

[http://support.microline.ru/index.php%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%
B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BF%D1%80%D0%BE%D1%82%D0%
%BE%D0%BA%D0%BE%D0%BB%D0%B0_MQTT%D0%B2%D0%BA%D0%BE%D0%BD%D1%82%D0%
D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80%D0%B0%D1%85_ZONT](http://support.microline.ru/index.php%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%
B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BF%D1%80%D0%BE%D1%82%D0%
%BE%D0%BA%D0%BE%D0%BB%D0%B0_MQTT%D0%B2%D0%BA%D0%BE%D0%BD%D1%82%D0%
D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80%D0%B0%D1%85_ZONT)

Использование протокола MQTT в контроллерах ZONT

[Наверх](#)



Содержание

- [1 О документе](#)
- [2 Общие сведения](#)
 - [2.1 Назначение и применение MQTT](#)
 - [2.2 Область применения](#)
 - [2.3 Основные понятия MQTT](#)
- [3 Применение протокола MQTT в контроллерах ZONT](#)
 - [3.1 Настройка Брокера \(MQTT сервера\)](#)
 - [3.2 Публикация данных](#)
 - [3.3 Подписка](#)
 - [3.4 Формат поля cmd для разных типов объектов](#)
- [4 Интеграция с Home Assistant](#)
 - [4.1 Список поддерживаемых типов объектов](#)
 - [4.2 Решения для Home Assistant \(примеры\)](#)
 - [4.2.1 Пример добавления датчика из ZONT в НА](#)
 - [4.2.2 Пример трансляции значения датчика из НА в ZONT](#)

О документе

Уважаемые пользователи!

В настоящем документе приведена информация по организации обмена данными контроллера ZONT со сторонним оборудованием по протоколу **MQTT**.

Обращаем Ваше внимание на то, что настоящий документ постоянно обновляется и корректируется. Это связано с разработкой и применением новых функций, которые не использовались ранее. В связи с этим тексты некоторых разделов могут изменяться и/или дополняться, а некоторые иллюстрации и скриншоты, представленные в документе, могут устареть.

Если Вы обнаружили ошибки и/или неточности — пожалуйста, сообщите нам на e-mail: support@microline.ru.

Актуальная версия документа также доступна на сайте [zont.online](#) в разделе "Поддержка".

[Техническая документация](#)". Документ доступен для чтения и скачивания в формате *.pdf.

Общие сведения

Назначение и применение MQTT

MQTT (Message Queue Telemetry Transport) – это протокол обмена сообщениями, который часто используется в IoT (интернете вещей) для организации связи между устройствами и серверами.

Область применения

MQTT используется в системах умного дома для решения различных задач:

Управление устройствами: Лампы, терmostаты, жалюзи, датчики движения и другие устройства могут отправлять и получать команды через MQTT. Например, терmostат может публиковать текущую температуру в доме, а также принимать команды на изменение установленной температуры.

Датчики: MQTT может быть использован датчиками для передачи данных в центральную систему. Это могут быть датчики температуры, влажности, освещенности, движения и т. д.

Уведомления: Система умного дома может использовать MQTT для отправки уведомлений пользователю о различных событиях, таких как обнаружение движения, открытие двери или протечка воды.

Интеграция с внешними службами: MQTT может быть мостом между устройствами умного дома и облачными сервисами или другими внешними системами. Например, умный дом может автоматически включать или прерывать полив, если прогноз погоды сообщает о дожде.

Межустройстvenное взаимодействие: Устройства могут общаться друг с другом через MQTT, чтобы координировать свои действия. Например, когда вы включаете режим охраны, уходя из дома, свет и другие потребители электроэнергии могут автоматически выключаться.

Интеграция оборудования разных производителей: Благодаря стандартизации MQTT устройства разных производителей могут легко взаимодействовать друг с другом.

Безопасность: С использованием SSL/TLS MQTT может обеспечить шифрование и аутентификацию, что критично для умных домов, чтобы обеспечить приватность и безопасность пользователей.

Одним из популярных решений на основе MQTT для умного дома является **Home Assistant**, платформа автоматизации дома с открытым исходным кодом, которая поддерживает MQTT и множество других протоколов и технологий.

Основные понятия MQTT

MQTT имеет клиент-серверную архитектуру. Обмен сообщениями происходит через центральный сервер (Broker – Брокер). Клиенты не могут общаться напрямую друг с другом, и весь обмен данными происходит через Брокера.

Клиенты могут выступать в роли поставщиков данных (Publisher – Издатель) и в роли получателей данных (Subscriber – Подписчик).

