

# Environment Monitoring with LoRaWAN



Markus van Kempen  
IBM **SPEED**

✉ [mvk@ca.ibm.com](mailto:mvk@ca.ibm.com)

🐦 @markusvankempen

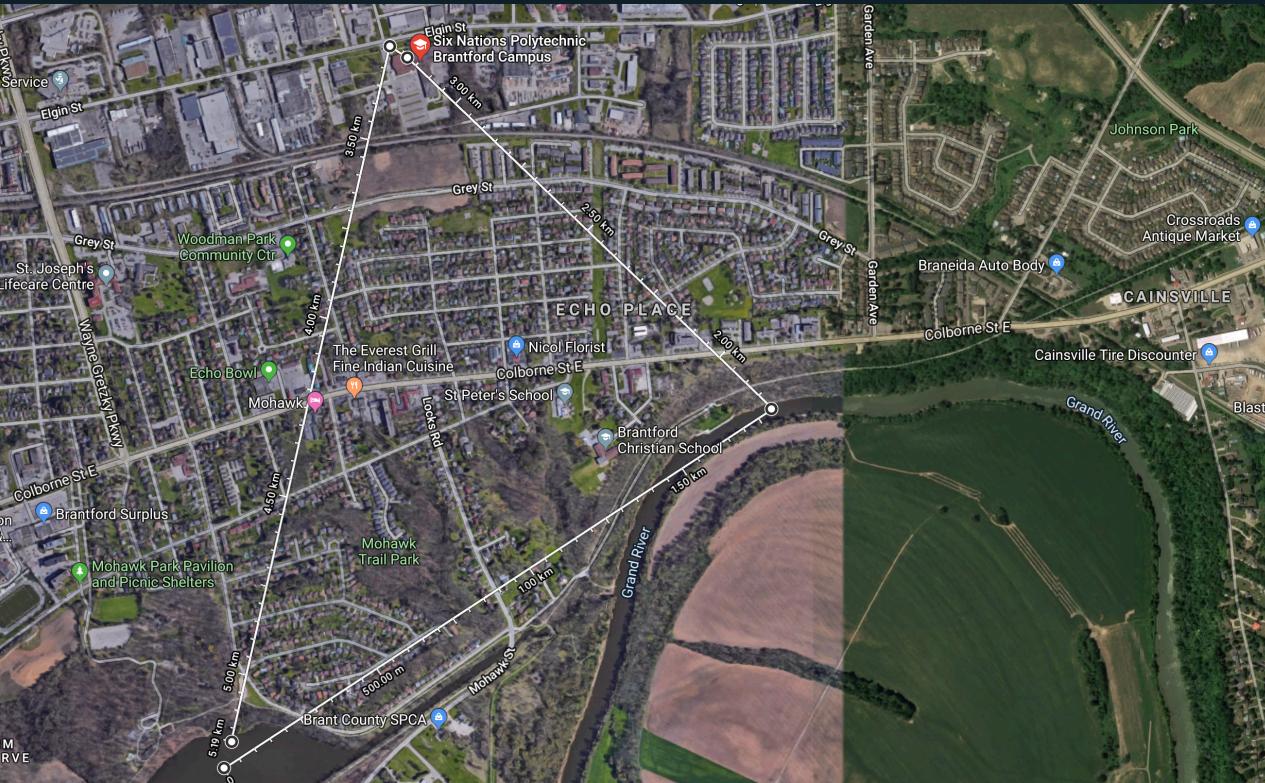
# SNP STEAM ACADEMY

Code and instruction for this class can be found here

<https://github.com/markusvankempen/esp8266andDHT11>



# Our Goal – Environmental Monitoring



## Why LoraWAN

- It's free - not Telco costs
- LORA = LongRange 10km and LowPower (LPWan)
- Lot of use case example 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>



