

# Cocktail Maker

## Introducere

Aparatul creat se va ocupa de turnarea ingredientelor unui cocktail într-un pahar, când i se da comanda, controlând totodata cantitățile. Scopul acestuia este ca oamenii sa petreacă mai puțin timp făcând băuturi și sa fie mai mult prezenți împreună.

Ideea pentru acest proiect a apărut fiind invitat la o petrecere si observând ca oamenii petreceau foarte mult timp făcând cocktail-uri; acest proiect putând fi de ajutor.

Pe lângă faptul ca in cadrul unei petreceri este foarte util tuturor invitaților; acest aparat sare si in ajutorul angajaților din bar, putând fi mult mai eficienți.

## Descriere generală

Folosind un Arduino împreună cu un senzor de atingere, voi putea seta cantitățile dorite pentru fiecare ingredient. La fiecare ingredient setat împreună cu măsurătorile dorite, valorile fiind afișate pe ecranul LED, plăcuta va trimite o comanda către unul dintre modulele cu driver pentru a activa pompa dorită, sursa de alimentare contribuind cu necesarul de curent. Din timp in timp, Arduino se va folosi de valoarea data de către modulul de citire al greutății, aceasta fiind măsurată de senzorul de greutate, când aceasta depășește un anumit prag, pompa se oprește si Arduino așteaptă comanda pentru următorul ingredient(cantitatea acestuia). Pompele peristaltice se folosesc de crearea presiunii pentru a face posibilă turnarea lichidelor(prin tubul de silicon). Mufa jack va fi folosită pentru a trimite curent modulelor cu driver de motoare de la sursa de alimentare.



## Hardware Design



Listă Piese:

- Arduino UNO R3
- Ecran LED IIC
- Senzor de atingere
- Senzor Greutate
- Modul citire greutate(hx711 ADC IC)

- 2 X L298N modul cu driver de motoare dual
- 3 X pompe dozatoare peristaltice
- Jack DC
- Sursa alimentare 6V 1200mA
- Tub silicon 10mm

Acest Cocktail Maker utilizează trei pompe peristaltice pentru a distribui lichide în mod controlat, gestionate de un microcontroller Arduino UNO R3.

Sistemul este alimentat printr-un jack DC care furnizează tensiune către toate componentele. Fiecare pompă este controlată de un driver L298N, care primește semnale de control de la Arduino pentru a gestiona funcționarea pompelor.

Un senzor de greutate (Load Cell) conectat la un amplificator HX711 măsoară cantitatea de lichid distribuit. Arduino interpretează aceste date și ajustează pompele pentru a asigura distribuirea precisă a lichidelor. Un afișaj LCD conectat la Arduino afișează informații despre procesul de preparare a cocktail-urilor.

Utilizatorul poate interacționa cu sistemul printr-un senzor de atingere pentru a porni prepararea cocktail-ului.

Conexiunile de alimentare și semnalele de control sunt distribuite între toate componentele pentru a asigura funcționarea corectă. Sistemul permite distribuția automată și precisă a lichidelor, utilizând pompe peristaltice controlate electronic, un senzor de greutate și o interfață ușor de utilizat pentru utilizator.

Urmează programarea plăcuței Arduino și montarea pieselor într-o cutie specifică/asemănătoare unui dozator.

## Software Design

### Biblioteci Incluse

- Wire.h: Biblioteca pentru comunicarea I2C.
- LiquidCrystal\_I2C.h: Biblioteca pentru controlul afișajului LCD I2C.
- HX711.h: Biblioteca pentru senzorul de greutate HX711.

### Definirea Pinilor

```
#define TOUCH_SENSOR_PIN 2
#define HX711_DT 4
#define HX711_SCK 5
#define PUMP1_IN1 7
#define PUMP1_IN2 8
```

```
#define PUMP2_IN3 9
#define PUMP2_IN4 10
#define PUMP3_IN1 11
#define PUMP3_IN2 12
```

- Definirea pinilor pentru senzorul tactil, senzorul de greutate și pompele controlate de L298N.
- Inițializarea Afișajului LCD și Senzorului de Greutate

LiquidCrystal\_I2C lcd(0x27, 16, 2); HX711 scale; float calibration\_factor = -7050.0;

- Inițializarea afișajului LCD cu adresa I2C și dimensiunea (16×2).
- Inițializarea senzorului de greutate HX711 și setarea unui factor de calibrare.

