DESIGN AND IMPLEMENTATION OF C OLOR DETECTING SYSTEM

A Project and Thesis submitted in partial fulfillment of the requir ements for the Award of Degree of Bachelor of Science in Electri cal and Electronic Engineering

SUBMITTED BY

Titas Ahmed
(ID : 123-33-1121)

SUPERVISED BY

SaikatBasak
Senior Lecturer
Department of EEE

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
FACULTY OF ENGINEERING
DAFFODIL INTERNATIONAL UNIVERSITY

December 2018

CERTIFICATION

This is to certify that this project and thesis entitled "**DESIGN AND IMPLEMENTATION OF COLOR DETECTING SYSTEM**" is done by the following students under my direct supervision and this work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering of Daffodil International University in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Signature of the candidates

____

Name: Titas Ahmed

ID #: 123-33-1121

Countersigned

____

SaikatBasak

Senior Lecturer

Department of Electrical and Electronic Engineering
Faculty of Engineering
Daffodil International University

Dedicated to

Our Parents
&
Beloved Teachers

CONTENTS

LIST OF FIGURE

LIST OF TABLES

LIST OF ABBREVIATION

IDE	Integrated Development Environment
IC	Integrated Circuit
DC	Direct Current
LCD	Liquid Crystal Display
LED	Light Emitting Diodes
SPD	Spectral power distribution

ACKNOWLEDGEMENTS

First, we would like to thank and express our gratefulness to Almighty Creator for giving us the strength and energy to complete this project successfully. We wish to acknowledge our gratefulness to our project supervisor **Mr. SaikatBasak , Senior Lecturer, Department of Electrical & Electronic Engineering**. For his valuables advice, endless patience, important suggestions, and above all scholarly guidance from beginning to the end of the work. We would like to express our heartiest gratefulness to all of our teachers here at the Department of Electrical & Electronic Engineering, Daffodil International University of Bangladesh. Discussions with many of them have enriched our conception and knowledge about this project work.We are grateful to **Professor Dr. M. ShamsulAlam, Dean, Faculty of Engineering, Daffodil International University** for his valuable suggestions. We are also obliged to Professor Dr. Md. ShahidUllah, Head of the Department of Electrical and Electronic Engineering, Daffodil International University for his Cooperation. Finally, I would like to thank all my teachers, students and staff of **Department of Electrical and Electronic Engineering** at Daffodil International University and all of my family members and friends whose names are not mentioned here.

ABSTRACT

In this project we are going to interface TCS3200 color sensor with Arduino UN O. TCS3200 is a color sensor which can detect any number of colors with right programming. TCS3200 contains RGB (Blue, Green, Red) arrays. As shown this fi gure on microscopic level one can see the square boxes inside the eye on sensor. These square boxes are arrays of RGB matrix. Each of these boxes contain three sensors, one is for sensing BLUE light intensity, one is for sensing GREEN light intensity and the last in for sensing RED light intensity. Each of sensor arrays i n these arrays selected separately depending on requirement. The particular color ca n be feature to sense by module and leave the colors. It contains filters for that selected purpose. There is forth mode that is no filter mode with no filter mode t he sensor detects white light. In this work a device has been made using color se nsor module and programmed using visual basic and identify the color interfaced with a computer and color shades of any object in textile and fabric industrial ap plications.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Color sensor module based on TCS3200 that is a programmable colorlight-to-frequency converter, it could filter RGB data source from light andconvert it to a square wave(50% duty cycle) with frequency straightproportional to light intensity (irradiance). The full-scale output frequency canbe scaled by one of three scheduled values via two control input pins(SO, S1Selectable Options 2%, 20%, 100% frequency),and pin S2, S3 control the filterof RGB. Digital inputs and digital output deliver interface to a microcontroller orother logic circuitry directly. Output qualify (OE) places the output in the statehigh-impedance for multiple-unit sharing of input line in a microcontroller.Then user can calculate the color of the light by RGB values.

1.2 Color Sensing Meter

Incidence Color Meters are like to Illuminance Meters in design, the Anomaly being that Color Meters incidence operate with a three element sensor crafted to measure photometric brightness as it decrease on a surface and provide color information.

CL-70F CRI ILLUMINANCE METER:The CL-70F CRI Illuminance Meter is an entry-level key for the measurement and appreciation of the illuminance, color temperature, and color rendering exponent (CRI) of various illumination sources such as LEDs and fluorescent lamps. Its high-resolution CMOS sensor binding and displays the spectral power distribute of current and future generation light sources cover LEDs, HID, Halogen, and OLEDs pro viding unparalleled color measurement exactness.It is most commonly used in Muse um, Restaurants, Theaters and Studios for exact light measurements.

1.3Problem Statement

It has limited problem. Sometimes we are not detect the proper color so using col or detecting system we detect the color easily.

1.4 Objectives

The main objective of our project is detect to the color of an object. Our target is to detects three to ten colors.The target colors are Blue, Red and Green.To des cribe color system In Order That color integrity and mixings is identified perfectly and is distinct of human Espial or subjective judgments.

1.5Project/Thesis Outline

This Project/thesis is organized as follows:

Chapter 1is entitled “Introduction”. It introduces problem statement, objectives andm ethodology used in this project.

Chapter 2is “Literature review”.

Chapter 3is entitled “Analysis and Simulation”.It summarizes automaticcontrol proces s. It also describes whole project control process.

Chapter 4is entitled “Hardware Development”.

Chapter 5 is entitled “Result and Discussions”. It describes briefly the hardwareco mponent use in this project. It also describes the works of the hardware in thispro ject.

Chapter 6 is entitled “Conclusion”. It describes about the software,programming lan guage and method used in this project.

CHAPTER 2
LITERATURE REVIEWS

2.1 Introduction

Color sensor is one of the most characteristics at light, though we may not alway s be discreet of this fact and of its magnitude, taking it for permitted. However, even though color starts a unique role of great value in our daily life and as we shall soon see in fully a few industrial and scientific applications the number of sensors dedicated to color sensing is surprisingly small and more importantly the existence or operation principles of such sensors are very little known to the gener al public.

2.2 Historical Background of Color Sensing

When first digital cameras enter in the nineties, the main technology using for the ir reflect sensors was the CCD technologywith the leading technology, another type of sensor started to arise the CMOS. In order to be Capable to list color inform ation, digital imaging sensors were (and still are) difficult decorated with a so-called Bayer model color filter. Today, CMOS sensors have replaced CCD sensors in greatest types of digital cameras. This article, we want to accept a look at th e different types of digital reflect sensors, and explain their technological features.

2.2.1 The CCD Sensor

Until a couple of years ago, whereas CMOS technology became easier to yield, a nd CMOS output of a CCD sensor at its basis sensitivity best to that of a CMO S sensor in period of sensors started to improve in period of the image quality th ey freed, the CCD sensor was the gold grade for digital reflecting sensors. In fact , many (myself included) still consider the color reproduction and detail but that may be a matter of individual preference.

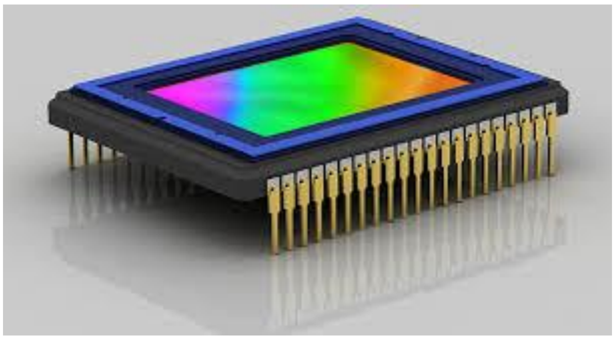

Fig 2.1: CCD Sensor

2.2.2 The CMOS Sensor

The CMOS sensor technology CMOS being short for ‘complementary metal–oxide–semiconductor’ has been around as long as the CCD sensor technology. However, in the early days of digital photography, CMOS sensors were more complex to pr oduce, and is not provide to the same image quality as CCDs. Today, the CMOS and CCD are intents and intention is most difference between them.CMOS and C CD sensors is the main difference is that in a CMOS sensor, the charges passed are not along a column of pixels, but more each pixel has its own readout unit. CMOS sensors is lower power consumption than CCDs, and don’t overheat as easi ly, which makes them especially suited for cameras and video recording with live-view features. This is why early DSLRs had neither, because they using CCD sen sors.Other advantage that CMOS sensors have over CCDs is that they’re less pron e to image noise, especially at higher ISO speeds (i.e. when the outgoing today), they are used in most current cameras because CMOS sensors are right up there with CCDs.

