T.C.
İSTANBUL
KÜLTÜR
ÜNİVERSİTESİ

Engineering Faculty

Computer Engineering Department

CSE0420 Embedded Systems

Fall 2023/2024

Color Sorting Machine

1. Introduction:

In this project, An Arduino uno microcontroller, a 3-6V DC motor, a color sensor (TCS3200), 2 servo motors, motor driver L298N and a 6-volts or 9-volts battery will be used to design and construct a color sorting machine. The color sorting machine will be constructed in the form of a conveyor belt and to do so we will be using the DC motor along with the motor driver L298N since we will be using more than 5-Volts and the Arduino uno microcontroller has only 5-volts. The conveyor belt requires some understanding of the mechanics behind how we can make a conveyor belt using a DC motor only, using the most basic components to build such mechanism we used a plastic food container along with 2-rollers and a rubber belt.

The main objective of the project is to utilize and assemble the components we have and achieve our goal which is to design and build a color sorting machine. which will take the shape of a conveyor belt, is to arrange items according to color. The DC motor that the Arduino Uno and the L298N motor driver control will power the conveyor belt system.

The project's scope involves the creation of the color sorting system, its design and construction, and the selection and integration of the required hardware elements, including the Arduino uno microcontroller, a 3-6V DC motor, a color sensor (TCS3200), 2 servo motors, motor driver L298N and a 6-volts or 9-volts battery. The system will work as one according to a set of instructions defined and a set of colors predefined. A controlled testing environment will be used to evaluate the accuracy of the sorting process and recognition of the colors and adjust it accordingly.

2. The components used in the project:

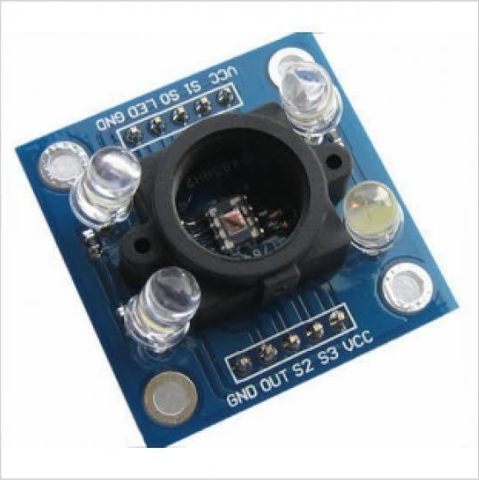
1. SERVO MOTOR

A servo motor is a type of motor that is specifically designed to provide precise control of angular position, velocity, and acceleration. It consists of a DC motor, a set of gears, and a feedback mechanism that allows the motor to rotate to a specific position and hold that position with a high degree of accuracy. They are often used to move and position objects, or to rotate and align parts in a precise manner.

2. DC MOTOR

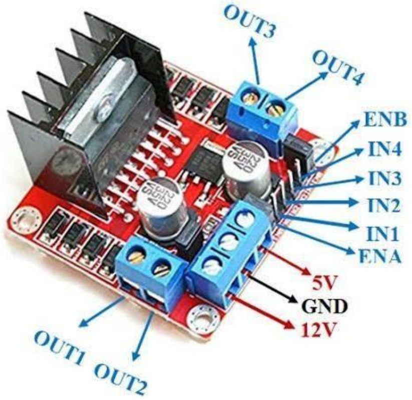
A DC motor is an electrical machine that converts direct current (DC) electrical energy into mechanical energy. DC motors have a simple design and are relatively easy to build, making them widely used in a variety of applications.

3. COLOR SENSOR (TCS3200)



TCS3200 is a color sensor capable of detecting a wide range of colors. It uses an array of photodiodes and a white LED to measure the intensity of light reflected from an object. By analyzing these readings, it can identify and quantify the color of the object.

4. MOTOR DRIVER



The L293N is a motor driver IC that controls the direction and speed of DC motors. It can drive two motors bidirectionally and is often used in robotics projects. The IC handles the motor-driving functions, simplifying the control of motors with microcontrollers.

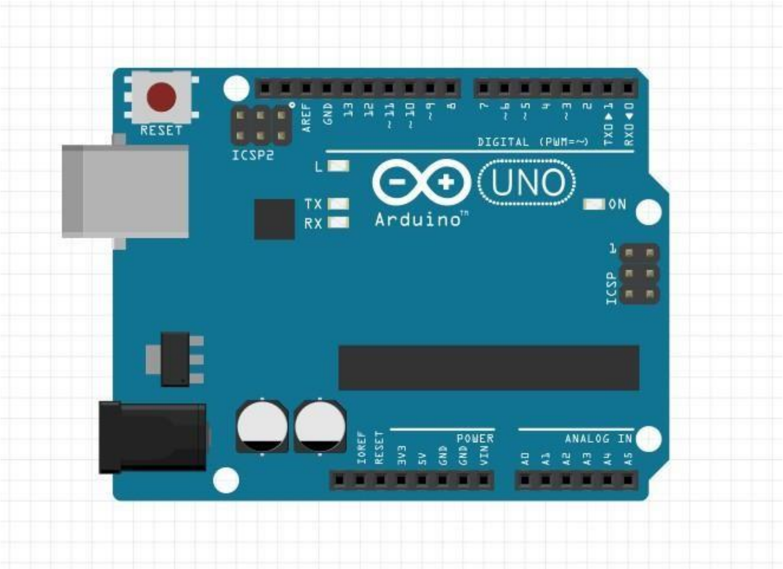
5. Batteries

AA batteries are a very easy and sufficient way to provide our sorting machine with power. In this Project, we used 4 AA batteries. Which can provide us with 6 volts since each battery has 1.5 volts. 4 pill AA battery case holder is used to connect them to the external power supply within the motor driver shield.

