

Instrument accesoriu pentru ajutorul in miscare al persoanelor nevazatoare - Blind omnitrix

Observatii:

- Arhivele care contin fisierele sursa se gasesc in sectiunea Download.
- Linkul catre repo-ul de GitHub unde am incarcat sursele se gaseste in sectiunea Software Design.
- Video-urile finale care evidentiaza functionalitatile se gasesc in sectiunea Rezultate Obtinute.
- Pozele si schemele se gasesc in sectiunea Hardware Design.

Introducere

Prezentarea pe scurt a proiectului vostru:

- **ce face:**
- Proiectul contine 2 functionalitati principale: bipaitul intermitent si modul vibratie al motorului servo. Acesta contine un senzor de distanta cu ultrasunete care are un range de 4 m. Frecventa bipaitului creste pe masura ce te apropii de un obiect si scade pe masura ce te indepartezi. Volumul in decibeli este modificat cu ajutorul unui potentiometru care in functie de valoarea citita va cauza pornirea sau oprirea unui buzzer. Pentru sunet se folosesc 3 sau 4 buzzere active de 5V. Exista 3 butoane: unul porneste/opreste global toata functionarea, altul porneste/opreste doar functionarea buzzerelor si ultimul porneste/opreste functionarea servoului. Nu in ultimul rand, prin rotatia pe baza de PWM a servoului se vor genera vibratii.
- **care este scopul lui:**
- Scopul este de a implementa un dispozitiv pentru ajutorul persoanelor nevazatoare care sa le ajute sa se deplaseze. In modul sunet, dispozitivul se va comporta asemanator senzorului de parcare de la masina. Volumul va putea fi reglat din potentiometru. In modul vibratie, gandit pentru situatii in care modul sunet nu poate fi folosit, servoul va genera vibratii care vor anunta apropierea unui obstacol.
- **care a fost ideea de la care ați pornit:**
- Ideea are la baza combinatia intre dorinta de crea un dispozitiv util pentru ajutorul celor care au nevoie, si imaginea unui dispozitiv extraterestru dintr-un desen animat:
- [Omnitrix Pictures](#)
- **de ce credeți că este util pentru alții și pentru voi:**
- Dispozitivul este util pentru a imbunatatii abilitatile de miscare ale persoanelor nevazatoare, adaugand un grad sporit de libertate fata de bastonul clasic. Pentru mine proiectul reprezinta

intrarea in lumea proiectelor Embedded si familiarizarea mea cu domeniul.

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.



Pentru o mai buna clarificare, butonul de shutdown global este de fapt un buton care genereaza resetarea programatica (din cod) a placii Arduino.

Hardware Design

Aici puneți tot ce ține de hardware design:

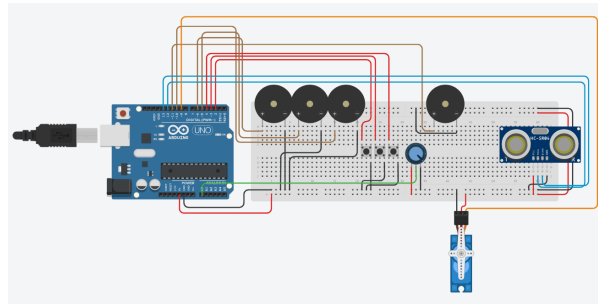
- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri)
- diagrame de semnal
- rezultatele simulării

1. Placa Arduino UNO
2. 4 * buzzer activ 5V
3. Potentiometru
4. 3 * butoane
5. Senzor digital cu ultrasunete HC-SR04
6. Motor MicroServo SG90 9g
7. Baterie externa 9V