2.2.3 The BSI-CMOS Sensor

Recently many smartphone and compact cameras areused in BSI-CMOS sensors have the circuitry behind the photosensitive layer. The normal CM OS sensor and the BSI-CMOS sensor difference is that the former has its circuitry on upper of the photo sensitive layer, which means that the incoming light is partially blocked A alternati ve of the CMOS sensor is the BSI-CMOS, the ‘back-side illuminated' is CMOS before it hits the pixels. Since their layout is technicall y destroyed, it is as if a regular CMOS sensor were illuminated from behind henc e the designation 'back-side illuminated'.

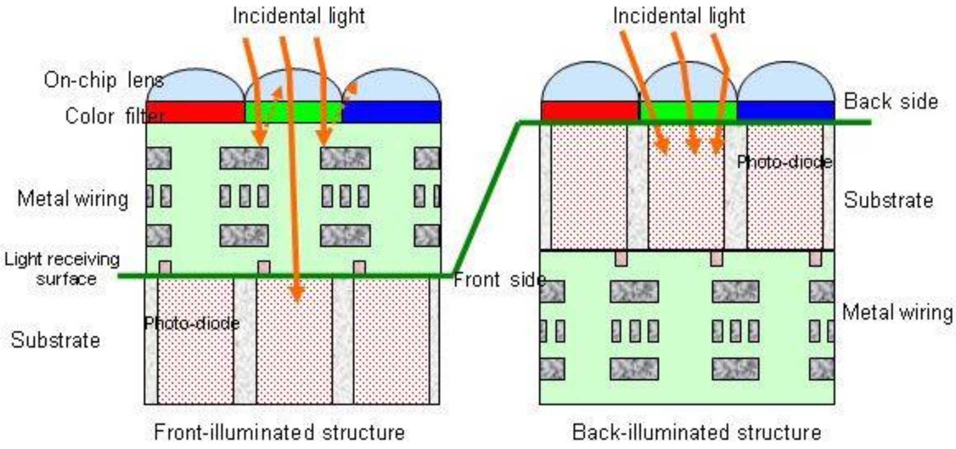

Fig 2.2: Regular CMOS vs. BSI-CMOS

2.2.4Color Splitters Instead of Color Filter

Earlier this year, Panasonic showed off a new kind of reflecting sensor that would use color-
Rending prisms instead of color filters. Light gathering capability promises much b etter this technology, since none of the coming light would get lost when it passe s through the color filters (which filter out most of the light wavelengths so that every pixel receives only red, green or blue light, depending on what filter sits in front of it.) Instead, the micro prisms on top of the pixels would split the inco ming light into red, green and blue wavelengths and would pass these on to the surrounding pixels, practically using around all of the incoming light.

2.2.5 Clear Pixels Instead of Green Pixels

One of the main developers,now a days introduced a new concept for a sensor th at uses clear pixels instead of green pixels, in order to coverage more of the inco ming light. The clear pixels, which registered total wavelengths of the visible light, would support to acquire a clear image, meaning shorter exposition times and les s uproar when taking pictures in low light. Green color is missing fact would be interpolated by comparing the reality from the clear pixels with that of the surrou nding blue and red pixels.

2.2.6 The Organic Sensor

Fujifilm and Panasonic have now a days collaborated on a new type of CMOS se nsor that uses an organic material instead of the general silicon for the photosensit ive layer. This technology promises to have greater light gathering capability than current sensors, which means shorter exposing times and less noise when photograp hing in low light. The final stages of development at organic sensors seems to be, and could be really complete in digital cameras in the near future.

2.3 Technological Overview

Arduino board is a one type of microcontroller based kit. The first Arduinotechnol ogy was developed in the year 2005 by Massimo Banzi.The designers thinking to provide easy and low cost board for professional to build device, students, and ho bbyists. We can be purchased from the seller ordirectly we can make at home usi ng various basic components. The best instances ofArduino for beginners and hobb yists including motor detectors and thermostats, andSimple robots. In the year 2011 , Ad fruit industries expected that over 3lakhs ArduinoBoards had been produced.

2.3.1 Arduino Technology

A typical instance of the Arduino board is Arduino Uno. It includes an ATmega3 28Microcontroller and it has 28-
pins.The pin configuration of the Arduino Uno board is shown in the above. It co nsists of14-
digital I/O pins. Where in 6 pins are used as pulse width modulation O/Ps and 6 Analog I/Ps, a USB connection, a power jack, a 16MHz crystal oscillator, a reset Button, and an ICSP headerIOREF pin or through the pin Vin. This board can op erate with an external supply of 7-12V by giving voltage reference through the

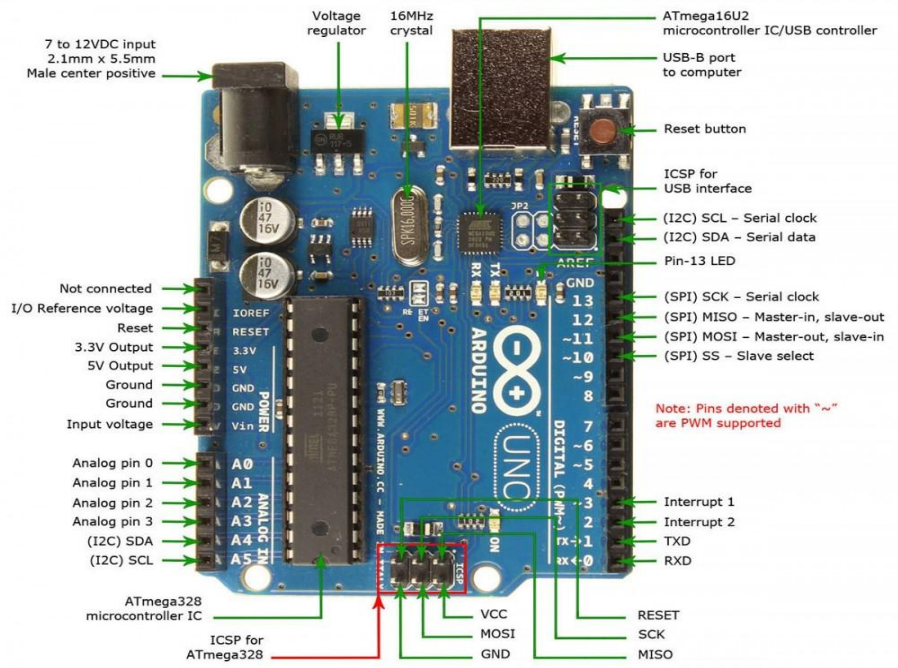

Fig. 2.3 Arduino board

An external supply of 7-12V can operate this board, by giving reference voltage. The board can be powered either from the personal.Computer through external source or a USB port like an adaptor or a battery.

Digital I/Ps: It comprises of 14-digital I/O pins, each pin takes up and provides 40mACurrent.For serial communication, pins-2 & 3 areExternal interrupts, 3, 5, 6, 9, 11 pins givePWM O/P and pin-13 is used to connectLED. Some pins have special functions like pins 0 & 1, which process as aTransmitter and receiver respectively.

AREF: This pin delivers a reference to the analog I/Ps.

Analog I/Ps: It has 6-analog I/O pins, each pin provides a 10 bits resolution.

Reset: The microcontroller reset the pin when it is low.

2.3.2 Arduino Architecture

Harvard architecture uses the Arduino board processor where the program data and code have separate memory. It consists of two memories.They are program memory and data memory. Wherein the data and the code is stored to data memory and the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM 1kb of EPROM and operates with a 16MHz clock speed.

2.3.3 How to program an Arduino

The Arduino programming language can be extended to multiple C++ libraries and you can also use AVR-C programming on this system. With this, you will be able to add AVR-C code onto the Arduino Programming directly. The main advantage of the Arduino technology is, you can directly load the programsinto the device without the need of a hardware programmer to burn the program. Thisis done because of the presence of the 0.5KB of boot loader, that allows the programto be dumped into the circuit. The Arduino tool window contains a toolbar with avarious buttons like new, open, verify, upload and serial monitor

