

# Relazione Assignment 02 IoT

Smart Drone Hangar

Lucia Pola

Matricola: 0001118424

Leonardo Meloni

Matricola: 0001128110

Università di Bologna

Corso di Laurea in Ingegneria e Scienze Informatiche

A.A 2025/2026

14 febbraio 2026

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Componenti . . . . .	2
1.1.1	DRONE HANGAR (basato su Arduino) . . . . .	2
1.1.2	DRONE REMOTE UNIT (DRU) . . . . .	2
1.2	Comportamento . . . . .	2
<b>2</b>	<b>Progettazione Drone Hangar</b>	<b>4</b>
2.1	Hardware . . . . .	4
2.2	Software . . . . .	4
2.2.1	I 5 Task Principali . . . . .	5
<b>3</b>	<b>Progettazione Drone Remote Unit</b>	<b>9</b>
<b>4</b>	<b>Protocollo di comunicazione</b>	<b>10</b>
4.1	Direzione DRU → DRH (Comandi) . . . . .	10
4.2	Direzione DRH → DRU (Stato) . . . . .	10
<b>5</b>	<b>Configurazioni</b>	<b>11</b>
5.1	Mappa dei Collegamenti (Pinout) . . . . .	11
5.2	Configurazione dello Scheduler . . . . .	11
5.3	Parametri Funzionali . . . . .	12

# 1 Introduzione

## 1.1 Componenti

### 1.1.1 DRONE HANGAR (basato su Arduino)

- **DRONE PRESENCE DETECTOR (DPD):** sensore PIR, utilizzato per rilevare la presenza di un drone nelle vicinanze (sopra) dell'hangar.
- **DRONE DISTANCE DETECTOR (DDD):** sensore sonar (ultrasuoni), per misurare la distanza del drone quando si trova all'interno dell'hangar.
- **HANGAR DOOR (HD):** servomotore che controlla l'apertura e la chiusura della porta dell'hangar.
- **OPERATOR LCD:** display LCD I2C, utilizzato per interagire con l'operatore.
- **L1 e L2:** LED verdi; **L3** è un LED rosso.
- **RESET:** pulsante tattile di reset.
- **TEMP:** sensore di temperatura analogico.

### 1.1.2 DRONE REMOTE UNIT (DRU)

- Programma Java in esecuzione su PC.
- Sottosistema basato su un'interfaccia grafica (GUI Swing) che simula un drone in grado di inviare e ricevere comandi.
- Comunica con il *DRONE HANGAR* tramite linea seriale (Serial Line).

## 1.2 Comportamento

All'avvio, il sistema inizia con la porta dell'hangar (HD) chiusa; si presume che il drone sia all'interno, in stato di riposo. Il LED L1 è acceso, L2 e L3 sono spenti e l'LCD visualizza il messaggio **DRONE INSIDE**.

**Fase di Decollo (Take-off)** Il drone attiva il comando di apertura della porta dell'hangar inviando un messaggio tramite il sottosistema DRU. Alla ricezione del comando, la porta HD si apre, l'LCD visualizza **TAKE OFF** e il sistema attende l'uscita del drone. Per determinare quando il drone è uscito, viene utilizzato il sensore DDD: quando la distanza misurata è superiore a  $D_1$  per più di  $T_1$  secondi, si assume che il drone sia uscito e la porta HD viene chiusa. L'LCD visualizza quindi **DRONE OUT**.

**Fase di Atterraggio (Landing)** Quando il drone si avvicina all'hangar, invia il comando di apertura (via DRU). Se, alla ricezione del comando, il sensore DPD rileva la presenza del drone, la porta HD si apre e l'LCD visualizza **LANDING**. Il sistema attende quindi che il drone entri e atterri. Quando la distanza misurata dal DDD è inferiore a  $D_2$  per più di  $T_2$  secondi, si assume che il drone sia atterrato e la porta viene chiusa. L'LCD visualizza nuovamente **DRONE INSIDE**.

- Durante le fasi di decollo e atterraggio, **L2 lampeggia** con un periodo di 0,5 secondi; negli altri casi è spenta.

