**MQTT PROTOCOL**

MQTT is a communication protocol based on a publish and subscribe system. Devices can subscribe to a topic or publish data on a topic. Devices receive messages that are published on topics they are subscribed to.

## MQTT Protocol:

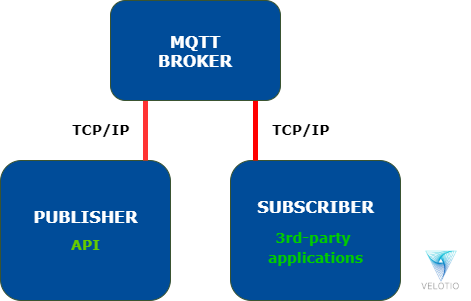
MQTT is a very simple publish / subscribe protocol. It allows you to send messages on a topic (channels) passed through a centralized message broker.

The MQTT module of API will take care of the publish/ subscribe mechanism along with additional features like authentication, retaining messages and sending duplicate messages to unreachable clients.

There are three parts of MQTT architecture:

* MQTT Broker: All messages passed from the client to the server should be sent via the broker.
* ‍MQTT Server: The API acts as an MQTT server. The MQTT server will be responsible for publishing the data to the clients.
* MQTT Client: Any third party client who wishes to subscribe to data published by API, is considered as an MQTT Client.

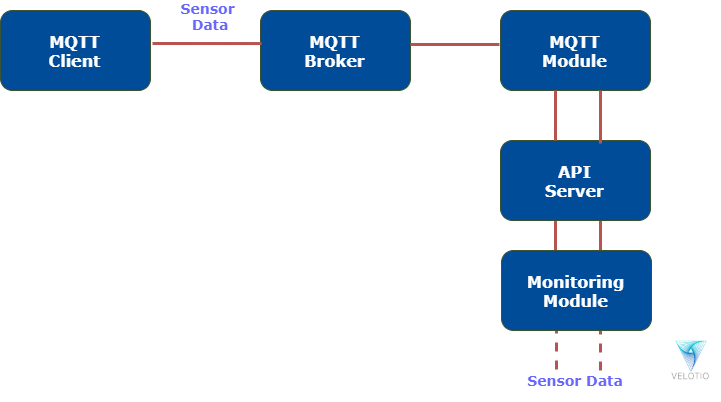
The MQTT Client and the MQTT Server need to connect to the Broker in order to publish or subscribe messages.



*Figure: MQTT Communication Program*

Suppose our API is sending sensor data to get more ideas on MQTT. API gathers the sensor data through the Monitoring module, and the MQTT module publishes the data to provide different channels. On the successful connection of an external client to the MQTT module of the API, the client would receive sensor data on the subscribed channel.

Below diagram shows the flow of data from the API Module to the External clients.



MQTT – Publish/Subscribe

The first concept is the publish and subscribe system. In a publish and subscribe system, a device can publish a message on a topic, or it can be subscribed to a particular topic to receive messages



For example Device 1 publishes on a topic.

Device 2 is subscribed to the same topic that device 1 is publishing in.

So, device 2 receives the message.

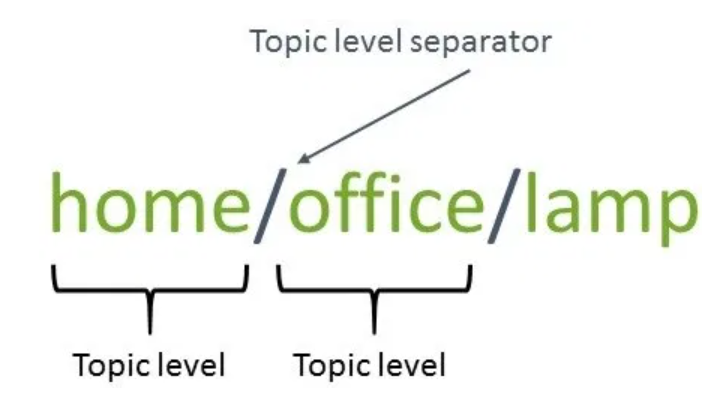
MQTT – Messages

Messages are the information that you want to exchange between your devices. It can be a message like a command or data like sensor readings, for example.

