

Универсальный настенный датчик WB-MSW v.3 Zigbee

Руководство по эксплуатации

Самая актуальная документация всегда доступна на нашем сайте по ссылке: https://wirenboard.com/wiki/WB-MSW_v.3_Zigbee_Sensor

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

Содержание

[WB-MSW v.3 Zigbee Sensor](#)

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

[Покраска корпуса MSW](#)

[Движок правил wb-rules](#)

[Программа для работы с MQTT](#)

WB-MSW v.3 Zigbee Sensor

Купить в интернет-магазине (<https://wirenboard.com/ru/product/wb-msw-zigbee-v3/>)

Contents

[Назначение](#)

[Модификации](#)

[Покраска корпуса](#)

[Технические характеристики](#)

[Общий принцип работы](#)

[Датчик ТН](#)

[Датчик движения](#)

[Светодиодные индикаторы и зуммер](#)

[Эмуляция ИК-пультов](#)

[Описание](#)

[Управление телевизором](#)

[Управление климатической техникой](#)

[Управление другой техникой по ИК](#)

[Датчик освещенности](#)

[Датчик шума](#)

[Датчик качества воздуха \(VOC\)](#)

[Датчик CO2](#)

[Автокалибровка](#)

[Принудительная калибровка](#)

[Зачем нужно измерять CO2?](#)

[Монтаж](#)

[Подключение](#)

[Монтаж на стену](#)

[Как открыть корпус датчика](#)

[Представление в контроллере](#)

[Программное обеспечение](#)

[Каналы устройства](#)

[Управление](#)

[Настройка](#)

[Как настраивать](#)

[Настройка интервала опроса](#)

[Параметры датчиков](#)

[Таймауты детекторов движения и шума](#)

[Прочие параметры](#)

[ИК-команды](#)

[Постоянная память \(ROM\)](#)

[Оперативная память \(RAM\)](#)

[Обновление прошивки](#)

[Из веб-интерфейса zigbee2mqtt](#)



Универсальный датчик WB-MSW v.3, вид спереди

Через MQTT

Примеры правил

Известные неисправности

Ревизии устройства

Изображения и чертежи устройства

Назначение

Комбинированный цифровой датчик температуры, влажности, освещенности, движения, уровня шума, концентрации CO2 и летучих органических соединений. Оснащён ИК приемопередатчиком.

Предназначен для контроля климата в жилых и офисных помещениях, для бытового использования. Датчик выполнен в пластиковом корпусе и предназначен для крепления на стену.

Устройство работает по протоколу Zigbee, поэтому его можно использовать:

- с контроллером Wiren Board, для этого нужно установить zigbee2mqtt или SprutHub;
- шлюзом SLS Smart Home с версии 2022.01.13d1, подробнее читайте на сайте устройства (https://slsys.io/action/supported_devices.html?device=160);
- и любым другим устройством, на который можно установить zigbee2mqtt.

Модификации

Пока датчик поставляется в полной комплектации.

Покраска корпуса

Разноцветных корпусов нет, но белые корпуса легко покрасить — получается неплохо, см. Покраска корпуса MSW

Технические характеристики

Измеряемая величина	Диапазон	Погрешность	Готовность к работе после подачи питания
Концентрация CO ₂	400 – 5000 ppm (миллионных долей)	50 ppm	3 мин, автокалибровка каждые 7 дн.
Температура	–40 °C – +80 °C	±0.3 °C (в диапазоне 0 – 70 °C) ±0.5 °C (в полном диапазоне)	1 сек постоянная времени (выравнивание с окружающим воздухом) ~4 мин
Относительная влажность	0 – 99.9 % (рабочий диапазон: 0 – 98 %)	±3 %	1 сек
Уровень шума (звуковое давление)	38 – 105 дБ (40 – 82 дБ в версии v.4.8)	±1 дБ (±3 дБ в v.4.8)	5 сек
Освещённость	0,02 – 100 000 лк	±10 %	1 сек
Качество воздуха (концентрация летучих органических соединений — VOC)	0 ppm – 60000 ppb (миллиардных долей) по этанолу	±15 % (тип) ±40 % (макс)	5 мин (самокалибровка спустя 12 ч)
Датчик движения	До 8 м, 120 градусов		8 сек
Передача ИК-команд	До 10 м (зависит от окружающих условий)		1 сек

Параметр	Значение
Питание	
Напряжение питания	9 В – 28 В постоянного тока
Потребляемая мощность	<ul style="list-style-type: none"> ▪ 0.5 Вт; ▪ (пиковая при измерении CO₂) до 1.6 Вт; ▪ (пиковая при измерении CO₂ и передаче ИК сигнала) до 4 Вт.
Количество запоминаемых ИК-команд	
Количество команд	32
Длительность команд	Максимальная длина команды — 508 регистров, плюс два регистра — признак окончания команды. Каждый регистр кодирует длительность высокого или низкого уровня сигнала (последовательно) в микросекундах.
Клеммники и сечение проводов	
Рекомендуемое сечение провода с НШВИ, мм ²	0.35 - 1 мм ² — одинарные, 0.35 - 0.5 мм ² — сдвоенные провода,
Длина стандартной втулки НШВИ, мм	8
Момент затяжки винтов, Н•м	0.2
Индикация	
Питание и обмен данными	Зеленый светодиод в нижней части корпуса
Пользовательские индикаторы	Зеленый и красный светодиод
Звуковая индикация	«Пищалка» — зуммер, beeper
Управление	
Zigbee	полная поддержка в SprutHub и zigbee2mqtt с версии 1.24.0 (страница устройства на сайте проекта (https://www.zigbee2mqtt.io/devices/WB-MSW-ZIGBEE_v.3.html))
Готовность к работе после подачи питания	~2 с
Условия эксплуатации	
Температура воздуха	От –40 до +80 °С
Относительная влажность	До 92 %, без конденсации влаги
При установленном датчике CO ₂	От 0 до +50 °С
Габариты	
Габариты	83 x 83 x 20 мм
Масса (с коробкой)	90 г

Общий принцип работы

Датчик ТН

Датчик температуры и влажности (ТН) распаян на плате модуля.

Модуль автоматически корректирует показания температуры и влажности, учитывая базовый нагрев платы от микросхемы питания (0.3 °С), а также нагрев от установленных датчиков CO₂ (0.31 °С) и VOC (0.14 °С).