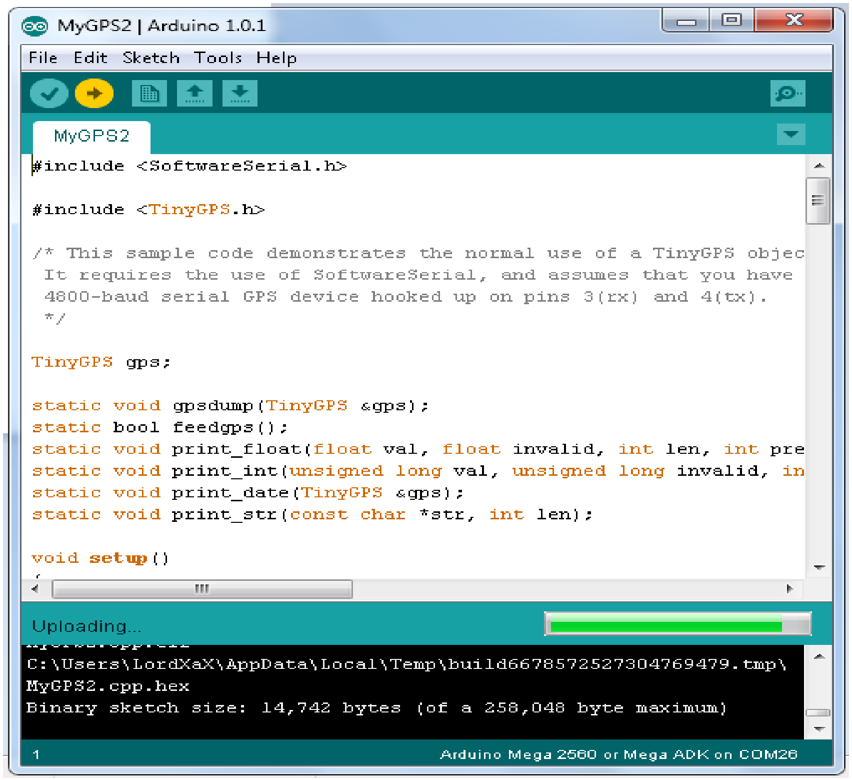

Fig. 2.4 Arduino board programming

Programming into the Arduino board is called as sketches. Every sketchcontains of three parts such as Control code, Variables Declaration and Initialization. Where,

Initialization is written in the Control Code and setup function and is written in the loop function.The sketch is saved with Arduino and any operation like opening a sketch,verifying and saving can be done using the tool menu.Where, Initialization is written in the setup function and Controlcode is written in the loop function. The sketch must be stored in the sketchbook directory.Select the suitable board from the port numbers and tools menu.
Select the tools menu and click on the upload button, then the bootloader uploads the code on the microcontroller.

2.3.4 Advantages of Arduino Technology

It is a cheap.It comes with an open supply hardware configuration that permits users to developtheir own kit.The Arduino software is well-suited with all kinds of in operationsystems like Windows, Linux, Macintosh, etc.It is use very simple for beginners.The Arduino has biggest advantage is its ready to use structure.You don't have to think about programmer connections for programming or any other interface. Just plug it into USB port of your computer and that's it. Itscomes in a complete package from which includes the 5V regulator,an oscillator, a burner, a micro-controller, serial communication interface, LED and headers for the connecting.

2.4 Summary

Now of the era of technological advancement. Hence different technology based smart or autonomous system are getting day by day. Color detection is help that people who loved the robotics competition.

CHAPTER 3

ANALYSIS AND SIMULATION

3.1 Introduction

In this chapter we discuss Arduino Uno Board. The background of color sensor. L ight, color perception and eye. Display board, Bread Board, Color sensor model et c.

3.2 Arduino Uno Board

The Arduino Uno is a microcontroller board based on the ATmega328.It has 6 an alog inputs, 14 digital input/output pins (of which 6 can be used as PWM outputs), a USB connection point a 16 MHz ceramic resonator, a power jack, an ICSP header, and a reset button. It contains everything needed to confirmation the micro controller and connect it to a computer with a USB cable or power it with a AC -to-
DC adapter or battery to get started.The Uno main different from all preceding bo ards 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.Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, to put into DFU mode making it easier.Revision 3 of the board has the following new features:1.0 pinout: SDA and SCL added pins that are near to the AREF pin and two another new pins placed near to the RESET pin, the IOREF that permit the shields to adapt to the voltage provided from the board. In future, that uses the AVR shields would be compatible with both the board, w hich operating with 5V and with the Arduino Due that operates with 3.3V. The s econd one is a not connected pin, that is reserved for future purposes.Stronger RE SET circuit.AT MEGA 16U2 replace the 8U2."Uno" means one in Italian and is n amed to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 wil

l be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platfor m.

Table 3.1 The section of Arduino

ARDUINO MICROCONTROLLER	
Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	6
EEPROM	1 KB
DC Current per I/O Pins	40 mA on I/O Pins; 50 mA on 3,3 V Pin
GENERAL	
Input Voltage	7-12 V
Digital I/O Pins	20 (of wich 6 provide PWM output)
PWM Output	6
PCB Size	53.4 x 68.6 mm
Weight	25 g
Product Code	A000066 (TH); A000073 (SMD)

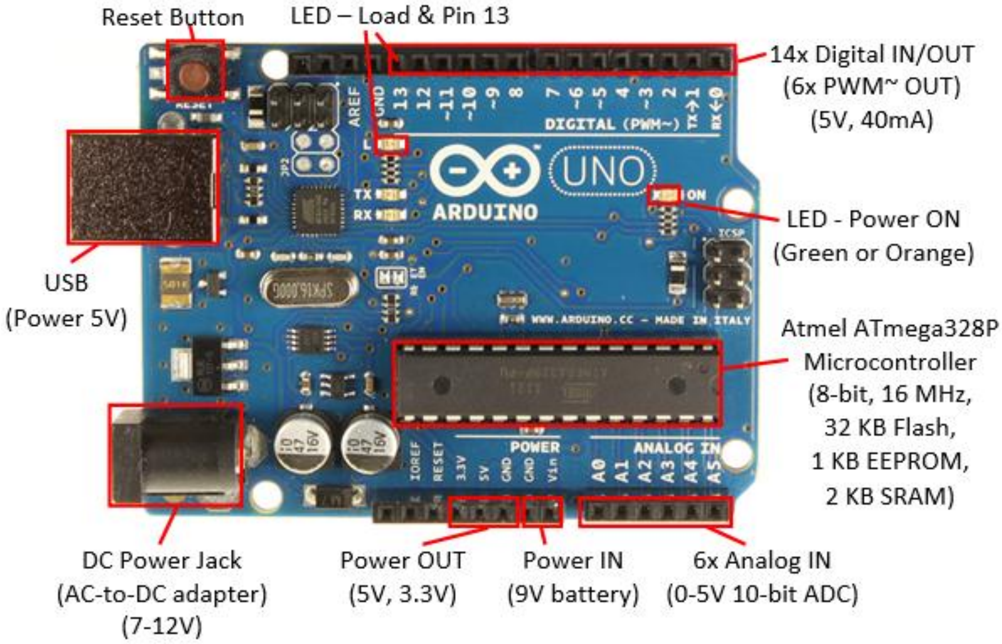

Fig 3.1: ARDUINO UNO BOARD

3.2.1Color sensor

A color sensor has the ability to determine different colors. They will utilize a m eans of emitting light and then look at the reflected light to determine an object's color. This will give the machine the actual color of that object. These sensors a re in use in quite a few different applications today. You can find them in qualit y control systems, packaging systems, and more.

3.2.2Light

Visible light range of 400 to 700nm which an electromagnetic wave. This is a sm all part of all existing wavelengths. We see the light combination of many wavele ngthsis not one wavelength.

The spectral power distribution -

SPD shows the proper dependence of the color in function of light source in the form of diagram at each wavelength that represents energy at each in the visible spectrum.

Fig 3.2:Light

Sunlight at noon of SPD diagram, for example, a very balanced light source clea rly shows. Visible light are present and approximately equal at all wavelength. Co mpared to a synthetic light source, sunlight is a huge amount of energy in the bl ue and red article of the spectrum. Understanding the different sources of light SP D diagram are very useful. The following figure shows a spectral power distributio n of a few typical light sources.

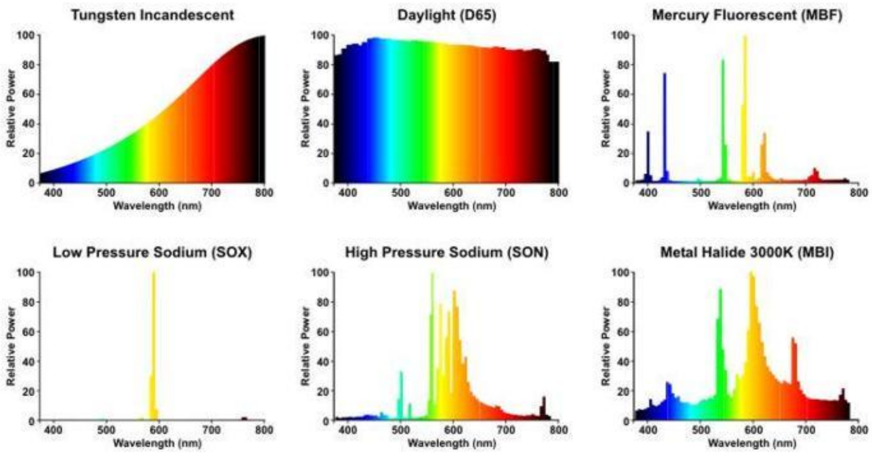

Fig 3.3:Light direction

3.2.3The Eye and Color Perception

The human eye works like a camera. Each neuron is a conical(cone) and stick (ro d). Just sensitive to color are conical neurons. The human has a special cell in th e eye for color detection. Every of the three primary colors have three types of c ones: Red, Blue and Green. Each of these cells reply different to frequencies of li ght, cones are also known as these cells because of their similarity with the afore mentioned geometrical object. Next figures show the light efficiency and spectral s ensitivity of the cone cells of the human eye.

