Environment Monitoring

Markus van Kempen

IBM **SPEED**

mvk@ca.ibm.com

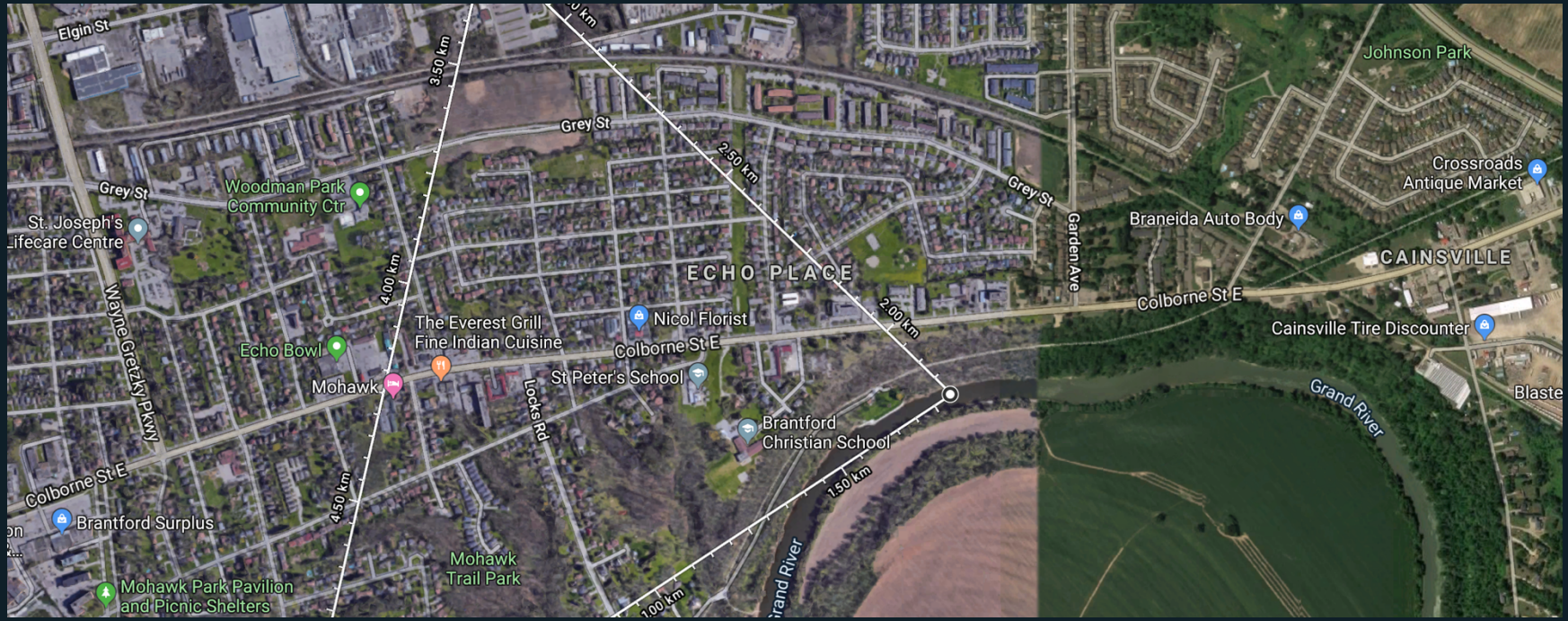
@markusvankempen

Version: 20210210

SNP STEAM ACADEMY

Code and instruction for this class can be found here
https://github.com/markusvankempen/esp8266andDHT11

Our Goal – Environmental Monitoring Project



Why LoraWAN

- It’s free - no Telco costs
- LORA = LongRange 10km and LowPower (LPWan)
- Lots of use-cases / examples and support available
-

- More infos
- https://www.semtech.com/lora/why-lora
- https://os.mbed.com/docs/mbed-os/v5.15/tutorials/LoRa-tutorial.html

Lora Gateway & LoraServer

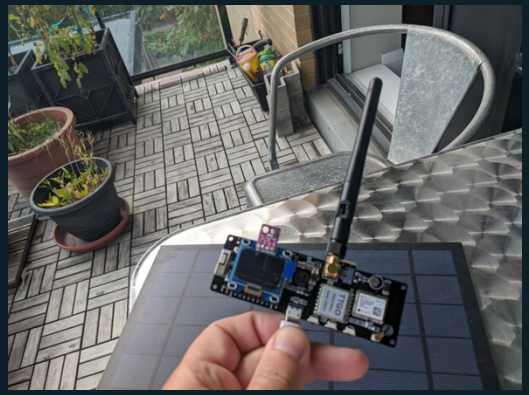
- Has Wifi/lora/GPS/LTE
- Power via Power over Ethernet (POE)

- More infos here
- https://www.rakwireless.com/en-us/products/lpwan-gateways-and-concentrators/rak7249

- So far tested Gateway setup with different lora Nodes connection to TTN and MQTT

- ToDo’s
  - Test LTE setup
  - Need Solar Kit
  - https://store.rakwireless.com/products/solar-kit?variant=31385712885805

Lora Nodes –with batteries and solar panel – TTGO -T-Beam



TTGO … lots of software support Arduino based and wifi support
Pro
Really good power management … all one … with gps options for tracking

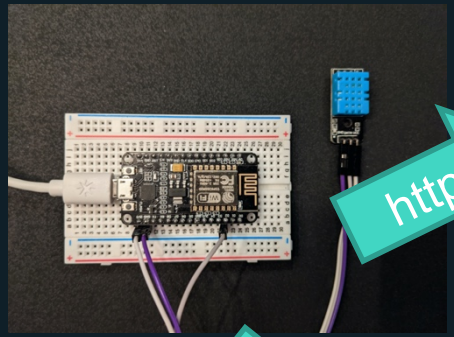
http://www.lilygo.cn/products.aspx?TypeId=50003&FId=t3:50003:3
https://github.com/JoepSchyns/Low_power_TTGO_T-beam
https://github.com/LilyGO/TTGO-T-Beam
https://tinymicros.com/wiki/TTGO_T-Beam

Today

We will simulate the LoRa network using a WIFI network with our esp8266 & DHT11 to send temperature data via mqtt to a server and save the data into a database and than “surf” the data and visualize it using Grafana.

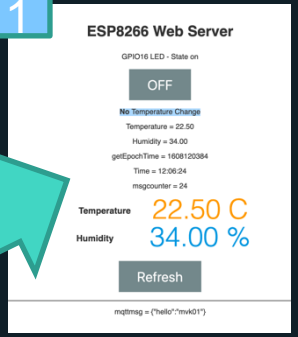
In our esp8266 journey we will setup the Arduino IDE, using the esp8266 as a webserver, Integrate the NetworkTimeProtocol and look at the deep sleep mode for es8266 and and of course send and store the temperature data.

Overview – ESP8266 and DHT11

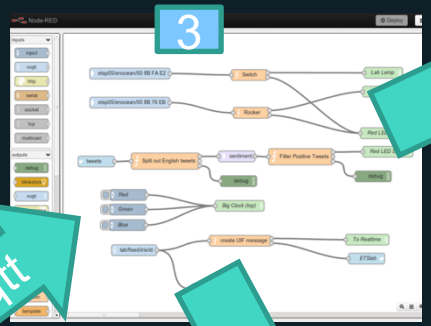


http

1



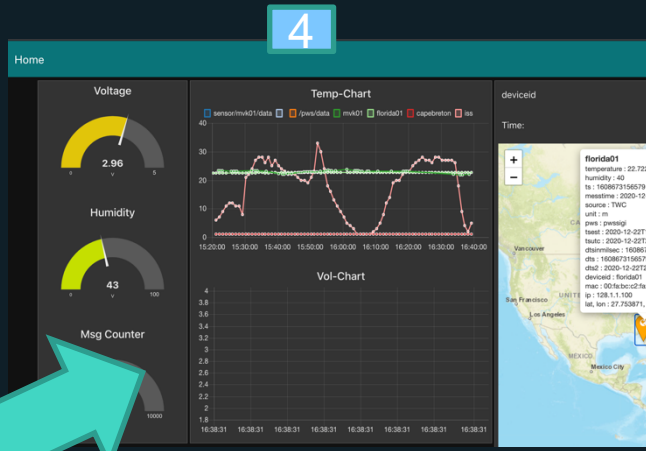
mqtt



3

mqtt

4



5

5

SQL

Surf

2

Virtual Server with docker images

52.117.240.201
Or
snpweather.xyz

mosquitto

IBM

Agenda

1st Chapter (One) – Arduin and ESP8266 Setup

2nd Chapter (two) – ESP8266 as a webserver

1st Chapter (One)

