

PlantCare

Introducere

PlantCare este un sistem ce își propune să *eficientizeze* și să *ușureze* îngrijirea plantelor de apartament. Proiectul se bazează pe un sistem *inteligent* și *automat* care decide când și cât să ude plantele pe care acesta le supraveghează. PlantCare este ideal pentru *persoanele începătoare* în domeniul îngrijirii plantelor, dar și pentru situațiile în care plantele sunt lăsate *nesupravegheate* mai mult timp. Sistemul este echipat cu ecran pentru o interacțiune mai prietenoasă cu utilizatorul.

Descriere generală

Mai jos sunt descrise **principalele module** ale proiectului și modul lor de funcționare:

- Placuta Arduino UNO ce folosește microcontroller-ul ATmega 328p, necesară în controlul sistemului
- Ecran LCD 16x2 ce afișează informații colectate de la senzori
- Adaptorul MicroSD este utilizat pentru a stoca informații referitoare la timpurile de udare
- Senzorul de temperatură colectează date referitoare la temperatura din încăpere pentru a fi afișate pe display
- Pompa de apă ajută la transportul apei prin furtun în ghiveci



Mod de funcționare

Sistemul este compact, plasat într-o cutie de dimensiuni 10x15 cm și supraveghează nivelul de umiditate din sol și activează pompa atunci când nivelul de apă din sol este insuficient. Pe display sunt afișate la intervale regulate mesaje referitoare la temperatura și umiditate. Dacă se dorește înregistrarea timpului trecut de la ultima udare, utilizatorul poate opta să introducă un card microSD în compartimentul din interior. Senzorul de umiditate se plasează în pământ, mai aproape de furtun dacă ghiveciul este mic, sau mai departe dacă ghiveciul este mai mare.

Hardware Design

Componentele folosite

Piesa	Cantitate
Arduino UNO(ATmega328P)	1
Modul cu senzor de umiditate sol	1
Ecran LCD 1602 IIC/I2C	1
Senzor de temperatura	1
Pompa de apa submersibila 3-6V	1
Adaptor microSD	1
Tranzistor NPN 2N2222	1
Dioda 1N4007	1
Condensator polarizat 10uF	2
Rezistenta 100ohm	2
Breadboard 400 de puncte	1

Descriere conexiuni

Senzor umiditate

Deoarece nu este necesara colectarea continua de date privind umiditatea din sol, senzorul de umiditate este conectat la un pin digital de la placuta pentru a putea fi controlat. Astfel, se evita corozionul senzorului prin folosire continua, dar si consumul in plus de curent. Acest lucru reprezinta o **optimizare** adusa proiectului.

Pompa submersibila

Pompa este conectata la sistem printr-un circuit format din:

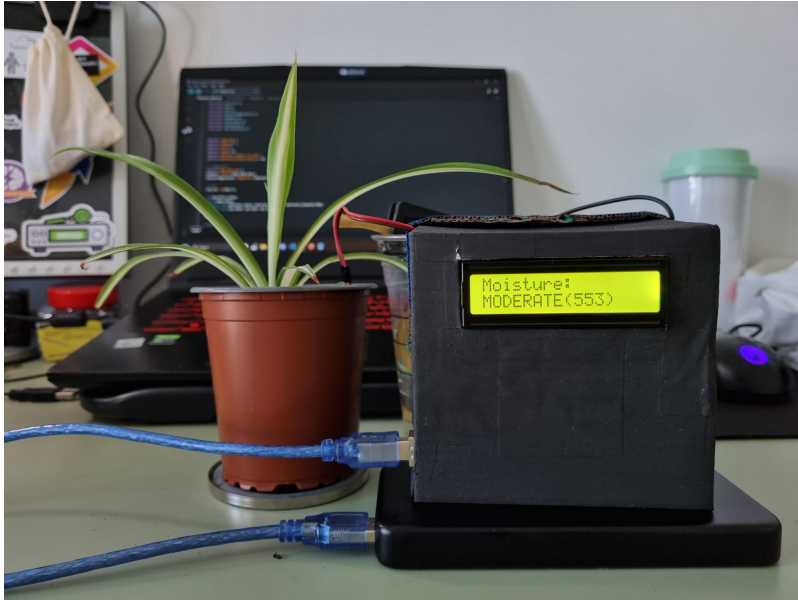
- 1 tranzistor
- 2 condensatoare pentru reducerea zgomotului produs de pornirea pompei(acest lucru a fost necesar deoarece pornirea pompei inducea zgomot in circuit, iar LCD-ul nu mai functiona cum trebuie)
- 1 dioda

