



COLEGIUL NAȚIONAL "GRIGORE MOISIL"

B-dul Timișoara nr. 33, Sector 6, București

Tel: 021 413 26 96; 021 413 26 47. Fax: 021 440 10 06

Website: <http://www.moisil.ro>

ROMÂNIA  
MINISTERUL  
EDUCAȚIEI



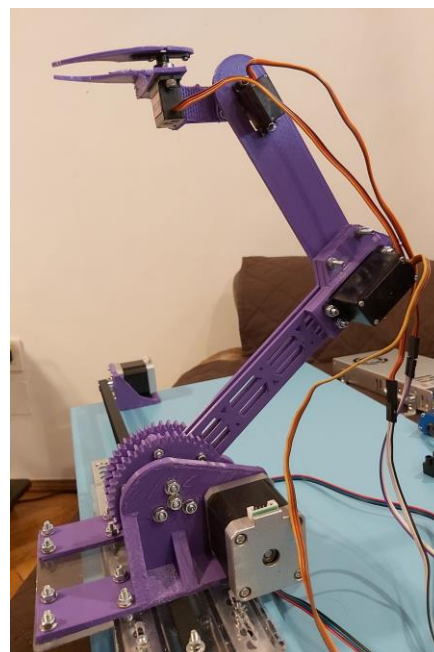
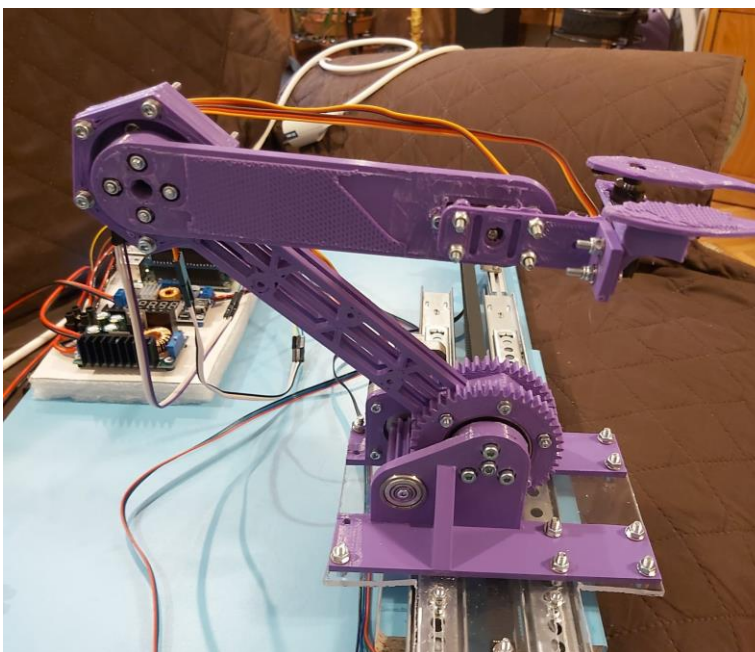
HSC D\_BOT

**Iatagan Andrei-Claudiu**

**Lăzăroiu Mihai**

### **Motto**

*În secolul XXI, robotul va prelua locul pe care îl ocupau sclavii în cadrul civilizațiilor antice. (Nikola Tesla)*



|  |  |
|--|--|
| <b>CUPRINS</b>                           | <b>Autori</b>                            |
| <b>CAPITOLUL I. UTILITATE PRACTICĂ</b>   | Iatagan Andrei Claudiu<br>Lăzăroiu Mihai |
| <b>CAPITOLUL II. MECANICA</b>            |  |
| Secțiunea II.1. Complexitate             | Iatagan Andrei Claudiu                   |
| Secțiunea II.2. Eficiența în construcție |  |
| <b>CAPITOLUL III. ELECTRONICA</b>        |  |
| Secțiunea III.1. Complexitate            | Lăzăroiu Mihai                           |
| <b>CAPITOLUL IV. SOFTWARE</b>            | Lăzăroiu Mihai<br>Iatagan Andrei Claudiu |
| <b>CAPITOLUL V. DESIGN INDUSTRIAL</b>    | Iatagan Andrei Claudiu<br>Lăzăroiu Mihai |



## CAPITOLUL I. UTILITATE PRACTICĂ

**HSCDBot** este un robot manipulator reprogramabil, destinat sectoarelor în care siguranța umană, în principal, este pusă în pericol, cum ar fi: unitățile industriale care utilizează substanțe periculoase, vopsitoriile, uzinele auto, robotul preluând sarcinile ce necesită precizie ridicată. Poate completa în mod natural și cu maximă precizie activități de sortare, manipulare și dozare realizate de către oameni. Brațele robotizate propuse sunt unelte flexibile, ușoare, gândite să răspundă facil unui număr mare de aplicații.

Robotul permite manipularea substanțelor ce reprezintă un pericol pentru oameni, animale, mediul înconjurător, precum: explozibili, substanțe toxice, acizi, baze, etc.; preia practic activitățile cu risc de explozie, incendiu sau contact cu aceste substanțe.

Volumul său de lucru este încadrat într– un sfert de cilindru. Acesta este generat de cele 4 grade de libertate destinate mișcării.

