SMART TRAFFIC MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted in partial fulfilment for the award of the degree of

B.TECH

in

Information Technology

By

Ashish Arora	**13BIT0024**
Nakul Sethi	**13BIT0068**
Chirag Arora	**13BIT0247**

Under the Guidance of
Dr. Usha Devi G

School of Information Technology & Engineering

MAY 2017

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled **"SMART TRAFFIC MANAGEMENT SYSTEM"** submitted by me to VIT University, Vellore in partial fulfilment of the requirement for the award of the degree of **B.Tech.(Information Technology)** is a record of bonafide project work carried out by me under the guidance of **Prof. USHA DEVI G**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore Signature of the Candidate:

Date: Ashish Arora | Nakul Sethi | Chirag Arora

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

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled "**SMART TRAFFIC MANAGEMENT SYSTEM"** submitted by Ashish Arora(13BIT0024), Nakul Sethi(13BIT0068), Chirag Arora(13BIT0247) to VIT University, Vellore, in partial fulfillment of the requirement for the award of the degree of B.Tech in Information Technology is a record of bonafide work carried out by him/her under my guidance. The project fulfils the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature **Dr. Dinakaran M**

Dr. Usha Devi G (Associate Professor) **Head of Department**

INTERNAL GUIDE **B.Tech(Information Technology)**

The project work is satisfactory / unsatisfactory

Internal Examiner External Examiner

Date: ……..

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr. G.Vishwanathan, G.V. Selvam, Dr. Anand A. Samuel, Dr. S. Narayanan, Dr. V Raju and Dr. Aswani Kumar Cherukuri, SITE for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to DR. Dinakaran M, Program Chair, Prof. Thippa Reddy G, Year Coordinator, all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is my pleasure to express with deep sense of gratitude to Prof. Usha Devi G, Associate Professor, SITE, VIT University for her constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavour. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of networks and data communications.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place : Vellore

Name of the Student:
Ashish Arora
Nakul Sethi
Chirag Arora

Date :

EXECUTIVE SUMMARY

Setup Raspberry Pi along with Web camera as an IoT Client to click the photos of the traffic junction where cameras can be installed and upload it on Dropbox Server in case any of the sensors (Water Logging) crosses a threshold value or periodically every minute. We can also view these photos remotely, using any mobile device equipped with a browser and having internet access. So the traffic authorities will have remote access to any traffic junction over the internet in real time and this can save the requirement of traffic police at every junction. We will use LED's simulating traffic lights and as soon as the RED Light is on, the IR sensors become active and any one crossing the IR sensor or jumping the red light is captured on the camera. This can help in reducing red light jumps as people will be conscious of the camera being used to monitor traffic and a record being maintained online. Generate a SMS alert as soon as a sensor is triggered or Panic button on the hardware is pressed. This will be generated by Raspberry Pi using internet services and can be sent to Police/Traffic/MCD, as required. The SMS would be sent using internet enabled SMS services. We'll be using the default distribution of Linux (Raspbian) on Raspberry Pi as our O.S. and will write our code and scripts in Python. We also have Light Sensors or LDR's to sense the breakdown of Traffic lights. As soon as any light breaks down, it is processed by Pi and it can send SMS.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Acronym	Full Form
LED	Light Emitting Diode
SMS	Short Message Service
IR	Infrared
LDR	Light Dependent Resistor
PC	Personal Computer
UNIX (UNICS)	Uniplexed Information and Computing Service
OS	Operating System
CPU	Central Processing Unit
ARM	Advanced RISC Machines
IC	Integrated Circuit
AC	Alternating Current
DC	Direct Current
I/O	Input/Output
GND	Ground
MHz	Mega Hertz
IoT	Internet of Things
SRAM	Static Random Access Memory
SPI	Serial Peripheral Interface
ITS	Institute for Telecommunications Sciences.
LCD	Liquid Crystal Display
SOC	System-On-Chip
UI	User Interface
SSH	Secure Shell
SCP	Secure Copy Protocol
VNC	Virtual Network Computing
RDP	Remote Desktop Protocol

CHAPTER 1
INTRODUCTION

1.1 Background

In the last couple of decades, communication technology has developed by leaps and bounds. The use of "Embedded System in Communication" has given rise to many interesting applications. Traditional wireless C.C.T.V. cameras are cheap but anyone with a wireless receiver can view your signal. On the other hand, IP cameras are secure but they can be quite expensive and usually the video quality is poor unless you go for a really expensive model. A lot has been talked regarding the instalment of CCTV throughout the capital and this will help in monitoring and controlling crimes, accidents, challans and would ultimately bring down crime rate, improve the traffic, lifestyle and reduce the pollution, etc.

1.2 Problem Statement

Imagine driving your way through the monstrous bumper to bumper Delhi traffic. Blasting horns, reckless Bluelines, motorcycles whizzing past, endless auto rickshaws, lazy Bullocks vying for space there's no solution to the complete chaos and madness.

Smart Traffic Management System is essentially the application of computer and communications technologies coming in aid of the transport problems. Its technologies enable gathering of data or intelligence and then providing timely feedback to traffic managers and road-users. It results in improved safety to drivers, better traffic efficiency, reduced traffic congestion, improved energy efficiency and environmental quality and enhanced economic productivity.

There would be less number of accidents, respective authority for the road conditions, water logging will be notified immediately.

We are hereby thinking to accomplish stuff that has not been thought by anyone till now. Improving our road safety is one of the most important aspect every country should think about.

Smart Traffic Management System using Raspberry pi:-After rapid growth of raspberry pi this system has been implemented. Compared from existing systems, this system consumes less power & standalone but the drawbacks are when Raspberry pi device fails to deliver the commands in time major problems occurs also we have to remember those commands for every time.

Earlier, we looked into the face of future when we talked about automated devices, which could do anything on instigation of a controller, but today it has become a reality. An raspberry pi device can replace good amount of human working force, moreover humans are more prone to errors and in intensive conditions the probability of error increases whereas, an raspberry pi device can work with diligence, versatility and with almost zero error. Replacing human operators in tasks that involve hard physical or monotonous work. From the convenience of a simple cell phone, a user is able to control and monitor virtually any traffic red light.

Smart Traffic Management System is essentially the application of computer and communications technologies coming in aid of the transport problems. ITS technologies enable gathering of data or intelligence and then providing timely feedback to traffic managers and road-users. ITS results in improved safety to drivers, better traffic

efficiency, reduced traffic congestion, improved energy efficiency and environmental quality and enhanced economic productivity.

There would be less number of accidents, respective authority for the road conditions, water logging will be notified immediately.

We are hereby thinking to accomplish stuff that has not been thought by anyone till now. Improving our road safety is one of the most important aspect every country should think about.

Traffic management system using Raspberry pi:-After rapid growth of raspberry pi this system has been implemented. Compared from existing systems, this system consumes less power & standalone but the drawbacks are when Raspberry pi device fails to deliver the commands in time major problems occurs also we have to remember those commands for every time.

Earlier, we looked into the face of future when we talked about automated devices, which could do anything on instigation of a controller, but today it has become a reality. An raspberry pi device can replace good amount of human working force, moreover humans are more prone to errors and in intensive conditions the probability of error increases whereas, an raspberry pi device can work with diligence, versatility and with almost zero error. Replacing human operators in tasks that involve hard physical or monotonous work. From the convenience of a simple cell phone, a user is able to control and monitor virtually any traffic red light.

1.2 .1 DRAWBACKS OF EXISTING SYSTEM

- No way of monitoring real time traffic.
- No way to manage traffic in case of an emergency.
- No way to stop the red light jumps.
- No way to acknowledge water levels at roads.
- No way to monitor traffic remotely.

1.2 .2 PROPOSED SYSTEM

New technology has IR+LDR sensor, Water level sensor, USB web Camera and sending of a message through IOT.

Advantages

- ✓ Panic button in case of emergency
- ✓ Reduces red light jumps

- ✓ Monitor real time traffic
- ✓ Low cost
- ✓ Low power consumption

1.3 Importance

This System helps in managing traffic. The way traffic is handled in our country is not an optimum way to handle traffic. This system helps to manage traffic congestion and alert the concerned authorities whenever there is a panic situation. This system will also be helpful whenever there is water logging on the roads. This will also reduce the traffic violations as whenever someone crosses the red light the IR sensor will get activated and the message will be sent to the concerned authority and image will also be uploaded. The system also aims in reducing the number of traffic officials at each corner and will help them in monitoring the real time traffic remotely.

1.4 Organization of Report

In chapter 1, we have covered the overall introduction and background of the project followed by the importance of the project . We have then concentrated on the problem that occurred during the development of this project.

In chapter 2, we have covered the overview and planning of the project. We have discussed the challenges and assumptions followed by the hardware and software requirements of the project. After this we have shared the gantt chart and the work breakdown structure of the project.

In chapter 3, we have initiated by mentioning the previous work done that we referred to for our suitability. Then we had a look upon the literature work that was done regarding the concept and designing of this project.

In chapter 4, we take a quick look at the different design approaches that were first taken into account followed by the block diagram and circuit diagram. After that we looked at the different codes, standards and constraints in the project. After looking at the codes and standards we looked at the different test cases in project.

In chapter 5, we take a look at the system implementation. First we looked at the code and architecture development. After that we performed and unit and integration testing.

In chapter 6, we analyzed the various results and the output of the project.

In chapter 7, we have given the conclusion to our project and the future advancements that can be changed in our project according to advancement in technologies.

