

AUTOMATIZARE

GEAM

Proiect C.N:

Automatizarea unui geam cu mai multe funcționalități, folosind Arduino Uno

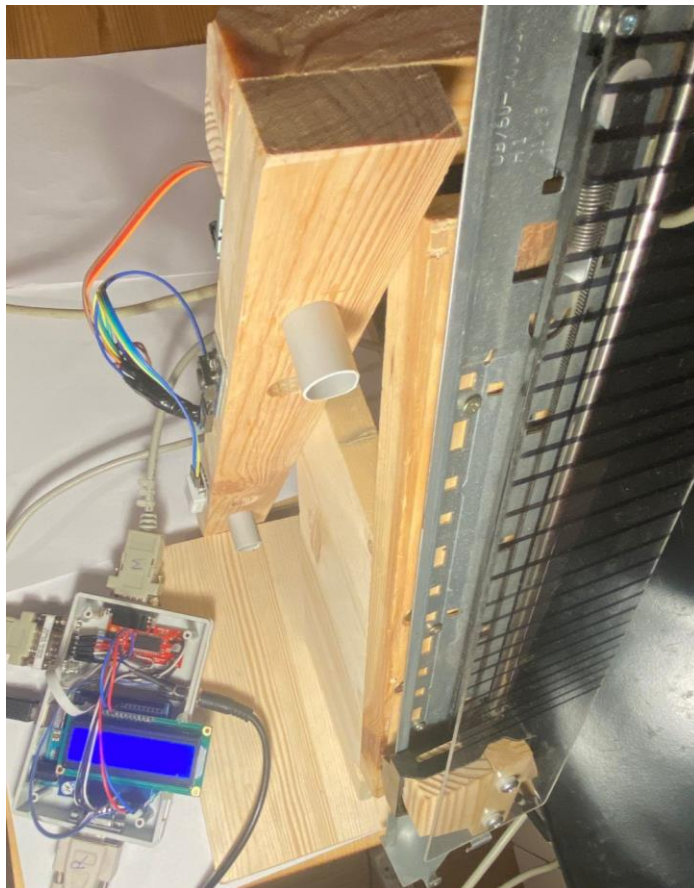
Bonea Mara
Boțoc Indi
2C.01.2

Cuprins

Descriere generală.....	4
Descriere funcționalității	5
Descrierea componentelor.....	7
Motorul pas cu pas	7
Driverul	7
Senzorii de prezență.....	9
Senzorul de ploaie	9
Senzorul de temperatură și umiditate	10
Ecranul LCD.....	11
Butonul	12
Placa Arduino.....	12
Despre microcontrolerul Arduino Uno	12
Conectarea componentelor în placa Arduino	14
Descrierea codului	16
Diagrama logică	16
Explicații.....	16
Observații și posibile îmbunătățiri.....	17
Resurse și bibliografie.....	18

Descriere generală

Proiectul de față prezintă automatizarea unui geam, a cărui sistem de închidere este realizat printr-un mecanism de tip glisare, adică la glisarea mâinii de jos în sus geamul se închide, iar la mișcarea inversă a mâinii acesta se deschide. De asemenea, în proiect am incorporat un mecanism de detectare a ploii, iar ca reacție, geamul se închide și se înregistrează numărul de picuri de ploaie, care este afișat pe un ecran sub nivele de intensitate a ploii. Pe acest ecran, sunt afișate și temperatura și umiditatea curenta, dar și direcția de mișcare geamului în cazul închiderii/deschiderii.

