

Содержание

Начало работы с MQTT.....	2
1. Описание работы.....	2
2. Развертывание брокера.....	2
2.1 Локальный брокер mosquitо.....	2
2.2 Облачный брокер cloudMQTT.....	2
3 Подключение к брокеру.....	2
3.1 Подключение ERD-4 к брокеру.....	2
3.2 Подключение клиента к брокеру.....	3
4 Инструкции и примеры.....	4

Начало работы с MQTT

1. Описание работы

По [ссылке](#) доступно описание и принцип работы протокола MQTT.ERD обеспечивает работу с данным протоколом в двух режимах: API и режим работы с приложением IoT manager. Оба режима включены по умолчанию и работают независимо друг от друга.

2. Развертывание брокера

Брокер - это сервер, обеспечивающий передачу сообщений от издателей к подписчикам. Подходит любой брокер. Есть несколько проверенных вариантов развертывания брокера:

2.1 Локальный брокер [mosquito](#).

Инструкция по установке для Windows доступна в разделе [4.1 Инструкция по установке на mosquito для Windows](#). Инструкция по установке для Debian доступна в разделе [4.2 Инструкция по установке на mosquito для Debian](#).

2.2 Облачный брокер [cloudMQTT](#).

Регистрация брокера в CloudMQTT описано в разделе [4.3 Регистрация брокера в CloudMQTT](#). Данный брокер позволяет бесплатное использование, но при это ограничивает пропускную способность до 10 кб/с и максимальное количество подключений до 10.

3 Подключение к брокеру.

3.1 Подключение ERD-4 к брокеру

В web интерфейсе переходим во вкладку Администрирование->MQTT. И вводим реквизиты брокера. ERD для подключения не использует шифрование, поэтому нужно ввести порт MQTT сокета, который не использует шифрование (для брокера CloudMQTT всегда начинается с 1). Введенные реквизиты изображены на рисунке 1. При необходимости можно указать корневой топик (Root topic) длиной не более 30 символов.



Рисунок 1 – Ввод реквизитов брокера

3.2 Подключение клиента к брокеру.

Также для подключения можно использовать любое приложение, совместимое с API. Описание API можно посмотреть в разделе [4.5 Описание API MQTT для ERD-4](#). Описание API для Project можно посмотреть в разделе [4.6 Описание API MQTT для Project-2](#). Пример реализации подключения с API в приложении MQTT Dash можно посмотреть в разделе [4.7 Пример реализации API MQTT Dash](#).

4 Инструкции и примеры.

4.1 Инструкция по установке на mosquito для Windows.

Follow the steps as is and hopefully you should be good to go:

1. Download Mosquitto cygwin build from [here](#)
2. Download Windows XP build from [here](#)
3. Extract the Windows XP build into a specific location in to your system.
4. Run the Mosquitto cygwin build (from step1) for installation, it will fail the first time but don't worry, everything is fine, keep following the steps
5. Now go to the extracted version of XP build (from step 2) and find the following files:

cygcrypto-1.0.0.dll
cyggcc_s-1.dll
cygssl-1.0.0.dll
cygwin1.dll
cygz.dll
libeay32.dll
libssl32.dll
msvcr100.dll
pthreadVC2.dll
ssleay32.dll

6. Copy the above files from XP build to the installation directory of Cygwin build which failed in step 4, by default mosquito will install itself in *C:\Program File(x86)\mosquitto*.
7. Now run the cygwin build *exe installer* again. It will run normally this time. click on Finish.
8. Go to services from Start Menu and find Mosquitto Broker. By default the service will not be started. Start the service.

