

Cantar electronic

Introducere

Proiectul constă în realizarea unui cântar digital de mare capacitate (până la 200 kg) care nu doar măsoară greutatea, ci permite și introducerea unui preț unitar/cod de identificare produs pentru a calcula în timp real valoarea totală a produselor. Datele pot fi exportate prin bluetooth către un laptop sau telefon.

Scopul este crearea unui instrument de măsură precis, eficient din punct de vedere energetic și capabil să interacționeze cu alte dispozitive pentru logarea datelor (data logging). Ideea a pornit de la nevoia de a avea un cântar versatil pentru uz industrial și comercial, care să elimine necesitatea calculului manual și să permită stocarea istoricului de cântăriri.

Este util pentru optimizarea procesului de inventariere și vânzare, oferind o interfață intuitivă (14 butoane) și un sistem de management al energiei (Sleep Mode) care prelungește durata de viață a bateriei.

Descriere generală

Sistemul procesează semnalele analogice de la celulele de sarcină prin intermediul amplificatorului HX711, care convertește tensiunile mici în date digitale pe 24 de biți. Microcontrolerul interpretează aceste date, gestionează tastatura extinsă și afișează informațiile pe un LCD I2C.

Funcții principale:

Amplificare de precizie: Modulul HX711 asigură citiri stabile și precise pentru cele 4 celule de sarcină.

Interfață cu 14 butoane: Numpad (0-9) + Save (salvare date), Sleep/On (energie), Tare (aducere la zero) și Clear (ștergere preț).

Feedback sonor inteligent: Melodie la pornire, bip scurt la tastare și un semnal sonor distinct pentru confirmarea salvării prin Bluetooth.

Management Energie: Funcție de Auto-Sleep după 60s de inactivitate, cu trezire instantanee (Wake-up).



Hardware Design

Componente și Conectivitate

- **Senzori de greutate:** 4 x celule de sarcină interconectate.
- **Amplificator HX711:** Interfața de conversie analog-digitală între senzori și microcontrolerul ATmega328P.
- **Display:** LCD 1602 echipat cu modul de conversie I2C pentru salvarea pinilor GPIO.
- **Bluetooth:** Modul HC-05 pentru transmiterea wireless a datelor prin protocolul USART.
- **Sunet:** Buzzer pasiv controlat prin PWM pentru feedback acustic la acționarea tastelor.
- **Indicator optic:** LED de stare care confirmă alimentarea corectă a sistemului și faptul că aparatul este gata de funcționare.
- **Input:** Scară de 14 butoane (Numpad + Funcții speciale), configurată ca divizor de tensiune rezistiv și citită printr-un singur pin analogic (ADC) pentru eficientizarea resurselor hardware.

Realizarea practică și Structura Mecanică

Pentru asigurarea unei funcționări stabile și a unei precizii ridicate a sistemului de cântărire, realizarea fizică a proiectului a fost împărțită în patru mari componente: structura de bază, subsistemul mecanic de suport, platforma de cântărire și blocul de control.

1. Șasiul și Structura de Bază

Întregul ansamblu este montat pe o placă din **PAL melaminat**. Aceasta a fost aleasă pentru a asigura o suprafață de bază perfect plană, rigidă și izolatoare din punct de vedere electric, prevenind torsiunile care ar putea influența negativ citirile senzorilor de greutate.

2. Integrarea Celulelor de Sarcină

Pentru funcționarea corectă și liniară a celulelor de sarcină (load cells), este critic ca acestea să fie suspendate corect și să nu atingă direct suprafața de bază pe toată lungimea lor. În acest sens, s-au proiectat și realizat **suportți speciali din tablă de aluminiu cu o grosime de 10 mm**. Acești suportți oferă:

- O fixare extrem de rigidă, eliminând orice flexare parazită a metalului sub acțiunea greutății.
- Spațiul liber (decalajul) necesar sub celulă pentru a permite deformația elastică corectă a

micro-strain-gauge-ului intern, asigurând astfel acuratețea măsurărilor.