## Local Area Network

Short Range  
Communication



**40%**



Well established standards  
In building



Battery life  
Provisioning  
Network cost & dependencies



## Low Power Wide Area (LPWAN)

Internet of Things

**45%**

Low power consumption  
Low cost  
Positioning

High data rate  
Emerging standards



## Cellular Network

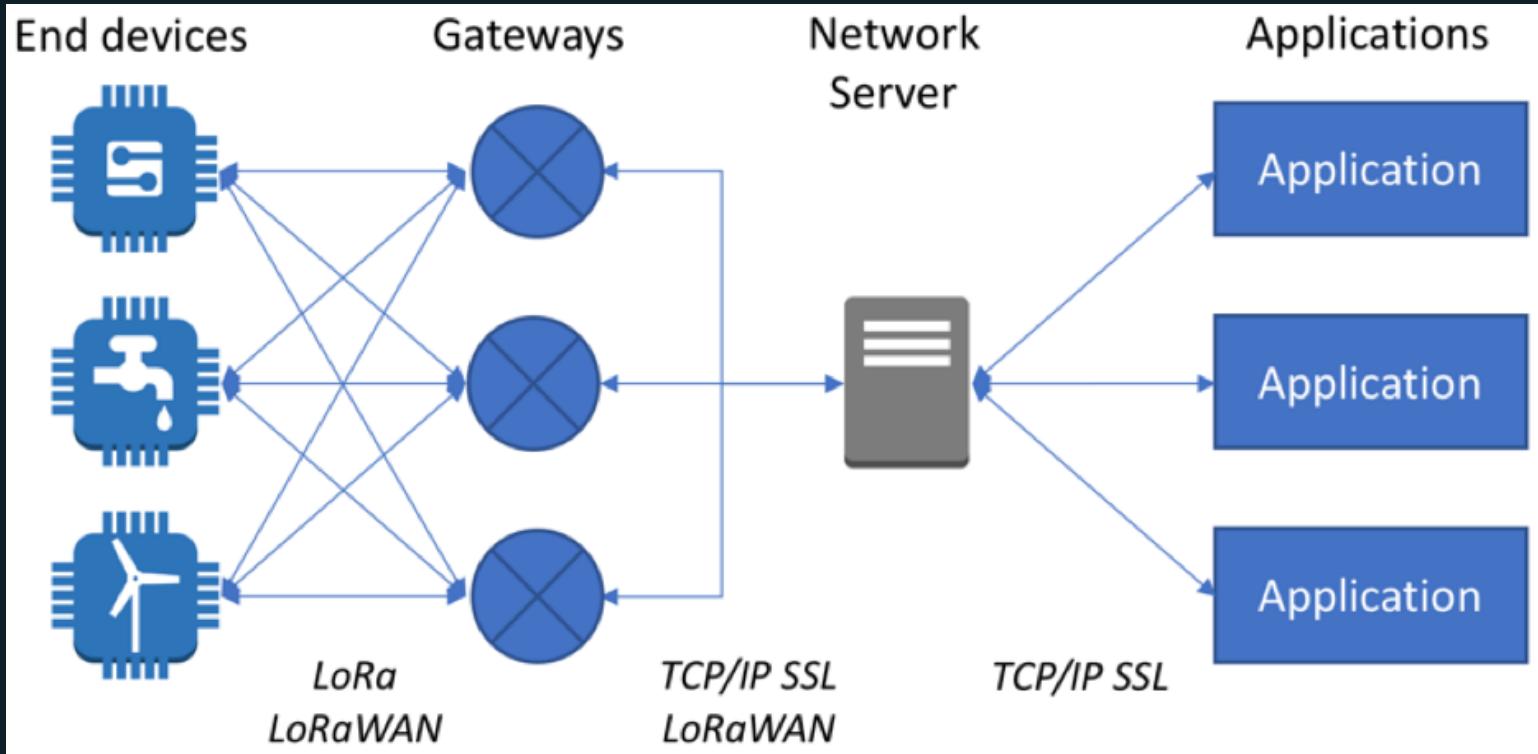
Traditional  
M2M

**15%**

Existing coverage  
High data rate

Autonomy  
Total cost of ownership





# Lora Gateway & LoraServer



- Has Wifi/lora/GPS/LTE
- Power via POE
- More infos here
- <https://www.rakwireless.com/en-us/products/lpwlan-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>



Status

Overview

LoRa Packet Logger

System Log

Firewall

Network

LoRa Gateway

LoRa Network Server

Services

System

## LoRaWAN Packet Logger

## LoRaWAN Packet Logger

Type All DevAddr Hide CRC\_ERR packet

Total : 3404 Uplink : 3382 Downlink : 22

Pause

Time Freq RSSI SNR TxPwr CRC mod. CR DataRate FCnt AirTime DevAddr FPort Payload Size MAC Command