6. JUMBER CABLES

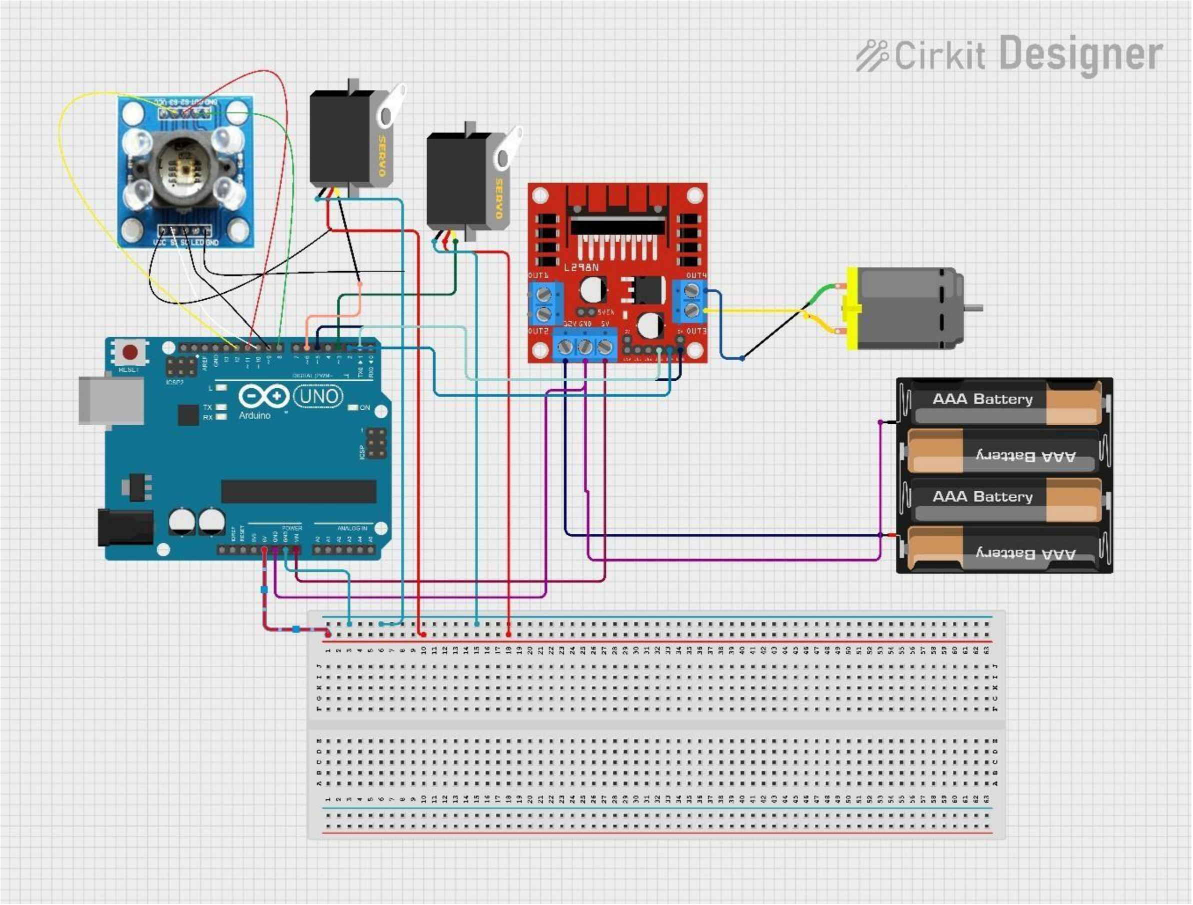
jumper cables are small wires with connectors on each-end that are used to connect electronic components together. They are commonly used with the Arduino microcontroller board to connect sensors, actuators, and other components to the board. In this project, Male-to-Female & Male-To-Male cables were used.

7. ARDUINO UNO



Arduino Uno is a popular open-source microcontroller board designed for easy prototyping and programming. Using the Arduino uno we can code and command the components we have.

3. The Design:



1. Color Sensor (TCS3200) - (OUT) is connected to pin 8.
2. Color Sensor (TCS3200) - (S0) is connected to pin 9.
3. Color Sensor (TCS3200) - (S1) is connected to pin 10.
4. Color Sensor (TCS3200) - (S2) is connected to pin 11.
5. Color Sensor (TCS3200) - (S3) is connected to pin 12.
6. Color Sensor (TCS3200) - (VCC) is connected to breadboard VCC line.
7. Color Sensor (TCS3200) - (GND) is connected to breadboard GND line.
8. Servo Motor ServorRED – (Digital pin) is connected to pin 3.
9. Servo Motor ServorRED - (VCC) is connected to breadboard VCC line.
10. Servo Motor ServorRED - (GND) is connected to breadboard GND line.
11. Servo Motor ServorBLUE – (Digital pin) is connected to pin 6.
12. Servo Motor ServorBLUE - (VCC) is connected to breadboard VCC line.
13. Servo Motor ServorBLUE - (GND) is connected to breadboard GND line.
14. DC motor (Motor shield input occupies Digital pin 1 & 4).
15. Motor Driver ENA is connected to pin 5.
16. Batteries are connected to external power supply on the Motor shield.
17. Arduino VIN pin is connected to 5 volts pin of the motor driver.
18. All sensors ground and vcc pins are connected directly to available 5v+ and GND pins on the breadboard.

