Materie: Proiectare cu Microprocesoare

**Proiect Cub Rubic cu Arduino Mega 2560 și OpenGL(C++)**

Bischin Adrian

*Grupa*: 30238

*Profesor îndrumător*: Mureșan Mircea Paul

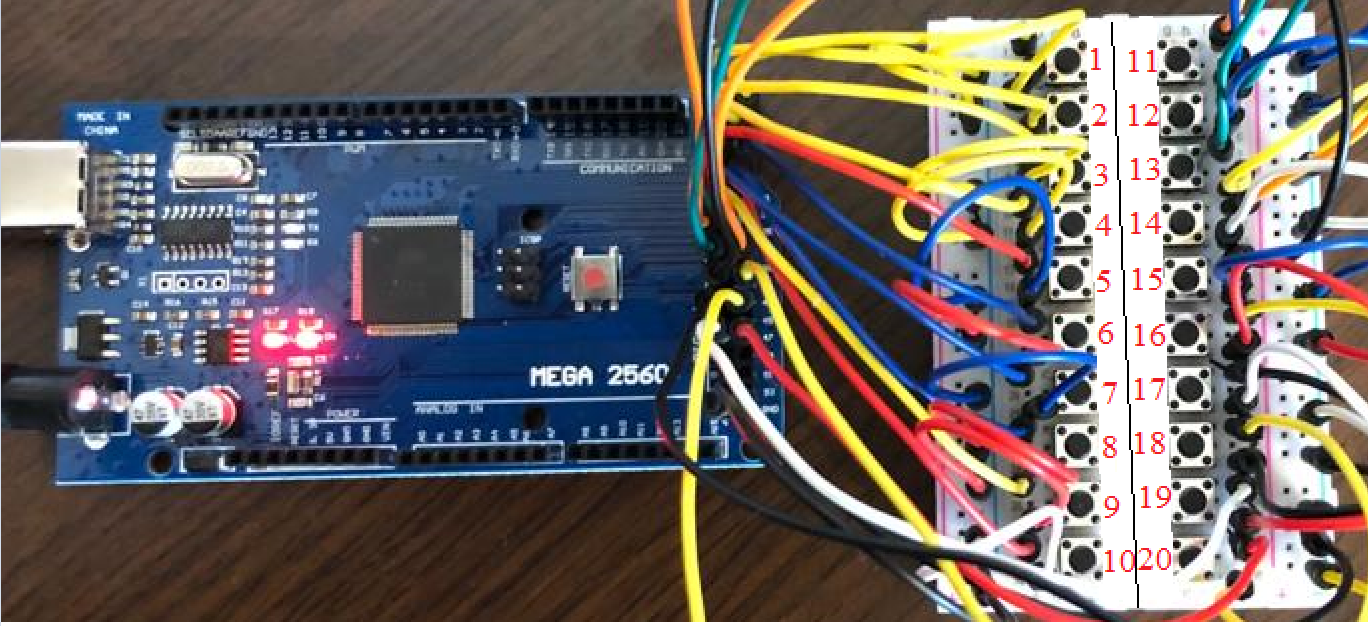
*Data*: 10.01.2021

In proiectul de fata s-a implementat un cub rubic virtual. Functionalitatile oferite sunt toate rotirile posibile ale cubului si afisarea acestuia pe monitorul unui calculator. Nu se doreste implementarea algoritmilor pentru rezolvarea automata a unui cub rubic, ci doar posibilitatea de a rezolva (pentru exercitiu sau amuzament) renumitul puzzle de catre o persoana care nu dispune la un moment dat de un cub rubic fizic. Numarul total de rotiri posibile este 18: pentru fiecare plan din spatiu sunt trei felii ale cubului care pot fi rotite, si fiecare felie poate fi rotita in 2 sensuri; => 3 x 3 x 2 = 18.

Toata logica proiectului este implementata pe placuta de dezvoltare (in arduino), iar cu ajutorul OpenGL si al limbajului de programare C++ s-a realizat doar componenta de afisare a cubului. Conexiunea si transmisia seriala intre Arduino si OpenGL s-a realizat cu ajutorul unei biblioteci seriale preluate de la adresa <https://github.com/ZainUlMustafa/Connect-And-Use-Arduino-via-Cpp-Software-Made-In-Any-IDE>.

