# CHAPTER 1
# INTRODUCTION

This chapter discusses the introduction of traffic congestion, IOT. Traffic management plays an important in the development of a country. Due to the increasing of population traffic controlling is becoming one of the challenging issue for the traffic police in urban as well metro cities. The major problem is that at the road junctions the signals of traffic lights set as a predefined time and it does not depend on varying of traffic density over time. So this creates waiting at road junctions, fuel wastage of vehicles, air pollution.

## 1.1 Introduction of Traffic Congestion

In recent years, there is lot of change in usage of vehicles which is becoming a challenge for transportation system. When predefined time is allocated at every road crossings which results in traffic congestion and becomes uncontrollable in peak traffic hours, which results the waiting at traffic junctions.

To resolve the traffic congestion by construction of roads, improving the road infrastructure by construction of flyovers, bridges. However construction of roads in urban areas is impossible because of limited land resources. So, in recent years traffic congestion problem has solution by using of enabling technologies like Internet of Things (IOT).

## 1.2 Internet of Things(IOT)

The term “The Internet of Things” (IOT) was coined by Kelvin Ashton in a presentation to proctor and gamble in 1999.

The Internet of Things (IOT) is the extension of internet connectivity into physical devices and everyday objects. Embedded with electronics, internet connectivity, and hardware (such as sensors), these devices can communicate and interact with others over the internet, and they can be remotely monitored and controlled. An IOT system is comprised of four main components.

### 1.2.1 Sensors

A sensor enables the devices to collect data from the environment surrounding the device like temperature, humidity, etc.

### 1.2.2 Connectivity

The data connected is sent to the cloud through Wi-Fi or Bluetooth.

### 1.2.3 Data processing

Once the data is received by the cloud infrastructure, it can be processed check if the data received adhere to the requirements or not

### 1.2.4 User Interface

Once the data is processed, the results are given to user

## 1.3 Objectives

The main objective is to develop a traffic management system to regulate the traffic lights according to the congestion level at road junctions and display the traffic situation results on the cloud.

## 1.4 Organization of the Thesis

The thesis is divided into five chapters as outlined below

In Chapter 1, the Introduction of traffic congestion, IOT are explained.

A comprehensive review of the literature survey on Traffic Congestion Controlling and Management using IOT are discussed in Chapter 2.

Implementation of traffic congestion controlling and management is explained in Chapter 3.

Simulation results are given in Chapter 4.

Finally, in chapter 5 Conclusion are given.

# CHAPTER 2
# LITERATURE SURVEY

The previous chapter described the introduction of traffic congestion, IOT. This chapter describes the literature survey of traffic congestion controlling and management.

P. Sadhukan, F. Gazi implemented the traffic congestion control system. This system controls the congestion level of the traffic [1]. Costea, I. M. Nemtanu, F. C. Dumitrescu, C. Banu, C.V. Banu, implemented the monitoring system with applications in road transport. This system monitors the road traffic which further uses for analysis the traffic congestion level [2]. S. Javaid, A. Sufian, S. Pervaiz and M. Tanveer, implemented the smart traffic management system using Internet of Things [3].

A. Saikar, M. Parulekar, A. Badve, S. Thakkar and A. Deshmukh, implemented the smart traffic management for smart cities. The main objective of IOT is the introduction of smart surroundings [4]. D.Serpanos, M.Wolf, implemented the Internet-of-Things (IOT) systems [5].

## 2.1 Problem Statement

In this work developing the traffic congestion level based management system using IOT.

In this chapter the work carried out by various researchers on traffic congestion controlling is surveyed. In next chapter implementation of traffic congestion controlling and management using IOT is presented.

# CHAPTER 3

# IMPLEMENTATION

The previous chapter describes the literature survey of traffic congestion controlling system. This chapter describes the implementation of traffic congestion controlling and management system using IOT.

## 3.1 Implementation of traffic congestion controlling and management system using IOT

The block diagram of the traffic congestion controlling and management system using IOT is shown in figure 3.1.The components used in the traffic congestion controlling and management system using IOT are Raspberry pi 3B+, ultrasonic sensor, LCD display.

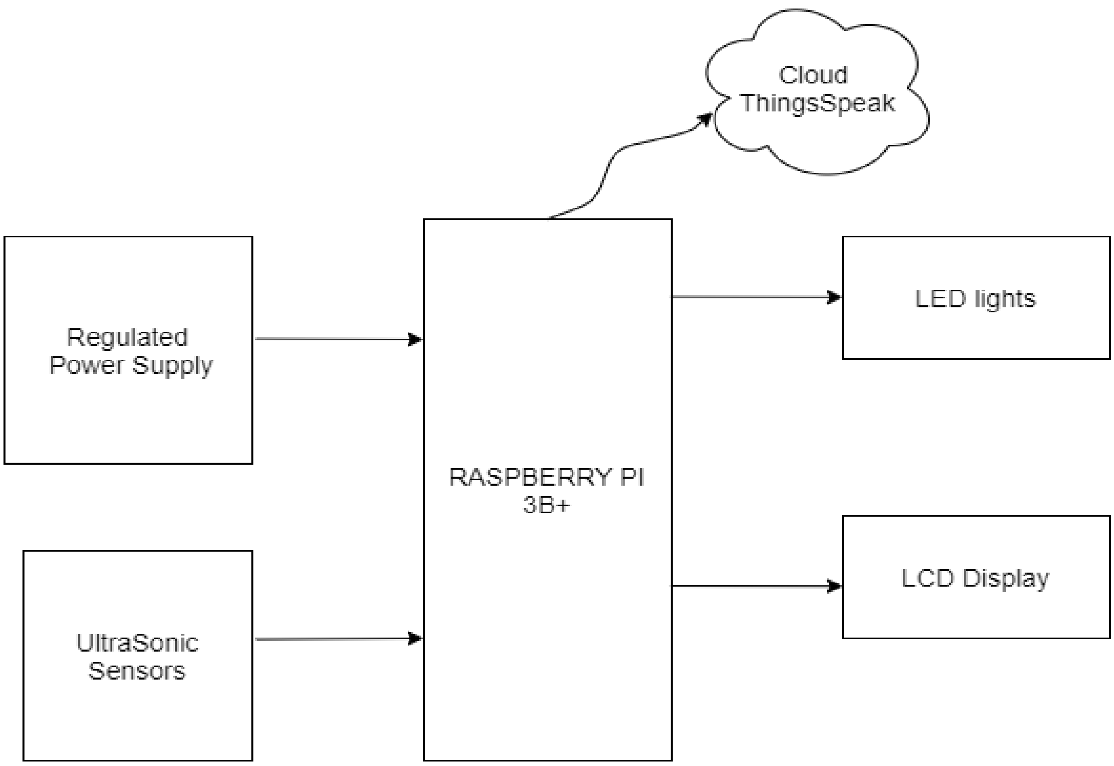



**Figure 3.1: Block Diagram of traffic congestion controlling and management system using IOT.**

### 3.1.1 Raspberry Pi 3B+

#### 3.1.1.1 Description

Raspberry pi is a series of small single-board computers (SBC is a complete computer built on a single circuit board, with microprocessors, memory, input/output and other features required of a functional computer) developed by the Raspberry Pi foundation. Raspberry pi 3 has on-board Wi-Fi, Bluetooth and USB boot capabilities.

All models feature a broad com system on chip (SOC) with an integrated ARM compatible CPU and on-chip graphics processing unit (GPU). The foundation provides Raspbian (OS for Raspberry pi) based Linux distribution for download, as well as third-party windows10 IOT core, RISC OS, and specialized media centre distributions. It promotes Python as the main programming language.

### 3.1.1.2 Raspberry Pi 3B+ Board

Raspberry Pi 3B+ model are shown in figure 3.2.