Датчик движения

PIR-датчик движения в MSW обнаруживает перемещение объектов на расстоянии до 8 м с углом обзора около 100-120 градусов.

Светодиодные индикаторы и зуммер

Также во всех комплектациях доступен зелёный индикатор обмена данными, который виден в технологическом отверстии на нижней части корпуса.

Пользователь может управлять:

- Двумя яркими светодиодами — зеленым и красным, которые подсвечивают линзу на корпусе. Индикаторы могут только мигать.;
- Звуковым индикатором (зуммер, buzzer).

Индикаторы позволяют обеспечить обратную связь при монтаже и эксплуатации. Например, датчик может мигать красным при превышении оптимальной концентрации углекислого газа ([примеры правил](#)). При монтаже большого количества датчиков индикаторы (или бипер) помогут определить, к какому конкретно датчику вы сейчас обращаетесь.

Про управление светодиодными индикаторами и зуммером, читайте в разделе [Управление](#).



Работа светодиодного индикатора в WB-MSW v.3

Эмуляция ИК-пультов

Описание

В WB-MSW v.3 под линзой расположены ИК-приёмник для обучения и 8 ИК-светодиодов.

Обучение проводится один раз — команды сохраняются в памяти устройства и могут быть воспроизведены командой с контроллера. Количество запоминаемых команд достаточно большое (до сотни, в зависимости от модели пульта), чтобы управлять сразу многими устройствами в помещении. Мощности передатчика хватает, чтобы управляемые устройства принимали не только прямой, но и отраженный сигнал.

Обычно используется для управления кондиционерами и тепловыми завесами. Учтите, что ИК-команды отправляются всем устройствам в зоне видимости, поэтому передать разные команды на одинаковые устройства не получится.

Вы можете записать сигнал в одну из двух видов памяти:

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

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

О том, как записывать ИК-команды и воспроизводить их, читайте в разделе [Настройка](#).

Управление телевизором

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

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

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

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

Управление климатической техникой

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

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

В итоге получится в памяти модуля набор готовых режимов:

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

Управление другой техникой по ИК

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

Датчик освещенности

Датчик освещенности имеет фильтр, который повторяет кривую спектральной чувствительности человеческого глаза. Это позволяет измерять освещенность в люксах, что позволяет обеспечить контроль освещенности в соответствии с нормами СанПиН.

Максимальное время реакции на резкое изменение освещённости — 1.5 с.

Датчик шума

В модуле используется микрофон с усилителем и фильтрами для коррекции по шкале А с учётом особенностей восприятия человеческим ухом звуков разных частот. Шум измеряется в акустических децибелах (дБА), что позволяет контролировать шумовую обстановку в соответствии со стандартами и санитарными нормами.

Датчик качества воздуха (VOC)

Измерение VOC сделано на отдельном модуле, устанавливаемых по стрелочке в разъемы.

Летучие органические вещества (ЛОВ, VOC) - это легкоиспаряющиеся вещества, выделяющиеся в атмосферу в виде газов. Датчик определяет суммарную концентрацию летучих органических веществ, в том числе испарения лаков/красок и элементов внутренней отделки помещений (фенол, формальдегид, толуол, стирол), спирты, бензол, гниющие овощи, выделяемые человеком газы, бытовой газ. Высокие концентрации опасных ЛОВ представляют угрозу жизни и здоровью человека.

Датчик VOC не работает как детектор утечки бытовых горючих газов и совсем не реагирует на дым!

Концентрация измеряется в единицах на миллиард ppb (также называемую ОЛОС — см. ГОСТ Р ИСО 16000-9-2009). Данный параметр характеризует общую концентрацию ЛОВ в усредненном помещении. На основании исследований производителем датчика установлены следующие пороги концентрации:

Концентрация (ppb)	Уровень	Соответствие гигиеническим нормам	Рекомендации	Предельное время воздействия
2200 - 5500	Опасно для здоровья	Ситуация неприемлема	Подвергаться воздействию только в критических случаях / Необходимо интенсивное вентилирование	часы
660-2200	Неудовлетворительно	Серьезные претензии	Необходимо интенсивное вентилирование или проветривание, требуется поиск источников загрязнения	< 1 месяца
220 - 660	Приемлемо	Некоторые претензии	Рекомендуется интенсивное вентилирование или проветривание, требуется поиск источников загрязнения	< 12 месяцев
65 - 220	Хорошо	Без особых претензий	Рекомендуется вентилирование или проветривание	нет предела
0-65	Отлично	Без претензий	Требуемое значение	нет предела

Важно! Датчик готов к работе через 6 минут после включения. До этого в регистре качества воздуха находится значение, сигнализирующее об ошибке (0xFFFF). Примерно каждые 12 часов производится самокалибровка датчика.

Датчик CO2

Для измерения концентрации CO₂ в воздухе используется недисперсионный инфракрасный (NDIR) датчик. Принцип действия основан на поглощении углекислым газом инфракрасного света. Оптический способ измерения CO₂ намного точнее, чем с помощью более дешевых электрохимических датчиков.



Концентрация CO₂ измеряется в ppm — частях на миллион.

Автокалибровка

Измеренное минимальное значение в течение 7 дней принимается за 400 ppm — это значение концентрации CO₂ на улице. Концентрация CO₂ упадет до уличной, если в помещении нет людей хотя бы несколько часов в день, или если в помещении работает вытяжная вентиляция, или в помещении иногда открывают окна.

Принудительная калибровка

В большинстве случаев отключение автокалибровки или принудительная калибровка не требуются — датчик показывает правильные значения без дополнительных манипуляций, но иногда без неё не обойтись:

1. Нужно срочно откалибровать датчик и некогда ждать, пока сработает автоматическая калибровка.
2. Датчик находится в помещении, которое плохо проветривается и уровень CO₂ никогда не достигает 400 ppm. В этом случае не забудьте отключить автоматическую калибровку.

Суть принудительной калибровки заключается в том, что мы помещаем датчик в среду, где уровень CO₂ равен атмосферному (400 ppm) и устанавливаем это значение как начало отсчёта. Чтобы уменьшить ошибки при измерении CO₂, калибруйте датчик при комнатной температуре.

Поместите работающий датчик в хорошо проветренное помещение и подождите 20 минут. Затем выполните следующие шаги:

1. Если нужно отключить автоматическую калибровку — установите параметр **co2_autocalibration** в значение **OFF**.
2. Включите параметр **co2_manual_calibration**, а через секунду выключите.
3. В течение пяти минут измеренное значение CO₂ станет около 400 ppm. Калибровка завершена.