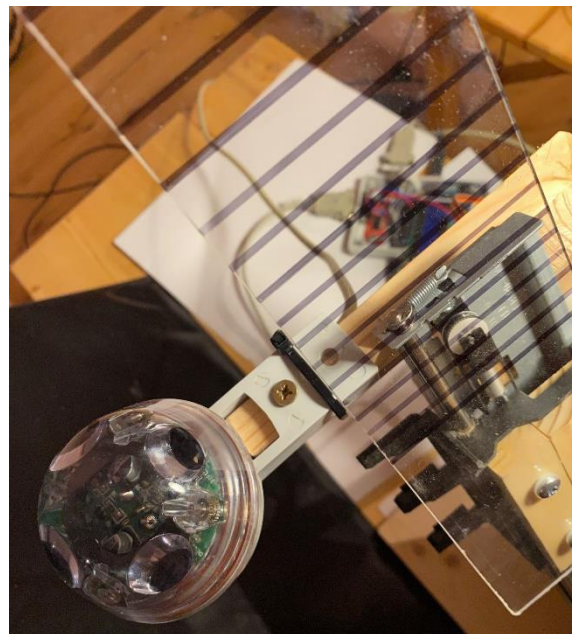
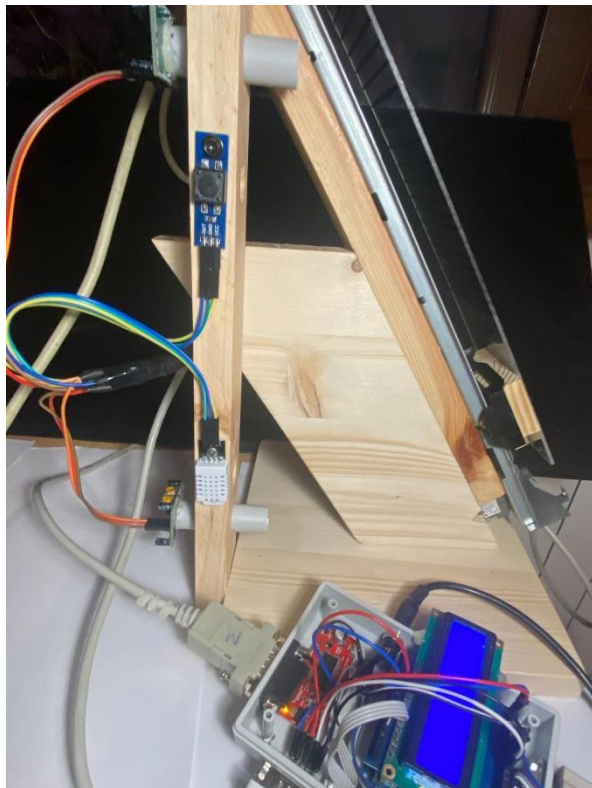
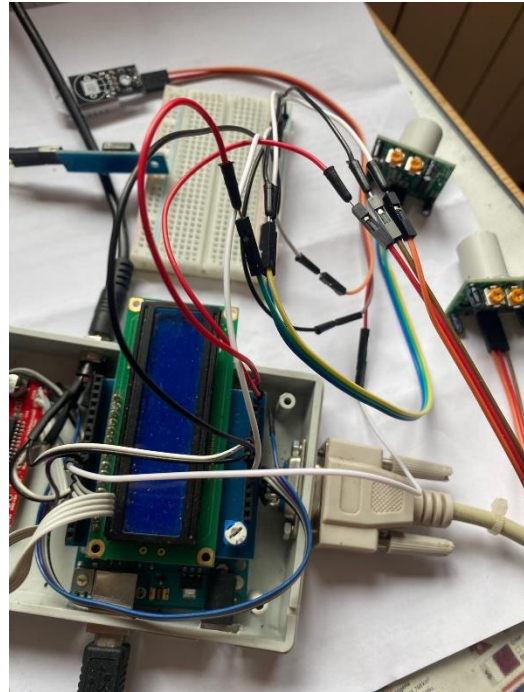
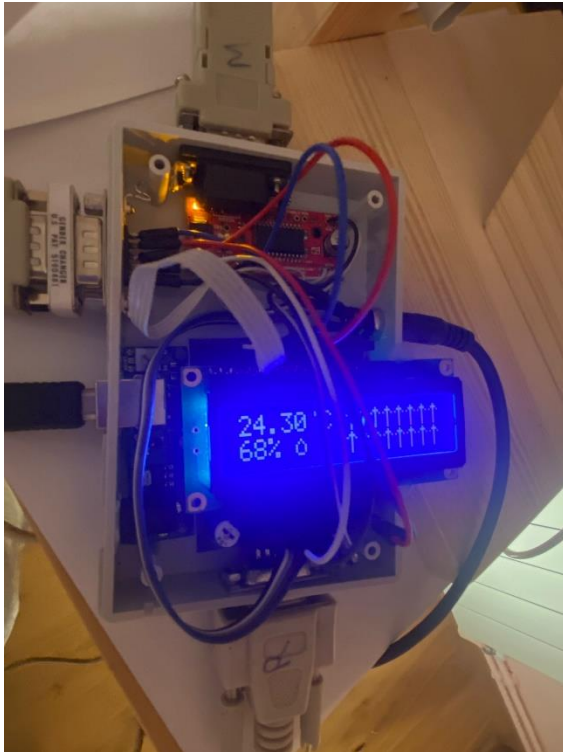


Descriere funcționalității

Dintre toate funcționalitățile ce le îndeplinește proiectul prezentat mai sus, pe scurt, cea mai de bază și cea mai importantă ar fi cea de închidere a geamului prin comanzi date prin mișcarea mâinii. Pentru a înregistra dacă mișcarea mâinii este făcută de jos în sus, sau invers, am folosit doi senzori de prezență ce receptează succesiv dacă apare un obstacol în câmpul lor de vizibilitate. Semnalul primit prin cei doi senzori activează motorușul, care este legat la un mecanism de culisare cu curea, componenta refolosită dintr-o imprimantă. Pe acest mecanism de culisare, este atașat geamul, reprezentat de o bucată de plexiglass. Deoarece mecanismul de culisare are o cursă scurtă, geamul nu se ridică foarte mult, astfel că, un cadran ar fi fost inutil. Deși ideea de baza a automatizării a fost pentru un geam de mașină, sistemul de funcționare a fost adaptat pentru componentele folosite, astfel că s-a îndepărtat de adevărata inginerie din spatele geamului unui autoturism.

