PROJECT REPORT ON

Color Sorter Machine

BY

1) ADHISH VELINGKAR

2) SAHIL RAJPURKAR

3) SHUBHAM KHANDHAR

4) BHAVIN PRAJAPATI

UNDER THE GUIDANCE OF

Mr. AMEY GAWDE

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION
K.J. SOMAIYA COLLEGE OF ENGINERING,
VIDYAVIHAR, MUMBAI-400077.
(Autonomous College Affiliated to University of Mumbai)
(2018-19)

K.J. SOMAIYA COLLEGE OF ENGINERING,
VIDYAVIHAR, MUMBAI -400077.

(Autonomous College Affiliated to University of Mumbai)

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the following students of semester VI

- ADHISH VELINGKAR - 1613129
- SAHIL RAJPURKAR - 1613130
- SHUBHAM KHANDHAR - 1723002
- BHAVIN PRAJAPATI - 1723003

have successfully completed the project titled "**Color Sorter Machine**" towards the partial fulfillment of degree of Bachelor of Technology in Electronics and Telecommunications of the University of Mumbai during academic year 2018-19.

Internal Guide

Examiner 1

Examiner 2

Abstract:

For sorting object in industry optical sorting is very much convenient. Color and size are the most important features for accurate classification and sorting of product which can be done by using some optical sensors or analyzing their pictures. The color sorting machine is mainly a device that can sense the different color of the object and assert them into different containers. When the object moves from one place to another with the rotation of motor, sensors as the input devices will send signal to microcontroller where microcontroller as the controller will give command to the actuator to do action.

This project describes a working prototype designed for automatic sorting of objects based on the color. OPT101 sensor was used to detect the color of the product and the STM microcontroller was used to control the overall process. The identification of the color is based on the ADC value of the output of OPT101 sensor. Two Servo motors were used, each controlled by the microcontroller. The first motor is used for carrying the product to be analyzed by the color sensor, and the second motor is used for sorting the product according to the reading taken by the color sensor, into the separate containers.

The experimental results promise that the prototype will fulfill the needs for higher production and precise quality in the field of automation.

Table of Contents

List of Figures

1. Introduction

1.1 Background

Nowadays, in the present state of intense competition, production efficiency is generally regarded as the key of success. Worker fatigue on assembly lines can result in reduced performance, and cause challenges in maintaining product quality. An employee who has been performing an inspection task over and over again may eventually fail to recognize the color of product. Automating many of the tasks in the industries may help to improve the efficiency of manufacturing system.

Production efficiency includes the speed at which production equipment and production line can be lowering material and labor cost of the product, improving quality and lowering rejects, minimizing downtime of production equipment and low-cost production equipment. Taking this matter under consideration the project is developed which is very useful for industries.

Main objectives of the study are studying various sorting processes, designing and assembling of servo motors, designing and fabrication of automatic color sorting system of product on the guide.

1.2 Motivation

Color sorters or color sorters (sometimes called optical sorters or digital sorters or electronic color sorters.) are machines that are used on the production lines in bulk food processing and other industries. They separate items by their colors, detecting the colors of things that pass before them, and using mechanical or pneumatic ejection devices to divert items whose colors do not fall within the acceptable range or which are desired to form a separate group from the rest.

The main advantages of the system are less time required to sort the product, as the whole system is performed by machine there is less possibility of mistake, less man power required. If the industry can produce the product within the required range then the demand of the product will be increased. So, the company will be benefited.

Since in order to reduce the human labor and avoid the negligence caused by them in case of hurry this automatic color sorter machine can easily can provide ease to the problem and sort the product accordingly thus giving a good efficiency and accurate results.

Hence, these were the reasons which motivated us to select this project.

1.3 Scope of the Project

Nowadays in highly competitive industrial manufacturing, the management of the integrity of supply of a product from raw material to finished product through quality manufacturing is of paramount importance. For the declaration of a product bearing high quality and dimensional accuracy is mandatory. So, this project of automatic color sorting is an excellent one because of its working principle and wide implementation. By applying the idea of this project an industry can easily sort the required product according to its color. Though it has some limitations, but by having done some modification this concept can be implemented in wide range of application.

