Experiments in Drip irrigation, Organic Gardening, Water Conservation and Garden Automation

INSTALLING MOSQUITTO MQTT BROKER ON RASPBERRY PI (WITH WEBSOCKETS)

August 20, 2015 · *by* Dan K. · *in Programming, Raspberry Pi · 7 Comments*

This post describes the steps I used to set up the Mosquitto MQTT broker (server) with websockets enabled on a Raspberry Pi .

This can provide two-way communication between Python programs and JavaScript.

For example:

A python program can send a message and have JavaScript update part of a web page in a way that is similar to AJAX.

A button on a web page can cause JavaScript to send a message telling Python to run a function or method.

A JavaScript example: MQTT With Websockets and HTML

Updated November 29 2017
This update includes instructions for installing Mosquitto on a Raspberry Pi running Raspbian Stretch.

Installing from the

Search this site...

RECENT POSTS

- A method for observing the soil Nitrogen cycle in action
- Mesuring Changes in Soil Nitrogen
- Insights from building an MQTT based User Interface
- Building a browser based User Interface with MQTT
- Installing Mosquitto MQTT broker on Raspberry Pi (with websockets)

ARCHIVES

- December 2017
- November 2015
- August 2015

Mosquitto Debian repository

There is a Debian repository with the latest Mosquito binary which has been compiled with websockets enabled out of the box.

At the time if this writing the install from the Mosquitto debian repository has problems that will hopefully be resolved before too long. If you try installing from this source and run into problems you may have better luck building Mosquitto from source. Instructions for that can be found further below.

The first few steps are for adding the Mosquitto Debian repository to your system. They are taken directly from the Mosquitto blog post and are reproduced here (with slight modifications) to keep everything in one place.

To use the new repository you should first import and install the repository package signing key.

```
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

```
sudo apt-key add mosquitto-repo.gpg.key
```

To make the repository available to apt:

```
cd /etc/apt/sources.list.d/
```

Then do one of the following, depending on which version of debian you are using.

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-wheezy.list
```

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-
```

- December 2012
- September 2012
- August 2012

CATEGORIES

- Drip Irrigation
- Instrumentation
- Organic Gardening
- Programming
- Raspberry Pi
- Soil science

```
jessie.list
```

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-
stretch.list
```

Update apt information.

```
sudo apt-get update
```

Install Mosquitto:

```
sudo apt-get install mosquitto
```

If you get an error such as:

```
The following packages have unmet dependencies:
mosquitto : Depends: libssl1.0.0 (>= 1.0.1) but it is
not installable
Depends: libwebsockets3 (>= 1.2) but it is not
installable
```

Then you will need to build Mosquitto from source but don't worry it is fairly painless and because the build process on the Raspberry Pi B2 or B3 can be set to use all 4 processor cores it doesn't take long.

The instructions which follow have been completely updated as of November 29 2017 and have been tested by building the Mosquitto MQTT broker on a Raspberry Pi running Raspbian Stretch but should also work on other Debian systems

Building from source

Prepare a fresh SD card with the latest Raspbian Stretch (optional)

If you will be accessing the Pi over ssh be sure to add a file named ssh to the root directory on the card to enable ssh.

Make any necessary changes to the password and local, time zone, etc with:

```
sudo raspi-config
```

Updtate with:

```
sudo apt-get update
sudo apt-get dist-upgrade
```

Install dependancies

```
sudo apt-get install cmake libssl-dev libwebsockets-dev
uuid-dev libc-ares-dev
```

Make a temporary directory for your build files

```
mkdir temp
```

move into your new directory:

```
cd temp
```

Download and unzip the source files for c-aries

```
wget http://c-ares.haxx.se/download/c-ares-1.12.0.tar.gz
tar -xvf c-ares-1.12.0.tar.gz
```

move into the c-ares directory

```
cd c-ares-1.12.0
```

configure and build c-ares

```
./configure
make -j4
```

The -j4 switch enables building on all 4 processor cores.

Install c-ares

```
sudo make install
```

move back to the temp directory

```
cd ..
```

download and unzip the mosquitto source files

```
wget http://mosquitto.org/files/source/mosquitto-1.4.14.tar.gz
tar xavf mosquitto-1.4.14.tar.gz
```

move into the mosquitto directory

```
cd mosquitto-1.4.14
```

Run cmake with the websockets option

```
cmake -DWITH_WEBSOCKETS=ON .
```

Be sure to include the dot (.) at the end after the space

Run the build and install Mosquitto

```
make -j4
sudo make install
```

Move back to your home directory

```
cd ~
```

Optional: remove the temp directory to free space

```
rm -r temp
```

Add Mosquitto as a user:

```
useradd -r -m -d /var/lib/mosquitto -s /usr/sbin/nologin
-g nogroup mosquitto
```

You should find the mosquitto executable at:

/usr/local/sbin/mosquitto

The config files should be at:

/usr/local/etc/mosquitto/

Enabling websockets

In order to enable websockets you will need to edit the Mosquitto configuration file. The config file should be found located at:

```
/usr/local/etc/mosquitto/mosquitto.conf
```

Edit mosquitto.conf

The mosquitto config file is where most of the settings that control mosquitto’s behavior, including security features, live.

