

SSH PROJECT

CHECK_REMOTE_TEMPERATURE

ANNO SCOLASTICO 2023/24

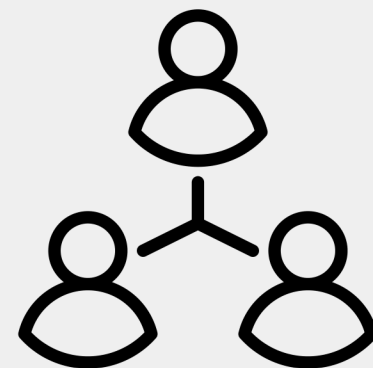
5F - 5H

OUR TEAM



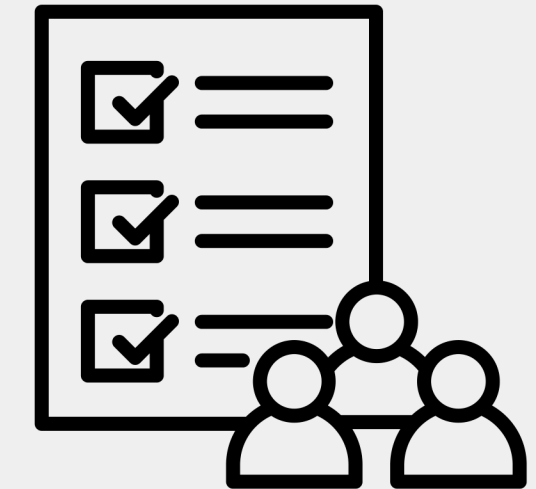
Hardware group

Berni Davide 5F
Scolfaro Alberto 5H
Farina Giuseppe 5H



Software group

Hernandez Rozwel 5F
Qiu Xianjie 5F
Costantini Emanuele 5H
Gasparini Gabriele 5H



Referent

Severi Roberta
Cuoghi Alessio
Beghetti Paolo

OBIETTIVI

monitoring system

creazione di un sistema che consenta l'acquisizione delle temperature dell'ambiente, tramite un sensore BMP280 e l'invio dei dati mediante un microcontrollore ESP32.

back_end

sviluppo di un architettura per la gestione delle trasmissioni dati, tra cui gestione delle informazioni in input dal web site, in output dal broker mqtt e dei dati del database.

front_end

sviluppo di un web site al fine della pubblicazione dei dati delle temperature e delle informazioni riguardanti il progetto

TIMELINE

FEBBRAIO

- sopralluogo alla parrocchia
- acquisto dei materiali
- inizio sviluppo architettura software

- termine sviluppo del software (c++) per il funzionamento dei dispositivi di monitoraggio

MARZO

APRILE

- termine sviluppo architettura software (front-end)
- termine saldatura dei dispositivi di monitoraggio

- termine sviluppo architettura software (back-end)
- collaudo finale
- installazione del sistema di monitoraggio

