Bram Debontridder

Robin Lievens

Dean Terweduwe

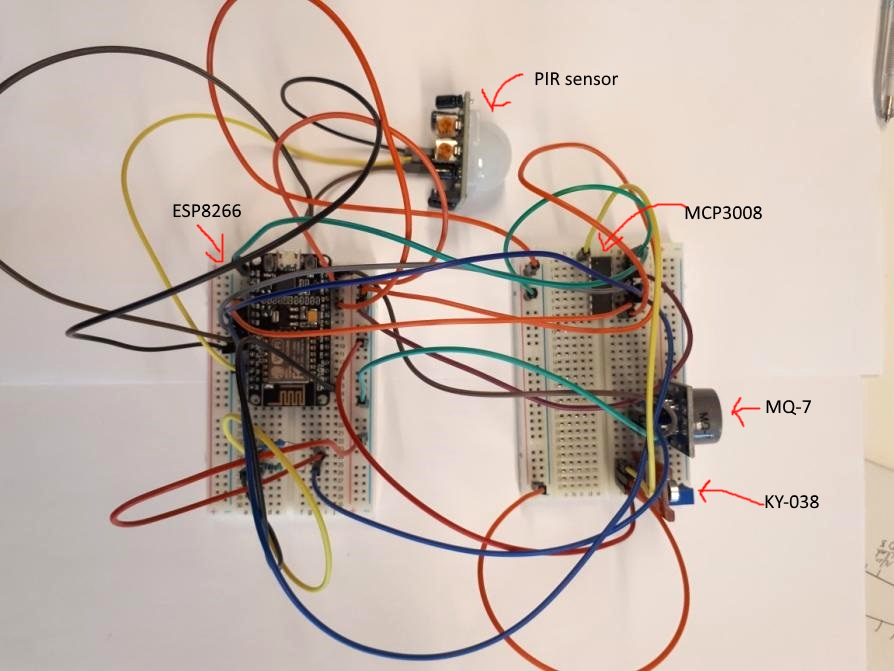
Felix Roels

dB²: Project IoT

**Design: Hardware**

The components of this project are:

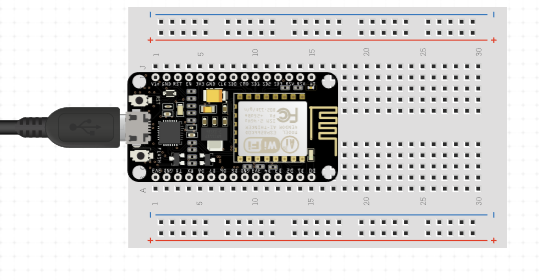
* ESP8266 NodeMCU 12-e
* 2 breadboards + a sufficient amount of Dupont wires
* MQ-7 CO sensor
* KY-038 sound level sensor
* PIR movement sensor
* MCP3008 10-bit Analog-to-Digital Converter
* 2 resistors: 470 Ohm + 1K Ohm



*Overview of the 2 breadboards with all the components*

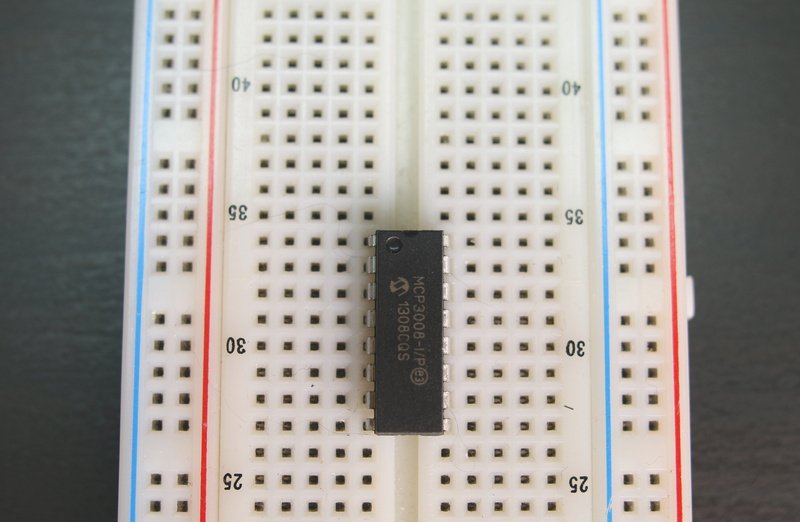
**Hardware 1.1: Connecting all the components**

Step 1: Plug in the ESP8266 on the breadboard.



*The controller plugged into the breadboard*

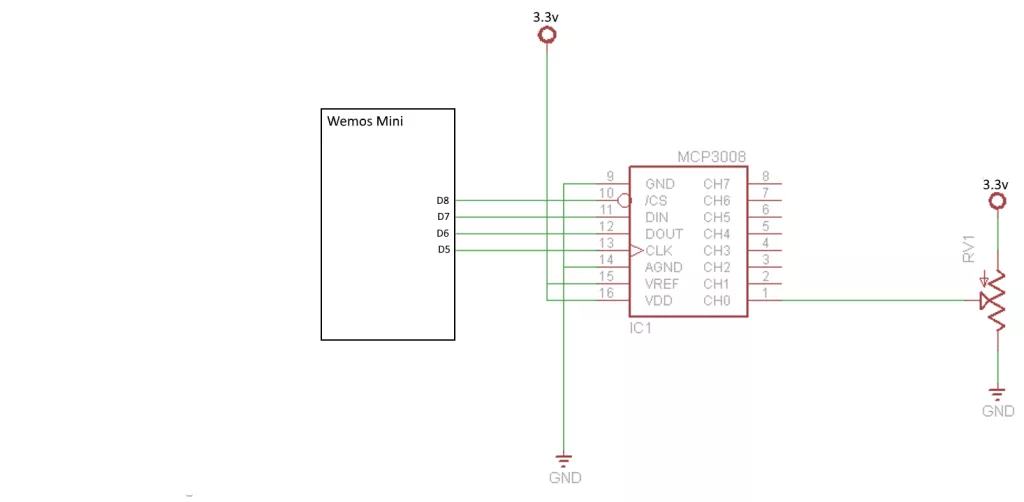
Step 2: Plug the MCP3008 into the other breadboard.



*The MCP3008 plugged in.*

Step 3: Connect the controllers to each other.