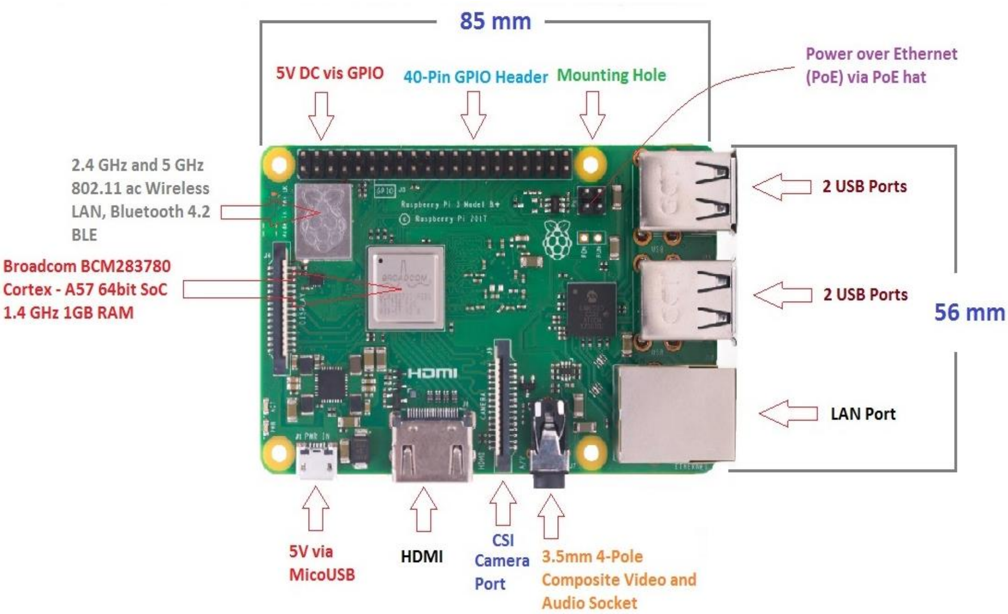



**Figure 3.2: Raspberry Pi 3B+ model**

### 3.1.1.3 Raspberry Pi 3B+ Structure Description

#### 3.1.1.3.1 Processor / Soc (System on Chip)

The Raspberry Pi has a Broadcom BCM2835 System on Chip module. It has an ARM1176JZF-S processor. The Broadcom SOC used in the Raspberry Pi is equivalent to a chip used in an old smart phone (Android or iPhone).The Raspberry Pi chip operating at 700 MHz by default, will not become hot enough to need a heat sink or special cooling.

#### 3.1.1.3.2 Power Source

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

**3.1.1.3.3 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.

**3.1.1.3.4 GPIO (General Purpose Input Output)**

Pin description of Raspberry Pi as shown in figure 3.3. General purpose input/output (GPIO) is a generic pin on an integrated circuit whose behavior, 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 used by default. 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.

The Raspberry Pi board has a 26-pin 2.54 mm (100 mil) expansion header, 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 Ground supply lines.

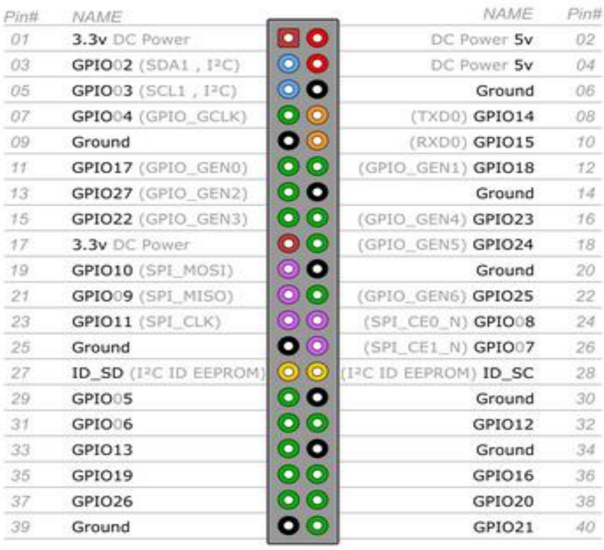



**Figure 3.3: Pin description of Raspberry Pi 3B+**

**3.1.1.3.5 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).

**3.1.1.3.6 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 Raspberry Pi.

**3.1.1.3.7 Audio Jack**

A standard 3.5 mm TRS connector is available on the Raspberry Pi for stereo audio output. Although this jack cannot be used for taking audio input, USB mics or USB sound cards can be used.

**3.1.1.3.8 LEDs on Raspberry pi**

There are 5 LEDs on the Raspberry Pi that show the status of various activities as shown in figure 3.4

- OK - SD Card Access (via GPIO16) - labeled as OK on Model B Rev1.0 boards.
- POWER- 3.3 V Power - labeled as PWR on all boards.
- FDX - Full Duplex (LAN) (Model B) - labeled as FDX on all boards.
- LNK - Link/Activity (LAN) (Model B) - labeled as LNK on all boards.
- 10M - 10/100Mbit (LAN) (Model B) - labeled (incorrectly) as 10M on model B.

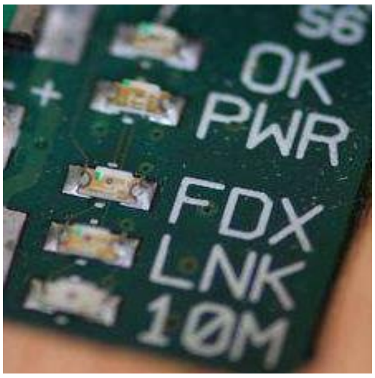



**Figure 3.4: LEDs on Raspberry pi 3B+**

**3.1.1.3.9 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.

**3.1.1.3.10 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 are controlled by Microchip LAN9512 LAN controller chip.

#### 3.1.1.3.11 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.

#### 3.1.1.3.12 HDMI

HDMI – High Definition Multimedia Interface.

HDMI 1.3 a type a port is provided on the Raspberry Pi to connect with HDMI screens.

### 3.1.1.4 Specifications

| | Raspberry Pi 3 Model B | Raspberry Pi Zero | Raspberry Pi 2 Model B | Raspberry Pi Model B+ |
|---|---|---|---|---|
| Introduction Date | 2/29/2016 | 11/25/2015 | 2/2/2015 | 7/14/2014 |
| SoC | BCM2837 | BCM2835 | BCM2836 | BCM2835 |
| CPU | Quad Cortex A53 @ 1.2GHz | ARM11 @ 1GHz | Quad Cortex A7 @ 900MHz | ARM11 @ 700MHz |
| Instruction set | ARMv8-A | ARMv6 | ARMv7-A | ARMv6 |
| GPU | 400MHz VideoCore IV | 250MHz VideoCore IV | 250MHz VideoCore IV | 250MHz VideoCore IV |
| RAM | 1GB SDRAM | 512 MB SDRAM | 1GB SDRAM | 512MB SDRAM |
| Storage | micro-SD | micro-SD | micro-SD | micro-SD |
| Ethernet | 10/100 | none | 10/100 | 10/100 |
| Wireless | 802.11n / Bluetooth 4.0 | none | none | none |
| Video Output | HDMI / Composite | HDMI / Composite | HDMI / Composite | HDMI / Composite |
| Audio Output | HDMI / Headphone | HDMI | HDMI / Headphone | HDMI / Headphone |
| GPIO | 40 | 40 | 40 | 40 |
| Price | $35 | $5 | $35 | $35 |

**Table 3.1 Different Models Specifications of Raspberry pi**

## 3.1.2 HC-SR04 Ultrasonic Sensor

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver as shown in figure 3.5

**Figure 3.5: Ultrasonic Sensor**

**3.1.2.1 Ultrasonic Sensor HC-SR04 Pin Diagram**

Pin diagram of ultrasonic sensor as shown in figure 3.6.

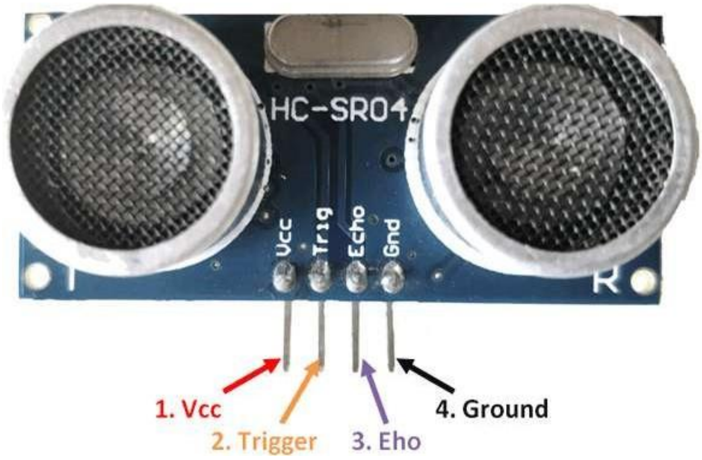



**Figure 3.6: Pin Diagram of Ultrasonic Sensor**

**3.1.2.2 Ultrasonic Sensor Pin Configuration:**

Ultrasonic sensor pin configuration is shown in table 3.2.

**Table 3.2: Ultrasonic Sensor Pin Configuration**

| Pin Number | Pin Name | Description |
|---|---|---|
| 1 | Vcc | The Vcc pin powers the sensor, typically with +5V. |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the Ultrasonic wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system. |

**3.1.2.3 HC-SR04 Sensor Features**

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 80cm
- Practical Measuring Distance: 2cm to 10cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

### 3.1.2.4 Ultrasonic Sensor – Working

HC-SR04 Ultrasonic sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple formula that

$$\text{Distance} = \text{Speed} \times \text{Time}$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in figure 3.7.

When ultrasonic sensors is connected to the Raspberry pi, ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets reflected by an object it gets reflected back toward the sensor this reflected wave is observed by the ultrasonic receiver module. Based on that information i.e, low, medium, high congestion the LED lights blink and each direction congestion level is display on LCD display.

