

Dozator de apă inteligent

Introducere

Proiectul constă într-un dozator de apă inteligent, care detectează volumul curent de apă pe baza unui senzor de greutate și care umple recipientul, cu ajutorul unei pompe și a unui releu. Pentru afișaj, am folosit un ecran LCD și o bară de 10 led-uri pentru progres. LCD-ul va avea mai multe moduri, printre care: progres, timp rămas și statistici (câtă apă am băut în ziua respectivă spre exemplu). Pentru a schimba modurile de afișaj voi folosi niște butoane, iar pentru a măsura timpul precis un RTC.

Scopul proiectului și utilitatea o reprezintă, tocmai, automatizarea procesului, marea majoritate a dozatoarelor fiind manuale, deci incoveniente.

Ideea de la care am pornit a fost creerea unui aparat util, nu foarte scump, de care toată lumea are nevoie.

Descriere generală



1. Microcontroler-ul ATmega328P-XMINI - „creierul”, prelucrează datele primite de la senzor, acționează pompa, realizează statisticile și trimite semnal la LED-uri și LCD pentru afișaj
2. Senzor Greutate 1kg (analogic) - pentru detectarea volumului de apă curent
3. Modul Citire Greutate HX711 - funcționează ca un amplificator și convertește semnalul analogic primit de la senzor într-unul digital pe care îl poate procesa microcontrolerul
4. Sursa 12V - pentru alimentarea pompei
5. Convertor Step-Down 5V - pentru a modifica tensiunea de la 12V la 5V, alimentând microcontrolerul
6. Releu - întrerupător (pornește sau oprește pompa)
7. Pompa - pentru a extrage apa din rezervor
8. RTC - folosit pentru a colecta date pentru statistici. Un avantaj al utilizării unui RTC este acuratețea acestuia, precum și funcționarea independentă de sursa principală de curent datorită bateriei proprii (comunică pe baza protocolului I2C)
9. LCD - folosit pentru afișaj (funcționează pe baza protocolului I2C)
10. Butoane - folosite pentru a porni dozatorul sau a schimba modul de afișaj

11. Bara 10 LED-uri - folosită pentru a afișa progresul. Funcționează folosind PWM pentru a varia intensitatea, în timp ce numărul de LED-uri aprinse se modifică în funcție de progres

Laboratoare folosite:

- Întreruperi (butoanele care: pornesc aparatul, schimbă modul de afișaj etc.)
- PWM (intensitatea LED-urilor de progres crește treptat)
- I2C (Afișajul LCD și modulul RTC)
- UART (Pentru calibrarea senzorului de greutate)

Module externe:

- Timere (RTC)
- ADC (modulul HX711, care transformă semnalul analogic primit de la senzor într-unul digital)

