

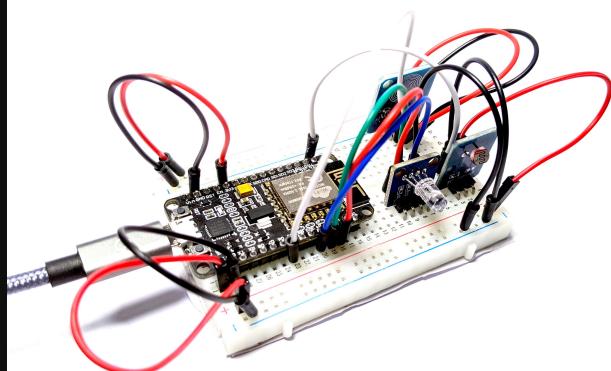
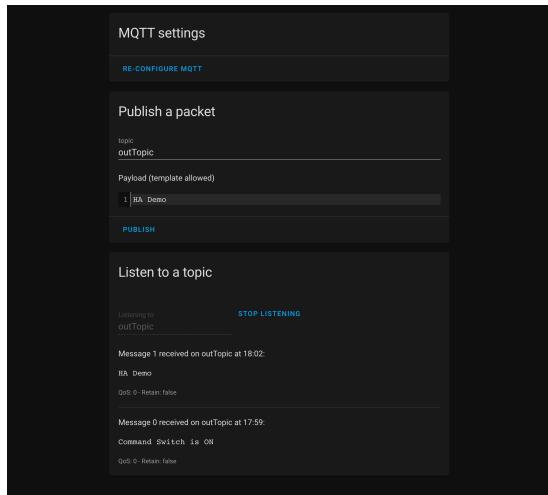
MQTT

A lightweight messaging protocol

Project description:



In this project, you will implement one additional lightweight communication protocol: MQ Telemetry Transport or MQTT. Precisely, you will work with Home Assistant, Mosquitto, and the Arduino IDE to setup a simple MQTT messaging protocol. There is no need for you to build a separate circuit for this exercise. You will use the smart chair circuit that you have built during the course and upload a different code sketch.



Project objectives:

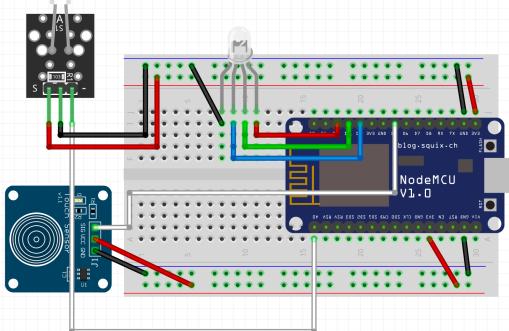


- Modify the smart chair circuit behaviour with alternative code
- Use Home Assistant and Mosquitto as the MQTT protocol broker
- Subscribe and send messages from different clients to the broker

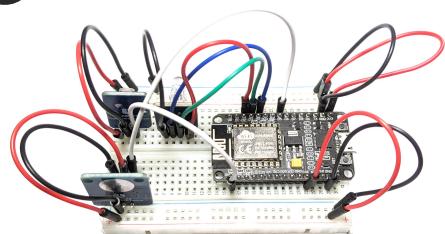
Project components:



Component Reference	Component Quantity	Component Name	Component Description
A	1	Smart Chair circuit	The smart chair circuit which features the photoresistor, the RGB LED, and the touch sensor
B	1	Micro USB Cable	A USB cable to power and upload instructions to a microcontroller



A



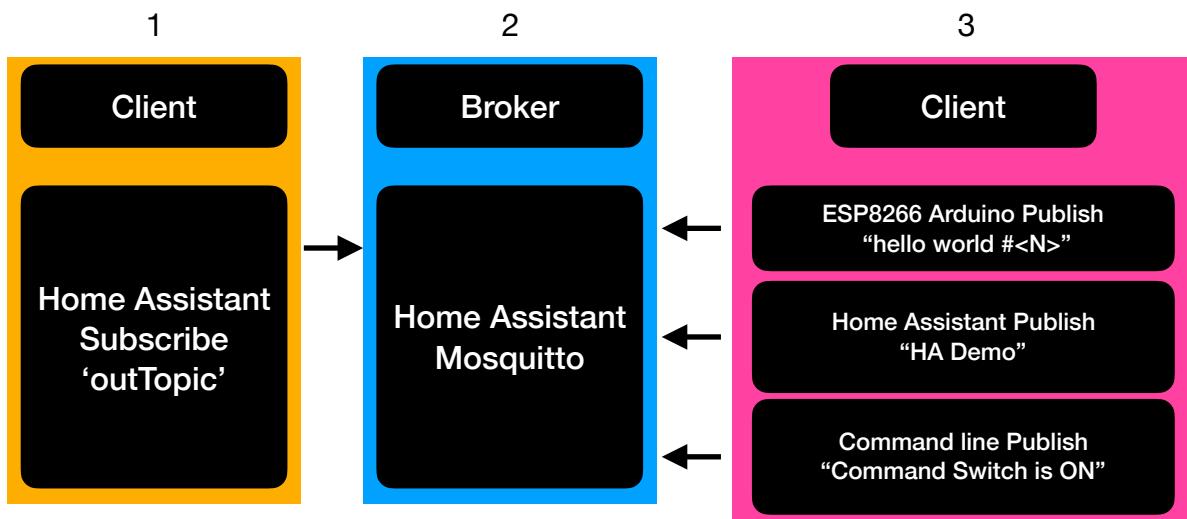
B

Step One

Project Introduction

The idea behind this exercise is for you to practice what you learnt in the video lectures and implement a simple MQTT messaging protocol.

