Journal of Engineering Science and Technology
EURECA 2014 Special Issue January (2015) 71 - 81

DESIGN AND DEVELOPMENT OF COLOUR SORTING ROBOT

LIM JIE SHEN*, IRDA HASSAN

School of Engineering, Taylor's University, Taylor's Lakeside Campus,
No. 1 Jalan Taylor's, 47500, Subang Jaya, Selangor DE, Malaysia
*Corresponding Author: jefflim88@gmail.com

Abstract

This paper shows a new approach for continuous recognition and sorting of objects into desired location. Image or colours processing nowadays attract massive attention as it leads to possibility of widening scope of application in different field with the help of modern technology. A colour sorting robot is researched, designed and created with Arduino Uno microcontroller, TCS3200D Colour Sensor, SG90 Tower Pro Servo Motor and other electronic components. The system has the ability to sort the object according to their colours into respective colour station in minimum time. Specific programming code for this system is written.

Keywords: Colour recognition, Colour sorting, Programming, Robot, Microcontroller.

1. Introduction

The ability to differentiate colours is essential for human's life as it gives us the awareness about the changes in surrounding through our vision [1]. Moreover, by exploiting the ability of colour capture, intelligent machine gains the function to differentiate, sort and organise. The project includes sensors that detect colour of the object then sends the information to Arduino Uno which in turn adjusts the servo motor which located just below the ball slider to move it left and right [2]. Based upon the colour detected, the slider will move according to the angle of 10°, 70° and 170° depends on the object colour. The stations are in red, green and blue respectively. After every ball placement, the slide will go back to its default angle position, awaiting the next colour ball.

Throughout the years, many people tried to use various ways to programme and create intelligence robot in various ways to have respective function or achieving goals. Some of the claims made have contributed directly or indirectly to the researched done in this paper. Yu et al. [3], found that a new way in construction turned out to be the upcoming technologies that can be used to accommodate private indoor robots that move with owners and lend them a helping hand when needed. In many cases, the interactions of a few robots are needed to carry out certain task, whereby it is crucial to fully control robots to operate at level which is desired [4]. This colour sorting robot is developed with the purpose of minimising the cost, optimising the productivity and reducing human mistakes.

The objectives of this project are summarised as below:

- To write colour recognising and colour sorting code.
- To integrate colour recognising, sorting and motor with microcontroller.
- To test the code and troubleshoot with the efficiency of the system.
- To sort objects according to their respective colours and stations accordingly.

2. Approach and Methodology

2.1. The significance of the work

The challenge often occurred in industry that involves colour sorting system such as diamonds industry or food industry such as separation of rice, coffee, and other cereals that could be solved via this project. The first factor is speed of sorting. The speed of colour sorting process by an operator is very slow. This is due to the limitation of response time for a human eye [5]. The eyes will always take some time to see an image and project this to the brain to initiate visual sensation. After the brain has received the image, it will take some time for the brain to determine the colour of the object too [6]. However, this limitation can be covered by using a computer. In this project, Arduino Uno, the microcontroller is used to increase the speed of colour sorting.

The other challenge is the accuracy of colour sorting process by an operator is very low in current industry. This is because an operator will need to handle hundreds or thousands of objects each day, tiredness will usually cause some fault when colour is being divided or sorted. It is very common for an operator to make this error. However, a machine will not have this problem. A machine will give accurate result even after it has repeated a process for billions of times unless fault to the system occurs. In this project, servo motor is used to substitute the operator and thus increase the accuracy of colour sorting. The implementation cost for a colour sorting process using the operator is very high. If an industry needs to sort a bulk quantity of product, many workers are needed and with the implement shift and overtime system, the total sum of cost will be relatively high.

2.2. Methods

Colours are light waves that are no different than the electromagnetic waves emitted by cell phones. It is our brain cells that interpret them as real colours [7]. For robot, the simplest way they can use to detect colours is by uses the filters of three main colours, which are red, green and blue and compared the value on the light reflected on it. The value taken will then send signal to Arduino which analyse which colour is the object. The colour sorting robot code is programmed using Arduino software. Programming code is researched and written in order for the colour sorting robot to carry out recognition and sorting mechanisms. The connection is done by connecting wires to connect up Arduino Uno which act as microcontroller, servo as well as colour sensor. The servo motor then slides the ball left and right at different angle to different location. The hardware and software flowchart of the whole project are shown in Figs. 1 and 2 respectively. The hardware is consists of colour sensing connection, colour recognising connection and robot body. The software is done by using Arduino UNO.

