

- Курсовая (дипломная) работа
 - Введение
 - 1. BME680
 - Программное обеспечение BSEC
 - BME68x-Sensor-API
 - Odroid-X2
 - ArchLinux
 - Клиент mqtt
 - Sense
 - Web-server
 - Cronie
 - IBM PC
 - Mosquitto
 - Nod-Red
 - Telegraf
 - InfluxDB
 - Grafana
 - Доступ в интернет
 - Список литературы:

Курсовая (дипломная) работа

Предмет: Программирование на языке C (IOT)

Тема: Электронно-измерительный комплекс фиксации, хранения и визуализации параметров окружающей среды

Ростислав Ромашин

Группа: 5471

Весь проект на гитхабе: <https://github.com/allseenn/ciot/tree/main/04.Tasks>

Составные части проекта:

- [sense](#) программа считывания показаний на СИ
- [python](#) программа считывания показаний на пайтоне
- [mqtt](#)-клиент для отправки показаний на сервер
- [web-server](#) веб-сервер мини-пк

Введение

В соответствии с [требованиями курсового проекта](#) был разработан и реализован комплекс по мониторингу окружающей среды, который состоит из следующих компонентов:

1. BME680 - Датчик газа, давления, температуры и влажности с низким энергопотреблением
 - BSEC - проприетарное ПО
 - BME68x-Sensor-API - открытое ПО
2. Odroid-X2 - Клиентский контроллер на базе CPU ARMv7
 - ArchLinux
 - Клиент mqtt на языке C
 - Sense - приложение сбора метрик с датчика BME680 на языке C с помощью открытых API
 - Web-сервер на языке C
 - cronie - crontab для archlinux
3. IBM PC - Серверная часть на базе CPU x86
 - Node-Red
 - Mosquitto
 - Telegraf
 - InfluxDB
 - Grafana
4. ЛВС - с возможностью организации интернет доступа ко всем вышеперечисленным компонентам
 - OpenWRT router + ddns
 - Nginx

1. BME680

Датчик газа, давления, температуры и влажности с низким энергопотреблением от компании BOSCH. BME680 — это цифровой 4-в-1 датчик, измеряющий загазованность, влажность, давление и температуру на основе проприетарных принципов измерения. Модуль датчика помещен в крайне компактный корпус с металлической крышкой LGA, имеющий размеры всего $3,0 \times 3,0 \text{ мм}^2$ и максимальную высоту 1,00 мм ($0,93 \pm 0,07 \text{ мм}$). Его небольшие размеры и низкое

энергопотребление позволяют интегрировать его в устройства, работающие от батареи или в устройства с частотной связью, такие как мобильные телефоны или носимые устройства.

Данный датчик 4-в-1 куплен на [алиэкспресс](#) за 353 рубля. Что, примерно в десять раз дороже [классических](#) "одномерных" датчиков. В чем же преимущества и недостатки?

Программное обеспечение BSEC

Датчик BME680 предназначен для использования вместе с решением Bosch Software Environmental Cluster (BSEC) и API датчика BME6хх, чтобы раскрыть его полный потенциал. Программное обеспечение BSEC включает интеллектуальные алгоритмы, которые позволяют использовать такие сценарии, как мониторинг качества воздуха по различным индексам.

Датчик оснащен специальной пластиной, нагревающейся до 320 и более градусов по Цельсию. Данная пластина замеряет сопротивление воздуха. В интернете можно встретить много отрицательных отзывов про точность данных датчиков. Но, просмотрев даташиты производителя понимаешь неоднозначность плохих выводов. С одной стороны допустимая погрешность измеряемой температуры ± 1 градус по Цельсию не позволит применять его в больницах. Но, прибор не позиционируется как высокоточный и медицинский.

Некоторые этот датчик "по-старинке" подсоединяют к системам наподобие arduino или esp, и наблюдают большие разночтения с рядом стоящими бытовыми приборами. Например, температура воздуха вопреки погрешности может быть больше чем на 5 градусов от бытового термометра. Дело тут не в оригинальности продукта, хоть и существует большая доля вероятности купить брак. Дело именно в расчетах. Датчик по своей сути является микроконтроллером со своими регистрами и своей проприетарной логикой работы. Для расчета показателей мощностей ардуино не достаточно. Можно запустить датчик в форсированном режиме и он будет сбрасывать постоянно неточную информацию. Весь смысл заключается именно в закрытом проприетарном ПО. Именно оно участвует в сложнейших расчетах и управлении режимов датчика. Во первых, для того чтобы снять показания сопротивления воздуха, необходимо нагреть пластину до 320 градусов. Этот процесс кратковременный, но все же нагрев отражается на термодатчике. Следовательно нужно замеры температуры делать в промежутках

между нагревами. При расчете температуры, проприетарным ПО учитываются остальные параметры, такие как влажность и давление. О сложности ПО говорит и тот факт, что драйвера для датчика BME680 и BME688 используются одни и те же. А вот датчик BME688 позиционируется уже как AI (ИИ).

Программное обеспечение BSEC от Bosch Sensortec доступно в виде закрытого двоичного файла, который будет предоставлен через Лицензионное Соглашение на Программное Обеспечение (SLA) на сайте Bosch Sensortec (https://www.bosch-sensortec.com/bst/products/all_products/BSEC). Забегая вперед замечу, что пока мне с проприетарным ПО разобраться не удалось, из-за его сложности.

Ключевые особенности системы аппаратного и программного обеспечения:

Расчет температуры окружающего воздуха вне устройства (например, телефона)
Расчет относительной влажности окружающего воздуха вне устройства
Расчет индекса качества воздуха (IAQ) вне устройства
Кроме того, программные алгоритмы обрабатывают компенсацию влажности, базовую линию, а также коррекцию долгосрочного дрейфа сигнала газового датчика.

