Universitatea Tehnică "Gheorghe Asachi" din Iași

Facultatea de Electronică , Telecomunicații și

Tehnologia Informației

Specializarea "Electronică aplicată"

Lucrare de Diplomă

Testarea inteligentă a unui bloc de senzori

Profesor Coordonator:

Asist.Univ.Drd.Ing. Pletea Ionica Marcela

Student:

Cazacu Andrei Tudor

Iași

Iulie 2023

Universitatea Tehnică "Gheorghe Asachi" din Iași

Facultatea de Electronică , Telecomunicații și

Tehnologia Informației

Specializarea "Electronică aplicată"

Lucrare de Diplomă

Profesor Coordonator:

Asist.Univ.Drd.Ing. Pletea Ionica Marcela

Student:

Cazacu Andrei Tudor

Iași

Iulie 2023

Universitatea “ Gheorghe Asachi ” din Iași

Facultatea de Electronică , Telecomunicații și Tehnologia Informației

Specializarea “ Electronică Aplicată”

Testarea inteligentă a unui bloc de senzori

Profesor Coordonator:

Asist.Univ.Drd.Ing. Pletea Ionica Marcela

Student:

Cazacu Andrei Tudor

Iași

Iulie 2023

Cuprins:

Capitolul 1

1. Introducere

Testarea inteligentă a unui bloc de senzori este o lucrare ce se bazează pe dezvoltarea unei metode avantajoase de testare a unor senzori relevanți folosind placa Arduino Uno , scopul fiind de a oferi o soluție cât mai eficientă și inteligentă pentru a verifica funcționarea corectă a diferitelor tipuri de senzori pe care i-am ales.

Prin utilizarea plăcuței Arduino Uno pentru a conecta toți pinii senzorilor și a programului Arduino IDE în care voi scrie codurile aferente acestor senzori , voi încerca să creez un sistem inteligent și cât mai automatizat.Prin compilarea , pe rând , a unui număr de nouă senzori , o să afișez pe un ecran LCD valorile de ieșire , iar acestea vor fi:

- Temperatura și umiditatea din laborator;
- Nivelul de gaz din laborator;
- Măsurarea distanței față de un obiect;
- Atingerea tactilă a unor butoane;
- Detectarea mișcării;
- Detectarea obstacolelor;
- Valorea apei;

Lucrarea se bazează pe un studiu amplu al senzorilor selectați și prezentarea tuturor componentelor ale acestora într-un mod cât mai expresiv , pentru a avea ca rezultat o testare rapidă și un proces de verificare a funcționalității cât mai precis.

1.1 Motivul alegerii temei

Motivul pentru care am ales tema “Testarea inteligentă a unui bloc de senzori” a plecat de la ideea că pot explora în profunzime domeniul senzorilor și conexiunea lor la placa Arduino Uno, astfel încât să dezvolt abilitați practice în testarea și evoluarea performanței acestora.

Astfel , lucrarea mea prezintă următoarele avantaje:

- Importanța testării senzorilor: aceștia sunt componentele esențiale care fac parte din multe aplicații și sisteme deoarece furnizează date pentru a lua diferite decizii, a controla diferite obiecte , iar testarea lor cât mai eficientă contribuie la evitarea erorilor ce pot apărea;

- Arduino Uno: este una dintre cele mai populare plăci produse de firma cu același nume, fiind foarte prietenoasă cu senzorii , astfel că putem obține un mediu cât mai flexibil și ușor de utilizat prin implementarea unei logici de testare inteligentă;
- Contribuția adusă domeniului electronicii: poate deschide noi direcții studenților care sunt pasionați atât de partea hardware , cât și de partea software și vor să dezvolte soluții de testare cât mai inteligente;

1.2 Principiul de funcționare al unui sistemului

Principiul de functionare al acestui sistem implica mai multe etape și componente.

a. Senzorizare: acest sistem utilizează senzori relevanți pentru a măsura diferiți parametri din laborator , cum ar fi temperatura , umiditatea , mișcarea , gazul , curentul, detectarea tactilă , nivelul apei , senzori conectați la placa Arduino Uno;
b. Interacțiunea cu utilizatorul: Sistemul este echipat cu o interfață de utilizator , iar eu voi folosi un ecran LCD pe care voi permite vizualizarea datelor afișate de către toți cei nouă senzori testați;
c. Colectarea datelor: senzorii aprovizonează date digitale sau analogice , care sunt convertite în semnale digitale folosind convertoare ADC , iar aceste date sunt preluate de către ecranul LCD;
d. Procesarea și luare deciziilor: datele colectate sunt procesate de către microcontroler , în cazul meu placa Arduino Uno , iar acesta poate lua decizii bazate pe valorile introduse și pe setările prestabilite;

1.3 Obiectivele lucrării

Lucrarea mea are că scop dezvoltarea unui sistem de testare cât mai rapid și eficient prin intermediul Arduino și senzorilor relevanți , pentru a monitoriza și controla anumiți parametrii cheie , având ca rezultat coletarea , interpretarea și procesarea datele provenite de la aceștia.

Pentru asta , lucrarea va trebui sa conțină următoarele componente atât hardware , cât și software de care mă voi folosi:

a. Componente hardware:
   - Placa de dezvoltare Arduino Uno;
   - Cablu USB A-B;
   - Breadboard;
   - Senzor de gaz metan;

- Senzor de temperatura si umiditate;
- Senzor ultrasonic;
- Senzor de curent;
- Releu tensiune 5V;
- Senzor de mișcare;
- Senzor infra-rosu IR;
- Senzor capacitiv;
- Senzor picături de ploaie;
- Dispozitiv de afișare (ecran LCD + adaptop I2C);
- Elementele de control: rezistențe 220 Ohmi, fire mamă-tată , led-uri 5mm , buzzer;

b. Componente software:

- Limbaj de programare Arduino;
- Platforma de dezvoltare Arduino IDE;
- Biblioteci Arduino;

1.4 Diagrama bloc

Figura 1.1 Diagrama bloc

În diagrama bloc , plăcuta Arduino Uno ocupa poziția centrală și este conectată la o varietate de senzori , elemente de control și un ecran LCD.Aceasta va acționa ca un microcontroler și vă gestiona interacțiunea și comunicarea cu elementele ce sunt conectate la ea.

Senzorii reprezintă elementele cheie ale sistemului meu.Acești senzori vor foruniza informații și date pentru a fi prelucrate și utilizate în funcționarea sistemului.

Ecranul LCD vă fi utilizat pentru afișarea datelor și a rezultatelor testelor senzorilor. Acesta poate oferi o interfață cu utilizatorul prietenoasă, unde acesta poate vedea și interpreta rezultatele testelor senzorilor.

Elementele de control, precum rezistențe, led-urile , buzzer-urile sunt utilizate pentru a controla diferite aspecte ale sistemului și pentru a ajusta setările și parametrii de testare a senzorilor.

1.5 Organigramă de lucru

Organigrama de lucru a fost realizată în scopul testării senzorilor , oferind un flux și o structură clară de activitați necesare pentru a realiza testarea într-un mod eficient și sistematic.

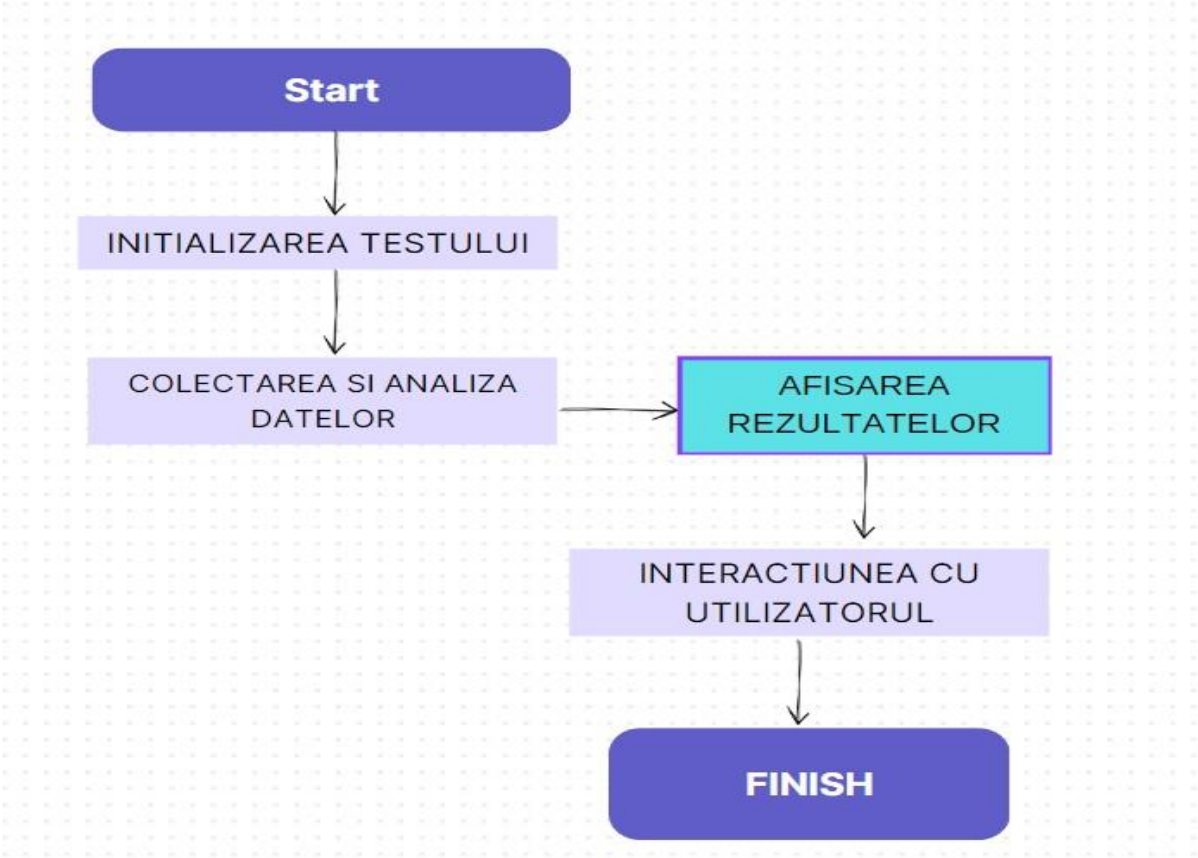

Figura 1.2 Organigramă de lucru

Inițierea testului este prima etapa în fuctionarea sistemului de testare, iar în aceasta fază utilizatorul inițializează un test specific pentru un anumit senzor și setează parametrii de testare.

Colectarea și analiza datelor este a doua etapă , în care placa Arduino primește semnale de la senzorul ce urmează să fie testat , semnalele analogice fiind transformate în semnale digitale prin intermediul conevertorului analog-digital (ADC) de pe placa Arduino Uno.Urmează a fi evaluate datele primite și comparate cu parametrii de testare predefiniti.

A treia etapa reprezintă afișarea rezultatelor testului.După ce datele au fost colectate și analizate , rezultatele sunt afișate utilizatorului pe ecranul LCD.

Ultima etapa este interacțiunea cu utlizatorul , în care acesta poate modifica sau seta parametrii de testare în funcție de cum își dorește.

Capitolul 2

2. Prezentarea componentelor hardware

2.1 Arduino Uno: prezentare și mod de funcționare

Arduino este o companie cunoscută în industria tehnologiei, care produce plăcute de dezvoltare bazate pe microcontrolere și partea de software destinată programării și funcționarii acesteia.Un student de la Institutul de Interacțiune a Design-ului din Italia a început demersurile proiectului Arduino în anul 2005.

Plăcile Arduino sunt compuse dintr-un microcontroler AtmelAVR de 8,16 sau 32 de biți și pe ele este instalat un bootloader numit optiboot cu rolul de a simplifica încărcarea programelor scrise, pe memoria flash.Conțin pini de intrare și ieșire (Diecimila , Duemilanove și Uno oferă 14 pini digitali de I/O , dintre care 6 pot produce semnale PWM și 6 intrări analogice care pot fi folosite că I/O digitale), buton de reset, dar și conectori standard, care permit utilizatorului să conecteze plăcuta cu procesorul la diferite module numite shield-uri.Unele shield-uri comunică cu Arduino direct cu ajutorul pinilor digitali/analogici , dar altele sunt adresate individual prin magistrala seriala I2C care permite utilizarea modulelor în paralel.Plăcutele au incorporate un port USB cu ajutorul căreia putem încarca codul scris de pe PC-ul nostru, având integrate cipuri de conversie USB-serial (FTDI FT232).Unele placi Arduino au implementate convertoare pentru conversia între nivelele logice RS-232 și TTL.

Pentru programrea unei plăci Arduino , aceasta are integrată un sistem de dezvoltare (IDE).

Exemple de plăcute Arduino:

- Arduino UNO
- Arduino Mega 2560
- Arduino Leonardo
- Arduino Yun
- Arduino LilyPad
- Arduino Nano

Arduino Mega 2560 este una dintre cele mai ulilizate placi , după Uno.Ea are următoarele caracteristici tehnice: memorie de 256 KB , tensiunea de alimetare 5V și de intrare 7-12V, frecventa de 16 MHz (conține un oscilator de cristal) , microcontroler Atmega2560 , are 54 pini (16 analogici, iar 14 pot fi utilizați că ieșiri PWM).

Arduino Yun este o placă de microcontroler bazată pe ATmega32u4 și Atheros AR9331. Procesorul Atheros acceptă o distribuție Linux bazată pe OpenWrt numită Linino OS. Placa are un port USB, slot pentru card micro-SD, 20 de pini de intrare/ieșire digitală (7 dintre ei pot fi utilizați ca ieșiri PWM și 12 ca intrări analogice), un cristal de 16 MHz oscilator, o conexiune micro USB, 3 butoane de resetare.

Arduino Nano este o placă mică, bazată pe ATmega328.Are 14 pini digitali care se pot folosi ca ieșire sau intrare și 8 intrari analogice.Memorie de 32KB.

Arduino LilyPad este conceput pentru proiecte textile și poate fi cusut pe țesătura hainelor.Are un mictrocontroler Atmega328 și doar 9 pini pentru intrare/ieșire.

Arduino Leonardo diferă de toate celelalte plăci prin faptul că Atmega32u4 are comunicație USB incorporată.Conține 20 de pini intrare/ieșire (7 pini ca ieșiri PWM și 12 ca intrări analogice) , oscilator cu cristal de 16MHz , o mufă de alimentare și un buton de resetare.

Arduino Diecimila este o placă de microcontroler bazată pe ATmega168 (fișă de date). Are 14 pini de intrare/ieșire digitale (6 pot fi utilizați ca ieșiri PWM), 6 intrări analogice, un oscilator cu cristal de 16 MHz, o conexiune USB, o mufă de alimentare, un antet ICSP și un buton de resetare.

Pentru lucrarea mea de licență , am folosit placa Arduino Uno , care este una dintre cele mai bune plăci pentru a începe studiul electronicii și pentru a începe programarea software.Aceasta este o placa de microcontroler Atmega328P și are 14 pini digitali de intrare/ieșire , dintre care 6 pot fi utilizați că ieșiri PWM , 6 intrări analogice , un oscilator ceramic de 16MHz , o conexiune USB și poate fi conenctata la calculator direct prin acest port, o mufa de alimentare , un header ICSP , un buton de resetare.

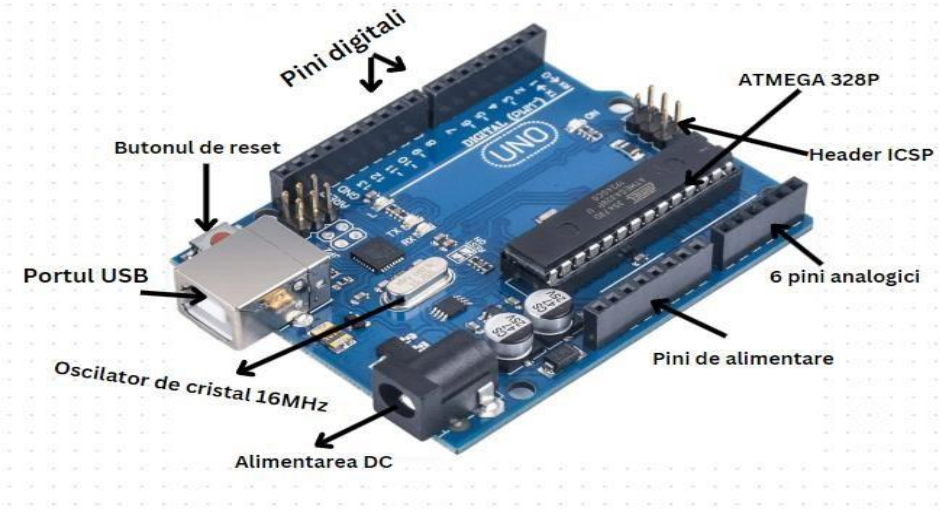

Figura 2.1 Părtile componente ale placii

Portul USB: permite conectarea și comunicarea cu calculatorul prin intermediul cablului USB și trimite date de la senzori la computer.

Alimentarea DC: placuța acceptă o gamă largă de tensiuni de alimentare DC , de la 7 până la 12V.Folosim alimentarea DC cu ajutorul unui acumulator și executăm testele dorite fara a fi nevoie de conexiunea la un calculator.

Butonul de reset: este utilizat pentru a reseta dispozitivul conectat , inclusiv placa Arduino Uno.Atunci când este apăsat , declașează o acțiune care pune în fucțiune din nou programul.

Oscilatorul de cristal 16MHz: furnizează un semnal de referintă stabil și precis pentru a sincroniza operațiile interne ale microcontrolerului Atmega 328P de pe placa Arduino , iar acest semnal este esențial pentru a asigura un cronometraj precis al execuției codului sursa.

Header ICSP: este utilizat pentru programarea și reprogramarea microcontrolerului Atmega 328P de pe placa Arduino și permite conectarea directa a unui dispozitiv de programare compatibil pentru a încărca codul sursă pe microcontroler.

Pinii digitali și analogici: pinii digitali , în numar de 14, numerotați de la D0 la D13 , pot fi utilizați pentru citirea cât si pentru scrierea valorilor digitale , pentru ieșiri sau intrări digitale și suportă diferite protocoale precum PWM.De asemnea, pinii analogici sunte numerotați de la A0 la A5, în număr de 6 și permit citirea valorilor analogice, au o rezoluție de 10 biți.

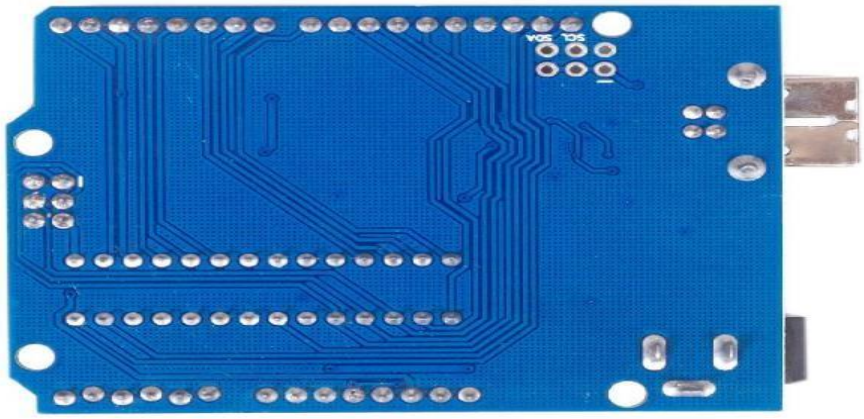

Figura 2.2 Spatele plăcii Arduino

În continuare , voi exemplifica câteva avantaje și dezavantaje ale plăcii Arduino Uno.

Avantaje: ○ Ușurința de

utilizare; ○ Costul

accesibil;

○ O putem folosi pe toate sistemele de operare precum Windows , Linux);

○ Poate fi utilizată într-o gama largă de aplicatii; ○ Programarea ei se face într-

un minim de secunde; Dezavantaje: ○ Lipsa de conexiune wireless incorcorată:

Wi-Fi sau Bluetooth; ○ Resurse limitate;

Specificațiile tehnice pentru Arduino Uno:

Microcontroler	ATmega328P – 8 bit AVR family microcontroller
Tensiunea de functionare	5V
Tensiunea recomandată de intrare	7-12V

Limitele tensiunii de intrare	6-20V
Pinii analogici	6 (A0 – A5)
Pinii digitali (intrare/iesire)	14 (dintre care 6 provide - iesire PWM)
Curent continuu pentru pinii intrare/ieșire	40 mA
Curent continu pentru pinul 3.3V	50 mA
Memoria Flash	32 KB (0.5 KB este pentru Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecvența (viteza ceasului)	16 MHz

Tabel 2.1 Caracteristicile tehnice

Mai departe , voi insera poza cu pinii plăcii Arduino și voi prezenta fiecare pin al acesteia.