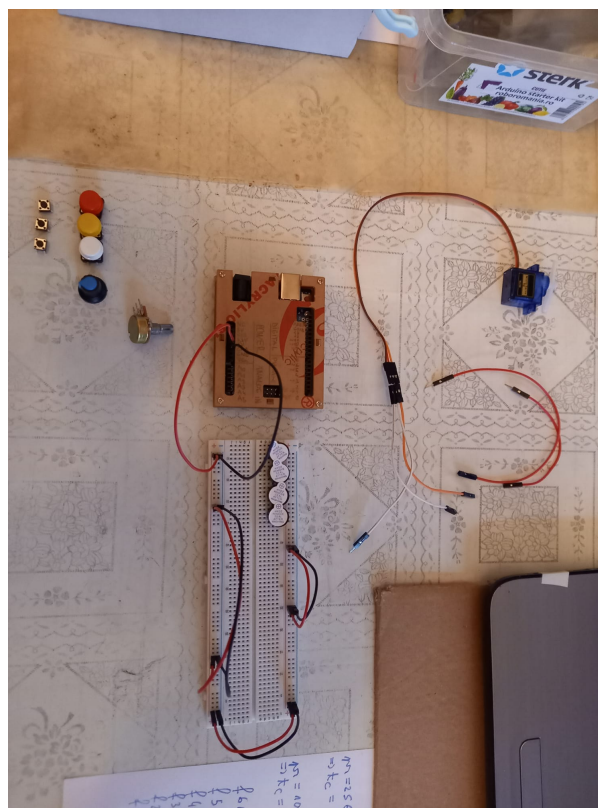


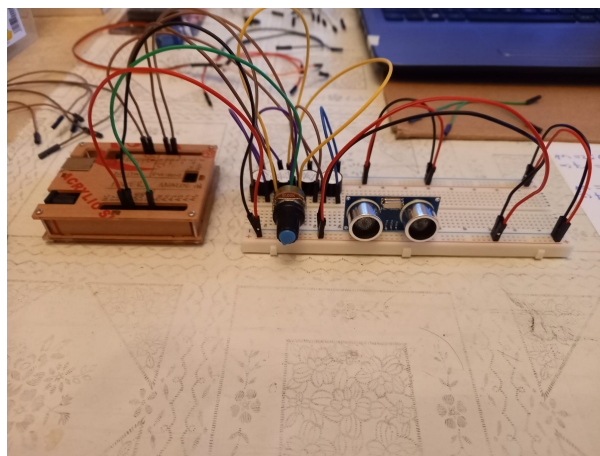
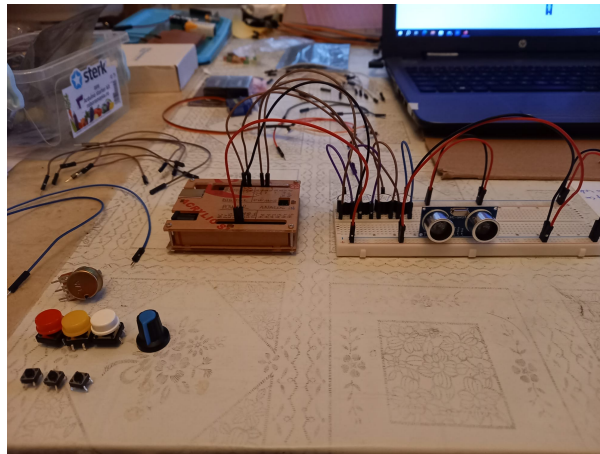
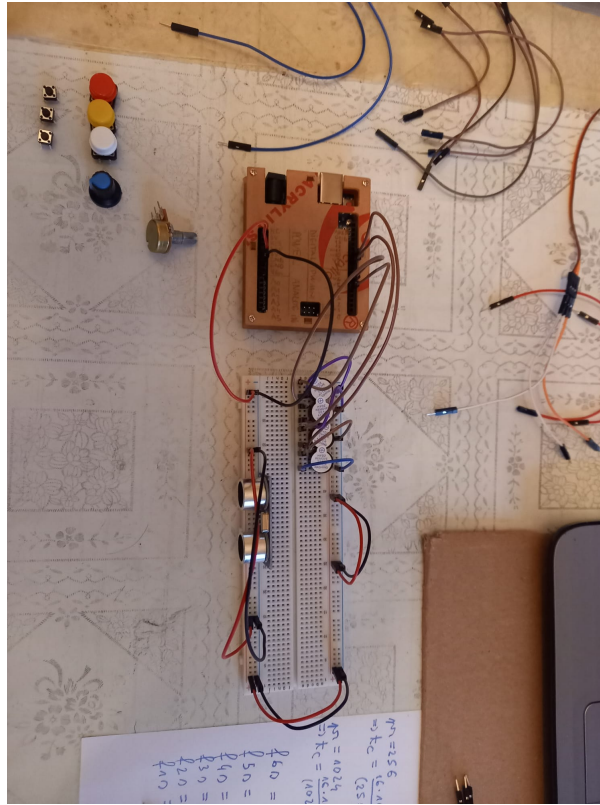
Ca urmare a implementarii si conectarii pieselor hardware ale proiectului am ajuns la concluzia ca schema initiala trebuie usor modificata. Mai exact, initial buzzerule active erau conectate pe pinii 8, 9, 10, 11 digitali de pe care isi luau semnalul de 5V, iar semnalul de PWM al servoului era luat de pe pinul 5 de pwm. Prima parte implementata a fost sunetul intermitent al buzzerelor. A doua parte a fost introducerea servoului. Cand am introdus servoul, folosind biblioteca Servo.h din laborator, am

