



UNIVERSITATEA DIN CRAIOVA
FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI
ELECTRONICĂ

DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI



PROIECT DE DIPLOMĂ

Georgiana - Lucreția Stoian

COORDONATOR ȘTIINȚIFIC

Ș.L. Dr. Ing. Eugen Dumitrașcu

IULIE 2016

CRAIOVA



UNIVERSITATEA DIN CRAIOVA
FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI
ELECTRONICĂ

DEPARTAMENTUL DE CALCULATOARE ȘI
TEHNOLOGIA INFORMAȚIEI



MONITORIZAREA ȘI CONTROLUL MEDIULUI DINTR-O SERĂ

Georgiana – Lucreția Stoian

COORDONATOR ȘTIINȚIFIC

Ș.L. Dr. Ing. Eugen Dumitrașcu

IULIE 2016

CRAIOVA

„Învățătura este o comoară care își urmează stăpânul pretutindeni.”

Proverb popular

DECLARAȚIE DE ORIGINALITATE

Subsemnatul GEORGIANA- LUCREȚIA STOIAN, student la specializarea CALCULATOARE CU PREDARE ÎN LIMBA ROMÂNĂ din cadrul Facultății de Automatică, Calculatoare și Electronică a Universității din Craiova, certific prin prezenta că am luat la cunoștință de cele prezentate mai jos și că îmi asum, în acest context, originalitatea proiectului meu de licență:

- cu titlul MONITORIZAREA ȘI CONTROLUL MEDIULUI DINTR-O SERĂ,
- coordonată de Ș.L. Dr. Ing. Eugen Dumitrașcu,
- prezentată în sesiunea IULIE 2016.

La elaborarea proiectului de licență, se consideră plagiat una dintre următoarele acțiuni:

- reproducerea exactă a cuvintelor unui alt autor, dintr-o altă lucrare, în limba română sau prin traducere dintr-o altă limbă, dacă se omit ghilimele și referința precisă,
- redarea cu alte cuvinte, reformularea prin cuvinte proprii sau rezumarea ideilor din alte lucrări, dacă nu se indică sursa bibliografică,
- prezentarea unor date experimentale obținute sau a unor aplicații realizate de alți autori fără menționarea corectă a acestor surse,
- însușirea totală sau parțială a unei lucrări în care regulile de mai sus sunt respectate, dar care are alt autor.

Pentru evitarea acestor situații neplăcute se recomandă:

- plasarea între ghilimele a citatelor directe și indicarea referinței într-o listă corespunzătoare la sfârșitul lucrării,
- indicarea în text a reformulării unei idei, opinii sau teorii și corespunzător în lista de referințe a sursei originale de la care s-a făcut preluarea,
- precizarea sursei de la care s-au preluat date experimentale, descrieri tehnice, figuri, imagini, statistici, tabele et caetera,
- precizarea referințelor poate fi omisă dacă se folosesc informații sau teorii arhicunoscute, a căror paternitate este unanim cunoscută și acceptată.

Data,

Semnătura candidatului,



UNIVERSITATEA DIN CRAIOVA
Facultatea de Automatică, Calculatoare și Electronică
Departamentul de Calculatoare și Tehnologia Informației

Aprobat la data de
.....
Șef de departament,
Prof. dr. ing.
Marius BREZOVAN

PROIECTUL DE DIPLOMĂ

Numele și prenumele studentului/-ei:	Georgiana – Lucreția Stoian
Enunțul temei:	Monitorizarea și controlul mediului dintr-o seră
Datele de pornire:	Pentru dezvoltarea proiectului de licență, am avut ca punct de plecare domeniul de activitate al firmei WADSWORTH Control Systems ce se ocupa printre altele cu producerea de echipamente pentru controlul serelor.
Conținutul proiectului:	Conținutul proiectului se va baza pe următoarele domenii: 1: Introducere 2: Tehnologii hardware folosite 3: Tehnologii software folosite 4: Arhitectura hardware a proiectului 5: Arhitectura software a proiectului 6: Funcționalitatea proiectului 7: Concluzii 8: Bibliografie Anexa
Material grafic obligatoriu:	În redactarea lucrării s-au utilizat: ▪ diagrame ▪ scheme ▪ capturi
Consultații:	Periodice
Conducătorul științific (titlul, nume și prenume, semnătura):	Șef lucrări dr. ing. Eugen Dumitrașcu
Data eliberării temei:	01.12.2011
Termenul estimat de predare a proiectului:	01.06.2012

Data predării proiectului de către student și semnătura acestuia:	
---	--



REFERATUL CONDUCĂTORULUI ȘTIINȚIFIC

Numele și prenumele candidatului/-ei: Georgiana – Lucreția Stoian
 Specializarea: Calculatoare cu predare în limba română
 Titlul proiectului: Monitorizarea și controlul mediului dintr-o seră
 Locația în care s-a realizat practica de documentare (se bifează una sau mai multe din opțiunile din dreapta):
 În facultate ☐
 În producție ☐
 În cercetare ☐
 Altă locație: [se detaliază]

În urma analizei lucrării candidatului au fost constatate următoarele:

Nivelul documentării		Insuficient <input type="checkbox"/>	Satisfăcător <input type="checkbox"/>	Bine <input type="checkbox"/>	Foarte bine <input type="checkbox"/>
Tipul proiectului		Cercetare <input type="checkbox"/>	Proiectare <input type="checkbox"/>	Realizare practică <input type="checkbox"/>	Altul [se detaliază]
Aparatul matematic utilizat		Simplu <input type="checkbox"/>	Mediu <input type="checkbox"/>	Complex <input type="checkbox"/>	Absent <input type="checkbox"/>
Utilitate		Contract de cercetare <input type="checkbox"/>	Cercetare internă <input type="checkbox"/>	Utilare <input type="checkbox"/>	Altul [se detaliază]
Redactarea lucrării		Insuficient <input type="checkbox"/>	Satisfăcător <input type="checkbox"/>	Bine <input type="checkbox"/>	Foarte bine <input type="checkbox"/>
Partea grafică, desene		Insuficientă <input type="checkbox"/>	Satisfăcătoare <input type="checkbox"/>	Bună <input type="checkbox"/>	Foarte bună <input type="checkbox"/>
Realizarea practică	Contribuția autorului	Insuficientă <input type="checkbox"/>	Satisfăcătoare <input type="checkbox"/>	Mare <input type="checkbox"/>	Foarte mare <input type="checkbox"/>
	Complexitatea temei	Simplă <input type="checkbox"/>	Medie <input type="checkbox"/>	Mare <input type="checkbox"/>	Complexă <input type="checkbox"/>
	Analiza cerințelor	Insuficient <input type="checkbox"/>	Satisfăcător <input type="checkbox"/>	Bine <input type="checkbox"/>	Foarte bine <input type="checkbox"/>
	Arhitectura	Simplă <input type="checkbox"/>	Medie <input type="checkbox"/>	Mare <input type="checkbox"/>	Complexă <input type="checkbox"/>
	Întocmirea specificațiilor funcționale	Insuficientă <input type="checkbox"/>	Satisfăcătoare <input type="checkbox"/>	Bună <input type="checkbox"/>	Foarte bună <input type="checkbox"/>
	Implementarea	Insuficientă <input type="checkbox"/>	Satisfăcătoare <input type="checkbox"/>	Bună <input type="checkbox"/>	Foarte bună <input type="checkbox"/>
	Testarea	Insuficientă <input type="checkbox"/>	Satisfăcătoare <input type="checkbox"/>	Bună <input type="checkbox"/>	Foarte bună <input type="checkbox"/>
	Funcționarea	Da <input type="checkbox"/>	Parțială <input type="checkbox"/>	Nu <input type="checkbox"/>	
Rezultate experimentale		Experiment propriu		Preluare din bibliografie	

	<input type="checkbox"/>		<input type="checkbox"/>	
Bibliografie	Cărți	Reviste	Articole	Referințe web
Comentarii și observații				

În concluzie, se propune:

ADMITEREA PROIECTULUI <input type="checkbox"/>	RESPINGEREA PROIECTULUI <input type="checkbox"/>
--	--

Data,

Semnătura conducătorului științific,

REZUMATUL PROIECTULUI

O seră este o construcție specială amenajată pentru adăpostirea și cultivarea plantelor, având un rol deosebit de important în agricultură.

Obiectivul acestui proiect este de a monitoriza și controla mediul dintr-o seră. Astfel se poate menține constantă temperatura mediului pornind sau oprind diverse surse de căldură, se poate păstra un anumit procent de umiditate al solului pornind pompe de apă atunci când solul este prea uscat, sau se poate controla nivelul fotoradiațiilor pentru a facilita fotosinteza.

În dezvoltarea acestui proiect se vor implementa algoritmi pentru:

- gestionarea temperaturii;
- gestionarea umidității;
- gestionarea luminii;

Sistemul ce va fi capabil să facă toate aceste lucruri va fi controlat cu ajutorul unui microcontroler ATmega2560.

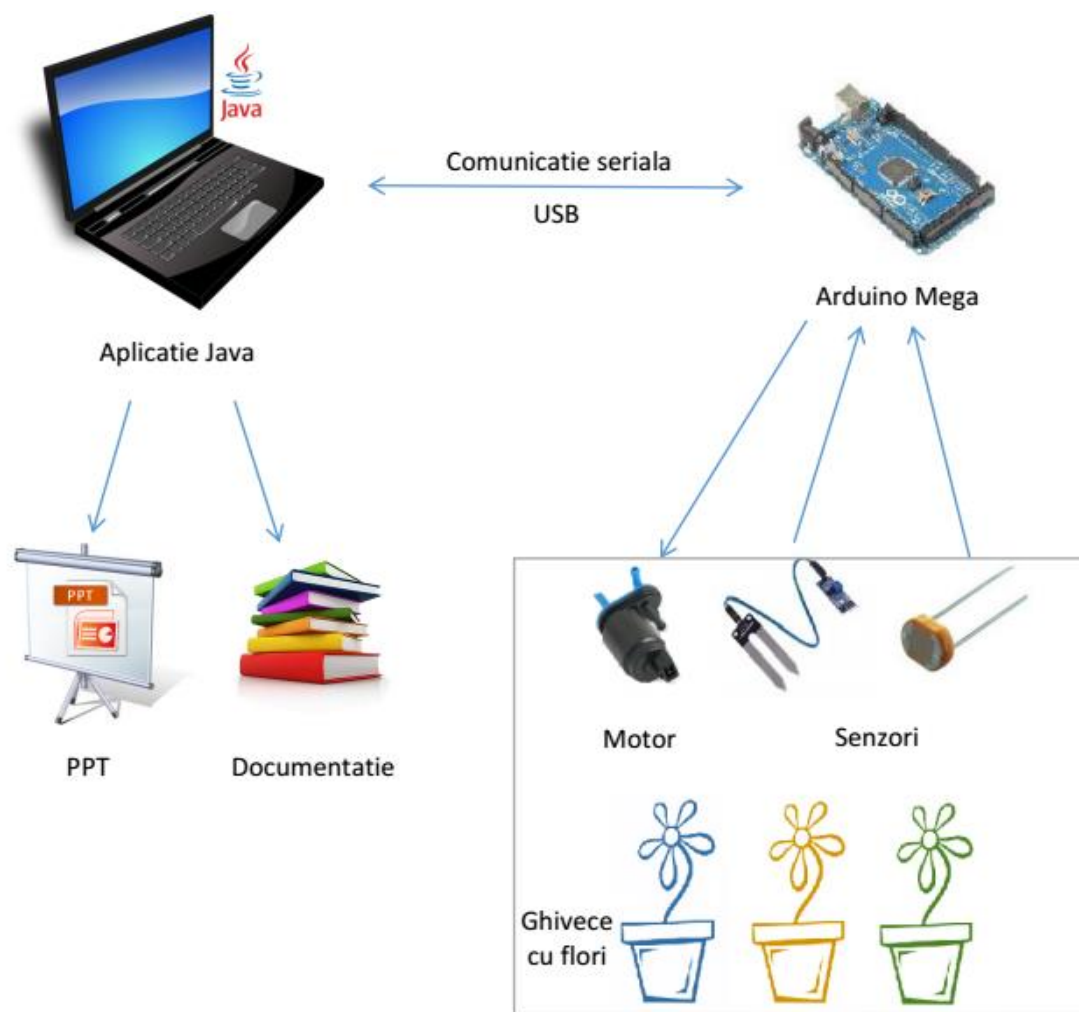
Se vor folosi:

- senzori de umiditate – pentru a monitoriza în permanență nivelul de apă din sol. Pe baza valorilor primite de la acești senzori se va hotărî dacă se va porni sau nu o pompă de apă.
- senzori de temperatură – pentru a menține constantă temperatura. Pe baza valorilor recepționate de la acești senzori se va porni sau nu o sursă de căldură.
- senzori de lumină – pentru a gestionata cantitatea de lumină necesară plantelor. În cazul în care nu este suficientă lumina solară, se vor porni Led-uri.

Se va dezvolta o aplicație desktop, în Java. Senzorii vor trimite în timp real către aplicația Java valorile acestora.

Aplicația Java constituie Panoul de control pentru User. Aici, un user poate să observe în permanență parametrii mediului din seră, și să poată acționa manual anumite comenzi. Aplicația Java va conține o interfață prietenoasă cu utilizatorul, astfel încât să fie ușor de folosit. Această aplicație poate să conțină și o bază de date unde se stochează datele pe o perioadă de timp și pe baza căreia se pot genera anumite statistici.

Arhitectura proiectului este următoarea:



Termenii cheie: seră, senzor, lumină, umiditate, temperatură, serial, Arduino, pompă apă, LED, ventilator.

MULȚUMIRI

În această secțiune, pot mulțumi întregii mele familii pentru susținere și în special soțului meu pentru ajutorul acordat în finalizarea acestui proiect, ce reprezintă încheierea cu succes a unui nou capitol din viața mea.

De asemenea, un ajutor foarte important a fost domnul profesor coordonator, care a avut încredere în mine și în acest proiect, care m-a sfătuit și susținut.

Și bineînțeles vreau să le mulțumesc prietenilor mei foarte apropiați, care nu m-au lăsat la greu și m-au ascultat atunci când am avut nevoie.

Vă mulțumesc tuturor încă odată tuturor, pentru întreg sprijinul acordat și pentru răbdarea de care a-ți dat dovadă atunci când am fost poate prea stresantă sau stresată.

PROLOG

CUPRINSUL

1	INTRODUCERE	1
1.1	SCOPUL.....	1
1.2	MOTIVAȚIA.....	1
2	COMPONENTE HARDWARE UTILIZATE	3
2.1	MICROCONTROLLER. DEFINIȚIE. STRUCTURĂ.....	3
2.2	MICROCONTROLLER-UL ATMEGA2560	6
2.3	MICROCONTROLLER-UL ARDUINO	8
2.4	SENZORI	9
2.4.1	<i>Senzori de temperatură</i>	<i>10</i>
2.4.2	<i>Senzori de umiditate</i>	<i>11</i>
2.4.3	<i>Senzori de lumină.....</i>	<i>11</i>
2.5	LED-URI.....	12
2.6	RELEE	13
2.7	POMPE HIDRAULICE	14
2.8	VENTILATOARE	15
3	COMPONENTE SOFTWARE UTILIZATE.....	16
3.1	ARDUINO IDE	16
3.2	NETBEANS IDE.....	17
3.3	ORACLE DATABASE.....	18
3.4	VISUAL STUDIO.....	18
3.5	FRITZING IDE	19
4	ARHITECTURA HARDWARE	20
4.1	ARHITECTURA HARDWARE GENERALĂ A SERIEI.....	20
4.2	CONECTAREA SENZORILOR	22
4.3	CONECTAREA POMPELOR DE APĂ.....	23
4.4	CONECTAREA VENTILATOARELOR	24
4.5	CONECTAREA LED-URILOR.....	25
4.6	ARDUINO MEGA2560	26
4.7	SURSE DE ALIMENTARE	27
5	ARHITECTURA SOFTWARE	28
5.1	LIBRĂRII UTILIZATE.....	28
5.1.1	<i>Librăria JFreeChart.....</i>	<i>28</i>

5.1.2	Librăria JDBC.....	29
5.1.3	Librăria PanamaHitek_Arduino	29
5.2	STRUCTURA BAZEI DE DATE.....	31
5.3	APLICAȚIA ARDUINO	32
5.3.1	Funcția setup.....	32
5.3.2	Funcția loop	33
5.4	APLICAȚIA JAVA.....	34
5.4.1	Pachetul Model.....	34
5.4.1.1	Clasa History.....	34
5.4.1.2	Clasa SensorValues.....	35
5.4.1.3	Clasa Users	35
5.4.2	Pachetul View	35
5.4.2.1	Clasa Application.....	35
5.4.2.2	Clasa Change_password	39
5.4.2.3	Clasa Chart.....	40
5.4.2.4	Clasa Login	44
5.4.2.5	Clasa MyPainter.....	45
5.4.2.6	Clasa Register	46
5.4.3	Pachetul Control.....	47
5.4.3.1	Application_listener.....	47
5.4.3.2	Change_password_listener.....	50
5.4.3.3	Chart_listener.....	51
5.4.3.4	Login_listener	51
5.4.3.5	Register_listener	52
5.4.3.6	SerialPort_listener.....	53
6	CONCLUZII	55
7	BIBLIOGRAFIE	57
8	REFERINȚE WEB.....	58
A.	LIVRĂRILE PROIECTULUI	59
B.	ARHITECTURA GENERALĂ A SEREI	60
C.	SCHEMA ELECTRONICĂ A SEREI	61
C.	CD / DVD	62
	INDEX	63

LISTA FIGURILOR

FIGURA 1. STRUCTURA UNUI MICROCONTROLLER	3
FIGURA 2. CONVERTOR ANALOGIC - DIGITAL / CONVERTOR NUMERIC - ANALOGIC.....	5
FIGURA 3. MICROCONTROLLER-UL ATMEGA2560	6
FIGURA 4. ATMEGA 2450-CONFIGURAȚIA PINILOR.....	7
FIGURA 5. LOGO-UL OFICIAL ARDUINO	8
FIGURA 6. SCHEMA BLOC A UNUI SENZOR	9
FIGURA 7. TERMISTOR NTC.....	10
FIGURA 8. SENZOR DE UMIDITATE A SOLULUI	11
FIGURA 9. FOTOREZISTOR	11
FIGURA 10. SIMBOLURILE FOTOREZISTORULUI	12
FIGURA 11. STRUCTURA UNUI LED	12
FIGURA 12. SCHEMA UNUI RELEU	13
FIGURA 13. MAȘINĂ CU ROȚI DINȚATE	15
FIGURA 14. VENTILATOR AXIAL	15
FIGURA 15. ARDUINO 1.6.8.....	17
FIGURA 16. NETBEANS IDE 8.1	18
FIGURA 17. ORACLE DATABASE 11G.....	18
FIGURA 18. MICROSOFT VISUAL STUDIO 2012.....	19
FIGURA 19. LOGO-UL FRITZING	19
FIGURA 20. MODUL DE INTERCONECTARE A COMPONENTELOR.....	21
FIGURA 21. SCHEMA ELECTRONICĂ A PROIECTULUI	21
FIGURA 22. CONECTAREA SENZORILOR.....	22
FIGURA 23. CONECTAREA POMPELOR DE APĂ PRIN INTERMEDIUL RELEELOR	23
FIGURA 24. RELEE SRD-05VDC-SL-C.....	24
FIGURA 25. CONECTAREA VENTILATOARELOR.....	24
FIGURA 26. CONECTAREA LED-URILOR.....	26
FIGURA 27. SHIELD ARDUINO MEGA.....	26
FIGURA 28. EXEMPLU DE GRAFIC DE TIP BARĂ	28
FIGURA 29. LOGO PANAMA HITEK	30
FIGURA 30. DIAGRAMA BAZEI DE DATE.....	31
FIGURA 31. MODELUL MVC.....	34
FIGURA 32. INTERFAȚA PRINCIPALĂ A APLICAȚIEI	36
FIGURA 33. ACCESAREA MENIULUI CONT	37
FIGURA 34. ACCESAREA MENIULUI	38

FIGURA 35. FEREASTRA DE SCHIMBARE A PAROLEI	39
FIGURA 36. GRAFIC TIP BARĂ 3D.....	40
FIGURA 37. GRAFIC TIP BARĂ	41
FIGURA 38. GRAFIC TIP LINIE.....	42
FIGURA 39. GRAFIC TIP LINIE 3D	43
FIGURA 40. GRAFIC TIP ZONĂ	43
FIGURA 41. FEREASTRA DE LOGARE.....	45
FIGURA 42. FEREASTRA DE ÎNREGISTRARE CONT	46
FIGURA 43. ȘTERGERE CONT.....	50
FIGURA 44. CONT ȘTERS	50
FIGURA 45. PAROLĂ SALVATĂ.....	50
FIGURA 46. PAROLE DIFERITE	51
FIGURA 47. CONT CREAT	52
FIGURA 48. PLANTĂ OFILITĂ.....	53
FIGURA 49. PLANTĂ USCATĂ	53
FIGURA 50. PLANTĂ NORMALĂ.....	53

LISTA TABELELOR

TABELUL 1. BARA DE BUTOANE A TOOL-ULUI ARDUINO IDE	17
TABELUL 2. CONECTAREA SENZORILOR LA PLĂCUȚĂ	22
TABELUL 3. CONECTAREA VENTILATOARELOR LA PLĂCUȚĂ	25
TABELUL 4. SPECIFICAȚII ARDUINO MEGA	27
TABELUL 5. CONFIGURAȚIA PINILOR	33

1 INTRODUCERE

1.1 Scopul

Proiectul de licență a fost ales pentru a facilita întreținerea unei sere, astfel prin acest sistem care se va ocupa de controlarea parametrilor, se va evita prezența umană.

