

Dozatorul Nu-ma-uita

Introducere

Dozatorul Nu-ma-uita este un sistem conceput pentru a automatiza procesul de administrare a vitaminelor si medicamentelor, asigurand respectarea intervalelor stricte de tratament recomandate de doctor(fierul dimineata inainte de micul dejun, vitamina D dupa micul dejun si magneziul seara). Dispozitivul permite sortarea prealabila a dozelor in compartimente si eliberarea automata a acestora la ore prestabilite. Sistemul monitorizeaza prezenta pastilei in zona de administrare si alerteaza utilizatorul sonor si vizual pana la confirmarea tratamentului.

Descriere generala

1. Module Hardware

Alimentare: Sursa de 5V DC asigura tensiunea stabila pentru microcontroler si senzorul IR, in timp ce un regulator de tensiune, daca este necesar, protejeaza circuitele logice de consumul instantaneu al servomotorului.

Procesare: Microcontrolerul Arduino Uno 16U2 este unitatea centrala care gestioneaza timpul, proceseaza comenzile primite prin interfata seriala si controleaza elementele mecanice.

Intrari: Senzor IR: Detecteaza prezenta sau ridicarea pastilei din recipientul de colectare.

Interfata UART: Permite utilizatorului sa introduca programul de administrare.

Iesiri: Servomotor care actioneaza compartimentele de sortare prin semnale PWM pentru a elibera doza corecta.

Sistem de alerta (Buzzer si LED-uri): Oferă feedback acustic si vizual pentru a semnaliza momentele de administrare si starea sistemului.

2. Module Software

Driver UART: Gestioneaza receptia datelor de configurare, adica a orelor de administrare, si trimite log-uri de confirmare catre utilizator.

Timer: Utilizeaza un timer intern pentru a genera o baza de timp precisa, comparand constant ora curenta cu orele de administrare salvate.

Generator PWM: Moduleaza latimea impulsurilor pentru a controla unghiul servomotorului, permitand alinierea precisa a compartimentului de pastile cu jghebul de eliberare.

Rutina de intrerupere: O functie prioritara declansata de senzorul IR care detecteaza imediat interactiunea utilizatorului cu pastila, asigurand oprirea instantanee a alertelor.

3. Interactiunea

Procesul incepe cu programarea orelor de administrare prin interfata UART, datele fiind stocate in memoria microcontrolerului. Sistemul monitorizeaza timpul in timp real folosind un Timer intern, semnalizand starea de asteptare printr-un LED verde conectat la un pin GPIO. La ora stabilita, Arduino Uno 16U2 genereaza un semnal PWM pentru a roti servomotorul si a elibera pastila, moment in care se activeaza alertele vizuale si sonore. Ciclul se incheie doar atunci cand senzorul optic detecteaza ridicarea medicamentului, declansand o intrerupere externa care opreste imediat buzzerul si reseteaza dispozitivul pentru urmatoarea doza.

