

Smart trash can

Student: Dragne Lavinia-Stefana

Grupa: 334CA

Introducere

Prezentarea pe scurt a proiectului:

- ce face: cos de gunoi "inteligent" care se deschide cand detecteaza mana la o distanta intre 5 si 20cm
- scopul lui: pastreaza utilizarea igienica, pentru ca nu trebuie atins pentru a fi folosit
- de ce este util: in unele locuri aruncarea deseurilor pe jos este o infractiune grava, de aceea cosul de gunoi cu deschidere automata vine in ajutorul cetatenilor. De asemenea, datorita ledului rosu si buzzerului te poate anunta cand trebuie golit. Daca este gol sau semi-plin, se va aprinde de asemenea, un led colorat corespunzator. Se poate forta deschiderea capacului apasand butonul alb atasat. Butonul poate fi folosit si in caz de urgenta sau ca backup (daca se strica senzorul de distanta).

Descriere generală

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

Cosul cu deschidere automata se foloseste de un senzor de distanta conectat la un Arduino pentru a detecta mana. Daca mana se afla la o distanta cuprinsa intre 5-20cm se actioneaza un servo-motor care va deschide capacul. Folosind tot un senzor de distanta se detecteaza daca cosul de gunoi este plin si se va aprinde un led rosu si un buzzer, ce vor avertiza, ca acesta trebuie sa fie golit, cat mai curand posibil. Daca cosul este gol se va aprinde un led verde, iar daca este semi-plin (distanta de la senzor la gunoi este cuprinsa intre 5-15cm) se va aprinde un led galben. In cazul in care se doreste fortarea deschiderii capacului sau mentinerea acestuia deschis pe o perioada mai mare de timp, se poate apasa butonul atasat. Astfel, se detecteaza schimbarea starii butonului, se actioneaza servomotorul, iar capacul va ramane deschis 5s, dupa ridicarea degetului de pe buton.

Schema bloc :



Hardware Design

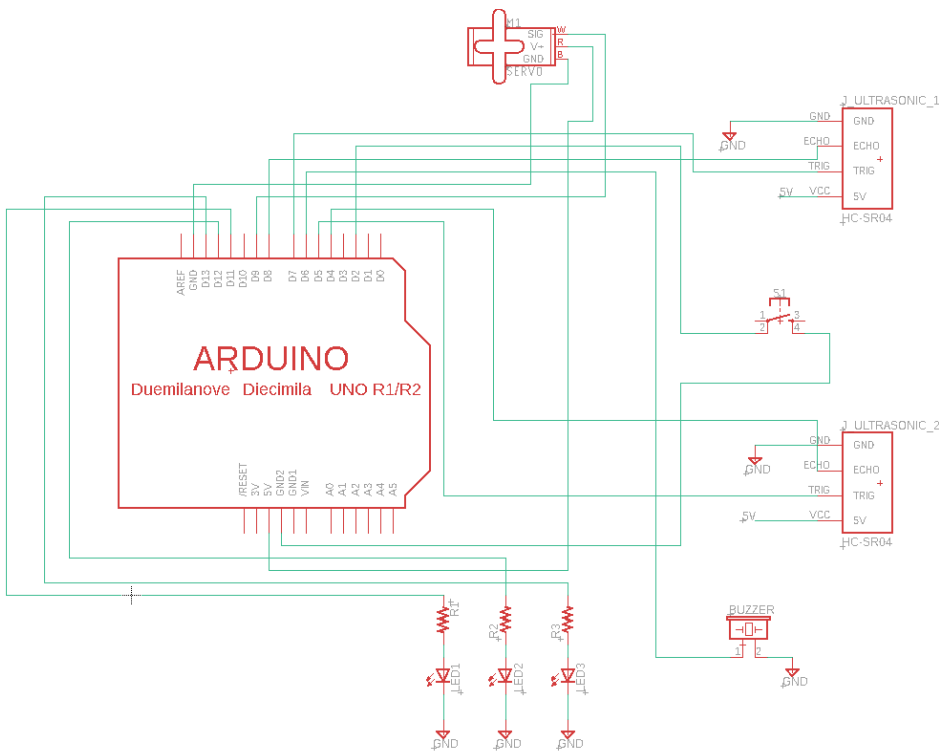
Tot ce ține de hardware design:

- listă de piese
- schema electrica

Lista piese

Componente	Cantitate
Arduino Uno	1
Breadboard	1
RGB Led	3
220 Resistor	3
Buzzer	1
Servo-motor	1
HC-SR04 Ultrasonic Sensor	2
Buton	1
Fire M-M, M-T, T-T	

Diagrama electrica



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

Pentru mediul de dezvoltare am folosit: [Arduino IDE](#) - versiunea 1.8.19.

Biblioteci folosite:

- [<Servo.h>](#) - pentru lucrul cu servomotorul

Mi-am definit cateva macro-uri pentru pinii pe care ii voi folosi la conexiunea cu senzorii de distanta, buzzer-ul, butonul si ledurilor. De asemenea, am folosit dimensiuni constante (de 5 sau 20 cm) pentru distanta la care se poate afla un obiect pentru a actiona servomotorul si distanta la care cosul se poate considera plin (sub 5cm distanta pana la capac).

In *functia de setup* am configurat modurile pentru pinii senzorialor (INPUT pentru cei ECHO si OUTPUT pentru cei TRIGGER), pinii ledurilor (OUTPUT), dar si pinul de date pentru servomotor si INPUT_PULLUP pentru buton.

Funcția *reset_trigger* se asigura ca pinul de trigger al unui senzor de distanta este setat pe LOW (se reseteaza, daca e cazul).

Funcția *get_distance* calculeaza distanta de la cosul de gunoi la mana persoanei care vrea sa arunce un obiect, folosind unul din senzorii de distanta. Distanta la care senzorul actioneaza servomotorul este cuprinsa intre MIN_DIST (5cm) si MAX_DIST (20cm). Cu ajutorul senzorului se masoara semnalul dus-intors cos-mana. Pentru a determina distanta, in centimetri, se face o conversie timp - distanta.

Funcția *get_distance* este apelata si pentru a determina, cu ajutorul celui de-al doilea senzor, cat de plin este cosul de gunoi.

Funcția *open_trash_by_hand* deschide sau inchide capacul, actionand servomotorul, in functie de distanta primita ca si parametru.

Funcția *open_trash_by_button* deschide fortat capacul cosului si il inchide, dupa 5s, daca a fost apasat butonul.

Funcția *check_content* aprinde un led corespunzator starii cosului (plin / gol) si eventual declanseaza buzzerul (daca este necesar). Se va aprinde ledul verde daca cosul este plin, galben daca gunoiul este la o distanta de 5-15cm de senzor si rosu daca gunoiul este la o distanta mai mica de 5cm de senzor. Atunci, se declaseaza si buzzerul pentru a avertiza ca gunoiul trebuie aruncat.

Funcția *loop* :

- masor distanta pana la obiectul din fata capacului

- masor distanta si o printez pe seriala pana la continutul cosului
- verific actionarea butonului
- declansez deschiderea capacului (daca este necesar)
- setez ce led sa se aprinda si daca se actioneaza buzzer-ul sau nu

