Indoor Wellbeing Solution Creating an end-to-end IoT solution

In part 2 of this workshop we will create an IoT node which will be connected to our TTN node as build in part 1 and connect them with MQTT / NodeRed.

Prepare the ESP Wemos modules

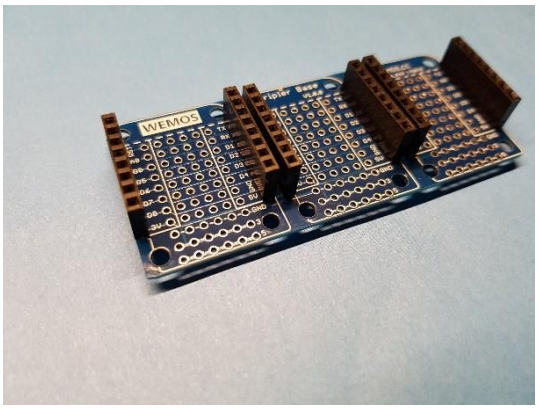
Parts needed:

- Wemos Dual Base
- Wemos D1 mini v3 – ESP8266
- Wemos OLED shield
- Servo SG90 mini servo
- 3 male-male jumper wires

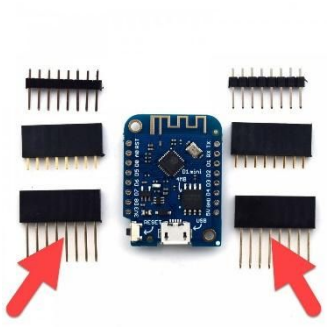
Soldering

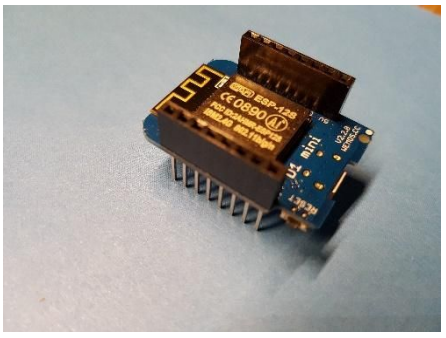
If your kit is not pre-soldered we have to solder the headers first. Beware of the pin numbers and orientation. Each Wemos board has a small notch on the left.

The Dual Base has four female headers to solder, put them on the front side:

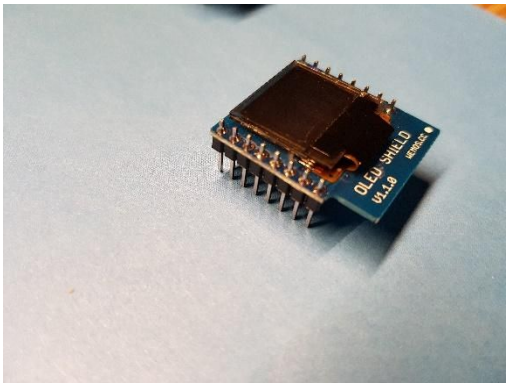


The Wemos D1 mini v3 – ESP8266 comes with three possible options. For experimental purposes use the longer female connectors.

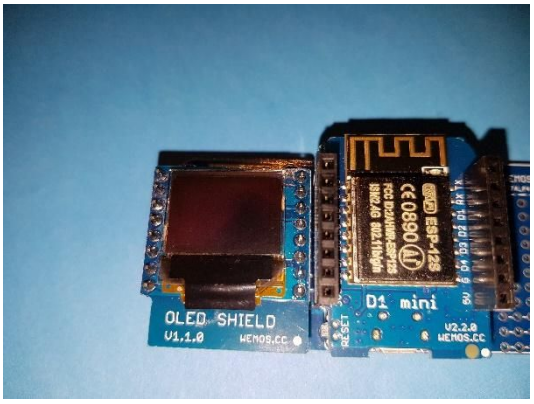




The Oled display must be always ‘on top’, so solder the connectors on the backside, display in front.



Now we can create our test configuration: Wemos and OLED: You can stack them (align the notches) or put them next to each other on the Dual Base:



Arduino Software

To use the ESP8266 we need the ESP toolchain. Installation instructions can be found here: http://esp8266.github.io/Arduino/versions/2.0.0/doc/installing.html

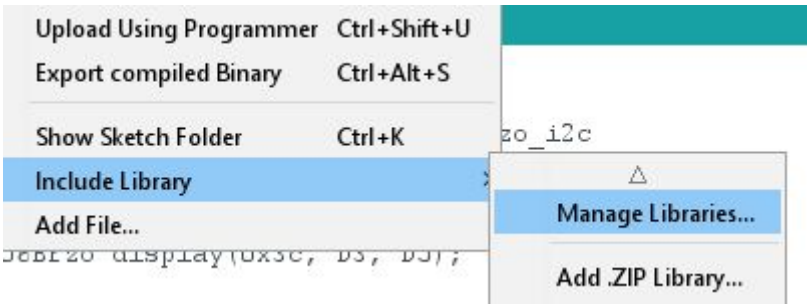
Installing with Boards Manager

Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).