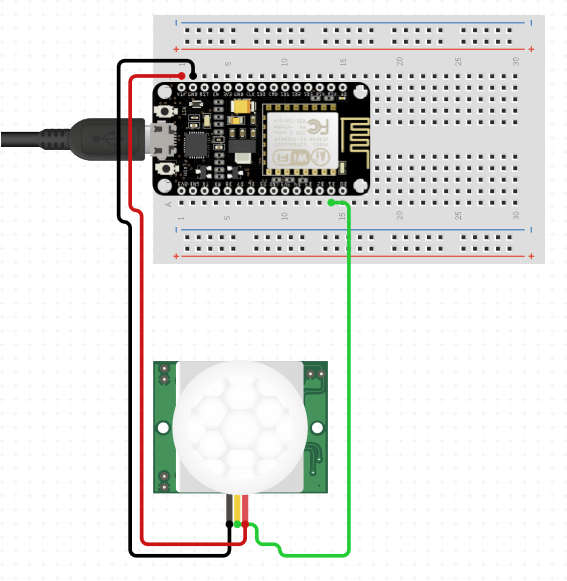
* Follow this overview to appropriately link the two together.



*Diagram of the layout of the MCP3008 chip*

Step 4: Connect the PIR sensor to the ESP8266 controller.

* 1 pin for ground
* 1 for signal
* 1 for power
* 3-5V → since the ESP8266 controller has 3 3v3 pins, we used one of these.

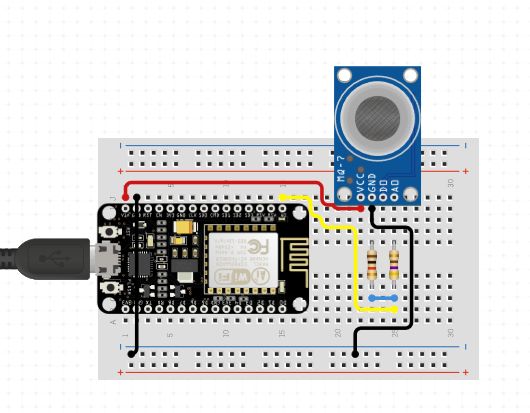


*The PIR sensor connected to the ESP8266.*

*Note: From here on the diagrams will be one sensor at a time.*

Step 5: Connect the MQ-7 CO sensor to everything.

* 4 pins in total
* We won’t be using the D0 pin, just connect the A0 pin



*The MQ-7 connected to the ESP8266 directly*

*Note: The analog out is connected to the 470 Ohm resistor.*

Step 6: Connect the KY-038.

* Because we want to use an analog value for our microphone but only have one analog port, we need to use the MCP3008 (analog-to-digital).
* Video for wiring schematics.
* In the video they use a potentiometer for our microphone just use the A0, G and + port.

→ <https://www.youtube.com/watch?v=NGNNDz_ylzs&t=385s>

**Arduino programming**

**2.1 Installation of software**

If you haven’t already downloaded the Arduino IDE, you should download it now.

<https://www.arduino.cc/en/Main/Software>

Install the ESP8266 Board

1. Open the preferences window from the Arduino IDE
2. Go to File > Preferences
3. Enter http://arduino.esp8266.com/stable/package\_esp8266com\_index.json into the “Additional Board Manager URLs” field as shown in the figure below.
4. Click OK
5. Go to Boards Manager: Tools > Board > Board Manager
6. Scroll down to ESP8266 and click install.
7. Choose the board from Tools > Board > NodeMCU V1.0 ESP 8266-12E and select CPU Frequency 80MHz and Upload Speed: 115200.

Install the PubSubClient

1. Tools > Board Manager > Type **PubSubClient** into the search.
2. Click on the one by **Nick O’Leary**

Install the EasyNTPClient

1. Tools > Board Manager > Type **EasyNTPClient** into the search.
2. Click on the one by **Harsha Alva**

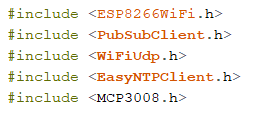
Install the MCP3008 library

From: <https://github.com/nodesign/MCP3008>

1. Sketch > Use libraries > .ZIP add library

**2.2 Writing the program**

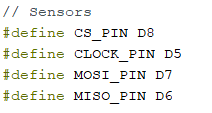
1. Include the right libraries.



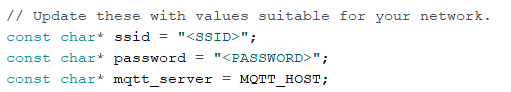
1. Write the code to make the connection via MQTT to IBM Watson



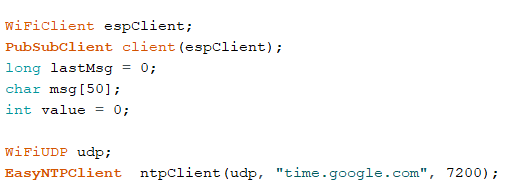
1. Define the right pins for the sensors



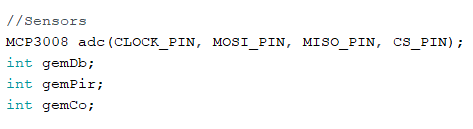
1. Connect to the network



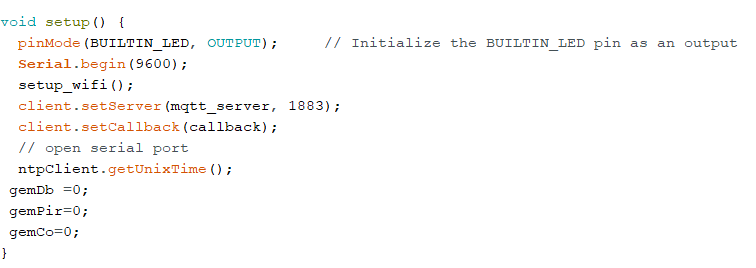
1. WifiCLient and PubSubClient we’ll be using to connect to our wifi network and the EasyNTPClient to get Unix Time for our timestamp later on.



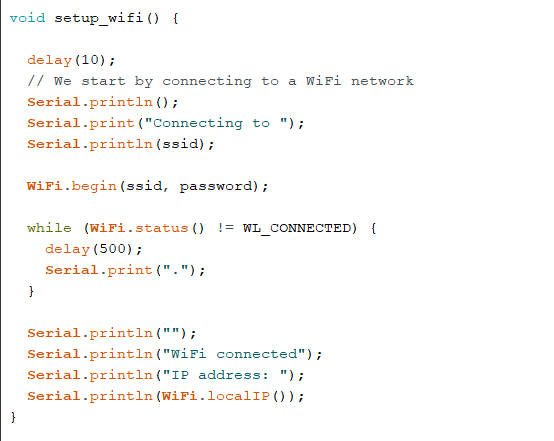
1. Some code so that the MCP3008 works with our sensors.



