

## MaxAir and ZigBee Devices

MaxAir can communicate with ZigBee devices using MQTT protocol and ZigBee toMQTT bridge. The bridge can be implemented using a suitable ZigBee adapter and 'Zigbee2MQTT', see <https://www.zigbee2mqtt.io/>.

**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.

Zigbee2MQTT will require access to the Mosquitto Broker.

### Example Hardware

A Sonoff ZigBee 3.0 USB Dongle (Model E) and Tuya Temperature/Humidity sensor were used to test the integration.



## Example Configuration

- Both the Mosquitto Broker and 'Zigbee2MQTT' will be installed on the same device which is hosting MaxAir.

## Installing Mosquitto

- The Mosquitto Broker can be installed and configured from 'Settings/System Maintenance/Install Software' menu option and select 'Install' for 'Mosquitto Broker'.
- If not already available then install paho-mqtt using the command '***pip3 install paho-mqtt***'.
- After the install has completed check that the mosquitto' service is running by executing on the commandline – 'sudo systemctl status mosquitto'.

## Installing Zigbee2MQTT

- Zigbee2MQTT can be installed from 'Settings/System Maintenance/Install Software' menu option and select 'Install' for 'Zigbee2MQTT Integration'.

## Zigbee2MQTT Onboarding

- Zigbee2MQTT will be run as a service, but requires initial configuration, see <https://www.zigbee2mqtt.io/guide/getting-started/#onboarding>.
- Start the zigbee2mqtt service from the command line using 'sudo systemctl start zigbee2mqtt.service'. After pressing the 'enter' key, the application will wait for the initial configuration to be completed. For example if the IP Address of the machine where Zigbee2MQTT is installed were 10.0.0.75, then browse to 10.0.0.75:8080 to open the Onboarding screen.
- Select the Adapter from the dropdown list

Found Devices

usb-ltead\_Sonoff\_Zigbee\_3.0\_USB\_Dongle\_Plus\_V2\_e666bc1785c2ef118323c3138148b910-if00-port0 (lthead), /dev/ttyUSB0, ember

Optionally allows to configure coordinator port and type (if known) automatically.

Coordinator/Adapter Port/Path

/dev/ttyUSB0

Coordinator/Adapter Type/Stack/Driver

ember

Coordinator/Adapter Baudrate

115200

Can be ignored for networked coordinators (TCP).

Coordinator/Adapter Hardware Flow Control ("rtscts: true")

☐

Can be ignored for networked coordinators (TCP).

- Enter the Username and Password for the Mosquitto Broker (zigbee/pihome) and check the 'Frontend enabled box.

MQTT User

zigbee

Optional. Set only if using authentication.

MQTT Password

.....

Optional. Set only if using authentication.

<https://www.zigbee2mqtt.io/guide/configuration/mqtt.html>

☒ Frontend enabled?

## Zigbee2MQTT Onboarding

Set the base configuration to start Zigbee2MQTT.

Optional fields will either be ignored or fallback to defaults if not set (see appropriate documentation page for more details).

If a field is disabled, it means [environment variables](#) are being used to override specific values (for example, through the Home Assistant add-on configuration page).

Found Devices

usb-ltead\_Sonoff\_Zigbee\_3.0\_USB\_Dongle\_Plus\_V2\_e666bc1785c2ef118323c3138148b910-if00-port0 (ltead), /dev/ttyUSB0, ember

Optionally allows to configure coordinator port and type (if known) automatically.

Coordinator/Adapter Port/Path

/dev/ttyUSB0

Coordinator/Adapter Type/Stack/Driver

ember

Coordinator/Adapter Baudrate

115200

Can be ignored for networked coordinators (TCP).

Coordinator/Adapter Hardware Flow Control ("rtscts: true")

☐

Can be ignored for networked coordinators (TCP).

<https://www.zigbee2mqtt.io/guide/configuration/adapter-settings.html>

Closest WiFi Channel

0

Optionally set to your closest WiFi channel to pick the best value for "Network channel" below.

