

Smart Seed Planter

Nume: Floarea Gabriela-Estera

Grupă: 332CB

Introducere

Smart Seed Planter este un sistem embedded automatizat care simulează procesul de plantare și îngrijire inițială a unei plante, de la pregătirea solului până la udarea automată, fără intervenție umană după pornire.

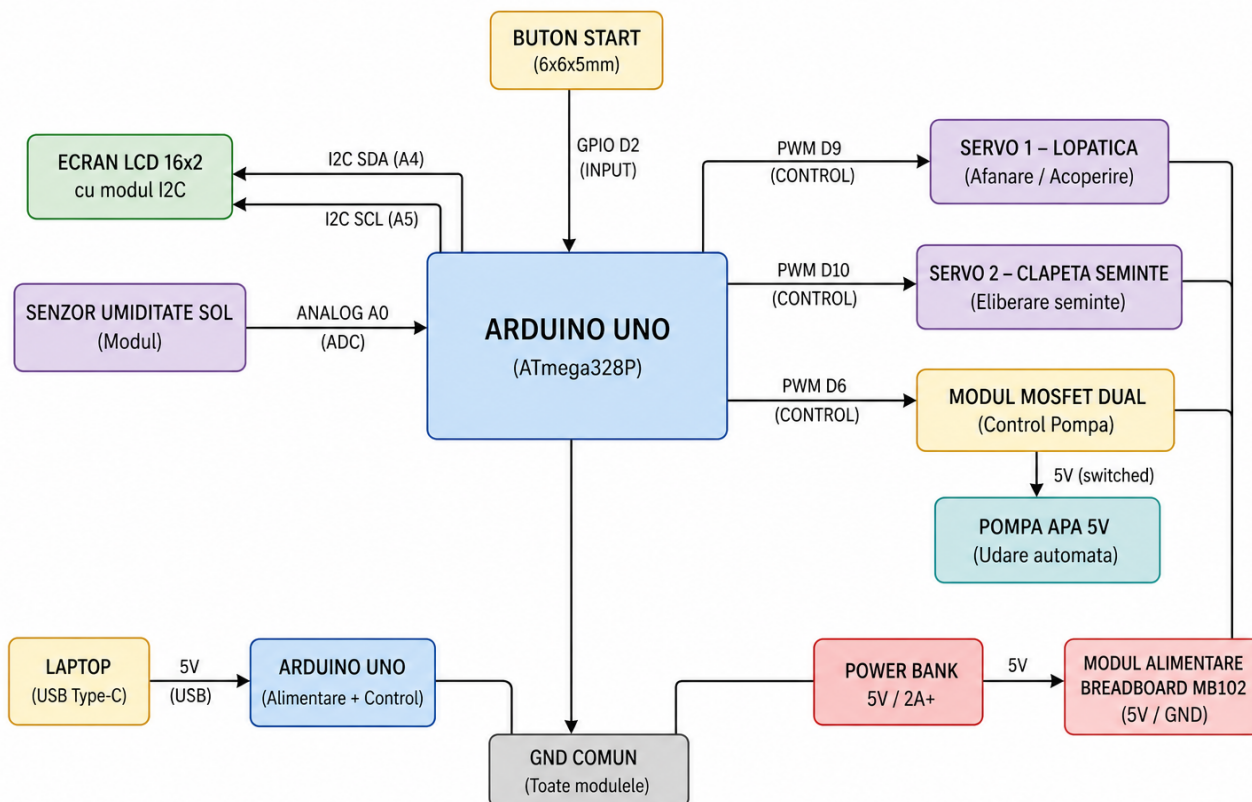
Ideea proiectului a pornit de la dorința de a automatiza un proces agricol de bază folosind componente electronice accesibile și un microcontroller Arduino UNO. Sistemul execută o secvență completă de operații: afânarea solului cu o lopățică acționată de servomotor, citirea umidității printr-un senzor analogic, luarea unei decizii automate privind plantarea, eliberarea unei semințe printr-o clapetă mecanică, acoperirea seminței cu pământ și udarea zonei plantate prin intermediul unei pompe controlate prin MOSFET.

