SMART TRAFFIC MANAGEMENT SYSTEM

A FINAL PROJECT REPORT(Review 2)

Submitted in partial fulfillment for the award of the degree of

B.TECH

in

Information Technology

by

Ashish Arora 13BIT0024

BTech IT WINTER SEMESTER 2017-18

Review 2

Under the Guidance of

Prof. Usha Devi G.

School of Information Technology and Engineering

February 2017

TABLE OF CONTENTS

Chapter No	Contents	Page No
1.	**ABSTRACT**	3
2.	**INTRODUCTION**	4
3.	**LITERATURE SURVEY**	5
4.	**DRAWBACKS OF EXISTING SYSTEM**	6
5.	**PROPOSED SYSTEM**	6
6.	**DETAILED DESIGN**	7-12
6.1	Block Diagram	7
6.2	Raspberry Pi	8-10
6.3	Voltage Regulator	11
6.4	Resistors	12
6.5	Power Supply	12
7.	**MODULE DESCRIPTION**	13-15
7.1	LED	13
7.2	Infrared Sensor	14
7.3	Water Sensor	14
7.4	Voltage Regulators	15
8.	**Diagrams**	16-17
8.1	Activity Diagram	16
8.2	Sequence Diagram	17
9.	**Code Implementation**	18-24

10.	References	25

ABSTRACT

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 perform some activity accordingly (SMS in this case).

INTRODUCTION

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.

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).

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

PROPOSED SYSTEM

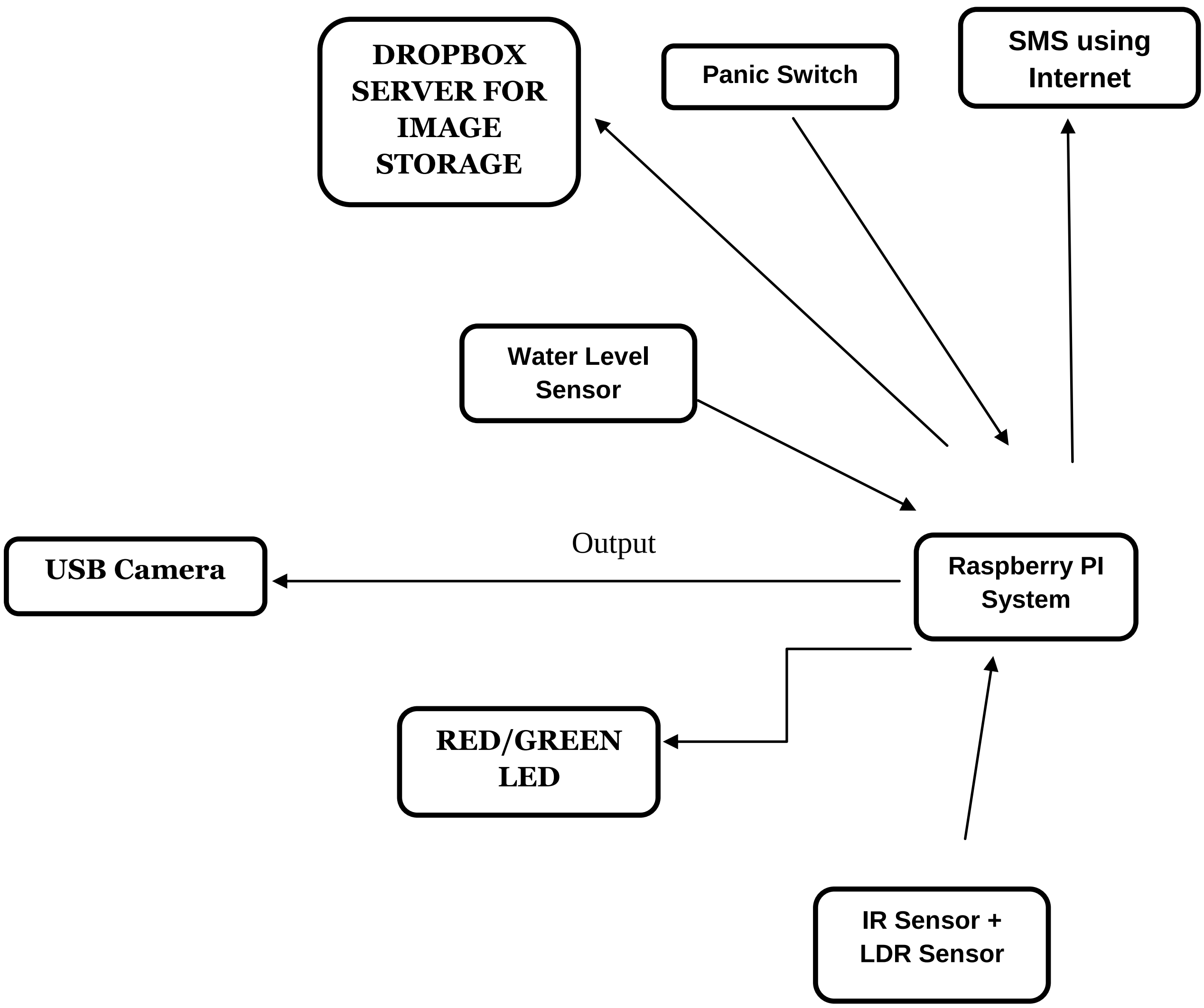
New technology has IR+LDR sensor, Water level sensor, USB web Camera and sending of a message through IOT.