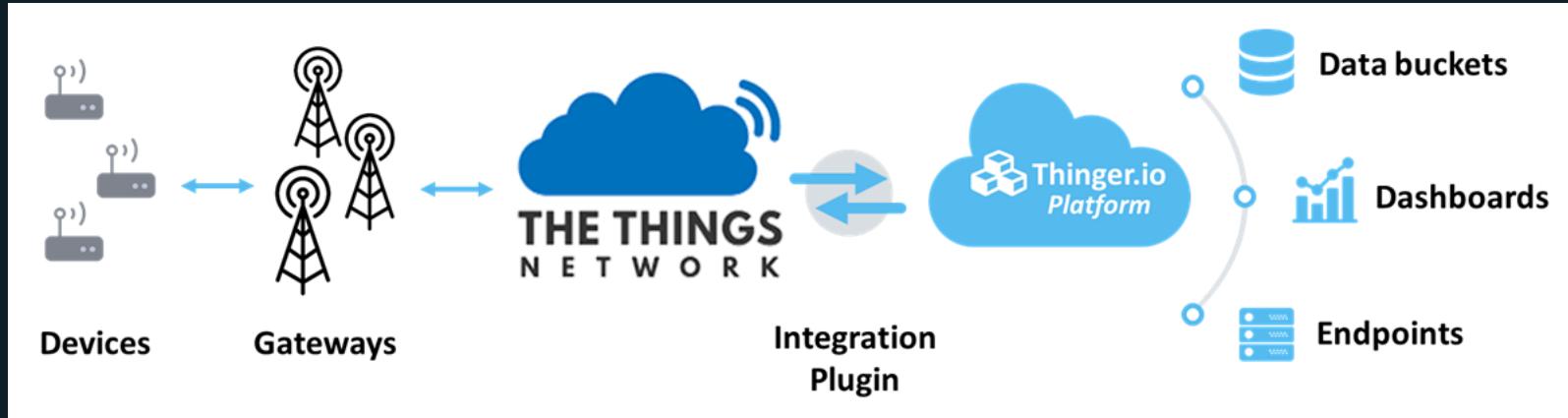
▲ 19:13:09	904.1	-74	10.8	-	CRC_OK	LORA	4/5	SF7BW125	1	46	2601352A	1	2	-
▲ 19:13:03	904.5	-71	9	-	CRC_OK	LORA	4/5	SF7BW125	0	46	2601352A	1	2	-
▼ 19:12:47	927.5	-	-	20	CRC	LORA	4/5	SF9BW500	1259	82	02000005	10	40	-

```
{ "freq": 927500000, "mode": "timersigned", "tmsr": 917017116, "rfch": 0, "powr": 20, "prea": 8, "ncre": false, "modu": "LORA", "datr": "SF9BW500", "code": "4/5", "ipol": true, "size": 53, "data": "YAUAAAKA6wQKpZiFxkFgNSAMlxk0j6LLXf/BR2E6YPxiuzccsrwsY6uu47I60qmmo1Q8G6E=" }, { "MHDR": { "MType": "Unconfirmed Data Down", "RFU": 0, "Major": 0 }, "MACPayload": { "FHDR": { "DevAddr": "02000005", "Fctrl": { "ADR": true, "RFU": 0, "Pending": false, "ACK": false, "FOptsLen": 0 }, "FCnt": 1259 }, "FPort": 10, "FRMPayload": "0A A5 98 85 C6 41 60 59 20 0C 97 19 0E 8F A6 62 BB 37 1C B2 BC 2C 63 AB AE E3 B2 3A D2 A9 A6 " }, "MIC": "543C1BA1" }
```

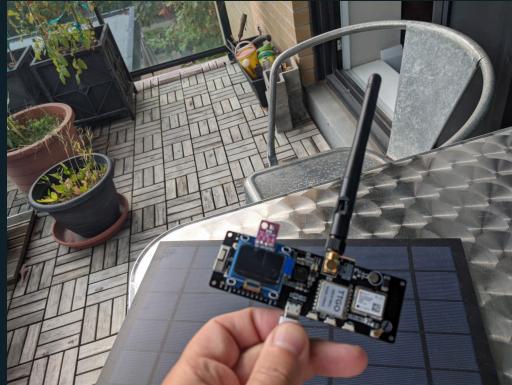


## What is TTN / TheThingsNetwork - <https://www.thethingsnetwork.org/>

We can use this as a cloud provider or we can use the GW lora server or we can forward the data via mqtt ...lots of options



# Lora Nodes – I test several with batteries and solar panel – TTGO -T-Beam



TTGO ... had 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&FlId=t3:50003:3>

[https://github.com/JoepSchyns/Low\\_power\\_TTGO\\_T-beam](https://github.com/JoepSchyns/Low_power_TTGO_T-beam)

<https://github.com/LilyGO/TTGO-T-Beam>

[https://tinyomics.com/wiki/TTGO\\_T-Beam](https://tinyomics.com/wiki/TTGO_T-Beam)

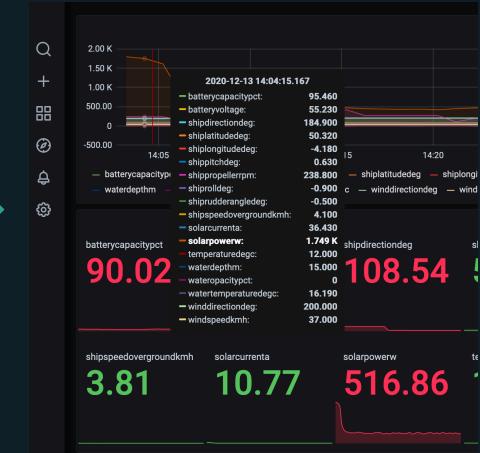
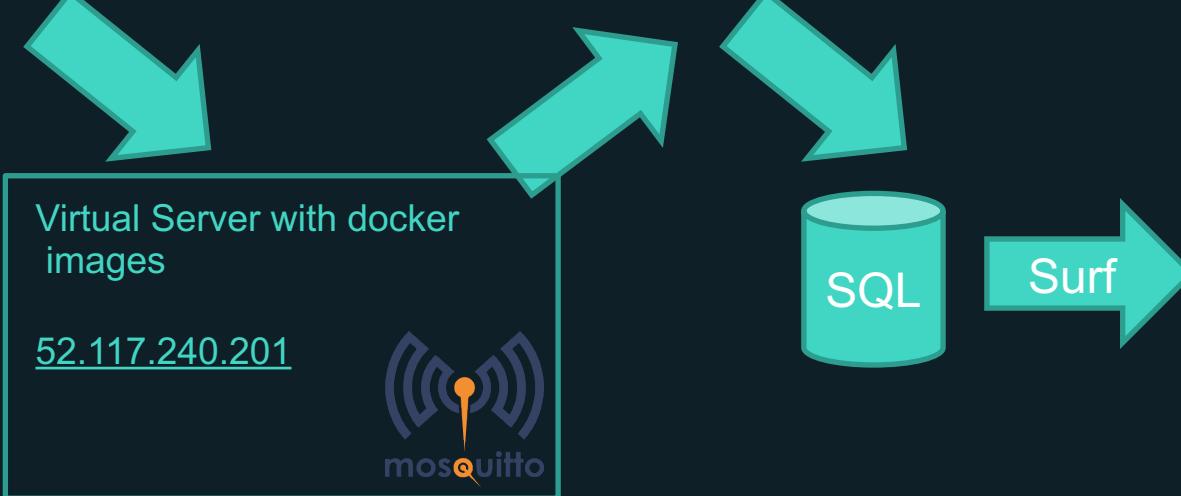
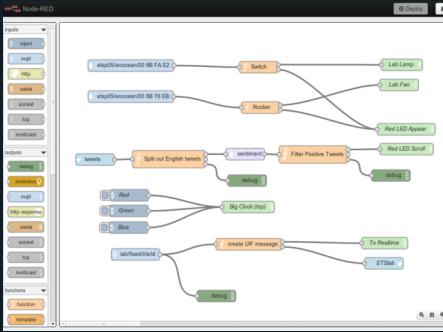
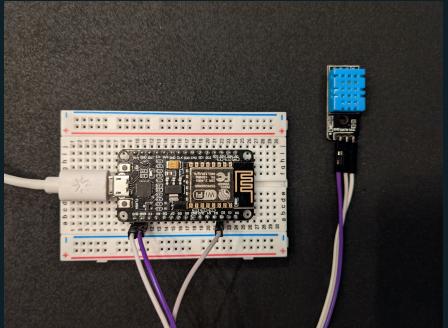