Hardware Design

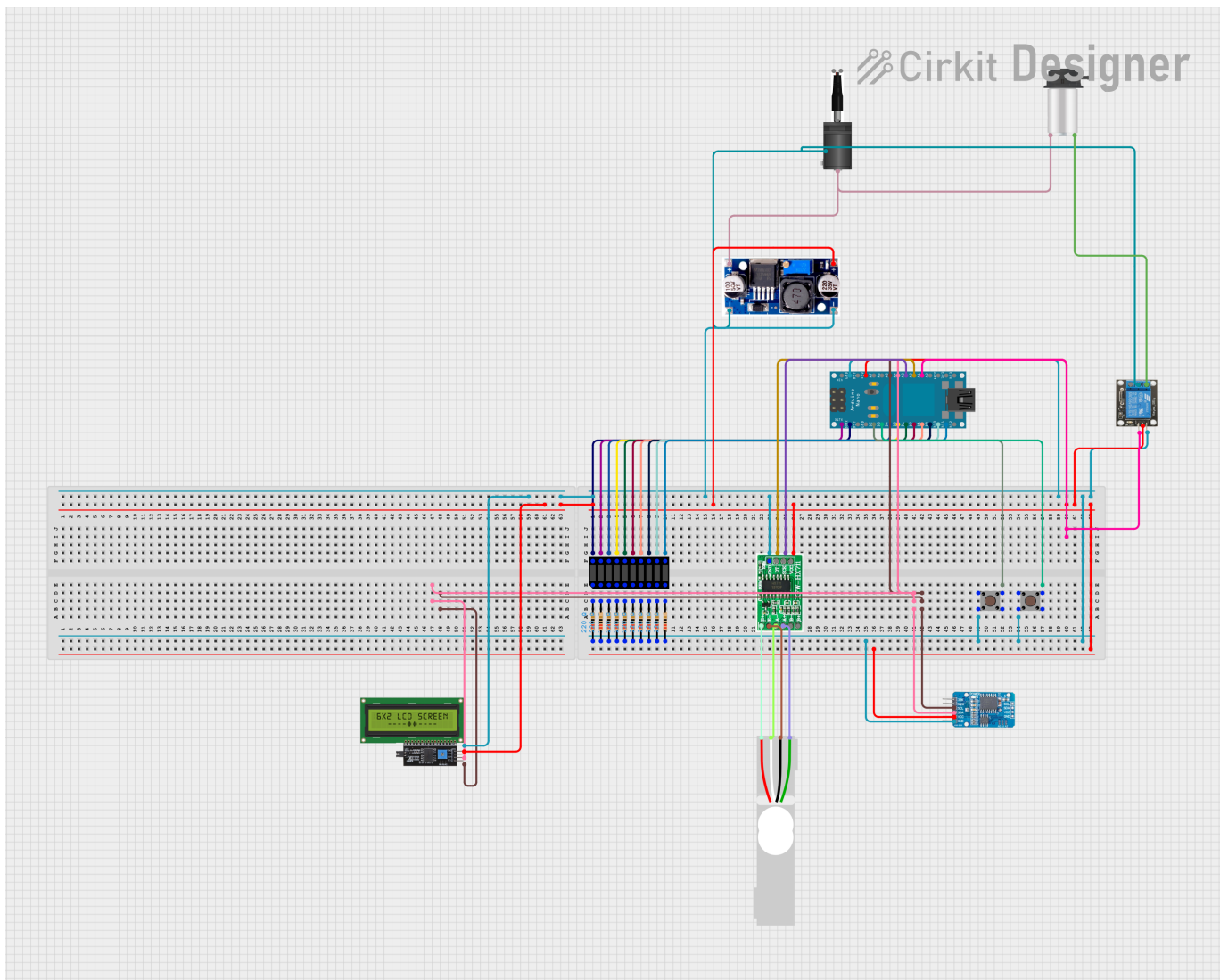
Lista de piese

Componenta	Model	Cantitate
Microcontroler	ATmega328P-XMINI	1
Modul Cântar Electronic 1kg	YZC-133	1
Modul Citire Greutate	HX711	1
Display LCD 2×16 I2C	1602	1
Pompă peristaltică mini 12VDC	Kamoer CKP-DC-S08	1
Modul Releu 1 canal 5V	SRD-05VDC-SL-C	1
Sursă alimentare 12V 2A	YDS-WA024	1
Buton 6x6x6	Sunrom 5827	4
Bară leduri cu 10 segmente	F2510BH	1
Rezistență	220 Ω	12
Rezistență	1k Ω	4
Rezistență	10k Ω	4
Condensator	0.1μF	4
Modul RTC	DS3231	1
Baterie	Varta CR2032	1
Breadboard	830 puncte	1
Fire DuPont M-F	10 bucăți 10cm	2
Fire DuPont M-M	10 bucăți 20cm	3
Furtun silicon	1.6m	1
Modul DC-DC Step Down	LM2596	1
Mufă alimentare panou DC mamă	5.5mm x 2.1mm 3 pini	1