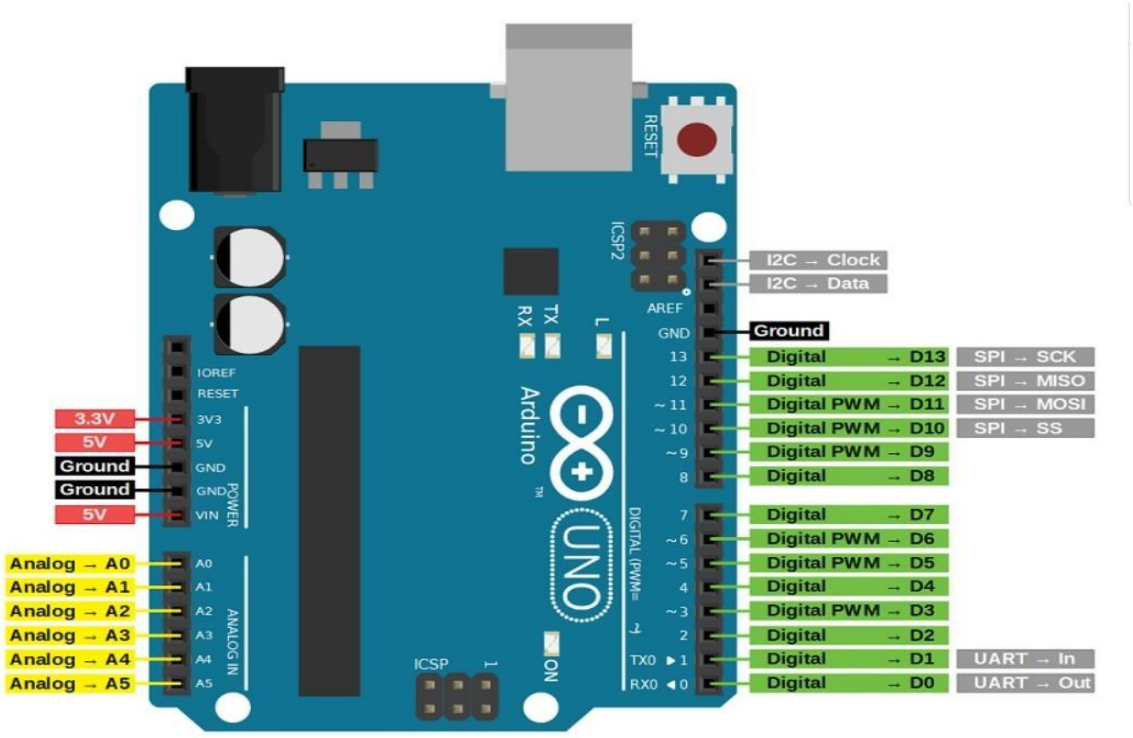

Figura 2.3 Pinii placutei Arduino Uno

Placa Arduino Uno are următorii pini prezentați în figura de mai sus: ○ Pinul D0 (RX) – utilizat pentru comunicarea serială de la dispozitive externe catre placa

Arduino; ○ Pinul D1 (TX) - utilizat pentru comunicarea seriala de la dispozitive externe catre placa

Arduino; ○ Pinul D2 (External Interrupts) – se poate declanșa o întrerupere; ○ Pinul D3 (External Interrupts + PWM) – se poate declansa o înrerupere; ○ Pinul D4 – pin de intrare/ieșire; ○ Pinul D5 – PWM; ○ Pinul D6 – PWM;

○ Pinul D7 - pin de intrare/ieșire; ○ Pinul D8 - pin de intrare/ieșire; ○ Pinul D9 - PWM; ○ Pinul D10 – PWM + SPI , se face comunicare

prin interfața serială; ○ Pinul D11 - PWM + SPI , se foloseste pentru MOSI; ○ Pinul D12 – suportă SPI , se foloseste pentru MISO; ○ Pinul D13 – suportă SPI , pinul are valoarea high – placa e alimentată; ○ Pinul GND – masa; ○ Pinul AREF – se foloseste pentru tensiunea de referință; ○ Pinul SDA – comunicarea I2C; ○ Pinul SCL – comunicarea I2C; ○ Pinul A0 – pin analogic; ○ Pinul A1 – pin analogic; ○ Pinul A2 – pin analogic; ○ Pinul A3 – pin analogic; ○ Pinul A4 – pin analogic SDA , comunică pe magistrala I2C; ○ Pinul A5 – pin analogic SCL , comunică pe magistrala I2C; ○ Pinul 1 Vin – intrarea; ○ Pinul 2 GND – masa; ○ Pinul 3 GND – masa; ○ Pinul 4: 5V – alimentare de 5V; ○ Pinul 5: 3.3V – alimentare de 3.3V; ○ Pinul 6: RESET – pentru resetare; ○ Pinul 7: IOREF – folosit de unele shield-uri; ○ Pinul 8 – neconectat;

2.2 Cablul USB A-B

Cablu care are un conector tip A la un capăt și un conector tip B la celălalt capăt.Este folositor pentru a conecta placa Arduino Uno la computer.

Mufa USB de tip A este cea mai comună și se potrivește în portul USB al calculatorului , este plată și are patru pini metalici care se potrivesc în portul USB , iar celălalt capăt de tip B se potrivește în portul USB al plăcuței Arduino , este mai pătrată și are tot 4 pini metalici.

Figura 2.4 Cablul USB tip A-B

Acesta este bine realizat astfel încât sa reziste la manipulare si conectări repetate , având următoarele caracteristici tehnice:

- Viteza de transfer: până la 480Mbps;
- Tip USB: 2.0;
- Conectori: USB tip A tată si USB tip B tată;
- Culoare: albastră;
- Lungime: 30 cm;

2.3 Ecranul LCD si modulul I2C

O noua componentă a lucrării mele este ecranul cu cristale lichide LCD.Am optat pentru un LCD 16X02 deoarece este ideal pentru proiectele de electronică ce au de afișat mai multă informație și poate fi folosit cu ușurintă și în condiții de iluminare joasă întrucât are lumină de fundal.

Un ecran cu cristalele lichide este un tip de ecran utilizat în multe dispoziție electronice (televizoare , tablete , telefoane) și funcționează pe baza proprietăților opti-electrice ale cristalelor lichide.Acestea este compus din mai multe straturi, inclusiv un strat de cristale lichide amplasat între doua placi de sticlă.

Ecranele LCD sunt de mai multe tipuri:

- Twisted Nematic (TN) LCD;
- Super Twisted Nematic (STN) LCD;
- In-Plane Switching (IPS) LCD;
- Vertical Alignament (VA) LCD;
- Advenced Fringe Field Switching (AFFS) LCD;

Figura 2.5 Ecranul LCD

Modulul LCD 1602 afișează 2 linii a câte 16 caractere, adică 32 de caractere posibile de afișat.

Caracteristicile tehnice:

Tip ecran	16 caractere pe 2 rânduri
Material	PCB + plastic
Dimensiune	41.5mm x 19mm x 15.3mm
Greutate	5g

Culoare PCB	negru
Tensiune alimentare	2.5V-6V
Compatibilitate	Protocol I2C
Curent	2mA
Curent lumină de fundal	250mA (MAX)
Rezolutie	80 X 16
Culoare text	Albastru

Tabel 2.2 Specificațiile tehnice ale ecranului LCD , Sursa []

Figura 2.6 Pinii ecranului LCD

Pinii ecranului sunt urmatorii:

- Pinul VSS: reprezintă conexiunea la masă sau la referința de tensiune zero, asigură conexiuni stabile și funcționale între ecranul LCD și restul componentelor;
- Pinul VDD: reprezintă conexiunea la sursa de alimentare pozitivă pentru funcționarea corectă a ecranului LCD;
- Pinul RS (Register Select): este utilizat pentru a selecta modul de comunicare cu controlerul LCD,indică dacă datele trimise către ecran sunt comenzi de control sau date pentru afișajul propriu-zis;
- Pinul R/W (Read/Write): controlează direcția de transfer a datelor între microcontroler și ecranul LCD și indică dacă acesta este în modul de scriere sau de citire;
- Pinul E (Enable): este folosit pentru a activa sau dezactiva modul de lucru al acestuia, controlează momentul în care datele trimise către ecran sunt înregistrate și procesate;
- Pinii D0 , D1 , D2 , D3 , D4 , D5 , D6 , D7 sunt utilizați pentru a transfera datele de afișare către ecran;
- Pinul LED pozitiv: furnizează alimentarea la iluminarea de fundal a ecranului, furnizează tensiunea necesară pentru a alimenta sursa de lumină din spatele panoului LCD;

- Pinul LED negativ: este folosit pentru a lega terminalul negativ al iluminarii de fundal și este asociat cu pinul LED pozitiv;

Modulul I2C se montează direct pe panoul LCD și conține un potențiometru pentru reglarea contrastului , are nevoie de doar 2 fire pentru a comunica cu plăcuța Arduino Uno (cele două fire sunt necesare pentru clock și pentru date).Acesta are 20 de pini dintre care 16 pentru conectarea la ecran și alti 4 (SDA,SCL,GND,VCC) pentru la plăcuța Arduino Uno.

Modulul I2C și modul de conectare al adaptorului I2C la ecranul LCD este prezentat în figura următoare:

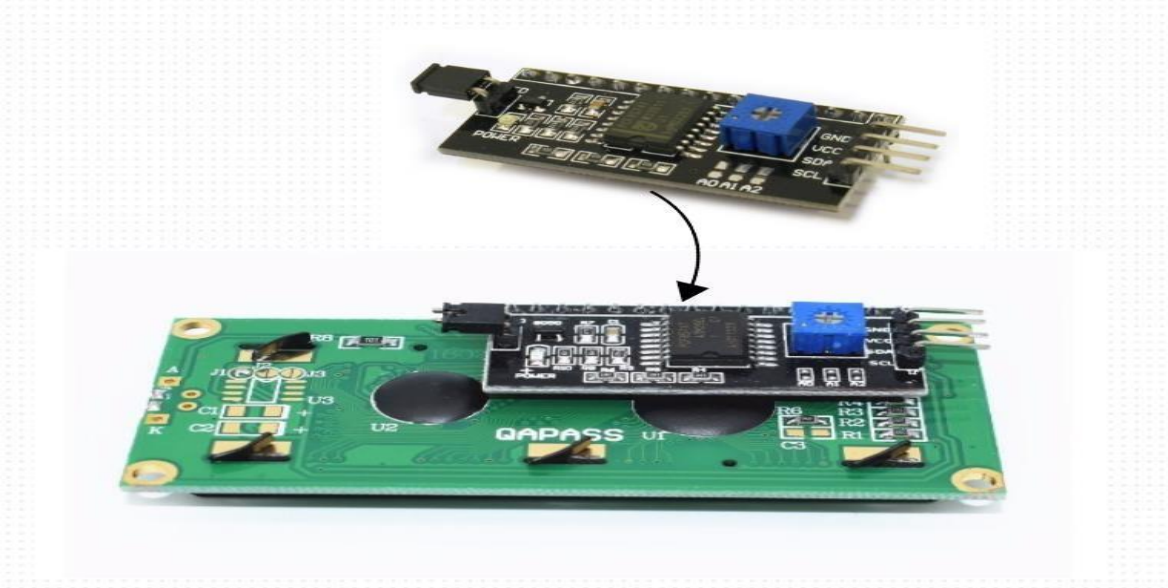

Figura 2.7 Modulul I2C și conectarea adaptorului la ecranul LCD

2.4 Protocolul de comunicație I2C

Protocolul de comunicare I2C este utilizat pentru a termina transferul de date între un afișaj LCD și un adaptor I2C conectat la panou.I2C utilizează doar două linii de comunicare, linia de date seriale (SDA) și linia de ceas serial (SCL), pentru a furniza o comunicație serială.I2C oferă o metodă simplă și eficientă de comunicare serială între mai multe dispozitive,folosind doar două linii de comunicare,linia de date seriale(SDA) și linia de clock serial (SCL).

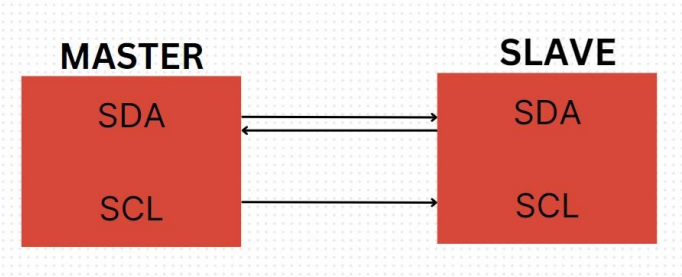

Figura 2.8 Protocolul de comunicatie I2C

În comunicarea I2C,există un dispozitiv master care inițiază și controlează transferul de date și unul sau mai multe dispozitive slave care răspund la comenzile primite de la master.În primul rând,comunicarea este inițiată prin trimiterea unui semnal START pe linia SDA (de la 1 la 0) atunci când linia SCL este la nivel înalt,ceea ce indică începutul funcționării I2C.Atunci când este selectat un dispozitiv slave,masterul trimite sau solicită date de la slave.Datele sunt trimise în format serial pe linia SDA, iar semnalul de ceas SCL sincronizează transferul de date.

Dispozitivul master primește sau trimite unul sau mai mulți octeți de date și așteaptă confirmarea din partea slave-ului prin semnalul ACK (Aknwoledge) pentru a continua transferul.Dispozitivul master răspunde cu un semnal ACK care indică dacă datele au fost primite cu succes.Atunci când toți octeții de date au fost recepționați sau transmiși ,dispozitivul master primește un semnal de STOP pe linia SDA (comutat de la 0 la 1) în timp ce linia SCL este la nivel înalt, fiind pregătită să încheie comunicarea.

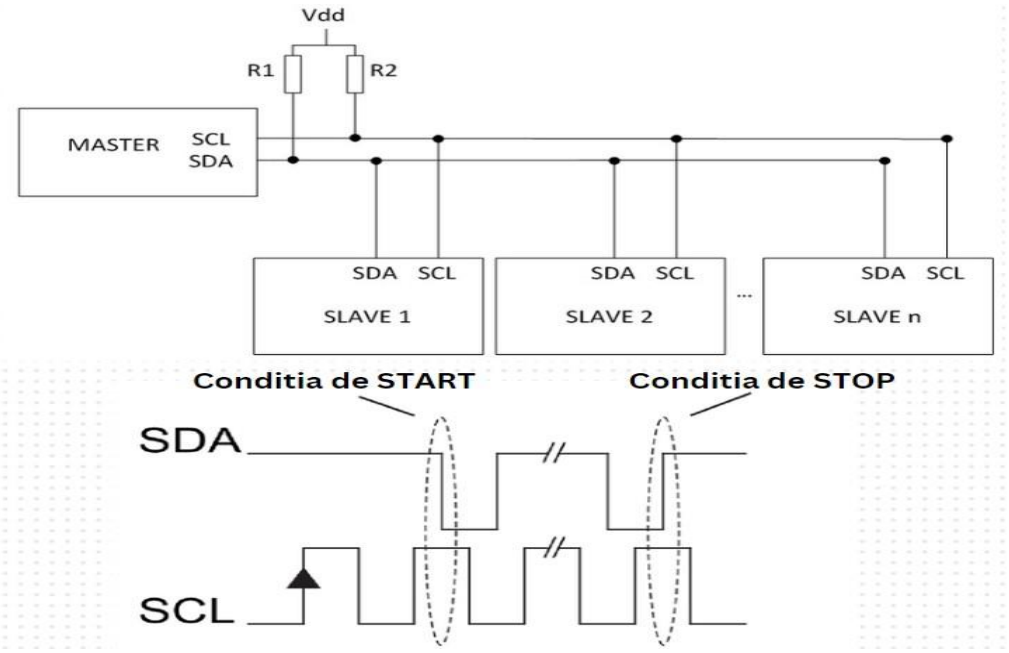

Figura 2.9 Liniile magistralei I2C și condițiile de Start și Stop

În continuare voi prezenta câteva avantaje și dezavantaje ale protocolului I2C.

Avantajele protocolului:

- Necesită doar două linii de comunicație (SDA și SCL);
- Fiecare dispozitiv conectat la magistrala I2C are o adresă unică;
- Protocolul permite atât transmiterea , cat și recepționarea datelor;
- Bitul de ACK recunoaște fiecare transfer de date cu succes;

Dezavantajele protocolului;

- Viteza de transfer mică față de SPI (Serial Peripheral Interface);
- Nu face fată la interfete electromagnetice;

În cele din urma , în figura de mai jos putem vizualiza modul de conectare a mai multe dispozitive master la mai multe dispozitive slave:

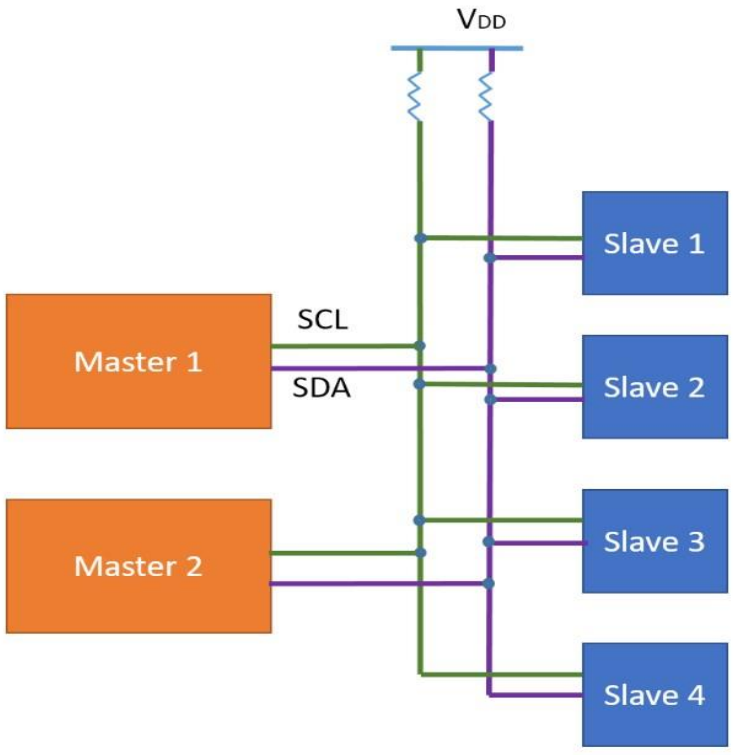

Figura 2.10 Conexiuni I2C care includ Multi-Master și Multi-Slave

2.5 Breadboard

O altă componentă folosită în lucrarea mea și una dintre cele mai esențiale pentru un număr mare de senzori este Breadboard-ul.Aceasta este o placă de test cu terminale și contacte în sistem grilă , unde putem introduce diferite componente electronice.Este folosit pentru realizarea mai ușoară a montajelor fara a fi nevoie săa folosim pistol de lipit și este ideal pentru testarea rapidă a proiectelor.Are un design dreptunghic , dar și pătratic și e acoperit cu găuri căptușite de metal , în care se introduc componentele (fire, leduri, rezistente, diode).

Pentru proiectul meu am ales un Breadboard MB-102 cu 830 de puncte , unul dintre cele mai folosite , prezentat în figura de mai jos:

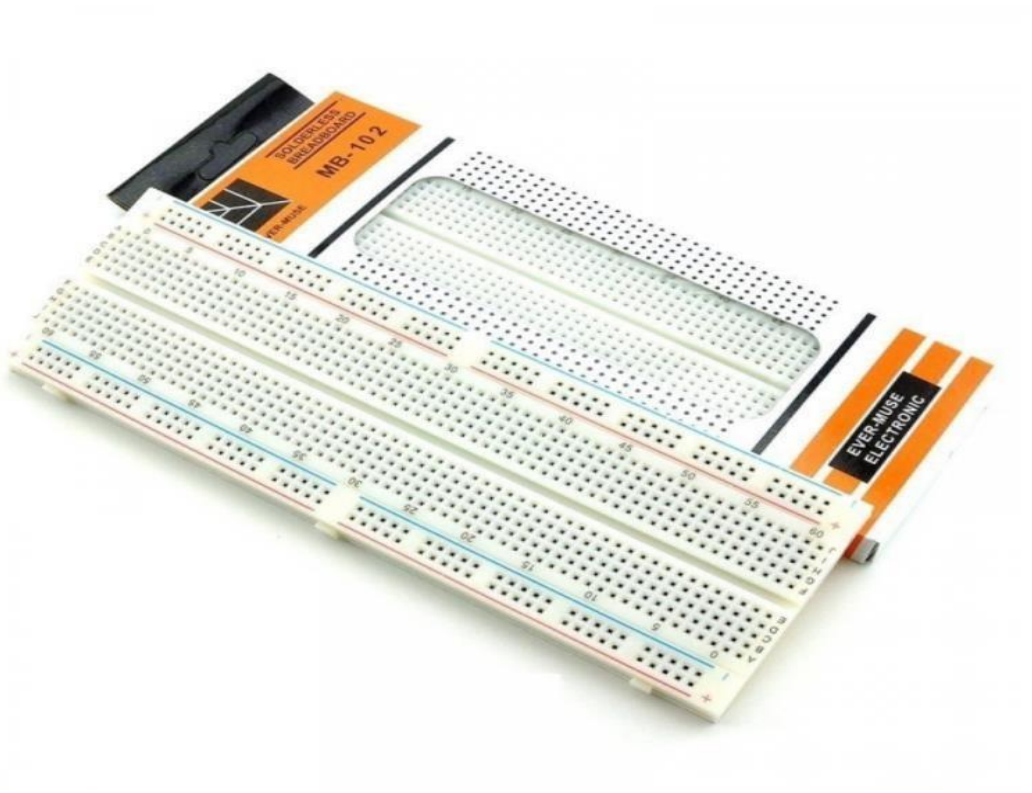

Figura 2.11 Breadboard MB-102

Pe suprafata acestuia avem un număr de 830 de găuri la o distanta de 0,1 inchi , iar în fiecare gaură se poate introduce doar un singur fir sau un terminal al unei componente.Sub capacul de

plastic , găurile sunt conectate în grupuri de o rețea de lamele din metal conceput din aliaje conducatoare de electricitate.

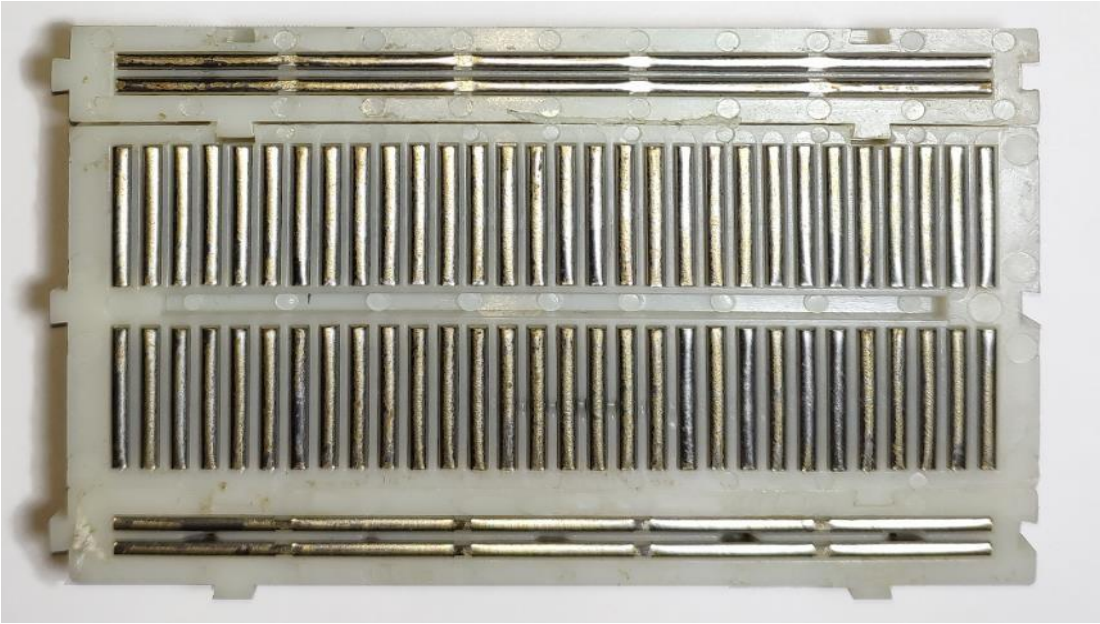

Figura 2.12 Interiorul Breadboard-ului

Un breadboard este împarțit în patru secțiuni , două interioare și două exterioare.Cele exterioare sunt dedicate în principal alimentării cu curent electric.Aceste secțiuni sunt marcate cu linii roșii și albastre , având semnele + și - , dar mai sunt marcate și cu cifre care au ca scop să te ajute să te orintezi în timpul asamblării circuitului.

2.6 Senzor IR infraroșu de obstacole și întrerupere

Pentru proiectul meu am ales doi senzori cu infraroșu , unul pentru evitarea obstacolelor si unul de întrerupere , ambii utilizând tehnologia infraroșie de detectare a obiectelor.

