SmartTank

Sofia Manno 1000067618

Andrea Anastasio 1000067579

[Introduzione 2](#_Toc193097296)

[Architettura 2](#_Toc193097297)

[Dispositivo Principale 2](#_Toc193097298)

[Thingsboard 3](#_Toc193097299)

[Applicazione Python 4](#_Toc193097300)

[Realizzazione del Bot 5](#_Toc193097301)

[Funzionamento e Risultati 6](#_Toc193097302)

[RISULTATI 7](#_Toc193097303)

# Introduzione

**SmartTank** è un sistema innovativo progettato per il monitoraggio intelligente del livello d’acqua in una cisterna. L'obiettivo è fornire agli utenti dati precisi sulla quantità d’acqua disponibile, misurata in percentuale, per una gestione più efficiente delle risorse idriche, particolarmente utile in contesti caratterizzati da carenze d’acqua.

Oltre al monitoraggio del livello, **SmartTank** è in grado di stimare con precisione il tempo rimanente prima dello svuotamento della cisterna, analizzando i consumi e fornendo previsioni tempestive.

Le informazioni raccolte vengono trasmesse in tempo reale a una piattaforma IoT (**ThingsBoard**) tramite connessione WiFi, consentendo agli utenti di visualizzare i dati attraverso grafici interattivi e uno storico dettagliato.

Un applicazione Python interroga periodicamente ThingsBoard per acquisire i dati sul livello dell’acqua, li analizza per stimare il tempo di svuotamento e, se necessario, invia notifiche di avviso tramite un bot Telegram.

Infatti, per garantire un controllo ancora più efficace, il sistema include un **servizio di notifiche automatiche**: un bot Telegram avvisa l’utente quando il livello dell’acqua scende sotto il **20%**, indicando il valore rilevato. Questa funzionalità consente un intervento tempestivo, evitando sprechi o situazioni di emergenza.

L’idea nasce dall’esperienza diretta vissuta nell’entroterra siciliano, dove la distribuzione idrica avviene con cadenza limitata, rendendo fondamentale una gestione consapevole delle risorse disponibili.

# Architettura

## Dispositivo Principale

Il dispositivo principale è quello che ci permette di raccogliere i dati.

È stato realizzato attraverso i seguenti componenti:

* Arduino MK1000 Wi-Fi
* Grove Ultrasonic Ranger v2.0
* Interruttore digitale con galleggiante

Immagine che contiene cavo, Impianto elettrico, elettronica, Ingegneria elettronica

Il contenuto generato dall'IA potrebbe non essere corretto.

Il dispositivo ha il compito di rilevare i dati dai sensori e trasmetterli a **ThingsBoard** tramite la connessione WiFi, garantendo un monitoraggio costante e in tempo reale.

Abbiamo valutato diverse soluzioni per il monitoraggio del livello dell'acqua nella cisterna, tra cui il **Continuous Fluid Level Sensor**. Questo tipo di sensore funziona misurando la resistenza elettrica tra due elettrodi immersi nel liquido, fornendo un'indicazione continua del livello dell'acqua. Tuttavia, nonostante le buone prestazioni di questo sensore, la sua principale limitazione è la scarsa **scalabilità**. Infatti, per gestire un ampio intervallo di misurazioni o per integrarlo facilmente in un sistema complesso, sarebbe necessario un numero maggiore di sensori o un sistema più complesso.

Abbiamo quindi scelto di utilizzare il **Grove Ultrasonic Ranger v2.0**. Sebbene i sensori a ultrasuoni non siano comunemente impiegati per misurare il livello dell’acqua, abbiamo deciso di testarli per verificarne l’efficacia in questo contesto.

I sensori a ultrasuoni funzionano emettendo onde sonore a **bassa frequenza** che rimbalzano sulla superficie dell'acqua e ritornano al sensore. La distanza viene calcolata in base al tempo di ritorno dell'onda. Nonostante il sensore fosse inizialmente pensato per rilevare ostacoli a distanza, abbiamo scoperto che funziona abbastanza bene per misurare la superficie dell'acqua, grazie alla sua capacità di operare con onde a bassa frequenza, che sono meno suscettibili alla riflessione da superfici irregolari come quelle dell’acqua.

