

Sistem Automat de Administrare a Medicamentelor

Introducere

Acest proiect reprezinta un sistem automat de administrare a medicamentelor. La ora programata, sistemul suna o alarma, roteste o roata compartimentata pentru a elibera pastilele zilei si umple un pahar cu apa folosind o pompa submersibila. Un ecran LCD afiseaza in permanenta ziua si ora curenta.

Scopul proiectului este de a automatiza complet procesul de administrare a medicatiei zilnice, eliminand necesitatea unei persoane care sa supravegheze sau sa reaminteasca acest lucru.

Ideea a pornit de la o situatie reala din familia mea. Bunicul meu ia medicamente zilnic si are nevoie ca cineva sa ii aminteasca sa le ia, sa ii pregateasca pastilele si sa stie ce zi este. Acest proiect incearca sa rezolve exact aceasta problema printr-un dispozitiv simplu de folosit si usor de inteles.

Consider ca un astfel de sistem poate fi util atat pentru persoanele in varsta care iau tratamente zilnice, cat si pentru familiile acestora, care nu mai trebuie sa fie prezente fizic pentru a se asigura ca medicatia a fost administrata corect si la timp.

Prezentarea pe scurt a proiectului vostru:

- ce face
- care este scopul lui
- care a fost ideea de la care ati pornit
- de ce credeti ca este util pentru altii si pentru voi

Descriere generala

Sistemul este compus din urmatoarele module hardware:

- **ATmega328P-XMINI** — microcontrollerul principal care coordoneaza toate modulele
- **RTC DS3231** — retine ora si data exacta, comunicatie prin I2C
- **LCD 1602 I2C** — afiseaza ora si ziua curenta, comunicatie prin I2C
- **Servo motor SG90** — roteste roata de pastile la ora programata, controlat prin PWM
- **Releu + pompa submersibila** — umple paharul cu apa, controlata printr-un pin digital
- **Buzzer activ 5V** — suna alarma la ora programata
- **Buton** — opreste alarma prin intrerupere hardware (INT0)

La ora programata, RTC-ul semnalizeaza microcontrollerul prin I2C. Buzzerul porneste si LCD-ul

afiseaza ora. Utilizatorul apasa butonul (intrerupere hardware) pentru a opri alarma. Simultan, servo-ul roteste roata de pastile, iar releul porneste pompa pentru un interval fix de timp pana cand paharul este plin.



O schema bloc cu toate modulele proiectului vostru, atat software cat si hardware insotita de o descriere a acestora precum si a modului in care interactioneaza.

Exemplu de schema bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>

Hardware Design

Lista de componente:

- ATmega328P-XMINI — microcontroller principal
- Modul RTC DS3231 — ceas timp real, comunicatie I2C
- LCD 1602 cu adaptor I2C — afisare ora si zi
- Servo motor SG90 — rotire roata de pastile
- Modul releu 5V cu optocuplor — control pompa
- Pompa submersibila 3-6V — umplere pahar cu apa
- Buzzer activ 5V — alarma sonora
- Modul buton microswitch — oprire alarma
- Breadboard 830 puncte
- Set fire jumper mama-tata/tata-tata
- Set rezistoare

Conexiuni principale:

Componenta	Pin componenta	Pin ATmega
RTC DS3231	VCC	5V
RTC DS3231	GND	GND
RTC DS3231	SDA	PC4 / A4
RTC DS3231	SCL	PC5 / A5
LCD 1602 I2C	VCC	5V
LCD 1602 I2C	GND	GND
LCD 1602 I2C	SDA	PC4 / A4
LCD 1602 I2C	SCL	PC5 / A5
Servo SG90	VCC (rosu)	Sursa externa 5V
Servo SG90	GND (maro)	GND
Servo SG90	Signal (portocaliu)	PB1 / D9
Modul releu	DC+	5V
Modul releu	DC-	GND
Modul releu	IN	PD7 / D7
Buzzer activ	+	PD6 / D6
Buzzer activ	-	GND

Buton	VCC	5V
Buton	GND	GND
Buton	S	PD2 / D2 (INT0)
Pompa	+	Releu contact NO
Pompa	-	GND