Proiectul se bazează pe implementarea unui sistem capabil să monitorizeze și să controleze mediul dintr-o seră. Acest sistem va comunica cu o interfață Java, ce va constitui panoul de control al utilizatorului. User-ul va putea decide ce mod de funcționare dorește: manual sau automat. În modul automat sistemul va controla (pe baza informațiilor primite de la senzori) toate dispozitivele menite să țină sera în parametrii normali, iar în modul manual user-ul va fi cel ce va acționa aceste dispozitive, fie pe baza presupunerilor lui, fie pe baza informațiilor recepționate de la senzori (informații ce sunt vizibile în timp real în aplicația Java).

1.2 Motivația

O seră este o construcție specială amenajată pentru adăpostirea și cultivarea plantelor, având un rol deosebit de important în agricultură. De obicei are acoperiș (și cu pereți) din sticlă sau din material plastic. Ea se construiește cu scopul de a crea un microclimat optim și artificial, creșterii și dezvoltării plantelor pe tot parcursul anului.

Lumina într-o seră este folosită în principal pentru procesul de fotosinteză, în care plantele absorb și acumulează până la 10% din energia radiației solare incidente. În același timp, din dioxidul de carbon și apă, sub acțiunea luminii solare se formează carbohidrații și oxigenul molecular. Din moleculele carbohidraților se dezvoltă substanțele organice necesare pentru viața și creșterea plantelor. De asemenea și temperatura este un factor important în viața unei plante. Serele pot fi încălzite cu apă fierbinte, aburi, aer cald, radiație infraroșie sau produsele de ardere ale combustiei.

Luând în considerare toți acești factori, ar fi de folos un sistem automat ce se ocupa cu controlarea acestor parametri, fără a mai fi nevoie permanent de prezența umană.

De ce am ales un proiect ce presupune lucrul cu o seră este foarte simplu. Având în vedere că trăim într-un secol în care oamenii preferă să lase natură deoparte, în favoarea confortului, ar fi foarte interesant să pun accentul pe un lucru indispensabil nouă și anume natura ce este o sursă de oxigen.

Acum explicația pentru care am ales să monitorizez și să controlez o seră, este faptul că, datorită acestui secol al vitezei în care nu mai avem timp să ne ocupăm de mai multe lucruri deodată, ar fi utilă o aplicație care să facă asta în locul nostru.

Dar cum funcționalitățile acestui proiect au fost deja dezvoltate de firma WADSWORTH Control Systems, eu am hotărât să încerc să îmi demonstrez mie că pot realiza la scară mai mică acest proiect, cu toate resursele și cunoștințele ce sunt disponibile unui student aflat la început de drum. Aceasta este adevărata mea motivație pentru care am ales să duc la bun sfârșit ceea ce am ales să fac.

2 COMPONENTE HARDWARE UTILIZATE

2.1 Microcontroler. Definiție. Structură

Dealungul timpului prin procesul de miniaturizare a controlerelor destinate unei interacțiuni caracteritice cu mediul exterior fără a mai fi nevoie de intervenția unui operator uman, a fost posibil ca majoritatea componentelor constitutive ale acestora să fie înglobate într-un singur circuit, ce poartă numele de microcircuit sau microcontroller. [LAU12]

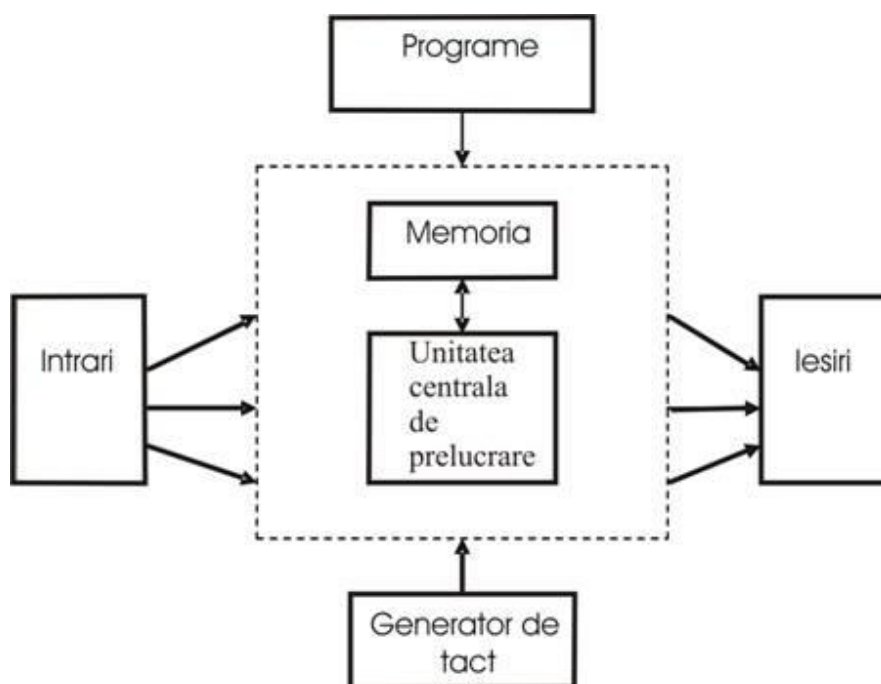


Figura 1. Structura unui microcontroller

Microcontrollerul este cel ce integrează pe lângă microprocesor și alte dispozitive, toate pe același chip. Cu alte cuvinte diferența dintre un microcontroller (μC) și un microprocesor (μP) este că microcontroller-ul este cel ce are în componența sa un microprocesor. [CIO03]

Un microcontroller este compus cel puțin din următoarele componente:

- Unitate centrală (CPU);
- O memorie locală tip ROM/PROM/EPROM/FLASH;
- O memorie RAM;
- Un sistem de întreruperi;
- I/O – intrari/ieșiri;

- Un sistem de timere;
- Un sistem de conversie analog numerică;
- Un sistem de conversie numeric analogic;
- Un comparator analogic;

Memoria FLASH

Memoria de program sau FLASH este memoria în care se stochează programul ce se dorește a fi executat, astfel microcontrollerul nu face altceva decât să urmărească instrucțiunile fără să le modifice în vreun fel. Scrierea acestei memorii este de obicei limitată.

Memoria RAM

Este o memorie volatilă (Random Access Memory), ceea ce presupune că datele sunt stocate doar temporar. Mai precis în cazul unui reset sau atunci când microcontroller-ul nu mai este alimentat, datele aflate în această memorie vor fi pierdute.

I/O

Intrările microcontroller-ului sunt de mai multe tipuri:

- Digitală: se pot primi numai date digitale și anume 0 sau 1;
- Analogică: tensiunea aplicată pe această intrare va fi convertită de un convertor A/D și ulterior va putea fi folosită de microcontroller ce va putea percepe mai multe valori, nu numai 1 și 0;

Ieșirile microcontroller-ului pot fi numai digitale.

Convertor Analogic - Digital

Acest convertor este folosit pentru a transforma un semnal analogic într-unul digital. Rezoluția disponibilă este de 8, 10, 12 biți.

Convertor Numeric – Analogic

Acest convertor este folosit pentru a transforma un semnal digital într-unul analogic. Cea mai răspândită tehnică este cea bazată pe modulația în factor de umplere (PWM – Pulse Width Modulation). [LAU12]

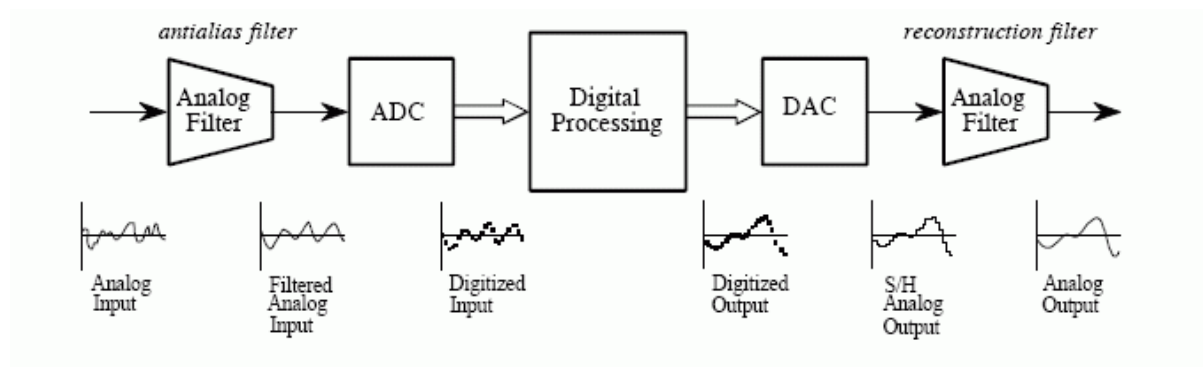


Figura 2. Convertor Analogic - Digital / Convertor Numeric - Analogic

Cele mai cunoscute familii de microcontrollere sunt:

- PIC
- AVR
- COP4 și COP8;
- Z8;
- Renesas;

2.2 Microcontroller-ul ATmega2560

Nucleul AVR combină un set de instrucțiuni cu 32 de regiștrii. Toți cei 32 de regiștrii sunt conectați direct la Unitatea Logică Aritmetică (ALU – Arithmetic Logic Unit), permițând ca doua registre independente să fie accesate într-o singură instrucțiune executată într-un singur ciclu de ceas. Arhitectura rezultată este mult mai eficientă fiind de aproximativ zece ori mai rapidă decât microcontrollere convenționale CISC. [ATM14]

Principala funcție a nucleului este să asigure corecta funcționare a programelor. CPU-ul trebuie să fie capabil să acceseze memoria, să facă calcule, să controleze perifericele și să gestioneze întreruperile. AVR-ul utilizează o arhitectura Harvard cu memorii și magistrale separate pentru program și date. Cât timp o instrucțiune este executată, o alta deja se citește din memoria de program. [ATM14]



Figura 3. Microcontroller-ul Atmega2560

Configurația microcontroller-ului Atmega2560 este:

- 256 KB memorie Flash;
- 4 KB memorie EEPROM;
- 8 KB memorie RAM;
- 86 pini de I/O;
- 12 canale PWM;
- 16 canale ADC;

Microcontroller-ul Atmega2560 suportă o suită întreagă de programe și tool-uri de dezvoltare incluzând: compilatoare C, asamblatoare macro, programe de debug/ simulare, kituri de evaluare.

Atmega2560 este un microcontroller puternic ce furnizează soluții foarte flexibile și ieftine pentru multe aplicații embedded.

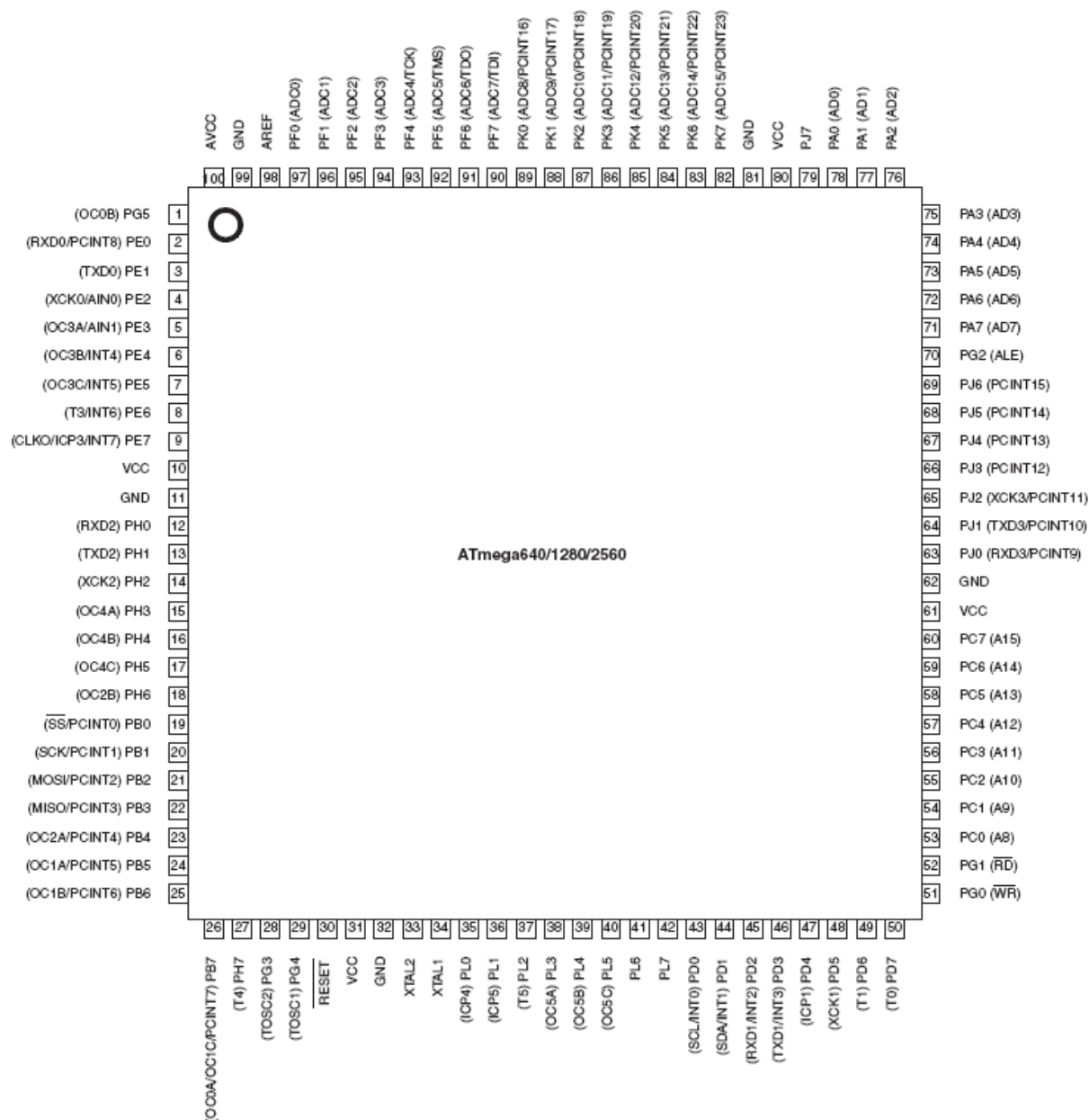


Figura 4. ATmega 2450-Configurația pinilor

2.3 Microcontrollere Arduino

Arduino este o companie open-source care le oferă studenților nenumarate avantaje deoarece produce atât plăcuțe de dezvoltare, cât și software ce este de mare ajutor programarii acestor plăcuțe. Pe lângă acest avantaj vizibil, există și o mare comunitate care dezvoltă diferite proiecte utilizând partea hardware cât și software furnizată de această companie.

Aceste plăci pun la dispoziția utilizatorilor pini de intrare/ieșire (I/O pins) digitali și analogici. Acestea dispun și de interfețe de comunicații seriale, pot dispune chiar de USB cu ajutorul căruia se pot descărca programe din calculatoarele dezvoltatorilor.

Primul Arduino a fost lansat în 2005, fiind o soluție simplă și ieftină pentru toate tipurile de utilizatori. O plăcuță a fost inițial compusă dintr-un microcontroller Atmel AVR, iar ulterior s-au adăugat cipuri de la alți producători. Aceasta mai conține pe lângă microprocesor: un scilator și un regulator liniar de 5 V. [ARD16]



Figura 5. Logo-ul oficial ARDUINO

Arduino este foarte simplu de utilizat astfel prin intermediul software-ului ARDUINO

IDE se poate scrie programul dorit după care acesta se poate descărca pe plăcuța de dezvoltare. Acest tool permite scrierea programelor în limbajul C/C++.

Arduino este astfel un mic calculator, deoarece se pot controla intrările și ieșirile de la chip, prin intermediul lui..

Există mai multe tipuri de plăcuțe Arduino:

- Arduino UNO (Atmega328);
- Arduino Mega (Arduino Mega2560);
- Arduino Leonardo (Atmega32u4);
- Arduino Diecimila (Atmega168);
- Arduino Nano (Atmega328);
- Arduino Duemilavove (Atmega168);

2.4 Senzori

Ce este un senzor? Senzorul poate fi numit și traductor. Mai precis senzorul preia informații din mediu și le furnizează mai departe microcontroller-elor. În acest mod semnalele neelectrice sunt convertite în semnale electrice ce pot fi interpretate.

Atunci când se dorește alegerea unui senzor trebuie să urmărim o serie de parametri, ca spre exemplu: costul, greutatea, consumul de energie, dimensiunile, precizia, sensibilitatea, domeniul de măsurare, viteza de răspuns, stabilitate. Există în schimb o mare varietate de senzori, care au caracteristici specifice grupei din care fac parte.[DUM06]

Putem privi un senzor ca pe o „cutie neagră”, deoarece aplicăm niste intrări ce urmează a fi măsurate și la ieșire ne așteptăm la rezultate.

Clasificari

Există mai multe tipuri de senzori (circa 2000 tipuri distincte). Una dintre cele mai importante clasificări ale senzorilor este în funcție de tipul mărimii de ieșire:

- Analogici – semnalul de ieșire este proporțional cu cel de intrare;
- Digitali – semnalul de ieșire are un număr limitat de valori care permit cuantificarea semnalului de intrare;

Tipuri de senzori

Există nenumerate tipuri de senzori, printre cei mai cunoscuți și utilizați sunt:

- Senzori de turație;
- Senzori de umiditate;
- Senzori de presiune;
- Senzori de lumină;
- Senzori de temperatură;
- Senzori de distanță;
- Senzori de mișcare;

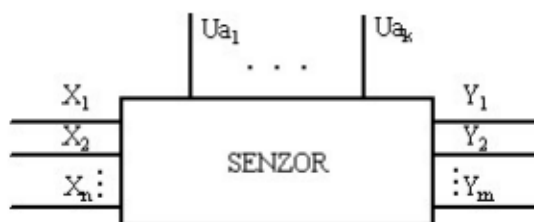


Figura 6. Schema bloc a unui senzor

- Senzori tactili;

Deoarece în acest proiect am folosit senzori de umiditate, temperatură și lumină, mai departe voi vorbi despre aceștia.

2.4.1 Senzori de temperatură

Senzorii de temperatură mai pot fi numiți și termometre. Aceștia pot fi de două tipuri și anume: pot avea contact direct cu obiectul asupra căruia se dorește să se execute măsurătoarea sau fără contact.

Termometrele au constanta de timp variabilă, datorită inerției termice ce depinde de rezistența termică dintre senzor și obiectul asupra căruia se dorește să se execute măsurătoarea, tipul senzorului, locul de montare al senzorului. Termometrele cu termistoare și termorezistoare au aceste constante de timp de aproape o secundă.

Senzorii de temperatură se clasifică în funcție de tipul ieșirii și de modul de comandă în:

- Analogici;
- Cu intrări și ieșiri digitale;
- Cu ieșiri digitale;



Figura 7. Termistor NTC

Pentru proiectul meu am ales să folosesc termistori (vezi Figura 8). Un termistor este un rezistor dependent de temperatură, realizat din materiale semiconductoare. Există două tipuri de termistoare și anume:

- NTC - au un coeficient negativ la variația rezistenței cu temperatura;
- PTC - au un coeficient pozitiv la variația rezistenței cu temperatura;

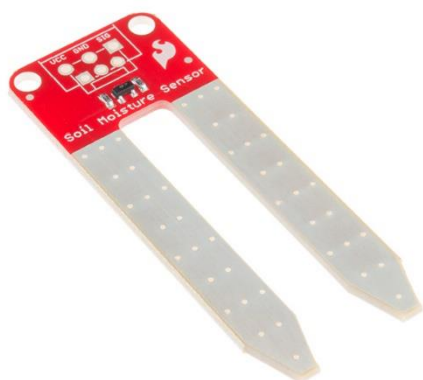
La termistoarele NTC, se realizează o scădere a rezistenței atunci când temperatura crește, iar la cele PTC este exact invers. [VOR09]

Eu am ales să folosesc un termistor ca cel din figură (vezi Figura 7). Acesta este un termistor NTC care dă o anumită tensiune de ieșire, în funcție de modificarea temperaturii.