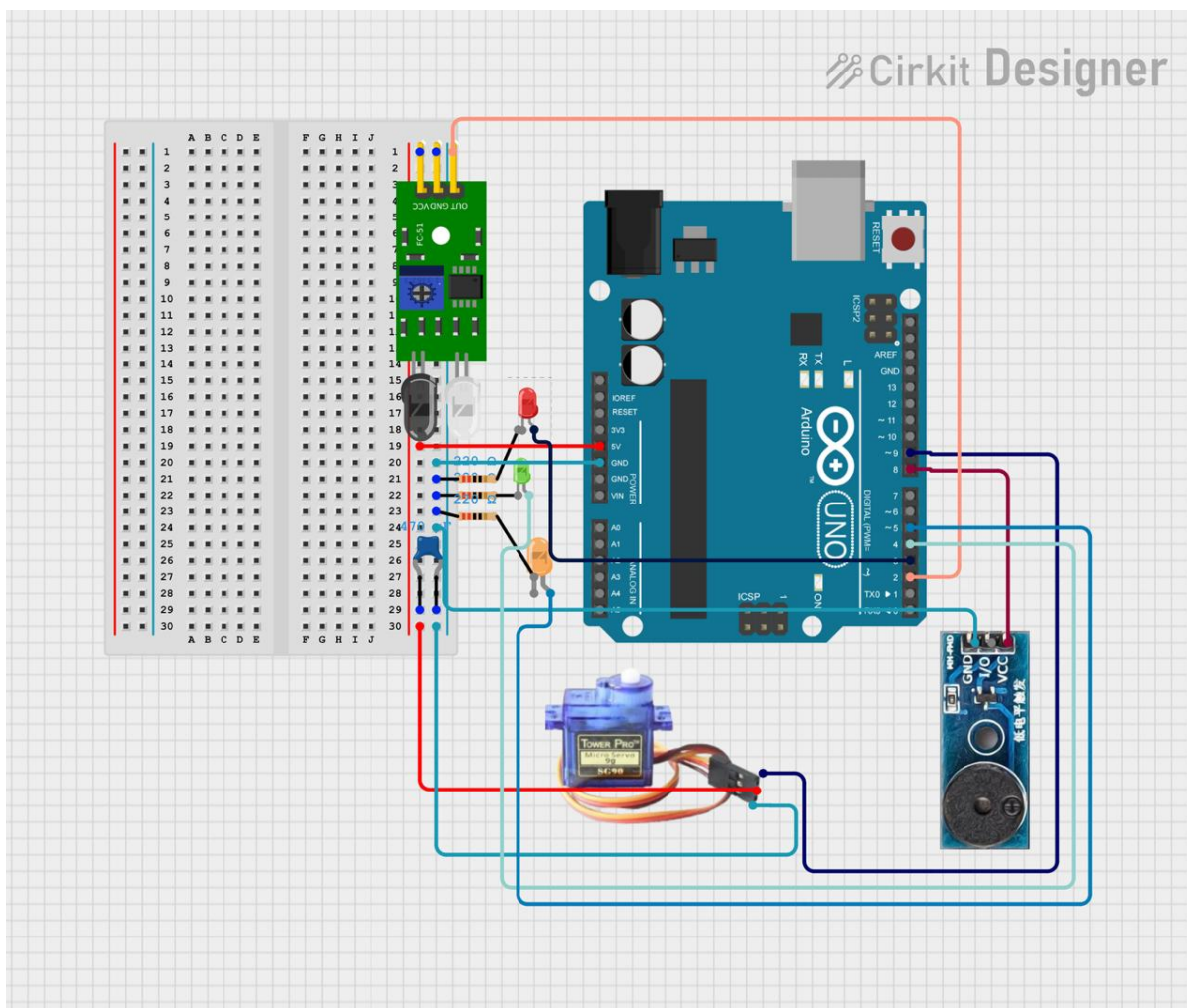


Hardware Design

Componenta	Cantitate	Tip / Model	Rol in proiect
Microcontroler	1	Arduino Uno 16U2	Unitatea centrala de procesare care gestioneaza timpul, senzorii, servomotorul si logica sistemului.
Servomotor	1	SG90 / MG996R	Controleaza mecanismul de eliberare a pastilelor prin rotatia compartimentelor.
Senzor IR / Bariera optica	1	Modul IR FC-51 / TCRT5000	Detecteaza prezenta sau ridicarea pastilei din recipientul de administrare.
Buzzer activ	1	5V Active Buzzer	Genereaza semnale sonore pentru notificarea utilizatorului la ora administrarii.
LED verde	1	LED 5mm	Semnalizeaza functionarea normala si starea de asteptare a sistemului.
LED rosu	1	LED 5mm	Indica existenta unei alerte sau a unei doze neadministrate.
LED galben	1	LED 5mm	Avertizeaza apropierea unei administrari programate.
Rezistente	3-5	220Ω / 330Ω	Protejeaza LED-urile si stabilizeaza semnalele electrice din circuit.
Sursa alimentare	1	5V DC Adapter	Furnizeaza energia necesara microcontrolerului si componentelor periferice.
Condensator electrolitic	1	470μF / 16V	Reduce fluctuatiile de tensiune produse de servomotor la pornire.
Breadboard / PCB	1	Breadboard MB-102	Permite realizarea conexiunilor electrice dintre componente.

Fire conexiune	set	Male-Male / Male-Female	Asigura conexiunile dintre modulele hardware si microcontroler.
Cablu USB	1	USB Type-B	Permite alimentarea placii Arduino si configurarea sistemului prin UART/Serial.