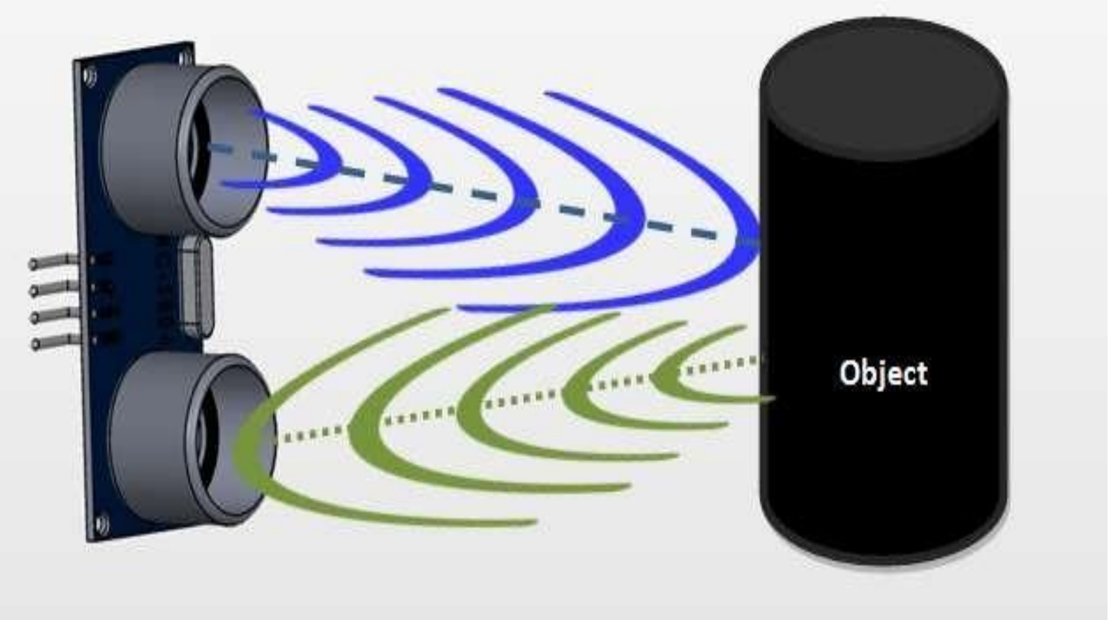



**Figure 3.7: Ultrasonic Sensor**

Now, to calculate the distance using the above formulae, we should know the Speed and time. The circuitry inbuilt on the module will calculate the time taken for the wave to come back and turns on the echo pin high for that same particular amount of time, by this way can also know the time taken.

## 3.1.3 LCD

Below figure 3.8 shows the LCD display. A Liquid Crystal Display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. The Liquid Crystal Display (LCD) is a low power device (microwatts). Now a days in most applications LCDs are using rather using of LED displays because of its specifications like low power consumption, ability to display numbers and special characters which are difficult to display with other displaying circuits and easy to program.

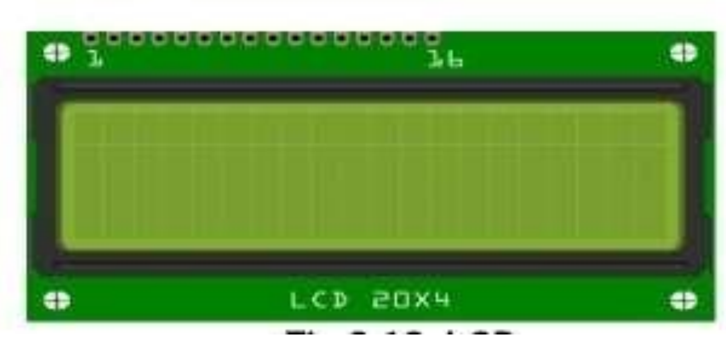



**Figure 3.8: LCD Display**

### 3.1.3.1 Pin Description of LCD display

Pin Description of LCD display as shown in figure 3.9.

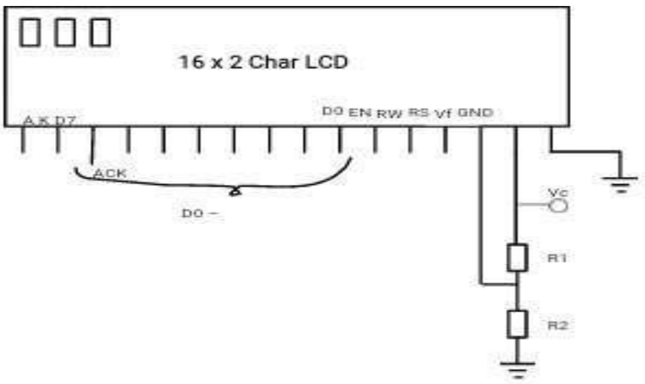



**Figure 3.9: LCD Pin Diagram**

### 3.1.3.2 RS (Command / Data)

This bit is to specify whether received byte is command or data. So that LCD can recognize the operation to be performed based on the bit status.

RS = 0 => Command

RS = 1 => Data

### 3.1.3.3 RW (Read / Write)

RW bit is to specify whether controller wants READ from LCD or WRITE to LCD. The READ operation here is just ACK bit to know whether LCD is free or not.

RW=0 => Write

RW=1 => Read

**3.1.3.4 EN (Enable LCD)**

EN bit is to ENABLE or DISABLE the LCD. Whenever controller wants to write something into LCD or READ acknowledgment from LCD it needs to enable the LCD.

EN =0 => High Impedance

EN =1 => Low Impedance

**3.1.3.5 ACK (LCD Ready)**

ACK bit is to acknowledge the Microcontroller unit that LCD is free so that it can send new command or data to be stored in its internal RAM locations

ACK =1 => Not ACK

ACK =0 => ACK

# CHAPTER 4

# RESULTS

The previous chapter describes the implementation of traffic congestion controlling and management system using IOT. This chapter describes simulation results, output view of the project-traffic congestion controlling and management using IOT.

## 4.1 Experimental Setup

The experimental setup of traffic congestion controlling and management system using IOT is shown in figure 4.1. The ultrasonic sensors are placed in four directions. Sensors measures the level of the congestion i.e., low, medium, high congestion level. Each direction congestion level is displayed on the LCD display.

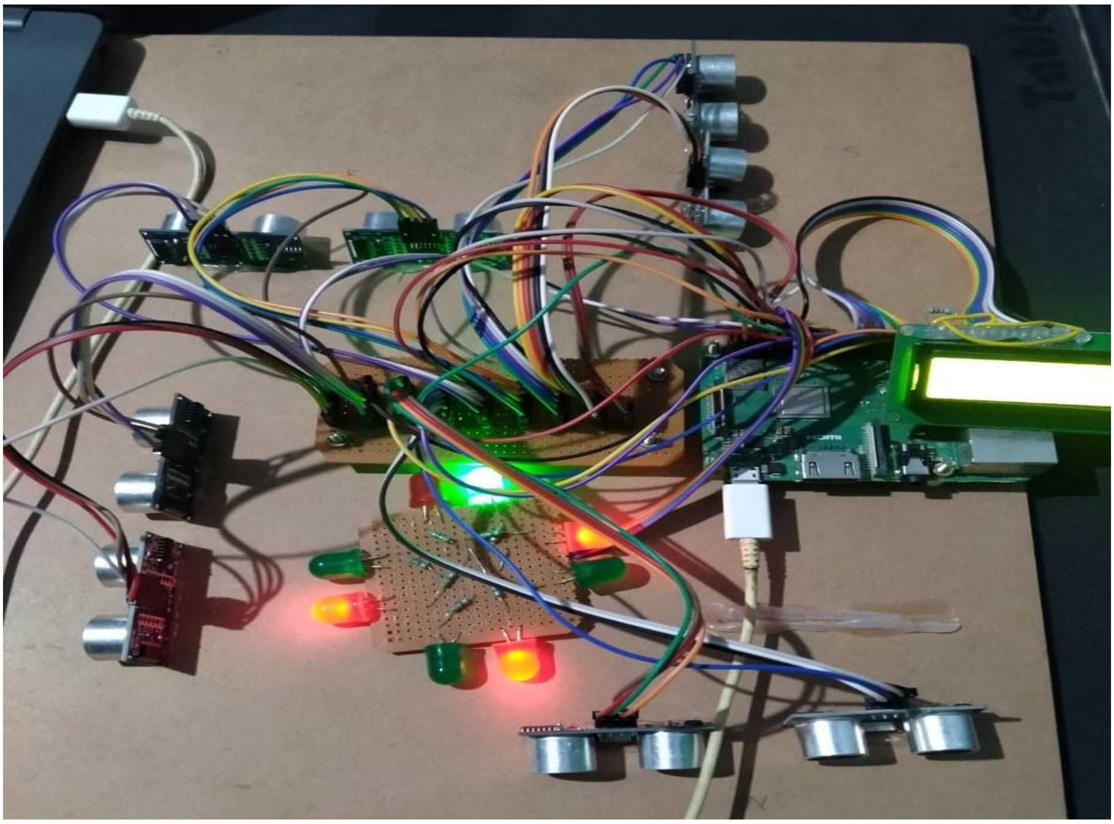


**Figure 4.1: The Traffic Congestion Controlling and Management System Using IOT**

The high traffic congestion displayed on LCD display is shown in figure 4.2. In this the output is display on LCD .Whenever high congestion occurs it is indicated as ‘H’ on LCD and for the clearance of the traffic at that road set timer from 20 seconds to 0 seconds.