Способы изменения параметров смотрите в разделе [Настройка](#).

Зачем нужно измерять CO2?

Углекислый газ в высоких концентрациях токсичен. Незначительные повышения концентрации, вплоть до 0,2–0,4 % (2000–4000 ppm), в помещениях приводят к развитию у людей сонливости и слабости. Для помещений нормальным является уровень CO₂ около 600 ppm. Повышенные концентрации углекислого газа снижают когнитивные способности людей. Уже при 1200 ppm расширяются кровеносные сосуды в мозге, снижается активность нейронов и уменьшается объём коммуникации между областями мозга.

Влияние на взрослых здоровых людей	Концентрация углекислого газа, ppm
Нормальный уровень на открытом воздухе	400—450
Приемлемые уровни	<600
Жалобы на несвежий воздух	600—1000
Общая вялость	1000—2500
Возможны нежелательные эффекты на здоровье	2500—5000
Максимально допустимая концентрация в течение 8 часового рабочего дня	5000

Классификация воздуха в помещениях по ГОСТ 30494-2011

Класс	Качество воздуха в помещении		Допустимое содержание CO ₂ *, см ³ /м ³
	Оптимальное	Допустимое	
1	Высокое	-	400 и менее
2	Среднее	-	400-600
3	-	Допустимое	600-1000
4	-	Низкое	1000 и более

* Допустимое содержание CO₂ в помещениях принимают сверх содержания CO₂ в наружном воздухе, см³/м³

Монтаж

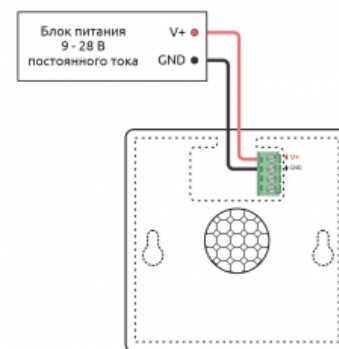
Подключение

Подключите питание к клеммам V+ и GND датчика, используйте комплектный блок питания, или любой с напряжением 9 В – 28 В постоянного тока.

Перед подключением датчика к контроллеру Wiren Board, настройте контроллер для работы по протоколу Zigbee.

Теперь нужно спарить датчик и Zigbee-координатор (контроллер):

1. Включите в настройках координатора спаривания устройств (Pairing).
2. Проведите магнит по задней нижней части корпуса или снимите крышку и нажмите кнопку на Zigbee-модуле. Зелёный индикатор в отверстии должен замигать.
3. Подождите, пока датчик не появится в списке устройств, известных координатору.
4. Отключите режим спаривания устройств (Pairing).



Подключение питания к датчику WB-MSW v.3 Zigbee

Монтаж на стену

Модуль имеет отверстия для крепления к поверхности. Мы подготовили установочный шаблон для корпуса датчика:

1. Скачайте файл [Msw3 mounting template.pdf](#) на компьютер.
2. Откройте в Acrobat Reader и при печати выберите опцию «Реальный масштаб».

Для крепления WB-MSW v.3 выбирайте винты/шурупы с головкой диаметром около 7 мм, если требуется, чтобы датчик был съемным, и 9-10 мм — для постоянной фиксации.

Устройство должно эксплуатироваться при рекомендованных условиях окружающей среды.

Рекомендуем располагать датчики на теплых (внутренних) стенах, на высоте 1-1.6 м от уровня пола, с учетом возможных сквозняков и освещенности солнцем. При креплении на потолке в жилом помещении температура будет завышена, а влажность занижена. Концентрация CO₂ от высоты не зависит. При креплении на внешних стенах зимой будут заниженные показания температуры на несколько градусов (из-за холодного пограничного слоя воздуха и охлаждения корпуса датчика от стены).

Сразу же после установки датчик CO₂ может показывать неверные значения: это может быть связано с неосторожным обращением во время транспортировки и монтажа. Вы можете подождать 7 дней без отключения питания, пока функция автокалибровки не приведёт показания датчика в норму или выполнить принудительную калибровку.

Как открыть корпус датчика



1. Найдите язычок защелки на нижней стороне корпуса датчика



2. Надавите отверткой на язычок перпендикулярно боковой стороне корпуса датчика до упора



3. Поднимите верхнюю крышку датчика

Представление в контроллере

Программное обеспечение

Датчик можно использовать на контроллере Wiren Board и других устройствах со SprutHub и zigbee2mqtt с версии 1.24.0.

Подробнее о поддержке Zigbee-устройств на контроллерах Wiren Board, читайте в статье [Zigbee](#).

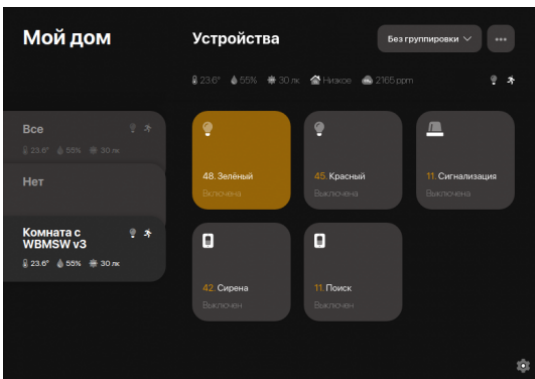
- Представление датчика в разном программном обеспечении

Ox842e14ffe8b184e	
Temperature	20.94 °C
Illuminance	13071
Humidity	41.5 %
Occupancy	Occupied
Occupancy Level	236
CO2	484 ppm
Voc	312 ppb
Noise	62.55 dBA
Noise Detected	True
State_L1 (I1)	<input type="checkbox"/>
State_I2 (I2)	<input checked="" type="checkbox"/>
State Default (default)	<input type="checkbox"/>
just now	184 LQI ?