Totuși , întâlnim câteva diferențe între aceste două tipuri de senzori:

- Detectarea întreruperilor versus detectarea obstacolelor: Senzorul de obstacole este conceput pentru a indentifica obstacolele din calea senzorului , prin emiterea unui fascicul de lumină si recepționarea lui de către obiectele aflate în apropierea lui.Pe de alta parte , cel de înterupere , detectează întreruperi ale fasciculului de lumină , iar acestea pot fi cauzate de intrarea unui lucru în acea zonă (forma de U pe care o are);

- Au un design total diferit , dar ambii senzori au aceiași 3 pini (GND , OUT , VCC) , același compoarator LM393 , iar senzorul de întrerupere nu dispune de potențiometru de reglare , în timp ce senzorul de obstacole il are incorporat lângă comparator;

2.6.1 Senzor IR infraroșu de obstacole

Modulul senzorului IR de obstacole se bazează pe reflexia radiației IR emisă de LED-ul emisie și este recepționată de către o fotodiodă pentru a detecta prezența oricărui obstacol din fața modulului.Acesta are doua LED-uri , unul pentru alimentare și unul pentru a detecta piedica (unul de emisie și unul de recepție).

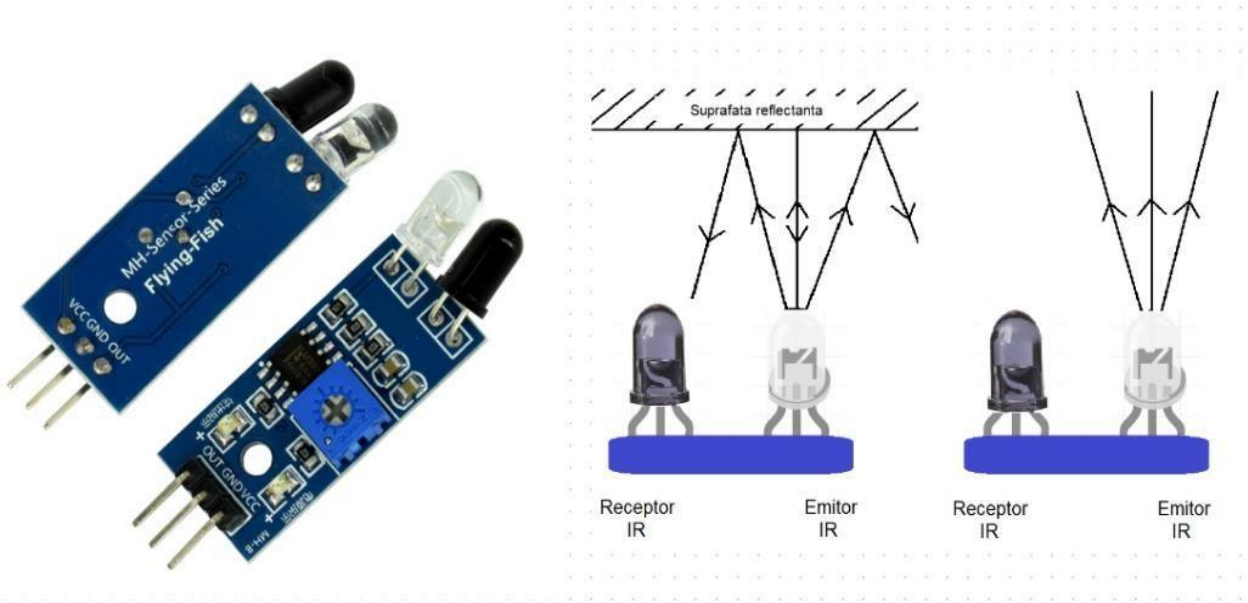

Figura 2.13 Modulul senzorui infraroșu de obstacole,recepția și emisia

Aceste modul conține 3 pini , Vcc , Gnd si Output , cele doua LED-uri menționate mai sus și un potențiometru care ajută la reglarea sensibilității sau distanței de detectare a

obstacolelor.Potențiometrul se rotește în sensul acelor de ceasornic pentru a crește distanța la care senzorul doreste să detecteze și invers daca dorim sa scădem distanța de detecție.

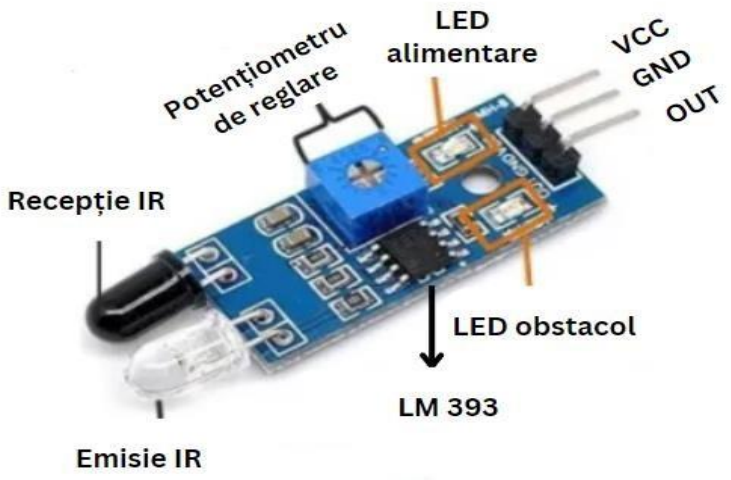

Figura 2.14 Componentele si pinii senzorului IR

Pinul VCC este utilizat pentru alimentarea senzorului IR și trebuie conectat la o sursă de tensiune între 3.3V si 5V , pinul GND este conectat la masă , iar pinul OUT este pentru a transmite semnalul detectat microcontrolerului.

Specificațiile tehnice:

- Distanța de detenție: 2 – 30 cm;
- Tensiunea de alimentare: 3.3V – 5V;
- Chip comparator: LM 393;
- Unghi de detenție: 35 °;

Principiul de funționare ala acestui senzor se bazează pe emiterea și recepționarea razelor infraroșii.Emițătorul IR transmite un fascicul de lumină infraroșie care este recepționată de către receptorul senzorului.Este măsurată cantitatea de lumină , iar pe baza ei se determină prezența sau absența unui obstacol în fața senzorului , astfel că dacă cantitatea de lumină este destul de semnificativă , senzorul știe ca are un obstacol în apropiere.

În continuare voi prezenta schema clasică a unui senzor IR, care e formată din două LEDuri , o fotodiodă , un potențiomentru , comparatorul LM393 , amplificatorul operațional care compară tenisuni și patru rezintențe.Primul Led se afla în serie cu un rezistor care reduce scurgerea curentului, iar LED-ul IR emite lumina intraroșie, fotodioda detectând-o.Comparatorul este in interiorul senzorului și oferă 1 logic la ieșire în cazul în care nu se detectează obstacole și 0 logic daca se detectează.

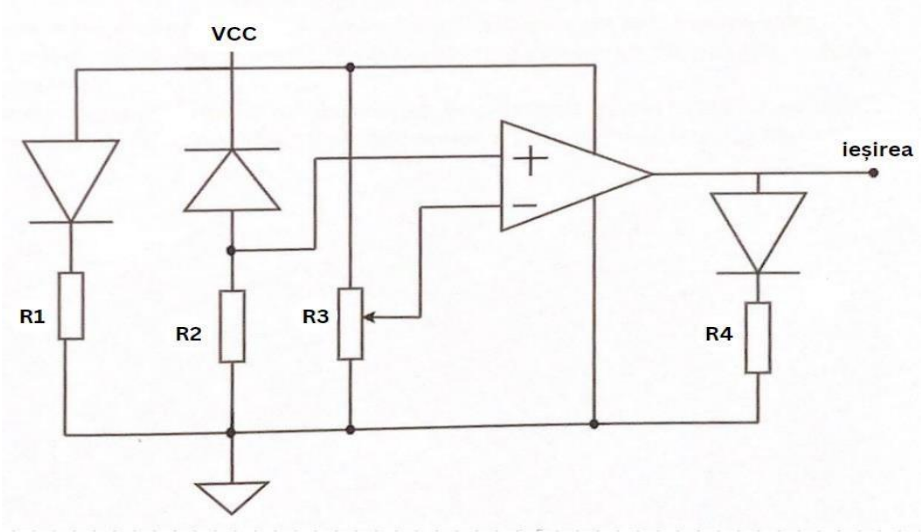

Figura 2.15 Schema clasică a senzorului de obstacole IR cu comparator LM 393

2.6.2 Senzor IR infraroșu de întrerupere

Senzorul de întrerupere IR are formă de U și are două componete: fototranzitorul (receptorul) și LED-ul ce emite infraroșu (emițătorul) , distanța dintre ele fiind de 10mm și se folosește pentru detectarea unui obiect aflat între cele două componente.

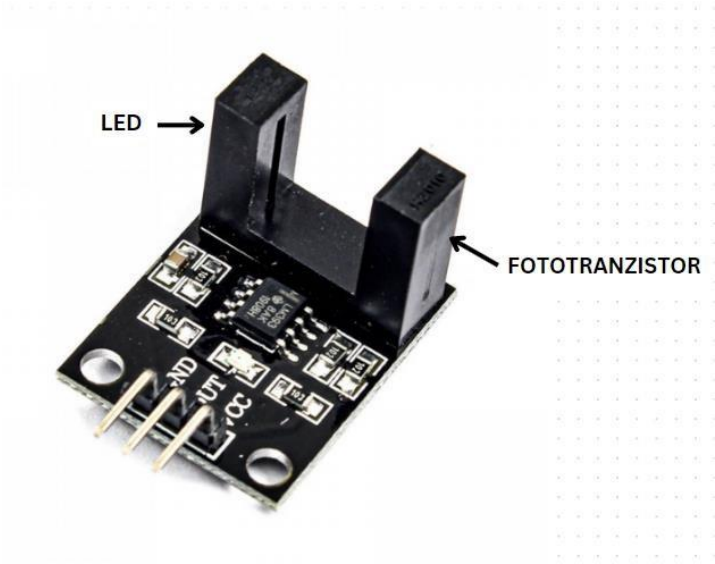

Figura 2.16 Senzorului IR de întrerupere

La fel ca la senzorul de obstacole , cel de întrerupere are aceiași 3 pini , GND , VCC și OUT , care au aceleași roluri la conectarea cu placa Arduino.

Caracteristicile tehnice:

- Cădere de tensiune LED: 1.2V;
- Curentul Led-ului: 50 mA maxim;
- Tensiune colector – emitor fototranzistor: 30V;
- Curent fototranzsitor: 20mA;
- Înălțime totală: 17.5 mm; Distanța între brațe: 10 mm;
- Chip comparator: Lm 393;

Principiul de funționare este unul destul de simplu , atunci când un obiect intră în zona dintre emițător (genereaza un fascicul de lumină infraroșie invizibilă) și receptor (recepționează acea lumină), raza infraroșie este întreruptă și receptorul nu mai recepționeaza semnalul IR , astfel acesta poate detecta prezența sau absența obiectului prin declanșarea unui dispozitiv , de exemplu aprinderea și stingerea unui LED.

Este utilizat in diverse aplicații și industrii , cum ar fi:

- În traficul rutier/feroviar: Senzorul poate fi montat de-a lungul unei căi pentu a detecta prezența unei mașini pentru a se monitoriza traficul;
- Sisteme de acces: Senzorul detecteaza prezența unei persoane și deschide automat o barieră sau o ușă;
- Sisteme de siguraţă: Senzorul identifică obiectul sau persoana și declanșeaza o alarmă;

Schema tipică a senzorului IR de întrerupere este la fel ca cea a senzorului de obstacole , ambii senzori având același comparator LM 393.

2.7 Senzorul ultrasonic

Senzorul ultasonic HC-SR04 este senzorul care utilizează undele ultrasunete pentru a detecta distanța. Undele ultrasonice sunt emise la o frecvență de 4000 Hz (0,04 MHz), deplasânduse prin aer și întorcându-se la senzor atunci când are în preajmă un obstacol.Timpul necesar pentru ca semnalul să se deplaseze înainte și înapoi este calculat ca distanță = viteză x timp.

Conține 4 pini:

1. VCC - pinul de alimentare al senzorului care trebuie conectat la o sursă de 5V pentru o funcționare corectă;
2. TRIG - pinul de trigger , utilizat pentru a iniția emisia ultrasunetelor , iar daca îl setăm la nivel logic înalt cu ajutorul funției “digitalWrite(trigPin , HIGH)” pentru o perioada scurtă de timp , senzorul incepe sa emită semnalul cu ultrasunete;
3. ECHO - pinul de ieșire care furnizează semnalul de revenire al undelor detectate de senzor.Acest pin emite un semnal de durată , proporțional cu timpul pe care semnalul îl parcurge dus-întors;
4. GND – pinul de masă (ground) care trebuie conectat la unii din pinii de masă al placuței Arduino Uno si creează o conexiune comună de masă între senzor si placă;

Figura 2.17 Senzorului ultrasonic (față – spate)

Specificațiile tehnice:

- Tensiunea de alimentare: 5V;
- Precizie: 0.3 cm;
- Curent consumat: 15mA;
- Distanța de funcționare: 2cm – 450 cm;
- Unghi de măsurare: 15 grade;
- Durată semnal input: 10 us(microsecunde);

Pentru a genera ultrasunetele , stabilim pinul TRIG pe o scară ridicată timp de 10us, iar acesta va trimite o rafală ultrasonică de 8 cicluri și se va deplasa cu viteza sunetului.Pinul ECHO trece la nivelul HIGH menționat mai sus imediat dupa trimiterea celor 8 cicluri și incepe să aștepte ca acea undă sa fie reflectată de un obiect.Daca nu se reflecteaza niciun impuls , pinul ECHO se va opri dupa 40us și va trece la starea joasă (LOW).

Daca primesc un impuls reflectat , pinul ECHO va coborî mai devreme de 40us , iar în funcție de perioada în care pinul a fost HIGH , aflăm distanța parcursă de unda sonoră.

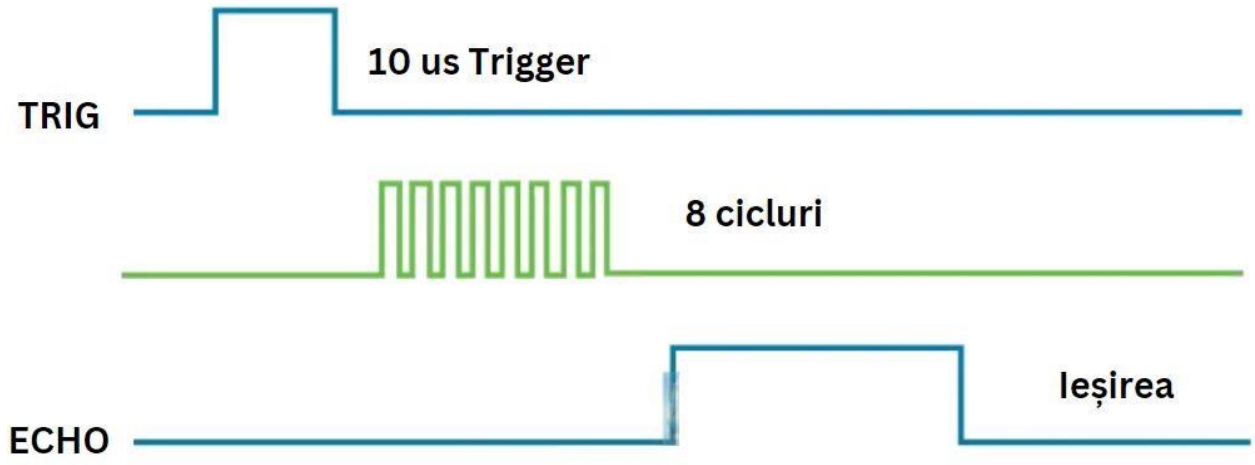

Figura 2.18 Pinii de trigger si echo

Senzorul HC-SR04 îl întâlnim într-o multiplitate de aplicatii , datorită abilității sale de măsurare prin intermediul undelor cu ultrasunete.Mai jos voi prezenta câteva aplicații în care senzorul se poate folosi:

- Robotică: senzorul este amplasat în roboți inteligenți pentru detectarea prezenței obiectelor și evitarea coliziunilor prin oprire sau schimbarea direcției;
- Monitorizarea nivelului de lichid: senzorul este folosit pentru a măsura nivelul de lichid dintr-un recipient cu ajutorul undelor ultrasunete care parcurg distanța dus-întors a repectivului rezervor;
- Sisteme de securitate si alarmă: poate declanșa alarme atunci când o persoană se apropie de o anumită distanță;
- Sisteme de parcare: Măsoară distanța dintre mașină și un obstacol din jur , face ca parcarea să fie mai sigură și oferă șoferului informații care îl ajuta la ghidare;
- Monitorizarea distanțelor în diferite locuri: Senzorul poate monitoriza distanța între rafturi, paleți, obiecte și ușurează planificarea aranjamentelor într-un mod mai benefic;

Voi prezinta mai jos câteva avantaje , dar si dezavantaje ale acestui senzor detector de distanță:

Avantaje:

- Cost redus: Senzorul HC-SR04 este un dispozitiv accesibil pentru toată lumea , având un preț mic , ceea ce îi atrage pe mulți utilizatori;
- Ușurința de utilizare: linii puține de cod și conexiuni simple; - Precizie: oferă o precizie exactă;
- Integrarea cu alte componente: este ușor de integrat , mai ales cu placa Arduino Uno și îi oferă posibilitatea de a lucra cu o gamă largă de componente electronice;

Dezavantaje:

- Necesite un mediu de lucru adecvat: În mediile cu zgomot , fum , praf, permformanțele senzorului ultrasonic pot fi afectate într-un mod negativ;
- Distanță limitată: poate măsura de la 2cm pana la 4m , ceea ce nu este tocmai în regulă atunci cand o aplicație necesită măsurători mai mari;
- Interfața cu alte mijloace care produc ultrasunete: Se pot crea interfețe cu alte dispozitive care produc ultrasunete , iar acest lucru duce la afectarea preciziei senzorului;

2.8 Senzorul de gaz metan

Senzorul de gaz MQ-4 este un senzor semiconductor care poate detecta prezența gazului metan (CH4) în aer și are o structură sensibilă la gaz și o rezistentă variabilă în funcție de concentrația de metan detectată.Este utilizat pentru monitorizarea și detectarea scurgerilor de gaz metan , dar poate detecta și alte tipuri de gaze.

Acesta conține , la fel ca cel ultrasonic , 4 pini:

1. VCC – pinul de alimentare al senzorului care trebuie conectat la o sursă de 5V pentru a furniza energia necesară funcționării senzorului;
2. GND – pinul de legare la masă (ground) trebuie conectat la masa placuței Arduino Uno pentru a stabiliti o conexiune comună de tenisune;
3. AO – pinul analog output , furnizează o valoare analogică relativă cu valoarea de gaz metan identificată și se conectează la un pin analogic de la placa Arduino;
4. DO – pinul de ieșire digitală , care poate fi setat să livreze un semnal digital LOW sau HIGH cu funcția digitalWrite(DOpin , LOW/HIGH);

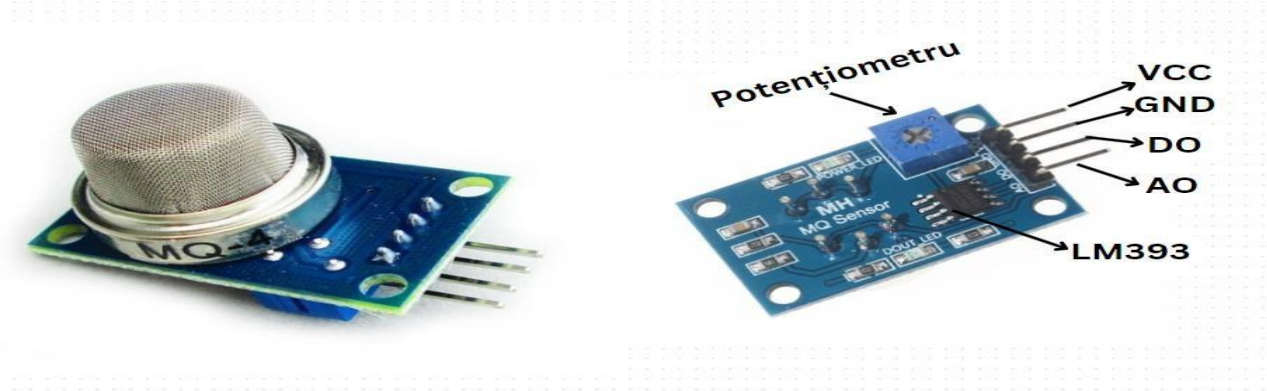

Figura 2.19 Senzorul de gaz metan (față – spate)

Pe fața senzorului observam o carcasă de metal care conține componentele interne pentru identificara metanului , iar pe spate avem , pe lângă pinii de conexiune , comparatorul LM393 și potențiometrul pe care îl utilizăm pentru calibrarea senzorului , dar și pentru a ajusta nivelul de detecție al gazului.El se reglează cu ajutorul unei chei de mărime potrivită și introducerea ei printrun orificiu mic din interior carcasei albastre. Caracteristicile tehnice:

- Tensiunea de alimentare: 5V;
- Curent: 150mA;
- Concentrație de detecție: 300 – 10.000 ppm (părți per milion);
- Tempratuă: -10 °C +50 °C;
- Dimensiuni: 32 x 20 x 22 mm;
- Timp de răspuns: rapid;

Principiul de funcționare al senzorului MQ-4:

Senzorul utilizează un oxid de staniu (SnO2) , care e un element semiconductor sensibil la gazul metan.Atunci când metanul intră în contact cu suprafața senzorului , are loc o reacție chimică între gaz și acest oxid , care duce la modificarea nivelului conductiv al senzorului.Prezența gazului metan produce scăderea rezistenței senzorului , iar cu cât concentrația este mai mare , cu atât rezistența e mai mică.Pentru a măsura variația rezistenței , avem integrat potențiometrul despre care am vorbit mai sus împreună cu comparatorul LM393 , situat tot pe spatele senzorului , care compară rezistența senzorului cu valaorea de referință.Placa Arduino citește semnalul furnizat de la senzor și indică prezenta gazului metan.

Factorii care influențează performațele senzorului MQ-4:

Un prim factor ar fi sensibilitatea la alte tipuri de gaze ,astfel încât acest tip de senzor nu doar ca poate detecta gaz metan , dar poate indentifica și alte tipuri de hidrocarburi (propan , butan), iar acest lucru poate duce la rezultate incorecte și la interferențe în cazul în care aceste gaze sunt prezente în mediu.

Al doilea factor se datorează creșterii umidității și a temperaturii, care au un impact negativ asupra performanței senzorilor. Creșterea umidității determină creșterea rezistenței, în timp ce temperaturile scăzute reține gazul metan de partea senzorului.

