

Integration with a ZigBee protocol

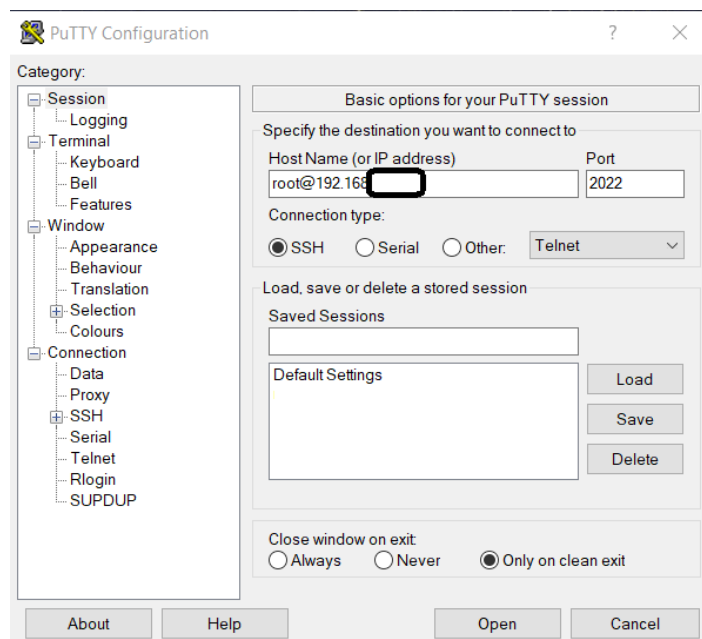
Document number: PO-197-EN Version: 1.0 Date of publication: March 7, 2023

Introduction

Integration of devices that support the ZigBee protocol with the Ampio system is possible, for example, by connecting a gateway to the M-SERV-s module. Using Node-RED is a prerequisite to establishing a connection. This guide presents the example of ZBDongle-E from Sonoff with a dedicated antenna as a gateway.

Gateway installation

In order to connect the gateway, disconnect M-SERV-s from its power supply, connect the gateway device and power up the server again. After a couple of minutes, activate an SSH connection via a www interface (more guidelines available in [M-SERV server configuration](#)). Log in onto the *root* account on the server with the password you have created e.g. through the *putty* application.



Searching for the gateway port

After entering the password, search for devices with the following command:

```
dmesg | grep tty
```

The gateway will most probably be added as *ttyACM0*.

Creating and configuring a folder

Create a folder:

```
sudo mkdir /ampio/rw/zigbee2mqtt
```

Grant permissions:

```
sudo chown -R ${USER}: /ampio/rw/zigbee2mqtt
```

Cloning of the zigbee2mqtt repository

```
git clone --depth 1 https://github.com/Koenkk/zigbee2mqtt.git /ampio/rw/zigbee2mqtt
```

Installing content

Change the current folder:

```
cd /ampio/rw/zigbee2mqtt
```

and install:

```
npm ci
```

Modification of the configuration file

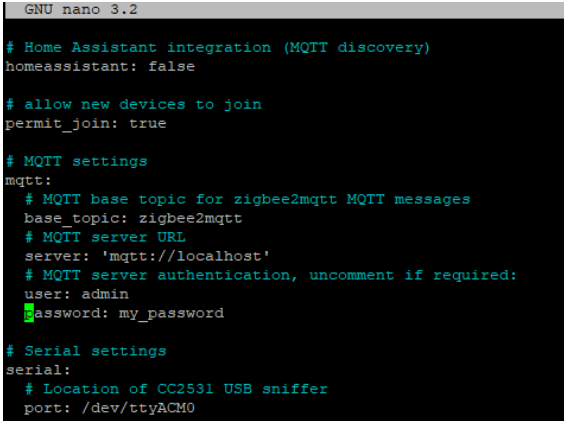
Open the file to be edited

```
nano /ampio/rw/zigbee2mqtt/data/configuration.yaml
```

Set the *server* field to *mqtt://localhost*.

The *port* field should be completed with information obtained in the previous step e.g., */dev/ttyACM0*.

MQTT connection field should be filled with your server settings, *user* set to *admin* and the correct password should be provided (login details are the same as for the *mqtt* blocks in Node-RED).



```
GNU nano 3.2
# Home Assistant integration (MQTT discovery)
homeassistant: false

# allow new devices to join
permit_join: true

# MQTT settings
mqtt:
  # MQTT base topic for zigbee2mqtt MQTT messages
  base_topic: zigbee2mqtt
  # MQTT server URL
  server: 'mqtt://localhost'
  # MQTT server authentication, uncomment if required:
  user: admin
  password: my_password

# Serial settings
serial:
  # Location of CC2531 USB sniffer
  port: /dev/ttyACM0
```

Once the changes are introduced, close the configuration file. You can do that in putty by clicking *Ctrl+x*, then *y* and *Enter* at the end.

First launch

Enter the following command:

```
npm start
```

Adding a secondary device

A device added as an example in this guide is a temperature and humidity sensor SNZB-02 from Sonoff. Manuals for specific devices are available on websites of particular manufacturers. In the case of the sensor described here, in order to have it added, you need to press and hold in the button on the casing for 5 seconds. After a successful addition of the device, you should see a couple of messages on the terminal, including for example:

```
zigbee2mqtt:info 2023-02-28 08:58:06: Successfully interviewed '0x00124b00250e039e', device has successfully been paired
```

After adding the sensor, it is possible to respond to data or send information to the CAN bus using certain nodes in Node-RED (e.g. *mqtt out*).

