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 Brocker which it can access.

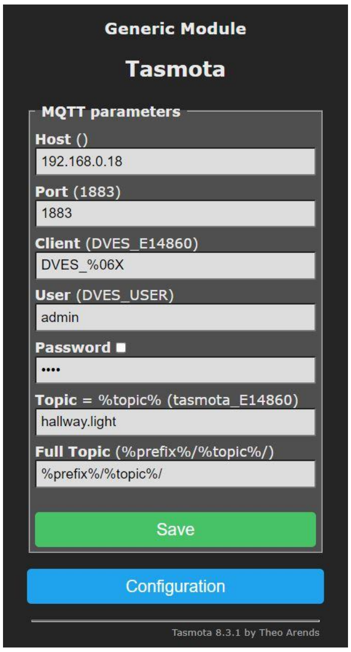
Example Configuration

- 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
- From the linux command line execute '***nano /etc/mosquitto/mosquitto.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***'

Configure the Tasmota DS18B20 D1 Mini Sensor



Generic Module

Tasmota

MQTT parameters

Host ()
192.168.0.18

Port (1883)
1883

Client (DVES_E14860)
DVES_%06X

User (DVES_USER)
admin

Password ■
••••

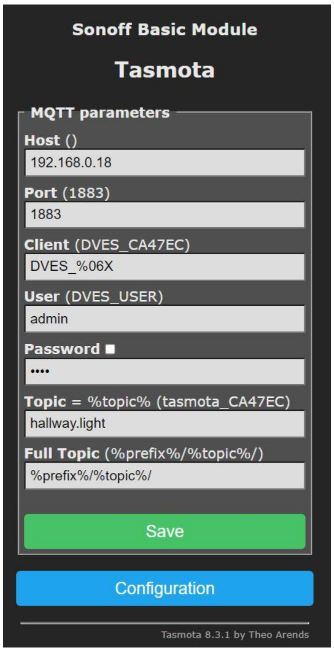
Topic = %topic% (tasmota_E14860)
hallway.light

Full Topic (%prefix%/%topic%/)
%prefix%/%topic%/

Connect to the Tasmota device using its IP address and configure the MQTT parameters. For this example the MQTT Broker has an IP address of 192.168.0.18 and the Topic, Full Topic settings are as shown.

At this point the Tasmota DS18B20 should be sending temperature data to the Mosquitto Broker.

Configure the Sonoff Basic Module as a Switch Device



Sonoff Basic Module

Tasmota

MQTT parameters

Host ()
192.168.0.18

Port (1883)
1883

Client (DVES_CA47EC)
DVES_%06X

User (DVES_USER)
admin

Password ■
••••

Topic = %topic% (tasmota_CA47EC)
hallway.light

Full Topic (%prefix%/%topic%/)
%prefix%/%topic%/

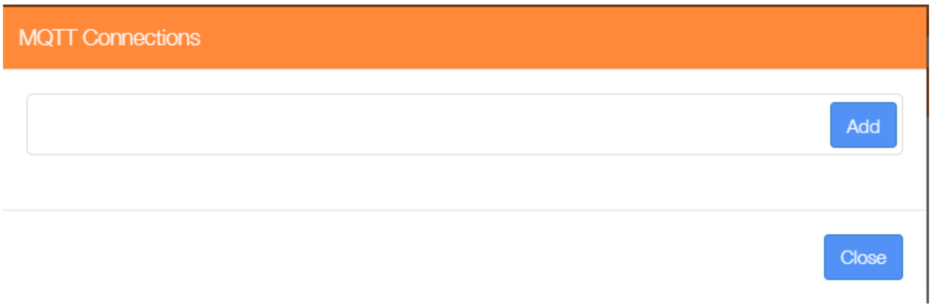
Connect to the Sonoff device using its IP address and configure the MQTT parameters. For this example the MQTT Broker has an IP address of 192.168.0.18 and the Topic, Full Topic settings are as shown (and are exactly the same as for the previous Sensor example).

At this point it is possible for the Sonoff switch state to be set using an MQTT message.