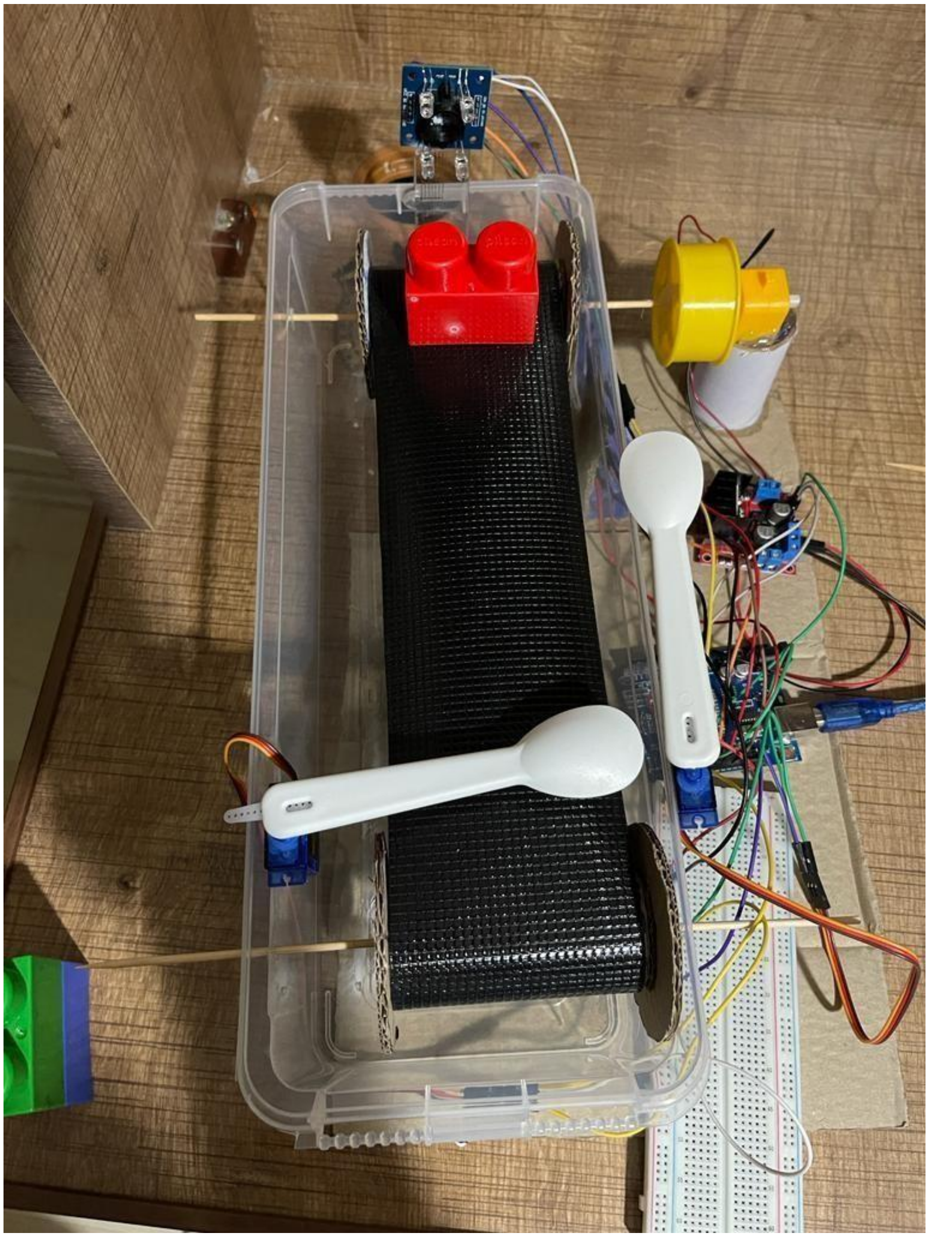
4. Code:

```
#include <Servo.h>

Servo servoRed;   // Create a servo object for the red color
Servo servoBlue;  // Create a servo object for the blue color
//defining the color sensor pins
#define outPin 8
#define s0 9
#define s1 10
#define s2 11
#define s3 12
int in1 = 2;
int in2 = 7;
int ena = 5;
boolean DEBUG = true;

// Variables
int red, grn, blu;
String color = "";
long startTiming = 0;
long elapsedTime = 0;

void setup(){
  Serial.begin(9600);
servoRed.write(10);    // Move the red servo back to 0 degrees
 pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(ena, OUTPUT);
servoBlue.write(180);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(outPin, INPUT); // out from sensor becomes input to Arduino

  servoRed.attach(6);   // Attach the servo for the red color to pin 6
  servoBlue.attach(3);  // Attach the servo for the blue color to pin 3

  // Setting frequency scaling to 100%
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);



  startTiming = millis();
}
void moveServoRed() {
servoRed.write(100);   // Move the blue servo to 90 degrees
dcmotor();
delay(500);          // Wait for 2 seconds
servoRed.write(10);    // Move the red servo back to 0 degrees
digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(ena, 0);

}
```