2.4.2 Senzori de umiditate

Senzorii de umiditate sunt acei senzori care pot măsura nivelul de umiditate. Există senzori care pot măsura fie umiditatea aerului ambiental și gaze, fie pe cea a solului.

Umiditatea aerului este măsurată prin acei senzori care pot stabili câți vapori de apă există în aer. Această măsurare este mult mai dificilă decât măsurarea cu exactitate temperaturii și vitezei vântului, pentru că acești senzori sunt mai puțin sensibili. Instrumentele prin care se măsoară umiditatea aerului sunt: psihometru, higrometrul cu punct de rouă, higrometrul volumic, higrometrul cu fir de păr, higrometrul electrolitic, higrometrul cu interferență, higrometrul cu infraroșu, etc.[STE04]



Măsurarea umidității solului este foarte importantă în special în agricultură, unde este nevoie să se cunoască umiditatea solului și pe baza acestei informații să se ia decizii de pornire a sistemelor de irigații.

Pentru acest proiect am ales să folosesc senzori de umiditate a solului ca cei din Figura 8.

Figura 8. Senzor de umiditate a solului

2.4.3 Senzori de lumină

Senzorii de lumină sunt acei senzori ce pot detecta o mărime de ieșire de natură optică: difuzia, intensitatea luminoasă, absorbanta.

În cadrul acestui proiect am folosit ca senzor de lumină fotorezistorul, așa că aș dori să vorbesc mai multe despre el.

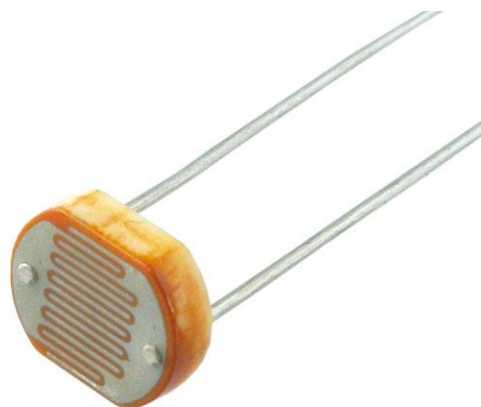


Figura 9. Fotorezistor

Fotorezistorul este un dispozitiv, a cărui rezistență se modifică sub acțiunea luminii, care cade pe suprafața acestuia. Acest dispozitiv electronic este format dintr-o peliculă de material semiconductor care este depusă pe un grătar metalic, fixat pe o placă izolatoare. [RUS15]

Rezistența fotorezistorului scade odată cu creșterea intensității luminii ce este aplicată pe suprafața acestuia.

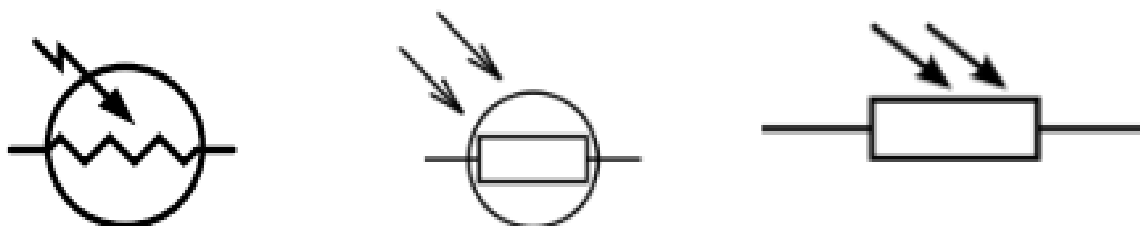


Figura 10. Simbolurile fotorezistorului

Dacă acest fotorezistor este conectat într-un circuit electric, el modifică intensitatea curentului. Astfel intensitatea curentului crește odată cu scăderea rezistenței electrice a acestuia. [RUS15]

2.5 LED-uri

Dioda luminiscentă (Light Emitting Diode) este dioda care poate produce lumină, atunci când este polarizată direct.

Această diodă este compusă dintr-o structură semiconductoare pn care are proprietatea de a emite lumină (vezi Figura 11 a).

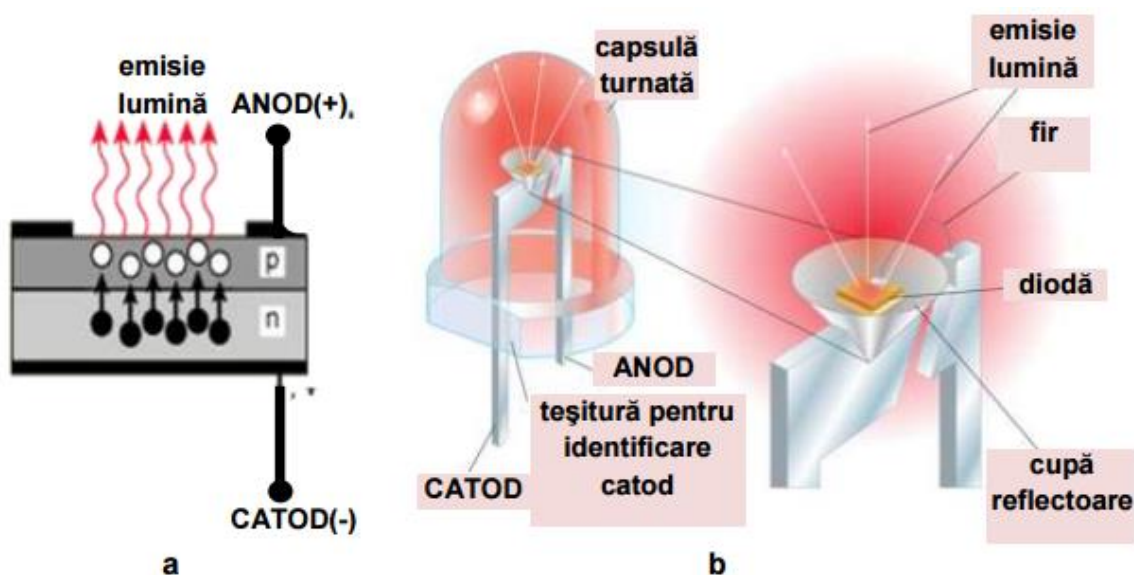


Figura 11. Structura unui LED

Această structură este pusă într-o cupă reflectoare (vezi Figura 11 b) care este conectată la terminale. Aceste terminale se numesc ANOD (+) și CATOD (-).

Culoarea luminii produse depinde de aliajul din care este făcută structura semiconductoră și anume:

- Roșu (Aluminiu – Galiu – Arsen);
- Verde (Aluminiu – Galiu – Fosfor);
- Albastru (Zinc – Seleniu);
- Ultraviolet (Carbon), etc.;

2.6 Relee

Releul este un aparat cu acțiune automată, care la o anumită valoare a intrării, produce o modificare a ieșirii, fiind cel puțin una dintre acestea două de natură electrică. Dacă ieșirea este electrică, atunci înseamnă că avem de a face cu un releu indirect.

Atunci când este sesizată o modificare a intrării, releu compară intrarea cu valoarea stabilită prin reglaj și transmite mai departe informația elementului de execuție. Acesta execută lucrul mecanic specific tipului de releu. [REL11]

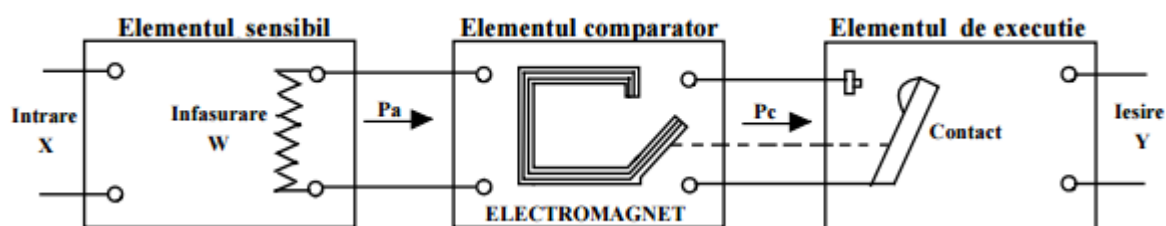


Figura 12. Schema unui releu

Clasificare

O clasificare importantă a releelor este în funcție de natura mărimii de intrare:

- Relee de curent;
- Relee de tensiune;

- Relee de putere;
- Relee de frecvență;
- Relee de timp;
-

2.7 Pompe hidraulice

Mașinile hidraulice sunt sisteme capabile să transforme energia mecanică în energie hidraulică și bineînțeles și invers. În cazul în care această energie hidraulică este transformată în energie mecanică, mașina poartă numele de motor hidraulic, iar în cazul în care această transformare este inversă, mașina se va numi pompă. [DIN08]

Mașinile hidraulice pot fi de doua tipuri:

- Rotative;
- Liniare:

Mașinile rotative produc o mișcare de rotație a unui arbore, iar cele liniare sunt formate dintr-un cilindru cu piston.

În acest proiect am folosit o pompă de apă cu roți dințate și aș dori să vorbesc puțin despre acestea.

Mașina cu roți dințate este o mașină hidraulică rotativă. Această mașină prezintă două camere separate și anume: camera de joasă presiune și camera de înaltă presiune. Pentru o pompă de apă lichidul circulă din camera de joasă presiune spre cea de înaltă presiune.

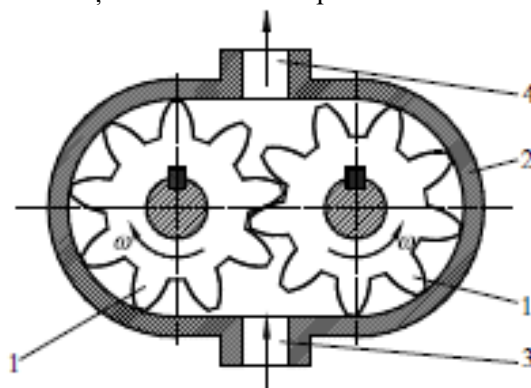


Figura 13. Mașină cu roți dințate

1 –roți dințate, 2 – carcasă, 3 – canalizație de admisie, 4 – canalizație de evacuare

În timpul funcționării ca pompă, lichidul este aspirat în spațiul dintre doi dinți și este transportat în camera de înaltă presiune. Deoarece lichidul nu este eliminat lin, ci este prins între doi dinți, apar curgeri de mare viteză și șocuri în angrenaj. Datorită acestui fapt pompa va produce mult zgomot., iar curgerile vor provoca uzura prematură a componentelor pompei. [DIN08]

2.8 Ventilatoare

Ventilatoarele sunt generatoare ce funcționează în medii gazoase și produc energie pneumatică ce este transformată din energie mecanică (preluată de la motorul de antrenare).

Cea mai importantă clasificare a ventilatoarelor este în funcție de direcția de mișcare a gazului. Astfel există două tipuri de ventilatoare: radiale și axiale.

În continuare voi vorbi despre ventilatoarele axiale, deoarece pe acestea le-am folosit în acest proiect.

Ventilatoarele axiale sunt cele în care particulele de gaz sunt transportate pe “traietorii paralele cu axul mașinii”, spre deosebire de cele radiale în care particulele de gaz ”sunt transportate pe traietorii ce se îndepartează de axul mașinii”. [NAG05]

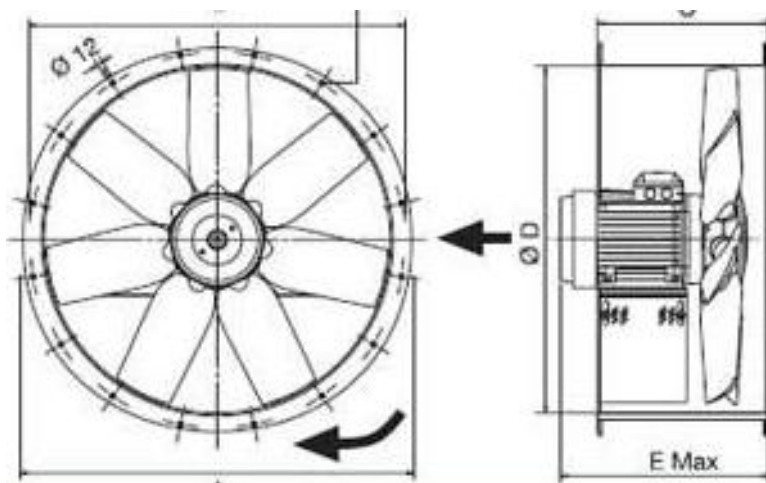


Figura 14. Ventilator axial

Acest tip de ventilator este compus dintr-un stator, rotor și o carcasă. Statorul ajută la conducerea favorabilă a curentului și poate lipsii, iar rotorul este constituit dintr-un butuc și palete.

3 COMPONENTE SOFTWARE UTILIZATE







3.1 Arduino IDE

Programele Arduino pot fi scrise în oricare limbaj de programare, sigura condiție ar fi că este necesar un compilator care să transforme acel cod scris într-un cod mașină pe care placa să îl poată interpreta.

Arduino IDE este o aplicație “cross-platform” ceea ce înseamnă că acest program este produs în mai multe variante, fiecare specifică pentru un anumit sistem de operare, astfel există acest software pentru Windows, Mac OS și Linux. Acest program este scris în Java și este proiectat astfel încât să fie ușor de folosit și de cei ce nu sunt familiarizați cu programarea.

Arduino IDE are un editor de cod ce este foarte util deoarece evidențiază termenii cheie și ține cont de potrivirea parantezelor. Un program scris în acest tool se numește “sketch”. [ARD16]

În tabelul de mai jos sunt descrise butoanele ce sunt disponibile în Arduino IDE.

Buton	Descriere
	Verify Verifică dacă codul prezintă erori.
	Upload Compilează codul și mai apoi îl încarcă pe microcontroler
	New Crează o nouă schiță
	Open Deschide un meniu cu toate schițele disponibile
	Save Salvează schița curentă
	Serial Monitor Deschide o fereastră numită Serial Monitor

Tabelul 1. Bara de butoane a tool-ului Arduino IDE

În Serial Monitor se pot vedea mesajele transmise serial de la placuța Arduino. Pentru a putea trimite date serial placuței trebuie avut grijă să fie aceeași configurație în această fereastră și în funcția setup. Odată ce configurația este aceeași mai rămâne doar să scriem mesajul și să apăsăm butonul de Send.



Figura 15. Arduino 1.6.8

Pentru a scrie codul C și pentru a îl pune pe microcontroller am folosit Arduino IDE versiunea 1.6.8, pentru Windows pe 64-biți.

3.2 NetBeans IDE

NetBeans este o platformă de dezvoltare ce este scrisă în Java. Această tool este cross- platform și are versiuni disponibile pentru Windows, Linux, Mac OS.

Acest program este de obicei utilizat pentru a scrie cod JAVA, dar nu numai. Astfel se pot scrie programe în PHP, C/C++, HTML5, C#. [NET16]



Acest program are în componența sa mai multe module ca spre exemplu:

- NetBeans Profiler – instrument ce ajută **Figura 16. NetBeans IDE 8.1** rie și să
optimizeze viteza.;
- Instrument de proiectare GUI ce permite facilitează crearea interfețelor;
- Editor NetBeans JavaScript ce oferă suport pentru JavaScript, Ajax, CSS;

Pentru realizarea aplicației desktop a proiectului s-a utilizat NetBeans IDE versiunea 8.1 pentru Windows pe 64-biți, iar codul a fost scris în JAVA.

3.3 Oracle Database

Oracle Database este un program ce permite gestionarea bazelor de date relaționare. Oracle stochează datele în mod logic, adică sub formă de tabele.

Acest software este disponibil în mai multe versiuni și pe mai multe sisteme de operare, ca spre exemplu: Windows, Linux. Acesta este simplu de utilizat și distribuit. [ORA14]



Figura 17. Oracle Database 11g

Oracle este foarte util pentru:

- Dezvoltatori care lucrează cu Python, Java, PHP, XML;
- Cei care au nevoie de o bază de date gratuită pentru a învăța și a implementa;
- Instituțiile de învățământ și studenți;

Acest tool a fost utilizat pentru a crea o bază de date pentru acest proiect. Această bază de date conține utilizatori precum și informații salvate de la senzori, informații pe baza cărora se pot face grafice.

3.4 Visual Studio

Visual Studio este un tool ce include un set de instrumente de dezvoltare pentru generarea de aplicații desktop, mobile, aplicații ASP.NET. Acest IDE este scris în C++ și C#.

Acest tool este disponibil numai pe Windows și suportă următoarele limbaje de programare: C++, VB.NET, C#, F#. [VIS16]

Am ales pentru acest proiect să folosesc Visual Studio 2012, pentru a face debug pe codul ce va fi scris pe microcontroller. Acest lucru a fost posibil numai după instalarea unui plugin. Acest plugin se numește Visual Micro și are versiunea 1606.7.10.



Figura 18. Microsoft Visual Studio 2012

3.5 Fritzing IDE

Fritzing IDE este un tool gratuit ce a fost dezvoltat de Universitatea de Științe Aplicate din Potsdam. Acest tool ajută dezvoltatorii oferindu-le oportunitatea de a realiza un prototip al circuitului înainte de a-l realiza fizic. [FRI16]

Codul sursă a acestui program este scris în C++ și poate fi descărcat și editat prin intermediul git-hub. Acest program este disponibil pe mai multe platforme și anume: Windows, Mac OS, Linux.



Figura 19. Logo-ul Fritzing

Am utilizat acest tool pentru a realiza schema hardware a aplicației practice din acest proiect și pentru asta am instalat ultima versiune a acestui tool (0.9.3b).

4 ARHITECTURA HARDWARE

4.1 Arhitectura hardware generală a serei

În acest capitol voi prezenta arhitectura serei și componentele folosite.

Componentele cele mai importante pe care le-am folosit sunt:

- Ghivece flori
- Microcontroller ATmega2560
- Senzori de temperatură
- Senzori de umiditate
- Senzori de lumină
- Pompe de apă
- Ventilatoare
- LED-uri
- Fire
- Sursă de tensiune
- Relee
- Laptop

În schema de mai jos se poate observa modul de interconectare a tuturor componentelor utilizate. Deși în schemă nu este figurate sursele de alimentare (datorită faptului că schema s-ar fi încărcat prea mult), ele există și sunt una de 5V și alta de 12V.

Am ales să utilizez pentru acest proiect numai doua plante, ceea ce înseamnă ca vor fi folosiți câte doi senzori din fiecare tip și bineînțeles câte două componente de control din fiecare tip.

Pe lângă schema din figura 20 am mai realizat o schemă electronică (vezi Figura 21) ce a fost și ea luată în calcul atunci când am realizat asamblarea proiectului. Odată realizată schema electronică, nu a mai rămas decât aranjarea într-un mod cât mai estetic a acestor componente ce au fost dispuse pe o machetă creată pentru fiecare plantă în parte.

