



STAZIONE DI CONTROLLO

VIGNA 2.5

Artemisia Sarteschi 829677

Gabriele Madotto 829783

CASO D'USO

Il caso d'uso scelto per il nostro progetto è il monitoraggio dello stoccaggio e la produzione di un'azienda vinicola.

Il proprietario dell'azienda può decidere di acquistare due schede, ciascuna per edificio, su cui sono già montati i sensori necessari per l'acquisizione dei dati.

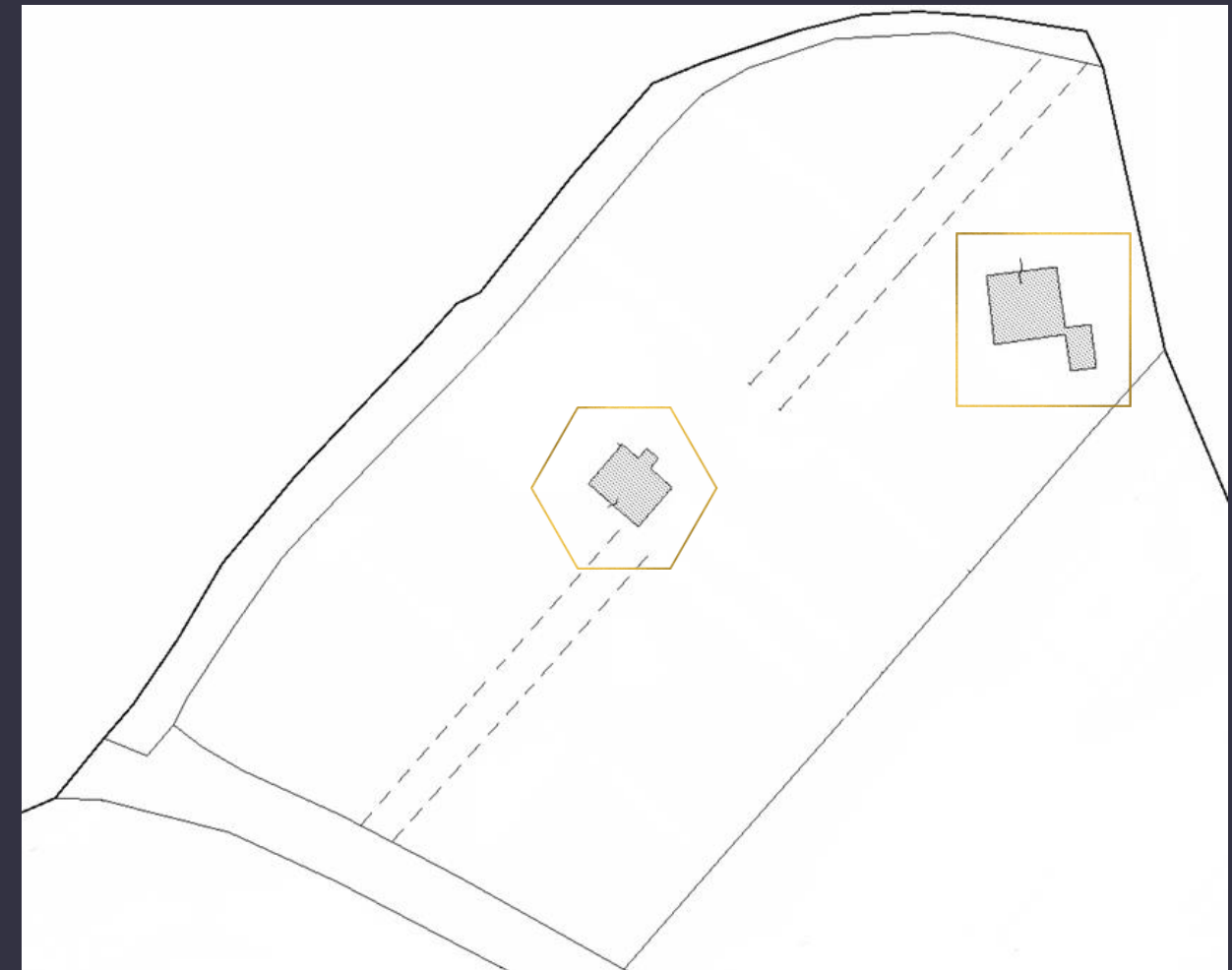
Esse sono gestite attraverso una web app che ne permette il settaggio e la visualizzazione dei dati acquisiti.