Стандартный порт MQTT-брокера для нешифрованных входящих TCP-соединений – 1883, а для использования защищенного SSL-подключения – 8883. Последний требует настройки использования клиентом и брокером сертификатов шифрования соединения.

Брокер – это центральный узел MQTT, обеспечивающий маршрутизацию сообщений между клиентами. Обмен данными между клиентами происходит только через брокера. В качестве брокера может выступать серверное ПО или контроллер. В его задачи входит получение сообщений от клиентов, временное хранение и доставка данных клиентам, контроль за доставкой сообщений.

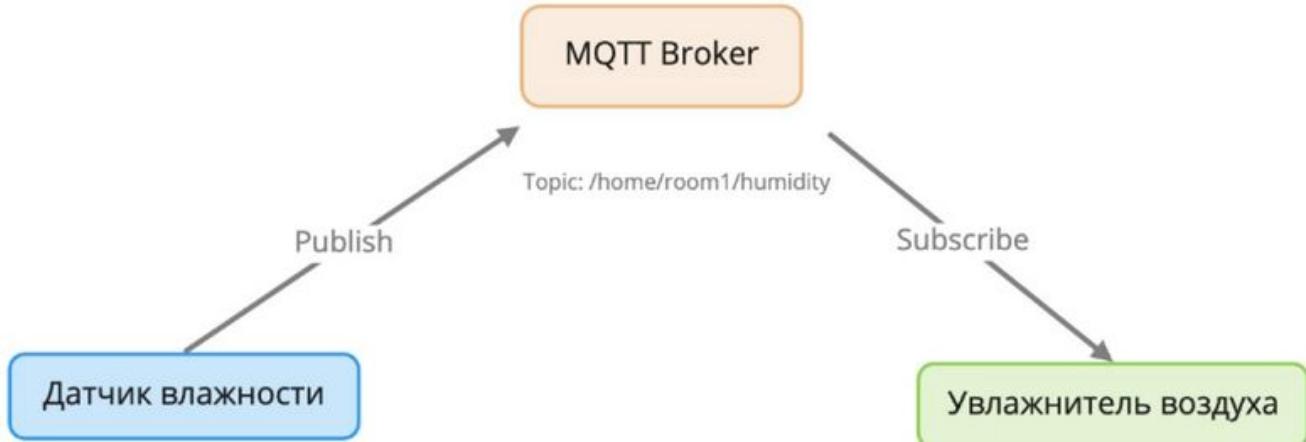
Издатели / Подписчики – устройства интернета вещей выступающие в качестве клиентов и взаимодействующие с брокером напрямую. Издатели публикуют данные MQTT, например рассылают текущие параметры окружающей среды, а Подписчики эти данные используют. В частности, в роли подписчика выступает увлажнитель воздуха в системе умного дома: датчик влажности публикует свои показатели, а увлажнитель на их основе регулирует интенсивность своей работы.

Топик (канал) – предназначен для разделения сообщений. Это удобный механизм, позволяющий называть датчики. Хорошей практикой организации топиков является разделение по уровням от общего к частному. Например:

```
# Датчик температуры на кухне  
home/kitchen/temperature  
  
# Датчик температуры в спальне  
home/sleeping-room/temperature  
  
# Датчик освещенности на улице  
home/outdoor/light
```

В схеме MQTT-клиенты не знают о существовании друг друга, и не взаимодействуют напрямую.

Брокер может получать данные из разных источников, возвращать подписчикам.



Publisher посылает данные брокеру, Subscriber подписывается на обновления этих данных

В протоколе MQTT гарантия доставки определяется параметром QoS (Quality of service), который указывается при отправке сообщения:

- 0 – максимум один раз. Подтверждение доставки не производится. Подходит для передачи телеметрической информации от устройств, потеря которой не критична.
- 1 – хотя бы один раз. Производится однократное подтверждение доставки, но при нестабильном соединении возможно дублирование переданного сообщения. Подходит для отправки важной телеметрии или команд, которые устанавливают конкретное значение.
- 2 – ровно один раз. Производится несколько подтверждений доставки для исключения дублирования. Подходит для инициализации действий повторение которых недопустимо или команд на увеличение/уменьшение значений параметров. Наиболее затратный тип доставки по времени и использованию трафика.

Применение протокола MQTT в контроллерах ZONT

Протокол MQTT поддерживается контроллерами ZONT, модели [H700+ PRO.V2](#), [H1000+ PRO.V2](#), [H1500+ PRO.V2](#), [H2000+ PRO.V2](#), C2000+ PRO, начиная с 420-ой версии прошивки.

Обмен данными контроллера с другими устройствами по протоколу MQTT работает ТОЛЬКО по каналам связи Wi-Fi и Ethernet.

