Retina Controlled Weaponry

By

GC Ahsan Bilal

GC Bilal Haider

GC Muhammad ZeeshanTufail

Submitted to the Faculty of Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad
in partial fulfillment for the requirements of a B.E Degree in Telecom Engineering

May 2015

Abstract

Motion sensing by vision technologies have come through advances in sensor design, material improvements, software innovations and progress in micro circuitry design and system integration. The signal from the camera i.e. tracking the retina is fed to retina tracking system. The retina tracking system tracks the retina motion and sends motion signals to the remote weapon. The remote weapon contains a processing unit which receives the commands from RTS (Retina Tracking System) and drives the weapon motion via its motor driver circuit. Furthermore for precise and accurate targeting the remote weapon returns a live video feed back to the user, from the camera mounted on the weapon.

This project presents solutions to the problems pertaining in conventional weaponry .i-e. Inaccurate and non-precise aiming of targets, direct engagement of enemy, Risk to human life, Inability to conduct and monitor remote operations.

It is hereby certified that the contents and form of the project report entitled “Retina Controlled Weaponry”, submitted by the syndicate of

GC Ahsan Bilal

GC Bilal Haider

GC Muhammad ZeeshanTufail

have been found satisfactory as per the requirement of the B.E. Degree in Electrical (Telecom) Engineering.

Supervisor:

Asst Prof EngrFazal Ahmed

MCS, NUST

_____

DECLARATION

We hereby declare that no content of work presented in this thesis has been submitted in support of another award of qualification or degree either in this institution or anywhere else.

DEDICATED TO

Almighty Allah,

Faculty for their help

Our parents for their support

And all the Shuhadas of Pakistan Army

ACKNOWLEDGEMENT

Nothing happens without the will of Allah Almighty. We thank Allah Almighty for giving us knowledge and strength to accomplish this task successfully.

We would like to thank our project supervisor, Asst Prof EngrFazal Ahmed, and our advisor, Lt Col Dr Adnan Ahmad Khan, without their support and encouragement; it would not have been possible to complete this project.

We would like to express our special gratitude and thanks to Abdul Moiz Ahmad Pirkani for his unflinching support in truly testing times.

We would also like to thank and our colleagues for helping in developing the project and people who have willingly helped us with their abilities.

Last but not least, we are very thankful to our parents, who bore with us in times of difficulty and hardship. Without their consistent support and encouragement we could not have accomplished our targets successfully.

Table of Contents

Contents

List of Abbreviations:

RTS	Retina Tracking System
TCP	Transfer Control Protocol
IP	Internet Protocol

List of Figures

Chapter 1

1.1 Introduction:

1.1.1 Background

Over the past decade, the art of weaponry has evolved tremendously worldwide. With the war on terror encompassing 1/3rd of the planet and the center being Pakistan. The need to combine latest technology with traditional warfare to overcome required objectives has never been more. Therefore the practice of war which had been in use for over a century, as a form of war policy, has been changed dramatically with greater awareness of tactical, operational, technical and strategic battle information.

Motion sensing by vision technologies have come through advances in sensor design, material improvements, software innovations and progress in micro circuitry design and system integration. The visual from the camera i.e. tracking the retina is fed to retina tracking system. The retina tracking system tracks the retina motion and sends motion signals to the remote weapon. The remote weapon contains a processing unit which receives the commands from RTS (Retina Tracking System) and drives the weapon motion via its motor drivers. Furthermore for precise and accurate targeting the remote weapon returns a live video feed back to the user, from the camera mounted on the weapon.

1.1.2 Problem Statement

This project presents solutions to the problems pertaining in conventional weaponry. i.e.

- Inaccurate and non-precise aiming of targets
- Direct engagement of enemy
- Risk to human life
- Inability to conduct and monitor remote operations

1.2 Project Description

The project involves movement of a gun through the retina movements of a person operating it. Tracking the retina movement is achieved by the techniques of image

processing using visual studio. The Retina tracking system is developed using C# and A-Forge libraries. After tracking the retina the next step is to send the signals from the RTS to the processing unit placed with the remote gun platform. The RTS is communicating with the processing unit through a TCP connection. The processing unit after receiving the signal from the RTS will operate the motor driver circuit which will govern the movement of the gun in up, down, left and right direction.

1.3 Prospective Application Area

- Keeping in view our current security situation of the country the project can be deployed in areas where you need protection from any intruder e-g government buildings, check posts etc.
- Can be deployed in ongoing military operations.
- Can be mounted over military vehicles.

1.4 Scope, Objectives, Specifications and Deliverables:

1.4.1 Scope and Objective:

The project is divided into following segments:

1. Detecting and tracking of Retina movement.
2. Interfacing the RTS with the processing unit i-e Raspberry Pi B+.
3. Development of motor driver circuit.
4. Corresponding weapon controls to Retina movements.

The Following are the Objectives of our project:

Project Objectives:

The gun will be able to detect the eye movement and the P-cap camera will feed a signal through RTS (Retina Tracking System) to the processing unit which will control the movement through the motor driver circuit.

Academic Objectives:

The academic objectives include:

- Retina Tracking
- 2.4 GHz Wi-Fi Communication over TCP.
- Programming and Hardware interfacing with Raspberry Pi Model B+
- Development of motor driver.

1.4.2 Specifications:

The technical specifications of our project include:

Hardware Specifications:

Raspberry Pi B+:

- It has a 700 MHz ARM processor
- 512 MB RAM
- 4 USB ports
- 1 Ethernet port
- 1 HDMI output port
- Audio output port
- 40 GPIO (General Purpose Input/output) pins. These GPIO pins can be used for hardware interfacing. We will only use 5 pins out of them:
    - 2 pins for left and right motion of the gun.
    - 2 pins for up and down motion of the gun.

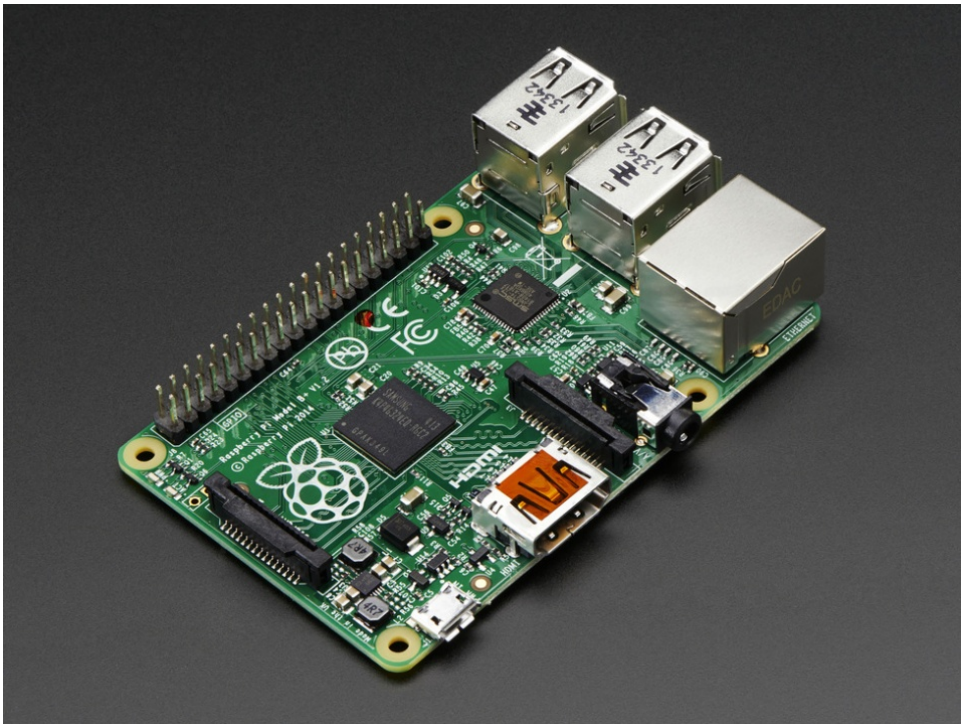

Fig.1 Raspberry Pi B+

Motor Driver Circuit:

The motor driver circuit is fabricated using 5 pin relays.

Relay Specifications:

- 12 volt relay.
- Support up to 10 amp current.

Software Specifications:

- The Retina Tracking System is developed using Visual Studio 2012.

1.4.3 Deliverables:

At the end of the project we will deliver Retina Controlled Weaponry, efficiently working and ready to be deployed in any ongoing Military operation in Pakistan.

Chapter 2

2.1 Literature Review:

2.1.1 Overview of existing literature:

Image Processing:

Image processing is used in a lot of applications as in medical imaging, factory automation, remote sensing, document image processing and defense/military applications[1].Image processing software required are too complex. Most of the imaging applications are real time applications that are associated with time deadlines. Hence the imaging software is expected to execute the tasks faster. Image processing software may be a general purpose programming language such as Java or C# or specialized programming environment such as MATLAB, to perform image processing tasks[2].

The RGB color model is the most common format, where the colors are represented in a cube. Each point in the cube is represented by a color. This model is used in TV, cameras, scanners and computer monitors[3]. The color image can be converted to grey scale image by replacing the RGB values by the luminance value for each pixel which can be obtained by[4].

$$y = R + G + B/3$$

The grey scale image can be converted to black and white image by setting a threshold value between 0 (Black) and 255(white). The values below the threshold value will be down converted into 0 and those above it will be up converted to 255 thus we will get a black and white image in which all the pixels will be either 1 or 0.

Wireless Communication:

WLANs: Wireless Local area networks work similar to the traditional LAN except for the wireless interface. WLANs provide high speed data communication in small areas such as a building or an office. WLAN uses the unlicensed ISM (Industrial, Scientific and Medical) frequency band. The ISM has band has three frequency ranges: 902-928 MHz, 2400-2483.5 MHz and 5725-5850 MHz. The most popular widely used WLAN standard is the IEEE 802.11b working at 2.4 GHz ISM band with a theoretical data rate of 11 Mbps[5].

TCP Connection: The TCP provides reliable transmission of data between the two communicating hosts. TCP corresponds to the layer 4 (transport layer) of the OSI mode. It is a connection oriented service through the underlying protocols for internetworking. To the host, it seems to be circuit switched connection. TCP does this by sequencing bytes with a forward acknowledgement number that indicates to the destination the next byte the source expects to receive. Bytes not acknowledged within a specified time period are retransmitted. TCP is a reliable transport protocol because it deals with lost, delayed, duplicate, or missed packets. TCP offers full duplex communication in which data can flow in both directions at the same time. TCP also provides flow control, error control and congestion control[6].

2.2 Problem Formulation:

Weapon is a device or system that is designed to permit humans to overcome natural physical and psychological limitations in order to protect themselves and to enable the killing and domination of other creatures, particularly their fellow human beings. Humans have proven themselves to be infinitely ingenious at creating and using devices to overcome their limitations. From one perspective human history can be seen as a series of ever-more-efficient devices to help humans communicate, travel, trade, work, and even to think. Similarly, the history of violence, peace, and conflict can be seen as the history, or the evolution, of a series of ever-more-efficient devices to enable humans to defend themselves and dominate their fellow human beings.

Different advancement in war strategy leads to smart-weapon technology e.g. off-road mines that can "listen" for moving vehicles and attack them, "thinking"

antipersonnel mines that are activated or deactivated automatically after a set period of time, and missile and artillery fire targeted by space-based intelligence-gathering satellites.

Smart weapons are attracting ever increasing funds and there is no military organization and branch that's not looking into ways to integrate them into its arsenal.

2.3 Existing Projects:

Wireless Aiming and Engaging Module:

Wireless aiming and engaging module was a final year project prepared by a group of students in MCS. In this project they made a wireless aiming system through a camera mounted on the gun and controlled by a wireless joystick controller. For the transmission of the signals from the joystick to the gun they used RF transmitter and receiver operating at 433 MHz. They used a PIC 16F877 as their processing unit.

Retina controlled operation of wheelchair for disabled:

Retina controlled operation of wheelchair was a final year project prepared by a group of students in MCS. In this project they made a wheelchair that is controlled by the motion of the retina using retina sensors and a PIC 16F877 microcontroller as their processing unit. The communication between the microcontroller and retina sensors was done using RF transmitter and receiver.

