

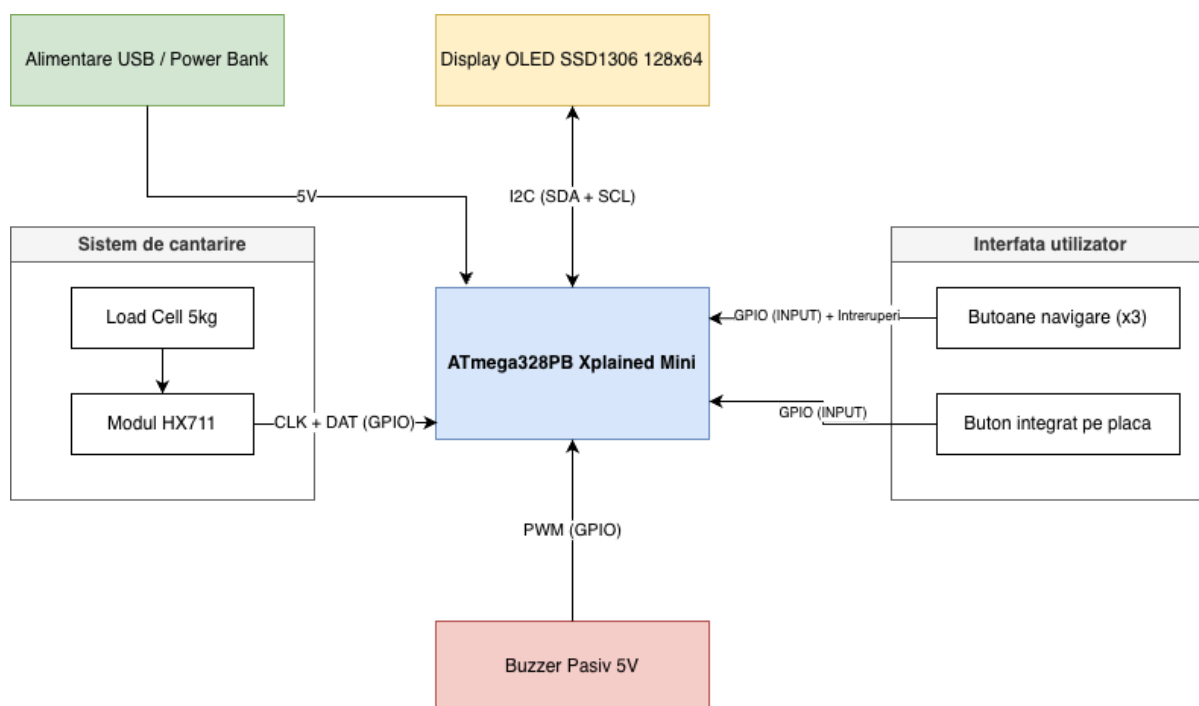
# Monitor de Hidratare

## Introducere

**Monitor de Hidratare** este un dispozitiv compact de monitorizare a consumului zilnic de apa.

- **Ce face:** Sistemul functioneaza ca un cantar inteligent pe care utilizatorul asaza sticla de apa. Dispozitivul masoara in timp real greutatea sticlei si calculeaza automat cantitatea de apa consumata, afisand progresul pe un ecran OLED. Un buzzer emite notificari sonore periodice pentru a reaminti utilizatorului sa bea apa.
- **Scopul proiectului:** Hidratarea corecta este esentiala pentru sanatate, dar majoritatea oamenilor uita sau nu stiu exact cat au baut intr-o zi. Dispozitivul automatizeaza acest proces fara ca utilizatorul sa faca nimic altceva decat sa aseze sticla pe platforma.
- **Ideea de pornire:** Proiectul a pornit de la nevoia personala de a monitoriza consumul de apa intr-un mod pasiv, fara aplicatii de telefon sau introducere manuala a datelor.
- **Utilitate:** Dispozitivul este util oricui doreste sa isi formeze obiceiuri sanatoase de hidratare, in special persoanelor care petrec mult timp la birou sau la calculator.

## Descriere generală



### Module si interactiuni:

- **ATmega328PB:** unitatea centrala de procesare. Citeste datele de la HX711, calculeaza consumul de apa, actualizeaza display-ul si gestioneaza butoanele si buzzerul.
- **Load Cell 5kg + HX711:** senzorul de greutate trimite un semnal analogic catre HX711, care il amplifica si il convertește la semnal digital pe 24 biti, transmis catre microcontroller prin 2 fire (CLK + DAT).
- **Display OLED SSD1306:** afiseaza informatii despre greutatea curenta, cantitatea de apa consumata si obiectivul zilnic. Comunicare prin I2C.
- **Butoane:** permit utilizatorului sa seteze obiectivul zilnic de hidratare, sa efectueze functia tare (zero) si sa navigheze in meniu.

- **Buzzer pasiv:** emite semnale sonore la intervale regulate (reminder de hidratare) sau la atingerea obiectivului zilnic. Controlat prin semnal PWM.

## Laboratoare utilizate

---

- **Lab 0 – GPIO** — butoanele de navigare si modulul buzzer sunt conectate prin pini GPIO
- **Lab 2 – Intreruperi** — apasarea butoanelor este detectata prin intreruperi externe, nu prin polling
- **Lab 3 – Timere si PWM** — buzzerul pasiv este controlat prin semnal PWM, iar remindererele de hidratare sunt generate cu un timer periodic
- **Lab 6 – I2C** — comunicarea cu display-ul OLED SSD1306 se face prin protocolul I2C

## Hardware Design

---

### Lista de piese

| Nr. | Componenta                | Cantitate | Rol in proiect   |
|-----|---------------------------|-----------|--|
| 1   | Load Cell 5kg             | 1 buc.    | Senzorul care simte cat cantareste sticla in fiecare moment                      |
| 2   | Modul HX711               | 1 buc.    | Preia semnalul slab de la load cell si il face utilizabil pentru microcontroller |
| 3   | Ecran OLED SSD1306 0.96"  | 1 buc.    | Afiseaza progresul de hidratare in timp real                                     |
| 4   | ATmega328PB Xplained Mini | 1 buc.    | Creierul intregului sistem, coordoneaza toate celelalte componente               |
| 5   | Modul Buzzer Pasiv 5V     | 1 buc.    | Reaminteste utilizatorului sa bea apa daca trece prea mult timp fara consum      |
| 6   | Butoane PCB Mini 6x6x5mm  | 3 buc.    | Permit setarea obiectivului zilnic si resetarea cantarului                       |
| 7   | Kit rezistente            | 1 set     | Rezistente de pull-up pentru butoane   |
| 8   | Fire Dupont M-F           | 1 set     | Leaga modulele externe la placa fara lipire                                      |
| 9   | Fire breadboard M-M       | 1 set     | Conexiuni rapide intre componente pe breadboard                                  |
| 10  | Breadboard 830 pini       | 1 buc.    | Permite testarea circuitului fara lipire permanenta                              |
| 11  | Cablu Micro-USB           | 1 buc.    | Alimentare si programare placa de la laptop sau power bank                       |

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png> [<http://www.captain.at/electronic-atmega16-mmc-schematic.png>])
- diagrame de semnal
- rezultatele simulării

## 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 plănuți să le implementați
- (etapa 3) surse și funcții implementate

## 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

pm/prj2026/atoader/cristiana.tabacaru.txt · Last modified: 2026/05/09 19:38 by cristiana.tabacaru