It is necessary to change some settings in order for websockets to operate.

Open the mosquitto config file in your favorite text editor

```
sudo editor /usr/local/etc/mosquitto/mosquitto.conf
```

As you scroll down through the file you will see many settings. I changed the following (mostly by removing the leading # to uncomment the setting):

```
line ~35: pid_file /var/run/mosquitto.pid
```

```
line ~43: user mosquitto
```

The following settings are for the default (Python) client.

```
line ~133: bind_address 0.0.0.0
```

```
line ~136: port 1883
```

```
line ~151: protocol mqtt
```

The following settings under Extra listeners are for the JavaScript client.

```
line ~277: listener 9001 0.0.0.0
```

```
line ~297: protocol websockets
```

Optional:

If you will be using mosquitto to build a web based UI, you can have it also act as a web server for your pages.

```
line ~301: http_dir /usr/share/mosquitto/www
```

Create the directory for the web files:

```
sudo mkdir -p  /usr/share/mosquitto/www
```

Place your HTML, CSS and Javascript files (or a symlink to them) in the www directory.

Note: If there is more that one space between a setting name and it's value, the file won't load.

Test your installation

```
mosquitto -c /usr/local/etc/mosquitto/mosquitto.conf
```

You should see something like:

```
1512012962: mosquitto version 1.4.14 (build date
2017-11-29 15:51:45-0800) starting
1512012962: Config loaded from /usr/local/etc/mosquitto
/mosquitto.conf.
1512012962: Opening websockets listen socket on port
9001.
1512012962: Opening ipv4 listen socket on port 1883.
```

Get and set up systemd unit file from GitHub

The following instructions include the installation of a systemd unit file in place of the older init.d script which is used with Raspbian Wheezy. The older init.d script is included with the Mosquitto installation. While the newer systemd which is used in Raspbian Jessie and Stretch is backward compatible with the init.d script, the systemd unit file is much simpler and more efficient. these steps can also be considered optional.

If you are running Raspbian light you may need to install Git. You can do so with:

```
sudo apt-get install git
```

First disable the init.d script:

```
sudo update-rc.d mosquitto remove
```

Then

```
sudo git clone https://github.com/Dan-in-CA/mosquitto_unit_file.git
```

Note: This unit file contains paths to the Mosquitto executable and config files. Please check to be sure the paths match your system and modify if necessary.

copy the mosquitto.service file

```
sudo cp mosquitto_unit_file/mosquitto.service /etc/systemd/system/mosquitto.service
```

Enable the service

```
sudo systemctl enable mosquitto.service
```

After a reboot you can use:

```
sudo systemctl status | stop | start | restart mosquitto
```

to control the mosquitto daemon. This is extremely useful when editing and testing the mosquitto.conf file.

The systemd unit file also tells mosquitto which config file to use. Otherwise the mosquitto daemon uses defalult settings which won’t work for our intended purpose. You can test mosquitto using the default settings with:

```
mosquitto -h
```

This will tell you the version of mosquitto you are running.

That’s it!

I did not set up any security or other options but it is a good idea to read through the config file to learn what is available.

Notes on usage:

Python and JavaScript MQTT **client software** is required and can be downloaded from the Paho project. See related post Building a browser based User Interface with MQTT

The **Python** client communicates with the mosquitto broker using the default MQTT protocol and port 1883.

The **JavaScript** client communicates with the mosquitto broker using the websockets protocol and a port such as 9001 .

A common mistake is to try and use port 9001 from Python. This will cause Python to hang without raising an exception and can be hard to debug.

Share this:

Twitter Facebook

Loading...

Tags: JavaScript, Python, User Interface, Web technology

DISCLAIMER

The information contained in http://xperimentia.com is for general information purposes only. The information is provided by http://xperimentia.com and its author(s) and while we endeavor to keep the information up to date and correct, we make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to http://xperimentia.com or the information, products, services, or related graphics contained on http://xperimentia.com for any purpose. Any reliance you place on such information is therefore strictly at your own risk.

In no event will we be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from loss of data or profits arising out of, or in connection with, the use of http://xperimentia.com its videos, pictures, podcasts, software and any other media or information of any kind.