Schema Electrica



Servo Motor:

PWM → D9 +5V → + de pe breadboard GND → - de pe breadboard

Senzor IR:

VCC → + de pe breadboard GND → - de pe breadboard OUT → D2

Buzzer:

Pozitiv → D8 Negativ → - de pe breadboard

LED Verde:

Anod → D3 Catod → Rezistor 220Ω → - de pe breadboard

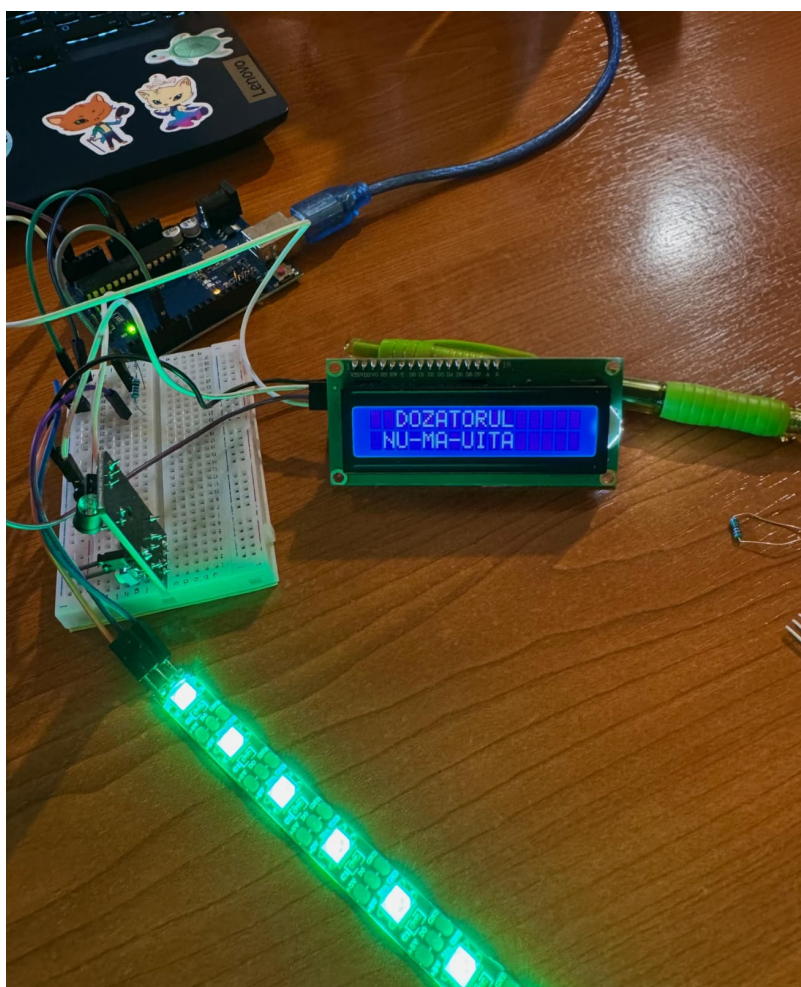
LED Galben:

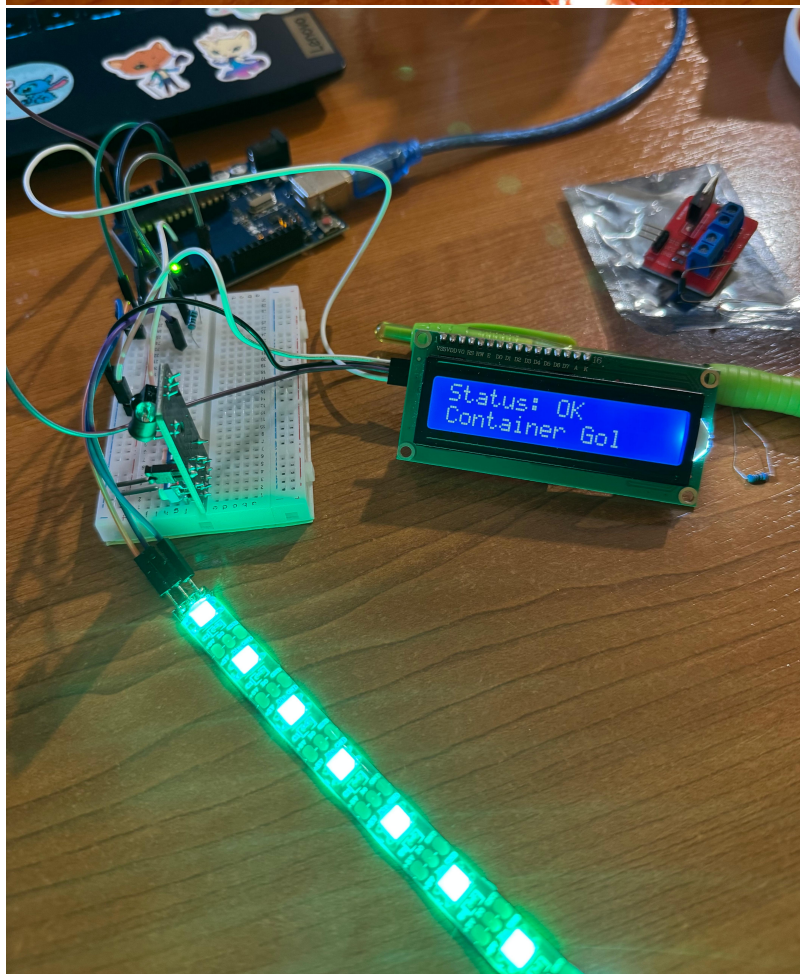
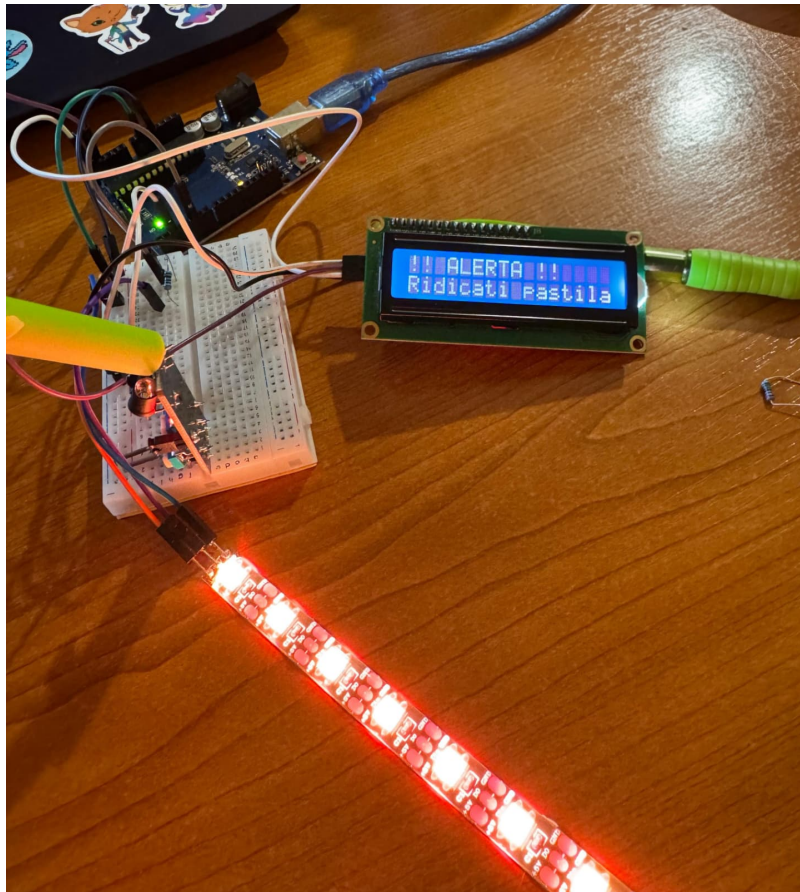
Anod → D4 Catod → Rezistor 220Ω → - de pe breadboard

LED Rosu:

Anod → D5 Catod → Rezistor 220Ω → - de pe breadboard

Imagini





Video demonstrativ

https://ctipub-my.sharepoint.com/:v:/g/personal/amarinescu0302_stud_acs_upb_ro/IQDb1TZ2S7NMSZ5OuOrSmtAsATJTHLpqKO6NYUTNvjqNkzo?nav=eyJyZWZlcnJhbEluZm8iOncicmVmZXJyYWxBcHAI0iJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldiYiIsInJmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=FQeUTS

Implementare Software

Implementarea software este complet functionala si stabila. Aplicatia este scrisa in limbajul C pentru microcontrolerul ATmega328P, transformand sistemul intr-un distribuitor inteligent de pastile autonom. Sistemul gestioneaza continuu un automat de stari care interpreteaza comenzi primite prin interfata seriala pentru a actiona un mecanism cu compartimente bazat pe servomotor. Concomitent, monitorizeaza activ un senzor de obstacole cu infrarosu IR folosind intreruperi hardware, controleaza o alarma acustica -buzzer- de tip Active-LOW si ofera feedback in timp real utilizatorului prin intermediul unui ecran LCD I2C si al unor LED-uri de stare.