O alta funcționalitate importantă ar fi cea de închidere a geamului la sesizarea ploii. Acest lucru este posibil prin intermediul unui senzor de ploaie, de mare sensibilitate, ce poate număra picuri de ploaie. De asemenea, nivelul de ploaie este afișat pe un ecran LCD sub următoarea semnificație: între 1 și 5 picuri este ploaie superficială, între 5 și 10 picuri este ploaie cu intensitate medie, iar peste 10 picuri se consideră că ploaia este densă.

Pe ecranul LCD, sunt afișate și temperatura și umiditatea, informații ce se obțin printr-un senzor DHT, dar și săgeți ce indică direcția de mișcare a geamului, la activarea prin comanda cu mâna sau la începerea ploii. De asemenea, întregul sistem este pornit și oprit printr-un buton.



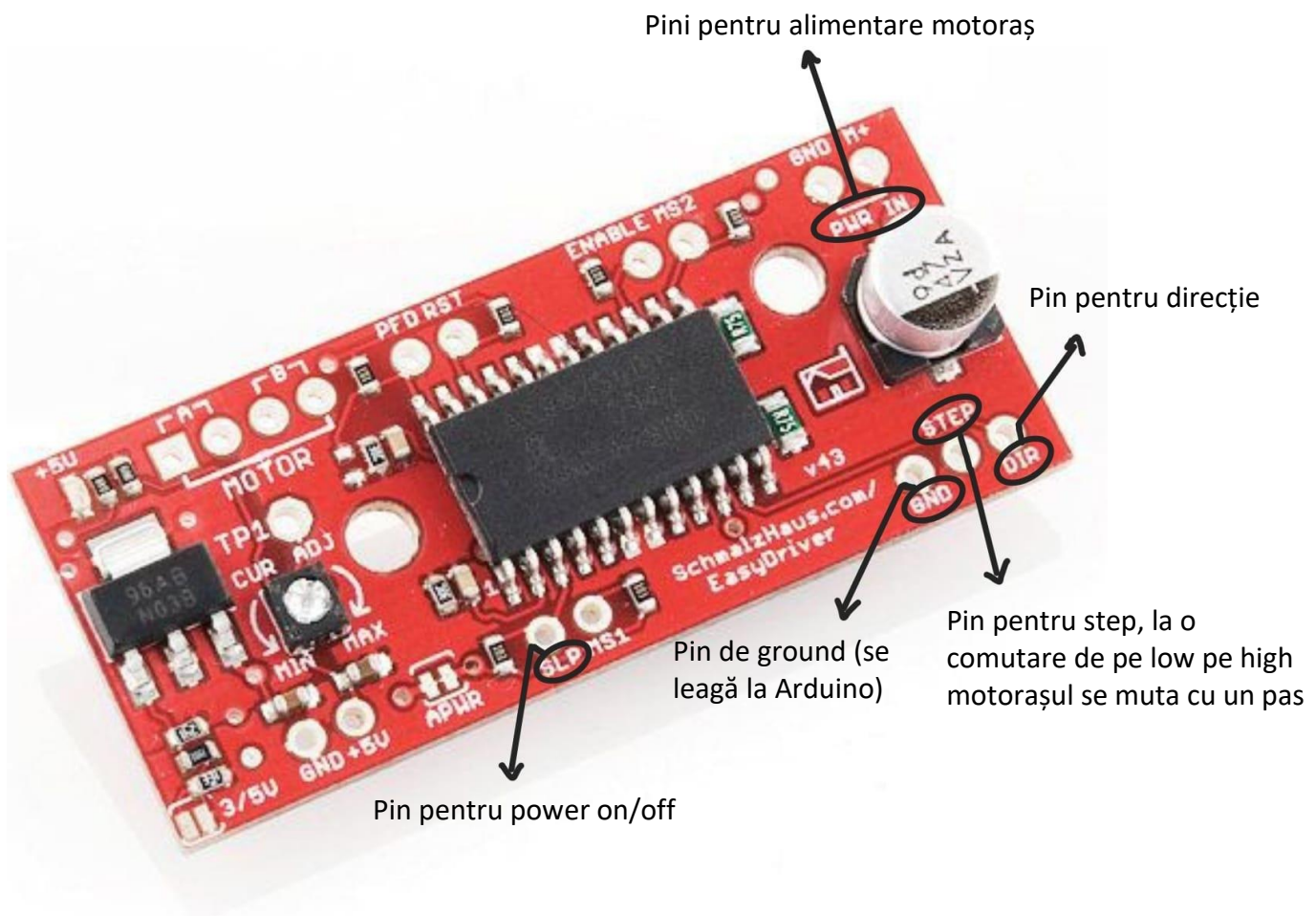
Descrierea componentelor

Motorul pas cu pas

Un motor pas cu pas în regim bipolar este un tip de motor electric utilizat în aplicații în care este necesară poziționare precisă și control fin al mișcării. Motorul pas cu pas este compus din două bobine independente, cunoscute și sub denumirea de faze. Aceste bobine sunt alimentate într-o secvență specifică pentru a genera un câmp magnetic care face rotorul să se rotească. Prin schimbarea secvenței de alimentare a bobinelor, motorul poate fi controlat pentru a realiza o varietate de mișcări, inclusiv mișcări înainte și înapoi, precum și rotire într-o poziție specifică.

Driverul

EasyDriver MCN-DRV-14 este bazat pe circuitul integrat A3967, care oferă un control precis al curentului și direcției pentru motoarele pas cu pas în regim bipolar. Oferă opțiuni de control flexibile, inclusiv posibilitatea de a utiliza micro-pași pentru o precizie mai mare