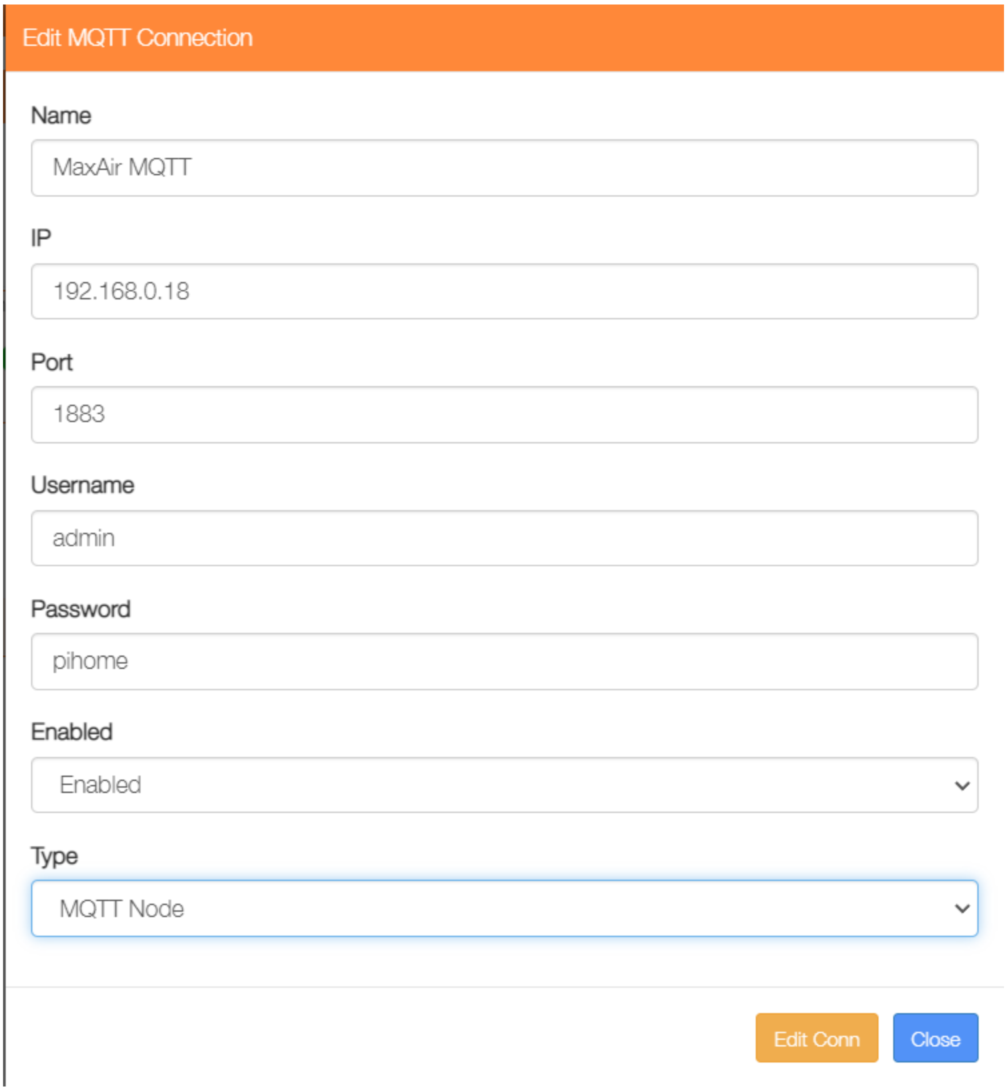
Configure MaxAir to Communicate Using MQTT

Create an MQTT Connection

From Settings/System Configuration/MQTT select 'Add'



The example shows is using the Mosquitto Broker IP address of 192.168.0.18, 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'.