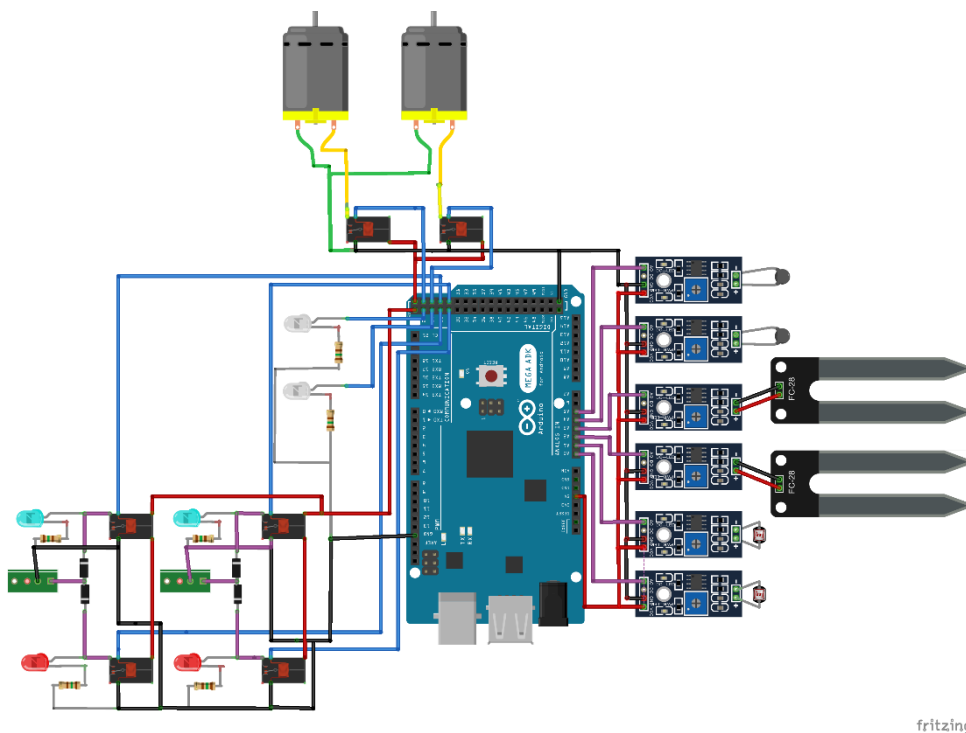


Figura 20. Modul de interconectare a componentelor

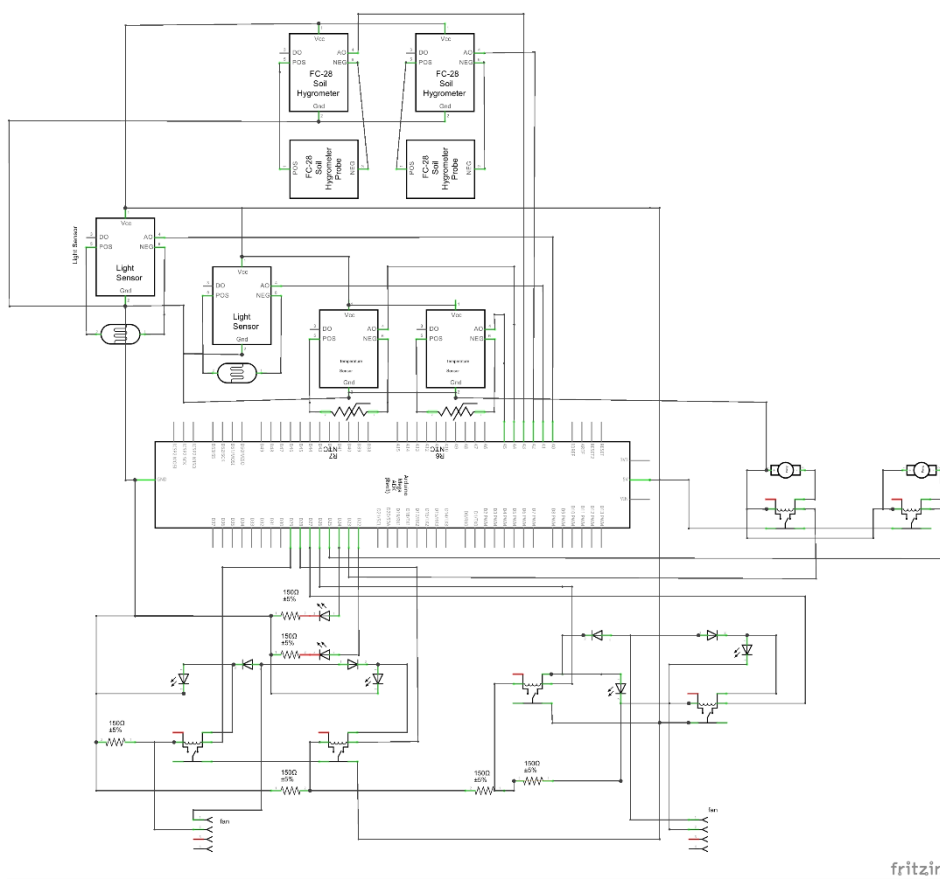


Figura 21. Schema electronică a proiectului

4.2 Conectarea senzorilor

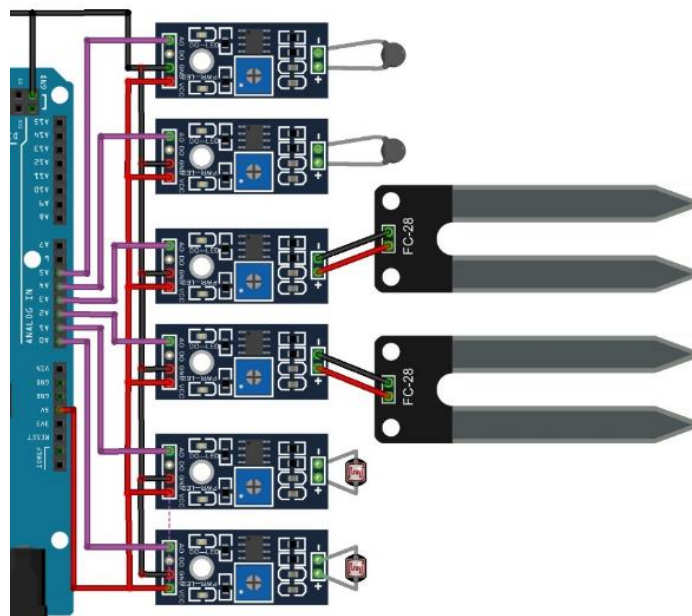


Figura 22. Conectarea senzorilor

Dupa cum se poate vedea în figura 22 de sus în jos primii doi senzori sunt cei de temperatură, urmatorii doi sunt cei de umiditate, iar ultimii sunt senzorii de lumină. Aceștia sunt conectați la placuță ca în tabelul 2.

Senzori	Arduino
Senzor de temperatură 1	A4
Senzor de temperatură 2	A5
Senzor de umiditate 1	A2
Senzor de umiditate 2	A3
Senzor de lumină 1	A0
Senzor de lumină 2	A1

Tabelul 2. Conectarea senzorilor la plăcuță

Toți cei șase senzori au 2 intrări și 2 ieșiri după cum urmează;

- VCC prin intermediul căreia se vor lega la plus;
- GND prin intermediul căreia se vor lega la minus;
- A0 prin intermediul căreia vor furniza date analogice;
- D0 prin intermediul căreia vor furniza date digitale.

Am ales să folosesc ieșirea analogică pentru a realiza în cod interpretarea datelor recepționate. Calibrarea senzorilor s-a efectuat manual, prin teste repetate aplicate sistemului.

Toți cei șase senzori prezintă un potențiomtru ce este util numai pentru ieșirea digitală și un comparator LM393. Așa cum am precizat într-un capitol anterior, senzorul de temperatură este defapt un termistor NTC, iar cel de lumină un fotorezistor.

Acești senzori sunt foarte importanți în proiect, deoarece pe baza informațiilor primite de la aceștia, se poate decide ce anume ar trebui făcut mai departe pentru a menține sera în parametrii normali.

4.3 Conectarea pompelor de apă

Am ales să utilizez o bandă de 8 rele de tipul celor din figura 24 (vor fi folosite numai 6 rele din cele 8) atât pentru conectarea pompelor cât și a ventilatoarelor, deoarece acestea au nevoie de o tensiune de alimentare mai mare de 5 V.

Așa cum am mai precizat pompele de apă sunt pompe cu roți dințate, ce vor fi alimentate la 12 V prin intermediul releelor.

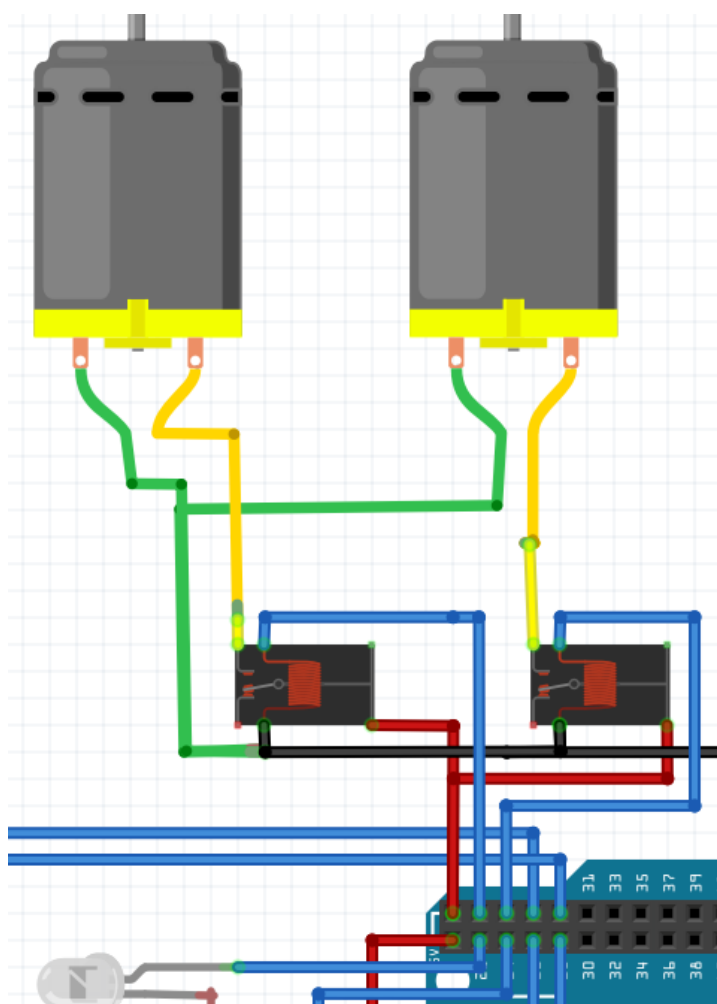


Figura 23. Conectarea pompelor de apă prin intermediul releelor



Figura 24. Relee srd-05vdc-sl-c

Releele ce vor fi folosite pentru ele se vor lega la pinul 23 respectiv pinul 25, ca în figura 23.

Releele au trei intrări respectiv și două ieșiri. Intrările sunt GND, VCC și comandă, iar cele două ieșiri vor fi active când una când cealaltă.

Am ales să folosesc ieșirea ce este activă atunci când pinul de comandă este zero, iar cealaltă ieșire este deconectată. Asta înseamnă ca programul va trimite 1 logic atunci când nu se dorește activarea ieșirii și 0 în rest.

4.4 Conectarea ventilatoarelor

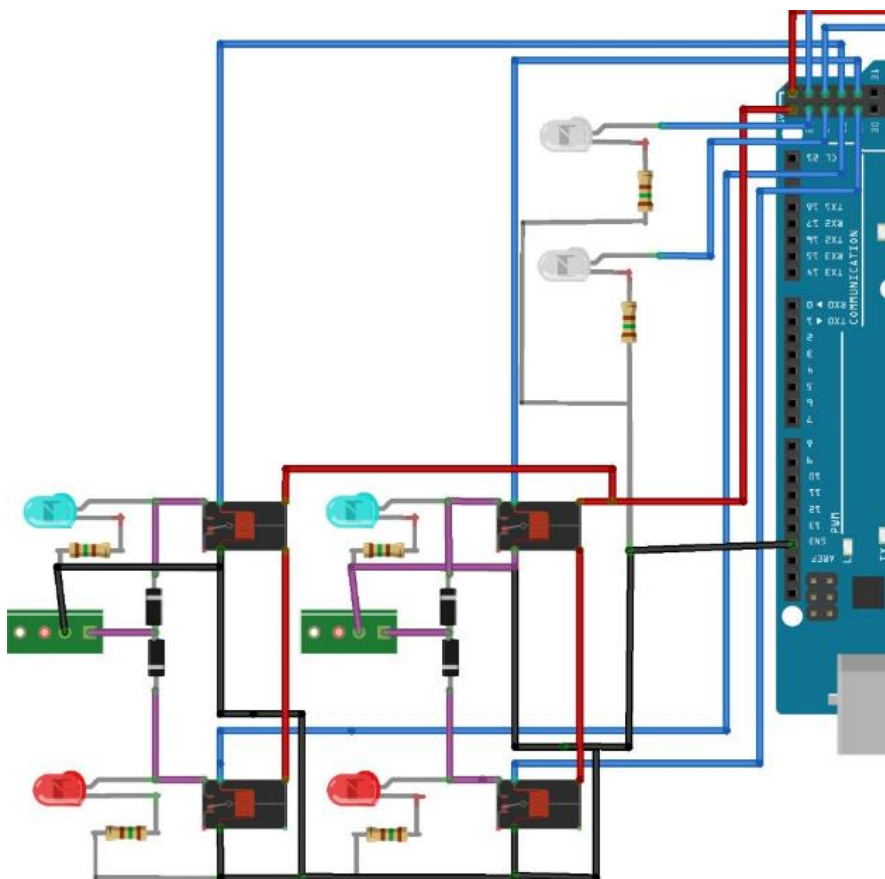


Figura 25. Conectarea ventilatoarelor

Ventilatoarele ca și pompele de apă au nevoie de curent mai mare și de aceea au fost conectate în ansamblu prin intermediul releelor. Aceste relee sunt folosite la fel cum am precizat în capitolul anterior.

După cum se poate vedea în figura 25 am utilizat două ventilatoare la care am legat două LED-uri albastre, respectiv două roșii.

Atunci când ventilatorul va fi pornit și LED-ul albastru aprins, va însemna că a fost pornită o aerotermă ce produce rece, iar atunci când LED-ul roșu va fi aprins și ventilatorul pornit va însemna că va fi pornită o aerotermă ce produce căldură. În continuare aș prefera să continui cu termenul de aeroterme.

Releele ce vor porni ventilatoarele sunt conectate la Arduino la fel ca în tabelul 3.

Aeroterme	Arduino
Aerotermă rece 1	27
Aerotermă cald 1	26
Aerotermă rece 2	29
Aerotermă cald 2	28

Tabelul 3. Conectarea ventilatoarelor la plăcuță

4.5 Conectarea LED-urilor

Am folosit două LED-uri UV pentru acest proiect, deoarece ajută plantele să crească mai repede. Ele sunt folosite în acest proiect pentru a suplinii cantitatea de lumină naturală, bineînțeles numai atunci când este nevoie.

Aceste LED-uri se vor lega cu anodul la pinii 22 respectiv 24, iar catodul va fi legat printr-o rezistență de 150Ω la ground (vezi figura 26). Aceste LED-uri se vor aprinde atunci când cantitatea de lumină este insuficientă.

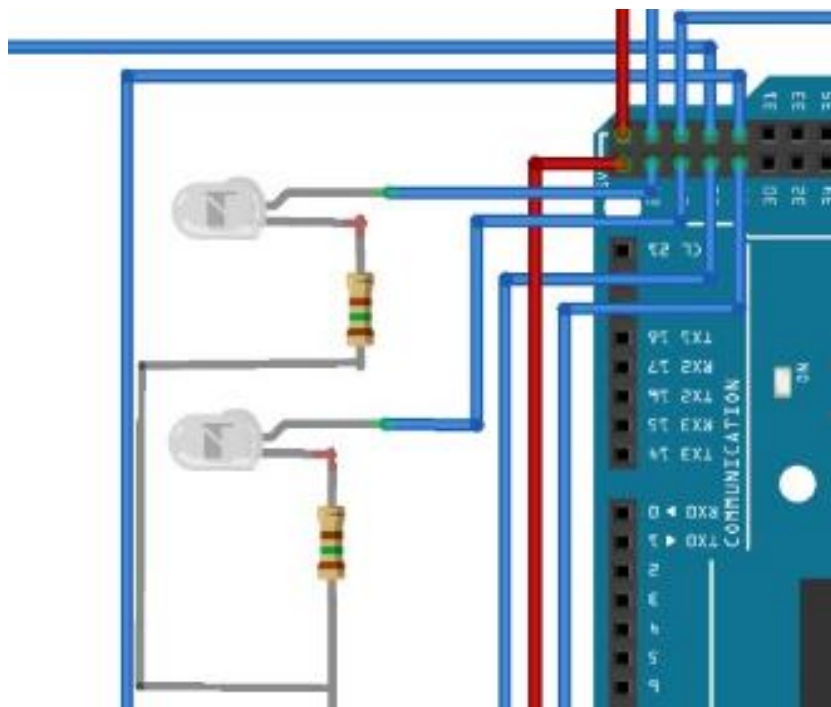


Figura 26. Conectarea LED-urilor

4.6 Arduino Mega2560

Arduino Mega2560 este o placuță de dezvoltare ce conține un microcontroller Atmega2560. Această placuță dispune de 54 pini digitali de intrare/ieșire (doar 14 dintre aceștia pot fi folosiți ca PWM), 16 intrări analogice, 4 porturi seriale hardware, un oscilator cu cristal de 16 MHz, o conexiune USB, un jack de putere și un buton de resetare. [MEG14]

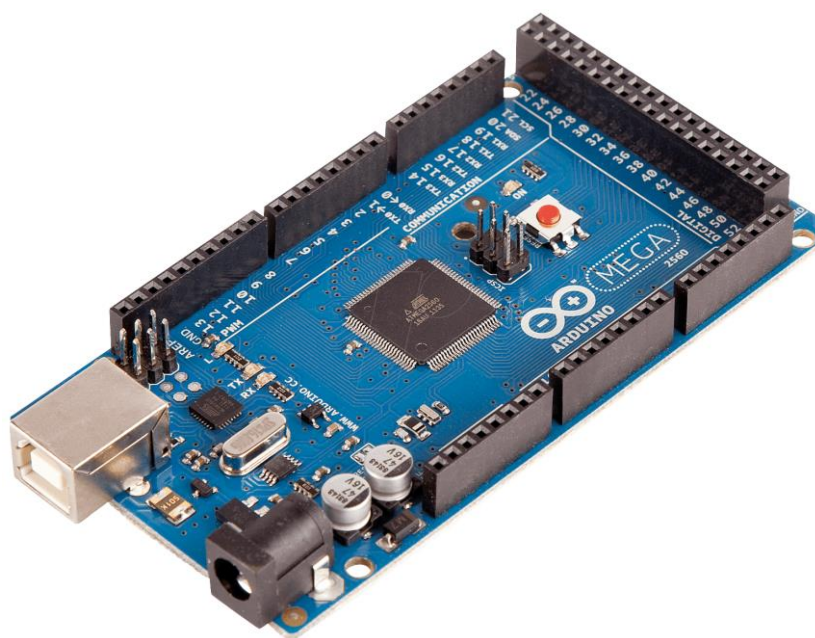


Figura 27. Shield Arduino Mega

Specificații:

Microcontroller	ATmega2560
Tensiune de lucru	5V
Tensiune de intrare (recomandat)	7-12V
Tensiune de intrare (limită)	6-20V
Pini digitali	54
Pini analogici	16
Intensitate de ieșire	40 mA
Intensitate de ieșire pe 3.3 V	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	12 MHz

Tabelul 4. Specificații Arduino Mega

4.7 Surse de alimentare

În acest proiect sunt utilizate ca surse de alimentare următoarele:

- Conexiunea plăcuței cu laptopul prin intermediul căreia se iau 5V;
- O sursă de tensiune externă de 5V;
- O sursă de tensiune externă de 12V;

De la această sursă de 5V se vor conecta defapt toți senzorii, de la cea de 12 V se vor conecta pompele de apă și ventilatoarele, iar restul componentelor inclusiv plăcuța se vor alimenta prin intermediul laptopului.

5 ARHITECTURA SOFTWARE

5.1 Librării utilizate

5.1.1 Librăria JFreeChart

Această bibliotecă este o bibliotecă gratuită scrisă în JAVA ce poate fi utilizată pentru a genera chart-uri, de aceea am utilizat-o în acest proiect. Scopul a fost generarea unor statistici pe baza informațiilor primite de la senzori.

Librăria JFreeChart este făcută pentru a fi utilizată spre exemplu în aplicații, servlet-uri și este distribuită cu tot codul sursă.

Tipurile de chart-uri oferite de această bibliotecă sunt:

- Grafice pie;
- Grafice tip bară;
- Grafice tip linie;
- Grafice Gantt;
- Grafice scatter, etc.