Network Channel

11

Network Key

GENERATE

Network PAN ID

GENERATE

Network Extended PAN ID

GENERATE

<https://www.zigbee2mqtt.io/guide/configuration/zigbee-network.html>

MQTT Base Topic

zigbee2mqtt

MQTT Server

mqtt://localhost:1883

MQTT User

zigbee

Optional. Set only if using authentication.

MQTT Password

.....

Optional. Set only if using authentication.

<https://www.zigbee2mqtt.io/guide/configuration/mqtt.html>

☒ Frontend enabled?

Frontend Port

8080

<https://www.zigbee2mqtt.io/guide/configuration/frontend.html>

☐ Home Assistant enabled?

<https://www.zigbee2mqtt.io/guide/configuration/homeassistant.html>

Log Level

info

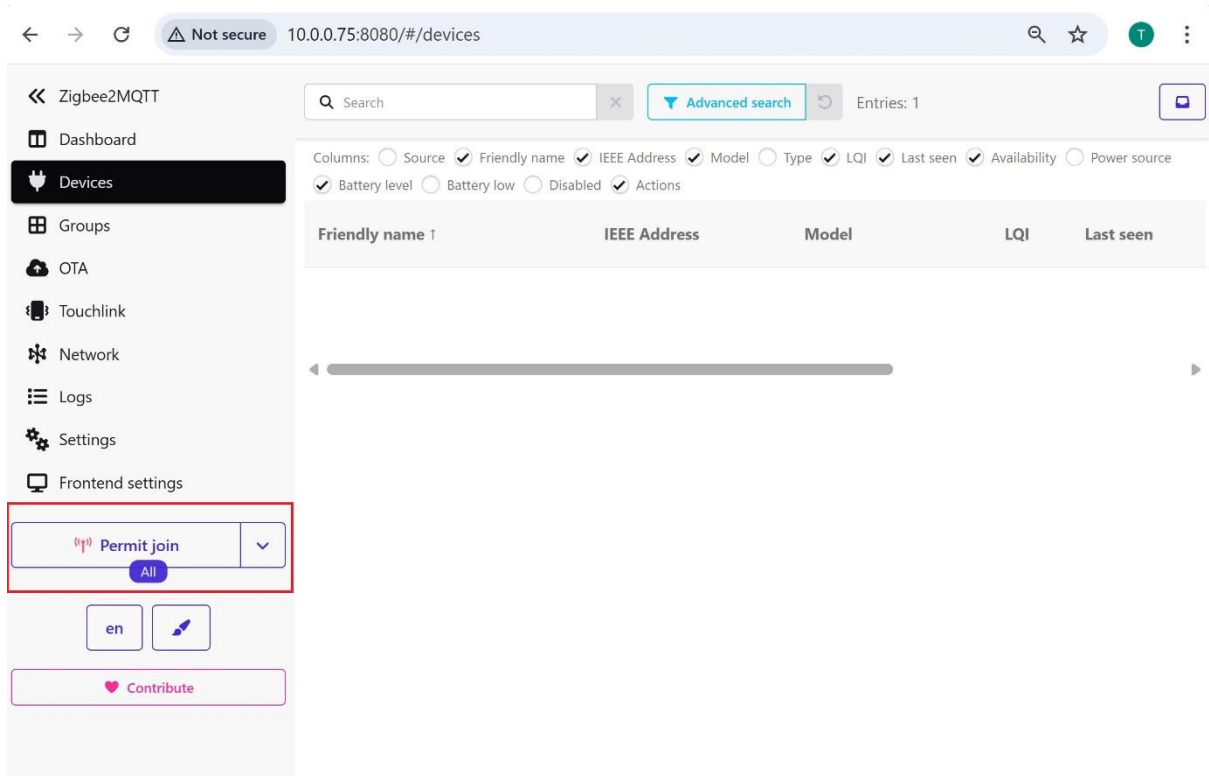
<https://www.zigbee2mqtt.io/guide/configuration/logging.html>

