**COLOR DETECTING ROBOT**

by

D. Shashank - 13BCE1036

Harshwardhan - 13BCE1052

S. Nikesh Kumar Reddy - 13BCE1136

*A project report submitted to the*

**SCHOOL OF COMPUTING SCIENCES ENGINEERING**



**VIT UNIVERSITY, CHENNAI**

**Vandalur – Kelambakkam Road**

**Chennai – 600 127.**

**MAY 2015**

**ACKNOWLEDGEMENT**

**Course Code: CSE221 Batch no: 18**

This project report entitled “**COLOR DETECTING ROBOT”** is a bonafide work of **SHASHANK. D (13BCE1036), HARSHWARDHAN AGARWAL(13BCE1052) and NIKESH. S (13BCE1136)** carried out the Project work under the supervision and guidance of **Prof. RAJESH KUMAR.** We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Prof.** **RAJESH KUMAR,** Associate Professor, School of Computing Sciences and Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work

**TABLE OF CONTENTS**

1. Abstract
2. Color sensor
3. Arduino
4. Working and Programming in Arduino
5. Advantages of Arduino
6. Circuit Diagram
7. Components Required
8. Arduino Code
9. Difficulty and learning.

**ABSTRACT**

One of the most wonderful creations by human is the automaton and its derivation, robot. Automatizing things give us many benefits such as accuracy, time saver, specifications and what not. One such robot is a color picking robot. It has a wide range of applications in day to day life. For an instance, A car which has an automatic driving/ auto pilot mode in it need to detect color in order to change its mode during the signals, A device separator in a product manufacturing factory need to have a color picking robot(expanded with additional features) to separate things etc.

**COLOR SENSOR**

A crucial part of color picking robot is sensing/detecting the color. We could use a sensor for the purpose but by using a sensor, there is no purpose of doing the project because if we have the sensor, we just need to assemble things. So we need to build a sensor and detect colors using it.

We perform this operation using an Arduino.

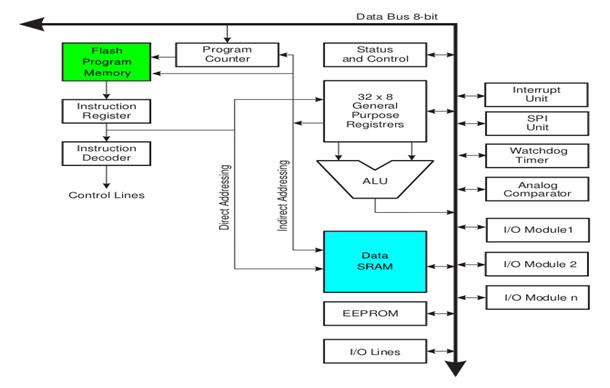
**ARDUINO**

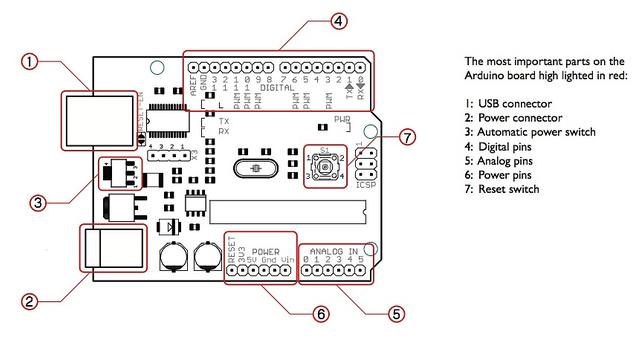
It is an open source hardware, software and user interface that can be used for designing/manufacturing interactive objects which can sense or control the physical world.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits.

**Working of Arduino:**

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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.





**Programming in an Arduino:**

The most important advantage with Arduino is the programs can be directly loaded to the device without requiring any hardware programmer to burn the program. This is done because of the presence of the 0.5KB of Bootloader which allows the program to be burned into the circuit. All we have to do is to download the Arduino software and writing the code.

* Programs written in Arduino are known as sketches. A basic sketch consists of 3 parts

1. Declaration of Variables  
 2. Initialization: It is written in the setup () function.  
 3. Control code: It is written in the loop () function.

* The sketch is saved with .ino extension. Any operations like verifying, opening a sketch, saving a sketch can be done using the buttons on the toolbar or using the tool menu.
* The sketch should be stored in the sketchbook directory.
* Chose the proper board from the tools menu and the serial port numbers.
* Click on the upload button or chose upload from the tools menu. Thus the code is uploaded by the bootloader onto the microcontroller.