CASO D'USO

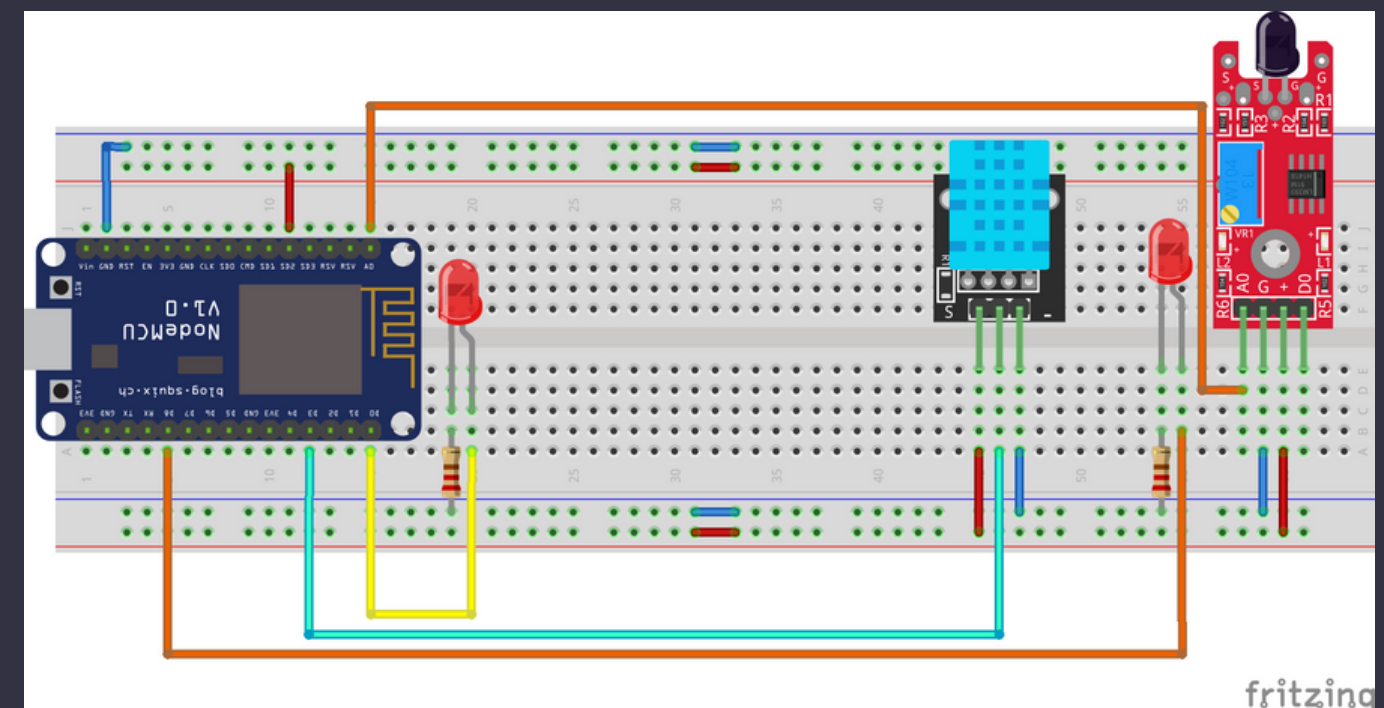
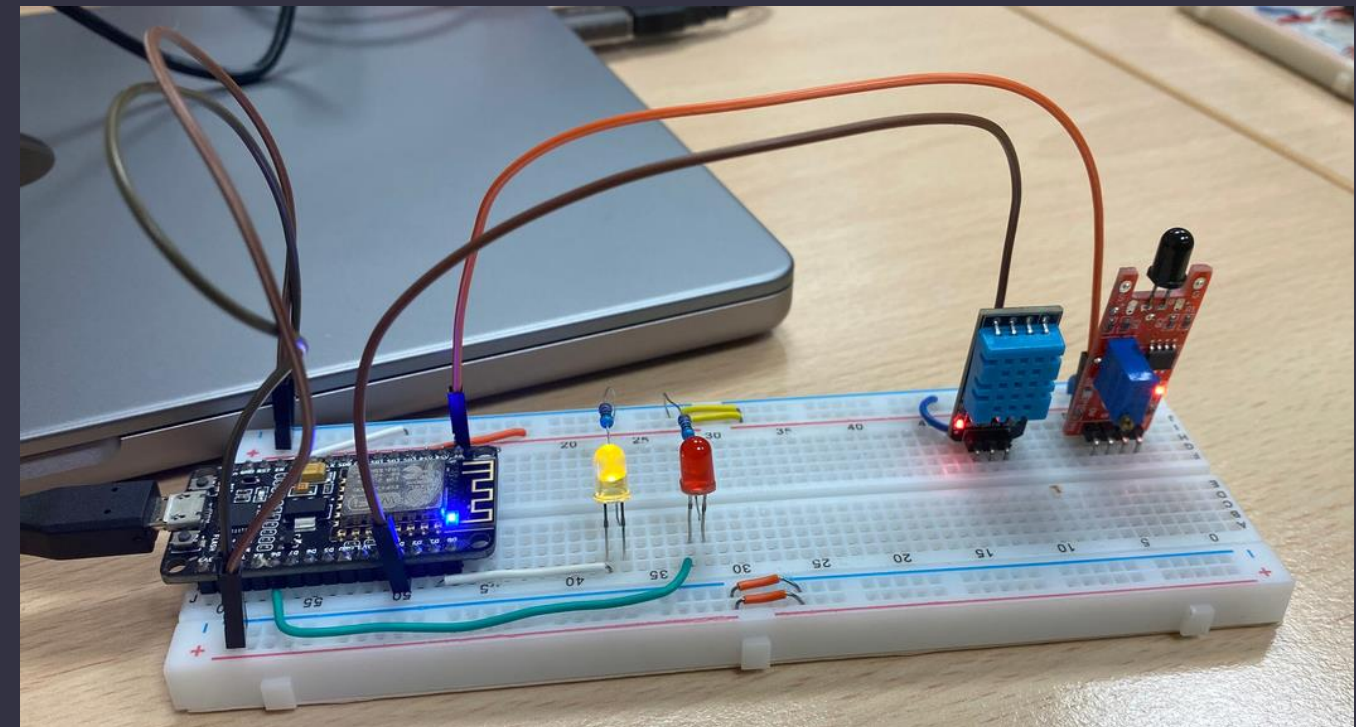
Per la struttura di produzione (NodeMCU ESP8266) viene ritenuto necessario rilevare la temperatura e l'umidità presente per assicurare la sanità del mosto, inoltre è stato inserito un sensore di fiamma per monitorare eventuali incendi.

Per la struttura di stoccaggio (Arduino MKR1000) viene ritenuto necessario installare un sistema di sicurezza, essendo il prodotto conservato in una cantina ad umidità e temperatura stabili. Viene rilevata l'eventuale intrusione nell'edificio (disattivabile) e di fiamma per eventuali incendi.



