SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

B.Tech (IT)

Traffic Signal Control

ANSHUL AHLUWALIA **13BIT0100**

MUDIT MAHESHWARI **13BIT0016**

AGBONJARU JOSHUA **13BIT0265**

K SAI VISWANADH **13BIT0176**

Project Report

of

ITE309-Web Technologies PBL Course

Winter 2014-15

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE • CHENNAI
www.vit.ac.in

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

B.Tech (IT)

Traffic Signal Control

ANSHUL AHLUWALIA 13BIT0100

MUDIT MAHESHWARI 13BIT0016

AGBONJARU JOSHUA 13BIT0265

K SAI VISWANADH 13BIT0176

Project Report

of

ITE309-Web Technologies PBL Course

Winter 2014-15

Submitted to

Faculty : Prof.B.R.Kavitha

Signature:

Date:

Automatic Traffic Control

A Raspberry Pi Project

Submitted to: Prof Kavitha B R

Web Technologies

ITE309

A1 Slot

ANSHUL AHLUWALIA
13BIT0100

MUDIT MAHESHWARI
13BIT0016

AGBONJARU JOSHUA
13BIT0265

K SAI VISWANADH
13BIT0176

Abstract

The purpose of the project is to control the switching of traffic lights using internet technology. As Internet Of Things is the newly developed technology which is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service. The traffic system today consists of an automated computer which controls the timing of the lights. Our project shall enable the traffic policeman to do the same using a web page. The traffic policeman is provided with a unique password which he has to enter in the web page so as to control the traffic lights As computer are prone to fail due to some hardware crashes, then this web page will come in handy for controlling the switching of traffic lights.

We have used raspberry pi and connected it to a model of traffic lights that is represented using LED's and breadboard.

We are able to manage two way traffic light control system.

Table of Contents

Automatic Traffic Control

A Raspberry Pi Project

Introduction

Motivation

This is a **remote traffic monitoring and control system** in which a policeman/admin can control the flow of traffic from a remote location through his computer. He can generate all three traffic signals (red, green & yellow) on a click of mouse only. Also he can divert the flow of traffic by monitoring it on his computer screen through camera.

The **program controls the traffic flow** in both the manners automatic and manually. In automatic mode traffic flow is controlled by predetermined time periods and in manual mode a police man can open or close any lane depending on the traffic density on each lane. Means if on one lane traffic is much more than any other then that lane is open till any other lane has a greater traffic density.

Problem Statement

The main objective of this traffic light controller is to provide sophisticated control and coordination to confirm that traffic moves as smoothly and safely as possible. This project makes use of LED lights for indication purpose and a microcontroller is used for auto changing of signal at specified range of time interval. LED lights gets automatically turns on and off by making corresponding port pin of the microcontroller "HIGH".

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

This project has been designed to help students get started with using the GPIO interface on the Raspberry PI. The principle hardware required to build the traffic lights consists of the following components:

Bread board

LED bulbs (x6)

Jumper Cables(x10)

Resistors

Raspberry Pi

- **Button** Raspberry Pi B+ Model
- Micro Sd Card
- Ethernet Cable
- USB Microphone
- USB Power Cable

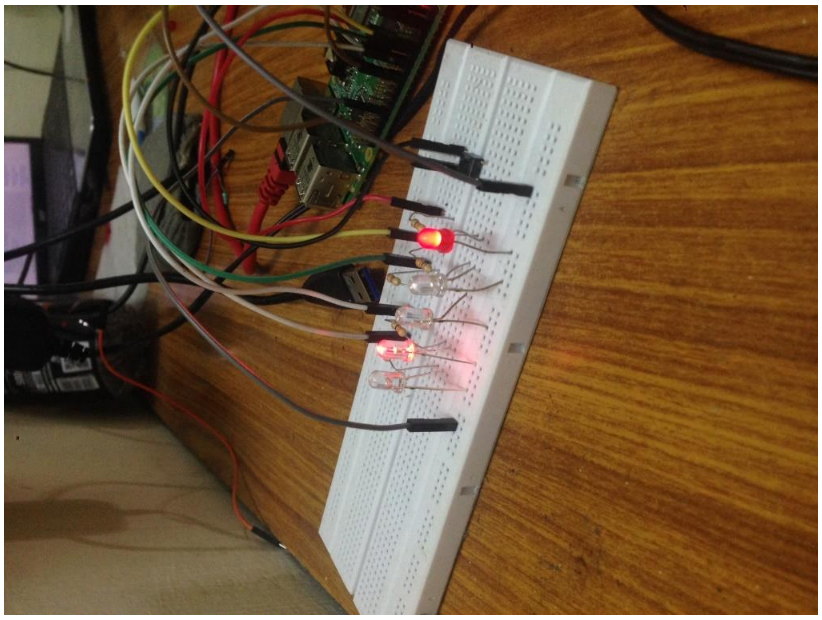


Fig 1.1 Hardware Circuit with Raspberry pi

RASPBERRY PI COMPUTER

The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn programming.

The default username for Raspbian is “pi” (without any quote marks) and the default password is “raspberry” (again, do not include the quote marks).

