# Dispenser automat de medicamente

## Introducere

În viața de zi cu zi, sarcinile simple pot deveni poveri copleșitoare pentru anumite persoane, mai ales pentru cele care se confruntă cu probleme de sănătate sau cu limitări cognitive. Administrarea medicamentelor la ore fixe, o rutină banală pentru unii, devine pentru bolnavi o provocare constantă, o luptă tăcută cu timpul. Această dificultate este resimțită cu atât mai acut de pacienții care suferă de afecțiuni precum boala Alzheimer, unde uitarea administrării tratamentului poate avea consecințe grave.

Pornind de la această nevoie reală, proiectul propus urmărește realizarea unui dispenser automat de pastile. Acesta va fi capabil să depoziteze pastilele și să le elibereze automat la orele programate. Pentru a asigura notificarea eficientă a utilizatorului, dispozitivul va utiliza semnalizare vizuală prin LED-uri și semnalizare auditivă printr-un buzzer. Scopul acestui proiect este de a oferi un sprijin esențial pacienților care trebuie să urmeze tratamente medicamentoase stricte, reducând astfel riscul uitării sau al administrării incorecte.

## Descriere generală

Lista de componente:

- Arduino Uno R3
- Breadboard HQ
- Fire de conectare
- LED-uri
- Buzzer activ de 5V
- Servomotor SGG92R
- Display SPI
- LCD 1602 cu interfata I2C
- Tastatura matriceala 4×4
- Modul ceas in timp real DS1307
- Container medicamente

Descriere componente:

- Arduino Uno R3: Microcontrolerul central care coordoaneaza intreg sistemul.
- Tastaura matriceala: Permite utilizatorului sa introduca orele si setarile dorite.
- LCD 1602 CU I2C: Afiseaza mesaje si confirmari pentru utilizatori.
- Modul ceas RTC DS1307: Pastreaza timpul exact, esential pentru declansarea eliberarii pastilelor.
- LED-uri: Ofera notificari vizuale
- Buzzer: Emite semnal sonor cand e timpul sa se administreze o pastila.
- Servomotor SGG92R: Roteste containerul pentru a selecta compartimentul corect.
- Container compartimentat: Stocheaza pastilele pe zile/ore (ex: 3-4 comportimente).

Interactiunea dintre componente
1) Utilizatorul introduce ora de administrare si setarile folosind tastatura matriceala. Acesta poate alege daca medicamentele sunt administrate o singura data sau zilnic recurent.

2) Arduino Uno preia inputul si il salveaza, folosind modulul RTC pentru a verifica constant ora curenta.

3) La ora programata:

1. LED-ul se aprinde pentru a capta atentia vizuala.
2. Buzzerul emite un sunet pentru a alertare sonora.
3. Pe LCD este afisat un mesaj de notificare.

4) Eliberare automata:

1. Arduino calculeaza ce compartiment corespunde orei programate.
2. Trimite semnal catre servomotor, care roteste containerul la pozitia corecta (ex: 0, 90, 180, 270).
3. Se deschide slotul corespunzator si pastila poate fi luata.