Arduino IDE setup

ESP and DHT11 wiring

Esp8266 and dht11 test program

Some exercises/play

Arduino IDE setup - esp8266 board

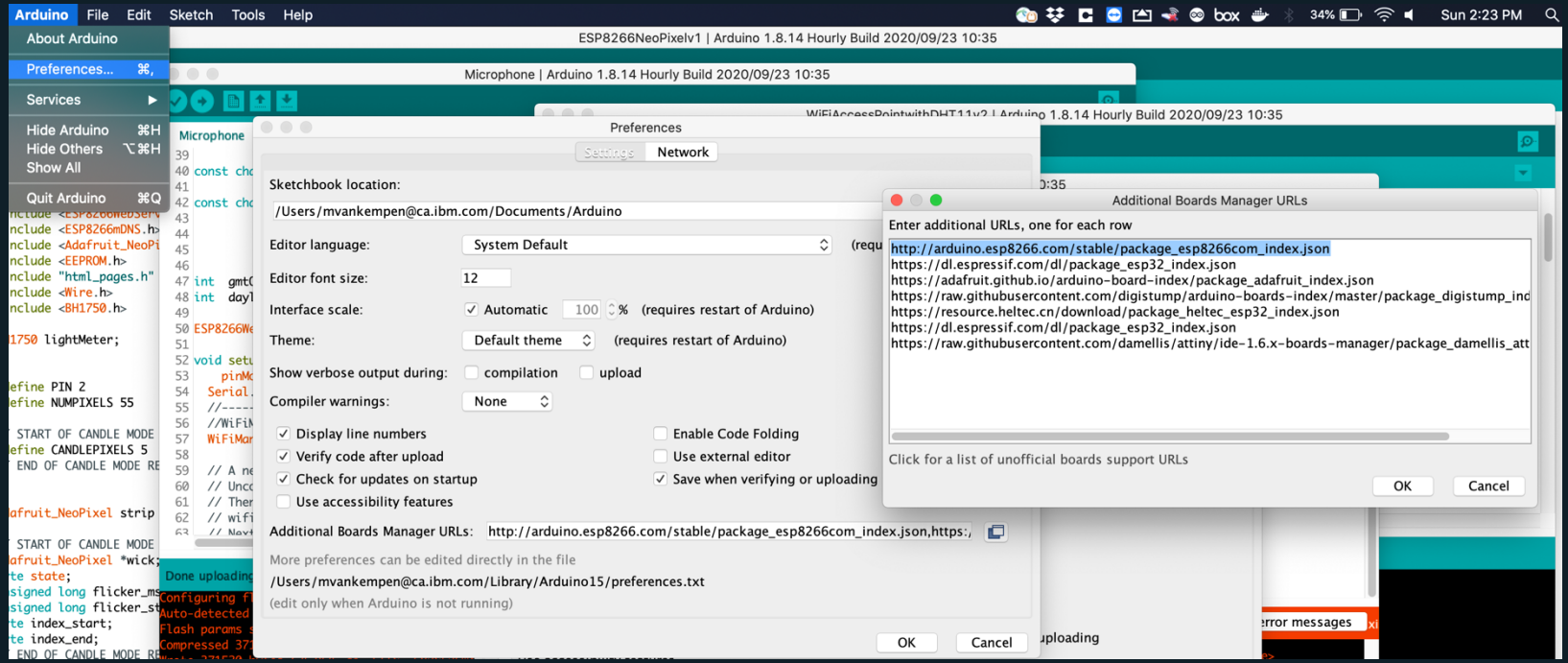
We need to add the esp8266 board and libraries to the IDE

Instructions

- Start Arduino and open Preferences window.
- Enter **https://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 find *esp8266* platform.
- Select the version you need from a drop-down box.
- Click *install* button.
- Don’t forget to select your ESP8266 board from Tools > Board menu after installation

More reference
https://arduino-esp8266.readthedocs.io/en/latest/installing.html
https://www.hackster.io/rajthavti/dht11-sensor-interfacing-with-nodemcu-086762

ESP8266 board import



Arduino IDE setup - dht11 library

You should have the **Arduino IDE** software running at this time. Next isto install our DHT library, which can be done though the Arduino Library Manager:

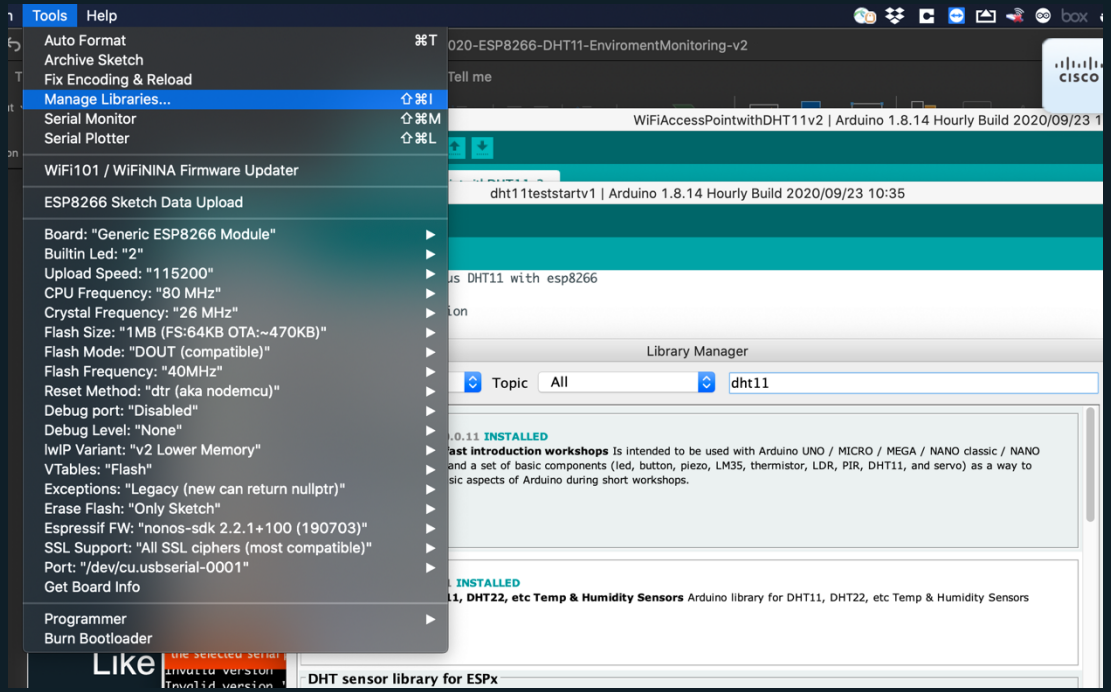
Sketch→Include Library→Manage Libraries…

Enter “**dht**” in the search field and look through the list for “**DHT sensor library** by **Adafruit**.” Click the “Install” button, or “Update” from an earlier version.

DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library

More reference
https://arduino-esp8266.readthedocs.io/en/latest/installing.html
https://www.hackster.io/rajthavti/dht11-sensor-interfacing-with-nodemcu-086762
https://learn.adafruit.com/dht/downloads?view=all

Import DHT11 Library



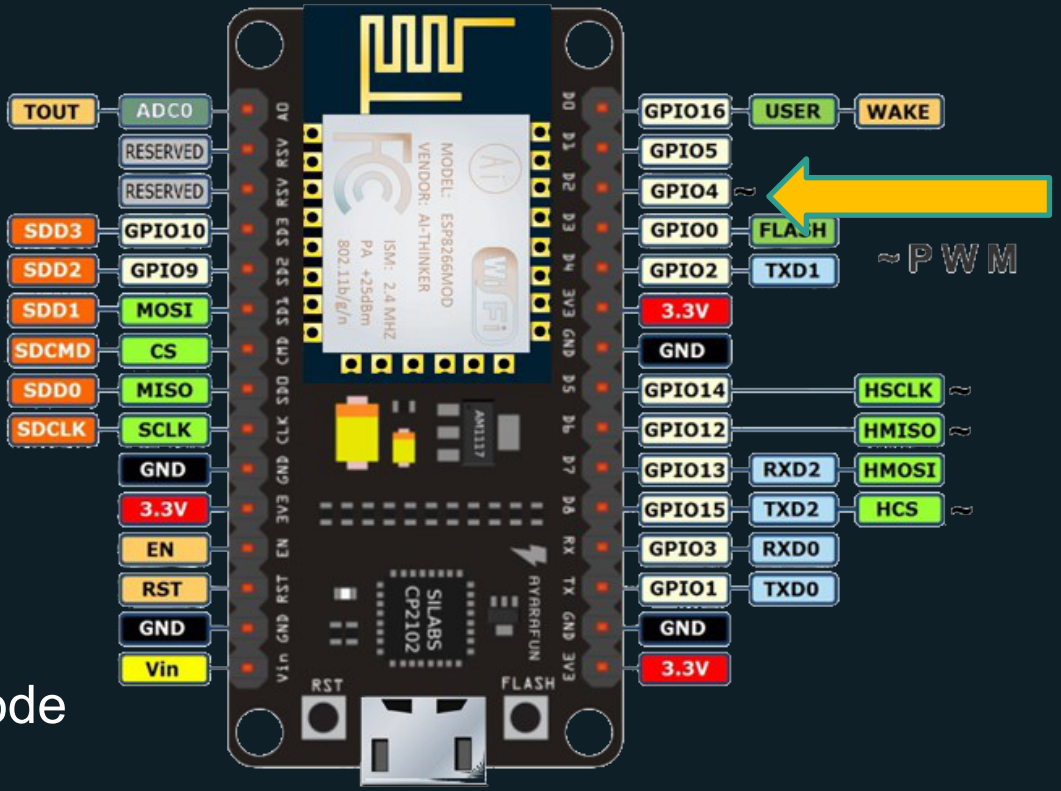
ESP8266 and DHT11 wiring

PIN on the esp8266

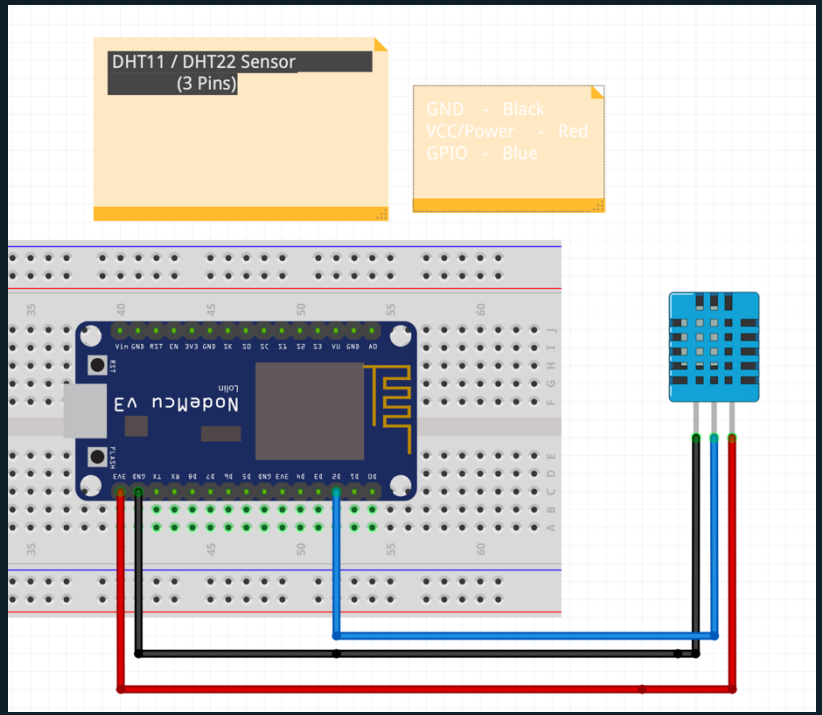
NOTE: we will be using
The GPIO numbers
NOT the PINs

