A Project Report on

Autonomous Fire Extinguishing Bot

Submitted By:
Shail Shah (13012041110)
Mayur Kagathara (13012041040)
Dhruv Bhavsar (13012041009)

Mechatronics Department
U. V. PATEL COLLEGE OF ENGINEERING
GANPAT UNIVERSITY
GANPAT VIDYANAGAR - 384012

2016-2017

A Project Report on

Autonomous Fire Extinguishing Bot

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

MECHATRONICS ENGINEERING

Submitted By:
Shail Shah (13012041110)
Mayur Kagathara (13012041040)
Dhruv Bhavsar (13012041009)

Guided By

Prof J.K.Prajapati
Assistant Professor

Mechatronics Department
U. V. PATEL COLLEGE OF ENGINEERING
GANPAT UNIVERSITY
GANPAT VIDYANAGAR - 384012

2016-2017

Department of Mechatronics Engineering
U. V. Patel College of Engineering
Ganpat University

CERTIFICATE

This is to certify that the work of User Defined Project entitled **"Autonomous Fire Extinguishing Bot"** has been carried out by **Shail Shah** ***(13012041110)*** in regards to the partial fulfillment for the requirement for the award of the degree of Bachelor of Technology in Mechatronics Engineering during the academic session 2016-2017 is a record of bona fide work carried out by him at the Department of Mechatronics Engineering, U. V. Patel College of Engineering, Ganpat University under my guidance and supervision. The work done here is a reflection of originality and transparency. The work incorporated in this thesis has not been, to the best of my knowledge, submitted to any other University or Institute for the award of any degree or diploma.

Prof. J.K.Prajapati,
Assistant Professor
Mechatronics Department
U V Patel College of Engineering

Prof. J.P.Patel,
Head of Department,
Mechatronics Department
U V Patel College of Engineering

Internal Examiner: _____

External Examiner: _____

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. J.K.Prajapati for their guidance and constant supervision as well as for providing necessary information regarding the project. We would also like to thank Prof T.R.Shah for his valuable inputs and suggestions regarding the project. We take this opportunity to thank all our friends and colleagues who started us out on the topic and provided extremely useful review feedback and for their all-time support and help in each and every aspect of the course of my project preparation. We are grateful to our college U. V. Patel College of Engineering, Ganpat University for providing us all required resources and good working environment.

I would like to express my gratitude towards Head of Department, **Prof J.P.Patel** for their kind co-operation and encouragement which helped us in this project.

Shail Shah (13012041110)

Mayur Kagathara (13012041040)

Dhruv Bhavsar (13012041009)

ABSTRACT

Fire departments in India always face difficulties to reach the fire location in stipulated time due to lack of information regarding congestion condition of the roads that connects to the location. This project is an alternative solution to overcome the problem by developing vision based fire recognition system. The system is operated on matlab environment for image processing and uses wireless communication to send the location of the fire to a bot stationed at a distance. The bot then travels on a predefined path to reach and extinguish the fire. This project aims to lessen the response time to fire incidents than conventional fire fighters hence reducing the casualties and property damage.

Contents

List of Figures

List of Table

1. Introduction

1.1 Overview of Project

Fire safety has always been an issue to tackle especially in country like india where the loss due to fire is to the tune of One lakh crore per annum and constitutes 0.8-1% of GDP. Autonomous fire extinguishing bot aims to provide fire safety using image processing as its core concept. Image Processing is processing of images using mathematical operations by using any form of signal processing for which the input is an image, a series of images, or a video, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image.

Entire project is focused on the fire detection using digital image processing. In achieving the goal, the project is divided into three subgroups which are:
(1) Machine Vision
(2) Electronics Circuit
(3) Mechanical design

Machine vision is prime component of the project. Here the image processing is done using MATLAB software.Image Processing is implemented because the fire detection task requires fire identification and thereby guiding the bot to reach the fire spot from its rest position to the desired position.

Electronics Circuit is based on open source environment Arduino Uno which is interfaced with Bluetooth module and servo motor. Entire bot mobility and fire extinguisher function using servo motor is controlled by programming done in the arduino IDE.

1.2 Scope of Project

The fundamental idea of extinguishing fire autonomously has vast scope in various fields. The fire that cannot by controlled by human intervention due to its size, form or nature can easily be extinguished by a fire extinguishing bot. This application will be of great help to the data centers and server rooms where the machines are constantly running and may possess a chance of fire.

1.3 Purpose of Project

The purpose of the project is to develop a prototype of an autonomous system that effectively tackles the fire of any form and provides primary assistance to the people trapped in fire. It also focuses on detection of fire through digital image processing, which makes it highly accurate, and extinguish all types of fire with a faster response rate than conventional fire fighters.

1.4 Literature review

The literature we reviewed was a case study based on analysis of four major fire outbreaks in the country which was published in International Journal of Engineering Trends and Technology (IJETT) and the result was based on the number of casualties and cause of fire for all four fire incidents. The failure aspect of all four fire incidents had a common factor which played a pivotal role in taking up this project. The failure aspects were common in all four cases and that was obsolete equipment being used by fire department and lack of active fire safety provisions.

