Arduino Explore IoT



[Xavier Conesa Expósito](mailto:xconesae@ies-sabadell.cat)

[Kevin Ruiz Varo](mailto:kruizv@ies-sabadell.cat)

Mario Martín Hernández

Índex

[**Participacions i rols 2**](#_q1ffahlnmqx2)

[**Node-RED 3**](#_n4uxkqwu3csg)

[Estructura 1 3](#_jmeh4wwztjeg)

[Estructura 2 5](#_s0w6dm18l0x)

[**Codi Arduino 6**](#_dn4bjl3fmw6s)

[**MQTT 13**](#_2zs61935hmcn)

[**ESP32 (Extra) 16**](#_psurqjgr9rq)

# Participacions i rols

Nosaltres, un grup que ha treballat amb uns dels kits més complexos, distintius i menys coneguts que els altres.

Treballant amb Arduino MKR IOT CARRIER que implementa sensors de temperatura, humitat i tàctils, amb una pantalla OLED, la capacitat de cinc llums LED RGB, amb una entrada per a targetes SD, tres entrades per connectar diferents sensors externs i un munt de coses més que podem accedir a treballar.

A més a més, el kit incorpora una placa Arduino MKR WIFI 1010 per poder treballar amb l’anterior component que s’ha explicat, cable USB que es connecta via Serial port, cables per connectar els sensors externs, pila, sensasor Soil Moisture i sensor d’infraroig.

Hem distribuït la feina en diferents parts i processos.

En començar, vam flashejar el SD amb l'ISO de Raspberry PI amb el programa oficial que podeu trobar [aquí](https://www.raspberrypi.com/software/), amb aquest software flasheja l'ISO i deixa el sistema operatiu de la Raspberry (Raspbian) instal·lat i llest per configurar-lo.

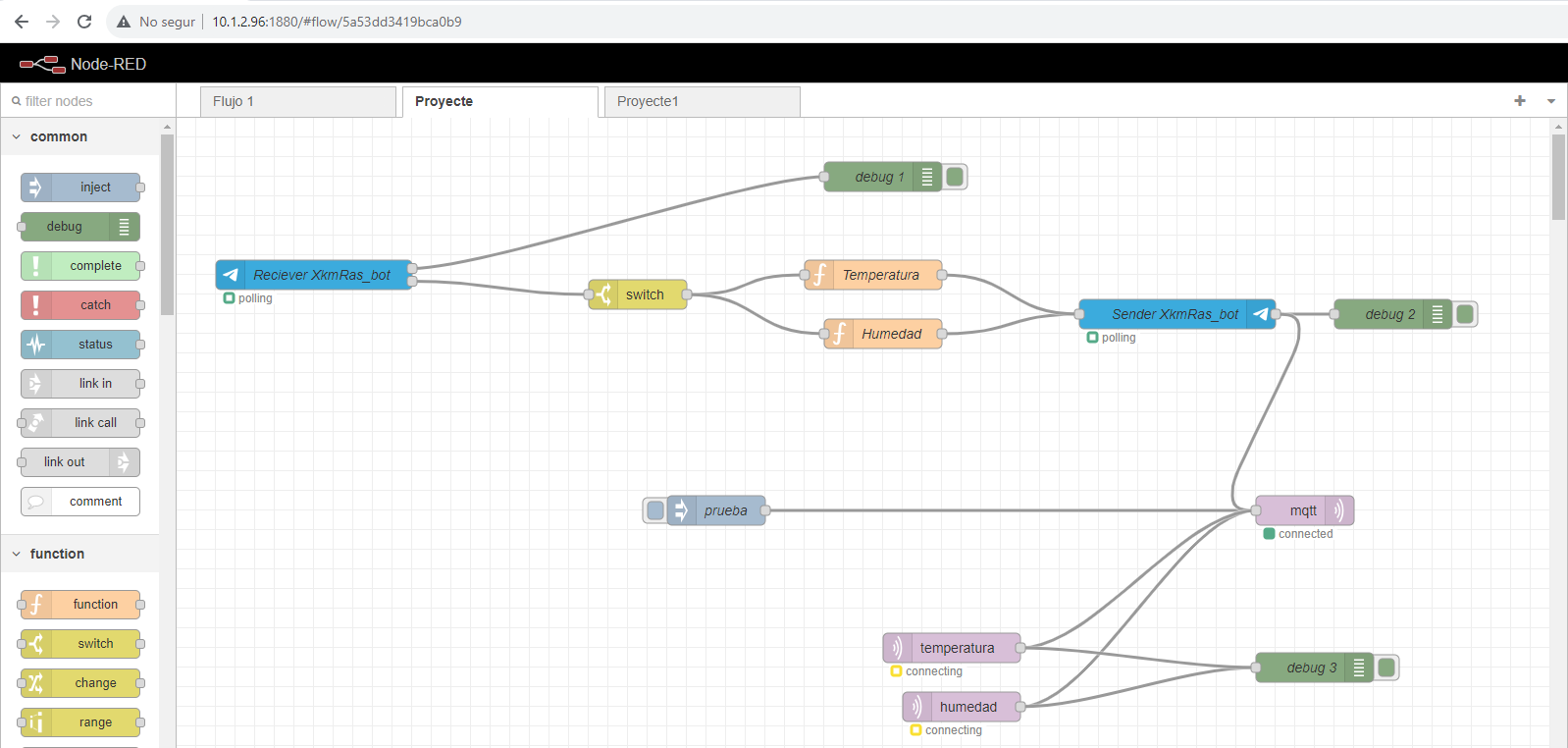
Un exemple de com és el programari:



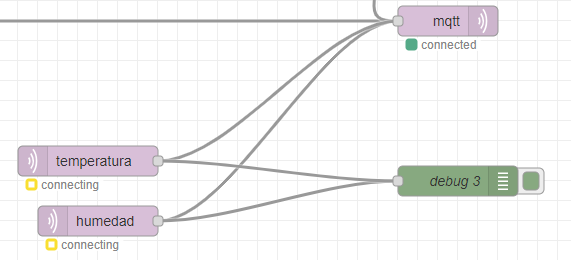
# Node-RED

### Estructura 1

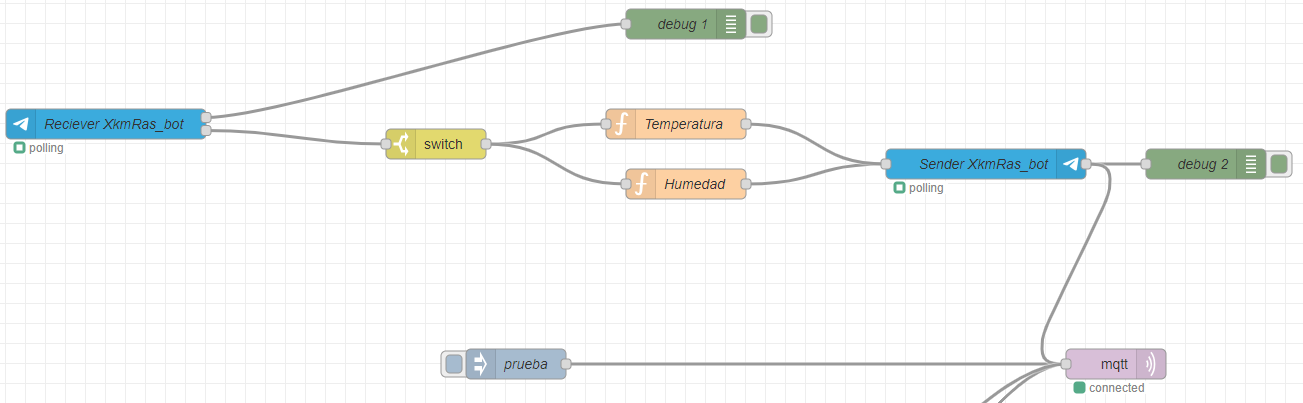
La nostra primera estructura en el Node consisteix per distribuir tota la informació.



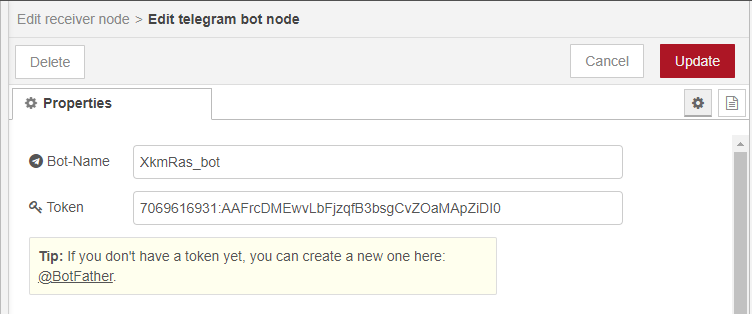
Tenim una part que llegeix la temperatura i humitat ambiental de l'Arduino MKR IOT CARRIER, quan ho fa ho mostra per pantalla del Node i ho transfereix a MQTT.



La següent part de la distribució és del Bot de Telegram, consisteix que quan està connectat surt pel xat del Node, després, en el chat de Telegram quan escrivim la paraula “Temperatura” o “Humedad” fa que ho mostra pel Node i també ho llança a MQTT.



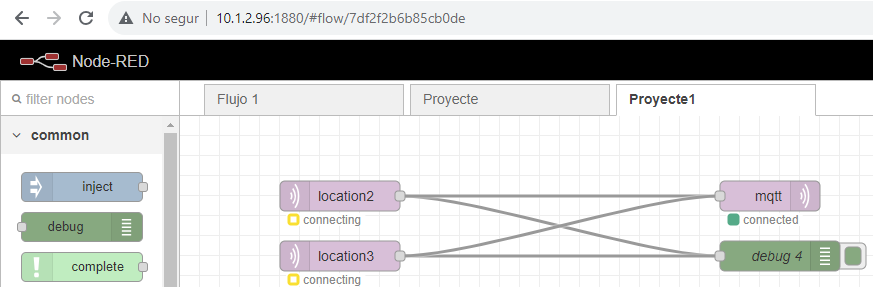
En el nostre cas no funciona actualment, però, sí que funcionava fa uns dies. Aquí està la prova de què està tot correcte, però d’un dia a l’altre ha deixat de servir.



### Estructura 2

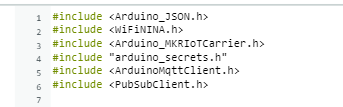
La nostra segona estructura s’utilitza per saber la temperatura i humitat de diferents llocs al voltant del món, en concret hem agafat la ciutat de Madrid i Copenhagen.

Consta en la rebuda d’informació de l’Arduino i ho transmet cap a MQTT i en el Node.

