

# Dozator bauturi

## Introducere

Deoarece mi-am dorit să folosesc în viața de zi cu zi proiectul pe care îl fac, m-am decis să construiesc un dozator de băuturi.

Dozatorul va umple paharul automat atunci când se detectează că acesta e în poziția corectă și se va opri atunci când paharul este plin.

Cred că este util pentru toată lumea deoarece este mai confortabil să îți umple paharul cu precizie un dozator decât să îți torni singur băutura în pahar. De asemenea, ar trebui să prevină și accidentele de neîndemânare (de exemplu, răsturnatul paharului atunci când vrem să îl umplem).

## Descriere generală



Atunci când se apasă pe buton, senzorul ultrasonic va trimite către Arduino distanța la care se află paharul, și se va verifica dacă acesta este pus în poziție (adică dacă se află într-o distanță limită față de senzor). Dacă se află în poziție ok, se va afișa pe ecran un mesaj cum că dozatorul începe să toarne băutura și plăcuța Arduino va porni pompa de apă pentru un timp definit ( volumul paharului / debitul pompei) folosind un timer. După ce paharul a fost umplut, pe ecran se va afișa un mesaj sugestiv.

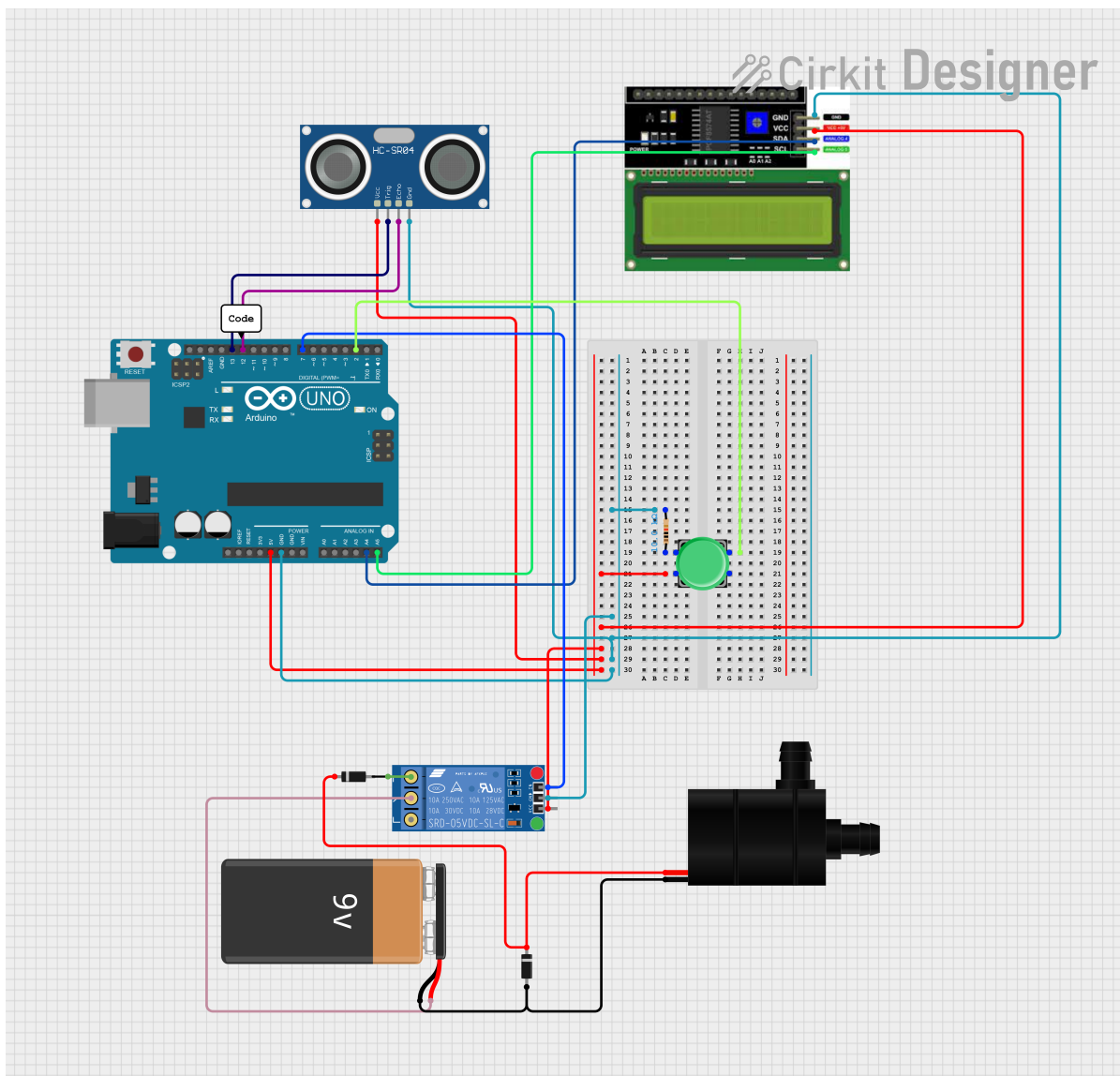
## Hardware Design

### Lista de piese

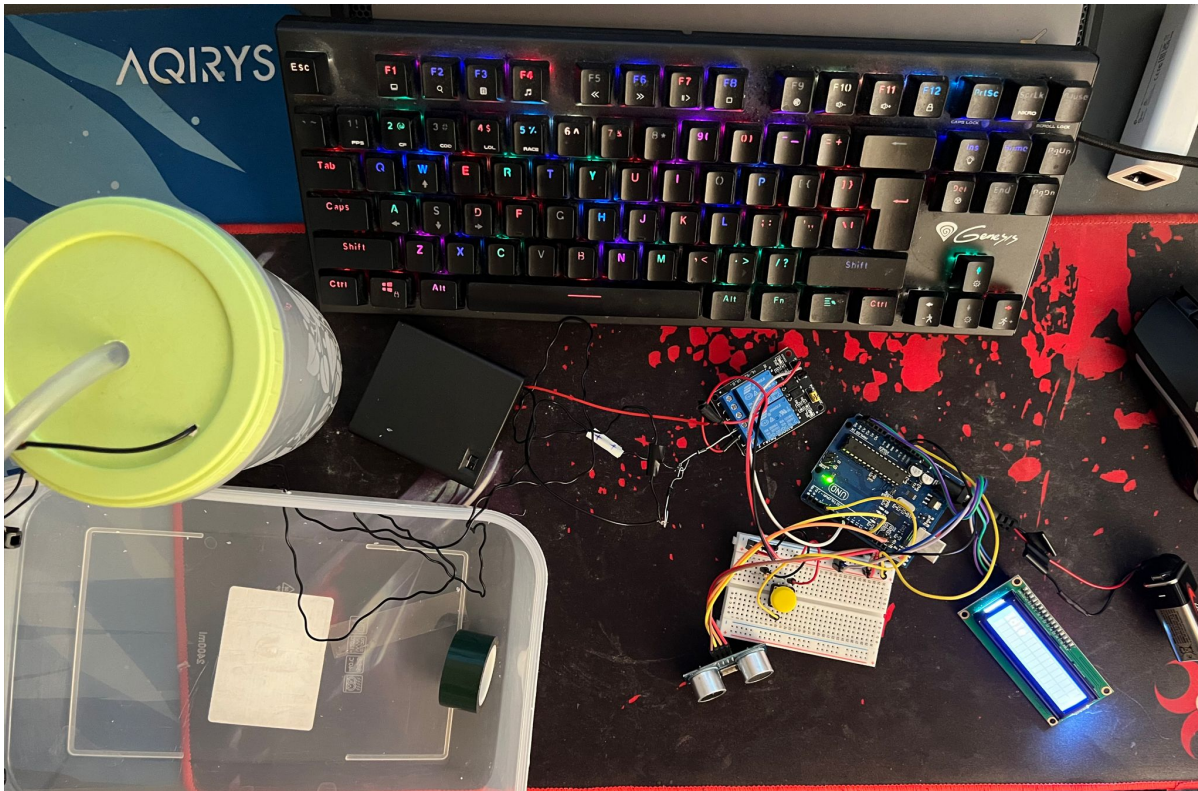
Componentă	Cantitate	Descriere
Plăcuță Arduino Uno R3	1	Placuta care controleaza celelalte piese
Ecran LCD 1602 I2C	1	Afișează mesaje sugestive legate de procesul de umplere al paharului
Buton	1	Declanșează începerea umplerii paharului
Rezistor 1.8KΩ	1	Sa functioneze corect butonul
Pompă de apă 3-6V	1	Responsabila pt a transfera apă din sticlă în pahar
Breadboard 400 puncte	1	Rol de conectare a pieselor
Fire jumper	x	Rol de conectare a pieselor
Modul releu 1 canal	1	Ajută la conectarea pompei de apă la Arduino și la baterie

Senzor ultrasonic HC-SR04	1	Detectează dacă există un pahar în care să se toarne apă
Dioda 1N4001	2	Asigura trecerea curentului într-o singura direcție și previne zgomotul produs de pompa
Socket baterii 4xAA	1	Alimentează pompa de apă
Socket baterie 9v	1	Alimentează placa Arduino