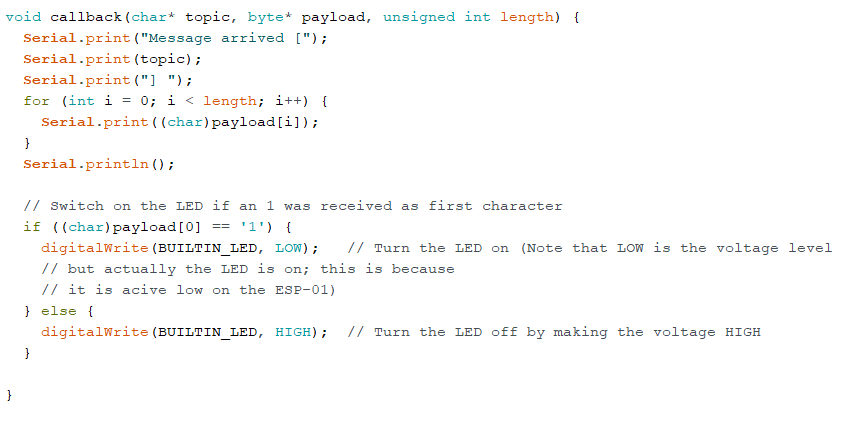
1. Write the setup() method



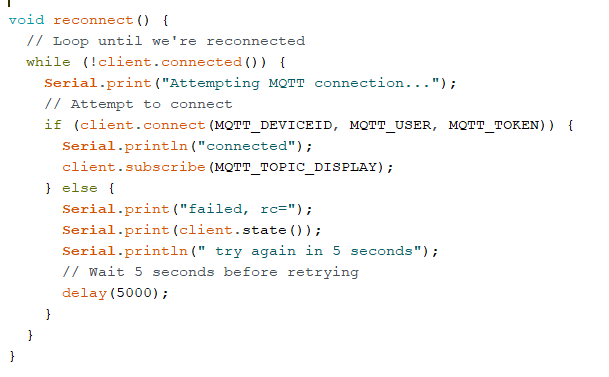
1. Write the setup\_wifi() method so our arduino can connect to the internet.



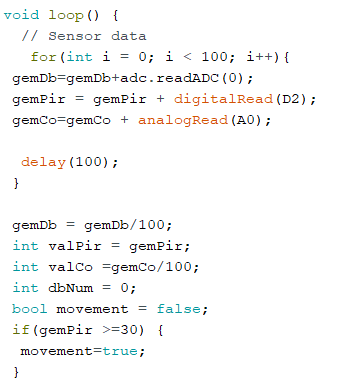
1. Next we write the callback() method

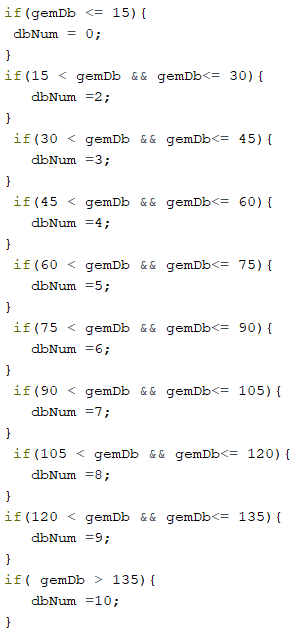


1. Reconnect() in case we lose our MQTT connection



1. loop() method: values coming from the sensors turned into values we can work with





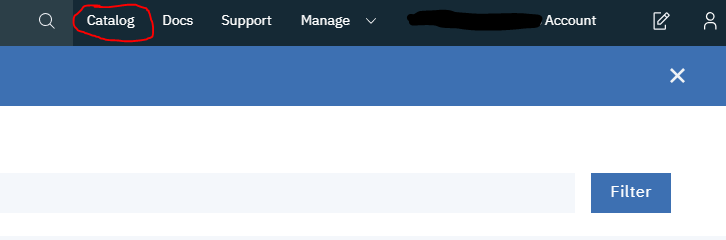
1. loop() method part 2: we put our values in a json format.



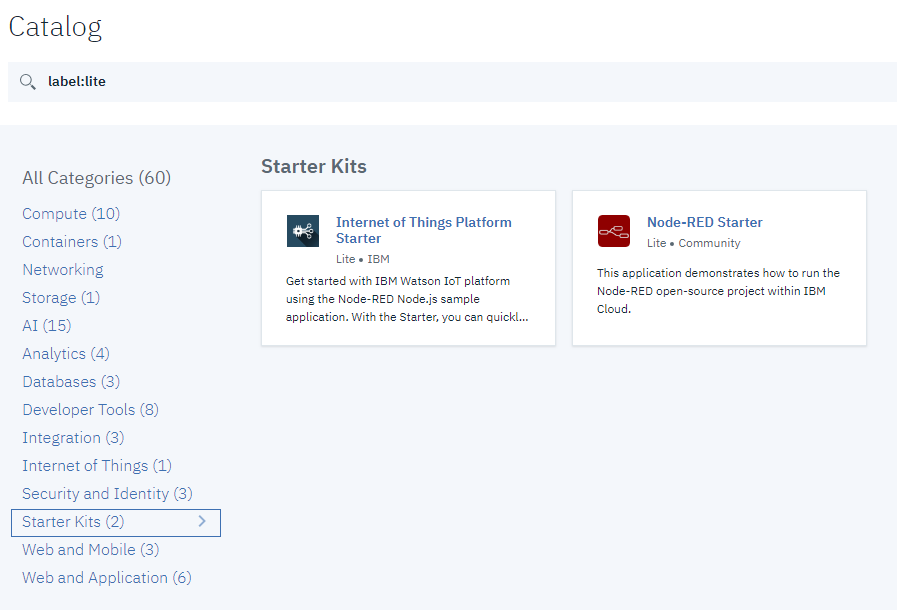
**MQTT**

For communication between our project and computer, we use the MQTT protocol, using IBM Cloud as a broker.  
To make this work we need access to the following software:  
- An IBM Cloud account (<https://cloud.ibm.com/login>) which grants access to:  
 - IBM Cloud  
 - Node-RED  
 - IBM Watson IoT Platform  
- MQTT.fx (<https://mqttfx.jensd.de/>) (or any other MQTT client application)