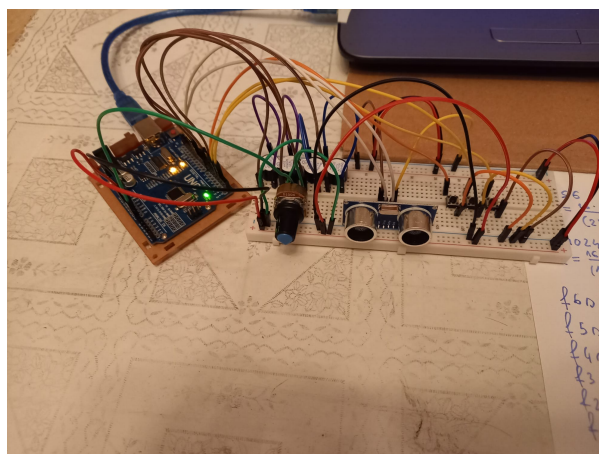
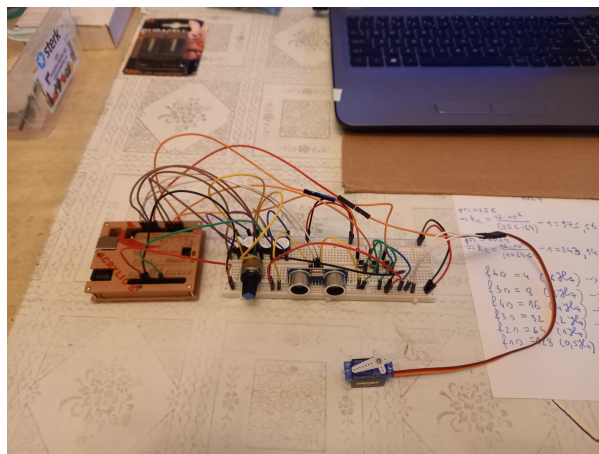
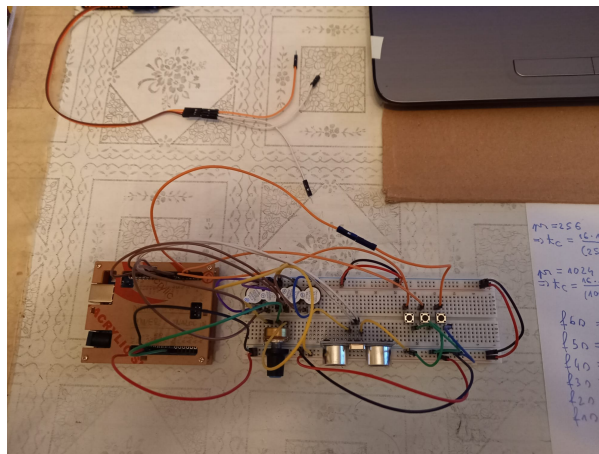
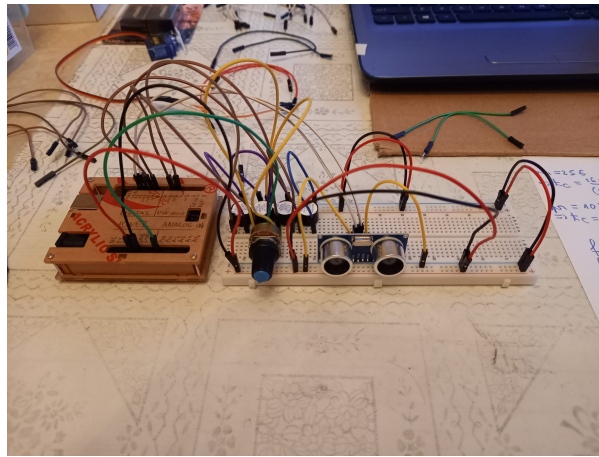
primit o eroare de compilare pe care nu am reusit sa o depasesc decat folosind o alta biblioteca, anume PWServo.h. Aceasta insa impune conectarea servoului doar pe pinii 9 sau 10 de PWM. Drept urmare am eliberat pinii 8 si 9, iar cele 2 buzzere le-am trecut pe pinii 5 si 6.