Chapter 3

3.1 Project Approach:

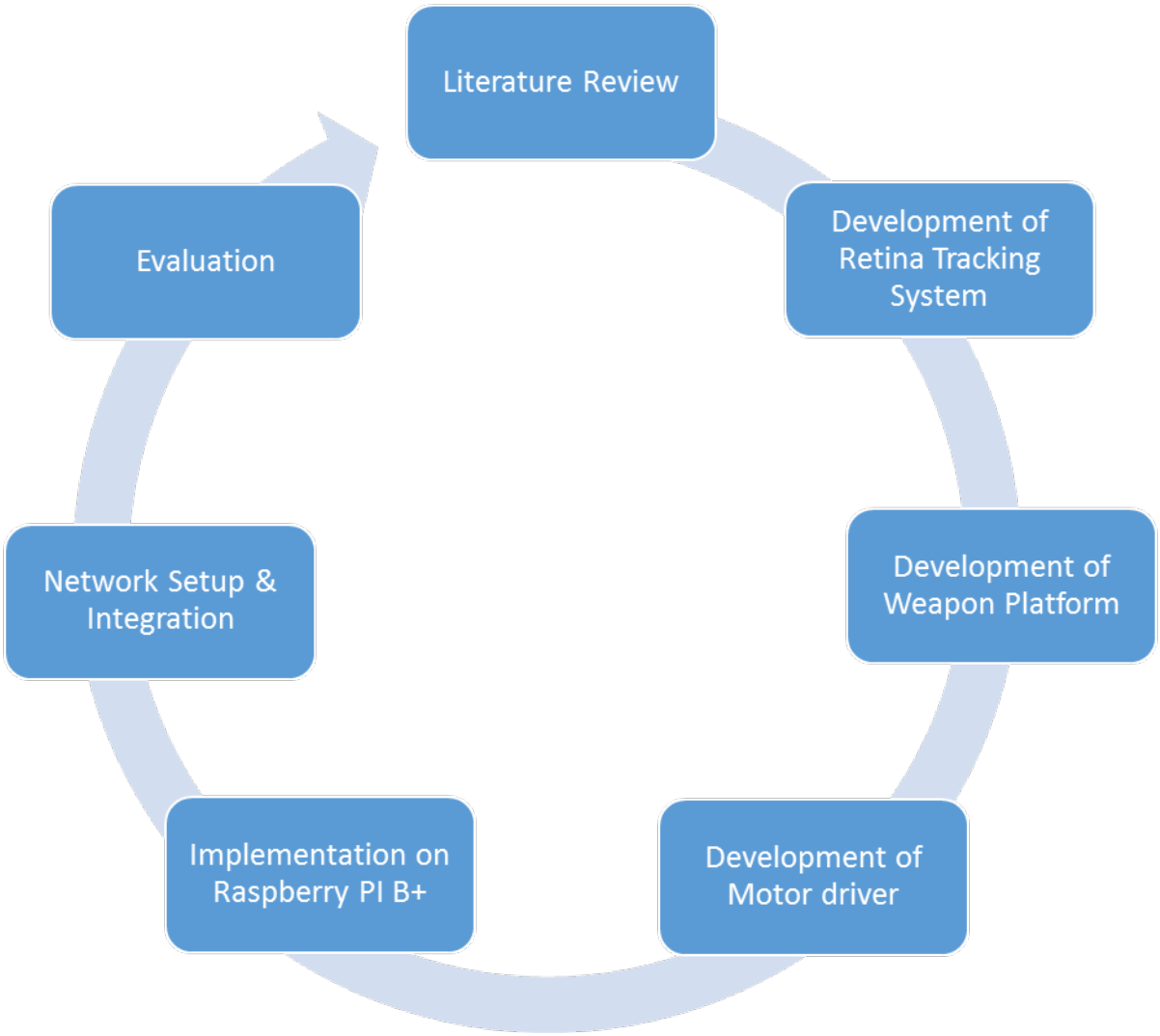

Fig. 2 Project Approach

3.2 Detailed Design:

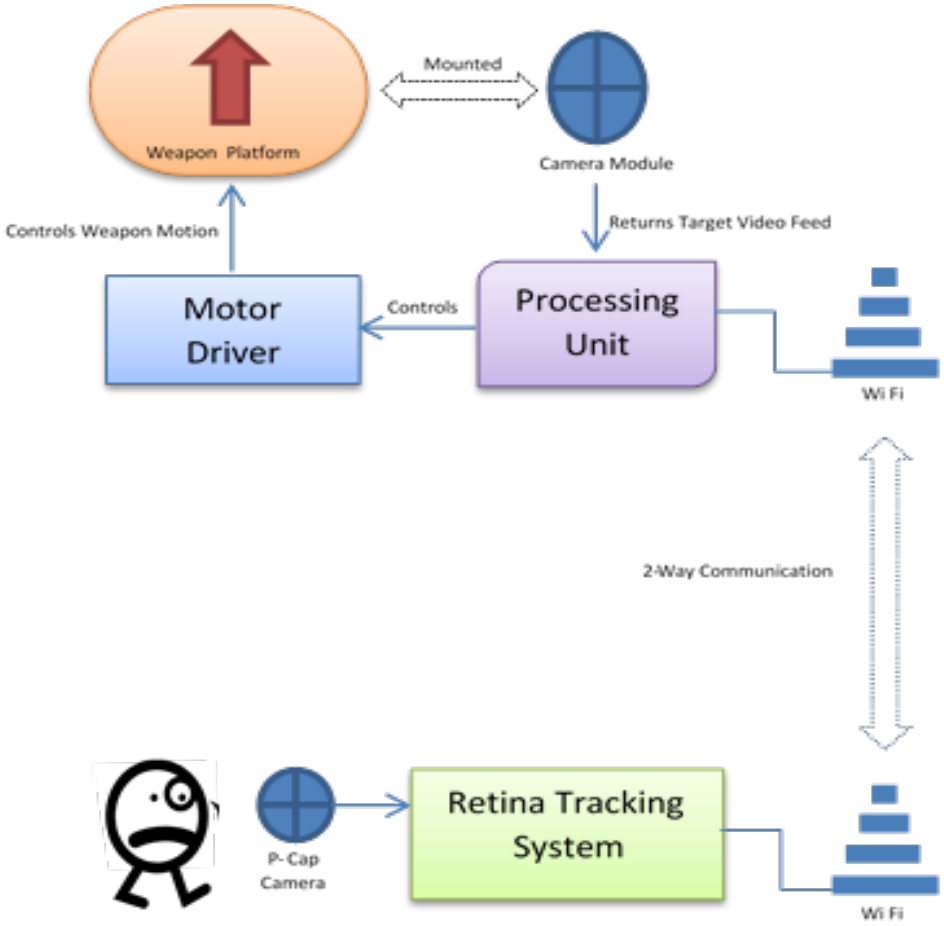

Fig.3 Detailed Design

The signal from the camera i.e. tracking the retina is fed to retina tracking system. The retina tracking system is basically a software designed to detect retina and its motion in horizontal and vertical direction. Coding of RTS (Retina tracking system) is done on in C# language using A-forge library for image processing.

The retina tracking system tracks the retina motion and sends motion signals to the remote weapon. The remote weapon contains a processing unit (Raspberry pi B+) which receives the commands from RTS (Retina Tracking System) and drives the weapon motion via its motor drivers. Furthermore for precise and accurate targeting the remote weapon returns a live video feed back to the user, from the camera mounted on the weapon.

3.2.1 Retina Tracking System:

The RTS is developed using Visual Studio 2102 using image processing techniques through C# and AForge libraries.

Following picture shows the appearance of main window of RTS (Retina Tracking System) and how the software tracks the retina position.

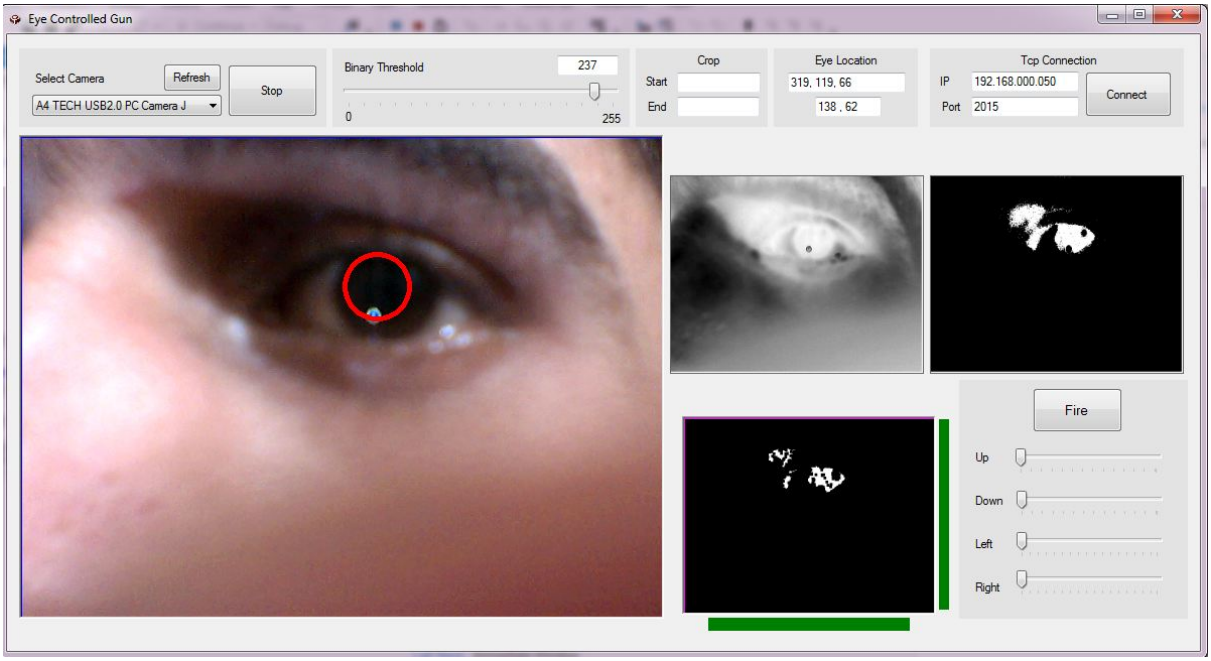

Fig.4 RTS Main Window

SELECT CAMERA BUTTON

First thing to do is to select camera which will be used for tracking the retina of the eye.Multiple cameras option will be available (built in cameras or connected through USB port) out of which select the camera you want to use and in this case will be a normal webcam which we are using A4 tech USB 2.0 PC Camera.Better the camera better and easier will be the detection.