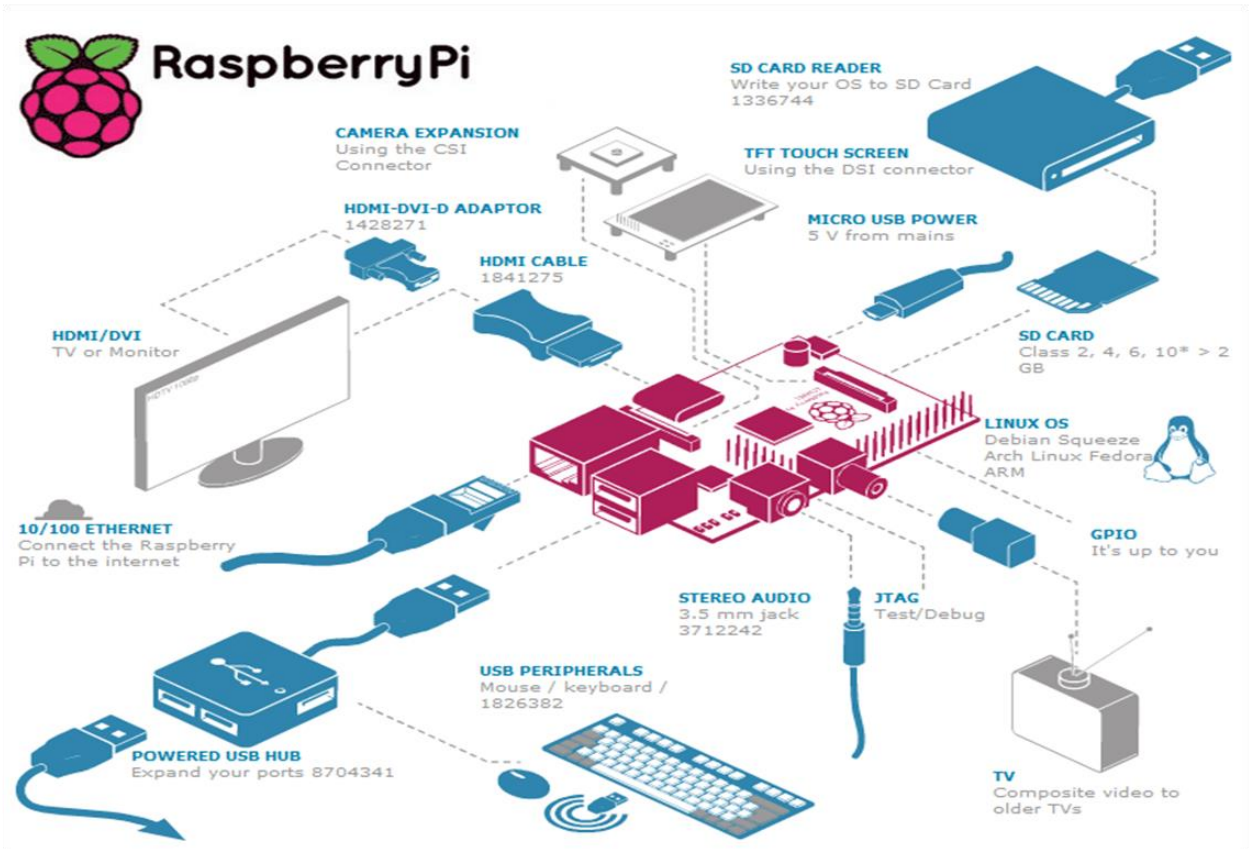


Fig 2.1 Raspberry Pi: Structure and Components

Wiring details

Wiring LEDs

LEDs have polarity and must be wired correctly to work. The diagram shows the polarity of a typical LED. The longer lead is the positive (+) connection and connects to the Anode (The smaller terminal inside the LED).

Also the LED must be wired in series with a resistor to limit the current, typically 220 Ohms is OK. Failure to do this will cause the LED to burn brightly for a short while then burn out.

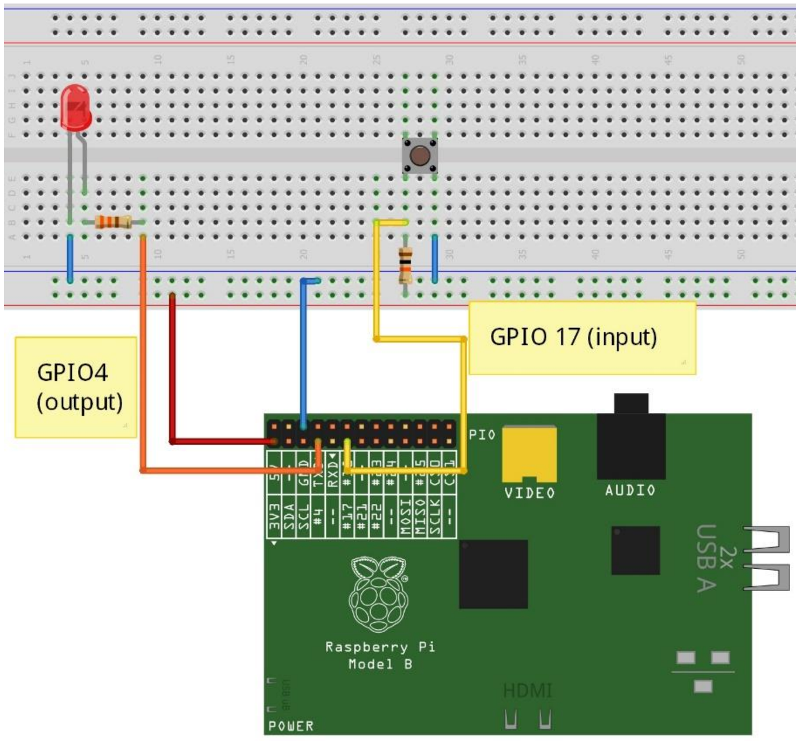


Fig 4.1 Raspberry Pi and GPIO Connections

Software Requirements

- Raspbian OS iso image file
- Win32 Disk Imager
- Putty
- Remote Desktop Connection

Technical Knowledge of Languages

HTML

HyperText Mark-up Language, commonly referred to as **HTML**, is the standard mark-up language used to create web pages. It is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some represent *empty elements* and so are unpaired, for example <img>. The first tag in such a pair is the *start tag*, and the second is the *end tag* (they are also called *opening tags* and *closing tags*).

Web browsers can read HTML files and render them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a mark-up language, rather than a programming language.

HTML elements form the building blocks of all websites. In the project, html is used to embed images and graphics, for tables and making interactive form.

It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

JavaScript

Java Script is a dynamic programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications.

JavaScript is classified as a prototype-based scripting language with dynamic typing and first-class functions. This mix of features makes it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

PHP

PHP (recursive acronym for *PHP: Hypertext Preprocessor*) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

```
h1 { color: white;
  background: orange;
  border: 1px solid bl
  padding: 0 0 0 0;
  font-weight: bold;
}
/* begin: seaside-theme */

body {
  background-color:white;
  color:black;
  font-family:Arial,sans-serif;
  margin: 0 4px 0 0;
  border: 12px solid;
}
```

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.

XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 Specification and by several other related specifications, all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.

XML has appeared as a first-class data type in other languages. The ECMAScript for XML (E4X) extension to the ECMAScript/JavaScript language explicitly defines two specific objects (XML and XMLList) for JavaScript, which support XML document nodes and XML node lists as distinct objects and use a dot-notation specifying parent-child relationships. E4X is supported by the Mozilla 2.5+ browsers (though now deprecated) and Adobe Actionscript, but has not been adopted more universally.