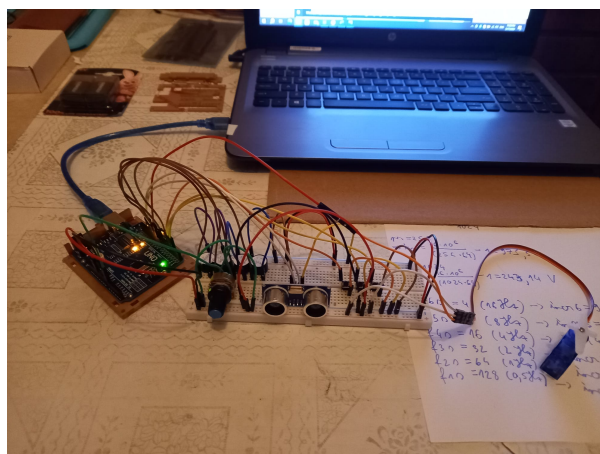
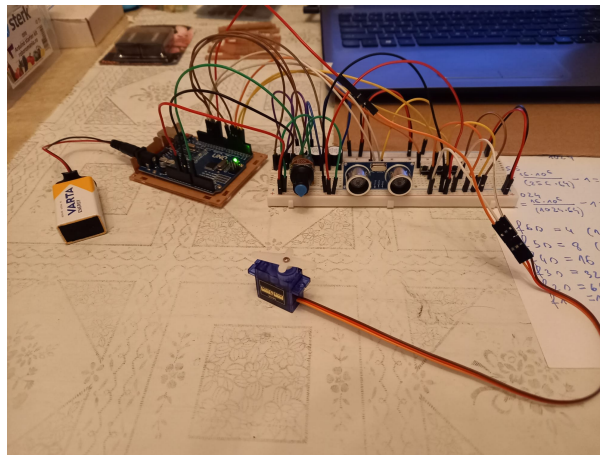
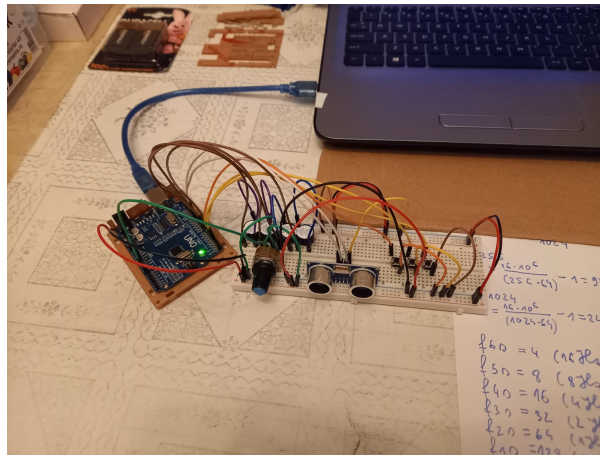
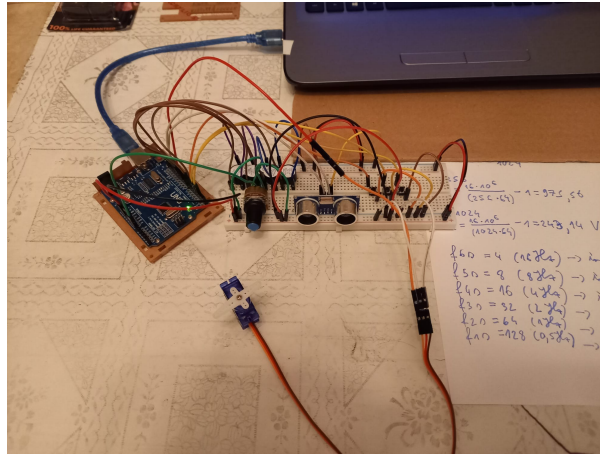


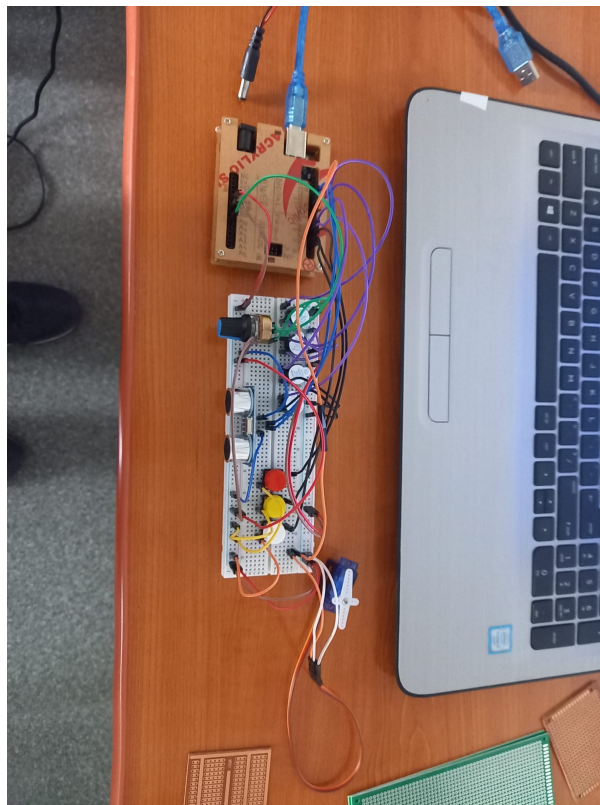
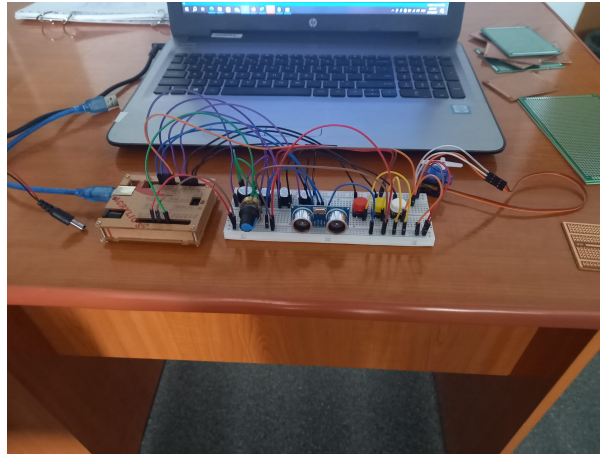
Poze Milestone Hardware:



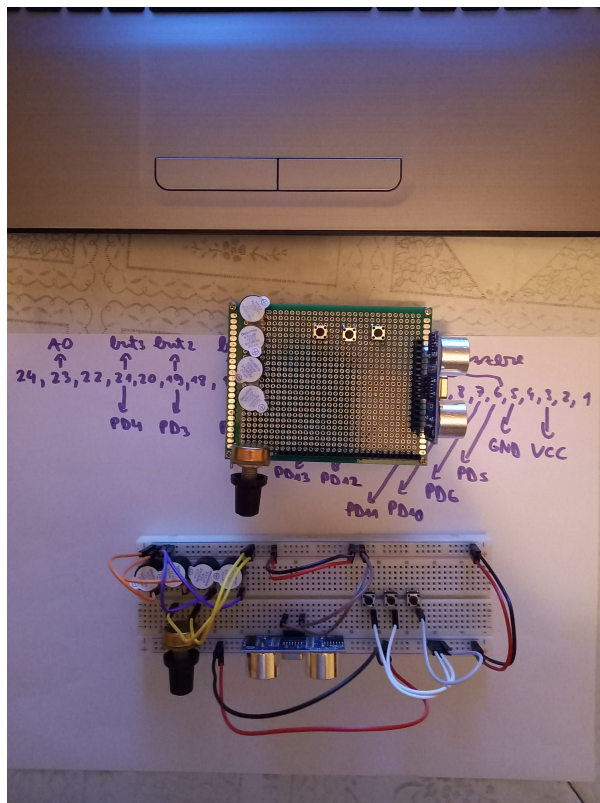
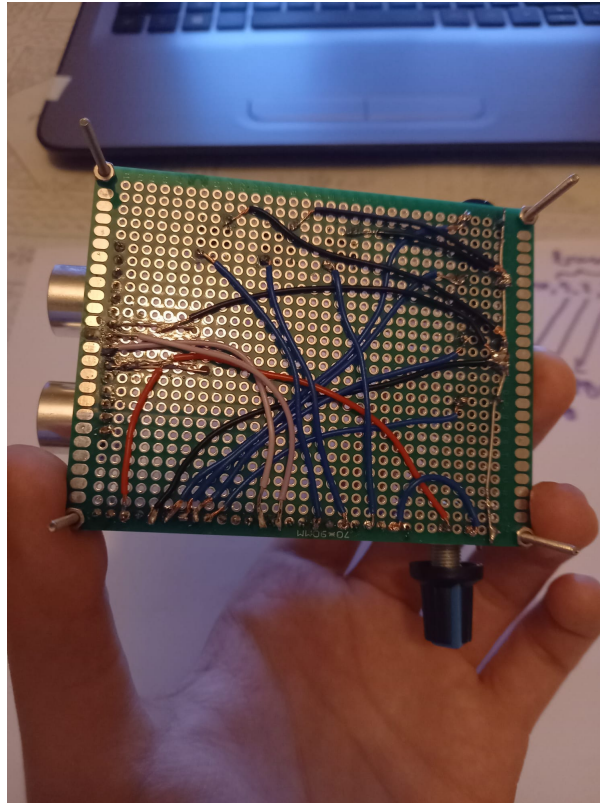