BME68x-Sensor-API

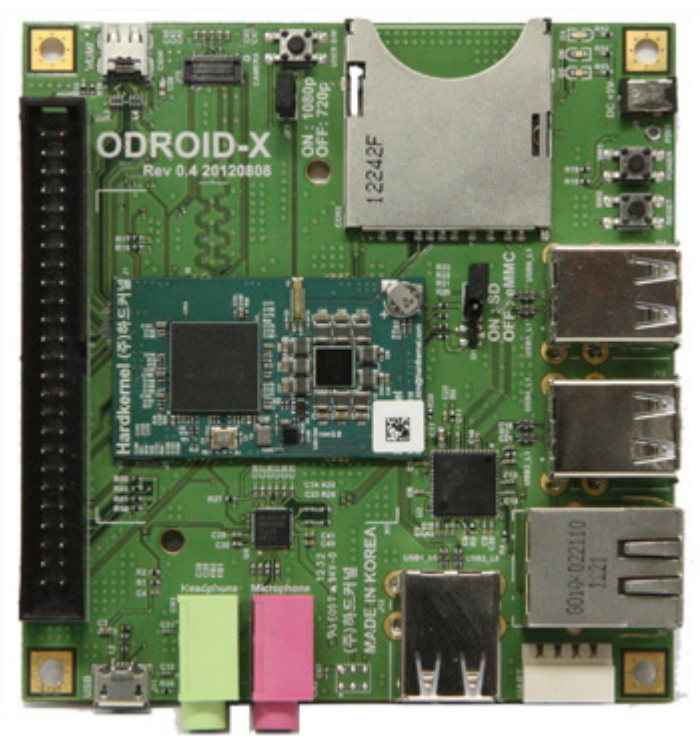
API датчика охватывает базовую коммуникацию с датчиком и функции компенсации данных, и доступен как открытый исходный код на Github (<https://github.com/BoschSensortec/BME68x-Sensor-API>). Т.е. функционал открытого ПО ограничен. Бош бережно хранит алгоритм расчетов в секрете. Более того, они удалили свой старый открытый репозиторий, который содержал более "простой" для понимания драйвер. В новом репозитории, упомянутом выше, лежит другой драйвер, который требует наличия "тяжелого" COINES SDK.

Старый репозиторий с открытым драйвером датчика BME680 до сих пор красуется в последнем даташите. Но, нашел старый открытый драйвер 3.5.9 на стороннем репозитории https://github.com/wintersteiger/BME680_driver. Драйвер содержит открытие API. Для того чтобы начать получать или записывать информацию на датчик, необходимо самостоятельно написать основную программу для конкретной платформы и операционной системы, которая с помощью апи будет обращаться через драйвер к датчику. Т.к. готовых примеров не нашел, то за основу взял решение для Raspberry Pi, но для более старого датчика BME280 <https://www.waveshare.com/w/upload/b/ba/BME280-Environmental-Sensor-Demo-Code.7z>. Код от BME280 частично совместим и позволяет

отображать только температуру. "Подсмотрев" основную логику организации взаимодействия, удалось заставить шайтан-машину работать).

Odroid-X2

Данный малино-подобный мини-ПК использовал в своем прошлом ДЗ для [веб-сервера](#) и [чатов](#).



Распиновка в соответствии с картинкой выше (внизу правее на гребенке идет первый пин)

GND	50	49	ADC_AIN3
VDD_IO	48	47	ADC_AIN2
SYS	46	45	VD16
VD13	44	43	VD22
VD4	42	41	VD1
VD23	40	39	VD9
VD17	38	37	VD14
VD10	36	35	VD5
VD12	34	33	VD3
GND	32	31	HSYNC
VD20	30	29	VDEN
VD6	28	27	VSYNC
VD11	26	25	VD18
VD7	24	23	VCLK
VD0	22	21	VD15
VD8	20	19	VD2
VD21	18	17	VD19
TXD	16	15	SPI_1_MISO

RXD	14 13	PWM_BRT
VDDQ_LCD	12 11	SPI_1_MOSI
SPI_1_CSN	10 09	XE_INT12
T_SDA	08 07	T_SCL
SPI_1_CLK	06 05	SCL
T_RST	04 03	SDA
BL_EN	02 01	T_INT

Схема подключения датчика BME680 к I2C Odroid-X2

```

BME680  Odroid-X2
+-----+-----+
GND      GND (50 pin)
VDD      VDD (48 pin) 1.8V
SDA      SDA (03 pin)
SCL      SCL (05 pin)

```

Датчик подключается без подтягивающих резисторов.

Основной проблемой [Odroid-X2](#) является его "древность". Снят с поддержки лет 10 назад. Но, по мощности не уступает многим Raspberry даже сегодня.

Официальной последней версией была Ubuntu 14.04. Умельцы сделали неофициальный образ на базе Ubuntu 16.04. Для языка СИ это не проблема, но хотелось, что-то посвежее.

ArchLinux

И чисто случайно обнаружил [образ](#) для odroid в официальном хранилище ArchLinux. Данная система позиционируется как более профессиональная версия Linux и одной из особенностей являются Rolling обновления. Т.е. как таковых версий у нее нету, периодически накатываются самые последние обновления и все. Таким образом получаем самое свежее ядро, библиотеки и ПО:

```

$ uname -a
Linux odroid 6.9.6-3-ARCH #1 SMP PREEMPT Thu Jun 27 07:33:29 MDT 2024 armv7l
GNU/Linux

```

```

$ cat /etc/os-release
NAME="Arch Linux ARM"
PRETTY_NAME="Arch Linux ARM"
ID=archarm
ID_LIKE=arch
BUILD_ID=rolling
ANSI_COLOR="38;2;23;147;209"

```

```
HOME_URL="https://archlinuxarm.org/"
DOCUMENTATION_URL="https://archlinuxarm.org/wiki"
SUPPORT_URL="https://archlinuxarm.org/forum"
BUG_REPORT_URL="https://github.com/archlinuxarm/PKGBUILDS/issues"
LOGO=archlinux-logo
```

Клиент mqtt

Основной проблемой является, то что пакета с библиотекой `raho-mqtt-c` нет в офрепозиториях ArchLinux. Пришлось собирать ее самостоятельно из исходников скачанных с [официального репозитория](#).

```
make
make install
```

Написал программу для [mqtt-клиента](#). Програма позволяет с помощью ключей командной строки задать следующие параметры:

- `-i` - ip-адрес и порт mqtt-сервера
- `-u` - имя пользователя
- `-p` - пароль

Программа позволяет также принимать информацию со стандартного потока ввода, до 4-х аргументов в следующем порядке:

- температура
- давление
- влажность
- газ

Sense

Используя полученный и ранее описанный, опыт инжиниринга драйвера от BME280, написал программу [sense](#) для получения информации с датчиков.