MAGGIO

SISTEMA DI MONITORAGGIO MQTT

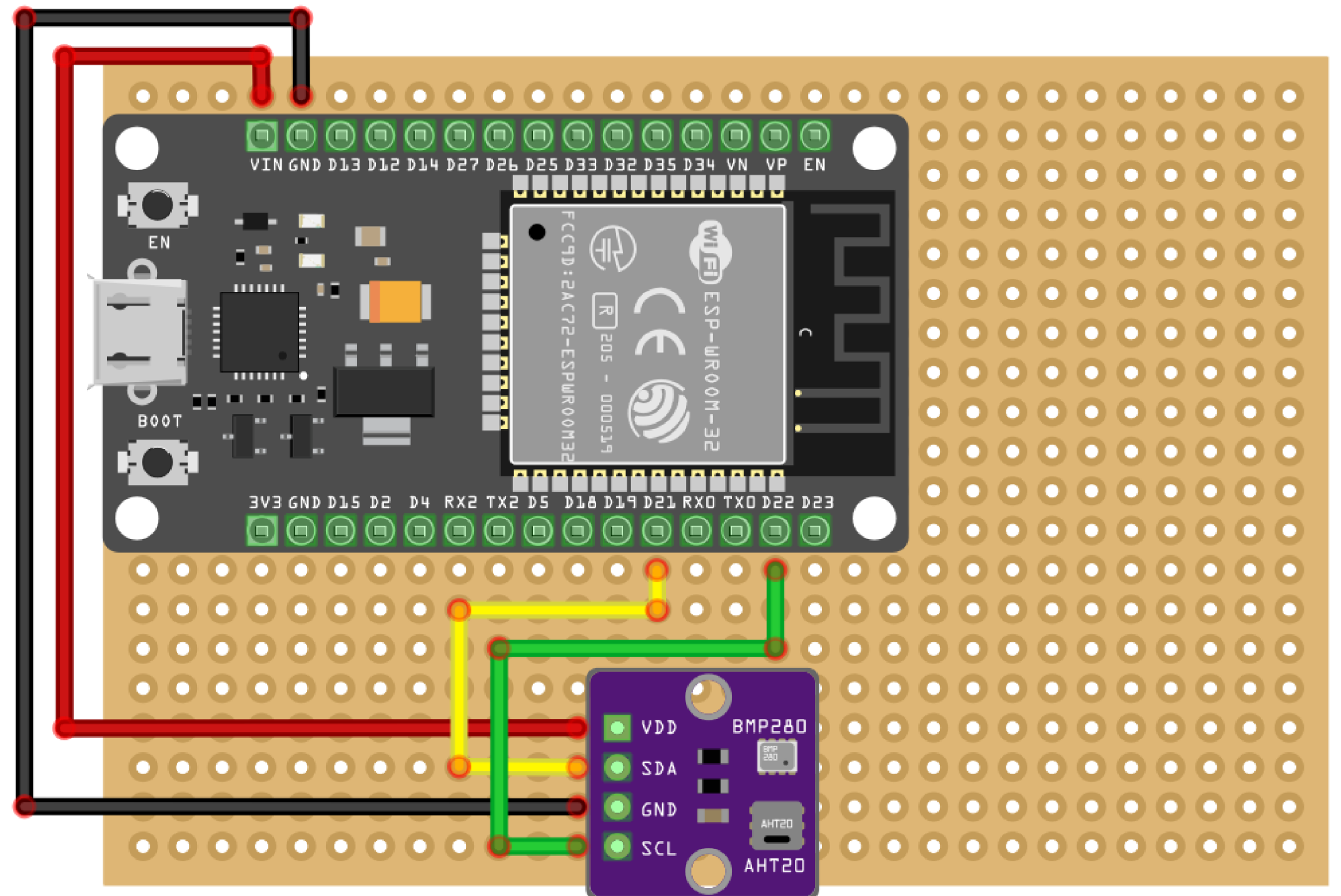
Il sistema di monitoraggio e acquisizione delle temperature è composto da un sensore Bmp280, un microcontrollore esp32 wroom e dei cavi jumper. La comunicazione dati è strutturata tramite protocollo I2C, dove l'esp32 è il master e il bmp280 è lo slave.

I dati acquisiti dal microcontrollore vengono inviati al broker mqtt, tramite la sua conversione in publisher mqtt. L'esp32, una volta collegato alla rete, tramite la connessione WiFi, manderà i dati al Broker sul topic: esp32dbs/temperature.

I dati vengono incapsulati nel topic, mediante il formato json :
`{"MAC":"val_mac","temperatura":"val_temp"}`



CIRCUITO SISTEMA MONITORAGGIO



DJANGO

Django è un framework web gratuito e open source, sviluppato in Python, ideale per la creazione di applicazioni web.

Offre un'interfaccia di amministrazione integrata, che consente agli sviluppatori di interagire con il database ed eseguire operazioni senza scrivere alcun codice SQL.

SQLite è un sistema di gestione di database relazionali (RDBMS) open-source. Gli RDBMS memorizzano i dati in più tabelle invece che in un'unica grande tabella. Ogni tabella è composta da righe che contengono una chiave.



The Django logo features the word "django" in a white, lowercase, sans-serif font, set against a dark green rectangular background.

AJAX

(Asynchronous JavaScript and XML) è una tecnologia web che permette di creare siti e applicazioni interattive e dinamiche.



Asincronia

Invio e ricezione di dati in background senza interrompere la visualizzazione della pagina.

Uso intenso di JavaScript e dell'oggetto XMLHttpRequest per le richieste al server.