In order to validate, go to command line and type the following command:

netstat -a (command to find out which all ports are running in the system)

If everything is working fine, you should be able to see the following output:

4.2 Инструкция по установке на mosquito для Debian.

```
cd ~
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
cd /etc/apt/sources.list.d/
sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list
sudo apt-get update

cd ~
wget http://security.debian.org/debian-security/pool/updates/main/o/openssl/libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb
sudo dpkg -i libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb
wget http://ftp.nz.debian.org/debian/pool/main/libw/libwebsockets/libwebsockets3_1.2.2-1_armhf.deb
sudo dpkg -i libwebsockets3_1.2.2-1_armhf.deb
sudo apt-get install mosquitto mosquitto-clients
##Изменение конфига:
sudo vi /etc/mosquitto/conf.d/protocols.conf
##содержание конфига:
##-----
listener 1883
protocol mqtt
listener 1884
protocol websockets
##-----
sudo service mosquitto restart ##после изменения конфига, нужен рестарт.
##создание пользователей
##создать пользователя с именем NAME:
sudo mosquitto_passwd -c /etc/mosquitto/passwd NAME
##после создания сразу просит ввести пароль пользователя
##для отключения анонимных подключений и включения авторизации меняем
конфиг, используя команду:
sudo vi /etc/mosquitto/conf.d/default.conf
##содержание конфига:
##-----
allow_anonymous false
password_file /etc/mosquitto/passwd
##-----
sudo service mosquitto restart ##после изменения конфига, нужен рестарт.
```

4.3 Регистрация брокера в CloudMQTT.

Переходим по ссылке <https://www.cloudmqtt.com/>. Нажимаем Log in. В поле, изображенном на рисунке 1 вводим email или регистрируемся с помощью аккаунта гугл.

Рисунок 1 – Форма регистрации

На указанную почту приходит письмо, переходим по ссылке в письме. Заполняем поля изображенные на рисунке 2 и завершаем регистрацию нажимая на кнопку "Submit".

Рисунок 2 – Форма завершения регистрации

Создаем новый экземпляр(брокер), нажимая на кнопку “Create New Instance”. Выбираем регион (Желательно выбрать EU регион, т.к. регион US может не работать на некоторых операторах). Набираем любое имя и нажимаем “Create New Instance”. Выбираем экземпляр нажимая на его имя. Попадаем на страницу с информацией о брокере. Представление страницы приведено на рисунке 3. На странице представлены реквизиты брокера, такие как:

- Доменное имя. поле “Server”;

- Порты для подключения. Поля “Port”, “SSL Port” и “Websocket Port”.

Выбирается в зависимости от сокета клиента.

Рисунок 3 – Основные реквизиты брокера

В навигационном поле, изображенном слева. Во вкладке Connections можно посмотреть текущие подключения к брокеру, во вкладке websocket ui можно увидеть все публикуемые сообщения.

Переходим во вкладку “Users & Acl” для создания пользователя. Создание пользователя необходимо для разделения прав доступа. Поскольку в данном проекте нет необходимости ограничивать доступ, то можно создать общего пользователя для ERD-4 и для смартфона, который будет иметь доступ ко всем публикациям. Вводим в поле изображенное на рисунке 4 логин и пароль. И нажимаем “+ Add”.

Рисунок 4 – Ввод реквизитов пользователя

Далее выдаем необходимые права доступа пользователю. Для этого в поле “ACLs” нажимаем Topic и вводим данные в соответствии с рисунком 5 и нажимаем “+ Add”. На рисунке 5 пользователю дан доступ ко всем публикуемым данным.

Рисунок 5 – Присвоение прав доступа пользователю

Если необходимо ограничить доступ некоторых пользователей к публикуемым данным, то для них можно создать отдельные правила чтения/записи. На рисунке 6 изображен пример, в котором пользователю owner разрешено получать(читать) данные с топика device1 и со всех вложенных в него топиков. И разрешено публиковать(записывать) данные в топик smarfon1 и все вложенные в него топика.

Рисунок 6 – Присвоение прав доступа пользователю

Перейти в пункт «WEBSOCKET UI», если брокер может подключиться сам к себе, вы увидите уведомление «Connected!».