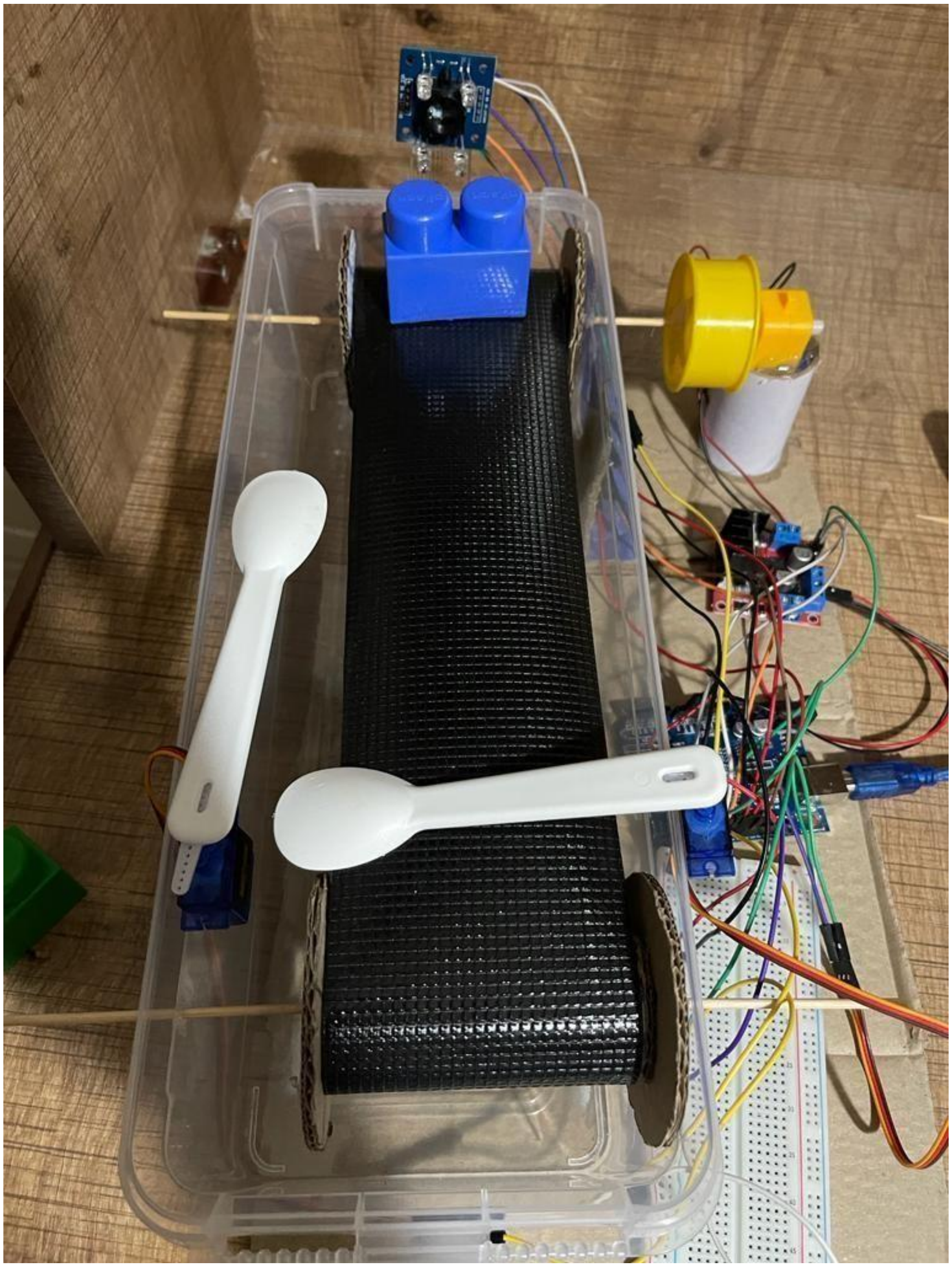
```
//servo for blue color
void moveServoBlue() {
  servoBlue.write(90);   // Move the blue servo to 90 degrees
  dcmotor();
  delay(500);           // Wait for 2 seconds
  servoBlue.write(180);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(ena, 0);


}
//if the color sensor is detecting green then do not move no servo motors and let the green object pass
void sortingGREEN() {


  dcmotor();
  delay(1500);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(ena, 0);
}
void dcmotor(){
 digitalWrite(in1, HIGH);
 digitalWrite(in2, LOW);
  analogWrite(ena, 210); // control the motor speed
  delay(500);


  // Run for a duration (you can adjust the time or conditions as needed)
//  delay(3000);

  // Stop the motor
  // digitalWrite(in1, LOW);
  // digitalWrite(in2, LOW);
  // analogWrite(ena, 0);
  //delay(1000);

}



void loop(){



  // Wait for a moment before repeating
  getColor();

  if (DEBUG) printData();
  elapsedTime = millis() - startTiming;
  if (elapsedTime > 1000) {
    if (color == "RED") { //the color is detected as red we will call the
moveservoblue function
      moveServoRed();
      //dcmotor();
    } else if (color == "BLUE") {
      moveServoBlue();
      //dcmotor();
    } else if (color =="GREEN"){
      sortingGREEN();
    }
    startTiming = millis();
  }


}


// read RGB components
void readRGB(){
  red = 0, grn = 0, blu = 0;

  int n = 10;
  for (int i = 0; i < n; ++i){
    //read red component
    digitalWrite(s2, LOW);
    digitalWrite(s3, LOW);
    red = red + pulseIn(outPin, LOW);

    //read green component
    digitalWrite(s2, HIGH);
    digitalWrite(s3, HIGH);
    grn = grn + pulseIn(outPin, LOW);

    //read blue component
    digitalWrite(s2, LOW);
    digitalWrite(s3, HIGH);
    blu = blu + pulseIn(outPin, LOW);
  }
  red = red / n;
  grn = grn / n;
  blu = blu / n;
}


//Showing captured data at Serial Monitor
void printData(void){
  Serial.print("red= ");
  Serial.print(red);
  Serial.print("   green= ");
  Serial.print(grn);
  Serial.print("   blue= ");
  Serial.print(blu);
  Serial.print(" - ");
  Serial.print(color);
  Serial.println(" detected!");
}
 //here we will determine which color is being detected base on the values we have prefined
void getColor(){
  readRGB();

  if (red > 40 && red < 90 && grn > 150 && grn < 230 && blu > 130 && blu < 200) color = "RED";
  else if (red > 120 && red < 400 && grn > 100 && grn < 190 && blu > 170 && blu < 240) color = "GREEN";
  else if (red > 70 && red < 200  && grn > 70 && grn < 170 && blu > 30  && blu < 95) color = "BLUE";

  else  color = "NO_COLOR";



  delay(500);

}

//servo for red color
```

5. Demo:

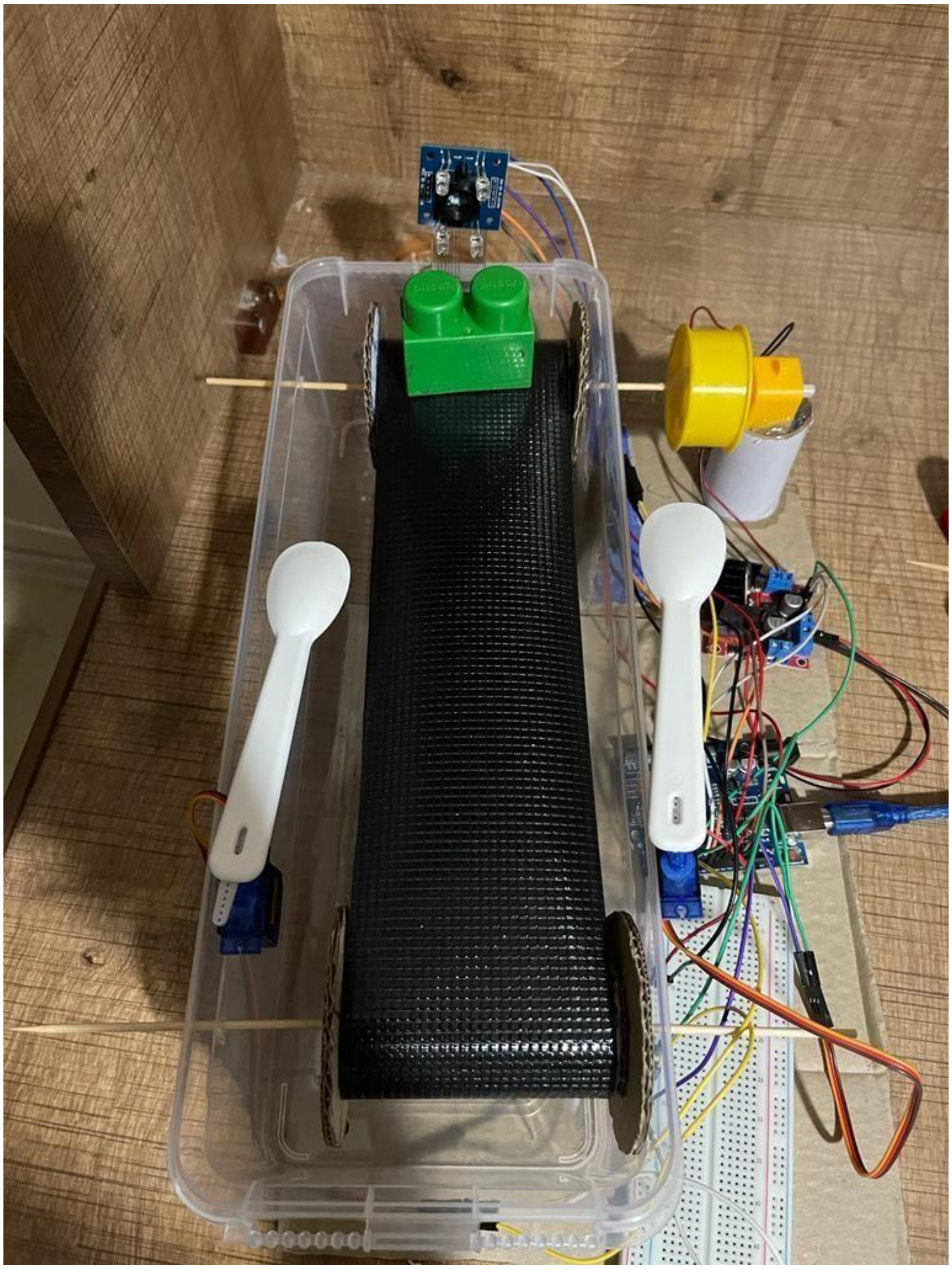
In the following image, we have placed a red object in front of the color sensor to determine whether we will let it pass through or push it to the left or right. The color is detected as red thus we will make the left servo motor to be 90 degrees and keep the right servo motor to be 0.