1.4 Brief Description of the Project

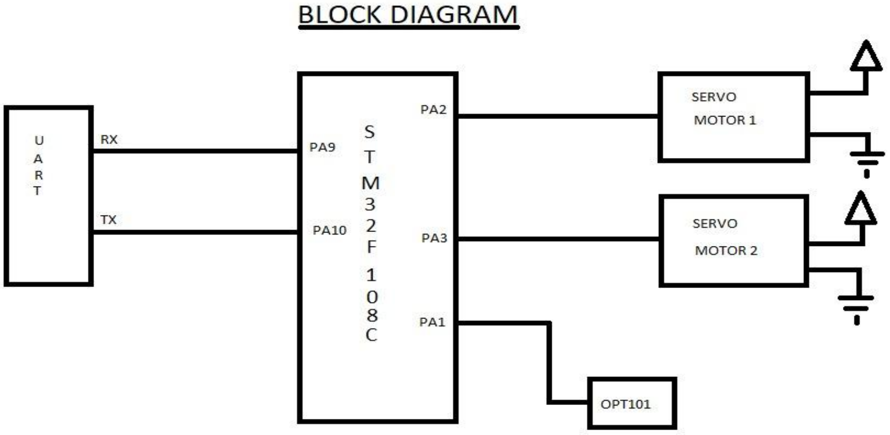

Figure 1: Circuit Block Diagram

Components used in the project are as follows.

- **<u>Servo Motor</u>**

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages.
Due to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of DC motor is converted into torque by Gears.

We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on required angle.

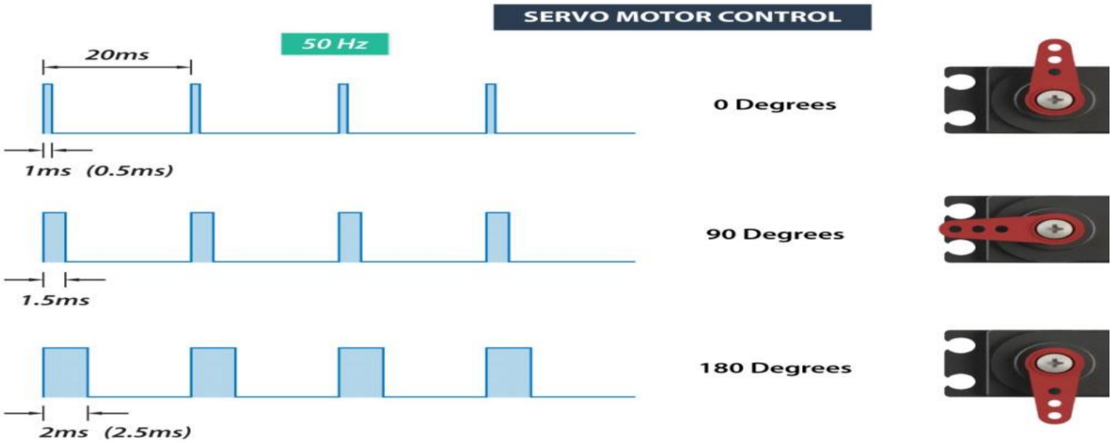

Figure 2: PWM principle of motor

Servo motor can be rotated from 0 to 180 degree, but it can go up to 210 degree, depending on the manufacturing. This degree of rotation can be controlled by applying the Electrical Pulse of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. Pulse of 1ms (1 millisecond) width can rotate servo to 0 degree, 1.5ms can rotate to 90 degree (neutral position) and 2ms pulse can rotate it to 180 degree.

All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume.

- **Color Sensor**

Here we have made a sensor which is a combination of OPT101 photodiode IC and piranha LED’s.

The OPT101 is a monolithic photodiode with on-chip transimpedance amplifier. The integrated combination of photodiode and transimpedance amplifier on a single chip eliminates the problems commonly encountered in discrete designs, such as leakage current errors, noise pick-up, and gain peaking as a result of stray capacitance. Output voltage increases linearly with light intensity. The amplifier is designed for single or dual power-supply operation.