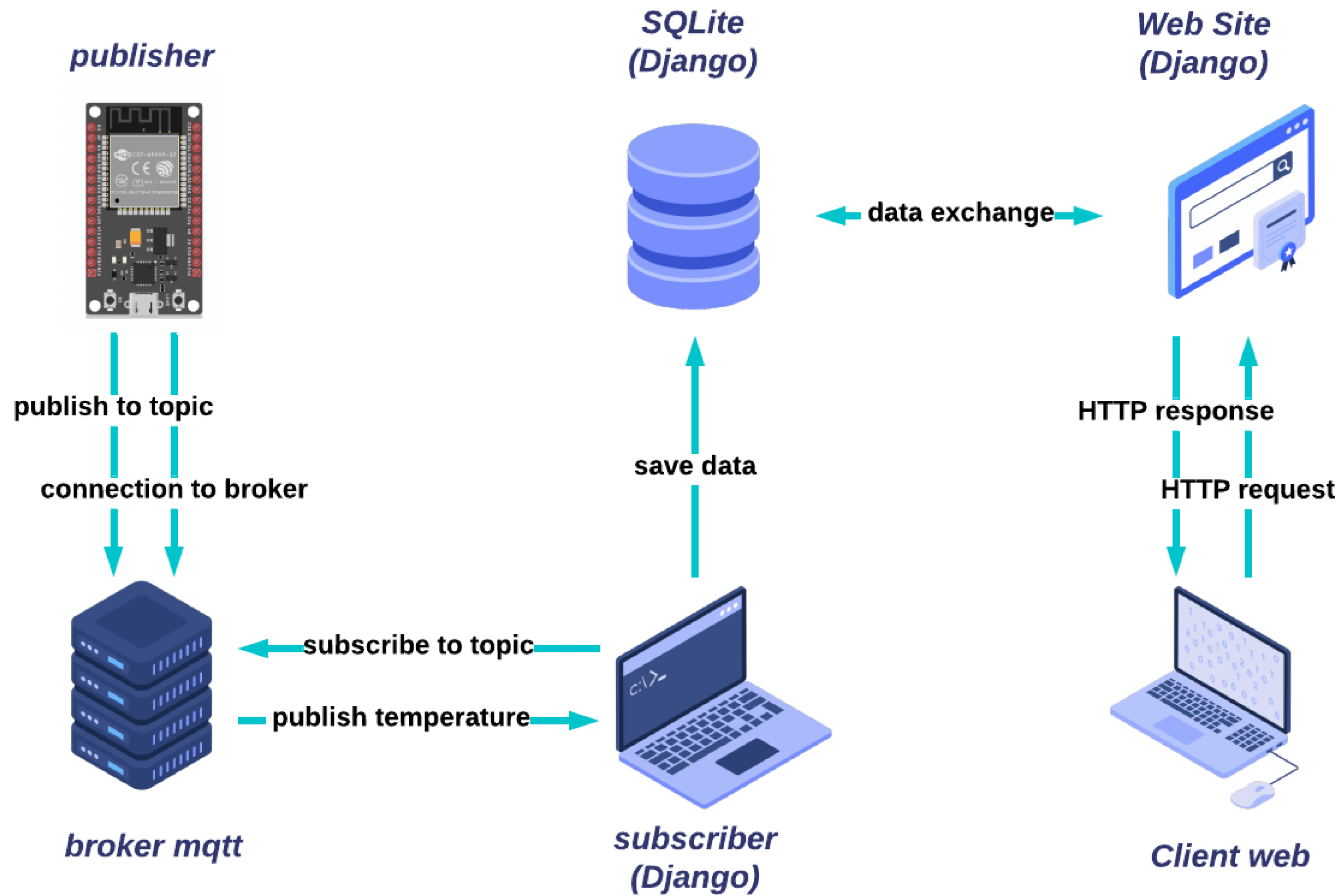
JavaScript

XML e JSON

Originariamente usava XML, ora più comunemente utilizza JSON per lo scambio di dati.

Compatibile con diversi server-side scripts (PHP, ASP.NET, Node.js, ecc.) per dati dinamici.

Interazione con il Server



FLUSSO DATI



RIDIMENSIONAMENTO DEL PROGETTO

RINUNCE

BROKER mqtt

Un broker MQTT è un'applicazione che permette la comunicazione tra i client attraverso il protocollo MQTT1. I client si dividono in due categorie: publisher e subscriber.

Il broker MQTT gestisce il flusso dei dati e permette ai client di iscriversi o pubblicare messaggi. In altre parole, il broker riceve tutti i dati sulle temperature dagli esp32, li elabora e li invia a Django

Un messaggio di alert, o messaggio di allarme, è un segnale o un messaggio di avviso o allarme. Questi messaggi sono spesso utilizzati in vari contesti per informare gli utenti di potenziali problemi o cambiamenti di stato.