Another literature was “Fire Recognition Using RGB and YCBCR Color Space”, which focused on image processing of the image captured and conversion of RGB image to YCBCR color space for fire detection. It also states the methodology of fire acquisition and image processing which helped us initially in getting a basic idea of image processing.

2. System Design

2.1 Objective

The prime objective of the project is to extinguish the fire. The project also aims to provide effective fire preventive and firefighting measures to life and property of public by reaching the fire spot with a faster response rate than conventional fire fighters. It also concentrates on making the fire extinguishing technique target specific rather than area specific which means that the surrounding of the fire gets less affected. Fire preventive action such as controlling fire before it takes a bigger form is difficult as fire is very volatile and needs to be under control within short span of time. The bot being compact it can access areas where a firefighter might not be able to reach and can give a glimpse of the situation remotely by sending the live video feed to the computer

2.2 Block Diagram

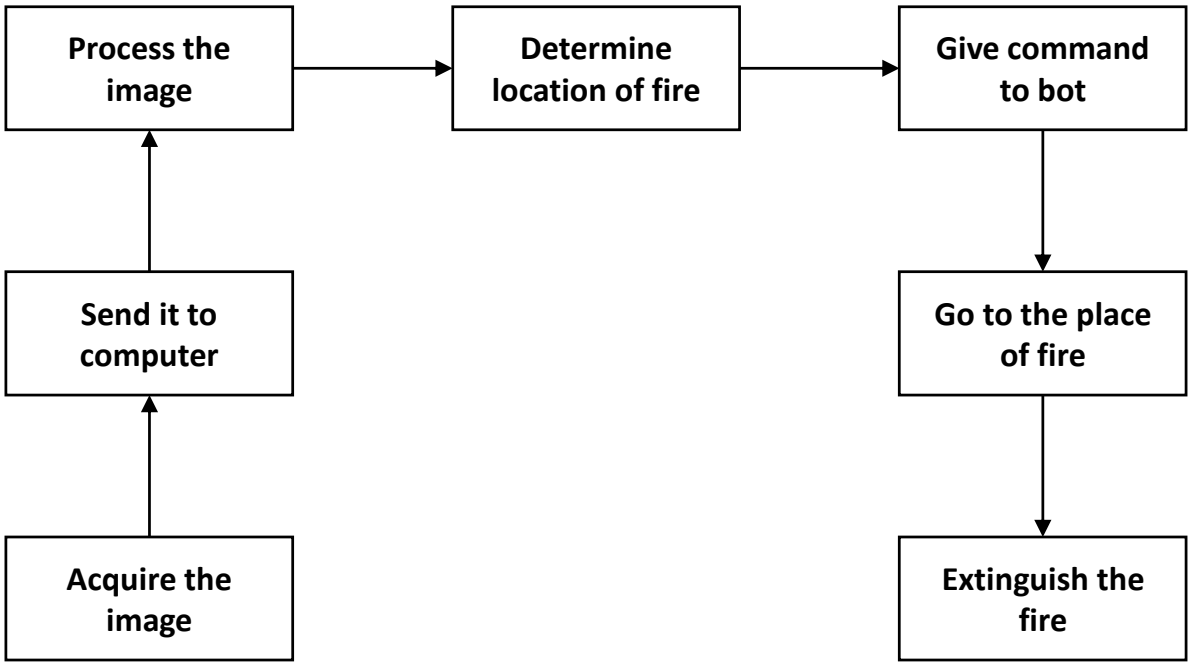

Figure 2.1 Block Diagram

2.3 Working

The entire working depends on two major parts of the project,
(1) Image processing
(2) Bot handling

The project focuses on detecting fire in the arena and the first step to detecting fire is acquiring the image. Image acquisition is possible by various means i.e. by Wired USB webcam or IP cam. Here we are using IP cam which will transmit the live feed to the computer for processing. Image acquired has to be processed in such a manner that it fulfils the requirement of fire detection. Image can be processed by different processing tools like matlab or python. Matlab IDE is the perfect processing tool for processing the acquired image as it has all the inbuilt functions already stored in its library for use. It is also compatible to work with other controllers like arduino and hence is used in the project.

The result from image processing is then sent to the bot via Bluetooth. Bot has controller which is interfaced to matlab. Matlab gives command to bot to go to target area. The bot utilizes the data to drive the motors in the direction of fire. Bot has extinguishing mechanism which is operated by matlab. The bot reaches the fire spot and the servo motor attached to it is given signal by matlab which is passed on to it by arduino and while facing the fire the servo mechanism presses the nozzle of the extinguisher. This extinguishes the fire.

The bot then returns to the initial position as programmed and waits for the next command from matlab.

2.4 Working environment

Arena consists of several stations which are supposed to be tested for fire detection. Bot is at particular place which is well defined and returns to this same position once the fire detection and extinguishing task is completed. Camera is steady and at some height to get the full picture of arena and wirelessly sends the image of the arena to the computer. Bot gets the command to go to the station under fire and the location is sent through computer and it reaches the spot and extinguishes the fire and returns back to its specified position.