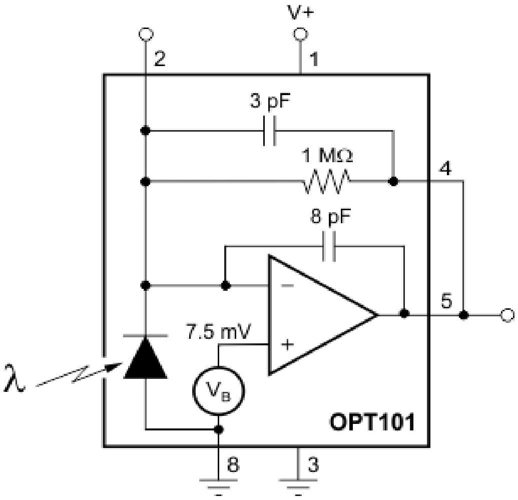

Figure 3: Circuit of OPT101

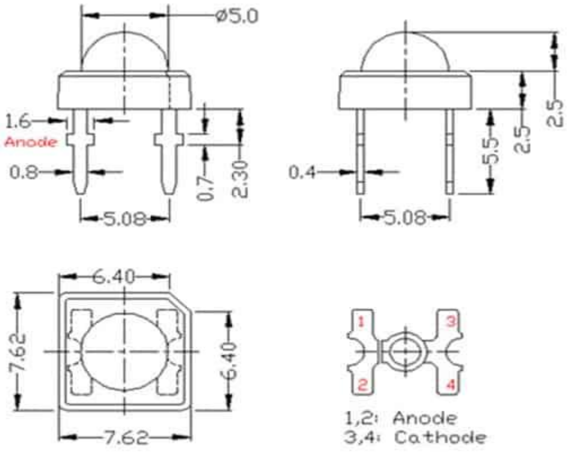

Figure 4: Schematic of Piranha LED

-STM32 CortexM-3

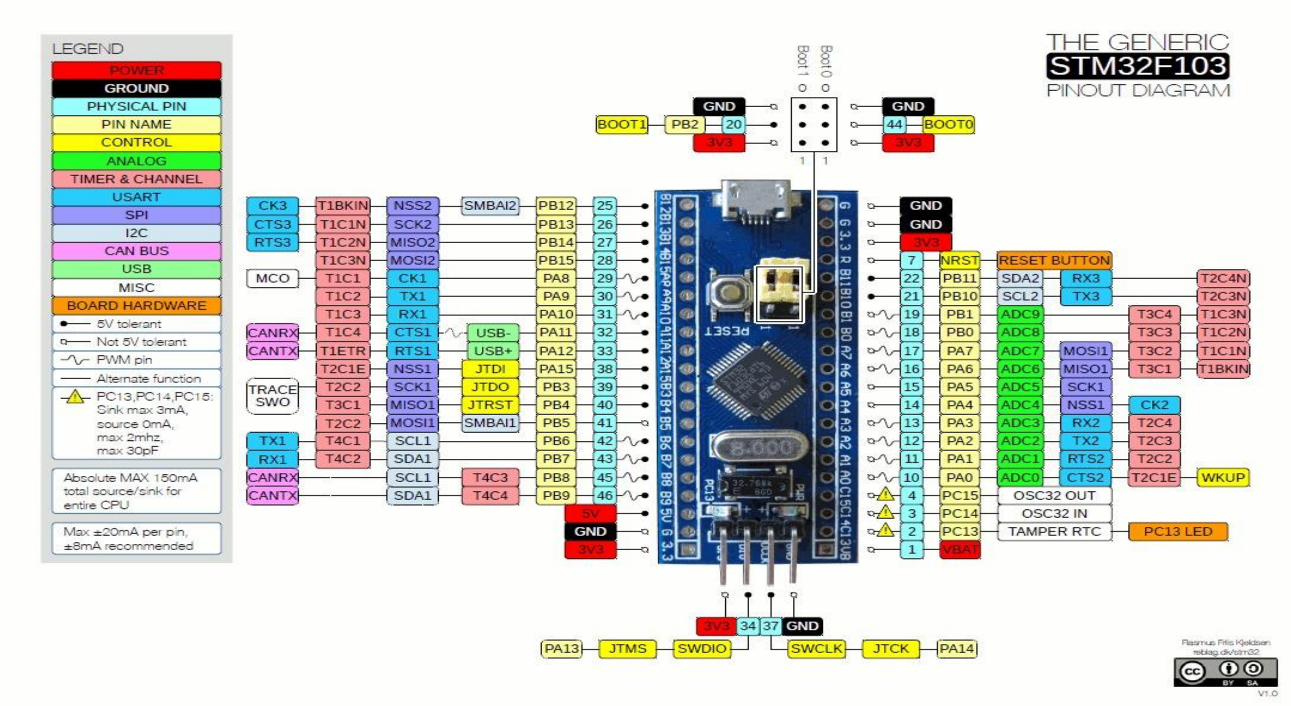

Figure 5: Pin Configuration of entire STM32F103XX

In order to control the motors and the sensor we used the STM32F103C8 of STM32F103xx family. Following are the features of the controller:

The STM32F103xx medium-density performance line family incorporates the high-performance ARM® Cortex®-M3 32-bit RISC core operating at a 72 MHz frequency, high-speed embedded memories (Flash memory up to 128 Kbytes and SRAM up to 20 Kbytes), and an extensive range of enhanced I/O’s and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, three general purpose 16-bit timers plus one PWM timer, as well as standard and advanced communication interfaces: up to two I2Cs and SPIs, three USARTs, an USB and a CAN. The devices operate from a 2.0 to 3.6 V power supply.