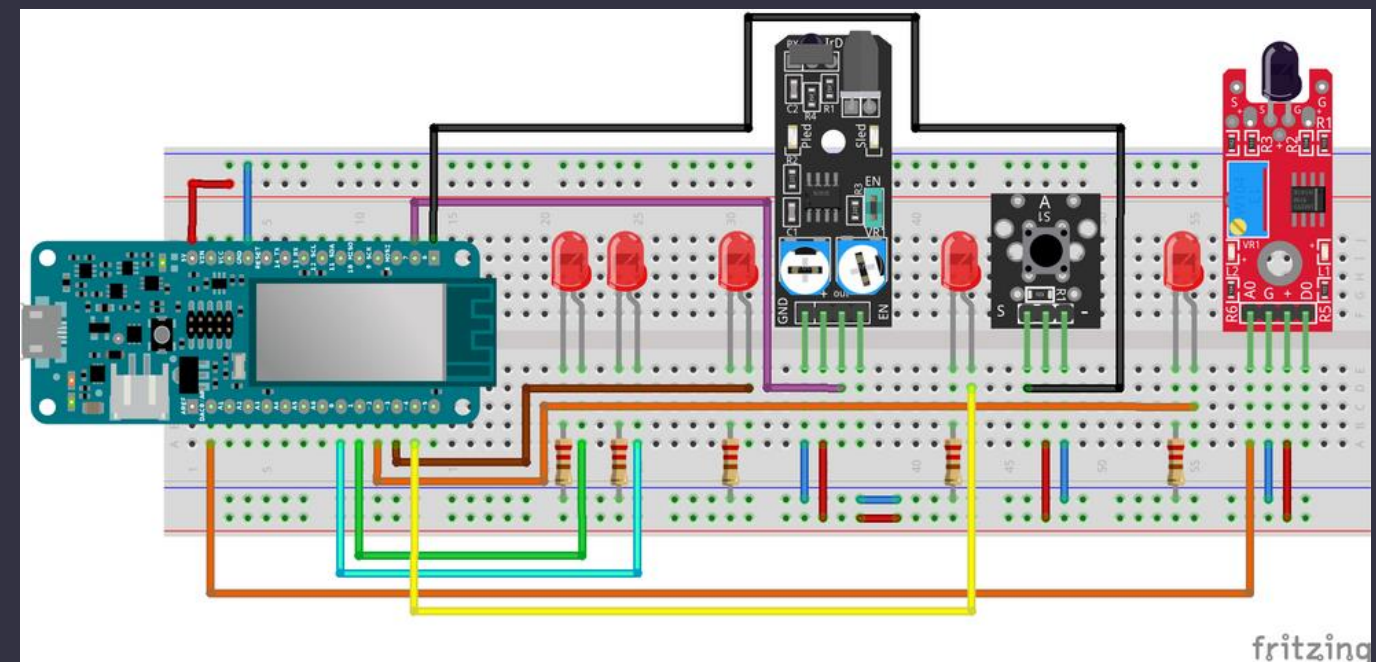
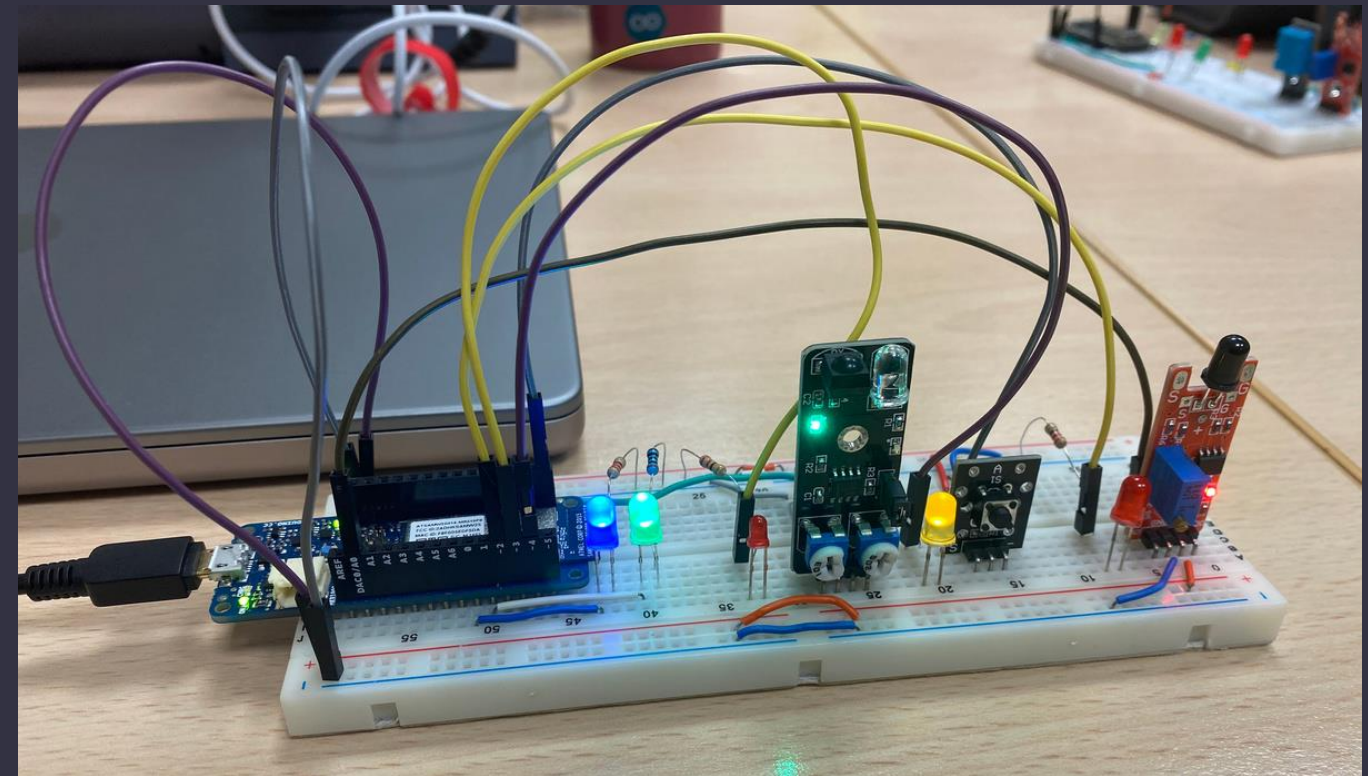
MATERIALI: NODEMCU ESP8266