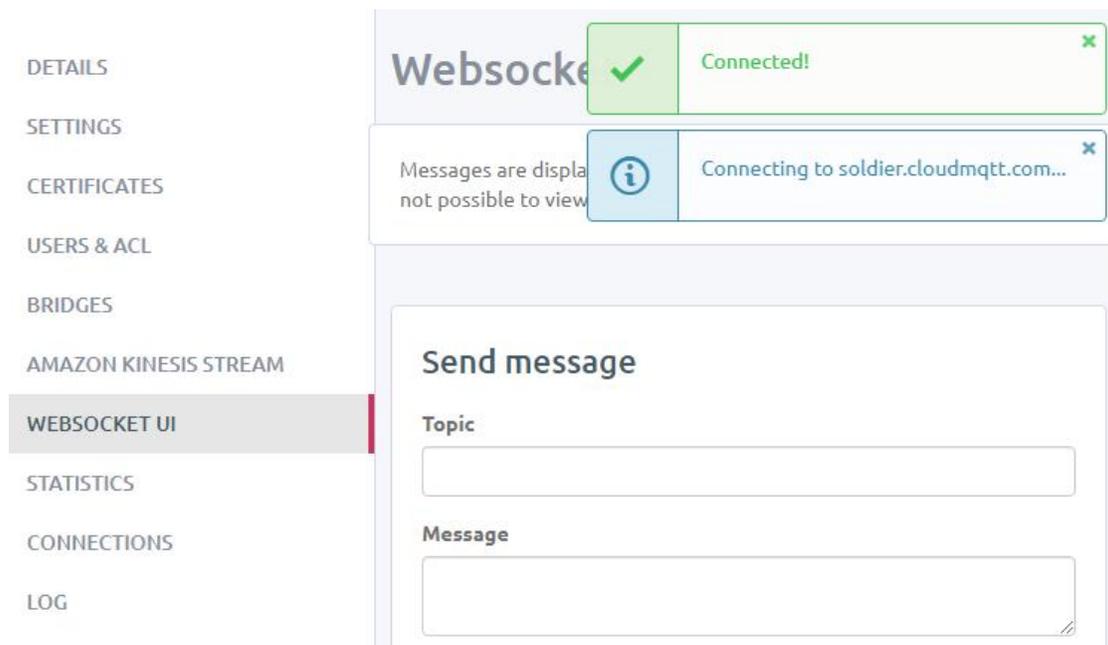


Рисунок 7 – Успешное подключение брокера к самому себе

Если брокер не может подключиться сам к себе увидите уведомление о том что не удалось подключиться, и тогда следует создать брокера выбрав другой регион (US)

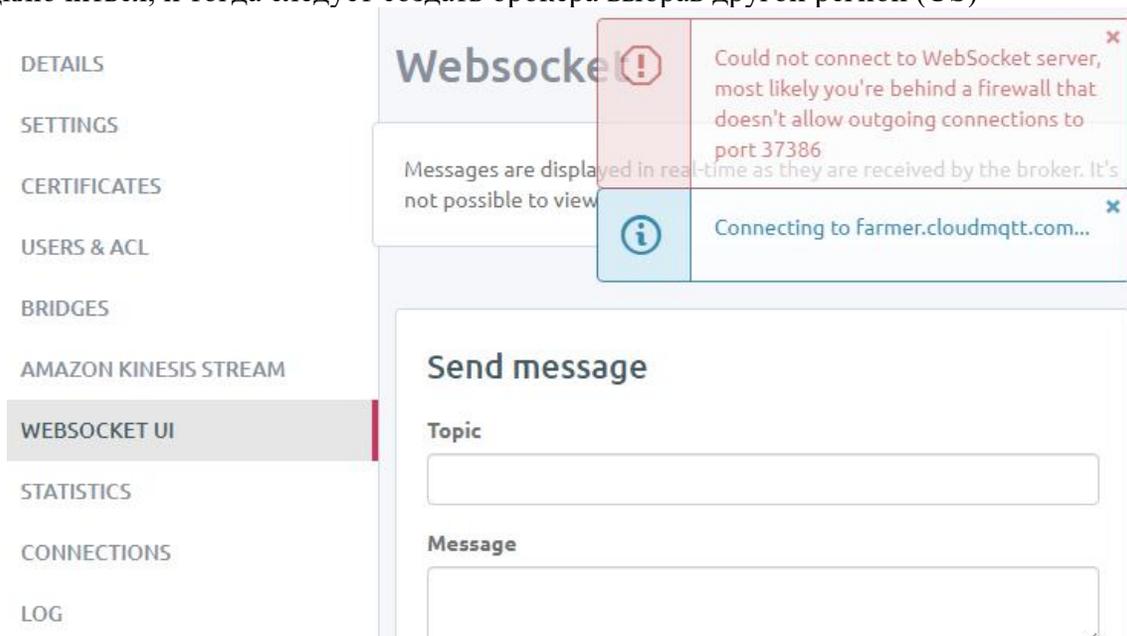


Рисунок 8 – Ошибка при подключении брокера к самому себе

4.5 Описание API MQTT для ERD-4.