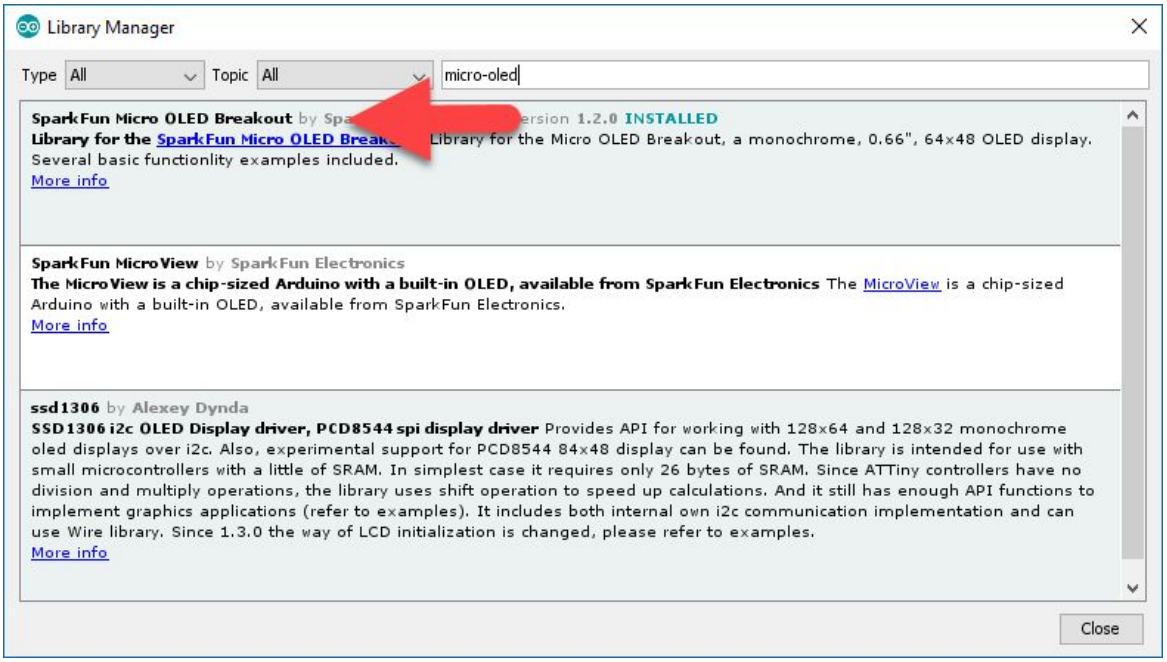
- Install the current upstream Arduino IDE at the 1.8 level or later. The current version is at the Arduino website.
- Start Arduino and open Preferences window.
- Enter `http://arduino.esp8266.com/stable/package_esp8266com_index.json` into *Additional Board Manager URLs* field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install *esp8266* platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

After installation we have to add three libraries we want to use, OLED, MQTT and JSON:

Go to Sketch, Include Library, Manage Libraries:



In the library manager search for ‘micro oled’ and find the ‘SparkFun Micro OLED Breakout’ and install this library:



Install in the same interface the following libraries:

- PubSubClient
- ArduinoJson

Connect your Wemos with a micro USB data cable, choose the right COM port and board settings:

These are default settings, except for the upload speed (you might choose 921600 to speed up the upload process). The COM port might differ!

Get the sketch wbs_ESP8266_MQTT_OLED and fill in the WiFi credentials in the code (ssid and password):

```
// WiFi Credentials
const char* ssid = "";
const char* password = "";
// MQTT Credentials
const char* mqtt_server = "broker.hivemq.com";
const char* mqtt_username = "";
const char* mqtt_password = "";
const int   mqtt_port=1883;
```

you might want to change the mqtt parameters if you want to use your own MQTT server.

Upload the code, and see what happens. Both the OLED and logging (115200 baud) will show you information. Save the 'ClientID' for later: WIFI-Display-XXXXXXXX.

This node is now waiting for MQTT messages on the given channel. We have to fill the channel and will use Node-Red for that.

Prepare NodeRed

To send your data to MQTT we can use 'NodeRed'. You can use a local instance of NodeRed on your computers or Raspberry PI, it is also possible to install it as a Cloud Service on IBM Cloud

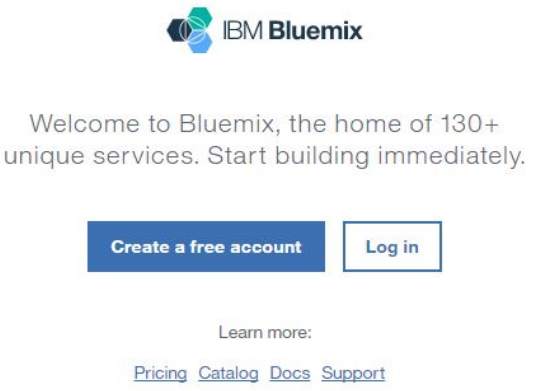
IBM Cloud (formerly: IBM BlueMix)

IBM Cloud is IBM's Cloud platform offering almost every service or application you need. Why use IBM?

- It gives a complete overview of a modern 'Cloud Service' (like Azure, Amazons AWS or BlueMix)
- It is free for 30 days, after that you need to register your Credit Card but it stays free for hobby use (<20 nodes, low traffic and memory).
- You can use Node Red running 'always' in the Cloud without bothering powering off your laptop or Raspberry PI. You get a 512MB server instance for 'free'.
- You can use it in a workshop 😊
- There are well documented examples of connecting 'things'.

Registration

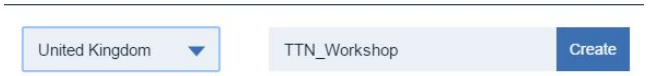
Go to www.bluemix.net

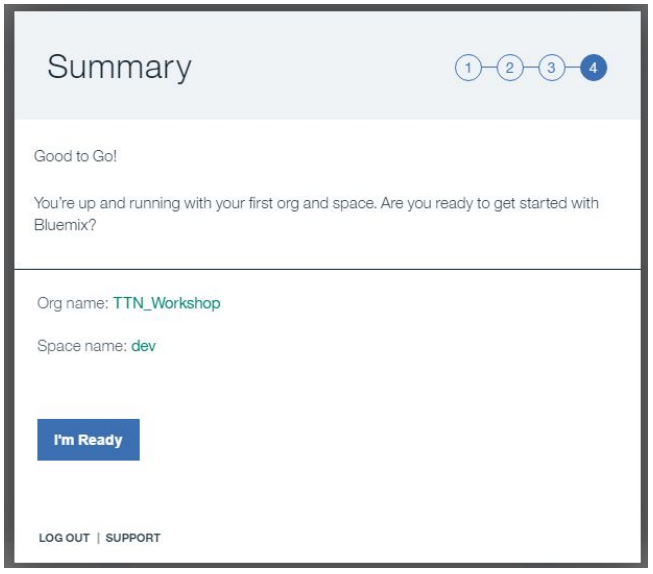


- 'Create a free account'
- Fill in the form, using a valid email address.
- You will receive a confirmation email shortly
- Confirm the BlueMix account email
- Now Log In to the service
- It will last some time for first-time setup
- You have to agree some stuff

- 

- Fill in an Organization name (‘TTN_Workshop’ is a nice name, but already in use) and select United Kingdom