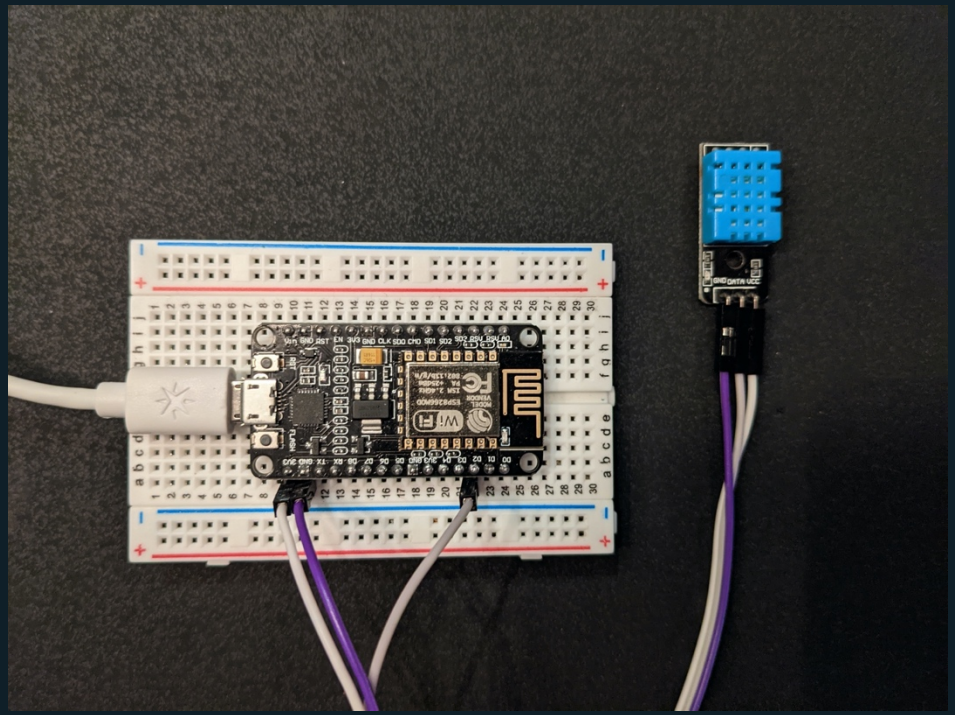
Like Pin D2 = GPIO4

We use number 4 in our code



ESP8266 & DHT11 Wiring





Code for esp8266 with dht11 - Number one

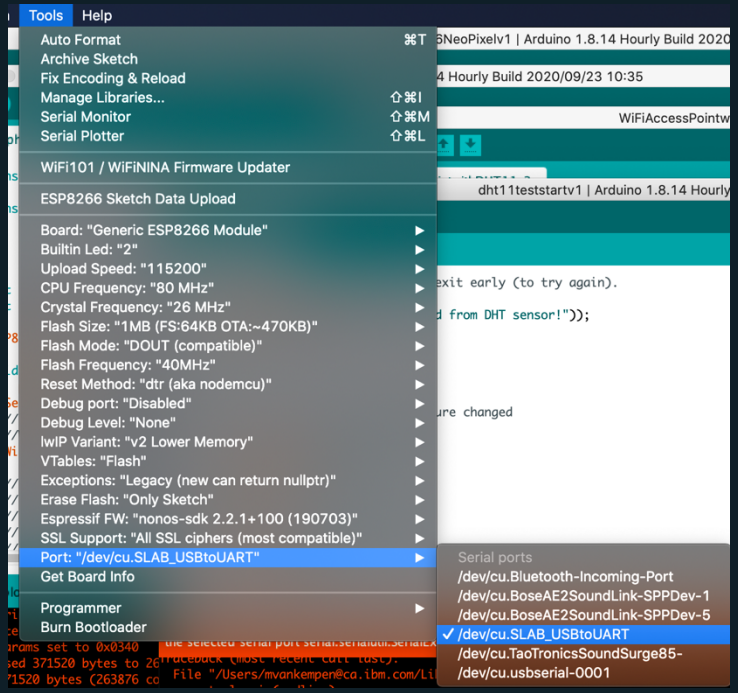
All code and instruction can be found here
https://github.com/markusvankempen/esp8266andDHT11

Here our 1st code to upload to the esp8266 which is connected to DH11

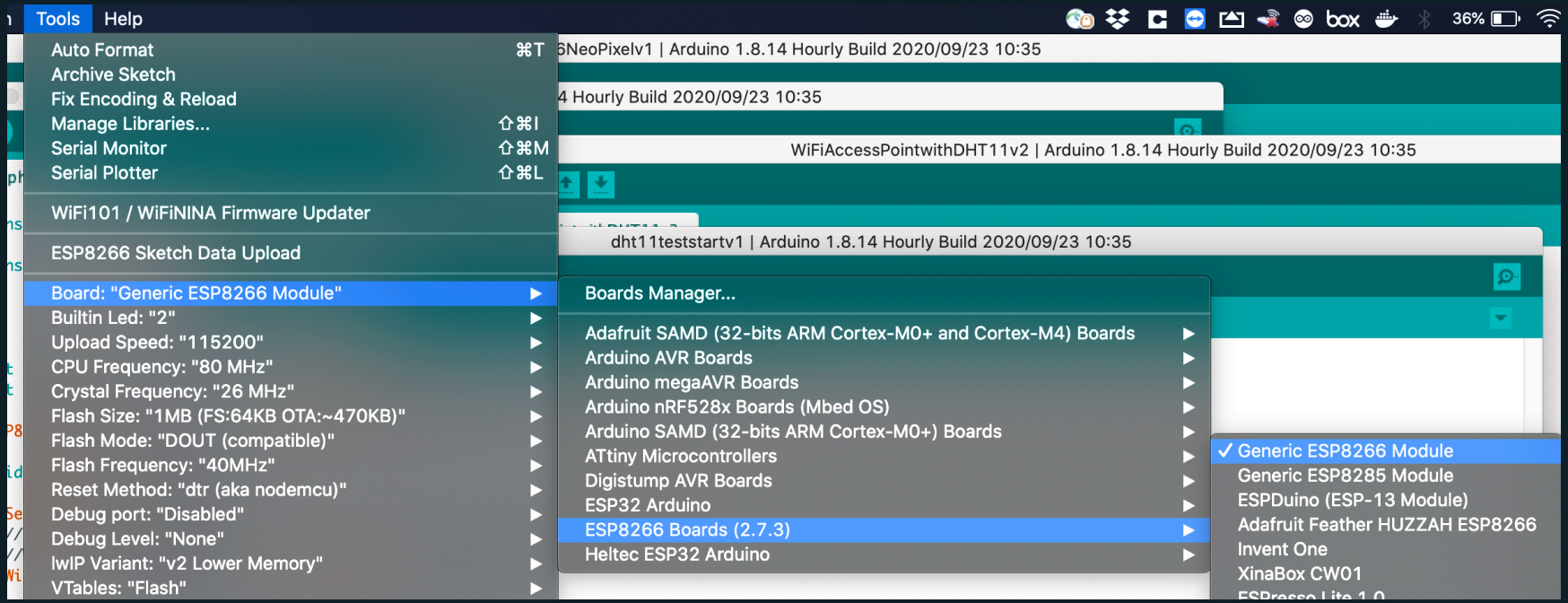
Note:

- Make sure u select the right board
- Copy the code into your IDE make sure to select the right serial Port

Serial Port Selection

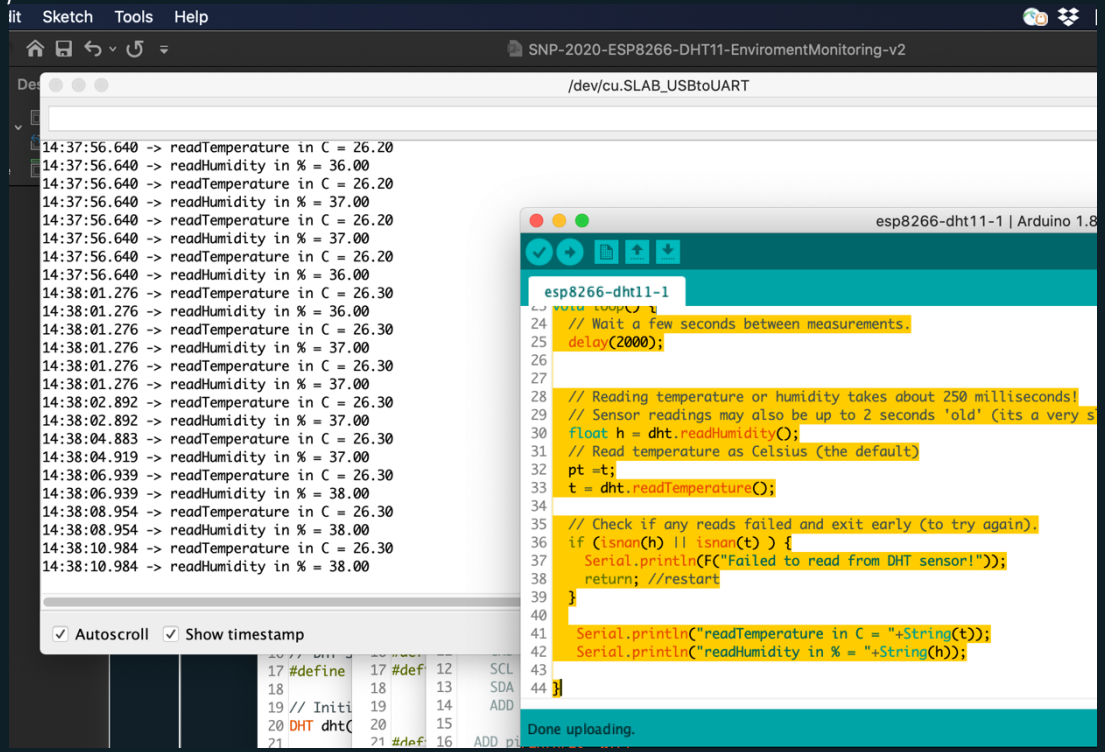


Board Selection



Upload the code to esp8266

Code :
https://raw.githubusercontent.com/markusvankempen/esp8266andDHT11/main/code/esp-dht11-1.ino
Once you upload and open the serial terminal /
make sure the Baudrate is 115200