**Monitoraggio della Temperatura e Allarmi** Ogni volta che il drone si trova all'interno dell'hangar (sia a riposo, sia durante il decollo o l'atterraggio), il sistema di monitoraggio della temperatura è attivo.

- **Pre-allarme:** Se viene rilevata una temperatura  $\geq Temp_1$  per più di  $T_3$  secondi, il sistema entra in stato di pre-allarme. In questo stato, nuovi decolli e atterraggi sono sospesi. Se una manovra è già in corso, viene completata. Se la temperatura scende sotto  $Temp_1$ , il sistema torna al funzionamento normale.
- **Allarme:** Se viene rilevata una temperatura  $Temp_2 > Temp_1$  per più di  $T_4$  secondi, la porta HD viene chiusa, la spia L3 si accende e l'LCD visualizza **ALARM**. Se il drone è all'esterno, il messaggio di **ALARM** viene inviato via DRU. Tutte le operazioni sono sospese fino alla pressione del pulsante di **RESET**.

**Parametri e Interfaccia (DRU)** I valori dei parametri  $D_1, D_2, T_1, T_2, T_3, T_4, Temp_1, Temp_2$  sono configurabili per la fase di test. La GUI della DRU permette di:

- Inviare comandi (decollo/atterraggio).
- Visualizzare lo stato del drone (REST, TAKING OFF, OPERATING, LANDING).
- Visualizzare lo stato dell'hangar (NORMAL, ALARM).
- Visualizzare la distanza dal suolo in fase di atterraggio.

## 2 Progettazione Drone Hangar

### 2.1 Hardware

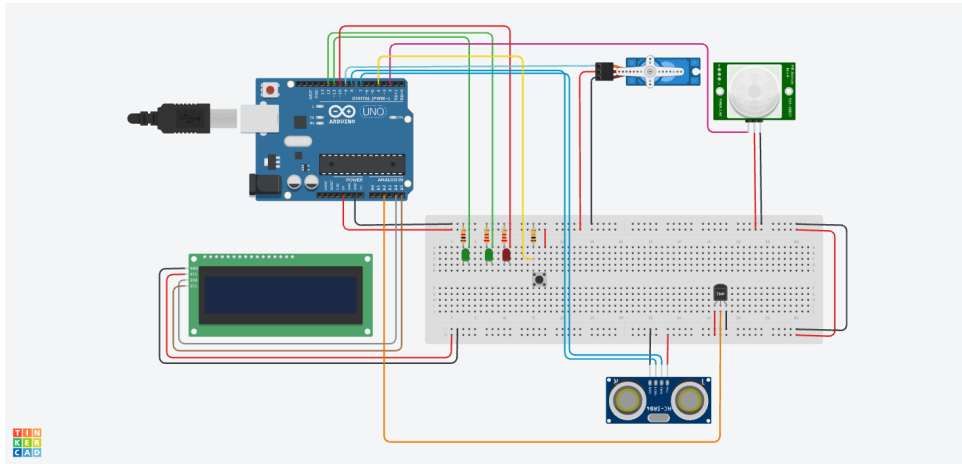


Figura 1: Schema dei collegamenti hardware del sistema Drone Hangar.

Per l'implementazione del prototipo sono stati utilizzati i seguenti componenti hardware:

- **Microcontrollore:** Arduino UNO R3.
- **Rilevamento Presenza:** Sensore PIR AM312
- **Monitoraggio Termico:** Sensore di temperatura analogico TMP36.
- **Interfaccia Utente:** Display LCD  $16 \times 2$  I2C.
- **Attuatore Porta:** Microservo HS-53.
- **Misurazione Distanza:** Sonar HC-SR04.
- **Input Operatore:** Pulsante tattile per la funzione di *RESET*.
- **Segnalazione Visiva:** 3 LED (2 verdi e 1 rosso).

### 2.2 Software

L'architettura del software è progettata seguendo il paradigma del **multitasking cooperativo**, orchestrato da uno *scheduler* che gestisce l'esecuzione periodica di 5 task indipendenti. Tale approccio garantisce che le operazioni critiche, come il monitoraggio della temperatura o del sonar, avvengano con tempistiche precise e deterministiche.

I task comunicano tra loro attraverso una struttura dati centrale denominata **Context**, che funge da memoria condivisa e permette il disaccoppiamento dei componenti.

