

# Smart Greenhouse

Student: Ciobanu Daria

Grupa: 332CC

Proiectul constă în realizarea unei mini-sere inteligente cu un sistem automat de irigare și ventilație determinat de factorii de mediu (umiditatea solului, respectiv temperatură), precum și alte două funcționalități extra care facilitează dezvoltarea armonioasă a plantelor. Serea va avea un ecran pe care se va afișa starea curentă a plantelor, dar și un buton pentru pornirea unei melodii ce stimulează creșterea florilor.

## Introducere

Dacă te-ai găsit în situația să nu îți amintești când ai udat ultima dată plantele din casă sau, oricât ai încercat să ai grijă de o floare, aceasta tot nu pare că se simte bine în habitatul ei, o seră inteligentă este ceea ce îți trebuie. Serea utilizează un senzor de umiditate pentru a evalua când trebuie udat solul, declanșând ulterior pompa de apă ce facilitează hidratarea plantelor. Există și un senzor de temperatură care pornește un mic ventilator în interiorul serei pentru a răcori atmosfera. Pentru a face plantele și mai fericite am adăugat un difuzor ce pornește o melodie din muzica clasică, aceasta având un efect benefic asupra creșterii lor. Toți acești factori contribuie la starea de sănătate a plantelor, care va fi prezentată pe un display în exteriorul serei.

## Descriere generală



## Hardware Design

### Lista pieselor:

- Arduino UNO
- breadboard
- senzor pentru temperatură DHT
- senzor pentru umiditatea solului
- pompă de apă
- buton
- mini-ventilator

- modul I2C
- LCD 1602
- difuzor

## Schema electrică:



## Software Design

**Mediu de dezvoltare:** Arduino IDE

### Biblioteci folosite:

1. <LiquidCrystal\_I2C.h>
  - bibliotecă folosită pentru comunicarea cu ecranul LCD prin intermediul I2C
2. <Wire.h>
  - bibliotecă folosită pentru a scrie pe LCD
3. <dht.h>
  - bibliotecă folosită pentru comunicarea cu senzorul de temperatură
4. "pitches.h"
  - bibliotecă în care am inclus notele muzicale
5. <TimeLib.h>
  - bibliotecă folosită pentru afișarea timpului

## Descrierea codului

### 1. **setup()**

- Am inițializat componentele hardware utilizând pinii aleși
- Am configurat o intrerupere folosind funcția attachInterrupt() având ca parametru pinul butonului ce va declanșa melodia

### 2. **loop()**

- Citesc valorile primite de la senzorul de umiditate a solului (pinul A0) și senzorul de temperatură pentru a determina parametrii în funcție de care se vor declanșa funcțiile serei
- Am printat aceste valori în interfața serială pentru a urmări mai ușor schimbările de mediu

```
int value = analogRead(A0);
Serial.println(value);
int chk = DHT.read11(DHT11_PIN);
Serial.print("Temperature = ");
Serial.println(DHT.temperature);
Serial.print("Humidity = ");
Serial.println(DHT.humidity);
delay(1000);
```

- Temperatura determinata de senzorul DHT pornește sau oprește ventilatorul

```
if(DHT.temperature > 20) {
  digitalWrite(FAN_PIN, HIGH);
} else {
  digitalWrite(FAN_PIN, LOW);
}
```

- Valorile citite de la senzorul de umiditate determină pornirea pompei de apă și schimbarea mesajului de pe display
- Am delimitat 4 stări:

```
// 1. Pamantul e prea umed pentru starea optima a plantei
if(value < 300) {
  digitalWrite(WATER_PIN, LOW);
  lcd.setCursor(0, 0);
  lcd.print("Too much water!!!");
  lcd.setCursor(0, 1);
  lcd.print("I'm drowning...");
}
// 2. Conditii optime pentru dezvoltarea plantelor
else if (value > 300 && value < 850) {
  digitalWrite(WATER_PIN, LOW);
  lcd.setCursor(0, 0);
  lcd.print("I'm happy <3  ");
  lcd.setCursor(0, 1);
  lcd.print("Perfect soil  ");
} else {
// 3. Pamant uscat - se porneste pompa de apa pentru a uda plantele
  digitalWrite(WATER_PIN, HIGH);
  lcd.setCursor(0, 0);
  lcd.print("Drinking water...");
  lcd.setCursor(0, 1);
  lcd.print("Pretty dry here :( ");
// 4. Pamantul este foarte uscat, deci pompa nu mai functioneaza
// => nu exista apa in rezervor: se va porni o alarma in acest caz
if(value > 1010) {
  for (int thisNote = 0; thisNote < notes2; thisNote = thisNote + 2) {
    // alarm for extreme soil dryness
    // ...
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Refill the tank!!!");
}
}
```

- Butonul care declanșează muzica atunci când posesorul serei dorește este activat prin folosirea unei întreruperi
- Am implementat funcția **playMusic()** pentru a structura mai bine codul, folosind laboratorul 3 pentru redarea notelor melodiei

```
if (interruptCount > 0) {
```