Messaggi alert

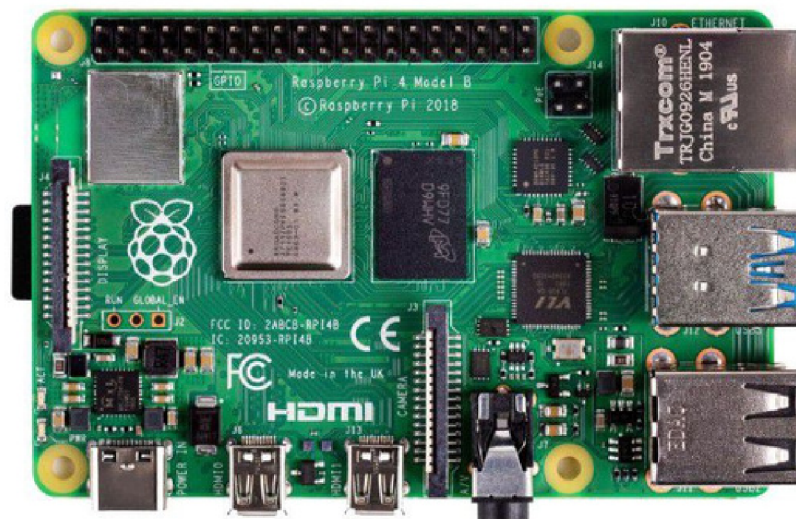
The background features several sets of thin, black, wavy lines that flow across the frame. One set on the left curves from the top left towards the bottom center. Another set on the right flows from the top right towards the bottom right. A third set at the bottom center flows from the left towards the right. These lines create a sense of movement and depth.

POSSIBILI IMPLEMENTAZIONI

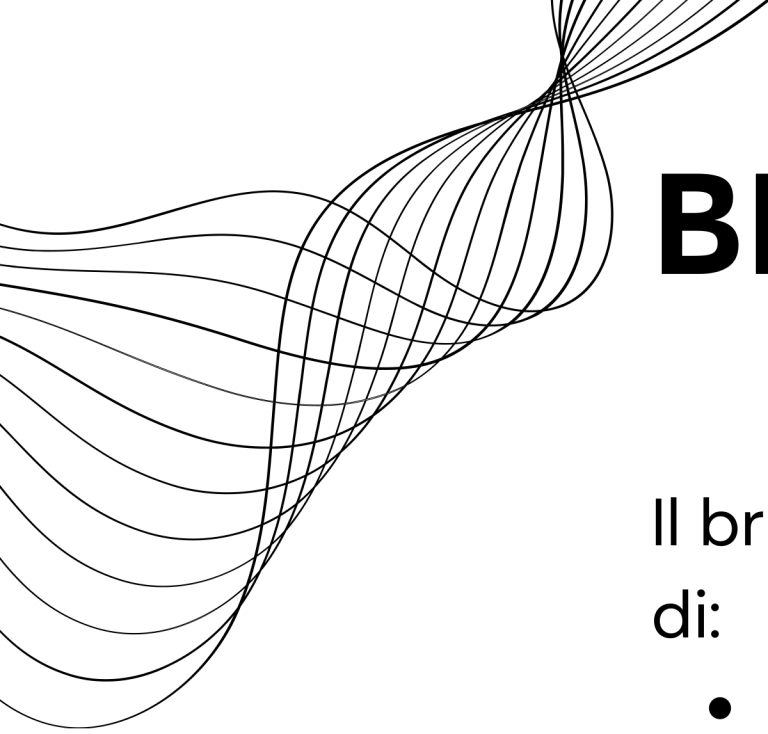
INDIPENDENZA BROKER MQTT

Esistono diversi broker che si possono utilizzare, ma il più versatile è "Mosquitto Broker".

Nei progetti di domotica è possibile utilizzare questo broker, installandolo su un microcontrollore, un end device oppure un server.



Anche se per un progetto IoT, che necessita l'indipendenza della parte hardware è consigliabile l'installazione del broker su una scheda Raspberry Pi.

