



MQTT is a protocol that specifically sends data from devices of the Internet of Things and is supported by most microcontrollers and systems. To use Raspberry Pi MQTT communication, not much is needed, which is why this type of transmission is very interesting. In addition, it is really easy to use.

In this tutorial, we install a Raspberry Pi MQTT broker, whereupon we receive and send data. We can either use several Raspberry Pi's for testing or just use a single device.

ADVERTISING

Accessory

This tutorial works without much additional hardware. However, you either have to have at least one **Raspberry Pi 3** (alternatively a Model Zero W) or connect a **wifi stick** to your Raspberry Pi to use MQTT.

Other devices such as an **ESP8266** can also communicate using MQTT. This is possible with Raspberry Pi's as well as other devices. Also for the NodeMCU there are ready-made packages.



Wat met het klimaat?

Ad Kernenergie stoot even
als hernieuwbare energie.

Forum_Nucléaire_DCM

Meer info >

What is MQTT?

Some may wonder what MQTT should be. To quote the official **MQTT** page:

“ **MQTT is a machine-to-machine (M2M)/”Internet of Things” connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.** ”

It is therefore a standardized network protocol with which short messages / commands can be transmitted. The big advantage is that the built-in wifi adapters (for example in the Raspberry Pi or in the ESP8266) are used for the internet connection. More accessories and complicated wiring is not necessary! This makes it possible to send the data via the local network or the Internet.

In detail, the transfer consists of three different components:

- **Publisher:** Sends messages.
- **Broker:** Forwards messages to registered subscribers.
- **Subscriber:** Receives messages through the broker.

Mosquitto

There are several applications that can be used to send and receive through MQTT, but the simplest on the Raspberry Pi is probably **Mosquitto**. We will install this on the Raspberry Pi first:

```
sudo apt-get install -y mosquitto mosquitto-clients
```

After installation, a Mosquitto server is started automatically. We open a subscriber in the channel "test_channel" waiting for messages:

```
mosquitto_sub -h localhost -v -t test_channel
```

The channel is here like a frequency, on which one hears. For example, different data may be sent in different channels (e.g., temperature, humidity, brightness, etc.).

In order to simply transfer data, we can either use the same Raspberry Pi (open new terminal / SSH connection) or send the data from another Pi. If we use the same Raspberry Pi, use is easily possible. For this we simply send a test message (as publisher) in the same channel:

```
mosquitto_pub -h localhost -t test_channel -m "Hello Raspberry Pi"
```

Otherwise you have to specify the internal IP address (eg 192.168.1.5) of the recipient instead of "localhost". On the receiver side, the message should appear so for now.

Raspberry Pi MQTT data exchange with Python

The communication is super easy, as we have seen. In order for us to be able to use the whole thing from scripts, we want to make it available to Python. For this purpose, we first install a library via the Python **package manager** (for Python3 also use pip3):

```
sudo pip install paho-mqtt
```

Alternatively (for example, if the package could not be properly installed) you can also install it manually:

```
git clone https://github.com/eclipse/paho.mqtt.python.git
cd paho.mqtt.python
python setup.py install
```

We start with the receiver. For this we create a new file with the content:

```
sudo nano mqtt_subscriber.py
```

```

6 # The callback for when the client receives a CONNACK response from the server.
7 def on_connect(client, userdata, flags, rc):
8     print("Connected with result code "+str(rc))
9
10    # Subscribing in on_connect() means that if we lose the connection and
11    # reconnect then subscriptions will be renewed.
12    client.subscribe(MQTT_PATH)
13
14    # The callback for when a PUBLISH message is received from the server.
15    def on_message(client, userdata, msg):
16        print(msg.topic+" "+str(msg.payload))
17        # more callbacks, etc
18
19    client = mqtt.Client()
20    client.on_connect = on_connect
21    client.on_message = on_message
22
23    client.connect(MQTT_SERVER, 1883, 60)
24
25    # Blocking call that processes network traffic, dispatches callbacks and
26    # handles reconnecting.
27    # Other loop*() functions are available that give a threaded interface and a
28    # manual interface.
29    client.loop_forever()
  
```

After saving and closing (CTRL + O, CTRL + X) we execute the file:

```
sudo python mqtt_subscriber.py
```

Either you send a command as usual by Shell Command, or we also create a Python Sender / Publisher.

The publisher's code looks like this (create a new file and then run it, keeping in mind the correct IP address):

```

1 import paho.mqtt.publish as publish
2
3 MQTT_SERVER = "192.168.1.5"
4 MQTT_PATH = "test_channel"
5
6 publish.single(MQTT_PATH, "Hello World!", hostname=MQTT_SERVER)
7
  
```

That's it! Now you can include the appropriate parts in your scripts and receive or send MQTT messages from other Raspberry Pi's. Of course, this is not just limited to Raspberry Pi's because the messages can also come from or be addressed to other devices.



Printed Circuit Boards
circuit-boards.eu

Ad PCB Prototypes & Seri
1WD Service Low Price & I
multi-circuit-boards.eu

Learn more

must have a network connection.