Proiectul este util din mai multe perspective:

- poate reprezenta baza unui sistem simplificat de agricultură inteligentă, capabil să automatizeze procese precum plantarea, monitorizarea umidității și udarea automată;
- reduce intervenția umană necesară pentru îngrijirea inițială a plantelor și poate fi adaptat pentru sere automatizate, grădini inteligente sau sisteme educaționale interactive;
- arhitectura modulară permite extinderea sistemului cu funcții suplimentare, precum conectivitate wireless, monitorizare prin telefon sau integrarea unor senzori de temperatură și lumină.

Descriere generală

Sistemul este structurat ca o **mașină de stări finite (FSM)**, coordonată de Arduino UNO. Utilizatorul apasă butonul de start, după care sistemul execută automat întreaga secvență: verificare umiditate, afânare sol, eliberare sămânță, acoperire, udare și finalizare. Dacă umiditatea nu este în intervalul optim (400-700), procesul este întrerupt, iar utilizatorul este notificat prin LCD.



Arduino UNO coordonează patru module principale. **Modulul de detecție** folosește un senzor rezistiv de umiditate conectat pe A0, cu media a 5 citiri consecutive pentru reducerea zgomotului. **Modulul mecanic** include două servomotoare SG90 alimentate din MB102: Servo 1 acționează lopățița pentru afânare și acoperire (D9), iar Servo 2 controlează clapeta de eliberare a semințelor (D11). **Modulul de irigație** controlează pompa de apă 5V printr-un modul dual MOSFET (D6), necesar deoarece pompa depășește curentul maxim al unui pin Arduino. **Modulul de interfață** afișează starea curentă și valoarea umidității pe LCD-ul 16x2 cu modul I2C (A4/A5).

Alimentarea este duală: laptopul alimentează Arduino prin USB Type-C, iar un adaptor extern alimentează servomotoarele și pompa prin modulul MB102. Toate modulele au masă comună.

Hardware Design

Listă de piese

Componentă	Cantitate	Rol în sistem
Arduino UNO R3, ATmega328P, CH340G, Type-C	1x	Microcontroller principal, FSM
Servomotor SG90 (0°-180°)	2x	Lopățiță (D9) + clapetă semințe (D11)
Condensator electrolitic 470μF 16V	2x	Stabilizare alimentare servomotoare
Condensator electrolitic 1000μF 16V	1x	Stabilizare alimentare principală
Senzor umiditate sol cu modul	1x	Citire umiditate analogică (A0)

Pompă apă 5V	1x	Udare automată
Furtun silicon ~40cm	1x	Transport apă către sol
Modul dual MOSFET 15A 400W	1x	Control pompă din Arduino
Diodă protecție 1N4007	1x	Protecție la curent invers
Buton tactil 6x6x5mm	1x	Pornire secvență
Ecran LCD 16x2 + modul I2C	1x	Afișare stare și umiditate
Modul alimentare breadboard MB102	1x	Distribuire alimentare externă
Adaptor 9V	1x	Alimentare externă
Breadboard 830 puncte	1x	Breadboard principal
Breadboard 400 puncte	1x	Zonă pompă + MOSFET
Fire jumper M-M / M-F	~70	Conexiuni între module

Interconectarea hardware

Sistemul este construit în jurul plăcii **Arduino UNO**, care coordonează toate componentele hardware. Componentele de intrare sunt senzorul de umiditate și butonul de start, iar componentele de ieșire sunt cele două servomotoare, LCD-ul și pompa de apă controlată prin MOSFET.

LCD-ul 16x2 cu modul I2C este conectat prin interfața I2C folosind pinii A4 și A5 ai plăcii Arduino UNO. Utilizarea I2C reduce numărul de pini necesari și simplifică schema electrică.

Senzorul de umiditate este conectat pe pinul analogic A0, deoarece furnizează o valoare analogică variabilă în funcție de nivelul de apă din sol.

Butonul de start este conectat pe pinul D2 și utilizează rezistența internă de tip INPUT_PULLUP a microcontrollerului. Astfel, nu mai este necesară utilizarea unei rezistențe externe.

Cele două servomotoare SG90 sunt conectate pe pinii D9 și D11. Servo 1 controlează lopățile pentru afânarea și acoperirea solului, iar Servo 2 controlează clapeta pentru eliberarea seminței.

