

# Primal Cortex's Weblog

*Amnesia sets in...*

[Home](#)   [About](#)

## Tag Cloud

[akonadi](#) [arch](#)  
[linux](#) [arduino](#)  
[asus](#) [ati](#) [avchd](#)  
[Azores](#) [Açores](#)  
[backup](#) [canon](#)  
[casa](#) [client](#)  
[cloud](#) [codes](#)  
[configuration](#)  
[debug](#) [dolphin](#)  
[drivers](#) [dropbox](#)  
[duplex](#) [casa](#)  
[error](#)  
[esp8266](#)  
[filenet](#) [firefox](#)  
[firewall](#) [grub](#)  
[hf200](#) [i7](#) [IBM](#)  
[imap](#) [IoT](#) [java](#)  
[jboss](#) [kde](#)  
[kubuntu](#)  
[LDAP](#) [Linux](#)  
[lua](#) [mailscanner](#)  
[Micro\\$oft](#)  
[MQTT](#)

## Setting up an IoT framework/dashboard with NodeRed, Mosca/Mosquitto and Freeboard.io dashboard

Posted on [February 25, 2015](#) by [primalcortex](#)

With the low cost of the esp8266 chip and boards, it is now easy and affordable to add connectivity to almost anything that we can think of. Still if we build an Internet of Things device by enabling it with the esp8266 chip to connect to the internet, or other networks, what we can do with the data that we gather? Basically we do what we do with any data that we get, store it, process it and show it, and that's what we are going to see on this post.

The software (running on Linux platform) that I'll use is the following:

**IBM NodeRed:** (<http://nodered.org/>) This great software will allow us to get data, process it, store and publish it, with a workflow based designer. It runs under Node-Js Javascript

máquina  
netbeans  
node-red  
nodemcu  
nodered nslu2  
odroid p8  
photography  
Portugal  
postfix problem  
proxy pyqt  
python pão  
rtlSdr security  
sigfox slow  
SMC sming  
SQL ssh sync  
synology  
ubuntu  
upgrade visio  
vmware  
VODAFONE  
webcam  
Windows  
x-windows



based server engine.

**MQTT Broker Mosca:** (<http://www.mosca.io/>) There are several MQTT brokers available, but this one runs on Node-Js and so is platform agnostic, and runs where Node-Js runs. Also it has websocket support out of the box, which allows to use the Freboard.io dashboard in a straight foward way.

**Freeboard.io Dashboard:** (<http://freeboard.io/>) Is a great dashboard and freely available at the project GitHub page (<https://github.com/Freeboard/freeboard>). It allows to design and store several dashboards with some cool widgets.

### Step 1: Setting up NodeRed:

First of all make sure that you have node and npm (the node package manager) installed on your distribution. You can see that with the commands **node -v** and **npm -v** for checking out their versions. In my case, and for now, it's version v0.10.33 for nodejs and 1.4.28 for npm.

First I've created a directory named **IoTServer** where I'll keep all software and configurations.

Download the zip file from nodered GitHub page <https://github.com/node-red/node-red> or just clone the repository. I clone the repository, because it's simpler in the future to update it with just **git pull**. First just make sure that you have the icu/libicu libraries installed.

```
cd ~  
mkdir IoTServer  
cd IoTServer
```

Follow

```
git clone https://github.com/node-red/node-red
cd node-red
npm install
```

We can now run the nodered server with the command **node red.js -v**. I use -v at the end to see if any error crops up and also to see if any node modules are missing, and need to be installed. For example:

```
pcortex@cloudsrv:~/IoTServer/node-red$ node red.js -v
Welcome to Node-RED
=====

24 Feb 11:35:16 - [info] Node-RED version: v0.10.3.0
24 Feb 11:35:16 - [info] Node.js version: v0.10.33
24 Feb 11:35:16 - [info] Loading palette nodes
24 Feb 11:35:59 - [warn] -----
24 Feb 11:35:59 - [warn] [arduino] Error: Cannot find module 'arduino'
24 Feb 11:35:59 - [warn] [rpi-gpio] Info : Ignoring rpi-gpio module
24 Feb 11:35:59 - [warn] [redisout] Error: Cannot find module 'redisout'
24 Feb 11:35:59 - [warn] [mongodb] Error: Cannot find module 'mongodb'
24 Feb 11:35:59 - [warn] -----
24 Feb 11:35:59 - [info] Server now running at http://localhost:1880/
24 Feb 11:35:59 - [info] Flows file not found : flows_clouds.json
24 Feb 11:35:59 - [info] Starting flows
```

