

Bogdan-Alexandru ANDREI (78677) - Rubik's Cube

Autorul nu poate fi contactat momentan. Va rugam sa reveniti mai tarziu. : **Login pentru adresa**

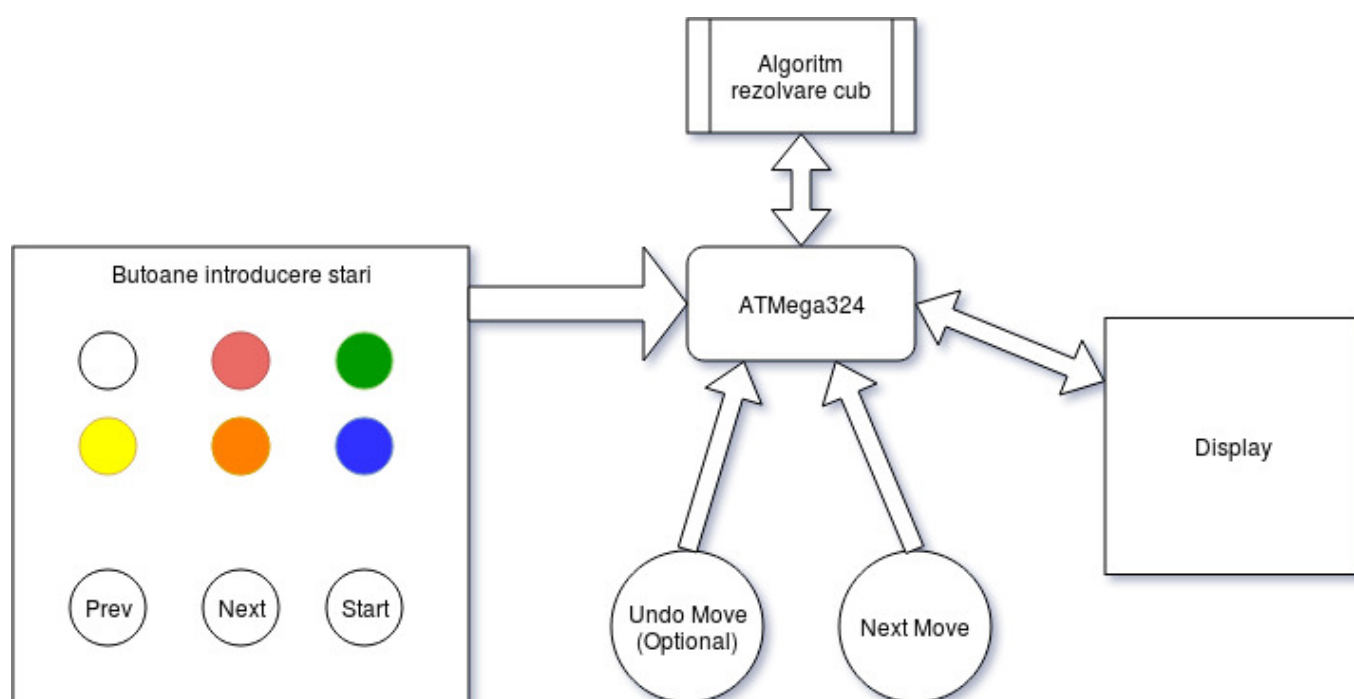
Introducere

Cubul Rubick este cel mai bine vandut puzzle din intreaga lume. In prezent 350 de milioane de cuburi au fost vandute, dar se estimeaza ca mai putin de 5.8% din populatia lumii poate rezolva celebrul joc.

Astfel ca proiectul meu va fi indreptat catre ceilalti 94.2%, avand ca scop indrumarea cuiva ce nu stie sa il rezolve inca cubul. Totul se va face prin afisarea pe un display a mutarilor ce vor fi efectuate de un om (pas cu pas) pentru a obtine solutia.