Ultimul factor este reprezentat de interferențele magnetice și electrice din mediul înconjurător, care distorsionează semnalul senzorului și duce la rezultate false, astfel încât este important să se ia în considerare protecția împotriva acestora.

Pe langă micile dezavantaje prezentate adineauri , senzorul poate fi folosit in diverse domenii:

- Monitorizarea mediului încojurător: monitorizează gazul metan din mediu pentru eventualele emisii , iar cu ajutorul lui se studiază calitatea aerului;
- Industria gazelor naturale (petrolieră , chimică): detectează gazul în diferite instalații pentru a previni oamenii de eventualele accidente , explozii;
- Industia automotive: detectează scurgerile de gaz din rezervoare și din sistemul de alimentare , dar poate fi integrat într-un sistem de alarmă care poate semnala șoferul cu privire la scurgere;

2.9 Senzorul de mișcare PIR

Senzorul de mișcare PIR (Passive Infrared) HC-RS501 este unul dintre cele mai comune și populare module PIR de pe piață , fiind un senzor sensibil care detectează mișcarile unei surse generatoare de căldură și se bazează pe principiul emisiei și recepționării radiației de căldură.Pe fața senzorului avem o carcasă de plastic , cu rol de protectie al elementelor aflate în interiorul ei și anume:

- Elementul PIR , care este componenta principală a senzorului și detectează lumina infraroșie;
- Lentilele , care optimizează performanțele de detectare a mișcării prin focalizarea radiației infraroșii de pe elementul PIR;

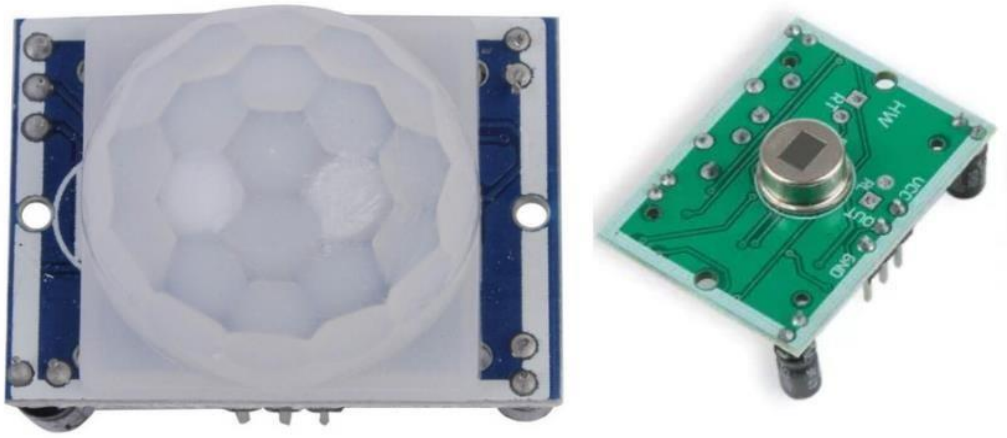

Figura 2.20 Fața senzorului PIR și interiorul carcasei

Pe spatele Senzorului PIR HC-RS501, sunt amplasate diverse componente:

- Cei trei pini: VCC (pinul de alimentare care trebuie conectat la un pin cu o sursă de 5V de pe placa Arduino) , GND (pinul conectat la masă) și OUT (pinul de ieșire care furnizează semnal de ieșire în funcție de detectarea mișcării , semnal ce poate fi digital (HIGH/LOW) sau analogic în funcție de configurație.
-
- Potențiometrul de delay (timp de întâriziere): regleză timpul pentru care semnalul de ieșire rămâne activ dupa ce mișcarea a fost detectată.Maximul este de 200 secunde, iar minimul de 5 secunde.
-
- Potențiometrul de sensibilitate: reglează cat de sensibilă să fie detectarea mișcării și ajustarea se face în sensul acelor de ceasonic , la fel si pentru cel de delay.Maximul este de 7 metri , iar minimul este de 3 metri.
-
- Jumper H (HIGH): atunci când jumperul este la conectat nivelul H , este configurat în modul de declanșare repetată și permite senzorului să declanșeze semnalul de ieșire atunci când detectează mișcare în câmpul său de vizualizare , iar semnalul de ieșire rămâne activ pe toată durata setată de potențiometrul de delay.
-
- Jumper L (LOW): senzorul este setat în modul de declanșare unică atunci când este conectat la nivelul L și declanșează semnalul de ieșire o singură dată atunci când senzorul detectează o mișcare în câmpul vizual.
-

- Regulator de 3,3 V: lucrează la tensiunea de 5 V, dar furnizează o tensiune de 3,3 V pentru a alimenta intern senzorul , iar unele dintre componentele interne ale acestuia pot necesita o tensiune mai mică de 3,3 V.
-
- Circuitul integrat BISS0001: prelucrează semnalele de la senzor pentru o funcționare cât mai corectă.

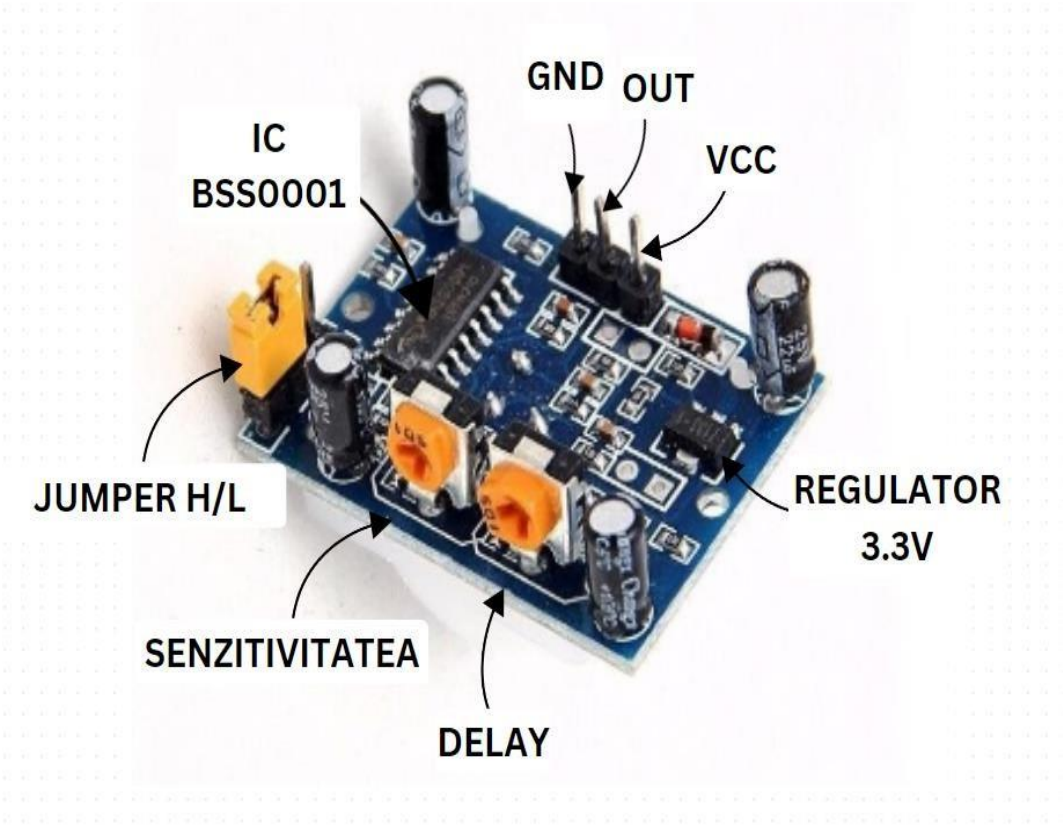

Figura 2.21 Spatele senzorului PIR HC-RS501 și componentele sale

Specificatiile tehnice ale senzorului:

- Culoare: alb + verde;
- Dimensiuni: 3.2 cm x 2.4 cm x 1.8 cm;
- Tensiune de operare: 3V – 5V;
- Curent consumat: < 60uA;
- Tensiunea de ieșire: High/Low level signal – 3.3V TTL output;
- Distanța de detecție: 3 – 7 metri (se poate ajusta);
- Unghi de detecție: <140°;
- Timp de întârziere: 5 – 200 secunde (se poate ajusta);
- Timp de blocare: 0.2 secunde;

- Temperatură de operare: -20 +80°C;
- Metodă de declanșare: H declanșare repetabilă / L declanșare nerepetabilă;

Principiul de funcționare al senzorului PIR HC-RS501:

Senzorul conține elementul PIR care este alcătuit dintr-un element semiconductor sensibil la radiația infraroșie și are în componența sa niște detectoare mici care reacționează la variațiile de temperatură cauzate de obiectele din jurul său.Acestea emit , în funcție de temperatură , radiație infraroșie , iar cand intră în câmpul lui de vizualizare , senzorul schimbă distribuția de radiație detectată de acele elemente de detectare menționate adineauri.Circuit integrat BISS0001 filtrează și prelucrează semnalul de la elementul PIR și elimină alte interferențe și zgomote , iar pe baza semnalului prelucrat , senzorul genereaza un semnal de ieșire care este digital și indica prezența sau absența mișcării (detectaza mișcare => semnal activat:HIGH , dacă nu => semnal dezactivat: LOW).

Astfel , prin utilizarea principiului infraroșu și prelucrarea semnalului de la elementul PIR, senzorul HC-RS501 poate fi utilizat într-o gamă largă de aplicații:

1. Automatizarea clădirilor: controlează și monitorizează prezența umană în diferite cladiri;

2. Iluminatul stradal inteligent: detectează prezența mișcării pietonilor pe timp de noapte și activează iluminatul atunci cand ei trec prin zona lor de vizualizare;

3. Sistemel de jocuri interactive: detectează mișcarea jucătorului și o transpune în acțiuni în joc;

4. Sisteme de control al climatizării: senzorul poate fi integrat într-un termostat , iar când detectează prezența umană , reglează fluxul de aer și temperatura pentru un bun ambient;

2.10 Senzorul de temperatură și umiditate

Senzorul DHT11 este senzorul digital care detectează temperatura și umiditatea,cu un preț de piată scăzut și utilizat în multe aplicații în care se calculează temperatura și umiditatea.Pe partea frontală a senzorului, sub capacul albastru de protecție, se află un termistor pentru a măsura temperatura ambientală, un senzor capacitiv pentru umiditate și patru pini (VCC, GND, NC și DATA). Pe partea din spate a senzorului se află un microcontroler pe 8 biți.

Principiul de funcționare al termistorului se bazează pe rezistența sa , care este invers proporțională cu temperatura , iar pe măsura ce crește temperatura , rezistența scade și invers ,

scade temperatura , crește rezistența.Astfel , măsoară temperatura ambientală , iar senzorul convertește această valoare într-un semnal digital.

Senzorii de umiditate constau într-un strat de sticlă format din două plăci unite între ele,separate de un dielectric.Principiul de funcționare constă în faptul că dielectricul modifică rezistența și capacitatea senzorului,capacitatea modificându-se în funcție de umiditate și variația acesteia , transformându-se într-un semnal digital.Acest material încorporează umezeala și astfel se schimbă capacitatea senzorului în funcție de umezeala din mediul.Astfel se convertește variația capacității într-un semnal digital.

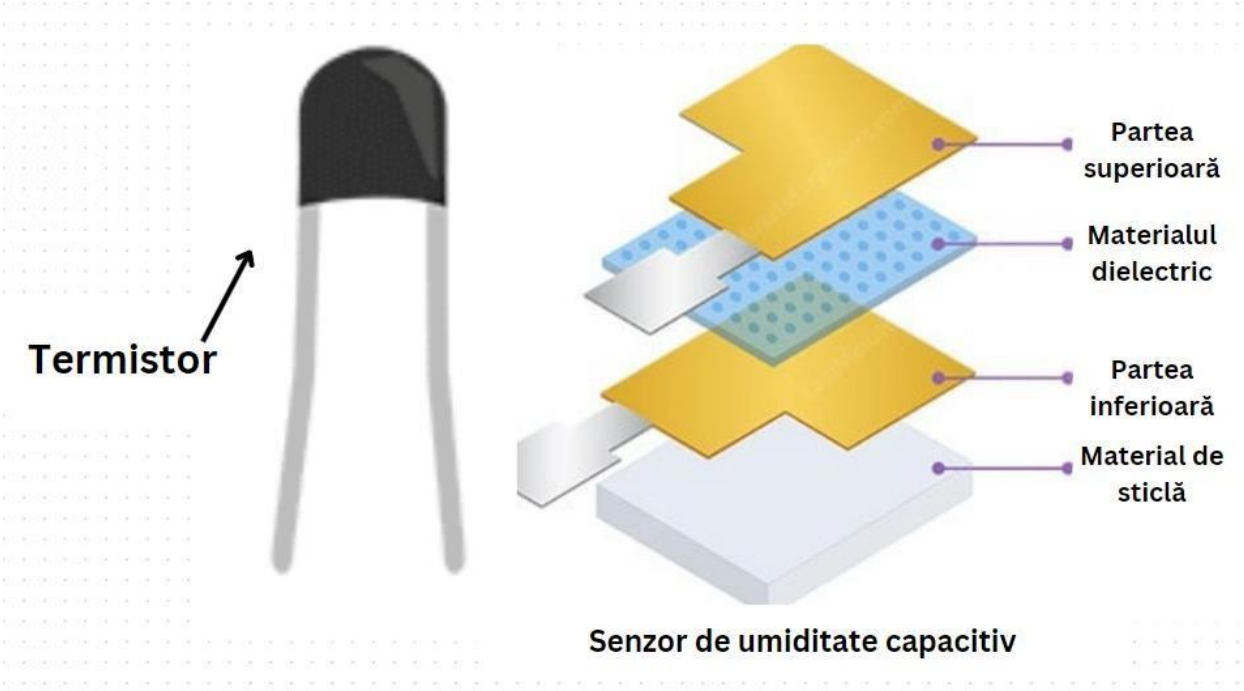

Figura 2.22 Elementele cheie ale senzorului DHT11

Cei patru pini din interiorul senzorului DHT11 îndeplinesc următoarele cerințe:

- Pinul VCC: pinul de alimentare care trebuie conectat la o sursă cuprinsă între 3.3V și 5V;
- Pinul GND: pinul de masă care trebuie conectat la masa plăcii Arduino;
- Pinul DATA: pinul de date este servit pentru comunicarea între senzor și placa Arduino, se transmit date digitale de temepratură și umiditate prin intermediul acestui pin;
- Pinul NC (Not Connected): acest pin nu este conectat pentru a ajuta la funcționalitatea senzorului și este lăsat neconectat , dar poate fi comutat la GND pentru a ne feri de eventualele interferențe;

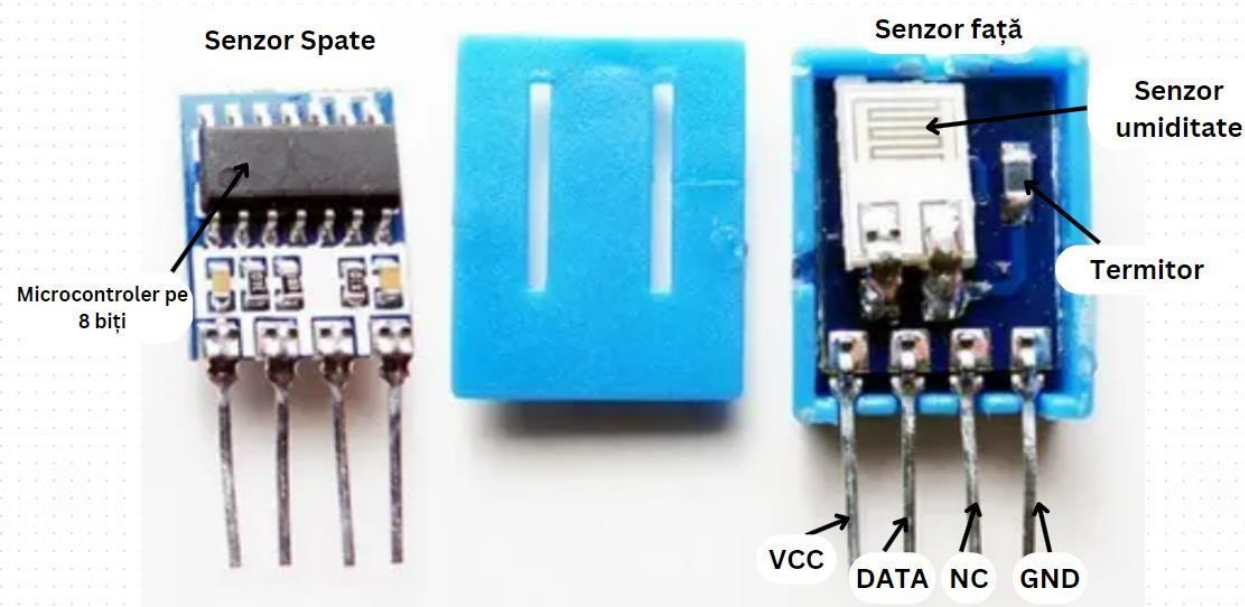

Figura 2.23 Fața , spatele și pinii senzorului DHT11

Pentru proiectul meu am ales modulul senzorului DHT11 , pentru o integrare mai ușoară, fiind convenabil în utilizare dearece are componente suplimentare precum gaura de montare care ajută la o instalare usoară și cei trei pini pe care îi conectăm la dispozitivul extern , placa Arduino Uno.

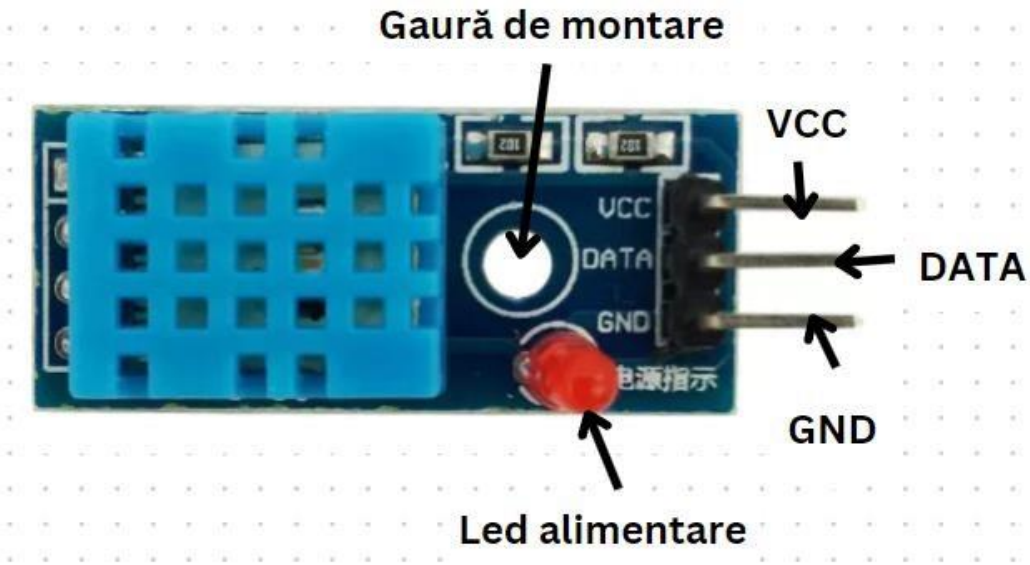

Figura 2.24 Modului senzorului DHT11

Ca și ceilalți pini explicați mai sus , pinul VCC este cel de alimentare 5V la placa Arduino, pinul GND este pinul masă , iar pinul digital OUT este pinul de ieșire care trimite comunicarea.Acesta

mai are incorporat un LED de alimentare , de culoare roșie , care arată starea de funcționare a senzorului.

Pentru a se începe măsurarea temperaturii și a umidității , senzorul se inițializează prin stabilirea unui nivel logic de tensiune înaltă (HIGH) pe pinul DATA , pe o perioadă de timp scurtă.Urmează o tranziție de joasă tenisune (LOW) pentru comunicarea cu senzorul, placa Arduino Uno trimite o cerere de date către senzor pentru a indica începerea măsurătorilor în mediu, iar la final senzorul creează semnale digitale în funcție de valorile obținute.Placa va prelua semnalele digitale de la senzor prin pinul de date și le va afișa pe un ecran LCD.

Specificații tehnice:

- Tensiunea de I/O: 3.3 - 5V;
- Curent maxim: 2.5mA;
- Interval umiditate: 20%-80% cu o acuratețe de + sau - 5%;
- Interval temperatură: 0 °C - 50°C cu o precizie de + sau - 2°C;
- Rata de eșanționare: 1Hz (una pe secundă);
- Dimensiuni: 15.5mm x 12mm x 5.5mm;
- Culoare: albastru și negru;

Senzorul DHT11 este întâlnit într-o gamă largă de aplicații în care măsurarea umidității și a temperaturii este una esențială.Mai jos voi prezenta câteva exemple:

- Monitorizarea mediului înconjurător: determină temperatura și umiditatea dintr-un spațiu și este folositor pentru controlul sistemelor de ventilație,irigare;
- Sisteme meteorologice: poate fi utilizat într-un proiect de construire a unui sistem de monitorizare a condițiilor meteo într-un oraș sau sat și afișarea datelor pe un panou;
- Agricultură: este integrat în sisteme de irigare pentru a controla si regla temperatura și umiditatea astfel încât să se creeze un mediu optim pentru creșterea plantelor;
- Diverse proiecte pentru începători: este utilizat adesea foarte mult în domeniu Internet Of Things (IoT) , fiind foarte accesibil pentru studenți , dar și pentru cei începători care vor sa învețe mai multe despre electronică și Arduino;
- Sistemel de ventilație și climatizare: supraveghează temperatura și umiditatea în diferite încăperi pentru a garanta o calitate a aerului perfectă și un confort termic potrivit;

2.11 Senzorul de curent

Pentru lucrarea mea am ales senzorul de curent ACS712 de 5A pentru a fi compatibil cu placa Arduino.El are rolul general de a măsoară curentul într-un circuit și a oferi o monitorizare în timp real a curentului.Acesta are efect Hall , ceea ce înseamnă că folosește un senzor Hall care detectează câmpul magnetic al curentului electric ce trece prin respectivul conductor , avantajul Hall-ului fiind că nu necesită întreruperea sau atașarea directă a senzorului la conductor.Senzorul ACS712 amplifică semnalul primit de la senzorul Hall și îl convertește într-un semnal analogic proporțional cu valoarea curentului.

Circuitul integrat ACS712 sau amplificatorul de instrumentație este partea centrală a senzorului și include 8 pini , 4 pe fiecare parte și este de tip ACS712 ELC-05.El frumizează o tensiune analogică de ieșire potrivită curentului care trece prin bornele de detecție.

