Intelligent color sensing robotic arm for industrial automation

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of
Bachelor in Computer Science & Engineering

by

Sumaiya Ahmed
CSE 05306698
&
Md. Johirul Islam
CSE 05306695

Supervised by: Mohammad Manzurul Islam
Assistant Professor

Department of Computer Science and Engineering
STAMFORD UNIVERSITY BANGLADESH

May 2017

Abstract

This paper presents an application to sort colored objects with a robotic arm. We have a robotic arm which picks different colored cubes and sorts them placing in different location . The detection of the particular colour is done by a light intensity to frequency converter method. The robotic arm is controlled by a microcontroller based system which controls servo motors. With the microcontroller as a controller for the system, a manually feed object which is a coloring cubes will be determined by the robot to take and eject them to their exact location or station. There are four stations in the Color Sorter system that we made. Each station had their own range of color. The ball will be inserting manually by the user. Then the RGB sensor will detect what color is the object, after the detection been done, the decision will be made and the claw will take the object to the station that been recognized for it.

Approval

The Project Report Intelligent color sensing robotic arm for industrial automation submitted by Sumaiya Ahmed ID: CSE 05306698, MD. Johirul Islam ID: CSE 05306695, to the Department of Computer Science & Engineering, Stamford University Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science (Hons) in Computer Science & Engineering and approved as to its style and contents.

Board of Examiner's Name, Signature, and Date:

....	....	....
Dr. M Shaharia Bhuyan	**Nazia Hossain**	**Adnan Ferdous Ashrafi**
Assistant Professor	Senior Lecturer	Lecturer
Department of CSE	Department of CSE	Department of CSE
Date:	Date:	Date:

Supervisor's Signature and Date:

....

Mohammad Manzurul Islam
Assistant Professor
Department of CSE
Date:

Declaration

We, hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Mohammad Manzurul Islam , Assistant Professor, Department of Computer Science & Engineering, Stamford University Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Signature and Date:

....	....
Sumaiya Ahmed	**MD. Johirul Islam**
Date:	Date:

Acknowledgments

First and foremost, we are grateful to Allah, the Almighty, the Merciful without whose blessing, this project would not have been successful. He gave us confidence, courage and determination to overcome the obstacles we faced during this journey. We would like to express our eternal gratitude to Mohammad Manzurul Islam, Assistant Professor, our respected supervisor, for taking the time from his hectic schedule to guide us in this project. He is the one who inspired us to take this project and supported us every step of the way in every possible way. Without him, this project would not have come to fruition. He provided with valuable technical advice and pointed us in the right directions which were much required for a project like this. We are truly grateful for having a teacher like him as our supervisor. We would also like to thank our parents and our best friend for pushing and encouraging us to do the best in our ability.

Table of Contents

List of Figures

List of Tables

1 Introduction

A robot is a mechanical artificial agent.It can carry a sense that it has goal of its own. Another common characteristic is that It is usually an electro-mechanical machine which is guided by computer or electronic programming, and is able to do tasks on its own. Although the appearance and capabilities of robot vary vastly, all robots share the feature of a mechanical movable structure under some form of control. Industrial Automation is the replacement with computers and machines to that of human thinking. It is the use of information technologies and also control systems for handling different processes and machineries in an industry to replace a human being.

1.1 Objective

The proposed system is a fully autonomous system which will increase the speed of product sorting procedure, decrease the cost of sorting process and optimize the productivity of an industrial object . This control of robot involves three distinct phase- perception, processing and action. In common the preceptors are sensors mounted on the robot, processing is done by micro controller and task is performed using servo motors.

This robot is used for pick the object from one place and place that objects in required boxes with respect to its color. Some industrial works are harmful for humans this robot is mainly used for reduce the risk process and consuming time and avoid labors. It is build by microcontroller, Servo motor and color sensor.

1.2 Features

The arm has three servos which are controlled through the use of only one Arduino uno microcontroller board based on the ATmega328.The system comprise of color sensor, servo motors, and microcontroller . Color sensor detects the specified color of the object and microcontroller read this from the data at its input ports.When object is determined by the robot, the gripper of the robot will pick the object and place it to the specified color differentiating station.The base is equipped with high torque servo.It can lift object up to weight of 110 gm.keeping the design of robotic arm gripper simple, as well as implementing the gripping mechanism without using gears and with one servo motors.The gripper is equipped with micro servo which makes it lighter .

1.3 Scope

Our project aims to develop a prototype for automatizing industrial sorting and distribution system. At present, many of the sorting and distribution system depends on human workforce. If we use this kind of industrial automation then we do not require to spend extra expenses involved in employee benefits. In case of mechanical failure, only computer and maintenance engineers are required to bring the system in working condition. It can make the production line safe for the employees by deploying robots to handle risky conditions. Although many companies hire hundreds of production workers for a up to three shifts to run the plant for the maximum number of hours.Industrial automation fulfills the aim of the company to run a manufacturing plant for 24 hours in a day 7 days in a week and 365 days a year. Though our color sorter robotic arm is a prototype of real project,it can sort 7 colored object per min. but if we built real project for industry it can sort approximately 15 to 20 object per min.

1.4 Chapter Summary

In this chapter, we explore the concepts of Industrial Automation and its prospects in building a Industry automation with its huge benefits, such as, increased productivity, quality and safety at low costs.

2 Literature Review

This chapter introduces a general preview about color theory, the relationship between light and color and background about the components used. Literature review and related works of historical systems are covered.

2.1 Color Identification

Color names can be used and conjure reasonably consistent perceptions. There have eleven basic color names have been identified, namely; white, gray, black, red, yellow, green, blue, orange, purple, pink, and brown. Most or all colors can be described in terms of variations and combinations of these colors [1]. Due to the fact that human color vision is accomplished in part by three different types of cone cells in the retina, it follows that three values are necessary and sufficient to define any color.

2.1.1 Color Theory

In 1969 Brent Berlin and Paul Kay advanced a theory of cross-cultural color concepts centered on the notion of a basic color term [2]. A basic color term (BCT) is a color word that is applicable to a wide class of objects (unlike blonde), is monolexemic (unlike light blue), and is reliably used by most native speakers (unlike chartreuse). The languages of modern industrial societies have thousands of color words, but only a very slender stock of basic color terms. English has 11: red,yellow,green,blue,black,white,gray,orange,brown,pink,and purple. Slavic(Indo-European) languages have 12, with separate basic terms for light blue and dark blue.[1]

Berlin (1969) said that there have three values can be thought of as coordinates of a point in three-dimensional space, giving rise to the concept of color space. Hue, saturation, luminance (HSL) is one such color coordinate system, or color space. A more precise method of describing color is by hue, saturation, and lightness. Hue is the attribute of a color according to its similarity with one of the colors red, green, or blue, or a combination of adjacent pairs of these colors considered in a closed ring, as shown in figure 2.3.

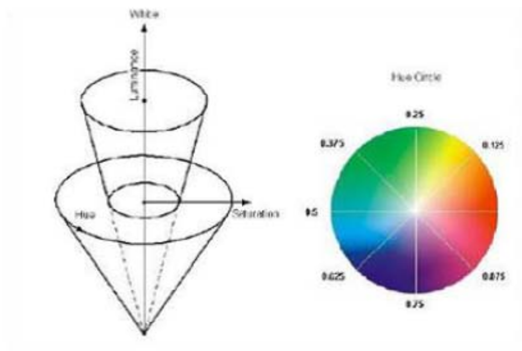

Figure 2.1: HSL diagram with Hue circle

Another method describes color by hue, saturation, and value is HSV. HSL and HSV color spaces are nearly similar except that HSL assigns the high color values for colors that approaching to the white color with a bounded saturation while HSV assigns the high color values for colors that approaching to black [3]. HSV color model is represented in a single cone as shown in figure 2.3.

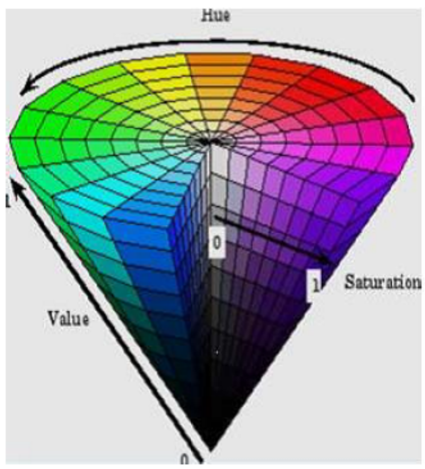

Figure 2.2: HSV color model single hex cone.

