

# Monitorizare plante

- Arpășanu Emilia-Oana, 331CA

Cu siguranță mai există cel puțin o persoană care este înconjurată de mici (sau mari) prietene necuvântătoare verzi, cu rădăcină, tulpină, frunze și chiar și flori. De cele mai multe ori, acestea sunt parcă uitate de lume: le neglijăm gândindu-ne că dacă nu observăm ceva greșit, ele sunt în regulă. Timpul trece și putem constata la un moment dat că locul în care au stat nu le-a priit și că doar ni s-a părut că le-am udat când a trebuit. Așadar, dacă ele ne încântă privirile și ne dau la schimb oxigen în locul dioxidului de carbon, de ce nu le-am putea oferi înapoi măcar puțină grijă? Mai ales că, în plină eră a tehnologiei și a microprocesoarelor, putem identifica soluții practice și eficiente care să ne ajute.

## Introducere

Proiectul are ca scop citirea multiplilor parametri ce afectează în mod direct starea unei plante (umiditate sol și aer, cantitatea de lumină, temperatura ambientală) pentru a determina mai apoi următoarele acțiuni necesare menite îngrijirii. Informațiile sunt afișate utilizatorului prin intermediul unui ecran LCD. Pentru a menține un nivel al umidității solului în parametrii normali am inclus o pompă de apă ce trage apa dintr-un rezervor și umezește solul ghiveciului.

## Descriere generală

Pentru a determina parametrii descriși mai sus se utilizează mai mulți senzori, selectarea tipului de informație afișat utilizatorului având loc cu ajutorul unui buton și a întreruperilor. Tipul informațiilor este semnalizat de un led RGB ce se aprinde cu o culoare specifică. Pompa de apă pornește (și trage apa pentru o perioadă finită de timp) dacă gradul de umiditate scade sub un anumit prag, acesta fiind calculat frecvent prin implementarea unui timer (o dată la 4 secunde).



## Hardware Design

Listă de componente necesare:

- Arduino Uno R3;
- breadboard;
- fire de legătură;
- modul releu 5V (JQC-3FF-S-Z);
- mini pompă de apă submersibilă (funcționează cu o alimentare de 5V);

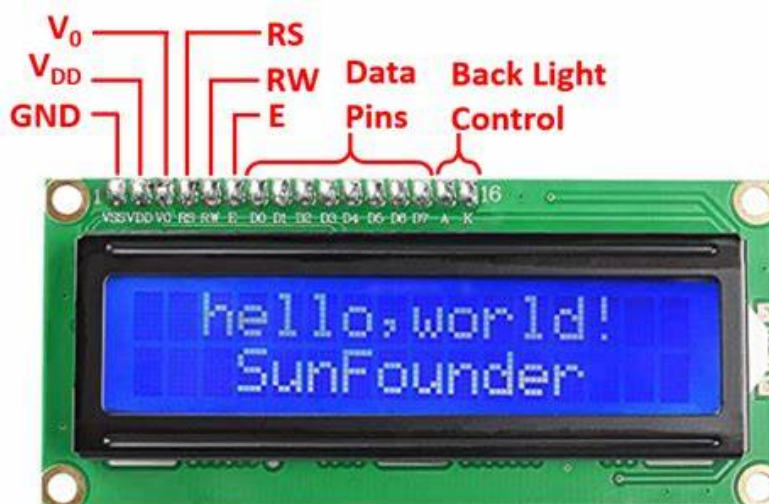
- furtun de plastic;
- recipient pentru apă;
- modul cu senzor de umiditate a solului (LM393) și sondă pentru introducerea în sol (FC-28);
- fotorezistență;
- modul DHT11 (temperatură + umiditate aer);
- modul ecran LCD 1602;
- led RGB;
- buton;
- rezistențe de pull-up (10KΩ și 220Ω);
- sursă alimentare (baterie 9V pentru Arduino, breadboard, senzori, releu și led și carcasă cu 4 baterii AA pentru pompă);
- și nu în ultimul rând, planta de analizat (cu tot cu ghiveci).

