

# STiMA WiFi

Stazione di monitoraggio ambientale



Configurazione

Firmware + Software

Python + Json

NodeRed

# Configurazione



Configurazione

Firmware + Software

Python + Json

NodeRed

# Configurazione stazione

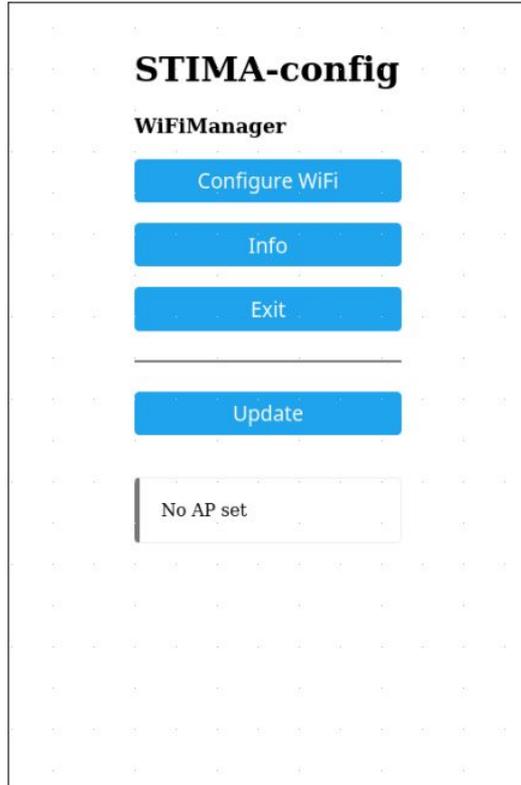
*Alla prima accensione, o meglio quando la stazione non è configurata, la stazione Stima crea una rete WiFi che si chiama*

***STIMA-config***

*con password di accesso (alla rete WiFi)*

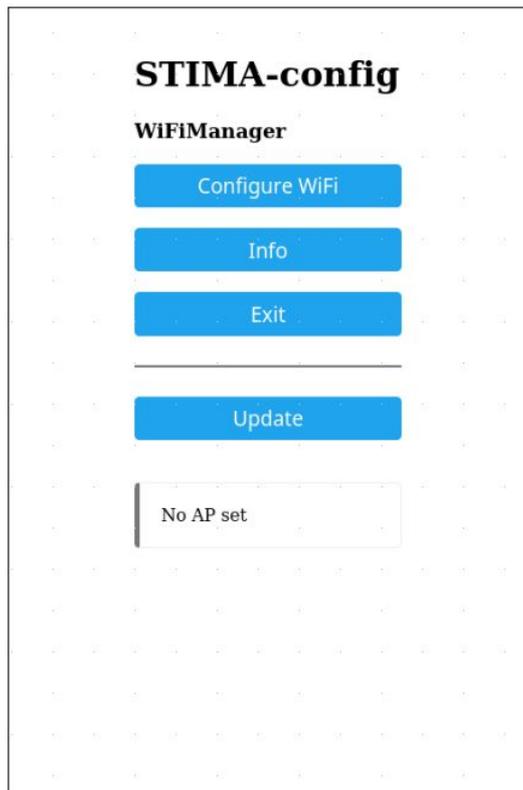
***bellastima***

# Configurazione stazione



Una volta collegati alla rete, il browser viene automaticamente portato alla pagina iniziale di configurazione della stazione

# Configurazione stazione



Una volta collegati alla rete, il browser viene automaticamente portato alla pagina iniziale di configurazione della stazione

Se ciò non si verifica aprire il browser (firefox o chrome, per esempio) e inserire nella barra di navigazione l'indirizzo

**<http://192.168.4.1>**

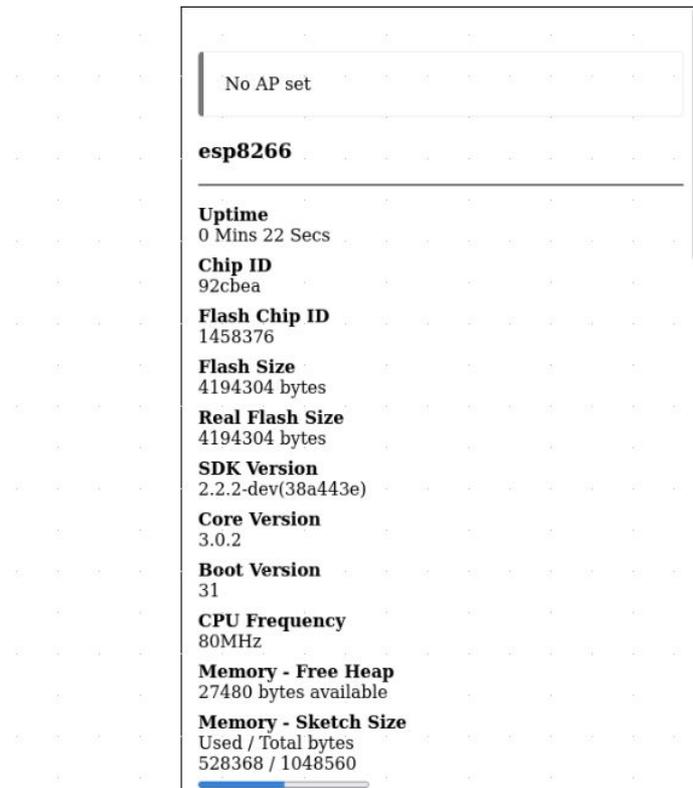
**ATTENZIONE il collegamento NON è https**

