



UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Facultatea: Automatica si Calculatoare

Specializarea: Calculatoare si Tehnologia Informatiei

# Proiectare cu microProcesoare

~Sistem de irigatie~

Student: Ioneasa Cristina Ioana

Profesor: Rednic Ana

## CUPRINS:

1. Introducere
  - a) Motivul pentru care am ales acest proiect
  - b) Scopul proiectului
2. Solutii anterioare
3. Implementare
  - a) Descriere
  - b) Implementare
    - i. Hardware
    - ii. Software
4. Testing and validation
5. Conclusion

## Introducere

Cum am ales acest proiect?

De fiecare data cand am ocazia de a alege singura tema unui proiect incerc sa aleg unul cu o practicabilitate cat mai ridicata. Asa am facut si acum, dupa niste cautari si cateva zile in care am fost indecisa cu privire la tema proiectul, am hotarat sa fac un mini sistem de irigatie deoarece chiar am nevoie de ceva care sa aiba grija de plantele mele de la camin atunci cand eu sunt plecata acasa. Mi s-a intamplat de multe ori ca atunci cand reveneam dintr-o vacanta sa imi gasesc florile preferate ofilite, ba chiar uscate de tot, asa ca acest proiect ar fi un real ajutor pentru mine.

Scopului proiectului

Asa cum am mentionat si mai sus, scopului proiectului este de a mentine plantele vii, udandu-le atunci cand pamantul este prea uscat. Cu ajutorul acestui sistem, umiditatea pamantului din ghiveci va fi intotdeauna una potrivita pentru plantele respective, avand optiunea de a seta un interval de umiditate adaptat nevoilor fiecarei flori. Se masoara umiditatea din sol, apoi se afiseaza pe LCD un mesaj pentru utilizator (daca solul este pre uscat, prea umed sau are umiditatea perfecta). Un plus al acestui proiect este ca poate fi adaptat pentru orice tip de planta. De exemplu, pentru un cactus, pamantul ar trebui sa fie mai uscat decat pentru Medinilla Magnifica.

## Solutii anterioare

Bineinteles, ideea nu este una proprie, multi au mai facut astfel de proiecte si inaintea mea. Spre exemplu, una din implementarile anterioare o puteti vedea accesand acest link:

<https://learnbywatch.com/automatic-watering-system-for-plants-using-arduino/>

+videoul de prezentare: <https://www.youtube.com/watch?v=nUHizmtyt74>

Acesta masoara nivelul de umiditate din sol, insa senzorul este conectat la un pin digital, astfel ca nu poate masura valori intermediare (doar HIGH si LOW). De aceea acest sistem se va comporta la fel cu orice tip de planta. Un plus pe care il are proiectul meu fata de acesta este ca poate citi valori intermediare, fiind conectat la un pin analogic (A0). In acest mod, utilizatorul poate adapta sistemul in functie de necesarul de apa fiecarei plante. Daca se doreste ca sistemul sa ude o floare care are nevoie de foarte putina apa, se poate seta intervalul de valori aferente umiditatii solului astfel incat pompa sa porneasca numai atunci cand pamantul este foarte uscat. Mai mult decat atat, putem stii valoarea umiditatii daca am conecta sistemul la un laptop si am dechide Serial Monitor.

