

MaxAir and MQTT

MaxAir can both send and receive information using the MQTT protocol.

Message Queueing Telemetry Transfer, or MQTT, is a lightweight IP-based messaging protocol designed for communication between sensors, controllers, and other devices. It's designed to support equipment that may not always be online, like automated devices built with microcontrollers. MQTT server programs are called **brokers**. A broker keeps track of messages from clients, and allows any client to query the last message sent by another client.

Messages are organized into **topics**. Typically, a topic represents a device, with each sub-topic representing its characteristics. For example, a weather station might have the main topic "station" with subtopics "temperature", "humidity", "air quality", and so forth. The weather station itself would send messages to each of the subtopics, and a web client might subscribe to those topics to graph them onscreen over time.

Clients either publish new messages to topics, or subscribe to topics, and the broker notifies them when new messages arrive. For this reason, MQTT is known as a **Publish & Subscribe**, or **PubSub** system.

The MaxAir Gateway script `/var/www/gateway.py` together with the Python library `paho-mqtt` are used to send and receive MQTT data.

MaxAir will require access to a Mosquitto Broker, which can exist on the same device hosting MaxAir or on a separate device.

MaxAir will require an account on the Mosquitto Broker which it can access.

Example Configuration

- The Mosquitto Broker will be installed on the same device which is hosting MaxAir.
- A sensor device will be employed which uses a DS18B20 1-wire temperature sensor, interfaced to a WeMos D1 Mini microcontroller, running the Tasmota software package.
- A Sonoff Basic Module running the Tasmota software package will be configured as a relay.

Installing Mosquitto

- From the linux command line execute ***'apt-get install mosquitto mosquitto-clients'***
- From the linux command line execute ***'systemctl enable mosquitto'***
- From the linux command line execute ***'mosquitto_passwd -c /etc/mosquitto/credentials admin'***
- Enter the password ***'pihome'*** and confirm
- Create and edit a new file from the linux command line by executing the command ***'nano /etc/mosquitto/conf.d/maxair.conf'***
- Add the following 3 lines and save the file
 - `per_listener_settings true`
 - `allow_anonymous false`

- password_file /etc/mosquitto/credentials
- If not already available then install paho-mqtt using the command '**pip3 install paho-mqtt**'
- Now, test the installation by running the following command '**mosquitto -v**'

Install and configuring zigbee2mqtt

Before that, you may need to install **nodejs** and **npm**.

Run command **uname -m** or **cat /proc/cpuinfo** to find out model name processor of SBC and from <https://nodejs.org/en/download/> find out the current version of **nodejs** and **npm** for your SBC. For example – for Raspberry Pi 3B and version Rasbian OS “Buster” with ARMv7 LTS Version nodejs 16.13.1 (includes npm 8.1.2).

Install nodejs and npm from the NodeSource Repo

- From the linux command line execute '**curl -fsSL https://deb.nodesource.com/setup_16.x | bash -**'
- From the linux command line execute '**sudo apt install nodejs**'
- Verify installation '**node -v**' '**npm -v**'

Install zigbee2mqtt from

https://www.zigbee2mqtt.io/guide/installation/01_linux.html

Connect the USB stick to your SBC, it is better to use a USB-to-USB cable for this purpose.

For clone Zigbee2MQTT repository and install use command:

- '**git clone https://github.com/Koenkk/zigbee2mqtt.git**'
- '**sudo mv zigbee2mqtt /opt/zigbee2mqtt**'
- '**cd /opt/zigbee2mqtt**'
- '**npm ci**'

Before run start Zigbee2MQTT you need to edit the **configuration.yaml** file. This file contains the configuration which will be used by Zigbee2MQTT. Open the configuration file

- '**nano /opt/zigbee2mqtt/data/configuration.yaml**'



```

192.168.1.5 - PuTTY
root@MaxAir:/# nano /opt/zigbee2mqtt/data/configuration.yaml
/opt/zigbee2mqtt/data/configuration.yaml

homeassistant: false
permit_join: true
mqtt:
  base_topic: zigbee2mqtt
  server: mqtt://localhost
  user: 
  password: ' '
  serial:
    port: /dev/ttyUSB0

```

The field **permit_join: true** will allow to pair **zigbee devices** with a **zigbee2mqtt**. In field **MQTT server authentication** uncomment fields **user** and **password** and set the following parameters that you used in the settings for Mosquitto Broker. If you don't know **port** parameter for your USB stick use the command '**ls -l /dev/serial/by-id**'