Conectarea componentelor arată cam așa (schema electrică a fost realizată cu ajutorul Tinkercad):

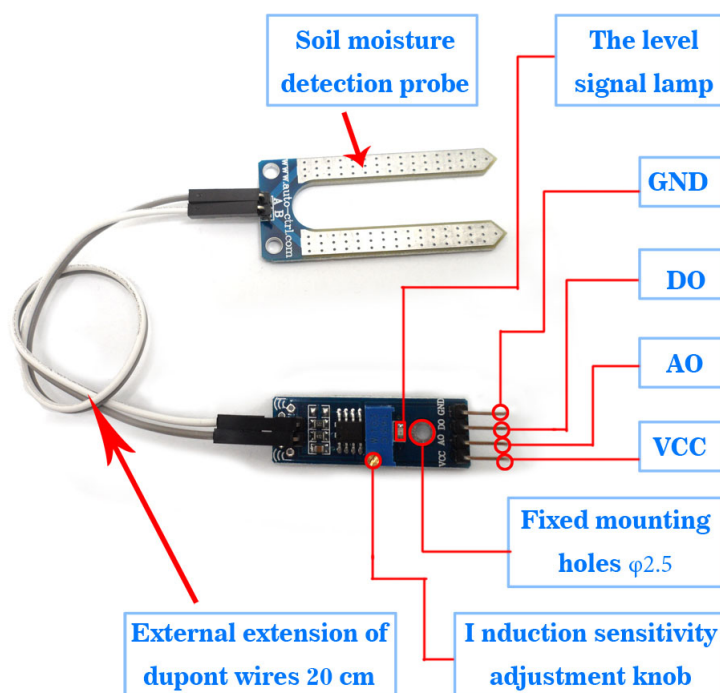


Pinii au fost conectați astfel:

- Pentru LCD:
  - GND → GND breadboard
  - VDD → VCC breadboard
  - V0 → potențiometru (pentru contrast)
  - Read/Write → GND breadboard (afișarea se face prin intermediul pinilor de date)
  - Register Select → D7 Arduino
  - Enable → D8 Arduino
  - Data Pin 0 → aer
  - Data Pin 1 → aer
  - Data Pin 2 → aer
  - Data Pin 3 → aer
  - Data Pin 4 → ~D9 Arduino
  - Data Pin 5 → ~D10 Arduino
  - Data Pin 6 → ~D11 Arduino
  - Data Pin 7 → D12 Arduino
  - Anod → VCC breadboard
  - Catod → GND breadboard



- Pentru DHT11:
  - VCC → VCC breadboard
  - GND → GND breadboard
  - IN → D4 (Arduino), rezistență pull-up (10K $\Omega$ )
- Pentru senzorul de umiditate a solului:
  - VCC → VCC breadboard
  - GND → GND breadboard
  - A0 (output analogic) → A5 Arduino
  - D0 (output digital) → aer (se folosește valoarea analogică citită)
  - + → A (sondă sol)
  - - → B (sondă sol)

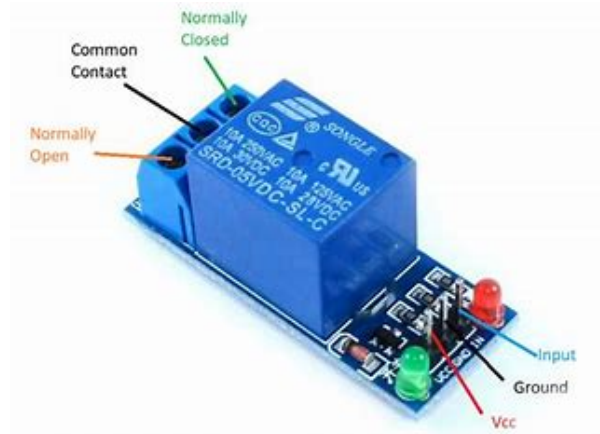


- Pentru led-ul RGB:
  - canal roșu → A3 Arduino, rezistență pull-up 220 $\Omega$
  - canal verde → A2 Arduino, rezistență pull-up 220 $\Omega$
  - canal albastru → A1 Arduino, rezistență pull-up 220 $\Omega$
  - catod → GND breadboard (led-ul este de tip catod comun; altfel, dacă acesta era de tip anod comun pin-ul s-ar fi conectat la VCC)
- Pentru buton:
  - VCC → rezistență pull-up (10K $\Omega$ ) legată și la VCC breadboard
  - Data Output → D2 Arduino (care primește semnalul pentru întreruperea externă implementată)
  - GND → breadboard



