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/Humidy 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

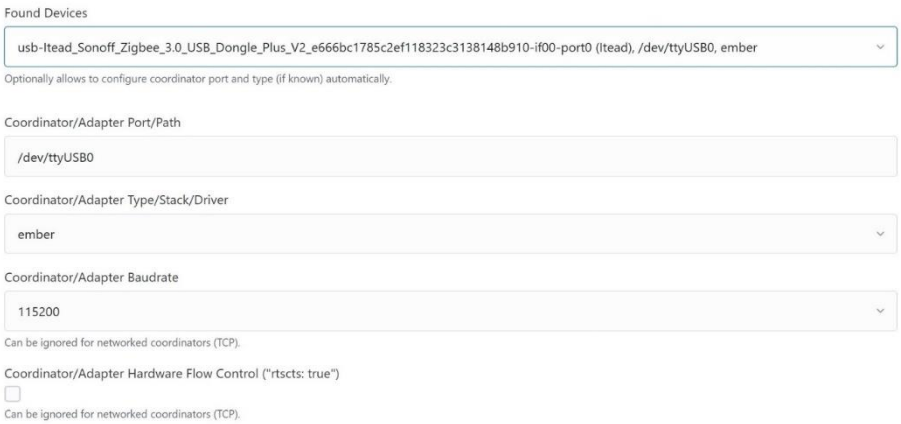
- The Mosquitto Broker can be installed and configured from 'Settings/System Maintenance/Install Software' menu option and select 'Install' for 'Mosquitto Broker'.
- If not already available then install paho-mqtt using the command '***pip3 install paho-mqtt***'.
- After the install has completed check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

Installing Zigbee2MQTT

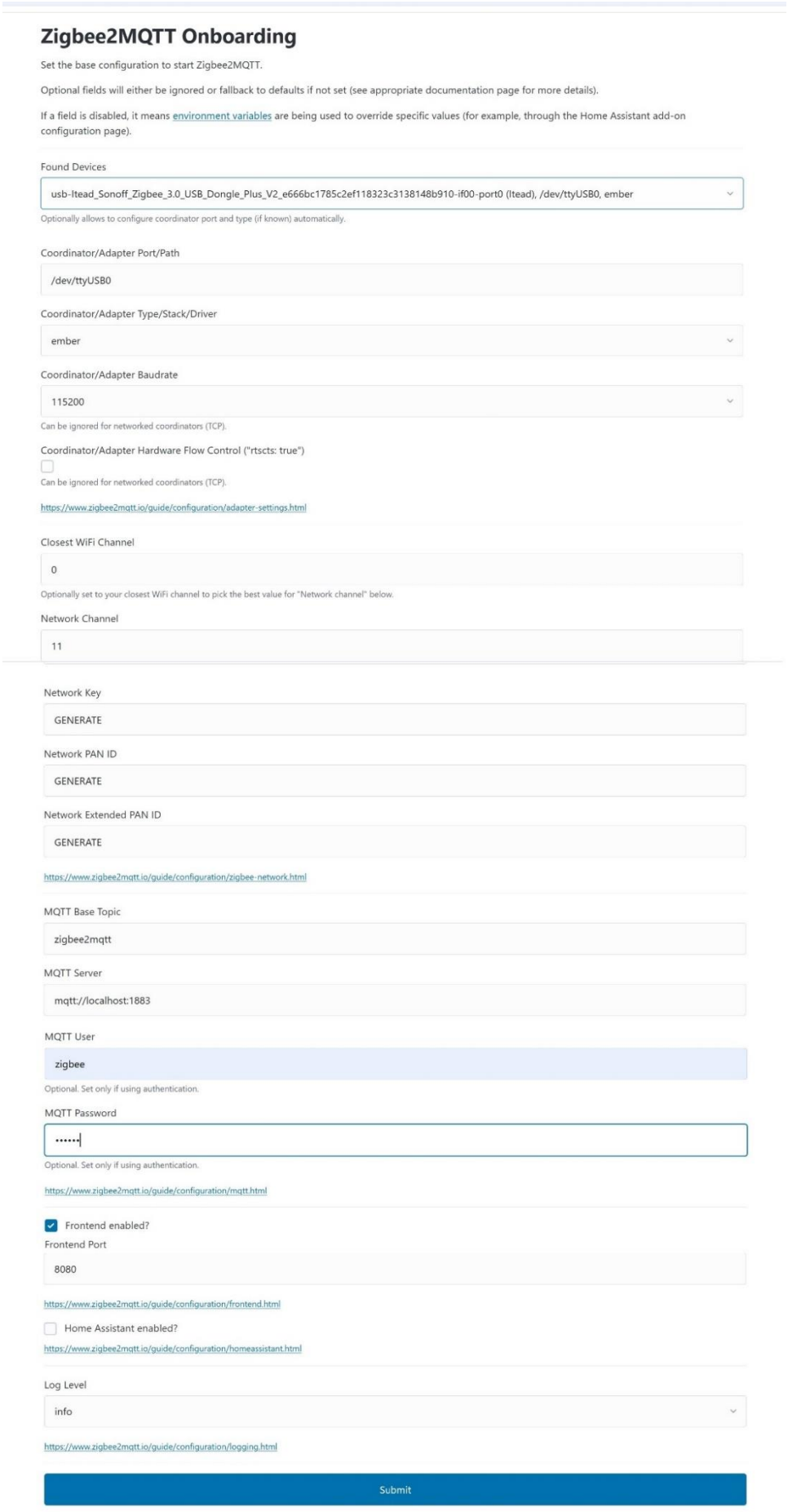
- Zigbee2MQTT can be installed from 'Settings/System Maintenance/Install Software' menu option and select 'Install' for 'Zigbee2MQTT Integration'.