Robotul este proiectat astfel încât să poată fi ușor dimensionat/reconfigurat și pentru alte aplicații, marea majoritate a pieselor fiind proiectate de noi și printate cu ajutorul imprimantelor 3D personale.

Avantajul **HSCDBot** propus ține de versatilitate. Practic, cu o investiție minimă se pot îndeplini un număr mare de task-uri, pentru că acesta poate fi configurat să îndeplinească o varietate mare de sarcini ale oamenilor, putând face echipă bună cu aceeașia.

Alte avantaje ale utilizării **HSCDBot**:

- Creșterea calității producției (datorată motoarelor, cu o repetabilitate mare, controlate cu precizie),
- Creșterea siguranței în exploatare (poate manipula substanțe periculoase în locul oamenilor și nu are nevoie de confort în spațiul de lucru și determină reducerea numărului de accidente de muncă),
- Creșterea eficienței procesului de producție (poate fi utilizat continuu, mentenanța necesară fiind minimă),
- Reducerea costurilor de producție (elimină practic pierderile de material, cu costuri de operare minime).



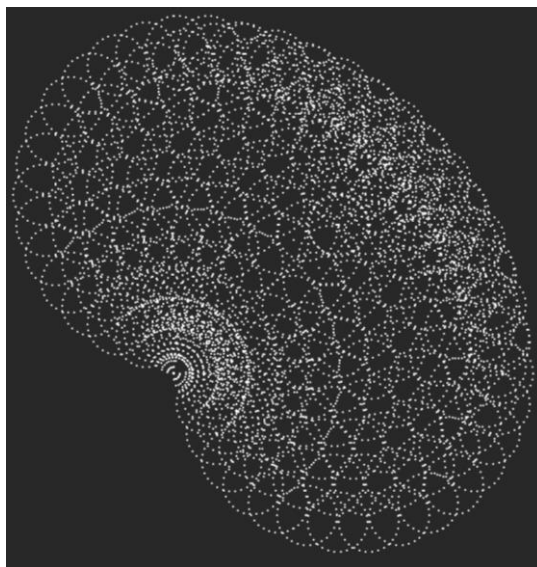
## CAPITOLUL II. MECANICA

### Secțiunea II.1. Complexitate

Pentru a determina numărul necesar de motoare, am studiat bazele limbajului de programare Processing. Acesta a fost folosit pentru a determina numărul minim de cuplaje de rotație necesare, proiectând mai multe poziții posibile ale efectorului.

Pentru 3 grade de libertate, brațele fiind simulate cu lungimile de 180, 150 respectiv 30 mm se obține următoarea diagramă.

```
void draw()
{
    for(/*ang1*/; ang1 < angmax_1; ang1 += 2 * PI / ticks_per_turn)
    {
        float arg1X = x0 + len1 * cos (ang1);
        float arg1Y = y0 - len1 * sin (ang1);
        point(arg1X, arg1Y);
        for(/*ang2*/; ang2 < angmax_2; ang2 += 2 * PI / ticks_per_turn)
        {
            float arg2X = arg1X + len2 * cos (ang1 + ang2);
            float arg2Y = arg1Y - len2 * sin (ang1 + ang2);
            point(arg2X, arg2Y);
            for(/*ang3*/; ang3 < angmax_3; ang3 += 2 * PI / 10)
            {
                float arg3X = arg2X + len3 * cos (ang3);
                float arg3Y = arg2Y - len3 * sin (ang3);
                point (arg3X, arg3Y);
            }
            ang3 = angmin_3;
        }
        ang2 = angmin_2;
    }
}
```



Folosind mai multe astfel de diagrame, am decis că 3 grade de libertate sunt suficiente atât pentru libertate completă de mișcare cât și pentru răsturnarea eprubetelor.

Robotul folosește în total 5 motoare:

- Motor stepper NEMA 17 asigură mișcarea liniară, realizată cu ajutorul a două glisiere cu bile și curea sincronă de tip GT2. Această metodă de mișcare a fost inspirată de funcționarea unor imprimante 3D. Fiind folosită o curea deschisă, prin simpla schimbare a glisierelor limitele mișcării liniare pot fi modificate. HSCDBot, în configurația actuală se poate mișca liniar, realizând o mișcare de translație de până la 300mm.
- Motor stepper NEMA 17 asigură mișcarea cuplajului R1 („umărul robotului”). Acesta folosește o reducere prin roți dințate având raportul de 4,5 la 1. Roțile dințate au modul 1.5, care a fost ales astfel încât roata dințată mică să aibă 10 dinți, dar să și poată



fi printată 3D, cu precizie. Acestea au un backlash minimal de 0.25mm, care imprimă brațului precizie în mișcare. Tot pentru precizie, mișcarea de rotație se face pe 2 rulmenți 6806 2RS, montați prin frecare direct pe un tambur printat 3D.