în mișcare. EasyDriver MCN-DRV-14 acceptă semnale de control de la un microcontroler, cum ar fi Arduino, prin intermediul unui set simplu de pini de conectare. În instalația de față, pe lângă GND și pinul de alimentare, se folosesc alți trei pini, de date, mai exact unul pentru pornire/oprire, unul pentru direcție și unul pentru executarea pasului.



Din punct de vedere al codului, funcția ce gestionează EasyDriverul este cea mai importantă și este cea care face posibilă închiderea și deschiderea geamului. Pentru o mai bună precizie a mișcărilor motorului, pașii sunt împărțiți în micro-pași. Știind direcția în care trebuie să se miște motorul, se fac atâtea comutări de low-high pe pinul de step, câți micro-pași trebuie executați la o apelare a funcției. După 20 de secunde de la ultima mișcare a motorului, driverul este pus pe power off, pentru a nu se supraîncălzi.

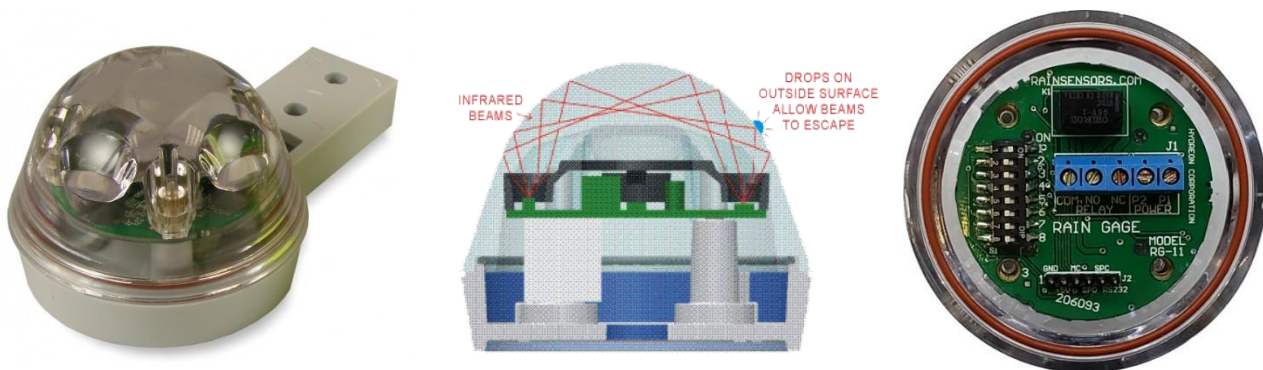
Senzorii de prezență

Mișcarea geamului prin comanda de glisare a mâinii este posibilă doar datorită celor doi senzori de prezență folosiți. Aceștia sunt montați pe o tijă, unul mai jos, iar unul mai sus, și le este limitat câmpul de vizibilitate prin intermediul unor tuburi.

Mișcarea mâinii prin fața lor, de la senzorul de jos la cel de sus, este interpretată ca și direcție de mers în sus, iar la mișcarea mâinii în sens invers, adică de la senzorul de sus la cel de jos, se consideră ca fiind direcția de mers în jos. Întreruperea mișcării curente se poate face glisând mana prin fața senzorilor în direcția nouă dorită de mișcare. Însă, din cauza nivelului slab de ajustare a senzorilor, aceștia nu detectează întotdeauna prezența unui obiect din câmpul lor vizual.