Advantages

- ✓ Panic button in case of emergency
- ✓ Reduces red light jumps
- ✓ Monitor real time traffic
- ✓ Low cost
- ✓ Low power consumption

DETAILED DESIGN

BLOCK DIAGRAM



Raspberry Pi

1) Processor / SoC (System on Chip)

The Raspberry Pi has a Broadcom BCM2835 System on Chip module. It has a ARM1176JZF-S processor

The Broadcom SoC used in the Raspberry Pi is equivalent to a chip used in an old smartphone (Android or iPhone). While operating at 700 MHz by default, the Raspberry Pi provides a real world performance roughly equivalent to the 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997-1999, but the GPU, however, provides 1 Gpixel/s, 1.5 Gtexel/s or 24 GFLOPS of general purpose compute and the graphics capabilities of the Raspberry Pi are roughly equivalent to the level of performance of the Xbox of 2001. The Raspberry Pi chip operating at 700 MHz by default, will not become hot enough to need a heatsink or special cooling.

2) Power source

The Pi is a device which consumes 700mA or 3W or power. It is powered by a MicroUSB charger or the GPIO header. Any good smartphone charger will do the work of powering the Pi.

3) SD Card

The Raspberry Pi does not have any onboard storage available. The operating system is loaded on a SD card which is inserted on the SD card slot on the Raspberry Pi. The operating system can be loaded on the card using a card reader on any computer.

4) GPIO

GPIO – General Purpose Input Output

General-purpose input/output (GPIO) is a generic pin on an integrated circuit whose behaviour, including whether it is an input or output pin, can be controlled by the user at run time.GPIO pins have no special purpose defined, and go unused by default. The idea is that sometimes the system designer building a full system that uses the chip might find it useful to have a handful of additional digital control lines, and having these available from the chip can save the hassle of having to arrange additional circuitry to provide them.

GPIO capabilities may include:

- ✓ GPIO pins can be configured to be input or output
- ✓ GPIO pins can be enabled/disabled
- ✓ Input values are readable (typically high=1, low=0)
- ✓ Output values are writable/readable
- ✓ Input values can often be used as IRQs (typically for wakeup events)

The production Raspberry Pi board has a 26-pin 2.54 mm (100 mil) expansion header, marked as P1, arranged in a 2x13 strip. They provide 8 GPIO pins plus access to (I²C,SPI, UART), as well as +3.3 V, +5 V and GND supply lines. Pin one is the pin in the first column and on the bottom row.