- Servomotor SG5010 asigură mișcarea cuplajului R2 („*cotul robotului*”). Acesta a fost ales deoarece oferă cuplu suficient, dar și libertate de mișcare, având 180° de mișcare.
- Servomotor MG90 asigură mișcarea articulației R3 („*încheietura palmei robotului*”). Acesta a fost ales datorită greutății reduse, de sub 15 grame. Acest cuplaj servește poziționării corecte a elementului efector.
- Servomotor MG90 este folosit pentru efectorul ales, de asemenea, pentru greutatea sa redusă.

Toate aceste elemente, montate în serie imprimă brațului robotic libertate de mișcare. Acestea sunt așezate pe o placă de bază fixă, din PAL melaminat, pe care este montată inclusiv componenta electronică a robotului.

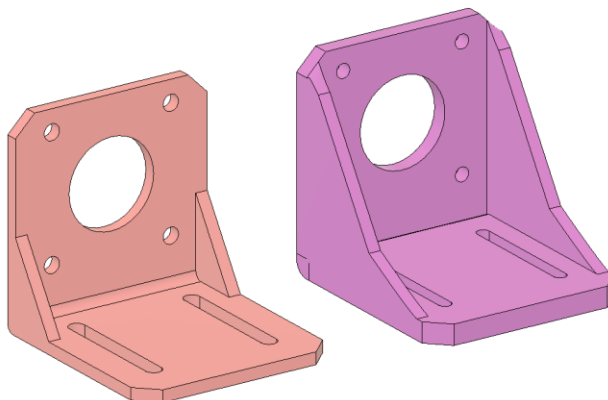
Soluția adoptată de noi pentru construcția robotului îl face multifunctional, având o gamă largă de aplicații.

### ***Secțiunea II.2. Eficiența în construcție***

Marea majoritate a pieselor **HSCDBot** au fost proiectate de către autori folosind Autodesk Fusion 360. Unele organe de asamblare necesare proiectării (șuruburi DIN912, șeibi DIN125, piulițe DIN 985) au fost preluate din librăria Mc-Master-Carr. De asemenea, câteva piese au fost luate de pe GrabCAD: rulmenți 6806 2RS, 608, fulie GT2.

Pentru suportul motorului care asigură mișcarea liniară ne-am inspirat de pe GrabCAD, modificând o piesă metalică, astfel încât să reziste la forte relative mari, chiar dacă este printată 3D (îngroșarea elementelor de suport, schimbarea diametrelor găurilor, adăugare unor *Rib*-uri pentru suport)





Stânga – model piesă.

Dreapta – Piesă dimensionată/adaptată pentru PLA

Printre aspectele eficienței în construcție se numără:

- Cuplajul R1 folosește rulmenți, frecarea între roțile dințate mari și tambur fiind aproape zero,
- Roțile dințate sunt bine aliniat, rezultat al montării unui rulment 608, care asigură paralelismul între axul motorului și axa de rotație a roților dințate mari,
- Glisierile sunt montate aproape perfect paralele, pentru a reduce rezistența la mișcare,
- Glisierile au fost curățate de lubrifiantul din fabrică și a fost aplicat ulei pe bază de silicon, care contribuie la reducerea rezistenței la mișcare,
- Brațele 2 și 3 sunt montate direct pe servomotoare, asigurând transfelul complet al energiei servomotoarelor,
- Fixarea robotului atât pe glisieră, cât și pe placa de bază asigură înlăturarea aproape completă a vibrațiilor nedorite.

Robotul este alimentat cu 12,5V DC, obținuți dintr-o sursă de 220V AC. Energia necesară ar putea fi oferită și de un sistem fotovoltaic, care utilizează pentru acumularea energiei o baterie Pb-AGM de 12V cu capacitate cuprinsă între 10 și 14 Ah

### CAPITOLUL III. ELECTRONICA

#### Secțiunea III.1. Complexitate

Partea electronică a robotului este alcătuită din două componente principale:



- O componentă de putere, care asigură alimentarea cu curent electric și controlul motoarelor de tip stepper și servo;
- O componentă de calcul (placă de dezvoltare Raspberry Pi), prin intermediul căreia este rulat programul software pentru comanda robotului și este asigurată interfața cu modulele de control ale motoarelor.

Componenta de putere, la rândul ei, este formată din:s

- Sursă de alimentare AC-DC, care permite permite alimentarea întregului ansamblu la 220V c.a. și furnizarea unei tensiuni de 12,5V, pentru un curent maxim de 20A;
- Sursă coborâtoare de tensiune DC-DC, de la 12,5V la 3,7V, care alimentează modulele de control pentru motoarele stepper, la un consum maxim de 9A;
- Sursă coborâtoare de tensiune DC-DC, de la 12,5V la 6V, care alimentează modulul de control pentru motoarele servo, la un consum maxim de 5A;
- Două module de control (*DC & Stepper Motor HAT Raspberry Pi*) pentru motoarele stepper NEMA 17, cu alimentează fiecare motor cu 3,7V, la un curent mediu 1,2A, asigurând un curent maxim de 3A, pentru situațiile în care oricare din motorele stepper trebuie să rămână în starea HOLD;
- Un modul de control (*Driver PWM / Servo HAT cu 16 canale Raspberry Pi*) pentru motoarele servo de tip SG5010, care alimentează fiecare motor cu 5.5V, la un curent mediu de 0,6A și un curent maxim de 1A.