Senzorul de ploaie

Senzorul de ploaie RG-11 detectează apa care lovește suprafața sa exterioară folosind fascicule de lumină infraroșie. Utilizează același principiu de detectare utilizat în ștergătoarele de parbriz cu senzori de ploaie. Senzorul este extrem de sensibil, astfel că acesta poate număra picurii de apa care cad pe suprafața sa.



Deoarece în mecanismul de funcționare al senzorului este integrat un releu, când circuitul este deschis trebuie să se asocieze o valoare acestei stări. În cazul de față, am setat pinul I/O asociat senzorului de ploaie pe high (rezistența pull-up), fiind conectat la 5V. Când releul va închide circuitul se conectează la GND și "trage" tensiunea pe pinul de I/O la 0V (ground).

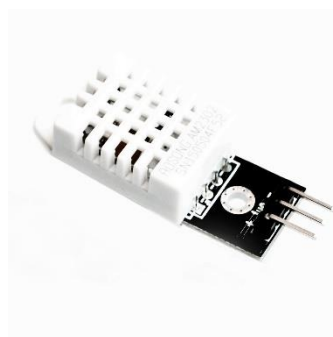
Principala funcționalitate a senzorului de ploaie, este de a detecta dacă a început ploaia, moment în care geamul se închide, însă senzorul este capabil și să sesizeze picurii individual.

Pentru a determina numărul de picuri, o dată la 200 de ms se verifică statusul senzorului, adică dacă a sesizat ploaie sau nu, și se numără, până când trec 5 secunde de la ultimul picur sesizat, moment în care se consideră ca ploaia s-a oprit. Dacă numărul de picuri se află între 1 și 5, atunci se consideră că ploaia este de intensitate mică, dacă este între 5 și 10, de intensitate medie, și peste 10, de intensitate mare.

Senzorul de temperatură și umiditate

DHT22 (digital output relative humidity and temperature) este un senzor digital de temperatură și umiditate, ce are o precizie de măsurare mare și timp de răspuns scurt, de 2 s. Acesta are evident 3 pini, unul de alimentare, unul GND și unul de date (OUT).

O dată la 5 s este citit senzorul și sunt afișate pe LCD informațiile obținute.



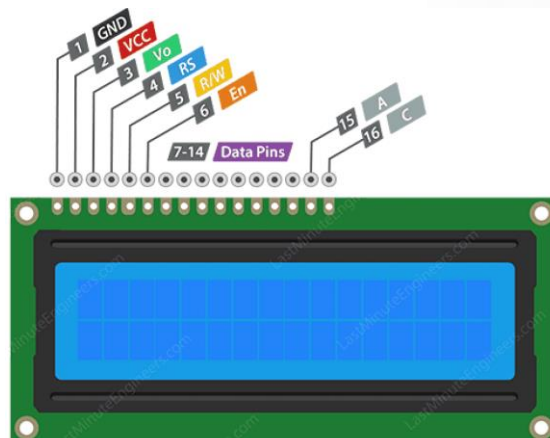
Ecranul LCD

LCD-ul (Liquid Crystal Display) de 16x2 folosește 9 pini de date de la Arduino. Acesta este legat, bineînțeles, și la un pin de alimentare cât și la unul de ground.

Pe acest ecran sunt afișate diverse informații receptate de Arduino, de la diverși senzori, cum ar fi: temperatura în grade Celsius și umiditatea în procente, unul, doi sau trei picuri în funcție de intensitatea ploii, și săgeți care să indice direcția mișcării. De asemenea, pe ecran este afișat și un mesaj de atenționare în momentul resetării.

Am menționat afișarea unor picuri pe ecranul LCD. Pe lângă posibilitatea de a afișa caractere predefinite pe ecran, LCD-ul vine și cu avantajul de a-ți putea crea singur un caracter special printr-un byte array de dimensiune 8.

```
byte raindrop[8] = {
  0b00000,
  0b00100,
  0b01010,
  0b10001,
  0b10001,
  0b10001,
  0b10001,
  0b01110,
  0b00000
};
```



Butonul

Componentă folosită pentru pornirea și oprirea driverului, pentru a facilita modul de funcționare a întregului sistem, întrucât unii senzori au nevoie de un scurt timp de adaptare la mediu. În momentul opririi driverului prin apăsarea butonului, se scrie în EEPROM (Electrically Erasable Programmable Read-Only Memory) poziția motorului, iar în momentul pornirii se recuperează această informație. În cazul în care această metodă eșuează, prin apăsarea timp de 3 secunde pe buton se resetează poziția motorului în cod și acesta se poate muta manual până în poziția 0.