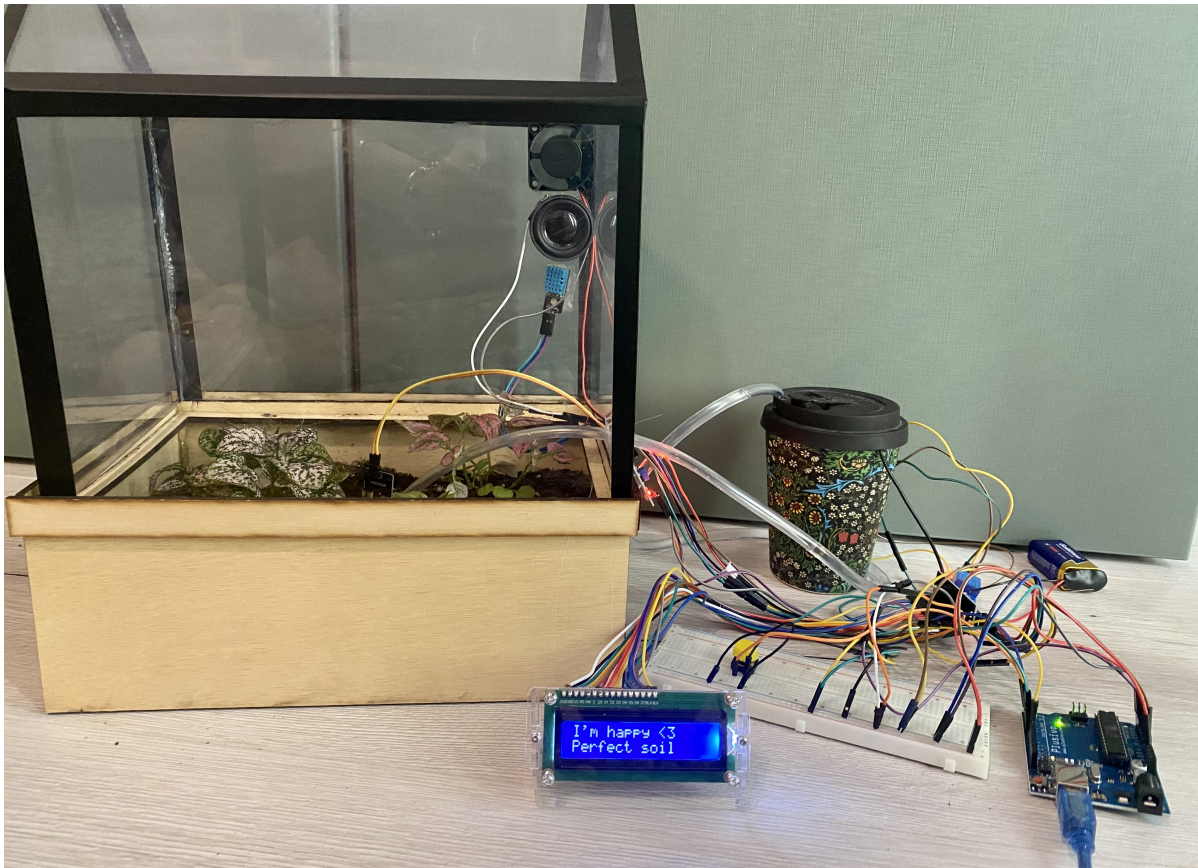
```
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Cool music :) ");
    lcd.setCursor(0, 1);
    lcd.print("Shake it off ");
    playMusic();
    interruptCount = 0;
}
```

- Pentru ca plantele să se bucure de muzică și în absența unei persoane, difuzorul va porni o dată pe zi
- Am implementat un timer care apelează funcția playMusic() când funcția millis() va ajunge la intervalul cerut (86400s),

și am afișat pe LCD timpul trecut de la pornirea sistemului serei

```
const unsigned long interval = 86400000;
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    digitalClockDisplay();
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("Music o'clock ");
    playMusic();
}
void digitalClockDisplay() {
    lcd.clear();
    lcd.setCursor(0, 0);
    printDigits(hour());
    lcd.print(":");
    printDigits(minute());
    lcd.print(":");
    printDigits(second());
}
```

## Rezultate Obținute



Demo:<https://drive.google.com/drive/folders/1kLPGIZ-ddIWddBkMJQOYuobudT4AaleN?usp=sharing>

## Concluzii

Realizarea acestui proiect mi-a oferit o înțelegere mai profundă a materiei și a aspectelor practice ale acesteia, în special cele legate de interconectarea și utilizarea componentelor hardware. Astfel, am reușit să creez un produs funcțional și util, îmbinând ideea proiectului cu pasiunea mea pentru plante. În timpul procesului, am întâmpinat mici probleme, cum ar fi sincronizarea LCD-ului cu informațiile primite de la senzor, dar și implementarea întreruperilor, unde am fost nevoită să adaug un delay pentru a elimina efectul de debouncing. Am depășit provocările întâmpinate și am obținut o implementare eficientă și un produs fizic pe care îl voi utiliza cu bucurie pe viitor.

## Download

Drive:<https://drive.google.com/drive/folders/1kLPGIZ-ddIWddBkMJQOYuobudT4AaleN?usp=sharing>

Zip:[smart\\_green.zip](#)

## Jurnal

- 05.05.2023 - Milestone proiect - documentație: alegerea temei proiectului și a funcționalităților de bază
- 18.05.2023 - Milestone - hardware: conectarea elementelor hardware și realizarea schemei electrice
- 26.05.2023 - Milestone - software: implementarea codului și finalizarea proiectului

## Bibliografie/Resurse

### Resurse Software

- <https://github.com/robsoncouto/arduino-songs>
- <https://projecthub.arduino.cc/plouc68000/simplest-uno-digital-clock-ever-03c185>
- [https://www.youtube.com/watch?v=7KdhL\\_gAqHQ&list=LL&index=4&ab\\_channel=PassionTechKLM](https://www.youtube.com/watch?v=7KdhL_gAqHQ&list=LL&index=4&ab_channel=PassionTechKLM)

### Resurse Hardware

- <https://mytectutor.com/soil-moisture-sensor-with-arduino-automatic-plant-irrigation-system/>

[Export to PDF](#)

From:  
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2023/dene/smartgreenhouse>



Last update: **2023/05/29 14:31**