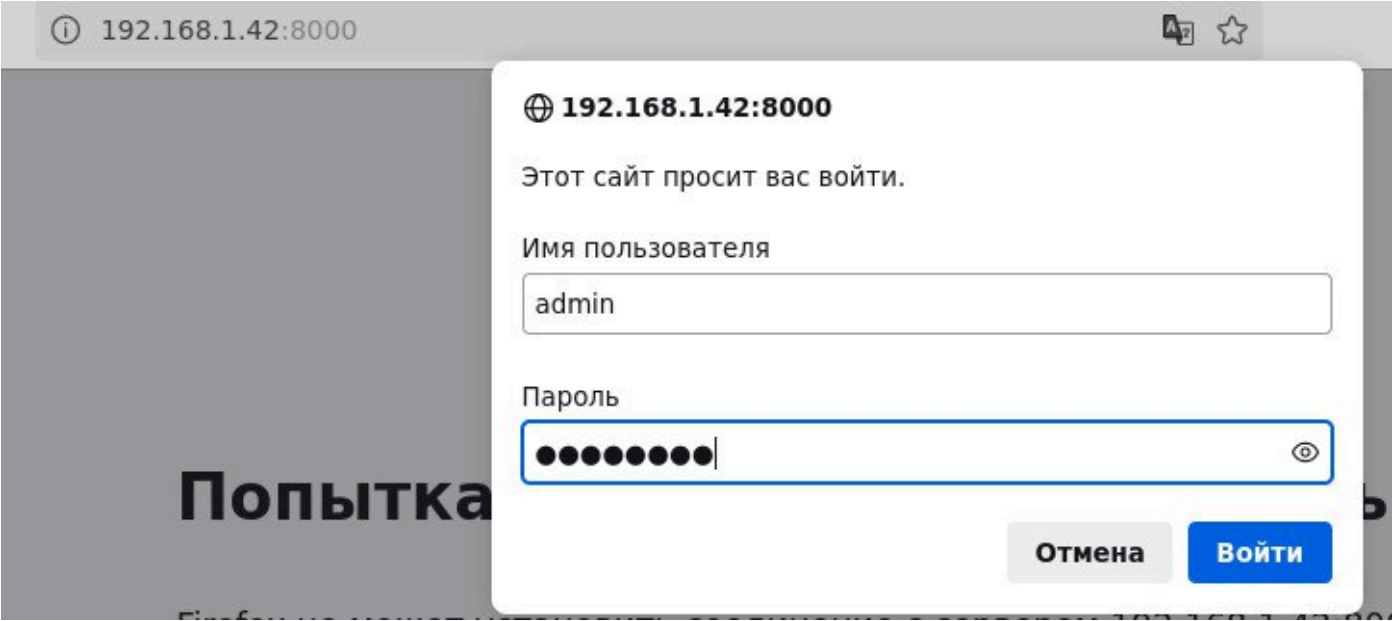
Учел, на будущее, особенности интеллектуальной работы датчика, и зашил в программу возможность получать информацию только об одном или нескольких конкретных показателей с помощью следующих ключей командной строки:

- -t - температура
- -p - давление
- -m - влажность
- -g - газ
- -h - справка

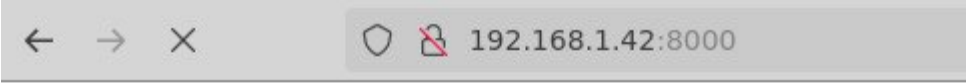
Web-server

Веб сервер был основан на [раней версии](#)

Добавлена авторизация с помощью логина и пароля.



Веб-сервер использует консольную программу sense, описанную ранее и ее значения после парсинга заносятся в соответствующие массивы. Далее информация в виде html-страницы отправляется авторизованному пользователю в виде фронтэнд-интерфейса.



ODROID: WEB-MET

Температура	Давление	Влажность	Воздух
31.280001	739.585388	46.202000	555.460999
Цельсия	mm/PтСт	проценты	Ом

Интерфейс простой, он выводит в таблице названия измеряемых величин, их значения и размерности в единицах измерения СИ.

Cronie

Cronie это стандартная служба планировщик для системы archlinux. Благодаря тому что ранее написанные программы адаптированы для работы в консоли можно организовать пайплан по получению показаний датчика и отправки его на mqtt-сервер по расписанию: один раз в минуту:

```
$ crontab -l
# send temerature, pressure, humidity and gas to mqtt broker
* * * * * /usr/local/sbin/sense -t -p -m -g | /usr/local/sbin/mqtt -i
192.168.1.8:1883 -u admin -p students
```

IBM PC

В качестве сервера используется IBM PC совместимый компьютер с установленной на него Ubuntu 24.04.

На сервере установлен docker и docker-compose. С помощью [bash-скрипта](#), созданного на прошлом курсе по распределенным сетям, генерируется docker-compose файл, устанавливающий следующие системы:

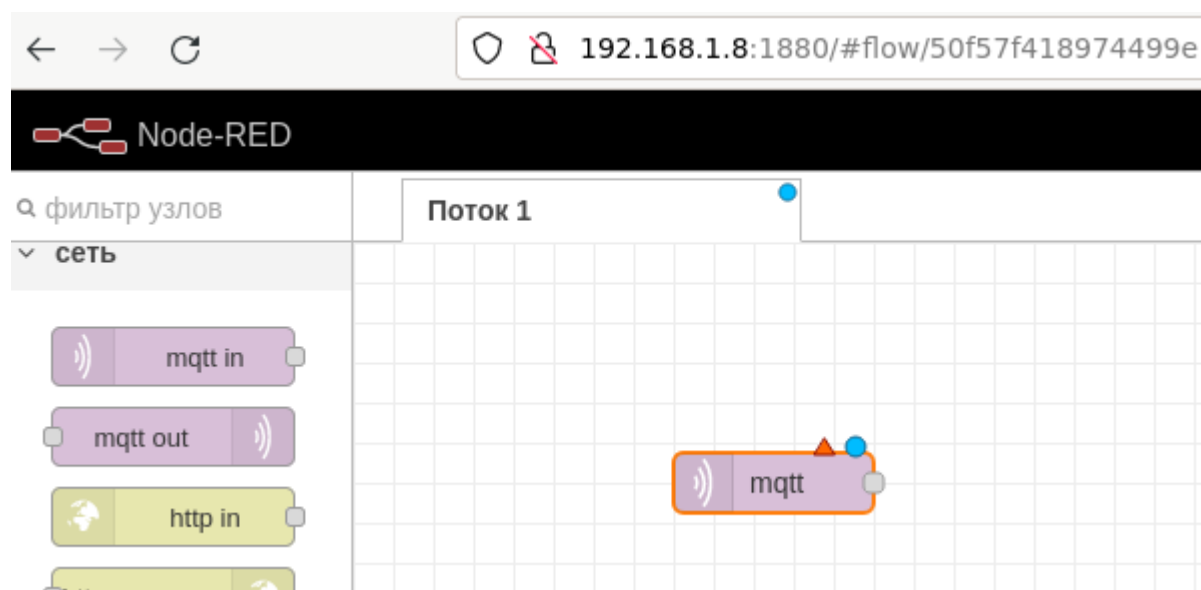
- Mosquitto
- Nod-Red
- Telegraf
- Influx-DB
- Grafana

Mosquitto

Mosquitto является сервером mqtt, он не нуждается в особой настройке, кроме как задании логина и пароля. Я же его еще настроил для работы с веб-сокетами. Это нужно для организации работы через сеть интернет, посредством веб-сервера nginx. Роль mqtt-сервера принимать сообщения от паблешеров и отдавать их

подписчикам. Поэтому большинство программ настраивается на mosquitto, а сам он не настраивается на них

Nod-Red



В качестве и подписчика и паблишера может выступать система node-red.