WB-MSW v.3 Zigbee в веб-интерфейсе zigbee2mqtt 1.24.0

Ox842e14ffe8b184e	
last_seen	1640007618964
linkquality	182
state_l1	OFF
state_l2	ON
state_default	OFF
temperature	25.84 °C
illuminance	14210
illuminance_lux	26
humidity	34.42 %, RH
occupancy_level	1876
occupancy	false
noise	40.18
noise_detected	false
co2	785
voc	498

WB-MSW v.3 Zigbee в веб-интерфейсе контроллера, требуется zigbee2mqtt 1.24.0



WB-MSW v.3 Zigbee в веб-интерфейсе SprutHub

Каналы устройства

Каналы устройства:

- *temperature* — температура, °C
- *humidity* — относительная влажность воздуха, %, RH
- *co2* — концентрация CO2, ppm
- *voc* — качество воздуха, ppb
- *illuminance* — освещённость в условных единицах
- *illuminance_lux* — освещённость, lux
- *occupancy* — флаг наличия движения, рассчитывается по внутреннему алгоритму. Можно изменять время его сброса в параметре occupancy_timeout
- *occupancy_level* — уровень движения

- *noise_detected* — флаг наличия шума, рассчитывается по внутреннему алгоритму. Можно изменять время его сброса в параметре *noise_timeout*
- *noise* — уровень шума, dBA
- *update_available* — флаг, указывающий на то, что доступна новая прошивка по OTA
- *linkquality* — качество связи
- *state_default* — статус зуммера
- *state_l1* — статус красного светодиода
- *state_l2* — статус зелёного светодиода.

Управление

Зуммером и светодиодами можно управлять, для этого нужно отправить сформированный по образцу JSON-запрос в mqtt-топик `zigbee2mqtt/FRIENDLY_NAME/set`:

- красный светодиод — `{"state_l1": "ON"}` или `{"state_l1": "OFF"}`
- зелёный светодиод — `{"state_l2": "ON"}` или `{"state_l2": "OFF"}`
- зуммер — `{"state_l3": "ON"}` или `{"state_l3": "OFF"}`

Отправлять JSON-запросы можно с помощью [wb-rules](#), например, включим зелёный светодиод в датчике с именем `0x842e14ffe8b184e`:

```
publish('zigbee2mqtt/0x842e14ffe8b184e/set', JSON.stringify({"state_l2": "ON"}), 2, false);
```

Про управление ИК-приёмопередатчиком и настройку параметров читайте в разделе [Настройка](#).

Настройка

Как настраивать

Датчик можно настраивать в SprutHub и в zigbee2mqtt с версии 1.24.0.

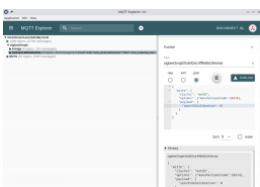
Вы можете настроить устройство веб-интерфейсе zigbee2mqtt, а также через запись JSON-команд в mqtt-топики устройства с помощью [MQTT Explorer](#) или [wb-rules](#). Про настройку датчика в SprutHub, читайте в его документации.

При настройке вам понадобится понимание некоторых терминов:

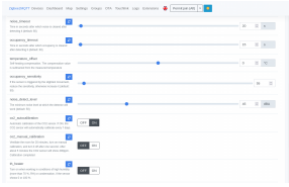
- Канал — измеренные значения.
- Параметр — элемент, который позволяет настраивать устройство.
- Атрибут — внутреннее наименование канала или параметра.
- Кластер — группа атрибутов.
- Настройка устройства в разном программном обеспечении



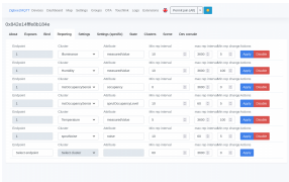
Настройка с помощью `wb-rules` в веб-интерфейсе контроллера.



Настройка с помощью программы MQTT Explorer



Настройка параметров в веб-интерфейсе zigbee2mqtt



Настройка интервалов опроса в веб-интерфейсе zigbee2mqtt

Настройка интервала опроса

Настройка интервалов опроса позволяет регулировать нагрузку на Zigbee-сеть и экономить ресурсы контроллера на обработке полученных от устройства данных. Чем реже устройство присылает данные, тем меньше требуется ресурсов для их обработки.

Интервалы настраиваются для каждого канала отдельно. Значения по умолчанию подойдут для большинства пользователей, но вы можете изменить их:

- **minimum_report_interval** — Минимальный интервал опроса, который будет выдержан перед отправкой изменённых значений датчиком. Может игнорироваться, если не было соблюдено условие, задаваемое в настройке «Отчётное значение».
- **maximum_report_interval** — Максимальный интервал опроса, по истечении которого датчик пришлёт данные независимо от того, изменились ли измеренные значения.
- **reportable_change** — Отчётное значение. Если значение параметра изменилось на заданную величину или больше, а также прошло время, указанное в параметре «Минимальный интервал опроса», устройство пришлёт отчёт. При настройке нужно учитывать, что для температуры и влажности 100 — это 1 °C и 1 % соответственно. Для других параметров значения равны 1 к 1.

Чтобы настроить интервал опроса, отправьте в MQTT-топик `zigbee2mqtt/bridge/request/device/configure_reporting` специально сформированный JSON-запрос.

В шаблоне ниже заполните *id*, *cluster* и *attribute*, значениями которые можно взять из таблицы и укажите параметры опроса в полях *minimum_report_interval*, *maximum_report_interval*, *reportable_change*:

```
{
  "id": "0x0000000000000000",
  "options": {},
  "cluster": "yourCluster",
  "attribute": "yourAttribute",
  "minimum_report_interval": 5,
  "maximum_report_interval": 10,
  "reportable_change": 10
}
```

Для каналов **occupancy_level** и **noise**, секция **options** должна выглядеть так: `"options": {"manufacturerCode": 26214},`. В остальных случаях оставьте её пустой `"options": {}`.

Например, настроим с помощью `wb-rules` период опроса уровня шума (`noise`) в датчике с адресом `0x842e14fffe8b184e`:

```
message = {
  "id": "0x842e14fffe8b184e",
  "options": {"manufacturerCode":26214},
  "cluster": "sprutNoise",
  "attribute": "noise",
  "minimum_report_interval": 10,
  "maximum_report_interval": 60,
  "reportable_change": 5
}

publish('zigbee2mqtt/bridge/request/device/configure_reporting', JSON.stringify(message), 2, false);
```

Интервалы опроса						
Канал	Описание	Кластер	Атрибут	Значения по умолчанию		
				Минимальный интервал, с	Максимальный интервал, с	Отчётное значение
temperature	Температура	msTemperatureMeasurement	measuredValue	10	3600	100
illuminance	Освещённость	msIlluminanceMeasurement	measuredValue	10	3600	5
humidity	Влажность	msRelativeHumidity	measuredValue	10	3600	100
occupancy	Флаг наличия движения	msOccupancySensing	occupancy	0	3600	0
occupancy_level	Уровень движения	msOccupancySensing	sprutOccupancyLevel	10	60	5
noise	Уровень шума	sprutNoise	noise	10	60	5

Параметры датчиков

При настройке через MQTT, сформируйте JSON-запрос по образцу и отправьте его в `mqtt`-топик:

- чтение `zigbee2mqtt/FRIENDLY_NAME/get` ответ будет опубликован в топике устройства и отобразится в веб-интерфейсе контроллера в карточке устройства на вкладке `Devices`.
- запись `zigbee2mqtt/FRIENDLY_NAME/set`

`FRIENDLY_NAME` — идентификатор устройства.