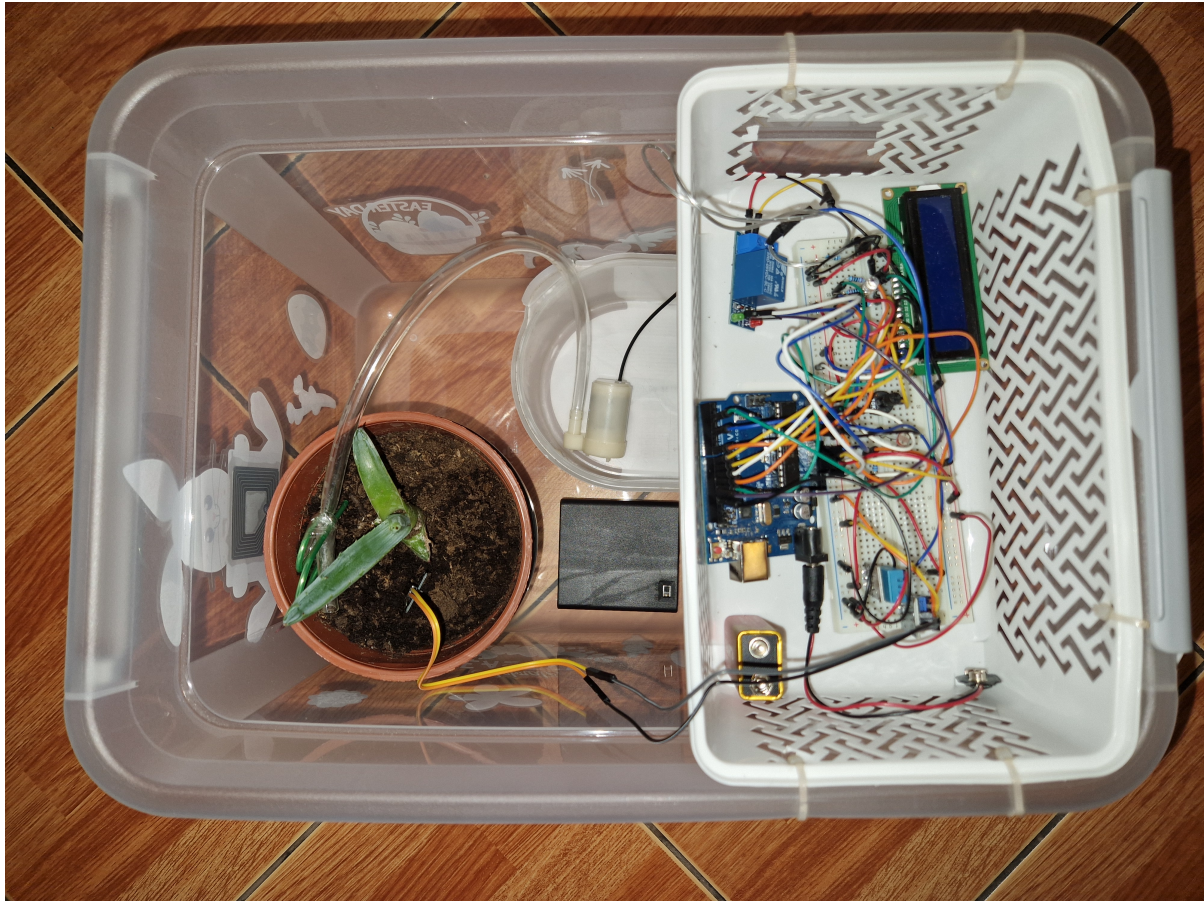
- Pentru potențiometrul:
  - GND → GND breadboard
  - VCC → VCC breadboard

- Wiper (setare valoare rezistență) → V0 LCD
- Pentru releu:
  - NC (normally closed) → pompă
  - COM (common contact) → + carcasă baterii
  - VCC → VCC breadboard
  - GND → GND breadboard
  - IN → D13 Arduino (control curent ce ajunge la pompă)