Name: MaxAir MQTT

IP: 192.168.0.18

Port: 1883

Username: admin

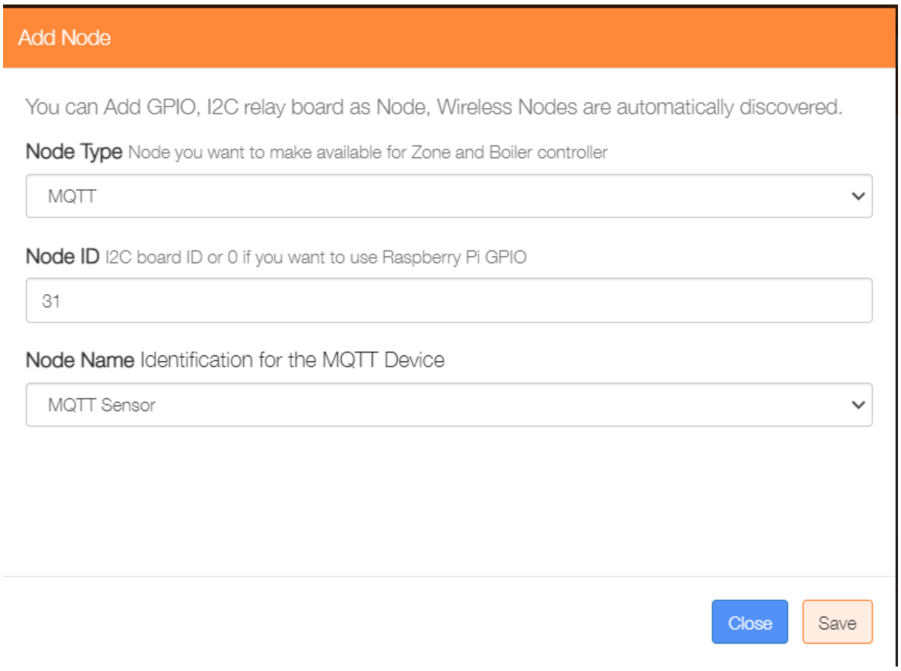
Password: pihome

Enabled: Enabled

Type: MQTT Node

Create MQTT Type Nodes for Both a Sensor and a Controller

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.

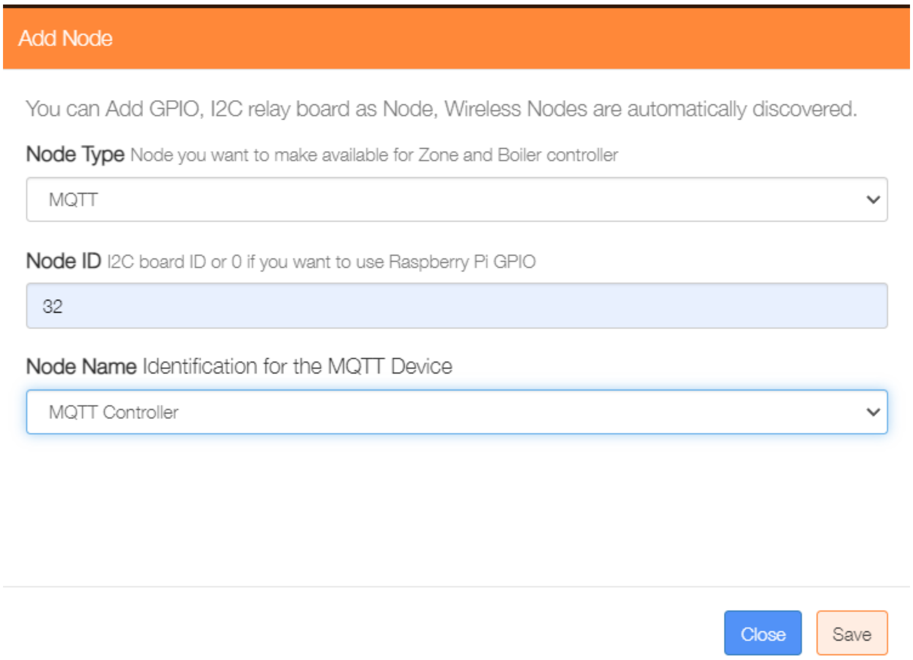
Node Type Node you want to make available for Zone and Boiler controller: MQTT

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

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

Add a second node for the MQTT Controller device, for this example a Node ID of 32 is used.



Add Node

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

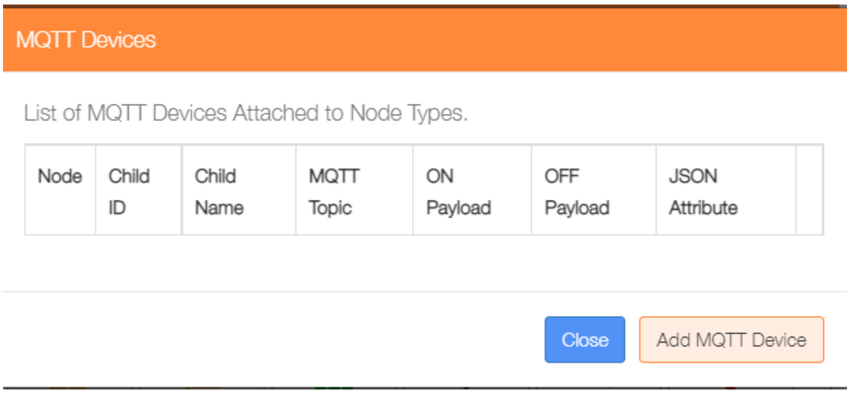
Node Type Node you want to make available for Zone and Boiler controller: MQTT