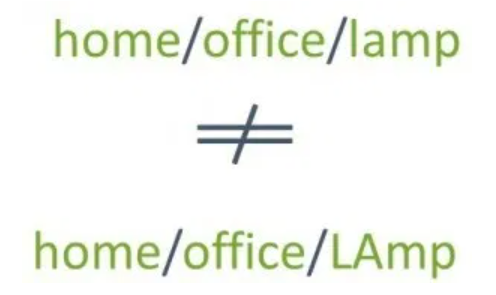
MQTT – Topics

Another important concept is the topics. Topics are the way you register interest for incoming messages or how you specify where you want to publish the message.

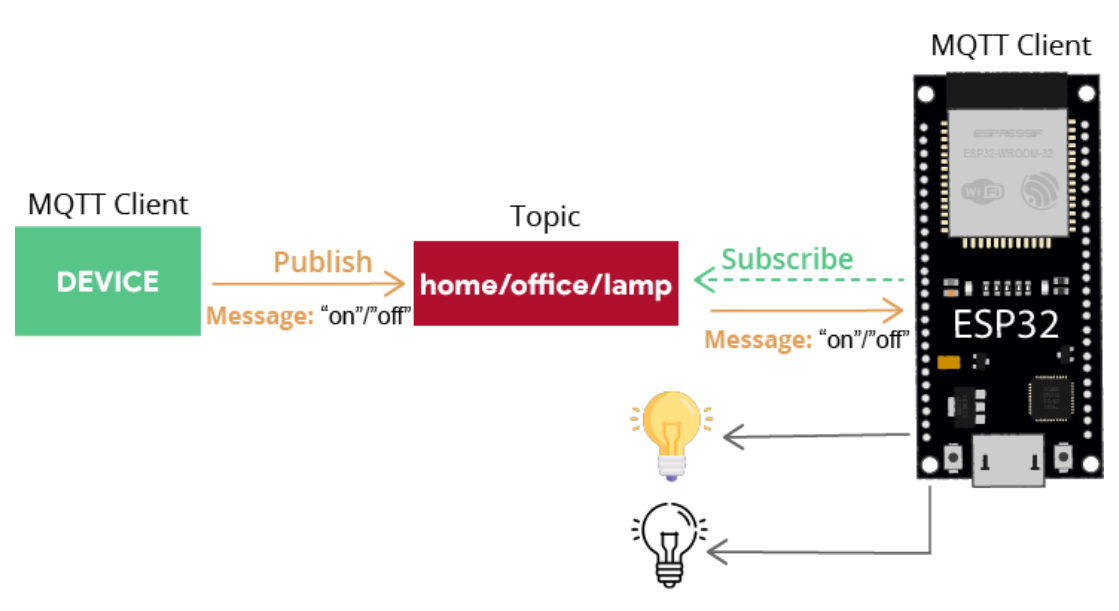
Topics are represented with strings separated by a forward slash. Each forward slash indicates a topic level. Here’s an example of how you would create a topic for a lamp in your home office:



Note: topics are case-sensitive, which makes these two topics different:

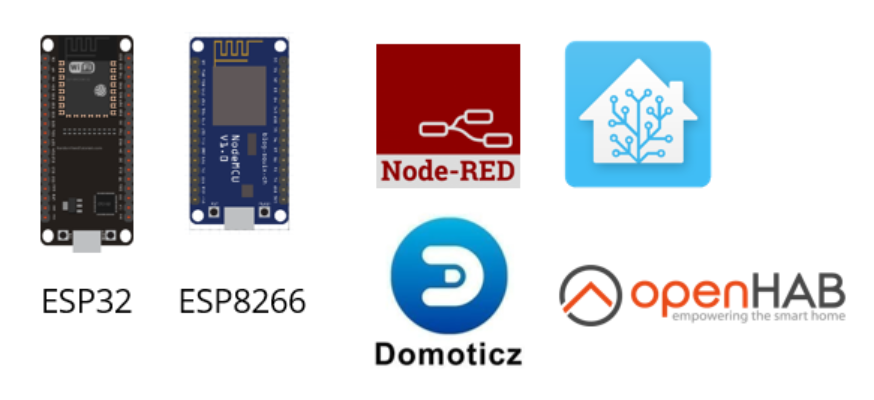


If you would like to turn on a lamp in your home office using MQTT you can imagine the following scenario:



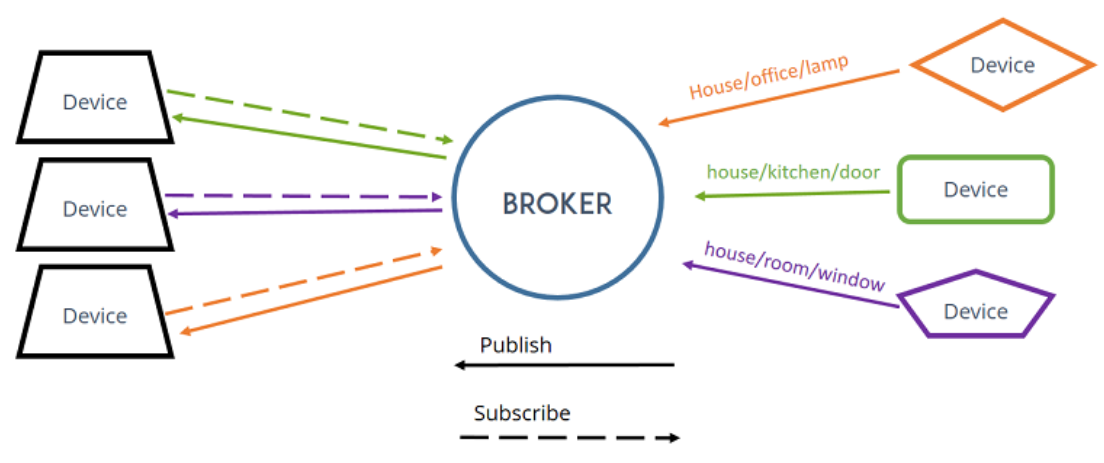
1. A device publishes “on” and “off” messages on the home/office/lamp topic.
2. You have a device that controls a lamp (it can be an ESP32, ESP8266, or any other board or device). The ESP32 that controls your lamp, is subscribed to that same topic: home/office/lamp.
3. So, when a new message is published on that topic, the ESP32 receives the “on” or “off” messages and turns the lamp on or off.

The device that is publishing the messages can be an ESP32, an ESP8266, or an Home Automation controller platform with MQTT support like Node-RED, Home Assistant, Domoticz, or OpenHAB, for example.



MQTT – Broker

Finally, another important concept is the broker. The MQTT broker is responsible for receiving all messages, filtering the messages, deciding who is interested in them, and then publishing the message to all subscribed clients.