Fig4.2 GPIO Pin

5) DSI Connector

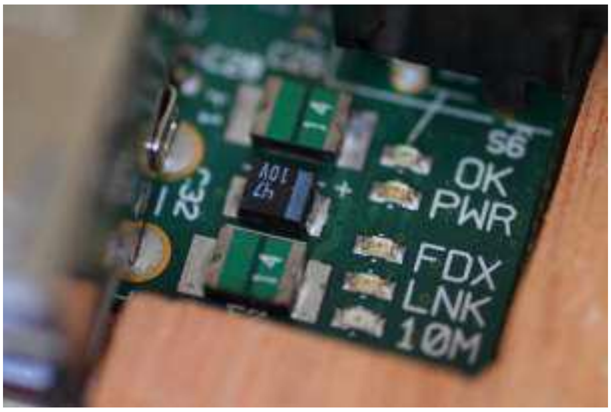
The Display Serial Interface (DSI) is a specification by the Mobile Industry Processor

Interface (MIPI) Alliance aimed at reducing the cost of display controllers in a mobile device. It is commonly targeted at LCD and similar display technologies. It defines a serial bus and a communication protocol between the host (source of the image data)
and the device (destination of the image data).
A DSI compatible LCD screen can be connected through the DSI connector, although it may require additional drivers to drive the display.

6) RCA Video
RCA Video outputs (PAL and NTSC) are available on all models of Raspberry Pi.Any television or screen with a RCA jack can be connected with the RPi.

7) Audio Jack
A standard 3.5 mm TRS connector is available on the RPi for stereo audio output.
Any headphone or 3.5mm audio cable can be connected directly. Although this jack cannot be used for taking audio input, USB mics or USB sound cards can be used.

8) Status LEDs
There are 5 status LEDs on the RPi that show the status of various activities as follows:
―OK‖ - SDCard Access (via GPIO16) - labelled as "OK" on Model B Rev1.0 boards and "ACT" on Model B Rev2.0 and Model A boards
―POWER‖ - 3.3 V Power - labelled as "PWR" on all boards
―FDX‖ - Full Duplex (LAN) **(Model B)** - labelled as "FDX" on all boards
―LNK‖ - Link/Activity (LAN) **(Model B)** - labelled as "LNK" on all boards
―10M/100‖ - 10/100Mbit (LAN) **(Model B)** - labelled (incorrectly) as "10M" on



Model B Rev1.0 boards and "100" on Model B Rev2.0 and Model A

9) USB 2.0 Port

USB 2.0 ports are the means to connect accessories such as mouse or keyboard to the Raspberry Pi. There is 1 port on Model A, 2 on Model B and 4 on Model B+. The number of ports can be increased by using an external powered USB hub which is available as a standard Pi accessory.

10) Ethernet

Ethernet port is available on Model B and B+. It can be connected to a network or internet using a standard LAN cable on the Ethernet port. The Ethernet ports a controlled by Microchip LAN9512 LAN controller chip.

11) CSI connector

CSI – Camera Serial Interface is a serial interface designed by MIPI (Mobile Industry Processor Interface) alliance aimed at interfacing digital cameras with a mobile processor.
The RPi foundation provides a camera specially made for the Pi which can be connected with the Pi using the CSI connector.

12) JTAG headers

JTAG is an acronym for _Joint Test Action Group', an organisation that started back in the mid 1980's to address test point access issues on PCB with surface mount devices. The organisation devised a method of access to device pins via a serial port that became known as the TAP (Test Access Port). In 1990 the method became a recognized international standard

(IEEE Std 1149.1). Many thousands of devices now include this standardized port as a feature to allow test and design engineers to access pins.

13) HDMI

HDMI – High Definition Multimedia Interface HDMI 1.3 a type A port is provided on the RPi to connect with HDMI screens.

Voltage Regulator

Then at last voltage controllers are associated over the circuit to secure the supplies from any kind of voltage variances. All the gadgets we utilize are delicate and sensitive, so protection from any kind of variances is very important. A voltage controller is intended to naturally keep up a consistent voltage level. We are using the 7805 Voltage Regulator in this project. IC 7805 is a DC regulated IC of 5V. This IC is very flexible and is widely employed in all types of circuit like a voltage regulator. It is a three terminal device and mainly called input, output and ground.