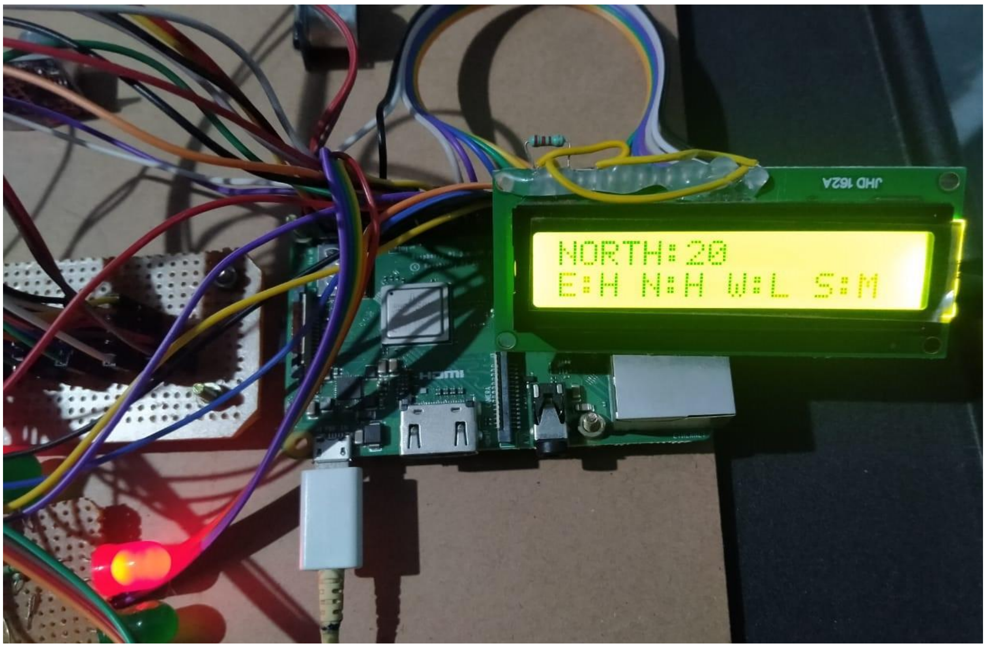



**Figure 4.2: High Traffic Congestion Displayed On LCD Display**

Whenever medium congestion occurs it is indicated as ‘M’ on LCD and for the clearance of the traffic at that road set timer from 15 seconds to 0 seconds as shown in figure 4.3

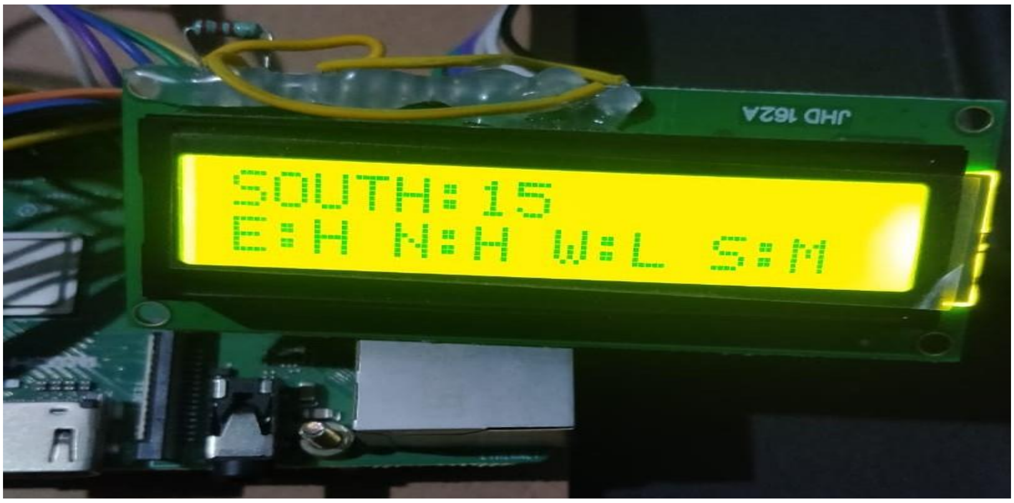



**Figure 4.3: Medium Congestion Displayed on LCD display**

Whenever low congestion occurs it is indicated as ‘L’ on LCD and for the clearance of the traffic at that road set timer from 10 seconds to 0 seconds as shown in figure 4.4.

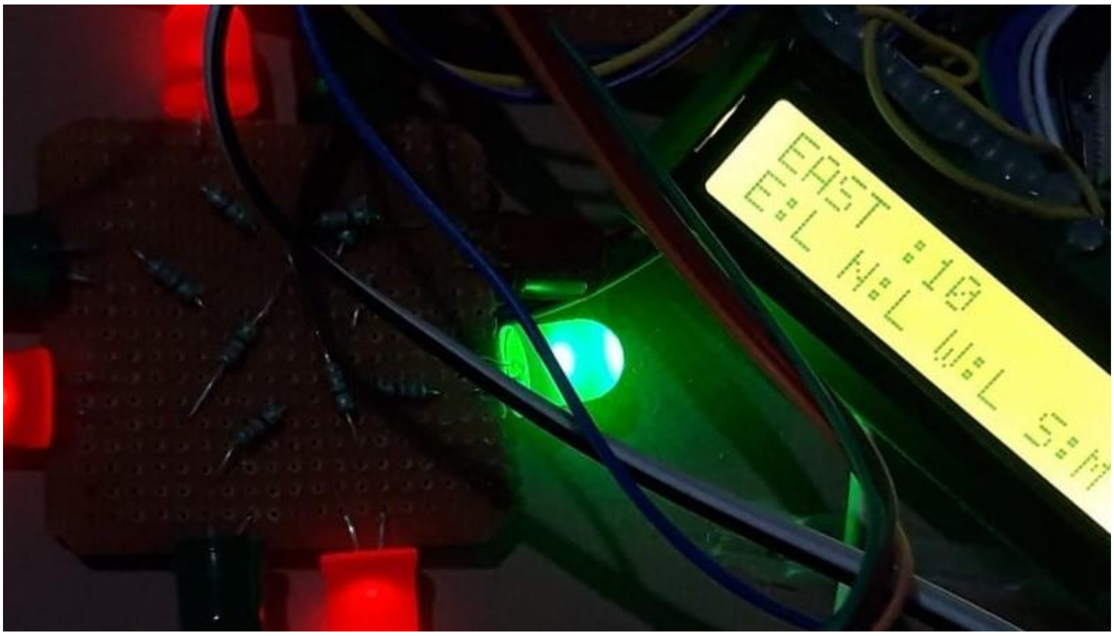
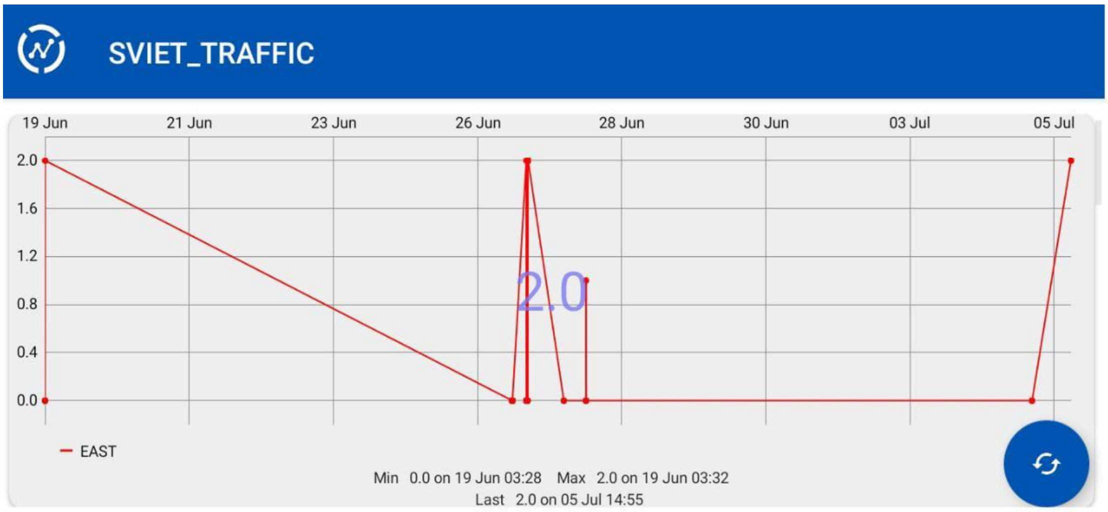
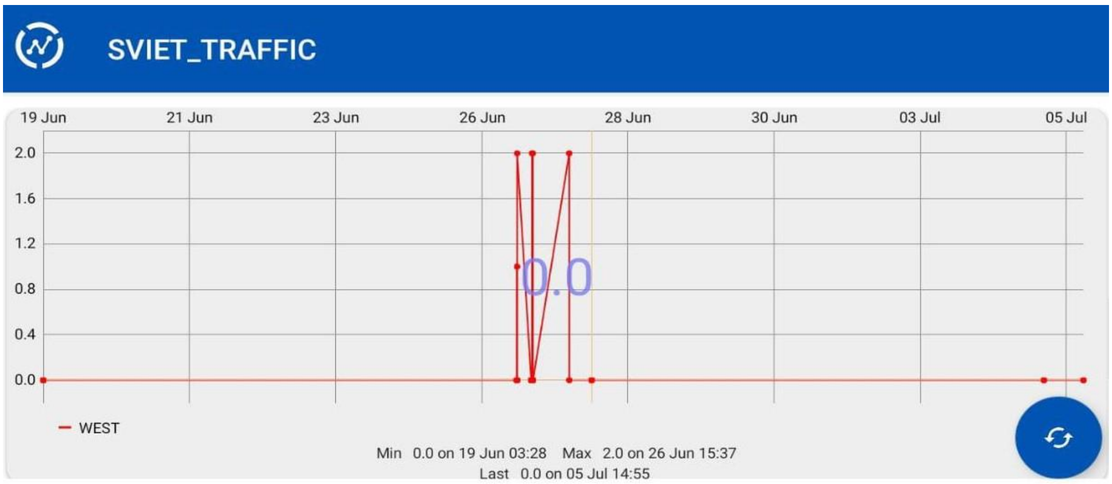