# Configurazione stazione

Il pulsante **INFO** porta alle informazioni sull'hardware che compone la stazione

Qui si trova il MAC address della stazione

Da usare per autenticazione differenziata sui server della scuola



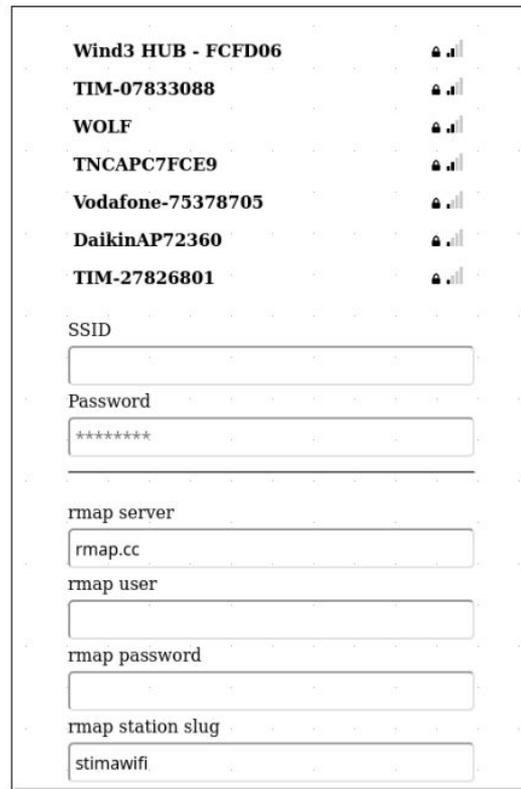
# Configurazione stazione

Il pulsante **Configure WiFi** ,

permette di scegliere la rete WiFi a cui collegare la stazione

STIMA-config serve solo alla configurazione

Basta selezionare la rete desiderata tra quelle rilevate dalla stazione



The screenshot shows a configuration window with a list of detected WiFi networks and several input fields. The networks listed are:

- Wind3 HUB - FCFD06
- TIM-07833088
- WOLF
- TNCAPC7FCE9
- Vodafone-75378705
- DaikinAP72360
- TIM-27826801

Below the list, there are input fields for:

- SSID (empty)
- Password (masked with asterisks)
- rmap server (rmap.cc)
- rmap user (empty)
- rmap password (empty)
- rmap station slug (stimawifi)

# Configurazione stazione

Va inserita la password di accesso alla WiFi (NON quella a RMAP)

Dopo i parametri per l'accesso ad internet vanno inserite le informazioni necessarie all'invio dei dati.

TIM-27826801

SSID  
Vodafone-75378705

Password  
\*\*\*\*\*

rmap server  
rmap.cc

rmap user

rmap password

rmap station slug  
stimawifi

Save

Refresh

No AP set

# Configurazione stazione

- il server su cui è registrata la stazione
- il nome dell'utente che ha registrato la stazione
- la password inserita nella pagina di registrazione della stazione
- lo SLUG della stazione