Fig 4.8 Voltage Regulator

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. Electronic voltage regulators are found in devices such as computer power supplies where they stabilize the DC voltages used by the processor and other elements. In automobile alternators and central power station generator plants, voltage regulators control the output of the plant. In an electric power distribution system, voltage regulators may be installed at a substation or along distribution lines so that all customers receive steady voltage independent of how much power is drawn from the line.

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). Photo 1.1a shows 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.

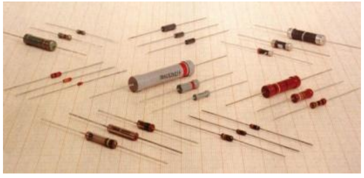


Fig 1.1a

Fig 1.1b

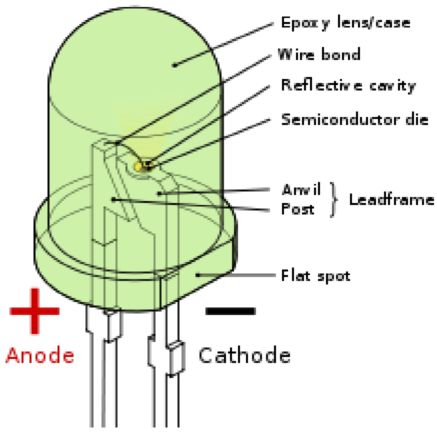
Power Supply

Power Supply is the device that transfers electric power from a source to a load using electronic circuits. Power supplies are used in many industrial and aerospace applications and also in consumer products. Some of the requirements of power supplies are small size, lightweight, low cost, and high power conversion efficiency. In addition to these, some power supplies require the following: electrical isolation between the source and load, low harmonic distortion for the input and output waveforms, and high power factor (PF) if the source is ac voltage. Some special power supplies require controlled direction of power flow. Typical application of power supplies is to convert utility's AC input power to a regulated voltage(s) required for the purpose of the regulation of the device.

MODULE DESCRIPTION

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.

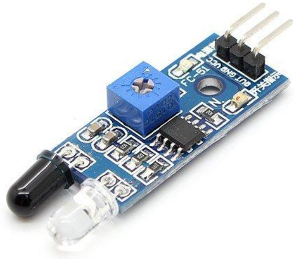


When a light-emitting diode is switched on, electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the colour of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. An LED is often small in area (less than 1 mm^2), and integrated optical components may be used to shape its radiation pattern. LEDs have many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. However, LEDs powerful enough for room lighting are relatively expensive, and require more precise current and heat management than compact fluorescent lamp sources of comparable output. Light-emitting diodes are used in applications as diverse as aviation lighting, automotive lighting, advertising, general lighting, and traffic signals. LEDs have allowed new text, video displays, and sensors to be developed, while their high switching rates are also useful in advanced communications technology. Infrared LEDs are also used in the remote control units of many commercial products including televisions, DVD players and other domestic appliances. LEDs are also used in seven segment display.

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) (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 v



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.