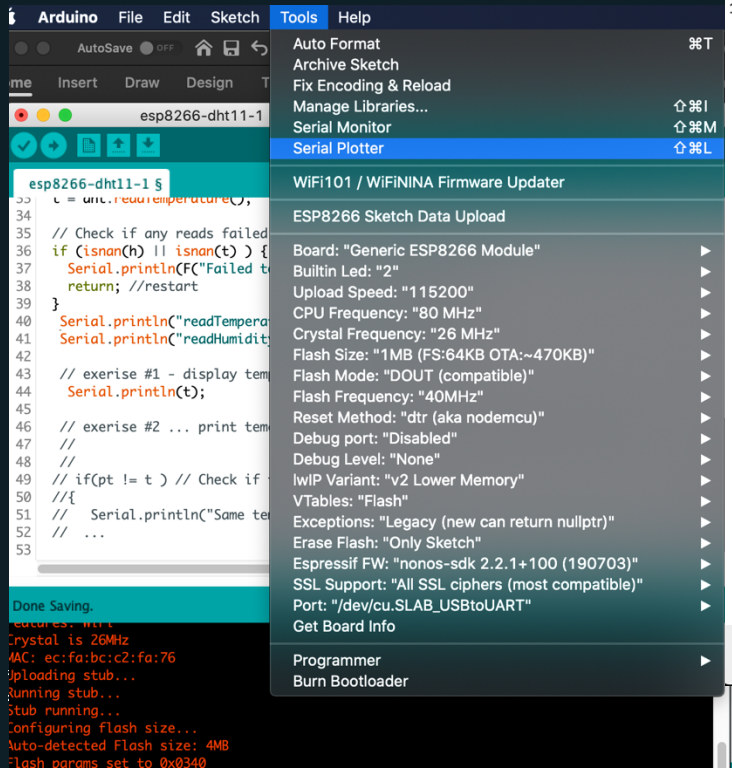


Exercise #1 … display temperature and Humidity in the Arduino IDE plotter

https://arduinogetstarted.com/tutorials/arduino-serial-plotter

Format is Label:Value Space Label: Value Space

```
Serial.print("readTemperature:"+String(t));
Serial.println(" readHumidity:"+String(h));
```



Exercise #2 … print temperature only if previous temperature has changes

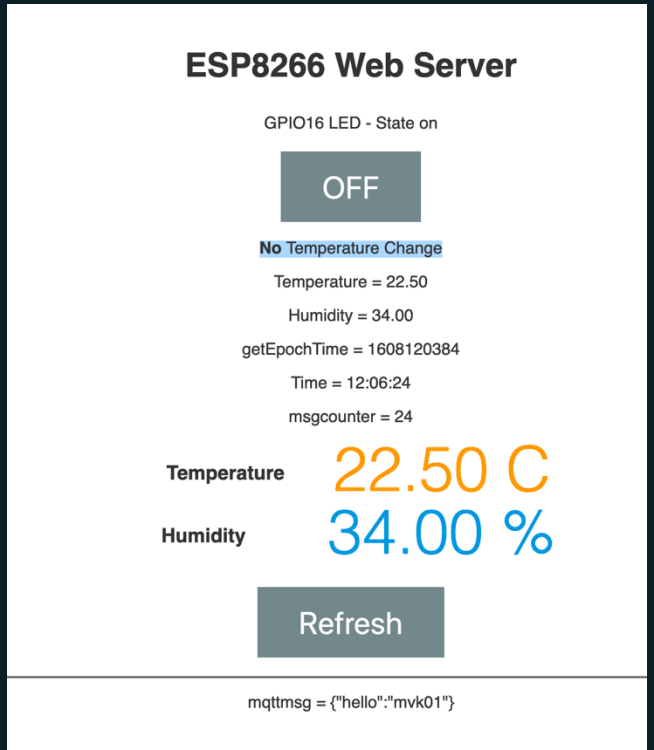
```
//
   //
  // if(pt != t ) // Check if temperature changed
  //{
  //   Serial.println("Same temp");
  //  ...
```

2nd Chapter (two)

Web Server

2nd Chapter (two)

Create a webserver and Display humidity and temperature and other data



Esp8266 and Wifi

Note:

The esp8266 and works as wifi access point – meaning you have to join the esp8266 wifi network to access a webpage

Or

The esp8266 can join your Wifi network which is what we will be using to display the humidity etc on a web pages

Esp8266 WebServer

In the code you need to adjust the SSID / password to your wifi credentials as well as personalize some other parameters – Note: 2.4 G wifi only

```
////>>>>>>>>>> CHANGE HERE
// Replace with YOUR network credentials
const char* ssid     = ”YOURSSID";    //>>>>>>>>>> CHANGE HERE
const char* password = ”YOURPASSWORD";        //>>>>>>>>>> CHANGE HERE
String DEVICEID   =    "markus01";           //>>>>>>>>>> CHANGE HERE

// find ur LAT LON  use https://www.latlong.net/ and ur cityname like Brantford,canada  for Branford =43.139410,-80.263650

String  LATITUDE  =   "43.6711581"   ;        //>>>>>>>>>> CHANGE HERE
String  LONGITUDE =   "-79.4129989"    ;      //>>>>>>>>>> CHANGE HERE
```

The Code and Deployment

Code can be found here

https://raw.githubusercontent.com/markusvankempen/esp8266andDHT11/main/2nd-Chapter/2-esp8266-webserver-with-dht11-c2b.ino

Note: If you can not upload the code via serial port maybe blocked just hit the rest button or unplug the device and select the serial port in the IDE again.

Upload the code and check the serial monitor to message

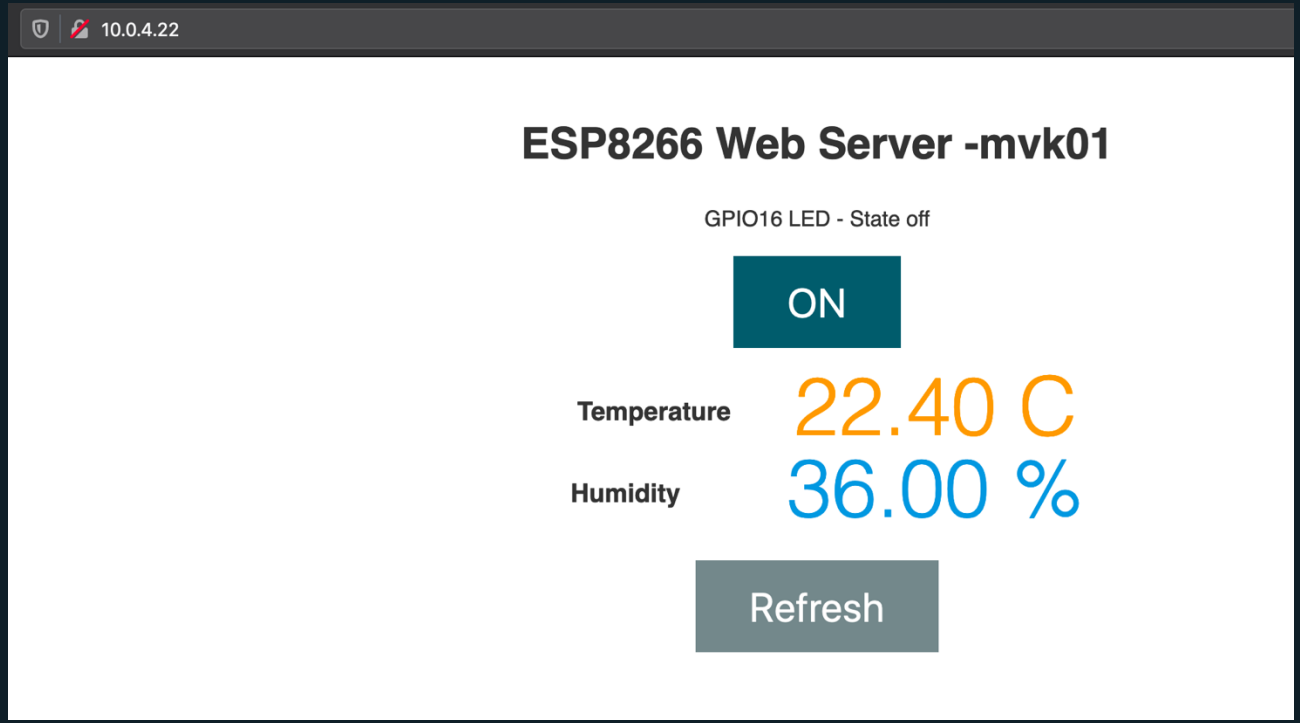
Serial Monitor should show

13:05:26.861 -> WiFi connected.
13:05:26.861 -> IP address:
13:05:26.861 -> 192.168.1.26 <<< Your IP address should be different