Deși fiecare modul de control al motoarelor stepper poate acționa două motoare de acest tip, am ales utilizarea câte unui modul pentru fiecare motor, astfel încât să fie menținut un nivel de curent care să nu conducă la defecțiuni în cazul unor utilizări prelungite sau a unor eforturi mecanice ridicate.

Pe lângă funcția importantă de a furniza motoarelor curenți de valori mari, pe care componenta de calcul nu o poate asigura, modulele de control au și rolul de a interfața motoarele cu pinii de comandă de pe placa de dezvoltare.

Spre deosebire de motoarele de curent continuu (DC), care sunt controlate doar prin doi pini, motoarele folosite în acest proiect folosesc un număr mai mare de pini, după cum urmează:

- Fiecare motor stepper folosește 4 pini, pentru alimentarea celor două bobine de control a pasului, vitezei și sensului de rotație;
- Fiecare motor servo folosește 3 pini, doi pentru alimentarea în curent continuu și unul pentru controlul vitezei și sensului de rotație.

Prin urmare, pentru controlul tuturor motoarelor din acest proiect, ar fi fost necesare  $4 \times 2 + 3 = 11$  pini GPIO (*General Purpose Input/Output*) de pe placa Raspberry Pi. Pentru a simplifica



modul de interconectare, dar și pentru a reduce consumul total de curent pe care placa de dezvoltare trebuie să-l furnizeze, am ales să utilizăm interfațarea serială I2C (*Inter-Integrated Circuit*) dintre Raspberry Pi și fiecare modul de control. Practic, în locul celor 11 pini GPIO se vor utiliza doar doi pini (Pin #3 GPIO2- Data și Pin #5 GPIO3-Clock), care vor asigura transmisia serială către cele trei module de control.

Pentru ca unitatea de calcul să identifice corect prin programul software modulele de control, acestea trebuie să fie definite cu o adresă hexa-numerică unică. Astfel, fiecare modul a fost configurat astfel:

- Modulul de control pentru stepper-ul care asigură mișcarea liniară: adresa 0x60
- Modulul de control pentru stepper-ul care asigură mișcarea cuplajului R1: adresa 0x61
- Modulul de control al motoarelor servo: adresa 0x40

Toate cele trei module de control vor fi cascade și interfațate cu placa Raspberry Pi prin intermediul conecturii GPIO, rezultând un bloc compact.

Pe toată durata construcției și testelor, consumul de curent a fost monitorizat permanent, pentru a evita funcționarea necorespunzătoare sau defectarea componentelor în cazul unor situații neprevăzute, cum ar fi blocarea unui motor sau o sarcină mecanică excesivă.

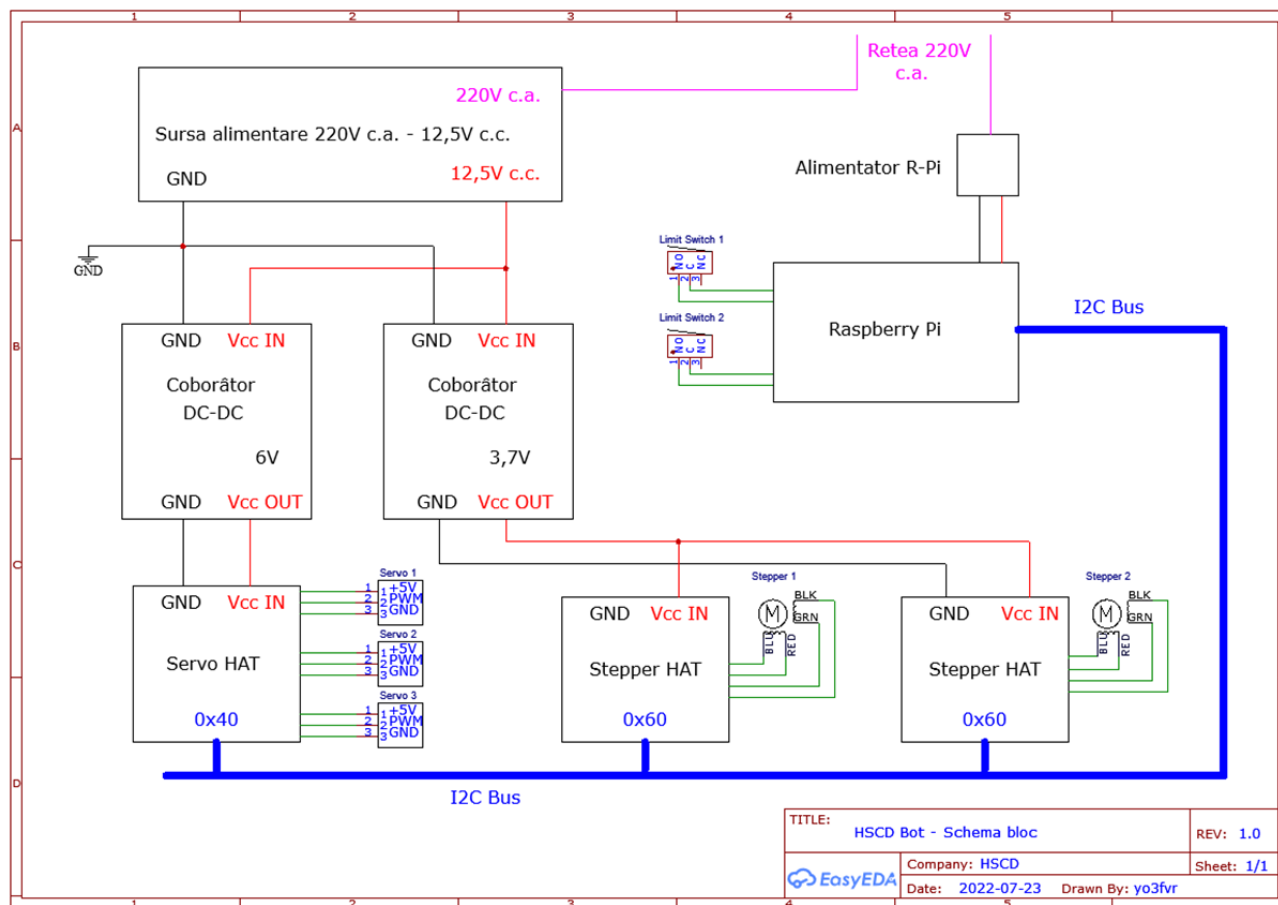
Componenta de calcul, reprezentată de o placă de dezvoltare Raspberry Pi 3B, cu 1GB memorie RAM, este cea care rulează codul software, scris în limbajul Python 3, pentru controlul motoarelor care pun în mișcare platforma și brațul robotului.

