

Domotic Home - Progetto IoT

Requisiti

La descrizione del sistema e i requisiti sono consultabili nel documento **requirements.pdf**

Video

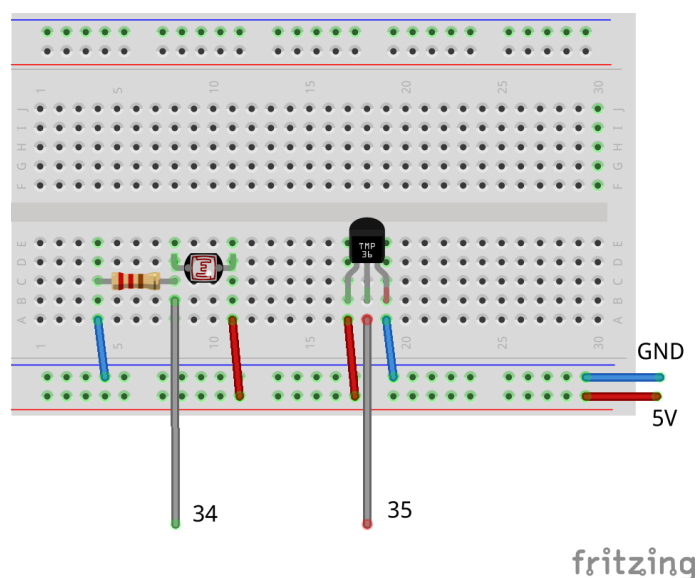
Video dimostrativo disponibile al link: [Domotic Home video link](#)

Design

Domus Sensor Board

Questo componente si occupa di rilevare la luminosità esterna e la temperatura interna della abitazione e di comunicarli all'applicativo java (Domus Service) utilizzando il protocollo MQTT su uno specifico topic.

Di seguito lo schema del circuito su breadboard, i pin indicati sono riferiti al SoC ESP32:



Ogni 5 secondi viene eseguito il SensorBoardController che implementa l'interfaccia MainController. Ad ogni esecuzione vengono letti i sensori quindi i valori vengono inviati in formato json via MQTT.

L'invio dei dati viene delegato al CommAgent la cui implementazione in questo progetto è MQTTCommAgent. In caso si volesse utilizzare un'altra tecnologia/protocollo per l'invio dei dati occorre implementare l'interfaccia CommAgent, interfaccia dedicata per la comunicazione dei dati.

Domus Service

Questo sottosistema permette la sincronizzazione dei dati fra Hub, Dashboard e Sensor Board.

Domus Service si compone di tre principali componenti:

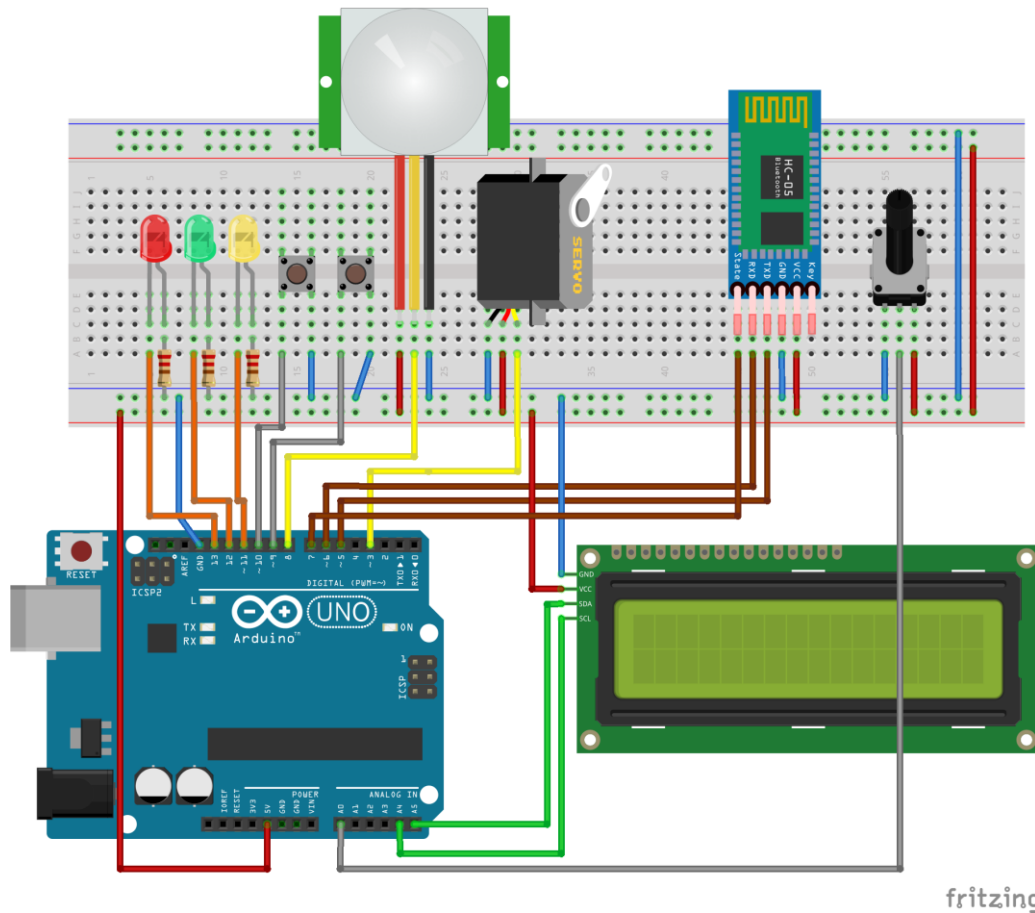
- *MQTTAgent*: si occupa di ricevere i dati dei sensori tramite protocollo MQTT, pubblicati dalla Sensor Board, salvarli e di inoltrarli via seriale a Domus Hub.
- *HTTPServerAgent*: ha il compito di rispondere alle richieste GET effettuate dalla Dashboard in modo asincrono (Ajax) ad intervalli regolari. La risposta consiste in un file json contenente lo stato della abitazione e i valori letti dai sensori, entrambi con relativo timestamp di ultimo aggiornamento.

