

# Progetto Embedded Systems and IoT

Davide Carità - 0000873616

---

<b>Definizione dei requisiti</b>	<b>1</b>
<b>Progettazione del sistema</b>	<b>3</b>
<b>Implementazione del sistema</b>	<b>4</b>
ModeManagerTask	5
WaterControlTask	6
CoolingControlTask	6

---

## Definizione dei requisiti

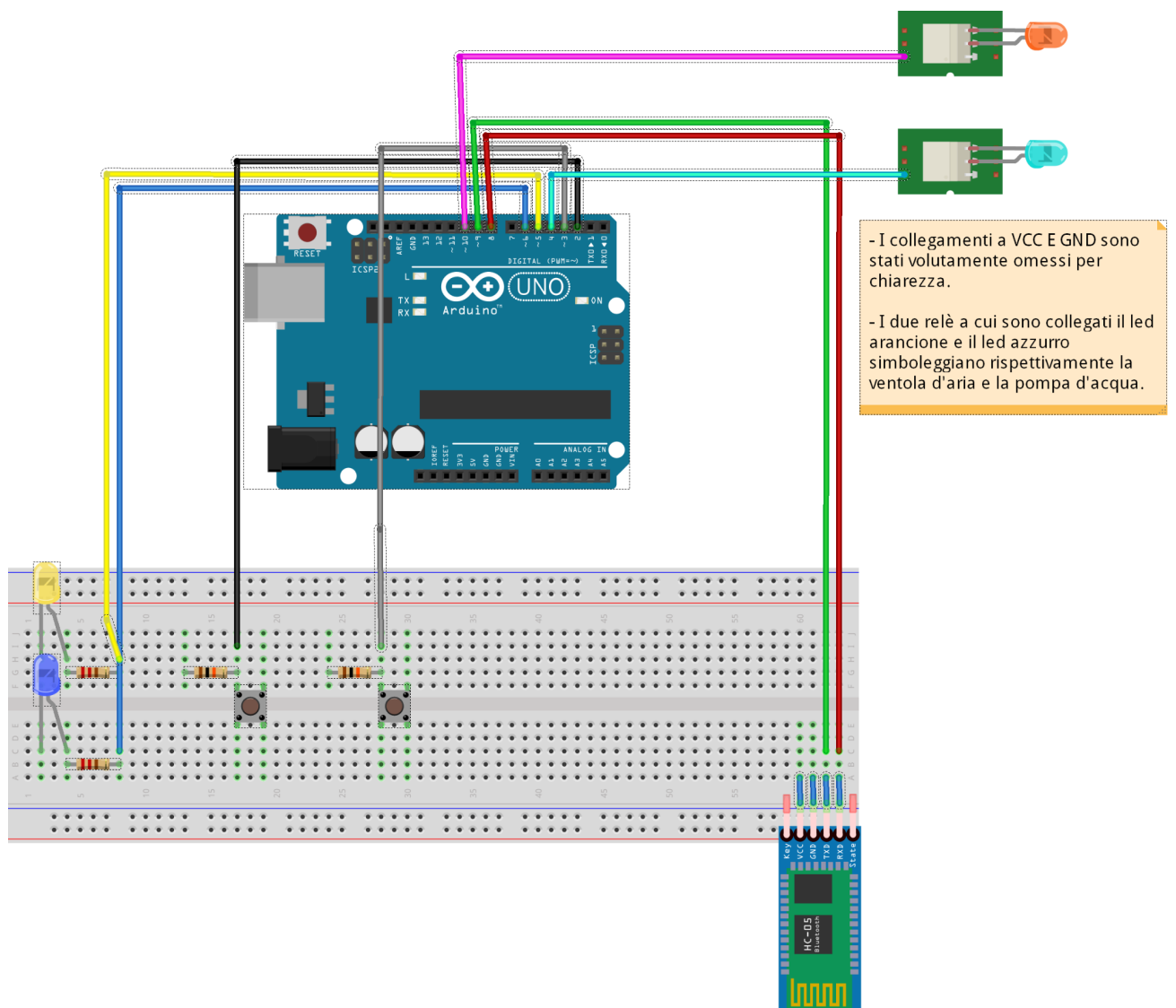
Lo scopo del progetto è quello di realizzare un sistema embedded integrato che possa monitorare, controllare e mantenere stabili i parametri ambientali di una serra contenente una pianta.