3. Platforma de Cântărire

Peste celulele de sarcină a fost așezată o suprafață din **plexiglass cu o grosime de 6 mm**, care servește drept platou de cântărire. Pe lângă durabilitatea și rigiditatea necesară susținerii obiectelor, alegerea unui material transparent are un rol estetic și didactic major: ea permite vizualizarea directă a legăturilor realizate între celule (conexiunile electrice din puntea Wheatstone și dispunerea geometrică a senzorilor), oferind o transparență totală asupra modului de interconectare hardware.

4. Blocul de Control și Interfața cu Utilizatorul

Toate componentele electronice active responsabile de procesare, amplificare și comunicație (placa de dezvoltare Arduino, amplificatorul HX711, modulul Bluetooth HC-05 și buzzerul pentru alertă sonoră) sunt adăpostite în siguranță **în interiorul unei cutii de control dedicate**. Aceasta protejează circuitele împotriva prafului și a atingerilor accidentale.

Pentru o ergonomie crescută, **pe capacul acestei cutii** a fost centralizată întreaga interfață de tip Man-Machine (MMI):

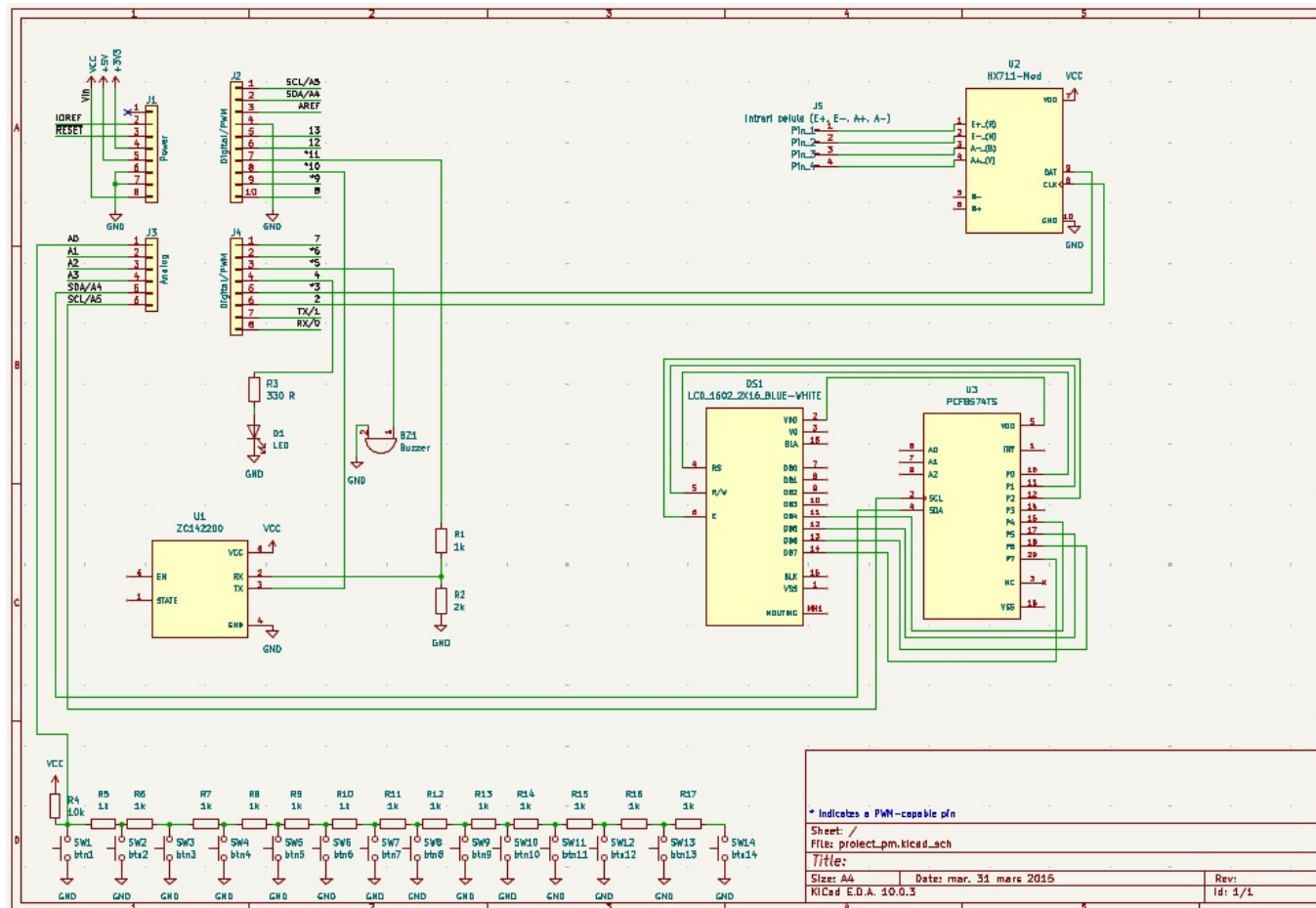
- **Ecranul LCD 1602 (I2C):** Poziționat central pentru o lizibilitate optimă a greutății afișate.
- **Scara de 14 butoane (ADC):** Aliniată pe panou pentru accesul rapid la funcțiile cântarului.
- **LED-ul de stare:** Montat la vedere pentru a oferi confirmarea vizuală instantanee că sistemul s-a inițializat cu succes și este pregătit pentru utilizare.