As we can see the node modules arduino, rpi-gpio, redisout and mongodb are missing. If we are going to use them, we need to install them with the npm tool. Also the following message: *Flows file not found : flows\_clouds.json* informs us that no saved nodered workflows were found. We can now access the nodered at adress <http://localhost:1880/>. If we want to change the bind address or add authentication,

Follow

we should change the settings.js file. Take a look at <http://nodered.org/docs/configuration.html> for more information, and start nodered with: **node red.js -v -s settings.js**

We can now start designing our nodered workflows that receive data from the esp8266 via MQTT protocol or even by simple REST based HTTP protocol, process it and store the data payload, if any.

For storing data into a database, we can use MongoDB, or mysql, for example, installing the needed modules.

For mysql we can do:

```
cd ~/IoTServer/node-red
npm install bignumber.js require-all readable-stream
npm install mysql
npm install node-red-node-mysql
```

And we need to restart nodered. The mysql node should be available now in storage pallet.

## Step 2: Installing Mosca MQTT broker (Alternative 1 for the MQTT Broker)

We can use several brokers, but since we already are running nodered on node-js, we will use Mosca. Both nodered and Mosca can provide a websocked based interface. This is of interest because we can use browser based websocket applications, like freeboard.io, to consume data from our IoT devices.

Follow



```
cd ~  
sudo -s  
sudo npm install mosca bunyan -g
```

And to run it just do: **mosca -v | bunyan**

For running it with websockets enabled, on TCP port 3000, just do:

```
mosca -v --http-port 3000 --http-bundle --http-stat:
```

Check out the documentation for more information at:

<https://github.com/mcollina/mosca/wiki/MQTT-over-Websockets>

## Step 2: Mosquitto MQTT broker with Websockets support (Alternative 2 for the MQTT Broker)

If having trouble installing or using Mosca, we can use the latest Mosquitto version 1.4 that has websockets support.

For that we need to make sure that cmake and uuid/uuid-dev are installed (For example: `sudo apt-get install cmake uuid uuid-dev`).

Also we need to download the libwesockets library, compile it and install it:

```
cd ~/IoTServer  
git clone git://git.libwebsockets.org/libwebsockets  
cd libwebsockets
```

Follow

```
mkdir build
cd build
cmake ..
make
sudo make install
```

We then need to download and compile Mosquitto 1.4:

```
cd ~/IoTServer
wget http://mosquitto.org/files/source/mosquitto-1.4
tar xvzf mosquitto-1.4.tar.gz
cd mosquitto-1.4
```

We need to edit the **config.mk** file and change the option  
**WITH\_WEBSOCKETS:=no** to  
**WITH\_WEBSOCKETS:=yes**

And finally:

```
make
sudo make install
```

To activate websocket support we need to edit the  
mosquitto.conf file located at /etc/mosquitto or some other  
directory and add the following lines, for example at the end  
of the file:

```
listener 1883

listener 9001 127.0.0.1
protocol websockets
```

Then we run it with:

Follow

```
pcortex@cloudsrv:~/IoTServer$ mosquitto -c /etc/mosca
```

```
1424790588: mosquitto version 1.4 (build date 2015-0
1424790588: Config loaded from /etc/mosquitto/mosqu
1424790588: Opening ipv4 listen socket on port 1883
1424790588: Opening ipv6 listen socket on port 1883
1424790588: Opening websockets listen socket on port
```

### Step 3: Installing freeboard.io dashboard

We are almost at the end of this long post. For our infrastructure we need now to install the dashboard that will allow us to see the data in the browser.

```
cd ~/IoTServer
git clone https://github.com/Freeboard/freeboard.git
```

For now I'll just use apache to serve the freeboard dashboards that is installed in my server. Because freeboard is not installed on the root web directory we need to make the freeboard available to Apache. The easiest way to do this, as long as Apache has the FollowSymLinks option enabled for the document root, is to create a link to the freeboard directory on the web document root:

```
sudo -s
cd /var/www
ln -s /home/pcortex/IoTServer/freeboard iot
```

And now freeboard is available at the url

Follow

<http://myserveraddress/iot>.