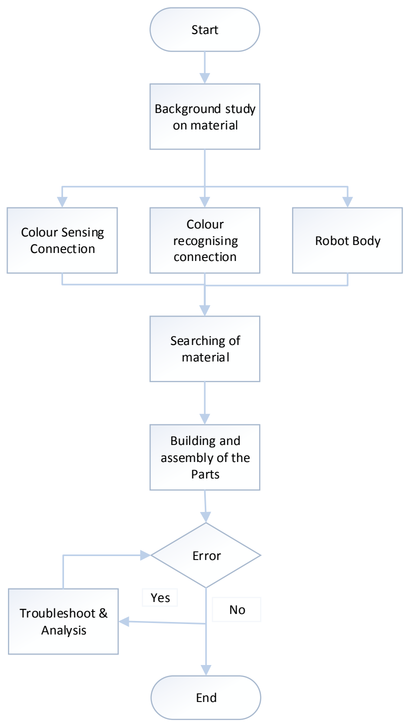

Fig. 1. Hardware flowchart.

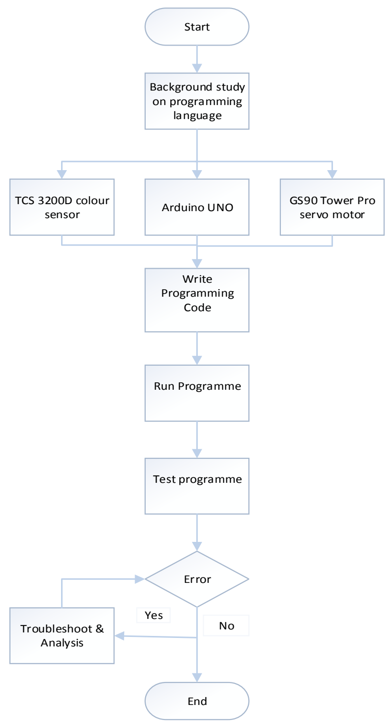

Fig. 2. Software flowchart.

2.3. Project development

For sorting mechanisms to work, an Arduino programme is written to carry out the colour sensing and recognising task correctly (see Appendix A). The programme has to be tested and simulation has to be carried out. After the design has been confirmed, the materials can be purchased. There are various ways of doing simulation which by using software in computer and simulation on a breadboard. The breadboard simulation has to be done with real components. It is important to ensure there are no faulty on the components and the board after the circuit has been powered up with voltage and current. If the programme is not functioning during the test or simulation, then troubleshooting process is essential. The next step is to analyse the programme and understand the code function. This is an important process until the best code has been simulated. The balls are tested on the circuits which are connected on a breadboard to make sure the code is right before all the parts are assembled to become a robot. For the hardware, a suitable plastic material is used as the body of the robot and compartment to place all the materials. After placing together both software and hardware, the robot is basically formed. The RGB (Red, Green, Blue) values taken from the sensor will then send signal to Arduino Uno to process. The connection is done by connecting all wires to connect up Arduino Uno which act as Arduino Uno, batteries, servo motor as well as the colour sensor. The sorting system is monitored and checked if there is any error during the testing. Once the combination between the hardware and software have some error of fault, either one or both of the parts will need to be modified. The troubleshooting process will be repeated until the objectives of this project are achieved. If the errors are solved, the project comes to an end which is the closure stage. Finally, the design and development of colour sorting robot is completed as shown in Fig. 3.

Fig. 1. Final product after merging software and hardware parts.

3. Data Validation and Verification

This section discusses on the validation and verification methods of this project. The method for a colour sensor to function is by using its green, blue and red filters to detect a common colour and compare which filter has the highest value.

3.1. Sorting mechanisms

The sorting mechanisms and tests are evaluated and the data are recorded. The overview of the mechanism is shown in Fig. 4. To ensure the validity and accuracy of the calculations, the RGB values are verified using colour software available in the market. Figure 5 shows the colour sorting process from green to blue ball during experiment using the final product.