TIM-27826801

SSID  
Vodafone-75378705

Password  
\*\*\*\*\*

rmap server  
rmap.cc

rmap user

rmap password

rmap station slug  
stimawifi

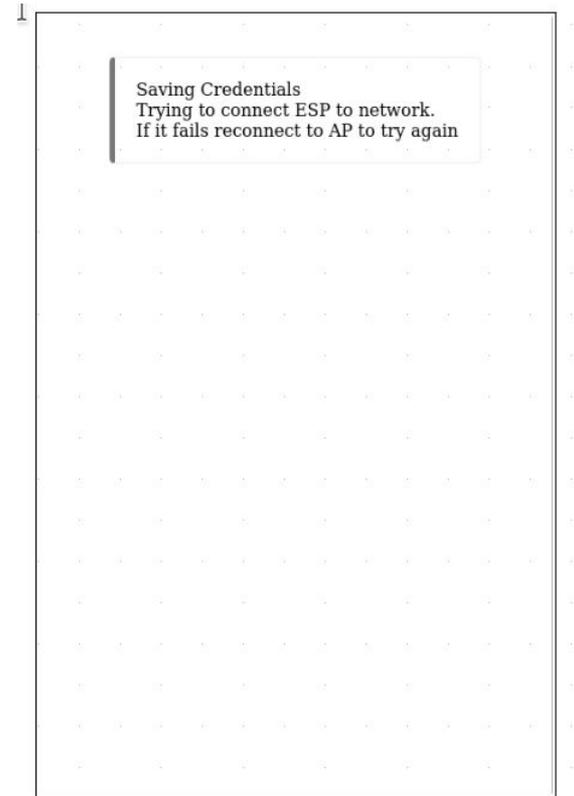
Save

Refresh

No AP set

# Configurazione stazione

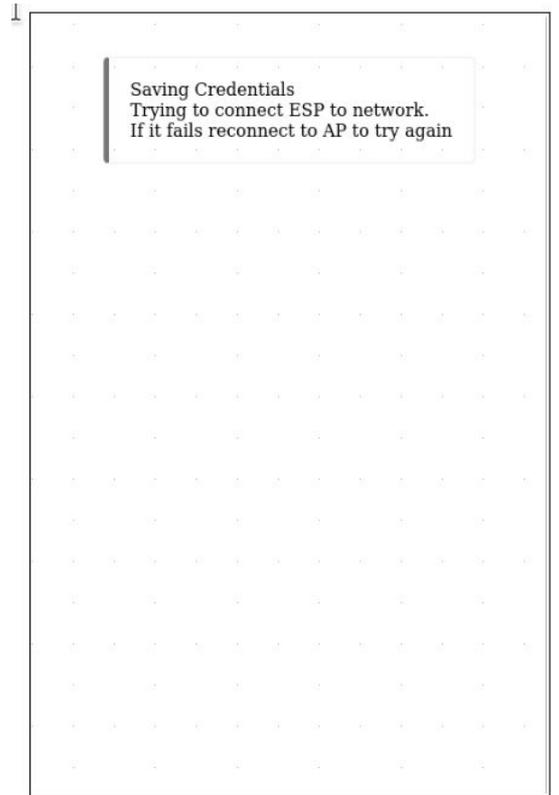
Se tutto va per il meglio, comparirà la maschera di conferma configurazione e la stazione si riavvierà per usare i parametri inseriti e inviare i dati al server.



# Configurazione stazione

Se tutto va per il meglio, comparirà la maschera di conferma configurazione e la stazione si riavvierà per usare i parametri inseriti e inviare i dati al server.

In caso contrario (controllare anche sul display in dotazione all'apparato) bisogna ripartire con la procedura di configurazione.



# RMAP

Rete di Monitoraggio Ambientale Partecipativa



Configurazione

Firmware + Software

Python + Json

NodeRed

# Firmware vs Software/Middleware

I programmi interagiscono con la stazione di monitoraggio su più livelli

## FIRMWARE

- C++ / Wiring
- Basso livello
- Fortemente accoppiato ad hardware
- Interfaccia uniforme strati superiori
- Descrive come svolgere compiti
- Impegnativo da scrivere
- Poco significativo per la didattica<sup>1</sup>

## MIDDLEWARE (Software)

- Molteplici linguaggi di sviluppo
- Alto livello / remoto
- Adattabile ad hardware diversi
- 
- Determina quali compiti svolgere
- Trasformazione e analisi dati raccolti

[1] Se non si insegna/studia informatica, elettronica o materie fortemente correlate

# RMAP

Rete di Monitoraggio Ambientale Partecipativa



Configurazione  
Firmware + Software  
Python + Json  
NodeRed

client.py ×

home > giacomo > Spike > rmap > client.py > ...

```
1  #!/usr/bin/python3
2
3  import json
4  import urllib3
5  http = urllib3.PoolManager()
6  url_stazione = "margherita.local"
7
8  resp = http.request('GET', f'http://{url_stazione}/data.json')
9  if resp.status != 200:
10     |   campionamenti = json.loads(resp.data)
11     |   print (campionamenti)
12     |   print (f"Particolato (PM10) {campionamenti['PM10']} ug/m³")
13 else:
14     |   print(f"Error retrieving data from {url_stazione}: Code {resp.status}")
15
```

## Formato Payload interrogazione diretta

```
{ "TEMP" : "nan" , "HUMID" : "-999" , "PM2" : "11" , "PM10" : "11"  
  , "CO2" : "1921" , "STAT" : "Connected" }
```

## Formato Payload interrogazione diretta

```
{ "TEMP": "nan", "HUMID": "-999", "PM2": "11", "PM10": "11",  
  "CO2": "1921", "STAT": "Connected" }
```