## Today

We will simulate the Lora network using wifi with our esp8266 & DHT1 to send temperature data via mqtt to a server and save the data into a database and than “surf” the data.



# Class Overview



# Chapter One (1)

Arduino IDE setup

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](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

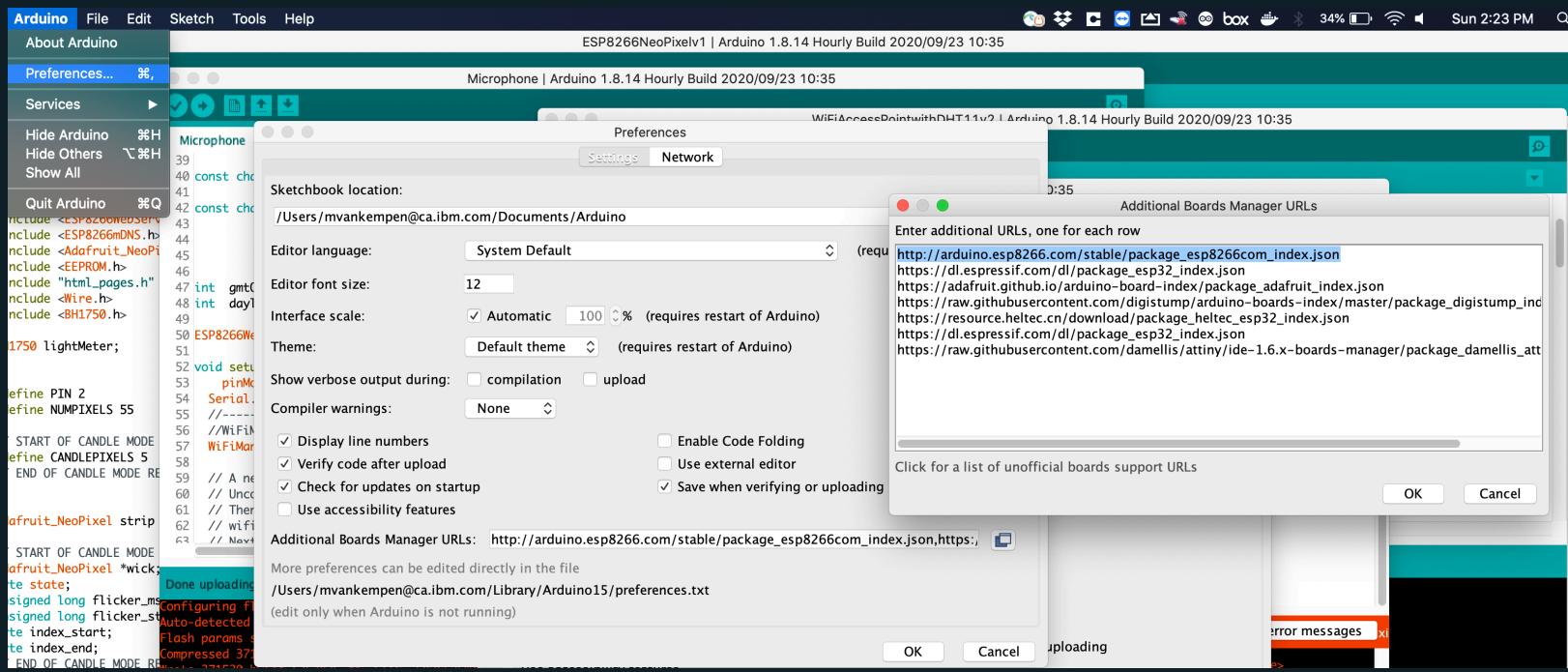
## More reference

<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 it's necessary to install our DHT library, which can be done through 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