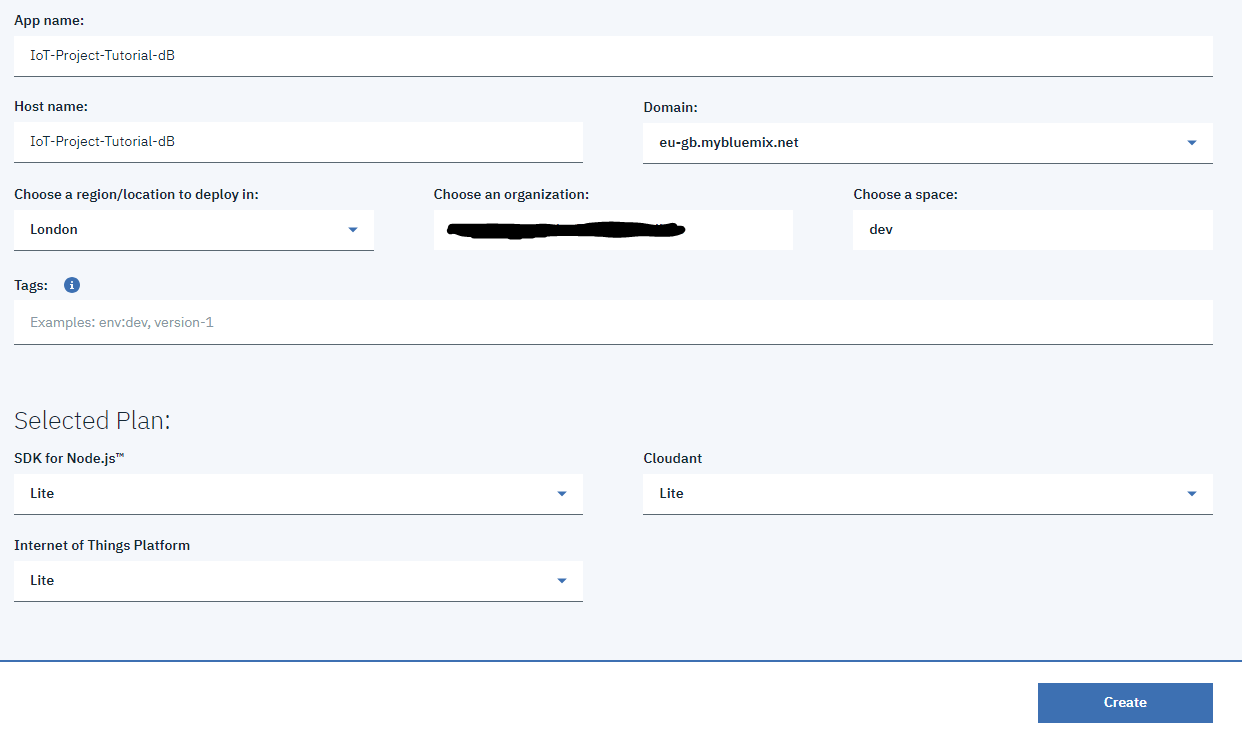
1. Register a device

Start by making a free account on IBM Cloud (<https://cloud.ibm.com/login>). Follow the steps of the registration and login on the platform.  
When you’re logged in, click on “Catalog” in the navigation bar on top.  


On the catalog page, click “Starter Kits” in the categories menu.  
Only starter kits should show up now. Click on the “Internet of Things Platform Starter”

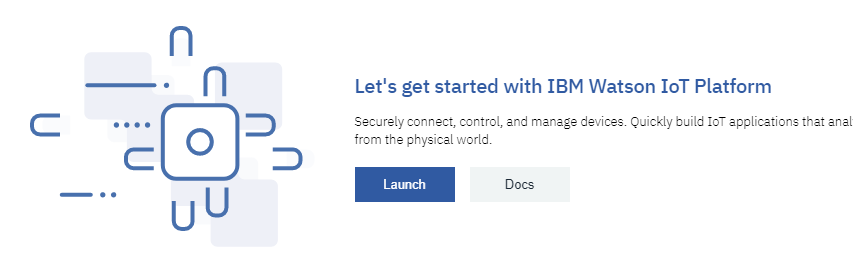


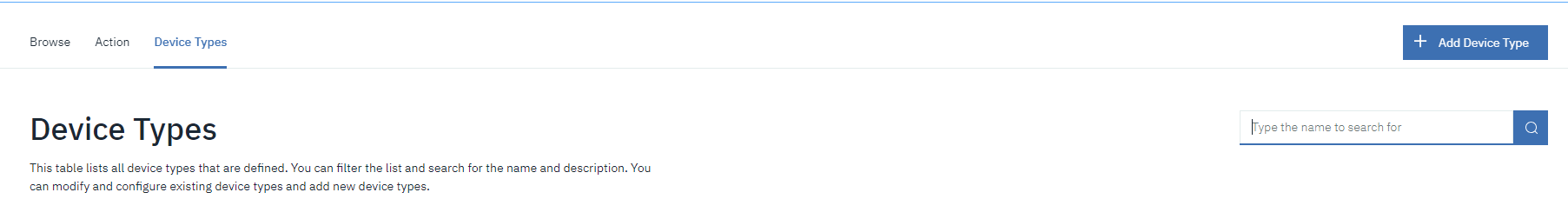
Change the “App name” to anything you want and leave the rest on default.



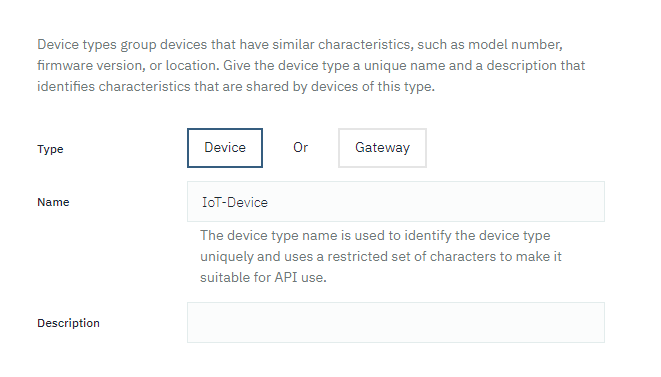
Click the blue “Create” button in the bottom right corner and wait a few minutes for it to finish.

Next, click on the hamburger menu in the top left corner and click on “Resource List”.  
Under “Cloud Foundry Services”, click on the name of your service (not the NoSQLDB).

This brings you to a new page, click the blue “Launch” button.  


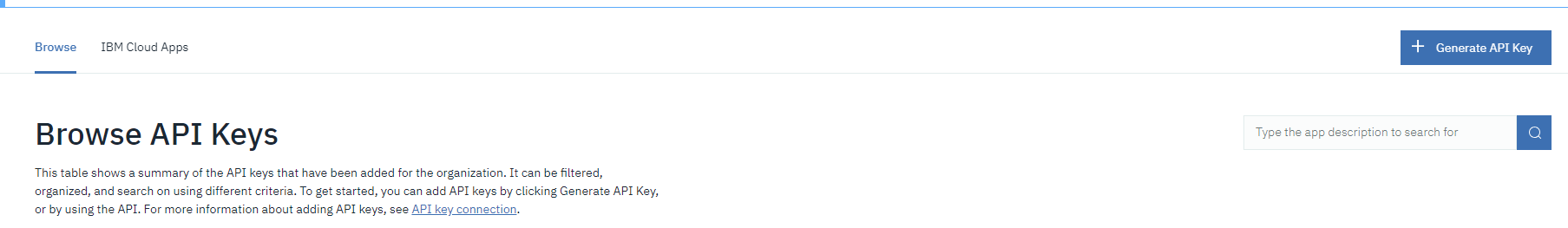
A new tab of the IBM Watson IoT Platform opens. Click “Device Types” and create a new device type by clicking the blue button “Add Device Type”.  


Choose a name for the device type and click “Next”. Click “Done”.

