

Activitat 1.3: Internet de les coses. Node-Red.Telegram

* Wiki d'en Jordi Binefa: https://wiki.binefa.cat/index.php?title=SMX_2018-2019

* [Grups SMX2B](#)

Apartat 1 - Instal·lació de Node-Red

1- Instal·la **NodeJS** sobre el servidor Debian. Com a **root** executa:

```
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -  
apt-get install -y nodejs
```

2- Instal·la altres programes que utilitza NodeJS:

```
apt-get install -y build-essential libssl-dev
```

3- Instal·la **Node-Red** sobre el servidor Debian. Com a **root** executa:

```
npm install -g --unsafe-perm node-red
```

4- Descarrega i instal·la el fitxer **node-red.service** per poder controlar **node-red** amb **systemctl**. Executa com a **root**:

```
cd /etc/systemd/system  
wget http://www.collados.org/smx2/m04/uf4/node-red.service  
systemctl daemon-reload
```

5- Comprova l'adreça IP del teu servidor Debian. Com a usuari **smx2** executa: **ip a**

6- Posa en marxa **Node-Red** i habilita'l perquè és posi en marxa durant l'arrancada de l'ordinador. Com usuari **root** executa:

```
systemctl start node-red  
systemctl enable node-red
```

7- Comprova executant com a **root** l'ordre **netstat -atupn | grep node-red** el **port** pel qual treballa **Node-Red**.

8- Des de la teva màquina física fes una connexió via navegador per l'adreça IP trobada a l'apartat 7 i el port trobat a l'apartat 9.

9- Documentació:

<https://linuxconfig.org/how-to-install-nodejs-on-debian-9-stretch-linux>

<https://nodered.org/docs/getting-started/installation>

Apartat 2 - Un primer projecte amb Node-Red

1- Amb la informació del primer enllaç de la documentació, crea un projecte de nom Projecte1 que injecti un missatge cada 2 segons.

2- Amb la informació del 2n enllaç de la documentació, deshabilita Projecte1

3- Amb la informació del 3r enllaç de la documentació, esborra Projecte1

4- Documentació:

<http://developer.opto22.com/nodered/general/getting-started/node-red-hello-world/>

<https://nodered.org/docs/user-guide/editor/workspace/flows>

Aparta 3 - Important un projecte

1- Descarrega des del teu equip físic descarrega el fitxer <http://www.collados.org/smx2/m04/uf4/webserv.txt>.

2- Copia el codi que hi ha dins del fitxer **webserv.txt**.

3- Des del **Node-Red** ves a la icona de menú de la dreta i escull **Import** --> **Clipboard**. Enganxa el codi i fes clic a **Import**. Enganxa el codi gràfic **Node-Red**.

4- Fes un **Deploy**

5- Des del teu equip físic connecta't a http://ip_servidor_debian:1880/smx i comprova que tens una web disponible i accessible.

Apartat 4 - Instal·lació del mòdul Dashboard per Node-Red

1- Com usuari **smx2** entra dins del directori **.node-red**. Executa:
`cd .node-red`

2- Dins del directori **.node-red** i com usuari **smx2** executa la següent ordre per descarregar el mòdul **Dashboard** de **Node-Red**:
`npm i node-red-dashboard`

3- Ara entra dins del directori a on s'ha descarregat el mòdul Dashboard i instal·la el mòdul descarregat. Executa com **smx2**:

```
cd node-modules
cd node-red-dashboardrd
npm install
```

4- Com a **root**, reinicia **Node-Red** perquè els canvis tinguin efecte executant:
`systemctl restart node-red`

5- Per comprovar que s'ha instal·lat correctament, accedeix amb el navegador a http://adreça_ip:1880/ui. Comprova que mostra la pàgina de benvinguda a **Node-RED Dashboard**. Comprova també que ha aparegut també a la part de dalt de la dreta la icona següent per accedir al dashboard:



i també que pots accedir amb el navegador al **Dashboard** fent clic a la icona:



6- Descarrega des de la teva màquina física (segurament Windows) el fitxer [projecte2.json](#). Des del **Node-Red** ves a la icona de menú de la dreta i escull **Import** --> **Clipboard**. Després escull **select a file to import** i selecciona el fitxer **projecte2.json**. Després fes clic a **Import**. Ara selecciona la nova pestanya **Projecte2_Dashboard** i fes un **Deploy**. A continuació fes clic a sobre de la icona:



i accedeix al **Dashboard** del projecte.