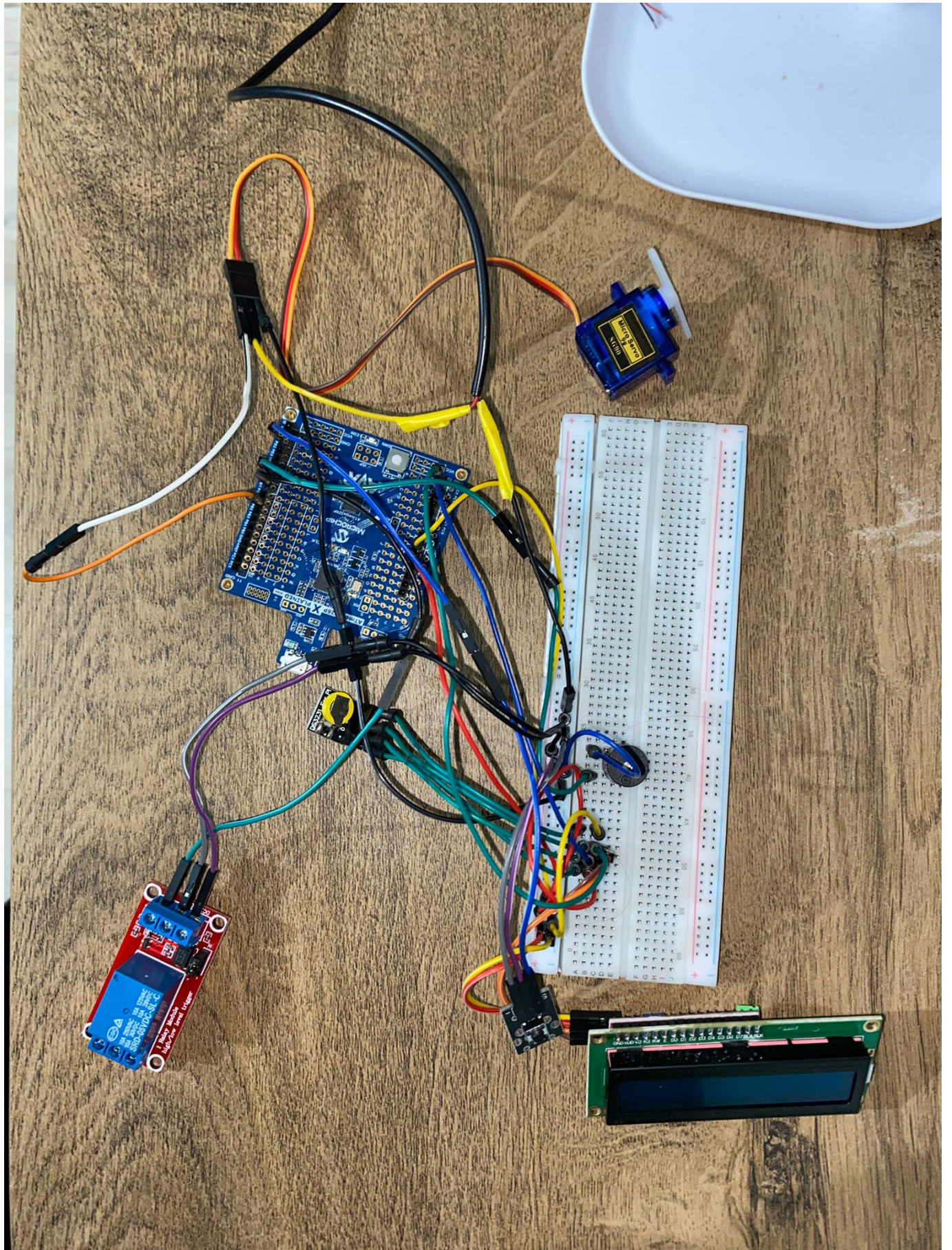
Obs: SDA si SCL sunt partajate intre RTC si LCD pe acelasi bus I2C. Servo-ul este alimentat dintr-o sursa externa de 5V, cu GND comun cu breadboard-ul.