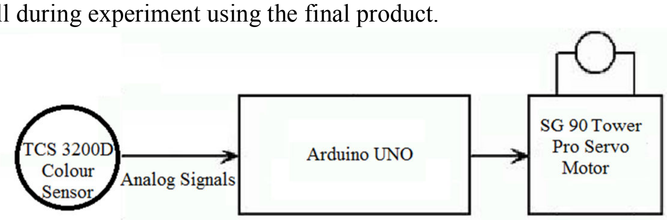

Fig. 4. System overview.

Fig. 5. Sorting process.

3.2. Validation and verification using software comparison

The method for a colour sensor to recognise colour of an object is by using the green, blue and red filters to detect a common colour and see which colour filter has the highest value recorded. If the ball is blue, the blue (B) filter has the highest value compared to others; red and green filter as shown in the first 4 lines of the monitoring lines. The RGB values can be checked using many design or paints software such as Adobe Photoshop. The RGB values shown are pictured in Fig. 6 when tested with blue and in Fig. 7 when tested with red.

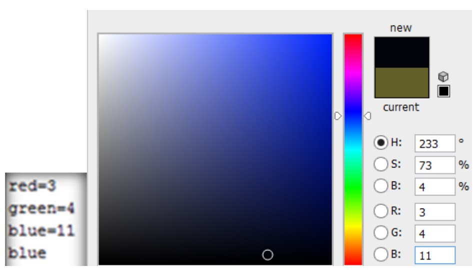

Fig. 6. RGB values of blue ball from monitoring screen verified using Adobe Photoshop.

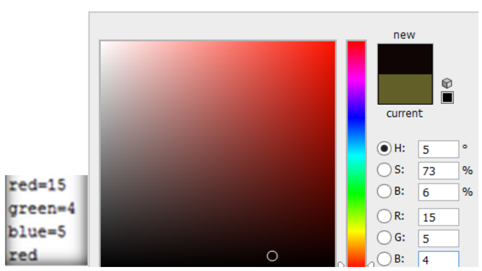

Fig. 7. RGB values of red ball from monitoring screen verified using Adobe Photoshop.

4. Results and Discussion

The sorting mechanisms and tests are presented in the following sections. In the discussion, the result is analysed and compared. The test carried out includes light intensity test and colour ball selection test.

4.1. Light intensity test

The tests are carried out indoor and outdoor to record the results and the values showed that the RGB values taken indoor are more or less higher than the one taken outdoor. This is due to the present of more than single light out door which affected the results. Tests taken indoor and outdoor are shown in Tables 1 and 2 respectively.

Table 1. RGB filter during indoor test.

Ball Colour	Average Filter Value		
	Red filter (R)	Blue filter (B)	Green Filter (G)
Red	17	6	5
Blue	3	11	6
Green	3	3	7

Table 2. RGB filter during outdoor test.

Ball Colour	Average Filter Value		
	Red filter (R)	Blue filter (B)	Green Filter (G)
Red	18	16	9
Blue	7	12	10
Green	2	5	6

It can be seen from Tables 1 and 2 that values of the filter is different when tested outdoor and indoor. The presence of other light has caused the TCS3200D colour sensor filters to record RGB (Red, Green, Blue) value that are closer to each other. For example, the most obvious results is when tested with green ball outdoor, the green filter and blue filter showed very close G and B value. This might cause the system to detect it as an error and not carried out sorting task as planned when two filters share the same reading. This means the system fails to detect if it is blue ball or green ball. It can be verified putting both red ball and blue lights together. The mixture resulting from the interaction of a coloured object with a coloured light source is an example of subtractive mixing. Figure 8 shows that the mixture section of both lights eventually becomes pink. The pink colour is getting more and more obvious when the white light is added. The same things happen in this outdoor experiment. The colour sensor will detect and record inconsistent RGB value which has high R and B values.

This scenario can be concluded that surrounding light is actually affecting the result and should be avoided. In the situation where more than one light is present, it is advisable to just focus on single light surface. RGB result will be more reliable when tested indoor or in a confined space. Besides preventing the presence of too many unwanted light, in order to get optimum result, there are another factor needed to be considered throughout the experiment which is listed below. It is the best to choose the objects with high concentration of colour, so the surrounding ray will not affected the result too much. For example, dark green, dark blue and dark red is chosen instead of light version of them. This is to increase the accuracy of the RGB values of an object and thus increase the efficiency of the recognising and sorting process by this system.