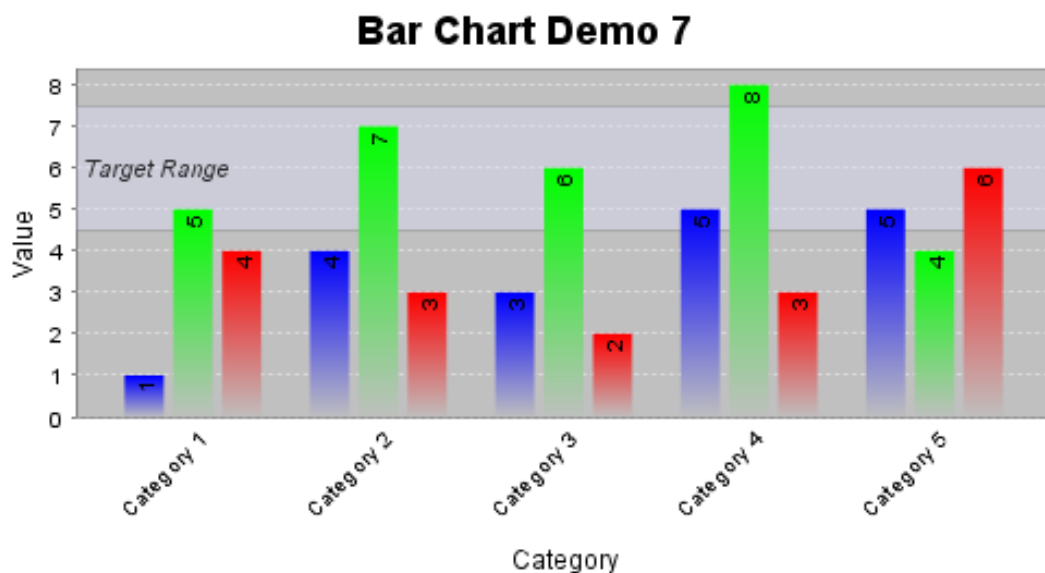


Figura 28. Exemplu de grafic de tip bară

Această librărie a fost descărcată de pe următorul site:

<http://jaist.dl.sourceforge.net/project/jfreechart/1.0.19/>

Dintre toate funcționalitățile pe care le are această librărie am ales să le folosesc doar pe următoarele:

- Datele sunt accesibile din orice parte a implementării;
- Zoom pe grafic;
- Disponibilitatea pentru aplicații JAVA;
- Mai multe tipuri de grafice. [GIL14]

5.1.2 Librăria JDBC

Această librărie este făcută pentru a permite unei aplicații JAVA să interacționeze cu o bază de date. În acest proiect am utilizat Oracle 11g împreună cu OJDBC7 ce poate fi descărcată de pe site-ul:

<http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html>.

Această librărie ne permite să:

- Deschidem o conexiune;
- Creeăm un obiect statement;
- Rulăm o interogare;
- Procesăm o interogare;
- Să facem modificări în baza de date;
- Să salvăm modificările;
- Să închidem o conexiune. [JDBC14]

5.1.3 Librăria PanamaHitek_Arduino

Această librărie a fost creată de Antony García González student al secției de inginerie electromecanică de la Universitatea Tehnologică din Panamá în scopul de putea distribui cunoștințe în spaniolă despre cele mai noi tehnologii și aplicațiile sale în inginerie. [GON14]

Librăria PanamHitek_Arduino se ocupă de realizarea comunicației și a conexiunilor cu Arduino. Conexiunea este realizată între o aplicație Java și un cod C++ ce va fi descărcat pe plăcuța Arduino. Comunicația dintre acestea se realizează serial. [GON14]

Am utilizat în acest proiect versiunea 2.7.0 acestei librării și dintre toate funcțiile disponibile le-am folosit decât pe următoarele:

- PanamaHitek_Arduino() ce este funcția constructor;
- arduinoRXTX(String PORT_NAME, int DATA_RATE, SerialPortEventListener events) - utilizată pentru a deschide o conexiune atât pentru recepționarea de date cât și pentru transmiterea de date; Această funcție primește ca parametrii numele portului prin intermediul căruia Arduino este conectat la calculator, rata de transmisie, și un eveniment serial cu ajutorul căruia aplicația Java va ști că a primit un mesaj;
- sendData(String data) – utilizată pentru a trimite date de la aplicația JAVA către Arduino; Această funcție primește ca parametru mesajul ce se dorește a fi trimis;
- MessageAvailable() – o funcție utilizată pentru a detecta atunci când există un mesaj; Această funcție are ca tip de return boolean, ceea ce înseamnă că dacă există un mesaj return-ul va fi TRUE, iar dacă nu FALSE;
- getSerialPorts() - această funcție va returna o listă de porturi disponibile;
- killArduinoConnection() – va închide conexiunea cu Arduino;
- printMessage() – va returna mesajul recepționat;



Această librărie a fost descărcată de pe site-ul:

Figura 29. Logo Panama Hitek

<https://sourceforge.net/projects/arduinojava/files/v2.7.0/>.

Librăria a fost tradusă, pentru că a fost dezvoltată și scrisă de un student spaniol.

5.2 Structura bazei de date

Am ales ca proiectul meu să aibă o listă cu utilizatori, să pot salva istoricul comenzilor efectuate și să pot salva informațiile de la senzori. Și ce e mai bun pentru asta decât o bază de date.

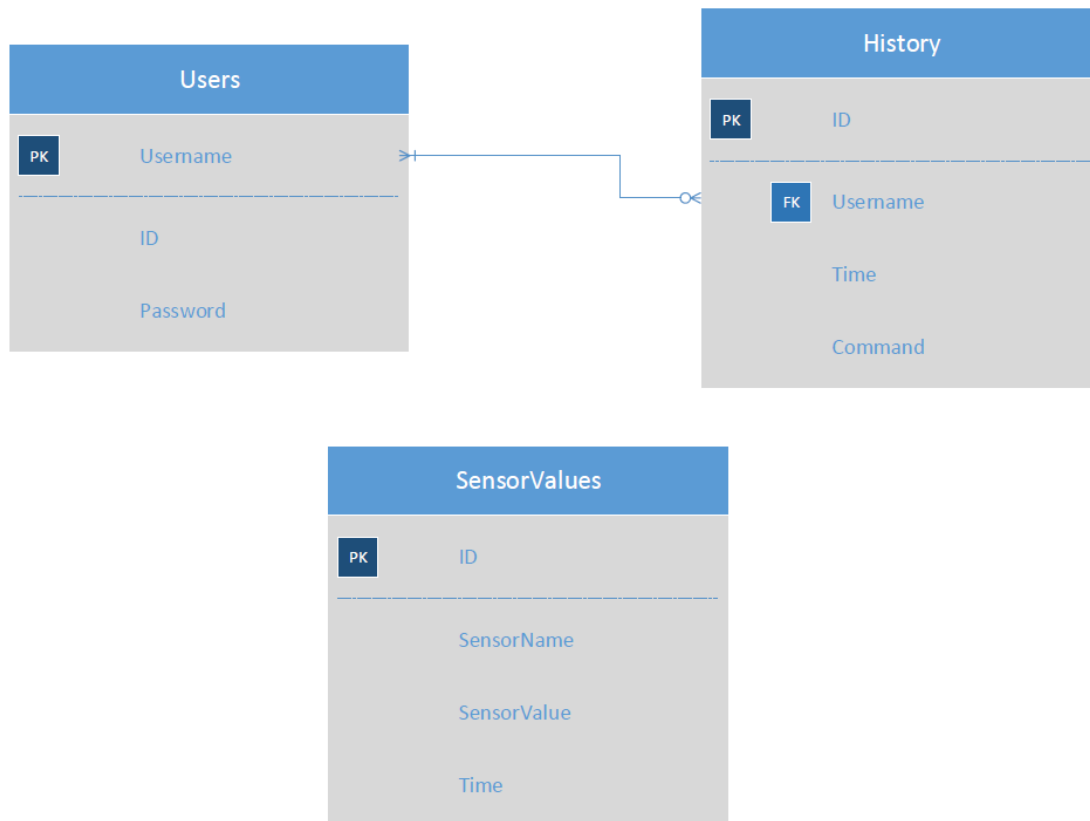


Figura 30. Diagrama bazei de date

După cum se poate în figura 30 am ales să am trei tabele. Primul tabel este cel al utilizatorilor în care se va stoca un id de utilizator, un username și o parolă și va avea ca și cheie primară username-ul ce va trebui să fie unic. Al doilea tabel este cel în care se va memora istoricul comenzilor astfel acesta va cuprinde un id, un username, o comandă executată precum și o data la care s-a executat comanda. Acest tabel are ca și cheie primară id-ul, iar cheia externă va fi username-ul.

Ultimul tabel este cel în care se vor stoca informațiile de la senzori. Acest tabel conține un id, un nume de senzor, o valoare a unui senzor precum și data salvării.

5.3 Aplicația Arduino

Aplicația Arduino este acea aplicație ce va fi descărcată pe plăcuța Arduino. Această aplicație se ocupă de configurarea pinilor precum și de scrierea/citirea pinilor.

Aplicația Arduino va primi date de la senzori și le va trimite serial către aplicația Java, dar tot odată va putea primi comenzi serial din aplicația Java. Aceasta aplicație utilizează două funcții și anume:

- Funcția setup;
- Funcția loop;

În subcapitolele următoare voi explica mai în detaliu aceste funcții.

5.3.1 Funcția setup

Această funcție se ocupă de configurarea pinilor de intrare/ ieșire și a stabilirii valorilor default a pinilor de ieșire și se execută o singură dată, atunci când sistemul este pornit.

Lista pinilor configurați poate fi observată în tabelul 5.

Număr pin	Tip	Utilizare
22	Output	Led-ul pentru planta 1
23	Output	Pompa de apă pentru planta 1
24	Output	Led-ul pentru planta 2
25	Output	Pompa de apă pentru planta 2
26	Output	Aerotema caldă pentru planta 1
27	Output	Aerotema rece pentru planta 1
28	Output	Aerotema caldă pentru planta 2
29	Output	Aerotema rece pentru planta 2
0	Input	Senzorul de lumină al plantei 1
1	Input	Senzorul de lumină al plantei 2
2	Input	Senzorul de umiditate al plantei 1

3	Input	Senzorul de umiditate al plantei 2
4	Input	Senzorul de temperatură al plantei 1
5	Input	Senzorul de temperatură al plantei 2

Tabelul 5. Configurația pinilor

Cum am mai zis aici sunt configurate valorile default, așa că am ales ca valori default pentru pinii ce controlează cele două Led-uri (pinul 22 și pinul 23) să fie LOW, iar pentru restul pinilor ce controlează componente prin intermediul releelor am ales valoarea HIGH (atunci când comanda este HIGH releul nu este cuplat).

Tot aici am configurat și seriala pe care am setat-o la un baudrate de 9600 bps.

5.3.2 Funcția loop

Această funcție este aceea care se execută ciclic adică pe fiecare clock al procesorului. În interiorul acesteia se realizează următoarele:

- Se citesc toți pinii de intrare;
- Se trimit serial valorile analogice ale acestora către aplicația Java;
- Se citește seriala;
- În cazul în care s-a recepționat ceva serial, se interpretează codul primit și în funcție de acest cod se iau următoarele decizii (aprinderea/stingerea Led-urilor, pornirea/oprirea pompelor de apă, pornirea/oprirea ventilatorului sau setarea modului automat/manual);

5.4 Aplicația JAVA

Aplicația Java este defapt aplicația cu care interacționează în mod direct utilizatorul. Am ales să folosesc pentru această aplicație modelul Model – View – Controller care este cel mai indicat pentru implementarea interfeței cu utilizatorul.

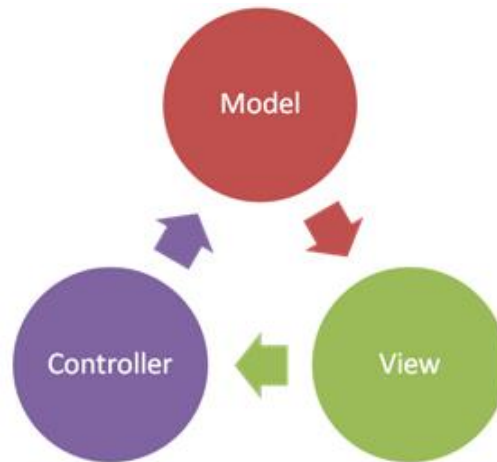


Figura 31. Modelul MVC

Mai departe voi descrie fiecare pachet și clasele din el.

5.4.1 Pachetul Model

Pachetul acesta se ocupă de gestiunea datelor și conține următoarele clase ce vor fi detaliate în subcapitolele următoare:

- History
- SensorValues
- Users

5.4.1.1 Clasa History

Clasa History conține ca și membrii de tip public un id, un utilizator de tipul clasei Users, o comandă și o dată de timp. Prin intermediul acestei clase se va salva un istoric al aplicației, având ca și corespondent un tabel din baza de date.

5.4.1.2 Clasa SensorValues

Clasa SensorValues conține ca și membrii de tip public un id, un nume de senzor, o valoare a unui senzor și o dată de timp. Cu ajutorul acestei clase se vor putea salva valorile senzorilor.

5.4.1.3 Clasa Users

Clasa Users conține ca și membrii un id, un nume de utilizator și o parolă. Cu ajutorul acestei clase se va putea ține o evidență a utilizatorilor.

5.4.2 Pachetul View

Pachetul View conține mai exacte ce poate vedea efectiv utilizatorul și conține următoarele clase:

- Application
- Change_password
- Chart
- Login
- MyPainter
- Register

Aceste clase vor fi detaliate în subcapitolele următoare.

5.4.2.1 Clasa Application

Această clasă se ocupă de dispunerea pe ecran a interfeței principale, din care se poate controla sera. Are ca membrii o clasă de tip SerialPort, una de tip PanamaHiteck_Arduino (această clasă face parte din librăria pe care am menționat-o), alta de tip Login și una de tip Application_listener, restul de membrii sunt panel-uri, label-uri, butoane, zone de text etc.

Această clasă are ca funcții un constructor ce se ocupă de inițializări și configurări, o funcție initComponents() ce se ocupă de setarea pozițiilor pentru toate componentele de interfață utilizate și o

funcție `app_start(Login login)` ce face vizibilă fereastra creată de această clasă și primește ca parametru o clasă `Login` prin intermediul căreia se pot duce mai departe setările inițiale.

Această clasă este apelată prin intermediul clasei `Login_listener` ce se ocupă de interpretarea evenimentelor din clasa `Login`.

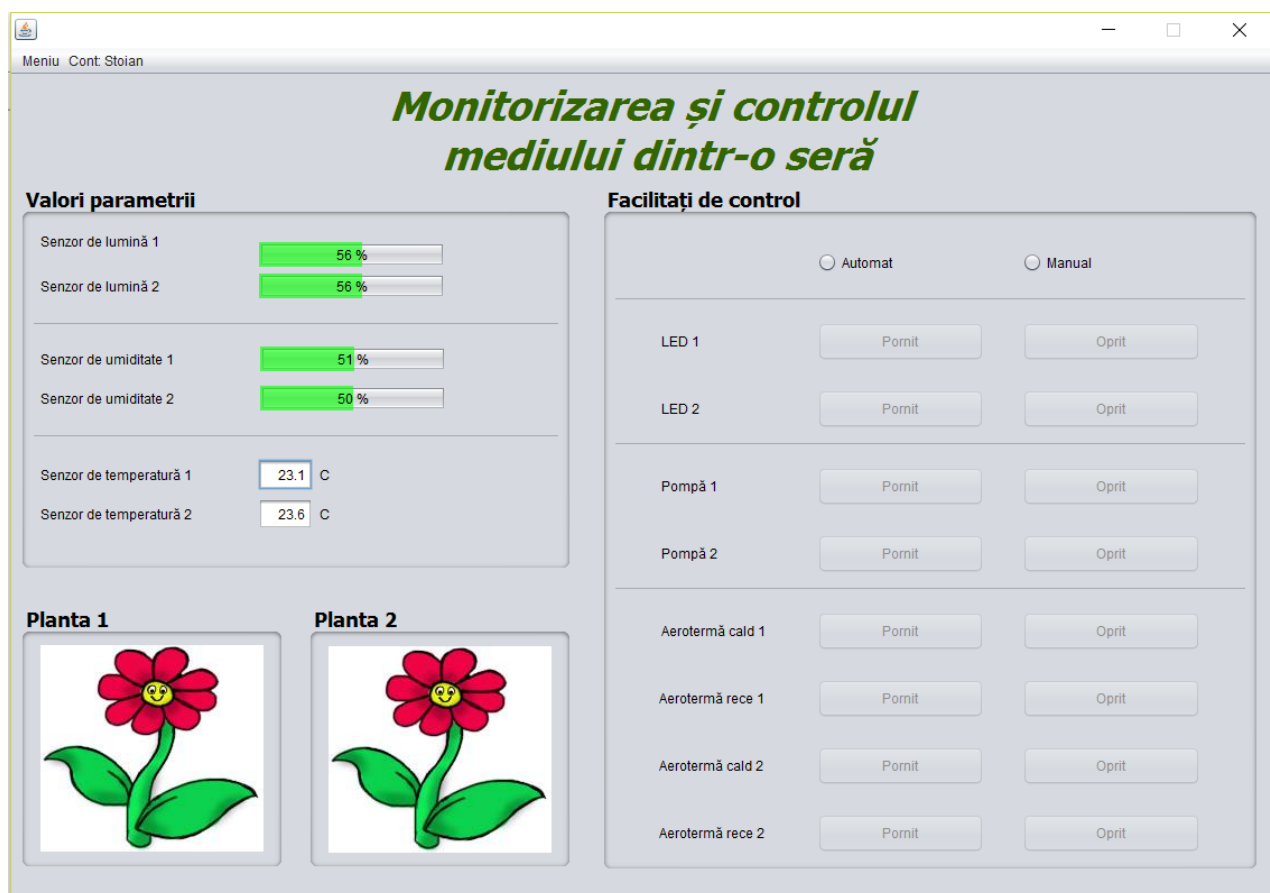


Figura 32. Interfața principală a aplicației

După cum se poate observa în figura 32 se află fereastra creată de această clasă. Pentru a crea tot ce se vede în imagine am folosit:

- `JMenuBar` - pentru a crea meniul din colțul stânga sus;
- `JMenuItem` - pentru a crea submeniurile ca în figura 33 și figura 34;
- `JLabel` - pentru titlul aplicației, pentru a spune ce fac diverse obiecte ca spre exemplu `Senzor de lumină 1`, `LED 2`, etc. Acest `JLabel` a fost utilizat și pentru a afișa imaginile cu starea

plantelor la un moment dat. Acest lucru a fost posibil numai prin utilizarea unei proprietăți, astfel am putut pune un icon căruia i-am atribuit calea de pe disk a pozei;

- JPanel – pentru a putea împărți întreaga fereastră în mai multe subpărți ca spre exemplu partea numită Valori parametrii, partea Facilități de control etc. Pentru a putea da nume acestor părți am utilizat o proprietate a JPanel-ului și anume bordura, astfel am ales o bordură ce conține titlu;
- JProgressBar – folosite pentru a afișa în procente valorile senzorilor de lumină și umiditate pentru cele două plante;
- JTextField – pentru a afișa temperatura mediului pentru cele două plante;
- JRadioButton – pentru a seta sistemul fie în starea automat fie în starea manual;
- JButton – pentru a porni sau opri diverse componente hardware destinate reglării parametrilor serei. Aceste butoane pot fi apăsate numai atunci când sistemul se află în starea manual;

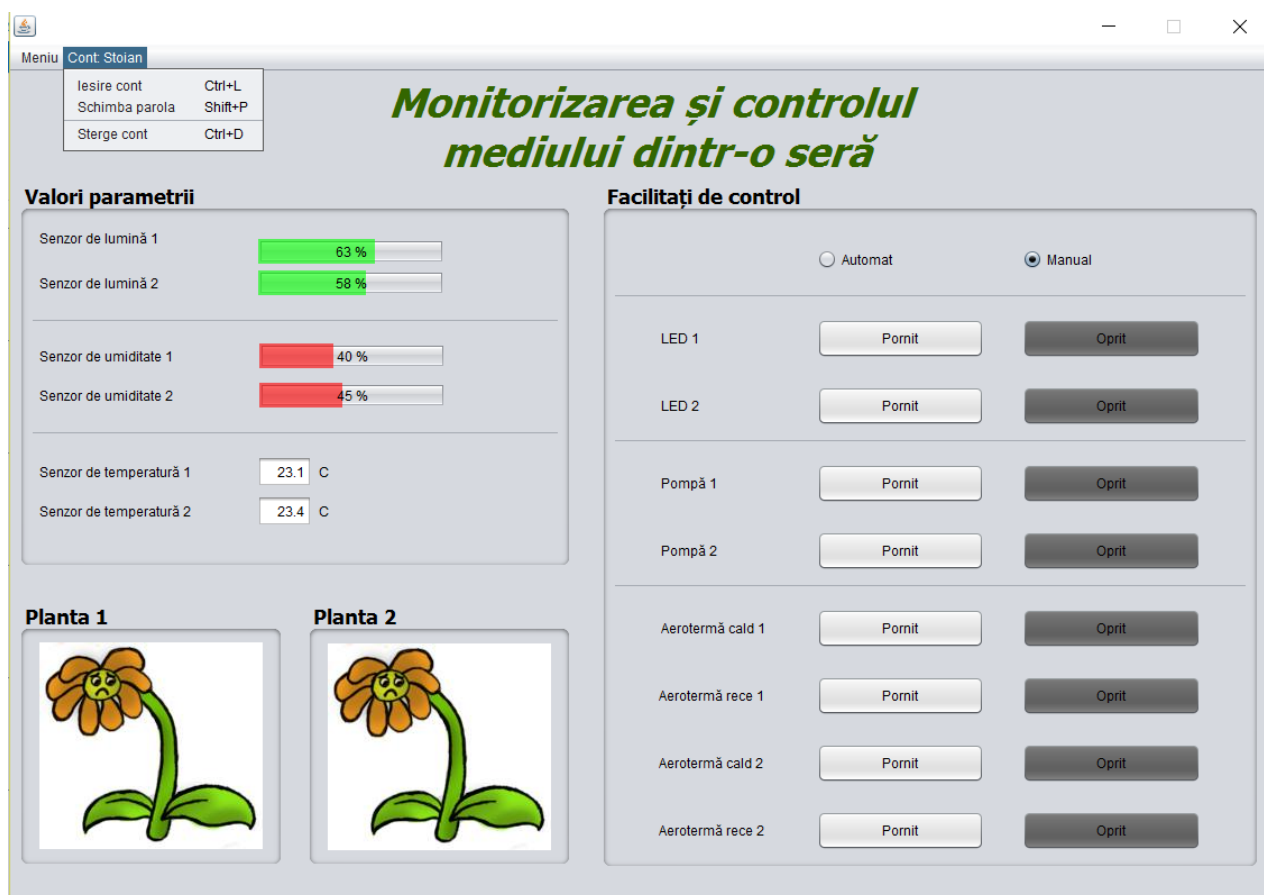


Figura 33. Accesarea meniului Cont

În figura 33 se poate vedea ce anume poate fi accesibil din meniul Cont, astfel dacă este apăsat submeniul:

- ieșire cont – se va închide această fereastră și se va deschide fereastra de logare;
- Schimbă parola – se va deschide o fereastră în care se va putea schimba parola;
- Sterge cont – se va deschide un JOptionPane prin care se va întreba utilizatorul dacă este sigur de alegerea făcută;

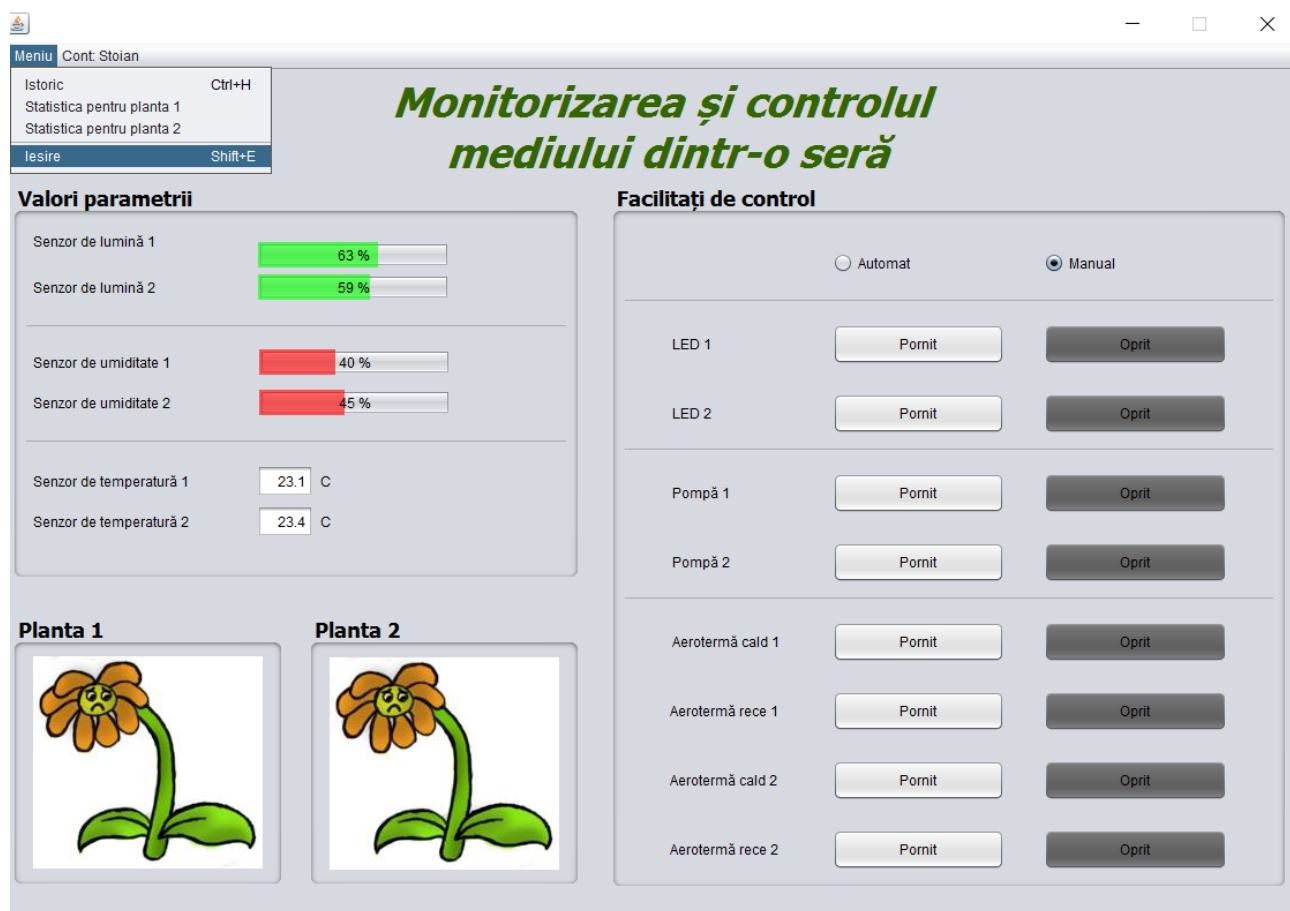



Figura 34. Accesarea meniului

În figura 34 se pot observa submeniurile disponibile din Meniu. Astfel dacă vom apăsa pe submeniul:

- Istoric – se va deschide o fereastră în care se va putea vedea istoricul;
- Statistică pentru planta 1 – se va deschide o fereastră în care se va putea observa un grafic pentru prima plantă, generat pe baza informațiilor salvate de la senzori;

- Statistică pentru planta 2 – se va deschide o fereastră în care se va putea observa un grafic pentru a doua plantă, generat pe baza informațiilor salvate de la senzori;
- Ieșire - se va închide aplicația la fel ca și prin apăsarea butonului  ;

5.4.2.2 Clasa Change_password

Clasa Change_password se ocupă de afișarea unei interfețe prin care utilizatorul își poate schimba parola. Această clasă are ca membrii o clasă de tip Application prin intermediul căreia poate accesa parola utilizatorului logat, o altă clasă de tip Change_password_listener cu ajutorul căreia se vor putea trata evenimentele apărute în fereastra generată de această clasă. Restul de membrii ai acestei clase sunt butoane, zone de text și label-uri.

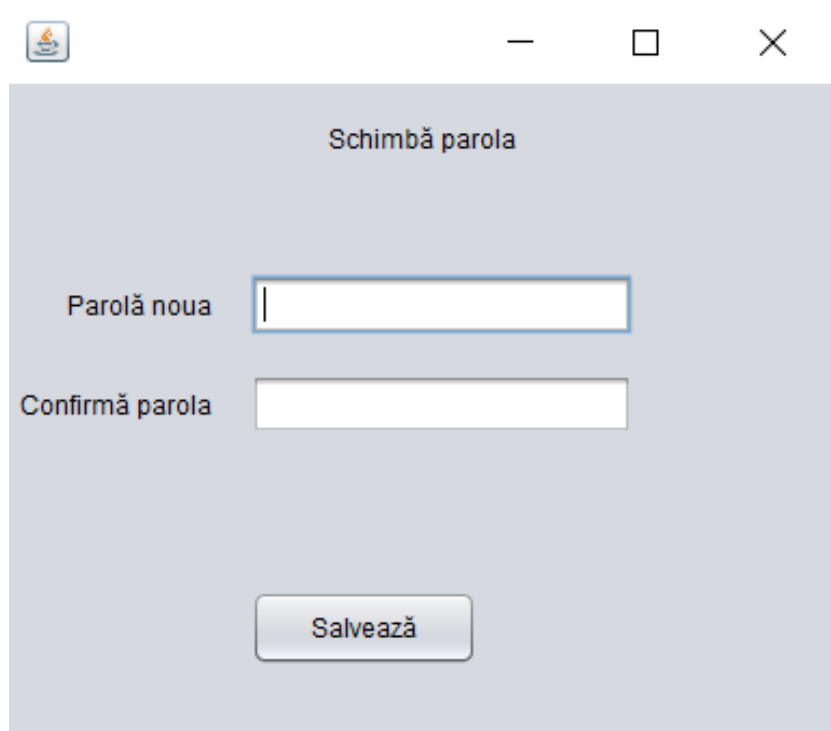


Figura 35. Fereastra de schimbare a parolei

Clasa aceasta are ca funcții un constructor ce se ocupă de inițializări și configurări și o funcție initComponents() ce se ocupă de setarea pozițiilor pentru toate componentele ce fac parte din această fereastră.

Această clasă este apelată prin intermediul clasei `Application_listener` ce se ocupă de interpretarea evenimentelor din clasa `Application`. Astfel fereastra generată de această clasă va fi vizibilă atunci când se va apăsa submeniul “Schimbă parola” sau prin apăsarea combinației de taste “Shift + P” din fereastra principală a aplicației.

În figura 35 se află fereastra generată de această clasă. Pentru crearea acesteia am folosit următoarele:

- `JLabel` – pentru numele ferestrei, precum și pentru numele căsuțelor de text;
- `JPasswordField` – pentru parole. A fost folosită această componentă, pentru a nu se vedea parola;
- `JButton` – pentru a crea butonul “Salvează”, prin apăsarea căruia va apărea un eveniment ce va fi tratat de clasa `Change_password_listener`;

5.4.2.3 Clasa Chart

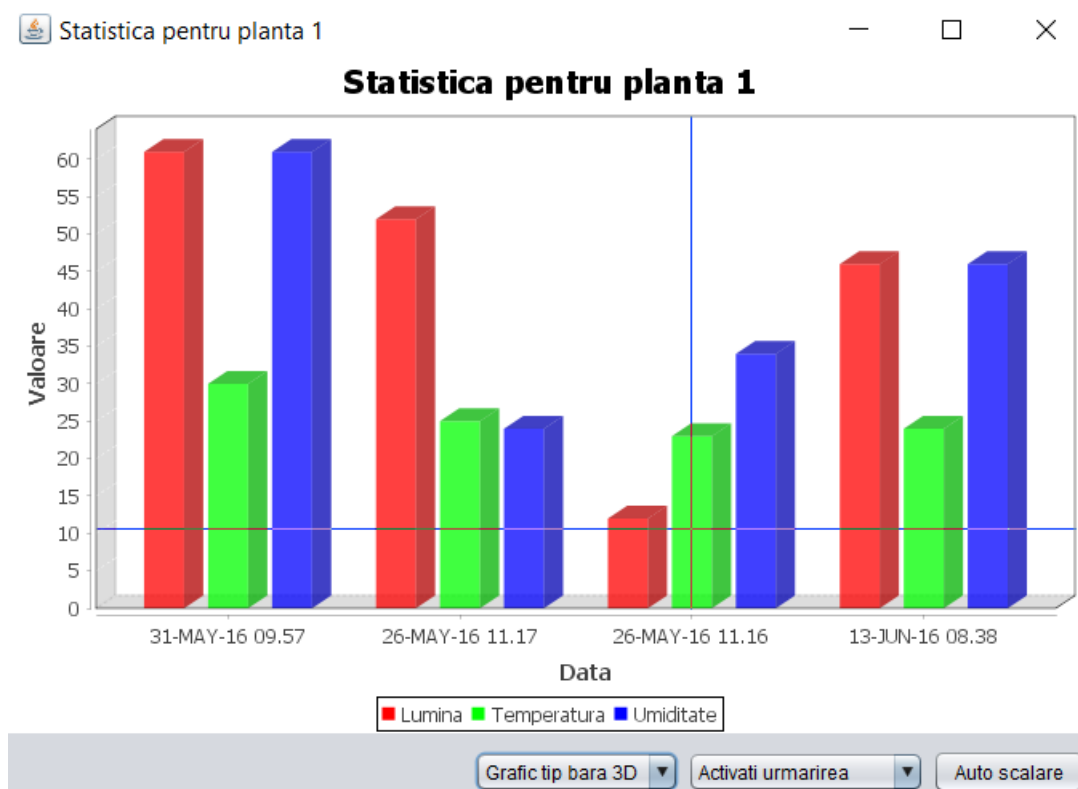


Figura 36. Grafic tip bară 3D

Clasa Chart se ocupă de afișarea unei interfețe prin care se vor putea vedea grafice. Această clasă are ca membrii o clasă de tip MYSQLCMD ce se ocupă cu executarea interogărilor către baza de date, o alta de tipul ChartPanel cu ajutorul căreia se realizează graficul, alta de tipul CategoryDataset care va reține toate datele graficului și nu în ultimul rând una de tipul Chart_listener ce se ocupă de tratarea evenimentelor acestei clase.

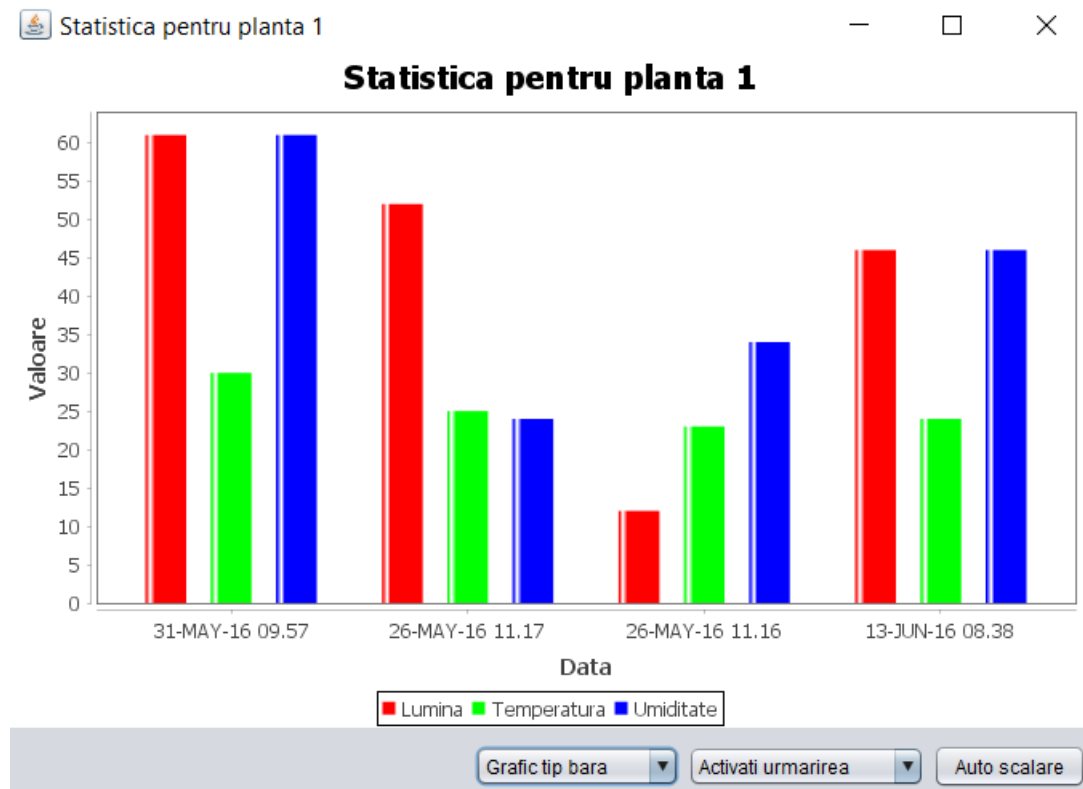


Figura 37. Grafic tip bară

Clasa aceasta are ca funcții:

- un constructor;
- o funcție createTrace() cu ajutorul căreia se va crea un JComboBox ce va permite să fie activată/dezactivată urmărirea. Prin acest lucru se va afișa sau nu un set de axe din dreptul săgeții de la mouse;
- o funcție createGraphic() cu ajutorul căreia se crea un JComboBox ce va permite schimbarea tipului de grafic. În acest proiect sunt disponibile cinci tipuri de grafice: grafic tip bară 3D, grafic tip linie 3D, grafic tip bară, grafic tip linie, grafic tip linie;

- o funcție `createZoom()` cu ajutorul căreia se va crea un buton prin apăsarea căruia graficul se va autoscala pentru a fi vizibil integral;
- o funcție `createChart()` ce are ca tip de return `ChartPanel`. Cu ajutorul acestei funcții se va crea efectiv graficul;
- o funcție `createDataset()` cu tipul de return `CategoryDataset`. În această funcție se va realiza interogarea bazei de date și preluarea datelor salvate de acolo;
- o funcție `rulează()` cu ajutorul căreia se va afișa fereastra.
- o funcție `setPlant(String plant)` prin apelare căreia se va ști pentru care plantă va trebui creat graficul;

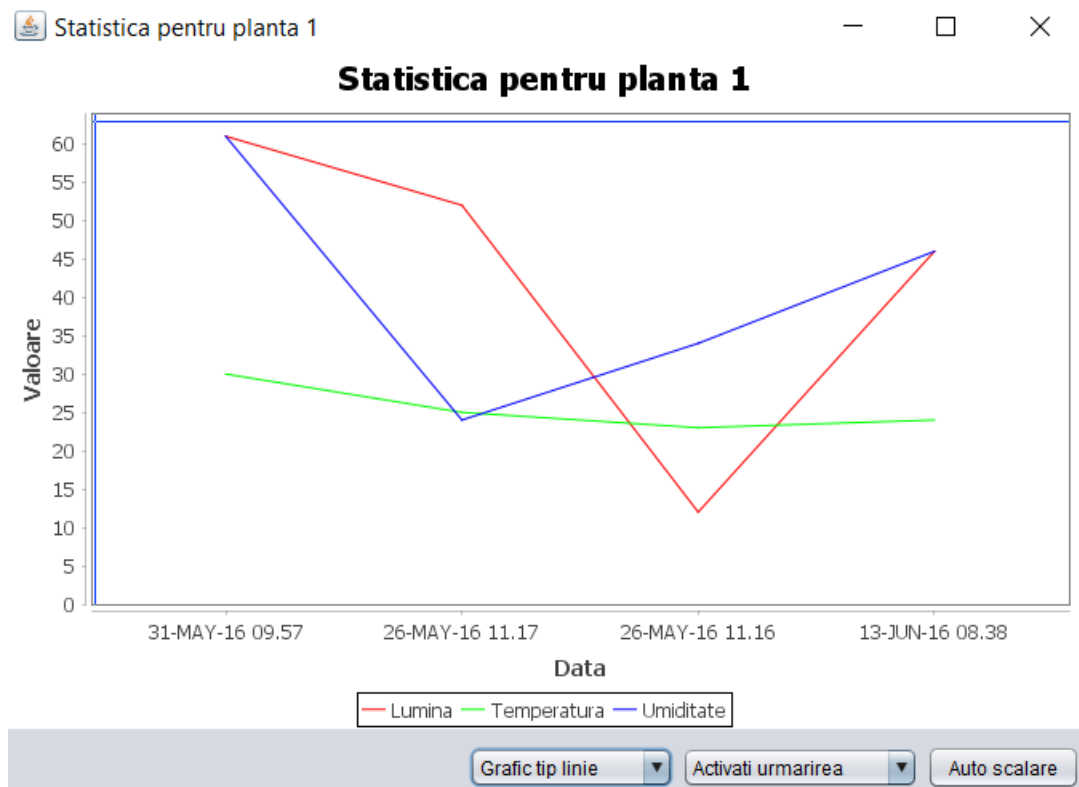


Figura 38. Grafic tip linie

Această clasă este apelată prin intermediul clasei `Application_listener` ce se ocupă de interpretarea evenimentelor din clasa `Application`. Astfel fereastra generată de această clasă va fi vizibilă atunci când se va apăsa submeniul “Statistică pentru planta 1” sau submeniul “Statistică pentru planta 2”. În funcție de butonul apăsător se va ști pentru ce plantă se va genera graficul.