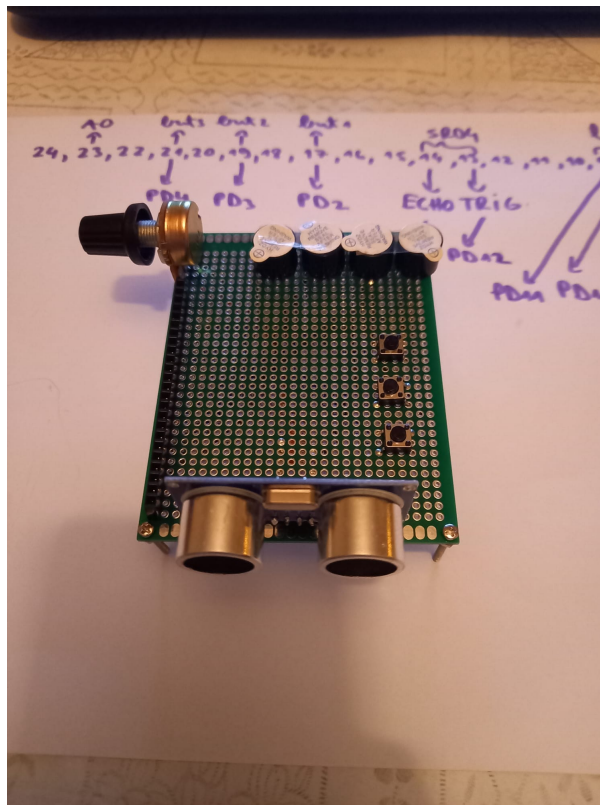
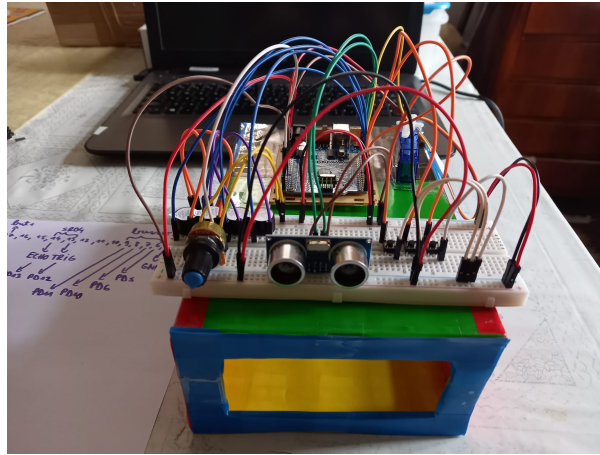


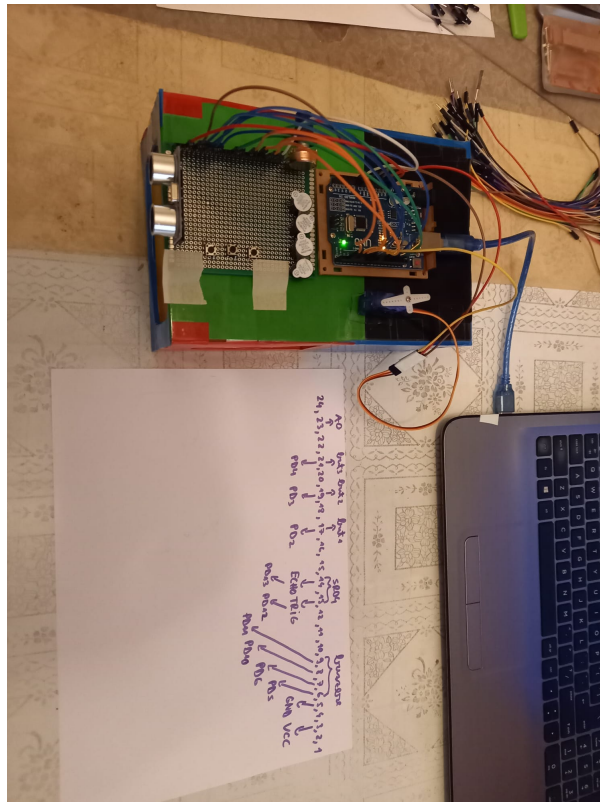
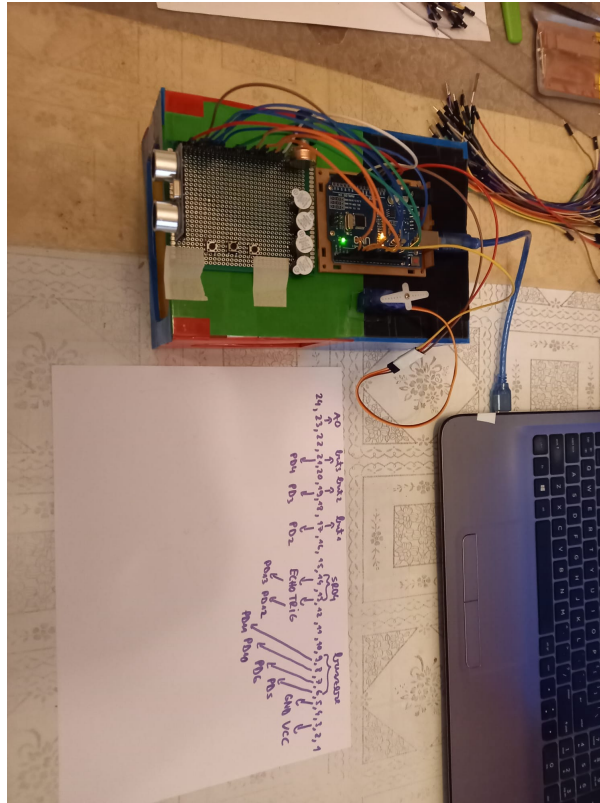