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 Adress 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



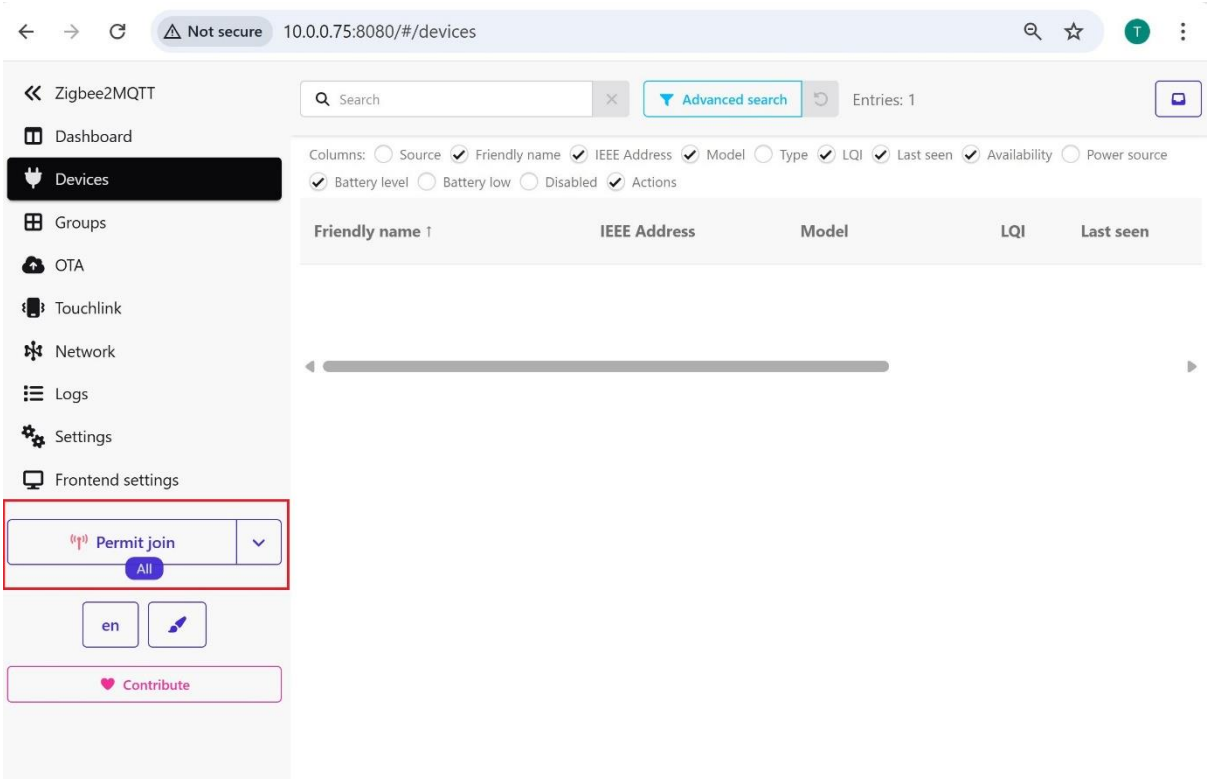
- Enter the Username and Password for the Mosquitto Broker (zigbee/pihome) and check the 'Frontend enabled box.