Schema bloc

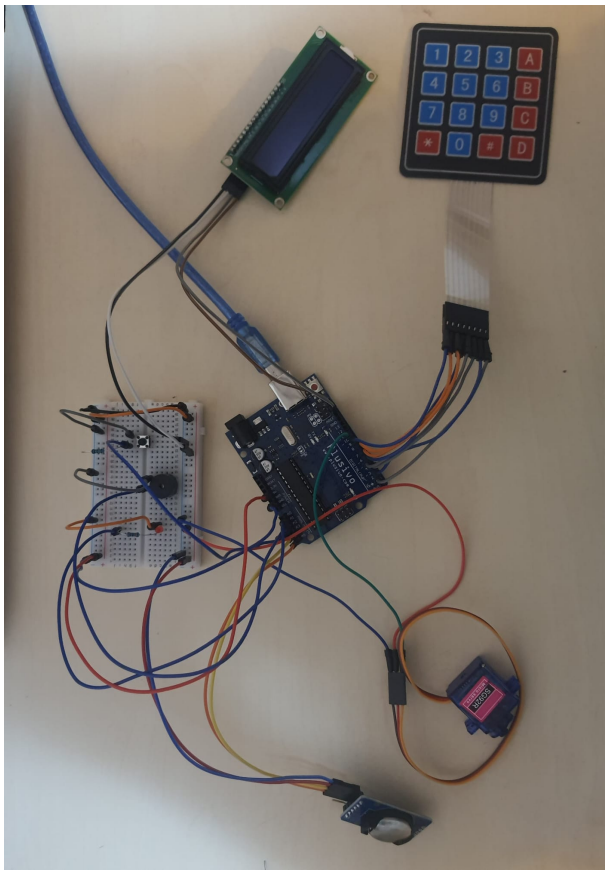
# Hardware Design

Componente

| Nr. crt. | Componentă | Cantitate | Link / Datasheet |
|---|---|---|---|
| 1 | Arduino UNO R3 ATmega328P | 1 | [Link](https://www.optimusdigital.ro/ro/placi-avr/4561-placa-de-dezvoltare-compatibila-cu-arduino-uno-r3-atmega328p-atmega16u2-cablu-50-cm.html?search_query=Placa+de+Dezvoltare+Compatibila+cu+Arduino+UNO+R3+%28ATmega328p+++ATmega16u2%29+++Cablu+50+cm&results=1) |
| 2 | Servomotor SG92R 9g 2.5 kg.cm, 4.8 V | 1 | [Link](https://www.optimusdigital.ro/ro/motoare-servomotoare/5706-servomotor-sg92r-9g-25-kgcm-48-v.html?search_query=Servomotor+SG92R+9g+2.5+kg.cm%2C+4.8+V&results=1) |
| 3 | Tastatura matriceala 4×4 cu conector pin de tip mama | 1 | [Link](https://www.optimusdigital.ro/ro/senzori-senzori-de-atingere/470-tastatura-matriceala-4x4-cu-conector-pin-de-tip-mama.html?search_query=Tastatura+matriceala+4x4+cu+conector+pin+de+tip+mama&results=1) |
| 4 | LCD 1602 cu Interfata I2C si Backlight Albastru | 1 | [Link](https://www.optimusdigital.ro/ro/optoelectronice-lcd-uri/2894-lcd-cu-interfata-i2c-si-backlight-albastru.html?search_query=LCD+1602+cu+Interfata+I2C+si+Backlight+Albastru&results=2) |
| 5 | Modul cu Ceas In Timp Real DS3231 | 1 | [Link](https://www.optimusdigital.ro/ro/altele/1102-modul-cu-ceas-in-timp-real-ds3231.html?search_query=Modul+cu+Ceas+in+Timp+Real+DS3231&results=3) |
| 6 | Breadboard HQ (400 Points) | 1 | [Link](https://www.optimusdigital.ro/ro/altele/1102-modul-cu-ceas-in-timp-real-ds3231.html?search_query=Modul+cu+Ceas+in+Timp+Real+DS3231&results=3) |

| | | | |
|---|---|---|---|
| 7 | LED Roșu de 3 mm cu Lentile Difuze | 1 | [Link](https://www.optimusdigital.ro/ro/optoelectronice-led-uri/696-led-rou-de-3-mm-cu-lentile-difuze.html?search_query=LED+Ro%C8%99u+de+3+mm+cu+Lentile+Difuze&results=2) |
| 8 | Buzzer Activ de 5 V | 1 | [Link](https://www.optimusdigital.ro/ro/audio-buzzere/633-buzzer-activ-de-5-v.html?search_query=Buzzer+Activ+de+5+V&results=18) |
| 9 | Buton 6x6x6 | 1 | [Link](https://www.optimusdigital.ro/ro/butoane-i-comutatoare/1119-buton-6x6x6.html?search_query=Buton+6x6x6&results=1) |

Schema electrica



# Software Design

# Stadiu actual al implementarii software

Proiectul este complet funcțional și a fost testat integral în condiții reale. Interfața permite programarea a până la 3 compartimente, fiecare cu ora și minut de activare, cu opțiune de repetare. Se utilizează un ecran LCD I2C pentru afișare, o tastatură 4×4 pentru introducerea datelor, și un modul RTC DS3231 pentru timp real. Sistemul deschide automat compartimentul potrivit folosind un servomotor, și oferă feedback sonor și vizual prin buzzer și LED.

# Mediu de dezvoltare utilizat:

- Platforma utilizata: Arduino IDE

- Limbaj folosit: C++
- Microcontroler: ATmega328P
- Toolchain: AVR-GCC + Arduino Core

# Librarii si software extern:

1. `Wire.h`

- Comunicatie I2C intre Arduino si componentele externe (LCD, RTC).