Schema electrica a proiectului:



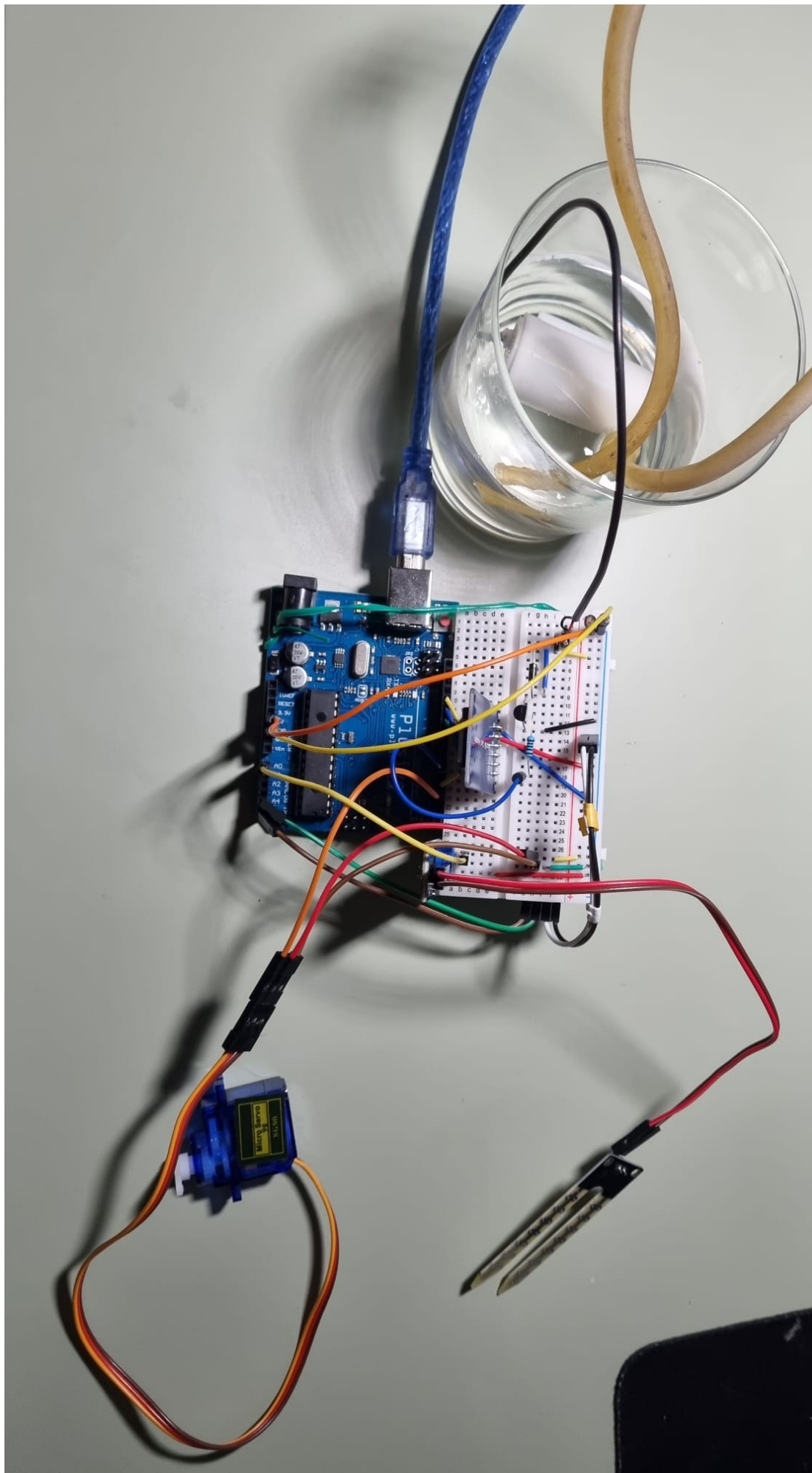
Stadiul actual al proiectului:

26.05.2024



18.05.2024





Software Design

Mediul de dezvoltare folosit: **Arduino IDE**

Biblioteci utilizate:

- LCD_I2C
- OneWire
- DallasTemperature
- SD
- SPI

Elementul de noutate al proiectului consta in faptul ca acesta dispune de un slot pentru cardul micro sd menit pentru a retine timpul trecut de la ultima udare, util pentru a realiza o analiza cu privire la frecventa udarilor.

```
unsigned long SD_START, SD_STOP, SD_TIME;
File file;

// functie pentru scriere in SDcard
void write_time(unsigned long start, unsigned long finished, unsigned long
elapsed, File file) {

    float h, m, s, ms;
    unsigned long over;

    elapsed = finished - start;
    h = int(elapsed / 3600000);
    over = elapsed % 3600000;
    m = int(over / 60000);
    over = over % 60000;
    s = int(over / 1000);
    ms = over % 1000;

    if (file) {
        file.print("Elapsed time: ");
        file.print(h,0);
        file.print("h ");
        file.print(m,0);
        file.print("m ");
        file.print(s,0);
        file.print("s ");
        file.print(ms,0);
        file.println("ms");
        file.println();
        Serial.println("Written");
    }
}
```

```
    file.close();
  }
}

// utilizare functie folosind millis()
SD_START = SD_STOP;
SD_STOP = millis();
SD_TIME = SD_STOP - SD_START;
write_time(SD_START, SD_STOP, SD_TIME, file);
```

Am utilizat **intreruperi** pentru a controla pompa submersibila atunci cand umiditatea depaseste valoarea inregistrata de 900. Dupa ce se realizeaza un ciclu de udat si umiditatea din sol creste, pompa fie se opreste fie continua udarea pana cand solul este suficient de umed. Apoi sistemul continua sa afiseze pe display datele colectate de la senzori.

Logica pentru rutina de tratare a intreruperii este urmatoarea:

```
#define INT0_PIN 2

// in interiorul setup()
pinMode(INT0_PIN, OUTPUT);
attachInterrupt(digitalPinToInterrupt(INT0_PIN), INT0_ISR, RISING);

// in interiorul loop()
if(sensorValue > 900) {
  digitalWrite(INT0_PIN, LOW);
  delay(1);
  digitalWrite(INT0_PIN, HIGH);
  // water for 5 seconds
  delay(5000);
  analogWrite(pump_pin, 0);
  delay(3000);
}

// ISR
void INT0_ISR() {
  analogWrite(pump_pin, speed);
}
```

Senzorul de umiditate porneste o data la o ora, dar acest interval poate fi marit(din cod).

```
// after an hour start the humidity sensor
if(moisture_elapsed_time >= 3600000) {
  digitalWrite(moisture_sensor_vcc_pin, HIGH);
  moisture_value = analogRead(moisture_sensor_data_pin);
  // delay for stability
  delay(2000);
  moisture_value = analogRead(moisture_sensor_data_pin);
}
```

```
}  
}
```

Rezultate Obținute

Un sistem automat de udare a plantelor care poate fi alimentat la o baterie externa.

Concluzii

Concluzia la care am ajuns dupa ce am lucrat la acest proiect timp de mai multe zile este ca "socoteala de acasa nu este la fel cu socoteala din targ". Ce vreau sa zic este ca initial am plecat cu o idee a proiectului ce utiliza un servomotor pentru a dirija apa in ghiveci, insa cand am realizat hardware-ul am observat un comportament neprevazut: servomotorul pornit in acelasi timp cu pompa incepea sa se invarta rapid la 360 de grade. Acest lucru era cel mai probabil cauzat de zgomotele produse in circuit de pompa de apa, facand servomotorul sa se comporte imprevizibil. Ignorand acest mic incident, lucrul la proiect a fost o activitate placuta din care am avut mai multe de invatat, atat din punct de vedere software, dar mai ales hardware.

Download

Arhiva cu proiectul si fisierele corelate se gaseste aici: [plantcare_project.zip](#)

Jurnal

09.05.2024: Comandare componente

18.05.2024: Finalizare Hardware Design

19.05.2024: Eliminare servomotor si inlocuirea acestuia cu senzorul de temperatura

23.05.2024: Finalizare Software Design

25.05.2024: Finalizare carcasa

26.05.2024: Finalizare pagina WIKI

Bibliografie/Resurse

Resurse Software:

1. <https://www.instructables.com/Arduino-Timing-Methods-With-Millis/> → temporizare folosind millis()

2. <https://docs.arduino.cc/learn/programming/sd-guide/> → scriere/ citire in sd card

Resurse Hardware:

1. <https://forum.arduino.cc/t/how-to-reduce-noise-from-motor/888641/4> → reducerea zgomotului de la pompa
2. <https://www.instructables.com/> → util in realizarea cablajului
3. <https://www.tinkercad.com/> → realizare schema electrica

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/alucaci/andreea.budulan>



Last update: **2024/05/26 22:26**