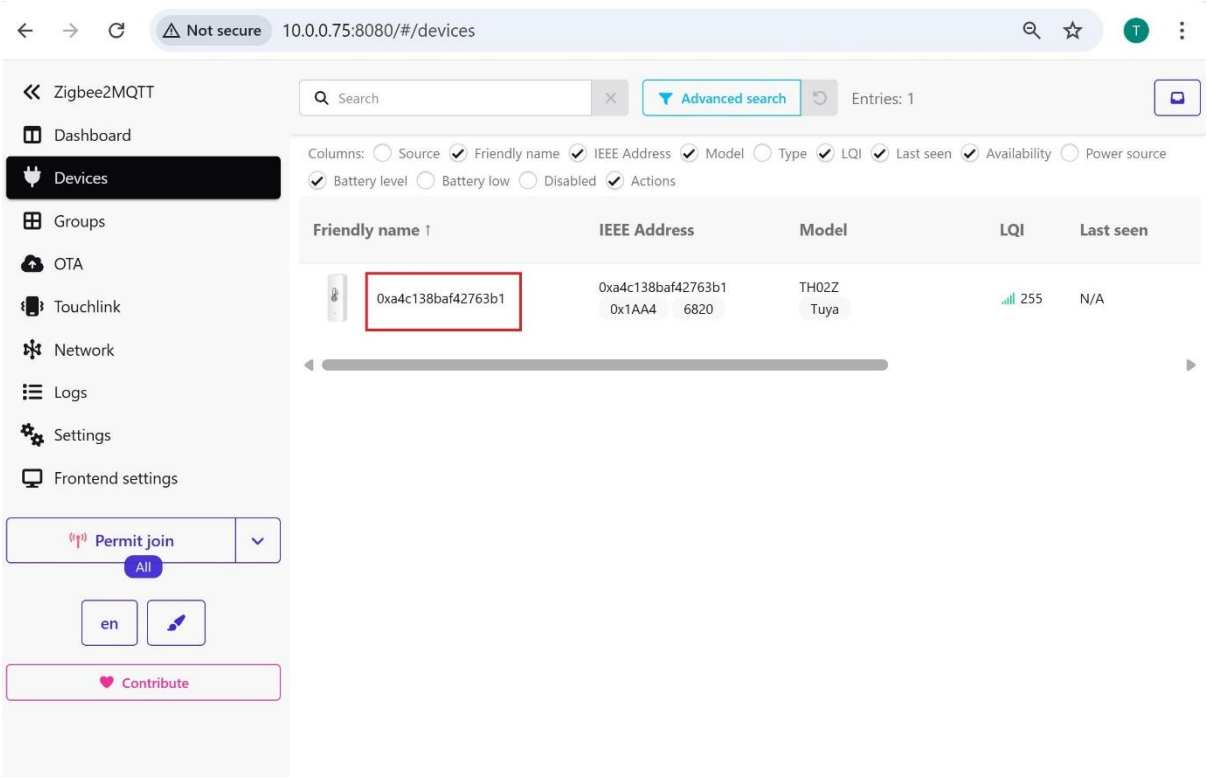
Once the configuration has been saved, the zigbee2mqtt service will move in to the running state.

