

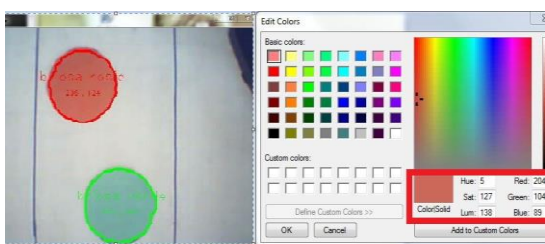


**Open CV** este un sistem pentru vizualizarea și scanarea imaginilor video. Este un soft conceput în anul 2003 de către marea companie INTEL și era conceput inițial în Microsoft Visual C++ , urmând ca în anul 2005 să îl licențieze cu licența BSD, adică să îl facă liber de acces la modificări și adaptări la nevoi.

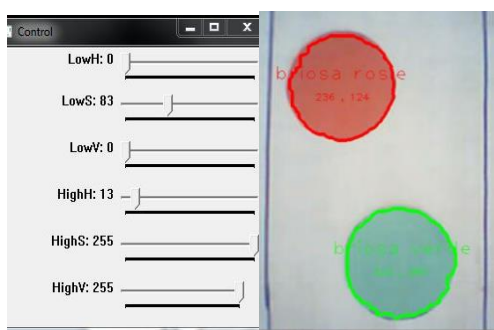
În cazul meu a fost adaptat aplicației de sortare și detecție a coordonatelor pe axele X,Y. Acest program permite ajustarea oricăror imagini sau imagini video și este capabil să facă scanarea după formă , culoare, poliție sau chiar și temperatura dacă avem aparatele necesare. Cu alte cuvinte este o bibliotecă Vision computerizată și deschisă la modificări în codul sursa.

În aplicație cu acest program am făcut scanarea după culorile roșu și verde. Le-am analizat formele pentru a le afla centrul lor pentru a le putea manevra cu effectorul robotului de tip electromagnet.

Ajustarea imaginilor s-a făcut cu ajutorul programului Paint unde am aflat nuanța, saturația și luminozitatea culorii în format binar.



Urmând a le trece manual în Open CV ca valori, pentru a le filtra de restul culorilor care apar în imaginea video.



După această calibrare softul poate filtra culorile pe care le-am selectat, în cazul nostru este culoare roșie și verde. Aceste 2 culori sunt culorile pe care el poate vedea și selecta în cazul nostru.

Un urma cărora le detectează poziționarea lor pe sistemul de axe X și Y în cadrul acero 2 marcaje puse pe conveior cu culoarea albastră.

Coordonatele sunt calibrate în program cu cele ale robotului și le putem vedea scrie chiar pe piesele detectate, mai întâi axa X urmând axa Y.



Aceste coordonate sunt trimise către calculator unde sunt procesate și convertite în coordonatele robotului care este comandat să ajungă în acele coordonate și să preia piesa pentru a le depune pe paleta în locașul corespunzător, conform programului de executat.

Programul este conceput așa încât piesele să fie depuse în diagonala paletelor astfel încât să fie roșu și roșu și verde cu verde.



Amplasarea pieselor pe paleta se face cu ajutorul rotației paletelor date de către motorul pas cu pas montat direct drive pe acest sistem perirobotic care permite amplasarea paselor pe diagonală, rotind paleta cu 90 sau respectiv 180 de grade pentru a depune obiectul în lăcașe diferite și important ne ocupate.

### 3.2 Componente utilizate

Controlul și comanda s-a realizat pe trei brațe ale robotului care sunt acționate paralel. Acestea sunt puse în mișcare de către trei servomotoare cu reductoare integrate având un cuplu de 17N/cm, la alimentarea de 5V.

Pentru comanda servomotorului s-a folosit plăcuța Arduino Delimvaire, iar alimentarea s-a

realizat de la sursa de tensiune SP-320-36 conectată la rețeaua electrică ( 220V ).