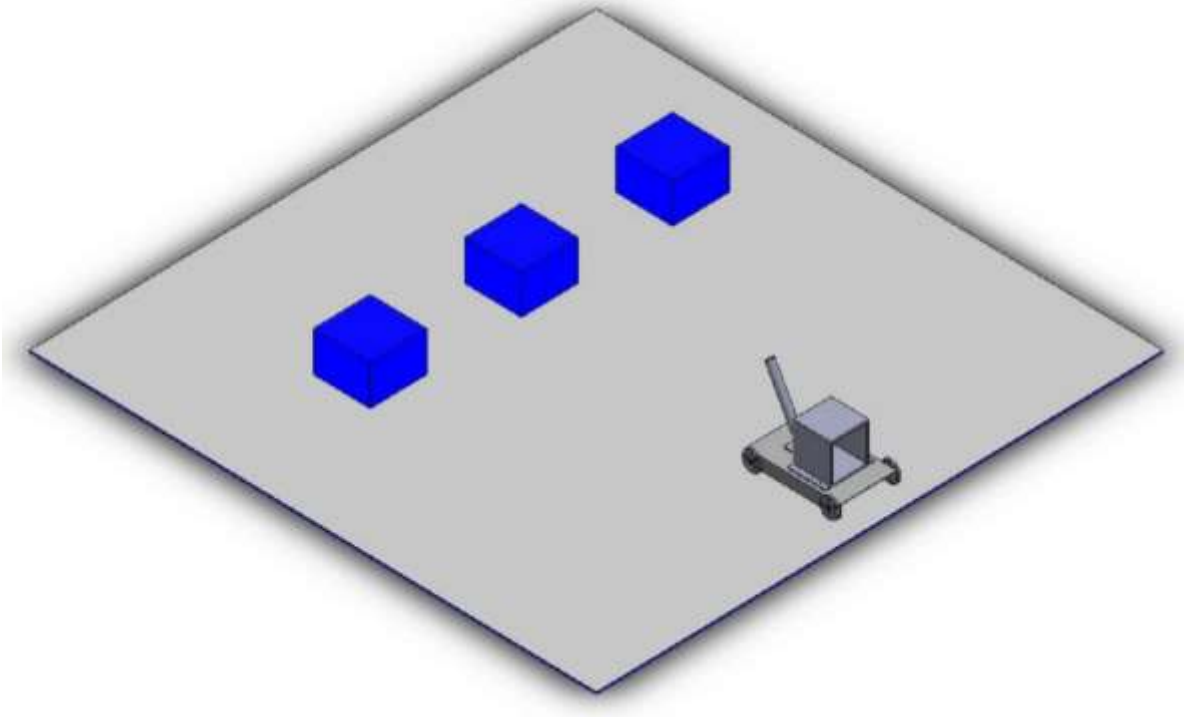

Figure 2.2 Arena and Bot arrangement

2.5 Fire detecting algorithm

The system is operated on MATLAB environment and based on a set of rules developed to identify the value of R, G, B, Y, Cb and Cr component in an image. The performance of the system is verified using plenty of images of various kind of brightness and environment. Experimental results show that the both colour spaces, RGB and YCbCr have different fire detection rates.

YCbCr colour space shows the best performance compared to RGB because it can separate luminance from chrominance more effectively than RGB colour space. Luminance in image is actually a light intensity or the amount of light ranges from black to white. While chrominance is a light wave with color Cyan Red and Cyan Blue.

Seven rules to detect fire as given in paper 'Fire Recognition Using Rgb And Ycbcr Color Space-ISSN 1819-6608' are as follow.

Colour space	**Rules**
RGB	1. R > G > B
	2. R>Rmean & G>Gmean & B<Bmean
YCbCr	1. Y ≥ Cb
	2. Cr ≥ Cb
	3. Y≥Ymean & Cb≤Cbmean & Cr≥Crmean
	4. Cb-Cr≥Th
	5. Cb≤100 & Cr≥150

Table 2.1 Rules to detect fire

Above rules are optimized and two rules were deduced as follows,

1) Y ≥ Cb & Cr ≥ Cb
2) Y ≥ 1.5*Ymean & Cb ≤ 0.8*Cbmean & Cr ≥ 1.1*Crmean

Testing of rules:

1) Image of a burning car.