Node ID I2C board ID or 0 if you want to use Raspberry Pi GPIO: 32

Node Name Identification for the MQTT Device: MQTT Controller

Create MQTT Devices

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

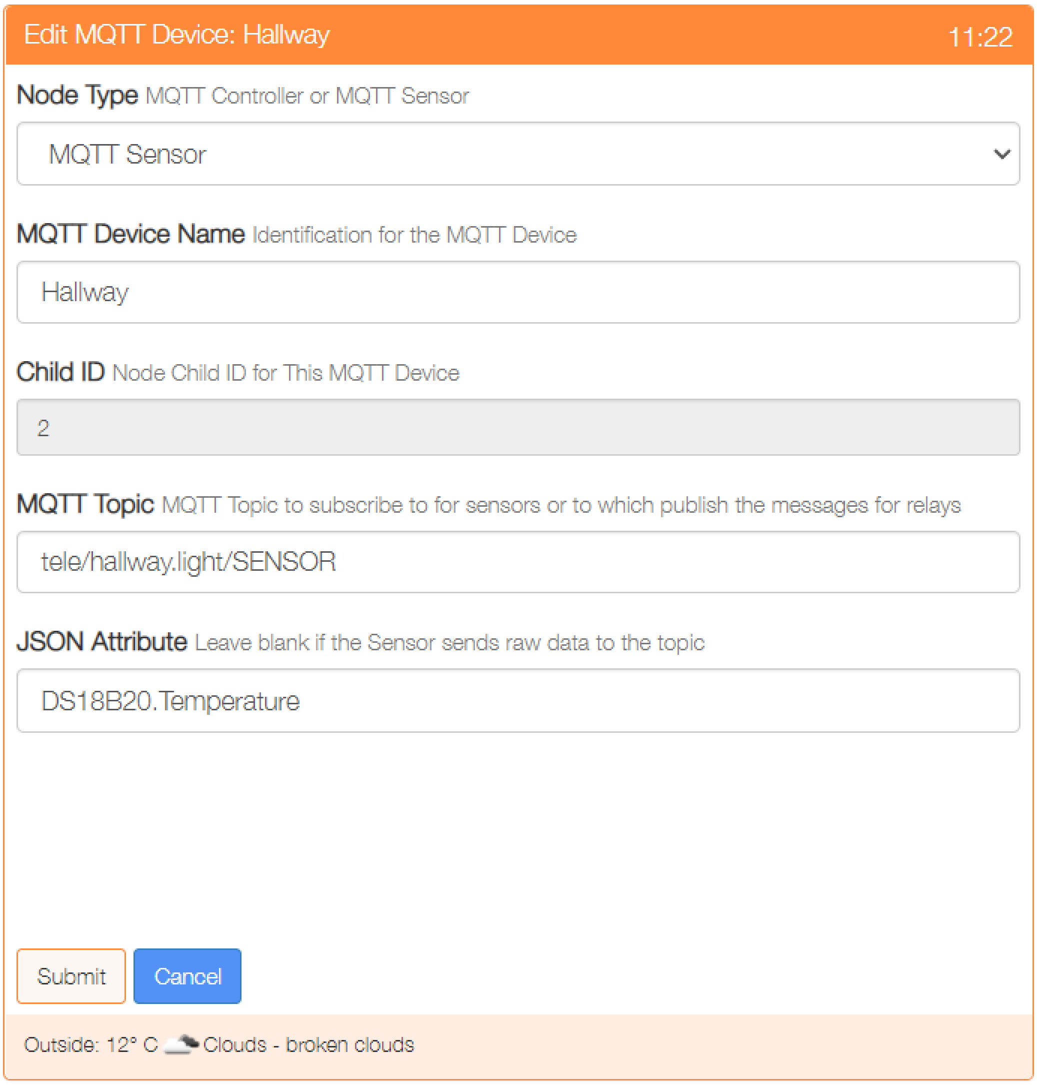


MQTT Devices

List of MQTT Devices Attached to Node Types.

