

Итоговая работа по курсу «Особенности С для IoT»

Казаков Владимир Сергеевич




Описание проекта

Проект по разработке системы сбора информации для «умного дома», построенную на основе протокола MQTT, одноплатного компьютера Raspberry Pi 4 и датчика температуры и влажности DHT_11. В процессе выполнения проекта возникли сложности с правильным опросом датчика DHT_11. Для решения этой проблемы применил библиотеку **WiringPi** и настроил интервалы опроса датчика согласно инструкции(Datasheet DHT_11).

Описание системы

Исходный код программы

 <https://github.com/VlKaz/>

Описание системы

Разворачивание системы происходит в несколько этапов:

1. Настройка виртуального сервера на базе операционной системы Linux (настройка брокера, Nod-red, Influx DB, Grafana)
2. Настройка и подключение к сети одноплатного компьютера Raspberry Pi4 (для визуального управления использовался **vncserver**)
3. Написание самого консольного приложения для одноплатного компьютера, по сбору информации с датчика и отправки ее в реальном времени.

Настройка виртуального сервера сервера.

После настройки виртуального сервера (см. Введение в IoT) получаем необходимые данные для работы с ним:

Логин в систему (и доступ по ssh)

root/student1

Подключение к mosquitto брокеру порт 1883:

IoT/student1

Подключение к node-red порт 1880 через браузер:

admin/student1

Подключение к influxdb порт 8086 через браузер:

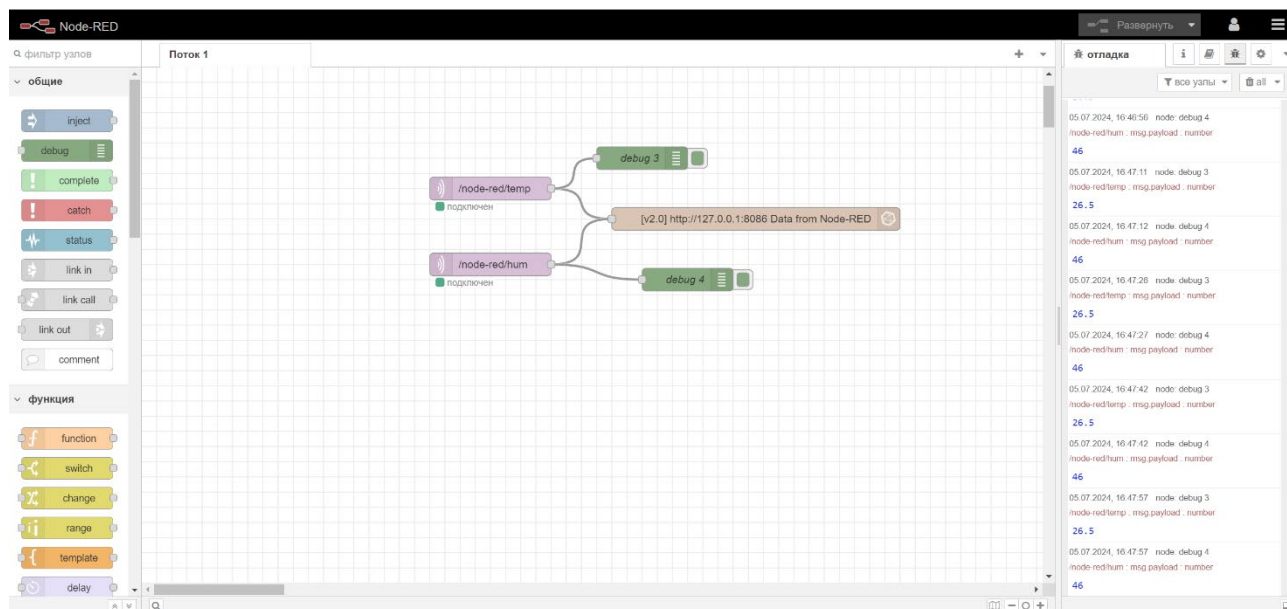
admin/student1

Подключение к grafana порт 3000 через браузер:

admin/student1

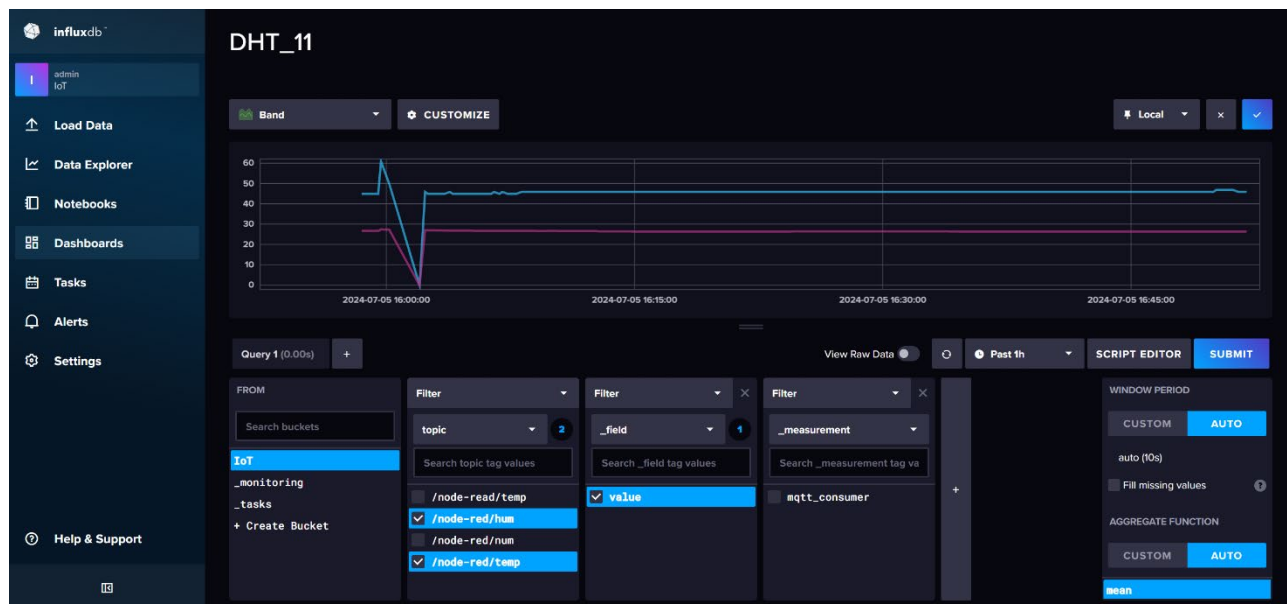
Подробнее остановимся на некоторых моментах:

NODE-RED



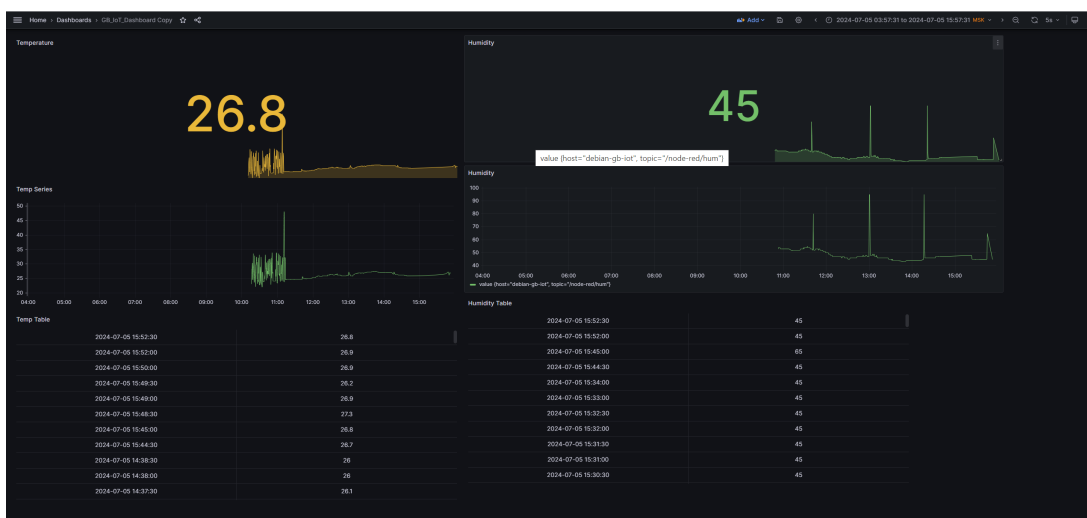
Создано два топика. Первый для сбора температуры /node-red/temp. Второй для сбора влажности /node-red/hum.