## Funcția setup()

```
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Put your glass");
  lcd.setCursor(0, 1);
  lcd.print("on the counter");
  pinMode(TOUCH_SENSOR_PIN, INPUT);
  scale.begin(HX711_DT, HX711_SCK);
  scale.set_scale(calibration_factor);
  scale.tare();
  pinMode(PUMP1_IN1, OUTPUT);
  pinMode(PUMP1_IN2, OUTPUT);
  pinMode(PUMP2_IN3, OUTPUT);
  pinMode(PUMP2_IN4, OUTPUT);
  pinMode(PUMP3_IN1, OUTPUT);
  pinMode(PUMP3_IN2, OUTPUT);
}
```

- Inițializarea comunicării seriale pentru debugging.
- Inițializarea afișajului LCD și activarea iluminării de fundal.
- Setarea mesajului inițial pe LCD.
- Configurarea pinului pentru senzorul tactil.
- Inițializarea senzorului de greutate HX711, setarea factorului de calibrare și resetarea la zero.
- Configurarea pinilor pentru pompele controlate de L298N ca ieșiri.

## Funcția loop()

```
void loop() {
  if (scale.is_ready()) {
```

```
float weight = scale.get_units(5);
Serial.print("Weight: ");
Serial.println(weight);
if (weight > 7) { // Pragul pentru detectarea unui pahar
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Press to prepare");
  lcd.setCursor(0, 1);
  lcd.print("your cocktail");
  if (digitalRead(TOUCH_SENSOR_PIN) == HIGH) {
    prepareCubaLibre();
  }
} else {
  lcd.setCursor(0, 0);
  lcd.print("Put your glass");
  lcd.setCursor(0, 1);
  lcd.print("on the counter");
}
} else {
  Serial.println("HX711 not found.");
}
delay(500);
}
```

- Verifică dacă senzorul de greutate este pregătit.
- Obține greutatea medie din 5 citiri.
- Dacă greutatea depășește pragul de 7 unități, afișează mesajul pentru a apăsa pentru a prepara cocktail-ul.
- Dacă senzorul tactil este activat, apelează funcția `prepareCubaLibre()`.
- Dacă greutatea este sub prag, reafișează mesajul inițial.

## Funcția `prepareCubaLibre()`

```
void prepareCubaLibre() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Preparing your");
  lcd.setCursor(0, 1);
  lcd.print("Cocktail!");
  Serial.println("Activating Pump 1");
  digitalWrite(PUMP1_IN1, HIGH);
  digitalWrite(PUMP1_IN2, LOW);
  delay(22000);
  digitalWrite(PUMP1_IN1, LOW);
  digitalWrite(PUMP1_IN2, LOW);
  Serial.println("Activating Pump 2");
  digitalWrite(PUMP2_IN3, HIGH);
  digitalWrite(PUMP2_IN4, LOW);
}
```

```
delay(40000);
digitalWrite(PUMP2_IN3, LOW);
digitalWrite(PUMP2_IN4, LOW);
Serial.println("Activating Pump 3");
digitalWrite(PUMP3_IN1, HIGH);
digitalWrite(PUMP3_IN2, LOW);
delay(15000);
digitalWrite(PUMP3_IN1, LOW);
digitalWrite(PUMP3_IN2, LOW);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Enjoy your");
lcd.setCursor(0, 1);
lcd.print("Cocktail!");
delay(5000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Put your glass");
lcd.setCursor(0, 1);
lcd.print("on the counter");
}
```

- Șterge afișajul și afișează mesajul de pregătire a Cuba Libre;
- Activează pompa 1 (rom) pentru 3 secunde;
- Activează pompa 2 (cola) pentru 5 secunde;
- Activează pompa 3 (suc de lămâie) pentru 1 secundă;
- Afișează mesajul de finalizare și așteaptă 5 secunde;
- Resetează mesajul inițial pe LCD;


## Rezultate Obținute

Video demonstrativ: <https://www.youtube.com/shorts/6Av4EtjQRUY>

## Concluzii

După nopți nedormite și multă perseverență, proiectul este unul reușit. Laboratoarele folosite sunt cele de I2C (ecran LCD), ADC (senzor de greutate) și PWM (motoarele care alimentează pompele). Pentru a putea îmbunătăți proiectul pe viitor, se pot ajusta motoarele L298N pentru a varia viteza de umplere a paharului în funcție de lichidele folosite pentru Cocktail. De asemenea, se poate îmbunătăți carcasa dispozitivului pentru a-l face mai rezistent.

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).

**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

## 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** și **Resurse Hardware**.

[Export to PDF](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2024/sseverin/rares.dumitrescu> 

Last update: **2024/05/27 13:28**