### Schemă Electrică



### Montaj (nearanjat in carcasa)



## Alegerea pinilor

- Pentru alimentare am conectat pinii de 5V si GND la breadboard pentru a putea conecta toate componentele de care am nevoie.
- Pentru toate piesele am conectat alimentarea pe breadboard VCC la + si GND la -.
- LCD I2C: am conectat cei 2 pini SDA (linia de date pentru comunicatia I2C) si SCL (linia de ceas pentru comunicatia I2C) la pinii analog A4 si A5 pe Arduino.
- Senzor Ultrasonic: am conectat Trig si Echo la pinii digitali D13 si D12 pe Arduino.
- Butonul functioneaza cu intreruperi deci trebuie conectat la un pin digital care suporta (D2 sau D3) asa ca am ales sa il conectez la D2.
- Releul il activez cu un semnal digital asa ca l-am conectat la un pin digital D7. La pinul NO (Normally Open) am conectat +ul bateriei, la pinul COM (Common) am conectat +ul pompei de apa apoi am conectat -ul pompei cu -ul bateriei

## Software Design

- Mediu de dezvoltare ARDUINO IDE
- Librării și surse 3rd-party: Wire.h și LiquidCrystal\_I2C.h pentru comunicație I2C și afișarea pe ecran LCD.

COD SURSA

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
initializeaza ecranul lcd LiquidCrystal_I2C lcd(0x27, 16, 2); declarare pini componente const int  
trigPin = 13; const int echoPin = 12; const int relayPin = 7; const int buttonPin = 2;
```

```
declarare variabile ajutatoare long duration; int distance; int buttonState = 0; void setup() {  
initializare lcd
```

```
lcd.init();  
lcd.backlight();
```

```
Serial.begin(9600);
```

```
// initializare senzor ultrasonic  
pinMode(trigPin, OUTPUT);  
pinMode(echoPin, INPUT);
```

```
// initializare releu  
pinMode(relayPin, OUTPUT);  
digitalWrite(relayPin, HIGH);
```

```
// initializare buton  
pinMode(buttonPin, INPUT);
```

```
}
```

```
void loop() {
```

```
// citeste state-ul butonului  
buttonState = digitalRead(buttonPin);
```

```
// daca se apasa pe buton  
if (buttonState == HIGH) {
```

```
    // se calculeaza distanta la care se afla paharul  
    // cu ajutorul senzorului ultrasonic  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(2);
```

```
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);
```

```
    duration = pulseIn(echoPin, HIGH);
```

```
    distance = duration * 0.034 / 2;
```

```
    // afiseaza distanta pe lcd  
    lcd.clear();  
    lcd.setCursor(0, 0);
```

```
lcd.print("Distance: ");  
lcd.print(distance);  
lcd.print(" cm");
```

```
Serial.print("Distance: ");  
Serial.print(distance);  
Serial.println(" cm");
```

```
// verifica daca paharul se afla in distanta dorita  
if (distance <= 5) {  
    // activeaza releul (adica porneste pompa)  
    digitalWrite(relayPin, LOW);
```

```
    // afiseaza un mesaj sugestiv pe ecran  
    lcd.setCursor(0, 1);  
    lcd.print("se umple");
```

```
    // timer pana se umple un pahar (6s)  
    delay(6000);
```

```
    // dupa trecerea timpului se dezactiveaza releul  
    digitalWrite(relayPin, HIGH);  
} else {
```

```
    // daca nu se afla pahar in distanta dorita  
    // se afiseaza un mesaj  
    lcd.setCursor(0, 1);  
    lcd.print("nu umple");  
    delay(1000);  
}  
    delay(500);  
} else {  
    // daca nu se apasa pe buton se afiseaza pe ecran  
    // un mesaj de idle  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Idle");  
    delay(500);  
}  
  
}
```

## Rezultate Obținute

## Concluzii

A fost interesant sa realizez proiectul. Am invatat cum sa conectez diferite componente hardware intre ele si cum sa le controlez. Mi-am dat seama ca un astfel de proiect trebuie planificat foarte bine si ca piesele trebuie atent studiate deoarece pare ca fac acelasi lucru si sunt asemanatoare, de fapt, nu e chiar asa.

## Download

[Download archive](#)

## Bibliografie/Resurse

RESURSE:

- <https://www.handsontec.com/dataspecs/HC-SR04-Ultrasonic.pdf>
- <https://www.optimusdigital.ro/ro/electronica-de-putere-module-cu-releu/39-modul-cu-releu-cu-doua-canale.html>
- <https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/62-lcd-1602-cu-interfata-i2c-si-backlight-galben-verde.html>
- <https://roboticsbackend.com/arduino-push-button-tutorial/>
- <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/robert.petre0406>



Last update: **2024/05/27 11:29**