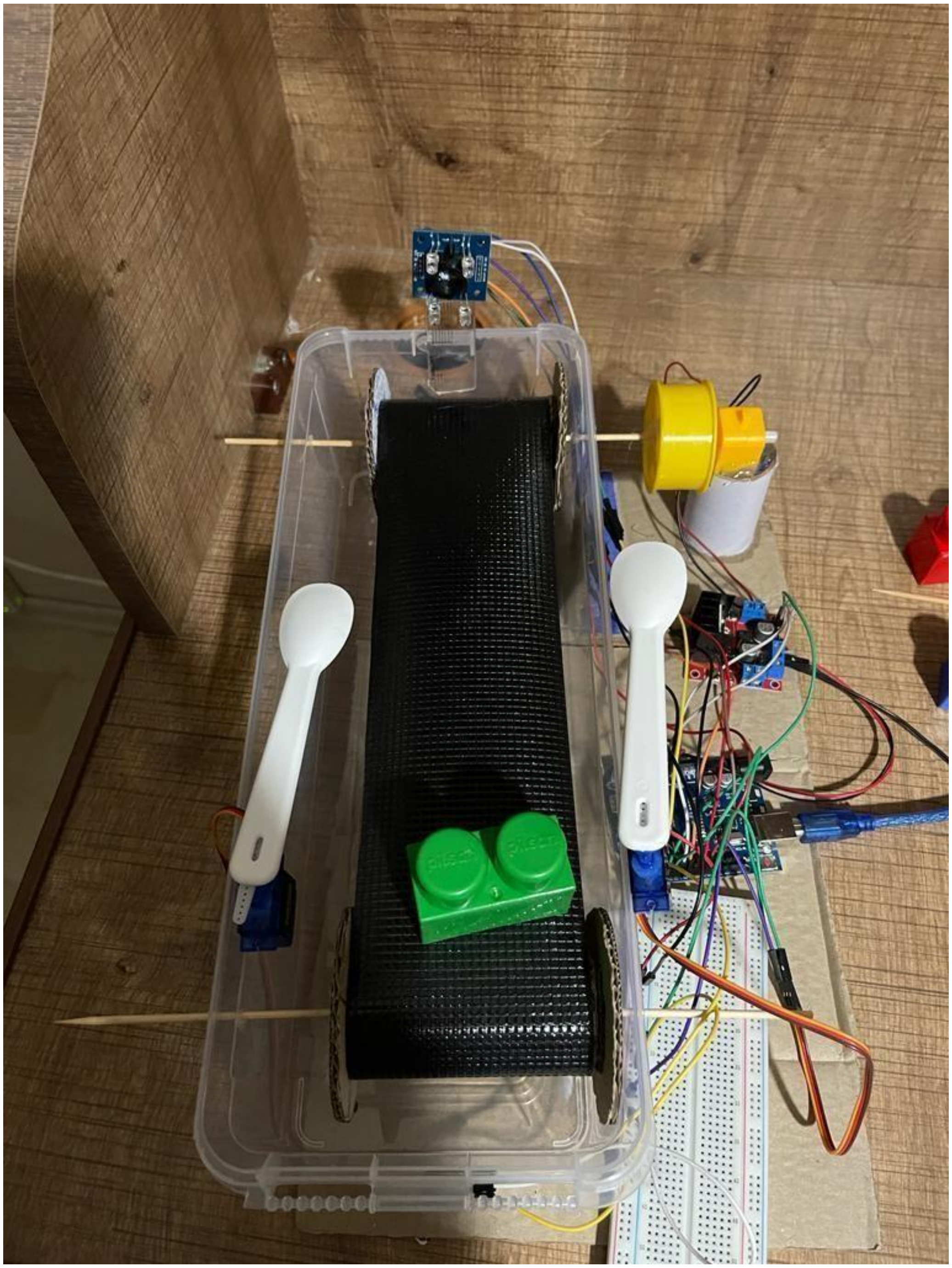


In the following image, we have placed a blue object in front of the color sensor to determine whether we will let it pass through or push it to the left or right. The color is detected as blue thus we will make the right servo motor to be 90 degrees and keep the left servo motor to be 0.