Node	Child ID	Child Name	MQTT Topic	ON Payload	OFF Payload	JSON Attribute	

The example shows that the Node Type has been selected as ‘MQTT Sensor’, its Device Name is ‘Hallway’, its Child ID has been set as 2, the MQTT Topic has been set as ‘tele/hallway.light/SENSOR’ and the JSON Attribute is set to ‘DS18B20.Temperature.



Edit MQTT Device: Hallway 11:22

Node Type MQTT Controller or MQTT Sensor

MQTT Sensor

MQTT Device Name Identification for the MQTT Device

Hallway

Child ID Node Child ID for This MQTT Device

2

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

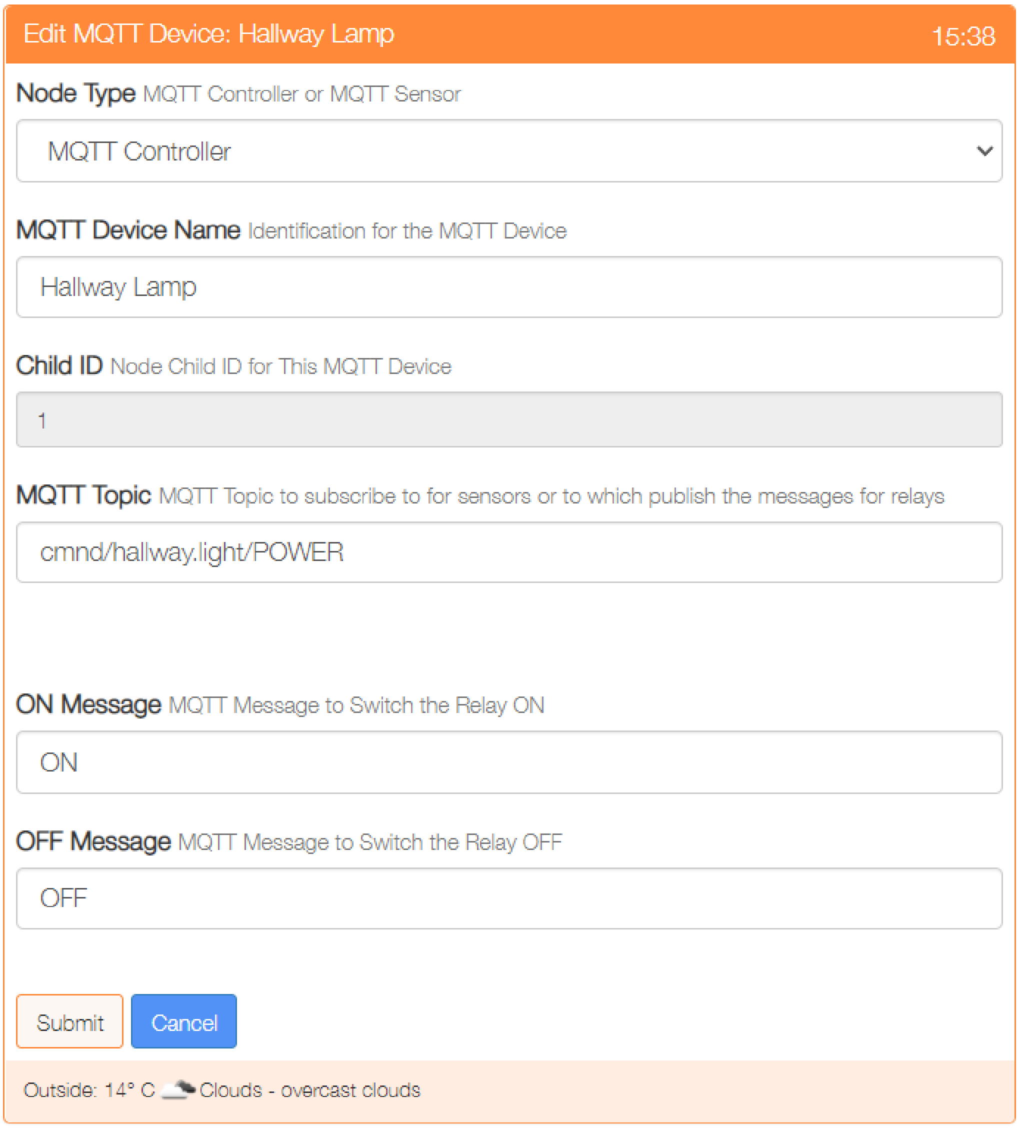
tele/hallway.light/SENSOR

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

DS18B20.Temperature

Outside: 12° C Clouds - broken clouds