Bash

Bash is a Unix shell written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell Released in 1989, it has been distributed widely as the shell for the GNU operating system and as a default shell on Linux and OS X. It has been ported to Microsoft Windows and distributed with Cygwin and MinGW, to DOS by the DJGPP project, to Novell NetWare and to Android via various terminal emulation applications. In the late 1990s, Bash was a minor player among multiple commonly used shells; at present Bash has overwhelming favour.

Bash is a command processor that typically runs in a text window, where the user types commands that cause actions. Bash can also read commands from a file, called a script. Like all Unix shells, it supports filename wildcarding, piping, here documents, command substitution, variables and control structures for condition-testing and iteration.

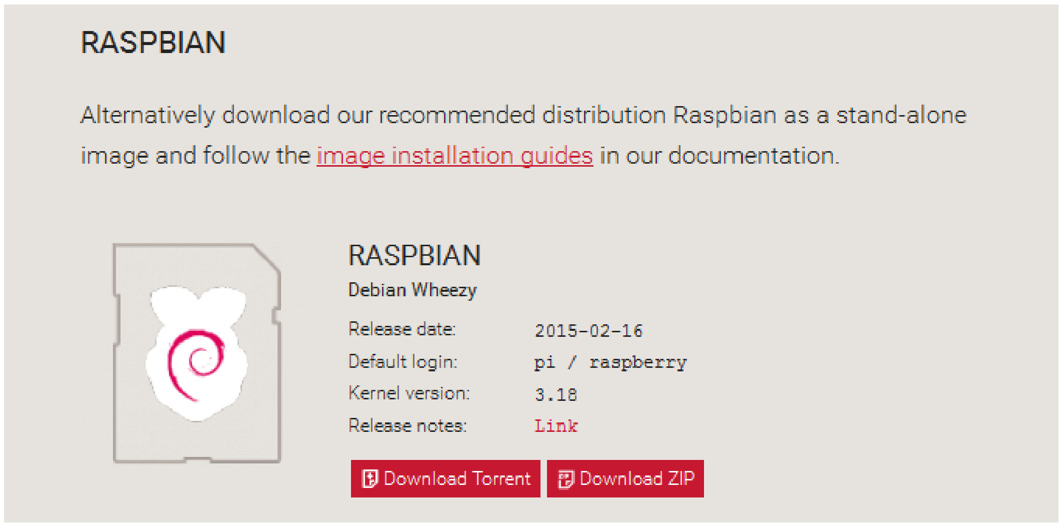
```
# (These are wiringPi pin numbers)

    red=0
  yellow=1
   green=2
  redMan=3
greenMan=4

# The input button

button=8

setup ()
{
  echo Setup
  for i in $red $yellow $green $redMan $greenMan ; do gpio mode  $i out ; done
  for i in $red $yellow $green $redMan $greenMan ; do gpio write $i   0 ; done
  gpio write $green  1
  gpio write $redMan 1
  gpio mode  $button in
}

waitButton ()
{
  echo -n "Waiting for button ... "
  while [ 'gpio read $button' = 1];
```

```
do
    sleep 0.5
  done
  echo "Got it"
gpio write $green 1
}

stopTraffic ()
{
  echo -n "Stopping traffic ... "
gpio write $green 0
  gpio write $yellow 1
  sleep 2
  gpio write $yellow 0
sleep  2
  gpio write $red    1
  sleep 2
  echo "Stopped"
}

walk ()
{
  echo "Walk now ... "
  gpio write $redMan   0
  gpio write $greenMan 1
  sleep 10
  gpio write $red    0
  gpio write $yellow 1
  echo "Walked"
}

graceTime ()
{
  echo "Grace time ... "
  for i in `seq 0 7` ; do
    sleep 0.5
    gpio write $greenMan 0
    gpio write $yellow   0
    sleep 0.5
    gpio write $greenMan 1
    gpio write $yellow   1
  done
  echo "time up!"
}

startTraffic ()
{
  echo "Starting traffic ... "
  gpio write $greenMan 0
  gpio write $redMan   1
  sleep 0.5

  gpio write $yellow 0
  gpio write $green  1
sleep 4
  echo "Going"
}

setup
while true; do
  waitButton
  stopTraffic
  walk
  graceTime
  startTraffic
done
```

INSTALLING RASPBERRY PI

PROCEDURE-

- Download Raspbian OS image file from raspberrypi.org .



RASPBIAN

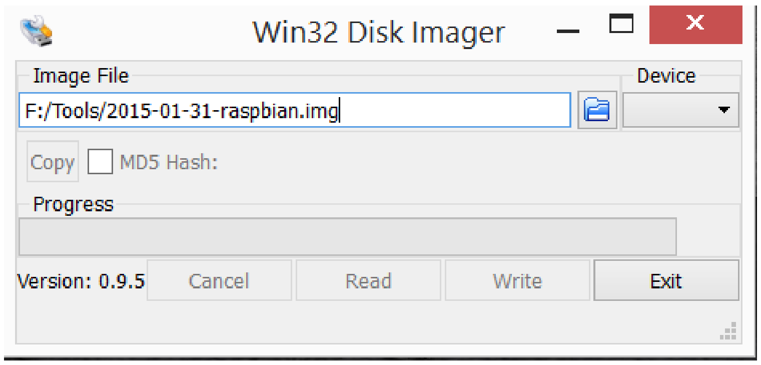
Alternatively download our recommended distribution Raspbian as a stand-alone image and follow the image installation guides in our documentation.

RASPBIAN
Debian Wheezy

Release date:	2015-02-16
Default login:	pi / raspberry
Kernel version:	3.18
Release notes:	Link

Fig 3.1 Raspberry Pi OS: Raspbian

- Mount the downloaded image file on the sd card using Win32 Disk Imager.



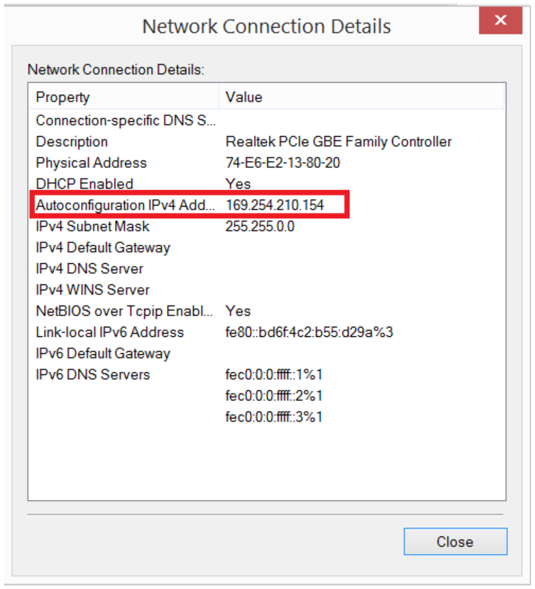
Win32 Disk Imager

Image File: F:/Tools/2015-01-31-raspbian.img

Version: 0.9.5

Fig 3.2

- Insert the SD card into the Raspberry Pi.
- Connect Raspberry Pi to the Laptop using Ethernet Cable and USB Power Cable. Open the Network Properties of Ethernet Connection and note down the AutoConfiguration IP of Raspberry Pi.