Pini folosiți

Componentă	Pin componentă	Pin ATmega328P (Arduino)	Explicație
Sursă 12V	V+ (Plus)	-	Tensiunea de 12V merge la intrarea IN+ a convertorului Step-Down și la (+) pompei
Sursă 12V	V- (Minus)	-	Se conectează la IN- (convertor) și la borna COM a releului
Convertor Step-Down LM2596	OUT+	5V	Furnizează 5V către magistrala breadboard-ului și pinul 5V al plăcii
Convertor Step-Down LM2596	OUT-	GND	Conectat la magistrala de masă (GND) a breadboard-ului
Pompă de apă 12V	Minus (-)	-	Conectat la borna NO a releului
Modul Releu 5V	VCC (DC+)	5V	Alimentarea părții logice a releului cu 5V
Modul Releu 5V	GND (DC-)	GND	Masă comună
Modul Releu 5V	S	PC1	Pin configurat ca OUTPUT digital pentru a da comanda de închidere a circuitului
Senzor Greutate 1kg	Fir Roșu	-	Se conectează la borna E+ (Excitation Plus) a modului HX711 pentru alimentarea cu curent de referință
Senzor Greutate 1kg	Fir Negru	-	Se conectează la borna E- (Excitation Minus) a modului HX711, reprezentând masa senzorului
Senzor Greutate 1kg	Fir Alb	-	Se conectează la borna A- (Analog Minus) a modului HX711. Este primul fir de semnal care citește variația de tensiune
Senzor Greutate 1kg	Fir Verde	-	Se conectează la borna A+ (Analog Plus) a modului HX711. Al doilea fir de semnal - împreună cu cel alb creează diferența de potențial amplificată de modul
HX711 (Senzor Greutate)	VCC	5V	Modulul este alimentat la 5V de la magistrala breadboard-ului
HX711 (Senzor Greutate)	GND	GND	Toate componentele trebuie să aibă masă comună
HX711 (Senzor Greutate)	DT / DAT	PC2	Pin analogic folosit ca I/O digital pentru citirea datelor
HX711 (Senzor Greutate)	SCK / CLK	PC3	Pin analogic folosit ca I/O digital pentru semnalul de clock
LCD 16x2 I2C	VCC	5V	Display-ul este alimentat la 5V
LCD 16x2 I2C	GND	GND	Masă comună
LCD 16x2 I2C	SDA	PC4	Pinul hardware standard dedicat pentru date I2C
LCD 16x2 I2C	SCL	PC5	Pinul hardware standard dedicat pentru clock I2C
RTC DS3231	VCC	5V	Modulul ceasului este alimentat la 5V
RTC DS3231	GND	GND	Masă comună
RTC DS3231	SDA	PC4	Conectat în paralel pe magistrala I2C cu LCD-ul
RTC DS3231	SCL	PC5	Conectat în paralel pe magistrala I2C cu LCD-ul

Buton 1 (Start/Pauză)	Semnal	PD2	Pin dedicat pentru întreruperi hardware (INT0). Configurat cu INPUT_PULLUP
Buton 2 (Schimbare Afișaj)	Semnal	PD3	Pin dedicat pentru întreruperi hardware (INT1). Configurat cu INPUT_PULLUP
Bară 10 LED-uri	Anozi (+)	PD0, PD1, PD4-PD7, PB0-PB3	Pini digitali folosiți pentru controlul progresului
Bară 10 LED-uri	Catozi (-)	GND	Masă comună, prin intermediul rezistențelor



Software Design

Am terminat partea de software design, integrând toate funcționalitățile menționate mai sus.