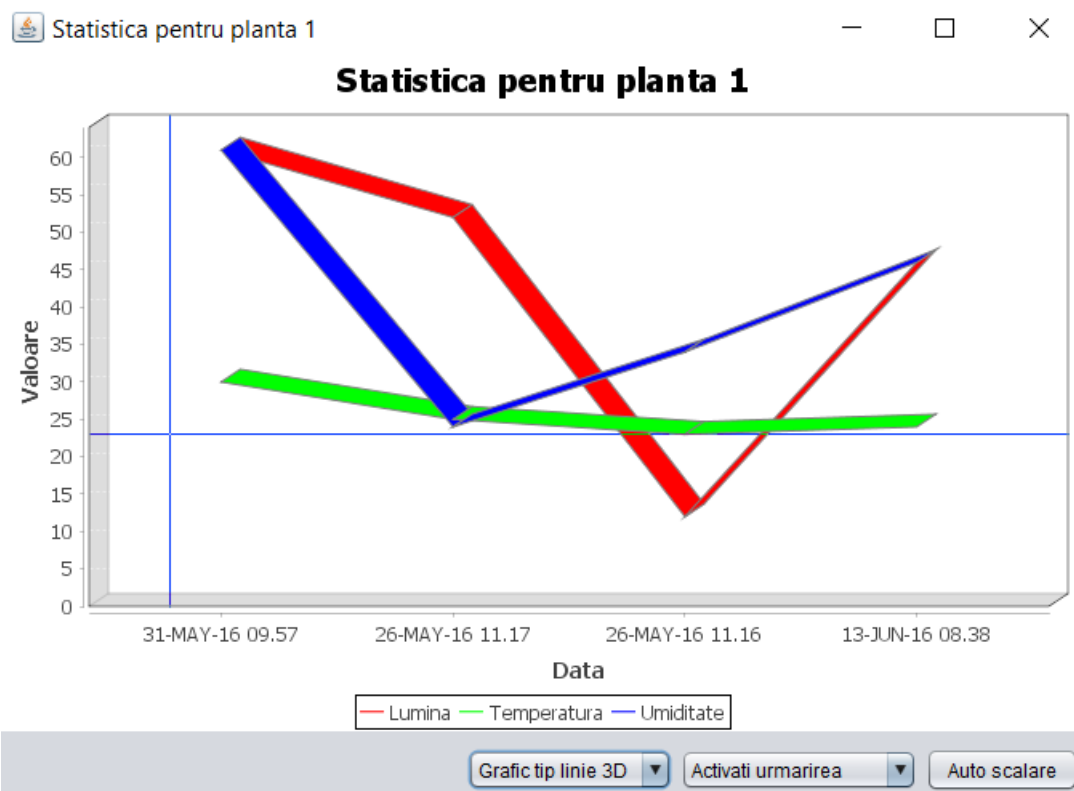


Figura 39. Grafic tip linie 3D

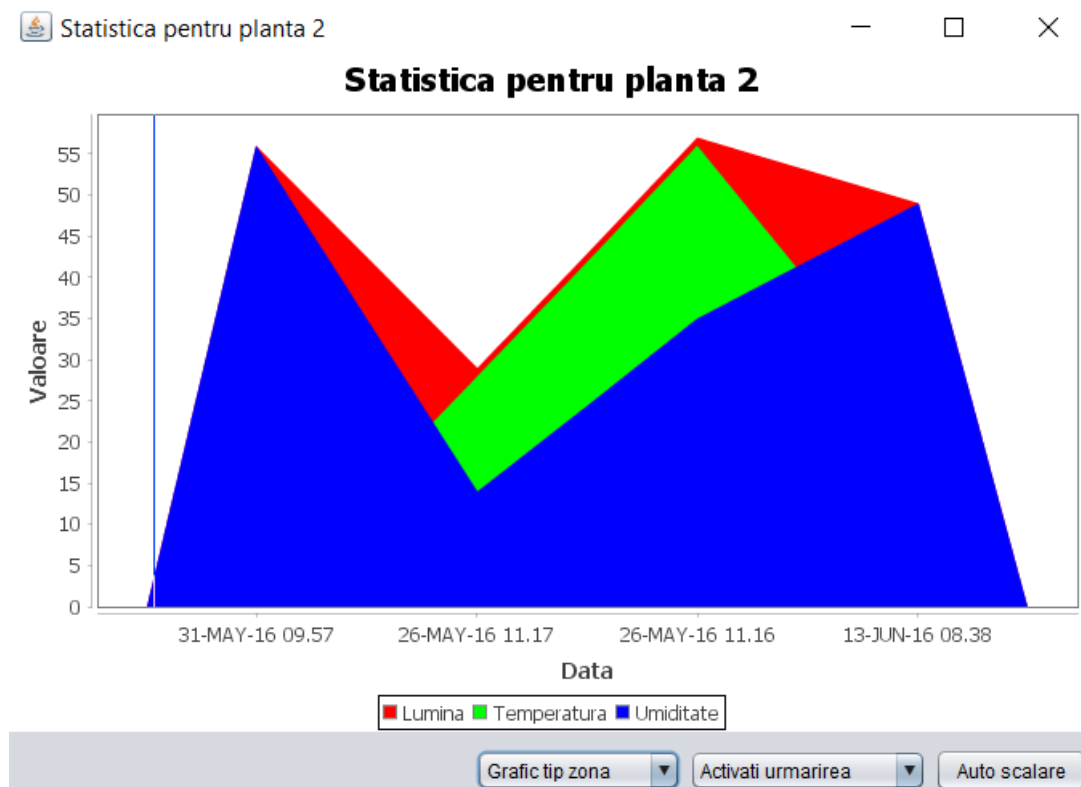


Figura 40. Grafic tip zonă

În figura 36, 37, 38, 39, 40 se poate observa fereastra generată de această clasă. Pentru creerea acesteia am folosit următoarele:

- JButton – pentru crearea butonului de autoscalare;
- JComboBox – pentru crearea unei liste de acțiuni dintre care se poate alege. Această componentă a fost folosită pentru alegerea tipului de grafic și pentru activarea/dezactivarea urmăririi;
- JPanel – pentru a pune separat butoanele de grafic;
- ChartPanel – componentă a librăriei JFreeChart cu ajutorul căreia se creează graficul;

Tot în figurile menționate mai sus se observă că se poate schimba tipul graficului generat, păstrându-se aceleași date ce au fost preluate din baza de date. În figura 36 se află un grafic de tip bară 3D, în figura 37 unul de tip bară, în figura 38 unul de tip linie, în figura 39 unul tip linie 3D, iar în figura 40 un grafic tip zonă.

5.4.2.4 Clasa Login

Clasa Login se ocupă de afișarea unei interfețe prin care utilizatorul se poate loga și va putea avea acces la funcționalitățile aplicației. Această clasă are ca membrii o clasă de tipul PanamHitek_Arduino disponibilă din librăria pe care am menționat-o într-un capitol anterior, o alta de tipul Application prin intermediul căreia se va putea avea acces la funcționalitățile aplicației și o alta Login_listener ce se va ocupa de tratarea evenimentelor apărute în această clasă. Restul membrilor clasei Login sunt butoane, zone de text și label-uri.

Clasa aceasta are ca funcții un constructor ce se ocupă de inițializări și configurări, o funcție initComponents() ce se ocupă de setarea pozițiilor pentru toate componentele ce fac parte din această fereastră și o funcție main, deci aceasta este clasa din care se va porni întreaga aplicație.

Când se va rula aplicația, se va apela funcția main care va face vizibilă această fereastră.

În figura 41 se află fereastra generată de această clasă. Pentru creerea acesteia am folosit următoarele:

- JLabel – pentru numele ferestrei, precum și pentru numele căsuțelor de text;

- JPasswordField – pentru parola introdusă de utilizator. A fost folosită această componentă, pentru a nu se vedea parola;
- JButton – pentru a crea butonul “Logare” și butonul “Înregistrare”, prin apăsarea cărora va apărea un eveniment ce va fi tratat de clasa Login_listener;
- JComboBox – pentru a putea selecta portul prin care este conectată plăcuța la calculator;

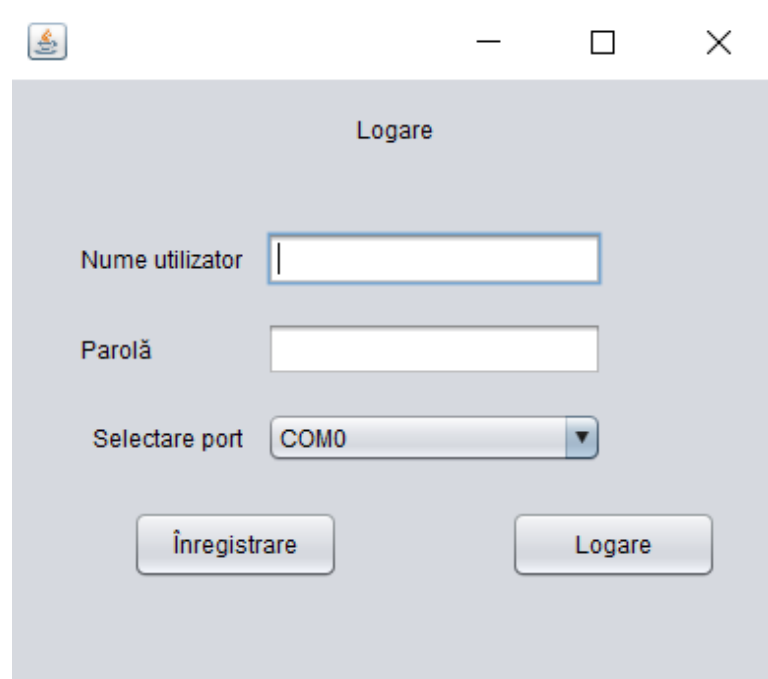


Figura 41. Fereastra de Logare

Prin intermediul constructorului se mai realizează și găsirea automată a portului serial disponibil, astfel se evită apariția erorilor de conectare datorită selectării greșite a portului COM. Acest lucru se realizează cu ajutorul următoarei linii de cod:

List_of_ports.setModel(new DefaultComboBoxModel(arduino.getSerialPorts().toArray())).

5.4.2.5 Clasa MyPainter

Clasa MyPainter se ocupă de culori și este folosită pentru schimbarea culorii de la componentele de tipul progress bar. Această clasă are ca membrii o clasă de tipul Color ce reprezintă culoarea și extinde clasa AbstractRegionPainter.

Clasa aceasta are ca funcții un constructor ce se ocupă de inițializări și configurări, o funcție doPaint() ce este suprascrisă.

5.4.2.6 Clasa Register

Clasa Register se ocupă de afișarea unei interfețe prin care utilizatorul se poate înregistra în baza de date și ulterior va putea avea acces la aplicație prin intermediul noului cont creat. Această clasă are ca membrii o clasă de tipul Login ce va putea prelua din această clasă datele utilizatorului și o alta Register_listener ce se va ocupa de tratarea evenimentelor apărute în această clasă. Restul membrilor clasei Register sunt butoane, zone de text și label-uri.

Clasa aceasta are ca funcții un constructor ce se ocupă de inițializări și configurări, o funcție initComponents() ce se ocupă de setarea pozițiilor pentru toate componentele ce fac parte din această fereastră și o funcție register_start cu ajutorul căreia fereastra generată de această clasa va fi făcută vizibilă.

Figura 42. Fereastra de înregistrare cont

În figura 42 se află fereastra generată de această clasă. Pentru crearea acesteia am folosit următoarele:

- JLabel – pentru numele ferestrei, precum și pentru numele căsuțelor de text;

- JPasswordField – pentru parola introdusă de utilizator. A fost folosită această componentă, pentru a nu se vedea parola;
- JButton – pentru a crea butonul “Înregistrare”, prin apăsarea căruia va apărea un eveniment ce va fi tratat de clasa Register_listener;

5.4.3 Pachetul Control

Pachetul Control conține mai exacte acele clase care se ocupă de tratarea de evenimente și conține următoarele clase:

- Application_listener
- Change_password_listener
- Chart_listener
- Login_listener
- Register_listener
- SerialPort_listener

Aceste clase vor fi detaliate în subcapitolele următoare.

5.4.3.1 Application_listener

Clasa Application_listener este clasa ce se ocupă de tratarea evenimentelor din clasa Application. Această clasa conține ca și membrii o clasă Application și o alta MYSQLCMD.

Clasa Application_listener implementează clasa MouseListener și clasa ActionListener, ceea ce înseamnă că poate trata și evenimentele primite de la mouse.

Am suprascris următoarele funcții:

- mouseClicked(MouseEvent evt);
- actionPerformed(ActionEvent e);

5.4.3.1.1 Funcția mouseClicked

Funcția mouseClicked tratează mai multe evenimente și anume:

- Butonul pentru modul automat. Atunci când acest buton este apăsat se dezactivează toate butoanele din panel-ul “Facilități de control” pentru a nu mai putea fi apăstate și se trimite serial către Arduino codul pentru modul automat;
- Butonul pentru modul manual. Atunci când acest buton este apăsat se vor activa toate butoanele din panel-ul “Facilități de control” și se va trimite serial către Arduino codul pentru modul manual;
- Butonul de pornit pentru led-ul 1. Prin apăsarea acestuia se va schimba culoarea butonului și se va trimite un cod serial către Arduino, urmând ca led-ul pentru prima plantă să se aprindă în urma interpretării mesajului primit;
- Butonul de pornit pentru led-ul 2;
- Butonul de oprit pentru led-ul 1. Prin apăsarea acestuia se va schimba culoarea butonului și se va trimite serial un cod către Arduino, care va ști ca led-ul va trebui stins;
- Butonul de oprit pentru led-ul 2;
- Butonul de pornit pentru pompa 1. Prin apăsarea acestuia se va schimba culoarea butonului și se va trimite un cod serial către Arduino care va cupla releul corespunzător aprinderii pompei de apă;
- Butonul de pornit pentru pompa 2;
- Butonul de oprit pentru pompa 1. După ce acest buton este apăsat se va schimba culoarea acestuia și se va decupla releul corespunzător prin trimiterea unui cod serial;
- Butonul de oprit pentru pompa 2;
- Butonul de pornit pentru aeroterma rece 1. În urma apăsării acestui buton se va trimite o comandă de cuplare a releului ce va permite aprinderea unui led albastru și a ventilatorului corespunzător;
- Butonul de pornit pentru aeroterma rece 2;
- Butonul de oprit pentru aeroterma rece 1. După apăsarea butonului se trimite codul de decuplare a releului și astfel ventilatorul și led-ul albastru corespunzător vor fi oprite;

- Butonul de oprit pentru aeroterma rece 2;
- Butonul de pornit pentru aeroterma caldă 1. În urma apăsării butonului se va trimite codul de cuplare a releului corespunzător și astfel se va aprinde un led roșu și ventilatorul corespunzător;
- Butonul de pornit pentru aeroterma caldă 2;
- Butonul de oprit pentru aeroterma rece 1. După apăsarea butonului, aplicația va trimite un cod serial ce va decupla releul corespunzător și astfel led-ul și ventilatorul vor fi oprite;
- Butonul de oprit pentru aeroterma rece 2;

5.4.3.1.2 Funcția actionPerformed

Această funcție se ocupă de tratarea evenimentelor primite de la următoarele componente:

- Butonul de “Ieșire” aflat în Meniu. Prin apăsarea acestuia, aplicația Java va executa o instrucțiune de închidere;
- Butonul “Statistică pentru planta 1” aflat în Meniu. Prin apăsarea acestuia se va deschide o fereastră ce conține un grafic generat pe baza valorilor salvate de la senzorii corespunzători primei plante;
- Butonul “Statistică pentru planta 2” aflat în Meniu;
- Butonul “Istoric” aflat tot în Meniu. După apăsarea acestuia se va deschide o fereastră în care se va regăsi istoricul aplicației;
- Butonul “Ieșire cont” din meniul Cont. În urma apăsării se va executa un cod de inactivare a ferestrei principale și de activare a ferestrei de logare;
- Butonul “Schimbă parola”. După apăsare se va deschide o fereastră în care se va putea schimba parola;
- Butonul “Sterge cont”. Prin apăsarea acestui buton se va executa un cod ce va deschide mai întâi o fereastră prin care utilizatorul se va putea răzgândi (vezi figura 43), în caz contrar se va șterge din baza de date contul (vezi figura 44) și se va activa fereastra de logare;

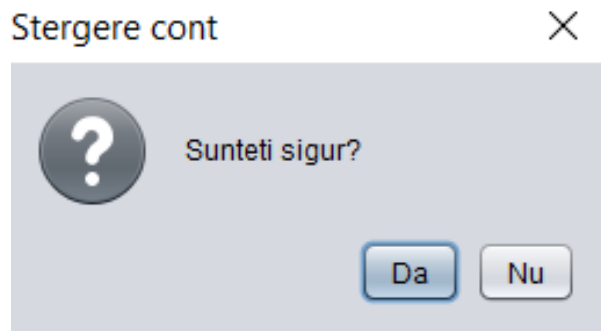


Figura 43. Ștergere cont

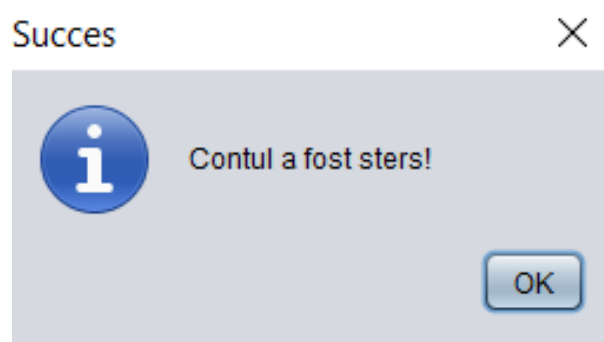


Figura 44. Cont șters

5.4.3.2 Change_password_listener

Clasa `Change_password_listener` este clasa ce se ocupă de tratarea evenimentelor din clasa `Change_password`. Această clasă conține ca și membrii o clasă `Change_password` și o altă `MYSQLCMD`.

Clasa `Change_password_listener` implementează clasa `MouseListener` ceea ce înseamnă că se vor trata evenimentele primite de la mouse.

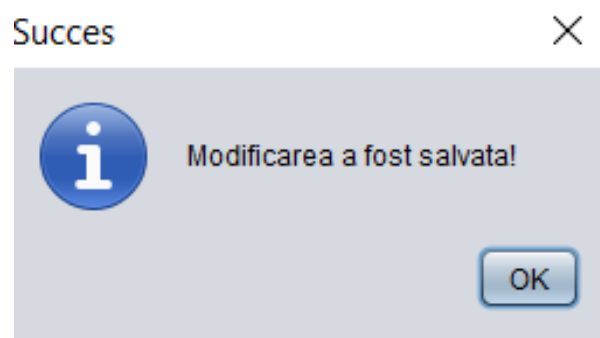


Figura 45. Parolă salvată

Am suprascris numai funcția `mouseClicked(MouseEvent evt)`. Această clasă poate trata numai evenimentul primit de la butonul de “Salvează”. Atunci când acest buton este apăsat se va executa o interogare ce va salva noua parolă în baza de date și se va dezactiva fereastra de schimbare a parolei. Dacă parola a fost salvată cu succes va apărea un mesaj ca în figura 45.

5.4.3.3 Chart_listener

Clasa `Chart_listener` este clasa care se ocupă de tratarea evenimentelor primite de la clasa `Chart`. Această clasă implementează o altă clasă numită `ActionListener`. De aceea am suprascris funcția `actionPerformed(ActionEvent e)`.

Această clasă se ocupă de tratarea evenimentelor primite de la următoarele componente:

- `ComboBox`-ul ce conține activarea/dezactivarea urmăririi;
- `ComboBox`-ul ce selectează tipul de chart. Aici se realizează schimbarea tipului de chart în funcție de ce este selectat în `ComboBox`;
- Butonul de autoscalare. Atunci când acest buton este apăsat se va executa o funcție de autoscalare disponibilă în librăria `JFreeChart`;

5.4.3.4 Login_listener

Clasa `Login_listener` este clasa ce se ocupă de tratarea evenimentelor din clasa `Login`. Această clasă conține ca și membrii o clasă `Login` și o altă `MYSQLCMD`.



Figura 46. Parole diferite

Clasa Login_listener implementează clasa MouseListener ceea ce înseamnă că se vor trata evenimentele primite de la mouse.

Am suprascris numai funcția mouseClicked(MouseEvent evt). Această clasă poate trata numai evenimentul primit de la butonul de “Logare” precum și de la butonul “Înregistrare”. Atunci când butonul de logare este apăsat se va verifica corectitudinea datelor introduse, iar în cazul în care ceva nu este bine va apărea un mesaj corespunzător (vezi figura 46) și se va deschide fereastra principală a aplicației. În schimb dacă se va apăsa butonul de înregistrare se va deschide fereastra de înregistrare cont.

5.4.3.5 Register_listener

Clasa Register_listener este clasa ce se ocupă de tratarea evenimentelor din clasa Register. Această clasă conține ca și membrii o clasă Register și o alta MYSQLCMD.

Clasa Register_listener implementează clasa MouseListener și se vor trata evenimentele primite de la mouse.

Am suprascris numai funcția mouseClicked(MouseEvent evt). Această clasă poate trata numai evenimentul primit de la butonul de “Înregistrare”. Atunci când butonul de înregistrare este apăsat se va verifica corectitudinea datelor introduse (dacă parolele coincid), iar în cazul în care ceva nu a decurs așa cum ar trebui va apărea un mesaj corespunzător (vezi figura 46). În cazul în care totul a decurs bine și datele au fost corecte va apărea un mesaj de informare conform căruia contul a fost creat, la fel ca în figura 47.