Добавим несколько подписчиков в его дашборд:

- temperature
- pressure
- humidity
- gas

Изменить узел mqtt in > **Добавить новый конфиг узел mqtt-broker**

ОтменаДобавить

⚙️ Свойства

⚙️📄

🔑 Имя

Имя

Соединение

Безопасность

Сообщения

🌐 Сервер

192.168.1.8

Порт

1883

☒ Connect automatically

☐ TLS

⚙️ Protocol

MQTT V5

🔑 ID клиента

Оставьте пустым для автоматически сгенерированного

💓 Keep-alive время (сек)

60

📄 Session

☒ Use clean start

Session Expiry (secs)

User Properties

▼ none

Для подключения к брокеру необходимо задать параметры такие как логин и пароль

Изменить узел mqtt in > **Добавить новый конфиг узел mqtt-broker**

ОтменаДобавить

⚙️ Свойства

⚙️📄

🔑 Имя

Имя

Соединение

Безопасность

Сообщения

👤 Имя польз.

admin

🔒 Пароль

••••••••

Также необходима задать топик, т.е. тему с которой будет связан информация

Изменить узел mqtt in

Удалить

Отмена

Готово

⚙ Свойства

⚙ 📄 🖨

🌐 Сервер

192.168.1.8:1883

✎

Action

Subscribe to single topic

▼

📋 Тема

temperature

⚙ QoS

2

▼

🚩 Flags

☐ Do not receive messages published by this client

☒ Keep retain flag of original publish

🔍 Retained message handling

Send retained messages

▼

➡ Выход

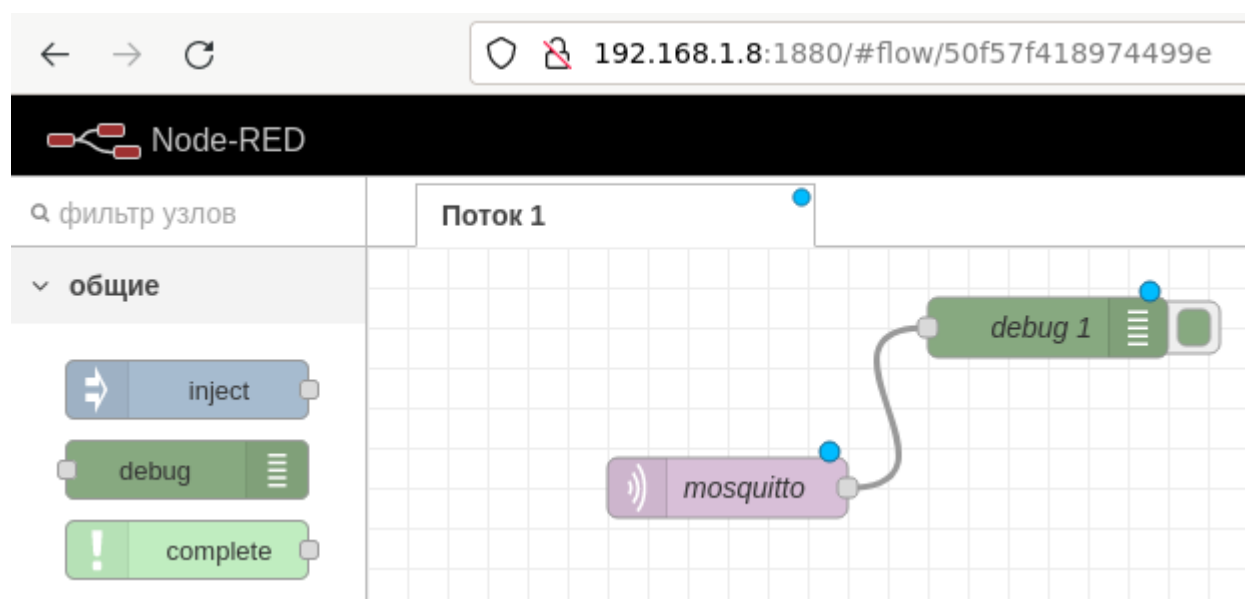
автоопределение (разобрать объект JSON, строка или

▼

🏷 Имя

mosquitto

К каждому подписчику подключить отладчик. Таким образом мы сможем видеть что посылается на наш брокер mqtt по конкретной теме



Для вывода информации необходимо выбрать маленького паучка справка на верху. Это часто забывается)

Изменить узел debug

Удалить

Отмена

Готово

⚙ Свойства

⚙

📄

🖨

☰ Выводить

▼ msg. payload

⚙ В

☒ окно отладки

☒ системную консоль

☐ статус узла (32 символа)

🏷 Имя

debug temperature

Как только нажали на паука, поступающая информация начнет поступать в окно. Помимо самих данных у нас отображается дата и время получения сообщения. По этой причине, отправка времени с программы sense (считывание датчика) является излишней. Т.к. далее в базе данных временных рядов, дата и время являются основополагающими для InfluxDB и нет необходимости задавать отдельное поле времени для этого.