Tabelul de Conexiuni Hardware

Componentă	Pin Componentă	Pin Arduino	Tip / Note
Senzor HX711	VCC	5V	Alimentare modul
	GND	GND	Masă comună
	DT	Pin 2	Date (Digital Input)
	SCK	Pin 3	Ceas (Digital Output)
Ecran LCD I2C	VCC	5V	Alimentare ecran
	GND	GND	Masă comună
	SDA	Pin A4	Date I2C
	SCL	Pin A5	Ceas I2C
Butoane (ADC)	Semnal	Pin A0	Scară rezistențe (*Notă: configurat pe A0 deoarece ADC-ul are nevoie de pin analogic*)
LED Stare	Anod (+)	Pin 4	Indicator pornire (Digital Output)
Buzzer	Pozitiv (+)	Pin 5	Semnalizare acustica (Digital Output)
Bluetooth HC-05	TXD	Pin 10	Transmisie (SoftwareSerial RX)

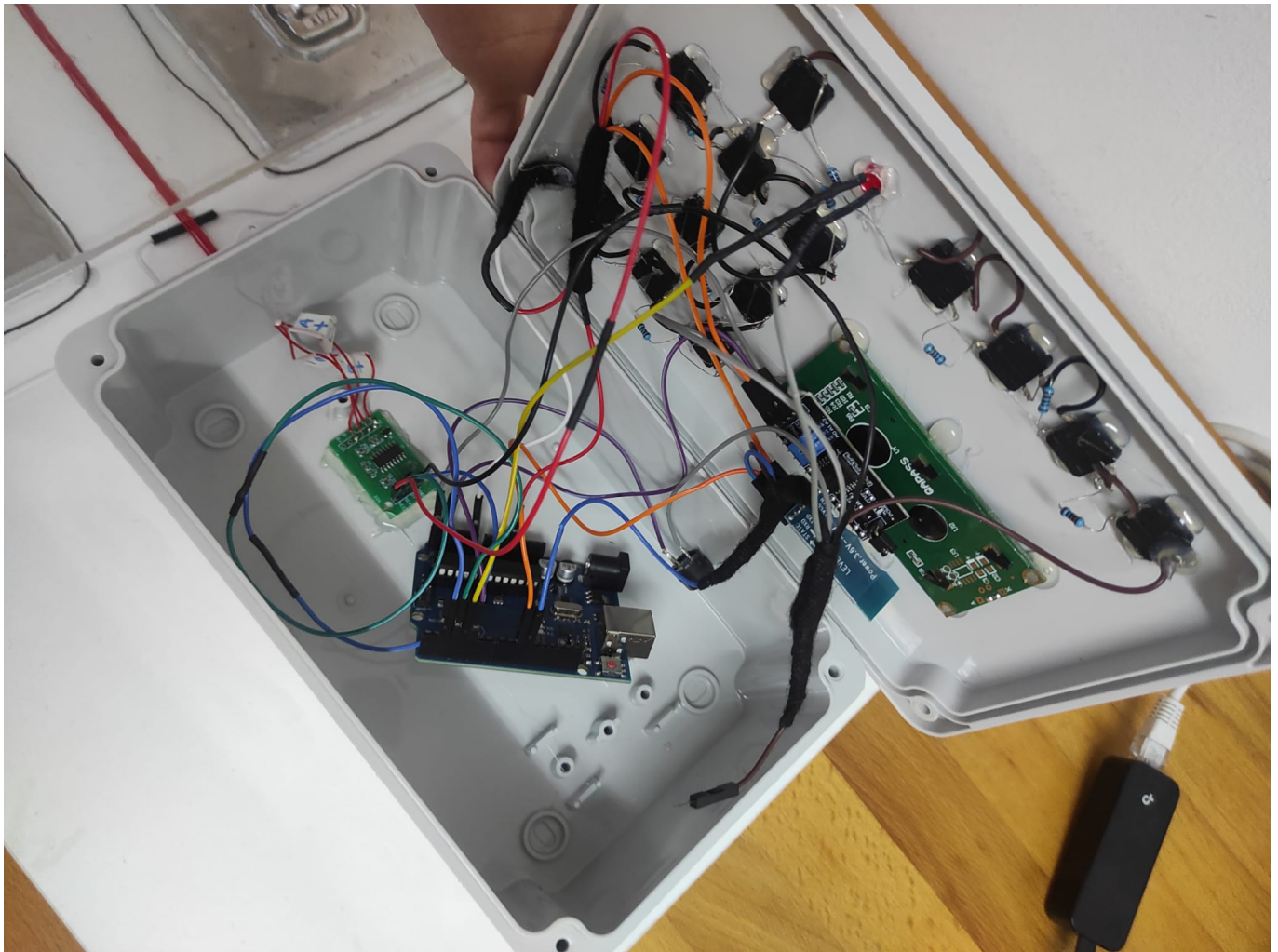
	RXD	Pin 11	Receptie (SoftwareSerial TX - Prin divizor de tensiune)
--	-----	--------	---

Demo(masurare greutate): <https://www.youtube.com/watch?v=COK7Svo93iw>









Software Design

Mediu de dezvoltare:

- Aplicația a fost dezvoltată în PlatformIO IDE (utilizând toolchain-ul AVR-GCC).
- Deși mediul de bază a pornit de la framework-ul Arduino, firmware-ul a fost rescris majoritar într-o arhitectură de tip bare-metal. S-a eliminat dependența de HAL-ul Arduino (funcții blocante precum `delay()`, `millis()`, `analogRead()`, clasa `Serial`), toate aceste funcționalități fiind implementate manual, de la zero, prin configurarea și manipularea directă a regiștrilor hardware specifici microcontrolerului ATmega328P.

Librării și surse 3rd-party:

- `<avr/io.h>` și `<avr/interrupt.h>` - Biblioteci native AVR folosite intens pentru accesarea regiștrilor de I/O, a timerelor și pentru declararea vectorilor de întrerupere (ISR).
- `HX711_ADC.h` (autor: Olav Kallhovd) - Păstrată strict pentru decodarea protocolului sincron pe 24 de biți necesar preluării datelor brute de la celula de sarcină.
- `LiquidCrystal_I2C.h` (autor: Marco Schwartz) - Utilizată pentru abstractizarea protocolului hardware I2C (TWI) către display-ul LCD.

