

PROGETTO SENSORI METEO

Introduzione:

- Titolo: Progetto sensori meteo
- Autori: Giatti Filippo, Malachin Lorenzo, Sescu Denis
- Descrizione generale: Abbiamo una stazione meteo che deve prendere i dati e visualizzarli su un interfaccia.

La stazione meteo è Fengtu FT-WQX10, che deve interfacciarsi con il LoRa Controller e mandare i dati a thingsboard.

Il protocollo di comunicazione utilizzato è il LoRa.

Analisi dei Requisiti

L'obiettivo del progetto è realizzare un sistema backend in grado di ricevere, gestire e salvare i dati meteorologici provenienti da una stazione meteo installata all'interno del laboratorio scolastico. Il sistema è basato su una rete locale realizzata dagli studenti tramite un router configurato nel laboratorio. Il Gateway LoRa e il server ThingsBoard sono collegati a questa rete e comunicano tra loro utilizzando indirizzi IP assegnati dal router.

Il backend deve essere in grado di ricevere correttamente i dati inviati dalla stazione meteo, verificare che siano completi e coerenti e salvarli in modo ordinato. Ogni dato ricevuto deve contenere l'identificativo della stazione, la data e l'ora della misurazione e i valori dei sensori. Inoltre, il server associa a ogni dato anche un orario di ricezione, in modo da mantenere la corretta sequenza temporale delle misurazioni.

Il sistema deve gestire eventuali problemi nei dati ricevuti. Nel caso in cui arrivino dati incompleti o duplicati, questi non devono compromettere il funzionamento del backend. I dati duplicati non vengono salvati due volte, mentre i dati incompleti vengono scartati. Se i valori dei sensori risultano fuori dai limiti normali, i dati vengono comunque salvati ma segnalati come anomali, così da poter essere analizzati in seguito.

Tutti i dati raccolti devono essere salvati sul server locale e resi disponibili sia in tempo reale sia come dati storici. Lo storico delle misurazioni deve permettere di analizzare l'andamento dei parametri meteorologici durante la giornata o in periodi più lunghi. I dati devono rimanere disponibili per tutto il periodo di utilizzo del sistema.

Analisi Funzionale

Analisi Funzionale

Misurazione dati sensore:

Il sensore raccoglie dati in tempo reale e li aggiorna ogni minuto.

Invio dati:

I dati vengono inviati ogni 15 minuti di default, ma l'amministratore è in grado di decidere con quale frequenza inviare i dati.

Gestione dei dati lato backend:

Il backend riceve i dati dal Gateway LoRa tramite la rete locale. Una volta ricevuti, i dati vengono controllati per verificare che siano completi e coerenti prima di essere salvati.

Gestione degli errori:

Se i dati ricevuti risultano incompleti, questi non vengono salvati.

Nel caso in cui arrivino dati duplicati, il backend evita di salvarli più volte.

Se i valori dei sensori risultano fuori scala, i dati vengono comunque salvati ma segnalati come anomali.

In caso di perdita temporanea della connessione, il sistema continua a funzionare senza bloccarsi.

Gestione della temporalità dei dati:

Il sistema distingue tra dati in tempo reale, che rappresentano le ultime misurazioni ricevute, e dati storici, che vengono salvati nel database per analisi successive.

Il timestamp della misurazione viene sempre mantenuto.

Visualizzazione Dati su ThingsBoard:

I dati ricevuti dalla stazione meteo devono essere visualizzati in un'interfaccia web o app, sviluppata da Laghi Davide e Crepaldi Mattia. L'interfaccia deve essere in grado di:

- visualizzare i dati in tempo reale;
- mostrare i dati storici tramite grafici e tavelle;
- consentire l'accesso remoto da diversi dispositivi.

Conservazione dei Dati:

Tutti i dati validi vengono salvati nel Network Server locale.

I dati devono essere disponibili sia in tempo reale sia come storico e consultabili in qualsiasi momento.

Notifiche e Allarmi:

Il sistema è in grado di generare notifiche in caso di valori fuori scala o eventi particolari, come vento forte o pioggia intensa.

Schede Tecniche:

In seguito alleghiamo il link git hub delle schede tecniche:

https://github.com/Denis0160/Progetto_Sensori_Meteo.git

Analisi Tecnica

Il sistema è basato su una rete IoT LoRaWAN che consente di raccogliere e visualizzare in tempo reale i dati meteorologici provenienti da una stazione meteo Fengtu FT-WQX10.

L'architettura è composta da quattro livelli:

Livello di acquisizione dati:

La stazione Fengtu misura parametri come temperatura, umidità, pressione, vento, pioggia e polveri. Comunica via RS485 utilizzando il protocollo Modbus RTU.