2.1.2 *Definition of Light*

Light is an electromagnetic radiation within a certain portion of the electromagnetic spectrum, the word usually refers to visible light, which is visible to the human eye and is responsible for the sense of sight. In physics, the term light sometimes refers to electromagnetic radiation of any wavelength, whether visible or not. In the sense, gamma rays, X-rays, microwaves and radio waves are also light [4].

2.1.3 *The Relationship between Color and Light*

Color is a sensation of the brain, and the ability to see colors is only possible in the presence of light. Sir Isaac Newton discovered that white daylight is actually made up of a spectrum of colors, namely; Red, Orange, Yellow, Green, Blue and Violet. It is only when light falls on an object that its characteristic color is seen and there are only three ways that an object interacts with light rays. When all the rays of light are absorbed human eye sees the color black; when all are reflected it sees white and when all but one are absorbed, it sees the color that is reflected [5].

Figure 2.3: Reflection of light

A leaf , for example, is Green. Under white light, the leaf appears green because it reflect light in the green portion of the spectrum and absorb light of other wavelengths. If a filter is used to remove green from the light source, the leaf reflects very little light and appears black. Each color has a unique wavelength that is processed and recognized in the eye and then transmitted to the brain. The rods and cones of the retina of the eye recognize the colors of light and degree of black and white thus making it possible to perceive the color characteristics of objects. The brain then translates this information to be felt by the human eye. Hence, seeing is actually a function of the brain, as it is known that one can have perfectly healthy eyes but still be totally blind if the vision part of the brain is damaged.

A factor that affects the quality of perception of color is illumination, or the level of light. Light that is too strong or too dim distorts and stresses vision and thus inhibits the ability

Table 2.1: **The visible colors and their corresponding range of wavelengths**

Color	Wavelength Range (nm)
Violet	380 to 410
Indigo	410 to 450
Blue	450 to 510
Green	510 to 560
Yellow	560 to 600
Orange	600 to 630
Red	630 to 780

to appreciate the true color characteristics of objects. Achieving the appropriate level of visual comfort is thus of paramount importance and is affected by the color scheme as well as the function of the space and the tasks performed in it, e.g. reading, watching television, eatingetc.

2.1.4 Color Sensor

Color sensors register items by contrast, true color index. True color sensors are based on one of the color models, most commonly the RGB model. A large percentage of the visible spectrum can be created using these three primary colors. Depending on the sensor, it can be programmed to recognize only one color, or multiple color types for sorting operations. Some types of color sensors do not recognize color intensity, instead focusing on light wavelengths.

2.1.5 Sorting

Sorting is any process of arranging items in some sequence and in different sets. It has two common distinct meanings such as ordering and categorizing. Ordering is arranging items of the same kind, class, nature, etc. in some ordered sequence while categorizing is grouping and labeling items with similar properties together. The main purpose of sorting information is to optimize its usefulness for specific tasks. Various sorting tasks are essential in industrial processes. For example, during the extraction of gold from ore, a device called a shaker table uses gravity, vibration, and flow to separate gold from lighter materials in the ore (sorting by size and weight)[6].

Other type of sorting is color sorting which is handled in this project. Color sorting system is one of the useful systems in Industrial. It can be used to differentiate items based on the color of the item itself. Traditionally, the color sorting process is done by the operator manually. However, this method has some shortage such as increasing the cost of the product, slow production speed, and inaccuracy due to the human mistakes.

Simply, a color sensor is used to identify the color based on the reflected light from the object to be sorted. Then sends signals to a controller to continue the sorting process.

2.1.6 Application of color sorting systems

The following are five examples of systems uses color sorting techniques in different applications.

1. Sorting automotive parts with different colors:

A manufacturer of automotive parts needs to differentiate parts whose only visible difference is a slight variation in color. One part is black, the other a dark grey. Because efficiency demands that the part be sorted at a high rate of speed, the opportunity for mistakes is extremely high. By using advance color sensing technology, the manufacturer can sort these parts at an enhanced rate of speed, saving time and virtually eliminating errors.

2. Assembly of medical closures with different color components:

In this situation, containers of liquid medications consist of an aluminium cap with a plastic cover. A complete closure assembly requires the assurance of proper color combination of two components. Because of this absolute necessity of color accuracy, color sensing technology proves invaluable.

3. Color sensing in the food industry:

Sensing a white target on a white background is challenging using conventional photoelectric sensors. A manufacturer who needs to insure the presence of the white cap on a jar of mayonnaise improves accuracy with a color sensor that employs the RGB color concept. This technology improves the contrast between two slightly different whites.

4. Ammunition final inspection:

An ammunition manufacturer codes the style and caliber of bullets with various colors on the tips. The need to insure that the proper type and caliber of product are correctly packaged necessitates the automation of the product line with color sensing technology. Because of the critical nature of this application, it is recommended that two color sensing stations be implemented to add an extra safety margin to the operation [7].

2.2 Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P. It is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on our computer, used to write and upload computer code to the physical board.

”Uno” means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

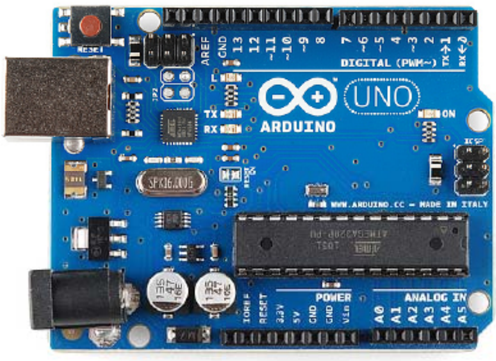

Figure 2.4: Arduino UNO

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package [8].

2.2.1 Arduino UNO Technical Specifications

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

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
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Figure 2.5: Technical Details of Arduino Uno

Arduino UNO has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller , simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. we can tinker with our UNO without worring too much about doing something wrong, worst case scenario we can replace the chip for a few dollars and start over again [8].

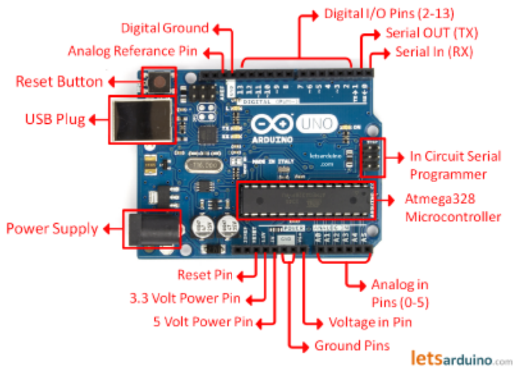

Figure 2.6: Arduino UNO pin Diagram

2.2.2 *Power*

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

VIN: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V : The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3.3v : A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND : Ground pins [8].

2.2.3 *Memory*

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library) [8].

2.2.4 *Input and Output*

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, arising or falling edge, or a change in value.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller [8].

2.2.5 Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-toserial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; [8].

2.2.6 Automatic Software Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". Its Also can be disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; [8].

2.2.7 USB Over-Current Protection

The Arduino Uno has a resettable poly fuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed [8].

2.2.8 Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins [8].

2.3 RGB Color Sensor

The TCS3200 programmable color light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line. In the TCS3200, the light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters [9]. In the TCS3210, the light-to-frequency converter reads a 4 x 6 array of photodiodes. Six photodiodes have blue filters, 6 photodiodes have green filters, 6 photodiodes have red filters, and 6 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All photodiodes of the same color are connected in parallel. Pins S2 and S3 are used to select which group of photodiodes (red, green, blue, clear) are active. Photodiodes are 110 m x 110 m in size and are on 134-m centers.

2.3.1 Specifications

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm   C Temperature Coefficient
- Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package

Table 2.2: Terminal Function

Terminal Name	No.	I/O	Description
GND	4		Power supply ground. All voltages are referenced to GND
OE	3	I	Enable for fo (active low).
OUT	6	O	Output frequency (fo).
S0, S1	1,2	I	Output frequency scaling selection inputs.
S2, S3	7,8	I	Photodiode type selection inputs.
VCC	5		Supply voltage

Table 2.3: Selectable Options

S0	S1	Output Frequency Scaling
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Table 2.4: The different type of Photodiode

S2	S3	Photodiode Type
L	L	Red
L	H	Blue
H	L	Clear(no Filter)
H	H	Green

2.3.2 *Terminal Function*

The color sensor should be connected to Arduino Uno as follows: VCC to 5V , GND to GND , S0 to digital pin 5 , S1 to digital pin 6 , S2 to digital pin 7, S3 to digital pin 8 , OUT to digital pin 9. The Terminal functions of color sensor are listed in Table 2.2.