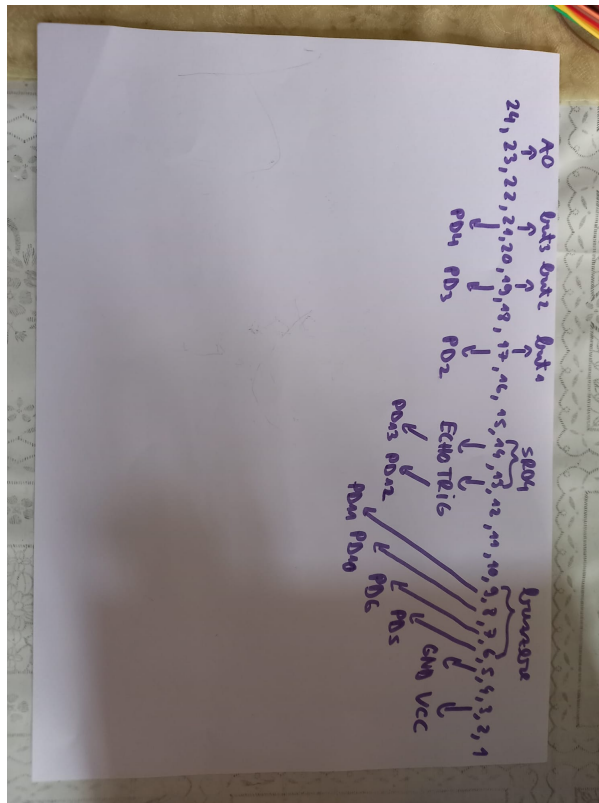
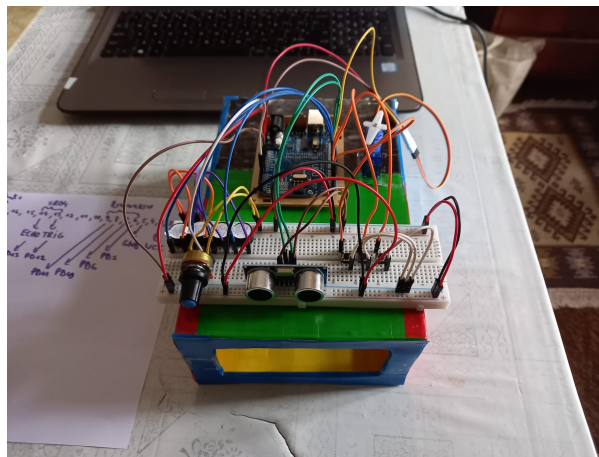
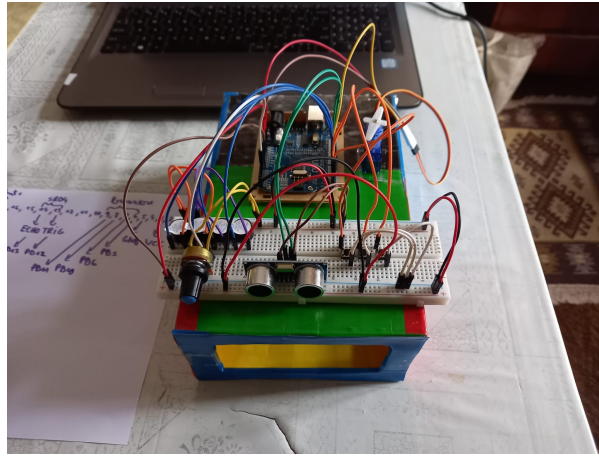


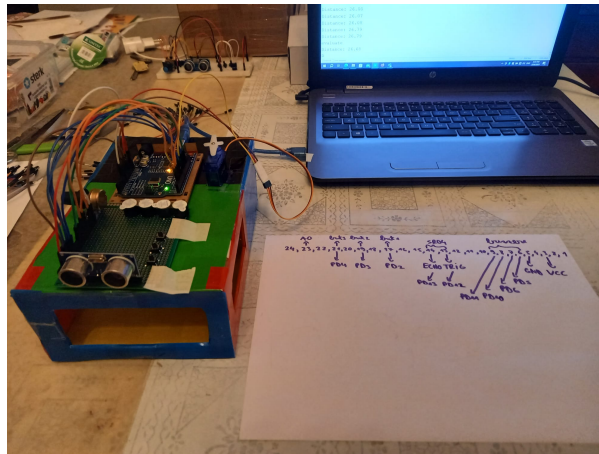
Alte poze ulterioare











Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Mediu de dezvoltare

- Arduino IDE 1.8.19

Librării și surse 3rd-party

- [PWMServo.h](#)
- [limits.h](#)
- [Arduino.h](#)

Cod proiect Github

- [Cod_Proiect](#)

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

[Videouri produs final functionare - youtube.](#)

In urma realizarii proiectului m-am familiarizat intr-o buna masura cu placa Arduino, trasaturile ei specifice si oportunitatile de programare pe care le ofera. De asemenea, am castigat experienta in selectarea si gasirea pieselor necesare pentru un proiect precum si in lipirea pe cablaj imprimat.