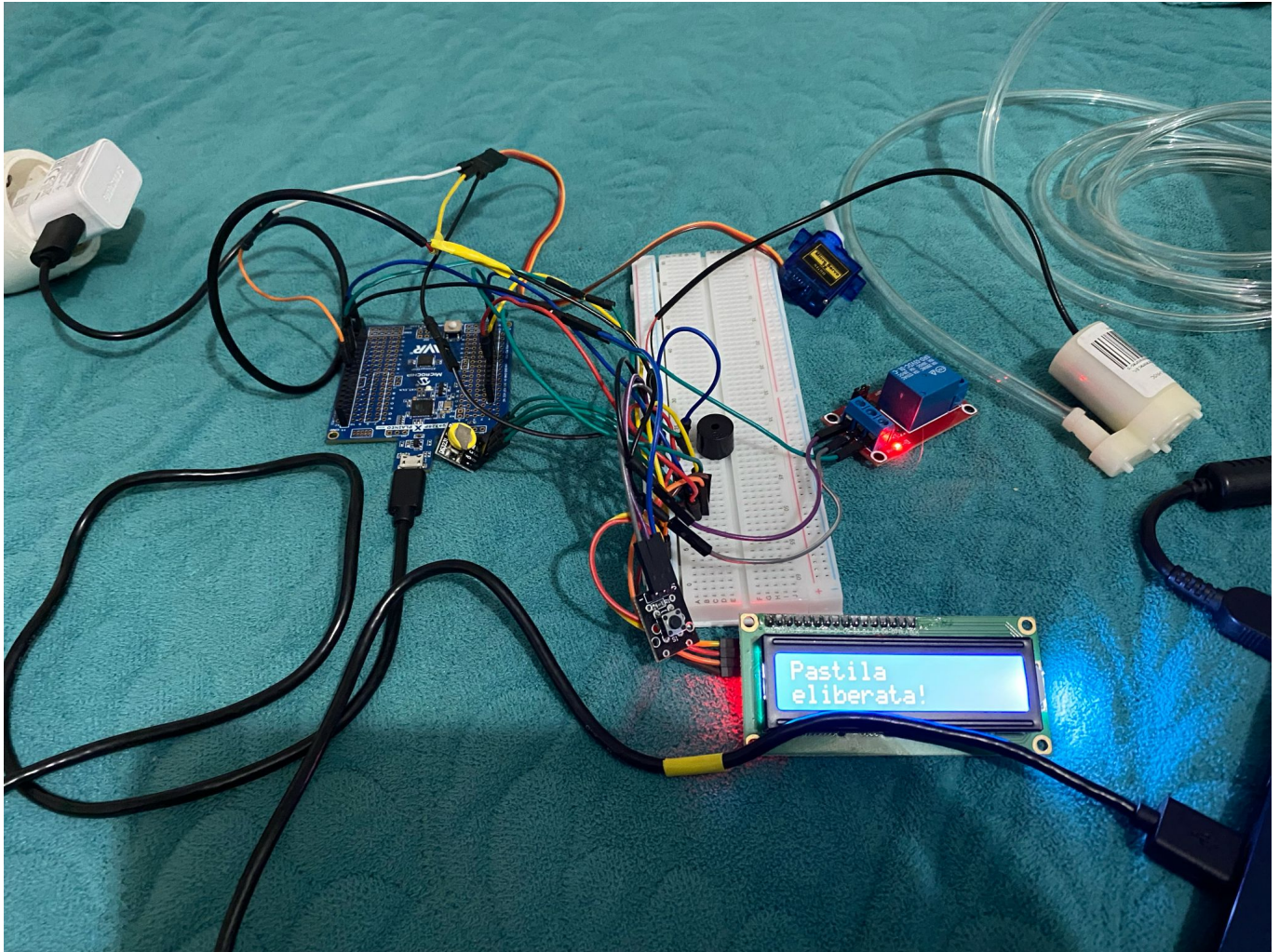
Schema electronica (Fusion 360):

In schema electronica realizata in Fusion 360, nu toate componentele proiectului au fost disponibile in libraria standard. Pentru modulele care nu au fost gasite (RTC DS3231, LCD 1602 I2C, modul releu, buzzer, buton), au fost folosite conectoare generice cu numarul exact de pini corespunzator fiecarui modul, etichetate cu numele componentei respective. Conexiunile electrice sunt identice cu cele din implementarea fizica.



Imagine hardware + testare:





Aici puneti tot ce tine de hardware design:

- lista de piese
- scheme electrice (se pot lua si de pe Internet si din datasheet-uri,

e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)

- diagrame de semnal
- rezultatele simularii

Software Design

Descrierea codului aplicatiei (firmware):


- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librarii si surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi si structuri pe care planificati sa le implementati
- (etapa 3) surse si functii implementate

Rezultate Obtinute

Care au fost rezultatele obtinute in urma realizarii proiectului vostru.

Concluzii

Download

O arhiva (sau mai multe daca este cazul) cu fisierele obtinute in urma realizarii proiectului: surse, scheme, etc. Un fisier README, un ChangeLog, un script de compilare si copiere automata pe uC creeaza intotdeauna o impresie buna .

Fisierele se incarca pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul in care se incarca fisierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (daca este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

- Saptamana 1: Alegere idee proiect si componente
- Saptamana 2: Comanda si receptie componente
- Saptamana 3: Testare individuala a fiecarei componente pe breadboard
- Saptamana 4: Integrare componente si scriere firmware
- Saptamana 5: Constructie roata de pastile din carton, testare finala

Puteti avea si o sectiune de jurnal in care sa poata urmari asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Lista cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** si **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/delia.voicu>



Last update: **2026/05/16 20:32**