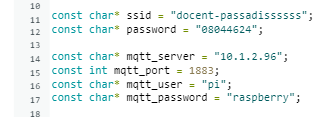


# Codi Arduino

Hem utilitzat les biblioteques següents:



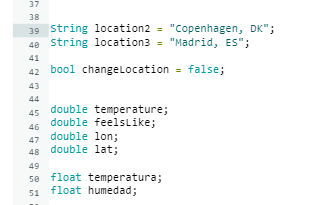
En aquesta part del codi primer li diem a l'Arduino que es connecti a la WiFi de docent-passadissssss dient-li el SSID de la xarxa amb la seva contrasenya. Seguidament en diem la IP la qual s’ha de connectar per al MQTT Server i per quin port a més de l’usuari i contrasenya de la Raspberry.



Aquestes dues línies diuen els tòpics d'on sortirà l’info a través del MQTT. En el nostre cas el “Proyecte” (mal escrit) mostrarà la temperatura i humitat que els sensors del IoT detecten i el tòpic “Proyecte1” mostrarà el temps de Madrid i Copenhaguen.



Aquí s’afegeix la informació del temps, és a dir, mostrarà la informació ordenada dient que és amb el valor corresponent (exemple: Madrid; temperature: 19; feels like: 22…).



Aquesta part del codi el que fa és mostrar per la pantalla del IoT informació de si s’ha pogut connectar o no. En cas de que estigui connectant-se dira: *Conectando al servidor MQTT.*

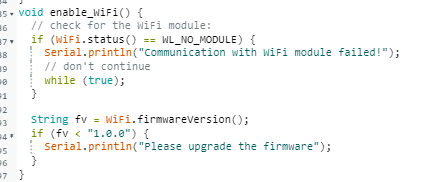
I si s’ha pogut connectar dirà: *Conectado al servidor MQTT.*

En cas de que no s’hagi pogut connectar o hagi hagut un error dirà: *Error al conectar al servidor MQTT. Intentando de nuevo en 5 segundos.*

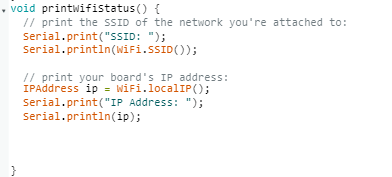


Aquí comprova si el mòdul de wifi funciona correctament, en cas que no detecti el mòdul o falli mostrara l’error “Communication with WiFi module failed!

A més si el firmware no està actualitzat llença el missatge “Please upgrade the firmware”



Això printeja el nom de la xarxa wifi a més de la IP de l’Arduino en la terminal serial i a la pantalla del IoT.

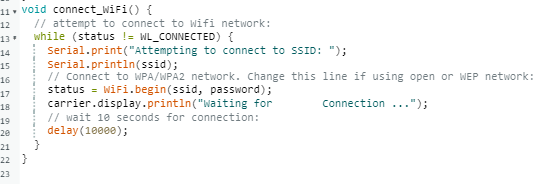


Mentre que l’Arduino es connecta a la xarxa WiFi, ens mostrarà l’estat de:

Quan estigui intentant connectar: *Attempting to connect to SSID:*

Quan ha introduït la contrasenya de la xarxa WiFi: *Waiting for Connection…*

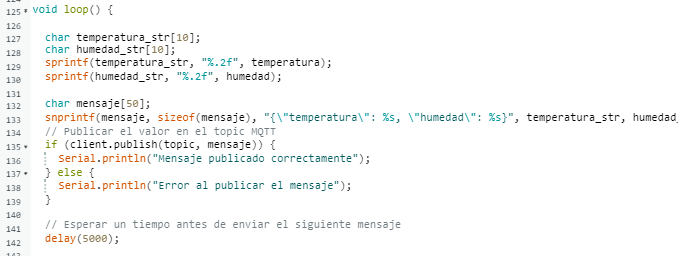
Té a més un temps de 10 segons de delay per connectar-se.



Això és un bucle el qual quan s’envia les dades cap al MQTT i a més ens informa si s’ha enviat dient-nos els següents missatges a través del terminal serial.

Si s’ha enviat dirà: *Mensaje publicado correctamente*

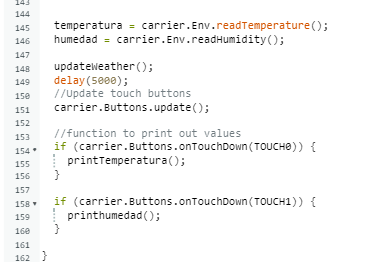
En cas que n’hagi hagut un error: *Error al publicar el mensaje.*



Primer llegeix els valors de la temperatura i humitat ambientals del sensor i les envia a les seves respectives variables (temperatura i humedad).

Executem la funció de updateWeather (es veurà més endavant).

I posem que si es prem el sensor tàctil número 0, cridarà a la funció “printTemperatura” o si es toca el sensor tàctil número 1 es cridarà a la funció “printhumedad”.



