

# Rubik Cube Solver

## Introducere

Proiectul constă într-un robot capabil să recunoască culorile fețelor unui Cub Rubik și să le reasambleze în ordinea corectă. Acesta funcționează ca o punte între lumea digitală și cea fizică, primind instrucțiuni și transformându-le în mișcări reale pentru a rezolva cubul.

Scopul principal este realizarea unui sistem interactiv care să poată demonstra vizual etapele de rezolvare ale cubului. Nu este doar o mașinărie care face totul singură, ci un dispozitiv care permite utilizatorului să urmărească și să controleze întreg procesul, pas cu pas.

Totul a plecat de la dorința de a vedea cum concepte precum inteligența artificială pot fi aplicate într-un obiect tangibil. Am pornit de la ideea unui robot care să poată comunica cu noi: el ne spune ce vede pe cub, iar noi îi putem spune ce mișcări să facă, evitând astfel o funcționare rigidă și complet automată.

Pentru mine, utilitatea este practică: învăț să stăpânesc interacțiunea dintre senzori și partea mecanică într-un mod fluid. Pentru ceilalți, robotul are un rol strict didactic. Este mult mai simplu să înveți să rezolvi cubul urmărind o demonstrație fizică, executată în timp real, decât încercând să descifrezi scheme abstracte dintr-o carte sau de pe internet.

## Descriere generală



PC (Interfața de control): Reprezintă nivelul superior de decizie. Acesta rulează scripturile de rezolvare, calculează soluția optimă pe baza datelor primite de la senzori și transmite șirul de mișcări către microcontroler.

ATmega328P (Microcontrolerul): Este unitatea centrală care realizează managementul datelor. Acesta „traduce” comenzile de la PC în semnale electrice pentru motoare și centralizează informațiile citite de la sistemul de senzori.

Senzori C (Senzorii de culoare): Șase unități de citire dispuse astfel încât să identifice culorile fiecărei fețe a cubului. Aceștia transformă lumina reflectată de fețele cubului în date digitale.

Modulul I2C: Acționează ca un multiplicator de porturi. Deoarece senzorii sunt numeroși, acest modul permite conectarea tuturor la microcontroler folosind doar doi pini de date, eficientizând cablajul.

CNC Shield: Plăci de extensie care facilitează conectarea și controlul driverelor. Acestea organizează pinii de control pentru direcție și pași, permițând gestionarea celor 6 motoare necesare proiectului.

Drivere: Module de putere care primesc semnale de joasă tensiune de la microcontroler și livrează

curentul necesar motoarelor pentru a efectua mișcarea fizică.

Motoare (Stepper motors): Actuatori de precizie care rotesc fețele cubului. Am ales motoare pas cu pas pentru a avea un control exact asupra unghiului de rotație (90 sau 180 de grade).

Sursa de tensiune: Unitatea care furnizează energia brută necesară întregului ansamblu, asigurând independența electrică a robotului.

Distribuitor de putere: Un nod central care împarte energia de la sursă către componentele de forță (motoare) și cele de control.

Regulator de tensiune: Coboară și stabilizează tensiunea de la sursă la nivelul necesar pentru componentele logice sensibile (microcontroler și senzori), prevenind arderea acestora.

Cooler: Sistem de răcire activ care previne supraîncălzirea driverelor și a reguletoarelor în timpul funcționării prelungite a robotului.

## Hardware Design

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>)
- 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](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2026/atoadar/razvan.lazar2108> 

Last update: **2026/05/09 22:09**