🕸 отладка

📄

🖨

⚙

▼

🔍 все узлы ▼

🗑 all ▼

05.07.2024, 07:20:30 node: debug temperature

temperature : msg.payload : number

30

05.07.2024, 07:24:03 node: debug temperature

temperature : msg.payload : number

31

Telegraf

Если node-red нам нужен только для отладки. И в дальнейшей схеме может быть исключен. То telegraf является связующим звеном между брокером и базой данных. Он подписывается на все топики брокера используя заданные в настройках авторизационные данные и передает их дальше, в нашем примере InfluxDB

Пример конфига telegraf:

```
[agent]
  interval = "3s"
  round_interval = true
  metric_batch_size = 1000
  metric_buffer_limit = 10000
  collection_jitter = "0s"
  flush_interval = "3s"
  flush_jitter = "0s"
  precision = ""
  hostname = ""
  omit_hostname = false

[[outputs.influxdb_v2]]
  urls = ["http://$LOC_IP:8086"]
  token = "$INFLUXDB_TOKEN"
  organization = "$ORG"
  bucket = "$BUCKET"

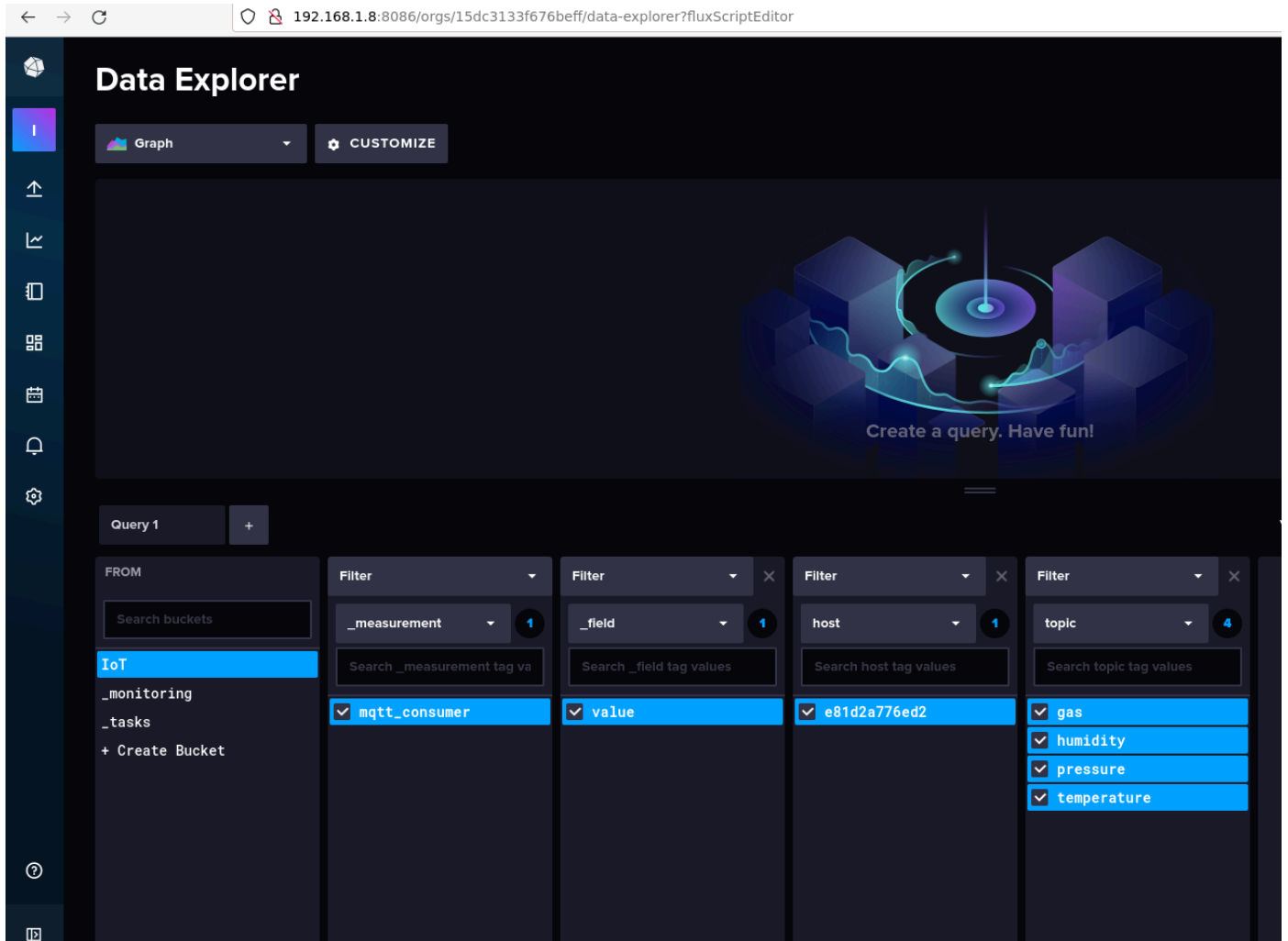
[[inputs.mqtt_consumer]]
  servers = ["tcp://$LOC_IP:1883"]
  topics = ["#"]
  username = "$USERNAME"
  password = "$PASSWORD"
  data_format = "value"
  data_type = "float"

[[inputs.docker]]
  endpoint = "unix:///var/run/docker.sock"
```