Aquí tenim la funció “printTemperatura”, mencionada abans, la qual agafa la variable “temperatura”, amb el valor llegit anteriorment, i fa una seqüència d’ifs, si la temperatura és superior a 80 (difícil, ho hem posat com a exemple, per això ens hem saltat des dels 30 als 80) la pantalla del dispositiu es posa vermella i posarà en lletres blanques a la pantalla (com a exemple): Temp: 89C

Si es entre 21 i 30, la pantalla es posarà verd i amb el text blanc posarà com a exemple: Temp: 25C

Si es entre 5 i 20, la pantalla es posarà blava i amb el text blanc posarà com a exemple: Temp: 8C



Així és com es mostra per la pantalla de la placa Arduino MKR IOT CARRIER:



Aquí tenim la funció “printhumedad”, mencionada abans, la qual agafa la variable “humedad”, amb el valor llegit anteriorment, i fa una seqüència d’ifs, si la humitat és superior a 80 la pantalla del dispositiu es posa vermella i sortirà com a exemple: Humi: 90%

En el cas de ser entre 61 i 80 sortirà amb la pantalla groga i com a exemple: Humi 70%

En el cas de ser entre 30 i 60 la pantalla es torna blava i com a exemple: Humi 46%



També un exemple de com es mostra amb la humitat:



Aquí tenim la funció “updateWeather” sencera:

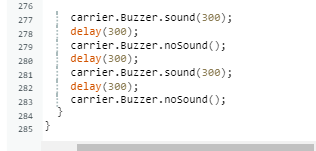


Amb el que llegeix al servidor de “api.openweathermap.org” crea l'estructura (els noms dels llocs, quin dia fa, sensació tèrmica, temperatura, longitud, etc…) amb la qual més tard es podrà llegir al serial monitor o al MQTT.

A més ens avisa si el missatge s’ha enviat correctament o no.

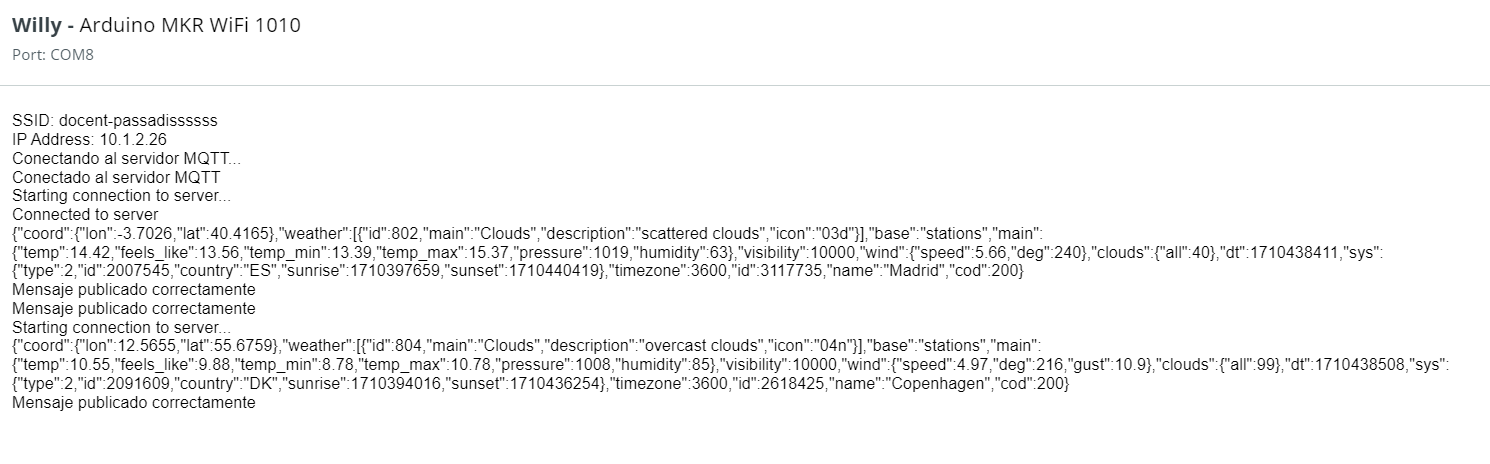


Ara en quant enviem la informació al MQTT i serial monitor, el dispositiu farà un soroll assegurant que s’ha realitzat l’acció.



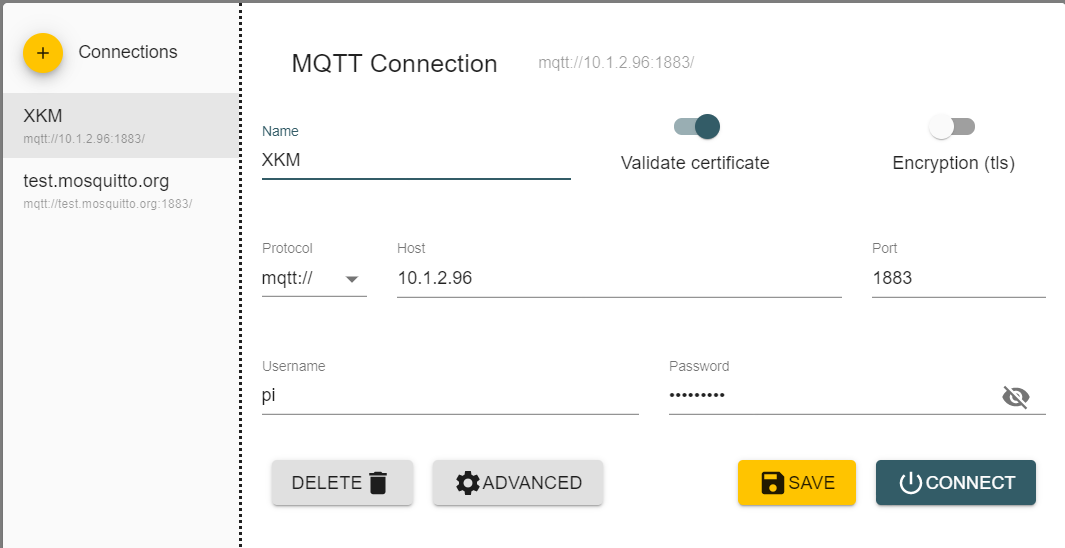
Un exemple de com s’escolta [aquí](https://drive.google.com/file/d/10cc1f4M0dV6ViIaIpLV6I-beknIc1DPs/view?usp=sharing).