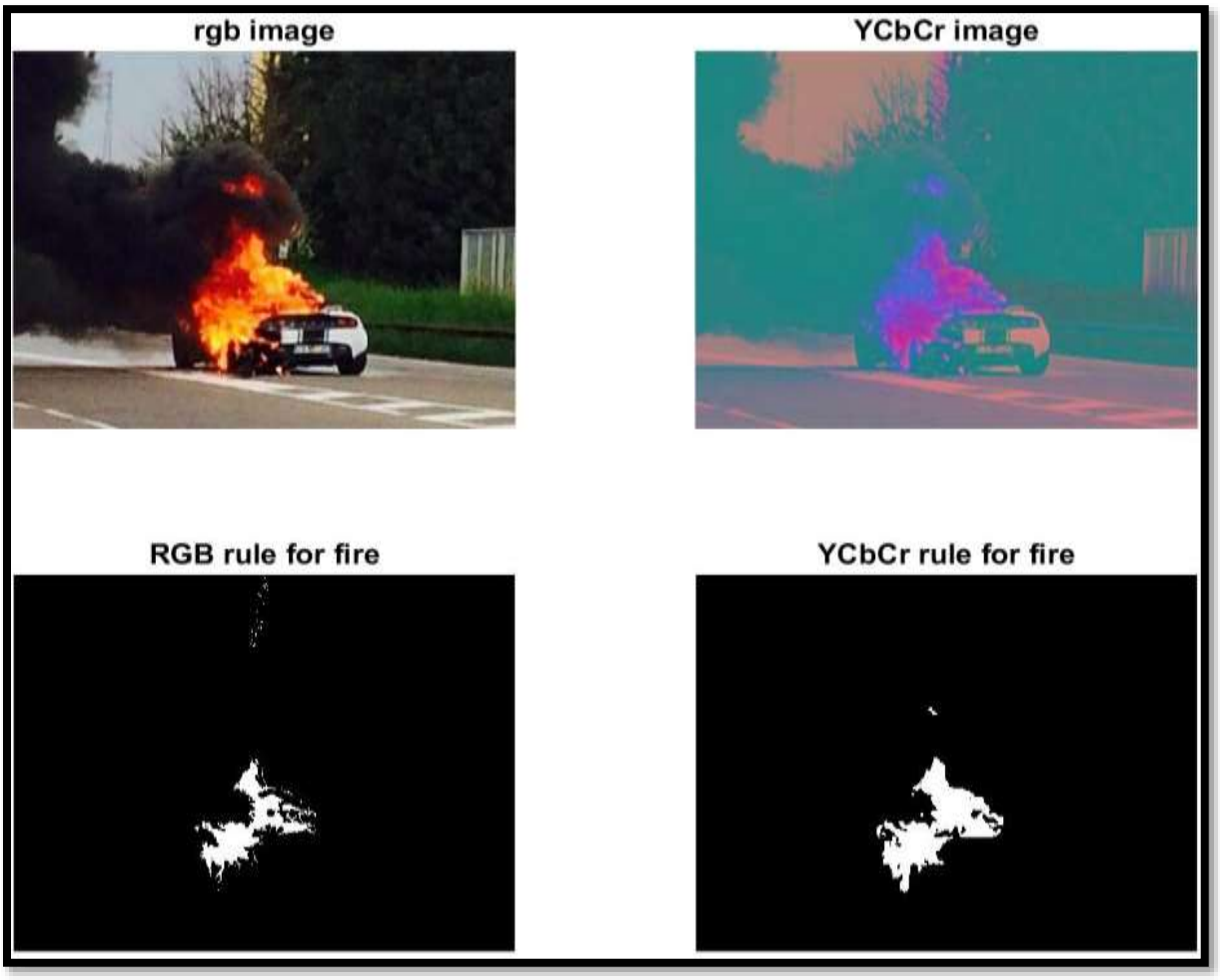

Figure 2.3 Test-1

2) An image of a landscape

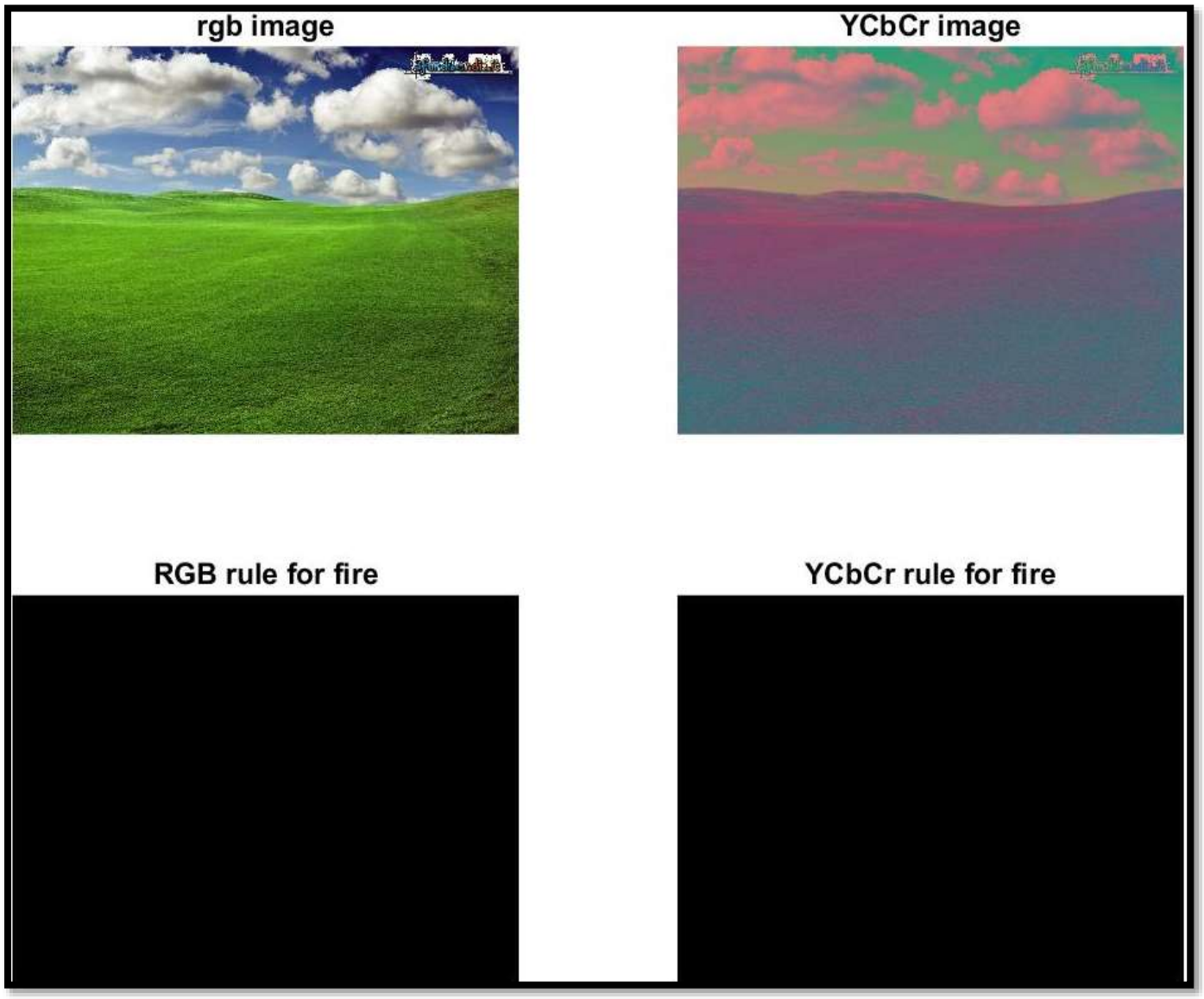

Figure 2.4 Test-2

3) Image of a house on fire

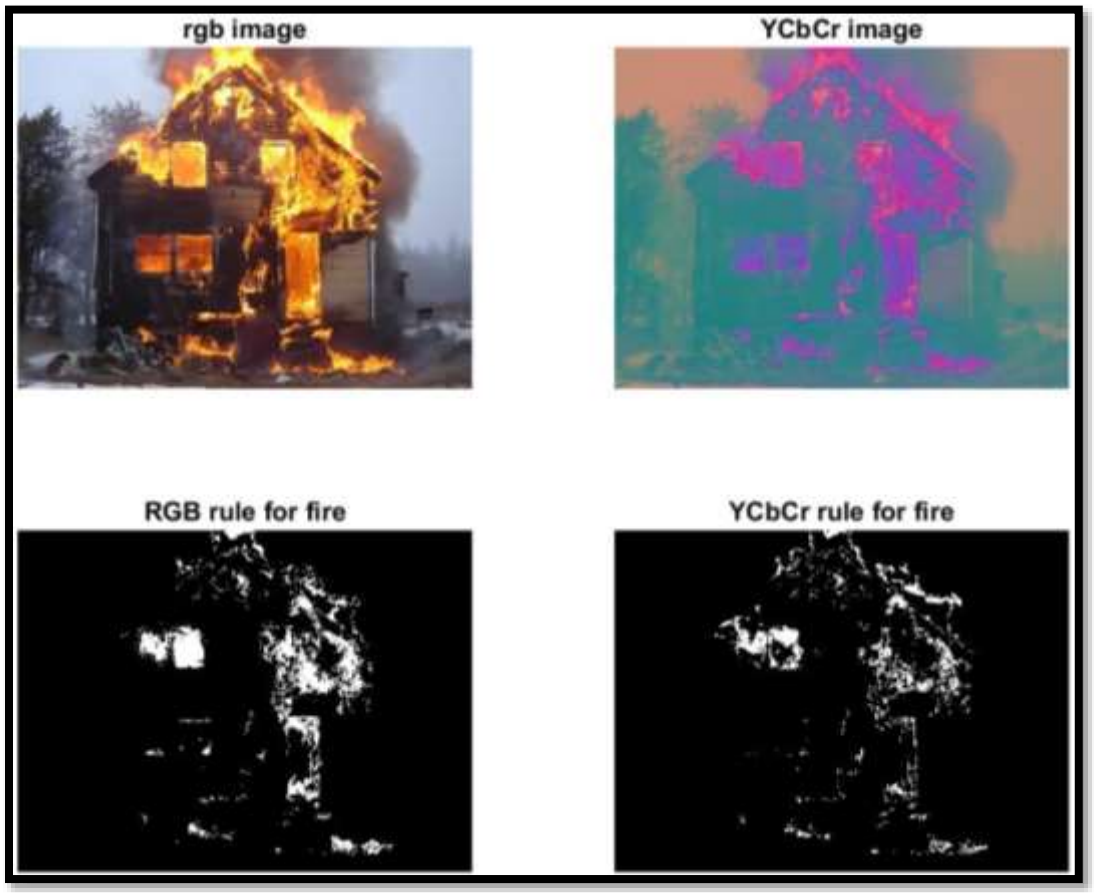

Figure 2.5 Test-3

2.6 Circuit Diagram

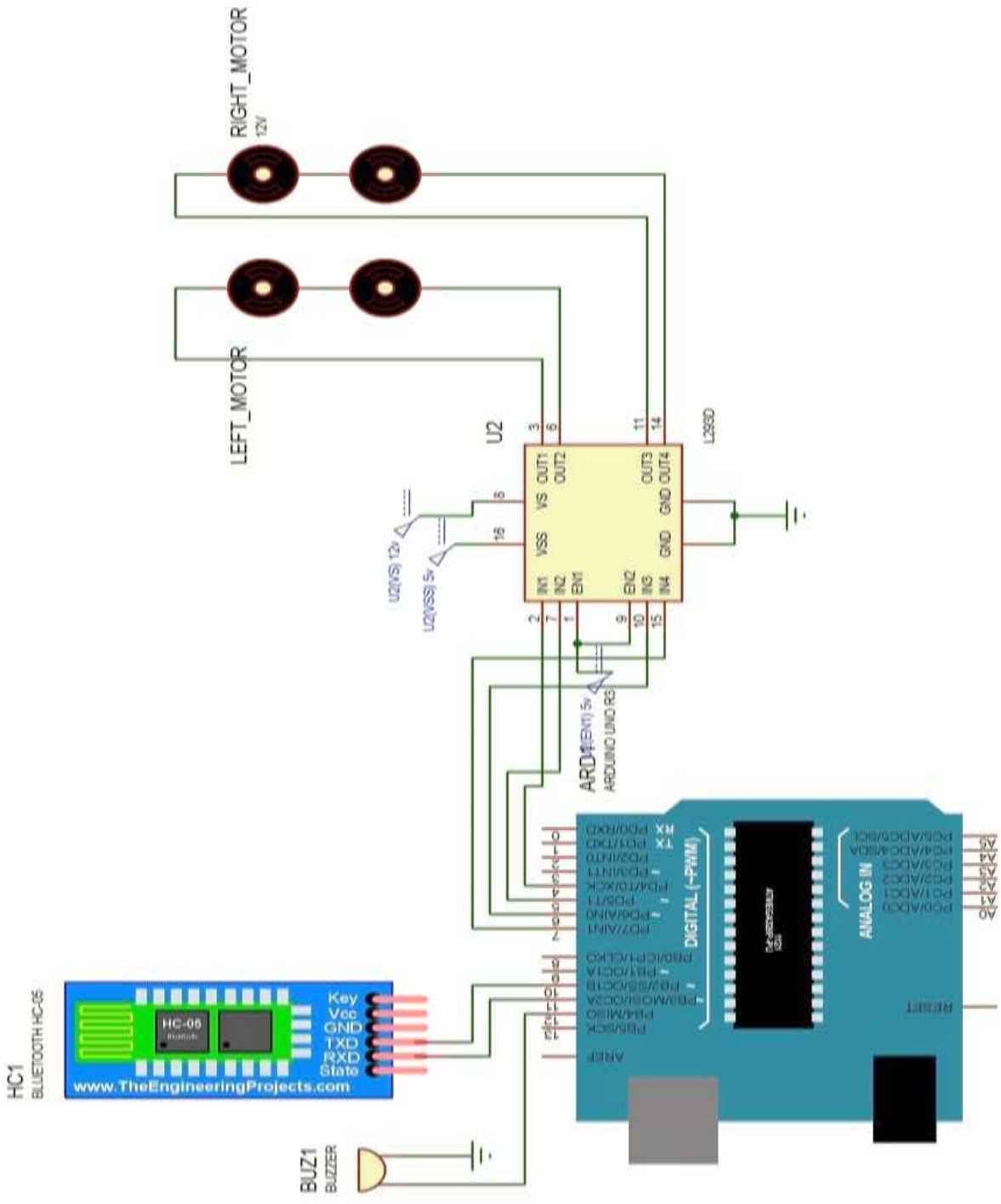

Figure 2.6 Circuit diagram

2.7 Program

2.7.1 Matlab program

Main program:

```
bt=Bluetooth('HC-05',1);
fopen(bt);
        url = 'http://192.168.43.120:8080/shot.jpg';
        rgb = imread(url);
        rgb = imresize(rgb,0.5);
        [i,v,f]=firedet(rgb,0);
        disp(f);


        u=im2bw(v);

    stats = regionprops(u, 'BoundingBox', 'Centroid');

    subplot 211;
    imshow(rgb);
    subplot 212;
    imshow(u);
    pause(0.5);

    a1=110;
    b1=50;
    a2=260;
    b2=260;
    a3=475;
    b3=360;
    a4=680;
    b4=260;
    a5=820;
    b5=50;


    hh=150;
    ww=150;
    bb1=[a1,b1,ww,hh];
```

```
    bb2=[a2,b2,ww,hh];
    bb3=[a3,b3,ww,hh];
    bb4=[a4,b4,ww,hh];
    bb5=[a5,b5,ww,hh];

    hold on

        bc = stats(1).Centroid;
        rectangle('Position',bb1,'EdgeColor','r','LineWidth',2)
        rectangle('Position',bb2,'EdgeColor','r','LineWidth',2)
        rectangle('Position',bb3,'EdgeColor','r','LineWidth',2)