**Figure 4.4 Low Congestion Displayed on LCD display**

By uploading data in cloud the graphs will appear as shown in figure 4.5.



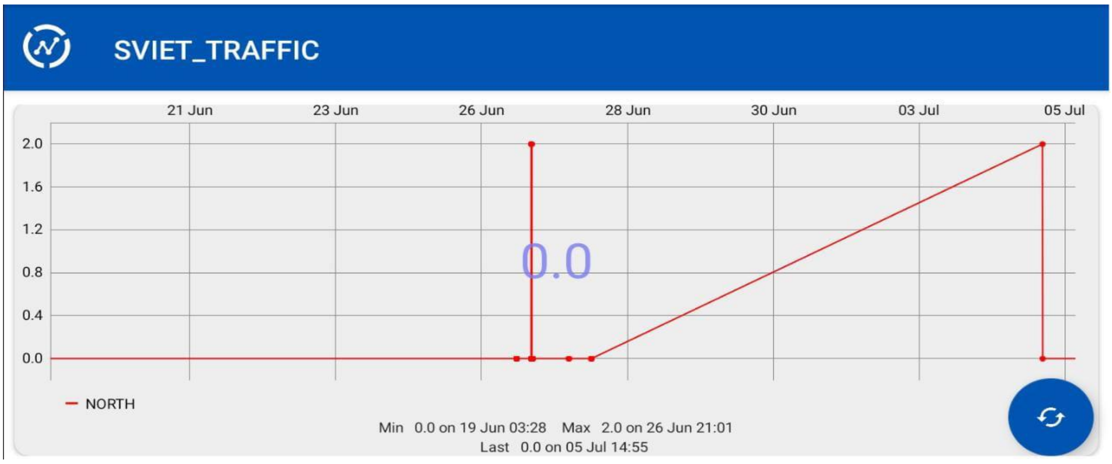




**Figure 4.5: Storing Data in Cloud**

# CHAPTER 5

# CONCLUSION

So by using this it is possible to control congestion level traffic management using IOT. Traffic situation is stored on the cloud in terms of congestion levels at four routes. This is further used for traffic management for improving of traffic. Finally real-time implementation of traffic management is developed.

# REFERENCES

[1]. P. Sadhukan, F. Gazi “An IoT based Intelligent Traffic Congestion Control System for Road Crossings”, IC3IOT, 2018

[2] Costea, I. M. Nemtanu, F. C. Dumitrescu, C. Banu, C. V. Banu, G. S: “Monitoring System with Applications in Road Transport”, SIITME, 2014

[3] S. Javaid, A. Sufian, S. Pervaiz and M. Tanveer, "Smart traffic management system using Internet of Things," 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon-si Gangwon-do, Korea (South), 2018

[4] A. Saikar, M. Parulekar, A. Badve, S. Thakkar and A. Deshmukh, "TrafficIntel: Smart traffic management for smart cities,", International Conference on Emerging Trends & Innovation in ICT (ICEI), Pune, 2017

[5] D.Serpanos, M.Wolf, “Internet-of-Things (IOT) systems, Springer, 2018.

# APPENDIX A

## Installing Raspbian OS using NOOBS

NOOBS stands for Net out of the Box software. NOOBS is an easy operating system installer which contains Raspbian and LibreELEC.

**Step 1 Download the NOOBS zip file**

As shown in Figure A.1.Download NOOBS or NOOBS Lite. Click on the NOOBS zip file and extract the files and copy the contents to a formatted SD card on computer.

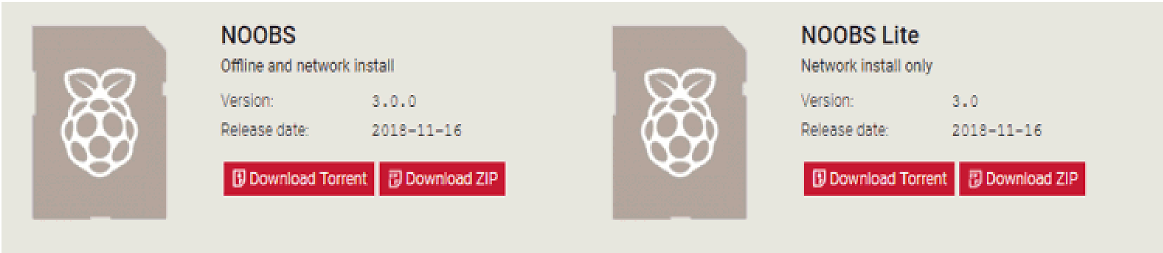



**Figure A.1 NOOBS zip file**

**Step 2 Format the SD card**

For Windows and MAC users, format SD card using SD Association's Formatting Tool. After installing this select card and formatting option.In formatting option always select quick format.

**Step 3 Install Raspbian**

After formatting SD card, copy extracted NOOBS file in SD and after copying files remove SD card. Now insert SD card and power up Raspberry pi. Connect raspberry pi to ethernet. When Raspberry pi turns on select Raspbian install on top left as shown in Figure A.2 and click on yes to confirm it.

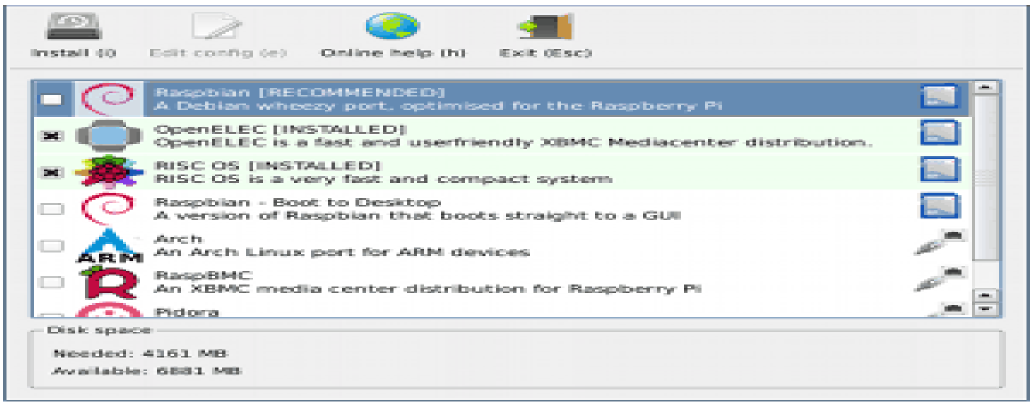


**Figure A.2 Install Raspbian**

After the above procedure OS starts installing. After that reboot the system then OS is installed then the SD card is inserted in the Raspberry Pi.

## Python (Programming Language)

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991. Python features a dynamic type system and automatic memory management. It supports multiple, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. Python is open source software.

## Raspberry pi IDE

An IDE(Integrated Development Environment)is a software application that provides all essential functionalities for software development. IDE consists of source code editor, debugger, compiler. Thonny IDE is used for Raspberry pi which comes pre-installed with Raspberry pi OS desktop version.

# APPENDIX B

## Steps to Connect Raspberry Pi to VNC Viewer

- As shown in Figure B.1. Enter the IP address in VNC Viewer search bar.