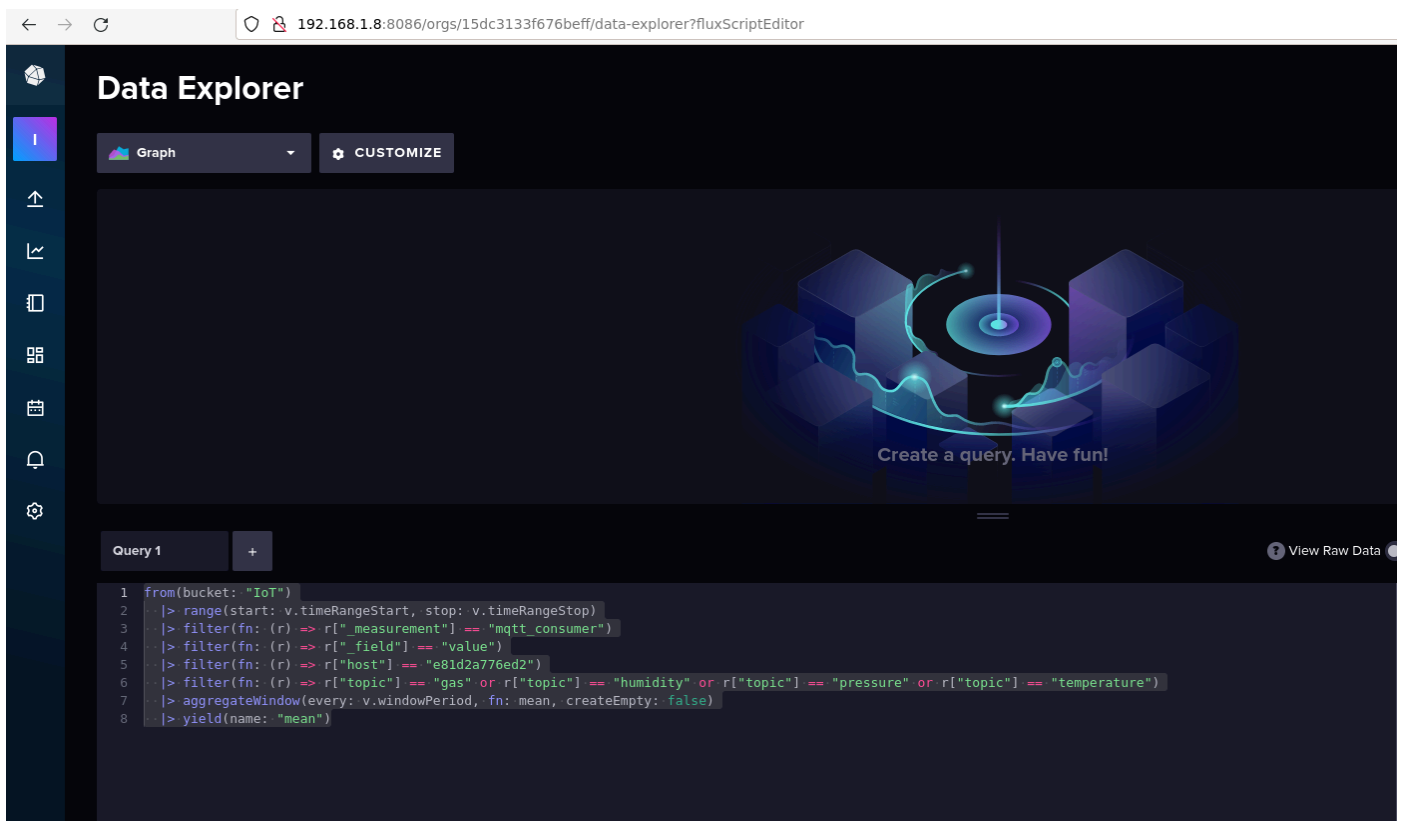
InfluxDB

Ключевым игроком в нашей связке выступает СУБД Influx-DB, это не просто система управления базой данных, но и визуализация, которая может заменить в простых случаях Grafana.

Для начала в веб-интерфейсе InfluxDB необходимо выбрать бакет (корзину) в которую скидывает информация с помощью Telegraf. В данном бакете можно выбрать те параметры и топики, которые мы хотим отслеживать