Prin intermediul aceluiași cod software sunt urmărite și eventualele depășiri ale deplasării robotului, pe axele pe care funcționează motoarele stepper (oX și oY), prin utilizarea a două limitatoare. Depășirea limitelor stabilite pentru aceste axe este semnalizată corespunzător printr-un buzzer, dublat de un indicator LED.

Pentru deplasările suportate de motoarele servo nu este necesară monitorizarea limitelor, deoarece aceste motoare au limitări integrate din construcție, care nu permit depășirea unei mișcări de rotație mai mare de 180 de grade.

Schema bloc este prezentată în Figura III.1





## CAPITOLUL IV. SOFTWARE

Programarea robotului a fost efectuată direct pe Raspberry Pi. Folosim Python 3, limbaj ales datorită librărilor compatibile cu raspberry pi dar și cu partea hardware folosită.

Pentru a schimba parametrii de funcționare ai robotului se pot modifica niște variabile globale, care oferă control asupra limitelor software, vitezelor de operare și pinilor prin care sunt montate motoare.



```
#desemnare HAT-uri
kit1 = Adafruit_MotorHAT(addr=0x60)
kit2 = Adafruit_MotorHAT(addr=0x61)
kit3 = ServoKit(channels=16)

#initializare limitatoare
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(20, GPIO.IN, pull_up_down=GPIO.PUD_UP)

myStepper1 = kit1.getStepper(200, 2) #initializare stepper liniar
myStepper2 = kit2.getStepper(200, 1) #initializare stepper umar

#homing stepper motor stepping rate
h_stepper1_steppingrate = 1
h_stepper2_steppingrate = 1

#normal stepper motor stepping rate
stepper1_steppingrate = 1
stepper2_steppingrate = 1

#limita stepper1 (counts)
linear_maxrange = 175

#limita stepper2 (counts)
first_joint_maxrange = 350

#valoare unghi home servo (deg)
servo1_hAng = 180
servo2_hAng = 90
servo3_hAng = 90

#variabila care retine unghiul curent al servo
servo1_cAng = servo1_hAng
servo2_cAng = servo2_hAng
servo3_cAng = servo3_hAng

myStepper1.setSpeed(10) #setare viteza implicita stepper #1
myStepper2.setSpeed(10) #setare viteza implicita stepper #2

#setare viteze servomotoare implicite
servo1_sStep = 10
servo2_sStep = 10
servo3_sStep = 10

#setare limite servomotoare
servo1_maxrange = 170
servo2_maxrange = 170
servo3_maxrange = 170
servo1_minrange = 10
servo2_minrange = 10
servo3_minrange = 10

#timestep servomotoare
time_step = 0.1
```

### Secvență cod – variabile globale

Înainte de orice operație, la inițializarea programului robotul are o secvență de „home”, în care, folosind limitatoare de tip limit switch, motoarele stepper se calibrează. Această secvență poate fi apelată oricând de utilizator, apăsând tasta H.

MainLoop – ul programului așteaptă practic input de la utilizator și, în funcție de tasta apăsată, mișcă motoarele. Acesta poate fi întrerupt rapid, apăsând tasta Q. De asemenea, pot fi dezactivate doar motoarele stepper. (Ultima acțiune este una de siguranță) Pentru motoarele servo, folosim limitare software, care să nu permită utilizatorului să miște gheara în poziții periculoase pentru el sau pentru electronică.

Pentru ca bucla principală să fie ușor de citit și de modificat, am pus unele acțiuni în funcții, definite după declararea de variabile globale.