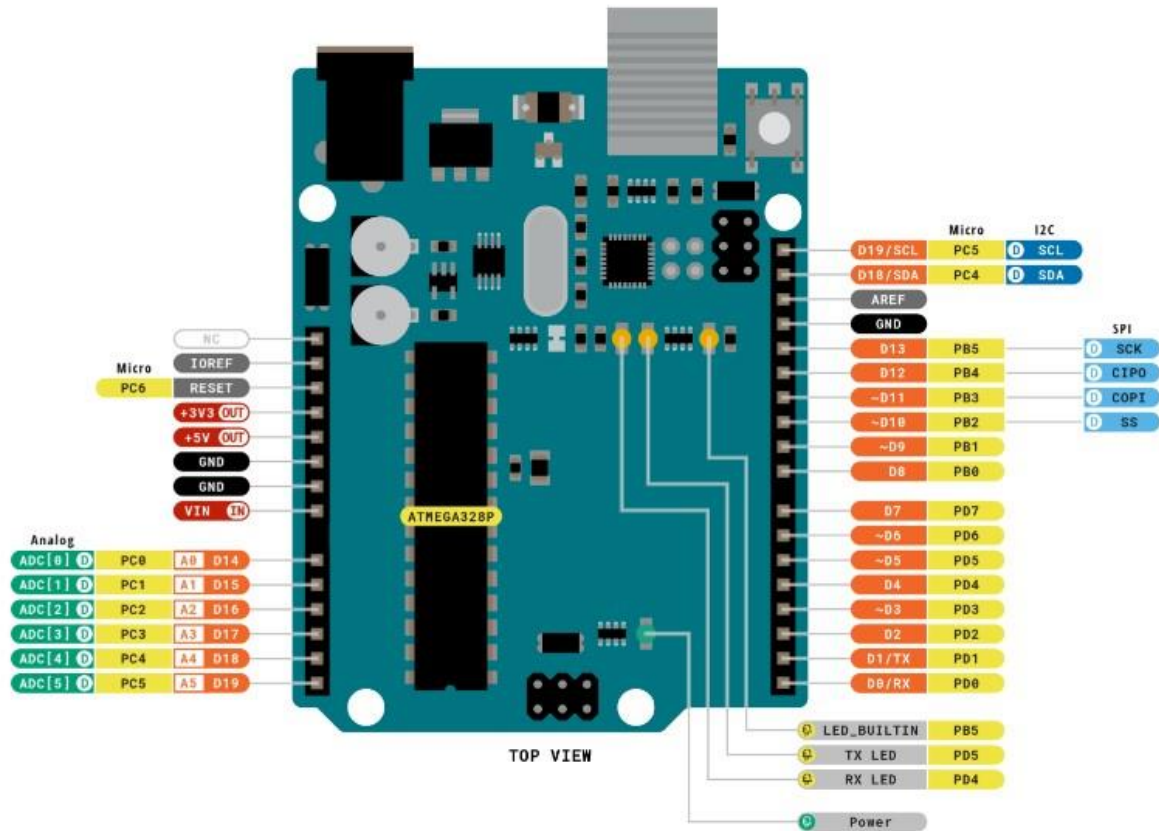
Placa Arduino

Despre microcontrolerul Arduino Uno

Arduino Uno este o placă de dezvoltare hardware open-source, proiectată pentru a facilita programarea și prototipizarea proiectelor electronice. Este una dintre cele mai populare și accesibile plăci Arduino și este potrivită atât pentru începători, cât și pentru utilizatori mai avansați.

Placa Arduino Uno este bazată pe un microcontroler ATmega328P și include o serie de pini de intrare/ieșire digitale și analogice (14 pini I/O digitali, 6 pini I/O analogici). Acești pini pot fi programați pentru a controla diferite componente și dispozitive electronice, precum LED-uri, motoare, senzori și multe altele. Placa dispune, de asemenea, de porturi USB și conectori pentru alimentare și programare.

Arduino Uno este compatibilă cu mediul de dezvoltare Arduino, care oferă un limbaj de programare simplu și unelte pentru a crea și încărca codul pe placă.