TCS3200 can output the frequency of different square wave ,different color and light intensity correspond with different frequency of square wave. There is a relationship between the output and light intensity. The range of the typical output frequency is 2HZ5̃00KHZ. We can get different scaling factor by different combinations of S0 and S1[10]. These are listed in Table 2.3.

TCS3200 has four photodiode types. Red , blue, green and clear, reducing the amplitude of the incident light uniformity greatly, so that to increase the accuracy and simplify the optical. When the light project to the TCS3200 we can choose the different type of photodiode by different combinations of S2 and S3 [10]. These combination are listed in table2.4 .

2.3.3 *Functional Block Diagram*

TCS3200 color sensor provides the intensity value of Red, Green and Blue colors detected at its input lens. The output is available on a single pin in the form of a square wave. Here is the block diagram of the TCS230 color sensor.

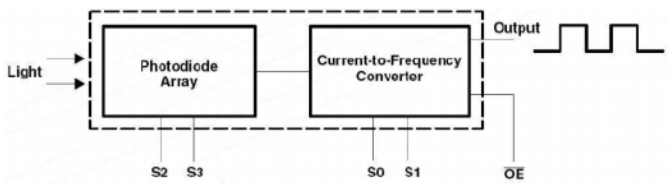

Figure 2.7: Functional Block Diagram

The first block that comes into picture is the Photodiode array. The photodiode array consists of three sets of diodes each for sensing Red, Green and Blue color respectively. So, when we are measuring the intensity of Red light we need to enable only the Red photodiode array and disable the other two photo diode arrays[11].

2.3.4 Applications

- Test strip reading.
- Sorting by color.
- Ambient light sensing and calibration
- Color matching.

2.4 Servo Motor

2.4.1 Servo Motor : MG995R

The 11kg.cm Metal Gear Servo Motor - MG995 can rotate approximately 160 degrees.The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spectrum and Hitec.

Figure 2.8: Servo Motor : MG995R

Specifications

- weight- 55.0g
- Dimension 40.7*19.7*42.9mm
- Stall torque 9.4 kg.cm (4.8V), 11kg.cm(6V)
- Operating speed 0.17sec/60degree(4.8v), 0.14sec/60degree(6v)
- Operating voltage 4.8-7.2V
- Temperature range 00 to 55C
- Dead band width 5us [12].

2.4.2 Servo Motor : MG996R

This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 11kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor.

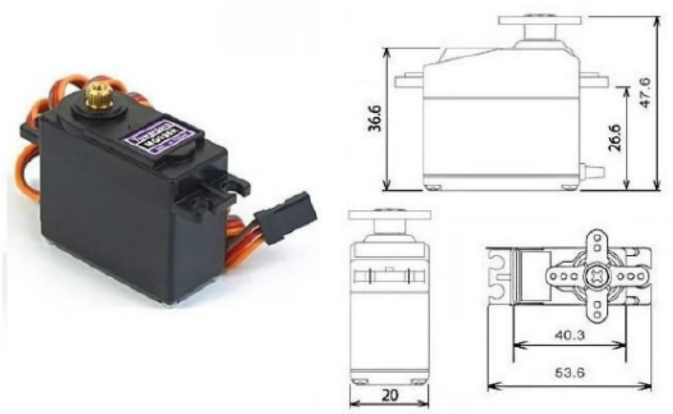

Figure 2.9: Servo Motor : MG996R

The gearing and motor have also been upgraded to improve dead bandwidth and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spectrum and Hitec. This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast [13].

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgfcm (4.8 V), 11 kgfcm (6 V)
- Operating speed: 0.17 s/60 (4.8 V), 0.14 s/60 (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA
- Stall Current 2.5 A (6V)
- Dead band width: 5 s
- Stable and shock proof double ball bearing design
- Temperature range: 0 C -55 C

2.5 Robotic Claw

The only materials needed to build this are a few pieces of hardware and a little bit of steel. Specifically:

- 8x M4 screws (length of about 14mm)
- 8x M4 nuts
- 4x small washers to fit M4 screws

- available spool of steel.

Figure 2.10: Robotic claw

The quantity we use to build claw is listed below:

- 2x Claw
- 1x Main Body
- 1x Main Gear (LEFT)
- 1x Main Gear(RIGHT)
- 8x Shaft Short
- 8x Shaft Tall
- 4x Support Link

2.5.1 Servo Clamps

This base stand also called servo holder stand bracket.This Aluminium Pan and Tilt is for horizontal surface, unassembled, Pan and Tilt Mount Servo Stand for Robot Clamp Claw.

This bracket used in the shoulders and knees or other joint of humanoid robots,biped robots etc.

Figure 2.11: Servo Base Stand

This Short 'U' shaped brackets is designed to fit most standard sized servos. This bracket is both lightweight and strong. Various holes are located on the bracket that fit most servo horns.

Figure 2.12: Wrist Stand

This high quality metal servo horn has the standard Servo Erector Set hole pattern.This heavy-duty aluminum servo horn lends extra strength to our servo applications.

Figure 2.13: Servo Horn

2.6 Power

Lithium Polymer batteries are a newer type of battery now used in many consumer electronics devices. They have been gaining in popularity in the radio control industry over the last few years, and are now the most popular choice for anyone looking for long run times and high power.

2.6.1 LiPo Packs versus NiMH Batteries

LiPo batteries offer three main advantages over the common Nickel-Metal Hydride (NiMH) or Nickel Cadmium (NiCd) batteries:

- LiPo batteries are much lighter weight, and can be made in almost any size or shape.
- LiPo batteries offer much higher capacities, allowing them to hold much more power.
- LiPo batteries offer much higher discharge rates, meaning they pack more punch.

But, just as a coin has two sides, there are some drawbacks to LiPo batteries as well.

- LiPo batteries have a shorter lifespan than NiMH/NiCd batteries. LiPos average only 300400 cycles.

- The sensitive chemistry of the batteries can lead to fire if the battery gets punctured and vents into the air.

- LiPo batteries need special care in the way they are charged, discharged, and stored. The required equipment can be expensive [14].

2.6.2 Voltage

A LiPo cell has a nominal voltage of 3.7V. For the 7.4V battery above, that means that there are two cells in series (which means the voltage gets added together). This is sometimes why we will hear people talk about a "2S" battery pack - it means that there are 2 cells in Series. So a two-cell (2S) pack is 7.4V, a three-cell (3S) pack is 11.1V, and so on.

The voltage of a battery pack is essentially going to determine how fast our vehicle is going to go. Voltage directly influences the RPM of the electric motor (brushless motors are rated by kV, which means 'RPM per Volt'). So if we have a brushless motor with a rating of 3,500kV, that motor will spin 3,500 RPM for every volt we apply to it. On a 2S LiPo battery, that motor will spin around 25,900 RPM. On a 3S, it will spin a whopping 38,850 RPM. So the more voltage we have, the faster we're going to go.

- 3.7 volt battery = 1 cell x 3.7 volts (1S)

- 7.4 volt battery = 2 cells x 3.7 volts (2S)

- 11.1 volt battery = 3 cells x 3.7 volts (3S)

- 14.8 volt battery = 4 cells x 3.7 volts (4S)

- 18.5 volt battery = 5 cells x 3.7 volts (5S)

- 22.2 volt battery = 6 cells x 3.7 volts (6S)

- 29.6 volt battery = 8 cells x 3.7 volts (8S)

- 37.0 volt battery = 10 cells x 3.7 volts (10S)

- 44.4 volt battery = 12 cells x 3.7 volts (12S) [14].

