



Gli Harduini

Ferrante Riccardo - 881061

Pappagallo Marco - 847963

Laboratorio di IOT

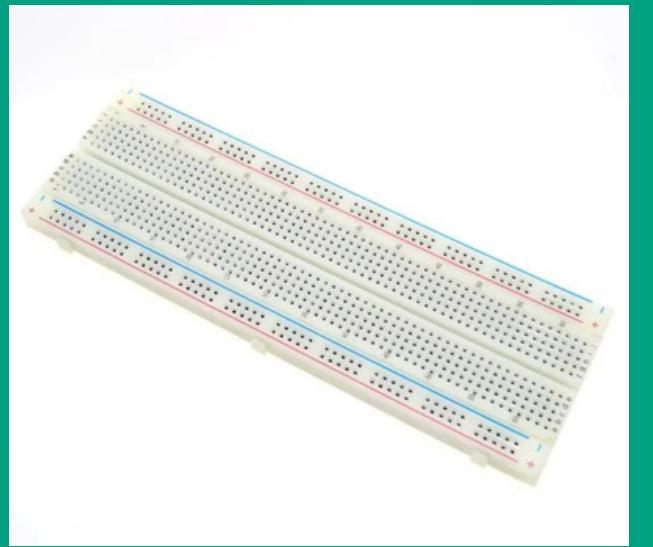
Prof. Paolo Napoletano & Flavio Maria De Paoli

a.a. 2022/2023

MATERIALS

LUA NODEMCU WIFI - ESP8266 (ESP-12E):

2 BREADBOARD Grandi - 830 CONTATTI



Supporto per l'ESP

3-COLOR RGB LEDMODULE (KY-016)



Led RGB

GROVE - Servo

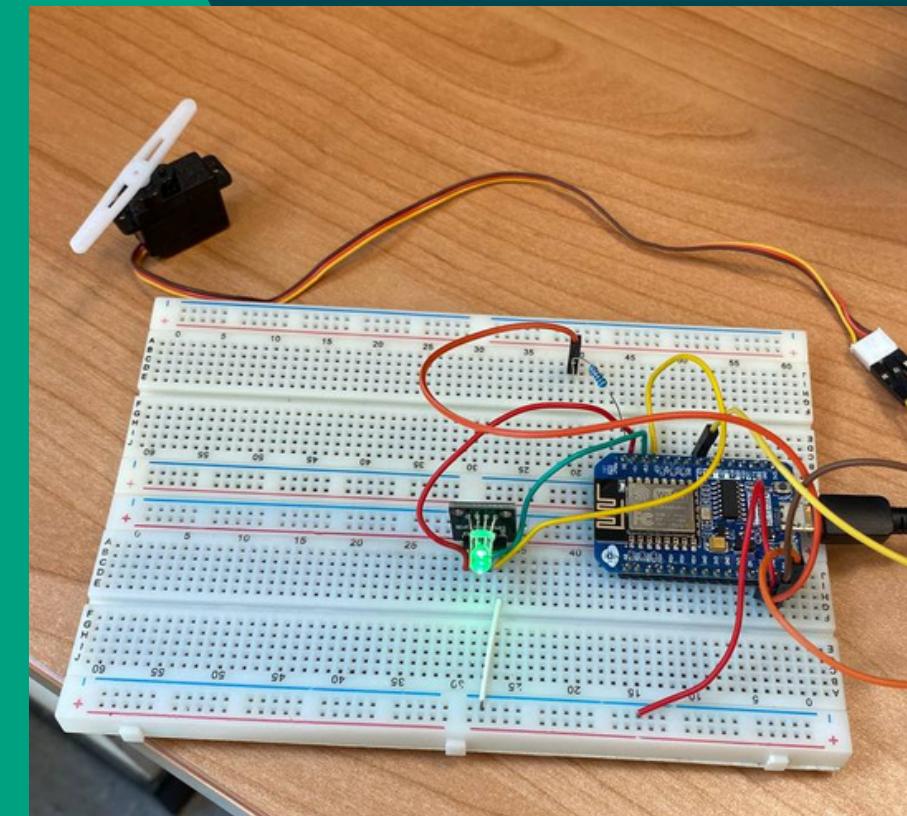


Simulazione motorino

CAVI JUMPER



Collegare i componenti all'ESP



MATERIALS

ARDUINO MKR WAN 1300 (LORA):