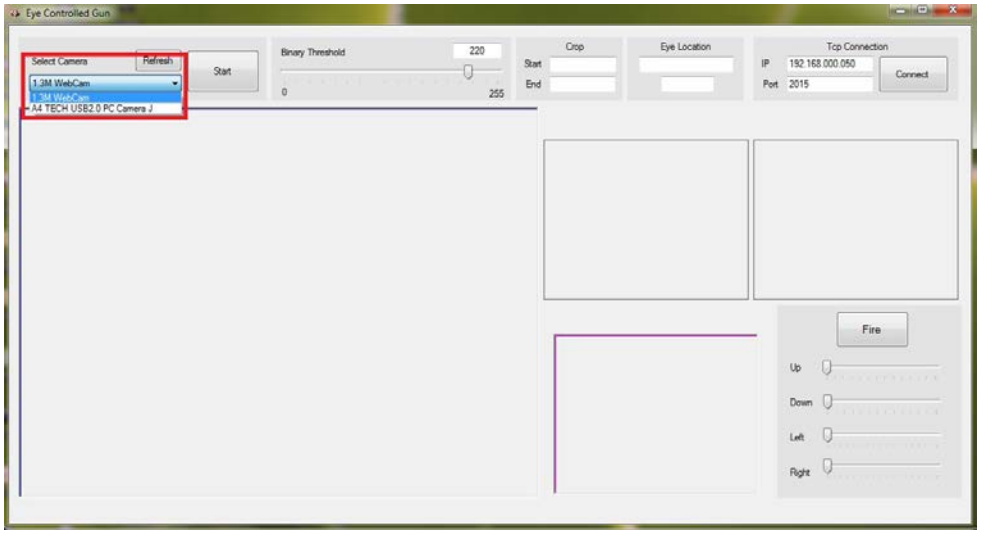

Fig. 5Selection of Camera

REFRESH AND START BUTTON:

- Refresh Button is used to refresh the main video stream.
- Start Button is used to start the functionality and tracking of the retina position and movement using the camera selected

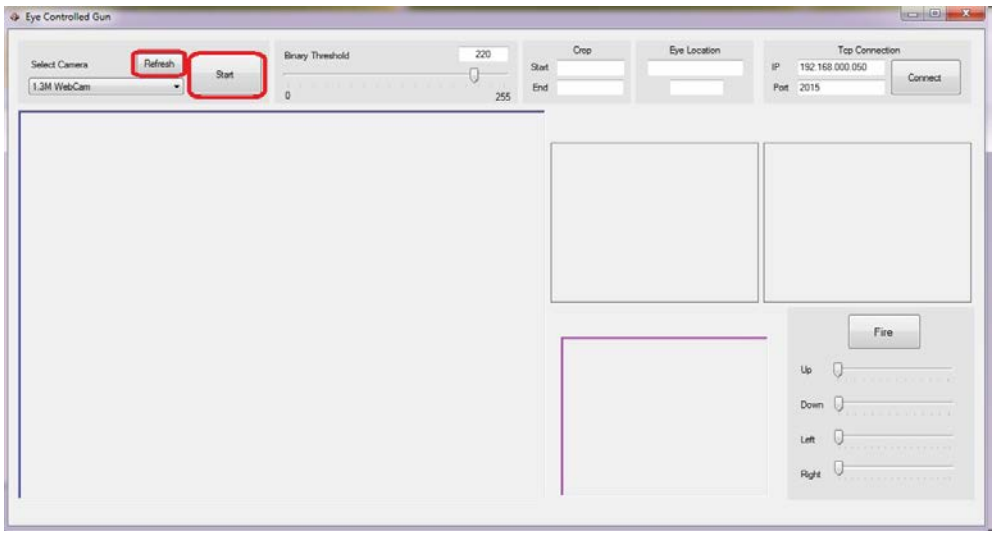

Fig. 6 Refresh and Start Button

BINARY THRESHOLD

This bar is used to set a threshold between black and white. Any pixel having grayscale value higher than the threshold will be changed to white and that having a lower value will be converted to black. Threshold value will be set according to our requirements but by default it is set to 220.

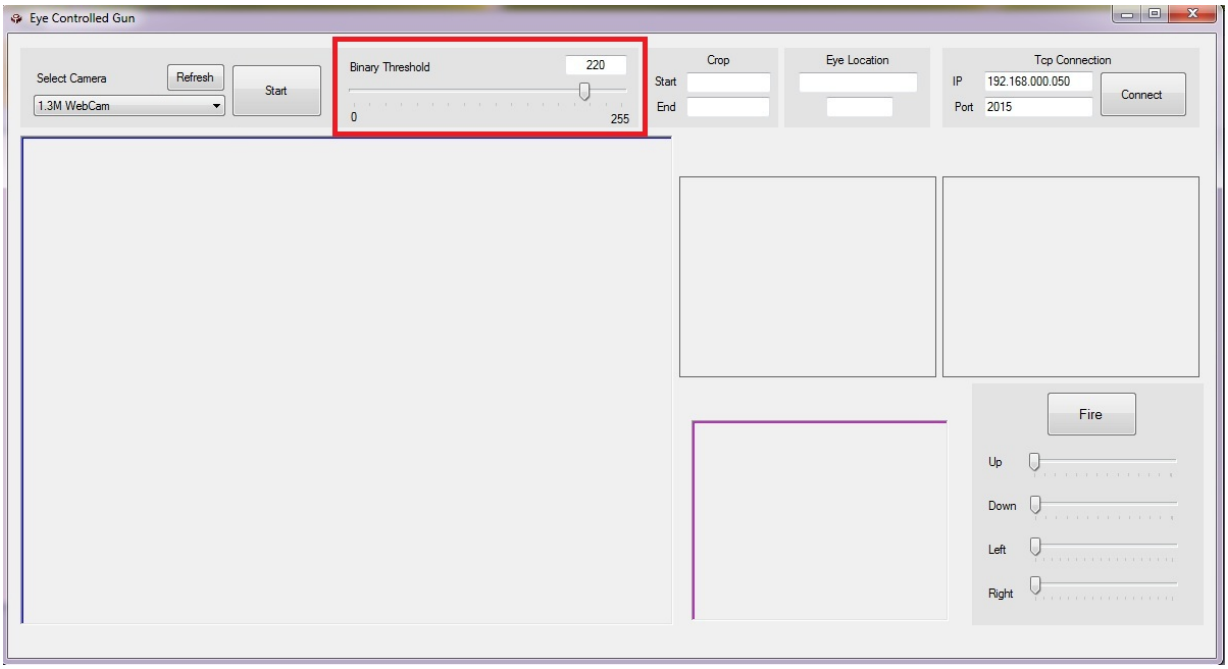

Fig. 7 Binary Threshold

CROP AND EYE LOCATION

I. Blue box around the eye in image is the cropped portion which limits the image detection to specific region.

II. Eye Location value will be according to the position of the eye in that specific cropped region.

Cropping can be done by feeding the values of X-axis and Y-axis in the start and end places in the window or it can be done more easily using the mouse.

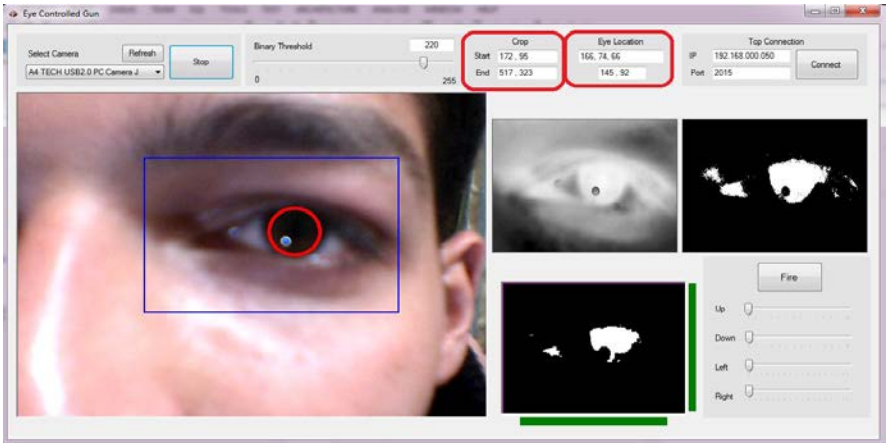

Fig. 8Crop and Eye Location

TCP CONNECTION

This is IP which will be used in connections which will be utilized in project for transferring information between Retina Tracking System and Raspberry Pi B+ which will then give commands to motor driver circuit to drive our motors and control the gun movement.

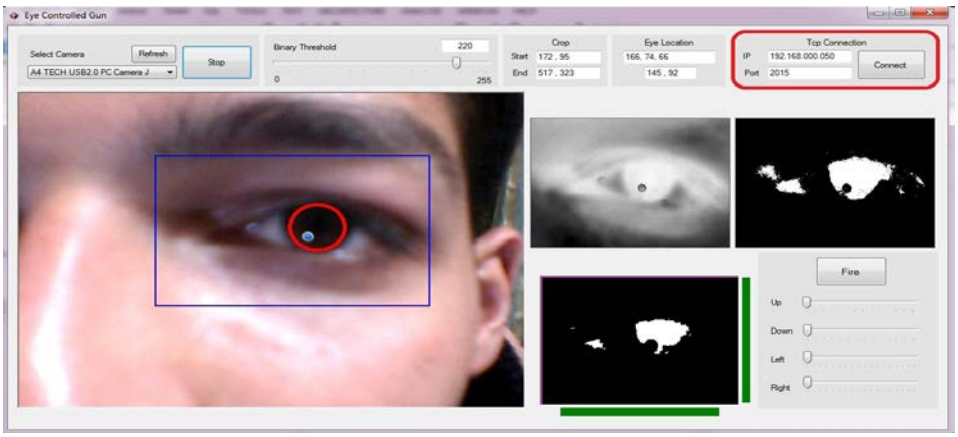

Fig. 9TCP Connection

Image Blocks

1) First Block which is shown in the figure below is a grayscale converted from RGB image. The software converts the RGB image to Grayscale by utilizing formula as follows:

Grayscale image=(R+G+B)/3

Grayscale conversion is done to make the image processing easy.

2) Second Block shows the black and white image (binary image) converted from the grayscale image.

3) Third block is 3*3 erosion. Erosion shrinks the boundary of the binary to be smaller.The structuring element M is a square matrix of binary values that is applied to the image I with the following rule for the eroded output.

Erosion rule: replace the central point of the mask M with the smallest value of I covered by the mask. Assuming I is a binary image, then the mask centered at the boundary of I will output zero because the mask overlaps both 0s (outside boundary) and 1s (inside boundary).

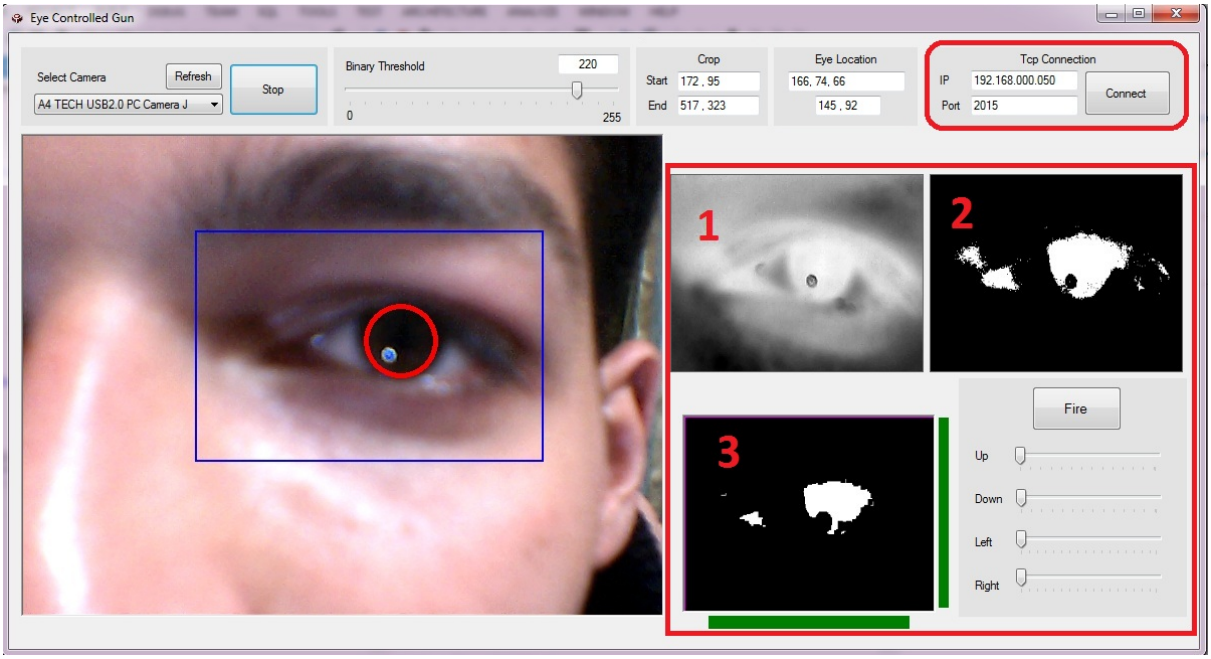

Fig.10Image Blocks

Setting boundary

The four purple lines are manually controlled by using the bars which are named as up, down, left and right to set a specific portion. Outside this central region will decide the movement of gun corresponding to the movement of retina. Whenever the retina moves out of this box the RTS will send the commands to Raspberry pi.