Am folosit bibliotecile:

- HX711 pentru a putea conecta senzorul de greutate la breadboard, cât și pentru a putea interpreta datele primite
- LiquidCrystal_I2C pentru a afișa diferite mesaje pe ecranul LCD

- RTCLib pentru a putea realiza statisticile menționate, spre exemplu câtă apă am băut în ziua respectivă
- SoftPWM pentru a varia intensitatea ledurilor
- EEPROM pentru a putea salva statisticile chiar și dacă dozatorul nu mai este alimentat
- Wire pentru comunicarea I2C

Elementele de noutate ale proiectului sunt reprezentate de funcționalitățile diverse:

- utilizatorul poate seta la început volumul sticlei folosite (de la 50 ml la 1000 ml, increment de 50 ml)
- după ce se așază sticla cu apa inițială, senzorul detectează exact masa, luând în calcul masa sticlei goale
- la pornirea dozatorului, progresul este afișat atât pe baza LED-urilor (acestea se aprind alternativ, variind intensitatea pe baza volumului rămas), cât și prin intermediul ecranului LCD (afișează volumul rămas)
- există mai multe moduri de afișaj (volumul rămas sau cantitatea de apă consumată în ziua respectivă, acest lucru fiind foarte important în nutriție)
- dozatorul se poate pune pe pauză în orice moment, respectiv se poate relua activitatea rămasă
- după ce sticla se umple, se poate pune o nouă sticlă, fără a fi necesară repornirea aparatului

Codul propriu-zis se bazează pe un automat de stări circular, astfel:

- starea 0 (selecția) - se citește starea butoanelor. În această etapă, butoanele sunt utilizate pentru selectarea volumului sticlei
- starea 1 (așteptare) - se așteaptă punerea sticlei pe cântar și calcularea masei
- starea 2 (umplere) - microcontrolerul acționează releul, pornind pompa
- starea 3 (pauză) - pompa este oprită, iar totul (LCD, releul etc.) este pus pe pauză, în standby
- starea 4 (finalizarea) - pompa s-a oprit, se salvează statisticile în EEPROM și se așteaptă ca sticla să fie ridicată. După ce se ridică sticla, se pornește din nou din starea 0

Senzorul de greutate a fost calibrat pe baza unui cod software. Acesta poate fi vizualizat pe [git](#). Am pus pe cântar o sticlă cu masă cunoscută (250 ml), iar apoi am reglat, prin UART, factorul de calibrare din software.

Optimizări:

- întreruperi externe (ca să nu facem polling) - proiectul este destul de complex. Fără întreruperi nu puteam detecta simultan și masa sticlei, și starea butonului etc.
- debouncing (software) - pentru a se înregistra o singură apăsare
- variabile volatile - pentru a nu fi optimizat accesul de către compilator

- timere în loc de delay - pentru a nu bloca execuția inutil

- accesul EEPROM - modificăm cantitatea de apă băută numai după ce sticla s-a umplut complet, pentru a proteja memoria sensibilă EEPROM

Ca mediu de dezvoltare am folosit VS Code cu extensia PlatformIO, framework Arduino.

Rezultate Obținute


[Demonstrație Video Prima Versiune](#)

[Demonstrație Video A Doua Versiune](#)

[Link github](#)

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

Am terminat proiectul

Bibliografie/Resurse

Resurse Hardware (datasheets)

[ATmega328P](#)

[ATmega328P Xplained Mini](#)

[Modul Cântar Electronic 1kg YZC-133](#)

[Modul Citire Greutate HX711](#)

[Display LCD 2×16 I2C 1602](#)

[Pompă peristaltică mini 12VDC Kamoer CKP-DC-S08](#)

[Modul Releu 1 canal 5V SRD-05VDC-SL-C](#)

[Sursă alimentare 12V 2A](#)

[Modul RTC DS3231](#)

[Modul DC-DC Step Down LM2596](#)

Resurse Software

[PlatformIO Documentație](#)

[HX711](#)

[LiquidCrystal_I2C](#)

[RTCLib](#)

[SoftPWM](#)

[EEPROM](#)

[Wire](#)

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/aureliu.antonie>



Last update: **2026/05/22 21:33**