- *SerialRecvListener*: si occupa esclusivamente della ricezione dello stato della abitazione inviata dal Domus Hub (arduino) sulla seriale. Viene verificata la presenza di dati sulla porta e successivamente viene aggiornato lo stato della abitazione memorizzato con relativo timestamp.

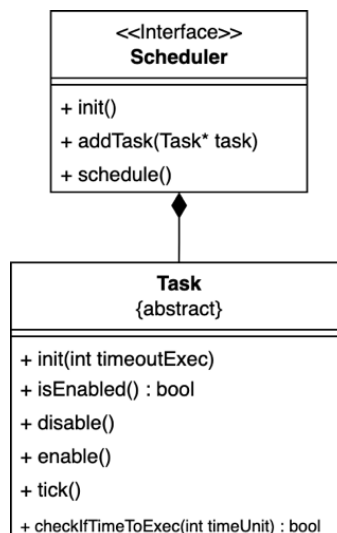
Domus Hub

È il sottosistema più importante, si occupa della gestione dello stato della casa e del controllo dei dispositivi della abitazione. Gli impianti considerati, come da specifiche, sono quelli della illuminazione, del riscaldamento, dell'allarme e del portellone del garage.

Schema del circuito su breadboard:



È stata utilizzata una **architettura task-based** con **scheduling cooperativo** dove ciascun task si occupa unicamente di gestire ed aggiornare lo stato di un impianto della casa ogni timeoutExec millisecondi.



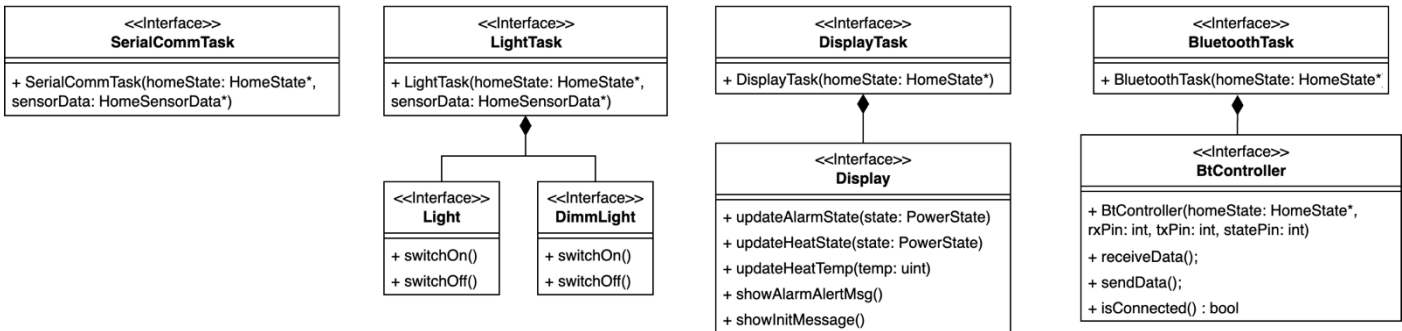
Lo stato di ogni impianto è modellato con una **macchina a stati finiti (FSM) sincrona**, ad ogni esecuzione del task vengono letti gli input e viene aggiornato lo stato.