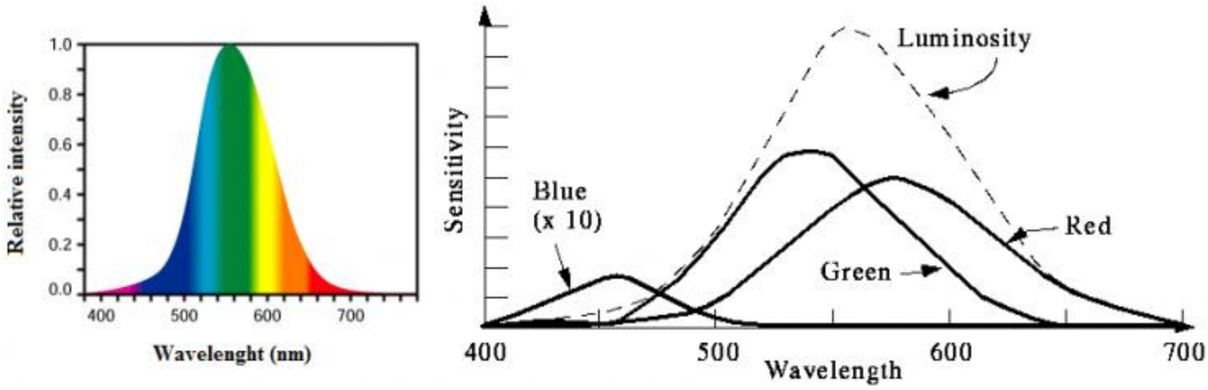

Fig 3.4: Eye and Color Perception

Color signal goes to the brain as a result of the cone cells of the three primary colors of the observed spectrum. This signal can be represented by three values: whereS the spectral sensitivity function and E is the spectral power distribution.The sum of three colors represents can be the color. Three-dimensional vector space means that colors form. The following charts show the a ppearance of the three primary colors necessary to represent all wavelengths of the visible spectrum.

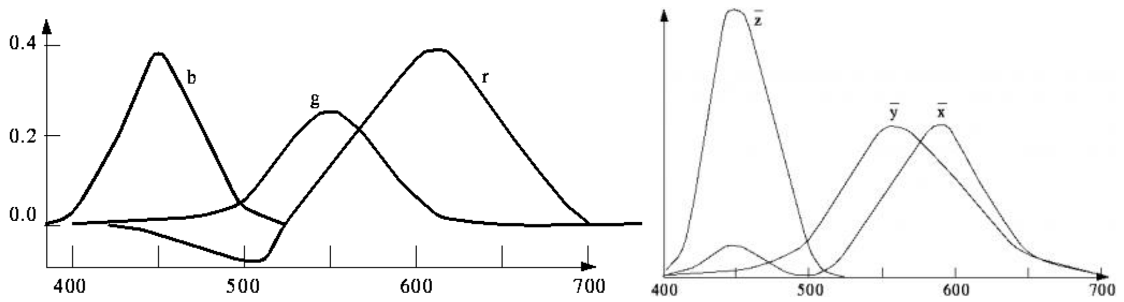

Fig 3.5: GRUPA 1

International Commission on Illumination has defined standards for three primary co lors (X, Y, Z).Negative values indicate that some colors may not exactly produce by simply added the primary colors. For easy color artificial is necessary that all the values are positive. The basic color of Y is deliberately chosen to be identical to the efficiency as a function of light the human eye. The following charts sho w the values of X, Y, and Z that are required to exactly reproduce any color fro m the visible spectrum.

All colors are visible in the "horseshoe" shaped cone in XYZ space. If you look at the plane X + Y + Z = 1 and its projection on the XY plane, we get the C IE chromaticity diagram which is given in the figure.

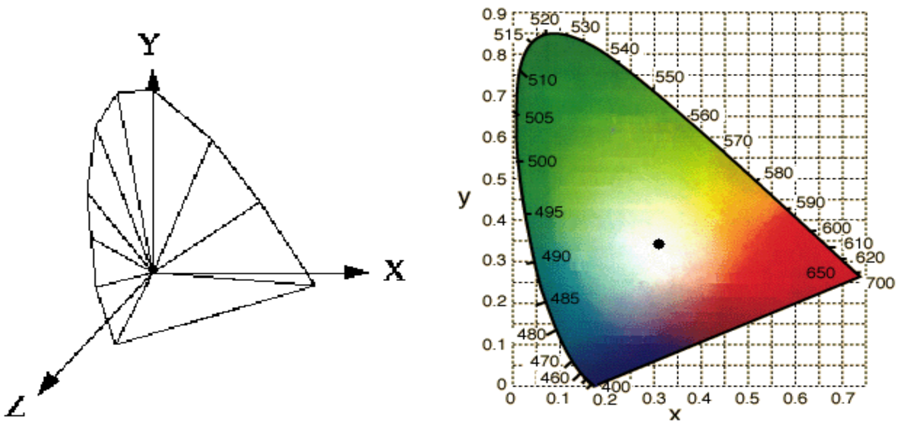

Fig 3.6: GRUPA 2

Edges represents pure color i.e. sinusoidal waves at the accurate frequency. The bl ack point on picture, and white light is same as light emitted from black body at a col of 6447 K

For finding and presenting complementary colors Lab (L * a * b) model is used, where L is the brightness (luminance), "a" the distance between the green and re d "b" the distance between the blue and yellow color.

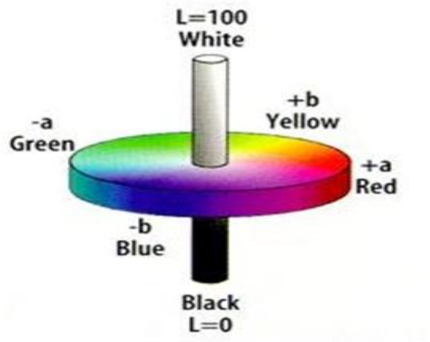

Fig 3.7: Color Types

Many of the color sensors are based on the same principles, i.e. on measuring the intensity of light reflected from a particular surface. Reflected light is detected by photo detectors mostly made in semiconductor technology. Most of the colors that have been detected belong to a group consisting of green, blue and red color. C ombined usage of these sensors it is possible to detect and different color.

Color sensors can be made from discrete components (such as photo resistors, phot o transistors), or they are build in integrated techniques. In this way, the color se nsors are becoming smaller, more accurate, reliable and affordable.

Color models:

RGB -

dependent color model: RGB value detect a different devices or reproduce a given differently, since the color elements (such as phosphors or dyes) and their respon se to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photogr aphy. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.Typical RGB input devices are colo r TV and video cameras, image scanners, and digital camerasTypical RGB output devices are TV sets of various technologies (CRT, LCD, plasma, etc.), computer a nd mobile phone displays, video projectors, multicolor LED displays and large scre ens such as JumboTron, etc.). CMYK refers to the four inks used in some color printing: cyan, magenta, yellow.

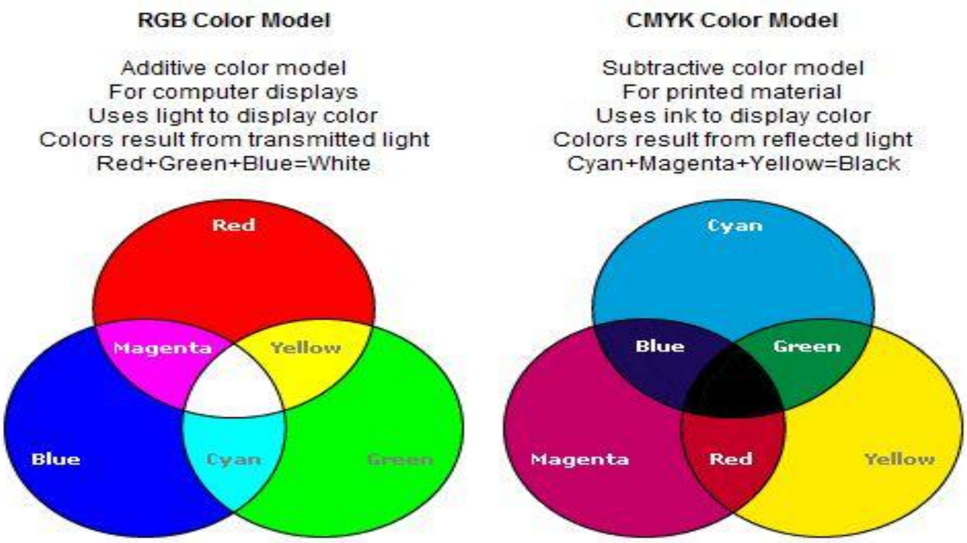

Fig 3.8: Color Rating

RGB Color vs CMYK Color ModelThe RGB color model isn't very intuitive, so h ere's a table containing some common RGB values:

Table: 3.2: Color rating

R	G	B	Hex Value	Color
0	0	0	000000	Black
255	0	0	FF0000	Red
0	255	0	00FF00	Green
0	0	255	0000FF	Blue
255	255	0	FFFF00	Yellow
255	0	255	FF00FF	Magenta
0	255	255	00FFFF	Cyan
255	128	128	FF8080	Bright Red
128	255	128	80FF80	Bright Green
128	128	255	8080FF	Bright Blue
64	64	64	404040	Dark Grey
128	128	128	808080	Intermediate Grey
192	192	192	C0C0C0	Bright Grey
255	255	255	FFFFFF	White

When you use ADC just scale this values for proper color representation.
By doing calculations on the RGB values of the pixels of an image you can perf orm various color effects. Here's a table of the operations you can do with RGB color. These operations are given for the 24-

bit color model with 8 bit per channel, so 255 is the maximum value of a color. Colors channels can also be represented as floating point numbers between 0.0 a nd 1.0, and then you have to replace the value "255" by "1.0". C represents the channel together or the total color, while R, G and B represent the Red, Green a nd Blue channel separately.

Table 3.3: RGB color model

Operation	**Formula**	**Effect**
Negative	255-C	Returns the opposite color, for instance Black becomes White , Red be
Darken	C/P or C-P	Divide the color though some constant(larger than 1) or subtract a co
Brighten	C*p or C+P	Multiply the color some constant (larger than 1), or add a constant to
Greyscale	(R+G+B)/3	Calculate the average of the 3 channels to get a grey color with the s
Remove Channel	R=0, G=0 and/ or B=0	By setting one or more channels to 0, you completely remove color fr
Swap Channels	R=G , G=R, …..	Swap the value of two color to get an image with a completely

3.2.4 Color Model Conversions

To draw the plots given above, color model conversion functions have to be used: first you describe the color as HSL or HSV, but to plot it on screen, it has to be converted to RGB first. Transformations from RGB to HSL/HSV are handy as well, for example if you load an RGB image and want to change it hue, you h ave to convert it to HSL or HSV first, then change the hue, and then change it back to RGB.

3.2.5 Display board

A display board is a board-
shaped material that is rigid and strong enough to stand on its own, and generally used paper or other materials affixed to it. Display board may also be referred t o as poster board.

3.2.6 Bread board

What is a breadboard:

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

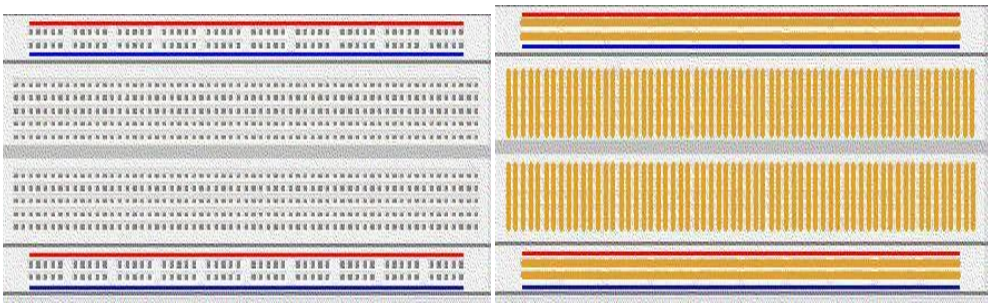

Fig 3.9: Breadboard

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 topand bottom rows of holes are connected horizontally while the remaining holes areconnected vertically.The long top and bottom row of holes are usually use d for power supply connections. The rest of the circuit is built by placing compo nents and connecting them together withjumper wires. ICs are placed in the middle of the board so that half of the legs are on oneside of the middle line and half on the other. To use the bread board, the legs of components are placed in the holes. Each set of holesconnected by a metal strip underneath forms a node. A node is a point in a circuit wheretwo components are connected. Connections betw een different components are formedby putting their legs in a common node.
A completed circuit might look like the following.

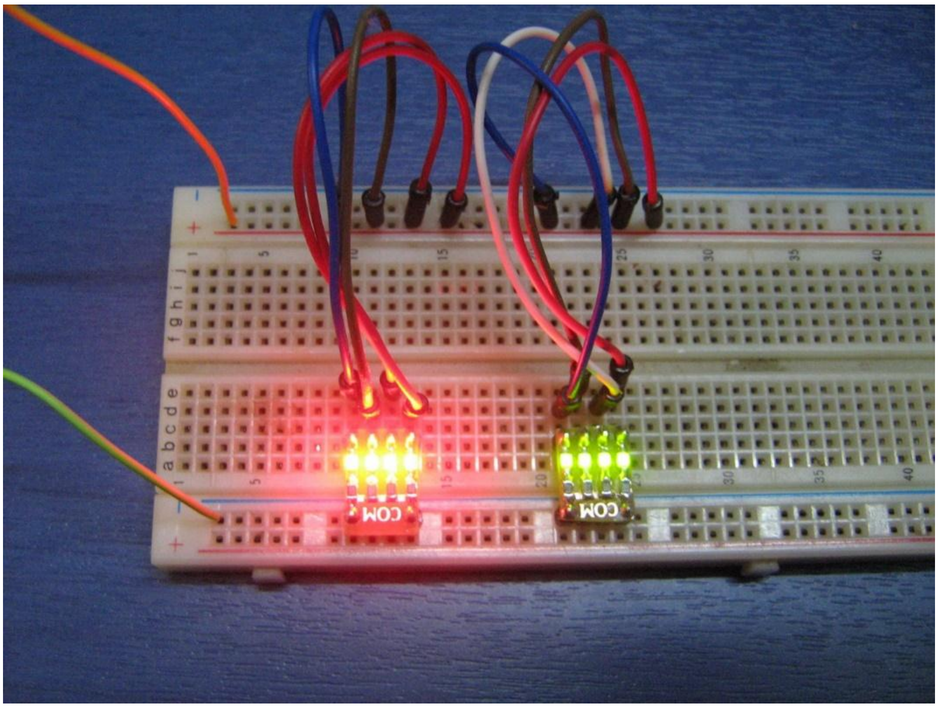

Fig 3.10: breadboard circuit

Breadboard tips:

It is important to breadboard a circuit neatly and systematically, so that one can d ebug itand get it running easily and quickly. It also helps when someone else nee ds tounderstand and inspect the circuit. Here are some tips:

1. Always use the side-lines for power supply connections. Power the chips from theside-lines and not directly from the power supply.
2. Use black wires for ground connections (0V), and red for other powerconnectio ns.
3. Route jumper wires around the chips and not over the circuit.

Which electronic parts are compatible with breadboards**:**

So, how do electronic components fit into a breadboard? Many electronic compone nts have long metal legs called leads (pronounced "leeds"). Sometimes, shorter met al legs are referred to as pins instead. Almost all components with leads will wor k with a breadboard (to learn more about these components and which types work with a breadboard, see the Advanced section).

Breadboards are designed so you can push these leads into the holes. They will b e held in place snugly enough that they will not fall out (even if you turn the b readboard upside-down), but lightly enough that you can easily pull on them to remove them.

Do I need any tools to use abreadboard?

You do not need any special tools to use a solderless breadboard. However, many electronic components are very tiny, and you may find them difficult to handle. A pair of miniature needle nose pliers or tweezers may make it easier to pick up small components.

3.2.7 Inside a Breadboard:

The leads can fit into the breadboard because the inside of a breadboard is made up of rows of tiny metal clips. This is what the clips look like when they are removed from a breadboard.

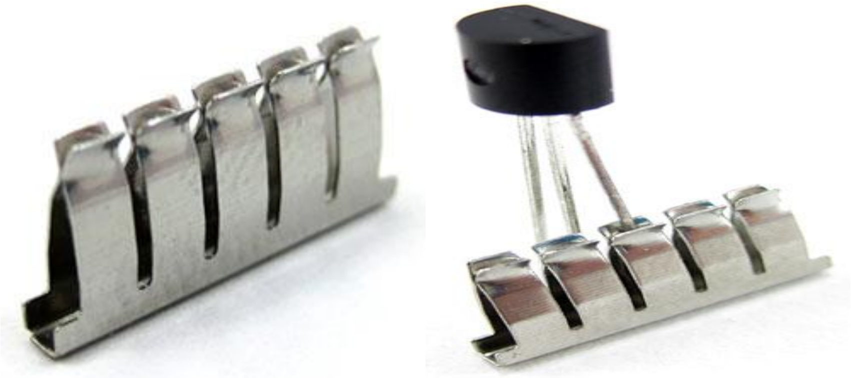

Fig 3.11:Inside a Breadboard

When you press a component's lead into a breadboard hole, one of these clips grabs onto it.Some breadboards are actually made of transparent plastic, so you can see the clips in side.

