COMPUTER ENGINEERING PROJECT II

TUTORIAL 3: Setting up Zigbee2Mqtt

Last revision: February 5th, 2020

1 INTRODUCTION & MOTIVATION

Your project must be able to monitor sensors and control actuators. These include movement sensors (PIR) and control devices such as LED light strips and power plugs. In order to communicate with these devices, you need to choose a communication standard, and here we have chosen Zigbee for you. Zigbee is one of the most popular platforms for wireless sensor networks. This can be controlled in various ways, but one of them is with the Zigbee2Mqtt (Z2M) [1], which is an open-source application that interfaces Zigbee based networks for home automation and translates its events to MQTT.

In this tutorial you will learn how to setup Z2M in a Raspberry Pi and interact with it.

2 WHAT TO READ BEFORE THE TUTORIAL?

It is expected that you have a basic knowledge of the following topics:

- What is the Zibgee standard [2, 3].

3 MATERIALS

You'll need the following materials:

- Zigbee USB adapter
- Raspberry Pi 4
- HDMI to mini-HDMI cable
- Power supply
- Keyboard, mouse and monitor (any wired keyboard and mouse should work)

4 INSTALLING Z2M IN THE RASPBERRY PI

The installation of Z2M in Linux is made in two stages: 1) install the Zigbee USB adapter and 2) install Z2M. The second stage can be done before the second stage, but Z2M will not start, because the adapter is not installed. If you have any problem with stage 1, then refer to [1] and [4] for help.

4.1 Installing the Zigbee USB adapter

The installation of Z2M will be done using the terminal of the Raspberry Pi OS (RPiOS). Start by connecting the Zigbee USB adapter to Raspberry Pi and check if it is detected by executing the command `lsusb`. This command lists all USB devices connected to the Raspberry. One of the listed devices should be a "`Texas Instruments, Inc.`", as depicted in Figure 1.

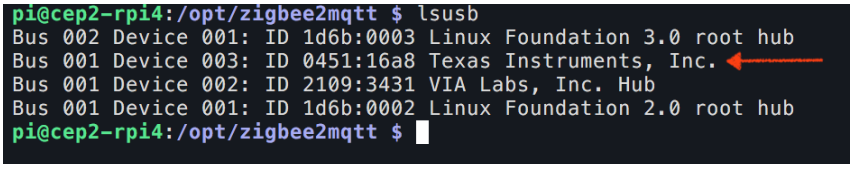

Figure 1. List of USB devices connected to the Raspberry Pi. Marked is the Zigbee USB adapter.

This device is detected as a serial device, so Linux will create a serial port where communication with the device will be performed. To know which serial port the device is, execute the command `ls -l /dev/serial/by-id`. A device with the prefix usb-`Texas_Instruments_TI_CC2531_USB_CDC` should be listed as depicted in Figure 2. This is a symbolic link (or shortcut) for the actual serial device. In the end of the line you can see that the device `ttyACM0` is listed. This is the serial port that refers to the USB adapter and, if you try to list it (`ls -l /dev/ttyACM0`), it should be listed in the `/dev` directory.

Figure 2. List of serial ports detected by RPiOS.

4.2 Installing and running Z2M

Z2M is a Node.js application. Therefore, Z2M installation will be done in two stages: 1) install Node.js and 2) actually install Z2M.

To install Node.js, execute the following commands (comments are prefixed with a '#' in grey):

```
# Setup Node.js repository in the packet manager
sudo curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash –
# Install Node.js and its dependencies
sudo apt-get install -y nodejs git make g++ gcc

# Verify that the correct nodejs and npm are installed
node --version  # Should output v12.X or v10.X
npm --version  # Should output 6.X
```

In Figure 3 it is depicted the output of the node and npm commands.

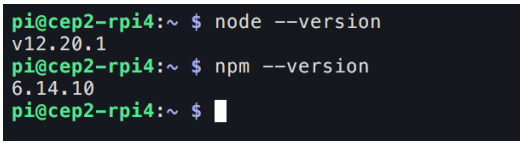

Figure 3. output of the node and npm commands.