Here, you will setup the MQTT protocol structure explained in the previous lecture:



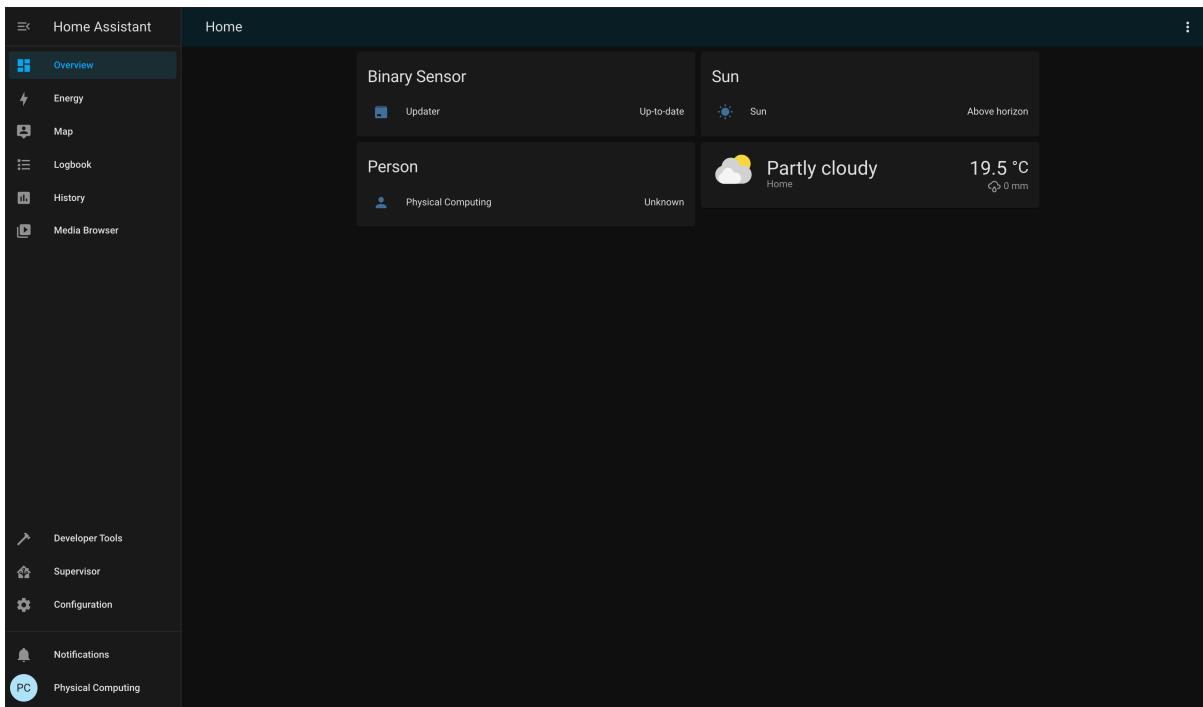
- 1) **Home Assistant** will act as the **client** which can register to a particular topic, in this case '**outTopic**', and listen for incoming messages.
- 2) **Home Assistant** will also act as the MQTT protocol **broker** with the integration of the the **Mosquitto** add-on.
- 3) The **ESP8266**, **Home Assistant**, and the **Mosquitto** (via Terminal) can then be used to publish messages to the '**outTopic**' topic.

Step Two

Home Assistant as the MQTT broker

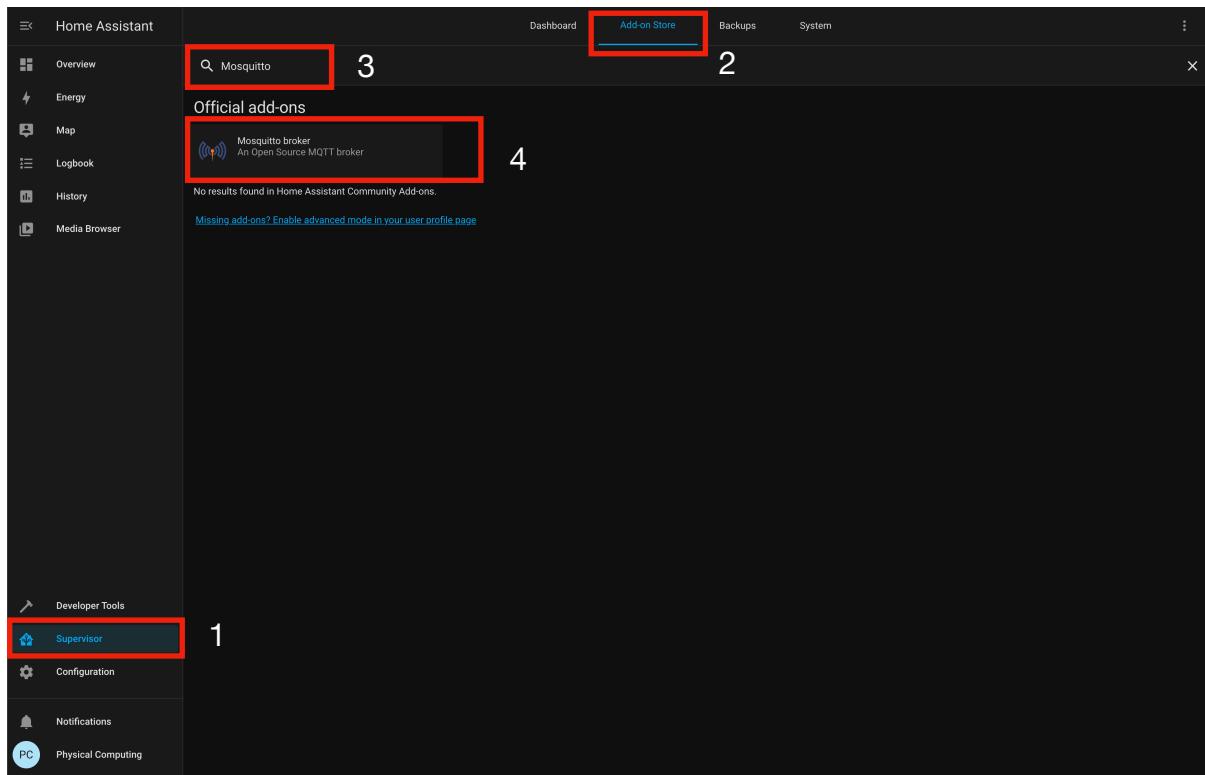
In this section of the exercise, you will setup Home Assistant as the MQTT broker using the Mosquitto add-on.