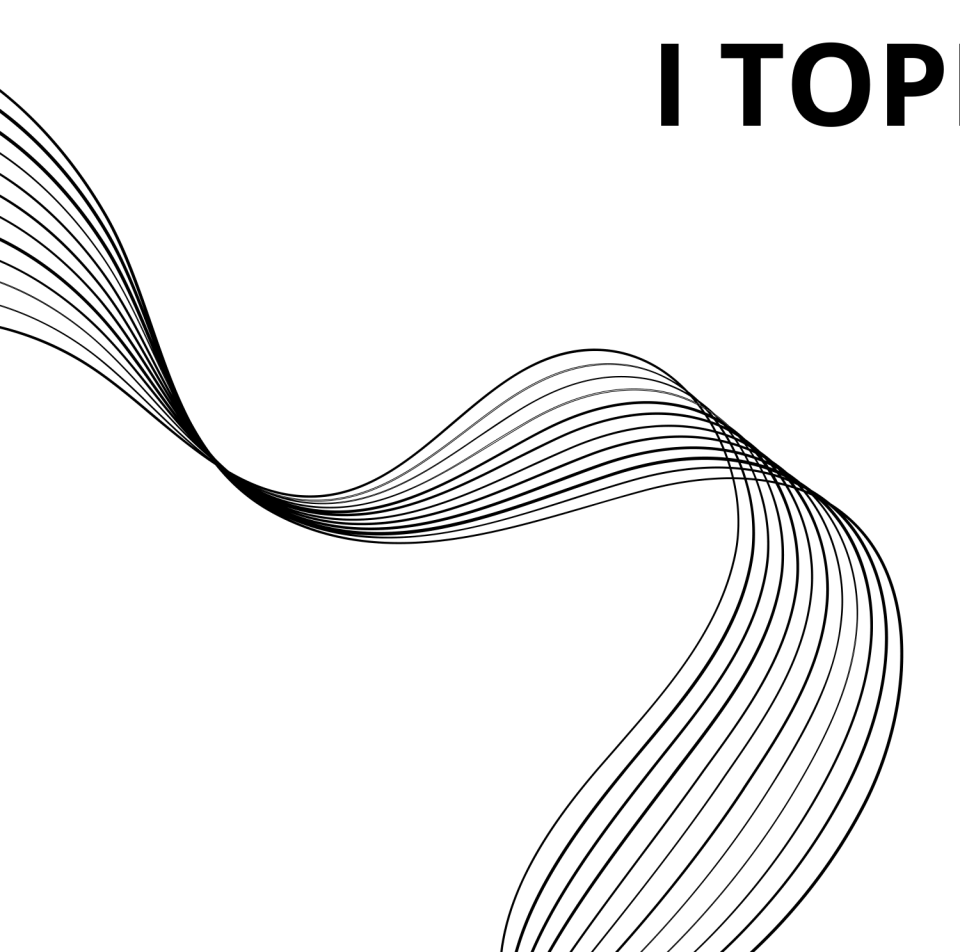


BROKER MQTT: ASPETTI PRINCIPALI

Il broker MQTT è il fulcro del sistema, responsabile di:

- Ricevere i messaggi dai publisher.
- Filtrare i messaggi in base ai topic (argomenti).
- Trasmettere i messaggi ai subscriber appropriati.