In chapter 8, we looked at the different references from which we got the idea for the project.

CHAPTER 2
OVERVIEW AND PLANNING

2.1 Proposed System Overview

The project focuses on the, Raspberry Pi is a credit-card sized computer manufactured and designed in the United Kingdom by the Raspberry Pi foundation with the intention of teaching basic computer Science to school students and every other person interested in computer hardware, Programming and DIY-Do-it Yourself projects.

Setup Raspberry Pi as a Web cam server to click the photos of the traffic junction where cameras can be installed and store these photos or upload it on Dropbox in case any of the sensors (Water Logging, etc) crosses a threshold value. We can also view these photos remotely, using any mobile device equipped with a browser and having internet access. So the traffic authorities will have remote access to any traffic junction over the internet in real time and this can save the requirement of traffic police at every junction.

We have used LED's simulating traffic lights and as soon as the RED Light is on, the IR sensors become active and any one crossing the IR sensor or jumping the red light is captured on the camera. This can help in reducing red light jumps as people will be conscious of the camera being used to monitor traffic and a record being maintained online.

Generate a SMS alert as soon as a sensor is triggered. This will be generated by Raspberry Pi using internet services and can be sent to Police/Traffic/MCD, as required. The SMS would be sent using internet enabled SMS services.

We'll be using the default distribution of Linux (Raspbian) on Raspberry Pi as our O.S. and will write our code and scripts in Python.

The main advantage of smart traffic management system is it is better than any other security systems. This system consist of a Water Sensor(which sense if there's water logging on the road), an IR sensor(activated when light is red, and checks if anyone is breaking it), and traffic red light(green and red LED's), a Web Camera to instantly click photos of any situation and another to monitor the whole street at all times. Whenever any of these sensor is activated we receive a message on our device (cell phone) and the mobaXterm software we are using is used to activate our raspberry pi device and system.

We also use a drop box which stores all the pictures that our webcam is clicking. Therefore with the help of this message and picture system we can instantly notify the specific authority so they can respond as soon as possible.

2.2 Challenges

- Managing Real Time Traffic.
- Stopping Traffic Violations (Red light crossing).
- Managing Traffic during Water Logging (During flooding).
- Remote traffic management.
- Constant updates to Traffic Controllers.
- Panic emergency situations (in case of road accidents).
- Traffic lights failure.

2.3 Assumptions

- This system assumes that systems will connect through internet all time to upload and report to the End user (Traffic controller).
- The module is able to communicate with the dropbox server and have enough signal strength to send data in cloud.
- This system assumes that the end user has an internet connection which will unable him to monitor the system remotely.

2.4 Architecture Specifications

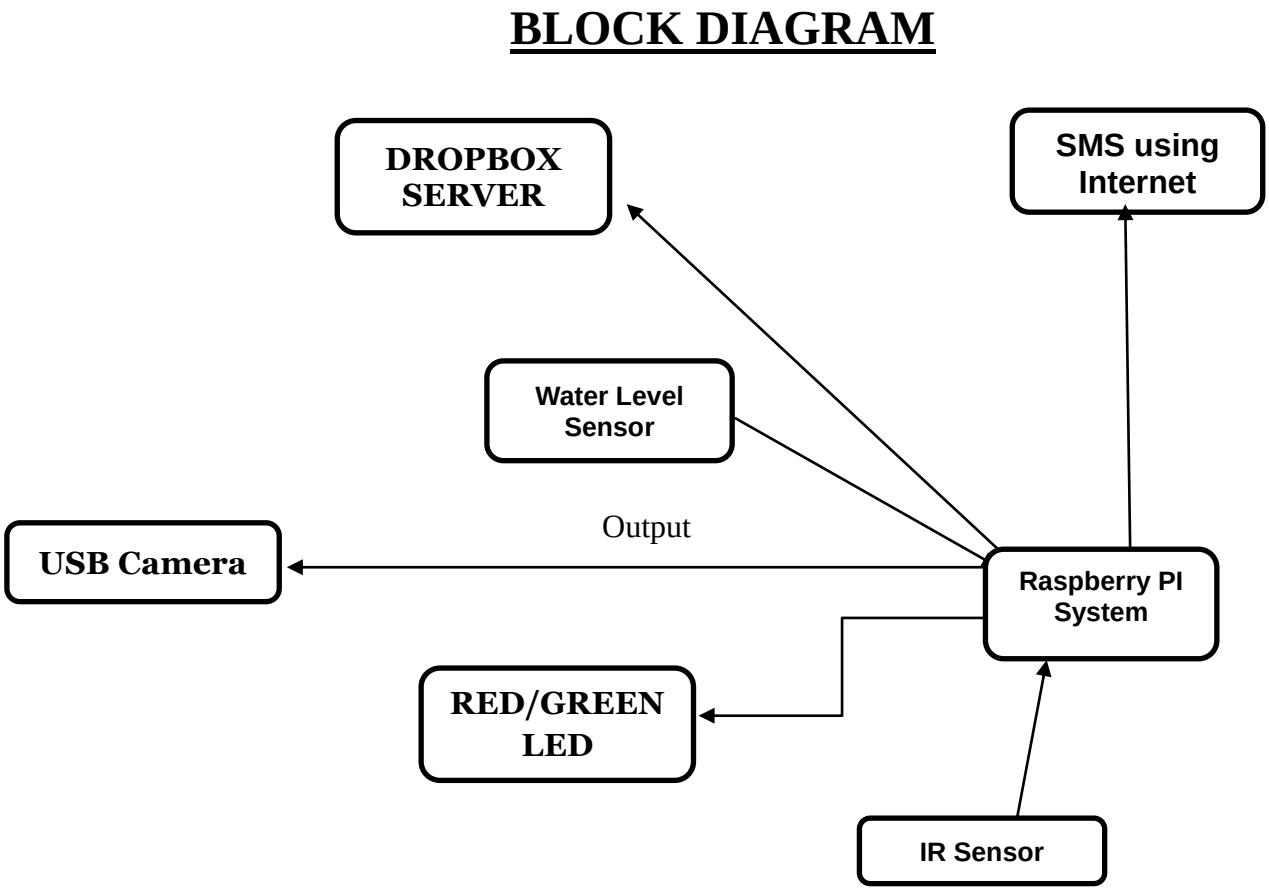


Fig 2.4.1-Block Diagram

2.5 Hardware Requirements

Raspberry Pi device

Raspberry Pi is a credit-card sized computer manufactured and designed in the United Kingdom by the Raspberry Pi foundation with the intention of teaching basic computer science to school students and every other person interested in computer hardware, programming and DIY-Do-it Yourself projects.

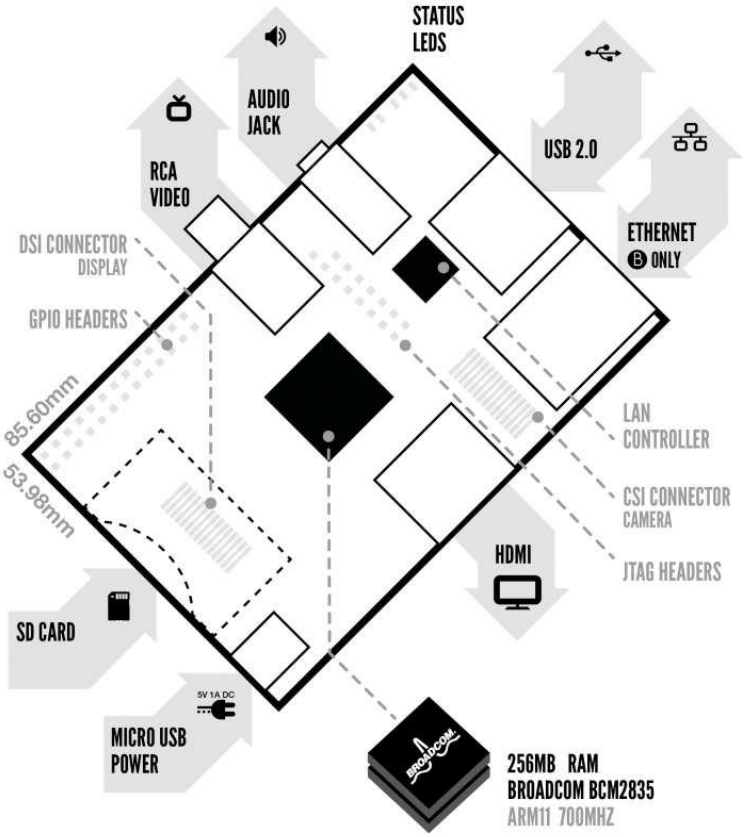


Fig 2.5.1-RaspberryPi3

Infrared Sensor

Infrared (**IR**) is an invisible radiant energy, electromagnetic radiation with longer wavelengths than those of visible light, extending from the nominal red edge of the visible spectrum at 700 Nano meter (frequency 430 THz) to 1000000 nm (300 GHz)[1] (although people can see infrared up to at least 1050 nm in experiments[). Most of the thermal radiation emitted by objects near room temperature is infrared.

Features: high reliability, high radiant density, PB free, Low forward voltage etc.