If Node.js is successfully installed, then you can proceed to install Z2M using the following commands:

```
# Clone Z2M repository to the directory /opt/zigbee2mqtt. This is the directory where it will be installed.
sudo git clone https://github.com/Koenkk/zigbee2mqtt.git /opt/zigbee2mqtt
# Change ownership of the directory /opt/zigbee2mqtt to user pi (user and group).
sudo chown -R pi:pi /opt/zigbee2mqtt
# Install dependencies (as user "pi", which was changed in the previous step).
cd /opt/zigbee2mqtt
# Build the application using Node.js building tools.
npm ci --production
```

The output of the last command should be similar to what is depicted in Figure 4. This also produces a lot of warnings which can be ignored (they are related with building Node.js' `serialport` package).

Figure 4. Output of the command npm ci –production

Once Z2M is installed, you can run it. To do so, you must be in the directory where Z2M is installed (if you followed the given steps correctly, you should already be there). To start it, execute the command `npm start`. Figure 5 depicts the truncated output of Z2M once it is started.

Figure 5. Output of Z2M on start.

To stop the process (or kill it) you can press Ctrl+C. Figure 6 depicts Z2M's log output when it stops.

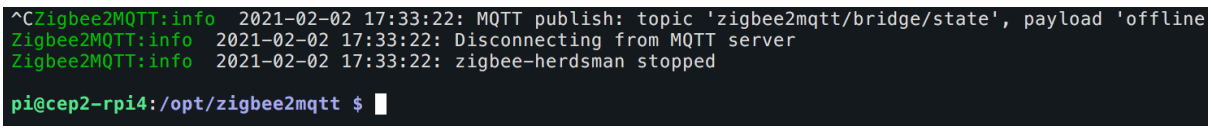

Figure 6. Output of Z2M when it is stopped.

4.3 Interact with Z2M

In the last tutorial you learned what MQTT is and how to use the Mosquitto clients to publish and subscribe to(from) other MQTT clients. Here you will use them to interact with Z2M, which is a publisher and subscriber.

The default Z2M's top level topic is `zigbee2mqtt`. You can use `mosquito_sub` to subscribe to all events with this topic to see the events published by Z2M. You can use the following command to subscribe[1]:

```
mosquitto_sub -h localhost -t "zigbee2mqtt/#" -F "[%t]   %p"
```

Note that the subscribed topic is `zigbee2mqtt/#`, which indicates that all topics with the top level `zigbee2mqtt` are subscribed. A sample (and truncated) output of running the subscriber command above is depicted in Figure 7. As you can see, the subscriber received messages with the topics `zigbee2mqtt/bridge/state` and `zigbee2mqtt/bridge/info`.

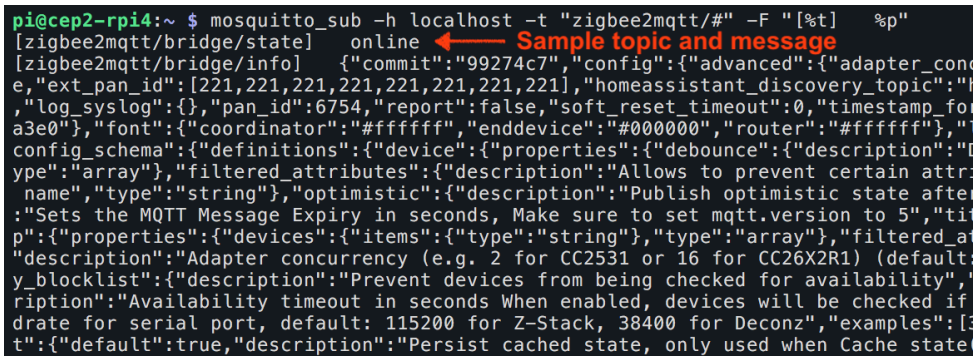

Figure 7. Output of subscribing to the top level topic zigbee2mqtt, using mosquitto_sub.