### 2.2.1 I 5 Task Principali

1. **HangarTask** È il cuore logico del sistema. Gestisce la macchina a stati di alto livello (REST, TAKE OFF, OPERATING, LANDING). Monitora i dati provenienti dal DDD (sonar) e dal DPD (PIR) e verifica il superamento delle soglie temporali  $T_1$  e  $T_2$  per convalidare gli spostamenti del drone.
2. **DoorTask** Si occupa esclusivamente del controllo del servomotore. Legge lo stato desiderato della porta dal *Context* e ne gestisce il movimento. In caso di stato di ALARM, esegue una chiusura forzata immediata per ragioni di sicurezza.
3. **BlinkingTask** Controlla il LED verde L2. Quando il sistema rileva una fase di decollo o atterraggio nel *Context*, attiva il lampeggio con un periodo di 0,5 s. Negli altri stati, si assicura che il LED rimanga spento.
4. **AlarmTask** Monitora costantemente il sensore *TEMP*. Gestisce la logica di protezione implementando gli stati di NORMAL, PRE-ALARM e ALARM. Utilizza le soglie temporali di sicurezza  $T_3$  e  $T_4$  per filtrare picchi di temperatura temporanei e reagisce alla pressione del pulsante di *RESET*.
5. **SerialMonitorTask** Gestisce il protocollo di comunicazione sulla porta seriale. Sincronizza il sistema con la DRU esterna inviando lo stato globale ogni 300 ms e decodifica i comandi in entrata (*takeoff-req*, *landing-req*) aggiornando i relativi flag nel *Context*.