UID устройства.

uid(уникальный идентификатор) устройства формируется из трех последних октетов mac-адреса. т.е. при mac-адресе: F8:F0:82:02:00:00. uid будет равен 131072. т.к $020000_{16} = 131072_{10}$

uid устройство можно посмотреть в веб-интерфейсе на вкладке: Администрирование→MQTT.

1. Публикуемые данные.

1.1 Публикация общей информации.

[Root topic]/[device]/[device uid]/info/{sysName | sysLocation | reload_count | uptime | ADCIN | DAC12V}

Соответствие топиков с публикуемыми данными:

- a) Root topic – корневой топик, отсутствует «[Root topic]/» если не был указан при выполнении пункта 3.1
- b) sysName– название устройства;
- c) sysLocation – sysLocation устройства;
- d) reload_count – количество перезагрузок устройства;
- e) uptime – время непрерывной работы устройства;
- f) ADCIN – текущее напряжение или ток на аналоговом входе устройства;
- g) DAC12V– текущее напряжение на ЦАП устройства.

Все данные, кроме uptime, публикуются при включении устройства и при изменении с установленным флагом retain(сохраняет последний статус сообщения на брокере и отправляет клиенту при подключении). Частота публикации Uptime задается в web в поле "Частота публикации данных"

Пример.

Топик: nameRootTopic/erd-4/1118481/info/sysName. Данные: SNR-ERD-4.

1.2 Публикация состояния DIO.

[Root topic]/[device]/[device uid]/DIO/[DIO number{0-6}]/[DIO mode{in| 1(out)}]/{state | count}

Соответствие топиков с публикуемыми данными:

а) state – состояние порта(высокий или низкий уровень напряжения);

б) count – количество изменений состояния порта.

count отправляется, только если порт настроен на вход. DIO с номером 0 – это реле (всегда настроен на выход). DIO с номером 6 – это датчик фазы. Всегда настроен на вход. Все данные публикуются с флагом retain и обновляются по изменению.

Пример.

Топик: nameRootTopic /erd-4/1118481/DIO/0/1/state. Данные: 0.

1.3 Публикация наименований DIO.

1.4

[Root topic]/[device]/[device uid]/DIO/[DIO number{1-5}]/name

Соответствие топиков с публикуемыми данными:

а) name – наименование порта;

Все данные публикуются с флагом retain и обновляются по изменению.

Пример.

Топик: nameRootTopic /erd-4/1118481/DIO/1/name. Данные: ALARM-1.

1.5 Публикация показаний датчиков.

1.6

[Root topic]/ [device]/[device uid]/sensor/[name or index]/[type of data{temperature | humidity | pressure}]

Соответствие топиков с публикуемыми данными:

а) temperature – температура;

б) humidity – влажность;

в) pressure – давление.

Поддерживаемые датчики:

– Dallas;

– DHT11/DHT22;

– SNR-Sensor.

Все данные публикуются в численном виде, без единиц измерения. Частота публикации показаний задается в web.

Пример.

Топик: nameRootTopic/erd-4/1118481/sensor/Dallas/temperature. Данные:30.1.

1.5 Публикация конфигурации термостата.

[Root topic]/[device]/[device uid]/thermostat/[DO number{0-5}]/{normal | limit}

Все данные публикуются с флагом retain и обновляются по изменению.

Пример.

Топик: nameRootTopic/erd-4/1118481/thermostat/0/normal. Данные:38.

1.6 Публикация конфигурации ПИД-регулятора.

[Root topic]/[device]/[device uid]/PID/{low_limit | up_limit | setting | amplitude_filter | proportionality | integration | differentiation | control_cycle}

Соответствие топиков с данными:

- а) low_limit – Нижний предел;
- б) up_limit – Верхний предел;
- в) setting – Уставка;
- г) amplitude_filter – Амплитудный фильтр;
- д) proportionality – Коэффициент пропорциональности;
- е) integration – Период интегрирования;
- ж) differentiation – Период дифференцирования;
- з) control_cycle – Период цикла регулирования.