As previously stated, Z2M is also a subscriber, i.e. it subscribes to topics that can be used to monitor or configure it. This feature is what enables other applications to interact with Z2M. One example is the topic `zigbee2mqtt/bridge/request/health_check`, which can be used to check the current state of Z2M. The response to this request is published to the topic `zigbee2mqtt/bridge/response/health_check`. The following commands can be used to, respectively, publish and subscribe to Z2M's `health_check` topics:

```
mosquitto_pub -h localhost -t zigbee2mqtt/bridge/request/health_check -m ""
mosquitto_sub -h localhost -t zigbee2mqtt/bridge/response/health_check
```

In Figure 8 it is depicted the use of the commands above, specifically publish and subscribe to the `health_check` topics. It is also depicted Z2M's log output.

[1] The `-F` argument is used to format the output of the Z2M messages. In this case, it will print a string with the topic in square brackets, followed by the message payload. This argument can be omitted, but it can be helpful for aiding read the received messages.

Figure 8. Using the Mosquitto clients to interact with Z2M's health_check topics. On top, the publisher, middle the subscriber and on the bottom the log output of Z2M.

5 NEXT STEPS

You should know have Z2M installed and running in your Raspberry Pi. A lot of aspects were not covered here, but you should consider them for your project according to your requisites. All of the topics described next can be consulted in Z2M's documentation page [1].

5.1 Running Z2M in other platforms

Z2M also run on other platforms such as Windows and Docker You can find more information on how to install them in the following links:

- Windows: https://www.zigbee2mqtt.io/information/windows.html
- Docker: https://www.zigbee2mqtt.io/information/docker.html

5.2 Configuring Z2M

As shown, the default configuration of Z2M is enough for running Z2M. But other options are available, which might have to be changed in order to adapt Z2M to the requirements of your project. Some examples that you might look into are MQTT configurations (username and password for connecting to the MQTT broker), change the serial port where the USB adapter is connected, or change the Zigbee network name and key. These options are documented in [5].

5.3 Running Z2M as a Linux service

With the current installation, you'll have to start Z2M manually every time you boot he Raspberry Pi. This can be avoided if you run Z2M as a Linux service. Information on how to do this can be consulted in [6].

5.4 Topics

In section 4.3 some examples of Z2M topics were presented. A list of all the topics where Z2M publishes/subscribes is presented in [7], as well as its messages content (tip: all communications are in JSON). This page will be especially helpful when Zigbee devices are paired and you must check which topics you must subscribe.

5.5 Python clients

In section 4.3 you used the Mosquitto clients to interact with Z2M. Using the Python examples of tutorial 2, how would you modify them in order to communicate with Z2M? You can start by creating two separate clients (a publisher and a subscriber) and then try to merge them into a single application. Tip: start by the subscriber and see if you can receive messages from Z2M.

6 FURTHER READING

You should further explore the Zigbee2Mqtt's documentation [1] in order to gain a deeper understanding of its capabilities and how they fit your project requirements.

You can also start exploring what devices are supported (this will be address in detail in the next tutorial) and see how they report their events [8].

Finally, you can see a practical implementation of the Zigbee protocol in the IKEA Trådfri system and its integration with the Google Home [9]. This reference should give you a better understanding of the application of the Zigbee for home automation (or domotics) and may help inspire you to design your system.

7 REFERENCES

[1] Zigbee2Mqtt documentation: https://www.zigbee2mqtt.io

[2] Zigbee: A complete beginners breakdown: https://www.smarthomebit.com/a-beginners-guide-to-zigbee/

[3] Zigbee2Mqtt's Zigbee network:
https://www.zigbee2mqtt.io/information/zigbee_network.html

[4] Zigbee2Mqtt Github: https://github.com/koenkk/zigbee2mqtt

[5] Zigbee2Mqtt's configuration:
https://www.zigbee2mqtt.io/information/configuration.html

[6] Running Zigbee2Mqtt as a Linux service:
https://www.zigbee2mqtt.io/getting_started/running_zigbee2mqtt.html#5-optional-running-as-a-daemon-with-systemctl

[7] Zigbee2Mqtt's MQTT topics and message structure:
https://www.zigbee2mqtt.io/information/mqtt_topics_and_message_structure.html

[8] Zigbee2Mqtt's supported devices:
https://www.zigbee2mqtt.io/information/supported_devices.html

[9] Complete Ikea Tradfri + Google Home Guide: https://www.smarthomebit.com/complete-ikea-tradfri-google-home-guide/