Term project submitted for the Course:

Mechatronics laboratory (ME69036)

Spring semester 2022.

PICK AND PLACE ROBOTIC ARM

Involves the application of ARDUINO, Color Detecting Sensor and Motors.

Submitted to:

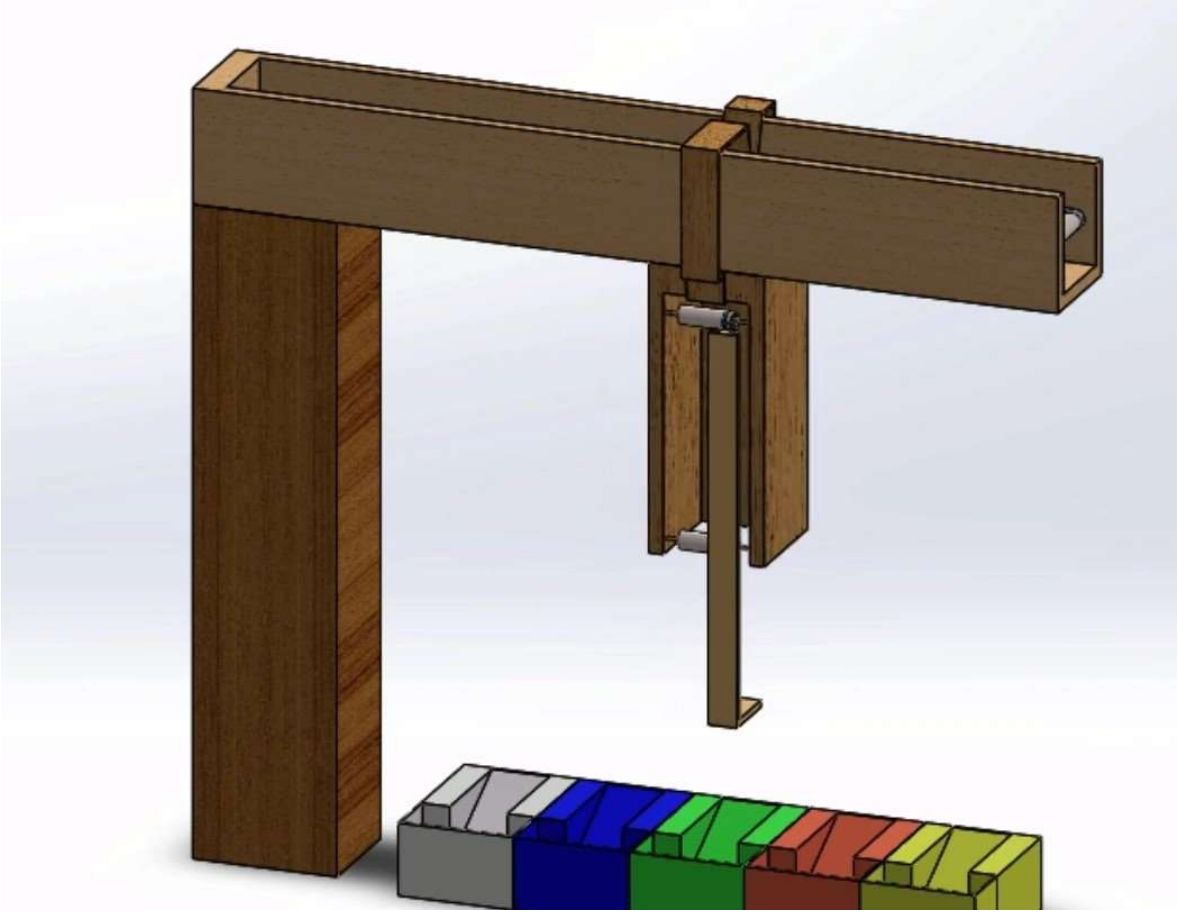
Professor Arun Kumar Samantaray
Professor Mihir Sarangi
Professor Cheruvu Siva Kumar

Submitted by:

Sreerag V	(21RJ92R02)
Gobind Kumar	(21ME63R36)
Vaibhav Tyagi	(21ME63R37)
Allam Vinay Nagendra Ramtarun	(21ME63R39)
Keshava G	(21ME63R40)
Komati Reddy Nitin Reddy	(21ME63R41)

Basic Idea of the project

- A pre-programmed Robotic arm controlled with an Arduino, picks a user specified colored ball among different balls, with the help of color detecting sensor, and places it near the user.