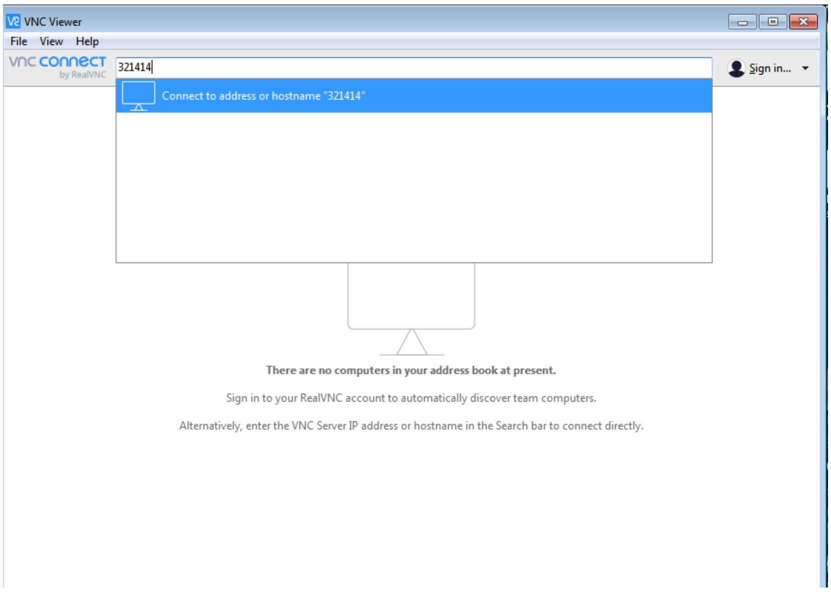



**Figure B.1: VNC Viewer Desktop with IP Address**

- After entering the IP address, a window will appear enter username: pi and password: raspberry as shown in Figure B.2.

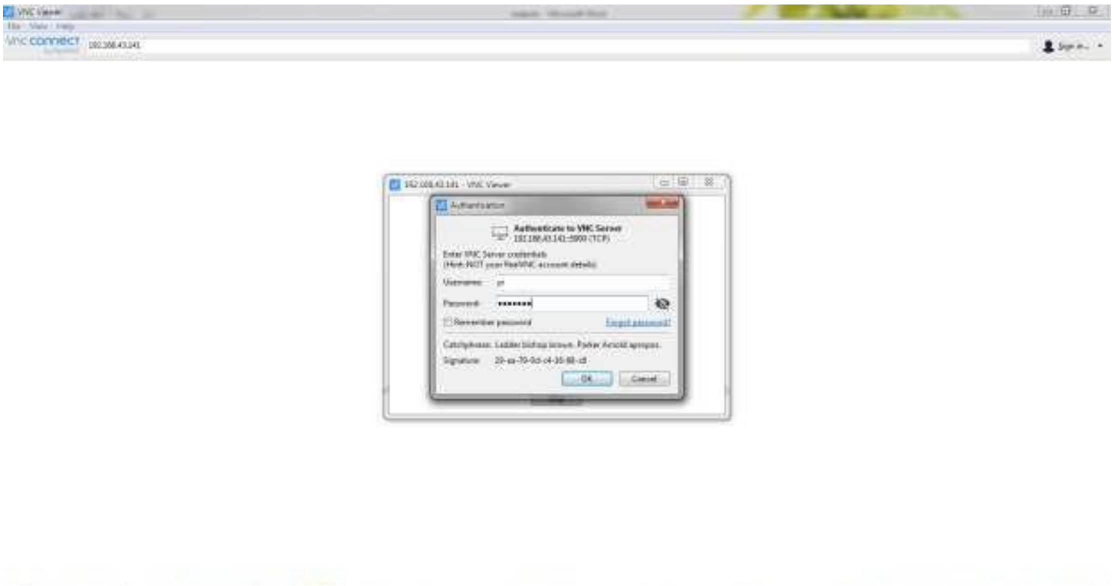


**Figure B.2: Forming a Network**

- After forming a network Raspberry Pi window will appears on the monitor and click on Raspberry pi symbol as shown in Figure B.3

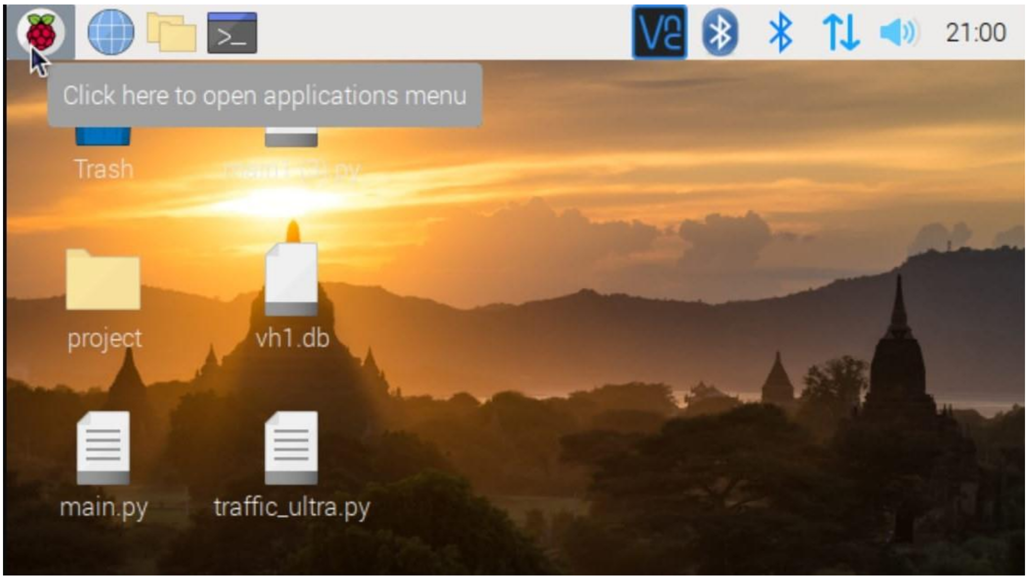



**Figure B.3: Raspberry Pi Desktop**

- Then click on Programming and then click on Python 3(IDLE) as shown in Figure B.4.

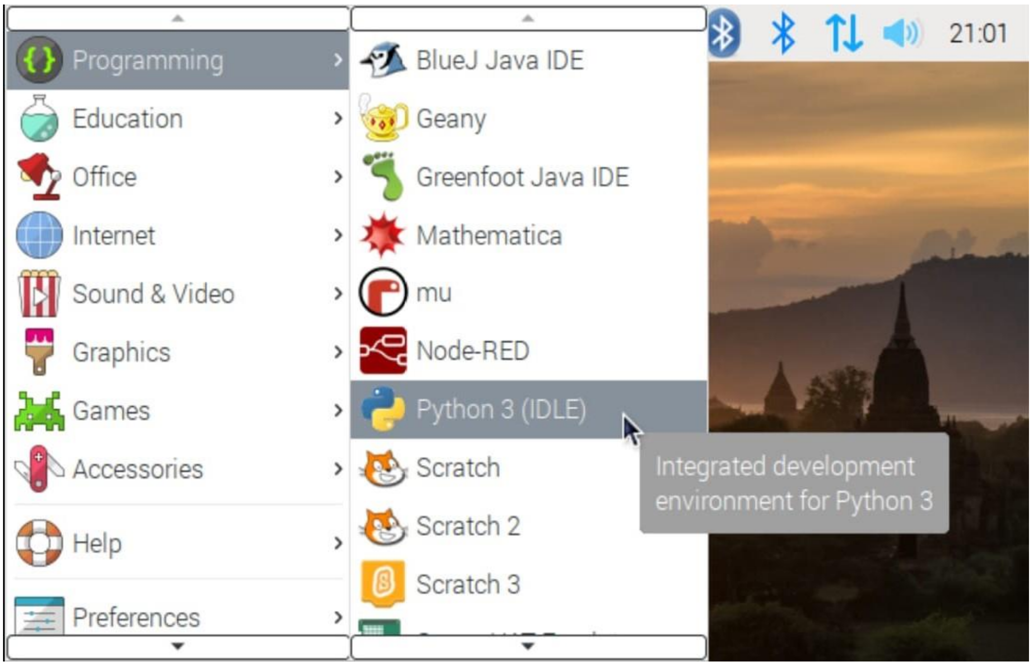



**Figure B.4: Python Window**

- Then Python 3.7.3 Shell window will open as shown in Figure B.5.