Algoritmi și concepte implementate:

- **Pseudo-RTOS bazat pe Întreruperi Hardware:** S-a creat un ceas de sistem propriu și generare

de semnale acustice folosind modul CTC al Timerului 2. Acest algoritm garantează o reacție de tip real-time (rezoluție de 0.5ms), asigurând frecvența buzzer-ului complet independent de blocajele din bucla principală.

- **Debouncing Temporal:** Algoritm software asincron pentru eliminarea bounce-ului mecanic al contactelor tastaturii rezistive. S-a implementat o fereastră de histererezis de 200ms activată exclusiv la eliberarea tastei (falling edge).
- **Comunicație Serială Hardware (USART):** Implementare manuală de low-level a transmisiei de date (baud rate generat direct prin formula pe registrul UBRR0) utilizând tehnica de polling pe registrul UDRE0, oferind un throughput optim pentru trimiterea chitanței Bluetooth.
- **Automat Finit de Stări:** Controlul interfeței UI și al modului de Sleep a fost modelat ca o mașină de stări non-blocantă, tranzițiile fiind dictate de timeout-uri evaluate asincron pe baza ceasului de sistem propriu.

Surse și funcții implementate:

Nume Funcție / Macrou	Nivel Implementare	Descriere Rol și Funcționalitate
ISR(TIMER2_COMPA_vect)	Hardware (Interrupt)	Rutina principală de întrerupere generată la fiecare 0.5ms. Gestionează incrementarea timpului de sistem și comutarea (toggle) atomică a pinului PD5 pentru oscilația hardware a buzzerului.
citesteTimpulNostru()	Software (Atomic)	Citește variabila de sistem pe 32 de biți mascând temporar întreruperile globale (cli() / sei()), prevenind astfel coruperea datelor concurente.
ADC_init() / ADC_read()	Bare-metal I/O	Inițializează manual multiplexorul pe canalul ADC0 (A0) și forțează conversia hardware prin setarea bitului ADSC, returnând rezultatul rezoluției pe 10 biți.
citesteTastaADC()	Software (Logică)	Aplică algoritmul de debouncing temporal pe citirile ADC și mapează divizorul rezistiv de tensiune la o tastă logică specifică proiectului.
USART_init()	Bare-metal I/O	Configurează manual baud rate-ul intern și setează formatul pachetelor de date pe 8N1 direct în registrul UCSR0C.
USART_transmitChar()	Bare-metal I/O	Execută tehnica de polling pe flag-ul Data Register Empty și plasează octetul curent în registrul tampon UDR0 pentru transmisia fizică pe pinul TX.
emiteBeepNonBlocant()	Software	Armează un timer asincron în mașina de stări pentru activarea semnalului acustic, fără a suspenda deloc firul de execuție principal al programului.
setup()	Configurare Sistem	Setează direcția pinilor (DDRD), calibrează senzorii și armează perifericele interne, încheind cu activarea întreruperilor globale prin sei().
loop()	Scheduler (FSM)	Firul principal de execuție; evaluează stările sistemului (TARA, SAVE, SLEEP), actualizează afișajul I2C și rutează evenimentele tastaturii asincron.

Rezultat obtinut

În videoclipul de mai jos este prezentată funcționarea practică a dispozitivului. Sunt demonstrate: inițializarea sistemului (Auto-Tara), răspunsul non-blocant al interfeței, precizia filtrului de debouncing la apăsarea tastelor (însoțită de confirmarea acustică generată asincron prin Timer) și, la final, transmisia datelor pe Bluetooth către un terminal mobil folosind protocolul hardware USART.

<https://www.youtube.com/watch?v=gNj1DtY83e4>

Concluzii

Realizarea proiectului “Cântar Inteligent” a marcat tranziția de la mediul de nivel înalt (Arduino) la ingineria de tip bare-metal, consolidând înțelegerea arhitecturii interne a microcontrolerului ATmega328P.