Now you ready to start **zigbee2mqtt** service by commands: **cd /opt/zigbee2mqtt**, then: **npm start** and pair the zigbee device. When all is well, you will see this in the console messages. After this you can reopen the **configuration.yaml** file and edit field **friendly_name** in section **devices** as you see in picture.

```
192.168.1.5 - PuTTY
on next report of device.", "name": "pressure_calibration", "property": "pressure_calibration", "type":
: "numeric"}], "supports_ota": false, "vendor": "Xiaomi"}, "friendly_name": "0x00158d000548a334", "ieee_ad
dress": "0x00158d000548a334", "status": "successful", "supported": true, "type": "device_interview"}'
info 2022-01-09 16:12:38: Configuring '0x00158d000548a334'
info 2022-01-09 16:12:38: Successfully configured '0x00158d000548a334'
info 2022-01-09 16:12:45: MQTT publish: topic 'zigbee2mqtt/0x00158d000548a334', payload '{"linkqu
ality":153,"temperature":21.61}'
info 2022-01-09 16:12:45: MQTT publish: topic 'zigbee2mqtt/0x00158d000548a334', payload '{"humidi
ty":54.5,"linkquality":150,"temperature":21.61}'
info 2022-01-09 16:12:45: MQTT publish: topic 'zigbee2mqtt/0x00158d000548a334', payload '{"humidi
ty":54.5,"linkquality":150,"pressure":968.7,"temperature":21.61}'
root@MaxAir: /opt/zigbee2mqtt/data/log/2022-01-08.22-57-50# nano /opt/zigbee2mqtt/data/configuratio
n.yaml
GNU nano 3.2 /opt/zigbee2mqtt/data/configuration.yaml

homeassistant: false
permit_join: true
mqtt:
  base_topic: zigbee2mqtt
  server: mqtt://localhost
  user: admin
  password: 'l67l'
serial:
  port: /dev/ttyUSB0
advanced:
  homeassistant_legacy_entity_attributes: false
  legacy_api: false
device_options:
  legacy: false
devices:
  '0x00158d000549b4f0':
    friendly_name: AqaraT_1
  '0x00158d000548a334':
    friendly_name: '0x00158d000548a334'
```

Running zigbee2mqtt as a daemon with systemctl

Use instruction from https://www.zigbee2mqtt.io/guide/installation/01_linux.html#installing to do this and now information about operation of zigbee devices you can view in log files and with the command **'sudo journalctl -u zigbee2mqtt.service -f'**.

```
192.168.1.5 - PuTTY
drwxr-xr-x 2 root root 4096 Jan  8 22:57 2022-01-08.22-57-50
drwxr-xr-x 2 root root 4096 Jan  9 16:08 2022-01-09.16-08-47
drwxr-xr-x 2 root root 4096 Jan  9 16:44 2022-01-09.16-44-40
root@MaxAir: /opt/zigbee2mqtt/data/log# cat 2022-01-09.16-44-40/log.txt
info 2022-01-09 16:44:44: Logging to console and directory: '/opt/zigbee2mqtt/data/log/2022-01-09.16
-44-40' filename: log.txt
info 2022-01-09 16:44:44: Starting Zigbee2MQTT version 1.22.2 (commit #301ba14b)
info 2022-01-09 16:44:44: Starting zigbee-herdsman (0.13.188)
info 2022-01-09 16:44:46: zigbee-herdsman started (resumed)
info 2022-01-09 16:44:46: Coordinator firmware version: '{"meta":{"maintrel":1,"majorrel":2,"minorrel":7,"product":1,"revision":20210708,"transportrev":2},"type":"zStack3x0"}'
info 2022-01-09 16:44:46: Currently 2 devices are joined:
info 2022-01-09 16:44:46: AqaraT_1 (0x00158d000549b4f0): WSDCGQ11LM - Xiaomi Aqara temperature, humi
dity and pressure sensor (EndDevice)
info 2022-01-09 16:44:46: Aqara_2 (0x00158d000548a334): WSDCGQ11LM - Xiaomi Aqara temperature, humid
ity and pressure sensor (EndDevice)
warn 2022-01-09 16:44:46: 'permit_join' set to 'true' in configuration.yaml.
warn 2022-01-09 16:44:46: Allowing new devices to join.
warn 2022-01-09 16:44:46: Set 'permit_join' to 'false' once you joined all devices.
info 2022-01-09 16:44:46: Zigbee: allowing new devices to join.
info 2022-01-09 16:44:46: Connecting to MQTT server at mqtt://localhost
info 2022-01-09 16:44:46: Connected to MQTT server
info 2022-01-09 16:44:46: MQTT publish: topic 'zigbee2mqtt/bridge/state', payload 'online'
info 2022-01-09 16:44:47: MQTT publish: topic 'zigbee2mqtt/AqaraT_1', payload '{"battery":91,"humidi
ty":77.87,"pressure":964.4,"temperature":10.64,"voltage":2985}'
info 2022-01-09 16:44:47: MQTT publish: topic 'zigbee2mqtt/Aqara_2', payload '{"humidity":51.77,"pre
ssure":969,"temperature":19.69}'
root@MaxAir: /opt/zigbee2mqtt/data/log# sudo journalctl -u zigbee2mqtt.service -f
-- Logs begin at Sat 2022-01-08 22:57:24 EET. --
Jan 08 22:57:35 MaxAir systemd[1]: Started zigbee2mqtt.
Jan 09 16:44:06 MaxAir systemd[1]: Stopping zigbee2mqtt...
Jan 09 16:44:06 MaxAir systemd[1]:
Jan 09 16:44:16 MaxAir systemd[1]: zigbee2mqtt.service: Succeeded.
Jan 09 16:44:16 MaxAir systemd[1]: Stopped zigbee2mqtt.
Jan 09 16:44:35 MaxAir systemd[1]: Started zigbee2mqtt.
```