Objectives:

- Understanding the working principles and programming methods of:
- Color detecting sensor
- Stepper motor
- Servo motor
- Developing a C++ code to establish a sync between Color detecting sensor, Stepper motor and servo motor.
- Creating the circuit connections in order to establish the sync physically in the assembly.
- Projecting our basic idea as clear as possible with a physical assembly involving procurement of necessary materials, design, teamwork and understanding the non- technical aspects of the project.

Abstract:

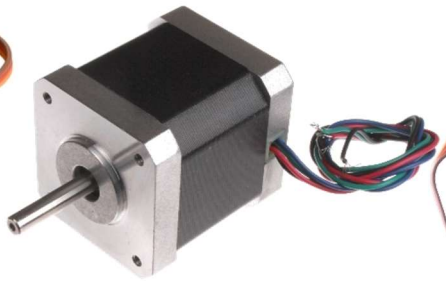
A Robotic arm is controlled and specified to perform an automatic task of detecting and picking a coloured ball and placing it in a predefined position. The task consists of two major things. One is the detecting the user specified coloured ball, with the help of colour detecting sensor and the second thing is to pick and place the same ball , in a predefined position, using a mechanical gripper. Both these sub-tasks are co-ordinated with each other and controlled using an Arduino UNO board, which acts as a motherboard to this entire task. It takes in power required for the task from an external source (Battery) and set of instructions from the user, co-ordinate the functions of each component attached to it and performs the task according to the C++ program instructed by the user.

- Components:

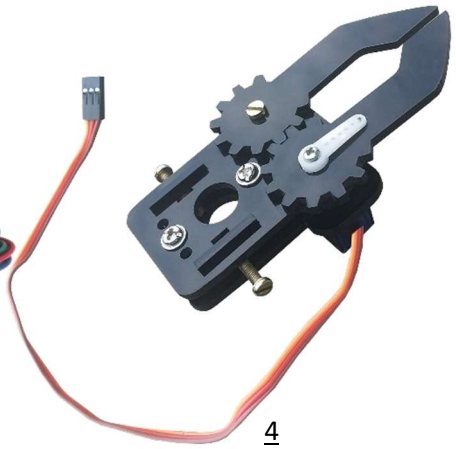
1

4

5



2

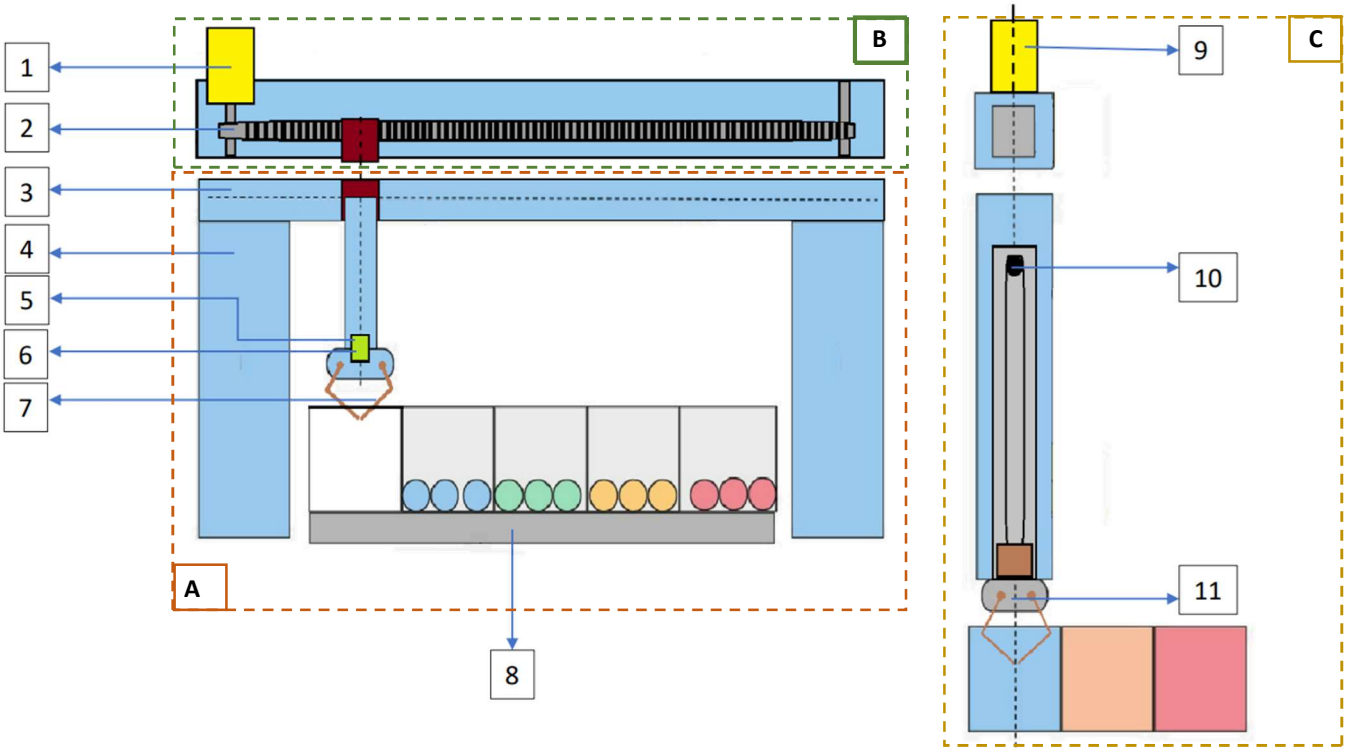


4

S. No.	Component	Specifications	Quantity	Price (Rs/- Per unit)
1.	Arduino UNO board	Microcontroller : ATmega328 Operating : Voltage 5V Input Voltage (recommended) : 7-12V Input Voltage (limits) : 6-20V Digital I/O : Pins 14 (of which 6 provide PWM output) Analog Input Pins : 6 Flash Memory : 32 KB of which 0.5 KB used by bootloader	1	-
2.	Stepper motor	Rated Voltage : 12V DC Current : 1.2A at 4V Step Angle : 1.8 deg. No. of Phases : 4 Motor Length : 1.54 inches steps per revolution : 200 Operating Temperature : -10 to 40 °C Unipolar Holding Torque : 22.2 oz-in	1	550
3.	Motor Driver module	Minimum operating voltage : 8V Maximum operating voltage : 35V Continuous current per phase : 1 Amp Maximum current per phase : 2 Amp Minimum logic voltage : 3V Maximum logic voltage : 5.5V Micro step resolutions : Full, 1/2, 1/4, 1/8, and 1/16 Reverse voltage protection? : No	1	95
4.	Servo motor	Operating Voltage : +5V Torque : 2.5kg/cm Operating speed : 0.1s/60° Gear Type : Plastic Rotation : Plastic Weight of motor : 37 grams	1	100

5.	Mechanical Gripper	Material Type Weight Arm's length Open arms space	: Acrylic : 240 grams : 7.3 cms : 10 cms	1	360
6.	Colour detecting sensor	Model Power Size Interface	: TCS 3200 : 2.7V to 5.5V : 28.4 x 28.4mm (1.12 x 1.12") : Digital TTL (Transistor -Transistor logic)	1	460
7.	Timing belt	Length Width	: 5 meters : 6 mm	1	550
8.	Pulleys	Bore diameter No. of teeth	: 5 mm : 20 teeth	4	120
9.	Balls			6	30
10.	Miscellaneous				300
Total cost					2565

- Schematic diagram:



Parts:

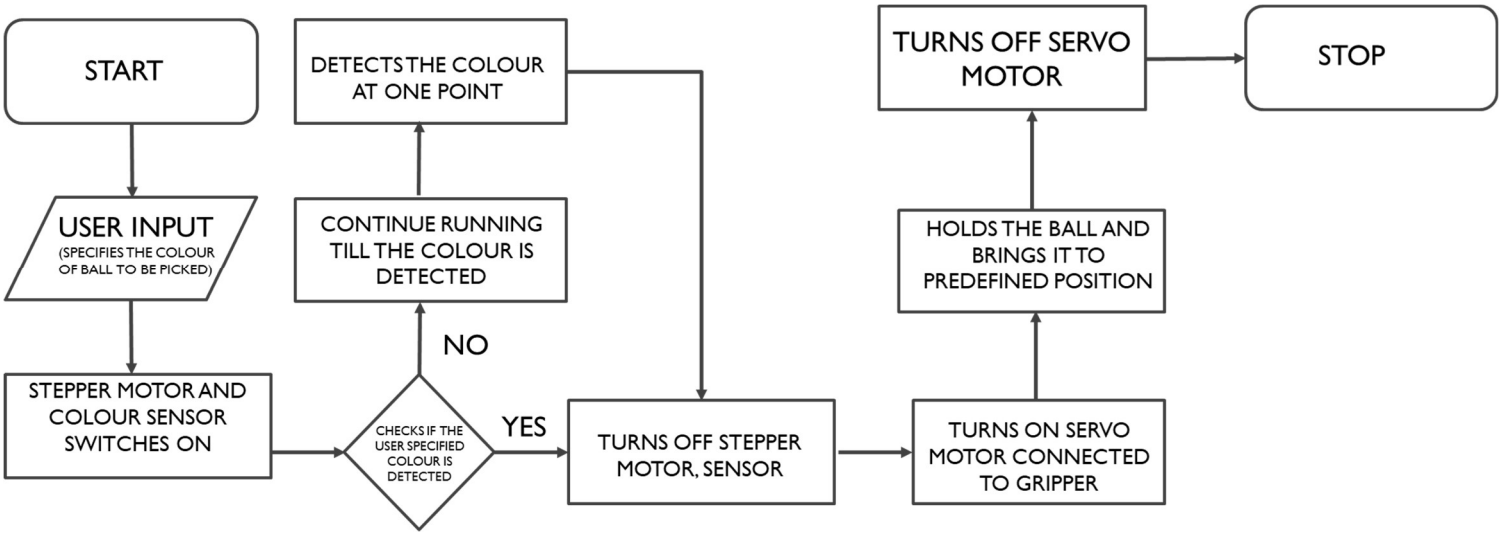
1.	Primary motor: 9V DC motor	2.	Timing Belt and Pulley
3.	Horizontal C section member	4.	Supporting member
5.	Moving C section arm	6.	Colour detecting sensor
7.&11.	Mechanical Gripper with servo motor	8.	Platform with Boxes containing Balls
9.	Secondary Motor : 5V stepper motor	10.	Timing Belt and Pulley

A: Front view of the mechanism B: Top view of the Horizontal C section C: Enlarged view of Moving C section arm

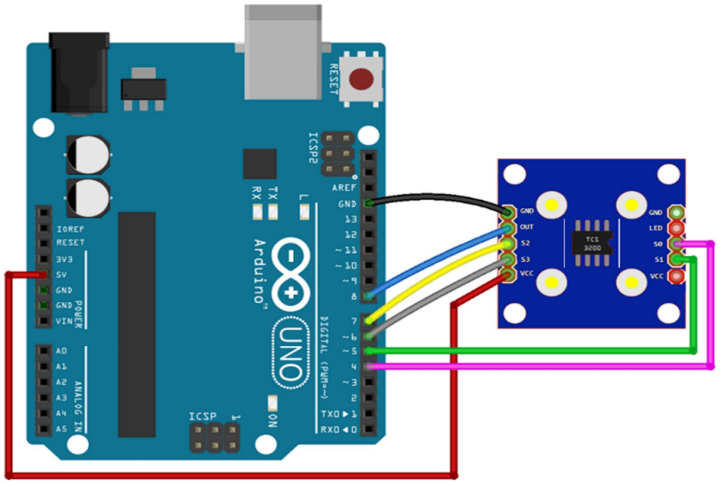
▪ Working principle:

The mechanism starts with the user, specifying the input in terms of which coloured ball the Robotic arm is required to pick and place it near him. This user input is given through the push buttons. There are three push buttons, one for each colour of the ball i.e. Red, Blue and green. Upon receiving the signal, the Primary Motor [1] starts rotating , which in turn moves the Moving C section arm [5]. This motion transmission from primary motor to the moving arm is ensured using Timing Belt and Pulleys[2]. The colour detecting sensor [6] attached to the moving arm also gets activated as soon as primary motor starts rotating and continuously detects the colour nearby it. As soon as the sensor detects the user specified colour, the primary motor stops rotating and along with it, after a certain second of delay, the secondary motor [9] switches on and lower the Mechanical gripper [7] using the timing belt and pulley. As soon as the Gripper reaches the base of box, the arms of the gripper, which were kept open till that point, closes gradually and firmly hold the ball by controlling the servo motor connected to the arms. Then again after ensuring the balls is firmly held by arms of gripper, the secondary motor [9] rotates in opposite direction, lifts up the mechanical gripper [7]. At the point, the gripper reaches original position, the primary motor [1] starts rotating in the opposite direction until the moving C section arm [5] reaches its original position. As soon as the moving arm reach the home position i.e. empty box, the servo motor at the gripper opens the arms rotating by same amount as it did while closing, and releases the ball. The ball is then picked up from the empty box by the user. This marks the end of mechanism and hence all the components were switched off until the next input.