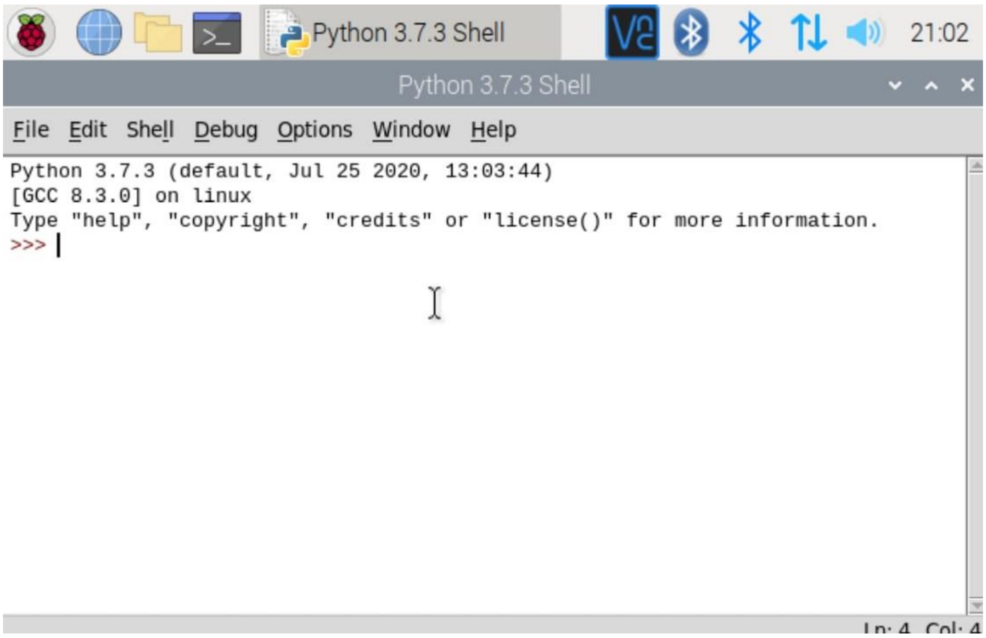



**Figure B.5: Python 3.7.3 Shell Window**

- Then click on file and click on open as shown in Figure B.6

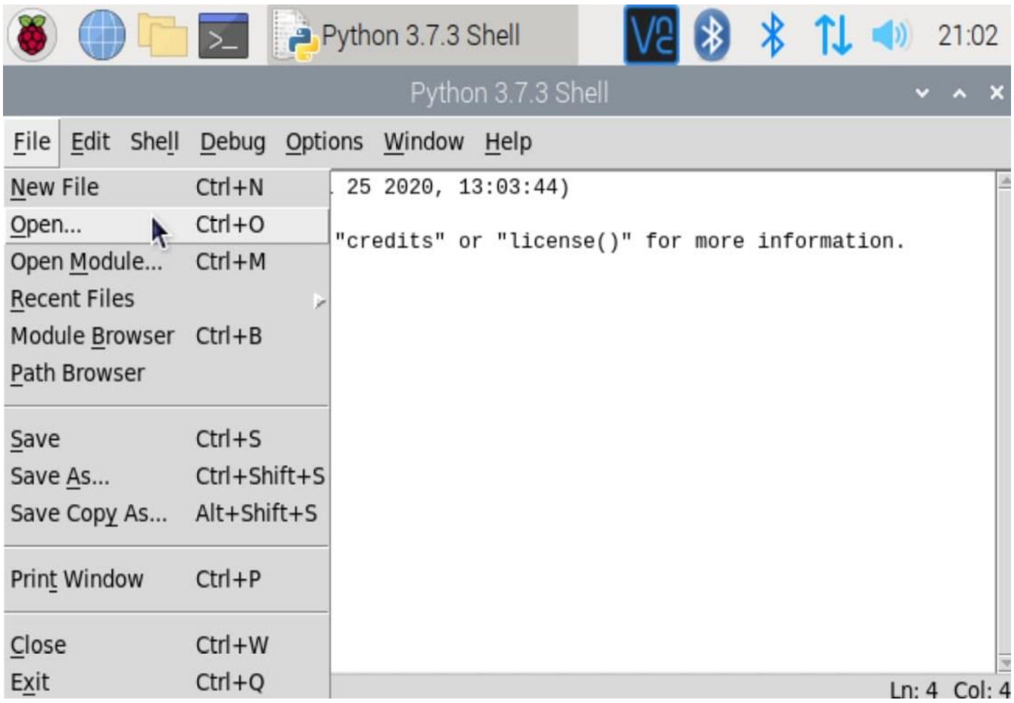



**Figure B.6: File menu in VNC viewer**

- Then directory window will open, click on desktop as shown in Figure B.7.

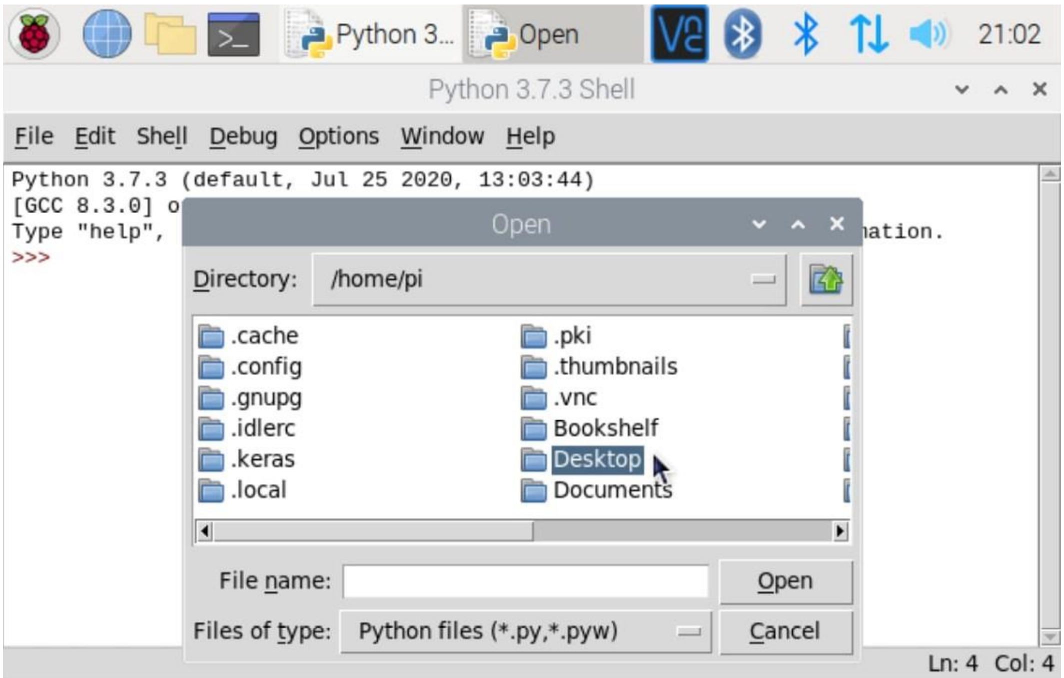



**Figure B.7: Open Window in VNC viewer**

- In directory window click on the traffic_ultra.py folder as shown in Figure B.8

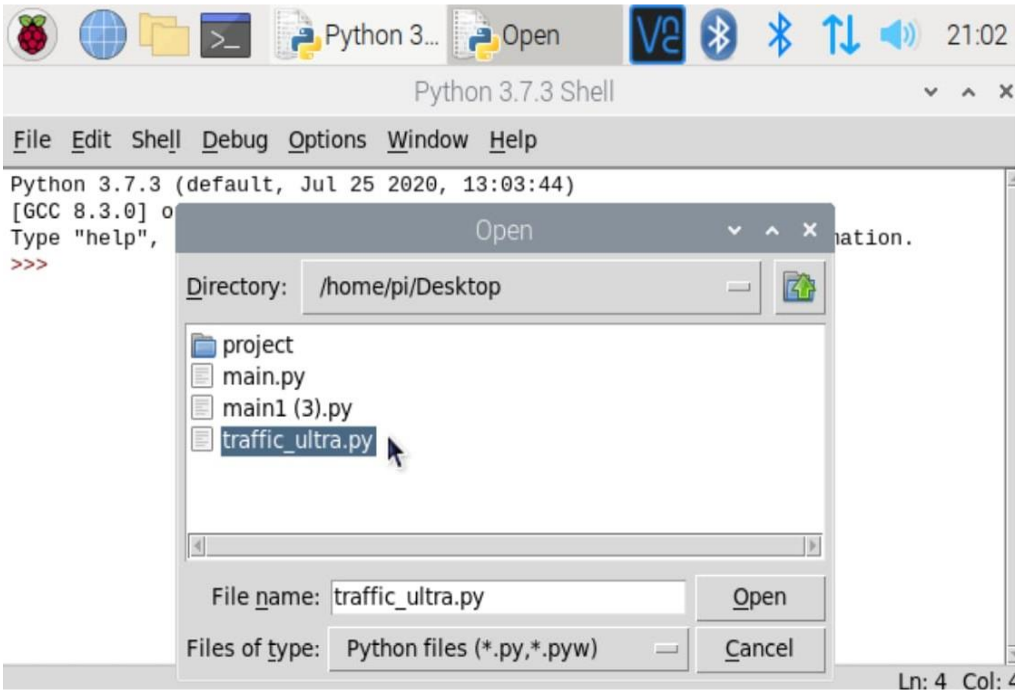



**Figure B.8: traffic_ultra.py window**

- Then program will open on the traffic_ultra.py window, click on the run then run module as shown in Figure B.9.

