

---

## **Ingegneria e scienze informatiche (Cesena)**

**A.A 2018/19**

**Valentini Luca**

**Rossolini Andrea**

**Orazi Filippo**

<b>obiettivi</b>	<b>2</b>
<b>Analisi</b>	<b>3</b>
GreenHouse Controller	3
Mobile App	3
<b>Progettazione</b>	<b>4</b>
Mobile App	4
Microcontrollore	4
Protocollo di comunicazione	5
Edge ESP8266	6
Java Server (PC)	6

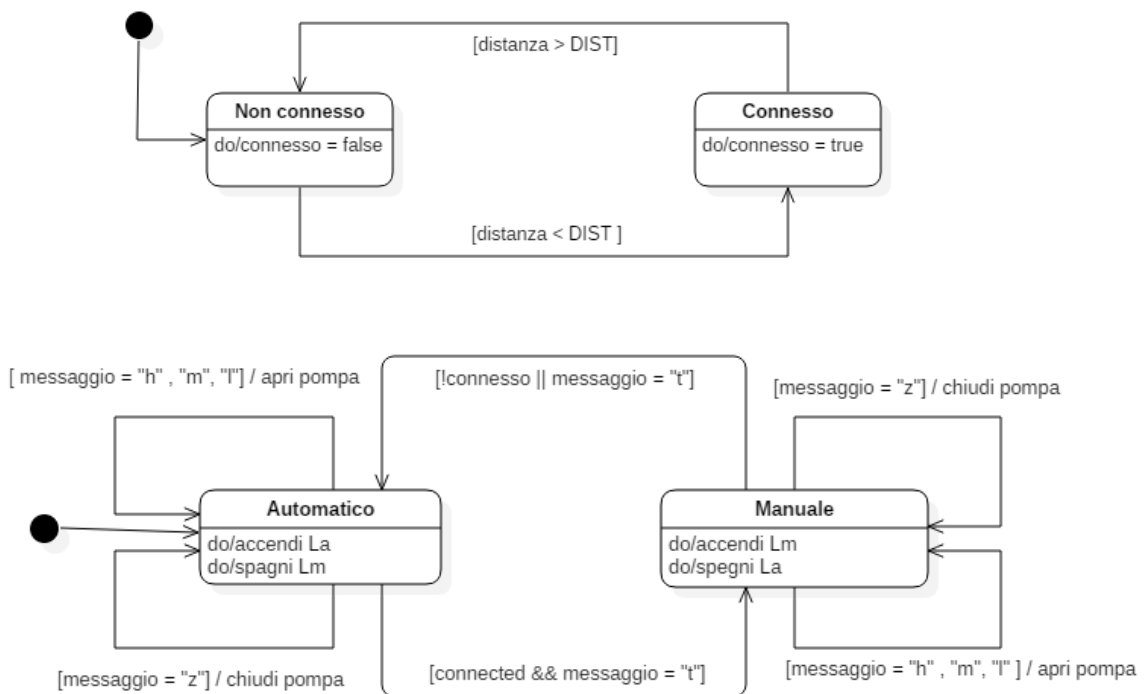
# SmartGreenHouse

## obiettivi

Realizzare un sistema embedded che rappresenti una versione semplificata di una serra smart.

Il compito della serra smart è l'irrigazione automatizzata (di un certo terreno o pianta) implementando una strategia che tenga conto dell'umidità percepita, con la possibilità di controllare e intervenire manualmente mediante mobile app.

## Analisi



## GreenHouse Controller

Permette di controllare l'apertura e chiusura degli irrigatori (pompe acqua), quindi della quantità di acqua erogata al minuto. Riceve messaggi da bluetooth (se il dispositivo è nelle vicinanze) e da serial e li interpreta in modo da gestire la pompa.

## GreenHouse Edge

Percepisce l'umidità del terreno e la invia al server tramite Wifi.

## GreenHouse Server

Event driven loop che gestisce la comunicazione tra edge controller e front end, definisce e attua la strategia di irrigazione.

# GreenHouse Front End

Permette di osservare e analizzare dati

## Mobile App

L'app Android permetterà il controllo manuale della serra in maniera remota tramite bluetooth.

Considerando di essere nelle vicinanze della serra a una distanza dal Controller inferiore o uguale a 30 cm, deve essere possibile connettersi al sistema e passare a una modalità MANUALE in cui mediante l'app si possa:

- manualmente aprire/chiudere/regolare l'erogazione dell'acqua.
- visualizzare il valore corrente dell'umidità percepita.

## Progettazione

L'approccio da noi utilizzato nello sviluppo di questo software concorrente è la decomposizione del comportamento complessivo in più task:

- Ogni task rappresenta una fase ben definita, con un relativo compito da svolgere.
- Il comportamento di ogni task viene rappresentato da un elemento della macchina a stati, come rappresentato in figura.