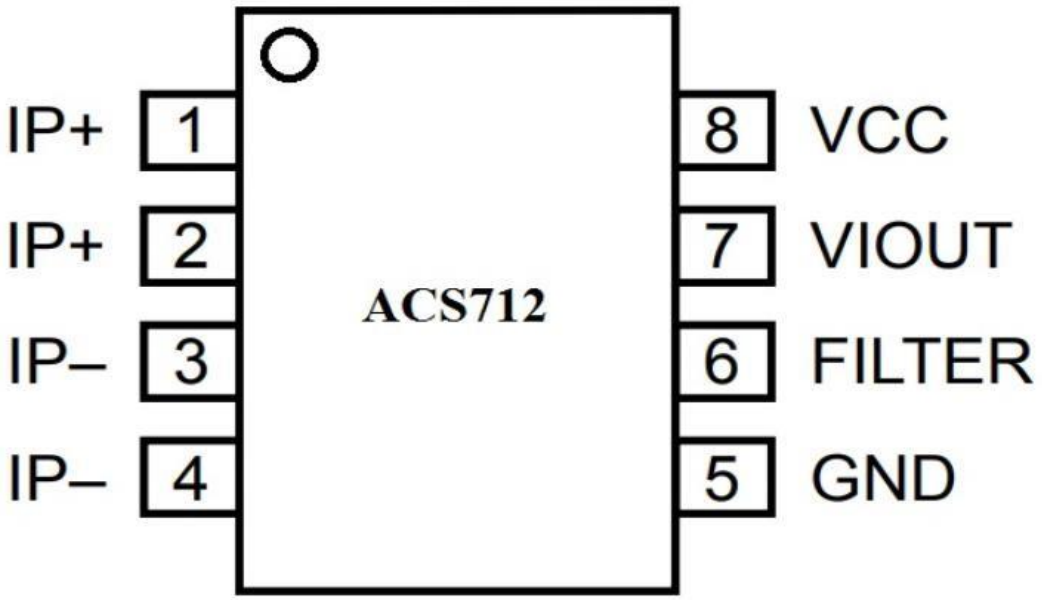

Figura 2.25 Digrama circuitului integrat ACS712

Pinii IP+ și IP- au rolul de a conecta semnalul diferențial de intrare (IP+ pentru semnalul pozitiv și IP- pentru semnalul negativ).Pe cealaltă parte , pinii VCC ș GND îi folosim pentru tenisunea de alimentare , respectiv tensiunea la masă.Pinul FILTER îl folosim pentru a conecta o anumită componentă suplimentară la circuit și pinul VIOUT pentru a furniza o ieșire a semnalului analogic în funcție de ce intensitate are curentul.

Principiul de funcționare al modulului senzorului ACS712 de 5A este bazat pe senzorul Hall integrat care , după cum am spus mai sus , detectează câmpul magnetic generat de curentul electric și produce un semnal proporțional cu intensitatea acesui câmp.Semnalul de la senzorul Hall este amplificat de către circuitul integrat prezentat adineauri pentru a se obține un semnal mai ușor de

măsurat , iar acest proces asigură o sensibilitate si o precizie mai optimă în măsurarea curentului.Alimentarea se face , ca la toți ceilalti senzori , folosind pinii VCC și GND , pe când pinul OUT este pinul care transmite semnalul de ieșire către placa Arduino.

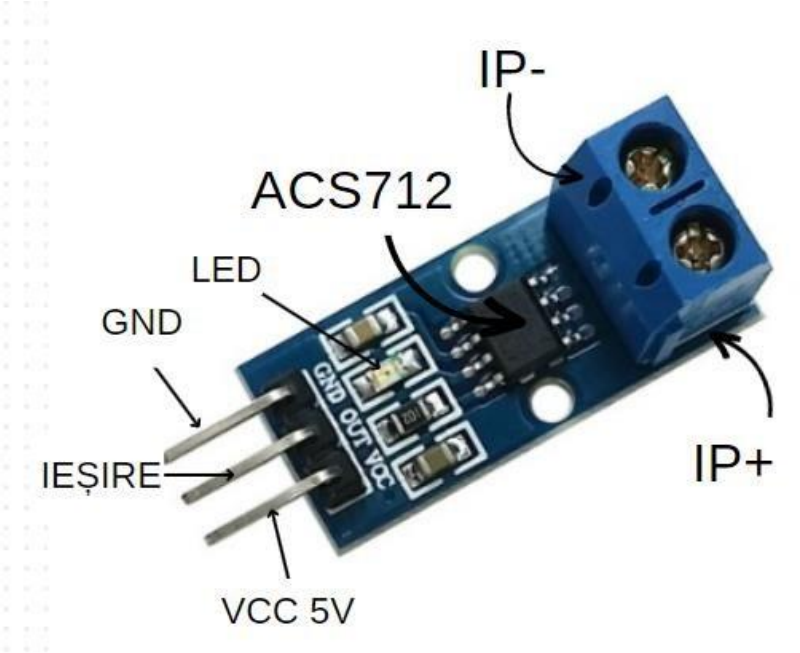

Figura 2.26 Modului senzorului de curent ACS712 si componentele sale

Caracteristici tehnice:

- Tensiunea de alimentare: 5V;
- Curent consumat: 10mA;
- Tensiunea maximă de curent consumat: 300v;
- Curent maxim măsurat: 5V;
- Led pentrua indica alimentarea;
- Rezistență internă: 1.2mOhmi; Output: 66 – 185 mA;
- Dimensiuni: 27 x 12 x 14 mm;
- Lățime bandă: 80kHz;

Acest senzor de curent prezintă unele avatantaje și dezavantaje și voi prezenta câteva dintre ele.

Avantaje:

- Ușurința de utilizare: modulul senzorului este conceput pentru a fi facil în diferite proiecte și aplicatii în care se măsoară nivelul de curent;
- Măsurare precisă: senzorul Hall și circuitul integrat asigură o măsurare precisă a curentului electric , permitând să îi determinăm valoarea cu o precizie care să ne convină;

- Compatibil cu nivelul de tensiune familiare: senzorul ACS712 de 5A este compatibil cu majoritatea nivelurilor de tensiune , iar de aceea il putem introduce într-o varietate de aplicații electronice , în cazul meu o tensiune de 5V;

Dezavantaje:

- Consumul de energie: senzorul are un consum de energie ridicat în timpul funcționării;
- Sensibil la câmpuri magnetice externe: acest lucru poate influența măsurătorile , iar uneori este necesară protecția față de aceste interfețe;
- Erori de offset: pot afecta precizia unei măsurători și ele pot necesita calibrare pentru a obține o valoare cat mai convenabilă;

Pe senzorul ACS712 il putem întâlni într-o multiplitate de aplicatii și sisteme electronice , în care se măsoară curentul electric.

- Monitorizarea consumului de energie pentru diferite dispozitive (electrocasnice,sisteme de iluminat) și identificarea surselor de consum in exces;
- Sistemele de control ale motoarelor în care senzorul este integrat , folosit pentru protecția supracurentului, iar cu ajutorul măsurătorii obținem informații despre performanțele acestuia;
- Încarcarea bateriilor la o mașină electrică;
- Echipamente de sudură;
- Sisteme de protecție împotriva scurtcircuitelor;

2.12 Senzorul picături de ploaie

Senzorul picături de apă HL-83 este un senzor ce detectează și măsoară prezența apei, utilizând tehnologia bazată pe conductivitatea electrică a apei.Acesta constă într-un set de piste care sunt sensibile la umezeală și sunt amplasate pe placa senzorului , iar atunci când apa ajunge pe aceasta placă , conductivitatea electrică a acestor piste se modifica și poate fi detectată de către dispozitivele electrice externe , în cazul meu placa Arduino Uno.

El vine la pachet cu un modul prin care este conectat la placa Arduino.Are ca și componente 4 pini (de alimentare , de masă și pentru semnal analogic/digital) , 2 pini (+ și -) pentru a conecta placa senzorului la modul , un potențiometru pentru a calibra sensibilitatea senzorului, comparatorul de tensiune LM393, două LED-uri , unul pentru alimentare și unul pentru semnalul digital.

Comparatorul de tensiune LM393 este circuitul integrat al modulului și are două comparatoare independente care compară tensiunile de intrare și furnizează un semnal digital în funcție de acestea.Pe de alta parte , potențiometrul rotativ de reglare al intensității permite ajustarea

sensibilității senzorului în contact cu apa și îl putem regla atunci cand senzorul este udat, pentru a obține rezultate cât mai bune.

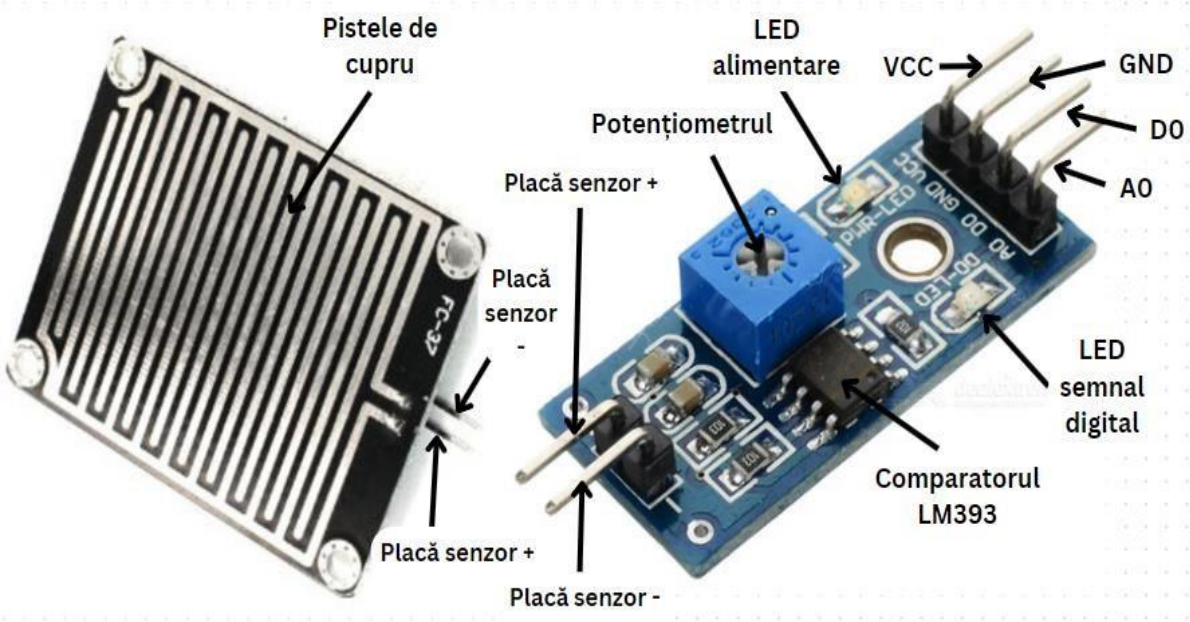

Figura 2.27 Senzorul picături de apă și modulul lui

Caracteristici tehnice:

- Tensiunea de operare: 3.3 – 5V;
- Dimensiunea modul: 3.2cm x 1.4 cm;
- Dimensiune placă: 5.5 cm x 4.0 cm;
- Anti-oxidare;
- Ieșire digitală și ieșire analogică;
- Chip comparator tenisune: LM393;
- Greutate: 8 grame;

Principiul de funcționare și modul de conectare al senzorului la placa Arduino Uno:

Principiul de funcționare constă în modificarea capacității în funcție de cantitatea de apă adăugată pe stratul senzorului , fiind foarte ușor de înteles.

Înțelegând principiul de funcționare , putem conecta senzorul HL-83 la placa Arduino Uno prin conectarea pinului VCC la pinul de 5V al plăcii pentru a asigura alimentarea senzorului , pinul GND la pinul masă Arduino.Pentru a se stabili conexiunea semnalului , el are doi pini de ieșire, unul digital (conectăm la pinul digital 2 de pe placă) și unul analogic (conectăm la pinul analogic A0 de pe placă) , iar în funcție de nevoie se conecteaza la unul din ei.După ce s-a realizat conexiunea , programăm placa prin programul Arduino IDE pentru a citi si a furniza semnalul de la senzor.

Ca să aflăm cantitatea de apa , ne bazăm pe rezistența electrică, astfel că senzorul este alcătuit din mai mulți de electrozi din material conductiv , care sunt răspândiți într-un anumit fel pentru a măsura modificarea rezistenței. Rezistența are o valoare inițială înainte de a intra în contact cu apa și scade atunci când intră în contact cu apa (au o valoare inițială înainte să intre in contact cu apa , iar la contact cu apa rezistența scade).

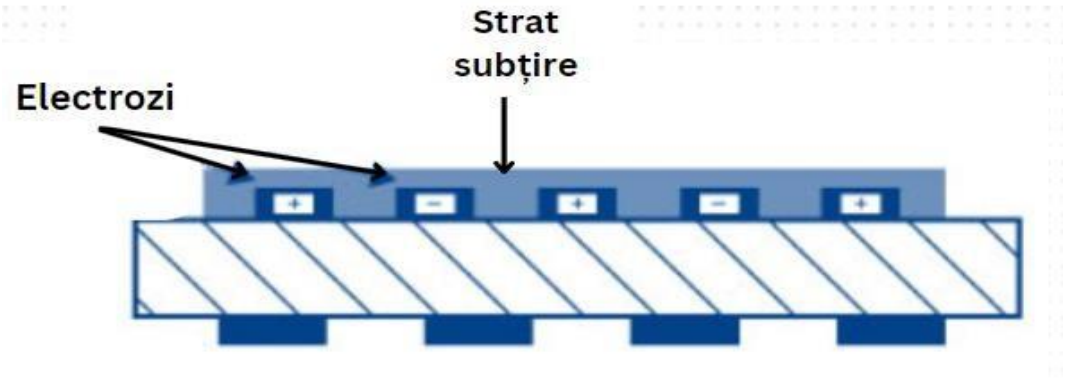

Figura 2.28 Structura senzorului

Senzorul HL-83 este utilizat într-o varietate de sisteme și aplicații în care se măsoară cantitatea de apă.Voi prezenta câteva dintre ele:

- Sisteme de irigare: monitorizează umiditatea solului și permite un control precis al cantității de apă destinată plantelor , în funcție de nevoia fiecăreia;
- Calitatea solului: crește sau scade nivelul de umiditate al solului și colectează date legate de pH-ul apei , nutrienții;
- Monitorizarea mediului: Evaluează umiditatea din aer în diferite medii pentru a evalua sănătatea ecosistemului;
- Sisteme de prevenire a inundațiilor: detectează prezența și nivelul ridicat din apă într-o zonă inundată;

Senzorul de apă poate fi afectat de factori de influență și este important să nu ignorăm acest aspect atunci când faceți ajustări:

- Fluctuațiile de temperatură: în jurul unor temperaturi foarte scăzute sau foarte ridicate, caracteristicile senzorului pot cauza erori de măsurare,astfel încât este important să se asigure o calibrare bună;
- Substanțele chimice: prezența lor în mediu pot afecta conductivitatea electrică și pot crea depuneri pe placa senzorului;
- Alte surse umede: pot apărea alte surse de umiditatea în jurul senzorului din cauza condensului sau a altor scurgeri care afectează măsurătorile deoarece senzorul nu poate face diferența între apa pe care dorim să o turnăm și alte surse;

2.13 Senzorul capacitiv cu patru canale

Senzorul capacitiv cu patru canale , numit TTP224 , este un senzor cu toch-pad care detectează atingerile și care oferă patru taste capacitive.Este proiectat pentu a oferi o interfață simplă pentru detectarea tactilă.Când o persoană sau un obiect atinge unele din cele patru suprafețe sensibile , capacitatea electrică a respectivului canal se schimbă , iar senzorul generează un semnal pe pinul de ieșire corespunzător.Are incoportat un circuit integrat de tip TTP224 , iar pe lângă componentele auxiliare (rezistențele , condensatoarele și diodele) care asigură funcționarea corectă și protecția circuitului , are un total de șase pini , dintre care patru sunt cei de ieșire (OUT1 , OUT2, OUT3 , OUT4) și doi de alimentare (VCC și GND).

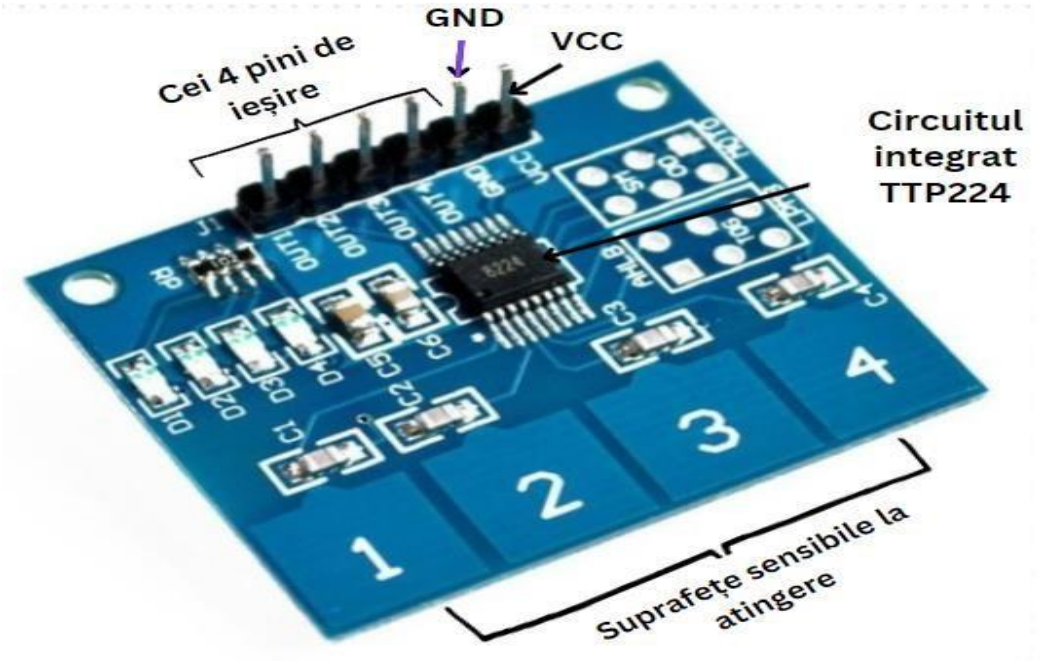

Figura 2.29 Senzorul capacitiv TTP224 și componetele sale

Circuitul integrat TTP224N-BSB este esențial în funcționarea senzorului și detectării tactile , fiind un dispozitivul specializat care include mai multe blocuri interne și permite detectarea capacității și generarea semnalelor de ieșire cu ajutorul celor 16 pini incorporați pe o parte și pe alta.

- Arhitectura internă: este una destul de complexă deoarece include multiple blocuri de logică digitală și circuite de detectare capacitivă , fiind proiectat pentru a oferi o performanță cât mai clară;
- Canalele de detectare tactilă: fiecare canal din cele patru poate fi configurat pentru a identifica atingerea;
- Logica capacitivă de detectare: diferite tehnici avansate sunt folosite de către circuit pentru a măsura și monitoriza capacitățile canalelor prin generarea unor semnale de testare;
- Generarea semnalelor de ieșire: acestea se generează pe baza capacității detectate și pot fi citite de către placa Arduino; Specificații tehnice:

- Tensiune de lucru: 2V – 5.5V;
- Dimensiuni: 35mm x 29mm;
- Timp de pornire: 0.8 – 1 secunde (în acest timp nu detectează atingerile);
- Rata de refresh a ieșirii: 55Hz la VDD=3V;
- Curent de funcționare: 100uA , maxim 160uA la VDD=3V;
- 64 pași de sensibilitate selectabili

Principiul de funcționare de bazează pe detectarea capacității și cuprinde mai mulți pași.Întâi trebuie să configurăm numărul și modul canalelor utilizate de senzor, iar asta se realizează prin conectarea pinilor la nivelurile corespunzătoare.Fiecare canale este echipat cu un electrod sensibil la atingere , astfel când o persoană sau un obiect atinge acest electrod , capacitatea canalului se schimbă și se creează un cămp electric în jurul lor.Urmează ca circuitul integrat să măsoare capacitătile de pe fiecare canal.O creștere sau o scădere a acesteia indică atingerea sau apropierea unui obiect.Rezultatele măsurătorilor sunt comparate cu pragurile setate pentru fiecare canal , iar daca acestea sunt depășite , înseamnă ca au fost furnizate semnale de ieșire pe pinii corespunzatori (OUT1,2,3,4) care sunt interpretate de către placa Arduino.

Conexiunea hardware la placa Arduino Uno:

- Pinul de alimentare: pinul VCC al senzorului TTP224 trebuie conectat la pinul de 5V al plăcii Arduino Uno;
- Pinul de masă: conectăm pinul GND al senzorului la unul din pinii GND al plăcii;
- Semnalele de ieșire: conectăm pinii OUT1 , OUT2 , OUT3 , OUT4 ai senzorului la pinii digitali ai plăcii pe care dorim să îi utilizăm;

Senzorul capacitiv de atingere are diverse utilizări în diferite domenii și aplicații.Voi prezenta câteva dintre ele:

- Jocuri interactive: poate fi implicat în jocuri ce implică atingeri , mișcări și se pot controla acțiunile din joc;
- Sisteme de securitate: poate detecta atingerile neautorizate , dar și în crearea codurilor de acces;
- Automatizarea casnică: poate permite controlul luminilor , aparatelor electrocasnice, ușilor, ferestrelor pentru un confort in locuință;

2.14 Modul releu 5V cu un canal comandat

Modulul releu are un canal comandat de 5V și este o componentă electronică utilizată în mare parte în proiectele ce utilizează plăcuțele Arduino , având ca scop controlarea și comutarea unui circuit sau dispozitiv de putere prin intermediul unui semnal de comandă cu sursa de 5V , în cazul meu paca Arduino Uno.Acesta are un circuit de comandă pe care îl conectăm la un pin digital de la placă , iar când pinul este la un nivel logic (HIGH sau LOW) , modulul activează și dezactivează releul.

Cu ajutorul acestui modul cu un singur canal comandat , putem controla diverse dispozitive (motoare , becuri , pompe) , dar mai ales senzori , fără a pune în pericol placa Arduino.