## Formato Payload interrogazione diretta

```
{  
  "TEMP" : "nan",  
  "HUMID" : "-999",  
  "PM2" : "11",  
  "PM10" : "11",  
  "CO2" : "1921",  
  "STAT" : "Connected"  
}
```

home &gt; giacomo &gt; Spike &gt; rmap &gt; client.py &gt; ...

```
1  #!/usr/bin/python3
2
3  import json
4  import time
5  import urllib3
6
7  http = urllib3.PoolManager()
8  url_stazione = "margherita.local"
9
10 resp = http.request('GET', f'http://{url_stazione}/data.json')
11 if resp.status == 200:
12     campionamenti = json.loads(resp.data)
13     campionamenti['time'] = time.strftime("%Y-%m-%dT%H:%m:%S %Z")
14     print (campionamenti)
15     print (f"Particolato (PM10) {campionamenti['PM10']} ug/m³")
16 else:
17     print(f"Error retrieving data from {url_stazione}: Code {resp.status}")
18
```

## Formato Payload interrogazione diretta

```
{  
  "TEMP": "nan",  
  "HUMID": "-999",  
  "PM2": "11",  
  "PM10": "11",  
  "CO2": "1921",  
  "STAT": "Connected",  
  "time": "2022-11-12T13:14:15 CET"  
}
```

# JavaScript Object Notation

- JSON è un formato di scambio dati facile da leggere (per tutti)
- Si basa su due soli concetti:
  - insieme di coppie Nome: Valore
  - Elenco ordinato di Valori

# JavaScript Object Notation

L'intera documentazione è lunga una pagina reperibile all'indirizzo

<https://json.org/json-it.html>

# STiMA WiFi

Stazione di monitoraggio ambientale



Configurazione  
Python + Json  
NodeRed

# Node-RED

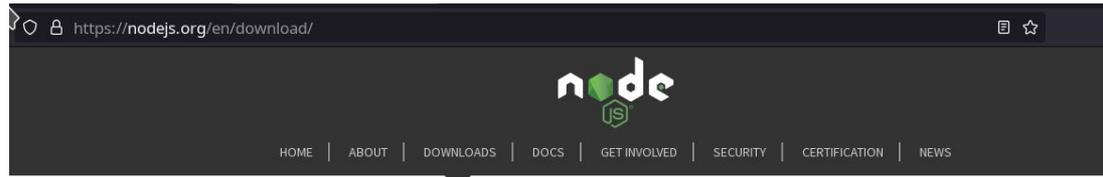
È un ambiente di programmazione pensato per far interagire

- dispositivi hardware
- API
- servizi online

È basato su Node.js e utilizza json sia come formato di scambio dati, sia per salvare e scambiare il codice sorgente in gergo **FLOW**

# Node-RED - Node.js

Per installarlo, si scarica ed installa Node.js per la propria piattaforma di riferimento.



## Downloads

Latest LTS Version: 18.12.1 (includes npm 8.19.2)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 <b>Windows Installer</b> node-v18.12.1-x86.msi	 <b>macOS Installer</b> node-v18.12.1.pkg	 <b>Source Code</b> node-v18.12.1.tar.gz

**Windows Installer (.msi)**

**Windows Binary (.zip)**

**macOS Installer (.pkg)**

**macOS Binary (.tar.gz)**

**Linux Binaries (x64)**

**Linux Binaries (ARM)**

**Source Code**

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v18.12.1.tar.gz	

## Additional Platforms

**Docker Image**

**Linux on Power LE Systems**

**Linux on System z**

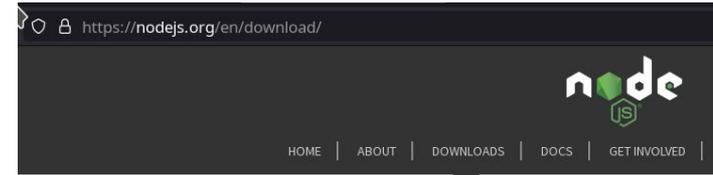
Official Node.js Docker Image	
64-bit	
64-bit	

# Node-RED - Node.js

Per installarlo, si scarica ed installa Node.js per la propria piattaforma di riferimento.