Pompa de apă este controlată prin intermediul modului dual MOSFET conectat pe pinul D6. Pompa nu este conectată direct la Arduino, deoarece consumă mai mult curent decât poate furniza un pin digital al microcontrollerului.

Condensatoarele electrolitice sunt conectate pe linia de alimentare externă pentru reducerea fluctuațiilor de tensiune produse de servomotoare și pompă.

Dioda 1N4007 este montată paralel pe pompă pentru protecție împotriva tensiunilor inverse generate la oprirea motorului.

Alimentare

Sistemul utilizează alimentare separată pentru partea logică și partea de putere. Arduino UNO este alimentat prin USB Type-C, iar servomotoarele și pompa sunt alimentate separat prin modulul MB102. În etapa actuală, MB102 este alimentat cu un adaptor de 9V conectat la priză.

Toate modulele folosesc **GND comun**, necesar pentru interpretarea corectă a semnalelor de control.

Justificarea pinilor folosiți

Pinii utilizați au fost aleși în funcție de rolul fiecărei componente și de funcționalitățile oferite de placa Arduino UNO. LCD-ul 16x2 utilizează comunicația I2C, motiv pentru care este conectat la pinii A4 și A5, dedicați liniilor SDA și SCL. Senzorul de umiditate este conectat la pinul analogic A0, deoarece furnizează o valoare analogică proporțională cu nivelul de apă din sol. Butonul de start este conectat pe pinul digital D2 și utilizează rezistența internă INPUT_PULLUP pentru simplificarea conexiunilor hardware. Cele două servomotoare sunt conectate pe pinii D9 și D11, iar modulul MOSFET este conectat pe pinul D6 pentru controlul pompei de apă.

Tabel de conexiuni

Componentă	Pin componentă	Pin Arduino / Alimentare	Rol
LCD I2C	SDA	A4	Linie date I2C
LCD I2C	SCL	A5	Linie ceas I2C
LCD I2C	VCC	5V Arduino	Alimentare LCD
LCD I2C	GND	GND	Masă comună
Senzor umiditate	AOUT	A0	Citire analogică
Senzor umiditate	VCC	5V Arduino	Alimentare senzor
Senzor umiditate	GND	GND	Masă comună
Buton START	pin 1	D2	Pornire secvență
Buton START	pin 2	GND	INPUT_PULLUP
Servo lopățică	semnal	D9	Control servo
Servo lopățică	VCC	5V_MB102	Alimentare externă
Servo lopățică	GND	GND	Masă comună
Servo clapetă	semnal	D11	Control clapetă
Servo clapetă	VCC	5V_MB102	Alimentare externă
Servo clapetă	GND	GND	Masă comună
MOSFET	TRIG/PWM	D6	Control pompă
MOSFET	VIN+	5V_MB102	Alimentare pompă
MOSFET	VIN-	GND_MB102	Masă alimentare
Pompă	+	VOUT+ MOSFET	Alimentare pompă
Pompă	-	VOUT- MOSFET	Alimentare pompă
Diodă 1N4007	catod	VOUT+	Protecție pompă
Diodă 1N4007	anod	VOUT-	Protecție pompă
Condensatoare	+	5V_MB102	Stabilizare tensiune
Condensatoare	-	GND_MB102	Stabilizare tensiune