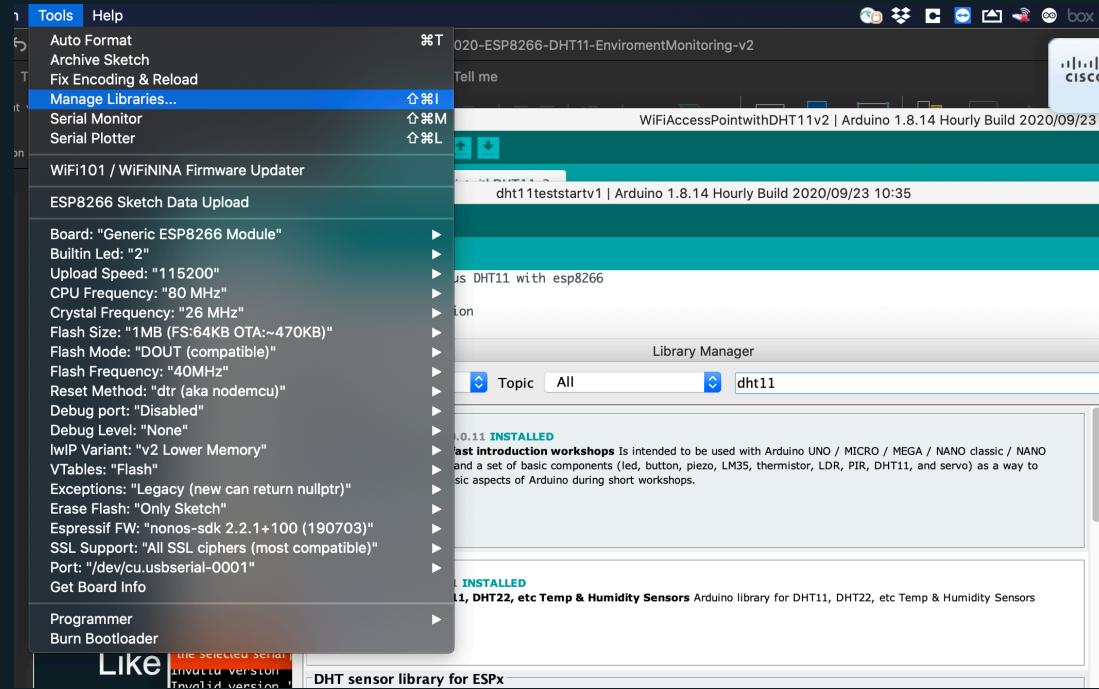
<https://arduino-esp8266.readthedocs.io/en/latest/installing.html>