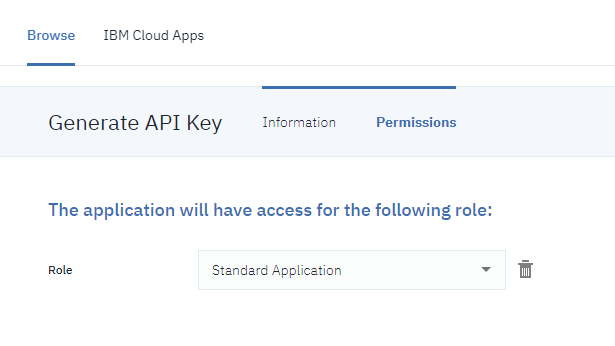


Now, register a new device of this type by clicking “Register Devices”.  
Choose a Device ID and click “Next”, “Next”, “Next”, “Done”.  
Your device is now registered. Copy the Authentication Token and save it somewhere, you will need this later.  
Press the return button in your browser to go back to the Devices page. Your device is now in the list.

2. Generate an API key

On the IBM Watson IoT Platform, navigate to the “Apps” page via the left menu.  


Next, generate an API key by pressing the blue button in the top right corner.  
Add a description if you want, this is not necessary. Click Next.  
Set the role to “Standard Application” and click “Generate Key”.

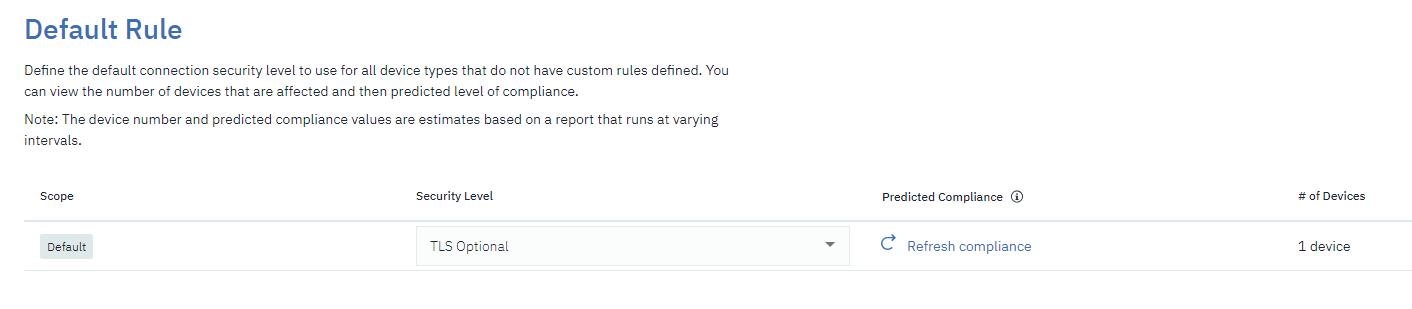


Copy the key and authentication token and save it somewhere, you will need this later.

3. Security settings

On the IBM Watson IoT Platform, navigate to the “Security” page via the left menu.

Click the blue icon on the right side of “Connection Security” to edit these settings.  
Set security level to “TLS Optional”.



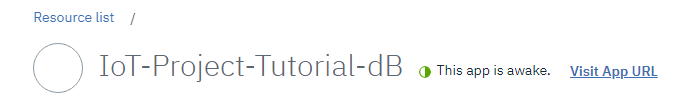
Press “Save” in the top right corner to save your settings.

**Node-RED**

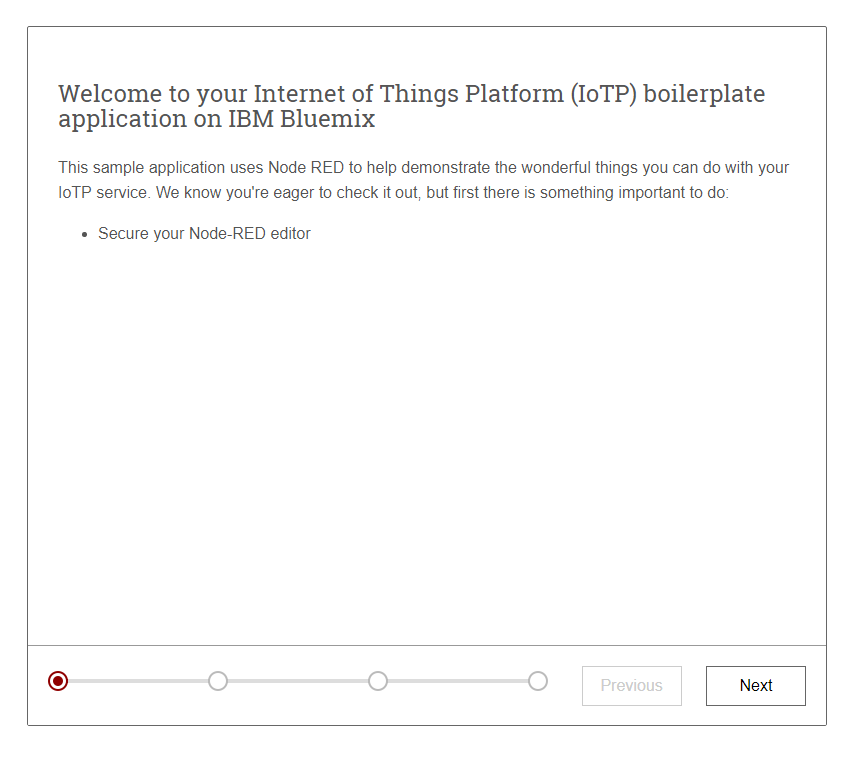
1. Setup

Go back to your Resource list on IBM Cloud (<https://cloud.ibm.com/resources>).

Now, under “Cloud Foundry Apps”, click on your listed app.  
A new pages opens. Click “Visit App URL”.

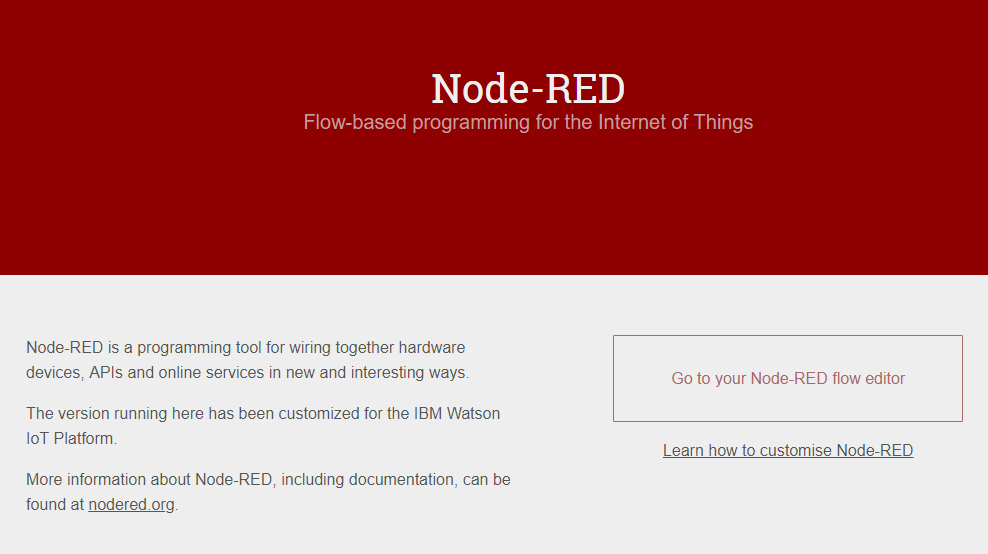


The Node-RED page opens.