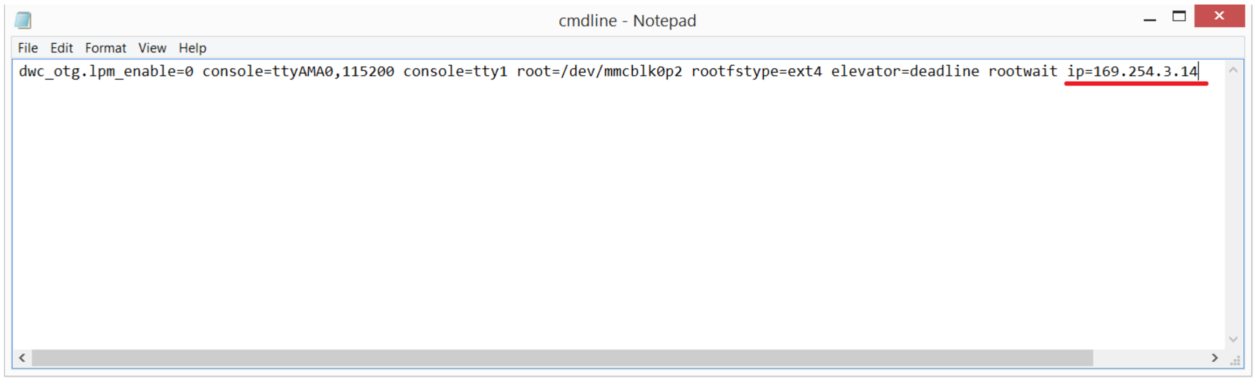
Network Connection Details

Network Connection Details:

Property	Value
Connection-specific DNS S...	
Description	Realtek PCIe GBE Family Controller
Physical Address	74-E6-E2-13-80-20
DHCP Enabled	Yes
Autoconfiguration IPv4 Add...	169.254.210.154
IPv4 Subnet Mask	255.255.0.0
IPv4 Default Gateway	
IPv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Tcpip Enabl...	Yes
Link-local IPv6 Address	fe80::bd6f:4c2:b55:d29a%3
IPv6 Default Gateway	
IPv6 DNS Servers	fec0:0:0:ffff::1%1
	fec0:0:0:ffff::2%1
	fec0:0:0:ffff::3%1

Fig 3.3 Configure IP Address

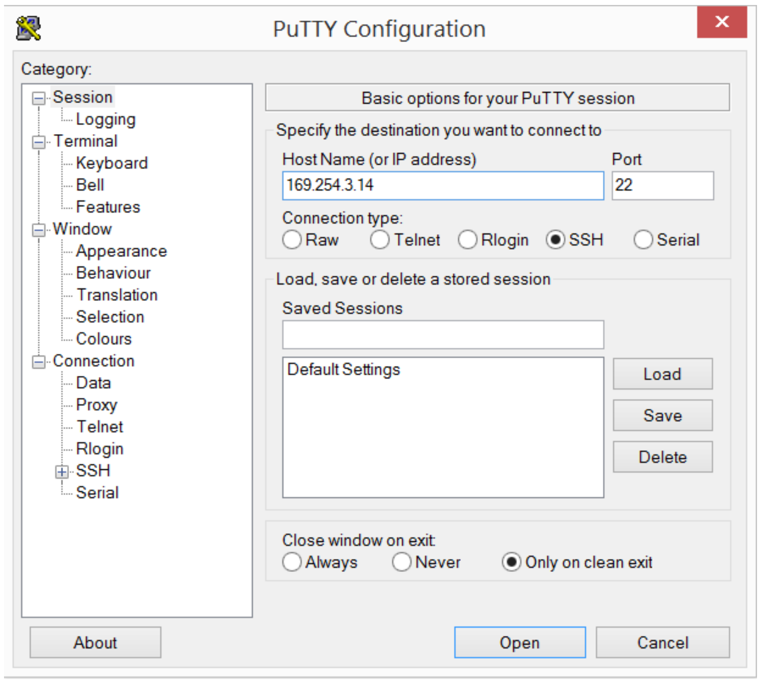
- Insert the SD card into the Laptop and open the file cmdline.txt
- At the end of the file, append ip=169.254.3.14(The first two ports should be the same as AutoConfiguration IP and the last two can be own choice).



```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait ip=169.254.3.14
```

Fig 3.4 Creating Static IP Address

- Again Insert SD card into Raspberry Pi and connect the Pi to the Laptop and launch Putty.



Specify the destination you want to connect to

Host Name (or IP address): 169.254.3.14 Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

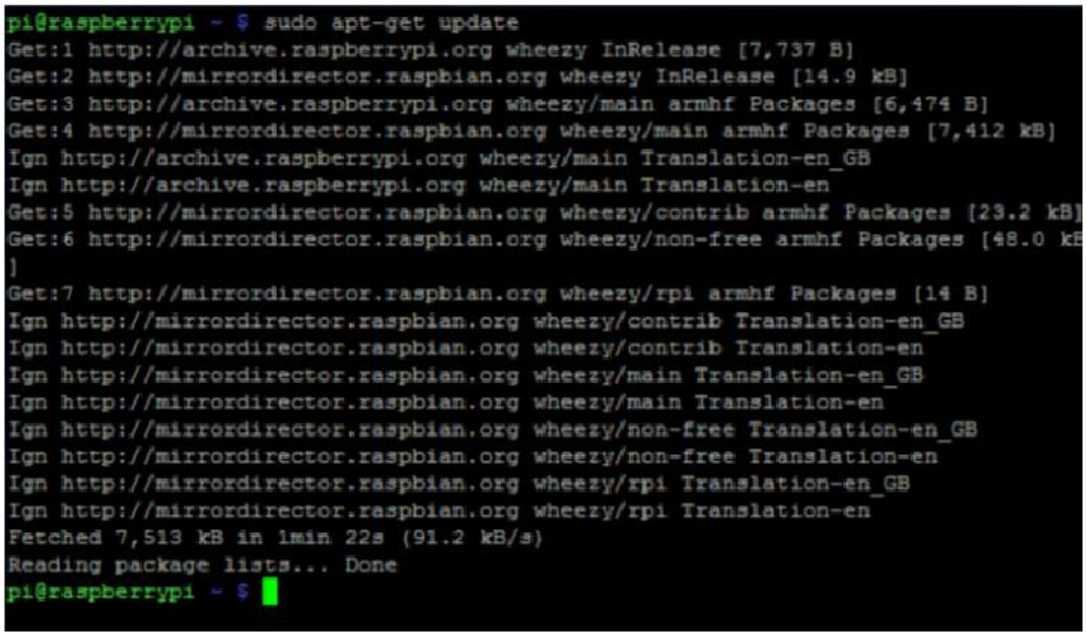
Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