Biblioteci alese

avr/io.h si avr/interrupt.h: Au fost selectate pentru a oferi acces direct la harta de registri a microcontrolerului ATmega328P de pe Arduino Uno (ex. DDRB, EIMSK, TCCR1A), asigurand manipularea bitilor cu zero overhead si gestionarea directa vectorilor de intrerupere hardware (ISR(INT0_vect));

util/delay.h: Utilizata pentru mici temporizari de debouncing si pentru a regla viteza de rotatie a servomotorului;

Wire.h si LiquidCrystal_I2C.h: Au fost incluse pentru a controla afisajul LCD 16x2 prin protocolul I2C;

Noutati

Principala noutate a acestui proiect consta in arhitectura sa de tip intrerupere hardware combinata cu filtrare software. In loc sa aiba incredere in datele brute ale senzorului sau să iroseasca ciclurile procesorului verificand constant un pin, sistemul foloseste o abordare hibrida:

La nivel hardware: Senzorul IR declanșează o întrerupere imediată pe procesor INT0 la orice schimbare logică, actualizând instantaneu un flag implementat cu volatile.

La nivel software: Bucla principala aplica un filtru strict de stabilitate temporala. O pastila este confirmata ca fiind eliberata doar daca senzorul citeste un semnal stabil si continuu timp de 300 de milisecunde, ignorand astfel zgomotul de semnal. De asemenea, este confirmata ca fiind preluata de

utilizator doar daca senzorul ramane liber timp de 500 de milisecunde.

Justificarea functionalitatilor de laborator

Laboratorul 0: Manipularea directa a registrilor PORTB si DDRB pentru a controla led-urile de stare (pinii PB2, PB4) si pentru a actiona buzzerul Active-LOW de pe pinul PB3.

Laboratorul 1 (USART): Protocolul de comunicare seriala este configurat manual la 9600 baud folosind registrii (UBRR0, UCSRA/B/C). Gestioneaza fluxul bidirectional de date (receptia comenzilor pentru compartimente si transmiterea log-urilor de diagnostic pe PC).

Laboratorul 2 (intreruperi externe): Implementat prin INT0 pe pinul PD2. Registrul de control (EICRA) este configurat sa declanseze o intrerupere la orice schimbare logica (ISC00=1, ISC01=0), permitand procesorului de pe Arduino Uno sa detecteze instantaneu atat sosirea, cat si preluarea pastilei.

Laboratorul 3 (Timere & PWM): Utilizează Timer-ul 1 (pe 16-biți) configurat in modul Fast PWM. Semnalul generat hardware actioneaza pe pinul PB1 (OC1A) pentru a pozitiona servomotorul.

Scheletul proiectului si interactiunea functionala

```

+-----+
|               Initializare hardware               |
| (usart_init() | pwm_servo_init() | gpio_init() | intreruperi_init()) |
+-----+
|
| v
+-----+
|               Bucla principala [while(1)]         |
+-----+
| Starea 0: in asteptare / asteapta comanda UART   |
+-----+
| (comanda primita: 1-4)                            |
| v
+-----+
| - aprinde LED-ul rosu (Avertizare miscare)       |
| - Mascheaza intreruperea INT0 (EIMSK &= ~(1 << INT0)) pt. a evita zgomotul |
| - Misca servo lent catre compartiment(muta_servo) |
| - Demascheaza intreruperea INT0 (Reactivare senzor) |
| - Asteapta max 10s validarea senzorului IR (filtru stabilitate 300ms) |
+-----+
|
| [Senzor Validat: FALS]                            |
| v                                                  |
+-----+
|
| [Senzor Validat: ADEVARAT]                        |
| v
+-----+
+-----+