Примечание: По GSM каналу (мобильная сеть) MQTT не работает.

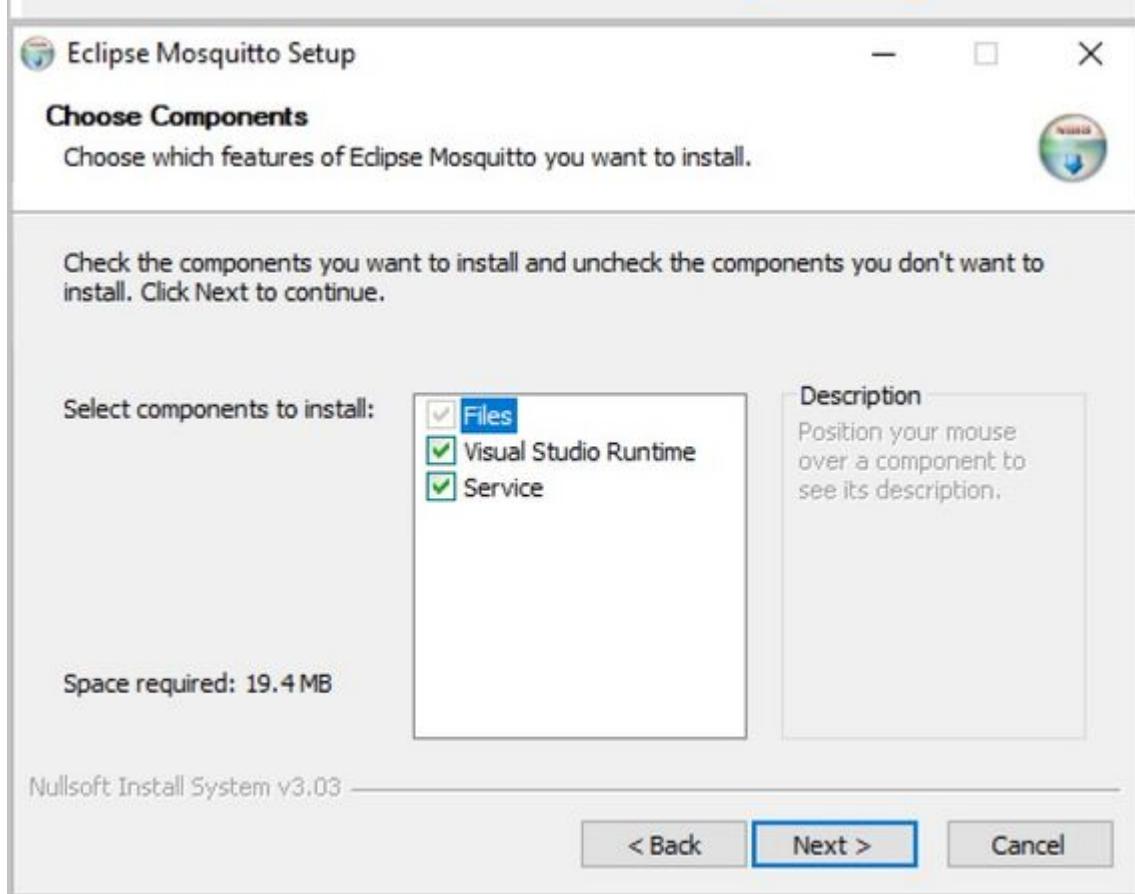
Настройка Брокера (MQTT сервера)