GROVE - Connector Carrier



Supporto per sensori e attuatori

GROVE - Cavi



Collegare i componenti

2 GROVE - Led Socket Kit + led



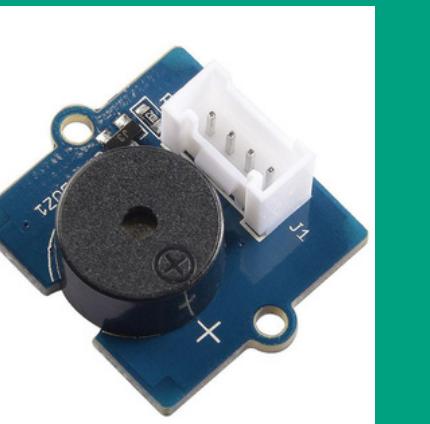
Supporto per led

GROVE - LCD RGB Backlight



Schermo di visualizzazione

GROVE - Buzzer

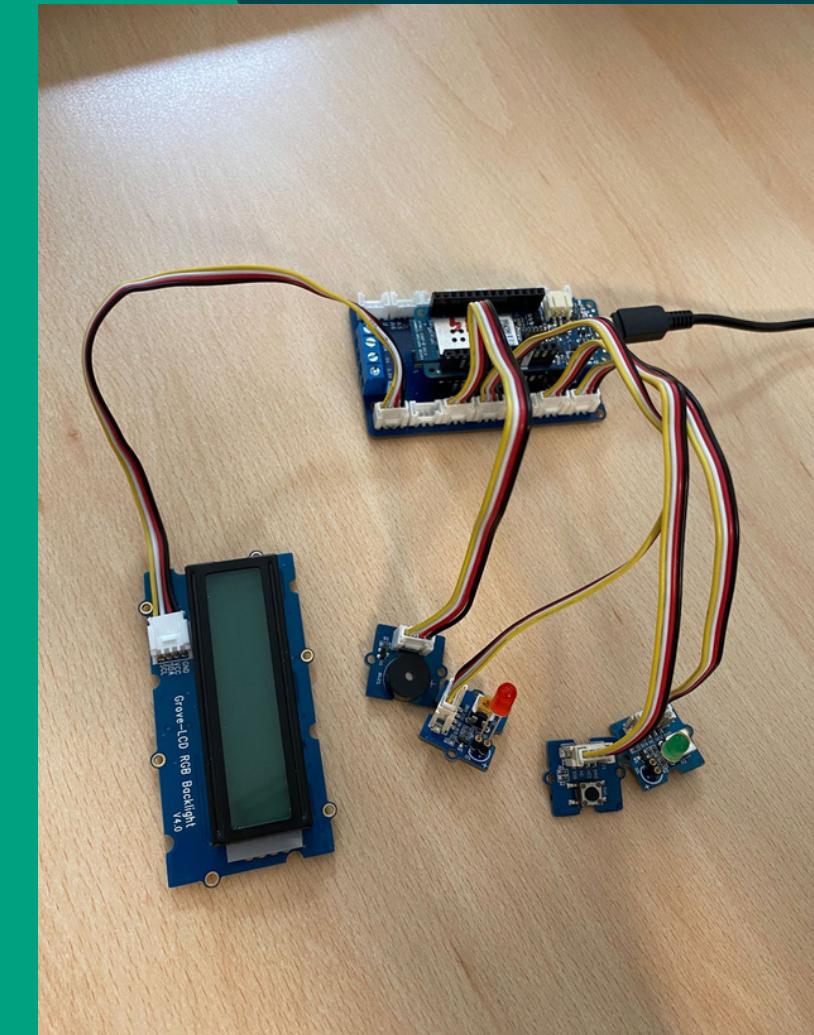


Emettitore di suono

GROVE - Button

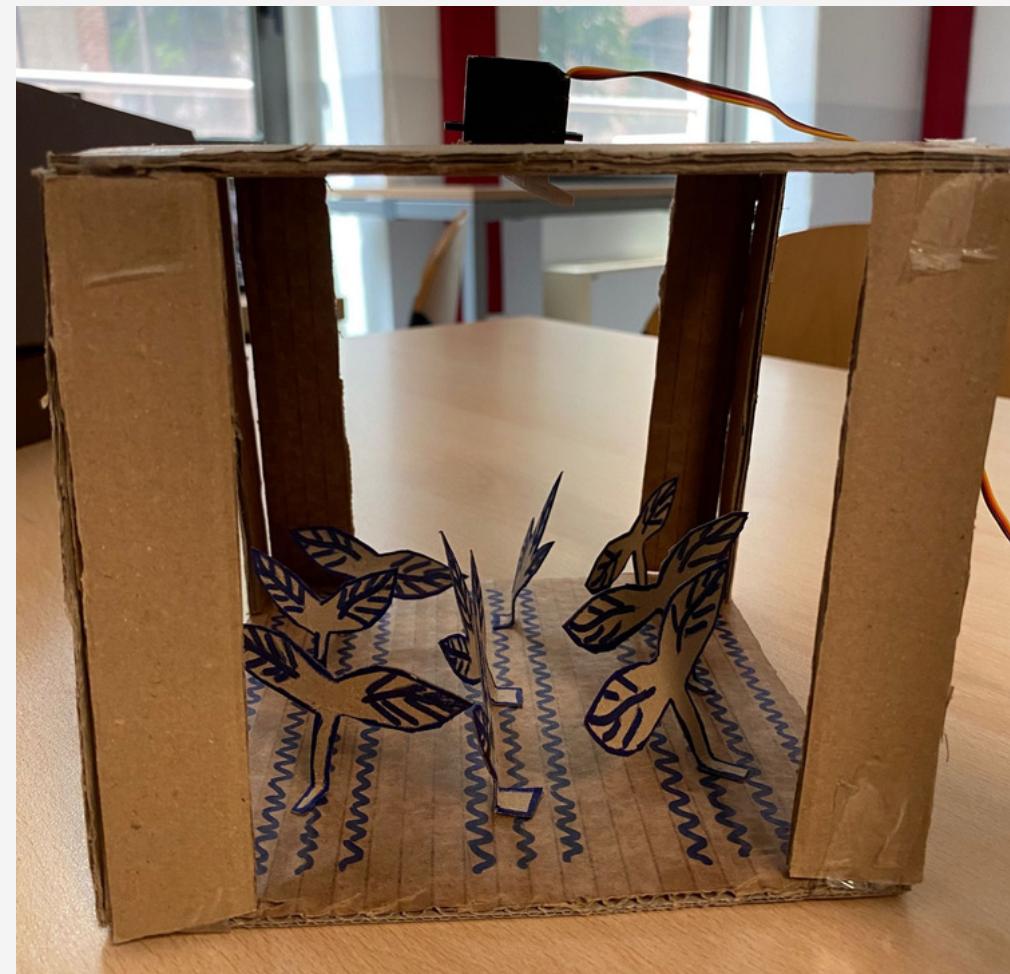


Bottone



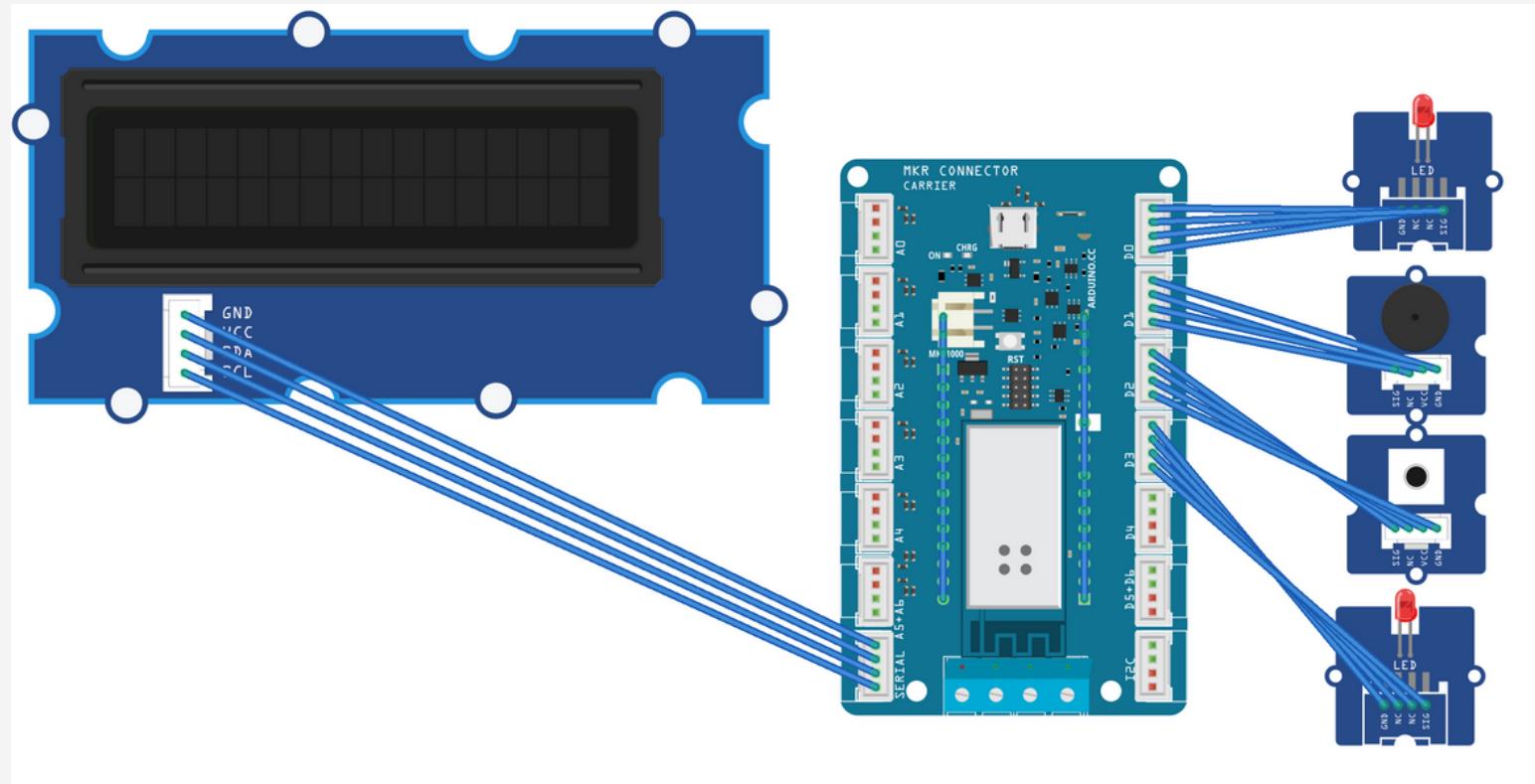
Prodotto: flowerbed

Come idea abbiamo optato per un sistema di monitoraggio e irrigazione di un'aiuola