```

```
| Starea 2: Eroare mecanica | | Starea 1: Alarma Activa  
| |  
| - tine LED-ul rosu APRINS | | - Declanșează Bip-uri scurte  
| |  
| - Afiseaza "EROARE BLOCAJ" | | - Clipeste LED-ul rosu (la  
500ms) | |  
| - Asteapta 5 secunde, Auto-Reset | | - Asteapta ridicarea pastilei  
| |  
+-----+  
+-----+  
|  
v (Pastila preluata)  
+-----+  
PB3) | | - Opreste Buzzerul (PORTB |=  
| | - Aprinde LED-ul Verde  
| | - Afisează "Status OK"  
| | - Revine in Starea 0  
| |  
+-----+
```

Eveniment de intrerupere asincrona

Independent de fluxul prezentat mai sus, procesorul de pe Arduino Uno evalueaza starea senzorului IR exclusiv in fundal. Daca pinul PD2 isi schimba starea logica in orice microsecunda, CPU-ul isi opreste temporar executia normala pentru a actualiza flag-ul de status brut: `ISR(INT0_vect) {`

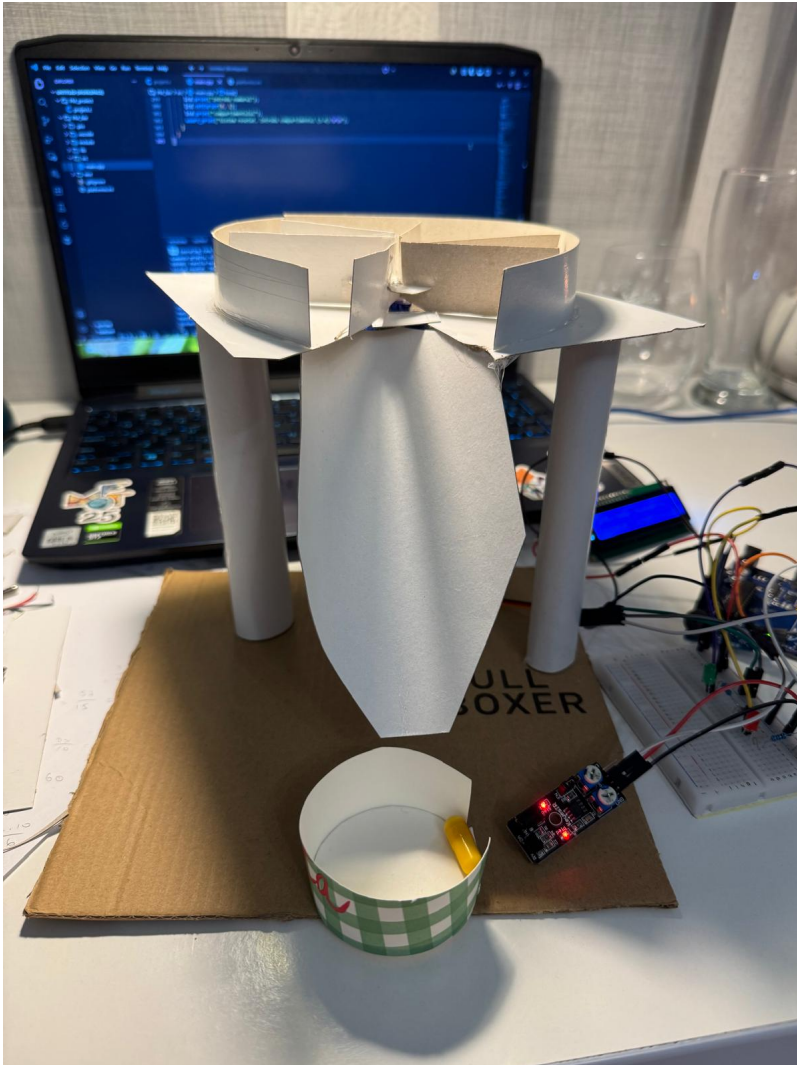
```
if ((PIND & (1 << PD2)) == 0) {  
    senzor_activ = 1; // pastila e detectata in suport  
} else {  
    senzor_activ = 0; // suportul e este gol  
}  
  
}
```