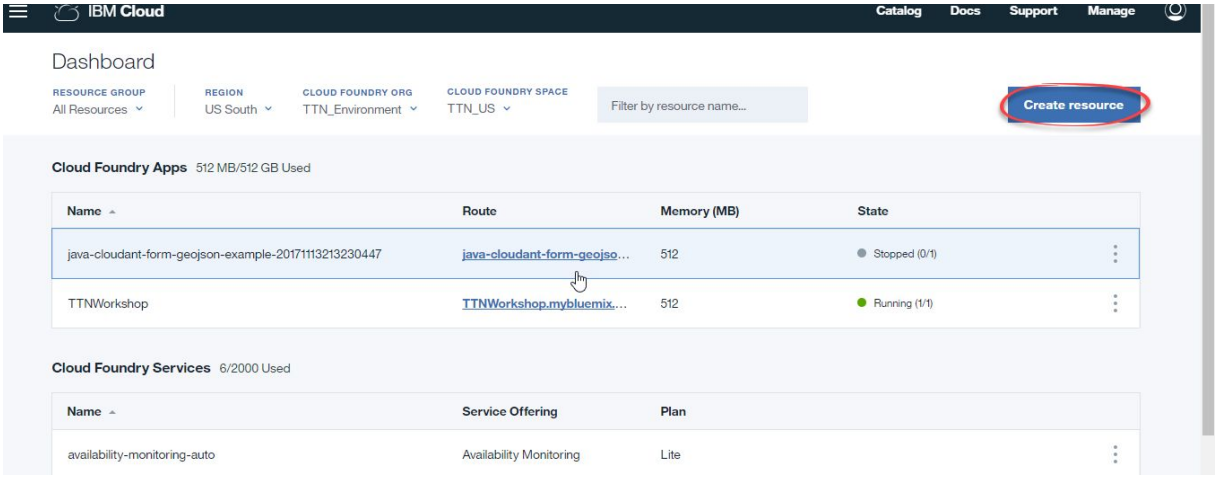


- Name your space like ‘DEV’
- 

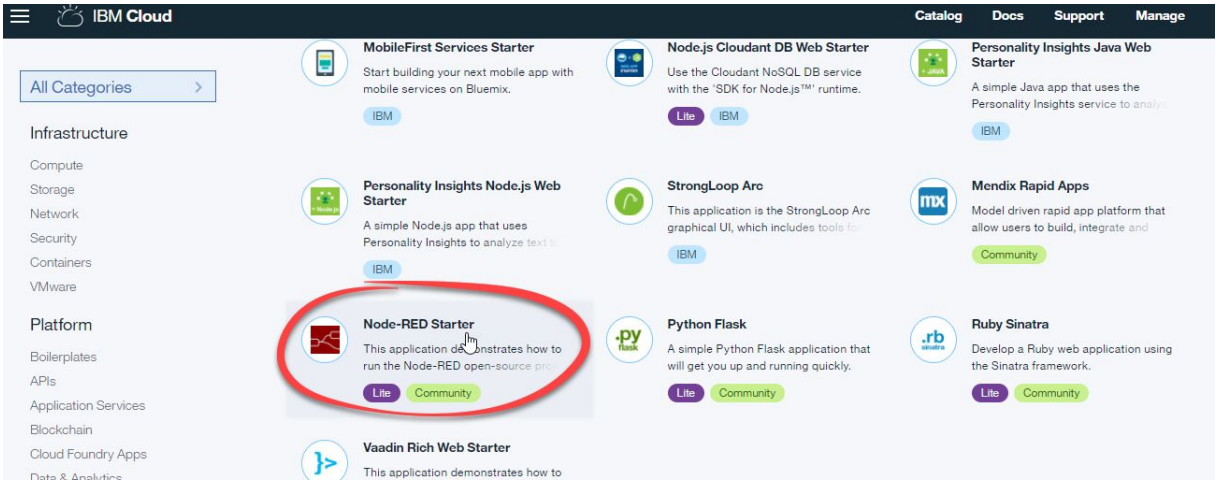
- Ready to start!

Create an App in BlueMix

An App is a functionality installed on the BlueMix platform. In the main dashboard choose ‘Create Resource’.

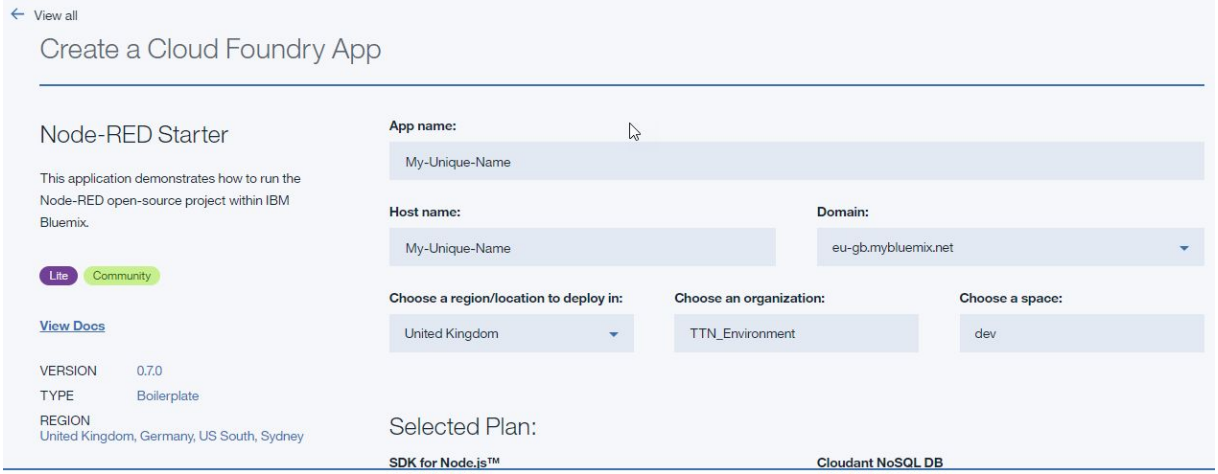


In the next screen we choose ‘Node-Red Starter’



Select ‘Node-RED Starter’:

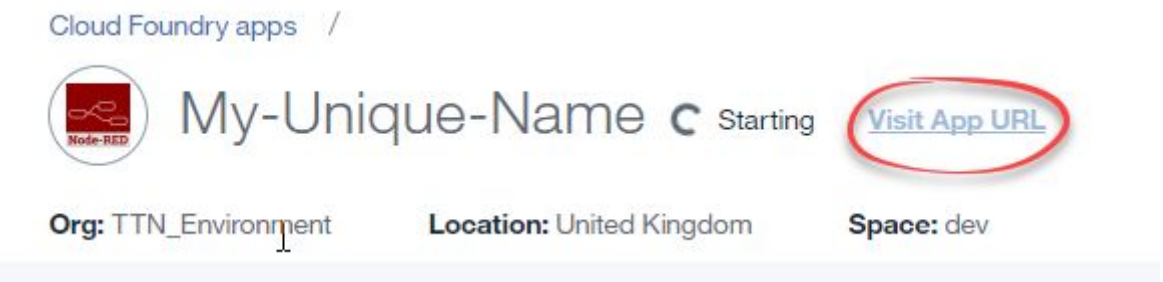
Now you need a hostname for your Node-Red server instance, so this should be globally unique! The App name is just unique for your own environment. Be creative!



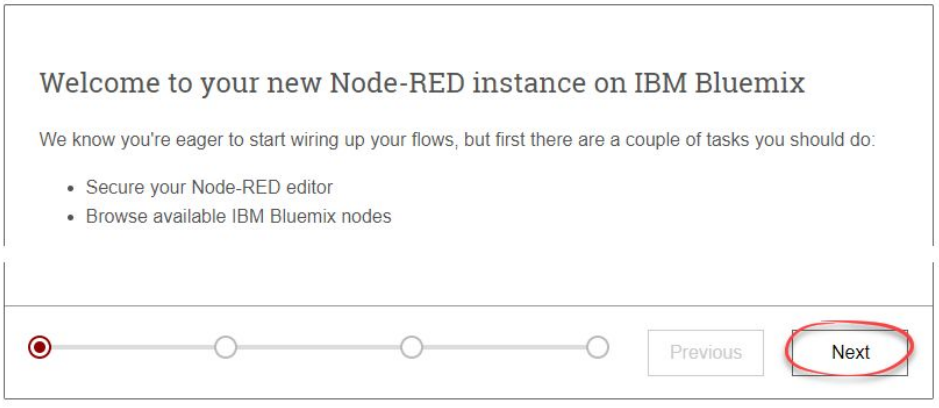
Choose ‘Create’ , this will create a Node-RED instance.

Your Node-RED server will be created and started. It can last up to 5 minutes!

After the creation you can select the ‘app url’:



A screen appears, asking you to secure your Node-Red editor



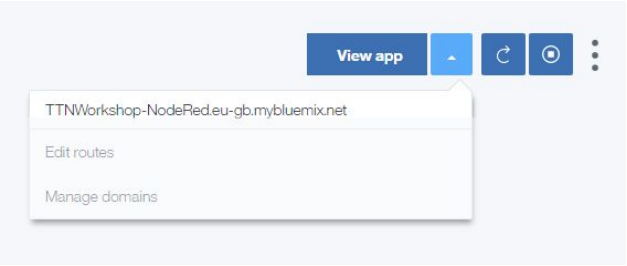
Enter (and remember) a username and password

If you do not enter any password, your Node-Red instance will be worldwide open and available!

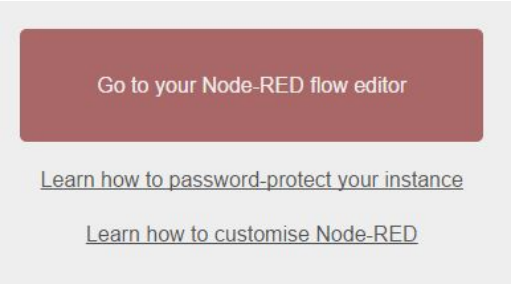
Skip the next info screen ‘Browse available IBM Bluemix nodes’.

Now press ‘Finish’ to finish the installation.

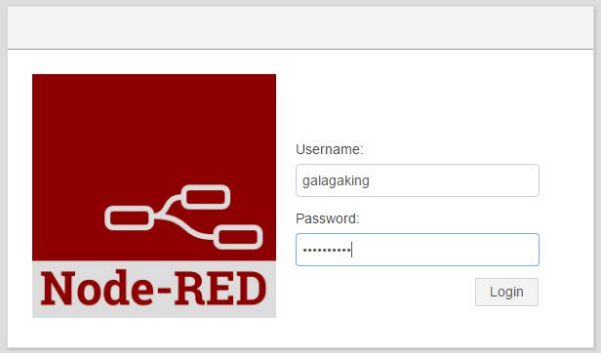
Access your Node-Red server



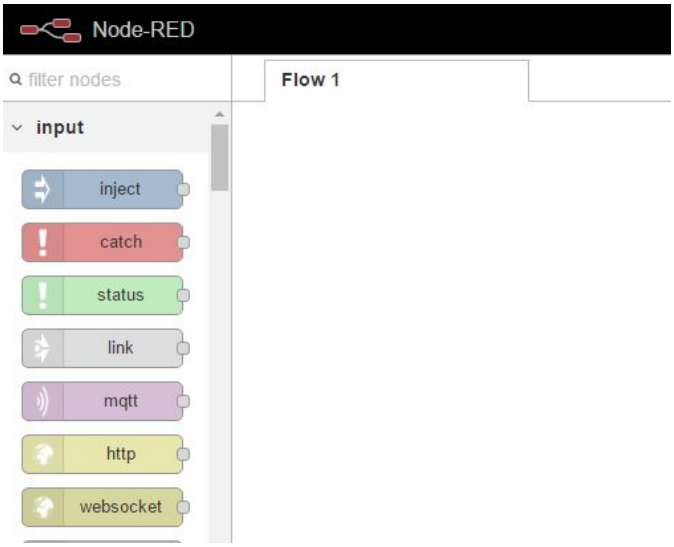
Select the name of your server, you can also use this URL to access your Node-Red server.