Gli impianti presenti nella abitazione e il corrispondente task sono:

- Riscaldamento (HeatingSysTask)
- Illuminazione (LightTask)
- Allarme (AlarmSysTask)
- Garage (GarageTask)

Oltre a questi sono presenti altri task che si occupano di gestire l'invio e ricezione dei dati via seriale (SerialCommTask) e via bluetooth (BluetoothTask) e di controllare il display (DisplayTask).

Di seguito gli schemi dei task sviluppati e la relativa FSM (ove rilevante):

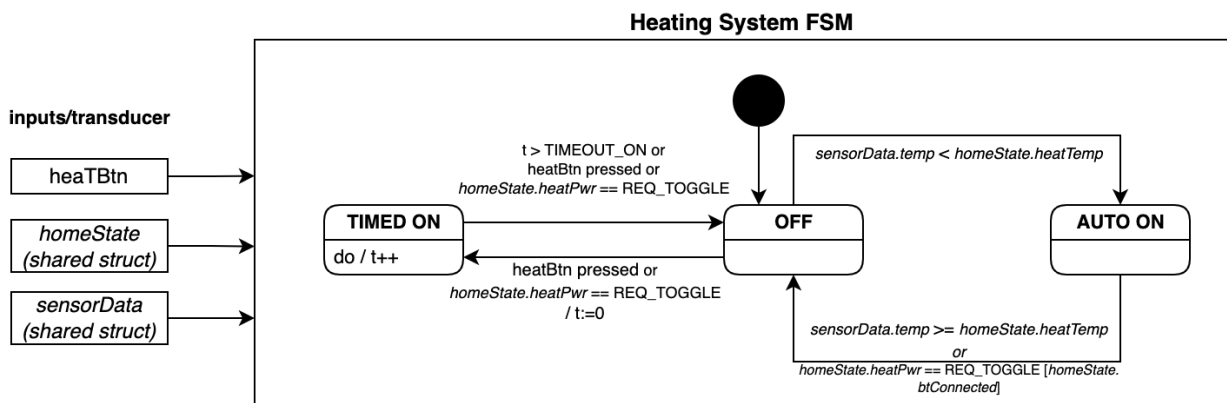
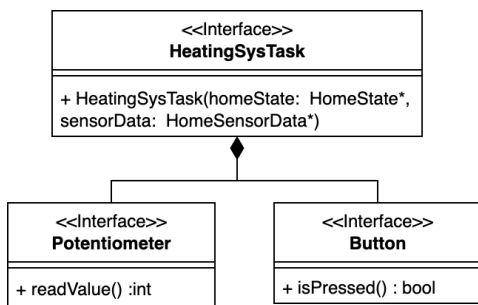


Heating System Task

HeatingSysTask gestisce l'impianto di riscaldamento. Legge il valore del potenziometro per impostare la temperatura desiderata e il pulsante per cambiare lo stato di accensione dell'impianto.

In caso la temperatura rilevata dalla Sensor Board sia inferiore alla temperatura impostata con il potenziometro l'impianto passa allo stato AUTO ON finché queste due temperature non si eguagliano (si passa allo stato OFF). In caso di collegamento con l'app è possibile forzare lo spegnimento.

Se ci si trova nello stato di OFF è possibile sia con pulsante che tramite app accendere/spegnere l'impianto che viene acceso temporaneamente (stato TIMED ON).