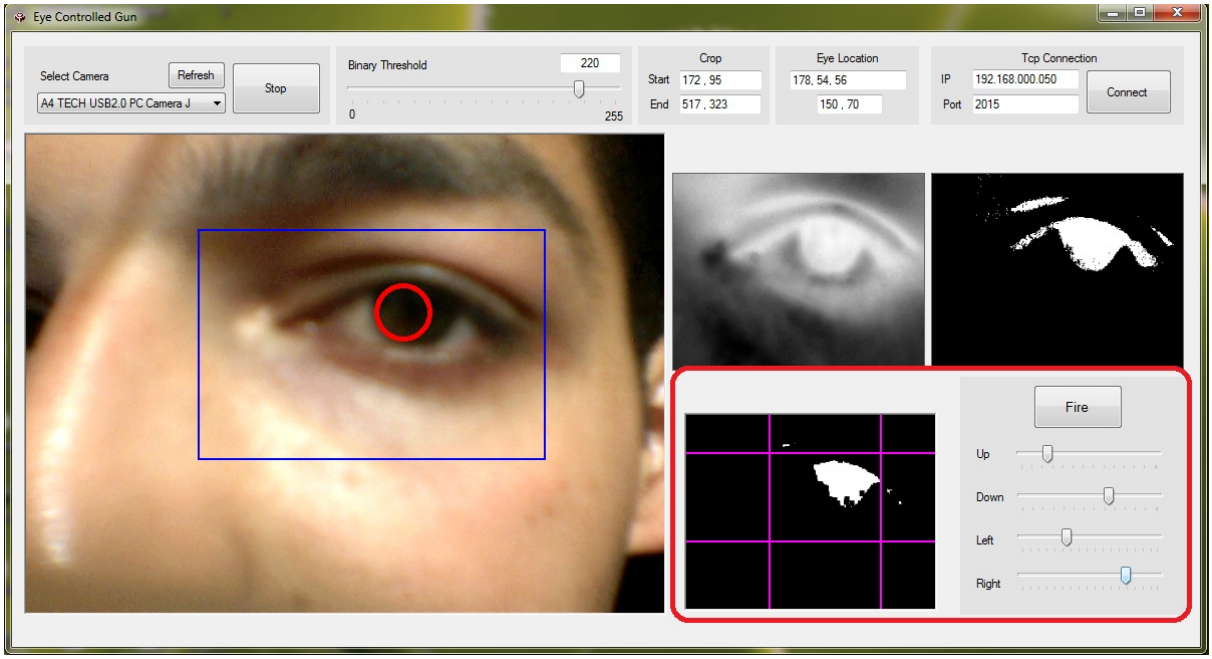

Fig.11Setting the boundary lines

3.2.2 Motor Driver Circuit:

Motor driver circuit consist of following four parts:

1. Step down Transformer
2. DC Conversion
3. Optocouplers
4. 5 pin Relays

1. Step Down Transformer

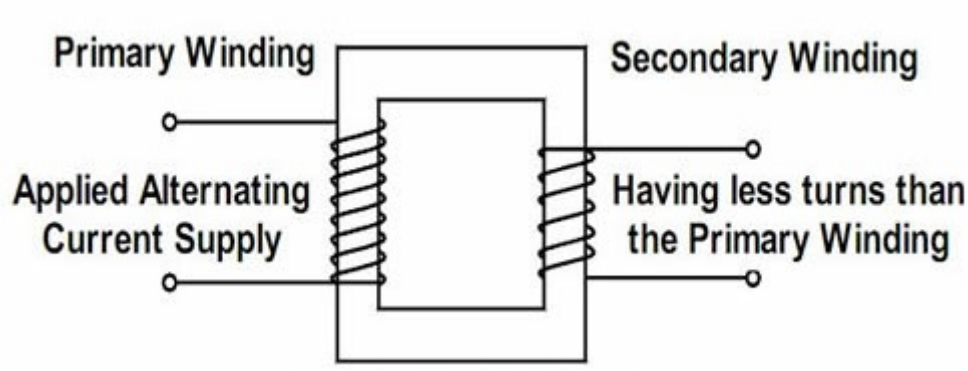

Fig. 12 Step Down Transformer

The step down transformer is used to step down 220V AC to 12V AC.

2. DC Conversion

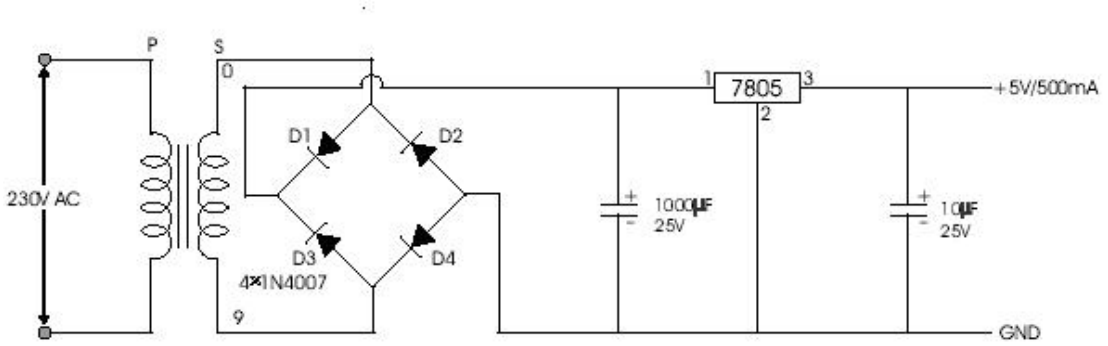

Fig. 13 DC Conversion

The 12V AC from the transformer is fed to the rectifier circuit for AC to DC conversion. The output after rectification and removing the ripples by capacitors is 5V DC which will be used to drive the motors.

3. Optocouplers

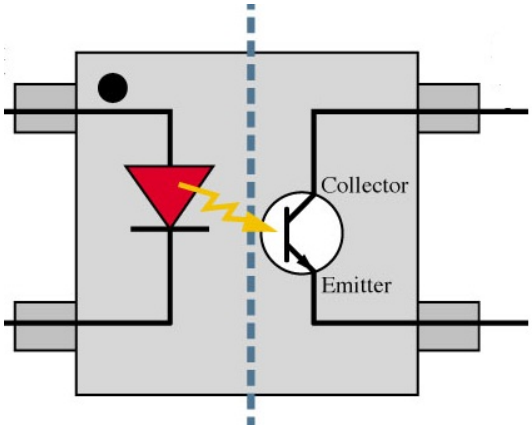

Fig. 14 OptoCoupler

The optocoupler is used to separate a high voltage circuit and a low voltage delicate or sensitive circuit so that the high voltage cannot damage the sensitive side.

The optocoupler has the signal from raspberry pi on one side and the output from the rectifier on the other side. The signal from the raspberry pi will turn the light emitting diode on which will complete the circuit on the other side so that the motors can get the driving voltage.

4. Relays

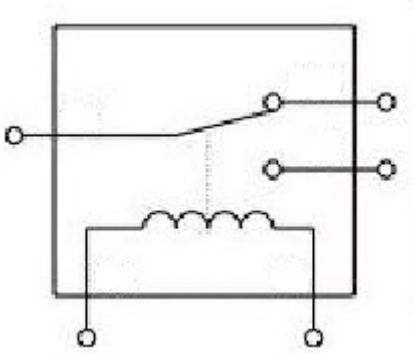

Fig. 15 5 pin relay

2 Relays are being used for driving one motor which will result in following outputs as shown in the table shown below resulting in the rotation of gun in desired direction:

Relay 1	Relay 2	Output 1	Output 2	Direction of movement
0	0	+	+	No movement
0	1	+	-	In one direction
1	0	-	+	Reverse direction
1	1	-	-	No movement

Table.1 Motor movement w.r.t Relay outputs

Circuit Diagram

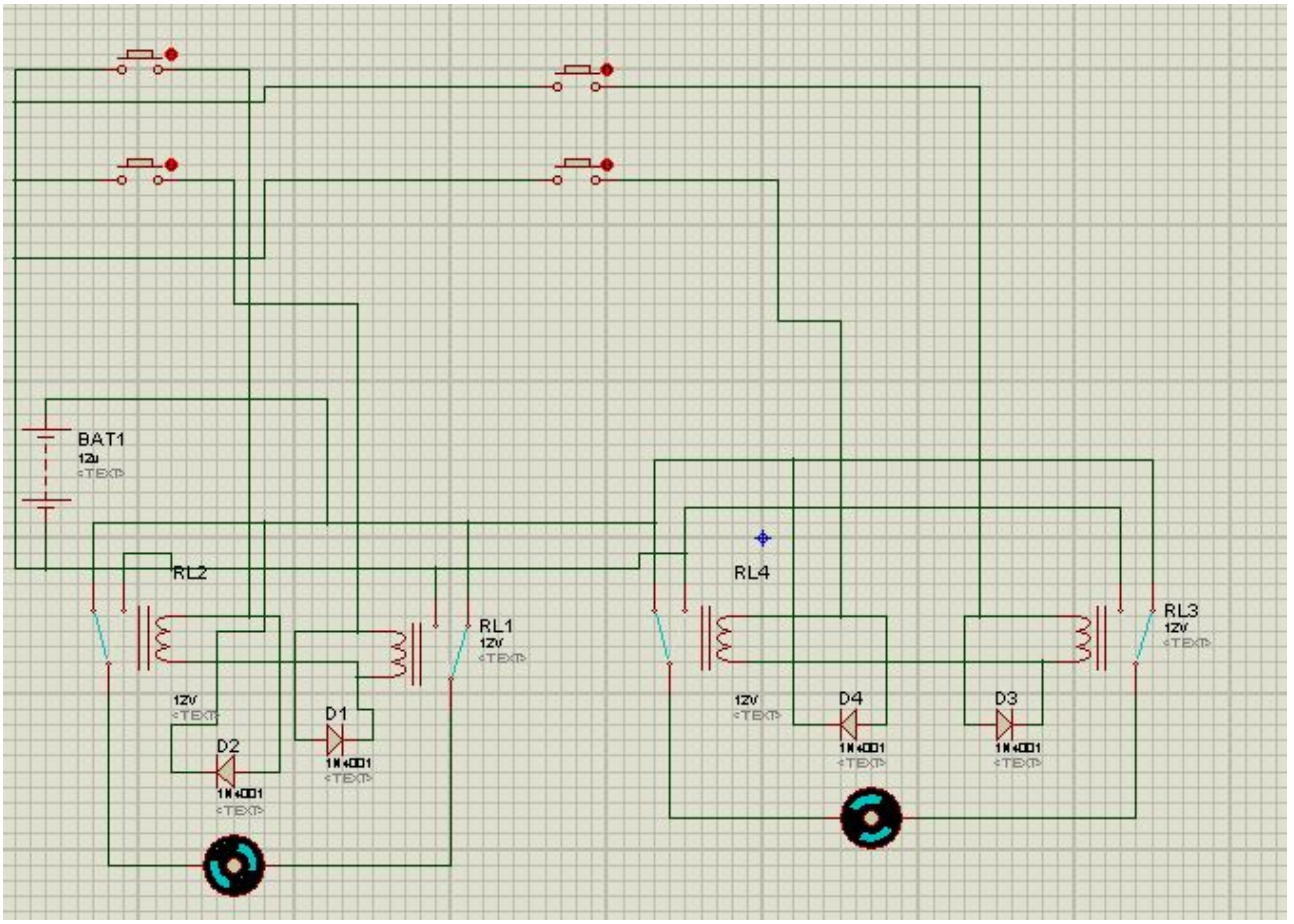

Fig. 16 Motor Driver Circuit Diagram

3.2.3 Weapon Platform:

Fig. 17 Weapon Platform

The movement of the motors will be governed by the raspberry pi which is getting its commands from the RTS. The motors used and their specifications are as follow:

Azimuth Movement Motor:

HorizontalMovementof the weapon platform is established by this motor.

Fig. 18 Azimuth Movement Motor

Elevation Movement Motor:

Vertical Movement of the weapon platform is established by this motor.

Fig. 19 Elevation Movement Motor

Nominal Voltage	**12 V**
Nominal Current	1.5 A
Nominal Power	50 W
High Speed	90 rpm

Table. 2 Motor Specification

Chapter 4

4.1 Project Analysis and Evaluation

All the results and simulations related to the modulator andsensors have been demonstrated below.

Retina Tracking System

1. First thing to do is to select camera which will be used for tracking the retina of the eye.Multiple cameras option will be available (built in cameras or connected through USB port) out of which select the camera you want to use and in this case will be a normal webcam which we are using A4 tech USB 2.0 PC Camera.Better the resolution of the camera better and easier will be the detection.