Ce trebuie imbunatatit

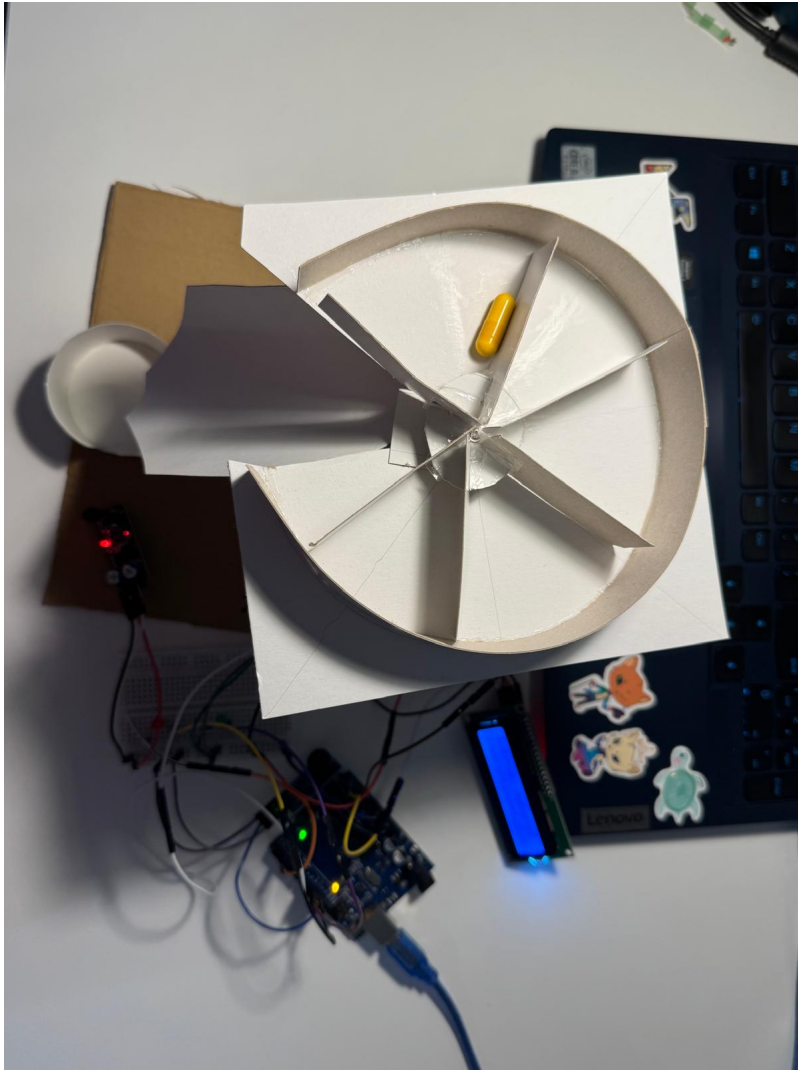
Senzorul IR trebuie calibrat astfel incat sa nu detecteze fundul sau peretii paharului,ci doar pastila. Voi adauga benzi led pentru feedback-ul vizual al utilizatorului in loc de cele 2 led-uri tip dioda.

Poze si video

servomotorul este lipit sub cerc, in mijloc si actioneaza fantele care imping pastilele intr-un tobogan, acestea cazand in pahar unde sunt detectate dupa de senzorul IR.







In video, este introdusa de la tastatura o cifra care indica pornirea dispozitivului. Cand automatul porneste, servomotorul invarte fantele, iar acestea imping pastila existenta in unul din compartimente. Pastila cade in pahar unde este detectata de senzorul IR. Buzzerul si led-ul rosu se activeaza pentru un feedback vizual si auditiv, iar pe display-ul LCD I2C apare un mesaj pentru utilizator "ridica pastila". Cand pastila este luata din pahar, senzorul IR se opreste, odata cu buzzerul si led-ul, iar pe ecran apare mesajul "Pastila ridicata, container gol". In video, senzorul este actionat cu mana, calibrarea acestuia facandu-se ulterior.

https://ctipub-my.sharepoint.com/:v:/g/personal/amarinescu0302_stud_acs_upb_ro/IQBwZuC0wOBcS5tocumC_ktPAaHzF-ELLkTitfASKDODtKA?nav=eyJyZWZlcnJhbEluZm8iOmsicmVmZXJyYWxBChAiOijPbmVEcmI2ZUZvckJ1c2luZXNzliwicmVmZXJyYWxBChBQbGF0Zm9ybSI6IldiYiIsInJlZmVycmFSTW9kZSI6InZpZXCiLCJyZWZlcnJhbFZpZXCiOijNeUZpbGVzTGlua0NvcHkifX0&e=ZZPUFt

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/alexandru.predescu/amarinescu0302>



Last update: **2026/05/24 10:39**