Si lancia il prompt dei comandi, o il terminale, e si esegue il comando per caricare l'intero ambiente di sviluppo in Node.js

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



## Downloads

Latest LTS Version: 18.12.1 (includes npm 8.19.2)

Download the Node.js source code or a pre-built installer for your platform



Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x64)

Linux Binaries (ARM)

Source Code

32-bit

32-bit

64-bit

ARMv7

## Additional Platforms

Docker Image

Linux on Power LE Systems

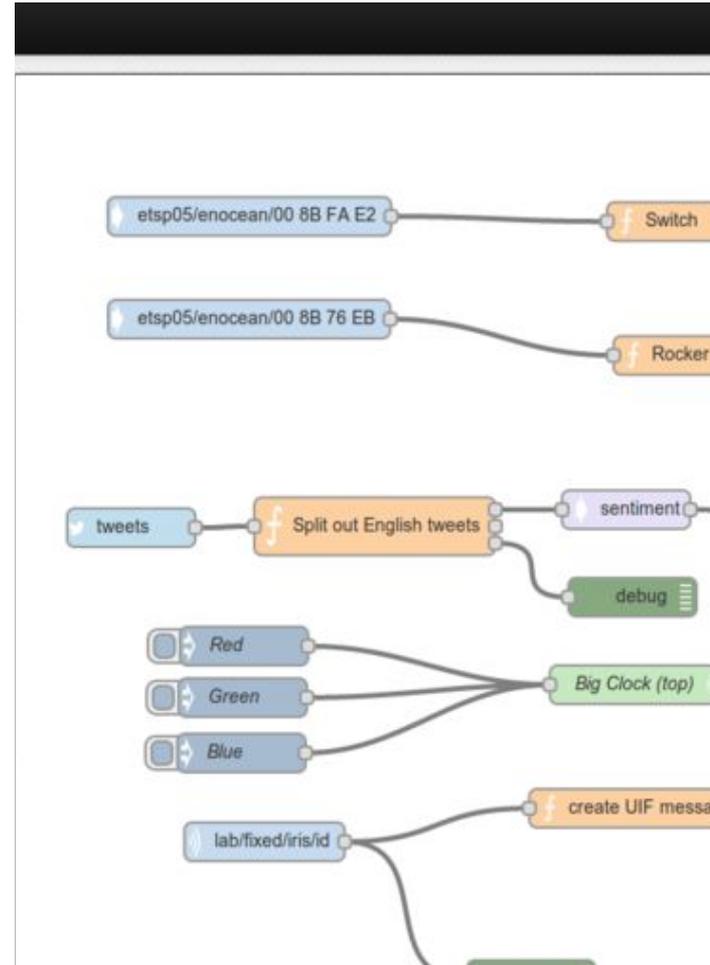
Linux on System z

Offic

# Node-RED - Node.js

Sempre da prompt dei comandi, o terminale, e si lancia l'ambiente di sviluppo

```
$ node-red
```



# Node-RED - Node.js

Sempre da prompt dei comandi, o terminale, e si lancia l'ambiente di sviluppo

```
$ node-red
```

```
Welcome to Node-RED
```

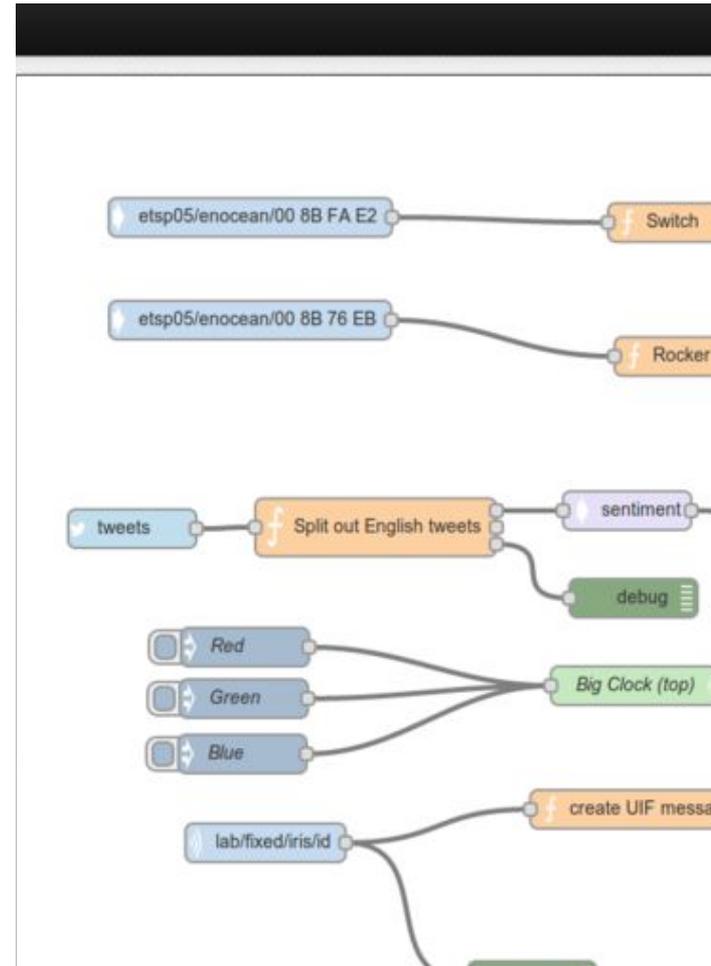
```
=====
```

```
30 Jun 23:43:39 - [info] Node-RED version: v1.3.5
```