Il sistema avrà la necessità di essere autonomo, attraverso una modalità automatica e allo stesso tempo dovrà essere controllabile manualmente.

Il terreno in cui è la pianta non dovrà mai sfiorare dei cut-off prestabiliti di umidità e temperatura e per fare ciò occorreranno dei sensori di umidità del terreno e un sensore di temperatura che verranno posti rispettivamente all'interno del vaso e in prossimità della pianta.

Oltre che ai sensori, per poter effettivamente controllare l'umidità e la temperatura, sarà installata una pompa d'acqua per poter irrigare la pianta in caso di umidità del terreno troppo bassa, e una ventola, che permetterà alla serra di avere una maggiore ventilazione e ricambio d'aria per poter abbassare la temperatura.

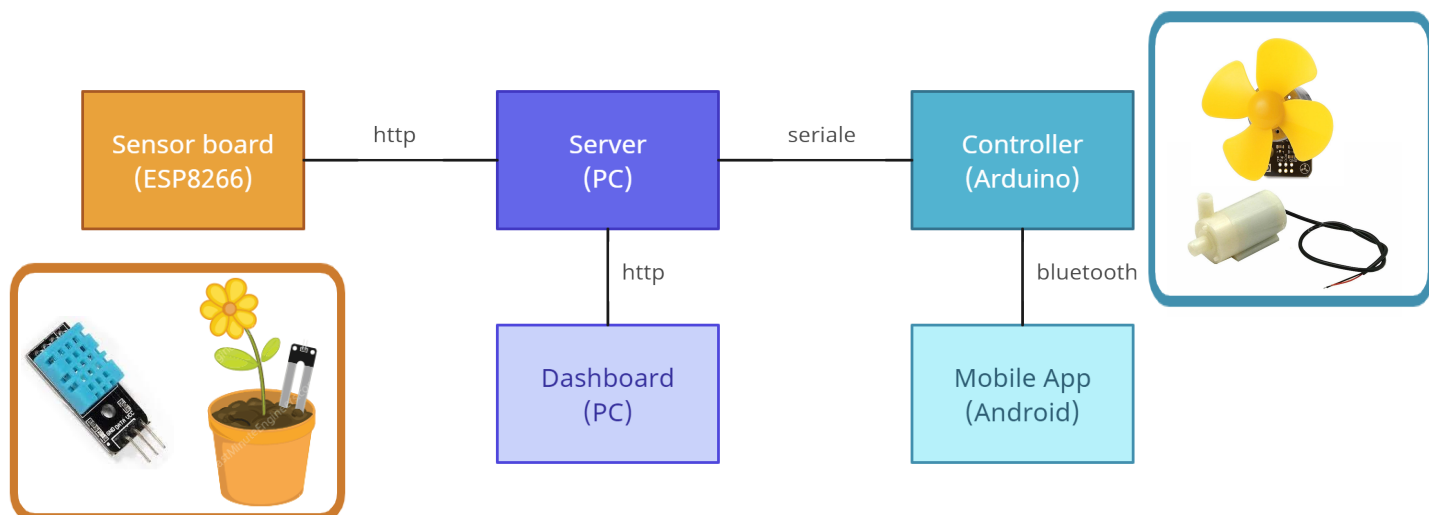
[https://github.com/DavideBlast/elab\\_iot](https://github.com/DavideBlast/elab_iot)



fritzing

# Progettazione del sistema

Per soddisfare la necessità di una modalità automatica e una manuale, il sistema partirà e funzionerà costantemente in modalità **AUTOMATICA**, ma consentirà tramite bottoni presenti sulla breadboard di controllare manualmente la pompa d'acqua e la ventola d'aria. Inoltre tramite l'app **GreenhouseBT** si avrà un secondo modo per poter accedere al controllo di pompa e ventola.