Per ridurre il rischio di malfunzionamenti dovuti a **basse distanze** o a condizioni in cui le onde ultrasoniche potrebbero non essere riflettute in modo ottimale, abbiamo deciso di integrare un **interruttore con galleggiante**. Questo sensore aggiuntivo funge da **doppio controllo**, garantendo un rilevamento più affidabile del livello dell’acqua, in particolare quando il sensore a ultrasuoni potrebbe essere meno preciso. In questo modo, il sistema è più robusto e può compensare eventuali imprecisioni o errori di lettura del sensore a ultrasuoni.

Infine, Arduino MK1000 WiFi dialoga con l’esterno proprio utilizzando la sua interfaccia di rete WiFi e il protocollo HTTP, invia un JSON ogni secondo contente tutti i dati raccolti alla telemetria di thingsboard.

## Thingsboard

Per adoperare thingsboard abbiamo deciso di usare la versione Demo online.

Arduino misura il livello dell’acqua nella cisterna, elabora il dato in percentuale e lo trasmette all’endpoint fornito da ThingsBoard tramite richieste HTTP POST, dove viene memorizzato ed elaborato per il monitoraggio e l’analisi.

ThingsBoard consente di salvare i dati come **telemetria**, permettendo di creare uno storico dettagliato e di monitorare costantemente ogni variazione del livello d’acqua nella cisterna, visualizzabile tramite **dashboard personalizzate**.

La nostra applicazione Python interroga periodicamente l’endpoint di ThingsBoard per recuperare i dati inviati da Arduino. In particolare, elabora previsioni sul tempo di svuotamento della cisterne e invia notifiche di avviso quando il livello dell’acqua scende sotto una soglia critica.

Immagine che contiene Blu cobalto, Blu elettrico, testo, blu

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 1. Rappresentazione grafica del livello d’acqua nella cisterna: il sistema indica un riempimento del 90%.

Immagine che contiene linea, Diagramma, diagramma, Parallelo

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 2. Grafico dell’andamento del livello dell’acqua nella cisterna durante la giornata. La variazione indica i momenti di riempimento e consumo.

# Applicazione Python

Per aumentare la flessibilità e le capacità di analisi, un’applicazione Python interroga periodicamente ThingsBoard tramite richieste HTTP per ottenere gli ultimi valori di livello della cisterna.

Vengono svolte le seguenti operazioni:

1. **Recupero dati**: Interroga l’API di ThingsBoard per ottenere i dati più recenti sul livello dell’acqua.
2. **Visualizzazione in tempo reale**: Mostra il livello della cisterna e l’orario dell’ultimo aggiornamento tramite un’interfaccia grafica.
3. **Archiviazione dati**: Mantiene uno storico dei dati raccolti per analisi e previsioni future.
4. **Stima del tempo di svuotamento**:

La stima del tempo di svuotamento viene calcolata applicando una regressione lineare ai dati raccolti. In pratica, il sistema analizza l’andamento del livello dell’acqua nel tempo e ne associa un’equazione lineare che descrive questa tendenza.

Il coefficiente angolare dell’equazione rappresenta la velocità di svuotamento della cisterna, ovvero la variazione del livello dell’acqua nell’unità di tempo.

Utilizzando questa relazione, il sistema calcola il tempo stimato affinché il livello dell’acqua raggiunga lo zero. La stima viene aggiornata dinamicamente ad ogni nuovo dato ricevuto, garantendo previsioni sempre più precise man mano che si accumulano informazioni.

1. **Notifiche Telegram**:
   * Un bot Telegram avvisa l’utente quando il livello dell’acqua scende sotto il 20%, includendo il valore attuale rilevato.
   * Gli avvisi vengono inviati con un intervallo minimo di 60 secondi per evitare notifiche ripetute in breve tempo.