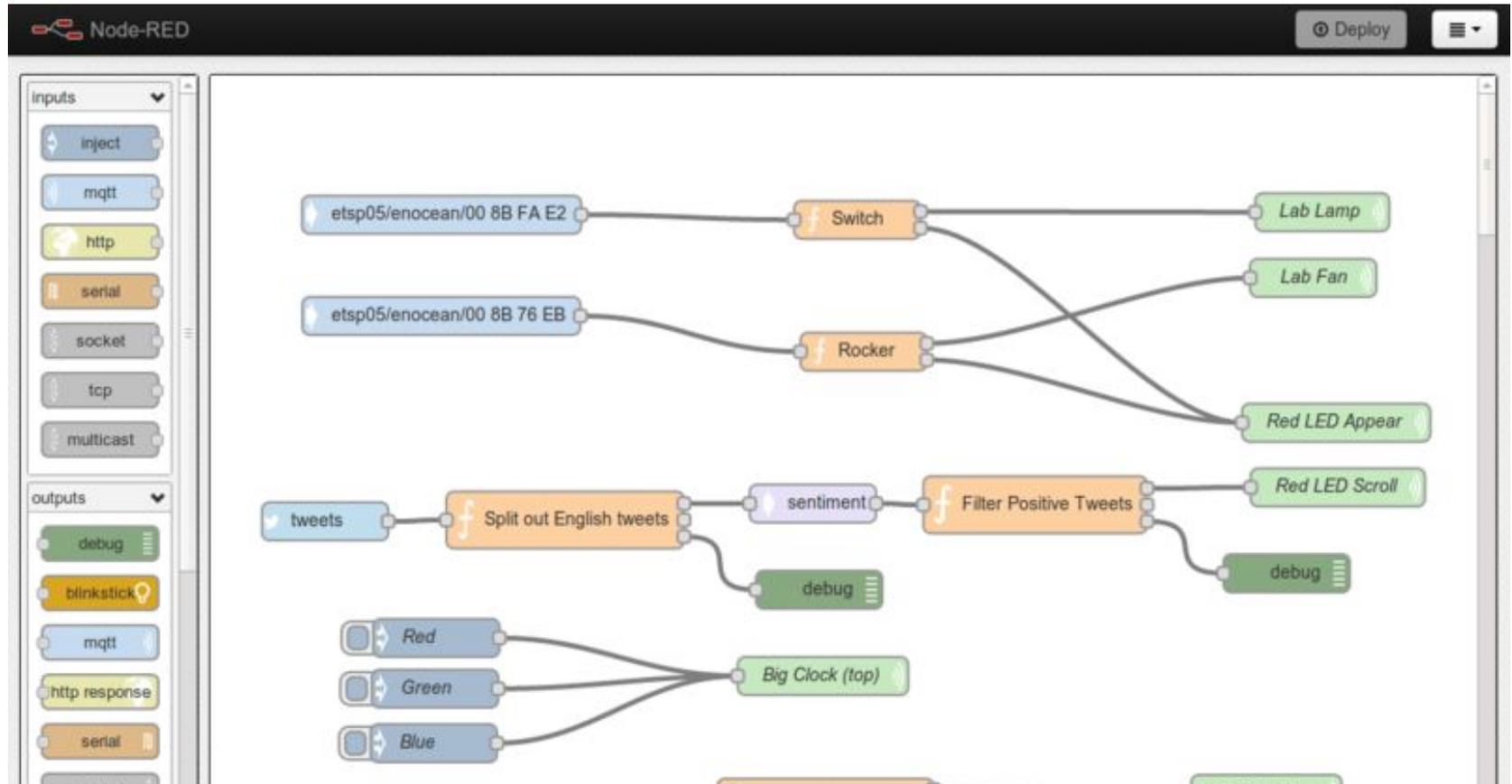
```
30 Jun 23:43:39 - [info] Node.js version: v14.7.2
```

```
...
```