Influx DB



Настроен прием значений по двум топикам.

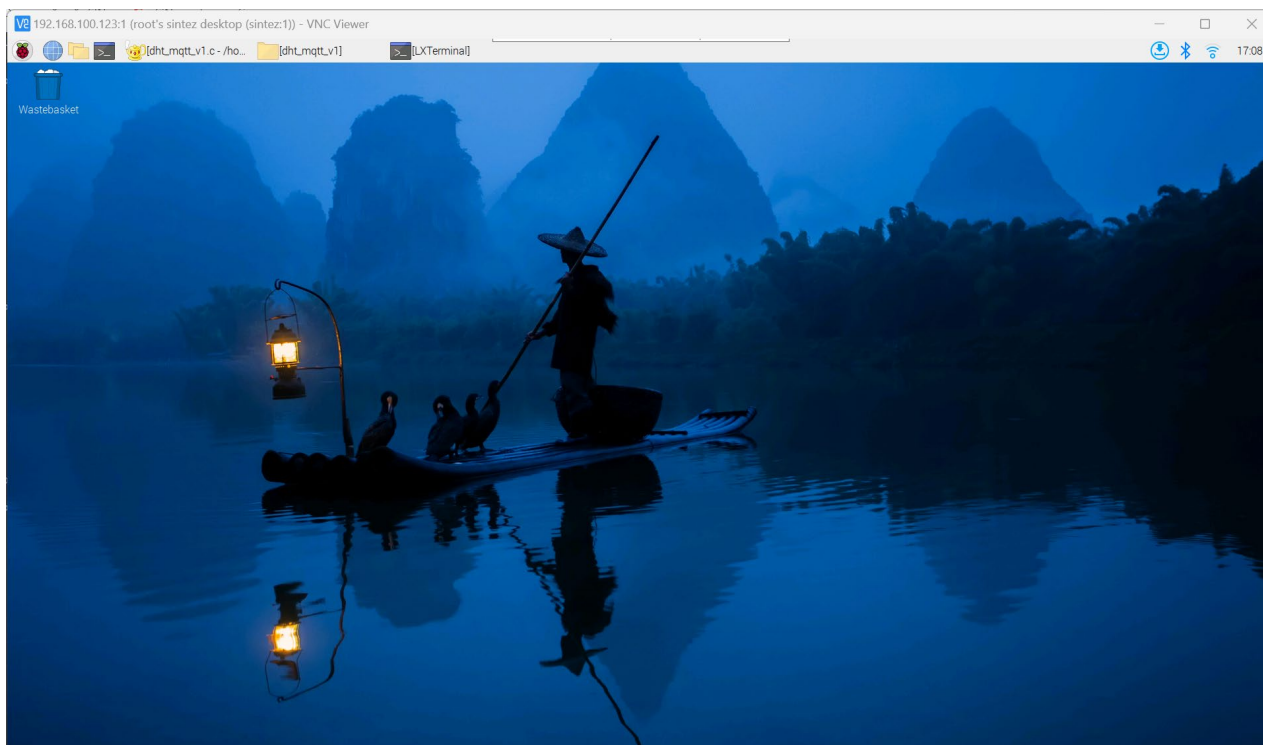
GRAFANA



Настроено визуальное отображение изменения температуры (желтый график) и влажности (Зеленый график). Сбор информации в режиме реального времени.

Настройка и подключение к сети одноплатного компьютера Raspberry Pi4

Настройка и подключение к сети одноплатного компьютера Raspberry Pi4 производится при помощи программы **Raspberry Pi Imager** с использованием последнего из предлагаемых по умолчанию 64-х битного образа (см. сайт производителя). После обновления и настройки VNCserver на самом Raspberry, мы можем подключиться к нему по сети через VNC Viewer.