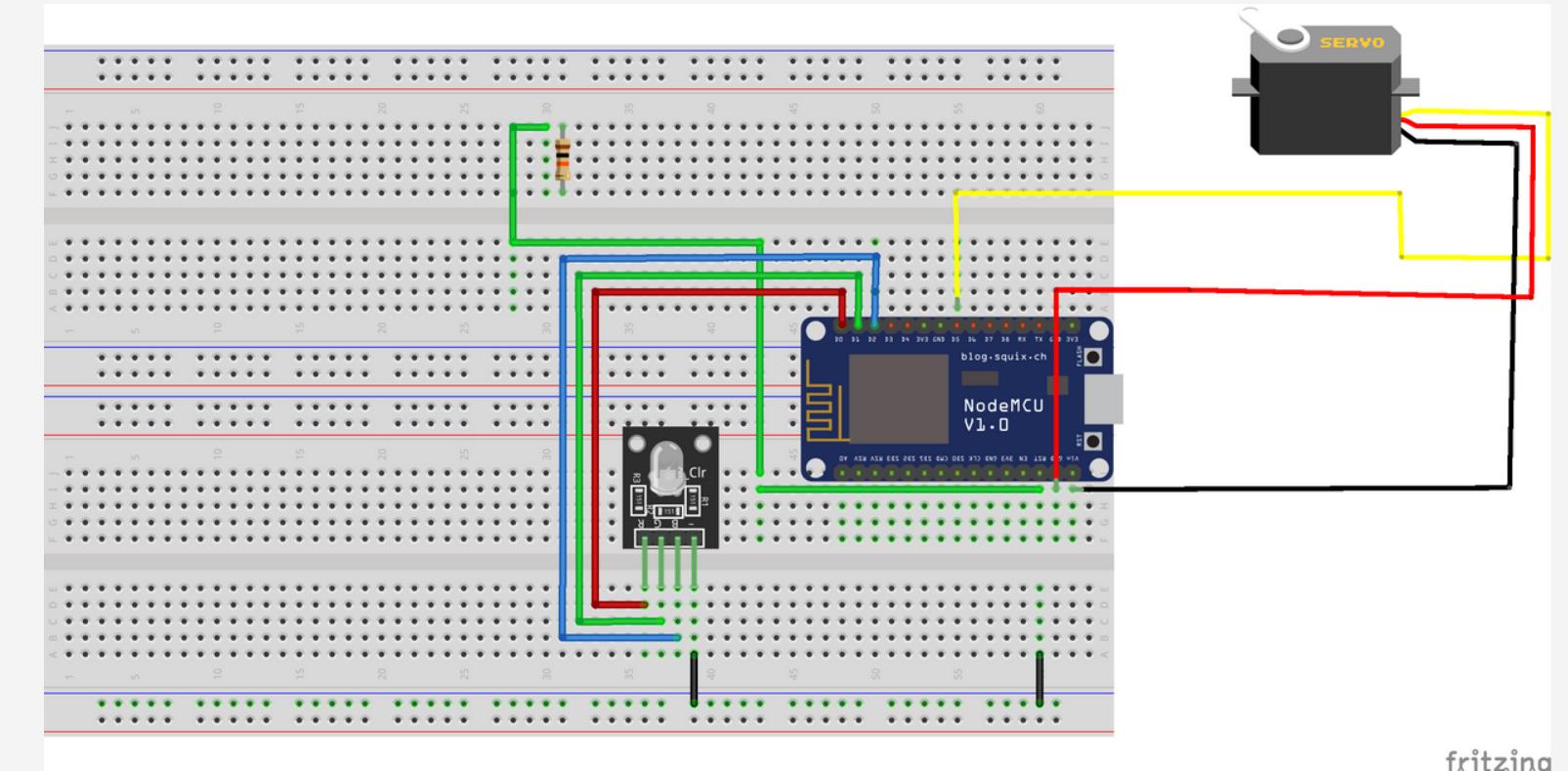


Micro-Controllers

"Master" MKR



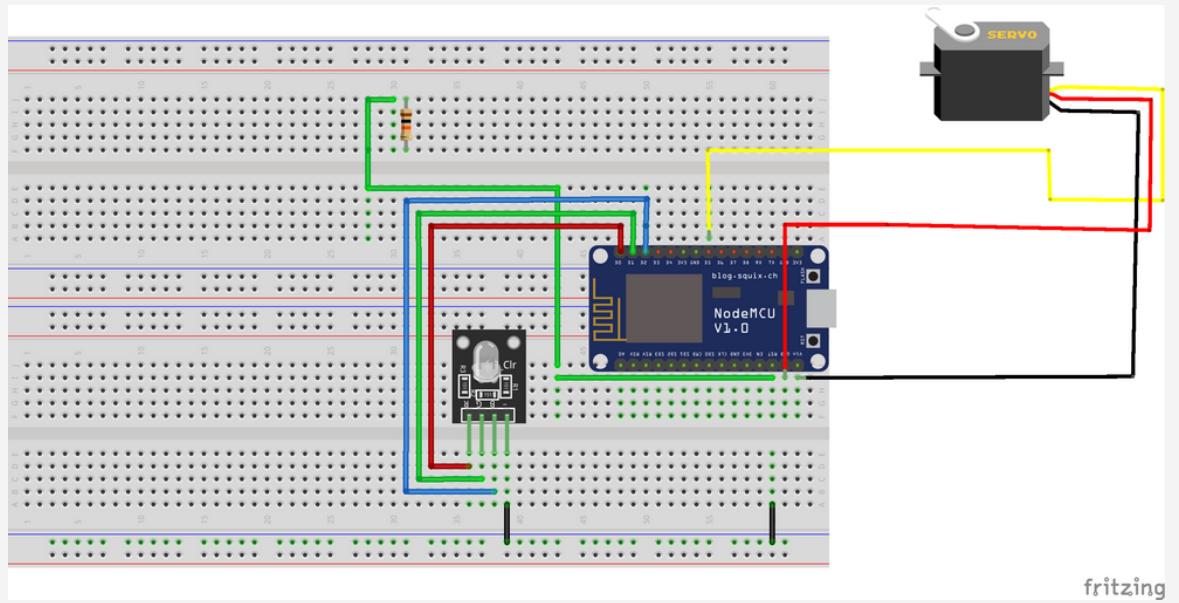
"Slave" ESP8266



FLOW

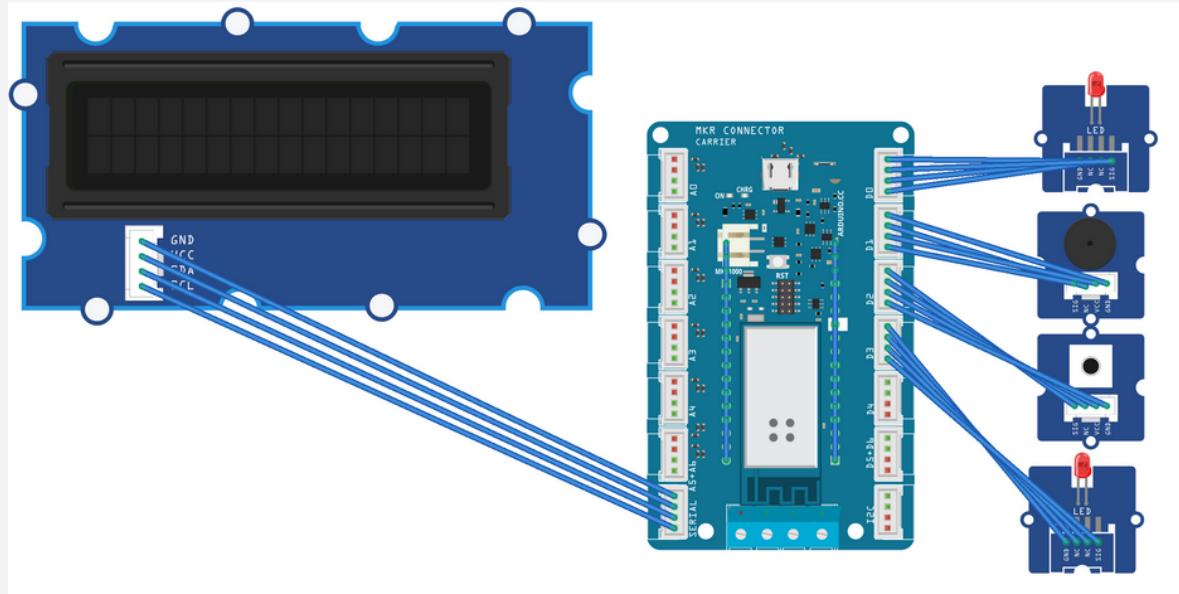
"Slave" ESP8266

- Chiamata dell'API;
 - Publish dati dell'API, tramite MQTT, al Master;
 - Ricezione input per attivare il Servo da Telegram;
 - Funzione Deep Sleep per risparmiare energia;
 - Registro dei dati su MySQL.