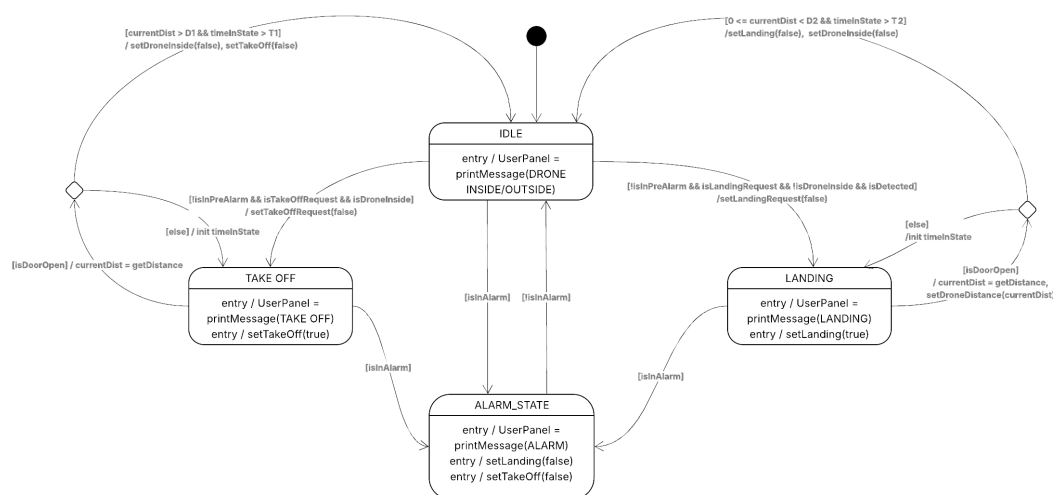


Figura 2: Diagramma di stato per l'hangar task

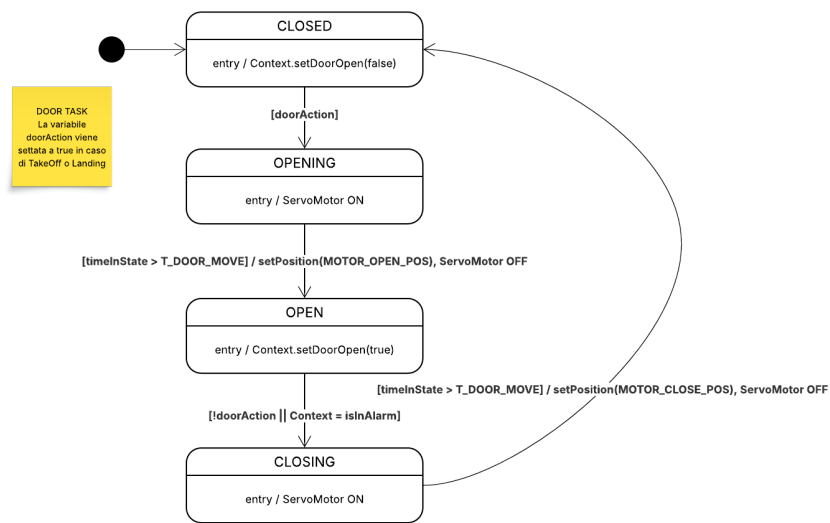


Figura 3: Diagramma di stato per la door task

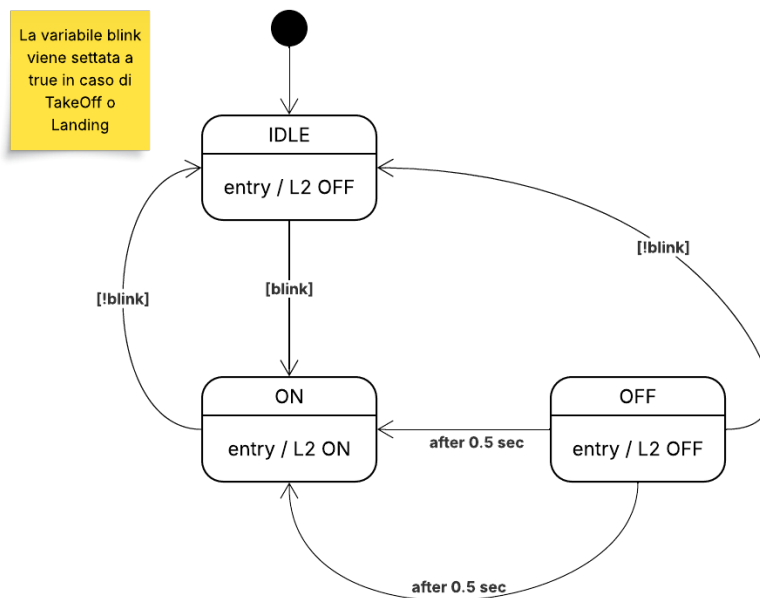


Figura 4: Diagramma di stato per il blink task

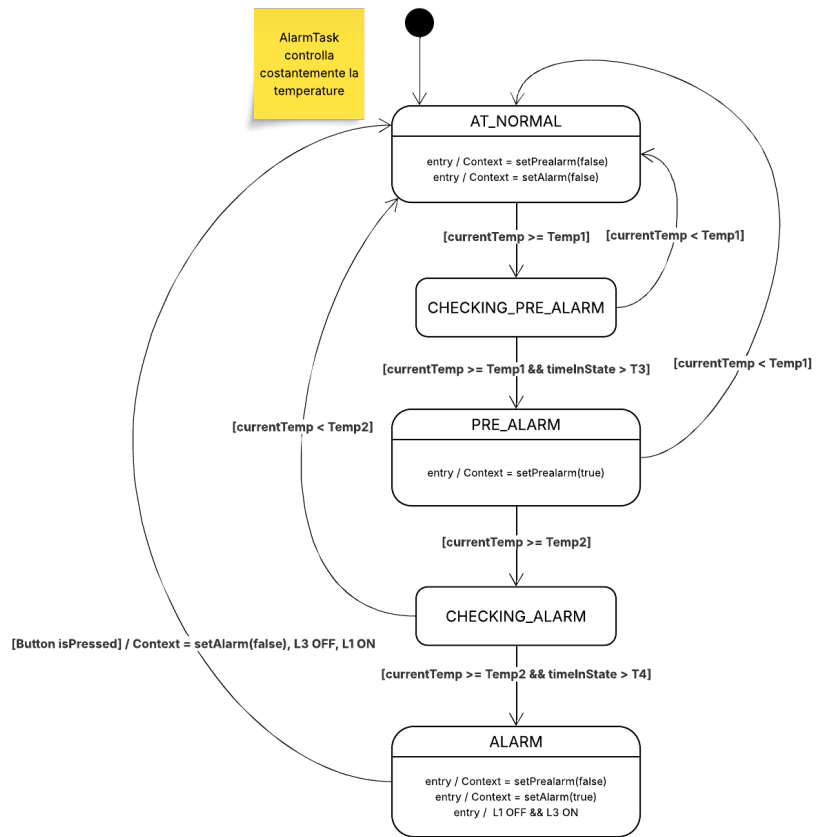


Figura 5: Diagramma di stato per l'alarm task



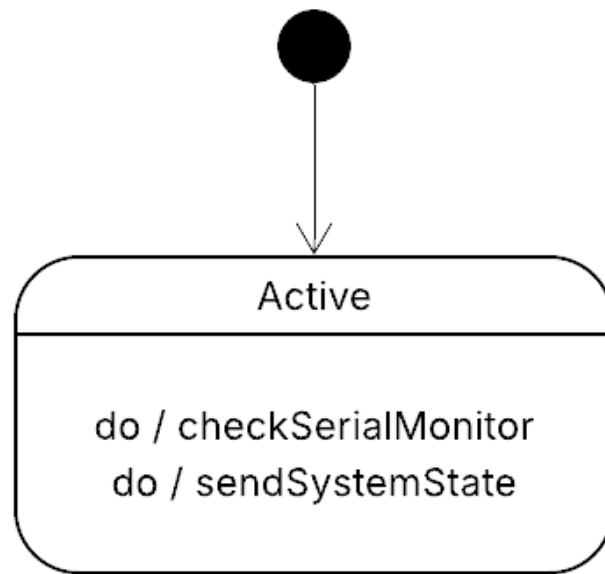


Figura 6: Diagramma di stato per la serial task

### 3 Progettazione Drone Remote Unit

Il software di simulazione del drone è stato sviluppato in linguaggio **Java** utilizzando la libreria grafica **Swing**. L'architettura del software si basa sul pattern architetturale **Model-View-Controller (MVC)**, che garantisce una netta separazione tra la logica di gestione dei dati, l'interfaccia utente e il controllo degli eventi.

Questo programma funge da pannello di controllo per l'operatore, permettendo di:

- Inviare richieste di decollo (**TAKE OFF**) o atterraggio (**LANDING**) verso l'hangar.
- Visualizzare in tempo reale lo stato dell'hangar.
- Monitorare l'operatività del drone.
- Visualizzare la distanza dal suolo del drone rilevata durante la fase di atterraggio.

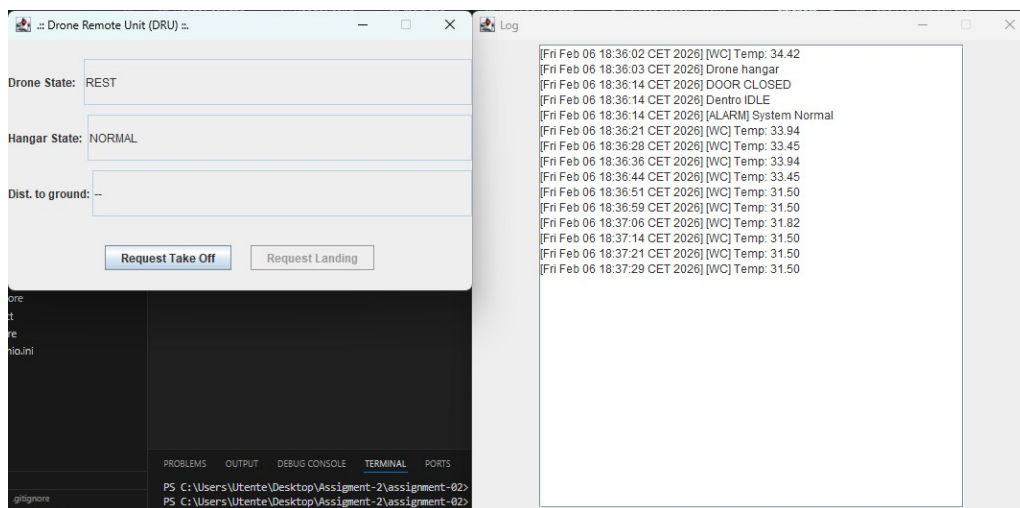


Figura 7: Interfaccia Grafica (GUI) della Drone Remote Unit che mostra i controlli di volo e lo stato del sistema.