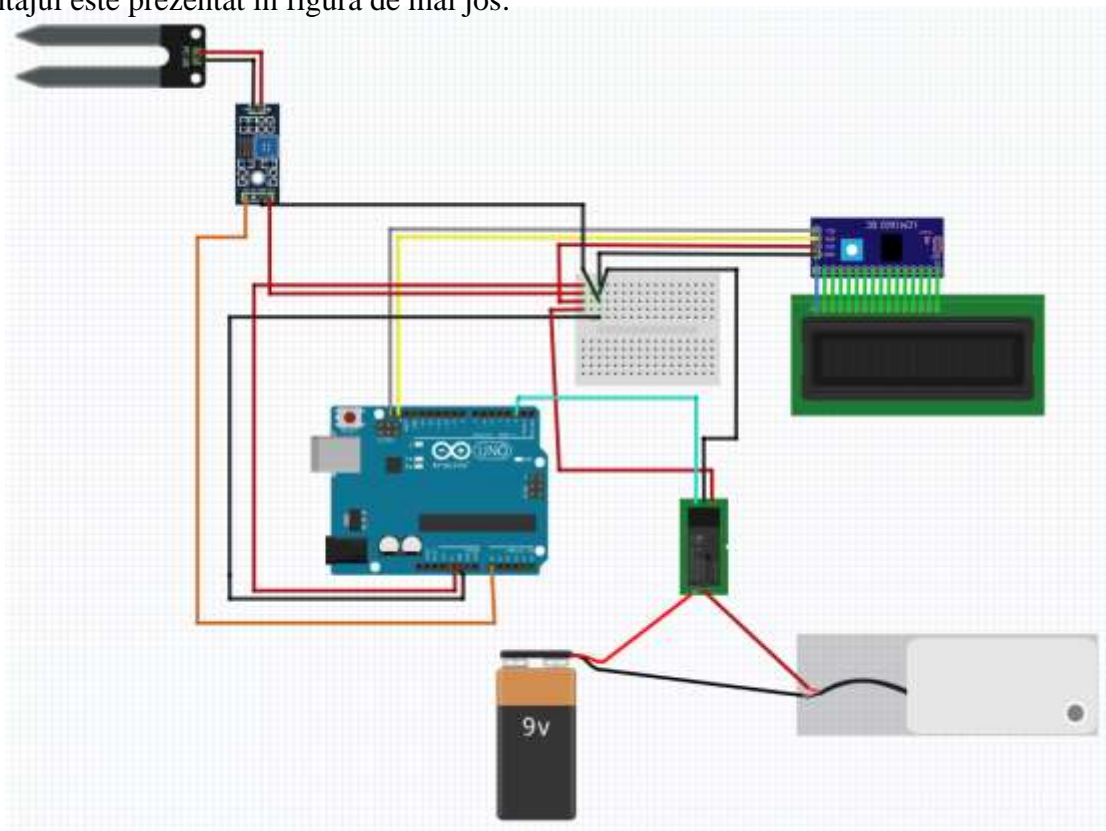
## Implementare

### Hardware

Pentru realizarea acestui proiect am folosit placa Arduino Uno si urmatoarele componente:

1. Senzor de umiditate + comparator LM393 – pentru a masura umiditatea din sol
2. Pompa de apa
3. LCD - pentru a afisa mesajul catre utilizator
4. Modul I2C - pentru reducerea pinilor folositi la legarea LCD-ului la controller
5. Releu – pentru a controla pompa
6. Baterie

Montajul este prezentat in figura de mai jos:



### Software

Pentru a actiona pompa doar la anumite valori de umiditate din sol, am folosit 2 constante soilWet si soilDry. Cu cat pamantul este mai ud, cu atat valoarea citita de senzor este mai mica. Atunci cand valoarea citita de senzor este mai mica decat soilWet, inseamna ca pamantul este prea ud si pompa trebuia sa ramana inchisa. Pentru acesta, pinul la care este conectata va fi HIGH, pentru ca releul folosit este low level trigger si va fi activat atunci cand trece de la HIGH la LOW.

In functia de setup setez pinii ca output si tinem pentru inceput pompa OFF. Apoi in functia loop, citesc valoarea de pe senzor, si in functie de aceasta decid daca trebuie sau nu pornita pompa

si trimite utilizatorului un mesaj, atat prin intermediul LCD-ului, cat si in Serial Monitor. Aici avem 3 cazuri:

- valoarea citita de senzor este mai mica decat soilWet, inseamna ca pamantul este prea ud si pompa trebuia sa ramana inchisa. Pentru acesta, pinul la care este conectata va fi HIGH, pentru ca releul folosit este low level trigger si va fi activat atunci cand trece de la HIGH la LOW.
- valoarea este intre soilWet si soilDry => pompa ramane inactiva, iar utilizatorul este anuntat ca umiditatea este una potrivita pentru planta respective.
- valoarea este mai mare decat soilDry(pamantul prea uscat) => se activeaza pompa punand pinul pe LOW

Pe langa cele doua functii de baza din orice proiect Arduino (setup si loop), mai am o functie numita readSensor care returneaza valoarea citita de senzorul de umiditate.