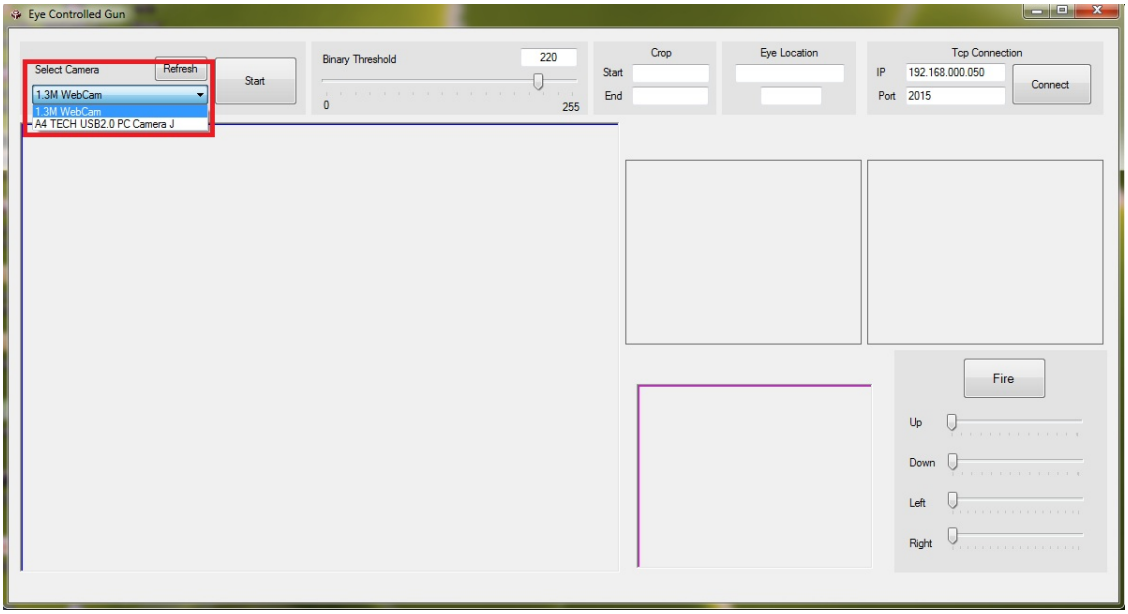

Fig. 20Camera Selection

2. After selecting the camera and clicking the start button the software starts working.

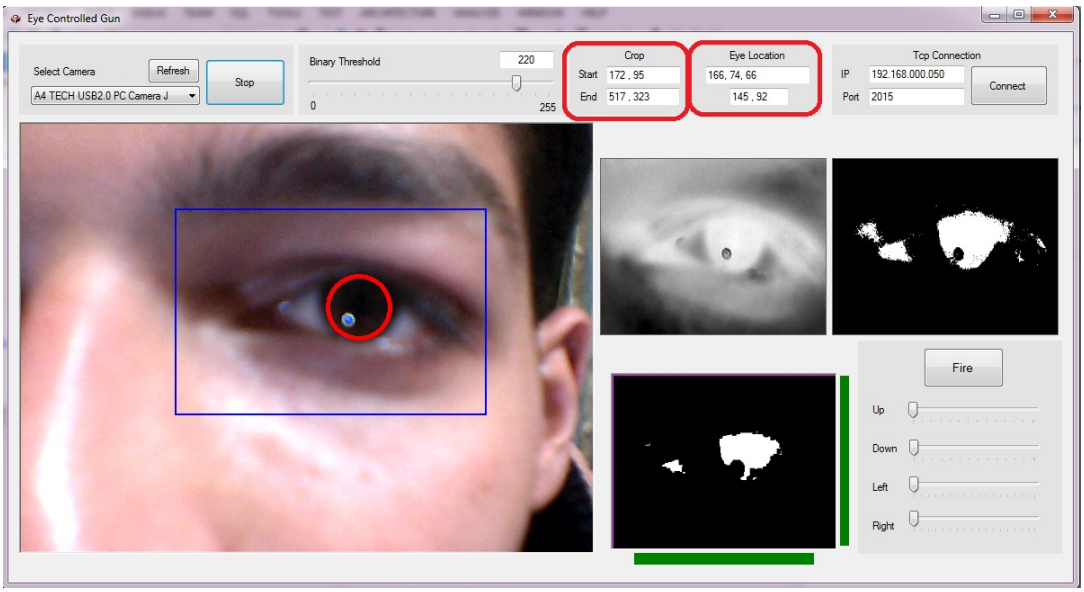

Fig. 21 Crop and Eye Location

The software can either work on the whole video from the camera or we can make a selection by cropping the desired section out of the whole picture box. Cropping will limit the detection only to the cropped portion.

3. The threshold value can be set during the run time as per the desire of the user (set 220 by default). Changing the threshold value changes the black and white image in the picture box 3.

4. The part used for connecting the RTS and Raspberry Pi B+ for the exchange of information is shown in figure 23. The IP and port address of the raspberry Pi is entered here to establish the TCP network.

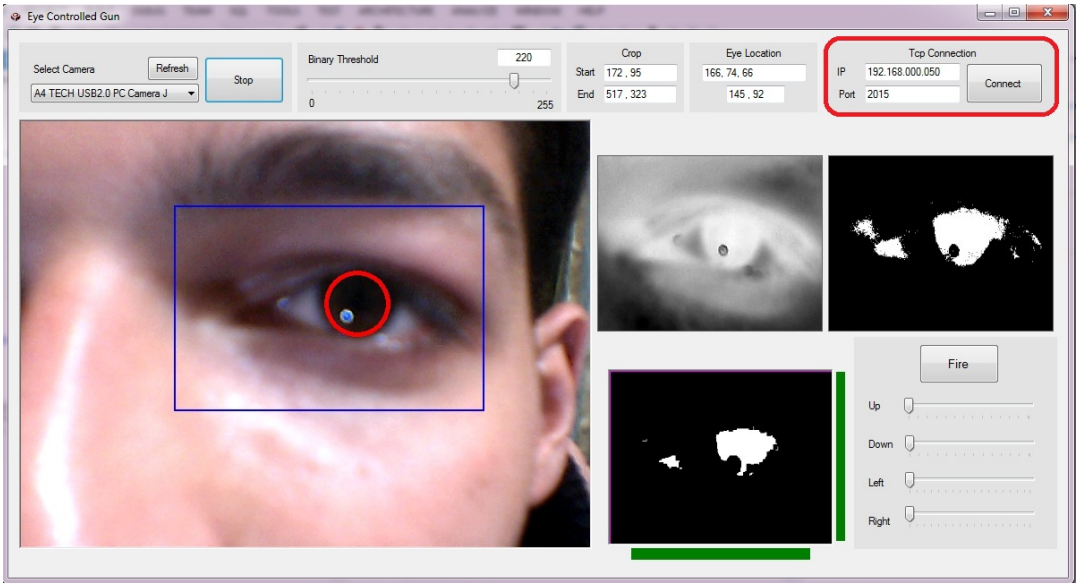

Fig. 22 TCP Connection

5. All the image blocks are shown in the following fig and are explained:

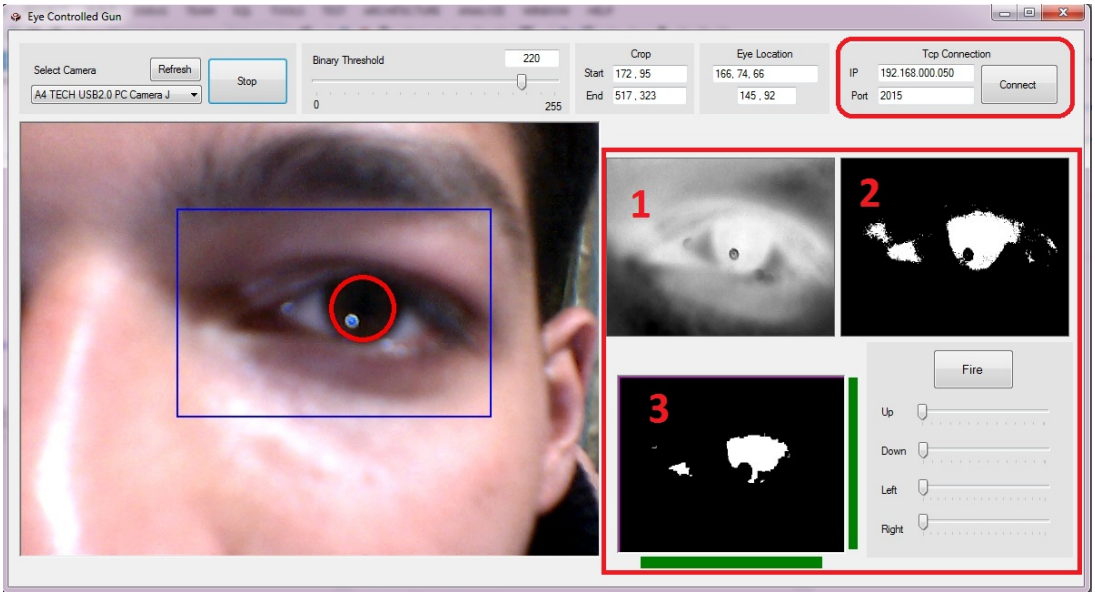

Fig. 23 Image Blocks

1. Picture Box 2: The system applies the grayscale BT709 to the main image box and converts it to grayscale and the resultant grayscale image is cloned here at picture Box 2.

2. Picture Box 3: After grayscale conversion a binary filter is applied to the picture box 2 to get a black and white image according to the set threshold value and is cloned here at picture box 3.
3. Picture box 4: After black and white conversion we can see a big white spot and a number of other small white spots. To remove the smaller white spots we apply 3x3 erosion to the picture box 3 and clone the resultant image to picture box 4. As can be seen that the smaller white spots are reduced in picture box 4 and here the biggest blob function is applied to track the movements of retina as the Biggest Blob is the retina.

6. After the retina is tracked, now we need to send its motion related commands to the processing unit (Raspberry Pi B+) installed at the remote weapon platform and for this purpose we define the boundaries so as whenever the retina moves out of those boundaries on any side the RTS will send the defined signals to the Raspberry Pi B+.

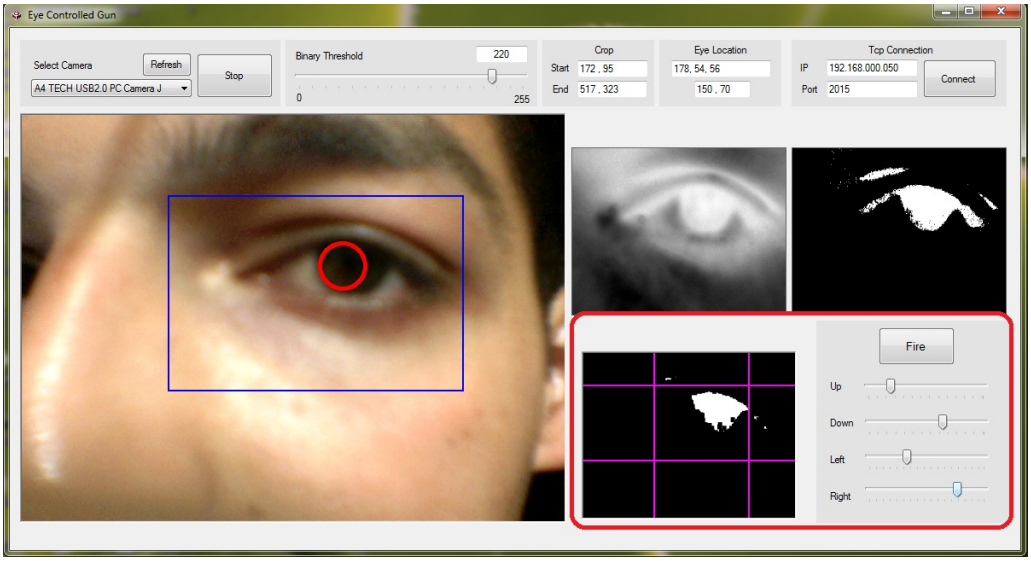

Fig. 24 Setting Boundaries

TCP Connection with Raspberry Pi B+ Test

The TCP connection between raspberry Pi is checked through a software known as TCP/IP Builder and the results are as under:

Establishing the connection

The IPs and port addresses of both the devices were entered and the connection was established and after that different commands were sent through the TCP/IP builder and they were received at the raspberry pi as shown in the following figures.