En connectar l’Arduino IoT, engegar-lo i iniciar el codi, a dins del serial monitor ens informa sobre la xarxa WiFi que s’ha connectat, la seva IP a més de l’estat de connexió de l’Arduino al servidor MQTT. Una vegada s’hagi connectat també mostra les dades de temperatura dels sensors del IoT a més del temps actual de Copenhaguen i Madrid, i si els missatges s’han enviat correctament o no.

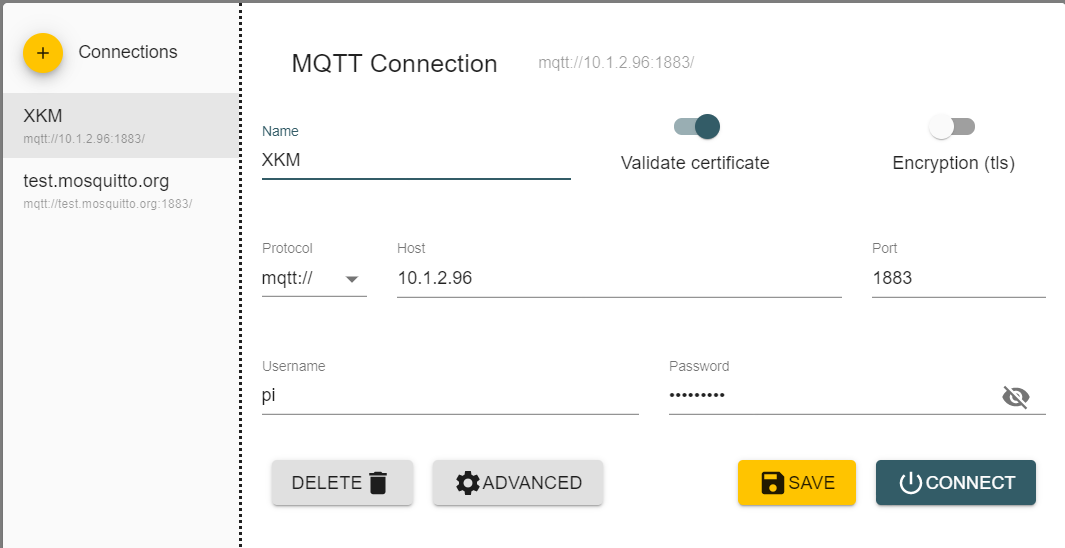


# MQTT

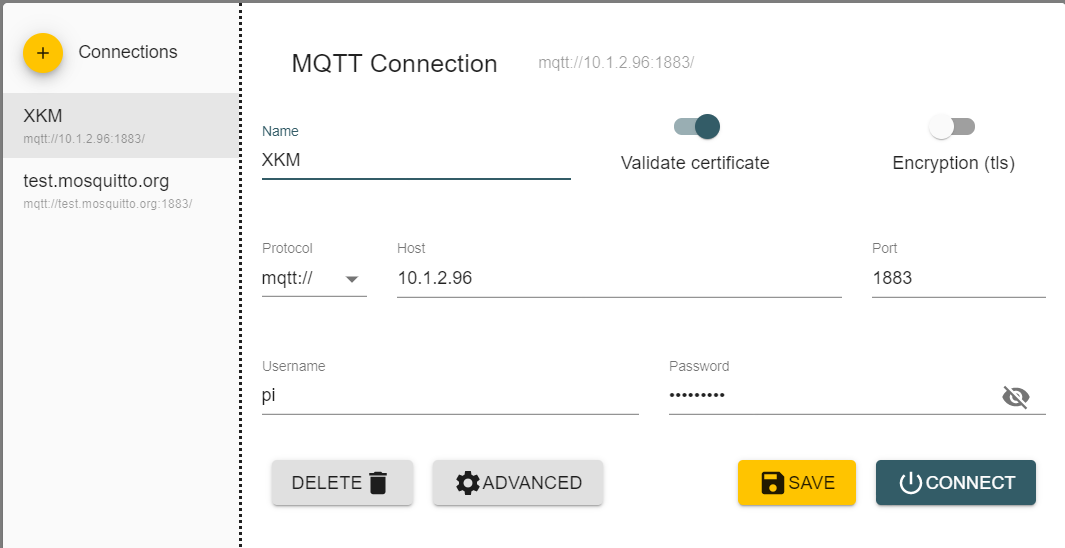
Per poder connectar i tenir accés al MQTT primer de tot l’hem donat un nom a la connexió, que són les nostres inicials.