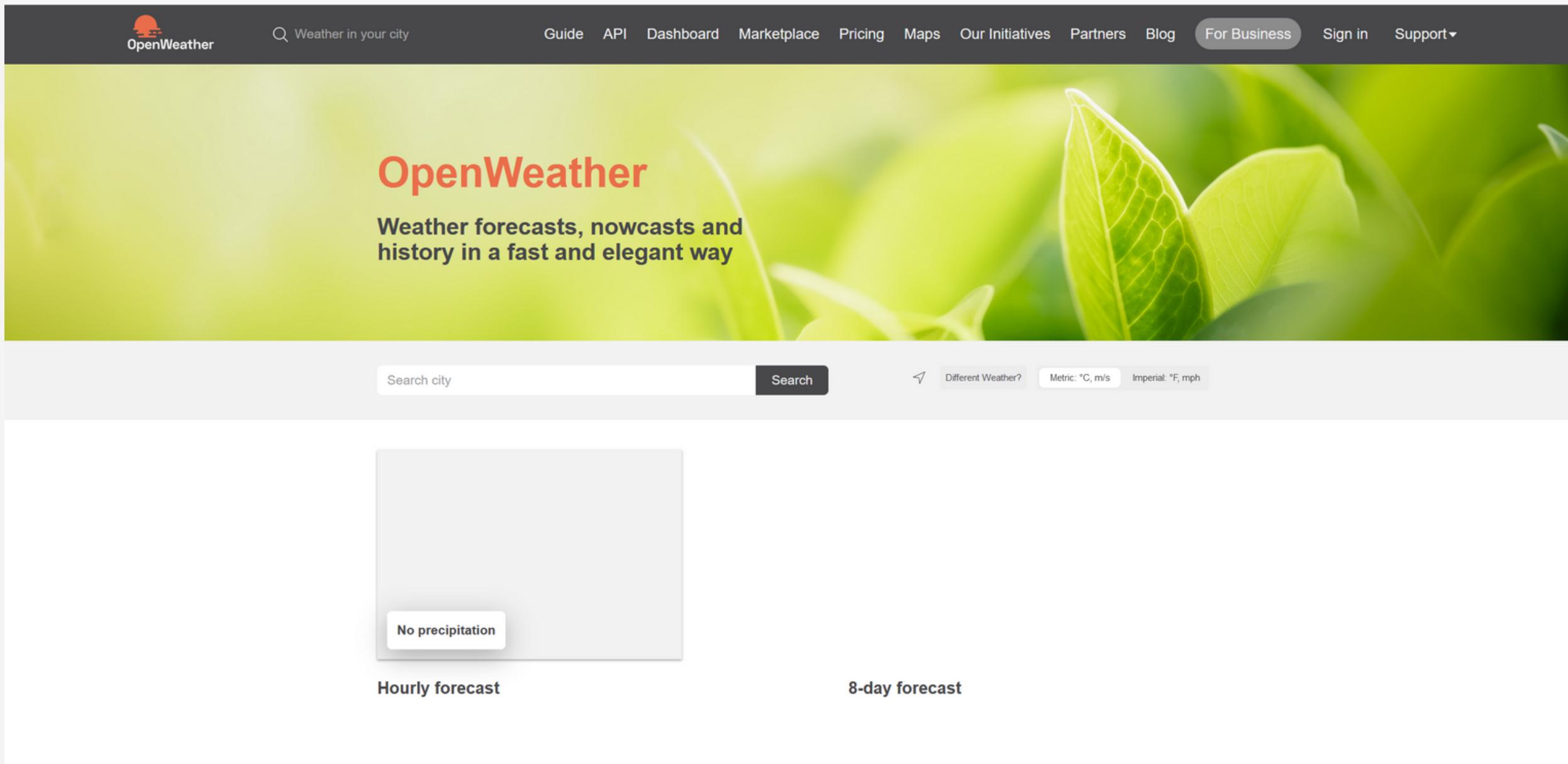
"Master" MKR14

- Legge i valori su MQTT inviati dallo Slave;
 - Sistema di allarme attraverso l'accensione del led rosso e del buzzer;
 - L'allarme si disattiva tramite bottone o web page;
 - Alcuni dati raccolti dall'API vengono stampati sul display.



API - OpenWeather

- API con architettura REST, fornisce informazioni sulle previsioni del meteo;
- Richiesta GET per ottenere le informazioni;
- Le informazioni ottenute arrivano in formato JSON e vengono stampate sul seriale di Arduino;
- Alcune di queste informazioni vengono selezionate dallo Slave e inviate tramite MQTT al Master.



MQTT- CONVERSATION

Topic "Ferrante_Pappagallo/meteo"

Dai dati raccolti dallo Slave, abbiamo selezionato 5 valori: "country", "location", "weather", "temperature" e "humidity". Questi dati vengono spediti sul topic "ferrante_pappagallo/meteo" in MQTT



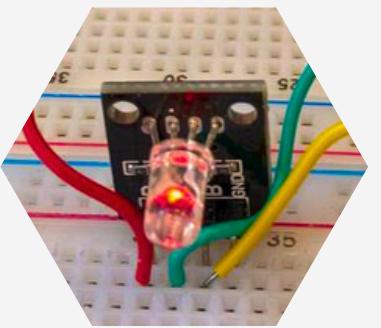
```
Topic: ferrante_pappagallo/meteo  QoS: 0
{
    "country": "IT",
    "location": "Milan",
    "weather": "Clear",
    "temperature": "24.67",
    "humidity": "53"
}
2023-06-16 11:36:42:836
```

QoS 0:

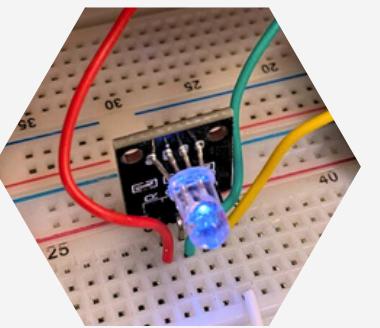
- Non è un problema se qualche messaggio viene perduto;
- Non è necessario utilizzare l'accodamento dei messaggi e le sessioni persistenti (anche se la connessione Wi-Fi non è stabile).

Final remarks

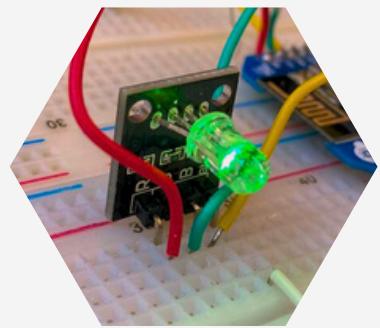
ESP8266



Accendendo il servo con Telegram, si accenderà sull'ESP la luce rossa



Va in modalità Sleep Mode per 15 sec, ogni 30sec.
All'attivazione del Deep Sleep, si accende il led blu



Con l'invio dei dati su MQTT, si accende il led di verde



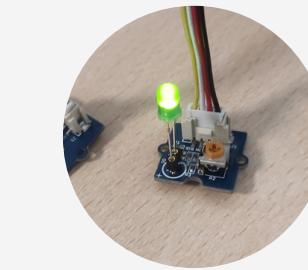
Final remarks

MKR

All'accensione dell'MKR, il display mostrerà il messaggio “*Good morning!*
Please wait” nel mentre che carica il programma.