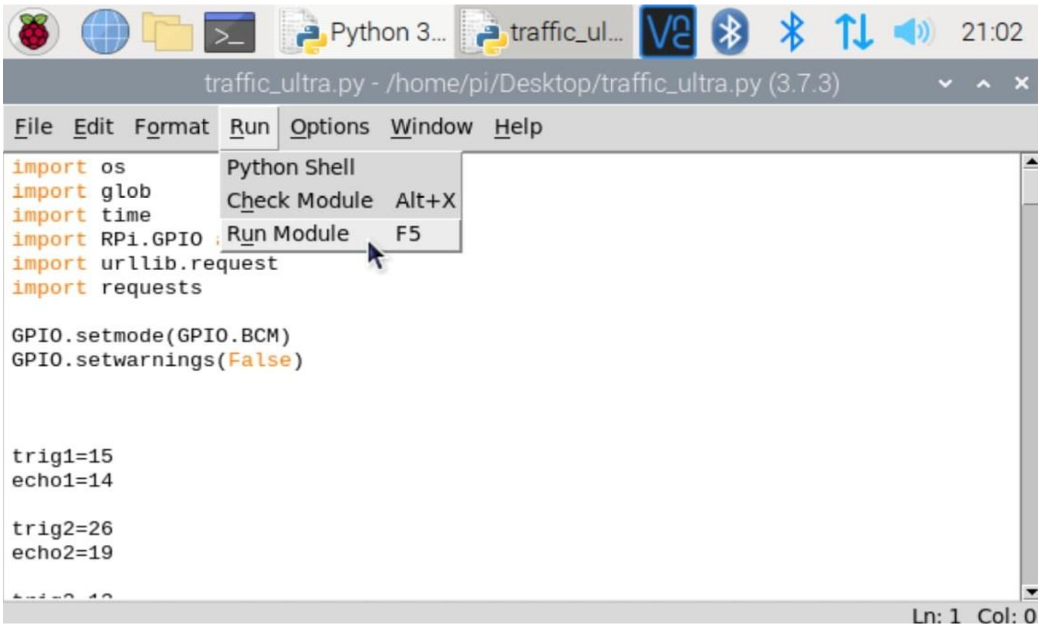



**Figure B.9: Program Window on VNC viewer**

- Then Output will display on the command prompt as shown in Figure B.10

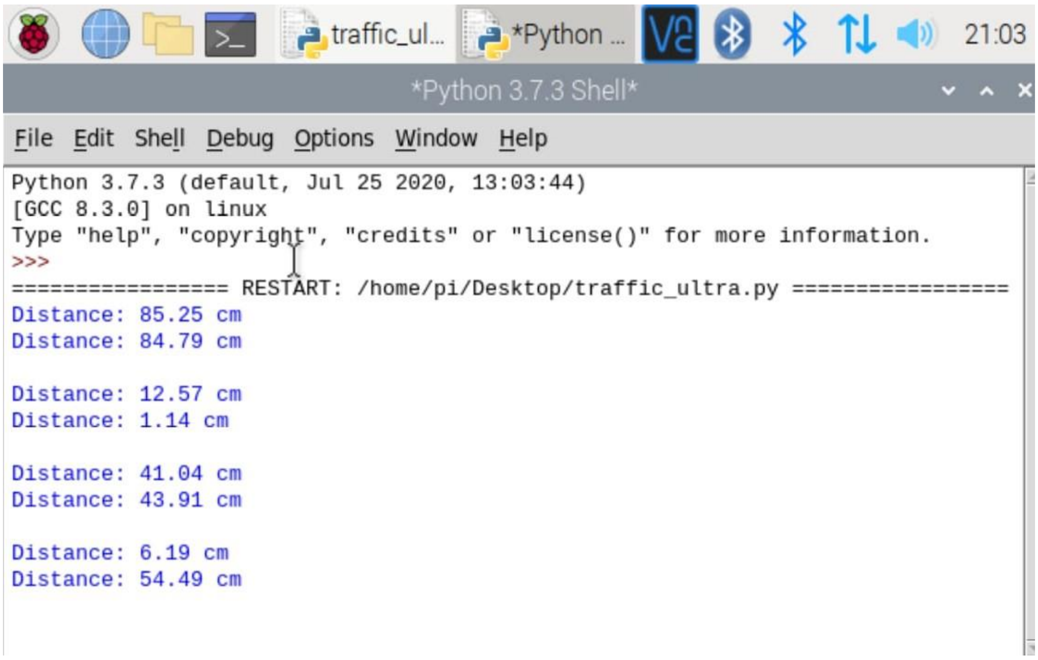



**Figure B.10: Command Prompt**

# APPENDIX C

- Create an account in the thing speak with an email id and then set password as shown in Figure C.1

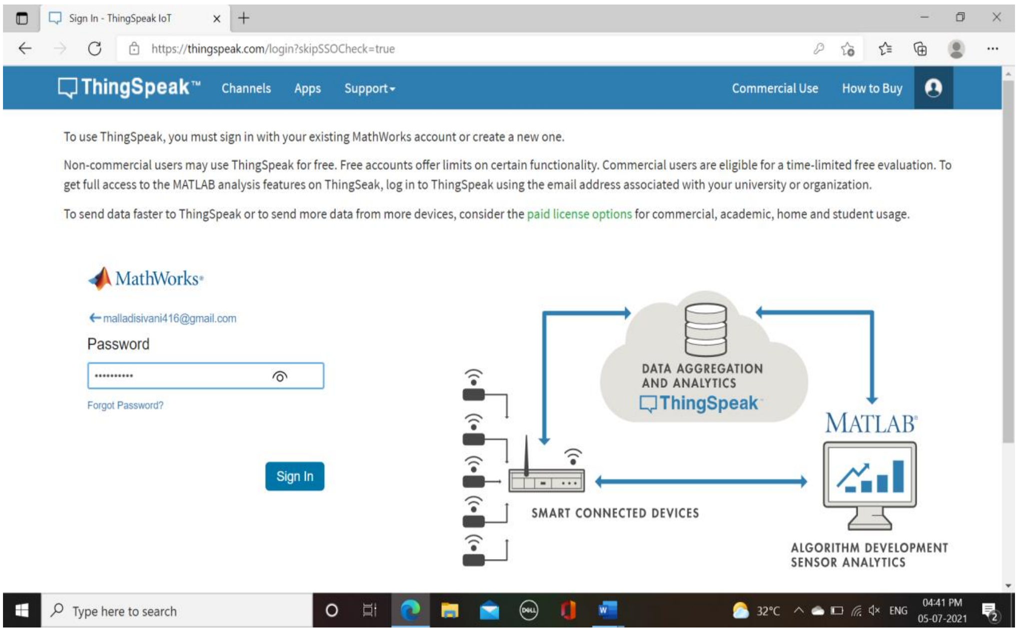



**Figure C.1: Creating an Account in ThingSpeak**

- Create the channel with the project name. Select number of fields and give names to the fields as East, West, North, South as shown in Figure C.2

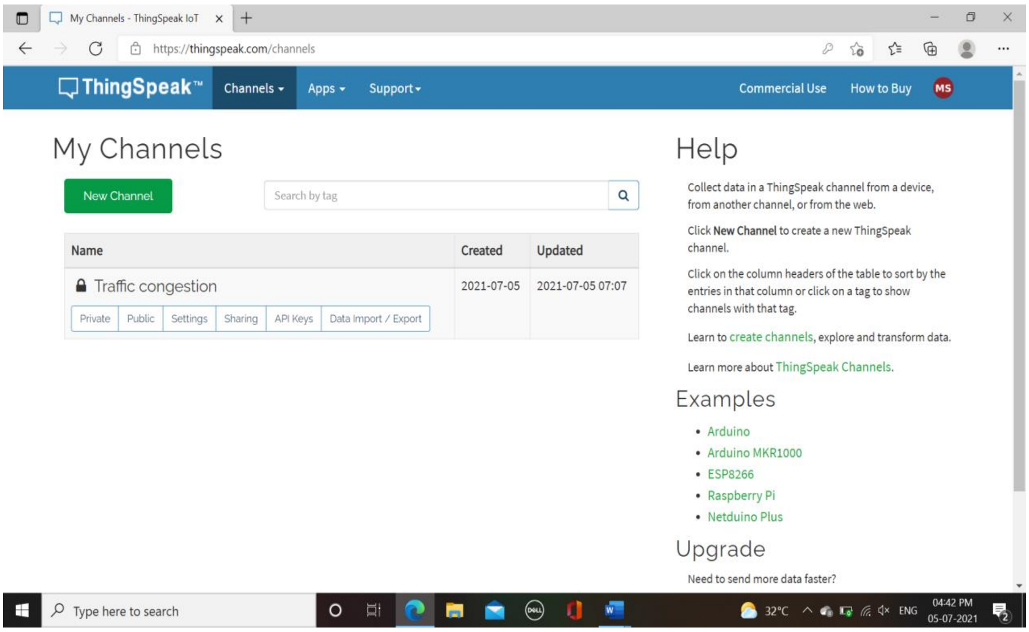



**Figure C.2: Creating a Channel in Thing Speak**