Starea initiala a cubului (*culoarea tuturor patratelor de pe fiecare fata*) va fi introdusa manual folosind butoane. Dupa ce se vor calcula miscarile necesare, trecerea de la o mutare la alta se va realiza printr-un buton.

Descriere generală



Initial ATmega324 va astepta introducerea celor 9×6 culori ale patratelor de pe fiecare fata a cubului, navigand prin matrice folosind butoanele din stanga diagramei. Vom pastra starea cubului intr-o matrice `cub[6][3][3]`, iar pentru fiecare pozitie vom selecta culoarea.

Apasarea butonului Start va porni algoritmul de rezolvare al cubului. In momentul in care avem solutia, fiecare mutare va fi afisata pe display, navigand prin mutari folosind butonul "Next Move". In functie de progresul proiectului si a resurselor hardware, voi incerca sa implementez si un buton de "Undo Move" pentru a remedia posibilele erori ale utilizatorului in momentul rezolvarii cubului.

Hardware Design

Componente	Cantitate
ATmega324	1
Componente de baza	1
Butoane	12
Display LCD (probabil 128×160)	1
Placa de test	1

Schema Electrica



Software Design

Codul a fost scris in Sublime, avand ca tutoriale exercitiile din laborator. Pentru a urca codul pe placuta am folosit `bootloadHID` oferit.

O parte din logica programului (reprezentarea mutarilor + algoritmul folosit) este inspirata de [Cube-Solver](#) si [rubiks-cube](#), modificand codul pentru a avea o varianta C ce poate fi folosita pe ATMEGA324 + adaptarea la cerintele proiectului meu.

Starea unui Cub

Culorile cubului sunt tinute codificat :

- 0 = YELLOW
- 1 = ORANGE
- 2 = BLUE
- 3 = RED
- 4 = GREEN
- 5 = WHITE

Starea initiala a cubului este citita si salvata intr-o matrice $3 \times 3 \times 6$, el fiind privit in **pozitie standard** pe tot parcursul rezolvarii.

Pozitie standard : Albastru in fata, Alb sus, Portocaliu in dreapta



Mutarile

Mutarile folosite sunt codificate standard:

- Right **R** si inversa ei **Ri**
- Left **L** si inversa ei **Li**
- Back **B** si inversa ei **Bi**
- Front **F** si inversa ei **Fi**
- Down **D** si inversa ei **Di**
- Up **U** si inversa ei **Ui**
- O mutare facuta de 2 ori consecutiv se va nota R2, L2, etc.



Rezolvare

Programul va aplica algoritmul clasic de rezolvare

1. Formarea crucii
2. Rezolvarea fetei de sus
3. Rezolvarea pieselor din mijloc
4. Rezolvarea fetei de jos

Se vor salva toti pasii necesari si se vor afisa pe ecran display-ul LCD, folosind codificarea mutarilor descrisa anterior.

Rezultate Obținute

In curs de dezvoltare, dar in final placuta te va indruma cu succes spre rezolvarea cubului Rubik.

Concluzii

În final, dezvoltarea acestui proiect m-a făcut să înțeleg cum funcționează dispozitivele din punct de vedere hardware (o să am mai mult respect pentru un mouse de acum), dar și cât de complicat este să faci un cod având resursele limitate ale unui microcontroller.

Download

Nefinalizat încă

Arhiva cod : [rubikcode.zip](#)

Jurnal

Implementare algoritm + Debug

Scrierea codului și debug pentru a verifica dacă mutările duc la rezultatul așteptat după fiecare etapă a algoritmului.



După partea Software, piesele folosite pentru a realiza proiectul.



Bibliografie/Resurse

- Documentația în format [PDF](#)

Resurse Software

- Algoritm 1 <https://github.com/matt2uy/Cube-Solver>]]
- Algoritm 2 [rubiks-cube](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/astratulat/rubik-s-cube>



Last update: **2021/04/14 15:07**