Fig 3.5 PuTTY

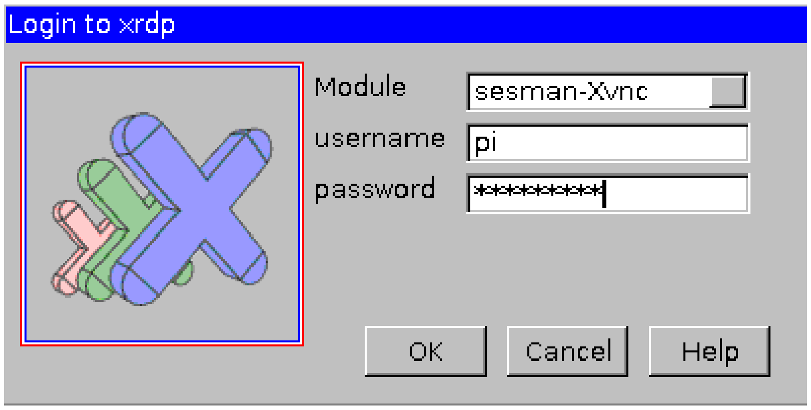
- Give the host name as the ip address saved in cmdline.txt .
- To update the Raspbian software share the internet connection with Ethernet and remove the ip provided in cmdline.txt .
- Open cmd and enter command- ping raspberrypi.mshome.net
- Note the ip address from where the response is received.
- Open Putty and give the above ip address as host name.
- Login as pi and give the password as raspberry.
- Enter command- sudo raspi-config and select Expand Filesystem. Now reboot the raspberry pi using command- sudo shutdown now. Then again login into the raspbian OS using above hostname, login and password.
- Give the command- sudo apt-get update. This will update the Raspbian OS.



```
pi@raspberrypi ~ $ sudo apt-get update
Get:1 http://archive.raspberrypi.org wheezy InRelease [7,737 B]
Get:2 http://mirrordirector.raspbian.org wheezy InRelease [14.9 kB]
Get:3 http://archive.raspberrypi.org wheezy/main armhf Packages [6,474 B]
Get:4 http://mirrordirector.raspbian.org wheezy/main armhf Packages [7,412 kB]
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
Get:5 http://mirrordirector.raspbian.org wheezy/contrib armhf Packages [23.2 kB]
Get:6 http://mirrordirector.raspbian.org wheezy/non-free armhf Packages [48.0 kB
]
Get:7 http://mirrordirector.raspbian.org wheezy/rpi armhf Packages [14 B]
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 7,513 kB in 1min 22s (91.2 kB/s)
Reading package lists... Done
pi@raspberrypi ~ $
```

Fig 3.6

- Give the command- sudo apt-get install xdrp. We will get remote Desktop connection.
- Open Remote Desktop connection in your Laptop.



Login to xrdp

Module: sesman-Xvnc
username: pi
password: **********

Fig 3.7

- Enter the ip address, login and password. You can see the home screen of Raspbian OS.