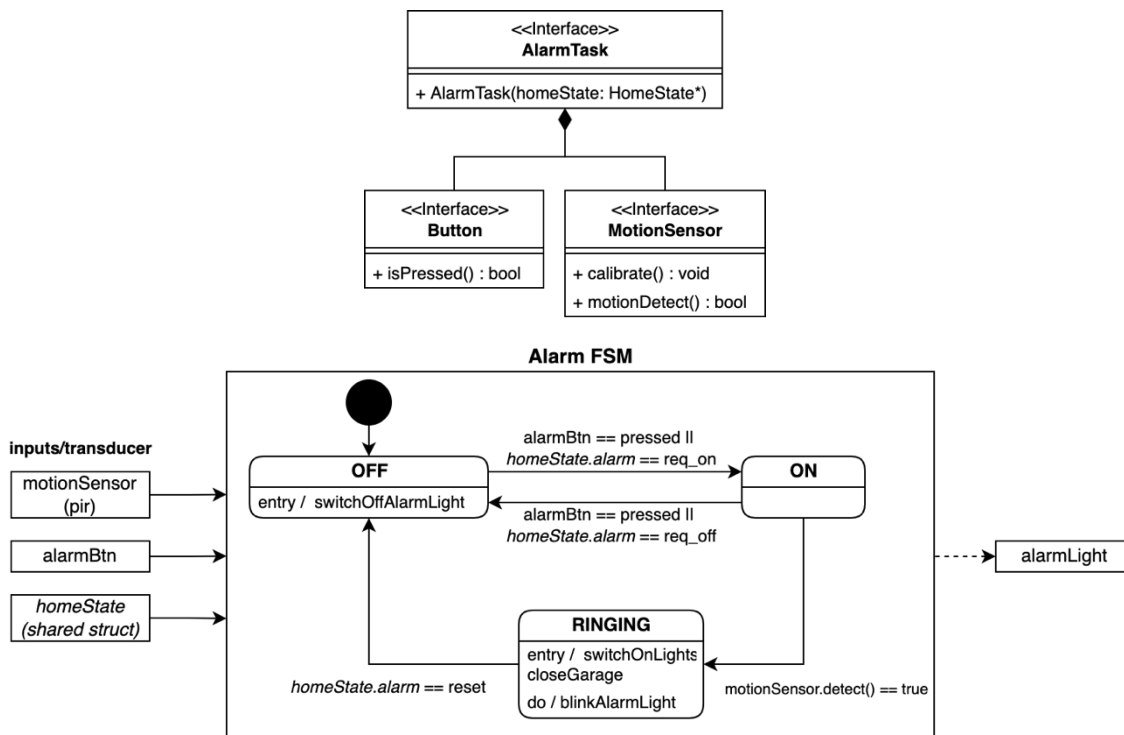


Alarm System Task

Questo task ad ogni esecuzione legge il sensore di movimento e il pulsante. Ad ogni pressione del pulsante lo stato di accensione viene cambiato. Solo se lo stato attuale è ON e in caso di movimento si passa allo stato di RINGING dove viene fatta lampeggiare alarmLight (led rosso), viene mandato il comando di chiudere il garage e vengono accese tutte le luci.

Per interrompere l'allarme occorre essere collegati con l'app bluetooth e deve essere stato ricevuto il comando di reset dell'allarme.

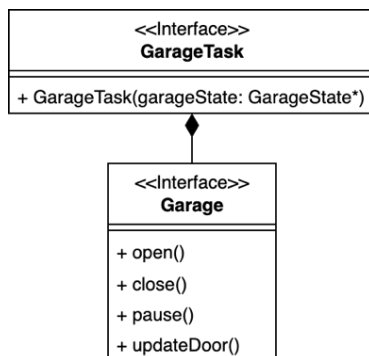
Viene eseguito ogni 200ms.