Use VirtualBox to run Home Assistant and go to the application dashboard:

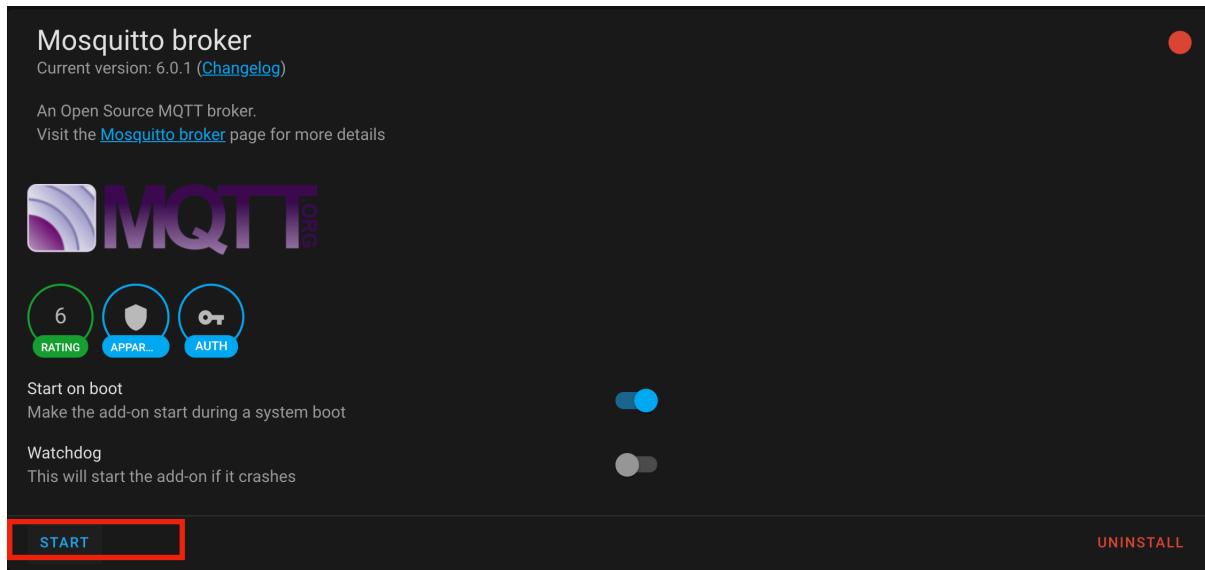


Note: If you did not attempt any of the Home Assistant exercises, or you simply forgot how to run Home Assistant, please refer to the Home Assistant exercises section in Topic 5 of this course.

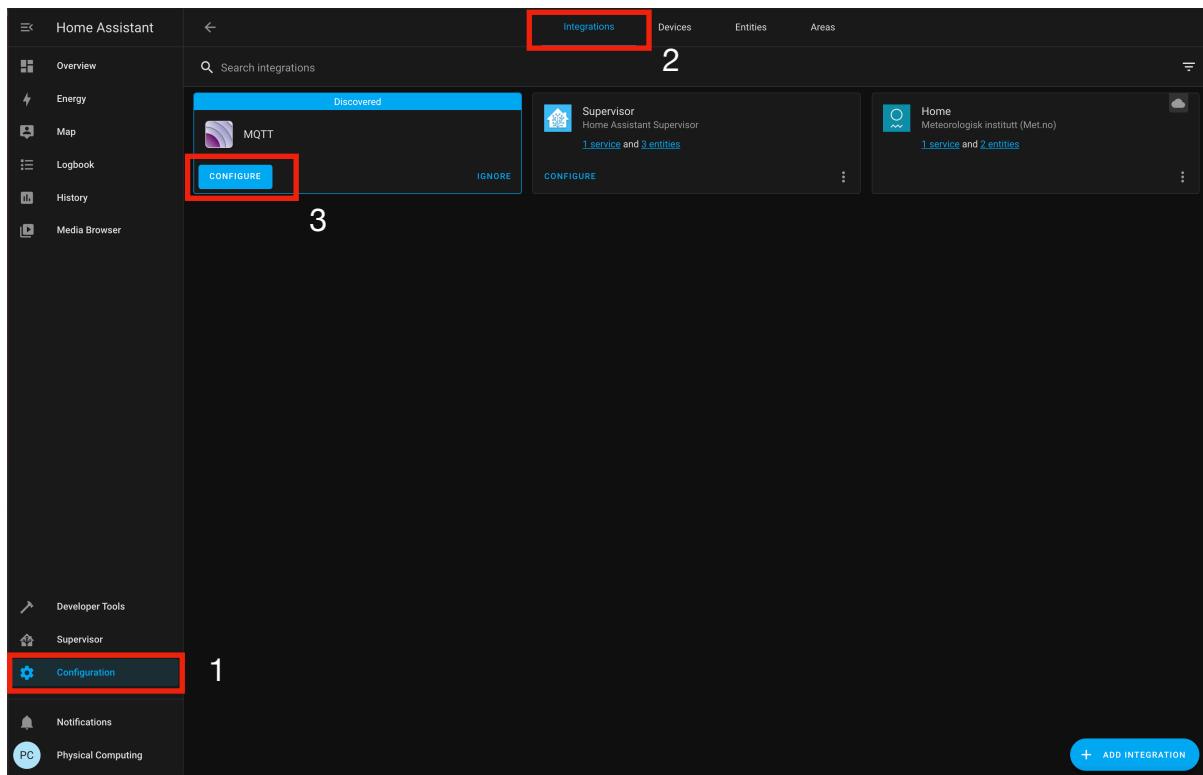
The first step to turn Home Assistant into an MQTT broker is to install the Mosquitto broker add-on. In the Home Assistant dashboard click on **Supervisor** first, then click **Add-on Store**, search for “**Mosquitto**”, and finally install the **Mosquitto broker** add-on as shown below:



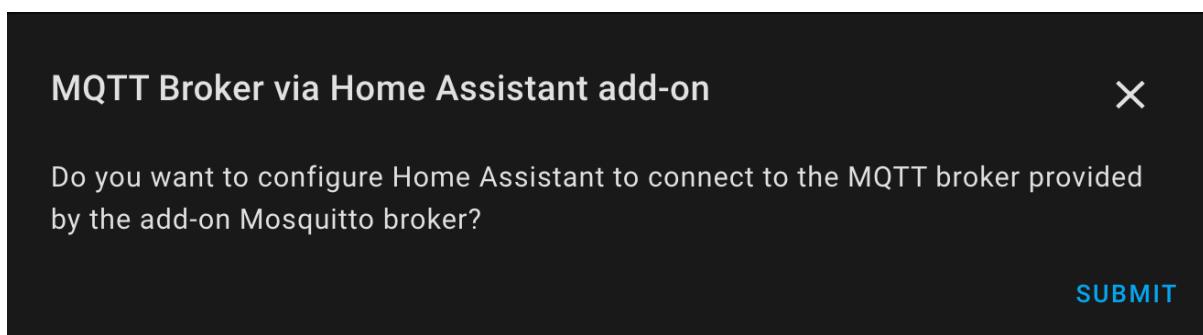
Once you have installed **Mosquitto**, simply click on **START** to run the add-on. This will initialise the broker:



Now that Mosquitto is running, the next step is to configure the broker. Click on **Configuration**, then click on **Integrations**, and finally click on the **CONFIGURE** button below the MQTT add-on.



A pop up message should appear. Click on the **SUBMIT** button to configure the MQTT broker via the Home Assistant add-on:



The default configuration is actually fine, you just need to replace the default Broker IP address with the IP address of Home Assistant. To do so, first open the VirtualBox terminal and make a note of the IP address of your running instance of Home Assistant like shown below:

```
Waiting for the Home Assistant CLI to be ready...
[| |] [^ ^] [O O] [| |]
[| |] [v v] [o o] [| |]
[| |] [^ ^] [o o] [| |]
[| |] [v v] [O O] [| |]

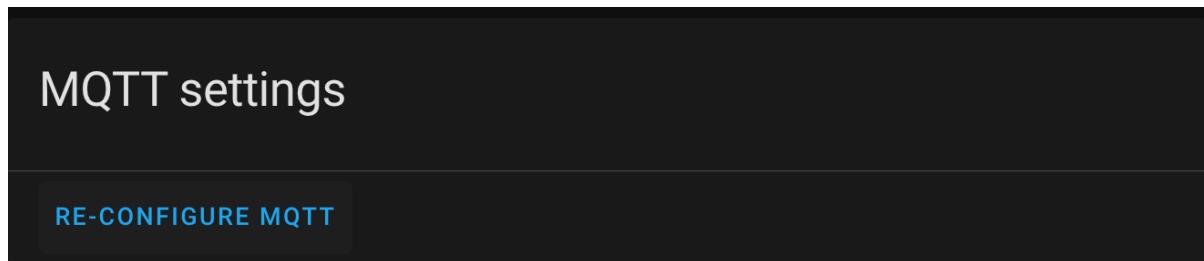
Welcome to the Home Assistant command line.

Waiting for Supervisor to startup...
System information
  IPv4 addresses for em0: 192.168.1.114/24
  IPv6 addresses for em0: 2a01:ab00:842d:c00:5bed:fa39:57f:4eb6/64, fe80::4280:74f5:6043:6259/64

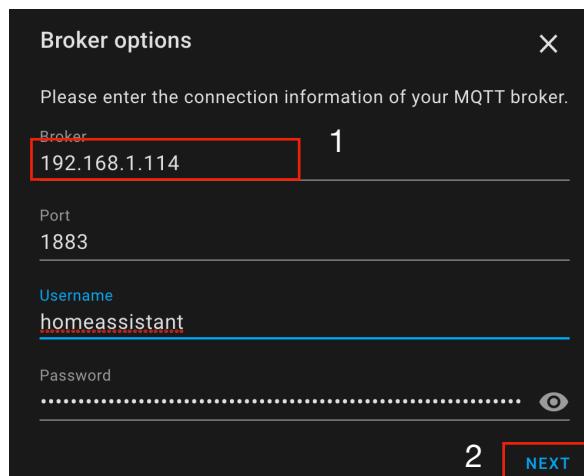
OS Version:          Home Assistant OS 6.4
Home Assistant Core: landingpage

Home Assistant URL:  http://homeassistant.local:8123
Observer URL:       http://homeassistant.local:4357
ba >
```