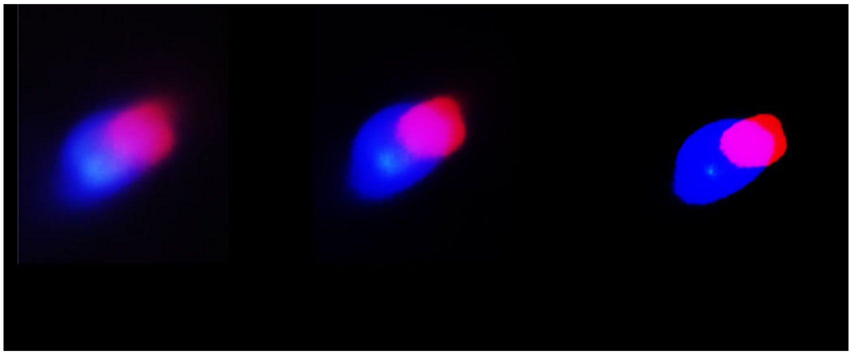

Fig. 8. Build-up of pink light.

4.2. Colour selection test

The test is carried out to determine which colour of the ball can be best detected by colour sensor TCS3200D. The variety colours of the balls used are blue, red, green, yellow, purple as well as orange. It can be seen from Table 3 that either two or all three of red, green and blue filters showed that the result is not too much differ from one another when the colours tested with secondary or tertiary colour. The secondary colours such as yellow, purple and orange cannot be detected as easily as colours such as red, green and blue.

Table 3. RGB filters with different colour balls.

Ball Colour	Average Filter Value		
	Red Filter(R)	Blue Filter (B)	Green Filter (G)
Red	17	6	4
Blue	3	12	7
Green	3	4	7
Yellow	15	6	14
Purple	13	15	6
Orange	14	6	12

For example, the orange ball is not suitable to be chosen as a tested object as it has R filter and G filter values that are too close to each other. G filter shows value of 12, R filter shows value of 14. If the light in the surrounding change, R filter might fall to 13 and G filter might increase to 13, this has a tendency for the robot to recognise it as error. The same goes for other secondary and complementary colours as well. It is too risky to make those colours as test object. This can be proved by the transparent RGB cube. From the cube, it can be seen that all red, green and blue are furthest away from each other and are only joint perpendicularly at the points where all three values are zero. Transparent RGB is illustrated in Fig. 9.

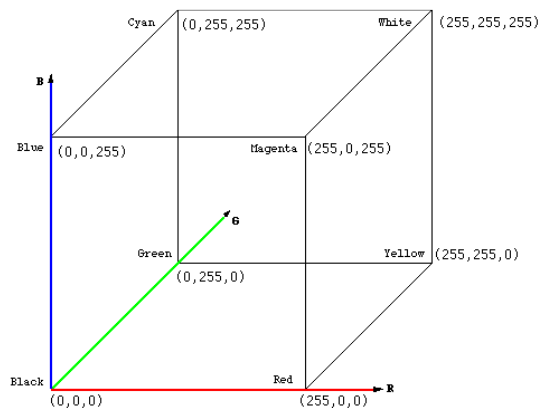

Fig. 9. Transparent RGB cube.

Almost every colour sensor recognise colour by using its green, blue and red filters. In the situation when yellow light is break down, it can become red light and green light which are mixed together. It can be observed by human but not the colour sensor. For green, blue and red colour, they are primary and cannot be breakdown even further. Or in other word, there is no other colours that can form them. Hence, it justified the reason these three primary colours filters are the used in the colour sensor nowadays and also act as optimum colour of to be chosen as testing object.

5. Conclusions

In order to create a smart robot that can recognised colour ball and placed them at the correct location, research in length is needed. There are existing robot using PIC and other microcontroller but there are not many created using Arduino Programme. The real contribution of this system is that it is able to save time to sort the colour hence making this Arduino-powered colour recognising and sorting robot more efficient than the existing sorting system. Upon finishing of this project, a robot that has capability to recognise colour of the ball and sorts them according to their colour is successfully created. In conclusion, all the objectives and scope are achieved. This project manages to finish in time and stay within budget allocated. The robot system has a huge potential to go into the market with proper implementation. It will be very useful in sorting industry that includes cereal sorting, marble sorting, paints, toys and many others.