- Pentru pompă:
  - + → NC releu
  - - → GND breadboard
- Pentru carcasa de baterii:
  - + → COM releu
  - - → GND breadboard
- Pentru fotorezistență:
  - OUT → A0 Arduino, rezistență pull-up (10K $\Omega$ ) legată și la GND breadboard
  - VCC → VCC breadboard

În final, proiectul arată așa:



## Software Design

Am dezvoltat programul ce a fost uploadat în cadrul Arduino IDE, fiind familiarizată cu acesta în urma proiectului de la DEEA (are o interfață mai prietenoasă și mai ușor de utilizat). În cadrul acestuia am inclus două biblioteci pentru o gestionare facilă a componentelor:

1. LiquidCrystal.h (preinstalată odată cu IDE-ul), pentru alocarea pinilor fizici și afișarea informațiilor;
2. DHT.h (DHT sensor library by Adafruit), pentru citirea umidității și a temperaturii ambientale.

Programul este compus în principal din 3 logici distincte:

- timer-ul, ce din 4 în 4 secunde citește gradul de umiditate al solului și pornește/oprește pompa cu ajutorul releului;

```
ISR (TIMER1_COMPA_vect)
{
  // citire valoare de la senzorul de umiditate al solului (pin analogic)
  int val = analogRead(moisturePin);

  // convertire valoare citita in procentaj
  int moisture = (100 - ((val / 1023.00) * 100));

  // valoare de prag a umiditatii
  if (moisture < 40)
  {
```

```

    digitalWrite(motorPin, HIGH); // pornire pompa
    digitalWrite(blinkPin, HIGH); // aprindere led de la senzorul de
umiditate al solului
}
else
{
    digitalWrite(motorPin, LOW); // oprire pompa
    digitalWrite(blinkPin, LOW); // stingere led de la senzorul de
umiditate al solului

    // deoarece nu as vrea sa se afiseze nimic in timpul udarii plantei,
ma asigur ca
    // dupa oprire pentru putin timp nu se afiseaza nimic in continuare
    lcd.clear();
    delay(1000);
}
}
}

```

- debouncer-ul, pentru detecția apăsării corecte a butonului (în cadrul întreruperii);

```

void buttonPressed()
{
    // daca semnalul s-a stabilizat la actionarea butonului,
    // se actualizeaza contorul ce numara apasarile si ultimul timestamp al
apasarii
    if (millis() - lastDebounceTime > debounceDelay)
    {
        pressCounter++;
        lastDebounceTime = millis();
    }
}

```

- bucla de funcționare, care comută între tipurile de informație de afișat și care asigură faptul că nu se afișează nimic în timpul udării solului (pentru o claritate mai bună a prelucrării datelor - să aibă loc când ansamblul nu suferă modificări).

```

void loop()
{
    // verific daca pompa este pornita
    int val = digitalRead(motorPin);

    // daca pompa inca este pornita, ma asigur ca nu afisez nimic
    if(val == HIGH)
    {
        lcd.noDisplay();
        lcd.clear();
    }
    else // daca pompa s-a oprit, revin la afisarea informatiilor
    {
        lcd.display();
        // cursor stanga-sus
    }
}

