

Cafetiera automata

Intro

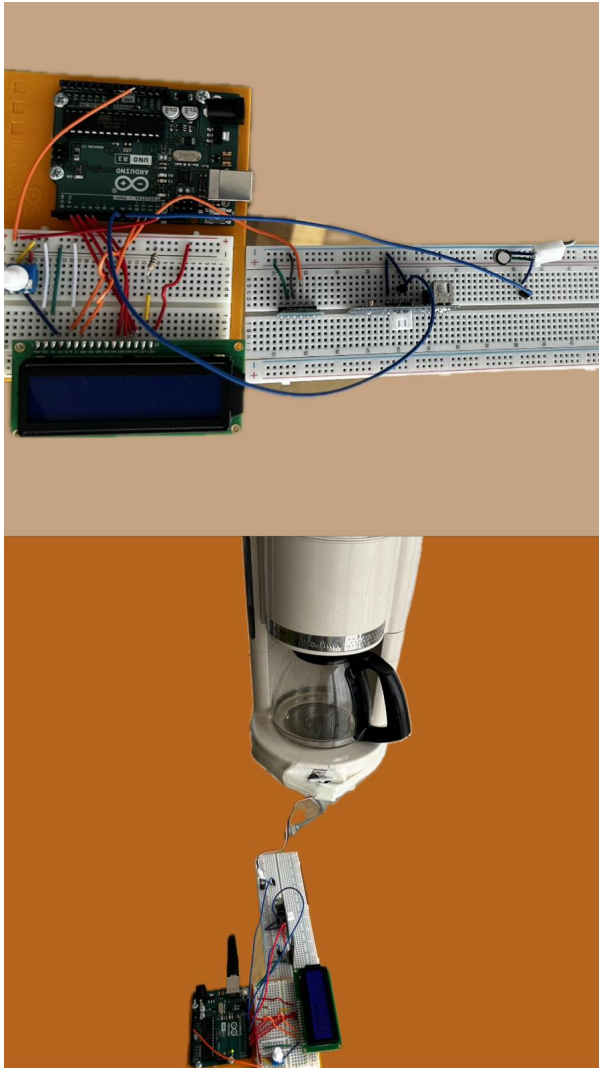
Proiectul constă într-o cafetieră automată controlată wireless, cu ajutorul unui modul Bluetooth conectat la un Arduino Uno. Utilizatorul poate porni procesul de preparare a cafelei direct de pe telefon, trimițând comenzi către Arduino prin aplicația Bluetooth. La primirea comenzii, senzorul ultrasonic verifica dacă este apa în rezervor după aceea, un servo motor acționează mecanic butonul de pornire al cafetierei. Astfel, întregul proces poate fi declanșat de la distanță, fără intervenție manuală directă.

Descriere generală

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



Hardware Design



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 plănuiești să le implementați
- (etapa 3) surse și funcții implementate
- `#include <SoftwareSerial.h>`

```
#include <SoftwareSerial.h>
#include <Servo.h>
#include <LiquidCrystal.h>

// Bluetooth
SoftwareSerial mySerial(7, 6);

// LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
// Servo
Servo servoMotor;
const int servoPin = 10;

// Senzor ultrasonic
const int trigPin = 9;
const int echoPin = 8;

// Configurări
const int posInitial = 0;
const int posApasare = 150;
const int pragNivelApa = 10; // cm
const int delayApasare = 500;

void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);

  lcd.begin(16, 2);
  lcd.print("Sistem pornit");

  servoMotor.attach(servoPin);
  servoMotor.write(posInitial);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

long masoaraNivelApa() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  long durata = pulseIn(echoPin, HIGH);
  long distanta = durata * 0.034 / 2;
  return distanta;
}

void loop() {
  if (mySerial.available()) {
    char comanda = mySerial.read();
    Serial.print("Comanda primita: ");
    Serial.println(comanda);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Comanda: ");
    lcd.print(comanda);
  }
}
```

```
if (comanda == '1') {
    long nivelApa = masoaraNivelApa();

    lcd.setCursor(0, 1);
    lcd.print("Apa: ");
    lcd.print(nivelApa);
    lcd.print("cm");

    if (nivelApa > pragNivelApa) {
        Serial.println("Apasare...");
        lcd.setCursor(0, 1);
        lcd.print("Cafea pornita ");

        servoMotor.write(posApasare);
        delay(delayApasare);
        servoMotor.write(posInitial);
    } else {
        Serial.println("Apa insuficienta");
        lcd.setCursor(0, 1);
        lcd.print("Apa insuficienta");
    }
}
}
```

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

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/prj2025/vradulescu/vlad_andrei.voicu



Last update: **2025/05/25 17:34**