Apartat 5 - Treballant amb packets UDP, Node-RED i mòbils Android (Part 1)

1- Instal·la:

- a) L'App **UDP Sender** de **hastarin** per **Android**.
- b) L'App **UDP Reciver and Sender** de **Wezzl Studios** per **Android**.

2- Crea un nou Flow de nom **udp_mòbil**.

3- Afegeix un node de tipus **input udp** que:

- a) Escolti missatges UDP
- b) Treballi amb el port UDP 33333 versió ipv4
- c) El missatge UDP rebut ho enviarà a la seva sortida en forma de cadena de caràcters (String)
- d) Nom del node: missatge enviat des del mòbil

4- Afegeix un node tipus **output debug** que:

- a) Doni com a sortida el msg.payload
- b) El nom sigui mostra missatge enviat pel mòbil

5- Connecta els nodes. Fes un **Deploy** (desplegament) del programa i després selecciona l'eina "**Debug messages**". Esborra qualsevol missatge anterior.

6- Executa **UDP Sender**. Escriu l'adreça IP del teu Linux amb el Node-RED. Indica el port 3333. Escriu a Text el missatge "Hola Node-RED del del meu mòbil". Fes clic a Send. Comprova que surt per la finestra "Debug messages".

7- Afegeix un node de tipus **output udp** que:

- a) Envii missatges UDP
- b) Treballi amb el port UDP 22222
- c) Envii el missatge a l'adreça IP del teu mòbil (versió ipv4).
- d) Nom del node: missatge enviat al mòbil per udp

8- Afegeix un node tipus **input inject** que:

- a) Payload: Hola mòbil
- b) Nom: Missatge enviat al mòbil

9- Connecta els 2 darrers nodes que has creat . Fes un **Deploy** (desplegament) del programa.

10- Executa **UDP Receiver and Sender**. Fes que el seu port UDP Receiver sigui el 2222. Envia missatges des del node inject i comprova que arriben al mòbil.

Apartat 6 - Treballant amb packets UDP, Node-RED i mòbils Android (Part 2)

1- Fes que el grup amb permisos especial del directori **/var/www/html** sigui **smx2**. Dona permís d'escriptura al grup **smx2**. Executa com a **root**:

```
chgrp smx2 /var/www/html  
chmod g+w /var/www/html
```

2- Afegeix un node **storage - file d'escriptura** que:

- a) Treballi amb el fitxer **/var/www/html/mobil.html**
- b) Acció: Append to file
- c) Nom: web de missatges del mòbil

3- Connecta-ho al node de tipus input udp creat al segon pas de l'apartat anterior i que es diu "missatge enviat des del mòbil".

4- Fes un despleament.

6- Executa **UDP Sender**. Escriu l'adreça IP del teu Linux amb el Node-RED. Indica el port 3333. Escriu a Text el missatge “**Hola Node-RED del del meu mòbil
”. Fes 5 clics a Send. Comprova que el missatge surt per la finestra “Debug messages i també s'afegeix a **mobil.html”.

7- Visualitza amb el navegador la web que s'ha creat.

8- Ara crea una funció de nom Afegeix Data amb aquest codi:

```
var dateNow = new Date();  
var timeAsString = dateNow.toLocaleTimeString();  
  
msg.payload = msg.payload + ' a les: ' + timeAsString + '<br>';  
  
return msg;
```

9- Connecta l'entrada de “Afegeix Data” a la sortida de “missatge enviat des del mòbil”. Connecta la sortida de “Afegeix Data” a l'entrada de “web de missatges del mòbil”.

10- Fes un desplegament.

11- Executa **UDP Sender**. Escriu l'adreça IP del teu Linux amb el Node-RED. Indica el port 3333. Escriu a Text el missatge “**Hola Node-RED del del meu mòbil**”. Fes uns quants clics a Send. Comprova que el missatge surt per la finestra “Debug messages i també s'afegeix a **mobil.html**”.