Все данные публикуются с флагом retain и обновляются по изменению.

Пример.

Топик: nameRootTopic/erd-4/1118481/PID/amplitude_filter. Данные:11.655.

1.7 Публикация критических оповещений.

Все критические состояния датчиков и портов публикуются в топик:

[Root topic]/[device]/[device uid]/alarm

Пример.

Топик: nameRootTopic/erd-4/1118481/alarm. Данные:Dallas: Critical temperature.

2. Обработываемые данные.

2.1 Переключения состояния DO.

[Root topic]/[device]/[device uid]/management/DIO/[DIO number{0-5}]

При получении 1 выставляет высокий уровень напряжения на порте с соответствующим номером.

При получении 0 выставляет низкий уровень напряжения на порте с соответствующим номером.

Пример.

Топик: nameRootTopic/erd-4/1118481/management/DIO/3. Данные:1. Выставляет высокий уровень напряжение на DO 3.

2.2 Напряжение на выходе DAC.

[Root topic]/[device]/[device uid]/management/DAC12V

При получении 0.00 – 10.00 выставляет на выходе DAC соответствующий уровень напряжения. При получении 12, переходит в режим 12В и на выходе выставляется соответствующий уровень напряжения.

Пример.

Топик: nameRootTopic/erd-4/1118481/management/DAC12V. Данные:1.22. На выход подается напряжение 1.22В.

2.3 Конфигурация термостата.

[Root topic]/[device]/[device uid]/management/thermostat/[DO number{0-5}]/{normal | limit}

Позволяет конфигурировать нормальную и предельную температуру для каждого выхода в режиме термостат.

Пример.

Топик: nameRootTopic/erd-4/1118481/management/thermostat/0/normal. Данные:38. Устанавливает значение нормальной температуры режима термостат для выхода Реле.

2.4 Конфигурация ПИД-регулятора.

[Root topic]/[device]/[device uid]/management/PID/{low_limit | up_limit | setting | amplitude_filter | proportionality | integration | differentiation | control_cycle}

Позволяет конфигурировать ПИД-регулятор.

Соответствие топиков с данными:

- а) low_limit – Нижний предел;
- б) up_limit – Верхний предел;
- в) setting – Уставка;
- г) amplitude_filter – Амплитудный фильтр;
- д) proportionality – Коэффициент пропорциональности;
- е) integration – Период интегрирования;
- ж) differentiation – Период дифференцирования;
- з) control_cycle – Период цикла регулирования.

Пример.

Топик: nameRootTopic/erd-4/1118481/management/PID/low_limit. Данные:60.51.

Устанавливает значение нижнего предела для ПИД-регулятора.

2.5 Перезагрузка устройства.

[Root topic]/[device]/[device uid]/management/reload

При получении 1 перезагружает устройство.

Пример.

Топик: nameRootTopic/erd-4/1118481/management/reload. Данные:1.

Перезагружает устройство.

4.6 Описание API MQTT для Project-2

UID устройства.

uid(уникальный идентификатор) устройства формируется из трех последних байт mac-адреса. т.е. при mac-адресе: DD:DD:DD:11:11:11. uid будет равен 1118481. т.к 11111116 =111848110

uid устройство можно посмотреть в веб-интерфейсе на вкладке Администрирование→MQTT.

1. Публикуемые данные.

1.1 Публикация общей информации.

```
[Root topic]/[device]/[device uid]/info/{sysName | sysLocation | reload_count | uptime | ADC1 | ADC2 | AC}
```

Соответствие топиков с публикуемыми данными:

- а) sysName– название устройства;
- б) sysLocation – sysLocation устройства;
- в) reload_count – количество перезагрузок устройства;
- г) uptime – время непрерывной работы устройства;
- д) ADC1– текущее напряжение на первом аналоговом входе устройства
- е) ADC2– текущее напряжение на втором аналоговом входе устройства;
- ж) AC - текущее напряжение на датчике фазы.