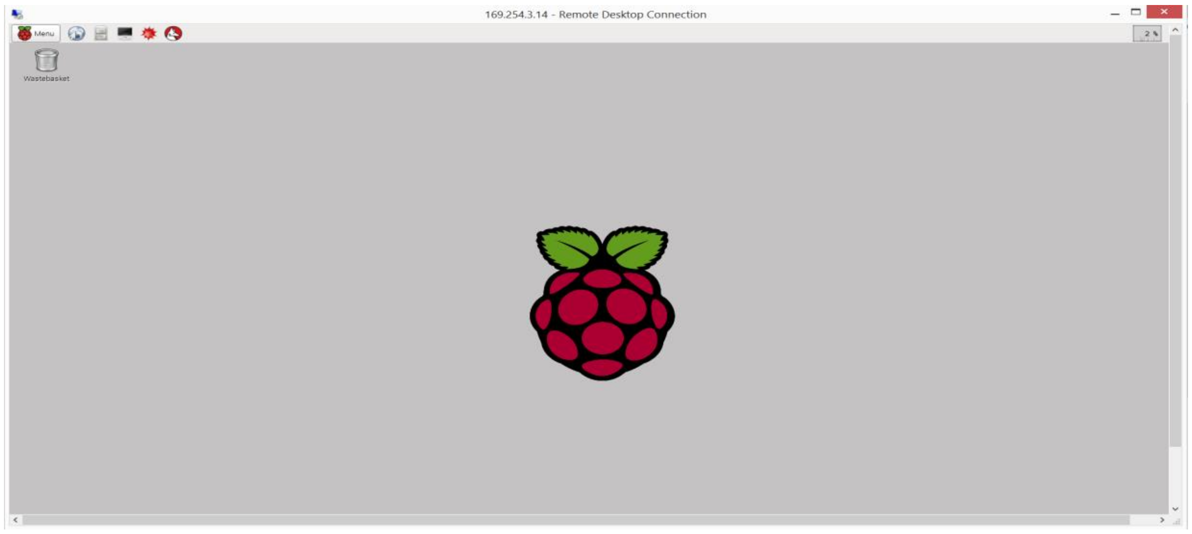


Fig 3.8

Screenshots and Pictures

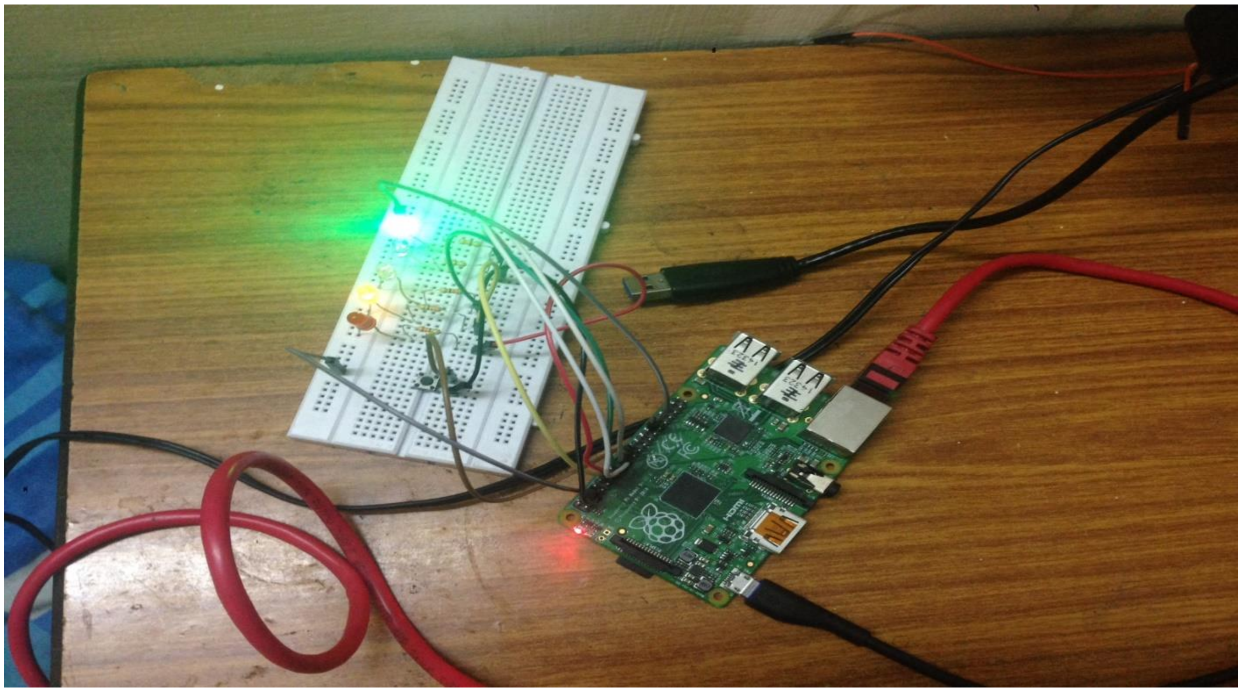
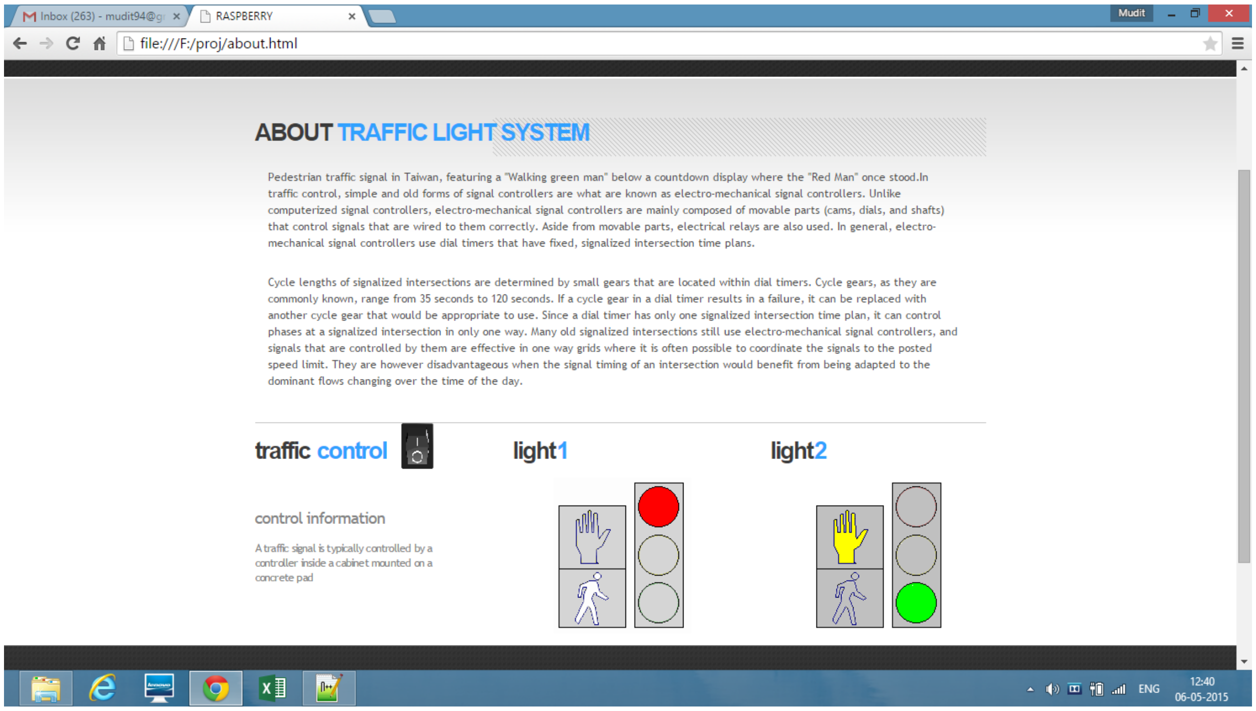


Fig 4.1 Raspberry Pi and GPIO



ABOUT TRAFFIC LIGHT SYSTEM

Pedestrian traffic signal in Taiwan, featuring a "Walking green man" below a countdown display where the "Red Man" once stood.In traffic control, simple and old forms of signal controllers are what are known as electro-mechanical signal controllers. Unlike computerized signal controllers, electro-mechanical signal controllers are mainly composed of movable parts (cams, dials, and shafts) that control signals that are wired to them correctly. Aside from movable parts, electrical relays are also used. In general, electro-mechanical signal controllers use dial timers that have fixed, signalized intersection time plans.

Cycle lengths of signalized intersections are determined by small gears that are located within dial timers. Cycle gears, as they are commonly known, range from 35 seconds to 120 seconds. If a cycle gear in a dial timer results in a failure, it can be replaced with another cycle gear that would be appropriate to use. Since a dial timer has only one signalized intersection time plan, it can control phases at a signalized intersection in only one way. Many old signalized intersections still use electro-mechanical signal controllers, and signals that are controlled by them are effective in one way grids where it is often possible to coordinate the signals to the posted speed limit. They are however disadvantageous when the signal timing of an intersection would benefit from being adapted to the dominant flows changing over the time of the day.