Battery Supply

Capacitor

Capacitors are common components of electronic circuits, used almost as frequently as resistors. The basic difference between the two is the fact that capacitor resistance (called

reactance) depends on the frequency of the signal passing through the item. The symbol for reactance is X and it can be calculated using the following formula:

$$X_C = \frac{1}{2\pi fC}$$

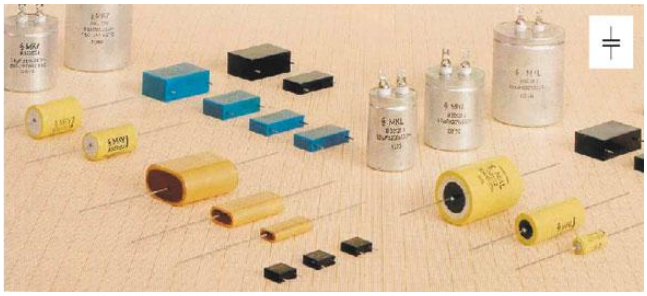


Fig 2.5.2-Capacitors

LED

A **light-emitting diode** (**LED**) is a semiconductor light source. LEDs are used as indicator lamps in many devices and are increasingly used for general lighting. Appearing as practical electronic components in 1962,] early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet, and infrared wavelengths, with very high bright.

Resistors

Resistors are the most commonly used component in electronics and their purpose is to create specified values of current and voltage in a circuit. A number of different resistors are shown in the photos. (The resistors are on millimeter paper, with 1cm spacing to give some idea of the dimensions. some low-power resistors, whilephoto 1.1b shows some higher-power resistors. Resistors with power dissipation below 5 watt (most commonly used types) are cylindrical in shape, with a wire protruding from each end for connecting to a circuit.

Resistors with power dissipation above 5 watt are shown below.

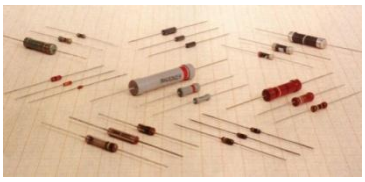


Fig 2.5.3 Low power Resistors

Fig 2.5.4 High power Resistors

- Resistors Markings

Resistance value is marked on the resistor body. Most resistors have 4 bands. The first two bands provide the numbers for the resistance and the third band provides the number of zeros. The fourth band indicates the tolerance. Tolerance values of 5%, 2%, and 1% are most commonly available. Common resistors have 4 bands. These are shown above. First two bands indicate the first two digits of the resistance; third band is the multiplier (number of zeros that are to be added to the number derived from first two bands) and fourth represents the tolerance. Marking the resistance with five bands is used for resistors with tolerance of 2%, 1% and other high-accuracy resistors. First three bands determine the first three digits, fourth is the multiplier and fifth represent the tolerance.

The following table shows the colours used to identify resistor values:

COLOR	DIGIT	MULTIPLIER	TOLERANCE	TC
Silver		x 0.01 ·	±10%	
Gold		x 0.1 ·	±5%	
Black	0	x 1 ·		
Brown	1	x 10 ·	±1%	$\pm 100*10^{-6}/K$
Red	2	x 100 ·	±2%	$\pm 50*10^{-6}/K$
Orange	3	x 1 k·		$\pm 15*10^{-6}/K$
Yellow	4	x 10 k·		$\pm 25*10^{-6}/K$
Green	5	x 100 k·	±0.5%	
Blue	6	x 1 M·	±0.25%	$\pm 10*10^{-6}/K$
Violet	7	x 10 M·	±0.1%	$\pm 5*10^{-6}/K$
Grey	8	x 100 M·		
White	9	x 1 G·		$\pm 1*10^{-6}/K$

** TC - Temp. Coefficient, only for SMD devices

Table 2.5.1- Resistor values

For SMD (Surface Mounted Device) the available space on the resistor is very small. 5% resistors use a 3 digit code, while 1% resistors use a 4 digit code.

Some SMD resistors are made in the shape of small cylinder while the most common type is flat. Cylindrical SMD resistors are marked with six bands - the first five are "read" as with common five-band resistors, while the sixth band determines the Temperature Coefficient (TC), which gives us a value of resistance change upon 1-degree temperature

change. The resistance of flat SMD resistors is marked with digits printed on their upper side. First two digits are the resistance value, while the third digit represents the number of zeros. For example, the printed number 683 stands for 68000 ohm, that is 68k ohm.
For some electrical circuits, the resistor tolerance is not important and it is not specified. In that case, resistors with 5% tolerance can be used. However, devices which require resistors to have a certain amount of accuracy need a specified tolerance.

Voltage Regulators

A **voltage regulator** is designed to automatically maintain a constant voltage level. A voltage regulator may be a simple "feed-forward" design or may include negative feedback control loops. It may use an electromechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

Water Sensor

A water sensor is an electronic device that is designed to detect the presence of water and provide an alert in time to allow the prevention of water damage. A common design is a small cable or device that lies flat on a floor and relies on the electrical conductivity of water to decrease the resistance across two contacts. The device then sounds an audible alarm together with providing onward signalling in the presence of enough water to bridge the contacts. These are useful in a normally occupied area near any infrastructure that has the potential to leak water, such as HVAC, water pipes, drain pipes, vending machines, dehumidifiers, or water tanks.

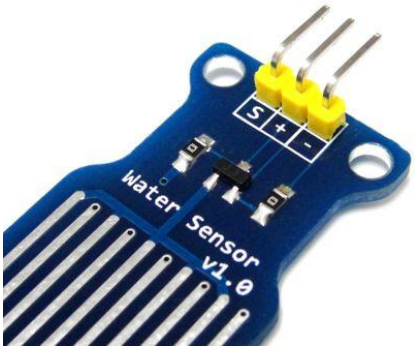


Fig 2.5.5 Water Sensor

2.6 Software Requirements

Operating System

The Raspberry Pi primarily uses Linux kernel-based operating systems. The ARM11 is based on version 6 of the ARM which is no longer supported by several popular versions of Linux, including Ubuntu. The install manager for Raspberry Pi is NOOBS. The OSs included with NOOBS are:

- Arch Linux ARM
- Open ELEC
- Pi dora (Fedora Remix)
- Rasp bmc and the XBMC open source digital media center
- RISC OS – The operating system of the first ARM-based computer
- Raspbian (recommended) – Maintained independently of the Foundation; based on ARM hard-float (armhf)- Debian 7 'Wheezy' architecture port, that was designed for a newer ARMv7 processor whose binaries would not work on the Raspberry Pi, but Raspbian is compiled for the ARMv6 instruction set of the Raspberry Pi making it work but with slower performance. It provides some available software packages, pre-compiled software bundles. A minimum size of 2 GB SD card is required, but a 4 GB SD card or above is recommended. There is a Pi Store for exchanging Programs. The Raspbian Server Edition (RSEv2.4)', is a stripped version with other Software packages bundled as compared to the usual desktop computer oriented Raspbian.

Putty/MobaXterm Software

- Putty is a free and open source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "PuTTY" has no definitive meaning. PUTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems. Official ports are available for some Unix-like platforms, with work-in-progress ports to Classic Mac OS and Mac OS X, and unofficial ports have been contributed to platforms such as Symbian,Windows Mobile and Windows Phone.

- MobaXterm is an enhanced terminal for Windows with an X11 server, a tabbed SSH client and several other network tools for remote computing (VNC, RDP, telnet, rlogin). MobaXterm brings all the essential Unix commands to Windows desktop, in a single portable exe file which works out of the box.

MobaXterm/Putty: How to start a SSH session from the command line

1) Type the path to .exe here.

2) Then type the connection type you wish to use (i.e. -ssh, -telnet, -rlogin, -raw)