        rectangle('Position',bb4,'EdgeColor','r','LineWidth',2)
        rectangle('Position',bb5,'EdgeColor','r','LineWidth',2)
        plot(bc(1),bc(2), '-m+')
        a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), '    Y: ', num2str(round(bc(2)))));
        set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'yellow');

    hold off
        centroidx=round(bc(1));
        centroidy=round(bc(2));



if (centroidx>=a1 && centroidx<=a1+ww && centroidy>=b1 && centroidy<=b1+hh )
    input=1;
end
if (centroidx>=a2 && centroidx<=a2+ww && centroidy>=b2 && centroidy<=b2+hh )
    input=2;
end
if (centroidx>=a3 && centroidx<=a3+ww && centroidy>=b3 && centroidy<=b3+hh )
    input=3;
end
if (centroidx>=a4 && centroidx<=a4+ww && centroidy>=b4 && centroidy<=b4+hh )
    input=4;
end
if (centroidx>=a5 && centroidx<=a5+ww && centroidy>=b5 && centroidy<=b5+hh )
    input=5;
end

if input==5
    pause(2);
    fprintf(bt,8);
    pause(2.5);
    fprintf(bt,5);
    pause(1);
```

```
    fprintf(bt,7);
    pause(0.5);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,8);
    pause(0.5);
    fprintf(bt,5);
    pause(1.5);
    fprintf(bt,3);
    pause(10);
    fprintf(bt,2);
