

Smart Recycle Bin

Autor: Sâmpetru Mario
Grupa: 331CA

Introducere

Am realizat un coș de gunoi inteligent care detectează apropierea utilizatorului prin intermediul unui senzor ultrasonic, deschide automat capacul folosind un servo. În plus, capacul acestuia se poate deschide de la distanță prin intermediul unei aplicații. Tot prin intermediul aplicației, utilizatorul poate verifica dacă a avut loc o scurgere de lichide.

Descriere generală

1. Arduino UNO

Deschide capacul coșului de gunoi și reacționează automat la apropierea utilizatorului.

Conectat la:

- **Servo motor** - acționează capacul pentru deschidere/închidere.
- **Conexiune serială cu ESP32** - primește comenzi de la ESP32 pentru control de la distanță.
- **SL067** - detectează nivelul de apă și anunță utilizatorul prin intermediul aplicației.
- **Buzzer** - Trimite alertă sonoră când senzorul de apă detectează o valoare peste prag.
- **LCD** - Afișează statusul capacului (deschis/închis) și valoarea citită de la senzorul de apă.

2. ESP32-WROOM

Comunicare wireless și extinderea funcționalității prin senzori suplimentari.

Conectat la:

- **HC-SR04** - detectează apropierea utilizatorului pentru deschiderea automată.
- **Conexiune Wi-Fi** - permite controlul coșului de la distanță printr-o interfață grafică.
- **Conexiune serială cu Arduino UNO**

3. Interfață Grafică (Python)

Descriere:

- Permite introducerea adresei IP a ESP32.

- Trimite comenzi HTTP (**/open**, **/close**, **/status**) către ESP32 pentru deschiderea sau închiderea capacului, și pentru a verifica dacă a avut loc un incident ce a cauzat scurgere de lichide în interiorul coșului.