СУБД поддерживает несколько языков запросов, один из которых SQL и еще один FLUX



Пример запроса на языке FLUX

```
from(bucket: "IoT")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "mqtt_consumer")
  |> filter(fn: (r) => r["_field"] == "value")
  |> filter(fn: (r) => r["host"] == "e81d2a776ed2")
  |> filter(fn: (r) => r["topic"] == "gas" or r["topic"] == "humidity" or
r["topic"] == "pressure" or r["topic"] == "temperature")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

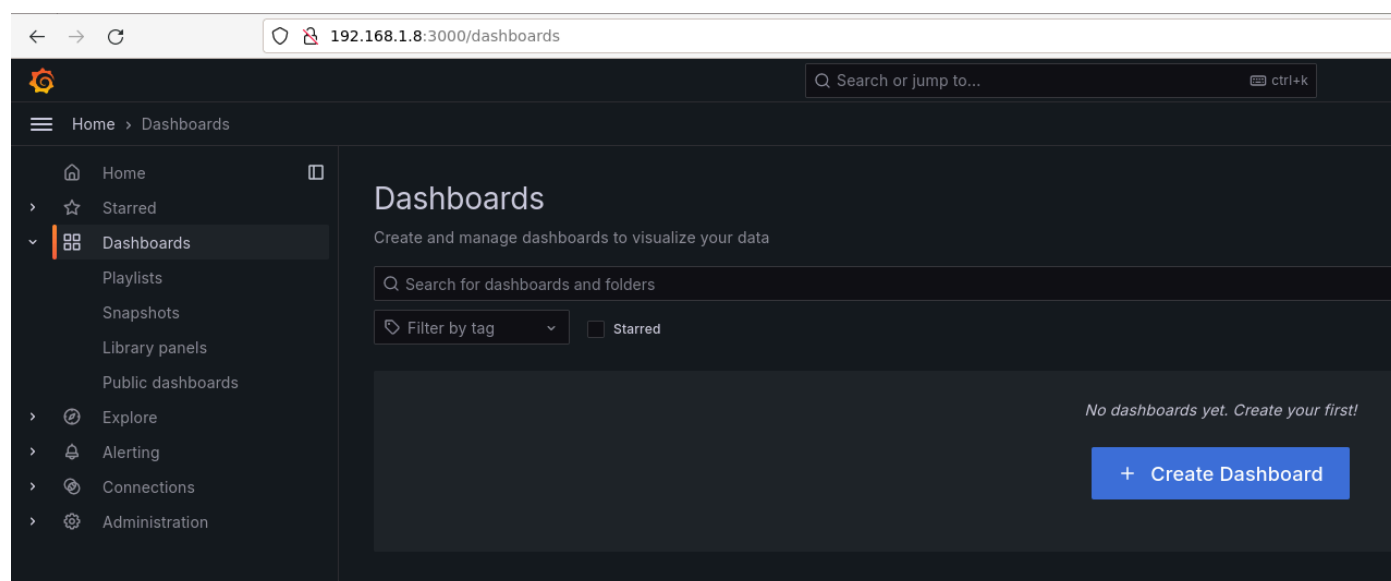
Данный запрос можно скопировать и использовать его в дальнейшем для построения визуализации в системе Grafana

Grafana

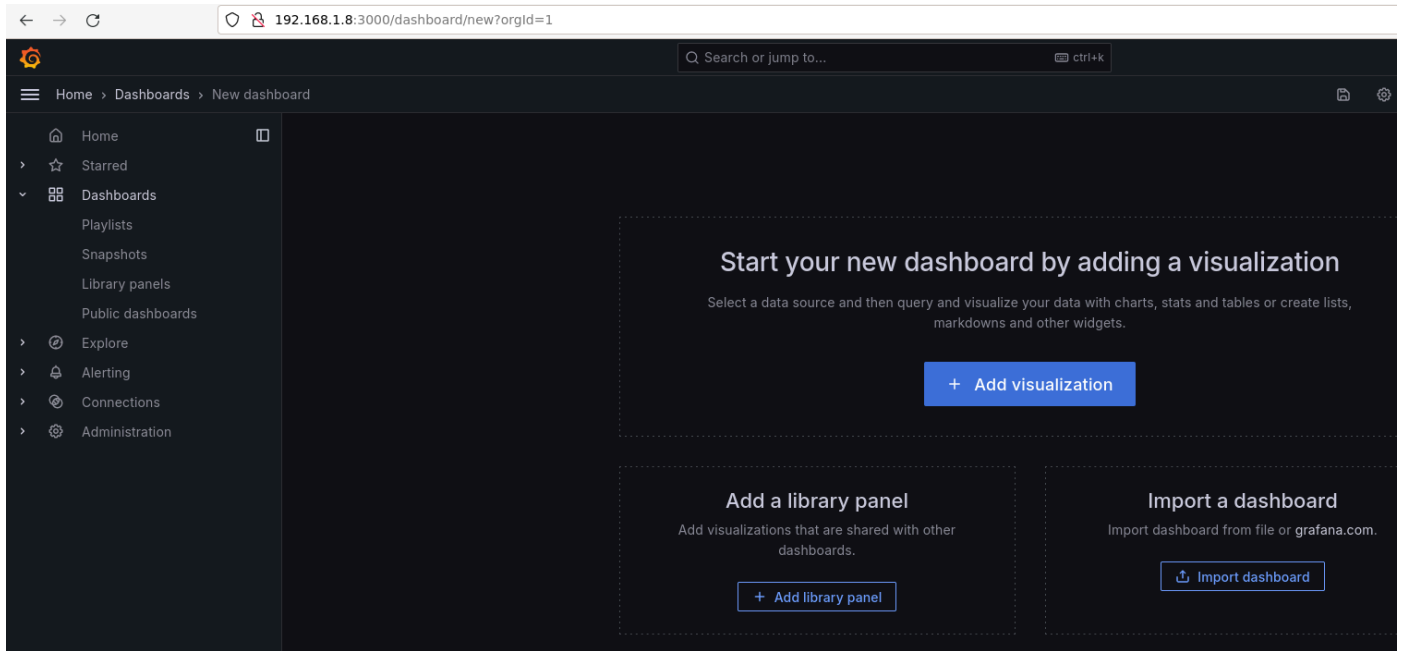
Графана - это система визуализации и анализа данных. Она поддерживает огромное количество типов данных и хранилищ, как встроенными средствами, так и с помощью сторонних плагинов.

Необходимо с помощью авторизационных данных (token) подсоединиться к InfluxDB.

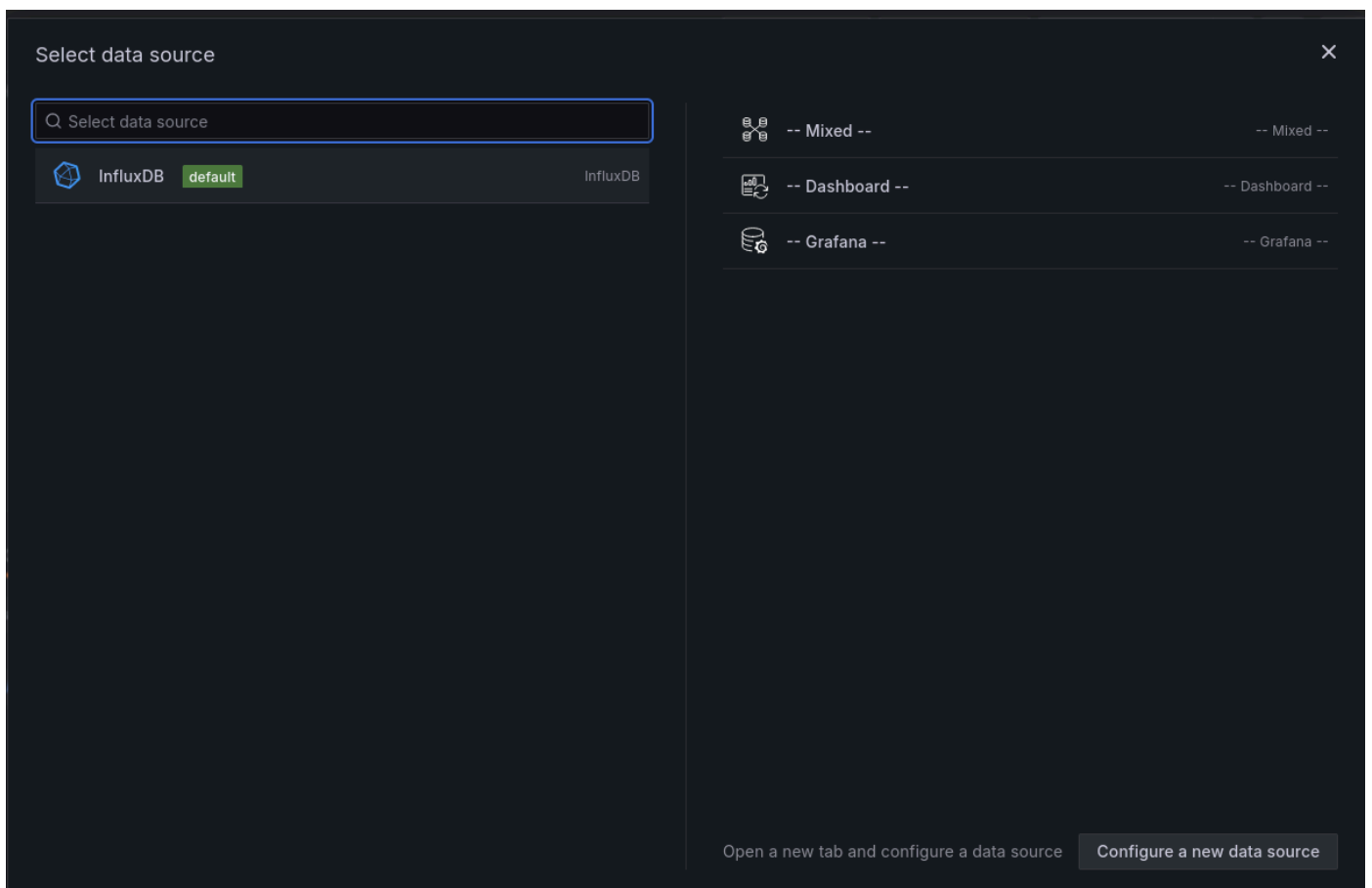
И в разделе Dashboards создать новый.



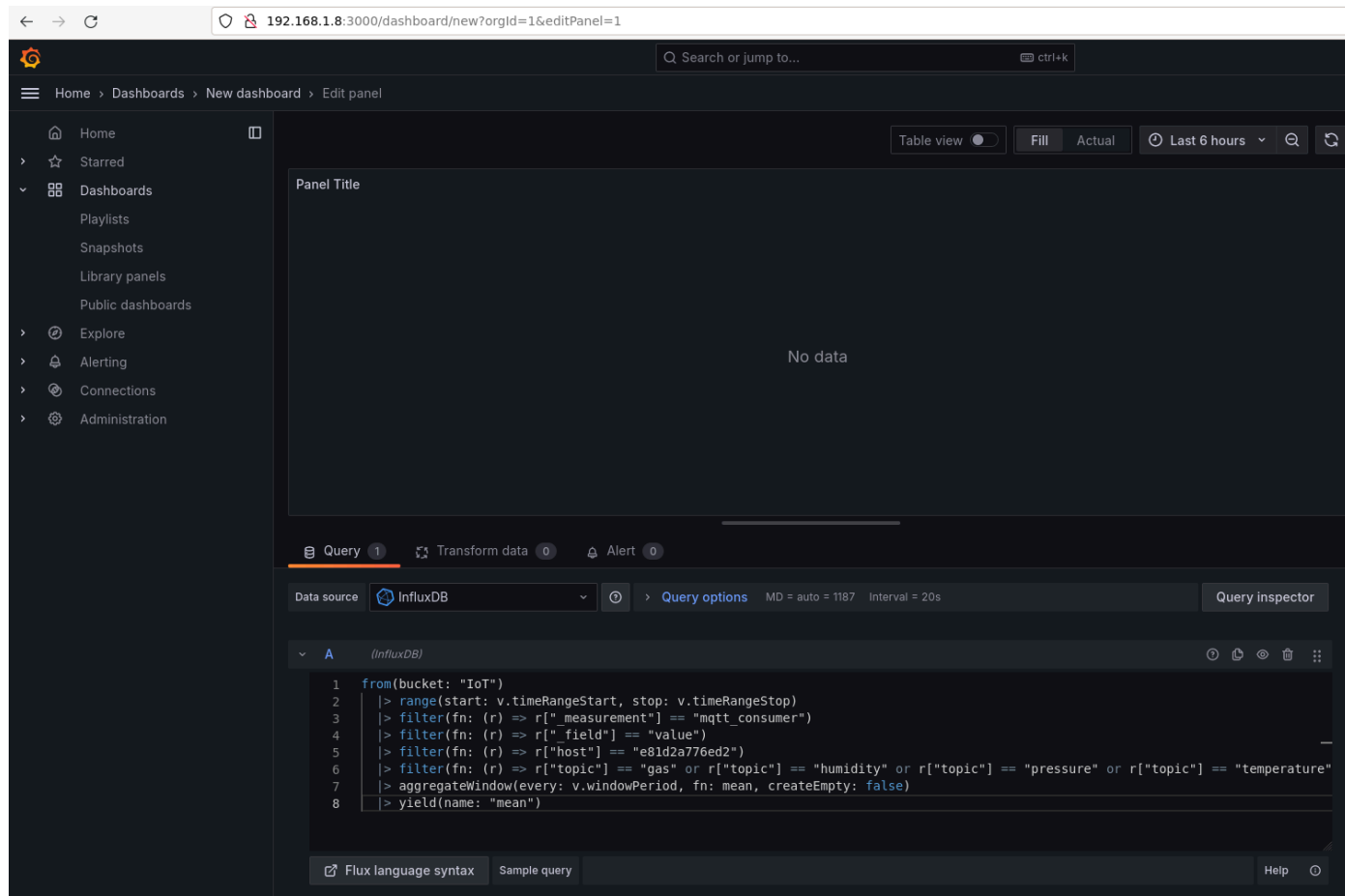
Можно создавать множество дашбордов на каждый случай жизни, причем можно использовать всевозможные комбинации одних и тех же показателей с разными системами представления.



При создании нового дашборда, отобразится список имеющихся соединений с хранилищами, в нашем случае это InfluxDB



Ранее созданный в InfluxDB запрос можно использовать в дашборде



Обязательно необходимо нажать на кнопку "Save" и сохранить дашборд, если нажать "Apply", то информация не сохранится.

+

?

Discard

Save

Apply

Time series

Search options

All

Overrides

Panel options

Title

odroid

Description

Показания температуры, давления, влажности и загазованности

Transparent background

Panel links

Repeat options

Tooltip

Tooltip mode

Single

All

Hidden

Hover proximity

How close the cursor must be to a point to trigger the tooltip, in pixels

Legend

Visibility

Mode

List

Table

Placement

Bottom

Right

На дашборде выведена информация о четырех показателях. Спидометры которые показывают текущее значение и график за час. Временные рамки можно гибко настраивать.



Доступ в интернет

С помощью веб-сервера Nginx организован доступ ко всем компонентам системы. Без порт-маппинга и файрвола. Только средствами веб-сервера. Включая доступ к mosquitto по веб-сокету

Пример конфига nginx.conf