## Mobile App

L'app è sviluppata in java in una singola activity chiamata MainActivity.

Questa schermata permetterà di svolgere tutte le funzioni richieste in fase di analisi, come:

- connessione ad arduino via bluetooth
- visualizzazione dello stato della connessione
- modalità automatica o manuale
- chiusura o apertura della pompa
- percentuale di umidità.

Per un corretto funzionamento dell'app è importante che prima di utilizzarla si effettui la connessione bluetooth con arduino, in modo tale da poter avere un canale di comunicazione tra lo smartphone e il micro-controllore.

A connessione avvenuta l'utente potrà cliccare i radio button per l'apertura e chiusura della pompa.

La comunicazione con arduino è bidirezionale, ovvero entrambi invieranno e riceveranno dei dati.

Ad ogni click dei pulsanti l'app invierà ad arduino una stringa secondo il protocollo di comunicazione da noi creato. Un'altra importante funzionalità è quella di ricevere informazioni relative alla percentuale di umidità.

## Microcontrollore

Il microcontrollore arduino viene impiegato per mantenere la logica di funzionamento degli attuatori e sensori della serra ad esclusione del sensore di umidità. L'arduino ha il compito di verificare l'eventuale connessione di un utente attraverso il sensore di distanza, in caso di utente sufficientemente vicino questo viene considerato connesso tramite bluetooth. Lo stato di connessione e di non connessione rappresentano i primi due stati della macchina e le loro meccaniche sono incapsulate nella classe TaskSearch.

Compito del microcontrollore è anche quello di gestire e interpretare i messaggi che gli arrivano sia dal server tramite seriale sia dalla Mobile App tramite bluetooth. Per fare ciò è presente la classe TaskComunicate che, con riferimento allo stato della macchina presente nella classe globale, gestisce e valuta i messaggi inerenti allo stato del controllore azionando in maniera coerente la pompa di irrigazione con lo stato di controllo del sistema (Manuale o Automatico)

## Protocollo di comunicazione

	invio di messaggi verso bluetooth
xx-a	xx percentuale di umidità nell'aria - modalità automatica
xx-m	xx percentuale di umidità nell'aria - modalità manuale
	invio di messaggi verso server
o	pompa aperta (open)
c	pompa chiusa (close)
0	modalità automatica
1	modalità manuale

	ricezione di messaggi
l	portata bassa (low)
m	portata media (medium)
h	portata alta (high)
z	portata nulla (zero)
Hxx	umidità percentuale = xx (humidity)

## Edge ESP8266

Il modulo ESP ha il solo scopo di misurare gli indici di umidità del terreno, tramite un modulo DHT11 (Nel codice implementato, tutto ciò che concerne questo modulo è commentato, poiché al suo posto, per semplice comodità, viene utilizzato un potenziometro, in questo modo siamo in grado di regolare manualmente l'umidità per svolgere le dovute analisi), ed inviarle al server.

Infatti l'edge misura l'umidità ogni 10 secondi, incapsulando questa informazione in un file Json che, all'interno di un pacchetto HTTP, viene inviata al server.

L'invio del pacchetto avviene tramite il software "Ngrok", questo programma fornisce un indirizzo verso il quale, il microcontrollore, può facilmente comunicare; il server e l'ESP dovranno però essere connessi alla stessa rete wifi. Ngrok si occuperà di fare 'tunneling', ascoltando i messaggi in arrivo all'indirizzo da lui messo a disposizione, verso una determinata porta del localhost.

## Java Server (PC)

Il server contiene la logica che gestisce Arduino quando è nello stato “Automatico”; tramite la comunicazione seriale infatti è possibile inviare i messaggi (rispettando il protocollo succitato) per controllare il livello dell’irrigazione, aprire/chudere la pompa e per inviare le informazioni sul livello di umidità (che poi il microcontrollore si invierà all’eventuale utente connesso tramite la mobile App).

Il server salva inoltre le informazioni relative all’umidità, all’apertura e alla chiusura della pompa, in un file testuale che può essere visualizzato sul browser tramite una semplice pagina php da noi creata.

La comunicazione con l’edge ESP avviene tramite protocollo IP/HTTP. Infatti il server rimane in ascolto sulla porta 8085, dove il programma Ngrok permette di fare tunneling. La classe DataService, che utilizza la libreria “Vertx”, è quella che riceve ed analizza i pacchetti inviatogli da ESP.

Il server implementa un pattern observable, dove le classi “osservabili” generano un evento che viene inserito all’interno di una lista dove, per ogni elemento, la classe “Observer” (BasicEventLoopController) li processa.