For the purpose of upholding security, after adding all required devices, the process must be stopped. One way to do it is by clicking *Ctrl+c*, opening the *configuration.yaml* file and setting *permit_join:false*. Once that is done, the process can be started again.

Configuration in Node-RED

A guide that describes the basics of Node-RED use in the Ampio system is available here: [Integration of the Ampio system with Node-RED](#). After adding secondary devices, data from the MQTT Ampio can be received. Topic on which the device is broadcasting can be viewed in a terminal via an SSH connection. In this case, it is the main topic and ID of the added device: *zigbee2mqtt/0x00124b00250e039e*.

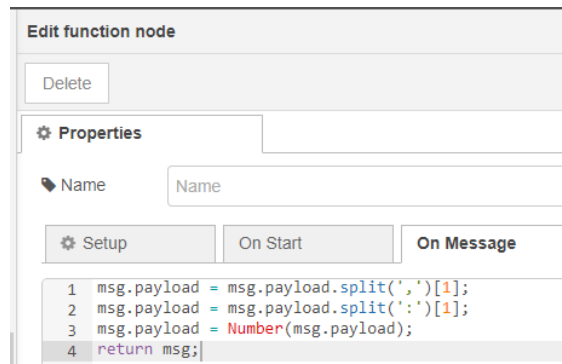
The screenshot shows the 'Edit mqtt in node' configuration window in Node-RED. It includes a 'Delete' button on the left and 'Cancel' and 'Done' buttons on the right. The 'Properties' section is expanded, showing the following settings:

- Server: localhost:1883
- Action: Subscribe to single topic
- Topic: zigbee2mqtt/0x00124b00250e039e
- QoS: 2
- Output: auto-detect (string or buffer)
- Name: Name

Data can be viewed after adding a *debug* node.

```
28.02.2023, 09:05:19 node: 3604926b8f956599
zigbee2mqtt/0x00124b00250e039e : msg.payload :
string[85]
"
{"battery":100,"humidity":33.31,"link
quality":220,"temperature":22.97,"vol
tage":3300}"
```

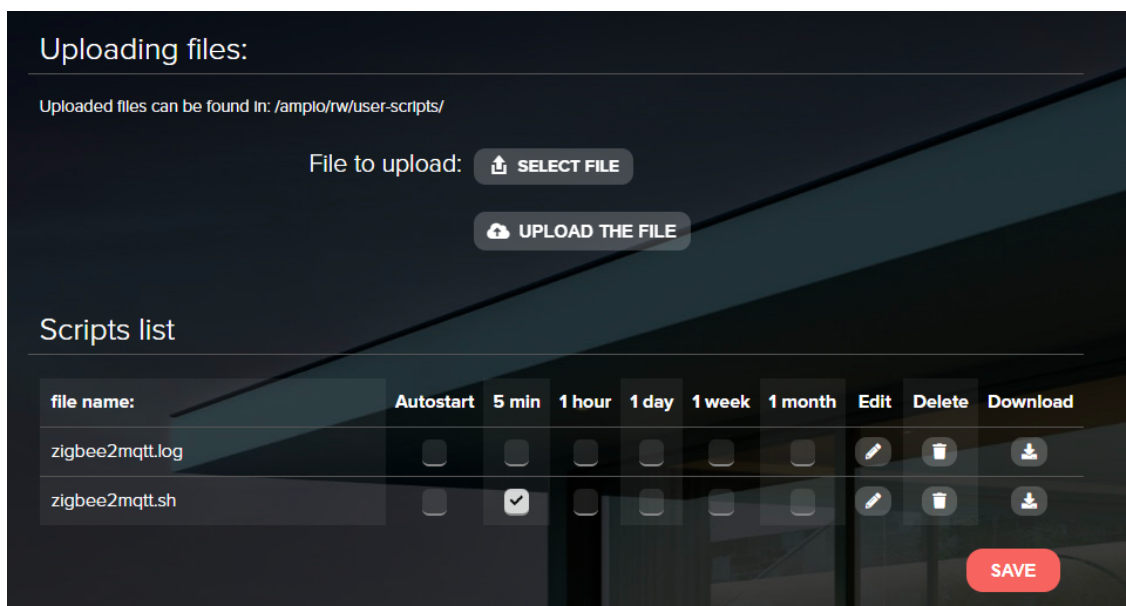
If you want to for example, read the value of humidity from a sensor, you will need to send the following information through the node *function*:



Different end devices can broadcast information in different ways, which is why it is worthwhile to view the data first in the *debug* window, before creating a function to send the information.

Automatic launching

For the application to launch automatically after a power supply reboot, a suitable script must be written. Open the *SYSTEM* tab in the *www* interface of *M-SERV* and go to *SCRIPTS*. Download the attached file *zigbee2mqtt.sh*. Upload it by clicking on *SELECT FILE*, and then *UPLOAD THE FILE*. Then, tick the *5 min* box next to the correct script and click *SAVE*.



In order to confirm the correctness of configuration, reboot the server's power supply and, after a couple of minutes, check whether everything is working as intended in *Node-RED*, in the *debug* window, for instance.

Download files:

- [zigbee2mqtt.sh](#)