Smart Dustbin

Introducere

- Acest proiect consta intr-un cos de gunoi inteligent care se deschide automat pentru a arunca deseurile si totodata semnaleaza cand trebuie schimbat.
- Cosul inteligent ofera mai multe beneficii si solutii pentru gestionarea eficienta a deseurilor, precum: instiintarea utilizatorului de nivelul de umplere al cosului, dar si semnalare audio cand acesta trebuie schimbat, asigura salubritatea mediului prin evitarea atigerii deseurilor si totodata simplifca procesul de administrare a reziduurilor prin automatizare.
- Ideea de la care am pornit a fost dorinta de a asigura utilizatorului un nivel mai ridicat de confort si siguranta in gestionarea deseurilor intrucat se mentine igiena mediului.

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât şi hardware însoţită de o descriere a acestora precum şi a modului în care interacţionează.

- Cosul de gunoi asigura o deschidere automata a capacului daca mana utilizatorului e afla la o distanta mai mica de 20 cm, functionalitate implementata cu ajutorul unui senzor ultrasonic de masurare a distantei si un servomotor care este conectat la capacul cosului.
- Totodata, am utilizat un alt senzor ultrasonic ce masoara nivelul reziduurilor din cos si il va printa pe LCD pentru a informa utilizatorul.
- Cand nivelul deseurilor ajunge la 100%, un buzzer va atentiona utilizatorul ca trebuie sa goleasca cosul. Daca acesta incearca sa adauge alt

deseu, cosul nu se va deschide si va porni alarma pentru 5 secunde.

- Alarma se dezactiveaza definitiv atunci cand cosul este golit.

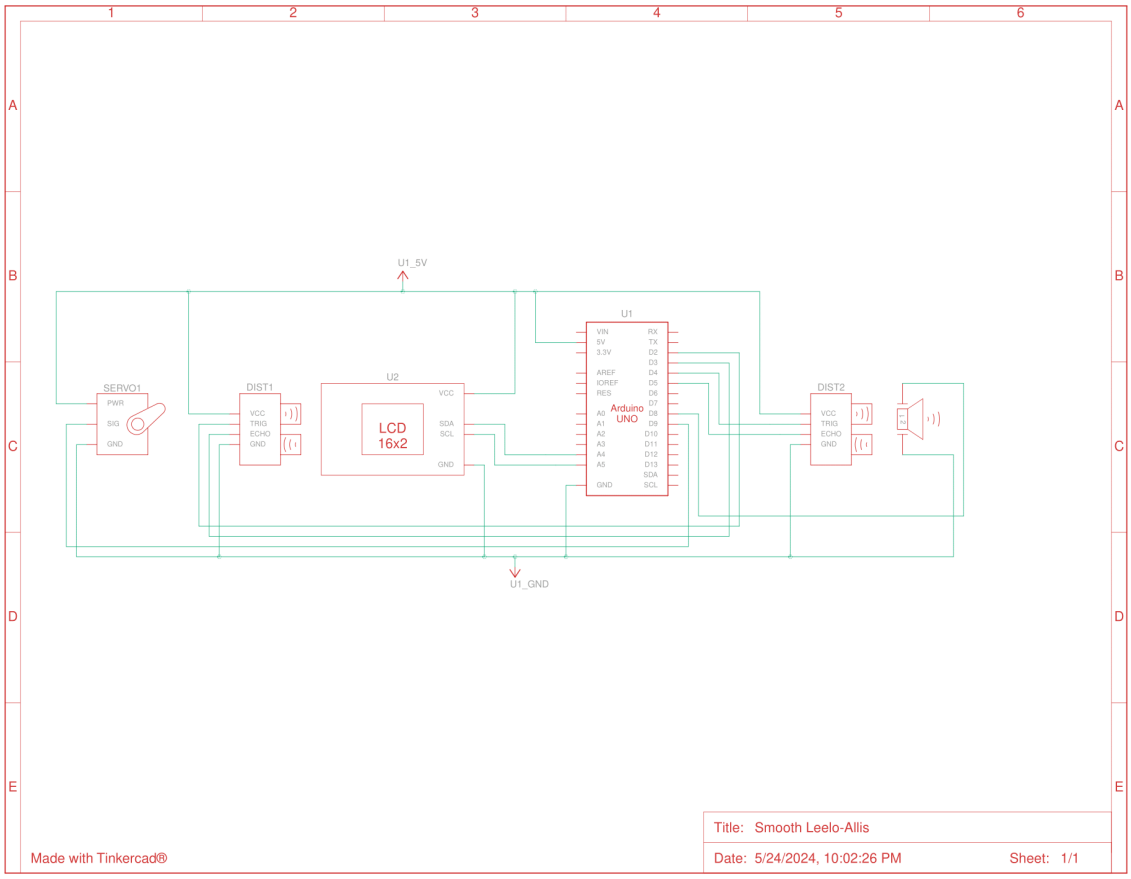
Schema bloc:

Hardware Design

Aici puneţi tot ce ţine de hardware design:

- listă de piese
- scheme electrice (se pot lua şi de pe Internet şi din datasheet-uri, e.g. http://www.captain.at/electronic-atmega16-mmc-schematic.png)
- diagrame de semnal
- rezultatele simulării

```
Lista de piese:
 1. Senzor Ultrasonic x2
 2. Servomotor
 3. Plusivo (compatibil Arduino Uno)
 4. Buzzer
 5. LCD 1602 cu modul I2C
```



Software Design

Descrierea codului aplicaţiei (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării şi surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi şi structuri pe care plănuiţi să le implementaţi
- (etapa 3) surse şi funcţii implementate

- Mediul de dezvoltare folosit este Arduino IDE, Version: 2.3.2.
- Librarii utilizate:
  - Wire.h → este utilizata pentru a comunica cu display-ul LCD care foloseste un modul I2C
  - LiquidCrystal_I2C.h → permite controlul unui display LCD printr-un convertor I2C
  - PWMServo.h → pentru a controla servomotorul
  - avr/interrupt.h → pentru a gestiona intreruperile si permite activarea si dezactivarea acestora

Configurarea pinilor - descrisa prin MACRO-uri

```
#define TRIG_PIN1 2
#define ECHO_PIN1 3
#define TRIG_PIN2 4
#define ECHO_PIN2 5
#define BUZZER_PIN 9
#define SERVO_PIN 8
```

Conectarea la placuta plusivo a componentelor utilizate: servomotor, senzori ultrasonici si buzzer.

Initializarea componentelor

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
PWMServo myServo;
```

Se initializeaza obiectele pentru afisajul LCD si servomotor.

Configurarea timerului pentru generarea de intreruperi

```
 cli(); // Dezactiveaza intreruperile

  // Seteaza modul timerului la CTC (Clear Timer on Compare Match)
  TCCR0A = (1 << WGM01); // Mod CTC cu OCR0A ca TOP

  // Seteaza valoarea de comparare pentru 0.15 secunde la 16 MHz
  OCR0A = 234; // 16e6 / (1024 * 0.15) - 1

  // Activeaza intreruperea de comparare A a timerului
  TIMSK0 = (1 << OCIE0A);
```

```
  // Porneste temporizatorul cu prescaler 1024
  TCCR0B = (1 << CS02) | (1 << CS00); // Prescaler 1024

  sei(); // Activeaza intreruperile
```

Timer0 este configurat pentru a genera o intrerupere la fiecare 0,15 secunde pentru a masura distanta folosind senzorii ultrasonici. Folosirea unui timer genereaza un consum optim de energie, calculul distantelor fiind mai rar realizat.

Functia principala - LOOP

```
void loop() {
  unsigned long currentMillis = millis(); // Obtine timpul curent in
milisecunde
  int currentDistance1;

  cli(); // Dezactiveaza intreruperile
  currentDistance1 = distance1; // Copiaza valoarea distance1 calculata
  // in cadrul intreruperii
  sei(); // Activeaza intreruperile

  // Gestioneaza miscarea servo-ului si alarma in bucla principala
  if (wasteLevel == 100 && currentDistance1 < 20 && servoState == 0) {
    alarm = true; // Activeaza alarma daca nivelul deseurilor este 100 si
distanta este mai mica de 20 cm
  } else if (currentDistance1 < 20) {
    if (servoState == 0 && servoCounter <= 0) {
      myServo.write(10); // Roteste servo motorul la 10 grade
      servoState++; // Creste starea servo-ului
      servoCounter = 0; // Reseteaza contorul servo
    } else if (servoState == 1 && servoCounter >= 50) {
      myServo.write(130); // Returneaza servo motorul la pozitia initiala
      servoState = 0; // Reseteaza starea servo-ului
    }
  }

  if (wasteLevel < 100) {
    alarm = false; // Dezactiveaza alarma daca nivelul deseurilor este sub
100
  }

  // Activeaza alarma daca nivelul deseurilor este 100
 if (alarm)
    tone(BUZZER_PIN, 1000); // Start the buzzer
  } else {
    noTone(BUZZER_PIN);
  }