Все данные, кроме uptime, публикуются при включении устройства и при изменении с установленным флагом retain(сохраняет последний статус сообщения на брокере и отправляет клиенту при подключении). Частота публикации Uptime задается в web в поле "Частота публикации данных"

Пример.

Топик: nameRootTopic/project-2/1118481/info/sysName. Данные: SNR-ERD-Project-2.

1.2 Публикация состояния DIO.

```
[Root topic]/[device]/[device uid]/DIO/[DIO number{0-6}]/[DIO mode{0(in) | 1(out)}]/{state | name | count}
```

Соответствие топиков с публикуемыми данными:

- а) state – состояние порта(высокий или низкий уровень напряжения);
- б) name – наименование порта;
- в) count – количество изменений состояния порта.

count отправляется, только если порт настроен на вход. DIO с номером 7 – это датчик фазы. Всегда настроен на вход. Все данные публикуются с флагом retain и обновляются по изменению.

Пример.

Топик: nameRootTopic/project-2/1118481/DIO/0/1/state. Данные: 0.

1.3 Публикация показаний датчиков.

[Root topic]/[device]/[device uid]/sensor/[name or index]/[type of data{temperature | humidity | pressure}]

Соответствие топиков с публикуемыми данными:

- а) temperature – температура;
- б) humidity – влажность;
- в) pressure – давление.

Поддерживаемые датчики:

- Dallas;
- SNR-Sensor.

Все данные публикуются в численном виде, без единиц измерения. Частота публикации показаний задается в web.

Пример.

Топик: nameRootTopic/project-2/1118481/sensor/Dallas/temperature. Данные:30.1.

1.4 Публикация конфигурации термостата.

[Root topic]/[device]/[device uid]/thermostat/[DO number{0-5}]/{normal | limit}

Все данные публикуются с флагом retain и обновляются по изменению.

Пример.

Топик: nameRootTopic/project-2/1118481/thermostat/0/normal. Данные:38.

1.5 Публикация критических оповещений.

Все критические состояния датчиков и портов публикуются в топик:
[Root topic]/[device]/[device uid]/alarm

Пример.

Топик: nameRootTopic/project-2/1118481/alarm. Данные:Dallas: Critical temperature.

2. Обрабатываемые данные.

2.1 Переключения состояния DO.

[Root topic]/[device]/[device uid]/management/DIO/[DIO number{0-5}]

При получении 1 выставляет высокий уровень напряжения на порте с соответствующим номером.

При получении 0 выставляет низкий уровень напряжения на порте с соответствующим номером.

Пример.

Топик: nameRootTopic/project-2/1118481/management/DIO/3. Данные:1.
Выставляет высокий уровень напряжение на DO 3.

2.2 Конфигурация термостата.

[Root topic]/[device]/[device uid]/management/thermostat/[DO number{0-5}]/{normal | limit}

Позволяет конфигурировать нормальную и предельную температуру для каждого выхода в режиме термостат.

Пример.

Топик: nameRootTopic/project-2/1118481/management/thermostat/1/normal.

Данные:38.

Устанавливает значение нормальной температуры режима термостат для выхода DO1.

2.3 Перезагрузка устройства.

[Root topic]/[device]/[device uid]/management/reload

При получении 1 перезагружает устройство.

Пример.

Топик: nameRootTopic/project-2/1118481/management/reload. Данные:1.

Перезагружает устройство.

4.7 Пример реализации API. MQTT Dash.

1. Установка соединения с брокером

После установки приложения. Нажимаем на значок “+” в верхнем правом углу для создания нового подключения. Вводим реквизиты брокера как показано на рисунке 1. Остальные параметры выставляем по своему усмотрению.

MQTT Dash

По умолчанию (подключаться автоматически при запуске).
Примечание: эта опция полезна, если у Вас сконфигурировано только одно соединение.

Если соединений больше, чем одно - Вы можете создать ярлык для любого соединения на домашнем экране.
Для создания ярлыка примените "длинное" нажатие на любом соединении в списке соединений.

Экран всегда включен, когда подключение к этому брокеру активно

Разрешить изменение метрик. Если отключено, то добавление, изменение и удаление существующих метрик будет отключено. Это служит защитой от непреднамеренного изменения.