Now back to the broker configuration dashboard, click on the **RE-CONFIGURE MQTT:**



Replace the existing **Broker** field with your Home Assistant IP address and click on **NEXT**:



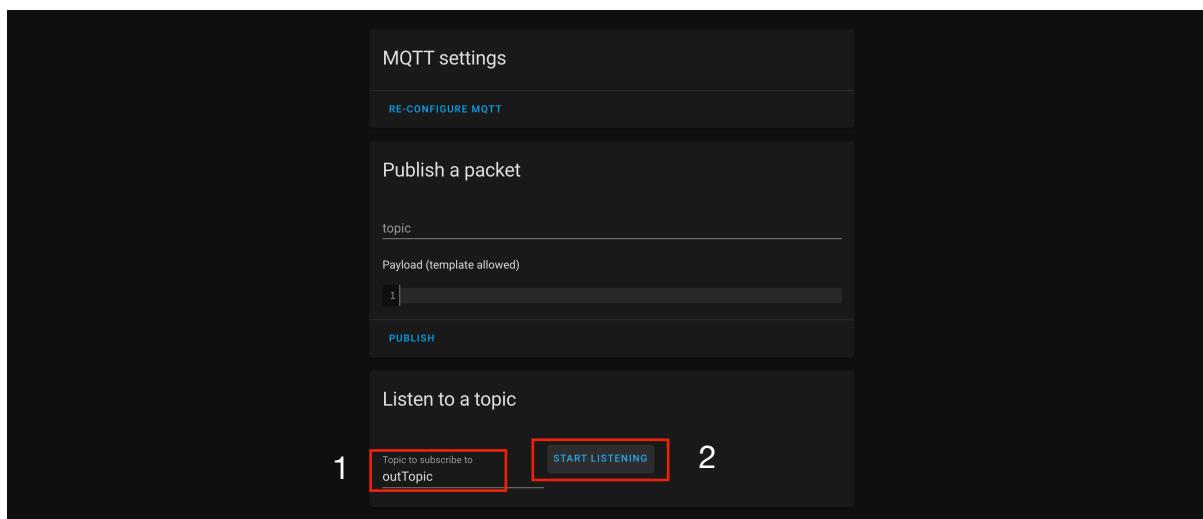
Great, **Home Assistant** is now configured with **Mosquitto** as the MQTT broker.

Step Three

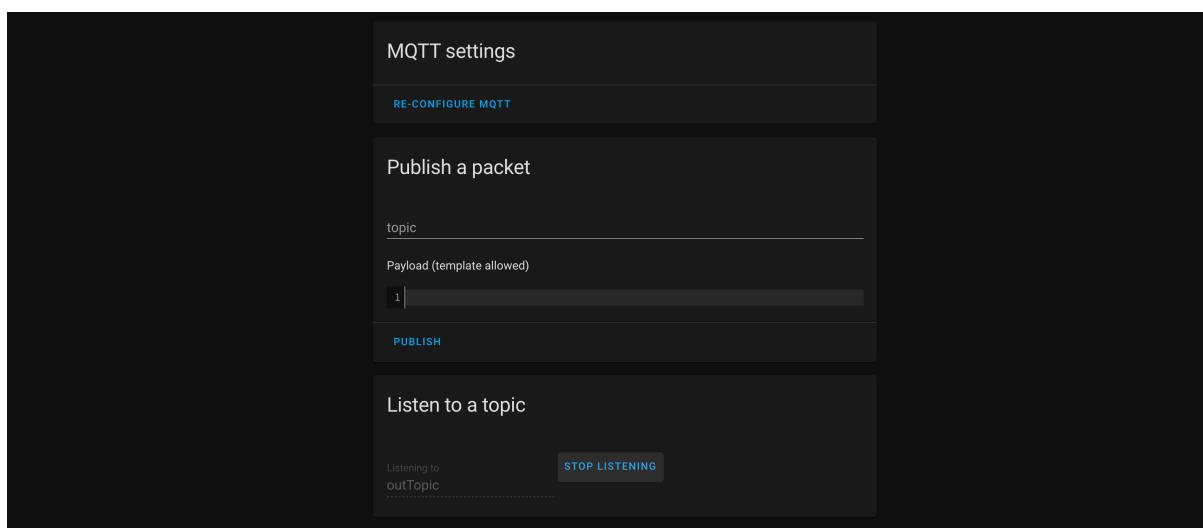
Subscribe Home Assistant to listen on a topic

This step of the exercise is very quick. Now that the MQTT broker is setup and running, you can subscribe Home Assistant to listen to a specific topic. Let's subscribe Home Assistant to listen to a topic called "**outTopic**".