OBIETTIVO

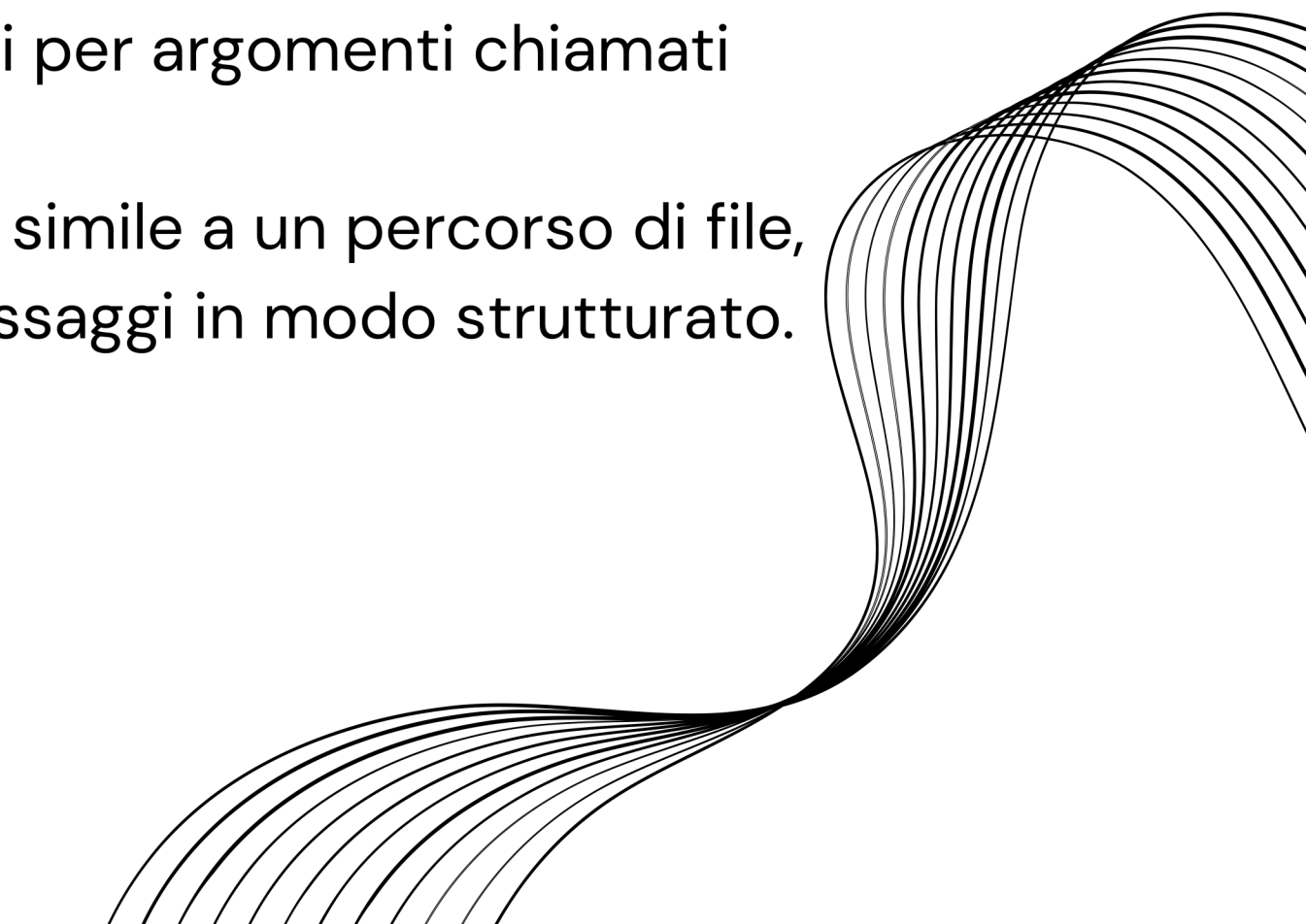


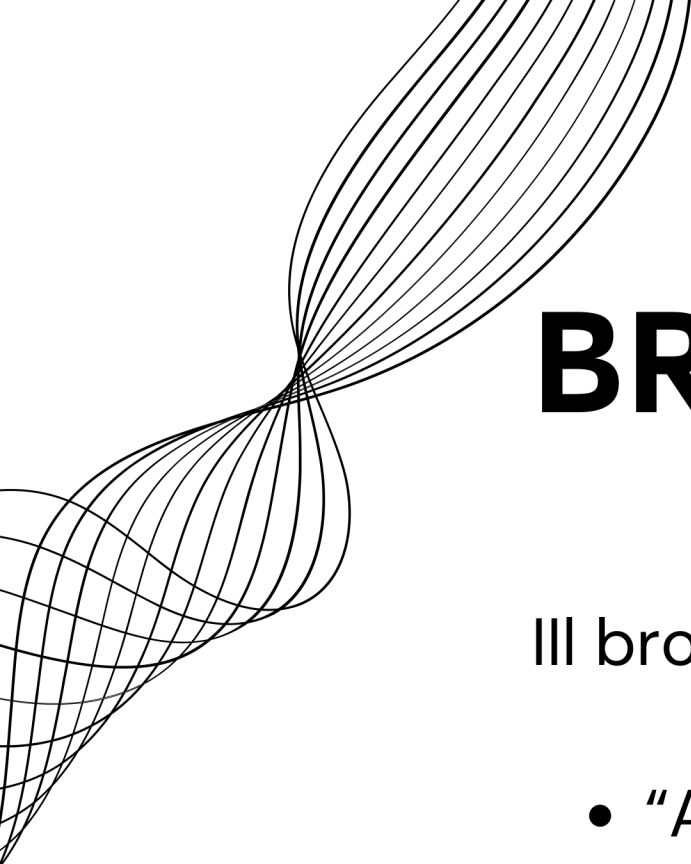
I TOPICS

I messaggi MQTT sono organizzati per argomenti chiamati “topics”.

Un topic è una stringa gerarchica simile a un percorso di file, che consente di organizzare i messaggi in modo strutturato.

Esempio di topic:

- esp32dbb/temperature
- 

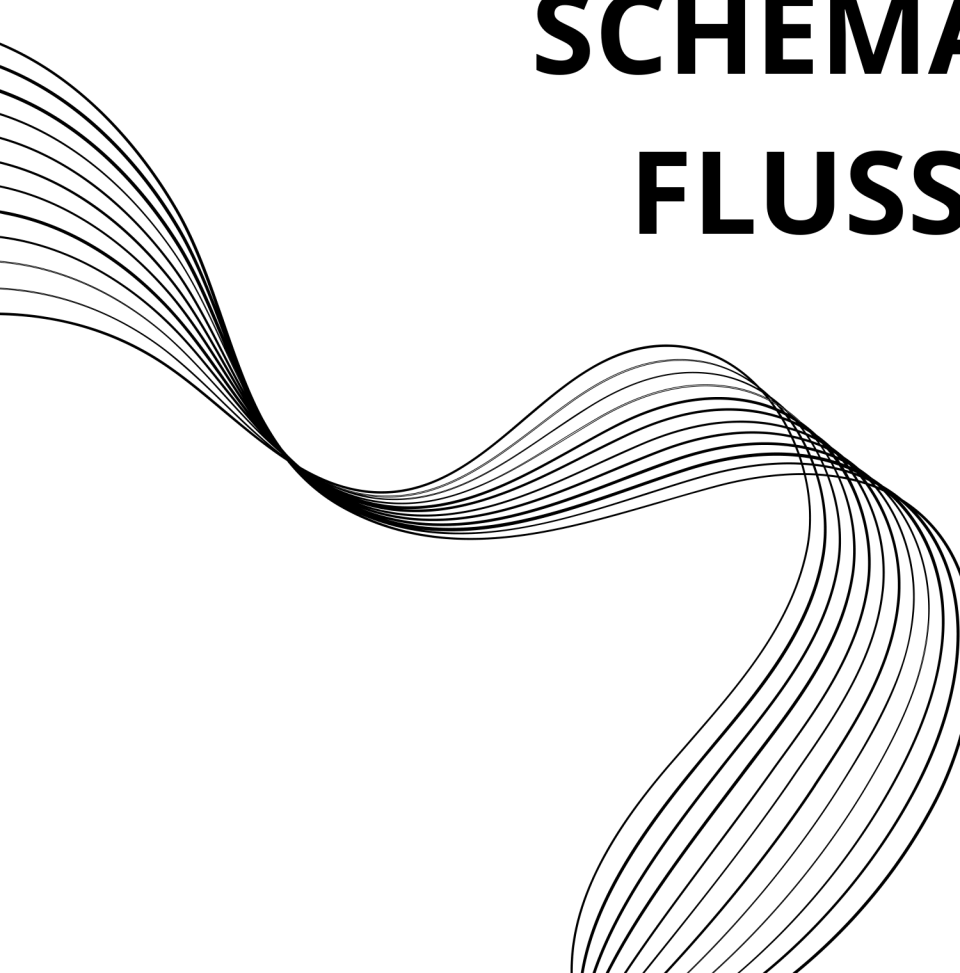


BROKER MQTT: ASPETTI PRINCIPALI

Il broker MQTT può implementare vari livelli di sicurezza, tra cui:

SICUREZZA

- “Autenticazione”: Utilizzando username e password.
- “Crittazione”: Utilizzando TLS/SSL per cifrare i dati in transito.
- “Autorizzazione”: Limitando l'accesso ai topic in base a regole predefinite.



SCHEMA DI FLUSSO

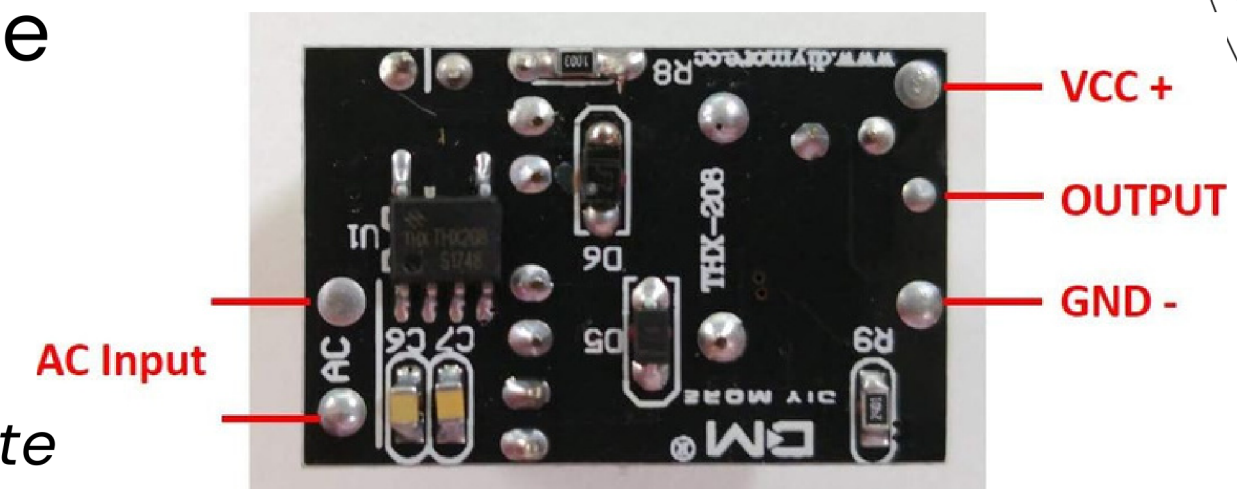
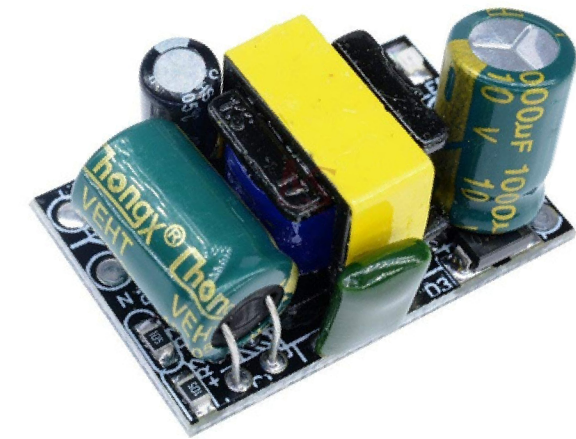
1. “Connessione”: Un client (publisher e/o subscriber) si connette al broker.
 2. “Autenticazione”: Il broker autentica il client.
 3. “Sottoscrizione”: Un subscriber si iscrive a un topic di interesse.
 4. “Pubblicazione”: Un publisher invia un messaggio a un topic.
 5. “Distribuzione”: Il broker riceve il messaggio e lo inoltra a tutti i subscriber iscritti a quel topic.
 6. “Conferma (se richiesto)”: Il subscriber invia una conferma di ricezione se è stato richiesto QoS 1 o QoS 2.
- 