We need now, and finally to add MQTT over Websockets support to freeboard... We are almost there. This post <http://jpmens.net/2014/11/12/freeboard-a-versatile-dashboard/> shows how it's done but I'll repeat it here again:

1st: Download the development version of mqtt.js here: (<http://git.eclipse.org/c/paho/org.eclipse.paho.mqtt/javascript.git/plain/src/mqttws31.js?h=develop>) and save it:

```
cd ~/IoTServer/freeboard/plugins
mkdir mqtt
cd mqtt
wget --output-document mqttws31.js http://git.eclips
```

We will now download the freeboard MQTT plugin:

```
cd ~/IoTServer
git clone https://github.com/alsm/freeboard-mqtt
cd freeboard/plugins/mqtt
cp ~/IoTServer/freeboard-mqtt/paho.mqtt.plugin.js .
```

We need to edit the paho.mqtt.plugin.js file and change the line:

```
"external_scripts" : [
    "<full address of the paho r
    ],
```

Follow



to

```
"external_scripts" : [  
    "plugins/mqtt/mqttws31.js"  
],
```

and finally we need to change the **index.html** file from:

```
<scripttype="text/javascript">  
    head.js("js/freeboard+plugins.min.js",  
        // *** Load more plugins here ***  
        function() {
```

to

```
<scripttype="text/javascript">  
    head.js("js/freeboard+plugins.min.js",  
        "plugins/mqtt/paho.mqtt.plugin.js",  
        // *** Load more plugins here ***  
        function() {
```

That's it. We are finally ready. All infrastructure is done. We need now just to configure it.

### **Final step: Testing it out with mosquitto websockets!:**

Let's start Mosquitto: **nohup mosquitto -c /etc/mosquitto/mosquitto.conf &**

And check if the WS protocol is active:

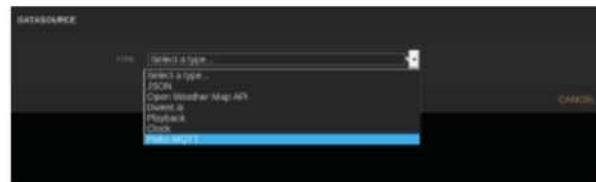
Follow

```
netstat -nap | grep 9001
```

```
tcp        0      0 0.0.0.0:9001          0.0.0.0:*
```

Setting up Freeboard.io: It's very important to notice that the Dashboard will run on our browser, making in fact our browser the client. This means that when we configure freeboard, we must take notice of this when filling up the MQTT broker address. It will be **localhost** if the browser **AND** the broker runs on the same machine, but for the dashboard to work anywhere we should use the public ip of the machine running the broker:

Access the freeboard.io dashboard, and select Datasources -> Add. From the dropdown box, we select the Paho MQTT provider.



We need to configure the data source as follows:



Change the MQTT server address to something that ma

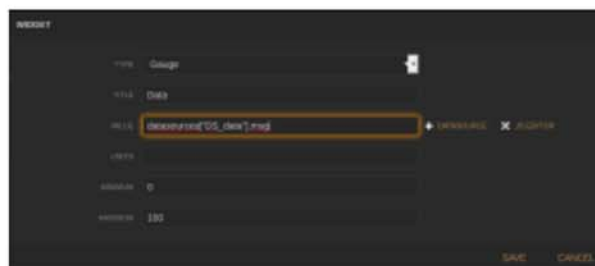
Follow

sense in your config.