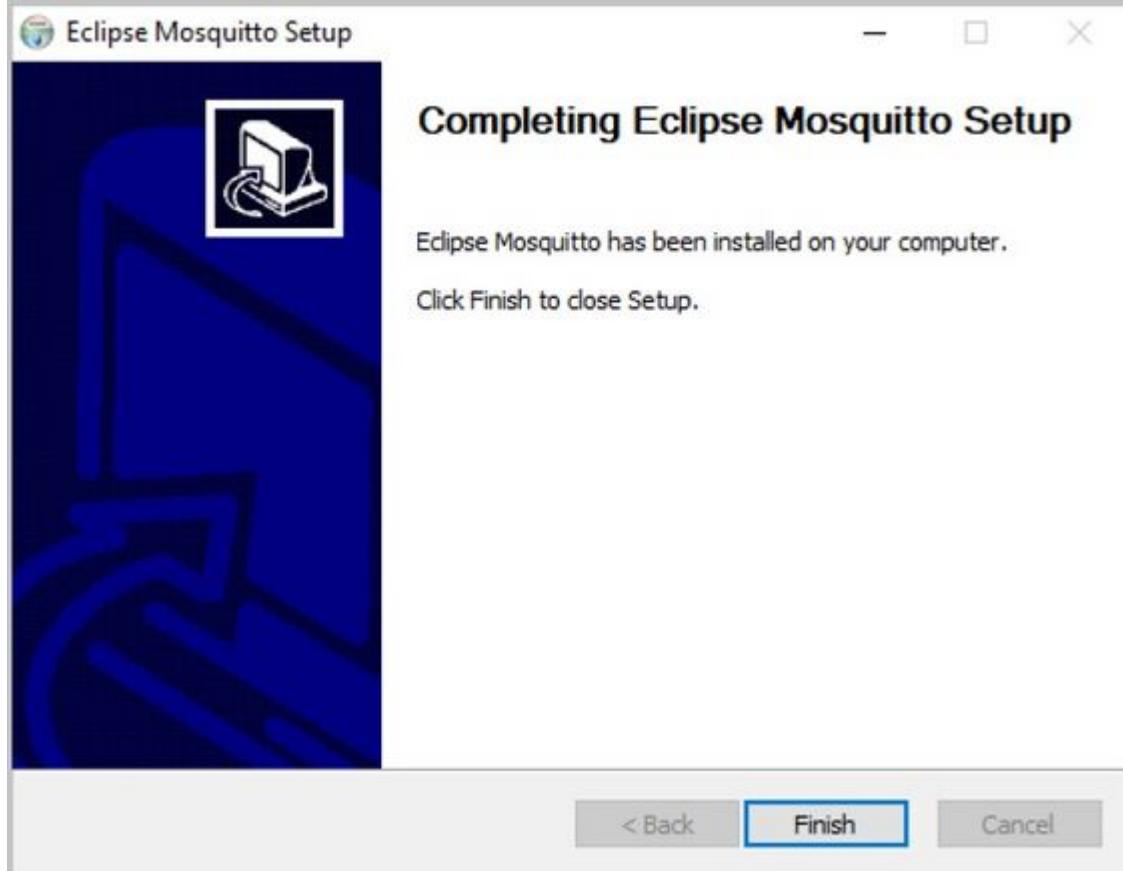
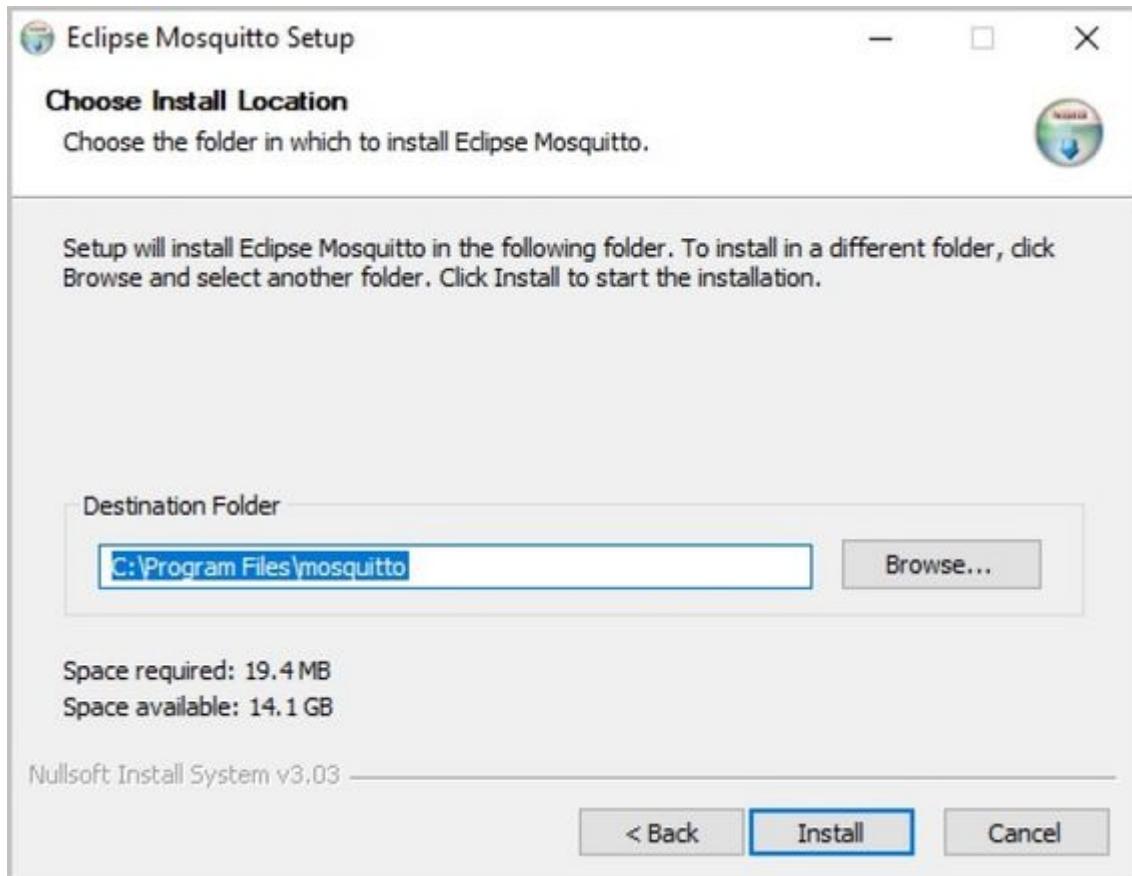
Запуск собственного MQTT брокера:

В этой инструкции мы возьмём популярный MQTT-брюкер с открытым исходным кодом - Mosquitto. Предполагаем, что наш компьютер находится в одной локальной сети с подключаемыми устройствами.

Настройка на Windows

- Скачать установщик Windows для 32 или 64 битной системы с <https://mosquitto.org/download/> и выполнить установку приложения





- Открыть файл C:\Program Files\mosquitto\mosquitto.conf и добавить в самый конец файла:
log_type all
listener 1883

```
protocol mqtt
```

```
allow_anonymous true
```

- Открыть Диспетчер задач -> вкладка Сервисы -> в контекстном меню строки mosquitto выбрать Перезагрузить
- Открыть Брандмауэр Windows и создать правило для Входящих соединений:

Тип правила: Порт

Протокол: TCP, Указать порт: 1883

Действие: Разрешить соединение

Профиль: выбрать все три

Имя: BrokerMQTT

- Теперь брокер доступен в локальной сети на порту 1883
- Открыть меню Пуск -> Служебные - Windows -> Командная строка
- Узнать IP адрес компьютера можно выполнив в командной строке: ipconfig | findstr "IPv4"
- Ссылкой для подключения тогда будет:

mqtt://{{IP-адрес компьютера в локальной сети}}:1883

Например, mqtt://192.168.10.273:1883

Настройка на Debian, Ubuntu

- Выполнить в терминале:

```
sudo apt update
```

```
sudo apt upgrade -y
```

```
sudo apt install -y mosquitto
```

* Для других дистрибутивов пакет mosquitto должен быть доступен в соответствующих им репозиториях. Также имеется возможность использовать Docker контейнер:
https://hub.docker.com/_/eclipse-mosquitto

- В папке /etc/mosquitto/conf.d/ создать файл local.conf со следующим содержимым:

```
log_type all
```

```
listener 1883
```

```
protocol mqtt
```

```
allow_anonymous true
```

- Перезагрузить сервис выполнив: sudo systemctl restart mosquitto
- Теперь брокер доступен в локальной сети на порту 1883

- Узнать IP адреса компьютера можно выполнив: hostname -I
- Ссылкой для подключения тогда будет:

mqtt://{{IP-адрес компьютера в локальной сети}}:1883

Например, mqtt://192.168.10.29:1883

Пример настройки адреса MQTT сервера в сервисе ZONT:

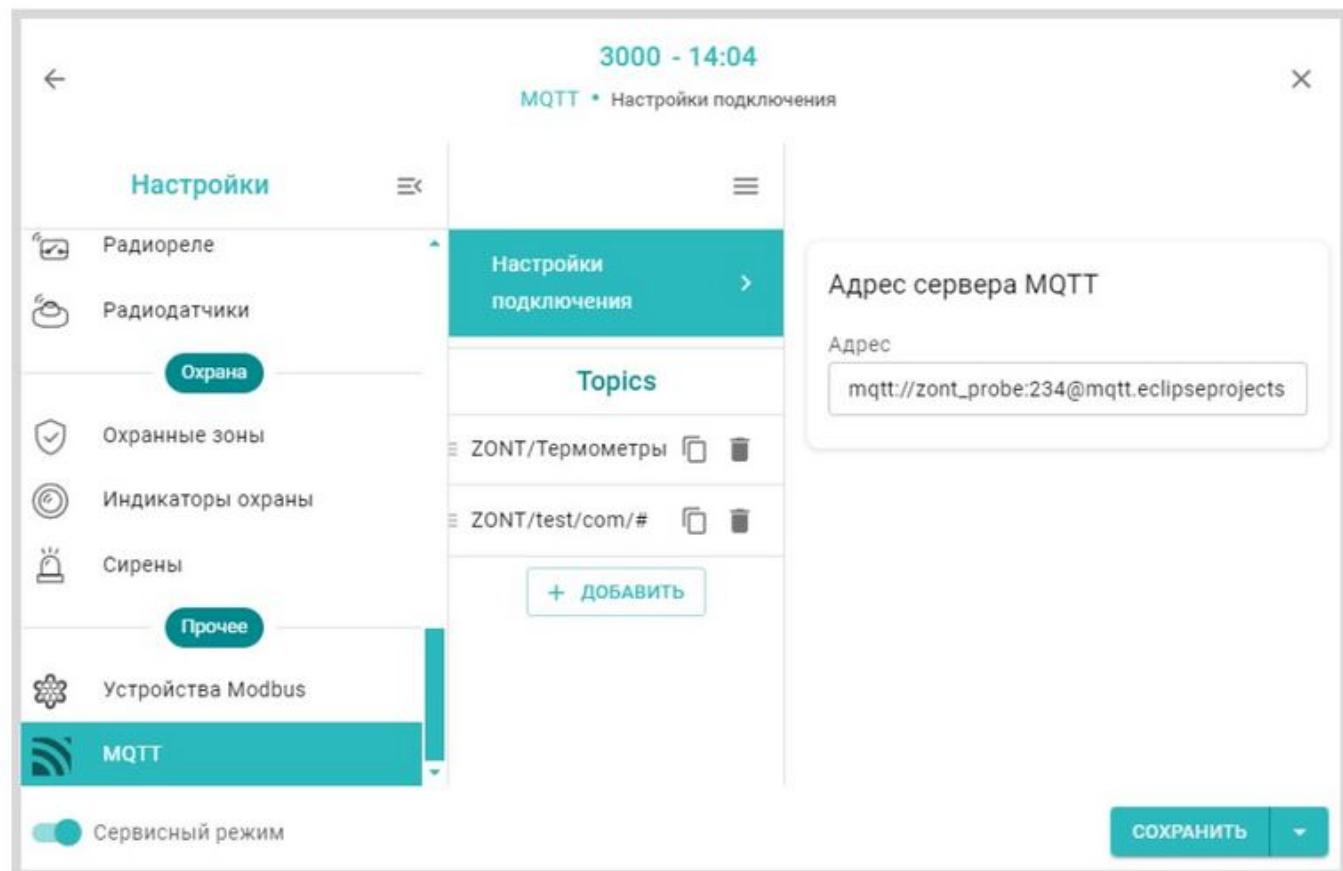
mqtt://username:password@mqtt.eclipseprojects.io:1883

Адрес : mqtt.eclipseprojects.io

Порт: 1883

Имя пользователя: username

Пароль: password



Примечание: Если имя пользователя и пароль не используются, то их можно не указывать:

mqtt://mqtt.eclipseprojects.io:1883

Публикация данных

Имя топика формируется из имени настройки топика/имя объекта.

Формат передаваемых данных:

Датчик

```
{  
    "v":12.3,  
    "a":1  
}
```

где:

v - напряжение Вольты (значение float)

a - доступность датчика (0-не доступен, 1-доступен)

Термодатчик

```
{  
    "t":23.5,  
    "a":1  
}
```

где:

t - температура по цельсию (значение float)

a - доступность датчика (0-не доступен, 1-доступен)

Радио термодатчик

```
{  
    "t": 23.5,  
    "a": 1,  
    "h": 85,  
    "b": 100,  
    "r": 78  
}
```

где:

t - температура по цельсию (значение float)

a - доступность датчика (0-не доступен, 1-доступен)

h - влажность

b - уровень заряда батареи

r - rssи уровень сигнала

Отопительный контур

```
{  
  "s": 23.5,  
  "c": 1  
}
```

где:

s - целевая температура

c - текущая температура

Реле

```
{  
  "s": 1  
}
```

где:

s - текущее состояние (0 - выкл, 1 - вкл)

Насос

```
{  
  "s": 1  
}
```

где:

s - текущее состояние (0 - выкл, 1 - вкл)

Смеситель

```
{  
  "s": 1  
}
```

где:

s - текущее состояние (0 - не активен, 1 - открытие, 2 - закрытие)

Пользовательские элементы управления

```
{  
  "s": 1,  
  "t": "Выключено"  
}
```

где:

s - текущее состояние

(0 - не активен, 1 - активен, > 0 - значения для аналогового регулятора)

t - текстовое представление статуса

(название из настроек для активного или неактивного состояния)

Адаптер цифровой шины котла

```
{  
    "water": 45.6,  
    "dhw": 34.5,  
    "return": 30.4,  
    "modul": 99,  
    "press": 2.4,  
    "state": 1,  
    "err": 0  
}
```

где:

water - температура теплоносителя

dhw - температура ГВС

return - температура обратки

modul - уровень модуляции %

press - давление теплоносителя

state - состояние котла (0-выкл 1-работает 2-ошибка)

err - код ошибки

Подписка

Для управления устройством в получаемом сообщении должен быть выбран объект, которому предназначена команда.