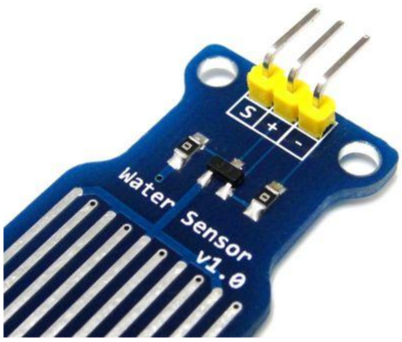
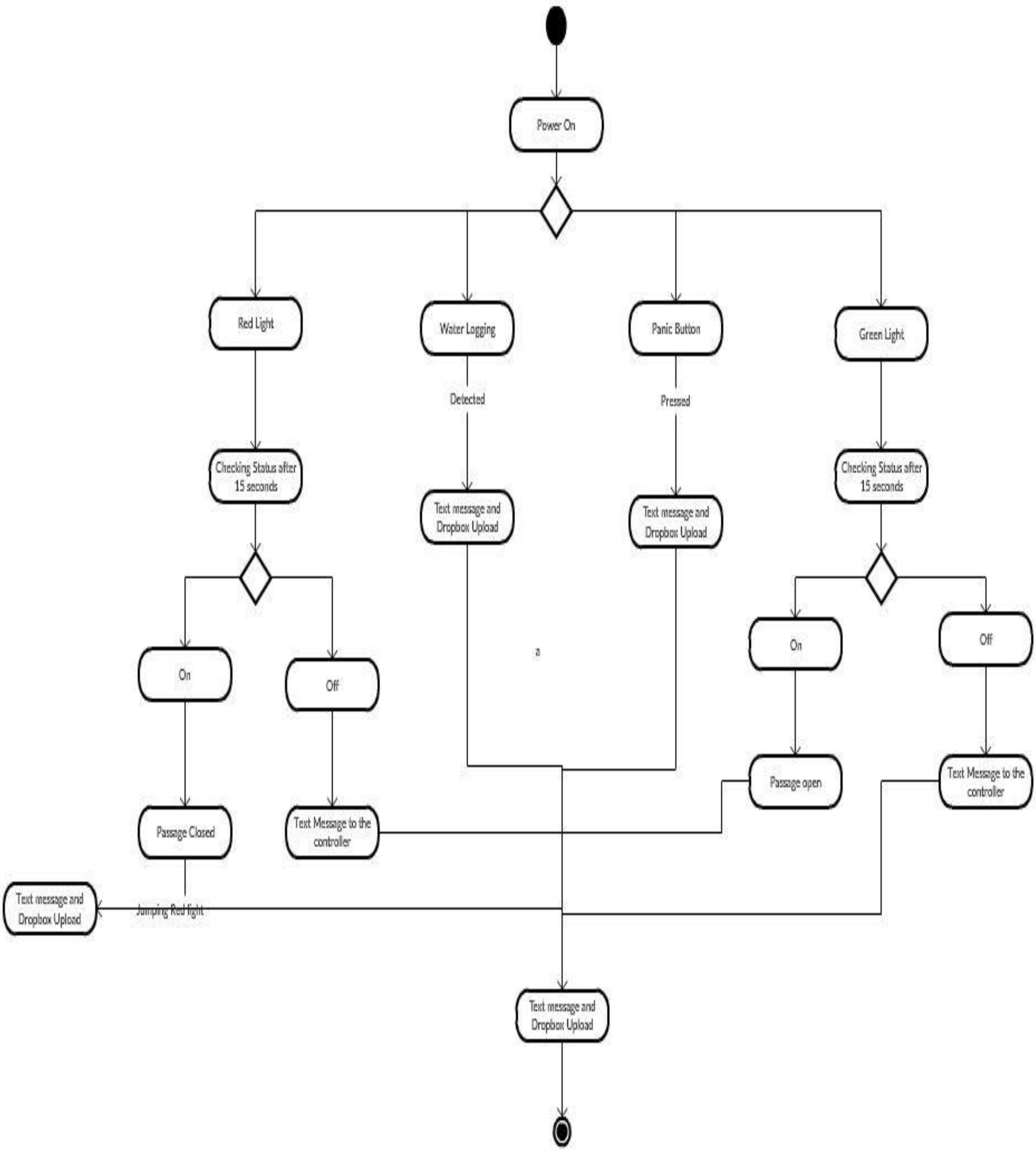