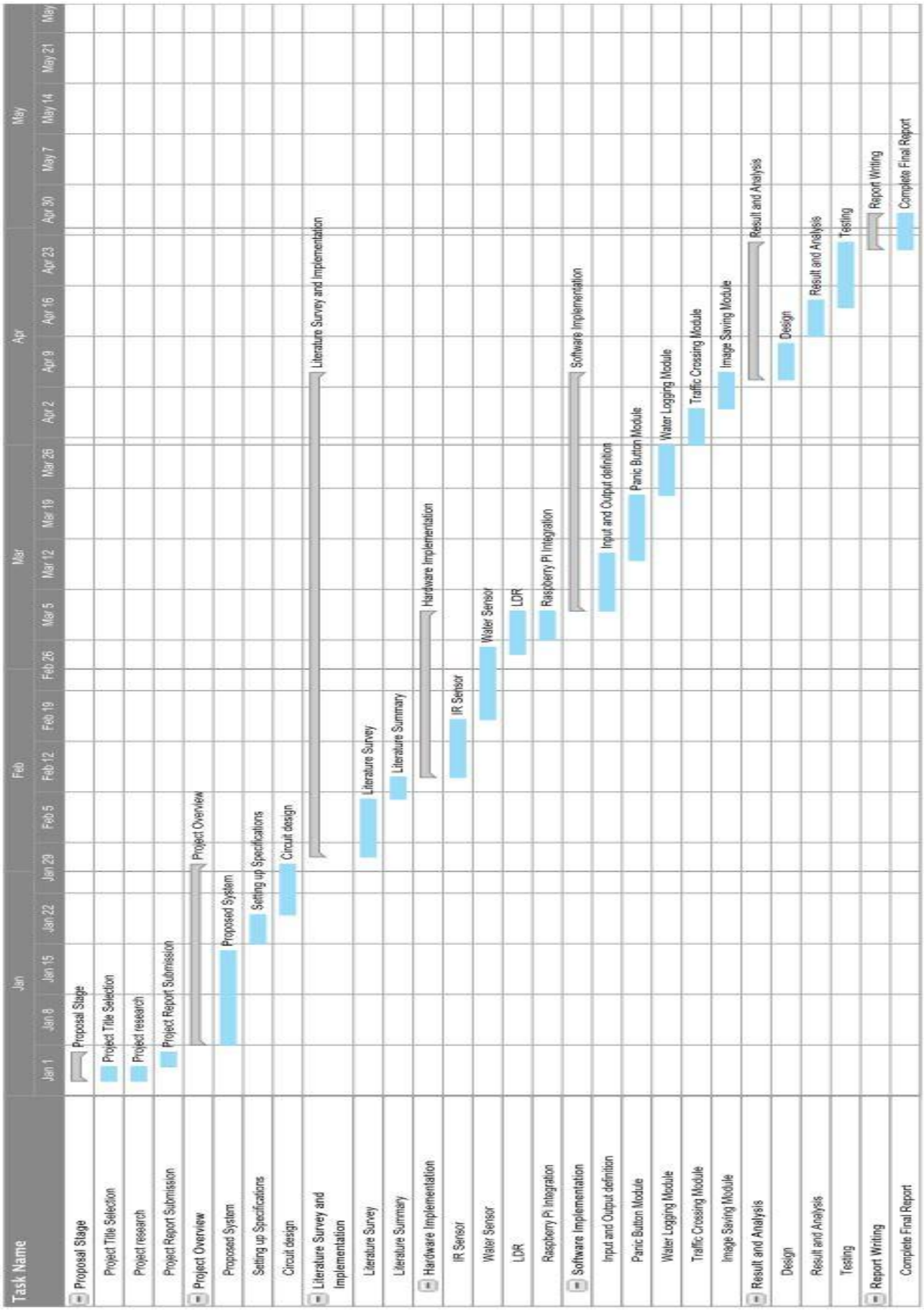
3) Type the username…

4) Then type '@' followed by the server IP address.

5) Finally, type the port number to connect to, then press <Enter>

2.7 Project Schedule (Gantt chart)

Fig 2.7.1- Gantt Chart



2.8 Work Breakdown Structure

	A	B	C	D
1	**Task Name**	**Start date**	**End Date**	**Duration**
2	**Proposal Stage**	**01/03/17**	**01/06/17**	**4d**
3	Project Title Selection	01/03/17	01/04/17	2d
4	Project research	01/03/17	01/04/17	2d
5	Project Report Submission	01/05/17	01/06/17	2d
6	**Project Overview**	**01/08/17**	**02/01/17**	**19d**
7	Proposed System	01/08/17	01/20/17	11d
8	Setting up Specifications	01/22/17	01/25/17	4d
9	Circuit design	01/26/17	02/01/17	5d
10	**Literature Survey and Implementation**	**02/03/17**	**04/10/17**	**47d**
11	Literature Survey	02/03/17	02/10/17	6d
12	Literature Summary	02/11/17	02/13/17	2d
13	**Hardware Implementation**	**02/14/17**	**03/08/17**	**17d**
14	IR Sensor	02/14/17	02/21/17	6d
15	Water Sensor	02/22/17	03/03/17	8d
16	LDR	03/03/17	03/08/17	4d
17	Raspberry Pi Integration	03/05/17	03/08/17	4d
18	**Software Implementation**	**03/09/17**	**04/10/17**	**23d**
19	Input and Output definition	03/09/17	03/16/17	6d
20	Panic Button Module	03/16/17	03/24/17	7d
21	Water Logging Module	03/25/17	03/31/17	6d
22	Traffic Crossing Module	04/01/17	04/05/17	4d
23	Image Saving Module	04/06/17	04/10/17	3d
24	**Result and Analysis**	04/10/17	04/28/17	15d
25	Design	04/10/17	04/14/17	5d
26	Result and Analysis	04/16/17	04/20/17	5d
27	Testing	04/20/17	04/28/17	7d
28	**Report Writing**	04/28/17	05/02/17	3d
29	Complete Final Report	04/28/17	05/02/17	3d

Table 2.8.1

CHAPTER 3

Literature Survey and Summary

3.1 Literature Survey

Now-a-days there is a huge advancement in the communication sector. Almost all people now-a-days have access to mobile phones and thus the world has indeed become a global village. At any given moment, any person across the world can be contacted with the help of a mobile phone. But mobile phones can not only used for the calling and sending SMS purposes but also new ideas can be generated and techniques can be developed from it that can further enhance its capabilities. Raspberry pi enables us to control the whole functionality by processing the activities. In this present age, safety has become an essential issue for most of the people especially in the field of road safety. Some people try to break the traffic rules of government, there are many road accidents happening, road safety measure are not very much concerned in India. To overcome the safety threat , the government assign the traffic officials to their respected duties but this doesn't help much. In this project we have implemented safety of the people walking on the road, in their vehicles and less effort for our traffic authorities by using Raspberry pi and raspbian technology which will be more secure than other systems.

The Raspberry Pi is a small, barebones computer developed by The Raspberry Pi Foundation, a UK charity, with the intention of providing low-cost computers and free software to students. The ultimate goal is to improve the traffic surveillance system using various sensors in order to make our road much safer to live. The Raspberry pi has 3 models. Raspberry pi 1, Raspberry 2 , Raspberry pi 3 each of which have different specifications .We are using the raspberry pi 3 as it is the latest version and costs less than other 2 models and have better specification than any of the raspberry pi 1 and 2. Some specifications of raspberry pi 3 are it have 1 GB LPDDR2-900 SDRAM, 4 USB ports, 1.2 GHZ quad-core ARM cortex A53 CPU, SOC: Broadcom BCM2837(which is 50% faster than the raspberry pi2).

3.2 Literature Summary

The issue for this project arises from the problems faced in some parts of India on roads. The project is combination of 3 different projects which can solve many problems which are Water logging, Panic button and Red light crossing. This project helps in capturing photos when a vehicle crosses the road when light is red. Another function is for detecting water logging which detects the level of water if it has reached certain level is reached and a panic button for alarming the authorities if some problem arises.

There are two parts that involves in accomplishing this project which are software and hardware implementation. Software part is the coding which starts the program and stores the data and circuit design and hardware parts consists of components required to run this project.