ALIMENTAZIONE SISTEMA DI MONITORAGGIO

metodo 1 - alimentazione cablata

Il circuito dovrà essere integrato nell'impianto elettrico preesistente, dunque si necessita dell'intervento di un tecnico specializzato.

La fase e il neutro dell'impianto elettrico verranno collegati ai pin "AC input" del "convertitore buck", dallo stesso modulo verranno collegati rispettivamente le uscite vcc e gnd agli omonimi pin del esp32.



***** il convertitore buck è un riduttore di tensione, che converte la corrente alternata casalinga 220 volt ad un voltaggio minore (tipo 3,5 - 5 volt) *****

<https://www.amazon.it/pezzi-AC-DC-Power-Supply-Module/dp/B08KH82YK1>

ALIMENTAZIONE SISTEMA DI MONITORAGGIO

metodo 2 - alimentazione a batteria



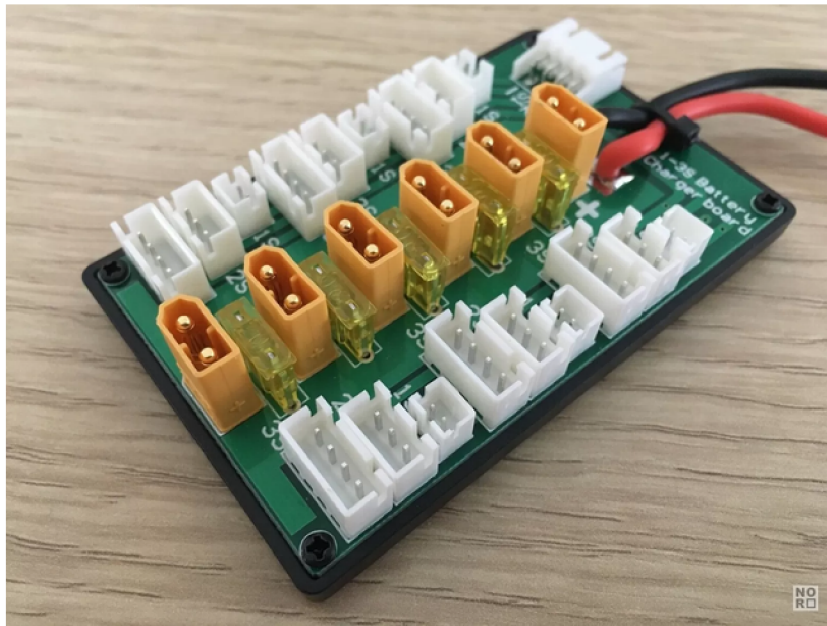
Per garantire la corretta alimentazione del sistema di monitoraggio è necessaria l'implementazione di una batteria (LiPo).

Tale modulo verrà collegato al esp32 wroom, tramite i cavi vcc e gnd, presenti in entrambi.

***** La batteria in questione è EEMB 3.7V 2000mAh 103454, un modulo di alimentazione a base litio, ricaricabile *****

ALIMENTAZIONE SISTEMA DI MONITORAGGIO

metodo 2 - alimentazione a batteria



Abbiamo scelto batterie a Polimeri di Litio (LiPo) perché sono molto efficienti e possono accumulare circa il 20% in più di energia rispetto a quelle a Ioni di Litio: ciò consentirebbe di aumentare l'autonomia e di conseguenza il lasso di tempo tra una ricarica e l'altra.

Per la ricarica di più batterie contemporaneamente è prevista una Scheda di Ricarica Parallela.

***** La scheda di ricarica in questione è la XT30 e permette di ricaricare più batterie in parallelo, risparmiando sui tempi di ricarica *****

A series of overlapping, flowing black lines that create a sense of movement and depth, starting from the bottom left and curving upwards and to the right.

THANKS FOR WATCHING