<https://www.hackster.io/rajthavti/dht11-sensor-interfacing-with-nodemcu-086762>

<https://learn.adafruit.com/dht/downloads?view=all>



# Import DHT11 Library



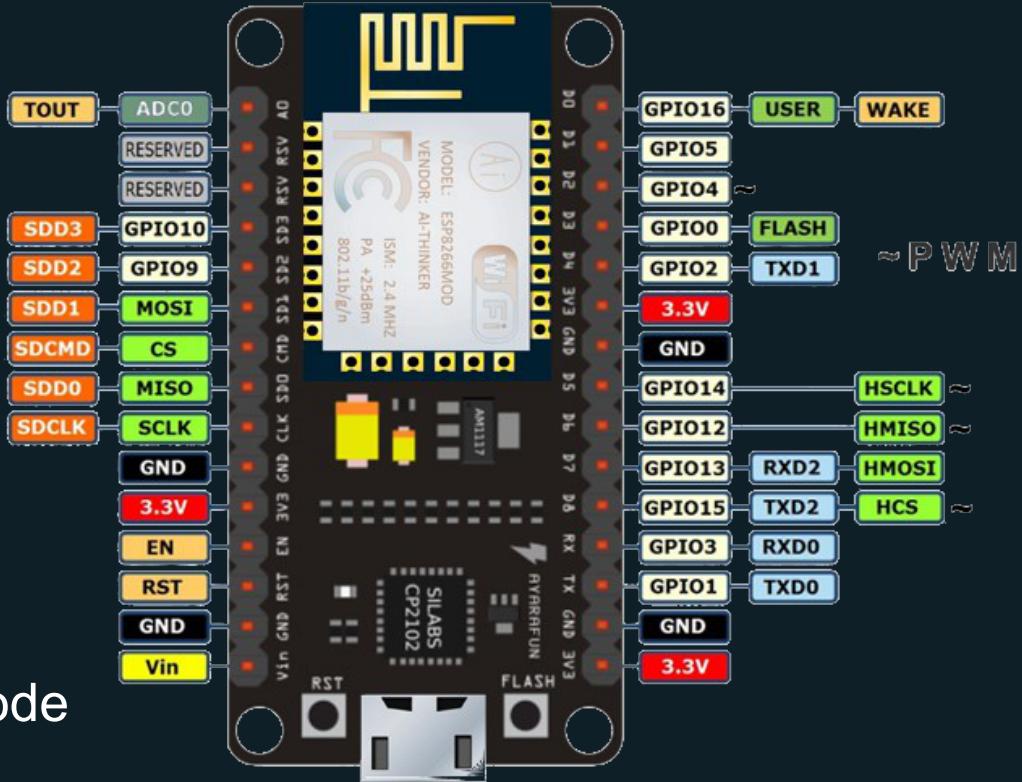
# ESP8266 and DHT11 wiring

PIN on the esp8266

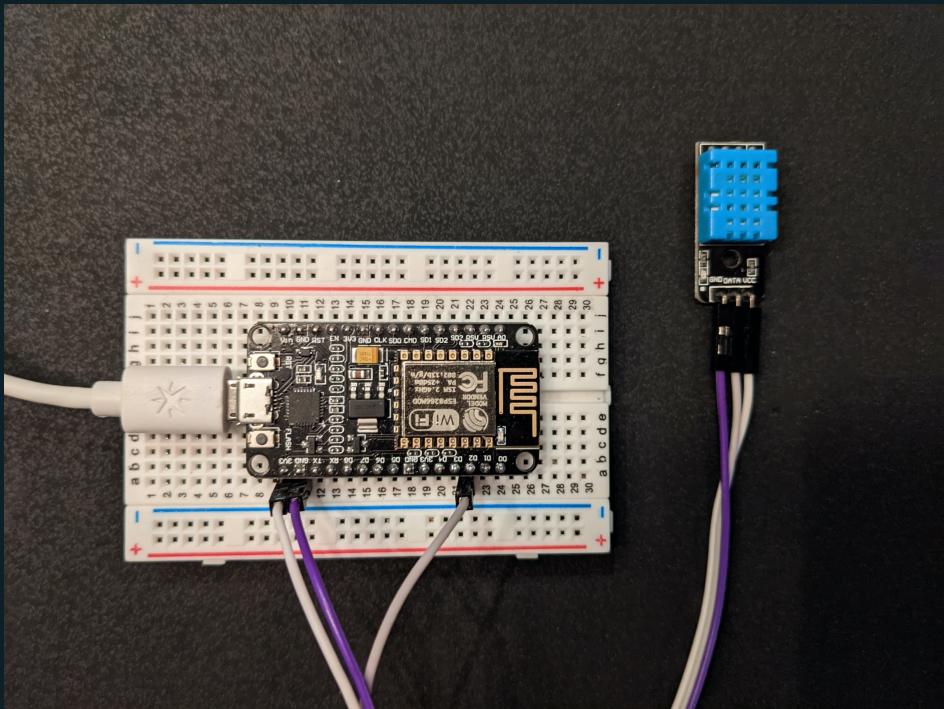
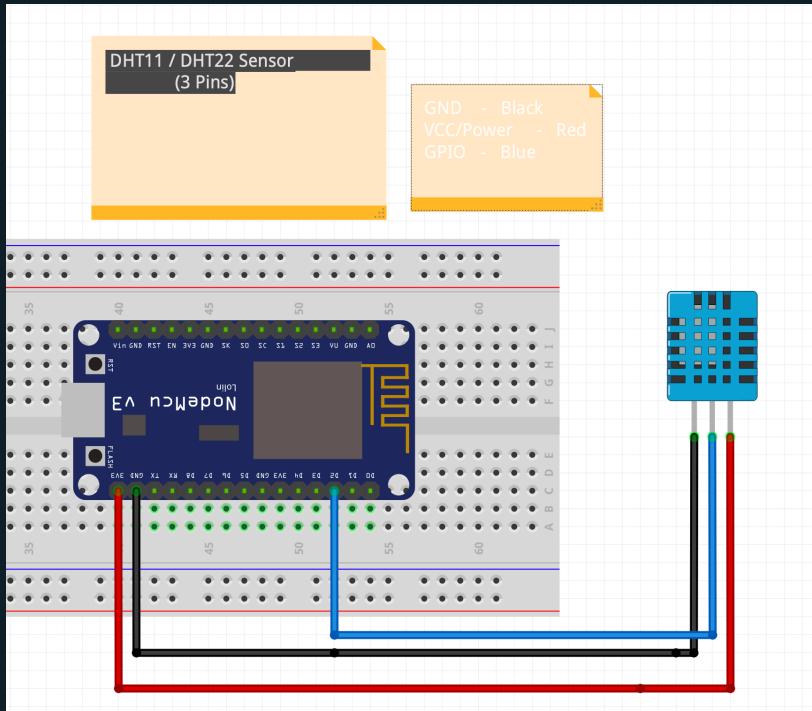
Note we will be using  
The GPIO numbers  
NOT the PINs

