Raspberry Pi 4 with Raspbian OS bullseye

* MQTT-broker
* Node-Red
* InfluxDB

ESP32

* DHT11-sensor
* Servo-motor
* LED
* MQTT-publisher
* MQTT-subscriber
* Ultrasonic-sensor HC-SR04

You can use Raspberry Pi 4 with VNC or SSH

To make VNC more responsive, comment out dtoverlay=vc4-kms-v3d and max\_framebuffers=2 from /boot/config.txt -file. Then change screen resolution from Start -> Preferences -> Raspberry Pi Config.

Install Node-Red to Raspberry Pi 4 (<https://nodered.org/docs/getting-started/raspberrypi>)

§ sudo apt install build-essential git curl

§ bash <(curl -sL <https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered>)

If you want Node-RED to run when the Pi is turned on:

§ sudo systemctl enable nodered.service

§ sudo systemctl disable nodered.service

You can access Node-RED through browser: http://<RASPBERRY\_PI\_IP>:1880

Install Mosquitto MQTT-broker to Raspberry Pi 4

(<https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/>)

**§ sudo apt install -y mosquitto mosquitto-clients**

**If you want to run mosquito when the Pi is turned on:**

**§ sudo systemctl enable mosquitto.service**

**Check that mosquito is running:**

service mosquitto status

You must add couple of commands to mosquitto .conf-file located at /etc/mosquitto/mosquitto.conf. You can access the file with § sudo nano /etc/mosquitto/mosquitto.conf.

Add next line at the bottom of the file:

listener 1883

If you want to allow anonymous usage of the broker, add:

allow\_anonymous true

If you want to allow only registered users to use the broker, add:

allow\_anonymous false

per\_listener\_settings true

password\_file /etc/mosquitto/pw.txt

You can change pw.txt -file to something else, you will define the file at the next step.

Save .conf -file by pressing CTRL+O, exit with CTRL+X.

You can create a password-file by typing in terminal:

§ mosquitto\_passwd -c /etc/mosquitto/pw.txt username

**You can check that MQTT-broker is working by opening two terminal-windows and writing in one:**

**§ mosquitto\_sub -h <RASPBERRY\_IP> -t test/test**

**This window listens to test/test -mqtt-topic.**

**Open another terminal window and write:**

**§ mosquitto\_pub -h <RASPBERRY\_IP> -t test/test -m “this is a test”**

**This will publish a message to test/test -mqtt-topic which you can see in the first terminal-window.**

**You can close mosquitto\_sub by pressing CTRL+C.**

**Install InfluxDB to Raspberry Pi 4**

§ curl https://repos.influxdata.com/influxdata-archive.key | gpg --dearmor | sudo tee /usr/share/keyrings/influxdb-archive-keyring.gpg >/dev/null

§ echo "deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg] https://repos.influxdata.com/debian $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/influxdb.list

§ sudo apt update

§ sudo apt install influxdb

To start InfluxDB when Raspberry Pi 4 is turned on:

§ sudo systemctl unmask influxdb

§ sudo systemctl enable influxdb

Start InfluxDB:

§ sudo systemctl start influxdb

Check that InfluxDB is running:

§ sudo service influxdb status

Create database to InfluxDB:

Start IinfluxDB-CLI with the command influx

Create a database with command create database

Check that database is created with show databases

Installing Grafana to Raspberry Pi 4

§ wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

§ echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list

§ sudo apt-get update

§ sudo apt-get install -y grafana

To start Grafana when Raspberry Pi is turned on:

§ sudo /bin/systemctl enable grafana-server

Starting Grafana:

§ sudo /bin/systemctl start grafana-server

Grafana can be found at http://<RASPBERRY\_PI\_IP>:3000

Username admin, Password admin.

To add data to show in Grafana, you have to save DHT11-sensordata in Node-RED to InfluxDB which is explained later in this guide.

When you have DHT11-data in InfluxDB-database, you can add that database to data source in Grafana:

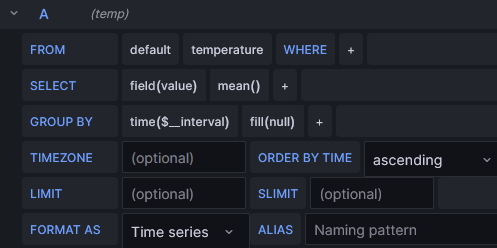
Add new data source from the Configuration-button from the left side in the Grafana UI.

Configuration -> Data sources

Add new datasource

You can name it what ever

In URL, you put influxdb-server url (in this case localhost or Raspberry Pi IP), eg. localhost:8086

In details add InfluxDB.

In database add temp.

Dashboard -> add new.

FROM -> temperature.

Now data should be visible

in the graph.

ESP32

Connect the sensors, LED, ultrasonic ranger, and servomotor. Take a note of the pins.

Install Arduino IDE.

Install the following libraries, you can add .zip-files in Arduino IDE using library manager:

PubSubClient (https://github.com/knolleary/pubsubclient)

DHT11-library (<https://www.arduino.cc/reference/en/libraries/dht-sensor-library/>)

UltraSonic-library (<https://github.com/Seeed-Studio/Seeed_Arduino_UltrasonicRanger/>)

Servo-library (https://github.com/RoboticsBrno/ServoESP32)

Code for ESP32 can be found at <https://github.com/severinkangas/raspi-esp32-sensorstack/blob/main/esp32.ino>

Node-RED flow

Node-RED flow can be found at <https://github.com/severinkangas/raspi-esp32-sensorstack/blob/main/node-red.txt>

Kuva, joka sisältää kohteen diagrammi

Kuvaus luotu automaattisesti

Nodes installed:

node-red-contrib-influxdb

node-red-contrib-whatsapp-cmb

node-red-dashboard

node-red-contrib-ibm-watson-iot

Saving DHT11-sensordata to InfluxDB

Move influxdb-out node to the flow.

Click the node and create new server.

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

Deploy.

You can check that data is being saved to database by opening influxdb in Raspberry Pi terminal by command:

§ influx

use temp

show measurements

SELECT \* FROM temperature