Another possibility without internet connection are normal radio transmitters and receivers. Here is e.g. The **433 MHz radio transmission** is very popular because the modules are very cheap and many devices such as radio-controlled sockets or garage doors are compatible with it. However, this transfer is limited to 24 bits, which is not very much.

More data per message can be transmitted via 2.4 GHz radio transmission. This transfer differs from MQTT because extra modules are used. However, it is also possible to send commands or messages to Arduinos.



2.4 GHZ 433 MHZ ARDUINO BROKER BROKER DATA TRANSFER ESP8266 MOSQUITTO MOSQUITTO MQTT MQTT PYTHON

PYTHON WIFIADAPTER WIFI STICK WIRELESS MODULE WIRELESS OUTLET WIRELESS RECEIVER WIRELESS TRANSMITTER



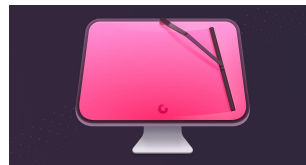
Printed Circuit Boards -
[multi-circuit-boards.eu](#)

Ad multi-circuit-boards.eu



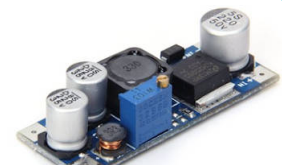
Connect a Raspberry Pi
Keypad - Code Lock

tutorials-raspberrypi.com



CleanMyMac X

Ad MacPaw



Run a Raspberry Pi on
rechargeable Batteries

tutorials-raspberrypi.com



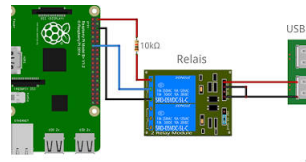
python databases

Ad crate.io



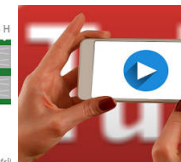
How to setup a
Raspberry Pi Security
Camera Livestream

tutorials-raspberrypi.com



Raspberry Pi - Build
your own UPS
(Uninterruptible Power...

tutorials-raspberrypi.com



Raspberry Pi:
Live Streaming

tutorials-raspberrypi.com

PREVIOUS ARTICLE

← **Use C# and .NET to develop your own GUI apps**
for the Raspberry Pi

NEXT ARTICLE

Raspberry Pi: YouTube Live Streaming Setup →

17 Responses

 **adam** 2 17 2018 at 17:17

We use cookies to ensure that we give you the best experience on our website. If you continue to use this site we will assume that you are happy with it.

OK

can i communicate using my internet router (Dlink).
what changes should i make ?
thank you

REPLY

**Felix** 3-17-2018 at 18:21

You want to communicate through or with the router?

REPLY

**adam** 3-18-2018 at 11:54

through the router.
and one more thing how do i subscribe to multiple topics.
thanks

**Felix** 3-18-2018 at 15:12

Yes, of course. Any router should be fine. Just use the subscribe command several times.

**Adam** 3-20-2018 at 17:07

Thank you so much
This code helped me a lot in my project ??

**Felix** 3-20-2018 at 20:29

You are welcome 😊

**adam** 4-1-2018 at 15:01

hi,
i wanted one more help .
how do we connect to multiple ip address ?
i have 2 topics on different raspberry pi
and im connecting them to a laptop
i am just able to connect to the first client.connect(ip1)
and not the client.connect(ip2)

can i communicate rpi3 and app phone via wifi? I need stream data with phone accelerometer to rpi3, it is possible? -?

REPLY



Felix 3-20-2018 at 20:29

I don't think that your Phone supports MQTT...

REPLY



Jimmy 5-11-2018 at 19:14

When I tried the code on this page, I used the app MQTTTool on my iPhone 5S, – and it worked!
 There are several MQTT apps for iPhone, but so far I have just tested MQTTTool, Mqtt buddy (a bit hard to understand), and IoT Manager (crashes sometimes and hard to get to connect).

So try search for “mqtt client” at the App Store.

For now I'm trying to find/learn how to get a notification-signal on my iPhone when I get a message from the mqtt-server/broker. No luck yet.

Felix: Thank you for the guide! You made me understand how mqtt works!

REPLY



Bhavya 4-5-2018 at 11:22

Hi I need complete information about using mqtt established msg send and received using telegram app with raspberry pi can u pls send the full coding part pls....

REPLY



maria 5-9-2018 at 13:11

I need it too.

REPLY



Madhu 6-24-2018 at 19:43

Could you please tell us how to send the sensors data to another pc using mqtt

REPLY



Mike. M. 10-2-2018 at 4:27

You need to process sensor data and send in mqtt message. There is python code out there for both, need to merge to get what you want. I am in process of trying to do that.

REPLY



DIVYA 8-3-2018 at 9:38

Hi,

Hello, i am having an issue with trying to communicate with the two python scripts.

I am using them both in two different terminals in the same raspberry pi, but none of them is sending or accepting anything.

When I try to test the communication with commands only it works fine, but not with python scripts 😞

REPLY



Erjona 4-12-2019 at 16:46

Fixed it, thanks

REPLY

Leave a Comment

Your email address will not be published.

Name

Email

☐ Notify me of follow-up comments by email.

SUBMIT COMMENT