CHAPTER 4

SYSTEM DESIGN

4.1 High-Level Design

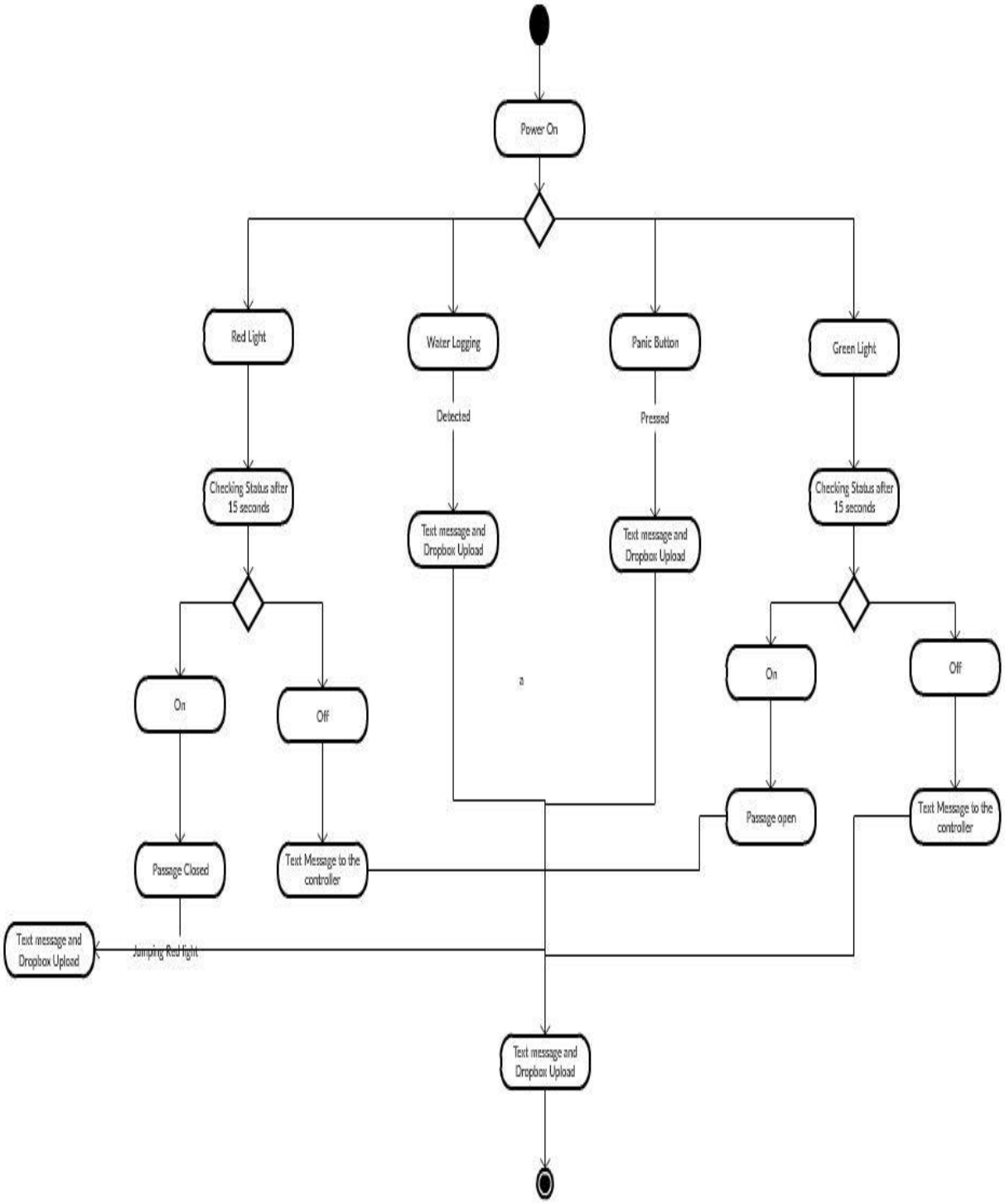


Fig 4.1.1-Activity Diagram

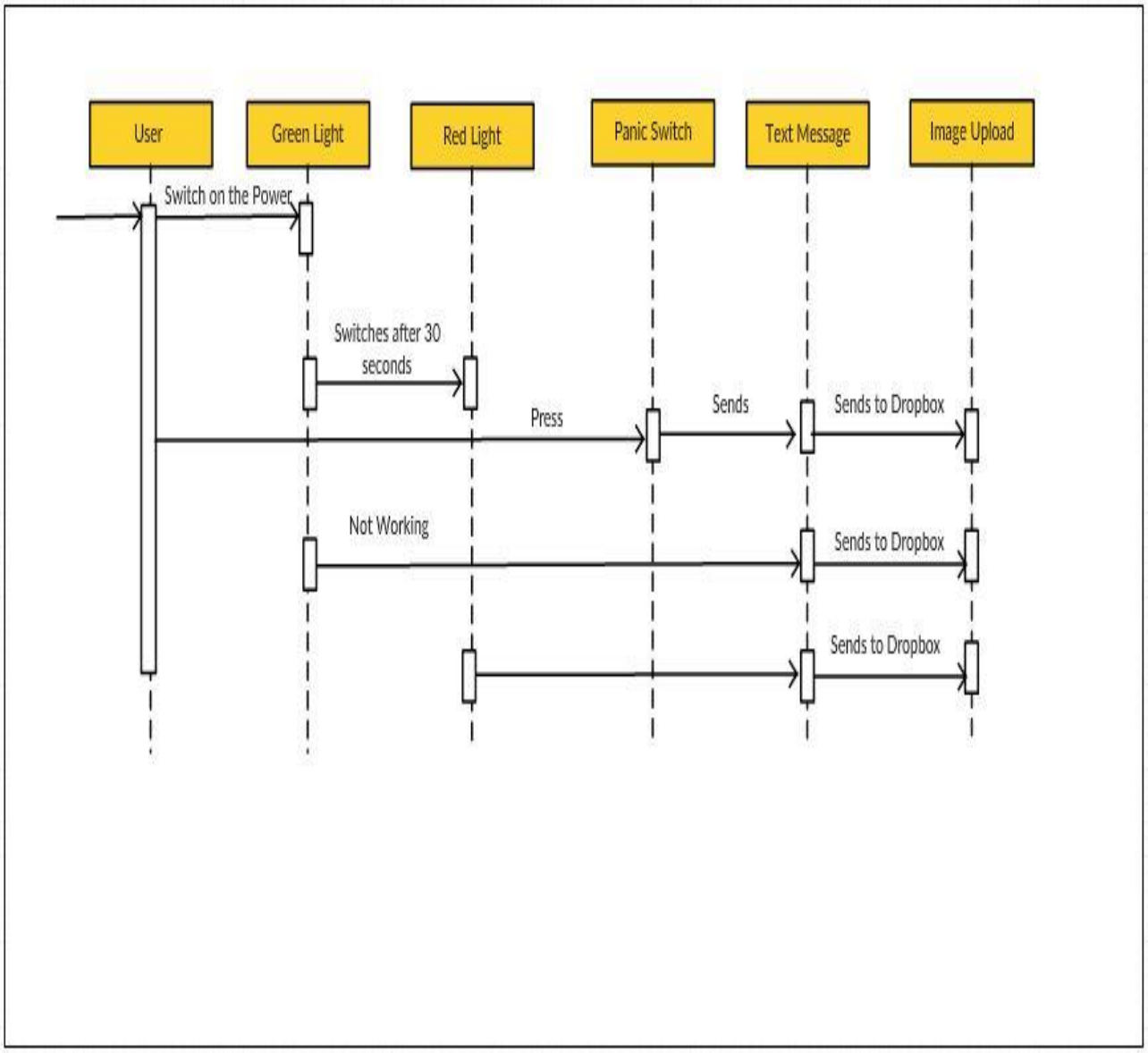


Fig 4.1.2-Sequence Diagram

4.2 Low-Level Design

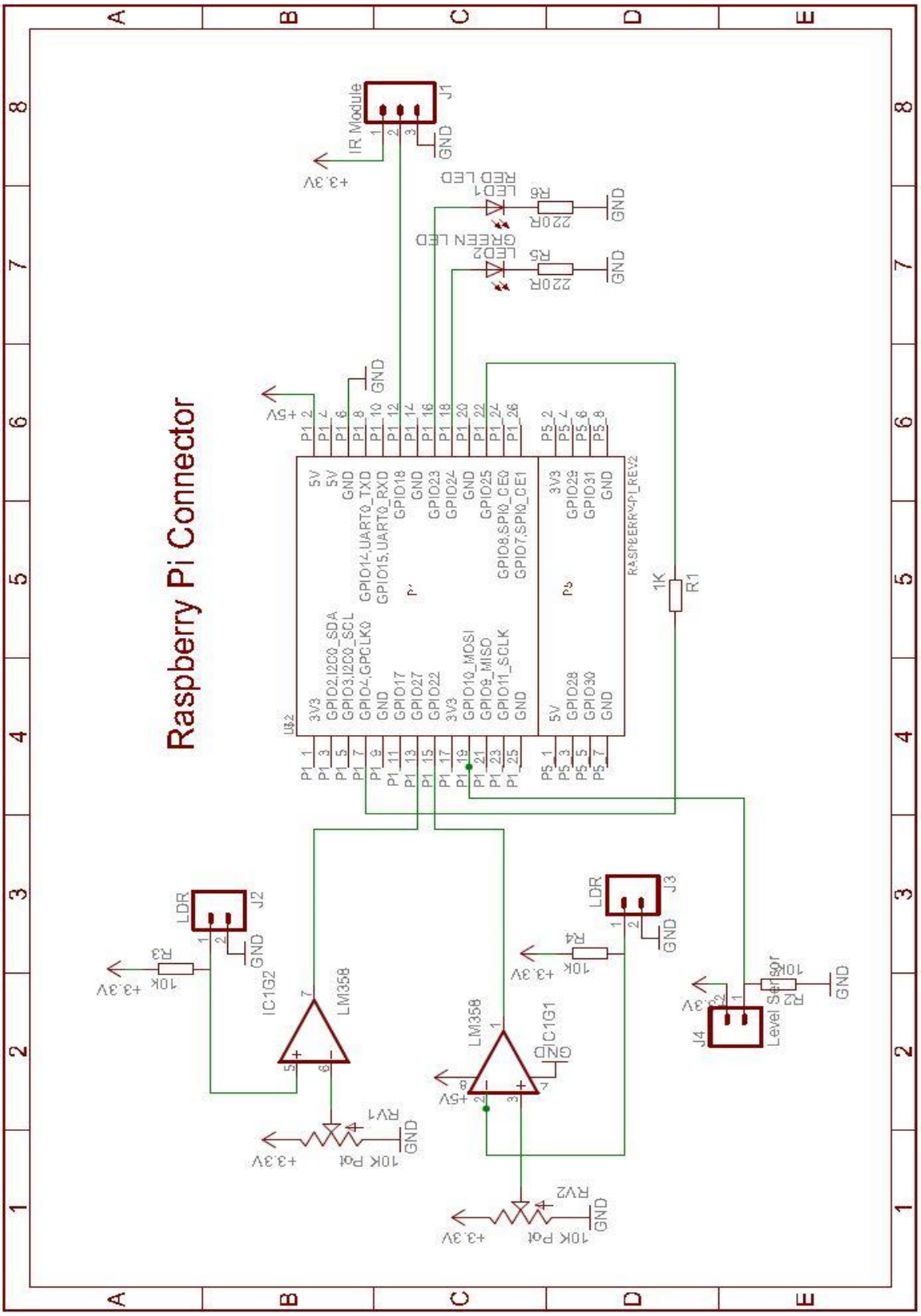


Fig 4.2.1- Raspberry Pi Connector

4.3 Codes and Standards

Work shall consist of furnishing and installing, modifying, removing or salvaging one or more traffic signal systems, electrical equipment on structures, cameras on streets, falsework lighting, partial installations for future systems, or combinations thereof, all as required by the Drawings, and as specified. All necessary labor and equipment to provide fully functioning traffic signals, intersection lighting is included.

Standard: Traffic control devices shall be defined as all signs, signals, markings, and other devices used to regulate, warn, or guide traffic, placed on, over, or adjacent to a street, highway, pedestrian facility, or bikeway by authority of a public agency having jurisdiction.

Purpose of Traffic Control Devices

Support: The purpose of traffic control devices and red light cameras, as well as the principles for their use, is to promote highway safety and efficiency by providing for the orderly movement of all road users on streets and highways throughout the Nation.

To be effective, a traffic control device should meet five basic requirements:

A. Fulfill a need

B. Command attention;

C. Convey a clear, simple meaning;

D. Command respect from road users;

E. Give adequate time for proper response.

F. Capture proper and clear photos.

G. Detects accurate water level.

H. Working of Panic button.

4.4 Test Cases Generation

Test cases:

1. In case of Green Light.

a. Working of green light is checked by removing of switch which sees that whether green is active or not before it has to be turned green if not then sends a message.
b. Checking of water level is determined at a certain level.
c. Capturing of photos after every minute and storing it in dropbox.
d. Working of panic button in any condition and capturing its photo at the moment.