Apartat 7 - Treballant amb Node-RED i telegram (Part 1)

1- Com usuari **smx2** entra dins del directori **.node-red**. Executa:
`cd .node-red`

2- Dins del directori **.node-red** i com usuari **smx2** executa la següent ordre per descarregar el mòdul **Dashboard de Node-Red**:
`npm i node-red-contrib-telegrambot`

3- Com a **root**, reinicia **Node-Red** perquè els canvis tinguin efecte executant:
`systemctl restart node-red`

4- Obre aquesta pàgina: <https://www.sohamkamani.com/blog/2016/09/21/making-a-telegram-bot/>

5- Obre el teu telegram i la secció de cerca escriu **@botfather** per trobar el bot de nom botfather que és el pare de tots els usuaris bot de telegra. Un bot és una aplicació que es comporta com un usuari de telegram. Es pot fer una aplicació d'aquest tipus amb node-RED.

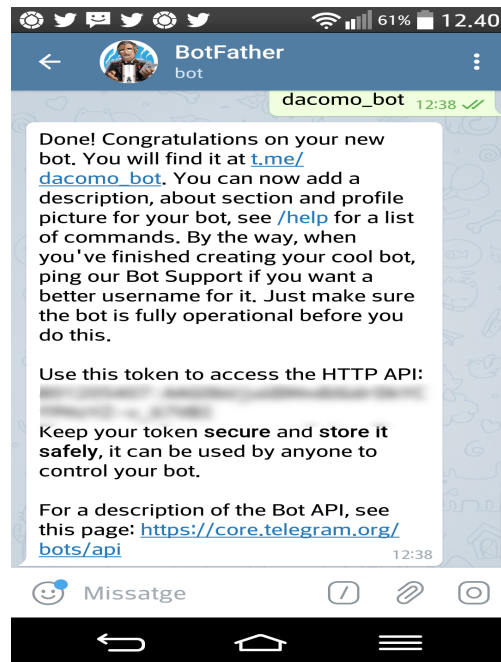
6- Dins del chat amb **@botfather**, envia els missatges:

/newbot

xyyyz bot (a on xx són les dues primeres lletres del teu nom, yy són les dues primeres lletres del teu primer cognom i z són les dues primeres lletres del teu 2n cognom)

xyyyz_bot (el guió baix és important)

7- Mira que en el darrer missatge enviat per @BotFather hi ha un codi anomenat **token**. Fes una captura de pantalla amb el teu mòbil per no perdre'l:



El token correspon a la zona difuminada.

8- Ara busca el teu bot a telegram i obre el chat.

9- Crea un nou Flow de nom **chat_telegram_xxyyzz_boot** (a on xx són les dues primeres lletres del teu nom, yy són les dues primeres lletres del teu primer cognom i z són les dues primeres lletres del teu 2n cognom).

10- Afegeix un node de tipus **telegram sender**:

a) Afegeix un nou bot de telegram amb aquestes característiques:

Bot-Name: **xxyyzz_bot**

Token: Escriu el token enviat via telegram pel teu bot.

b) Nom del node: **envia_missatge_a_xxyyzz_bot**

11- Afegeix un node de tipus **telegram receiver**

a) Afegeix el bot de telegram **xxyyzz_bot**

b) Download Directory: **/home/smx2/Documents**

b) Nom del node: **rep_missatge_de_xxyyzz_bot**

12- Connecta l'entrada del node de **telegram sender** a la primera sortida del node **telegram receiver**.

13- Deshabilita la resta de Flows.

14- Selecció dashboard --> layout --> Tabs & Links --> +tab. Selecció la nova tab, fes clic a edit, i escriu:

a) nou nom = **tab_telegram_dacomo_bot**

b) Estat: **enable**

c) Nav. Menu: Visible

i després fes un **Update**.

15- selecciona el nodes de tipus dashboard **gauge**. Edita'l:

a) Grup

Membre de tab **tab_telegram_dacomo_bot**

Nom = **missatge_de_telegram_dacomo_bot**

b) Range: **0 a 1000**

c) Units: **Mbps**

d) Name del node: **missatge_de_telegram_dacomo_bot**

i després fes un **Done**.

16- Crea un funció de nom **ObtenirValor** amb el contingut:

```
msg.payload=msg.payload.content ;  
return msg;
```

17- Connecta la primera **sortida** del node de **telegram receiver** al l'**entrada** de la funció **ObtenirValor**.