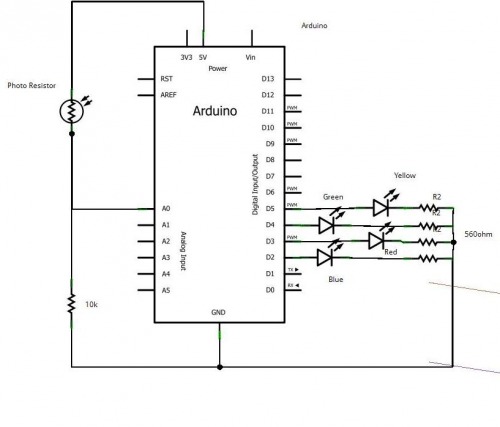
**Advantages of Arduino:**

* It is inexpensive
* It comes with an open source hardware feature which enables users to develop their own kit using already available one as a reference source.
* The Arduino software is compatible with all types of operating systems like Windows, Linux, and Macintosh etc.
* It also comes with open source software feature which enables experienced software developers to use the Arduino code to merge with the existing programming language libraries and can be extended and modified.
* It is easy to use for beginners.
* We can develop an Arduino based project which can be completely stand alone or projects which involve direct communication with the software loaded in the computer.
* It comes with an easy provision of connecting with the CPU of the computer using serial communication over USB as it contains built in power and reset circuitry.

**Arduino Specifications:**

* 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
* Length 68.6 mm
* Width 53.4 mm
* Weight 25 g

**SCHEMATIC CIRCUIT DIAGRAM**

****

**COMPONENTS REQUIRED**

1. Photo resistor
2. 10k ohms resistors
3. Arduino
4. Connecting wires
5. Led (blue, red, green, yellow)
6. Breadboard

**ARDUINO CODE**

int photoResistor = 0;

int b = 2;

int r = 3;

int g = 4;

int y = 5;

double bAdj = 1.0;

double rAdj = 1.02;

double gAdj = 1.0;

double yAdj = 1.17;

double ambientAdj = 1.0;

void setup()

{

pinMode(g, OUTPUT);

pinMode(y, OUTPUT);

pinMode(b, OUTPUT);

pinMode(r, OUTPUT);

pinMode(photoResistor, OUTPUT);

Serial.begin(9600);

}

void loop()

{

allOff();

delay(100);

asses(100,100);

}

void allOff()

{

digitalWrite(g, LOW);

digitalWrite(y, LOW);

digitalWrite(r, LOW);

digitalWrite(b, LOW);

}

void allOn()

{

digitalWrite(g, HIGH);

digitalWrite(y, HIGH);

digitalWrite(r, HIGH);

digitalWrite(b, HIGH);

}

void pulse(int colour)

{

for(int i=0;i<3;i++)

{

digitalWrite(colour, HIGH);

delay(100);

digitalWrite(colour, LOW);

delay(100);

}

}

void asses(int colldownTime, int saturationTime)

{

delay(500);

double ambient = analogRead(photoResistor);

Serial.print("Ambient: ");

Serial.println(ambient\*ambientAdj);

digitalWrite(g, HIGH);

delay(saturationTime);

double green = analogRead(photoResistor);

Serial.print("Green: ");

Serial.println(green\*gAdj);

digitalWrite(g, LOW);

delay(colldownTime);

digitalWrite(r, HIGH);

delay(saturationTime);

double red = analogRead(photoResistor);

Serial.print("Red: ");

Serial.println(red\*rAdj);

digitalWrite(r, LOW);

delay(colldownTime);

digitalWrite(b, HIGH);

delay(saturationTime);

double blue = analogRead(photoResistor);

Serial.print("Blue: ");

Serial.println(blue\*bAdj);

digitalWrite(b, LOW);

delay(colldownTime);

digitalWrite(y, HIGH);

delay(saturationTime);

double yellow = analogRead(photoResistor);

Serial.print("Yellow: ");

Serial.println(yellow\*yAdj);

digitalWrite(y, LOW);

delay(colldownTime);

double rawData[] = {(green\*gAdj), (red\*rAdj), (blue\*bAdj), (yellow\*yAdj)};

double maximum = ambient\*ambientAdj;

int decision;

for (int i=0; i<4; i++)

{

if (maximum<rawData[i])

{

maximum = rawData[i];

decision = i+1;

}

}

Serial.println("");

if(decision == 0)

{

Serial.println("No object detected.");

allOn();

delay(1000);

allOff();

}

else if(decision == 1)

{

Serial.println("I think its green.");

pulse(g);

}

else if(decision == 2)

{

Serial.println("Looks red to me.");

pulse(r);

}

else if(decision == 3)

{

Serial.println("Blue for sure!");

pulse(b);

}

else if(decision == 4)

{

Serial.println("None of the three");

pulse(y);

}

Serial.println("");

}

**DIFFICULTY AND LEARNING**:

Understanding the concept of a photoresistor in how it can be used to detect colors by taking light as an input was a bit time taking.

Learning arduino programing was a new experience for us.

Setting up the hardware: soldering and de-soldering consumed time.

[overall had a good learning experience by this project]