Codul folosit:

```
#include <Servo.h> // Biblioteca pentru servo-motor

#define SERVO 9
#define ECHO_PIN_DIST 8
#define TRIGGER_PIN_DIST 7
#define ECHO_PIN_FULL 4
#define TRIGGER_PIN_FULL 5
#define BUZZER 6
#define LED_ROSU 13
#define LED_GALBEN 12
#define LED_VERDE 11
#define FULL 5
#define MEDIUM 15
#define BUTTON 2
#define MIN_DIST 5
#define MAX_DIST 20

Servo my_servo;

// Starile butonului
int last_state = HIGH;
int current_state;

void setup()
{
  my_servo.attach(SERVO);
  pinMode(TRIGGER_PIN_DIST, OUTPUT);
  pinMode(ECHO_PIN_DIST, INPUT);
  pinMode(TRIGGER_PIN_FULL, OUTPUT);
  pinMode(ECHO_PIN_FULL, INPUT);
  pinMode(LED_ROSU, OUTPUT);
  pinMode(LED_GALBEN, OUTPUT);
  pinMode(LED_VERDE, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);
  delay(100);
  Serial.begin(9600);
}

// Resetez trigger-ul in starea de low
void reset_trigger(int trigger_pin)
{
  digitalWrite(trigger_pin, LOW);
  delayMicroseconds(5000);
  digitalWrite(trigger_pin, HIGH);
}
```

```
    delayMicroseconds(5000);
    digitalWrite(trigger_pin, LOW);
}

// Calculeaza distanta pana la un obiect
float get_distance(int echo_pin)
{
    float dist;

    // Setez pinul
    pinMode(echo_pin, INPUT);

    // Masor distanta pana la mana
    long duration = pulseIn(echo_pin, HIGH);
    // Distanta in cm (convertire timp - distanta)
    dist = duration / 58.82;
    return dist;
}

// Deschide/ inchide capacul in functie
// de distanta la care se afla obiectul
void open_trash_by_hand(float dist)
{
    int pos;

    if (dist <= MAX_DIST && dist >= MIN_DIST) {
        pos = 90;
        my_servo.write(pos);
        delay(1000);
    }
    else {
        pos = 0;
        my_servo.write(pos);
        delay(50);
        if (dist < 2) {
            tone(BUZZER, 1000, 500);
        }
    }
}

void open_trash_by_button() {
    int pos;

    // Citesc starea curenta a butonului
    current_state = digitalRead(BUTTON);

    if (last_state == LOW && current_state == HIGH) {
        pos = 90;
        my_servo.write(pos);
        delay(5000);
        pos = 0;
    }
}
```

```
    my_servo.write(pos);
    delay(50);
}

// Salvez ultima stare
last_state = current_state;
}

// Verifica cat de plin e cosul
void check_content(float dist_full) {
    if (dist_full <= FULL) {
        // Aprinde ledul corespunzator
        digitalWrite(LED_ROSU, HIGH);
        delay(100);
        digitalWrite(LED_GALBEN, LOW);
        delay(100);
        digitalWrite(LED_VERDE, LOW);
        delay(100);
        tone(BUZZER, 1000, 500);
        delay(1000);
    }
    else {
        digitalWrite(LED_ROSU, LOW);
        if (dist_full > FULL && dist_full < MEDIUM) {
            digitalWrite(LED_GALBEN, HIGH);
            digitalWrite(LED_VERDE, LOW);
            delay(1000);
        }
        else {
            digitalWrite(LED_GALBEN, LOW);
            digitalWrite(LED_VERDE, HIGH);
            delay(1000);
        }
    }
}

void loop()
{
    // Masor distanta pana la mana
    reset_trigger(TRIGGER_PIN_DIST);
    float dist = get_distance(ECHO_PIN_DIST);

    // Masor distanta in interiorul cosului
    // pana la gunoi
    reset_trigger(TRIGGER_PIN_FULL);
    float dist_full = get_distance(ECHO_PIN_FULL);

    // Printez distanta pana la gunoi pe seriala
    Serial.println(dist_full);

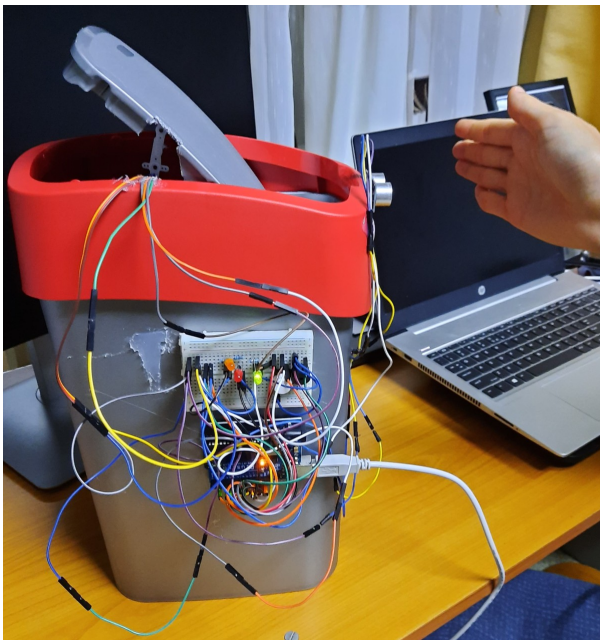
    // Daca s-a actionat manual deschiderea cosului
```

```
open_trash_by_button();  
  
// Deschidere cos declansata de mana aflata in fata capacului  
open_trash_by_hand(dist);  
  
// Verificare umplere cos  
check_content(dist_full);  
}
```