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.

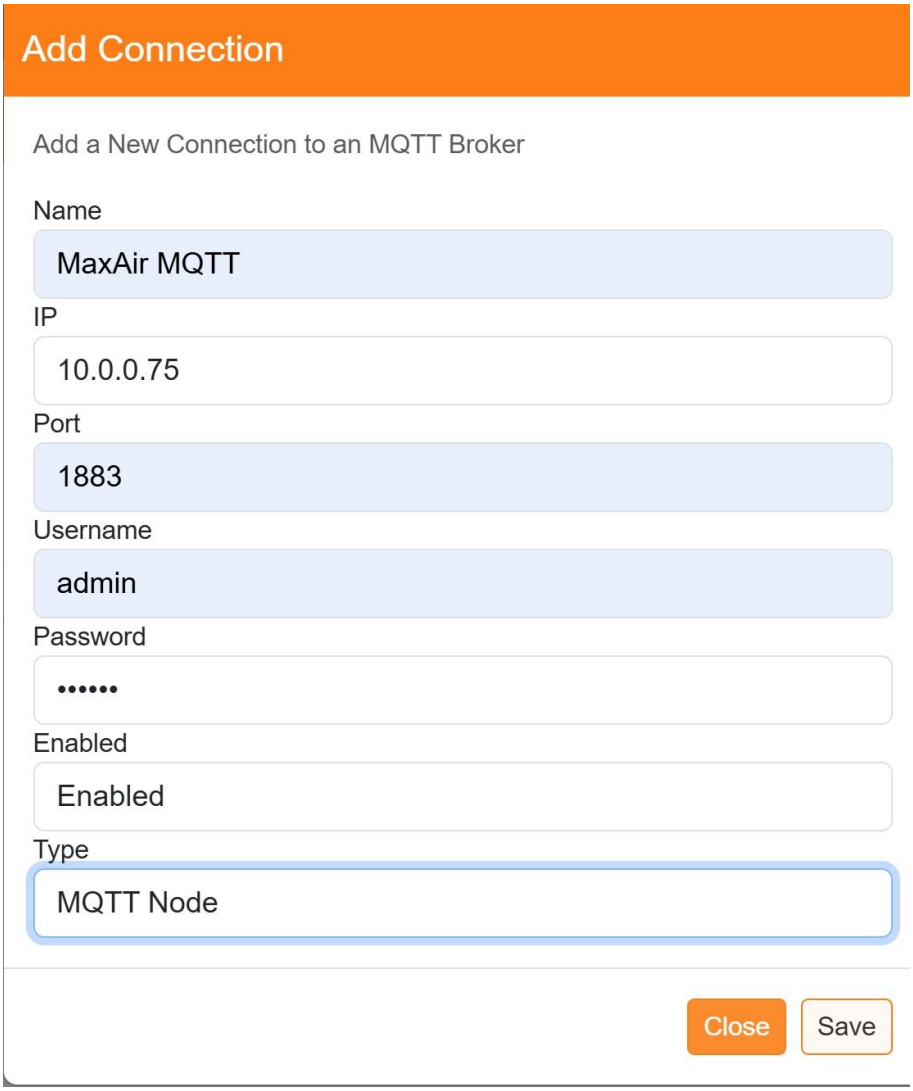


Configure MaxAir to Communicate Using MQTT

Create an MQTT Connection

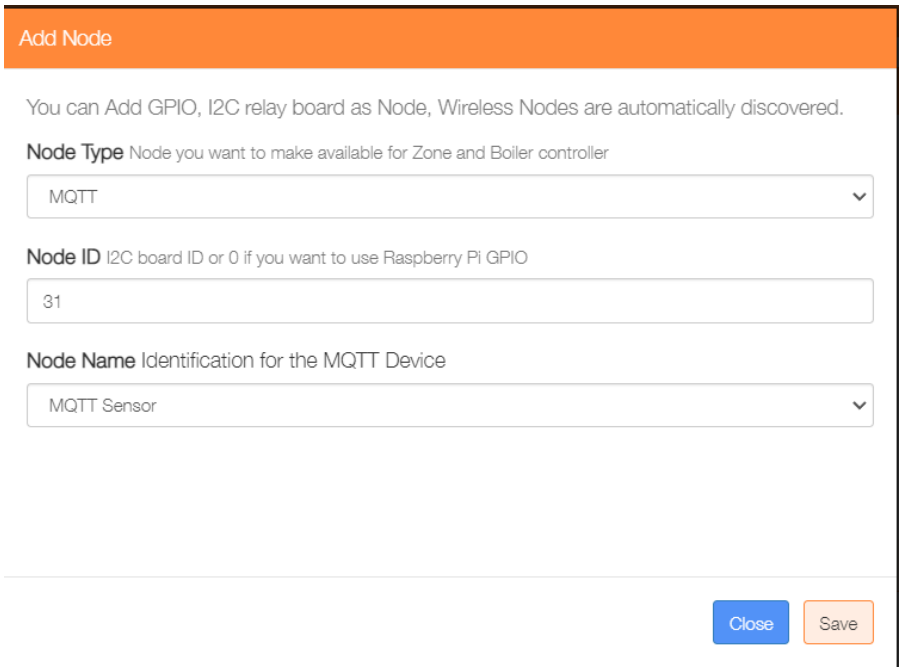
From Settings/System Configuration/MQTT select ‘Add’