Submit

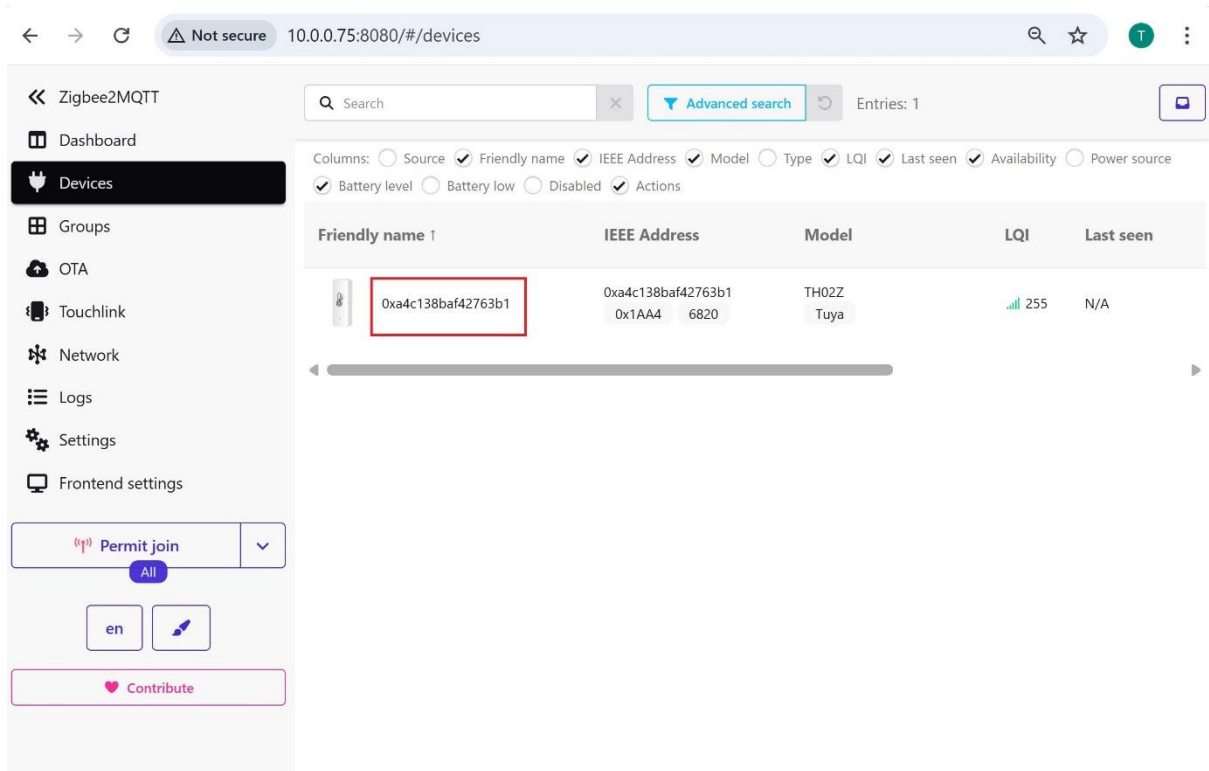
Once the configuration has been saved, the zigbee2mqtt service will move in to the running state.

## Adding a ZigBee Sensor to the ZigBee Network

- Open a browser connection to the Zigbee2MQTT Frontend, eg. 10.0.0.75:8080



- Click on the 'Permit join' button, then press the 'identify' button on the sensor device which you wish to join the network.
- Once the sensor has joined the network it will be displayed in the browser, note the 'eg.