They work:

Most breadboards have a backing layer that prevents the metal clips from falling out. The backing is typically a layer of sticky, double-
sided tape covered by a protective layer of paper. If you want to permanently "sti ck" the breadboard to something (for example, a robot), you just need to peel off the paper layer to expose the sticky tape underneath. In this picture, the breadbo ard on the right has had its backing removed completely (so you can see all the metal clips). The breadboard on the left still has its sticky backing, with one corn er of the paper layer peeled up.

Fig 3.12: Paper layer

3.2.8 Breadboard labels: rows, columns, and buses

What do the letters and numbers on a breadboard means:

Most breadboards have some numbers, letters, and plus and minus signs written on them. What does all that mean? While their exact appearance might vary from b readboard to breadboard, the general purpose is always the same. These labels help you locate certain holes on the

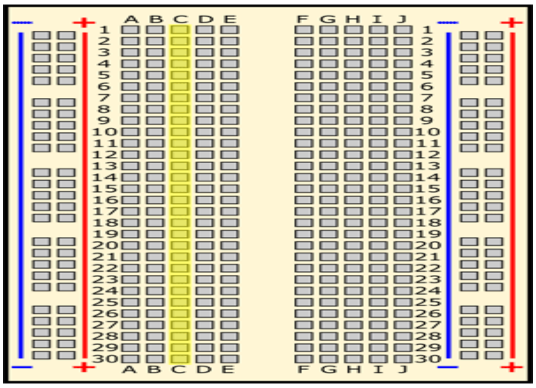

Fig 3.13: breadboard.

breadboard so you can follow directions when building a circuit. If you have ever used a spreadsheet program like Microsoft Excel® or Google Sheets™, the conce pt is exactly the same. Row numbers and column letters help you identify individ ual holes in the breadboard, just like cells in a spreadsheet. For example, all of t he highlighted holes are in "column C."

What do the colored lines and plus and minus signs mean:

What about the long strips on the side of the breadboard, highlighted in yellow here**.**

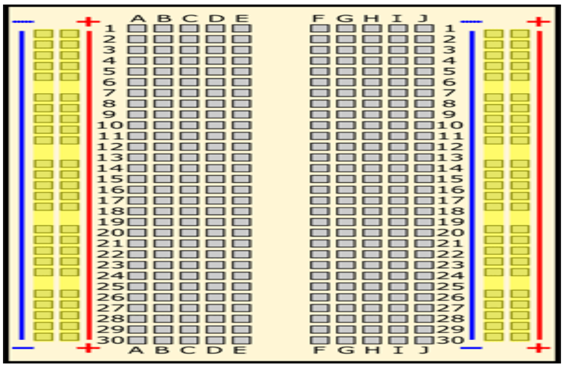

Fig 3.14: plus and minus signs

These strips are typically marked by red and blue (or red and black) lines, with plus (+) and minus (-

) signs, respectively. They are called the buses, also referred to as rails, and are t ypically used to supply electrical power to your circuit when you connect them to a battery pack or other external power supply. You may hear the buses referred to by different names; for example, power bus, positive bus, and voltage bus all r efer to the one next to the red line with the plus (+) sign. Similarly, negative bu s and ground bus both refer to one next to the blue (or black) line with the min us (-) sign. Sound confusing? Use this table to help you remember—

there are different ways to refer to the buses, but they all mean the same thing. Do not worry if you see them referred to by different names in different places (for example, in different Science Buddies projects or other places on the internet). Sometimes you might hear "power buses" (or rails) used to refer to both of the buses (or rails) together, not just the positive one.

Note that there is no physical difference between the positive and negative buses, and using them is not a requirement. The labels just make it easier to organize y our circuit, similar to color-coding your wires.

How are the holes connected:

Remember that the inside of the breadboard is made up of sets of five metal clip s. This means that each set of five holes forming a half-row (columns A–E or columns F–J) is electrically connected. For example, that means hole A1 is electrically connec ted to holes B1, C1, D1, and E1. It is not connected to hole A2, because that h ole is in a different row, with a separate set of metal clips. It is also not connec ted to holes F1, G1, H1, I1, or J1, because they are on the other "half" of the breadboard—

the clips are not connected across the gap in the middle (to learn about the gap in the middle of the breadboard, see the Advanced section). Unlike all the main b readboard rows, which are connected in sets of five holes, the buses typically run the entire length of the breadboard (but there are some exceptions). This image s hows which holes are electrically connected in a typical half-sized breadboard, highlighted in yellow lines

Buses on opposite sides of the breadboard are not connected to each other. Typica lly, to make power and ground available on both sides of the breadboard, you wo uld connect the buses with jumper wires, like this. Make sure to connect positive to positive and negative to negative (see the section on buses if you need a remi nder about which color is which).

of how they are labeled and the left/right positions, the function of the buses rem ains the same.

3.2.9Using a breadboard

What are jumper wires and what kind should I Use:

Jumper wires are wires that are used to make connections on a breadboard. They have stiff ends that are easy to push into the breadboard holes. There are several different options available when purchasing jumper wires.This makes it easy to co lor-code your circuit (see the section on color-coding). While these wires are easy to use for beginner circuits, they can get very messy for more complicated circuits; because they are so long, you will wind up with a tangled nest of wires that are hard to trace (sometimes called a "rat's nes t" or "spaghetti").Flexible jumper wires are made of a flexible wire with a rigid p in attached to both ends. These wires usually come in packs of varying colors

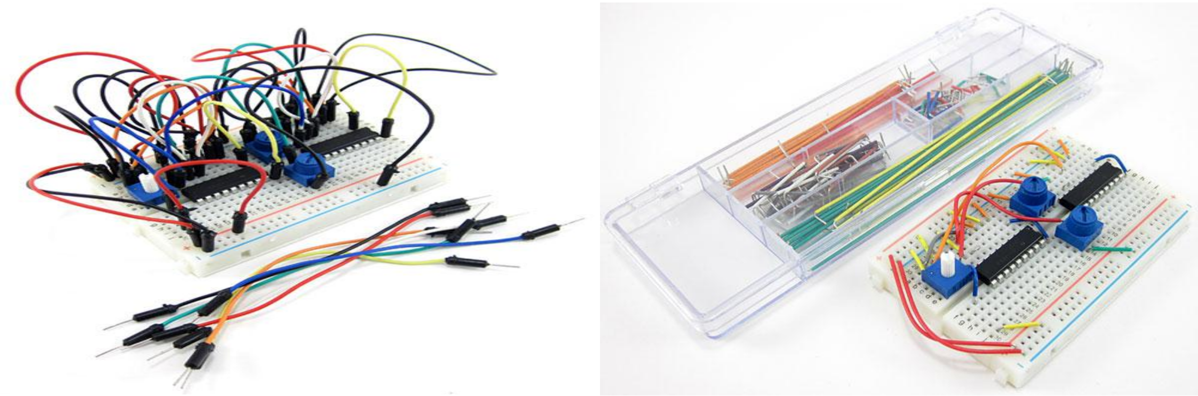

Fig 3.15: Flexible jumper wires

Finally, you can also buy spools of solid-core hookup wire and a pair of wire strippers and cut your own jumper wires. T his is the best long-term option if you plan on doing lots of electronics projects, because you can cut wires to the exact length you need, and pick which colors you want. It is also much more cost-effective per length of wire. Buying a kit of six different colors is a good place to start.

How do I build a circuit:

Follow the breadboard diagram for the circuit, connecting one component at a time .

- Always connect the batteries or power supply to your circuit last. This will give you a chance to doublecheck all your connections before you turn your circuit on for the first time.

- Keep an eye out for common mistakes that many beginners make when using a breadboard.

If your circuit does not work, you need to troubleshoot (or debug, meaning to look for problems or "bugs" in your circuit). See the common mistakes section for things you should check.

Follow the project directions to use your circuit (for example, shining a flashlight at a light-tracking robot, or waving your hand in front of a motion sensor).

3.2.10 Integrated circuits (ICs)

Integrated circuits, or ICs for short (sometimes just referred to as "chips") are specialized circuits that serve a huge variety of purposes, such as controlling a robot's motors or making LEDs respond to music. Many ICs come in something called a dual in-line package, or DIP, meaning they have two parallel rows of pins. The gap in the middle of a breadboard (between columns E and F) is just the right width for an IC to fit, straddling the gap, with one set of pins in column E, and one set of pins in column F. Projects that use ICs will always tell you to connect them to the breadboard in this manner.

3.2.11Wire

This library allows you to communicate with I2C / TWI devices. On the Arduino boards with the R3 layout (1.0 pinout), the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. The Arduino Due has two I2C / TWI interfaces SDA1 and SCL1 are near to the AREF pin and the additional one is on pins 20 and 21.

