

Assignment 1

1) Stap 1: arduino code

Kopieert het code op github en plakt dat op Arduino IDE:

- ESP32-MQTT.ino

Daarna moet je nieuwe bijlage opmaken, klik op pijl naar beneden.



En voeg dan “nieuwe tabblad” en noem de nieuwe tabblad naar “setup.h”.

Daarna kopieert je het code op github en plakt dat op Arduino IDE in setup.h:

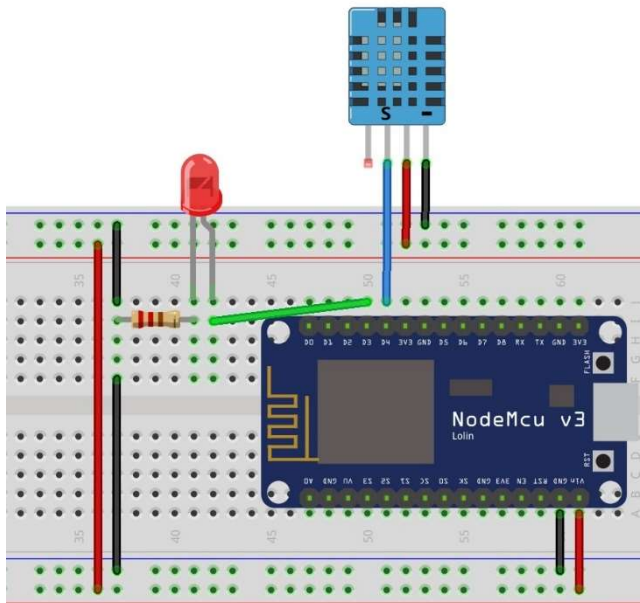
- Setup.h

2) Stap 2: esp32 opbouwen

Nu pak je materiaal:

- LED
- 2x Weerstand (100Ω)
- DHT11
- Esp32

Zie het voorbeeld van opbouw van esp32



Let op waar je led en DHT11 aansluit op esp32

3) Stap 3: Download je nodejs

Gaat naar website: <https://nodejs.org/en/> en download het LTS



4) Stap 4: NODE-RED activeren op websie

Open je opdrachtprompt "node.js command"

Type "NODE-RED"

```
node-red
Your environment has been set up for using Node.js 14.15.5 (x64) and npm.

C:\Users\Carels Cédric>NODE-RED
8 Mar 10:17:16 - [info]

Welcome to Node-RED
=====

8 Mar 10:17:16 - [info] Node-RED version: v1.2.9
8 Mar 10:17:16 - [info] Node.js version: v14.15.5
8 Mar 10:17:16 - [info] Windows_NT 10.0.18363 x64 LE
8 Mar 10:17:17 - [info] Loading palette nodes
8 Mar 10:17:18 - [info] Dashboard version 2.28.1 started at /ui
8 Mar 10:17:18 - [info] Settings file : C:\Users\Carels Cédric\.node-red\settings.js
8 Mar 10:17:18 - [info] Context store : 'default' [module=memory]
8 Mar 10:17:18 - [info] User directory : \Users\Carels Cédric\.node-red
8 Mar 10:17:18 - [warn] Projects disabled : editorTheme.projects.enabled=false
8 Mar 10:17:18 - [info] Flows file : \Users\Carels Cédric\.node-red\flows_LAPTOP-AJ8GJ9A7.json
8 Mar 10:17:19 - [info] Server now running at http://127.0.0.1:1880/
8 Mar 10:17:19 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
```

kopieert het "Server now running at" IP address => "http://172.0.0.1:1880/"

En plakt dat op website dan kom je in NODE-RED

Nu kan je beginnen programmeren,

- Voeg, indien je werkt met node-red, in een dashboard een gauge en een grafiek toe die de temperatuur en vochtigheid laat zien.
- Voeg een drukknop toe aan je hardware opstelling.
- Publish de staat van de knop (0 voor niet ingedrukt, 1 voor ingedrukt) naar het topic <U-NUMMER>/button.
- Zorg ervoor dat je de led kan sturen met de knop (via de MQTT broker, niet rechtstreeks binnen de ESP32 code).
- Werk met een andere student samen zodat de knop van student 1 de led van student 2 bedient.