Legend:

■ Digital	■ I2C
■ Power	■ SPI
■ Ground	■ Analog
■ Main Part	

ARDUINO

ARDUINO UNO REV3
SKU code: A000866
Pinout
Last update: 6 Oct, 2022

Conectarea componentelor în placa Arduino

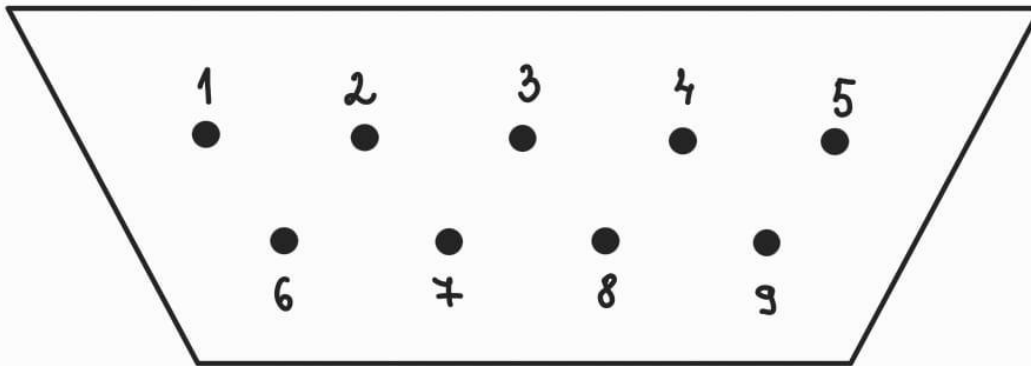
Fiecare dintre componentele amintite mai sus sunt conectate la Arduino în pini I/O, GND (ground = masa comuna) și alimentare (5V). Placa în sine este alimentată prin pinul Vin (Voltage in), cu curent de tensiune între 7V și 12V, curent care este mai apoi transformat intern în curent de 5V.

Pinii de date (I/O) sunt organizați după cum arată tabelul următor:

		Stepper Driver	Rain Sensor	LCD	DHT	PIRD	PIRU	BTN
Digital	D0							
	D1							
	D2			LCD D7				
	D3			LCD D6				
	D4			LCD D5				
	D5			LCD D4				
	D6			LCD Enable				
	D7			LCD RS				
	D8						PIRU	
	D9				DHT			
	D10		Rain					
	D11	Sleep						
	D12	Dir						
	D13	Step						
Analog	A5							BTN
	A4					PIRD		
	A3							
	A2			LCD				
	A1			LCD				
	A0			LCD				

Deoarece pinii digitali sunt toți ocupați, am folosit și pini analogici, dar pe care i-am citit în format digital.