Schema Bloc

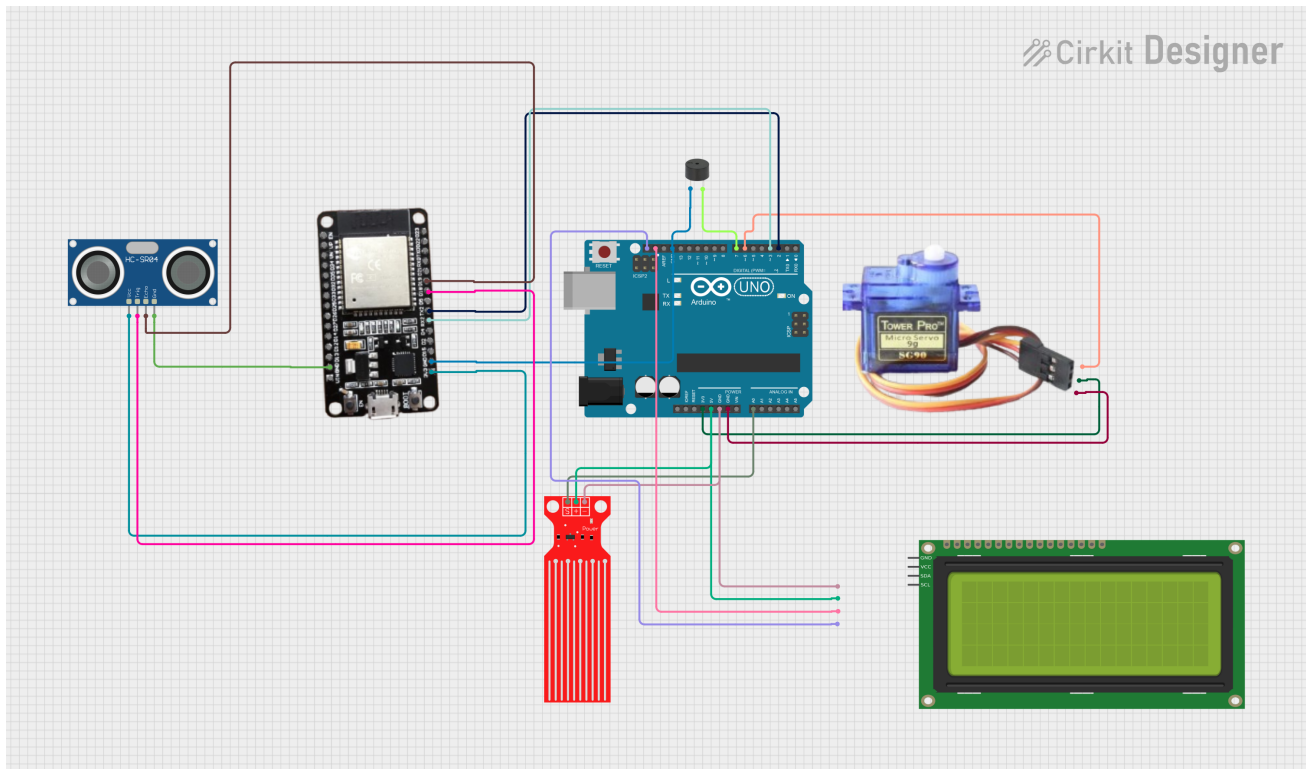


Hardware Design

Listă piese:

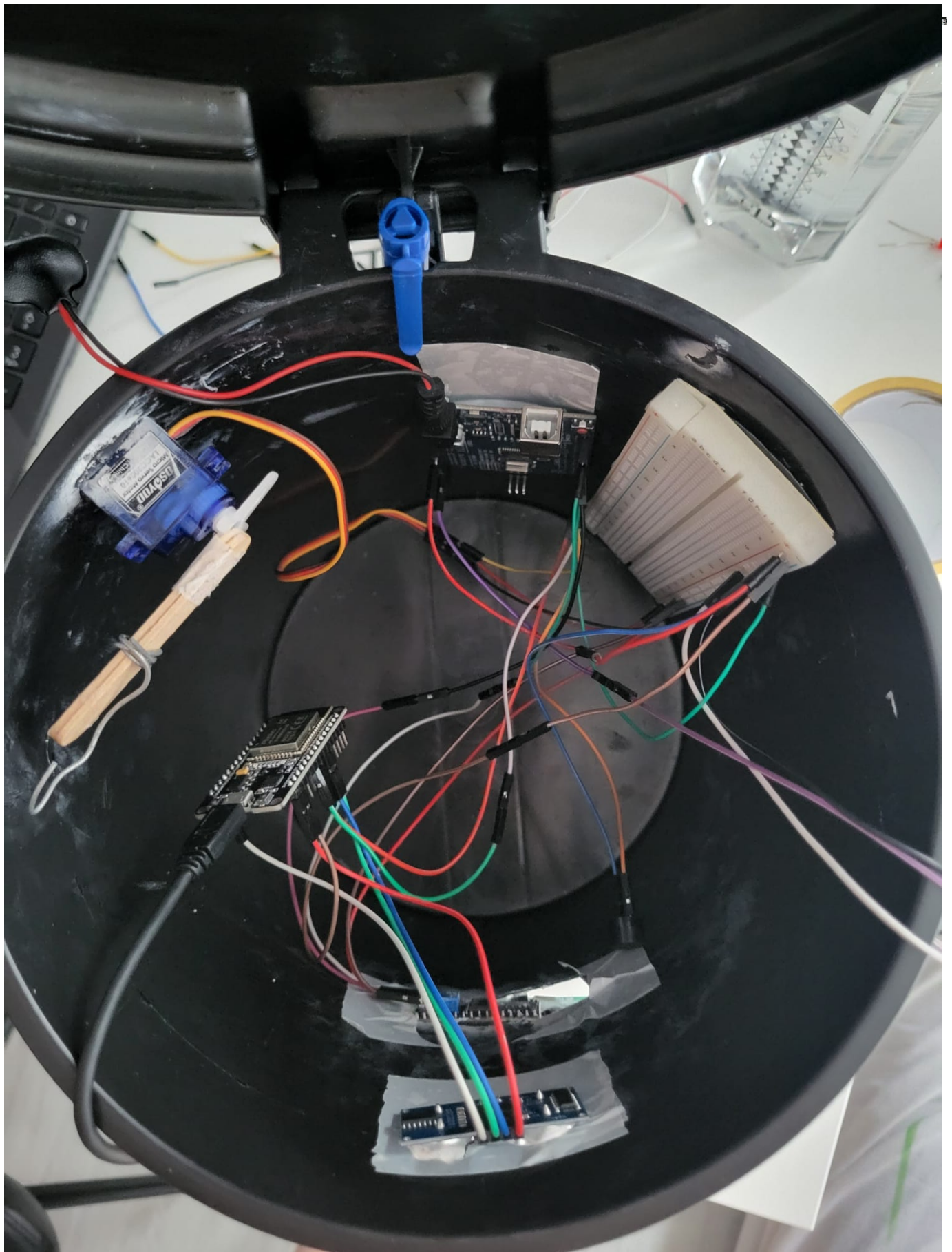
Componentă	Cantitate	Preț
Arduino UNO	1	49 Lei
ESP-Wroom 32	1	45 Lei
Servomotor SG90	1	15 Lei
SL067 (Water Sensor)	1	10 Lei
HC-SR04	1	12 Lei
LCD I2C	1	14 Lei
Buzzer	1	2 Lei
Baterie 9V 650mAh	1	57 Lei
Baterie 5V 2Ah	1	100 Lei