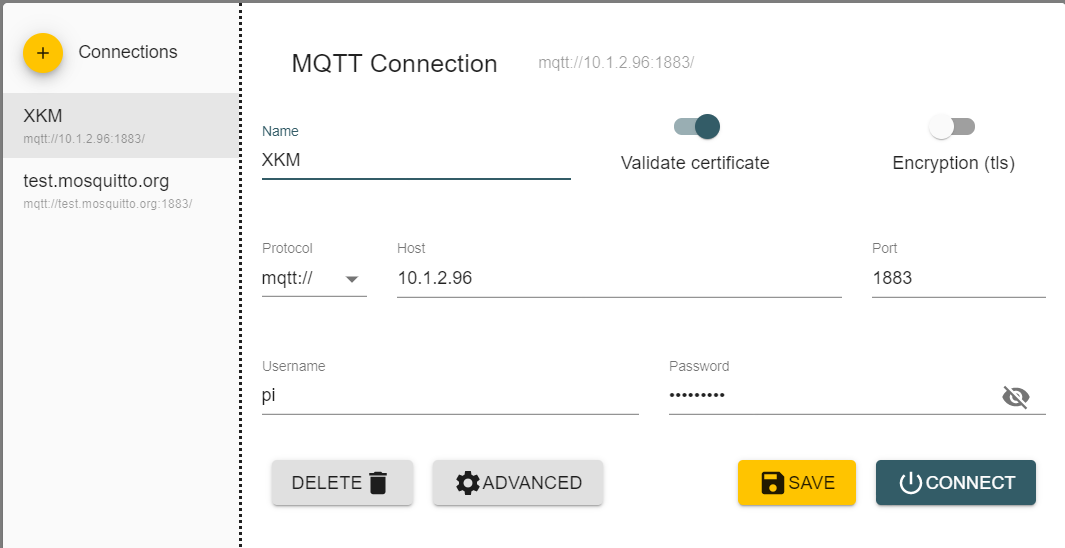
Seguidament pel protocol que és el mqtt://



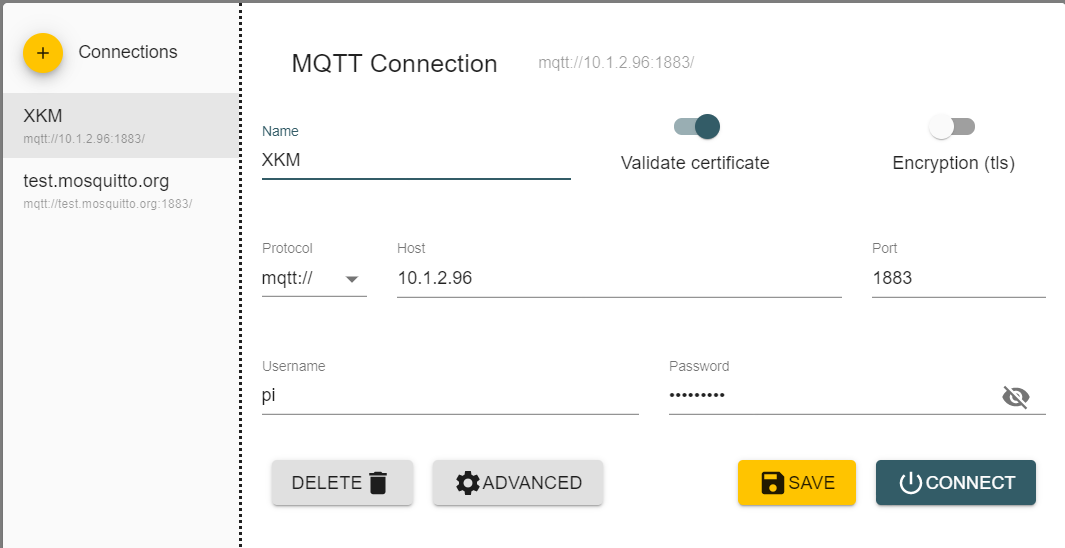
Després en host hem posat la IP de la Raspberry Pi i el port, el qual és el 1883.