Sending Command to Raspberry pi.

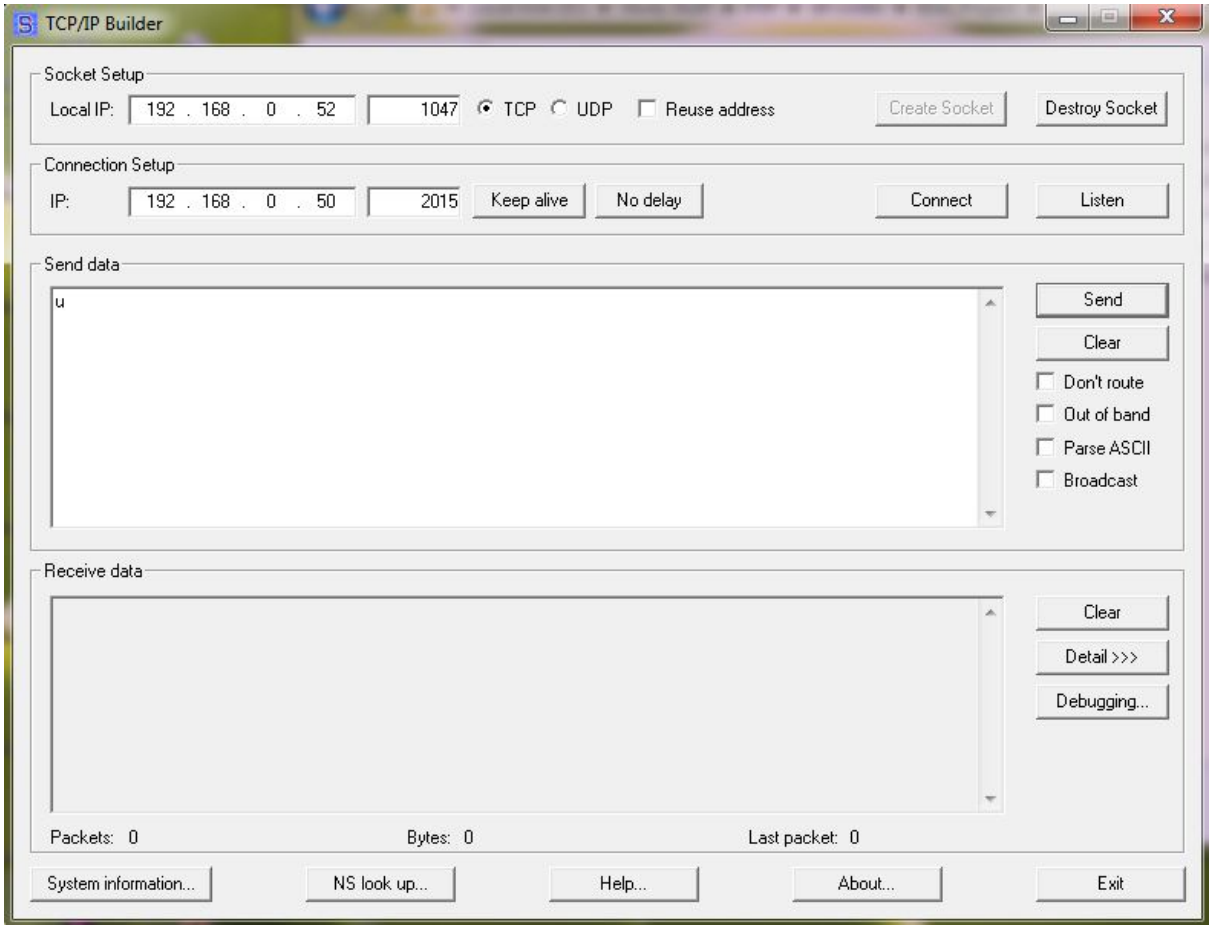

Fig. 25 Sending Command for up movement to Raspberry pi

Receiving command at raspberry pi:

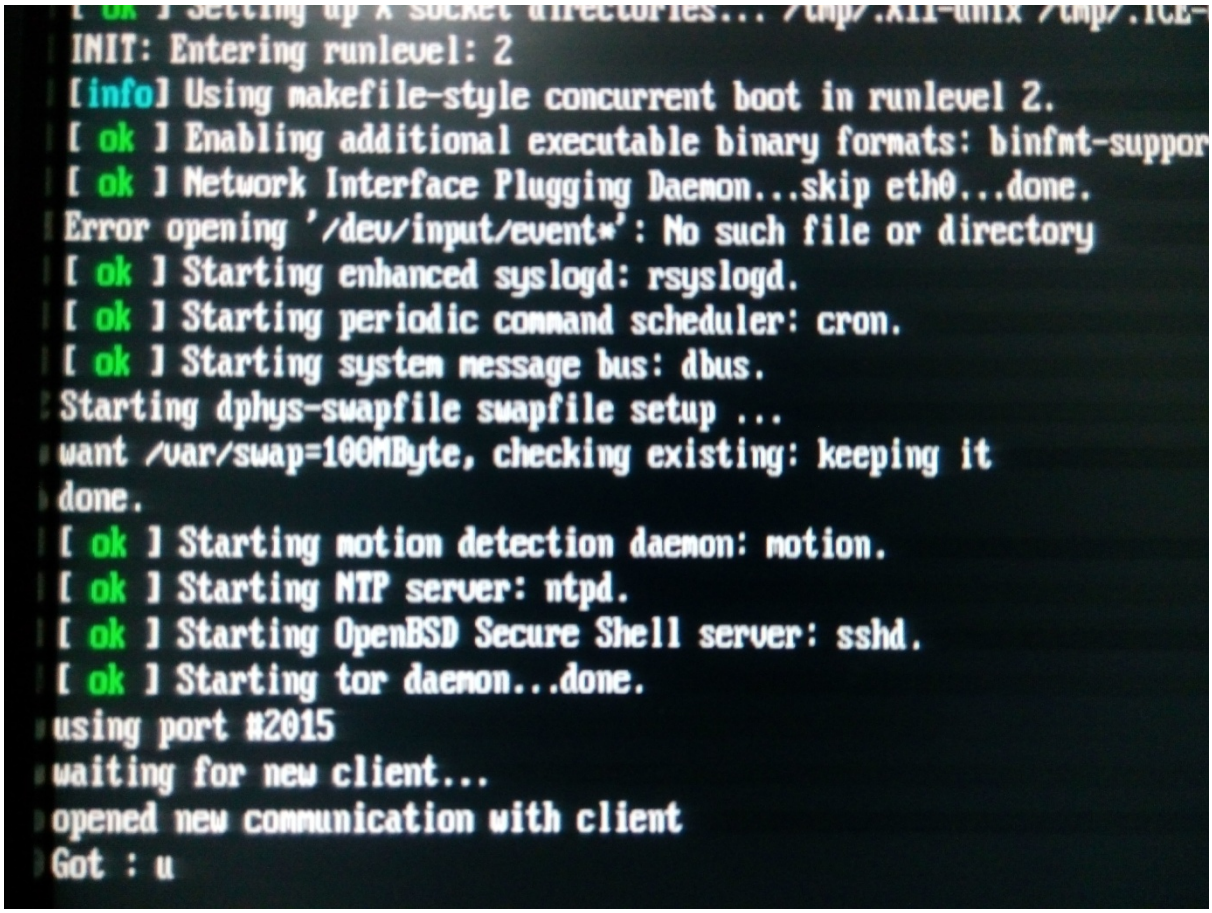

Fig. 26 Receiving Command at Raspberry Pi

Raspberry Pi operating Motor Driver Circuit

After checking the TCP Connection we checked operation of raspberry pi with motor driver circuit and the results are as follow:

Response of raspberry pi and motor driver circuit on receiving up command:

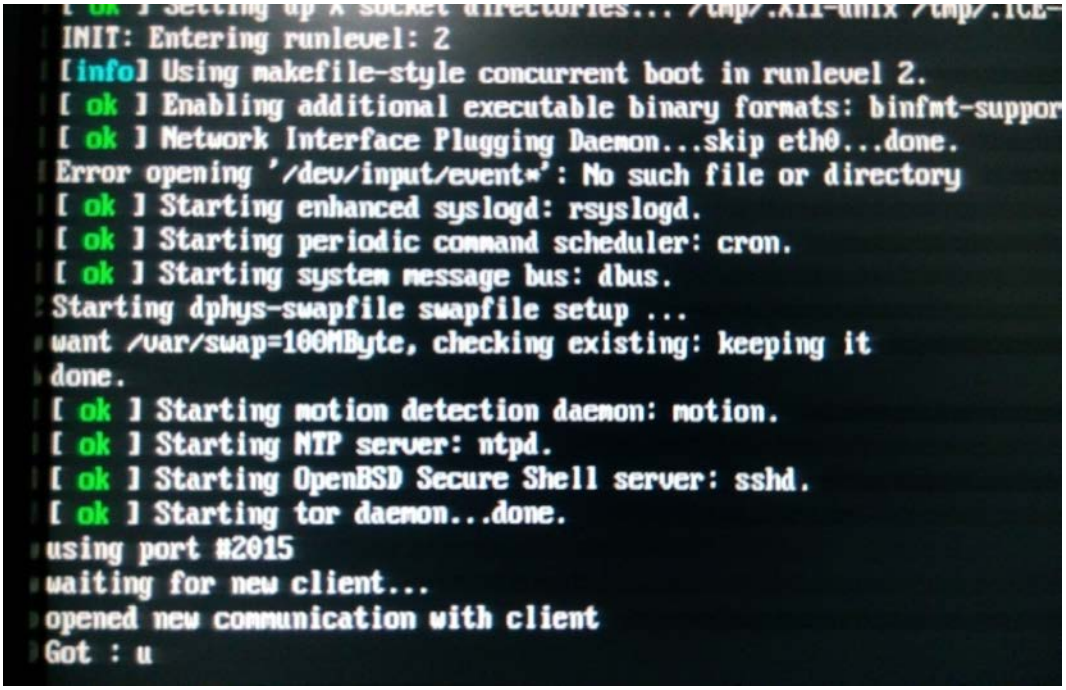

Fig. 27Up command received

The practical reception of up command translated into motor driver circuit.