Dupa fiecare apasare pe buton, Arduino proceseaza actiunea si trimite serial matricile cu culorile fetelor cubului spre aplicatia OpenGL; aplicatia proceseaza datele primate si afiseaza cubul corect.

Exista 20 butoane de control pentru Arduino: 18 pentru rotiri, unul pentru a aduce cubul in starea de rezolvat, iar ultimul pentru a adduce cubul intr-o stare amestecata dar prestabilita. De pe tastele w,a,s,d ale calculatorului folosit, se poate roti cubul pentru a ii observa toate fetele.



Imaginea 1.

**Corespondenta dintre *actiune - pin arduino – numar butoan*, conform montajului din imaginea de mai sus:**

|  |  |  |
| --- | --- | --- |
| **Actiune** | **Pin Arduino** | **Numar Buton** |
| plan z=0, trig. | 22 | 1 |
| plan z=0, invers trig. | 23 | 2 |
| plan z=1, trig. | 24 | 3 |
| plan z=1, invers trig. | 25 | 4 |
| plan z=2, trig. | 26 | 5 |
| plan z=2, invers trig. | 27 | 6 |
| plan x=0, trig. | 30 | 7 |
| plan x=0, invers trig. | 31 | 8 |
| plan x=1, trig. | 32 | 9 |
| plan x=1, invers trig. | 33 | 10 |
| plan x=2, trig. | 34 | 11 |
| plan x=2, invers trig. | 35 | 12 |
| plan y=0, trig. | 38 | 13 |
| plan y=0, invers trig. | 39 | 14 |
| plan y=1, trig. | 40 | 15 |
| plan y=1, invers trig. | 41 | 16 |
| plan y=2, trig. | 42 | 17 |
| plan y=2, invers trig. | 43 | 18 |
| rezolva cub | 46 | 19 |
| pozitie amestecata | 47 | 20 |

Matricile corespunzatoare fetelor cubului:

Sus:

B B B

B B B

B B B

Stanga: Fata: Dreapta:

O O O R R R Y Y Y

O O O R R R Y Y Y

O O O R R R Y Y Y

Jos:

G G G

G G G

G G G

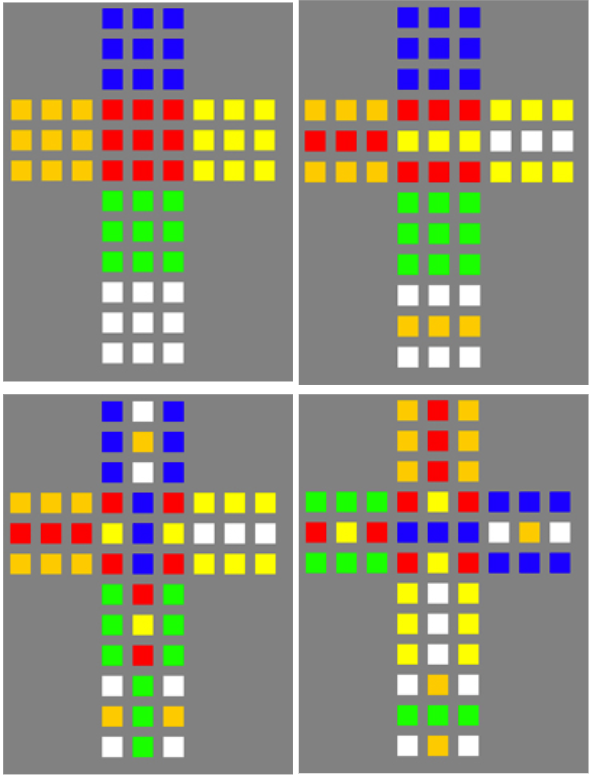
Spate:

W W W

W W W

W W W

Ilustratia 1 .



Ilustratia 2. (Sursa: https://www.algosome.com/articles/rubiks-cube-computer-visualize.html)

**!! Maparea matricilor pe la fetele cubului:**

Consideram ca privim cubul rubic drept spre una dintre fete. Sistemul de coordinate ales este raportat la ochiul privitorului si la cub (considerand ca acesta nu se roteste cu totul, ci doar culorile de pe suprafata lui pot fi interschimbate), dupa cum urmeaza:

* axa x: dinspre stanga inspre dreapta
* axa y: dinspre jos inspre sus
* axa z: dinspre ochiul privitorului inspre cub

Consideram ca matricile sunt lipite intre ele in formatul reprezentat in schema 1; matricea din mijloc se suprapune peste fata cubului spre care privim, celelalte matrice infasurandu-se in jurul cubului precum o hartie de impachetat cadouri. Matricile corespund fetelor cubului, in modul urmator:

* matricea notata cu eticheta “Fata:” corespunde fetei cubului spre care privim si se mapeaza 1 la 1.
* celelalte matrici corespund fetelor cubului din partea Stanga, Dreapta, de Sus, de Jos, respectiv din Spate, dupa corespondenta notata din schema de mai sus. Atentie insa la faptul ca orientarea matricilor mapate pe fetele cubului difera in functie de planul in care fata cubului se afla.