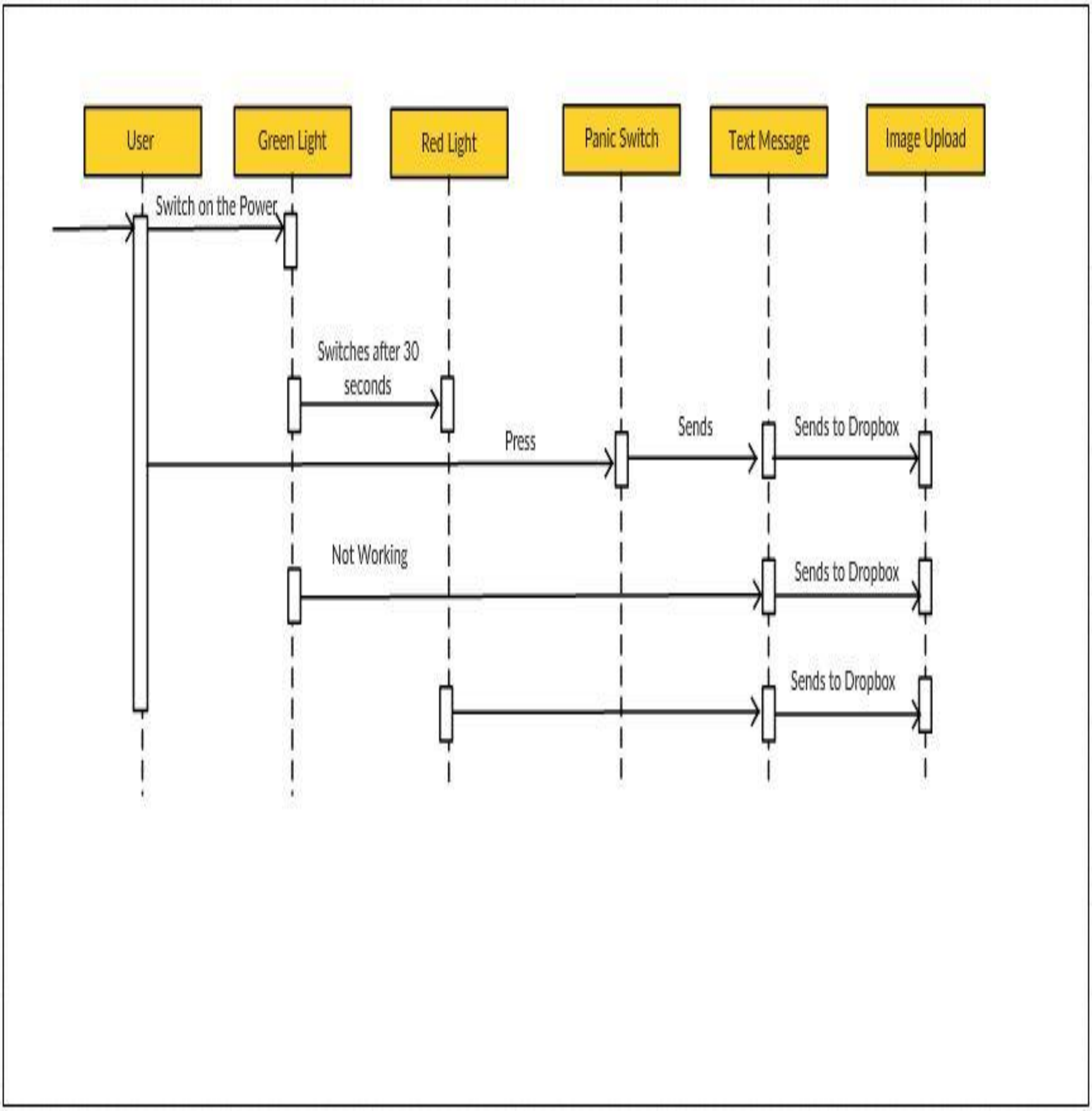
Fig3.4 Water Sensor

Diagrams

Activity Diagram



Sequence Diagram



Code- Implementation

Main.py

```
import os

import time

import urllib

import urllib2

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@gmail.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 :#red light 6 to 12 seconds
		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@gmail.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')
```

Save1.sh

```
#!/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

```
#!/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

```
#!/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

```
#!/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
```

References

- Smart Traffic Management System-International Journal of Computer Applications (0975 – 8887) Volume 75– No.7, August 2013
- Monahan, Torin. 2007. “War Rooms” of the Street: Surveillance Practices in Transportation Control Centres. The Communication Review 10 (4):367-389.
- . 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.