The 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’.



Create MQTT Type Nodes for MQTT Sensor

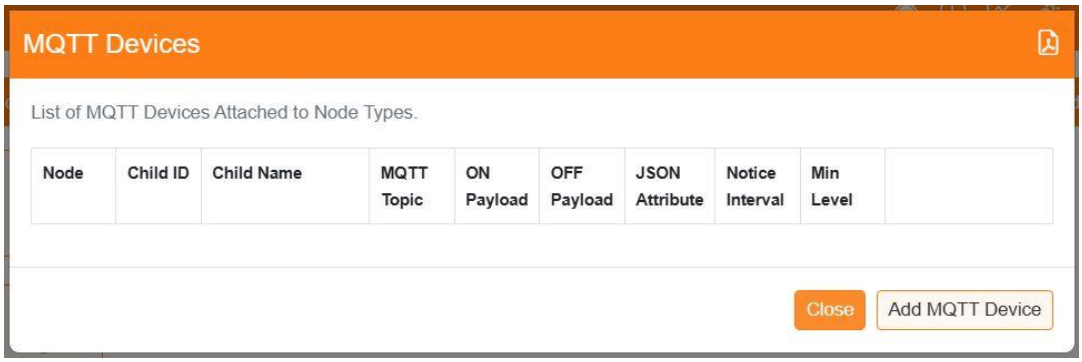
From Settings/Node and Zone Configuration/Nodes Add Node. For the example case a Node ID of 31 has been chosen and the Node Name selected as 'MQTT Sensor'.