Use the ip address and open it in a browser like **http://192.168.1.26/**

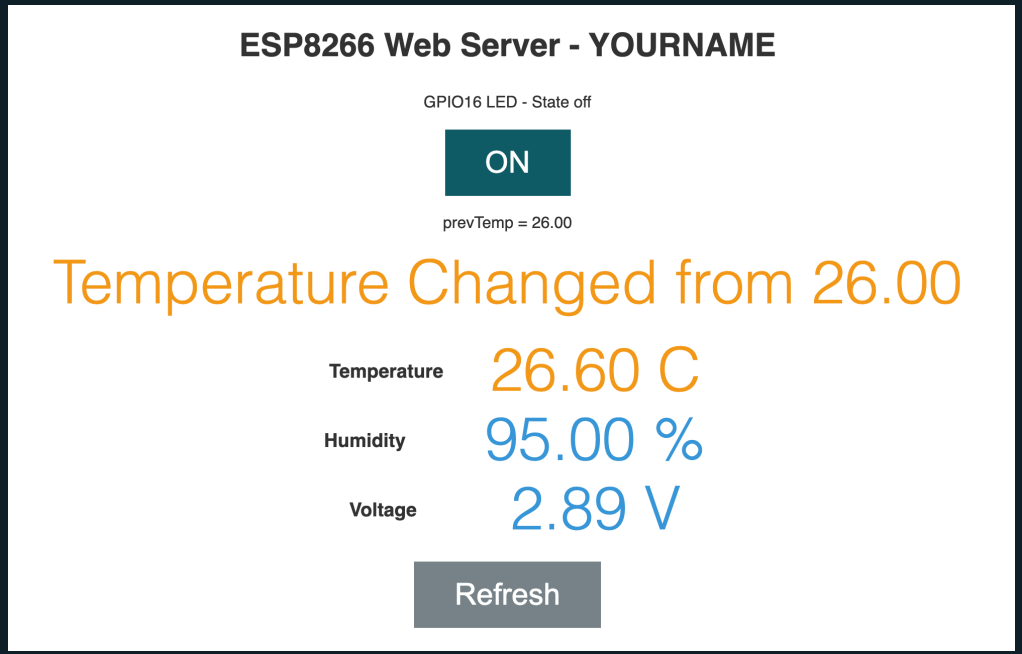
Display play the WebPage of the esp8266

http://your-ip-address - you should see



Exercise:

- Display the Voltage Parameter
- Display a message if the Temperature changes



3rd CHAPTER (three)

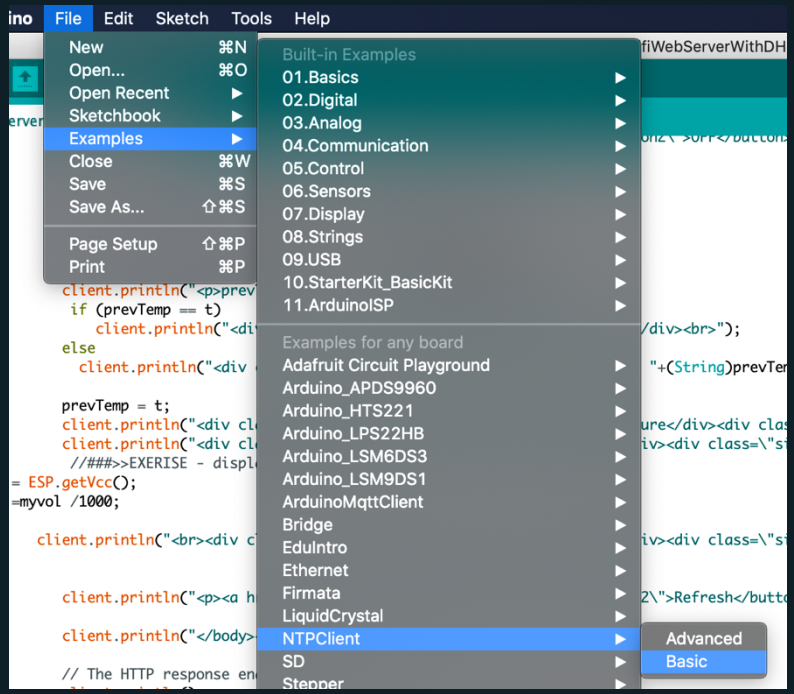
The esp8266 has no RealTimeClock aka RTC so we need to connect to a NTP(**Network Time Protocol)** Server to get the current time.

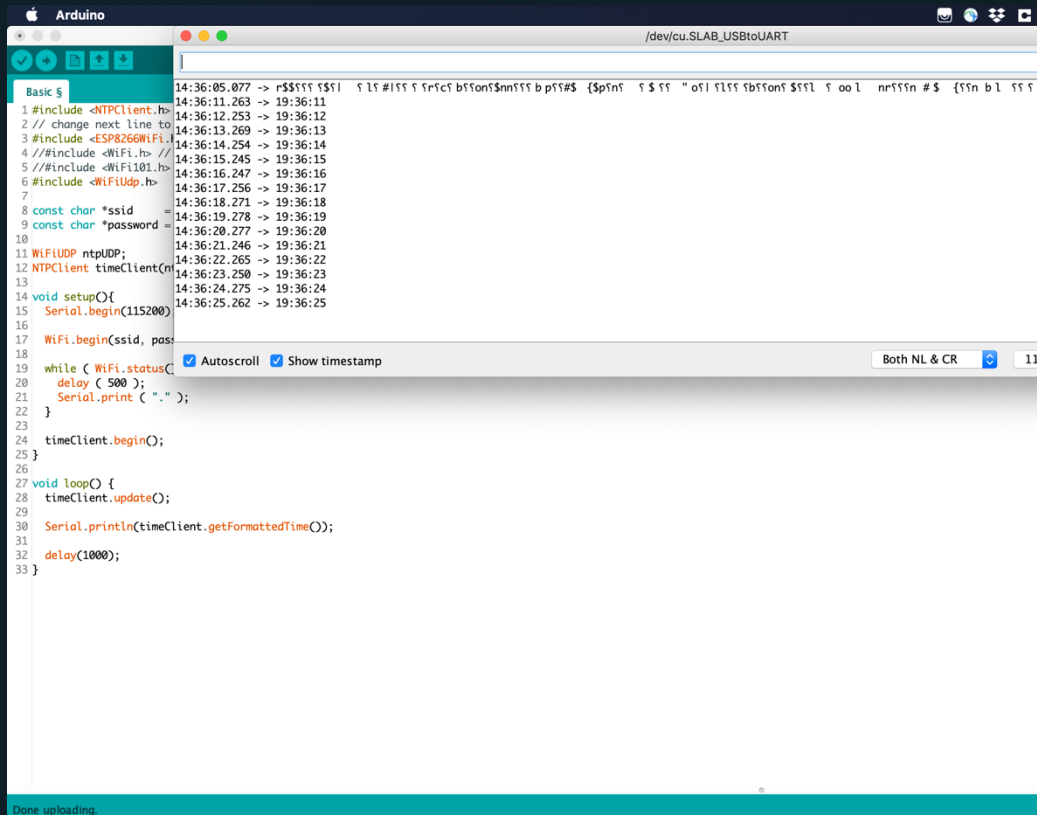
We need to load the NTPClient Library I the IDE

You can use the example code from the library for testing

File>Examples>NTPClient-Basic

Test NTP Client via the example code





NTP service

The NTP service will return the GMT/UTC time which is 5 Hour before EST
We will need to adjust the time using an offset

Detailed information here
https://github.com/arduino-libraries/NTPClient

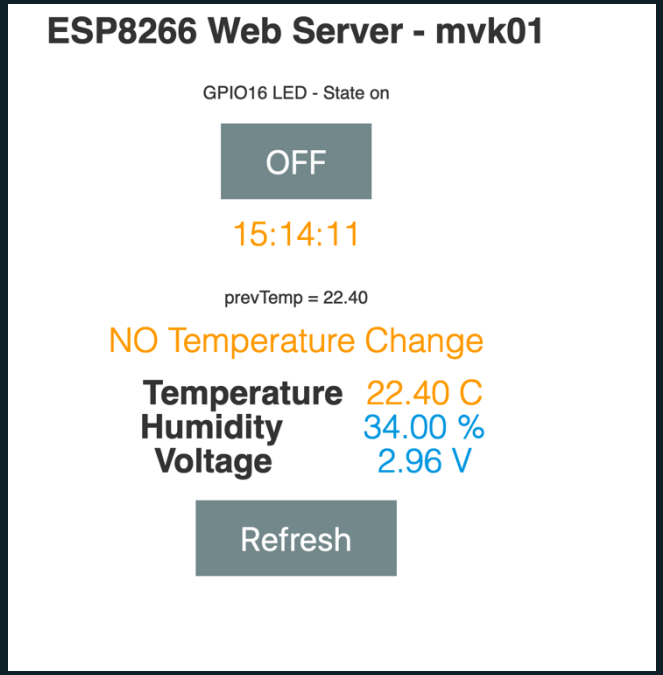
Adjust the code with the offset time in seconds e.g 5*60*60 =18000

EST Timezone time

NTPClient timeClient(ntpUDP, "pool.ntp.org", -18000, 60000);

Exercise

Add NTP code to the WebServer and display the time
On the WebPage

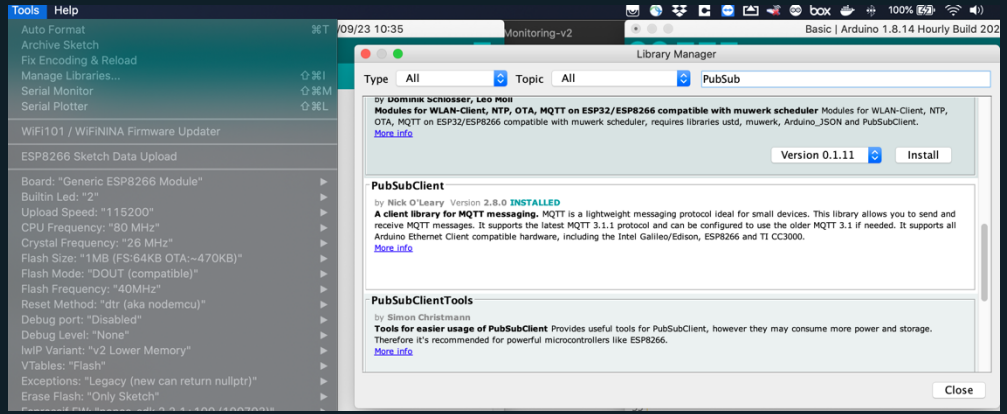


Code here
https://raw.githubusercontent.com/markusvankempen/esp8266andDHT11/main/3rd%20Chapter/3-WifiWebServerWithDHT11AndNTP.ino

4th CHAPTER (four)

In this section we will add a mqtt client to our WebServer and send a json message with temperature, voltage etc to a mqtt broker.