Написание самого консольного приложения

Прежде чем приступить к написанию приложения необходимо дополнительно установить библиотеку для работы с GPIO (WiringPi) используя следующие команды:

```
# fetch the source
    sudo apt install git
    git clone https://github.com/WiringPi/WiringPi.git
    cd WiringPi

# build the package
    ./build debian
    mv debian-template/wiringpi-3.0-1.deb .

# install it
    sudo apt install ./wiringpi-3.0-1.deb
```

и библиотеку для работы с MQTT(paho_mqtt) используя следующие команды:

```
git clone https://github.com/eclipse/paho.mqtt.c.git
cd /путь до папки/paho.mqtt.c
make
sudo make install
```

1. Приложение для работы с датчиком DHT_11 называется DHT_MQTT.

Приложение состоит из 3-х файлов:

dht_mqtt_v1 – исполняемый файл самой программы для ОС Linux.

dht_mqtt_v1.c – файл с исходным кодом на языке СИ.

dht_mqtt_v1.o – объектный файл который получается в процессе компиляции.

Сборка программы производилась в терминале с использованием следующей команды:

```
gcc -o dht_mqtt_v1 dht_mqtt_v1.c -lwiringPi -lwiringPiDev -lpaho-mqtt3cs
```

Описание и демонстрация работы программы

Программа запускается при помощи командой строки в терминале. Нужно перейти в каталог с программой и запустить файл *dht_mqtt_v1* при этом нужно указать 5 параметров через пробел в следующей последовательности:

1. Адрес и порт – в моем случае 192.168.100.120:1883

2. Имя пользователя – IoT
3. Пароль – student1
4. Топик для температуры - /node-red/temp
5. Топик для влажности - /node-red/hum

Если же параметры не будут введены по то выскачет подсказка с примером:

```

root@sintez:/home/kaz/dhtWiringPi/dht_mqtt_v1# ./dht_mqtt_v1 192.168.100.120:1883 IoT student1 /node-red/temp
***Information to help you run the program correctly***

=====
For the program to work correctly, you must specify 5 parameters separated by a space:
1. Address and port
2. Username
3. Password
4. Topic for Temperature
5. Topic for Humidity
=====
* example: ./dht_mqtt_v1 192.168.100.120:1883 IoT student1 /node-red/temp /node-red/hum *
=====
root@sintez:/home/kaz/dhtWiringPi/dht_mqtt_v1#

```

После правильного ввода параметров, вам будет предложено ввести интервал опроса датчика и отправки запроса в секундах (устанавливаем 10 сек.):

```

root@sintez:/home/kaz/dhtWiringPi/dht_mqtt_v1# ./dht_mqtt_v1 192.168.100.120:1883 IoT student1 /node-red/temp /node-red/hum
Enter interval for sending packets in seconds: 10

```

если все корректно, то программа заработает как показано на рисунке ниже:

```

root@sintez:/home/kaz/dhtWiringPi/dht_mqtt_v1# ./dht_mqtt_v1 192.168.100.120:1883 IoT student1 /node-red/temp /node-red/hum
Enter interval for sending packets in seconds: 10

Raspberry Pi wiringPi DHT11 Temperature and Humidity
*****START TO WORK*****
Humidity = 47.0 % Temperature = 26.2 C (78.8 F)
Temperature with delivery token 1 delivered
Humidity with delivery token 2 delivered

```

программа выводит в консоль данные о температуре и влажности так же токены отправки в топики, это можно видеть **node-red**:

⌘ отладка

i

▼

▼ все узлы ▼

all ▼

/node-red/hum : msg.payload : number

47

05.07.2024, 18:05:20 node: debug 3

/node-red/temp : msg.payload : number

26.2

05.07.2024, 18:05:21 node: debug 4

/node-red/hum : msg.payload : number

47

05.07.2024, 18:05:31 node: debug 3

/node-red/temp : msg.payload : number

26.2