- Modulo sensore di temperatura e umidità (KY-015)
- Modulo sensore di rilevazione della fiamma (KY-026)
- Led giallo e rosso standard
- Resistenze 220 Ohm [x2]
- Cavi jumper e breadboard



MATERIALI: ARDUINO MKR1000

- Modulo sensore di rilevazione della fiamma (KY-026)
- Modulo sensore evitamento ostacoli (KY-032)
- Modulo bottone (KY-004)
- Led colorati [x4]
- Resistenze 220 Ohm [x4]
- Resistenza 200 Ohm
- Cavi jumper e breadboard



METODO

Le schede vengono collegate all'alimentazione e al loro interno hanno già le informazioni:

- necessarie per raccogliere i dati dai sensori a loro collegati
- le impostazioni per la rete WiFi
- essendo di una stessa casa produttrice conoscono il topic MQTT su cui porsi in ascolto (gmadotto1/general) per aspettare l'ack a scrivere sul topic predisposto (gmadotto1/data).

L'azienda produttrice delle schede mette a disposizione una web app per il settaggio delle schede tramite l'inserimento del loro Mac Address.

Stazione di Controllo Vigna 2.0

MAC address dei nuovi nodi

Nodo produzione

Nodo stoccaggio

Conferma



METODO

Una volta che le schede vengono inserite viene dato loro il permesso di scrivere sul topic dei dati ed i messaggi che vengono inviati vengono parsati dalla web app e inviati al db MySQL.

La stessa fa query al db per aggiornare la pagina e restituire lo stato corrente del sistema ed eventuali allarmi.

La scrittura dei sensori di allarme avviene ad ogni cambiamento di stato invece quelli di rilevazione ogni 10s e il db si svuota ogni 1000 record per non venir intasato.

Stazione di Controllo Vigna 2.0

Incendio nella Produzione

Tranquillo

Allarme Intrusi nello Stoccaggio

Nessun intruso

Incendio nello Stoccaggio

Tranquillo

Sensori Produzione

Temperatura	28 °C	Temperatura alta, ventole attive
Temperatura percepita	29.13 °C	Temperatura alta, ventole attive
Umidità	57 %	Tutto ok



M E T O D O

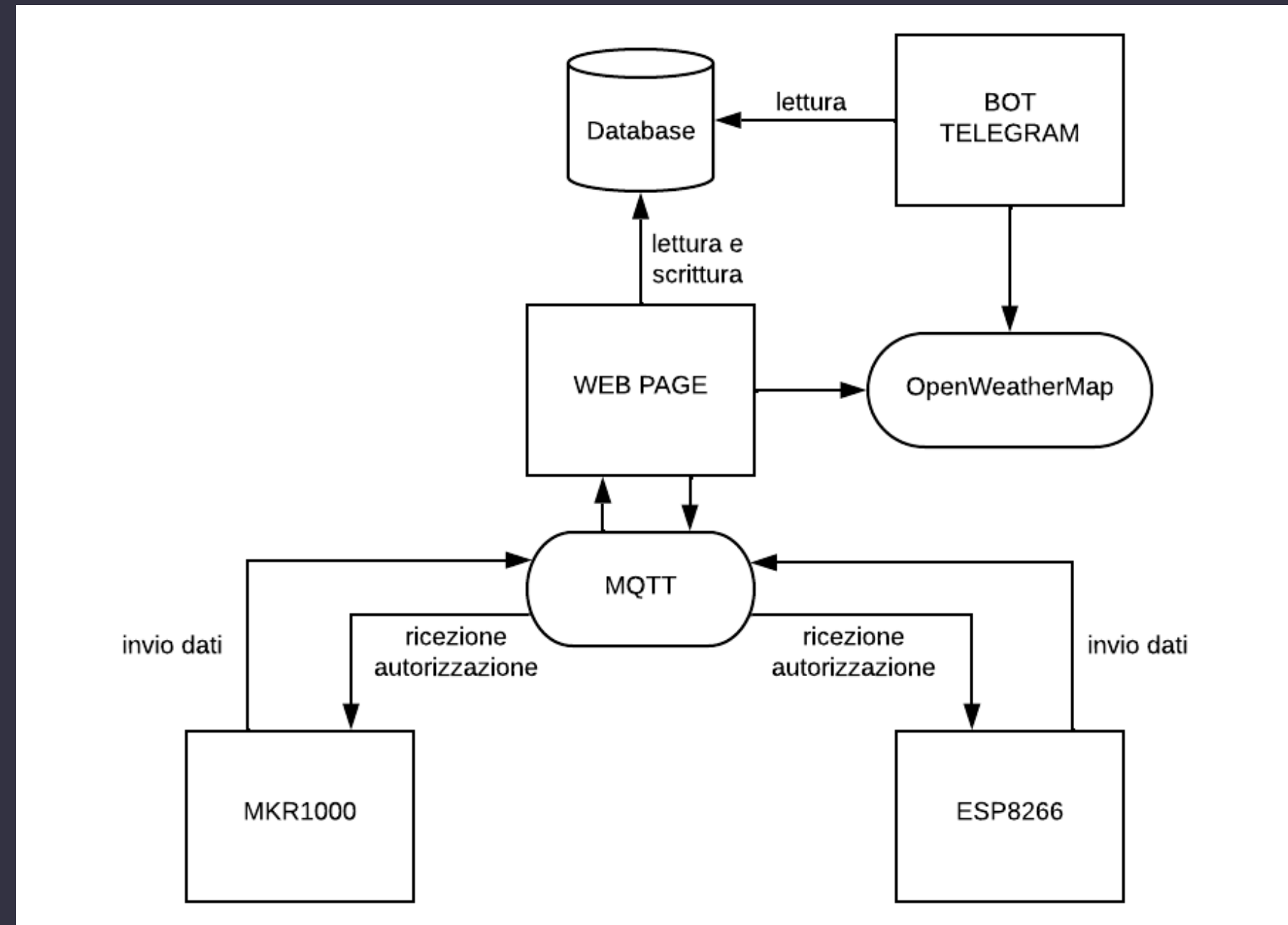
Le soglie dei sensori sono le seguenti:

- Sensori di allarme (booleani)
 - Prossimità
 - Intruso
 - Nessun intruso
 - Incendio
 - Incendio
 - Tranquillo
 - Errore -> Problema al sensore
- Sensori di rilevazione
 - Temperatura
 - $\leq 24^{\circ}\text{C}$ -> tutto ok
 - $> 24^{\circ}\text{C}$ -> Temperatura alta, ventole attive
 - Umidità
 - $\leq 70\%$ -> Tutto ok
 - $> 70\%$ -> Umidità alta, deumidificazione
 - Errore -> Problema al sensore



ARCHITETTURA

Schema ad alto livello in cui sono mostrati i vari componenti del sistema e come essi interagiscono tra di loro.



DATABASE

Il database è composto da tre tabelle:

Production_low(*ID*, TIMESTAMP, TEMPERATURE, REAL_TEMPERATURE, HUMIDITY)

Production_high(*ID*, TIMESTAMP, FLAME, PROXIMITY)

Storage_high(*ID*, TIMESTAMP, FLAME, PROXIMITY)

Con 'high' si intendono i valori dei sensori ad alta priorità, cioè gli allarmi



API (OPENWEATHERMAP)

Il sistema supporta anche una connessione ad un servizio api esterno, in questo caso OpenWeatherMap. Il servizio permette di ottenere informazioni metereologiche attuali in una determinata area geografica.

I dati sono restituiti in formato JSON, da cui è possibile ottenere:

- coordinate (latitudine, longitudine)
- luogo
- temperature (corrente, min, max)
- umidità
- vento
- nuvolosità
- precipitazioni
- ecc...

Per questo progetto si è ritenuto sufficiente mostrare sulla pagina web le informazioni di temperatura, vento e nuvolosità.

Previsioni del tempo a Massa 	
Temperatura Esterna	28.43 °C
Vento	4.92 m/s
Nuvole	poche nuvole

```
https://api.openweathermap.org/data/2.5/  
weather?q={city name},{country  
code}&appid={API key}
```

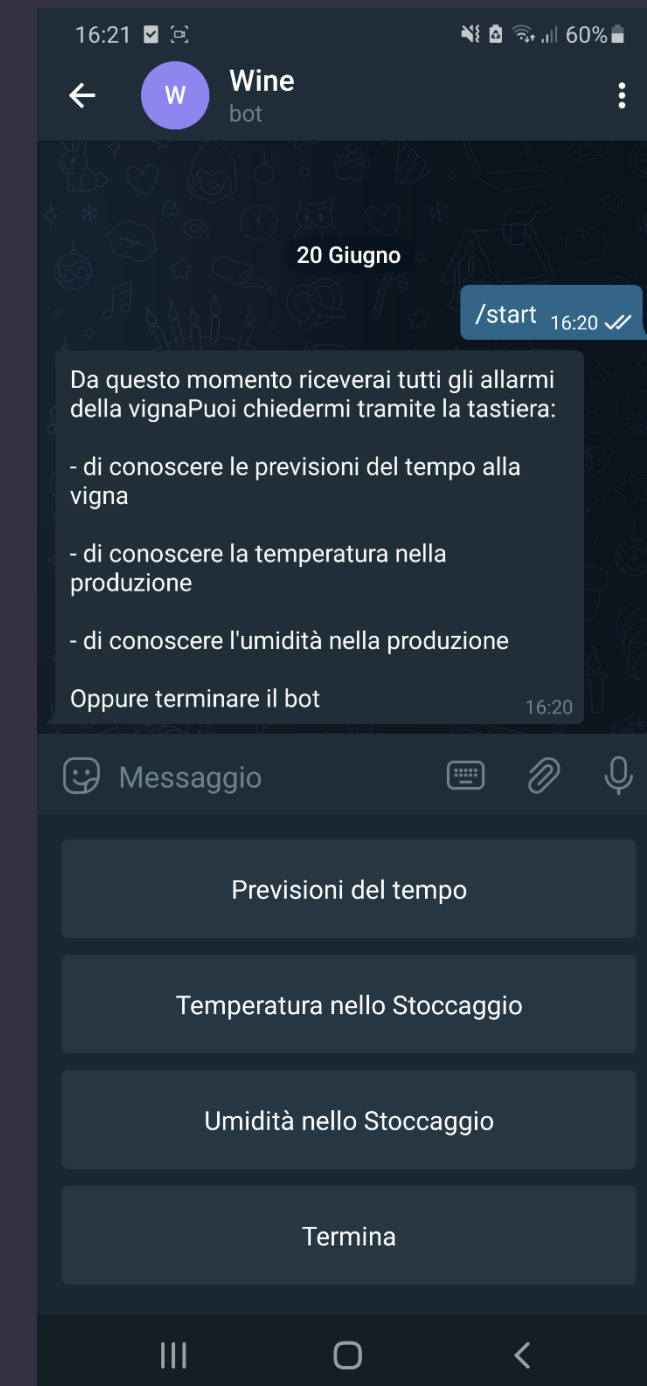


BOT TELEGRAM

Il progetto è stato esteso anche all'uso di un bot telegram

Lo scopo di esso è simile a quello della webapp, ovvero:

- Fornire valori dei sensori di produzione e stoccaggio
 - Fornire condizioni meteo dell'area locale in cui è situata la vigna
 - Mandare notifiche di allarme qualora si verificassero incendi o intrusioni
-
- La differenza tra la webapp e il bot, è che quest'ultimo deve solo leggere valori già presenti del database.
 - Non si occupa nè di aggiungere nuovi nodi al sistema e nè di scrivere record nel database.
 - Il programma del bot è avviato su una istanza separata dalla webapp.



RISPARMIO ENERGETICO

Mantenere una scheda wifi costantemente a piena potenza può risultare oneroso in termini di consumi elettrici, soprattutto se attaccata a una sorgente con capacità limitata (batterie).

Per ovviare a ciò è stata fatta una modifica alla scheda ESP8266, in modo tale che utilizzasse il wifi solo nei momenti di caricamento dei dati dei sensori, oppure qualora la scheda rilevasse un cambiamento di stato per il sensore di fiamma (modalità modem sleep).

Sono stati usati anche dei timer per stabilire il momento di risveglio e spegnimento del wifi.

```
if(check_timer()){  
    update_sensor_values();  
    wake_mcu();  
    publish_sensor_values();  
    send_mcu_to_sleep();  
}
```

```
if(check_timer()){  
    update_high_priority_sensors();  
    if(fire_level != previous_fire_state){  
        previous_fire_state = fire_level;  
        wake_mcu();  
        mqttClient.loop();  
        publish_high_priority_sensor_values();  
        send_mcu_to_sleep();  
    }  
}
```



CRITERI DI COMUNICAZIONE (MQTT)

In seguito sono riportati i topic di mqtt utilizzati per il progetto, con le strutture json associate e i valori che sono passati

TOPIC MQTT	METHOD	OPERATION	JSON STRUCTURE (REQUEST)	JSON STRUCTURE (RESPONSE)
gmadotto1/general	GET	topic per autorizzare i nodi a pubblicare i dati su gmadotto1/data	<pre>{ "id": "invite", "mac": MAC_ADDR }</pre>	<pre>{ "id": "confirm", "mac": MAC_ADDR }</pre>
gmadotto1/data	PUBLISH	pubblica i dati che vengono letti dal nodo main (webpage)	(SLIDE SUCCESSIVA)	NONE



CRITERI DI COMUNICAZIONE (MQTT)

Si è scelto di utilizzare un topic solo per effettuare tutte le pubblicazioni, in modo tale da facilitare una possibile estensione futura con ulteriori schede, e per permettere un'implementazione semplice della logica di controllo dalla parte della pagina web. Cioè un solo canale di ascolto su mqtt, su cui poi sono fatti i controlli di id e priorità per indirizzare i dati nelle apposite tabelle del database

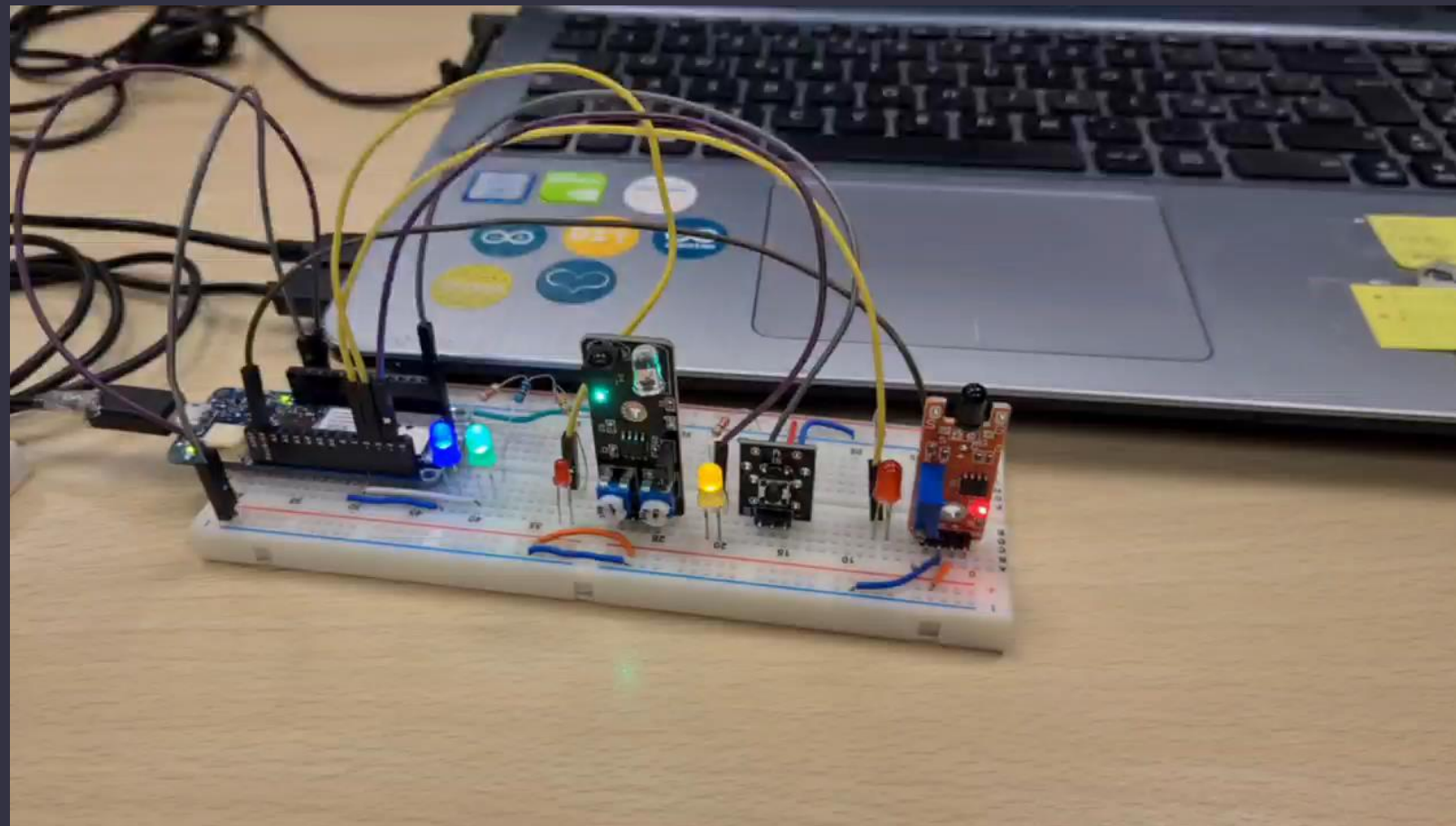
Tutti i topic sono settati con QoS 0, cioè si garantisce di ricevere lo stesso payload al più una volta

