

Relazione dell'Assignment 2: Smart-Drone-Hangar

Lorenzo Antonioli
Eleonora Bianco
Giulia Fares
Luca Varale Rolla

3 febbraio 2026

Indice

1	Introduzione	2
2	Architettura del Sistema (Drone Hangar)	2
2.1	Hardware	2
2.2	Software: Architettura Task-Based	3
3	Macchina a Stati Finiti Sincrona (FSMs)	3
4	Drone Remote Unit (Sottosistema PC)	5
5	Protocollo di Comunicazione	5

1 Introduzione

Questo progetto implementa un sistema embedded per la gestione automatizzata di un hangar per droni. Il sistema è composto da due sottosistemi principali che comunicano tramite linea seriale:

1. **Drone Hangar (Arduino):** Gestisce i sensori, gli attuatori e la logica di controllo locale.
2. **Drone Remote Unit (PC):** Un’interfaccia per il controllo remoto e il monitoraggio della telemetria.

2 Architettura del Sistema (Drone Hangar)

2.1 Hardware

Il sistema hardware è basato su Arduino Uno e comprende i seguenti componenti:

- **Sensori:**
 - **Sonar:** Misura la distanza del drone dall’hangar.
 - **PIR:** Rileva la presenza del drone nell’area di atterraggio.
 - **Sensore di Temperatura:** Monitora la temperatura interna.
- **Attuatori e Output:**
 - **Servo Motor:** Controlla l’apertura e la chiusura della porta dell’hangar.
 - **LCD:** Visualizza lo stato corrente del sistema.
 - **LED:** L1 (Stato INSIDE), L2 (Operazioni in corso), L3 (ALARM).
- **Input:** Pulsante di RESET per ripristinare il sistema da uno stato di allarme.