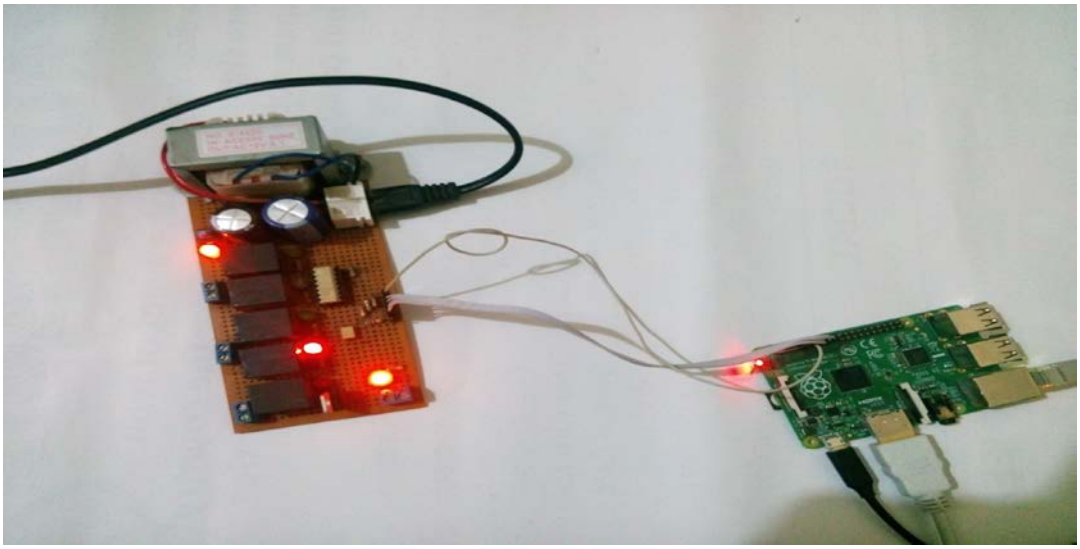

Fig.28 Practical Reception of Up Movement

Response of raspberry pi and motor driver circuit on receiving down command:

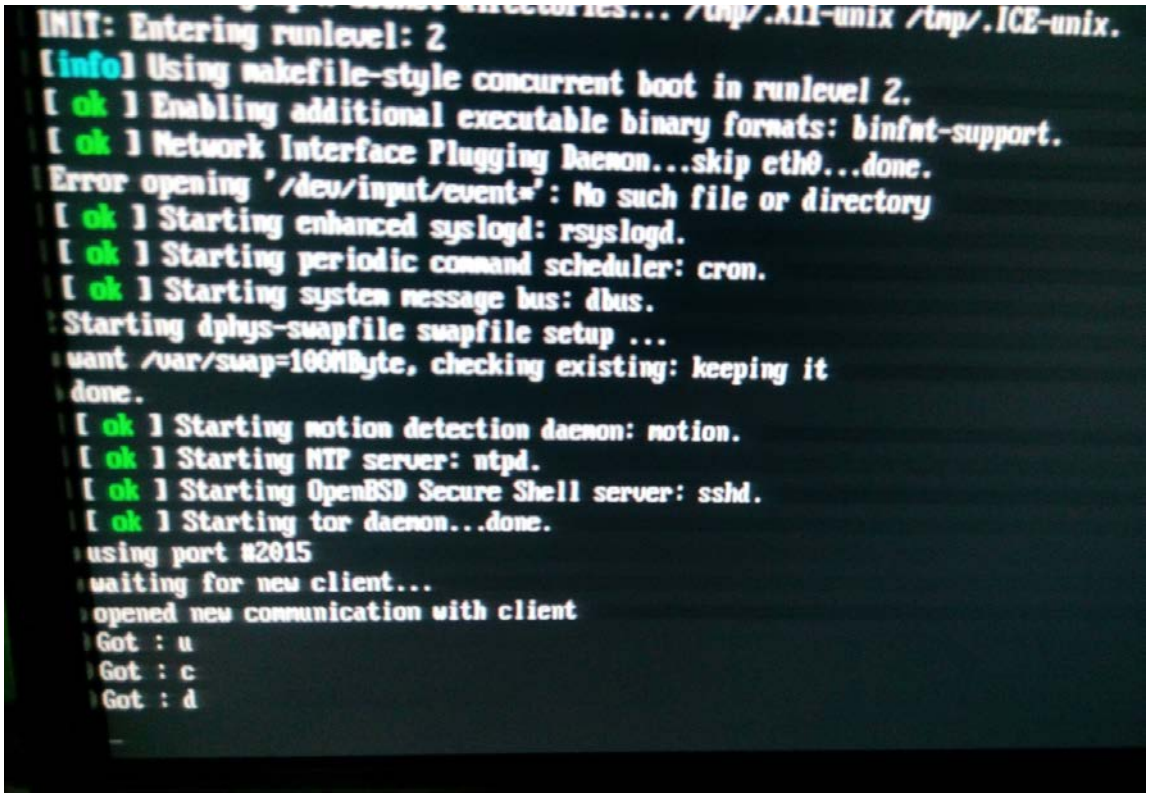

Fig. 29Down command received

The practical reception of down command translated into motor driver circuit.

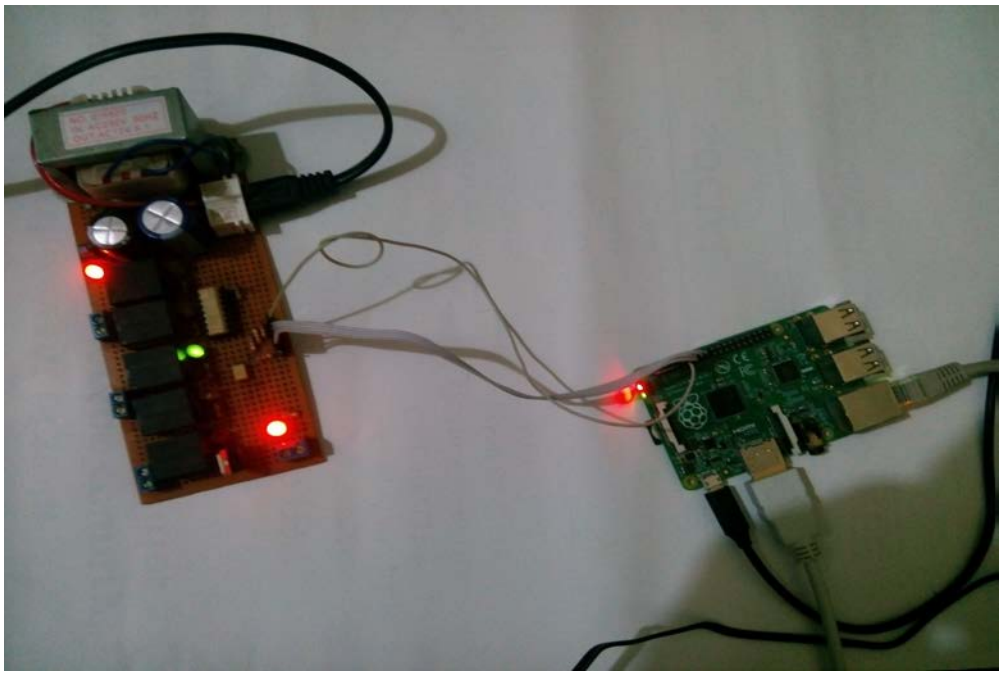

Fig. 30 Practical Reception of Down Movement

Chapter 5

5.1 Recommendations for future work

So far we have achieved the movement of a gun through the movement of retina but still there is a lot of space for future work regarding this project.

1. Face recognition system can be incorporated with the current project to make it more applicable from security point of view.
2. Retina scanning system can be developed so that only authorized persons can operate the project.
3. Auto lock feature can be installed so that the gun keeps on pointing to the target locked.
4. Properly adjusting the power of the motors according to the weight of the weapon heavy machinery can also be easily controlled.

Chapter 6

6.1 Conclusion

Project overview

The retina tracking system tracks the movements of the retina and send the respective movement signal to the processing unit (raspberry pi B+) wirelessly through a TCP connection.The processing unit is mounted at the remote weapon platform. The Raspberry Pi B+ receive the signals from the RTS and operate the motor driver circuit according to the commands/ signals received by it. The motor driver circuit drive the motors to achieve the desired movement with the help of relays.

Objectives Achieved

Successful tracking of the retina, establishing a TCP connection between the Raspberry Pi B+ and the RTS over the Wi-Fi, controlling the gun movement trough retina by a motor driver circuit operated by Raspberry Pi B+.

Limitations

There are certain limitations of the project which are mentioned below:

1. Requirement of proper lighting for tracking of the retina.
2. No security mechanism developed so can be used by anyone.
3. Lag in the video received from the camera mounted over the Gun.

Applications

1. Can be deployed at security check posts and other places where security is required without risking a human life.
2. Can be mounted over a vehicle.

Chapter 7

7.1 Demonstration Outline

All the result and simulation related to retina tracking and gun movement have been demonstrated in the report. These are the basic outline of demonstration:

1. Retina Tracking System
    i. Selecting the camera
    ii. Start and refresh button
    iii. Cropping the eye location
    iv. Setting Binary threshold
    v. Image boxes
    vi. Forming the TCP connection
    vii. Selecting the boundaries for command sending

2. TCP Connection Test
    i. Establishing the connection using TCP/IP Builder software

3. Motor Driver Circuit Integration with Raspberry Pi B+
    i. Checking output on GPIO pins of Raspberry Pi B+

4. Motor Driver Circuit response to Commands sent by Raspberry Pi B+
    i. Translation of up and down command into motion through the motor driver circuit

References

S.Sridha, *DIGITAL IMAGE PROCESSING*, 2011

[1] Chapter 1: Introduction to image processing

1.8 Image processing applications

[2] Chapter 1: Introduction to image processing

1.9.3 Image processor

[3] Chapter 8: Colour image processing

8.4.1 RGB Colour Model

[4] Chapter 8: Colour image processing

8.3.1 Conversion of colour image to Grey scale image

ITI SahaMisra, *Wireless Communication and Networks*, 2010

[5] Chapter 3: Evolution of Modern Mobile Wireless Communication Systems

3.4 Wireless Local Area Networks (WLANs)

[6] Chapter 9: Overview of Internet Protocol and Mobile Internet Protocol

9.1 Transmission Control Protocol (TCP)

Research Papers

Research paper on Retina Based mouse control system (Hardik N. Mewada .Parul Institute of Engineering & Technology, Vadodara. India: Dhairya Vyas. Parul Institute of Engineering & Technology. Vadodara, India).

Research paper on Lye Movement-Based Iluman-Computer Interaction Techniques by (Robert.I.K. Jacob Human-Computer Interaction Lab Naval Research Laboratory Washington, D.C.)

Appendix A

RETINA CONTROLLED WEAPONRY

Extended Title:

Retina controlled movement of weaponry.

Brief Description of The Project / Thesis with Salient Specs:

Over the past decade, the art of weaponry has evolved tremendously worldwide. With the war on terror encompassing l/3`dof the planet and the center being Pakistan. The need to combine latest technology with traditional warfare to overcome required objectives has never been more. Therefore the practice of war which had been in used for over a century, as a form of war policy, has been changed dramatically with greater awareness of tactical, operational, technical and strategic battle information.

Motion sensing by vision technologies have come through advances in sensor design, material improvements, software innovations and progress in micro circuitry design and system integration. The signal from the sensor is fed to a microcontroller via appropriate device running on preprogrammed instructions. The sensor after tracking retina movement will feed a signal to microcontroller which will control the movement of weaponry at hand.

Scope of Work :

The project is divided into following segments:

- Sensing of Retina movement
- Calibration of sensor
- Interfacing the sensor and weapon with the microcontroller.
- Corresponding weapon controls to Retina movements

Academic Objectives :

Getting the students familiarize with sensor calibration, microcontroller programming, servo motors and interfacing between sensor and microcontroller.

Application / End Goal Objectives :

At the end of the project we will deliver Retina Controlled Weaponry, efficiently working and ready to be deployed in any ongoing Military operation in Pakistan.

Previous Work Done on The Subject :

1. RF controlled movement of Gun. MCS NUST. 2014.
2. Retina controlled operation of wheelchair for disabled, MCS NUST, 2013.
3. Research paper on Retina Based mouse control system (Hardik N. Mewada .Parul Institute of Engineering & Technology, Vadodara. India: Dhairya Vyas. Parul Institute of Engineering & Technology. Vadodara, India).
4. Research paper on Lye Movement-Based Iluman-Computer Interaction Techniques by (Robert.I.K. Jacob Human-Computer Interaction Lab Naval Research Laboratory Washington, D.C.)

Material Resources Required :

Retina Sensor. Microcontroller (PIC I6F877A), Servo motors (HS322), Transmission module,

Goggles. PCB Board

No of Students Required :

Three students are required

1. GC Ahsan Bilal
2. GC Bilal Haider
3. GC ZeeshanTufail

Special Skills Required :

Microcontroller Programming, Microcontroller interfacing to devices. Sensor calibration. Modulation

Techniques. Proteus professional, PCB Designing.