```

```
lcd.home();

// actualizare corecta a selectiei
if (sel != pressCounter)
    sel = pressCounter;

if(sel >= 3) // pentru a avea selectie circulara
{
    sel = 0;
    pressCounter = 0;
}

if(!sel)
{
    // se afiseaza umiditatea si temperatura ambientala
    // ...
}

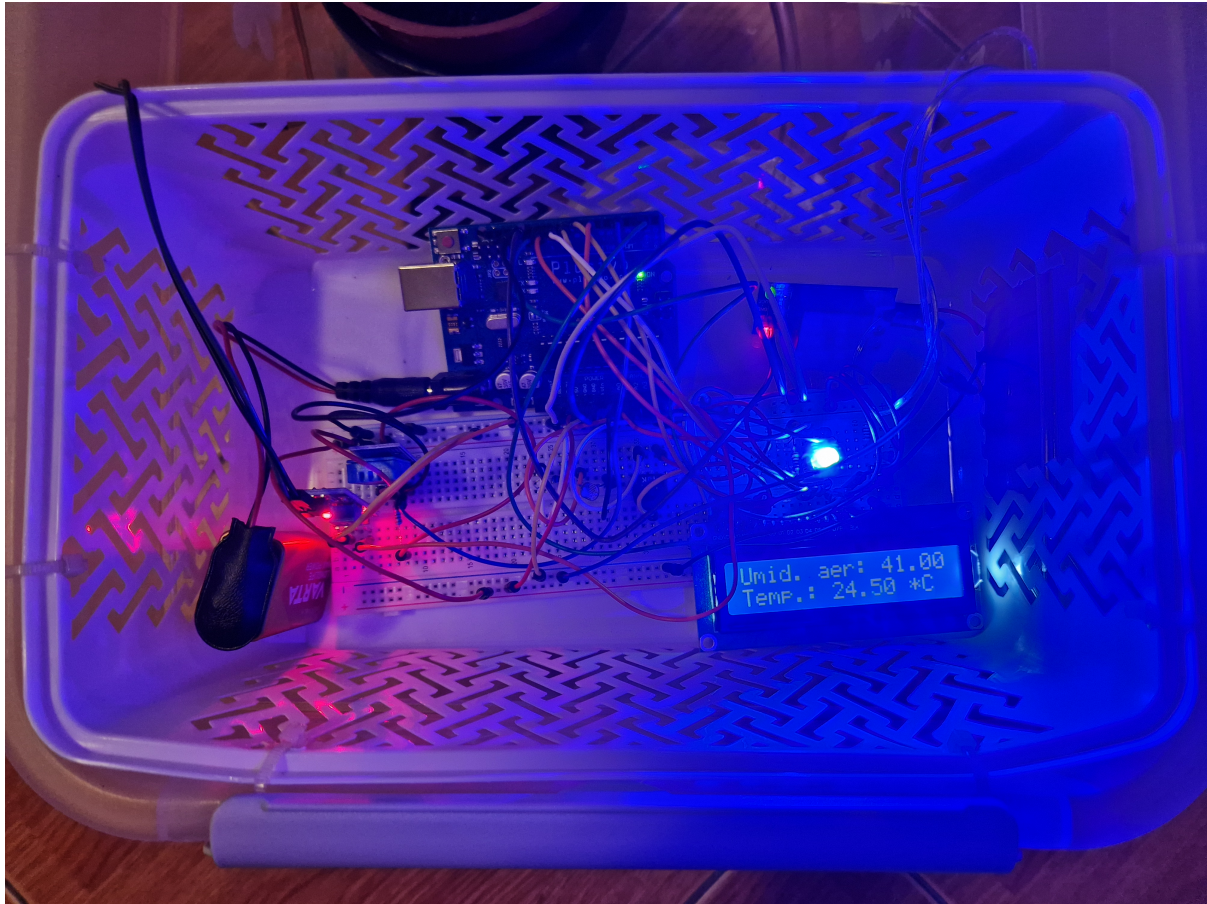
if(sel == 1)
{
    // se afiseaza valoare intensitatii luminoase si caracterizarea
acesteia
    // (intuneric/lumina slaba/normala/tare/ foarte tare)
    // ...
}

if(sel == 2)
{
    // se afiseaza gradul de umiditate al solului si caracterizarea
acesteia (normala/scazuta)
    // ...
}
}
```