Similar with almost every project, there is always a room for future improvement. In order to upgrade the capability of this system to the next level, here are some recommendations:

- There are ways to make the robot recognised more colours than now. This can be enhanced by programming more RGB values comparison code to allow the system to sort more colour such as yellow, purple, orange and other colour.
- Another area of improvement can be made is to allow the robot to move freely in all direction. This can be carried out with the additional motion sensor that allow robot to detect object and move freely to place the ball into the correct station by avoiding the obstacles and to take an object from different place and place it at another place.
- Another upgrade is allowing people to read the values of RGB of an object easily without the need to connect the robot to PC and check from Arduino Serial Monitoring Screen. An LCD Display can be added so the robot actually showing the RGB values whenever it detect an object in the robot body itself. This will be useful for some people.
- The other improvement that can be made is allowing the robot to carry heavier object. The can be done by increasing the strength of the robot itself. Using a strong material instead of normal plastic will be a good way as an upgrade.

References

1. Khojastehnazhand, M.; Omid, M.; and Tabatabaeefar, A. (2010). Development of a lemon sorting system based on colour and size. *Journal of Plant Science*, 4(4), 122-127.
2. Avishay, D.; Pavlov, V; and Avramov, I. (2011). Designing and testing a calibrating procedure for combining the coordination systems of a handling robot and a stationed video camera. *Robotics and Computer-Integrated Manufacturing*, 27(3), 514-520.
3. Yu, Y-H.; Kwok, N.M.; and Ha, Q.P. (2011). Color tracking for multiple robot control using a system-on-programmable-chip. *Automation in Construction*, 20(6), 669-676.
4. Do, Y. (2012). Intelligent worm sorting using robot vision. *Procedia Engineering*, 41(2), 917-922.
5. Unay, D.; Gosselin, B.; Kleynen, O.; Leemans, V.; Destain, M.; and Debeir, O. (2011). Automatic grading of Bi-colored apples by multispectral machine vision. *Computers and Electronics in Agriculture*, 75(1), 204-212.
6. Mehl, P.M.; Chen, Y.R.; Kim, M.S.; and Chan, D.E. (2004). Development of hyperspectral imaging technique for the detection of apple surface defects and contaminations. *Journal of Food Engineering*, 61(1), 67-81.
7. Dawkins, R. (2006). *Unweaving the Rainbow: Science, Delusion and the Appetite for Wonder* (1st ed.). United Kingdom: Penguin UK, Kensington.

Appendix A

The programme is then transfer from PC to Microcontroller Arduino Uno.

```
#include <Servo.h>
int s0=3,s1=4,s2=5,s3=6;
int flag=0;
int counter=0;
int countR=0,countG=0,countB=0;
Servo servo1;
void setup()
{  Serial.begin(9600);
  pinMode(s0,OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
  pinMode(s3,OUTPUT);
  servo1.attach(12);}
void TCS()
{  digitalWrite(s1,HIGH);
  digitalWrite(s0,LOW);
  flag=0;
  attachInterrupt(0, ISR_INTO, CHANGE);
  timer2_init();}
void ISR_INTO()
{  counter++;
  TCNT2=100;
  flag++;
  if(flag==1)
  {    counter=0;  }
  else if(flag==2)
  {    digitalWrite(s2,LOW);
    digitalWrite(s3,LOW);
    countR=counter/1.051;
    Serial.print("red=");
    Serial.println(countR,DEC);
    digitalWrite(s2,HIGH);
    digitalWrite(s3,HIGH);     }
  else if(flag==3)
  {
    countG=counter/1.0157;
```

```
  Serial.print("green=");
  Serial.println(countG,DEC);
  digitalWrite(s2,LOW);
  digitalWrite(s3,HIGH);
 }
 else if(flag==4)
   Serial.print("red");
   Serial.print("\n");
   servo1.write(10);
   delay(10);
   servo1.write(90) \\servo back to original position
  }
  else if((countG>countR)&&(countG>=countB))
  {
   Serial.print("green");
   Serial.print("\n");
   servo1.write(140);
   delay(10);
   servo1.write(90) \\servo back to original position
  }
  else if((countB>countG)&&(countB>countR))
  {
   Serial.print("blue");
   Serial.print("\n");
   servo1.write(170);
   delay(10);
   servo1.write(90)\\ servo back to original position
  }
 }
 else
 {
  delay(1000);
 }
}
```