Add a second MQTT device for the controller. The example shows that the Node Type has been selected as 'MQTT Controller', its Device Name is 'Hallway Lamp', its Child ID has been set as 1, the MQTT Topic has been set as 'cmnd/hallway.light/POWER', the ON Message is set as 'ON' and the OFF Message is set as 'OFF'.



Edit MQTT Device: Hallway Lamp 15:38

Node Type MQTT Controller or MQTT Sensor

MQTT Controller

MQTT Device Name Identification for the MQTT Device

Hallway Lamp

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

cmnd/hallway.light/POWER

ON Message MQTT Message to Switch the Relay ON

ON

OFF Message MQTT Message to Switch the Relay OFF

OFF

Outside: 14° C Clouds - overcast clouds

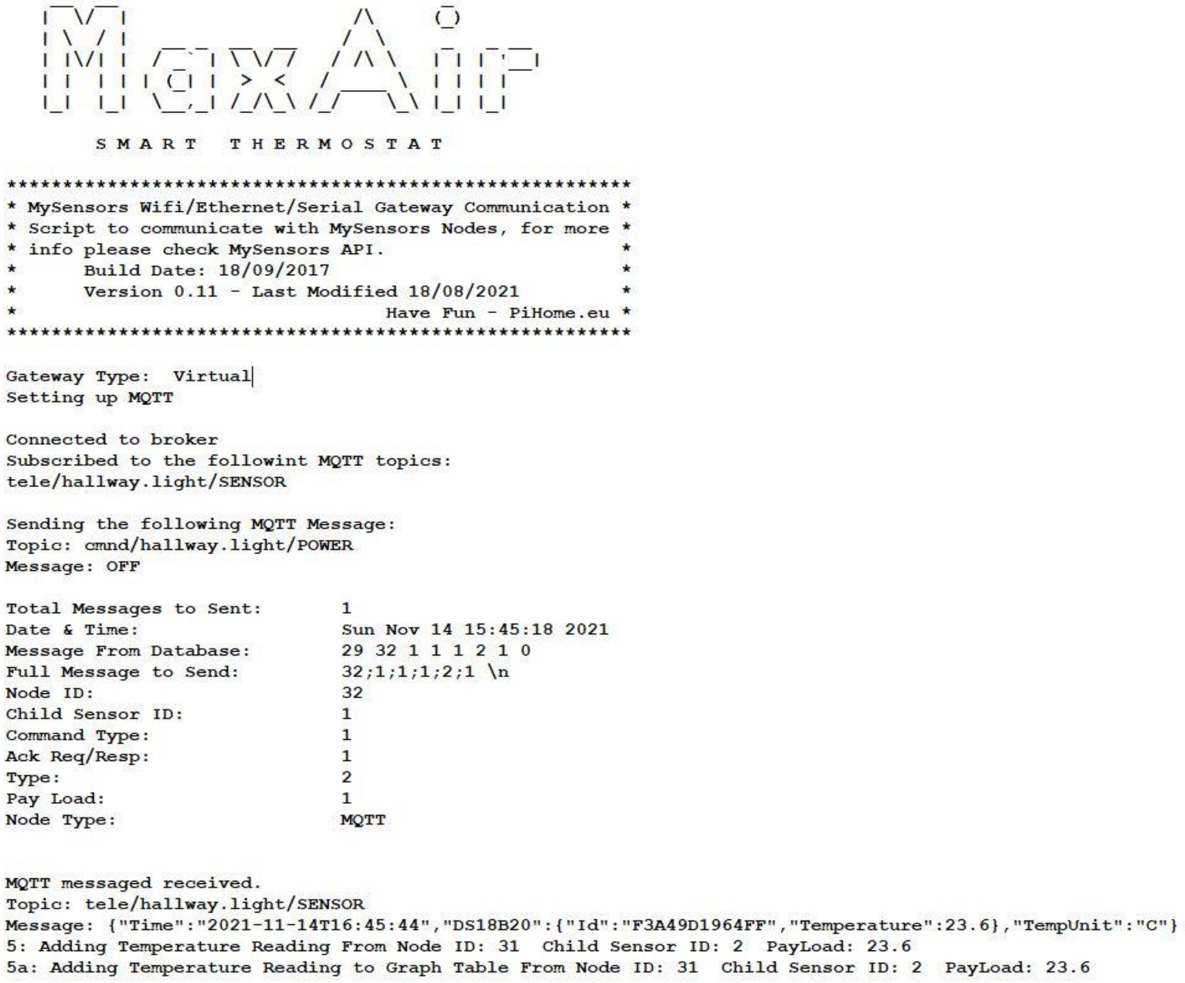
Finally add a Sensor and Relay device to MaxAir using the GUI menus under Settings/Node and Zone Configuration, using the Node IDs and Child IDs configured above.