Il sistema è suddiviso in cinque sottosistemi:

- **Sensor board:**
  - Basato su ESP32 invia dati a **Server** via HTTP, possiede due sensori, il primo è un **DHT11** posizionato in prossimità della pianta e permetterà di rilevare la temperatura e l'umidità relativa all'interno della serra; il secondo è un **Resistive Soil Moisture Sensor**, che, una volta inserito nel vaso in cui è la pianta, rileva la resistenza elettrica del terreno, ciò vuol dire che un valore più basso indica un quantitativo minore d'acqua, in quanto l'acqua è un ottimo conduttore elettrico.
- **Server:**
  - In esecuzione su PC, contiene il server locale a cui **Sensor board** invierà le richieste HTTP e la logica per poter inviare i dati a via **Controller** Seriale.
- **Dashboard:**
  - Anche questo su PC, basato su GUI java, fornisce una visualizzazione dello stato del sistema generale e un logger degli eventuali messaggi scambiati con **Controller**.
- **Controller:**
  - Basato su Arduino, consente di azionare la **ventola** per il controllo della temperatura e la **pompa d'acqua** per la gestione dell'umidità del terreno. Possiede due bottoni che permettono la gestione manuale e un modulo HC05 per la ricezione di messaggi via Bluetooth da **Mobile App**.

- **Mobile App:**

- Su Android, permette di controllare la pompa d'acqua e la ventola d'aria di **Controller.**

## Implementazione del sistema

### Componenti fisici del sistema:

#### **Controller:**

- PIN 2: BTN\_FAN: bottone tattile che ha come scopo quello di azionare manualmente la ventola d'aria.
- PIN 3: BTN\_WATER: bottone tattile che ha come scopo quello di azionare manualmente la pompa d'acqua.
- PIN 6: LED\_WATER\_MODE: led blu che indica che il sistema è in modalità "WATERING"
- PIN 5: LED\_COOL\_MODE: led giallo che il sistema è in modalità "COOLING"
- PIN 4: WATER\_PUMP: pompa d'acqua controllata da un relè 5V a canale singolo
- PIN 11/12/13: COOLING\_FAN: ventola per il controllo della temperatura, basata su motore DC.