Click “Next”.

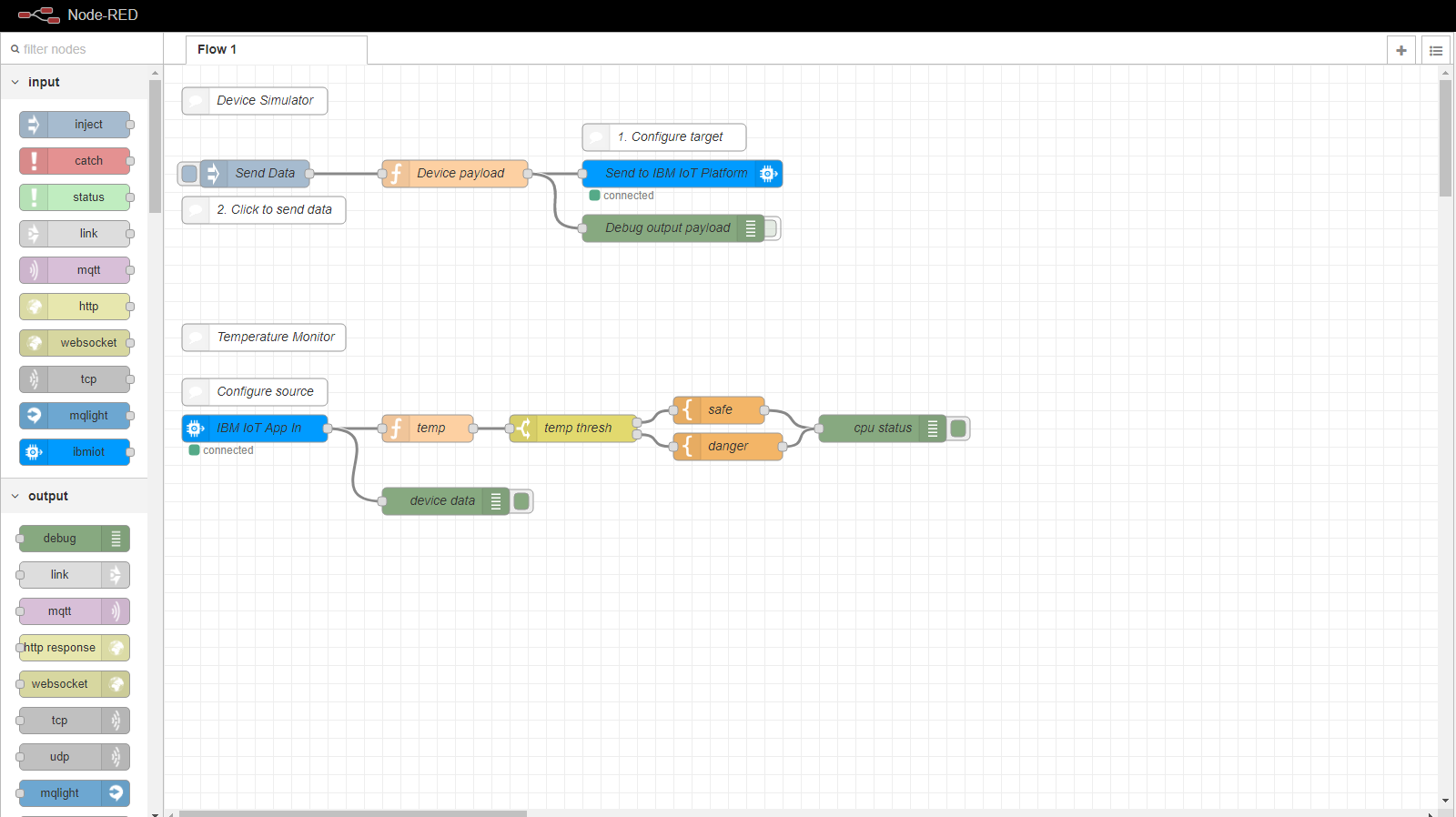
Choose a username and password and click “Next”. Click “Finish”.



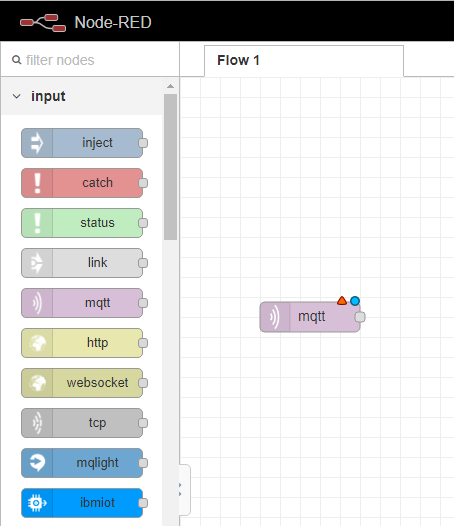
Click “Go to your Node-RED flow editor”.

Login with your credentials you just created.

On this screen, you can select everything and delete it. We don’t need it.



Start by selecting and dragging an “mqtt” input node on the grid.



Double click the node to change the properties.

Next to “Add new mqtt-broker…” click the icon on the right to create one.

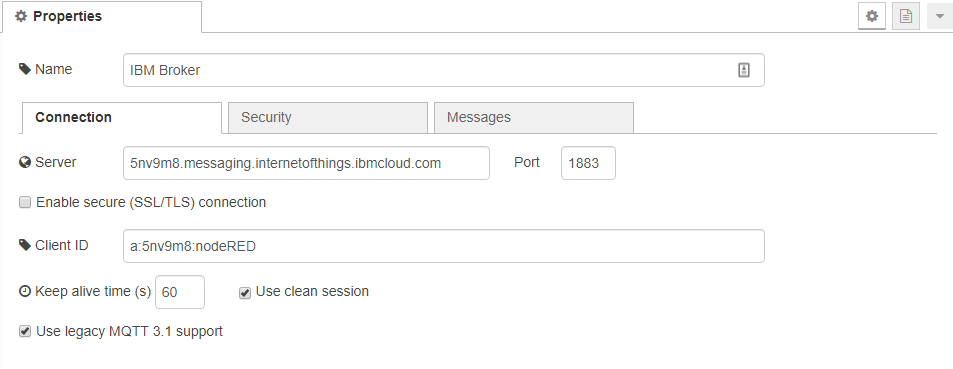
Fill in a name of your choice.

