

Ultrasonic Distance Alert System - Abudal Ali

Introducere

Sistemul de alertă pentru distanțe cu ultrasunete utilizează un senzor de distanță ultrasonic pentru a măsura distanța obiectelor din raza sa. Sistemul oferă feedback vizual prin LED-uri roșii, galbene și verzi și feedback auditiv printr-o sonerie piezo. Pe măsură ce distanța până la un obiect se modifică, LED-ul corespunzător se aprinde și un ton de alarmă este produs atunci când obiectul se află într-o anumită distanță.

• De ce credeți că este util pentru alții și pentru voi?

1. Siguranța: Sistemul de alertă pentru distanțe cu ultrasunete poate contribui la reducerea accidentelor și a coliziunilor prin detectarea în timp real a obiectelor din apropiere și oferirea de feedback vizual și auditiv. Aceasta poate ajuta în prevenirea situațiilor potențial periculoase și a pagubelor cauzate de coliziuni.
2. Asistență pentru persoanele cu dizabilități: Proiectul poate fi de mare ajutor pentru persoanele cu deficiențe de vedere, oferind asistență în navigare prin intermediul alarmei sonore pe care o produce.

Descriere generală

Arduino Uno joacă rolul de microcontroler, care colectează măsurătorile distanței de la senzorul de distanță ultrasonic. Pe baza distanței măsurate, Arduino controlează LED-urile și buzzer-ul piezo. Când distanța este sub o anumită valoare, LED-ul roșu se aprinde, iar soneria redă un ton de alarmă. Pentru distanțe între X și Y cm, LED-ul galben se aprinde (X și Y încă nu le-am decis). Când distanța depășește o anumită valoare, LED-ul verde se aprinde, iar soneria este silențioasă. Sistemul măsoară în mod continuu distanța și actualizează LED-ul și soneria în consecință.



Hardware Design

Numar Piese	Nume Piesa
1	Arduino Uno R3
1	Breadboard
1	Piezo

3	LED
5	Rezistor
1	Detector de Distanță Ultrasonic
1	Pushbutton
12	Fire tata-tata

Schema circuitului:



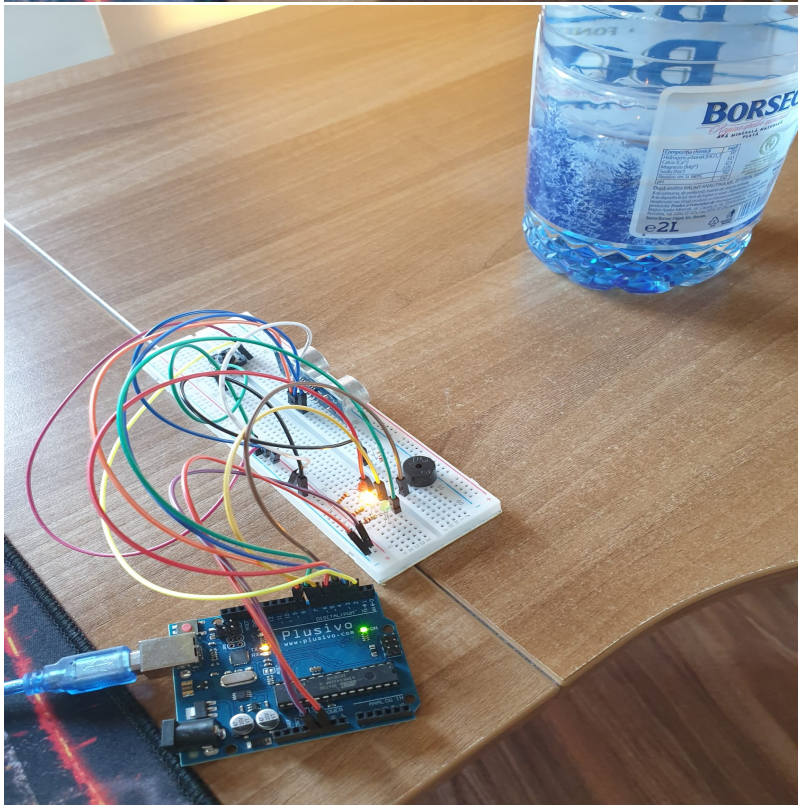
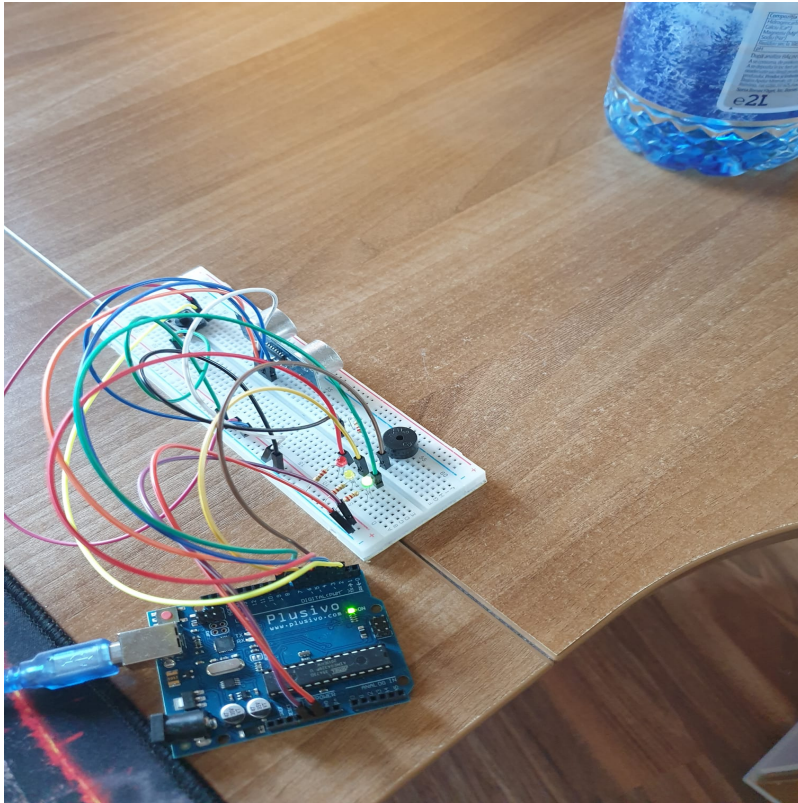
Schematic:

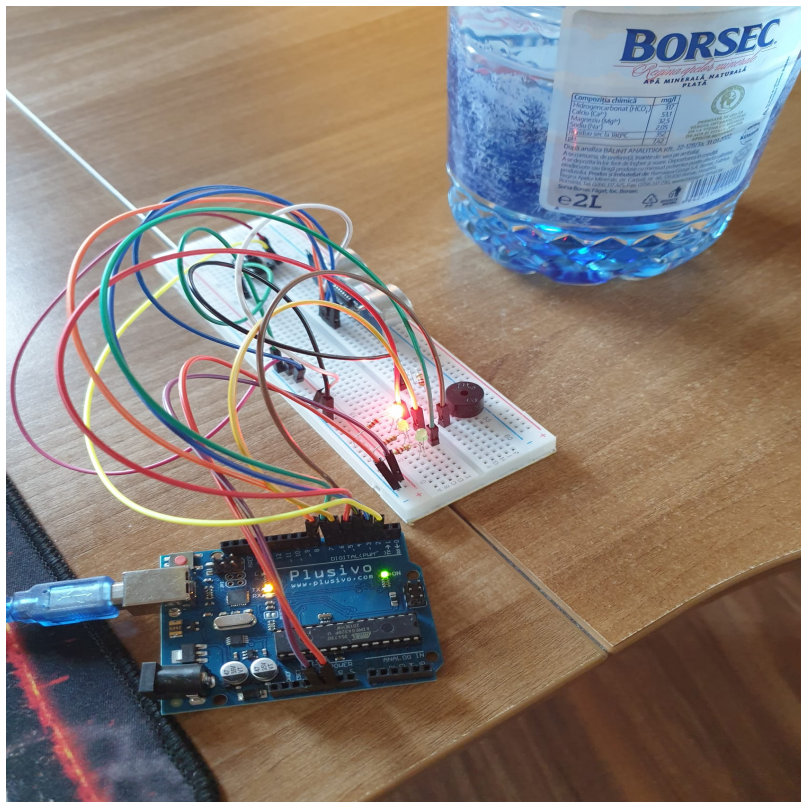


Software Design

- **setup():** Această funcție este folosită pentru a configura pinii și a atașa o întrerupere la buton. Aici, setăm pinii ca intrări sau ieșiri și, de asemenea, stabilim o comunicare serială cu o viteză de 9600 bauds.
- **loop():** Aceasta este funcția principală care se repetă în mod continuu după ce este apelată funcția setup(). În această funcție, distanța este măsurată prin apelarea funcției `measureDistance()`, iar apoi se schimbă starea LED-urilor și a buzzerului în funcție de distanța măsurată și de starea butonului.
- **debounce():** Această funcție este folosită pentru a asigura că semnalul de intrare este stabilit și că semnalele nedorite (bounce) sunt ignorate. Folosește un contor și un timp de întrerupere pentru a determina dacă starea butonului ar trebui să se schimbe sau nu.
- **measureDistance(int trig, int echo):** Această funcție este folosită pentru a măsura distanța utilizând un senzor ultrasonic. Generează un puls pe pinul trig al senzorului, apoi măsoară durata acestui puls pe pinul echo al senzorului. Apoi, folosește această durată pentru a calcula distanța în centimetri.
- **Funcțiile `digitalWrite()`, `analogWrite()`, `tone()`, și `noTone()`:** Acestea sunt funcții predefinite în Arduino care sunt utilizate pentru a controla starea pinilor digitali și analogici, precum și pentru a genera și opri un sunet pe un pin specific.

Rezultate Obținute





Sticla este folosită ca obiect de măsurare a distanței - se poate observa schimbarea de la verde la galben și apoi la roșu în funcție de cum se modifică distanța.

Concluzii

Acest proiect a reprezentat o oportunitate extraordinară de a-mi consolida înțelegerea despre interacțiunea dintre partea de hardware și software a unui sistem embedded, în acest caz, un sistem de alarma cu distanță ultrasonică.

Proiectul a funcționat exact cum mi-am propus, reușind să soluționez toate problemele și bug-urile cu care m-am confruntat pe parcurs. A fost necesară o bună înțelegere a logicii programării și a modului în care aceasta interacționează cu diferitele componente hardware, cum ar fi senzorul ultrasonic, LED-urile și buzzerul.

În concluzie, acest proiect a fost extrem de instructiv și m-a ajutat să înțeleg mai bine cum funcționează un sistem embedded. Deși a existat o curbă de învățare abruptă, sentimentul de satisfacție obținut după rezolvarea fiecărei probleme și văzând sistemul funcționând exact așa cum am intenționat a fost neprețuit.

Download

[abudal_ali_331cb_proiect_pm.zip](#)

Jurnal

- **29.04.2023:** Comandare piese
- **31.04.2023:** Primire piese
- **07.05.2023:** Finalizare partea principala de documentatie pentru prima etapa a proiectului (Introducere, Descriere, Schema circuit, Hardware design, Lista componente)
- **16.05.2023:** Asamblare circuit + testare componente hardware
- **23.05.2023:** Software design + finalizare proiect

Bibliografie/Resurse

Toate piesele au fost achizionate de pe <https://www.optimusdigital.ro/ro/>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2023/fstancu/ali.abudal>

Last update: **2023/05/23 16:10**