## Configure MaxAir to Communicate Using MQTT

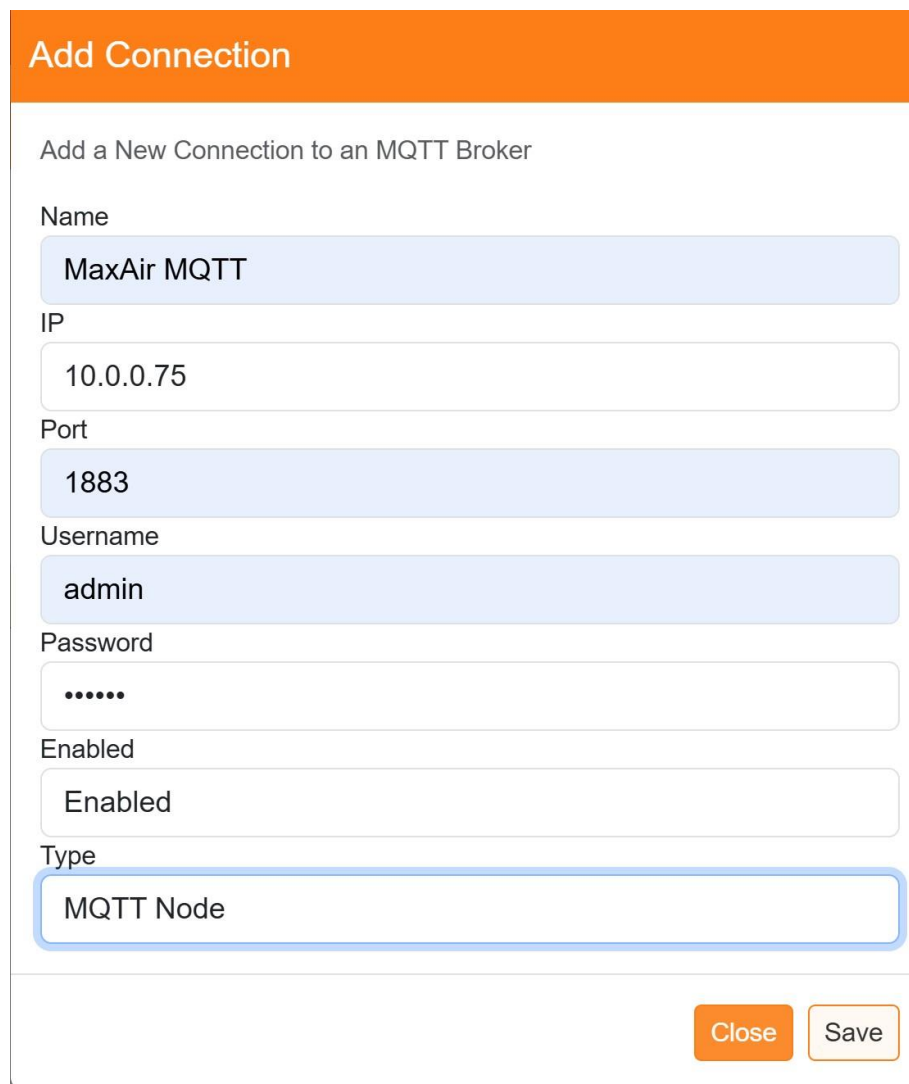
### Create an MQTT Connection

From Settings/System Configuration/MQTT select 'Add'



The image shows a dialog box titled "MQTT Connections" with an orange header. Inside, there is a large empty text input field. To the right of this field is a blue button labeled "Add". Below the input field, there is a blue button labeled "Close".

The example shows is using the Mosquitto Broker IP address of 10.0.0.75, with a default Port number of 1883, the Username and Password were as setup when configuring the broker, the connection is Enabled and the Type is selected as 'MQTT Node'.



The image shows a form titled "Add Connection" with an orange header. Below the header, the text "Add a New Connection to an MQTT Broker" is displayed. The form contains several input fields: "Name" with the value "MaxAir MQTT", "IP" with the value "10.0.0.75", "Port" with the value "1883", "Username" with the value "admin", "Password" with masked characters ".....", "Enabled" with the value "Enabled", and "Type" with the value "MQTT Node". At the bottom right of the form, there are two buttons: "Close" (orange) and "Save" (light orange).

### Create MQTT Type Nodes for MQTT Sensor

From Settings/Node and Zone Configuration/Nodes Add Node. For the example case a Node ID of 31 has been chosen and the Node Name selected as 'MQTT Sensor'.

Add Node

You can Add GPIO, I2C relay board as Node, Wireless Nodes are automatically discovered.

**Node Type** Node you want to make available for Zone and Boiler controller

MQTT

**Node ID** I2C board ID or 0 if you want to use Raspberry Pi GPIO

31

**Node Name** Identification for the MQTT Device

MQTT Sensor

Close

Save

Click on 'Save' to store the new node in the nodes table.