Plăcuța Arduino Delimvaire face legătura dintre programul de comandă care rulează pe calculatorul personal și driver-ul integrat in servomotor.

Tabelul 1. Specificații Arduino Delimvaire

Caracteristici	Valoare/De numire
Microcontroler	Atmega328
Tensiune de lucru	5V
Tensiun de intrare (recomandată)	7-12V
Tensiune de intrare (limită)	6-20V
Pini digitali	14(6PWM output)
Intensitate de ieșire	40mA
Intensitate de ieșire pe 3.3V	50mA
Clock Speed	16Mhz

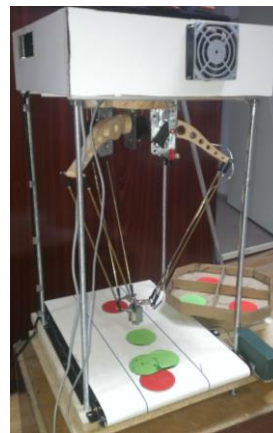
Placa Arduino Delimvaire se conectează la portul USB al calculatorului folosind un cablu de tip USB A-B, ce are rolul de a face schimb de informații între plăcuță și mediul de programare care este compilat in calculator si comanda trimisa către motoare , conveior și electromagnetului.

Ca EndEffector am folosit un electromagnet cărui ia fost modificat miezul cu un șurub, din cauza ca era un electromagnet oscilant cu revenire pe arc.



Acest electromagnet este alimentat la 24V si are o forță de prindere și menținere de 0.8N.

## 4 RELIZAREA PROIECTULUI



### 4.1 Realizarea modelul virtual

Modelul virtual al axelor comandate numeric de rotație s-a realizat în softul de modelare 3D, CATIA după modelul fizic studiat similar celui de la proiectul de diploma.

Fiecare reper a fost realizat după modelul fizic și adaptat la dimensiuni reduse.

Reperele au fost condiționate în ansamblu în așa fel încât să reproducă cât mai bine axele comandate numeric. Ansamblul final care reprezintă axele de rotație este prezentat mai jos.



După realizarea simulării este nevoie de inițializarea calcului pe care softul îl realizează pentru fiecare punct de traiectorie. Softul realizează calculul de poziție, viteză și accelerație pentru ca simularea sa fie cat mai precisă.

### 4.2 Realizarea Programului

Controlul și comanda simulării se realizează în mediul Programului C++, unde este si compilat sub SO Linux.

```

11 Object::Object(string name){
12
13     setType(name);
14
15     if(name=="briosa verde"){
16
17         setHSVmin(Scalar(70,50,50));
18         setHSVmax(Scalar(90,220,200));
19
20         //BGR value for Yellow:
21         setColor(Scalar(0,255,0));
22
23     }
24     if(name=="briosa rosie"){
25
26         setHSVmin(Scalar(0,90,60));
27         setHSVmax(Scalar(10,255,255));
28
29         //BGR value for Red:
30         setColor(Scalar(0,0,255));
31     }
32 }
33

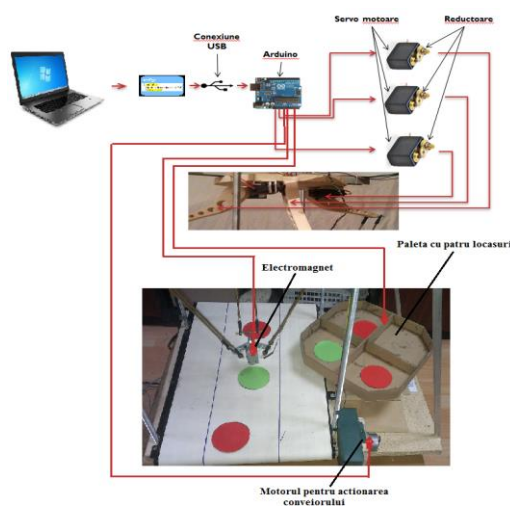
```

În figura de mai sus este prezentată interfața de programare și mai exact este bucățica de subprogram în care se face filtrarea culorii roșii și celei verzi. Filtrarea se face după cum am mai spus cu ajustarea gamei de nuanță, saturația și luminozitate a culorilor.

