

Cântar de bucătărie

Introducere

Proiect realizat de Preda Raluca-Aida, 334CA

Despre ce este proiectul?

Proiectul constă într-un cântar de bucătărie bazat pe o Placă de Dezvoltare compatibilă cu Arduino UNO. Dispozitivul măsoară greutatea ingredientelor și afișează valoarea pe un ecran LCD.

Care este scopul proiectului?

Scopul acestui proiect este de a oferi o soluție precisă și accesibilă pentru cântărirea ingredientelor în bucătărie, astfel încât utilizatorul să poată prepara rețete cu doze exacte și rapiditate.

Ce a inspirat ideea?

Inspirația a venit din nevoia de precizie în gătit și cofetărie, unde diferența de câteva grame poate afecta reușita unei rețete. Fascinația pentru proiecte embedded și senzori de forță m-a motivat să explorez modul în care un load cell și un microcontroller simplu precum ATmega328p pot lucra împreună pentru un dispozitiv practic și educațional.

De ce este util acest proiect?

Acest cântar de bucătărie demonstrează cum componente electronice low-cost pot fi combinate pentru a crea un aparat util în viața de zi cu zi. Pe lângă beneficiile practice ale cântării precise, proiectul are valoare educațională, întrucât învață interfețarea senzorilor, protocolul HX711, gestionarea afișajelor.

Descriere generală



Descrierea componentelor

Arduino UNO R3 (ATmega328P + ATmega16U2)

- Primește datele de greutate de la HX711.
- Proceasează intrările de la butoane.
- Gestionează afișajul și redarea audio.
- Comunică cu toate celelalte componente prin GPIO, I2C, PWM și SPI.

Load Cell + Amplificator HX711

- Transformă forța mecanică (greutatea ingredientelor) în semnal digital.
- Load cell-ul generează un semnal analog proporțional cu greutatea, iar modulul HX711 realizează conversia A/D.
- Comunică cu Arduino prin două linii digitale.

Butoane (Tare și Selectare Unitate)

- Tare: resetează greutatea afișată la zero.
- Selectare Unitate: comută între grame și uncii.
- Fiecare buton este conectat la un pin GPIO al plăcii Arduino UNO R3.

LCD Display

- Afișează greutatea măsurată și unitatea selectată.
- Conectat la Arduino pe magistrala I2C, folosind doar două fire pentru date și ceas.

Modul SD (Card Reader)

- Stochează fișiere audio.
- Arduino citește aceste fișiere WAV/RAW de pe card prin SPI și le trimite către buzzer pentru redare.

Buzzer

- Redă audio-ul citit de pe cardul SD.
- Este controlat de Arduino printr-un pin PWM, care produce semnalul necesar decodării și redării pe buzzer.

Hardware Design

Lista Componentelor

Nume	Cantitate	Preț per unitate (lei)
Placa de Dezvoltare Compatibila cu Arduino UNO R3 (ATmega328p + ATmega16u2) + cablu	1	54.37
LCD Display	1	16.34
Buzzer	1	1.40
Button	2	0.36
Breadboard	1	9.98
Celula de Sarcina de 5 kg cu Amplificator HX711	1	25.85
Cititor de card SD	1	28.16
Prețul total: 136.82 lei		



1. Arduino UNO
 1. Microcontroller principal care orchestrează întregul sistem.
 2. Tensiune de lucru: 5 V
 3. Interfețe utilizate:

- I2C → comunicație cu LCD-ul 1602 (SDA pe A4, SCL pe A5)
 - SPI → comunicație cu modulul SD-card (CS → D10, MOSI → D11, MISO → D12, SCK → D13)
 - Digital I/O → legături cu HX711 (DT → D3, SCK → D2)
 - PWM → generare de tonuri cu buzzerul pasiv (tone()) pe D9
 - GPIO → citirea stării butoanelor pe D7 și D8 cu pull-down extern
4. Motivație: UNO oferă toate interfețele necesare (I2C, SPI, PWM, GPIO), are suport extins de biblioteci pentru HX711, SD-card și LCD, și e ideal pentru prototipare rapidă.
2. HX711 Load Cell Amplifier + Celulă de sarcină
1. Modulul HX711 convertește semnalul analogic de la celula de sarcină într-un stream digital, permițând măsurarea greutatei.
2. Pini utilizați:
- DT (Data) → D3 (digital input)
 - SCK (Clock) → D2 (digital output)
 - VCC → 5 V
 - GND → GND
3. Motivație: HX711 oferă o rezoluție înaltă (24 biți) și comunică ușor prin doi pini digitali, ideal pentru măsurători precise de greutate.
3. Modul SD-Card (microSD)
1. Permite stocarea locală a datelor (greutate, timpi, evenimente).
2. Interfață: SPI
3. Pini utilizați:
- CS → D10
 - DI (MOSI) → D11
 - DO (MISO) → D12
 - SCK → D13
 - VCC → 5 V
 - GND → GND
4. Motivație: SD-card-urile sunt ieftine, ușor de integrat și permit logarea pe termen lung fără a încărca memoria internă a Arduino.
4. LCD 1602 cu interfață I2C
1. Afișează valorile măsurate și mesaje de stare.
2. Pini utilizați:
- SDA → A4
 - SCL → A5
 - VCC → 5 V
 - GND → GND
3. Motivație: Interfața I2C reduce numărul de fire la doar 2 (plus VCC/GND), eliberând pini digitali și simplificând cablajul.
5. Buzzer pasiv
1. Emite semnale sonore de alertă.
2. Pin semnal → D9 (PWM-capable)
3. GND → GND
4. Motivație: Pinul D9 suportă tone(), permițând generarea de tonuri la diferite frecvențe pentru semnalizare acustică.
6. Butoane tactile (x2)
1. Permit utilizatorului să interacționeze cu sistemul (pornire/oprire, reset, navigare meniu etc.).
2. Pini semnal → D7 și D8 (GPIO)
3. Pull-down extern → rezistențe de 10 kΩ către GND
4. Configurare software: pinMode(pin, INPUT); digitalRead(pin);
5. Motivație: folosirea rezistențelor de pull-down externe asigură stabilitate a citirii și pini digitali liberi pentru butoane.

7. Breadboard + cabluri jumper

1. Asigură interconectarea rapidă și flexibilă a componentelor.
2. Tipuri de fire: mamă-mamă pentru module, tată-mamă pentru legături la Arduino.
3. Motivație: prototipare fără lipire, posibilitate de reconfigurare rapidă.



Software Design

Mediu de dezvoltare:

- Arduino IDE

Librării:

- SPI.h – pentru comunicația SPI cu modulul SD.
- SD.h – pentru citirea de fișiere de pe cardul SD (findFirstWav, parseWavHeader).
- HX711.h – driverul standard pentru conversia celulei de sarcină (tare, get_units).
- avr/interrupt.h – pentru definirea și folosirea ISR-ului Timer1_COMPA_vect.
- În plus, nu folosim nicio librărie I²C: am implementat “bit-banged” direct pe D2/D4.

Structuri și algoritmi:

- Structuri:
 - Enumerare enum Unit { KG, G, ML, DL, OZ } — facilitează ciclarea între unități de măsură.
 - Arrays pentru debounce (lastDebounceTime[3], lastBtnState[3]).
- Algoritmi
 - Debounce simplu: compară starea curentă a fiecărui buton cu ultima, cu o întârziere minimă (debounceDelay).
 - State-machine PWM-WAV:
 - playing = false/true → ISR sau buclă principală pornește/oprește redarea.
 - Logica threshold (dacă greutatea > 4.5 kg → pornește WAV, altfel → oprește).
 - Bit-banged I²C: secvență de start, byte-by-byte, ACK/NACK, stop – implementată manual pentru LCD.
 - Header parsing WAV:
 - Sari peste „RIFF” + „fmt” (12 octeți), apoi citești câte 8 bytes per chunk până găsești „data”.

Funcții implementate:

- setup() – inițializări (SD, PWM, Timer ISR, LCD, HX711, butoane).
- loop() –
 - Debounce & acțiuni butoane (tare, lock, schimb unitate).
 - Citire greutate și aplicare offset/tare.
 - Control redare WAV după pragul de 4.5 kg.
 - Conversie greutate în unități și afișare pe LCD.
- ISR(TIMER1_COMPA_vect) – citește octet cu octet și scrie în registrele PWM.
- findFirstWav(), parseWavHeader(File &f) – căutare și poziționare pentru fișierul WAV.
- I²C bit-banged routines: i2c_start(), i2c_write_bit(), i2c_read_ack(), pcf_write(), lcd_cmd(), lcd_data() etc.

- Debounce logic – în loop compararea stărilor și actualizarea timpurilor.

Codul sursă: cod-cantar.zip

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/atoader/raluca_aida.preda



Last update: **2025/05/25 16:43**