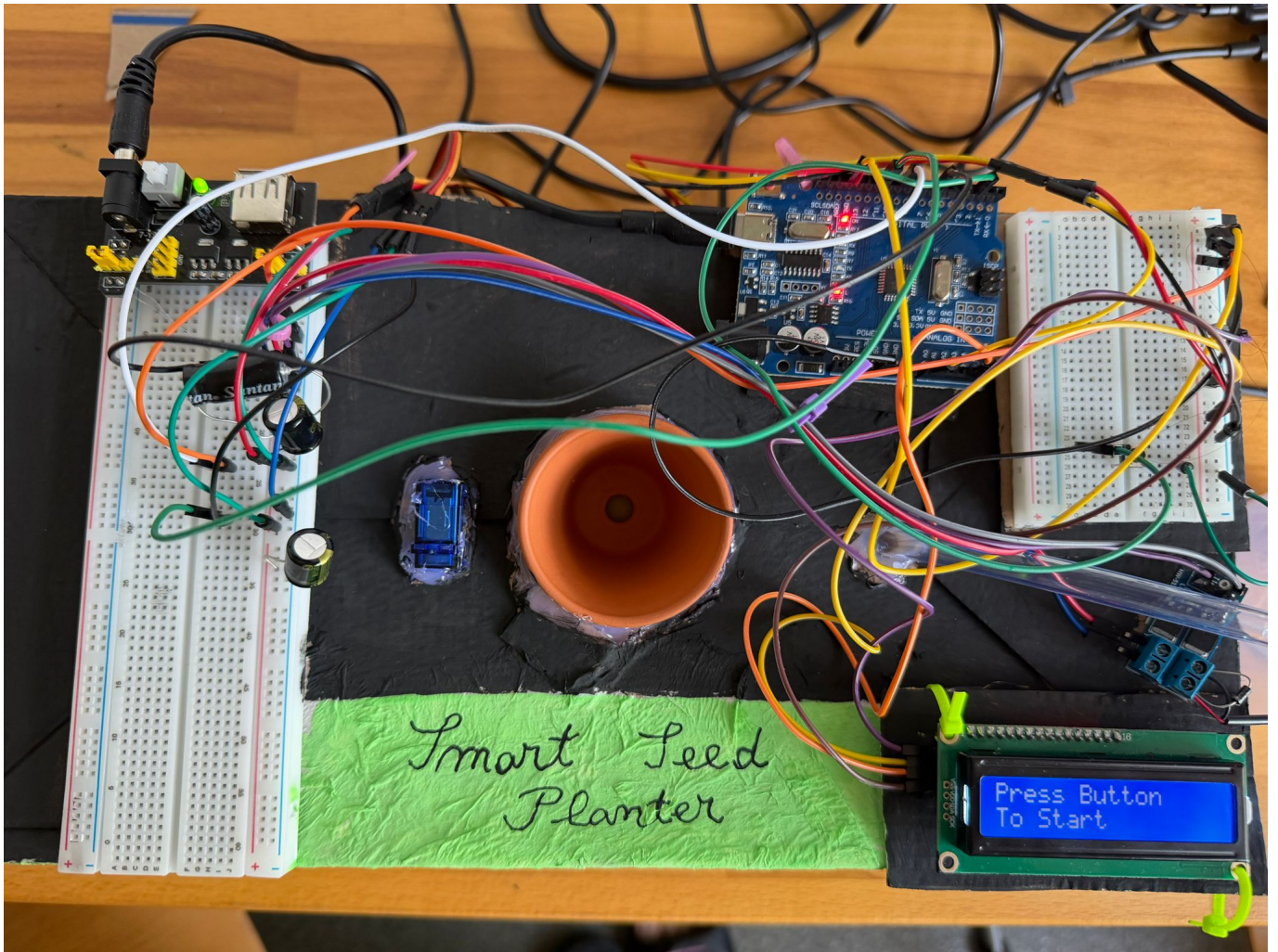
Schema electrică



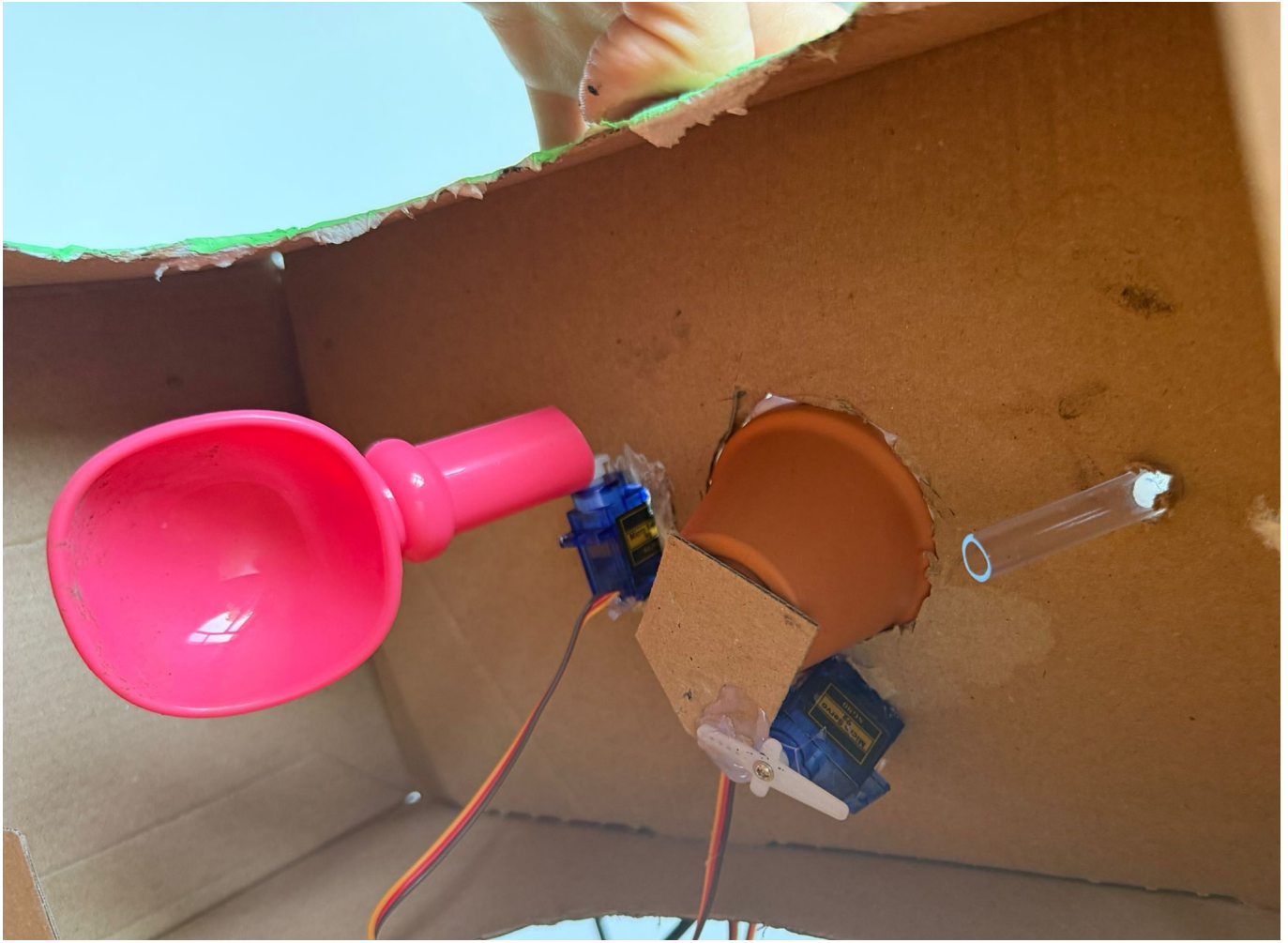
Explicarea schemei electrice

- **Arduino UNO** reprezintă unitatea centrală a sistemului și coordonează toate componentele conectate.
- **LCD-ul** utilizează protocolul I2C pentru reducerea numărului de conexiuni necesare. Comunicarea se realizează prin liniile SDA și SCL, conectate la pinii A4 și A5.
- **Senzorul de umiditate** transmite către Arduino o valoare analogică proporțională cu nivelul de apă din sol. Această valoare este utilizată pentru luarea deciziei privind continuarea sau oprirea procesului de plantare.
- **Butonul de start** este conectat între D2 și GND. În configurația INPUT_PULLUP, pinul are în mod normal valoarea HIGH și devine LOW atunci când butonul este apăsat.
- **Servomotoarele** sunt controlate prin impulsuri generate de Arduino, însă sunt alimentate separat prin MB102 pentru a evita instabilitatea alimentării microcontrollerului.
- **Pompa de apă** este controlată prin intermediul modului MOSFET. Arduino trimite doar semnalul de comandă, iar alimentarea efectivă a pompei se face din sursa externă.
- **Dioda 1N4007** este montată paralel pe pompă și protejează circuitul împotriva tensiunilor inverse produse la oprirea motorului pompei.
- **Condensatoarele electrolitice** stabilizează alimentarea și reduc fluctuațiile de tensiune produse de actuator.
- Toate modulele folosesc aceeași masă comună pentru funcționarea corectă a întregului sistem.