2.6.3 *Capacity*

The capacity of a battery is basically a measure of how much power the battery can hold. Think of it as the size of your fuel tank. The unit of measure here is milliamp hours (mAh). This is saying how much drain can be put on the battery to discharge it in one hour. Since we usually discuss the drain of a motor system in amps (A), here is the conversion: 1000mAh = 1 Amp Hour (1Ah) The capacity of the battery above is 5000mAh. This means that a load of 5000mAh (or 5A) would drain the battery completely in one hour. We use this information on charging as well, because it works in the opposite way as well. If we charge the above battery at 5 Amps, it will be completely charged in about an hour. Physics is a bit fickle, and there is energy lost along the way, so it won't be an hour on the dot, but it's a good ballpark time. The capacity of the battery is like the fuel tank - which means the capacity determines how long we can run before we have to recharge. The higher the number, the longer the run time. For R/C cars and trucks, the average is 5000mAh - that is our most popular battery here in the store. But there are companies that make batteries with larger capacities. Traxxas even has one that is over 12000mAh! That's huge, but there is a downside to large capacities as well. The bigger the capacity, the bigger the physical size and weight of the battery. Another consideration is heat build up in the motor and speed control over such a long run. Unless periodically checked, we can easily burn up a motor if it isn't given enough time to cool down, and most people don't stop during a run to check their motor temps. Keep that in mind when picking up a battery with a large capacity Discharge Rating ("C" Rating) The last two specs had a direct impact on certain aspects of the vehicle, whether it's speed or run time. This makes them easy to understand. The Discharge Rating is a bit harder to understand, and this has lead to it being the most over-hyped and misunderstood aspects of LiPo batteries. The C Rating is simply a measure of how fast the battery can be discharged safely and without harming the battery. One of the things that make it complicated is that it's not a stand-alone number; it requires us to also know the capacity of the battery to ultimately figure out the safe amp draw (the "C" in C Rating actually stands for Capacity). Once we know the capacity, it's pretty much a plugand-play math problem. Using the above battery, here's the way you find out the maximum safe continuous amp draw: 20C = 20 x Capacity (in Amps) 20 x 5 = 100A The resulting number is the maximum sustained load we can safely put on the battery. Going higher than that will result in the battery becoming, at best, unusable. At worst, it could burst into flames. So our example battery above can handle a maximum continuous load of 100A. Most batteries today have two C Ratings: a Continuous Rating (which we've been discussing), and a Burst Rating. The Burst rating works the same way, except it is only applicable in 10-second bursts, not continuously. For example, the Burst

Rating would come into play when accelerating a vehicle, but not when at a steady speed on a straightaway. The Burst Rating is almost always higher than the Continuous Rating. Batteries are usually compared using the Continuous Rating, not the Burst Rating. Our example battery has a Burst Rating of 30C. That means it can handle a load of 150A, but only for 10 seconds or less [14].

2.7 Breadboard

A breadboard is used to build and test circuits quickly before finalizing any circuit design. The breadboard has many holes into which circuit components like ICs and resistors can be inserted. A typical breadboard is shown below:

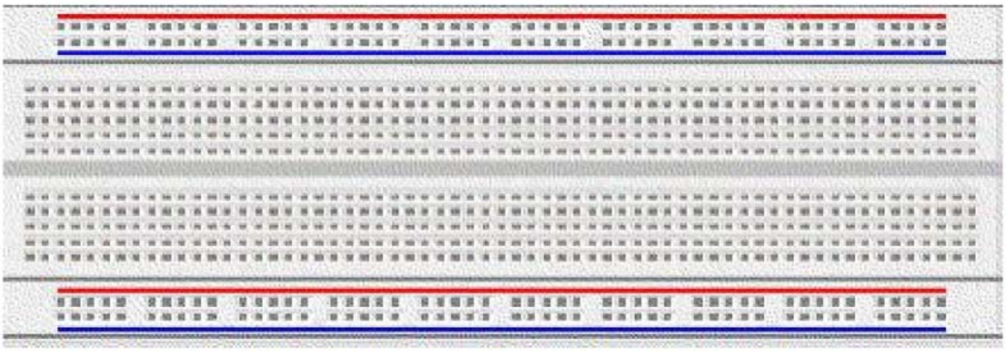

Figure 2.14: A typical breadboard

The bread board has strips of metal which run underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally while the remaining holes are connected vertically.

To use the bread board, the legs of components are placed in the holes. Each set of holes connected by a metal strip underneath forms a node. A node is a point in a circuit where two components are connected. Connections between different components are formed by putting their legs in a common node. The long top and bottom row of holes are usually used for power supply connections. The rest of the circuit is built by placing components and connecting them together with jumper wires. ICs are placed in the middle of the board so that half of the legs are on one side of the middle line and half on the other [15].

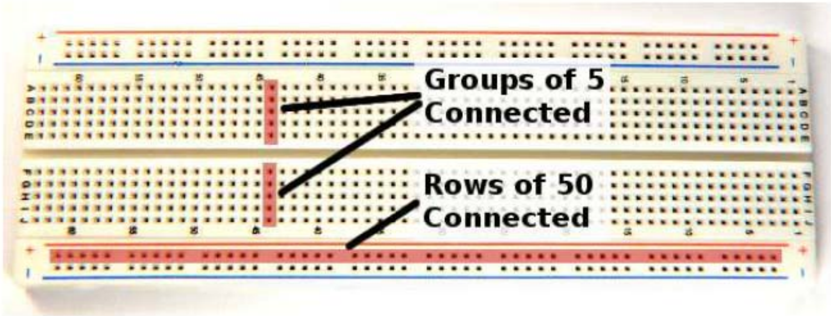

Figure 2.15: Breadboard pin diagram

2.8 Related Works

1. P. Agarwal , R. K. tiwari and S. Banyal (March 2014) , In their journal , they are attempting to solve is to create an autonomous robot that can identify objects when placed on the conveyor belt based on colour sensing and then sort by relocating them to a specific location.The pick and place robotic arm with conveyor belt is a system that detects the object on the conveyor belt, picks that object from source location and places at desired location based on colour identified or metal/non-metal.For detection of object, firstly infrared sensors are used which detect presence of object as the transmitter to receiver path for infrared sensor is interrupted by placed object.Also the colour of the object is detected by the colour sensor. Its metallic nature is detected by the metal detector [16].

2. Vishnu R. Kale and V. A. Kulkarni (july 2013) , presents a smart approach for a real time inspection and selection of objects in continuous flow. The basic theme of this project is object flowing on conveyor are sensed, selected and sorted depending on their color using image processing . Image processing procedure senses the objects in an image captured in real-time by a webcam and then identifies color and information out of it. This information is processed by image processing for pick-and-place mechanism. The sorting process is based on a 2 phase operative methodology defined 1) a self-learning step where the apparatus learns to identify objects ; 2) an operative selection process where objects are detected, classified using a decisional algorithm and selected in real time. The Project deals with an automated material handling system.It aims in classifying the colored objects by colour, size, which are coming on the conveyor by picking and placing the objects in its respective pre-programmed place [17].

3. Manjunatha V. G. (june 2014), The goal of his project is to implement a programmable industrial robot for color sorting. he investigates the development of an intelligent and

low-cost monitoring system for color identification and segregation. The main purpose is to optimize the productivity and avoid human mistakes. A serial image acquisition device (camera) is used to capture the image and are sent to SD card through a microcontroller. The microcontroller performs color detection algorithm to recognize the dominant color of the object, and it sends commands to the robotic arm to pick and place the objects to their respective locations. A robotic arm is a manipulator, which has about same number of degrees of freedom as in human arm. He use DC motors for joint rotations involved in the robotic arm, these motors are interfaced with microcontroller through motor driver circuits. These motor drivers are able to efficiently control the speed and direction of motors [18].

2.9 Chapter summary

Throughout this chapter, we saw multiple techniques proposed in different journals about some of the famous applications. A brief introduction to our Micro-controller and its benefits are also discussed. Also a brief introduction of color theory, light and color relationship and application of color sorting system in various industry.

3 System Design & Development

In this chapter, we describe the development details of the project. Here we present the RGB color sensor configuration with Microcontroller , servo motors, battery & switch for our proposed Industrial automation system. Here we have written our program with Arduino IDE using various standard libraries .

3.1 Robot arm

The robot arm is probably the most mathematically complex robot we could ever build. But here we have made very simple arm for picking or placing the object according to its colour. Design of robot arm includes:

A. Degrees of Freedom (DOF)

The degrees of freedom, or DOF, are a very important term to understand. Each degree of freedom is a joint on the arm, a place where it can bend or rotate or translate. You can typically identify the number of degrees of freedom by the number of actuators on the robot arm. The robotic arm which we are going to use has 3 DOF.

B. Robot Workspace

The robot workspace (sometimes known as reachable space) is all places that the gripper can reach. The workspace is dependent on the DOF angle/translation limitations, the arm link lengths, the angle at which something must be picked up at, etc. The workspace is highly dependent on the robot configuration. The figure shows the free body diagram of robotic arm which we will be used for our project

C. Torque calculations

Torque calculation is a very important factor for designing the robot arm. As our system needs 20cm of robot arm we can calculate motor torque from that. Here is the FBD of given robot arm. Calculation W1 - Weight of block (0.150 kg)

W2 - Weight of gripper (0.350kg,motor weight is included)

W3 - Weight of link (0.500 kg)

Wm1 Weight of motor 1 (.250kg)

L - Length between gripper and motor1 (20 cm)

Distance between M2 & M3 is 5cm.