Таймауты детекторов движения и шума

Таймауты для сброса флагов о зафиксированных движении и шуме:

- noise_timeout** — Таймаут обнаружения шума. Время в секундах, которое должно пройти в тишине перед сбросом флага `noise_detected`. Можно указывать от 0 до 2000, по умолчанию 60 секунд. JSON запросы:
 - чтение — `{"noise_timeout": ""}`
 - запись — `{"noise_timeout": NEW_VALUE}`
- occupancy_timeout** — Таймаут обнаружения движения. Время в секундах, которое должно пройти без движения перед сбросом флага `occupancy`. Можно указывать от 0 до 2000, по умолчанию 60 секунд. JSON запросы:
 - чтение — `{"occupancy_timeout": ""}`
 - запись — `{"occupancy_timeout": NEW_VALUE}`

Например, установим с помощью `wb-rules` в датчике с именем `0x842e14fffe8b184e` параметр `noise_timeout` в значение 20:

```
publish('zigbee2mqtt/0x842e14fffe8b184e/set', JSON.stringify({"noise_timeout": 20}), 2, false);
```

Прочие параметры

Для настройки параметров, указанных в таблице «Параметры», нужно отправить устройству подготовленный JSON-запрос.

Для чтения параметра заполните в шаблоне ниже `cluster` и `attributes`, значениями которые можно взять из таблицы:

```
{
  "read": {
    "cluster": "yourCluster",
    "options": {"manufacturerCode":26214},
    "attributes": [
      "yourAttribute"
    ]
  }
}
```

Для записи параметра заполните в шаблоне ниже *cluster* и *payload*, значениями которые можно взять из таблицы и укажите новое значение вместо *NEW_VALUE*:

```
{
  "write": {
    "cluster": "yourCluster",
    "options": {"manufacturerCode":26214},
    "payload": {
      "yourAttribute": NEW_VALUE
    }
  }
}
```

Например, включим с помощью *wb-rules* принудительную калибровку датчика CO2 в устройстве с адресом *0x842e14fffe8b184e*:

```
message = {
  "write": {
    "cluster": "msCO2",
    "options": {"manufacturerCode":26214},
    "payload": {
      "sprutCO2Calibration": 1
    }
  }
}

publish('zigbee2mqtt/0x842e14fffe8b184e/set', JSON.stringify(message), 2, false);
```

Параметры					
Параметр	Описание	Кластер	Атрибут	Значения	
				Возможное	По умолчанию
noise_detect_level	Минимальный уровень шума в дБА, при котором будет работать детектор.	sprutNoise	noiseDetectLevel	0 – 150	50
occupancy_sensitivity	Чувствительность датчика движения. Если датчик срабатывает при малейшем движении, уменьшите чувствительность, в противном случае увеличьте ее.	msOccupancySensing	sprutOccupancySensitivity	0 – 2000	50
th_heater	Подогрев датчика ТН. Включается при работе в условиях высокой влажности (более 70 %, RH) или конденсации, если датчик показывает 0 или 100 %.	msRelativeHumidity	sprutHeater	0 или 1	0
temperature_offset	Пользовательская компенсация измеренного значения температуры в °С. Указанное значение вычитается из измеренного. Доступна только запись.	msTemperatureMeasurement	sprutTemperatureOffset	-10...+10	0
co2_autocalibration	Автоматическая калибровка датчика CO2 на атмосферный уровень.	msCO2	sprutCO2AutoCalibration	0 или 1	1
co2_manual_calibration	Принудительная калибровка датчика CO2 на атмосферный уровень. Инструкция.	msCO2	sprutCO2Calibration	0 или 1	0

ИК-команды

Для управления ИК приёмопередатчиком, сформируйте JSON-запрос по образцу и отправьте его в mqtt-топик zigbee2mqtt/FRIENDLY_NAME/set.

О принципах управления ИК-устройствами, читайте в разделе [Эмуляция ИК-пультов](#).

Постоянная память (ROM)

Сигналы, записанные в постоянную память (ROM) сохраняются после отключения питания.

Доступные команды, *rom* — номера ячеек памяти от 0 до 31:

- Начать обучение:

```
{
  "learn_start": {
    "rom":0
  }
}
```

- Завершить обучение:

```
{
  "learn_stop": {
    "rom":0
  }
}
```


- Воспроизвести сигнал:

```
{
  "play_store": {
    "rom":0
  }
}
```

- Удалить сигналы из всех ячеек ROM:

```
{
  "clear_store": {}
}
```

Например, воспроизведём ИК-сигнал из ячейки с адресом 0, которая находится в датчике с именем *0x842e14fffe8b184e*:

```
publish('zigbee2mqtt/0x842e14fffe8b184e/set', JSON.stringify({"play_store": {"rom":0}}), 2, false);
```

Оперативная память (RAM)

Сигналы, записанные в оперативную память (RAM) стираются при отключении питания или при операциях с банками постоянной памяти (ROM).

Доступные команды:

- Начать обучение:

```
{
  "learn_ram_start": {}
}
```

- Завершить обучение:

```
{
  "learn_ram_stop": {}
}
```

- Воспроизвести сигнал:

```
{
  "play_ram": {}
}
```

Например, воспроизведём ИК-сигнал из оперативной памяти в датчике с именем *0x842e14fffe8b184e*:

```
publish('zigbee2mqtt/0x842e14fffe8b184e/set', JSON.stringify({"play_ram": {}}), 2, false);
```

Обновление прошивки

Файлы прошивок для обновления через zigbee2mqtt пока недоступны, следите за новостями.