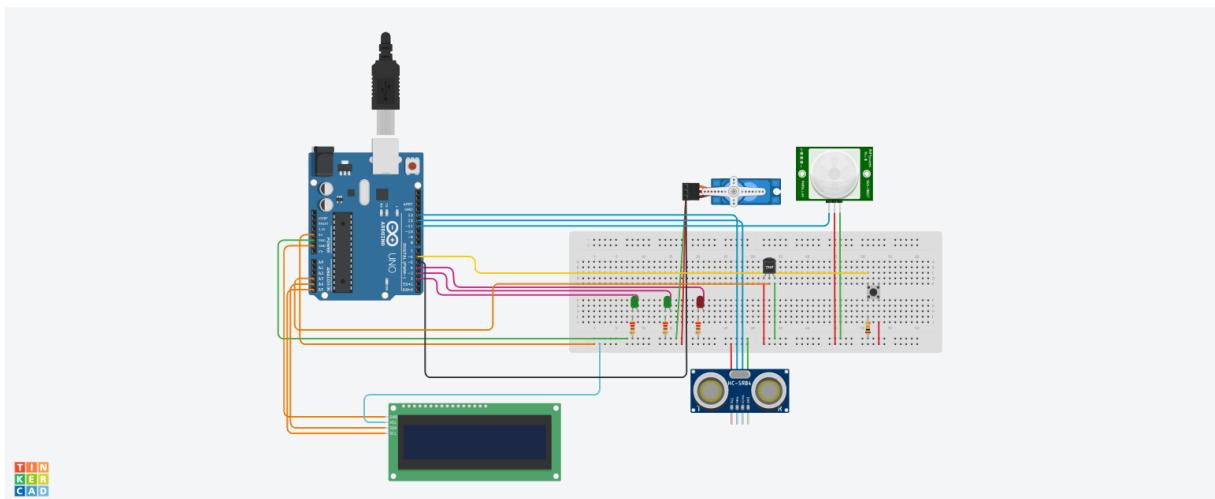


Figura 1: Schema dell’Hangar

2.2 Software: Architettura Task-Based

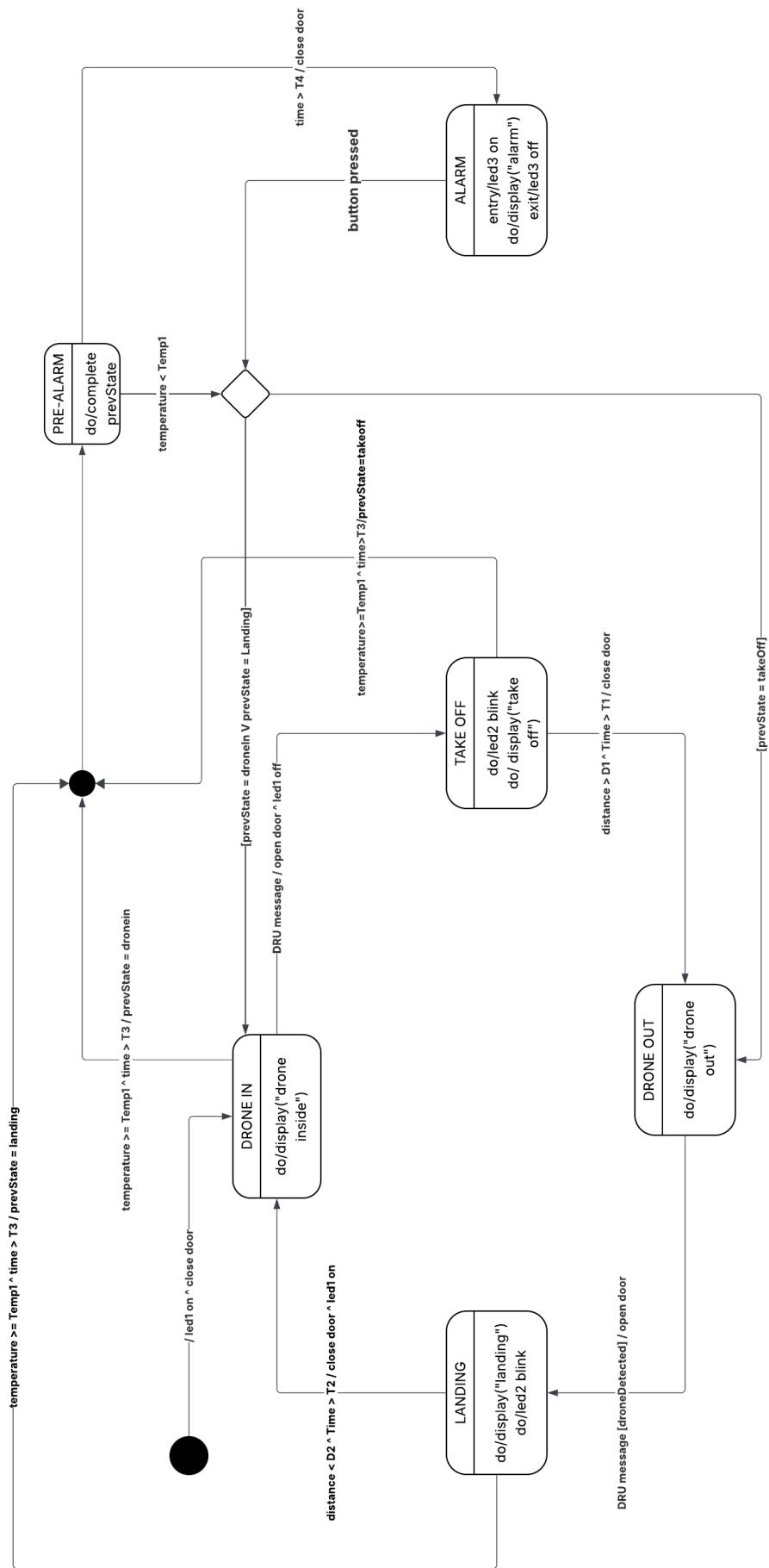
Il firmware Arduino segue un'architettura Task-Based gestita da uno Scheduler. Lo scheduler esegue i task basandosi su un tick base (configurato a 50ms in ‘main.ino’). I task principali implementati sono:

- **SensingTask (Periodo: 50ms):** Acquisisce i dati dai sensori (Sonar, Temperatura, PIR) e aggiorna l’oggetto condiviso ‘Context’..
- **LogicTask (Periodo: 100ms):** Implementa la Macchina a Stati Finiti (FSM). Controlla gli attuatori (Servo, LED, LCD) in base allo stato del sistema e ai dati dei sensori.
- **CommTask (Periodo: 200ms):** Gestisce la comunicazione seriale con il PC. Riceve i comandi (es. decollo/atterraggio) e invia i dati (stato, sensori).

3 Macchina a Stati Finiti Sincrona (FSMs)

Il comportamento del sistema è modellato tramite una FSMs implementata nella classe ‘LogicTask’. Gli stati sono:

- **INSIDE:** Stato iniziale. Porta chiusa, drone a riposo. Il sistema attende il comando di decollo.
- **TAKEOFF:** La porta si apre, il LED L2 lampeggi. Il sistema transita nello stato *DroneOut* quando il sonar rileva una distanza $> D1$ per un tempo T1.
- **DRONE OUT:** Il drone è all'esterno. Porta chiusa. Il sistema attende il comando di atterraggio e il rilevamento del PIR.
- **LANDING:** La porta si apre, il LED L2 lampeggi. Il sistema transita in *Inside* quando la distanza $< D2$ per un tempo T2.
- **PRE ALARM:** Attivato se la temperatura supera Temp1 per un tempo T3. Sospende nuove operazioni ma permette il completamento di quelle in corso.
- **ALARM:** Attivato se la temperatura supera Temp1 per un tempo T4. La porta viene chiusa immediatamente, il LED Rosso (L3) si accende e il sistema si blocca fino alla pressione del tasto RESET.



4 Drone Remote Unit (Sottosistema PC)

Il DRU è sviluppato in Java utilizzando la libreria Swing per l'interfaccia grafica. La comunicazione seriale è gestita tramite ‘CommChannel’ implementata nella classe ‘SerialCommChannel’ (basata su JSSC).

5 Protocollo di Comunicazione

Lo scambio dati avviene tramite messaggi testuali su seriale:

- **PC → Arduino:**

- ‘CMD:TAKEOFF’: Richiesta di decollo.
- ‘CMD:LAND’: Richiesta di atterraggio.

- **Arduino → PC:**

- ‘STATE:<valore>’: Invia lo stato corrente della FSM.
- ‘DIST:<valore>’: Invia la distanza misurata dal sonar.
- ‘TEMP:<valore>’: Invia la temperatura corrente.
- ‘MOV:<SI/NO>’: Invia SI o NO in base a ciò che ha rilevato il PIR.