Deoarece codul a fost scris de 2 persoane, am considerat necesară utilizarea unor variabile cu nume destul de lung, care să poată fi înțelese de ambii programatori. De asemenea, pentru a nu avea neclarități legat de ceva scris de celălalt, am folosit comentarii.



```
#MAINLOOP
try:
    while True:

        updateServoAngle()

        #inregistreaza input de la tastatura
        char = screen.getch()

        #inchidere program
        if char == ord('q'):
            break

        #realizeaza home total
        elif char == ord('h'):
            totalHome()
            servo1_cAng = servo1_hAng
            servo2_cAng = servo2_hAng
            servo3_cAng = servo3_hAng

#CONTROL SERVOMOTOARE#

elif char == ord('w'):
    if servo1_cAng <= servo1_maxrange:
        servo1_cAng = servo1_cAng + servo1_sStep
        time.sleep(time_step)
    if servo1_cAng == servo1_maxrange:
        print("Elbow uppermost limit reached!")

elif char == ord('s'):
    if servo1_cAng >= servo1_minrange:
        servo1_cAng = servo1_cAng - servo1_sStep
        time.sleep(time_step)
    if servo1_cAng == servo1_minrange:
        print("Elbow bottommost most limit reached!")

elif char == ord('e'):
    if servo2_cAng <= servo2_maxrange:
        servo2_cAng = servo2_cAng + servo2_sStep
        time.sleep(time_step)
    if servo2_cAng == servo2_maxrange:
        print("Wrist uppermost limit reached!")

elif char == ord('d'):
    if servo2_cAng >= servo2_minrange:
        servo2_cAng = servo2_cAng - servo2_sStep
        time.sleep(time_step)
    if servo2_cAng == servo2_minrange:
        print("Wrist bottommost limit reached!")
```

Secvență cod – începutul MainLoop



## CAPITOLUL V. DESIGN INDUSTRIAL

Piesele de plastic sunt concepute astfel încât să fie ușor printabile 3D. Piesele de uzură se pot schimba cu ușurință, chiar și de către personal nespecializat. Timpul total de printare (folosind un nozzle de 0.4 la piesele precise și unul de 0.8 la brațe este de cca 20 de ore).

De-a lungul procesului de proiectare am căutat, creat și executat componente ce pot fi ușor reproduse și din alte materiale, folosind mai multe procese (unele piese printate 3D trebuia inițial să fie tăiate din PC compact, așa că au fost proiectate pentru a fi tăiate cu ușurință din foi)

Pentru configurarea design-ului acestui robot au fost folosite cunoștințe de mecanică, geometrie, desen tehnic, electronică, informatică, proiectare. Astfel, putem considera că proiectarea robotului asigură utilizarea și reproducerea acestuia și la scară mare.