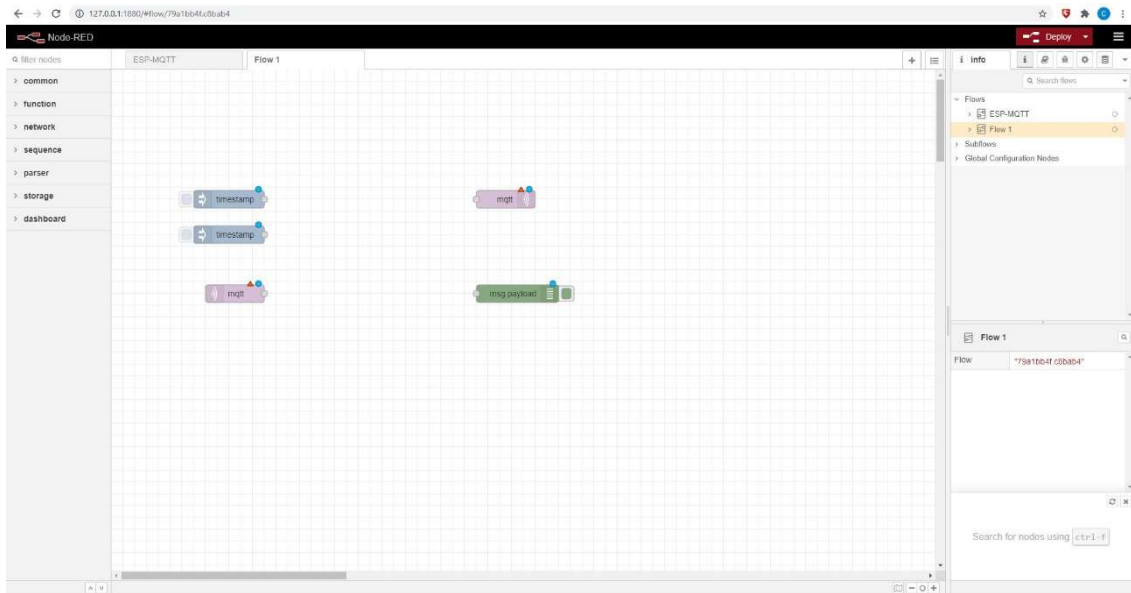
5) Stap 5: Flow node-red maken

Je voeg je functies blok toe in flow:

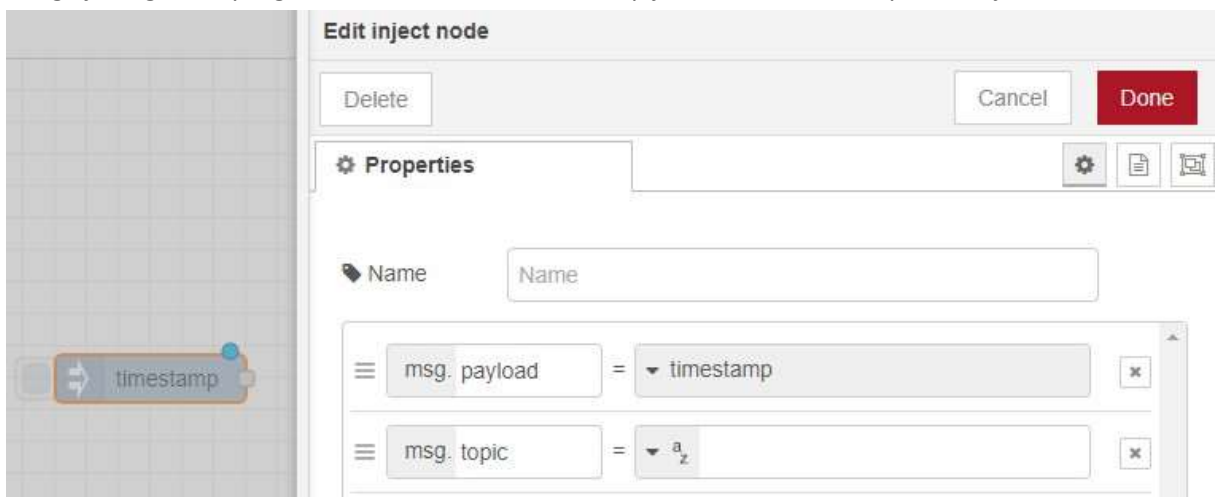
- “common” 2x inject
- “common” 1x debug
- “network” 1x mqtt in
- “network” 1x mqtt out



Dan krijg je dit:



Nu ga je beginnen programmeren, doe dubbel klik op je eerste “timestamp” van inject



En vul dat gegevens in:

En klik dan op “done”, nu doe je hetzelfde voor de 2^{de}.