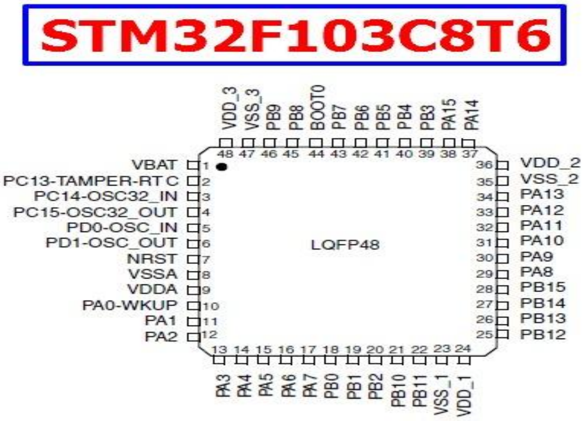

Figure 6: Pins of STM32F103C8

These features make the STM32F103xx medium-density performance line microcontroller family suitable for a wide range of applications such as motor drives, application control, medical and handheld equipment, PC and gaming peripherals, GPS platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems, video intercoms, and HVACs.

2. Working

2.1 Interfacing Diagram

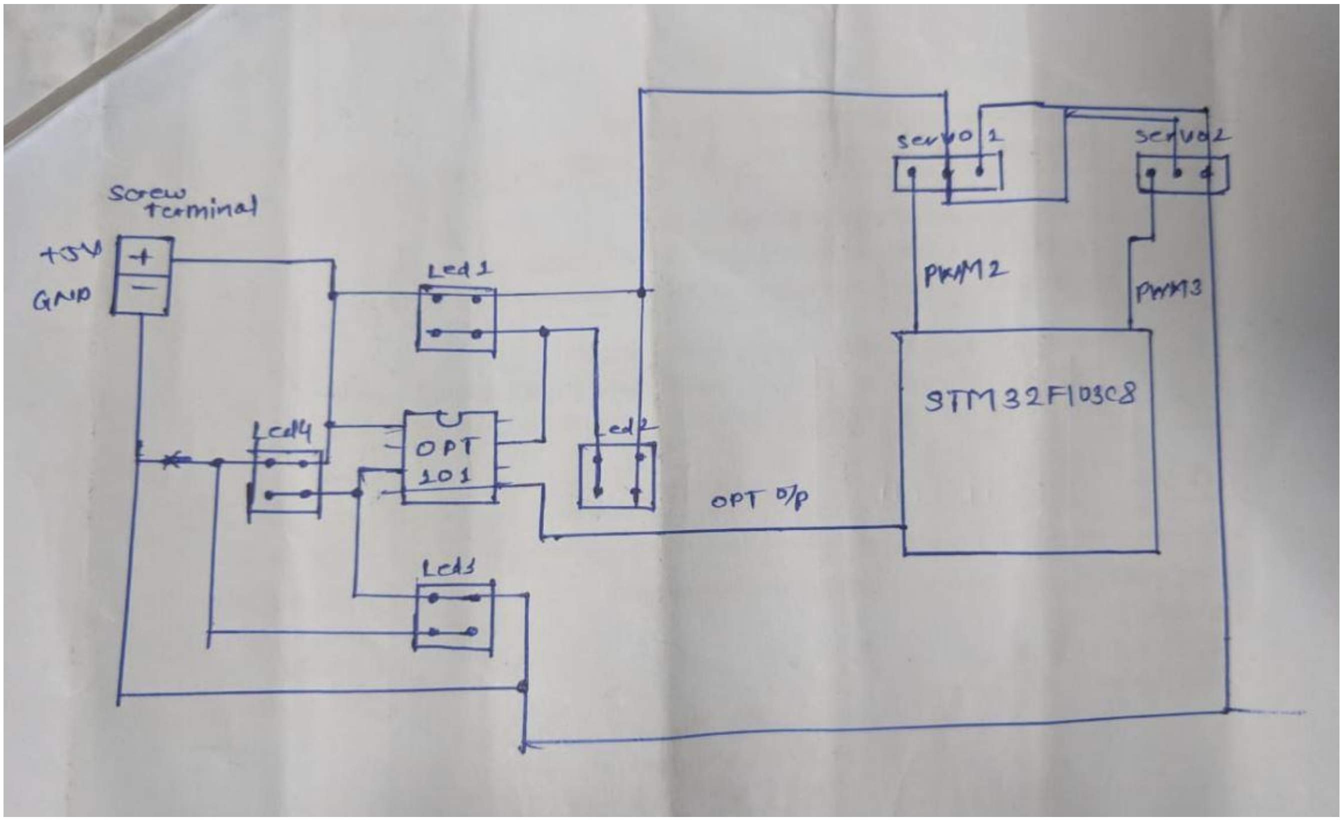

Figure 7: Interfacing of each component

- When a supply of 5V is given to the system the sensor gets the supply and it activates. And correspondingly the two servo motors are also activated in order to control the movement of the guide on which the product is placed. Firstly, through the inlet pipe the product (i.e. marbles in our case) is taken by the first servo motor which is specified with different angles mentioned in the program in order to rotate it and position it below the sensor and after detection of the color it is brought to the inlet pipe of second servo motor .

