

Smart Dustbin

Autor: Simionescu Maria-Adriana
Grupa: 332CB

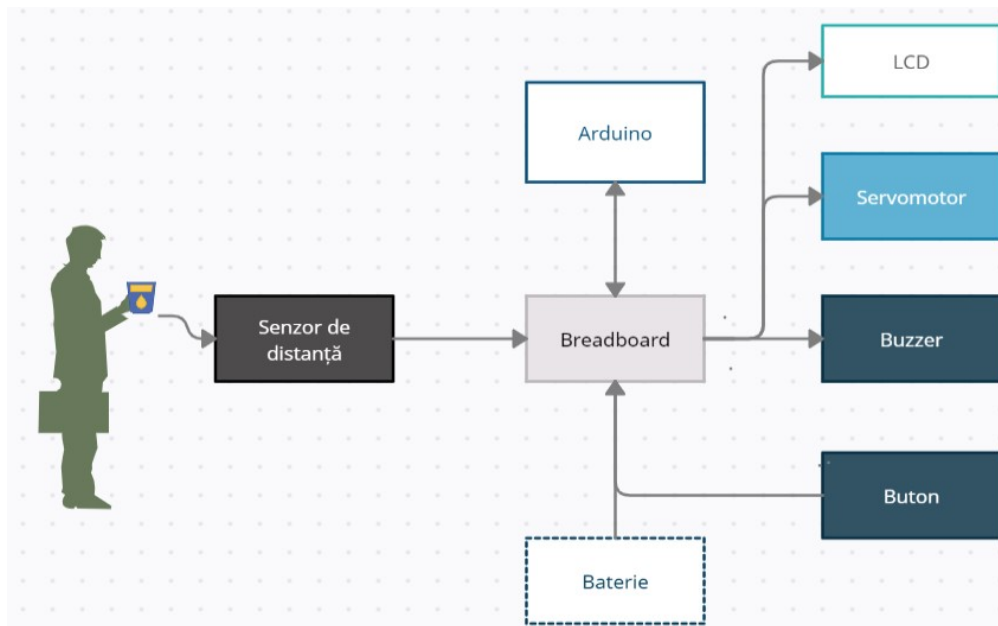
Introducere

„Smart Dustbin” este un dispozitiv care are funcționalitatea unui coș de gunoi automat. Acesta este util deoarece printr-o deschidere automată a capacului unui coș de gunoi se evită contactul fizic al oamenilor cu acesta, deci este conform cu normele de igienă și de asemenea mult mai ușor de gestionat.