Выбрать объект можно одним из трех способов:

- Заданием id объекта в поле данных сообщения.

```
{  
    "Id": 12345,  
    "cmd": "$command"  
}
```

- Заданием имени объекта в поле данных сообщения.

```
{  
  "name": "Кнопка",  
  "cmd": "$command"  
}
```

- Заданием имени объекта в поле имени topic.

Пример имени топика:

ZONT/Дом/Управление/Кнопка

В настройках можно указать для подписки все подтопики топика обычным способом:

ZONT/Дом/Управление/#

Формат поля cmd для разных типов объектов

Код	Команда	Описание
Аналоговый вход		
0 N	Установка значения N	Для аналогового входа можно установить значение, для использования этой возможности в настройках входа не следует назначать привязку к реальному входу. (N - значение x 0,1 вольт)
1 N	Установка значения N	Для аналогового входа можно установить значение, для использования этой возможности в настройках входа не следует назначать привязку к реальному входу. (N - значение float)
Аналоговый термодатчик		
N	Установка температуры N = 2945)	N - значение в деци кельвинах (пример: 21.5 гр -> 2730 + 215 = 2945)
0 N	Установка температуры N = 2945)	N - значение в деци кельвинах (пример: 21.5 гр -> 2730 + 215 = 2945)
1 N	Установка температуры N	N - значение в градусах цельсия float
Цифровой термодатчик (предпочтительнее использовать чем аналоговый)		
N	Установка температуры N = 2945)	N - значение в деци кельвинах (пример: 21.5 гр -> 2730 + 215 = 2945)
0 N	Установка температуры N = 2945)	N - значение в деци кельвинах (пример: 21.5 гр -> 2730 + 215 = 2945)
1 N	Установка температуры N	N - значение в градусах цельсия float
Охранная зона		
0	Снятие с охраны	
1	Постановка на охрану	
2	Инверсия охраны	
Оповещение		
0	Выполнить	Производится настроенное оповещение
Действие с выходом		
0	Выполнить	Выполняется действие с выходом
Трёхходовой кран		

- 1 Полностью открыть
- 2 Полностью закрыть
- 3 Открыть на шаг
- 4 Закрыть на шаг

Контур отопления

T Установка температуры команда это значение температуры в децибеллинах
Пример: {"cmd":3000}

Режим терморегулирования

- 0 Установка режима

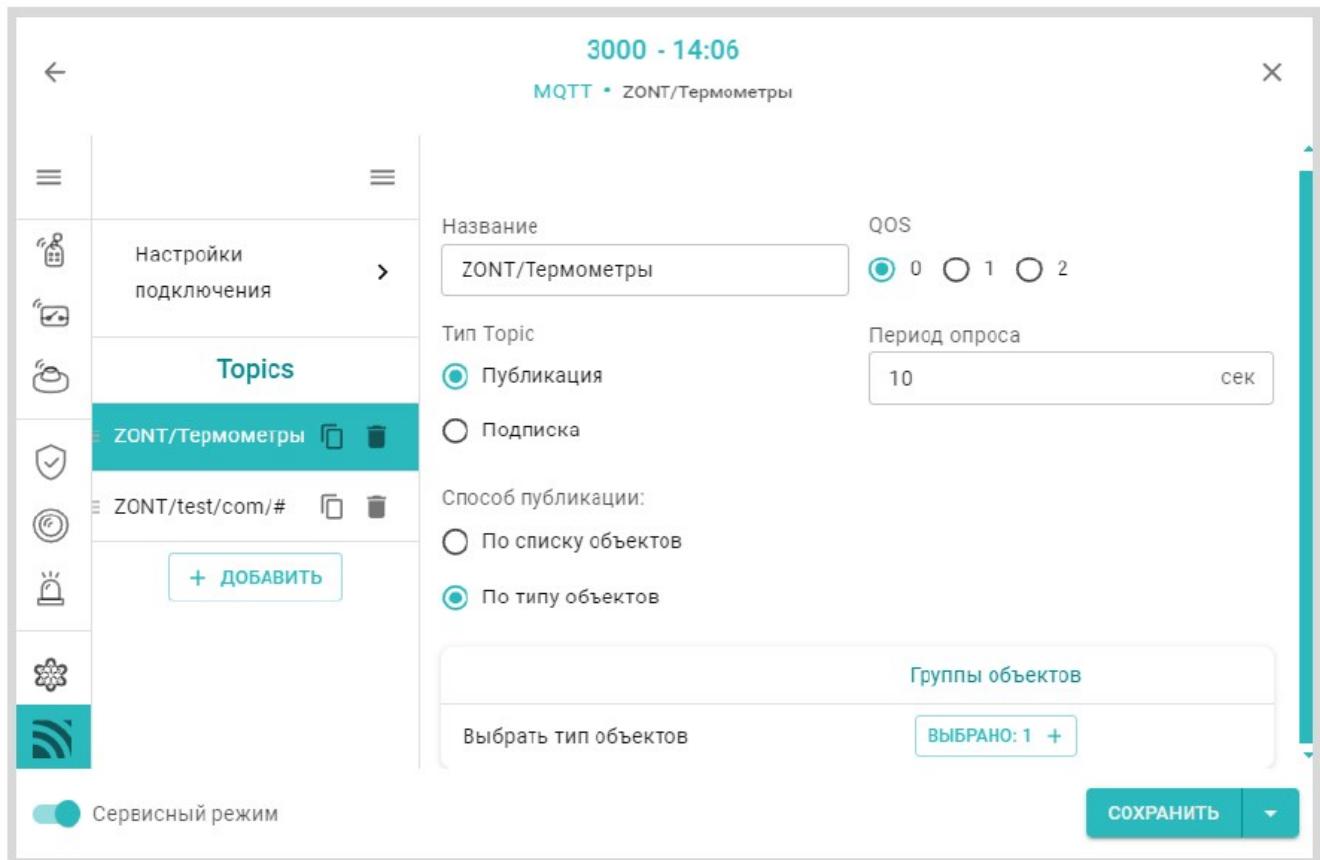
Элемент управления (кнопки)