#### 4.5 Schema bloc de conectare

În figura de mai jos este prezentată schema bloc de conectare a întregului ansamblu. Schema bloc include partea de comandă și control a axelor.

PC-ul este „creierul” aplicației unde scriem codul sursă și în care deschidem portul de comunicare PuTTI cu ajutorul căreia facem schimbul de comenzi cu placa Arduino.



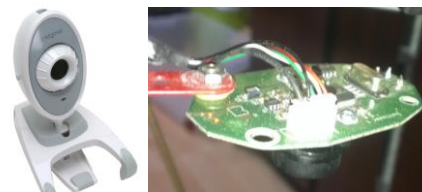
#### 5 SISTEMUL DE RECUNOASTERE VIDEO

Pentru realizarea acestui sistem, avem nevoie în primul rând de o cameră video pe care să o putem conecta la un calculator și pe care să o putem

focaliza cât mai bine, urmând să instalăm programul OpenCV.

OpenCV (Open Source Computer Vision Library) este un program de liberă utilizare în domeniul vederii artificiale dotat cu o bogată bibliotecă software și cu o foarte bună și flexibilă mașină de învățare. OpenCV a fost construit pentru a oferi o infrastructură comună pentru aplicații video de la calculator și pentru a accelera utilizarea de percepție mașină în produsele comerciale. Fiind un produs cu licență BSD, OpenCV face ușor pentru întreprinderile de a utiliza și modifica codul de bază.

În aplicație am folosit o cameră web obișnuită, model produs de către firma CREATIVE. Camera are o rezoluție 640 x 480p și o dimensiune relativ mică. În aplicație s-a folosit fără carcasa din plastic, pentru a permite o montare cât mai eficientă sub baza robotului și pentru a nu crea obstacole în spațiul de lucru al robotului.



#### 6 REALIZAREA FIZICĂ A ROBOTULUI.

În asociere cu proiectarea sistemului la scară reală ce constituie obiectul proiectului de diplomă, în cadrul temei de cercetare s-a decis realizarea practică și programarea unui robot cu acționare paralelă la scară redusă, integrat într-o aplicație de manipulare piese metalice de tip „capac” cu ajutorul unui sistem video inteligent de scanare și detecție a pieselor metalice pentru a putea și manipula de către robotul dotat cu effector electromagnetic.

Cu ajutorul programului OpenCV și cu o plăcuță cu micro-controler Arduino Delimvair și programarea elaborată în limbajul C++ pe calculatorul personal.

Aplicația prezentă poate fi prezentată ca drept exemplu didactic cât și ca o simulare în miniatură a unei celule de paletizare cu roboți de tip acționare paralelă cât și alți roboți în care poate fi integrat un sistem video de scanare și recunoaștere a obiectelor.

Pentru dezvoltarea ulterioară ne propunem integrarea acestui într-o celulă robotizată cu două manipulatori, benzi conveioare independente și echiparea lor cu sisteme de vedere artificială, independente, integrați într-o linie de producție automatizată a produselor alimentare.

## 7 MULTUMIRI

Mulțumim domnului Prof.dr.ing. Adrian NICOLESCU pentru tot sprijinul acordat pe durata acestei lucrări.

## 8 BIBLIOGRAFIE

[1]. Lupea I. (2008), *Programare Grafică*, Editura Risoprint, Cluj-Napoca.

[2]. Cinematia inversa disponibil la:

<http://www.trossenrobotics.com/>

[3]. Ghionea, I. (2013), *Inițiere în CATIA*, disponibil la: <http://www.catia.ro/articole.htm>

[4]. Programul de libera utilizare Open CV  
<http://opencv.org/>