Type the topic name "**outTopic**" in the "Topic to subscribe to" field and click on the **START LISTENING** button:



Now **Home Assistant** is subscribed to "**outTopic**" and it is listening for incoming messages on such a topic:

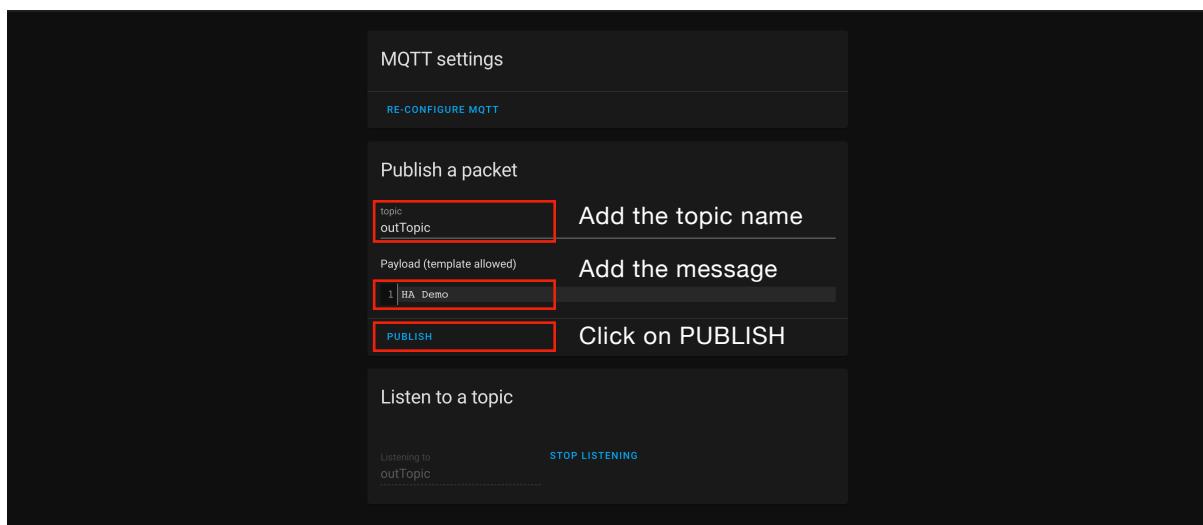


Step Four

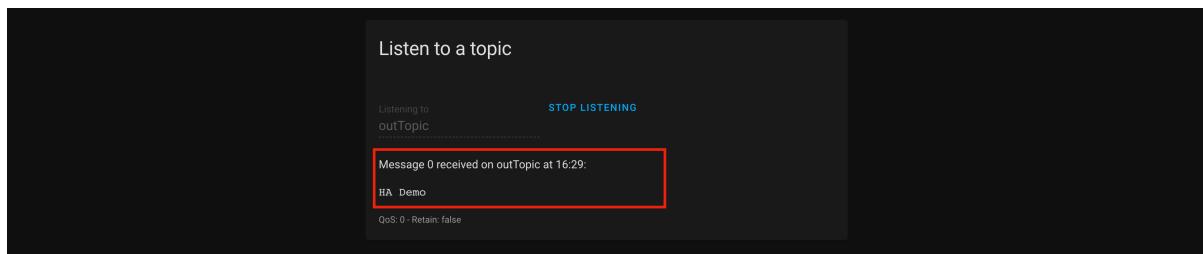
Publishing messages from your ESP, Home Assistant, and the Terminal

Now that Home Assistant is subscribed and it is listening to the “**outTopic**” topic, you can publish messages to such a topic. This part of the exercise will allow you to publish a message to the “**outTopic**” topic via three clients: Home Assistant itself, your ESP board, and via command line using the Terminal.

Let’s begin by publishing a message using Home Assistant. Head again to the MQTT add-on settings and publish a simple message “HA Demo” from the “Publish a packet” section:



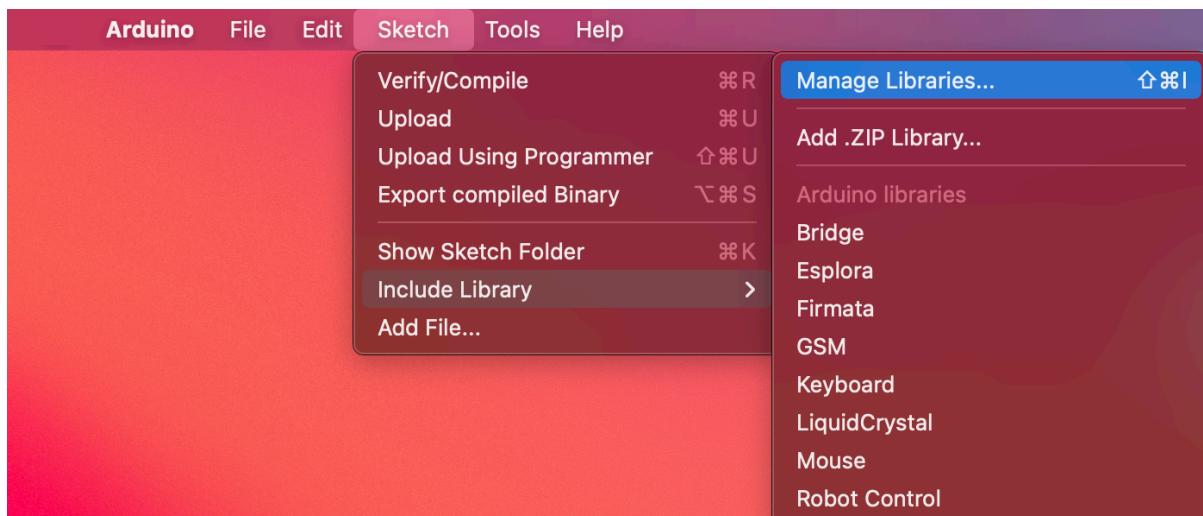
Make sure that Home Assistant is still listening to the “outTopic” topic. You should now see a new message listed on the “Listen to a topic” section:



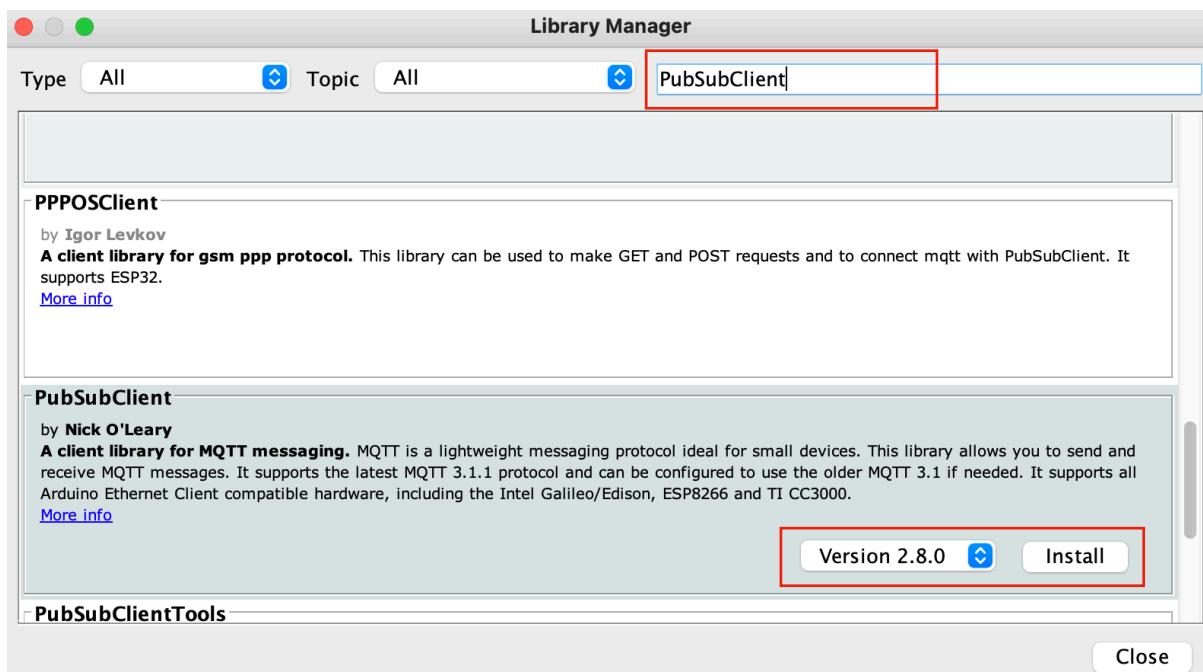
Great, the first client can successfully send messages. Let’s now see how you can send a message to the ‘outTopic’ topic via your ESP board.

In order to use your ESP board as another MQTT client, you will need to install a third-party library to deal with the communication: the **PubSubClient** library.

With your Arduino IDE open, click on Sketch -> Include Library -> Manage Libraries... like shown below:

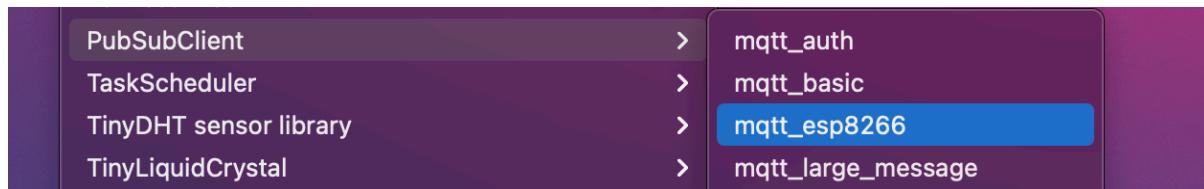


Next, search for “RFID” and install the latest version of the **PubSubClient** library as shown below:



Now that you have installed the **PubSubClient** library, go ahead and open the following Arduino sketch example:

File > Examples > PubSubClient > mqtt_esp8266



Feel free to save the sketch and rename it to something sensible:
mqtt_smat_chair for instance.

There are three main parts that you need to change before uploading this sketch to your smart chair ESP circuit:

- Change the **ssid** variable to your WiFi network name
- Change the **password** variable to your WiFi network name
- Change the **mqtt_server** variable to your Home Assistant IP address

```
const char* ssid = "your WiFi network name";
const char* password = "your WiFi network password";
const char* mqtt_server = "your Home Assistant IP";
```