Corespondenta dintre matrici si fetele cubului:

* **Fata de Sus:** paralela cu planul xOz din spatiu; liniile matricei (Ox) pe directia Ox din spatiu; coloanele matricei (Oy) pe directia Oz din spatiu.
* **Fata din Stanga:** paralela cu planul (-z)Oy din spatiu; liniile matricei (Ox) pe directia -Oz din spatiu; coloanele matricei (Oy) pe directia Oy din spatiu.
* **Fata dinspre privitor (din fata):** paralela cu planul xOy din spatiu; liniile matricei (Ox) pe directia Ox din spatiu; coloanele matricei (Oy) pe directia Oy din spatiu.
* **Fata din Dreapta:** paralela cu planul zOy din spatiu; liniile matricei (Ox) pe directia Oz din spatiu; coloanele matricei (Oy) pe directia Oy din spatiu.
* **Fata de Jos:** paralela cu planul xO-z din spatiu; liniile matricei (Ox) pe directia Ox din spatiu; coloanele matricei (Oy) pe directia -Oz din spatiu.
* **Fata din Spate:** paralela cu planul xO-y din spatiu; liniile matricei (Ox) pe directia Ox din spatiu; coloanele matricei (Oy) pe directia -Oy din spatiu.

**Operatiile de rotatie si alterarea matricilor, prezentate in PSEUDOCOD (Consideram originea sistemului de coordinate fixat in coltul cubului din stanga jos care se vede dinspre privitor):**

1. **Rotire in planul xOy (rotire in sens trigonometric fata de axa z):**
   1. **Rotirea feliei din planul z = 0:**
      1. **Rotire cu pi/2:**

Aux1[] <= Stanga[coloana3];

Stanga[coloana3] <= Sus[linia3 inversata];

Sus[linia3] <= Dreapta[coloana1];

Dreapta[coloana1] <= Jos[linia1 inversata];

Jos[linia1] <= Aux1[];

Aux1[] <= Fata[coloana1];

Aux2[] <= Fata[coloana2];

Aux3[] <= Fata[coloana3];

Fata[linia1] <= Aux3[];

Fata[linia2] <= Aux2[];

Fata[linia3] <= Aux1[];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Stanga[coloana3];

Stanga[coloana3] <= Jos[linia1];

Jos[linia1] <= Dreapta[coloana1 inversata];

Dreapta[coloana1] <= Sus[linia3];

Sus[linia3] <= Aux1[inversata];

Aux1[] <= Fata[coloana1];

Aux2[] <= Fata[coloana2];

Aux3[] <= Fata[coloana3];

Fata[linia1] <= Aux1[inversata];

Fata[linia2] <= Aux2[inversata];

Fata[linia3] <= Aux3[inversata];

* 1. **Rotirea feliei din planul z = 1:**
     1. **Rotire cu pi/2:**

Aux1[] <= Stanga[coloana2];

Stanga[coloana2] <= Sus[linia2 inversata];

Sus[linia2] <= Dreapta[coloana2];

Dreapta[coloana2] <= Jos[linia2 inversata];

Jos[linia2] <= Aux1[];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Stanga[coloana2];

Stanga[coloana2] <= Jos[linia2];

Jos[linia2] <= Dreapta[coloana2 inversata];

Dreapta[coloana2] <= Sus[linia2];

Sus[linia2] <= Aux1[inversat];

* 1. **Rotirea feliei din planul z = 2:**
     1. **Rotire cu pi/2:**

Aux1[] <= Stanga[coloana1];

Stanga[coloana1] <= Sus[linia1 inversata];

Sus[linia1] <= Dreapta[coloana3];

Dreapta[coloana3] <= Jos[linia3 inversata];

Jos[linia3] <= Aux1[];