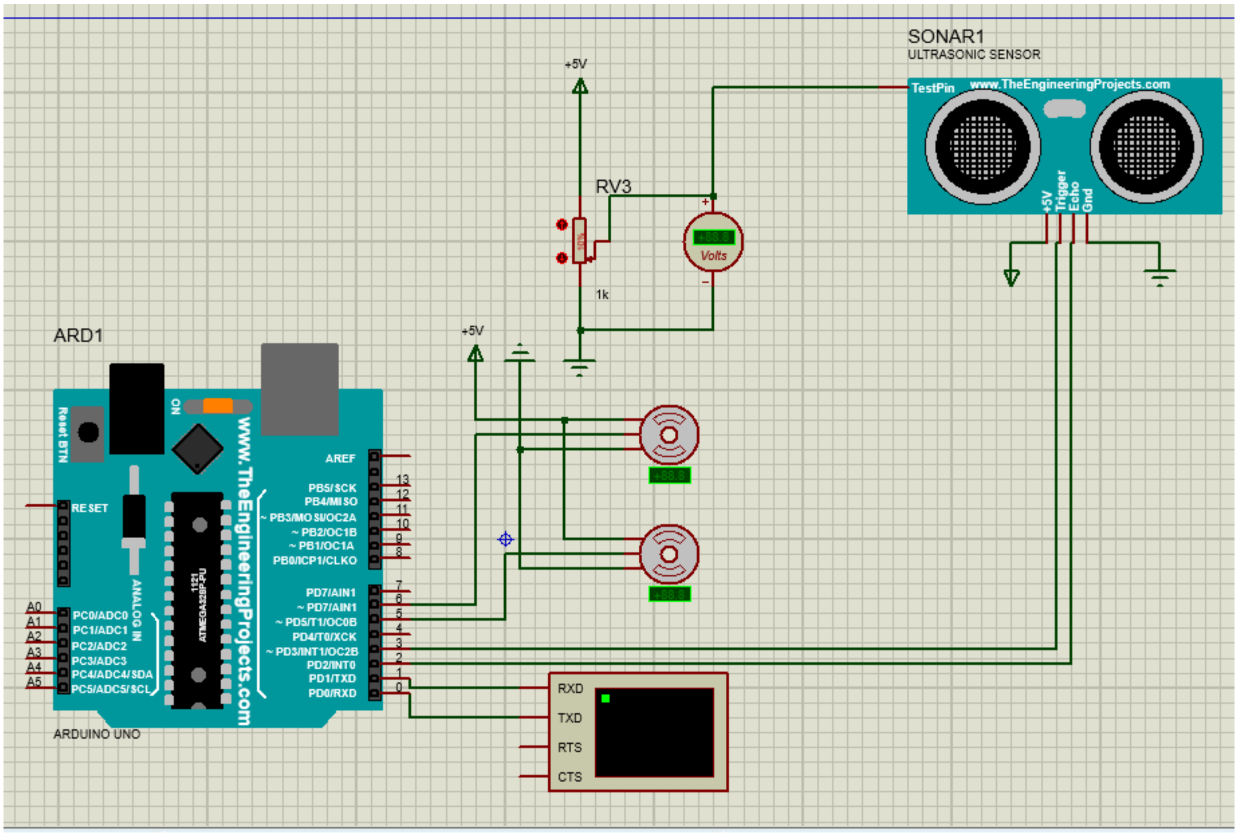
In the following image, we have placed a green object in front of the color sensor to determine whether we will let it pass through or push it to the left or right. The color is detected as green thus we will make the both servo motors to be 0 degrees and let it pass through.





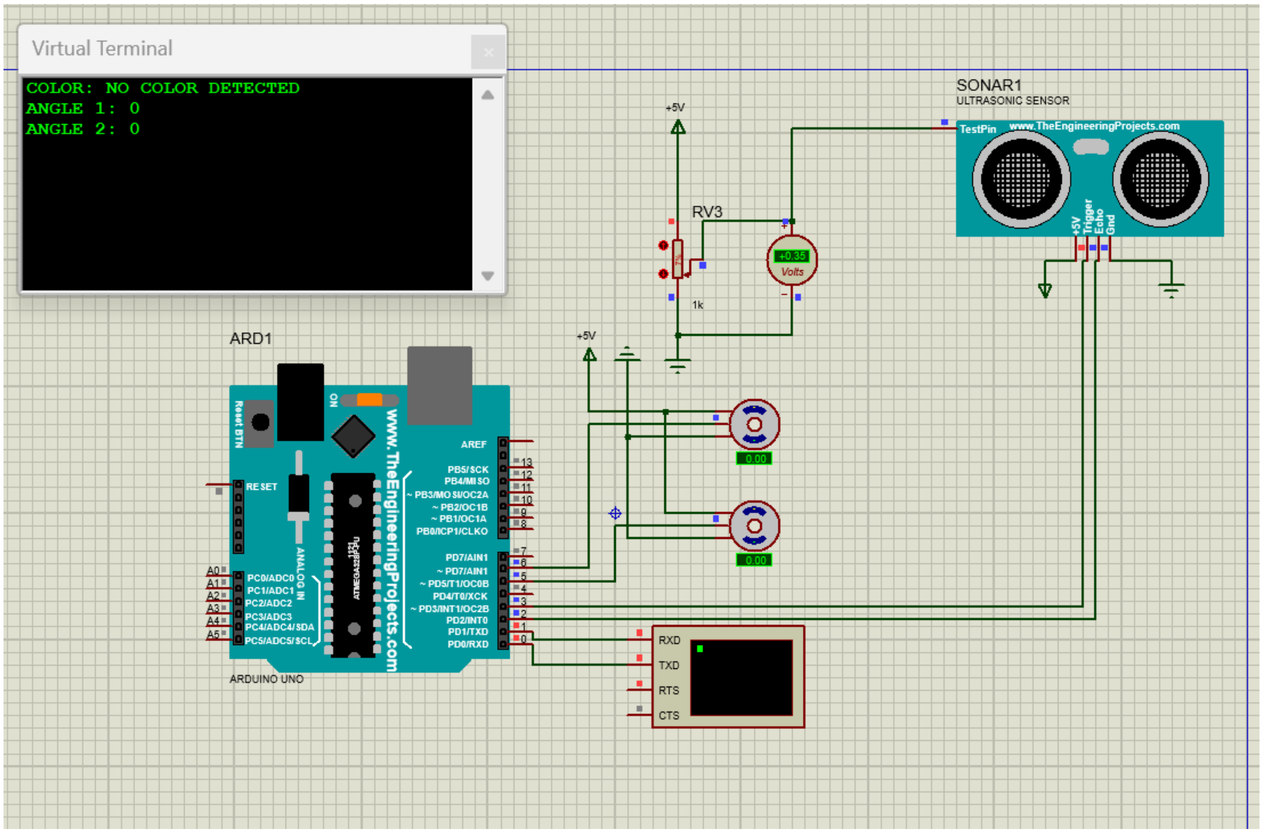
6. Proteus simulation:

- **Proteus simulation before running the simulation.**



Since we don't have the color sensor in proteus we have used the ultra-sonic sensor instead, the ultra-sonic acts as its detecting colors by predefined values. We have set the values to mimic the color values.