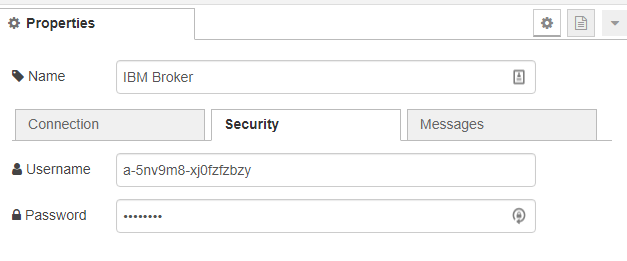
As server address, fill in the following:  
  
YOUR\_ORG\_ID.messaging.internetofthings.ibmcloud.com  
  
to find your ORG\_ID, open IBM Watson. Your ORG\_ID is shown under your name as “ID:”.

As Client ID, fill in the following:

a:YOUR\_ORG\_ID:YOUR\_CHOICE

YOUR\_ORG\_ID is the same as above.  
YOUR\_CHOICE is a name of your choice, fill in any string you want.



Now, click on the “Security” tab.  
In the username field, fill in your API key, which you generated on IBM Watson and saved somewhere.  
In the password field, fill in your API authentication token, which you saved somewhere.  


You are now configured to connect to your broker.

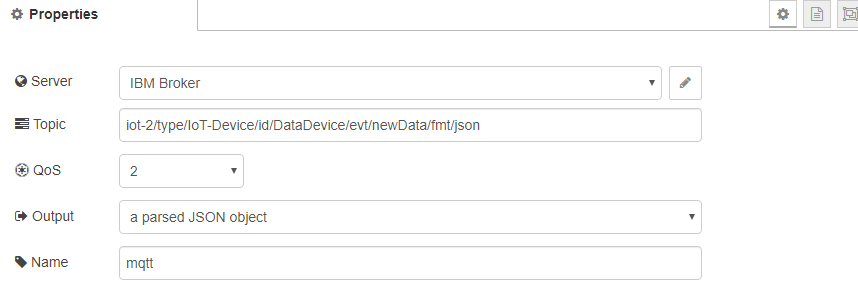
Go back to the client configuration.

In the topic field, fill in the following:

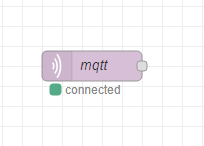
iot-2/type/YOUR\_DEVICE\_TYPE/id/YOUR\_DEVICE\_ID/evt/YOUR\_EVENT/fmt/json

YOUR\_DEVICE\_TYPE is the device type you created on IBM Watson.  
YOUR\_DEVICE\_ID is the device ID you created on IBM Watson.  
YOUR\_EVENT is the event you created in your ESP.

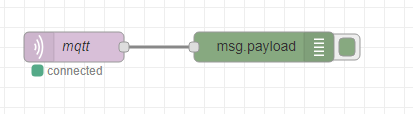
Fill in a name of your choice.

Change the Output to “a parsed JSON object” and click “Done”.  


Click “Deploy” in the top right corner. Your mqtt node should now have“connected” displayed under it.



If you add a debug output node and connect it to your mqtt node, you can check if your data is coming through. Deploy and open the debug window on the right. Normally it should work and your data should be coming through.

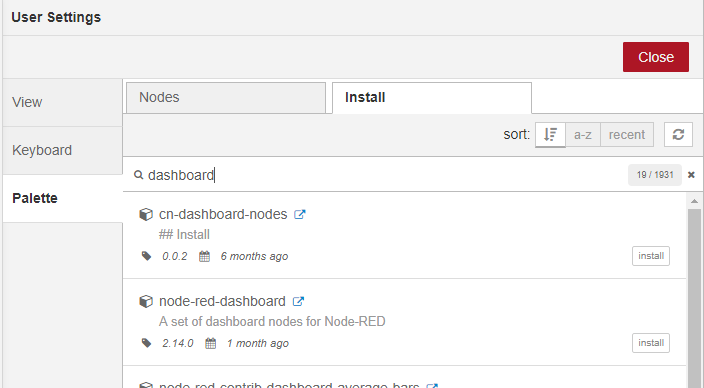


2. visualization of your data

To make our data visual, we need to import some extra nodes.  
Click on the hamburger menu in the top right corner and click on “Manage palette”.

Click on the install tab and type “dashboard”.

Install the “node-red-dashboard”.



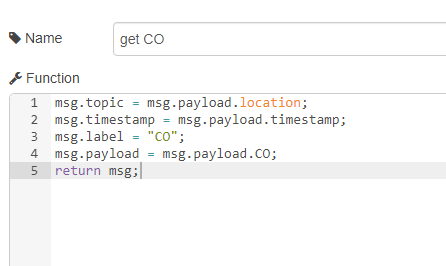
if the installation fails for any reason, just try again.

As an example we will show you how to make one value of our data visible, the other data is similar.

Example chart:

First, click and drag a function node on the grid. Double click it. This node will extract the CO from our data that is coming through.

The code for this is the following:



You can name if however you want.

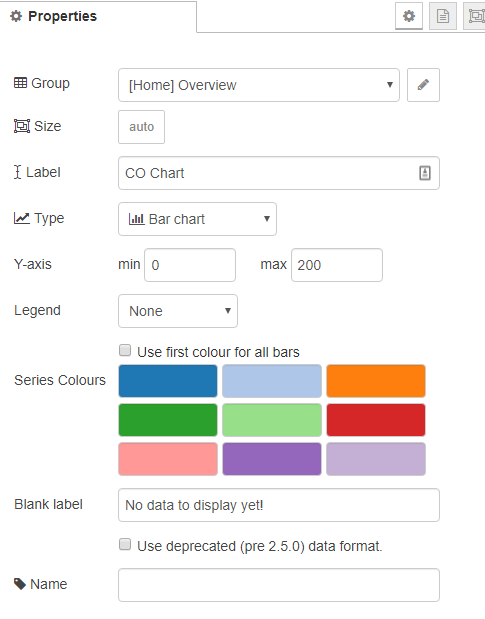
Next, add a chart node and connect it to the function node.

Choose a group name or create a new one.

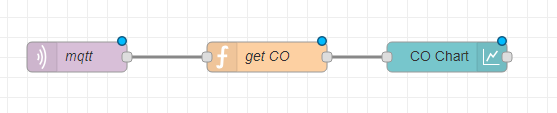
Choose a label.

Choose your minimum and maximum values to be shown.

You can customize it however you like.