Aux1[] <= Spate[coloana1];

Aux2[] <= Spate[coloana2];

Aux3[] <= Spate[coloana3];

Spate[linia3] <= Aux3[inversata];

Spate[linia2] <= Aux2[inversata];

Spate[linia1] <= Aux1[inversata];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Stanga[coloana1];

Stanga[coloana1] <= Jos[linia3];

Jos[linia3] <= Dreapta[coloana3 inversata];

Dreapta[coloana3] <= Sus[linia1];

Sus[linia1] <= Aux1[inversata];

Aux1[] <= Spate[coloana1];

Aux2[] <= Spate[coloana2];

Aux3[] <= Spate[coloana3];

Spate[linia3] <= Aux1[];

Spate[linia2] <= Aux2[];

Spate[linia1] <= Aux3[];

1. **Rotire in planul (-z)Oy (rotire in sens trigonometric fata de axa x):**
   1. **Rotirea feliei din planul x = 0:**
      1. **Rotire cu pi/2:**

Aux1[] <= Sus[coloana1];

Sus[coloana1] <= Fata[coloana1];

Fata[coloana1] <= Jos[coloana1];

Jos[coloana1] <= Spate[coloana1];

Spate[coloana1] <= Aux1[];

Aux1[] <= Stanga[linia1];

Aux2[] <= Stanga[linia2];

Aux3[] <= Stanga[linia3];

Stanga[coloana1] <= Aux1[inversata];

Stanga[coloana2] <= Aux2[inversata];

Stanga[coloana3] <= Aux3[inversata];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Sus[coloana1];

Sus[coloana1] <= Spate[coloana1];

Spate[coloana1] <= Jos[coloana1];

Jos[coloana1] <= Fata[coloana1];

Fata[coloana1] <= Aux1[];

Aux1[] <= Stanga[linia1];

Aux2[] <= Stanga[linia2];

Aux3[] <= Stanga[linia3];

Stanga[coloana1] <= Aux3[];

Stanga[coloana2] <= Aux2[];

Stanga[coloana3] <= Aux1[];

* 1. **Rotirea feliei din planul x = 1:**
     1. **Rotire cu pi/2:**

Aux1[] <= Sus[coloana2];

Sus[coloana2] <= Fata[coloana2];

Fata[coloana2] <= Jos[coloana2];

Jos[coloana2] <= Spate[coloana2];

Spate[coloana2] <= Aux1[];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Sus[coloana2];

Sus[coloana2] <= Spate[coloana2];

Spate[coloana2] <= Jos[coloana2];

Jos[coloana2] <= Fata[coloana2];

Fata[coloana2] <= Aux1[];

* 1. **Rotirea feliei din planul x = 2 (rotire in sens trigonometric fata de axa x):**
     1. **Rotire cu pi/2:**

Aux1[] <= Sus[coloana3];

Sus[coloana3] <= Fata[coloana3];

Fata[coloana3] <= Jos[coloana3];

Jos[coloana3] <= Spate[coloana3];

Spate[coloana3] <= Aux1[];

Aux1[] <= Dreapta[linia1];

Aux2[] <= Dreapta[linia2];

Aux3[] <= Dreapta[linia 3];

Dreapta[coloana1] <= Aux3[];

Dreapta[coloana2] <= Aux2[];

Dreapta[coloana3] <= Aux1[];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Sus[coloana3];

Sus[coloana3] <= Spate[coloana3];

Spate[coloana3] <= Jos[coloana3];

Jos[coloana3] <= Fata[coloana3];

Fata[coloana3] <= Aux1[];

Aux1[] <= Dreapta[linia1];

Aux2[] <= Dreapta[linia2];

Aux3[] <= Dreapta[linia 3];

Dreapta[coloana1] <= Aux1[inversata];

Dreapta[coloana2] <= Aux2[inversata];

Dreapta[coloana3] <= Aux3[inversata];

1. **Rotire in planul xO-z (rotire in sens trigonometric fata de axa y):**
   1. **Rotirea feliei din planul y = 0:**
      1. **Rotire cu pi/2:**

Aux1[] <= Fata[linia3];

Fata[linia3] <= Dreapta[linia3];

Dreapta[linia3] <= Spate[linia1 inversata];

Spate[linia1] <= Stanga[linia3 inversata];

Stanga[linia3] <= Aux1[];

Aux1[] <= Jos[coloana1];