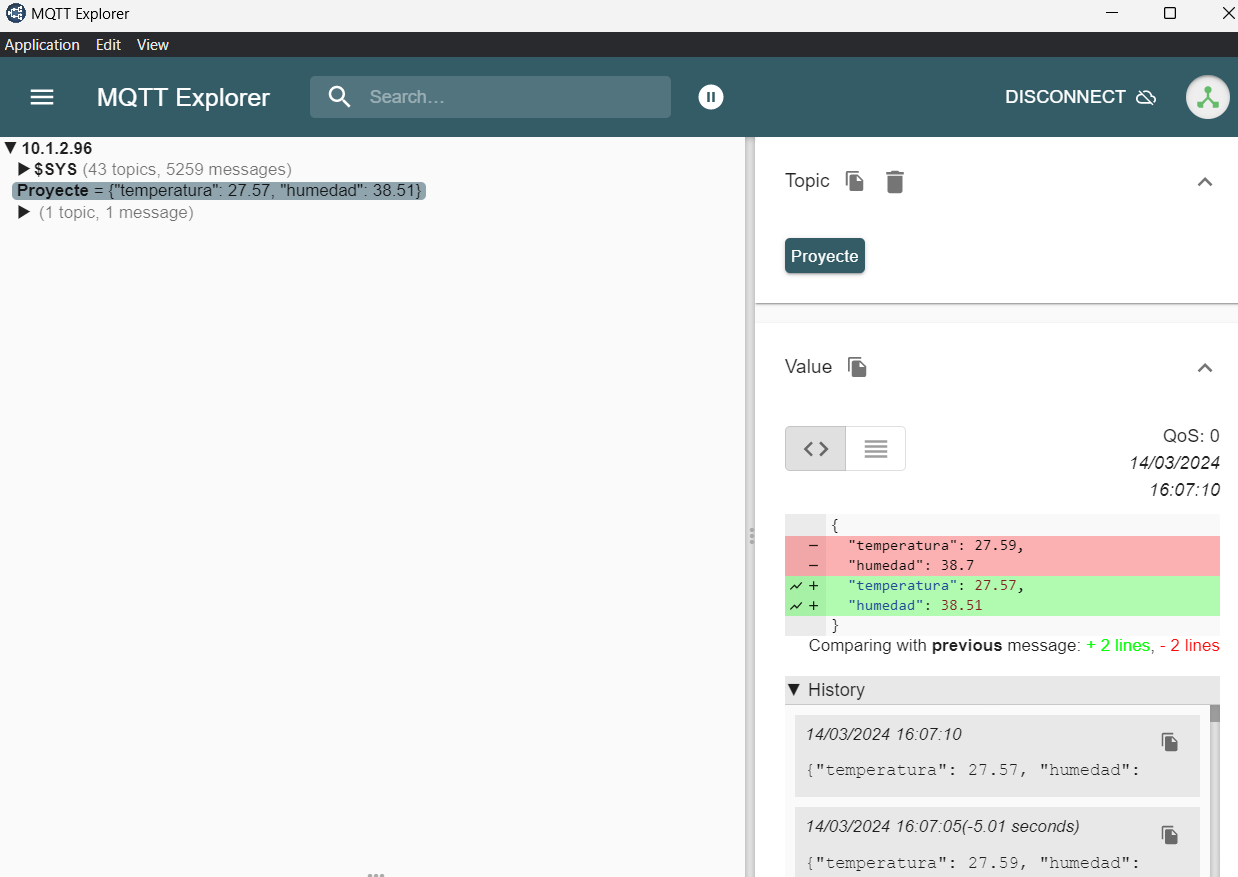
Finalment, hem de posar l’usuari que tinguem en la Raspberry i la seva contrasenya.



Una vegada omplertes les dades necessàries, cliquem el botó verd “Connect”, però nosaltres abans d’això hem guardat la connexió per tenir-la guardada i no haver d'escriure totes les credencials cada vegada que volem connectar-nos. Ens ha quedat així:

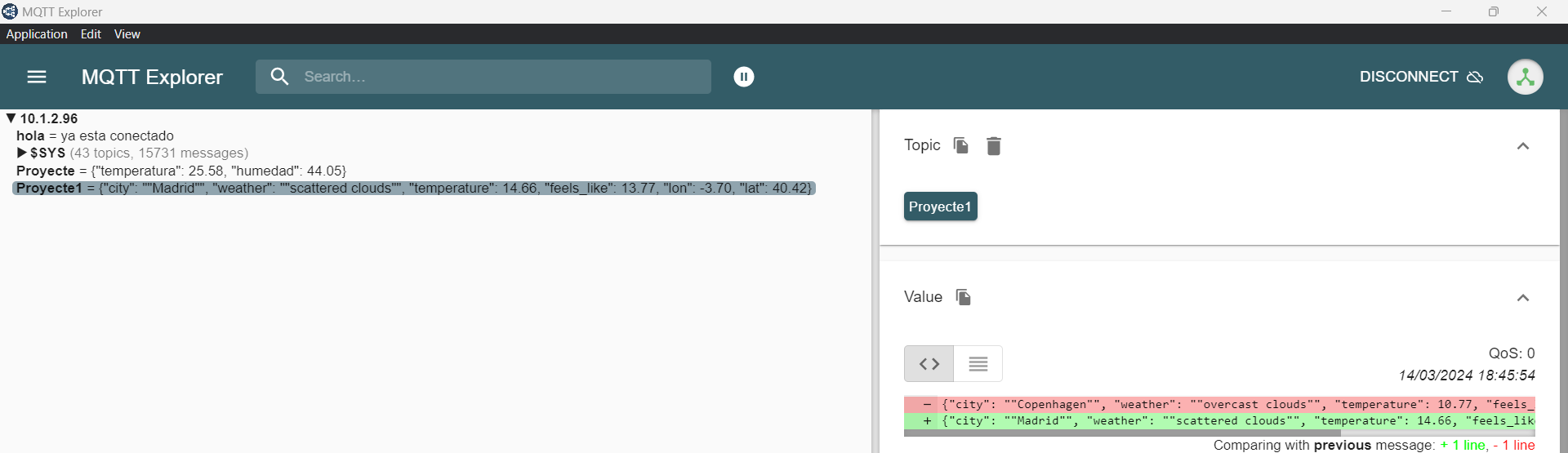


Una vegada a dins podem veure uns quants apartats amb subapartats què venen per defecte. El que hem afegit és el nostre que es diu “Proyecte”(mal escrit) on mostra les dades de temperatura i humitat les quals es van actualitzant cada 5 segons que l’Arduino IoT agafa i les envia.