As you see from picture full topic for new zigbee device include **base_topic** of service and **friendly_name** of device: '**zigbee2mqtt/Aqara_2**' and **JSON Attribute** is '**battery**', '**temperature**', etc.

Configure MaxAir to Communicate Zigbee devices Using Mosquitto Broker

Create an MQTT Connection for MQTT node

If you use services Mosquitto Broker and Zigbee2MQTT on the same SBC as HVAC, set in the field IP address of the localhost

Edit MQTT Connection

Name

MaxAirMQTT

IP

127.0.0.1

Port

1883

Username

Password

Enabled

Enabled

Type

MQTT Node

Edit Conn

Close

Create MQTT Type Nodes for Both a Sensor and a Controller

See document '[setup_guide_mqtt](#)'

Create MQTT Devices

In field **MQTT Device Name** set **friendly_name** of **zigbee device**, in field **MQTT Topic** – full topic of device, in field **JSON Attribute** - selected **attribute** of device (see '[Running zigbee2mqtt as a](#)

Edit MQTT Device: Aqara_2

Node Type

MQTT Controller or MQTT Sensor

MQTT Sensor

MQTT Device Name

Identification for the MQTT Device

Aqara_2

Child ID

Node Child ID for This MQTT Device

3

MQTT Topic

MQTT Topic to subscribe to for sensors or to which publish the messages for relays

zigbee2mqtt/Aqara_2

JSON Attribute

Leave blank if the Sensor sends raw data to the topic

temperature

+ Add MQTT Device

Node Type

MQTT Controller or MQTT Sensor

MQTT Sensor

MQTT Device Name

Identification for the MQTT Device

Aqara_2

Child ID

Node Child ID for This MQTT Device

4

MQTT Topic

MQTT Topic to subscribe to for sensors or to which publish the messages for relays

zigbee2mqtt/Aqara_2

JSON Attribute

Leave blank if the Sensor sends raw data to the topic

humidity

daemon with systemctl').

Adding zigbee device as a New Sensor

Edit Sensor: AqaraH_2

☐ Before System Controller When Sensor is NOT Allocated to a Zone, Locate Tile either Before or At

Index Number In the List of sensors where you want to place this sensor on home screen

6

Sensor Type Temperature, Humidity, etc

Humidity

Sensor Name Select either Outside Weather or Sensor to be used to calculate the Start Time Offset Applied.

AqaraH_2

Sensor ID Node ID for the Sensor

31 - MQTT Sensor

Sensor Child ID Node Child ID for the Sensor

4

Sensor Correction Factor Positive or Negative Correction Factor

0.00

Home

One Touch

TIMER
MODE

Studio
22.1°
● 22.0°

Gas Boiler
●

CabinetT
19.3°
●

AqaraT_1
20.3°
●

AqaraH_1
52.2%
●

AqaraT_2
19.6°
●

AqaraH_2
52.5%
●

Boost
●

Outside: -6° C ☁ Fog - fog