Имя
nag

Адрес
soldier.cloudmqtt.com

Порт
15404

Включить шифрование соединения (SSL/TLS).
Внимание: если сертификат сервера само-подписан (self-signed), необходимо установить его на ваше устройство или включить опцию ниже, иначе соединение будет невозможно. Если сертификат сервера выпущен любым известным Центром Сертификации (CA), то он будет работать как есть, без предварительной установки на устройство. Так-же не забывайте, что для зашифрованных и обычных соединений MQTT сервера используют различные порты.

Этот брокер использует само-подписанный (self-signed) SSL/TLS сертификат. Я доверяю этому сертификату на мой страх и риск.

Имя пользователя
jdrvufdq

Пароль пользователя
.....

Идентификатор клиента (должен быть уникальным)
mqttdash-a274108f

Размер плиток
 Маленький
 Средний

Рисунок 9 – Ввод реквизитов подключения

2. Конфигурация приложения

Нажимаем на значок "+" в верхнем правом углу для создания новой метрики. Далее в соответствии с описанием API. Создаем метрику с мониторингом или публикацией необходимой нам информации. Примеры метрик и итоговая конфигурация приведена на рисунках 2-9.

MQTT Dash 📄

Эта метрика предназначена для простого отображения данных принятого пакета (например, температуры, статуса или любого другого произвольного текста). Данные принятого пакета должны быть строкой.

Имя

sysName-3

Топик (sub)

3/erd-4/131072/info/sysName

Извлечь, используя JSON path (если полученные данные в формате JSON), например: \$.level.value. JSON path документация по ссылке ниже:
<https://github.com/jayway/JsonPath/blob/master/README.md>

Включить публикацию

Префикс Постфикс

Размер главного текста

Маленький

Средний

Крупный

Цвет текста

Другие настройки

QoS(0)

QoS(1)

QoS(2)

Плитка мигает для привлечения внимания, если выражение (expression) истинно (true).
 Выражение должно быть корректным JavaScript boolean выражением, которое возвращает true или false.
 В выражении Вы можете использовать константы 'val' и 'secs'.
 'val' содержит принятое сообщение (payload), либо значение, извлеченное из JSON, если Вы используете JSON path.

Рисунок 10 – Пример конфигурации метрики текст, отображающей sysName устройства имеющего “Root topic” = ”3”