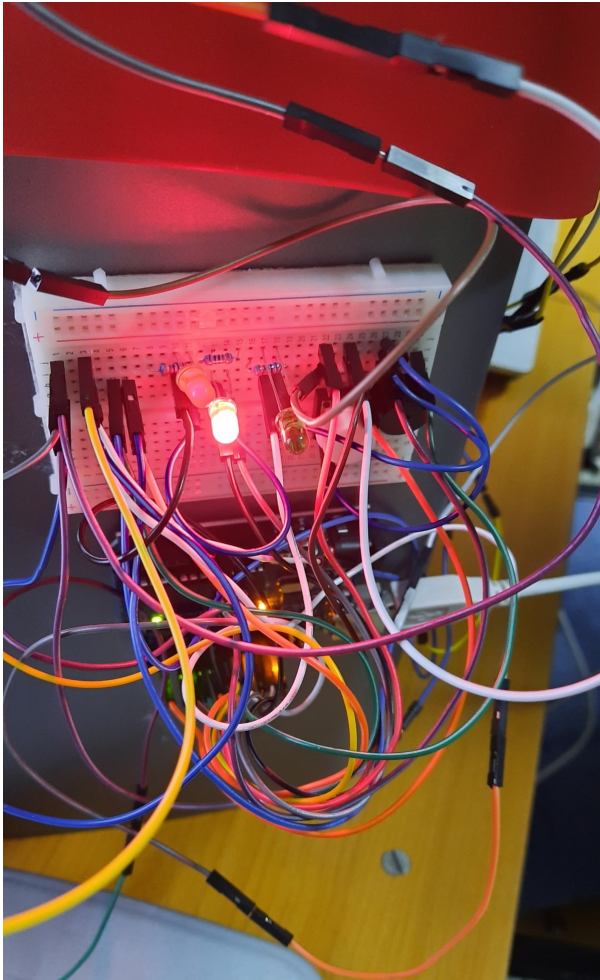
Rezultate Obținute

Rezultatele obținute în urma realizării proiectului

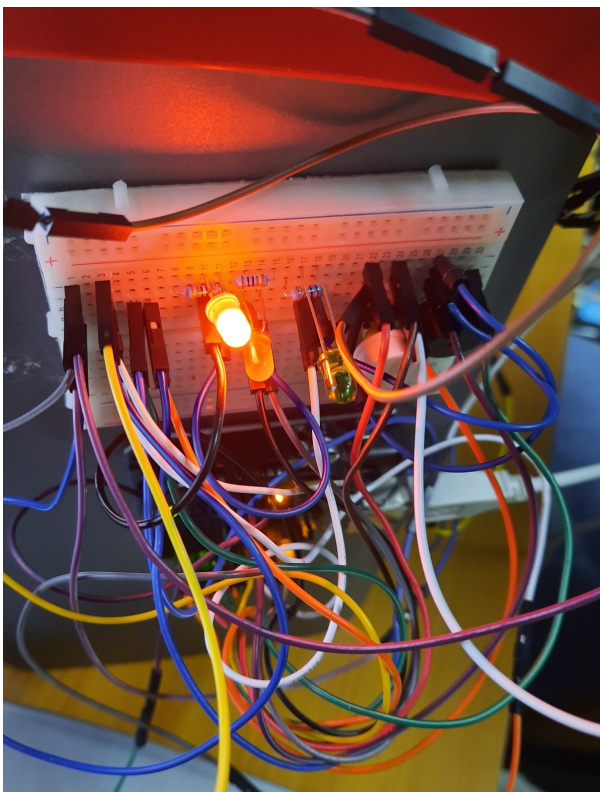
Montaj (mana la o distanta de aproximativ 8cm de senzor, capacul se deschide). Continutul cosului este gol, este aprins ledul verde.



