a. Arduino

```
#include <Servo.h>                                   //librarie pentru servomotoare
Servo servo3;

int servoPin = 11;                                        //declarare variabile
int senzor = 0;
int val_senzor = 0;
int LED = 8;
int aux = 100;



void set_target(unsigned char servo, unsigned int pozitie)
{                                                         //merge la pozitia respectiva

  Serial.write(0x84);                                  //byte start
  Serial.write(servo);                                 //numar servo
  Serial.write(pozitie & 0x7F);                         //bytii de la 1 la 7
  Serial.write((pozitie >> 7) & 0x7F);                  //bytii de la 7-13

}
void set_speed(unsigned char servo, unsigned char viteza)
{                                                         //seteaza viteza de miscare

  Serial.write(0x87);                                  //byte start
  Serial.write(servo);                                 //numar servo
  Serial.write(viteza & 0x7F);                          //bytii de la 1 la 7
  Serial.write((viteza >> 7) & 0x7F);                   //bytii de la 7-13

}

void set_acceleration(unsigned char servo, unsigned char acceleratie)
{                                                         //seteaza acceleratia

  Serial.write(0x89);                                  //byte start
  Serial.write(servo);                                 //nr servo
  Serial.write(acceleratie & 0x7F);                     //data1
  Serial.write((acceleratie >> 7) & 0x7F);              //data2

}
```

```
void servoOff(unsigned char servo)
{                                                                    //opreste servo-ul

   Serial.write(0x84);                                          //byte start
   Serial.write(servo);                                    //nr servo
   Serial.print(0x00);                                     //data1
   Serial.write(0x0f);                                     //data2

}

void power()
{                                                                     //declansare LED
  for (int i=1; i<=3; i++)
  {
    digitalWrite(LED, HIGH);                      //aprinde LED-ul
    delay(50);
    digitalWrite(LED, LOW);                                 //stinge LED-ul
    delay(200);
  }

}

void miscare()
{

  set_speed(1,15);                                                //seteaza viteza
  set_target(1,9000);                                     //inchide clestele
  delay(2000);

  set_speed(0,40);                                               //seteaza viteza
  set_acceleration(0,30);                          //seteaza acceleratia
  set_target(0,2500);                                    //sus
  delay(1200);

  servo3.write(1000);                                          //dreapta
  delay(420);
  servo3.write(1500);                                    //opreste servoul
  delay(200);

   set_speed(0,10);                                              //seteaza viteza
  set_acceleration(0,0);                                     //seteaza acceleratia
  set_target(0,5600);                                         //jos
  delay(3500);

  set_speed(1,30);                                               //seteaza viteza
  set_target(1,6000);                                          //deschide clestele
  delay(1000);



  set_speed(0,40);                                    //seteaza viteza
  set_acceleration(0,30);                             //seteaza acceleratia
  set_target(0,2500);                                  //sus
  delay(1200);

  servo3.write(2000);                                 //stanga
  delay(400);
  servo3.write(1500);                           //opreste servoul
  delay(200);

  set_speed(0,10);                                    //seteaza viteza
  set_acceleration(0,0);                              //seteaza acceleratia
  set_target(0,5600);                           //jos
  delay(3500);

}

void pozitie_asteptare()
{

  servo3.write(2000);                           //stanga
  delay(230);
  servo3.write(1500);                           //opreste servoul
  delay(2000);

  set_speed(0,10);                                    //seteaza viteza
  set_acceleration(0,0);                              //seteaza acceleratia
  set_target(0,5600);                                 //jos
  delay(3500);

  set_speed(1,30);                                    //seteaza viteza
  set_target(1,6000);                           //deschide clestele
  delay(1000);

}

void setup()                                          //program principal
{                                                    //ruleaza o singura data
  Serial.begin(9600);                         //incepe comunicarea cu M.M. si PC