2. In case of Red Light

a. Working of red light is checked by removing of switch which sees that whether red is active or not, if not then sends a message.
b. Checking the working of water logging and panic button and sending text message at the same time.

3. Working of video surveillance at every moment.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Code and/or Architecture Development

5.1.1 Architecture Development

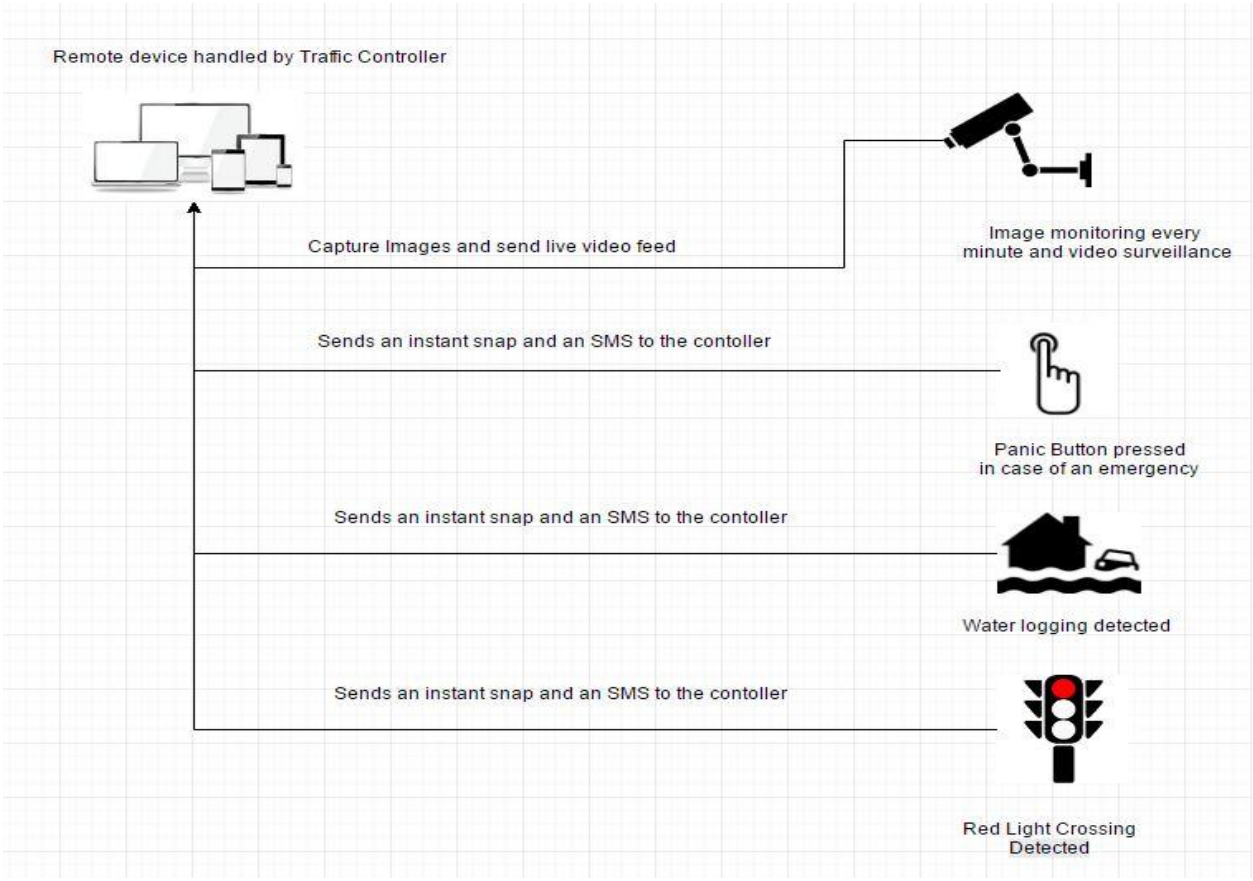


Fig 5.1.1- System Architecture

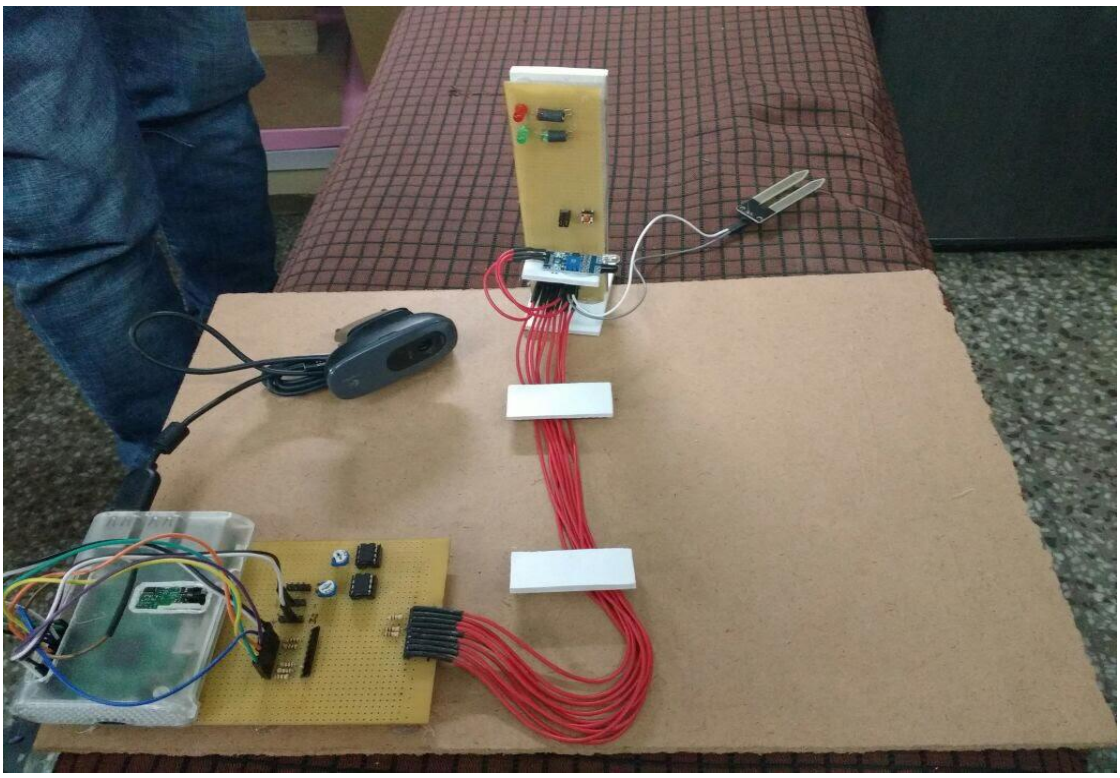


Fig 5.1.2-Implementation

5.1.2 Source Codes

Save1.sh- This is the shell script used for saving images when WATER LOGGING is detected, to the local memory and uploading it to the dropbox.

```
#!/bin/bash
DATE=W$(date +"%Y-%m-%d_%H%M%S")
sudo fswebcam -r 640x480 /home/pi/script/images/$DATE.jpg
/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
/home/pi/script/images/$DATE.jpg $DATE.jpg
```

Save2.sh- This is the shell script used for saving images when PANIC BUTTON is detected, to the local memory and uploading it to the dropbox.

```
#!/bin/bash
DATE=P$(date +"%Y-%m-%d_%H%M%S")
```

```
sudo fswebcam -r 640x480 /home/pi/script/images/$DATE.jpg
/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
/home/pi/script/images/$DATE.jpg $DATE.jpg
```

Save3.sh- This is the shell script used for saving images when RED LIGHT CROSSING is detected, to the local memory and uploading it to the dropbox.

```
#!/bin/bash
DATE=I$(date +"%Y-%m-%d_%H%M%S")
sudo fswebcam -r 640x480 /home/pi/script/images/$DATE.jpg
/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
/home/pi/script/images/$DATE.jpg $DATE.jpg
```

Save4.sh- This is the shell script used for saving images AFTER EVERY MINUTE, to the local memory and uploading it to the dropbox.

```
#!/bin/bash
DATE=M$(date +"%Y-%m-%d_%H%M%S")
sudo fswebcam -r 640x480 /home/pi/script/images/$DATE.jpg
/home/pi/Dropbox-Uploader/dropbox_uploader.sh upload
/home/pi/script/images/$DATE.jpg $DATE.jpg
```