### Create MQTT Devices

From Settings/Node and Zone Configuration/MQTT Devices select Add MQTT Device

MQTT Devices

List of MQTT Devices Attached to Node Types.

Node	Child ID	Child Name	MQTT Topic	ON Payload	OFF Payload	JSON Attribute	Notice Interval	Min Level	
------	----------	------------	------------	------------	-------------	----------------	-----------------	-----------	--

Close

Add MQTT Device

The example shows that the Node Type has been selected as 'MQTT Sensor', its Device Name is 'Zigbee1', its Child ID has been set as 1, the MQTT Topic has been set as 'zigbee2mqtt/0xa4c138baf42763b1' (the first part of the topic before the / will always be zigbee2mqtt, the second prt wil be the 'Friendly name' shown in the 'Frontend' screen above) and the JSON Attribute is set to 'temperature. The 'Notice Interval' can be set so that the Home Screen will display warnings if the sensor does not report within the set period. The 'Minimum Value' can be set for battery powered sensors, so that a notification is generated if the value falls below the setting.

Edit MQTT Device: Zigbee113:06

Node Type MQTT Controller or MQTT Sensor

MQTT Sensor

MQTT Device Name Identification for the MQTT Device

Zigbee1

Child ID Node Child ID for This MQTT Device

1

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

zigbee2mqtt/0xa4c138baf42763b1

JSON Attribute Leave blank if the Sensor sends raw data to the topic

temperature

Notice Interval Set Notice Interval in minutes.

0

Set the Minimum Threshold Value If the Sensor Reading Falls Below This Value, then a Notification Will be Raised.

0

Submit

Cancel

Outside: 11° C Clouds - overcast clouds

Finally add a Sensor device to MaxAir using the GUI menus under Settings/Device Configuration/Sensors/, using the Node IDs and Child IDs configured above, eg.

Device Configuration

Relays

Test Relays

Sensors

Sensor Msg

Sensor Type

Reboot PI

Shutdown PI

Outside: 11° C Clouds - overcast clouds

Sensor Settings

Edit or Delete the Sensor's Configuration.

Sensors Allocated to a Zone Cannot be Deleted.

Last Seen Date/Time is shown with Sensor Name.

Sensor Name	Node ID	Sensor Child ID	Mode	Correct Factor	Res +/-	Zone Name	Show	DB Msg	
ZigBee Test (2025-10-16 12:44:16)	31	1		0	0.0	Not Allocated			

Close

Save

Add Sensor

The MQTT devices should now be active. Correct operation can be verified by running the Gateway Script in console mode, from the command line enter

```
'pkill -f gateway.py && python3 /var/www/cron/gateway.py'
```

Monitor the output for a few minutes

A connection to the MQTT Broker was established together with a subscription to the MQTT Sensor topic. After a few minutes data was received from the MQTT Sensor device

```

  M A X A I F
  SMART THERMOSTAT
*****
* MySensors Wifi/Ethernet/Serial Gateway Communication *
* Script to communicate with MySensors Nodes, for more *
* info please check MySensors API.                      *
*   Build Date: 18/09/2017                               *
*   Version 0.32 - Last Modified 16/10/2025              *
*                                     Have Fun - PiHome.eu *
*****
Gateway Type: Virtual
Setting up MQTT

Connected to broker
Subscribed to the followint MQTT topics:
zigbee2mqtt/0xa4c138baf42763b1

MQTT messaged received.
Topic: zigbee2mqtt/0xa4c138baf42763b1
Message: {"battery":100,"humidity":45.67,"linkquality":224,"temperature":23.31,"voltage":3000}
5: Adding temperature Reading From Node ID: 31 Child Sensor ID: 1 Payload: 23.31
5b: MQTT Sensor Processed on Node ID: 31 Child Sensor ID: 1
```

An entry for the node will be added to the messages\_in table, which will contain the returned temperature.

←T→		id	sync	purge	node_id	child_id	sub_type	payload	datetime		
				Mark For Deletion							
<input type="checkbox"/>	Edit	Copy	Delete	63	0	0	31	1	0	23.82	2025-10-15 21:50:05