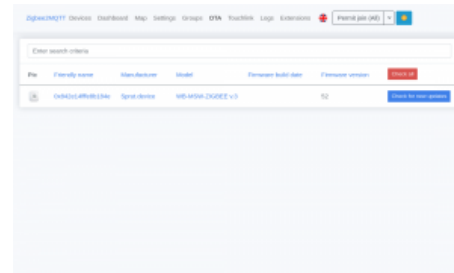
Устройство поддерживает обновление прошивок «по воздуху» (OTA). Здесь мы рассмотрим обновление при использовании датчика с zigbee2mqtt. Если вы используете SprutHub, смотрите инструкцию по обновлению в его документации.

Из веб-интерфейса zigbee2mqtt

Обновить прошивку можно через веб-интерфейс zigbee2mqtt:

1. Включите [веб-интерфейс zigbee2mqtt](#) и перейдите в него.

- Откройте раздел **OTA**.
- Найдите в списке устройств датчик WB-MSW-ZIGBEE v.3 и нажмите кнопку **Check for new updates**.
- Если прошивка обновление будет доступно, название кнопки сменится на **Update**, нажмите её.
- Начнётся процесс обновления прошивки, который длится около 30 минут, не отключайте питание датчика.



Раздел OTA в веб-интерфейсе zigbee2mqtt

Если в процессе прошивки произошёл сбой, например, устройство было обесточено, удалите устройство из Zigbee-сети и подключите его заново.

Через MQTT

Если вы используете zigbee2mqtt, но по каким-то причинам не хотите обновлять прошивку устройства через веб-интерфейс, вы можете это сделать с помощью отправки JSON-сообщений в MQTT с помощью MQTT Explorer. Название топиков и примеры JSON-команд, смотрите в инструкции [OTA updates \(https://www.zigbee2mqtt.io/guide/usage/ota_updates.html#automatic-checking-for-available-updates\)](https://www.zigbee2mqtt.io/guide/usage/ota_updates.html#automatic-checking-for-available-updates).

Примеры правил

Для работы в составе «умного дома» лучше использовать определенные правила. Их можно посмотреть в статье [Примеры правил](#)

Известные неисправности

[Аппаратные ошибки/особенности WB-MSW v.3](#), найденные при эксплуатации устройства.

Ревизии устройства

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

Ревизия	Партии	Дата выпуска	Отличия от предыдущей ревизии
1.3	v1.3B	12.2021 - ...	<ul style="list-style-type: none"> наклейки Wiren Board. Партия на базе MSW v4.19A/v4.19A/M.
1.3	v1.3A	08.2021 - 11.2021	<ul style="list-style-type: none"> версия с кнопкой для программирования. Со спец.наклейками.
1.2	v1.2A, v1.2B	04.2021 - 07.2021	<ul style="list-style-type: none"> Первая версия, на чипе MGM210P032JIA2. Версия со спец.наклейками.

Изображения и чертежи устройства

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

Corel Draw PDF: [Файл:WB-MSW-v.3.cdr.pdf](#)

Autocad 2013 DXF: [Файл:WB-MSW-v.3.dxf.zip](#)

Autocad PDF: [Файл:WB-MSW-v.3.pdf](#)

■



Универсальный датчик WB-MSW
v.3,
вид сзади

■



Плата WB-MSW

Подключение устройств Zigbee к контроллеру Wiren Board

Contents

Описание

zigbee2mqtt

[Описание](#)

[Установка](#)

[Установка с помощью apt](#)

[Сборка из исходников](#)

[Конфигурация](#)

[Управление сервисом](#)

[Добавление устройств](#)

[Веб-интерфейс \(Frontend\)](#)

wb-zigbee2mqtt

[Описание](#)

[Принцип работы](#)

[Установка](#)

[Правильная настройка zigbee2mqtt](#)

[Добавление устройств](#)

[Управление устройствами](#)

Описание

Для использования Zigbee-устройств с контроллером [Wiren Board](#) понадобятся:

1. Модуль расширения [WBE2R-R-ZIGBEE](#) или любой USB Zigbee стик.
2. Дополнительное ПО:
 - [zigbee2mqtt](#) — мост, с помощью которого можно управлять Zigbee-устройствами через MQTT.
 - [wb-zigbee2mqtt](#) — конвертер из [zigbee2mqtt](#) в Wiren Board MQTT Conventions.

zigbee2mqtt

Описание

[Zigbee2mqtt](#) — это мост, с помощью которого вы можете управлять Zigbee устройствами через MQTT. Он работает на Node.js и позволяет интегрировать Zigbee устройства в различные системы автоматизации: Home Assistant, Node Red и тд. Официальный сайт — [zigbee2mqtt \(https://www.zigbee2mqtt.io/\)](https://www.zigbee2mqtt.io/).

Установка

Установка с помощью apt

Так как пакет в свой репозиторий мы собираем сами, его версия может отличаться от текущей актуальной версии [zigbee2mqtt](#).

Запустите консоль и введите команду:

```
apt update && apt install -y zigbee2mqtt
```

Сборка из исходников

zigbee2mqtt версии 1.22 и выше может не работать с прошивкой модулей WBE2R-R-ZIGBEE, выпущенных до конца 2021 года. Такие модули нужно прошить свежей прошивкой.

Это альтернативный метод установки, который позволяет установить актуальную версию zigbee2mqtt. Оригинальная инструкция (https://www.zigbee2mqtt.io/guide/installation/01_linux.html).

1. Установите необходимые компоненты

```
apt install -y nodejs git make g++ gcc
```

2. Скопируйте файлы из репозитория zigbee2mqtt:

```
git clone https://github.com/Koenkk/zigbee2mqtt.git /mnt/data/root/zigbee2mqtt
```

3. Настройте права на папку:

```
chown -R root:root /mnt/data/root/zigbee2mqtt
```

4. Перейдите в папку с исходниками zigbee2mqtt:

```
cd /mnt/data/root/zigbee2mqtt
```

5. Запустите установку:

```
npm ci
```

В процессе установки будет показано несколько предупреждений. Их можно игнорировать.