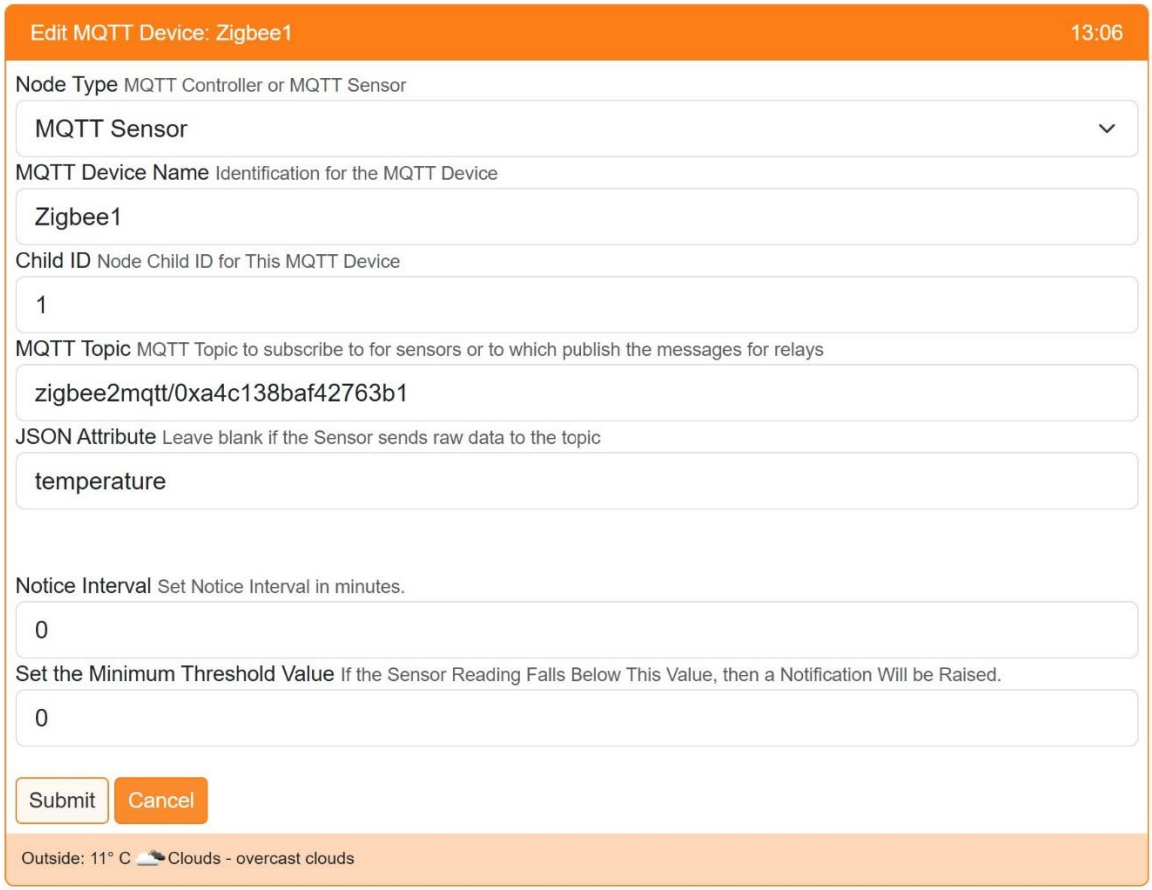
Click on 'Save' to store the new node in the nodes table.

Create MQTT Devices

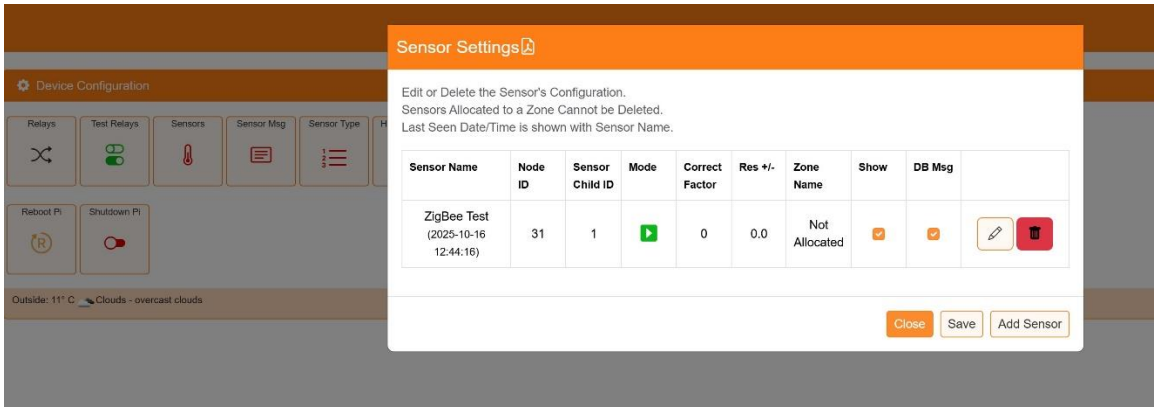
From Settings/Node and Zone Configuration/MQTT Devices select Add MQTT Device