Your flow should look like this:

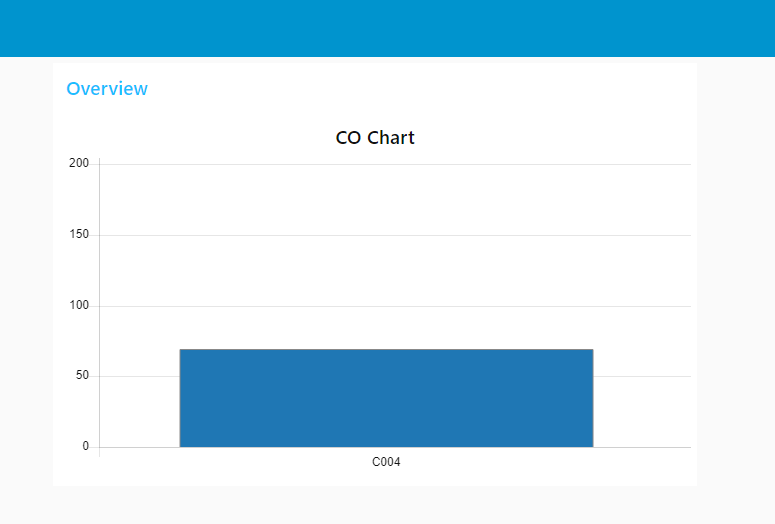


Deploy it.

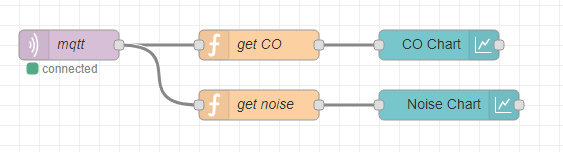
To look at the result, click on the dashboard tab. Clicking the small icon on the right will open up a browser tab.



This is the result:



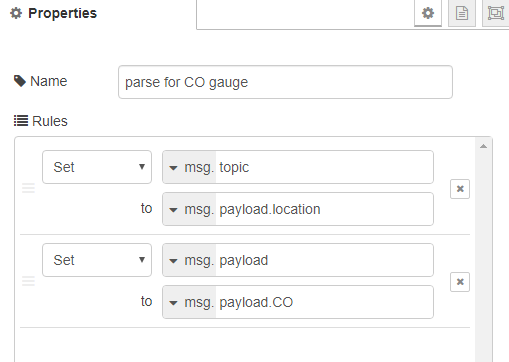
You can do the same steps to display the noise.



Example gauge:

Select and drag a switch node to the grid.

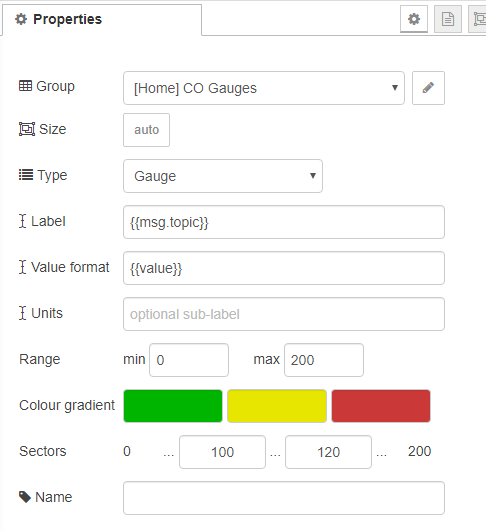
Configure it as shown in the following image:



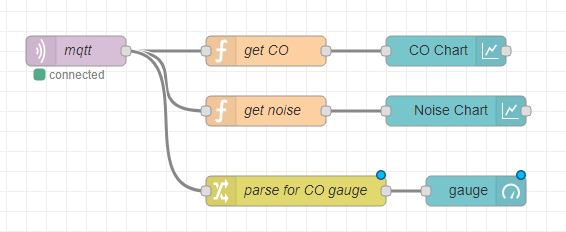
Connect it to your mqtt node.

Select and drag a gauge node to the grid.

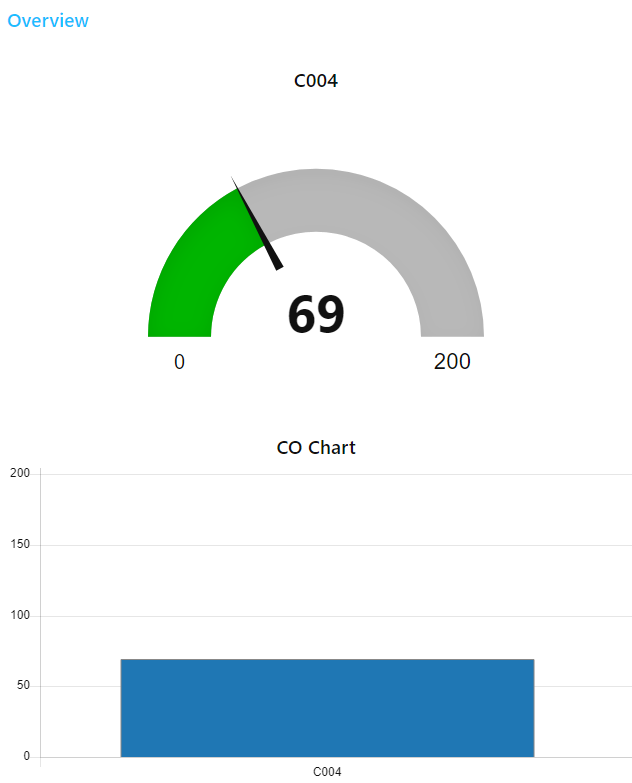
Configure it as shown in the following image:



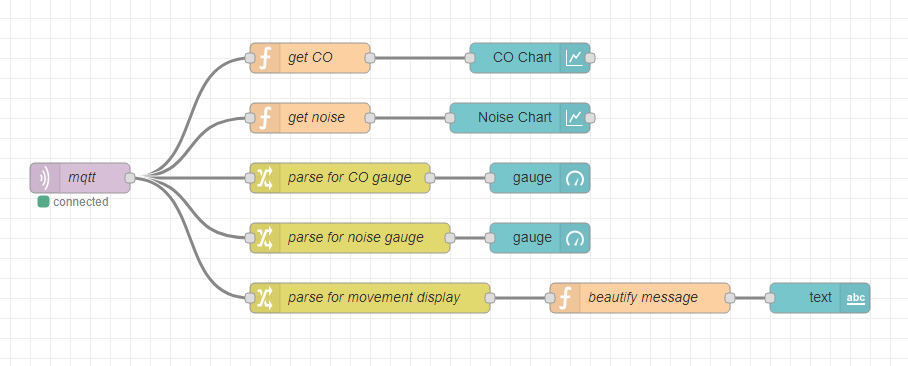
Connect it with the change node to create the following flow:



Deploy. Your dashboard should now look like this:



You can do this for all other variables to become the following result:



Congratulations, your data is now visualized.