Torque about motor2

= (W1+W2)*L + W3*(L/2)

=(0.150+0.350)*20 + (.500*10)

=10+5

=15 kg.cm

Torque about motor3

= (W1+W2)*(L+5) + W3*(L/2 + 5) + Wm1*5

= (.150+0.350)*25 + (.500*15) + (.250*5)

= 12.5+7.5+1.25

= 21.25 kg.cm

D. Motors

We use servo motors in arm to get good accuracy, precision, repeatability. we use 3 servo motors (Motor1,2&3) which have 24kg.cm torque.

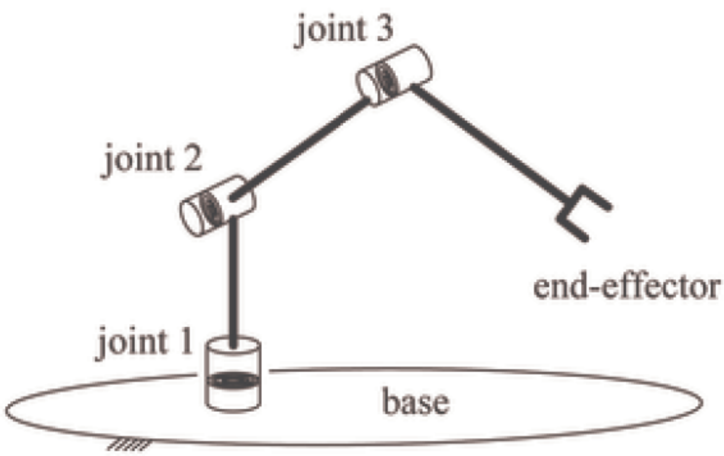

Figure 3.1: 3-DOF robotic arm Diagram

In each joint we use servo motor which have 180 degree rotation power and 24kg.cm torque. The gripper is powered by strong metal servo motor, allowing our robot to interact with soft, light objects. We recommend colored paper object.It looks like -

Figure 3.2: Gripper Connecting with servo motor

Second joint is wrist of robot arm. It works only when robotic claw needs to go up and down. When color sensor detect color then robotic claw go down and pick the object, then rotate 120 or 90 degree and place the object.Here both servo holder bracket and u-shape bracket used.

Figure 3.3: Wrist Connecting with servo motor

Third one is base of our robotic arm. Here we use servo holder stand bracket with servo. This bracket is both lightweight and strong.

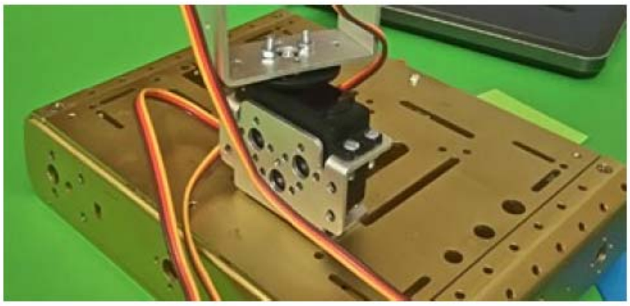

Figure 3.4: Robot base Connecting with servo motor

3.2 Servo

A Servo is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as coded signal exists on the input line, the servo will maintain the angular position of the shaft. Servos are extremely useful in robotics. The motors are small, as we can see by the picture above, have built in control circuity, and are extremely powerful for their size. A standard servo such as the Futaba S-148 has 42 oz/inches of torque, which is pretty strong for its size. It also draws power proportional to the mechanical to the mechanical load. A lightly loaded servo, therefore doesnt consume much energy. The guts of a servo motor are shown in the picture below.

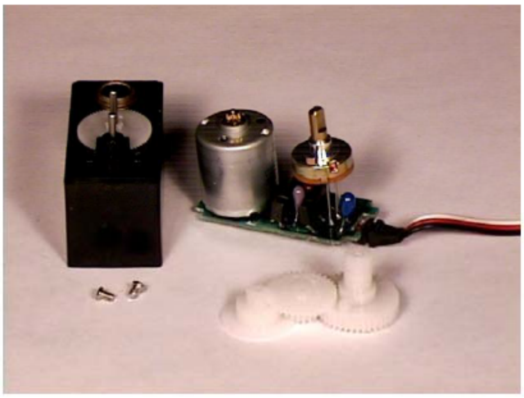

Figure 3.5: A servo disassembled

So how does a servo work? The servo motor has some control circuits and a potentiometer (a variable resistor, aka pot) that is connected to the output shaft. In the picture above, the pot can be seen on the right side of the circuit board. This pot allows the control circuitry to monitor the current angle of the servo motor. If the shaft is at the correct angle, then the motor shuts off. If the circuit finds that the angle is not correct, it will turn the motor the correct direction until the angle is correct. The output shaft of the servo capable of travelling somewhere around 180 degrees. Usually, its somewhere in the 210degree range, but it varies by manufacturer. A normal servo is used to control an angular motion of between 0 and 180degrees. A normal servo mechanically not capable of turning any further due to a mechanical stop built on to the main output gear [19].

3.2.1 Servos connecting with Arduino UNO

The servo motor has three leads. The color of the leads varies between servo motors, but the red lead is always 5V and GND will either be black or brown. The other lead is the control lead and this is usually orange or yellow. This control lead is connected to digital pin 9.

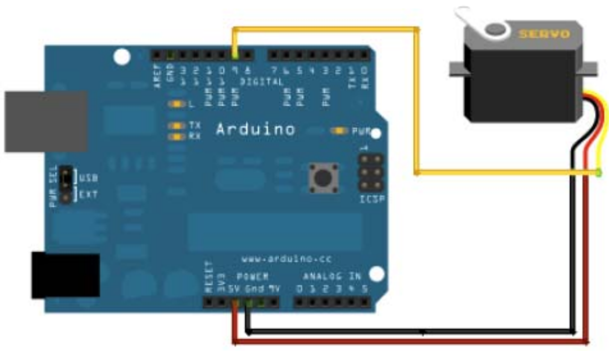

Figure 3.6: Servo motor connecting with arduino uno

- We use 3 servo motors to implement this project. One MG995R and two MG996R. When we are going to use commands in the servo library, we need to tell the Arduino IDE that we are using the library with this command:

  #include ⟨*Servo.h*⟩

- As usual, we then use a variable 'servoPin' to define the pin that is to control the servo. This line:

Servo servo;

- In the 'setup' function we need to link the 'servo' variable to the pin that will control the servo using this command:

  servo.attach(servoPin);

- The variable 'angle' is used to contain the current angle of the servo in degrees. In the 'loop' function, we use two 'for' loops to first increase the angle in one direction and then back in the other when it gets to 180 degrees. The command:

  servo.write(angle);

- Connect the Arduino UNO and servo motors with jumper wire as below:

  servo_1 connect with D1

  servo_2 connect with D2

  servo_3 connect with D8

  servo_4 connect with D9

Listing 3.1: Arduino Code for servos

```
#include<Servo.h>
Servo servo_1;
Servo servo_2;
Servo servo_3;
Servo servo_4;
int pos = 0;
void setup()
{
servo_1.attach(1);
servo_2.attach(2);
servo_3.attach(8);
servo_4.attach(9);
}
void loop()
{
  servo_1.write(180);
```

```
delay(1000);
  for(pos = 90; pos>=23; pos-=1)
{
 servo_2.write(pos);
delay(30);
}
 servo_3.write(100);
delay(1000);
  for(pos =45; pos < 175; pos += 1)
{
 servo_4.write(pos);
delay(20);
}
delay(1000);
 for(pos = 175; pos>=90; pos-=1)
{
 servo_4.write(pos);
delay(20);
}
  servo_2.write(90);
delay(2000);
  servo_1.write(90);
delay(1000);
  for(pos = 90; pos>=23; pos-=1)
{
  servo_2.write(pos);
delay(30);
}
  servo_3.write(100);
delay(1000);
  for(pos =90; pos < 175; pos += 1)
{
  servo_4.write(pos);
delay(15);
}
delay(1000);
   for(pos = 175; pos>=45; pos-=1)
```

```
    {
      servo_4.write(pos);
    delay(20);
    }
      servo_2.write(90);
    delay(1000);
  }
```

3.3 Configuring RGB color sensor with Arduino Uno

We use TCS3200 color sensor .This Color Sensor is a complete color detector, including a TCS3200 RGB sensor chip and 4 white LEDs. The TCS3200 can detect and measure a nearly limitless range of visible colors. The TCS230 has an array of photodetectors, each with either a red, green, or blue filter, or no filter (clear). The filters of each color are distributed evenly throughout the array to eliminate location bias among the colors.