Like Pin D2 = GPIO4

We use number 4 in our code



# ESP8266 & DHT11 Wiring



## Code esp8266-dht11. - Number one

All code and instruction can be found here

<https://github.com/markusvankempen/esp8266andDHT11>

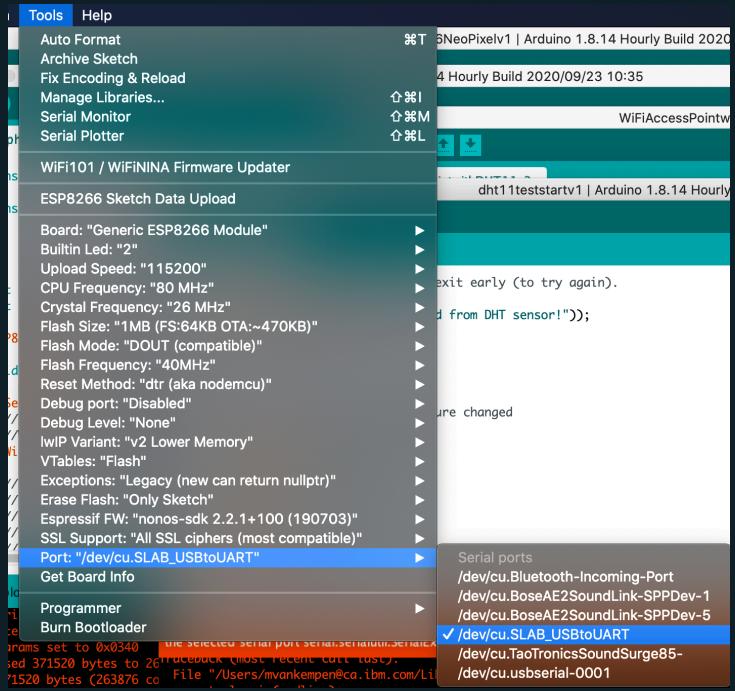
Here our 1st code to upload to the esp8266 which is connected to DH11

### **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

The screenshot shows the Arduino IDE interface on a Mac OS X desktop. The menu bar includes 'File', 'Tools' (selected), 'Help', and system status icons. The main window displays a list of sketches: '6NeoPixelv1 | Arduino 1.8.14 Hourly Build 2020/09/23 10:35', '4 Hourly Build 2020/09/23 10:35', 'WiFiAccessPointwithDHT11v2 | Arduino 1.8.14 Hourly Build 2020/09/23 10:35', and 'dht11teststartv1 | Arduino 1.8.14 Hourly Build 2020/09/23 10:35'. A context menu is open over the sketch 'dht11teststartv1', with the 'Boards Manager...' option highlighted. The 'Tools' menu on the left lists various board and port options. The 'Board' dropdown shows 'Generic ESP8266 Module' selected. The 'Boards Manager...' dropdown shows a list of available boards, with 'Generic ESP8266 Module' checked.

Tools Help

Auto Format ⌘T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... ⌘I

Serial Monitor ⌘M

Serial Plotter ⌘L

WiFi101 / WiFiNINA Firmware Updater

ESP8266 Sketch Data Upload

Board: "Generic ESP8266 Module"

- Builtin Led: "2"
- Upload Speed: "115200"
- CPU Frequency: "80 MHz"
- Crystal Frequency: "26 MHz"
- Flash Size: "1MB (FS:64KB OTA:~470KB)"
- Flash Mode: "DOUT (compatible)"
- Flash Frequency: "40MHz"
- Reset Method: "dtr (aka nodemcu)"
- Debug port: "Disabled"
- Debug Level: "None"
- lwIP Variant: "v2 Lower Memory"
- VTables: "Flash"

Boards Manager...

- Adafruit SAMD (32-bits ARM Cortex-M0+ and Cortex-M4) Boards
- Arduino AVR Boards
- Arduino megaAVR Boards
- Arduino nRF528x Boards (Mbed OS)
- Arduino SAMD (32-bits ARM Cortex-M0+) Boards
- ATtiny Microcontrollers
- Digistump AVR Boards
- ESP32 Arduino
- ESP8266 Boards (2.7.3) **Selected**
- Heltec ESP32 Arduino

✓ Generic ESP8266 Module

Generic ESP8285 Module

ESPDuino (ESP-13 Module)

Adafruit Feather HUZZAH ESP8266

Invent One

XinaBox CW01

ESPresso Lite 1.0



# 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

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Sketch, Tools, and Help. The title bar indicates the sketch is named "SNP-2020-ESP8266-DHT11-EnviromentMonitoring-v2" and is connected to port "/dev/cu.SLAB\_USBtoUART". The bottom status bar shows "Done uploading." The left sidebar displays the file structure: "Sketch" > "src" > "main.cpp". The main window has two panes: the Serial Monitor on the left and the Code Editor on the right.