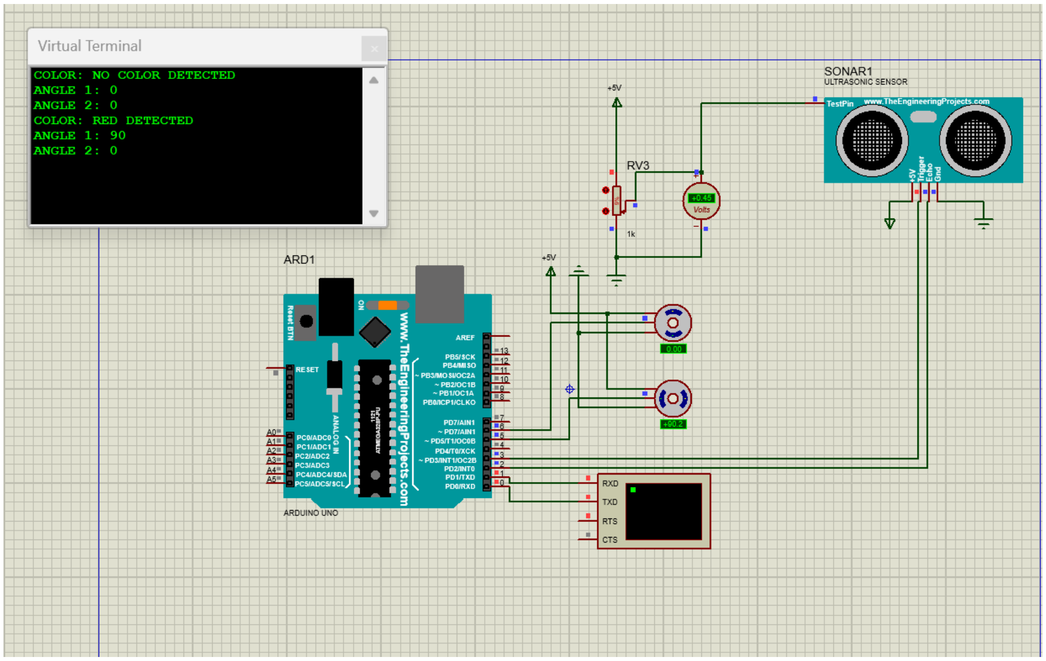
First state:



```
COLOR: NO COLOR DETECTED
ANGLE 1: 0
ANGLE 2: 0
```

As depicted in the image above, when we have values from the ultra-sonic as lower than 100 that means we don't have any colors detected at the moment. Thus, we will be displaying that there are no colors, and we will set the servo motors to 0.

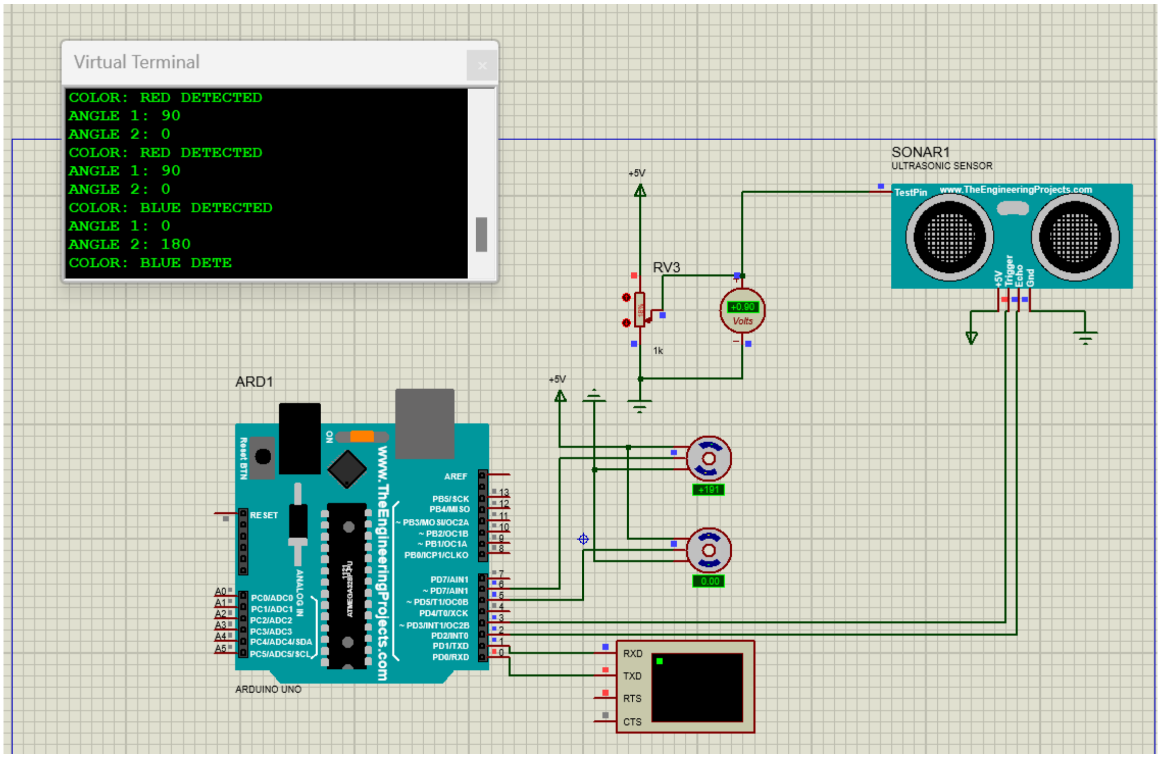
second state:



```
COLOR: NO COLOR DETECTED
ANGLE 1: 0
ANGLE 2: 0
COLOR: RED DETECTED
ANGLE 1: 90
ANGLE 2: 0
```

as depicted in the above image, we have detected the color Red, the way that happened is that we have defined that if we get values between 100 and 200, we will say that it’s a Red color and let one of the servo motor to be 90 degrees or closed.