-When the light falls on the product it is reflected back to the color sensor. Because, color sensor OPT101 senses the color in the form of reading, which is opted by the select pin of STM32 i.e. PA1 in our case. The value of the color is in the form of 12-bit ADC. So, for different colors we have measured the range of values for which the product comes under it. Therefore, using the switch case and ‘if- else if’ loop the second stage of our system is selected by the program saved in the microcontroller.

- After the color detection, the second servo motor gets in contact with another guide, on which the containers are placed. The containers are been divided into three sections: 1st section for Green (Left side), 2nd section for White (Center), and 3rd section for Black (Right side). According to the color, the servo motor 2 will be rotated in forward or backward direction by the inclined guide. The product will finally fall to the corresponding sections in the container.

2.2 Flowchart

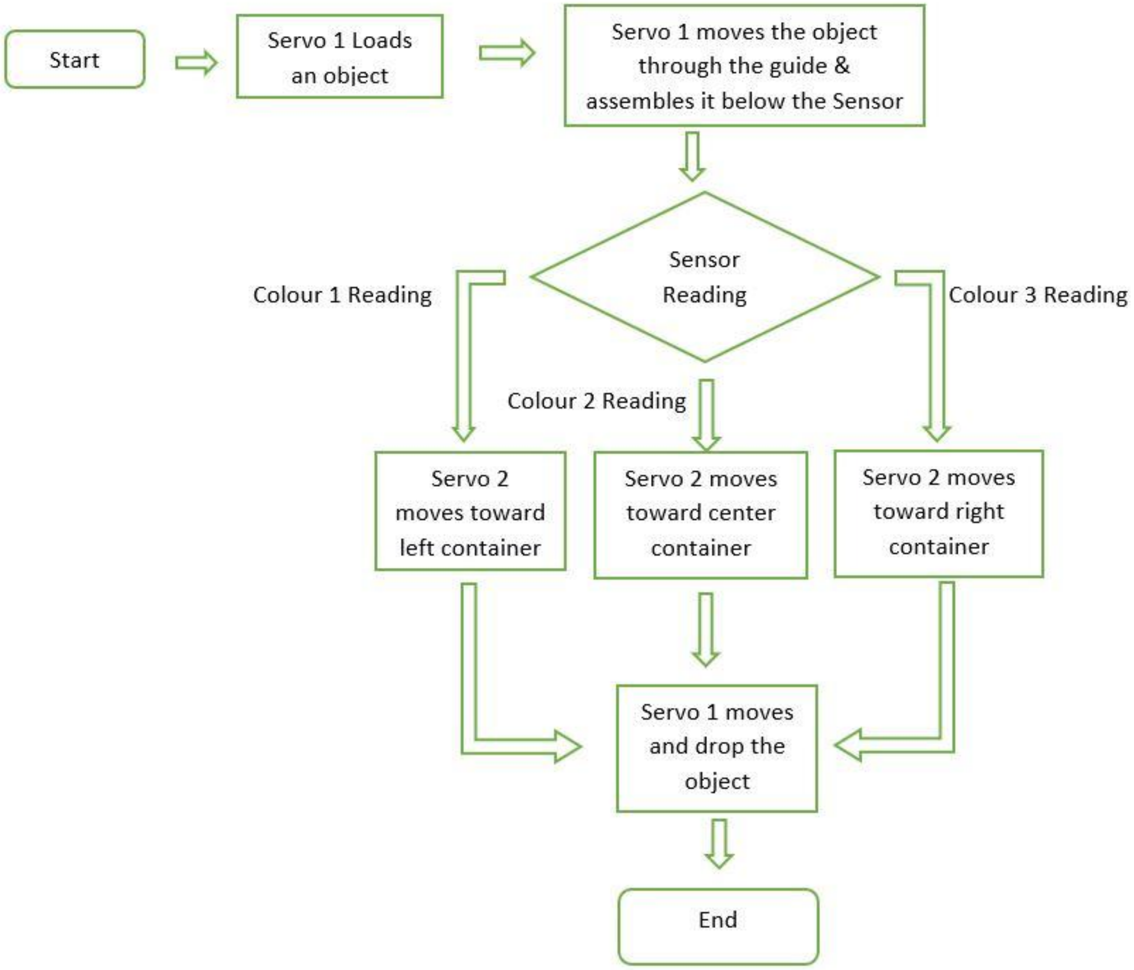
Start
Servo 1 Loads an object
Servo 1 moves the object through the guide & assembles it below the Sensor
Sensor Reading
Colour 1 Reading
Colour 2 Reading
Colour 3 Reading
Servo 2 moves toward left container
Servo 2 moves toward center container
Servo 2 moves toward right container
Servo 1 moves and drop the object
End