```
server {
    listen 443 ssl;
    server_name mosquitto-rrg-5471.gb-iot.ru;

    location / {
        proxy_http_version 1.1;
        proxy_pass http://192.168.1.8:8081;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
    }

    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}

server {
    listen 443 ssl;
    server_name influxdb-rrg-5471.gb-iot.ru;
    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
```

```

include /etc/letsencrypt/options-ssl-nginx.conf;
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

location / {
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_pass http://192.168.1.8:8086;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
}

server {
    listen 443 ssl;
    server_name grafana-rrg-5471.gb-iot.ru;
    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_pass http://192.168.1.8:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

server {
    listen 443 ssl;
    server_name nodered-rrg-5471.gb-iot.ru;
    ssl_certificate /etc/letsencrypt/live/gb-iot.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/gb-iot.ru/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_pass http://192.168.1.8:1880;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

```

Прошивка OpenWRT с установленным пакетом DDNS позволяет постоянно обновлять доменные зоны. И легко находить доступ по имени.

Список литературы:

1. [BME680 – Datasheet 1.9 02-2024](#) - 53 страниц
2. [BME680 – Datasheet 1.3 07-2019](#) - 54 страниц

3. [BSEC Integration Guideline](#) - 56 страниц
4. [BSEC Binary Size Information](#) -3 страницы
5. [BME68x - Shipment packaging details](#) - 9 страниц
6. [BME680 – Application Note](#) - 13 страниц
7. [BME680 Integrated Environmental Unit](#) - 2 страниц
8. [BME680: Handling, Soldering and Mounting Instructions](#) - 16 страниц
9. [BME680 Shuttle board 3.0 flyer](#) - 2 страницы
10. [Release Notes 2.5.0.2](#)

*Итого, около 200 страниц технической информации. Т.к. информации не для публикации, то ссылки на данные материалы ведут в закрытый репозиторий.