In my last video, I started to use MQTT and a Raspberry Pi as a Broker This week, I got my new Pi zero with Wi-Fi. With this board, there is no excuse to not use MQTT for my projects. So, today, I build a cheapo MQTT broker for under 20 dollars. Look at the length of this video and you see, how easy it is to build one. And you will be astonished, what this little bugger is able to do in addition to broker MQTT messages! So, let’s start:

The secret behind the whole thing is Peter Scargill. He is a wizard in small devices and I recommend to check out his blog. Pete produced a Christmas gift for the community: A script, which automatically installs a lot of things on Linux systems, including the Raspberry Pi. You chose, which packets you want and wait for a few minutes or hours. After that, you get your system more or less completely configured. Really great stuff!

I wanted to use this script for my new Pi zero, when I read, that Peter was not happy with Raspian on the Pi Zero. It was way too slow for him. Fortunately, he found a better solution: Diet Pi, which can also be used for our project. So, I started with installing dietPi on an SD card, enabled it for my Wi-Fi, and copied Peter’s script to my device. After that, I created two new users, Pi and admin, and started the script. Then, I had to wait for a few hours. Eventually, the script finished, and I had to reboot my Zero. You find the link in the description, if you want to use Peters script for your purposes.

To check my newly created system out, I started the Mosquitto broker. It worked. Next, I started Node-Red using the admin user. And it worked, too. Maybe not as fast as I runs on my pi 3, but usable without any problems. When I looked at the available nodes, I discovered, that Peter did completed staff work. He already added many nodes like his famous big-timer, johnny5, pushbullet, some google stuff, raspberry stuff and so on, and so on. This alone is worth many hours of work! And he added also the Node-Red dashboard. So, you get a more-or less complete Node-Red installation.

In addition, he added the SQlite database system and phpadmin light for its administration. Absolutely great!

There is only one thing I can add here: I made a copy of my SD card and make it available to you for download. This saves you all the hassles of downloading and installing dietpi, and wait for the completion of Peters Script. If you want, you can download and copy this file to a SD card using Win32 Disk Imager, insert the card into your Raspby Pi Zero, connect your Zero to a display and a keyboard and enter your Wi-Fi credentials into the file /etc/network/interfaces. That’s all. And the whole thing did take for sure less than half an hour.

If you kept the assigned fixed IP address, you can Putty into a fully configured MQTT broker by connecting to 192.168.0.202.

Just for your information, you find these three users on your system:

|  |  |  |
| --- | --- | --- |
| **Username** | **Password** |  |
| pi | password |  |
| admin | admin | For Node-Red |
| root | dietpi |  |

Of course, you can change the passwords, or add new users, if you wish.

Because we saved so many hours, we still have some time left. So, I will build a small data logger for my Sonoff switch using MQTT, Node-Red, and the SQlite data base. From now on, I assume, that you used my SD card image for your Pi Zero. This image contains all material needed.

So, first, we test the Mosquitto broker. We could do this by using the built-in functions like mosquitto\_sub and mosquitto\_pub in the terminal window. But, because we already have Node-Red available, we use this tool to do that. So, we can also check the connection between the two packages.

We use the “MQTT test flow” in Node-Red to do that. First we have to adjust the IP address of the MQTT broker to the address of your Pi zero, if you changed it. This has to be done in these two nodes. In Node-Red, you have to “deploy” after every change to activate the changes. Now, Node-Red should connect to Mosquitto and you should see the green points here.

If we now press this button, a text is published to Mosquitto. And because we subscribed to the all topics, the message comes back from Mosquitto to Node-Red and into our debug window. So, our “Hello world” scenario is live. Node-Red and Mosquitto are working together. Cool.

Next, we go to the “MQTT data Logger Flow”. This scenario needs three main nodes to perform the task. I added two debug nodes to show you, what happens. Let’s start with the MQTT node. This node is subscribed to the “Relay topic of our Sonoff device from last video. The next node inserts the topic, the payload, and a timestamp to SQlite. The last node creates the connection to the database.

So, Let’s first look at the setup of the database. To do that, we have to connect to phpadminlite. Here, we can manage our database. If you used my image file, you will already find a database called IOT with a few tables. We will only use table Sonoff for this logger. This table has three fields: The topic, a timestamp, and the relay state. If we browse this database, we see, that it is empty.

Now, we go back to Node-Red and to the function which creates the SQL statement which inserts a record into the database.

I have to confess, I have no idea of Java script and I just used an example and adapted it till it did, what I wanted. So, for sure, there are better ways of doing it. But still, it might be useful for you.

The first part creates this line in Node-Red. It shows us, when the last update took place. The next part creates a string called SQL which is filled with an SQL statement. At the end, this string is pushed to the database. If we look at the creation of this string, we see “Insert into Sonoff, which is the table we want to fill. Then, we find all three fields we want to populate. Next, we find the Values, also separated by commas. The string values have to be framed by quotes. If we have a closer look, we see, that we created an ordinary SQL statement. If we go back to our SWL administration, we can execute this statement. And we see the result in the database.

The last node does exactly that: It transfers the SQL statement to the database. That’s all. If we now switch the Sonoff of- and on again, we see two additional SQL statements in Node-Red and the two entries in the database. So, our MQTT logger works!

Summarized:

* We can use our Pi Zero W as a MQTT broker.
* If we use dietpi and Peter Scargill’s script, we easily can create such a device.
* If you use my pre-fabricated SD card image, you are even faster and you get your broker in less than half an hour.
* This broker does not only work as a MQTT broker. It includes also many other pre-installed packets
* Because viewers asked me, how to log MQTT data, I used this newly created device to build a simple MQTT messages broker with SQlite
* All-in all, it is astonishing what we can get for less than 20 dollars and half an hour of work. It does what we wanted and we even can extend its usage .

I hope, this video was useful, or at least interesting for you. If true, then like. Bye

<http://tech.scargill.net/raspberry-pi-zero-wifi/>

vi /var/www/html/phpliteadmin.php

<http://flows.nodered.org/flow/36f2de38ed4bb62cf3a5e8fdc76d779e>

<http://dietpi.local/phpliteadmin>