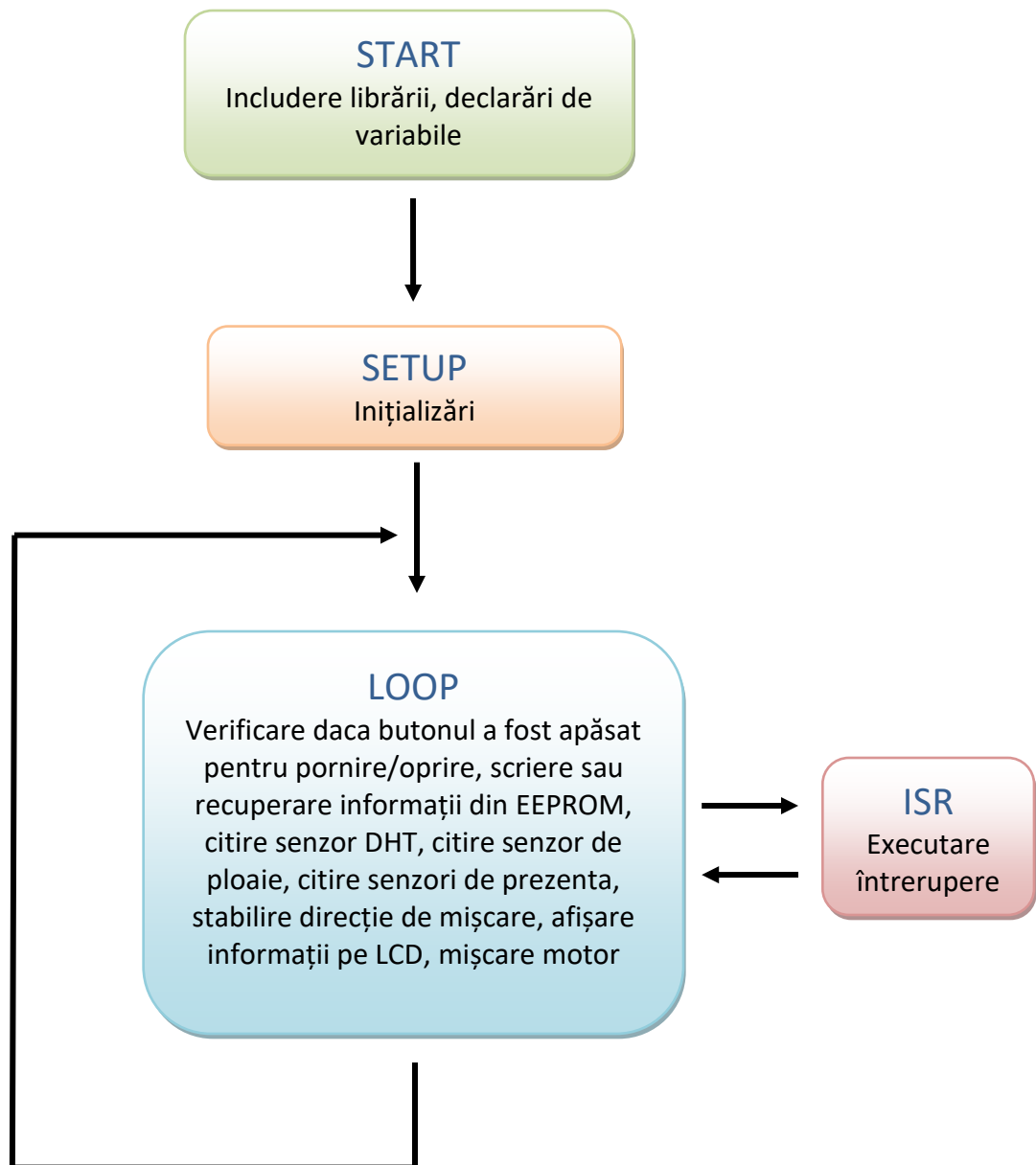
LEGARE SENZORI



	DB9 Pins	Sensors			
		DHT	PIRD	PIRU	BTN
GND	1	green	brown	brown	green
5V	2	yellow	orange	orange	yellow
Out	3			red	
Out	4		red		
Out	5				blue
Out	6	blue			

Descrierea codului

Diagrama logică



Explicații

Specific mediului de dezvoltare Arduino, în cod se regăsesc două funcții principale, setup și loop. O componentă neobișnuită ce se regăsește în schema de mai sus și un aspect important al eficienței funcționării senzorilor de prezență este introducerea metodei ISR (Interrupt Service Routine). Această funcție este folosită pentru a întrerupe executarea în continuare a codului, la activarea senzorilor de prezență, moment în care se execută codul din funcția ISR. Motivul pentru care am folosit această metodă este faptul că, din cauza execuției extrem de rapide a funcției loop, uneori poate sări peste citirea unui senzor, în sensul că acesta se activează, însă în acel moment se execută altă parte din cod, iar până ajunge la citirea senzorului, acesta s-a dezactivat.

Observații și posibile îmbunătățiri

- Un aspect ce ar putea fi îmbunătățit este acuratețea senzorilor de prezență, întrucât aceștia nu receptează tot timpul obstacolul. De asemenea, la pornire au nevoie de o perioadă de timp pentru a „se încălzi”, timp în care nu funcționează în parametrii normali.
- De luat în considerare ar fi și capacitatea de stocare limitată a EEPROM-ului și posibilitatea de a scrie doar de 100.000 ori.
- O altă problemă întâmpinată este greutatea geamului, în sensul că mecanismul de angrenare nu poate susține în repaos un geam de dimensiuni mari. Problema ar putea fi rezolvată alimentând constant motorul, însă asta ar duce la supraîncălzirea driverului. Cea mai bună soluție ar fi utilizarea unui mecanism de tip macara (și cu melc), cum ar fi cel folosit la geamurile mașinilor.

Resurse și bibliografii

- ❖ <https://dronebotworkshop.com/interrupts/>
- ❖ <https://components101.com/sensors/dht22-pinout-specs-datasheet>
- ❖ <https://www.instructables.com/How-to-Interface-Easy-Driver-With-Stepper-Motor-Us/>
- ❖ <https://rainsensors.com/products/rg-11/>
- ❖ <https://www.youtube.com/watch?v=vxxnPJBxG3M>
- ❖ <https://docs.arduino.cc/hardware/uno-rev3>
- ❖ <http://omerk.github.io/lcdchargen/>
- ❖ <https://www.electronics-tutorials.ws/logic/pull-up-resistor.html>
- ❖ <https://ardushop.ro/ro/>
- ❖ <https://lasopadreams127.weebly.com/blog/display-lcd-16x2-datasheet-pdf>
- ❖ <https://docs.arduino.cc/learn/programming/eprom-guide>
- ❖ <https://reference.arduino.cc/reference/>
- ❖ <https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board>
- ❖ <https://rainsensors.com/applications/drop-detection/>
- ❖ <http://cactus.io/hookups/weather/rain/hydreon/hookup-arduino-to-hydreon-rg-11-rain-sensor>
- ❖ <https://www.instructables.com/Arduino-TempHumidity-with-LCD-and-Web-Interface/>
- ❖ <https://forum.arduino.cc/t/measuring-how-long-a-button-was-pressed/404724/4>