Настройте автоматический запуск zigbee2mqtt, для этого создайте сервис:

1. Создайте новый файл и откройте его на редактирование:

```
mcedit /etc/systemd/system/zigbee2mqtt.service
```

2. Скопируйте в него следующий текст:

```
[Unit]
Description=zigbee2mqtt
After=network.target

[Service]
ExecStart=/usr/bin/npm start
WorkingDirectory=/mnt/data/root/zigbee2mqtt
StandardOutput=inherit
StandardError=inherit
Restart=always
User=root

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

3. Сохраните файл.

4. Разрешите автозапуск сервиса командой:

```
systemctl enable zigbee2mqtt.service
```

Конфигурация

Если вы используете модуль расширения WBE2R-R-ZIGBEE, убедитесь, что правильно выставили модуль в **Settings** → **Configs** → **Hardware Modules Configuration**.

1. Откройте файл configuration.yaml для редактирования:

```
mcedit /mnt/data/root/zigbee2mqtt/data/configuration.yaml
```

2. Приведите его к виду

```
homeassistant: false
permit_join: false
mqtt:
  base_topic: zigbee2mqtt
  server: 'mqtt://localhost'
serial:
  port: /dev/ttyMOD3
advanced:
  rtscts: false
  last_seen: epoch
```

в параметре **port** укажите порт, куда физически вставлен модуль.

3. Сохраните и закройте файл.

4. Перезапустите сервис zigbee2mqtt:

```
systemctl restart zigbee2mqtt
```

Теперь вы можете установить конвертер wb-zigbee2mqtt и Zigbee-устройства можно будет добавить на вкладку **Devices**.

Управление сервисом

Сервис запускается по умолчанию, но вы можете управлять им с помощью команд:

```
systemctl start zigbee2mqtt # запуск
systemctl stop zigbee2mqtt # остановка
systemctl restart zigbee2mqtt # перезапуск
systemctl status zigbee2mqtt # получить статус
```

Добавление устройств

Если вы не используете wb-zigbee2mqtt, то для добавления нового устройства в настройках zigbee2mqtt включите параметр **permit_join**:

```
permit_join: true
```

Далее следуйте инструкции к устройству. В основном требуется нажать кнопку «pair» на несколько секунд и дождаться «мигания» светодиода. После этого устройство должно передать информацию о себе на контроллер. Иногда требуется дополнительно нажимать на кнопку каждую секунду, что бы устройство не «уснуло». После успешного добавления устройство автоматически появится во вкладке Devices.

Более подробная информация на [сайте zigbee2mqtt \(https://www.zigbee2mqtt.io/\)](https://www.zigbee2mqtt.io/).

Веб-интерфейс (Frontend)

Веб-интерфейс zigbee2mqtt потребляет ресурсы контроллера — не включайте его, если не используете: добавлять устройства удобно через wb-zigbee2mqtt, а удалять можно из файла конфигурации вручную.

Настройка веб-интерфейса zigbee2mqtt:

1. Откройте файл configuration.yaml для редактирования:

```
mcedit /mnt/data/root/zigbee2mqtt/data/configuration.yaml
```

2. Добавьте в него блок:

```
frontend:
  port: 8081
  host: 0.0.0.0
```

3. Сохраните и закройте файл.

4. Перезапустите сервис zigbee2mqtt:

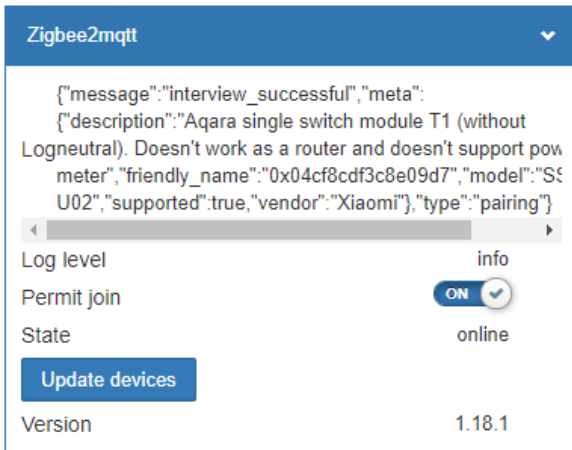
```
systemctl restart zigbee2mqtt
```

Веб-интерфейс моста zigbee2mqtt будет доступен по адресу `http://wb-ip-address:PORT`, например, если IP-адрес контроллера **192.168.42.1**, а порт указан **8081**, то веб-интерфейс будет доступен по адресу `http://192.168.42.1:8081`

wb-zigbee2mqtt

Описание

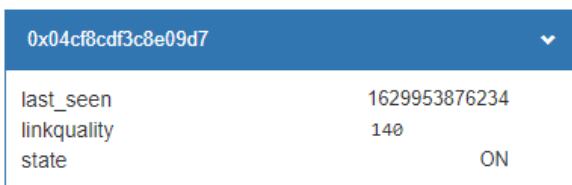
Wb-zigbee2mqtt — это простой конвертер из zigbee2mqtt в Wiren Board MQTT Conventions. Конвертер написан на движке правил `wb-rules` и устанавливается в папку с системными правилами `wb-rules-system`.



The screenshot shows the Zigbee2mqtt interface with a dropdown menu for a device. The device details are as follows:

Log level	info
Permit join	<input checked="" type="checkbox"/>
State	online
Update devices	<input type="button" value="Update devices"/>
Version	1.18.1

Добавление устройства в `wb-zigbee2mqtt`.
Найдено устройство **Aqara single switch module T1 (without neutral)**



The screenshot shows the Zigbee2mqtt interface with a dropdown menu for a device. The device statistics are as follows:

last_seen	1629953876234
linkquality	140
state	ON

Новое Zigbee-устройство на вкладке Devices — в заголовке указан `friendly_name`

Принцип работы

Конвертер `wb-zigbee2mqtt` запрашивает у `zigbee2mqtt` список всех подключенных устройств, подписывается на их топике. При получении данных от физических устройств конвертер парсит json, создает виртуальное устройство и посылает в него полученную информацию. Так же конвертер создает виртуальное устройство «`zigbee2mqtt`» и передает в него логи, версию и состояние (`online\offline`) моста `zigbee2mqtt`. Можно включить и выключить режим добавления устройств. Также там расположена кнопка «`update devices`». При нажатии на эту кнопку конвертер заново запрашивает у моста список добавленных устройств и если они появились — подписывается на них.

Установка

Для установки откройте консоль и введите команду:

```
apt update && apt install wb-zigbee2mqtt
```

Если вы собирали zigbee2mqtt версии 1.22 и новее из исходников, то не устанавливайте пакет, а положите этот скрипт (<https://github.com/wirenboard/wb-zigbee2mqtt/blob/f358d584a1a2dff3bc48bc588a307d99b78f4cde/wb-zigbee2mqtt.js>) в папку `/usr/share/wb-rules-system/rules/`.

Правильная настройка zigbee2mqtt

Если вы установили zigbee2mqtt через apt, то мост уже настроен для правильной работы с конвертером и этот шаг можно пропустить.

Если вы установили zigbee2mqtt исходников, то убедитесь, что параметр базового топика выставлен верно:

1. Откройте файл `configuration.yaml`