18- Connecta la sortida de la funció **ObtenirValor** a l'entrada del node de tipus **gauge**.

19- Fes un deploy

20- Escriu **/start** al teu **bot** de telegram per començar les comunicacions.

21- Envia valors entre 0 i 1000 des del bot i comprova que canvia l'indicador del dashboard.

Apartat 8 - Treballant amb Node-RED i telegram (Part 2)

1- Deshabilita la resta de Flows. Crea un nou Flow de nom **ChatIdTelegram**.

2- Afegeix un node de tipus **telegram receiver**:

a) Selecciona el bot **xyyyz_bot** de l'apartat anterior

b) Download Directory: **en blanc**

c) Nom del node: **xyyyz_bot**

3- Crea un **funció** de nom **ObtenirChatId** amb el contingut:

```
msg.payload=msg.payload.chatId;  
return msg;
```

4- Afegeix un node tipus **output debug** que:

a) Doni com a sortida el **msg.payload**

b) El nom sigui **mostra_ChatId**

5- Connecta la primera **sortida** del node de **telegram receiver** al l'**entrada** de la funció **ObtenirChatId**.

6- Connecta la sortida de la funció **ObtenirChatId** a l'entrada del node de tipus **output debug**.

7- Fes un **Deploy** (desplegament) del programa i després selecciona l'eina "**Debug messages**". Esborra qualsevol missatge anterior.

8- Escriu un missatge des del teu **bot** i comprova que pots obtenir el teu ChatId. Apunta'l..

Apartat 9 - Treballant amb Node-RED i telegram (Part 3)

1- Com usuari **smx2** entra dins del directori **.node-red**. Executa:

```
cd .node-red
```

2- Dins del directori **.node-red** i com usuari **smx2** executa la següent ordre per descarregar el mòdul **Dashboard** de **Node-Red**:

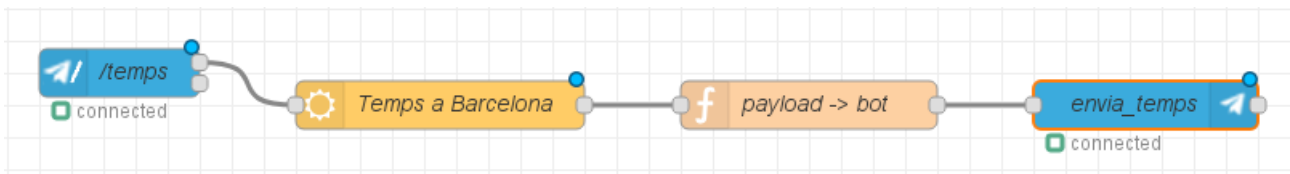
```
npm i node-red-node-openweathermap
```

3- Com a **root**, reinicia **Node-Red** perquè els canvis tinguin efecte executant:

```
systemctl restart node-red
```

4- Aconseguir una clau per poder treballar amb node-red i openweathermap. Connecta't a **openweathermap.org**. Crea un compte d'openweathermap. Després demana una API Key i copia-la.

5- Deshabilita altres flows. Crea un flow de nom **temps a Barcelona**. Crea el següent flow de node-red:



6- El primer node és de tipus **telegram command**. Ha d'estar connectat amb el teu **bot** de telegram. L'ordre que s'ha d'escriure en telegram per actuar és **/temps**. El nom és **/temps**.

7- Configura el node openweather així:

La imatge mostra la configuració del node openweathermap a Node-RED:

- API Key**: [.....]
- Language**: Catalan
- Location**: Current weather for
- City**: Barcelona
- Country**: Spain
- Name**: Temps a Barcelona

8- La funció **payload -> bot** ha de ser aquesta:

```
msg.payload = {  
  "content": msg.payload.detail,  
  "chatId" : xxxxxxxx,  
  "type" : "message"  
}
```

```
return msg;
```

A on has de canviar xxxxxxx pel teu ChatId.

9- El darrer node ha de ser de tipus **telegram sender**. Ha d'estar connectat amb el teu **bot** de telegram. El nom és **envia_temps**.


10- Fes un Deploy.

11- Des del teu **bot** envia l'ordre **/temps** i espera a rebre com a resposta l'estat del temps a Barcelona.

Apartat 10 - Treballant amb Node-RED, telegram i nano arduino sobre placa make it!