Livello di controllo e trasmissione:

Il controller Milesight UC501 converte i dati Modbus in pacchetti LoRaWAN e li invia al gateway.

Livello di rete:

Il gateway Milesight UG6x riceve i pacchetti LoRa e li trasmette tramite la rete locale del laboratorio al server ThingsBoard.

Livello applicativo:

Il server ThingsBoard elabora, salva e visualizza i dati tramite dashboard e grafici accessibili da browser o smartphone.

Flusso dei dati:

- La stazione Fengtu misura i dati ogni minuto.
- Il controller UC501 legge i registri Modbus ogni 5-15 minuti.
- I dati vengono inviati via LoRaWAN (frequenza EU868 MHz) al gateway UG6x.
- Il gateway inoltra i dati al server ThingsBoard tramite la rete locale.
- ThingsBoard archivia i dati e li mostra su grafici e pannelli.

Gestione tecnica del backend:

Il backend è installato su una macchina con sistema operativo Linux collegata alla rete del laboratorio.

Il server ThingsBoard si occupa della ricezione dei dati, della loro validazione di base e del salvataggio nel database locale.

Schema logico dei dati:

Dispositivo: identifica la stazione meteo.

Misurazione: contiene data, ora e valori dei sensori.

Ogni dispositivo può generare più misurazioni nel tempo, che vengono salvate in modo ordinato.

Interfacce di comunicazione:

Comunicazione tra Gateway LoRa e server tramite protocollo MQTT.

I dati vengono trasmessi in formato JSON.

Ogni valore ricevuto viene associato a una metrica all'interno di ThingsBoard.

Affidabilità e continuità del sistema:

I dati vengono salvati su database locale PostgreSQL.

Il sistema continua a funzionare anche in caso di dati errati o temporanei problemi di comunicazione.

I dati storici rimangono disponibili per analisi future.

L'architettura consente l'aggiunta di nuovi dispositivi IoT in futuro.

Parametri tecnici principali:

Parametro	Valore
Frequenza LoRaWAN	868 MHz (EU868)
Intervallo di misura Fengtu	1 minuto
Intervallo di invio UC501	5-15 minuti
Protocollo locale	RS485
Protocollo remoto	LoRa WAN
Portata LoRa	15 km
Alimentazione	Batteria + Pannello solare
Database	PostgreSQL
Interfaccia cloud	Thingsboard,

Configurazione del server:

La configurazione del server avviene su una macchina con sistema operativo Linux, inizialmente configurata in locale.

Viene installato e configurato il server ThingsBoard seguendo la documentazione ufficiale.

Una volta configurato il server, viene aggiunto il Gateway LoRa. Il gateway invia i dati meteo al server ThingsBoard tramite protocollo MQTT.

Dopo la configurazione è possibile visualizzare i dati provenienti dalla stazione meteo tramite la dashboard di ThingsBoard.

Codifica e Decodifica del Payload (Payload Codec)

Il controller Milesight UC501 invia i dati in formato binario compresso (Hexadecimal) per minimizzare il consumo di banda e batteria. Il backend ThingsBoard deve quindi implementare un **Uplink Data Converter** in JavaScript.

- **Processo di Decoding:** Il server riceve un messaggio grezzo (es. 0367fc01). Lo script estrae i byte e li converte in valori decimali leggibili.

- **Esempio di trasformazione:**
- Input Raw: 0x016700E2 (2 byte per il canale, 2 per il valore)
- Output JSON: {"temperature": 22.6}

Architettura del Software e Rule Engine

La logica di gestione descritta nell'analisi funzionale viene implementata su ThingsBoard tramite il **Rule Engine** (Motore di Regole).

- **Validazione e Filtro:** Ogni dato in arrivo passa attraverso un nodo "Script" che verifica se il valore è nel range corretto (es. Umidità tra 0 e 100). Se il dato è fuori scala, viene aggiunto un tag "Anomalo".
- **Gestione Allarmi:** Se la stazione rileva "Vento Forte" (superiore a una soglia impostata), il Rule Engine genera automaticamente un evento di allarme che compare sulla dashboard e invia una notifica.
- **Storage:** Solo dopo aver superato i controlli di integrità, il dato viene scritto nel database **PostgreSQL**.

Politiche di validazione del dato

Caso	Comportamento
Dato incompleto	Scartato
Dato duplicato (stesso device + timestamp)	Non salvato
Dato fuori range	Salvato + flag "anomalo"
Dato in ritardo (>10 min)	Salvato + flag "late"