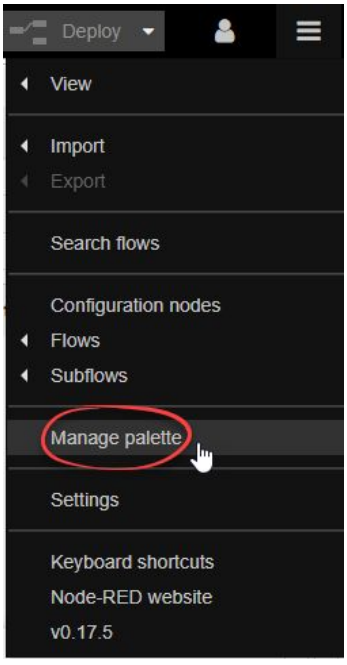
Now you have to enter your previous password:



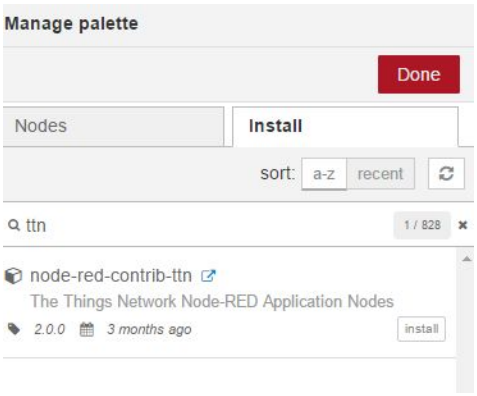
Your Node-Red is nearly ready to roll!



Now select the top right menu:



Choose manage Palette



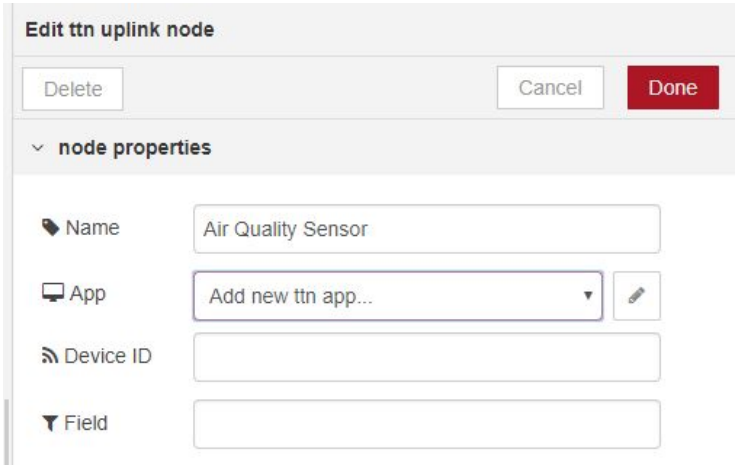
- Search for TTN in the Install tab and click install to install the TTN nodes.
- Click again install on the warning screen.
- ‘Done’ to exit the palette management.

TTN nodes can now be added in your Node-Red flow.

We start with a clean ‘flow’.

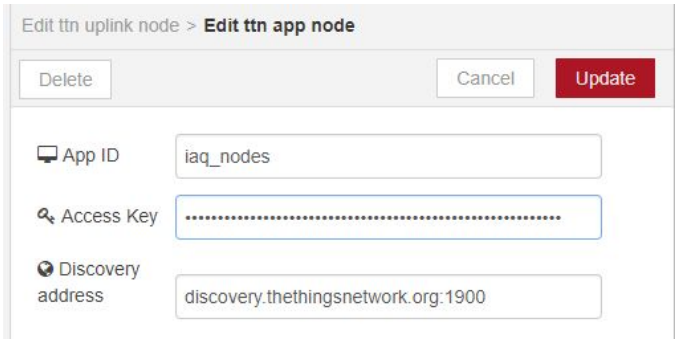
Add a TTN uplink node (formerly known as message node) by drag and drop it on your flow screen.

By double clicking on the TTN uplink node you can change the values:



The name can be any name,

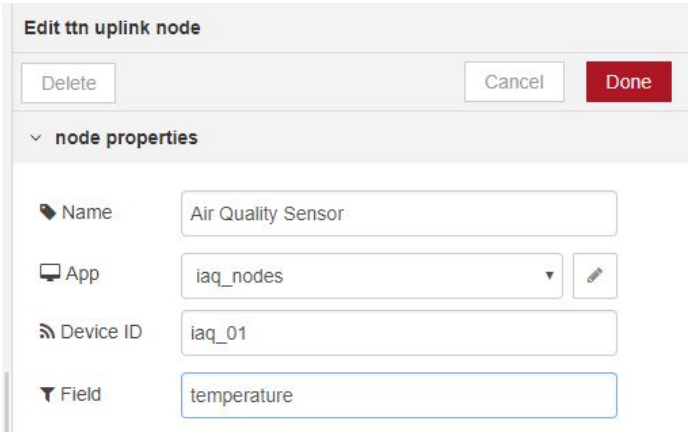
App refers to your TTN console. With the Pencil you can add your application here:



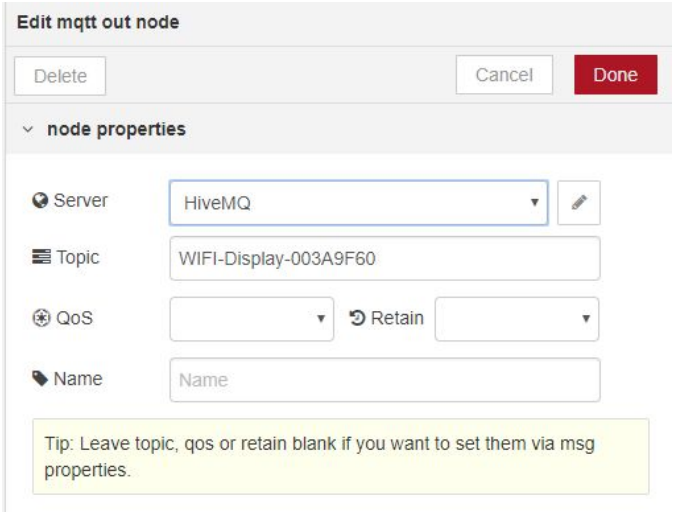
- App ID is the written name ‘Application ID’ from the TTN console

- Access Key is a copy of the access key of your application (access key of application, default key).
- Check Update and select your App in the previous screen.
- Device ID is the Device ID in TTN console

Beware that you use the same payload function in TTN as used in the previous nodes workshop!! If you do not have a payload function loaded, you can use an empty ‘field’, all the output will be shown.