For Mosquitto with websockets enabled, the port is 9001, and for Mosca, the port is 3000, but of course this can change, depending of the configuration used.

We can now add a Panel and a gauge to see data from the topic **/data** that we defined on the above datasource.

So, **Add Pane**, and on the new Pane select the + (plus) to add a widget. In our case we will select **Gauge**. We fill out the required info and on the Value field we press Datasource, and select the newly previous created datasource named **Data** and because we are not using JSON for data, we just append .msg at the end. So something like **datasource["DS\_data"].msg** should be written.



We press Save, and that's it.

We can now publish into the topic, and the dashboard should update:



For example, in this case: **mosquitto\_pub -t /data -m 23**

Follow

That's it. It works fine with Mosquitto 1.4 with websockets enabled and with the Mosca MQTT broker.

### Final setup:

We should make our dashboard permanent and for doing so we need to save the dashboard locally as a JSON file, and move it to the server running the dashboard to the root directory of freeboard.

After that we call directly the saved dashboard with the following URL:

**[http://myserveraddress  
/iot/index.html?load=dashboard.json](http://myserveraddress/iot/index.html?load=dashboard.json)**

**Edit:** For Websocket support on the Freeboard.io Dashboard, please checkout [Freeboard and WebSockets](#)

**Warning:** The current freeboard.io project at Github (as of 12/2015) has problems saving dashboards when using Firefox. For saving dashboards use Chrome/Chromium.

[About these ads](#)



The advertisement is a rectangular banner. On the left, there's a close-up of a violin. Overlaid on the violin is the text 'Create an unforgettable website' in a large, white, sans-serif font. Below this text is a blue button with the white text 'Get Started'. At the bottom left of the banner is the WordPress logo and the text 'WordPress.com'. On the right side of the banner, there's a white smartphone. The screen of the phone displays a website with a dark header that says 'VIOLINS', a large image of a violin, and some text below it. The background of the entire banner is a blurred image of a violin.

Follow



One blogger likes this.

#### Related

IoT framework:  
Using NodeRed.  
In "All about all"

ESP8266 - Logging  
data in a backend  
with end to end  
encryption  
In "All about all"

SparkFun Phant  
server data stream  
graphing  
In "All about all"

This entry was posted in [All about all](#) and tagged [esp8266](#), [IoT](#), [Linux](#), [mosca](#), [mosquitto](#), [MQTT](#). Bookmark the [permalink](#).

← MQTT Publishing on the ESP8266

IKEA Fredde desk →

## 32 thoughts on “Setting up an IoT framework/dashboard with NodeRed, Mosca/Mosquitto and Freeboard.io dashboard”

Pingback: [IoT framework: Using NodeRed. | Primal Cortex's Weblog](#)



Руслан И. says:

May 13, 2015 at 4:31 am

Great tutorial! I've followed your instructions. I've used mosquitto as mqtt broker. I could send data to freeboard with `mosquitto_pub -t /data -m 23`, but it is not displayed. freeboard says that it received a message. mosquitto says that a client was connected.

Follow

Maybe I should convert data to number?

[Reply](#)



[Руслан И.](#) says:

May 14, 2015 at 3:08 am

I've changed datasource["data"].msg to  
datasource["DS\_data"].msg and it worked!

[Reply](#)



primalcortex says:

May 14, 2015 at 5:53 pm

Thanks for the feedback. It is probably  
something that is different with your  
freeboard.io version, not sure. I'll have a go  
with the latest version to check the behaviour.



[Руслан И.](#) says:

May 14, 2015 at 8:17 pm

Thank you for reply.  
today I couldn't get it working even with the "DS\_data".  
Some data is coming to freeboard, because I see that  
'last updated' time changes. Problem is that gauge is  
always 0. Is there a way to debug?

[Reply](#)



[Руслан И.](#) says:

[Follow](#)





May 14, 2015 at 8:23 pm

Problem is solved: I changed  
datasource["DS\_data"].msg to datasource["DS\_data"].  
["msg"]

[Reply](#)



Руслан И. says:

May 16, 2015 at 7:01 pm

A remote controller publishes three topics to mosquitto  
broker: /board/temperature, /board/potentiometer,  
/board/soundlevel.