Step1: Upload Code

- Connect the Arduino UNO to pc with Mini USB cable.
- Open the Arduino IDE1.1, and upload the following code.
- Choose the corresponding board and serial port.
- Compiling sketch until done uploading appears.

Listing 3.2: Arduino Code for RGB color sensor

```
// the used PINs
const int s0=3;
const int s1=4;
const int s2=5;
const int s3=6;
const int sensorData=7;

// Global variable to store data
```

```
int dataR=0;
int dataG=0;
int dataB=0;

void setup()
{
    pinMode(s0,OUTPUT);
    pinMode(s1,OUTPUT);
    pinMode(s2,OUTPUT);
    pinMode(s3,OUTPUT);
    pinMode(sensorData,INPUT);
    Serial.begin(9600);
    digitalWrite(s0,HIGH);
    digitalWrite(s1,HIGH);

}

void loop()
{
    digitalWrite(s2,LOW);
    digitalWrite(s3,LOW);
    dataR=pulseIn(sensorData,LOW);
    delay(20);

    digitalWrite(s2,LOW);
    digitalWrite(s3,HIGH);
    dataG=pulseIn(sensorData,LOW);
    delay(20);

    digitalWrite(s2,HIGH);
    digitalWrite(s3,HIGH);
    dataB=pulseIn(sensorData,LOW);
    delay(10000);
    if(isRed())
    {
      Serial.println("this is the RED color");
    }
```

```
    if(isBlue())
    {
     Serial.println("this is the Blue color");
    }
    if(isGreen())
    {
     Serial.println("this is the GREEN color");
    }
    if(isYellow())
    {
     Serial.println("this is the YELLOW color");
    }
}

bool isRed()
{
  if((dataR<18 && dataR>12) && (dataG<49 && dataG>43) && (dataB<70 && dataB>64))
  {
    Serial.println("red");
    Serial.println(dataR);
    return true;
  }
  else
{
    return false;
  }
}
bool isGreen()
{
  if((dataR<53 && dataR>47) && (dataG<44 && dataG>38) && (dataB<34 && dataB>28))
  {
    Serial.println("green");
    Serial.println(dataG);
    return true;
  }
```

```
    else
    {
      return false;
    }
}
bool isBlue()
{
    if((dataR<103 && dataR>90) && (dataG<41 && dataG>27)
        && (dataB<88 && dataB>73))
    {
      Serial.println("blue");
      Serial.println(dataB);
      return true;
    }
    else
    {
      return false;
    }
}
bool isYellow()
{
    if((dataR<14 && dataR>8) && (dataG<31 && dataG>25) &&
          (dataB<19 && dataB>13))
    {
      return true;
    }
    else
    {
      return false;
    }
}
```

Step2: Connecting RGB sensor with Arduino

1. The color sensor module power supply is from 3.27v to 5.5v
2. Connect the Arduino UNO and color sensor with jumper wire as below.

S0 connect to D5

S1 connect to D6

S2 connect to D7

S3 connect to D8

VCC connect to 5v

OUT connect to D9

GND connect to GND

OE connect to GND

3. Use white objects to perform white balance correction, after doing that, do not move the color sensor module and light source to ensure the accuracy of white balance value, therefore, we put the color sensor on the paper white part .

4. Open Arduino IDE and click serial monitor and set the BaudRate as 9600

5. The serial Monitor would display lots of parameter values, and we choose some of them to explain below

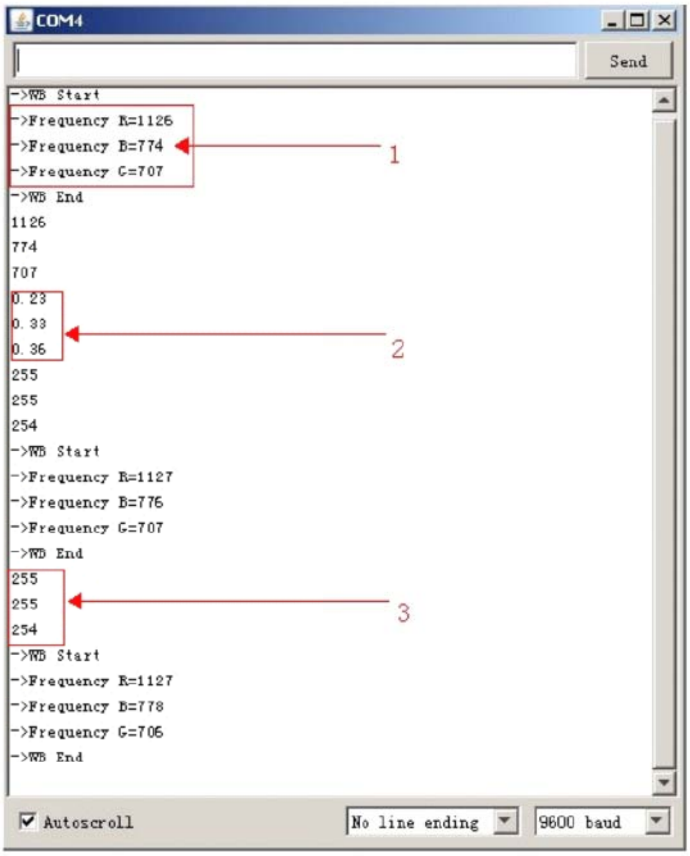

Figure 3.7: RGB color sensor Serial Port readings.

- As the figure , the mark 1 is the RGB value at 1s period.The RGB value is Red=1126,Green=707 and blue=774.
- Then the mark 2 is the scale factor.
- Then use the scale factor for adjusting the RGB value to 255 as mark 3.

Step3: Test Color With different color paper

1. The normal measurement distance is 0-2 cm, so we put the color sensor module on the paper another part.
2. Open the serial monitor and now got the objects RGB value like (13,14,14), which means we have got the object color.

3. If r= 4 to 7, g = 4 to 7 and b = 3 to 5 then it detects as red color object. These values depend on the selected frequency-scaling, as well as from the surrounding lighting and also distance between object and sensor.

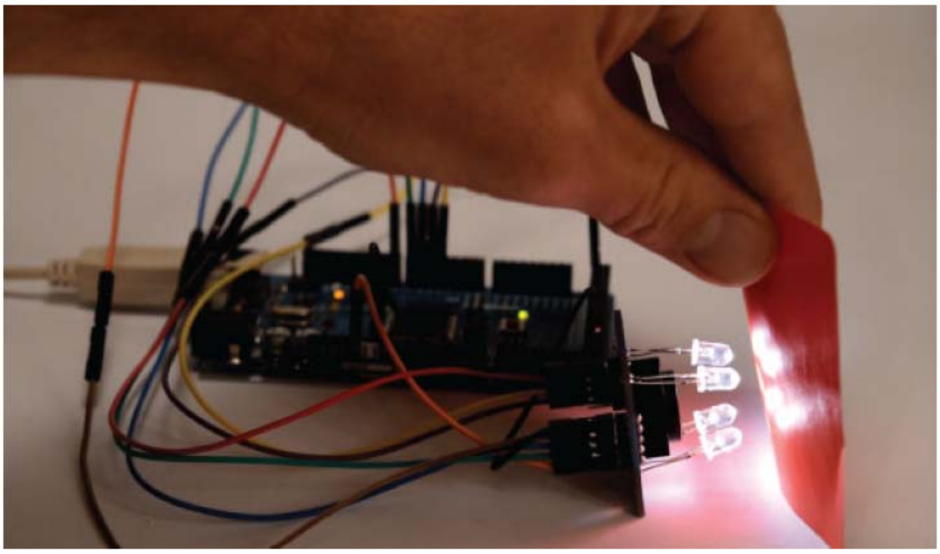

Figure 3.8: Test sensor with color paper

3.4 Conveyor Belt

Conveyor Belts are durable and reliable components used in automated distribution and warehousing.In combination with computer controlled pallet handling equipment this allows for more efficient retail, wholesale, and manufacturing distribution. It is considered a labor saving system that allows large volumes to move rapidly through a process, allowing companies to ship or receive higher volumes with smaller storage space and with less labor expense. Rubber conveyor belts are commonly used to convey items with irregular bottom surfaces, small items that would fall in between rollers or bags of product that would sag between rollers

A conveyor belt is the carrying medium of a belt conveyor system (often shortened to belt conveyor). A belt conveyor system is one of many types of conveyor systems. Today there are different types of conveyor belts that have been created for conveying different kinds of material available in PVC and rubber materials. A conveyor belt can be a slide and be controlled by the force of gravity [20].