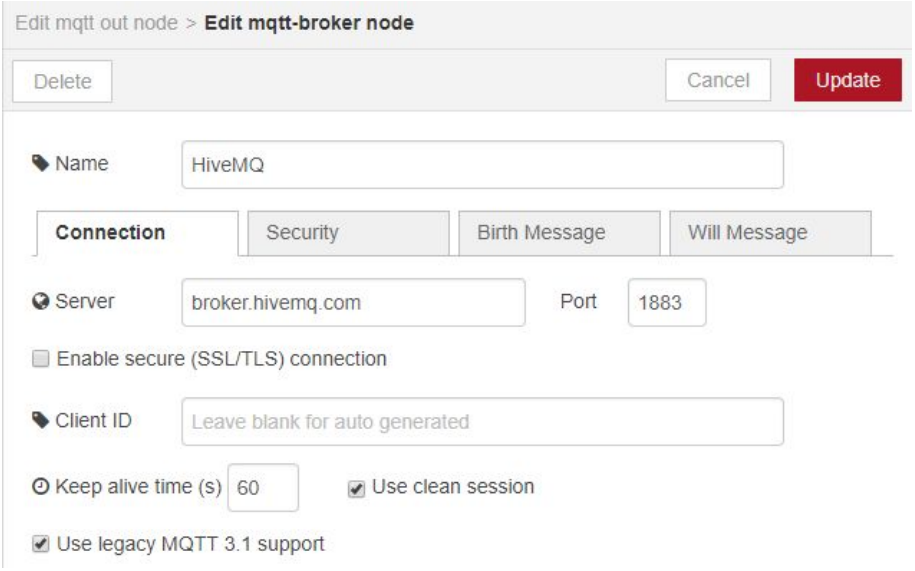


Now add a ‘MQTT’ output node to your flow and connect the both with a wire:

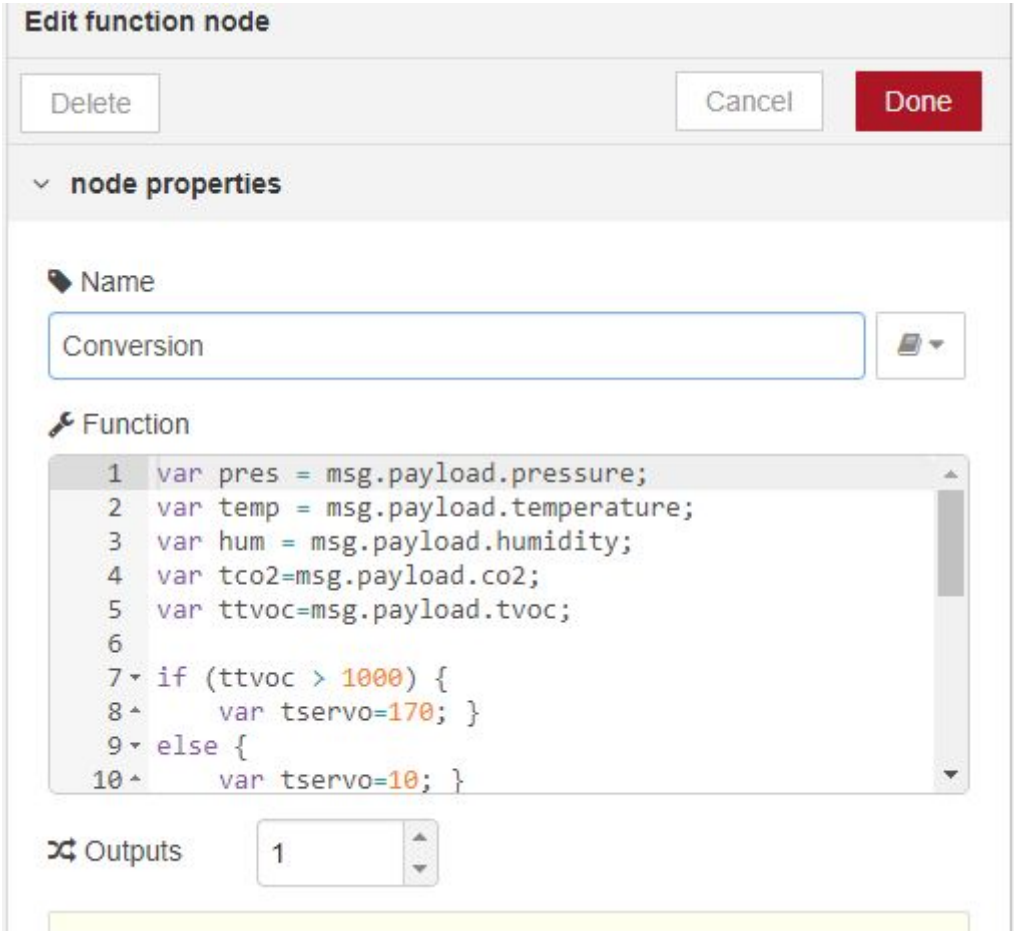


Get the Topic name from your Wemos LOG (WIFI-Display-XXXXXXXX).

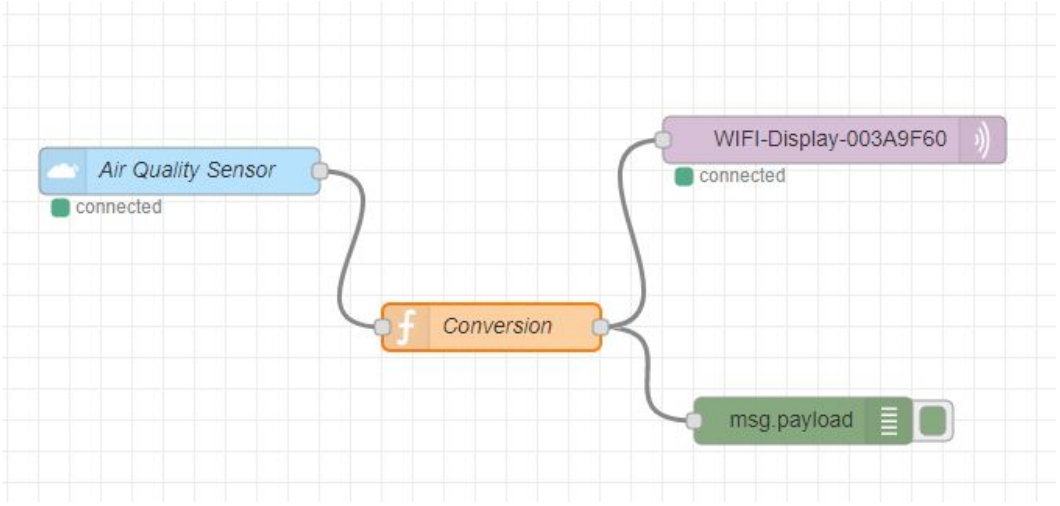
Fill in the server name: broker.hivemq.com. If you want to use your own MQTT service, fill in accordingly. This should be the same as your Wemos node!