Figura 5.1 Robot HSCDBot

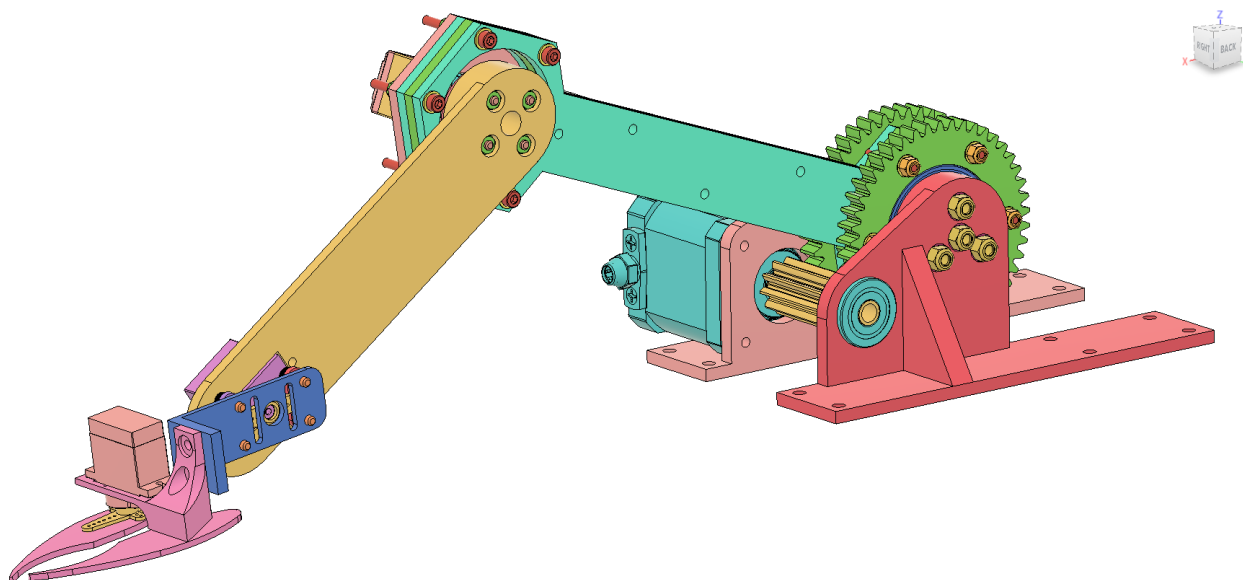


Figura 5.2 - Cupla R1

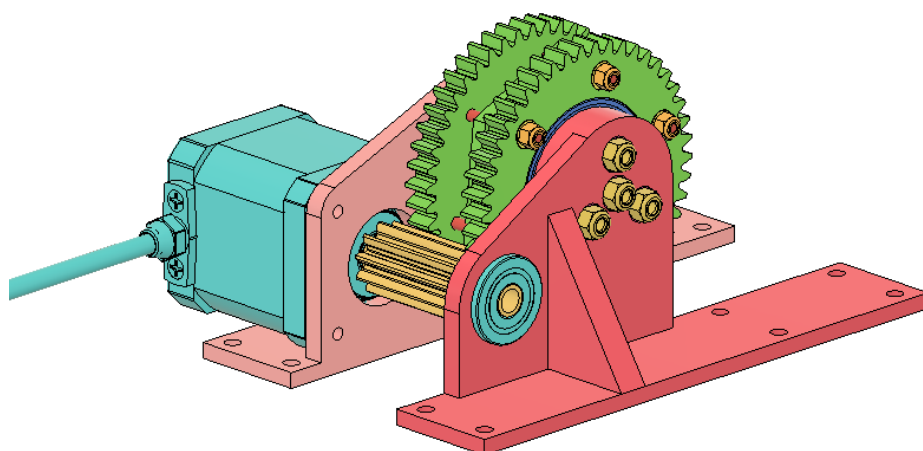


Figura 5.3 - Secțiune R1

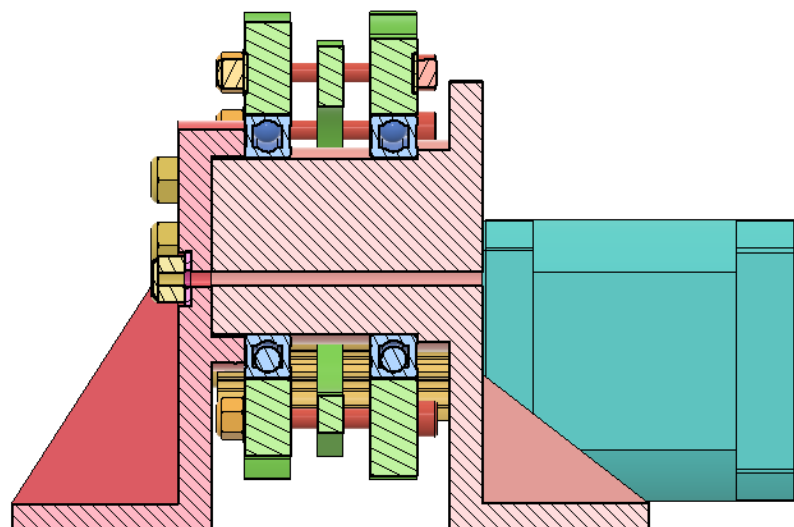




Figura 5.4 - Angrenaj Cupla R1

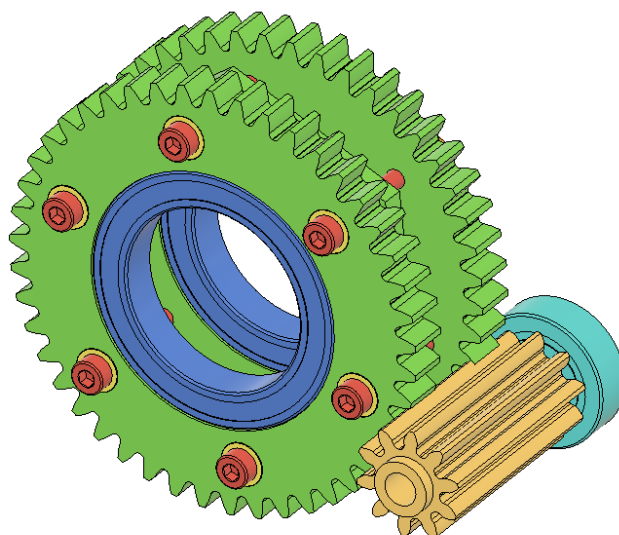


Figura 5.5 - Cupla R3 - spate

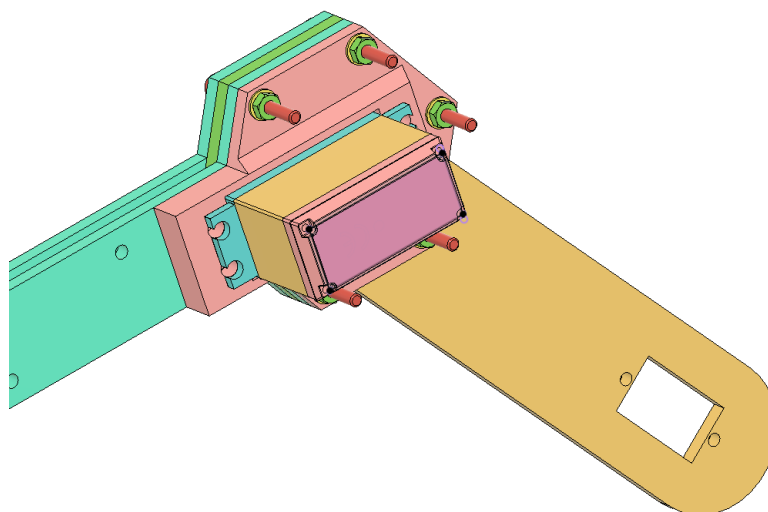






Figura 5.6 - Cupla R3 - față

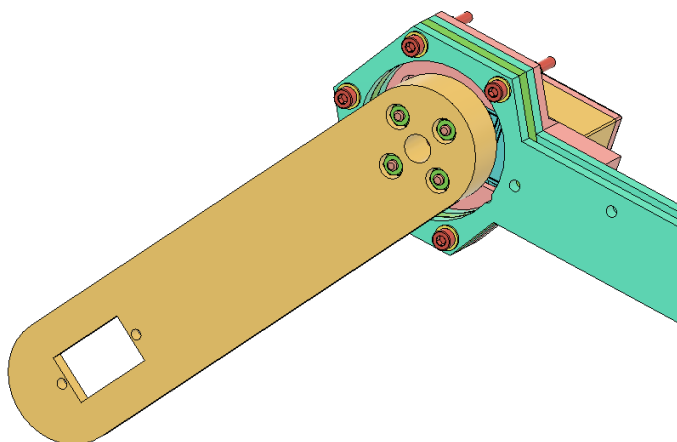


Figura 5.7- Cupla R4

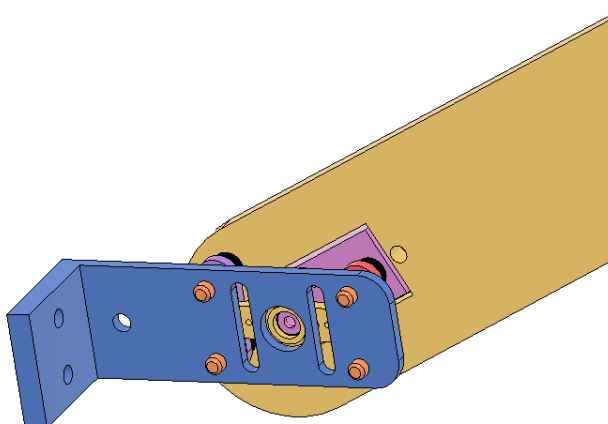