As a reference the table below shows where TWI pins are located on various Arduino boards.

Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)

Due	20 (SDA), 21 (SCL), SDA1, SCL1

As of Arduino 1.0, the library inherits from the Stream functions, making it consistent with other read/write libraries. Because of this, send() and receive() have been replaced with read() and write().

Examples

Digital Potentiometer: Control an Analog Devices AD5171 Digital Potentiometer.

Master Reader/Slave Writer: Program two Arduino boards to communicate with one another in a Master Reader/Slave Sender configuration via the I2C.

Master Writer/Slave receiver: Program two Arduino boards to communicate with one another in a Master Writer/Slave Receiver configuration via the I2C.

3.3 Summary

In this chapter we know how to using the Arduino Uno board, historical background of Arduino Uno. The working procedure of color sensing meter and light. How much pins at Arduino Uno. About and work at display board. Breadboard using procedure and why using breadboard. How to build integrated circuit.

CHAPTER 4

HARDWARE DEVELOPMENT

4.1 Introduction:

The Arduino is a open source microcontroller.The Arduino IDE incredibly minimalistic,for most Arduino-based project yet it provides a near-complete environment for most Arduino-based project.

4.2 Arduino IDE

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedure Sketches are saved on the development computer as text files with the file extension Arduino. Arduino Software (IDE) pre-1.0 saved sketches with the extension. A program written with the IDE for Arduino is called a sketch. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic ind enting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The Arduino IDE employs the program to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

4.3 Arduino program

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consists of only two functions:

setup(): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

loop(): After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

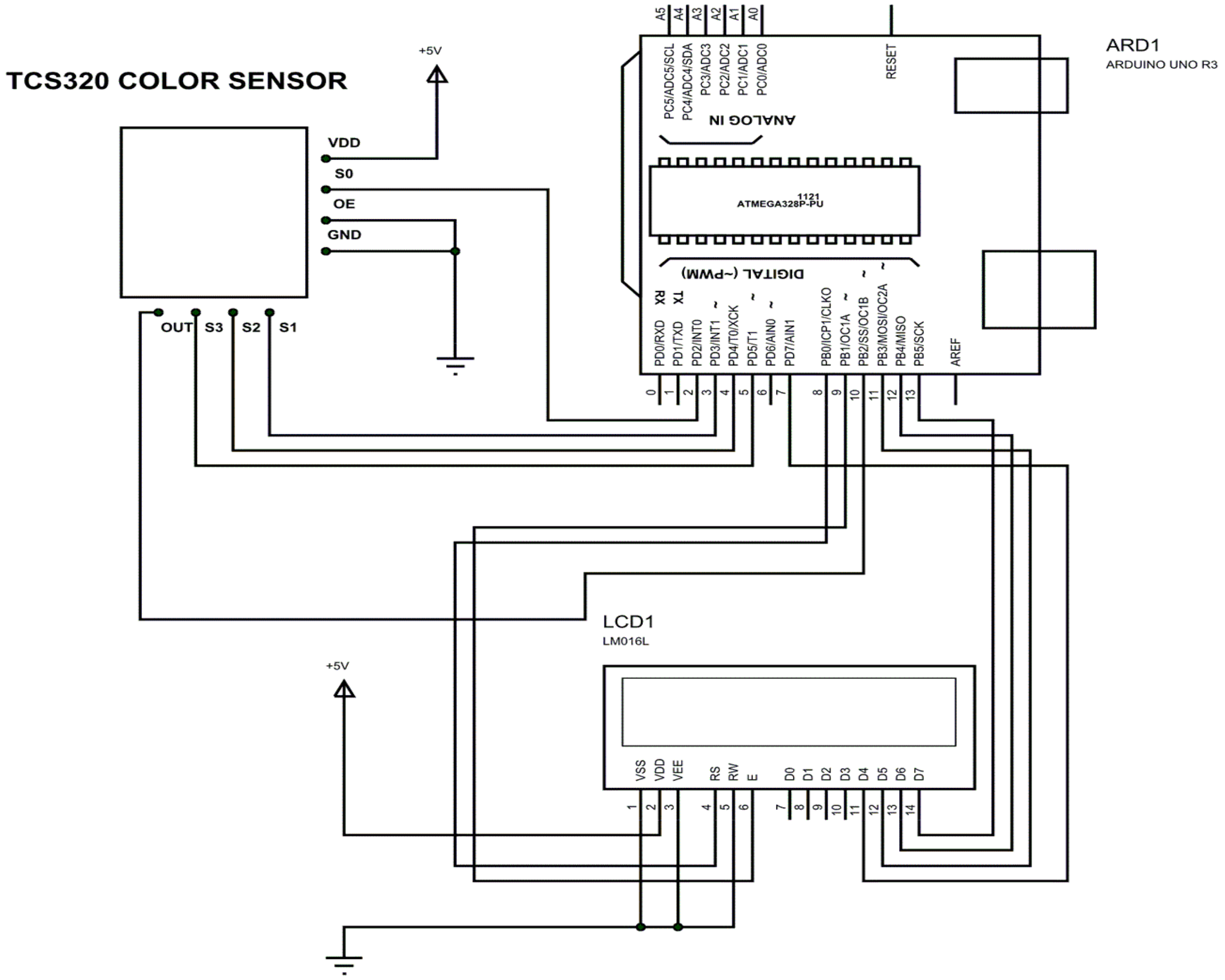

Fig4.1: Circuit diagram of full system

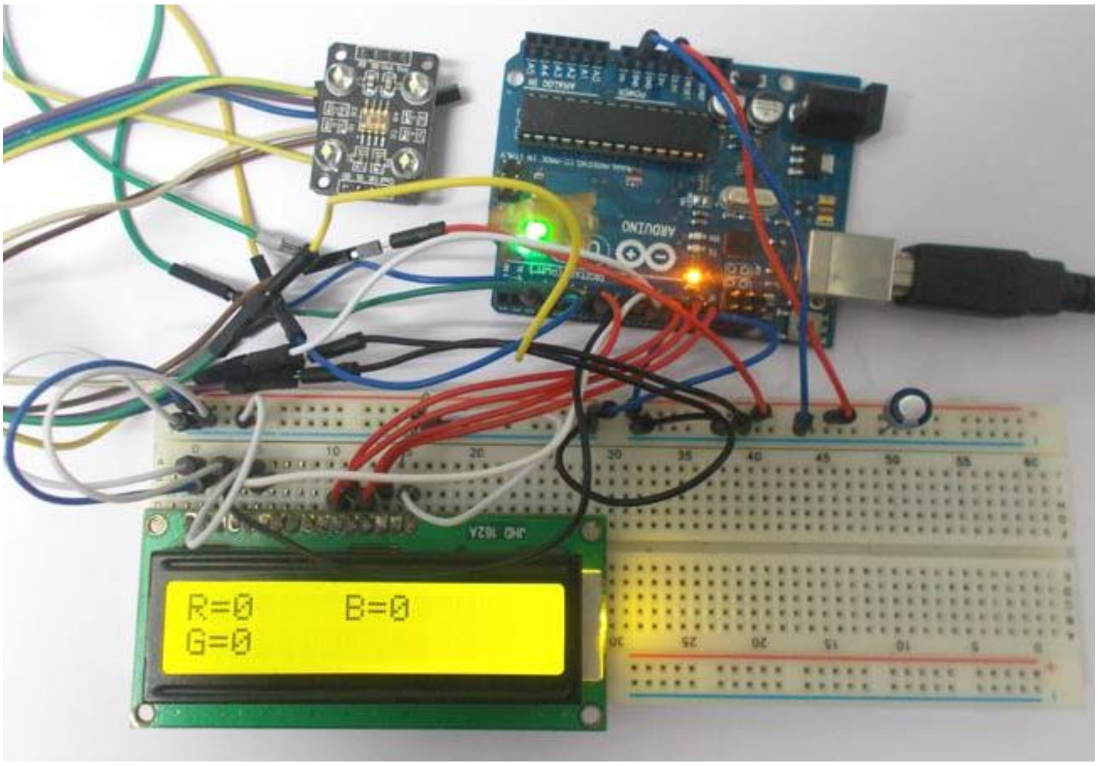

Fig 4.2: Complete hardware setup

4.4 Summary

A circuit diagram like as electrical diagram, elementary diagram, electronic schematic is a graphical representation of an electrical circuit. A pictorial circuit diagram uses simple images of components, while a schematic diagram shows the components and interconnections of the circuit standardization symbolic representation. The presentation of the interconnections circuit components in the schematic diagram does not necessarily correspond to the physical arrangements in the finished device.