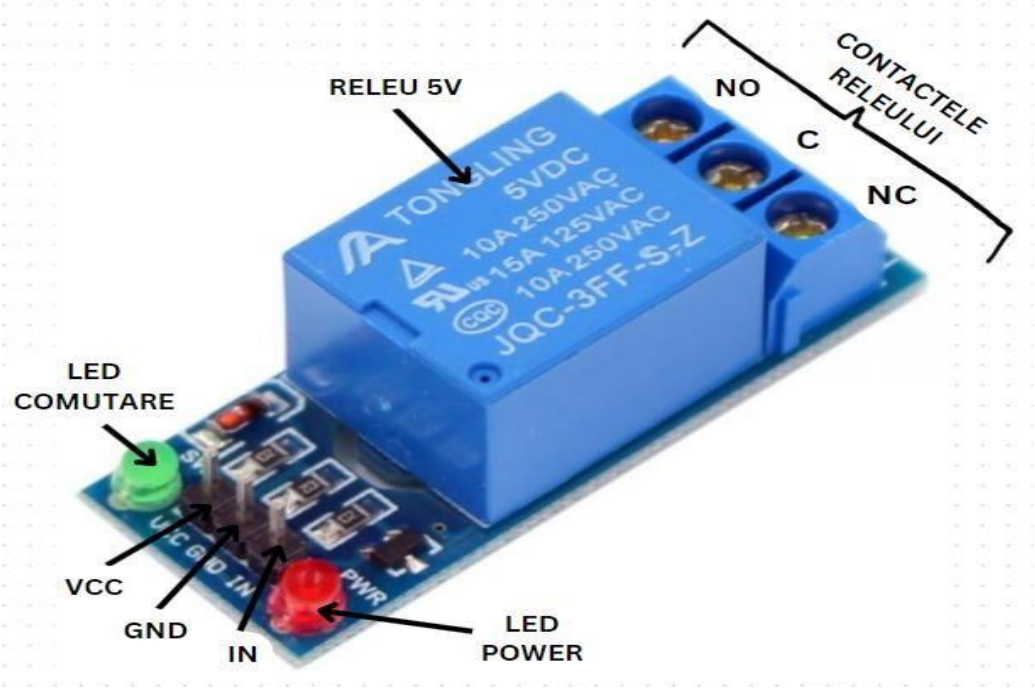

Figura 2.30 Componentele modulului releului de 5V cu canal comandat

Caracteristici tehnice:

- Sarcina: AC 250V/10A , DC 30V/10A;
- Tensiunea de operare: 5V;
- Consum curent: 80mA;
- Înâlțimea modulului: 20 mm;
- Dimensiuni PCB: 43 mm x 17 mm;
- Led-uri indicatoare de stare; Un releu cu un singur canal;

Modulul conține cei trei pini pe care îi conectăm la placa Arduino Uno , pinul GND la terminalul de referință al alimentării , pinul VCC de alimentare la tenisunea de 5V de la placă , iar pinul IN controlează starea de comutarea a releului prin aplicarea unui semnal logic menționat mai sus , fiind conectat la un pin digital.Acesta mai conține și cele două led-uri , cel cu roșu este ledul de alimentare (power) conectat la pinul de alimentare al modulului și cand este aprins atunci însemna ca releul este pregătit sa funcționeze.Led-ul verde , cel de comutare (Switching Signal) care se aprinde și se stinge în funcție de semnalul de comandă de pe pin.

Releul conține următoarele componente electronice:

- Bobina electromagnetică: generează câmpul magnetic necesar mecanismului de comutare al releului și este compusă dintr-o sârmă înfășurată în jurul unui nucleu din material feromagnetic;
- Armătura: componentă a releului care circulă într-o anumită direcție (când e activată bobina) prin intermediul câmpului magnetic generat de bobină;
- Contactul normal deschis (NO): este deschis doar atunci când releul nu este activat , iar când nu este activat , contactul se închide și permite trecerea curentului;
- Contactul normal închis (NC): este închis doar atunci când releul nu este activat , iar când nu este activat , contactul se deschide și blochează trecerea curentului;
- Contactul comutat (C): poate fi normal închis sau normal deschis;
- Arcurile de contact: ajută ca armătura să revină la poziția inițială atunci când bobina nu e activată și garantează conexiunile dintre contacte si armătură;
- Carcasa: releul este învelit cu o carcasă albastră care protejează componentele interne ale acestuia;

Sunt cunoscute două avantaje ale principiului de funcționare al releului cu un singur canal.În primul rând , curentul care activează releul este mai mic decât cel pe care contactele îl comută , asta însemnând că este foarte convenabil în aplicații în care avem comenzi simple și precise.În al doilea rând , cele trei contacte și bobina sunt izolate galvanic și nu exista nicio conexiune între ele , astfel că putem comuta curentul de rețea prin intermediul acestui sistem de izolare și putem reduce riscul deteriorărilor și interferențelor între cele două.

2.15 Alte componente electronice

Alte componente hardware pe care am ales să le folosesc în lucrarea mea sunt LED-urile de 5mm , rezistențele de 200 Ohmi , buzzere de 5V și bineînteles firele de tip mamă/tată.

LED-urile cu diametrul de 5mm sunt unele dintre cele mai comune componente electronice folosite , fiind gasite într-o gamă largă de culori (albastru , galben , roșu).Ele au o lungă durată de viață și consumă foarte puțină energie.
Led-ul de 5mm are doi pini prin intermediul cărora îl putem conecta la un circuit electronic, fiind important să respectăm polaritatea acestora deoarece dacă o inversăm , îl putem deteriora.

- Pinul anod: pinul pozitiv pe care îl conectăm pe un pin digital de la placa Arduino , este si mai lung decât celălalt pin;
- Pinul catod: este pinul negativ al LED-ului și este conectat ground-ul plăcii Arduino , este mai mic ca anodul;

Figura 2.31 LED-urile de 5mm Pentru o funcționare cât mai corectă , aceste leduri necesită un curent limitat pentru a le proteja de supracurent.De aceea, se folosesc rezistențe în serie pentru a asigura o alimentare corectă a LED-urilor.

Rezistențele cu o valoare nominală de 220 Ohmi oferă această valoare la trece curentului electric prin ele.Ele sunt deobicei legate în serie cu aceste tipuri de LED-uri pentru a le proteja și a le lungi durata de viață , fiind important să alegem rezistorii de valori adecvați în funcție de specificațiile pe care le avem la îndemână.

Ele au două terminale nemarcate , pe care le putem conecta unde vrem și nu se ține cont în care parte punem catodul și în care anodul.

Figura 2.32 Rezistențele de 220 Ohmi

Buzzerul de 5V este o componentă electronică care generează sunet în diferite aplicații , ca un fel de sonerie și funcționează la o tensiune de alimentare de 5V , de atât fiind necesar pentru a se produce sunetul în circuit.Pot fi de 2 tipuri:

- Buzzer piezoelectric: utilizează acest tip de efect piezoelectric , în care materialul lor generează o deformare sub acțiunea unui câmp electric , iar când se produce semnal electric, materialul piezoelectric produce sunet;
- Buzzer electromagnetic: este prezentă o bobină de sârmă în jurul unui miez feromagnetic , iar când trece curent prin bobină , se creează un câmp magnetic , generându-se sunet;

Pentru a utiliza un buzzer , trebuie să ținem cont că dispune de doi pini , unul pentru alimentarea de 5V la placa Arduino care este și mai înalt și unul pentru conectarea la masă care este mai scurt.

Figura 2.32 Buzzer de 5V

Firele utilizate sunt folosite pentru a lega conexiunile între două sau mai multe componete.Pot fi de diferite dimensiuni , culori și sunt de trei tipuri:

- Firele mamă-mamă: au la ambele capete conectori de tip mamă , adică au pini pe care ii putem introduce doar in conectorii de tip tată;
- Firele mamă-tată: au la un capăt un conector de tip mamă de care am menționat anterior și la un capăt un conector de tip tată care nu are pin și pe care ii putem introduce doar la conectorii de tip mamă;
- Firele tată-tată: au la ambele capete conectori de tip tată;

Figura 2.33 Fire mamă-tată

Capitolul 3

3. Prezentarea componentelor software

3.1 Software-ul Arduino IDE

Programul pe care l-am folosit pentru a programa plăcuța se numește Arduino IDE , versiunea 2.1.1 și poate fi folosit pentru programarea oricărei plăci Arduino , oferind un mediu de lucru simplu și accesibil.Este compatbil cu toate tipurile de sisteme de operare , precum Windows, Linux , Machintosh OSX.

El se poate descărca de pe site-ul oficial Arduino , dar se poate găsi și pe alte site-uri , fiind foarte ușor de descărcat și instalat.Programele scrise pentru plăcile Arduino se numesc “sketchuri”.Aceste sunt o variantă simplificată a limbajului de programare C/C++ deoarece se folosesc multe funcții specifile: if , else , for , while , break , continue , else , return , goto ,continue.

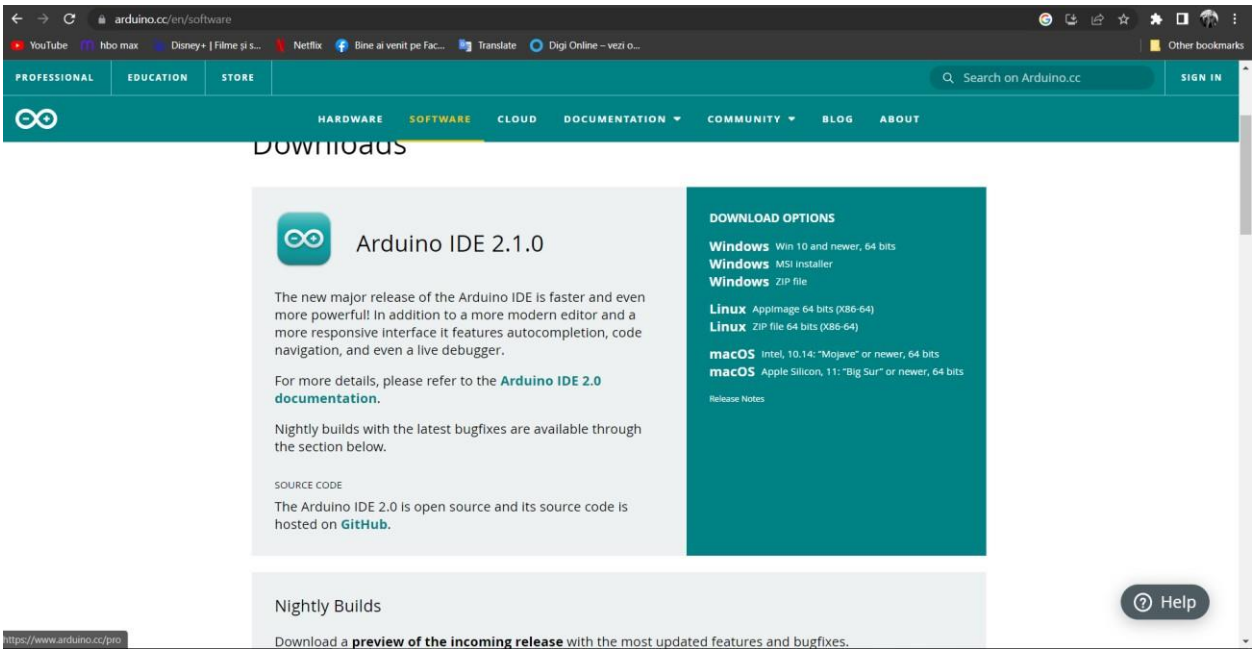

Figura 3.1 Site-ul oficial de descărcare al softului Arduino IDE

Un sketch este compus din două funcții principale , setup() și loop().Funcția setup() o executăm o singură dată când pornim placa Arduino și o folosim pentru a inițializa variabilele , pentru a configura pinii și a face alte setări pentru program.Funcția loop() conține codul principal al programului și se scrie mereu după funcția setup() , poate fi executată în mod repetat.

Totodată , pentru a programa codul în Arduino folosim o varietate de operatori pentru a efectua diferite operații , iar unele din cele mai utlizate sunt:

- Operatorii aritmetici: include operațiile matematice (adunare + , scădere - , înmulțire * , împărțire / , modul %);
- Operatorii de atribuire: atribuie valori unor variabile (= , += , -= , *= , /=);
- Operatii de comparație: aceștia compară două sau mai multe valori și returnează o valoare de adevăr (egal == , diferit != , mai mare > , mai mic <);
- Operatorii logici: efectuează operații logice (și && , sau || , nu !); Operatorii pe biți;

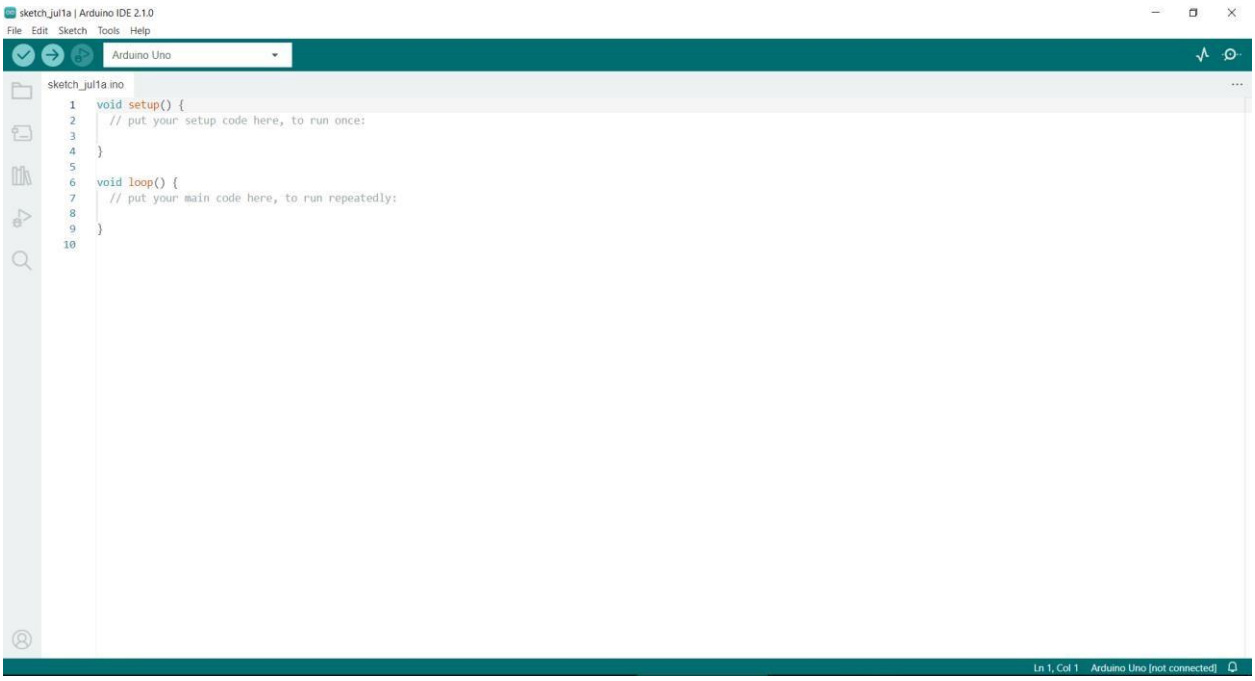

Figura 3.2 Ecranul de start al programului – Sketch-ul

Pentru ca programul să îmi funcționeze , iar cu ajutorul plăcii Arduio Uno să testez toți cei nouă senzori prezentați capitolul trecut , am avut nevoie sa intalez o serie de biblioteci , pe care leam găsit în meniul Library Manager din partea stângă a programului sau din meniul Sketch - Include Library - Manage Libraries din partea de sus a programului:

- Biblioteca <LiquidCrystal_I2C.h> este folosită pentru funcționarea ecranului LCD conectat la adaptorul I2C și furnizează o serie de funcții pentru configurarea afișajului,poziționarea cursorului și iluminarea ecranului;
- Biblioteca <Wire.h> este folosită pentru activarea comunicării I2C între dispozitive , ușurează recepționarea și transmiterea datelor între placă și ecran sau senzori.Acesta include funcții pentru a inițializa protocolul I2C,pentru a seta modul pe operare și controla adresa dispozitivelor;
- Biblioteca <DHT.h> o folosim când vrem sa testăm senzorul de temperatură și umiditate DHT11 pentru a simplifica procesul de citire a valorile rezultate de la senzor , oferind funcții pentru inițializarea pinului de conectare la senzor , citirea și furnizarea valorilor de la senzor;

Atât timp cat trebuie să lucrez cu placa Arduino Uno și nu cu alt tip de placă , trebuie ca în meniul Tools – Board – Arduino AVR Boards să selectez placa folosită , altfel programul nu va merge și totodată să selectez din meciul Tools – Portul în care am conectat placa Arduino la computer cu ajutorul cablului USB.Pe lângă aceste două chestii , mai trebuie să selectăm din meniul Tools –

Programmer – Arduino as ISP pentru a ne permite să utilizăm placa pentru a programa alte dispozitive , în cazul meu senzori.

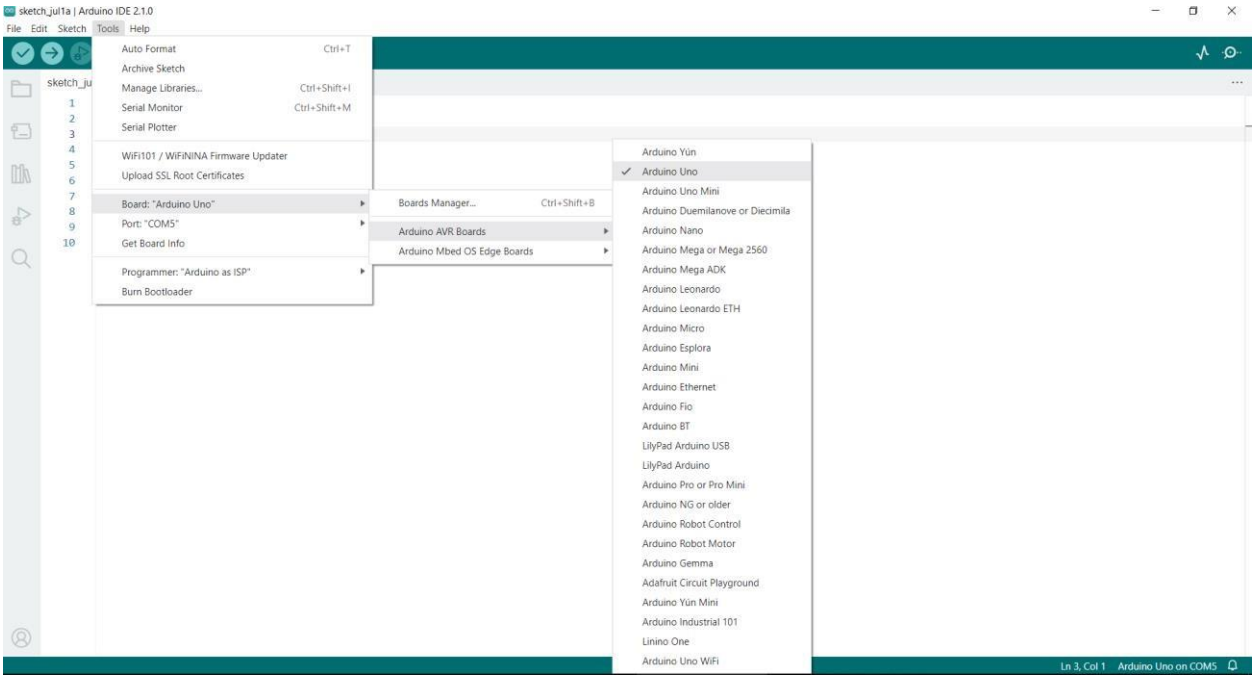

Figura 3.3 Meniul “Tools” din programul Arduino IDE

Din meniul File – Exemples putem rula coduri deja facute pentru a testa funcționalitatea dispozitivelor de care avem nevoie și a nu ne trezi cu surprize că unii senzori sau chiar placa sunt defecte.În figura ce urmează avem bara ce ne afișează tot ce putem testa , asta în funcție și de numărul de biblioteci instalate.Fiecare cod cuprinde , după am spus și mai sus , funcțiile setup() și loop().

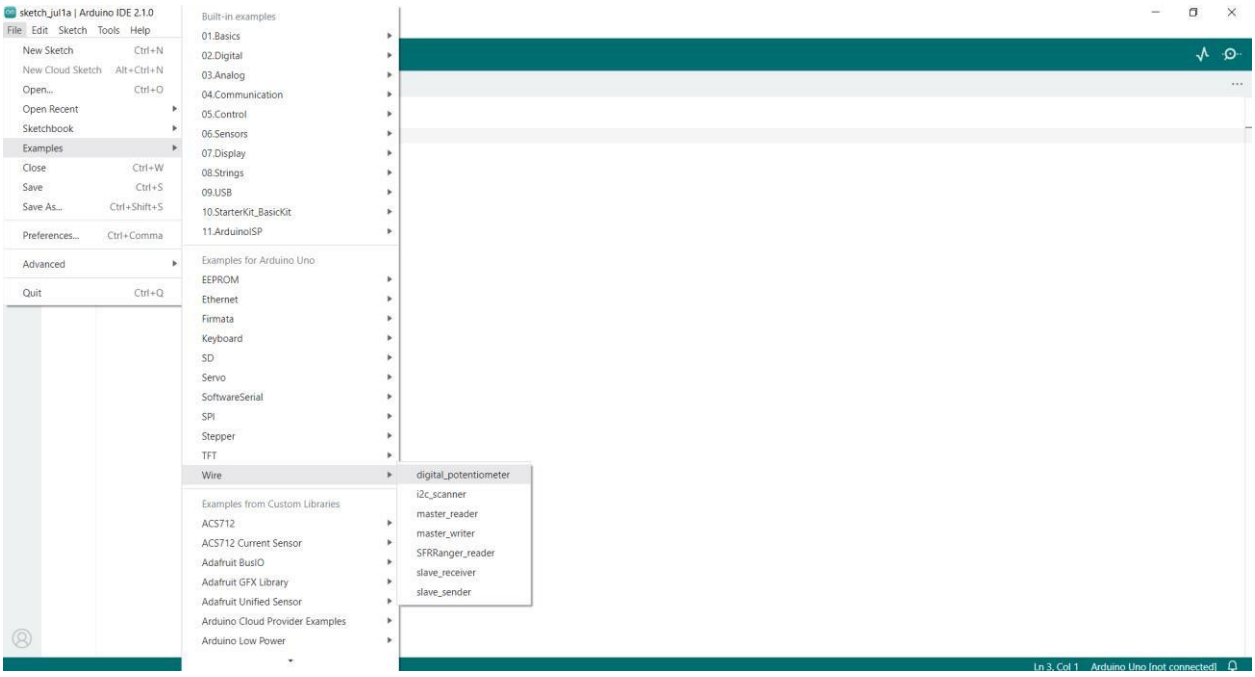

Figura 3.4 Meniul “File - Exemples” din programul Arduino IDE

Pentru a da un exemplu mai concret , din acest meniu de mai sus voi alege librăria Liquid Crystal I2C pe care am instalat-o , iar de acolo voi lua codul pentru ca pentru ecranul meu LCD 16x02 să imi afiseze un mesaj ales de mine.