We made a slope type conveyor belt which is not autonomous that means No use of timing belt and gear. Our autonomous robot that can identify objects when the object placed on the slope, based on color sensing and then sort by relocating them to a specific location. Figure 3.9 shows the slope type conveyor belt.

Figure 3.9: Conveyor Belt

3.5 Chapter Summary

In this chapter, System implementation of the project was shown throughly. Necessary Steps, Hardwares, codes and schematics were shown. Procedure for connecting all the components together was also briefly described.

4 Project Simulation & Implementation

In this chapter we describe the simulation details of the project. we present how this robotic arm works with some flow chart and circuit diagram.

4.1 Color Sensing Procedure

When sensor detects an object then it sends information to microcontroller and then microcontroller sends signal to the various motors of the robotic arm to grip the object and place it in the specified location.

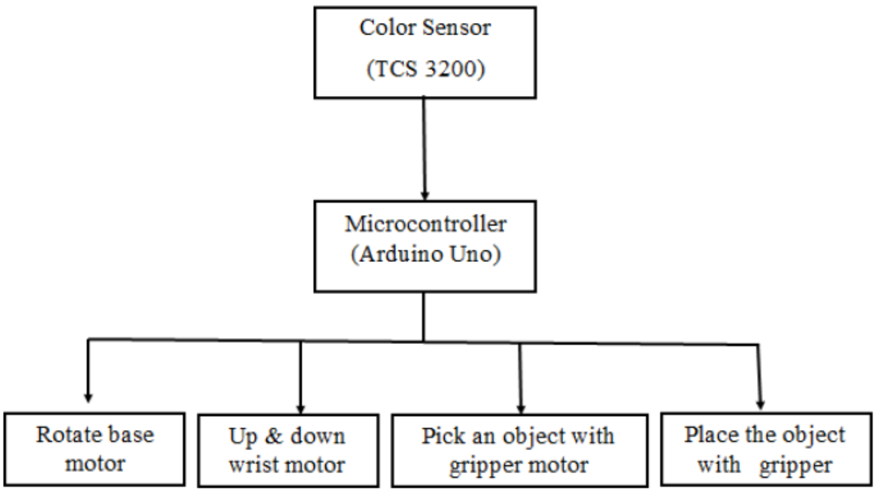

Figure 4.1: Information Flow Diagram

Based upon the colour detected, the robotic arm moves to the specified location, and placing the objects in its respective pre-programmed place and comes back to the original position. Three motors are used in the robotic arm. One to control the rotational motion of the base, one to control the wrist movement and last one to control the gripper, that is to hold and drop the ball. Here we provide 6.5v voltage for servo motor operation and provide 5v voltage for RGB color sensor operation. The figure 4.2 shows the initial position of the robotic arm when power is applied and the robot is ready for operation.

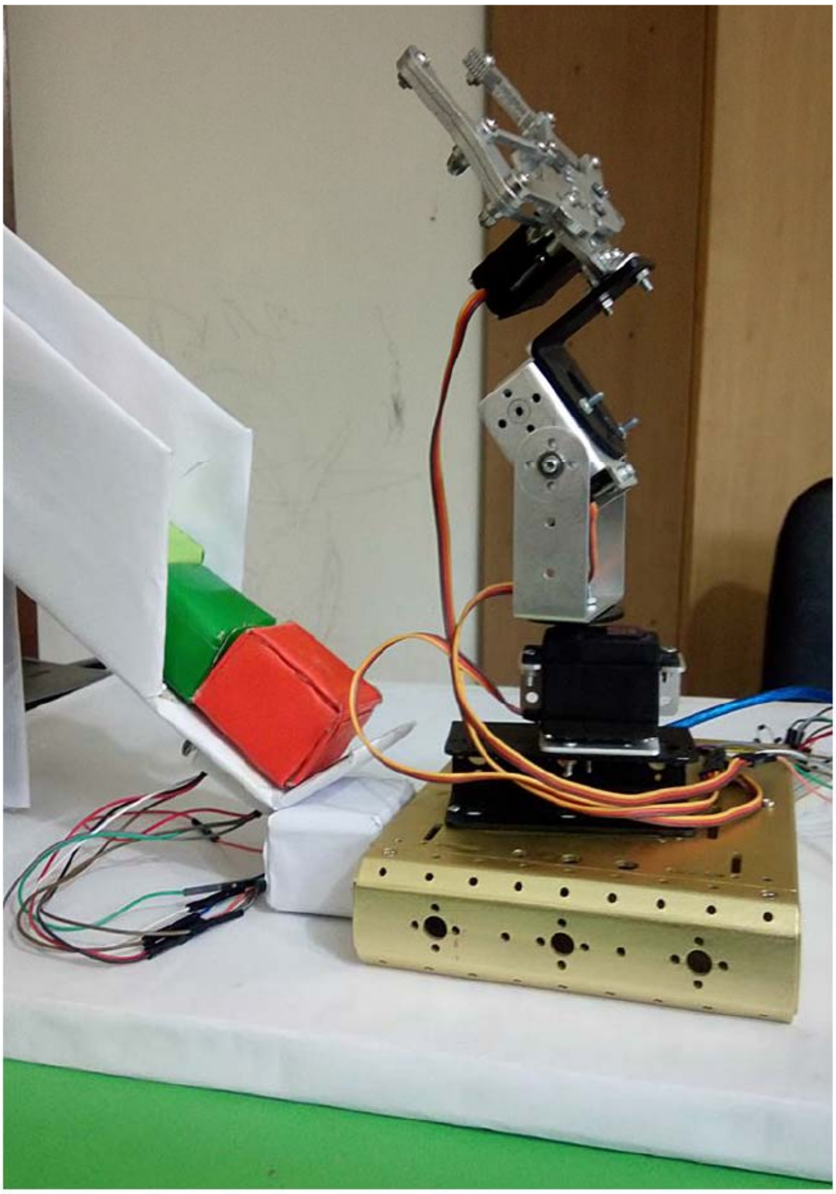

Figure 4.2: Initial position of robotic arm

When one object arrive into the color sensor then it detect the frequency of that color. If the object is red color then it sends a frequency like the following figure 4.3. These values depend on the selected frequency-scaling, as well as from the surrounding lighting and also distance between object and sensor.

R= 5	G= 5	B= 5
R= 5	G= 4	B= 5
R= 7	G= 4	B= 5
R= 5	G= 4	B= 5
R= 7	G= 5	B= 7
R= 5	G= 5	B= 5
R= 5	G= 5	B= 5
R= 4	G= 4	B= 5
R= 5	G= 5	B= 5
R= 4	G= 4	B= 5
R= 7	G= 7	B= 5
R= 7	G= 5	B= 8
R= 7	G= 5	B= 7

Figure 4.3: Frequency Scaling Values of RGB color sensor in Serial Monitor

Now if we run the Serial Monitor we will start getting this kind of values shows in figure 4.3. These values depend on the selected frequency-scaling, as well as from the surrounding lighting. This figure 4.3 shows that if red=4 to 7,green = 4 to 7 and blue =5 to 7 then it returns that the object is red color.Here three values differ due to the different sensitivity of each photodiode type, as seen from the photodiode spectral responsivity diagram from the datasheet of the sensor.

As well as figure 4.4 shows the frequency values for blue color. It shows that if red=13 to 15 , green= 4 to 5 and blue= 13 to 15 then it returns that the object is blue color. These values depend on the selected frequency-scaling, as well as from the surrounding lighting and also distance between object and sensor.

```
R= 13  G= 4  B= 13
R= 14  G= 4  B= 14
R= 14  G= 4  B= 14
R= 14  G= 5  B= 14
R= 13  G= 4  B= 13
R= 13  G= 4  B= 13
R= 13  G= 5  B= 14
R= 13  G= 4  B= 13
R= 13  G= 5  B= 14
R= 13  G= 5  B= 13
R= 13  G= 5  B= 13
R= 13  G= 4  B= 13
R= 14  G= 5  B= 14
```

Figure 4.4: Frequency Scaling Values for blue color in Serial Monitor

4.2 Robotic Arm Control Procedure

In figure 4.5 shows the flow chart of Robotic arm control procedure.