CHAPTER 5

RESULT AND DISCUSSION

5.1 Introduction

The system was powered and operated upon using several possibilities they include making sure that the color detect only the color correct. There are three color in put in Arduino. When these three color shows at front the color sensor module th en display shows these color. This chapter will present all the results and calculati ons and relevant discussions.

5.2 Results

A result is the final consequence of a sequence of actions or events expressed qu alitatively or quantitatively. Possible results include advantages, disadvantage, gain, i njury, loss, value, and victory. There may be a range of possible outcomes associa ted. Reaching no result can mean that actions are inefficient, ineffective**.**

5.2.1 Design a color detecting system

There are three color coding which is made by using ATMega238 microcontroller circuit. These are a Arduino, LCD display, breadboard and a color sensor module. Here an ATMega328 microcontroller used which requires +5V DC voltage supply. **Resetting the Circuit Design:**When the power supply is within the range as well as oscillation level is minimum, the reset pin of microcontroller must be kept act ive**.**

Compilation of Arduino Coding: After designing the circuit then code needed. To write the code, we have used Arduino software. The using programming Langu age is C+. After writing the coding it needs to be saved with extension**.**

5.2.2 Step by Step pro tecting System

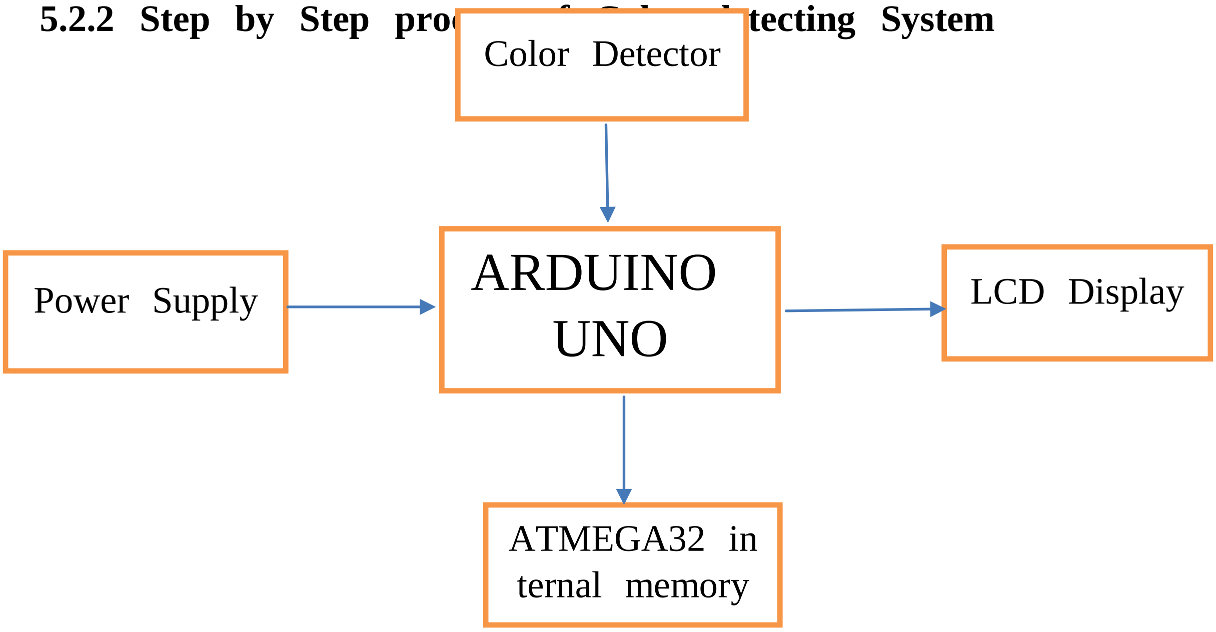

Fig 5.1: Block diagram of full system

a) Started power supply at Arduino UNO, LCD and color sensor module initialize d.

b) Now show the color at color sensor module.

c)When Red color shows, the display show the color detected at Red.

d)Same process at the color of Blue and Green.

e) If the user show the module at black color then the display does not show th e color, because we are using coding at RED, GREEN, and RED color only.

5.3Advantage

* This project detect color.
* Power consumption less.
* Used commonly available components.
* It is simple and easy.

5.4 Summary

Basically it is small and easy to making. We are trying to easily detect the color at using this project. The conventional controllers in market mostly using this mi crocontroller and increasing the cost.

CHAPTER 6

CONCLUSIONS

6.1 Introduction:

Color sensing is one of the important subjects of optical sensors. Chemical sensing is mostly implemented with particular emphasis at color detection based on colori metric sensors because many parameters, likeconcentration like pH, and chemical ga ses can cause direct or indirect color changing in biological and chemical species. Color sensors have a variety of applications including detection of, biological,enviro nmental and chemical parameters.

6.2 Limitations of the work

CMYK system is complicated for this project. It is low range circuit that is not possible to operate remotely. Sometimes the display does not show the detect col or.

6.3 Future Scope of Work

We can send this data to a remote location using mobile detect the color. In futu re the Robotics competition we using this project at the head of that robot, then robots find the only color that using the coding at Arduino. Garments and textile industry needed for color detecting.

6.4 Conclusions:

A low-
cost optical sensor based on reflective color sensing is presented. Moreover, some

neural network models are used as artificial intelligent technique to improve the co lor regeneration from the sensor signals. That is to say, analog voltages of the se nsor can be successfully converted to RGB colors. The artificial intelligent models presented in this work enable color regeneration from analog outputs of the color sensor. Besides, an inverse modeling supported by an intelligent technique enables the sensor probe for use of a colorimetric sensor that relates color changes to an alog voltages.

REFERENCES

[1] https://www.sciencebuddhttp://home.roboticlab.eu/en/examples/sensor/color

[2] https://en.wikipedia.org/wiki/Display_boardies.org/science-fair-projects/references/how-to-use-a-breadboard

[3] http://what-is-a-sensor.com/what-are-color-sensors/

[4] https://www.thephoblographer.com/2013/07/31/an-introduction-to-and-brief-history-of-digital-imaging-sensor-technologies/

[5] http://www.electronics-base.com/general-description/measurement/175-color-sensor-theory-for-dummies.

APPENDIX

```
#include<LiquidCrystal.h>
LiquidCrystallcd(13,12,11,10,9,8);
#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensor 6
int frequency = 0;
intred_filter = 0;
intgreen_filter = 0;
intblue_filter = 0;
 void setup() {
pinMode(S0, OUTPUT);
 //defining mode pin
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(sensor, INPUT);

pinMode(13, OUTPUT);
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);
pinMode(10, OUTPUT);
pinMode(9, OUTPUT);
pinMode(8, OUTPUT);
 // Setting scaling to 20%

digitalWrite(S0, LOW);
digitalWrite(S1, HIGH);
```

```
Serial.begin(9600);
lcd.begin(16,2);

lcd.clear();
lcd.print("COLOR SENSOR");
lcd.setCursor(0,2);
lcd.print("INTERFACING BY:");
delay(2000);

lcd.clear();
lcd.print("CREATIVE");
lcd.setCursor(8,2);
lcd.print("ENGINEER");
delay(2000);
}

void loop() {
// Reading red filter first
digitalWrite(S2,LOW);
digitalWrite(S3,LOW);
  frequency = pulseIn(sensor,LOW);
red_filter = map(frequency,0,10000,255,0);
  // mapping frequency for red color
Serial.print("R= ");
Serial.print(red_filter);
delay(100);

  // reading green filter
digitalWrite(S2,HIGH);
digitalWrite(S3,HIGH);

  frequency = pulseIn(sensor, LOW);
green_filter = map(frequency,0,10000,255,0); //green
  //frequency mapping.
```

```
Serial.print("  G=  ");
Serial.print(green_filter);
delay(100);

  //  reading  blue  filter
digitalWrite(S2,LOW);
digitalWrite(S3,HIGH);

  frequency  =  pulseIn(sensor,  LOW);
blue_filter  =  map(frequency,0,10000,255,0);
  //blue  frequency  mapping.

Serial.print("  B=  ");
Serial.println(blue_filter);
delay(500);
if(red_filter>=blue_filter&&red_filter>=green_filter  )
{
lcd.clear();
lcd.print("COLOR  DETECTED");
lcd.setCursor(0,2);
lcd.print("RED");
delay(500);
}

else  if  (green_filter>=blue_filter&&green_filter>=  red_filter)
{
lcd.clear();
lcd.print("COLOR  DETECTED");
lcd.setCursor(0,2);
lcd.print("GREEN");
delay(500);
}
```

```
else if (blue_filter>= red_filter&&blue_filter>= green_filter)
{
lcd.clear();
lcd.print("COLOR DETECTED");
lcd.setCursor(0,2);
lcd.print("BLUE");
delay(500);
}

else

{
lcd.clear();
lcd.print("COLOR DETECTED");
lcd.setCursor(0,2);
lcd.print("NIL");
delay(500);
}
```