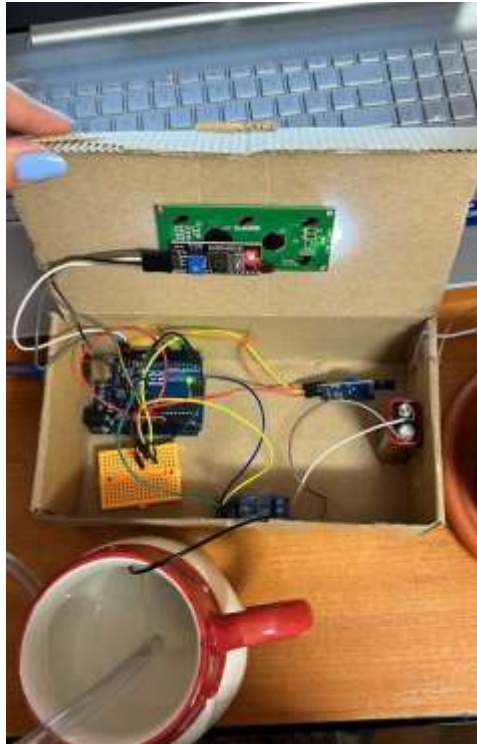
Codul il voi atasa la final, in anexa 1.

## Testarea

Fiind la inceput in lucrul cu Arduino, proiectul nu a fost atat de complex incat sa intampin probleme prea mari. Singurele probleme pe care le-am intampinat au fost din cauza ca nu am fost foarte atenta la functionarea componentelor alese. Pentru inceput, am avut o problema cu LCD-ul pentru ca nu afisa nimic, cel putin asa credeam eu. De fapt contrastul era prea slab si a trebuit sa il ajustez cu potentiometrul pentru ca scrisul sa fie vizibil. O alta problema a fost ca pompa avea un comportament neasteptat. Dupa mai multe testari mi-am dat seama ca functioneaza invers de cum ar fi trebuit. Asa ca doar am inversat LOW cu HIGH si HIGH cu LOW, dar fara sa am vreo explicatie. Ulterior, am aflat ca releul a fost “de vina” pentru acest comportament, fiind un releu low trigger, care actiona pompa doar de la trecerea de la HIGH la LOW.



Dupa ce am vazut ca proiectul este functional si nu mai necesita schimbari pe partea hardware, am introdus majoritatea elementelor intr-o cutie pentru a fi mai usor de transportat si folosit.



## Concluzii

Mi-a placut sa lucrez la acest proiect, mai ales ca il voi utiliza cand voi pleca mai multe zile de acasa. Desi am intampinat mici probleme, a mers totul destul de repede si sunt multumita de final. Bineinteles ca ceea ce am facut eu nu este foarte complex, asa ca mai poate fi imbunatatit. Una dintre imbunatatiri ar putea fi sa adaug acestui proiect mai multi senzori pe care sa ii pun in ghivece diferite pentru a uda mai multe flori. In viitor voi face asta daca floarea mea nu va mai fi singura si va trebui sa am grija de mai multe.

## Anexa 1

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define soilWet 500
#define soilDry 750
// Sensor pins
#define sensorPower 7
#define sensorPin A0
//Pump pin
#define pumpPin 2

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  pinMode(sensorPower, OUTPUT);
  pinMode(pumpPin, OUTPUT);

  digitalWrite(sensorPower, LOW);
  digitalWrite(pumpPin, HIGH);

  Serial.begin(9600);
  lcd.begin ();
}

void loop() {

  int moisture = readSensor();

  Serial.println(moisture);

  lcd.backlight();
  lcd.setCursor(0, 0);

  if (moisture < soilWet) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Soil is too wet");
    digitalWrite(pumpPin, HIGH);
    Serial.println("Status: Soil is too wet");
```

```

}
else if (moisture >= soilWet && moisture < soilDry) {
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Soil moisture is");
  lcd.setCursor(4, 1);
  lcd.print("perfect");
  digitalWrite(pumpPin, HIGH);
  Serial.println("Soil moisture is perfect");
}
else if(moisture > soilDry){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Soil is too dry ");
  lcd.setCursor(0, 1);
  lcd.print("Time to water!");
  Serial.println("Soil is too dry - time to water!");
  digitalWrite(pumpPin, LOW);

}

delay(1000);
Serial.println();
}

```

```

int readSensor() {
  digitalWrite(sensorPower, HIGH);
  delay(10);
  int val = analogRead(sensorPin);
  digitalWrite(sensorPower, LOW);
  return val;
}

```