  servo3.attach(servoPin);                           //face legatura intre variabile
  pinMode(senzor, INPUT);                            //configureaza var. senzor ca intrare
  pinMode(LED, OUTPUT);                              //configureaza var. LED ca iesire
```

```
 delay(5000);                                //asteapta 5000ms=5s
 pozitie_asteptare();

}



void loop()                                        //program bucla
{                                                  //ruleaza la infinit

 val_senzor = analogRead(senzor);          //citire valoare senzor
 Serial.println(val_senzor);                       //afisare pe ecran valoare senzor

 if ((val_senzor < 80)&&(aux>80))
   {
     power();
     delay(1000);
     miscare();
   }
   aux=val_senzor;

   delay(250);

}
```

b. Processing

```
import processing.opengl.*;

import processing.serial.*;

Serial port;

int i,j,k,l;
PImage a;
String data = "";
int nr = 0;
String data2 = ",";
float m1 = PI/4;
float m2 = -PI/2;
float m3 = -PI/6;
```

```
void cylinder(float w, float h, int sides)
{
  float angle;
  float[] x = new float[sides+1];
  float[] z = new float[sides+1];

  //get the x and z position on a circle for all the sides
  for(int i=0; i < x.length; i++){
    angle = TWO_PI / (sides) * i;
    x[i] = sin(angle) * w;
    z[i] = cos(angle) * w;
  }

  //draw the top of the cylinder
  beginShape(TRIANGLE_FAN);

  vertex(0,   -h/2,    0);

  for(int i=0; i < x.length; i++){
    vertex(x[i], -h/2, z[i]);
  }

  endShape();

  //draw the center of the cylinder
  beginShape(QUAD_STRIP);

  for(int i=0; i < x.length; i++){
    vertex(x[i], -h/2, z[i]);
    vertex(x[i], h/2, z[i]);
  }

  endShape();

  //draw the bottom of the cylinder
  beginShape(TRIANGLE_FAN);

  vertex(0,   h/2,    0);

  for(int i=0; i < x.length; i++){
    vertex(x[i], h/2, z[i]);
  }

  endShape();
}
```

```
void baza()
{
  pushMatrix();
  translate(width/2, 495, 0);
  fill(0, 255, 0);
  box(180, 10, 180);
  popMatrix();

  pushMatrix();
  translate(width/2, 550, -100);
  fill(230);
  box(200, 100, 400);
  popMatrix();
}
void segment1()
{
  fill(100,200);
  translate(width/2, 485, 0);
  cylinder(80, 10, 100);
  translate(-width/2, -485, 0);

  fill(15, 120, 252, 150);
  translate(width/2-45, 450, 70);
  box(10, 60, 140);
  translate(90, 0, 0);
  box(10, 60, 140);
  translate(-width/2-45, -450, -70);
}

void segment2()
{
  translate(width/2-25, 355, 120);
  box(10, 250, 40);
  translate(50, 0, 0);
  box(10, 250, 40);
  translate(-width/2-25, -355, -120);
}

void segment3()
{
  translate(width/2-40, 250 , 150);
  box(10, 40, 100);
  translate(80, 0, 0);
  box(10, 40, 100);
  translate(-width/2-40, -250, -150);
  translate(width/2, 235, 210);
  box(70, 10, 100);
  translate(-width/2, -235, -210);

  translate(width/2+20, 227.5, 300);
  box(10, 5, 115);
  translate(-width/2-20, -227.5, -300);
}

void cleste1()
{
  translate(width/2-20, 227.5, 270);
  box(10, 5, 60);
  translate(-width/2+20, -227.5, -270);
}

void cleste2()
{
  translate(width/2-20, 232.5, 332.5);
  box(10, 5, 75);
  translate(-width/2+20, -232.5, -332.5);
}

void setup()
{
  size(600, 700, OPENGL);
  smooth();
  a = loadImage("wood2.jpg");
  port = new Serial(this, "COM4", 9600);
  port.bufferUntil('\n');

}

void tex()
{
  noStroke();
  noFill();
  beginShape();
  texture(a);
  vertex(0, 600, -400, 0, 0);
  vertex(0, 600, 600, 0, 3392);
  vertex(600, 600, 600, 2560, 3392);
  vertex(600, 600, -400, 2560, 0);
  endShape();
}
```

```
void draw()
{

  background(255);
  lights();

  rotateY(PI/6);
  translate(90, 0, -300);

  tex();

  stroke(0);

  baza();

  translate(width/2, 495, 0);
  rotateY(m1);
  translate(-width/2, -495, 0);

  segment1();

  translate(width/2, 450, 120);
  rotateX(m2);
  translate(-width/2, -450, -120);

  segment2();

  translate(width/2, 250, 120);
  rotateX(-m2);
  translate(-width/2, -250, -120);

  segment3();

  translate(width/2-20, 227.5, 247.5);
  rotateY(m3);
  translate(-width/2+20, -227.5, -247.5);

  cleste1();

  translate(width/2-20, 232.5, 297.5);
  rotateY(-m3);
  translate(-width/2+20, -232.5, -297.5);

  cleste2();
  if ((nr == 4) && (data2 != data))
```

```
    m1 = PI/4;

  if ((nr ==5) && (data2 != data))
    m1 = -PI/4;

  if ((nr == 3) && (data2 != data))
    m2 = -PI/2;

  if ((nr == 1) && (data2 != data))
    m2 = 0;

  if ((nr == 6) && (data2 != data))
    m3 = PI/6;

  if ((nr == 9) && (data2 != data))
    m3 = -PI/6;

  if ((data != "") && (data2 != data))
{
    println(data);
    println(data.length());
    data2 = data;
}
}


void serialEvent (Serial port)
{
  data = port.readStringUntil('\n');
  if (data.length() > 1)
    {
      data = data.substring(0, data.length()-2);
      nr = data.length();
    }

}
```