  // Afiseaza nivelul de deseuri pe LCD
  lcd.setCursor(0, 1);
```

```
  lcd.print("    ");
  lcd.setCursor(0, 1);
  lcd.print(wasteLevel);
}
```

Aceasta descrie logica de control a servomotorului si a buzzerului. La fiecare 0,15 secunde se genereaza o interupere **TIMER0_COMPA** de tip Timer/Counter0 Compare Match A care face o recalculare a celor 2 distante folosite, distanta dintre cos si mana utilizatorului si cea dintre senzor si gunoi. Apoi, in functie de noile distante, se verifica daca este necesara deschiderii capacului, adica modificarea pozitiei servomotorului.

- Daca nivelul deseurilor (wasteLevel) este 100 si distanta masurata de senzorul 1 (currentDistance1) este mai mica de 20 cm si servo-ul nu a fost miscat (servoState == 0), se seteaza alarma la true.
- Daca distanta masurata de senzorul 1 este mai mica de 20 cm:
  - Daca servo-ul nu a fost miscat (servoState == 0) si contorul servo-ului este ⇐ 0, se misca servo-ul la 10 grade, se incrementeaza starea servo-ului si se reseteaza contorul.
  - Daca servo-ul a fost miscat (servoState == 1) si contorul servo-ului este >= 50 (aproximativ 5 secunde), se reintoarce servo-ul la pozitia initiala si se reseteaza starea.

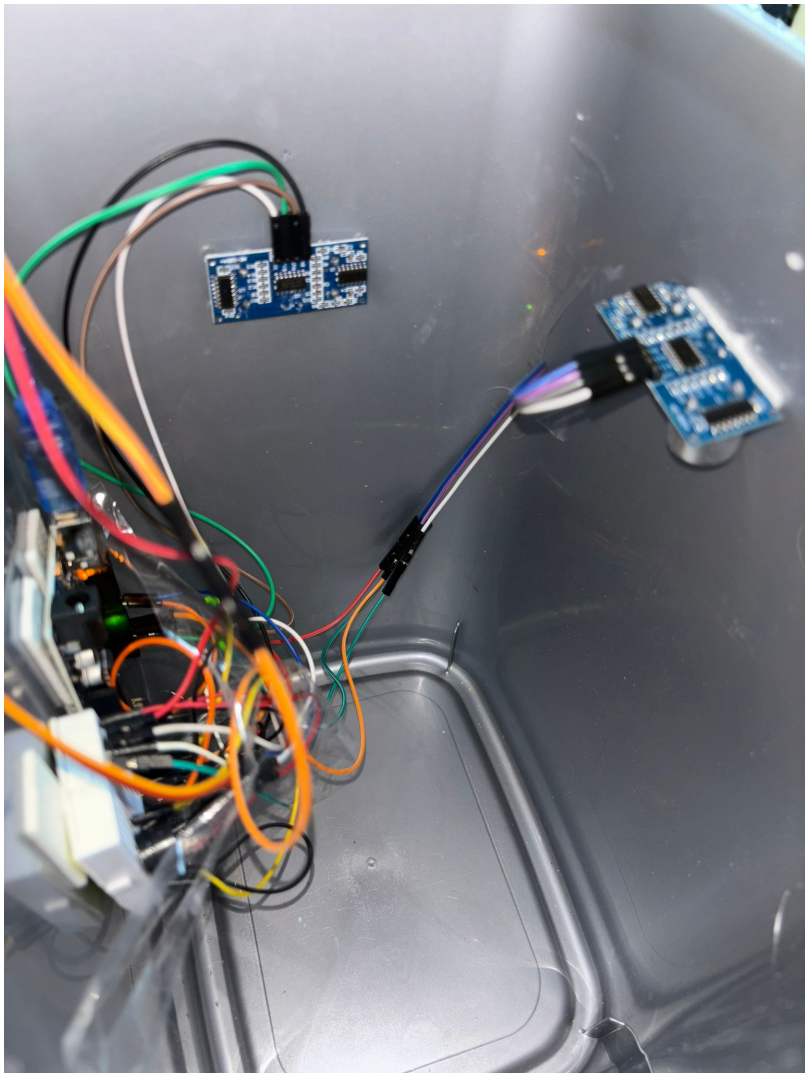
```
ISR(TIMER0_COMPA_vect) {
  // Recalculeaza distantele la fiecare 0.15 secunde
  unsigned long currentMillis = millis(); // Obtine timpul curent in
milisecunde

  // Verifica primul senzor ultrasonic
  digitalWrite(TRIG_PIN1, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN1, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN1, LOW);
  distance1 = pulseIn(ECHO_PIN1, HIGH) * 0.034 / 2; // Calculeaza distanta
in cm