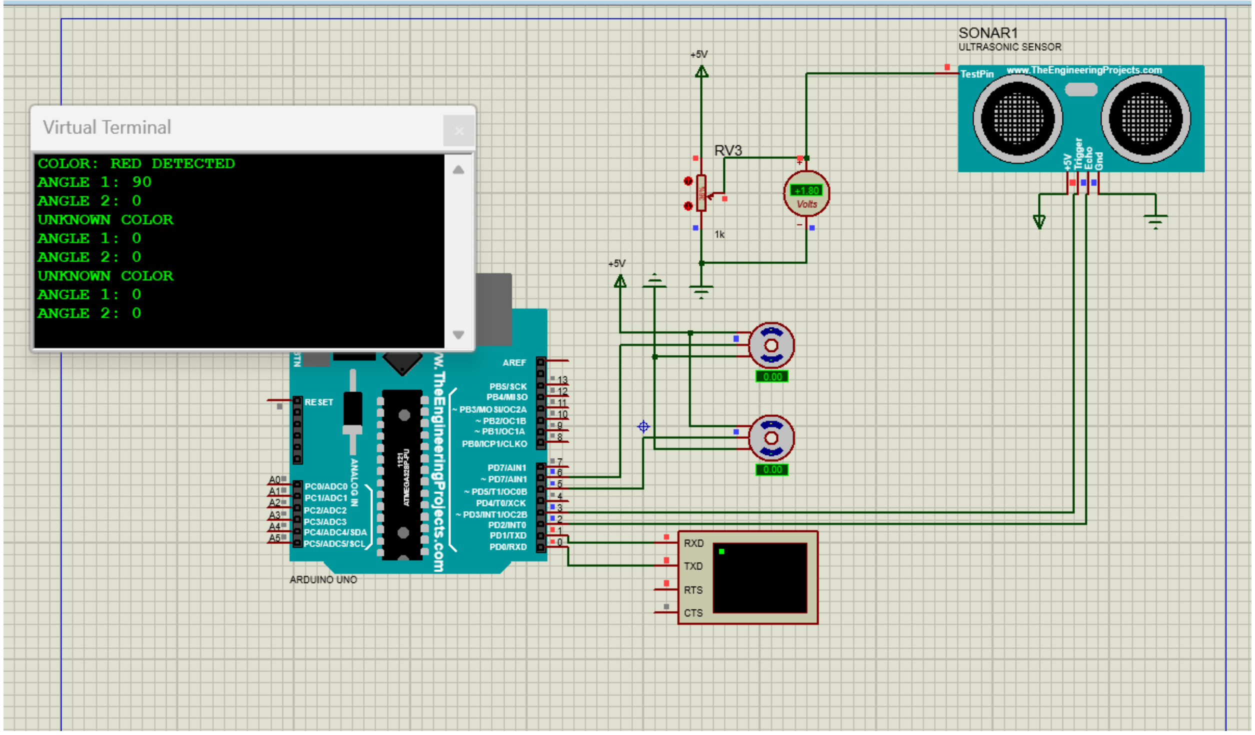
third state:



```
COLOR: RED DETECTED
ANGLE 1: 90
ANGLE 2: 0
COLOR: RED DETECTED
ANGLE 1: 90
ANGLE 2: 0
COLOR: BLUE DETECTED
ANGLE 1: 0
ANGLE 2: 180
COLOR: BLUE DETE
```

as depicted in the above image, we have detected the color Blue, the way that happened is that we have defined that if we get values between 200 and 300, we will say that it's a blue color and let one of the servo motor to be 180 degrees or closed.

fourth state:



```
COLOR: RED DETECTED
ANGLE 1: 90
ANGLE 2: 0
UNKNOWN COLOR
ANGLE 1: 0
ANGLE 2: 0
UNKNOWN COLOR
ANGLE 1: 0
ANGLE 2: 0
```

as depicted in the above image, we have detected an UNKNOWN color, the way that happened is that we have defined that if we get values between 300 and 1000, we will say that it’s an UNKNOWN color and let one of the servo motors to be 0 degrees or open.

Proteus code:

```
#include <Servo.h>


#define triggerPin 3

#define echoPin 2


long duration, distance;

long angle1, angle2;

Servo servoMotorRED;

Servo servoMotorBLUE;
```

```
void setup() {
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);

  servoMotorRED.attach(5); // Pin 5 for servo motor RED
  servoMotorBLUE.attach(6); // Pin 6 for servo motor BLUE

  Serial.begin(9600);

  // Set the initial servo motor values to 0
  angle1 = 0.0;
  angle2 = 0.0;
  servoMotorRED.write(angle1); //print the angle to the serial monitor
  servoMotorBLUE.write(angle2);//print the angle to the serial monitor
}

void loop() {
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance in CM
  distance = ((duration / 29) / 2);
  delay(50);
//if we have distance as less than 100 we have no color and we keep the servo at 0
  if (distance < 100) {
    Serial.println("COLOR: NO COLOR DETECTED");//print the to the serial monitor
    Serial.println("ANGLE 1: 0");
    Serial.println("ANGLE 2: 0");
    angle1 = 0.0;
    angle2 = 0.0;
    servoMotorRED.write(angle1);
    servoMotorBLUE.write(angle2);
    //if we have distance as less than 100 we have red color and we keep the servo at 90 and the other servo at 0
  } else if (distance >= 100 && distance < 200) {
    Serial.println("COLOR: RED DETECTED"); //print red to the serial monitor
    Serial.println("ANGLE 1: 90");
    Serial.println("ANGLE 2: 0");
    angle1 = 90.0;
    angle2 = 0.0;
    servoMotorRED.write(angle1);
    servoMotorBLUE.write(angle2);
    //if we have distance as less than 100 we have blue color and we keep the servo at 0 and the other servo at 180
  } else if (distance >= 200 && distance < 300) {
    Serial.println("COLOR: BLUE DETECTED");//print blue to the serial monitor
    Serial.println("ANGLE 1: 0");
    Serial.println("ANGLE 2: 180");
    angle1 = 0.0;
    angle2 = 180.0;
    servoMotorRED.write(angle1);
    servoMotorBLUE.write(angle2);
  }
  //if we have distance as less than 100 we have unknown color and we keep the both servo motors at 0
 else if  (distance >= 300 && distance < 1000) {
    Serial.println("UNKNOWN COLOR"); //print unknown to the serial monitor
    Serial.println("ANGLE 1: 0");
    Serial.println("ANGLE 2: 0");
    angle1 = 0.0;
    angle2 = 0.0;
    servoMotorRED.write(angle1);
    servoMotorBLUE.write(angle2);
  }

  delay(1000); // Wait one second and do the same operations
}
```

References:

- https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/
- https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/
- https://docs.arduino.cc/learn/electronics/servo-motors
- https://howtomechatronics.com/projects/arduino-color-sorter-project/
- https://circuitdigest.com/microcontroller-projects/arduino-color-sorter-machine-using-tcs3200-color-sensor