Rezultatele concrete obtinute in urma proiectului sunt:


1. Codul proiectului
2. Proiectul implementat pe:
 - breadboard
 - cablaj imprimat

Ambele variante functioneaza corect asa cum au fost gandite. Singura posibila diferenta este nuanta sunetului(pe breadboard am buzzere aproape noi). Am verificat si nu sunt scurturi pe placa de PCB lipita deci nu ar trebui sa fie probleme in utilizare.

Concluzii

- Atunci cand se planifica un proiect trebuie analizate cu atentie resursele hardware disponibile, in speta capabilitatile placii Arduino Atmega 328P. Asta implica si verificarea daca senzorii sau perifericele folosite folosesc implicit vreo resursa hardware (timer de exemplu).
- Codul intreruperilor ISR este bine sa fie cat mai mic si in niciun caz nu trebuie sa contina delayuri. Functia delay() este blocanta, ceea ce inseamna ca, de exemplu, daca vrei sa folosesti millis() in acelasi program, aceasta functie e posibil sa nu apuce sa se execute, deoarece pentru a fmerge are nevoie de o cuanta de timp disponibila cam la fiecare milisecunda.
- Functia millis() nu este blocanta, se foloseste implicit de Timer0 si poate inlocui usor delay-urile blocante. De asemenea, poate fi folosita pentru a simula un timer.

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

[Arhiva surse format zip.](#)

[Arhiva surse format rar.](#)

Jurnal

Descriere globala a pasilor parcursi in realizarea proiectului.

1. Cumpararea pieselor
2. Testarea individuala a pieselor: buzzere, senzor cu ultrasunete HC-SR04, potentiometru, motor micro-servo SG90, butoane
3. Inceperea implementarii fizice a pieselor cu ajutorul breadboardului si scrierea simultana a codului.
4. Initial am folosit toate cele 3 timere. Timer1 pentru actionarea temporizata a servoului, Timer2 pentru sunetul intermitent al buzzerelor, Timer 3 pentru citirea periodica a valorii din registrul ADC-ului luata de pe potentiometru.
5. Am inceput prin a implementa functionalitatea buzzerelor. Am reusit, dar am constatat ca servo-ul foloseste in spate timerul1.
6. Prin urmare, am eliberat complet timerul1 si am pus actionarea servoului si a buzzerelor pe timerul 0. Am setat frecventa intreruperii la 64Hz, astfel incat sa am mai multi divizori sau multipli de care sa ma pot folosi pentru a alterna numarul de sunete emise pe secunda de buzzere.
7. Am facut si conectarea servoului, care acum poate folosi Timer-ul1.
8. Am facut intreruperile butoanelor (2 externe: PD2, PD3 si una de tip pin change.)
9. Saptamana12: Inceperea lipirii pieselor pe cablajul imprimat.
10. Saptamana13:
 - o Am terminat lipirea pieselor si firelor pe placa de cablaj imprima
 - o Am terminat de scris codul
11. O alta modificare pe care am adus-o codului a venit dupa ce am realizat ca functia millis() pe care o folosesc pentru debounce-ul butoanelor se foloseste de timerul 0. Prin urmare am eliberat si timer0 si am pus actionarea servoului si a buzzerelor pe timer 2, iar citirea potentiometrului o fac in loop(), in functia potentiometer(), odata la 500 ms (millis() - lastT_Pot >= 500).
12. Am testat lipiturile facute si nu au scurturi. De asemenea, am conectat PCB-ul la Arduino si proiectul functioneaza. Timerele nu au sunetul absolut perfect. Incerc sa corectez asta, dar functionalitatile de variatie a frecventei sunetului intermitent si modificarea a volumului sunetului cu potentiometrul sunt in regula. Am si varianta pentru breadboard care este in regula.

- [which-timer-is-used-by-the-servo-h-library](#)
- [millis_function](#)

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

Resurse Software:

- [Intreruperi](#)
- [Arduino Millis Tutorial - How to use millis\(\) in Arduino Code for Multitasking](#)
- [Arduino Programming Course Contents](#)

- [Arduino - Button - Debounce | Arduino Tutorial](#)
- [Which timer is used by the "Servo.h" library - Using Arduino / Programming Questions - Arduino Forum](#)
- [GitHub - PaulStoffregen/PWMServo: Control RC Servo motors with interrupt-resilient PWM](#)
- [Using Interrupts on Arduino - Technical Articles](#)
- [Arduino Interrupt Tutorial | Microcontroller Tutorials](#)
- [How to Reset Arduino Programmatically - The Engineering Projects](#)
- [Sg90 servo PWM settings - Using Arduino / Project Guidance - Arduino Forum](#)

Resurse Hardware:

- [Buzzer activ 5V](#)
- [Micro-Servo SG90](#)
- [HC-SR04 senzor ultrasunete Datasheet](#)
- [Arduino Uno](#)

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/alexau/180>



Last update: **2023/05/27 00:27**