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

Figure 3. Interfaccia di monitoraggio del livello del serbatoio con aggiornamenti in tempo reale e avvisi tramite Telegram

### Realizzazione del Bot

Per la creazione del bot su Telegram e la sua integrazione nel sistema, abbiamo seguito una serie di passaggi:

1. **Creazione del bot con BotFather**  
   BotFather è uno strumento ufficiale di Telegram che permette di creare nuovi bot. Durante la configurazione, è stato richiesto di assegnare un nome e un username univoco al bot.
2. **Generazione del token API**  
   Una volta creato il bot, BotFather ha generato un token API, necessario per consentire all’applicazione Python di interagire con il bot e inviare messaggi.
3. **Interazione con il bot**  
   Dopo la configurazione iniziale, è stato inviato un messaggio **/start** dal telefono al nuovo bot. Questo passaggio è stato necessario per avviare la comunicazione tra il bot e l'account Telegram.
4. **Recupero del CHAT\_ID**  
   Per poter inviare messaggi a un account specifico, è necessario conoscere il suo **CHAT\_ID**. Questo valore è stato ottenuto accedendo al link: https://api.telegram.org/ILTUOTOKEN/getUpdates (sostituendo ILTUOTOKEN con il token API del mio bot).
5. **Creazione della classe TelegramBot**  
   Nel codice dell’applicazione Python è stata definita la classe **TelegramBot**, la quale viene inizializzata con il **TOKEN**. All'interno della classe è stato implementato il metodo **send\_alert**, che prende come parametro il **CHAT\_ID** e si occupa di inviare un messaggio tramite il bot utilizzando la libreria **pyTelegramBotAPI**.
6. **Integrazione con il sistema di monitoraggio**  
   Infine, il metodo **send\_alert** viene richiamato nel codice quando il livello dell’acqua nel serbatoio scende sotto il **20%** e sono trascorsi più di **60 secondi** dall'ultimo avviso. In questo modo, il sistema invia automaticamente una notifica all’account Telegram, avvisando dell’abbassamento del livello dell’acqua.

# Funzionamento e Risultati

L'idea iniziale prevedeva l'installazione del sistema direttamente all'interno di una cisterna, ma per motivi di praticità e per facilitare i test, abbiamo optato per l'uso di un contenitore di plastica con chiusura. Sebbene questo contenitore fosse significativamente più piccolo rispetto a una cisterna tradizionale, ciò ha comportato alcune difficoltà, soprattutto in relazione al sensore a ultrasuoni. Il Grove Ultrasonic Ranger, infatti, ha un range massimo di 512 cm, ma in spazi più contenuti come quelli del contenitore di plastica, la distanza tra il sensore e la superficie dell'acqua si riduceva, rendendo la misurazione più delicata.

Nonostante questi limiti, il sistema ha continuato a funzionare correttamente, rilevando con precisione il livello dell'acqua. I dati, infatti, sono stati raccolti e inviati in tempo reale a ThingsBoard, permettendo una visualizzazione accurata del livello tramite interfaccia. Questo ha confermato che, pur operando su distanze relativamente corte, il sensore è stato comunque in grado di garantire letture affidabili anche in un ambiente ristretto. In aggiunta, la connessione al Wi-Fi, la trasmissione dei dati in tempo reale e la gestione delle notifiche tramite Telegram hanno funzionato senza intoppi, consentendo di monitorare il sistema con un’interfaccia utente fluida e ben integrata. Di conseguenza, siamo fiduciosi che, una volta implementato su una cisterna di dimensioni maggiori, il sistema sarà in grado di offrire performance altrettanto efficaci.

Immagine che contiene interno, tavolo, plastica, cibo

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene acquario, Materiale trasparente, Trasparenza, vetro

Il contenuto generato dall'IA potrebbe non essere corretto.

### RISULTATI

Immagine che contiene testo, schermata, software, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

La GIF realizzata evidenzia il funzionamento integrato del sistema, mostrando in tempo reale tre componenti fondamentali. In alto, lo SmartThings Dashboard visualizza il disegno dinamico della cisterna che si adatta alla variazione della percentuale di riempimento, accompagnato da un grafico a linee che traccia lo storico dei livelli d’acqua.

A destra, l’interfaccia dell’applicazione Python presenta il livello attuale della cisterna, l'orario dell'ultimo aggiornamento e una stima del tempo rimanente per lo svuotamento, calcolata tramite una regressione lineare sui dati raccolti. Durante il test, il livello dell'acqua è sceso dal 62% al 14%, evidenziando chiaramente l'andamento del monitoraggio.

Inoltre, il sistema ha automaticamente inviato un messaggio di alert tramite Telegram (vedi screenshot del messaggio), garantendo un supporto tempestivo in caso di situazioni critiche.

Immagine che contiene testo, schermata, Software multimediale

Il contenuto generato dall'IA potrebbe non essere corretto.