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.



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.

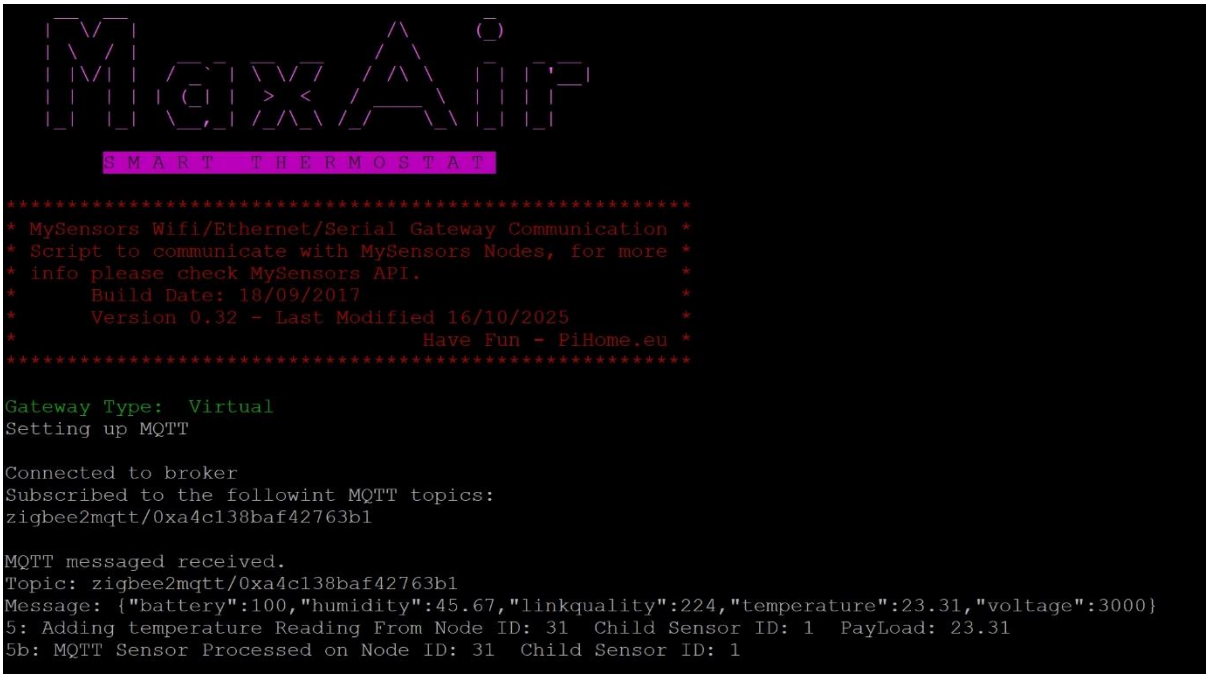


The MQTT devices should now be active. Correct operation can be verified by running the Gateway Script in console mode, from the command line enter

‘**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



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

←T→	id	sync	purge Mark For Deletion	node_id	child_id	sub_type	payload	datetime
Edit Copy Delete	63	0	0	31	1	0	23.82	2025-10-15 21:50:05