EDIFICIO	SENSORI NORMALI	ALLARMI
PRODUZIONE	<pre>{ "id": "production", "priority": "low", "temperature": (float), "humidity": (float) }</pre>	<pre>{ "id": "production", "priority": "high", "fire": (boolean), "proximity": (boolean) }</pre>
STOCCAGGIO	NONE	<pre>{ "id": "production", "priority": "high", "fire": (boolean), "proximity": (boolean) }</pre>

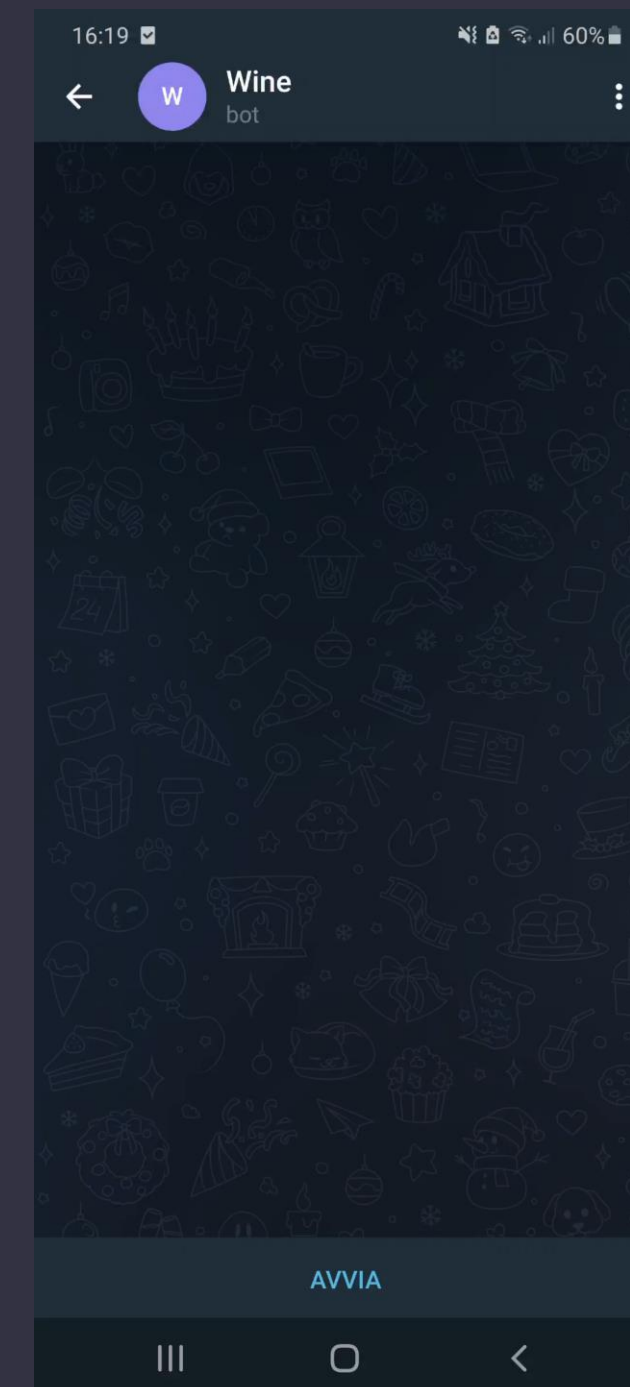
I vari tipi di json passati al topic
gmadotto1/data