Cea mai mare reușită a proiectului a fost implementarea unei arhitecturi software complet asincrone (non-blocante). S-au eliminat funcțiile de abstractizare (HAL) în favoarea unui pseudo-RTOS bazat pe întreruperi hardware (ISR) pe Timerul 2.

Soluțiile tehnice cheie care au asigurat stabilitatea sistemului:

- **Multitasking Real-Time:** Controlul buzzerului direct prin rutine ISR a eliminat complet jitter-ul acustic, chiar și în timpul transmisiilor masive de date.
- **Filtrare Software:** Implementarea algoritmului de debouncing temporal a anulat zgomotul mecanic al tastaturii rezistive ADC.
- **Comunicație Optimizată:** Folosirea regiștrilor USART prin tehnica de polling a asigurat transmisia rapidă și stabilă a datelor via Bluetooth.

În concluzie, sistemul final este robust și eficient, demonstrând importanța lucrului direct cu regiștrii hardware.

Ca dezvoltări viitoare, pe lângă proiectarea unui cablaj imprimat (PCB) dedicat, se vizează crearea unei aplicații software dedicate (desktop/mobile). Această interfață grafică (GUI) va prelua fluxul de date transmis prin protocolul Bluetooth, oferind un sistem complet de management al bazei de date, unde utilizatorul va putea vizualiza istoricul tranzacțiilor, edita informații și șterge centralizat “chitanțele” salvate de către cântar.

Download

Toate resursele obținute în urma realizării proiectului (codul sursă complet al firmware-ului optimizat bare-metal, schemele electrice hardware, fișierele de configurare PlatformIO și documentația aferentă) sunt găzduite public și pot fi descărcate direct de pe platforma GitHub.

În cadrul repoziitoriului se găsesc:

- **Surse Firmware:** Codul complet structurat pentru PlatformIO IDE (stil bare-metal, drivere USART/ADC manuale și ISR Timer 2).
- **Fișier README:** Documentație completă cu instrucțiunile de configurare, compilare și deployment automat pe microcontroler.
- **ChangeLog:** Istoricul complet al modificărilor și optimizărilor aduse algoritmilor software (inclusiv trecerea de la SoftwareSerial la USART și implementarea filtrelor de debounce).

Link-ul oficial către repozitoriul proiectului este:

- [\[\] Repozitoriu GitHub - Cantar Electronic PM](#)

Jurnal

Evoluția proiectului a decurs treptat, parcurgând structurat atât etapele de construcție mecanică și asamblare hardware, cât și optimizarea software-ului prin lucrul direct pe regiștri:

- **Etapa 1: Proiectare și Construcție Mecanică**
 - Dimensionarea, debitarea și asamblarea structurii fizice a cântarului. S-au construit manual suportul principal și suportii speciali pentru celulele de sarcină din aluminiu, asigurând o structură rigidă și stabilă, esențială pentru acuratețea măsurărilor de greutate.
 - Proiectarea și realizarea incintei (carcasei) dedicate, concepută special pentru a adăposti în siguranță circuitele electronice și pentru a integra ergonomic display-ul LCD și butoanele tastaturii.
- **Etapa 2: Setup Electronic și Validarea Conexiunilor (PoC)**
 - Montarea componentelor electronice în incinta realizată (placa de dezvoltare, amplificatorul HX711, modulul Bluetooth HC-05, tastatura rezistivă și buzzerul pasiv).
 - Scrierea unui cod inițial de test folosind bibliotecile standard Arduino (`SoftwareSerial`, `analogRead()`) cu scopul exclusiv de a valida corectitudinea legăturilor electrice și calibrarea inițială a senzorului.
- **Etapa 3: Trecerea pe Regiștri (USART și ADC Manual)**
 - Eliminarea completă a librăriei `SoftwareSerial` și trecerea modulului Bluetooth direct pe perifericul hardware USART0 (pinii 0 și 1). S-au implementat manual funcțiile de inițializare a baud rate-ului și de transmisie a datelor prin tehnica de **polling** pe registrul tampon.
 - Înlocuirea funcției standard `analogRead()` cu configurarea și citirea manuală a regiștrilor hardware `ADMUX` și `ADCSRA` pentru gestionarea tastaturii pe pinul A0.
- **Etapa 4: Logica Meniului și Controlul Asincron al Buzzerului**
 - Structurarea întregii logici a interfeței (Auto-Tara, introducere preț, salvare chitanță, mod Sleep) sub forma unei Mașini de Stări (FSM) non-blocante.
 - Implementarea unei metode de generare a frecvenței acustice pentru buzzerul pasiv prin comutarea (toggle) directă a stării pinului PD5 în interiorul buclei principale.
- **Etapa 5: Optimizarea prin Întreruperi (ISR) și Finisarea Sistemului**
 - Identificarea unei probleme de **jitter acustic** (distorsionarea sunetului buzzerului din cauza timpului blocant necesar trimiterii datelor prin USART).
 - Soluția: Configurarea manuală a Timerului 2 în mod CTC pentru a declanșa o întrerupere

