), 87 м^2 (https://wirenboard.com/statics/content/files/5f7b0327c4c2e.pdf), 98 м^2 (https://wirenboard.com/statics/content/files/60752d2698a09.pdf), 100 м^2 (https://wirenboard.com/statics/content/files/60c3567a0fd43.pdf), 105 м^2 (https://wirenboard.com/statics/content/files/5f92809c4fd27.pdf), 199 м^2 (https://wirenboard.com/statics/content/files/5fb53bbbe8d08.pdf)
- Вам просто нужен кто-то, кто всё сделает «под ключ» — обратитесь к нашим партнёрам (https://wirenboard.com/ru/pages/partners/).

Подбор оборудования

- Обзор модулей Wirenboard
- Оценка стоимости системы автоматизации "Умный дом"
- Рекомендации по выбору компонентов умного дома
- Подбор оборудования Wirenboard под задачи автоматизации
- Рекомендации по выбору устройств сторонних производителей
- Подбор оборудования для однокомнатной квартиры

Выбор расходных материалов

- Какие кабели использовать в проекте

Технологии

- Как правильно организовать шину RS-485
- Можно ли использовать топологию "звезда" у шины RS-485?
- Рекомендации по подключению датчиков 1-wire к контроллеру
- Подключение индуктивных нагрузок (двигателей) к модулям реле

Работа с контроллером

Обслуживание

- Как обновить прошивку контроллера

Настройка контроллера

- Как настроить дату, время и часовой пояс
- Как установить пароль на Wi-Fi
- Как настроить мобильный интернет
- Как ускорить опрос устройств, подключённых к шине RS-485
- Как обновить контроллер Wiren Board и прописать модули (https://youtu.be/eUUxqaUAP2w?t=164), видео от партнёра.

Веб-интерфейс

- Архивация значений History (работа и настройка системы архивации)

Установка и настройка стороннего ПО

- Как установить Node-RED
- Как настроить Telegram-бота с помощью Node-RED
- Как установить Home Assistant на контроллер Wirenboard?

Информация

- Свободное место на eMMC
- IP-адрес контроллера

Решение проблем

- Забыли пароль root
- Как узнать modbus-адреса устройств на шине
- Как сбросить контроллер к заводским настройкам
- Контроллер не распознает флеш-карту при обновлении прошивки или сбросе к заводским настройкам
- Как очистить историю измерений (History) (https://support.wirenboard.com/t/ochistka-history/1924)

Модули расширения

- Управление котлами по интерфейсам OpenTherm и eBus

Боковые модули ввода-вывода

Работа с Modbus-модулями от Wirenboard

- Как обновить прошивку Modbus-устройства
- Какое максимальное количество Modbus-модулей можно подключить к контроллеру?
- Какие задержки при управлении возникают при работе с Modbus-модулем?
- Доступ к Modbus-модулям, расположенным за модулем WB-MGE или WB-MIO-E, из контроллера
- Какие лампы на 12 В нужно купить, чтобы они работали с WB-MRGBW-D (https://support.wirenboard.com/t/wb-mrgbw-d-i-lampochki-12v/5862)
- Как подключить Modbus-устройство Wiren Board к Arduino (https://support.wirenboard.com/t/kak-podklyuchit-tryohfaznyj-voltmetr-wb-map3ev-k-arduino/9717/3)
- Шлюз RS-485 в Wi-Fi
- Работа с Modbus-модулям от Wirenboard без контроллера

Подключение сторонних устройств по Modbus

- Как подключить Modbus-устройство стороннего производителя к контроллеру?

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