Aux2[] <= Jos[coloana2];

Aux3[] <= Jos[coloana3];

Jos[linia1] <= Aux3[];

Jos[linia2] <= Aux2[];

Jos[linia3] <= Aux1[];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Fata[linia3];

Fata[linia3] <= Stanga[linia3];

Stanga[linia3] <= Spate[linia1 inversata];

Spate[linia1] <= Dreapta[linia3 inversata];

Dreapta[linia3] <= Aux1[];

Aux1[] <= Jos[coloana1];

Aux2[] <= Jos[coloana2];

Aux3[] <= Jos[coloana3];

Jos[linia1] <= Aux1[inversata];

Jos[linia2] <= Aux2[inversata];

Jos[linia3] <= Aux3[inversata];

* 1. **Rotirea feliei din planul y = 1:**
     1. **Rotire cu pi/2:**

Aux1[] <= Fata[linia2];

Fata[linia2] <= Dreapta[linia2];

Dreapta[linia2] <= Spate[linia2 inversata];

Spate[linia2] <= Stanga[linia2 inversata];

Stanga[linia2] <= Aux1[];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Fata[linia2];

Fata[linia2] <= Stanga[linia2];

Stanga[linia2] <= Spate[linia2 inversata];

Spate[linia2] <= Dreapta[linia2 inversata];

Dreapta[linia2] <= Aux1[];

* 1. **Rotirea feliei din planul y = 2:**
     1. **Rotire cu pi/2:**

Aux1[] <= Fata[linia1];

Fata[linia1] <= Dreapta[linia1];

Dreapta[linia1] <= Spate[linia3 inversata];

Spate[linia3] <= Stanga[linia1 inversata];

Stanga[linia1] <= Aux1[];

Aux1[] <= Sus[coloana1];

Aux2[] <= Sus[coloana2];

Aux3[] <= Sus[coloana3];

Sus[linia1] <= Aux1[inversata];

Sus[linia2] <= Aux2[inversata];

Sus[linia3] <= Aux3[inversata];

* + 1. **Rotire cu -pi/2:**

Aux1[] <= Fata[linia1];

Fata[linia1] <= Stanga[linia1];

Stanga[linia1] <= Spate[linia3 inversata];

Spate[linia3] <= Dreapta[linia1 inversata];

Dreapta[linia1] <= Aux1[];

Aux1[] <= Sus[coloana1];

Aux2[] <= Sus[coloana2];

Aux3[] <= Sus[coloana3];

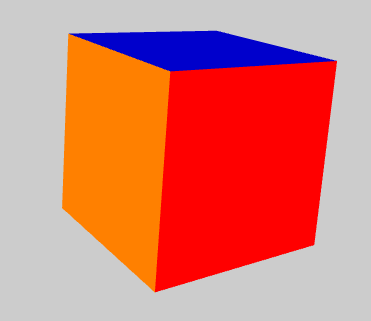
Sus[linia1] <= Aux3[];

Sus[linia2] <= Aux2[];

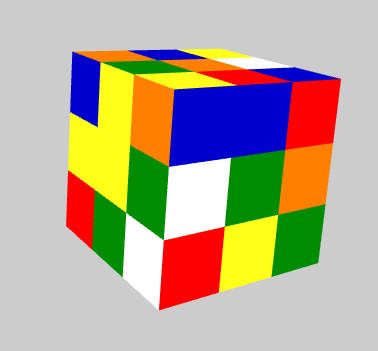
Sus[linia3] <= Aux1[];

**Afisarea implementata in OpenGL:**

Pentru modelarea cubului am definit varfurile necesare (cate 4 pentru fiecare celula) impreuna cu atributele de culoare pentru fiecare varf. Deoarece nu am folosit texturi, ci doar attribute de culoare pentru varfuri, calulele de pe fetele cubului nu sunt delimitate, asadar vizualizarea desi este posibila, este incomoda. Nu sunt implementate si rotirile feliilor cubului, ci doar schimbarea brusca a culorilor de pe fetele cubului. In schimb, de pe tastele w,a,s,d se poate roti cubul pentru a ii vizualiza toate fetele.



Imaginea 2.



Imaginea 3

**Posibilitati ulterioare de dezvoltare:**

* Aplicarea de texture in loc de lucori solide pentru a fi delimitate celulele de pe fetele cubului
* Implementarea tranzitiei rotirilor feliilor cubului