## 4 Protocollo di comunicazione

La comunicazione tra la DRU e il Drone Hangar avviene tramite connessione seriale asincrona. Il protocollo è basato sullo scambio di messaggi testuali ASCII per garantire semplicità di debugging e interoperabilità.

### 4.1 Direzione DRU → DRH (Comandi)

La DRU invia comandi al microcontrollore per richiedere cambi di stato nel drone. I messaggi sono stringhe semplici senza header complessi.

Comando	Descrizione Funzionale
takeoff-req	Richiede l'avvio della procedura di decollo.
landing-req	Richiede l'avvio della procedura di atterraggio.

Tabella 1: Comandi inviati dalla DRU all'Hangar.

### 4.2 Direzione DRH → DRU (Stato)

Il Drone Hangar invia periodicamente (ogni 300 ms) lo stato completo del sistema per aggiornare l'interfaccia grafica. Il formato del messaggio è:

`dh:st:<DRONE_STATE>:<HANGAR_STATE>:<DISTANCE>`

Dove:

- `dh:st:` è l'header del messaggio che identifica l'applicazione ("dh") e il tipo di messaggio ("st" = state).
- `<DRONE_STATE>` indica lo stato operativo del drone. Valori possibili:
  - `REST`: Drone a riposo dentro l'hangar.
  - `TAKE OFF`: Fase di decollo in corso.
  - `OPERATING`: Drone all'esterno operativo.
  - `LANDING`: Fase di atterraggio in corso.
- `<HANGAR_STATE>` indica lo stato di sicurezza dell'hangar. Valori possibili:
  - `NORMAL`: Temperatura nella norma.
  - `PRE-ALARM`: Temperatura elevata, operazioni bloccate.
  - `ALARM`: Temperatura critica, sistema bloccato.
- `<DISTANCE>` è il valore numerico (float) della distanza rilevata dal sonar (utile in fase di atterraggio) oppure stringa "--" se non pertinente.

**Esempio:** `dh:st:REST:NORMAL:--` indica drone a riposo e sistema ok.

## 5 Configurazioni

Tutte le costanti e le configurazioni hardware del sistema sono centralizzate nel file header `include/config.h`.

### 5.1 Mappa dei Collegamenti (Pinout)

Di seguito sono riportati i collegamenti fisici tra i componenti e la scheda Arduino UNO R3.

Componente	Pin	Tipo
Bottone Reset	4	Digital Input
LED 1 (Power/Rest)	11	Digital Output
LED 2 (Blinking)	12	Digital Output
LED 3 (Alarm)	10	Digital Output
Sonar Echo	8	Digital Input
Sonar Trigger	7	Digital Output
Servomotore (HD)	9	PWM Output
Sensore PIR (DPD)	2	Digital Input
Sensore Temperatura	A2	Analog Input
LCD SDA	A4	I2C
LCD SCL	A5	I2C

Tabella 2: Configurazione dei Pin Hardware.

### 5.2 Configurazione dello Scheduler

La frequenza di esecuzione di ciascun task è definita per bilanciare la reattività del sistema e l'uso delle risorse.

Task	Periodo (ms)
AlarmTask	100
BlinkingTask	500
DoorTask	100
HangarTask	100
SerialMonitorTask	250

Tabella 3: Periodi di esecuzione dei Task.

### 5.3 Parametri Funzionali

Il comportamento del sistema è regolato dai seguenti parametri di soglia e tempo.

Parametro	Descrizione	Valore (Default)
$D_1$	Soglia distanza per uscita drone	0.15 m
$D_2$	Soglia distanza per atterraggio drone	0.10 m
$T_1$	Tempo permanenza per conferma uscita	5000 ms
$T_2$	Tempo permanenza per conferma atterraggio	4000 ms
$T_3$	Tempo persistenza per Pre-Allarme	3000 ms
$T_4$	Tempo persistenza per Allarme	3000 ms
$Temp_1$	Soglia temperatura Pre-Allarme	40 °C
$Temp_2$	Soglia temperatura Allarme	50 °C

Tabella 4: Parametri di configurazione del sistema.