Descriere generală

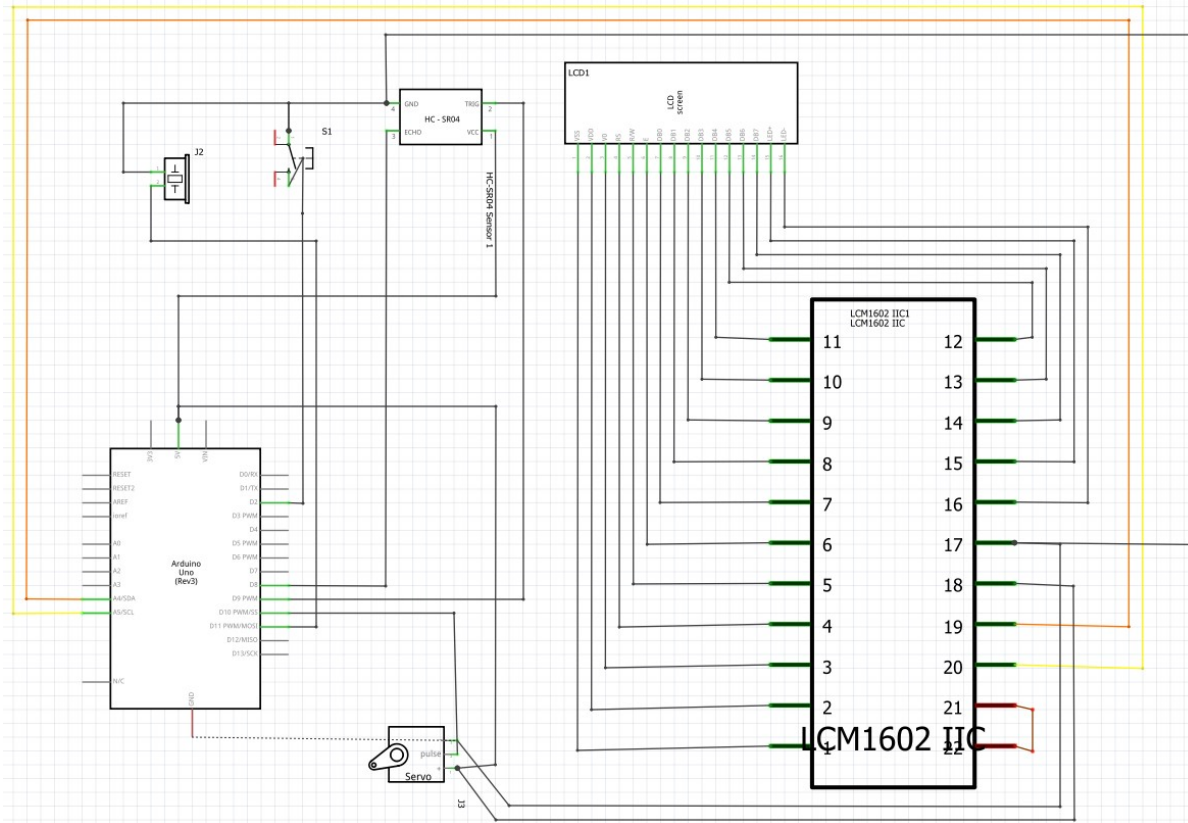
- Dispozitivul detectează printr-un senzor de distanță dacă un obiect se apropie de coșul de gunoi, capacul acestuia se deschide folosind 1 servomotor și obiectul poate fi aruncat.
- Deschiderea capacului are atașat un semnal sonor și un mesaj.
- După ce a fost deschis de un anumit număr de ori, pe un ecran LCD este afișat un mesaj ce sugerează verificarea coșului și pe fundal se va auzi o scurtă melodie.
- Când coșul de gunoi este golit, se apasă un buton din interiorul coșului pentru a reseta numărul de deschideri al coșului. După apăsarea butonului, se va auzi un semnal sonor iar pe ecran va fi afișat un mesaj corespunzător.