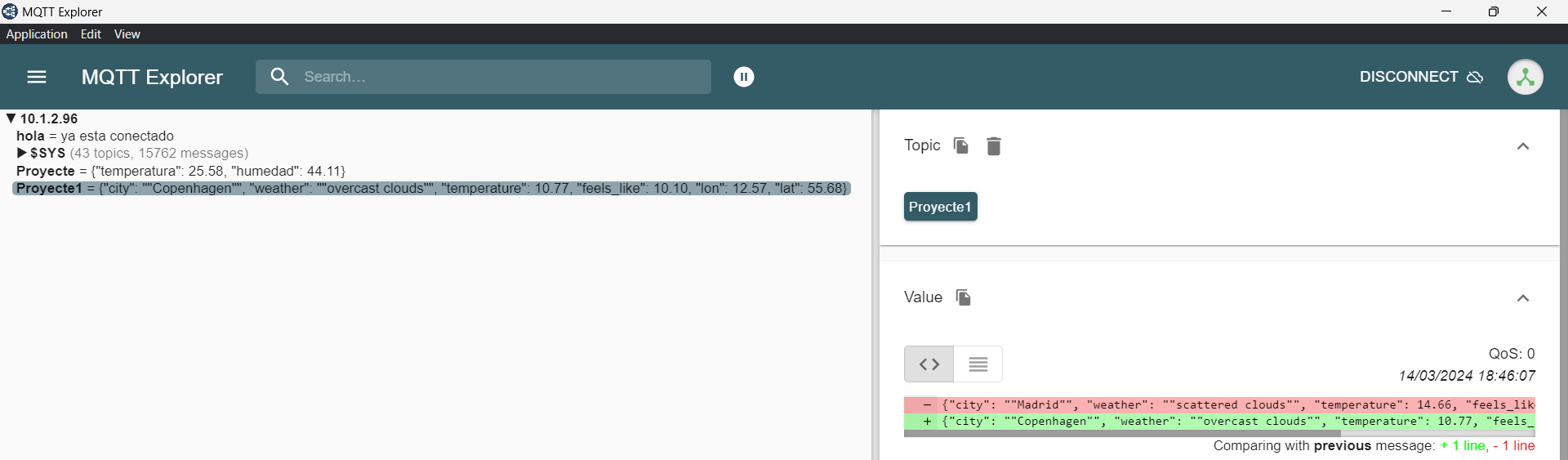


Aquí podem veure que s’ha rebut la informació del climatica a Madrid amb el format que hem posat abans.

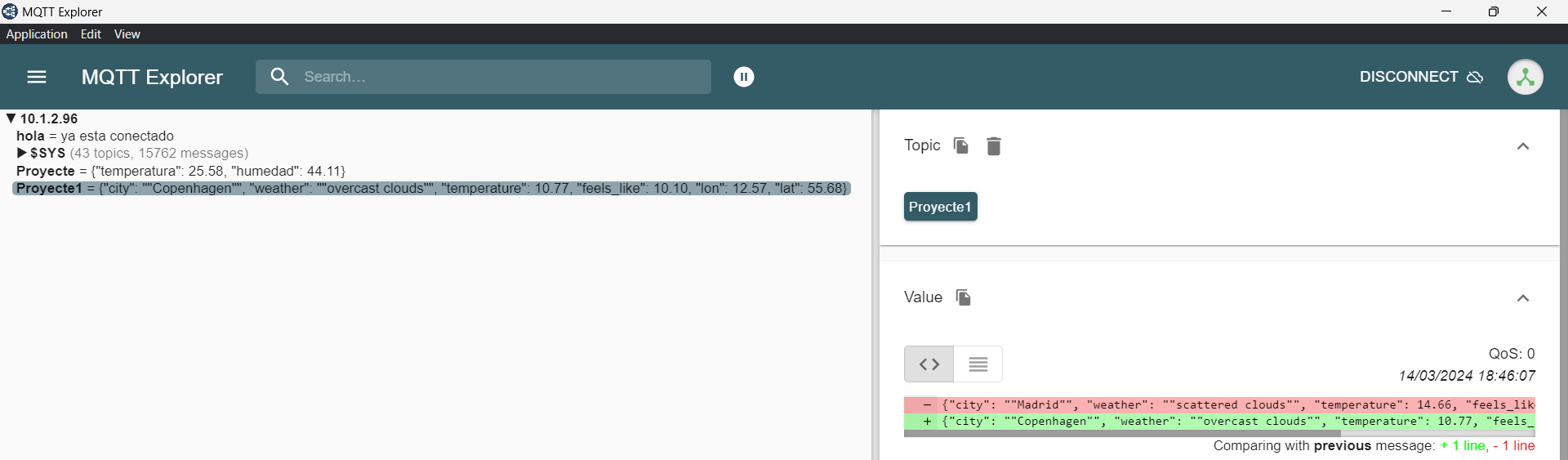
Tenim el nom, clima, temperatura, sensació termica, longitud i latitud.



Ara podem veure la de Copenhaguen, es van intercanviant cada una estona.



Aquí es pot veure com el de Madrid ha sigut substituït pel de Copenhaguen, encara que d'aquí a una estona es tornarà a substituir pel de Madrid.



# ESP32 (Extra)

Vam arribar a connectar el ESP32 i que tingués comunicació amb el MQTT

A continuació, posaré el codi que vaig utilitzar per fer-ho.

Aquest codi de manera resumida, el que fa és que es connecta al servidor MQTT a el topic “Proyecte”, amb el nom d’usuari “pi” i la contrasenya “raspberry” amb el port 1883.

Una vegada connectat se subscriurà al tòpic “Proyecte” i enviara el missatge: “Hi, I’m ESP32 ^^”