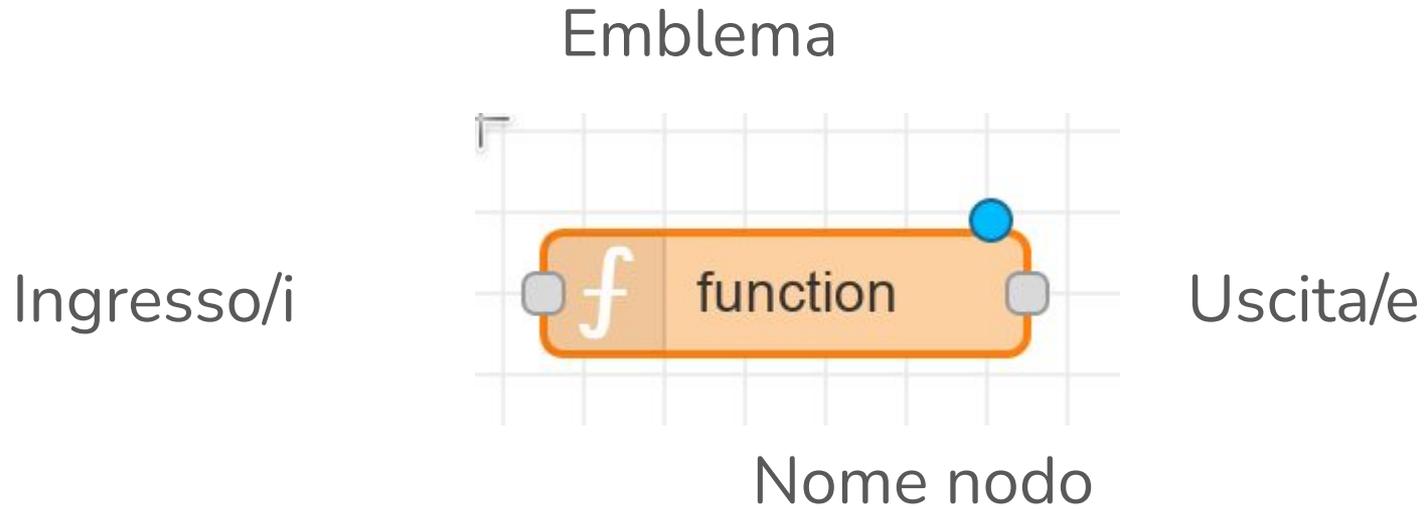
```
30 Jun 23:43:44 - [info] Server now running at  
http://127.0.0.1:1880/red/
```



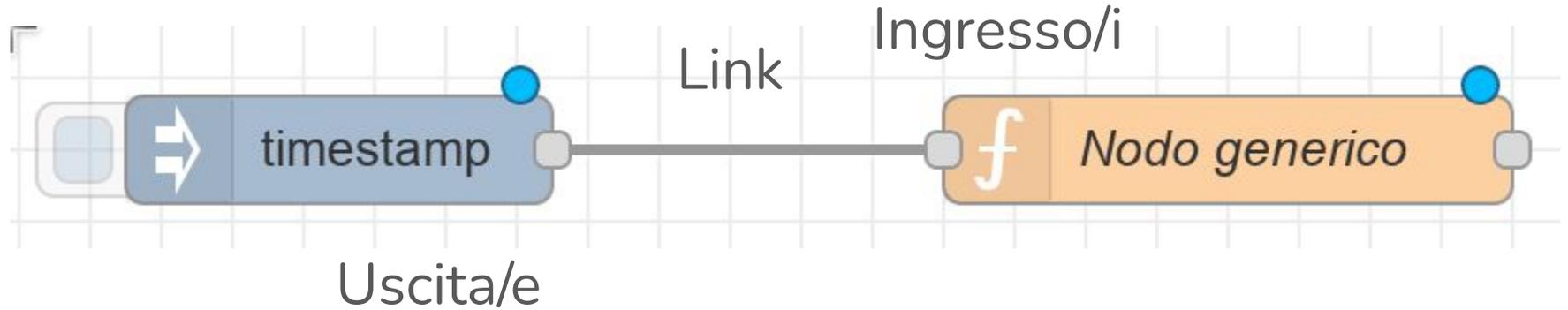
# Node-RED - Node.js



# Node-RED <Nodo>



## Node-RED <Nodo>



- Un link indica lo scambio di dati tra i due nodi.
- Ogni nodo elabora i dati in ingresso
- Il risultato viene inviato al nodo successivo.