1- [Descarrega](#) i instal·la el driver de Windows **CH341SER** per poder realitzar la comunicació de Windows amb la placa **Makeit!**. Per instal·lar-lo, executa el programa **SETUP** que pots trobar un cop has descomprimit el fitxer descarregat.

2- Connecta la placa **Makeit!** al port USB del teu ordinador. Comprova que a l'Administrador de dispositius de Windows ha aparegut **Ports (COM i LPT)** i a dins **USB-SERIAL CH340 (COM3)**. El valor del COM pot ser diferent a cada ordinador. Si això surt, vol dir que Windows té connexió amb la placa **Makeit!**.

3- Transfereix la connexió de Windows a Linux. Fes clic amb el botó de la dreta a la icona  i selecciona l'opció "**QinHeng Electronics USB2.0-Serial**". Després executa:

```
lsusb
```

i comprova que a la llista surt **QinHeng Electronics HL-340 USB-Serial adapter**. Ja hem transferit la connexió a la màquina virtual Linux.

4- Entra com a **root** a Linux i executa:

```
gpasswd -a smx2 dialout
```

5- Surt d'usuari **root** i entra com usuari **smx2**. Executa:

```
id
```

i comprova que a la llista surt el grup **dialout**. Ara l'usuari **smx2** pot comunicar-se amb la placa **Makeit!** a través de la connexió del **QinHeng Electronics HL-340 USB-Serial adapter**.

6- Com usuari **smx2** executa:

```
ls -a /dev/ttyUSB0
```

i si la resposta és:

```
/dev/ttyUSB0
```

llavors la connexió **QinHeng Electronics HL-340 USB-Serial adapter** sabem que el Linux la identifica com el dispositiu **/dev/ttyUSB0**.

7- Com usuari **smx2** entra dins del directori **.node-red**. Executa:

```
cd .node-red
```

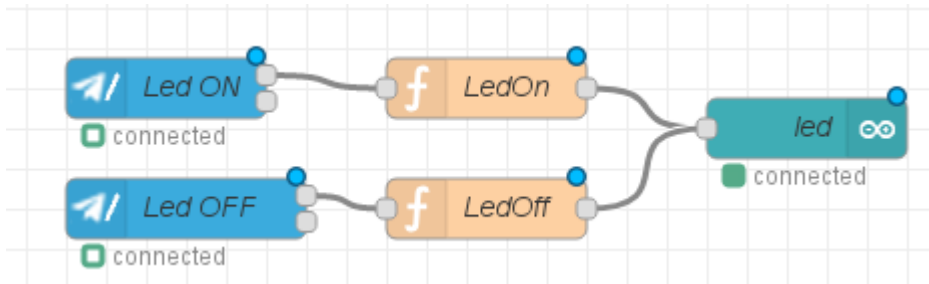
8- Dins del directori **.node-red** i com usuari **smx2** executa la següent ordre per descarregar el mòdul **Dashboard** de **Node-Red**:

```
npm i node-red-node-arduino
```

9- Com a **root**, reinicia **Node-Red** perquè els canvis tinguin efecte executant:

```
systemctl restart node-red
```


10- Crea un Flow de nom **Makeit** amb els següents nodes i connexions:



11- El primer node és de tipus **telegram command**. Ha d'estar connectat amb el teu **bot** de telegram. L'ordre que s'ha d'escriure en telegram per actuar és **/on**. El nom del node és **Led ON**.

12- El segon node és de tipus **telegram command**. Ha d'estar connectat amb el teu **bot** de telegram. L'ordre que s'ha d'escriure en telegram per actuar és **/off**. El nom del node és **Led OFF**.

13- El tercer node és una funció de nom **LedON** amb aquest programa:

```
msg.payload=true;  
return msg;
```

14- El quart node és una funció de nom **LedOFF** amb aquest programa:

```
msg.payload=false;  
return msg;
```

15- El darrer node és de tipus arduino output amb la següent configuració:

```
Port: /dev/ttyUSB0   Pin: 13  
Tipus: Digital (0/1)   Name: Control Led Pin 13
```

16- Des del teu bot de telegram envia l'ordre **/on** i l'ordre **/off** i comprova que passa amb el Led del Pin 13 de la placa **Makelt!**.