#include <Wire.h> // biblioteca pentru funcționarea I2C

#include <LiquidCrystal_I2C.h> // biblioteca folosită pentru ecranele LCD

LiquidCrystal_I2C lcd(0x27, 16, 2); // Crearea unui element de tip LiquidCrystal_I2C cu adresa //I2C 0x27, 16 caractere pe linie și cu 2 linii. void setup()

{ lcd.init(); // Inițializarea LCD-

ului.

lcd.init(); // Inițializarea suplimentară a LCD-ului (este recomandată) lcd.backlight();

// Activarea luminilor afișajului LCD. lcd.setCursor(1, 0); // Setarea poziției cursorului

la linia 1, coloana 0 a afișajului LCD. lcd.print("Buna ziua!"); // Afișarea textului

"Buna ziua!" la poziția curentă a cursorului.

} void loop()

{

} // nu am adăugat nimic in funcția loop() deoarece nu a fost nevoie

Progamul Arduino IDE include câteva caracteristici principale , care îl fac un program foarte ușor de folosit mai ales de către studenții pasionați de electronică.

- Editarea codului: programul oferă un editor de cod care ajută la evitarea erorilor de sintaxă, iar pentru a obține un cod cât mai clar și mai corect , editorul oferă diverse funcții pentru a evidenția sintaxa , fiind de diferite culori (trebuie să scriem cuvintele cheie pentru a ne apărea în program funcția anume de care avem nevoie);
- Compilarea și uploadarea: Codul sursă scris de către utilizator este transformat într-un cod mașină înteles de către microcontroler-ul Arduino prin compilarea acestuia , verificânduse sintaxa codului și generându-se fișierul hex , care conține codul mașină menționat;
- Monitorul Serial: oferă o fereastră de comunicare între Arduino și calculator atunci când rulăm codul respectiv prin intermediul portului serial și permite vizualizarea datelor de afișare trasmise de Arduino la PC , mai exact monitorizează datele provenite de la senzori, iar în caz de erori suntem înștiințați prin intermediul acestui monitor;

Ca în orice program de acest tip , sunt prezente diverse tipuri de erori , care uneori sunt utile deoarece ne ajuta să învățam mai repede cum se folosește acest program.Le voi prezenta pe cele mai comune dintre ele:

- Erorile de sintaxa: apar atunci când uitâm un punct și virgulă la sfârșitul unei linii de cod, când folosim greșit ghilimelele sau parantezele , dar și când denumim greșit variabilele;
- Erorile de compilare: apar atunci când se compilează codul sursă iar unele variabile nu sunt definite cum trebuie , sau unele biblioteci nu sunt corect folosite;
- Erorile de încărcare: când procesul de încărcare a codului are probleme , cauza poate fi selectarea portului greșit , selectarea greșită a plăcii;
- Erorile logice: deși codul este corect , nu obținem rezultatele dorite din cauza unor greșeli în algoritmul de scriere și de aceea trebuie sa revedem din nou codul;
- Erorile de comunicare: apar când sunt probleme între placa Arduino și senzori , probleme ce pot fi cauza în principal de firele defecte;

Un ultim aspect important ce ține de acest program ar fi faptul că se pot folosi și alte tipuri de plăci hardware compatibile , nu doar Arduino , care se pot instala din meniul Tools – Board – Boards Manager:

- ESP8266: este o placă destul de puternică care are integrat un microcontroler cu conexiune Wi-Fi , folosindu-se în diverse proiecte care cuprind IoT; o ESP32: acesta are incoporat un microcontroler cu conexiune Wi-Fi și Bluetooth și se folosește cam în același domeniu IoT;
- Raspberry Pi: o placă destul de populară , la fel ca Arduino Uno , cu care se pot dezvolta proiecte cu o putere mare de calcul;
- BeagleBone: este similară cu Raspberry Pi , oferind o putere mare de calcul, iar când utilizează Arduino IDE avem nevoie de librăria Bonescript;

Capitolul 4

4. Prezentarea blocului de senzori și a machetei

Pentru o prezentare cât mai rapidă și eficientă , am ales să fac o machetă din material de polistiren , pe care l am învelit într-un carton de culoare mov , pentru a da o nuanță de culoare atrăgătoare.Pe acest material de formă pătratică am așezat pe partea centrală placa Arduino Uno împreună cu breadboardul , care conține majoritatea firelor de conexiune de la placă la breadboard și de la acesta la senzori , dat si celelalte componente electronice precum led-urile , rezistențele , și buzzerele pe care le-am folosit.

Ecranul LCD l-am amplasat pe partea din față a machetei , fiind unul dintre cele mai importante componente deoarece toți senzorii au valorile de ieșire afișate pe el.Am mai folosit o bucata de polistiren pe care am lipit-o în sus , astfel încât să îl pot prinde la un nivel mai înalt pentru a putea vizualiza valorile fiecarui senzor într-un mod frumos , iar sub el îi este scris numele împreună cu adaptorul I2C.

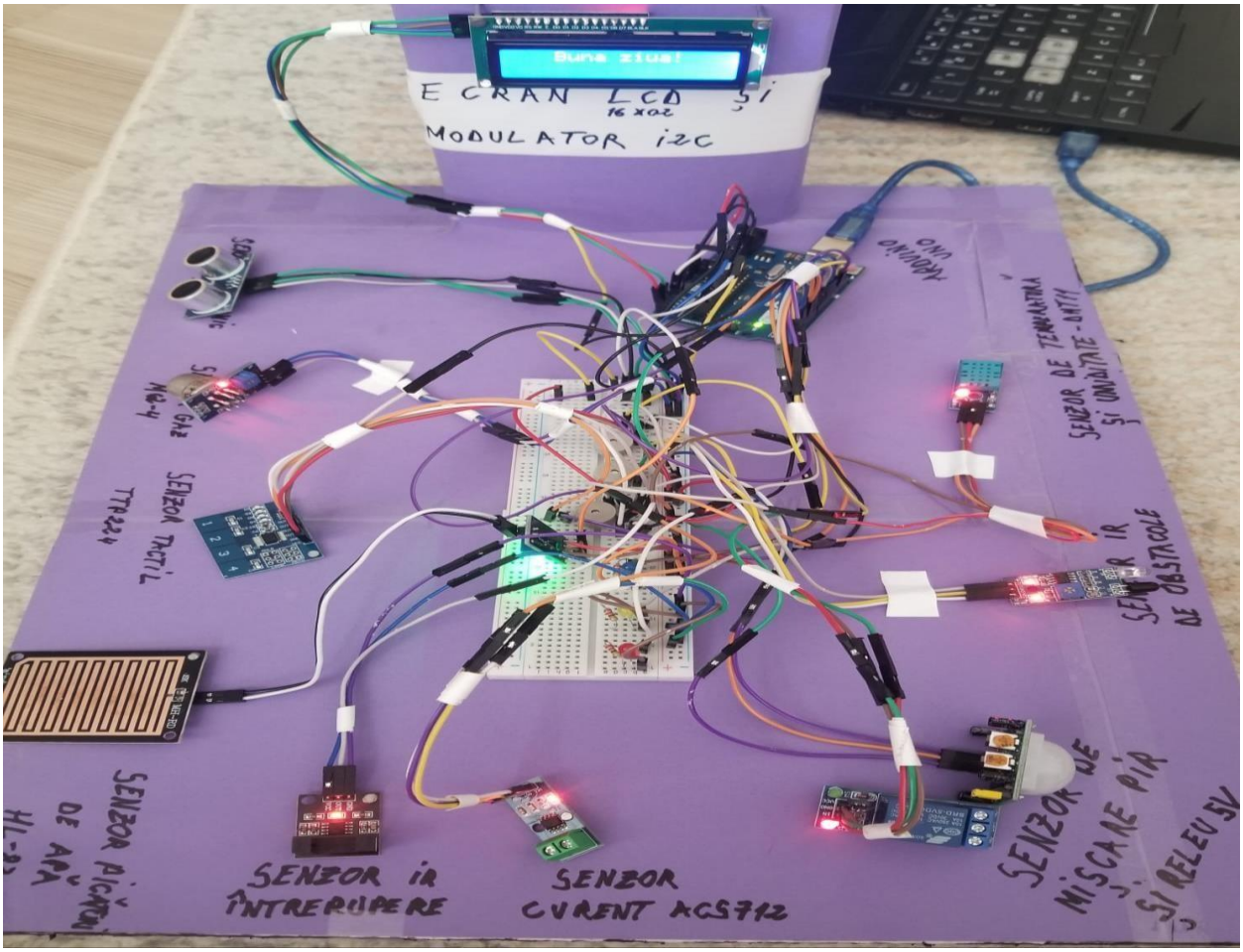

Figura 4.1 Macheta blocului de senzori

După cum se poate observa , voi testa un număr de nouă senzori pe care i-am amplasat pe câte o parte a machetei , astfel încât să păstrez o mica distanță între ei pentru a nu se produce interferențe sau alte erori , iar pe unii i-am lipit pentru a nu se mișca astfel că sunt mai ușor de folosit.

Primul senzor pe care îl voi testa este cel ultrasonic HC-SR04 , care măsoara distanța de la senzor și până la un obiect aflat în aria sa și afișează valoarea distanței pe ecran.Pentru a-l programa să facă asta , în primă fază am inclus cele doua librării necesare pentru comunicarea cu LCD-ul prin protocolul I2C (<Wire.h> și <LiquidCrystal_I2C.h>) , după care am inițializat ecranul cu adresa specifică (0x27) și am declarat pinii analogici și două variabile pentru distață și durată.În funcția setup() am inițializat ecranul și i-am pornit luminile cu funcțiile specifice , după care am configurat pinii trigger și echo.În funcția loop() am inclus încă o funcție care determină distanța prin formula: distanța=durata * 0.034/2 pentru a se afișa pe ecranul LCD.

Al doilea senzor este cel de gaz metan MQ-4 , care măsoara nivelul de gaz din încăpere.Am ales să îl programez ca atunci când nivelul de gaz este mai mare sau egal cu 12 , atunci să se declanșeze semnalul sonor de la buzzer și pe LCD să apară cuvântul pericol.Pentru asta , pe lângă includerea librăriilor și inițializarea LCD-ului , am declarat pinii și un procent , iar în funcția setup() am

configurat pinul pentru buzzer.În loop() am citit valoarea senzorului analogic conectat la pinul A0 , am calculat procentul de gaz și am impus condiția cu un if() ca procentul să fie mai mare sau egal cu 12 , atunci se declanșează buzzer-ul și un mesaj de pericol , altfel , un mesaj de stare normală.După ce am adăugat o întârziere de jumătate de secundă , informațiile pot fi afișate pe ecran.

Al treilea senzor este cel capacitiv TTP224 , cu atingere tactilă , pe care l-am codat astfel încât la apăsarea fiecărui buton dintre cele patru , pe ecran să îmi afișeze numărul respectiv.Am început cu includerea librăriilor și inițializarea ecranului , după care am declarat pinii digitali pentru fiecare canal al senzorului.În funcția setup() am configurat fiecare pin în parte și cu funcția lcd.print() am inclus un mesaj de apăsare a unui buton.În funcția loop() am verificat fiecare stare a canalelor cu if() și else if() , iar ca acest lucru sa fie corect , fiecare pin din cei patru trebuie să fie la nivel HIGH și sa afișeze o informație despre butonul respectiv.

Al patrulea senzor este cel de apă HL-83 , iar cu ajutorul unui buzzer , valoarea care începe de la 950-1000, când va scădea sub 400 la introducerea unei cantități de apă pe senzor , acesta va începe să sune și să afișeze un mesaj pe ecran.Pentru acest cod , a fost nevoie să includ librăria I2C și să declar constantele și variabilele pentru pinii senzorului și pentru buzzer , urmând ca în functia setup() să pornesc ecranul și să configurez pinii necesari cu funcția pinMode().În funcția loop() am citit valoarea senzorului prin intermediul pinului analog și am impus condiția de pornire a buzzer-ului cu un if() , de care am vorbit mai sus.

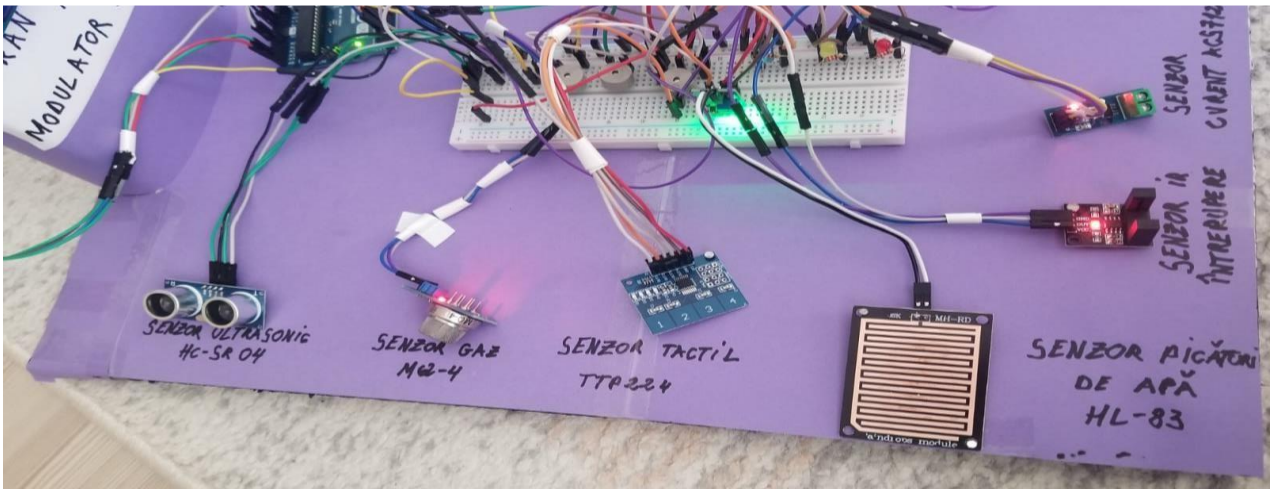

Figura 4.2 Primii patru senzori testați și amplasarea lor pe machetă

Al cincilea senzor este senzorul cu infraroșu de întreurpere , care detectează obiectele care sunt introduse în acel canal în formă de U pe care îl are.Astfel , am ales să afișez pe ecranul LCD un mesaj în care dacă senzorul identifică un obiect va apărea pe ecran mesajul DA , iar în caz contrar va afișa NU.Pentru început am inclus librăriile I2C și LCD și am inițializat ecranul cu adresa

specifică , urmând să declar o variabilă cate citește numărul pinului digital.În funcția setup() am configurat pinul senzorului , pornirea ecranului și afișarea textului de bază(Obiect introdus:).În funcția loop() este citită valoarea pinului digital , întrucât cu un if() verifică dacă este la nivel LOW sau nu.

Al șaselea senzor este cel care măsoară curentul , numit ACS712.Am programat ca acest senzor să măsoare valorile de curent a trei led-uri conectate în serie cu rezistențe de 220 Ohmi, astfel că în funcție de nivelurile de intensitate ale led-urilor , senzorul va afișa pe ecranul LCD valorile mai mari sau mai mici ale curentului.Pentru asta am inclus cele doua librării , am inițializat ecranul și am declarat varaibilele pentru pinul analogic , cele trei led-uri , o constantă pentru sensibilitate și una pentru tensiunea de referință.În setup() am pornit LCD-ul și am afișat un mesaj (Curent:) , pentru ca în funcția loop() să citesc tensiunea de pe pinul analogic , să convertesc tensiunea la volți , să calculez curentul în funcție de sensibilitate , urmând ca funcțiile analogWrite() și map() să scaleze valorile curentului în intervalul 0-255 , acesta fiind intervalul acceptat de luminozitatea LED-urilor.

Al șaptelea senzor este cel numit PIR HC-RS501 care detectează mișcarea în aria sa de acoperire.Pentru acest senzor , am ales să folosesc un releu de 5V , iar cu ajutorul lui , led-ul verde al acestuia să se activeze atunci când senzorul detectează mișcare , urmând ca un buzzer să fie activat , iar pe ecran să se afișeze mesajul “Mișcare detectată”.Pentru asta , am inclus cele două librării și am inițializat ecranul , dar am și declarat pinii pentru senzorul PIR , releu și buzzer.În funcția setup() , am inițiat LCD-ul și am configurat pinii pentru cele trei , urmând sa afișez textul de bază “Mișcare: ”.În loop() se citește valoarea de la senzor cu funcția digitalRead(PirPin) și se impune condiția că dacă acestă valoare este la nivel HIGH se afișează mesajul corespunzător și se activează buzzer-ul și releul.În sens contrar , acestea rămân dezactivate și pe ecran vom avea mesajul “Mișcare: nedetectată”.

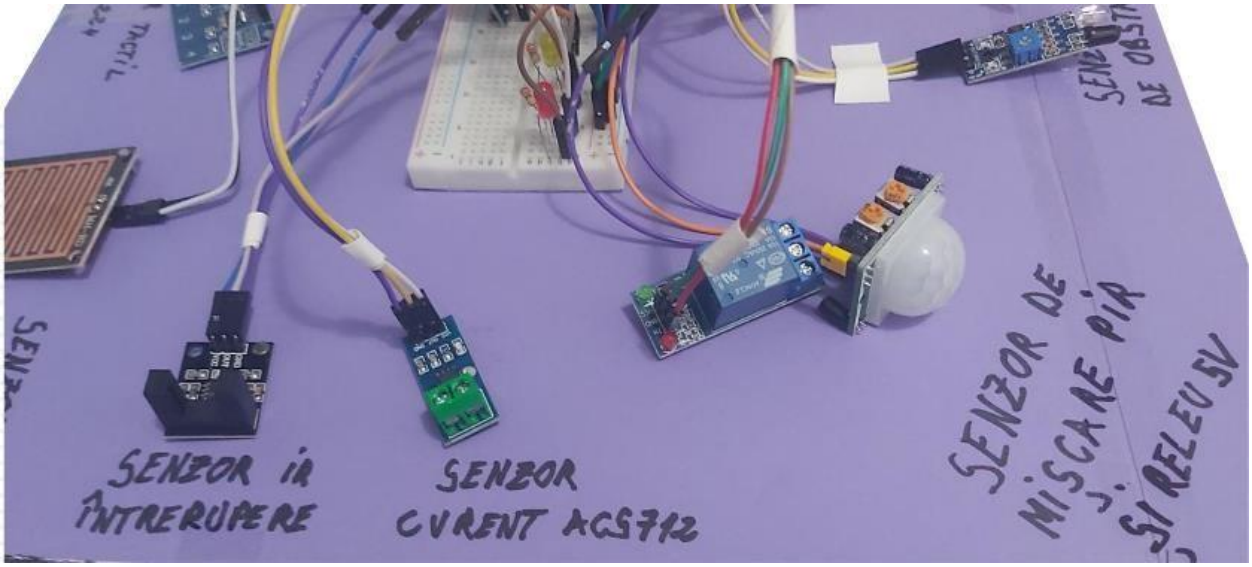

Figura 4.3 Următorii trei senzori testați și amplasarea lor pe machetă

Al optulea senzor testat este senzorul infraroșu de obstacole , care este foarte apropiat cu cel de întrerupere , doar ca acopera o arie mult mai mare și are inclus și un potențiometru de reglare.Pe acesta l-am programat la fel ca celălalt , să imi numere obstacolele pe care le identifică în aria sa.Pentru asta am declarat bibliotecile LCD-ului și al protocolului I2C , am inițializat ecranul , am declarat o constantă pentru numărul de obstacole cu 0 și am definit pinul senzorului.În funcția setup() am configurat pinul și am inițiat ecranul cu lcd.init() și luminile lui cu lcd.backlight().A urmat ca în funcția loop() să citesc valoarea digitală de la pinul senzorului și să pun condiția ca atunci valoarea citită este 0 , se incrementează numărul de obstacole , urmând a se afișa pe ecran cu lcd.print().

Ultimul senzor testat este senzorul DHT11 , care îmi afisează pe ecran temperatura și umiditatea din încăpere.Pentru a coda acest senzor , am inclus librăria pentru LCD <LiquidCrystal_I2C.h> și librăria <DHT.h> , folosită mereu pentru acest tip de senzor.A urmat să declar obiectul DHT , numărul pinului la care este conectat , tipul senzorului DHT și două variabile pentru temperatură și umiditate.Dupa aceea am inițiat ecranul , iar în funcția setup() am inițiat senzorul cu dht.begin() și am configurat ecranul LCD.În bulca loop(), am adăugat o pauză de 2 secunde cu delay(2000) , dupa care am citit valorile temperaturii și a umidității cu funcțiile dht.readTemperature() și dht.readHumidity().La afișarea pe ecran , am pus mesajele “Temp: ” și “Umd: ” pe câte o linie cu funcția lcd.setCursor().

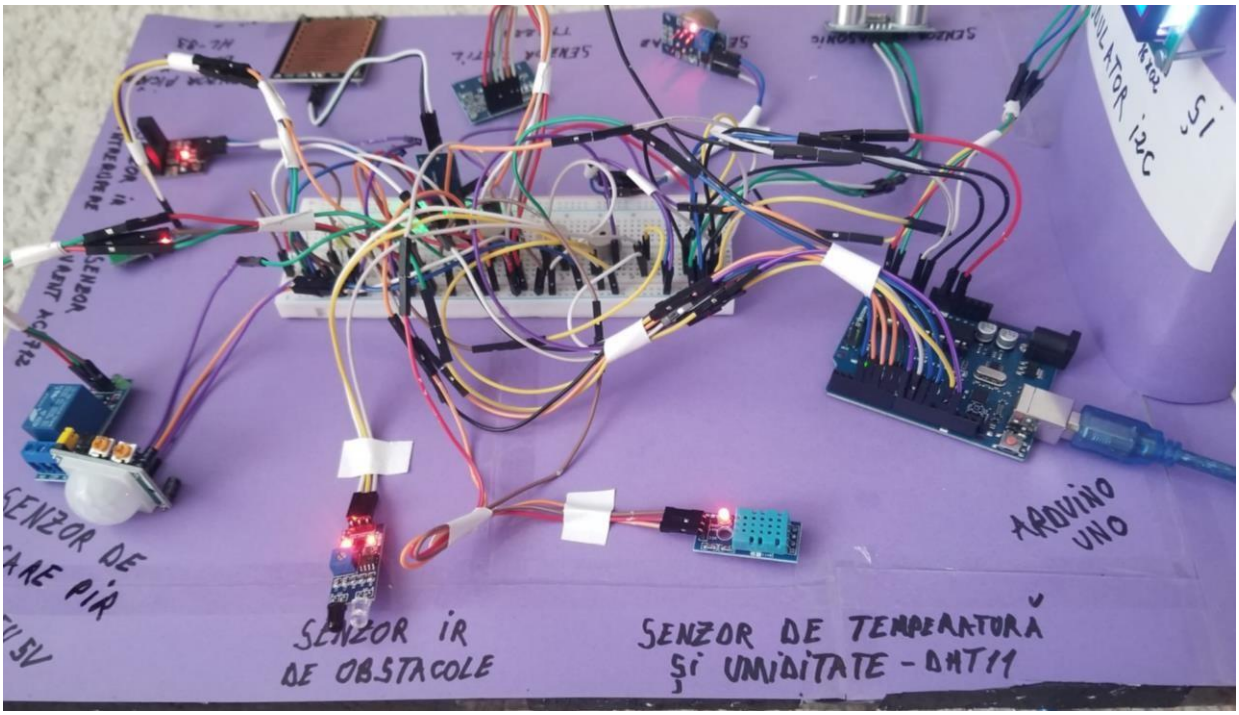

Figura 4.4 Ultimii doi senzori testați și amplasarea lor pe machetă

Capitolul 5

5. Concluzii

În concluzie , lucrarea mea de licență intitulată “Testarea inteligentă a unui bloc de senzori” a avut ca scop testarea eficientă și rapidă a acestor senzori relevanți și am urmărit obținea unor rezultate cât mai precise în ceea ce privește funcționarea și performanța senzorilor.