Through http://xperimentia.com you are able to link to other websites which are not under the control of http://xperimentia.com. We have no control over the nature, content and availability of those sites. The inclusion of any links does not necessarily imply a recommendation or endorse the views expressed within them.

7 comments

Le T · November 29, 2017 - 12:45 pm · *Reply→*

Thank you this blog post. I found it very helpful.

However, I ran into the following error when adding "protocol" (line 137) in my mosquitto.conf file in /etc/mosquitto/conf.d

```
● mosquitto.service – Mosquitto MQTT Broker daemon
Loaded: loaded (/etc/systemd/system
/mosquitto.service; enabled)
Active: activating (auto-restart) (Result: exit-code) since
Wed 2017-11-29 15:37:14 EST; 1s ago
Process: 1151 ExecStart=/usr/sbin/mosquitto -c
/etc/mosquitto/conf.d/mosquitto.conf -d (code=exited,
status=3)
```

```
Nov 29 15:37:14 MQTTBROKER mosquitto[1151]: Error
found at /etc/mosquitto/conf.d/mosquitto.conf:137.
```

```
Nov 29 15:37:14 MQTTBROKER mosquitto[1151]: Error: Unable to open configuration file.
Nov 29 15:37:14 MQTTBROKER systemd[1]: mosquitto.service: control process exited, code=exited status=3
Nov 29 15:37:14 MQTTBROKER systemd[1]: Failed to start Mosquitto MQTT Broker daemon.
Nov 29 15:37:14 MQTTBROKER systemd[1]: Unit mosquitto.service entered failed state.
```

It is worth noticing that the broker works fine when starting mosquitto broker from the same config file manually using:

```
mosquitto -v -c /etc/mosquitto/conf.d/mosquitto.conf
```

The BROKER daemon also runs fine if I remove "protocol" from my local config file.

Do you have any insight on why perhaps this is the case?

Dan K. · November 29, 2017 - 6:55 pm · *Reply→*

The post has been updated as of November 29 2017.
Let me know if the new instructions are not working.

Gs70 · October 29, 2016 - 3:36 pm · *Reply→*

This following did not work:

```
wget http://git.warmcat.com/cgi-bin/cgit/libwebsockets/snapshot/libwebsockets-1.3-chrome37-firefox30.tar.gz
```

it ended up in a loop reporting errors…

I am assuming there is an update. (what appears on the libwebsockets website is version 2.1). But not sure how to reconfigure the wget statement to download…

I am new to the Linux/RaspPi world so it would be great if there was an update. to this..

BTW:
Thanks for the instructions… I am able to get mosquito installed and communicating via mosquito_pub and mosquitto_sub and python scripts.. But unable to get it to communicate to a Node.Js or any other HTML page for that matter…

It would be nice to see a turtorial showing from a fresh image of NOOBS v2_0_0 how to install and setup the Raspberry pi so that mosquito and websockets would work on a local intranet (server being the Raspberry Pi) serving a web page that showed messages from an MQTT broker provided by the Raspberry Pi.. Much of the information I have been able to find is outdated and does not work..

Thanks in advance…

Dan K. · October 30, 2016 - 8:53 am · *Reply→*

Thanks for reporting this problem.
I will work through the process and update the post shortly.

The post “Insights from building an MQTT based User Interface” has some information about running mosquitto as a web server on a Raspberry Pi. During the time since I wrote that post I have developed a more efficient method of communication between JavaScript and a Python program running on the Pi. I will be posting about that in the next few weeks.

Dan

Edit: The post has been updated. Let me know if you are still having problems getting it to work.

Dan K. · November 28, 2015 - 12:56 pm · *Reply→*

This problem resulted from using an older style init script on Raspbian Jessie. The solution was to write a systemd unit file as described in the follow up post "Insights from building an MQTT based User Interface"

Zack · September 22, 2015 - 9:58 am · *Reply→*

also mosquitto_pub and mosquitto_sub don't work after following these steps. You'd need to copy libmosquittopp.so.1 to /lib directory:
cp /usr/local/lib/libmosquittopp.so.1 /lib
and also:
cp /usr/local/lib/libcares.so.2 /lib

Zack · September 22, 2015 - 9:55 am · *Reply→*

Init process didn't work for me.
When I'd try to stop and start the service; it did work, so I decided to change couple of things:
1)change init default: "update-rc.d mosquitto disable"
2)add cron job to start this service after one minute each reboot:
"sudo crontab -e"
add the this line to the bottom:
"@reboot sleep 60 && sudo /etc/init.d/mosquitto start &"

Leave a Reply

Enter your comment here...

← Setting up a Raspberry Pi to work with OpenSprinkler (Old post)

Building a browser based User Interface with MQTT →