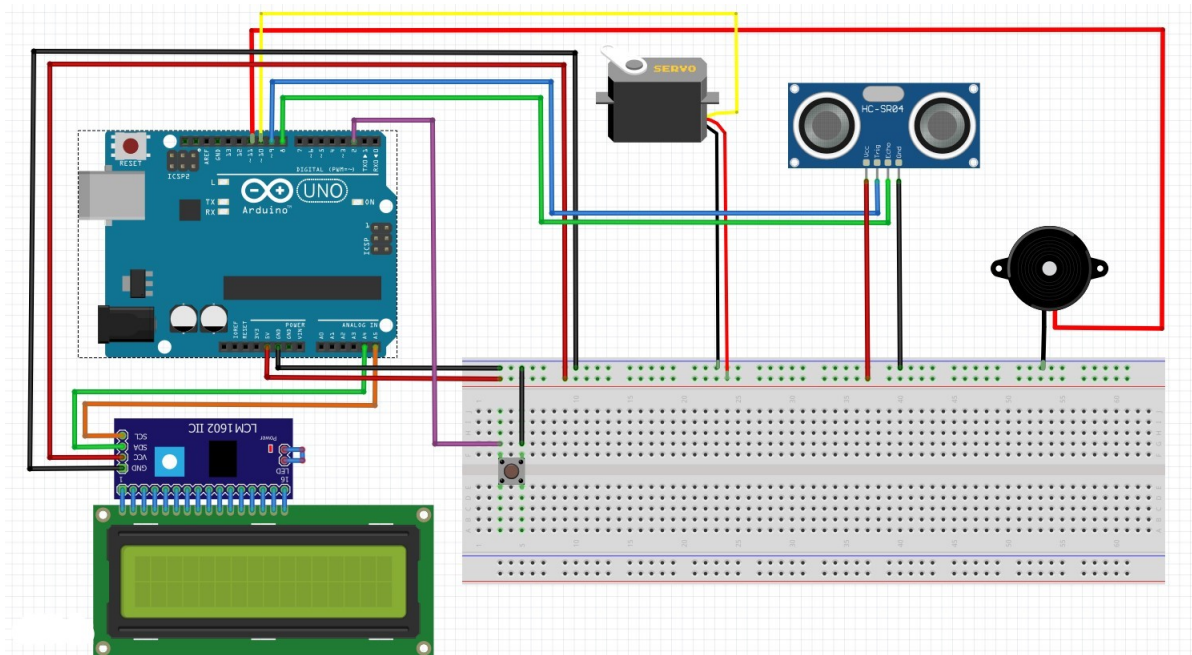
Schema Bloc



Hardware Design

Schema electrică





Lista de Piese

- Arduino UNO
- 1 servomotor
- Modul senzor Ultrasonic
- Fire de legătură
- Buzzer
- Buton
- Breadboard
- LCD
- Modul I2C pentru LCD
- Baterie
- Adaptor baterie

Software Design

Am folosit Arduino IDE [<https://www.arduino.cc/en/software>] pentru a implementa partea de software.

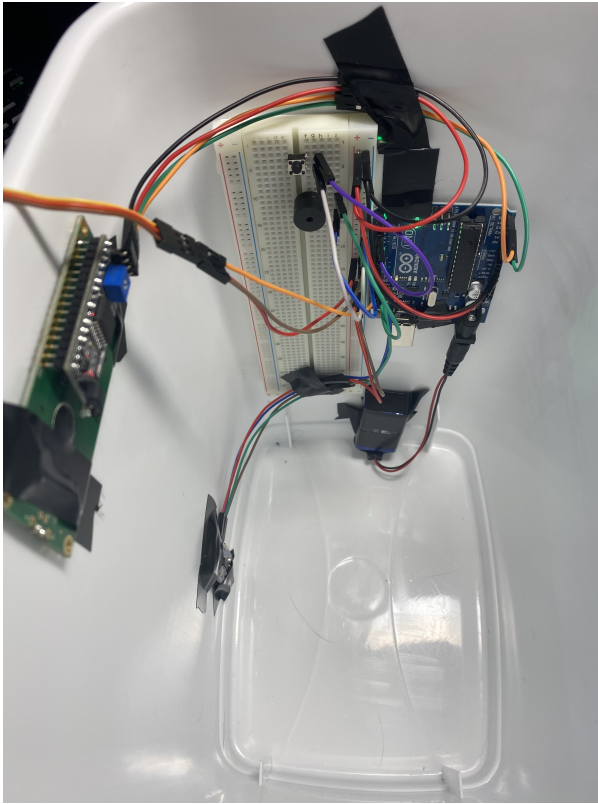
- Bibliotecile folosite sunt următoarele:

- <Servo.h> [<https://www.arduino.cc/reference/en/libraries/servo/>] pentru servomotor;
 - <Wire.h> [<https://www.arduino.cc/en/reference/wire>] pentru comunicarea cu dispozitivele I2C;
 - <LiquidCrystal_I2C.h> [<https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>] pentru LCD.
- Header-ul inclus:
 - "pitches.h" [<https://gist.github.com/mikeputnam/2820675>] folosit pentru notele muzicale replicate de buzzer.
- Descrierea funcțiilor din cod:
 - LiquidCrystal_I2C lcd(0x27, 16, 2).
 - se realizează conectarea LCD-ului prin I2C folosind adresa modului I2C: 0x27 și se specifică faptul că e un lcd cu 2 linii, fiecare linie de 16 caractere;
 - setup() - folosită pentru inițializare:
 - se configurează pinii fie pentru INPUT, fie pentru OUTPUT folosind funcția pinMode() [<https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>];
 - pentru buton se pornește rezistența de pull-up folosind parametrul INPUT_PULLUP.
 - se inițializează cu 0 variabila «count_openings» folosită pentru a contoriza de câte ori s-a introdus ceva în coș;
 - se folosește init() pentru a inițializa lcd-ul și noBacklight() pentru a opri lumina de fundal a ecranului;
 - se atașează variabila destinată servomotorului «servo1» la un pin folosind attach() [<https://www.arduino.cc/en/Reference/ServoAttach>];
 - se inițializează brațul servomotorului la 0 grade folosind write() [<https://www.arduino.cc/en/Reference/ServoWrite>];
 - measure()- folosită pentru a măsura distanța detectată de senzor:
 - se folosește digitalWrite() [<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>] pentru a genera o undă sonoră pentru pin-ul trig al senzorului;
 - se extrage durata pulsului folosind pulseIn() [<https://www.arduino.cc/reference/en/language/functions/advanced-io/pulsein/>];
 - se calculează distanța folosind formula: $Distance = (Time \times Sound\ speed\ in\ Air\ (340\ m/s))/2$
 - play()- folosită pentru a genera un cântec cu ajutorul buzzer-ului:
 - se iterează printre notele melodiei ce sunt păstrate în vectorul «song», se calculează durata fiecărei note folosind vectorul «note_duration», se trimite unda cu frecvența sugerată buzzer-ului folosind tone() [<https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>], iar după un timp se oprește unda generată cu noTone() [<https://www.arduino.cc/reference/en/language/functions/advanced-io/notone/>].
 - loop()-folosită pentru a modifica funcționalitățile programului în funcție de ce recepționează:
 - se verifică starea butonului folosind digitalRead() [<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>];
 - dacă butonul este apăsat, înseamnă că a fost golit coșul de gunoi și în acest caz:
 - se resetează contorul pentru numărul de deschideri la 0;
 - se generează un semnal sonor folosind buzzer-ul;

- se afișează un mesaj corespunzător pe lcd folosind `setCursor()` pentru a seta cursorul lcd-ului la poziția dorită și `print()` pentru a afișa mesajul;
- se așteaptă o perioadă de timp golirea coșului folosind un `delay()`;
- se oprește buzzer-ul, se șterge mesajul de pe lcd și se oprește lumina de fundal a acestuia;
- dacă butonul nu fost apăsat:
 - se calculează o distanță medie realizând 3 măsurători;
 - dacă distanța este mai mică de 40 de cm atunci:
 - dacă se depășește un anumit număr de deschideri:
 - se sugerează golirea coșului de gunoi pe lcd;
 - se deschide capacul coșului cu ajutorul servomotorului;
 - se generează un scurt cântec pe buzzer;
 - altfel:
 - se deschide capacul coșului cu ajutorul servomotorului:
 - se rotește brațul servomotorului la 180 de grade;
 - se generează un semnal sonor pe buzzer;
 - se afișează mesaje la deschiderea și închiderea capacului.
 - se incrementează numărul de deschideri al coșului.

Rezultate Obținute





Link demo: <https://www.youtube.com/watch?v=KaoE2kiASEU> [<https://www.youtube.com/watch?v=KaoE2kiASEU>]

Concluzii

- Am reușit să implementez un coș de gunoi automat funcțional.
- Am învățat ce stă la baza proiectării cu microprocesoare făcând un lucru practic.
- Am învățat greutatea realizării unui design, încercând să poziționez toate piesele într-un mod cât mai aerisit, fără a folosi foarte mult volum din coșul de gunoi. De asemenea, poziționarea servomotorului pe capac pentru a închide și a deschide într-un mod corespunzător capacul coșului a fost un lucru complicat, deoarece acesta are o forță destul de mică.
- Am descoperit cât de frumos este să creezi ceva funcțional aproape de la 0, realizând atât partea de back-end cât și cea de front-end.

Download

Arhiva cu codul:

smart_dustbin_code.zip

<https://ocw.cs.pub.ro/courses/pm/prj2021/alazar/smartbin>

Bibliografie/Resurse

<https://create.arduino.cc/projecthub/mohd-shahid/smart-dustbin-using-arduino-c0bb7a> [<https://create.arduino.cc/projecthub/mohd-shahid/smart-dustbin-using-arduino-c0bb7a>]

<https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>
[<https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>]

<https://www.arduino.cc/en/reference/libraries> [<https://www.arduino.cc/en/reference/libraries>]

<https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/> [<https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/>]

pm/prj2021/alazar/smartbin.txt · Last modified: 2021/05/30 21:22 by maria.simionescu