Consider că acest proiect ar putea fi considerat ca o bună resursă pentru studenții din anii mai mici sau cei care doresc să studieze la această facultate , astfel încât ii putem introduce în lumea tehnologiei și a senzorilor , iar ei să își dezvolte abilitățile practice , să-și stimuleze creativitatea.Implementarea a implicat atât componentele hardware prezentate în capitolul doi , cât și programul software Arduino IDE și algorimii de testare inteligentă prezentați la capitolul trei și patru.Din puctul meu de vedere , am pus la dispoziție majoritatea informațiilor de care un student are nevoie pentru a se familiariza cu aceste componente.

Deși lucrarea mea a avut succes în testarea acestor senzori , sunt unele îmbunătățiri care poti fi adăugate pe viitor și anume:

- Un număr mai mare de senzori;
- Utilizarea a două sau mai multe plăci Arduino , pe lângă cea Uno;
- Programarea mai complicată a senzorilor , care să includă mai multe componente și mai multe funcții în program;
- O machetă mai mare și mai complexă;
- Afișarea simultană pe mai multe ecrane LCD;

Consider ca în urma realizării acestui proiect , au reușit să învăț multe lucruri teoretice și practice legate de senzori și Arduino , deși la inceput am fost puțin neîncrezător în mine când miam ales această temă.Totodată , mi-am imbunătățit capacitatea de a programa în softul Arduino IDE și sper ca pe viitor , să mai am ocazia să fac asta și de ce nu să am un job in acest domeniu.

Capitolul 6

6.1 Bibliografie

[1] https://www.wago.com/ro/tehnologia-automatiz%C4%83rii

[2] https://ro.wikipedia.org/wiki/Arduino

[3] https://sites.google.com/site/arduinoelectronicasiprogramare/prima-lectie/5-introducere in-arduino

[4] https://cleste.ro/placa-de-dezvoltare-arduino-mega-2560.html

[5] https://docs.arduino.cc/retired/boards/arduino-yun

[6] https://store.arduino.cc/products/arduino-nano

[7] https://docs.arduino.cc/hardware/leonardo

[8] https://docs.arduino.cc/retired/boards/arduino-duemilanove

[9] https://store.arduino.cc/products/arduino-uno-rev3

[10] https://www.optimusdigital.ro/ro/placi-avr/1685-uno-r3-atmega328p-atmega16u2-placa de-dezvoltare-compatibila-cu-arduino.html

[11] https://components101.com/microcontrollers/arduino-uno

[12] https://robu.in/arduino-pin-configuration/

[13] https://ardushop.ro/ro/electronica/36-lcd-1602.html

[14] https://components101.com/displays/16x2-lcd-pinout-datasheet

[15] https://www.optimusdigital.ro/ro/adaptoare-i-convertoare/89-adaptor-i2c-pentru-lcd-1602.html

[16] https://ardushop.ro/ro/home/98-modul-i2c.html?gclid=CjwKCAjwp6CkBhB_EiwAlQVyxSG6E0ZR9MSCGC_y0GuaCmGoEUvvTZWphs4JhMF1qyImj6mm9-RFQRoCf5sQAvD_BwE

[17] https://inthou.pl/protokoly-komunikacyjne-i2c-teoria/

[18] https://openlabpro.com/guide/basics-of-i2c/

[19] https://blog.robofun.ro/2021/09/24/ce-este-un-breadboard-si-care-este-scopul-in-proiecte de-electronica/

[20] https://en.wikipedia.org/wiki/Breadboard#/media/File:Breadboard.png

[21] https://www.optimusdigital.ro/ro/senzori-senzori-optici/4514-senzor-infrarosu-de obstacole.html

[22] https://www.electronicshub.org/ir-sensor/

[23] https://ardushop.ro/ro/home/146-senzor-de-intrerupere-infrarosu.html

[24] https://cleste.ro/senzor-ultrasonic-hc-sr04.html

[25] https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc sr04/?utm_content=cmp-true

[26] https://cleste.ro/modul-senzor-detector-gaz-mq-4-dc5v-lpg.html

[27] https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-4.pdf

[28] https://ardushop.ro/ro/home/1044-cablu-usb-a-b-13m.html?gclid=CjwKCAjw-b-kBhB EiwA4fvKrDJTbrq56gr_21qg6WU3Fw3R6GDMzD1lcS0vxYlJtLrt9RlK0NhpA RoCdx gQAvD_BwE

[29] https://arduin.ro/products/modul-senzor-miscare-pir-hc-sr501

[30] https://ardushop.ro/ro/electronica/45-modul-pir-senzor-de-prezenta miscare.html?search_query=PIR&results=44

[31] https://lastminuteengineers.com/dht11-module-arduino-tutorial/

https://www.electronicshub.org/interfacing-acs712-current-sensor-with-arduino/

[32] https://www.kynix.com/Blog/Ceramic-capacitive-rain-sensor-avoids-false-

positives.html https://ardushop.ro/ro/home/1341-modul-senzor-atingere-

capacitiv-4-canale-ttp224.html

[33] https://ardushop.ro/ro/electronica/299-led-5mm.html?gclid=Cj0KCQjwnf-kBhCnARIsAFlg493XmmgeIl7n5JiwosDTryAn1kohKu_yoX4wArMISgUzIfxKIzDT2H saAhSZEALw_wcB

[34] https://www.sigmanortec.ro/Buzzer-pasiv-5v-p172425809?gclid=Cj0KCQjwnf-kBhCnARIsAFlg490A_Q0wkTqAyb9nPqC8nGAvj90Ct3NfgphzCPgjNe4do5H8EXF8C E0aAkgnEALw_wcB

[35] https://www.emag.ro/set-40-fire-de-conectare-tata-tata-10-cm-multicolor-dupont-40-10 mm/pd/D1G798MBM/

[36] https://www.arduino.cc/en/software

6.2 Anexă coduri Arduino IDE

1. Cod Arduino pentru senzorul ultrasonic HC-SR04:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // librăriile pentru comunicarea I2C și controlul ecranului LCD

LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);  // inițializează LCD-ul din clasa
                                //LiquidCrystal și specifică adresa I2C

int trigPin = A0;   int echoPin = A1; // se declară doua variabile cărora le sunt atribuite
valorile A0 și A1

int distanta; int durata; // se declară doua variabile întregi pentru a se stoca valorile distanței și
duratei

void setup()     // funcția de configurare
{
  lcd.init();                 // inițializarea LCD-ului
  lcd.backlight();          // activarea luminilor de fundal ale ecranului
pinMode(trigPin, OUTPUT); // setarea pinului de ieșire
pinMode(echoPin, INPUT);   // setarea pinului de intrare   delay(500);
// 500 de milisecunde pauză pentru stabilizare  }

void loop()      // funcția principală {
  DeterminareDistanta();         // se aplează această funcția care determină distanța   lcd.clear();
        // sterge conținutul de pe ecran și îl pregătește pentru noi informații

  lcd.setCursor(4,0);   // setează poziția cursorului de pe ecran , 4 e coloana și 0 linia
lcd.print("Distanta");  // afișează valoarea specificată pe LCD
```

```
  lcd.setCursor(6,1);    // setează poziția cursorului de pe ecran , 6 e coloana și 1
linia   lcd.print(distanta);  // afișează valoarea distanței pe LCD   lcd.print("cm");

  delay(200);
}

void DeterminareDistanta() // funcție pentru a măsura distanța {
  digitalWrite(trigPin, LOW);            // setează pinul trig la nivel LOW
  delayMicroseconds(2); // se așteaptă 2 microsecunde pentru a se stabili semnalul
digitalWrite(trigPin, HIGH); // setează pinul trig la nivel HIGH
  delayMicroseconds(10);                 // menține semnalul HIGH timp de 10 microsecunde
digitalWrite(trigPin, LOW);   // setează din nou semnalul la nivel LOW

  durata = pulseIn(echoPin, HIGH); // se măsoară durata semnalului la nivel HIGH
distanta = durata * 0.034 / 2;  // se calculează distanța prin formula dată }
```

2. Cod Arduino pentru senzorul de gaz metan MQ-4:

```
#include <LiquidCrystal_I2C.h>    // biblioteca necesară pentru afișajul ecranului LCD

LiquidCrystal_I2C lcd(0x27,16,2);   // inițializează LCD-ul din clasa
                                    //LiquidCrystal și specifică adresa I2C
int buzzer = 13; int
pin = A0;
int procent;              // definește pin conectat la buzzer , la senzor și o variabilă care stochează
                          //nivelul de gaz
void setup()
{

 lcd.init();              // inițializarea LCD-ului
 lcd.backlight();         // activarea luminilor de fundal ale ecranului
 Serial.begin(9600);    // inițializază comunicarea serială către monitorul serial
pinMode(buzzer, OUTPUT);            // setează pinul buzzer ca ieșire
delay(2000);              // se face o pauza de 2 secunde
}

void loop()
{
  int analogSensor = analogRead(pin);          // stochează in variabila analogSenzor valoarea de
                                               //pe pinul senzorului cu funcția analogRead()
```

```
  int procent=(analogSensor-50)/35;              // calculează nivelul de gaz , scăzându-se 50 pentru
                                                 //offset si se împarte la 50 pentru a se converti în procent

  lcd.setCursor(0,0);
lcd.print("Nivel Gaz:");
lcd.setCursor(10,0);
lcd.print(procent);
lcd.setCursor(12,0);
  lcd.print("%");
 if (procent >= 12)      // verifică daca nivelul de gaz este mai mare sau egal ca 12
    lcd.setCursor(0,1);
    lcd.print("PERICOL");       // pe a doua linie se afișează mesajul PERICOL
    tone(buzzer, 1000, 10000);  // generează sunetul buzzer-ului cu o frecvență de 1000Hz pe o
                                //durată de 10 secunde
  }
else
{
  lcd.setCursor(0,1);
  lcd.print("NORMAL");        // pe a doua linie se afișează mesajul NORMAL

  noTone(buzzer);
  }
  delay(500); // se face o pauză de jumătate de secundă    lcd.clear();
          // se sterge conținutul de pe ecran
}
```

3. Codul Arduino pentru senzorul capacitiv TTP224:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // librăriile pentru comunicarea I2C și controlul ecranului LCD

LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2); // inițializează LCD-ul din clasa
                                                        //LiquidCrystal și specifică adresa I2C

const int Pin1senzor = 4;  const int Pin2Senzor = 5;  const int Pin3Senzor = 6;  // se definesc
pinii senzorului de atingere și se folosește const int
 //deoarece pinii trebuie să rămână neschimbați,iar valoarea lor să nu fie modificată  const int
Pin4Senzor = 7;

void setup()
```

```
{
lcd.init();
  lcd.backlight();

  pinMode(Pin1senzor, INPUT);
pinMode(Pin2senzor, INPUT);   pinMode(Pin3senzor,
INPUT);
  pinMode(Pin4senzor, INPUT);        // se setează pinii senzorului ca intrări

  lcd.print("Apasa un buton!");
}
void loop()
{
  if (digitalRead(Pin1senzor) == HIGH)        // se verifica daca pinul 1 este activat
  {
    lcd.setCursor(0, 1);
    lcd.print("Butonul: 1");      // afișează textul “Butonul: 1”
  }
  else if (digitalRead(Pin2senzor) == HIGH) // se verifica daca pinul 2 este activat
  {
    lcd.setCursor(0, 1);
    lcd.print("Butonul: 2");      // afișează textul “Butonul: 2”pe a doua linie
  }
  else if (digitalRead(Pin3senzor) == HIGH) // se verifica daca pinul 3 este activat
  {
    lcd.setCursor(0, 1);
    lcd.print("Butonul: 3");      // afișează textul “Butonul: 3”pe a doua linie
  }
  else if (digitalRead(Pin4senzor) == HIGH) // se verifica daca pinul 4 este activat
  {
    lcd.setCursor(0, 1);
    lcd.print("Butonul: 4");      // afișează textul “Butonul: 4”pe a doua linie
  }

  delay(100); // se așteaptă 100 milisecunde pentru a evita citirile rapide }
```

4. Codul Arduino al senzorului picături de ploaie Hl-83:

```
#include <LiquidCrystal_I2C.h>        // biblioteca necesară pentru afișajul ecranului LCD
```

```
const int senzorPin = A0;          // constantă inițializată cu A0 – pinul analogic pentru citirea
                                                                                    //senzorului
const int buzzerPin = 3;         // constantă inițializată cu 3 – pinul digital pentru buzzer

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
 {
  pinMode(senzorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);   // setează pinul senzorului ca intrare și al buzzer-ului ca ieșire

  lcd.init();
  lcd.backlight();          // inițializăm LCD-ul
  lcd.setCursor(0, 0);
 lcd.print("Valoare: ");
}

void loop()
 {
  int valsenzor = analogRead(senzorPin);      //variabila valsenzor citește valoarea analogica de la
                                                                                    //pinul senzorului
  Serial.println(senzorPin);      // se afișează valoarea și pe monitorul serial

  lcd.setCursor(9, 0);
  lcd.print(valsenzor);

  if (valsenzor < 400)  // verifică dacă valoarea citită de la senzor este mai mica ca 400
{
    digitalWrite(buzzerPin,    HIGH);                    //    activare    buzzer
lcd.setCursor(0, 1);
    lcd.print("Sub 400!      ");  // afișează textul “Sub 400!” pe a doua linie
    delay(1000);                    // se așteaptă o secundă
digitalWrite(buzzerPin, LOW);      // se dezactivează buzzer-ul
lcd.setCursor(0, 1);
    lcd.print("            ");          delay(1000);
  }
}
```

5. Cod Arduino pentru senzorul IR de întrerupere:

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

```
#include <LiquidCrystal_I2C.h>  //librăriile pentru comunicarea I2C și controlul ecranului LCD

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int pinsenzor = 2;  // declar o constantă de tip întreg căreia îi atribui valoarea 2 a pinului

void setup()
{
  pinMode(pinsenzor, INPUT_PULLUP); // setez modul pinului ca fiind intrare cu rezistență de
                                                          //pull-up activată

  lcd.init();
  lcd.backlight(); // inițializez și activez luminile ecranului LCD   lcd.setCursor(0, 0);
  lcd.print("Obiect introdus:");  // setează cursorul la începutul primului rând și afișează acel
                                                                                          //mesaj
}
void loop()
{
  int valoaresenzor = digitalRead(pinsenzor);  //citește valoarea digitală de la pin și o atribuie
                                                          //variabilei
lcd.setCursor(0, 1);
  if (valoaresenzor == LOW)  // verifică dacă obiectul este introdus
  {
    lcd.print("Nu ");
  }
else
{
    lcd.print("Da ");
  }

  delay(100); // pauză de 100 milisecunde pentru a preveni afișarea rapidă }
```

6. Codul Arduino pentru senzorul de curent ACS712:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // librăriile pentru comunicarea I2C și controlul ecranului LCD

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
const int analogPin = A0;  // constantă de tip int care păstrează numărul pinului analogic const float
senzitivitate = 0.185; // constantă de tip float care stochează sensibilitatea senzorului
const float tensiunereferinta = 5.0; // constantă de tip float care stochează tensiunea de referință
                                        //pentru conversia tensiunii

const int ledPin1 = 9;  const int ledPin2 = 10;  const int ledPin3 = 11; // se
stochează numărul pinilor pentru cele trei led-uri

void setup()
{
  pinMode(ledPin1,                    OUTPUT);
pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);   // se setează modul pinilor led ca ieșiri

  lcd.init();
lcd.backlight();
  lcd.print("Curent: ");
}
void loop()
 {
  int valsenzor = analogRead(analogPin); // se citește valoarea tensiunii de pe pinul analogic   float
voltage = valsenzor * tensiunereferinta / 1023.0;   // se calculează valoarea tensiunii,
          //utilizând valoarea citită , tensiunea de referință și rezoluția ADC de 1023
  float curent = voltage / senzitivitate; // se calculează valoarea curentului cu ajutorul tensiunii și
                                            //a sensibilității

  lcd.setCursor(8, 0);
  lcd.print(curent/70, 3);          // afișăm valoarea curentului cuprins între 10mA și 20mA


  analogWrite(ledPin1, map(curent, 0, 5, 0, 255));
analogWrite(ledPin2, map(curent, 0, 5, 0, 255));   analogWrite(ledPin3,
map(curent, 0, 5, 0, 255));
 // se setează luminozitatea led-urilor în funcție de curent cu funcția map() pentru a mapa
//intervalul curentului 0-5A și al luminozității 0-255

  delay(1000);  // se așteaptă o secundă pentru a se citi din nou
}
```

7. Codul Arduino al senzorului de mișcare PIR:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>  // librăriile pentru comunicarea I2C și controlul ecranului LCD


LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27, 16, 2);

int pinsenzor = 6;  int
pinreleu = 8;
int pinbuzzer = 5;       //variabile care stochează numărul pinilor senzorului , al releului și al
                         //buzzer-ului

void setup()
{   lcd.init();
lcd.backlight();

  pinMode(pinsenzor, INPUT);          // setează modul pinului ca fiind intrare
pinMode(pinreleu, OUTPUT);          // setează modul releului ca fiind ieșire
pinMode(pinbuzzer, OUTPUT);         // setează modul buzzer-ului ca fiind ieșire

 digitalWrite(pinreleu, LOW);
digitalWrite(pinbuzzer, LOW);       // se setează starea inițială la nivel LOW a releului și
                                    //buzzer-ului

  lcd.print("Miscare:");
}

void loop()
{
  int pirValuare = digitalRead(pinsenzor); // se citește valoarea digitală de la pinul senzorului și
                                           //se stochează în variabila pirValuare

  if (pirValuare == HIGH)       // se verifică dacă se detectează mișcare
  {
    lcd.setCursor(0,                        1);
lcd.print("Detectata!");
    digitalWrite(pinreleu, HIGH);       //          se          activează          releul
digitalWrite(pinbuzzer, HIGH);          // se activează buzzer-ul
  }
else
{
   lcd.setCursor(0, 1);     lcd.print("Nedetectata");     // se afișează
mesajul “Nedetectata”     digitalWrite(pinreleu, LOW);        // se
```

```
dezactivează releul     digitalWrite(pinbuzzer, LOW);          // se
dezactivează buzzer-ul
  } }
```

8. Codul Arduino al senzorului IR de obstacole:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>  // librăriile pentru comunicarea I2C și controlul ecranului LCD

#define IR_SENSOR_PIN 12
#define LCD_ADDRESS 0x27  // se definește pinul senzorului și adresa ecranului LCD

LiquidCrystal_I2C lcd(LCD_ADDRESS, 16, 2);  // inițializează LCD-ul din clasa
                                            //LiquidCrystal și specifică adresa I2C
int nrobstacole = 0;     // declar variabila nrobstacole cu 0 bool detectie = false;   // bool indică o
stare , variabila detectie este false pentru că la început nu
este                                 //detectat nimic

void setup() {
  pinMode(IR_SENSOR_PIN, INPUT); // configurez pinul senzorului ca pin de intrare

 lcd.init();
lcd.backlight();          // inițializez și activez luminile ecranului LCD
 lcd.print("Obiecte: ");
}

void loop() {
  if (digitalRead(IR_SENSOR_PIN) == HIGH)//dacă nu se detectează obiect - detectia falsă
{
    detectie = false;
  }

  if (digitalRead(IR_SENSOR_PIN) == LOW && !detectie)
{
    nrobstacole++;
    detectie = true;        // dacă s-a detectat obiect atunci crește numărul de obstacole

    lcd.setCursor(9, 0);
    lcd.print("                ");   //   Șterge    numărul    anterior
lcd.setCursor(9, 0);
    lcd.print(nrobstacole);        // afișăm pe ecran numărul de obstacole
```

```
  }

  delay(200); }
```

9. Codul Arduino al senzorului de temeperatură și umiditate DHT11:

```
#include <DHT.h> // includem librăria DHT carre permite citirea valorilor de temperatură și
                                                                    //umiditate
#include<LiquidCrystal_I2C.h> // includem biblioteca LiquidCrystal pentru funcționarea
                                    //ecranului LCD și a protocolului I2C

DHT dht(A2 , DHT11);    // se citesc valorile de la senzorul DHT11 , conectat la pinul A2 int
temperatura;
int umiditate; // variabile de tip intreg care stochează valorile temeperaturii și umidității

LiquidCrystal_I2C lcd(0x27 , 16 , 2); // Adresa LCD 0x27 , 16 coloane și 2 linii

void setup()
{
  dht.begin(); // se inițializează comunicarea cu senzorul;

  lcd.init();       // inițializăm LCD-ul
  lcd.backlight(); // pornim luminile de fundal ale ecranului

}

void loop()
{
  delay(2000); // se așteptă 2 secunde înainte de a citi valorile senzorului   temperatura
= dht.readTemperature();
  umiditate = dht.readHumidity(); // cele două funcții citesc valorile temperaturii și umidității de
                                    //la senzor și le stochează in variabilele cu același nume

  lcd.setCursor(0 , 0);   lcd.print("Temp: ");    lcd.print(temperatura);   lcd.print(" C");       // la
inceputul primei linii avem mesajul “Temp:” , iar după el se afișează
                              //valoarea temperaturii , urmat de C

  lcd.setCursor(0 , 1);   lcd.print("Umd: ");   lcd.print(umiditate);   lcd.print(" %");  // la
inceputul celei de a doua linii avem mesajul “Umd:” , iar după el se
```

```
                                    //afișează valoarea umidității , urmat de procentul %
}
```

6.3 Listă abrevieri

LCD – Liquid Crystal Display;

LED – Light Emitting Diode;

I/O – Input/Output;

PWM – Pulse Width Modulation;

USB – Univer Serial Bus;

TTL – Tranzition Transition Logic;

IDE – Integrated Development Environment;

ICSP – In Circuit Serial Programming;

VCC – Voltage Common Collector;

GND – Ground;

I2C – Inter Integrated Circuit;

SDA – Serial Data Line;

SCL – Serial Data Clock;

IR – Infrared;

DC – Direct Current;