Figura 5.8 -*Montaj Cupla R4*

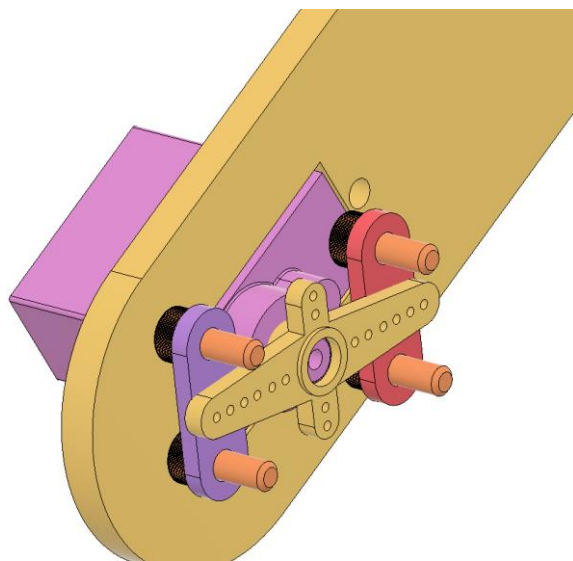
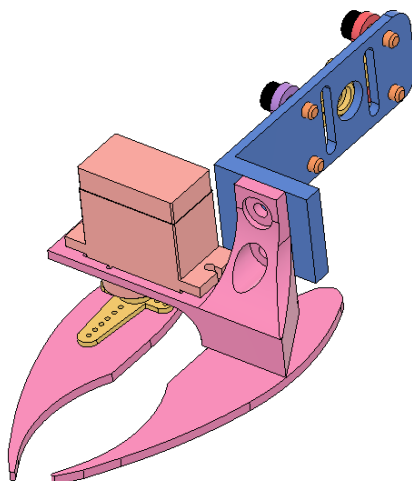


Figura 5.9- *Efactor 1 tip gheară*





COLEGIUL NAȚIONAL "GRIGORE MOISIL"

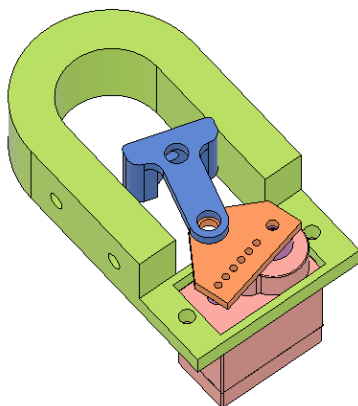
B-dul Timișoara nr. 33, Sector 6, București

Tel: 021 413 26 96; 021 413 26 47. Fax: 021 440 10 06

Website: <http://www.moisil.ro>

ROMÂNIA  
MINISTERUL  
EDUCAȚIEI

Figura 5.10- Efactor 2 pentru eprubete



Identitate vizuală robot :