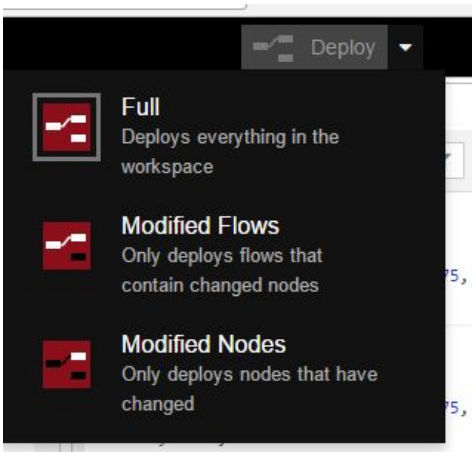
Add a function and put the Conversion.js code from github in it:



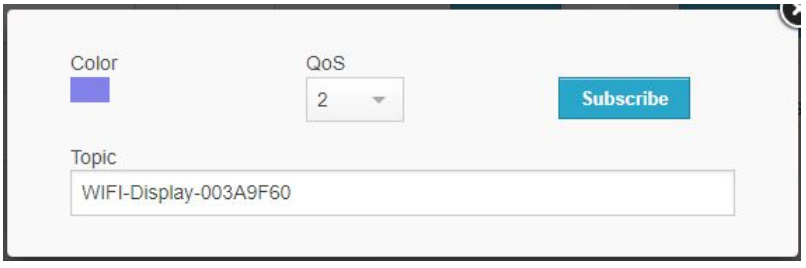
Connect the TTN Node, the Function and the MQTT node together. Add a debug node for debugging purposes:



Now activate your flow with Deploy->Full:



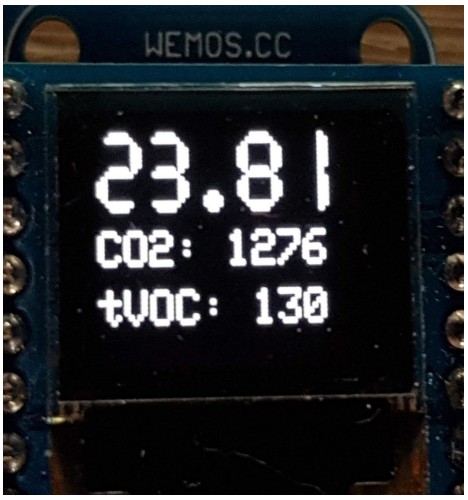
We can check the working of MQTT in the HiveMQTT console: http://www.hivemq.com/demos/websocket-client/ . Subscribe to your WEMOS:



Fill in your WEMOS name as a topic and select Subscribe. You will see the messages published by node-red.

ESP MQTT node

After the first measurement, the Wemos Node will show the temperature, CO2 and tVOC:



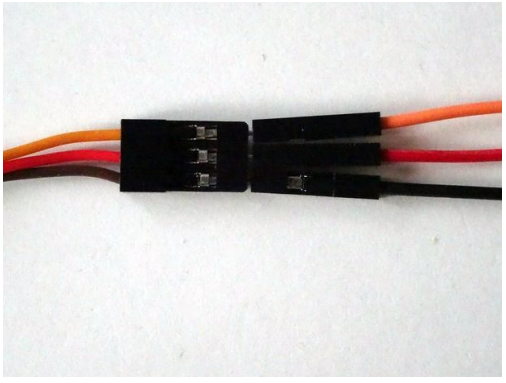
The console will display the values as well:

```
Subscribed to: WIFI-Display-003A9F60
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"160","co2":"1460","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"165","co2":"1484","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"165","co2":"1484","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"165","co2":"1484","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"155","co2":"1428","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"155","co2":"1428","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"160","co2":"1460","led":0,"servo":10}
Message arrived [WIFI-Display-003A9F60] -> {"pressure":"1019","temp":"23.50","tvoc":"155","co2":"1428","led":0,"servo":10}
```

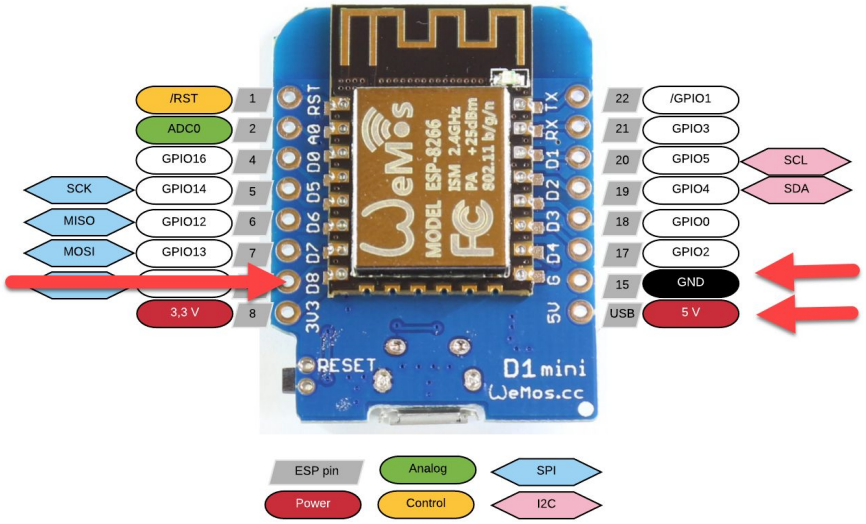
You can alter the ‘display’ part of the code to show other values. In the Conversion.js function there is some ‘logic’ which will light the onboard LED of the WeMOS if the temperature is > 25 degrees. Also the servo position is changed if the air quality is ‘bad’ (>1000 tVOC). You may alter these values for your own application.