  // Verifica al doilea senzor ultrasonic
  digitalWrite(TRIG_PIN2, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN2, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN2, LOW);
  distance2 = pulseIn(ECHO_PIN2, HIGH) * 0.034 / 2; // Calculeaza distanta
in cm

  // Mapeaza distanta la nivelul de deseuri (100 cand cosul este plin, 0
cand este gol)
  wasteLevel = map(distance2, 5, 20, 100, 0); // Mapeaza distanta (5-20 cm)
la nivelul de deseuri (100-0)
  wasteLevel = constrain(wasteLevel, 0, 100); // Asigura ca wasteLevel
ramane in intervalul 0-100

  // Incrementeaza contorul servo daca servo-ul este in pozitia 1
  if (servoState == 1 && servoCounter < 50) {
```

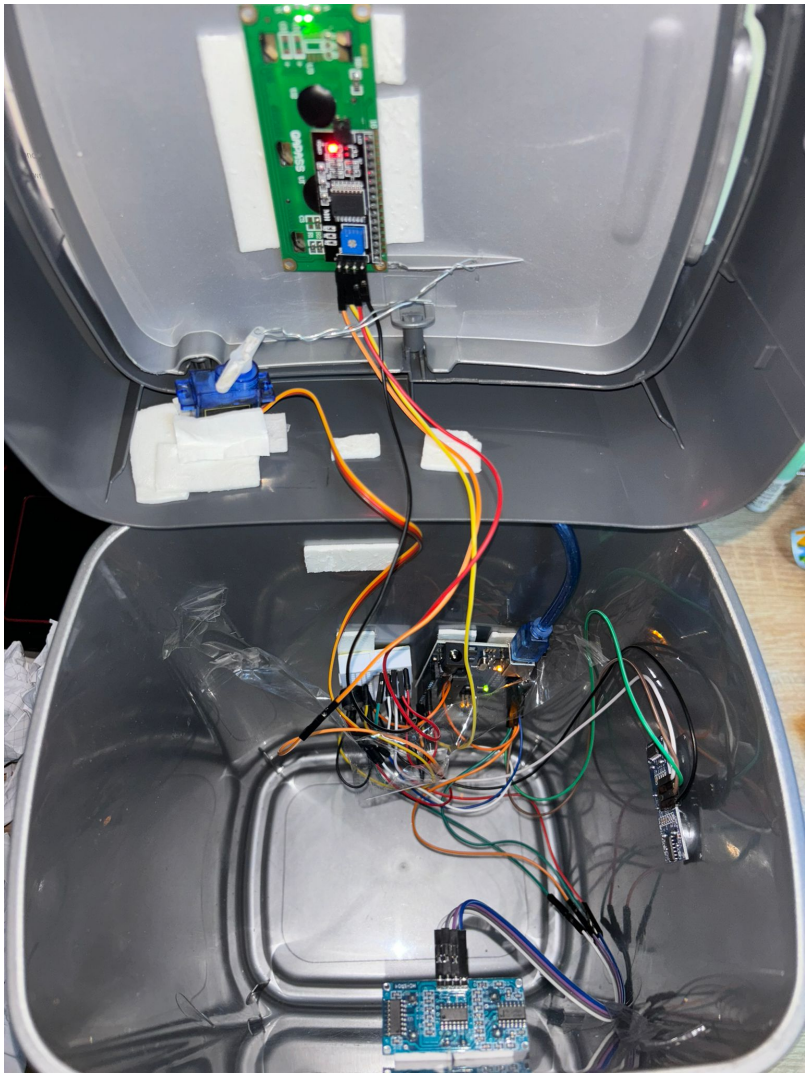
```
    servoCounter++;
  }

  // Decrementeaza contorul servo daca servo-ul este in pozitia 0
  if (servoState == 0 && servoCounter > 0) {
    servoCounter--;
  }
}
```

Rezultate Obţinute

Care au fost rezultatele obţinute în urma realizării proiectului vostru.

Video care prezinta functionalitatea proiectului





Concluzii

- Proiectul realizat doreste automatizarea procesului de strangere a deseurilor, lucru realizat

prin deschiderea capacului cu ajutorul unui senzor ultrasonic care sa activeze servomotorul daca distanta este mai mica decat 20. Totodata, acesta aduce la cunostinta utilizatorului ca trebuie sa arunce gunoiul prin declansarea unei alarme cand cosul este plin.

Download

comsavioleta_proiectpm.zip

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

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

Export to PDF