Figure 8: Data flow of sorting

2.3 Design and Construction

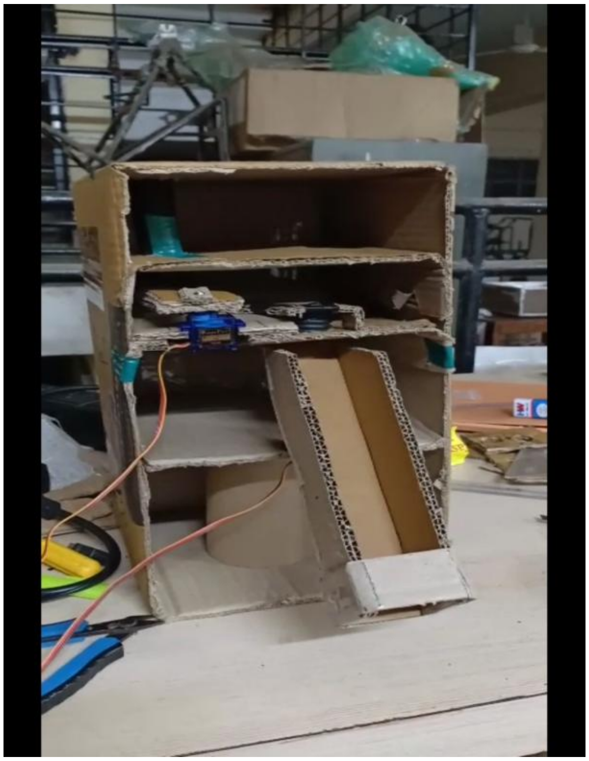

Figure 9: Testing of lower servo

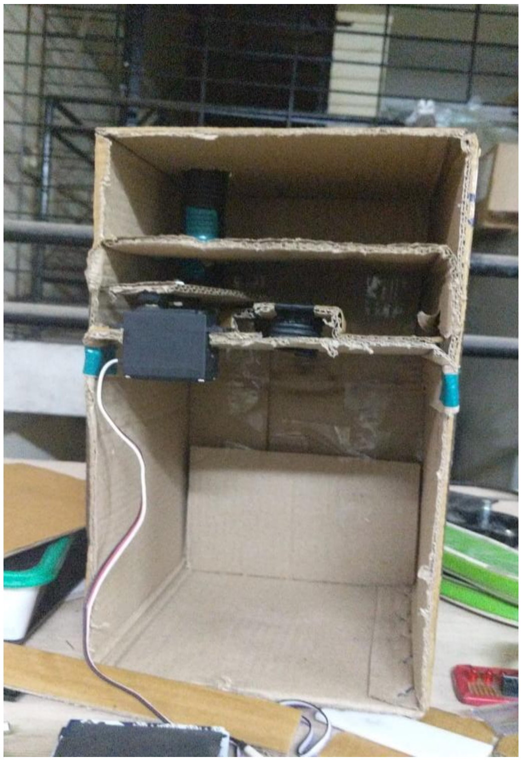

Figure 10: Testing of upper servo

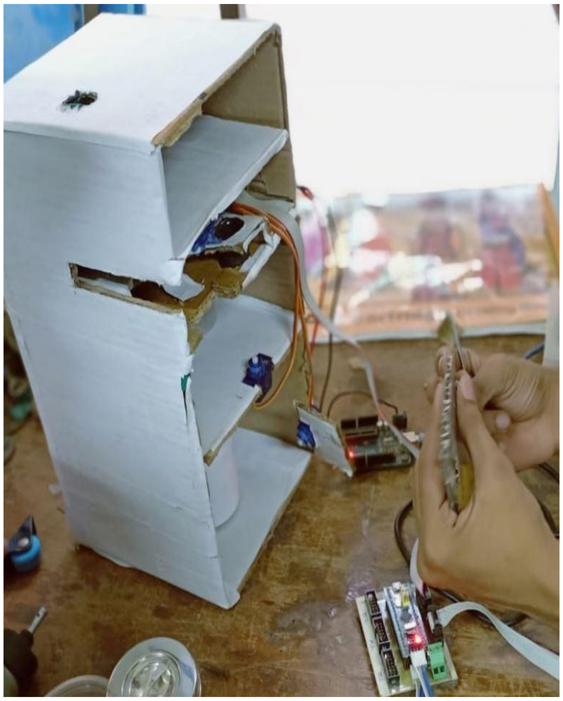

Figure 11: Assembling of guide, sensor and µC

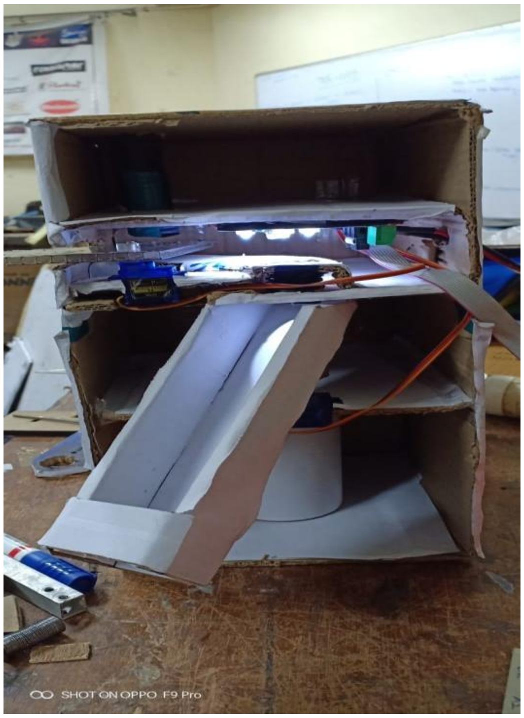

Figure 12: Final Design

3. Results and Discussions

The final result was quite satisfactory. The color detecting sensors worked well and it was able to detect different color object quite nicely and change the direction of servo on right and left side to sort the object in proper place.

3.1-Dimensional analysis

The prototype is designed for sorting objects of any shape but having fixed sizes of 1cm diameter. We can of course change this parameter by adjusting the frame of the color sensor. But one may note that it usually results in a change in the light ambience forcing us to do further frequency analysis of the sensor output for test colors.

The prototype will get more complicated as we increase the number of colors that have to be detected.

The placement of the object on the first servo guide is very crucial. It must be so placed that the center of the object and that of the sensor should be aligned with the same vertical plane, so that perfect detection takes place.

3.2 Time Cost

The object once placed on the first servo guide it takes less than half a second to reach the sensor. It takes another 2ms for the sensor to detect the color. An additional 0.6secs is required if to position the product into the correct compartment in the sorting container, which implies that an additional 0.6secs will be consumed to reposition the container back the normal position on the second. Of course, these time values are dependent on the speed of the servo motors used.

4. Conclusion

The color detecting sensors worked well and it was able to detect different color object quite nicely and change the direction of servo on right and left side to sort the object in proper place. The guide through second motor moved from starting point to the end point without conflicting with the walls. The system performed well as programmed and detected the object according to their color and sorted accordingly.

5. Future work

It is very useful in wide varieties of industries along with the help of PLC and SCADA, especially in the packaging section. Automatic sorting machine enhances efficiency, practicality, and safety of operators. It ensures remarkable processing capacity as well as peerless performance including color detection. Of course, we need to add high speed DC motors and sensors with appreciable response to speed up the system for industrial application.
The model can be improved by making some changes in the program and components. Some suggestions are given below.

- We can add a load cell for measurement and control of weight of the product.
- We can also add a counter for counting the number of products.
- Speed of the system can be increased accounting to the speed of production.
- The system can be used as a quality controller by adding more sensors.
- The sensor can be changed according to the type of product.
- The servo motor can be replaced with stepper motor.
- The STM can be replaced with PLC.