Successivamente, verranno mostrati in maniera sequenziale i valori del meteo: "Country", "Location", "Weather", "Temperature" e "Humidity" a ripetizione, accompagnati dall'accensione del led verde.

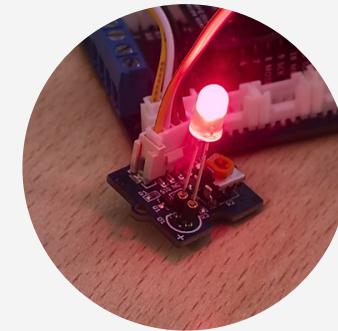


Allarm condition



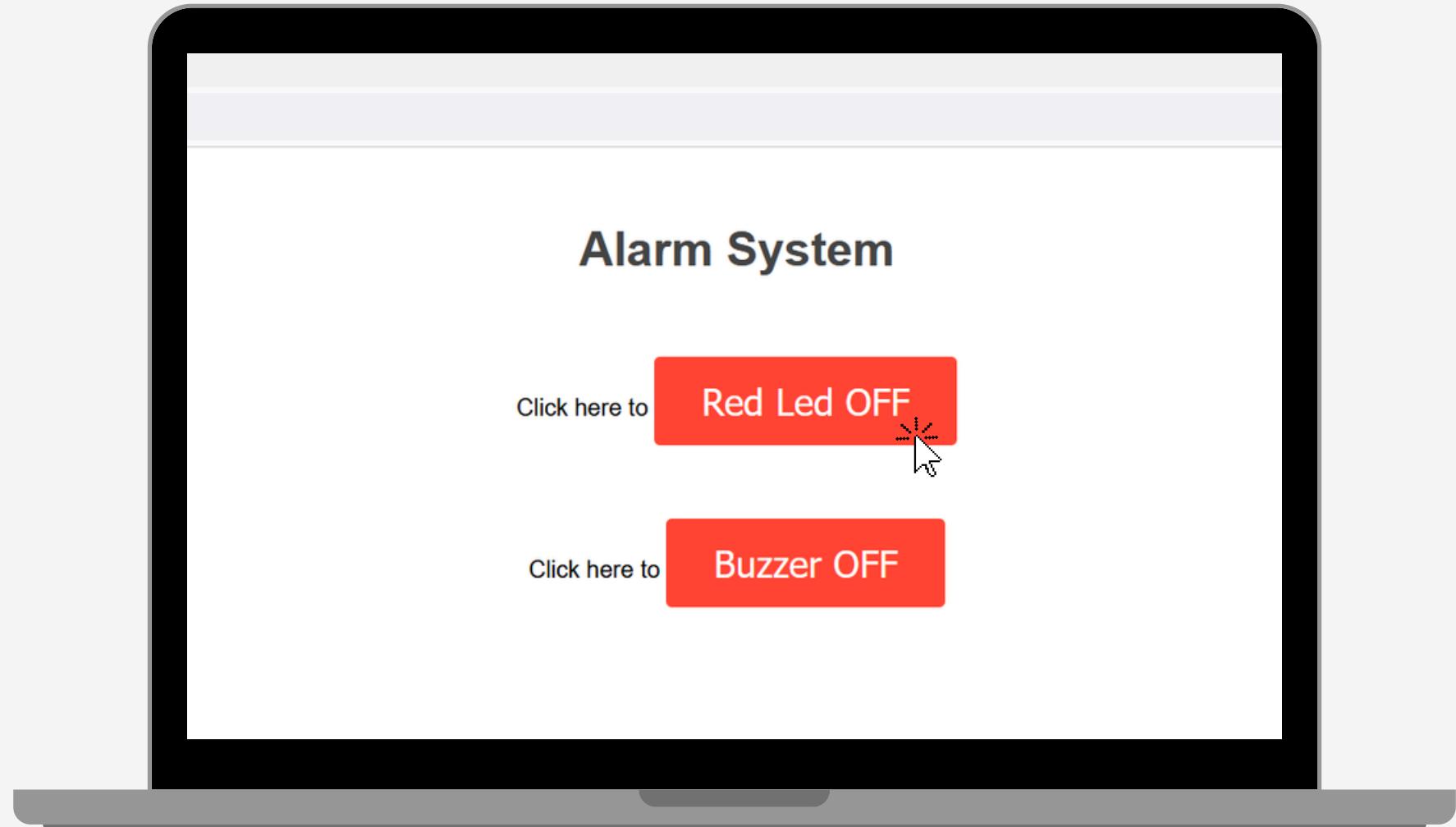
Il led rosso e il buzzer si accenderanno alla presenza di valori che vanno oltre la soglia indicata per la temperatura e l'umidità:

- temperatura $\geq 30^{\circ}\text{C}$ (con relativo messaggio sul display "HIGH TEMP!");
- umidità $< 40\%$ (con relativo messaggio sul display "DRY ENVIRONMENT!").