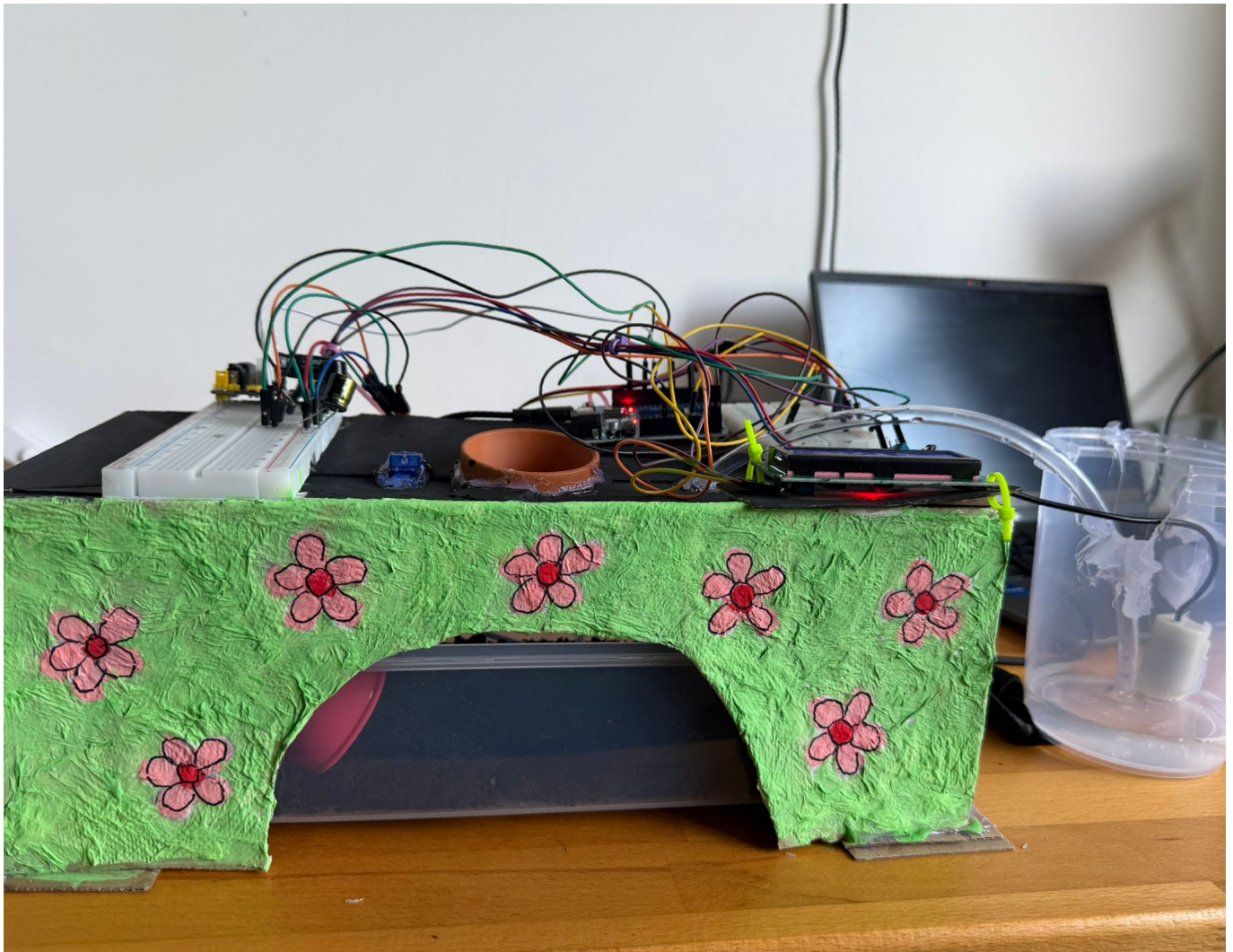
Componentele conectate - LCD Test



Componentele conectate - Servomotoare si pompa



Componentele conectate - Design final



Software Design

Mediu de dezvoltare

Proiectul a fost dezvoltat în **Visual Studio Code**, folosind **PlatformIO** pentru placa **Arduino UNO / ATmega328P**.