Appendix B

Visual Studio Code used for Retina Tracking System

```
using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Windows.Forms;
usingAForge.Video.DirectShow;
usingAForge.Video;
using System.IO;
usingAForge.Imaging;
usingAForge.Imaging.ComplexFilters;
usingAForge.Imaging.Filters;
usingSystem.Net.Sockets;

namespaceEye_Tracking_System
{
publicpartialclassMainForm : Form
    {
privateFilterInfoCollectionusbcameras;
privateVideoCaptureDevice camera;

privatePoint start, end, current;

privateInvertinvert = newInvert();
privateThresholdbinaryFilter = newThreshold(220);
privatebool clicked = false;

privateTcpClient client;
privateNetworkStream stream;

privateint x, y, h;
privatefloat width=640, height=480;

privatechar lastCommand1 = ' ',lastCommand2 = ' ';
privatebool firing = false;

//application strt from here
publicMainForm()
        {
//initalize control on form
InitializeComponent();
```

```
//detect video device(cam etc)
camera = newVideoCaptureDevice();
//add video device in combobox
btn_Refresh_Click(null, EventArgs.Empty);

start = newPoint(0, 0);
end = newPoint(640, 480);
            x = y = h = 0;
client = newTcpClient();
        }

//start button is clicked for starting video capturing
privatevoidbtn_Start_Click(object sender, EventArgs e)
        {
try
            {
//if camera device exists
if (camera != null)
                {
//if already camera device is runing stop it
if (camera.IsRunning)
                    {
btn_Start.Text = "Start";
camera.Stop();
camera.NewFrame += null;
                    }
else
                    {
//if not already camera device is runing start it

//a   moniker   identifies   a   COM   object(cam,printer   ports
adressetc) in the directory namespace.
camera                                                          =
newVideoCaptureDevice(usbcameras[tb_cameras.SelectedIndex].Mon
ikerString);

//capture  sceen  and  display  it  camera_NewFrame()  to  capture
also its passing the last capture image frame in arguments
//its  been  done  using  threading  so  continously  it  will  be
calling it untill video stop button cliked
// that the main function which is doing every thing
camera.NewFrame += camera_NewFrame;
btn_Start.Text = "Stop";
camera.Start();
                    }
                }

            }
catch (Exceptionee)
```

```
            {
MessageBox.Show(ee.Message, "Error in Start_Click");
            }
        }

privatevoidcamera_NewFrame(object sender, NewFrameEventArgs e)
        {
try
            {
//make copy of last frame
Bitmap Frame = (Bitmap)e.Frame.Clone();
//rotate frame 180
Frame.RotateFlip(RotateFlipType.RotateNoneFlipY);

//set image taken to big image as it is
pb_InputPicture.Image = (Bitmap)Frame.Clone();

//crop picture only to the small box drawn on big image
Crop c = newCrop(newRectangle(start.X, start.Y, Math.Abs(end.X
- start.X) + 3, Math.Abs(end.Y - start.Y) + 3));

//aaply now the crop criteria to frame
                Frame = c.Apply(Frame);

invert.ApplyInPlace(Frame);
//pictureBox1.Image = (Bitmap)Frame.Clone();

//  apply  grayscale  filter  to  frame  and  assign  it  to
picturebox2
                Frame                                          =
Grayscale.CommonAlgorithms.BT709.Apply(Frame);
                pictureBox2.Image = (Bitmap)Frame.Clone();

// apply binary filter to frame and assign it to pictureBox3
binaryFilter.ThresholdValue = int.Parse(tb_BinThreshold.Text);
binaryFilter.ApplyInPlace(Frame);
                pictureBox3.Image = (Bitmap)Frame.Clone();

// apply BinaryErosion3x3 filter and assign to pictureBox4
// check google BinaryErosion3x3 i don't know
BinaryErosion3x3ee = newBinaryErosion3x3();
ee.ApplyInPlace(Frame);
ee.ApplyInPlace(Frame);
ee.ApplyInPlace(Frame);
                pictureBox4.Image = (Bitmap)Frame.Clone();

//count  objects  in  image  which  are  seprated  black  (not  back
its black ) background
BlobCounterbl = newBlobCounter(Frame);
```

```
inti = bl.ObjectsCount;

try
                {
if (i> 0)
                    {
// get bigest object from image nd take it as eye cordinates

ExtractBiggestBlob fil2 = newExtractBiggestBlob();
                        Frame = fil2.Apply(Frame);

                        x = fil2.BlobPosition.X;
                        y = fil2.BlobPosition.Y;
                        h = fil2.Apply(Frame).Height;

// update eye cond values in textbox for cord
this.Invoke(newAction(() =>
                        {
tb_eyeCoordinates.Text = x.ToString() + ", " + y.ToString() +
", " + h.ToString();
tb_eyeCenter.Text      =      (x+h/2).ToString()      +"      ,
"+(y+h/2).ToString();
                        }
                        ));

floateye_x = (float)x + (float)h / 2;
floateye_y = (float)y + (float)h / 2;

eye_x = (eye_x / width) * 252;
eye_y = (eye_y / height) * 197;

this.Invoke(newAction(() =>
                        {

tb_eyeCenter.Text   =   ((int)eye_x).ToString()   +   "   ,   "   +
((int)eye_y).ToString();

//ajdust value of eye cordinates manually using trakbar up,
down, left and right
if(eye_x<left.Value) // left
                        {
if(lastCommand1!='l')
                            {
p_left.BackColor = Color.Green;
p_right.BackColor = SystemColors.Control;
                                lastCommand1 = 'l';
send("l");
                            }
                        }
```

```
elseif(eye_x>right.Value) // right
                        {
if(lastCommand1!='r')
                            {
p_right.BackColor = Color.Green;
p_left.BackColor = SystemColors.Control;
                                lastCommand1 = 'r';
send("r");
                            }
                        }
else
                        {
if(lastCommand1!='s') // stop
                            {
p_left.BackColor = SystemColors.Control;
p_right.BackColor = SystemColors.Control;
                                lastCommand1 = 's';
send("s");
                            }
                        }

if(eye_y<up.Value) // up
                        {
if(lastCommand2!='u')
                            {
p_up.BackColor = Color.Green;
p_down.BackColor = SystemColors.Control;
                                lastCommand2 = 'u';
send("u");
                            }
                        }
if(eye_y>down.Value) // down
                        {
if(lastCommand2!='d')
                            {
p_down.BackColor = Color.Green;
p_up.BackColor = SystemColors.Control;
                                lastCommand2 = 'd';
send("d");
                            }
                        }
else
                        {
if(lastCommand2!='c')
                            {
p_down.BackColor = SystemColors.Control;
p_up.BackColor = SystemColors.Control;
                                lastCommand2 = 'c';
send("c");
```

```
                                }
                            }
                            }
                            ));
                        }
                    }
catch(Exception)
                    {}

                }
catch(Exceptionee)
                {
MessageBox.Show(ee.Message);
                }
            }

privatevoidbtn_Refresh_Click(object sender, EventArgs e)
            {
try
                {
// camera already runing stop thread
if (camera !=  null&&camera.IsRunning)
                    {
Application.ExitThread();
camera.Stop();
                    }
//clear combobox
tb_cameras.Items.Clear();
//detect video devies
usbcameras                                                              =
newFilterInfoCollection(FilterCategory.VideoInputDevice);

//add name in combobox
foreach (FilterInfovcdinusbcameras)
                    {
tb_cameras.Items.Add(vcd.Name);
                    }

//if some device is detected
if (usbcameras.Count != 0)
                    {
//select first item from combox
tb_cameras.SelectedIndex = 0;
//a  moniker  identifies  a  COM  object  in  the  directory
namespace.
camera.Source = usbcameras[0].MonikerString;
//pictureBox in form
pb_InputPicture.Image = null;
//pictureBox1.Image = null;
```

```
                    pictureBox2.Image = null;
                    pictureBox3.Image = null;
                }
else
                {
//if no device is detected
tb_cameras.SelectedIndex = -1;
camera = null;
                }
            }
catch (Exceptionee)
            {
MessageBox.Show(ee.Message, "Error in Refreshing");
            }
        }

//form close action: close tcp connection and camera
privatevoid          Form1_FormClosing(object          sender,
FormClosingEventArgs e)
        {
try
            {
if (client.Connected)
                {
client.Close();
                }

if (camera!= null&&camera.IsRunning)
                {
Application.ExitThread();
camera.Stop();
                }
            }
catch(Exceptionee)
            {
MessageBox.Show(ee.Message);
            }
        }

//draw big pic
privatevoidpb_Picture_Paint(object sender, PaintEventArgs e)
        {
Point s, nd;
            s = newPoint(); nd = newPoint();
if (start.X<end.X)
{ s.X = start.X; nd.X = end.X; }
else
{ s.X = end.X; nd.X = start.X; }
```

```
if (start.Y<end.Y)
{ s.Y = start.Y; nd.Y = end.Y; }
else
{ s.Y = end.Y; nd.Y = start.Y; }

width = Math.Abs(s.X - nd.X);
height = Math.Abs(s.Y - nd.Y);
e.Graphics.DrawRectangle(newPen(Color.Blue,   2),   s.X,   s.Y,
width, height);
e.Graphics.DrawEllipse(newPen(Color.Red,5),x+start.X,y+start.Y
,h,h);
        }

//detect movement of mouse in pic area and refreach picture
area
privatevoidpb_Picture_MouseMove(object  sender,  MouseEventArgs
e)
        {
current = e.Location;
if (clicked)
            {
end = current;
pb_InputPicture.Refresh();
            }

        }

//click on big pic get the click point and set its value to
start or end textbox on form
privatevoidpb_Picture_Click(object sender, EventArgs e)
        {
clicked = !clicked;
if (clicked)
            {
start = current;
tb_StartPoint.Text = start.X + " , " + start.Y;
            }
else
            {
end = current;
tb_EndPoint.Text = end.X + " , " + end.Y;
//tb_Size.Text = Math.Abs(start.X - end.X).ToString() + " , "
+ Math.Abs(start.Y - end.Y).ToString();
            }
        }

//trackbar for threshold
privatevoid trackBar1_Scroll(object sender, EventArgs e)
        {
```

```
tb_BinThreshold.Text = ((TrackBar)sender).Value.ToString();
        }

//connect using tcp client
privatevoidbtn_connect_Click(object sender, EventArgs e)
        {
try
            {
//if already connected disconnect it
if (client.Connected)
                {
client.Close();
btn_connect.Text = "Connect";
                }
else
                {
//create new connection
client             =            newTcpClient(tb_IpAddress.Text,
int.Parse(tb_portNo.Text));

btn_connect.Text = "Disconnect";

stream = client.GetStream();

//byte[]                          buffer                          =
System.Text.Encoding.ASCII.GetBytes("connected");

//stream.Write(buffer, 0, buffer.Length);
                }
            }
catch(Exceptionee)
            {
MessageBox.Show(ee.Message);
            }
        }

//sending data to network start and stop command "f" start .
"o" stop
privatevoid send(string s)
        {
try
            {
if (client != null&&client.Connected)
                {
stream.Write(System.Text.Encoding.ASCII.GetBytes(s),         0,
s.Length);
                }
            }
catch(Exceptionee)
```

```
            {
MessageBox.Show(ee.Message);
            }
        }

//pic last one
privatevoid pictureBox4_Paint(object sender, PaintEventArgs e)
        {
Pen p = newPen(Color.Magenta,2); //252, 197

e.Graphics.DrawLine(p,          newPoint(left.Value,          0),
newPoint(left.Value, 197));    //left Vertical
e.Graphics.DrawLine(p,          newPoint(right.Value,         0),
newPoint(right.Value, 197)); //Right

e.Graphics.DrawLine(p,  newPoint(0,  up.Value),  newPoint(252,
up.Value));          //Up Horizontal
e.Graphics.DrawLine(p,  newPoint(0,  down.Value),  newPoint(252,
down.Value));      //Down
        }

//strt or stop network data sending
privatevoidbtn_fire_Click(object sender, EventArgs e)
        {
if(firing)
            {
send("f"); //stop firing
            }
else
            {
send("o"); // start firing
            }
firing = !firing;
        }

//binary threshold click
privatevoid label1_Click(object sender, EventArgs e)
        {

        }

//cord start textbox
privatevoidtb_StartPoint_TextChanged(object  sender,  EventArgs
e)
        {

        }

privatevoid panel1_Paint(object sender, PaintEventArgs e)
```

```
        {

        }

privatevoidtb_eyeCoordinates_TextChanged(object          sender,
EventArgs e)
        {

        }

privatevoid label8_Click(object sender, EventArgs e)
        {

        }

privatevoid pictureBox2_Click(object sender, EventArgs e)
        {

        }

privatevoid panel5_Paint(object sender, PaintEventArgs e)
        {

        }
    }
}
```

Appendix C

Timeline

Appendix D

Cost Breakdown

Items Purchased	Cost
Raspberry Pi Model B+	6000/- Rs
2x Camera	3500/- Rs
Wi-Fi Router	1500/- Rs
Weapon Platform	9000/- Rs
Miscellaneous	8000/- Rs
Total	28000/- Rs