    pause(0.5);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,99);
    pause(0.5);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,2);
    pause(2.5);
    fprintf(bt,5);
    pause(1);
end


if input==4
        pause(2);
    fprintf(bt,8);
    pause(2.5);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,4);
    pause(0.1);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,8);
    pause(1);
    fprintf(bt,5);
    pause(1.5);
    fprintf(bt,3);
    pause(10);
    fprintf(bt,2);
    pause(1);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,6);
```

```
    pause(0.2);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,2);
    pause(2);
    fprintf(bt,5);
    pause(1);
end

if input==3
    pause(2);
    fprintf(bt,8);
    pause(3);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,3);
    pause(10);
    fprintf(bt,2);
    pause(2.5);
    fprintf(bt,5);
    pause(1);
end

if input==2
    pause(2);
    fprintf(bt,8);
    pause(2.5);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,6);
    pause(0.2);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,8);
    pause(1);
    fprintf(bt,5);
    pause(1.5);
    fprintf(bt,3);
    pause(10);
    fprintf(bt,2);
    pause(1);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,4);
    pause(0.2);
    fprintf(bt,5);
```

```
    pause(1);
    fprintf(bt,2);
    pause(2);
    fprintf(bt,5);
    pause(1);

end

if input==1
        pause(2);
    fprintf(bt,8);
    pause(2);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,9);
    pause(0.6);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,8);
    pause(1);
    fprintf(bt,5);
    pause(1.5);
    fprintf(bt,3);
    pause(10);
    fprintf(bt,2);
    pause(1);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,77);
    pause(0.5);
    fprintf(bt,5);
    pause(1);
    fprintf(bt,2);
    pause(2);
    fprintf(bt,5);
    pause(1);
end
```

Functions:

1.firedet()

```
function [x,v1,fper]=firedet(a,per)
cnt=0;
ycc=rgb2ycbcr(a);

sz=size(a);
rw=sz(1);
cl=sz(2);
tcnt=rw*cl;

ym=mean(mean(ycc(:,:,1)));
cbm=mean(mean(ycc(:,:,2)));
crm=mean(mean(ycc(:,:,3)));

for i=1:rw
   for j=1:cl
      y=ycc(i,j,1);
      cb=ycc(i,j,2);
      cr=ycc(i,j,3);

      if y>=cb && cr>=cb && y>=1.5*ym && cb<=0.8*cbm && cr>=1.1*crm
         v1(i,j)=255;
         cnt=cnt+1;
      else
         v1(i,j)=0;
      end
   end
end
   if cnt>per*tcnt/100
      disp('fire detected');
      x=1;
   else
      disp('fire not detected');
      x=0;
   end
   fper=cnt/tcnt*100;
```