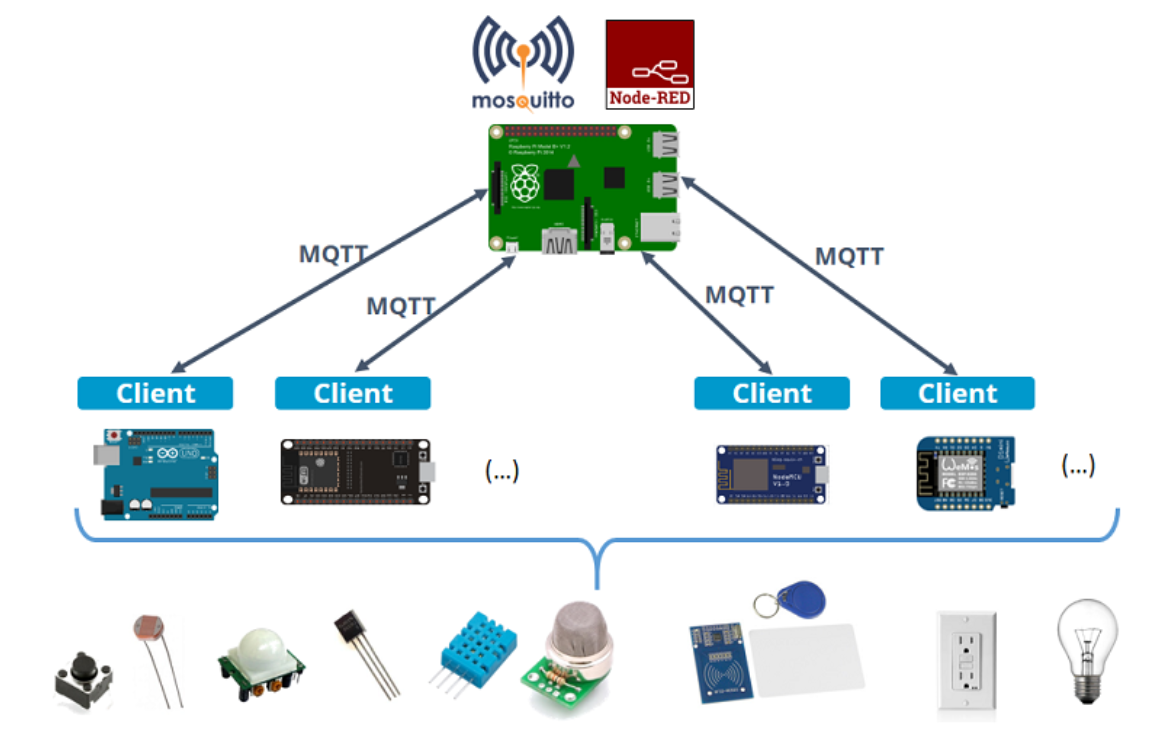
There are several brokers you can use. In home automation projects, we use the Mosquitto Broker installed on a Raspberry Pi. You can also install the Mosquitto broker on your PC (which is not as convenient as using a Raspberry Pi board, because you have to keep your computer running all the time to keep the MQTT connection between your devices).



Having the Mosquitto broker installed on a Raspberry Pi on your local network allows you to exchange data between your IoT devices that are also connected to that same network. You can also run Mosquitto MQTT broker in the cloud. Running the MQTT Mosquitto Broker in the cloud allows you to connect several IoT devices from anywhere using different networks as long as they have an Internet connection.

Use MQTT in Home Automation and IoT Projects

MQTT is great for home automation and internet of things projects. Here’s an example of how it can be used in a Home Automation System built with low-cost development boards like a Raspberry Pi, ESP32, ESP8266, and Arduino.



* A Raspberry Pi runs the Mosquitto broker, which is essential for MQTT protocol.
* The same Raspberry Pi runs Node-RED, which is a Home Automation Platform with MQTT support—this means it can subscribe to topics to receive messages from the other IoT devices, and publish messages on specific topics to send messages to other devices.
* Node-RED also allows you to build an User Interface with buttons to control outputs and charts to display sensor readings.
* The Arduino, the ESP32 and ESP8266 can act as MQTT clients that publish and subscribe to topics.
* These boards are connected to actuators like LEDs or relays, and sensors like temperature, humidity, smoke sensors, etc..
* These boars can publish data about the sensor’s state on a specific topic, that Node-RED is also subscribed to. This way, Node-RED receives the sensor readings and can display them on the user interface.
* On the other side, Node-RED can publish data on a specific topic to control outputs when you use the buttons on the interface. The other boards are also subscribed to that topic and control the outputs accordingly.