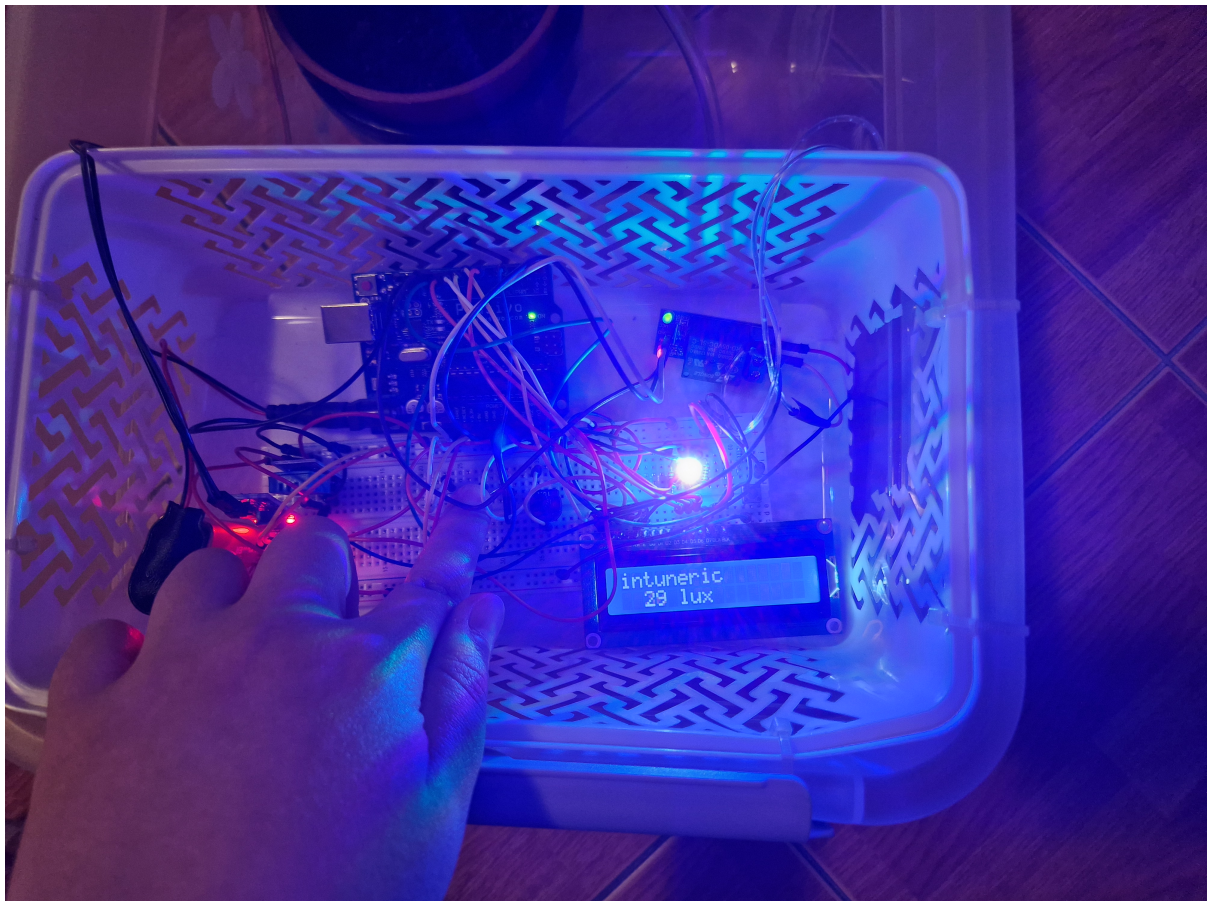
## Rezultate Obținute

În urma pornirii circuitului se pot observa următoarele rezultate:

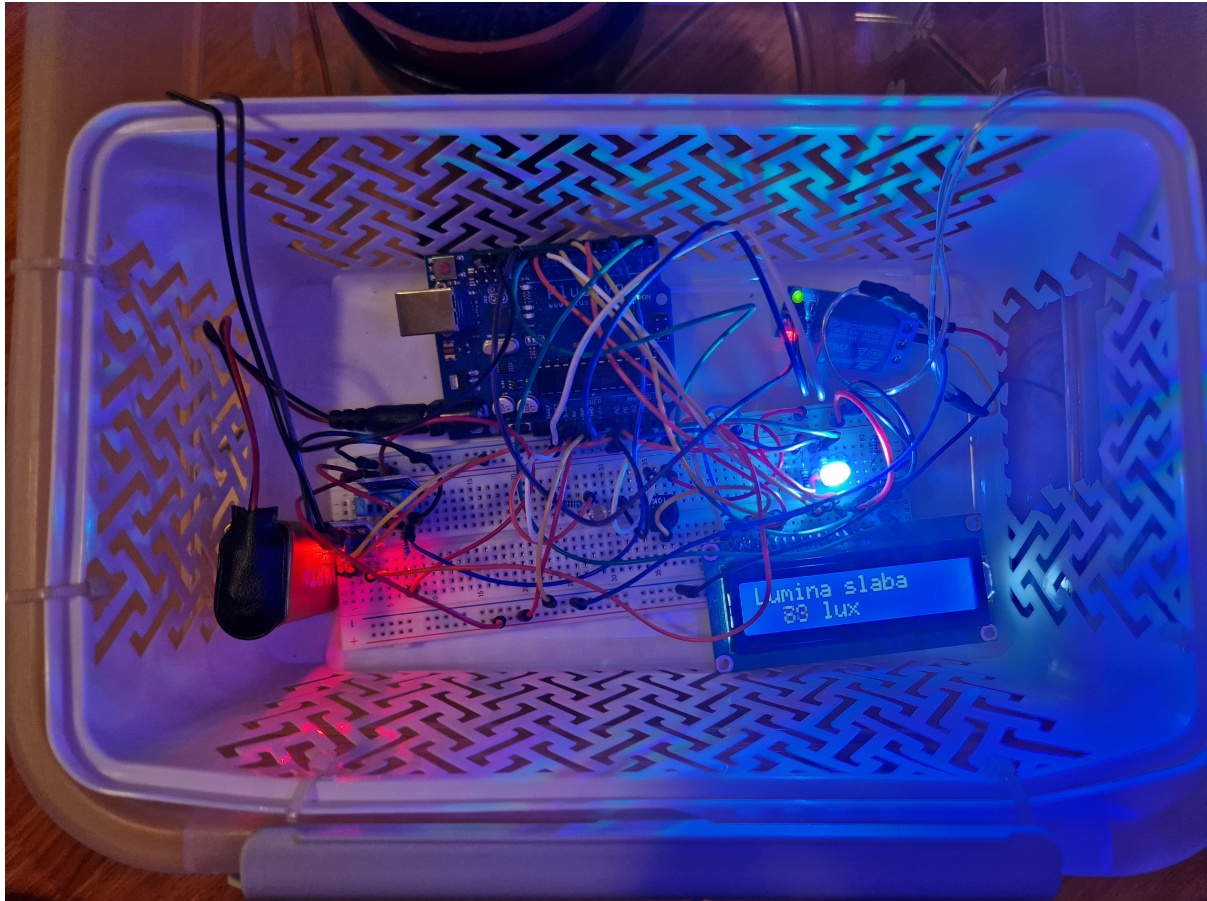
- La nicio apăsare a butonului:



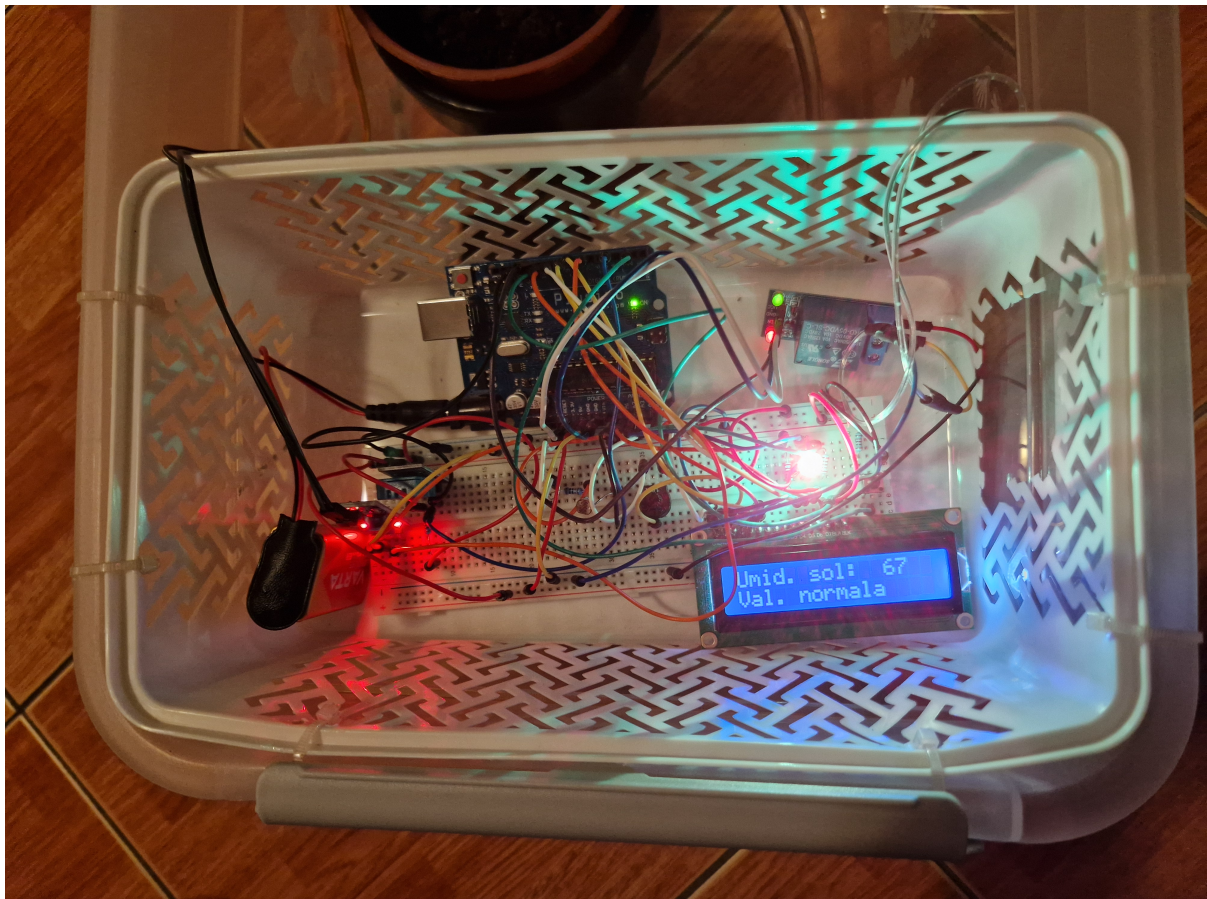
- La prima apăsare a butonului:







- La a doua apăsare a butonului:



La următoarele apăsări se va cicla prin cele 3 tipuri de informație conform exemplurilor de mai sus.

Un demo complet al utilizării proiectului poate fi văzut aici: [Demo dispozitiv de monitorizare al plantelor](#)

## Concluzii

Proiectul a avut dificultate medie, fiind nevoie de atenție la conectarea componentelor, gestionarea senzorilor și aranjarea per ansamblu pentru a avea un rezultat fiabil. Partea importantă a programului a constat în timer și întreruperi, acestea reprezentând elementele de bază funcționării corecte. Chiar dacă ansamblul este voluminos și a durat mai mult timp finalizarea, plănuța de aloe vera din demo este cu siguranță udată corespunzător acum, stând într-un loc prielnic, cu soare și temperatură medie constantă. Ceea ce a ajutat cel mai mult a fost faptul că am lucrat concomitent la partea de hardware și de software, pentru a putea depana mai ușor și a nu acumula prea multe probleme la sfârșit. Scopul a fost în general atins, mai ales că una dintre provocări a fost să mă descurc cu piesele pe care le aveam deja și cu ce mai aveam prin casă pentru a lucra cât mai curat și econom.

## Download

Întreg programul descris mai sus se află aici: [program.zip](#)

## Jurnal

- 1.05.2024 - 5.05.2024 - construire circuit + definitivarea primei părți din documentație (până la descrierea hardware);
- 11.05.2024 - definitivare program sursă, corectat schemă electrică + redactarea descrierii software.
- 12.05.2024 - definitivarea descrierii hardware (descriere pini, prezentare aspect proiect per ansamblu) și adăugarea de exemple de afișări din timpul funcționării circuitului.
- 23.05.2024 - adăugare arhivă cu implementarea completă a programului + link demo
- 26.05.2024 - adăugare concluzie

## Bibliografie/Resurse

1. <https://kits.plusivo.com/microcontroller-super-starter-kit/claim.html> - ghid introductiv Plusivo (pentru funcționare + legare LCD, fotorezistență)
2. [https://arduinogetstarted.com/tutorials/arduino-dht11?utm\\_content=cmp-true](https://arduinogetstarted.com/tutorials/arduino-dht11?utm_content=cmp-true) - legare + funcționare senzor DHT11
3. <https://learn.littlebirdelectronics.com.au/arduino/automatic-plant-watering-with-arduino> - legare + funcționare pompă, releu, senzor de umiditate sol
4. <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024> - configurare timer
5. <https://circuitdigest.com/microcontroller-projects/arduino-interrupt-tutorial-with-examples> - întreruperi cu metode specifice Arduino

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/rrusu/emilia.arpasanu>



Last update: **2024/05/26 18:24**