Schema circuitului



Imagini cu Hardware-ul





Software Design

Medii de dezvoltare: Arduino IDE pentru ESP32 și Arduino UNO; Vscod pentru Python

Listă Librării și Surse Terțe Utilizate

Nume Librărie	Descriere și Utilizare
<i>tkinter</i>	Bibliotecă GUI nativă Python. A fost folosită pentru a crea interfața grafică din care se trimit comenzile (deschidere/închidere capac) către ESP32.
<i>requests</i>	Modulul HTTP pentru Python, folosit în aplicația GUI pentru a trimite cereri GET către ESP32 și a comunica cu dispozitivul.
<i>WiFi.h</i>	Oferă conectivitate Wi-Fi și gestionează serverul HTTP care primește comenzile de la interfața Python.
<i>soc/gpio_reg.h</i>	oferă definiții pentru registrele GPIO, facilitând controlul pinilor ESP32.
<i>LiquidCrystal_I2C.h</i>	Pentru controlul display-ului LCD.
<i>Servo.h</i>	A fost utilizată pentru a controla mișcarea servo-ului (deschiderea/închiderea capacului).

Funcții Arduino UNO

Funcție	Descriere
<code>setupServo()</code>	Configurează servo-motorul prin intermediul librăriei <code>Servo.h</code>
<code>openServo()</code>	Modifică unghiul de deschidere al servo-ului astfel încât să poată efectua deschiderea capacului.
<code>closeServo()</code>	Operația inversă <code>openServo()</code> .
<code>updateLcd()</code>	Actualizează afișajul LCD folosind librăria <code>LiquidCrystal_I2C</code> . Mesajele afișate reprezintă statusul capacului (deschis/închis), precum și valoarea citită de senzorul de apă.
<code>checkWaterSensor()</code>	Citește valoarea analogică de la senzorul de apă și trimite o alertă dacă valoarea depășește pragul.
<code>handleEspCommand()</code>	Procesează comenzile primite (de ex. OPEN sau CLOSE) de la ESP32 pentru a controla deschiderea/închiderea servo-ului.
<code>autoCloseCheck()</code>	Închide automat servo-ul după 3000 ms dacă acesta este deschis.

Funcții ESP32

Funcție	Descriere
<code>setupWiFi()</code>	Conectează ESP32 la rețeaua Wi-Fi folosind SSID și parolă, așteptând obținerea unei conexiuni stabile.
<code>measureDistance()</code>	Utilizează senzorul HC-SR04 pentru a măsura distanța dintre senzorul <i>HC-SR04</i> și cel mai apropiat obiect.
<code>handleHandDetection()</code>	Detectează prezența unei mâini la distanțe între 2.0 și 15.0 cm și declanșează comanda de deschidere a capacului când pragul este atins.