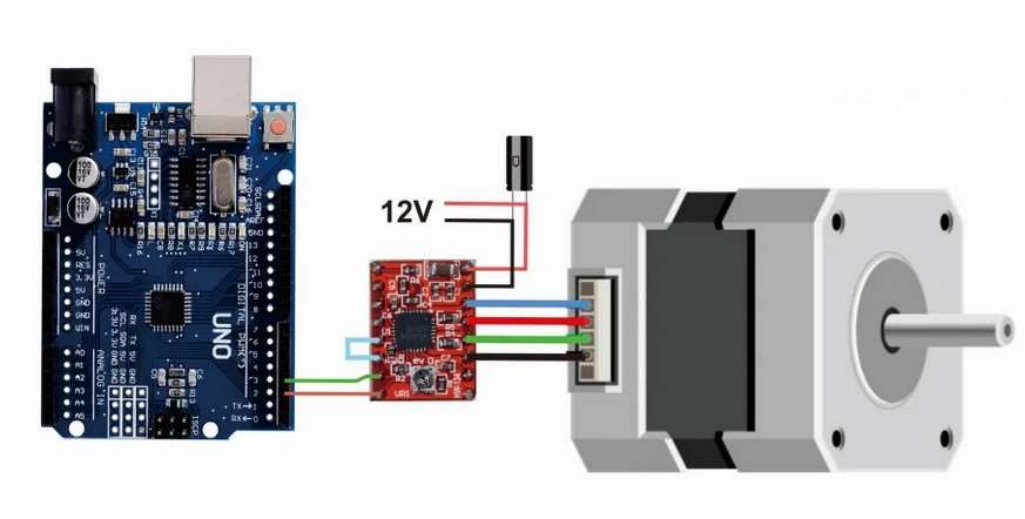
▪ Flow diagram:



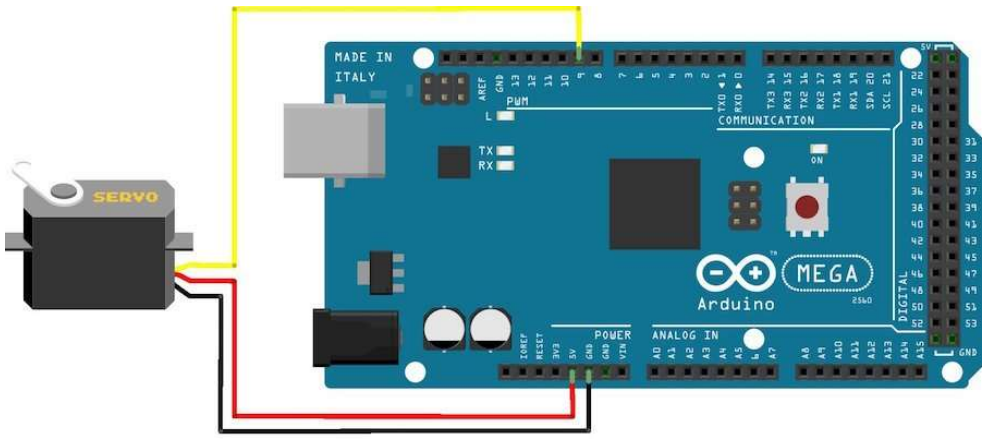
▪ Circuit diagrams:



a) Colour detecting sensor circuit



b) Stepper motor circuit



c) Servo motor circuit

- Arduino code:

```
// TCS230 or TCS3200 pins wiring to Arduino

#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6
String c = "";
int a,b;

int dummyvar = 0;
int z;
int countrevs_stepper = 0;

//Gripper servo
#include <Servo.h>
int pos = 0;
Servo servo_11;

// Define pin connections & motor's steps per revolution
const int dirPin = 8;
const int stepPin = 9;
const int stepsPerRevolution = 60;
int enablepin = 10;
```