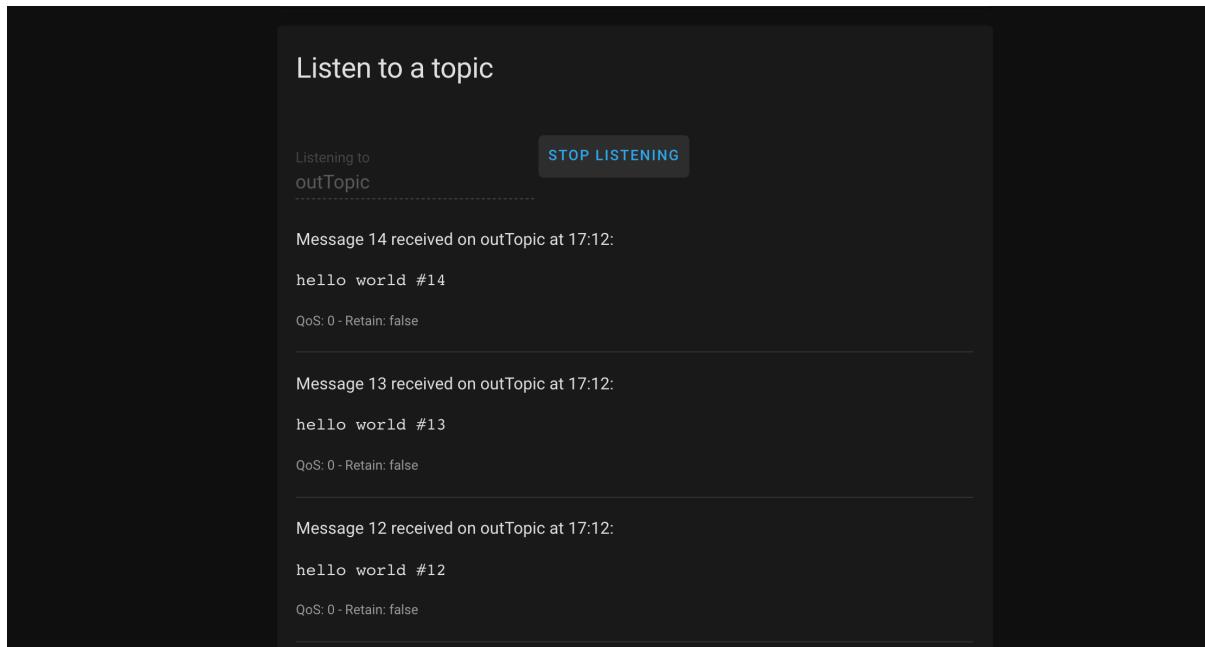
Among other functionalities, the sketch publishes the message "hello world" to the topic "outTopic" every two seconds to the provided broker details.

Go ahead, compile your code, and upload it to your ESP board.

Take a look at the serial monitor and make sure that the ESP board connects to the WiFi and that it sends the "hello world" message to the broker every two seconds.

Now back to Home Assistant, reconfigure the broker to listen to "outTopic" and press the RST button on your ESP board to restart the loaded sketch.

You should now see incoming messages from the ESP client:



Hurray, you have successfully sent messages to the broker from two different clients: Home Assistant itself and your ESP board.

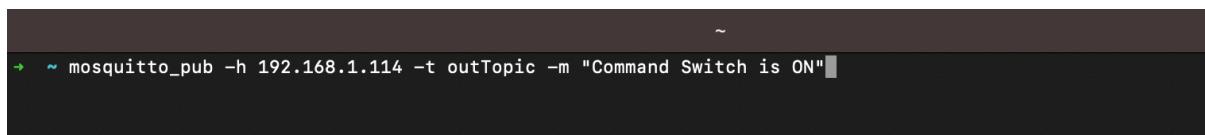
As a final step, let's see how you can send messages to the broker with one more client: the **Mosquitto** client.

Go ahead and download the Mosquitto client for your operative system:

<https://mosquitto.org/download>

Follow the installation instructions on the above link to setup **Mosquitto** client.

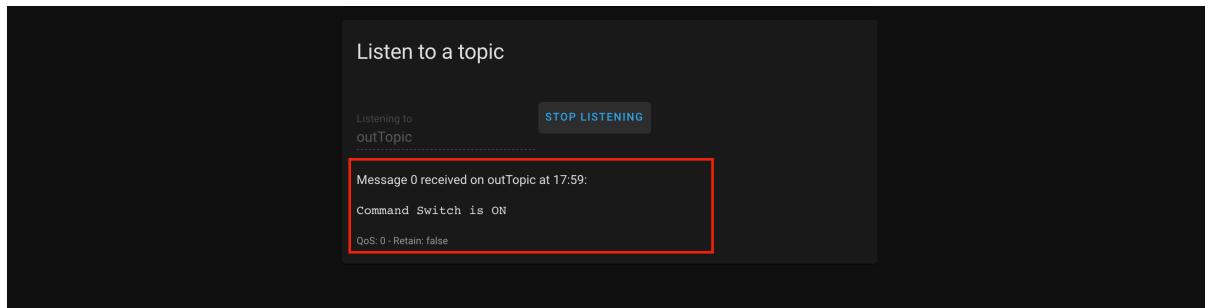
Once you have installed the program, **Mosquitto** allows you to send messages to a particular broker topic via the Terminal. For instance, if you want to send the “Command Switch is On” message to the “outTopic” topic of the Home Assistant broker, you can use the following command line on your Terminal:



```
~
→ ~ mosquitto_pub -h 192.168.1.114 -t outTopic -m "Command Switch is ON"
```

`mosquitto_pub -h <IP> -t <Topic_Name> -m <Message>`

Run the above command on your Terminal and you will see the “**Command Switch is On**” message appear in the Home Assistant broker list:



Hurray, you have now setup and run a functional MQTT protocol and explored three different clients to send messages to the broker.

Step Four Practice with MQTT

Now that you are familiar with MQTT and also explored how to implement the protocol, why don't you try to adapt the **PubSubClient** example sketch code to one of your circuits?

You can first spend some time to understand the sketch code structure. You can then adapt the client code to one or more of your circuits to send information about their components to the Home Assistant broker.

Alternatively, you can attempt to subscribe the client to a particular topic and read incoming messages. There is a code section inside the **PubSubClient** example sketch that allows the client to subscribe and listen to a topic called “inTopic”.