hardware (ISR) la fiecare 0.5 milisecunde. Această rutină a preluat complet generarea semnalului curat pentru buzzer și funcția de ceas de sistem (înlocuind definitiv funcția `millis()`).

- Implementarea algoritmului de *debouncing* temporal (fereastră de 200ms la eliberarea tastei) pe citirile ADC, eliminând declanșările multiple accidentale și stabilizând perfect comutarea în modul Sleep.

Bibliografie/Resurse

Gruparea documentației, a fișelor tehnice (datasheets) și a resurselor software utilizate pe parcursul dezvoltării proiectului:

Resurse Hardware

- **Microchip ATmega328P Official Datasheet:** [Fișa tehnică oficială ATmega328P](#) — Utilizată intensiv pentru configurarea manuală a regiștrilor de USART (capitolele specifice transmisiei serie), ADC (configurarea multiplexorului și a pre-scalerului) și Timer 2 (modul CTC și masca de întreruperi).
- **Avia Semiconductor HX711 Datasheet:** [Fișa tehnică amplificator celulă de sarcină](#) — Pentru înțelegerea modului de eșantionare și comunicare sincronă a datelor de greutate.
- **Modul Bluetooth HC-05 Reference Manual:** [Documentație tehnică și comenzi AT HC-05](#) — Folosit pentru stabilirea conexiunii fizice și setarea ratei de transmisie implicite (9600 baud).
- **Teoria tastaturilor rezistive (Resistor Ladder):** Ghiduri teoretice privind divizoarele de tensiune succesive utilizate pentru multiplexarea mai multor butoane pe un singur pin analogic.

Resurse Software

- **AVR Libc Reference Manual:** [Documentația oficială AVR-GCC](#) — Consultată special pentru macro-urile de manipulare a biților și gestionarea rutinelor de întrerupere hardware (`<avr/interrupt.h>` și `<avr/io.h>`).
- **PlatformIO IDE Documentation:** [Ghidul oficial PlatformIO](#) — Pentru configurarea mediului de dezvoltare, a compilatorului și a procesului de upload automat pe microcontroler.
- **Librăria HX711_ADC (Olav Kallhovd):** [Repozitoriu GitHub HX711_ADC](#) — Sursa librăriei folosite pentru calibrarea și filtrarea semnalului brut de la cântar.
- **Librăria LiquidCrystal_I2C (Marco Schwartz):** [Repozitoriu GitHub LiquidCrystal_I2C](#) — Pentru controlul ecranului prin protocolul TWI/I2C.
- **Codul Sursă al Proiectului:** [Repozitoriu GitHub - Cantar Electronic PM](#) — Repoziatoriul oficial al acestui proiect unde se regăsesc toate fișierele finale implementate.

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2026/ciprian.popescu0411/andrei_daniel.dinca

Last update: 2026/05/24 21:34