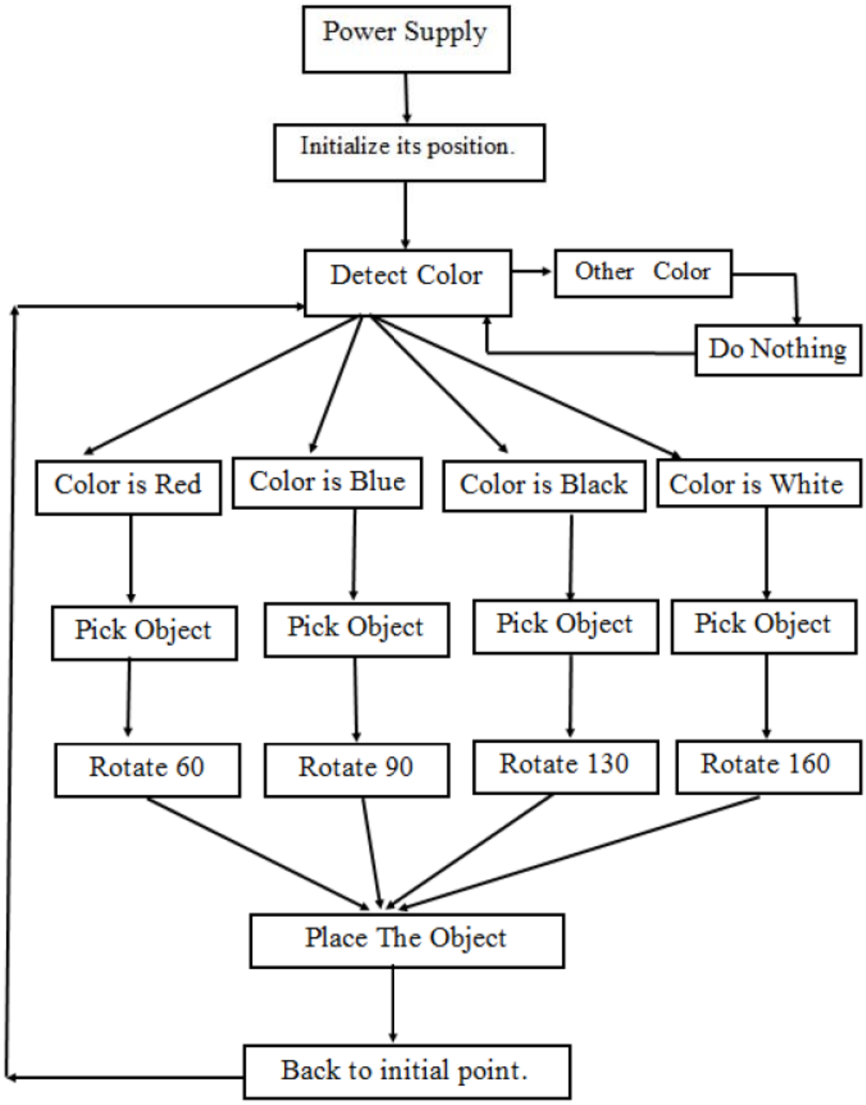

Figure 4.5: Flow Chart of Arm Control Procedure

When power is applied to servo motors and microcontroller then it initialize its position to work. Here for servo motor we used 6.5v voltage and for microcontroller we used 5v voltage. Initially robotic arm is not working unless RGB color sensor detect any color. We fixed this arm for 4 color (Red, Blue, White and Black). If another color is detect then it does not work.

If red color is detected then robotic claw pick up the object with its gripper then rotate 60 degree and place the object in the same colored area. After that it back to its initial point. Again if blue colored object is detected by color sensor then robotic claw pick the object then rotate 90 degree and place the object in the same colored area and then back to its initial point.

Moreover if the color sensor detect black color then it pick up the object and then rotate 130 degree and place the object in the same colored area. After that it back to its initial point. Again if white colored object is detected by color sensor then robotic claw pick the object then rotate 160 degree and place the object in the same colored area and then back to its initial point.

5 Conclusion

This paper demonstrates implementation of automatic color object sorting system. The aim of the project was to have a fully functional robotic arm which sorts different colored balls and the target is achieved successfully. In the final run of the project red, blue , black and green object were successfully sorted. The colour sensor IC TCS3200 show almost stable response in various sunlight conditions. our demo shows that our system can recognize various colored objects, do the calculations to physically move the arm to that point, pick up and drop objects is specified end locations, and distinguish objects to sort them into different locations.

5.1 Limitations

Every successful invention has its own limitations. There are a number of limitations in the project also, which are as follows:

a) Here in case of color measurement and detection of color over a wide range which might not gave the proper result always.

b) Here only four colored (red , green,blue and black) products are asserted precisely. Other color can be asserted by preparing a good and more accurate structure and highly efficient coding of microcontroller.

c) Sometime the object slip on the slope, as gripping mechanism of slope is not good enough.Here we don't use autonomous conveyor belt. No use of timing belt and gear also causes the faulty power transmission to the slope .

d) Any point on this robotic arm can only move along a circular path (60 to 160 degree). Any task involving motion other that the circular motion cannot be performed by this system.

5.2 Future Works

Time and resources take an important part to have a good product. Day by day good resources come and help to improve these products. Even there are many sectors in our project to improve. We can apply artificial intelligence for color sorting in our system, where robotic arm can gather information from environment. That things make our system fully automated. Now we are using color sensor for objects color sorting, instead of using color sensor we will using camera for sorting objects. Now we are using color sensor for detect objects color only, we will adding shape detection for sorting object by their shape also.

we will use camera instead of sensors and also use machine learning to gather information from environment. We will improve our system as If there is no object detect in about 30 to 45 sec then system will be turn off for saving power.

References

[1] C. Hardin, “Berlin and kay theory,” *Analysis And Implementation of New Solutions for Home Port*, 2013.

[2] K. Berlin, B., “Basic color terms,” *University of California Press, Berkeley*, 1969.

[3] G. A. Agoston, *Color theory and its application by arts and design (chapter 8-10)*. Springer-Verlag Berlin Heidelberg GmbH.

[4] wikipedia.org, “Light,” https://en.wikipedia.org/wiki/Light, Accessed on: 2017-01-20.

[5] physicsclassroom.com, “Light absorption, reflection, and transmission,” http://www.physicsclassroom.com/class/light/Lesson-2/Light-Absorption,-Reflection,-and-Transmission, Accessed on: 2017-01-28.

[6] wikipedia.org, “Sorting,” https://en.wikipedia.org/wiki/Sorting, Accessed on: 2017-01-28.

[7] wikipedia.org , “Colour sorter,” https://en.wikipedia.org/wiki/Colour_sorter, Accessed on: 2017-01-28.

[8] arduino.cc, “Arduino board uno,” https://www.arduino.cc/en/main/arduinoBoardUno, Accessed on: 2017-02-06.

[9] howtomechatronics.com, “Arduino color sensing tutorial tcs230 tcs3200 color sensor,” http://howtomechatronics.com/tutorials/arduino/arduinocolorsensingtutorial-tcs230tcs3200colorsensor/, Accessed on: 2017-02-10.

[10] dfrobot.com, "Tcs3200 color sensor pin discription," https://www.dfrobot.com/wiki/index.php/TCS3200_Color_Sensor_(SKU:SEN0101), Accessed on: 2017-02-10.

[11] eprolabs.com, "Color sensor tcs3200," https://wiki.eprolabs.com/index.php?title=Colour_Sensor_TCS320(Accessed on: 2017-02-15.

[12] towerpro.com, "Servo motor mg995," http://www.towerpro.com.tw/product/mg995/, Accessed on: 2017-02-15.

[13] towerpro.com , "Servo motor mg996r 180rotation," http://www.towerpro.com.tw/product/mg995robotservo180rotation/, Accessed on: 2017-02-15.

[14] rogershobbycenter.com, "A guide to understanding lipo batteries," https://rogershobbycenter.com/lipoguide/, Accessed on: 2017-02-21.

[15] acadgild.com, "Brief intro to various components used in arduino & raspberry pi," https://acadgild.com/blog/briefintrovariouscomponentsused-arduinoraspberry-pi/, Accessed on: 2017-03-03.

[16] S. B. S. K. S. C. Pankaj Agarwal, Ratnesh Kumar tiwari, "Color/metal sensing sorting system," *International Journal of Engineering Research and Management Technology*, vol. 1, pp. 168–178, March 2014.

[17] V. A. K. Vishnu R.Kale, "Object sorting system using robotic arm," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, pp. 3400–3406, July 2013.

[18] V. Manjunatha, "Camera Based Color Identification Robot for Typecasting," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, March 2014.

[19] phas.ubc.ca, "Brief description of servo," http://www.phas.ubc.ca/k̃otlicki/Physics_319/servos.html, Accessed on: 2017-03-25.

[20] wikipedia.org, "Conveyor belt," https://en.wikipedia.org/wiki/Conveyor_belt, Accessed on: 2017-04-10.