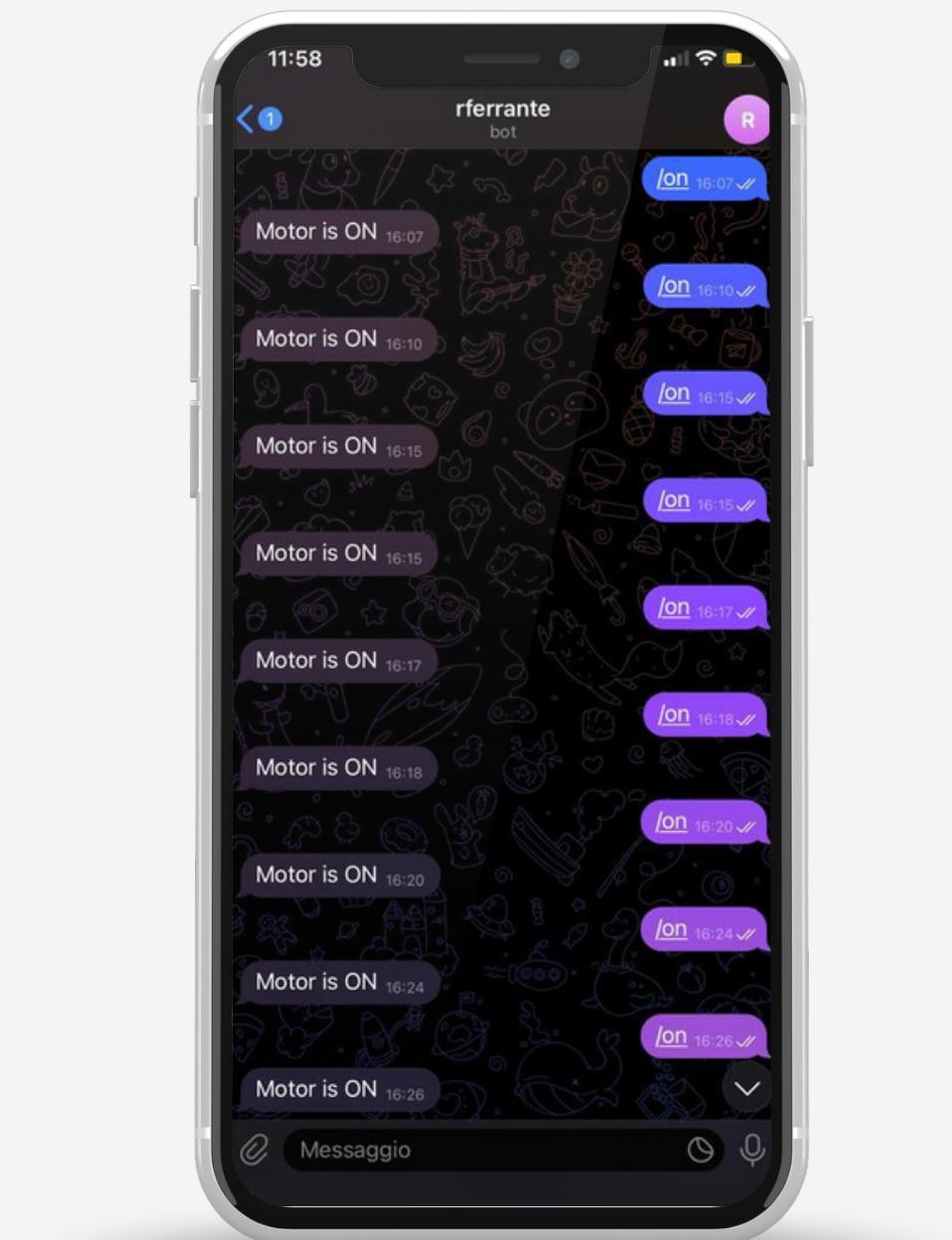
Web Site

L'allarme (led e buzzer) può essere disattivato attraverso il sito web OPPURE attraverso il bottone presente sull'MKR:



Telegram

Attraverso il bot "rferrante" è possibile attivare il servo con un messaggio "/on"



Database

MySQL

	id	datetime	weather	temp	humidity
▶	1	2023-06-08 14:58:53	Clouds	27	46
	2	2023-06-08 14:58:57	Clouds	27	46
	3	2023-06-08 14:59:00	Clouds	27	46
	4	2023-06-08 14:59:05	Clouds	27	46
	5	2023-06-08 14:59:07	Clouds	27	46
	6	2023-06-08 14:59:12	Clouds	27	46
	7	2023-06-08 14:59:15	Clouds	27	46

I dati mandati vengono ogni volta registrati su **MySQL** in una tabella chiamata: *meteo_data*

I valori raccolti riguardano le variabili: "weather", "temp" e "humidity"

All things together

URI or MQTT Topic	Method (GET,POST,PUT,Publish, Subscribe)	Operation	Format of request payload	Return Codes	Format of return payload
mqtt://149.132.178.180:8086/fer rante_pappagallo/meteo	Subscribe	Subscribe to the topic	none	none	{Subscribed to the topic!}
mqtt://149.132.178.180:8086/fer rante_pappagallo/meteo	Publish	Update weather data	JSON {"country": "IT", "location": "Milan", "weather": "Clear", "temperature": "24.67", "humidity": "53"}	none	none
https://149.132.178.180/	GET	Retrieve a html page	none	200 404 not found	{ "new Client" "client disconnected" }
https://openweathermap.org/ current	GET	Retrieve weather data	JSON	none	none



Grazie per l'attenzione!