Codul este scris în C/C++ și este împărțit în patru fișiere principale:

- `main.cpp` - conține logica principală și mașina de stări finite;
- `drivers.cpp` - conține implementarea funcțiilor pentru periferice;
- `drivers.h` - conține prototipurile funcțiilor;
- `config.h` - conține pragurile și constantele configurabile.

Deși proiectul folosește structura standard Arduino, cu `setup()` și `loop()`, controlul perifericelor nu este realizat prin funcții Arduino high-level precum `digitalWrite()`, `analogRead()`, `Serial`, `Wire` sau `Servo`. Componentele sunt controlate direct prin registrii microcontrollerului ATmega328P.

Arhitectura software

Aplicația este implementată ca o **mașină de stări finite (FSM)**. Stările principale sunt:

```
IDLE,  
CHECK_HUMIDITY,  
DIG_SOIL,  
DROP_SEED,  
COVER_SEED,  
WATER_PLANT,  
DONE,  
ERROR_HUMIDITY
```

Fluxul programului este următorul:

1. sistemul așteaptă apăsarea butonului de start;
2. se citește umiditatea solului;
3. dacă umiditatea este în intervalul acceptat, procesul continuă;
4. servomotorul acționează lopățița pentru afânarea solului;
5. al doilea servomotor eliberează sămânța;
6. lopățița revine pentru acoperirea seminței;
7. pompa este pornită pentru udare;
8. sistemul afișează mesajul de finalizare și revine în starea inițială.

Dacă umiditatea nu este între valorile definite în `config.h`, procesul este oprit și se afișează mesajul de eroare.

Inițializarea sistemului

În funcția `setup()` sunt inițializate toate modulele necesare:

```
timerInit();  
uartInit(9600);  
gpioInit();  
adcInit();  
i2cInit();  
lcdInit();  
servoInit();  
pumpOff();
```

Aceste funcții configurează timerul, comunicația serială, pinii GPIO, ADC-ul, I2C-ul, LCD-ul, servomotoarele și pompa.

Funcționalități implementate

Citirea umidității

Senzorul de umiditate este conectat pe A0. Citirea se face prin ADC, folosind registrii ADMUX, ADCSRA și ADC, nu prin `analogRead()`.

Pentru stabilizarea valorii, se face media a 5 citiri consecutive:

```
for (int i = 0; i < 5; i++)
{
    sum += adcRead(0);
    delayMs(20);
}
```

Valoarea este comparată cu pragurile:

```
#define HUMIDITY_MIN 400
#define HUMIDITY_MAX 700
```

Afișarea pe LCD

LCD-ul 16x2 este controlat prin I2C. Nu este folosită biblioteca `LiquidCrystal_I2C`, ci o implementare proprie bazată pe registrii TWI:

- TWSR
- TWBR
- TWCR
- TWDR

LCD-ul afișează starea curentă a sistemului: așteptare, verificare umiditate, plantare, udare, eroare sau finalizare.

Controlul servomotoarelor

Servomotoarele sunt controlate fără biblioteca `Servo.h`. Semnalul este generat manual prin impulsuri pe pinii:

- PB1 - servomotor lopățiță;
- PB3 - servomotor clapetă.

Unghiul este transformat într-un impuls între aproximativ 1000 us și 2000 us, iar temporizarea se face cu `Timer1`.

Controlul pompei

Pompa este controlată prin pinul D6, corespunzător bitului PD6. Aceasta este acționată prin modulul MOSFET și funcționează active HIGH.

```
void pumpOn()  
{  
    PORTD |= (1 << PD6);  
}  
  
void pumpOff()  
{  
    PORTD &= ~(1 << PD6);  
}
```

Debugging prin UART

Pentru debugging este folosită comunicația USART la 9600 baud. Nu este folosită clasa Serial, ci o implementare proprie cu registrii:

- UBRR0H, UBRR0L
- UCSR0B, UCSR0C
- UCSR0A
- UDR0

Exemple de mesaje transmise:

```
uartPrintln("Smart Seed Planter started");  
uartPrint("Humidity: ");  
uartPrintNumber(humidity);
```

Utilizarea conceptelor din laboratoare

Proiectul folosește concepte din mai multe laboratoare PM:

- **Laboratorul 1 - UART:** comunicație serială pentru debugging prin USART;
- **Laboratorul 3 - Timere/PWM:** Timer1 pentru delay-uri precise și impulsuri pentru servomotoare;
- **Laboratorul 4 - ADC:** citirea senzorului de umiditate pe A0;
- **Laboratorul 6 - I2C:** comunicația cu LCD-ul 16x2 prin TWI/I2C.