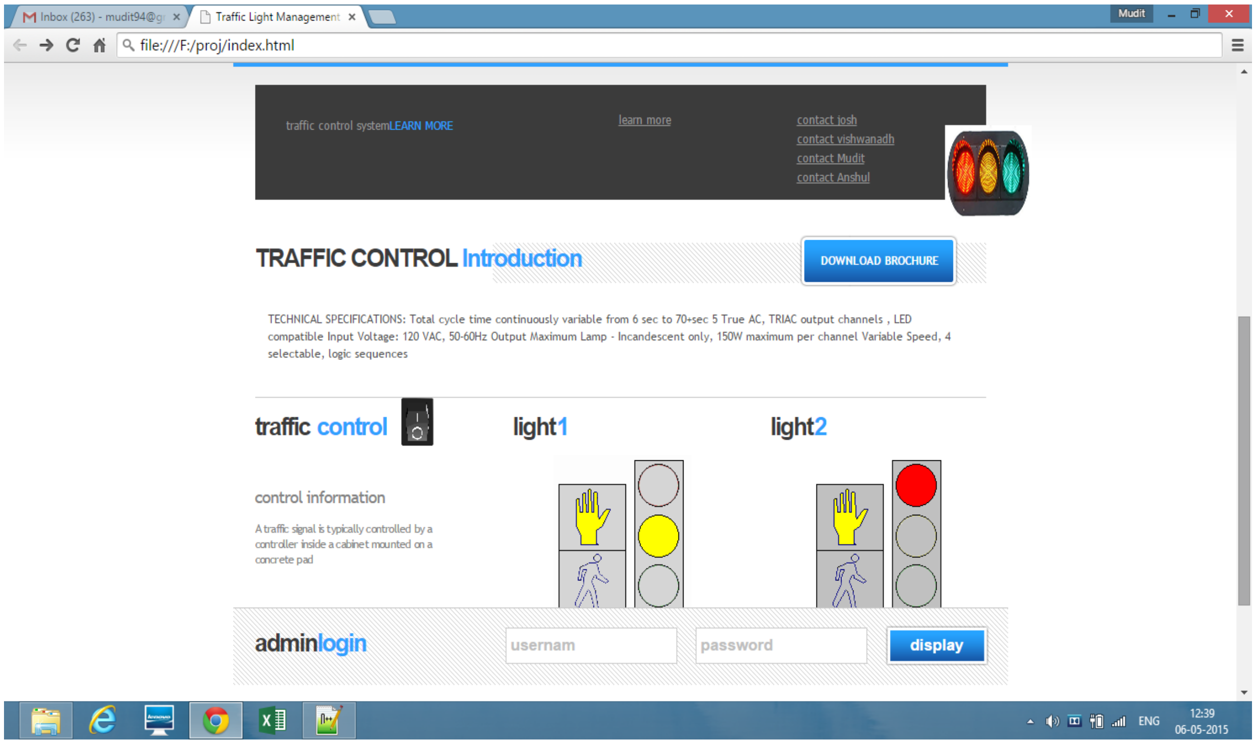
traffic control

control information

A traffic signal is typically controlled by a controller inside a cabinet mounted on a concrete pad

light1

light2



Inbox (263) - mudit94@g | Traffic Light Management

file:///F:/proj/index.html

traffic control systemLEARN MORE

learn more

contact josh
contact vishwanadh
contact Mudit
contact Anshul

TRAFFIC CONTROL Introduction

DOWNLOAD BROCHURE

TECHNICAL SPECIFICATIONS: Total cycle time continuously variable from 6 sec to 70+sec 5 True AC, TRIAC output channels , LED compatible Input Voltage: 120 VAC, 50-60Hz Output Maximum Lamp - Incandescent only, 150W maximum per channel Variable Speed, 4 selectable, logic sequences

traffic control

light1

light2

control information

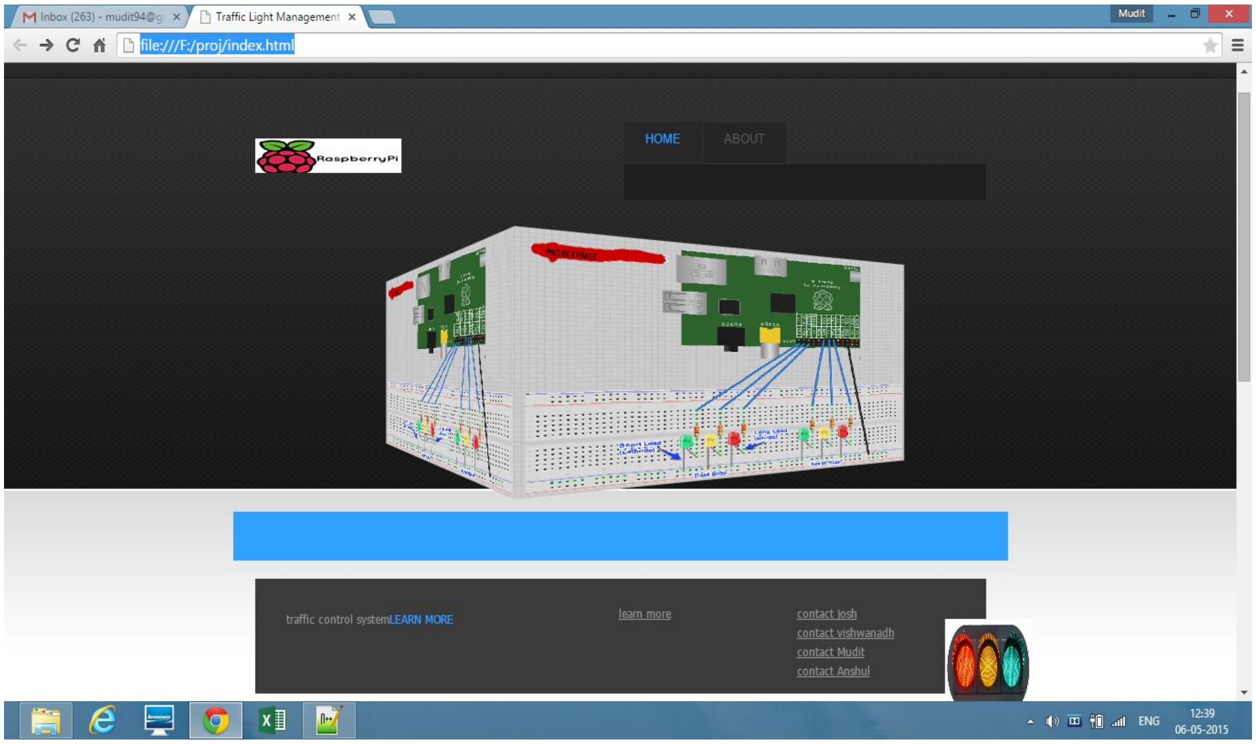
A traffic signal is typically controlled by a controller inside a cabinet mounted on a concrete pad