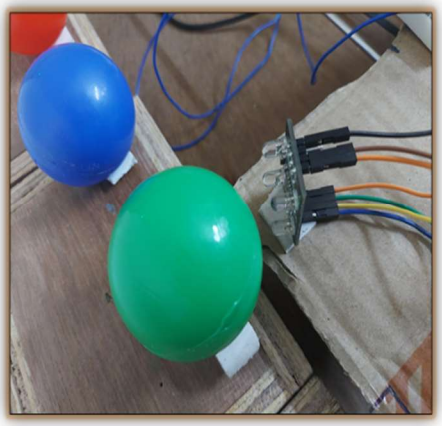
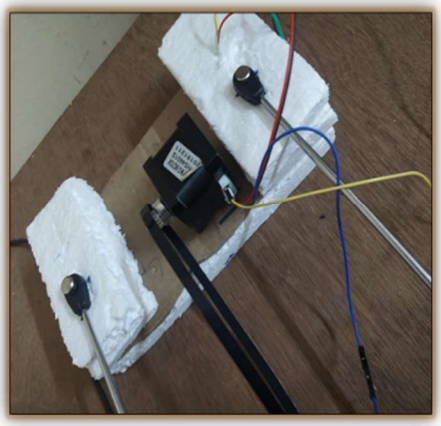
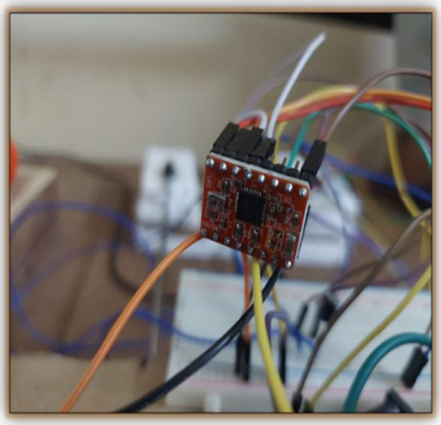
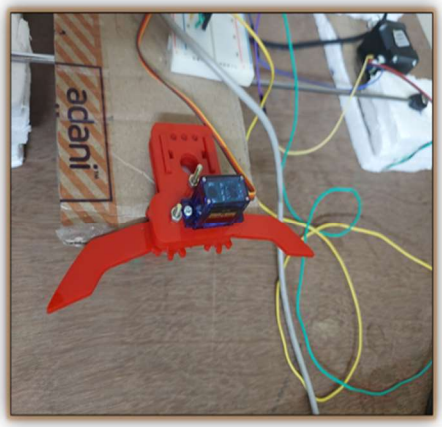
```
// Stores the red. green and blue colors
int redColor = 0;
int greenColor = 0;
int blueColor = 0;

// Stores frequency read by the photodiodes
int redFrequency = 0;
int greenFrequency = 0;
int blueFrequency = 0;

//Function for opening and closing of the gripper
void gripperclosing()
{
for (pos = 45; pos <= 120; pos += 1) {
// tell servo to go to position in variable 'pos'
servo_11.write(pos);
// wait 15 ms for servo to reach the position
delay(15); // Wait for 15 millisecond(s)
}
delay(3000);
}

void gripperopening()
{
for (pos = 120; pos >= 60; pos -= 1) {
// tell servo to go to position in variable 'pos'
servo_11.write(pos);
// wait 15 ms for servo to reach the position
delay(15); // Wait for 15 millisecond(s)
}

}

void readcolor()
{
// Setting RED (R) filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
// Reading the output frequency
redFrequency = pulseIn(sensorOut, LOW);
// Remaping the value of the RED (R) frequency from 0 to 255
redColor = map(redFrequency, 52, 103, 255,0);

// Setting GREEN (G) filtered photodiodes to be read
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);
// Reading the output frequency
greenFrequency = pulseIn(sensorOut, LOW);
// Remaping the value of the GREEN (G) frequency from 0 to 255
greenColor = map(greenFrequency, 78, 123, 255, 0);
```