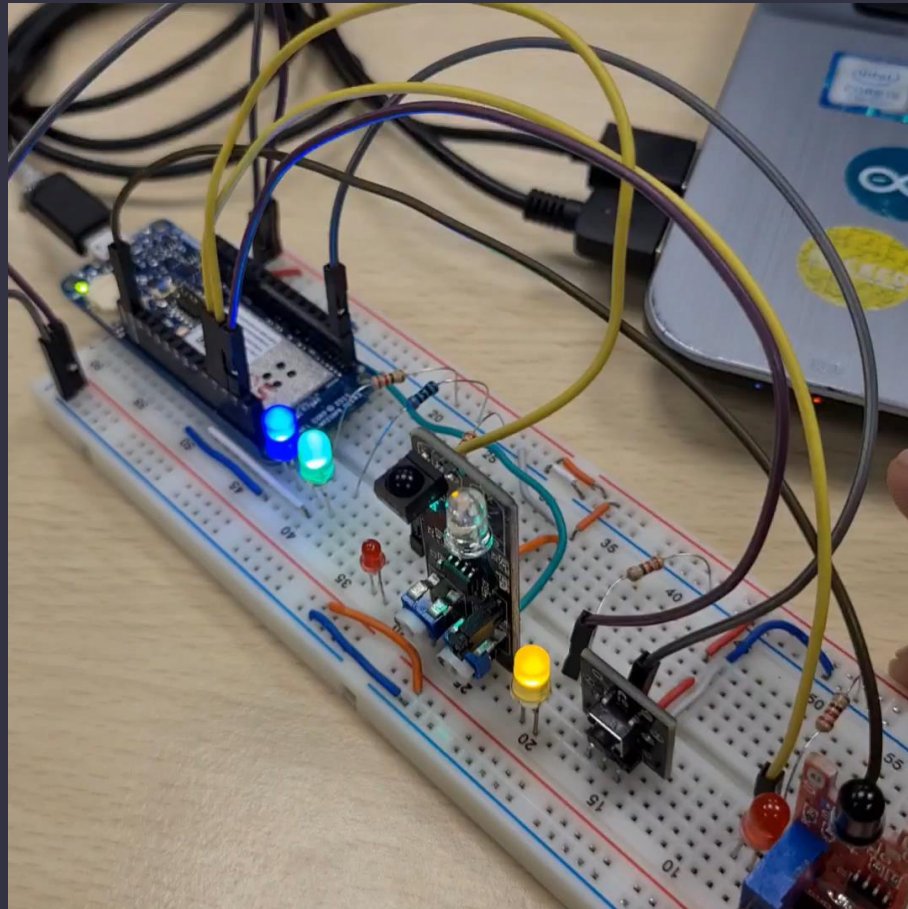
DEMO



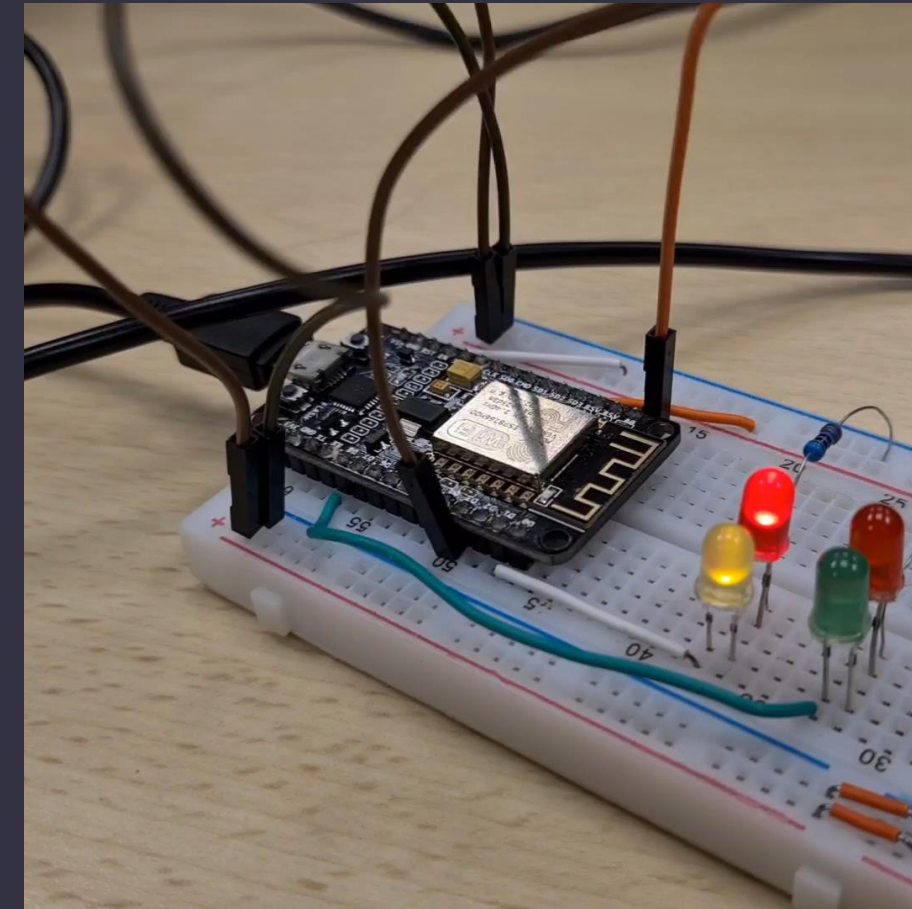
Panoramica del circuito e schede in funzione
A destra una demo del bot



DEMO



simulazione di allarme intrusione



demo aggiornamento valori



CONSIDERAZIONI FINALI

Possibili miglioramenti

- implementare l'auto-refresh della pagina web
- implementare la possibilità che le schede non abbiano già le impostazioni per il WiFi ma che vengano impostate tramite web app o con schermo e tastiera integrati.
- implementare grafici con lo storico dell'andamento di temperatura e umidità
- implementare una pagina in cui l'utente possa scegliere le soglie
- Effettuare un ping periodico da parte della web page per sapere se le schede sono ancora collegate

Limitazioni:

- A causa di compatibilità con librerie, la modalità sleep è disponibile solo per la scheda MCU
- L'api del meteo mostra solo il meteo corrente e su una località hardcoded (MASSA)
- Le notifiche potrebbero non essere immediate (timesleep di 3 secondi sulla web page)