Astfel, proiectul folosește cel puțin 3 funcționalități din laboratoare și le integrează într-un sistem embedded complet.

Rezultate Obținute

În urma implementării, proiectul a ajuns într-o formă funcțională. Sistemul pornește corect, afișează mesajul inițial pe LCD și așteaptă apăsarea butonului de start.

După apăsarea butonului, Arduino citește valoarea umidității solului prin senzorul analogic conectat la A0. Valoarea este afișată atât pe LCD, cât și în Serial Monitor prin UART. Dacă umiditatea este în intervalul acceptat, sistemul afișează mesajul "Humidity OK" și începe procesul de plantare.

Secvența de plantare este executată automat. Primul servomotor acționează lopățița pentru afânarea solului, al doilea servomotor deschide clapeta pentru eliberarea seminței, iar apoi lopățița revine pentru acoperirea seminței. La final, pompa este pornită pentru udare și oprită după timpul configurat în cod.

Pe parcursul testării, LCD-ul a afișat corect stările sistemului. Serial Monitor a fost folosit pentru verificarea valorilor citite și pentru debugging. Prin acesta s-a confirmat că programul ajunge în etapele principale și că funcția de oprire a pompei este apelată corect.

O problemă observată în timpul testării a fost curgerea apei după oprirea pompei. Aceasta nu a fost cauzată de cod, ci de poziționarea furtunului și de efectul de sifon. Problema poate fi redusă prin poziționarea rezervorului mai jos, folosirea unei supape anti-sifon sau modificarea traseului furtunului.

O altă dificultate a fost poziționarea corectă a servomotoarelor, astfel încât acestea să realizeze mișcările specifice lopățiții și clapetei pentru rezervorul de semințe. Lopățița acționează mai eficient atunci când pământul nu este dens, ci mai afânat și granulat.

Rezultatul final este un prototip funcțional de sistem embedded pentru plantare automată, care integrează citirea unui senzor analogic, controlul a două servomotoare, controlul unei pompei, afișarea pe LCD și debugging prin UART.

Demo: <https://youtube.com/shorts/nhgc7JsO4m4?is=UIIFKVMUwB3DC9uY>

Concluzii

Se plantează!!!

Proiectul a demonstrat că un proces simplificat de plantare poate fi automatizat folosind un microcontroller Arduino UNO și componente electronice accesibile. Sistemul reușește să citească umiditatea solului, să ia o decizie pe baza acesteia, să acționeze mecanic procesul de plantare și să pornească udarea automată.

Implementarea a permis utilizarea practică a mai multor concepte studiate în laboratoare, precum comunicația UART, citirea ADC, controlul prin registri, utilizarea timerelor și comunicația I2C. În plus, proiectul a evidențiat importanța alimentării separate pentru actuatori și a proiectării corecte a părții mecanice.

În forma actuală, prototipul este funcțional și poate fi extins ulterior prin adăugarea unor senzori suplimentari, conectivitate wireless sau o carcasă mecanică mai stabilă.

Download

Jurnal

- Alegerea temei și stabilirea componentelor principale.
- Realizarea schemei hardware și conectarea componentelor.
- Testarea LCD-ului prin I2C.
- Testarea senzorului de umiditate.
- Implementarea controlului pentru servomotoare.
- Implementarea controlului pentru pompă prin MOSFET.
- Integrarea componentelor într-o mașină de stări finite.
- Testarea finală și corectarea problemelor observate.

Bibliografie/Resurse

- <https://docs.arduino.cc/>
- <https://www.microchip.com/en-us/product/ATmega328P>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab1-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab4-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/andrei.batasev/gabriela.floarea>



Last update: **2026/05/25 12:19**