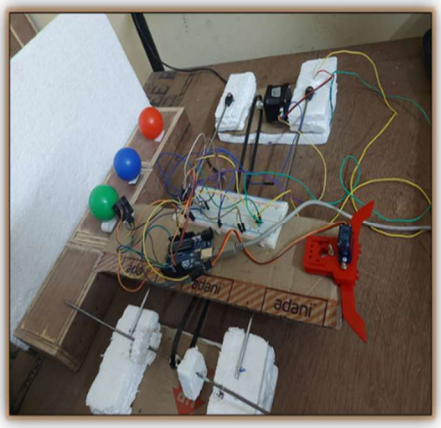
```
// Setting BLUE (B) filtered photodiodes to be read
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);
// Reading the output frequency
blueFrequency = pulseIn(sensorOut, LOW);
// Remaping the value of the BLUE (B) frequency from 0 to 255
blueColor = map(blueFrequency, 45, 106, 255, 0);
}

void checkcolor()
{

if(redColor > greenColor && redColor > blueColor && a==4)
{
Serial.println(" - Red detected!");
b = a+1;
dummyvar = dummyvar+1;
digitalWrite(enablepin,HIGH);  //Stopping the stepper motor
delay(3000);
gripperclosing();

}

if(greenColor > redColor && greenColor > blueColor && a==6)
{
Serial.println(" - GREEN detected!");
b=a+1;
dummyvar = dummyvar+1;
digitalWrite(enablepin,HIGH);   //Stopping the stepper motor
delay(3000);
gripperclosing();
}

if(blueColor > redColor && blueColor > greenColor && a==5)
{
Serial.println(" - BLUE detected!");
b=a+1;
dummyvar = dummyvar+1;
digitalWrite(enablepin,HIGH);   //Stopping the stepper motor
delay(3000);
gripperclosing();
}
}

void setup() {
// Setting the outputs of color sensor
pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);

// Setting Outputs for stepper motor driver
pinMode(stepPin, OUTPUT);
pinMode(dirPin, OUTPUT);
pinMode(enablepin ,OUTPUT);
```

```
// pinMode(input1, OUTPUT);
//pinMode(input2, OUTPUT);
servo_11.attach(11, 500, 2500);

// Setting the sensorOut as an input
pinMode(sensorOut, INPUT);

// Setting frequency scaling to 20%
digitalWrite(S0,HIGH);
digitalWrite(S1,LOW);

// Begins serial communication
Serial.begin(9600);

// Declare pins as Outputs for stepper
pinMode(stepPin, OUTPUT);
pinMode(dirPin, OUTPUT);
}

void loop()
{
//Taking input from the user in Serial Monitor
Serial.println("Enter the color");
while(Serial.available()==0)
{

}
while(Serial.available() >0)
{
c = Serial.readString();
}
b = 0 ;
a = c.length();
Serial.println(c);

digitalWrite(enablepin,LOW);  //Enabling the driver module
digitalWrite(dirPin, LOW);    // Stepper motor rotates in CW direction (viewed from front)
z = 0;

while(a>b)
{
// Stepper motor
for(int x = 0; x < stepsPerRevolution; x++)
{
digitalWrite(stepPin, HIGH);
delayMicroseconds(10000);
digitalWrite(stepPin, LOW);
delayMicroseconds(10000);
z = z+1;
}

readcolor();
checkcolor(); // Checks the current detected color and prints a message in the serial monitor

}
```

```
// Stepper Motor returning to home position
if(dummyvar==1)
{
digitalWrite(enablepin,LOW);  //Enabling the driver module
}

if( dummyvar ==1)
{
digitalWrite(dirPin, HIGH);    // Stepper motor rotates in CCW direction (viewed from front)

for(int x = 0; x < z-50; x++)
{
digitalWrite(stepPin, HIGH);
delayMicroseconds(5000);
digitalWrite(stepPin, LOW);
delayMicroseconds(5000);
}
delay(1000);
gripperopening();
}
dummyvar = 0;
}
```

- Physical Assembly:



- Demonstration video: (click on the hyperlink)

- # Outcomes of the project:
  - Acquainted with the Arduino and other electronic components.
  - Understood the working principles and controlling the functions of,
    - TCS 3200 colour detecting sensor
    - Stepper motor
    - Servo motor
  - Understood the essence of
    - Team work
    - Collaborative tasks
    - Planning and implementations of projects
    - Modelling and design using usual DIY components.
    - Making an alternative plan in case of unforeseen situations.