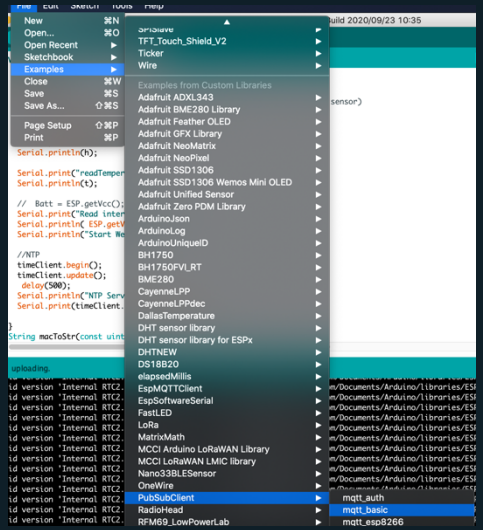
We will need to add a mqtt client library called PubSubClient



MQTT Client and Message Publishing

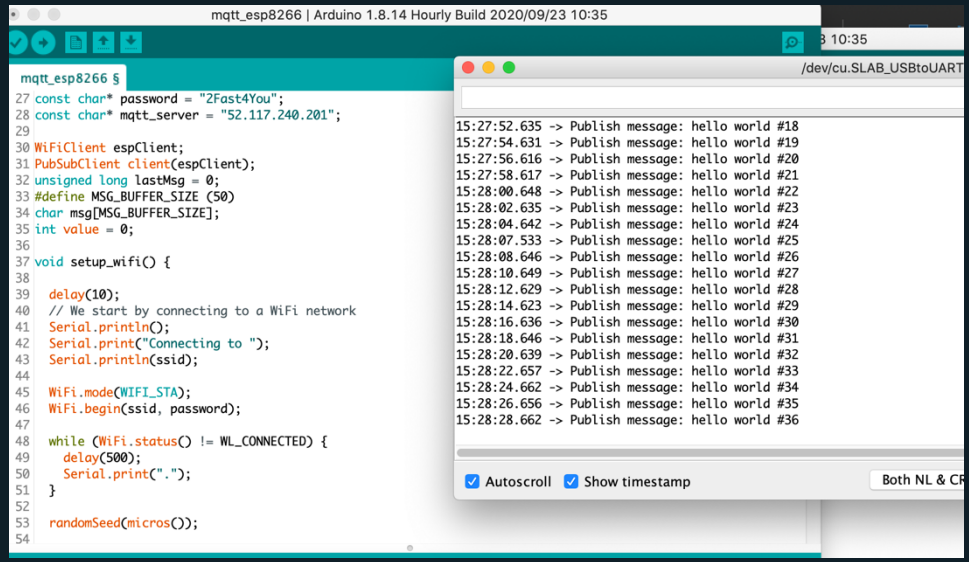
You can use the mqtt client example to send a test message to our mqtt broker (which is a docker on our virtual Server) once you add the library to the IDE. The Library is called PubSubClient.h

The example Code is under Examples – choose esp8266 example



Test MQTT Message Publishing

Add the our Mqtt broker IP
const char* mqtt_server = "52.117.240.201";
And the topic or message



Example code also here:
https://raw.githubusercontent.com/markusvankempen/esp8266andDHT11/main/4th%20Chapter/4-mqtt-esp8266-test.ino

Check if message are received by the mqtt broker

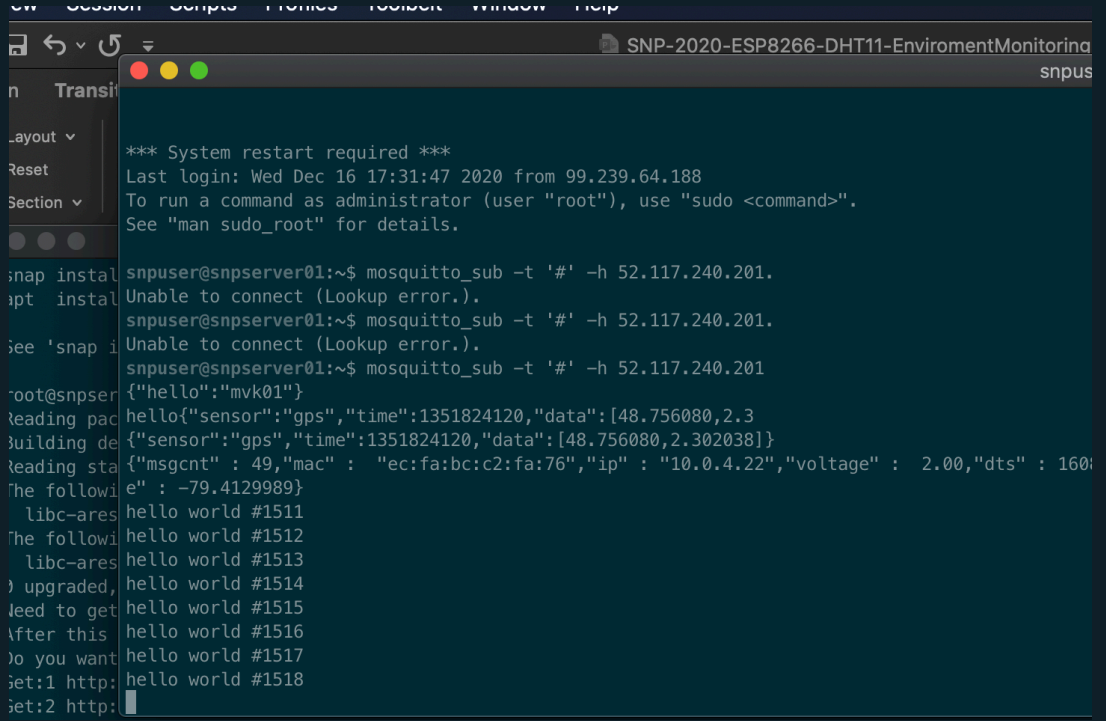
You can check if mqtt messages are received on the server using a mosquito client. U can install mosquito client locally or use the server via ssh.

Example:

ssh snpuser@52.117.240.201

Or snpuser@snpweather.xyz

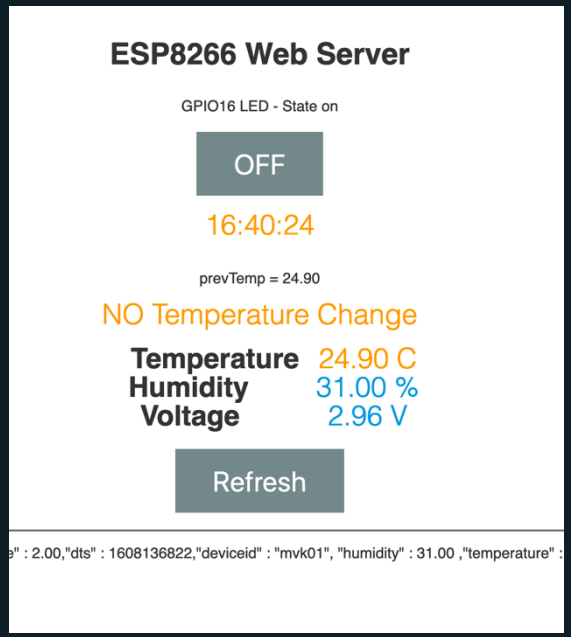
mosquitto_sub -v -t "sensor/YOURDEVICEID/#"



Add the mqtt client code to our WebServer

The code for the webserver,ntp and mqtt client is here

Code is here:




```
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136708,"deviceid" : "mvk01", "humidity" : 33.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136718,"deviceid" : "mvk01", "humidity" : 33.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136728,"deviceid" : "mvk01", "humidity" : 33.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136738,"deviceid" : "mvk01", "humidity" : 32.00 ,"temperature"
9}
"hello":"mvk01"}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136782,"deviceid" : "mvk01", "humidity" : 34.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136792,"deviceid" : "mvk01", "humidity" : 32.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136802,"deviceid" : "mvk01", "humidity" : 32.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136812,"deviceid" : "mvk01", "humidity" : 32.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136822,"deviceid" : "mvk01", "humidity" : 31.00 ,"temperature"
9}
"hello":"mvk01"}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136861,"deviceid" : "mvk01", "humidity" : 34.00 ,"temperature"
9}
"mac" :  "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" :  2.00,"dts" : 1608136871,"deviceid" : "mvk01", "humidity" : 31.00 ,"temperature"
9}
```

WebServer With NTP and MQTT client

The WebServer is now getting the current time/date and send mqtt with temperature , humidity etc. as json to the broker

:31:08.584 -> Sending Sensor Data

16:31:08.584 -> {"msgcnt" : 59,"mac" : "ec:fa:bc:c2:fa:76","ip" : "10.0.4.22","voltage" : 2.00,"dts" : 1608136267,"deviceid" : "mvk01", "humidity" : 36.00 ,"temperature" : 22.80 ,"latitude" :43.6711581,"logitude" : -79.4129989}

Exercise

Add the message counter (see field msgcounter) to the json message and validate the message is received by the broker.

\" msg\" : "+String(msgcounter)+ ”

Exercise – Enable/Test Deep sleep (optional)

Our device are eventually powered by batteries/Solarpanels and therefore we want to conserve energy and shutdown/ go into deepsleep if there is nothing to do. We also only want to send if the has changed.

Enable the code deepsleep code and check if the wifi connection is available on wake up!