<code>handleSerialInput()</code>	Monitorizează conexiunea serială (Serial2) pentru a prelua mesajele trimise de la Arduino. La primirea unui mesaj, acesta este afișat pe consolă și transmis funcției <code>updateStatus()</code> pentru eventuale acțiuni ulterioare.
<code>sendToArduino(String& cmd)</code>	Trimite comenzi (cum ar fi "OPEN" sau "CLOSE") către placa Arduino prin intermediul conexiunii Serial2.
<code>handleHttpRequest()</code>	Gestionează cererile HTTP GET pentru <code>/open</code> , <code>/close</code> și <code>/status</code> , permițând controlul de la distanță al funcțiilor dispozitivului.
<code>updateStatus()</code>	Procesează mesajele primite de la Arduino pentru a actualiza starea sistemului.

Funcții Python

Funcție	Descriere
<code>send()</code>	Trimite cereri GET către ESP32 (cu timeout de 2 secunde) pentru a executa comenzi precum deschiderea sau închiderea capacului.
<code>poll_status()</code>	Interoghează periodic ESP32 pentru a verifica starea și a afișa avertismente dacă se detectează WATER.

Rezultate Obținute

- **Interfață GUI Python:** Aplicația oferă o interfață prietenoasă, din care se pot trimite rapid comenzi către ESP32, facilitând controlul dispozitivului.
- **Conectivitate Wi-Fi și Server HTTP:** ESP32 se conectează cu succes la rețeaua Wi-Fi, iar serverul HTTP integrat răspunde corect la cereri GET (`/open`, `/close` și `/status`), permițând controlul la distanță.
- **Măsurarea Distanței:** Sensorul HC-SR04 a măsurat cu precizie distanțele, permițând detectarea prezenței unei mâini între 2.0 și 15.0 cm, activând comanda de deschidere.
- **Comunicare între ESP32 și Arduino Uno**
- **Gestionarea Stării de Apă:** Arduino Uno monitorizează sensorul de apă și transmite alerta "WATER" atunci când valoarea măsurată depășește pragul.

Jurnal

- M-am gândit la proiect.
- Am procurat piesele necesare. Până când au venit piesele, am lucrat la schema bloc și la schema electrică.
- Odată ajunse piesele, am realizat etapa hardware, cu piesele conectate exact la fel ca în schema electrică.
- Am lipit piesele în interiorul coșului de gunoi și am mai cumparat o baterie externa pentru a alimenta complet proiectul.
- Am scris Software-ul pentru proiect.

Concluzii

- Integrarea hardware-software a fost reușită, demonstrând control precis al servo-ului și monitorizarea senzorilor prin comunicare eficientă (HTTP, HardwareSerial, SoftwareSerial) între GUI (Python), ESP32 și Arduino Uno.
- Conectivitatea Wi-Fi a fost stabilă, serverul HTTP răspunzând corect cererilor, permițând controlul la distanță.
- Proiectul este scalabil, având potențial pentru îmbunătățiri și adăugiri viitoare.

Download

- **Codul:** <https://github.com/mariosam23/SmartRecycleBin>
- **Demo Video 1:** <https://github.com/mariosam23/SmartRecycleBin/blob/main/demo1.gif>
- **Demo Video 2:** <https://github.com/mariosam23/SmartRecycleBin/blob/main/demo2.gif>
- **Codul arhivat:** [smart_r_b_code.zip](#)

Bibliografie/Resurse

Componentă	Cantitate	Datasheet
Arduino UNO	1	Arduino UNO
ESP-Wroom 32	1	ESP-Wroom 32
Servomotor SG90	1	Servomotor SG90
SL067	1	Water Sensor SL067
HC-SR04	1	HC-SR04
LCD I2C	1	LCD

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/mario.sampetru>



Last update: **2025/05/27 20:35**