2.7.2 Arduino program

```
#include<SoftwareSerial.h>
#include<Servo.h>
int bt, buzzer = 12, pos = 0;
Servo myservo;
SoftwareSerial hc(10, 11); //Rx,Tx
void forward(void)
{
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
}
void reverse(void)
{
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
}
void stopit(void)
{
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
}
void left(void)
{
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);

}
void softleft(void)
{
  digitalWrite(4, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
```

```
}
void softrleft(void)
{
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);

}
void right(void)
{
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);

}
void softright(void)
{
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);

}
void softrright(void)
{
  digitalWrite(4, LOW);
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);

}
void setup()
{
  myservo.attach(9);
  Serial.begin(9600);
  hc.begin(9600);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(buzzer, OUTPUT);
```

```
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(buzzer, LOW);
  myservo.write(0);
}

void loop()
{
  bt = hc.read();
  Serial.println(bt);
  switch (bt)
  {

    case 8: forward();    break;
    case 65: forward();    break;
    case 2: reverse();    break;
    case 67: reverse();    break;
    case 6: right();      break;
    case 54: right();      break;
    case 4: left();       break;
    case 50: left();       break;
    case 5: stopit();     break;
    case 66: stopit();     break;
    case 7: softleft();   break;
    case 49: softleft();   break;
    case 77: softrleft();   break;
    case 51: softrleft();   break;
    case 9: softright();  break;
    case 53: softright();  break;
    case 99: softrright();  break;
    case 55: softrright();  break;


    case 0: digitalWrite(buzzer, HIGH); break;
    case 57: digitalWrite(buzzer, HIGH); break;
    case 1: digitalWrite(buzzer, LOW);  break;
    case 73: digitalWrite(buzzer, LOW);  break;

    case 3:
      {
        for (pos = 0; pos <= 130; pos += 1)
```

```
      {
        myservo.write(pos);
        delay(15);
      }
      delay(3000);
      for (pos = 130; pos >= 0; pos -= 1)
      {
        myservo.write(pos);
        delay(15);
      }
    } break;
  case 56:
    {
      for (pos = 0; pos <= 130; pos += 1)
      {
        myservo.write(pos);
        delay(15);
      }
      delay(100);
      for (pos = 130; pos >= 0; pos -= 1)
      {
        myservo.write(pos);
        delay(15);
      }
    } break;

 }
}
```

3. Conclusion & Future Work

3.1 Conclusion

The system works completely as expected. The image from IP cam is effortlessly acquired in matlab environment without any noticeable lag by wireless communication. The image is then processed in matlab with the algorithm which detects the fire and displays its percentage and accordingly derives the location of fire spot. The resulting data i.e. the location of the fire is sent via Bluetooth to the arduino Uno. According to the program in arduino the bot receives the location of fire and reaches the fire spot and extinguishes the fire. The bot returns to the initial position specified. Hence all the objectives are accomplished.

3.2 Applications

- Server room and data centers
- Go-downs
- Inventories
- Multi story building

3.3 Future Expansion

- Path planning
- Automatic fire alert system
- GPS module which sends the exact location to rescue team

4. Appendix

4.1 ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328. 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 an AC-to-DC adapter or battery to get started.

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.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will 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 platform.

Specifications:-

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 which 6 provide PWM output)
PWM Output	6
PCB Size	53.4 x 68.6 mm
Weight	25 g
Product Code	A000066 (TH); A000073 (SMD**)**

4.2 Bluetooth (HC05):-

It is a class-2 Bluetooth module with Serial Port Profile, which can configure as either Master or slave. A Drop-in replacement for wired serial connections, transparent usage. You can use it simply for a serial port replacement to establish connection between MCU, PC to your embedded project and etc.

HC-05 Specification:

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port
- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm

Figure 4.1 HC-05

Application:

- Computer and peripheral devices
- GPS receiver
- Industrial control
- MCU projects

4.3 L293D Motor Driver

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Figure 4.2 L293D

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.

Pin Diagram:-

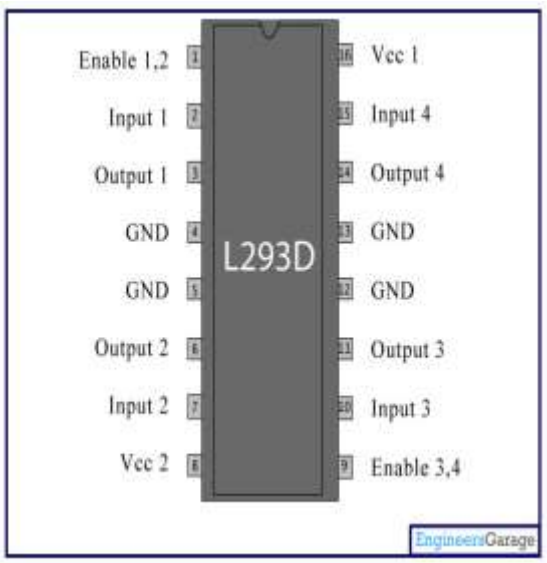

Figure 4.3 Pin Diagram of L293D

Pin Description:

Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc2
9	Enable pin for Motor 2; active high	Enable 3,4

10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc1

Table 4.1 Pin Description

Bibliography

1. A Research paper on “Fire Recognition Using RGB and YCBCR Color Space”
2. Matlab help function.