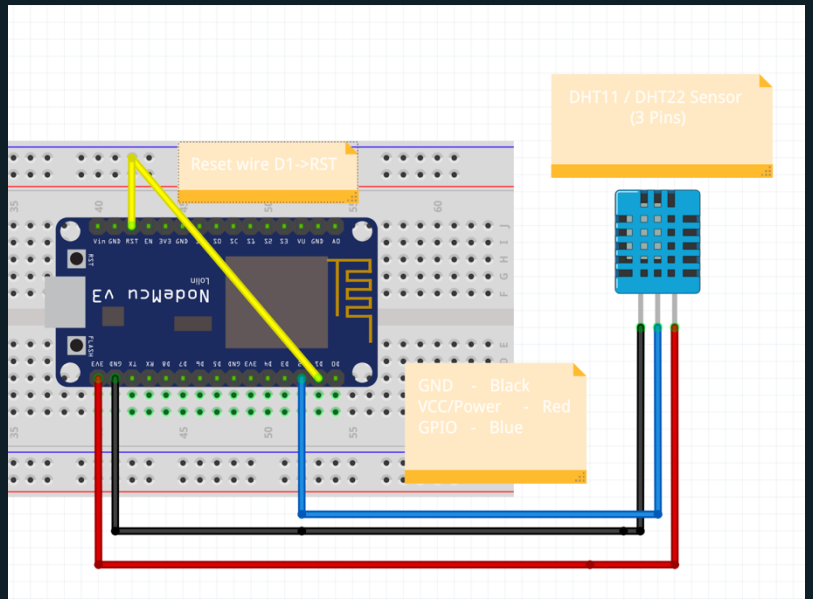
Exercise – Soft vs Hard reset (optional)

Your can force a reboot via ESP.reset()/ESP.reboot().

You can also use the Digital pin to enable Hard reset like via D1->RST

```
pinMode(D1, OUTPUT);
digitalWrite(D1, HIGH);  //D! -> RESET PIN via a wire
```

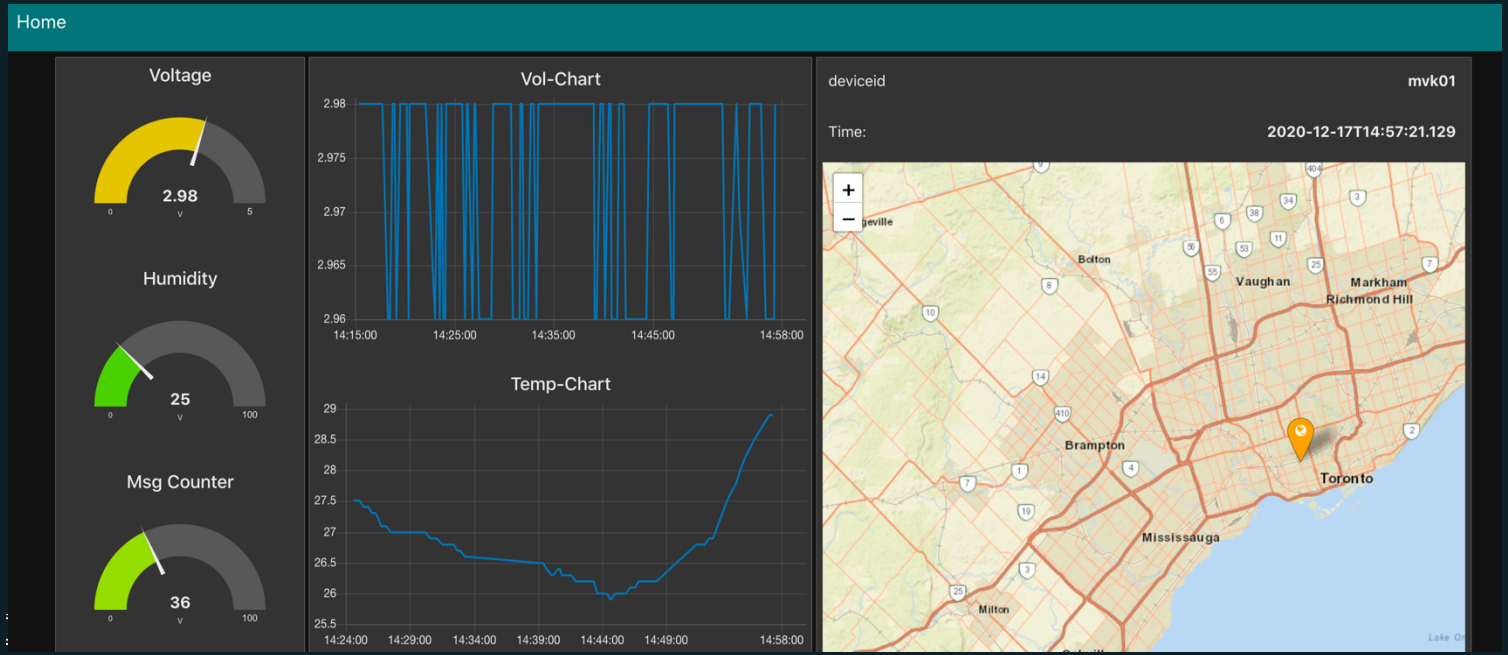
- digitalWrite(D1, LOW); // =>RESET



IBM

5th CHAPTER – Real time Visualization

Now we have the mqtt message flowing to the server id would be nice to see visualize the data in a real time dash board



Import missing plugins

Like

node-red-contrib-web-worldmap

Via Menu -> Manage Pallet

Real Time Dashboard

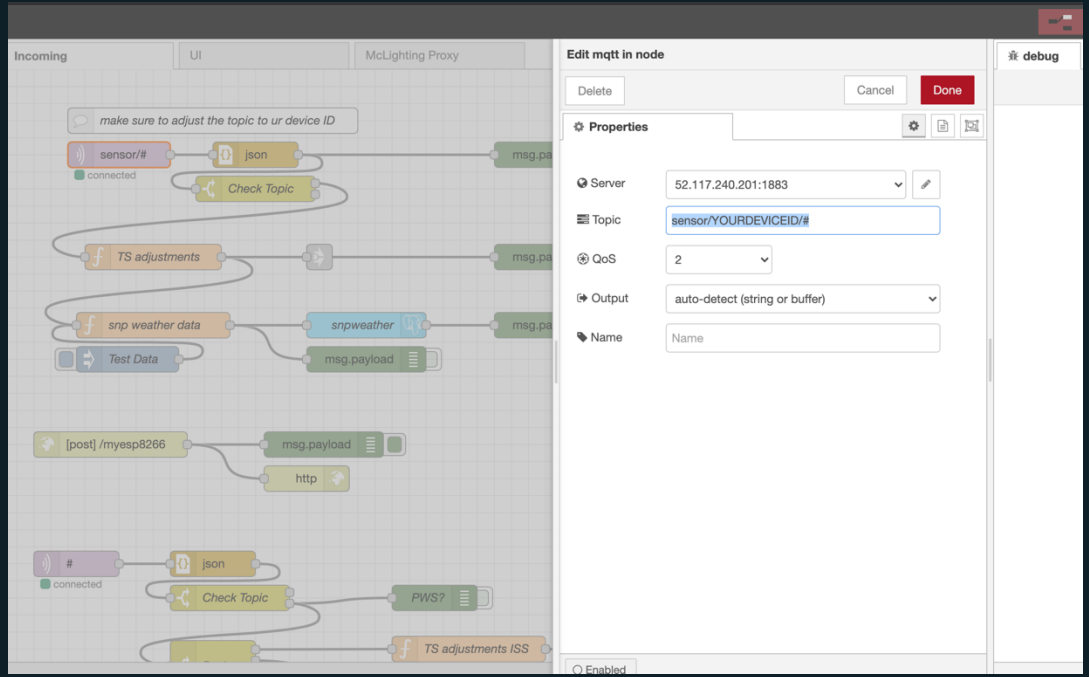
Choose one of Node-RED instance http://52.117.240.201:1000 to http://52.117.240.201:1007

- Import the node-red flow from here
- https://raw.githubusercontent.com/markusvankempen/esp8266andDHT11/main/5th%20Chapter/node-red.flow

- Adjust the deviceid in the mqtt topic to your deviceid
- Add the missing humidity charts and other message data to the dashboard
- Map all device from all students on your map
- You can access the

Node-RED flow

Make sure to adjust the Topic with Your Device ID



6th CAPTER (6) – Saving the data to an SQL database

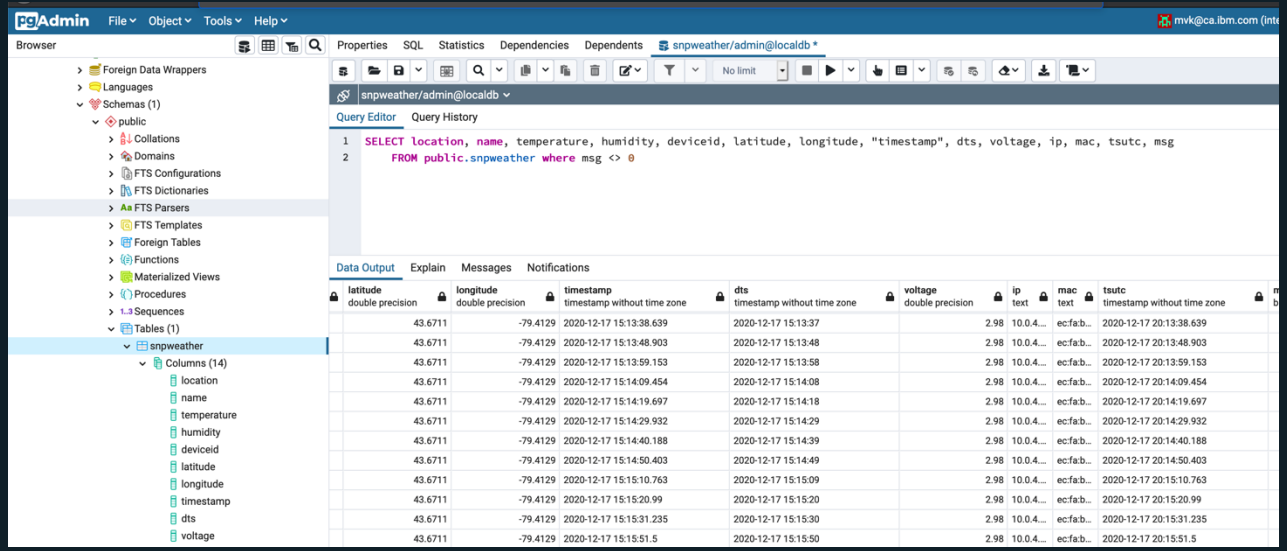
The server has a posgres sql database docker instance which we can use to store the sensor data.