References:

1) https://en.wikipedia.org/wiki/Colour_sorter
2) https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino
3) http://www.ti.com/product/OPT101
4) image.dfrobot.com/image/data/DFR0269/STM32F103C8T6.pdf
5) https://info.pcboard.ca/led-specifications/superflux-piranha-led-specifications/

APPENDIX A: Code

```
#include <stm32f1xx.h>
#define  Servo1MinAngle  600            // 0
#define  Servo1MaxAngle  3000 // 180
#define  Servo2MinAngle  500            // 0
#define  Servo2MaxAngle  3000 // 180

void port_init();
void timer2_init();
void servo1_angle(uint32_t angle);
void servo2_angle(uint32_t angle);
void _delay_ms(int time);
void init_ADC();
uint16_t ADC_read(int channel_num);

int s_mode=0;
unsigned int data = 0,i =0,j = 0;
uint16_t value = 0;

int main(void)
{
port_init();
timer2_init();
init_ADC();

while(1)
{
switch(s_mode)
{
case 0:
servo1_angle(180);
_delay_ms(2000);
servo1_angle(97);
_delay_ms(1000);
s_mode++;
break;
case 1:
value=ADC_read(3);
if(value>3400 && value<3900)
{
s_mode=2;
}
else if(value>3950)
{
s_mode=3;
}
else if (value>3100 && value<3400)
{
s_mode=4;
}
else
{
s_mode=5;
}
break;
case 2:
servo2_angle(0);
_delay_ms(2000);
servo1_angle(85);
_delay_ms(1000);
servo1_angle(90);
_delay_ms(1000);
break;
case 3:
servo2_angle(60);
_delay_ms(2000);
servo1_angle(85);
_delay_ms(1000);
servo1_angle(90);
_delay_ms(1000);
break;
case 4:
servo2_angle(90);
_delay_ms(2000);
servo1_angle(85);
_delay_ms(1000);
servo1_angle(90);
_delay_ms(1000);
break;
}}}

void port_init()
{
RCC->CR|=(1<<0);                  //HSE
selected as system clock
RCC->CFGR = (1<<15)|(1<<14);      // adc
prescalar by 6 , 72/6 = 12

RCC->APB2ENR|=(1<<2)|(1<<9);
RCC->APB1ENR|=(1<<0)
GPIOA-
>CRL&=~(1<<12)&(1<<13)&(1<<14)&(1
<<15);}

void timer2_init()
{
```

```
TIM2->CCER|= (1<<8);
//Capture/Compare 1 output enable
TIM2->CCER|= (1<<4);
//Capture/Compare 2 output enable
TIM2->CR1 |= (1<<0)|(1<<7);
//Counter enabled. TIMx_ARR register is
buffered
TIM2->ARR = 8399;                //Auto-
reload value
TIM2->CR2 |= (1<<10)|(1<<11);
TIM2->CR2 |= (1<<12)|(1<<13);
TIM2->PSC = 10;                  //Prescaler
value
TIM2->CCMR2|=
(1<<7)|(1<<6)|(1<<5)|(1<<3)|(1<<2);  //
TIM2->CCMR1|=
(1<<15)|(1<<14)|(1<<13)|(1<<11)|(1<<10);
//
TIM2->BDTR|= (1<<15);            //Main
output enable
}

void servo1_angle(uint32_t angle)
{
GPIOA->CRL=(1<<8)|(1<<9)|(1<<11);
//Output mode, max speed 50
MHz.Alternate function output Push-pull

uint32_t step = ((Servo1MaxAngle -
Servo1MinAngle)/180)*angle;

if(step == 0)
step = Servo1MinAngle;

if(step > Servo1MaxAngle)
step = Servo1MaxAngle;

TIM2->CCR3=step;
}

void servo2_angle(uint32_t angle)
{
GPIOA->CRL=(1<<4)|(1<<5)|(1<<7);
//Output mode, max speed 50
MHz.Alternate function output Push-pull
```

```
uint32_t step = ((Servo2MaxAngle -
Servo2MinAngle)/180)*angle;

if(step == 0)
step = Servo2MinAngle;

if(step > Servo2MaxAngle)
step = Servo2MaxAngle;

TIM2->CCR2=step;
}


void init_ADC()
{
ADC1->CR2 =
(1<<0)|(1<<2)|(1<<22)|(1<<17);//|(1<<1);//|(
1<<20)|(1<<17) ; // ADC ON | start of
conversion of regular channel
ADC1->SQR1 = 0x00100000;                //
Total number of channels
ADC1->SMPR2 = (1<<0)|(1<<1)|(1<<2);
}

uint16_t ADC_read(int channel_num)
{
ADC1->SQR3 = channel_num;
while((ADC1->CR2 & 0x00000004)!=0);
ADC1->CR2 |= (1<<0);
while((ADC1->SR&0x00000002)==0) ;  //
wait for end of conversion bit to be 1
data = ADC1->DR;                // data stored
in adc1->dr is stored in data(unsigned int)
ADC1->SR = 0x00000000;            //
clearing eoc and start for regular channel
ADC1->CR2 |= (1<<22);
return data;
}

void _delay_ms(int time)
{
for(int i=0;i<time;i++)
{
for(int j=0;j<1000;j++);
}
```