**Serial Monitor (Left):**

```
14:37:56.640 -> readTemperature in C = 26.20
14:37:56.640 -> readHumidity in % = 36.00
14:37:56.640 -> readTemperature in C = 26.20
14:37:56.640 -> readHumidity in % = 37.00
14:37:56.640 -> readTemperature in C = 26.20
14:37:56.640 -> readHumidity in % = 37.00
14:37:56.640 -> readTemperature in C = 26.20
14:37:56.640 -> readHumidity in % = 36.00
14:38:01.276 -> readTemperature in C = 26.30
14:38:01.276 -> readHumidity in % = 36.00
14:38:01.276 -> readTemperature in C = 26.30
14:38:01.276 -> readHumidity in % = 37.00
14:38:01.276 -> readTemperature in C = 26.30
14:38:01.276 -> readHumidity in % = 37.00
14:38:02.892 -> readTemperature in C = 26.30
14:38:02.892 -> readHumidity in % = 37.00
14:38:04.883 -> readTemperature in C = 26.30
14:38:04.919 -> readHumidity in % = 37.00
14:38:06.939 -> readTemperature in C = 26.30
14:38:06.939 -> readHumidity in % = 38.00
14:38:08.954 -> readTemperature in C = 26.30
14:38:08.954 -> readHumidity in % = 38.00
14:38:10.984 -> readTemperature in C = 26.30
14:38:10.984 -> readHumidity in % = 38.00
```

**Code Editor (Right):**

```
#include "DHT.h"
// DHT sensor is connected to pin 4
#define DHTPIN 4
// DHT sensor type is DHT11
#define DHTTYPE DHT11
// Initialize the library with the pins and sensor type
DHT dht(DHTPIN, DHTTYPE);

void loop() {
    // Wait a few seconds between measurements.
    delay(2000);

    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    pt = t;
    t = dht.readTemperature();

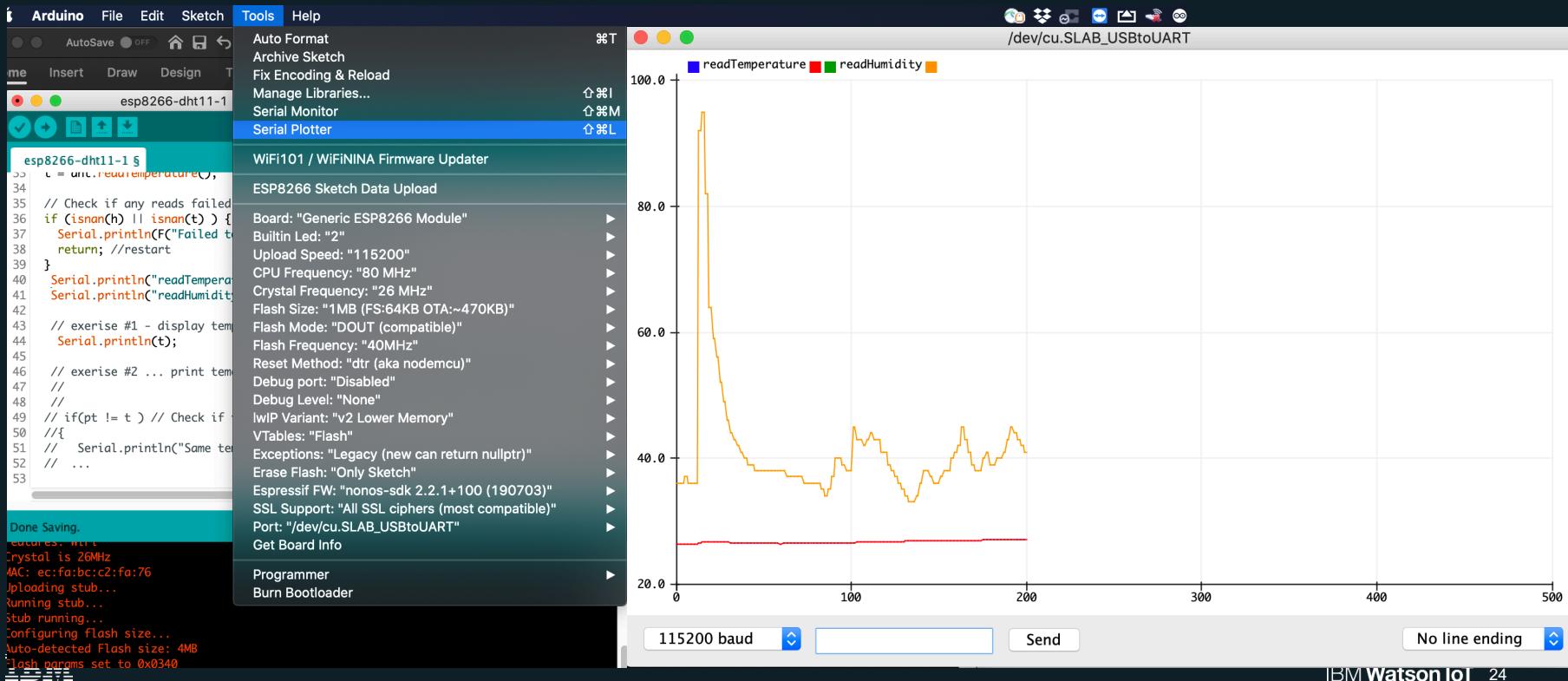
    // Check if any reads failed and exit early (to try again).
    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return; //restart
    }

    Serial.println("readTemperature in C = "+String(t));
    Serial.println("readHumidity in % = "+String(h));
}
```



# Exercise #1 ... display temperature and Humidity in the plotter

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



## Chapter two (2)

Sending data to mqtt broker

Save the Data to an SQL database

Display the data in Realtime and via Historian