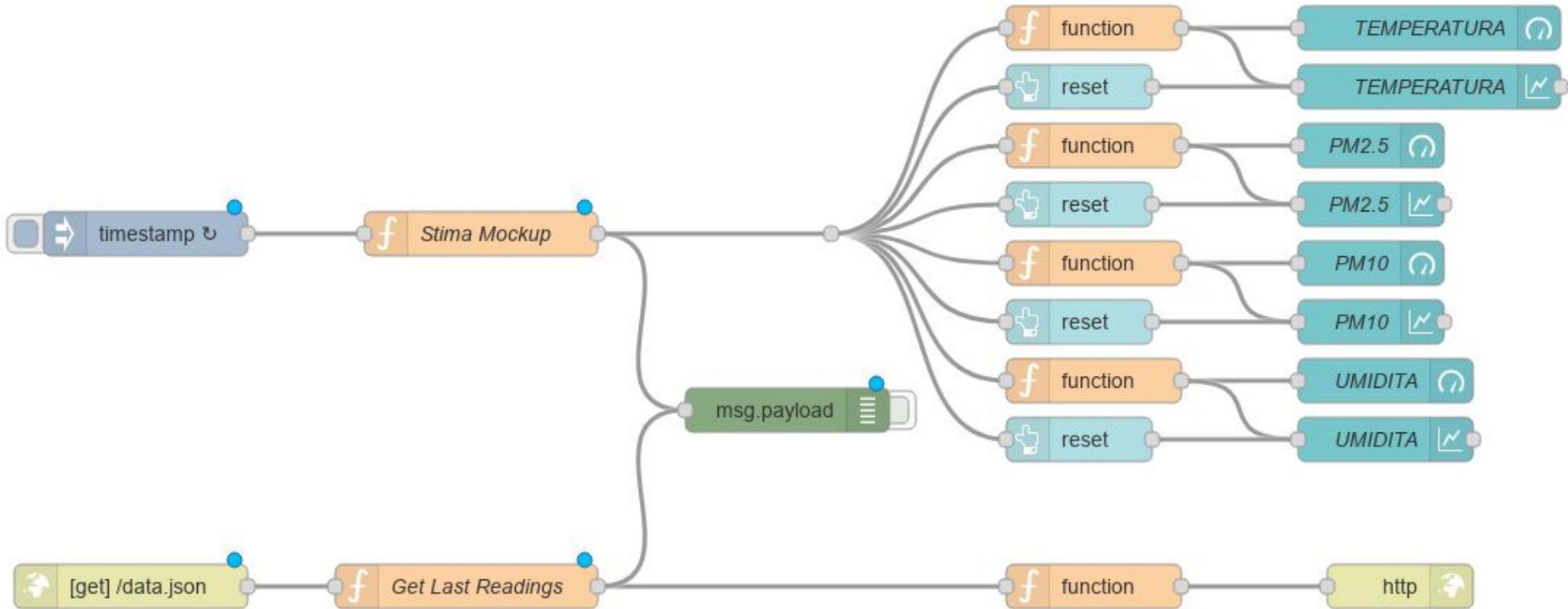
# Node-RED <Nodo>

Node-red mette a disposizione diverse tipologie di nodi.

Alcuni generici, altri specializzati, possono essere costruiti dall'utente programmatore se necessario.



# Node-RED Simulatore di Stima WiFi (FLOW)

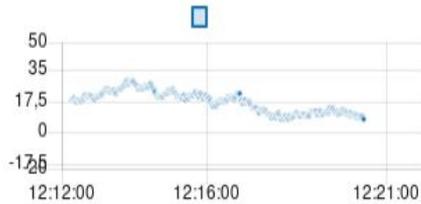


# Node-RED Simulatore di Stima WiFi (UI)

Home

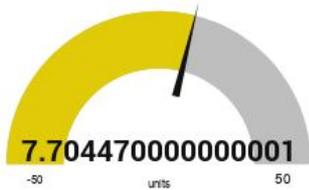
STIMA\_TEMP

chart



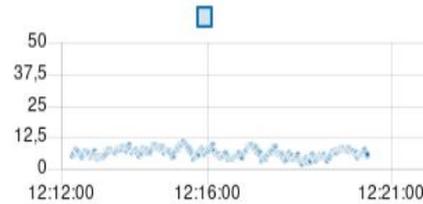
RESET

°C



STIMA\_PM2.5

chart



RESET

PM 2.5 $\mu$



STIMA\_PM10

chart



RESET

PM 10 $\mu$



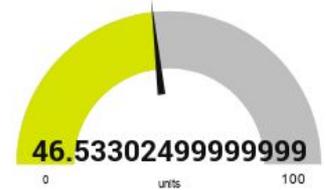
STIMA\_UMIDITA

chart



RESET

% Umidità



## Node-RED Simulatore di Stima WiFi (/data.json url)

```
{  
  "TEMP" : 10.90353,  
  "HUMID" : 47.3242,  
  "PM2" : 1.1022,  
  "PM10" : 6.4360,  
  "STAT" : "Simulator"  
}
```

# Verso l'infinito e oltre...

- <https://doc.rmap.cc>
- <https://github.com/r-map/rmap>
- <https://platformio.org>
- <https://www.arduino.cc/en/software>
- <https://nodered.org/docs/user-guide/>
- <https://nodered.org/>
- <https://www.python.it/doc/libri/#pensare-in-python-come-pensare-da-informatico>
- <https://docs.arduino.cc/micropython/>

Esplorare in autonomia l'ecosistema RMAP/Stima

Ambiente per programmazione embedded

Programmazione visuale

Introduzione non troppo formale a Python