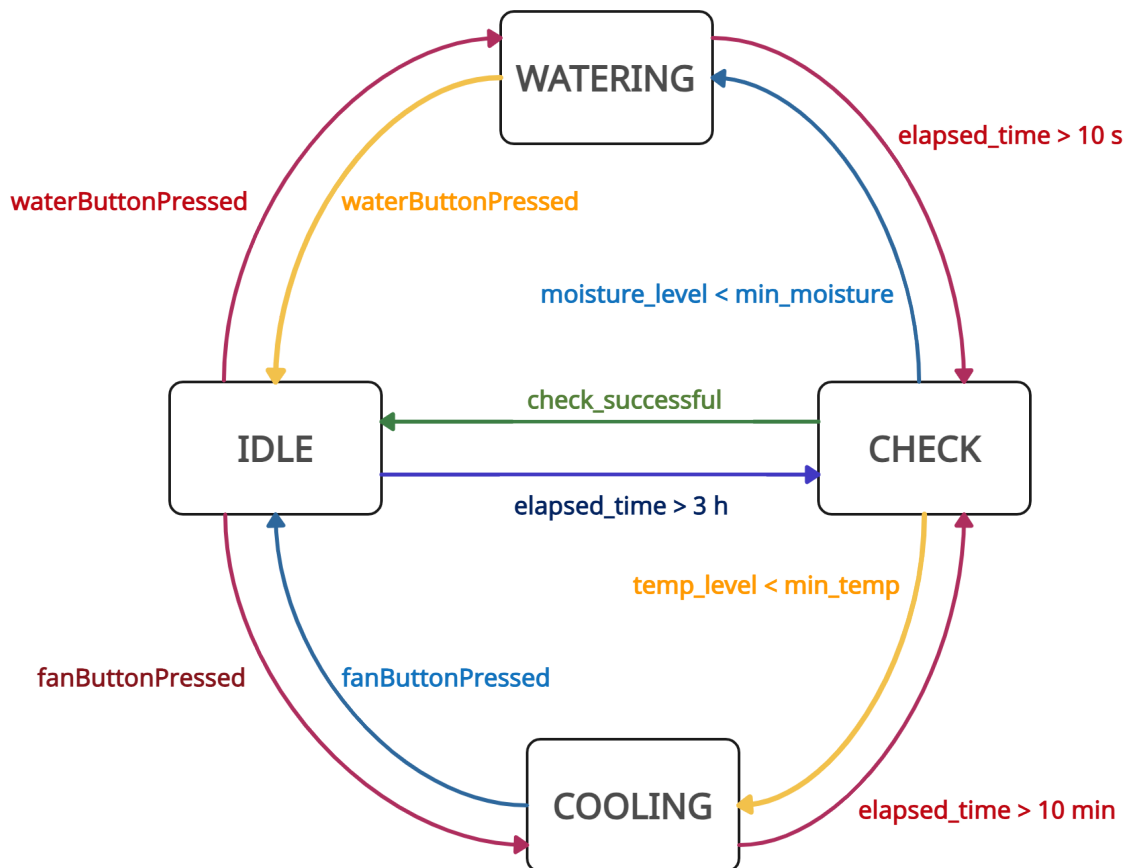
#### **Sensor board:**

- PIN 36: SOIL\_MOISTURE\_SENSOR: sensore resistivo dell'umidità del terreno
- PIN 26: TEMPERATURE\_SENSOR: sensore DHT11 per la temperatura e l'umidità relativa esterna

Il sottosistema **Controller** è stato sviluppato secondo un principio basato su task e di macchina a stati finiti sincrona. I task principali che compongono il sistema sono tre:

## ModeManagerTask

Task che gestisce la modalità di operazione del sistema e permette di cambiarla.



Il sistema parte in modalità IDLE, successivamente, questo sono le possibilità:

IDLE:

- Se è passato il tempo IDLE\_TIME, va in modalità CHECK
- Se è stato premuto il tasto BTN\_WATER o si riceve un segnale bluetooth che dice di accendere la pompa d'acqua, va in modalità WATERING
- Se è stato premuto il tasto BTN\_FAN o si riceve un segnale bluetooth che dice di accendere la ventola d'aria, va in modalità COOLING

CHECK:

- Se i parametri di temperatura e umidità del terreno rientrano nei threshold prestabiliti, va in modalità IDLE
- Se la temperatura è troppo alta, va in modalità COOLING
- Se l'umidità del terreno è troppo bassa, va in modalità WATERING

WATERING:

- Se è passato il tempo WATERING\_TIME, va in modalità CHECK
- Se è stato premuto il tasto BTN\_WATER o si riceve un segnale bluetooth che dice di spegnere la pompa d'acqua, va in modalità IDLE

COOLING:

- Se è passato il tempo COOLING\_TIME, va in modalità CHECK
- Se è stato premuto il tasto BTN\_FAN o si riceve un segnale bluetooth che dice di spegnere la ventola d'aria, va in modalità IDLE

## WaterControlTask

Task che controlla l'azionamento della pompa d'acqua e di LED\_WATER\_MODE.

Presenta due semplici stati: ON e OFF.

Quando è in modalità ON, si assicura che la pompa d'acqua sia accesa e che il led blu LED\_WATER\_MODE lampeggi. Al contrario, quando in OFF, spegne la pompa d'acqua e spegne anche il led.

## CoolingControlTask

Task che controlla l'azionamento della ventola d'aria e di LED\_COOL\_MODE.

Presenta due semplici stati: ON e OFF.

Quando è in modalità ON, si assicura che la ventola d'aria sia accesa e che il led giallo LED\_COOL\_MODE lampeggi. Al contrario, quando in OFF, spegne la ventola d'aria e spegne anche il led.

Oltre ai task, il file **Greenhouse.cpp** gestisce il model della serra smart, tenendo traccia del suo stato e dei controlli dei componenti. **UserInput.cpp** gestisce i bottoni tattili sulla breadboard e **BTHandler.cpp** la comunicazione bluetooth con **Mobile App**

Il sottosistema **Sensor board** è invece un semplice programma in C++/Arduino che rileva la temperatura e l'umidità del terreno e invia richieste HTTP-POST al server locale contenuto in **Server**.

Il sottosistema **Server/Dashboard** possiede MonitoringAgent.java che permette di aggiornare la view a seconda del messaggio ricevuto via seriale.

GreenhouseView.java che crea la view riguardante lo stato generale del sistema, della pompa d'acqua e della ventola d'aria. LogView.java, che mostra i messaggi ricevuti via seriale da **Controller**.

Il sottosistema **Mobile App** è una semplice app Android chiamata **GreenhouseBT** che permette di inviare 4 messaggi: "won", "woff", "fon", "foff" che servono rispettivamente per accendere e spegnere la pompa d'acqua, accendere e spegnere la ventola d'aria. Questi verranno letti dal modulo HC-05 presente nel **Controller**.