```
mcedit /mnt/data/root/zigbee2mqtt/data/configuration.yaml
```

2. Проверьте строки:

```
mqtt:
  base_topic: zigbee2mqtt
  server: 'mqtt://localhost'
```

другие настройки на работу конвертера не влияют.

Добавление устройств

Устройства можно добавлять по одному или все разом. Чтобы добавить новые устройства:

1. Перейдите в веб-интерфейс контроллера.
2. Откройте вкладку **Devices**.
3. Найдите там карточку устройства **Zigbee2mqtt** и проверьте статус в поле **State** — должно быть написано **online**.
4. Включите переключатель **Permit join**.
5. Следуйте инструкции к Zigbee-устройству: в основном требуется зажать на устройстве кнопку **pair** на несколько секунд и дождаться «мигания» светодиода.
6. Если устройство будет добавлено, то в карточке устройства **Zigbee2mqtt**, в логе появится новая строка с именем устройства.
7. Повторите шаг 5 для каждого Zigbee-устройства, которое у вас есть.
8. Завершите добавление устройств, для этого выключите переключатель **Permit join**.
9. Нажмите на кнопку **Update devices**.

Добавленные устройства появятся на вкладке **Devices**.

Управление устройствами

Управление устройствами через конвертер wb-zigbee2mqtt пока не реализовано. Но управлять устройствами можно стандартными инструментами zigbee2mqtt (https://www.zigbee2mqtt.io/information/mqtt_topics_and_message_structure.html#zigbee2mqttfriendly_nameset) через wb-rules (функция `publish`)

Пример управления устройством в правилах `wb-rules`:

```
publish("zigbee2mqtt/0x00158d0001f3fc34/set", JSON.stringify({ state: "ON" }), 2, false);
publish("zigbee2mqtt/0x00158d0001f3fc34/set", JSON.stringify({ state: "OFF" }), 2, false);
```


Покраска корпуса MSW

Белые корпуса легко покрасить, получается неплохо. Для этого потребуется:

2 аэрозольных балончика:

- Грунт (желательно для пластика, но если нет подойдет обычный)
- Эмаль (Черная матовая или другого цвета).
- Наждачная бумага (~1000 grit, водостойкая).

Разберите датчик:

- Вытащите плату
- Аккуратно отклейте наклейку с корпуса, что бы не осталось клея. Остатки оттереть спиртом.
- Открутите линзу Френеля с обратной стороны крышки датчика.

Далее:

1. Обработайте корпус наждачной бумагой под теплой водой. Постарайтесь пройтись по всей окрашиваемой поверхности. От этого зависит качество покраски. Не оставляйте глубоких царапин.
2. Высушите корпус от воды. Лучше не используйте салфетки - они могут оставить ворс.
3. Обезжирьте окрашиваемую поверхность корпуса спиртом или уайт-спиритом, после этого старайтесь не прикасаться к ней руками.
4. (Желательно на свежем воздухе) Нанесите грунт в несколько тонких слоев так, что бы он перекрыл всю видимую поверхность.
5. После высыхания грунта снова обработать наждачной бумагой под теплой водой. Не перестарайтесь - если вы сотрете грунт до пластика, придется грунтовать заново. Нужно убрать неровности грунта и возможные пылинки. После обработки поверхность должна быть ровная, гладкая.
6. Снова высушите и обезжирьте поверхность
7. Нанесите краску (эмаль). Инструкцию по нанесению эмали читайте на балончике. Не заливайте сильно - потеки могут всё испортить.

Если датчиков много или нет желания заниматься всем этим, то можно обратиться к профессионалам в ближайший хороший автосервис, где занимаются покраской автомобилей. Детали маленькие, поэтому это не должно стоить дорого.



Окрашенный корпус MSW

Движок правил wb-rules

Описание

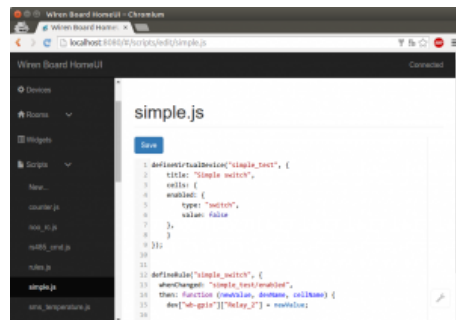
Читайте полное описание движка правил на [Github \(https://github.com/wirenboard/wb-rules\)](https://github.com/wirenboard/wb-rules).

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](#).



Редактирование правил в веб-интерфейсе

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 \(http://forums.contactless.ru/t/dvizhok-pravil-primery-koda/483\)](http://forums.contactless.ru/t/dvizhok-pravil-primery-koda/483);
- в исходниках написанного на `wb-rules` конвертера `esphome2wb` (<https://github.com/wb-adeptyarev/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.

Программа для работы с MQTT

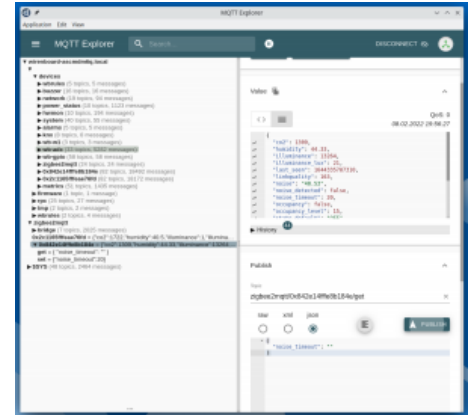
MQTT Explorer — это мультиплатформенный клиент MQTT, который предоставляет структурированный обзор ваших тем MQTT и упрощает работу с устройствами/сервисами вашего брокера.

Скачать можно на сайте <http://mqtt-explorer.com/> (<http://mqtt-explorer.com/>)

Как работать с MQTT контроллера читайте в статье [MQTT](#).

Параметры подключения к MQTT контроллера по умолчанию:

- Protocol — mqtt
- Host — IP-адрес вашего контроллера
- Port — 1883
- Username и Password — пусто.



Окно MQTT Explorer

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

- Privacy policy
- About Wiren Board
- Disclaimers
-