In "the topic to subscribe to" I put "/board/#".

In gauge setting I wrote datasources["data"]["msg"].

Now all three topics are displayed in the same gauge at  
the same time.

I want to display received data in three separate  
gauges. How to do this from single data source?

Thank you for your support.

[Reply](#)



primalcortex says:

May 17, 2015 at 9:04 pm

Instead of publishing a simple value like 20,  
publish a JSON structure. Then you can access  
the members with different values, using the same  
datasource.

[Reply](#)

[Follow](#)



[peterretief](#) says:

[May 23, 2015 at 3:46 pm](#)

Thanks for the tutorial, I have temperature and humidity in the gauge pane working perfectly!! Only thing is I cant get it to load the json settings – I am going to check my apache configs, sure its something there

[Reply](#)



[Neil Franken](#) says:

[June 8, 2015 at 4:23 am](#)

I might be asking a stupid question here but how does Node-Red fit into this?

[Reply](#)



[primalcortex](#) says:

[June 8, 2015 at 8:30 am](#)

Hi: There are no stupid questions 😊

Regarding you question, the main issue, regarding at least the ESP8266, is that NodeRed is used to convert the MQTT protocol to the WebSockets protocol used by the freeboard.io. That is one of the purposes, but other purposes are also for the freeboard.io isolation, that means that fb.io communicates with a server (NodeRed) and not volatile devices that come and go. And also NodeRed can be used to add logic and calculation for the input data before publishing it to freeboard. But of course, NodeRed is not mandatory.

[Follow](#)

I hope that the answer clarified your question.

[Reply](#)



Nuttapong says:

[July 20, 2015 at 4:08 pm](#)

I do encounter a strange behavior of Freeboard on my local server, it allows only 1 data source, once I have finish adding the second datasource the first one stop update, by the way I'm using freeboard with nginx and I have to specify the user "root" in the nginx.conf file.

Any Idea what has happen?

[Reply](#)



primalcortex says:

[July 20, 2015 at 6:48 pm](#)

Hi:

Sorry, no idea 😊 But I believe that I had the same issue, with Apache, and saving the Dashboard and accessing it directly after a restart of the web server did the trick.

Not sure.

I had my freeboard dashboard running on hosted VPS, but the company went bust and I've lost my dashboards. So right now I have no way to easily test.

Ohh, also after adding the second datasource close the browser and try again, it might help.

[Follow](#)

[Reply](#)



Nuttapong says:

[July 21, 2015 at 7:10 am](#)

Thanks will try your trick... No wonder why I could not get thru the first landing page of freeboard.io and nobody at freeboard seem to read my email.

Rgds

Nuttapong

---

Pingback: [Frameworks | Redlab](#)



pfe says:

[August 12, 2015 at 7:44 am](#)

hello

Great idea but I can't follow your install instructions on windows!

Please can you give the instructions for windows?

Thanks a lot

[Reply](#)



primalcortex says:

[August 12, 2015 at 9:10 am](#)

Hi:

Sorry, I don't own or have access to any Microsoft

[Follow](#)

Windows based machine, so I'm afraid I can't help you in setting this up on Windows.

[Reply](#)



albertvanwessel says:

August 20, 2015 at 2:42 pm

i have followed your tutorial to the letter but my mqtt datasource is never updated. When i test the websocket with the hivemq webclient which is using the same mqttws31.js everything works any tips

[Reply](#)



primalcortex says:

August 20, 2015 at 6:33 pm

the freeboard.io dashboard can be a bit unreliable regarding the datasources. The datasource must be created from scratch without any errors. What I mean is that if we change something on the datasource afterwards it stops working. Can you create the datasource from scratch?