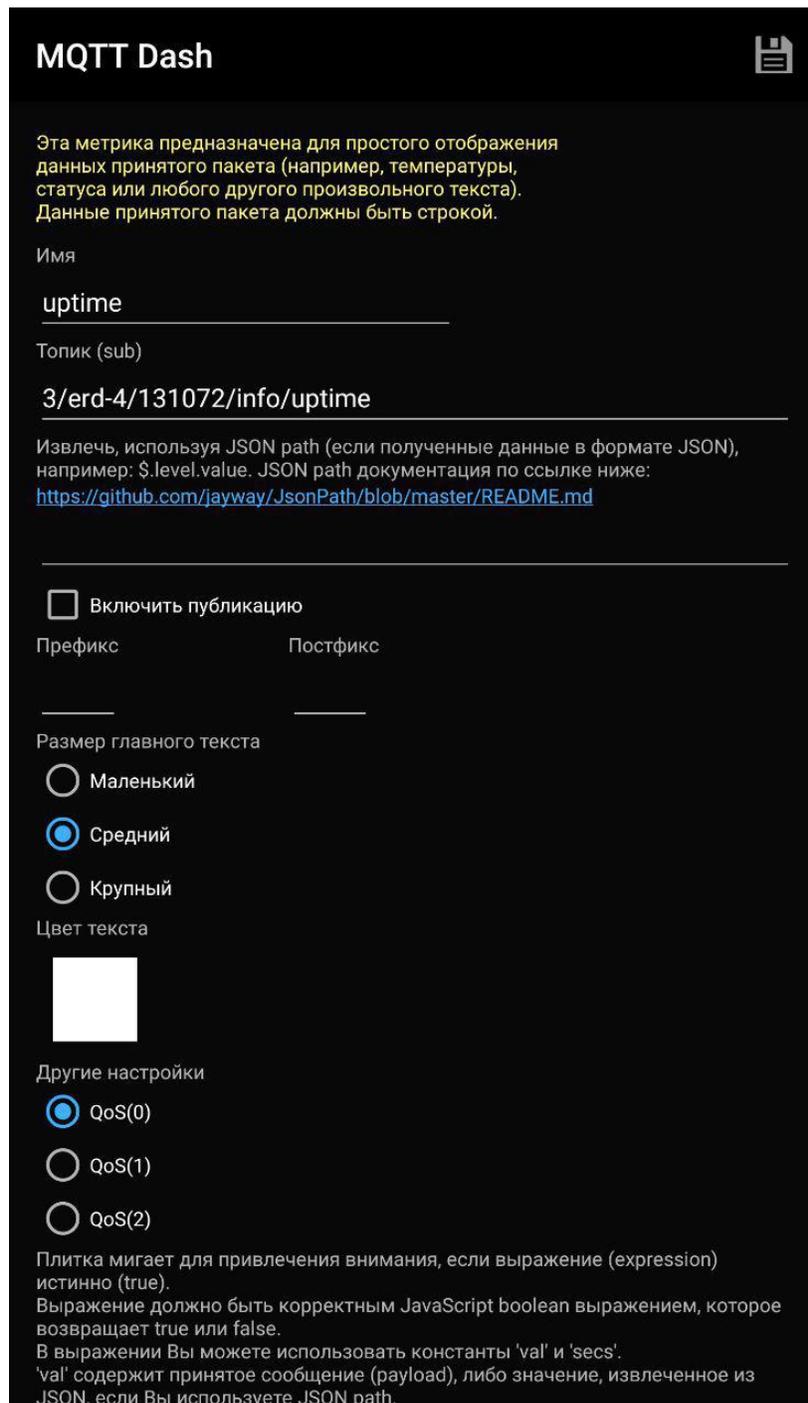


Рисунок 11 – Пример конфигурации метрики текст, отображающей uptime устройства имеющего “Root topic” = “3”

MQTT Dash



Эта метрика предназначена для простого отображения данных принятого пакета (например, температуры, статуса или любого другого произвольного текста). Данные принятого пакета должны быть строкой.

Имя

critical

Топик (sub)

3/erd-4/131072/alarm

Извлечь, используя JSON path (если полученные данные в формате JSON), например: \$.level.value. JSON path документация по ссылке ниже:

<https://github.com/jayway/JsonPath/blob/master/README.md>

Включить публикацию

Префикс

Постфикс

Размер главного текста

Маленький

Средний

Крупный

Цвет текста



Другие настройки

QoS(0)

QoS(1)

QoS(2)

Плитка мигает для привлечения внимания, если выражение (expression) истинно (true).

Выражение должно быть корректным JavaScript boolean выражением, которое возвращает true или false.

В выражении Вы можете использовать константы 'val' и 'secs'.

'val' содержит принятое сообщение (payload), либо значение, извлеченное из JSON, если Вы используете JSON path.

Рисунок 12 – Пример конфигурации метрики текст, отображающей последнее критическое оповещение с устройства имеющего “Root topic” = ”3”

Рисунок 13 – Пример конфигурации метрики прогресс, отображающей температуру с датчика с наименованием SNR-Sensor

Рисунок 14 – Пример конфигурации метрики переключатель, позволяющей отображать и изменять состояние реле устройства

Рисунок 15 – Пример конфигурации метрики диапазон, позволяющей отображать и изменять состояние выхода DAC/12V устройства

Рисунок 16 – Пример итоговой конфигурации

Рисунок 17 – Пример итоговой конфигурации

3. Импорт/Экспорт конфигурации

Для экспорта конфигурации клиенты должны быть подключены к одному брокеру. Нажимаем на значок со стрелками, указывающими в разные стороны в правом верхнем углу. На устройстве, где нужно получить конфигурацию нажимаем на кнопку “подписать и ждать метрики”. И публикуем конфигурацию на другом устройстве с нужной конфигурацией, нажимая кнопку “опубликовать метрики”. Для сохранения конфигурации метрик на брокере и отправления каждому клиенту при подключении нужно нажать кнопку “опубликовать метрики(retained)”.