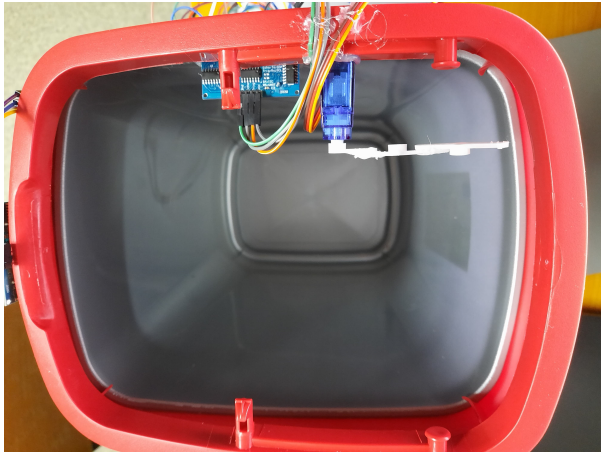
Cosul este plin, este aprins ledul rosu si declansat buzzerul.



Cosul este semi-plin, este aprins ledul portocaliu.



Montaj (in interior) pentru declansare capac si detectare continut.



Demo proiect: [demo](#)

Concluzii

Cosul se deschide cand detecteaza o apropiere de senzorul sau de distanta. Poate fi folosit atat automat, cat si manual, actionand butonul de pe breadboard. Poate fi foarte util pentru a detecta cand trebuie sa duci gunoiul. Pot aparea probleme daca nu se mentine mana in proximitatea senzorului o perioada indeajuns de lunga, pentru a o detecta. Acest proiect mi-a aratat ca este destul de usor sa lucrezi cu Arduino si cu ajutorul lui poti implementa chiar obiecte "inteligente" pe care le poti folosi mai apoi in viata de zi cu zi.

Download

Arhiva contine un scurt *video* cu prezentarea functionalitatilor, *codul sursa*, un scurt fisier *README*, *diagrama electrica* si *schema bloc* :

[334ca_dragnelaviniastefana_proiectpm.zip](#)

Jurnal

Secțiune de jurnal

25.04.2022 - alegere tema proiect

04.05.2022 - comanda piese de pe <https://www.optimusdigital.ro/ro/>

09.05.2022 - au ajuns piesele :)

12.05.2022 - realizare prima parte din proiect (conectare servo-motor pentru deschidere capac si senzor de distanta) (am folosit o schema de Tinkercad dupa care m-am ghidat)

16.05.2022 - conectare led-uri si al 2-lea senzor de distanta (am descoperit ca mai am nevoie de fire : ()

17.05.2022 - comanda fire T-M

20.05.2022 - realizare proiect final (adaugare o noua functionalitate: deschiderea bazata pe buton)

21.05.2022 - testare

23.05.2022 - prezentare proiect la laborator

Bibliografie/Resurse

Export to PDF

<https://www.optimusdigital.ro/ro/>

<https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/>

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

<https://www.instructables.com/Arduino-Servo-Motors/>

<https://ocw.cs.pub.ro/courses/icalc>

Testare pe: <https://www.tinkercad.com/>

Schema bloc: <https://app.diagrams.net/>

Diagrama electrica: Eagle

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/ncaroi/smartrash>



Last update: **2022/05/26 21:56**