2. `LiquidCrystal_I2C.h`

- Controlul LCD-ului 16×2 cu interfata I2C.

3. `Servo.h`

- Controlul servomotorului pentru deschiderea compartimentelor

4. `Keypad.h`

- Interfatare usoara cu tastatura 4×4 pentru introducerea timpilor si comenzilor.

5. `RTClib.h`

- Utilizata pentru comunicarea cu modulul de timp real (DS3231) si gestionarea programarii.

# Algoritmi si concepte folosite:

**FSM (Finite State Machine)** pentru logica de interactiune:

- Stari predefinite: *ASTEPTARE*, *INTRO_HHMM*, *INTRO_RECURENTA*
- Permite o interfata clara cu utilizatorul.

#### Starea ASTEPTARE

Aceasta este starea implicita, in care dispozitivul:

- Afiseaza mesajul “Apasa butonul” pe LCD.
- Permite apasarea tastelor pentru acces rapid:
  - Tasta A: Afiseaza numarul de compartimenre libere.
  - Tasta B: Afiseaza programarile existente pentru compartimentele ocupate.
  - Tasta C: Afiseaza ora curenta de la modulul RTC.
- Detecteaza apasarea fizica a butonului (BUTTON pe A3), moment in care se intra in modul de programare (`INTRO_HHMM`), daca mai sunt compartimente disponibile.

#### Starea INTRO_HHMM

In aceasta stare:

- Utilizatorul introduce o ora de forma HHMM de la tastatura numerica.
- Ora introdusa este validata si afisata partial pe LCD.
- Apasarea tastei # finalizeaza introducerea si trece in starea `INTRO_RECURENTA`.

**Starea `INTRO_RECURENTA`**

- Aici utilizatorul introduce daca programarea este recurenta(`1`) sau unica (`0`).
- Programarea este salvata in vectorul `compartimente[]`, iar sistemul deschide compartimentul temporar pentru plasarea medicamentelor.
- Se revine in ASTEPTARE sau se incheie daca toate cele 3 compartimente sunt programate.

**Loop-ul principal:**

- Verifica ora curenta.
- Deschide compartimentul daca ora/minutul se potrivesc.
- Revine la pozitia initiala dupa delay.

**Debouncing software** pentru tastatura si buton:

- folosind delay() scurt

**Manipulare directa** a registrelor (PORTC, PINC, DDRC) pentru:

- performanta crescuta
- control fin asupra buzzerului, LED-ului si butonului

**Programare cu RTC:**

- fiecare compartiment este activat automat in functie de ora si minut, cu obtiunea de executie unica sau recurenta.

**Servo control secvential:**

- fiecare compartiment are un unghi predefinit (60, 120, 180), corespunzator rotatiei pentru deschidere.

# Structuri de date:

```
struct Programare {
  int ora;
  int minut;
  bool recurent;
};
```

Este folosit pentru a stoca programarile fiecarui compartiment. Vectorul :

```
Programare compartimente[3];
```

reprezinta cele 3 compartimente ale sistemului.

## Notiuni preluate din laboratoare:

- GPIO: manipulat direct prin registre pentru LED, buzzer, buton.
- USART: pentru debugging si afisarea informatiilot in Serial Monitor.
- I2C: utlizat pentru comunicarea cu LCD I2C, RTC.
- Timer: configurat manual pentru a controla exact durata buzzerului si deschiderea servomotorului.
- ISR(Intreruperi): utilizate pentru a declansa actiuni ppe baza de timp, fara blocarea buclei `loop()`.

## Rezultate Obţinute

DEMO: [Link](https://www.youtube.com/watch?v=778pTGQ6WSM&ab_channel=AlexandraElena)

Github: [Link](https://github.com/alexandra200312/Dispenser_medicamente)

## Concluzii

A fost un proiect solicitant. Am constientizat ca am nevoie neaparat de o surubelnita in casa.

## Download

dispenser_de_pastile.zip

## Jurnal

# Bibliografie/Resurse


- [Link](
https://docs.arduino.cc/resources/datasheets/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf)
- [Link](https://www.youtube.com/watch?v=zvCdkV52cis&ab_channel=RosieResearch)
- [Link](https://www.youtube.com/shorts/9kcwwIKglWU?feature=share)


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

Permanent link:
**http://ocw.cs.pub.ro/courses/pm/prj2025/ccristi/alexandra.dragan**

Last update: **2025/05/28 06:23**