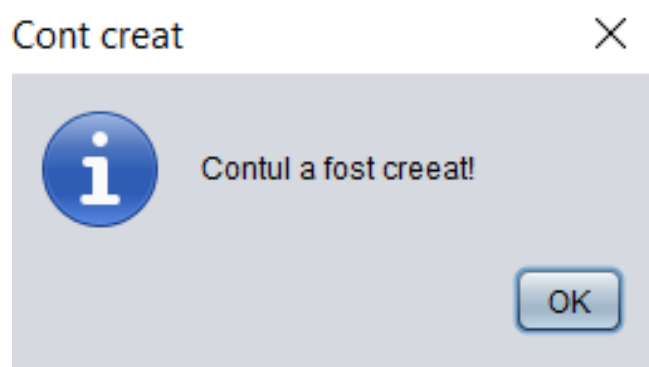


Figura 47. Cont creat

5.4.3.6 SerialPort_listener

Clasa SerialPort_listener implementează o clasă numită SerialPortEventListener și se ocupă cu tratarea evenimentelor apărute pe port, adică cu ce s-a primit serial. Această clasă conține un membru de tipul clasei Application.

Dacă s-a primit ceva pe serială atunci această clasă se va ocupa de mesajul recepționat. Plăcuța Arduino trimite serial pe o singură linie toate informațiile recepționate de la senzori. Aici se va interpreta acest mesaj luând valoarea corespunzătoare primei codări găsite. Pentru a lua întreaga valoare ce se află între două codări de senzori se vor folosi două obiecte unul de tipul clasei Pattern și un altul de tipul clasei Matcher.

Un mesaj primit serial arată astfel:

L1453L2459H11023H21023T122.52T222.63

Codările utilizate sunt următoarele:

- L1 – corespunzător primului senzor de lumină;
- L2 – pentru cel de al doilea senzor de lumină;
- H1 – corespunzător primului senzor de umiditate;
- H2 – pentru informația primită de la cel de al doilea senzor de umiditate;
- T1 – pentru primul senzor de temperatură;
- T2 – corespunzător celui de al doilea senzor de temperatură;

Pe lângă primirea de informații această clasă se ocupă și de actualizarea valorilor primite în interfața principală, de aceea este nevoie de un membru de tipul clasei Application. Elementele de tip progress bar vor avea și culori diferite în funcție de aceste valori.

Această clasă se mai ocupă de asemenea de actualizarea imaginilor plantelor din interfața principală tot pe baza acestor informațiilor



Figura 48. Plantă ofilită



Figura 49. Plantă uscată



Figura 50. Plantă normală

primit serial. Pentru asta am folosit trei poze diferite care pot fi observate în figurile 48, 49, 50 . Cele trei imagini se vor schimba în timp real.

6 CONCLUZII

Proiectul de licență a fost ales pentru a facilita întreținerea unei sere, astfel prin acest sistem care se ocupă de controlarea parametrilor, se va evita prezența umană.

Pornind de la obiectivele proiectului, am început dezvoltarea unei platforme capabile să îndeplinească toate aceste cerințe. În plus scopul proiectului este de a arăta că pentru un astfel de proiect se pot utiliza cunoștințe din mai multe domenii (ca de exemplu: bazele electrotehnicii, programarea microcontroller-elor, programarea orientată pe obiect, inginerie software, medii de programare vizuală, electronică, etc.) precum și mai multe alte tehnologii.

Prima etapă în dezvoltarea proiectului a fost selectarea componentelor și realizarea unei machete provizorii ce am utilizat-o pentru testarea funcționalităților. Astfel în acest pas am conectat microcontroller-ul cu senzorii, cu LED-urile precum și cu pompele de apă, urmând ulterior să atașez la această machetă și senzorii de temperatură, precum și ventilatoarele.

Următoarea etapă a constat în scrierea unui cod capabil să recepționeze informațiile primite de la senzori și să le scrie pe serială, de unde vor putea fi folosite mai departe de aplicația Java. Tot aici a trebuit să se calibreze valorile senzorilor, deoarece senzorii au fost conectați la porturi analogice. În funcție de valorile pe care le-au dat senzorii în diferite circumstanțe sau mai bine zis în diferite condiții la care au fost supuși, a fost posibilă calibrarea lor. Această parte a fost realizată în Arduino IDE, cu alte cuvinte, această parte de funcționalitate se regăsește pe microcontroller-ul ATmega2560.

Ulterior am reușit integrarea senzorilor de temperatură și a ventilatoarelor atât în aplicația Arduino cât și în aplicația JAVA. A urmat după asta o etapă de verificare a eventuale probleme și rezolvarea acestora.

Odată finalizată întreaga aplicație nu a mai rămas decât de făcut macheta finală a proiectului, documentația și prezentarea finală.

În acest proiect am investit mult suflet și am vrut să iasă ceva cât mai bun de pe urma tuturor cunoștințelor dobândite în urma facultății. De asemenea am vrut să combin în acest proiect tot ce îmi place mie mai mult. Astfel am revenit la o veche pasiune de a mea, desenatul realizând imaginile cu cele trei plante. Și de ce nu pot să zic că și tema proiectului este o veche pasiune de a mea adică grădinăritul și cum majoritatea nu avem timp pentru asemenea lucruri, dar totuși am vrea o grădină cât mai frumoasă sau am vrea să mâncăm cât mai sănătos, acest proiect s-a potrivit ca o mânășă.

Aș vrea să continui acest proiect prin extinderea lui pe mai multe plante, eventual să pot controla parametrii serei în funcție de tipurile de plante și de ce nu să le și pot supraveghea online pe bază unor

camere de filmat. Aș putea extinde aplicația astfel încât să nu mai fie decât desktop, eventual o aplicație web sau o aplicație pe mobil.

7 BIBLIOGRAFIE

[DUM06] - *Bazele sistemelor mecatronice Curs*, Adrian Dumitriu, Braşov, 2006

[DIN08] - *Echipamente şi sisteme hidropneumatice de bord*, Liviu Dincă, Editura Universitaria, Craiova, 2008

[STE04] - *Fizica Atmosferei, vremea si clima*, Sabina Stefan, Editura Universităţii din Bucureşti, 2004

[MAR10] - *Ghid practic de programare în C*, MARIAN Gheorghe, MARIAN Marius - Adrian, ENESCU Nicolae, DUMITRAŞCU Eugen, Editura UNIVERSITARIA, Craiova, 2010

[JDBC14] - *Oracle Database JDBC Developer's Guide 12c Release 1 (12.1)*, 2014

[GIL14] - *The JFreeChart Class Library Version 1.0.19 Installation Guide*, David Gilbert, 2014

8 REFERINȚE WEB

[LAU12] - Laurean Bogdan, *Microcontrolere Introducere*, disponibil via web la adresa <http://web.ulbsibiu.ro/laurean.bogdan/html/Microcontrolere%20introducere.pdf>

[CIO03] - Andrei Ciorba, *Introducere în microcontrollere*, disponibil via web la adresa <http://andrei.clubcisco.ro/cursuri/3pm/lab1.pdf>

[ATM14] - Atmel Atmega640/V-1280/V-1281/V-2560/V-2561/V, disponibil via web la adresa http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

[ARD16] - *Arduino*, disponibil via web la adresa <https://ro.wikipedia.org/wiki/Arduino>

[VOR09]- Liliana Vornicu, *Senzori și traductoare*, disponibil via web la adresa http://ep.etc.tuiasi.ro/index_sen.html

[REL11] - *Relee și declanșatoare*, disponibil via web la adresa <http://elth.ucv.ro/student1/Cursuri/Echipamente%20electrice%20I/Laborator/Studiul%20releelor%20si%20declansatoarelor.pdf>

[NAG05] - Ștefan Nagy, *Ventilatoare și instalații de ventilare*, disponibil via web la adresa <http://test.mrxl.ro/joomla/images/Cursuri/uem/Cap5.pdf>

[NET16] - *NetBeans*, disponibil via web la adresa <https://en.wikipedia.org/wiki/NetBeans>

[ORA14] - *Oracle Database 11g Express Edition*, disponibil via web la adresa <http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/index.html>

[VIS16] - *Microsoft Visual Studio*, disponibil via web la adresa https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

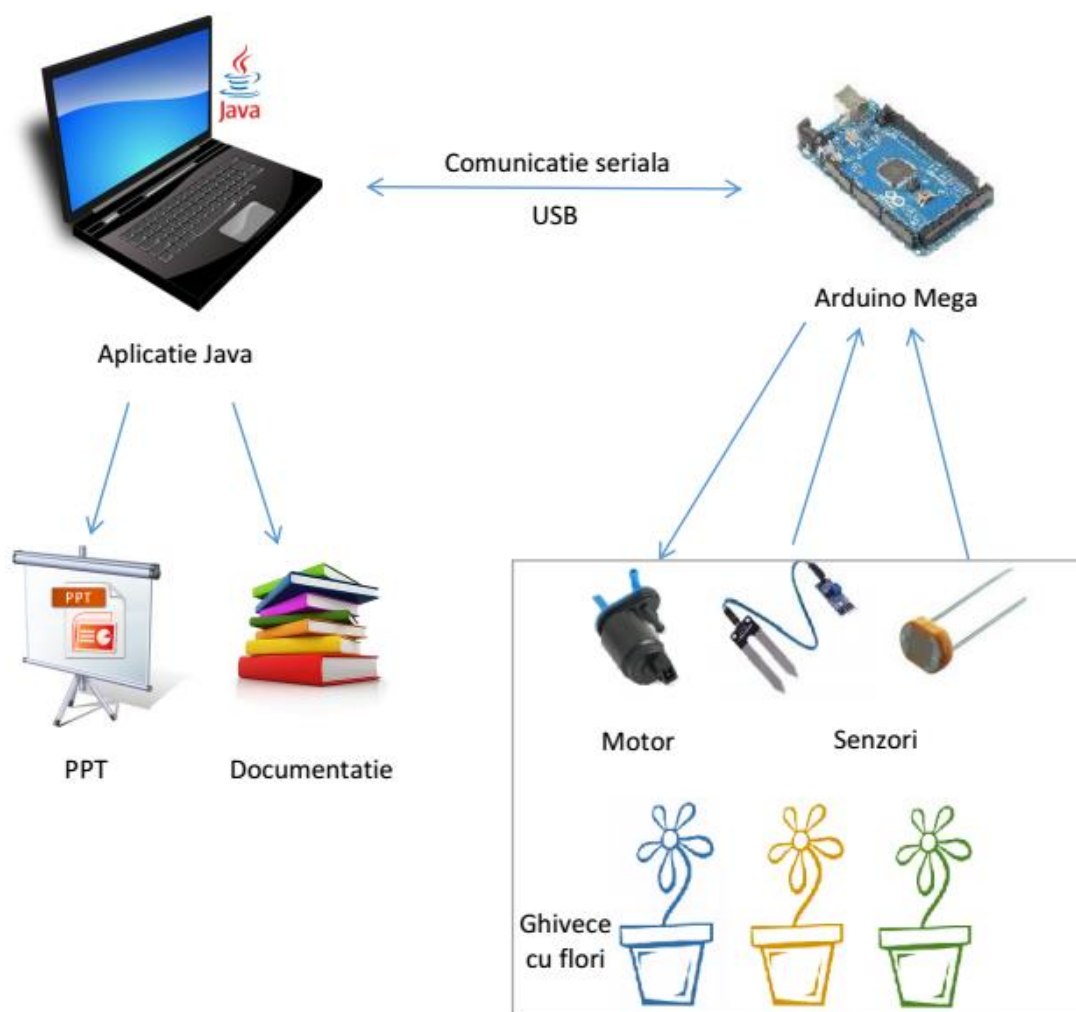
[FRI16] - *Fritzing*, disponibil via web la adresa <https://en.wikipedia.org/wiki/Fritzing>

[MEG14] - *Arduino Mega 2560*, disponibil via web la adresa http://www.geeetech.com/wiki/index.php/Arduino_Mega_2560

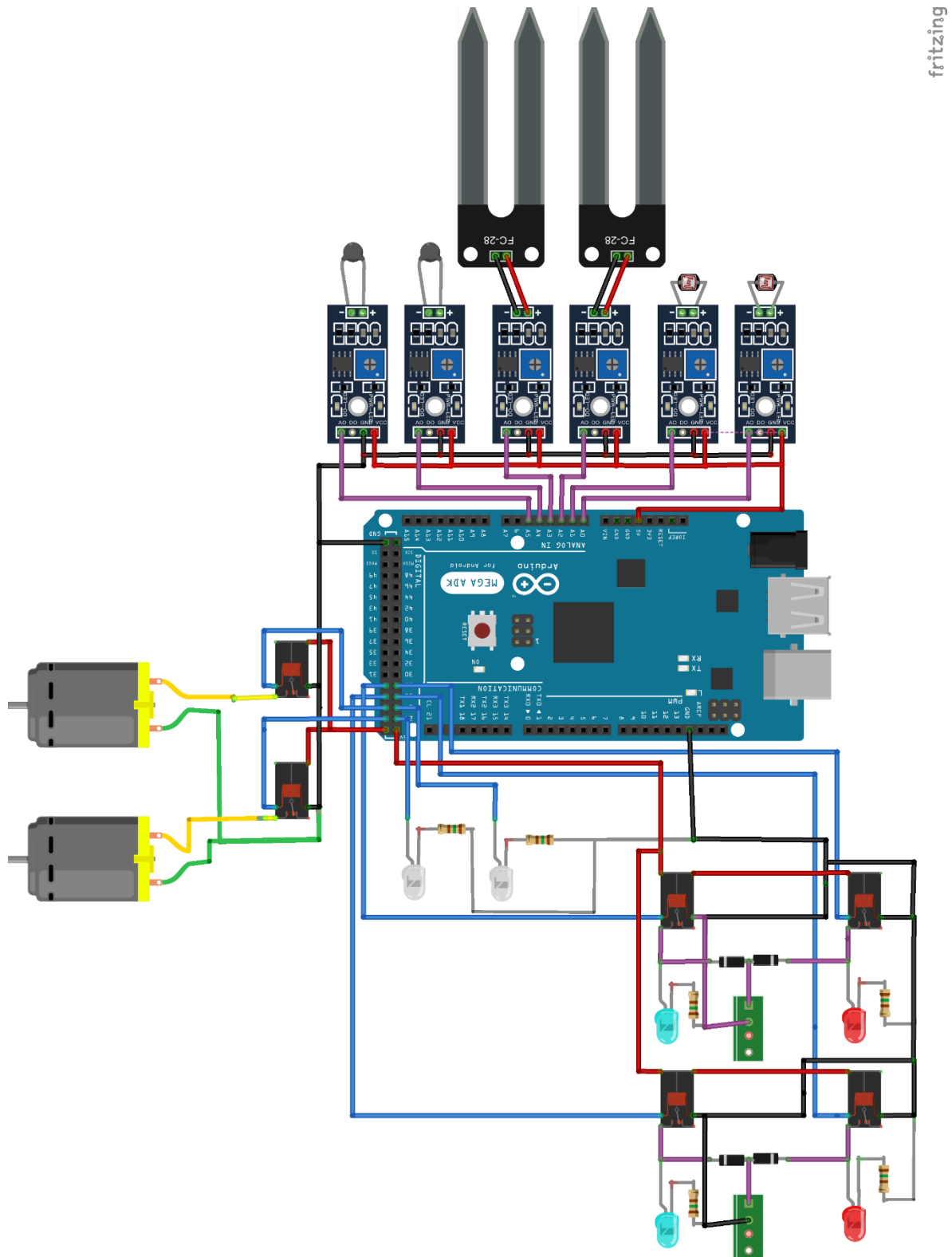
[GON14] - *Panama Hitek*, disponibil via web la adresa <http://panamahitek.com/>

[REG16] - *Regular expressions*, disponibil via web la adresa http://www.tutorialspoint.com/java/java_regular_expressions.htm

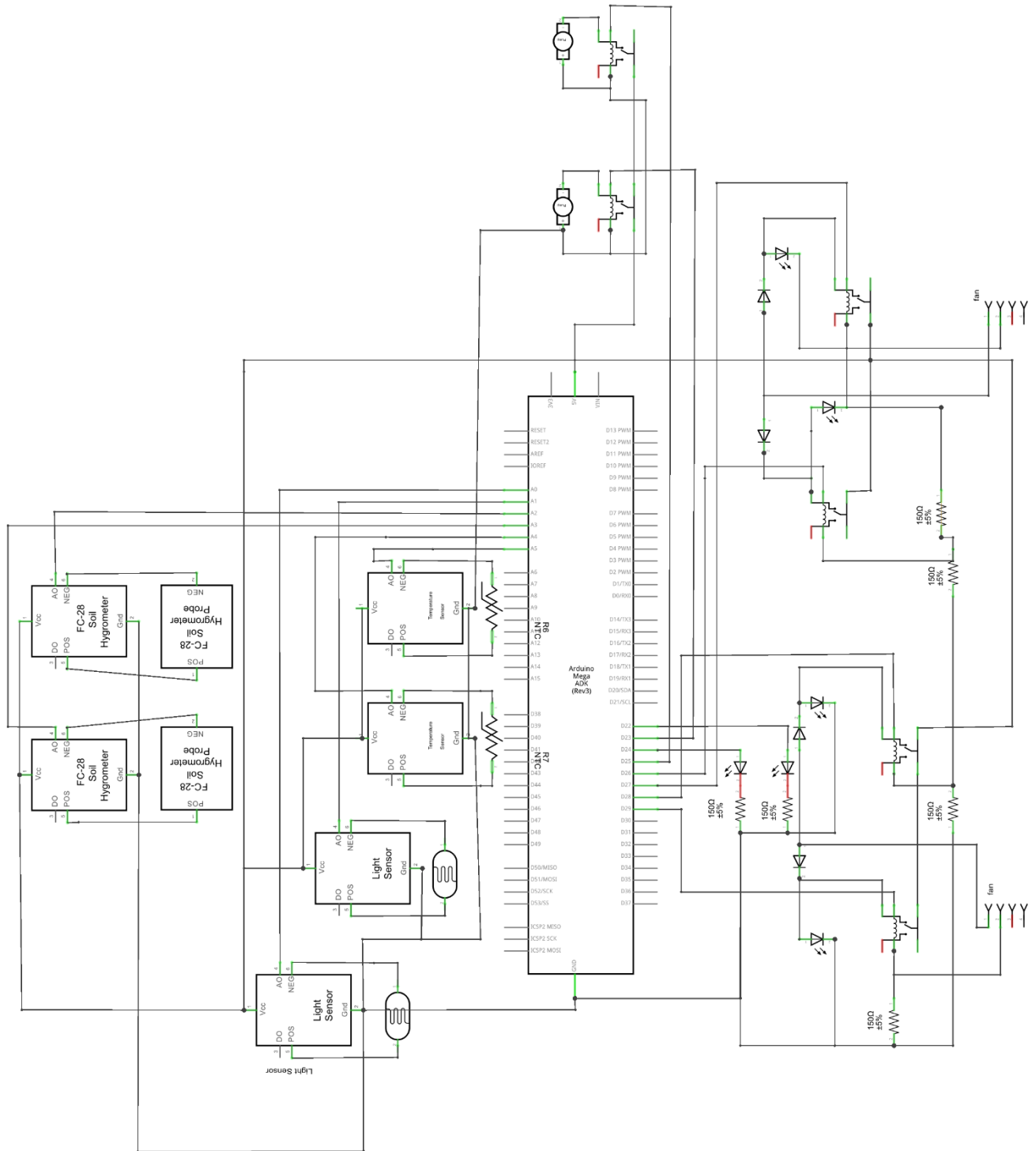
A. LIVRĂRILE PROIECTULUI



B. ARHITECTURA GENERALĂ A SEREI



C. SCHEMA ELECTRONICĂ A SEREI



C. CD / DVD

În această anexă, este atașată versiunea electronică a aplicației, a acestei lucrări, precum și prezentarea finală a tezei.



INDEX

A

Arduino IDE	16
ATmega2560	6

B

Bibliografie	57
--------------	----

C

CUPRINSUL	xii
-----------	-----

F

Fritzing IDE	19
--------------	----

L

LED-uri	12
Librăria JDBC	29
Librăria JFreeChart	28
Librăria PanamaHitek	29
LISTA FIGURILOR	xv
LISTA TABELELOR	xvii

N

NetBeans IDE	17
--------------	----

O

Oracle Database	18
-----------------	----

P

Pompe hidraulice	14
------------------	----

R

Referințe web	58
Relee	14

S

Senzori	10
---------	----

V

Ventilatoare	15
Visual Studio	18