- 0 Действия по выключению Выполнение действий веб элемента
- 1 Действия по включению Выполнение действий веб элемента

Сирена / Индикатор / Реле / Насос

- 0 Выключение
- 1 Включение

Вид интерфейса настройки топиков



Интеграция с Home Assistant

Прежде чем приступить к интеграции, необходимо убедиться, что в профиле в Home Assistant включен "расширенный режим".

В настройках топика устройства ZONT необходимо активировать формат Home Assistant.

The screenshot shows the ZONT MQTT configuration page for a device named 'Устройство Topic'. It includes fields for QoS (set to 0), publication period (10 seconds), and object selection methods (set to 'By object list'). A red arrow points to the 'format Home Assistant' checkbox, which is described in a tooltip as enabling value transmission to the Home Assistant system.

При использовании этой опции соответствующие элементы из ZONT автоматически отображаются и синхронизируются с интерфейсом Home Assistant.

формат Home Assistant  

Для передачи значений в систему Home Assistant. В этом случае имя настройки не будет соответствовать имени передаваемого топика. Имена топиков для передачи и подписки будут автоматически соответствовать правилам НА.

Список поддерживаемых типов объектов

Объект ZONT

Датчик

Компонент НА

Sensor

Датчик температуры	Sensor
Контур отопления	Climate
Элемент пользователя - статус	Binary_sensor
Элемент пользователя - простая кнопка	Button
Элемент пользователя - сложная кнопка	Switch
Элемент пользователя - аналоговый регулятор	Valve
Насос	Switch
Реле	Switch

Примечание: В именах топиков допускается использование русских и английских букв, цифр, пробелов и других стандартных ASCII символов.

Не допускается использование нестандартных символов (например символа градусов °), в этом случае соединение с брокером может быть разорвано.

Решения для Home Assistant (примеры)

Пример добавления датчика из ZONT в HA

Запись в файл конфигурации HA:

```
mqtt:
sensor:
- name: "MQTT_Миша"
state_topic: "HA/ZONT/TS/TD/DT_Миша"
suggested_display_precision: 1
unit_of_measurement: "°C"
value_template: "{{ value_json.t }}"
```

Пояснения:

name - имя какое хотим

suggested_display_precision - знаки после запятой

state_topic - имя топика

value_template: "{{ value_json.t }}"

"value_json." - способ обработки сообщения

t - имя ключа в json по которому получаем значение

Пример трансляции значения датчика из HA в ZONT

Создается автоматизация – Настройки – Автоматизация и сцены – Создать – Когда – ставим условие, на которое будет срабатывать отправка в топик

Шаблон значения:

```
"cmd": {{ ((states('sensor.datchik_temperatury_spalnia_temperature') | float) * 10 + 2730 ) | int } | string }}
```

тут что в {{*}}

states('sensor.datchik_temperatury_spalnia_temperature') - имя сенсора (датчика)

внутри НА

| float |int |string переводы по типам переменных

```
"cmd": { 1 (states('sensor.datchik_temperatury_spalnia_temperature')) | string }}
```