Script.py- This is the main script used for running the program in the raspberrypi.

```
import os
import time
import urllib
import urllib2
import cookielib
import subprocess
from subprocess import call
```

```
import RPi.GPIO as gpio

#=================================================================

#message = input("Enter Message :")
message = 'System Started'
#number = input("Enter reciever phone number :")
number = '9873707004'
flag1=100
flag2=100
flag3=100
flag4=100
count=0
#=================================================================
def sendSMS(uname, hashCode, numbers, sender, message):
        data =  urllib.urlencode({'username': uname, 'hash': hashCode, 'numbers':
numbers, 'message' : message, 'sender': sender})
        data = data.encode('utf-8')
        request = urllib2.Request("http://api.textlocal.in/send/?")
        f = urllib2.urlopen(request, data)#I need to use urllib2 and urllib because urlopen
in urllib2 can take a request class but urllib does not include a function like urlencode.
        fr = f.read()
        return(fr)
#=================================================================
def sw1_detect(pin):#switch for water logging
        print 'Water Logging Detected'
        gpio.remove_event_detect(19)#The water sensor is connected to the general
purpose I/O pin 19
```

```
        global flag1
        if (flag1==100):
                flag1=0
                subprocess.call("./save1.sh", shell=False)
                print('W button inactive')
                resp =  sendSMS('bishan@gmail.com', 'satnamWAHEGURU123', number,
'Water Logging Detected', 'TXTLCL')
                print (resp)

#=================================================================
==================
def sw2_detect(pin):
        print 'Panic Button Detected'
        gpio.remove_event_detect(11)
        global flag2
        if (flag2==100):
                flag2=0
                subprocess.call("./save2.sh", shell=False)
                print('P button inactive')
                resp =  sendSMS('bishan@bm-es.com', 'satnamWAHEGURU123', number,
'TXTLCL', 'Panic Button Detected')
                print (resp)
#=================================================================
==================
def sw3_detect(pin):
        print 'Traffic Light Crossing Detected'
        global count
        if count>=6 and count<=12 :
                gpio.remove_event_detect(12)
                global flag3
                if (flag3==100):
```

```
                    flag3=0
                    subprocess.call("./save3.sh", shell=False)
                    print('T button inactive')
                    resp =  sendSMS('bishan@bm-es.com', 'satnamWAHEGURU123',
number, 'TXTLCL', 'Light Crossing Detected')
                    print (resp)
#=====================================================
==================
def sw4_detect(pin):
        print 'Image Saving Detected'
        gpio.remove_event_detect(7)
        global flag4
        if (flag4==100):
                flag4=0
                subprocess.call("./save4.sh", shell=False)
                print('I inactive')
#=====================================================
==================
def red_on():
        print 'Red On'
#=====================================================
==================
def green_on():
        print 'Green On'
#=====================================================
==================
def init_io():
        gpio.setmode(gpio.BOARD) # Set pin numbering to board numbering
        gpio.setwarnings(False)
        gpio.setup(7, gpio.IN) # Set up pin 7 as an input for 1 minute
        gpio.add_event_detect(7, gpio.RISING, callback=sw4_detect, bouncetime=200) #
```

```
Set up an interrupt to look for button presses
        gpio.setup(11, gpio.IN) # Set up pin 11 as an input for panic
        gpio.add_event_detect(11, gpio.RISING, callback=sw2_detect, bouncetime=200) #
Set up an interrupt to look for button presses
        gpio.setup(12, gpio.IN) # Set up pin 12 as an input for traffic
        gpio.add_event_detect(12, gpio.FALLING, callback=sw3_detect, bouncetime=200)
# Set up an interrupt to look for button presses
        gpio.setup(19, gpio.IN) # Set up pin 7 as an input
        gpio.add_event_detect(19, gpio.RISING, callback=sw1_detect, bouncetime=200) #
Set up an interrupt to look for button presses



        gpio.setup(13, gpio.IN) # Set up pin 13 as an input for
        #gpio.add_event_detect(13, gpio.RISING, callback=sw4_detect, bouncetime=200)
# Set up an interrupt to look for button presses
        gpio.setup(15, gpio.IN) # Set up pin 15 as an input
        #gpio.add_event_detect(15, gpio.RISING, callback=sw5_detect, bouncetime=200)
# Set up an interrupt to look for button presses



        gpio.setup(22, gpio.OUT) # Set up pin 22 as an output pin for 1 minute capture
        gpio.output(22, False)

        gpio.setup(16, gpio.OUT) # Set up pin 16 as an output red color
        gpio.output(16, False)
        gpio.setup(18, gpio.OUT) # Set up pin 18 as an output green color
        gpio.output(18, True)

        if (gpio.input(13) == True): # Physically read the pin now
                print('13 High')
        else:
```

```
            print('13 Low')

        if (gpio.input(15) == True): # Physically read the pin now
            print('15 High')
        else:
            print('15 Low')


#=================================================================
====================
def check_red_green() :
        global count
        count=count+1
        #Red 6-12
        if count==6 :
            gpio.output(16, True)
            gpio.output(18, False)
        #Green 1-6
        if count==12 :
            gpio.output(16, False)
            gpio.output(18, True)
            gpio.output(22, True)
            count=0
        #Check
        if count==3 :
            if (gpio.input(15) == False): # Physically read the pin now
                print('Green Not Okay')
                resp =  sendSMS('bishan@bm-es.com', 'satnamWAHEGURU123',
number, 'TXTLCL', 'Green Not Okay')
                print (resp)
            else :
```

```
                print('Green Okay')



        #Check
        if count==9 :
                if (gpio.input(13) == False): # Physically read the pin now
                        print('Red Not Okay')
                        resp =  sendSMS('bishan@bm-es.com', 'satnamWAHEGURU123',
number, 'TXTLCL', 'Red Not Okay')
                        print (resp)
                else :
                        print('Red Okay')

#=================================================================
====================
def check_sw1() :
        global flag1
        flag1=flag1+1
        if flag1>=6 and flag1<=15 :
                flag1=100
                print('W button active')
                gpio.add_event_detect(19, gpio.RISING, callback=sw2_detect,
bouncetime=200) # Set up an interrupt to look for button presses
        if flag1>=100 :
                flag1=100

#=================================================================
====================
def check_sw2() :
        global flag2
        flag2=flag2+1
```

```
        if flag2>=6 and flag2<=15 :
                flag2=100
                print('P button active')
                gpio.add_event_detect(11, gpio.RISING, callback=sw2_detect,
bouncetime=200) # Set up an interrupt to look for button presses
        if flag2>=100 :
                flag2=100
#=================================================================
==================
def check_sw3() :
        global flag3
        flag3=flag3+1
        if flag3>=6 and flag3<=15 :
                flag3=100
                print('T button active')
                gpio.add_event_detect(12, gpio.FALLING, callback=sw3_detect,
bouncetime=200) # Set up an interrupt to look for button presses
        if flag3>=100 :
                flag3=100


#=================================================================
==================
def check_sw4() :
        global flag4
        flag4=flag4+1
        if flag4>=6 and flag4<=15 :
                flag4=100
                print('I active')
                gpio.output(22, False)
                gpio.add_event_detect(7, gpio.RISING, callback=sw4_detect,
bouncetime=200) # Set up an interrupt to look for button presses
```

```
        if flag4>=100 :
                flag4=100

#=====================================================
==================
# main() function
def main():
        print("\n******** System Started ********")
        print ("Sending Initial SMS...")
        #resp =  sendSMS('bishan@bm-es.com', 'satnamWAHEGURU123', number,
'TXTLCL', message)
        #print (resp)
        time.sleep(2)
        init_io()
        global count
        while (True):
                time.sleep(5)
                check_red_green()
                check_sw1()
                check_sw2()
                check_sw3()
                check_sw4()

#=====================================================
==================
if __name__ == '__main__':
        main()
#=====================================================
==================
```

5.2 Unit Testing

After the completion of hardware and software coupling several tests were done on design. This section of the report explains our procedures for testing the final products. Using the following tests we were able to verify that our product was sound. Some observations and corrections were made as follows:

The system would not work when the raspberry pi device was switched on after switching on the rest of the equipment. This was due to the fact that the signals sent at first by the system to the phone do not die of the line immediately and affect subsequent signals sent even after connecting and resetting the system. This can be prevented by connecting the raspberry pi adopter to power supply first before powering the system.

It was observed that actions could not be performed by the IR sensor on the device (or LED in the equipment) when the LED was GREEN. This was due to on green light which made it not to process new data sent by phone during an existing cycle, because on the green light IR sensor need not to capture any traffic rule breaker image because its already green however web cam will keep taking picture every minute as it functioned that way. The solution was flushing the buffer after each command so that the next command could be executed well. When the RED LED is lit, then all the four corresponding sensors work perfectly fine.

The raspberry pi device will not work if it is not connected to wifi which is used to obtain the ip address of the raspberry pi device.

5.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one steps up – software applications at the company level – interact without error.

In this project, integration testing involves the testing of all the sub modules and modules together.

We have 5 basic modules(IR sensor module, Water Sensor module, Image saving module, LDR module, Live video surveillance) of testing which we test together with each other over 2 phases(Red light phase and green light phase). During the phase where green light

is on the IR sensor module is inactive and does not upload any images or sends any text messages rest all the modules work. During the Red light phase all the modules are working and if there is any detection the traffic controller is notified.

5.4 Test Results

- The green light and red light are working fine.
- Both the webcams are working properly.
- RaspberryPi is working without any errors.
- LDR is detecting if there is any problem in the traffic lights.
- IR sensor is detecting the traffic light crossing perfectly.
- Water sensor detects any water logging immediately.
- The first webcam is capturing photos during traffic light crossing and water logging.
- The second webcam is capturing video at all times.
- The webcam is also capturing images after every minute.

Chapter 6

Results and Discussion

6.1 Output/Results

Once we fire up the raspberry pi kernel it connects to lamp web camera instantly to give the live video feed of the street. And after execution of the program we start receiving text messages on the number we specified.

Also the traffic controller (end user) can access the photos clicked at several intervals through their drop box account.