In order to use the MQTT Controller device it will need to be added to a Zone, so that an entry is added the the messages_out queue.

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**’

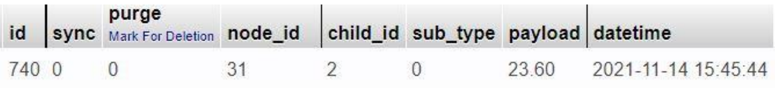
Monitor the output for a few minutes



```
 __  __                                 _
|  \/  |                  /\           (_)
| \  / |  __ _ __  __    /  \     _  _ __
| |\/| | / _` |\ \/ /   / /\ \   | || '__|
| |  | || (_| | >  <   / ____ \  | || |
|_|  |_| \__,_|/_/\_\ /_/    \_\ |_||_|

      S M A R T   T H E R M O S T A T

*********************************************************
* 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 0.11 - Last Modified 18/08/2021          *
*                                 Have Fun - PiHome.eu *
*********************************************************

Gateway Type:  Virtual
Setting up MQTT

Connected to broker
Subscribed to the followint MQTT topics:
tele/hallway.light/SENSOR

Sending the following MQTT Message:
Topic: cmnd/hallway.light/POWER
Message: OFF

Total Messages to Sent:        1
Date & Time:                   Sun Nov 14 15:45:18 2021
Message From Database:         29 32 1 1 1 2 1 0
Full Message to Send:          32;1;1;1;2;1 \n
Node ID:                       32
Child Sensor ID:               1
Command Type:                  1
Ack Req/Resp:                  1
Type:                          2
Pay Load:                      1
Node Type:                     MQTT


MQTT messaged received.
Topic: tele/hallway.light/SENSOR
Message: {"Time":"2021-11-14T16:45:44","DS18B20":{"Id":"F3A49D1964FF","Temperature":23.6},"TempUnit":"C"}
5: Adding Temperature Reading From Node ID: 31  Child Sensor ID: 2  PayLoad: 23.6
5a: Adding Temperature Reading to Graph Table From Node ID: 31  Child Sensor ID: 2  PayLoad: 23.6
```

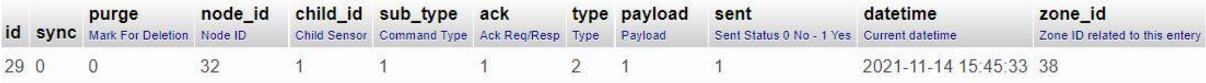
A connection to the MQTT Broker was established together with a subscription to the MQTT Sensor topic. Data was sent to the MQTT Controller device to set its initial state, and after a few minutes data was received from the MQTT Sensor device

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



id	sync	purge Mark For Deletion	node_id	child_id	sub_type	payload	datetime
740	0	0	31	2	0	23.60	2021-11-14 15:45:44

The messages_out table will have an entry for the Node ID allocated to the MQTT Controller.



id	sync	purge Mark For Deletion	node_id Node ID	child_id Child Sensor	sub_type Command Type	ack Ack Req/Resp	type Type	payload Payload	sent Sent Status 0 No - 1 Yes	datetime Current datetime	zone_id Zone ID related to this entery
29	0	0	32	1	1	1	2	1	1	2021-11-14 15:45:33	38