The admin tool is on port 80 and is called pgAdmin

http://52.117.240.201/browser/

Time series database

The database “snpweather” is already created and matches the json mqtt message format. We will walk you thru the admin tool and show you the fields and existing data.

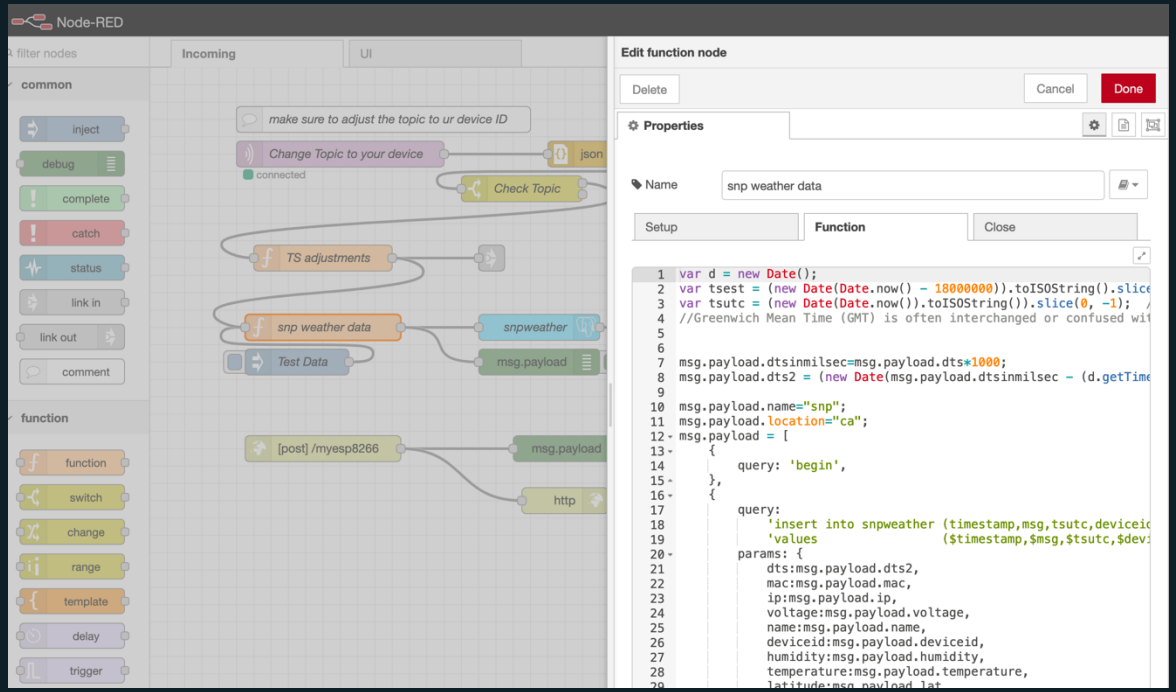


Storing sensor database in a time series database

We will use Node-RED to store the data with the mqtt topic sensor/YOURDEVICEID/data into the database. The flow is already there and we can add some missing fields to the table.

You may have to connect some of the flows and deploy the changes.

Storing the data
The PostGres Plugin has a nice SQL function to map and store the data



Exercise

Add some of the missing fields like msg counter and mac address to same them into the database.

Add the mapping

ip:msg.payload.ip,

mac:msg.payload.mac,

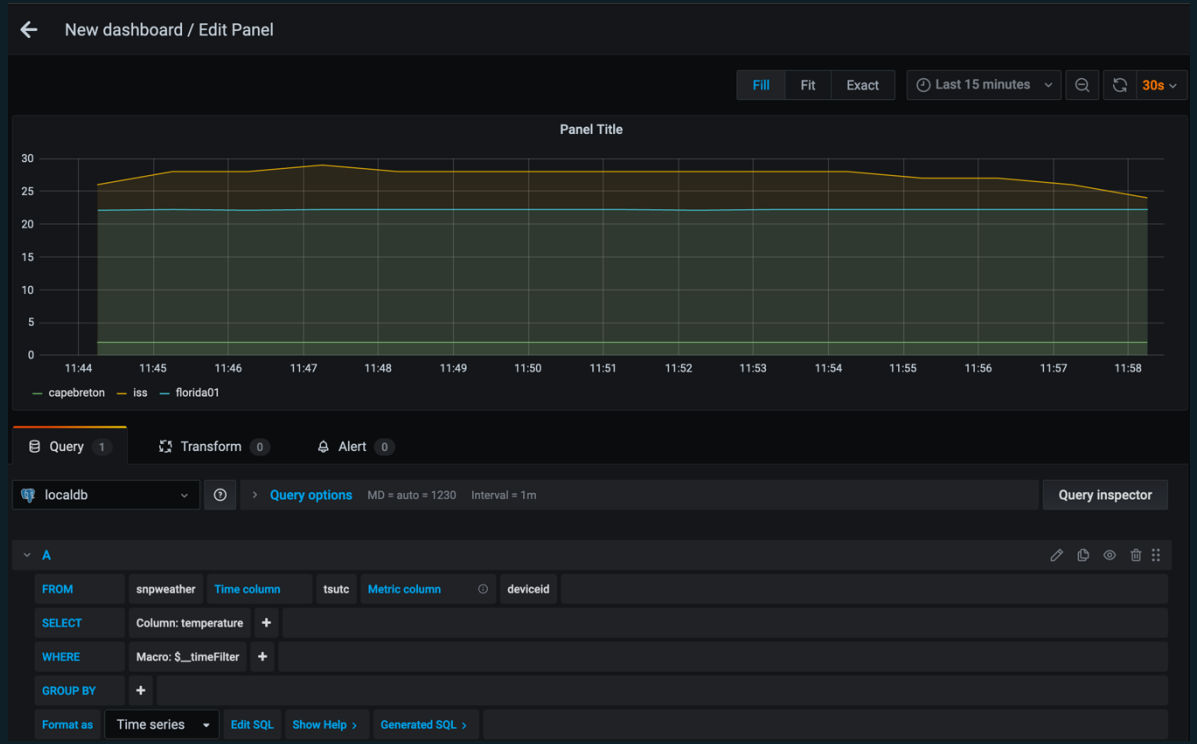
7th CHAPTER (7) - Visualizing Time series Data

This this chapter we will visualize the timeseries Data using Grafana

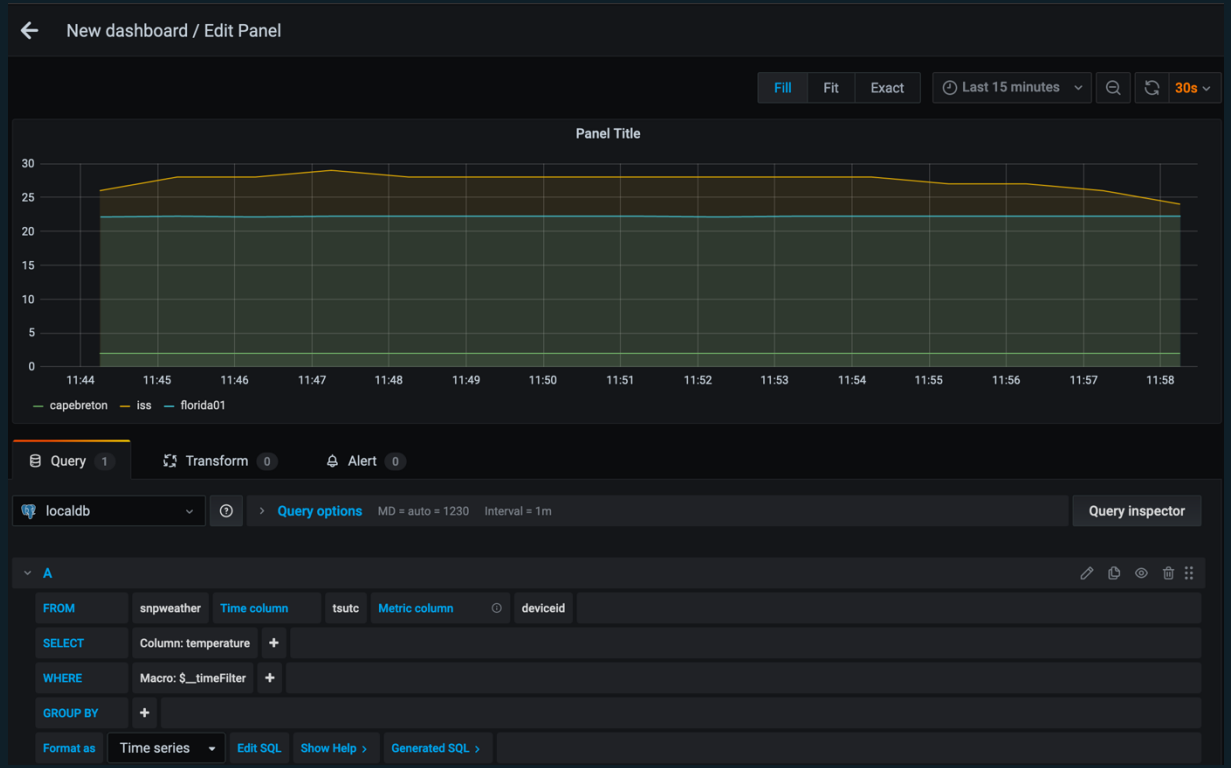
Grafana is installed on our virtual server on port 3000
http://52.117.240.201:3000/

We have user1-6 setup to create dashboard.

Exercise – Create a dashboard which show the Temperature walkthru



Exercise – create a Dashboard which show all device on a Map



Review

- We create a WebServer using an esp8266 with DHT11
- We use an NTP client to get the right time
- We send a json message to an mqtt broker
- We added some esp8266 deepsleep and reset functionality
- We displayed the DHT11 in a Node-RED realtime Dashboard
- We stored the Data into a SQL Database as a time series
- We visualized the time series data using grafana

Questions ?


IBM