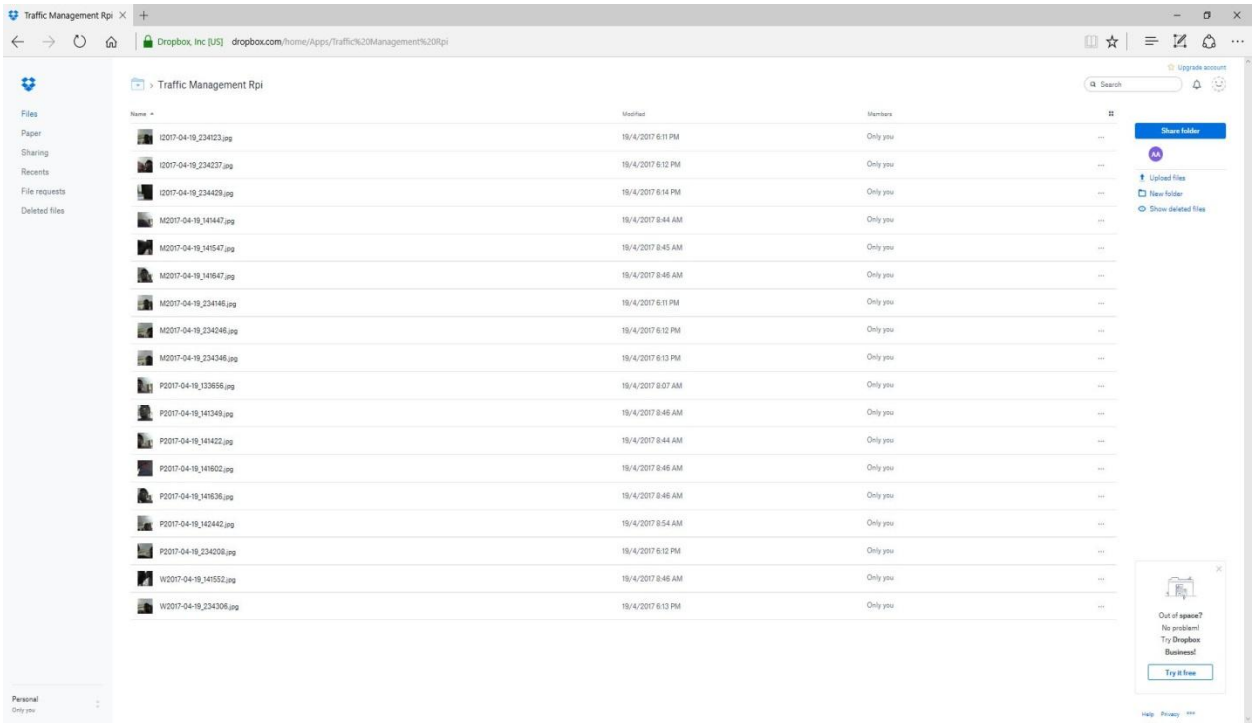


Fig 6.1.1- Output 1- Dropbox Account

Screenshots of the DropBox account(top) and Text Messages received by the controller(bottom)

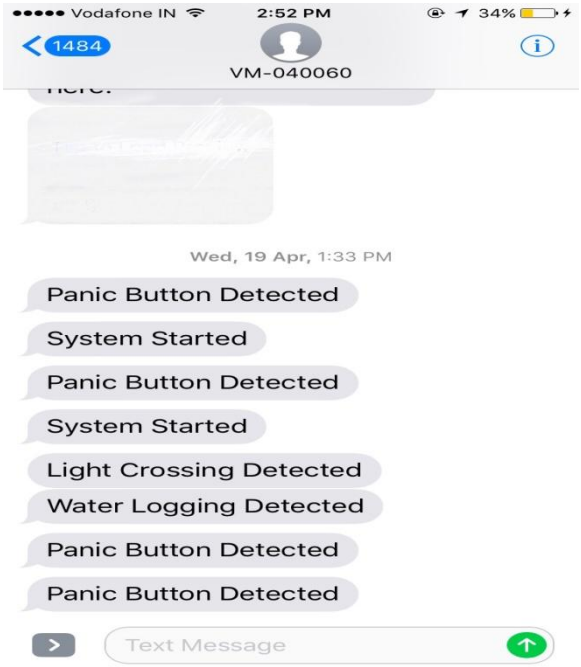


Fig 6.1.2- Output 2- Text Message

The images are categorized under 4 categories (M-photos clicked after each minute, W-photos clicked if water logging is detected, P-photos clicked if panic button is pressed, I-photos clicked if IR sensor detects a road crossing.) Whenever any of the 4 scenarios occur the controller is also notified via Text messages.

The photos are stored locally to the SD card on the raspberry pi.

6.2 Results Analysis/Discussion

According to the results obtained by testing different modules with many test cases, the working of model and output gained are optimized to maximum accuracy. The working of panic button sounds the alarm instantly. Detection of water level is working properly. Photos are stored according to the correct detection of module like which interrupt has occurred.

Chapter 7
Conclusion and Future Work

7.1 Conclusion

Human and computer interaction has been developed into a wide and sophisticated field. Earlier automating electrical devices were completely mechanical. But with the invention of computer system, many computer researchers have tried to create computer based intelligent systems to accomplish many of their functionalities. Raspberry pi controlled smart traffic management system is one of them. These fascinating efforts to create intelligent systems is to provide human being a more convenient & comfortable life. Moreover, it would accelerate the working speed of users.

Hence, Smart Traffic Management System will provide traffic management solutions that enable highway operators or government authorities to take actions that ultimately result in improving the safety of road users along with improving the traffic flow, increase transportation system efficiency, tackle water logging situation to avoid floods. Implementing such a system in metropolitan cities like Delhi, Mumbai, Chennai will help us in avoiding over-crowding at several places and with a full control system the traffic operator can operate the traffic lights as per the traffic congestion. The system will ensure remote traffic management and detect emergency situations.

In this project we have implemented a raspberry pi based traffic enhancement system which we Setup Raspberry Pi as a Web cam server to click the photos of the traffic junction where cameras can be installed and store these photos or upload it on Dropbox in case any of the sensors (Water Logging, etc) crosses a threshold value. We can also view these photos remotely, using any mobile device equipped with a browser and having internet access. So the traffic authorities will have remote access to any traffic junction over the internet in real time and this can save the requirement of traffic police at every junction. We have used LED's simulating traffic lights and as soon as the RED Light is on, the IR sensors become active and any one crossing the IR sensor or jumping the red light is captured on the camera. This can help in reducing red light jumps as people will be conscious of the camera being used to monitor traffic and a record being maintained

online. Generate a SMS alert as soon as a sensor is triggered. This will be generated by Raspberry Pi using internet services and can be sent to Police/Traffic/MCD, as required. The SMS would be sent using internet enabled SMS services. We'll be using the default distribution of Linux (Raspbian) on Raspberry Pi as our O.S. and will write our code and scripts in Python.

The proposed system is user friendly, low cost and easy to understand. The range of raspberry pi is global so the user can use the system from any corner of the world. It will automate the home appliances. For future work we can add many features in the system such as sensors for gas and temperature. After testing the system, we will implement it on the hardware. The project is aimed to design and implement a Raspberry pi bas traffic control system.

After doing different tests and programming different codes, eventually the obliged outcome is put forward. It is a fast and efficient approach to control the devices. This equipment works anywhere with a great gathering sign. With regards to the requirements gathered the manual work and the complexity in counting can be achieved with the help of electronic devices.

The project "SMART TRAFFIC MANAGEMENT SYSTEM USING RASPBERRY PI" has been successfully designed and tested. Integrating features of all the hardware components used have developed it. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced IC's and with the help of growing technology the project has been successfully implemented.

7.2 Scope of Future Work

Continuous growth of population all over the world creates a great challenge to the transport management systems. The conventional methods are no longer effective enough for solving complex and challenging transportation management problems. More economical, more efficient and thus more intelligent methods have to be developed to deal with these challenging problems. Knowledge from different research areas is needed for developing these systems. Very often complex transportation systems require integration of different methods from different branches of science. Due to the increased amount of vehicles, it is necessary to take effective steps in order to control the traffic and hence avoid all types of loses that is caused due to traffic. Once we have predicted a high

traffic density for a network segment, we can initiate strategies to avoid this problem. In case of a road network, navigation systems can try to bypass the critical zone. Furthermore, any traffic control systems can inform the drivers about the traffic jam risk in order to guide them around the critical zone. In order to detect the traffic different sensors are being used and different techniques are used to determine the traffic and thus solve the problem related to traffic. Computer vision presents significant advantage over other traditional vehicle measurement technologies. Computer vision systems are more flexible, less invasive, and more precise, more robust, easier to maintain, produce richer information, do not affect the integrity of the road and offer as an a added bonus, the possibility to transmit images for human supervision.

Use of Artificial Intelligence can be increased with the sensors and predicts the problems which can arise. A project on traffic management solution detecting real-time traffic density estimation and prediction can be made.

Chapter 8
References

References

Weblinks:

- About Raspberry Pi, http://www.raspberrypi.org/about/
- Raspberry Pi Hardware information, http://elinux.org/RPi_Hardware
- How SoC works, http://www.androidauthority.com/how-it-works-systems-on-a-chip-soc-93587/

Journal:

- Monahan, Torin. 2007. "War Rooms" of the Street: Surveillance Practices in Transportation Control Centres. The Communication Review 10 (4):367-389.
- Smart Traffic Management System-International Journal of Computer Applications (0975 – 8887) Volume 75– No.7, August 2013
- S.A. Mulay, C.S. Dhekne, R.M. Bapat, T.U. Budukh, S.D. Gadgil Department of Computer, PVG's COET, Intelligent traffic management and public transportation system.