[Reply](#)



clausqr says:

October 4, 2015 at 5:55 pm

If you get the “/usr/bin/env: node: No such file or directory” error when running “mosca -v --http-port 3000 --http-bundle --http-static ./ | bunyan” just add a

[Follow](#)

symlink to the actual binary

“`

```
ln -s /usr/bin/nodejs /usr/local/bin/node
```

“`

[Reply](#)



roadfox says:

October 5, 2015 at 9:18 am

hi

i'm trying to follow your description. do i understand this right that about in the middle you switched from mosca to mosquitto?

i think i have mosca up and running, but i'm not shure how to verify, is there also a command like "mosquitto\_pub" to send a message to my mosca install?

or would you recommend to install msquitto anyway? in meantime i'll try to send a message with node-red, many thanks

[Reply](#)



primalcortex says:

October 5, 2015 at 9:49 am

Hi. Yes, sorry for the confusion. At first I've tried to build everything based on NodeJS/Javascript, where Mosca fits better due to the fact it uses the same technology.

But I had trouble making it work on my Synology NAS due to several issues, and changed to Mosquitto where there is a package for the

[Follow](#)



Synology. In my work PC, Mosca works fine. Still for the freeboard dashboard either Mosca or Mosquitto can be used.

Anyway, I'm not aware of any Mosca specific tools for publishing MQTT topics.

What I recomend is to download and use the MQTT-SPY tool, and use the tool to connect to the broker and use it's graphical interface to publish and receive the topics.

[Reply](#)



[roadfox](#) says:

October 5, 2015 at 10:29 am

Many thanks for your reply. Makes thing clearer to my.

Meanwhile im succesfully in sending data from node-red to the mqtt, with an mqtt out module, broker = localhost:1883 and topic = /data.

In freeboard i created a data source as per your description which is working too (last updatet time changing when i send in node-red.

This is very cool, and i have to say thanks again for describing this setup.

Now i'm strungling on the gauage, when i add a text widget, i see that i have the follwing:

`datasource = datasources["data"]["msg"]` or

`datasources["data"].msg` (same result)

text widget output =

`{"direction":274,"speed":3.4}`

so i tryed to address the single values with:

`datasources["data"]["msg"].direction` but that's

[Follow](#)

not working, do you have maybe any idea how to solve this?



primalcortex says:

October 5, 2015 at 11:01 am

If the msg is {"direction":274,"speed":3.4}, then it's a JSON object. So I think that the correct expression should be something like  
datasource = datasources["data"]["msg"]  
.speed and  
datasource = datasources["data"]["msg"]  
.direction



roadfox says:

October 5, 2015 at 2:21 pm

sorry i did overlook that there is an "js messages" option on the datasorce prpoerties page. now everithing is working.  
what i don't like at the moment that on freeboard reload all the graphs start at zero.

[Reply](#)

---

Follow



primalcortex says:

October 5, 2015 at 2:40 pm

I don't know the code base for freeboard, but that it's normal, at start up time it has no data...

[Reply](#)



Priti says:

November 24, 2015 at 10:27 am

Hello,  
I am not able to find paho MQTT option in datasources.  
Can anybody help me regarding this?

[Reply](#)



primalcortex says:

November 24, 2015 at 10:40 am

Hi: Did you install and configured the MQTT plugin? You need to add the plugin to the freeboard plugins list.

[Reply](#)



Priti says:

November 24, 2015 at 10:52 am

Thanks for your replay  
I will go through the suggested procedure

[Follow](#)

[Reply](#)



Priti says:

[November 28, 2015 at 6:10 am](#)

Hi there>>!

I am having problem while executing mosquitto.conf  
file

I have added

listener 1883

listener 9001 127.0.0.1  
protocol websockets

when I fire the command mosquitto -c /etc/mosquitto  
/mosquitto.conf  
it is showing error unknown configuration variable  
'protocol'.

[Reply](#)

[Follow](#)



primalcortex says:

November 28, 2015 at 5:56 pm

What version of the Mosquito broker are you using?

This only works for version 1.4 or above for the mosquito broker.

You can see what version you are running with the command: `mosquitto -v`

[Reply](#)

---

Pingback: [MQTT Dashboards | Primal Cortex's Weblog](#)

## Leave a Reply

[Create a free website or blog at WordPress.com.](#) | [The Able Theme.](#)

---

Follow