adminlogin

usernam | password | display

ENG 12:39 06-05-2015

Fig4.3 Webpage Interface



Inbox (263) - mudit94@g | Traffic Light Management

file:///F:/proj/index.html

RaspberryPi

HOME ABOUT

traffic control systemLEARN MORE

learn more

contact josh
contact vishwanadh
contact Mudit
contact Anshul

ENG 12:39 06-05-2015

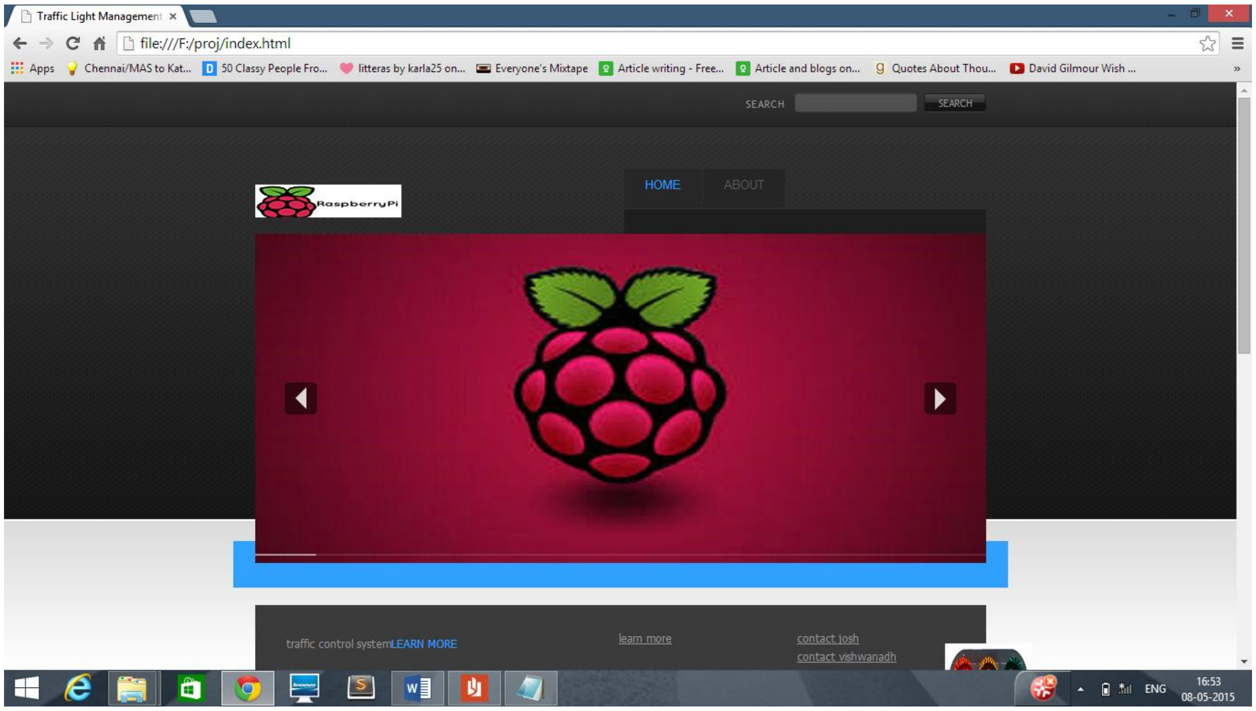
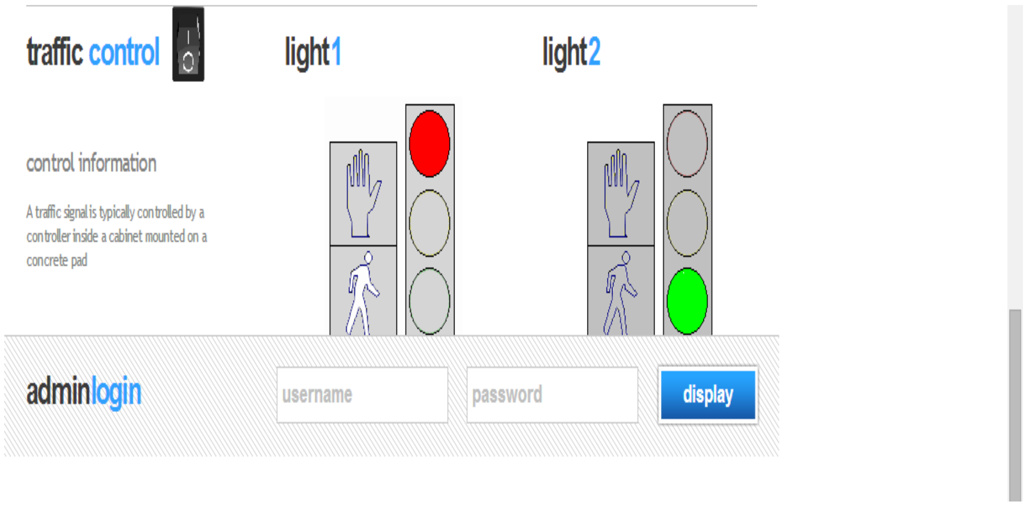


Fig 4.4 Slideshow of Images



A traffic signal is typically controlled by a controller inside a cabinet mounted on a concrete pad

Fig 4.5 Traffic Light Controller

CONCLUSION AND FUTURE SCOPE

1.) For now, this project proposes replacing manual led traffic lights with a well-designed network of automatic traffic signals. This can be extended over a large scale too, say covering Traffic Lights of a particular district or the whole city.
2.) Sensors such as Motion Sensors and Weight sensors can be added to the set up and be linked with the traffic signal. This will enable us to change duration of the lights depending on the traffic.
3.) Camera: Cameras be installed with traffic lights so as to give the picture of traffic at the particular signal, which can facilitate the functioning of the same.

REFERENCES

www.wikipedia.com

www.google.com

www.cnet.com

www.raspberrypi.org

www.hackaday.io