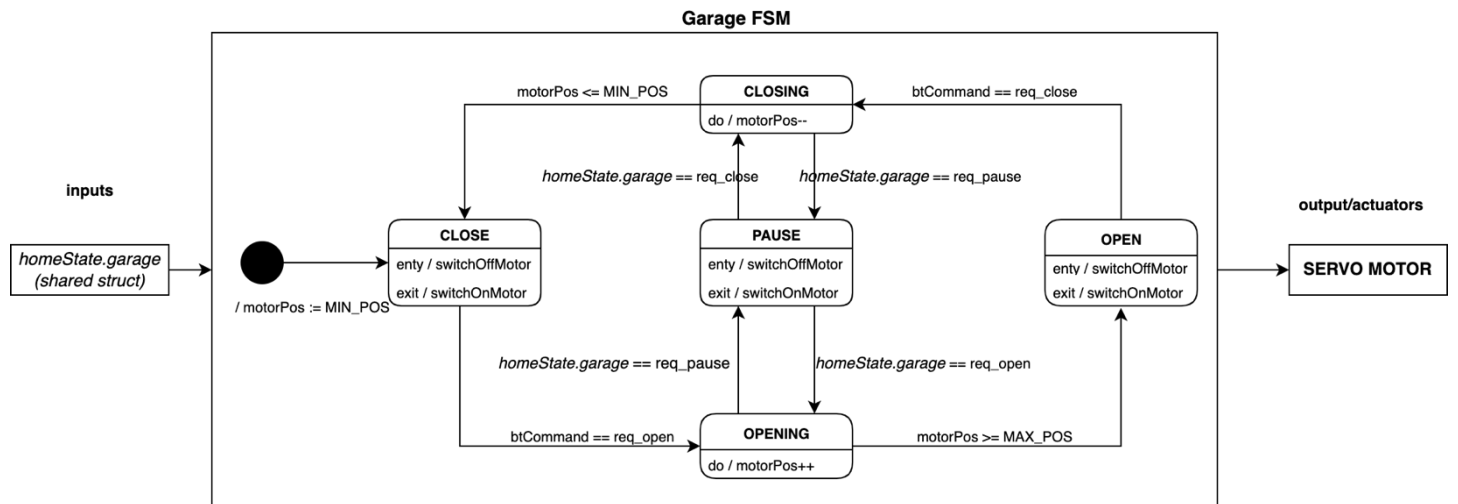


Garage Task

Il GarageTask delega la gestione dello stato del portellone all'oggetto Garage. In base al comando ricevuto via bluetooth questo task chiama il metodo corrispondente corretto su Garage e ne richiede il suo aggiornamento chiamando `updateDoor()`. Ogni aggiornamento del portellone si riflette sul servo motore.

Questo task viene eseguito ogni 40ms.

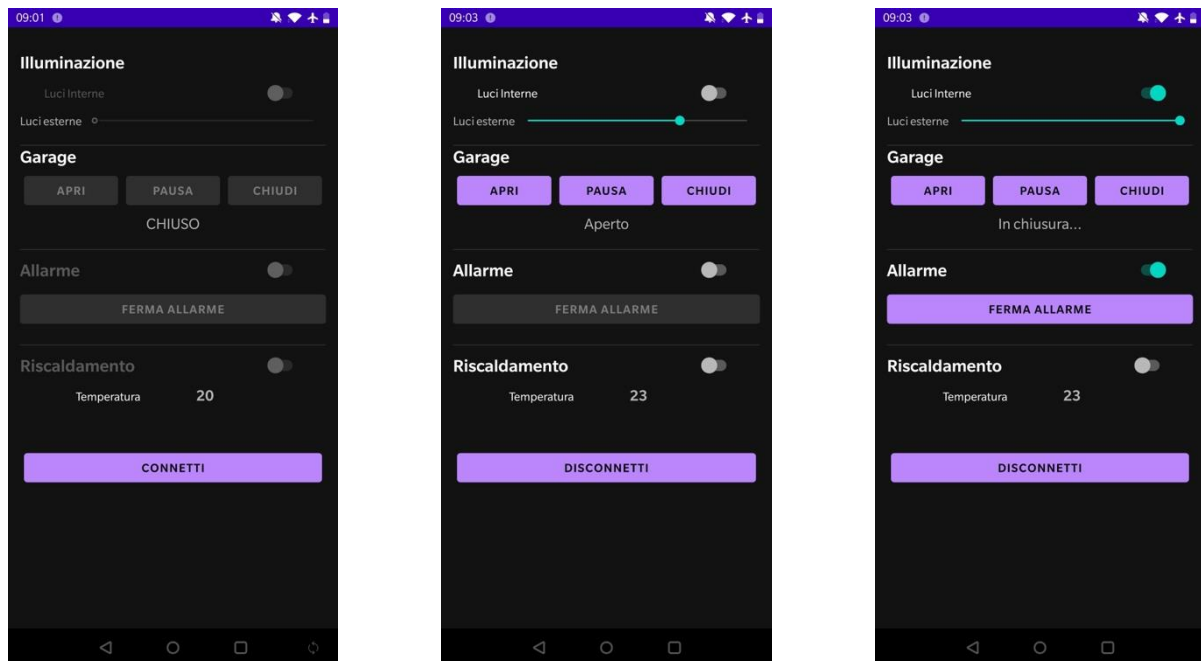




Domus App

L'applicazione mobile (Android) mostra lo stato in real-time della casa e permette di controllare direttamente le componenti domotiche. In particolare, solo attraverso l'app è possibile controllare l'apertura del garage e l'interruzione dell'allarme in caso di intrusione.

Di seguito le schermate:



Domus Dashboard

Consiste in una pagina web html dove vengono visualizzati in tempo reale lo stato della casa e i dati rilevati dai sensori. Viene indicato inoltre l'ora e data di ultimo aggiornamento delle informazioni mostrate e lo stato di connessione con l'app mobile. I dati vengono forniti da Domus Service sotto formato json a seguito di richieste GET asincrone (ajax) effettuate ogni secondo.

Domus Dashboard

Stato connessione app: Disconnessa - Controllo automatico