Connect the servo

Use three jumper wires to connect your servo.



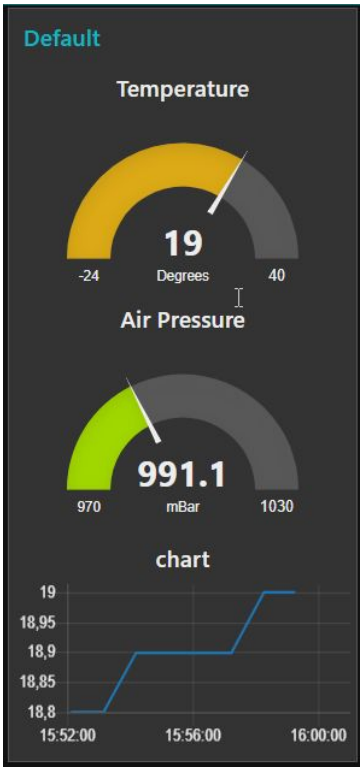
Connect the black wire (GND) to GND / G, the RED one to 5V, and the orange one to D8:



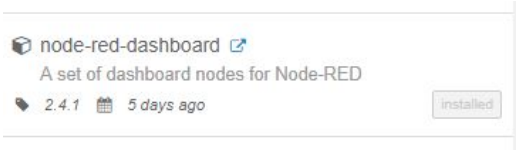
The ESP node can be used anywhere, just change the wifi credentials and it will connect to the (cloud) MQTT server and show your results. In this way you can move servos, but also switch lights on and off by publishing values to MQTT. The ‘callback’ function can do the work for you, controlled by Node-Red.

Using a web interface

Having a web-interface to get info about your nodes can be useful. We will add this to Node-RED.

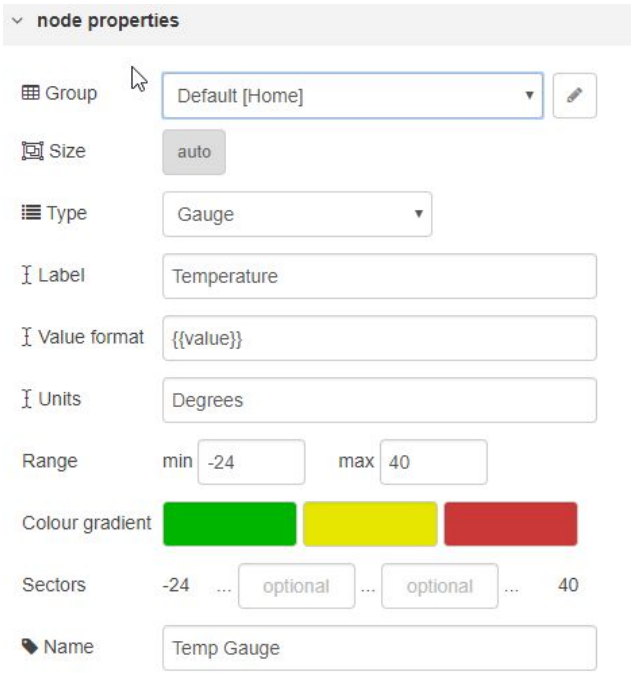


Choose ‘manage palette’, select ‘install’ and find ‘node-red-dashboard’



Install this module.

There are more functions added to your palette at the left side of your screen. You can add ‘chart’ to display a chart of your values:



You can select the colours, sizing and Legend. You have to create a ‘ui group’ to Group together even more charts, buttons etc.

You have to filter the JSON object for just having the value you want. You can do this by changing your TTN message node, or by adding a function:

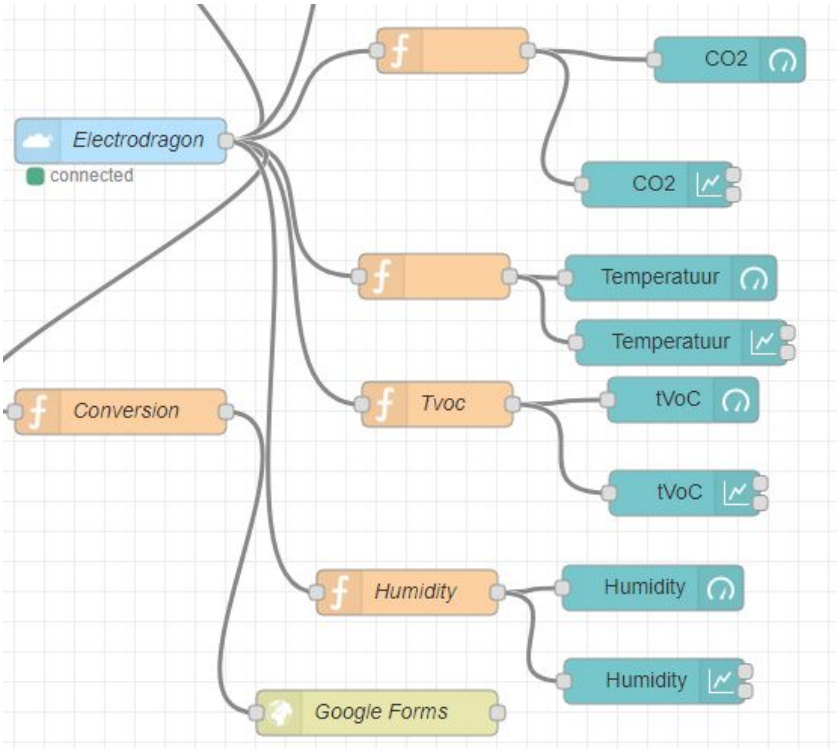
Add a ‘function’ object:

And edit the function in the way you want:

```
var newMsg = { payload: msg.payload.tvoc };

return newMsg;
```

The flow will look like this:



After creation, you can visit: http://<yourapplicationname>-eu.mybluemix.net/ui/#/0 to show your values

Usefull links

- Node-Red: https://node-red.org , http://noderedguide.com/
- Node-Red TTN docs: https://www.npmjs.com/package/node-red-contrib-ttn
- Node-Red and TTN: https://hansboksem.wordpress.com/2017/03/02/thethingsnetwork-weather-station-with-node-red-and-domoticz/
- http://developers.sensetecnic.com/article/a-node-red-dashboard-using-node-red-dashboard/