The screenshot shows the MQTT configuration interface. At the top, the 'Name' field is set to 'LED on'. Below it, there are two rows of configuration. The first row is for 'msg. payload', which is set to '1'. The second row is for 'msg. topic', which is set to 'a_z'. Each row has a small 'x' icon to its right.

Maar voor LED off, zie voorbeeld:

The screenshot shows the MQTT configuration interface. At the top, the 'Name' field is set to 'LED off'. Below it, there are two rows of configuration. The first row is for 'msg. payload', which is set to '0'. The second row is for 'msg. topic', which is set to 'a_z'. Each row has a small 'x' icon to its right.

Nu ga je dat doen voor “network” met mqtt out en mqtt in.

Je moet de juiste dezelfde naam hebben als in arduino IDE

```
const char* mqtt_server = "54.75.32.245";  
const int mqtt_port = 1883;  
#define MQTT_SERIAL_PUBLISH_CH "r0804148/measurement"  
#define MQTT_SERIAL_RECEIVER_CH "r0804148/led"  
const String DEVICE_ID = "r0804148";
```

Dus je moet “publisch en Receiver” communiceren met led en DHT11.

Voor MQTT IN vul je deze gegevens in:

The screenshot shows the MQTT IN configuration interface. The 'Server' field is set to 'CC.C'. The 'Topic' field is set to 'r0804148/led'. The 'QoS' field is set to '0'. The 'Retain' checkbox is checked. The 'Name' field is set to 'Name'.

Voor MQTT OUT vul je deze gegevens in:

The screenshot shows the MQTT OUT configuration interface. The 'Server' field is set to 'CC.C'. The 'Topic' field is set to 'r0804148/measurement'. The 'QoS' field is set to '2'. The 'Output' field is set to 'auto-detect (string or buffer)'. The 'Name' field is set to 'Name'.

Nu ga je dat doen voor debug van comment doen en vul dat gegevens in:

Output ▼ msg. payload

To ☒ debug window

☐ system console

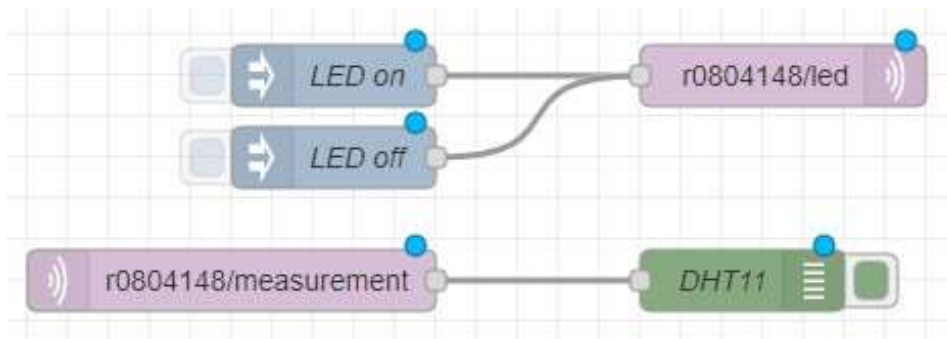
☐ node status (32 characters)

Name DHT11

Nu moet je verbinding maken:

- Led on en Led off met r0804148/led
- r0804148/measurement met DHT11

dan krijg je dit:



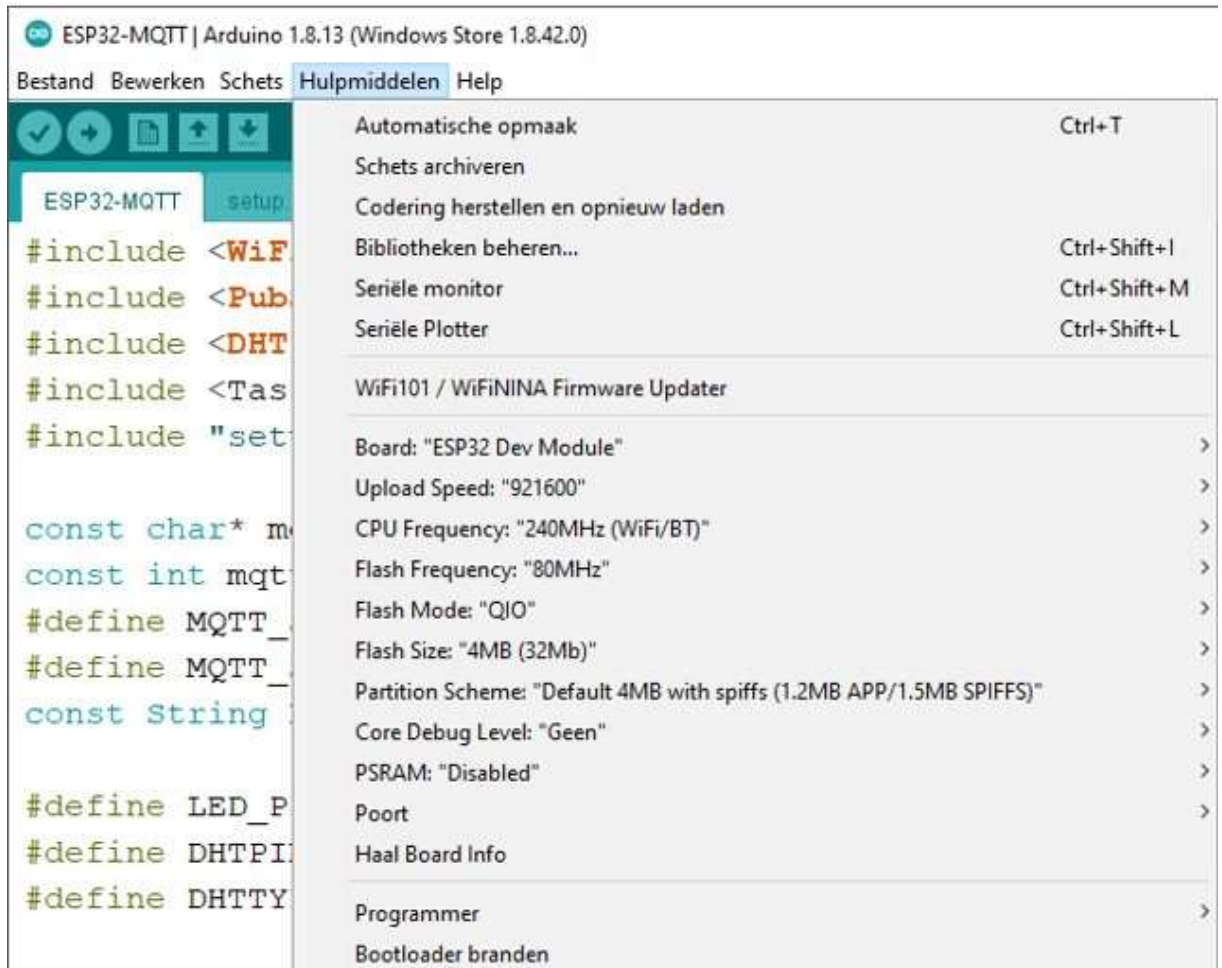
Klik dan op “deploy” en ok dan ben je klaar met communiceren met esp32



6) Stap 6: Arduino IDE goed instellen en uploaden

Verbind nu je esp32 met je laptop of pc, nu ga je esp32 goed instellen zodat esp32 met wifi kan communiceren.

Zorgt ervoor dat alles na WIFI101/ WIFININA FirmWare Updater zo is ingesteld



Klik op "uploaden" voor esp32



Dan zie je dat bezig is, na gedaan met uploaden krijg je dit:

```
De schets gebruikt 706549 bytes (53%) programma-opslagruimte. Maximum is 1310720 bytes.  
Globale variabelen gebruiken 38484 bytes (11%) van het dynamisch geheugen. Resteren 289196 bytes voor lokale variabelen. Maximum is 327680 bytes.  
esptool.py v3.0-dev  
Serial port COM5  
Connecting.....  
Chip is ESP32-D0WDQ6 (revision 1)
```

Dan zie je Connecting.....

Nu moet je op esp32 op “boot” blijven drukken totdat er geen niet meer weergeeft, dan zal die beginnen laden met connectie te maken met je Node-red.

Als dat gedaan is krijg je dit:

```
Writing at 0x00070000... (100 %)  
Wrote 706672 bytes (406786 compressed) at 0x00010000 in 6.2 seconds (effective 907.7 kbit/s)...  
Hash of data verified.  
Compressed 3072 bytes to 128...  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1536.0 kbit/s)...  
Hash of data verified.  
  
Leaving...  
Hard resetting via RTS pin...
```

Nu kan je testen op je Node-red met on en off op te drukken.